



Prüfung: μ-Computertechnik - Bachelor
Termin: Montag, 15.02.2010; 11:30-13:00
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	14	
2	18	
3	18	
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.



Überblick

Am 19. Mai 2010 werden die schnellsten Draisinenläufer gegeneinander antreten. In zwei Wettbewerben werden die Sieger ermittelt. Die Rundenstrecke beträgt: 222m

1. Wettbewerb: Jeweils zwei Läufer starten auf der gegenüberliegenden Seite und fahren 3 Runden.
2. Wettbewerb: Drei Läufer starten gleichzeitig miteinander und fahren die Badische Meile mit 8,888 km.

Die Rundenzeiten der Läufer werden mit einem VC_2 gemessen und sollen auf einem Bildschirm angezeigt werden.

Für den ersten Schritt der Entwicklung machen Sie folgende Annahmen, die für die Aufgaben grundlegend sind:

1. Es ist jeweils nur 1 Fahrer am Start.
2. Eine Kontaktmatte mit Elektronik liefert ein Signal: 5V = keine Person 0V = eine Person
3. Der Draisinenfahrer startet unmittelbar vor der Matte und legt 3 / 40 Runden zurück. Die maximale Geschwindigkeit liegt bei 25km/h die minimalste Geschwindigkeit bei 12km/h.
4. Für die Zeitmessung wird die PCA-Einheit des Controllers benutzt.

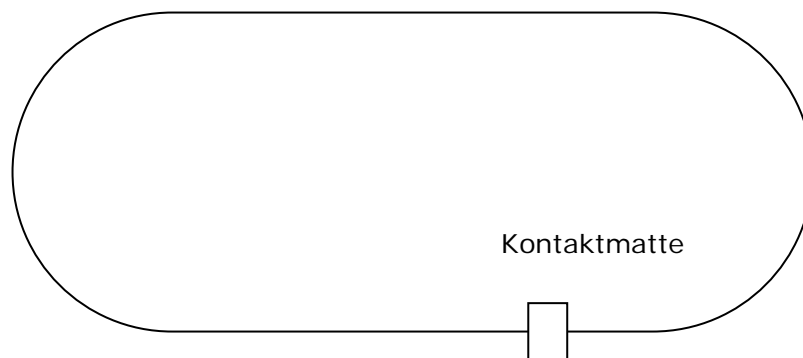


Abb.: Skizze der Strecke



Aufgabe 1a

Berechnen Sie die benötigte Bitbreite für die Zeitmessung von 40 Runden bei einer Auflösung von $\Delta t = 1/3\text{MHz}$

Lösung Aufgabe 1a (4P)

40 Runden – geringste Geschwindigkeit 12 km/h Tgesamt=?
Tgesamt = $s/v = 40 \cdot 222\text{m} / (12000\text{m} / 3600\text{s}) = 2664\text{s} = \mathbf{44,4 \text{ Minuten}}$

Größte Zahl: $2664 / 333\text{ns} = 8.000.000.000$
= 1 1101 1100 1101 0110 0101 0000 0000 0000 → maximale Bitbreite = **33 Bit**

Aufgabe 1b

Mit welcher Einstellung für Δt müssen Sie den PCA Timebase Input bei 40 Runden betreiben, damit Sie die höchste Auflösung bei 4 Bytes Breite bieten?

Lösung Aufgabe 1b (4P)

Bereits eine Auflösung von 666ns würde die Anforderung erfüllen. 33 Bit → 32 Bit aber die Timebase Input kann nur $\text{SYSCLK} / 4 = 750\,000\text{ Hz} = 750\text{ kHz} = 1,3333\text{... } \mu\text{s}$

Aufgabe 1c

Zeichnen Sie das Schaltbild für eine Zusatzplatine zu VC_2, um das Signal der Kontaktmatte anzubinden. Die Kontaktmatte wird über einen Pfostenstecker und einen Schmitt-Trigger an P1.1 angeschlossen.

Lösung Aufgabe 1c (6P)

Der Schaltplan lässt sich sehr einfach aus Lösung Stoppuhr_A1 erzeugen.

Auch nur ein Pfostenstecker an P1.1 und GND würde die Aufgabe bereits lösen, da laut Vorgabe der Pegel: 5V und 0V



Aufgabe 2

Schreiben und testen Sie das Programm „DRAIS_2A“:

Mit T3 wird die Messung in den Zustand „warten“ versetzt. Beim Überfahren der Kontaktmatte wird der PCA-Timer gestartet. Das Ergebnis der Zeitmessung steht nach drei Runden folgendermaßen im Speicher:

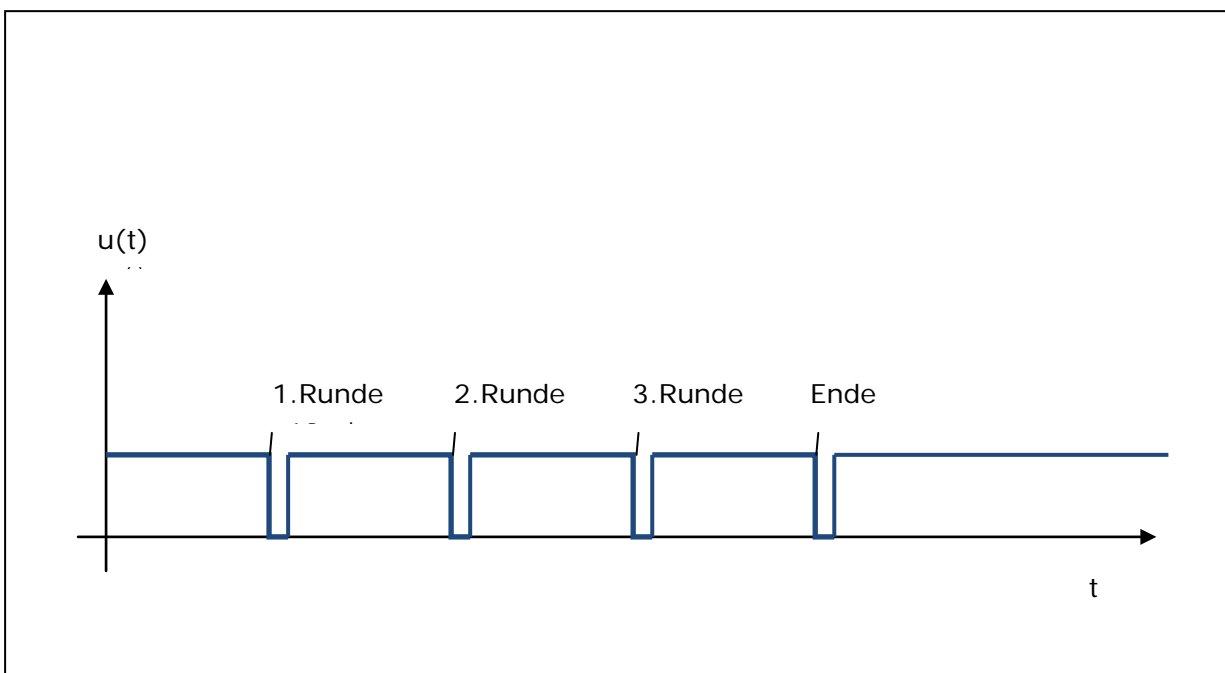
Rundenmessungen

X:0x0000		Fahrer-Nummer
X:0x0001		Runde 1
X:0x0002		0.-Byte Zeit LB
X:0x0003		1.-Byte Zeit
X:0x0004		2.-Byte Zeit
X:0x0005		3.-Byte Zeit HB
X:0x0006		Fahrer-Nummer
X:0x0007		Runde 2
X:0x0008		0.-Byte Zeit LB
X:0x0009		1.-Byte Zeit
X:0x000A		2.-Byte Zeit
X:0x000B		3.-Byte Zeit HB
X:0x000C		Fahrer-Nummer
X:0x000D		Runde 3
X:0x000E		0.-Byte Zeit LB
X:0x000F		1.-Byte Zeit
X:0x0010		2.-Byte Zeit
X:0x0011		3.-Byte Zeit HB

Mit T3 wird der Speicher gelöscht und eine neue Messung gestartet. Die Bedeutung der LEDs ist: L1 an – Messung gestartet; L2 an – Messung läuft; L3 an – Messung beendet.

Lösung Aufgabe 2 (18P)

Signalzeitdiagramm:





Grundüberlegung

Das Programm kann aus zwei bekannten Programmen gewonnen werden.

- 1. Aus PRELL_T1
 - a. Nur Fallende Flanke ist wirksam
 - b. Zusätzlich müssen die Interrupts gezählt werden
- 2. Aus STOPPUHR_A3
 - a. T1 nur fallende Flanke wirksam
 - b. Weiteres Zählregister Runde notwendig
 - c. PCA Timebase Input = SYSCLK/4

```
$NOMOD51 ;der Modus fuer 8051 wird abgeschaltet
$debug
$no1ist ;es wird kein Listing fuer reg535 erstellt
#include(C8051F340.inc) ;die C8051F340-spezifischen Daten
$list ;es wird ein Listing erstellt
$title (DRAIS_2A.A51)
;-----
;Programmbeschreibung
;-----
; PDL:
; Programm: DRAIS_2A
; HARDWARE:
; T1 liegt an CEX0
; Es wird die Zeit zwischen vier Tastendrücken von T1 gemessen.
; T3 Ein neuer Messzyklus kann gestartet werden.
; Hardware: VC_2
; Sysclock: 3 MHz
;
; 1.Runde 2.Runde 3.Runde STOPP
; T1 -----|_|-----|_|-----|_|-----|_|-----
;
; Die fallende Flanke von T1 startet die Zaehlung
; Der PCA-Interrupt zaehlt mit den Registern 3,2 die Anzahl der Ueberlaufe
; Sysclk 3 MHz
; Zeit Runde 1 - 3 = PCA0_T1 - PCA0_T2 + R3,R2-1
; -1 da bereits beim Start das Register R2 um 1 erhoehrt wird
;
; ZeitMAX=< 2^n n=32 Bit --> 2^32*(1/750kHz) = ca. 5727 s = 95,4 min= 1h 35,4 Min
; Die Zeitberechnung:
;
; Danach wird der PCA0-Zahler gestoppt
;
; Die Anzahl der Überlaeufe wird über Interrupt
; gezaehlt und in R2,R3 gespeichert.
; SPEICHERUNG: der Werte im externen Speicher

;X:0x0000: 0.-Byte Fahrer Nr.
;X:0x0001: 1.-Byte Runde Nr.
;X:0x0002: 0.-Byte Zaehlerstand
;X:0x0003: 1.-Byte Zaehlerstand
;X:0x0004: 2.-Byte Zaehlerstand
;X:0x0005: 3.-Byte Zaehlerstand
;.....
;X:0x0010: 2.-Byte Zaehlerstand
;X:0x0011: 3.-Byte Zaehlerstand
;
; PDL
; Initialisierung mit Configuration Wizard;
; Drei Zustaende
; Auf Messung Warten F0=0
; Messung laeuft F0=1
; Messung stoppen
```



```
;
;
;Erstellt am: Samstag, 23. Januar 2010 12:43:43
;Programmiert: Juergen Walter
;
;Verwendete Einspruenge: keine
;
;Verwendete Unterprogramme: keine
;
;
;Verwendete Register und Variable:
;Registerbank(0)
;PCA0L=LB
;PCA0H=HB
;R2=LB
;R3=HB
;4-Byte-Wert Zeit=PCA0L - PCA0H - R2 - R3
;R4 = Rundenzaehler 1-3 - Impulszaehler = R4=4
;Kommentar:
;
;Aenderungen:
;Geaendert am: Samstag, 23. Januar 2010 12:43:43
;-----
;Initialisierungsteil fuer allgemeine Konstanten
;-----

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

ORG 05BH
call ISR_PCA0
RETI

;-----
;Initialisierungsteil fuer On-Chip Peripherie
;-----

ORG 100H                  ;Programmstart bei 100H
INIT:
call Init_Device         ;Aufruf zur Initialisierung der Controller Funktionen
call X_LOESCH            ;Speicherzellen X:0x0000 - X:0x00011 werden auf 0 gesetzt
mov DPTR,#0h            ;Datenpointer auf 0000h
mov R2,#0               ;LB Interrupt-Zaehler
mov R3,#0               ;HB Interrupt-Zaehler
mov R4,#0               ;Runden Nr
mov R5,#3               ;Zaehlregister fuer die Rundenanzahl
clr F0                  ;Messung laeuft --> F0=1
clr P3.2                ;Leuchtdioden
setb P3.3
setb P3.4

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

ABFRAGE:
jnb P1.3, INIT          ;Warteschleife
jmp ABFRAGE            ;Nach ABFRAGE

;-----
;PCA-Interrupt
;-----

ISR_PCA0:
;Welcher Interrupt? 1. Start 2. Ueberlauf 3. Kontaktmatte
```



```
jnb F0,UEBERLAEUFE_PCA0 ;F0 = 0 falls T1 nicht zuvor betaetigt F0=1 Messung laeuft
jnb CCF0,NUR_SCHARF ;T1 betaetigt? War das der Beginn der Messung?
```

```
; 1. START
setb CR ;PCA-Timer laeuft
setb F0 ;Flag F0 Messung laeuft
setb P3.2 ;Leuchtdioden
clr P3.3
setb P3.4 ;MESSUNG LAEUFT
clr CCF0 ;Fallende Flanke T1 bestaetigt
RET
```

```
;2. UEBERLAUF
UEBERLAEUFE_PCA0:
jnb CCF0,WERT_WRn ;Falls fallende Flanke T1
clr CF ;Ueberlauf Interrupt erkannt
call UEBERLAEUFE_ZAEHLEN
jmp NUR_SCHARF
```

```
;3. KONTAKTMATTE
WERT_WRn:
clr CCF0 ;Fallende Flanke T1 bestaetigen
call WR_ZAEHLERSTAND ;
jnb CF,NUR_SCHARF ;ist auch Ueberlauf PCA0 aufgetreten?
clr CF ;Falls zwischenzeitlich oder vorher aufgetreten
call UEBERLAEUFE_ZAEHLEN
```

```
NUR_SCHARF:
ret
```

```
UEBERLAEUFE_ZAEHLEN:
clr C ;Carry null setzen
mov A,R2 ;LB in Akku
add A,#1 ;1 addieren
mov R2,A ;LB in R2 zurueck
jnc KEIN_UEBERTRAG ;R3 erhaelt keinen Uebertrag
inc R3 ;Uebertrag trat auf
KEIN_UEBERTRAG:
RET
```

```
WR_ZAEHLERSTAND:
clr CCF0 ;
mov A,#1 ;Fahrer 1
movx @DPTR,A ;0000H - Fahrer 1
inc DPTR ;
inc R4 ;0001H in R4 die Runden-Nr.
mov A,R4
movx @DPTR,A ;
inc DPTR
mov A,PCA0CPL0 ;0002H - PCA0L
movx @DPTR,A ;
inc DPTR
mov A,PCA0CPH0 ;0003H - PCA0H
movx @DPTR,A ;
inc DPTR ;DPTR erhoehen
mov A,R2 ;0004H - R2
movx @DPTR,A ;
inc DPTR
mov A,R3 ;0005H - R3
movx @DPTR,A ;
inc DPTR
djnz R5,MESSUNG_NB ;Messung nicht beendet
clr CR ;PCA0-Zaehler STOPP

setb P3.2 ;MESSUNG BEENDET
setb P3.3
```




clr P3.4

MESSUNG_NB:
RET

```
X_LOESCH:
mov A,#0 ;mit 0 wird ueberschrieben
mov DPTR,#0 ;Datenpointer
mov R2,#12H ;Anzahl der Werte - 11H Damit ersichtlich
SCHLEIFE_X:
movx @DPTR,A ; X:0x0000
inc DPTR
djnz R2,SCHLEIFE_X ;Bereits 12h Werte?
ret
```

```
;-----
;- Generated Initialization File --
;-----
```

```
; Peripheral specific initialization functions,
; Called from the Init_Device label
PCA_Init:
    anl  PCA0MD,    #0BFh
    mov  PCA0MD,    #003h
    mov  PCA0CPM0,  #011h ;PCA Timebase SYSCLK/4
    ret
```

```
Port_IO_Init:
; P0.0 - Skipped,      Open-Drain, Digital
; P0.1 - Skipped,      Open-Drain, Digital
; P0.2 - Skipped,      Open-Drain, Digital
; P0.3 - Skipped,      Open-Drain, Digital
; P0.4 - Skipped,      Open-Drain, Digital
; P0.5 - Skipped,      Open-Drain, Digital
; P0.6 - Skipped,      Open-Drain, Digital
; P0.7 - Skipped,      Open-Drain, Digital

; P1.0 - Skipped,      Open-Drain, Digital
; P1.1 - CEX0 (PCA),   Push-Pull,  Digital
; P1.2 - Unassigned,   Push-Pull,  Digital
; P1.3 - Unassigned,   Push-Pull,  Digital
; P1.4 - Unassigned,   Open-Drain, Digital
; P1.5 - Unassigned,   Open-Drain, Digital
; P1.6 - Unassigned,   Open-Drain, Digital
; P1.7 - Unassigned,   Open-Drain, Digital

; P2.0 - Unassigned,   Open-Drain, Digital
; P2.1 - Unassigned,   Open-Drain, Digital
; P2.2 - Unassigned,   Open-Drain, Digital
; P2.3 - Unassigned,   Open-Drain, Digital
; P2.4 - Unassigned,   Open-Drain, Digital
; P2.5 - Unassigned,   Open-Drain, Digital
; P2.6 - Unassigned,   Open-Drain, Digital
; P2.7 - Unassigned,   Open-Drain, Digital

; P3.0 - Unassigned,   Open-Drain, Digital
; P3.1 - Unassigned,   Open-Drain, Digital
; P3.2 - Unassigned,   Open-Drain, Digital
; P3.3 - Unassigned,   Open-Drain, Digital
; P3.4 - Unassigned,   Open-Drain, Digital
; P3.5 - Unassigned,   Open-Drain, Digital
; P3.6 - Unassigned,   Open-Drain, Digital
; P3.7 - Unassigned,   Open-Drain, Digital
```



```
mov P1MDOUT, #00Eh
mov POSKIP, #0FFh
mov P1SKIP, #001h
mov XBR1, #041h
ret

Oscillator_Init:
mov OSCICN, #081h ;SYSCLK 3MHz
ret

Interrupts_Init:
mov EIE1, #010h
mov IE, #080h
ret

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

end
```

Aufgabe 3

Um die Messung gegen Fehlmessungen zu sichern wird:

1. die fallende Flanke nur dann akzeptiert, wenn nach ca. 10ms der Pegel weiterhin auf LOW liegt. Verwenden Sie hierzu Timer 2.
2. nach jeder Messung des Fahrers ca. 4s - 6s gewartet und erst dann wieder eine neue Messung ermöglicht. Verwenden Sie hierzu die PCA-Einheit und ein Register.

Aufgabe 3a)

Schreiben Sie das Unterprogramm für Aufgabe 3 Punkt 1.

Aufgabe 3b)

Schreiben Sie das Unterprogramm für Aufgabe 3 Punkt 2.

Aufgabe 3c)

Schreiben Sie das Programm „DRAIS_3C“ indem Sie die Unterprogramme in Aufgabe 2 einsetzen.

Lösung Aufgabe 3a (6P)

Timer 2:

Takt SYSCLK → $10\text{ms}/333\text{ns}=30.030 \rightarrow 65.536-30.030=35506=8AB2\text{H}$

TIMER 2 wird mit 8AB2H geladen und sobald fallende Flanke

Lösung Aufgabe 3b (6P)

PCA hat alle 22ms einen Überlauf → $5\text{s}/22\text{ms}\approx 228 \rightarrow$ Register R6 zählt Überläufe und setzt das Bit 20H.0 nach Überlauf.



Lösung Aufgabe 3c (6P)

3a und 3b sind in das Programm eingebunden:

```
$NOMOD51 ;der Modus fuer 8051 wird abgeschaltet
$debug
$no1ist ;es wird kein Listing fuer reg535 erstellt
#include(C8051F340.inc) ;die C8051F340-spezifischen Daten
$list ;es wird ein Listing erstellt
$title (DRAIS_3C.A51)
;-----
;Programmbeschreibung
;-----
; PDL:
; Programm: DRAIS_2A + 10ms Entprellen mit Timer 2 + 5s warten nach Fahrer
; HARDWARE:
; T1 liegt an CEX0
; Es wird die Zeit zwischen vier Tastendrücken von T1 gemessen.
; T3 Ein neuer Messzyklus kann gestartet werden.
; Hardware: VC_2
; Sysclock: 3 MHz
;
; 1.Runde 2.Runde 3.Runde STOPP
; T1 -----|_|-----|_|-----|_|-----|_|-----
;
; Die fallende Flanke von T1 startet die Zaehlung
; Der PCA-Interrupt zaehlt mit den Registern 3,2 die Anzahl der Ueberlaufe
; Sysclk 3 MHz
; Zeit Runde 1 - 3 = PCA0_T1 - PCA0_T2 + R3,R2-1
; -1 da bereits beim Start das Register R2 um 1 erhoehrt wird
;
; ZeitMAX=< 2^n n=32 Bit --> 2^32*(1/750kHz) = ca. 5727 s = 95,4 min= 1h 35,4 Min
; Die Zeitberechnung:
;
; Danach wird der PCA0-Zaehler gestoppt
;
; Die Anzahl der Überlaeufe wird über Interrupt
; gezaehlt und in R2,R3 gespeichert.
; SPEICHERUNG: der Werte im externen Speicher

;X:0x0000: 0.-Byte Fahrer Nr.
;X:0x0001: 1.-Byte Runde Nr.
;X:0x0002: 0.-Byte Zaehlerstand
;X:0x0003: 1.-Byte Zaehlerstand
;X:0x0004: 2.-Byte Zaehlerstand
;X:0x0005: 3.-Byte Zaehlerstand
;.....
;X:0x0010: 2.-Byte Zaehlerstand
;X:0x0011: 3.-Byte Zaehlerstand
;
; PDL
; Initialisierung mit Configuration Wizard;
; Drei Zustaende
; Auf Messung Warten F0=0
; Messung laeuft F0=1
; Messung stoppen
;
;
;Erstellt am: Samstag, 23. Januar 2010 12:43:43
;Programmiert: Juergen Walter
;
;Verwendete Einspruenge: keine
;
;Verwendete Unterprogramme: keine
;
;
;Verwendete Register und Variable:
```



```
;Registerbank(0)
;PCA0L=LB
;PCA0H=HB
;R2=LB
;R3=HB
;4-Byte-Wert Zeit=PCA0L - PCA0H - R2 - R3
;R4 = Rundenzaehler 1-3 - Impulszaehler = R4=4
;Kommentar:
;
;Aenderungen:
;Geaendert am: Samstag, 23. Januar 2010 12:43:43
;-----
;Initialisierungsteil fuer allgemeine Konstanten
;-----

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

ORG 05BH
call ISR_PCA0
RETI

;-----
;Initialisierungsteil fuer On-Chip Peripherie
;-----

ORG 100H                ;Programmstart bei 100H
INIT:
call Init_Device        ;Aufruf zur Initialisierung der Controller Funktionen
call X_LOESCH           ;Speicherzellen X:0x0000 - X:0x00011 werden auf 0 gesetzt
mov DPTR,#0h           ;Datenpointer auf 0000h
mov R2,#0              ;LB Interrupt-Zaehler
mov R3,#0              ;HB Interrupt-Zaehler
mov R4,#0              ;Runden Nr
mov R5,#3              ;Zaehlregister fuer die Rundenanzahl
mov R6,#228            ;=5s/22ms
clr F0                 ;Messung laeuft --> F0=1
clr P3.2               ;Leuchtdioden
setb P3.3
setb P3.4
;-----
;Programmschleife
;-----

ABFRAGE:
jnb P1.3, INIT         ;Warteschleife
jmp ABFRAGE            ;Nach ABFRAGE

;-----
;PCA-Interrupt
;-----

ISR_PCA0:
;Welcher Interrupt? 1. Start 2. Ueberlauf 3. Kontaktmatte
jnb F0,UEBERLAEUFE_PCA0 ;F0 = 0 falls T1 nicht zuvor betaetigt F0=1 Messung laeuft
jnb CCF0,NUR_SCHARF    ;T1 betaetigt? War das der Beginn der Messung?

;ENTPRELLEN
call WARTEN_10MS
jnb P1.1,WARTEN_NOK    ;war keine 10ms betaetigt

; 1. START
setb CR                ;PCA-Timer laeuft
setb F0                ;Flag F0 Messung laeuft
```



```
setb P3.2 ;Leuchtdioden
clr P3.3
setb P3.4 ;MESSUNG LAEUFT
clr CCF0 ;Fallende Flanke T1 bestaetigt
WARTEN_NOK:
RET

;2. UEBERLAUF
UEBERLAEUFE_PCA0:
jb CCF0,WERT_WRn ;Falls fallende Flanke T1
clr CF ;Ueberlauf Interrupt erkannt
djnz R6,NOCH_KEINE5s ;Fuer den Fahrer 5s warten
setb 20H.0 ;20H.0
NOCH_KEINE5s:
call UEBERLAEUFE_ZAEHLEN
jmp NUR_SCHARF

;3. KONTAKTMATTE
WERT_WRn:
clr CCF0 ;Fallende Flanke T1 bestaetigen
call WARTEN_10MS
jb P1.1,NUR_SCHARF ;P1.1 war keine 10ms auf 0
jnb 20H.0,NUR_SCHARF ;noch keine 5s fuer den Fahrer 1 abgelaufen
clr 20H.0 ;Flag für 5s zuruecksetzen
mov R6,#228 ;Um neue 5s zu ueberprüfen
call WR_ZAEHLERSTAND ;
jnb CF,NUR_SCHARF ;ist auch Ueberlauf PCA0 aufgetreten?
clr CF ;Falls zwischenzeitlich oder vorher aufgetreten
call UEBERLAEUFE_ZAEHLEN

NUR_SCHARF:
ret

UEBERLAEUFE_ZAEHLEN:
clr C ;Carry null setzen
mov A,R2 ;LB in Akku
add A,#1 ;1 addieren
mov R2,A ;LB in R2 zurueck
jnc KEIN_UEBERTRAG ;R3 erhaelt keinen Uebertrag
inc R3 ;Uebertrag trat auf
KEIN_UEBERTRAG:
RET

WR_ZAEHLERSTAND:
clr CCF0 ;
mov A,#1 ;Fahrer 1
movx @DPTR,A ;0000H - Fahrer 1
inc DPTR ;
inc R4 ;0001H in R4 die Runden-Nr.
mov A,R4
movx @DPTR,A ;
inc DPTR
mov A,PCA0CPL0 ;0002H - PCA0L
movx @DPTR,A ;
inc DPTR
mov A,PCA0CPH0 ;0003H - PCA0H
movx @DPTR,A ;
inc DPTR ;DPTR erhoehen
mov A,R2 ;0004H - R2
movx @DPTR,A ;
inc DPTR ;
mov A,R3 ;0005H - R3
movx @DPTR,A ;
inc DPTR ;
djnz R5,MESSUNG_NB ;Messung nicht beendet
clr CR ;PCA0-Zaehler STOPP
```



```
setb P3.2 ;MESSUNG BEENDET
setb P3.3
clr P3.4
```

```
MESSUNG_NB:
RET
```

```
WARTEN_10MS:
setb TR2 ;TIMER2 rennt los
jnb TF2H,$ ;10ms vorbei?
clr TR2 ;Timer STOPP
clr TF2H
ret
```

```
X_LOESCH:
mov A,#0 ;mit 0 wird ueberschrieben
mov DPTR,#0 ;Datenpointer
mov R2,#12H ;Anzahl der Werte - 11H Damit ersichtlich
SCHLEIFE_X:
movx @DPTR,A ;
inc DPTR
djnz R2,SCHLEIFE_X ;Bereits 12h Werte?
ret
```

```
;-
;- Generated Initialization File --
;-
```

```
; Peripheral specific initialization functions,
; Called from the Init_Device label
```

```
PCA_Init:
    anl  PCA0MD,    #0BFh
    mov  PCA0MD,    #003h
    mov  PCA0CPM0,  #011h
    ret
```

```
Timer_Init:
    mov  CKCON,     #010h
    mov  TMR2RLH,   #08Ah
    mov  TMR2RLH,   #08Ah
    mov  TMR2L,     #0B2h
    mov  TMR2H,     #08Ah
    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 - Skipped, Push-Pull, Digital
; P0.5 - Skipped, Push-Pull, Digital
; P0.6 - Skipped, Push-Pull, Digital
; P0.7 - Skipped, Push-Pull, Digital

; P1.0 - Skipped, Push-Pull, Digital
; P1.1 - CEX0 (PCA), Push-Pull, Digital
; P1.2 - Unassigned, Push-Pull, Digital
; P1.3 - Unassigned, Push-Pull, Digital
; P1.4 - Unassigned, 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 - Unassigned,   Push-Pull, Digital
; P3.5 - Unassigned,   Push-Pull, Digital
; P3.6 - Unassigned,   Push-Pull, Digital
; P3.7 - Unassigned,   Push-Pull, Digital

mov  P0MDOUT,    #0FFh
mov  P1MDOUT,    #0FFh
mov  P2MDOUT,    #0FFh
mov  P3MDOUT,    #0FFh
mov  P0SKIP,     #0FFh
mov  P1SKIP,     #0E1h
mov  P2SKIP,     #0FFh
mov  P3SKIP,     #00Fh
mov  XBR1,       #041h
ret

Oscillator_Init:
  mov  OSCICN,    #081h
  ret

Interrupts_Init:
  mov  EIE1,      #010h
  mov  IT01CF,    #010h
  mov  IE,        #080h
  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
```

