

# Übungen zu Systemnahe Programmierung in C

## Abschnitt 7.3: Schlafmodi

---

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



- AVR-basierte Geräte oft batteriebetrieben (z.B. Fernbedienung)
- Energiesparen kann die Lebensdauer drastisch erhöhen
- AVR-Prozessoren unterstützen unterschiedliche Powersave-Modi
  - Deaktivierung funktionaler Einheiten
  - Unterschiede in der "Tiefe" des Schlafes
  - Nur aktive funktionale Einheiten können die CPU aufwecken
- Standard-Modus: Idle
  - CPU-Takt wird angehalten
  - Keine Zugriffe auf den Speicher
  - Hardware (Timer, externe Interrupts, ADC, etc.) sind weiter aktiv
- Dokumentation im ATmega328PB-Datenblatt



- Unterstützung aus der avr-libc: (`#include <avr/sleep.h>`)
  - `sleep_enable()` - aktiviert den Sleep-Modus
  - `sleep_cpu()` - setzt das Gerät in den Sleep-Modus
  - `sleep_disable()` - deaktiviert den Sleep-Modus
  - `set_sleep_mode(uint8_t mode)` - stellt den zu verwendenden Modus ein
- Dokumentation von `avr/sleep.h` in avr-libc-Dokumentation

```
01 #include <avr/sleep.h>
02
03 set_sleep_mode(SLEEP_MODE_IDLE); // Idle-Modus verwenden
04 sleep_enable(); // Sleep-Modus aktivieren
05 sleep_cpu(); // Sleep-Modus betreten
06 sleep_disable(); // Empfohlen: Sleep-Modus danach deaktivieren
```



- Dornröschenschlaf  
⇒ **Problem:** Es kommt genau ein Interrupt

## Hauptprogramm

```
01 sleep_enable();
02 event = 0;
03
04
05 while(!event) {
06     sleep_cpu();
07 }
08
09
10
11
12 sleep_disable();
```

## Interruptbehandlung

```
01 ISR(TIMER1_COMPA_vect) {
02     event = 1;
03 }
```



- Dornröschenschlaf  
⇒ **Problem:** Es kommt genau ein Interrupt

## Hauptprogramm

```
01 sleep_enable();
02 event = 0;
03
04
05 while(!event) {
06     ⚡ Interrupt ⚡
07     sleep_cpu();
08
09 }
10
11
12 sleep_disable();
```

## Interruptbehandlung

```
01 ISR(TIMER1_COMPA_vect) {
02     event = 1;
03 }
```



## ■ Dornröschenschlaf

⇒ **Problem:** Es kommt genau ein Interrupt

⇒ **Lösung:** Interrupts während des kritischen Abschnitts sperren

### Hauptprogramm

```
01 sleep_enable();
02 event = 0;
03
04 cli();
05 while(!event) {
06     sei();
07     sleep_cpu();
08     cli();
09 }
10 sei();
11
12 sleep_disable();
```

### Interruptbehandlung

```
01 ISR(TIMER1_COMPA_vect) {
02     event = 1;
03 }
```



## ■ Dornröschenschlaf

⇒ **Problem:** Es kommt genau ein Interrupt

⇒ **Lösung:** Interrupts während des kritischen Abschnitts sperren

### Hauptprogramm

```
01 sleep_enable();
02 event = 0;
03
04 cli();
05 while(!event) {
06     sei(); ⚡ Interrupt ⚡
07     sleep_cpu();
08     cli();
09 }
10 sei();
11
12 sleep_disable();
```

### Interruptbehandlung

```
01 ISR(TIMER1_COMPA_vect) {
02     event = 1;
03 }
```

⇒ Was ist wenn der Interrupt zwischen Zeile 6 und 7 auftritt?

⇒ sei() ist um eine Instruktion verzögert!