

---

## SPiC-Aufgabe #2: Interrupt Zähler

(12 Punkte, keine Gruppen)

Schreiben Sie das Programm `izaehler.c`, das die Tastendrucke an Taster 0 zählt (fallende Flanke). Wird der Taster 1 gedrückt, sollen die gezählten Tastendrucke angezeigt werden. Die Einer- und Zehnerstelle wird auf der Siebensegmentanzeige und die Hunderterstelle mit den LEDs angezeigt (LED0 = 100, LED1 = 200, etc.).

- Durch Tastendruck auf Taster 0 soll der Zähler inkrementiert werden.
- Die Anzeige soll bei einem Tastendruck auf Taster 1 aktiviert werden. Ändert sich der Zählerstand während die Anzeige aktiv ist, dann soll die Anzeige aktualisiert werden.
- Um Energie zu sparen, soll der Mikrocontroller so oft wie möglich in den Ruhezustand versetzt werden. Dazu soll insbesondere die Erkennung von Tastendrucke nicht durch aktives Warten, sondern durch Interrupts implementiert werden.
- Des Weiteren soll die Anzeige nach einer bestimmten Zeit ohne Tastendruck auf Taster 1 abgeschaltet werden. Die Anzeigedauer soll über das Potentiometer im Bereich von 1 bis 10 Sekunden eingestellt werden können.
- Sollte der Zähler den anzeigbaren Wertebereich verlassen, wird er auf 0 zurückgesetzt.
- Dokumentieren Sie im Quellcode aus welchem Grund die Interrupts aktiviert bzw. deaktiviert werden können/müssen.

### Hinweise:

- Nutzen Sie die `libspicboard` sowohl zur Ansteuerung von Siebensegmentanzeige und LEDs als auch zum Auslesen des Potentiometer-Wertes. Die Verwendung der Button-Funktionen der `libspicboard` ist jedoch **nicht zulässig**.
- Um die Wahrscheinlichkeit, dass Tastendrucke verloren gehen, zu minimieren, dürfen die Interrupts während des Aufrufs der `sb_led`-, `sb_7seg`- und `sb_adc`-Funktionen nicht gesperrt sein.
- Verwenden Sie das Datenblatt um herauszufinden an welchen Interrupts die Taster (PD2, PD3) angeschlossen sind. Dort finden Sie auch die nötigen Informationen um das EICRA und EIMSK-Register richtig zu konfigurieren.
- Interrupts sollen, wie immer, möglichst kurz gehalten werden.

### Abgabezeitpunkt

T01 08.11.2017 18:00:00