



**Prüfung:** μ-Computertechnik  
**Termin:** Mittwoch, 31.01.2007 ; 8:30-10:30  
**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>15</b>	
<b>2</b>	<b>14</b>	
<b>3</b>	<b>11</b>	
<b>4</b>	<b>10</b>	
<b>Gesamt</b>	<b>50</b>	
	<b>Note</b>	

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

**Viel Erfolg!**

**Bemerkungen:**



**1. 8051/80535 Messeinrichtung „HAM.ASM“ 15 Punkte  $\Sigma$ \_\_\_\_\_**

Die Intelligenz eines Hamsters wird getestet. Ein Laufrad mit einer Gabellichtschranke und 32 Zähnen pro Umdrehung dient als Testeinrichtung. Sobald die Lichtschranke 1024 mal unterbrochen wurde, fährt ein Servomotor - an P1.1 angeschlossen - von Stellung 1 (ca. 0° = 0F650H) auf Stellung 2 (ca. 180° = 0FD70H), um damit Futter für den Hamster frei zu geben. Der Servomotor arbeitet bei einer Periodendauer von 20ms.

Der Ausgang der Gabellichtschranke wird auf P3.4 gelegt. Der Timer 0 zählt im Reload-Betrieb jeweils 1 Umdrehung und löst danach ein Interrupt aus. R2 enthält die noch abzuleistenden Umdrehungen. Annahme: Die Lichtschranke prellt nicht.

- a) (1P) Nach wieviel Umdrehungen erhält der Hamster Futter?
- b) (4P) Zeichnen Sie ein Blockschaltbild für die Anschlüsse am Mikrocontroller für den Servomotor und die Gabellichtschranke. Bitte geben Sie auch die Anschlüsse für die VG-Leiste an.
- c) (4P) Schreiben Sie die PDL
- d) (6P) Schreiben Sie das Programm

a) Anzahl der Umdrehungen: Nach 32 Umdrehungen

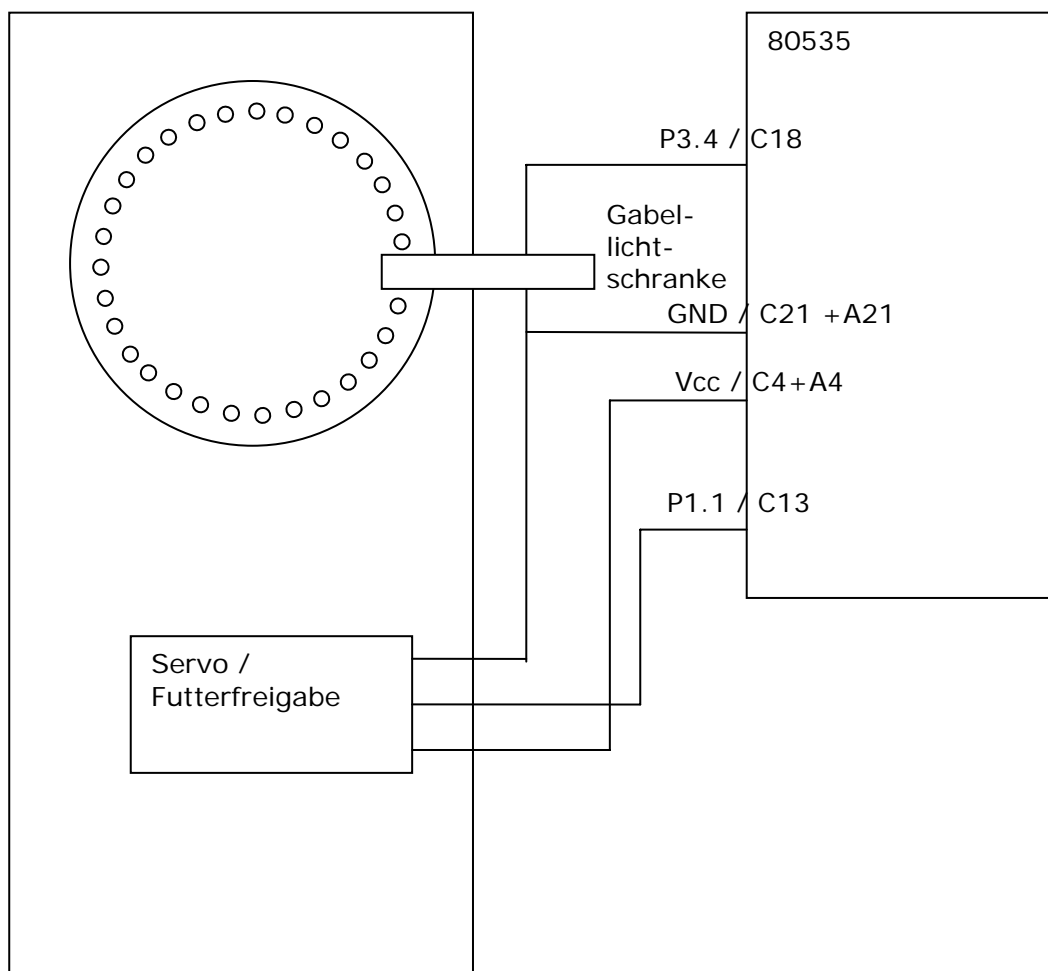


Bild: Messeinrichtung am Mikrocontroller



Bemerkung 1: Im Blockschaltbild müssen +5V und GND nicht vermerkt sein. Für die volle Punktzahl genügt folglich der Anschluß P3.4 / C18 und P1.1 / C13

Bemerkung 2: In der Vorlesung wurde das Programm: Schrauben zählen behandelt. Durch Variation dieses Programmes konnte die Aufgabe sehr leicht gelöst werden.

```
$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 (HAM.A51)
;-----
;Programmbeschreibung
;-----
;Programm:
;PDL - Was wird gemacht?
;1024 Zaehne= 32 Zaehne / Umdrehung * 32 Umdrehungen
;Anzahl der Zaehne = TL0-TH0
;Anzahl der Umdrehungen in R2 = Anzahl der auftretenden Interrupts von
Timer 0
;Servomotor mit PWM: Periodendauer 20ms
;   S1:0F650H Stellung 1
;   S2:0FD70H Stellung 2
;Lichtschranke 1024 mal unterbrochen?
;R2 mit 32 laden + Timer 0 Reload-Betrieb: Mit HB= 256-32
;Ansteuerung Servomotor Stellung 1 / 2
;DEBUG
;P3.4 komplementieren, Breakpoint auf ISR_T0
;
;
;HARDWARE: Ausgabe an Port 1.1 (VG1/C13). +5V - (VG1/A4) Masse VG1/A21
;oder VG1/C21

;INITIALISIERUNG:
;Timer0 Autoreload, Timer2 PWM, Interrupt EAL, Timer0
;
;Erstellt am: 11. Januar 2007 19:52:12
;Programmiert: Juergen Walter
;
;Verwendete Einspruenge: keine
;
;Verwendete Unterprogramme: keine
;
;
;Verwendete Register und Variable:
;Registerbank(0)
;R2
;
;Kommentar:
;
;Aenderungen:
;Geaendert am: Sonntag, 11. Februar 2007 19:52:12
;
;
;-----
;Initialisierungsteil fuer allgemeine Konstanten
;-----
```



```
Periode_low EQU 0E0H ;legt die Periodendauer fest 20ms
Periode_high EQU 0B1H ;45536 = 0B1E0H
S1_low EQU 050H ;Compare Wert
S1_high EQU 0F6H ;HB
S2_low EQU 070H ;Compare Wert
S2_high EQU 0FDH ;HB

CSEG AT 0H ;Legt absolute Codesegmentadresse auf 0h
jmp INIT
;
;-----
;Interrupt-Vektoren
;-----
ORG 0BH ;Timer 0
call ISR_T0
reti

;-----
;Initialisierungsteil fuer On-Chip Peripherie
;-----
ORG 100H ;Programmstart bei 100H
INIT:
;TIMER 2
mov T2CON,#00010001B ;Timer2 Einstellung:Timer mit f=f(osz)/12
;Mode 0 ->Reload bei Ueberlauf

mov CCEN,#00001000B ;Compare Mode des CC1 - P1.1 freigeben
mov CRCL,#Periode_low ;Low Byte von Compare Reload Capture
mov CRCH,#Periode_high ;High Byte von Compare Reload Capture
;Stellung 1 mit Servomotor anfahren
mov CCL1,#S1_low ;P1.1 Compare Capture Low Byte
mov CCH1,#S1_high ;CC High Byte

; Timer 0 - Reload-Betrieb
mov TH0,#224 ;Timer0 TH0=256-32=224
mov TL0,#224 ;Zaehregister initialisieren
mov TMOD,#00000110b ;Timer0 auf Autoreload Betrieb
;Zaehlertakt c/T#

setb TR0 ;Timer Takt Freischaltung
;Interruptfreigabe
setb EAL ;allgemeine Interruptfreigabe
setb ET0 ;Timer0

mov R2,#32 ;R2 bis 32 zaehlen
;-----
;Programmschleife
;-----

ABFRAGE:
jmp ABFRAGE ;Endlosschleife

ISR_T0:
djnz R2,WEITER ;32 Umdrehungen?
;Stellung 2 mit Servomotor anfahren
mov CCL1,#S2_low ;P1.1 Compare Capture Low Byte
mov CCH1,#S2_high ;CC High Byte

WEITER:
```



ret  
end



2. 8051/80535 Programm „HAM1S.asm“

14 Punkte Σ\_\_\_\_\_

Realisieren Sie mit Hilfe von Timer 2 und dem Register R3 eine Sekunde und komplementieren die Leuchte L3. Verwenden Sie hierzu die Einstellungen des Timer 2 von „HAM.ASM“ und den T2-Interrupt.

- a) Schreiben Sie die PDL
- b) Schreiben Sie das Programm

```
$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 (HAM1S.A51)
;-----
;Programmbeschreibung
;-----
;Programm:
;Es wird mit Timer2 und Interrupt eine Sekunde erzeugt: Timer2
Periodendauer 20ms * 50
;R3 50
;PWM Periodendauer 20ms
;
;
;DEBUG
;
;HARDWARE: Ausgabe an Port 1.1 (VG1/C13). +5V - (VG1/A4) Masse VG1/A21
;oder VG1/C21
;PDL:
;Interrupt T2
;INITIALISIERUNG:
;Timer2 PWM, Interrupt EAL, Timer2
;SCHLEIFE
;ISR_T2
;
;Erstellt am: 11. Januar 2007 19:52:12
;Programmiert: Juergen Walter
;
;Verwendete Einspruenge: keine
;
;Verwendete Unterprogramme: keine
;
;
;Verwendete Register und Variable:
;Registerbank(0)
;R2
;
;Kommentar:
;
;Aenderungen:
;Geaendert am: Sonntag, 11. Februar 2007 19:52:12
;
;
;-----
```



```
;Initialisierungsteil fuer allgemeine Konstanten
;-----
Periode_low      EQU 0E0H          ;legt die Periodendauer fest 20ms
Periode_high     EQU 0B1H          ;45536 = 0B1E0H

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

ORG 2BH
Timer 2
call ISR_T2
reti
;-----
;Initialisierungsteil fuer On-Chip Peripherie
;-----
ORG 100H                ;Programmstart bei 100H
INIT:
;TIMER 2
mov T2CON,#00010001B    ;Timer2 Einstellung:Timer mit
f=f(osz)/12              ;Mode 0 ->Reload bei Ueberlauf
mov CCEN,#00001000B    ;Compare Mode des CC1 - P1.1 freigeben
mov CRCL,#Periode_low  ;Low Byte von Compare Reload Capture
mov CRCH,#Periode_high

;Interruptfreigabe
setb EAL                ;allgemeine Interruptfreigabe
setb ET2                ;spezielle Interruptfreigabe

mov R3,#50              ;50*20ms
;-----
;Programmschleife
;-----

ABFRAGE:
jmp ABFRAGE

ISR_T2:
clr TF2
djnz R3,WEITER          ;50 Interrupts
cpl P3.4
mov R3,#50              ;Wieder 50 Interrupts
WEITER:
ret
end
```







3. 8051/80535 Programm „HAM1S\_S“

11 Punkte Σ \_\_\_

Für die dauernde Überwachung des Hamsters muss zusätzlich jede Sekunde die Summe der zurückgelegten Inkremente (Anzahl der Unterbrechungen der Gabellichtschranke) über die serielle Schnittstelle an das Serial Window von Keil gesendet werden. Hierzu können ein Byte für die Anzahl der Umdrehungen und ein Byte für die Anzahl der Unterbrechungen während der Umdrehung gesendet werden.

- a) (1P) Wie berechnen Sie aus dem Wert von Timer 0 die Anzahl der Unterbrechungen.
- b) (4P) Schreiben Sie die PDL
- c) (6P) Schreiben Sie das Programm

```
$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 (HAM.A51)
;-----
;Programmbeschreibung
;-----
;Programm:
;PDL - Was wird gemacht?
;Periodendauer 20ms
;   S1:0F650H Stellung 1
;   S2:0FD70H  Stellung 2
;Lichtschranke 1024 mal unterbrochen?
;R2 mit 32 laden + Timer 0 Reload-Betrieb: Mit HB= 256-32=224=E0h
;Ansteuerung Servomotor Stellung 1 / 2
;Jede Sekunde Anzahl Zaehne, Anzahl Umdrehungen senden
;Sekunde mit Timer 2 50*20ms=1 Sekunde
;DEBUG
;P3.4 komplementieren, Breakpoint auf ISR_T0
;
;HARDWARE: Ausgabe an Port 1.1 (VG1/C13). +5V - (VG1/A4) Masse VG1/A21
;oder VG1/C21

;INITIALISIERUNG:
;Timer0 Autoreload, Timer2 PWM + Interrupt, Interrupt EAL, Timer0
;Serielle Schnittstelle programmieren, 9600 Bps, 8-BIT UART
;Erstellt am: 11. Januar 2007 19:52:12
;Programmiert: Juergen Walter
;
;Verwendete Einspruenge: keine
;
;Verwendete Unterprogramme: keine
;
;
;Verwendete Register und Variable:
;Registerbank(0)
;R2
;
;Kommentar:
;
;Aenderungen:
;Geaendert am: Sonntag, 11. Februar 2007 19:52:12
```



```
;
;
;-----
;Initialisierungsteil fuer allgemeine Konstanten
;-----
Periode_low      EQU 0E0H      ;legt die Periodendauer fest 20ms
Periode_high     EQU 0B1H      ;45536 = 0B1E0H
S1_low           EQU 050H      ;Compare Wert
S1_high          EQU 0F6H      ;HB
S2_low           EQU 070H      ;Compare Wert
S2_high          EQU 0FDH      ;HB

CSEG AT 0H                          ;Legt absolute Codesegmentadresse auf 0h
jmp INIT
;
;-----
;Interrupt-Vektoren
;-----
ORG 0BH                          ;Timer 0
call ISR_T0
reti

ORG 02BH                          ;Timer 2
clr TF2                            ;Interrupt bestaetigen
call ISR_T2
reti

;-----
;Initialisierungsteil fuer On-Chip Peripherie
;-----
ORG 100H                          ;Programmstart bei 100H
INIT:
;TIMER 2
mov  T2CON,#00010001B              ;Timer2 Einstellung:Timer mit f=f(osz)/12
                                       ;Mode 0 ->Reload bei Ueberlauf
mov  CCEN,#00001000B              ;Compare Mode des CC1 - P1.1 freigeben
mov  CRCL,#Periode_low             ;Low Byte von Compare Reload Capture
mov  CRCH,#Periode_high            ;High Byte von Compare Reload Capture
                                       ;Stellung 1 mit Servomotor anfahren
mov  CCL1,#S1_low                  ;P1.1 Compare Capture Low Byte
mov  CCH1,#S1_high                 ;CC High Byte

; Timer 0 - Reload-Betrieb
mov  TH0,#224                      ;Timer0      TH0=256-32=224
mov  TL0,#224                      ;Zaehregister initialisieren
mov  TMOD,#00000110b              ;Timer0      auf Autoreload Betrieb
                                       ;Zaehlertakt  c/T#

setb TR0                            ;Timer Takt Freischaltung
;serielle Schnittstelle
mov  A,PCON                        ;Power Saving modes
orl  A,#10000000b                  ;SMOD 1 9600 Bps
mov  PCON,A
setb SM1                            ;8-BIT UART
setb REN                            ;Freigabe Empfaenger
setb BD                            ;interner Baudratengenerator einschalten

;Interruptfreigabe
```



```
setb EAL                ;allgemeine Interruptfreigabe
setb ET0                ;Timer0
setb ET2                ;Freigabe Timer 2

mov R2,#32              ;R2 Anzahl Umdrehungen bis 32 zaehlen
mov R3,#50              ;50*20ms = 1 Sekunde
;-----
;Programmschleife
;-----

ABFRAGE:
jmp ABFRAGE            ;Endlosschleife

ISR_T2:
djnz R3,WEITER_T2     ;Bereits 1 Sekunde?
cpl P3.3              ;Port 3.4 ist von Aufgabe 1 belegt
mov R3,#50            ;Neue Sekunde beginnt
mov A,TL0             ;Counterstand
subb A,TH0            ;Anzahl Zaehne in Akku
mov SBUF,A            ;Anzahl der Zaehne an SBUF
jnb TI,$              ;Ist das Byte Anzahl Zaehne weg?
clr TI                ;Byte Zaehne ist weg
mov A,#32             ;Zahl der Umdrehungen
subb A,R2             ;Anzahl der geleisteten Umdrehungen
mov SBUF,A            ;Anzahl der Umdrehungen
jnb TI,$              ;Byte Umdrehungen ist weg
clr TI

WEITER_T2:
ret

ISR_T0:
djnz R2,WEITER        ;32 Umdrehungen?
;Stellung 2 mit Servomotor anfahren
mov CCL1,#S2_low      ;P1.1 Compare Capture Low Byte
mov CCH1,#S2_high     ;CC High Byte
clr EAL               ;Stop - Alle Interrupts sperren
clr P3.2              ;Hamster hatte Futter ;-)
```

Bemerkung: In Keil lässt sich das Fenster in HEX umschalten. Das Format ASCII war nicht gefordert.



**4. 8051/80535 Prozessorauslastung**

**10 Punkte Σ\_\_\_\_\_**

Berechnen Sie die Prozessorauslastung für die Messung und die Übertragung der Werte.

Annahme: Ein Hochleistungshamster läuft mit konstanter Geschwindigkeit und benötigt 32s, um das begehrte Futter zu erhalten.

**Lösung:**

**pro Sekunde:**

$$32 * \text{ISR\_T0} + 50 * \text{ISR\_T2 kurz} + 1 * \text{ISR\_T2 lang} = \\ 320\mu\text{s} + 500\mu\text{s} + 2010\mu\text{s} = 2830\mu\text{s} = 2,83\text{ms} = 2,83 \text{ Promille}$$

**Auslastung: ca. 0,283 %**

$$32 * \text{ISR\_T0} = 32 * 5 \text{ Befehle} * 2\mu\text{s} = 320\mu\text{s}$$

$$50 * \text{ISR\_T2 kurz} = 50 * 5 \text{ Befehle} * 2\mu\text{s} = 500\mu\text{s}$$

$$\text{ISR\_T2lang} = \text{ISR\_T2 kurz} + 2 \text{ Byte über serielle Schnittstelle} = 10\mu\text{s} + 2\text{ms} = 2010\mu\text{s}$$

**Bemerkung: Bei der Korrektur wurde die Abschätzung bewertet:**

$$0,1\% < \text{Auslastung} < 0,5\%$$



