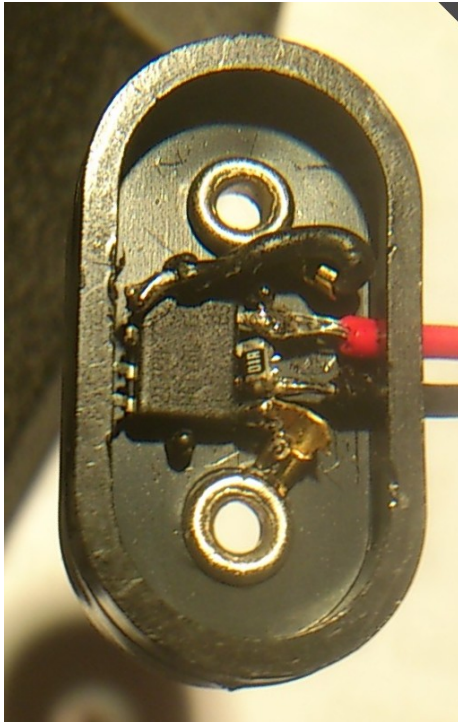


## ATt13 – Trick



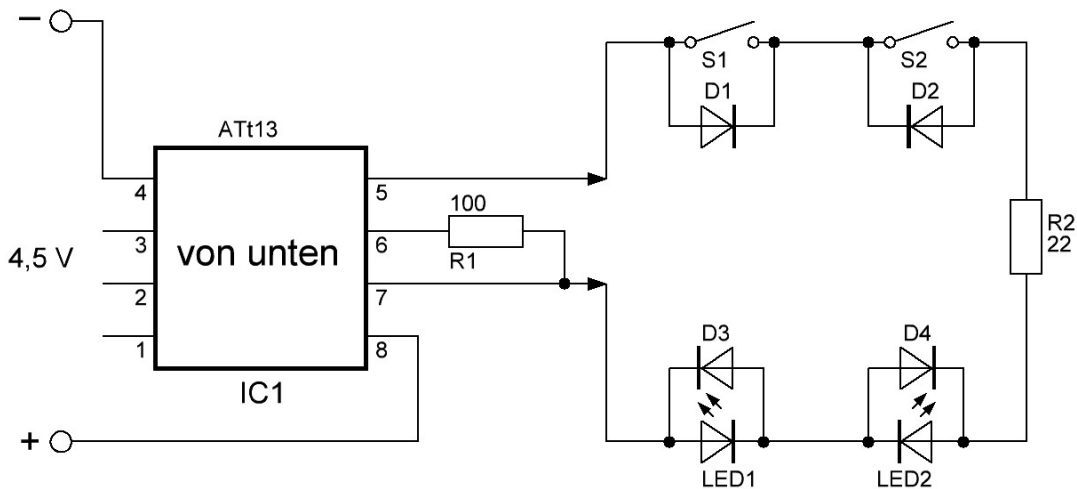
Die schwarze Minus-Litze (im Bild oberer Anschluss) etwas länger lassen, damit man im Bogen von oben an Pin4 (GND-Anschluss) des ATt13 kommt.

Den 100 $\Omega$ -Widerstand vor dem Einbau von der Oberseite auf die Pins 6 und 7 (PB1, PB2) löten, dann hat man nach dem Einbau mehr Platz, den schwarzen Draht zur Schaltung (untere Litze rechts) an Pin7 zu löten.

Auch für den roten Draht zur Schaltung (obere Litze rechts) ist dann mehr Raum.

Durch die Lötstelle Pin8 (Vcc) mit dem Kabelschuh bekommt der ATt13 halt.

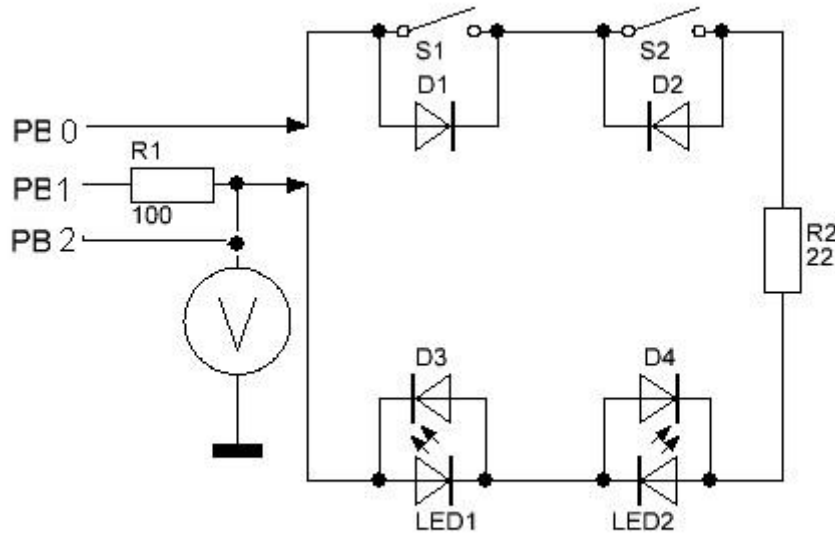
Für den unbedarften Zuschauer sollte die schwarze Litze oben und die rote Litze unten aus dem Klipp kommen, weil das dann auch den Batteriepolen entspricht!  
Also schwarz an Pin5 und rot an Pin7.



## ATt13 – Trick

Die Messschaltung ist an die Ausgänge PB1 und PB2 des Attiny2313 angeschlossen. Mit den angegebenen Widerständen fließt jeweils ein Strom von ca. 13mA (gemessen als Spannungsabfall über R1).

Mit einem hochohmigen Voltmeter – wie eingezeichnet – ergeben sich laut Tabelle die Spannungswerte, die vom ATt13 an Pin3 ausgewertet werden können



Vers. Nr.	PB0	PB1	S1	LED1	S2	LED2	U in V
1	0	1	Ein	<b>An</b>	Ein	Aus	<b>3,4</b>
2	0	1	Ein	<b>An</b>	Aus	Aus	<b>3,8</b>
3	0	1	Aus	Aus	Ein	Aus	5,0
4	0	1	Aus	Aus	Aus	Aus	5,0
5	1	0	Ein	Aus	Ein	<b>An</b>	<b>1,6</b>
6	1	0	Ein	Aus	Aus	Aus	0
7	1	0	Aus	Aus	Ein	<b>An</b>	<b>1,2</b>
8	1	0	Aus	Aus	Aus	Aus	0

Interessant sind die Zeilen 1 und 2:  $U = 3,4V$  bzw.  $U = 3,8V$

bzw. die Zeilen 5 und 7:

$U = 1,6V$  bzw.  $U = 1,2V$ .

Die ca. 3,5V oder ca. 1,5V differieren genügend stark voneinander, die Differenzen von ca. 0,4V müssten vom Programm abgefangen werden.

Programmteil für „Reihenschaltung“: Zunächst ist PB0 = 1 und PB1 = 0 eingestellt; wenn S1 beliebig ist und S2 aus ist, dann sind beide LEDs aus und die gemessene Spannung ist kleiner als 1,2 V. Wenn nun S2 eingeschaltet wird, dann ist die Spannung größer oder gleich 1,2 V und die erste while-Schleife wird abgebrochen; jetzt wird PB0 = 0 und PB1 = 1 eingestellt (umgekehrte Polung).

Die folgende while-Schleife wird solange durchlaufen, wie die gemessene Spannung größer als 3,8 V ist. Dies ist der Fall, wenn S1 aus ist und S2 beliebig ist; die LEDs sind dann beide aus. Dies wiederholt sich immer wieder; die beiden LEDs sind also aus, solange mindestens einer der beiden Schalter aus ist.

## ATt13 – Trick

Wenn S1 und S2 beide eingeschaltet sind, dann werden die beiden while-Schleifen jeweils nicht durchlaufen; es wird lediglich zwischen den Zuständen  $PB0 = 1$  und  $PB1 = 0$  (LED2 an) und  $PB0 = 0$  und  $PB1 = 1$  (LED1 an) hin und her geschaltet. Dies kann das Auge nicht auflösen: Es sieht beide LEDs gleichzeitig (nicht ganz so hell wie bei permanenten Betrieb) leuchten. Das Programmteil simuliert also eine Reihenschaltung.

Wenn die while-Schleifen nicht durchlaufen werden, dann wird der Zähler z auch nicht auf 0 gesetzt. Das bedeutet: Wenn S1 und S2 beide eingeschaltet ist, dann wird der Zähler nach einiger Zeit (einige Sekunden) die Grenze von 200 überschreiten; in diesem Fall bricht die do-loop-Schleife ab und damit beginnt nun das Programmteil „Parallelschaltung“. Dieses sorgt lediglich dafür, dass die Versorgungsspannung für die Schaltung über die Ausgänge PB0 und PB1 fortwährend ihre Polung ändert. Die Dioden in der Schaltung sorgen nun dafür, dass LED1 und LED2 getrennt über die Schalter S1 und S2 in den entsprechenden Halbphasen ein- und ausgeschaltet werden. Auch hier kann das Auge das Blinken nicht auflösen.

## ATt13 – Trick

```
' Datei für attiny13 von E. Eube, G. Heinrichs und U. Ihlefeldt
' wunderbare Reihenschaltung V2
' Fusebit clkdiv8 nicht gesetzt!
```

```
'-----
-----
```

```
$regfile = "attiny13.dat"
'Attiny13
$crystal = 9600000          '9,6 MHz
```

```
$hwstack = 20
$swstack = 20
$framesize = 5
```

```
'*****
'***** Deklarationen *****
```

```
Dim Messwert As Byte
Dim Z As Byte
Declare Function Messen() As Byte
```

```
'***** Initialisierung *****
```

```
Ddrb = &B11
Admux = &B00100001      ' REFS0 = 0, ADLAR = 1, MUX = 1 = PB2
Adcsra = &B10000001    ' ENABLE, CLOCKPRESCALER = 1 = /1
```

```
Tccr0b = &B00000101
Timer0 - prescale = 1024
Tmsk.toie0 = 1
Z = 0
Enable Interrupts
On Timer0 Overflow
```

```
'*****
'***** Hauptprogramm *****
```

```
' Schalterfunktion wie bei Reihenschaltung
Do
```

```
    Portb = &B01
    Messwert = Messen()
    While Messwert < 30
        Messwert = Messen()
        Z = 0
    Wend
```

```
    Portb = &B10
    Messwert = Messen()
    While Messwert > 220
        Messwert = Messen()
```

## ATt13 – Trick

```
    Z = 0
Wend

Loop Until Z > 200      'ca. 5 Sekunden beide Schalter geschlossen

' Schalterfunktion wie bei Parallelschaltung
Do
  Portb = &B01
  Waitms 1
  Portb = &B10
  Waitms 1
Loop

'*****
'***** Unterprogramme *****

Function Messen() As Byte
  Adcsra.adsc = 1
  Do
  Loop Until Adcsra.adsc = 0
  Messen = Adch
End Function

'*****
'*****Interruptroutinen*****

Overflow:
  If Z <= 200 Then Z = Z + 1
Return
```