

# Echtzeitsysteme

---

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

*Die Lehrveranstaltung ist grundsätzlich für alle Studiengänge offen. Sie verlangt allerdings gewisse Vorkenntnisse. Diese müssen nicht durch Teilnahme an den Lehrveranstaltungen von I4 erworben worden sein.*

# Echtzeitsysteme – Eine Begriffsdefinition

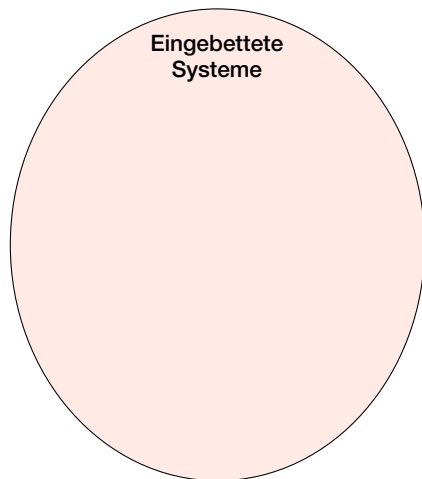
- Echtzeit ist ein strapazierter Begriff

# Echtzeitsysteme – Eine Begriffsdefinition

- Echtzeit ist ein strapazierter Begriff  
Es geht nicht um **Geschwindigkeit**  
sondern um **Rechtzeitigkeit!**

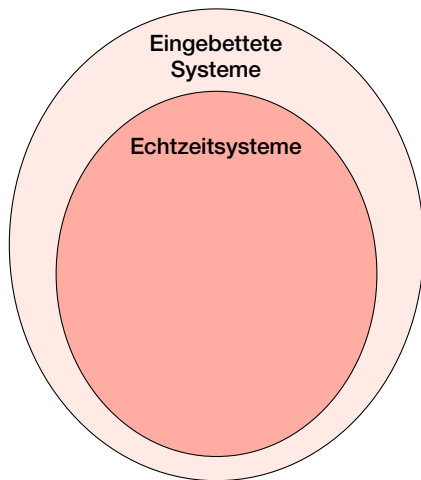
# Echtzeitsysteme – Eine Begriffsdefinition

- Echtzeit ist ein strapazierter Begriff  
Es geht nicht um **Geschwindigkeit**  
sondern um **Rechtzeitigkeit!**
- Echtzeitsysteme, eine (strikte)  
Definition und Einordnung:
  - Eingebettet in die Umwelt und  
abhängig von der Hardware



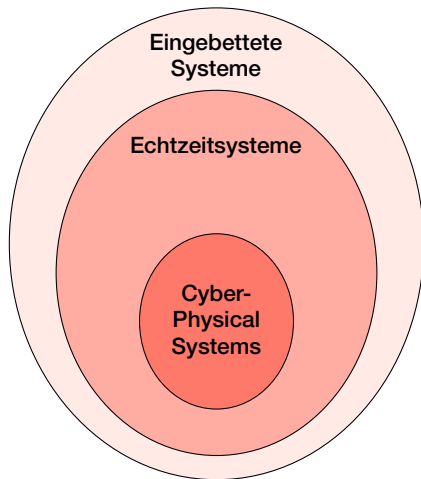
# Echtzeitsysteme – Eine Begriffsdefinition

- Echtzeit ist ein strapazierter Begriff  
Es geht nicht um **Geschwindigkeit**  
sondern um **Rechtzeitigkeit!**
- Echtzeitsysteme, eine (strikte) Definition und Einordnung:
  - Eingebettet in die Umwelt und abhängig von der Hardware
  - An die Realzeit gekoppelt



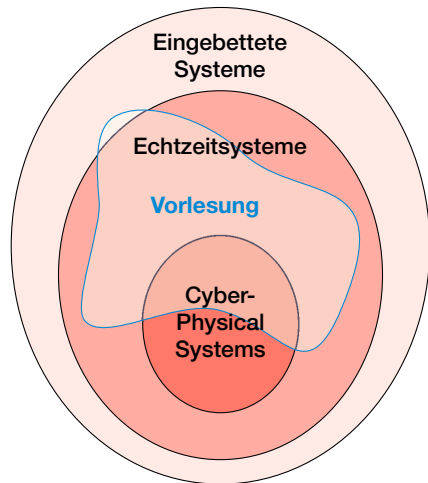
# Echtzeitsysteme – Eine Begriffsdefinition

- Echtzeit ist ein strapazierter Begriff  
Es geht nicht um **Geschwindigkeit**  
sondern um **Rechtzeitigkeit!**
- Echtzeitsysteme, eine (strikte) Definition und Einordnung:
  - Eingebettet in die Umwelt und abhängig von der Hardware
  - An die Realzeit gekoppelt
  - Steuerung und Regelung von physikalischen Prozessen



# Echtzeitsysteme – Eine Begriffsdefinition

- Echtzeit ist ein strapazierter Begriff  
Es geht nicht um **Geschwindigkeit**  
sondern um **Rechtzeitigkeit!**
- Echtzeitsysteme, eine (strikte) Definition und Einordnung:
  - Eingebettet in die Umwelt und abhängig von der Hardware
  - An die Realzeit gekoppelt
  - Steuerung und Regelung von physikalischen Prozessen
- Entwicklung erfolgt typischerweise interdisziplinär!





- 1 Lehrveranstaltungskonzept & Organisation
- 2 Die Veranstaltung
  - Lernziele und Voraussetzungen
  - Einordnung
- 3 Organisatorisches
  - Die Beteiligten
  - Vorlesung und Übung
  - Leistungsnachweise
  - Literaturempfehlungen

# Echtzeitsysteme – Die Veranstaltung

- Querschneidender Einblick in die Welt der Echtzeitsysteme

# Echtzeitsysteme – Die Veranstaltung

- Querschneidender Einblick in die Welt der Echtzeitsysteme
  - Ausgehend von den Eigenschaften der Hardware



Hardware

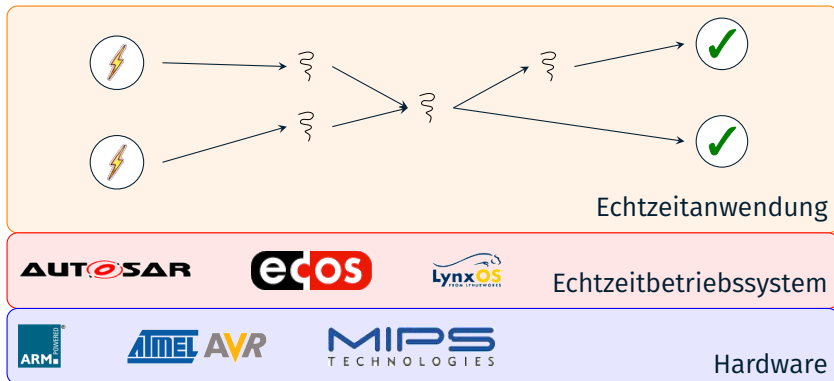
# Echtzeitsysteme – Die Veranstaltung

- Querschneidender Einblick in die Welt der **Echtzeitsysteme**
  - Ausgehend von den Eigenschaften der Hardware
  - Über das Echtzeitbetriebssystem und seine Implementierung



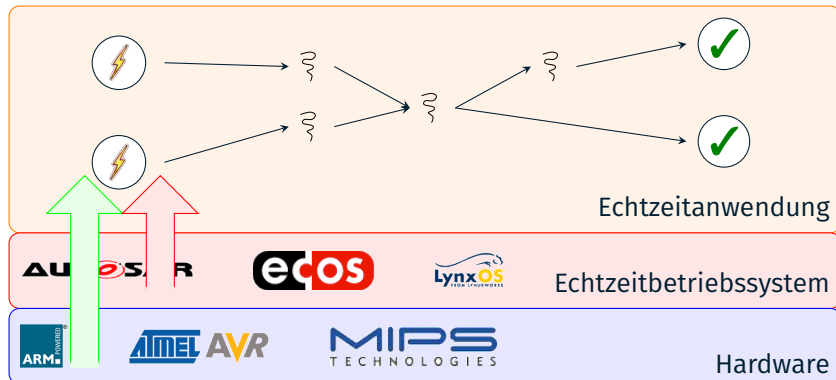
# Echtzeitsysteme – Die Veranstaltung

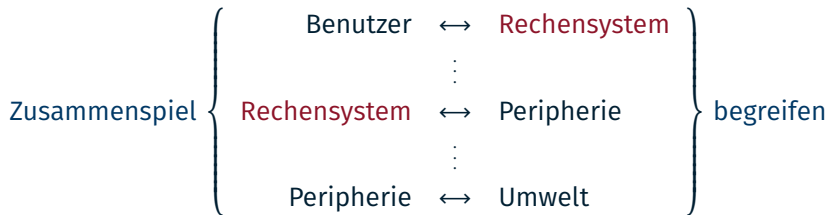
- Querschneidender Einblick in die Welt der **Echtzeitsysteme**
  - Ausgehend von den Eigenschaften der Hardware
  - Über das Echtzeitbetriebssystem und seine Implementierung
  - Bis zum strukturellen Aufbau von Echtzeitanwendungen



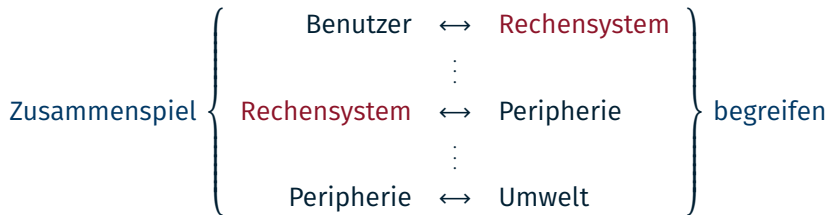
# Echtzeitsysteme – Die Veranstaltung

- Querschneidender Einblick in die Welt der **Echtzeitsysteme**
  - Ausgehend von den Eigenschaften der Hardware
  - Über das Echtzeitbetriebssystem und seine Implementierung
  - Bis zum strukturellen Aufbau von Echtzeitanwendungen



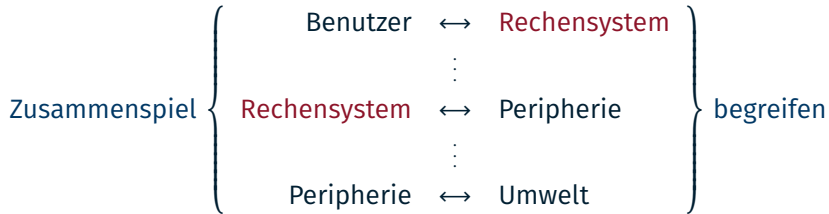


- Echtzeitsysteme als Ganzes **verstehen**:
  - Grad der Echtzeitfähigkeit eines Systems erkennen
  - Einfluss der Hard-/Softwareplattform bewerten
  - Temporale Aspekte physikalischer Prozesse erfassen



- Echtzeitsysteme als Ganzes **verstehen**:
  - Grad der Echtzeitfähigkeit eines Systems erkennen
  - Einfluss der Hard-/Softwareplattform bewerten
  - Temporale Aspekte physikalischer Prozesse erfassen
- Echtzeitsysteme **entwickeln**:
  - Anwendungen analysieren und Werkzeuge einsetzen (Oszilloskop, aiT, ...)
  - Systeme praktisch und (betriebs-)systemnah bauen (eCos, C/C++, ...)





- Echtzeitsysteme als Ganzes **verstehen**:
  - Grad der Echtzeitfähigkeit eines Systems erkennen
  - Einfluss der Hard-/Softwareplattform bewerten
  - Temporale Aspekte physikalischer Prozesse erfassen
- Echtzeitsysteme **entwickeln**:
  - Anwendungen analysieren und Werkzeuge einsetzen (Oszilloskop, aiT, ...)
  - Systeme praktisch und (betriebs-)systemnah bauen (eCos, C/C++, ...)
- **Vertiefen** des Wissens über Echtzeitbetriebssysteme
  - Ablaufplanung und Betriebsmittelverwaltung
  - Mehrkern-Rechensysteme

- Vorlesung: Vorstellung und detaillierte Behandlung des Stoffs
  - Grundlagen von Echtzeitsystemen
    - Zeit- und ereignisgesteuerte Systeme
    - Periodische und sporadische Aufgaben (engl. *tasks*)
    - Einplanung und Koordination
  - Anwendung dieser Konzepte innerhalb von Echtzeitanwendungen
    - Wie beeinflussen diese Konzepte das Ablaufverhalten?
    - Wie implementieren Echtzeitbetriebssysteme diese Konzepte?

- Vorlesung: Vorstellung und detaillierte Behandlung des Stoffs
  - Grundlagen von Echtzeitsystemen
    - Zeit- und ereignisgesteuerte Systeme
    - Periodische und sporadische Aufgaben (engl. *tasks*)
    - Einplanung und Koordination
  - Anwendung dieser Konzepte innerhalb von Echtzeitanwendungen
    - Wie beeinflussen diese Konzepte das Ablaufverhalten?
    - Wie implementieren Echtzeitbetriebssysteme diese Konzepte?
- Übung: Vertiefung und praktische Anwendung
  - Anwendungs- und Systemprogrammierung (Software-Oszilloskop)
  - Ablaufverhalten durch das EZ-Betriebssystem beeinflussen
  - Werkzeuge aus dem industriellen Umfeld einsetzen

# Bedeutung von Tafel- und Rechnerübungen

- Tafelübungen  $\leadsto$  „*learning by exploring*“
  - Besprechung der Übungsaufgaben, Skizzierung von Lösungswegen
  - Vertiefung des Vorlesungsstoffes, Klärung offener Fragen

# Bedeutung von Tafel- und Rechnerübungen

- Tafelübungen  $\leadsto$  „*learning by exploring*“
    - Besprechung der Übungsaufgaben, Skizzierung von Lösungswegen
    - Vertiefung des Vorlesungsstoffes, Klärung offener Fragen
  - Rechnerarbeit  $\leadsto$  „*learning by doing*“
    - Selbstständiges Bearbeiten der Übungsaufgaben am Rechner
      - Abgabe der bearbeiteten Übungsaufgaben
      - Klärung von Unklarheiten/Problemen bei/mit den Übungsaufgaben
    - Rechner ist allerdings **kein Tafelersatz**
- Bereitet euch vor! Wir erwarten konkrete Fragen!

# Bedeutung von Tafel- und Rechnerübungen

- Tafelübungen  $\leadsto$  „*learning by exploring*“
    - Besprechung der Übungsaufgaben, Skizzierung von Lösungswegen
    - Vertiefung des Vorlesungsstoffes, Klärung offener Fragen
  - Rechnerarbeit  $\leadsto$  „*learning by doing*“
    - Selbstständiges Bearbeiten der Übungsaufgaben am Rechner
      - Abgabe der bearbeiteten Übungsaufgaben
      - Klärung von Unklarheiten/Problemen bei/mit den Übungsaufgaben
    - Rechner ist allerdings **kein Tafelersatz**
- Bereitet euch vor! Wir erwarten konkrete Fragen!

Der, die, das.

Wer, wie, was?

Wieso, weshalb, warum?

Wer nicht fragt, bleibt dumm!



# Voraussetzungen

- **Systemprogrammierung**, Grundlagen der Informatik
- **C / C++**, Java
- Ein gewisses Maß an **Durchhaltevermögen**
- Freude an systemnaher und **praktischer Programmierung**

Wir arbeiten mit **eingebetteten Systemen!**

# Voraussetzungen

- **Systemprogrammierung**, Grundlagen der Informatik
- **C / C++**, Java
- Ein gewisses Maß an **Durchhaltevermögen**
- Freude an systemnaher und **praktischer Programmierung**

Wir arbeiten mit **eingebetteten Systemen!**

Nicht von *hardwarenaher Programmierung* abschrecken lassen!



# Voraussetzungen

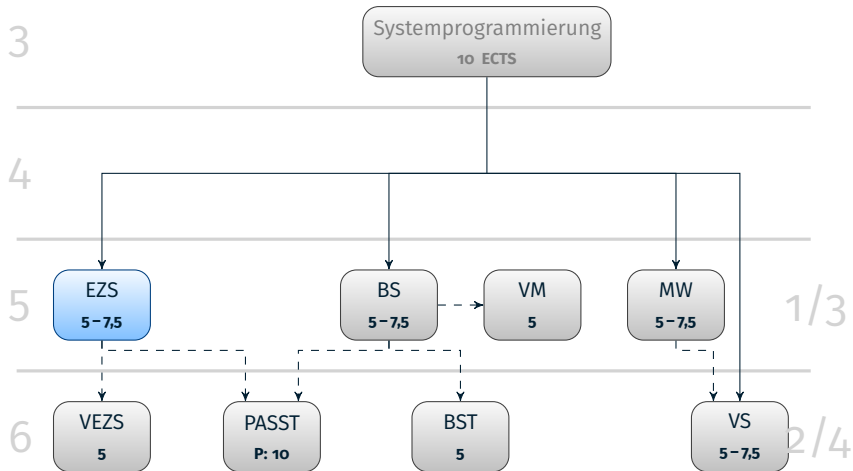
- **Systemprogrammierung**, Grundlagen der Informatik
- **C / C++**, Java
- Ein gewisses Maß an **Durchhaltevermögen**
- Freude an systemnaher und **praktischer Programmierung**

Wir arbeiten mit **eingebetteten Systemen!**

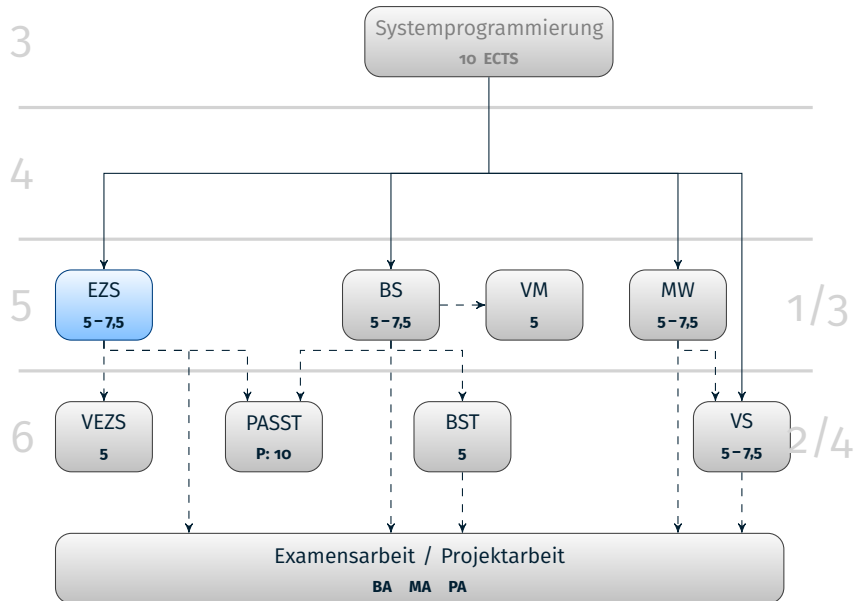
Nicht von *hardwarenaher Programmierung* abschrecken lassen!

**Test für Programmierkenntnisse**  Webseite: Fragebogen

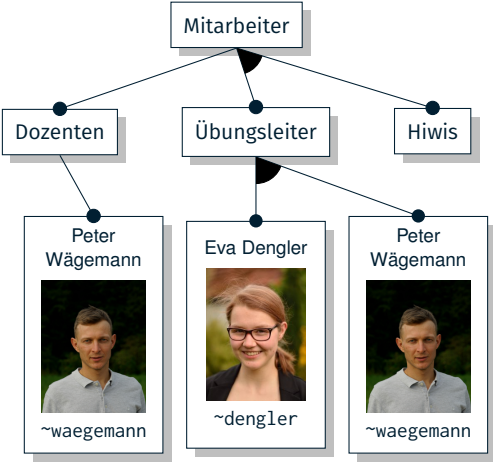
# Einpassung in den Studienplan



# Einpassung in den Studienplan



- 1 Lehrveranstaltungskonzept & Organisation
- 2 Die Veranstaltung
  - Lernziele und Voraussetzungen
  - Einordnung
- 3 Organisatorisches**
  - Die Beteiligten
  - Vorlesung und Übung
  - Leistungsnachweise
  - Literaturempfehlungen



## Vorlesung: Zeit und Ort

- Dienstag, 8:30 – 10:00, H4

## Vorlesung: Zeit und Ort

- Dienstag, 8:30 – 10:00, H4

Änderungen und Hinweise: siehe Webseite bzw. Mailingliste

## Vorlesung: Zeit und Ort

- Dienstag, 8:30 – 10:00, H4

Änderungen und Hinweise: siehe Webseite bzw. Mailingliste

- Vorlesungsfolien sind verfügbar wie folgt:
  - <https://sys.cs.fau.de/lehre/ss24/ezs/vorlesung>



## Vorlesung: Zeit und Ort

- Dienstag, 8:30 – 10:00, H4

**Änderungen und Hinweise:** siehe Webseite bzw. Mailingliste

- Vorlesungsfolien sind verfügbar wie folgt:
  - <https://sys.cs.fau.de/lehre/ss24/ezs/vorlesung>
- Wosch's Glossar :
  - Relevante Begriff der Informatik außerhalb des Vorlesungskerns
  - Als Wiederholung (Inf) beziehungsweise zum Einstieg ( $\neg$ Inf)
  - <https://www4.cs.fau.de/~wosch/glossar.pdf>

## Vorlesung: Zeit und Ort

- Dienstag, 8:30 – 10:00, H4

Änderungen und Hinweise: siehe Webseite bzw. Mailingliste

- Vorlesungsfolien sind verfügbar wie folgt:
  - <https://sys.cs.fau.de/lehre/ss24/ezs/vorlesung>
- Wosch's Glossar :
  - Relevante Begriff der Informatik außerhalb des Vorlesungskerns
  - Als Wiederholung (Inf) beziehungsweise zum Einstieg ( $\neg$ Inf)
  - <https://www4.cs.fau.de/~wosch/glossar.pdf>
- Literaturempfehlungen siehe Folie 17

## Tafelübung: Zeit und Ort

- Dienstag, 10:15 – 11:45,  
Aquarium

## Rechnerübung: Zeit und Ort

- Dienstag, 12:15 – 13:45, CIP1
- Mittwoch, 12:15 – 13:45, CIP1

## Tafelübung: Zeit und Ort

- Dienstag, 10:15 – 11:45, Aquarium

## Rechnerübung: Zeit und Ort

- Dienstag, 12:15 – 13:45, CIP1
- Mittwoch, 12:15 – 13:45, CIP1

- Übung
  - Übungsaufgaben sind bevorzugt in Gruppen zu bearbeiten
  - Tafel- und Rechnerübung
  - Rechnerarbeit: größtenteils in Eigenverantwortung

# Studien- und Prüfungsleistungen (1)

VL – Vorlesung

**2,5**

Vorstellung und detaillierte Behandlung des Lehrstoffs

# Studien- und Prüfungsleistungen (1)

VL – Vorlesung

2,5

Vorstellung und detaillierte Behandlung des Lehrstoffs

+

**Ü – Übung**

2,5

- Praktische Übungen
- 7 Übungsaufgaben
- Abnahme alle 14 Tage

# Studien- und Prüfungsleistungen (1)

VL – Vorlesung

2,5

Vorstellung und detaillierte Behandlung des Lehrstoffs

+

**Ü – Übung** 2,5

- Praktische Übungen
- 7 Übungsaufgaben
- Abnahme alle 14 Tage

oder

**EÜ – Erweiterte Übung** 5

- Übung (Ü)
- + erweiterte Aufgaben
- + vertiefende Abfrage

+

**RÜ – Rechnerübung**

0

- **Betreutes** Arbeiten am Rechner
- Hilfe zu eCos, Oszilloskop, aiT, ...

## Studien- und Prüfungsleistungen (2)

### ■ **Wahlpflichtmodul** (Bachelor/Master) der Vertiefungsrichtung **Verteilte Systeme und Betriebssysteme**

- eigenständig (nur EZS)
- mit weiteren Veranstaltungen

VL + Ü oder VL + EÜ  
siehe Modulhandbuch

### ■ Studien- und Prüfungsleistungen

- Bachelor
- Master

Prüfungsleistung  
Prüfungsleistung

erworben durch

- erfolgreiche Teilnahme an den Übungen
- erfolgreiche Bearbeitung aller Übungsaufgaben
- 30 min. (bzw. 20 min) mündliche Prüfung

### ■ Berechnung der Modulnote

- Note der mündlichen Prüfung + “Übungsbonus” in Zweifelsfällen



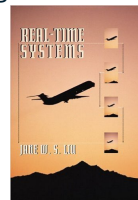


## Wanted:

- Bachelor- und Masterarbeiten
- Bachelor-Praktikum und Master-Projekte
- studentische Hilfwissenschaftler (Hiwis)

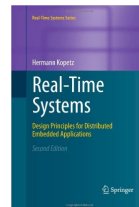
[2] Eine hervorragende Begleiterin der Veranstaltung:

Jane W. S. Liu. *Real-Time Systems*.  
**Prentice Hall PTR, Englewood Cliffs, NJ,  
USA, 2000**



[1] Der „Klassiker“ für zeitgesteuerte EZS:

Hermann Kopetz. *Real-Time Systems: Design Principles for Distributed Embedded Applications*.  
**Kluwer Academic Publishers, first edition  
edition, 1997**



42