



Prüfung: μ-Computertechnik - Bachelor
Termin: Montag, 11.02.2013; 08:30-10:00
Prüfer: Prof. Walter
Hilfsmittel: beliebig, keine Kommunikationsmittel,
kein Servo ;-)

Name:	_____
Vorname:	_____
Studiengang:	_____
IZ-Abkürzung:	_____

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	14	
3	14	
4	10	
Gesamt	50	
	Note	

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

Viel Erfolg!

Bemerkungen: Bitte erstellen Sie die Lösungen auf der eigenen Festplatte im Ordner MCWS12_IZABKÜRZUNG (Bsp. waju0001). Am Ende der Klausur belassen Sie ihren Rechner am Platz und verlassen den Raum. Zum Kopieren ihrer Lösung werden Sie jeweils mit Namen im Anschluss an die Klausur aufgerufen. Bitte senden Sie ihre Lösung zusätzlich an: waju0001@web.de Betreff: MCSS12_IZABKÜRZUNG Schreiben Sie in jeden Programmkopf ihren Namen! Bei nicht vorhandenem Namen wird die Lösung NICHT gewertet.



WICHTIG!

Alle Programme sind für die VC_2-Hardware zu schreiben. Der Sysclk liegt bei 3 MHz. Speichern Sie jeweils die dazugehörige **Configuration Wizard Datei!!**
Nur dokumentierte Software existiert!

1. C8051F430-Programmierung „ABTAstrate_2“ (12 Punkte)

Für das Fluggerät „VC25“ muss ein Flugschreiber entwickelt werden. Dieser speichert wesentliche Flugdaten wie Höhe, Geschwindigkeit, Kurs, Neigungswinkel, Motordrehzahlen, Außentemperatur und Batterietemperaturen des Flugzeuges. Durch unterschiedliche Abtastraten können die Daten reduziert werden. Für einen ersten Test werden Teilprogramme entwickelt, welche mit T3 starten.

- a) Erzeugen Sie mit Hilfe von Timer 0 eine Abtastrate von 250 Hz und schalten Sie jeweils in der Interrupt-Service-Routine ISR_T0 die LED L1. Erzeugen Sie mit Hilfe von Timer 2 eine Abtastrate von 5Hz und schalten Sie jeweils in der Interrupt-Service-Routine ISR_T2 die LED L2.

```
$NOMOD51                ;der Modus fuer 8051 wird abgeschaltet
$debug
$nolist                 ;es wird kein Listing fuer reg535 erstellt
#include(C8051F340.inc) ;die C8051F340-spezifischen Daten
$list                   ;es wird ein Listing erstellt
$title (ABTAstrate_2.A51)
;-----
;Programmbeschreibung
;-----
;Programm:
;Oszillator: 3MHz
;T3 startet Programm P1.3
;Timer 0 Abtastrate 250Hz --> 3MHz/(48*250) --> 6
;ISR_0 --> LED1 -->P3.2
;TIMER 2 Abtastrate 5Hz = 200ms
;ISR_2 --> LED2 --> P3.3
;
;Erstellt am: Sonntag, 10. Februar 2013 15:34:22
;Programmiert: Juergen Walter
;
;Verwendete Einspruege: keine
;
;Verwendete Unterprogramme: keine
;
;
;Verwendete Register und Variable:
;Registerbank(0)
;R2
;
;Kommentar:
;
;Aenderungen:
;Geaendert am: Sonntag, 10. Februar 2013 15:34:22
;
;
;-----
;Initialisierungsteil fuer allgemeine Konstanten
;-----

CSEG AT 0H                ;Legt absolute Codesegmentadresse auf 0h
call Init_Device          ;Aufruf zur Initialisierung der Controller
Funktionen
```



```
jmp INIT
;
;-----
;Interrupt-Vektoren
;-----
ORG 0BH
call ISR_T0           ;Timer 0
reti

ORG 2BH
clr TF2H             ;Bestaetigung T2 Interrupt
call ISR_T2
reti
;-----
;Initialisierungsteil fuer On-Chip Peripherie
;-----
ORG 100H             ;Programmstart bei 100H
INIT:

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

ABFRAGE:
jnb P1.3,LOS        ;T1?
jmp ABFRAGE

LOS:
setb TR0             ;Timer 0 rennt
setb TR2             ;Timer 2 rennt
jmp ABFRAGE

ISR_T0:
cpl P3.2            ;L1 schaltet
ret

ISR_T2:
cpl P3.3            ;L2 schaltet
ret

; Peripheral specific initialization functions,
; Called from the Init_Device label
PCA_Init:
    anl  PCA0MD,    #0BFh
    mov  PCA0MD,    #000h
    ret

Timer_Init:
    mov  TMOD,      #002h
    mov  CKCON,     #002h
    mov  TL0,       #006h
    mov  TH0,       #006h
    mov  TMR2RLH,   #0B0h
    mov  TMR2RLH,   #03Ch
    mov  TMR2L,     #0B0h
    mov  TMR2L,     #03Ch
    ret

Port_IO_Init:
; P0.0 - Skipped, Push-Pull, Digital
; P0.1 - Skipped, Push-Pull, Digital
; P0.2 - Skipped, Push-Pull, Digital
; P0.3 - Skipped, Push-Pull, Digital
; P0.4 - TX0 (UART0), Push-Pull, Digital
```



```
; P0.5 - RX0 (UART0), Push-Pull, Digital
; P0.6 - Skipped, Push-Pull, Digital
; P0.7 - Skipped, Push-Pull, Digital

; P1.0 - CEX0 (PCA), Push-Pull, Digital
; P1.1 - CEX1 (PCA), Push-Pull, Digital
; P1.2 - CEX2 (PCA), Push-Pull, Digital
; P1.3 - Skipped, Push-Pull, Digital
; P1.4 - CEX3 (PCA), Push-Pull, Digital
; P1.5 - Skipped, Push-Pull, Digital
; P1.6 - Skipped, Push-Pull, Digital
; P1.7 - Skipped, Push-Pull, Digital

; P2.0 - Skipped, Push-Pull, Digital
; P2.1 - Skipped, Push-Pull, Digital
; P2.2 - Skipped, Push-Pull, Digital
; P2.3 - Skipped, Push-Pull, Digital
; P2.4 - Skipped, Push-Pull, Digital
; P2.5 - Skipped, Push-Pull, Digital
; P2.6 - Skipped, Push-Pull, Digital
; P2.7 - Skipped, Push-Pull, Digital

; P3.0 - Skipped, Push-Pull, Digital
; P3.1 - Skipped, Push-Pull, Digital
; P3.2 - Skipped, Push-Pull, Digital
; P3.3 - Skipped, Push-Pull, Digital
; P3.4 - CEX4 (PCA), Push-Pull, Digital
; P3.5 - T0 (Timr0), Push-Pull, Digital
; P3.6 - T1 (Timr1), Push-Pull, Digital
; P3.7 - Unassigned, Push-Pull, Digital

mov P0MDOUT, #0FFh
mov P1MDOUT, #0FFh
mov P2MDOUT, #0FFh
mov P3MDOUT, #0FFh
mov P0SKIP, #0CFh
mov P1SKIP, #0E8h
mov P2SKIP, #0FFh
mov P3SKIP, #00Fh
mov XBR0, #001h
mov XBR1, #075h
ret

Oscillator_Init:
mov OSCICN, #081h
ret

Interrupts_Init:
mov IT01CF, #010h
mov IE, #0A2h
ret

; Initialization function for device,
; Call Init_Device from your main program
Init_Device:
lcall PCA_Init
lcall Timer_Init
lcall Port_IO_Init
lcall Oscillator_Init
lcall Interrupts_Init
ret
```

end



2. C8051F430-Programmierung „ADC_LED_VAR“ (14 Punkte)

Um die A/D-Wandlung inkl. Speicherung zu testen wird an Port 2.0 über ein Potentiometer eine Spannung zwischen 0V und 3,3V angelegt und gegen GND gemessen. Sobald T3 betätigt wird soll mit einer Abtastrate von 5Hz die Spannung an Port 2.0 gewandelt und hierdurch die Helligkeit der LED 3 verändert werden. Hierzu können Sie eine PWM-modulierte Spannung über CEX0 an LED 3 ausgeben. Hierzu soll das 8-Bit Ergebnis aus der AD-Wandlung in PCA0CPH0 geschrieben werden.

```
$NOMOD51                ;der Modus fuer 8051 wird abgeschaltet
$debug
$nolist                 ;es wird kein Listing fuer reg535 erstellt
#include(C8051F340.inc) ;die C8051F340-spezifischen Daten
$list                   ;es wird ein Listing erstellt
$title (ADC_LED_VAR.A51)
;-----
;Programmbeschreibung
;-----
;Programm:
;Wie Programm ABTASTRATE_2 und
;Port 2.0 ADC
;
;
;Erstellt am: Sonntag, 10. Februar 2013 17:20:14
;Programmiert: Juergen Walter
;
;Verwendete Einspruenge: keine
;
;Verwendete Unterprogramme: keine
;
;
;Verwendete Register und Variable:
;Registerbank(0)
;R2
;
;Kommentar:
;
;Aenderungen:
;Geaendert am: Sonntag, 10. Februar 2013 17:20:14
;
;
;-----
;Initialisierungsteil fuer allgemeine Konstanten
;-----

CSEG AT 0H                ;Legt absolute Codesegmentadresse auf 0h
call Init_Device          ;Aufruf zur Initialisierung der Controller Funktionen
jmp INIT
;
;-----
;Interrupt-Vektoren
;-----
ORG 2BH
clr TF2H                  ;Interrupt bestaetigen
call ISR_T2
reti
;-----
;Initialisierungsteil fuer On-Chip Peripherie
;-----
ORG 100H                  ;Programmstart bei 100H
INIT:
```



```
-----  
; Programmschleife  
-----  
  
ABFRAGE:  
jnb P1.3, PWM ; T3?  
jmp ABFRAGE  
  
PWM:  
setb TR2 ; Timer 2 startet  
jmp ABFRAGE  
  
ISR_T2:  
setb ADOBUSY ; Wandlung starten  
jnb ADOBUSY, $ ; Warten bis Wandlung fertig  
; Zuerst LOW-Byte  
mov A, PCA0CPL0 ; LOW Byte Compare PCA0  
mov PCA0CPL0, A ; LB schreiben  
mov A, ADC0H ; Wert von Wandlung  
mov PCA0CPH0, A ; in Compare  
cpl P3.2 ; PWM laeuft  
ret  
  
-----  
; Peripheral specific initialization functions,  
; Called from the Init_Device label  
-----  
PCA_Init:  
mov PCA0CN, #040h  
anl PCA0MD, #0BFh  
mov PCA0MD, #008h  
mov PCA0CPM0, #0C2h  
ret  
  
Timer_Init:  
mov TMOD, #002h  
mov CKCON, #002h  
mov TL0, #006h  
mov TH0, #006h  
mov TMR2RLL, #0B0h  
mov TMR2RLH, #03Ch  
mov TMR2L, #0B0h  
mov TMR2H, #03Ch  
ret  
  
ADC_Init:  
mov AMX0N, #01Fh  
mov ADC0CF, #0FCh  
mov ADC0CN, #080h  
ret  
  
Port_IO_Init:  
; P0.0 - Skipped, Push-Pull, Digital  
; P0.1 - Skipped, Push-Pull, Digital  
; P0.2 - Skipped, Push-Pull, Digital  
; P0.3 - Skipped, Push-Pull, Digital  
; P0.4 - TX0 (UART0), Push-Pull, Digital  
; P0.5 - RX0 (UART0), Push-Pull, Digital  
; P0.6 - Skipped, Push-Pull, Digital  
; P0.7 - Skipped, Push-Pull, Digital  
  
; P1.0 - Skipped, Push-Pull, Digital  
; P1.1 - Skipped, Push-Pull, Digital  
; P1.2 - Skipped, Push-Pull, Digital  
; P1.3 - Skipped, Push-Pull, Digital
```



```
; P1.4 - Skipped,      Push-Pull,  Digital
; P1.5 - Skipped,      Push-Pull,  Digital
; P1.6 - Skipped,      Push-Pull,  Digital
; P1.7 - Skipped,      Push-Pull,  Digital

; P2.0 - Skipped,      Push-Pull,  Analog
; P2.1 - Skipped,      Push-Pull,  Digital
; P2.2 - Skipped,      Push-Pull,  Digital
; P2.3 - Skipped,      Push-Pull,  Digital
; P2.4 - Skipped,      Push-Pull,  Digital
; P2.5 - Skipped,      Push-Pull,  Digital
; P2.6 - Skipped,      Push-Pull,  Digital
; P2.7 - Skipped,      Push-Pull,  Digital

; P3.0 - Skipped,      Push-Pull,  Digital
; P3.1 - Skipped,      Push-Pull,  Digital
; P3.2 - Skipped,      Push-Pull,  Digital
; P3.3 - Skipped,      Push-Pull,  Digital
; P3.4 - CEX0 (PCA),   Push-Pull,  Digital
; P3.5 - Unassigned,   Push-Pull,  Digital
; P3.6 - Unassigned,   Push-Pull,  Digital
; P3.7 - Unassigned,   Push-Pull,  Digital

mov  P2MDIN,    #0FEh
mov  P0MDOUT,   #0FFh
mov  P1MDOUT,   #0FFh
mov  P2MDOUT,   #0FFh
mov  P3MDOUT,   #0FFh
mov  P0SKIP,    #0CFh
mov  P1SKIP,    #0FFh
mov  P2SKIP,    #0FFh
mov  P3SKIP,    #00Fh
mov  XBR0,      #001h
mov  XBR1,      #041h
ret

Oscillator_Init:
  mov  OSCICN,   #081h
  ret

Interrupts_Init:
  mov  IT01CF,   #010h
  mov  IE,       #0A2h
  ret

; Initialization function for device,
; Call Init_Device from your main program
Init_Device:
  lcall PCA_Init
  lcall Timer_Init
  lcall ADC_Init
  lcall Port_IO_Init
  lcall Oscillator_Init
  lcall Interrupts_Init
  ret

end
```



3. C8051F430-Programmierung „ADC_LED_SPEICHER“ (14 Punkte)

Um die A/D-Wandlung inkl. Speicherung zu testen wird an Port 2.0 über ein Potentiometer eine Spannung zwischen 0V und 3,3V angelegt und gegen GND gemessen. Damit die Wandlung sehr schnell ausgeführt wird, setzen Sie den SAR Conversion Clock auf 3MHz. Sobald T3 betätigt wird soll mit einer Abtastrate von 5Hz die Spannung an Port 2.0 gewandelt und 50 Werte in den externen Speicher ab Adresse 0000H geschrieben werden. Während der Wandlung blinkt LED 3. Nachdem 50 Werte gewandelt sind blinkt LED 3 nicht mehr.

```
$NOMOD51          ;der Modus fuer 8051 wird abgeschaltet
$debug
$no!ist           ;es wird kein Listing fuer reg535 erstellt
#include(C8051F340.inc) ;die C8051F340-spezifischen Daten
$list             ;es wird ein Listing erstellt
$title (ADC_LED_SPEICHER.A51)
;-----
;Programmbeschreibung
;-----
;Programm:
;Wie Programm ABTASTRATE_2 und
;Port 2.0 ADC
;50 Werte in externen Speicher ab Adresse 0
;
;Erstellt am: Sonntag, 10. Februar 2013 17:20:14
;Programmiert: Juergen Walter
;
;Verwendete Einspruenge: keine
;
;Verwendete Unterprogramme: keine
;
;
;Verwendete Register und Variable:
;Registerbank(0)
;R2
;
;Kommentar:
;
;Aenderungen:
;Geaendert am: Sonntag, 10. Februar 2013 17:20:14
;
;
;-----
;Initialisierungsteil fuer allgemeine Konstanten
;-----

CSEG AT 0H          ;Legt absolute Codesegmentadresse auf 0h
call Init_Device   ;Aufruf zur Initialisierung der Controller Funktionen
jmp INIT
;
;-----
;Interrupt-Vektoren
;-----
ORG 2BH
clr TF2H           ;Interrupt bestaetigen
call ISR_T2
reti
;-----
;Initialisierungsteil fuer On-Chip Peripherie
;-----
ORG 100H           ;Programmstart bei 100H
INIT:
;-----
```




```
;Programmschleife
;-----

ABFRAGE:
jnb P1.3,PWM          ;T3?
jmp ABFRAGE

PWM:
mov DPTR,#0          ;Startadresse
mov R2,#50          ;50 Werte schreiben
setb TR2            ;Timer 2 startet
jmp ABFRAGE

ISR_T2:
setb AD0BUSY        ;Wandlung starten
jb AD0BUSY,$        ;Warten bis Wandlung fertig
;Werte schreiben

mov A,ADC0H          ;Wert von Wandlung in Akku
movx @DPTR,A        ;Wert in Speicher
inc DPTR            ;X-Adresse+1
djnz R2,ENDE_ISR_T2 ;Bereits 50?
clr TR2             ;Timer 2 Halt
ENDE_ISR_T2:
cpl P3.2            ;PWM laeuft
ret

;-----
; Peripheral specific initialization functions,
; Called from the Init_Device label
;-----
PCA_Init:
    anl  PCA0MD,    #0BFh
    mov  PCA0MD,    #000h
    mov  PCA0CPM0,  #0C2h
    ret

Timer_Init:
    mov  TMOD,      #002h
    mov  CKCON,     #002h
    mov  TL0,       #006h
    mov  TH0,       #006h
    mov  TMR2RLH,   #0B0h
    mov  TMR2RLH,   #03Ch
    mov  TMR2L,     #0B0h
    mov  TMR2H,     #03Ch
    ret

ADC_Init:
    mov  AMX0N,     #01Fh
    mov  ADC0CF,    #0FCh
    mov  ADC0CN,    #080h
    ret

Port_IO_Init:
; P0.0 - Skipped,      Push-Pull,  Digital
; P0.1 - Skipped,      Push-Pull,  Digital
; P0.2 - Skipped,      Push-Pull,  Digital
; P0.3 - Skipped,      Push-Pull,  Digital
; P0.4 - TX0 (UART0), Push-Pull,  Digital
; P0.5 - RX0 (UART0), Push-Pull,  Digital
; P0.6 - Skipped,      Push-Pull,  Digital
; P0.7 - Skipped,      Push-Pull,  Digital

; P1.0 - Skipped,      Push-Pull,  Digital
; P1.1 - Skipped,      Push-Pull,  Digital
```



```
; P1.2 - Skipped,      Push-Pull,  Digital
; P1.3 - Skipped,      Push-Pull,  Digital
; P1.4 - Skipped,      Push-Pull,  Digital
; P1.5 - Skipped,      Push-Pull,  Digital
; P1.6 - Skipped,      Push-Pull,  Digital
; P1.7 - Skipped,      Push-Pull,  Digital

; P2.0 - Skipped,      Push-Pull,  Analog
; P2.1 - Skipped,      Push-Pull,  Digital
; P2.2 - Skipped,      Push-Pull,  Digital
; P2.3 - Skipped,      Push-Pull,  Digital
; P2.4 - Skipped,      Push-Pull,  Digital
; P2.5 - Skipped,      Push-Pull,  Digital
; P2.6 - Skipped,      Push-Pull,  Digital
; P2.7 - Skipped,      Push-Pull,  Digital

; P3.0 - Skipped,      Push-Pull,  Digital
; P3.1 - Skipped,      Push-Pull,  Digital
; P3.2 - Skipped,      Push-Pull,  Digital
; P3.3 - Skipped,      Push-Pull,  Digital
; P3.4 - CEX0 (PCA),   Push-Pull,  Digital
; P3.5 - Unassigned,   Push-Pull,  Digital
; P3.6 - Unassigned,   Push-Pull,  Digital
; P3.7 - Unassigned,   Push-Pull,  Digital
```

```
mov  P2MDIN,    #0FEh
mov  POMDOUT,   #0FFh
mov  P1MDOUT,   #0FFh
mov  P2MDOUT,   #0FFh
mov  P3MDOUT,   #0FFh
mov  POSKIP,    #0CFh
mov  P1SKIP,    #0FFh
mov  P2SKIP,    #0FFh
mov  P3SKIP,    #00Fh
mov  XBR0,      #001h
mov  XBR1,      #041h
ret
```

```
Oscillator_Init:
  mov  OSCICN,   #081h
  ret
```

```
Interrupts_Init:
  mov  IT01CF,   #010h
  mov  IE,       #0A2h
  ret
```

```
; Initialization function for device,
; Call Init_Device from your main program
```

```
Init_Device:
  lcall PCA_Init
  lcall Timer_Init
  lcall ADC_Init
  lcall Port_IO_Init
  lcall Oscillator_Init
  lcall Interrupts_Init
  ret
```

```
end
```



4. Eagle Schaltplan „ADC“ (10 Punkte)

Ein Schaltplan für die Aufgaben soll erstellt werden.

- a) Überprüfen Sie im Configuration-Wizard welche Anschlüsse für ein Zusatzboard notwendig sind. **An Port 2.0 der Poti-Abgriff.**
- b) Zeichnen Sie den Schaltplan mit Eagle unter Verwendung der Vorlage für Projekte. Hilfe: Unter Resistor finden Sie R-Trimm.

