

Aufgabe 3: pLED-Modul (14 Punkte, 2er-Gruppen, Abgabe bis Mi. 10.06.09 10:00 Uhr)

In dieser Aufgabe sollen Sie den Umgang mit Zeigern und Arrays erlernen und hierzu Ihre Lösung zu Aufgabe 2 (soweit abweichend) an die folgenden Regeln anpassen sowie ein ebenfalls im Folgenden beschriebenes neues Testprogramm in einer Datei **test.c** entwickeln. Legen Sie zu Beginn analog zu Aufgabe 2 ein neues Projekt **aufgabe3** an und kopieren Sie den Inhalt Ihrer **led.c** aus Aufgabe 2 in die neu angelegte **led.c**.

a) LED-Modul

- Bessern Sie zunächst alle in der Korrektur erkannten Fehler in Ihrem LED-Modul aus.
- Achten Sie darauf, dass Ihr Modul exakt die durch das gegebene **led.h** beschriebene Schnittstelle (Vertrag des Moduls) erfüllt. Weitere Funktionen oder globale Variablen im LED-Modul, die in diesem Header nicht beschrieben werden, sind in ihrer Sichtbarkeit auf das Modul zu beschränken.
- Legen Sie zwei Arrays mit je 8 Elementen an, wobei die Nummer einer LED zur Indizierung des Arrays verwendet wird. Das erste Array soll für jede LED einen Zeiger auf den Port, das zweite die Pin-Nummer (entweder direkt oder kodiert in eine Maske Ihrer Wahl) enthalten, an denen die LED angeschlossen ist (an Index 4 würden sich also die Parameter der LED RED1 befinden, also ein Zeiger auf das PORTC-Register im ersten und die PIN-Nummer 7 oder eine passende Maske im zweiten Array).
- Passen Sie alle Funktionen des Arrays, in denen der Zustand von LEDs direkt geändert wird, so an, dass diese die soeben angelegten Arrays verwenden und ggf. vorhandene IF-ELSE-Kaskaden durch ein einheitliches, von der Zahl der vorhandenen LEDs unabhängiges Stück Code ersetzt werden.

b) Testprogramm

- Kopieren Sie sich das vorgegebene Testprogramm aus dem Vorgabeverzeichnis. Dieses enthält eine Bewegungssequenz, die durch Arrays beschrieben wird. Implementieren Sie nun die zwei leeren Funktionen, die zur Anzeige erforderlich sind.
- Funktion `simpleSeq()`: Die Funktion erhält als Parameter einen Zeiger auf ein Array von Bytes sowie dessen Zahl an Elementen. Die Werte des Arrays beschreiben jeweils eine LED-Konstellation, die direkt an `sb_led_set_all_leds()` übergeben werden kann. Die Funktion soll nun der Reihe nach jede der Konstellationen anzeigen und hierbei bei jeder Konstellation die aktuelle Position (=Index) im Feld auf den 7-Segment-Anzeigen darstellen. Nach der Anzeige soll jeweils exakt eine Sekunde gewartet werden (Timer-Modul der Bibliothek).
- Funktion `complexSeq()`: Die Funktion erhält als Parameter einen Zeiger auf ein Array, dessen Elemente wiederum Zeiger auf Arrays sind, wie sie der Funktion `simpleSeq()` übergeben werden, sowie die Länge des Arrays und die Länge der referenzierten Arrays (die Länge sämtlicher referenzierter Arrays soll identisch sein).
- Testen Sie Ihre Implementierung mit dem vorgegebenen Testprogramm. Erstellen Sie dann eine eigene Bewegungssequenz, und geben Sie das nun erhaltene Testprogramm mit ab.

Feiertagsregelung für die Tafelübungen:

- Übung 4: Am 21.5. finden aufgrund des Feiertags keine Übungen statt. Betroffene Teilnehmer werden gebeten, eine andere Tafelübung im Zeitraum 22.5.-27.5. zu besuchen.
- Übung 5: Aufgrund der Pfingsttage und Fronleichnam entfällt die Übungswoche 28.5-3.6. Die Übungen 5 finden in der Übungswoche 4.6. - 10.6. statt. Am 12.6. entfallen die Tafelübungen.
- Ab Übung 6 wird bis auf weiteres eine Mo-Fr-Übungswoche gefahren. Übung 6 findet in der Woche 15.6.-19.6. statt.
- Die Rechnerübungen werden mit Ausnahme der gesetzlichen Feiertage wie gewohnt betreut.