Automatisierungstechnik

Prof. Dr. Mark Ross

ross@hs-koblenz.de

WS 2019/20
Stand: 28. September 2019

Normale Slides mit Hyperlinks

Modalitäten

Modul:

Modul	Pflicht	TWPF	V+P+SW	ECTS	Aufwand
E030	BMT, BET(alt)	BWI	3+1+0	5	60h + 90h
E446	BET (neu)		3+1+4	10	120h + 180h

Material: www.hs-koblenz.de/ross

Vorkenntnisse:

	ET	IT	MT	WI	BBS	
AUT	PF	WPF	PF	WPF	PF	
DIGT	✓	✓	✓		✓	Aussagenlogik, Endliche Automaten
GDI1	✓	✓		✓	✓	OSI-Modell, Netzwerktechnik

Vorlesung

Kontakt: ross@hs-koblenz.de

Termine: Di 8:15 Vorlesung
Do 12:15 Vorlesung oder Praktikum (C013)
weitere Praktikumstermine nach Absprache

Klausur: 90 min, keine Hilfsmittel

Praktikum

Kontakt: halfmann@hs-koblenz.de (Anmeldung, Gruppeneinteilung und Termine in OLAT)

Scheinerwerb: Anwesenheit, mündl. Testate

Versuche: TIA-Einführung, Timer & Zähler, Analogwerte & SCL, Visualisierung & Simulation, evtl. Schrittketten in SCL, Debugging, OPC, Motion Control

Softwareprojekt/Ausarbeitung mit Vortrag

Termine: wird in Vorlesung bekannt gegeben

Literatur

[Hei15] T. Heimbold. Einführung in die Automationisierungstechnik
Hanser, 2015. ISBN: 978-3-446-42675-7.

[KHT00] W. Kriesel, T. Heimhold und D. Telschow. Bustechnologien für die Automation.
Hüthig, 2000. ISBN: 3-7785-2778-9.

[Lit13] Lothar Litz. Grundlagen der Automatisierungstechnik - Regelungssysteme, Steuerungssysteme, hybride Systeme.
Oldenbourg, 2013, S. I–XIII, 1–540. ISBN: 978-3-486-70888-2.

[Sei15] Matthias Seitz. Speicherprogrammierbare Steuerungen für die Fabrik- und Prozessautomation.
Hanser, 2015. ISBN: 978-3-446-43325-0.

[WZ09] G. Wellenreuther und D. Zastrow. Automatisieren mit SPS - Theorie und Praxis.
Viewegs Fachbücher der Technik. Vieweg, 2009. ISBN: 9783834802316.

Inhalt

1. Einführung

Beispiele und Definition

Beispiele einer Automatisierung

. . . . . .

. . . . . .

. . . . . .

Definition

Durch *Automatisierung* werden dynamische Prozesse in ihrem Verlauf . . . . . und derart gezielt . . . . . , dass sie vorgegebene Aufgaben und Funktionen . . . . . erfüllen. [Lit13]

Teildisziplinen der Automatisierungstechnik

- Steuerungstechnik
- Regelungstechnik
- Robotik
- Antriebstechnik
- Mess- und Sensortechnik
- Bustechnologie
- Informationsverarbeitung
- Visualisierung, Überwachung, Diagnose

Begriffe: Signalarten

Definition I:

analog: Signal kann beliebige Werte in bestimmten Grenzen annehmen (stetig)

diskret: Signal kann nur diskrete Werte annehmen (nicht stetig)

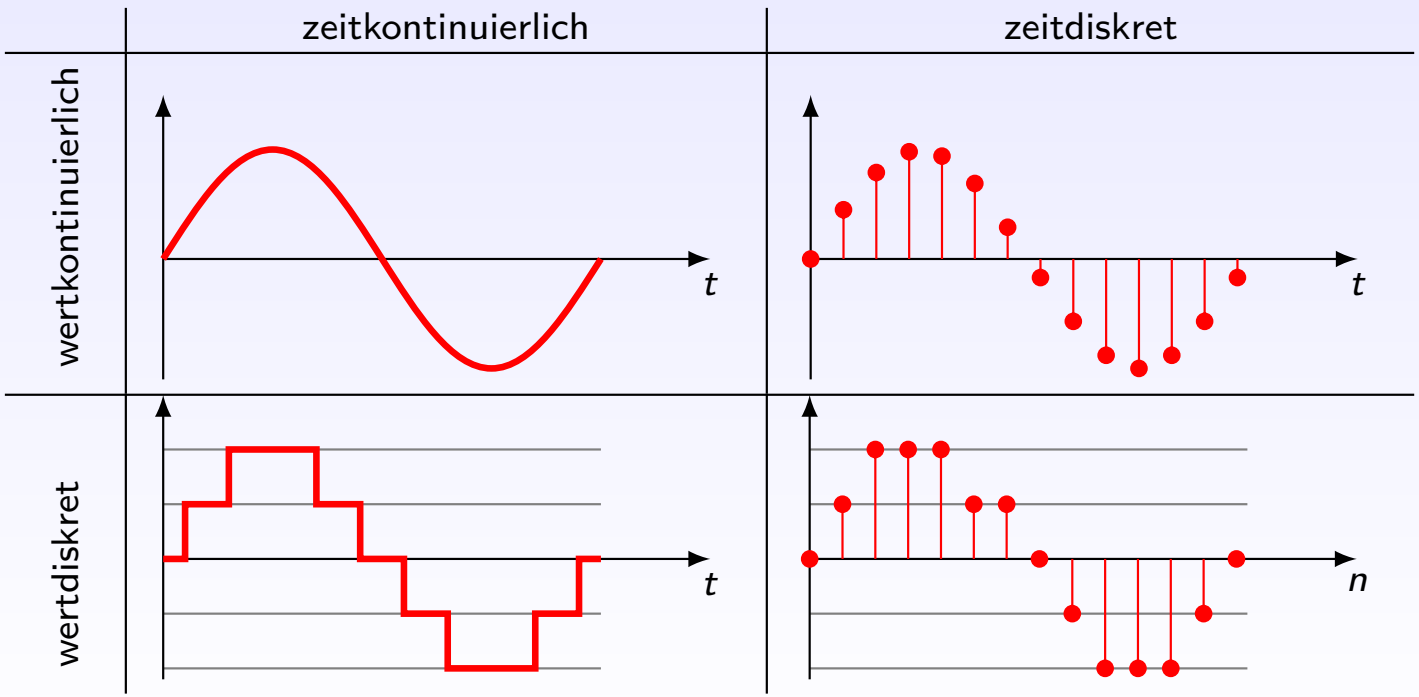
kontinuierlich: Signal kann sich zu beliebigen Zeitpunkten ändern

diskontinuierlich: Signal kann sich nur zu bestimmten Zeitpunkten ändern, z.B. mit festem Takt

Definition II:

analog: zeit- UND wertkontinuierlich

diskret: zeit- UND wertdiskret



1. Einführung

Ziele einer Automatisierung

Betrieb einer Anlage (z.B. Fabrik) oder eines Produktes (z.B. Auto) wird möglichst:

- ökonomisch: .....
- gleichmäßig: .....
- zuverlässig: .....
- sicher: .....
- ökologisch: .....
- komfortabel: .....
- flexibel: .....

$\Rightarrow$ .....

Funktionen einer Automatisierung

Zur Erreichung der Ziele kommen spezielle Funktionen zum Einsatz:

Regeln: Kontinuierliche, physikalische Größen auf gewünschte .....

und ..... ausregeln

Steuern: Diskrete Zustände oder Zustandsfolgen eines technischen Prozesses

.....

Überwachen: Physikalische Größen oder Prozesszustände kontinuierlich auf Abweichungen von Sollgrößen und -zuständen überwachen

Melden: automatische Meldung bei Abweichungen vom Sollverhalten

Bedienen: Beeinflussung durch Operator

Anzeigen: Visualisierung des Zustandes

Zustände/Folgen	Beispiel Aufzug
ermöglichen:	.....
verhindern:	.....
erzwingen:	.....

Regelung vs. Steuerung

	Regelung	**Steuerung**
Beispiel	Temperaturregelung	Aufzugsteuerung
Aufgaben	Physikalische Größen auf Sollwerte bringen und Störungen ausregeln	Diskrete Zustände ermöglichen, verhindern oder Zustandsfolgen erzwingen
Signalart (Sensor&Aktor)	Kontinuierlicher Wertebereich, z.B. Temperatur, Ventilstellung in %	Diskreter/binärer Wertebereich, z.B. Tür auf/zu, Motor an/aus
Signalanzahl		
Theorie, Methodik	Regelungstheorie in Zeit- und Bildbereich, Stabilitätstheorie, Simulation	Automatentheorie, Petri-Netze, Tests, Temporale Logik, Model Checking

früher/manchmal

Regelung: closed loop control, Rückkopplung

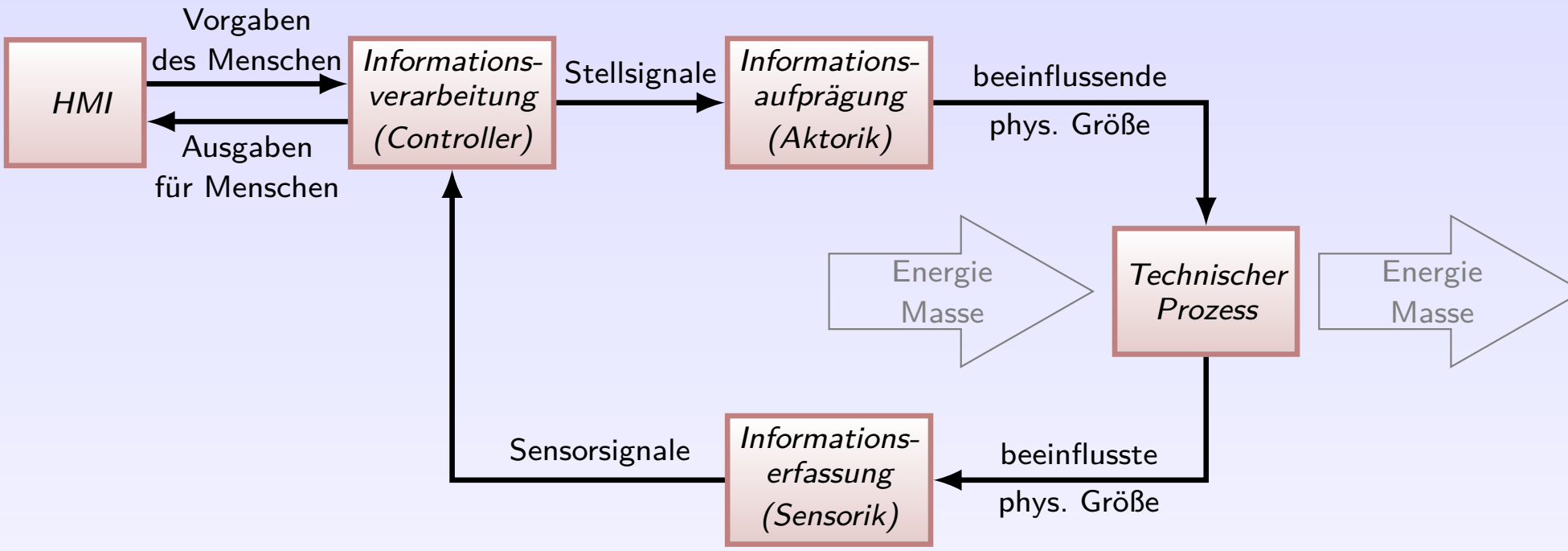
Steuerung: feed forward control, keine Rückkopplung

heute/hier

Regelung: Sollwert, wenige analoge IO-Größen, immer Rückkopplung, Standardverfahren (z.B. PID-Regler), Parameter werden optimiert

Steuerung: viele diskrete IO-Größen, meist Rückkopplungen, individuelle Algorithmen

Struktur mit Rückkopplung



- .....
- .....
- .....

1. Einführung

Industrie 4.0 - Zukunft oder Gegenwart?

- Zukunftsprojekt der ..... ,
  um Informatisierung der klassischen Industrien (z.B. Produktionstechnik) voranzutreiben
- Ziel: Erhalt des Wirtschafts- und Technologiestandorts Deutschland
- Entspricht „Advanced Manufacturing“ in USA
- Vgl. Vorteil in Asien: billige Arbeitskräfte (Handarbeit)
- Vision: intelligente Fabrik (Smart Factory), ausgezeichnet durch
  - Wandlungsfähigkeit (starke Indiviualisierung der Produkte),
  - Ressourceneffizienz (Just-In-Time-Lager) und Ergonomie (keine monotonen Tätigkeiten)
  - Integration von Kunden und Partnern
- Extremfall: menschenleere, sich selbst optimierende Fabrik
- Realität: Fabrik ohne schlecht/mittelmäßig qualifizierte Fachkräfte, sondern einige spezielle Techniker, Ingenieure, Informatiker

Industrie 4.0 - Vier Industrielle (R)evolutionen

1. .....
2. .....
3. .....
4. .....

$\Rightarrow$ .....

Industrie 4.0 - Fazit

- Erste Ansätze seit Beginn des 21. Jahrhunderts begonnen
- Höhepunkt voraussichtlich erst in den nächsten zwanzig Jahren
- Vierte Revolution steht im Zeichen der .....

  .....
- Wichtigstes Ziel ist nicht ....., sondern

  Produktion und Logistik ..... und ..... zu gestalten

Movies:

- Industrie 4.0
- Audi: Smart Factory
- BMBF: Fabrik von Morgen

Soziale Aspekte der Automatisierungstechnik

- (Massen-)Produktion in fast menschenleeren Fabriken
- Reduzierung monotoner oder gefährlicher Tätigkeiten

– Wegfall von Arbeitsplätzen .....

\+ Zunahme von Arbeitsplätzen .....
  - Entwicklung und Design von Produkten,
  - Überwachung der Produktion,
  - Vertrieb, Marketing, Service
  - Konstruktion und Instandhaltung von Produktionsanlagen

2. SPS

SPS - Speicherprogrammierbare Steuerung

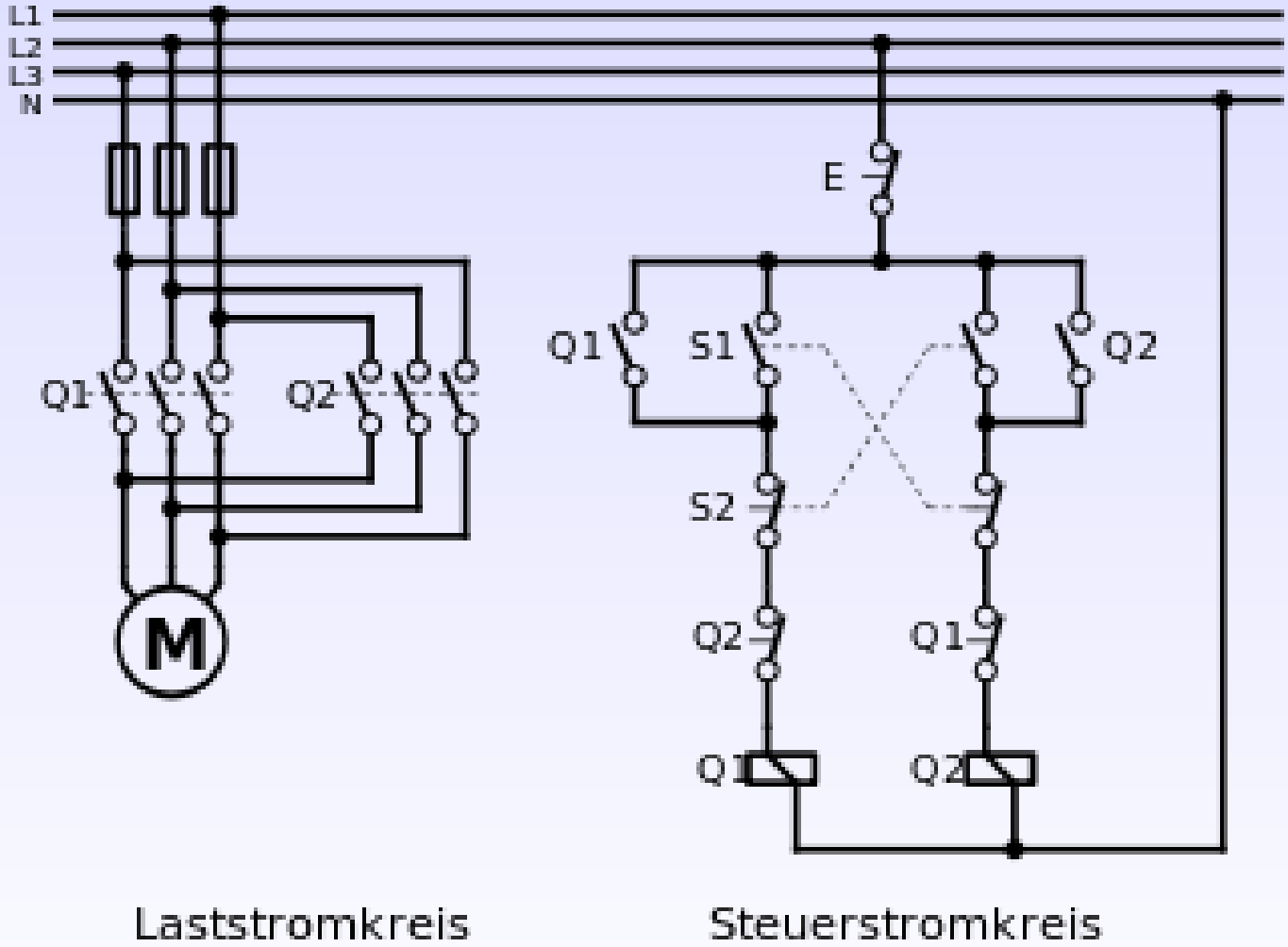
- engl. .....
- Anfangs nur binäre Steuerungen
- Später auch Regelungen (Analoggrößen)
- Verschiedene Modelle z.B. auf www.tpautomation.de
  kompakt/modular, mit/ohne Display, klein/groß, schnell/langsam, preiswert/teuer

SPS - Funktionsumfang

- Optimiert für logische Funktionen (z.B. bitweise Verknüpfungen)
- Verarbeitung von Analoggrößen
- Komplexe mathematische Operationen (sin, cos, PID-Regler, . . .)
- Neben Kernaufgaben (Regelung, Steuerung) auch zusätzliche Funktionen:
  - Visualisierung, Bedienung, Alarmierung (Mensch-Maschine-Interface)
  - Protokollierung von Ereignissen (Data-Logging)
  - Webserver, Datenbank-Server, OPC-Server
- Feldbusschnittstellen verringern Verdrahtungsaufwand und Fehleranfälligkeit
- Module für Motion Control: Drehzahlregelung, Synchrone Bewegung, Sanftanlauf, . . .
- Anbindung an Verwaltungsrechner (vertikale Integration)
  $\rightarrow$ aktuelle Daten über Fertigungsstände, Lagerbestände, Anlagenzustand und -auslastung

VPS - Verbindungsprogrammierte Steuerungen

Beispiel:



Festverdrahtete Logik: NOT=Öffner, AND=Serie, OR=Parallel

SPS - Vorteile (im Vergleich zu VPS)

+ Flexibilität: Einfacher Austausch von Programmen
+ Geringer Platzbedarf, Höhere Zuverlässigkeit, Geringere Kosten
+ Vernetzung mit anderen Systemen, Fernwartung
+ Einfachere Fehlerdiagnose

Ausführungsformen

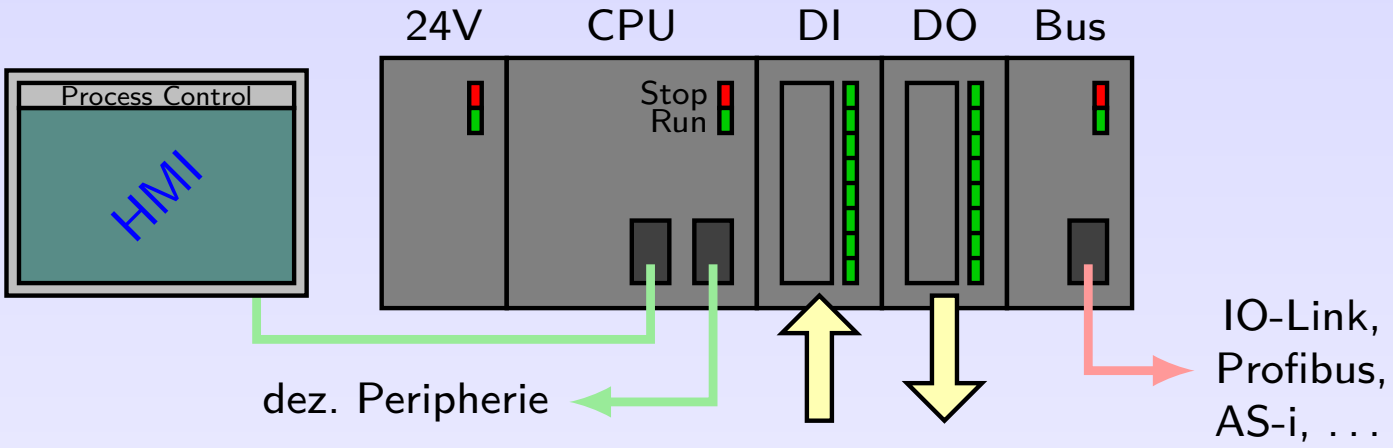
(Hardware-)SPS: Kompaktgerät oder Komponenten als Einsteckkarte in Baugruppenträger mit Rückwandbus

Slot-SPS: .....mit Echtzeitbetriebssystem zum Einbau in einen Host-Industrie-PC

Soft-SPS: softwaremäßige Nachbildung der SPS-Funktionalität auf einem Industrie-PC

Fehlersichere SPS: sicherheitsrelevante Systeme erfordern Festverdrahtung oder spezielle, fehlersichere SPS

SPS - Komponenten



Baugruppenträger: Aufnahme der steckbaren Baugruppen, Rückwandbus

Netzteil: meist 24 VDC, separat oder integriert

CPU: arbeitet Programm zyklisch und sequentiell ab
Prozessor und Firmware optimiert für Echtzeitverarbeitung von Logik/Arithmetik

DI-Baugruppe: Oft 24V-Pegel (-35...4,5V für Low, 13...35V für High), galvanische Trennung über Optokoppler, Entstörung mit Tiefpässen, gemeinsames Massepotential

DO-Baugruppe: Transistoren (24VDC/500mA), Triac (230VAC/1A), Relais (230V/2A)

AI-Baugruppe ADU wandeln analoge Signale (4/20mA, 0/10V, Pt100) in 8-15 Bit Eingangswort

AO-Baugruppe: DAU wandeln digitale Ausgangsworte in analoge Signale

Sonder-Baugruppen .....

.....

Hardware–SPS

Kompakter Aufbau

+ kleiner, kompakter Aufbau
+ preiswerter (als modularer Aufbau)

Modularer Aufbau

+ .....
+ .....

Beispiel: Simatic S7-1200

- Kompakter Aufbau, modular erweiterbar
- Für einfache Automatisierungsaufgaben
- I/O: 14DI, 10DO, 2AI
- Unterschiedliche CPU-Leistungsklassen: 1211C, 1212C, 1214C, 1215C und 1217C
- Erweiterungen:
  - Signal Board für digitale/analoge I/O (Einbau)
  - Je nach CPU bis zu acht Signal Module für digitale/analoge I/O (Anbau rechts)
  - Bis zu drei Kommunikationsmodule für verschiedene Bussysteme (Anbau links)

Soft–SPS

- SPS-Software auf Industrie-PC mit Feldbus-Schnittstelle
- Prozessor führt SPS-Programm in Echtzeit aus
  - abhängig von PC-Hardware und Betriebssystems
  - geteilte Prozessorleistung für Betriebssystem und Anwendersoftware

Beispiel: Simatic WinAC - Windows Automation Center

- Software-SPS für PC-basierte Automatisierung mit Echtzeitverhalten
- Integration von Datenverarbeitung, Kommunikation, Visualisierung und Technologie auf IPC
- gesamter PC-Arbeitsspeicher als Programmspeicher nutzbar
- Bitoperationszeiten von 0,004 µs möglich
- Fehlersichere Variante (zertifiziert vom TÜV Süd)

PC- vs. SPS-Programmbearbeitung

	PC (konventionell)	**SPS**
Programm	einmaliger Durchlauf	wird permanent zyklisch durchlaufen
Eingänge	jederzeit, Events	synchrones Einlesen, PAE
Ausgänge	jederzeit	synchrone Ausgabe, PAA
Reaktionszeit	undefiniert	Echtzeit = garantierte Reaktionszeit
Sicherheit	undefiniert	Überwachung und Abschalten im Fehlerfall für sicheren Zustand

PAE - Prozessabbild der Eingänge

Vor jedem Zyklus wird Zustand der Eingänge in Speicher kopiert. Programm greift nie direkt auf Eingänge zu, sondern immer auf diese Kopie.

$\Rightarrow$ .....

PAA - Prozessabbild der Ausgänge

Während Programmbearbeitung werden Ausgänge nie direkt gesetzt, sondern in Speicher geschrieben, der nach dem Zyklus auf physikalische Ausgänge übertragen.

$\Rightarrow$ .....

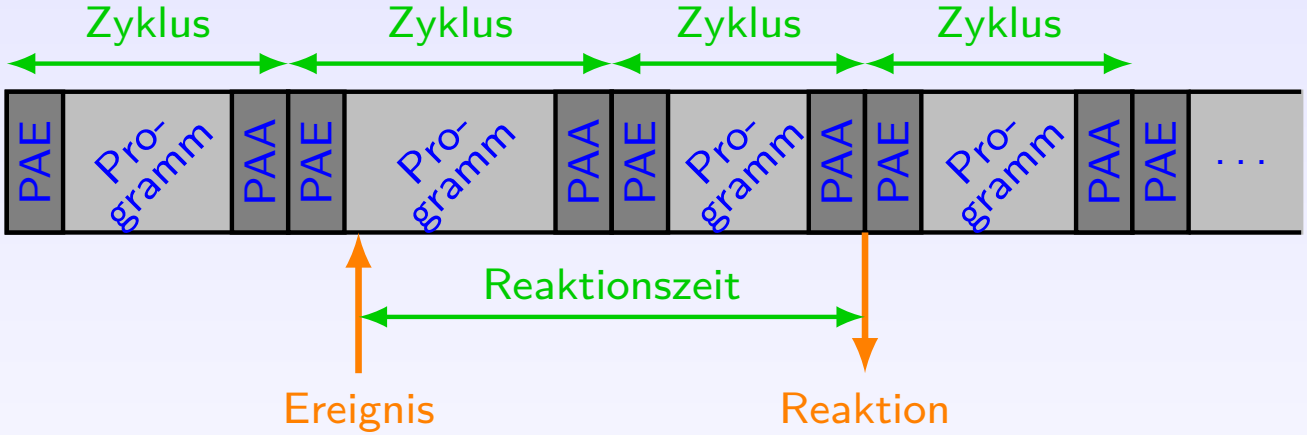
Zeiten bei Programmbearbeitung

Rechenzeit Herstellerangabe zur Beurteilung der CPU-Geschwindigkeit, z.B. für S7-317-2: 0,2 µs für Wortbefehle, 1 µs für Gleitpunktoperationen oder Angabe wie 0,8 ms für 1000 AWL-Befehle

Zykluszeit Zeit für einmalige Programmabarbeitung inkl. Kommunikationsaufgaben, typisch $\leq$10 ms

Zyklusüberwachungszeit Einstellbarer Maximalwert für Zykluszeit; bei Überschreitung geht SPS in Stopp

Reaktionszeit Zeitdauer zwischen Änderung am Eingang und Reaktion durch Ausgangssignal



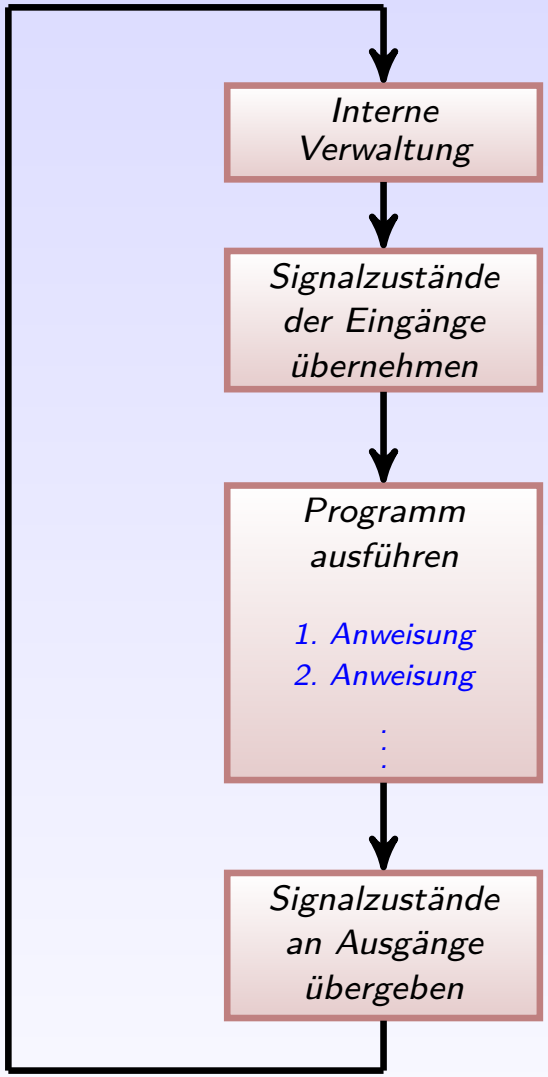
$\Rightarrow$ Reaktionszeit (für „normale“ Ereignisse) = ...

Für zeitkritische Ereignisse: Alarme/Interrupts (spez. Baugruppen)

SPS - Codebausteine

OB	$\hat{=}$ Programmorganisationseinheit	www.sps-lehrgang.de/organisationsbausteine-in-step7/
FC, SFC	kein Speicher	www.sps-lehrgang.de/funktion-fc-sfc/
FB, SFB	Speicher im Instanz-DB	www.sps-lehrgang.de/funktion-fb-sfb/
DB	Instanz-DB, Global-DB	https://www.sps-lehrgang.de/datenbausteine/

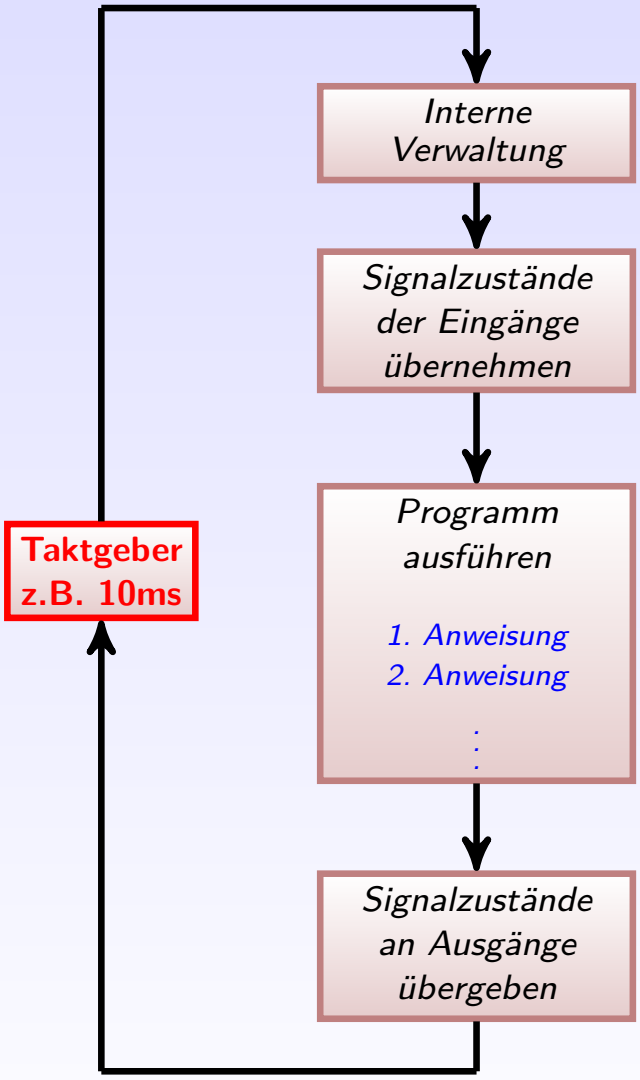
SPS - Zyklische Bearbeitung



- Zyklische Bearbeitung eines SPS-Programms
- Interne Verwaltung: Speicherverwaltung, interne Diagnose (Selbsttest)
- Variable Zeit der Abarbeitung ergibt variable Zyklus- und Reaktionszeit
- Zykluszeit vom Betriebssystem der SPS überwacht
- Programmiergerät zeigt online aktuelle, kürzeste und längste Zykluszeit an
- Step7: Organisationsbaustein OB1

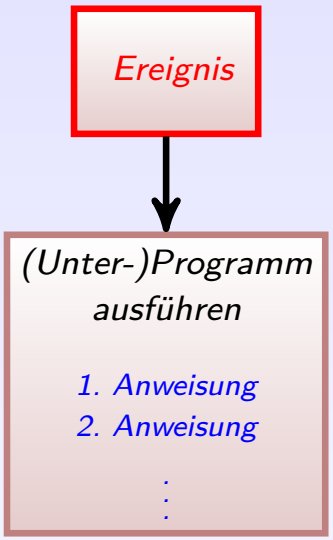
  CoDeSys: Programmorganisationseinheit PLC_PRG
- niedrige Priorität: zyklische Prozesse können von anderen unterbrochen werden
- Standard: Ablaufsteuerungen, Grenzwertüberwachung, langsame Zähler, allgemeine Aufgaben

SPS - Periodische/Zeitgesteuerte Bearbeitung



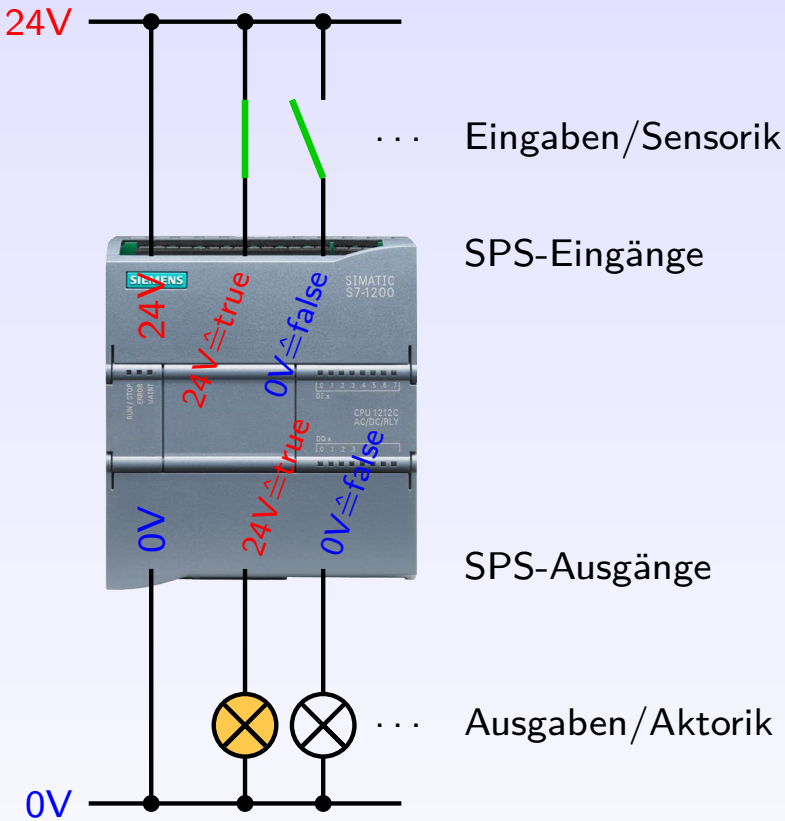
- Periodische Bearbeitung eines SPS-Programms
- Abarbeitung eines Zyklus durch feste Zykluszeit eingestellt
- Typische Anwendung: Regelungstechnik mit festen Abtastintervallen, von Weckalarm OB35 aufgerufen (`www.sps-lehrgang.de/weckalarm-ob/`)
- .....

SPS - Ereignisgesteuerte Programmbearbeitung



- Vergleichbar mit Interrupt bei konventioneller Programmierung
- Für besonders zeitkritische Prozesse
- Höchste Priorität, unterbricht andere Programmteile
- spezielle interruptfähige Digital-Eingangskarten für Auswertung von Ereignissen des Prozesses nötig
- In Step7 durch Fehlerbausteine (z.B. OB86: Ausfall DP-Slave)

SPS - Eingänge & Ausgänge



Eingänge

- SPS erhält Informationen über Prozess von Signalgebern an Eingängen
- Signalgeber: Taster, Endlagenschalter, Temperatursensor,. . .
- Öffner: öffnet bei Betätigung, Normally Closed
- Schließer: schließt bei Betätigung, Normally Opened
- Einschalten
- Ausschalten

Ausgänge

- SPS beeinflusst Prozess, indem Aktoren von Ausgängen mit Steuerspannung (z.B. 24V) beschaltet werden
- Aktoren: Motoren, Ventile, Relais, Lampen, . . .

SPS - Adressierung

- Adressierung: Angabe eines bestimmten Eingangs, Ausgangs oder Merkers
- Signale:
  - ..... (Schaltelemente)
  - ..... (Encoderstände)
  - ..... (Messwerte)

  $\Rightarrow$ .....

Größe	Bits	Kennung	Schlüsselwort
Bit	1	X*	BOOL
Byte	8	B	BYTE
Wort	16	W	WORD
Doppelwort	32	D	DWORD

* entfällt bei S7

- Bits in 8er-Gruppen (Byte) auf DI- oder DO-Baugruppen zusammengefasst $\Rightarrow$ Byteadresse
- Einzelner Ein- oder Ausgang durch Bitadresse (0 bis 7)
- Je nach Sprache Präfix E/A oder I/Q

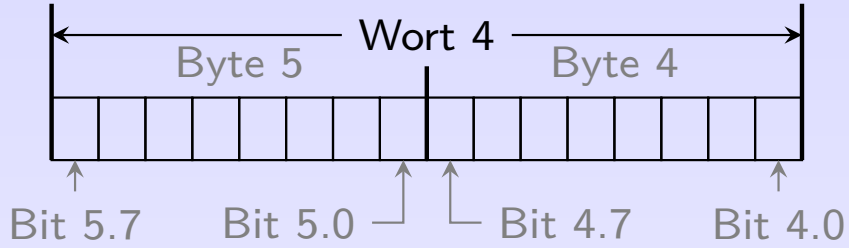
14 Integrierte Digitaleingänge:

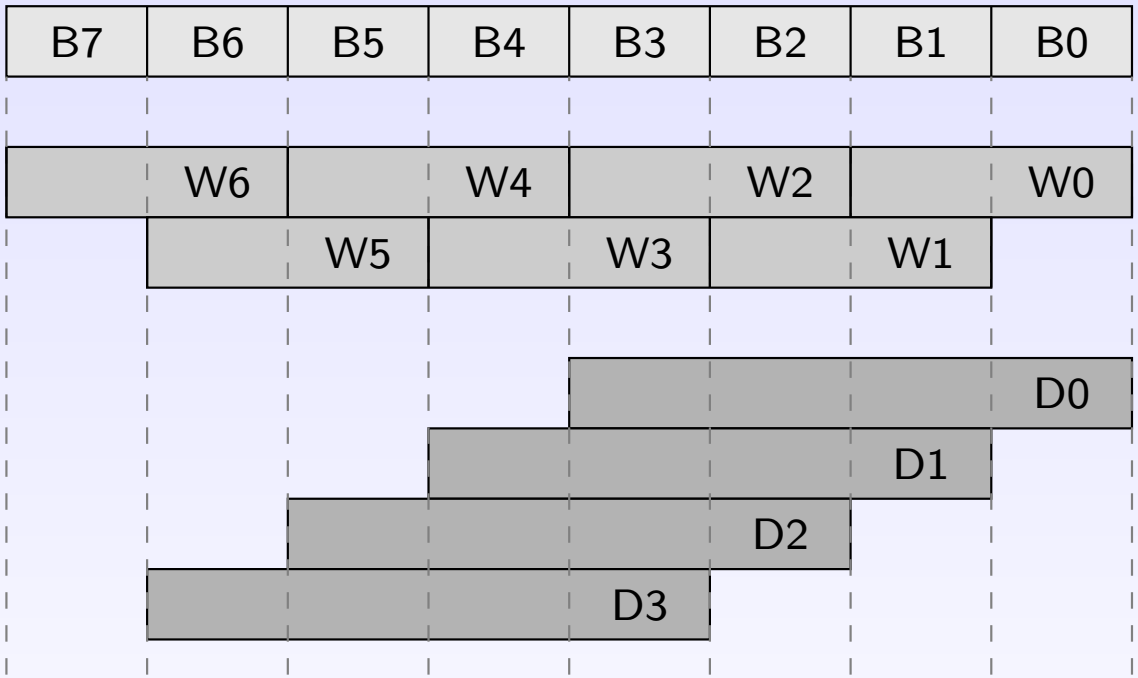
S7-1200

10 Integrierte Digitalausgänge:

SPS - Speicherorganisation

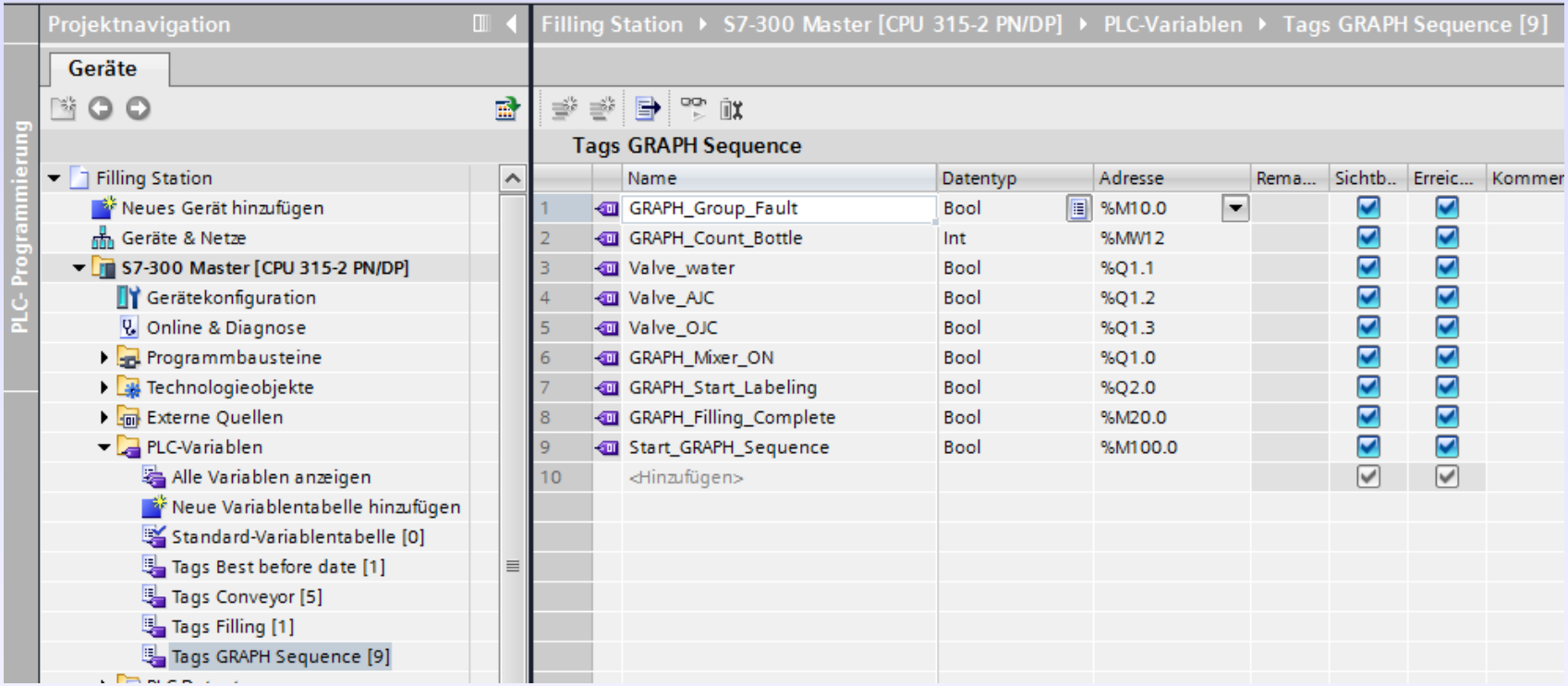
- Adressierung bei Wort/Doppelwort:
  Angabe des niederwertigsten Byte
  Beispiel: Zahl 256 in Wort 4





$\Rightarrow$ . . . . . .

Symbolische Adressierung



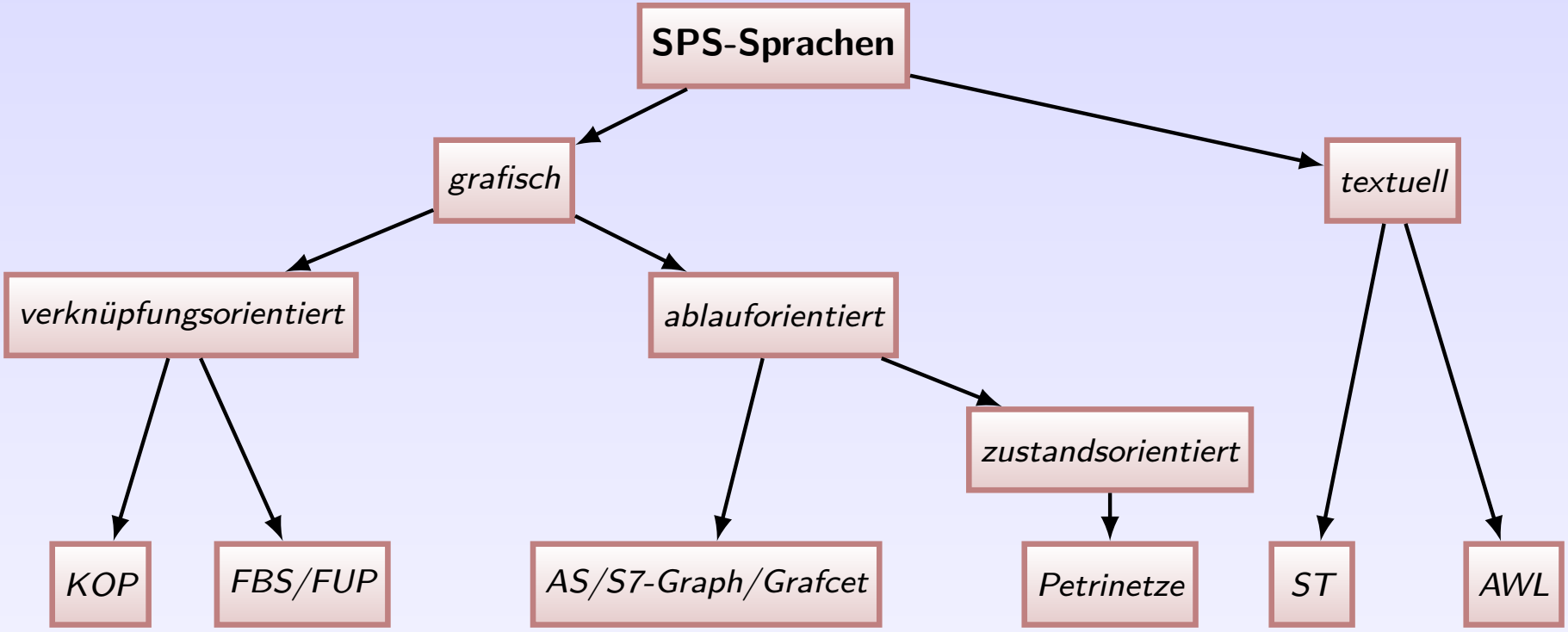
- Vergabe/Deklaration von symbolischen Namen für Variablen und Adressen
- Variablentabelle(n)

\+ Verbesserte Transparenz/Lesbarkeit, weniger Programmierfehler

\+ Bessere Wiederverwertbarkeit und Änderbarkeit des Codes

2. SPS

SPS-Sprachen und grafische Darstellungsarten



verknüpfungsorientiert: .....

ablauforientiert: .....

Norm EN 61131-3: .....

Siemens Step 7 – Steuerungen einfach programmieren

- Software zur Programmierung von Steuerungen der Simatic S7 Familie von Siemens
- seit 1995, hoher Verbreitungsgrad in Deutschland → Quasi-Standard
- Auch andere Hersteller nutzen Step 7 kompatible Steuerungen
- Eigene „Dialekte“ der genormten Sprachen
- siehe: `TIA Portal - Tutorial Center`

CoDeSys – Controller Development System

- Hardwareunabhängig, über 300 Hersteller, hoher Verbreitungsgrad → Quasi-Standard
- seit 1994 von 3S-Smart Software Solutions, Kempten
- Lizensfreies Programmiersystem, jedoch Kosten für Runtime-Lizenzen, manchmal Kosten für spezielle Hardware-Bibliotheken
- Alle fünf IEC-Standardsprachen (AWL, ST, KOP, FBS, SFC)
- Code kann in Maschinencode für viele gängige CPU-Familien übersetzt werden
- `de.codesys.com`

AWL - Anweisungsliste

= STL - Statement List, IL - Instruction List

- unübersichtlich, schlecht wartbar, umständliche Strukturierungsmöglichkeiten nur durch Sprungbefehle
- in Norm als veraltet gekennzeichnet, auf S7-1200 nicht mehr verfügbar

	Step 7		**CoDeSys**	
Eingangsbit	E0.7 . . . E0.0		IX0.7 . . . IX0.0	
Ausgangsbit	A0.7 . . . A0.0		QX0.7 . . . QX0.0	
Eingangsbyte	EB0 = E0.7 . . . E0.0		IB0 = IX0.7 . . . IX0.0	
Ausgangsbyte	AB0 = A0.7 . . . A0.0		QB0 = QX0.7 . . . QX0.0	
Abfrage	UN	E0.0	LDN	IX0.0
UND	U	E1.4	AND	IX1.4
ODER	O	E0.1	OR	IX0.1
Zuweisung	=	A0.0	ST	A0.0
Unbedingter Sprung	SPA	Marke	JMP	Marke
	U	E0.0	LD	IX0.0
Bedingter Sprung	SPB	Marke	JMPC	Marke

KOP - Kontaktplan

= LD - Ladder Diagramm

- Anordnung als Reihen- oder Parallelschaltung zwischen zwei Stromschienen
- ähnelt Elektroschaltplan
- Reihenfolge von oben nach unten und links nach rechts
- Strukturierung durch Sprungbefehle

```
|   E0.0      E0.1       A0.0   |
|---|/|---+---| |---+---( )--|
|         |          |          |
|         |   A0.0   |          |
|         +---| |---+            |
|                                |
|   E0.2                  A0.1   |
|---| |----------------(S)--|
|                                |
|   E0.3                  A0.1   |
|---| |----------------(R)--|
|                                |
|   E0.1                         |
|---| |-->> Marke                |
```

Kontakte:

`--| |--`	Schließer
`--|/|--`	Öffner
`--|P|--`	Schließer bei pos. Flanke
`--|N|--`	Schließer bei neg. Flanke

Spulen:

`--( )--`	Spule
`--(/)--`	Negative Spule
`--(P)--`	Spule bei positiver Flanke
`--(N)--`	Spule bei negativer Flanke
`--(S)--`	Setzspule (Selbsthaltung)
`--(R)--`	Rücksetzspule

Sprünge:

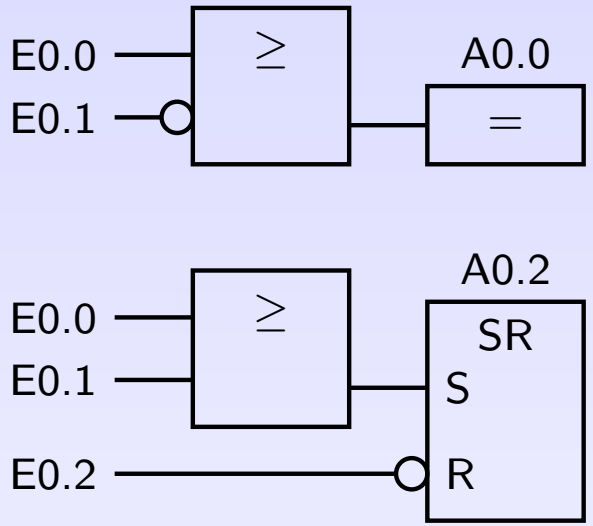
`------->> Marke`	Unbedingter Sprung
`--| |-->> Marke`	Bedingter Sprung
`-------<return>`	Unbedingter Rücksprung
`--| |--<return>`	Bedingter Rücksprung

FBS - Funktionsbausteinsprache

= FBD - Function Block Diagram,

. . . . . .

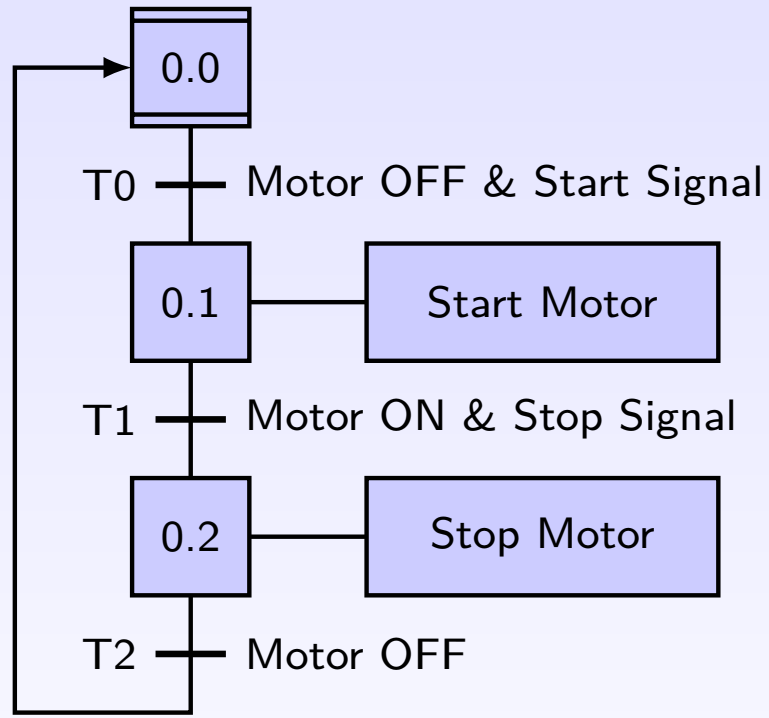
- Simuliert Logikgatter aus IC-Technik
- Grafisch, verknüpfungsorientiert
- Logikgatter, Zähler, Timer, Flip-Flops (SR-rücksetzdominant, RS-setzdominant), Funktionsaufrufe, . . .



AS - Ablaufsprache

= SFC - Sequential Function Chart, ...

- Trotz Normung nicht auf jeder SPS verfügbar (Extra-Lizenzen)
- vernetzte Kette von Steuerungsschritten, verbunden durch Weiterschalt-Bedingungen (Transitionen)
- Genau ein Startschritt
- Transitionen: aussagenlogische Formeln, auch in FUP möglich
- Ausgaben: verknüpfend, speichernd, verzögernd, zeitbegrenzt, ...
- Serielle und parallele Abläufe (Synchronisationsstriche)
- Große Ähnlichkeit mit Petrinetzen (Transitionen, Plätze)



ST - Strukturierter Text

= .....

- Strukturierte/höhere Programmiersprache, ähnlich Pascal
- IF, CASE, FOR, WHILE, REPEAT, EXIT, CONTINUE, RETURN, Funktionsaufrufe, Pointer, Felder, Arithmetik, Logik, Mathematische Funktionen

```
IF <ausdruck> THEN
    <anweisung>
ELSIF <ausdruck> THEN
    <anweisung>
ELSE
    <anweisung>
END_IF;
```

```
CASE <variable> OF
    1,2: <anweisung>
    7: <anweisung>
    ELSE <anweisung>
END_CASE;
```

```
FOR i:=10 TO 20 BY 2
    <anweisung>
END_FOR;
```

Literatur, Vertiefung - SPS

WIKIPEDIA Speicherprogrammierbare Steuerungen
de.wikipedia.org/wiki/Speicherprogrammierbare_Steuerung

WIKIPEDIA Programmiersprachen nach EN 61131
de.wikipedia.org/wiki/EN_61131

AWL de.wikipedia.org/wiki/Anweisungsliste
Simatic AWL_d.pdf

KOP de.wikipedia.org/wiki/Kontaktplan
Simatic KOP_d.pdf

FBS de.wikipedia.org/wiki/Funktionsbausteinsprache
Simatic s7fup_a.pdf

AS de.wikipedia.org/wiki/Ablaufsprache
Simatic Graph7_d.pdf

ST de.wikipedia.org/wiki/Strukturierter_Text
Simatic SCL_d.pdf

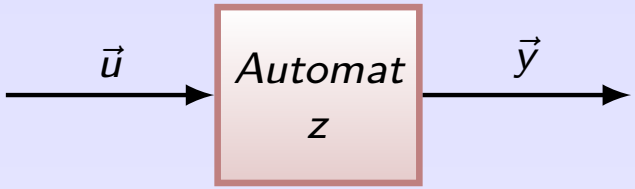
TIA TIA Portal - Tutorial Center

[WZ09] Wellenreuther, Zastrow: Automatisieren mit SPS

3. Modellierung mit endlichen Automaten

Endliche Automaten

- Engl. Finite State Machine
- von Moore (1955) und Mealy (1956)
- Allgemeiner, flexibler Modellierungsansatz für sequentielle Abläufe
- Natürlichsprachliche Beschreibung $\rightarrow$ Formale Spezifikation
- Endlich $\hat{=}$ Endlicher Speicher, Endliche Anzahl an Zuständen
- Begriffe: Eingabe $\vec{u}$, Ausgabe $\vec{y}$, Zustand z



Signale, Eingaben und Ausgaben

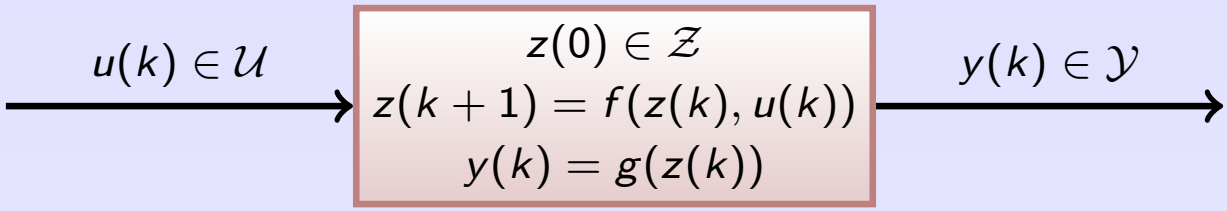
- Eine Ein- oder Ausgabe kann sich auf mehrere Signale gleichzeitig beziehen
- Eingabe: $\vec{u} \in \{0, 1, *\}^n \hat{=} \{\text{false, true, } \ldots\ldots\ldots\ldots \}^n$
- Ausgabe: $\vec{y} \in \{0, 1, -\}^n \hat{=} \{\text{false, true, } \ldots\ldots\ldots\ldots \}^n$
- Bei n binären Komponenten eines Signalvektors $\rightarrow 2^n$ unterschiedliche Kombinationen
  I.d.R. kommt nur eine kleine Teilmenge vor

Definition: Endlicher Automat (Moore-Modell)

Ein endlicher Automat ist ein Tupel $A = (\mathcal{Z}, \mathcal{U}, \mathcal{Y}, f, g, z(0))$. Dabei ist

- $\mathcal{Z} = \{Z_1 \dots Z_{n_Z}\}$ eine endliche Menge von Zuständen,
- $\mathcal{U} = \{U_1 \dots U_{n_U}\}$ eine endliche Eingabemenge,
- $\mathcal{Y} = \{Y_1 \dots Y_{n_Y}\}$ eine endliche Ausgabemenge,
- $z(0) \in \mathcal{Z}$ der Startzustand,
- $f : \mathcal{Z} \times \mathcal{U} \to \mathcal{Z}$ die Zustandsübergangsfunktion und
- $g : \mathcal{Z} \to \mathcal{Y}$ die Ausgabefunktion.

■



Automatengraf

= State Chart, State-Transition-Diagram

- gerichteter Graf mit Knoten als Zustände (Elemente der Menge $\mathcal{Z}$)
- gerichtete Kanten beschreiben Übergänge (Transitionen) zwischen den Zuständen (f)
- Kantenbeschriftung U_k gemäß f
- Ausgaben Y_k in/an Knoten gemäß g
- Startzustand $z(0)$ durch Symbol $>$ gekennzeichnet

3. Modellierung mit endlichen Automaten

Beispiel: Fahrkartenautomat

Eine Fahrkarte soll 3€ kosten. Es sind nur 1€ und 2€ Münzen zugelassen. Der Automat gibt bei Überbezahlung das Wechselgeld zurück.

Tupelschreibweise

$\mathcal{U} =$

$\mathcal{Y} =$

$\mathcal{Z} =$

$z(0) =$

$f =$

$g =$

Automatengraf

Wie viele Elemente hätten Eingabe- und Zustandsmenge, falls der Automat alle gängigen Münzen annehmen würde?

$\mathcal{U} = \{\}, |\mathcal{U}| = \ldots$

$\mathcal{Z} = \{\}, |\mathcal{Z}| = \ldots\ldots$

Beispiel: Steuerung eines Sammelbeckens

Modellieren Sie die Steuerung eines Sammelbeckens durch einen Endlichen Automat.

- Befüllung wird mit Taster S1 gestartet; das Becken wird über das Zulaufventil Y1 gefüllt.
- Beim oberen Füllstand meldet Schwimmschalter S2 ein 1-Signal, worauf die Befüllung endet und das Becken über das Ablaufventil Y2 vollständig geleert wird.
- Ist das Becken entleert, meldet der untere Schwimmschalter S3 ein 0-Signal.
- Durch einen Aus-Schalter (0-Signal) kann das Füllen und Leeren jederzeit unterbrochen werden.

Zustände: .....

.....

.....

Ausgaben: .....

.....

Eingaben: .....

.....

.....

.....

Beispiel: Lastenaufzug

Ein kleiner Lastenaufzug für vier Etagen wird nur von außen bedient. Es gibt pro Etage vier Taster, um den Aufzug in die entsprechende Etage zu schicken. Die Türen werden manuell bedient. Ein Öffnen der Türe ist mechanisch nur möglich, wenn der Aufzug in der entsprechenden Etage steht. Türkontakte signalisieren, ob die Türen geschlossen sind.

Gesucht: Eingabemenge, Ausgabemenge, Automatengraf

Eingabemenge: pro Etage i

- .....
- .....
- .....

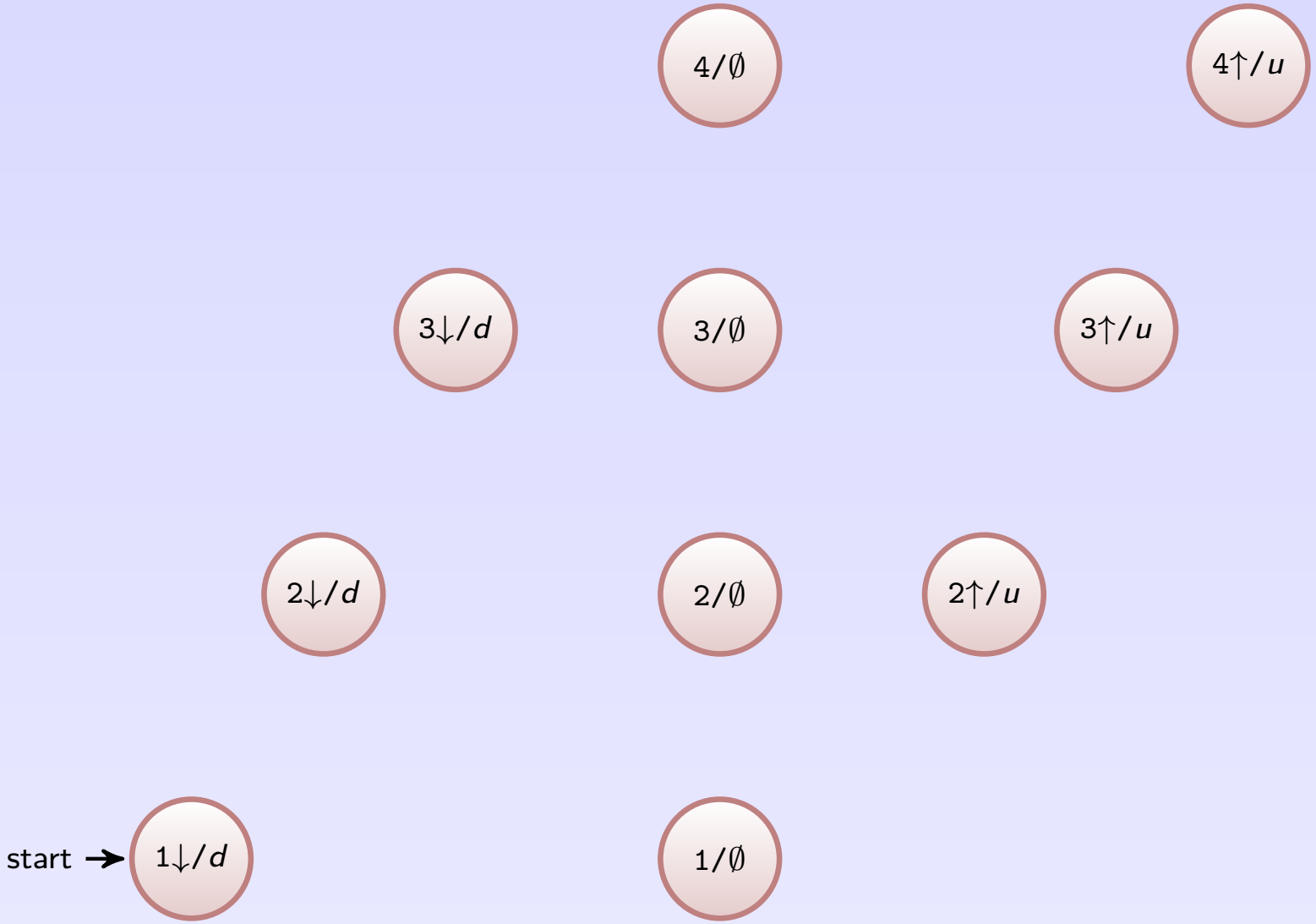
$\rightarrow$ .....

Ausgabemenge:

- .....
- .....

$\rightarrow$ .....

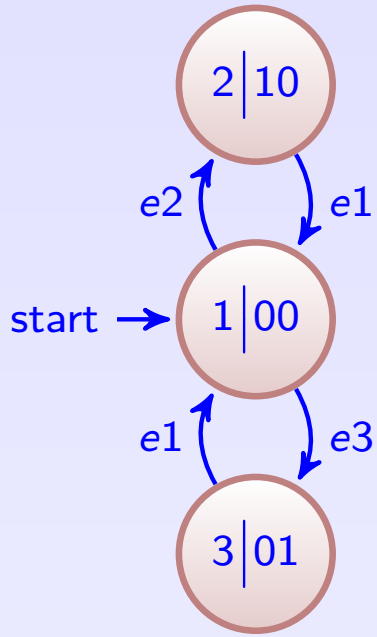
Automatengraf:



$$e_i = (e_{T1} \wedge e_{T2} \wedge e_{T3} \wedge e_{T4}) \wedge (e_{1i} \vee e_{2i} \vee e_{3i} \vee e_{4i})$$

SPS-Code

```
VAR
    state : INT  := 1;      // Zustand, statisch
END_VAR;
VAR_TEMP
    e1    : BOOL := E0.0;   // Eingabe der SPS, Symboltabelle
    e2    : BOOL := E0.1;   // Eingabe der SPS, Symboltabelle
    e3    : BOOL := E0.2;   // Eingabe der SPS, Symboltabelle
    a1,a2 : BOOL;           // Ausgaben der SPS, Symboltabelle
END_VAR;
CASE state OF
    1 : IF (e2) THEN state := 2 END_IF;
        IF (e3) THEN state := 3 END_IF;
    2 : IF (e1) THEN state := 1 END_IF;
    3 : IF (e1) THEN state := 1 END_IF;
END_CASE;
CASE state OF
    1 : a1:=false; a2:=false; // Ausgabe 1 = 00
    2 : a1:=true;  a2:=false; // Ausgabe 2 = 10
    3 : a1:=false; a2:=true;  // Ausgabe 3 = 01
END_CASE;
A0.0 := a1;
A0.1 := a2;
```



Begriffe

DEA - Deterministischer EA
In . . . . . Zustand ist der Folgezustand bei . . . . . Eingabe eindeutig festgelegt.

NEA - Nichtdeterministischer EA
Bei . . . . . einem Zustand ist der Folgezustand bei einer Eingabe nicht eindeutig festgelegt

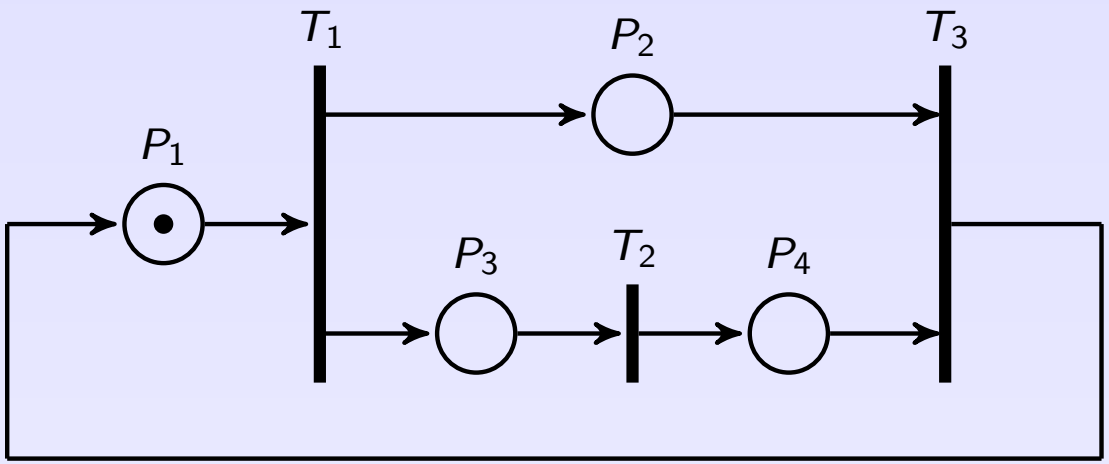
4. Modellierung mit signalinterpretierten Petrinetzen

Modellierung mit Petrinetzen

- Dissertation von C. A. Petri (1962) „Kommunikation mit Automaten“, engl. Petri-Net
- Allgemeiner, flexibler Modellierungsansatz
- Natürlichsprachliche Beschreibung $\rightarrow$ Formale Spezifikation

+ Modellierung von seriellen, parallelen und nebenläufigen Abläufen
+ Grafischer Entwurf
+ Einfache Umwandlung in Programmcode

- Ein Petrinetz ist ein gerichteter, bipartiter Digraph.



P_i Platz, T_i Transition, • Markierung, $\rightarrow$ Kante

Komponenten eines Petrinetzes und Begriffe

Knoten Plätze + Transitionen

Plätze dargestellt durch Kreise, repräsentieren Zustände, z.B. „Lampe leuchtet“

Transitionen dargestellt durch Striche/Rechtecke, repräsentieren Zustandsübergänge, z.B. „Übergang von Hand in Automatikbetrieb“

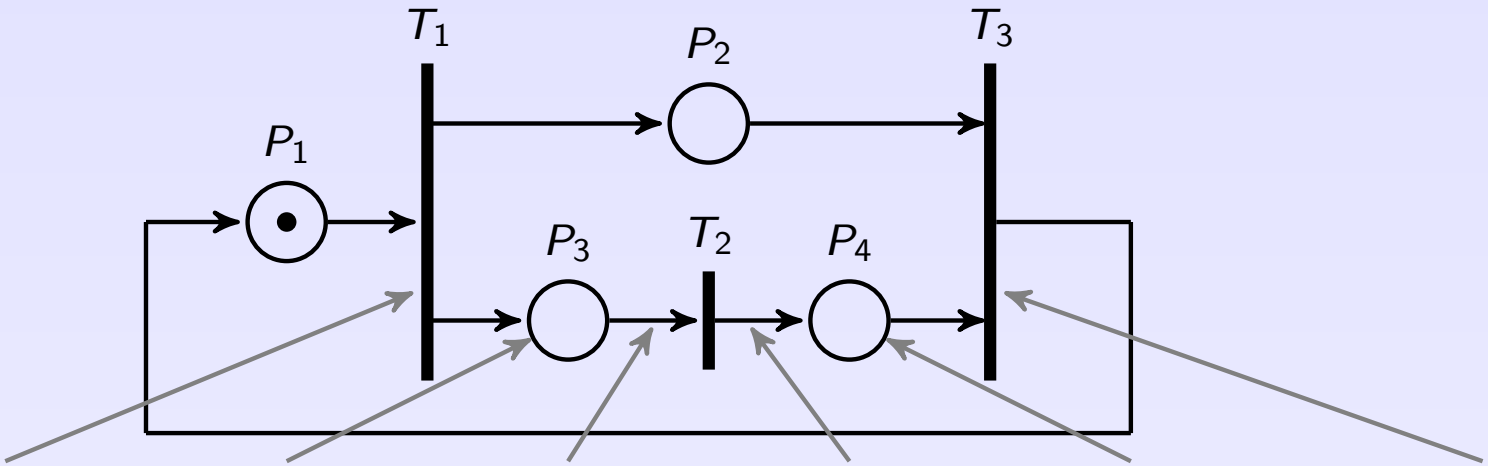
Kanten verbinden Plätze mit Transitionen und umgekehrt

Marken Plätze können . . .

- eine (BEN - Bedingungs-Ereignis-Netz, SIPN)
- mehrere (gleichartige) oder (PTN - Platz-Transitionen-Netz)
- mehrere unterscheidbare (Gefärbte Netze)

Marken besitzen, die durch Transitionen im PN geschaltet werden

Begriffe



von P_2 und P_3 von T_2 von T_2 von T_2 von T_2 von P_2 und P_4

BEN - Bedingungs-Ereignis-Netz

- Der Besitz einer Marke im Vorplatz kann als . . . . . gedeutet werden.
- Die Transition kann dann als . . . . . die Marke weiterschalten.

Ein **Bedingungs-Ereignis-Netz** ist ein Tupel $P = (\mathcal{P}, \mathcal{T}, \mathcal{K}, \mathbf{m}(0))$ mit

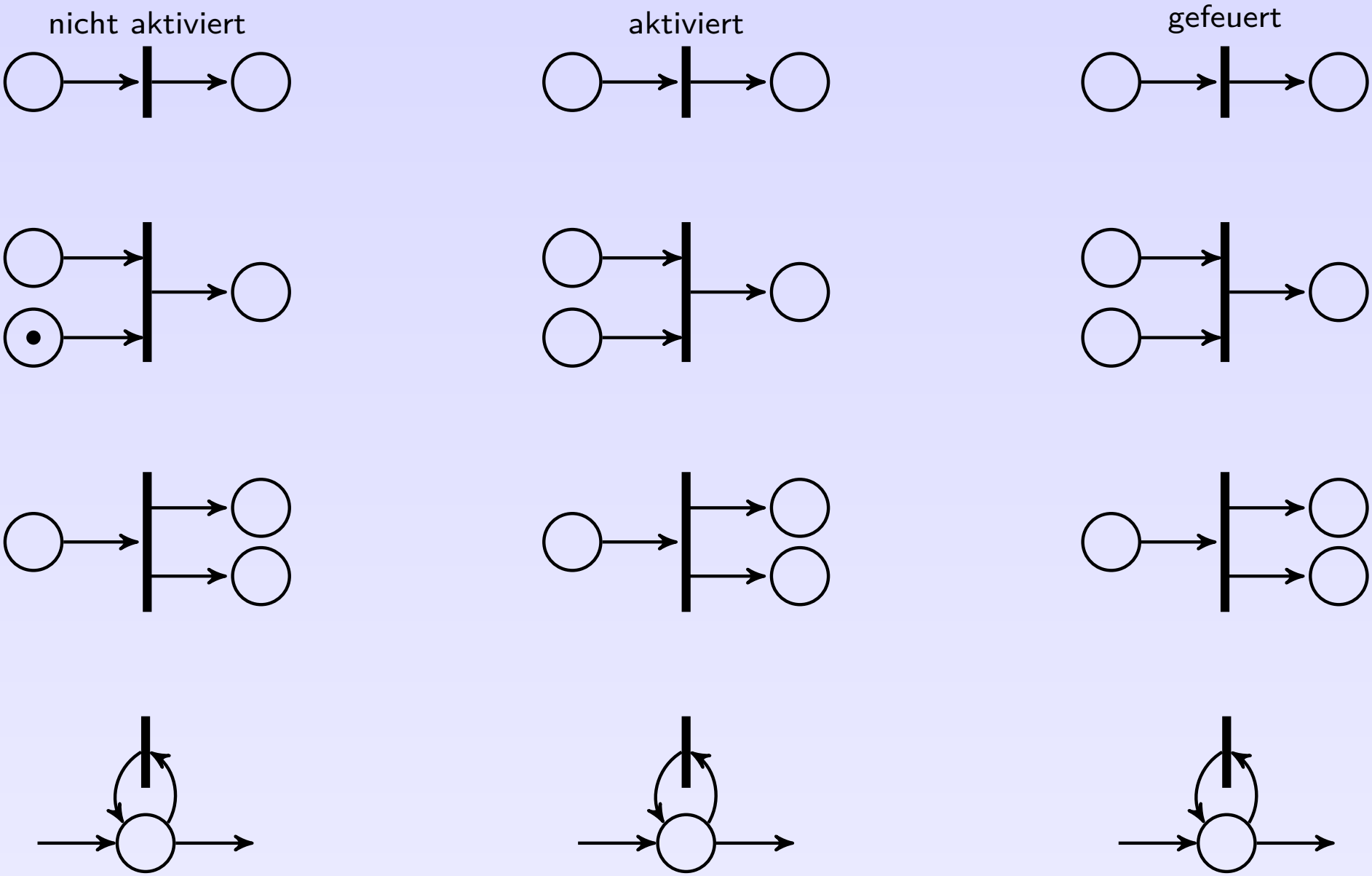
- $\mathcal{P} = \{P_1, \ldots, P_{|\mathcal{P}|}\}$ endliche, nichtleere Platzmenge,
- $\mathcal{T} = \{T_1, \ldots, T_{|\mathcal{T}|}\}$ endliche, nichtleere Transitionenmenge,
  $\mathcal{P} \cap \mathcal{T} = \emptyset$, d.h. $\mathcal{P}$ und $\mathcal{T}$ sind disjunkt,
- $\mathcal{K} \subseteq \mathcal{P} \times \mathcal{T} \cup \mathcal{T} \times \mathcal{P}$ Kantenmenge und
- $\mathbf{m}(0) = [m_1(0), \ldots, m_{|\mathcal{P}|}(0)]^{\mathrm{T}}$ Anfangsmarkierung mit $\mathbf{m} : \mathcal{P} \to \{0, 1\}$.

Schaltregeln

Konzessionsregel Eine Transition ist genau dann aktiviert/schaltbereit, wenn alle ihre Vorplätze markiert sind und durch das Schalten in keinem ihrer Nachplätze die Markenanzahl größer Eins wird.

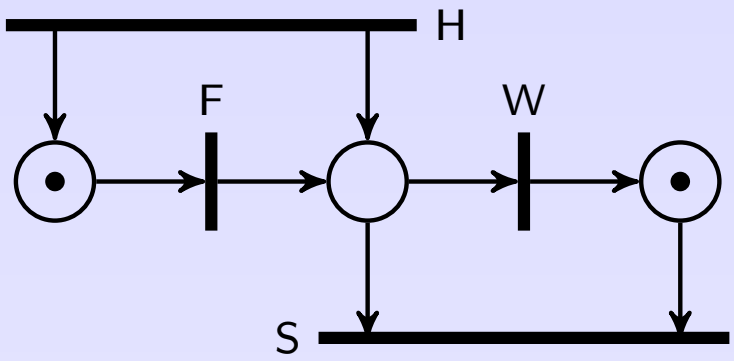
Markenflussregel Beim Schalten einer Transition wird jedem ihrer Vorplätze eine Marke entnommen und jedem ihrer Nachplätze eine Marke hinzugefügt.

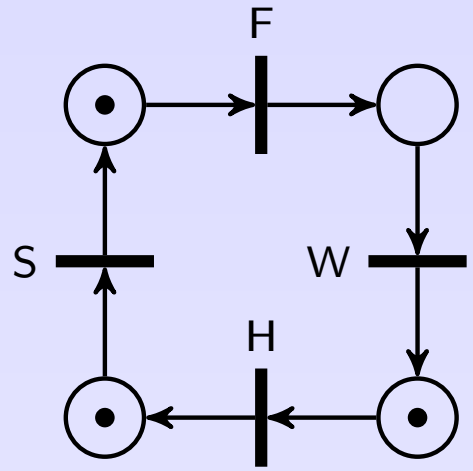
Schaltregeln - Beispiele

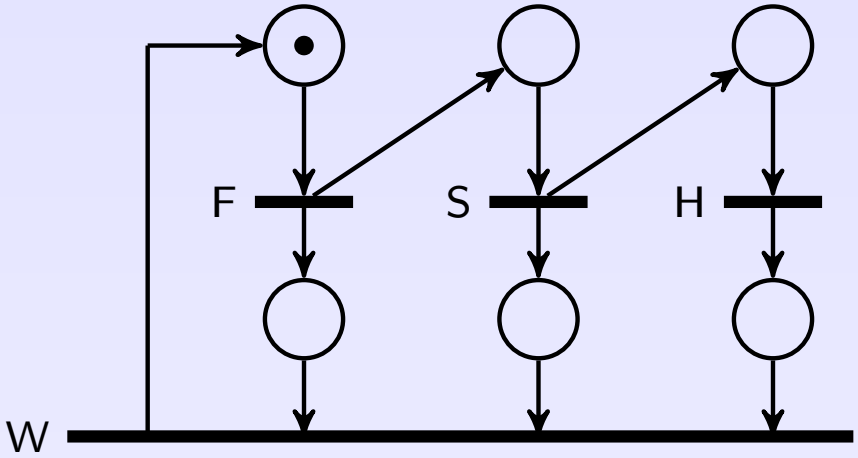


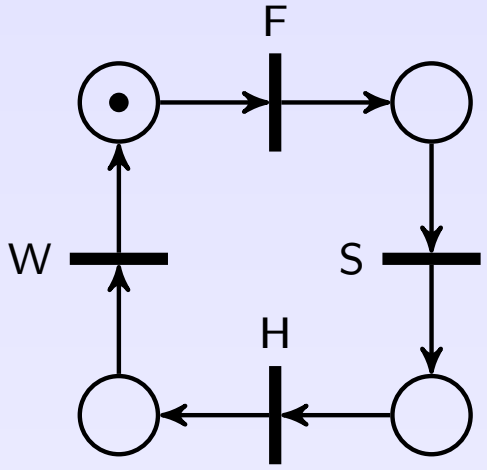
Beispiele

Ein PN ist ein generatives Schema für Folgengeflechte.
In welcher Folge können die Transitionen schalten?



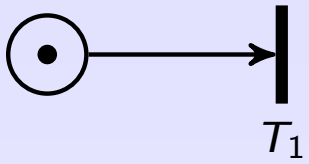






BEN - Beispiel

Erzwingen Sie das Schaltverhalten $T_1, T_1, T_1, T_2, T_1, T_1, T_1, T_2, \ldots$ durch ein BEN. Dabei ist das Schalten von anderen Hilfstransitionen zwischendurch unerheblich.

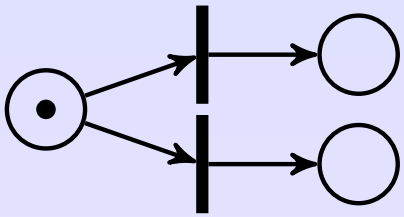


Konflikte

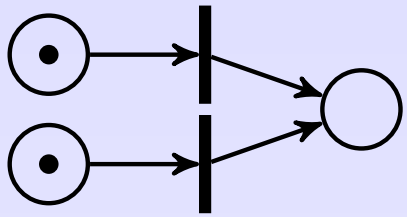
In **Konflikt** stehende Transitionen sind aktivierte Transitionen, bei denen das .....

der einen zur .....der anderen führt.

Rückwärtskonflikt: Das Schalten einer Transition entzieht dem Vorplatz einer anderen Transition die Marke



Vorwärtskonflikt: Das Schalten einer Transition belegt den Nachplatz einer anderen Transition



Beachte: **Indeterminismus** wird bei Implementierung in Programmcode aufgrund der Reihenfolge

in **Determinismus** überführt $\Rightarrow$ .....

Priorisierung bei Konflikten:

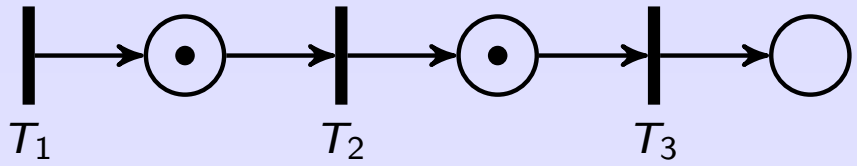
- durch Programmierreihenfolge $\rightarrow$ implementierungsabhängig, zufällig?
- durch Transistionspriorität $\rightarrow$ z.B. kleinere Nummern (lexikalische Reihenfolge) haben Vorrang

$\Rightarrow$ Priorisierung möglich, aber wenig transparent/übersichtlich $\rightarrow$ besser: Konfliktvermeidung

Kontakte

Beim **Kontakt** ermöglicht das Schalten einer Transition sofort das Schalten einer weiteren Transition.

Beispiel:



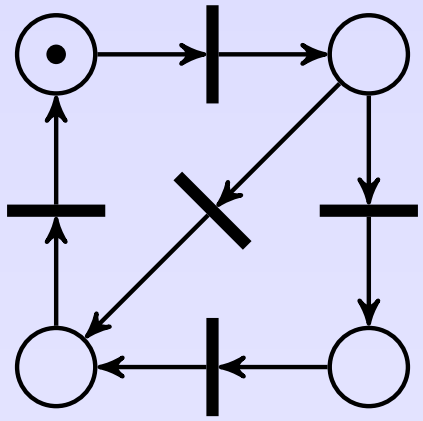
Bei Kontakten ist das zeitliche Verhalten abhängig von der Implementierung

Reihenfolge T_1, T_2, $T_3 \rightarrow T_2$ und T_3 schalten .....

Reihenfolge T_3, T_2, $T_1 \rightarrow T_2$ und T_3 schalten .....

Zustandsmaschine - Spezialfall des BEN

Die **Zustandsmaschine** ist ein spezielles Bedingungs-Ereignis-Netz, in dem jede Transition genau einen Vorplatz und höchstens einen Nachplatz hat.



- Anzahl der Marken kann nicht zunehmen, bei entsprechender Anfangsmarkierung stets ≤ 1
- Falls nur eine Marke zu Beginn, entspricht Markenbesitz in Platz p_i dem Zustand eines EA

$\Rightarrow$ Die Zustandsmaschine ist dem EA sehr ähnlich

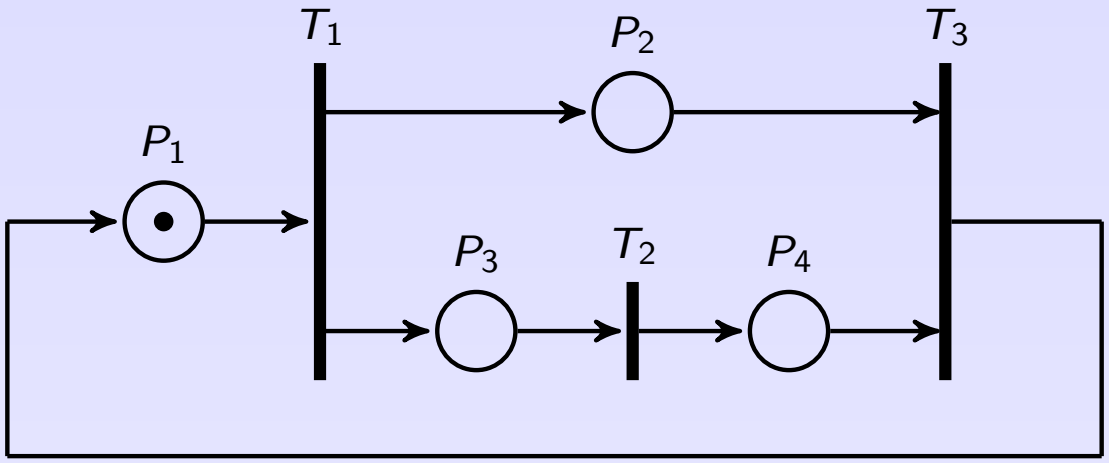
$+$ sehr übersichtlich

$+$ einfache Implementierung als `Switch-Case`-Anweisung

$-$ Parallele oder nebenläufige Abläufe können nicht modelliert werden

Synchronisationsgraf - Spezialfall des BEN

Der **Synchronisationsgraf** ist ein spezielles Bedingungs-Ereignis-Netz, bei dem jeder Platz höchstens eine Vortransition und höchstens eine Nachtransition besitzt.

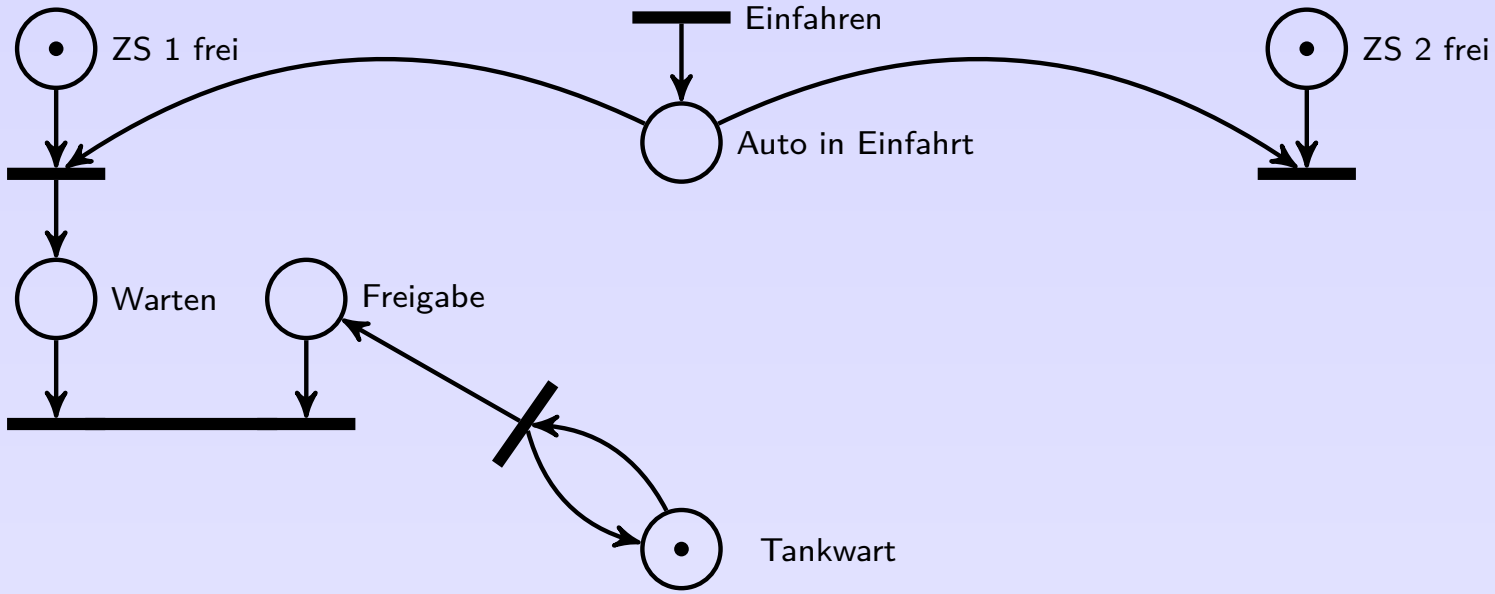


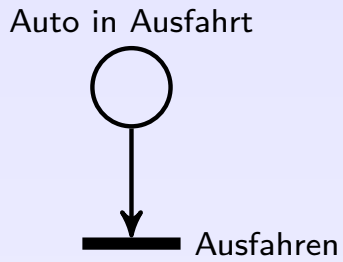
- Jeder Platz kann nur von einer einzigen Transition aktiviert/deaktiviert werden
- Eine Transition führt eine Startsynchronisation ihrer Nachplätze durch (falls sie mehrere Nachplätze hat). Teilprozesse werden hier gleichzeitig gestartet.
- Eine Transition führt eine Abschlusssynchronisation ihrer Vorplätze durch (falls sie mehrere Vorplätze hat). Teilprozesse werden hier gleichzeitig beendet.
- Synchronisationsgrafen sind stets konfliktfrei.

Übung: BEN mit Nebenläufigkeit

Erstellen Sie das Systemmodell einer Tankstelle mit zwei Zapfsäulen als Petrinetz.
Hinweise:

- In Ein- und Ausfahrt und an den Zapfsäulen ist nur Platz für jeweils ein Auto, d.h. insgesamt sind maximal vier Autos auf dem Gelände.
- Das Tanken erfordert die Freigabe eines Tankwarts. Er kann nicht mehrere Kunden gleichzeitig bedienen.
- Das Ausfahren ist nicht mit dem Einfahren eines neuen Autos gekoppelt.
- Aktionen eines Kunden: in Einfahrt, an ZS warten, Tanken, Zahlen, Fertig, in Ausfahrt
- Aktionen des Tankwarts: kassieren, warten


ZS 1 frei
Einfahren
ZS 2 frei
Auto in Einfahrt
Warten
Freigabe
Tankwart


Auto in Ausfahrt
Ausfahren

Netzeigenschaften und Netzanalyse

Ziel: Überprüfung der Korrektheit eines Steuerungsprogramms

Weg 1: Sofort programmieren, anschließend debuggen und testen
→ üblich, fehleranfällig, iteratives Vorgehen

Weg 2: PN-Entwurf, anschließend manuell programmieren, debuggen und testen
→ besser, aber Fehler werden erst spät gefunden

Weg 3: PN-Entwurf mit Analyse, programmieren mit debuggen oder automatische Codegenerierung, testen
→ noch besser, Fehler werden früh gefunden

Erreichbarkeit Die Markierung $\mathbf{m}_i$ heißt von $\mathbf{m}_j$ aus erreichbar, wenn eine Transitionenfolge existiert, die $\mathbf{m}_j$ in $\mathbf{m}_i$ überführt.
→ d.h. im Erreichbarkeitsgraf existiert ein Pfad von $\mathbf{m}_j$ nach $\mathbf{m}_i$.

Lebendigkeit Ein PN heißt lebendig, wenn ..... Transition durch eine geeignete Schaltungsfolge immer wieder aktiviert werden kann.

Partielle Verklemmung Es existiert ..... Transition, die nie wieder aktiviert werden kann, während das bei anderen noch möglich ist.

Totale Verklemmung Bei einer bestimmten Markierung kann ..... Transition mehr aktiviert werden. (Deadlock)

Erreichbarkeitsmenge und Erreichbarkeitsgraf

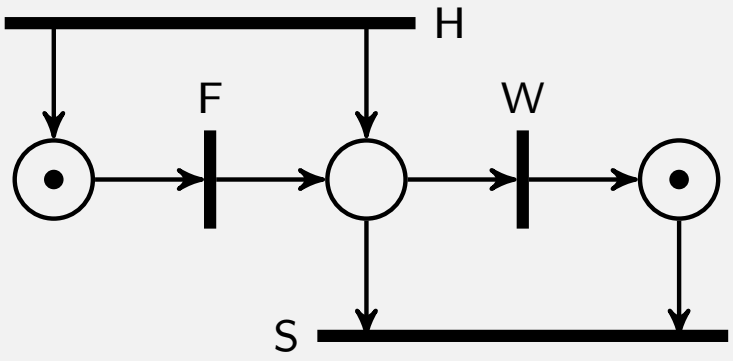
Die Erreichbarkeitsmenge (EM) $\mathcal{E}(m(0))$ enthält alle von der Anfangsmarkierung $m(0)$ aus erreichbaren Markierungen $m(i), i \geq 0$.

$\mathcal{E}(m(0)) = \{m(0), m(1), m(2), \ldots\}$.

Der Erreichbarkeitsgraf (EG) $\mathcal{G}(m(0))$ enthält als Knoten die Elemente (Markierungen) der EM und als Kanten die Transitionen, welche die Markierungen ineinander überführen.

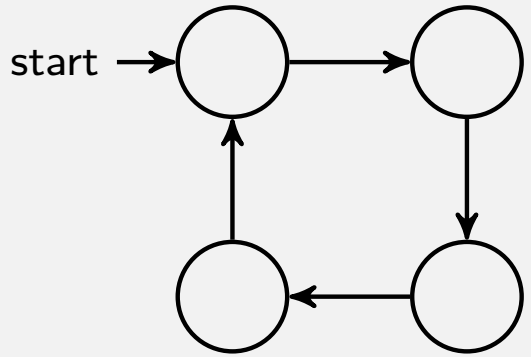
Beispiel

BE-Netz:



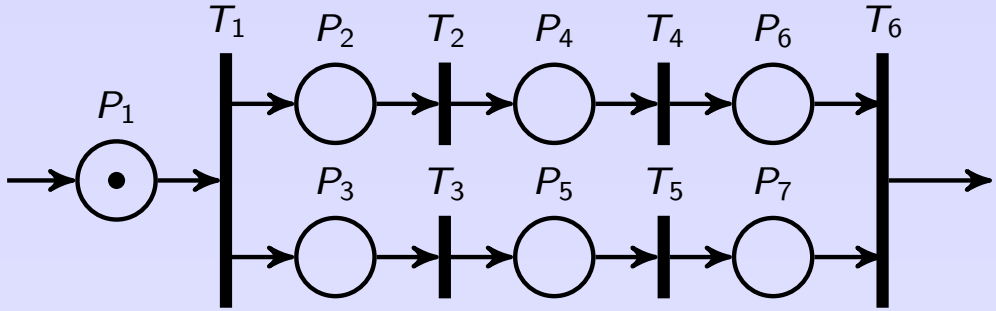
Ereichbarkeitsmenge:
$\mathcal{E}(101) = \{$ $\}$

Ereichbarkeitsgraf:



Im EG lassen sich die Eigenschaften Erreichbarkeit, Lebendigkeit und Verklemmung einfach (Rechner-gestützt) ermitteln

Gegeben ist der folgende Teil eines SIPN



- Um welche Art SIPN handelt es sich? . . . . . .
- Wo erkennt man sequentielles, nebenläufiges oder paralleles Verhalten
- Konstruieren Sie den Erreichbarkeitsgraf

- Wie viele Knoten hätte der EG, wenn das SIPN 3 Zweige mit je 4 Plätzen hätte? . . . . . .

Endlicher Automat versus Petrinetz

Endlicher Automat	**Petrinetz**
1-partiter, gerichteter Graph	bipartiter, gerichteter Graph
Endliche Zustandsmenge	Endliche Menge von Transitionen und Plätzen
Startzustand	Anfangsbelegung
Immer genau ein Zustand aktiv	kein, genau ein oder mehrere Transitionen aktiv
Modell für sequenzielle Abläufe	Modell für sequentielle, parallele und nebenläufige Abläufe

Ein EA kann leicht in ein PN überführt werden.
Ein PN kann nur für Spezialfälle einfach in EA überführt werden.

4. Modellierung mit signalinterpretierten Petrinetzen

Signalinterpretiertes Petrinetz (SIPN)

- *Konventionelles* PN enthält keine Eingaben und Ausgaben
- Für Steuerungstechnik besonders geeignet ist das *signalinterpretierte PN (SIPN)*
- Informationsfluss von außen (Eingaben):
  Markierung der Plätze bestimmt nur Schaltbereitschaft
  Genauer Schaltzeitpunkt wird über *Eingaben an Transitionen* festgelegt.
- Informationsfluss nach außen (Ausgaben):
  Die Markierung eines Platzes ist ein Ereignis, dem eine *Ausgabe* zugeordnet wird.
- Modellierung von *Zeitbewertungen* möglich (z.B. Delay)
- Modellierung von *Hierarchien* möglich

Ein **SIPN** ist ein 8-Tupel $(\mathcal{P}, \mathcal{T}, \mathcal{K}, \mathbf{m}(0), \mathcal{E}, \mathcal{A}, \mathcal{X}, \mathcal{Y})$ mit

- $\mathcal{P}, \mathcal{T}, \mathcal{K}$ und $\mathbf{m}(0)$ wie beim BEN,
- $\mathcal{E} = \{e_1, e_2, \ldots\}$ Eingangssignale,
- $\mathcal{A} = \{a_1, a_2, \ldots\}$ Ausgangssignale, $\mathcal{E} \cap \mathcal{A} = \emptyset$
- $\mathcal{X} : \mathcal{T} \to \mathrm{AL}(\mathcal{E})$ Schaltbedingungen (AL-Formeln) und
- $\mathcal{Y} : \mathcal{P} \to \{0, 1, -\}^{|\mathcal{A}|}$ Aktionen.

Signalinterpretiertes Petrinetz (SIPN)

Schaltregeln

Konzessionsregel Eine Transition ist genau dann aktiviert, wenn alle ihre Vorplätze markiert sind und durch das Schalten in keinem Platz die Markenanzahl größer Eins wird.

Markenflussregel Beim Schalten einer Transition wird jedem ihrer Vorplätze eine Marke entnommen und jedem ihrer Nachplätze eine Marke hinzugefügt.

Synchronisationsregel Alle aktivierten, nicht in Konflikt stehenden Transitionen schalten genau dann, wenn ihre Schaltbedingungen erfüllt sind (also sofort und gleichzeitig).

Iteratives Schalten Bei gegebener Eingabe werden die Schaltvorgänge so lange fortgesetzt, bis keine Transition mehr schalten kann („stabile Markierung“).

Ausgabe des SIPN

- Nicht jeder Platz legt jeden Ausgang explizit fest
- Gesamtausgabe muss eindeutig und widerspruchsfrei sein
- Verknüpfung der Teilausgaben aller markierten Plätze zur Gesamtausgabe

$$a_{\mathrm{Ges}} = \begin{cases} 1 & \forall a_i : a_i \in \{1, -\} \\ 0 & \forall a_i : a_i \in \{0, -\} \\ - & \forall a_i : a_i \in \{-\} \quad \text{(undefined)} \\ w & \textit{sonst} \quad \text{(Widerspruch)} \end{cases}$$

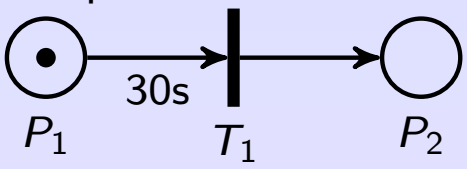
P1	nicht markiert
P2	000 111 ---
P3	01- 01- 01-
a_{Ges}	

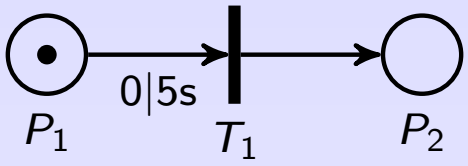
Zeitbewertetes SIPN

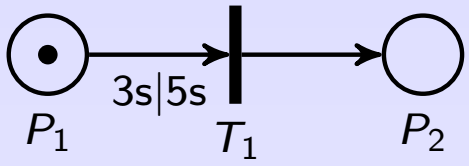
Um zeitliche Bedingungen beim Schalten zu modellieren, erhält jede Präkante eine Zeitbewertung $(t_{min}|t_{max})$, so dass die betreffende Transition nur im Zeitraum $t_{min} \leq t \leq t_{max}$ schalten kann, wobei $t = 0$ der Zeitpunkt der Aktivierung ist.

Eine fehlende Zeitbewertung impliziert $(0|\infty)$; eine einzige Zahl τ bedeutet $(\tau|\infty)$.

Beispiele:





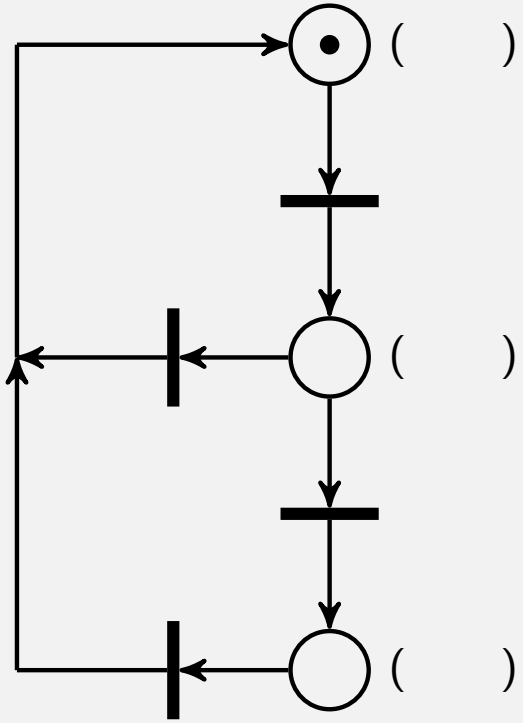


Konzessionsregel Eine Transition ist genau dann aktiviert, wenn alle ihre Vorplätze während ihrer Präkanten-Zeitintervalle $t \in [t_{min}, t_{max}]$ markiert sind und durch das Schalten in keinem Platz die Markenanzahl größer Eins wird.

Zeitbewertetes SIPN

Entwerfen Sie das SIPN für einen Motor, der mit zwei Tastern ein- und ausgeschaltet wird. Bei Überhitzung gibt ein Temperatursensor ein binäres Signal. Der Motor soll im Stern anlaufen und nach 5s in Dreieck umgeschaltet werden.

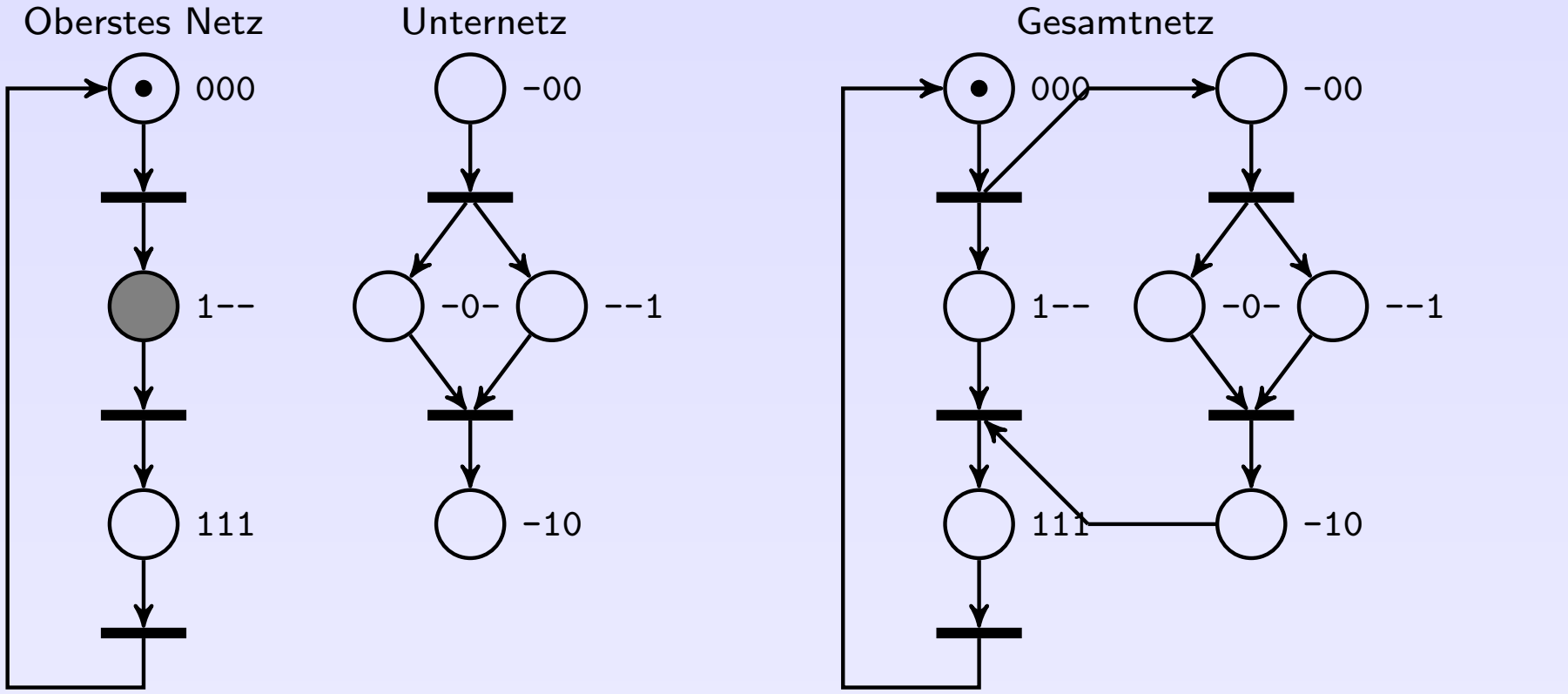
e_1	Ein-Taster	(Schließer)
e_2	Aus-Taster	(Öffner)
e_3	Temperatur	(Öffner)
a_1	Stern	
a_2	Dreieck	



Hinweis: abschaltende Signale (Ausschalter, Temperaturüberwachung) müssen wegen möglichem Drahtbruch stets als Öffner ausgelegt sein.

Hierarchien und Kopplung für SIPN

- Hierarchie = ...
+ besserer Überblick bei großen Netzen
- Teilnetze sind durch Plätze umrandet, d.h. sie beginnen und enden mit einem Platz
- Hierarchieplätze und Teilnetze dürfen Ausgaben machen
- Wegen ihrer Teilausgaben bleiben Hierarchieplätze auch im entfalteten Netz erhalten



Inhärente Eigenschaften des SIPN

Lokalität

- Alle Informationen zum Schalten einer Transition sind lokal vorhanden. Es werden also z.B. keine internen Variablen eingekoppelt.
- Alle Informationen für Gesamtaussage lokal an markierten Plätzen vorhanden. Unmarkierte Plätze oder Vergangenheit unberücksichtig. Verletzt z.B. bei speichernd gesetzten Ausgaben

Markierungs-Zustands-Äquivalenz

- Der Markierungsvektor eines SIPN entspricht dem Zustand eines EA
- Verletzt bei internen Merkern

Moore-Eigenschaft

- Gesamtausgabe hängt nur von aktueller Markierung ab, also vom aktuellen Zustand
- Verletzt z.B. bei Ausgaben als Funktion von Eingaben.

Parallelität modellierbar

- Gleichzeitige Markierung von Plätzen $\rightarrow$ gleichzeitige Teilausgaben
- Gleichzeitiges Setzen mehrerer Ausgänge $\rightarrow$ gleichzeitige (parallele) Teilprozesse

Nebenläufigkeit modellierbar

- Transitionen schalten in verschiedenen Zweigen unabhängig voneinander
- zeitlich unabhängige Teilausgaben $\rightarrow$ zeitlich unabhängige (nebenläufige) Teilprozesse

Analyse des SIPN, formale Korrektheit I

Formale Korrektheit

Ein SIPN heißt **formal korrekt**, wenn folgende vier Eigenschaften, die mit Determiniertheit zu tun haben, erfüllt sind

- Nur stabile Zyklen
- Determiniertes Schalten
- Widerspruchsfreie Ausgaben
- Vollständige Ausgaben

(Funktionale Korrektheit: SIPN stimmt funktionell mit Spezifikation überein)

Stabile Zyklen

Ein **stabiler Zyklus** ist ein geschlossener Pfad im EG für den die Konjunktion aller Schaltbedingungen für beliebige Eingaben stets False liefert.

- Durch Iteratives Schalten hängt sich bei instabilen Zyklen und bestimmten Eingaben das gesamte SPS-Programm auf, da es ständig iterativ schaltet ohne weitere Eingaben abzufragen.
- Auch ohne Iterativem Schalten (z.B. SFC) kann es instabile Zyklen geben, bei denen Markierungen und zugehörige Ausgaben nur in minimaler Zeit anstehen

Analyse des SIPN, formale Korrektheit II

Determiniertes Schalten

Ein SIPN schaltet determiniert, gdw. die Konjunktion der Schaltbedingungen aller jeweils in Konflikt stehenden Transitionen für beliebige Eingaben stets den Wert Null liefert.

- Dies gilt für Vorwärts- und Rückwärtskonflikte
- Konfliktfreie SIPN schalten stets determiniert
- Die Implementierung von indeterminierten PN führt i.d.R. zu determiniertem Code (z.B. wegen Programmierreihenfolge). Modell und Realisierung stimmen nicht überein

Widerspruchsfreie Ausgaben

Die Ausgaben eines SIPN sind widerspruchsfrei, gdw. die Gesamtausgabe für alle möglichen Belegungen niemals den Wert w enthält.

- Widerspruch (w) kommt nur vor, wenn gleichzeitig wirkende Teilausgaben für denselben Ausgang 0 und 1 fordern $\Rightarrow$ Entwurfsfehler

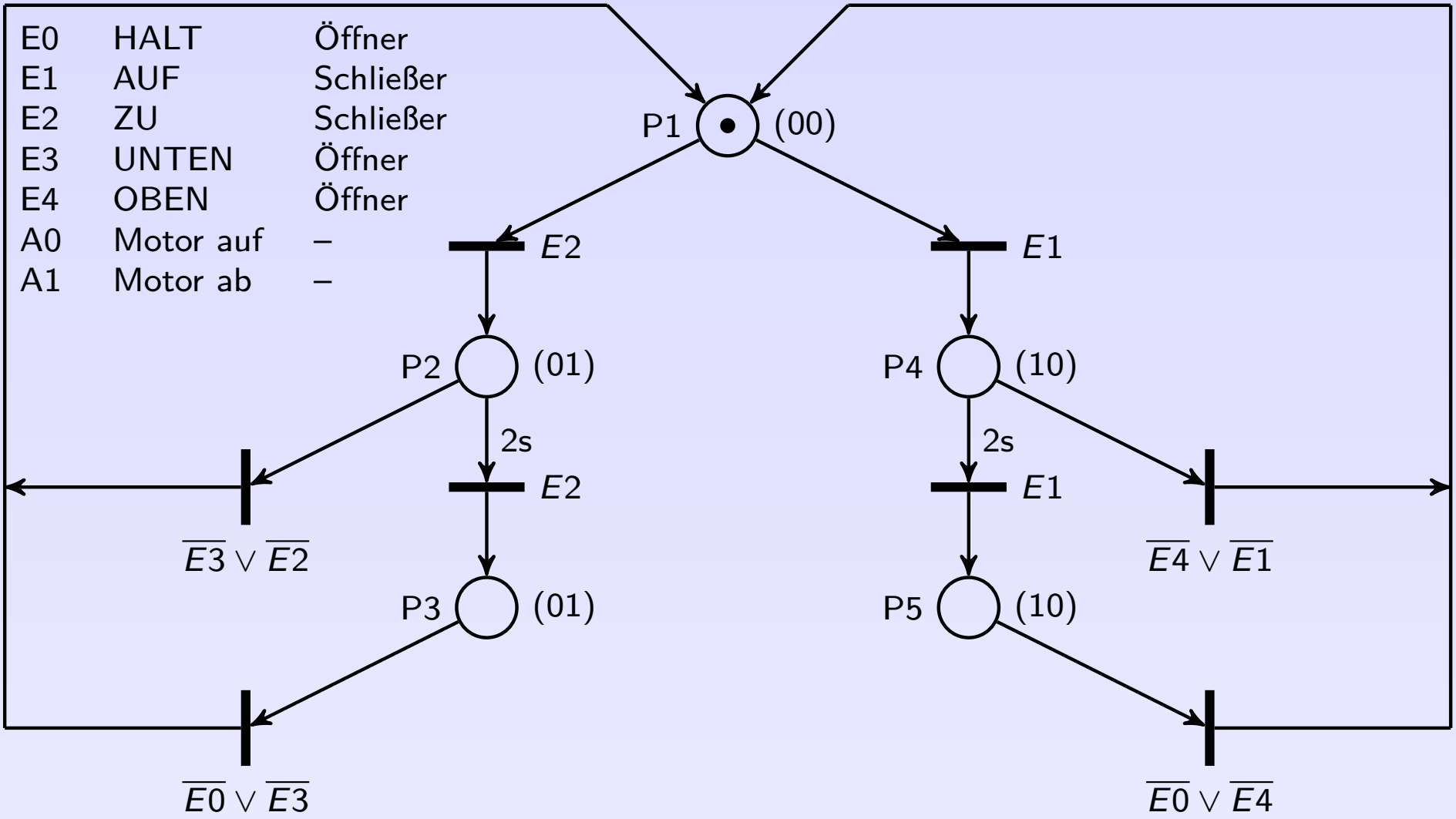
Vollständige Ausgaben

Die Ausgaben eines SIPN sind vollständig, gdw. die Gesamtausgabe für alle möglichen Belegungen niemals den Wert - enthält.

- Undefinierter Ausgang (–) kommt vor, wenn in keiner wirkenden Teilausgaben für einen Ausgang ein Wert zugewiesen wird

Analyse des SIPN, formale Korrektheit III

Beispiel: Rolltor-Steuerung



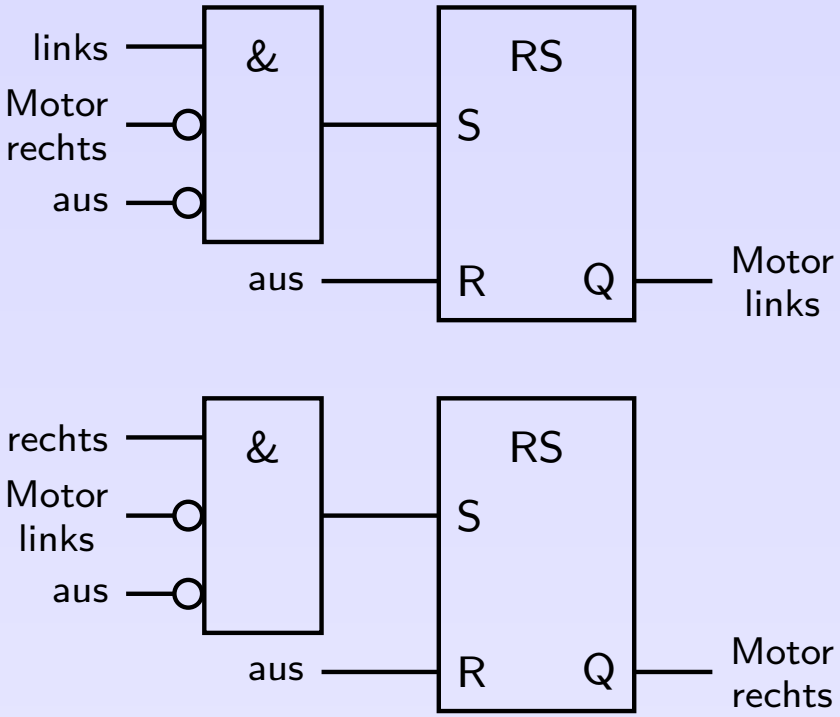
4. Modellierung mit signalinterpretierten Petrinetzen

PN, SIPN und EA

- Der EG eines PN ist ein zum PN gleichwertiger EA
- Jedes formal korrekte SIPN kann in einen äquivalenten deterministischen EA umgewandelt werden. Dieser EA ist ein Moore-Automat, der aus dem EG eindeutig hervorgeht.
- Dabei werden die Transitionen an Kanten durch Eingaben/Schaltbedingungen ersetzt.
- Die Markierungen in den Knoten des EG werden durch die entsprechenden Ausgaben ersetzt.

SIPN versus FBS/FUP

FBS



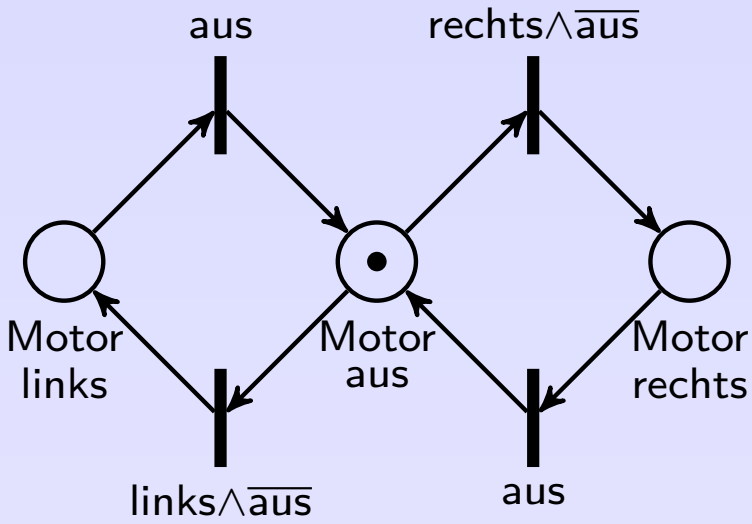
- Zustand `Motor aus`

. . . . . .

- Gegenseitige Verriegelungen zwischen `Motor links` und `Motor rechts`

. . . . . .

SIPN



- Zustand `Motor aus`

. . . . . .

- Topologie lässt Übergang von `Motor links` nach `Motor rechts` nicht zu

  Richtungswechsel nur über `Motor aus`,

  Verriegelung . . . . . .

SIPN versus SFC

Unterschiede zum SIPN

- Determinismus: Konflikte durch explizite Priorisierung oder Reihenfolge vermieden.

  Schalten trotz markierter Nachplätze erzeugt nur binäre Markierungen$\rightarrow$ . . . . . .
- Defaultwerte für undefinierte Variablen (Ausgangsgrößen)
- Keine transienten Markierungen
- Aktionen können nicht nur Ausgaben machen, sondern komplette Programme aufrufen.
- Ausgaben: speichernd S/R, nicht speichernd N, zeitverzögert D, zeitlimitiert L,

  eingabeabhängig $\rightarrow$ . . . . . .
- Nur ein Initialschritt
- Hierarchien nicht definiert
- Semantik nicht präzise definiert, unterschiedliche Tools erzeugen aus Code unterschiedliches Verhalten

Fazit

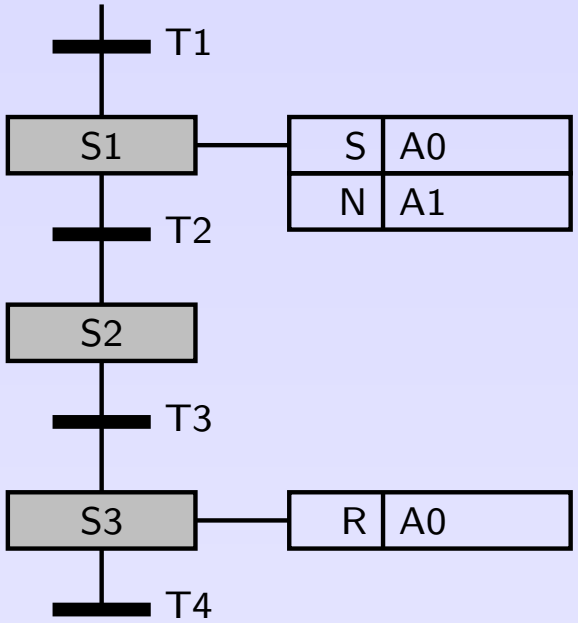
- Jedes SIPN lässt sich direkt in SFC umsetzen
- Ein SFC kann zu deutlich größeren äquivalenten SIPN führen

\+ SFC hat größere Modellierungsmöglichkeit als SIPN . . .

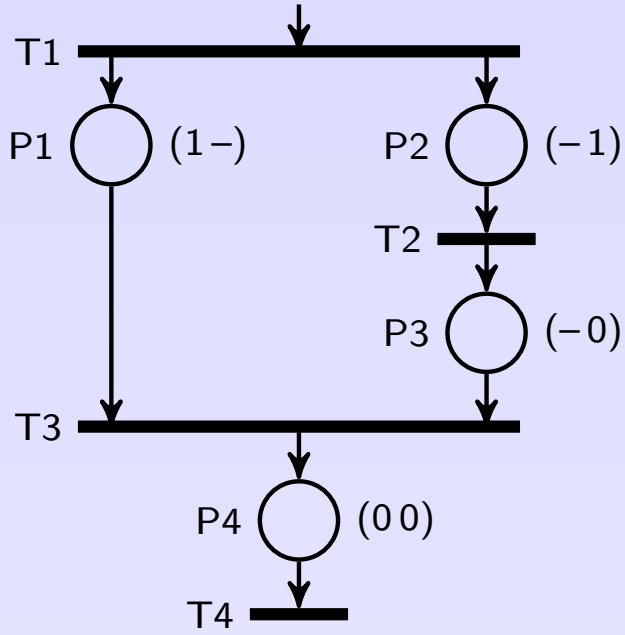
− . . . und größere Komplexität

SIPN versus SFC

SFC



SIPN



- Welche Ausgänge sind in S2 gesetzt?
- Funktion unübersichtlich wegen S-Befehl
- Problem steigt mit Abstand von S- zu R-Befehl
- Problem steigt bei zeitbewerteten Ausgabearten

- Nebenläufigkeit der Kausalketten von A0 und A1 sichtbar
- Synchronisation von A0 und A1 über T1 und T3
- Kein speicherndes Verhalten → Lokalität → Übersichtliches Verhalten

- SFC erlaubt viele Ausgabearten: speichernd, verzögernd, zeitlimitiert, Eingabe-abhängig...

⇒ dadurch kompakte Darstellung, aber unübersichtliches Verhalten

GRAFCET

- Akronym aus GRAphe Fonctionnel de Commande Etapes/Transitions
- Französischer Standard seit 1977, international genormt in IEC 60848
- Entwurfs-/Spezifikationssprache für die Darstellung von Ablaufbeschreibungen
- AS/SFC/S7-GRAPH sind mögliche Implementierung eines Grafcet-Plans
- Struktur: Schritte (Plätze), Weiterschaltbedingungen (Transitionen), Wirkverbindungen (Kanten), Kommentare
- Mindestens ein Intitialschritt (doppelt umrandet)

Unterschiede zum SIPN

- Schaltbedingungen können von Markierungen abhängen (keine Lokalitätsprinzip)
- Schaltet auch bei Konflikten/Kontakten deterministisch; schwache Konzessionsregel erzeugt hier nur binäre Markierungen (erhöht Lauffähigkeit bei Entwurfsfehlern und verschleiert diese)
- Einstellung, ob transiente Markierungen möglich sind
- Ausgaben: Not Stored, Stored, Delayed, Limited, Eingabe-abhängig (kein Lokalitätsprinzip)

\+ Letzteres erhöht Modellierungsmächtigkeit ...

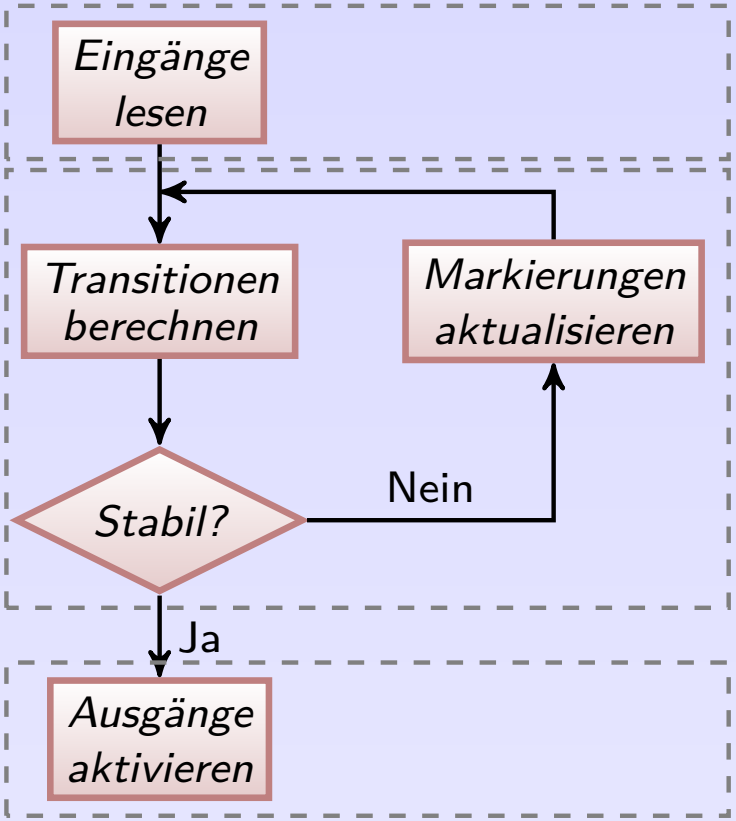
\− ... und reduziert Übersichtlichkeit und formale Analysemöglichkeiten

4. Modellierung mit signalinterpretierten Petrinetzen

Codeerzeugung aus SIPN

- SPS-Zyklus: Eingänge lesen, Algorithmus abarbeiten, Ausgänge aktivieren
- Codeerzeugung aus SIPN kann automatisiert werden
- Standardsprachen: AWL, SFC, ST
- SFC und ST kennen Transitionen: Implementierung dadurch einfacher/schneller, die entsprechenden Schaltregeln können vom SIPN abweichen
- AWL kennt keine Transitionen: Schaltregeln müssen selbst implementiert werden und stimmen dann mit den SIPN-Regeln überein

Grundstruktur zur Codeerzeugung



Verletzung der formalen Korrektheit → Abweichungen zwischen SIPN und Code:

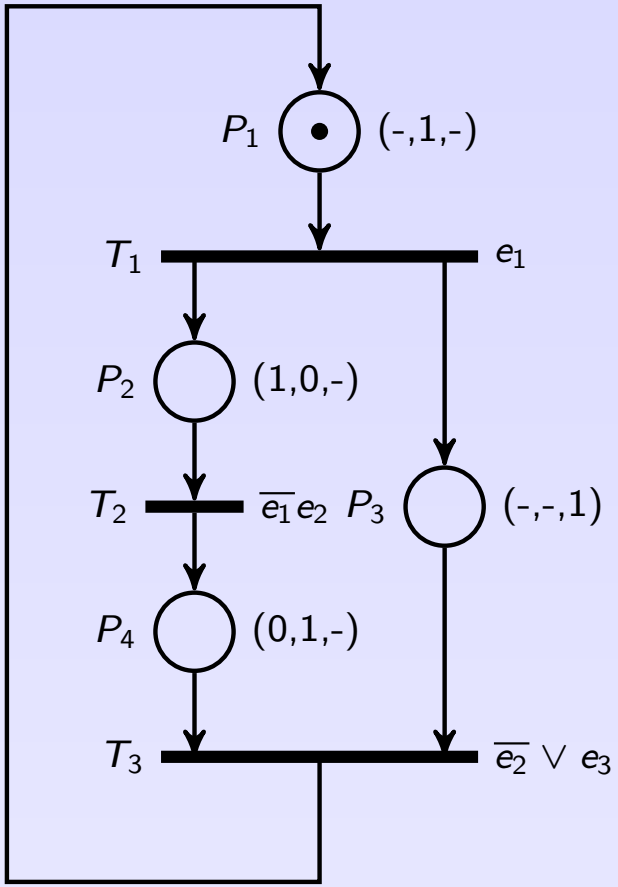
- Instabiler Zyklus → Endlosschleife → Abbruch mit Fehlermeldung
- Nichtdeterminismus im SIPN → deterministisches Verhalten im Code (abhängig von Programmierreihenfolge)
- Widersprüchlich definierte SIPN-Ausgaben → eindeutige Ausgabe im Code (abhängig von Programmierreihenfolge)
- Unvollständige SIPN-Ausgaben → eindeutige Ausgabe im Code (letzter Wert, Default-Wert)

Codeerzeugung aus SIPN $\rightarrow$ AWL

- Je Platz ein binärer Merker, der Markierung speichert
- Zusätzliche binäre Variable (eoc - end of cycle) zur Erkennung der Stabilität
- Erster Programmteil: Schalten der Transitionen, bis Markierung stabil ist
- Zeiter Programmteil: Ausgänge entsprechend der Markierung setzen
- Der folgende Programmcode verwendet die CoDeSys Befehle:

CoDeSys	Step7
JMPCN Jump Conditional on Zero	SPBN Springe bei VKE=0
AND/ANDN/OR/ORN	U/UN/O/ON
LD - Load	U
S - Set, R - Reset	S - Setze, R-Rücksetze

Codeerzeugung aus SIPN → AWL




```
PROGRAM Main
(*** Variables ***)
VAR
   P1:     BOOL := TRUE;
   P2:     BOOL := FALSE;
   P3:     BOOL := FALSE;
   P4:     BOOL := FALSE;
   eoc:    BOOL := TRUE;
END_VAR
```

```
VAR_GLOBAL
   e1    at %IX0.0: BOOL;
   e2    at %IX0.1: BOOL;
   e3    at %IX0.2: BOOL;
   a1    at %QX0.0: BOOL;
   a2    at %QX0.1: BOOL;
   a3    at %QX0.2: BOOL;
END_VAR

(*** Transitions ***)
TS: S       eoc
T1: LD      P1
   ANDN     P2
   ANDN     P3
   AND      e1
   JMPCN    T2
   R        P1
   S        P2
   S        P3
   R        eoc
T2: LD      P2
   ANDN     P4
   ANDN     e1
   AND      e2
   JMPCN    T3
   R        P2
   S        P4
   R        eoc
```

```
T3: LD      P4
   AND      P3
   ANDN     P1
   AND(
   ORN      e2
   OR       e3
   )
   JMPCN    TE
   R        P4
   R        P3
   S        P1
   R        eoc
TE: LD      eoc
   JMPCN    TS
(*** Output ***)
Q1: LD      P1
   JMPCN    Q2
   S        a2
Q2: LD      P2
   JMPCN    Q3
   S        a1
   R        a2
Q3: LD      P3
   JMPCN    Q4
   S        a3
Q4: LD      P4
   JMPCN    Q5
   R        a1
   S        a2
Q5: RET
END_PROGRAM
```

Codeerzeugung aus SIPN $\rightarrow$ ST/SCL

Codesys und IE 61131-3

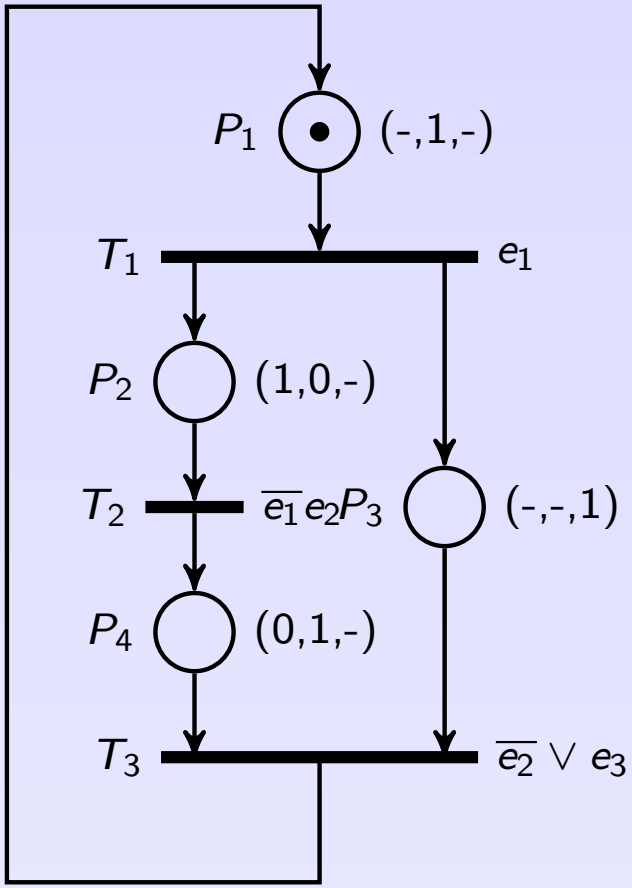
In CoDeSys sind folgende Befehle definiert (in Step7 leider nicht)

- TRANSITION-Anweisung für Transitionen
- STEP-Anweisung für Plätze
- genau eine INITIAL_STEP-Anweisung als Anfangsschritt
- Kein iteratives Schalten: bei instabilen Markierungen erfolgt kurzzeitige Ausgabe

Step 7

- Für Zustandsmaschine: SWITCH-CASE
- Sonst: IF-THEN-ELSE
- Iteratives Schalten mit REPEAT-UNTIL und eoc-Variable

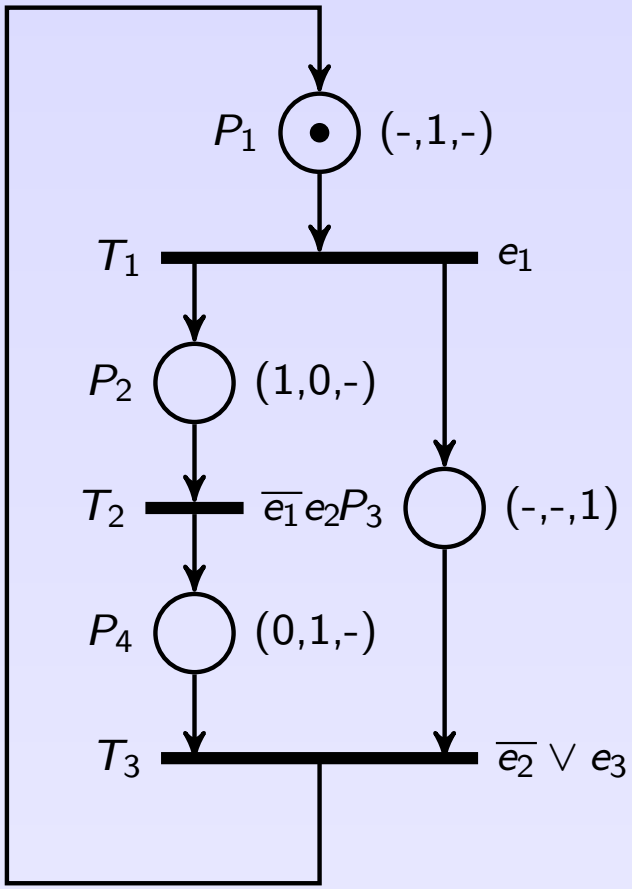
Codeerzeugung aus SIPN → ST/SCL (CoDeSys)




```
TRANSITION FROM P1 TO (P2,P3)
    := E1;
END_TRANSITION;
TRANSITION FROM P2 TO P4
    := NOT E1 AND E2;
END_TRANSITION;
TRANSITION FROM (P4,P3) TO P1)
    := NOT E2 OR E3;
END_TRANSITION;

INITIAL_STEP P1:
     A2 := 1;
END_INITIAL_STEP;
STEP P2:
     A1 := 1;
     A2 := 0;
END_STEP;
STEP P3:
     A3 := 1;
END_STEP;
STEP P4:
     A1 := 0;
     A2 := 1;
END_STEP;
```

Codeerzeugung aus SIPN → ST/SCL (Step7)




```
REPEAT          // Iteratives Schalten
  eoc:=true;
  IF (P1 & NOT P2 & NOT P3 & e1) THEN
    P1:=false; P2:=true; P3:=true; eoc:=false;
  END_IF;
  IF (P2 & NOT P4 & not e1 & e2) THEN
    P2:=false; P4:=true; eoc:=false;
  END_IF;
  IF (P3 & P4 & NOT P1 & (not e2 OR e3)) THEN
    P3:=false; P4:=false; P1:=true; eoc:=false;
  END_IF;
  UNTIL (eoc=true) // Abbruchbedingung
END_REPEAT;

IF (P1) THEN
    a2:=true;
END_IF;
IF (P2) THEN
    a1:=true; a2:=false;
END_IF;
IF (P3) THEN
    a3:=true;
END_IF;
IF (P4) THEN
    a1:=false; a2:=true;
END_IF;
```

Literatur, Vertiefung - Modellierung mit signalinterpretierten Petrinetzen

WIKIPEDIA Petri-Netz
de.wikipedia.org/wiki/Petri-Netz

G. Frey, Integration von Petrinetzen in den Steuerungsentwurf nach IEC 61131
https://www2.informatik.uni-erlangen.de/publication/download/SPS-Drives2001.pdf

5. Netzwerktechnik

Wichtige Begriffe

Unicast	Genau ein Empfänger
Multicast	Ausgewählte Gruppe von Empfängern
Broadcast	Empfänger sind alle Stationen des Netzes
Simplex	Daten nur in eine Richtung übertragen
Halbduplex	Daten abwechselnd in beide Richtungen
Vollduplex	Daten gleichzeitig in beide Richtungen
LAN	Local Area Network
WAN	Wide Area Network

Server Programm, das Dienst/Funktion/Service für ein oder mehrere Clients anbietet
Server in Bereitschaft, um jederzeit auf Client reagieren zu können

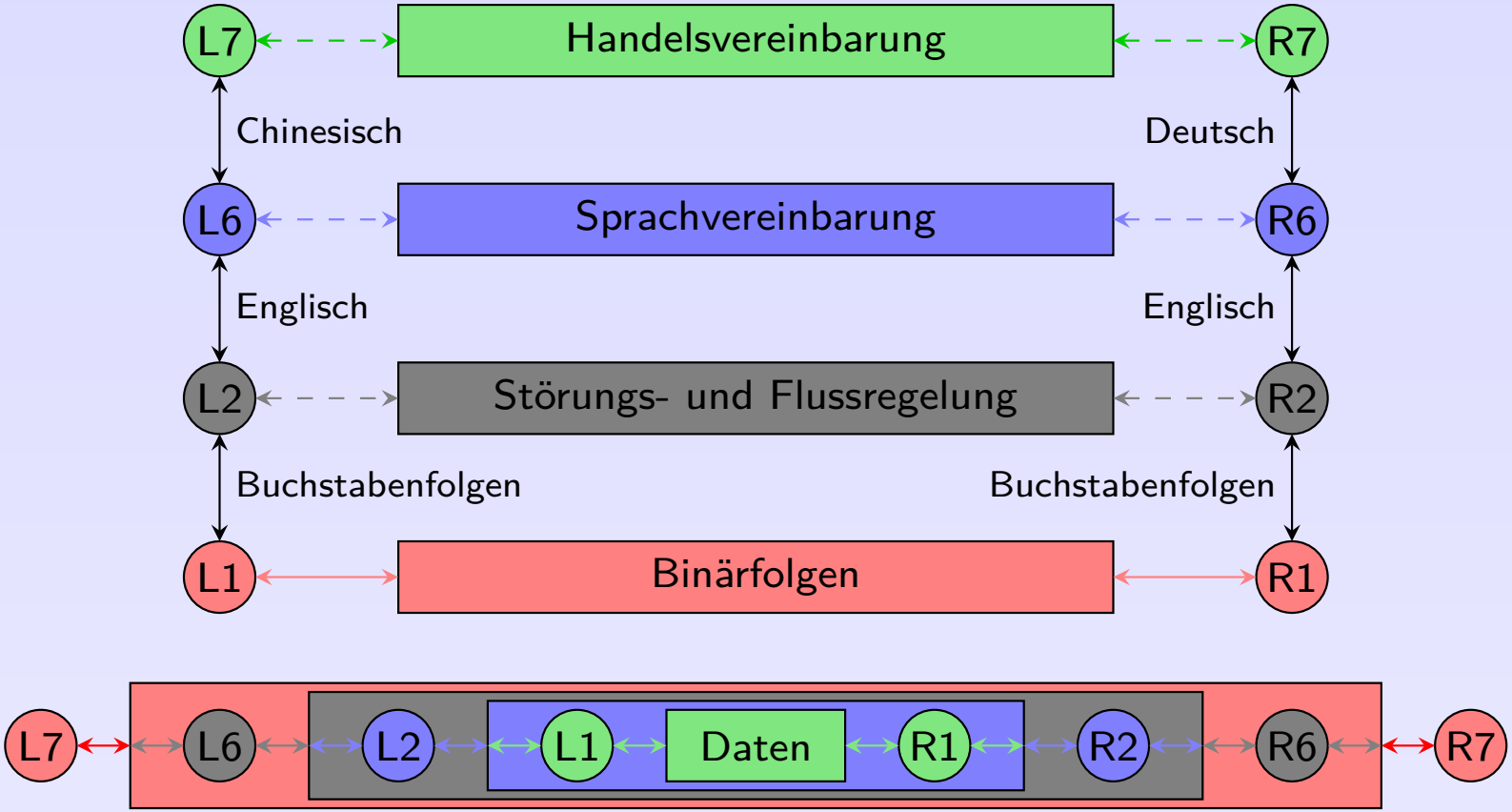
Client Programm, das diesen Dienst nutzen kann,
Client fordert aktiv Dienst an.

ISO-OSI Schichtenmodell

- Schichtenmodell (ISO 7458) für die Entwicklung offener Kommunikationsprotokolle
- ISO = International Standard Organisation, OSI = Open System Interconnection
- Regelt Kommunikation zwischen Sender und Empfänger
- Untere Schichten (1-4): Transportfunktionen, Datenübertragung zwischen Endgeräten
- Obere Schichten (5-7): Anwenderfunktionen, Zusammenwirken zwischen Anwenderprogramm und Betriebssystem
- Schichteneinteilung aufgabenorientiert $\Rightarrow$ Abstraktion des Kommunikationsprozesses
- keine Angaben über konkrete Implementierung
- Jede Schicht stellt bestimmte Dienste für darüber liegende Schicht bereit
- Jede Schicht austauschbar, ohne dass andere Schichten davon betroffen sind
- Handlungen verlaufen bei Sender und Empfänger meist umgekehrt. Z.B. Schicht 1 kodiert die Information beim Sender in Bitfolgen, Empfänger dekodiert
- Nicht alle Schichten immer realisiert

ISO-OSI Schichtenmodell - Logische und Reale Kommunikation

Beispiel: Chinesischer und deutscher Ingenieur wollen kommunizieren. Kommunikation verläuft indirekt über Dienste. Der jeweilige Dienstnutzer merkt davon nichts.



Schichtenmodell

5. Netzwerktechnik

Schicht 1 - Physikalische Schicht

Physikalische Schicht beinhaltet Definition physikalischer Parameter

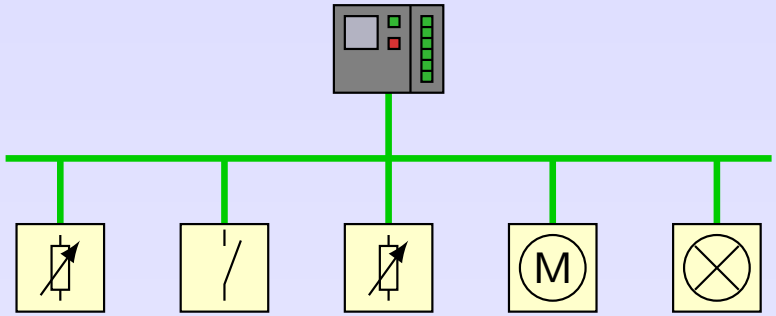
- Übertragungsmedium: .....
- Anschlüsse: .....
- Strecken: .....
- Zugriffsverfahren: .....
- Elektrische Daten: .....

Zugriffsverfahren I

Zugriffsverfahren regeln das Senderecht auf der Busleitung

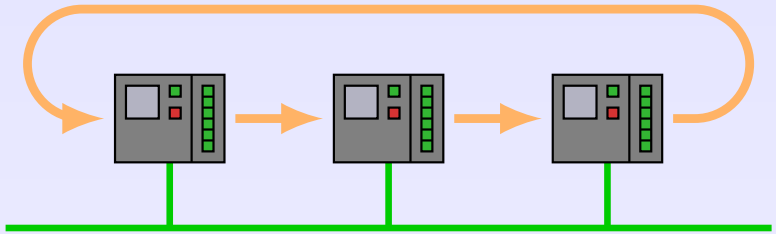
Master/Slave

- Zentrale Bussteuereinheit (Master) stellt jeweils Verbindung zum passiven Teilnehmer (Slave) her.
- Slave antwortet sofort auf Anfrage des Masters
- \+ Einfaches Verfahren, garantierte Reaktionszeit
- – Kompliziert für Datenaustausch zwischen Slaves
- Polling = zyklische Abfrage der Slaves durch Master
- Bsp: Profibus DP (Dezentrale Peripherie), ASi-Bus



Token-Passing

- Alle Teilnehmer können Kommunikationssteuerung übernehmen
- Senderecht durch spezielle Nachrichten (Token) koordiniert
- Nur Teilnehmer mit Tokenbesitz darf senden. Danach wird Token weitergereicht.
- \+ Multimasterfähig, garantierte Reaktionszeit

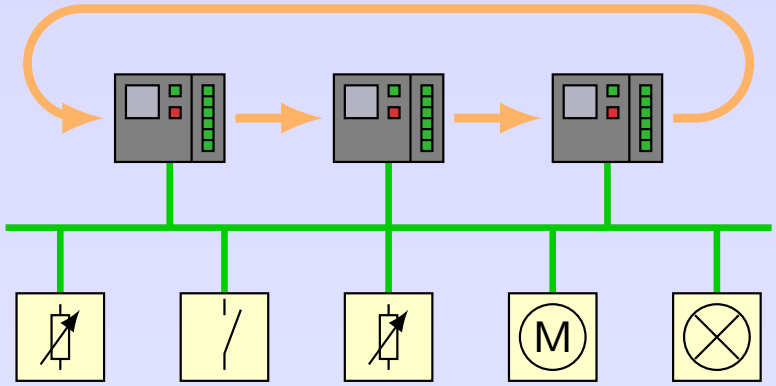


Zugriffsverfahren II

Hybride Verfahren

- Logischer Tokenring zwischen Mastern
- Abfrage passiver Teilnehmer durch Master/Slave

\+ Multimasterfähig, garantierte Reaktionszeit



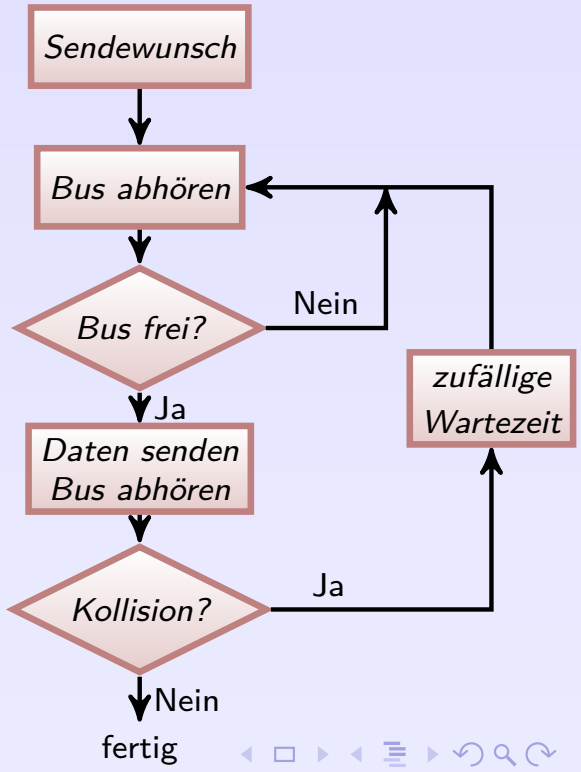
CSMA/CD - Carrier Sense Multiple Access with Collision Detection

- Teilnehmer hört, ob Bus frei ist (Carrier Sense)
- Falls Bus besetzt, erneuter Versuch nach Zufallszeitintervall (Multiple Access).
- Zufälliges, gleichzeites Senden mehrerer Teilnehmer $\hat{=}$ Kollision (Zerstörung der Nachricht)

  wird erkannt (Collision Detection), erneutes Senden

– Keine maximale Reaktionszeit garantiert, nicht echtzeitfähig

- Anwendung in Bürokommunikation (z.B. Ethernet)

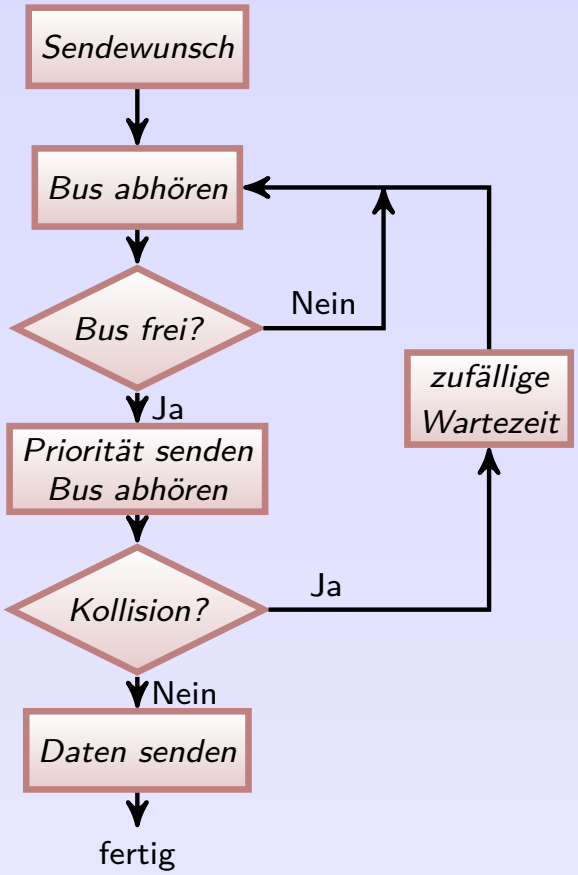


Zugriffsverfahren III

CSMA/CA - Carrier Sense Multiple Access with Collision Avoidance

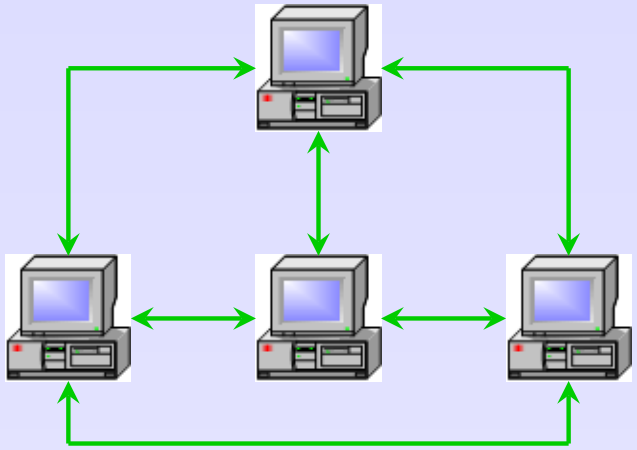
- Steigerung der Effizienz des CSMA durch Vermeidung von Datenkollision (Collision Avoidance).
- Teilnehmer sendet zuerst seine Priorität, anschließend Nutzdaten
- Sendung wird ständig überwacht, im Konfliktfall zieht sich Teilnehmer mit niedrigerer Priorität zurück.
- Beispiel:
  Bus ist nulldominant (Nullsignal setzt sich durch)

Priorität	Signal	
5	0101	sieht keinen Konflikt, sendet weiter
7	0111	sieht Konflikt, zieht sich zurück
Bus	0101	



Topologien I

Punkt-zu-Punkt



- Jeder Teilnehmer mit jedem anderen direkt verbunden
- Engmaschiges Netz
- n Teilnehmer $\rightarrow (n-1)$ Schnittstellen pro Teilnehmer, $\frac{1}{2}(n)(n-1) \approx \frac{1}{2}n^2$ Leitungen

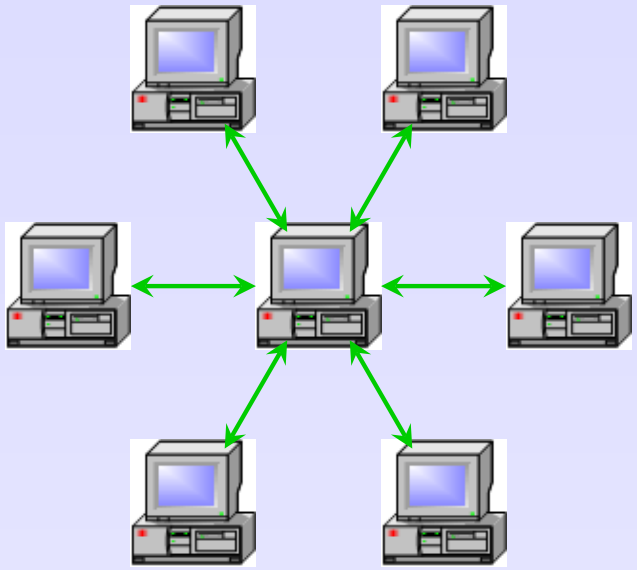
– Hohe Kosten für Verkabelungen

\+ Leichte Diagnose im Fehlerfall

\+ Beschränkte Auswirkung bei Ausfall eines Teilnehmers oder einer Leitung

Topologien II

Sternstruktur



- Alle Komponenten mit zentraler Vermittlungsstelle verbunden
- Verkabelungsaufwand niedriger als Punkt-zu-Punkt-Struktur, aber höher als Linie

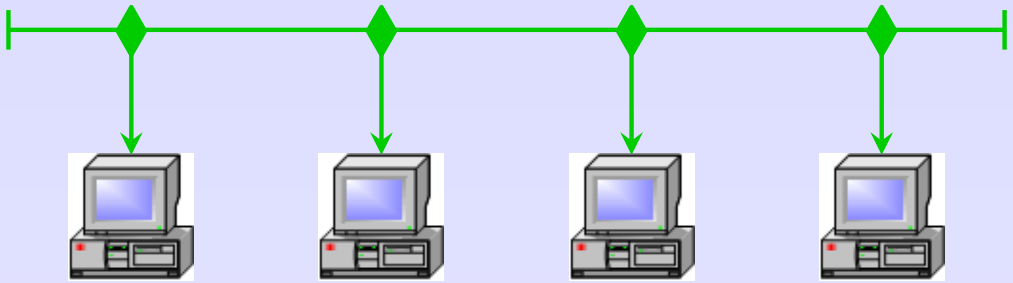
\+ Beschränkte Auswirkung bei Ausfall eines Teilnehmers

– Ausfall der Zentrale gleich Totalausfall

- Beispiel: konventionelle Verdrahtung von SPS mit Sensorik und Aktorik

Topologien III

Linienstruktur (mit Stichleitungen)



- Alle Komponenten an gemeinsame Leitung (dem Bus) angeschlossen
- Endpunkte werden häufig mit Widerstand abgeschlossen (Reflektion von Wellen)
- Anbindung der Teilnehmer an Buskabel geschieht oft über kurze Stichleitungen
- Jeder Teilnehmer kann über den Bus mit jedem anderen Teilnehmer direkt kommunizieren
- Kein Master notwendig, aber möglich

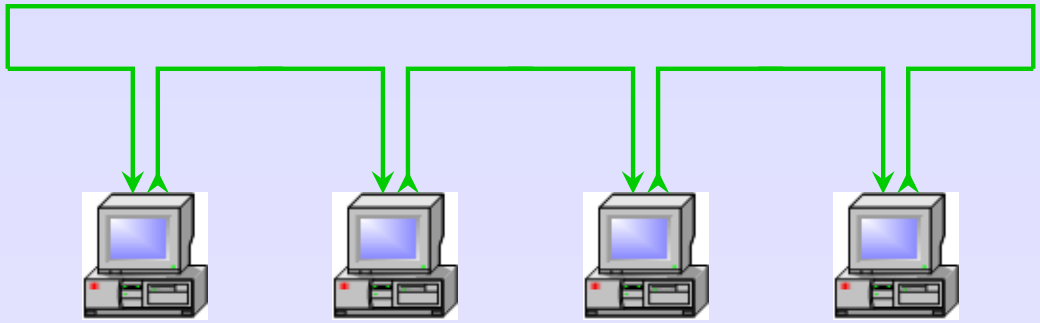
– Ausfall des zentralen Mediums gleich Totalausfall

\+ Leichte Erweiterbarkeit, geringer Verkabelungsaufwand

\+ Multimasterfähig

Topologien IV

Ringstruktur



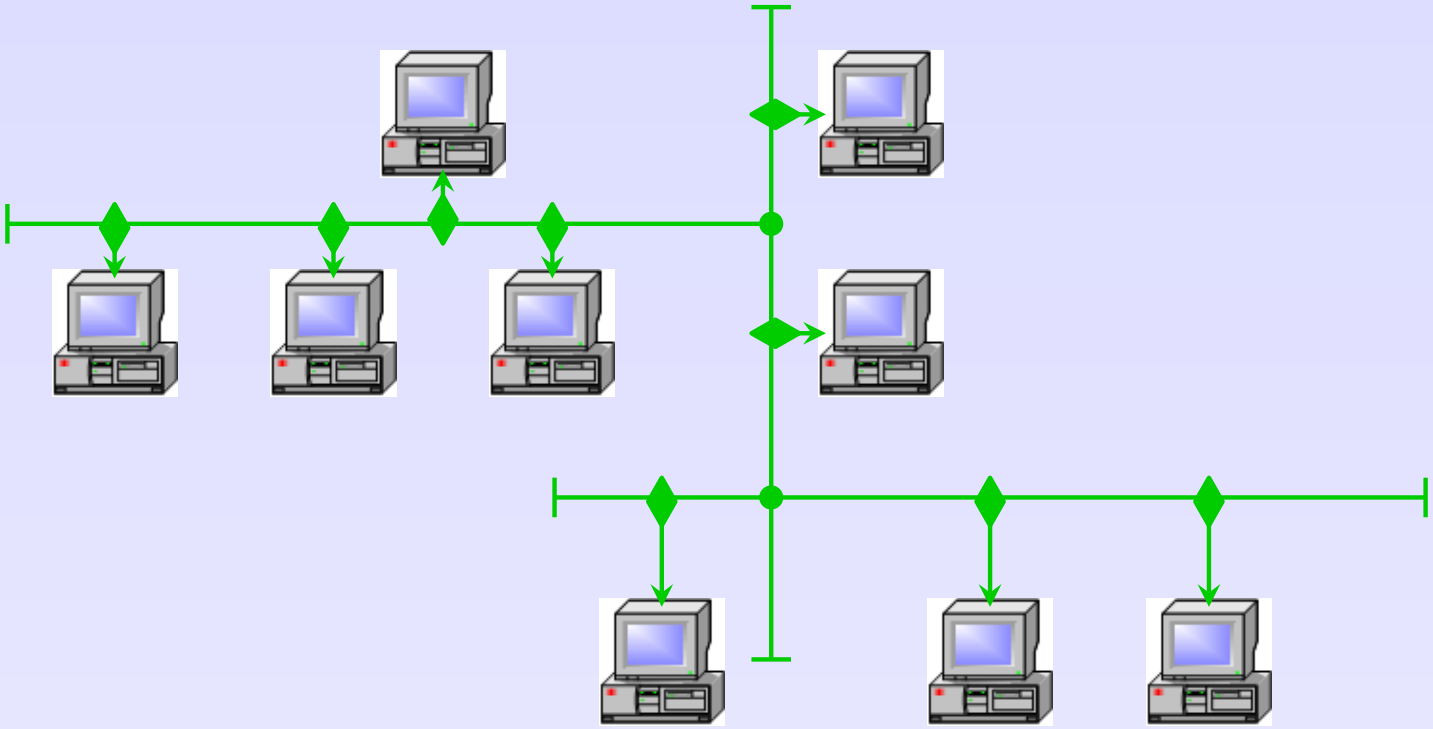
- Jeder Teilnehmer hat genau zwei Nachbarn, fester Umlaufsinn
- Kein Master notwendig

\+ geringer Verkabelungsaufwand, relativ große Reichweite, leichte Erweiterbarkeit

– Ausfall der Ringleitung oder eines Teilnehmers gleich Totalausfall

Topologien V

Baumstruktur



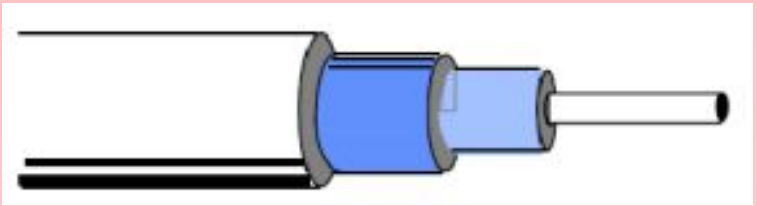
- Weiterentwicklung der Linienstruktur, kann größere Flächen abdecken
- Oft Verstärkerelemente (Repeater) zur Signalverstärkung an Baumzweigen

Übertragungsmedien - Twisted Pair

- Kupferkabel mit verdrillten Adernpaaren, meist PVC-Mantel
- Paarweises Verdrillen reduziert störende Einflüsse von äußeren magnetischen Wechselfeldern durch andere stromführende Kabel so wie Übersprechen zwischen benachbarten Adernpaaren innerhalb des Kabels
- Schirm: Aluminiumfolie, beschichtete Kunststofffolie oder Drahtgeflecht
- Manchmal mit antistatischer Kunststofffolie
- Verschiedene Standards (US, ISO und EN: Cat. oder Class)
- Bezeichnungssystem nach ISO/IEX-11801 (2002) E der Form G/AV
  G Gesamtschirm, A Adernpaarschirm, V Verdrillung

G	A	V	
U	U		Unshielded, Ungeschirmt
F	F		Foiled, Folienschirm
S	S		Shielded, Geflechtschirm
SF			Geflecht- und Folienschirm
		TP	Twisted Pair (Standard)
		QP	Quad Pair

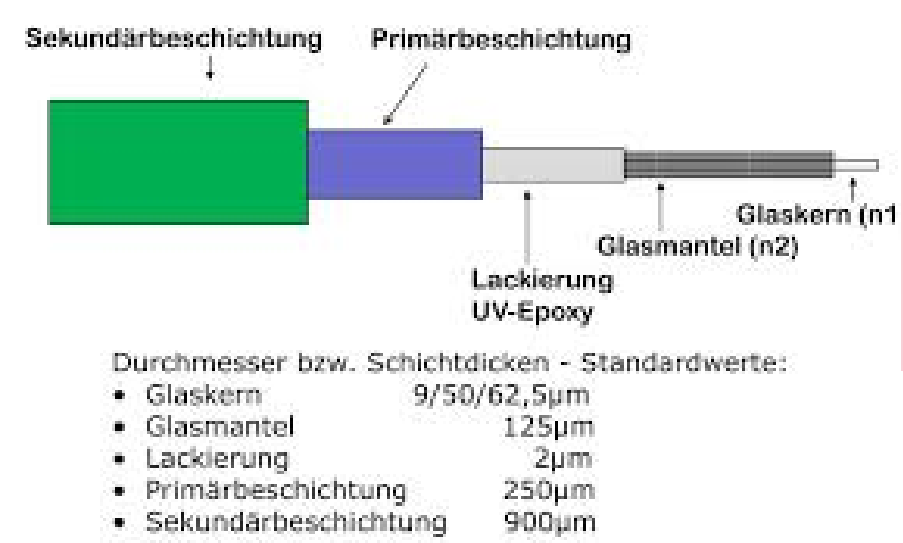
Übertragungsmedien - Koaxialkabel

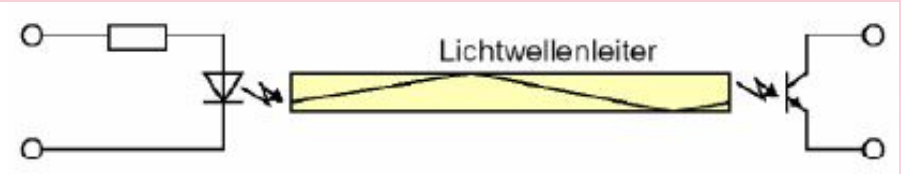


Mantel Außenleiter Dielektrikum Seele

- Zweipolige Kabel mit konzentrischem Aufbau
- Innenleiter massiv oder flexibel
- Definierter Wellenwiderstand beträgt für TV/Radiotechnik üblicherweise 75Ω, für andere Anwendungen meist 50Ω
- Bei Änderung des Wellenwiderstandes (ungeeignete Verbindungsstellen, falscher Abschlusswiderstand, falscher Biegeradius) entstehen bei höheren Frequenzen Reflexionen (Echo)
- Reflexion kann Signal durch Überlagerung vollständig zerstören
- Wellenwiderstand unabhängig von Leitungslänge, jedoch leicht frequenzabhängig
- Kapazitätsbelag bei 50Ω-Koaxialkabel etwa 100 pF/m

Übertragungsmedien - Lichtwellenleiter I



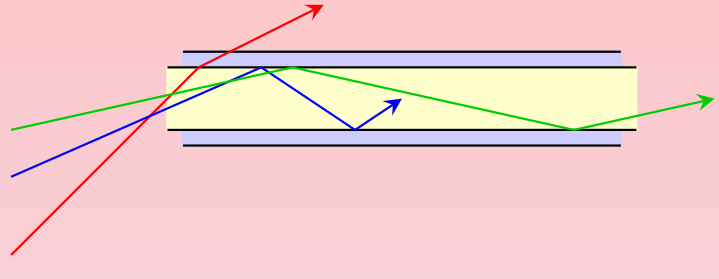


- weite Strecken, bis 100km ohne Verstärker
- Standardmäßig drei Typen: Multimode-Fasern mit 62,5 µm (US-Norm) und 50 µm (EU-Norm) und Singlemode-Fasern mit 9 µm Kerndurchmesser
- Aufbau z.B. Kern, Führungshülle, Gelmantel (erlaubt Kernbewegung, reduziert maximalen Biegeradius), Schutzmantel
- Manchmal integrierte Kupferadern zur Hilfsenergieversorgung
- Kern leitet Licht, z.B. Silizium-Germanium-Oxid oder Kunststoffe
- Da Kern sehr dünn, bricht er beim Biegen nicht
- Kern sehr zugempfindlich, daher Zugentlastung durch Hülle

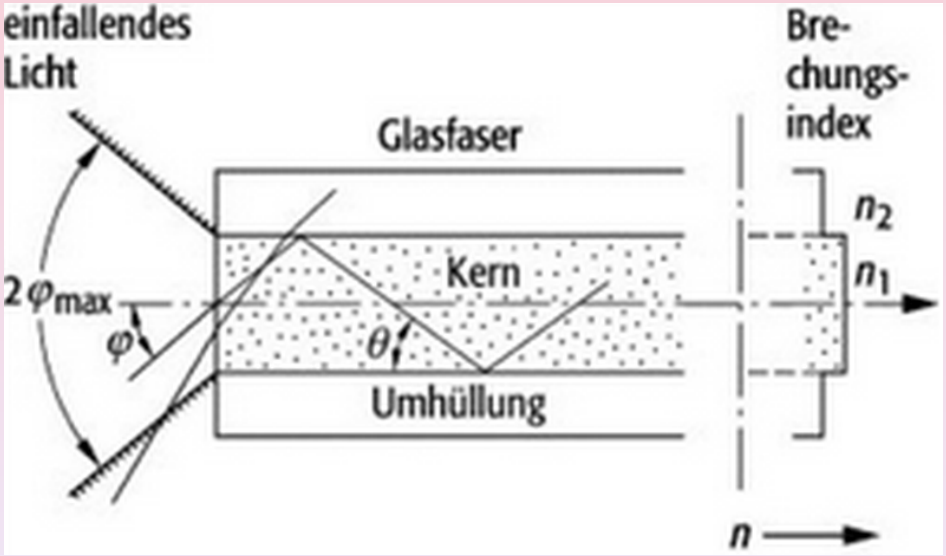
Übertragungsmedien - Lichtwellenleiter II

- Lichtstrahl bewegt sich durch Totalreflexion an Grenzflächen (verschiedener Brechungsindex) durch Lichtleiter

- Lichtstrahl muss innerhalb Aperturkegel in Faser eingekoppelt werden, sonst Brechung statt Reflexion



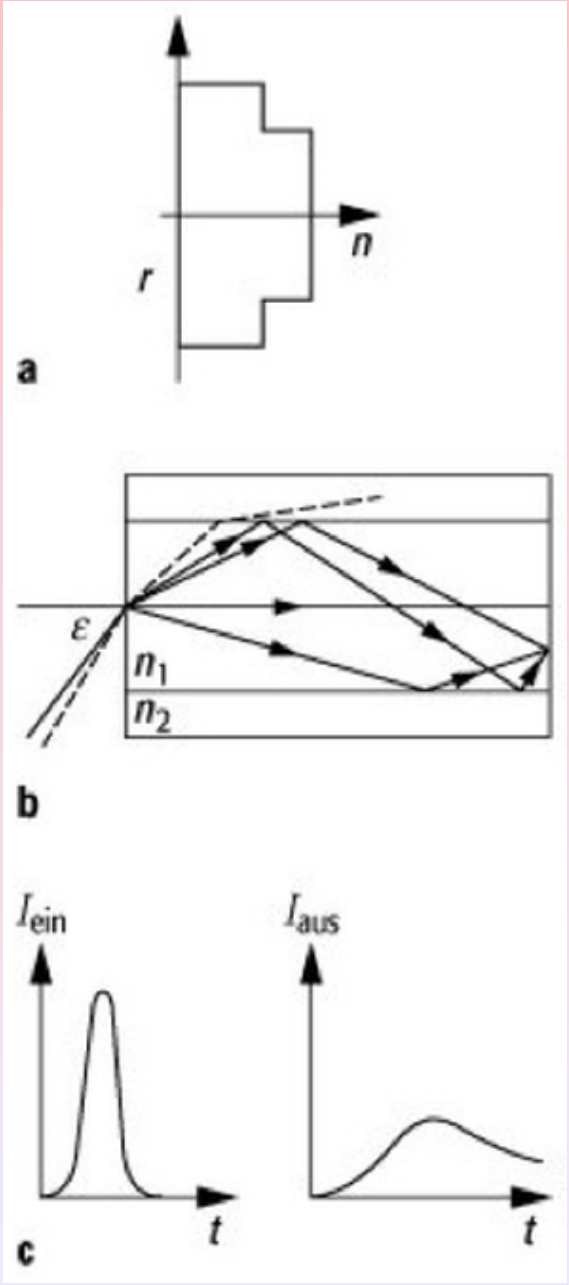
- Brechungsindex $n = \frac{\vec{v}_{\text{Vakuum}}}{\vec{v}_{\text{Medium}}} > 1$



spektrum.de/lexikon/physik/glasfaser/5912

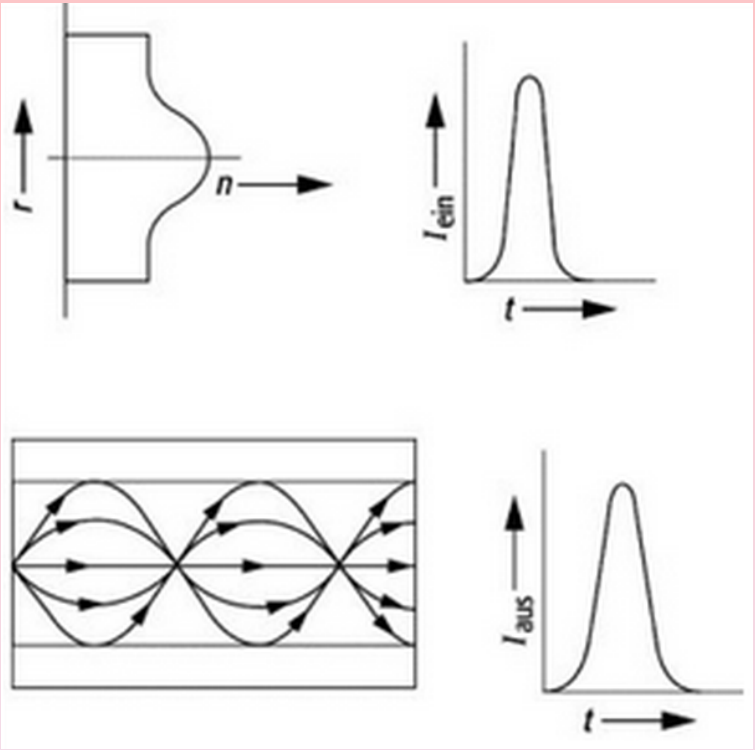
Übertragungsmedien - Lichtwellenleiter III

- Modendispersion: Lichtquelle emittiert i.d.R. verschiedene Wellenlängen, die zu unterschiedlichen Einkoppelwinkeln dieser Eigenwellen (Moden) führen. Dadurch unterschiedlich lange Wege im Medium und Signalverfälschung $\rightarrow$ Längenbeschränkung von Kabeln
- Stufenindexfaser: Faserkern aus zwei Schichten mit unterschiedlichem, aber jeweils konstantem Brechungsungsindex, d.h. Brechungsindex wird in einer Stufe im Kern verändert
- a) Querschnitt
  b) Lichtausbreitung
  c) Pulsverformung in Stufenindexfaser
  spektrum.de/lexikon/physik/stufenindexfaser/14125



Übertragungsmedien - Lichtwellenleiter IV

- Gradientenindexfaser: zur Minimierung von Laufzeitunterschieden der Moden wird Faserkern durch Dotierung verändert, so dass ein Gradient des Brechungsindex entsteht. Dadurch bewegen sich Moden im Zentrum (kürzerer Weg) langsamer als im Außenbereich (längerer Weg)
- a) Querschnitt
  b) Lichtausbreitung
  c) Pulsverformung in Gradientenindexfaser
- Bei Singlemode-Fasern keine Modendispersion, daher längere Strecken möglich - aber teuer und schwerer zu verlegen



spektrum.de/lexikon/physik/gradientenindexfaser/6049

Übertragungsmedien - Lichtwellenleiter V

- Verluste: durch Materie der Glasfaser, Streuung bei Inhomogenität der Faser, Feuchtigkeit, starke Krümmung, Staub auf Enden

\+ robust gegen elektromagnetische Störungen, selbst keine Störstrahlung (abhörsicher)

\+ hohe Übertragungsrate

\+ galvanische Trennung (Überspannung durch z.B. Blitzschlag wird nicht weitergeleitet)

– teuer, mechanisch empfindlich (Knicke, Biegen), Installation aufwendig, keine Hilfsenergieübertragung

! Achtung: nie dem (z.T. nicht sichtbaren) Laser exponieren, sonst Verbrennung und Erblindung

Standard-Schnittstellen

- 0-20mA oder 4-20 mA (live-Zero), DIN IEC 60381-1
- 0-10 V oder 2-10 V (live-Zero), DIN IEC 60381-2
- Stromsignal gegenüber Spannungssignal bevorzugt, da unempfindlich gegen elektromagnetische Störungen (Einschalten von benachbarten Verbrauchern) und Spannungsverlusten durch Leitungswiderstand

\+ Standardisierung $\rightarrow$ Herstellerunabhängigkeit, Kompatibilität

\+ Preiswert, einfach

\+ Bei live-Zero Drahtbruchüberwachung

\+ Bei 4-20 mA Versorgung des Sensors mit Hilfsenergie möglich

– Bei analogen Signalen geringe Auflösung wegen kleinem Skalenbereich (vgl. digitale Signale $\geq$ 8Bit)

Repeater-Hub

Netgear Ethernet 4-Port Hub EN104TP

- Kurzbezeichnung Hub (Knotenpunkt)
- Verbindet Teilnehmer/Segmente auf unterster OSI-Schicht
- Keine Intelligenz
- Nur Anschlüsse/Ports mit gleicher Geschwindigkeit
- Reine Kopplung und Signalverstärkung
- Daten werden stets an alle Teilnehmer/Ports weitergeleitet
- Heute veraltet, Funktion in Geräten höherer Schichten integriert

5. Netzwerktechnik

Schicht 2 - Sicherungschicht

Sicherstellen funktionierender Verbindungen zwischen . . . . . direkt benachbarten Stationen

= Data Link Layer, Datenverbindungsschicht

- Hardware-Adressen: . . . . . .
- Datensicherung: . . . . . .

Adressen

- Jedes Endgerät in Netzwerksegment sieht alle Datenpakete und muss entscheiden, ob es Empfänger ist

→ Adressierungssystem

	Physikalische Adresse	**Logische Adresse**
ISO-OSI-Schicht	Schicht 2	Schicht 3
Kommunikation	lokal,	weltweit,
	innerhalb LAN-Segment	zwischen LAN-Segmenten
Verwendung	von Hardware	von Software
Beispiele	DE:Koblenz:K-Zuse-1:Schnick	HS-Koblenz:IW:Dekan
	MAC-Adresse	IP-Adresse, Windows-Name

Physikalische Adresse

= MAC-Adresse (Medium Access Control)

- Jede Netzwerkkarte hat weltweit einmalige, feste physikalische Adresse
- Vom Hersteller in Hardware fest eincodiert
- Sechs Byte lang, wird hexadezimal geschrieben
- Erste drei Byte = Herstellercode, z.B.

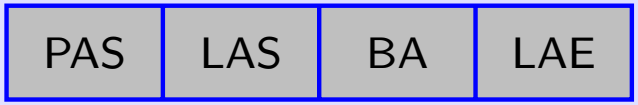
00:02:B3:XX:XX:XX	Intel
00:60:52:XX:XX:XX	Realtek
00:C0:4F:XX:XX:XX	Dell
00:A0:40:XX:XX:XX	Apple

- Letzte drei Byte = laufende Seriennummer
- Rundsendung auf Schicht 2 an alle Netzteilnehmer (Broadcast) wird an Adresse FF:FF:FF:FF:FF:FF adressiert

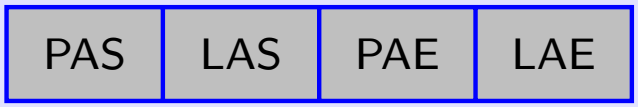
Beispiel: MAC-Adresse und Hersteller der Netzwerkkarte ermitteln
Windows: Start → „cmd“→ „ipconfig /all“→ Physikalische Adresse
Linux: Konsole → „ifconfig“
Erste drei Byte eingeben z.B. bei heise.de/netze/tools/mac/

Adressermittlung mit ARP - Adress Resolution Protocol

- Sender S muss physikalische Adresse des Empfängers (PAE) kennen/ermitteln
- Ohne PAE weiß E nicht, dass Daten für ihn bestimmt sind, und die anderen Teilnehmer wissen nicht, dass sie die Daten ignorieren sollen
- Zunächst nur logische Adresse (LAE) von E bekannt
- S sendet an alle Teilnehmer im Segment über Broadcast-Adresse (BA) ein ARP-Request



- E merkt sich Daten von S und antwortet mit ARP-Reply



- Adresstabelle wird im ARP-Cache gespeichert; Einträge verfallen i.d.R nach 300 s
- Geräte bleiben so erreichbar trotz Änderung der PA oder LA
- Zusätzlich können andere Teilnehmer die LAS-PAS-Kombinationen lernen

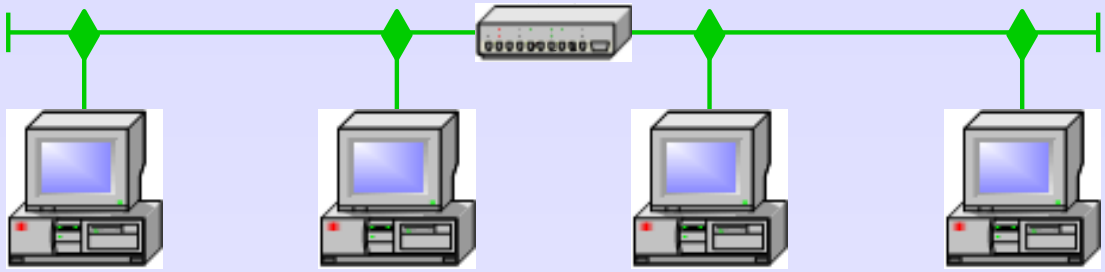
Beispiel: ARP-Cache unter Windows → „arp -a“

(Weitere Felder für Adresstypen und Protokollgrößen im ARP-Paket, siehe : `Plate, Jürgen: Computernetze`)

Bridge

- Viele Teilnehmer → viele Datenpakete → hohe Netzauslastung/viele Kollisionen (CSMA)
- Teilnehmer müssen stets prüfen, ob Pakete für Sie bestimmt sind

⇒ Trennung in Segmente, die durch Bridge verbunden sind



- Bridge: meist zwei Anschlüsse und trennt Medium in separate Segmente
- Überwacht ARP-Pakete und lernt, welche MAC-Adresse an welcher Seite hängt
- Datenverkehr innerhalb eines Segments gelangt nicht ins andere Segment
- Broadcasts (z.B. ARP-Request) werden weitergeleitet
- Kann Signale auch verstärken
- MAC-Bridge für Netze mit gleichen Zugriffsverfahren und Ü-Rate
- LLC-Bridge (Logical Link Control) für Netze mit unterschiedlichen Zugriffsverfahren
  Daten ins andere Segment werden zwischengespeichert falls:
  - Medium besetzt oder
  - unterschiedliche Übertragungsrate

Switch

Netgear 16-Port Gigabit Switch GS116

- Früher: Switch = Multiport-Bridge
- Multiport-Gerät, hat auf jedem Anschlussport eigene Bridge vorgeschaltet
- Intern Hochleistungsbus oder mehrere über Bridges gekoppelte Busse
- Kann mehrere Pakete zeitgleich zwischen verschiedenen Portpaaren übertragen
- Switch arbeitet stets auf MAC-Basis, kann unterschiedliche Architekturen (Ethernet – Token Ring) nicht überbrücken
- Netzwerk mit zentralem Switch bildet Stern-Topologie

\+ In voll geswitchter Umgebung gibt es keine Kollisionen mehr, Broadcast-Anfragen passieren jedoch noch alle Bridges und Switches

\+ Steigerung der Abhörsicherheit: Nur noch Empfänger bekommt das Datenpaket

Keine Kollision → Duplex

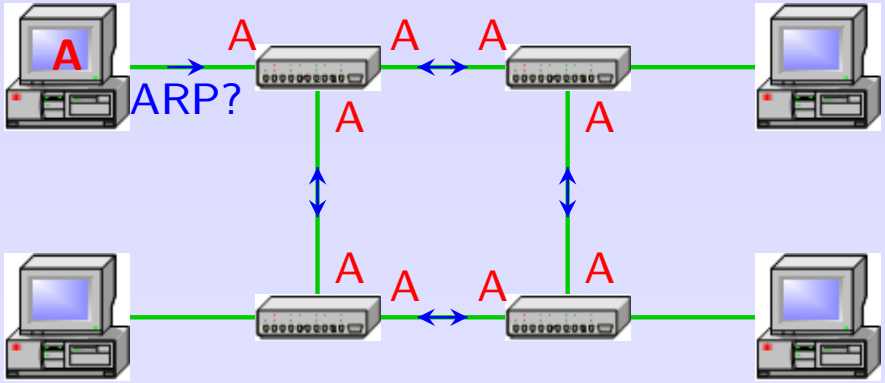
- Für Geräte direkt am Switch gibt es keine Kollision mehr
- Abschaltung von Collision Detection möglich
- Ader an Kollisionssensor frei, vier Drähte zur Datenübertragung
- Gleichzeitiges Senden/Empfangen möglich
- Im Vollduplex erfolgt bidirektionaler Datenaustausch doppelt so schnell
- Achtung: Vollduplex bei Anschluss an Hub nicht möglich

Loop

- Gefahr in geswitchten Netzen: Ring durch falsche Patch-Kabel
- Beispiel: Rechner A sendet ARP-Request

  Switches lernen Position von A irrtümlich an zwei Ports, dadurch werden Pakete künftig im Kreis geschickt

→ Zusammenbruch des Netzwerkes



Spanning Tree Protokoll

- Ziel: Blockieren von Loops und redundante Netzwerke ermöglichen
- Intelligente Switches tauschen Information über Netzwerktopologie aus
- Nach verschiedenen Kriterien (Pfadkosten) wird ein Switch als Master (Root) bestimmt und Baumstruktur als Topologie hergestellt
- Reduntante Ports werden blockiert (und auf Bereitschaft gesetzt)
- Bei Ausfall eines Switches oder Erweiterung des Netzes erneute Berechnung der Topologie
- Berechnung des Baumes dauert höchstens 30 s (keine Nutz-Kommunikation möglich)
- Verschiedene Switch-Hersteller → z.T. inkompatible Algorithmen

Geräte

- Unterschiede zwischen preiswerten und hochwertigen Switches:
  - Spanning Tree,
  - Performance des Backplane (Datendurchsatz),
  - Größe des MAC-Adressen Speichers,
  - Möglichkeiten zur Konfiguration
  - Hilfe bei Fehlersuche: Auslesen der MAC-Tabelle, Hardware-Überwachung (Temperatur, Lüfter)

$\Rightarrow$ Herstellerhomogenität erzeugt Abhängigkeiten, aber erleichtert Arbeit durch garantierte Kompatibilität

Datensicherung, Fehlererkennung und Fehlerbehebung

Zur weiteren Vertiefung können folgende Schlagwörter eigenständig erarbeitet werden (nicht Bestandteil der Vorlesung/Klausur):

- Entstehung eines fehlerhaften Bits
- Bitfehlerrate: $\frac{\Sigma Fehler}{\Sigma Gesamt}$
- Qualität eines Codes (Hamming-Distanz)
- Fehlererkennung vs. Fehlerkorrektur
- Prüfbits
- Blockkodierung
- Paritätsbit
- CRC (cyclic redundancy check)
- Blocksicherung

5. Netzwerktechnik

Schicht 3 - Vermittlungsschicht

= Netzwerkschicht, Network Layer

- bisher: Lokale Netze (LAN), jetzt weltweite Vernetzung (WAN)
- Subnetze im LAN durch Switches $\rightarrow$ höhere Nutzdatenrate. Problem hoher Broadcast-Verkehr bleibt
- Problem: Woher kennt Endgerät weltweit alle MAC-Adressen? Weltweiter ARP-Request geht nicht. $\rightarrow$ Vermittlung
- Problem: Wie ist der beste Weg von Sender zu Empfänger? $\rightarrow$ Wegfindung, Routing
- Schicht 3 = Kommunikation zwischen LAN-Bereichen
- In vielen Feldbusarchitekturen nicht implementiert
- Einführung von Logischen Adressen (Hier: IP-Adressen)

IP-Adressen

- IP = Internet Protokoll
- IP-Adressen und IP-Bereiche von IANA (Internet Assigned Numbers Authority) weltweit zugeteilt
- Bsp: Browser erfragt IP-Adresse von Webserver (z.B. www.google.com) über Nameserver
- IPv4: 4 Byte dezimal, z.B. 192.168.0.3
- IPv6: 16 Byte hexadezimal in 8 Blöcken durch : getrennt, z.B. 2001:0db8:85a3:0000:0000:8a2e:0370:7344

Ermittlung der IP-Addresse von www.google.com:
`ping www.google.com` → `Antwort von 173.194.44.84`

IPv4 Netzwerkklassen

- IP-Adresse spezifiziert Subnetz und einzelnen Teilnehmer
- Vorläufer von IPv4: 8 Bit für Netzwerkadresse, 24 Bit für Hostadresse (einzelner Teilnehmer)
- Unterschiedlich große Subnetze → Netzwerkklassen (ergibt sich aus erstem Byte)

Klasse	Von (Dezimal)	Bis (Dezimal)	Von (Binär)	Bis (Binär)
A	0.0.0.0	127.255.255.255	00000000.X.X.X	01111111.X.X.X
B	128.0.0.0	191.255.255.255	10000000.X.X.X	10111111.X.X.X
C	192.0.0.0	223.255.255.255	11000000.X.X.X	11011111.X.X.X
D	224.0.0.0	239.255.255.255	11100000.X.X.X	11101111.X.X.X
E	240.0.0.0	255.255.255.255	11110000.X.X.X	11111111.X.X.X

- Anzahl möglicher Netzwerkgeräte ist je nach Klasse unterschiedlich begrenzt

Klasse	Von	Bis	Fest	Adressen	Bsp NetzHost
A	0.0.0.0	127.255.255.255	1 Byte	16777216	120.X.X.X
B	128.0.0.0	191.255.255.255	2 Byte	65535	178.16.X.X
C	192.0.0.0	223.255.255.255	3 Byte	256	220.0.16.X

Wie lautet Netzwerkteil von www.google.com und www.1und1.de?

`ping www.google.com` → 173.194.44.84 → .....

`ping www.1und1.de` → 217.160.87.61 → .....

Subnetze

- Administrator bekommt Adressraum (fester Netzwerkteil), kann Hostteil in Subnetze unterteilen
  $\rightarrow$ reduziert Broadcast-Verkehr, begrenzt Fehler/Ausfälle auf Teilnetze
- Unterteilung erfolgt auf Routern (Schicht 3)
- Router regeln weltweite Kommunikation zwischen LAN-Segmenten
- Router leiten Broadcasts nicht weiter, Trennung in Broadcast-Domänen
- Beispiel: Admin bekommt Adresse 178.0.X.X der Klasse B zugeteilt und zerlegt diesen in 256 virtuelle C-Klassen mit je 256 Adressen, also 178.0.0.X, 178.0.1.X, 178.0.2.X, ... 178.0.255.X
- Im Subnetz ist niedrigste Adresse die Netzwerkadresse und höchste Adresse die Broadcastadresse. Im Beispiel hat Admin also nur noch 256×254 Adressen frei
- Woher weiß ein Gerät, ob ein anderes im selben Subnetz ist? $\rightarrow$ .....

(Sub)Netzmaske

- Teilt IP-Adresse in Netzwerk- und Hostteil, genauso vorgegeben wie Adressbereich
- Zur IP-Adresse gehört stets auch Angabe der Netzmaske
- Format: 32 Bit, Folge von n Einsen und $32 - n$ Nullen
  Daher auch Angabe als Postfix $/n$ hinter IP-Adresse, z.B. 178.16.0.0 /16
- Subnetzadresse = bitweise UND-Verknüpfung von IP-Adresse und Netzmaske

Beispiel: Das Netz 178.16.40.0/24 sei in zwei Subnetze geteilt. In welchen Subnetzen liegen die Rechner 18 und 160?

Adresse	178.16.40.18	10110010.00010000.00101000.0	0010010
SNM	255.255.255.128	11111111.11111111.11111111.1	0000000
Netz	178.16.40.	10110010.00010000.00101000.	
Adresse	178.16.40.160	10110010.00010000.00101000.1	0100000
SNM	255.255.255.128	11111111.11111111.11111111.1	0000000
Netz	178.16.40.	10110010.00010000.00101000.	

Übung: Segmentierung von Netzen

Eine Firma hat vier Abteilungen: Forschung (mit 100 Rechnern), Marketing (55 Rechner), Technik (22 Rechner) und Vertrieb (29 Rechner). Alle vier Abteilungen sollen eigene Netze bekommen, die Router sollen mit Zugangsfiltern den Verkehr regeln, die Abteilungen sollen verschiedene Broadcast-Domänen werden. Zugeteilt ist das Netz 178.16.35.0/24. Nehmen Sie eine sinnvolle Aufteilung in Subnetze vor.

Forschung	178.16.35.	→	Adressen: 178.16.35.	bis 178.16.35.
	(Subnetzadresse 178.16.35.0, Broadcastadresse 178.16.35.127, nutzbarer Bereich 126 Adressen von 178.16.35.1 bis 178.16.35.126)			
Marketing	178.16.35.	→	Adressen: 178.16.35.	bis 178.16.35.
Technik	178.16.35.	→	Adressen: 178.16.35.	bis 178.16.35.
Vertrieb	178.16.35.	→	Adressen: 178.16.35.	bis 178.16.35.

Ermittlung des Subnetzes

Subnetzadresse = bitweise UND-Verknüpfung von IP-Adresse und Netzmaske

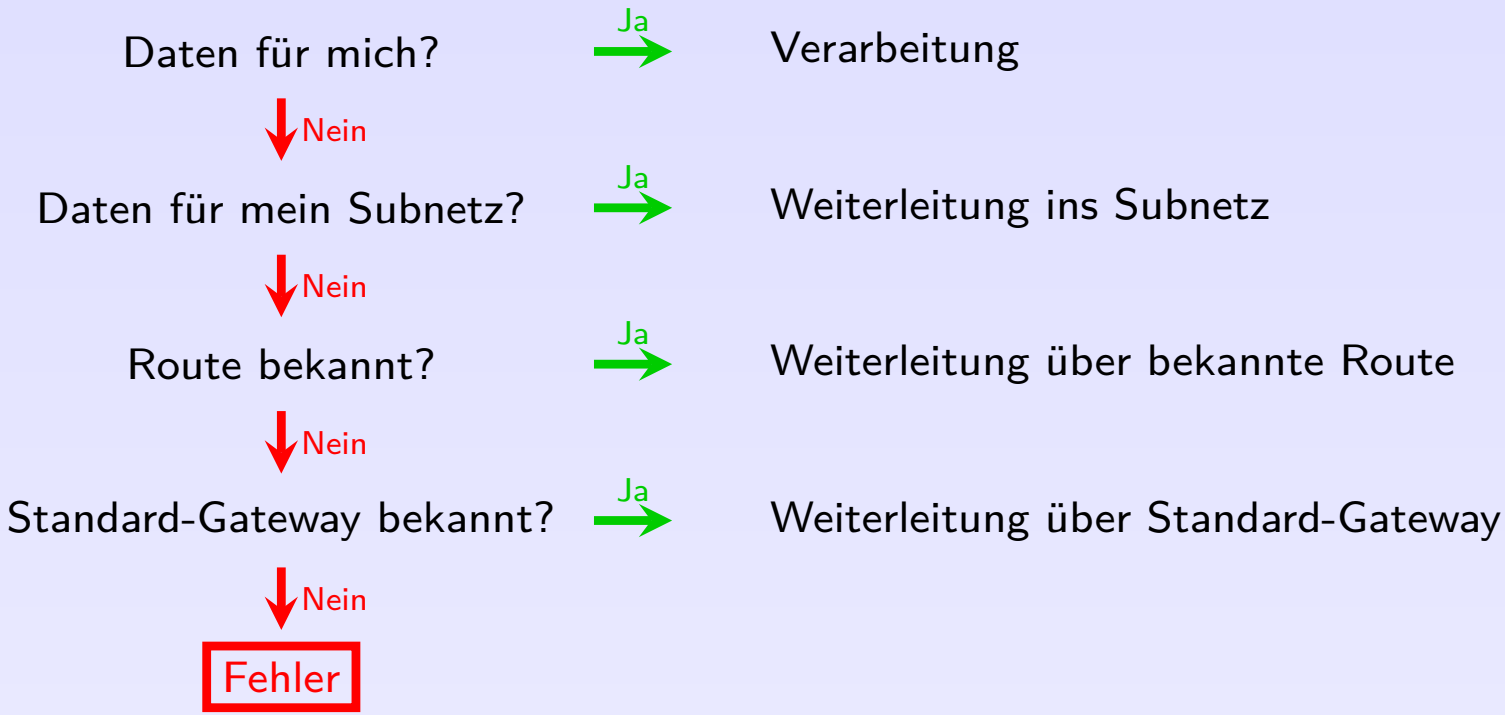
Beispiel: Bereich 178.16.40.0/24 sei in zwei Subnetze geteilt:
178.16.40.0/25 und
178.16.40.128/25.
In welchen Subnetzen liegen die Rechner 18 und 160?

Adresse	178.16.40.18	10110010.00010000.00101000.0	0010010
SNM	255.255.255.128	11111111.11111111.11111111.1	0000000
Netz	178.16.40.	10110010.00010000.00101000.	

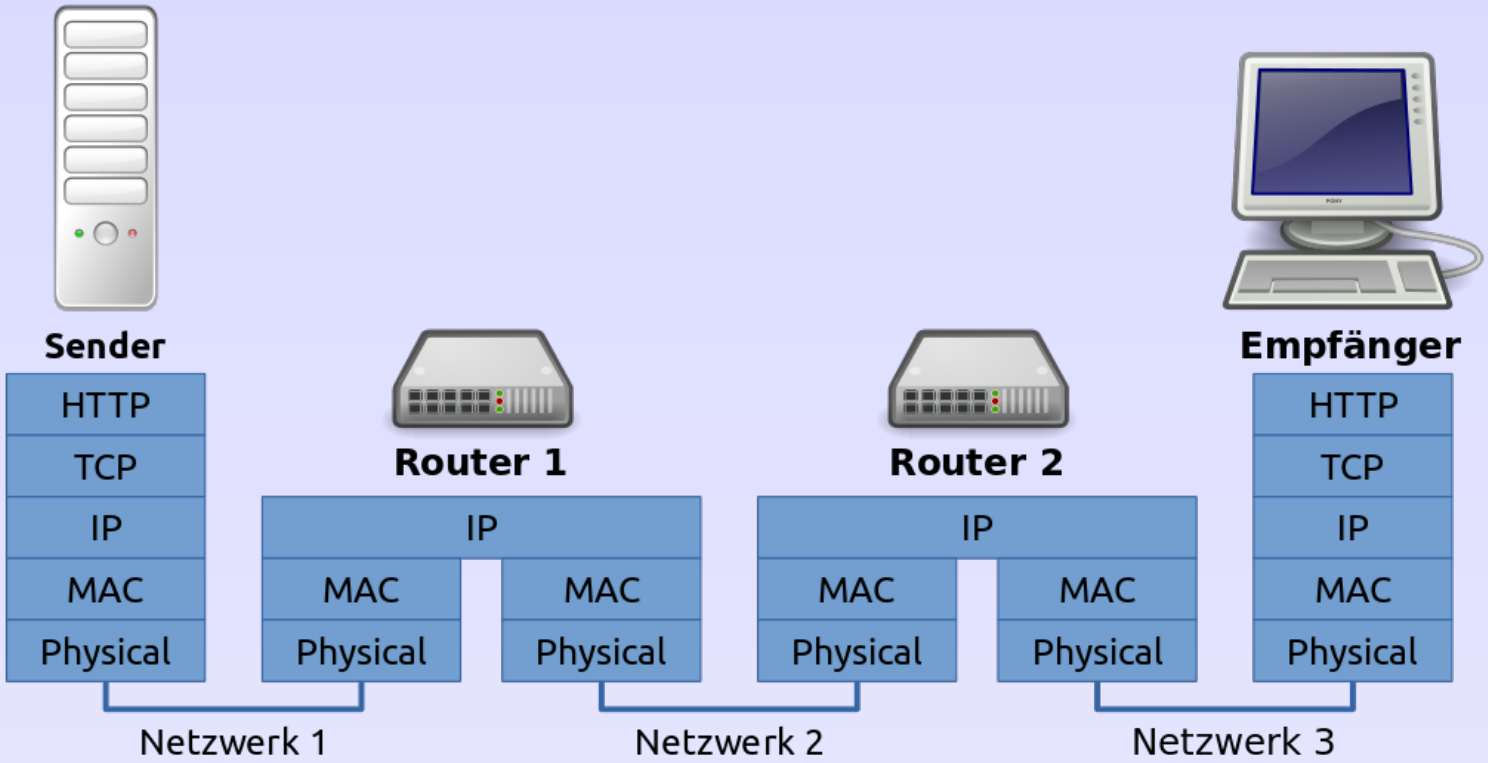
Adresse	178.16.40.160	10110010.00010000.00101000.1	0100000
SNM	255.255.255.128	11111111.11111111.11111111.1	0000000
Netz	178.16.40.	10110010.00010000.00101000.	

Routing

- Routing = Wegfindung
- In kleinen Netzen oft manuelle Konfiguration der Wege: Statisches Routing
- In großen Netzen meist komplexe, dynamische Topologie: Dynamisches Routing
- Router tauschen permanent Informationen über Topologie aus → Routing-Tabellen
- Kriterien für Routing: Zieladresse, Verbindungskosten, notwendige Bandbreite, u.a.



Routing mit IP-Adressen



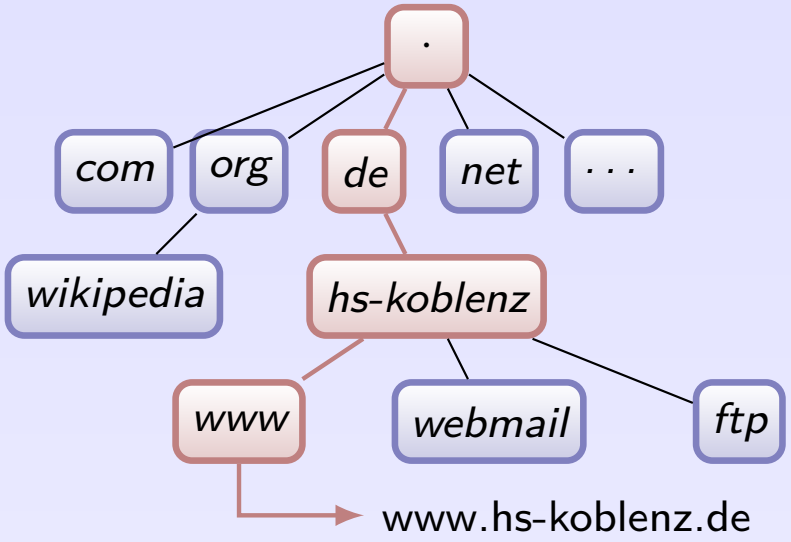
Routenverfolgung mit tracert (Linux: traceroute) oder pathping:
sendet Pakete mit jeweils aufsteigendem TTL (Time-to-Live) an Ziel-Host;
jeder Router dekrementiert TTL und schickt bei TTL=0 einen Fehlercode zurück.

DNS - Domain Name System I

- IP-Adressen schwer zu merken → Vergabe von Namen
- Admin darf zugeteilten Namen erweitern, aber nicht verändern
- Bsp.: Wikipedia erhält Namensraum „wikipedia.org“ und legt darin eigene Namensräume oder Rechnernamen fest, z.B. „www“ oder „de“ als Präfix
- Fully Qualified Domain Name (FQDN)

Computername	. . .	Second-Level	Top-Level	Root-Domain
www.		hs-koblenz.	de	.
de.		wikipedia.	org	.

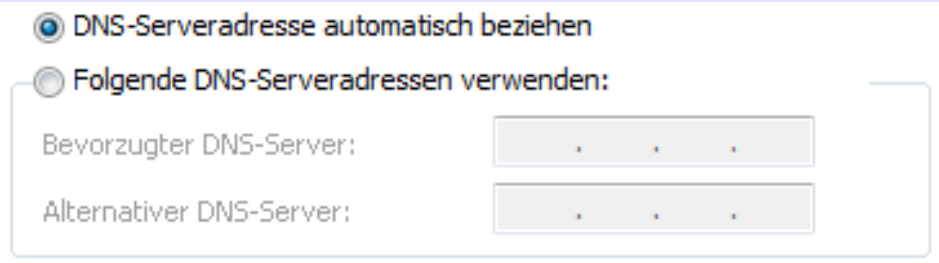
- Domain-Namensraum hat baumförmige Struktur



- Zuordnung von Name zu IP-Adresse über hierarchisches Datenbanksystem

DNS - Domain Name System II

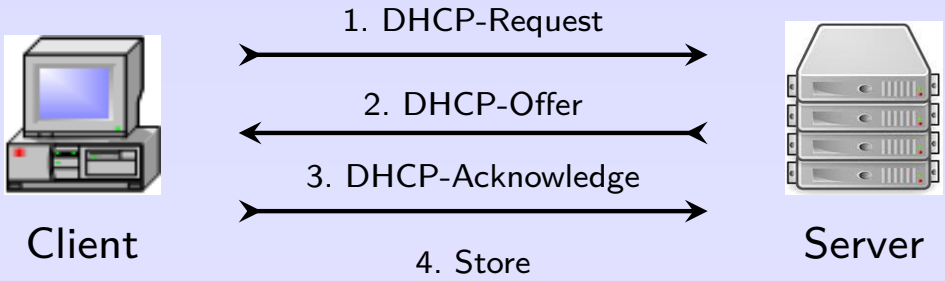
- Weltweit 14 Root-Server (.), darunter jeweils Nameserver für Top-Level-Namensräume (com, org, net, de, . . .)
- Jeder (Admin), der Namen und IP-Adressbereiche registriert, muss Nameserver betreiben, der mindestens den Parent-Server, alle Child-Server und die Root-Server kennt.
- Analog auch Reverse-Lookup (IP→Name) über DNS-Server möglich
- Praxis:
  1. DNS-Server ermitteln: cmd → nslookup
  2. Ermittle/Vergleiche die IP von www.hs-koblenz.de und www.fh-koblenz.de
  3. Ändern des DNS-Servers unter Windows 7: → Netzwerkcenter → Adaptereinstellungen ändern → z.B. LAN-Verbindung → Internetprotokoll V4 → Eigenschaften → DNS-Serveradresse



- Top-Level-Domain-Datenbank: www.elektronik-kompendium.de/service/tld.php

DHCP - Domain Host Configuration Protocol I

- Manuelles Einstellen von IP, SNM, Default Gateway, DNS Server usw. umständlich/fehleranfällig
- DHCP = Adressen automatisch beziehen

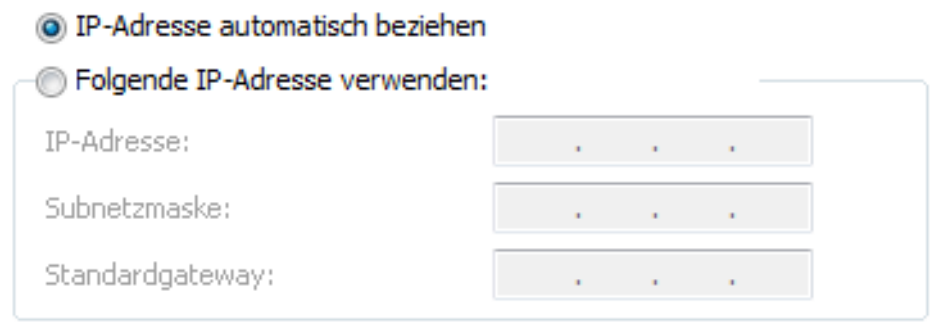


- DHCP-Server (z.B. Router) pflegt Pool verfügbarer IP-Adressen
- Bei Vergabe: Speichern mit Verfallzeit und Streichen aus Pool
- Vor Ablauf der Verfallzeit muss IP-Lebensdauer verlängert werden
- Häufige Anwendung: Mobile Geräte, die oft Subnetz wechseln
- Optional: MAC-Adressenbindung, Client bekommt stets die gleiche IP
- Analog: Kommunikation in lokalen Netzen mit Windows-Namen (z.B. Computername, Netzwerk-Drucker, max. 15 Zeichen, cmd $\rightarrow$ whoami) oder SAMBA (Linux) Auflösung über WINS-Server (Windows Internet Name Service)

DHCP - Domain Host Configuration Protocol II

Praxis:

1. Eigene IP-Adresse ermitteln:
   cmd → ipconfig (Linux: ifconfig)
2. Ändern der IP-Adresse unter Windows 7:
   → Netzwerkcenter → Adaptereinstellungen ändern → z.B. LAN-Verbindung → Internetprotokoll V4 → Eigenschaften → IP-Adresse



5. Netzwerktechnik

Schicht 4 - Transportschicht

- Transportschicht schirmt Details des Transports gegenüber höheren Schichten ab
- Kann zur Übertragung mehrere Kanäle einrichten
- Teilt lange Nachrichten in mehrere Pakete auf
- bringt durcheinander geratene Paketfolgen wieder in richtige Reihenfolge
- Reagiert auf Wiederholungsanforderungen
- Kann gezielt Dienste sperren

Port & Sockets I

- Port: Zuordnung zwischen Datenpaket und Anwendung
- Socket = IP-Adresse + ':' + Port (entspricht Postadresse:Person)
- Über Socketpaar (Source-IP:Port,Destination-IP:Port) ist Verbindung von zwei Rechnern im Internet eindeutig charakterisiert
- Verschiedene Ports $\rightarrow$ verschiedene Anwendungen des selben Clients können gleichzeitig verschiedene Verbindungen aufbauen
- TCP-Port-Übersicht

Well Known Ports	0 - 1023	feste Zuordnung zu Anwendung/ Protokoll, weltweit geregelt
Registered Ports	1024 - 49151	für bestimmte Dienste vorgesehen, teilweise definiert durch IANA
Dynamically Allocated	49152 - 65535	dynamische/freie Nutzung für Clients

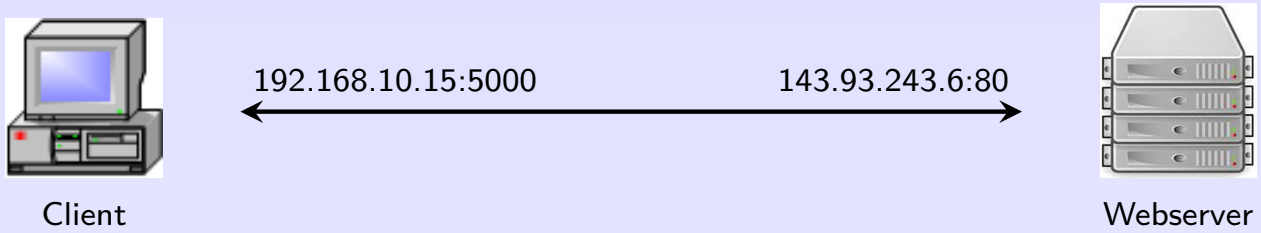
- Gezieltes Sperren einzelner Ports möglich $\hat{=}$ Sperren bestimmter Funktionen

Port & Sockets II

- Beispiele für TCP-Ports

Port	Protokoll	Anwendung
21	FTP	Dateitransfer (FTP-Server)
22	SSH	Secure Shell
25	SMTP	E-Mail Postausgang (SMTP-Server)
80	HTTP	World Wide Web (Webserver)
110	POP	Posteingang (POP-Server)

- Beispiel: Client möchte Webseite von Webserver ansehen. Er sucht willkürlich eigenen freien Port aus und sendet Anfrage an Webserverport 80.



Praxis: Ermittlung von Portnummern und Diensten:
elektronik-kompendium.de/service/tcp.php

UDP - User Datagram Protocol

- Verbindungsloses Transport-Protokoll
- Verbindungslos → unsicher, keine Empfangsbestätigung durch Empfänger
- Bei Stückelung von großen Datenströmen keine Paketnummerierung
- Anwendung: falls Paketverlust unkritisch, z.B. DNS- oder DHCP-Anfragen
- Kommunikation über Port-Struktur
- Aufbau: 8 Byte Header plus variabler Datenblock

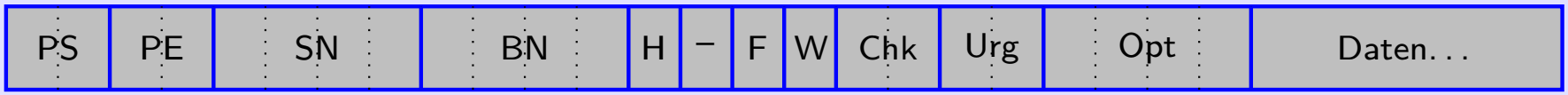
PS Port Sender der anfragenden Anwendung
PE Port Empfänger
Len Gesamtlänge des UDP-Pakets zur Kontrolle der Vollständigkeit
Chk Checksumme

\+ Kleiner Header → hohe Nutzdatenrate, effizient

\+ Mehrere Empfänger möglich, kein Fehler bei Abwesenheit des Empfängers

TCP - Transmission Control Protocol

- Verbindungsorientiertes, sicheres Transport-Protokoll
- Gezielter Verbindungsauf- und Abbau, Quittierung von Paketen
- Bei Stückelung von großen Datenströmen erfolgt Paketnummerierung
- Bei Verlust/Beschädigung von Paketen erneutes Senden
- Kommunikation über Port-Struktur
- Aufbau: 24 Byte Header plus variabler Datenblock



PS Portnummer der sendenden Anwendung
PE Portnummer der empfangenden Anwendung
SN fortlaufende Sequenznummer des Datenpakets
BN Bestätigungsnummer des zuletzt korrekt empfangenen Pakets
H Header Length in 32-Bit Blöcken (4 Bit)
– unbenutzt (6 Bit)
F Flags (6 Bit): URG, ACK, PSH, RST, SYN, FIN
W Window-Size, maximale Paketgröße
Chk Checksumme zum Prüfen des Headers
Urg gibt mit Sequenznummer genaue Lage des Paketes an
Opt Optionale Daten oder aufgefüllte Nullen

Firewall

- Auf Schicht 3 kann Router nach IP-Adressen filtern und diese komplett sperren
- Auf Schicht 4 können einzelne Ports gesperrt werden, z.B. Zugriff auf FTP-Server (Port 21) von außen
- Firewall = Router mit invertierter Philosophie:
  Router ist maximal offen und kann unerwünschten Verkehr explizit sperren
  Firewall ist maximal gesperrt, Weiterleitung muss explizit erlaubt werden

NAT & PAT

Gründe für NAT & PAT

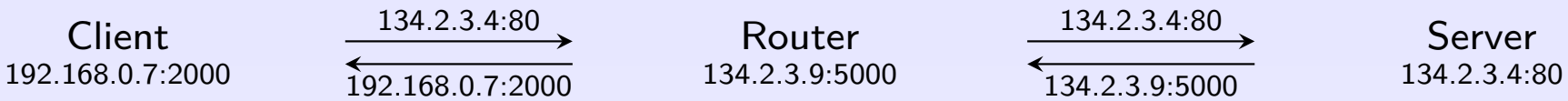
- IPv4-Adressen sind mittlerweile knapp geworden
- Sicherheitsaspekte: Client mit privatem Socket im Internet nicht sichtbar/angreifbar

NAT - Network Address Translation

- Router kann Adressen von Paketen ändern und Antwortpakete rückadressieren
- In (privaten) Subnetz kommunizieren alle Teilnehmer (Rechner, Drucker) miteinander und einige davon (z.B. nur Rechner) mit NAT in nach außen gültigen Adressen im Internet. Interne Netzwerkstruktur bleibt nach außen verborgen.
- Zuordnung: n private Adressen $\leftrightarrow$ n öffentliche Adressen

PAT - Port and Address Translation (NAT Overload, IP-Masquerading)

- Sendet Client Anfrage an Server im Internet, wird sein privater Socket (Adresse plus Port) vom Router in offizielle Adresse mit freiem Port gewandelt
- Zuordnung: n private Sockets $\leftrightarrow$ 1 öffentliche Adresse mit n Ports
- Es werden nur Source-Ports vom Client umgeschrieben und nicht Destination-Ports



5. Netzwerktechnik

Schicht 5 - Sitzungsschicht

= Session Layer, Kommunikationssteuerungsschicht

- steuert logische Verbindungen, organisiert und synchronisiert Datenaustausch
- Definiert Fixpunkte (Check Points), an denen Sitzung nach Ausfall der Verbindung wieder synchronisiert werden kann, ohne Übertragung von vorne zu beginnen

Schicht 6 - Darstellungsschicht

= Presentation Layer

- Zeichenkodierung: EBCDIC (Extended Binary Coded Decimal Interchange Code), ASCII (American Standard Code for Information Interchange)
- Datenkompression für bessere Ausnutzung der Bandbreite
- Verschlüsseln der Daten während der Übertragung, z.B. mit SSL (Secure Socket Layer), SSH (Secure Shell), abhörsichere Telefone
- Codierung für Video- und Audioübertragung
- Beispiele: MPEG, TIFF, GIF, ASCII

Schicht 7 - Anwendungsschicht

= Application Layer

- stellt Mittel für die Kooperation zwischen verteilten Anwendungsprozessen zur Verfügung
- Dienste:
  - HTTP - Hypertext Transfer Protocol
  - E-Mail Protokolle: SMTP, POP, IMAP
  - FTP, SFTP
  - Virtueller Terminal (VT), ermöglicht einem Benutzer das arbeiten an entfernten Computern, als säße er am lokalen Rechner (z.B. VNC, Teamviewer)

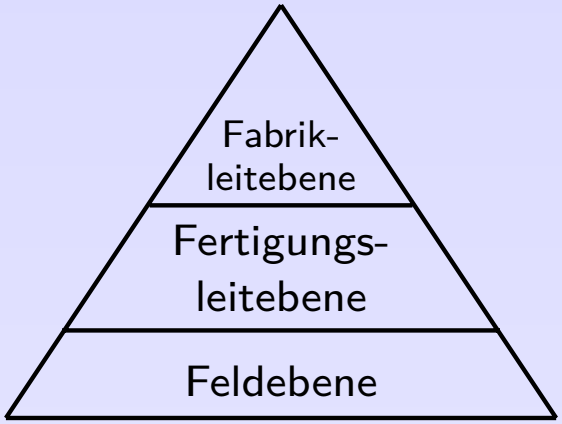
ISO-OSI Schichtenmodell - Überblick der Schichten

Nr.	Bezeichnung	Aufgaben
7	Anwendungsschicht (Application Layer)	Funktionen für Anwender,z.B. E-Mail, Datenbankabfrage
6	Darstellungsschicht (Presentation Layer)	Syntax der Datentypen vereinbaren (Sprache), Formatierung, Komprimierung, Verschlüsselung, Zeichensatz
5	Sitzungsschicht (Session Layer)	Schnittstelle zur Benutzung der logischen Kanäle, Auf- und Abbau von Sitzungen
4	Transportschicht (Transport Layer)	Verbindungen/Kanäle auf-/abbauen, Paketierung, Quittierung, TCP, UDP, NAT, PAT
3	Vermittlungsschicht (Network Layer)	Vermittlung (IP-Adressen, DNS, DHCP), Segmentierung, Routing, Router
2	Datenverbindungsschicht (Data Link Layer)	MAC-Adressen, ARP, Loops, Switch
1	Physikalische Schicht (Physical Layer)	el. / mech. Eigenschaften des Übertragungsmediums, Topologie, Zugriffsverfahren, Repeater

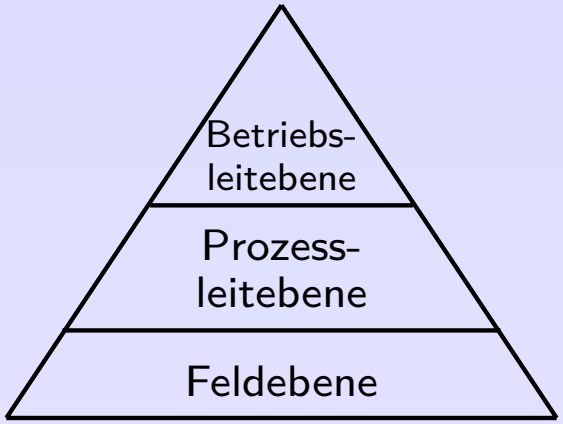
6. Industrielle Kommunikation

Automatisierungshierarchie

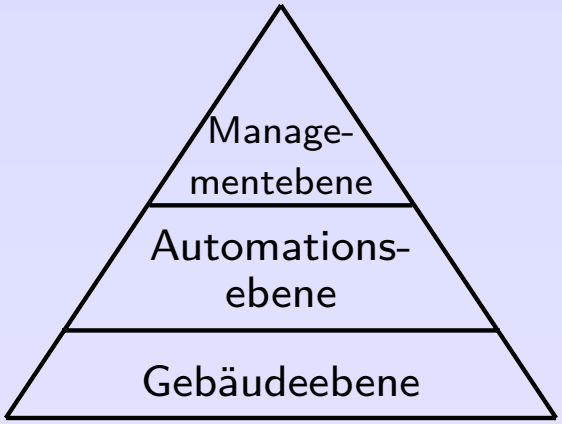
Industrielle/Gewerbliche Kommunikation wird für unterschiedliche Anwendungsbereiche in (drei) typische Anforderungsebenen strukturiert.



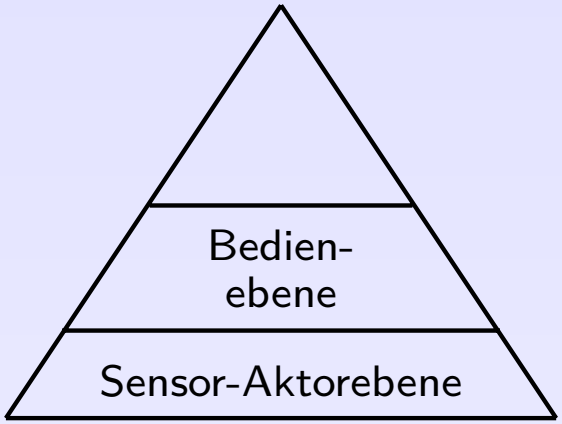
Fertigungsautomatisierung



Prozessautomatisierung



Gebäudeautomatisierung



Geräteautomatisierung

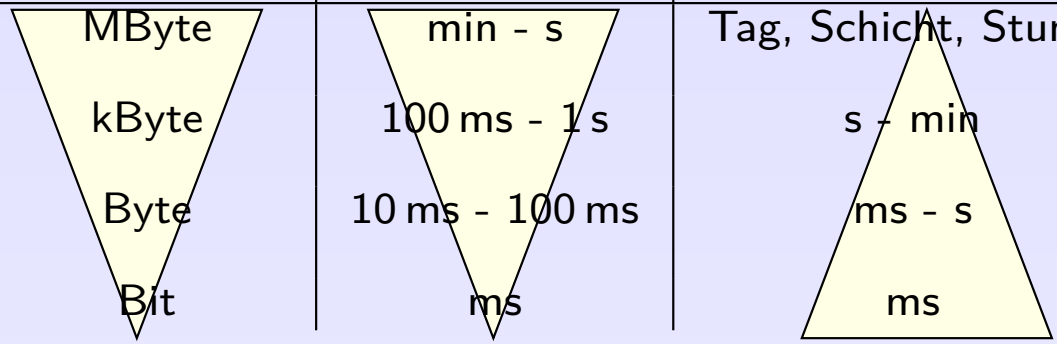
Ebene	**Aufgaben**	**Beispiele**
Unternehmens-leitebene	Produktionssteuerung, Wirtschaftlichkeit	Aufträge, Instandhaltung
Betriebs-leitebene	Verfügbarkeit, Qualität	Anlagenvisualisierung, Programmierung, Diagnose
Steuerungs-ebene	Produktion	SPS, Aktoren, Sensoren, Bedienen, Beobachten

Informationsfluss in der Automatisierungshierarchie

Informationsfluss innerhalb und zwischen Ebenen (Horizontale und Vertikale Vernetzung) mit unterschiedlichen Übertragungsanforderungen

Ebene	**Automatisierungsgrad**	**Zeitliche Anforderung**	**Räumlicher Abstand**
Disponieren	Gering	Offline	Prozessfern
Leiten	Mittel	Online	Prozessfern
Steuern/Regeln	Hoch	Echtzeit	Prozessnah

	Datenmenge	**Reaktionszeit**	**Häufigkeit**
Leitebene	MByte	min - s	Tag, Schicht, Stunde
Zellebene	kByte	100 ms - 1 s	s - min
Feldebene	Byte	10 ms - 100 ms	ms - s
AS-Ebene	Bit	ms	ms



- Untere Ebene: Feldbus
- Obere Ebene: Prozessbus
- Oft Vernetzung von Feldbus und Prozessbus in SPS als Gateway

Arten der Kommunikation in der Automatisierungshierarchie

Feldkommunikation: Feldgeräte (Sensor, Aktor) mit Steuerrechner
→ Kosteneinsparung im Schaltschrank und bei Verkabelung durch Dezentralisierung der Feldebene mittels serieller Bustechnik

Datenkommunikation: Steuerrechner miteinander, Visualisierung, Programmierung, Diagnose
→ Dezentralisierung der Steuerintelligenz vereinfacht Wartung, Fehlersuche und Inbetriebnahme

IT-Kommunikation: Steuerebene mit Prozessleitebene
→ Prozessüberwachung, Fernwartung

Beispiel: Ein Sensor . . .

- liefert kontinuierlich Prozessdaten
- wird gelegentlich parametriert oder liefert Statusdaten
- sendet im Fehlerfall Alarmsignal

Feldbusse - Anforderungen und Auswahlkriterien

Allgemeine Anforderungen an Feldbussysteme

- Zuverlässige und montagefreundliche Busanschlüsse
- Bidirektionaler Informationsfluss
- An- bzw. Abkoppeln von Geräten im Betrieb
- Versorgungsspannung über Bus
- Offenheit, Herstellerunabhängigkeit
- Erdfrei und galvanisch getrennt
- Zündschutzart „Eigensicher“

Nachrichtenbehandlung

Bandbreite benötigte Übertragungskapazität

Echtzeit garantierte maximale Übertragungszeit

bevorrechtigte Behandlung von Nachrichten

Systemsicherheit

- Sicherheit gegen Ausfall des Übertragungsnetzwerkes
- Sicherheit gegen Ausfall der Teilnehmer
- Sicherheit gegen Übertragungsstörungen

Echtzeitverhalten

Echtzeit $\hat{=}$ garantierte Reaktionszeit, Rechtzeitigkeit

Prozessdaten (E/A Daten)

- Echtzeitdatenübertragung
- unmittelbare Auswirkung auf den physikalischen Prozess
- zyklische Informationen
- Informationslänge einige Bits bzw. Bytes
- Aktualisierung im Bereich von Milli- / Mikrosekunden
- ständig aktualisiertes Prozessabbild

Parameterdaten

- asynchrone Übertragung
- Einstellung, Beobachtung und Programmierung von Geräten
- Informationslänge bis einige hundert (k)Bytes
- Zeitanforderungen unkritisch
- Besondere Sicherungs- und Quittierungsmechanismen

6. Industrielle Kommunikation

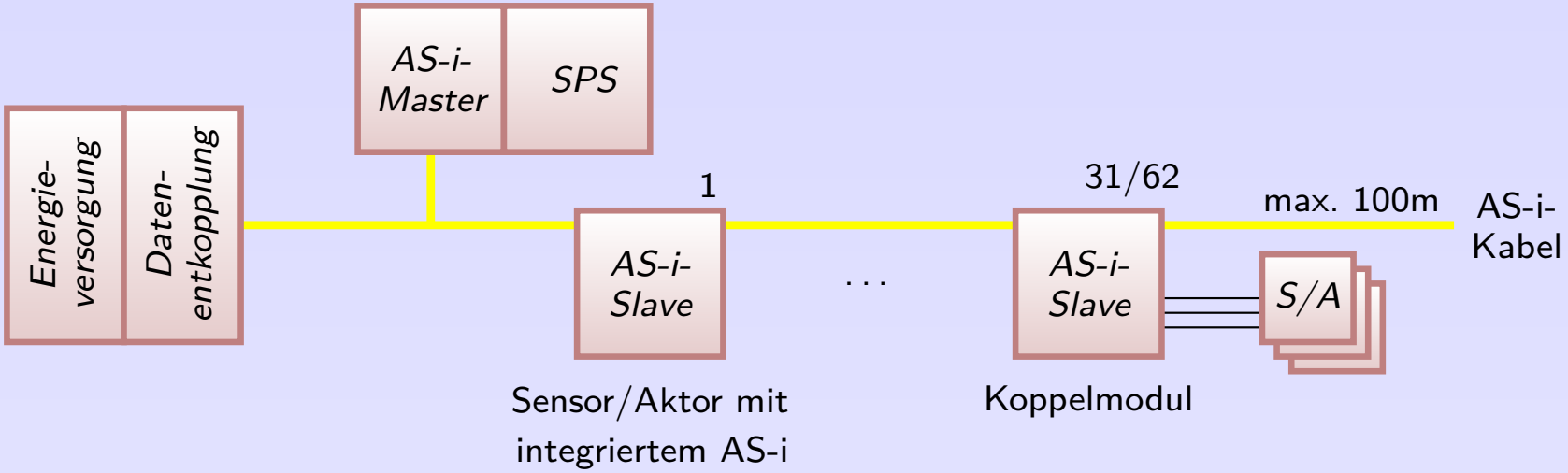
AS-i - Aktor-Sensor-Interface

- Internationaler Standard für Feldbuskommunikation (IEC 62026-2)
- Seit 1993
- AS-International Association: weltweit etwa 300 Mitgliedsunternehmen
- Alternative zur Parallelverkabelung von Sensoren und Aktoren
  - \+ Flexibilität
  - \+ Wirtschaftlichkeit
  - \+ Einfachheit, Reduktion von Installationsfehlern
  - \+ Hohe Verbreitung, gute Vernetzungsmöglichkeiten
- Einfachheit $\Rightarrow$ Einsatz nur in unterster Automatisierungsebene
- Andere Feldbussysteme decken diese Ebene nicht ab oder sind unrentabel (teuer, aufwendiges Protokoll) und verbinden eher (dezentrale) Steuerungen und I/O-Geräte miteinander.
- Auch Safety-Funktion erhältlich (z.B. Not-Aus-Schalter am ASi-Bus)

Literatur & Info

- AS-International Association e.V.: `as-interface.net`
- AS-Interface von Siemens: `youtu.be/GG2prQ_15Og`
- AS-Interface Modul Pepperl + Fuchs: `youtu.be/5XUaS6j8z_w`
- [WZ09, Kap. 16]
- [Hei15, Kap. 5.6]
- TU Braunschweig, Ind. Kommunikation mit Feldbussen, Kap. 6

AS-i - Prinzip



Schicht 1 (Bitübertragung)

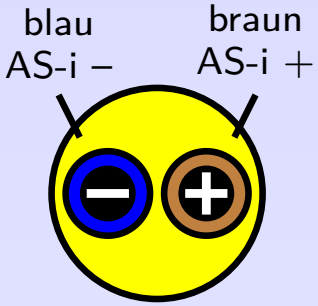
- Kabel, Stecker
- Topologie, Leitungslänge
- Buszugriff: zyklisches Polling
- Bitübertragung durch Manchester-Code und APM mit $\sin^2$-Impulsen

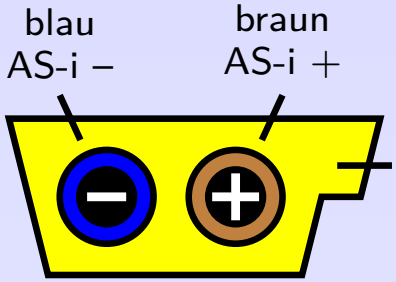
Schicht 2 (Datensicherung)

- Telegrammaufbau
- Kontrolle von Telegrammlänge, Impulspause, Start- und Endebit, Paritätsbit $\rightarrow$ Hamming-Distanz $= 4$

AS-i - Kabel

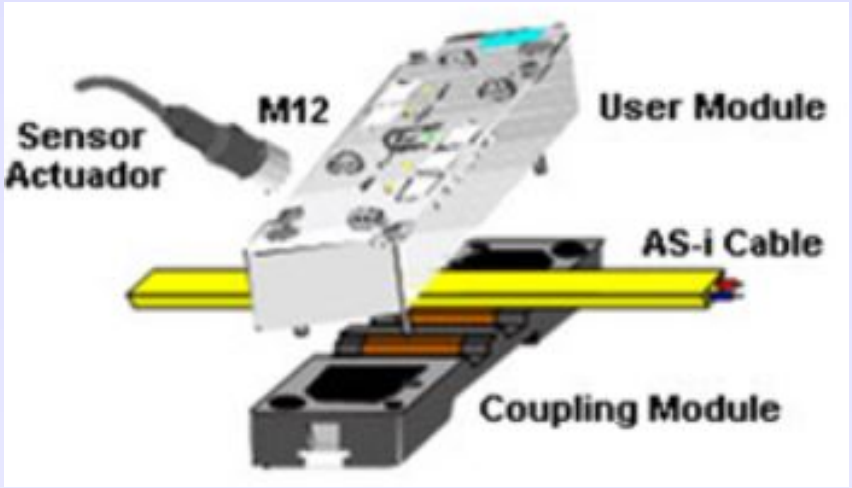
- runde ungeschirmte Zweidrahtleitung oder ungeschirmte Flachbandleitung

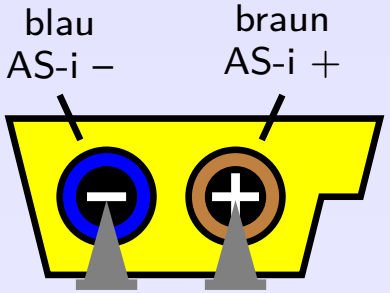




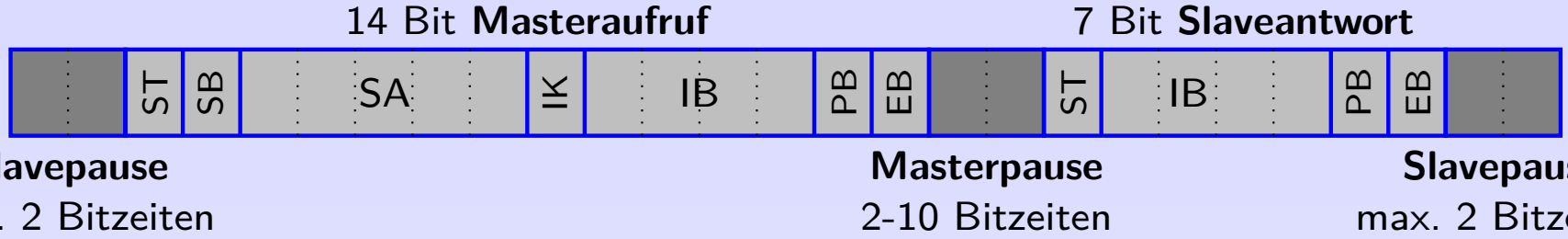
gelb = Daten/Hilfsenergie
schwarz = Energie 24V
rot = Energie 230V

- Installation ohne Trennung des Kabels möglich





AS-i - Datentelegramm (V2.0)



Bitzeit	$\approx 6\,\mu s$ (167 kBit/s)
Telegrammzeit	
Zykluszeit	

- ST Startbit, immer 0
- SB Steuerbit:
  0≙Daten-/Parameter-/Adressaufruf
  1≙Kommandoaufruf
- SA Adresse des Slave $2^5 = 32$
  0 reserviert zum Adressieren
- IK Informationskennung:
  0≙Datenaufruf, 1≙Parameteraufruf
- IB 4 Informationsbits: Daten/Parameter
- PB Paritätsbit von (ST bis IB)
- EB Endebit, immer 1

Beispiele:

Datenaufruf

0	0	A_4	A_3	A_2	A_1	A_0	0					1

Parameteraufruf

0	0	A_4	A_3	A_2	A_1	A_0	1	D_3	D_2	D_1	D_0	1

Adressieraufruf

0	0	0	0	0	0	0	A_4	A_3	A_2	A_1	A_0	1

Kommandoaufruf, z.B. Reset Slave

0	1	A_4	A_3	A_2	A_1	A_0	1	1	1	0	0	1

AS-i - Komponenten

- Master, Slaves (Sensoren und Aktoren mit AS-i Anschluss oder AS-i Koppelmodule für konventionelle Geräte), Leitungen, Netzteil, Verstärker, Gateway, ...
- Konfiguration der Slaves mit 4 Codes: IO (Anzahl I/O), ID, ID1 und ID2 (je 4 Bit, 0...F)
- www.as-interface.net/products

AS-i - Technologie

Busverwaltung	Master tauscht zyklisch E/A-Daten mit Slaves aus (zyklisches Polling)
Netzstruktur	Linie, Baum
Protokoll	seriell, 4 Bit Nutzdaten
Ü-Medien	runde Zweidrahtleitung oder Flachbandleitung, ungeschirmt gelb = Daten, 24V-30VDC, Hilfsenergie über Datenkabel möglich schwarz = Energie 24VDC, rot = Energie 230VAC
Leitungslänge	max. 100 m, Verlängerung auf max 300 m mit Repeater
Teilnehmer	1 Master, max. 31 Slaves mit je max 4 EA (max. 124 EA) V2.11: 1 Master, max. 62 Slaves (Unterscheidung: Bit 4)
Adressierung	feste Adressen (durch Adressiergerät oder Master)
Nettodatenrate	V2.0: 4 Bit von Master, 4 Bit von Slave ab V2.11: 3 Bit von Master, 4 Bit von Slave
Übertragungsrate	167 kBit/s bei 31 Slaves = 5 ms abhängig von der Slaveanzahl
Fehlersicherung	Paritätsbit, diverse Formfehlererkennung
Sonstiges	Daten- und Energieübertragung auf einem Kabel möglich
Einsatzgebiete	unterste Feldebene, Schnittstelle zwischen binären Sensoren / Aktoren und (dezentralen) Steuerungen, Fertigung, Elektroindustrie, Maschinenbau
Versionen	AS-i 1 (1993/94, V2.04) Markteinführung, 31 Slaves AS-i 2 (1998, V2.11) bis 62 Slaves AS-i 3 (2005/07, V3.0) Gateways zu Ethernet, Profinet u.a. erhältlich

AS-i Safety at Work

- Sicherheitsmonitor (nicht Master) erfüllt Sicherheitsfunktion (z.B. bei Geräteausfall)
- Standard AS-i Bus + sicherheitsgerichtete Geräte wie
  Not-Aus, Türkontakte, Lichtgitter/-vorhänge, optische Scanner
- Zertifizierter Standard (TÜV, Institut für Arbeitsschutz IFA) zum Einsatz von sicherheitsgerichteten Komponenten im AS-i Netz (bis SIL 3, PL e)

6. Industrielle Kommunikation

Profibus - Process Field Bus

- 1987: Zusammenschluss von 21 Firmen/Instituten zur Realisierung eines offenen, bitseriellen Feldbusses
- 1989: Profibus Nutzer-Organisation e.V. www.profibus.com
- 1999: Normung in IEC 61158 und IEC 61784
- Litaratur: de.wikipedia.org/wiki/Profibus
- Nur OSI-Schichten 1,2 und 7 festgelegt
- Schicht 1: bei drahtgebundener Übertragung RS 485, überwiegend geschirmte Zwei-Drahtleitungen, Busabschlusswiderstände erforderlich

Profibus - Varianten

FMS - Field Message Specification

- Vernetzung von *Steuerungen* auf Automatisierungsebene
- sehr leistungsfähig, geeignet für umfangreiche/komplexe Kommunikationsaufgaben
- durch Industrial Ethernet und DP (einfacher, schneller) abgelöst
- ab 2007 nicht mehr normiert

DP - Dezentrale Peripherie

- Einsatz in Fertigungstechnik
- Vernetzung von Steuerungen und Anschaltung von Aktoren und Sensoren
- Weiterentwicklung/Vereinfachung von FMS, identische Schichten 1 (Leitungen, Stecker, Repeater) und 2 (Telegrammformat)
- nur Schicht 1+2, bis 12 Mbit/s $\rightarrow$ sehr schnell

PA - Prozessautomation

- Einsatz in Verfahrenstechnik
- Anschaltung von Feldgeräten an Prozessleitsysteme, Versorgung von Feldgeräten möglich
- 31,25 kbit/s (hohe EMV, lange Kabelwege)
- eigene physikalische Schicht: Eigensicher, Ex-i

Profibus - Spezifikation

Busverwaltung	Multi-Master mit Token-Passing und unterlagertem Master-Slave
Netzstruktur	Linear mit Stichleitungen und Abschlusswiderständen
Ü-Medien	Verdrillte (abgeschirmte) 2-Drahtleitung, LWL, Integrierte Hilfsenergieübertragung
Leitungslänge	Mit Repeatern bis 10 km bei niedriger Datenrate
Teilnehmer	32 pro Segment, mit Repeatern erweiterbar auf 126, max. 32 Master
Adressierung	Sender- und Empfängeradresse im Telegramm
Nettodatenrate	0 .. 244 Byte je Telegramm
Übertragungsrate	9600 bit/s bis 12 Mbit/s (31,25 kbit/s bei PA), abhängig von Leitungslänge
Fehlersicherung	Parität je Byte, Frame Check Sequence (Quersumme)
Einsatzgebiete	Profibus FMS im Zellbereich Profibus DP in Feldebene, Fertigungstechnik Profibus PA in Prozessautomatisierung und eigensicherer Bereich

6. Industrielle Kommunikation

Profinet - Process Field Network

- youtu.be/u2vjb9qFZpM
- Offener Industrial Ethernet Standard für Automation
- Entwickelt von Profinet Nutzerorganisation (Siemens u.a.)
- Vernetzung von SPSen, Ankopplung an PCs, Anbindung von Sensoren/Aktoren
- Modulares Konzept, Funktionalität wählbar
- Getrennte Übertragung zeitunkritischer und zeitkritischer Echtzeitdaten
- Zykluszeiten unter 1 ms im taktsynchronen Modus möglich
- Kompatible Kabel- und Netzwerktechnik mit Ethernet
- Relativ einfache Erweiterbarkeit
- Autocrossover
- Zwei Sichtweisen: IO und CBA, können separat betrieben oder kombiniert werden

Profinet IO - Input/Output

- Anbindung dezentraler Peripherie (IO-Devices) an Steuerung (IO-Controller)
- Drei Konformitätsklassen, Zykluszeiten in Klasse C (für Motioncontrol) bis 31,25 µs
- Mehrere Controller zur Realisierung von Modularisierung oder Systemredundanz möglich
- CR - Communication Relations:

  IO Data CR für zyklischen Prozessdatenaustausch, minimale Headerlänge, Kommunikation auf MAC-Ebene, keine IP-Adressen, nicht routingfähig

  Record Data CR für azyklischen Parametertransfer wie Konfiguration oder Diagnosemeldungen, nutzt UDP/IP Protokoll

  Alarm CR für Übertragung von Alarmen in Echtzeit, spezielle azyklische Nachrichten, zeitkritisch, Empfang muss quittiert werden
- IO-Device Eigenschaften vom Hersteller in GSD-Datei beschrieben (General Station Description)

Profinet CBA - Component Based Automation

- Komponentenbasierte Kommunikation über TCP/IP
- Grundgedanke: Gliederung einer gesamten Automatisierungsanlage in autonom arbeitende, überschaubare Teilanlagen
- Komponente umfasst hier alle mechanischen, elektrischen und informationstechnischen Größen/Geräte
- Komponente beschrieben in PCD-Datei (Profinet Component Description)
- Profinet CBA (ohne Real-Time) geeignet für Reaktionszeiten $\approx 100\,\text{ms}$. Im parallel angeordneten RT-Kanal sind Datenzyklen wie bei Profinet IO möglich.

Profinet Dienste und Protokolle I

<table>
<tr><th colspan="2">OSI-Schicht</th><th>Englisch</th><th colspan="4">PROFINET</th></tr>
<tr><td>7a</td><td rowspan="2">Anwendung</td><td rowspan="2">Application</td><td colspan="2">IO Dienste & Protokolle</td><td colspan="2">CBA Dienste & Protokolle</td></tr>
<tr><td>7b</td><td rowspan="5">leer</td><td>RPC</td><td>DCOM & RPC</td><td rowspan="5">leer</td></tr>
<tr><td>6</td><td>Darstellung</td><td>Presentation</td><td colspan="2">--</td></tr>
<tr><td>5</td><td>Sitzung</td><td>Session</td><td colspan="2">--</td></tr>
<tr><td>4</td><td>Transport</td><td>Transport</td><td>UDP</td><td colspan="2">TCP</td></tr>
<tr><td>3</td><td>Netzwerk</td><td>Network</td><td colspan="3">IP, ARP, SNMP, DHCP</td></tr>
<tr><td>2</td><td>Sicherung</td><td>Data Link</td><td colspan="4">CSMA/CD, VLAN, DCP, MRP, MRRT, LLDP</td></tr>
<tr><td>1</td><td>Bitübertragung</td><td>Physical</td><td colspan="4">100BASE-TX, 100BASE-FX</td></tr>
</table>

[de.wikipedia.org]

100BASE-TX/FX 100 MBit/s Full-Duplex elektrisch (TX) oder optisch (FX)

VLAN mit Priority Tagging Echtzeitdaten erhalten größte Priorität für Weiterleitung

MRP - Media Redundancy Protocol Aufbau einer redundanten, protokollunabhängigen Ringtopologie mit Umschaltzeit unter 50 ms

Profinet Dienste und Protokolle II

MRRT - Media Redundancy Real-Time Protocol ermöglicht redundante Ringschaltungen ohne Zeitverzögerung im Fehlerfall

LLDP - Link Layer Discovery Protocol zur Erkennung von Nachbarn und Signallaufzeiten

IP - Internet Protocol Verbindungsaufbau azyklischer Dienste

ARP - Address Resolution Protocol Ermittlung von MAC-Adressen

SNMP - Simple Network Management Protocol Überwachung und Konfiguration von Geräten (Router, Switches, Computer) durch zentrale Station

DHCP - Dynamic Host Configuration Protocol Automatische Vergabe von IP-Adressen

RPC - Remote Procedure Call Aufbau und Verwaltung von Verbindungen

DCOM - Distributed Component Object Model

Protokollstufen

Protokollstufe	Profinet	Anwendungen
TCP/IP	CBA	Reaktionszeiten $\approx 100\,\text{ms}$
RT (Real-Time)	CBA, IO Klasse A&B	typisch 10 ms Zykluszeiten
IRT (Isochronous RT)	IO Klasse C	Zykluszeiten unter 1 ms

6. Industrielle Kommunikation

IO-Link I

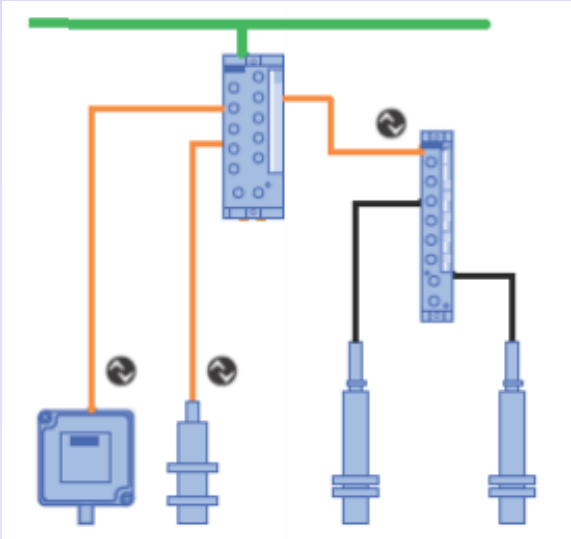
IO-Link

- herstellerübergreifender Standard (IEC 61113-9)
- Kommunikation mit (intelligenten) Sensoren/Aktoren über Master
- IO-Link Firmengemeinschaft c/o PROFIBUS Nutzerorganisation e. V. (PNO)
- Serielle, bidirektionale Punkt-zu-Punkt-Verbindung $\rightarrow$ kein Feldbus
- Energie und Daten über einheitliche Verdrahtung
- kostengünstig, effizient, flexibel
- IO-Link Master: Gateway zu höherwertigen Schnittstellen (Industrial Ethernet, OPC-UA)
- „IO-Link hat Potenzial, in wenigen Jahren 4..20mA Schnittstelle zu ersetzen“ (Anm.: abwarten!)

IO-Link II

Komponenten

- IO-Link Master: Kopplung an SPS oder Feldbus
- Devices (Sensoren, Aktoren),
  auch ohne IO-Link (als DI oder DO)
- Ungeschirmte 3- oder 5-Leiter-Kabel
- Engineering-Tool zur Projektierung und Parametrierung



Schnittstelle

- 5-polige M12 Steckverbinder
- Pin 1: 24V (max 200mA), Pin 3: 0V, Pin 4 Daten, Pin 2+5:
  zusätzliche Versorgung

Kommunikation mit intelligentem Sensor:

- Messdaten, Prozesswerte (zyklisch)
- Wertstatus: (zyklisch) zeigt Gültigkeit der Prozessdaten an
- Diagnose, Statusmeldungen: Verschmutzung einer Lichtschranke, Überhitzung
- Parameterdaten: Schwellwert für (Reflektions)Lichtschranke oder Positionssensor
- Geräteidentifikation

IO-Link III

Geräteprofile

- Beschreibungsdatei IODD (IO Device Description)
- Baudrate, Datenstruktur und -größe (bis 32 Byte), minimale Zykluszeit, Identifikation, Textbeschreibung, Bild des Gerätes, Logo des Herstellers

Datenzugriff

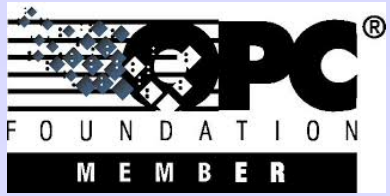
- Zyklische Daten werden vom Master auf zuvor eingestellte Adressbereiche gelegt oder davon gelesen. Steuerung greift über diese Adressen auf Prozesswerte zu
- Azyklische Daten über Index-Bereich oder Funktionsbausteine zur Programmierung
- Parameter im IO-Master gespeichert, einfacher Device-Wechsel möglich durch automatische Neuparametrierung

Literatur, Vertiefung

- `www.io-link.com` → `Downloads` → `Systembeschreibung`

6. Industrielle Kommunikation

OPC - Einführung



- OPC (früher): OLE for Process Control
  OLE (Object Linking and Embedding): Technologie von Microsoft
  auch bekannt als COM (Component Object Model): Standard für Zusammenarbeit von SW-Komponenten innerhalb PC, verteilte Systeme: DCOM (Distributed COM)
- OPC (seit 2011): Open Platform Communications
- Standardisierte Software-Schnittstelle zum Datenaustausch in AUT
- Industrieller Kommunikationsstandard zur Anbindung von AUT-Geräten unterschiedlicher Hersteller an übergeordnete OPC-fähige Programme der Betriebsleitebene (z.B. Excel)
- seit 1996 OPC Foundation, über 400 Firmen

- Kostenlose Zertifizierungssoftware nur für Mitglieder: OPC Compliance Test,
  testet vollständige OPC-Funktionalität,
  simuliert Fehlverhalten eines Clients, überprüft alle Fehlercodes,
  macht logische Tests, Stress- und Performance-Tests
  $\Rightarrow$ Zertifikat

OPC - Spezifikationen

früher

OPC DA (Data Access): Übertragung von Echtzeitwerten (nutzt Microsoft DCOM)

OPC A/E (Alarms and Events): Übertragung von Alarmen und Ereignissen

OPC HDA (Historical Data Access): Übertragung historischer Werte

OPC DX (Data eXchange): direkte Kommunikation zwischen OPC Servern.

OPC Command Ausführung von Befehlen & Kommandos

OPC XML DA XML-basierte Übertragung von Echtzeitwerten. Vorläufer von OPC UA

---

seit 2011

OPC UA (Unified Architecture): plattform- und DCOM-unabhängige Realisierung aller bisherigen Spezifikationen. IEC-Norm (IEC 62541)

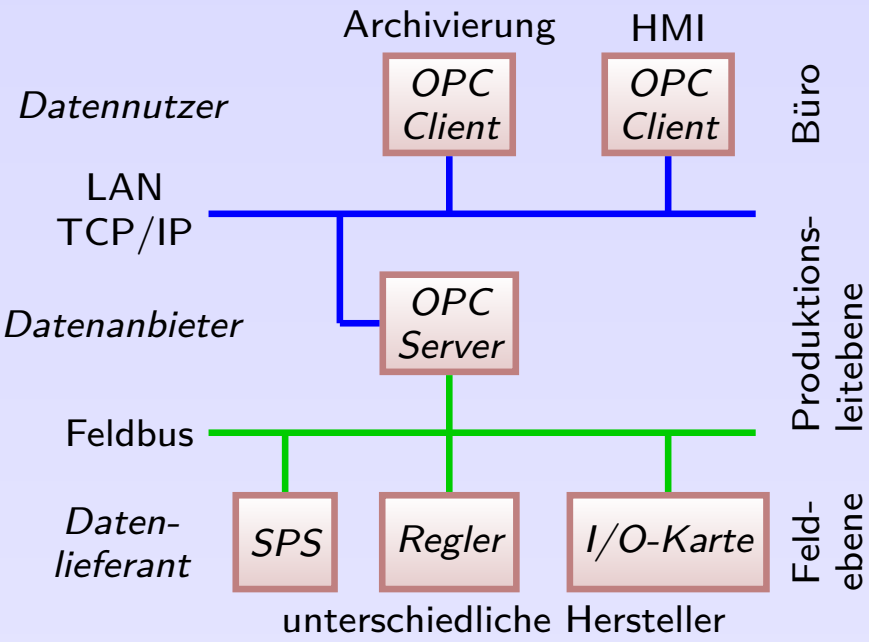
OPC - Ziele, Funktionen, Struktur

Ziele

Prozessvisualisierung: Überwachung

Prozesssteuerung: Schalten, Führungsgrößen, Sollwerte

Betriebsführung: Auftragswesen, Qualitätskontrolle, Instandhaltung, Materialverwaltung, Produktionsplanung



Funktionen

- Überwachung von Echzeitdaten
- Datenarchivierung
- Alarm-Meldungen
- Befehlsübermittlung (Steuerung)

Struktur

- Basiert auf (Multi-)Server/(Multi-)Client-Architektur
- Client initiiert Anfrage: Lesen, Schreiben Überwachen
- Server liefert Antwort/führt Befehl aus
- Client und Server i.d.R. auf unterschiedlicher Hardware, z.B. Server in SPS integriert, Client im Büro-PC

OPC - Server

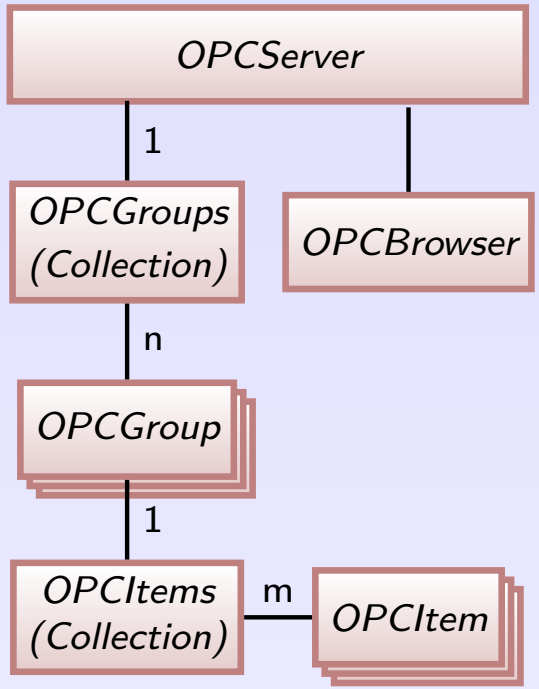
- Software-Applikation, welche als API (Application Programming Interface) oder als Protokoll-Konverter funktioniert.
- Übersetzt Daten aus Feldebene in standardisiertes OPC-Format
- Name: lesbare Name (ProgId) + eindeutiger Identifier (ClassId, 32 Hex-Ziffern)
- Namensraum: Server enthält meist nur Teil des SPS-Adressraumes, Auswahl durch Projektierungstool (Variablenauswahl bei SimaticNET mit Symboldatei-Konfigurator)

Objekthierarchie

OPCItem entspricht Prozessvariable (E, A, M), eindeutige ItemID

OPCGroup fasst mehrere Items logisch zusammen

OPCServer Oberstes OPC-Objekt (root), Verwaltung der OPC-Groups, ermöglicht z.B. Suche nach erreichbaren Variablen



OPC-Client

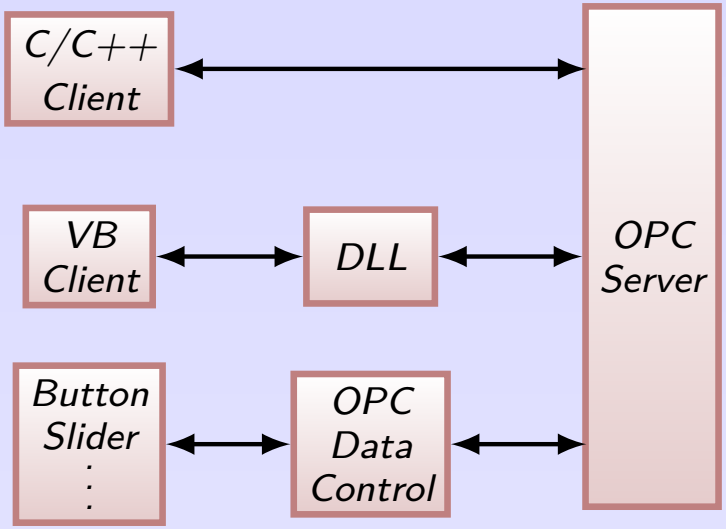
- OPC konforme Applikation (z.B. HMI) verbindet mit OPC-Server zum Lesen, Schreiben und Überwachen von Device-Daten
- Initiiert OPC-Kommunikation
- Lese- oder Schreibanforderungen für Prozesswerte
- Überwachen: Client überträgt Überwachung auf Server. Meldung bei Wertänderung
- Synchrones Lesen: Client hält an, bis Server Wert geliefert hat
- Asynchrones Lesen: Client erhält Leseauftragsbestätigung und arbeitet weiter. Wert wird durch Ereignismeldung nachgeliefert
- Datenpaket: Wert + Status + Zeitstempel

OPC-Interfaces

Custom Interface: für Hochsprachen mit Funktionszeigern wie C/C++

Automation Interface: für Hochsprachen die mittels Objektnamen zugreifen wie Visual Basic, benötigen DLL vom OPC-Server-Hersteller als Wrapper

Schnittstelle mit OPC Data Control: Bedien- und Anzeige-Elemente von HMI mit Variablen verknüpft, z.B. mit SimaticNET



Literatur/Vertiefung

- [WZ09, Kap. 24]
- OPC-UO Server und Client Library: https://freeopcua.github.io/
- https://de.wikipedia.org/wiki/Open_Platform_Communications

7. Sicherheitsaspekte

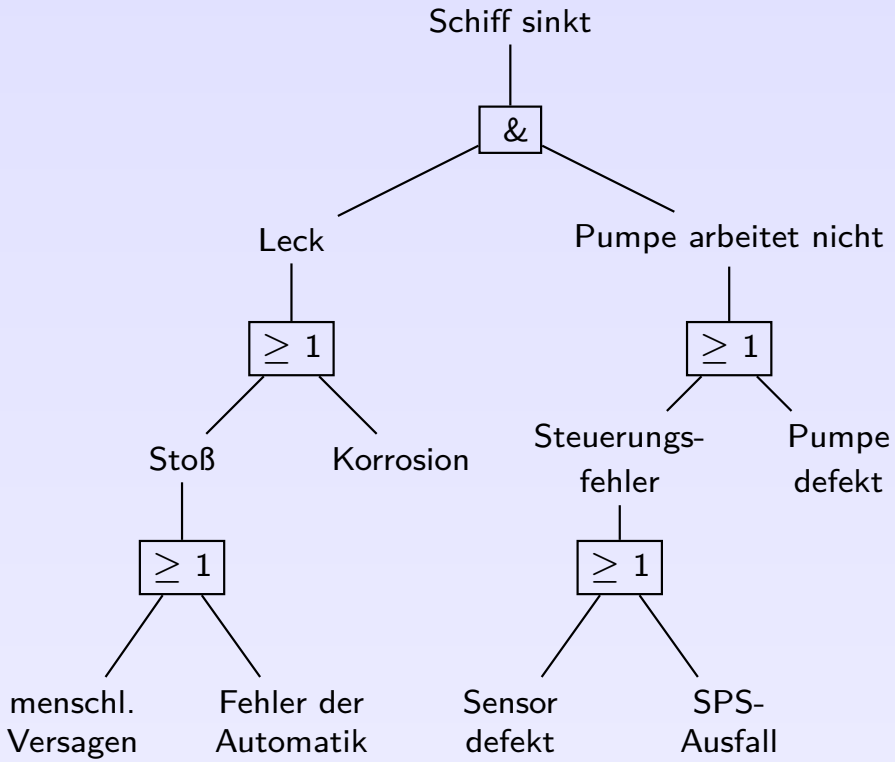
Sicherheit

- Security = Angriffssicherheit, Schutz des Objektes vor der Umgebung
- Safety = Betriebssicherheit, Schutz der Umgebung vor einem Objekt
- Unterschiedliche Definitionen, z.B. IEC 61508: Freiheit von unvertretbaren Risiken
- Aspekte: Arbeitssicherheit, Umweltschutz, Produktsicherheit (Nahrungsmittel), ...
- Grundlage für Betriebssicherheit: Bauteilzuverlässigkeit (mtbf)
- Sicherheitsgerichtete Softwareentwicklungsprozesse in Normen z.B. für Lufttahrt
- Unmittelbare Sicherheit: Verhinderung von Gefahrentstehung (Sichere Betriebszustände, Redundanz)
- Mittelbare Sicherheit: Schutzeinrichtungen zur Abweisung von Gefährdung (Verkleidungen, Lichtschranken, Zweihandbedienung, Tipptaster)
- Hinweisende Sicherheit: Gefahrensymbole, Warnhinweise, Lichtzeichen, Signalfarben

Fehlerbaumanalyse

= Fault Tree Analysis (FTA)

- Zuverlässigkeitsanalyse von techn. Anlagen und Systemen
- Wahrscheinlichkeit eines Ausfalls
- Unterteilung des Gesamtsystems in Minimalschritte
- Logische Verknüpfung von Teilsystemausfällen auf allen kritischen Pfaden
- Ausgehend vom fehlerhaften Resultat schließt man auf mögliche verursachende Ereignisse
- Komplexe FTA mit speziellen Softwarepaketen



Risikoanalyse

Das mit einem Fehler verbundene Risiko ist

$$\text{Risiko} = \text{Schadensausmaß} \cdot \text{Auftrittshäufigkeit} \tag{1}$$

Exakte, rechnerische Ermittlung meist nicht möglich, daher Risikograf (IEC 61508) mit

- Risikoparameter: GASE
  - G Möglichkeiten zur Gefahrenabwendung
  - A Aufenthaltsdauer von Personen im Gefahrenbereich
  - S Schadensausmaß
  - E Eintrittswahrscheinlichkeit eines Fehlers
- PL - Performance Level (a. . . e) für Maschinensicherheit
- SIL - Safety Integrity Level (SIL1. . . SIL4) für Anlagensicherheit

SIL und PL sind Maße für Zuverlässigkeit

Ferngesteuerte Kräne: SIL1
Pressen mit Einlegearbeiten: SIL3
Reaktorschutzsysteme: SIL4

Risikograf

Schadensausmaß

S1 leichte Verletzung
S2 Tod einer Person
S3 Tod mehrerer Personen
S4 Katastrophe

Aufenthaltsdauer von Personen

A1 selten
A2 häufig

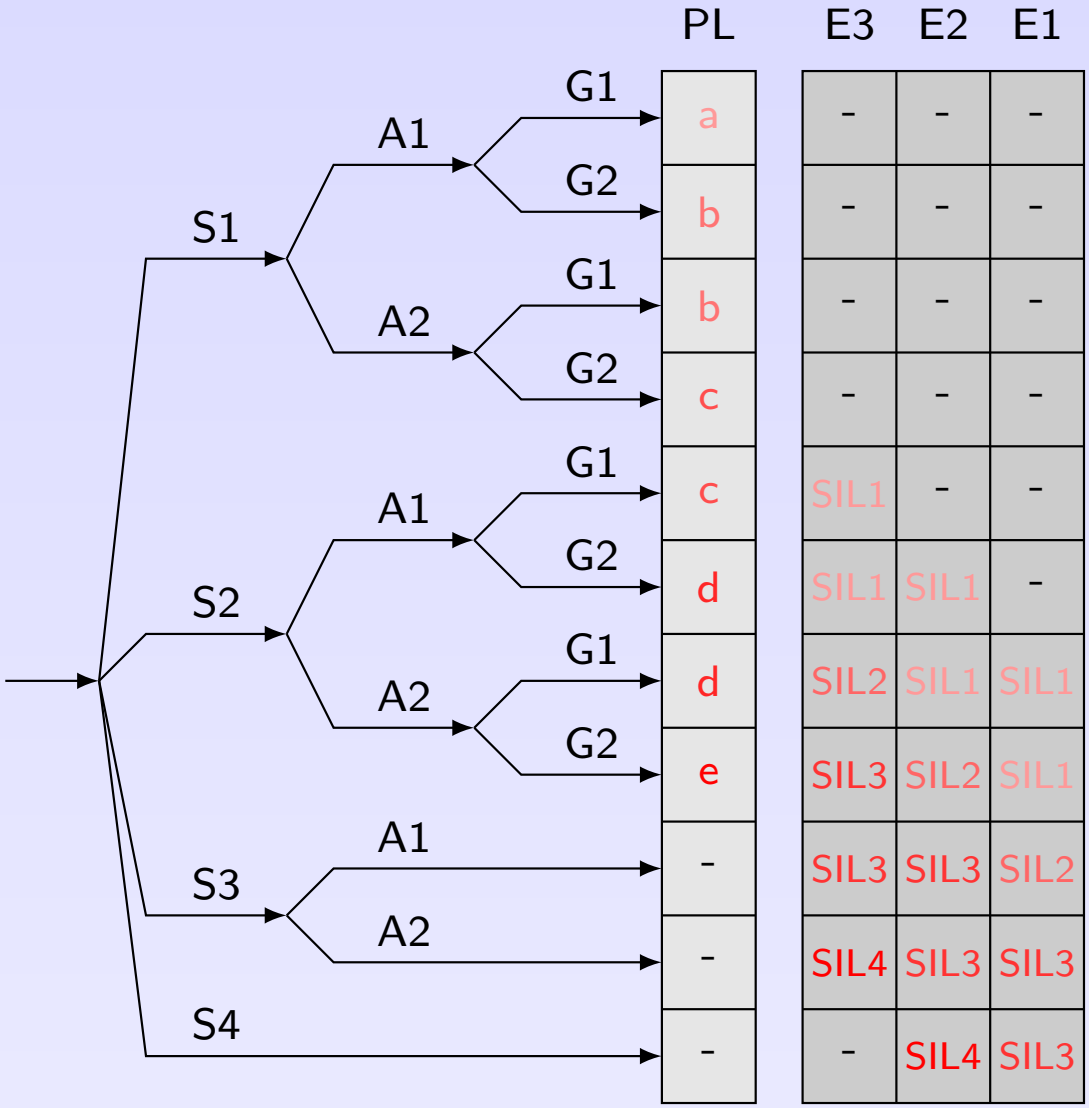
Gefahrenabwendung

G1 möglich
G2 kaum möglich

Eintrittswahrscheinlichkeit

E1 sehr gering
E2 gering
E3 relativ hoch

Beispiel:
youtu.be/iVftIfCTc5w



[Sei15, S. 203]

7. Sicherheitsaspekte

Gegenmaßnahmen I

Fehlervermeidung: Maßnahmen bis zur Inbetriebnahme

- Fehler in Programmierung, Fertigung oder Design
- Fehler können erkannt und behoben werden

Fehlerbeherrschung: Maßnahmen während des Betriebes

- Prozessfehler (Überdruck), Benutzerfehler, Komponentenfehler (Sensorausfall)
- Von Fehler darf keine Gefahr ausgehen
- Im Fehlerfall wird Sicherheitsfunktion ausgeführt (sicherer Zustand, Umschalten auf Redundante Komponenten)

PFH: Probability of Failure per Hour

- Ausfallrate der Steuerung

PFD: Probability of Failure on Demand

- Ausfallwahrscheinlichkeit bei Ausführung einer Sicherheitsfunktion

SIL	1	2	3	4
PFD	10^{-1}	10^{-2}	10^{-3}	10^{-4}
PFH	10^{-5}	10^{-6}	10^{-7}	10^{-8}
Einsatz von	bewährte Standardgeräte	einkanalige SSPS	zweikanalige SSPS	bauteilsichere VPS

Gegenmaßnahmen II

Beispiel [Sei15, S.204]: Brennersteuerung eines Gaskraftwerks
Bei Erlöschen der Brennerflamme hohe Explosionsgefahr durch Gas

Schadensausmaß: Tod mehrerer Personen ___
Aufenthaltsdauer im Brennerraum: gering ___
Möglichkeiten zur Gefahrenabwendung: keine ___
Eintritsswahrscheinlichkeit: relativ hoch ___
SIL = ___

Vorgeschriebener Einsatz von: _____
SPS macht umfangreiche Selbsttests, um Erkennung gefährlicher Betriebszustände sicherzustellen
Ausfallwahrscheinlichkeit im Bedarfsfall (worst case):

$$\mathrm{PFD}_{\mathrm{ges}} = \mathrm{PFD}_{\mathrm{Sensor}} + \mathrm{PFD}_{\mathrm{SPS}} + \mathrm{PFD}_{\mathrm{Aktor}} \overset{!}{<} \text{___}$$

Falls $\mathrm{PFD}_{\mathrm{Sensor}}$ zu groß: parallele, redundante Sensoren
Falls $\mathrm{PFD}_{\mathrm{Aktor}}$ zu groß: redundante Aktoren in Reihe

Sicherheitsgerichtete Steuerung

- SSPS: Anforderungen bis SIL3
- Fail-Safe-Prinzip (trotz Fehler sicher)
- Beispiele
  - Signale bei Bahn $\in$\{Halt,Fahrt\}, Fehlerfall $\Rightarrow$ Halt=Sicher
  - Bremse steht unter Druck, um nicht zu bremsen; Druckabfall $\Rightarrow$ Halt=Sicher
  - Not-Aus als Öffner; Kabelbruch $\Rightarrow$ Aus=Sicher
- SSPS muss auch im Fehlerfall gefährlichen Zustand verhindern
- Verfügbarkeit der Sicherheitsfunktion $\neq$ Betriebsfunktion
- Prinzipien: Redundanz (Mehrzahl) und Diversität (Vielfalt)
- Aktive Redundanz: Funktion wird auf mehreren Systemen parallel ausgeführt. Aktor wird nur angesteuert, wenn Ergebnisse identisch. Im Fehlerfall geht Aktor in sicheren/energielosen Zustand
- Passive Redundanz: Funktion wird von Master ausgeführt. Im Fehlerfall übernimmt Slave die Funktion. Steuerung bleibt verfügbar
- Physikalische Diversität: unterschiedliche Sensorhersteller oder Messprinzipien für die selbe Größe
- Implementierungsdiversität:
  Unterschiedliche Algorithmen, komplementäre Speicherung und Diversitäre Operatoren, z.B. $Y = X_1 \wedge X_2$ und $Y' = \overline{X_1} \vee \overline{X_2}$ $Q := Y$, falls $Y = \overline{Y'}$, sonst $Q := 0$ (Implementierung oft zeitredundant)

Notausschalter

- Bei Gefahr $\rightarrow$ sicherer Zustand
- Kein Selbstanlauf nach Entriegelung
- Prinzipien: Einrasten, Öffner (Ruhestrom), z. T. Schlüsselschalter
- Signalfarbe: Rotes Betätigungselement auf gelbem Grund
- Erreichbarkeit: im Notfall unmittelbare Betätigung vom Bediener
- Bauarten: Pilztaster, Reißleine, Sicherheitslichtschranke, ...
- Anlage mit mehreren Komponenten (Antrieb + Laser): Jeder Notaus schaltet Alles ab

Stopp-Kategorien

0 Sofortige Trennung der Energiezufuhr von Antrieben
1 Gesteuertes Stillsetzen, Maschine in sicheren Zustand bringen, danach Energie abschalten
2 Maschine in sicheren Zustand bringen, Energie nicht trennen

Unterscheide

Notaus: gesamte Anlage zum Schutz gegen elektrischen Schlag spannungsfrei

Not-Halt: Antriebe zum Schutz vor Gefährdung abgeschaltet, Anlagenteile (SPS, Kühlung, Sicherheitstechnik) nicht spannungsfrei

Zwangsgeführte Kontakte sind so verbunden, dass Öffner und Schließer niemals gleichzeitig geschlossen sind