

**Aufgabe 1: (12 Punkte)**

Bei den Multiple-Choice-Fragen ist jeweils nur **eine** richtige Antwort eindeutig anzukreuzen. Auf die richtige Antwort gibt es die angegebene Punktzahl.

Wollen Sie eine Multiple-Choice-Antwort korrigieren, kreisen Sie bitte die falsche Antwort ein und kreuzen die richtige an.

Lesen Sie die Frage genau, bevor Sie antworten.

- a) Welche der folgenden Aussagen über den C-Präprozessor ist **richtig**? 2 Punkte
- Der Präprozessor ist eine Softwarekomponente, welche Java-Klassen durch C-Funktionen ersetzt, die dann von einem C-Compiler übersetzt werden.
  - Der Präprozessor optimiert Makros durch Zeigerarithmetik.
  - Nach dem Übersetzen und dem Binden müssen C-Programme durch den Präprozessor nachbearbeitet werden, um Makros aufzulösen.
  - Die Syntax von Präprozessoranweisungen ist unabhängig vom Rest der Sprache C.
- b) Welche der folgenden Aussagen zur Sichtbarkeit von Variablen ist **richtig**? 2 Punkte
- Globale static-Variablen sind in allen Programmteilen immer direkt zugreifbar.
  - Eine lokale static-Variable ist nur innerhalb der Funktion, in der sie definiert wurde, sichtbar.
  - Lokale static-Variablen sind aus anderen Modulen nur dann zugreifbar, wenn sie ausserdem als "extern" deklariert wurden.
  - Wenn eine globale und eine lokale Variable gleichen Namens existieren, dann hat die globale Variable Vorrang.
- c) Was ist ein Stack-Frame? 2 Punkte
- Der Speicherbereich, in dem der Programmcode einer Funktion abgelegt ist.
  - Ein spezieller Registersatz des Prozessors zur Bearbeitung von Funktionen.
  - Ein Fehler, der bei unberechtigten Zugriffen auf den Stack-Speicher entsteht.
  - Ein Bereich des Speichers, in dem u.a. lokale automatic-Variablen einer Funktion abgelegt sind.

d) Welche Aussage zu Zeigern ist **richtig**?

2 Punkte

- Beim Rechnen mit Zeigern wird immer der Typ des Zeigers beachtet.
- Die Speicherstelle, auf die ein Zeiger verweist, kann niemals selbst einen Zeiger enthalten.
- Ein Zeiger kann zur Manipulation von schreibgeschützten Datenbereichen verwendet werden.
- Zeiger vom Typ `void*` benötigen weniger Speicher als andere Zeiger, da bei anderen Zeigertypen zusätzlich die Größe gespeichert werden muss.

e) Was versteht man unter Polling?

2 Punkte

- Ein Konzept zur Abarbeitung von Interrupts.
- Das regelmäßige Anheben eines Pegels, um einem Gerät einen bestimmten Zustand zu signalisieren.
- Wenn ein Programm regelmäßig eine Peripherie-Schnittstelle überprüft, ob Daten oder Zustandsänderungen vorliegen.
- Wenn ein Programm zum Zugriff auf kritische Daten Interrupts sperrt.

f) In Betriebssystemen wie Linux oder Windows unterscheidet man die Begriffe Programm und Prozess. Welche Aussage ist **richtig**?

2 Punkte

- Programme sind Anwendungen des Benutzers, während Prozesse Aktivitäten des Betriebssystems sind.
- Ein Prozess hat einen eigenen virtuellen Adressraum. Daten des Prozesses sind vor direktem Zugriff durch andere Prozesse geschützt.
- Programme sind C-Quellcode-Dateien, die durch einen C-Compiler in einen lauffähigen Prozess übersetzt werden können.
- Ein Programm ist ein Prozess in Ausführung.

## Aufgabe 2: Tankkontrolle (30 Punkte)

Schreiben Sie ein AVR-Mikrocontroller-Programm, das den Füllstand eines Tanks mit einem Fassungsvermögen von 50000 Litern überwacht und insbesondere einen Überlauf verhindert.

Befüllung: Der Zufluss ist mit einer Zuflusssperre und einem Durchflusssensor ausgestattet. Der Sensor signalisiert jeden durchgeflossenen Liter durch einen Interrupt. Der Zufluss kann durch Aktivierung der Sperre gestoppt werden.

Entnahme: Der Abfluss ist durch einen manuellen Absperrhahn geregelt. Öffnen und Schließen des Hahns werden durch Interrupt an den Controller gemeldet. Ein Durchflusssensor signalisiert jeden abgeflossenen Liter durch Zustandswechsel  $0 \leftrightarrow 1$  (Pollen!).

Das Programm soll im Einzelnen wie folgt funktionieren:

- Zum Programmstart ist der Tank leer und der Abfluss ist gesperrt.
- Die main-Funktion ruft zunächst die Funktion `void init()`; auf, welche die Initialisierung der I/O-Ports und Interruptquellen durchführt. Hierbei dürfen keine Annahmen über den initialen Zustand der Register gemacht werden. Am Ende der Initialisierung wird der Zufluss geöffnet.
- Stellt die Steuerung fest, dass der Tank voll ist, wird die Zuflusssperre aktiviert. Sinkt die Tankfüllung unter 20000 Liter, wird wieder aufgefüllt.
- Während eines Abflusses überwacht die Steuerung die abfließende Wassermenge durch Pollen des Abflusssensors.
- Die Steuerung soll den Prozessor in den Standardstromsparmodus versetzen, wenn gerade nichts zu tun ist.

### Information über die Hardware

Zuflusssensor: **PORTD, Pin 2**

- steigende Flanke pro einfließendem Liter Wasser
- externe Interruptquelle **0**, ISR-Vektor-Makro: **INT0\_vect**
- Aktivierung der Interruptquelle erfolgt durch Setzen des **INT0**-Bits im Register **GICR**

Zuflusssperre: **PORTD, Pin 0**

- 1 = offen, 0 = gesperrt

Abflusshahn: **PORTD, Pin 3**

- 1 = offen, 0 = geschlossen (=> steigende Flanke = öffnen, fallende = schließen)
- externe Interruptquelle **1**, ISR-Vektor-Makro: **INT1\_vect**
- Aktivierung der Interruptquelle erfolgt durch Setzen des **INT1**-Bits im Register **GICR**

Abflusssensor: **PORTD, Pin 1**

- Zustandswechsel = 1 Liter Durchfluss
- Zustand kann durch Lesen von **Bit 1** des Registers **PIND** abgefragt werden

Eingänge, Ausgänge, Externe Interruptquellen **0** bzw. **1** konfigurieren:

- Pin als Eingang konfigurieren: entsprechendes Bit in **DDRD**-Reg. auf 0
- Pin als Ausgang konfigurieren: entsprechendes Bit in **DDRD**-Reg. auf 1
- angeschlossene Sensoren verbinden den Pin mit Masse, es muss der interne Pullup-Widerstand verwendet werden (entspr. Bit in **PORTD**-Reg. auf 1 setzen).
- Konfiguration der externen Interruptquellen (Bits in Register **MCUCR**):

externe Int.quelle 0		externe Int.quelle 1		Beschreibung
ISC01	ISC00	ISC11	ISC10	
0	0	0	0	Interrupt bei low Pegel
0	1	0	1	Interrupt bei beliebiger Flanke
1	0	1	0	Interrupt bei fallender Flanke
1	1	1	1	Interrupt bei steigender Flanke











- c) Skizzieren Sie die Schritte, die durchzuführen sind, um aus einem C-Quellprogramm ein lauffähiges Programm auf einem Mikrocontroller zu erhalten.  
(6 Punkte)



-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----

-----