

# Hardwarenahe Programmierung

## Übungsaufgaben 8 – 12. Dezember 2024

Diese Übung enthält Punkteangaben wie in einer Klausur. Um zu „bestehen“, müssen Sie innerhalb von 90 Minuten unter Verwendung ausschließlich zugelassener Hilfsmittel 16 Punkte (von insgesamt 32) erreichen.

### Aufgabe 1: Trickprogrammierung

Wir betrachten das folgende Programm (Datei: [aufgabe-1.c](#)):

```
#include <stdio.h>
#include <stdint.h>

int main (void)
{
    uint64_t x = 4262939000843297096;
    char *s = &x;
    printf ("%s\n", s);
    return 0;
}
```

Das Programm wird kompiliert und auf einem 64-Bit-Little-Endian-Computer ausgeführt:

```
$ gcc -Wall -O aufgabe-1.c -o aufgabe-1
aufgabe-1.c: In function 'main':
aufgabe-1.c:7:13: warning: initialization from incompatible pointer type [...]
$ ./aufgabe-1
Hallo
```

- (a) Erklären Sie die Warnung beim Compilieren. (2 Punkte)
- (b) Erklären Sie die Ausgabe des Programms. (5 Punkte)
- (c) Wie würde die Ausgabe des Programms auf einem 64-Bit-Big-Endian-Computer lauten? (3 Punkte)

Hinweis: Modifizieren Sie das Programm und lassen Sie sich Speicherinhalte ausgeben.

### Aufgabe 2: Thermometer-Baustein an I<sup>2</sup>C-Bus

Eine Firma stellt einen elektronischen Thermometer-Baustein her, den man über die serielle Schnittstelle (RS-232) an einen PC anschließen kann, um die Temperatur auszulesen. Nun wird eine Variante des Thermometer-Bausteins entwickelt, die die Temperatur zusätzlich über einen I<sup>2</sup>C-Bus bereitstellt.

Um das neue Thermometer zu testen, wird es in ein Gefäß mit heißem Wasser gelegt, das langsam auf Zimmertemperatur abkühlt. Alle 10 Minuten liest ein Programm, das auf dem PC läuft, die gemessene Temperatur über beide Schnittstellen aus und erzeugt daraus die folgende Tabelle:

Zeit / min.	Temperatur per RS-232 / °C	Temperatur per I <sup>2</sup> C / °C
0	94	122
10	47	244
20	30	120
30	24	24
40	21	168

- (a) Aus dem Vergleich der Meßdaten läßt sich auf einen Fehler bei der I<sup>2</sup>C-Übertragung schließen. Um welchen Fehler handelt es sich, und wie ergibt sich dies aus den Meßdaten? (5 Punkte)
- (b) Schreiben Sie eine C-Funktion `uint8_t repair(uint8_t data)`, die eine über den I<sup>2</sup>C-Bus empfangene fehlerhafte Temperatur `data` korrigiert. (5 Punkte)

### Aufgabe 3: Kondensator

Ein Kondensator der Kapazität  $C = 100 \mu\text{F}$  ist auf die Spannung  $U_0 = 5 \text{ V}$  aufgeladen und wird über einen Widerstand  $R = 33 \text{ k}\Omega$  entladen.

- (a) Schreiben Sie ein C-Programm, das den zeitlichen Spannungsverlauf in einer Tabelle darstellt. (5 Punkte)
- (b) Schreiben Sie ein C-Programm, das ermittelt, wie lange es dauert, bis die Spannung unter  $0.1 \text{ V}$  gefallen ist. (4 Punkte)
- (c) Vergleichen Sie die berechneten Werte mit der exakten theoretischen Entladekurve:  $U(t) = U_0 \cdot e^{-\frac{t}{RC}}$  (3 Punkte)

Hinweise:

- Für die Simulation zerlegen wir den Entladevorgang in kurze Zeitintervalle  $dt$ . Innerhalb jedes Zeitintervalls betrachten wir den Strom  $I$  als konstant und berechnen, wieviel Ladung  $Q$  innerhalb des Zeitintervalls aus dem Kondensator herausfließt. Aus der neuen Ladung berechnen wir die Spannung am Ende des Zeitintervalls.
- Für den Vergleich mit der exakten theoretischen Entladekurve benötigen Sie die Exponentialfunktion `exp()`. Diese finden Sie in der Mathematik-Bibliothek: `#include <math.h>` im Quelltext, beim `gcc`-Aufruf `-lm` mit angeben.
- $Q = C \cdot U, \quad U = R \cdot I, \quad I = \frac{dQ}{dt}$

*Viel Erfolg!*