

# Treiberentwicklung, Echtzeit- und Betriebssysteme

Prof. Dr. rer. nat. Peter Gerwinski

30. Mai 2022

## 4 Echtzeit

### 4.1 Was ist Echtzeit?

- Animation in Echtzeit:  
schnelle Berechnung anstatt Wiedergabe einer Aufzeichnung
- Fantasy-Rollenspiel in Echtzeit:  
Der Zeitverlauf der Spielwelt entspricht dem der realen Welt.
- Datenverarbeitung in Echtzeit:  
Die Daten werden so schnell verarbeitet, wie sie anfallen.
- speziell: Echtzeit-Steuerung von Maschinen:  
Die Berechnung kann mit den physikalischen Vorgängen schritthalten.

→ „Schnell genug.“

## 4.1 Was ist Echtzeit?

„Schnell genug.“ – „Und wenn nicht?“

- „Ganz schlecht.“ → *harte Echtzeit*  
rechtzeitiges Ergebnis funktionsentscheidend
- „Unschön.“ → *weiche Echtzeit*  
verspätetes Ergebnis qualitätsmindernd
  - verwenden und Verzögerung in Kauf nehmen
  - verwerfen und Ausfall in Kauf nehmen
- „Es gibt keinen festen Termin. Möglichst schnell halt.“  
→ *keine Echtzeit*

## 4.1 Was ist Echtzeit?

Das Problem:

Beispiel:

- Eine Motorsteuerung benötigt alle  $2000\text{ }\mu\text{s}$  einen Steuerimpuls, dessen Berechnung maximal  $10\text{ }\mu\text{s}$  dauert.
- Entweder: Der Steuer-Computer macht noch andere Dinge.  
→ Risiko der Zeitüberschreitung
- Oder: Der Steuer-Computer macht nichts anderes.  
→ Verschwendung von Rechenzeit  
→ Na und?

## 4.1 Was ist Echtzeit?

Das Problem:

### „Verschwendung von Rechenzeit – na und?“

Große Stückzahlen

- 138 000 Toyota Prius V von Mai 2011 bis April 2012
- Software viel im Einsatz: Server, Rechenzentrum

Wertvolle Ressourcen

- Fähigkeiten einer Raumsonde optimieren
- Implantat: Platz- und Stromverbrauch minimieren

### → **Echtzeitprogrammierung**

Echtzeitanforderungen erfüllen, ohne Ressourcen zu verschwenden

## 4 Echtzeit

### 4.2 Multitasking

#### Qualitätssicherung beim Multitasking

- Verschiedene Anforderungen:  
*Latenz* vs. *Jitter*  
vs. *Durchsatz*
- Ressourcen reservieren:  
*Mutexe* (= spezielle *Semaphore*)
- Verschiedene Methoden  
der Priorisierung
- Umgehung der Probleme durch  
speziell geschriebene Software  
(MultiWii, RP6, ...)

#### Qualitätssicherung für Netzwerke:

- Verschiedene Anforderungen:  
*Latenz* vs. *Jitter* vs. *Verluste*  
vs. *Durchsatz*
- Ressourcen reservieren:  
*IntServ* mit *Resource Reservation Protocol (RSVP)*
- Klassifizierung und Priorisierung:  
*DiffServ* mit Type-of-Service-Bits  
(IPv4) bzw. Traffic-Class-Bits (IPv6)  
im IP-Header
- Eigenes Protokoll (Telefondienste):  
*Asynchronous Transfer Mode (ATM)*

## Beispiele für Multitasking

### Quadrocopter-Steuerung *MultiWii*

- Konfiguration durch bedingte Compilierung (Präprozessor)
- In der Hauptschleife wird 50mal pro Sekunde der RC-Task aufgerufen, ansonsten zyklisch einer von bis zu 5 weiteren Tasks.

### RP6-Steuerung

- ...