

<p style="text-align: center;">Ostfalia Hochschule für angewandte Wissenschaften</p> <p>Fakultät Fahrzeugtechnik Prof. Dr.-Ing. V. von Holt Institut für Fahrzeuginformatik und Fahrzeugelektronik</p>	<p style="text-align: center;">Modulprüfung Embedded Systems</p> <p style="text-align: center;">WS 2012/13 23.01.2013</p>	<p>Name:.....</p> <p>Vorname.....</p> <p>Matr.Nr.:.....</p> <p>Unterschrift.....</p>
---	---	--

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

Echtzeitbetriebssysteme:

1 (12)	2 (22)	3 (20)	Summe (54)	Note

Embedded Systems:

Ausarbeitung (50%)	Labor (50%)	Summe	Note

Aufgabe 1 (12 Punkte) – Kurzfragen

- a) (5 P) Bei Echtzeitsystemen müssen Aktionen u.a. „**rechtzeitig**“ erfolgen. Was versteht man unter dem Begriff „**Rechtzeitigkeit**“ und welche **Varianten** der „**Rechtzeitigkeit**“ gibt es (Skizze)?

- b) (3 P) Was versteht man unter einer „**Task**“?
Was versteht man unter einem Wechsel des „**Taskkontexts**“?

- c) (4 P) Warum ist es bei **prioritätsgesteuerten Echtzeitbetriebssystemen** wichtig, dass alle Tasks sog. **blockierende Betriebssystemaufrufe** (z.B. Warten auf Nachricht/Warten auf Timer) ausführen? Was passiert, wenn dies nicht der Fall ist?

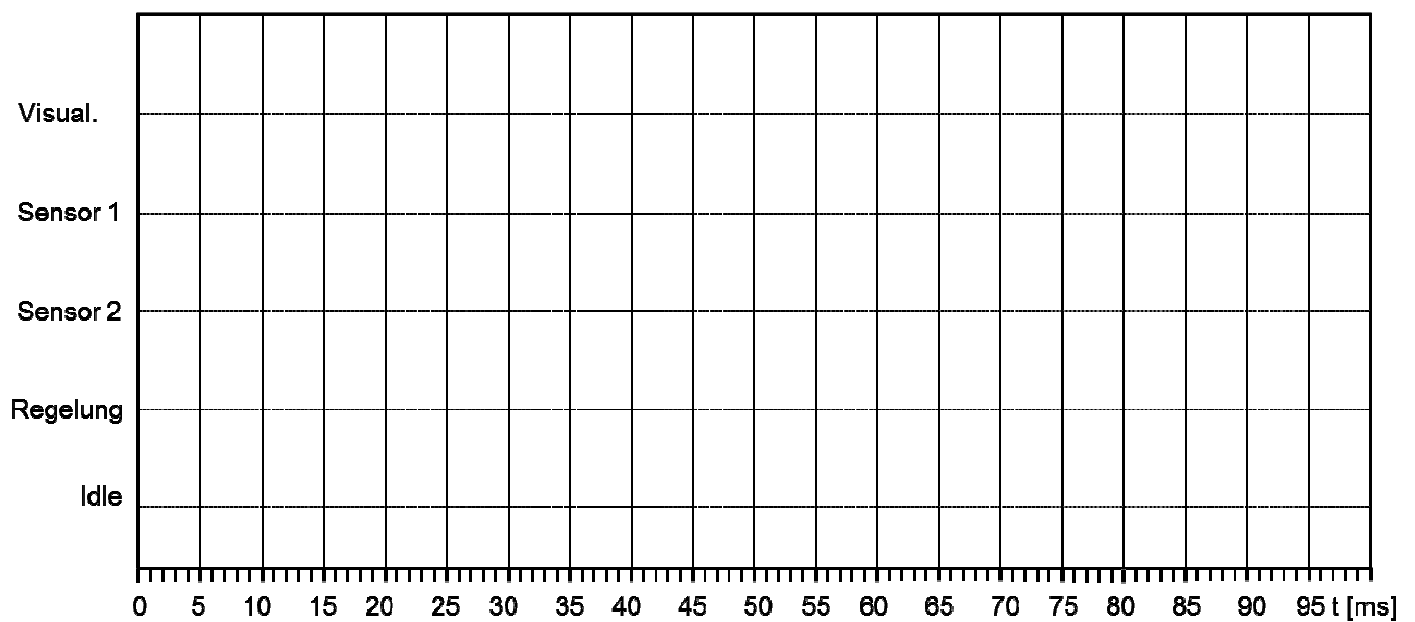
Aufgabe 2 (22 Punkte) – Scheduling

Ein Regelungssystem verfügt zur Messwerterfassung über 2 Sensoren, die in unterschiedlichen Intervallen Messwerte zur Zustandserfassung des Systems liefern. Die Sensormesswerte sowie die Regelung sollen in jeweils eigenen Tasks ablaufen. Zur Visualisierung und Steuerung des Systems ist eine weitere Task vorgesehen. Die folgende Tabelle enthält die Zykluszeiten sowie die Laufzeiten der einzelnen Tasks:

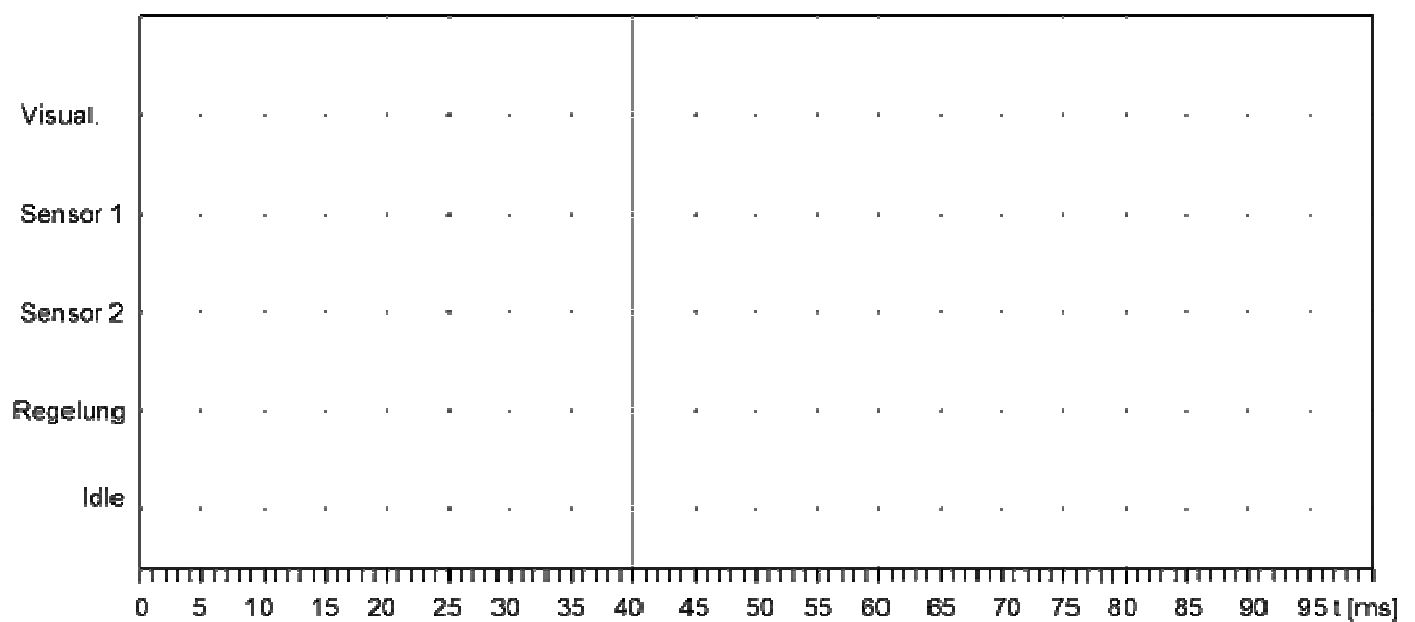
Tasks	Zykluszeit [ms]	Laufzeit[ms]
Sensor 1	10	1...2
Sensor 2	5	1...2
Regelung	20	4
Visualisierung	50	5

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

- a) (4 P) Berechnen Sie die **Prozessorlast**, die durch das **Taskset** verursacht wird! Ist das gegebene Taskset **realisierbar**?
- b) (10 P) Das Taskset soll durch ein **Rate-Monotonic-Scheduling** realisiert werden soll.
Welche **Prioritäten** müssen den **Tasks** jeweils zugewiesen werden? (**Höchste Priorität : 0**)
Nach welcher **Regel** werden die **Prioritäten vergeben**?
Ist das **Taskset** in jedem Fall mit RMS-Scheduling **umsetzbar**?
Weisen Sie den Schedule für den **Worst-Case** anhand des untenstehenden **Schedulediagramms** nach! Gehen Sie dabei davon aus, dass zum **Zeitpunkt t=0 alle Tasks lauffähig** sind bzw. alle Ereignisse eintreffen!



- c) (8 P) Alternativ soll geprüft werden, ob das Taskset auch durch ein **Time-Slice-Scheduling** realisierbar ist. Für die Task mit der **höchsten Prozessorlast** soll die Zeitscheibe zu **2ms** gewählt werden. Wählen Sie passende Zeitscheiben für die anderen Tasks und weisen Sie im untenstehenden Schedulediagramm nach, ob die Realisierbarkeit gegeben ist oder nicht!

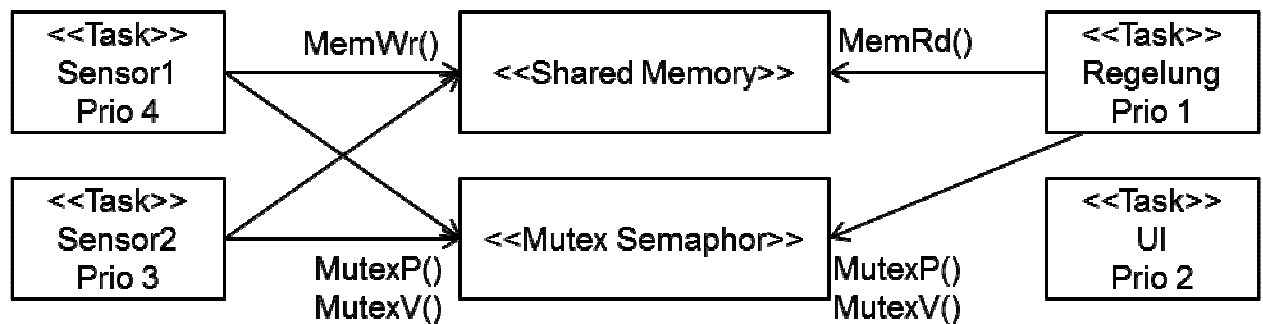


Aufgabe 3 (20 Punkte) – Semaphore / Synchronisation / Kommunikation

- a) (8 P) In einem Regelungssystem ähnlich der Aufgabe 2 existieren 4 Tasks. 2 Tasks dienen dazu Sensorwerte einzulesen, die über ein Shared Memory der Regelungs-Task übergeben werden. Um das Shared Memory gegenüber konkurrierenden Zugriffen zu schützen, ist ein Mutex-Semaphor zur Absicherung vorgesehen..

Folgende Funktionen stehen zur Verfügung:

<code>void MutexP()</code>	Passieren des Mutex-Semaphor.
<code>void MutexV()</code>	Verlassen des Mutex-Semaphor.
<code>void MemWr()</code>	Schreibt eine Nachricht in das Shared Memory.
<code>int MemRd()</code>	Liest eine Nachricht aus dem Shared Memory.
<code>void LeseSensor()</code>	Funktion der Sensor-Tasks zum Einlesen der Messwerte
<code>void Regelung()</code>	Funktion der Regelungs-Task zur Verarbeitung der Messwerte und Regelung



Geben Sie den **Ablauf des Zugriffs** auf das **Shared Memory** aus einer der **Sensor-Tasks** und aus der **Regelungs-Task** in Form eines **Aktivitätsdiagramms** oder in Form von **Pseudocode** an!

- b) (8 P) Das o.a. Kommunikationskonzept hat mehrere **Nachteile**: Einer besteht darin, dass **keine Reihenfolgesynchronisation** zwischen den **Sensor-Tasks** und **der Regelungs-Task** erfolgt. Die **Regelungs-Task** kann das **Shared Memory belegen**, auch wenn gar **keine neuen Sensormesswerte** vorliegen. **Erweitern** Sie das **vorhandene** Konzept um einen Kommunikations-/Synchronisationsmechanismus, der diesen **Nachteil behebt**!

- c) (4 P) Ein weiterer Nachteil besteht in **der Vergabe der Prioritäten** an die Tasks. Welcher **Fehler** fällt Ihnen dabei auf bzw. welche **besondere Eigenschaften** muss das Echtzeitbetriebssystem besitzen, damit das Scheduling zufriedenstellend läuft?