



Prüfung: μ-Computertechnik - Diplom
Termin: Dienstag, 5.02.2008; 11:00-13:00
Prüfer: Prof. Walter
Hilfsmittel: beliebig, keine Kommunikationsmittel

Name:	_____
Vorname:	_____
Studiengang:	_____
Labor:	_____
USB-Stick:	_____

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

(nicht ausfüllen)!

Aufgabe	mögl. Punkte	erreichte Punkte
1	12	
2	18	
3	14	
4	6	
Gesamt	50	
	Note	

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

Viel Erfolg!

Bemerkungen: Leeren Sie bei Prüfungsbeginn den Stick. Bitte erstellen Sie die Lösungen auf der eigenen Festplatte und kopieren diese anschließend auf den Stick.

Schreiben Sie in jeden Programmkopf ihren Namen! Bei nicht vorhandenem Namen wird die Lösung NICHT gewertet.



1. 8051/80535 Assembler „K-SYNC.ASM“

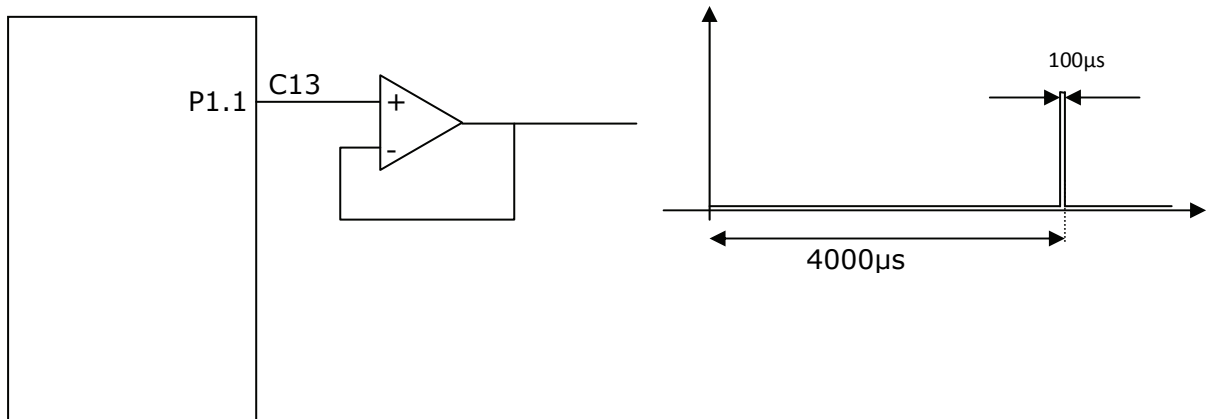
12 Punkte Σ _____

Für die Synchronisation von Video und Audio muss alle (2) 4 ms ein Impuls der Breite 100µs erzeugt werden. Das Ausgangssignal von P1.1 wird über einen Spannungsfolger ausgegeben.

- a) Zeichnen Sie die Schaltung und das zu erzeugende Signal
- b) Erstellen Sie die PDL
- c) Erstellen Sie das Programm in 8051-Assembler

Bemerkung zur Lösung: Im Diplom wurden andere Zahlenwerte verwendet als im Bachelor

Lösung a)



Lösung b)

Verfahren: Timer 2 als eigenständige Einheit
Periodendauer: 2ms oder 4ms (Bachelor – Diplom)

```
Pulsweitenmodulation mit Timer 2 zur Synchronisation von Audio und Video,  
; Ausgabe an Port 1.1 (VG1/C13). +5V - (VG1/A4) Masse VG1/A21  
; oder VG1/C21  
; Initialisierung Timer 2:  
; T2CON: Autoreload bei Ueberlauf OSC/12  
; CRC: Periodendauer 2ms = 65536-2000 = 63536 = 0F830H  
; Diplom: 4ms = 65536-4000 = 61536 = F060H  
; CC1: Pulsweite 0,1ms = 65536-100 = 65436 = 0FF9CH  
; CCEN: Compare-Mode CC1  
; Warteschleife
```



Lösung c)

```
$nomod51 ;der Modus fuer 8051 wird abgeschaltet
$debug
$no1ist ;es wird kein Listing fuer reg535 erstellt
#include(reg535.pdf) ;die 80535-spezifischen Daten
$list ;es wird ein Listing erstellt
$title (K-SYNC.A51)
;-----
;Programmbeschreibung
;-----
;Programm PDL:
; Pulsweitenmodulation mit Timer 2 zur Synchronisation von Audio
und Video,
; Ausgabe an Port 1.1 (VG1/C13). +5V - (VG1/A4) Masse VG1/A21
; oder VG1/C21
; Initialisierung Timer 2:
; T2CON: Autoreload bei Ueberlauf OSC/12
; CRC: Periodendauer 2ms = 65536-2000 = 63536 = 0F830H
; Diplom: 4ms = 65536-4000 = 61536 = F060H
; CC1: Pulsweite 0,1ms = 65536-100 = 65436 = 0FF9CH
; CCEN: Compare-Mode CC1
; Warteschleife
;
;Grundlegende Funktion:
;* der Reloadwert im CRC legt die Periodendauer fest
;* der Wert im Register CC1 legt das Tastverhaeltnis fest
;* Der Prozessor wird nach Start des Timers nicht mehr belastet!
;* Werte in Klammern für Diplom FT
;
;Datum: 12.Januar 2008
;Programmiert: J. Walter
;
;Verwendete Einspruenge: keine
;
;Verwendete Unterprogramme: keine
;
;
;Verwendete Register und Variable:
;Registerbank(0)
;
;
;Kommentar:
;
;Aenderungen:
;Datum:
;
;-----
;Initialisierungsteil für allgemeine Konstanten
;-----
Periode_low EQU 030H ;legt die Periodendauer fest 2 ms / 4ms
Periode_high EQU 0F8H ;63536 = 0F830H (61536 = 0F060H) Diplom
PWM_CCL1 EQU 09CH ;Compare Wert 0.1ms = 100µs
PWM_CCH1 EQU 0FFH ;Compare Wert FF9CH

CSEG AT 0H ;Legt absolute Codesegmentadresse auf 0h
jmp INIT
```



```
;-----  
;Initialisierungsteil für On-Chip Peripherie  
;-----  
ORG 100H                                ;Programmstart bei 100H  
  
INIT:  
  
mov  T2CON,#00010001B                    ;Timer Intitialisierung  
                                           ;Timer2 Einstellung:Timer mit f=f(osz)/12  
                                           ;Mode 0 ->Reload bei Ueberlauf  
mov  CCEN,#00001000B                    ;Compare Mode des CC1 freigeben  
  
mov  CRCL,#Periode_low                   ;Low Byte von Compare Reload Capture  
mov  CRCH,#Periode_high                  ;High Byte von Compare Reload Capture  
  
mov  CCL1,#PWM_CCL1                      ;P1.1 Compare Capture Low Byte  
mov  CCH1,#PWM_CCH1                      ;CC High Byte  
  
  
;-----  
;Programmschleife  
;-----  
  
PWM:  
jmp  PWM  
  
end
```



Überprüfung des Programmes erfolgt mit dem Logik Analysator im Simulator:

The screenshot shows the µVision3 Logic Analyzer interface. The main window displays a timing diagram for PORT1.1, showing a series of pulses between 0x0 and 0x1. The x-axis represents time, with markers at 0.065000 s, 0.065543 s, 0.075000 s, and 0.085000 s. The y-axis represents the signal level, with markers at 0x0 and 0x1. The interface includes a menu bar (File, Edit, View, Project, Debug, Flash, Peripherals, Tools, SVCS, Window, Help), a toolbar, and a Project Workspace panel on the left. The Project Workspace panel shows a list of registers and system variables with their current values.

Register	Value
Regs	
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
Sys	
a	0x00
b	0x00
sp	0x07
sp_max	0x07
dptr	0x0000
PC \$	C:0x0112
states	815502
sec	0.81550200
psw	0x00

Output Window: Running with Code
Load "D:\\8051\\K-LA (PORT1 & 0x2) >
>
ASM ASSIGN

Watches: Name, Address: (empty)

Locals: Watch #1

Memory: Memory #1, Mem



2. 8051/80535 Programm „SANFTx.ASM“

18 Punkte Σ_____

Autos der höheren Klasse schalten die Innenraumbelichtung nicht einfach aus oder ein. Die Innenraumbelichtung wird „sanft“ ein-, und ausgeschaltet.

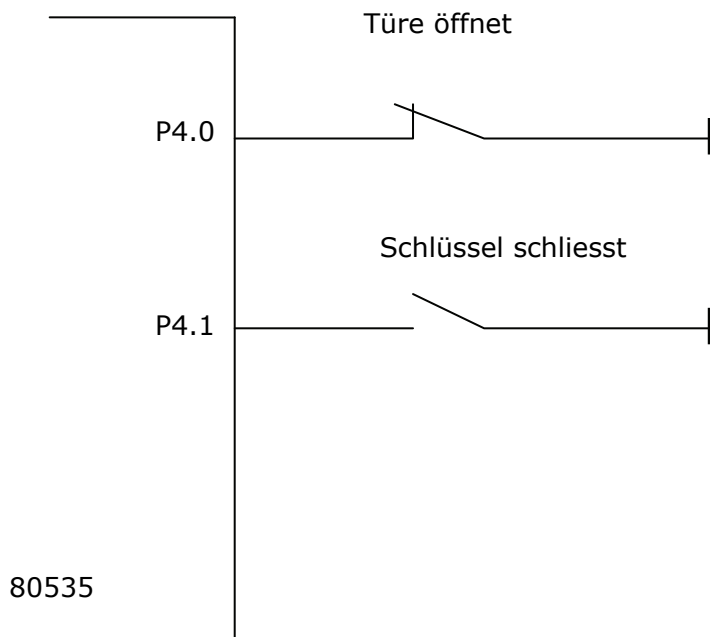
Folgende Tests werden mit Timer 2, Periodendauer 20ms durchgeführt:

2.1 Geht die Türe auf, wird die Innenraumbelichtung in 1 Sekunde von 10% auf 90% Helligkeit gefahren.

2.2 Wird die Türe mit einem Schlüssel abgeschlossen, wird in 20 Sekunden von 90% Helligkeit auf 10% Helligkeit heruntergefahren und danach die Innenraumbelichtung vollständig ausgeschaltet.

Bem.: Es genügt, das High-Byte des CC1 Registers zu verändern

- a) Zeichnen Sie ein Blockschaltbild für die Anlage.
Türe auf → Schalter 1 an Port 4.0 wird geöffnet. Abschließen mit Schlüssel → Schalter 2 an Port 4.1 wird geschlossen.
- b) Schreiben Sie die PDL und das Programm SANFT1.ASM für Fall 2.1
- c) Schreiben Sie die PDL und das Programm SANFT2.ASM für Fall 2.2





b) Sanft 2.1

```
;Programm - PDL:
; Pulsweitenmodulation mit Timer 2 zur Beleuchtungsansteuerung,
; Ausgabe an Port 1.1 (VG1/C13). +5V - (VG1/A4) Masse VG1/A21
; oder VG1/C21
; Periodendauer 20ms
;   Schalter 1 an P4.0 Tuer auf - Schalter oeffnet
;   10% von 20ms = 2ms 45536+2000=47536=B9B0H
;   90% von 20ms = 18ms 65536-2000=63536= F830H
; Loesung 1:
;   16000 / 50 = 320 addieren 50 mal
;   1s laesst sich mit 50 mal 20ms realisieren --> Bei jedem Ueberlauf
;   320 addieren
;   Schalter 4.0 - "1"
;   PWM auf 10% setzen
;   R2=50 Zaehlregister
;   Interrupt freigeben
;   ISR_TIMER_2
;   Addition mit 320
;   Zaehlregister =0?
; Loesung 2:
;   Werden nur das obere Byte geaendert --> F8-B9=3F=63 Aenderungen pro
;   Sekunde
;   oder alle 16 ms = CCH1 +1 --> 16ms werden mit Timer 0 erzeugt
;   oder alle 3 * 20ms + 1 und 1 * 20 ms + 2
; Loesung 3:
;   Der Timer wird mit Periodendauer 2µ betrieben
;   10% von 20ms = 2ms --> 55536+1000=56436
;   90% von 20ms = 18ms 65536-1000=64536= F830H
;   8000 / 50 = 160 addieren 50 mal
```

```
$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 (SANFT1.A51)
```

;Programmbeschreibung

```
;Programm - PDL:
; Pulsweitenmodulation mit Timer 2 zur Beleuchtungsansteuerung,
; Ausgabe an Port 1.1 (VG1/C13). +5V - (VG1/A4) Masse VG1/A21
; oder VG1/C21
; Periodendauer 20ms
;   Schalter 1 an P4.0 Tuer auf - Schalter oeffnet
;   10% von 20ms = 2ms 45536+2000=47536=B9B0H
;   90% von 20ms = 18ms 65536-2000=63536= F830H
; Loesung 1:
;   16000 / 50 = 320 addieren 50 mal
;   1s laesst sich mit 50 mal 20ms realisieren --> Bei jedem Ueberlauf
320 addieren
;   Schalter 4.0 - "1"
;   PWM auf 10% setzen
;   R2=50 Zaehlregister
;   Interrupt freigeben
;   ISR_TIMER_2
;   Addition mit 320
```



```
; Zaehlregister =0?
; Loesung 2:
; Werden nur das obere Byte geaendert --> F8-B9=3F=63 Aenderungen pro
Sekunde =
; oder alle 16 ms = CCH1 +1 --> 16ms werden mit Timer 0 erzeugt
; oder alle 3 * 20ms + 1 und 1 * 20 ms + 2
; Loesung 3:
; Der Timer wird mit Periodendauer 2µ betrieben
; 10% von 20ms = 2ms --> 55536+1000=56436
; 90% von 20ms = 18ms 65536-1000=64536= F830H
; 8000 / 50 = 160 addieren 50 mal
;Grundlegende Funktion:
;* der Reloadwert im CRC legt die Periodendauer fest
;* der Wert im Register CC1 legt das Tastverhaeltnis fest
;
;Datum: 12. Januar 2008
;Programmiert: J. Walter
;
;Verwendete Einspruenge: keine
;
;Verwendete Unterprogramme: keine
;
;
;Verwendete Register und Variable:
;Registerbank(0)
;R1 Wertuebergabe des A/D Wertes
;
;Kommentar:
;
;Aenderungen:
;Datum:
;
;-----
;Initialisierungsteil für allgemeine Konstanten
;-----
Periode_low EQU 0E0H ;legt die Periodendauer fest 20ms
Periode_high EQU 0B1H ;45536 = 0B1E0H
PWM_10_CC1L EQU 0B0H ;Compare Wert
PWM_10_CC1H EQU 0B9H ;Compare Wert 47536

ANZAHL_ADDITION EQU 50 ;Anzahl der Additionen

CSEG AT 0H ;Legt absolute Codesegmentadresse auf 0h
jmp ANFANG

ORG 02BH
clr TF2
call ISR_TIMER_2
reti

;-----
;Initialisierungsteil für On-Chip Peripherie
;-----
ORG 100H ;Programmstart bei 100H
ANFANG:
call INIT

;-----
```




```
;Programmschleife
;-----

PWM:
jb P4.0,VON_10_AUF_90      ;Oeffner betaetigt?
jmp PWM

VON_10_AUF_90:
setb T2I0                  ;Timer 2 laeuft
jmp PWM

ISR_TIMER_2:
call ADDITION
djnz R2,ENDE_ISR_TIMER_2
mov R2,#ANZAHL_ADDITION   ;
jb P4.0,$                  ;Warten bis Tuere wieder zu
call INIT                  ;Neu initialisieren
ENDE_ISR_TIMER_2:
ret

ADDITION:
clr C                      ;LOW-BYTE in AKKU
mov A,CCL1                 ;320 = 140H
add A,#40H                 ;Wieder in CC1L zurueck
mov CCL1,A                ;(kann weggelassen werden)
;
mov A,CCH1
addc A,#1                  ;Addiere 1 und Carry
mov CCH1,A
ret

INIT:
mov CCEN,#0                ;Timer 2 Intitialisierung
setb P1.1                  ;Damit P1.1 wieder auf 1
mov T2CON,#00010000B      ;P1.1 auf 1
mov CCEN,#00001000B      ;Timer2 Einstellung:Timer mit f=f(osz)/12
mov CCEN,#00001000B      ;Mode 0 ->Reload bei Ueberlauf
mov CRCL,#Periode_low     ;Compare Mode des CC1 freigeben
mov CRCH,#Periode_high
mov CCL1,#PWM_10_CC1L     ;Low Byte von Compare Reload Capture
mov CCH1,#PWM_10_CC1H     ;High Byte von Compare Reload Capture
;P1.1 Compare Capture Low Byte
;CC High Byte
;Interrupt
setb EAL                   ;Allgemeine Interruptfreigabe
setb ET2                   ;Interrupt Freigabe Timer 2
mov R2,#ANZAHL_ADDITION   ;50 Additionen

ret

end
```



Test im Simulator – Logik Analysator

The screenshot shows the µVision3 Logic Analyzer interface. The main window displays a timing diagram for PORT1.1, showing a series of pulses between 0x0 and 0x1. The x-axis represents time, with markers at 3.300000 s, 3.550000 s, and 3.800000 s. The y-axis represents the signal level, with 0x1 at the top and 0x0 at the bottom. The interface includes a menu bar (File, Edit, View, Project, Debug, Flash, Peripherals, Tools, SVCS, Window, Help) and a toolbar with various icons. The Project Workspace on the left shows a list of registers and system variables:

Register	Value
r0	0x00
r1	0x00
r2	0x32
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
a	0xf8
b	0x00
\$p	0x0b
\$p_max	0x0d
dptr	0x0000
PC \$	C:0x0111
states	6761501
sec	6.76150100
psw	0x01

Two dialog boxes are open: "Parallel Port 4" and "Parallel Port 1". Both show the port number, a value (0xFF for Port 4 and 0xFD for Port 1), and a 7-bit mask. The "Output Window" at the bottom left shows the command "ASM ASSIGN" and the output "LA (PORT1 & 0x2) >> 1". The "Watches" and "Memory" windows at the bottom right are empty.



Sanft 2.2

PDL unter Programmbeschreibung

```
$nomod51 ;der Modus fuer 8051 wird abgeschaltet
$debug
$no1ist ;es wird kein Listing fuer reg535 erstellt
#include(reg535.pdf) ;die 80535-spezifischen Daten
$list ;es wird ein Listing erstellt
$title (SANFT2.A51)
;-----
;Programmbeschreibung
;-----
;Programm - PDL:
; Pulsweitenmodulation mit Timer 2 zur Beleuchtungsansteuerung,
; Ausgabe an Port 1.1 (VG1/C13). +5V - (VG1/A4) Masse VG1/A21
; oder VG1/C21
; Periodendauer 20ms
; Schluessel an P4.1=0 abschliessen
; 10% von 20ms = 2ms 45536+2000=47536= B9B0H
; 90% von 20ms = 18ms 65536-2000=63536= F830H
; Loesung 1:
; 16000 / 500 = 32 abziehen 500 mal
; 10s laesst sich mit 500 mal 20ms realisieren --> Bei jedem ;
; Ueberlauf 32 abziehen
; Schalter 4.1 - "0"
; PWM auf 90% setzen
; R2=50 Zaehlregister
; Interrupt freigeben
; ISR_TIMER_2
; Addition mit 320
; Zaehlregister =0?
;
; Werden nur das obere Byte geaendert --> F8-B9=3F=63 Aenderungen
; pro Sekunde =
; oder alle 16 ms = CCH1 +1 --> 16ms werden mit Timer 0 erzeugt
; oder alle 3 * 20ms + 1 und 1 * 20 ms + 2
;Grundlegende Funktion:
;* der Reloadwert im CRC legt die Periodendauer fest
;* der Wert im Register CC1 legt das Tastverhaeltnis fest
;
;Datum: 12. Januar 2008
;Programmiert: J. Walter
;
;Verwendete Einspruenge: keine
;
;Verwendete Unterprogramme: keine
;
;
;Verwendete Register und Variable:
;Registerbank(0)
;
;
;Kommentar:
;
;Aenderungen:
;Datum:
;
;-----
```



```
;Initialisierungsteil für allgemeine Konstanten
;-----
Periode_low      EQU 0E0H      ;legt die Periodendauer fest 20ms
Periode_high     EQU 0B1H      ;45536 = 0B1E0H
PWM_90_CC1L     EQU 030H      ;Compare Wert
PWM_90_CC1H     EQU 0F8H      ;Compare Wert 63536

ANZAHL_SUB      EQU 50        ;Anzahl der Subtraktionen
ANZAHL_SUB_2    EQU 10

CSEG AT 0H      ;Legt absolute Codesegmentadresse auf 0h
jmp ANFANG

ORG 02BH
clr TF2          ;Interrupt Timer 2 bestaetigen
call ISR_TIMER_2
reti

;-----
;Initialisierungsteil für On-Chip Peripherie
;-----
ORG 100H          ;Programmstart bei 100H
ANFANG:
call INIT        ;Aufruf der Initialisierung

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

PWM:
jnb P4.1,VON_90_AUF_10      ;Taste 2 betaetigt?
jmp PWM

VON_90_AUF_10:
setb T2I0          ;Timer 2 laeuft
jmp PWM

ISR_TIMER_2:
call SUBTRAKTION
djnz R2,ENDE_ISR_TIMER_2    ; 50
mov R2,#ANZAHL_SÜB
djnz R3,ENDE_ISR_TIMER_2    ; mal 10
;
jnb P4.1,$          ;Warten bis Tuere wieder aufschliessen
call INIT          ;Neu initialisieren
ENDE_ISR_TIMER_2:
ret

SUBTRAKTION:
clr C
mov A,CCL1          ;LOW-BYTE in AKKU
subb A,#32          ;subtrahiere 32
mov CCL1,A          ;Und wieder in CC1L zurueck
mov A,CCH1
subb A,#0           ;subtrahiere das Carry
mov CCH1,A
ret

INIT:
```



```
mov CCEN,#0                ;Timer Intialisierung
setb P1.1                  ;Damit P1.1 wieder auf 1
mov T2CON,#00010000B      ;P1.1 auf 1
                           ;Timer2 Einstellung:Timer mit f=f(osz)/12
                           ;Mode 0 ->Reload bei Ueberlauf
mov CCEN,#00001000B      ;Compare Mode des CC1 freigeben

mov CRCL,#Periode_low     ;Low Byte von Compare Reload Capture
mov CRCH,#Periode_high    ;High Byte von Compare Reload Capture
mov CCL1,#PWM_90_CC1L     ;P1.1 Compare Capture Low Byte
mov CCH1,#PWM_90_CC1H     ;CC High Byte
                           ;Interrupt
setb EAL                   ;Allgemeine Interruptfreigabe
setb ET2                   ;Interrupt Freigabe Timer 2
mov R2,#ANZAHL_SUB        ;50 Subtraktionen
mov R3,#ANZAHL_SUB_2      ;50 mal 10 = 500 Subtraktionen
ret

end
```



Test im Simulator – Logik Analysator

The screenshot displays the µVision3 Logic Analyzer interface. The main window shows a logic trace for PORT1:1, with a time range from 9.650000 s to 10.650000 s. The trace shows a series of vertical red lines representing digital signals. On the left, the Project Workspace shows the Register and Sys views. The Register view lists registers r0 through r7, and the Sys view lists system variables like a, b, sp, sp_max, dptr, PC, states, sec, and psw. Three dialog boxes are open: 'Compare Capture Register' (showing a table of compare registers), 'Parallel Port 4' (showing port configuration), and 'Timer/Counter 2' (showing timer settings).

x	CCx	Mode	IRQ	Pin
0	B1E0H	Com./Cap Disabled	0	1
1	B9B0H	Compare Mode 0/1	1	1
2	0000H	Com./Cap Disabled	0	1
3	0000H	Com./Cap Disabled	0	1

Port 4	7	Bits	0
P4: 0xFD	✓	✓	✓
Pins: 0xFD	✓	✓	✓

Mode: 1: Timer running
Reload: on T2 overflow
T2CON: 0x11 <input checked="" type="checkbox"/> T2EX
T2: 0xEAB7 <input checked="" type="checkbox"/> T2 Pin
<input type="checkbox"/> T2PS
CRC: 0xB1E0 <input checked="" type="checkbox"/> TF2

Output Window: Running with Code Size Limit
Load "D:\8051\SANFT2\Vorl
LA (PORT1 & 0x2) >> 1
ASM ASSIGN BreakDisable

Watches: Name Value
Locals: Watch #1 Watch #2
Memory: Address: Memory #1 Memory #2 Memory #3



3. 8051/80535 Programm „BREMS“

14 Punkte Σ ____

Am Bremspedal eines Fahrzeugs wird ein Potentiometer angebracht. Ein Bremslicht signalisiert folgende Stufen:

- | | |
|-----------------------------|---|
| 1. Bremse nicht betätigt | $0V \leq UPoti \leq 2V$ |
| 2. Bremse mit 33% betätigt | $2V < UPoti \leq 3V$ Port 4.0 = "0" |
| 3. Bremse mit 66% betätigt | $3V < UPoti \leq 4V$ Port 4.0; P4.1 = "0" |
| 4. Bremse mit 100% betätigt | $4V < UPoti \leq 5V$ Port 4.0; P4.1; P4.2 = "0" |

- Zeichnen Sie das Schaltbild für die Aufgabe mit jeweils einer von drei Leuchtdioden an den Ports 4.x
- Schreiben Sie die PDL
- Schreiben Sie das Programm

PDL ist im Programm



```
$nomod51                ;der Modus fuer 8051 wird abgeschaltet
$debug
$no!ist                 ;es wird kein Listing fuer reg535 erstellt
#include(reg535.pdf)     ;die 80535-spezifischen Daten
$list                   ;es wird ein Listing erstellt
$title (BREMS.A51)
;-----
;Programmbeschreibung
;-----
;Programm:  A/D-Wandler Spannungen dekodiert
;U/V        P4.2          P4.1          P4.0
;0<=U<=2V   1            1            1
;2< U<=3V   1            1            0
;3< U<=4V   1            0            0
;4< U<=5V   0            0            0
;
;PDL
;   Wandlung
;   Subtraktion von 2V, 1V, 1V
;   Carry gesetzt? --> Leuchtenkombination geht an

;Datum: 08.01.2008
;Programmiert: J. Walter
;
;Verwendete Einspruenge: keine
;
;Verwendete Unterprogramme: keine
;
;
;Verwendete Register und Variable:
;Registerbank(0)
;R2
;
;Kommentar:
;
;Aenderungen:
;Datum: 29.10.2007
;
;-----
;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 ADCON,#1000000B      ;Kanal AN0 angewaehlt, Stop nach der Wandlung
```




```
                                ;1 für Baudrate 4800bps ->9600bps
;-----
;Programmschleife
;-----

ABFRAGE:
clr P3.2                        ;Leuchte 1 Port 3.2 ruecksetzen, Programm laeuft
mov DAPR,#0                    ;Wandlung Start
jb BSY,$                       ;warten bis Wandlung stop
mov A,ADDAT                    ;Wert im ACCU
clr C                          ;Carry loeschen
subb A,#103                    ;entspricht U>2V
jc AUS_AUS                    ;Sprung da U<=2V
subb A,#51                     ;entspricht U>3V
jc AUS_EIN                    ;Sprung da U<=3V
subb A,#51                     ;entspricht U>4V
jc EIN_AUS                    ;Sprung da U<=4V
clr P4.0                      ;Leuchte 0 ein
clr P4.1                      ;Leuchte 1 ein
clr P4.2                      ;Leuchte 2 ein
jmp ABFRAGE                   ;Sprung zum Programmbeginn

AUS_AUS:
setb P4.0                    ;Leuchte 0 aus
setb P4.1                    ;Leuchte 1 aus
setb P4.2                    ;Leuchte 2 aus
jmp ABFRAGE

AUS_EIN:
clr P4.0                    ;Leuchte 0 ein
setb P4.1                    ;Leuchte 1 aus
setb P4.2                    ;Leuchte 2 aus
jmp ABFRAGE

EIN_AUS:
clr P4.0                    ;Leuchte 0 ein
clr P4.1                    ;Leuchte 1 ein
setb P4.2                    ;Leuchte 2 aus
jmp ABFRAGE

end
```



4. Prozessorauslastung

6 Punkte Σ_____

Schätzen Sie die Prozessorauslastung von Aufgabe 2 ab.
Annahme: Durchschnittliche Befehlszykluszeit = $2\mu\text{s}$

Wird die Periodendauer geändert:

Anzahl der Befehle * Zykluszeit * Anzahl pro Sekunde =

$$15 * 2\mu\text{s} * 50 = 1500\mu\text{s} = 1,5 \text{ ms}$$

Prozessorauslastung = $\frac{\text{Zeiten für Befehlsausführung}}{1\text{s}} = 0,15\%$

Wird die Periodendauer nicht geändert hat der Prozessor keinerlei Auslastung.