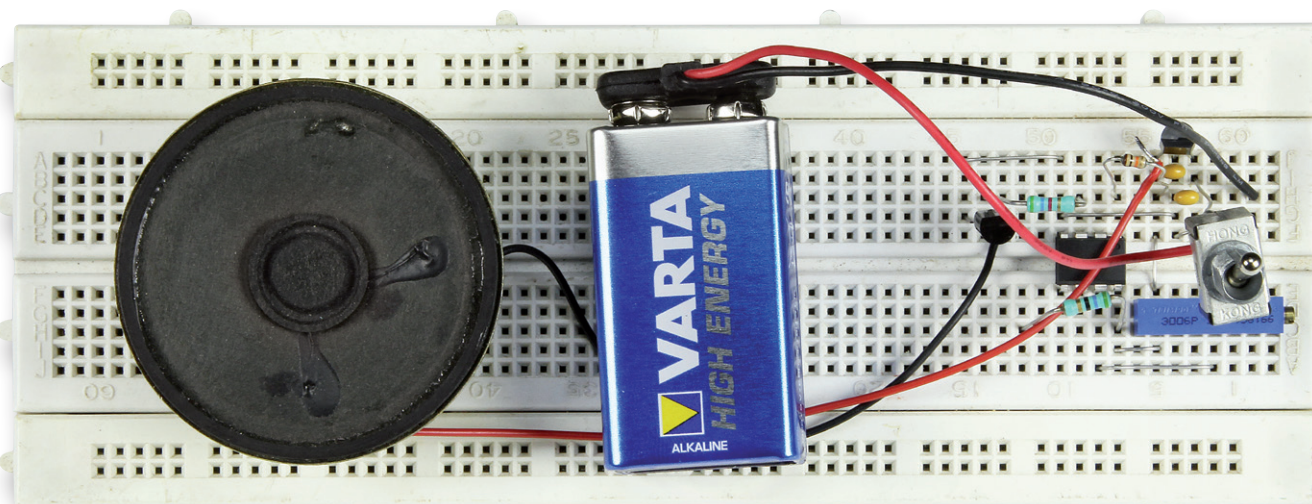


# Bombarie 2.0

## Nieuw lawaai met minder moeite

In de Elektoor-uitgave van april 1979 publiceerden we een schakeling om lawaai te maken. De auteur was terecht trots dat hij daarvoor alleen een CD4040-teller, een zesvoudige CD4049-inverter en wat klein spul had gebruikt. Wie het destijds leuk vond, kan nu opnieuw blij zijn, want dank zij de technische vooruitgang kunnen we tegenwoordig met nog minder componenten herrie maken!

**Friedrich Lischeck**  
(Duitsland)



Het valt meteen op in het schema in **figuur 1**: In plaats van een of ander teller en inverter wordt nu een moderne, kleine microcontroller van het type ATtiny45 gebruikt voor het opwekken van het signaal. Natuurlijk gebruiken we nu een andere techniek: In plaats van een ingewikkelde terugkoppeling gebruiken we software. Eén transistor geeft voldoende versterking om het signaal hoorbaar te maken. Natuurlijk hoort er ook nog een kleine dynamische luidspreker van 8  $\Omega$  bij. De basisfrequentie van de schakeling kan met een instelpotmeter geregeld worden.

### Software

De intelligentie van moderne elektronica zit meestal niet meer in de slimme aansluiting van onderdelen, maar eerder in de software

voor de microcontroller. We komen er tegenwoordig, zelfs voor eenvoudige schakelingen, vrijwel niet meer omheen om een  $\mu\text{C}$  te programmeren. De software voor de AVR-controller (zie **tekst kader**) is geschreven met het bekende programma Bascom-AVR [1]. Bombarie versie 1.0 van 34 jaar geleden wekte een steeds hoger wordend geluid op, dat bij de maximale frequentie weer bij de laagste toon begon. De snelheid waarmee de toonhoogte toenam, hing samen met de ingestelde frequentie.

Nu wekken we eenzelfde signaal op met een ATtiny45. Met de potmeter kunnen we een spanning tussen 0 V en 5 V instellen op analoge ingang ADC.1. De ADC van de controller zet die spanning om in een numerieke waarde tussen 0 en 1023. Dit getal wordt gebruikt om de frequentie 'Fg' in te stellen. Dat gaat

met behulp van Timer0. Telkens als de teller van Timer0 overloopt, wordt een interrupt gegenereerd. In de bijbehorende interrupt-routine 'Oszillator' wordt de timer geladen met een nieuwe beginwaarde en wordt digitale uitgang B.4 omgeschakeld. Zo ontstaat een symmetrische blokgolf. Verder incrementeert de interrupt-routine de variabele 'Fd' tot die een waarde van 1000 bereikt. Daarna wordt hij teruggezet naar 100. In het hoofdprogramma worden de berekeningen uitgevoerd: 'Fd' en de van de potentiometer afhankelijke waarde 'Fg' worden bij elkaar opgeteld. Zo wordt een frequentie tussen 200 Hz en 4 kHz berekend. Dat wordt de beginfrequentie. Vanaf die waarde neemt de frequentie elke keer toe tot hij 1000 Hz hoger is, om dan weer terug te gaan naar de beginfrequentie.

**Beginwaarde**

Hoe wordt nu de beginwaarde (Preload) van de timer berekend? Zoals bekend is de tijd 'T' van een 8-bits timer als volgt te berekenen:

$$T = T_{max} - Preload * T_{clk}$$

De waarde van 'Preload' mag tussen 0 en 255 liggen. 'Tclk' is de van de controller-klok afgeleide kloktijd, die soms nog door een prescaler (1/8/64/256) is verlaagd. 'Tmax' is de maximale tijd waarop we de timer kunnen instellen:

$$T_{max} = Prescale * 256 / f$$

Hierin is 'f' de klokfrequentie. Door 'Preload' naar de linkerkant van de vergelijking te brengen, vinden we:

$$Preload = (T_{max} - T) / T_{clk}$$

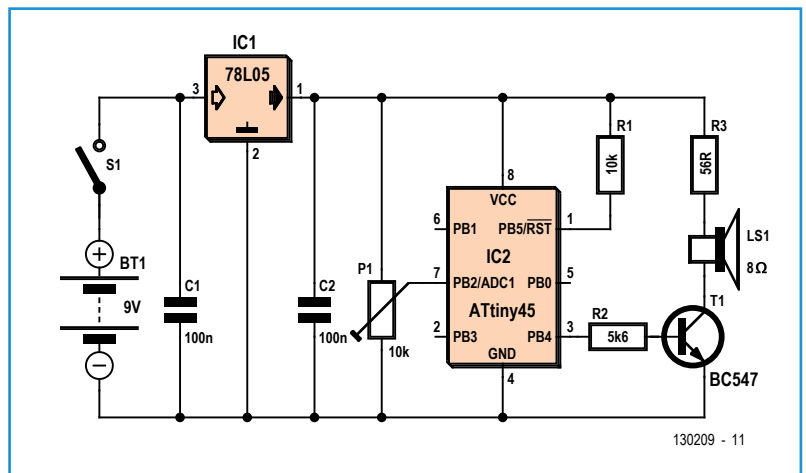
of

$$Preload = (T_{max} - 1 / 2 * F) / T_{clk}$$

Daarin is F de op te wekken frequentie en T de daarmee overeenkomstige tijd, d.w.z. de halve periodetijd.

**Tenslotte**

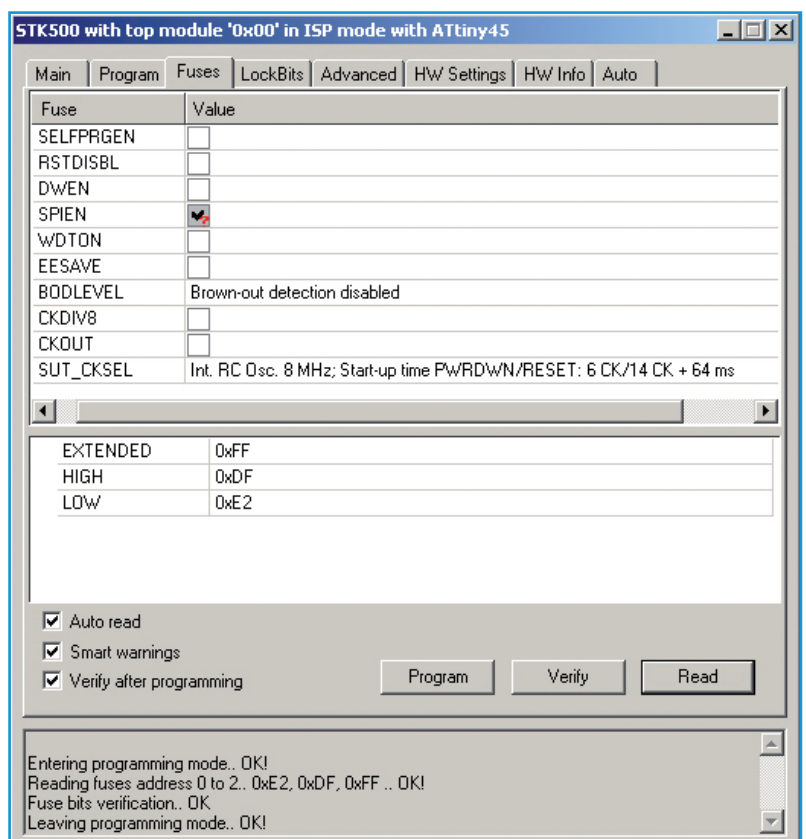
Het programma is zó klein dat een ATtiny45 al wat te groot is. Maar die hadden we toevallig bij de hand. Eigenlijk is het kleinste model, de

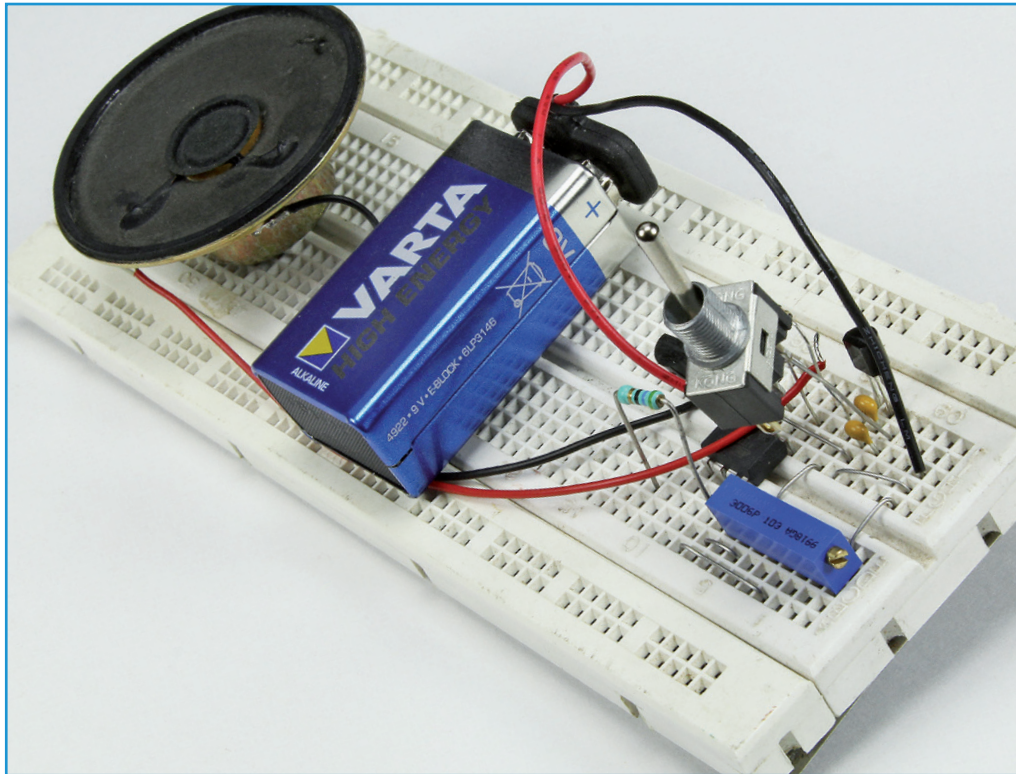


ATtiny25 met een flash-geheugen van 2 KB, al voldoende. Voor het compileren kunnen we de gratis demoversie van Bascom-AVR gebruiken, want die kan maximaal 4 KB code genereren. Als we de gegenereerde code met een zelf gebouwde of gekochte programmer in de controller branden, moeten we op twee dingen letten: Kies het juiste controller-type en vergeet de fuses niet: In **figuur 2** zien we een screenshot van de fuse-instellingen in het

Figuur 1. De schakeling van de Bombarie 2.0 is nog eenvoudiger dat die van het origineel in 1979.

Figuur 2. Zo worden de fuses van de ATtiny45-microcontroller ingesteld.





## Onderdelen-lijst

### Weerstanden:

(Alle 0,25 W)  
 R1 = 10 k  
 R2 = 5k6  
 R3 = 56  $\Omega$   
 P1 = 10 k  
 instelpotmeter

### Condensatoren:

C1, C2 = 100 n/16 V  
 keramisch

### Halfgeleiders:

T1 = BC547  
 IC1 = 78L05  
 IC2 = ATtiny45,  
 geprogrammeerd

### Verder:

DIL-8-voetje voor IC2  
 Batterijclip voor  
 9-V-batterij  
 B = 9-V-batterij  
 S1 = enkelpolige  
 schakelaar  
 LS = luidspreker,  
 8  $\Omega$ /0,5 W

programma Atmel Studio. Hier is te zien dat de klokdeeler 'CKDIV8' is uitgeschakeld en dat de interne 8-MHz-oscillator met lange starttijd is gekozen. Als er iets fout gaat, wordt meestal een te laag of helemaal geen geluid geproduceerd. De broncode en een kant-en-klaar hex-bestand kunnen gratis gedownload worden van [2].

Over de opbouw valt niet veel te vertellen. Die paar onderdelen zijn gemakkelijk op een stukje gaatjesprint te zetten. Het is aan te raden voor IC2 een IC-voetje te gebruiken.

Tenslotte nog een citaat uit het artikel uit 1979: Zo'n apparaat is een uitstekend hulpmiddel tegen... ja tegen wát eigenlijk?"

(130209)

## Weblinks

[1] Bascom-AVR: [www.mcselec.com](http://www.mcselec.com)

[2] [www.elektor-magazine.nl/130209](http://www.elektor-magazine.nl/130209)

## De code

```
'Buhei Tiny45
$regfile = "attiny45.dat"
$crystal = 8000000
```

```
'Pin B.0 wird Ausgang:
Ddrb = &B00010000
'Alles aus:
Portb.4 = 0
```

```
Dim Preload As Byte
Dim F As Word
```

```

Dim Fg As Word
Dim Fd As Word
Dim Tmax As Single
Dim Tclk As Single
Dim H As Single

On Timer0 Oszillator
Config Timer0 = Timer , Prescale = 256
Enable Timer0
Enable Interrupts
Preload = &H64:                               '100 Hz
Timer0 = Preload

Config Adc = Single , Prescaler = Auto , Reference = Avcc
Start Adc
'-----
'Init
Tclk = 256 / 8000000
Tmax = 256 * Tclk
Fg = 100
Fd = 100
'-----
'Hauptprogramm
'F=200-4000
Do
  Fg = Getadc(1)
  H = Fg + 100
  Fg = Fg + H
  Fg = Fg + H
  F = Fg + Fd
  H = 2 * F
  H = 1 / H
  H = Tmax - H
  H = H / Tclk
  Preload = Int(h)
  H = Preload * Tclk
  H = Tmax - H
  H = 0.5 / H
  F = Int(h)
Loop
End
'-----
Oszillator:
  Timer0 = Preload
  Portb.4 = Not Portb.4
  Incr Fd
  If Fd > 1000 Then Fd = 100
Return
'-----

```