michael.engel [☺] hs-coburg.de

Praktische Aufgaben zur Vorlesung
Eingebettete Betriebssysteme
Wintersemester 2017/18

Praktikumsaufgabe 1

Besprechung am Freitag, 13. Oktober 2017

Hinweise: Praktikumsaufgaben müssen in der vorgegebenen Zeit in den eingeteilten Gruppen bearbeitet werden. Die Lösung der Aufgaben wird bewertet, jedes Gruppenmitglied soll Fragen zu den einzelnen Aufgaben beantworten können. Bei Unklarheiten und Rückfragen wird selbstverständlich eine Hilfestellung gegeben.

1.1 Blinkenlights

In dieser Aufgabe legen Sie das erste Projekt für den Cortex-M3 an und geben ein sich änderndes Muster auf die Leuchtdioden an Port 1 aus.

a. Legen Sie ein neues Projekt in Keil an, wählen Sie als Prozessortyp den NXP LPC1778 aus und kopieren Sie die angegebenen Startup- und Systemdateien in das Projekt (wird vorgestellt).

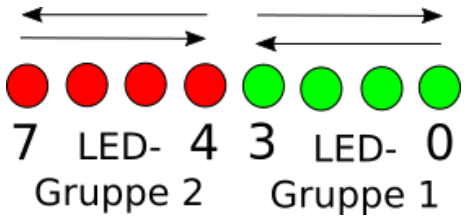
b. Erstellen Sie eine Datei `main.c` mit einer Funktion `int main(void)` und fügen Sie diese dem Projekt zu. Fügen Sie dann zwei Headerfiles dazu, die Standard-Zahlentypen und Definitionen für die Hardware des LPC1778 definieren:

```
#include <stdint.h>
#include <LPC177x_8x.h>
```

c. Die Platine mit acht Leuchtdioden ist an den GPIO-Port 0 Bits 8...15 angeschlossen.

Erstellen Sie eine `void init(void)`-Funktion, die die Pins 8...15 des GPIO-Ports 0 auf Ausgang schaltet und die LEDs ausschaltet und rufen Sie diese von `main` aus auf. Ziehen Sie zur GPIO-Programmierung das Datenbuch des LPC1778 zu Rate (im Moodle). Die GPIO-Ports sind *memory mapped* über eine Struktur anzusteuern, auf die der (im Header vorgegebene) Zeiger `LPC_GPIO0` zeigt.

d. Implementieren Sie zwei Funktionen `void led1(void)` und `void led2(void)`.



Mit `led1` sollen die LEDs 0...3 zyklisch hin und her laufen, also 0→1→2→3 →2→1→0→1 usw.

Mit `led2` sollen die LEDs 7...4 zyklisch hin und her laufen, also 7→6→5→4 →5→6→7→6 usw.

Testen Sie die beiden Funktionen.

1.2 Kooperatives Multitasking

In dieser Aufgabe sollen nun die LED-Gruppen 0...3 und 7...4 *unabhängig voneinander* laufen können.

a. Rufen Sie von einer Endlosschleife in `main` aus abwechselnd die Funktionen `led1` und `led2` auf.

   Welches Problem tritt hier auf?

b. Versuchen Sie, die Funktionen so abwechselnd aufzurufen, dass die Zyklen der beiden LED-Gruppen *gleichzeitig* zu laufen scheinen. Sie können einen Zustand in den `led`-Funktionen mit Hilfe einer `static`-Variablen speichern, z.B.

```
void led1(void) {
  static uint32_t led1_wert = 4711;
  ...
}
```

   Die Variable `led1_wert` wird nur beim ersten Eintritt in die Funktion initialisiert und behält ihren Wert auch, wenn Sie die Funktion mit `return` verlassen.

c. Ändern Sie Ihr Programm nun so ab, dass die Zyklen der beiden LED-Gruppen *mit unterschiedlicher Frequenz* laufen, z.B. Gruppe 1 dreimal so schnell wie Gruppe 2.

   Hinweis: Verwenden Sie hierfür in `main` z.B. eine Variable `uint32_t tick_counter`, die Sie in jedem Durchlauf der Endlosschleife inkrementieren und nutzen Sie diese als Zeitgeber.