

Echtzeitsysteme

Einleitung

Sommersemester 2024

Peter Wägemann

Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU)

Lehrstuhl Informatik 4 (Systemsoftware)

<https://sys.cs.fau.de>



Lehrstuhl für Informatik 4
Systemsoftware



Friedrich-Alexander-Universität
Technische Fakultät

Das erste Echtzeitrechensystem

■ Whirlwind I

- Zweck: Flugsimulator
(Ausbildung von Bomberbesatzungen)
- Auftraggeber: U.S. Navy
- Auftragnehmer: MIT
- Laufzeit: 1945 – 1952

■ Technische Daten

- Digitalrechner, bit-parallele Operationen
- 5000 Röhren, 11000 Halbleiterdioden
- magnetischer Kernspeicher
- Röhrenmonitore mit Lichtgriffel

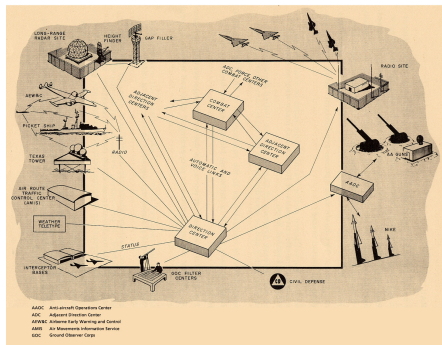
Spätere Nutzung in SAGE durch die U.S. Air Force



(Quelle: Alex Handy from Oakland, Nmibia)

SAGE – Semi-Automatic Ground Environment

- Erstes verteiltes Echtzeitrechensystem
- Automatisiertes Kontroll- und Abwehrsystem gegen Bomber



- 27 Installationen
 - verteilt über die USA
 - *Nonstop*-Betrieb
- Kopplung durch Datenfernleitungen
 - Telefonleitungen
 - Internet-„Mutter“
- pro Installation...
 - 100 Konsolen
 - 500 KLOC Assembler

- Entwicklung eines leistungsfähigeren Nachfolgers: Whirlwind II

- Der Nachfolger AN/FSQ-7 alias „Whirlwind II“:



(Quelle: Steve Jurvetson from Menlo Park, USA)

← SAGE Bedienstation

- Technische Daten

- Auftraggeber: U.S. Air Force
- Auftragnehmer: MIT, später IBM
- Bauweise: 55000 Röhren, 2000 m^2 , 275 t, 3 MW, 75 KIPS

- Betriebsdaten von SAGE:

- Installation: 22 - 23 Stationen im Zeitraum 1959 - 1963
- Betrieb: bis 1983 (Whirlwind I bis 1979)
- Kosten: 8–12 Milliarden \$ (1964) \rightsquigarrow ca. 97 Milliarden \$ (2019)
- Nachfolger: u.a. AWACS

Zivile & humanitäre Einsatzzwecke



Problematik der Doppelverwendungsfähigkeit

- Zahlreiche humanitäre Echtzeitsysteme
- Übersicht über einige Einsatzszenarien [6]
- Beispiel: Transport von Medikamenten in entlegene Regionen [5]

Moderne Echtzeitsysteme

Wo immer Rechensysteme mit ihrer physikalischen Umwelt interagieren:



CAN CLASS B

- 1) SAK/SRB Fahrer
- 2) SAK/SRB Beifahrer
- 3) SAK/SRB Heck 1
- 4) SAK/SRB Heck 2
- 5) Sitzsteuergerät Fahrer
- 6) Sitzsteuergerät Beifahrer
- 7) Sitzsteuergerät hinten links
- 8) Sitzsteuergerät hinten rechts
- 9) Türsteuergerät vorne Fahrerseite
- 10) Türsteuergerät vorne Beifahrerseite
- 11) Türsteuergerät hinten Fahrerseite
- 12) Türsteuergerät hinten Beifahrerseite
- 13) Steuergerät Trennwand
- 14) Dachdeckerentlast
- 15) Dachknoten Mitte (DKM)
- 16) Vorderes-Bedien-Feld (VBF)
- 17) Hinteres-Bedien-Feld (HBF)
- 18) Elektronisches Zündschloss (EZS)
- 19) Kombiinstrument
- 20) Mantelrohrmodul
- 21) Frontklimaheizung
- 22) Fondklimaheizung
- 23) Audiogateway

- 24) Parktronicystem (PTB)
- 25) Reflexfunktionskontrolle (RFK)
- 26) Pneumatische Steuerungseinheit (PSE)
- 27) Heckdeckschließung-Öffnung
- 28) Zentrales Gateway
- 29) Elektronisches Wählbehelmodul
- 30) Airbag-SG (Armada)
- 31) Multifunktionssteuergerät (MSS)
- 32) Bordnetz-Steuergerät
- 33) Wandler Lenkradheizung
- 34) Standheizung
- 35) Türzuziehung hinten Fahrerseite
- 36) Türzuziehung hinten Beifahrerseite

CAN CLASS C

- 37) Elektronisches Zündschloss (EZS)
- 38) Kombiinstrument
- 39) Mantelrohrmodul
- 40) Zentrales Gateway
- 41) Elektronisches Wählbehelmodul
- 42) Luftleitung (SLF)
- 43) Diatronic (DTR)
- 44) Lichtverleerregulierung
- 45) Motor Elektronik (ME)
- 46) Sensotronic Brake System (FSG)
- 47) Elektronische Getriebe-Steuerung

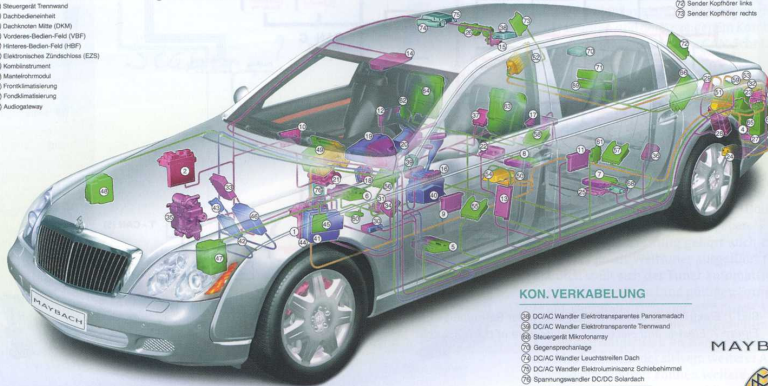
MOST-BUS

- 48) Audiogateway
- 49) HeadUnit
- 50) Steuergerät Sprachbedienung
- 51) TV-Tuner MOST
- 52) Soundverstärker
- 53) Navigationsrechner
- 54) Kommunikationsplattform (CPI)

PRIVATE-BUS

- 55) Sitzsteuergerät Fahrer
- 56) Sitzsteuergerät Beifahrer
- 57) Sitzsteuergerät hinten links
- 58) Sitzsteuergerät hinten rechts
- 59) TV-Tuner CAN
- 60) Dachstrahler
- 61) Sensotronic Brake System (FSG)
- 62) Sensotronic Brake System (ASG 1)
- 63) Sensotronic Brake System (ASG 2)
- 64) Multifunktionssteuergerät
- 65) DVD Spieler
- 66) Multifunktionssteuergerät

- 67) Multifunktionssteuergerät
- 68) Keyless Go Heckmodul
- 69) Keyless Go Innenraummodul
- 70) Keyless Go Tür hinten links
- 71) Keyless Go Tür hinten rechts
- 72) Fondblitzschirm links
- 73) Fondblitzschirm rechts
- 74) Kommunikationsplattform Ford (CP2)
- 75) Surround Amplifier
- 76) Audio Video Controller
- 77) CD-Wechsler
- 78) DVD Spieler
- 79) Sender Kopfhörer links
- 80) Sender Kopfhörer rechts



KON. VERKABELUNG

- 81) DC/AC Wandler Elektrotransparentes Panoramadach
- 82) DC/AC Wandler Elektrotransparente Trennwand
- 83) Steuergerät Mikroklimatry
- 84) Gegensprechanlage
- 85) DC/AC Wandler Leuchtschirmen Dach
- 86) DC/AC Wandler Elektroklimatisierung Schiebehimmel
- 87) Spannungswandler DC/DC Solardach

Σ aller Steuergeräte: 76

MAYBACH



- 1 Historischer Bezug
- 2 Echtzeitbetrieb**
 - Definition
 - Realzeitbetrieb
 - Termine
 - Deterministische Ausführung
- 3 Aufbau und Abgrenzung
- 4 Zusammenfassung

*Echtzeitbetrieb 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.*

*Echtzeitbetrieb 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.*



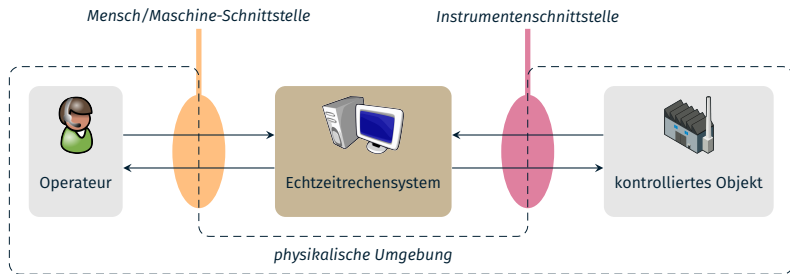
Operateur



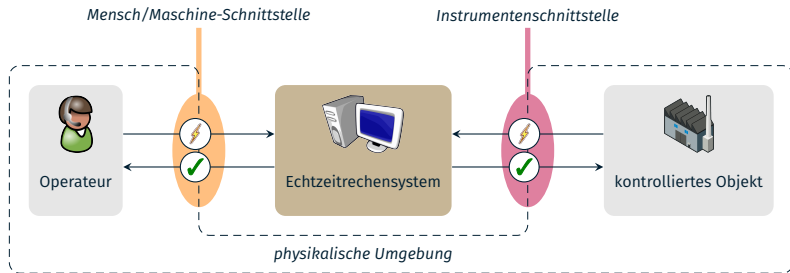
Echtzeitrechnungssystem



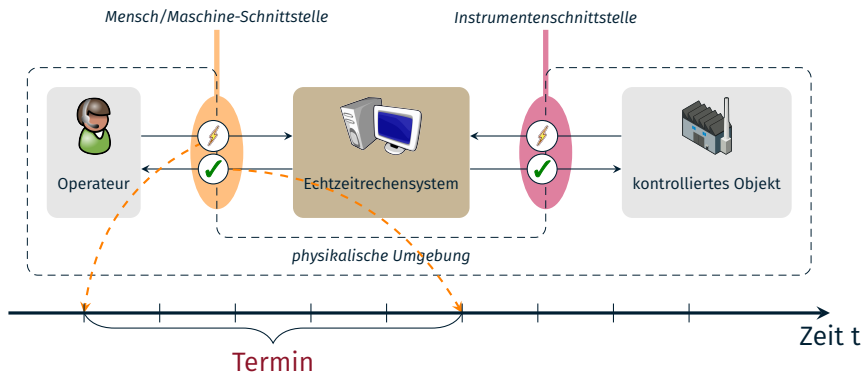
kontrolliertes Objekt



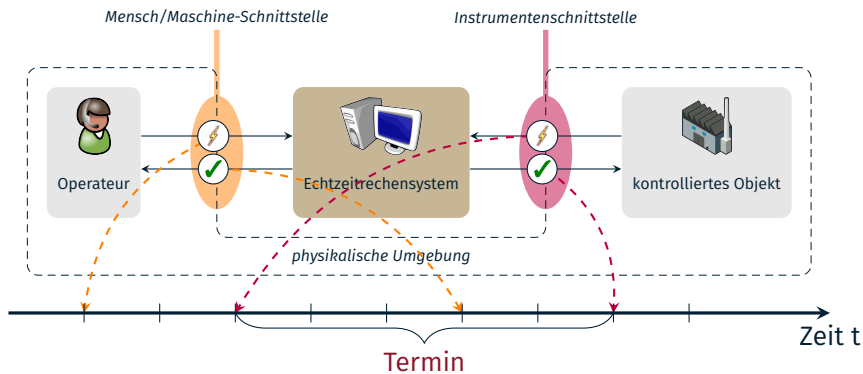
- Echtzeitrechner interagiert mit der **physikalischen Umwelt**



- Echtzeitrechnungssystem interagiert mit der **physikalischen Umwelt**
- Berechnet als Reaktion auf **Ereignisse** ⚡ (engl. *event*, Stimuli) der Umgebung **Ergebnisse** ✓ (engl. *result*)



- Echtzeitrechnungssystem interagiert mit der **physikalischen Umwelt**
- Berechnet als Reaktion auf **Ereignisse** ⚡ (engl. *event*, Stimuli) der Umgebung **Ergebnisse** ✓ (engl. *result*)
- Zeitpunkt, zu dem ein Ergebnis vorliegen muss, wird als **Termin** oder **Frist** (engl. *deadline*) bezeichnet



- Echtzeitrechnungssystem interagiert mit der **physikalischen Umwelt**
- Berechnet als Reaktion auf **Ereignisse** ⚡ (engl. *event*, Stimuli) der Umgebung **Ergebnisse** ✓ (engl. *result*)
- Zeitpunkt, zu dem ein Ergebnis vorliegen muss, wird als **Termin** oder **Frist** (engl. *deadline*) bezeichnet

Verarbeitung von Programmen in Echtzeit

Realzeitverarbeitung (engl. *real-time processing*)

Echtzeitbetrieb bedeutet **Rechtzeitigkeit**

- Funktionale Korrektheit reicht für korrektes Systemverhalten nicht aus
- **Rechtzeitige** Bereitstellung der Ergebnisse ist **entscheidend**

Verarbeitung von Programmen in Echtzeit

Realzeitverarbeitung (engl. *real-time processing*)

Echtzeitbetrieb bedeutet **Rechtzeitigkeit**

- Funktionale Korrektheit reicht für korrektes Systemverhalten nicht aus
 - **Rechtzeitige** Bereitstellung der Ergebnisse ist **entscheidend**
-
- Den Rahmen stecken der Eintrittspunkt des Ereignisses und der entsprechende Termin ab

Verarbeitung von Programmen in Echtzeit

Realzeitverarbeitung (engl. *real-time processing*)

Echtzeitbetrieb bedeutet **Rechtzeitigkeit**

- Funktionale Korrektheit reicht für korrektes Systemverhalten nicht aus
 - **Rechtzeitige** Bereitstellung der Ergebnisse ist **entscheidend**
-
- Den Rahmen stecken der Eintrittspunkt des Ereignisses und der entsprechende Termin ab
-
- Termine hängen dabei von der Anwendung ab
 - wenige Mikrosekunden** z.B. Drehzahl- und Stromregelung bei der Ansteuerung von Elektromotoren
 - einige Millisekunden** z.B. Multimedia-Anwendungen (Übertragung von Ton- und Video)
 - Sekunden, Minuten, Stunden** z.B. Prozessanlagen (Erhitzen von Wasser)

Geschwindigkeit impliziert nicht unbedingt Rechtzeitigkeit

Zuverlässige Reaktion des Rechensystems auf Umgebungsereignisse:

Geschwindigkeit ist keine Garantie für die rechtzeitige Bereitstellung von Ergebnissen

- Asynchrone Programmunterbrechungen (engl. *interrupts*) können **unvorhersagbare Laufzeitvarianzen** verursachen
- Schnelle Programmausführung ist bestenfalls hinreichend für die rechtzeitige Bearbeitung einer Aufgabe

Geschwindigkeit impliziert nicht unbedingt Rechtzeitigkeit

Zuverlässige Reaktion des Rechensystems auf Umgebungsereignisse:

Geschwindigkeit ist keine Garantie für die rechtzeitige Bereitstellung von Ergebnissen

- Asynchrone Programmunterbrechungen (engl. *interrupts*) können **unvorhersagbare Laufzeitvarianzen** verursachen
 - Schnelle Programmausführung ist bestenfalls hinreichend für die rechtzeitige Bearbeitung einer Aufgabe
 - **Zeit ist keine intrinsische Eigenschaft des Rechensystems**
 - Die Zeitskala des Rechensystems muss nicht mit der durch die Umgebung vorgegebenen (Realzeit) übereinstimmen \leadsto Zeitgeber?
- Temporale Eigenschaften des kontrollierten (physikalischen) Objekts müssen im Rechenystem geeignet abgebildet werden

- **Weich** (engl. *soft*) auch „schwach“
 - Ergebnis verliert mit zunehmender Terminüberschreitung an Wert (z.B. Bildrate bei Multimediasystemen)
- Terminverletzung ist tolerierbar

- **Weich** (engl. *soft*) auch „schwach“
 - Ergebnis verliert mit zunehmender Terminüberschreitung an Wert (z.B. Bildrate bei Multimediasystemen)
 - Terminverletzung ist tolerierbar
- **Fest** (engl. *firm*) auch „stark“
 - Ergebnis wird durch eine Terminüberschreitung wertlos und wird verworfen (z.B. Abgabetermin einer Übungsaufgabe)
 - Terminverletzung ist tolerierbar, führt zum Arbeitsabbruch

- **Weich** (engl. *soft*) auch „schwach“
 - Ergebnis verliert mit zunehmender Terminüberschreitung an Wert (z.B. Bildrate bei Multimediasystemen)
 - Terminverletzung ist tolerierbar
- **Fest** (engl. *firm*) auch „stark“
 - Ergebnis wird durch eine Terminüberschreitung wertlos und wird verworfen (z.B. Abgabetermin einer Übungsaufgabe)
 - Terminverletzung ist tolerierbar, führt zum Arbeitsabbruch
- **Hart** (engl. *hard*) auch „strikt“
 - Terminüberschreitung kann zum Systemversagen führen und eine „Katastrophe“ hervorrufen (z.B. Airbag)
 - Terminverletzung ist keinesfalls tolerierbar



- **Fest/Hart** \mapsto Terminverletzung ist nicht ausgeschlossen¹
 - Terminverletzung wird vom Betriebssystem erkannt
 - \rightarrow Weiteres Vorgehen hängt von der Art des Termins ab

¹ Auch wenn Ablaufplan und Betriebssystem auf dem Papier Determinismus zeigen, kann das im Feld eingesetzte technische System von unbekanntem/unvermeidbarem Störeinflüssen betroffen sein!



- **Fest/Hart** \mapsto Terminverletzung ist nicht ausgeschlossen¹
 - Terminverletzung wird vom Betriebssystem erkannt
 - \rightarrow Weiteres Vorgehen hängt von der Art des Termins ab

Fest \rightsquigarrow plangemäß weiterarbeiten

- Betriebssystem bricht den Arbeitsauftrag ab
- Nächster Arbeitsauftrag wird (planmäßig) gestartet
- \rightarrow Transparent für die Anwendung

¹ Auch wenn Ablaufplan und Betriebssystem auf dem Papier Determinismus zeigen, kann das im Feld eingesetzte technische System von unbekanntem/unvermeidbarem Störeinflüssen betroffen sein!



- **Fest/Hart** \mapsto Terminverletzung ist nicht ausgeschlossen¹
 - Terminverletzung wird vom Betriebssystem erkannt
 - \rightarrow Weiteres Vorgehen hängt von der Art des Termins ab

Fest \rightsquigarrow plangemäß weiterarbeiten

- Betriebssystem bricht den Arbeitsauftrag ab
- Nächster Arbeitsauftrag wird (planmäßig) gestartet
- \rightarrow Transparent für die Anwendung

Hart \rightsquigarrow sicheren Zustand finden

- Betriebssystem löst eine **Ausnahmesituation** aus
- Ausnahme ist **intransparent für die Anwendung**
- \rightarrow **Anwendung** behandelt diese Ausnahme

¹ Auch wenn Ablaufplan und Betriebssystem auf dem Papier Determinismus zeigen, kann das im Feld eingesetzte technische System von unbekanntem/unvermeidbarem Störeinflüssen betroffen sein!

Radikale Unterschiede im Systementwurf zeichnen sich ab:

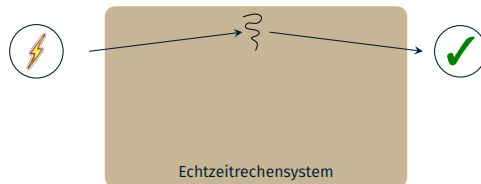
- Hard real-time computer system
(dt. hartes Echtzeitrechensystem)
 - Rechensystem mit mindestens einem hartem Termin
 - Garantiert unter allen (spezifizierten) Last- und Fehlerbedingungen
 - Laufzeitverhalten ist ausnahmslos deterministisch
 - Typisch für sicherheitskritische Echtzeitrechensysteme
 - engl. *safety-critical real-time computer system*
 - Beispiel: Fluglageregelung, Airbag, ...

Radikale Unterschiede im Systementwurf zeichnen sich ab:

- **Hard real-time computer system**
(dt. hartes Echtzeitrechensystem)
 - Rechensystem mit mindestens einem hartem Termin
 - Garantiert unter allen (spezifizierten) Last- und Fehlerbedingungen
 - Laufzeitverhalten ist ausnahmslos deterministisch
 - Typisch für sicherheitskritische Echtzeitrechensysteme
 - engl. *safety-critical real-time computer system*
 - Beispiel: Fluglageregelung, Airbag, ...
- **Soft real-time computer system**
(dt. weiches Echtzeitrechensystem)
 - Rechensystem welches keinen harten Termin erreichen muss
 - Termine können gelegentlich verpasst werden

Herausforderung: Gewährleisten von Rechtzeitigkeit

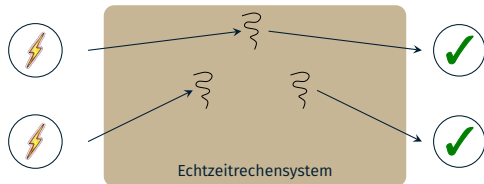
Ereignisbehandlungen müssen termingerecht abgearbeitet werden:



- Ereignisse aktivieren Ereignisbehandlungen
 - Wie viel Zeit benötigt die Ereignisbehandlung **maximal**?
 - Lösung des trivialen Falls ist (scheinbar) einfach, wenn man die **maximale Ausführungszeit** der Ereignisbehandlung kennt

Herausforderung: Gewährleisten von Rechtzeitigkeit

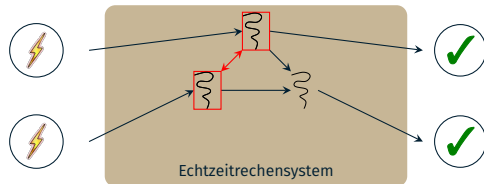
Ereignisbehandlungen müssen termingerecht abgearbeitet werden:



- Ereignisse aktivieren Ereignisbehandlungen
 - Wie viel Zeit benötigt die Ereignisbehandlung **maximal**?
 - Lösung des trivialen Falls ist (scheinbar) einfach, wenn man die **maximale Ausführungszeit** der Ereignisbehandlung kennt
- Reale Echtzeitsysteme sind **komplex**
 - Mehrere Ereignisbehandlungen \leadsto Konkurrenz
 - Verwaltung gemeinsamer Betriebsmittel, allen voran die CPU

Herausforderung: Gewährleisten von Rechtzeitigkeit

Ereignisbehandlungen müssen termingerecht abgearbeitet werden:



- Ereignisse aktivieren Ereignisbehandlungen
 - Wie viel Zeit benötigt die Ereignisbehandlung **maximal**?
 - Lösung des trivialen Falls ist (scheinbar) einfach, wenn man die **maximale Ausführungszeit** der Ereignisbehandlung kennt
- Reale Echtzeitsysteme sind **komplex**
 - Mehrere Ereignisbehandlungen \leadsto Konkurrenz
 - Verwaltung gemeinsamer Betriebsmittel, allen voran die CPU
 - Abhängigkeiten zwischen verschiedenen Ereignisbehandlungen

Vorhersagbarkeit des Laufzeitverhaltens

Echtzeitsysteme sind (schwach, stark oder strikt) deterministisch:

Determiniertheit

*Bei identischen Eingaben sind **verschiedene Abläufe** zulässig, sie liefern jedoch stets das gleiche Resultat.*

Vorhersagbarkeit des Laufzeitverhaltens

Echtzeitsysteme sind (schwach, stark oder strikt) deterministisch:

Determiniertheit

*Bei identischen Eingaben sind **verschiedene Abläufe** zulässig, sie liefern jedoch stets das gleiche Resultat.*

- Im Allgemeinen **unzureichend** für den Entwurf von Echtzeitsystemen
Transparenz von Programmunterbrechungen
 - **Interrupts** verursachen vom normalen Ablauf abweichende **ausnahmebedingte Abläufe**

Vorhersagbarkeit des Laufzeitverhaltens

Echtzeitsysteme sind (schwach, stark oder strikt) deterministisch:

Determiniertheit

*Bei identischen Eingaben sind **verschiedene Abläufe** zulässig, sie liefern jedoch stets das gleiche Resultat.*

- Im Allgemeinen **unzureichend** für den Entwurf von Echtzeitsystemen
Transparenz von Programmunterbrechungen
 - **Interrupts** verursachen vom normalen Ablauf abweichende **ausnahmebedingte Abläufe**

Determinismus

*Identische Eingaben führen zu **identischen Abläufen**. Zu jedem Zeitpunkt ist bestimmt, wie weitergefahren wird.*

Vorhersagbarkeit des Laufzeitverhaltens

Echtzeitsysteme sind (schwach, stark oder strikt) deterministisch:

Determiniertheit

*Bei identischen Eingaben sind **verschiedene Abläufe** zulässig, sie liefern jedoch stets das gleiche Resultat.*

- Im Allgemeinen **unzureichend** für den Entwurf von Echtzeitsystemen
Transparenz von Programmunterbrechungen
 - **Interrupts** verursachen vom normalen Ablauf abweichende **ausnahmebedingte Abläufe**

Determinismus

*Identische Eingaben führen zu **identischen Abläufen**. Zu jedem Zeitpunkt ist bestimmt, wie weitergefahren wird.*

- Notwendig, falls Termine einzuhalten sind
 - Nur so lässt sich das Laufzeitverhalten verlässlich abschätzen

Echtzeitsysteme sind (schwach, stark oder strikt) deterministisch:

Vorhersagbarkeit

Der **Ablauf** lässt sich zu jedem Zeitpunkt **exakt angeben** und hängt nicht von den aktuellen Eingaben oder vom aktuellen Zustand ab.

Echtzeitsysteme sind (schwach, stark oder strikt) deterministisch:

Vorhersagbarkeit

Der **Ablauf** lässt sich zu jedem Zeitpunkt **exakt angeben** und hängt nicht von den aktuellen Eingaben oder vom aktuellen Zustand ab.

- **Vorteilhaft** für zeitkritische Systeme
 - Exakte Angaben zum zeitlichen Ablauf sind bereits à priori möglich
- Von Umgebung und Eingaben entkoppeltes Laufzeitverhalten
 - Aktivitäten folgen einem strikt vorgegebenem Stundenplan

Echtzeitsysteme sind (schwach, stark oder strikt) deterministisch:

Vorhersagbarkeit

Der **Ablauf** lässt sich zu jedem Zeitpunkt **exakt angeben** und hängt nicht von den aktuellen Eingaben oder vom aktuellen Zustand ab.

- **Vorteilhaft** für zeitkritische Systeme
 - Exakte Angaben zum zeitlichen Ablauf sind bereits à priori möglich
 - Von Umgebung und Eingaben entkoppeltes Laufzeitverhalten
 - Aktivitäten folgen einem strikt vorgegebenem Stundenplan

Echtzeitsysteme müssen stets ein deterministisches oder besser vorhersagbares Laufzeitverhalten gewährleisten!

- Insbesondere beim **Zugriff auf gemeinsame Betriebsmittel**
 - CPU** → Umschaltung zwischen verschiedenen Aktivitäten
 - Kommunikationsmedium** → Versand von Nachrichten

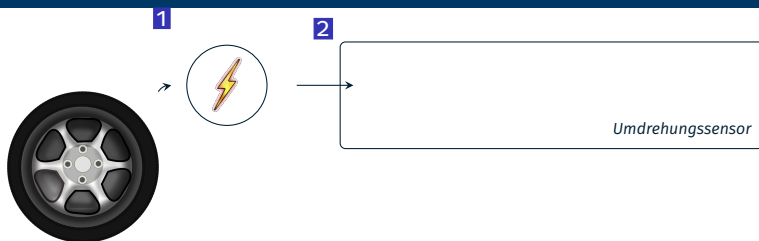
🏠 Beispiel: Ein (fiktives) Anti-Blockier-System

1



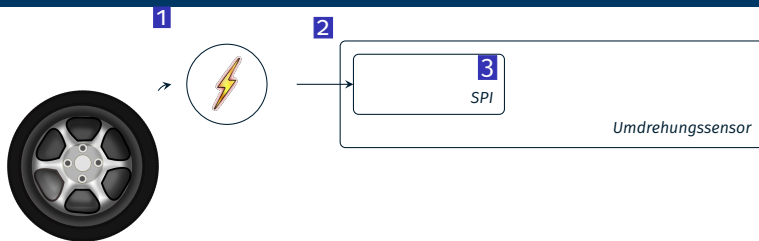
- ABS überwacht kontinuierlich Umdrehungszahl des Rads
→ Messfühler erzeugt Signale (Ereignisse)

🏠 Beispiel: Ein (fiktives) Anti-Blockier-System



- ABS überwacht kontinuierlich Umdrehungszahl des Rads
→ Messfühler erzeugt Signale (Ereignisse)
- Intelligenter Sensor (engl. *smart sensor*) führt Vorverarbeitung der Daten durch (erkennt z.B. Stillstand)

🏠 Beispiel: Ein (fiktives) Anti-Blockier-System

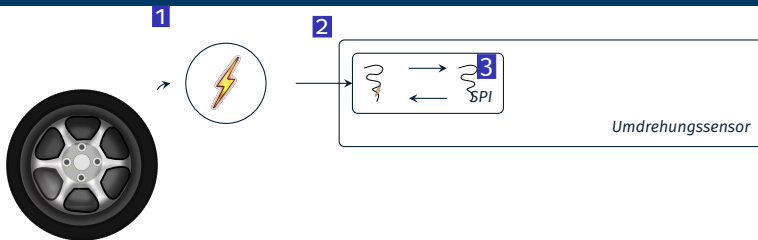


- Meßfühlerdaten werden über den SPI²-Bus entgegengenommen

²serial peripheral interface

³interrupt service routine

🏠 Beispiel: Ein (fiktives) Anti-Blockier-System

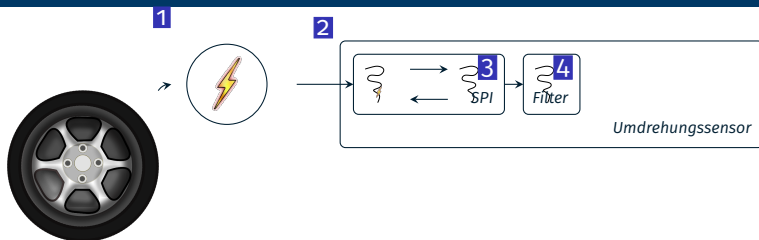


- Meßfühlerdaten werden über den SPI²-Bus entgegengenommen
 - Buskommunikation erfordert eine ISR³ und einen Faden
 - Wann wird die ISR angesprungen? Sind Unterbrechungen gesperrt?
 - Wann wird der Faden eingeplant? Muss er auf Betriebsmittel warten?

²serial peripheral interface

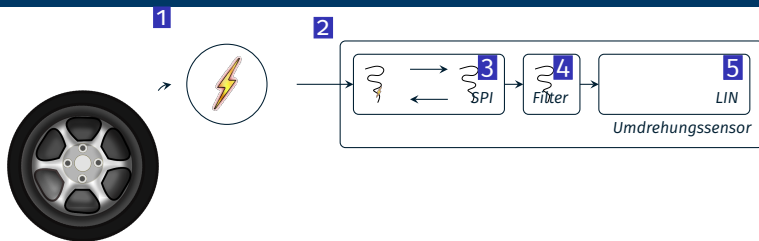
³interrupt service routine

🏠 Beispiel: Ein (fiktives) Anti-Blockier-System



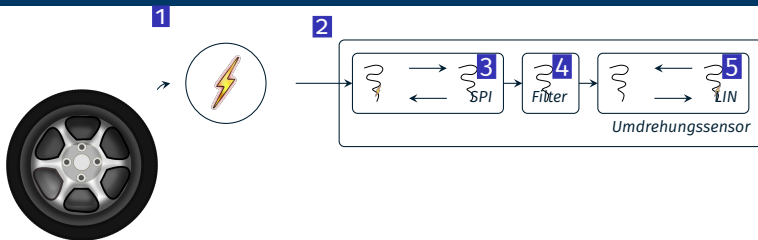
- Filter übernimmt die Signalvorverarbeitung
 - Angleichung diverser Abtastraten durch gesonderten Faden
 - der Filter verarbeitet immer mehrere Messwerte auf einmal
 - Wann wird der Faden eingeplant? Muss er auf Betriebsmittel warten?

🏠 Beispiel: Ein (fiktives) Anti-Blockier-System



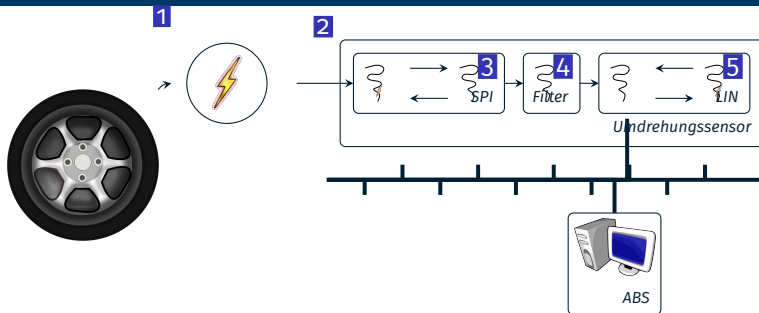
- Konsolidierte Messwerte werden an ABS-Steuergerät gesendet

🏠 Beispiel: Ein (fiktives) Anti-Blockier-System



- Konsolidierte Messwerte werden an ABS-Steuergerät gesendet
 - Komplexer Gerätetreiber notwendig
 - Wann wird die ISR angesprungen? Sind Unterbrechungen gesperrt?
 - Wann wird der Faden eingeplant? Muss er auf Betriebsmittel warten?
 - Können alle Daten „auf einmal“ übertragen werden?

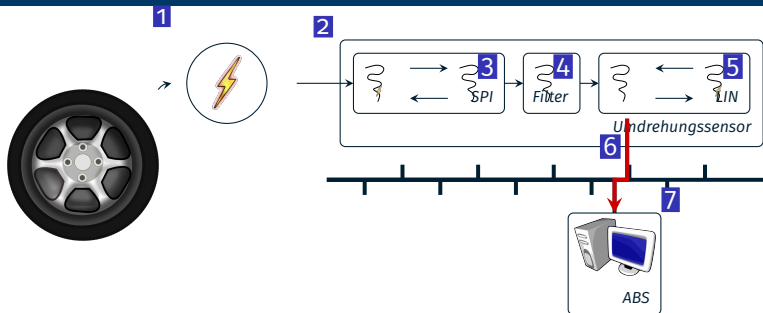
🏠 Beispiel: Ein (fiktives) Anti-Blockier-System



- Sensor und ABS-Steuergerät sind per LIN-Bus² verbunden

²local interconnect network

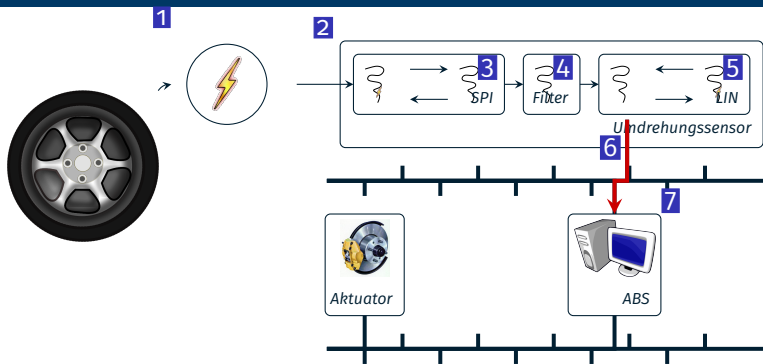
🏠 Beispiel: Ein (fiktives) Anti-Blockier-System



- Sensor und ABS-Steuergerät sind per LIN-Bus² verbunden
 - Datenübertragung benötigt Zeit ...
 - Wie lange muss ich warten, bis ich auf das Medium zugreifen kann?
- Vorgänge im ABS-Steuergerät sind noch deutlich komplexer

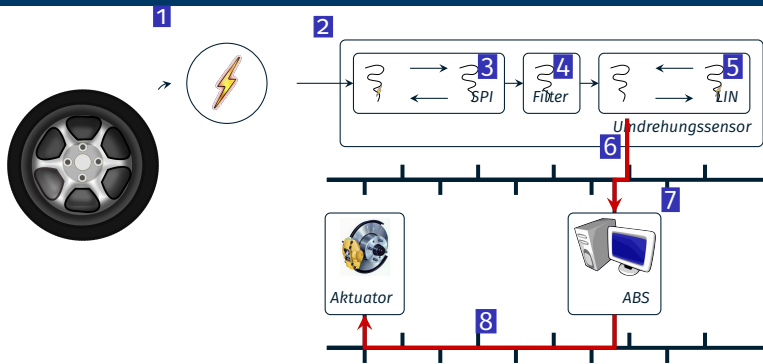
²local interconnect network

🏠 Beispiel: Ein (fiktives) Anti-Blockier-System



- Stellwert wird dem Aktor zugestellt
 - CAN-Bus verbindet ABS-Steuergerät und Aktor

🏠 Beispiel: Ein (fiktives) Anti-Blockier-System

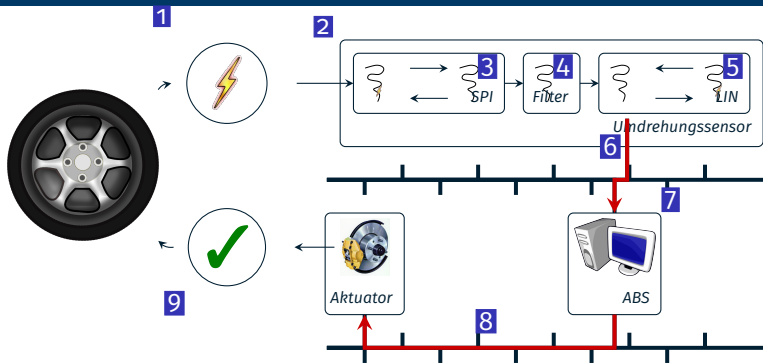


■ Stellwert wird dem Aktor zugestellt

■ CAN-Bus verbindet ABS-Steuergerät und Aktor

- Wieviele Bytes schafft der Bus in einer bestimmten Zeit?
- Wie lange muss ich warten, bis ich auf das Medium zugreifen kann?

🏠 Beispiel: Ein (fiktives) Anti-Blockier-System



- Stellwert wird dem Aktor zugestellt
 - CAN-Bus verbindet ABS-Steuergerät und Aktor
 - Wieviele Bytes schafft der Bus in einer bestimmten Zeit?
 - Wie lange muss ich warten, bis ich auf das Medium zugreifen kann?
- schließlich wird die Bremskraft geeignet beeinflusst

Wie lange dauert das Ganze nun?

Die korrekte Funktion des ABS erfordert eine Reaktion auf eine Blockierung des Rades innerhalb einer bestimmten Zeitspanne

Wie lange dauert das Ganze nun?

Die korrekte Funktion des ABS erfordert eine Reaktion auf eine Blockierung des Rades innerhalb einer bestimmten Zeitspanne

- Zu dieser Zeitspanne tragen zwei Komponenten bei:

Aktive Zeitintervalle \rightsquigarrow „Fortschritt“ im ABS

- Berechnungen benötigen Zeit \rightsquigarrow maximale Ausführungszeit
- Geschwindigkeit der Datenübertragung ist beschränkt

Wie lange dauert das Ganze nun?

Die korrekte Funktion des ABS erfordert eine Reaktion auf eine Blockierung des Rades innerhalb einer bestimmten Zeitspanne

- Zu dieser Zeitspanne tragen zwei Komponenten bei:

Aktive Zeitintervalle \rightsquigarrow „Fortschritt“ im ABS

- Berechnungen benötigen Zeit \rightsquigarrow maximale Ausführungszeit
- Geschwindigkeit der Datenübertragung ist beschränkt

Inaktive Zeitintervalle \rightsquigarrow „Wartezeit“ für das ABS

- Fortschritt erfordert die Zuteilung von Betriebsmitteln
- z. B. CPU oder Kommunikationsmedium

Wie lange dauert das Ganze nun?

Die korrekte Funktion des ABS erfordert eine Reaktion auf eine Blockierung des Rades innerhalb einer bestimmten Zeitspanne

- Zu dieser Zeitspanne tragen zwei Komponenten bei:

Aktive Zeitintervalle \rightsquigarrow „Fortschritt“ im ABS

- Berechnungen benötigen Zeit \rightsquigarrow maximale Ausführungszeit
- Geschwindigkeit der Datenübertragung ist beschränkt

Inaktive Zeitintervalle \rightsquigarrow „Wartezeit“ für das ABS

- Fortschritt erfordert die Zuteilung von Betriebsmitteln
- z. B. CPU oder Kommunikationsmedium
- Die Frage ist, wie lange man auf die Zuteilung warten muss!
 - **Determiniertheit** alleine reicht für die Beantwortung nicht aus!
 - Determinismus erfordert die vollständige Kenntnis der Umgebung!
 - **Vorhersagbarkeit** liefert die gewünschte Aussage zu dieser Frage!

- Deterministische Abarbeitung von Ereignisbehandlungen?

Charakterisierung von Echtzeitanwendungen [4, S. 25]

- Deterministische Abarbeitung von Ereignisbehandlungen?
- **Rein zyklisch** \leadsto periodische Ereignisbehandlungen, Abfrage-Betrieb
 - (Nahezu) konstanter Betriebsmittelbedarf von Periode zu Periode

Charakterisierung von Echtzeitanwendungen [4, S. 25]

- Deterministische Abarbeitung von Ereignisbehandlungen?
- **Rein zyklisch** \rightsquigarrow periodische Ereignisbehandlungen, Abfrage-Betrieb
 - (Nahezu) konstanter Betriebsmittelbedarf von Periode zu Periode
- **Meist zyklisch** \rightsquigarrow überwiegend periodische Ereignisbehandlungen
 - System muss auf externe Ereignisse reagieren können
 - Betriebsmittelbedarf schwankt bedingt von Periode zu Periode

Charakterisierung von Echtzeitanwendungen [4, S. 25]

- Deterministische Abarbeitung von Ereignisbehandlungen?
- **Rein zyklisch** \leadsto periodische Ereignisbehandlungen, Abfrage-Betrieb
 - (Nahezu) konstanter Betriebsmittelbedarf von Periode zu Periode
- **Meist zyklisch** \leadsto überwiegend periodische Ereignisbehandlungen
 - System muss auf externe Ereignisse reagieren können
 - Betriebsmittelbedarf schwankt bedingt von Periode zu Periode
- **Asynchron/vorhersagbar** \leadsto kaum periodische Ereignisbehandlungen
 - Aufeinanderfolgende Aktivierungen können zeitlich stark variieren
 - Zeitdifferenzen haben eine obere Grenze oder bekannte Statistik
 - Stark schwankender Betriebsmittelbedarf

Charakterisierung von Echtzeitanwendungen [4, S. 25]

- Deterministische Abarbeitung von Ereignisbehandlungen?
- **Rein zyklisch** \rightsquigarrow periodische Ereignisbehandlungen, Abfrage-Betrieb
 - (Nahezu) konstanter Betriebsmittelbedarf von Periode zu Periode
- **Meist zyklisch** \rightsquigarrow überwiegend periodische Ereignisbehandlungen
 - System muss auf externe Ereignisse reagieren können
 - Betriebsmittelbedarf schwankt bedingt von Periode zu Periode
- **Asynchron/vorhersagbar** \rightsquigarrow kaum periodische Ereignisbehandlungen
 - Aufeinanderfolgende Aktivierungen können zeitlich stark variieren
 - Zeitdifferenzen haben eine obere Grenze oder bekannte Statistik
 - Stark schwankender Betriebsmittelbedarf
- **Asynchron/nicht vorhersagbar** \rightsquigarrow aperiodische Ereignisbehandlungen
 - Ausschließlich externe Ereignisse
 - Hohe, nicht deterministische Laufzeitkomplexität einzelner Ereignisbehandlungen

- 1 Historischer Bezug
- 2 Echtzeitbetrieb
 - Definition
 - Realzeitbetrieb
 - Termine
 - Deterministische Ausführung
- 3 Aufbau und Abgrenzung**
- 4 Zusammenfassung

Aufbau der Vorlesung

- Die Vorlesung orientiert sich vor allem ...
 - an der Ausprägung des Spezialzweckbetriebs

Einleitung

Grundlagen

Aufbau der Vorlesung

- Die Vorlesung orientiert sich vor allem ...
 - an der Ausprägung des Spezialzweckbetriebs

Einleitung

Grundlagen

vorranggesteuerte
Systeme

taktgesteuerte
Systeme

Analyse

Aufbau der Vorlesung

- Die Vorlesung orientiert sich vor allem ...
 - an der Ausprägung des Spezialzweckbetriebs
 - und den Eigenschaften der Ereignisse und ihrer Behandlungen,

Einleitung			
Grundlagen			
	vorranggesteuerte Systeme	taktgesteuerte Systeme	Analyse
periodische Echtzeitsysteme			
nicht-periodische Echtzeitsysteme			
Rangfolge			
Zugriffskontrolle			

Aufbau der Vorlesung

- Die Vorlesung orientiert sich vor allem ...
 - an der Ausprägung des Spezialzweckbetriebs
 - und den Eigenschaften der Ereignisse und ihrer Behandlungen,
 - blickt aber auch über den Tellerrand (z.B. Regelungstechnik).

Einleitung

Grundlagen

vorranggesteuerte
Systeme

taktgesteuerte
Systeme

Analyse

periodische Echtzeitsysteme

nicht-periodische Echtzeitsysteme

Rangfolge

Zugriffskontrolle

Aktuelle Forschungsthemen (Mehrkernrechnensysteme)

Aktuelle Forschungsthemen II

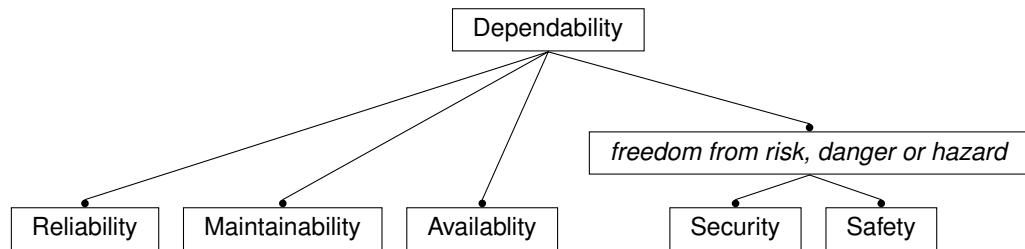
Aufbau der Vorlesung

- Die Vorlesung orientiert sich vor allem ...
 - an der Ausprägung des Spezialzweckbetriebs
 - und den Eigenschaften der Ereignisse und ihrer Behandlungen,
 - blickt aber auch über den Tellerrand (z.B. Regelungstechnik).

Einleitung			
Grundlagen			
	vorranggesteuerte Systeme	taktgesteuerte Systeme	Analyse
periodische Echtzeitsysteme			
nicht-periodische Echtzeitsysteme			
Rangfolge			
Zugriffskontrolle			
Aktuelle Forschungsthemen (Mehrkernrechnensysteme)			
Aktuelle Forschungsthemen II			
Zusammenfassung und Ausblick			

Echtzeitsysteme sind oft **sicherheitskritische Systeme** und erfordern ein hohes Maß an **Verlässlichkeit**.

Echtzeitsysteme sind oft **sicherheitskritische Systeme** und erfordern ein hohes Maß an **Verlässlichkeit**. Verlässlichkeit selbst hat viele Gesichter.



The trustworthiness of a computing system which allows reliance to be justifiably placed on the service it delivers. [3]

Verlässlichkeit erfordert Rechtzeitigkeit!

- Verpasste Termine stellen Fehler dar
- Diese Fehler müssen ggf. erkannt oder maskiert werden

Verlässlichkeit erfordert Rechtzeitigkeit!

- Verpasste Termine stellen Fehler dar
- Diese Fehler müssen ggf. erkannt oder maskiert werden

■ **Andererseits:** Rechtzeitigkeit erfordert Verlässlichkeit!

- Fehler können zum Verpassen eines Termins führen
- Maskieren solcher Fehler hilft, die Rechtzeitigkeit zu gewährleisten

Verlässlichkeit erfordert Rechtzeitigkeit!

- Verpasste Termine stellen Fehler dar
- Diese Fehler müssen ggf. erkannt oder maskiert werden

■ **Andererseits:** Rechtzeitigkeit erfordert Verlässlichkeit!

- Fehler können zum Verpassen eines Termins führen
- Maskieren solcher Fehler hilft, die Rechtzeitigkeit zu gewährleisten

■ Betrachtung Rechtzeitigkeit unter **Annahme des fehlerfreien Falls**

- Verletzte Termine werden auf einer höheren Ebene behandelt
- Toleranz gegenüber Fehlern dient der Verlässlichkeit

Verlässlichkeit erfordert Rechtzeitigkeit!

- Verpasste Termine stellen Fehler dar
- Diese Fehler müssen ggf. erkannt oder maskiert werden

■ **Andererseits:** Rechtzeitigkeit erfordert Verlässlichkeit!

- Fehler können zum Verpassen eines Termins führen
- Maskieren solcher Fehler hilft, die Rechtzeitigkeit zu gewährleisten

■ Betrachtung Rechtzeitigkeit unter **Annahme des fehlerfreien Falls**

- Verletzte Termine werden auf einer höheren Ebene behandelt
- Toleranz gegenüber Fehlern dient der Verlässlichkeit

● Thema **Verlässliche Echtzeitsysteme**  kommendes Semester WS '24

- 1 Historischer Bezug
- 2 Echtzeitbetrieb
 - Definition
 - Realzeitbetrieb
 - Termine
 - Deterministische Ausführung
- 3 Aufbau und Abgrenzung
- 4 Zusammenfassung**

- Echtzeitbetrieb eines Rechensystems in seiner Umgebung
 - Ereignis, Ereignisbehandlung, Ergebnis, Termin
- Komponenten eines Echtzeitsystems
 - Operateur, Echtzeitrechensystem, kontrolliertes Objekt
- Weiche, feste und harte Echtzeitbedingungen
- Determiniertheit, Determinismus, Vorhersagbarkeit
- Verhalten von Echtzeitanwendungen
 - Rein/meist zyklisch
 - Asynchron/vorhersagbar & asynchron/nicht vorhersagbar

- Echtzeitbetrieb eines Rechensystems in seiner Umgebung
 - Ereignis, Ereignisbehandlung, Ergebnis, Termin
- Komponenten eines Echtzeitsystems
 - Operateur, Echtzeitrechensystem, kontrolliertes Objekt
- Weiche, feste und harte Echtzeitbedingungen
- Determiniertheit, Determinismus, Vorhersagbarkeit
- Verhalten von Echtzeitanwendungen
 - Rein/meist zyklisch
 - Asynchron/vorhersagbar & asynchron/nicht vorhersagbar
- Abgrenzung: Fokus dieser Vorlesung liegt auf der **Rechtzeitigkeit**

[1] DaimlerChrysler AG.

Der neue Maybach.

ATZ/MTZ Sonderheft, page 125, September 2002.

[2] Deutsches Institut für Normung.

DIN 44300: Informationsverarbeitung – Begriffe.

Beuth-Verlag, Berlin, Köln, 1985.

[3] IFIP.

Working Group 10.4 on Dependable Computing and Fault Tolerance.

<http://www.dependability.org/wg10.4>, 2003.

[4] Jane W. S. Liu.

Real-Time Systems.

Prentice Hall PTR, Englewood Cliffs, NJ, USA, 2000.

[5] Médecins Sans Frontières.

Innovating to reach remote tb patients and improve access to treatment.

2014.

[6] D. Soesilo, P. Meier, A. Lessard-Fontaine, J. Du Plessis, and C. Stuhlberger.

Drones in humanitarian action: A guide to the use of airborne systems in humanitarian crises, 2016.