

---

## SPiCAufgabe #2: Interrupt Zähler

(12 Punkte, Mittwoch, 04.11.2015, 18:00, keine Gruppen)

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

- Durch Tastendruck auf Taster 1 soll der Zähler inkrementiert werden.
- Die Anzeige soll bei einem Tastendruck auf Taster 0 aktiviert werden.
- Um Energie zu sparen, soll der Mikrocontroller so oft wie möglich in den Ruhezustand versetzt werden.
- Des Weiteren soll die Anzeige nach einer bestimmten Zeit ohne Tastendruck auf Taster 0 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 `MCUCR` und `GICR`-Register richtig zu konfigurieren.
- Interrupts sollen, wie immer, möglichst kurz gehalten werden.