

Name:	Matr.-Nr.:
<i>FH Dortmund</i>	<i>FB Informations- und Elektrotechnik</i>

Hard- und Software-Engineering (Bachelor) HS2

Klausur vom 4. 4. 2008

1. Skizzieren Sie eine Konfiguration aus zwei Mikrocontrollern, die über serielle Schnittstellen miteinander verbunden sind (Abb. 1). (8 Punkte)

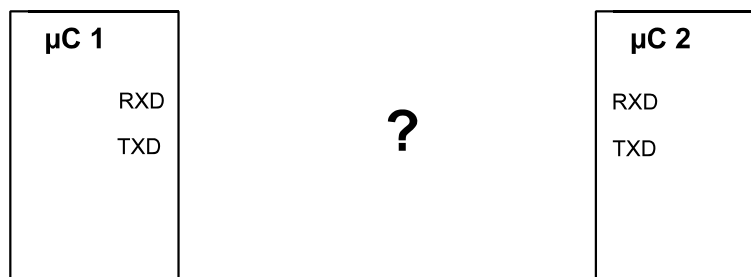


Abb. 1

2. Welche Vorteile hat die serielle Schnittstelle beim Aufbau von Mehrprozessorkonfigurationen im Vergleich zu anderen typischen Alternativen, z. B. SPI oder I²C? (12 Punkte)
3. Skizzieren Sie den Aufbau eines Taskzustandsbereichs (TSA) für Atmel- AVR-Mikrocontroller. (10 Punkte)
4. Erläutern Sie kurz den Fachbegriff *preemptives Multitasking*. Welche Vorkehrungen muß die Hardware aufweisen, damit dieses Prinzip verwirklicht werden kann? (8 Punkte)
5. Erläutern Sie kurz, wozu der Frame Pointer (Base Pointer) dient. (8 Punkte)
6. Es geht um Zeitstufen in Mikrocontrollern. Erläutern Sie kurz, wodurch sich die Betriebsarten "Zähler" und "Zeitgeber" voneinander unterscheiden. (10 Punkte)

Programmieraufgaben (Assemblerprogrammierung Atmel AVR)

7. Eine Interruptserviceroutine benutzt die Register R0, R1, R20, R24, R25. Geben Sie die Befehlsfolgen des Eintritts und der Rückkehr an. (12 Punkte)
8. Die ISR von Aufgabe 7 soll ihrerseits unterbrechbar sein. Geben Sie die entsprechenden Änderungen bzw. Ergänzungen an. (10 Punkte)

9. Ein LCD-Display soll für zwei Anzeigeaufgaben ausgenutzt werden (Abb. 2). Die ersten zwei Zeilen werden von einer im Grundzustand laufenden Task genutzt, die verbleibenden zwei Zeilen von einer Interruptserviceroutine (ISR). Es stehen folgende Unterprogramme zur Verfügung:

- DAT_OUT überträgt ein darzustellendes Zeichen,
- CTL_OUT überträgt ein Kommando.

Da es im Grunde gleichgültig ist, wie die Parameter übergeben werden, verwenden wir folgende Notation:

- DAT_OUT Datenbyte,
- CTL_OUT Kommandobyte.

Geben Sie für beide Fälle (Grundzustand und ISR) an, wie jeweils ein darzustellendes Zeichen übertragen wird (ggf. Mischung aus Kurzbeschreibung und AVR-Assemblercode). Fiktives Beispiel:

```
LDI          irgendwas
DAT_OUT     Datenbyte
MOV         irgendwas irgendwo hin
```

Es geht jeweils nur um die typischen Abläufe, nicht um das Retten und Wieder-Einstellen. Alternativ oder ergänzend dazu können Sie auch beschreiben, was passieren kann und worauf besonders zu achten ist.

(12 Punkte)

Zeile 1	– Anzeige Grundzustand –
Zeile 2	– Anzeige Grundzustand –
Zeile 3	– Anzeige ISR –
Zeile 4	– Anzeige ISR –

Abb. 2

10. Es ist eine ISR zu programmieren, die eine bestimmte Anzahl von Impulsen ausgibt. Der Anzahl-Wert befindet sich im SRAM an der symbolischen Adresse P_CNT. Die ISR wird von der Zeitgeberhardware ausgelöst. Je Auslösung ist ein Impuls auszusenden. Verwenden Sie hierfür das fiktive Unterprogramm P_EMIT (das keine Register benötigt). Ist der Anzahl-Wert gleich Null, so soll kein Impuls abgegeben werden. Sie dürfen beliebige Register verwenden.

(15 Punkte)

Zusatzaufgaben

- Z1. Erläutern Sie kurz wenigstens eine Möglichkeit, einen Stacküberlauf während des Betriebs zu erkennen.

(10 Punkte)

- Z2. Es sind mehrere Funktionen nahezu gleichzeitig auszuführen. Die naheliegenden Lösungen: (1) für jede Funktion ein eigener Prozessor (Mehrprozessorsystem), (2) ein einziger – hinreichend leistungsfähiger – Prozessor und Multitasking. Diskutieren Sie kurz das Für und Wider beider Ansätze. Wofür würden Sie sich entscheiden?

(10 Punkte)

Klausur vom 27. 3. 2009

1. Wozu dient ein Taskzustandsbereich? Wie könnte ein Taskzustandsbereich für Atmel- AVR-Mikrocontroller aussehen? (Bitte so darstellen, daß der Aufbau bis aufs Byte genau ersichtlich wird. Für Mehrfachangaben genügt eine Kurzdarstellung (z. B. "22 Bytes für den Stromsparzustand").)
(10 Punkte)
2. Der folgende Quelltext ist ein Anwendungsprogramm, das die Funktion eines elektronischen Würfels darstellt (Zählen modulo 6, solange eine Taste betätigt wird; vgl. Praktikumsversuch 3). Dieses Programm ist zeitmultiplex mehrfach auszuführen. Hierzu soll das Prinzip des kooperativen Multitasking verwendet werden. Die entsprechende Systemfunktion heißt BREAK. Sie ist als Unterprogramm aufzurufen.
 - a) Geben Sie an, an welcher Stelle / an welchen Stellen solche Unterprogrammaufrufe einzuschieben sind.
 - b) Was muß eine solche Systemfunktion leisten? Es ist nicht nötig, alles auszuprogrammieren. Die Funktion sollte aber umgangssprachlich oder anhand eines Flußdiagramms hinreichend genau beschrieben werden.

(20 Punkte)

wue:

```
lds y1,current_task
lds yh,current_task+1
ldd z1,y+0           ; Virtuelle Portadresse
ldi zh,0
ldi einer,1
```

wdisplay21:

```
mov temp,einer
rcall convseg       ; Siebensegmentanzeige
std z+2,r0
```

wwarten2:

```
           ; Tastenabfrage
ldd r16,z+0
andi r16,0x80
brne wwarten2
```

```
inc einer
cpi einer,7
brne wdisplay21
rjmp wue
```

3. Mehrere parallel laufende Anwendungen sollen Daten auf einen Protokolldrucker ausgeben. Das eigentliche Druckkommando kann jeweils nur ein einziges Zeichen übertragen. Der Protokolldruck wird jeweils durch eine Druckroutine *PRNT_LOG* (*Daten- und Formatparameter*) erledigt, das seinerseits Druckkommandos *PRNT* (*Zeichencode*) ausführt. Die Grundstruktur:

PRNT_LOG:

```
....
....
```

PRNT (char)

```
....
```

PRNT (char)

```
.... usw.
```

READY:

```
RET
```

- a) Kann man die Druckroutine einfach mit *CALL PRNT_LOG (Parameter)* aufrufen? Worauf wäre ggf. zu achten?

Geben Sie nachfolgend an, wie *PRNT_LOG* in der einzelnen Anwendung aufgerufen werden könnte (verbale Beschreibung und/oder Pseudo-Code und/oder Flußdiagramm).

- b) Wenn alle entsprechenden Programme auf einem einzigen Prozessor laufen (zeitgesteuert erzwungenes Multitasking).
- c) Wenn jedes Programm auf einem anderen Prozessor läuft (wobei gemeinsame Speicherbereiche vorgesehen sind (Shared Memory)).

(24 Punkte)

4. Abb. 1 zeigt eine Konfiguration aus drei Mikrocontrollern, die über die SPI- Schnittstelle miteinander zu verbinden sind. Neben den SPI-Signalen sind jeweils vier universelle E-A-Signale nutzbar (die aber nicht unbedingt verbraucht werden müssen ...).

- a) Wie sind die Mikrocontroller zu verschalten, wenn μC 1 der SPI-Master sein soll? (Einzeichnen.)
- b) Wie stellen Sie es an, daß die Slaves von sich aus Anforderungen an den Master richten können? (Signale einzeichnen und Funktionsweise kurz erläutern.)

(20 Punkte)

5. Erläutern Sie kurz, wie man mehrere (mehr als zwei) Mikrocontroller über ihre seriellen Schnittstellen zu einem Multiprozessorsystem zusammenschalten kann (Skizze + grundsätzliche Funktionserläuterung).

(20 Punkte)

6. Erläutern Sie kurz, wie die Watchdog-Vorkehrungen in Mikrocontrollern funktionieren. (Wie sie wirken, wozu sie gut sind und was beim Programmieren zu beachten ist.)

(12 Punkte)

7. Erläutern Sie kurz, wie die Capture-Funktion einer Zähler/Zeitgeber-Einheit funktioniert und wozu man sie anwenden kann (ein Beispiel genügt).

(10 Punkte)

Zusatzaufgaben

- Z1. Welche peripheren Einrichtungen sind in übliche Mikrocontroller eingebaut? (Aufzählung genügt.)

(5 Punkte)

- Z2. Es ist eine ISR zu programmieren (für Atmel AVR), die jeweils ein Datenbyte ausgibt. Sie nutzt die Register R5 bis R7 und R24 bis R27. Um das Datenbyte auszugeben, rufen Sie das fiktive Unterprogramm *P_EMIT* (das keine Register benötigt).

(10 Punkte)

- Z3. Die ISR von Aufgabe Z2 soll ihrerseits unterbrechbar sein. Geben Sie die entsprechenden Änderungen bzw. Ergänzungen an.

(5 Punkte)

- Z4. Es ist offensichtlich sinnvoll, Programme reentrant zu schreiben, d. h. so, daß es nur eine Objektdatei gibt, aber mehrere Instanzen des Programms – in mehreren Tasks – gleichzeitig laufen können. Wie steht es jedoch mit der Ein- und Ausgabe? Die eine Task nutzt beispielsweise Port A, eine andere Port C usw. Erläutern Sie kurz, wie Sie dieses Problem beim Atmel AVR erledigen können.

(8 Punkte)

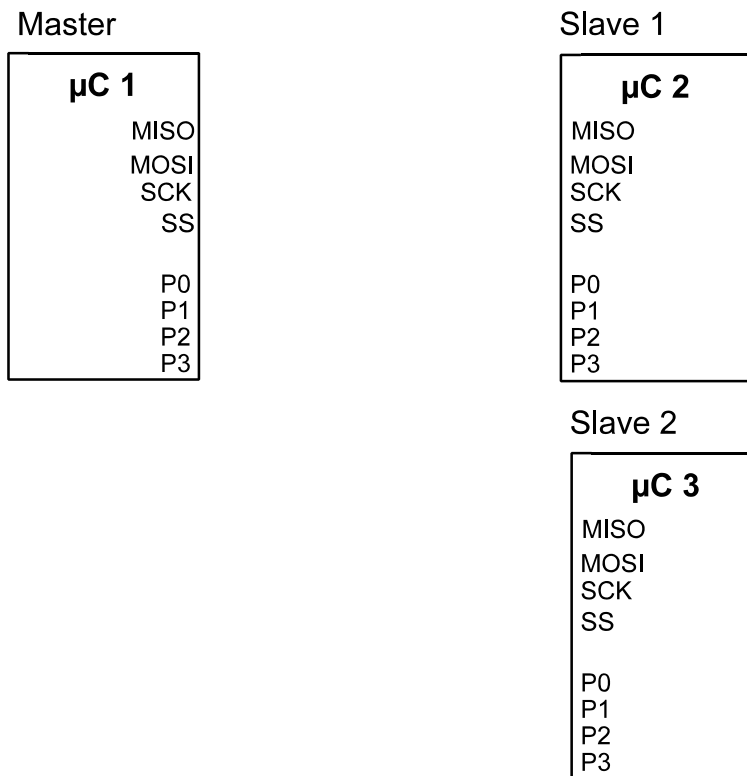
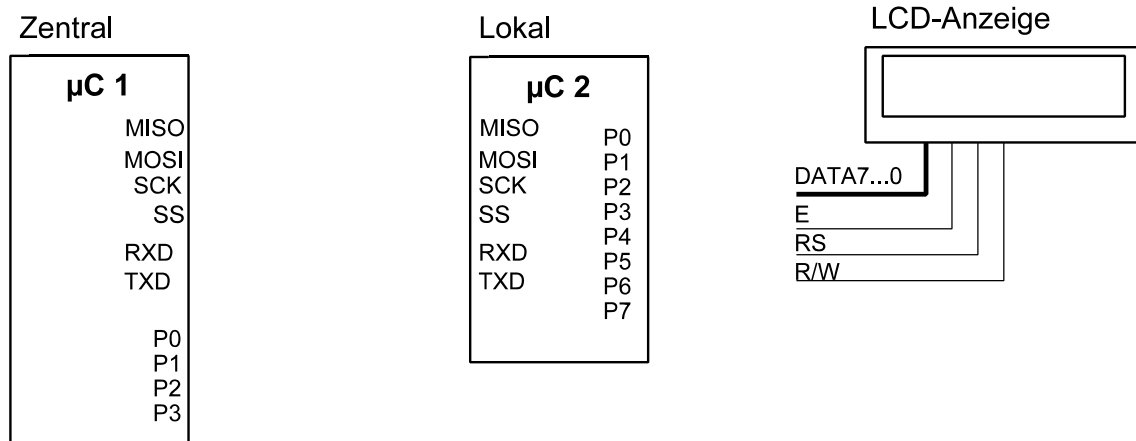


Abb. 1

Klausur vom 30. 9. 2009

1. Es sind mehrere Funktionen gleichzeitig auszuführen. Die naheliegenden Lösungen: (1) für jede Funktion ein eigener Prozessor (Mehrprozessorsystem), (2) ein einziger – hinreichend leistungsfähiger – Prozessor und Multitasking. Diskutieren Sie kurz das Für und Wider beider Ansätze. Wovon hängt die Wahl im wesentlichen ab?
(10 Punkte)
2. An einer Maschine sind an verschiedenen Stellen alphanumerische LCD-Displays (wie im Praktikum verwendet) anzubringen. Sie sind von einem zentralen AVR-Mikrocontroller aus anzusteuern. Dabei können die Leitungslängen bis zu 10 m betragen. Es ist zulässig, jedem LCD-Display einen weiteren (lokalen) AVR-Mikrocontroller vorzuschalten.
 - a) Wir betrachten zunächst ein einziges LCD-Display (Abb. 1). Schließen Sie die LCD-Einheit so an den lokalen Mikrocontroller an, daß Sie mit den verfügbaren acht Bitpositionen P7...P0 auskommen.
 - b) Welche Schnittstelle verwenden Sie, um den lokalen Mikrocontroller an den zentralen anzuschließen? Weshalb? Zeichnen Sie die jeweiligen Signalwege ein (Direktverbindungen unter Vernachlässigung von Pegelwandlungen usw.). Denken Sie hierbei nicht nur an die Ausgabe, sondern auch an einen Rückweg.
 - c) Wie lösen Sie das Problem, mehrere LCD-Displays an den zentralen Mikrocontroller anzuschließen? Hierfür stehen nur die Schnittstellen von Abb. 1 zur Verfügung. Sie dürfen aber zusätzliche Schaltmittel einsetzen. (Nur die Grundsatzlösung zeichnerisch darstellen und erläutern. Entwerfen auf Gatterebene nicht erforderlich.)
(25 Punkte)

**Abb. 1**

3. Erläutern Sie kurz den Fachbegriff *kooperatives Multitasking*. Welche Vorteile hat dieses Prinzip?
(10 Punkte)
4. Es geht um Zeitstufen (Counter/Timer Units) in Mikrocontrollern.
 - a) Erläutern Sie kurz, wie eine solche Zeitstufe im PWM-Betrieb arbeitet.
 - b) Erläutern Sie kurz, wie die Capture-Funktion abläuft und wozu man sie anwenden kann (ein Beispiel genügt).

(16 Punkte)

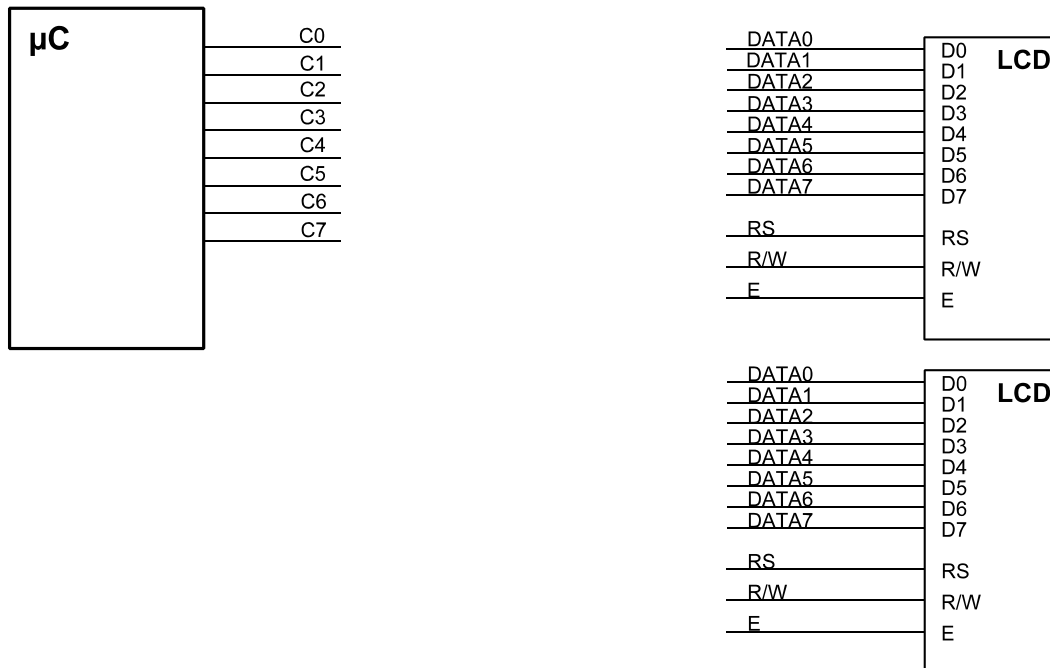
5. Es sind binäre Sensorsignale auszuwerten (also nur Aus-Ein, keine Signalwandlung). Hierfür gibt es zwei grundsätzliche Möglichkeiten: das Abfrageprinzip und das Unterbrechungsprinzip. Welches Prinzip ist unter welchen Einsatzbedingungen "schneller", verspricht also die kürzeren Latenzzeiten?
(10 Punkte)
6. Was sind einfache (lightweigh) Interrupts? (Kurze Erläuterung.)
(10 Punkte)
7. In den Werbetexten der Prozessoranbieter wird oftmals die Anzahl der Universalregister herausgestellt (RISC-Prinzipien). So wird u. a. stolz verkündet, daß der AVR 32 Register aufweist (und demzufolge besser sein muß als eine Maschine mit weniger Registern). Andererseits werden immer noch Prozessoren verkauft, die vergleichsweise wenige Register aufweisen (die Java Virtual Machine (JVM) hat gar keine). Erläutern Sie kurz, ob für Echtzeit-Multitasking-Anwendungen viele Register wirklich so ein Segen sind, wie es die Werbung behauptet.
(10 Punkte)
8. Erläutern Sie kurz, wie die Watchdog-Vorkehrungen in Mikrocontrollern funktionieren. (Wie sie wirken, wozu sie gut sind und was beim Programmieren zu beachten ist.)
(10 Punkte)

Zusatzaufgaben

- Z1. Welche periphere Einrichtungen sind in übliche Mikrocontroller eingebaut? (Aufzählung genügt.)
(5 Punkte)
- Z2. Es ist eine ISR zu programmieren (für Atmel AVR), die jeweils ein Datenbyte ausgibt. Sie nutzt die Register R20 bis R27. Um das Datenbyte auszugeben, rufen Sie das fiktive Unterprogramm P_EMIT (es erledigt die eigentliche Arbeit und braucht keine Register).
(10 Punkte)
- Z3. Die ISR von Aufgabe Z2 soll ihrerseits unterbrechbar sein. Geben Sie die entsprechenden Änderungen bzw. Ergänzungen an.
(5 Punkte)
- Z4. Erläutern Sie kurz das Prinzip der Ereignissteuerung.
(10 Punkte)

Klausur vom 26. 3. 2010

1. Was soll ein virtueller Speicher leisten? Erläutern Sie kurz zwei grundsätzliche Verfahren, mit denen man einen virtuellen Speicher implementieren kann.
(10 Punkte)
2. Erläutern Sie kurz, wie ein Stack Frame aufgebaut ist und wie er adressiert wird. Mit welcher Adressierungsweise kommt man an die übergebenen Parameter heran, mit welcher an die lokalen Variablen?
(10 Punkte)
3. Was sind einfache (lightweigh) Interrupts? (Kurze Erläuterung.)
(8 Punkte)
4. Es geht um Zeitstufen in Mikrocontrollern. Erläutern Sie kurz, wodurch sich die Betriebsarten "Zähler" und "Zeitgeber" voneinander unterscheiden.
(6 Punkte)
5. Es ist eine ISR zu programmieren (für Atmel AVR), die jeweils ein Datenbyte ausgibt. Sie nutzt die Register R20 bis R25. Um das Datenbyte auszugeben, rufen Sie das fiktive Unterprogramm PRNT (es erledigt die eigentliche Arbeit und braucht keine Register).
(10 Punkte)
6. Die ISR von Aufgabe 5 soll ihrerseits unterbrechbar sein. Geben Sie die entsprechenden Änderungen bzw. Ergänzungen an.
(10 Punkte)
7. Erläutern Sie kurz, wie man mehrere (mehr als zwei) Mikrocontroller über ihre seriellen Schnittstellen zu einem Multiprozessorsystem zusammenschalten kann (Skizze + grundsätzliche Funktionserläuterung). Es genügt eine Lösungsvariante. Dabei ist es auch zulässig, weitere E-A-Anschlüsse auszunutzen und zusätzliche Schaltmittel vorzusehen (es sollten aber jeweils nicht allzu viele sein...)
(10 Punkte)
8. Wir haben ein erzwungenes Multitasking mit fester Zeiteilung ausprobiert. Wie würden Sie das Problem lösen, den einzelnen Tasks eine jeweils unterschiedlich lange Zeitscheibe zuzuteilen? Tip: Denken Sie daran, daß jede Task einen eigenen Taskzustandsbereich hat.
(10 Punkte)
9. Erläutern Sie kurz, wie die Watchdog-Vorkehrungen in Mikrocontrollern funktionieren. (Wie sie wirken und wozu sie gut sind.)
(10 Punkte)
10. Eine Bedientafel soll zwei Dotmatrix-LCDs erhalten. Zur Ansteuerung steht jedoch nur ein 8-Bit-Port zur Verfügung. Lassen Sie sich was einfallen ...
(15 Punkte)



11. Wir bleiben bei den beiden LCDs. Die eine Anzeige soll von einem im Grundzustand laufenden Programm genutzt werden, die andere von einer Interruptserviceroutine. Kann in diesem Zusammenhang etwas Grundsätzliches schiefgehen? Wenn ja: (1) was könnte passieren, (2) was könnte man tun, um es zu verhindern? (12 Punkte)
12. Windows CE (und dessen Nachfolger) stellen gar keine so hohen Anforderungen an die Hardware. Was sie aber fordern, ist ein vom Programmablauf unabhängiger Zeitgeber, der einen Interrupt auslösen kann. Frage: wozu? (8 Punkte)

Zusatzaufgaben

- Z1. Erläutern Sie kurz wenigstens eine Möglichkeit, einen Stacküberlauf während des Betriebs zu erkennen. (10 Punkte)
- Z2. Ein Programmierer kommt auf die Idee, sich die Nutzung der Watchdog-Vorkehrungen zu vereinfachen. Alle fünf Befehle einen Watchdog-Befehl einfügen – und schon läuft alles bestens. Ist diese Lösung aber wirklich so großartig? (Diskutieren Sie kurz das Problem.) (10 Punkte)
- Z3. Es geht darum, auf Eingangssignale zu reagieren. Unter welchen Bedingungen ist eine Abfragegeschleife schneller als eine von diesen Signalen ausgelöste Unterbrechung? (10 Punkte)