



Ralf Pongratz

Grundversion

Reaktivlicht mit Watchdog-Abschaltung
und LDR-Spannungsteiler

Grundversion Reaktivlicht mit Watchdog-Abschaltung und LDR-Spannungsteiler

Ralf Pongratz

30. Juni 2012

Inhaltsverzeichnis

I. Die Schaltung	2
1. Funktionsbeschreibung	2
2. Schaltplan	3
3. Programmierung	3
II. Bekannte Probleme	6

Teil I. Die Schaltung

1. Funktionsbeschreibung

Diese Schaltung ist die Grundversion der Reaktivlichter. Sie ist einfach mit nur wenigen Bauteilen zu bauen und ohne Parametrierung einsatzbereit.

Die Helligkeitsmessung erfolgt über einen Fotowiderstand (LDR). Tagsüber fällt die Schaltung in einen Ruhezustand, indem sie inaktiv ist. Es wird lediglich periodisch die Helligkeit abgefragt um festzustellen, wann die Nacht anbricht und die Schaltung sich wieder scharf schaltet. Wird der LDR in diesem Zustand angeleuchtet, gibt die Schaltung

einige Lichtblitze mit der Leuchtdiode (LED) zurück und wartet dann auf eine erneute Aktivierung.

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

2. Schaltplan

Abbildung 1 zeigt den Schaltplan des Reaktivlichtes.

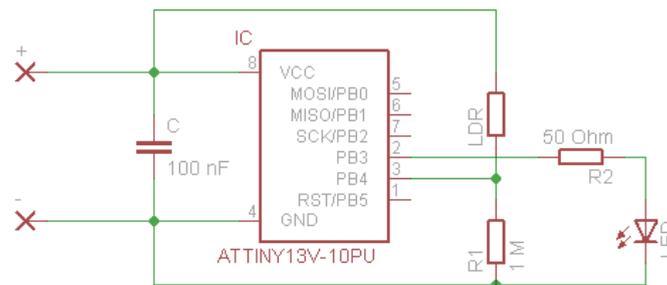


Abbildung 1: Schaltplan.

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.

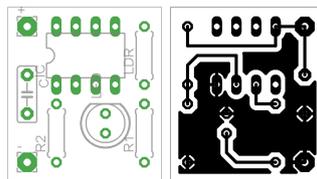


Abbildung 2: Platinenlayout.

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 Wdtcr = &B11010011
10 Enable Interrupts
11
12 Const Schwelle = 50
13 Const Tagschwelle = 800
14 Const Zwangsimpuls = 8
15 Dim A As Byte
16 Dim Tagzaehler As Byte
17 Dim Schlafzaehler As Byte
18 Dim Ldr As Integer
19 Dim Alt As Integer
20 Dim Merker As Integer
21
22 Do
23     Reset Watchdog
24     Powerdown
25     Start Adc
26     Ldr = Getadc(2)
27     Stop Adc
28     Merker = Ldr - Alt
29     Alt = Ldr
30     If Merker > Schwelle Then
31         Gosub Blinken
32     End If
33     If Ldr > Tagschwelle Then
34         If Tagzaehler < 255 Then
35             Tagzaehler = Tagzaehler + 1
36         End If
37     Else
38         Tagzaehler = 0
39     End If
40     If Tagzaehler > 200 Then
41         Gosub Pause
```

```

42     End If
43 Loop
44
45 Blinken:
46     For A = 0 To 10
47         Portb.3 = 1
48         Reset Watchdog
49         Powerdown
50         Portb.3 = 0
51         Reset Watchdog
52         Powerdown
53     Next A
54     Alt = 1023
55 Return
56
57 Pause:
58     Wdtcr = &B11110001
59     Reset Watchdog
60     Powerdown
61     Wdtcr = &B11010011
62     Schlafzaehler = Schlafzaehler + 1
63     If Schlafzaehler = Zwangsimpuls Then
64         Portb.3 = 1
65         Reset Watchdog
66         Powerdown
67         Portb.3 = 0
68         Schlafzaehler = 0
69     End If
70 Return
71
72 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 Eingänge. 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. **Schwelle** gibt die Mindestgröße der Helligkeitsänderung zwischen zwei Durchläufen an, damit das Reaktivlicht ausgelöst wird. Hiermit kann die Empfindlichkeit der Schaltung eingestellt werden. **Tagschwelle** gibt den Wert vor, oberhalb dessen die Schaltung in den Tagmodus wechselt. Um den Tagmodus anzuzeigen, kann mit der Konstante **Zwangsimpuls** festgelegt werden, nach wieviel Schlafzyklen (hier etwa 8 s) ein Impuls auf der LED ausgegeben werden soll. **A** ist eine Zählvariable, die in der Unterprozedur **Blinken** genutzt wird. **Tagzaehler** zählt während des Programmablaufs die aufeinanderfolgenden Zyklen, in denen der Helligkeitswert über der Tagschwelle liegt. **Schlafzaehler** zählt die Schlafzyklen für die Ausgabe des Kontrollimpulses auf der LED. Der aktuelle Helligkeitswert wird in der Variablen **Ldr**, derjenige des vorangegangenen Durchlauf in **Alt** gespeichert. **Merker** ist eine Variable, in der die Differenz zwischen diesen beiden Werten gespeichert wird.

Die Zeilen 22 bis 43 enthalten das Hauptprogramm. Zu Beginn wird der Controller für 0,125 s in den Schlafmodus versetzt. Dadurch wird die Wahrscheinlichkeit einer Fehlauflösung verringert, da beim Wechsel zwischen unbeleuchtetem und beleuchtetem Sensor ein größerer Unterschied ist als bei kontinuierlicher Abfrage. Danach wird in Zeile 25 bis 27 der Analog-Digital-Wandler gestartet, der Helligkeitswert eingelesen und der Wandler wieder ausgeschaltet. Es wird die Differenz zum vorangegangenen Zyklus berechnet und anschließend der jetzige Helligkeitswert für den nächsten Durchlauf in der Variablen **Alt** gespeichert. Ist der Helligkeitsunterschied größer als die eingestellte Schwelle, wird die Unterprozedur **Blinken** aufgerufen. Ab Zeile 33 werden die Bedingungen für den Tagmodus überprüft. Ist der Helligkeitswert über der Tagschwelle, wird der Tagzaehler hochgezählt, ansonsten zurück auf 0 gesetzt. Sollte der Tagzähler den Wert 200 erreichen, was bedeutet, dass über 200 Zyklen a 0,125 s Tag detektiert wurde, wird die Unterprozedur **Pause** aufgerufen.

Die Unterprozedur **Blinken** befindet sich in den Zeilen 45 bis 54. Hier wird zehnmal in Folge die Port 3, an dem die LED angeschlossen ist, angeschaltet, mittels des Watchdog-Timers 0,125 s gewartet, der Ausgang wieder ausgeschaltet und erneut gewartet. Im Anschluss an diesen Vorgang wird der Wert der gemessenen Helligkeit auf den Maximalwert gesetzt, sodass eine Doppelauslösung verhindert wird.

Die Zeilen 57 bis 70 enthalten die Prozedur **Pause**. Damit wird der Microcontroller während des Tagmodus für 8 s ausgeschaltet. 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 und der Schlafzähler inkrementiert. Erreicht der Schlafzähler den Wert, der in der Konstanten **Zwangsimpuls** angegeben ist, wird ein einzelner Lichtblitz auf der LED ausgegeben und der Schlafzähler auf 0 zurückgesetzt.

Teil II.

Bekannte Probleme

Es sind keine Probleme bekannt.