


<p style="text-align: center;"><b>Ostfalia</b> Hochschule für angewandte Wissenschaften</p>  <p>Fakultät Fahrzeugtechnik Prof. Dr.-Ing. V. von Holt Institut für Fahrzeugsystem- und Servicetechnologien</p>	<p>Modulprüfung Embedded Systems BPO 2011</p> <p style="text-align: center;">WS 2020/2021 22.02.2021</p>	<p>Name:.....</p> <p>Vorname.....</p> <p>Matr.Nr.:.....</p> <p>Unterschrift.....</p>
---	--	--

Zugelassene Hilfsmittel: **Einfacher Taschenrechner**  
Zeit: 60 Minuten

---

1 (10)	2 (30)	3 (20)	Summe (60)	Note

### Aufgabe 1 (10 Punkte) – Kurzfragen

- a) (2 P) Was unterscheidet das Versetzen einer Task in den **Wait-Zustand** von dem Aufruf einer „klassischen“ **Delay-/Wait-Funktion**?
  
  
  
  
  
  
  
  
  
  
- b) (2 P) Wann bezeichnet man ein **Schedulingverfahren** als „optimal“?
  
  
  
  
  
  
  
  
  
  
- c) (2 P) Was versteht man unter einer **Prioritäteninversion**?
  
  
  
  
  
  
  
  
  
  
- d) (4 P) Wozu dient eine **Flusskontrolle** bei der Kommunikation zwischen 2 Tasks?  
Wie lässt sich diese z.B. einfach realisieren?

## Aufgabe 2 (30 Punkte) – Scheduling

Ein System zur Messdatenverarbeitung liest die Daten von **2** analogen Sensoren über **einen** A/D-Umsetzer ein, wozu es 2 Einlese-Tasks **SensIn-Task1** und **SensIn-Task2** besitzt. Die Messdaten werden danach an eine Signalverarbeitungstask **SV-Task** weitergeleitet. Anschließend werden die verarbeiteten Daten zum einen über eine **CAN-Task** an andere Systeme versendet als auch durch die **LCD-Task** auf einem Display visualisiert.

Die folgende Tabelle enthält die Zykluszeiten sowie die Laufzeiten der einzelnen Tasks:

Tasks	Zykluszeit [ms]	Laufzeit[ms]
SensIn-Task1	10	1
SensIn-Task2	20	1
SV-Task	(zu berechnen!)	2 (Daten von SensIn-Task1) 2 (Daten von SensIn-Task2)
CAN-Task	20	2
LCD-Task	800	200

(Die **Deadline** der Tasks **entspricht** der **Periodendauer/Zykluszeit**.)

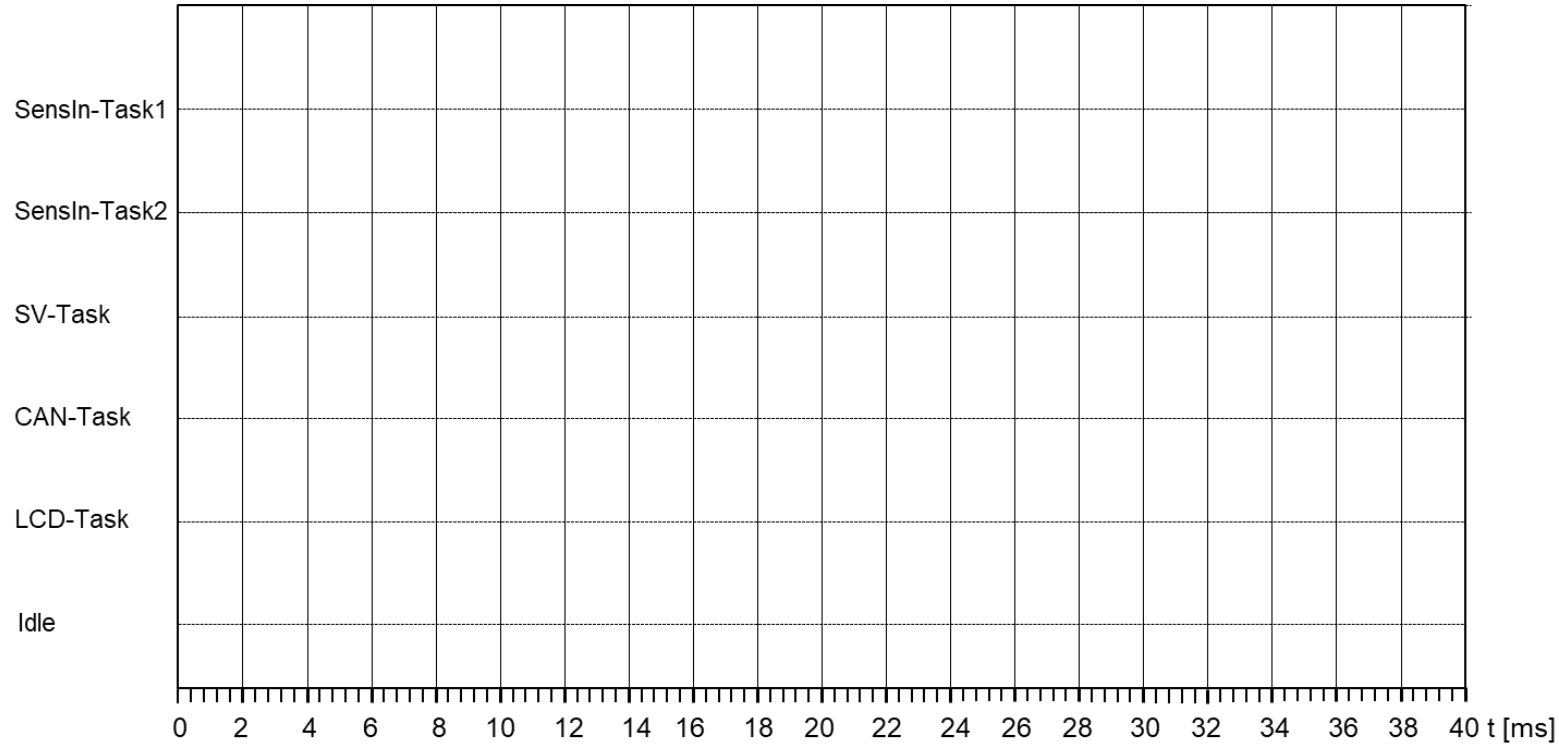
- a) (4 P) Berechnen Sie die maximale **Prozessorlast**, die durch das **Taskset** verursacht wird! Ist das gegebene Taskset grundsätzlich **realisierbar**?

- b) (14 P) Als eine Realisierungsvariante soll ein **Time-Slice-Scheduling** betrachtet werden. Zeichnen Sie unter der Annahme einer **Zeitscheibengröße** von **T=1ms** einen beispielhaften Schedulingverlauf für den Zeitraum 0..40ms in das beiliegende Diagramm.

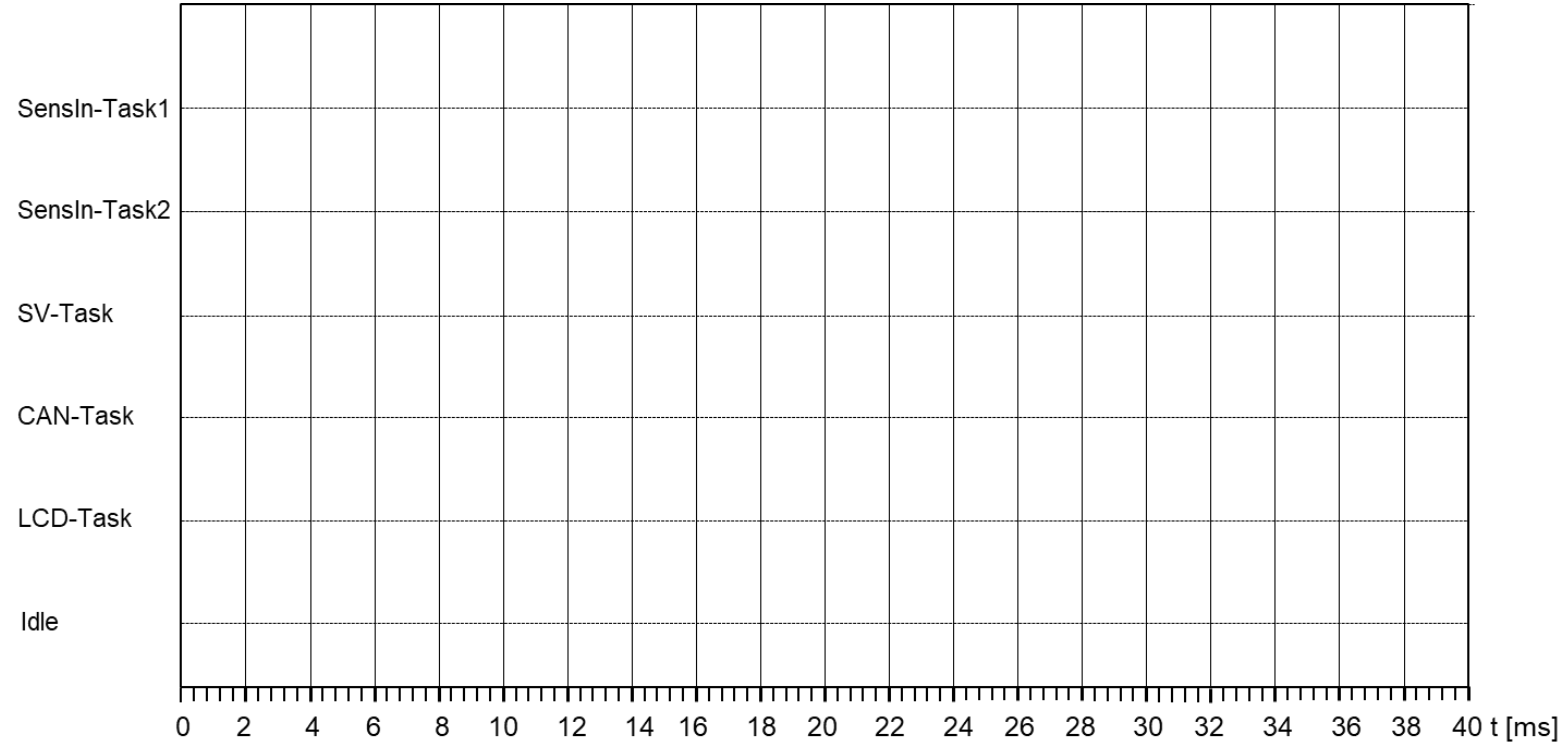
Ist das System mit diesem Verfahren zu realisieren? Begründen Sie Ihre Aussage!

- c) (6 P) Als weitere Realisierungsvariante soll untersucht werden, ob sich das Taskset durch ein **preemptives RMS-Scheduling** realisieren lässt.
- Welche **Prioritäten** müssen den **Tasks** dann jeweils zugewiesen werden?  
(**Höchste Priorität : 0**)
  
  - Wie lautet das Kriterium, damit das Taskset in jedem Fall mit RMS umsetzbar ist?
  
  - Ist dieses Kriterium in diesem Fall erfüllt?
- d) (6 P) Passen Sie die **Periode der LCD-Task** (nach oben oder nach unten) so an, dass bei Einsatz des preemptiven RMS-Scheduling eine Umsetzung in jedem Fall gegeben ist und gleichzeitig die Daten mit **maximaler Rate visualisiert** werden!

zu b)



(Reserve)



### Aufgabe 3 (20 Punkte) – Synchronisation/Kommunikation

Entwerfen Sie für das in Aufgabe 2 gegebene System eine Synchronisations-/Kommunikationsstruktur, welche den in der Aufgabe genannten Anforderungen genügt, jedoch ohne diese überzuerfüllen!

Das System besteht aus den folgenden relevanten Hardware-/Software-Komponenten:

- A/D-Umsetzer (Hardware)
- SensIn-Task1
- SensIn-Task2
- SV-Task
- CAN-Task
- CAN-Schnittstelle (Hardware)
- LCD-Task
- LCD (Hardware)

Anmerkung: Die genannten Hardware-Komponenten sind nur relevant hinsichtlich damit eventuell in Verbindung stehender Synchronisations-/Kommunikations-Konstrukte und nicht als Systeme an sich!

Folgende Funktionen stehen seitens des Betriebssystems zur Verfügung:

Kommunikationsmittel	Funktionen
Shared Memory	SharedMemWrite(), SharedMemRead()
Memory Pool	MemPoolAlloc(), MemPoolRelease()
Message Queue	MsgQPost(), MsgQPend()
Mutex	MuxPost(), MuxPend()
Semaphore	SemPost(), SemPend()
Event Flags	FlagPost(), FlagPend()

- a) (10 P) Entwerfen Sie eine Kommunikationsstruktur in UML-/SysML-Notation, welche die o.g. Anforderungen umsetzt! Vermerken Sie an den Assoziationen der Tasks mit den Kommunikationsmitteln die jeweils benutzten Methoden.

- b) (10 P) Erläutern Sie die Funktionsweise Ihrer gewählten Anordnung anhand von Pseudocode oder eines Aktivitätsdiagramms für die **SensIn-Task1**, **SV-Task** und die **CAN-Task**!