```
}
/* Ende Gong() ------------------------------------------------------------*/



/* ReadPots( byAD) -------------------------------------------------------*/
/*                                                                        */
/*   Liest den Analogkanal 2 (P1.2) als Wert für die Ablaufgeschwindigkeit  */
/*     und den Analogkanal 3 (P1.3) als Wert für die Lautstärke           */
/*                                                                        */
/*   byAD gibt den Analogkanal an; byAd= [2,3]                            */
/*------------------------------------------------------------------------*/
#define Start_ADC0  0x28
#define Start_ADC1  0x29
#define Start_ADC2  0x2A
#define Start_ADC3  0x2B
#define Start_ADC4  0x2C

void ReadPots( BYTE byAD)
{
   switch( byAD)
   {
     case 2:           /* AD2.....Grundfrequenz einlesen                  */
       ADCON= Start_ADC2;      /* start conversion  Ch2 an P1.2            */
       while( !(ADCON&0x10));  /* auf EOC warten                           */
       wZeitgeber= ADAT+ 120;
      break;

     case 3:           /* AD3.....Lautstärke einlesen                     */
       ADCON= Start_ADC3;      /* start conversion  Ch3 an P1.3            */
       while( !(ADCON&0x10));  /* auf EOC warten                           */
       byAmpPot= ADAT;
      break;

     default:
      break;
   }
}
/* Ende ReadPots() -------------------------------------------------------*/
```

```
;/*  PWMGEN  ---------------------------------------------------------------*/
;/*                                                                        */
;/*  PWM- interrupt routine  für den 8xC752                                */
;/*                                                                        */
;/*       Die PWM- interrupt routine liegt in einem absoluten CodeSegment  */
;/*       an der PWM- Interruptadresse 0x33. Dadurch muß kein Sprung aus-   */
;/*       geführt werden. Der erste Befehl nach einem Interrupt ist das    */
;/*       Nachladen des Pulsbreiteregisters (MOV PWCM, byTmp), d.h. nach    */
;/*       spätestens 5us ist das Register geladen (3us Interrupt, 2us MOV).*/
;/*                                                                        */
;/*       Wird die maximale Pulsbreite auf 42,5us-6us beschränkt (damit    */
;/*       keine Knackgeräusche auftreten), so ist die maximale Amplitude:  */
;/*       byAmplitude(Max) = 220.                                          */
;/*                                                                        */
;/*------------------------------------------------------------------------*/
```

```
$ NOMOD51
$ INCLUDE (REG752.INC)

NAME    PWMGEN

INTR    EQU     33H               ; interrupt Einsprungadresse
EXTRN   DATA   ( byTmp)           ; definiert im C- Programm
EXTRN   DATA   ( byCnt)           ;  -"-
EXTRN   DATA   ( byAmplitude)     ;  -"-
EXTRN   DATA   ( byfh)            ;  -"-
EXTRN   DATA   ( byft)            ;  -"-
EXTRN   DATA   ( wZeitgeber)      ;  -"-
EXTRN   DATA   ( byFlag)          ;  -"-
EXTRN   DATA   ( wCnt)            ;  -"-

PUBLIC  PWMGen                    ;

CSEG   AT INTR                    ; Einsprung Timer2 interrupt
                                  ; Absolutes Segment

PWMGen:
     MOV     PWCM,byTmp           ; CCL= byTmp;   Pulsbreite nachladen
     PUSH    ACC
     PUSH    PSW

     MOV     A,byCnt              ; if( byCnt == byfh)
     CJNE    A,byfh,?LBL1

     MOV     byTmp,#0FFH          ;   byTmp= 255;
     SJMP    ?LBL2
?LBL1:                            ; else
     MOV     A,byCnt
     CJNE    A,byft,?LBL2         ;  if( byCnt == byft)
     CLR     C                    ;  { byTmp= 255- byAmplitude;
     MOV     A,#0FFH
     SUBB    A,byAmplitude
     MOV     byTmp,A
     CLR     A                    ;    byCnt= 0;
     MOV     byCnt,A              ;  }
?LBL2:
     INC     byCnt                ; byCnt++;

     INC     wCnt+01H             ; wCnt++;
     MOV     A,wCnt+01H
     JNZ     ?LBL3
     INC     wCnt
?LBL3:
     CLR     C                    ; if( wCnt >= wZeitgeber)
     MOV     A,wCnt+01H           ; {
     SUBB    A,wZeitgeber+01H
     MOV     A,wCnt
     SUBB    A,wZeitgeber
     JC      ?LBL4
     MOV     byFlag,#01H          ;   byFlag= 1;
     CLR     A                    ;   wCnt= 0;
     MOV     wCnt,A
     MOV     wCnt+01H,A           ; }
?LBL4:
     POP     PSW
     POP     ACC
     RETI

END
```

□

Ergänzung zum folgenden Beitrag *Rund um das INTEL-HEX-Format*

Interrupts mit Dualboot

Anwendung für HEXPAT.EXE

Die Interruptvektoren des FSD51 enthalten einen Sprung auf die Adressen `A0xx`, sodaß auch eigene Interruptvektoren eingesetzt werden können, sofern sie entweder händisch oder durch das Anwenderprogramm auf die Adressen `A0xx` geschrieben werden.

Der C-Compiler von KEIL generiert selbsttätig Interruptvektoren auf die Adressen `00xx`. Die neuen Versionen ab Versionsnummer 3.4 können die Interruptvektoren mit der `#pragma`-Anweisung `IV (0xaaaa)` auch auf die Adresse `aaaaH` parallelverschieben.

Bei Verwendung älterer Versionen kann man sich mit einem Patch der HEX-Datei behelfen.

Annahme:

Man bearbeitet ein Programm `TESTI` und erhält durch

```
OHS51 TESTI.ABS
```

die HEX-Datei `TESTI.HEX`.

Durch Anwendung des HEX-Patchers `HEXPAT`, können alle Ladeadressen eines bestimmten Bereichs auf einen anderen Bereich verschoben werden:

```
HEXPAT TESTI A0 00
```

Das Programm `HEXPAT` erzeugt `TESTI.HEY`, die gepatchte HEX-Datei.

Man ruft den Debugger auf:

```
TS51 TESTI.ABS INIT (TESTI.INI)
```

Die Datei `TESTI.INI` enthält unter anderem die Zeile zum Nachladen der gepatchten HEX-Datei.

```
...
LOAD <pfad>TESTI.HEY
```

Die Datei TESTI wird dabei eigentlich zweimal geladen: zuerst beim Aufruf des Debugger in der Kommandozeile und dann in der INI-Datei. Beim ersten Laden über die Kommandozeile sind die Interruptvektoren noch auf 00xx ausgerichtet. Glücklicherweise meldet der TS51 dabei keinen Fehler. In diesem Ladevorgang werden auch alle Symbolinformationen im TS51 bekanntgegeben. Der zweite Ladevorgang ist eigentlich redundant mit Ausnahme eben der verschobenen Interruptvektoren. □

```
For those who believe in Winter Wonderland's, the Rockies present
excitement galore:


            *
             \
              \
               |\
              \ | \ (__)
              \\|| \(oo)
               \||\ \\/
                 ^^ \||
                   \\ ||
                    \\||
                     \||
                      ^^
                       \\_
                        \_
   Cow skiing a Black Diamond at Aspen


                     \ (__)
                     \\(oo)
                   /-----\\\/
                  / |    (##)
                 *  ||----||"
                    ^^    ^^
    This cow plays bagpipes.
```