SPiC-Aufgabe #3: led

(14 Punkte, in Zweier-Gruppen)

Reimplementieren Sie das LED-Modul der `libspicboard` in der Datei `led.c`. Die Beschreibung der zu implementierenden Schnittstelle finden Sie in der API-Dokumentation auf der SPiC-Webseite unter:

https://sys.cs.fau.de/lehre/ss25/spic/uebung/spicboard/libapi/extern/group_LED.html

Schreiben Sie außerdem ein Testprogramm `test-led.c`, welches alle öffentlichen Funktionen des Moduls testet.

LED-Modul

- Halten Sie sich genau an die vorgegebene Schnittstelle. Binden Sie hierzu die `led.h` aus der `libspicboard` in Ihre `led.c` ein.
- Weitere Funktionen oder globale Variablen, die in diesem Header nicht deklariert werden, sind in ihrer Sichtbarkeit auf das Modul zu beschränken.
- Verzichten Sie zum Ansteuern der Pins auf `if`-Kaskaden und `switch`-Anweisungen oder vergleichbare C-Konstrukte. Verwenden Sie stattdessen Arrays (und ggf. Zeiger), um Port- und Pinnummer zu bestimmen. Dazu dürfen Sie davon ausgehen, dass der LED-Aufzählungstyp die LEDs von 0 bis 7 durchnummeriert (siehe `led.h`) und diesen als Index benutzen.
- Beachten Sie, dass die I/O-Ports des ATmega-Mikrocontrollers vor der ersten Verwendung entsprechend initialisiert werden müssen. Dieses Detail soll jedoch (wie im Original) innerhalb des LED-Moduls verborgen bleiben.

Testprogramm

- Ihr Testprogramm soll alle Funktionen des LED-Moduls testen. Vergessen Sie hierbei nicht, auch das korrekte Verhalten bei fehlerhaften Eingaben maschinell zu überprüfen.
- Ihr Testprogramm soll für alle Funktionen des LED-Moduls jeweils eine repräsentative Auswahl an Übergabeparameter testen. Denken Sie auch daran Randfälle zu testen (zum Beispiel keine LEDs, alle LEDs, usw.).
- Überprüfen Sie Ihr Testprogramm auch mit der Referenzimplementierung aus der `libspicboard` und vergleichen Sie das Verhalten.
- Für Ihr Testprogramm können Sie auch die anderen Module der `libspicboard` verwenden. Sie können z. B. das ADC-Modul verwenden, um den Wert des Photosensors auszulesen und mit der LED-Pegelanzeige den Helligkeitspegel im Raum anzuzeigen.

Hinweise:

- Da in dieser Aufgabe die SPiC-IDE mehrere verschiedene Quelldateien zu einer ausführbaren Datei zusammenbinden muss, enthält sie für diesen Vorgang extra Regeln. Achten Sie darauf, dass Sie ihre Dateien **exakt** wie in der Aufgabenstellung vorgegeben (`led.c` und `test-led.c`) nennen. Ansonsten werden die Regeln nicht angewendet.
- „Simulate“ in der SPiC IDE funktioniert nur in der `test-led.c`, weil dort die `main` Funktion liegt, die simuliert werden soll.
- Der Linker verwendet stets die „lokalen“ Funktionen, die in den lokalen Quelldateien definiert sind. Nur wenn eine Funktion dort nicht definiert ist, werden die Bibliotheken durchsucht. Um testweise statt der eigenen LED-Funktionen in der Datei `led.c` die LED-Funktionen der `libspicboard` zu verwenden, können Sie einfach Ihre eigene Implementierung auskommentieren, z. B. durch eine Zeile
`#if 0`
am Anfang der Datei und eine Zeile
`#endif`
am Ende der Datei. Aktivieren können Sie Ihre Implementierung dann wieder, indem Sie die 0 in eine 1 abändern.

-
- Begründen Sie die Verwendung von allen `volatile` Variablen. Wenn für mehrere Variablen die selbe Begründung gilt, dürfen Sie diese gemeinsam begründen.
 - Treffen Sie keine Annahmen über den initialen Zustand der Hardware.

Abgabezeitpunkt

Per Skript im CIP abfragbar: `/proj/i4spic/bin/get-deadline aufgabe3 Txx`