

# Informationstechnik-Praxis / AVR-Übungsblatt 10

---

## Lernziele / Tipps / allgemeine Voraussetzungen:

- **Alternative Programmierung in C (statt in Assembler)**
- **Nutzung von Programmablaufplänen zur Visualisierung des Programmablaufs**

Beschaltung des AVR-Boards für alle Aufgaben:

**Taster 1** an Port **D**, Pin-Nummer **2**

**Taster 2** an Port **D**, Pin-Nummer **3**

**Rote LED** an Port **C**, Pin-Nummer **0**

**Gelbe LED** an Port **C**, Pin-Nummer **1**

**Grüne LED** an Port **C**, Pin-Nummer **2**

## Was ist bei der letzten Klausur so schwer gefallen? (Notenschnitt 3,8)

### 1. Denken Sie an die vollständige Vorbereitung des Mikrocontrollers im main-Programm:

- Setzen des Stackpointers (wie gehabt und in der Regel vorgegeben)
- Setzen der I/O-Ports auf Eingang/Ausgang (wie gehabt)
- Aktivieren der Pullup-Widerstände (wie gehabt)
- **Für die Interruptverarbeitung:**
  - Eintrag der Adressen für die Interrupt-Service-Routinen in der Interrupt-Vektoren-Tabelle
  - Setzen des Registers GICR für die Auswahl von INT0 und/oder INT1
  - Setzen des Registers MCUCR für die Art der Interrupt-Triggerung (Flanken)
- **Für die Timer0-Benutzung:**
  - Bestimmen der Timerquelle bzw. des Vorteiler-Faktors im Register TCCR0
  - Timer-Funktionsweise bestimmen im Register TIMSK
  - Berechnen der Initialisierungswerte des TCNT0-Zählers
  - **(Gleiches gilt sinngemäß für die Timer 1 und 2)**

### 2. „Assembler ist mir ein Rätsel“

- Das kann man jetzt unterschiedlich kommentieren...  
Man könnte sich auch am Unterricht beteiligen und nachfragen – dafür ist der Unterricht ja gedacht!!! Oder sich eine passende Lernstrategie auswählen:  
Flucht, Angriff oder Totstellen? Überlegen Sie mal...

### 3. Vielleicht hilft ja, sich den ABLAUF eines Programms zu skizzieren!?

- Recherchieren Sie, ob der PAP-Designer auf den Laptops installiert ist.
- *Wer etwas an seiner Note drehen will, führt am Mittwoch den PAP-Designer vor, also: kurz einarbeiten und die Möglichkeiten ausloten! Gerne auch im Zweier-Team! (Mindest-Präsentationsdauer inkl. Rückfragen aus dem „Publikum“: 10 Minuten.)*

### 4. Vielleicht hilft auch die Verwendung der Programmiersprache C statt Assembler!?

- Öffnen Sie im myAVR\_Workpad die Beispielprogramme in C.
- Korrigieren Sie die PORT-Angaben in diesen Programmen, so dass sie den oben angeführten Hardware-Portbeschaltungen entsprechen!
- Dann müssten die Programme ohne weitere Änderungen auf Ihren AVR-Boards laufen!
- *Wer etwas an seiner Note drehen will, führt am Mittwoch die wesentlichen Unterschiede zwischen Assembler- und C-Programmierung vor, also: etwas einarbeiten und die Möglichkeiten ausloten! Gerne auch im Zweier-Team! (Mindest-Präsentationsdauer inkl. Rückfragen aus dem „Publikum“: 10 Minuten.)*  
*Beispiel-Fragestellungen: Was ist einfacher zu programmieren?  
Wo komme ich weder bei Assembler noch bei C drumherum, muss ich also so oder so wissen (oder wenigstens nachschlagen können)?*

## **Aufgabe 1 – rot-grünes Wechsellicht 1:**

Lassen Sie die **rote** und die **grüne** LED im Wechsel blinken, bei einer beliebigen, aber gut sichtbaren Wechselfrequenz.

**Realisieren Sie das Programm in der Programmiersprache C!**

## **Aufgabe 2 - Lauflicht:**

Die 3 LEDs sollen nacheinander aufleuchten und danach wieder von vorne beginnen, also: **rot-gelb-grün-rot-gelb-grün-rot-gelb-grün...**

**Realisieren Sie das Programm in der Programmiersprache C!**

## **Aufgabe 3 – Taster 2 steuert Wechsellicht:**

Wenn Taster 2 NICHT gedrückt ist, soll NUR die **grüne** LED leuchten, wird der Taster 2 gedrückt, soll NUR die **gelbe** LED leuchten!

**Realisieren Sie das Programm in der Programmiersprache C!**

Lässt sich die Tastensteuerung auch über Interrupt realisieren? Schauen Sie in die Beschreibung zum Register MCUOCR! (LO-Pegel löst fortlaufend Interrupts aus???)

## **Aufgabe 4 – KFZ-Fahrtrichtungsanzeiger („Blinker“):**

Die **rote LED** soll für den linken Fahrtrichtungsanzeiger stehen, die **grüne LED** entsprechend für den Fahrtrichtungsanzeiger auf der rechten Seite.

**Wenn kein Taster gedrückt wird, leuchtet und blinkt auch nichts.**

**Taster 1 steuert den linken Fahrtrichtungsanzeiger, Taster 2 den rechten Blinker.**

**Es darf nicht sein, dass beide LEDs zur gleichen Zeit blinken!**

**Realisieren Sie das Programm in der Programmiersprache C!**

Für die Funktionsweise können wir uns nun mehrere Varianten vorstellen:

### **Aufgabe 4a:**

Ein Druck auf einen Taster lässt die entsprechende LED 5 mal blinken. Während des Blinkens ist keine weitere Eingabe möglich.

Danach geht die LED selbstständig aus und eine neue Eingabe ist möglich.

### **Aufgabe 4b:**

Ein Druck auf einen Taster lässt die entsprechende LED ständig blinken. Während des Blinkens ist keine weitere Eingabe möglich.

Erst ein weiterer Druck auf die gleiche Taste beendet das Blinken und ermöglicht wieder neue Eingaben.

## **Aufgabe 5 – „Hello World – Goodbye World“:**

Zu Beginn sind alle LEDs ausgeschaltet.

**Taster 1** schaltet per Interrupt die **rote LED** ein,

**Taster 2** schaltet per Interrupt die **rote LED** wieder aus.

**Realisieren Sie das Programm in der Programmiersprache C!**

## **Aufgabe 6**

Blinken soll die **gelbe LED**. Stellen Sie bei einer Frequenz von 1 Hz einen Tastgrad von 0,2 ein (also 0,2 Sekunden EIN und 0,8 Sekunden AUS). Welchen Timer verwenden Sie und warum?

Berechnen Sie die Initialisierungswerte für den/die TCNT-Zähler des gewählten Timers.

**Realisieren Sie das Programm in der Programmiersprache C!**