

Fahrradrücklicht mit Reaktiv-Modus

Revision der Revision

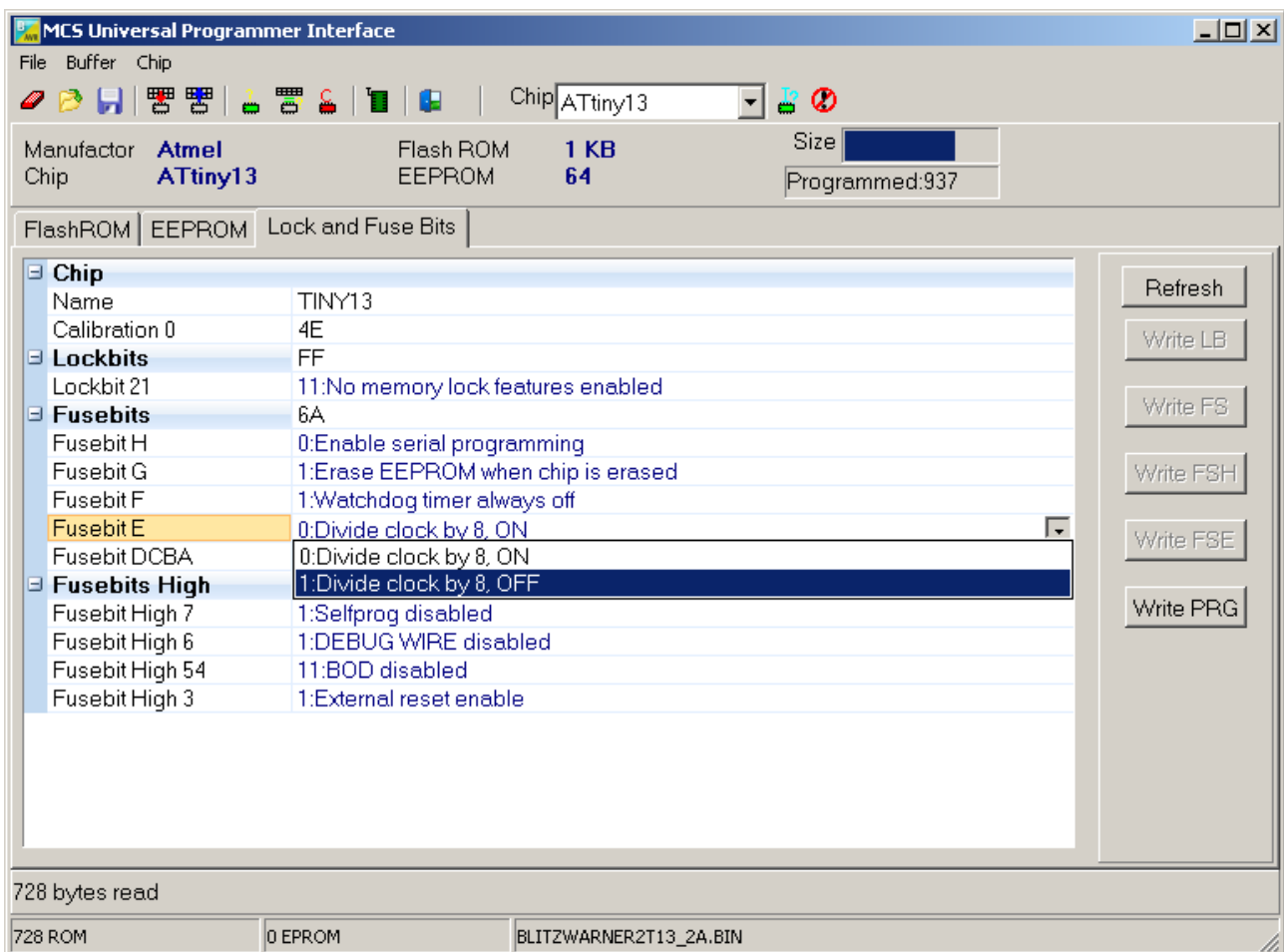
von Martin Müller

Habe mich total gefreut mal eine Reaktion und kritische Würdigung eines von mir schon vor längere Zeit veröffentlichten Projekts erhalten. Hr. Krueger hat sich die Mühe gemacht, das genannte Rücklicht in leicht modifizierter Weise nachzubauen und auszutesten.

Etwas überrascht war ich dann aber doch zu lesen, dass das nachgebaute Lichtchen nicht ganz so funktioniert wie meine Version.

Nach kritischer, ergebnisoffener Überprüfung des ursprünglichen und von Hr. Krueger modifizierten Quellcodes bin ich zu dem Ergebnis gekommen, dass die Ursache für einen Teil der Funktionsabweichungen in der Verwendung unterschiedlicher Taktfrequenzen zu suchen ist. Werksmässig ist der Tiny 13 mit einer Frequenz von ca. 1,2 MHz (9,6 MHz / 8) unterwegs.

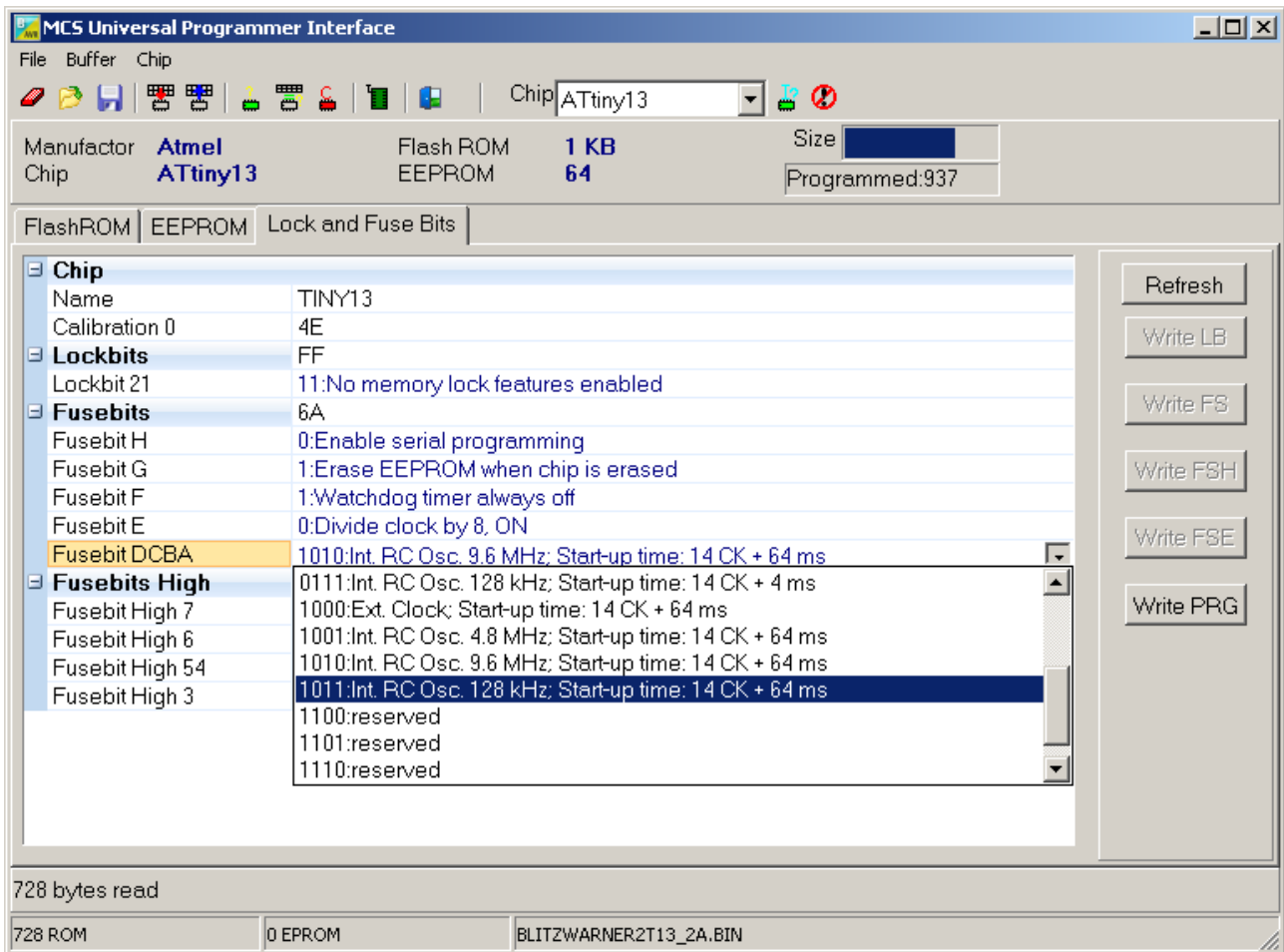
In der veröffentlichten Ursprungsversion arbeitet der Oszillator des Tinys mit ca. 128 kHz. Um diese Frequenz zu erhalten müssen die Fusebits E und DCBA verändert werden.



The screenshot shows the MCS Universal Programmer Interface window. The 'Chip' dropdown is set to 'ATtiny13'. The manufacturer is 'Atmel' and the chip is 'ATtiny13'. The Flash ROM is 1 KB and the EEPROM is 64 bytes. The 'Programmed' count is 937. The 'Lock and Fuse Bits' tab is active, showing a list of fusebits and their values. The 'Fusebit E' is highlighted, and its value is '0: Divide clock by 8, ON'. The 'Fusebit DCBA' is also highlighted, with a value of '0: Divide clock by 8, ON'. The 'Fusebits High' section is expanded, showing values for Fusebit High 7, 6, 54, and 3.

Category	Parameter	Value
Chip	Name	TINY13
Chip	Calibration 0	4E
Lockbits	Lockbit 21	11: No memory lock features enabled
Fusebits	Fusebit H	0: Enable serial programming
Fusebits	Fusebit G	1: Erase EEPROM when chip is erased
Fusebits	Fusebit F	1: Watchdog timer always off
Fusebits	Fusebit E	0: Divide clock by 8, ON
Fusebits	Fusebit DCBA	0: Divide clock by 8, ON
Fusebits High	Fusebit High 7	1: Selfprog disabled
Fusebits High	Fusebit High 6	1: DEBUG WIRE disabled
Fusebits High	Fusebit High 54	11: BOD disabled
Fusebits High	Fusebit High 3	1: External reset enable

Zunächst wird die Frequenzteilung f/8 ausgeschaltet.



Dann wird die gewünschte Oszillatorfrequenz (128 kHz) ausgewählt.

Anschließend werden die geänderten Fusebits in den Tiny geschrieben und selbiger dann mit dem zugehörigen Programm geflashed.

Die Originalversion funktioniert mit diesem Setting und insgesamt 3 angeschlossenen LEDs einwandfrei.

Nachfolgend möchte ich hier noch auf einige Anmerkungen zu den von Herrn Krueger durchgeführte Modifikationen machen:

'Original-Version: rucklichtx.bas, Fahrrad-Rücklicht mit Reaktiv-Modus, von Martin Müller 09.04.2012.

'Korrektur: Ruecklicht-V1.bas / Peter Krueger, 18.06.2014

'Verwendeter uController: ATtiny13V-10PU (UB-min:1V8, UB-max:5V5) Flash-ROM wird 991 von 1023 Byte belegt!

'Strombedarf im Idle Mode: 296uA. Laut Datenblatt < 0,1uA (Wieso ? Hat jemand eine Idee ?)

Das mir bekannte Datenblatt für den ATtiny13/ATtiny13V enthält folgende Daten:

- Active Mode: 1 MHz, 1.8V: 240 µA
- Power-down Mode: < 0.1 µA at 1.8V

Also < 0,1µA bei 1,8V und Powerdown nicht bei Idle.

'Zeile 20: Reference = "Avcc" Eine rote LED bringt im Dämmerlicht bereits eine Spannung von > 1V (max 1V5)

'bei einer Ref von 1V1 wäre die ADC-Wandlung schon am Anschlag > 1023, die ADC-Reglschleife nicht mehr wirksam

Wieviel Spannung bei Lichteinfall eine LED abgibt, hängt vom verwendeten LED-Typ ab. Da gibt's eine große Streubreite. 1 V bei Dämmerlicht finde ich erstaunlich viel. Bei der Entwicklung des Programms wurde davon ausgegangen, dass das Rücklicht bei Dunkelheit betrieben wird. Eine gute Ansprechempfindlichkeit ergab sich mit den von mir verwendeten LEDs bei einer Referenzspannung von 1,1V für den ADC.

'Zeile 42: Mit "Powerdown" kein Startup möglich. Deshalb "Powerdown" ersetzt durch "Idle"

Wie schon Ralf Beesner geschrieben hat, kann es problematisch bis unmöglich sein, den Tiny flankengesteuert mit INT0 aus Powerdown zu erwecken. Bei den von mir aufgebauten zwei Exemplaren funktioniert es bei einer Taktrate von 128 kHz merkwürdigerweise zuverlässig. Um die Betriebssicherheit zu erhöhen, ist es sicher besser die Zeile

Config Int0 = Rising

durch

Config Int0 = Low Level

zu ersetzen. Vielen Dank für diesen Hinweis!

'Zeile 130: Portb = &B00101010 'PB3.1, Start Zeitfenster LED1 = Input, Helligkeit

'Zeile 140: Portb = &B00100010 'PB3.0, Ende Zeitfenster LED1 = Input, Helligkeit

'Befehl: "Waitms 3". Ich hätte eigentlich eine Verzögerung von 3ms erwartet, Gemessen 450us

Das ist sicher auf eine höhere Taktfrequenz (128 kHz vs. 1,2 MHz) zurückzuführen.

'Zeile 146: Zeitschleife "For Anzahl = 1 To 10 '3 LED für etwa 10 Sekunden" auf "1 To 80" erhöht

Ursprünglich hieß es „1 TO 10“ wenn man das um den Faktor 8 erhöht, deutet das auch wieder auf eine zu hohe Taktfrequenz hin.

'Zeile 153: "Anzah" ersetzt durch "Anzahl"

„Anzah“ steht hinter einem „Next“-Befehl (For-Next-Schleife). Bascom ordnet „Next“ automatisch dem zugehörigen „For“ zu. Es ist also unerheblich, ob man „Next Anzah“, „Next Anzahl“ oder nur einfach „Next“ schreibt.