

Übung 4

Aufgabe 1 Prozesssteuerung mit Zeitbedingungen in Ada

Das "einfache eingebettete" System aus der Vorlesung (vgl. S. 2-41 ff, Übung 2/Aufgabe 4 und Übung 2/Aufgabe 1) soll als Echtzeitsystem realisiert werden und die folgenden Zeitbedingungen erfüllen:

1. Die Steuerung der Heizung soll als periodischer Prozess mit der Periode 1s realisiert werden. D.h. innerhalb einer Sekunde muss ein Messwert gelesen, eine Stellwert berechnet und ausgegeben sein.
2. Bei der Eingabe der Temperatur ist ein Eingabeflattern von höchstens 5ms zu berücksichtigen. Bei der Ausgabe des Heizungs-Stellwerts ein Ausgabeflattern von 10ms zu berücksichtigen
3. Die Steuerung des Drucks soll als periodischer Prozess mit einer Periode von 2s realisiert werden.
4. Ein-/Ausgabeflattern beim Druckprozess muss wie beim Temperaturprozess berücksichtigt werden (5ms Eingabe, 10ms Ausgabe).
5. Für die Ausgaben der gemessenen Werte (Druck und Temperatur) gibt es keine Zeitbedingungen. Die Konsolenausgabe darf aber die Steuerungsprozesse nicht "ausbremsen", d.h. wenn durch die Konsolenausgabe eine Deadline in einem Steuerungsprozess (Temperatur und Druck) nicht eingehalten werden kann, soll die Konsolen-Ausgabe unterbleiben.

Aufgabe 2 Prozessbedingungen mit Zeitsteuerung in C/Real-Time POSIX

Entwerfen Sie eine Lösung für die Prozesssteuerung als Echtzeitsystem mit den Bedingungen von Aufgabe 1 in C/Real-Time POSIX.

Aufgabe 3 Zufahrtskontrolle zu einem Parkplatz

Die Lösung zu Aufgabe 2, Übung 3, beinhaltet die Task Einfahrt für die Steuerung der Einfahrtschranke folgende Codesequenz, mit der festgestellt wird, ob das Fahrzeug nach Öffnen der Schranke den Schrankenbereich verlassen hat:

```
ED: EDurchfahrt;
...
ED = True;          -- Durchfahrt beginnt
while (ED = True) loop Read(ED); endloop; -- Durchfahren lassen
```

Analog beinhaltet in der Task Ausfahrt für die Steuerung der Ausfahrtschranke eine ähnliche Codesequenz:

```
AD: ADurchfahrt;
...
AD = True;          -- Durchfahrt beginnt
while (AD = True) loop Read(AD); endloop; -- Durchfahren lassen
```

Verändern Sie die Lösung so, dass diese beiden Codesequenzen mit einer Deadline von 10s ausgeführt werden müssen. Wenn nach 10s das Fahrzeug den Schrankenbereich nicht verlassen hat (d.h. die while-Schleife beendet ist), soll das System in einen "Notfall-Modus" gehen.

Im Notfallmodus übernimmt die Signal-Steuerung die Kontrolle. D.h. das Signal wird auf Rot gesetzt, die anderen Tasks werden blockiert solange bis der Notfall-Modus wieder aufgehoben wird.

Der Notfallmodus wird kann "von außen" wieder aufgehoben werden. Sehen Sie hierfür einen geeigneten Eintrittspunkt der Signal-Steuerung vor, dem die aktuelle Anzahl der belegten Parkplätze übergeben wird. Wer diesen Eintrittspunkt aufruft, soll hier nicht berücksichtigt werden.

Nach Aufhebung des Notfallmodus werden alle Schranken geschlossen und der Prozess läuft "normal" weiter.