



**Prüfung:** μ-Computertechnik  
**Termin:** Dienstag, 06.07.2004 ; 13:30-15:30  
**Prüfer:** Prof. Walter  
**Hilfsmittel:** beliebig

<b>Name:</b>	_____
<b>Vorname:</b>	_____
<b>Studiengang:</b>	_____
<b>Projekt:</b>	_____
<b>URL:</b>	_____

Bitte überprüfen Sie, ob alle Protokolle der Projekte in den Webseiten sind, inkl. Protokoll der Projektpräsentation  
bitte keine rote Farbe verwenden

(nicht ausfüllen)!

<b>Aufgabe</b>	<b>mögl. Punkte</b>	<b>erreichte Punkte</b>
<b>1</b>	<b>14</b>	
<b>2</b>	<b>14</b>	
<b>3</b>	<b>6</b>	
<b>4</b>	<b>8</b>	
<b>5</b>	<b>8</b>	
<b>Gesamt</b>	<b>50</b>	
	<b>Note</b>	

**Bearbeiten Sie die Aufgaben nur, falls Sie keine gesundheitlichen Beschwerden haben.**

**Viel Erfolg!**

**Bemerkungen:**



**Hinweis:**

Mit dem Programm „BZ“ Binaerzaehler veranschaulichen die drei LEDs der EURO\_535 einen Binärzähler. Der Zähler zählt jeweils ab 0 bis 7. Es wird jede Sekunde ein neuer Wert angezeigt.

Zur Vereinfachung werden zunächst zwei Teilprogramme programmiert:

- BZ\_ZEIT      Binaerzaehler Zeit – erzeugt den 1-Sekunden Takt.
- BZ\_ANZ      Binaerzaehler Anzeige – steuert die LEDs an

**1. 8051/80535 Programm „BZ\_ZEIT“ (14 Punkte)    Σ\_\_\_\_\_**

Entwickeln Sie ein 80535-Programm (8051-Assembler) welches mit Hilfe von TIMER 2 einen 1 Sekundentakt erzeugt. Die Periodendauer der PWM ist 50ms. Mit Taste T1 wird das Programm gestartet und mit T2 beendet. Die Interrupt-Service-Routine für Timer 2 „TIMER2\_ISR“ schaltet jeweils nach genau einer Sekunde 3 LEDs ein oder aus.

- a. Mit welcher Zahl müssen Sie die Reload-Register laden?
- b. Führen Sie die wesentlichen Schritte für das Programm auf.
- c. Schreiben Sie das Programm in 8051-Assembler.

**Lösung a:**

Mit der Zahl 0B03CH = 15536

**Lösung b:**

Interruptvektoren:  
Timer 2

Initialisierung:  
Interruptsteuerung  
Timer 2  
Register

Programmschleife:  
Abfrage Taster

InterruptService\_T2:  
1s = 20 mal zählen  
Leuchten an

Ende

```
$nomod51                                   ; der Modus fuer 8051 wird abgeschaltet
$debug
$nolist                                   ; es wird kein Listing fuer reg535 erstellt
#include(reg535.pdf)                   ; die 80535-spezifischen Daten
$list                                   ; es wird ein Listing erstellt
$ title (BZ_ZEIT.A51)
;-----
```



```
; Programmbeschreibung
;-----
;
; Programm:      Blinkprogramm nach 1s Umschaltung
; Interrupt mit Timer2 und 50ms CRC
; Betaetigung Taster 1: Programm startet
; Betaetigung Taster 2: Ende
;
;
; Programmiert:  J. Walter
;
; Verwendete Einspruege: keine
;
; Verwendete Unterprogramme: keine
;
;
; Verwendete Register und Variable:
; Registerbank(0)
; R3           Zaehlregister für Zeit
; R2           Zaehlregister für Ausgabe
; Kommentar:
; Aenderungen: Anpassung auf KEIL
; Datum: 3.1.2004
;
;-----
; Initialisierungsteil für allgemeine Konstanten
;-----
V_BLINK EQU 20           ;Blinkfrequenz 20 Dezimal
PERIODE_L EQU 0B0h      ;Low Byte von Compare Reload Capture
PERIODE_H EQU 3Ch       ;High Byte von Compare Reload Capture

CSEG AT 0H              ;Legt absolute Codesegmentadresse auf 0h
jmp INIT
;
;-----
; Interrupt-Vektoren
;-----
org 2Bh                 ;Sprungmarke zur ISR fuer Timer0
TIME_INT_2:             ;ISR fuer Timer0 (Adresse 0Bh)
clr TF2
call TIMER2_ISR         ;Aufruf: Interrupt Service Routine
reti

;-----
; Initialisierungsteil für On-Chip Peripherie
;-----
ORG 100H                ;Programmstart bei 100H
INIT:
setb P3.2               ;Lampen ausschalten
setb P3.3
setb P3.4
mov R3,#V_BLINK         ;R3 Ist Zaehlvariable 20*50ms
setb EAL                ;Generelle Interrupt Freigabe
setb ET2                ;Timer 2 Interruptsfreigabe
mov T2CON,#00010000B    ;Timer2 Einstellung:Timer mit f=f(osz)/12
```



```
                                ;Mode 0 ->Reload bei Ueberlauf
mov CRCL,#PERIODE_L           ;Low Byte von Compare Reload Capture
mov CRCH,#PERIODE_H           ;High Byte von Compare Reload Capture
mov TL2,#PERIODE_L           ;Timer 2 laden
mov TH2,#PERIODE_H

;-----
;Programmschleife
;-----

ABFRAGE:
jnb P1.1,BLINKEN
jnb P1.2,ENDE
ajmp ABFRAGE
;-----
;Teilprogramme
;-----

BLINKEN:
setb T2IO                      ;Timer2 zaehlt
ajmp ABFRAGE                   ;Springe zur Adresse ABFRAGE

ENDE:
clr T2R0                       ;Timer0 ausschalten
setb P3.2                      ;Lichter ausschalten
setb P3.3
setb P3.4
ajmp ABFRAGE                   ;Springe zur Adresse ABFRAGE

TIMER2_ISR:
djnz R3,SPRUNG                 ;Sprung falls R2 noch nicht 0
cpl P3.2                       ;komplementiere P3.2
cpl P3.3
cpl P3.4
mov R3,#V_BLINK                ;Register 3 erhaelt wieder den Wert V_BLINK

SPRUNG:
ret

end
```



**2. 8051/80535-Programm „BZ\_ANZ“ (14 Punkte)    Σ\_\_\_\_\_**

Schreiben Sie ein 8051-Programm welches die letzten drei Bits von Register 2 - das kontinuierlich inkrementiert wird – an den LEDs ausgibt.

Die LEDs der EURO\_535 werden den Bits von Register R2 zugeordnet:

L1=R2 – Bit 0

L2=R2 – Bit 1

L3=R2 – Bit 2

Hierzu wird der Inhalt von R2 in den Akkumulator geschoben und invertiert. Danach erfolgt die Zuordnung von Bit 0 an P3.2, das Bit 1 an P3.3, das Bit 2 an P3.4.

Hilfe: Rotieren Sie den Akku über das Carry-Bit und geben Sie das Carry an dem zugehörigen Port aus.

Der Start des Programmes erfolgt mit T1.

- a. Welche Helligkeit an den LEDs erwarten Sie?
- b. Führen Sie die wesentlichen Schritte für das Programm auf.
- c. Schreiben Sie das Programm in 8051 Assembler.

**Lösung a:**

Halbe Helligkeit – jede Leuchtdiode wird eingeschaltet und ausgeschaltet

**Lösung b:**

Initialisierung:

R2

ABFRAGESCHLEIFE:

ANZEIGE:

R2 in Akku

Akku invertieren

Ausgeben

end

```
$nomod51           ;der Modus fuer 8051 wird abgeschaltet
$debug
$nolist            ;es wird kein Listing fuer reg535 erstellt
$include(reg535.pdf) ;die 80535-spezifischen Daten
$list             ;es wird ein Listing erstellt
$title (BZ_ANZ.A51)
;-----
;Programmbeschreibung
;-----
;Programm: Binaerzahl Anzeige
;T1 startet
;R2 wird fortlaufend inkrementiert und Bit 0,1,2 an LEDs 3.2, 3.3, 3.4
;ausgegeben
;Datum: 04.07.2004
```



```
;Programmiert: J. Walter
;
;Verwendete Einspruenge: keine
;
;Verwendete Unterprogramme: keine
;
;
;Verwendete Register und Variable:
;Registerbank(0)
;R2
;
;Kommentar:
;
;Aenderungen:
;Datum:
;
;
;-----
;Initialisierungsteil für allgemeine Konstanten
;-----

CSEG AT 0H          ;Legt absolute Codesegmentadresse auf 0h
jmp INIT
;
;-----
;Interrupt-Vektoren
;-----

;-----
;Initialisierungsteil für On-Chip Peripherie
;-----

ORG 100H           ;Programmstart bei 100H
INIT:
mov R2,#0         ;R2 wird initialisiert

;-----
;Programmschleife
;-----

ABFRAGE:
jnb P1.1,ANZEIGE ;Start mit T1
jmp ABFRAGE

ANZEIGE:
inc R2            ;R2 inkrementieren
mov A,R2         ;In Akku schieben
cpl A            ;invertieren
rrc A            ;über Carry rotieren
mov P3.2,C       ;Bit 0 auf P3.2 kopieren
rrc A
mov P3.3,C       ;Bit 1 auf P3.3 kopieren
rrc A
mov P3.4,C       ;Bit 2 auf P3.4 kopieren
jmp ANZEIGE
end
```



**8051/80535-Programm „BZ“ (6 Punkte)     Σ\_\_**

Schreiben Sie das Programm „BZ“, indem Sie Aufgabe 1 und Aufgabe 2 kombinieren. Sie können die Programmteile mit Teil 1 und Teil 2 durch eine Klammer markieren und diese Teile in das Gesamtprogramm einfügen.

Lösung a:

**Einfache Lösung:**

Im Programm BZ-ZEIT wird in der Interrupt Service Routine Programm 2 aufgerufen. Das Register 2 wird mit 0 initialisiert.

Gesamtes Programm: (War nicht gefordert)

```
$nomod51           ;der Modus fuer 8051 wird abgeschaltet
$debug
$nolist            ;es wird kein Listing fuer reg535 erstellt
#include(reg535.pdf) ;die 80535-spezifischen Daten
$list             ;es wird ein Listing erstellt
$title (BZ.A51)
;-----
;Programmbeschreibung
;-----
;
;Programm: Zaehlprogramm + Blinkprogramm 2s Periode
;Interrupt mit Timer2 und 50ms CRC
;Betaetigung Taster 1: Programm startet
;Betaetigung Taster 2: Ende
;
;
;Programmiert: J. Walter
;
;Verwendete Einspruenge: keine
;
;Verwendete Unterprogramme: keine
;
;
;Verwendete Register und Variable:
;Registerbank(0)
;R3           Zaehlregister für Zeit 20*50ms
;R2           Zaehlregister für Anzeige
;Kommentar:
;Aenderungen: Anpassung auf KEIL
;Datum:3.1.2004
;
;
;-----
;Initialisierungsteil für allgemeine Konstanten
;-----
V_BLINK EQU 20           ;Blinkfrequenz 20 Dezimal
PERIODE_L EQU 0B0h      ;Low Byte von Compare Reload Capture
PERIODE_H EQU 3Ch       ;High Byte von Compare Reload Capture
;
CSEG AT 0H              ;Legt absolute Codesegmentadresse auf 0h
jmp INIT
;
;-----
```



```
;Interrupt-Vektoren
;-----
org 2Bh                ;Sprungmarke zur ISR fuer Timer0
TIME_INT_2:           ;ISR fuer Timer0 (Adresse 0Bh)
clr TF2
call TIMER2_ISR       ;Aufruf: UP Schleife
reti

;-----
;Initialisierungsteil für On-Chip Peripherie
;-----
ORG 100H               ;Programmstart bei 100H
INIT:
setb P3.2              ;Lampen ausschalten
setb P3.3
setb P3.4
mov R3,#V_BLINK        ;R3 Zaehlvariable fuer Interrupts
mov R2,#0              ;R2 Zaehlvariable fuer Anzeige
setb EAL               ;Generelle Interrupt Freigabe
setb ET2               ;Timer 2 Interruptsfreigabe
mov T2CON,#00010000B  ;Timer2 Einstellung:Timer mit f=f(osz)/12
                        ;Mode 0 ->Reload bei Ueberlauf
mov CRCL,#PERIODE_L   ;Low Byte von Compare Reload Capture
mov CRCH,#PERIODE_H   ;High Byte von Compare Reload Capture
mov TL2,#PERIODE_L    ;Timer 2 laden
mov TH2,#PERIODE_H

;-----
;Programmschleife
;-----
ABFRAGE:
jnb P1.1,BLINKEN
jnb P1.2,ENDE
ajmp ABFRAGE

BLINKEN:
setb T2I0              ;Timer2 zaehlt
ajmp ABFRAGE           ;Springe zur Adresse ABFRAGE

ENDE:
clr T2R0               ;Timer0 ausschalten
setb P3.2              ;Lichter ausschalten
setb P3.3
setb P3.4
ajmp ABFRAGE           ;Springe zur Adresse ABFRAGE

TIMER2_ISR:
djnz R3,SPRUNG        ;Sprung falls R3 noch nicht 0
ANZEIGE:
inc R2                 ;R2 inkrementieren
mov A,R2               ;In Akku schieben
cpl A                  ;invertieren
rrc A                  ;über Carry rotieren
mov P3.2,C             ;Bit 0 auf P3.2 kopieren
rrc A
mov P3.3,C             ;Bit 1 auf P3.3 kopieren
```





```
rrc A
mov P3.4,C           ;Bit 2 auf P3.4 kopieren

mov R3,#V_BLINK     ;Register 3 erhaelt wieder den Wert V_BLINK

SPRUNG:
ret

end
```



**Prozessorwahl (8 Punkte)**

Σ\_\_\_\_\_

Wählen Sie für die nachfolgenden Aufgaben den optimalen Prozessor / Controller (bzgl. Kosten - Programmieraufwand bei der angegebenen Stückzahl) aus. Bitte begründen Sie Ihre Lösung.

**a. Aufgabe: Balanceakt Prototyp**

Auf einer runden Fläche welche in zwei Richtungen x,y geneigt werden kann, soll ein Tischtennisball balanciert werden. Hierzu müssen zwei Servomotoren angesteuert und ein Kamerasignal über V24-Schnittstelle an die Hardware weitergegeben werden.

Bitte geben Sie die genaue Bezeichnung des einzusetzenden Prozessors / Controllers an?

**Prozessor / Controller: 80535**

**Begründung: 2 Servomotoren mit PWM, zweite V24 Schnittstelle per SW**

**b. Aufgabe: 5.1-Aufnahme**

Für eine 5.1 Aufnahme eines Konzertes werden gleichzeitig 7 Mikrofonsignale verstärkt und digitalisiert.

Bitte geben Sie die genaue Bezeichnung des einzusetzenden Prozessors / Controllers an?

**Prozessor / Controller: DSP 96000, SHARC**

**Begründung: Signalfrequenzen bis 20kHz**

**c. Aufgabe: U-Boot**

Über zwei Servomotoren wird ein Modellbau U-Boot gesteuert. Gleichzeitig wird der Salzgehalt des Wassers gemessen, da die Funkverbindung bei Salzwasser nicht funktioniert. Der Salzgehalt wird über einen Sensor mit Spannungsausgang gemessen. Sobald der Salzgehalt über ein bestimmtes Maß ansteigt, muss das U-Boot umgehend nach oben gesteuert werden. Auch muss nach spätestens 30 Minuten gewährleistet sein, dass das U-Boot wieder an die Wasseroberfläche kommt. Es kommen maximal 3 Systeme zum Einsatz.

Bitte geben Sie die genaue Bezeichnung des einzusetzenden Prozessors / Controllers an?

**Prozessor / Controller: 80535**

**Begründung: Servomotor + Watchdog + A/D-Wandler**

**d. Aufgabe: Chipkarte**

Ein Chipkartenleser gibt digitale Signale mit TTL-Pegel aus. Für 10000 Systeme soll ein geeignetes System Karteninformationen auslesen und über eine PIN-Nummer die Öffnung einer Türe durch ein Relais auslösen.

Bitte geben Sie die genaue Bezeichnung des einzusetzenden Prozessors / Controllers an?

**Prozessor / Controller: 8051**

**Begründung: keine Rechenleistung, günstig**



5. μ-Controller (8 Punkte, jeweils nur eine und beste Antwort) Σ\_\_\_\_\_

1. Welcher Prozessor / Controller enthält einen Timer 2?

a) 8080	
b) 80515	X
c) 68000	
d) ADSP 2101	

2. Warum wird für den Anschluß der EURO\_535-Karte an ein Notebook ein USB-Umsetzer benötigt?

a) Notebooks sind prinzipiell langsamer.	
b) Die V24-Schnittstelle an Notebooks hat nicht genügend Leistung.	
c) Es gibt meist keine RS232-Schnittstelle an Notebooks neuerer Bauart.	X
d) Damit die Kommunikation zwischen EURO_535 und Keil uVision schneller wird.	

3. Welche Bitkombination wählt bei Timer 0,1 den 16-Bit Counter Betrieb - TMOD?

a) xx01 xx01	X
b) xx11 xx10	
c) xx10 xx11	
d) xx00 xx01	

4. Welche Datenübertragungsrate wird an USB 1.1 erreicht?

a) 768 KBit/s	
b) 128 MBit/s	
c) 12 MBit/s	X
d) 480 MBit/s	

5. Welches Register bestimmt die Funktionsweise der E/A-Steuerung von Timer 2?

a) CRCH	
b) CCH2	
c) CCEN	X
d) TL2	

6. Welcher Befehl kann mit dem Register R2 durchgeführt werden?

a) mov @A,R2	
b) mov R2,A	X
c) mov A,@R2	
d) movx R2,@A	

7. Welche Aussagen über Datenraten von Bussen sind nicht korrekt?

a) IEEE-1394 -> 400MBit/s	
b) USB 2.0 -> 480 MBit/s	
c) RS 232 -> 1 Mbit/s	
d) USB 1.1 -> 12 Mbit	X

8. Welchen Wert hat der Stackpointer nach dem Einschalten des Controllers?

a) 0000 0111b	X
b) 00	
c) 030h	
d) 020h	



9. Welche Funktion hat die ORG-Anweisung?

a) Die Datenpointer werden initialisiert.	
b) Der PC wird auf die gewünschte Programmspeicheradresse gesetzt.	<b>X</b>
c) Der Datenpointer wird auf die gewünschte Programmspeicheradresse gesetzt.	
d) Der Datenspeicher wird initialisiert.	

10. Welche Aussage für die Bestückung von Leiterplatten ist nicht korrekt?

a) Es kann ein beliebiger unregelmäßiger Lötkegel verwendet werden.	<b>X</b>
b) Das Kupfer sollte nicht direkt mit Händen angefasst werden.	
c) Ein Bestückungsplatz mit Halterung für die Platine ist vorteilhaft.	
d) Eine Prüfung der Versorgungsspannung auf Kurzschlüsse ist sinnvoll.	

11. Welcher Interruptvektor ist für Timer 1 Overflow zu verwenden?

a) 0Bh	
b) 0B	
c) 27	<b>X</b>
d) 2Bh	

12. Welches Bit befindet sich im bitadressierbaren Bereich des internen Datenspeichers?

a) SBUF	
b) 20H.0	<b>X</b>
c) EXT12	
d) 30.0B	

13. Was bewirkt eine HAL im Betriebssystem?

a) Es sind direkte Zugriffe auf die Hardware möglich.	
b) Hardwarezugriffe erfolgen nur über das Betriebssystem	<b>X</b>
c) Die Abkürzung steht für High ALL und versetzt den Prozessor in einen sicheren Zustand	
d) Das Betriebssystem wird hierdurch multitaskingfähig	

14. Welche Flags werden bei Aufruf der ISR automatisch von der Hardware zurückgesetzt?

a) TF0	
b) TF2	
c) TF0, TF1	<b>X</b>
d) TF1	

15. In welchem Speicherbereich liegt Port 1?

a) externer Datenspeicherbereich ab 2000h	
b) im SFR-Bereich externer Datenspeicher	
c) im internen Datenspeicher ab 80h	<b>X</b>
d) im internen Datenspeicher unter 20h	

16. Was ist I2C.

a) Der neue Intercity	
b) Ein geräteinterner Bus.	<b>X</b>
c) Ein serieller Bus für Fabriken	
d) Ein serieller Bus zur Vernetzung von Fernsehgeräten mit PCs	