

3.1. Melodiegong

Zur Erzeugung von Tonfrequenzen kann beim 68HC11 mit einem Timer die interne Taktfrequenz von 2MHz auf die entsprechende Tonfrequenz heruntergeteilt werden. Dazu stellt man den Timer so ein, daß der Ausgang bei jedem Nulldurchgang des Timers invertiert wird. Im Timer muß also ein Wert geladen werden, der der halben Periodendauer der Tonfrequenz entspricht. Für den Kammerton A1=440Hz muß der Teiler mit $2000000/880 = 2272,7272$ geladen werden. Da der Teiler nur ganzzahlige Werte verarbeiten kann, wird dieser Wert auf 2273 aufgerundet. Da in der Tonleiter eine Oktave genau einer Verdopplung oder Halbierung der Frequenz entspricht, muß für A0 4546 und für A 9092 geladen werden. Die übrigen 11 Töne einer Oktave kann man dann berechnen, da der Logarithmus der Tonfrequenzen einen konstanten Abstand besitzt. Der logarithmische Abstand zwischen zwei Tönen ist also gleich $1/12$ des Abstandes einer Oktave. Der lg von 9092 beträgt 3,958659 , von 4546 ist der lg 3,657629 , und von 2273 ist der lg 3,356599. Der Oktavabstand des lg beträgt also:

$$\begin{aligned} 3,958659 - 3,657629 &= 3,657629 - 3,356599 \\ &= 0,30103 \end{aligned}$$

Nimmt man davon $1/12$ $0,30103/12 = 0,0250858$, so ist dieser Wert nun 12 mal von 3,958659 abzuziehen, und man erhält die Logarithmen der 12 Teiler einer Oktave. Die Teiler erhält man nun, indem 10 mit diesen Werten potenziert wird. Für die Frequenztafel des Programms müssen dann die Teiler aus der im Anschluß folgenden Tabelle verwendet werden. Diese Werte können dann noch durch einen Faktor von 2, 4 oder 8 geteilt werden. Damit kann die Tonlage um 3 Oktaven nach oben verschoben werden.

Nr.	Ton	Teiler	lg des Teilers	berechneter Teiler
1	A	9092	3,958659	9092
2	Ais		3,933573	8582
3	H		3,908487	8100
4	C0		3,883402	7645
5	Cis0		3,858316	7216
6	D0		3,833230	6811
7	Dis0		3,808144	6429
8	E0		3,783059	6068
9	F0		3,757973	5728
10	Fis0		3,732887	5406
11	G0		3,707801	5103
12	Gis0		3,682715	4816
13	A0	4546	3,657629	4546
14	Ais0		3,632544	4291
15	H0		3,607458	4050
16	C1		3,582372	3823
17	Cis1		3,557286	3608
18	D1		3,532200	3406
19	Dis1		3,507114	3215
20	E1		3,482029	3034
21	F1		3,456943	2864
22	Fis1		3,431857	2703
23	G1		3,406771	2551
24	Gis1		3,381685	2408
25	A1	2273	3,356599	2273
26	Ais1		3,331514	2145
27	H1		3,306428	2025
28	C2		3,281342	1911

Bei einem Teiler von 2 = Default ergeben sich dann folgende Kodierungen für die Noten im Programm, wenn die Nummer der Note in den oberen 5 Bit und die Zeit in den unteren 3 Bit verschlüsselt werden. Die Pausenfrequenz wird durch den Teiler nicht verändert. Sie bleibt für alle Teiler bei 31250 Hz.

Nr	Note	1	3	1	3	1	3	1	3	Frequenz [Hz]	HC11 (Hex)	
		16	32	8	16	4	8	2	4			
0	Pause	00	01	02	03	04	05	06	07	31250,00	0064	0040
1	a0	08	09	0A	0B	0C	0D	0E	0F	220,00	4546	11C2
2	ais0	10	11	12	13	14	15	16	17	233,08	4291	10C3
3	h0	18	19	1A	1B	1C	1D	1E	1F	246,94	4050	0FD2
4	c1	20	21	22	23	24	25	26	27	261,63	3822	0EEE
5	cis1	28	29	2A	2B	2C	2D	2E	2F	277,18	3608	0E18
6	d1	30	31	32	33	34	35	36	37	293,66	3405	0D4D
7	dis1	38	39	3A	3B	3C	3D	3E	3F	311,13	3214	0C8E
8	e1	40	41	42	43	44	45	46	47	329,63	3034	0BDA
9	f1	48	49	4A	4B	4C	4D	4E	4F	349,23	2864	0B30
10	fis1	50	51	52	53	54	55	56	57	369,99	2703	0A8F
11	g1	58	59	5A	5B	5C	5D	5E	5F	391,99	2551	09F7
12	gis1	60	61	62	63	64	65	66	67	415,30	2408	0968
13	a1	68	69	6A	6B	6C	6D	6E	6F	440,00	2273	08E1 *
14	ais1	70	71	72	73	74	75	76	77	466,16	2145	0861
15	h1	78	79	7A	7B	7C	7D	7E	7F	493,88	2025	07E9
16	c2	80	81	82	83	84	85	86	87	523,25	1911	0777
17	cis2	88	89	8A	8B	8C	8D	8E	8F	554,36	1804	070C
18	d2	90	91	92	93	94	95	96	97	587,33	1703	06A7
19	dis2	98	99	9A	9B	9C	9D	9E	9F	622,25	1607	0647
20	e2	A0	A1	A2	A3	A4	A5	A6	A7	659,25	1517	05ED
21	f2	A8	A9	AA	AB	AC	AD	AE	AF	698,46	1432	0598
22	fis2	B0	B1	B2	B3	B4	B5	B6	B7	739,99	1351	0547
23	g2	B8	B9	BA	BB	BC	BD	BE	BF	783,99	1275	04FB
24	gis2	C0	C1	C2	C3	C4	C5	C6	C7	830,61	1204	04B4
25	a2	C8	C9	CA	CB	CC	CD	CE	CF	880,00	1136	0470
26	ais2	D0	D1	D2	D3	D4	D5	D6	D7	932,33	1072	0430
27	h2	D8	D9	DA	DB	DC	DD	DE	DF	987,76	1012	03F4
28	c3	E0	E1	E2	E3	E4	E5	E6	E7	1046,50	955	03BB

Für die Notenummer haben dann die Kodierungen von 29 bis 31 besondere Bedeutung. Mit 29 wird ein Kommando zur Einstellung des Faktors für die Frequenztabelle und zum Ein-/Aus-schalten des Stacatoeffektes kodiert. In den unteren beiden Bits steht dann der Exponent des Faktors mit der Basis 2 für die Frequenzen aus Tabelle 1, und das 3. Bit schaltet den Stacatoeffekt ein (1) oder aus (0). Ein Kommando hat die Länge einer 1/16 Note und die Frequenz der unmittelbar vorher gespielten Note. Folgende 8 Kommandos sind möglich:

Kom.	Stacato	Frequenzbereich für 1 bis 28		
e8	aus	a	...	c2
e9	aus	a0	...	c3
ea	aus	a1	...	c4
eb	aus	a2	...	c5
ec	ein	a	...	c2
ed	ein	a0	...	c3
ee	ein	a1	...	c4
ef	ein	a2	...	c5

Die Kodierung 30 mit den unteren 3 Bit gleich 0 (F0h) wird zur Kennzeichnung des Anfangs einer Melodie benutzt. Das Programm wartet dann auf einen Impuls am Signal IC1 des 68HC11 und startet nach einem Impuls den Song. Bei der Kodierung 31 und alle unteren 3 Bit gleich 1 (FFh) springt das Programm zum 1. Song zurück. Jeder Song muß dann wie folgt aufgebaut sein:

Melodieanfang F0h - Ist das 1. Byte jedes Songs, danach steht ein Faktor für das Tempo des Songs, gefolgt von Noten und Kommandos.

Syntax: <F0h> <Tempobyte> <Notenbytes nach Tabelle,...>

Blockende FFh - muß hinter dem letzten Song stehen

Signal Ausgang an: OC2 -- ST 3.21 am ZWERG11A
 Starteingang an: IC1 -- ST 3.13 am ZWERG11A

Die Länge des Programms mit den Tabellen beträgt 264 Byte. Die restlichen 248 Byte können mit Songs belegt werden. Die 6 Songs des Programms belegen 189 Byte. Es sind also noch 59 Byte für weitere Songs frei. Für die Tonerzeugung wird das Compare-Register 2 (TOC2 = \$1018##\$1019) des Timers genutzt. Der Timer des 68HC11 taktet mit der 2 MHz Systemfrequenz einen 16 Bit Zähler. Dessen Inhalt wird dann mit dem Inhalt des Vergleichsregisters auf Übereinstimmung getestet, und es kann in diesem Fall ein Interrupt ausgelöst werden. In der Interruptbehandlung muß nun immer der alte Wert im Vergleichsregister mit dem Wert des aktuellen Teilers addiert werden und die Summe in das Vergleichsregister zurückgeschrieben werden. Dadurch wird dann immer nach der Anzahl von Takten, die addiert wurden, ein Interrupt ausgelöst. Gleichzeitig wird dabei immer das Bit OC2 vom 68HC11 invertiert. Dadurch entsteht an diesem Pin die gewünschte Tonfrequenz. Für die Tonlänge wird als Zeitbasis das Real Time Interrupt Flag (Bit 6 im Register TFLG2) benutzt. Dieses Bit wird bei 2 MHz Systemtakt aller 4,1 ms gesetzt, wenn im Register PACTL die unteren beiden Bits rückgesetzt bleiben. Die Zeit für 1/16 ergibt sich dann im 1. Song zu $10^{*24*4,1/10}$ ms = 98,4 ms, während sie im letzten Song $10^{*37*4,1/10}$ ms = 151,7 ms beträgt. Das zeigt den großen Spielraum, den man für die Einstellung der Geschwindigkeit eines Songs hat. Der Start eines Songs wird durch eine Flanke am Signal IC1 ausgelöst. Ein Start erfolgt dabei bei einer fallenden und auch bei einer steigenden Flanke. Das Quellprogramm ist in der folgenden Liste zu sehen:

```
*****
* Melodiegeneratorprogramm für den MC68HC11A1 *
* - Entwurf : Heiko Böttcher *
* Meisenweg 1, 01589 Riesa *
* - Merkmale : 3 Oktaven Tonumfang mit Pausen *
* Notenlängen von 1/16 bis 3/4 *
* Tempo jedes Liedes einstellbar *
* - Erweitert: Dr. Siegmars Schöne *
* - neues : Stacato abschaltbar, Frequenz- *
* bereich kann um 4 Oktaven ver- *
* schoben werden. 3 neue Songs *
*****
```

```
#define _____ *****
```

```
* ___ *** Konstanten ****
```

```
REG EQU $1000 * Anfangsadresse des Registerblocks
otoc2 EQU $18 * Offset des toc2-Registers zu REG
otctl1 EQU $20 * Offset des tctl1-Reg.
otctl2 EQU $21 * Offset des tctl2-Reg.
otmsk1 EQU $22 * Offset des tmsk1-Reg.
otflg1 EQU $23 * Offset des tflg1-Reg.
otflg2 EQU $25 * Offset des tflg2-Reg.
```

```
* ___ *** RAM-Belegung
```

```
bss
ORG $0000
adr_nfrq ds.w 1 * Ablageplatz der Notenfrequenz
mode ds.b 1 * Bit 2 Stacato 10 F-Faktor
tempo ds.b 1 * Songtempo (Faktor für Notenlänge)
laenge ds.w 1 * Notenlänge
re1 equ *
```

* __*** Programm Stack

	ds.b	\$db-re1	* Stack von re1 bis stack
stack	ds.b	1	* oberste Stackadresse
adr_iroc2	ds.b	3	* Einsprungadresse fuer OC2 Interrupts
*			* muß auf \$dc stehen !!!

	text		
	ORG	\$B600	
* __***	Programminitialisierung	****	
main	LDS	#stack	* Stackbereich ab RAM-Adresse \$DB
	LDAA	#\$7E	* OC2 Interruptadresse eintragen
	STAA	<adr_iroc2	* OPC vom JMP laden!
	LDX	#frq_ir	
	STX	<adr_iroc2+1	* ADR vom JMP laden!
	LDX	hifrequenz	* Frequenz initialisieren
	STX	<adr_nfrq	
	ldaa	#1	
	staa	<mode	
	LDX	#melanf	* Melodiezeiger initialisieren
	PSHX		
	LDX	#REG	
	BSET	\$30,otctl2,X	* eine Aktion löst IC1 aus
	BSET	\$40,otmsk1,X	* OC2 Interrupts gestatten
	BSET	\$40,otctl1,X	* TOGGLE - Funktion für OC2
	CLI		* Freigabe maskierter Interrupts

```

* ___*** Hauptprogramm ****
start      LDX   #REG          * Anfang des Steuerregisterblocks
inloop    BRCLR $04,otflg1,X,inloop * Warten auf Eingabe
          BSET  $04,otflg1,X
          PULX
          LDAA  1,X          * Notenzeiger laden
          STAA  <tempo      * Tempo der Melodie laden
          INX
replay    INX
          LDY   #length
          LDAA  <tempo
          LDAB  0,X
          cmpb  #$E8        * Anfang der Längentabelle
          bcs  no_kom       * Faktor für rel. Notenlänge
          stab  <mode      * relative Notenlänge ermitteln
          clrb
no_kom    ANDB  #$07        * 1/16 fnr Modeumschaltung
          ABY
          LDAB  0,Y        * nur Nr. der rel. Notenlänge
          MUL
          STD  <laenge     * Pos. rel. Notenlänge in IY
          LDY   #freqs-2  * rel. Notenlänge in B laden
          LDAB  0,X        * = Faktor * rel. Notenlänge
          cmpb  #$E8        * = abs. Notenlänge
          bcc  no_ton      * Anfang der Frequenztafel
          ANDB  #$F8        * Notenfrequenz ermitteln
          beq  hi_ff       * Kommando von E8 ... EF
          LSRB
          LSRB            * Frequenz nicht ändern!
          ABY            * nur Notenummer
          LDY   0,Y        * Pause
          clra
          ldab  <mode
          andb  #3         * Frequenzen sind 16 Bitwerte
          xgdy
          iny
mul_ff    dey
          beq  en_ff
          lsr  d
          bra  mul_ff
hi_ff     ldd  0,Y
en_ff     STD  <adr_nfrq   * Frequenz einstellen
no_ton    JSR  nigt       * Note spielen
          LDAA  #$F0       * Melodie - Blockende erreicht?
          CMPA  1,X
          BHI  replay     * sonst, nächste Note ausgeben
          BEQ  noblockend * bei Blockende, Zeiger auf
          LDX  #melanf-1  * Blockanfang
noblockend INX
          PSHX
          ldd  hifrequenz * Notenzeiger retten
          std  <adr_nfrq   * Pausenfrequenz einstellen
          BRA  start      * Melodieende - Zurück zum Anfang

* ___*** Unterprogramme ****
* ___*** IR - Frequenz **
frq_ir    LDX   #REG          * Anfang des Steuerregisterblocks
          LDD  <adr_nfrq   * neuen Interrupt Zeitpunkt
          ADDD  otoc2,X    * bestimmen
          STD  otoc2,X    * ...
          BSET  $40,otflg1,X * Interruptflag wieder löschen
          RTI            * IR - Ende

```

```

* ___ *** UP - Note spielen **
nlgt          LDY  #REG
              LDD  <laenge
wait          BRCLR $40,otflg2,Y,wait      * Note abzählen
              BSET  $40,otflg2,Y
              SUBD  #10
              BGT  wait
              brclr 4,mode,nostc          * Stacato ist aus
              LDY  hifrequenz            * Ton aus
              STY  <adr_nfrq
nostc         LDY  #$0500                * kurze Pause zwischen 2 Noten
pause        DEY
              BNE  pause
              RTS                          * UP Ende

endpg         equ  *

              DATA
              org  endpg

```

* ___ *** Daten ****

```

hifrequenz   DC.W  $0040                * Pausenfrequenz 31.25 kHz
* ___
              A   Ais H   C0  Cis0 D0  Dis0
freqs        DC.W  9092,8582,8100,7645,7216,6811,6429
* ___
              E0  F0  Fis0 G0  Gis0 A0  Ais0
              DC.W  6068,5728,5406,5103,4816,4546,4291
* ___
              H0  C1  Cis1 D1  Dis1 E1  F1
              DC.W  4050,3823,3608,3406,3215,3034,2864
* ___
              Fis1 G1  Gis1 A1  Ais1 H1  C2
              DC.W  2703,2551,2408,2273,2145,2025,1911

```

```

* ___ relative      1 3 1 3 1 3 1 3
* ___ Notenlängen  -- -- - -- - - - -
* ___ Tabelle       16 32 8 16 4 8 2 4

```

```

length       DC.B  10,15,20,30,40,60,80,120

```

```

* ___      1.Byte =   Anfangskennzeichen
* ___      2.Byte =   Faktor für relative Notenlänge
* ___      3. ... =   Noten- und Notenlängen- nummern
* ___      Bitaufteilung =   nnnn nlll
* ___      nnnnn =   0 ... 28 = Pos. in Frequenzliste
* ___      =   29 in lll steht ein Kommando Zeitdauer = 1/16
* ___      Ton wie vorhergehender. Das höchste Bit von lll
* ___      schaltet Stacato ein/aus, die unteren beiden Bits
* ___      sind Faktor für Frequenzen. 001 ist Default!
* ___      lll =   0 ... 7 = Pos. in Längentabelle

```

* ___ Song 1 C-Dur Sonate KV545

```

melanf       DC.B  $F0,$18
              DC.B  $86,$A4,$BC,$7D,$80,$90,$84,$04,$CE,$BC,$E4,$BC
              DC.B  $AA,$A0,$A8,$A4

```

* ___ Song 2 A-Dur Sonate KV331

```

              DC.B  $F0,$25
              DC.B  $8B,$90,$8A,$A4,$A2,$7B,$88,$7A,$94,$92,$6C,$7A
              DC.B  $8C,$92,$8C,$7A,$6C

```

* __ Song 3 Alla Turka KV331

DC.B \$F0,\$20
DC.B \$6A,\$60,\$68,\$82,\$02,\$82,\$78,\$80,\$A2,\$02,\$A2,\$98
DC.B \$A0,\$CA,\$C0,\$C8,\$CA,\$C0,\$C8,\$E5

* __ Song 4 Glory Glory Halleluja

dc.b \$F0,\$25
dc.b \$5B,\$48,\$40,\$58,\$80,\$30,\$44,\$22,\$80,\$20,\$6B,\$78
DC.B \$20,\$78,\$80,\$68,\$5C,\$44,\$58,\$02,\$48,\$40,\$58,\$20
DC.B \$30,\$42,\$30,\$00,\$20,\$00,\$80,\$20,\$32,\$00,\$30,\$20
DC.B \$00,\$18,\$00,\$25

* __ Song 5

dc.b \$F0,\$25
DC.B \$22,\$32,\$42,\$20,\$00,\$22,\$32,\$42,\$20,\$00,\$42,\$4A
DC.B \$5B,\$00,\$42,\$4A,\$5B,\$00,\$58,\$68,\$58,\$48,\$42,\$20
DC.B \$80,\$58,\$68,\$58,\$48,\$40,\$00,\$22,\$82,\$5A,\$83,\$00
DC.B \$82,\$5A,\$83

* __ Song 6

dc.b \$F0,\$25
DC.B \$20,\$40,\$48,\$5B,\$EA,\$00
DC.B \$20,\$40,\$48,\$5B,\$E9,\$00,\$20,\$40
DC.B \$48,\$5A,\$42,\$22,\$40,\$00,\$40,\$33,\$00,\$30,\$40,\$30
DC.B \$22,\$00,\$80,\$42,\$58,\$00,\$58,\$4B,\$00,\$48,\$40,\$48
DC.B \$58,\$00,\$42,\$22,\$32,\$25

melende DC.B \$FF