

Entwurf von Echtzeitsystemen

Prof. Dr. Kay Römer
WS 2014/15

Organisatorisches

- **Vorlesung** Termin: Do, 11.00-13.00,
keine Vorlesung am 9.10., 23.10.
- **Ort:** HS i12, Inffeldgasse 16b
- **Skriptum:** Vorlesungsfolien jeweils
vor Beginn der Vorlesung online
- **Prüfung:** Schriftlich im Anschluss an Vorlesung
- **Unterlagen in TeachCenter:**
<http://tugtc.tugraz.at/wbtmaster/courseMain.htm?448023>



Übungen / Labor

■ Übungen „Entwurf von Echtzeitsystemen“

- Schriftliche Übungen
- Inhaltlich abgestimmt auf die Vorlesung, d.h. Vorlesung besuchen wenn Sie Übungen belegen!

■ Labor „Entwurf von Echtzeitsystemen“

- Praktische Übungen mit SPS
- Besuch der Vorlesung als Vorbereitung ratsam!
- Einleitender Kurs mit anschließender Bearbeitung eines Projektes

1. EINLEITUNG

Was sind eingebettete Systeme ?

Definition [W. Wolf 2000]:

- Embedded system: any device that includes a programmable computer but is not itself a general-purpose computer.
- Take advantage of application characteristics to optimize the design:
 - don't need all the general-purpose bells and whistles.

Definition [P. Marwedel 2003]:

Information processing systems embedded into a larger product.

Main reason for buying is **not** information processing

Anwendungsbereiche für Eingebettete Echtzeitsysteme

■ Verkehrstechnik

■ Automotive



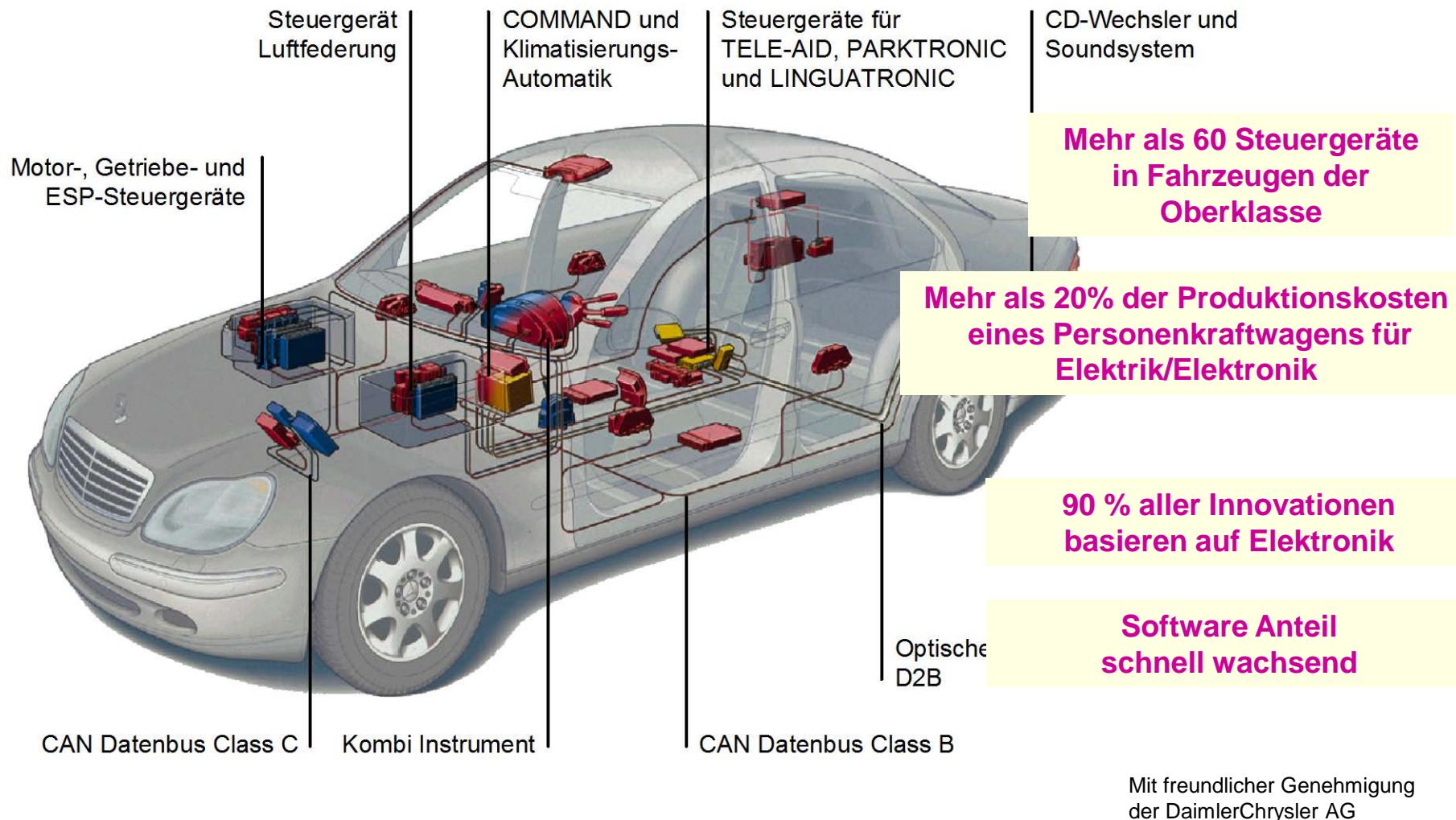
■ Avionik

■ Bahn



■ Schifffahrt

Beispiel: Vernetzte Steuergeräte im Auto



■ Telekommunikation



■ Medizintechnik



Narkosegerät

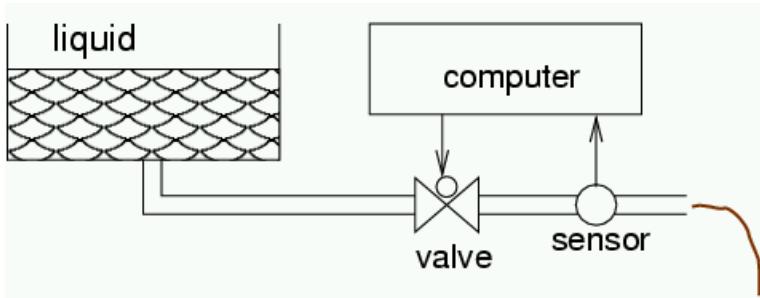


Röntgen C-Bogen

■ Consumer Elektronik



■ Industrielle Automation



■ Smart buildings



■ Robotik



ASIMO



Echtzeitsysteme

Eingebettete Systeme sind oftmals gleichzeitig Echtzeitsysteme



Echtzeitverarbeitung:
Die Betriebsart eines
Rechensystems, bei der die
Programme für die Verarbeitung von
Daten, die von außen eintreffen,
permanent betriebsbereit sind, so
dass ihre Ergebnisse innerhalb
einer *vorbestimmten Zeitperiode* zur
Verfügung stehen; die
Ankunftszeiten der Daten können
anwendungsabhängig *zufällig verteilt* sein oder *a priori* feststehen
[nach DIN 44300].

Echtzeitsystem:

System, das explizite (begrenzte) *Antwortzeitbedingungen* einhalten muss, um nicht ernste Konsequenzen zu riskieren, einschließlich einem Systemausfall.

Ein System ist *ausgefallen*, wenn es eine oder mehrere Anforderungen, die in der formalen Systemspezifikation festgelegt sind, nicht erfüllt [Laplante 97].

Harte Echtzeitsysteme (hard real-time systems):

Verletzung der Zeitbedingungen führt unweigerlich zum Systemausfall.

Weiche Echtzeitsysteme (soft real-time systems):

Verletzung der Zeitbedingungen führt zur Leistungsminderung, aber nicht unbedingt zum Systemausfall.

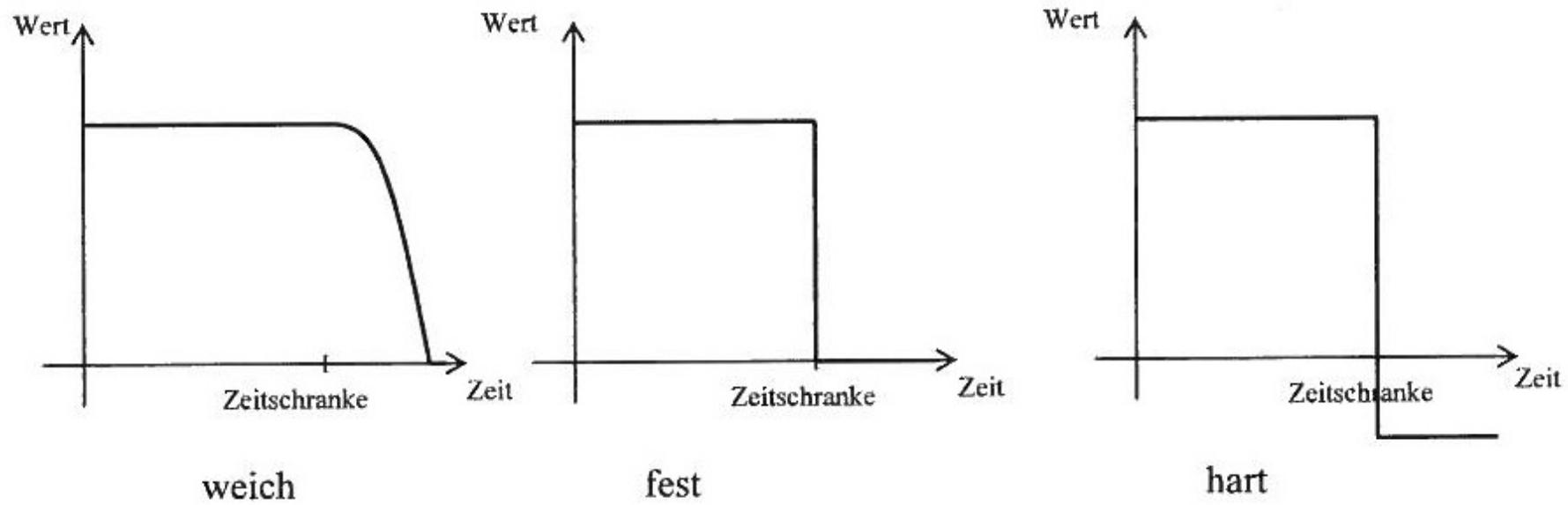
Harte und weiche Echtzeitumgebungen:

Real-time environment	Hard	Soft
Consequences of not responding in time	useless results, or danger for people or the process	costs increasing with waiting time, quality degradation
Examples	car driver, housewife power plant flight guidance system steel mill	beginning of a party airline reservation bank library system
Analysis objectives	worst case performance	average performance

Echtzeitbedingungen

weich: Zeitschranke darf in gewissem Rahmen überschritten werden

hart: Zeitschranke muss auf jeden Fall eingehalten werden, sonst entsteht Schaden.



fest: Aktion wird bei Überschreiten der Zeitschranke wertlos, aber kein Schaden („Verfallsdatum“, z. B. bei Überwachungsaufgaben). *Weniger gebräuchlich*.

Merkmale von Echtzeitsystemen [Zöbel 08]

Korrektheit

Funktionale und zeitliche Richtigkeit des Rechenergebnisses

Rechtzeitigkeit

Einhalten von Zeitbedingungen

Gleichzeitigkeit

Erledigung von vielen parallelen Aufgaben unter Einhaltung von jeweils eigenen Zeitanforderungen.

Determiniertheit

Verhalten des Systems unter allen äußeren Bedingungen *eindeutig* im Voraus bestimmbar.

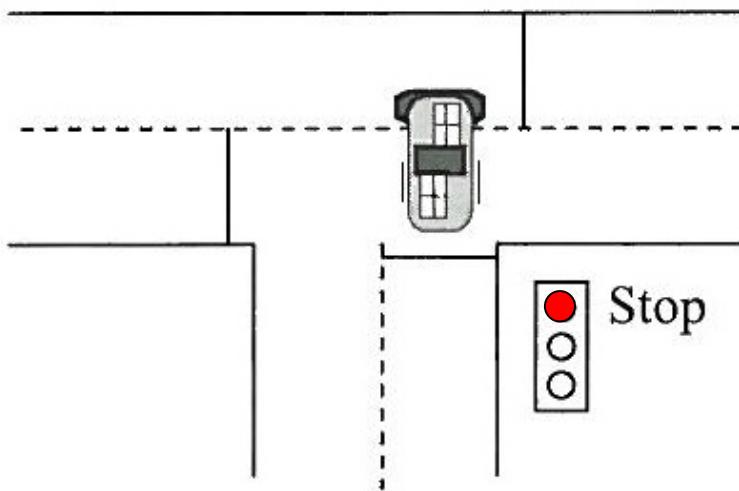
Für viele komplexe Systeme eine zu strenge Forderung!

Vorhersagbarkeit genügt, d. h. funktionales und zeitliches Verhalten muss in seiner *Wirkung* abschätzbar bleiben (z. B. Auswirkung von Fehlern).

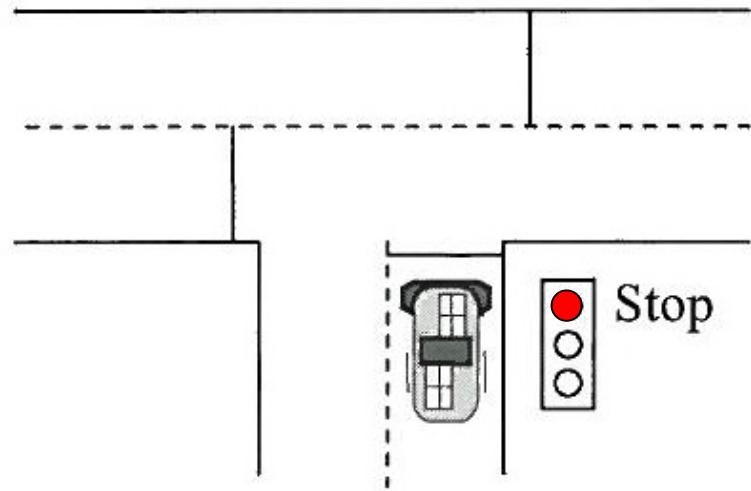
Korrektheit

Nicht-Echtzeitsysteme: logische Korrektheit → Korrektheit

Echtzeitsysteme: logische + zeitliche Korrektheit → Korrektheit



logisch korrekt, zeitlich inkorrekt

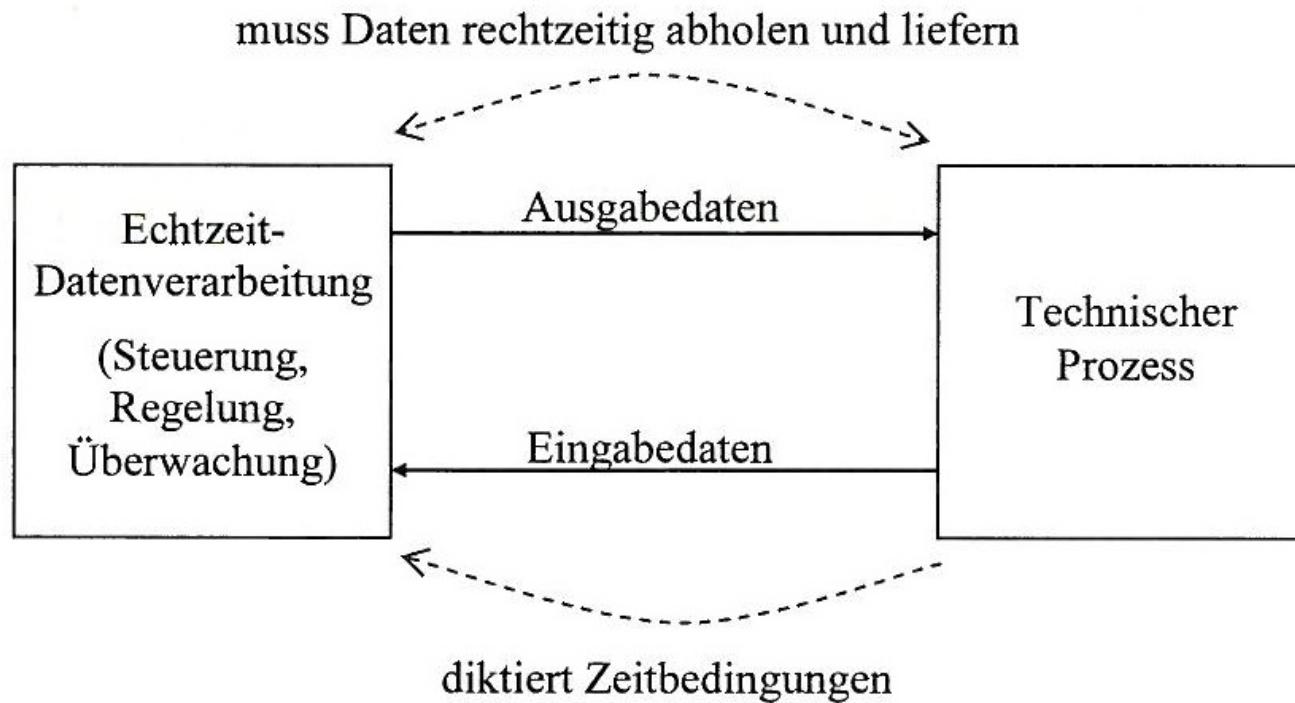


logisch und zeitlich korrekt

Bei Echtzeitsystemen muss neben der logischen auch die *zeitliche Korrektheit* gewährleistet sein, daher eigene Programmiertechniken erforderlich.

Rechtzeitigkeit

Ausgabedaten müssen *rechtzeitig* berechnet sein und zur Verfügung stehen, Eingabedaten müssen *rechtzeitig* abgeholt werden.

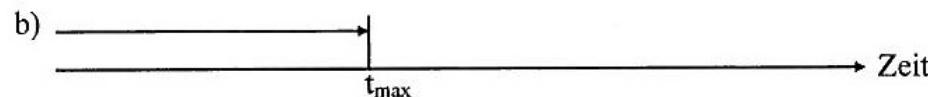


Die Zeitbedingungen werden durch den technischen Prozess festgelegt und sind damit *abhängig von der jeweiligen Anwendung*.

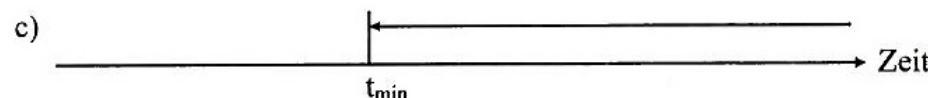
Zeitbedingungen



Exakter Zeitpunkt



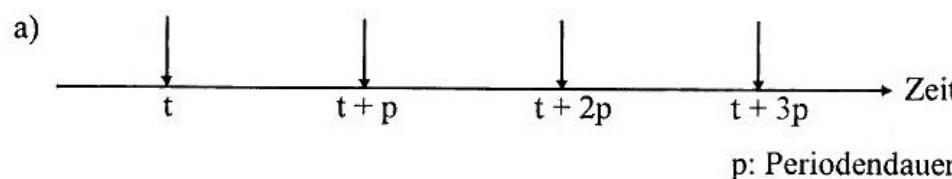
Spätester Zeitpunkt (Deadline)



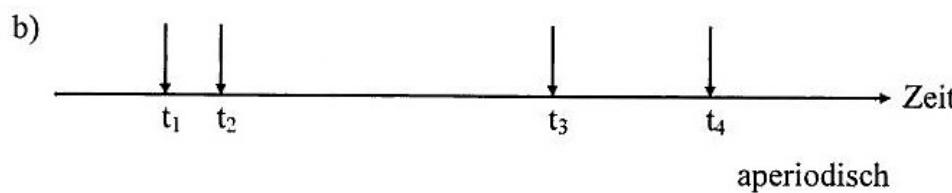
Frühester Zeitpunkt



Zeitintervall



Periodisch



Aperiodisch

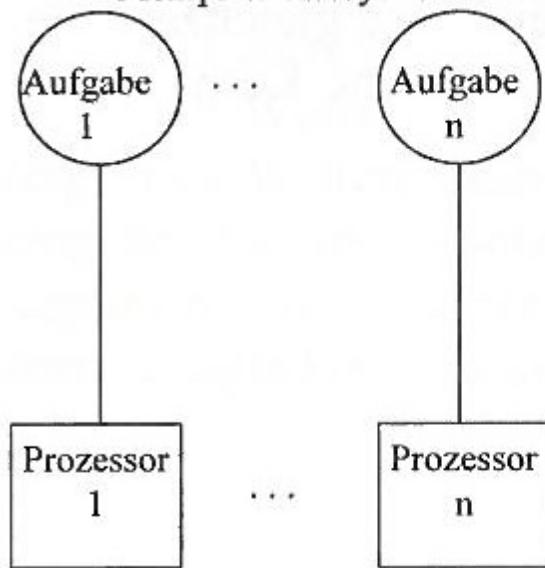
Vorlesungsbeginn um 10.15 Uhr

Vorlesungsende 90 Minuten nach Beginn

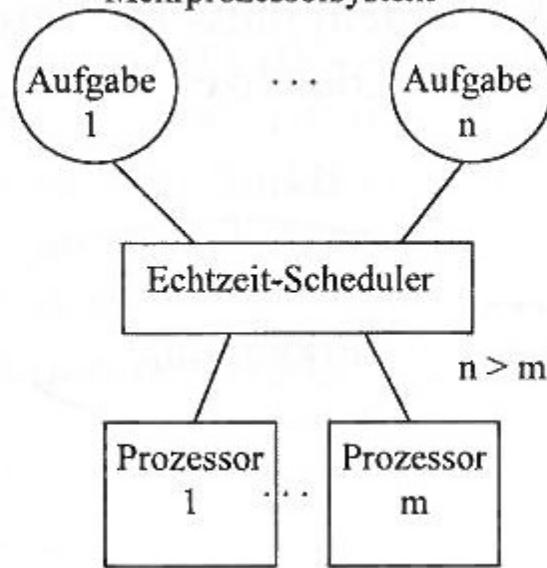
Absolut
Relativ

Gleichzeitigkeit

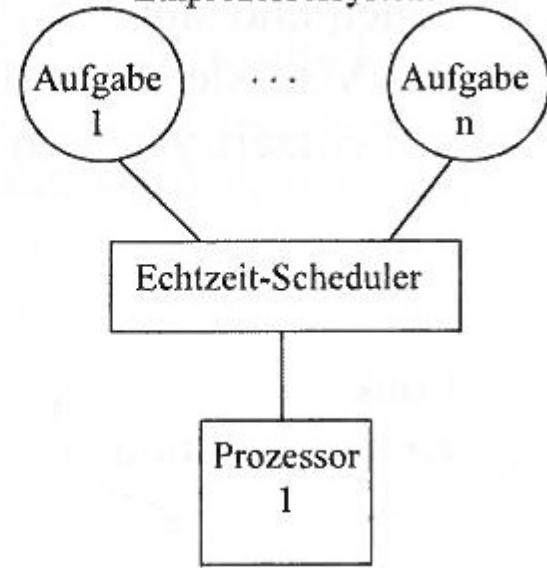
Vollständige Parallelverarbeitung,
Mehrprozessorsystem



Quasi-Parallelverarbeitung,
Mehrprozessorsystem



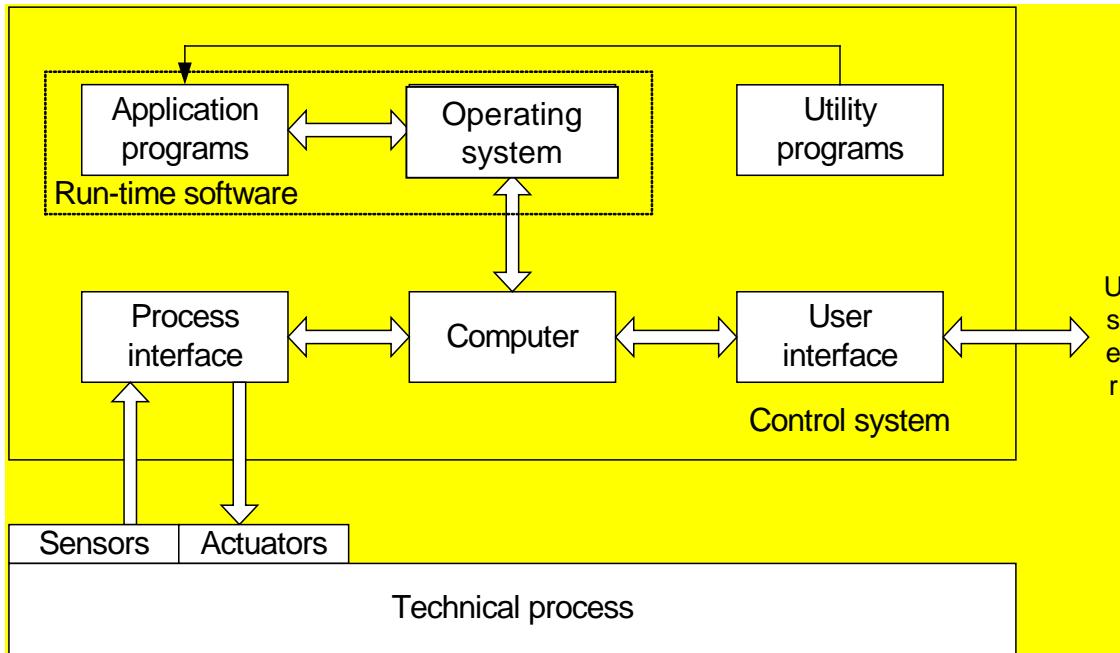
Quasi-Parallelverarbeitung,
Einprozessorsystem



Parallelität im technischen Prozess muss durch echte oder quasi-parallele Verarbeitung realisiert werden (unter Berücksichtigung der Zeitbedingungen).

Echtzeit-Scheduler zur Ablaufplanung bei Quasi-Parallelverarbeitung.

Prinzipieller Aufbau Eingebetteter Echtzeitsysteme



Schnittstellen

zum *technischen Prozess*
(Sensoren/Aktoren)
zum *Benutzer* (User)

Computersystem kann ein
Mikrocontroller, ein
Parallelrechner oder ein räumlich
verteiltes System sein.

Software (Betriebssystem,
Anwendungsprogramme, Utilities
etc.) müssen den
Zeitanforderungen für
Echtzeitsysteme genügen.

Standard-Software meist nicht ohne weiteres einsetzbar (d.h. spezielle Echtzeitsoftware)!

Meilensteine in der Historie Eingebetteter Echtzeitsysteme

- 1930-50: Frühe Computer zu groß und energieintensiv für eingebettete Systeme
- 196X: Speicherprogrammierbare Steuerungen (getrieben durch US Automobilindustrie)
- 1961: Minuteman Steuercomputer (Transistorlogik, Festplatte), erstmals „Massenfertigung“
- 196X: Apollo Steuercomputer (4100 ICs mit je 3 NOR Gattern)
- 1967: Minuteman Steuercomputer, erstmals Massenfertigung von ICs (Preisreduktion um Faktor 300 ermöglicht erste kommerzielle Produkte)
- 1971: Intel 4004, erster integrierter 4-Bit Mikroprozessor
- 1980: Analoge Eingabeelemente weitgehend durch digitale Knöpfe und Mikrocontroller verdrängt
- Heute: Weitgehende Durchdringung unserer Alltagswelt durch eingebettete Systeme

Aktuelle Trends

- Eingebettete Echtzeitsysteme dringen in immer mehr Produkte vor (Massenmarkt)
- *Systematischer, weitgehend automatisierter Entwurf (Werkzeuge vergleichbar dem Chipentwurf)*
- Anwendungen werden immer komplexer, Systeme immer autonomer (z. B. mobile Roboter)
- *'Intelligente' Techniken (Computational Intelligence, Organic Computing), die das Verhalten von biologischen Systemen nachahmen (z. B. Fuzzy, Neuronale Netze, Neuro-Fuzzy, KI)*
- Echtzeitsysteme/eingebettete Systeme werden zunehmend in kritischen Anwendungsgebieten eingesetzt, wo ein Ausfall katastrophale Folgen haben kann (z. B. 'Fly-by-wire', 'Drive-by-wire', medizinische Geräte, ...).
- *Techniken zur Erhöhung der Verlässlichkeit (Dependability), insbesondere Fehlertoleranz*

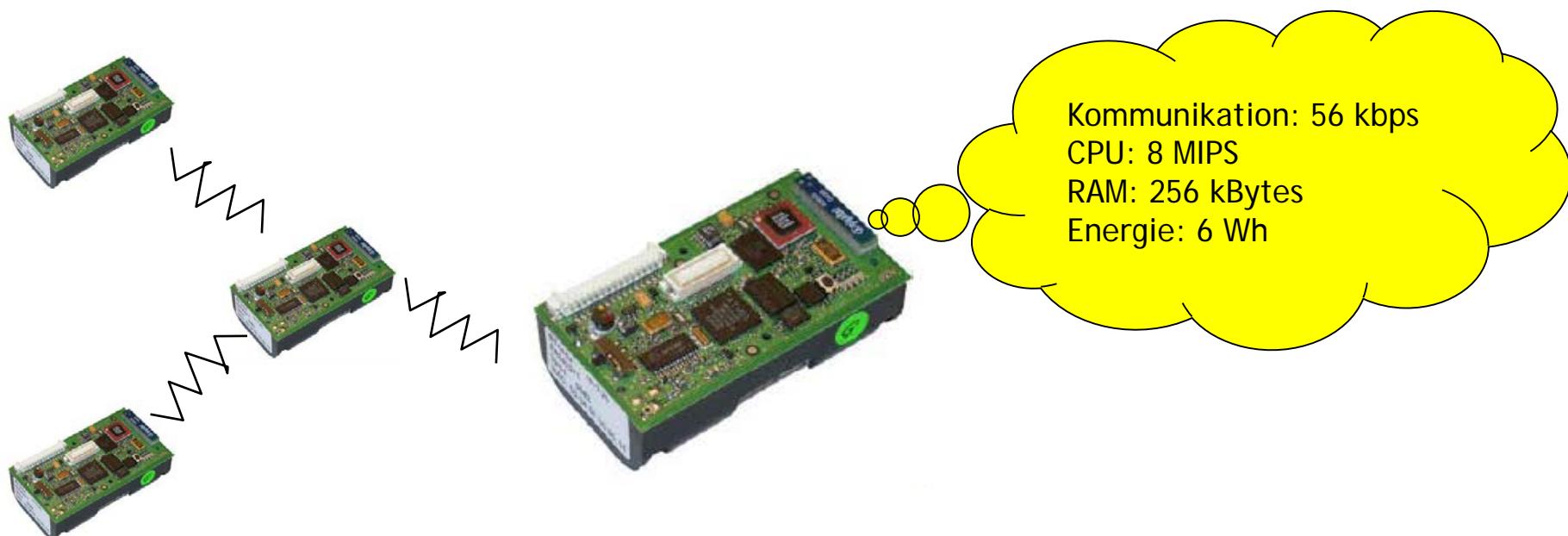
- Mobile/tragbare Geräte mit Funkvernetzung und Internetzugriff
(Handies, GPS-Navigationssysteme, MP3-Player, Digitalkameras etc.)
- *Low-Power Design, Miniaturisierung durch SoCs (System on Chip), rekonfigurierbare Logik (z.B. FPGA)*
- Ubiquitäre Systeme: Alltagsgegenstände mit Rechnern versehen und vernetzt
(Sensornetze, Smart Dust, Ubiquitous Computing, Internet der Dinge)
- *(Funk-)Kommunikation und Sensorik/Aktorik wachsen zusammen*

95% aller Mikroprozessoren arbeiten heute schon in eingebetteten Systemen, Tendenz stark steigend.

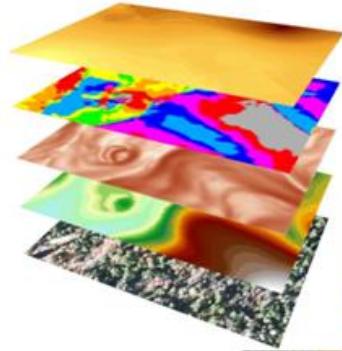
90% allen Programmcodes wird für eingebettete Systeme geschrieben.

Beispiel: Drahtlose Sensornetze

- Drahtlose Netze von Sensorknoten
 - Sensoren → Umgebungswahrnehmung
 - Prozessoren → Verarbeitung
 - Funkmodul → drahtlose Kommunikation



Drahtlose Sensornetze: Anwendungen



Enable New Knowledge



Preventing Failures



Improve Food

Save Resources



Improve Productivity



Enhance Safety & Security



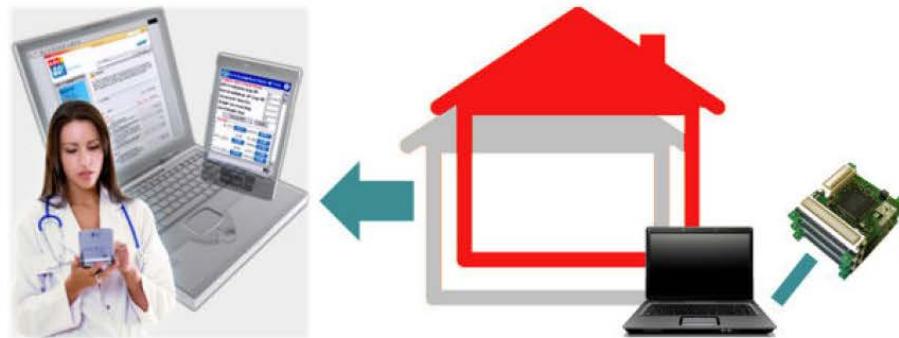
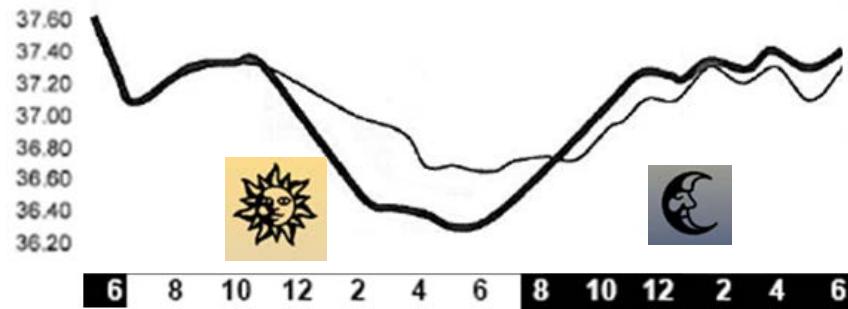
High-Confidence Transport

Protect Health

Drahtlose Sensornetze am Körper



Einfluss von Schlafmangel auf
Regulation der Körpertemperatur



MEDIC
(Laptop / PDA)

HOME
(Base Station)

**SINK
NODE**
(25mm)

**SENSOR
NODES**
(10mm)

INHALT

1. Einleitung

2. Zielhardware

- Digitale Systemkomponenten
- Systemarchitekturen
- SPS
- Prozess-Schnittstellen
- Echtzeit-Kommunikationsnetze

3. System Software

- Betriebssysteme
- Echtzeit-Scheduling
- Synchronisation
- Echtzeit-Programmiersprachen

4. Konzeptionelle Modelle

- Lebenszyklus
- Modelltaxonomie
- Zustandsorientierte Modelle
- Aktivitätsorientierte Modelle
- Strukturorientierte Modelle
- Heterogene Modelle

5. Spezifikationssprachen

- Anforderungen
- Statecharts
- Echtzeit-UML

6. Spezifikation → Implementierung

- Umsetzung in Programmiersprachen
- System-Partitionierung
- Spezifikationsverfeinerung

Literatur

Marwedel, P.: Eingebettete Systeme – Eine Einführung, Springer-Verlag, Berlin 2007, ISBN 978-3-540-34048-5

Wolf, W.: Computers as Components – Principles of Embedded Computing Systems.
Morgan Kaufmann Publishers, Academic Press, San Diego 2001, ISBN 1-55860-693-9

H. Woern, U. Brinkschulte: Echtzeitsysteme, Springer-Verlag, Berlin, 2005, ISBN 3-540-20588-8

Besuchen Sie regelmäßig die Vorlesung!

- Interaktive Wiederholung der wesentlichen Aspekte
- Erläuterung von Details
- Beispiele an der Tafel
- Deutlich besseres Abschneiden in Prüfungen!