



Ralf Pongratz

Nachtaktiver Blinker

Nachtaktiver Blinker

Ralf Pongratz

11. November 2012

Inhaltsverzeichnis

I. Die Schaltung	2
1. Funktionsbeschreibung	2
2. Schaltplan	2
3. Programmierung	3
II. Bekannte Probleme	5

Teil I. Die Schaltung

1. Funktionsbeschreibung

Diese Schaltung ist ein nachtaktiver Blinker. Sie ist einfach mit nur wenigen Bauteilen zu bauen und ohne Parametrierung einsatzbereit.

Die Helligkeitsmessung erfolgt über einen Fotowiderstand (LDR). Fällt der Wert unter eine festgelegte Schwelle, wird alle acht Sekunden ein Lichtblitz über die LED ausgegeben.

Aufgrund des äußerst geringen Stromverbrauchs ist die Schaltung mit einem Satz Batterien über Jahre hinweg einsatzbereit.

2. Schaltplan

Abbildung 1 zeigt den Schaltplan des Blinkers.

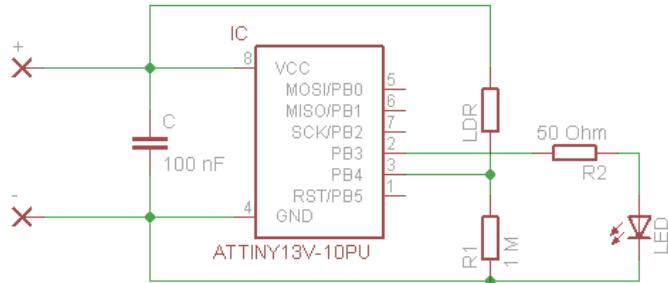


Abbildung 1: Schaltplan.

In der Mitte befindet sich der Microcontroller, links davon die Spannungsversorgung. Die Schaltung benötigt eine Spannung von 3 V. Zwei Standardbatterien in Reihe geschaltet ergeben diese Spannung. Der Pluspol muss an den Pin +, der Minuspol an den Pin - angeschlossen werden. Rechts vom IC befindet sich zum einen die Helligkeitsmessung, bestehend aus R1 und dem Fotowiderstand LDR, zum anderen die LED zur Ausgabe der Blinksignale mit dem Vorwiderstand R2.

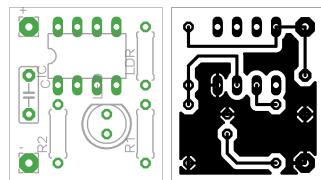


Abbildung 2: Platinenlayout.

Ein Entwurf für eine Platine ist in Abbildung 2 zu sehen. Die linke Abbildung zeigt die Bestückungsseite, die rechte die Leiterbahnen von der Unterseite aus gesehen. Bei der Bestückung ist auf die Polung der Leuchtdiode (die abgeflachte Seite bzw. das kurze Beinchen muss zum IC zeigen) und des ICs (die Einkerbung muss zum Kondensator hin zeigen) zu achten.

3. Programmierung

```

1 $regfile = "ATTiny13.DAT"
2 $crystal = 16000
3 $hwstack = 2
4
5 Config Adc = Single , Prescaler = Auto
6 Config Portb = &B00001000

```

```

7 Portb = 0
8 Stop Ac
9
10 Wdtcr = &B11010011
11 Enable Interrupts
12
13 Const Tagschwelle = 800
14 Dim Ldr As Integer
15
16 Do
17   Gosub Pause
18   Start Adc
19   Ldr = Getadc(2)
20   Stop Adc
21   If Ldr <= Tagschwelle Then
22     Gosub Blinken
23   End If
24 Loop
25
26 Blinken:
27   Portb.3 = 1
28   Reset Watchdog
29   Powerdown
30   Portb.3 = 0
31 Return
32
33 Pause:
34   Wdtcr = &B11110001
35   Reset Watchdog
36   Powerdown
37   Wdtcr = &B11010011
38 Return
39
40 End

```

In den Zeilen 1 bis 3 werden allgemeine Einstellungen an der Hardware vorgenommen. Zuerst wird dem Compiler mitgeteilt, um welchen Prozessortyp es sich handelt. Danach wird die Frequenz des internen Oszillators gesetzt. Zum Schluss wird der Stack auf 2 gesetzt, damit für die Variablen des Programms genügend Platz zur Verfügung steht. Dies hat allerdings zur Folge, dass die Verschachtelungstiefe von Funktionsaufrufen maximal zwei betragen darf, was für dieses Programm jedoch ausreichend ist.

Danach wird der Microcontroller über die Register konfiguriert. Zuerst wird der Analog-Digital-Wandler eingestellt, mit dem der Helligkeitsgrad eingestellt wird. Danach wird der Port 3 als Ausgang konfiguriert und auf low gesetzt. Die restlichen Port sind Ein-

gänge. In Zeile 8 wird der Analog-Komparator abgestellt, um Strom zu sparen. Er wird in diesem Programm nicht benötigt. Zeile 9 und 10 stellen den Watchdog-Timer, über den die Wartezeiten realisiert werden, auf 0,125 s und in den Interrupt-Modus.

Nun werden die Konstanten und Variablen definiert, die im Programm benötigt werden. **Tagschwelle** gibt den Wert vor, unterhalb dessen die Schaltung blinkt. Der aktuelle Helligkeitswert wird in der Variablen **Ldr** gespeichert.

Die Zeilen 16 bis 24 enthalten das Hauptprogramm. Zu Beginn wird der Controller für 8 s in den Schlafmodus versetzt. Danach wird in Zeile 18 bis 20 der Analog-Digital-Wandler gestartet, der Helligkeitswert eingelesen und der Wandler wieder ausgeschaltet. Ist der Helligkeitswert kleiner als der Schwellwert **Tagschwelle**, wird die Unterroutine **Blinken** aufgerufen.

Die Unterprozedur **Blinken** befindet sich in den Zeilen 26 bis 31. Hier wird der Port 3, an dem die LED angeschlossen ist, angeschaltet, mittels des Watchdog-Timers 0,125 s gewartet und der Ausgang wieder ausgeschaltet.

Die Zeilen 33 bis 38 enthalten die Prozedur **Pause**. Damit wird der Microcontroller für 8 s in den StandBy-Modus geschaltet. Zu Beginn wird die Zeit des Watchdog-Timers auf den Maximalwert gesetzt. Anschließend wird der Controller in den Schlafmodus versetzt. Wenn er daraus wieder erwacht ist, wird die Zeit des Watchdog-Timers wieder zurückgesetzt.

Teil II.

Bekannte Probleme

Es sind keine Probleme bekannt.