

# ECHTZEITSYSTEME

Vorlesung Wintersemester 2003/2004

Dr. Gerhard Schrott

Institut für Informatik

Lehrstuhl für Echtzeitsysteme und Robotik

Robotics and Embedded Systems

TU München

Anschriften , Sprechstunde

**Prof. Dr.-Ing. habil. Alois Knoll**

Sekretariat: Frau M. Knürr, Frau R. te Vehne

Boltzmannstr. 3, 85748 Garching

Zimmer 03.07.054, Telefon 289-18104

E-mail: [knoll@in.tum.de](mailto:knoll@in.tum.de)



**Dr.-Ing. G. Schrott**

Boltzmannstr. 3, 85748 Garching

Zimmer 03.07.053, Telefon 289-18134

E-mail: [schrott@in.tum.de](mailto:schrott@in.tum.de)

Aktuelles zur Lehrveranstaltung auf  
<http://www6.in.tum.de/>

### **3 SWS Vorlesung im Bereich Informatik II (Technische Informatik)**

Wahlpflichtvorlesung im Gebiet Echtzeitsysteme  
und Robotik

Wahlpflichtvorlesung für Studenten der  
Elektrotechnik und Informationstechnik und für  
Studenten MW Richtung Mechatronic

**Hörerkreis:** Informatiker ab 5. Semester  
Studierende der Fachrichtungen  
Technomathematik,  
Elektrotechnik/Informationstechnik und  
Maschinenwesen (Mechatronik).

**Praktikum zur Vorlesung:** Echtzeitsysteme  
und laufend Diplomarbeiten und Systemprojekte

**Schein:** Soweit im Einzelfall ein Schein benötigt  
wird, ist die Basis dafür eine 30minütige  
mündliche Prüfung. Dies gilt insbesondere für  
Wahlfachscheine für Studentinnen/Studenten  
anderer Fachrichtungen.

**Skript:** Folien zur Vorlesung im Netz auf den  
Lehrstuhlseiten

# Inhaltsverzeichnis

## **1 Echtzeitbetrieb**

- 1.1 Grundlegende Definitionen
- 1.2 Aufgaben Echtzeitsysteme
- 1.3 Arten Rechnerkopplung an Prozeß
- 1.4 Das Prozeßrechensystem als Regelkreis
- 1.5 Anforderungen an Echtzeitsysteme
- 1.6 Historie
- 1.7 Beispiele zur Automatisierung
- 1.8 Einsatzbeispiele
- 1.9 Ziel der Automatisierung
- 1.10 Interdisziplinäres Gebiet
- 1.11 Themen der Prozeßinformatik  
(Echtzeit–Datenverarbeitung)
- 1.12 Funktionen zur Prozeßführung
- 1.13 Prozeßmodelle

## **2 Architekturen von Prozeßrechensystemen**

- 2.1 Klassischer Prozeßrechner
- 2.2 Eingebettete Systeme (embedded systems)
- 2.3 Hierarchische Rechner–Architektur

## **3 Prozeßrechner–Peripherie**

- 3.1 Sensoren und Aktoren
- 3.2 Digital–/Analog–Umsetzer (DAU)
- 3.3 Analog–/Digital–Umsetzer (ADU)
- 3.4 Digital–Ein–/Ausgabe
- 3.5 Echtzeituhren
- 3.6 Mikroperipherik
- 3.7 Identifikationssysteme
- 3.8 Industrieterminals
- 3.9 Spezielle Prozessoren

## **4 Digital Signal Processor (DSP)**

- 4.1 Grundablauf, Motivation
- 4.2 Eigenschaften am Beispiel TMS320C30
- 4.3 Architektur und Programmier–Schnittstelle
- 4.4 Anhang: Ausgewählte Befehle

## **5 Speicherprogrammierbare Steuerungen (SPS)**

- 5.1 Hardware
- 5.2 Betriebsweise einer SPS
- 5.3 Programmierung

## **6 Feldbusprotokolle**

- 6.1 Signalübertragung
- 6.2 Protokolle
- 6.3 Aufgabe und Anforderungen
- 6.4 Verbindungsstrukturen
- 6.5 Buszugang
- 6.6 CAN-Bus
- 6.7 MAP, Überblick
- 6.8 MMS bei MAP
- 6.9 PROFI-Bus
- 6.10 SERCOS-Bus
- 6.11 INTERBUS-S
- 6.12 Busse zur Gebäude-Leittechnik

## **7 Regelungstechnik**

- 7.1 Einführung
- 7.2 Einfache Regler
- 7.3 Explizites Rechenbeispiel für PID-Regler
- 7.4 Lineare Rückführregelung
- 7.5 Anwendungsbeispiel: Balancierter Stab
- 7.6 Maple-Sheet für balancierten Stab

- 7.7 Einführung in die Fuzzy-Logik
- 7.8 Fuzzy-Regelung
- 7.9 Fuzzy-Regelung eines aufrechten (inversen) Pendels (balancierter Stab)
- 7.10 Fuzzy-Regler als Chips

## **8 Echtzeitplanung (real-time scheduling)**

- 8.1 Grundbegriffe
- 8.2 Dimensionierung der Rechnerleistung
- 8.3 Schadenstaxonomie
- 8.4 Echtzeitanforderung am Beispiel Paketverteilungsanlage
- 8.5 Weitere Bezeichnungen
- 8.6 Präzedenzsysteme
- 8.7 Pläne
- 8.8 Planen durch Suchen
- 8.9 Zeitplanung bei Einprozessor-Systemen
- 8.10 Zeitplanung bei Mehrprozessor-Systemen
- 8.11 Anomalien bei nichtpräemptiven Systemen
- 8.12 Zuteilung nach Prioritätszahlen
- 8.13 Zeitplanung periodischer Prozesse

8.14 Vergleich der Planungsverfahren

## **9 Echtzeit-Betriebssysteme**

9.1 Anforderungen

9.2 Beispiele für Produkte

9.3 Schichtenmodell

9.4 Rechenprozesse (tasks, threads)

9.5 Unterbrechungs–Antwortprogramme

9.6 UNIX für Echtzeitaufgaben?

9.7 RT-Linux

9.8 QNX realtime OS

9.9 LynxOS

9.10 VxWorks 5.4

9.11 Windows-CE

## **10 Echtzeitsprachen**

10.1 Bemerkungen zur Historie

10.2 Anforderungen an Echtzeitsprachen

10.3 Modul–Strukturen

10.4 Programmstruktur und Moduln in Ada

10.5 Objektorientierung in ADA95

10.6 Tasks in ADA

- 10.7 Partitionen in ADA 95
- 10.8 Synchronisation
- 10.9 Ausnahmen (exceptions)
- 10.10 Unterbrechungs–Antwortprogramme
- 10.11 Real-Time JAVA
- 10.12 Multi-Agent Distributed Real-Time System
- 10.13 Synchrone Programmier- sprachen:  
ESTEREL
- 10.14 Programmiermethode für Embedded  
Sytems: GIOTTO

## **11 Zuverlässigkeit und Sicherheit**

- 11.1 Definitionen
- 11.2 Fehlertolerante Systeme
- 11.3 Prüfung und Qualitätssicherung von  
Steuerungssoftware
- 11.4 Fehlererkennung und –behandlung
- 11.5 Hardware–Redundanz
- 11.6 Zuverlässige Software
- 11.7 Software-Redundanz
- 11.8 Maßnahmen zur Sicherheit

# Literatur

- [1] Adam, J.A. : The Seargant York Gun: a Massive Misfire. IEEE Spectrum, 24(2):28–35, 1987.
- [2] Adam, J.A. : Technology combats disabilities. IEEE Spectrum, 31(10):24–26, 1994.
- [3] Avizienis, A. : Toward Systematic Design of Fault-Tolerant Systems. IEEE Computer, 30(4):51–58, 1997.
- [4] Barnes, J. : **Programming in Ada 95**. Addison-Wesley, 1995.
- [5] Bell, T.E. , Esch, K. : The Fatal Flaw in Flight 51-L. IEEE Spectrum, 24(2):36–51, 1987.
- [6] Bell, T.E. et al.: Special Report: Designing and Operating a Minimum Risk System. IEEE Spectrum, 26(6), 1989.
- [7] Bell, T.E. : Electronic and the Stars. IEEE Spectrum, 32(8):16–24, 1995.

- [8] Bender, K. : PROFIBUS - Der Feldbus für die Automation. Carl Hanser Verlag, 2 Aufl., 1992.
- [9] Berry, G. , Gonthier, G. : The ESTEREL Synchronous Language: Design, Semantic, Implementation. Journal of Science of Computer Programming, 19(2):87–152, 1992.
- [10] Blazewicz, J. et al.: Scheduling in Computer and Manufacturing Systems. Springer-Verlag, 1994.
- [11] Bolch, G. , Vollath, M.M. : Prozeßautomatisierung. Teubner, 2 Aufl., 1991.
- [12] Burns, A. , Wellings, A. : Real-Time Systems and their Programming Languages. Addison-Wesley, 1992.
- [13] Conrads, D. : Datenkommunikation. Vieweg-Verlag, 1993.
- [14] Dagnelie, G. , Massof, R.W. : Toward an Artificial Eye. IEEE Spectrum, 33(5):20–29, 1996.

- [15] Douglas, B.P. : **Real-Time UML: Developing Efficient Objects for Embedded Systems.** Addison-Wesley, 1997.
- [16] Driankov, D. , Hellendoorn, H. , Reinfrank, M. : **An Introduction to Fuzzy Control.** Springer-Verlag, 2 Aufl., 1996.
- [17] Etschberger, K. : **CAN Controller–Area–Network.** Carl Hanser Verlag, 1994.
- [18] Färber, G. : **Prozeßrechentchnik.** Springer-Verlag, 2 Aufl., 1992.
- [19] Gemmer, A. : **Risk Management: Moving Beyond Process.** IEEE Computer, 30(5):33–43, 1997.
- [20] Gomaa, H. : **SoftwareDesign Methods for Concurrent and Real-Time Systems.** Addison-Wesley, 1993.
- [21] Görke, W. : **Fehlertolerante Rechensysteme.** Oldenbourg Verlag, 1989.
- [22] Haag, S. , Raja, M.K. , Schkade, L.L. : **Quality Function Deployment Usage in Software Development.** Communications of the ACM, 39(1):41–49, 1996.

- [23] Halang, W. , Konakovsky, R. :  
**Sicherheitsgerichtete Echtzeitsysteme.**  
Oldenbourg Verlag, 1999.
- [24] Halbwachs, N. : Synchronous Programming  
of Reactice Systems. Kluwer Academic  
Publishers, 1993.
- [25] Heidepriem, J. : **Prozessinformatik 2.**  
Oldenbourg Verlag, 2001.
- [26] Herrtwich, R.G. , Hommel, G. : **Kooperation  
und Konkurrenz - Nebenläufige, verteilte und  
echtzeitabhängige Programmsysteme.**  
Springer-Verlag, 1989.
- [27] Hohler, B. , Villinger, U. : Normen und  
Richtlinien zur Prüfung und  
Qualitätssicherung von Steuerungssoftware.  
Informatik-Spektrum, 21(2):63–72, 1998.
- [28] Hüsener, Th. : **Entwurf komplexer  
Echtzeitsysteme - State of the Art.** BI  
Wissenschaftsverlag, 1994.
- [29] Jacobson, E. : **Einführung in die  
Prozeßdatenverarbeitung, 2. Aufl.** Carl  
Hanser Verlag, 1996.

- [30] Jakoby, W. : Automatisierungstechnik - Algorithmen und Programme. Springer-Verlag, 1996.
- [31] Jurgen, R.K. : The Electronic Motorist. IEEE Spectrum, 32(3):37–48, 1995.
- [32] Kahlert, J. : Fuzzy Control für Ingenieure. Vieweg-Verlag, 1995.
- [33] Kasser, J. : Applying Total Quality Management to Systems Engineering. Artech House Publishers, 1995.
- [34] Kassakian, J.G. et al.: Automotive Electrical Systems circa 2005. IEEE Spectrum, 33(8):22–27, 1996.
- [35] Kavi, K.M. (Hrsg.): Real-Time Systems: Abstraction, Languages, and Design Methodologies. IEEE Computer Society Press, 1992.
- [36] Kratzer, K.P. : ADA, Eine Einführung für Programmierer. Carl Hanser Verlag, 1996.
- [37] Laplante, P.A. : Real-Time Systems: Design and Analysis. IEEE Computer Society Press, 1992.

- [38] Lauber, R. : Prozeßautomatisierung I. Springer-Verlag, 2 Aufl., 1989.
- [39] Leveson, N.G. , Thurner, C.S. : An Investigation of the Therac-25 Accidents. IEEE Computer, 26(7):18–41, 1993.
- [40] Pham, Hoang (Hrsg.): Fault-Tolerant Software Systems: Techniques and Applications. IEEE Computer Society Press, 1992.
- [41] Rzehak, H. (Hrsg.): **Echtzeitsysteme und objektorientierte Entwurfstechniken**. Vieweg-Verlag, 1996.
- [42] Schiebe, M. , Pferrer, S. : Real-Time Systems: Engineering and Applications. Kluwer Academic Publishers, 1992.
- [43] Schwarz, K. : Manufacturing Message Specification MMS. Automatisierungstechnische Praxis atp 31, pp. 23–29, 1989.
- [44] Siegert, H.-J. : Betriebssysteme - Eine Einführung. Oldenbourg Verlag, 4 Aufl., 1998.

- [45] Sweet, W. : The Glass Cockpit. IEEE Spectrum, 32(9):30–38, 1995.
- [46] Voges, U. : Software - Diversität und ihre Modellierung. Informatik–Fachberichte 224, Springer-Verlag, 1989.
- [47] Zhu, H. , Hall, P.A.V. , May, J.H.R. : Software Unit Test Coverage and Adequacy. ACM Computing Surveys, 29(4):366–427, 1997.
- [48] Zöbel, D. , Albrecht, W. : **Echtzeitsysteme – Grundlagen und Techniken**. Int. Thomson Publ., 1995.

- Zeitschriften:
  - ★ atp–Automatisierungstechnische Praxis
  - ★ IEEE Real-Time Systems Journal
  - ★ Embedded Systems Programming Magazine
  - ★ Elektronik
- Tagungen:
  - ★ GI Fachgruppe Echtzeitprogrammierung  
PEARL
  - ★ IEEE Real-Time Systems Symposium
  - ★ IFAC/IFIP Workshop on Real-Time  
Programming
  - ★ Euromicro Conference on Real-Time  
Systems
  - ★ Embedded Systems (Messe München)
- Ergänzende Vorlesungen:
  - ★ Betriebssysteme
  - ★ Rechnernetze
  - ★ Mikroprozessoren
  - ★ Zuverlässigkeit und Fehlertoleranz von  
Rechensystemen
  - ★ Software-Technik

# 1 Echtzeitbetrieb

## 1.1 Grundlegende Definitionen

(nach DIN 66201 Prozeßrechen-systeme)

- **Prozeß**

Umformung und/oder Transport von Materie, Energie und/oder Information (deterministisch oder stochastisch).

Anmerkung: Prozeß in der Informatik

üblicherweise: Ablauf eines Programms, wenn dieser Ablauf Verwaltungseinheit des Betriebssystems ist.

- **Technischer Prozeß**

Prozeß, dessen Zustandsgrößen mit technischen Mitteln gemessen, gesteuert und/oder geregelt werden können.

- **Prozeßführung**

Erreichen eines vorgegebenen Ziels aus Kenntnis der Istwerte unter Berücksichtigung der Kapazitäten; flexible Anpassung an kleine Störungen; Anzeige in Prozeßwarten.

Die Führung technischer Prozesse kombiniert Planung, Steuerung und Regelung.

- **Steuerung**

Einwirkung mittels Stellgrößen in eine gewünschte vorausgeplante Richtung.

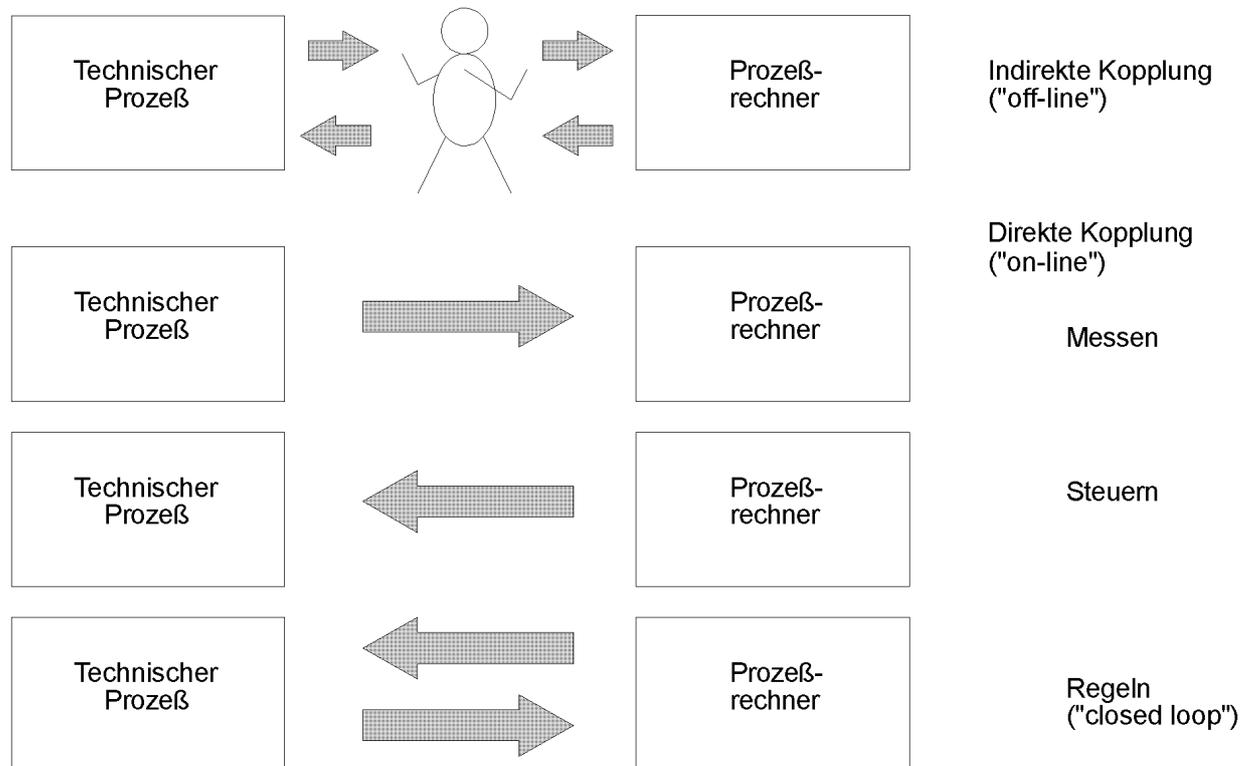
Beispiel: Heizung um 6:30 Uhr einschalten.

- **Regelung**

Überwachung und Minimierung der Abweichungen zwischen Ist- und Sollwerten nach gegebenen Kriterien.

Beispiel: Kesseltemperatur auf 43 Grad halten.

## 1.2 Aufgaben Echtzeitsysteme

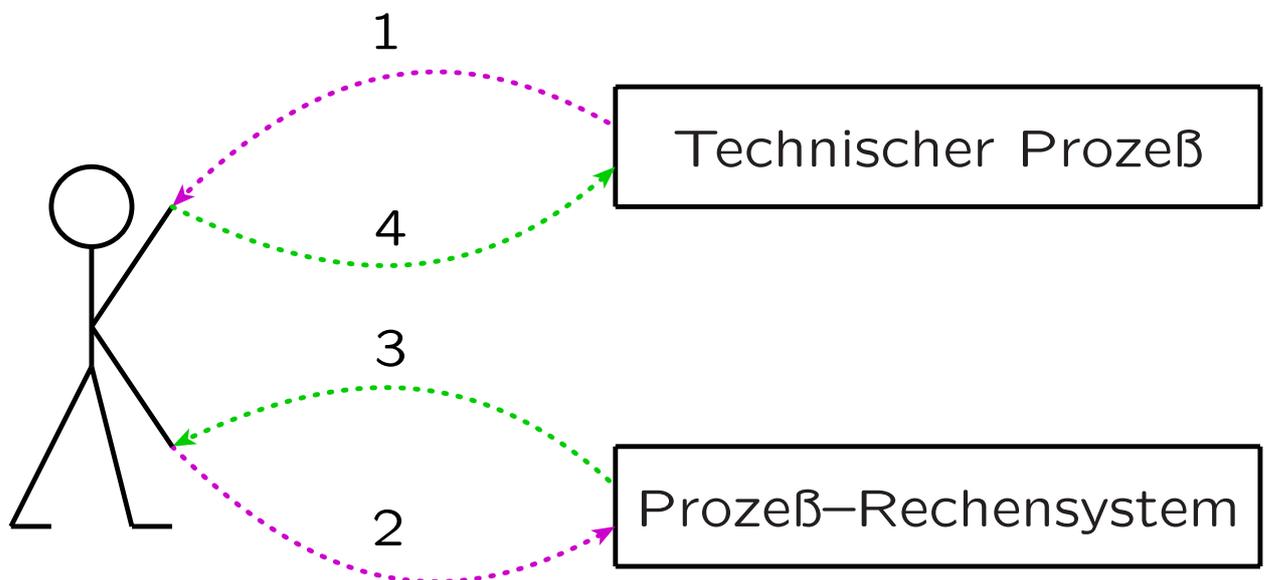


Je nach Komplexität:

- Datensammlung über den technischen Prozeß und Datenaufbereitung
- Steuerung technischer Prozesse
- Regelung technischer Prozesse

## 1.3 Arten Rechnerkopplung an Prozeß

- **Indirekte Kopplung** (off line–Betrieb)
  - ★ Zeitliche und gerätetmäßige Entkopplung
  - ★ Bedienung durch Personal, z.B.
    - Versuchsauswertungen
    - Medizin EKG
    - Profilaufnahme von Vorgängen
    - Qualitätsstatistik
  - ★ Nachteil: fehleranfällig, langsam



1: Messen

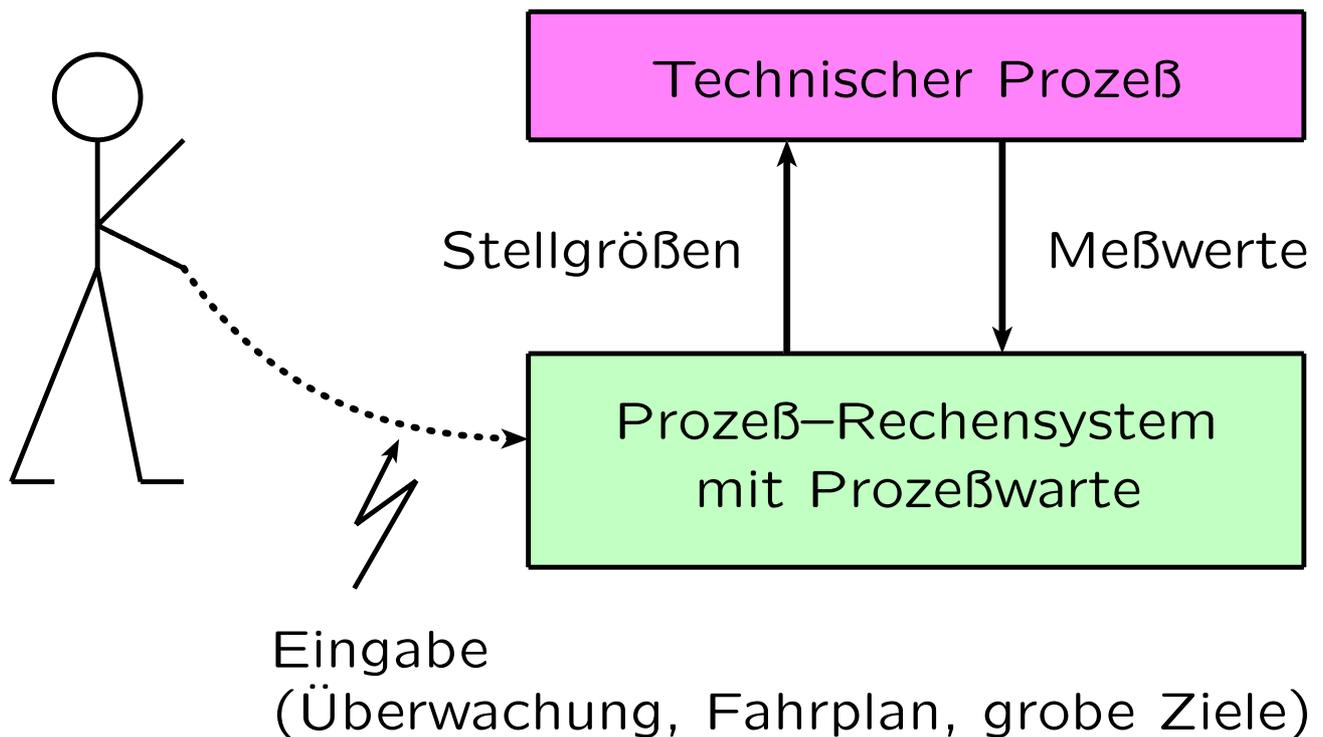
2: Zustandsdaten

3: Steuerungsvorschlag

4: Stellen

- **Direkte Kopplung** (on line–Betrieb)

Der Rechner übernimmt die Meßwerte direkt vom technischen Prozeß und greift mittels Stellgrößen ein



- ★ **offene Prozeßkopplung**

(on line open loop control)

Die Eingabedaten (Meßwerte) haben keine Rückwirkung auf die Ausgabedaten (Stellwerte), z.B. Stimulation eines Versuchs mit festen Eingabewerten, Protokollierung (z.B. Prüfstände), Qualitätskontrolle (Profil eingeben und Reaktion beobachten).

★ geschlossene Prozeßkopplung

(on line closed loop control)

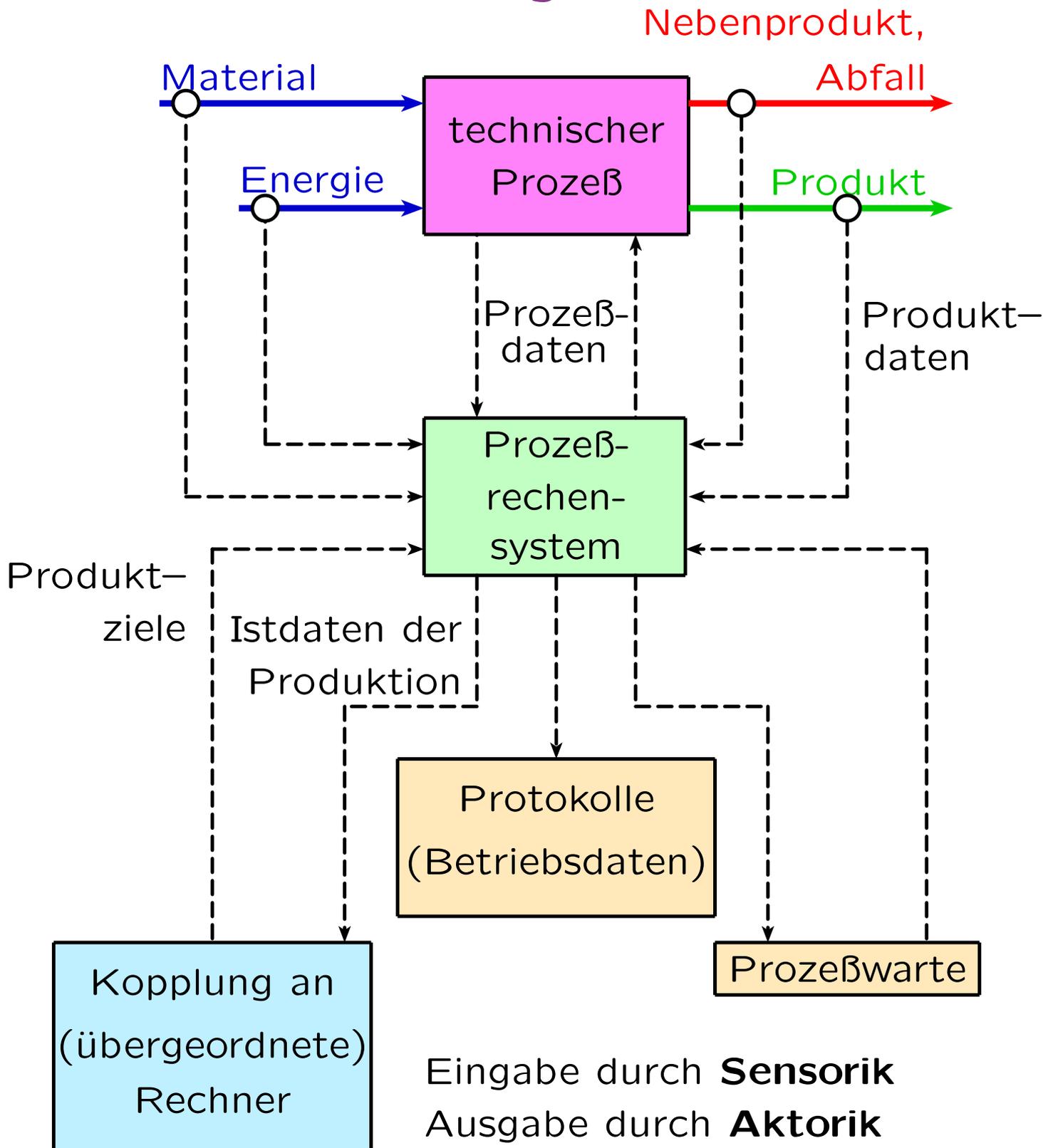
Die Eingabedaten (Meßwerte) beeinflussen die Ausgabedaten (Stellwerte), z.B.

Rückführregelung, nachgeführtes oder neues Betriebsziel.

Stichwort DDC (**d**irect **d**igital **c**ontrol)

"dedicated systems": viele kleine Regelkreise mit vorgefertigten Programmbausteinen zum Konfigurieren und mit automatischer Dokumentation.

# 1.4 Das Prozeßrechnungssystem als Regelkreis



## 1.5 Anforderungen an Echtzeitsysteme

- **Echtzeitfähigkeit**  
harte Zeitbedingungen, kurzfristige Reaktionszeiten
- Anschluß vielfältigster **Peripherie**  
(Geräte, Sensorik, Aktorik)
- **Algorithmische** Erfassung der Struktur des technischen Prozesses
- **Mensch–Maschine–Schnittstelle**  
für Datenrepräsentation, Datensammlung und ggf. Datenauswertung für die Steuerung
- **Zuverlässigkeit, Sicherheit**
- **Fehlertoleranz**
- **Intelligenz**, um Vorgehenvorschläge in bestimmten Situationen, insbesondere im Störfall, machen zu können

## 1.6 Historie

Wichtige Entwicklungsschritte der Echtzeitsysteme:

- Bis 1960 Automatisierung einzelner Regelstrecken (Relaistechnik, Transistorschaltungen, Analogrechner)
- Ab ca. 1960 Einsatz von Digitalrechnern  
Begriff "Prozeßrechner" (historisch, unpräzise) für die damaligen Eigenentwicklungen:
  - ★ Kurzwortmaschinen (8- oder 12-Bit-Worte)
  - ★ Zentralgerät
  - ★ robuste Ausführung
  - ★ wenig Speicher (teuer!),  
Bit-Programmierung
  - ★ spezielle EA-Kanäle
  - ★ kein allgemeines Betriebssystem, sondern  
anwendungsspezifische Ablaufsteuerungen

- Produktion spezieller "Prozeßrechner" der Firma DEC
  - ★ pdp8 (1965)
  - ★ pdp11 (1970)
  - ★ VAX (1977)
- Ab ca. 1976 Massenproduktion von Mikrocomputern, insbesondere auch für "embedded systems" aber auch klassische "Prozeßrechner" wie LSI11, MicroVAX (1982)
  - ★ Führung selbst kleinster technischer Prozesse durch Rechner (Tisch-PC; kundenspezifische Chips (ASIC, application specified IC) in Geräten und Maschinen, embedded systems)
  - ★ globale Regelungs- und Überwachungsstrategien
- seit Anfang 90er Standardrechner (Workstations und PC) mit Echtzeit-Betriebssystemen
  - ★ Bussysteme und Rechnernetze
  - ★ CIM (computer integrated manufacturing)
  - ★ Expertensysteme zur Prozeßführung



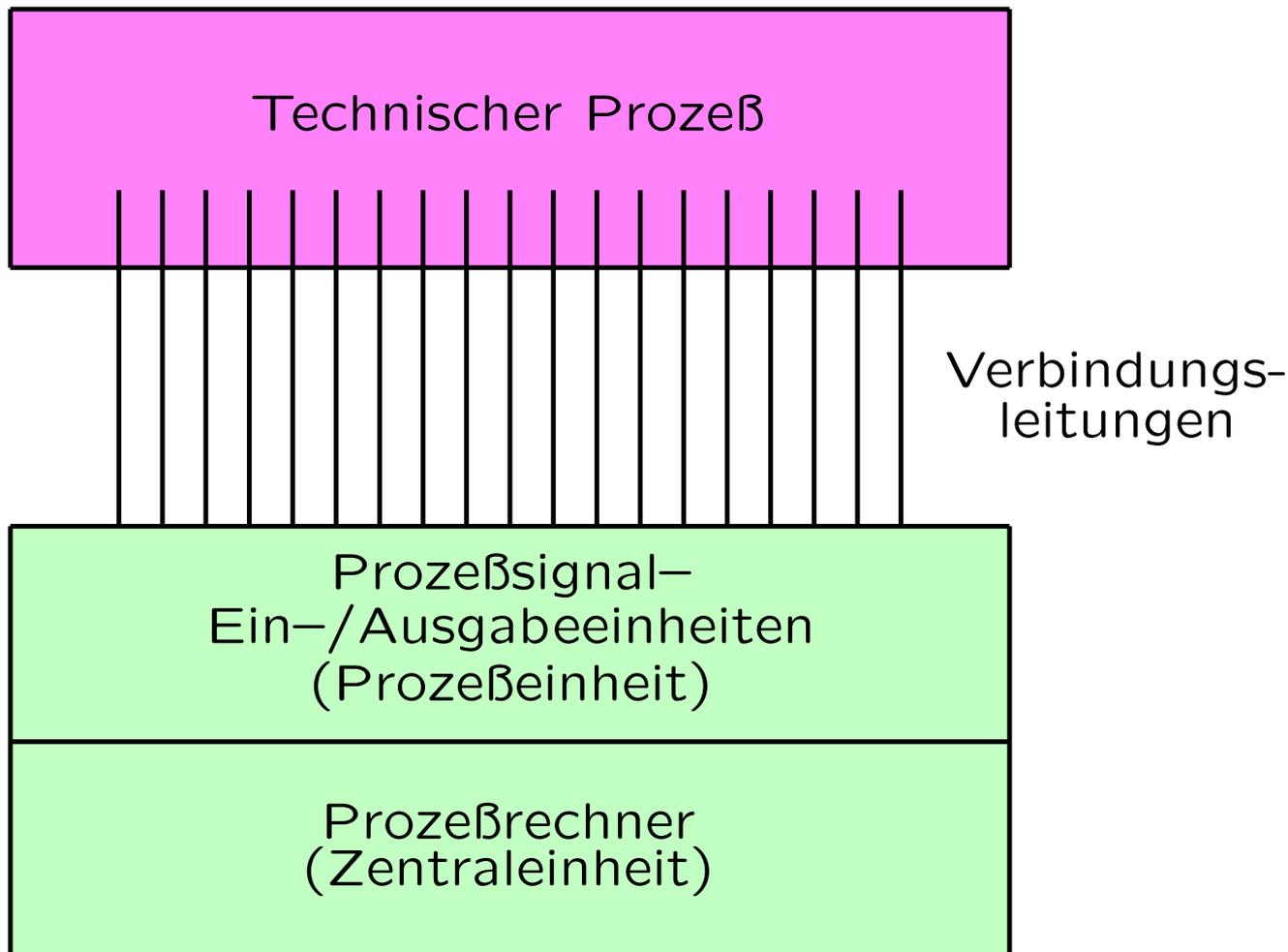
pdp11-20



VAX11-780

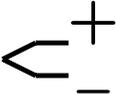
## 1.7 Beispiele zur Automatisierung

konventionelles Prozeßrechensystem,  
Blockschaltbild

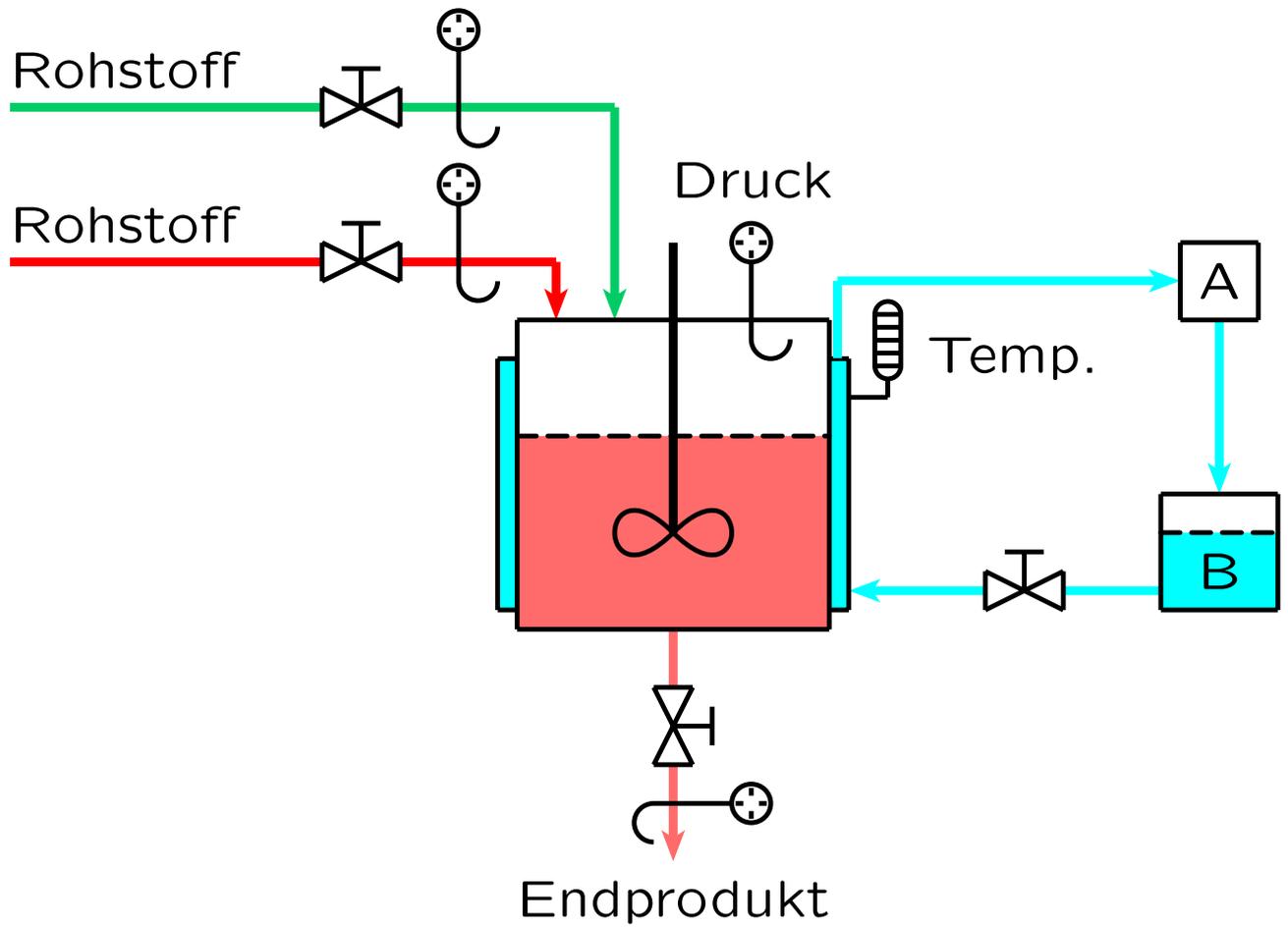


- Ein-/Ausgabeeinheiten und ihre Symbole

	Lichtschranke
	Füllstandsmelder
	Thermostat
	Endschalter
	Taster
	Schrittmotor, Servo-Motor
	Ventil, handverstellbar
	Magnetventil
	Alarm
	Lampe

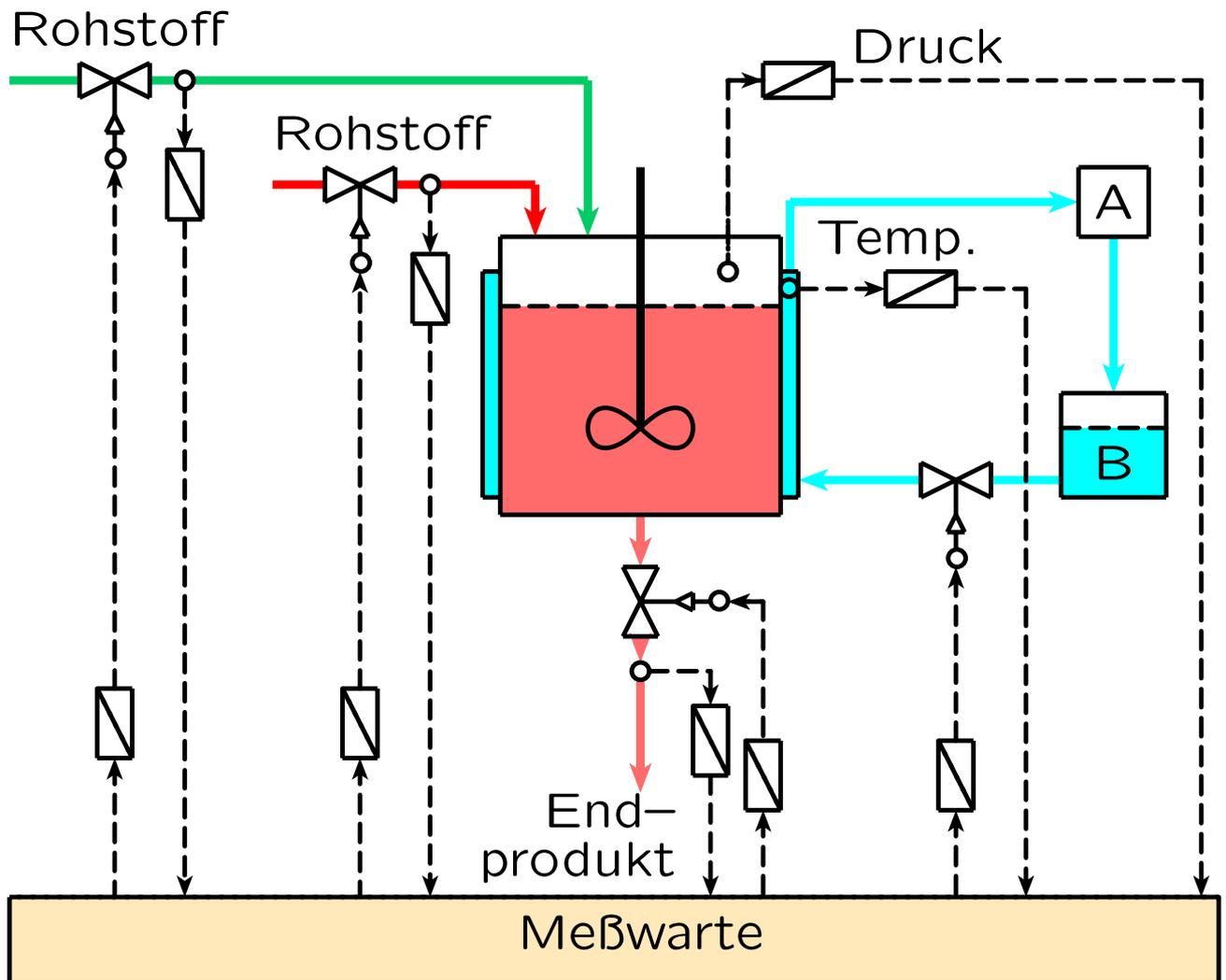
	Heizung
	Schütz
	Weggeber
	Widerstandsthermometer
	Thermoelement
	Hallgenerator
	Tachogenerator
	XY-Schreiber
	automatisches Testsystem
	Meßgerät
	Meßfühler
	Meßwertumformer, Signalumformer
	Regler mit Ist-/Sollwerteingang
	Stellglied (Stellantrieb+Stellorgan)

- Automatisierung eines Rührkesselreaktors, Stufe 1: handgesteuert



A = Kühlung  
B = Kühlflüssigkeit

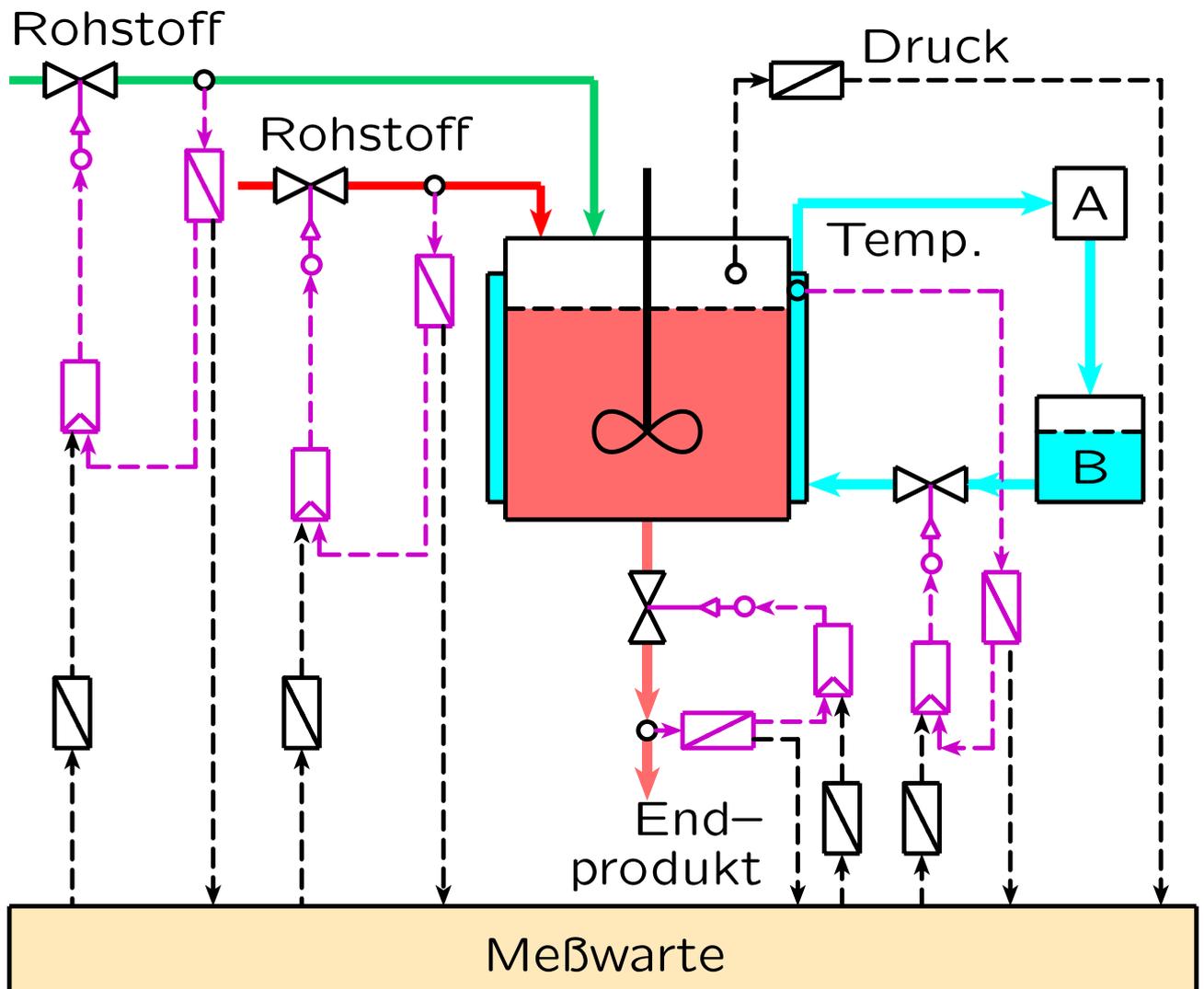
- Automatisierung eines Rührkesselreaktors, Stufe 2: wartengesteuert



A = Kühlung  
B = Kühlflüssigkeit

Meßwarte: im einfachsten Fall abgesetzte Bedienung, sonst Prozeßrechner mit Steuerung oder Regelung

- Automatisierung eines Rührkesselreaktors, Stufe 3: wartengesteuert, lokale Regler

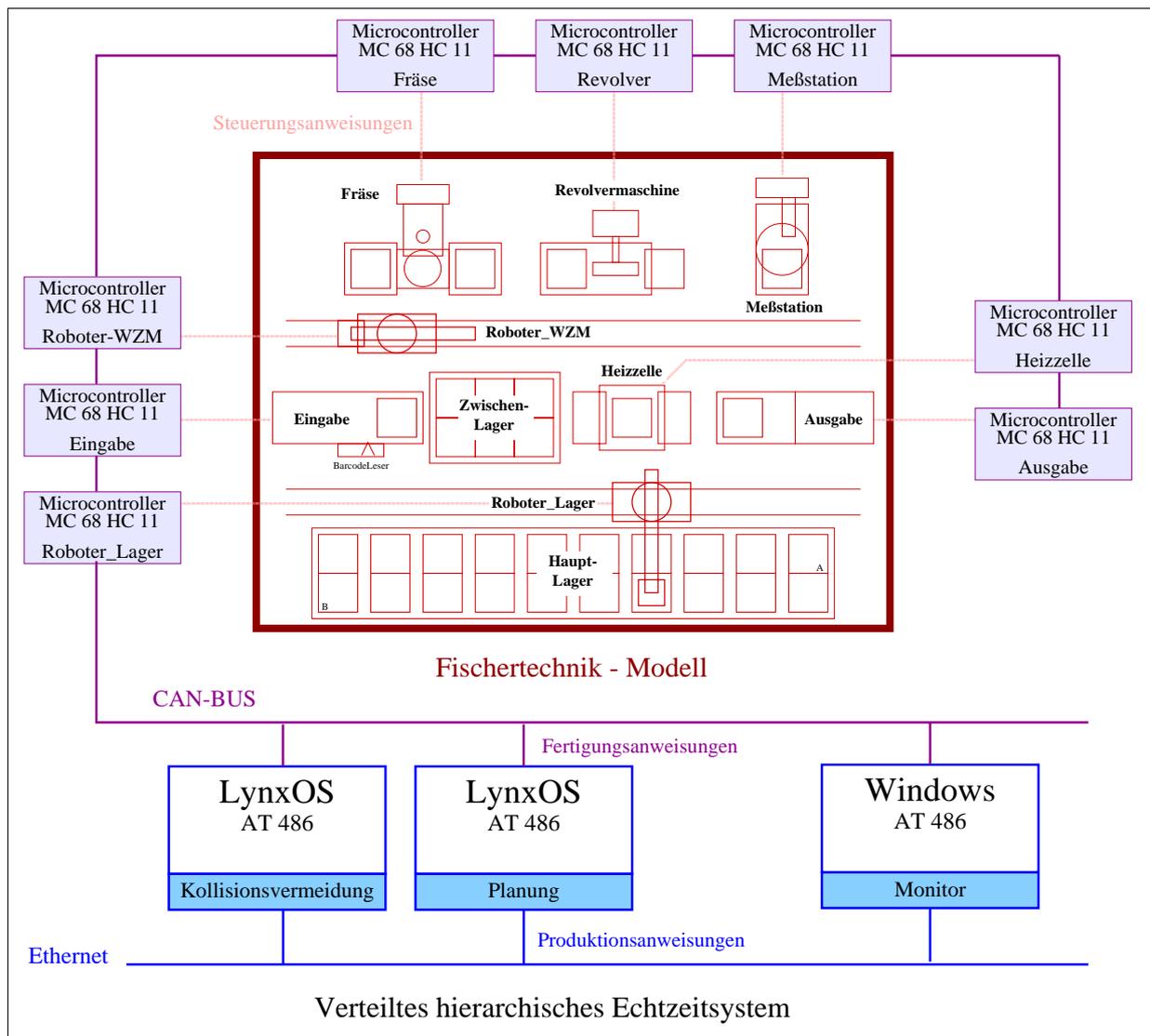


A = Kühlung  
B = Kühlflüssigkeit

Lokale Regler können als unterste Ebene einer hierarchischen Prozeßführung betrachtet werden

## Moderne Entwicklungen:

- Zentraler Prozeßrechner durch vernetzte Rechensysteme ersetzt
- lokale Prozeßrechner bei den einzelnen Komponenten des Prozesses
- heutige Kosten der Rechner erlaubt Regelungen durch Rechner
- Punkt-zu-Punkt-Verbindungen zu den Ein-/Ausgabeeinheiten durch Feldbusse (Netze) ersetzt
- intelligente automatisierte Geräte  
Meßgeräte,  
Robotersteuerung,  
NC-Steuerung,  
SPS



## Beispiel der verteilten Steuerung einer Fertigungsanlage

## 1.8 Einsatzbeispiele

- Steuerung einzelner Geräte und Maschinen
  - ★ Geräte des täglichen Bedarfs  
(Mikrowellenherd, CD-Spieler, Kamera, Video-Recorder, Anti-Blockier-System, Anti-Schlupf-Regelung und Elektronisches Stabilitätsprogramm im Auto)
  - ★ Fertigungsmaschinen (Werkzeugmaschine, Roboter)
  - ★ Verkehrsmittel (Aufzug, Magnetbahn, Flugzeug)
  - ★ naturwissenschaftlich-medizinische Geräte  
(Massenspektrometer, Tomograph)
  - ★ militärische Geräte (Panzer, Radarsystem, Rakete)
- Führung komplexer technischer Vorgänge
  - ★ Steuerung von Fertigungsanlagen, Walzwerken, etc.
  - ★ Kontrolle und Überwachung von Kraftwerksanlagen (Bsp. 700 MW Kohlekraftwerk)
  - ★ Umweltschutz-Warnanlagen

- ★ Verkehrslenkung
- Konkrete Einzelbeispiele
  - ★ Boeing 777: über 1000  
Rechner/Mikrocontroller pro Flugzeug  
z.B. 10 Stück fürs Bremssystem
  - ★ 700 MW Kohlekraftwerk:  
750 + 1250 Antriebe  
3000 + 3000 Meßwerte  
2200 + 5000 Zustandsmeldungen  
4500 + 15000 Alarmmeldungen  
Hauptblock + Nebenanlagen Entschwefelung
  - ★ Channel Tunnel: ca. 100 Rechner und  
Workstations, ca 200 Mikrocontroller
  - ★ Herzschrittmacher:  
500 000 Lines of Code (loc) in C
  - ★ TGV-Subsystem:  
380 000 loc, davon 220 000 loc in Ada
  - ★ Autoradio: 1 MB Code
  - ★ Fernsehgerät: Übergang zum Multitasking

## 1.9 Ziel der Automatisierung

- viele (oft kleine) Regelkreise überwachen
- komplexe Steuerungsvorgänge durchführen
- stets hohe Präzision und Geschwindigkeit einhalten, trotzdem schnelle Reaktion auf Fehler
- große Datenflut bewältigen und verwalten, insbesondere auch bei Störfällen
- monotone oder gefährliche Arbeiten konstant gut ausführen (z.B. lackieren, schweißen, füllen, prüfen)
- Wirtschaftlichkeit steigern, Termine halten und verkürzen (flexibel, schnell, kleine Loszahlen, billig)

## 1.10 Interdisziplinäres Gebiet

Automatisierungstechnik ist ein Zusammenwirken  
von:

- Halbleitersensorik, Mikroperipherik, Gerätetechnik (neue Materialien)
- rechnergestützter Meßtechnik
- Steuerungs–und Regelungstechnik (globale, robuste, adaptive Algorithmen; Fuzzy–Regelungen)
- VLSI–Entwurfssysteme und Techniken für Spezialchips in kleiner Stückzahl (ASIC–Chips)
- Informatik:
  - ★ Spezifikations– und Entwurfssysteme
  - ★ Sprachen
  - ★ Betriebssysteme
  - ★ Feldbus– und Netzprotokolle
  - ★ Expertensystem-Rahmen
  - ★ Entwurf und Realisierung der Anwendungen
  - ★ Integration in technische und organisatorische Umgebung

## 1.11 Themen der Prozeßinformatik (Echtzeit–Datenverarbeitung)

Fragestellungen aufgrund (detaillierterer)  
Anforderungen, wie:

- ein Echtzeitsystem muß mit dem vom technischen Prozeß vorgegebenem Zeitverhalten Schritt halten
- spezielle Scheduling-Strategien
- Rechtzeitigkeit, Gleichzeitigkeit, ständige Betriebsbereitschaft (determinierter vorhersehbarer Gesamtablauf bei stochastischen Ereignissen)
- besonders hohe Verfügbarkeit und Sicherheit (insbesondere bei militärischen oder medizinischen Anwendungen, bei Personenverkehr, in der Produktion)
  - ★ robust gegen unsachgemäße Handhabung,
  - ★ unempfindlich gegen Einzelfehler,
  - ★ allmählicher Leistungsabfall bei Fehlern (gracefull degradation)

- ★ fail-safe-Übergänge, d.h. keine Gefahr für Geräte und Menschen bei beliebigen Fehlerkombinationen
- ★ Spezifikationsfehler
- ★ Hardware-, Software-, Schnittstellenfehler
- ★ Restrisiko,
- ★ TÜV-Abnahme
- Vielfältige technische Prozesse
- viele Peripherie-Varianten

Daneben nichttechnische Fragen, wie:

- Qualitätsmanagement
  - ★ Qualitätssicherung
  - ★ Qualitätsoptimierung
  - ★ Produkthaftung
  - ★ Senkung Reklamationskosten
  - ★ Senkung Produktionskosten bei frühzeitiger Fehlererkennung
  - ★ Qualitätsnachweise
  - ★ Chargennachweise

- Mensch–Maschine–Kommunikation
  - ★ (humane) Leitwarten
  - ★ selbsterklärende Bedienung (auch bei seltener Benutzung einer Aktion oder Anzeige)
  - ★ Anzeige nur der relevanten Daten, die in der jeweiligen Situation aktuell sind
  - ★ alle Aktionen des Menschen, ggf. nach Warnung, Begründung warum eigentlich nicht zulässig und Bestätigung, erlauben
  - ★ Unterstützung durch intelligente Systeme im Fehlerfall
- soziale und gesellschaftspolitische Fragen (höhere Qualifikation, Wegrationalisieren von Arbeitsplätzen, wirklich humaner?, Fehlen menschlicher Kontakte)

## 1.12 Funktionen zur Prozeßführung

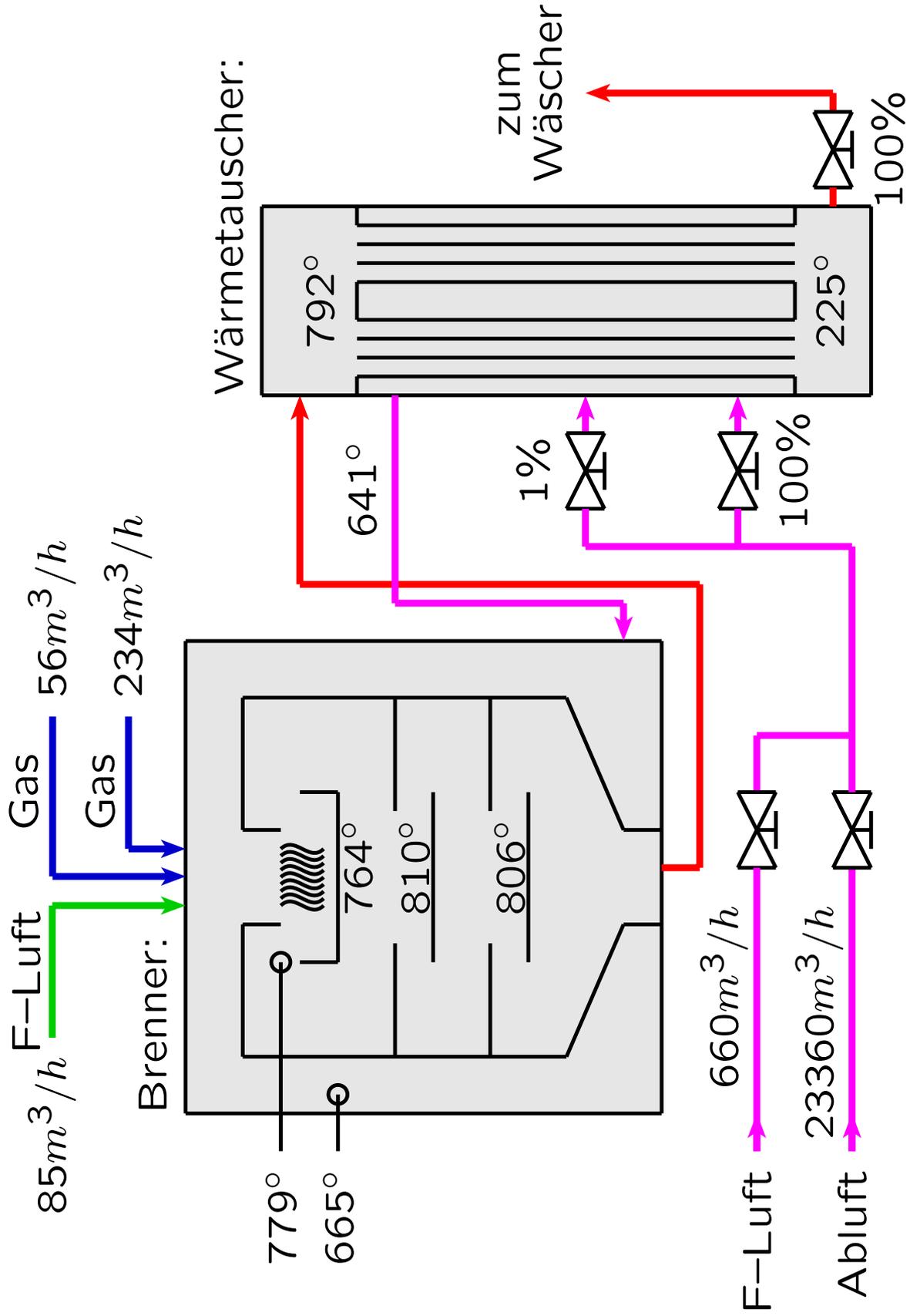
- **Betreiben der Peripherie**

Lesen/Schreiben der Meß- und Stellgrößen  
(digital/analog)

- **Überwachen und Melden**

- ★ Test auf Grenzwertüberschreitung und Änderungsgeschwindigkeit
- ★ Test auf gleichartiges Verhalten abhängiger Größen, beispielsweise gleicher Trend bei Druck und Temperatur
- ★ Test auf defekte Komponenten, beispielsweise durch
  - Plausibilitätskontrolle
  - zeitliche Redundanz (mehrfaches Lesen)
  - Selbst- und Fremdtests
  - redundante Komponenten
  - verschiedene Meßmethoden

- Höhere Meßwertverarbeitung
  - ★ Datenreduktion
  - ★ Mittelwerte
  - ★ Glätten
  - ★ Filtern
  - ★ nur Anzeige momentan relevanter Daten
  - ★ Trendanzeigen)
- Führung lokaler Regelstrecken  
(z.B. PI-,PID-Regler; robuste und adaptive Verfahren; Fuzzy-Regler)
- lokal optimierte Ablaufsteuerung (**Leitsysteme**)
- globale Gesamtoptimierung von Fertigungszellen (**Planungssysteme**)
- **Gesamtkoordination**: Optimierung nach betriebswirtschaftlichen und technischen Gesichtspunkten (cooperate model)
- **Protokollierung, Prozeßvisualisierung**  
(Prozeßwarte, Logbuch, Störungsstatistik)



Beispiel 2 zur Prozessvisualisierung (mit Zuständen)

## 1.13 Prozeßmodelle

- **Typische Prozeßgrößen**
  - ★ Kennwerte oder Anlagenwerte (z.B. Geometrie, Maße, Frequenzen, Grenzwerte)
  - ★ Zustandswerte (Prozeßvariable), setzen Modell voraus
  - ★ Parameter (einstellbare Werte, z.B. Waschprogramm)
  - ★ Eingangsgrößen (unabhängig vom Prozeß, z.B.
    - Zufluß
    - Stellgrößen
    - Störgrößen (von außen, unbeabsichtigt; z.B. Wärme, Rauschen)
  - ★ Ausgangsgrößen (z.B. Abfluß, Meßgrößen, Alarme)

- **Klassifikation der Prozeßmodelle** nach
  - ★ Aufgabenstellung (Produktion, Transport, Qualitätsprüfung)
  - ★ Material (Fertigung, Verarbeitung in der Chemie, in der ...)
  - ★ Energie (Wärme, Strom, Gas)
  - ★ Information (Meß- und Prüftechnik; Fernsehen)
  - ★ zeitlichem und räumlichem Ablauf (kontinuierlich bzw. diskontinuierlich)
  - ★ Modellbildung
    - analytisch (math., phys. oder chem. Gleichungen)
    - empirisch aus Beobachtung oder Erfahrungen
    - Prozeßidentifikation aus Meßwerten
    - gegenständliches Modell in anderem Maßstab

- Genauer betrachtet: Klassifikation nach zeitlichem und räumlichem Ablauf:
  - ★ **Fließprozesse** (dynamische P., kontinuierliche P.)
    - (stückweise) stetige Prozeßgrößen
    - zeit- und/oder ortsabhängige Variable
    - mathematische Modelle durch DGL-Systeme oder Übertragungsfunktionen
  - ★ **Folgeprozesse** (sequentielle P., diskontinuierliche P.)
    - binäre, diskrete Informationsfolgen, die Ereignissen oder bestimmten Zeitpunkten zugeordnet sind
    - mathematische Modelle durch Boole'sche Gleichungen, Automaten, Petri-Netze oder Ablaufpläne
  - ★ **Stückgutprozesse**
    - objektbezogen einzeln identifizierbare Objekte, die Position und/oder Zustand verändern
    - Simulationsmodelle

★ **Chargenprozesse**

Teilmengen (Phasen) eines sonst kontinuierlichen Prozesses (vor allem in der chemischen Industrie)

● Genauer betrachtet: Klassifikation der analytischen Modelle

★ **deterministisch** (exakt), z.B. Fahrpläne beim Schienenverkehr

★ **stochastisch** (Wahrscheinlichkeitsaussagen), z.B. Straßenverkehr

★ **zeitabhängig**

- statische Ein- und Ausgangsgrößen
- stationärer Zusammenhang der Größen
- dynamisch (zeitl. Änderungen der Größen)

★ **adaptiv** (Parameter und Koeffizienten werden laufend aktuell nachjustiert, z.B. zur Elimination von Störungen)

★ **lernend** (Bestimmung von Typ und Grad der Funktionen aus Meßwerten), z.B. neuronale Netze