



Ralf Pongratz

Basic Version

Reactive Light with Watchdog-Deactivation
and LDR-Voltage Divider

Basic Version

Reactive Light with Watchdog-Deactivation and LDR Voltage Divider

Ralf Pongratz

8. November 2012

Inhaltsverzeichnis

I. The circuit	2
1. Functional description	2
2. Circuit diagram	2
3. Programming	3
II. Known problems	6

Teil I.

The circuit

1. Functional description

This circuit is the basic version of the reactive lights. It is easy to make with just a few elements and usable without parameterization.

The measurement of brightness is done by a photo resistor (LDR). During daylight the circuit goes into a standby mode and is inactive. It just measures periodically the brightness to detect the beginning of the night and set the circuit into an active mode. If in this mode the LDR is lit, the circuit will blink several times using the light emitting diode (LED) and wait for the next activation.

Because of the very small power consumption the circuit can be run for years with just a set of batteries.

2. Circuit diagram

Figure 1 shows the circuit diagram of the reactive light.

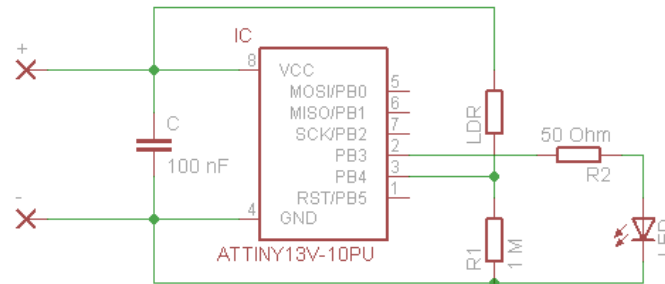


Abbildung 1: Circuit diagram.

In the center one can find the microcontroller, left of it the power supply. The circuit needs a voltage of 3 V. Two standard batteries in sequence make this voltage. The plus terminal must be connected to pin +, the minus terminal to the pin -. In the right of the IC the measurement of the brightness, consisting of R1 and the photo resistor LDR, and the LED for output of the blinking sequence with the resistor R2 are located.

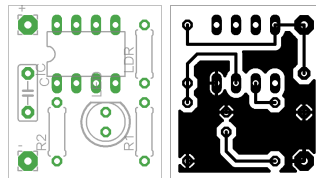


Abbildung 2: Layout of the circuit board.

A draft of the circuit board is shown in figure 2. The left figure shows the component side, the right one the circuit paths seen from the lower side. When mounting one has to care about the polarity of the LED (the flattened side respectively the short leg has to point to the IC) and the IC (the denting has to point to the capacitor).

3. Programming

```
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 Treshold = 50
13 Const DaylightTreshold = 800
14 Const ForceImpulse = 8
15 Dim A As Byte
16 Dim DaylightCounter As Byte
17 Dim SleepingCounter As Byte
18 Dim Ldr As Integer
19 Dim Alt As Integer
20 Dim Marker As Integer
21
22 Do
23     Reset Watchdog
24     Powerdown
25     Start Adc
26     Ldr = Getadc(2)
27     Stop Adc
28     Marker = Ldr - Alt
29     Alt = Ldr
30     If Marker > Treshold Then
31         Gosub Blink
32     End If
33     If Ldr > DaylightTreshold Then
34         If DaylightCounter < 255 Then
35             DaylightCounter = DaylightCounter + 1
36         End If
37     Else
38         DaylightCounter = 0
39     End If
40     If DaylightCounter > 200 Then
41         Gosub Pause
42     End If
43 Loop
44
45 Blink:
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     SleepingCounter = SleepingCounter + 1
63     If SleepingCounter = ForceImpulse Then
64         Portb.3 = 1
65         Reset Watchdog
66         Powerdown
67         Portb.3 = 0
68         SleepingCounter = 0
69     End If
70 Return
71
72 End

```

In line 1 to 3 general settings are done. First the type of the processor is told to the compiler. Afterwards the frequency of the intern oscillator is set. At last the stack is set to 2 so that there is enough space for the variables of the program. The consequence is that the nesting depth of function calls can be maximum two, which is enough for this program.

Then the microcontroller is configured by the registers. First the analog digital converter that is used to measure the brightness is configured. Afterwards port 3 is configured as an output and set to low. The other ports are inputs. In line 8 the analog comparator is switched off to reduce the power consumption. It is not used in this program. Line 9 and 10 configure the watchdog timer that is used for realizing the delays to 0.125 s and interrupt mode.

Now the constants and variables of the program are defined. **Threshold** is the minimum change of the brightness between two cycles that triggers the reactive light. By changing this value the sensitivity can be adjusted. If the brightness is above **DaylightThreshold**, the controller will go to standby mode. To show this mode, the constant **ForceImpulse** defines after how many standby cycles (about 8 seconds) an impulse is output by the LED. **A** is a counter variable that is used in the procedure **Blink**. **DaylightCounter**

counts the consecutive cycles with the brightness above `DaylightThreshold`. `SleepingCounter` counts the standby cycles for the output of the control impulse on the LED. The current brightness is stored in `Ldr`, the one of the previous cycle in `Alt`. `Marker` is a variable for the difference between these two values.

The lines 22 to 43 contain the main program. At the beginning the controller is set to standby for 0.125 s. That's the interval for calculating the difference of the brightness between two cycles. Afterwards the analog digital converter is started in lines 25 to 27, the value of brightness read and the converter switched off again. The difference to the previous cycle is calculated and the actual brightness stored at the variable `Alt` for the next cycle. If the difference is bigger than the threshold, the routine `Blink` is called. As of line 33 the conditions for the standby mode are checked. If the brightness is bigger than the daylight threshold, the daylight counter is incremented. Otherwise it is set to 0. If the daylight counter reaches the value 200, which means that during 200 cycles at 0.125 s daylight was detected, the procedure `Pause` is called.

The procedure `Blink` is in the lines 45 to 54. Ten times the port 3, where the LED is connected to, is switched on. The program waits for 0.125 s using the watchdog timer and afterwards switches off the port and waits again. At the end the measured brightness is set to the maximum value in order to prevent the program from being triggered double.

The lines 57 to 70 contain the procedure `Pause`. The microcontroller uses it to go to standby for 8 s during standby mode. At the beginning the watchdog timer is set to the maximum value. Afterwards the controller is set to standby. When it awakes again, the time of the watchdog timer is reset and the sleeping counter incremented. If the sleeping counter reaches the value `ForceImpulse`, a single blink is output by the LED and the sleeping counter reset to 0.

Teil II.

Known problems

No problems are known.