

Modul ST

Steuerungstechnik



Prof. Dr.-Ing. Ulfert Meiners

Url: www.etch.haw-hamburg.de/~meiners

Email: meiners@etch.haw-hamburg.de

Raum 682

Organisatorisches

- **ST = Vorlesung (3 SWS)**
 - NEU: nun 3SWS im Bachelor! Neues Skript!
 - Begleitmaterial:
 - Kopie der in der Vorlesung gezeigten Folien:
Ersetzt nicht Mitschrift von Ergänzungen und Beispielen!
Ausschließlich für Lehrzwecke.
 - Download von meiner Homepage.

- **SGP = Praktikum (1 SWS):**
 - Laboringenieur Hr. Krebbel (Raum 404, Tel.: 040 42875 8053)
 - 4 Versuche
 - Laboraufgaben inkl. Zusatzmaterial auf meiner Homepage
 - WICHTIG: Vorbereitung der Aufgaben unter Benutzung SPS-Software
SPS-Software zu Lernzwecken bei Herrn Krebbel abholen

- **Zugriff Homepage:**
 - username: StudentST
 - password:



Gliederung

1. Einführung
2. Grundlagen Steuerungstechnik
3. Grundprinzip SPS
4. Programmierung SPS - AWL, FUP, SCL
5. Programmierung SPS - FC, FB, DB
6. Petrinetze - Grundlagen
7. Petrinetze - Erweiterungen
8. Petrinetze - Synchronisation von parallelen Abläufen
9. Petrinetze - Betriebskopf
10. Ablaufkette (SFC)
11. Zusammenhang SIPN <-> Automaten

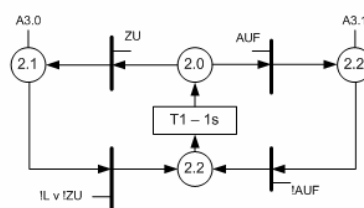
50%

Entwicklungsmethodik:
Steuerungstechnisch
interpretierte
Petrinetze (SIPN)

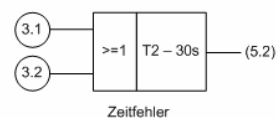
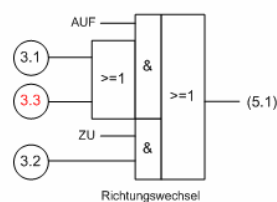
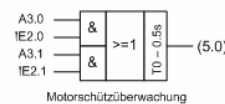
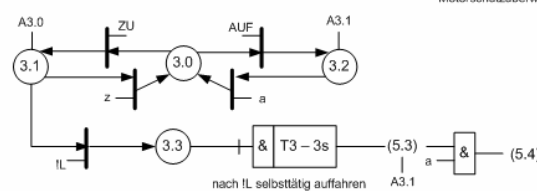


Beispiel für Petrinetz = „Steuerung Werkstor“

Handbetrieb:



Automatikbetrieb:



Literatur

1. **Günter Wellenreuther, Dieter Zastrow: Automatisieren mit SPS – Theorie und Praxis. Vieweg, 3. Auflage, 2005.**
2. Günter Wellenreuther, Dieter Zastrow: Automatisieren mit SPS – Übersichten und Übungsaufgaben. Vieweg, 2. Auflage, 2005.
3. **Hans Berger: Automatisieren mit STEP 7 in AWL und SCL. Siemens, 5. Auflage, 2006.**
4. **Habermann, Weiß: Step 7 Crashkurs. VDE Verlag, 2. Auflage, 1999.**
5. Rainer Pickhardt: Grundlagen und Anwendung der Steuerungstechnik. Petrinetze, SPS, Planung. Vieweg, 2000.
6. **Lothar Litz: Grundlagen der Automatisierungstechnik. Oldenbourg, 2005.**
7. Rainer König, Lothar Quäck: Petrinetze in der Steuerungs- und Digitaltechnik. Oldenbourg, 1988.
8. Jens von Aspern: SPS-Softwareentwicklung mit Petrinetzen. VDE-Verlag, 2003.

Zeitschriften:

- Automatisierungstechnische Praxis (atp)
- Energie & Automation
- Markt&Technik
- SPS Magazin: Systeme – Geräte –Komponenten - Peripherie



Normen

DIN 19226	Teil 1: Regelungs- und Steuerungstechnik Begriffe, Allgemeiner Grundlagen
DIN 19235	Steuerungstechnik: Meldungen von Betriebszuständen
DIN 40700	Logische Verknüpfungspläne
DIN 40713	Stromlaufpläne
DIN 41719	Teil 6: Funktionspläne
DIN 60001	Programmablaufpläne
DIN EN 61131	Speicherprogrammierbare Steuerungen Teil 1: Allgemeine Informationen Teil 2: Betriebsmittelanforderungen und Prüfungen Teil 3: Programmiersprachen Teil 3, Beiblatt1: Leitlinien für die Anwendung und Implementierung von Programmiersprachen für Speicherprogrammierbare Steuerungen Teil 4: Anwenderrichtlinien Teil 5: Kommunikation



1. Einführung

- **Video** [Automatisierung Spritzgussfertigung bei Weidmann](#)



Geräteautomatisierung vs. Anlagenautomatisierung

Geräte	Anlagen
Waschmaschine	Hochregallager
Spülmaschine	Hafenlager
Nähmaschine	Paketverteilung
Fernsehgerät	Flaschenabfülleinrichtungen
Spielzeuge	Schienenverkehr
Telefone	Telefonanlagen
Automobile • Motormanagement • Getriebe- und Fahrwerksteuerung • Fahrerassistenzsysteme • Schutzsysteme	Versorgungsanlagen • Kraftwerke • Energieverteilung • Wasser, Abwasser • Kläranlagen
Maschinen • Werkzeugmaschinen • Industrieroboter	Produktionsanlagen • Montagestraßen • Lackieranlagen

Quelle: Skript Linnemann



Beispiele für Anlagenautomatisierung

Automobilindustrie

Transferstraßen
Handhabungsgeräte
Schweiß- und Farbspritz-
automaten



Werkzeugmaschinenbau

Bohrwerke
Stanzautomaten
Pressensteuerungen
Prüf- und Kontroll-
automaten



Fördertechnik

Kran- und Aufzugs-
steuerungen
Verpackungsmaschinen
Palettierautomaten
Hochregallager
Abfüllanlagen



Nahrungsmittelindustrie

Überwachung von
Rezepturen
Dosiermaschinen



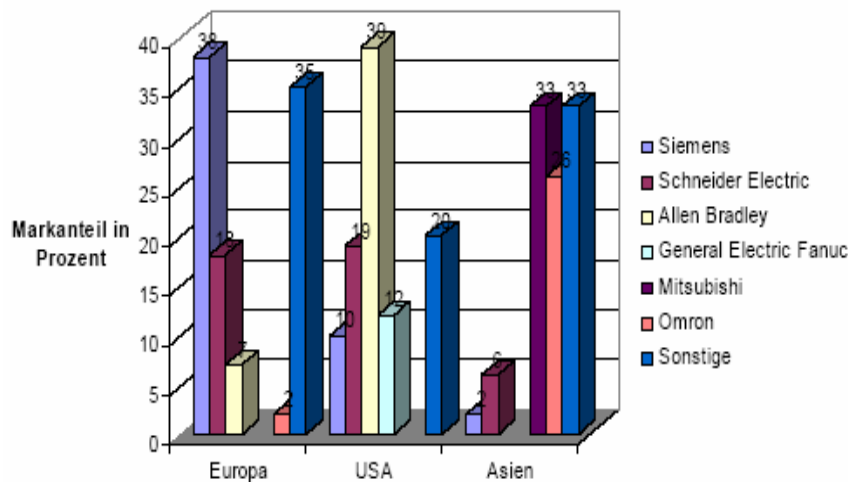
Unterschiedliche Möglichkeiten der Realisierung



Quelle: Skript Linnemann



Anbieter von SPS



Vielzahl von Anbietern = Gesunde Konkurrenz!

Quelle: Skript Linnemann



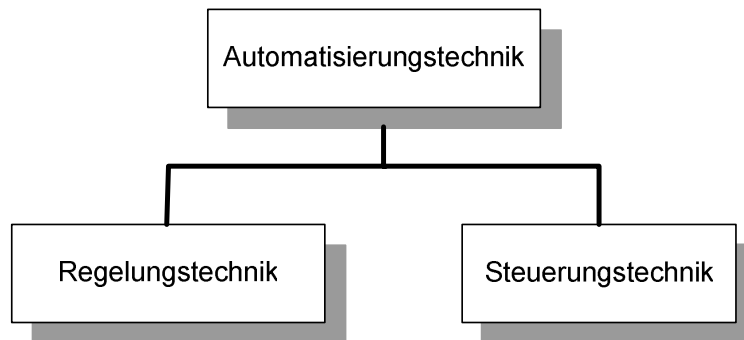
2. Grundlagen der Steuerungstechnik

Was lernen Sie kennen?

- Begriff Automatisierungstechnik
- Zentrale Steuerung / Dezentrale Steuerung / Feldbus
- Klassifikation von Steuerungen:
 - Verbindungsprogrammierte Steuerung
vs. Speicherprogrammierte Steuerung
 - Verknüpfungssteuerung
vs. Ablaufsteuerung
- Anforderungen an Steuerungssysteme



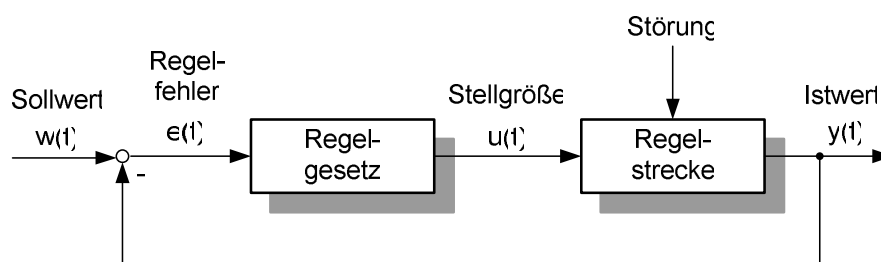
Begriff Automatisierungstechnik



Durch Automatisierung werden *dynamische Prozesse* in ihrem Verlauf erfasst und derart *gezielt beeinflusst*, dass sie *vorgegebene Aufgaben und Funktionen selbsttätig erfüllen*. [Quelle: Litz]



Begriff Regelungstechnik



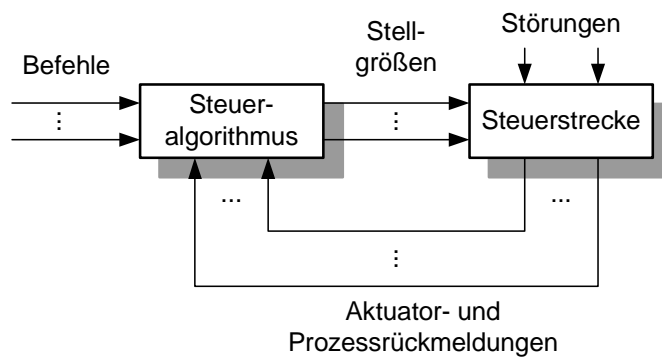
Charakteristika:

- Quasi-analoge Größen: $w(t)$, $e(t)$, $u(t)$, $y(t)$
- Fortlaufender Soll-Istwert Vergleich = Bildung Regelabweichung
- Regelaufgabe (ideal): Zustand $y(t)=w(t)$ herstellen
- oftmals Eingrößenregelung = nur ein Istwert wird geregelt

Beispiel: Temperaturregelung



Begriff Steuerungstechnik



unser Thema!

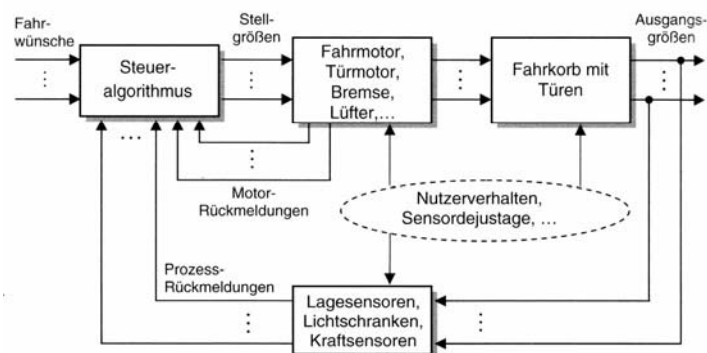
Charakteristika:

- Kein Soll-/Ist Vergleich!
- Befehle = binäre Größen (z.B. Start/Stop-Befehl)
- Rückmeldungen = binäre Größen (z.B. Unterbrechung Lichtschranke, Überschreitung Temperatur, Erreichen einer Endlage, Ausfall Motor)
- Vielzahl von Befehlen und Rückmeldungen

Beispiel: Werktorsteuerung, Fahrstuhlsteuerung



Beispiel Fahrstuhlsteuerung



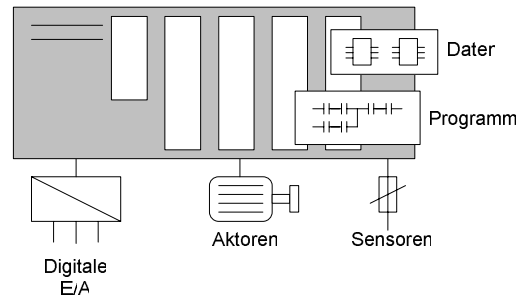
Allgemein: Steuerung reagiert auf Eintreffen eines Ereignisses mit einer gewünschten, logischen Antwort. In vielen Fällen ist diese logische Funktion nicht nur von den aktuellen Eingängen, sondern auch von der „Vorgeschichte“ abhängig („Vorgeschichte“ in „**Merkern**“ gespeichert).

Quelle: Litz



Zentrale Steuerung – seit 1970

Zentrale Steuerung



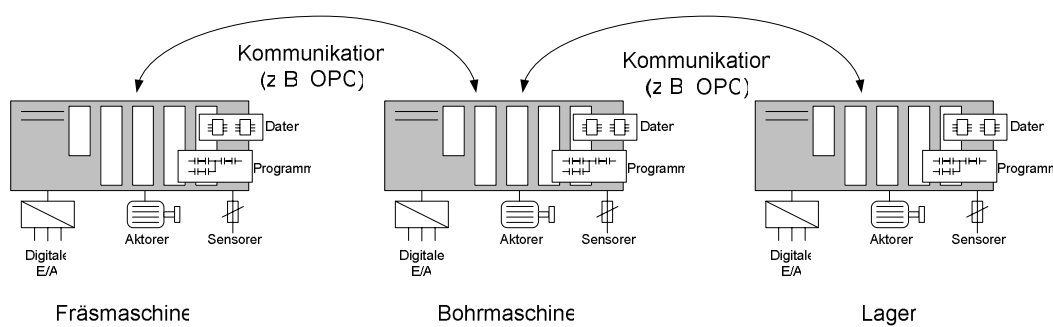
Zentrale Steuerung übernimmt komplette Automatisierung:

- Kompaktsteuerung: 16/32 E/A
- Mittlere Steuerung: bis 256 E/A
- Große Steuerung: bis 4096 E/A

- ☺ Gut Strukturierbare Software
- ☺ Geringe Kommunikationsanforderungen
- ☹ Verdrahtung aufwändig, teuer, unübersichtlich
- ☹ Modularisierung schwierig
- ☹ Kaum Wiederverwendbarkeit von Software



Dezentrale Steuerung – seit 1980

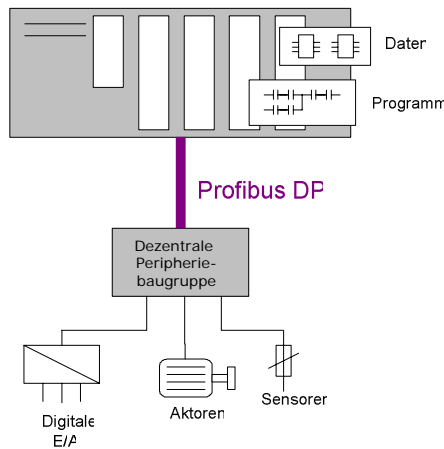


Automatisierungsaufgabe wird heruntergebrochen auf **mehrere, dezentral (=lokal) angeordnete Steuerungen**. Diese müssen zur Lösung der Gesamtaufgabe Daten austauschen = kommunizieren.

- ☺ Verdrahtung kostengünstiger als bei zentraler Steuerung
- ☺ Möglichkeit der Modularisierung von Anlagen
- ☺ Ansätze zur Wiederverwendbarkeit von Software
- ☹ Aufwändige, kostenintensive Kommunikation
- ☹ Aufwändige Synchronisation und Datenabgleich
- ☹ Inbetriebnahme aufwändiger



Steuerung mit Feldbus – seit 1990



E/A, Aktorik und Sensorik wird jetzt über **dezentrale Peripheriebaugruppen** angeschlossen, die über einen Feldbus (meist 2-adrig) mit den Steuerungen verbunden sind

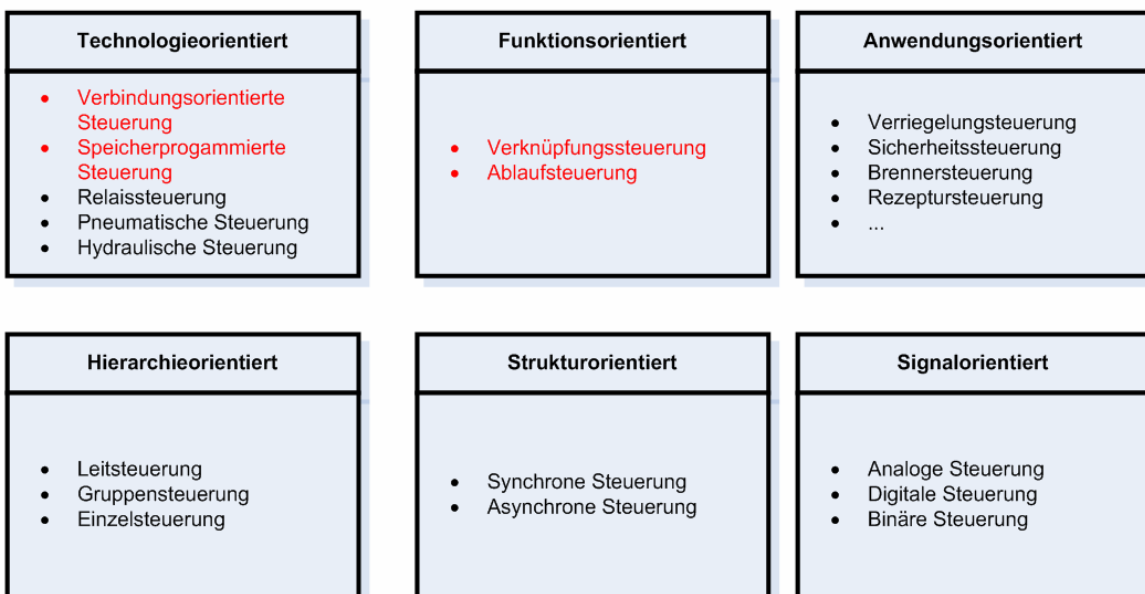
Beispiele für Feldbus: Interbus, **Profibus DP**, **Profinet**

😊 **Verdrahtung** bei großer Anzahl von Sensoren und Aktoren nochmals **kostengünstiger**. Einfacher Schaltschrankaufbau.

😊 **Entlastung der Steuerung** durch intelligente, dezentrale Peripheriebaugruppen (z.B. Zeitstempelung der Eingangssignale)



Klassifikation von Steuerungen



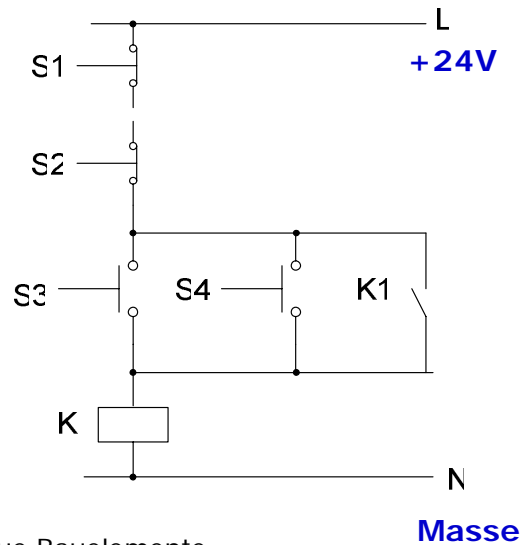
Quelle: Linnemann



Verbindungsorientierte Steuerung

Beispiel: Selbsthalteschaltung mit zwei Ein- und Aus-Tastern

- Realisierung mit Schützen oder Relais
- Wirkungsweise der Steuerung durch Verdrahtung festgelegt
- Einsatz heute nur noch für sicherheitsrelevante Funktionen (falls z.B. Systemabsturz SPS)

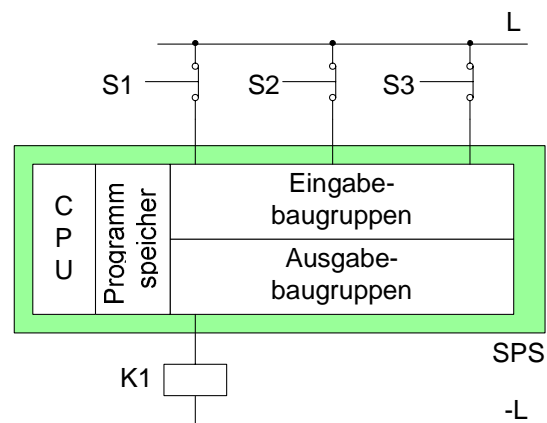


- ⊗ Änderung Steuerungsablauf
 - > Änderung Verdrahtung / ggf. neue Bauelemente
 - > zeitaufwendig und unflexibel und damit teuer!



Speicherprogrammierbare Steuerung (SPS)

- Schaltelemente werden an die Eingänge, die Betätigungsspulen an die Ausgänge angeschlossen
- Wirkungsweise der Steuerung durch „**Programm = Folge von Anweisungen**“ festgelegt
- Programm wird durch **Programmiergerät (PG)** in Programmspeicher geschrieben. Zyklische Ausführung!
- **SPS = Rückrat der Automatisierungstechnik!**



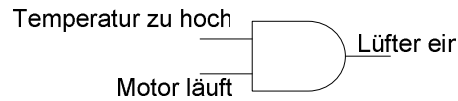
- ☺ Änderung Steuerungsablauf -> Änderung Programm (einfach, kostengünstig)



Verknüpfungssteuerung

Beispiel:

WENN Temperatur zu hoch
UND Motor läuft
DANN Schalte Lüfter ein

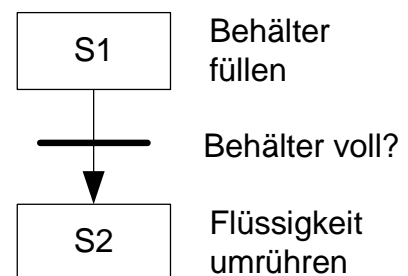


- Ordnet den Zuständen der Eingangssignalen bestimmte Zustände der Ausgangssignale im Sinne **Boolescher Verknüpfungen** zu
- Verknüpfungen müssen in Echtzeit ausgewertet und ausgeführt werden
- Für Aufgabenstellungen, die **unabhängig vom Zeitablauf** sind
- Beschreibung von Verknüpfungssteuerungen durch:
 - **Kontaktplan (KOP)** ein um 90° gedrehter Stromlaufplan
 - **Funktionsplan (FUP)** Netzwerk von Logik- und Zeitbausteinen
 - **Anweisungsliste (AWL)** Hardwarenahe Programmiersprache („ASM“)
 - **Strukturierter Text (ST)** Höhere Programmiersprache („PASCAL“)



Ablaufsteuerung

- Steuerung mit einem zwangsläufig **schrittweisen Ablauf**. Weiterschalten von einem Schritt auf den programmgemäß folgenden abhängig von **Weiterschaltbedingungen (Transitionen)**
- Viele Probleme der Fertigungs- und Verfahrenstechnik haben mit zeitlichen Abläufen zu tun und lassen sich als Folge von Schritten beschreiben.
- Gut für zeitliche, parallele oder zu synchronisierende Abläufe



Programmierung:

- Mit Mitteln der Verknüpfungssteuerung möglich (z.B. AWL)
- Spezielle Programmiersprachen:
 - Sequential Flow Chart (SFC)
 - Ablaufsprache (AS)

**Unser Thema:
Petrietze (SIPN)!**



Typische Anforderungen:

- ❑ Robust, zuverlässig, hoch verfügbar
- ❑ Software / Komponenten wiederverwendbar
- ❑ „Plug&Play“
- ❑ Definiertes Verhalten im Fehlerfall
- ❑ Hohe Effizienz
- ❑ Modulare SW/HW, je nach Aufgabenstellung
- ❑ Echtzeitverhalten



3. Grundprinzip einer SPS

- Kennzeichen SPS
- Aufbau SPS
- Bauformen SPS
- Zyklische Programmabarbeitung
- Organisationsbausteine OBs
- Zykluszeit

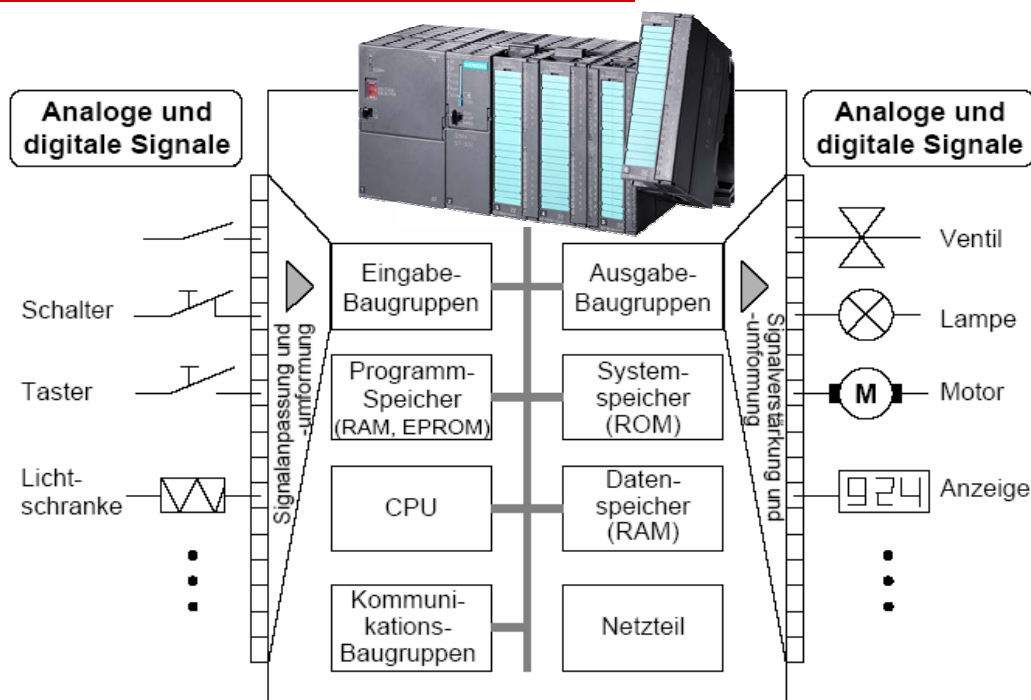


Kennzeichen einer SPS

- ❑ Einsatzgebiet:
Messen, Steuern, Regeln (MSR)
- ❑ **Robust**, hohe **Zuverlässigkeit**,
hohe **Verfügbarkeit**
- ❑ **Modular** und **kompakt**
- ❑ Einfache Erweiterbarkeit mit E/A-Baugruppen für analoge oder digitale Größen, Zähler, Regler, Positionierbaugruppen
- ❑ **Definiertes Verhalten** bei Spannungsausfall und Wiederanlauf
- ❑ **Kommunikation** mit anderen Rechnern über spezielle Bussysteme (Profibus, Profinet, Industrial Ethernet)
- ❑ **Echtzeitverhalten** durch zyklische Programmabarbeitung
- ❑ Anwendungsorientierte **Programmiersprachen** (AWL, KOP, FUP, ST, AS)



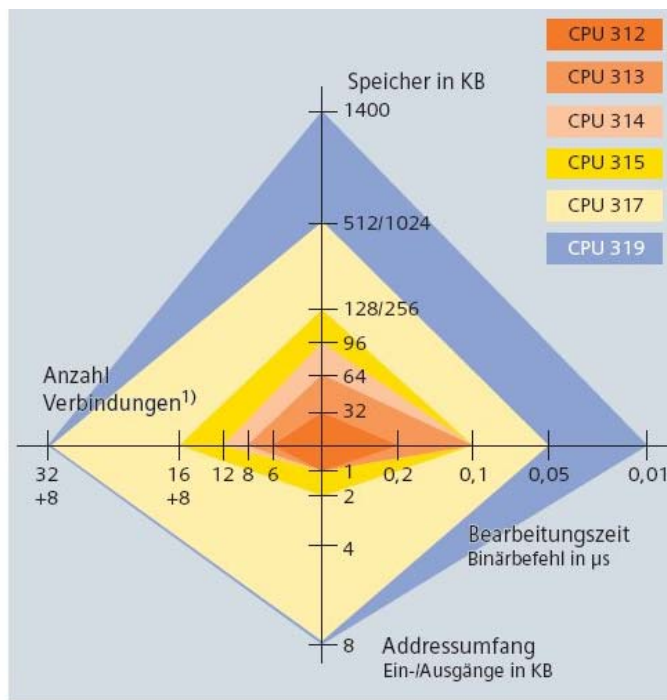
Aufbau SPS (Prinzip)



Quelle: Skript Linnemann



Leistungsüberblick Siemens S7-300 CPUs



Quelle: Siemens



Technische Daten einiger Siemens-SPS

CPU	CPU 312	CPU 314 ¹⁾	CPU 315-2 DP ¹⁾	CPU 315-2 PN/DP ¹⁾	CPU 317-2 DP	CPU 317-2 PN/DP ¹⁾	CPU 319-3 PN/DP
Abmessungen (mm)	40 x 125 x 130		40 x 125 x 130	80 x 125 x 130	80 x 125 x 130		120 x 125 x 130
Bestellnummer-Rumpf: 6ES7	312-1AE.	314-1AG.	315-2AG.	315-2EH.	317-2AJ.	317-2EK.	318-3EL.
Speicher							
Arbeitsspeicher	32 KB	96 KB	128 KB	256 KB	512 KB	1 MB	1,4 MB
Anweisungen	10 K	32 K	42 K	84 K	170 K	340 K	470 K
Bearbeitungszeiten							
Bit-Operation	0,2 µs	0,1 µs	0,1 µs		0,05 µs		0,01 µs
Wort-Operation	0,4 µs	0,2 µs	0,2 µs		0,2 µs		0,02 µs
Festpunkt-Operation	5 µs	2 µs	2 µs		0,2 µs		0,02 µs
Gleitpunkt-Operation	6 µs	3 µs	3 µs		1 µs		0,04 µs
Merker/Zeiten/Zähler							
Merker	128 Byte	256 Byte	2048 Byte		4096 Byte		8192 Byte
S7-Zeiten/Zähler	128/128	256/256	256/256		512/512		2048/2048
IEC-Zeiten/Zähler	•	•	•		•		•
Adressbereiche							
Peripherie E/A (Byte)	1024/1024	1024/1024	2048/2048		8192/8192	8192/8192	8192/8192
Prozessabbild E/A (Byte)	128/128	128/128	128/128		256/256	2048/2048	2048/2048
Digitale Kanäle (zentral)	256	1024	1024		1024	1024	1024
Analoge Kanäle (zentral)	64	256	256		256	256	256
DP-Schnittstellen							
DP-Mastersysteme intern / CP 342-5	• / •		• / •		• / •	• / •	• / •
DP-Slaves			•		•	•	•
PROFINET-Schnittstelle							
PROFINET CBA				•		•	•
PROFINET IO				•		•	•
PROFINET mit IRT							• ³⁾
TCP/IP				•		•	•
UDP				•		•	•
ISO-on-TCP (RFC 1006)				•		•	•
Webserver				•		•	•

Quelle: Siemens



Bauformen SPS

„Klassische“ SPS

Vorteile:

- Robust, hoch Zuverlässig

Nachteile:


- Bedienfunktionen extern
- Aufwändige Kommunikation



Steuerungsintegriert

Gaststeuerungen:

- CNC (Computerized Numerical Control)
- RC (Robot-Control)



Soft-SPS mit PC

Vorteile:

- Flexibilität des PC
- Bedienfunktionen gut realisierbar
- Vielseitige Kommunikation

Nachteile:

- Flexibilität mindert Zuverlässigkeit
- E/A nur über Netzwerk anschließbar




Slot-SPS im PC

Vorteile:

- Robust und Zuverlässig durch gesonderte SPS-Karte
- PC-Teil für Bedienfunktionen und Kommunikation

Nachteile:

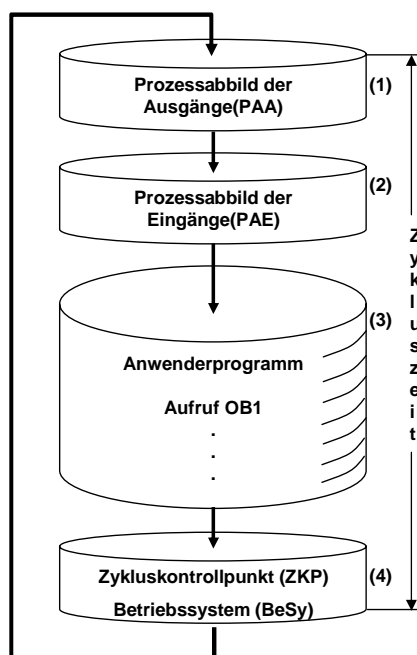
- E/A nur über Netzwerk anschließbar
- Kosten für SPS-Karte



Quelle: Linnemann



Zyklische Programmabarbeitung



Das Betriebssystem startet die Zykluszeitüberwachung

- (1) Die CPU schreibt die Werte aus dem **Prozessabbild der Ausgänge** in die Ausgabebaugruppen.
- (2) Die CPU liest den Zustand der Eingänge an den Eingabebaugruppen und aktualisiert das **Prozessabbild der Eingänge**
- (3) Die CPU bearbeitet das **Anwenderprogramm** in Zeitscheiben und führt die im Programm angegebenen Operationen aus
- (4) Am Ende eines Zyklus führt das Betriebssystem anstehende Aufgaben aus, z. B. Laden und Löschen von Bausteinen

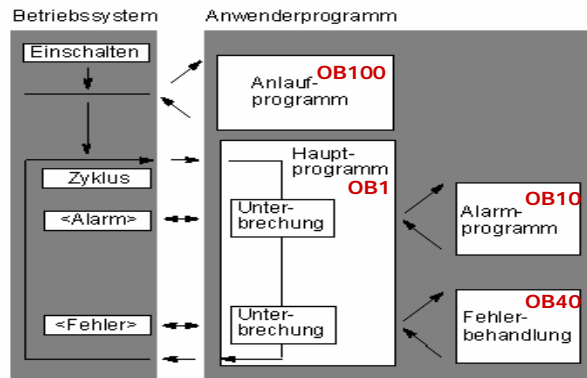
Anschließend kehrt die CPU zum Zyklusanfang zurück und startet erneut die **Zykluszeitüberwachung**

Quelle: Siemens



Organisationsbausteine und Zykluszeit

Organisationsbausteine (OBs) sind Schnittstelle zwischen Betriebssystem und Anwendungsprogramm:



Anlauf: **OB100 (Kaltstart)**,
OB101 (Wiederanlauf)

Zyklus: **OB1**

Periodische Unterbrechung:
Uhrzeit **OB10..17**
Weckalarm **OB30..38**

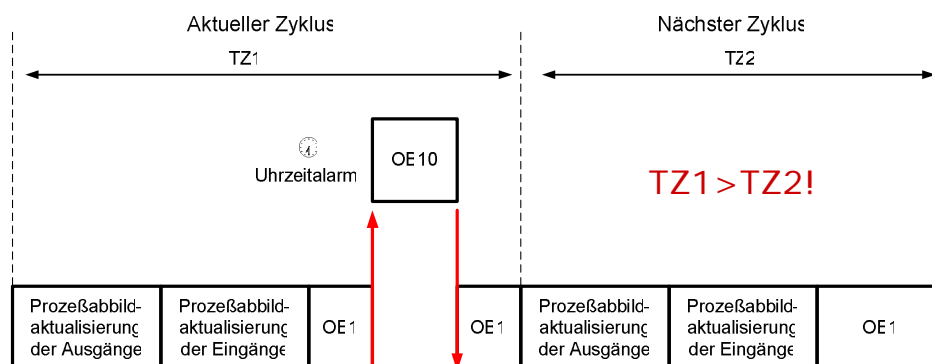
Ereignisgesteuerte Unterbrechung:
Verzögerung **OB20..23**
Prozessalarm **OB40..47**
Fehler OBs

Einflussnahme auf Programmablauf; ähnlich ISR!



Zykluszeit

- Die **Zykluszeit (TZ)** ist die Zeit, die das Betriebssystem für die Bearbeitung des zyklischen Programms sowie aller diesen Zyklus unterbrechenden Programmteile (z. B. Bearbeitung anderer Organisationsbausteine) und Systemtätigkeiten (z. B. Prozessabbildaktualisierung) benötigt. Diese Zeit wird überwacht.
- Die Zykluszeit ist **nicht für jeden Zyklus gleich lang**



Quelle: Siemens

