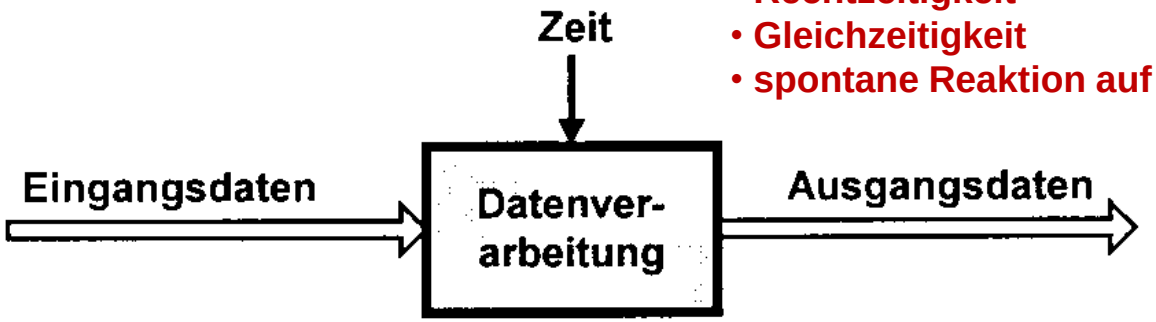
Kapitel 1

Echtzeitprogrammierung

Vergleich Nicht-Echtzeitsystem mit Echtzeitsystem

- Bei ***Nicht-Echtzeitsystemen*** kommt es ausschließlich auf die **logische Korrektheit** der Ergebnisse an

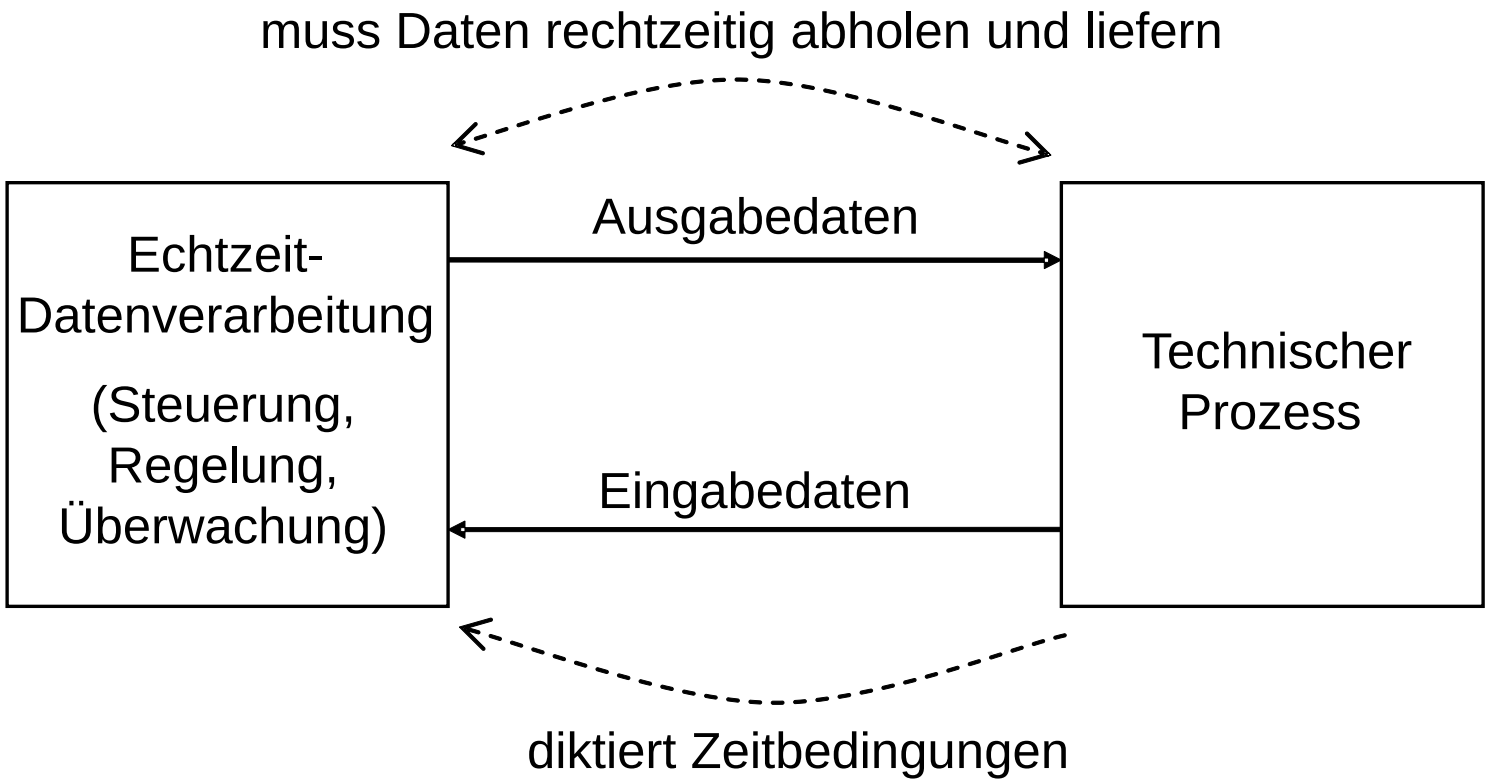
- Bei ***Echtzeitsystemen*** kommt neben der **logischen Korrektheit** auch die **zeitliche Korrektheit** der Ergebnisse



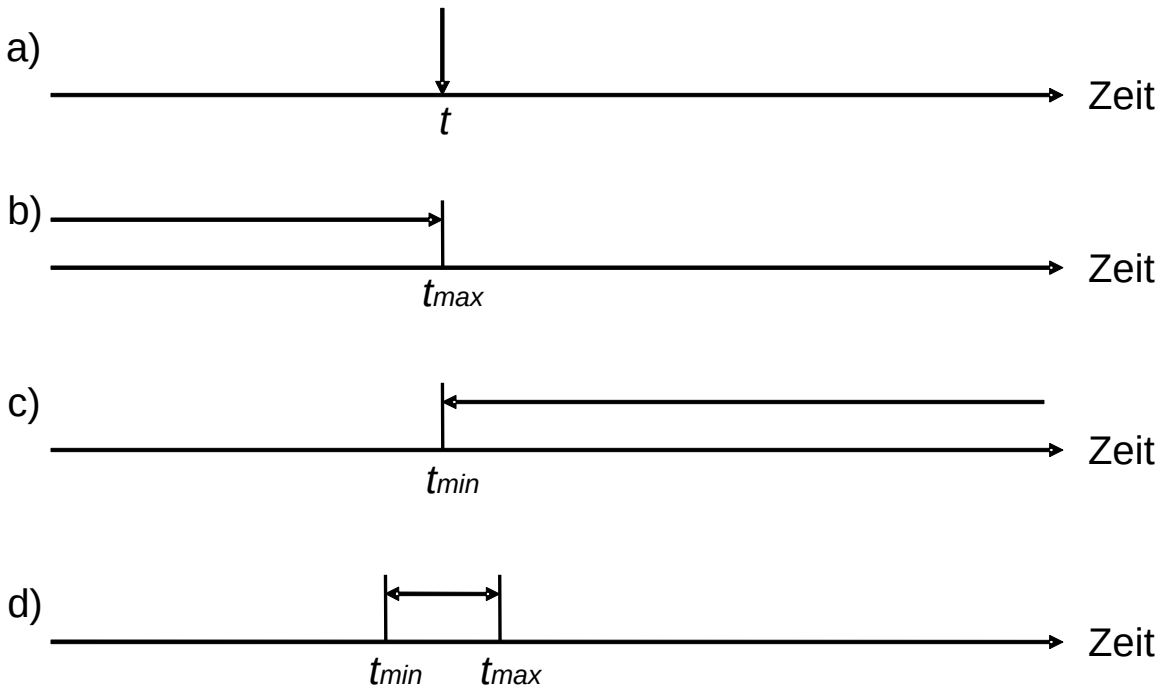
- **Rechtzeitigkeit**
- **Gleichzeitigkeit**
- **spontane Reaktion auf Ereignisse**

Zusammenspiel von Echtzeit-Datenverarbeitung und technischem Prozess

Rechtzeitigkeit heißt, die Ausgabedaten müssen rechtzeitig berechnet werden und zur Verfügung stehen.

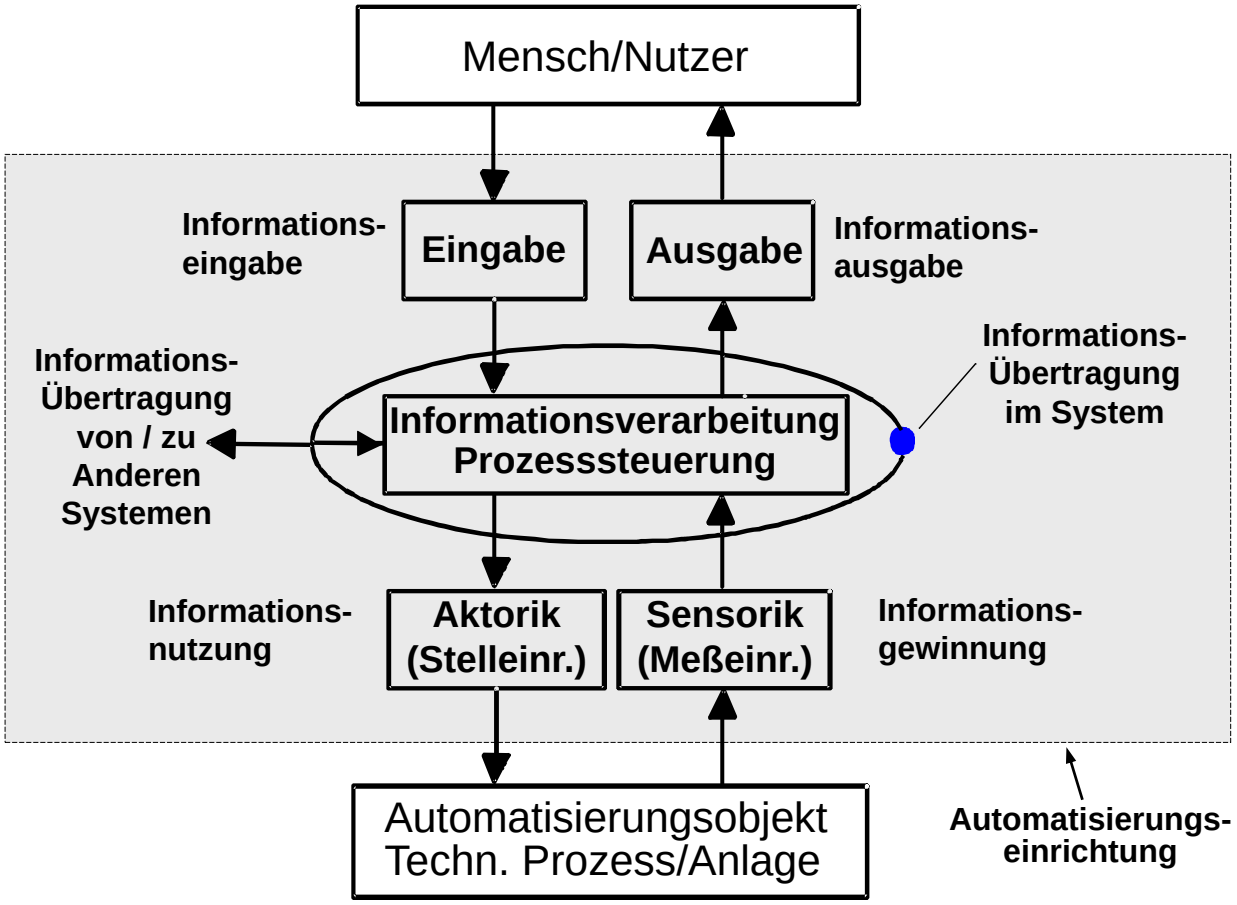


Varianten zur Angabe einer Zeitbedingung



a) Angabe eines genauen Zeitpunktes
b) Angabe eines spätesten Zeitpunktes
c) Angabe eines frühesten Zeitpunktes
d) Angabe eines Zeitintervalls

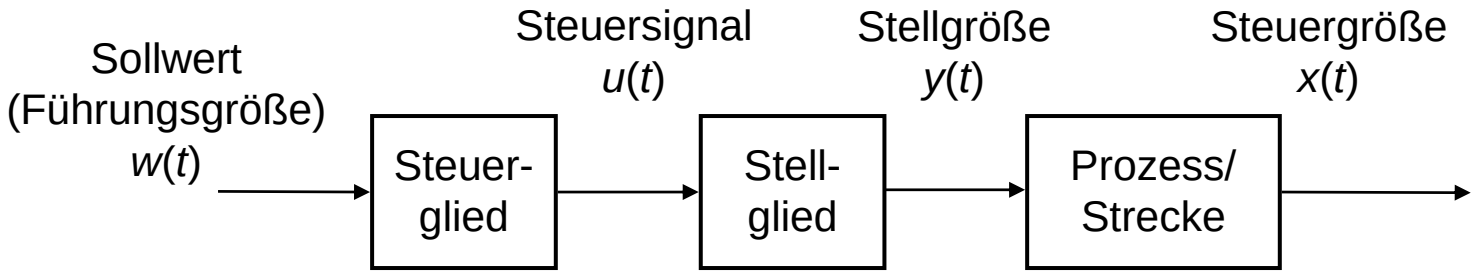
Grundprinzip der Automatisierung



Informationsbezogene Funktionalität des Informationsmanagements (Grundschema der Automation)

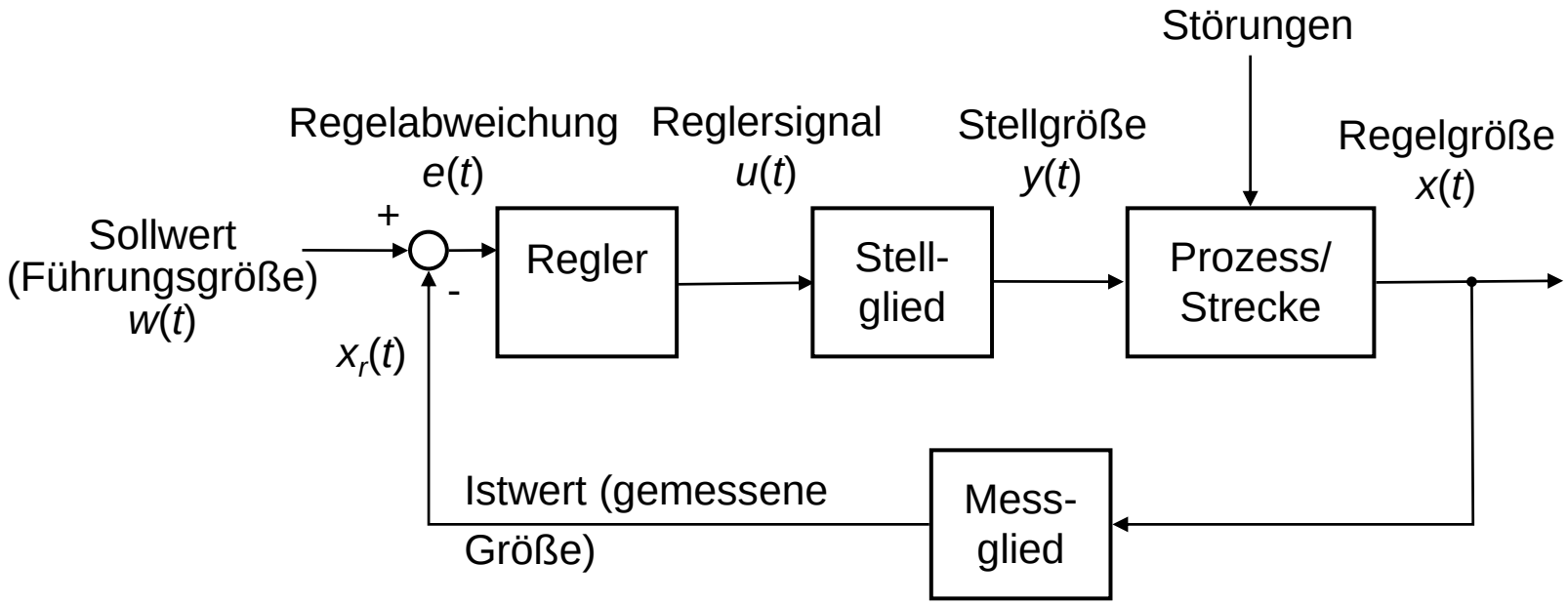
Wirkungskette einer Steuerung

Nach DIN 19226 ist die **Steuerung** definiert: Die Steuerung ist ein Vorgang in einem abgegrenzten System, bei dem eine oder mehrere Größen als Eingangsgrößen andere Größen als Ausgangsgrößen aufgrund der dem System eigenen Gesetzmäßigkeiten beeinflussen.

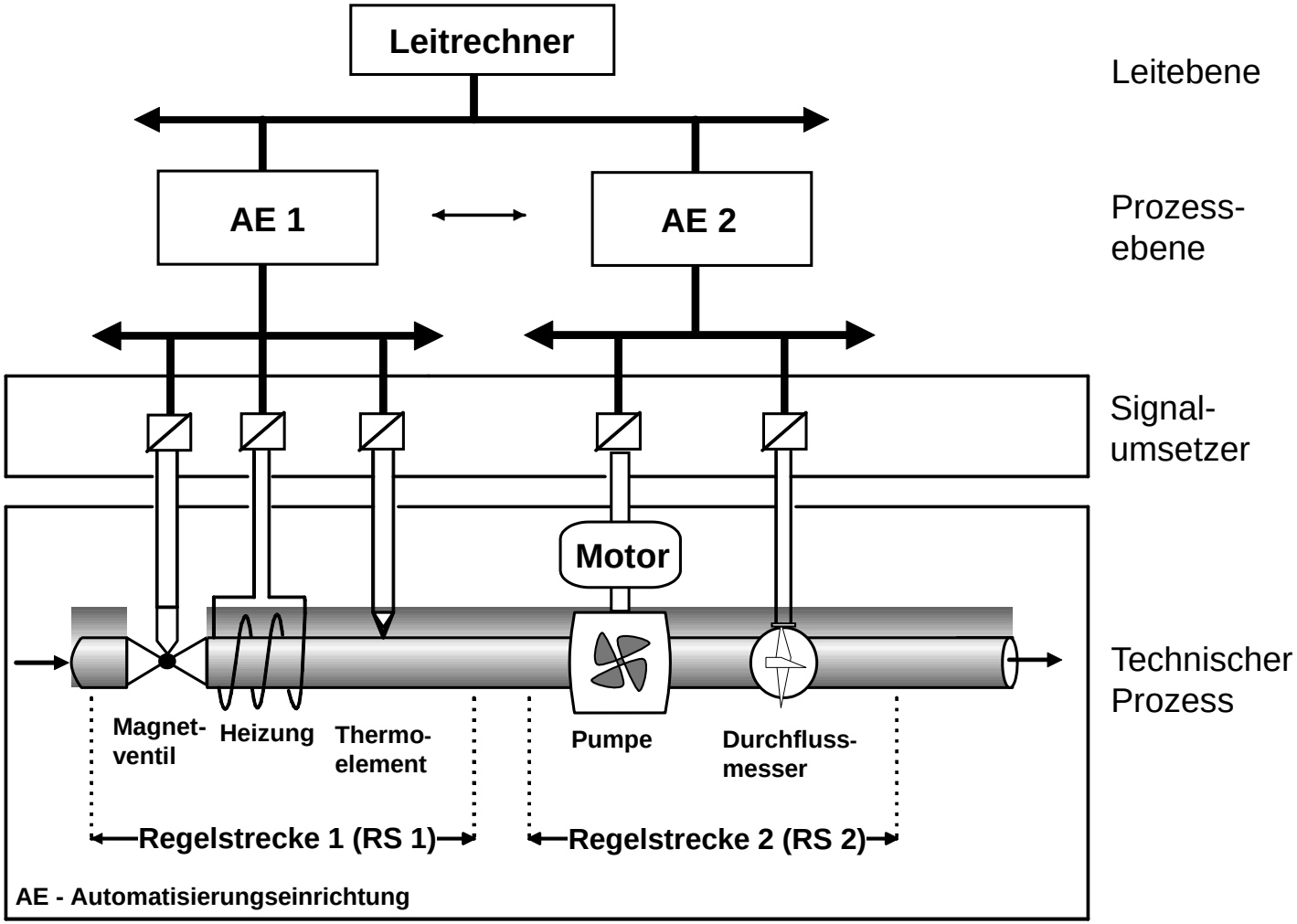


Wirkungskette einer Regelung

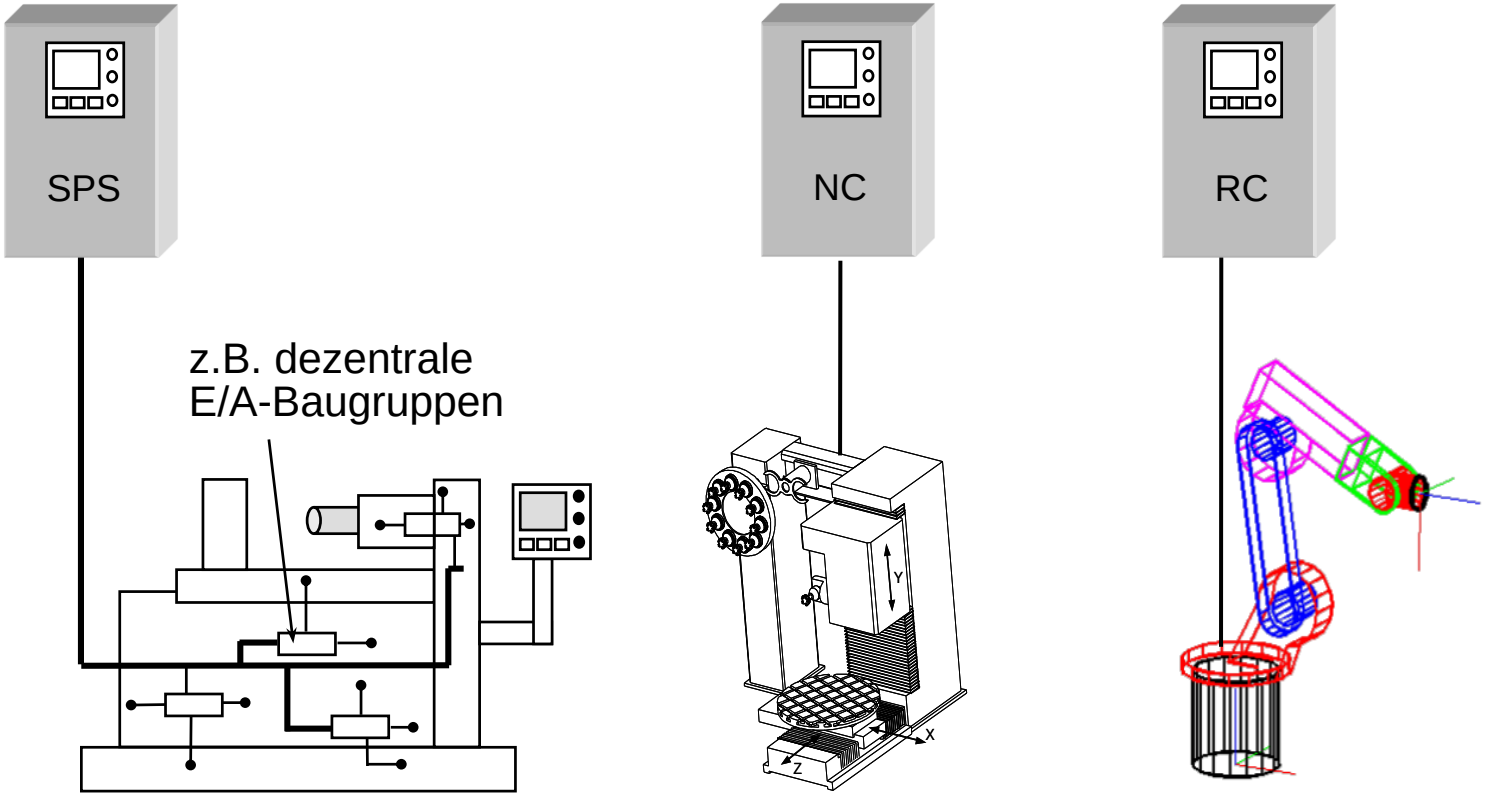
Nach DIN 19226 ist die **Regelung** wie folgt definiert: Die Regelung ist ein technischer Vorgang in einem abgegrenzten System, bei dem eine technische oder physikalische Größe, die sogenannte Regelgröße oder der Istwert, fortlaufend erfasst und durch Vergleich ihres Signals mit dem Signal einer anderen von außen vorgegebenen Größe, der Führungsgröße oder dem Sollwert, im Sinne einer Angleichung an die Führungsgröße beeinflusst wird.



Beispiel: Heiz- und Durchflussregelung



Die Prozesssteuerungen SPS, NC und RC

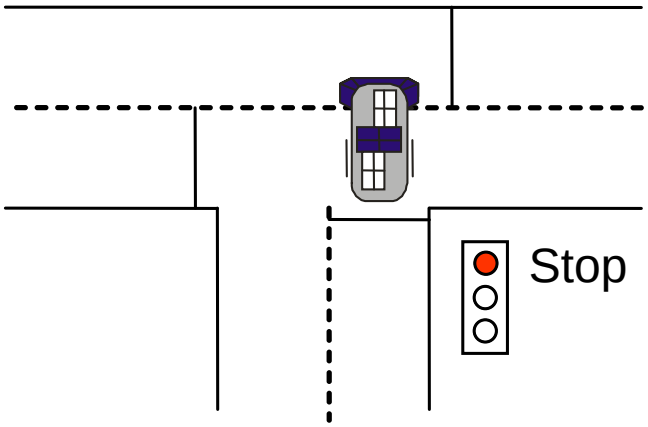


Logische und zeitliche Korrektheit in Echtzeitsystemen

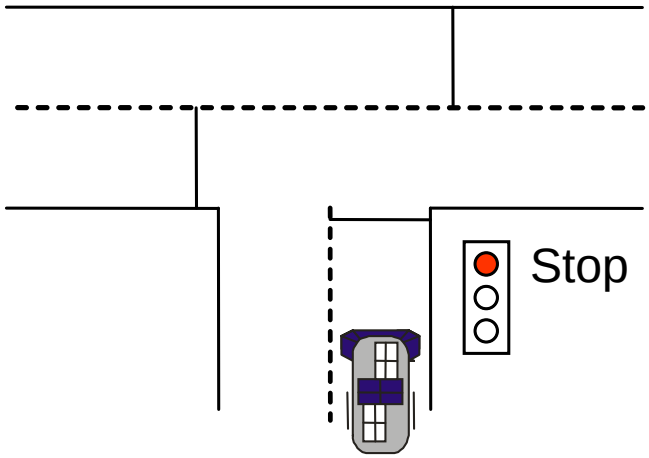
Die Aussage, die sich für alle Echtzeitsysteme verallgemeinern lässt:

Das Ergebnis von Echtzeit-Datenverarbeitung ist nur dann korrekt, wenn es logisch und zeitlich korrekt ist.

Nicht-Echtzeitsysteme: logische Korrektheit→Korrektheit
Echtzeitsysteme: logische + zeitliche Korrektheit →Korrektheit



logisch korrekt, zeitlich inkorrekt

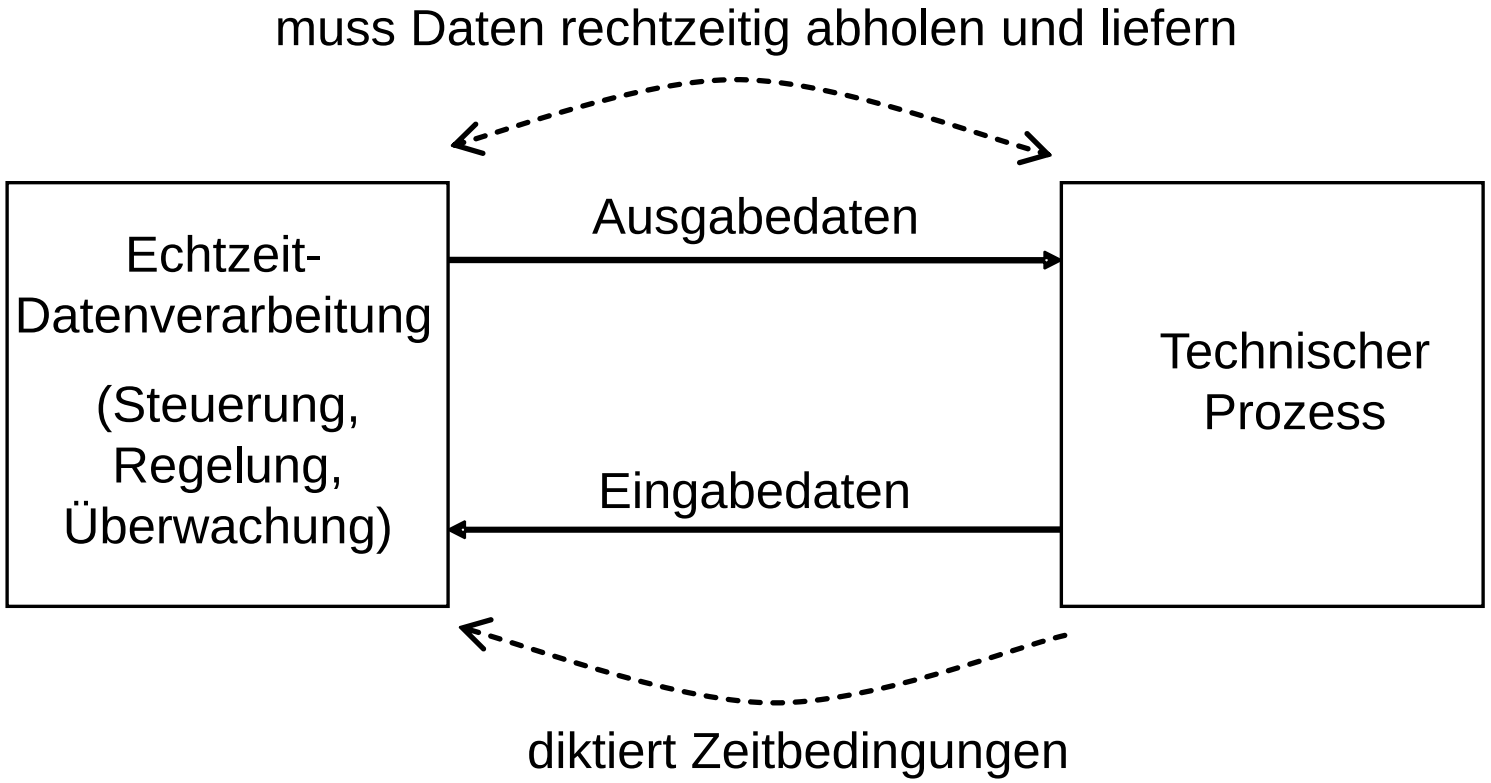


logisch und zeitlich korrekt

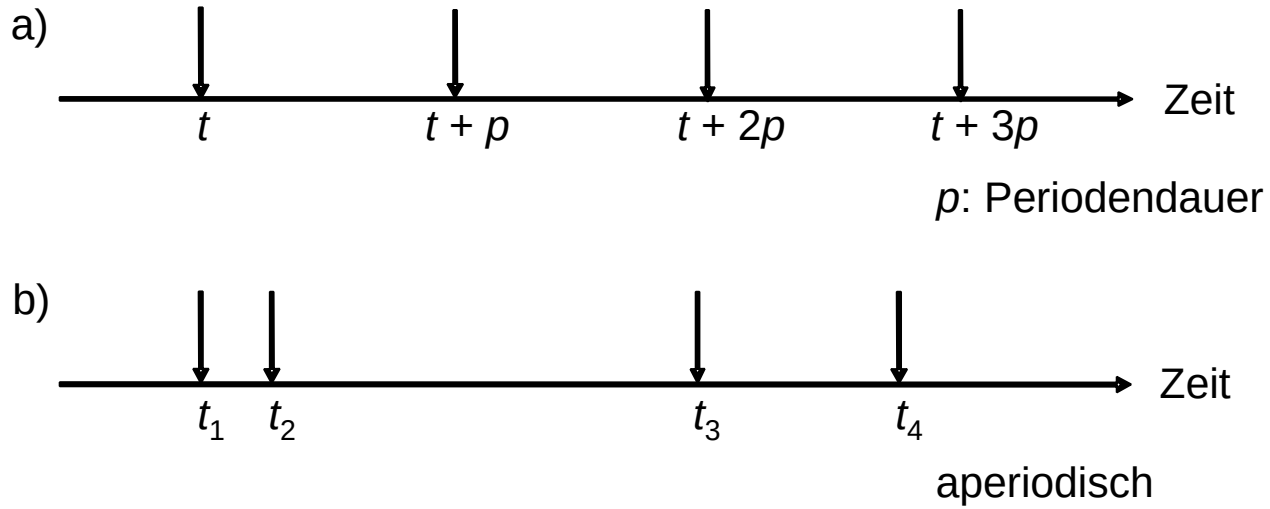
Wichtige Anforderungen an Echtzeitsysteme:
Rechtzeitigkeit, Gleichzeitigkeit, Verfügbarkeit

Zusammenspiel von Echtzeit-Datenverarbeitung und technischem Prozess

Rechtzeitigkeit heißt, die Ausgabedaten müssen rechtzeitig berechnet werden und zur Verfügung stehen.



Rechtzeitigkeit: Periodische und aperiodische Zeitbedingungen



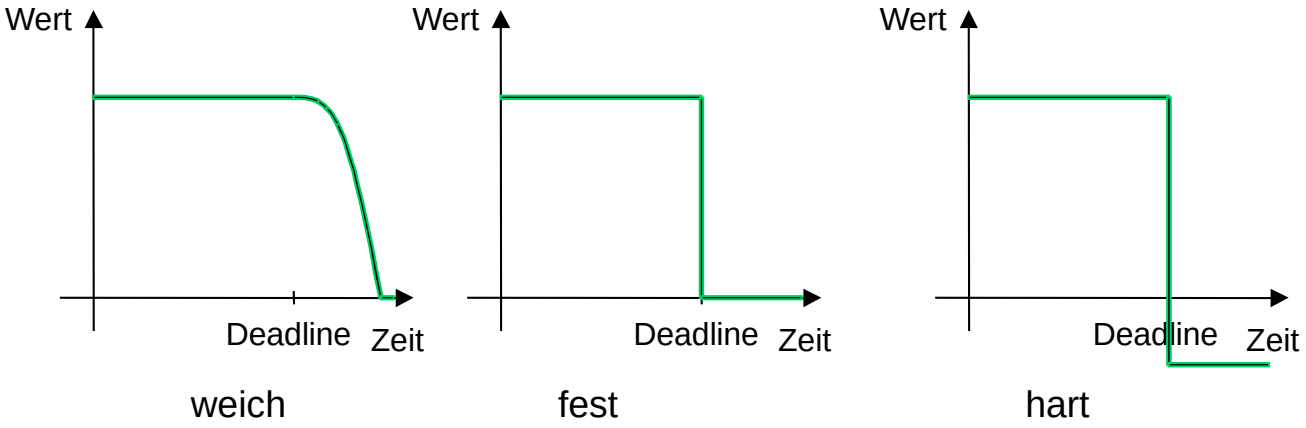
a) periodische Zeitbedingungen: zyklische (periodische) Tasks

b) aperiodische Zeitbedingungen: asynchrone Tasks

Wertfunktionen zur Bewertung von Zeitbedingungen

Je nach Strenge der einzuhaltenden Zeitbedingungen unterscheidet man

- **weiche Echtzeitbedingungen (soft deadline)**
  können in gewissen Grenzen überschritten werden, ohne zu fatalen Systemzuständen zu führen(periodische Abfrage Temperatursensor)
- **feste Echtzeitbedingungen (firm deadline)**
  bewirken bei Überschreiten einen Abbruch der Aktion (Überwachung von Verfall, Flugpositionen)
- **harte Echtzeitbedingungen (hard deadline)**
  müssen auf jeden Fall eingehalten werden, anderenfalls droht Schaden (Anhalten vor Ampel, Notaus)
- Die Bewertung der Überschreitung von Zeitbedingungen erfolgt i.A. durch eine **Wertfunktion**:

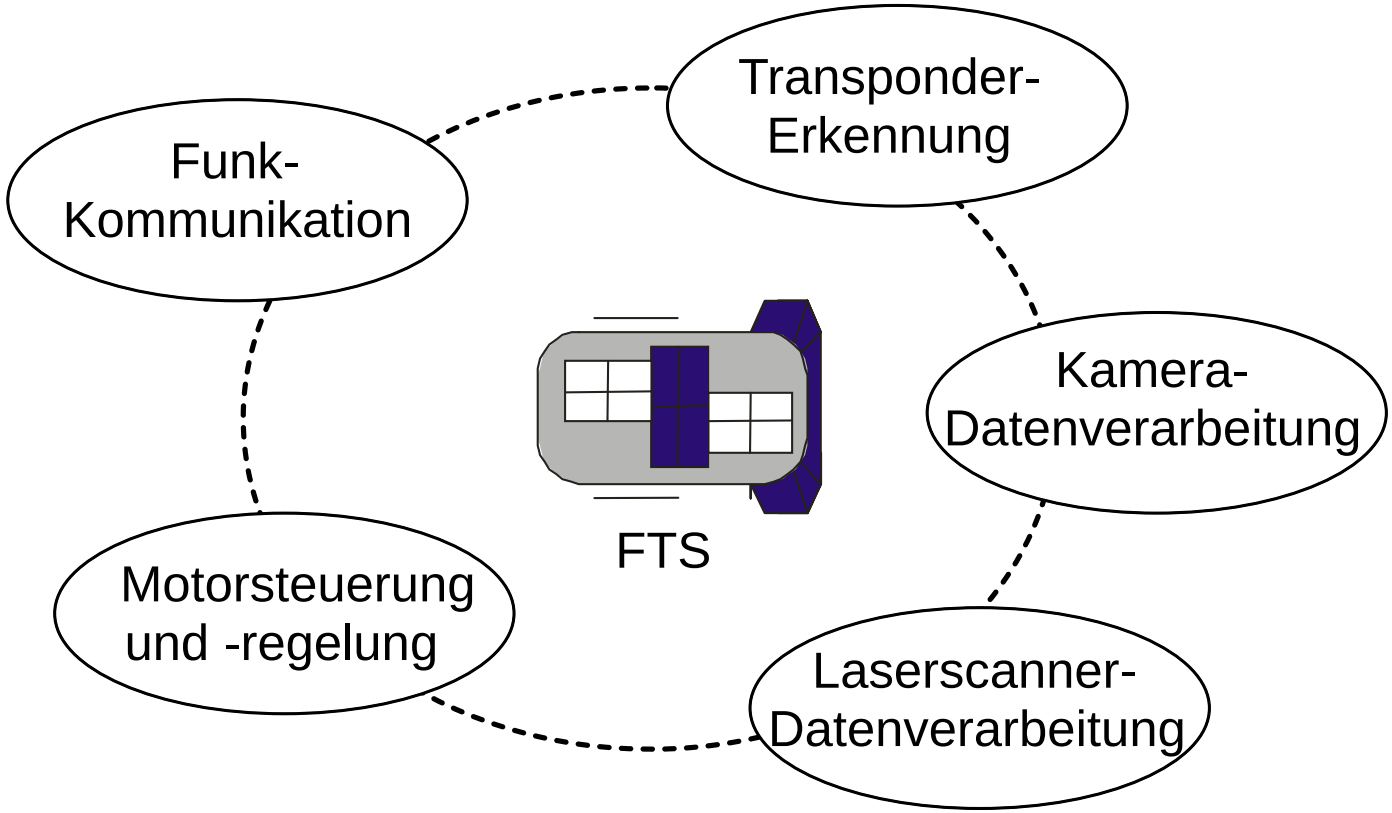


weich fest hart

Wert > 0: ausführen,
Wert = 0: abbrechen,
Wert < 0: Schaden

Gleichzeitig durchzuführende Aufgaben bei einem fahrerlosen Transportsystem

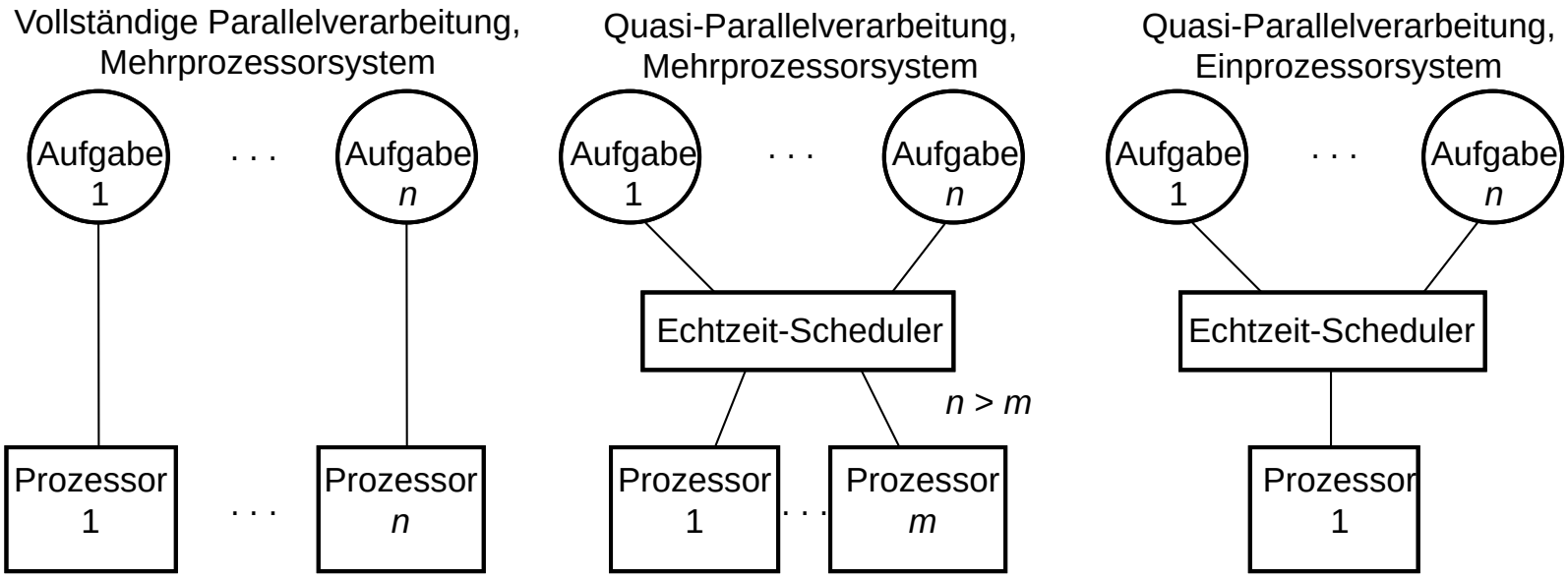
Die ***Gleichzeitigkeit*** bedeutet, dass die Rechtzeitigkeit für mehrere Aktionen gleichzeitig gewährleistet sein muss .



Realisierungsformen der Gleichzeitigkeit

Möglichkeiten zur Erfüllung der **Gleichzeitigkeit**:

- Vollständige Parallelverarbeitung in einem Mehrprozessorsystem
- Quasi-parallele Verarbeitung in einem Mehrprozessorsystem
- Quasi-parallele Verarbeitung in einem Einprozessorsystem

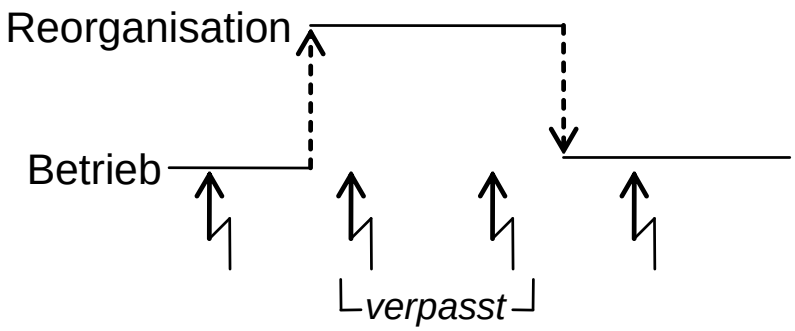


Verfügbarkeit bei verschiedenen Reorganisations-techniken (z.B. zur Speicherbereinigung in Java)

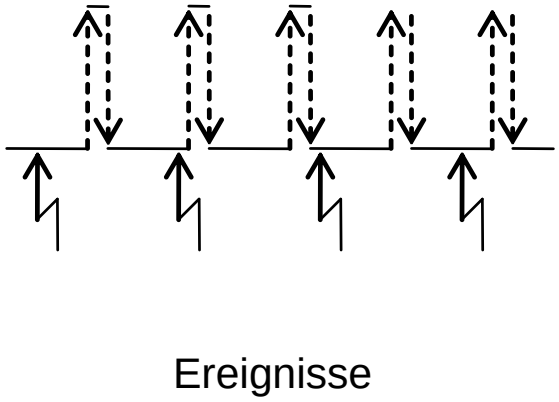
Die ***Verfügbarkeit***: Echtzeitsysteme müssen über einen längeren Zeitraum hinweg verfügbar sein, einige sogar rund um die Uhr für 24 Stunden. Verfügbarkeit erfordert keine Unterbrechungen des Betriebs für Reorganisationsphasen.

Problematisch:
Reorganisation von DB, Speicherbereinigung von z.B. Java

a) Speicherreorganisation im Block



b) Speicherreorganisation in kleinen Schritten



Programmierverfahren

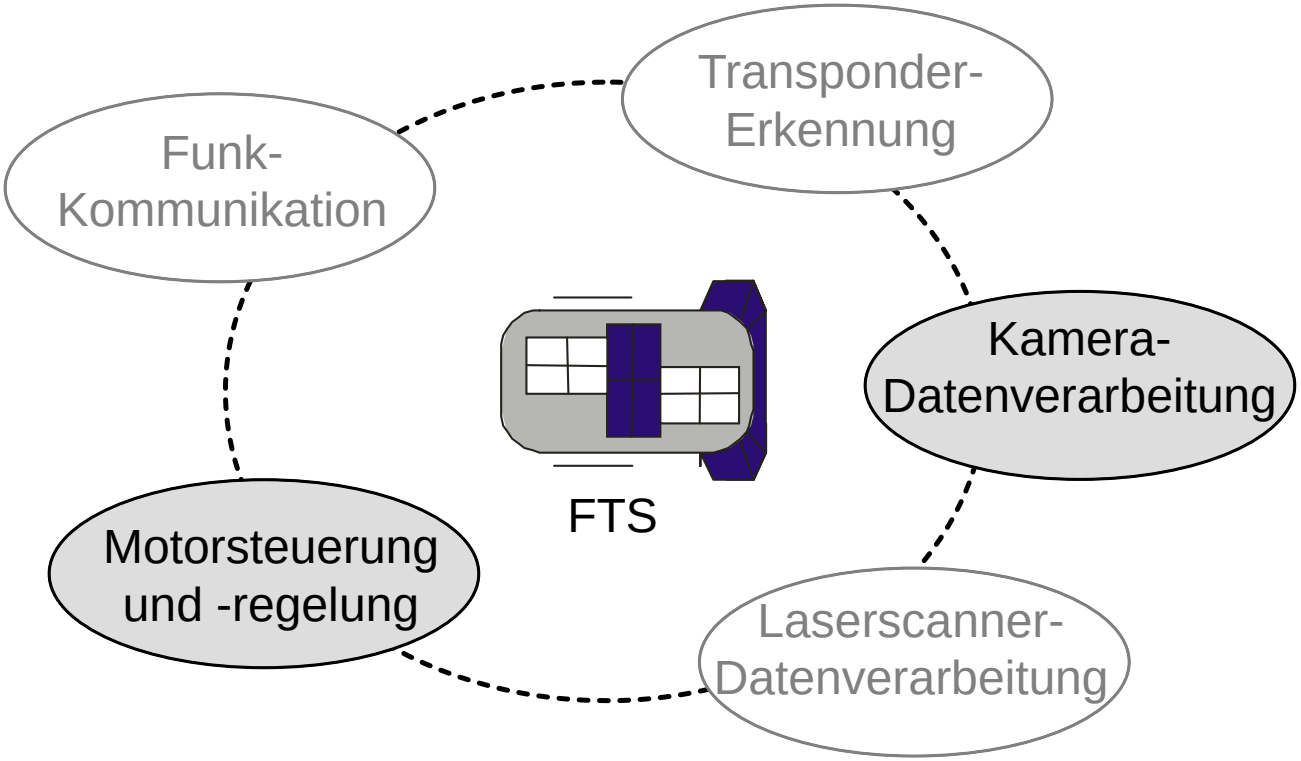
Es gibt zwei grundlegende Programmierverfahren:

- Die **synchrone Programmierung** zur Konstruktion zeitgesteuerter Systeme.
- Die **asynchrone Programmierung** zur Konstruktion von ereignisgesteuerten Systemen.

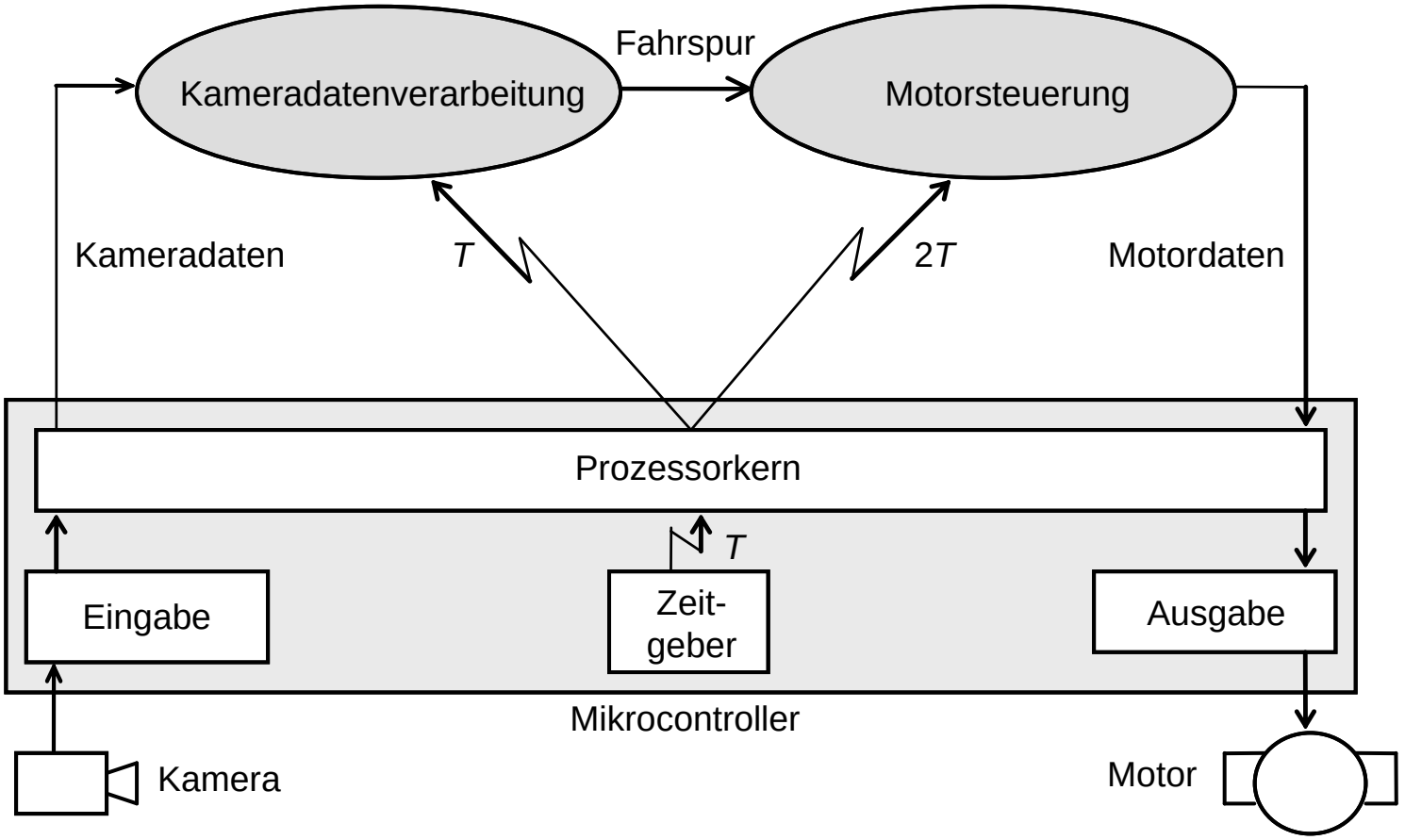
Synchrone Programmierung

- Das zeitliche Verhalten periodischer Aktionen wird vor deren Ausführung geplant
- Die periodischen Aktionen werden in ein Zeitraster T synchronisiert (daher Name des Verfahrens)
- Aktionen werden immer zu ganzzahligen Vielfachen von T ausgeführt
- Das Zeitraster wird durch einen Zeitgeber gewonnen
- Die Reihenfolge des Ablaufs der Teilprogramme wird fest vorgegeben

Vereinfachtes FTS Beispiel mit zwei periodischen Aufgaben



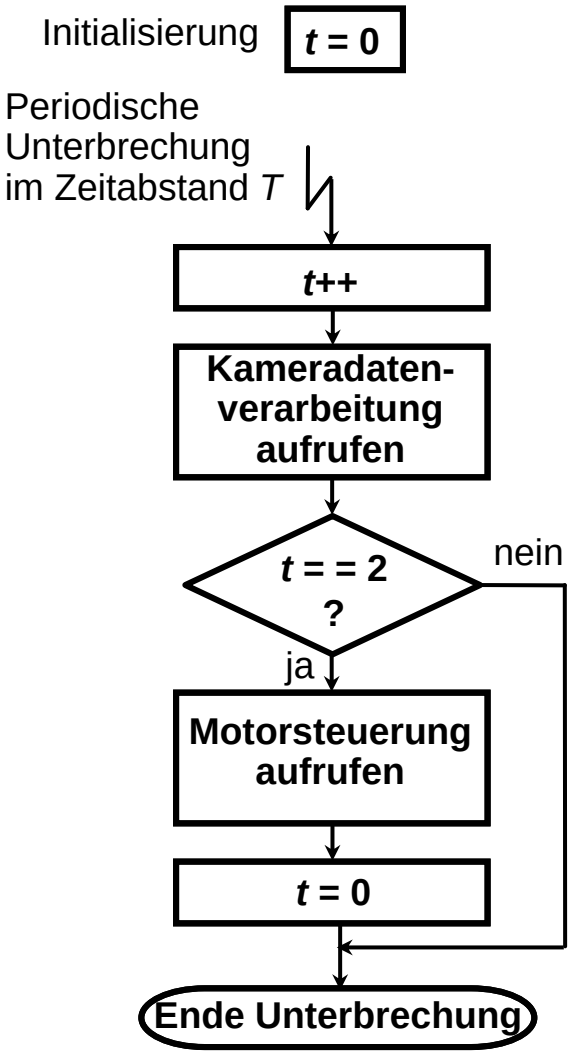
Aufbau der FTS Steuerung mit einem Mikrocontroller



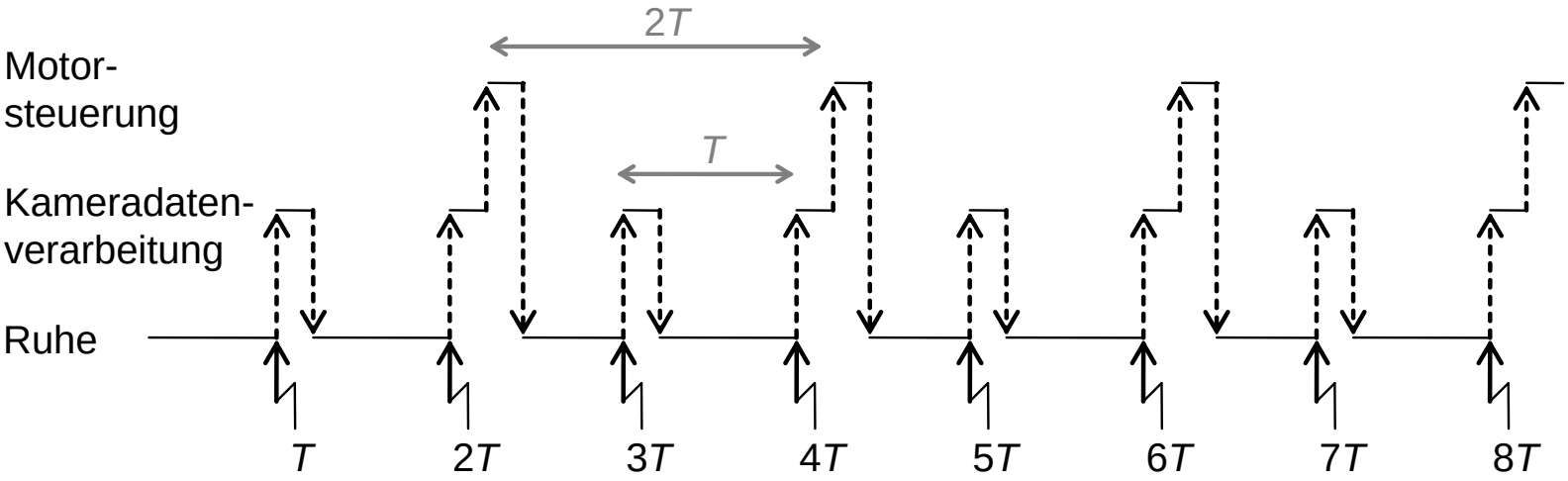
Ablaufplanung und deren Realisierung für die einfache FTS-Steuerung

Ablaufplanung

Aufgabe	Periode
Kameradaten-verarbeitung	*T*
Motor-steuerung	2*T*



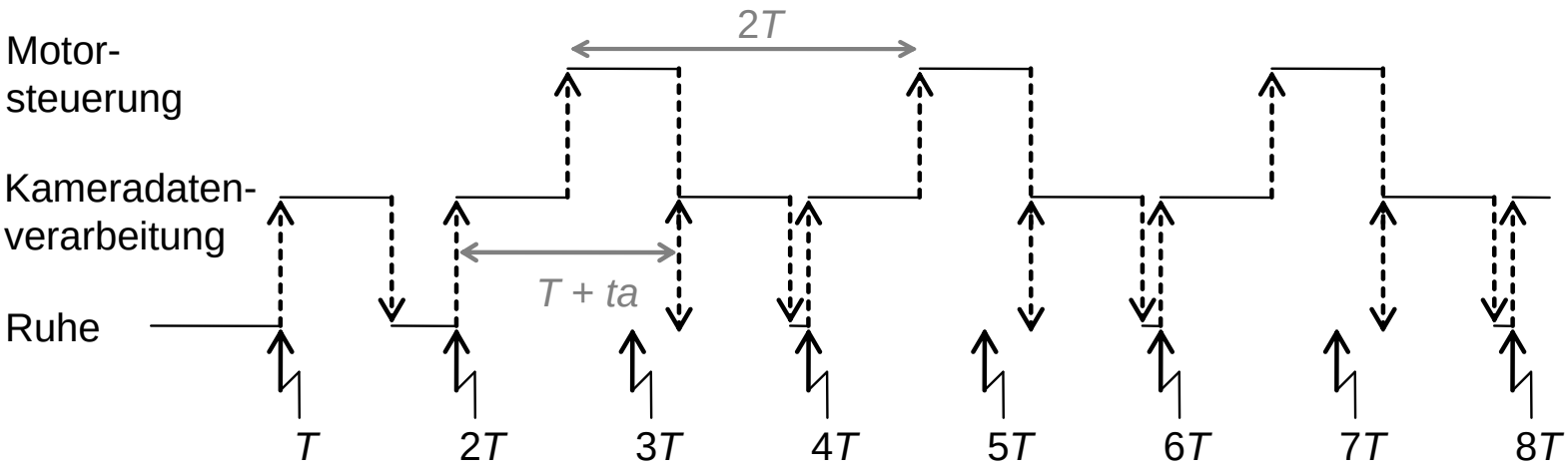
Zeitlicher Ablauf der einfachen FTS-Steuerung



Exakte Einhaltung der Periodenzeiten unter zwei Bedingungen:

- **Die Summe der Ausführungszeiten** der einzelnen Aufgaben ist **kleiner *T*.**
- Jede längere Periodenzeit ist ein ganzzahliges Vielfaches der nächst kürzeren Periodenzeit.

Periodenabweichung bei Summe der Ausführungszeiten größer *T*

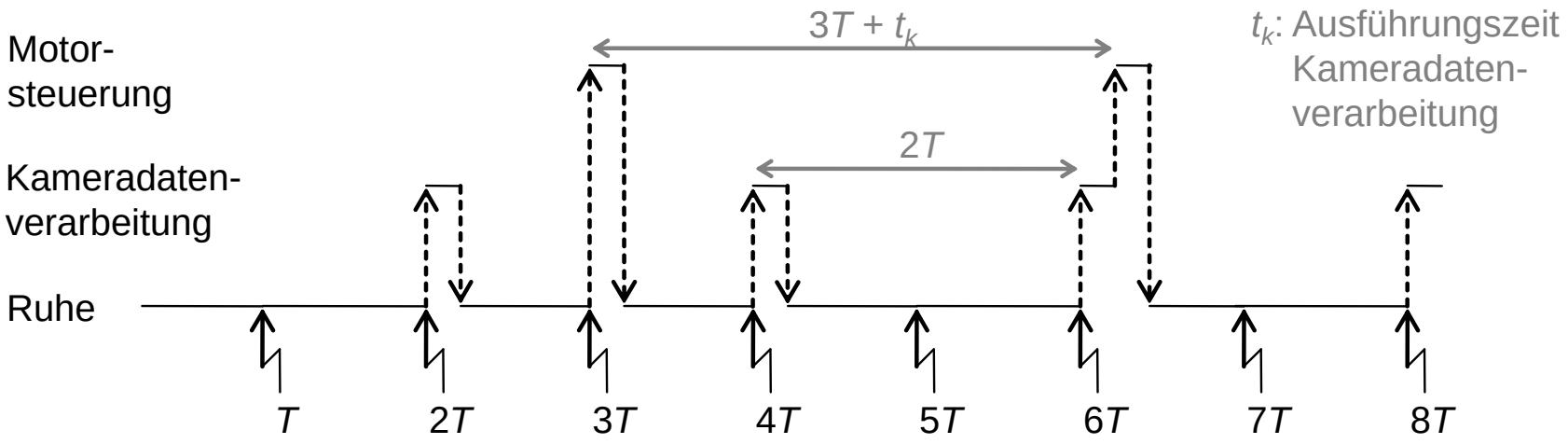


ta: Ausführungszeit Kameradatenverarbeitung + Ausführungszeit Motorsteuerung –*T*

Periodenabweichung der Task Kameradatenverarbeitung: *T* + *ta*

Periodenabweichung bei Verletzung ´Periodenzeit ist ganz. Vielfaches der nächst kürzeren Periodenzeit.´

Angestrebte Periodendauer: Kameradatenverarbeitung: $2T$
Motorsteuerung: $3T$



Abweichung der Periodendauer = Summe der Laufzeiten der vorher ablaufenden Tasks

Vorteile und Nachteile der synchronen Programmierung

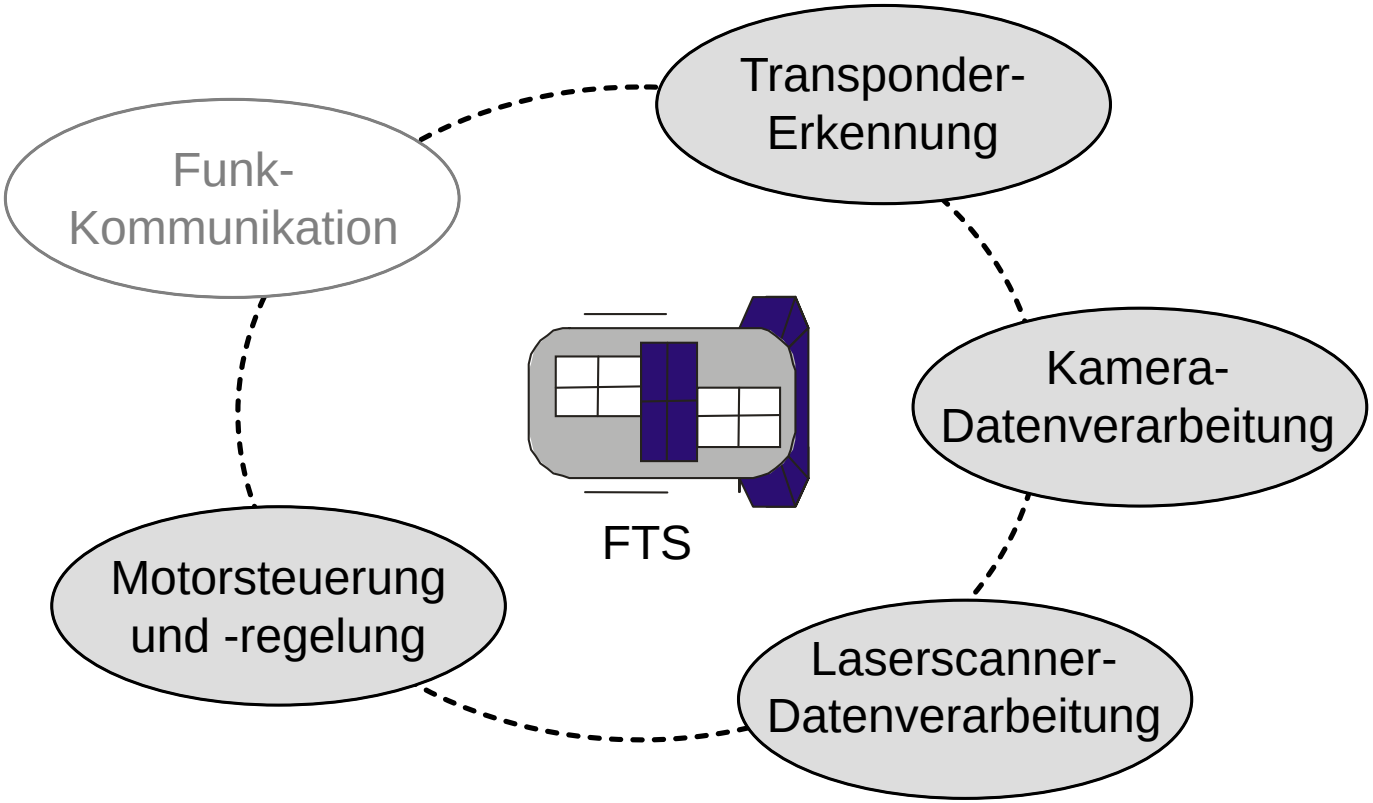
Die **Vorteile** der synchronen Programmierung sind offensichtlich:

- Sie besitzt ein festes, vorhersagbares Zeitverhalten.
- Sie erlaubt eine einfache Analyse und einen einfachen Test des Systems.
- Sie ist hervorragend bei zyklischen Abläufen anwendbar.
- Bei richtiger Planung können Rechtzeitigkeit und Gleichzeitigkeit garantiert werden.
- Dadurch ist die synchrone Programmierung insbesondere für zyklische und sicherheitsrelevante Aufgaben geeignet.

Daneben sind aber auch zwei wesentliche **Nachteile** zu nennen:

- Geringe Flexibilität gegenüber Änderungen der Aufgabestellung.
- Eine Reaktion auf aperiodische Ereignisse ist nicht vorgesehen.

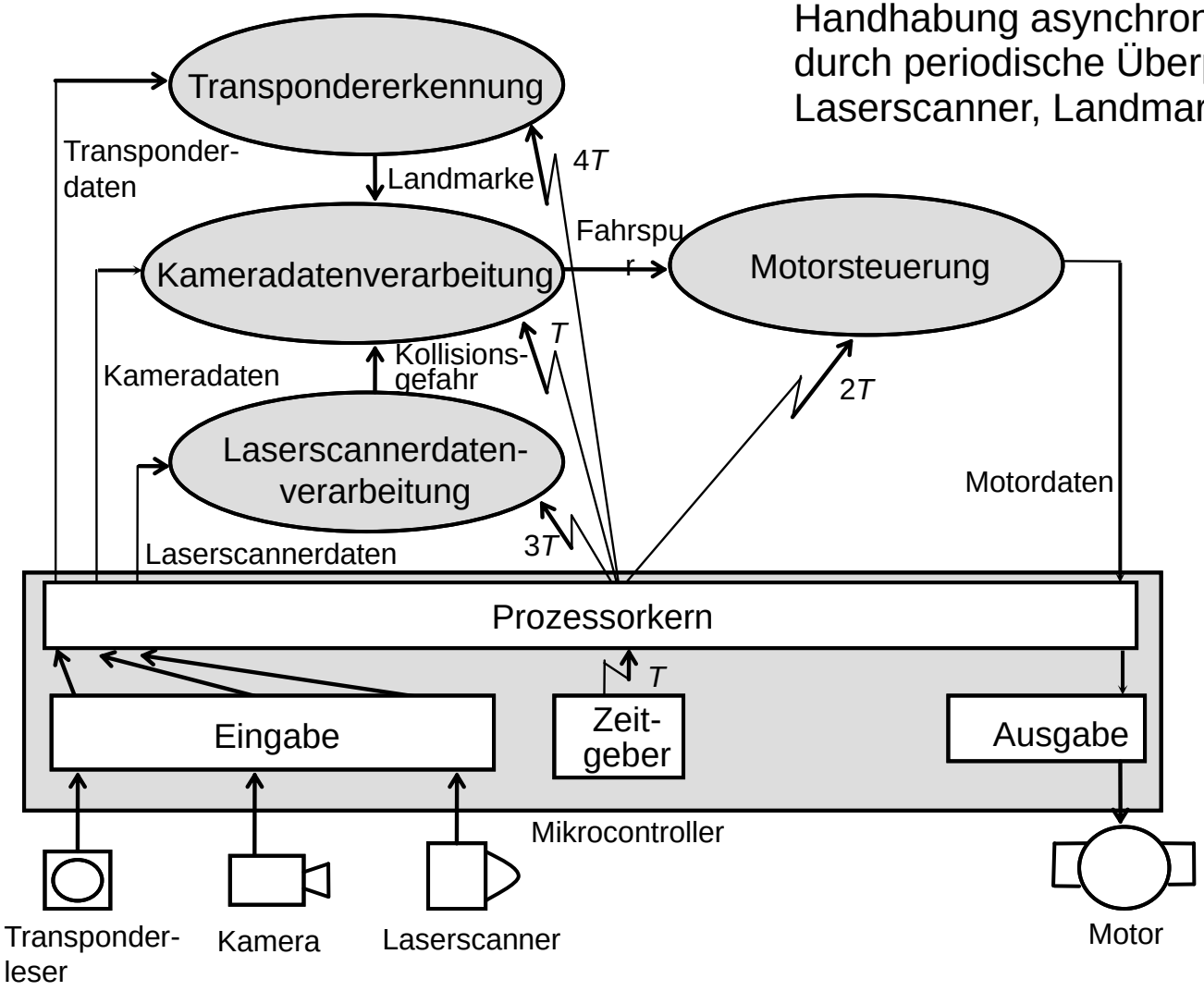
Komplexeres FTS Beispiel mit vier Aufgaben



Aperiodische Ereignisse (Transponder, Laserscanner) bei der synchronen Programmierung nur mittels periodischer Überprüfung möglich

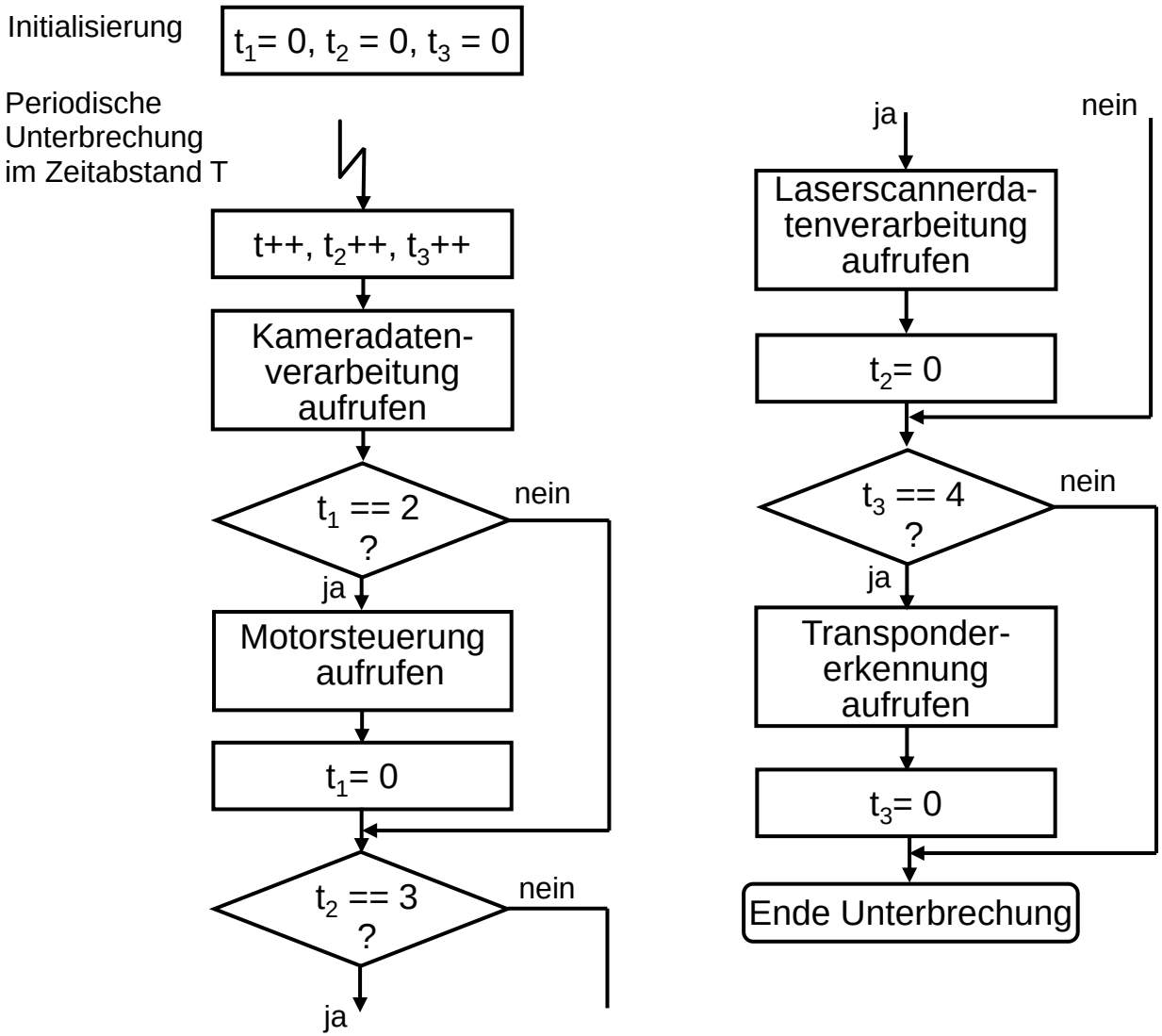
Aufbau der komplexeren FTS Steuerung mit einem Mikrocontroller

Handhabung asynchroner Ereignisse durch periodische Überprüfung
Laserscanner, Landmarke

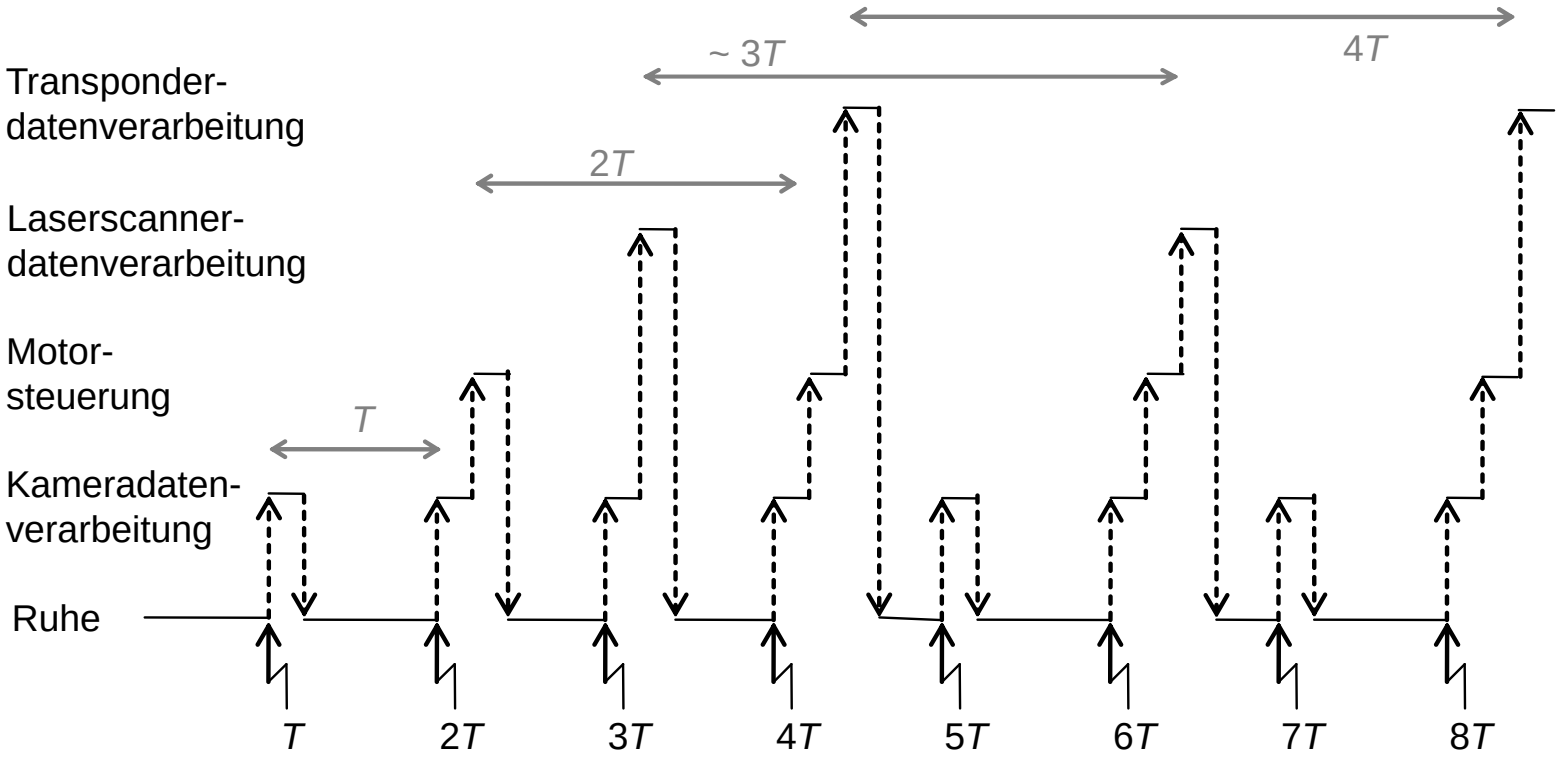


Ablaufplanung und deren Realisierung für eine komplexere FTS-Steuerung

Ablaufplanung	
Aufgabe	**Periode**
Kameradaten-verarbeitung	T
Motorsteuerung	2T
Laserscannerdaten-verarbeitung	3T
Transponder-erkennung	4T



Zeitlicher Ablauf der komplexeren FTS-Steuerung



Zykluszeiten für Laserscanner und für Transponder werden näherungsweise eingehalten

Asynchrone Programmierung

Ablaufplanung der Aktionen zur Laufzeit durch ein Organisationsprogramm (Echtzeitbetriebssystem)

Aufgaben des ***Organisationsprogramm***:

- Ereignisse überwachen
- Informationen über einzuhaltende Zeitbedingungen entgegennehmen
- Ablaufreihenfolge von Teilprogrammen zur Bearbeitung der eingetretenen Ereignisse ermitteln
- Teilprogramme gemäß der ermittelten Ablaufreihenfolge aktivieren

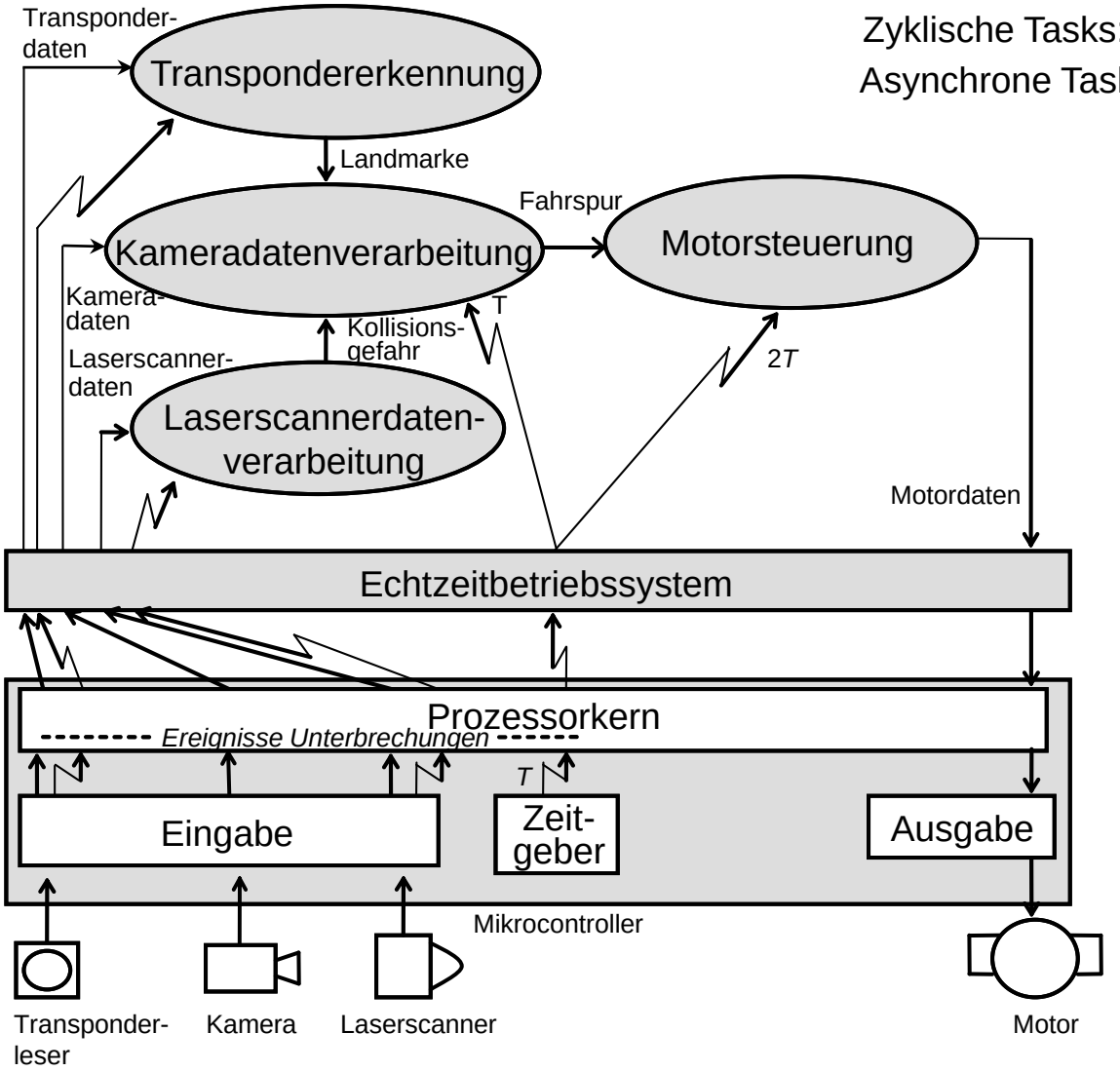
Dies nennt man **Echtzeitscheduling**

Ein einfaches Echtzeitscheduling-Verfahren

Fixed Priority Preemptive Scheduling besitzt folgende Eigenschaften:

- Jedes Ereignis erhält eine feste Priorität (*Fixed Priority*)
- Die Reihenfolge der Ausführung richtet sich nach dieser Priorität, Ereignisse hoher Priorität werden vor Ereignissen niederer Priorität behandelt.
- Ereignisse höherer Priorität unterbrechen Ereignisse niederer Priorität (*Preemption*).
- Die Bearbeitung eines Ereignisses niederer Priorität wird erst wieder aufgenommen, wenn alle Ereignisse höherer Priorität abgearbeitet oder inaktiv geworden sind.

FTS Steuerung mittels Mikrocontroller und asynchroner Programmierung



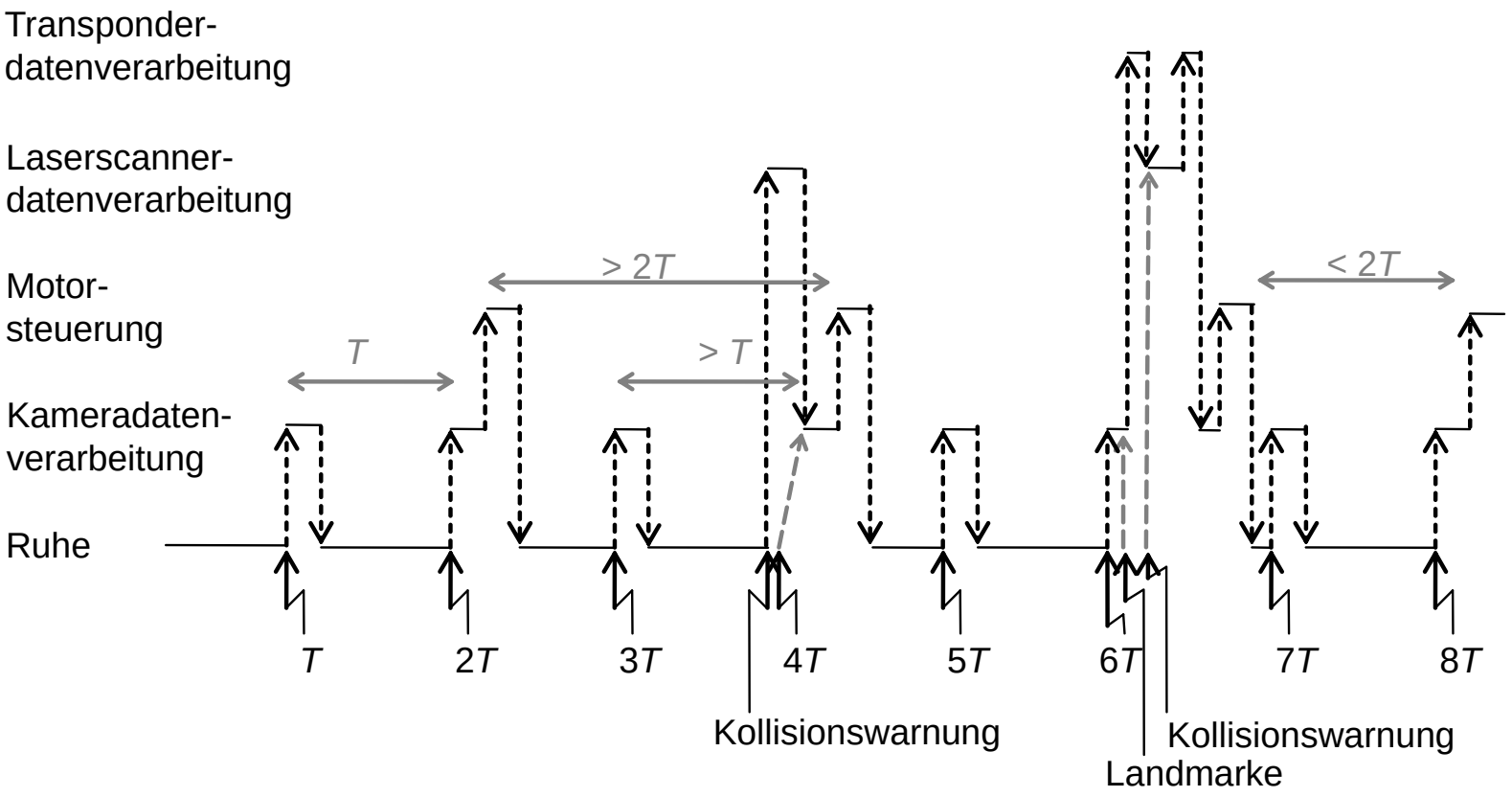
Zyklische Tasks: Kameradatenv., Motorst.

Asynchrone Tasks: Lasersc., Transponder

Prioritätenzuordnung für das FTS bei asynchroner Programmierung

Prioritätenzuordnung			
Aufgabe	**Ereignis**	**Periode**	**Priorität**
Laserscanner-datenverarbeitung	Kollisions-warnung	-	1 (höchste)
Transponder-erkennung	Landmarke erkannt	-	2
Kameradaten-verarbeitung	Zeitgeber	T	3
Motor-steuerung	Zeitgeber	2T	4 (niedrigste)

Zeitlicher Ablauf der FTS-Steuerung bei asynchroner Programmierung

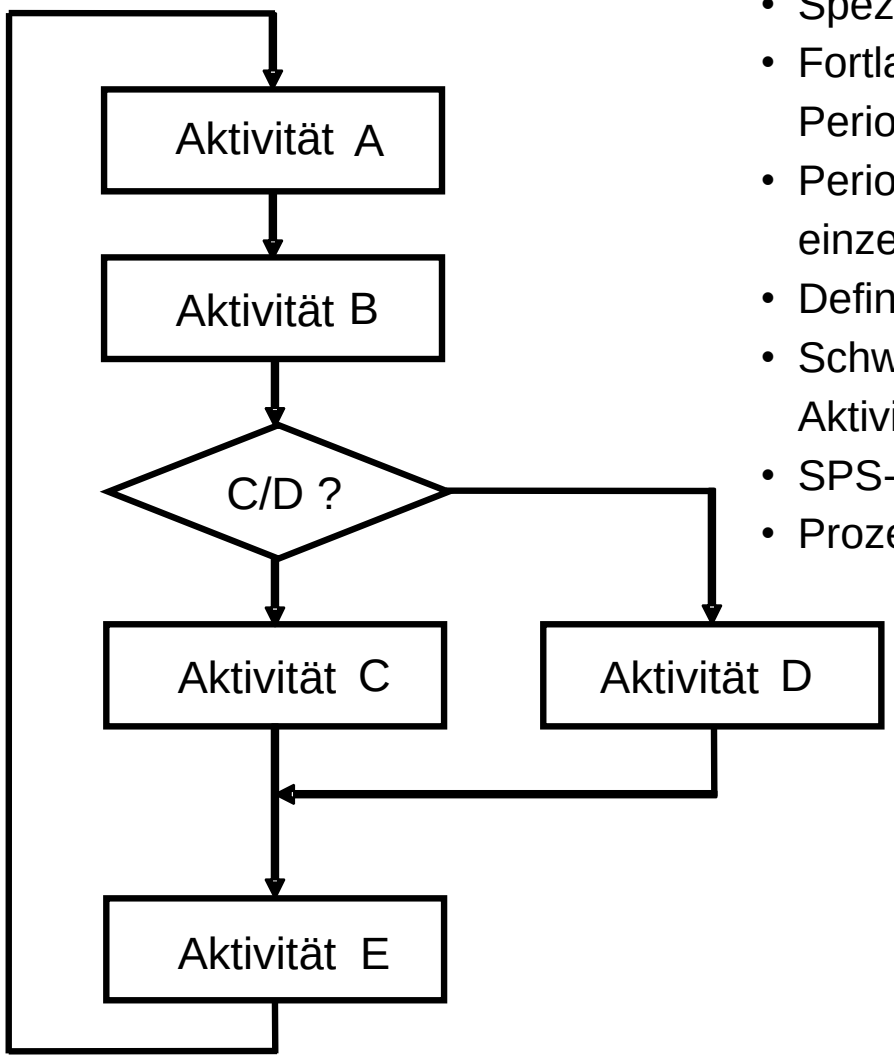


Veränderung der Perioden für niederpriorere Tasks durch das dynamische Einschieben von hochprioreren Ereignissen (Tasks)

Charakterisierung der asynchronen Programmierung

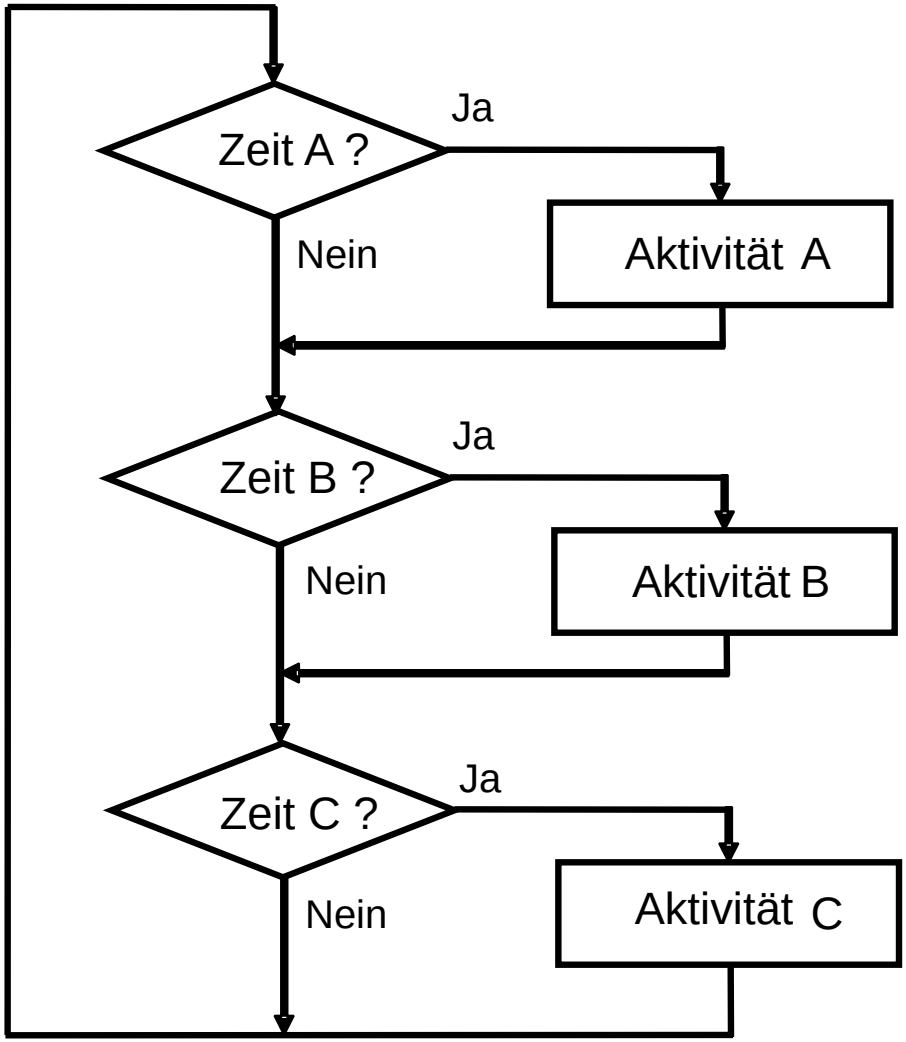
- Flexibler Programmablauf und flexible Programmstruktur
- Reaktion auf periodische und aperiodische Ereignisse
- Rechtzeitigkeit nicht in jedem Fall im Voraus garantierbar
- Je niederer die Priorität einer Aktivität, desto größer die möglichen Zeitschwankungen
- Systemanalyse und Systemtest daher schwieriger als bei der synchronen Programmierung
- Je nach verwendeter Echtzeitschedulingstrategie ist jedoch bei Kenntnis bestimmter Parameter (kleinste Periodendauer, größte Ausführungszeit der Aktivitäten) eine Vorabanalyse zur Rechtzeitigkeit möglich (*Feasibility Analysis*)

Zyklische Ablaufsteuerung



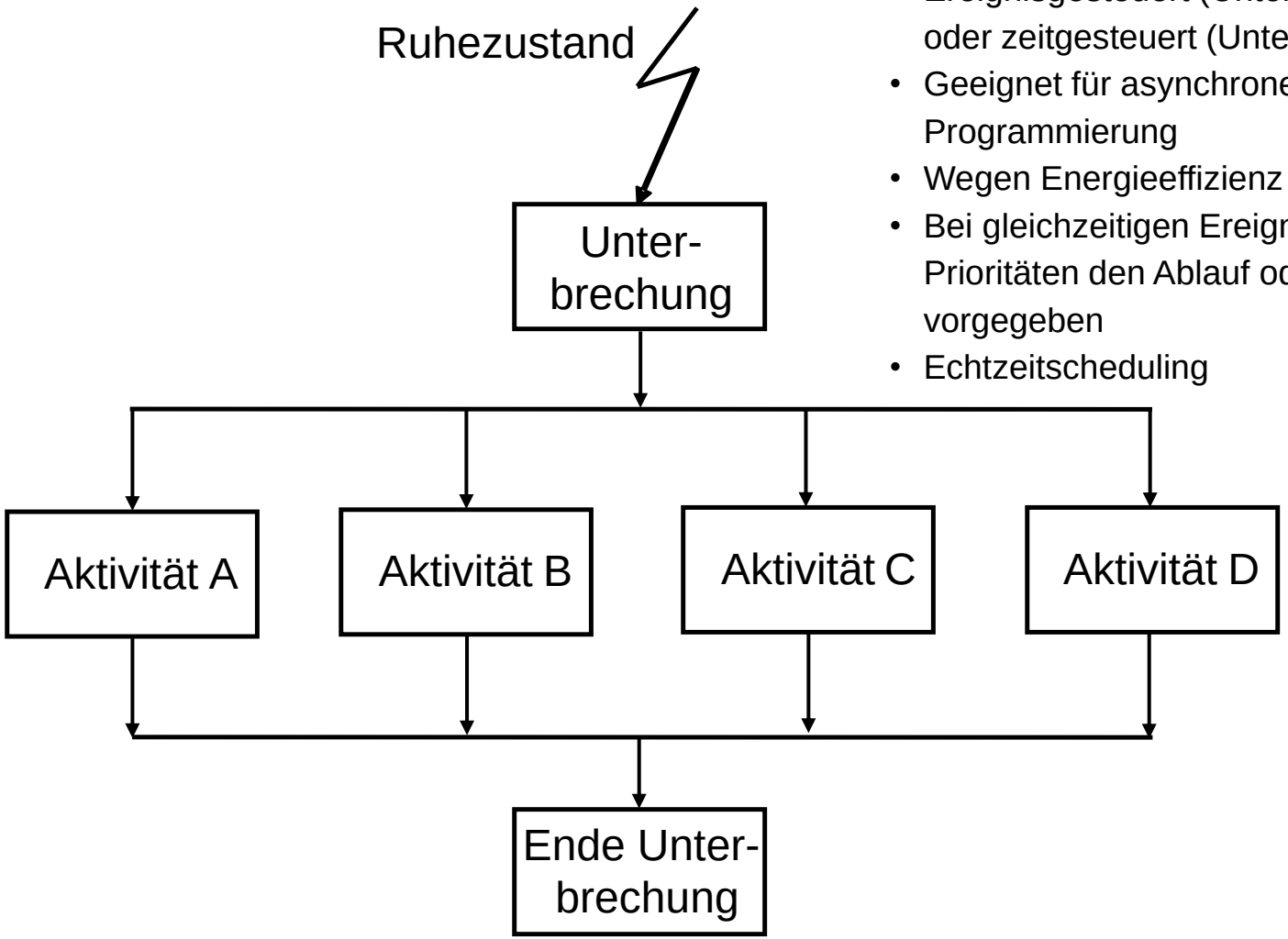
- Spezielle Form der synchronen Programmierung
- Fortlaufende Schleife – schnellstmögliche Periodendauer
- Periodendauer = Summe der Ausführungszeiten einzelner Aktivitäten.
- Definierte Periodendauer durch Warteschleifen
- Schwankende Periodendauer durch alternative Aktivitäten
- SPS-Ablauf
- Prozessor ständig aktiv

Zeitgesteuerte Ablaufsteuerung



- Für synchrone Programmierung gut geeignet
- Fortlaufende Schleife
- Periodendauer ist bestimmt durch die Abfragezeiten A, B und C und ggf. durch die Ausführungszeiten der Aktivitäten
- Konstante Periodendauer, falls folgende Bedingungen erfüllt sind:
  - Summe aller Ausführungszeiten < T
  - Längere Periodenzeit = ganzzahl. Vielfaches der kürzere Periodenzeit
- Prozessor ständig aktiv

Unterbrechungsgesteuerte Ablaufsteuerung



- Prozessor nicht ständig aktiv
- Ereignisgesteuert (Unterbrechung durch Ereignis) oder zeitgesteuert (Unterbrechung durch Zeitgeber)
- Geeignet für asynchrone und synchrone Programmierung
- Wegen Energieeffizienz bevorzugte Lösung
- Bei gleichzeitigen Ereignissen bestimmen die Prioritäten den Ablauf oder es wird ein fester Ablauf vorgegeben
- Echtzeitscheduling