

Übungen zu Systemnahe Programmierung in C

Abschnitt 7.5: Aufgabe (spiel)

08.06.2020

Tim Rheinfels
Benedict Herzog
Bernhard Heinloth

Lehrstuhl für Informatik 4
Friedrich-Alexander-Universität Erlangen-Nürnberg



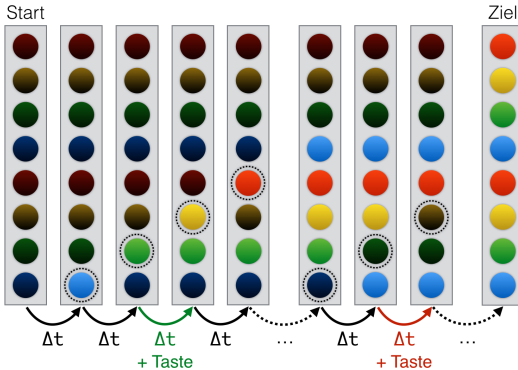
Lehrstuhl für Verteilte Systeme
und Betriebssysteme



FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

TECHNISCHE FAKULTÄT

- Spielcursor wandert dabei über LED-Reihe hin und her und invertiert (engl. toggle) den LED-Zustand
- LED-Zustand bleibt durch Drücken des Tasters erhalten
- Ziel: alle LEDs zum Leuchten bringen





- Nach einem Level wird eine Siegessequenz auf den LEDs dargestellt

```
01 void main(void) {  
02     // Initialisierung  
03     // [...]  
04  
05     while(1) {  
06         // starte Level  
07         // [...]  
08  
09         // Siegessequenz anzeigen  
10         // [...]  
11  
12         // Level aktualisieren  
13         // [...]  
14     }  
15 }
```

■ Ziele:

- Flankenerkennung in Hardware
- Ereignisverarbeitung mittels Interrupts
- **Keine** Verwendung der libspicboard

■ Details:

- BUTTON0 ist an PD2 angeschlossen
- PD2 als Eingang (mit aktivierten Pull-Up) konfigurieren
- PD2 ist der Eingang von INT0
- Welche Flanke/Pegel muss für den Interrupt konfiguriert werden?
- Wie sieht ein minimaler Interrupthandler für die Aufgabe aus?

→ 7.4

- Spielgeschwindigkeit bestimmt Schwierigkeit
 - ⇒ Passives Warten mittels timer Modul der libspicboard
- Schwierigkeit steigt mit jedem Level l
- Schwierigkeit nähert sich einer maximalen Geschwindigkeit an
 - ⇒ Folge von Wartezeiten $f_l = \frac{a}{l} + b$ (a und b sind Konstanten)

