

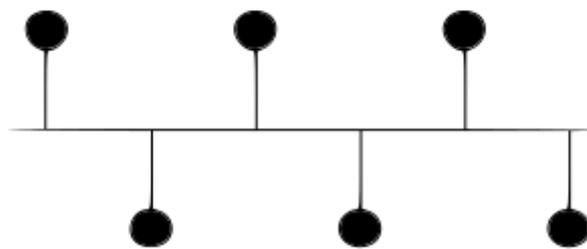
Netzwerk

OSI

 OSI Schichtenmodel mit Beispielen

Topologien

Bus:

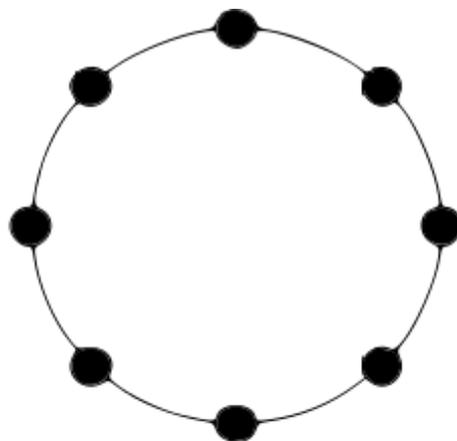


Buslogopogie

- Anzahl zerstörter Verbindungen = 1 (nur Ausgefallener Knoten ist betroffen)
- minimalem Aufwand beim Anschluss neuer Knoten
- zunehmende Kollision bei Überlastung

Eigenschaft	Wert
Anschlüsse	1
Anzahl zerstörter Verbindungen	1 (nur Ausgefallener Knoten Betroffen)
max. Entfernung	$n-1$

Ring



Ringlogopogie

- einfach und kostengünstig

Eigenschaft	Wert
Anschlüsse	2
Anzahl zerstörter Verbindungen	2
max. Entfernung	$\frac{n}{2}$

Kette

- einfach und kostengünstig

Eigenschaft	Wert
Anschlüsse	2
Anzahl zerstörter Verbindungen	1
max. Entfernung	$n-1$

Barrel Shifter:

- Erweiterung des Rings um zusätzliche Verbindungen
- Überspringen eines oder mehrerer Knoten möglich
- Anzahl von Anschlüssen einer Station wird erhöht
- verringert max. Entfernung
- erhöht die Redundanz durch alternative Wege

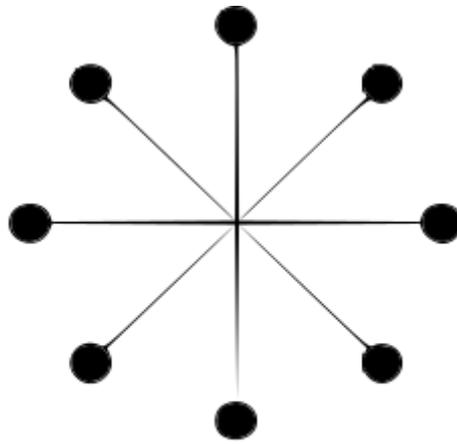
Eigenschaft	Wert
Anschlüsse	
Anzahl zerstörter Verbindungen	$2 \cdot \log_2(n) - 1$
max. Entfernung	

Baumtopologie:

- geeignet für Broadcastartige Kommunikation (von der Wurzel aus)
- Engpass ist der Pfad über die Wurzel

Eigenschaft	Wert
Anschlüsse	
Anzahl zerstörter Verbindungen	1
max. Entfernung	$\log(n)$

Sterntopologie



Sterntopologie

- Spezialfall eines einstufigen Baums
- Große Anzahl von Anschlüssen an der Zentralstation
- Hauptlast liegt auf der Zentralstation
- Zentralstation = Single Point of Failure (Ausfall = Totalausfall)
- Anschluss neuer Knoten sorgt für höhere Last auf der Zentralstation

Eigenschaft	Wert
Anschlüsse	1 (Zentralstation = n)
Anzahl zerstörter Verbindungen	1
max. Entfernung	2

2D-Gitter:

- Jeder Knoten ist mit 4 Nachbarn verbunden
- leicht Skalierbar

Eigenschaft	Wert
Anschlüsse	
Anzahl zerstörter Verbindungen	2 (beim Eckknoten)
max. Entfernung	\sqrt{n}

3D-Torus:

- 6 Anschlüsse pro Station

Eigenschaft	Wert
Anschlüsse	6
Anzahl zerstörter Verbindungen	$2 \cdot n^{\frac{2}{3}}$
max. Entfernung	$\sqrt[3]{n}$

Vollvermascht:

- Jede Station ist mit jeder Station verbunden
- sehr hoher Hardwareaufwand
- nur bei kleiner Anzahl von Stationen realisierbar

Eigenschaft	Wert
Anschlüsse	n
Anzahl zerstörter Verbindungen	$n-1$
max. Entfernung	1

Bussysteme

LIN (Local Interconnect Network)

- geringe Anforderungen gegenüber CAN
- kostengünstig
- geringe Datenrate
- Einsatz: Sitze, Türen, Dach, Klimaanlage, Steuereinheiten (Motorsteuerung)

Eigenschaft	Wert
Übertragungsmedium	
Übertragungsrate	1-20 kbit/s
Teilnehmer	max. 16 Knoten
Daten/Nachricht:	1-8 Byte (ab Version 2.0 oder höher)
Busmanagement:	Master/Slave (Zeitslots, deterministisch)
Einsatzgebiet	KFZ (Sitze, Türen, Dach, Klimaanlagen)

CAN (Controller Area Network)

- vernetzung komplexer Controller und Steuergeräte
- Echtzeitfähig (verbesserung in TTCAN)
- preisgünstig aufgrund von hoher Stückzahl
- max. Übertragungsrate in der Praxis effektiv 500kBit/s
- max. Teilnehmer kann durch Aufteilung in Segmenten erhöht werden

Eigenschaft	Wert
Übertragungsmedium	verdrillte Zweidrahtleitung
Übertragungsrate	10kbit/s (bei wenigen Kilometern) - 1MBit/s (40m)
Teilnehmer	max. 32 Knoten
Daten/Nachricht:	max. 8 Byte
Busmanagement:	serieller Multi-Master-Bus
Buszugriffsverfahren:	CSMA/CA
Einsatzgebiet	KFZ, Haushaltsgeräte, Medizintechnik

LON (Local Operating Network)

-

Eigenschaft	Wert
Übertragungsmedium	verdrillte Zweidrahtleitung und Funk
Übertragungsrate	4,9 kBit/s (Funk 49 MHz) - 1.25 MBit/s (Zweidraht 500m)
Teilnehmer	max. 32000 (127 Endpunkte pro Subnetz, 255 Subnetze)
Daten/Nachricht:	
Busmanagement:	
Buszugriffsverfahren:	CSMA/CD
Einsatzgebiet	verteilte industrielle Anwendungen, Gebäudeautomatisierung

EIB (European Installation Bus)

-

Eigenschaft	Wert
Übertragungsmedium	
Übertragungsrate	
Teilnehmer	
Daten/Nachricht:	
Busmanagement:	
Buszugriffsverfahren:	
Einsatzgebiet	

Profibus (Process Field Bus)

-

Eigenschaft	Wert
Übertragungsmedium	
Übertragungsrate	
Teilnehmer	
Daten/Nachricht:	
Busmanagement:	
Buszugriffsverfahren:	
Einsatzgebiet	

FlexRay

-

Eigenschaft	Wert
Übertragungsmedium	
Übertragungsrate	
Teilnehmer	
Daten/Nachricht:	
Busmanagement:	
Buszugriffsverfahren:	
Einsatzgebiet	

MOST (Media Oriented System Transport)

-

Eigenschaft	Wert
Übertragungsmedium	
Übertragungsrate	
Teilnehmer	
Daten/Nachricht:	
Busmanagement:	
Buszugriffsverfahren:	
Einsatzgebiet	

Echtzeitsysteme

Echtzeitsysteme sind also Systeme, die korrekte Reaktionen innerhalb einer definierten Zeitspanne produzieren müssen. Falls die Reaktionen diese Zeitlimits überschreiten, führt dies zu Leistungseinbußen und/oder Fehlfunktionen.

Unterteilung

- **harte Echtzeitsysteme**
 - Katastrophale Folgen beim Verpassen einer Deadline
 - Deadline muss zwingend eingehalten werden
 - bsp. flight control system (Flugzeugsteuerung)
 - Zeitliche Bindung die vom System stets erfüllt werden muss, da eine auch nur gelegentliche Verletzung erhebliche Folgen nach sich ziehen würde.
- **weiche Echtzeitsysteme**
 - Einhaltung von Deadlines wichtig aber nicht notwendig zur korrekten funktionsweise.
 - zeitliche Bedingung bei der gelegentliche Verletzungen tolerierbar sind
 - bsp. Multimediasysteme (Darstellung 25fps -> verletzung meist tollerierbar)
- **oberen Zeitschranken** -> maximal erlaubte Zeitspanne
 - Mindestgeschwindigkeit des Systems oder Teilen wird vorausgesetzt
 - erreicherung durch konstruktive Maßnahmen (aufwändig)
- **untere Zeitschranke** -> minimale dauer einer Phase
 - anforderung kann durch warten erfüllt werden
- **beidseitige Zeitschranken**

Unterscheidung

- **Ereignis gesteuert** (event triggered)
 - Auslösung eines Interrupts bei auftreten eines Ereignisses (der Umgebung)
 - resultiert in kurzen Reaktionszeiten
 - anfällig bei der Verarbeitung vieler gleichzeitiger Ereignissen (sogennante **event showers**)
- **Zeit gesteuert** (time triggered)
 - Auslösung durch periodische Zeitgeber
 - Aktives abfragen von Sensoren durch das Steuergerät (Polling)

- ggf. puffern kurzer Signale
- planbares zeitliches Verhalten sämtlicher Systemaktivitäten
- Prüfbar ob Echtzeitanforderungen eingehalten werden können.

Echtzeitbetriebssysteme

Ein **Betriebssystem** nach **DIN44300 sind die Programme eines digitalen Rechensystems, die zusammen mit den Eigenschaften dieser Rechenanlage die Basis der möglichen Betriebsarten des digitalen Rechensystems bilden und die insbesondere die Abwicklung von Programmen steuern und überwachen.

Komponente	Beispiele
Prozess-Verwaltung	Kreieren und Terminieren von Applikations- und System-Prozessen, Suspendieren und Reaktivieren von Prozessen, Prozess-Synchronisation und Kommunikation, Deadlock-Behandlung
Speicher-Verwaltung	„Buchführung“ über freie und belegte Hauptspeicherbereiche, Kommunikation mit den in der Hierarchie angrenzenden Speichermedien, Dynamische Allokation und Deallokation von Hauptspeicher, Speicherschutz und Zugriffskontrolle
Prozessor-Verwaltung	Multiprogramming, Dispatching, Zuteilungsalgorithmen (Scheduling), Vorrangunterbrechung (Interrupt, Preemption)
Geräte-Verwaltung	
Datei-Verwaltung	Datei-Konzepte, -Attribute und -Operationen, Zugriffs-Methoden, Verzeichnis-Strukturen und Implementierungen, Allokations-Methoden

Definition (**Echtzeitbetrieb**, DIN 44300) (engl. **Real Time Operating System**, kurz: **RTOS**) ist ein Betrieb eines Rechensystems, bei dem Programme zur Verarbeitung anfallender Daten ständig betriebsbereit sind derart, dass die Verarbeitungsergebnisse innerhalb einer vorgegebenen Zeitspanne verfügbar sind. Die Daten können je nach Anwendungsfall nach einer zeitlich zufälligen Verteilung oder zu vorbestimmten Zeitpunkten anfallen.

Anforderungen:

- deterministisches zeitliches Verhalten -> Reihenfolge der Verarbeitungsschritte liegt eindeutig und wiederholbar fest
- Bearbeitung vieler externer Ereignisse mithilfe eines Unterbrechungskonzepts
- definierte Antwortzeiten einer Unterbrechung
- schnelle Reaktion -> geringer Overhead bei direkten Zugriffen oder Kontextwechsel
- Multitaskingfähigkeit (Vergabe des Prozessors an andere Prozesse bei Interrupts oder Timerablauf)
- prioritätsgesteuerter anlauf von Programmen
- effiziente und schnelle Interprozesskommunikation
- Einsatz in Systemen mit begrenzten Ressourcen (durch Skalierbarkeit)
- Lastunabhängigkeit muss garantiert sein
- Verwendung von Standardschnittstellen
- Kommunikation mit nicht Echtzeitsystemen
- Verfügbarkeit für verschiedene Prozessorarchitekturen
- optimierte Werkzeuge zur Entwicklung von Echtzeitanwendungen

Embedded Systems

Ein **verteiltes System** besteht aus Komponenten, die räumlich oder logisch verteilt sind und mittels einer Kopplung bzw. Vernetzung zum Erreichen der Funktionalität des Gesamtsystems beitragen.

Ein **Steuergerät** (engl. Electronic Control Unit, ECU) ist die physikalische Umsetzung eines eingebetteten Systems. In mechatronischen Systemen bilden Steuergeräte und Sensorik/Aktuatorik oft eine Einheit.

Wird Elektronik zur Steuerung und Regelung mechanischer Vorgänge räumlich eng mit den mechanischen Systembestandteilen verbunden, so sprechen wir von einem **mechatronischen System**. Der Forschungsbereich der sich mit der Entwicklung mechatronischer Systeme befasst nennt sich **Mechatronik**.

Klassifikation und Charakteristika

Technische Ausprägung

- **kontinuierliche** Systeme
- **diskreten** Systeme
- **verteilten** Systeme
- **monolithischen** Systeme
- **hybriden** Systeme (sowohl kontinuierlich als auch diskretes Verhalten)

Systeme die sowohl kontinuierliche (analoge), als auch diskrete Datenteile (wertkontinuierlich) verarbeiten und/oder sowohl über kontinuierliche Zeiträume (zeitkontinuierlich), als auch zu diskreten Zeitpunkten mit ihrer Umgebung interagieren, heißen **hybrid Systeme**.

Sicherheitsrelevanz

- **Sicherheitskritische** Systeme (wenn Menschenleben oder die Unversehrtheit von Einrichtungen abhängt)
z.B. Avionik, Medizintechnik und Kraftfahrzeugbereich
- **nicht Sicherheitskritische** Systeme
z.B. Konsumelektronik hauptsächlich nicht Sicherheitskritische Systeme
- **Zeitkritische** Systeme
- **nicht Zeitkritische** Systeme

Produktkategorien

- Telekommunikation (Telefon, Fax, etc.)
- Haushalt (Waschmaschine, Mikrowelle, Fernseher, etc.)
- Periphere Geräte (Tastatur, Modem, Drucker, etc.)
- Bürotechnik (Kopierer, Schreibmaschine etc.)
- Geräte für Freizeit, Hobby und Garten
- Automobiltechnik (ABS, Wegfahrsperr, Navigationssysteme, etc.)
 - Massenmarkt für eingebettete Systeme

- häufig Zeit- und zunehmend auch sicherheitskritisch
- Öffentlicher Verkehr (Fahrkartenautomat, etc.)
- Luft- und Raumfahrttechnik
- Fertigungstechnik
- Steuerungs- und Regelungstechnik, Medizintechnik, Umwelttechnik, Militärtechnik
- Avionik (Flugzeugbau)
- uvm

Klassifikation

Transformationelle Systeme

- Eingaben müssen zu Beginn der Systemverarbeitung vollständig vorliegen
- Ausgaben nur verfügbar wenn die Verarbeitung vollständig terminiert
- Keine Interaktion während der Verarbeitung möglich (=> kein Einfluss auf die Ergebnisse)

Interaktive Systeme

- Ausgabe nicht nur bei Terminierung
- Interaktion und Synchronisierung mit der Umgebung
- Interaktion wird durch das System bestimmt
- proaktive Synchronisierung durch das System

reaktive Systeme

Ein reaktives System kann aus Software und/oder Hardware bestehen und setzt Eingabeereignisse deren zeitliches Auftreten meist nicht vorhergesagt werden kann - oftmals aber nicht notwendigerweise unter Einhaltung von Zeitvorgaben - in Ausgabeereignisse um.

- Synchronisierung mit Systemumgebung
- Reagieren auf ihre Umwelt
- Arbeiten häufig Nebenläufig
- müssen sehr zuverlässig sein
- müssen Zeitschranken einhalten (Echtzeitsysteme)
- sind in Hardware als auch Software realisiert
- werden in komplexen, verteilten Systemplattformen implementiert
- Die funktionale Korrektheit ist ein wichtiger Faktor der Entwicklung
- eingebettet in komplexe bsp. mechanische, chemische oder biologische Systemumgebung => eingebettete Systeme

Bestandteile

- Kontrolleinheit
- Regelstrecke
- Benutzerschnittstelle
- technische Systemumgebung
- menschliche Systembenutzer

Definitionen

Als **Steuergerät** (engl. **Electronic Control Unit, ECU**) wird die eigentliche Steuereinheit eines mechatronischen Systems verstanden.

Steuergeräte sind im Prinzip wie folgt aufgebaut: Die Kernkomponente des Steuergeräts stellt ein Mikrocontroller oder Mikroprozessor (Beispiele: Power PC, Alpha, PC) dar. Zusätzlich kann es optional ein externes RAM und/oder ROM besitzen sowie sonstige Peripherie und Bauelemente.

Mikrocontroller kennzeichnen eine Klasse von Mikroprozessoren, die auf den speziellen Anwendungsbereich der Steuerung von Prozessen zugeschnitten sind. Wir betrachten im Folgenden einige Spezialfälle genauer.

Ein **ASIP (applikationsspezifischer Prozessor)** ist ein Prozessor, der von seiner Struktur als auch von seinem Befehlssatz her auf seinen Einsatz für bestimmte Anwendungen hin optimiert ist. Er besitzt spezielle Instruktionssätze, funktionale Einheiten, Register und spezielle Verbindungsstrukturen.

- kostengünstiger aufgrund abgespekter Prozessoren
- aufgrund Programmierbarkeit flexibel
- höhere Verarbeitungsgeschwindigkeit und geringere Leistungsaufnahme aufgrund optimierter Strukturen
- **Nachteil** = komplizierte und aufwendige Entwicklung

Ein **DSP** ist ein spezieller Mikroprozessor, der Befehle (und damit Verarbeitungseinheiten) zur Durchführung von Signal-Verarbeitungs-Aufgaben (bsp. Fast Fourier Transformation, **FFT**) besitzt.

- Optimiert auf die häufig vorkommenden Operationen bei der digitalen Signalverarbeitung (bsp. schnelle Multiplikation)
- hoher Stellenwert ist die Effizienz dieser Operationen
- **Einsatz** bsp. MP3 decoder oder Sprachsignalverarbeitung oder Bildverarbeitung

Das **Field Programmable Gate Array (FPGA)** ist ein komplexer, programmierbarer Logikbaustein, der zum Aufbau digitaler, logischer Schaltungen dient. Er besteht im Wesentlichen aus einzelnen Funktionsblöcken, die in einer regelmäßigen Struktur (**Array**) angeordnet sind, und einen Netzwerk von Verbindungen zwischen diesen Blöcken. Bei FPGAs wird die Implementierung von logischen Funktionen hauptsächlich durch die Programmierung der Verbindungsleitungen zwischen den Logikblöcken erreicht.

- zwei Arten von FPGAs
 - rekonfigurierbare (unter Verwendung von Speichertechnologien wie SRAM)
 - Nachteil: sie sind flüchtig
 - nicht rekonfigurierbare (einmal Konfiguriert immer Konfiguriert, Umsetzung durch physikalische Zerstörung der nicht benötigten Verbindungsleitungen)
- Realisierung von Speicherzellen möglich
- Eignung zur Realisierung von Steuerwerken (in Form endlicher Automaten)
- Im Gegensatz zu gewöhnlichen **Gate Arrays (GA)** sind FPGAs programmierbare Logikbausteine, deren Funktionalität durch das Zusammenschalten verschiedener Funktionsblöcke erreicht wird

