

# Hardwarenahe Programmierung

## Übungsaufgaben – 21. November 2019

Diese Übung enthält Punkteangaben wie in einer Klausur. Um zu „bestehen“, müssen Sie innerhalb von 70 Minuten unter Verwendung ausschließlich zugelassener Hilfsmittel 14 Punkte (von insgesamt 25) erreichen.

### Aufgabe 1: Zahlensysteme

Wandeln Sie ohne Hilfsmittel

- |                    |                            |                 |
|--------------------|----------------------------|-----------------|
| • nach Dezimal:    | • nach Hexadezimal:        | • nach Binär:   |
| (a) $0010\ 0000_2$ | (d) $0010\ 0000_2$         | (g) $750_8$     |
| (b) $42_{16}$      | (e) $42_{10}$              | (h) $42_{10}$   |
| (c) $17_8$         | (f) $192.168.20.254_{256}$ | (i) $AFFE_{16}$ |

Berechnen Sie ohne Hilfsmittel:

- (j)  $750_8 \& 666_8$   
(k)  $A380_{16} + B747_{16}$   
(l)  $AFFE_{16} >> 1$

Die tiefgestellte Zahl steht für die Basis des Zahlensystems. Jede Teilaufgabe zählt 1 Punkt.  
(In der Klausur sind Hilfsmittel zugelassen, daher ist dies *keine* typische Klausuraufgabe.)

### Aufgabe 2: Mikrocontroller

An die vier Ports eines ATmega16-Mikrocontrollers sind Leuchtdioden angeschlossen:

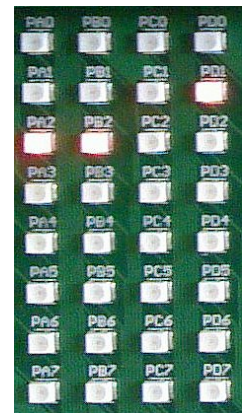
- von links nach rechts an die Ports A, B, C und D,
- von oben nach unten an die Bits Nr. 0 bis 7.

Wir betrachten das folgende Programm ([aufgabe-2.c](#)):

```
#include <avr/io.h>
```

```
int main (void)
```

```
{  
    DDRA = 0xff;  
    DDRB = 0xff;  
    DDRC = 0xff;  
    DDRD = 0xff;  
    PORTA = 0x1f;  
    PORTB = 0x10;  
    PORTD = 0x10;  
    PORTC = 0xfc;  
    while (1);  
    return 0;  
}
```



- (a) Was bewirkt dieses Programm? (4 Punkte)  
(b) Wozu dienen die ersten vier Zeilen des Hauptprogramms? (2 Punkte)  
(c) Was würde stattdessen die Zeile `DDRA, DDRB, DDRC, DDRD = 0xff;` bewirken? (2 Punkte)  
(d) Schreiben Sie das Programm so um, daß die durch das Programm dargestellte Figur spiegelverkehrt erscheint. (3 Punkte)  
(e) Wozu dient das `while (1)`? (2 Punkte)  
• Alle Antworten bitte mit Begründung.

*Viel Erfolg!*