Echtzeitsysteme

Einleitung

Sommersemester 2024

Peter Wägemann

Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) Lehrstuhl Informatik 4 (Systemsoftware) https://sys.cs.fau.de





Friedrich-Alexander-Universität Technische Fakultät

Das erste Echtzeitrechensystem

Whirlwind I

 Zweck: Flugsimulator (Ausbildung von Bomberbesatzungen)

Auftraggeber: U.S. Navy

Auftragnehmer: MIT

Laufzeit: 1945 - 1952



(Quelle: Alex Handy from Oakland, Nmibia)

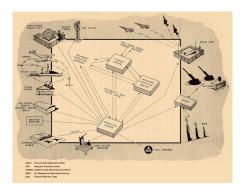
■ Technische Daten

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

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

SAGE – <u>Semi-Automatic Ground Environment</u>

- 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

AN/FSQ-7 Echtzeitrechensystem

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 m²,
 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) ~ 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:







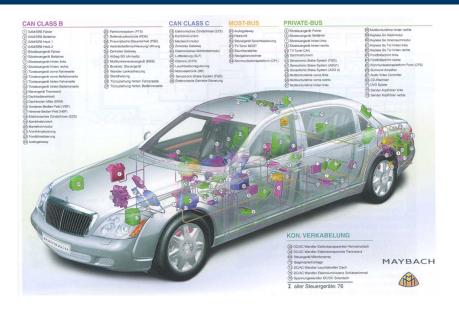












Übersicht

- 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.

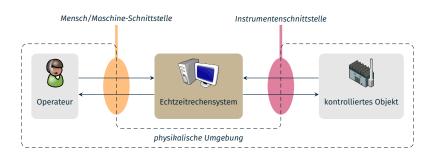
Echtzeitbetrieb ist ein Betrieb eines Rechensystems, bei dem Programme zur Verarbeitung anfallender Daten ständig betriebsbereit sind derart, dass die **Verarbeitungsergebnisse innerhalb einer vorgegebenen Zeitspanne verfügbar** sind.

Die Daten können je nach Anwendungsfall nach einer zeitlich **zu- fälligen Verteilung** oder zu **vorbestimmten Zeitpunkten** anfallen.

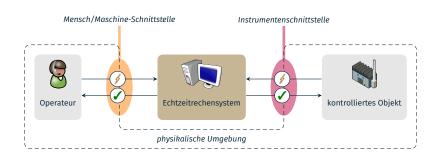




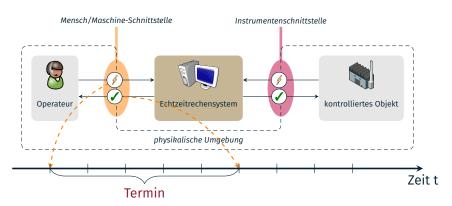




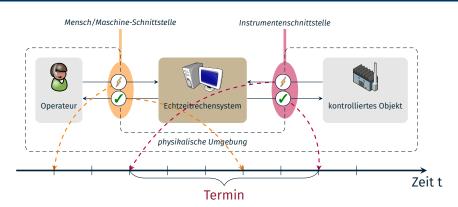
■ Echtzeitrechensystem interagiert mit der **physikalischen Umwelt**



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



- Echtzeitrechensystem 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



- Echtzeitrechensystem 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 <u>nicht</u> 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 <u>nicht</u> 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 <u>nicht</u> 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 ~ Zeitgeber?
 - → Temporale Eigenschaften des kontrollierten (physikalischen) Objekts müssen im Rechensystem geeignet abgebildet werden



⚠ Konsequenzen überschrittener Termine



⚠ Konsequenzen überschrittener Termine

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



∧ Konsequenzen überschrittener Termine

- 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

Arten von Echtzeitsystemen

- Fest/Hart → Terminverletzung ist nicht ausgeschlossen¹
 - Terminverletzung wird vom Betriebssystem erkannt
 - → 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 unbekannten/unvermeidbaren Störeinflüssen betroffen sein!

Arten von Echtzeitsystemen

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

Fest → plangemäß weiterarbeiten

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

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

Arten von Echtzeitsystemen

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

Fest → plangemäß weiterarbeiten

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

Hart → sicheren Zustand finden

- Betriebssystem löst eine Ausnahmesituation aus
- Ausnahme ist intransparent für die Anwendung
- → Anwendung behandelt diese Ausnahme

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

Arten von Echtzeitsystemen (Forts.)

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, ...

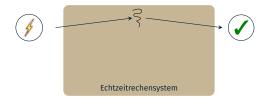
Arten von Echtzeitsystemen (Forts.)

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

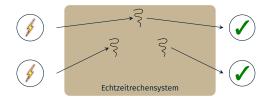
Ereignisbehandlungen müssen termingerecht abgearbeitet werden:



- 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

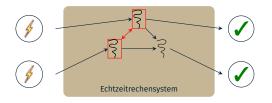
Ereignisbehandlungen müssen termingerecht abgearbeitet werden:



- 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 → 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 → Konkurrenz
 - Verwaltung gemeinsamer Betriebsmittel, allen voran die CPU
 - Abhängigkeiten zwischen verschiedenen Ereignisbehandlungen

Echtzeitsysteme sind (schwach, stark oder strikt) deterministisch:

Determiniertheit

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

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

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.

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

Vorhersagbarkeit des Laufzeitverhaltens (Forts.)

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.

Vorhersagbarkeit des Laufzeitverhaltens (Forts.)

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
 - Unabhängigkeit von Eingabedaten
 - → Aktivitäten folgen einem strikt vorgegebenem Stundenplan

Vorhersagbarkeit des Laufzeitverhaltens (Forts.)

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
 - Unabhängigkeit von Eingabedaten
 - → Aktivitäten folgen einem strikt vorgegebenem Stundenplan

Echtzeitsysteme müssen stets ein **deterministisches** oder besser **vorhersagbares** Laufzeitverhalten gewährleisten!

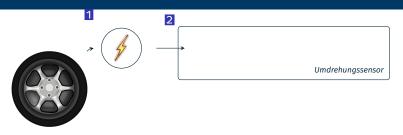
■ Insbesondere beim Zugriff auf gemeinsame Betriebsmittel

CPU → Umschaltung zwischen verschiedenen Aktivitäten

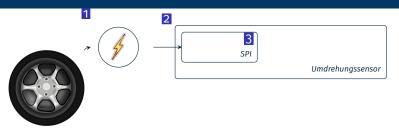
Kommunikationsmedium → Versand von Nachrichten



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



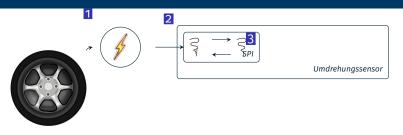
- ABS überwacht kontinuierlich Umdrehungszahl des Rads
 - → Messfühler erzeugt Signale (Ereignisse)
- Intelligenter Sensor (engl. smart sensor) führt Vorverarbeitung der Daten durch (erkennt z.B. Stillstand)



■ Meßfühlerdaten werden über den SPI²-Bus entgegengenommen

²serial peripheral interface

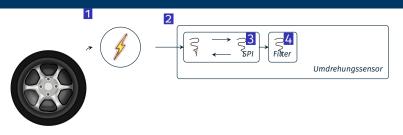
³interrupt service routine



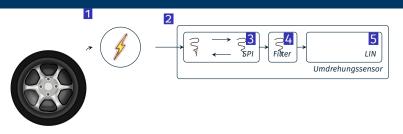
- Meßfühlerdaten werden über den SPI²-Bus entgegengenommen
 - Buskommunikation erfordert eine ISR³ und einen Faden
 - ightarrow Wann wird die ISR angesprungen? Sind Unterbrechungen gesperrt?
 - → Wann wird der Faden eingeplant? Muss er auf Betriebsmittel warten?

²serial peripheral interface

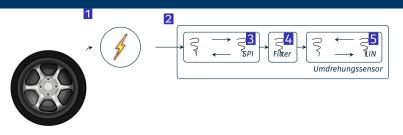
³interrupt service routine



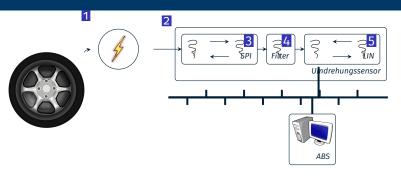
- Filter übernimmt die Signalvorverarbeitung
 - Angleichung diverser Abtastraten durch gesonderten Faden
 - der Filter verarbeitet immer mehrere Messwerte auf einmal
 - → Wann wird der Faden eingeplant? Muss er auf Betriebsmittel warten?



■ Konsolidierte Messwerte werden an ABS-Steuergerät gesendet

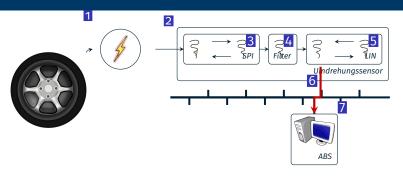


- Konsolidierte Messwerte werden an ABS-Steuergerät gesendet
 - Komplexer Gerätetreiber notwendig
 - → Wann wird die ISR angesprungen? Sind Unterbrechungen gesperrt?
 - → Wann wird der Faden eingeplant? Muss er auf Betriebsmittel warten?
 - → Können alle Daten "auf einmal" übertragen werden?



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

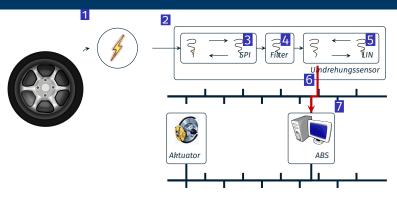
²local interconnect network



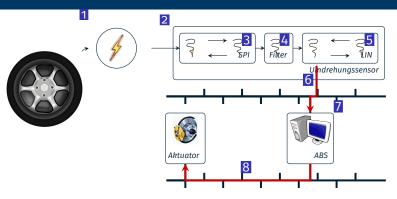
- Sensor und ABS-Steuergerät sind per LIN-Bus² verbunden
 - Datenübertragung benötigt Zeit ...
 - → Wie lange muss ich warten, bis ich auf das Medium zugreifen kann?

⚠ Vorgänge im ABS-Steuergerät sind noch deutlich komplexer

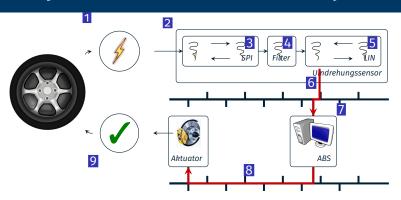
²local interconnect network



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



- Stellwert wird dem Aktor zugestellt
 - CAN-Bus verbindet ABS-Steuergerät und Aktor
 - → Wieviele Bytes schafft der Bus in einer bestimmten Zeit?
 - $\ensuremath{\rightarrow}$ Wie lange muss ich warten, bis ich auf das Medium zugreifen kann?



- Stellwert wird dem Aktor zugestellt
 - CAN-Bus verbindet ABS-Steuergerät und Aktor
 - → Wieviele Bytes schafft der Bus in einer bestimmten Zeit?
 - $\boldsymbol{\rightarrow}$ 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** ~> "Fortschritt" im ABS
 - Berechnungen benötigen Zeit ~ 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** ~> "Fortschritt" im ABS
 - Berechnungen benötigen Zeit ~> maximale Ausführungszeit
 - Geschwindigkeit der Datenübertragung ist beschränkt

Inaktive Zeitintervalle → "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** ~> "Fortschritt" im ABS
 - Berechnungen benötigen Zeit ~> maximale Ausführungszeit
 - Geschwindigkeit der Datenübertragung ist beschränkt

Inaktive Zeitintervalle → "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?

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

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

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

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

Übersicht

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

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

Einleitung Grundlagen

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

Einleitung Grundlagen			
	vorranggesteuerte Systeme	taktgesteuerte Systeme	Analyse

- 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 Echtzeit	systeme			
nicht-periodische Ec <mark>htzeitsysteme</mark>				
Rangfolge				
Zugriffskontrolle				

- 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).

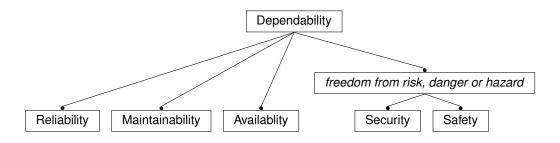
Einleitung				
Grundlagen				
	vorranggesteuerte Systeme	taktgesteuerte Systeme	Analyse	
periodische Echtzeits <mark>ysteme</mark>				
nicht-periodische Ec	htzeitsysteme			
Rangfolge				
Zugriffskontrolle				
Aktuelle Forschungsthemen I (Mehrkernrechensysteme)				
Aktuelle Forschungsthemen II				

- 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 Echtzeits <mark>ysteme</mark>				
nicht-periodische Ec	htzeitsysteme			
Rangfolge				
Zugriffskontrolle				
Aktuelle Forschungsthemen I (Mehrkernrechensysteme)				
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

- \triangle
- 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

- \triangle
- 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

Übersicht

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

Resümee

- 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

Resümee

- 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

Literaturverzeichnis (1)

[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.

Literaturverzeichnis (2)

[4] Jane W. S. Liu.

Real-Time Systems.

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

[5] Médecins Sans Frontières.

Innovating to reach remote to 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.