

Vorlesungsskript zu **„Vertiefung Programmieren“** **Zeitbasis und Scheduler**



Dozent: Dipl.-Inf. (FH) Andreas Schmidt

Zeitbasis

Zeitbasis in Embedded Software

Zentraler „Takt“

- In Embedded-Systemen spielt die Zeit häufig eine wichtige Rolle. Es müssen bestimmte Aktionen in definierten zeitlichen Abständen durchgeführt werden oder es müssen z.B. Ereignisse zeitlich überwacht werden.
- Eine solche Zeitmessung wird oft mit Hilfe eines zentralen Zeittakts sowie eines Zählers realisiert (ähnlich einer „Stoppuhr“)
- Der STM32G474 Controller hat hierzu einen sog. SysTick Timer
 - Diese Peripherie erzeugt in einem definierten Zeitabstand (meist 1ms) einen Interrupt.
 - Die Interrupt-Routine incrementiert bei jedem SysTick Timer Interrupt eine globale Zählervariable
 - Somit ist der Wert der Zählervariable * 1ms = Zeit in Millisekunden seit dem Initialisieren des SysTick Timers (was häufig gleichgesetzt wird mit dem Systemstart)

Zentraler „Takt“

SysTick Takt (HW)



SysTick Interrupt (SW)



Increment
Tick Counter

1

2

3

4



$4 * 1\text{ms} = 4\text{ms seit „Systemstart“}$

Scheduler

Kooperatives Scheduling

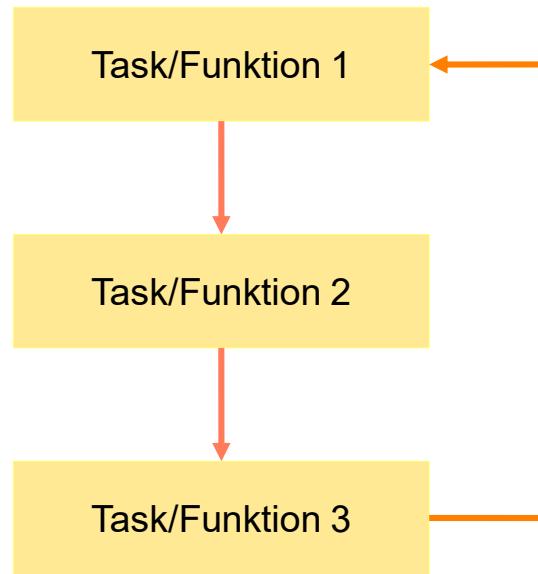
Kooperative Scheduling

- Beim sog. kooperativen Scheduling in Embedded-Systemen ist die Reihenfolge der Task/Funktionsaufrufe festgelegt, aber eine aufgerufene Task/Funktion muss selbstständig zurückkehren damit die nächste Task/Funktion aufgerufen werden kann.

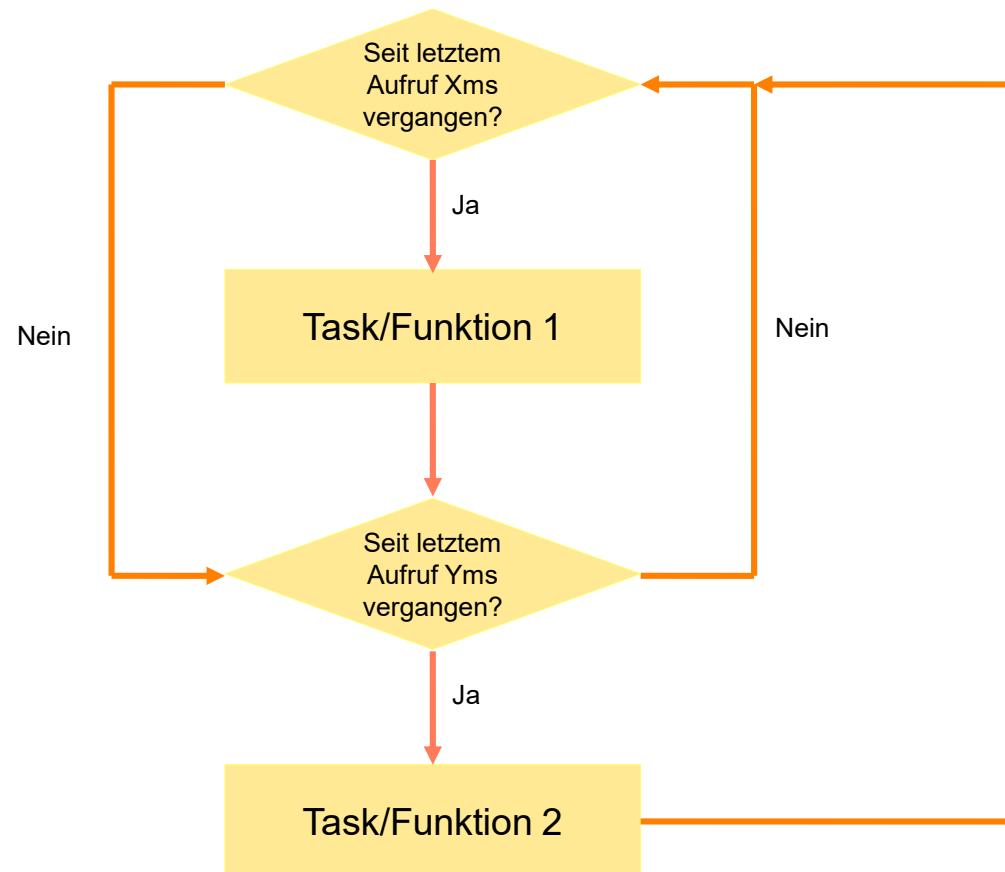
Die Funktion muss sich **kooperativ** verhalten

- Kehrt eine Funktion nicht zurück, können alle weiteren Tasks/Funktionen nicht ausgeführt werden
- Benötigt eine Task/Funktion länger in der Ausführung, so kann eine zeitlich korrekte Abfolge nicht mehr garantiert werden

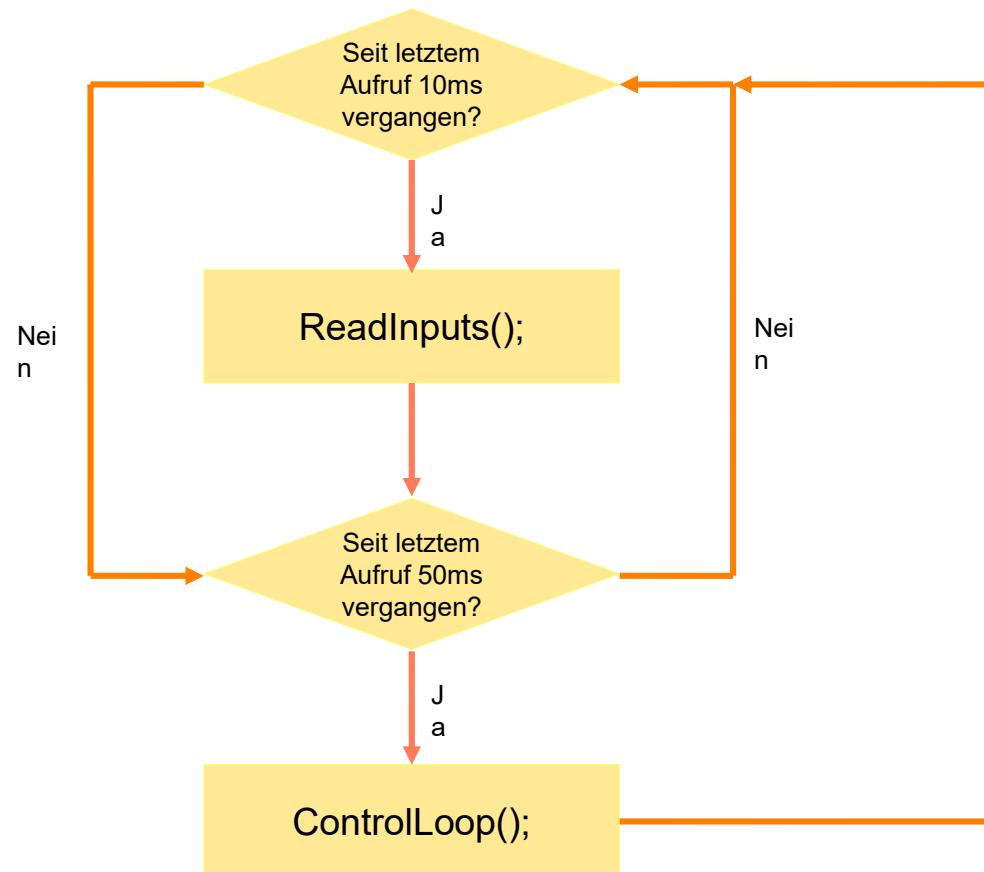
Kooperative Scheduling



Kooperative Scheduling (zeitbezug)



Kooperative Scheduling (zeitbezug)



Präemptives Scheduling

- Beim sog. präemptiven Scheduling sorgt der sog. Scheduler dafür, dass ein Task-Wechsel nach bestimmten Kriterien durchgeführt wird, unabhängig davon, ob die Task/Funktion beendet bzw. zurückgekehrt ist
- Durch dieses „unterbrechen“ (preemption) der Task, erhalten alle Tasks im System nach bestimmten Regeln entsprechende Rechenzeit der CPU
- Dieses Scheduling wird häufig bei Betriebssystem eingesetzt
 - Hierbei wird der Scheduler in der Regel in einem festen Zeitraster (z.B. SysTick Timer) aufgerufen und entsprechend der Scheduling Regeln wird bestimmt, welche Task als nächstes ausgeführt wird
 - Dabei wird ein sog. Context-Switch durchgeführt, welche der aktuell laufenden Tasks den „Prozessor entzieht“ und die neu ausgewählte Task den „Prozessor erhält“