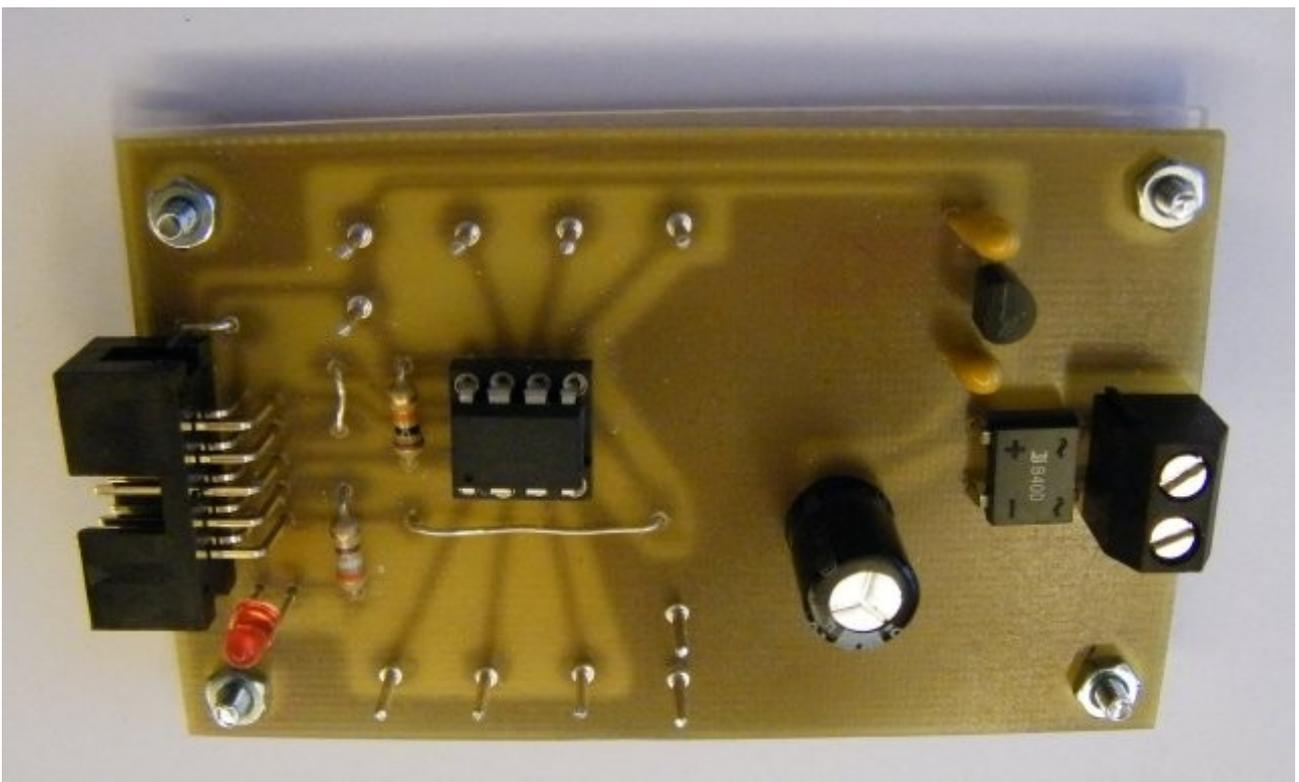


ATtiny13-Experimentier-Board



Gerhard Schmidt, Kastanienallee 20, D-64289 Darmstadt, <http://www.avr-asm-tutorial.net>

Februar 2009

Inhalt

Beschreibung

Hardware

Copyright und Quellen

Anhang

- Technische Daten, Bestückungsplan, Layout, Bauteilliste, Programmierbeispiel

Beschreibung

Das ATtiny13-Experimentierboard ermöglicht

- das Programmieren eines ATMEL ATtiny13 über eine ISP10-Standard-Schnittstelle,
- einfache Experimente mit dem Prozessor, die die Ports, den Timer/Counter, die AD-Wandler-Kanäle und das EEPROM des Prozessors verwenden,
- die Versorgung des Prozessors mit 5,0 V Betriebsspannung über einen Spannungsregler aus einer Batterie oder einem Netztrafo ermöglichen.

Alle Ein- und Ausgänge des Prozessors sowie die Versorgungsspannungen des Boards sind über 1 mm-Steckpins zugänglich.

An die Steckpins können angeschlossen werden:

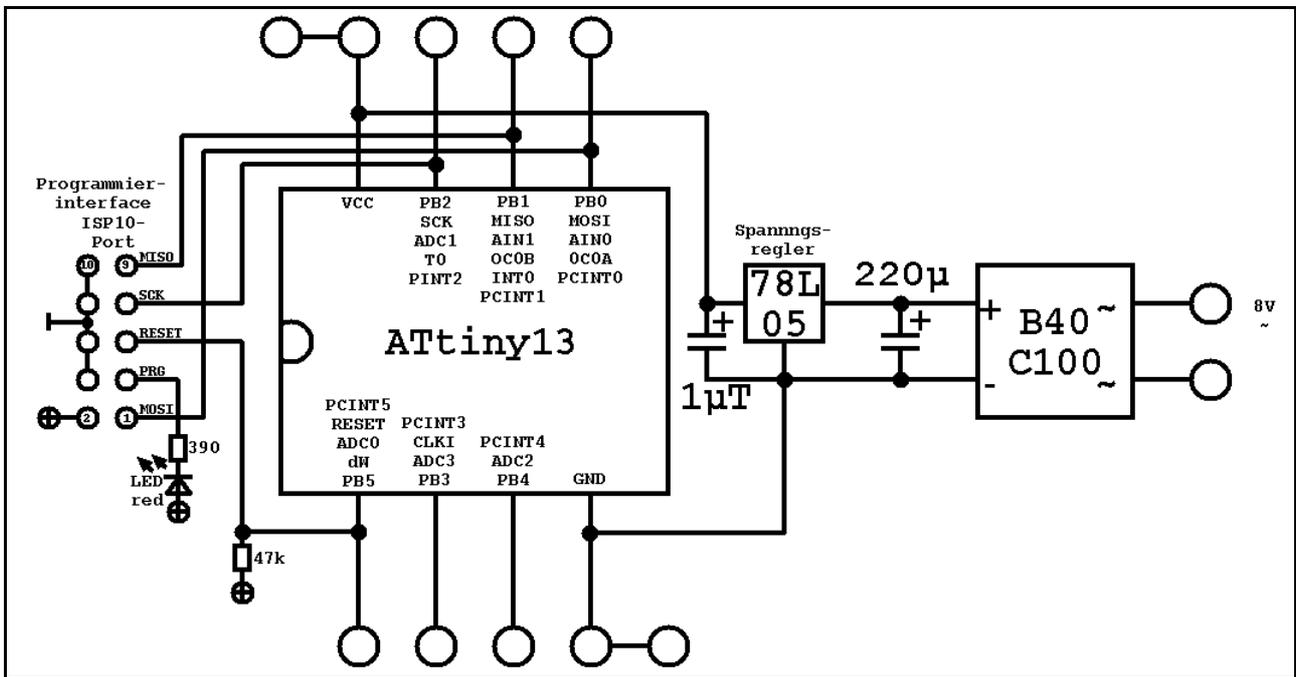
- eine gelbe LED mit Vorwiderstand 390 Ω ,
- ein Taster,
- ein Potentiometer 47 k Ω lin.,
- ein Lautsprecher 45 Ω / 0,2 W über einen Elko 47 μ F.

Hardware

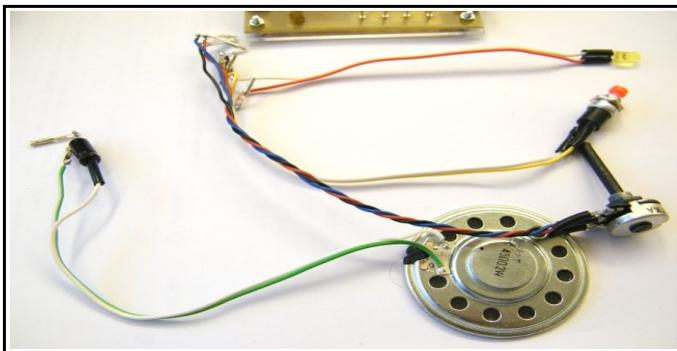
Die Hardware des Experimentierboards ist im Schaltbild dargestellt. Sie besteht aus dem

- Programmier-Interface, einem ISP10-Wannenstecker mit den Programmiersignalen MOSI, RESET, SCK und MISO, einer roten 3 mm-LED mit 390 Ω -Vorwiderstand und den Betriebsspannungsanschlüssen +5 V und GND,
- dem Prozessor ATtiny13 mit den Steckpins und dem Pull-Up-Widerstand (47k bzw. 10k) am Reset-Eingang,
- der Stromversorgung mit der Schraubbuchse, einem Brückengleichrichter, dem Siebelko

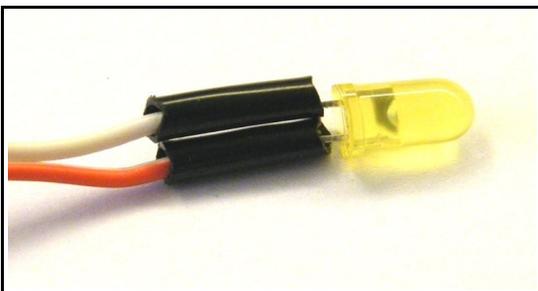
mit 220 μ F, einem Spannungsregler 78L05 und zwei Tantalelkos am Ein- und Ausgang (1 μ F bzw. 0,1 μ F, hier nur ein Elko eingezeichnet).



Am ISP10-Anschluss sind im Unterschied zum Schaltbild die Pins 4 und 6 offen und nicht mit GND verbunden!



Für die Experimente hat das Board mit Steckanschlüssen verlötete Bauteile.



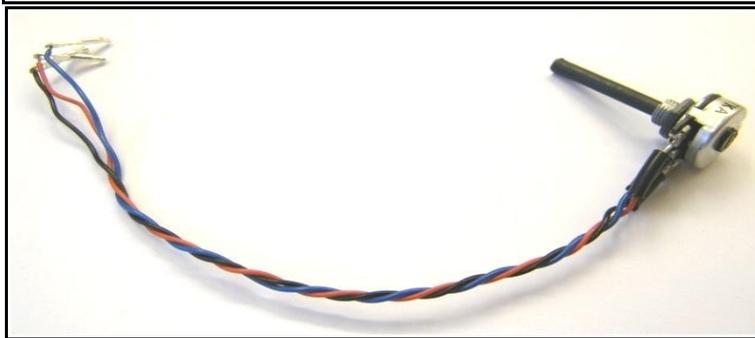
Bei der LED ist die Anode am roten Verbindungskabel angeschlossen, der Pin ist an +5 V anzuschließen. Am Stecker ist ein Vorwiderstand von 390 Ω angelötet, der mit einem Portpin zu verbinden ist.



Der Taster ist über ein gelbes und weißes Kabel angeschlossen.



Der Lautsprecher ist über einen Elko 47 μ F mit den Steckanschlüssen verbunden. Die grüne Anschlussleitung ist an Plus, die weiße Leitung mit dem Elko an einen Portausgang anzuschließen.



Das Poti 47 k Ω lin. ist mit mit der roten Leitung an +5 V, mit der schwarzen an GND anzuschließen und liefert am blauen Anschluss, der mit einem AD-Wandler-Eingang verbunden werden kann, eine lineare Einstellspannung.

Copyright und Quellen

Diese Anleitung wurde von Gerhard Schmidt, Darmstadt, erstellt:

<http://www.avr-asm-tutorial.net>

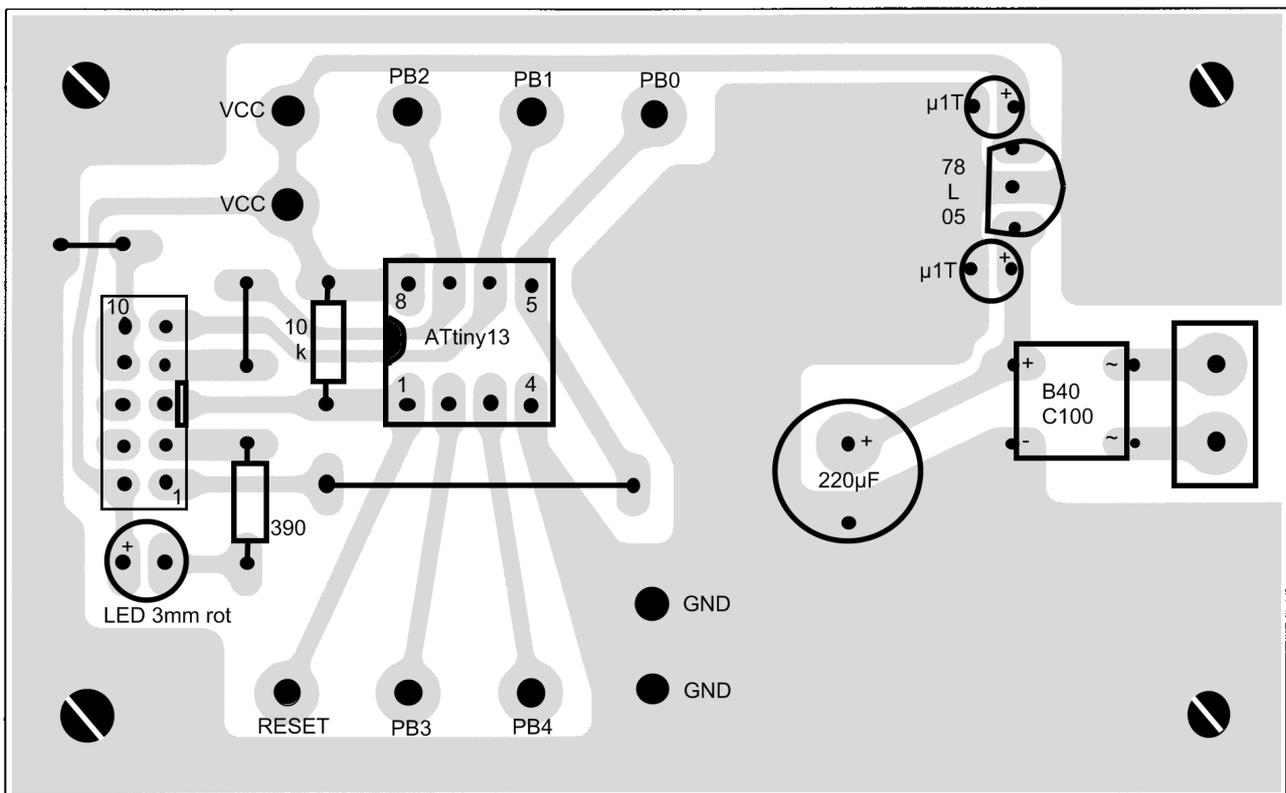
Technische Daten

Versorgungsspannung: 8..15 V =, 6..12 V ≈

Betriebsspannung: 5 V +/- 10% aus Versorgungsspannung, max. 100 mA Spitzenlast,
Versorgung über ISP10-Schnittstelle: Spannung 2,7...5,5 V

Strombedarf: Idle Mode oder Sleep Idle Mode ohne angeschlossene Peripherie: 5 mA; ohne
Prozessor: ca. 2 mA (Eigenbedarf Spannungsregler)

Bestückungsplan ATtiny13-Experimentierboard

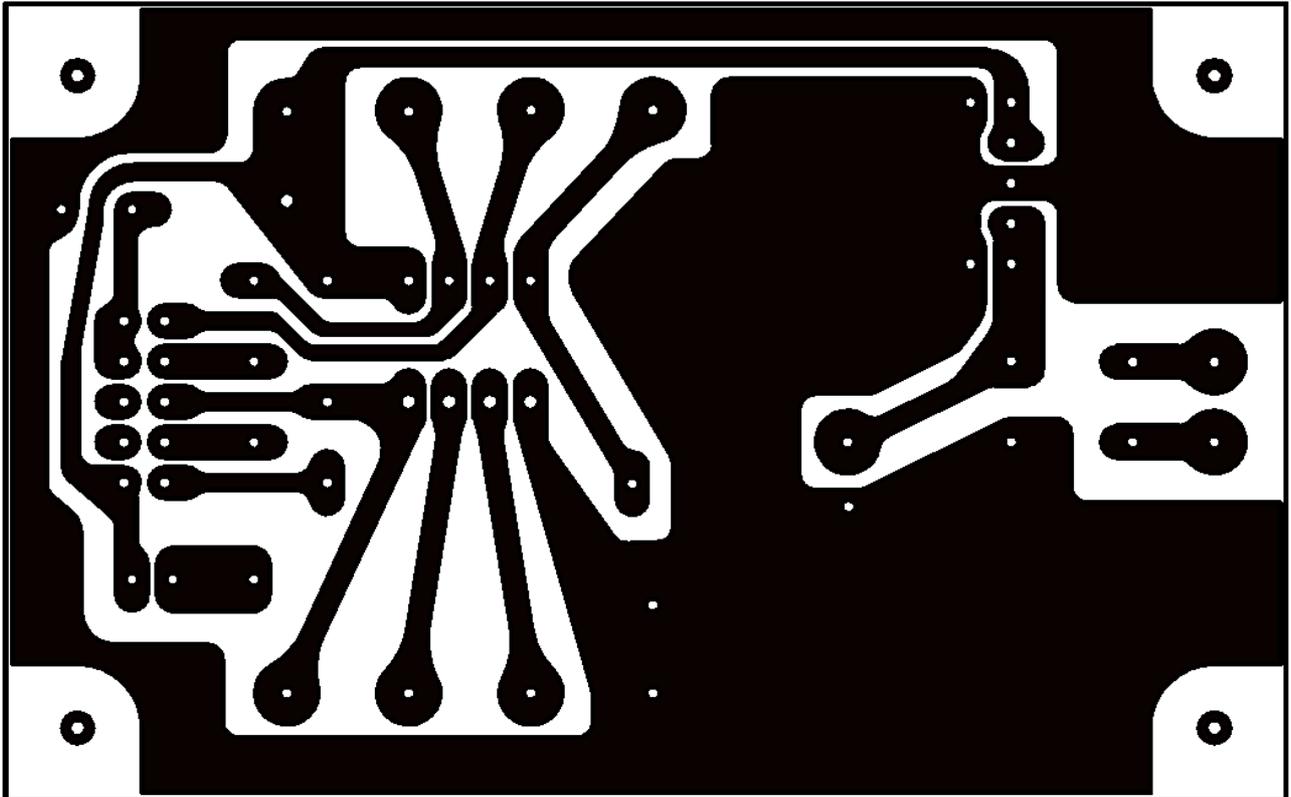


Hinweise:

- Als Pull-Up am Reset-Eingang wurde ein 10k-Widerstand verbaut, kein 47k.
- Die Schrauben wurden isoliert verbaut, sie sind nicht mit GND verbunden.
- Layer stimmt nur sinngemäß mit Layout überein, nicht maßstäblich.

Layout

Originalformat 50*80 mm, vergrößert



Bauteilliste

Die Teileliste enthält alle Einzelteile. Die angegebenen Preise sind Nettopreise eines großen Elektronikversenders bei Abnahme von ca. 15 Stück.

Teil	Anzahl	Einzelpreis	Teilpreis
Wannenstecker 10-polig gewinkelt	1	0,0862	0,0862
Widerstand 390 Ohm	1	0,0276	0,0276
LED 3mm rot	1	0,0431	0,0431
Widerstand 10k	1	0,0276	0,0276
ATMEL ATtiny13-P DIL	1	0,6722	0,6722
IC-Fassung 8-polig gedreht	1	0,0840	0,0840
Tantal-Elko 0,1µF/35V	2	0,0960	0,1920
Spannungsregler 78L05	1	0,0840	0,0840
Elko 220µ/25V	1	0,0603	0,0603
Gleichrichterbrücke B40C800DIP	1	0,0840	0,0840
Anschlussklemme 2-polig 5mm	1	0,1207	0,1207
Löt Nägel 1mm	10	0,0109	0,1092
LED 5mm gelb	1	0,0431	0,0431
Lautsprecher 45 Ohm 0,2W	1	0,7394	0,7394
Drehpoti 47k 4mm	1	0,5798	0,5798
Drucktaster	1	0,0775	0,0775
Lötösen 1mm	9	0,0235	0,2118
Widerstand 390 Ohm	1	0,0276	0,0276
Elko 47µ/16V radial	1	0,0924	0,0924
Pfostenbuchse 10-polig	2	0,0776	0,1552
Flachbandkabel 10-polig 15 cm	1	0,0861	0,0861
Fotoplatine 5*8cm	1	0,3466	0,3466
Entwickler	1	0,2333	0,2333
Ätzmittel	1	0,1633	0,1633
Schrauben M2,5*10	4	0,0197	0,0790
Muttern M2,5	8	0,0122	0,0975
Plexiglasscheibe 5*8 cm	1	0,1600	0,1600
Litze 2*0,14mm, 20cm	4	0,0460	0,1840
Verpackungs- und Versandkosten	1	0,3140	0,3140
Nettosumme			5,18
Bruttosumme			6,17

Programmierbeispiel „Hänschen Klein“

Quellcode:

```

; *****
; * Haenschen klein mit ATTiny13 *
; * Lautsprecher an PB0=Pin5, Taster an PB4=Pin3 *
; * (C)2009 by http://www.avr-asm-tutorial.net *
; *****
;
.nolist
.include "tn13def.inc"
.list
;
; Funktionsweise:
;
; Der Prozessortakt betraegt 1,2 MHz. Der Timer 0
; wird mit einem Vorteiler von 8 gespeist und
; und arbeitet im CTC-Modus. Der Compare-Wert A
; legt dabei die Tonhöhe fest.
;
; Register
;
.def rSreg = R15
.def rmp = R16
.def rimp = R17
; X = XH:XL = Zeiger in die SRAM-Tabelle mit den
; Noten
; Y = YH:YL = Zaehler fuer Anzahl Zyklen des Tons
; Z = ZH:ZL = Zeiger in Notentabelle des Lieds
;
; Hardware
;
.equ pbLsp = 0 ; Lautsprecher-Anschluss
.equ pbLed = 2 ; LED-Anschluss
.equ pbKey = 4 ; Taster-Anschluss
;
; Konstanten
;
; Reset und Interrupt Vektoren
;
.CSEG
.ORG $0000
    rjmp Main
    reti ; INTO
    rjmp KeyInt ; PCINT
    reti ; TCO_OVFLW
    reti ; EERDY
    reti ; ANA_COMP
    rjmp Tc0CmpA
    reti ; TCO_COMPB
    reti ; WDT
    reti ; ADC
;
; Pin Change Int
;
KeyInt:
    in rSreg,Sreg
    ldi ZH,HIGH(2*mtab)
    ldi ZL,LOW(2*mtab)
    clr YH
    ldi YL,1
    ldi rimp,10
    out OCR0A,rimp
    ldi rimp,(1<<CS01)
    out TCCR0B,rimp
    ldi rimp,(1<<COM0A0)|(1<<WGM01)
    out TCCR0A,rimp
    ldi rimp,1<<OCIE0A
    out TIMSK0,rimp

```

```

    out Sreg,rSreg
    reti
;
; Int Service TCO Compare A
;
Tc0CmpA:
    in rSreg, SREG ; save SREG
    sbiw YL,1
    brne Tc0CmpA5
    lpm
    adiw ZL,1
    mov XL,R0
    cpi XL,0xFF
    brne Tc0CmpA1
    ldi rimp,0
    out TCCR0A,rimp
    out TCCR0B,rimp
    out TIMSK0,rimp
    cbi PORTB,pbLsp
    sbi PORTB,pbLed
    rjmp Tc0CmpA5
Tc0CmpA1:
    ldi XH,HIGH(SRAM_START)
    ldi XL,LOW(SRAM_START)
    add XL,R0
    add XL,R0
    ld R0,X+
    ld YL,X
    tst YL
    brne Tc0CmpA2
    ldi rimp,(1<<WGM01)|(1<<COM0A1) ;no toggle
    out TCCR0A,rimp
    ldi YL,nc/4
    sbi PORTB,pbLed
    ldi rimp,dc
    out OCR0A,rimp
    rjmp Tc0CmpA3
Tc0CmpA2:
    out OCR0A,R0
    cbi PORTB,pbLed
    ldi rimp,(1<<WGM01)|(1<<COM0A0) ; Toggle
    out TCCR0A,rimp
Tc0CmpA3:
    clr YH
    lsl YL
    rol YH
Tc0CmpA5:
    out SREG,rSreg ; restore SREG
    reti
;
; Main Init
;
Main:
    ldi rmp,LOW(RAMEND) ; setze Stack
    out SPL,rmp
    sbi DDRB,pbLsp
    sbi DDRB,pbLed
    sbi PORTB,pbKey
    ldi rmp,18 ; kopiere Notentabelle in SRAM
    ldi XH,HIGH(SRAM_START)
    ldi XL,LOW(SRAM_START)
    ldi ZH,HIGH(2*ftab)
    ldi ZL,LOW(2*ftab)
Main1:
    lpm
    st X+,R0
    adiw ZL,1
    dec rmp
    brne Main1
    ldi rmp,1<<pbKey ; Pin Change Int
    out PCMSK,rmp
    ldi rmp,1<<PCIE
    out GIMSK,rmp
    ldi ZH,HIGH(2*mtab)
    ldi ZL,LOW(2*mtab)
    clr YH
    ldi YL,1

```

```

        ldi rmp,10
        out OCR0A,rmp
        ldi rmp,(1<<CS01)
        out TCCR0B,rmp
        ldi rmp,(1<<COM0A0)|(1<<WGM01)
        out TCCR0A,rmp
        ldi rmp,1<<OCIE0A
        out TIMSK0,rmp
        sei
Loop:
        nop ; dummy
        rjmp Loop
;
; Notentabelle
;
;
.equ fc = 528 ; c''
.equ fd = 594 ; d''
.equ fe = 660 ; e''
.equ ff = 704 ; f''
.equ fg = 794 ; g''
.equ fa = 880 ; a''
.equ fh = 990 ; h''
.equ fc2 = fc*2 ; c'''
;
; Delays
;
.equ dc = 1200000/16/fc ; c''
.equ dd = 1200000/16/fd ; d''
.equ de = 1200000/16/fe ; e''
.equ df = 1200000/16/ff ; f''
.equ dg = 1200000/16/fg ; g''
.equ da = 1200000/16/fa ; a''
.equ dh = 1200000/16/fh ; h''
.equ dc2 = 1200000/16/fc2 ; c'''
;
; Dauer
;
.equ nc = fc/5 ; c''
.equ nd = fd/5 ; d''
.equ ne = fe/5 ; e''
.equ nf = ff/5 ; f''
.equ ng = fg/5 ; g''
.equ na = fa/5 ; a''
.equ nh = fh/5 ; h''
.equ nc2 = fc2/5 ; c'''
;
ftab:
.db 0,0 ; pause
.db dc,nc ; c''
.db dd,nd ; d''
.db de,ne ; e''
.db df,nf ; f''
.db dg,ng ; g''
.db da,na ; a''
.db dh,nh ; h''
.db dc2,nc2 ; c'''
;
mtab:
.db 0,0
.db 6,0,4,0,4,0,0,5,0,3,0,3,0,0,2,0,3,0,4,0,5,0
.db 6,0,6,0,6,0,0,6,0,4,0,4,0,0,5,0,3,0,3,0
.db 2,0,4,0,6,0,6,0,0,2,0,0,0,0,3,0,3,0,3,0,3,0
.db 3,0,4,0,5,0,0,4,0,4,0,4,0,4,0,4,0,4,0,5,0,6,0
.db 6,0,4,0,4,0,0,5,0,3,0,3,0,0,2,0,4,0,6,0,6,0
.db 0,1,0,0xFF

```

Hexcode

Zur Programmierung den Code in die Zwischenablage kopieren, mit einem Texteditor in eine Textdatei mit der Endung .hex einfügen und mit der Programmiersoftware in den Flashspeicher des ATtiny13 brennen.

```

:02000000200000FC
:100000003FC0189507C018951895189512C01895F7
:1000100018951895FFB6F0E0E6EDDD27C1E01AE08F
:1000200016BF12E013BF12E41FBD14E019BFFFBEDC
:100030001895FFB6219711F5C8953196A02DAF3FC1
:1000400039F410E01FBD13BF19BFC098C29A16C083
:10005000B0E0A0E6A00DA00D0D90CC91CC2339F41A
:1000600012E81FBDCAE1C29A1EE816BF04C006BE50
:10007000C29812E41FBDD27CC0FDD1FFFBE18950F
:10008000FE90DBFB89ABA9AC49A02E1B0E0A0E6AF
:10009000F0E0E4ECC8950D9231960A95D9F700E1AD
:1000A00005BB00E20BBFF0E0E6EDDD27C1E00AE0B2
:1000B00006BF02E003BF02E40FBD04E009BF78946D
:1000C0000000FECF00008E697E7671846A8C5E9E91
:1000D00055B04BC647D300000600040004000005DD
:1000E00000030003000002000300040005000600F6
:1000F00006000600000600040004000005000300DE
:100100003000200040006000600000200000000D8
:10011000030003000300030003000300040005000004C3
:1001200000040004000400000400050006000600AE
:1001300004000400000500030003000002000400A6
:0801400006000600000100FFAB
:00000001FF

```