



Ralf Pongratz

Das kleine Einmaleins

Schritt für Schritt zum Reaktivlicht

Ralf Pongratz

Das kleine Einmaleins

Schritt für Schritt zum Reaktivlicht

31. Juli 2011

www.reaktivlicht.de

Inhaltsverzeichnis

1	Einleitung	5
1.1	Warnhinweis	5
1.2	Bestelldaten	5
1.3	Programmieradapter	6
1.4	Programmiersoftware	6
1.4.1	Voraussetzungen	7
1.4.2	Bedienung von „Bascom AVR“	7
2	Aufbau	13
3	Programmierung	15
4	Inbetriebnahme	19

Einleitung

Dieses Buch stellt eine Schritt für Schritt Anleitung dar, mit Hilfe derer ein Reaktivlicht aufgebaut werden kann. Es beruht dabei auf das Kochbuch¹. Der Anwender sollte Grundkenntnisse in der Elektronik und der Programmierung mitbringen.

1.1 Warnhinweis

Diese Anleitung erhebt keinen Anspruch auf Vollständigkeit und Richtigkeit. Jeder, der die hier angegebene Schaltung nachbaut und programmiert, tut dies auf eigene Verantwortung. Was vielen wahrscheinlich nicht klar ist: Ihr bastelt an einem offenen PC rum. Eure Schaltung ist direkt mit der Hauptplatine verbunden. Von der Einkopplung von Störungen und unangepassten Leitungswiderständen abgesehen (ist bei dieser älteren Schnittstelle nicht so wild) kann ein Kurzschluss zwischen einer Signal- und Masseleitung der parallelen Schnittstelle ernsthafte Folgen für den Rechner haben. Also sollte jeder, der keine Erfahrungen auf dem Gebiet der Elektrotechnik hat, im eigenen Interesse sich zuerst eine Tasse Tee und einen bequemen Sessel besorgen und die Anleitung komplett durcharbeiten. Die meisten Probleme lösen sich dadurch automatisch.

1.2 Bestelldaten

Hier sind die Bestelldaten der benötigten Bauteile bei Reichelt angegeben. Wer der Meinung ist, trotz der Liste Hilfe zu benötigen, darf mich gerne

¹ <http://www.reaktivlicht.de>

kontaktieren². Gegen Übernahme der Portokosten bin ich bereit, Euch die benötigten Bauteile zum Unkostenpreis zu überlassen.

Bauteil	Reichelt
ATtiny 13V-10PU Sockel DIL 8-polig	ATTINY 13V-10PU GS8
Widerstand 56 Ω	1/4 W 56 Ω
Widerstand 220 Ω	1/4 W 220 Ω
Widerstand 1 M Ω	1/4 W 1 M
Kondensator 100 nF	MKS-2 100N
Leuchtdiode grün 13000 mcd	LED 5-13000 GN
Leuchtdiode weiß 18000 mcd	LED 5-18000 WS
SUB-D Stecker 25 polig	D-SUB ST 25
Kappe für Stecker	KAPPE CG25G
Fotowiderstand	A 905014 A 906014

Tabelle 1.1. Bestelldaten der Bauteile

1.3 Programmieradapter

Zum Überspielen des Programms auf den Microcontroller ist ein Anschluss an einen PC notwendig. Am einfachsten geschieht das Programmieren über die parallele Schnittstelle. Abbildung 1.1 zeigt einen einfachen Adapter hierfür.

X1 ist ein 25-poliger SUB-D-Stecker, der an die parallele Schnittstelle des PCs angeschlossen wird. IC1 ist der zu programmierende Microcontroller. Ferner werden noch zwei Widerstände mit dem Wert 220 Ohm benötigt. Die Versorgungsspannung erhält der Microcontroller über eine externe Spannungsquelle. Das Kabel zum PC sollte möglichst kurz und abgeschirmt sein, damit keine Störungen in die Leitungen einkoppeln.

Sofern Pin 1 nicht genutzt wird und die Pins 5, 6 und 7 nicht auf feste Potentiale gezogen oder untereinander verbunden werden, kann der Programmieradapter in die Schaltung des Reaktivlichts integriert werden. So entfällt ein Umstecken des ICs bei der Programmierung.

1.4 Programmiersoftware

Für die Programmierung des Microcontrollers sind verschiedene Programme auf dem Markt. Hier wird das Programm „Bascom AVR“ mit dem parallelen

² kontakt@reaktivlicht.de

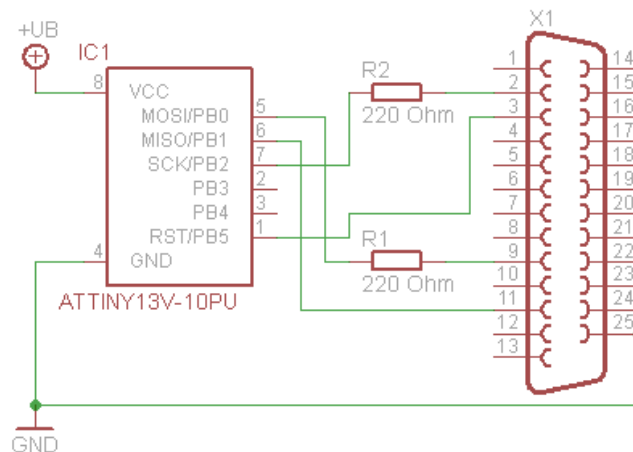


Abb. 1.1. Programmieradapter für die parallele Schnittstelle.

Programmieradapter (Kap. 1.3) benutzt. Eine kostenlose Demoversion, bei der der Codeumfang auf 4 kB begrenzt ist, ist im Internet verfügbar³.

1.4.1 Voraussetzungen

Benötigt wird ein PC mit einer parallelen Schnittstelle (Druckerschnittstelle), der parallele Programmieradapter und die Programmierumgebung, mit der die Programme geschrieben und auf den Prozessor übertragen werden können.

Die parallele Schnittstelle muss im BIOS des Rechners auf ECP+EPP (In- und Output) eingestellt werden. Hier eine Beispieleinstellung:

Onboard Parallel Port: 378/IRQ7

Parallel Port Mode: ECP+EPP

ECP Mode Ust DMA: 3

Parallel Port EPP Type: EPP1.7

1.4.2 Bedienung von „Bascom AVR“

Alle Angaben basieren auf die Programmversion 1.11.8.3. Neuere Versionen können davon abweichen.

Zu Beginn wird „Bascom AVR“ gestartet und im Menü „File“ mittels „New“ ein neues Programmfenster erstellt. Danach werden der Programmieradap-

³ http://www.mcselec.com/index.php?option=com_docman&task=cat_view&gid=99&Itemid=54

ter und die Chip-Parameter eingestellt. Dazu wird im Menü „Options“ der Eintrag „Programmer“ ausgewählt.

Nun wird der Reiter „Compiler“ und darin der Reiter „Chip“ (Abb. 1.2) aufgerufen. Im Drop-Down-Menü „Chip“ wird der Eintrag „attiny13.dat“ ausgewählt und in die darunterliegenden Felder folgende Werte eingetragen:

HW Stack = 2

Soft Stack = 8

Framesize = 24

Anschließend werden die Werte durch Klicken auf „Default“ gespeichert.

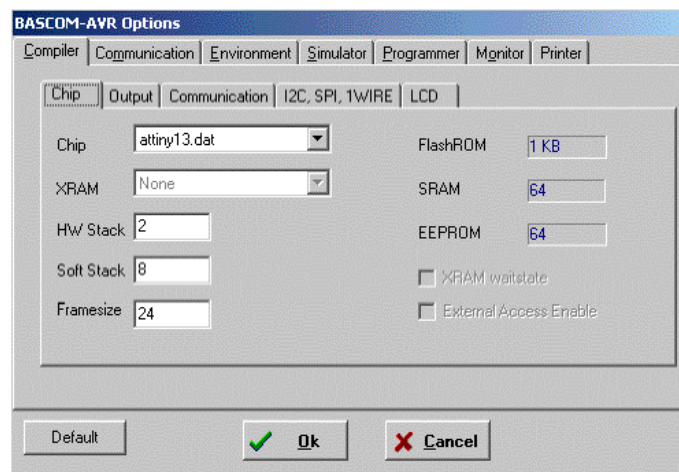


Abb. 1.2. Einstellen der Chip-Parameter.

Danach wird im Reiter „Programmer“ im Drop-Down-Menü „Programmer“ der Eintrag „Universal MCS Interface“ ausgewählt (Abb. 1.3). Im Reiter „Universal“ wird als Programmer „WinAVR and SP12“ ausgewählt. Anschließend wird der Dialog durch Klicken auf „OK“ geschlossen.

Als nächstes werden die Fuse-Bits eingestellt. Dies sind Speicherzellen, die das Grundverhalten des Microcontrollers festlegen. Dazu klickt man in der Icon-Leiste auf den kleinen grünen IC-Sockel und wählt den Punkt „Manual Program,“ aus (Abb. 1.4).

Im sich öffnenden Dialogfenster klickt man auf den Reiter „Lock and Fuse Bits“. Nun liest Bascom die Einstellungen der Fuse-Bits aus dem Controller aus und zeigt sie an (Abb. 1.5).

Um Veränderungen an den Fuse-Bits vorzunehmen, klickt man auf die entsprechende Zeile, in der sich das gewünschte bzw. zu ändernde Bit befindet. In dem sich öffnenden Drop-Down-Menü kann daraufhin der gewünschte Wert

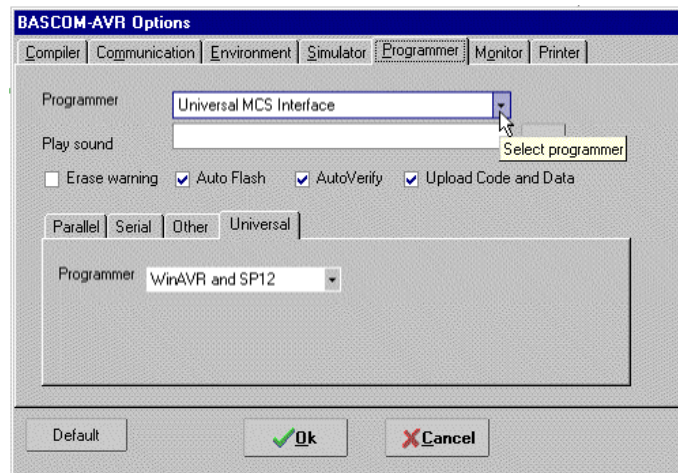


Abb. 1.3. Einstellen des Programmieradapters.

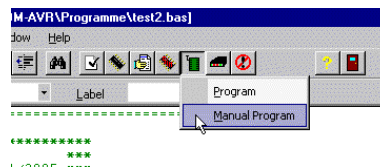


Abb. 1.4. Einstellen der Fuse-Bits.

ausgewählt werden. Folgende Einstellungen müssen hier vorgenommen werden:

Fusebit DCBA auf „1101:Int. Osc. 128 kHz; start-up time: 14 CK + 64 ms“

Fusebit E auf „1:Divide clock by 8, OFF“

Bitte die Einstellungen noch einmal sehr genau kontrollieren. Ist alles richtig, rechts auf „Write FS“ klicken. Die geänderten Einstellungen werden daraufhin in den Controller geschrieben.

Anschließend wird der Quellcode des Programms geschrieben. Ist dies erledigt, erfolgt die Kompilierung. Dazu im Menü „Program“ den Punkt „Compile“ auswählen (Abb. 1.6). Der Compiler startet nun. Enthält das Programm Fehler, wird eine entsprechende Meldung in der Fußzeile ausgegeben. Wurde das Programm erfolgreich kompiliert, kann es auf den Microcontroller übertragen werden. Dazu wird im Menü „Program“ auf den Punkt „Send to Chip“ geklickt. Im sich öffnenden Fenster muss dann aus dem Menü „Chip“ der Eintrag „Autoprogram“ aufgerufen werden (Abb. 1.7). Das zuvor kompilierte Programm wird nun in den Programmspeicher des Controllers geschrieben.

Aufbau

Abbildung 2.1 zeigt den Schaltplan des Reaktivlichtes.

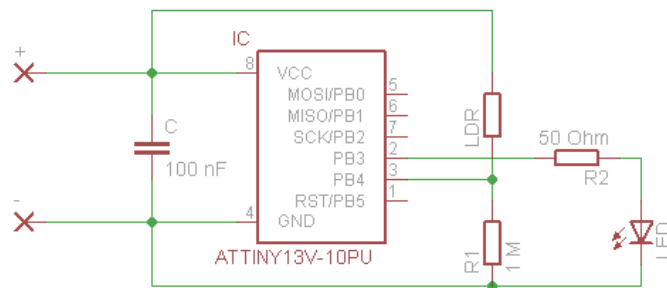


Abb. 2.1. Schaltplan.

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.

Ein Entwurf für eine Platine ist in Abbildung 2.2 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.

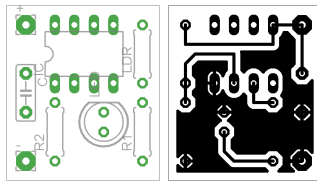


Abb. 2.2. Platinenlayout.

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

```

Das Listing zeigt den Programmcode. Dieser ist in die Programmieroberfläche zu kopieren, zu kompilieren und auf den Microcontroller zu spielen. Wie dies

funktioniert, ist in Kapitel 1.4.2 erklärt. Für die Funktionsweise des Programms ist das Kochbuch zur Rate zu ziehen. Dort ist es ausführlich erklärt.

Inbetriebnahme

Nach dem Anschließen der Batterie startet das Programm automatisch. Ist es hell, verfällt der Controller in einen Schlafmodus, in dem er sich alle 64 Sekunden mit einem einzelnen Blinken meldet. Wird es dunkel, kann die Schaltung mittels einer Lichtquelle angetriggert werden. Dann blinkt sie zehn Mal zurück.