

Atmel AVR RISC μ C unter GNU/Linux programmieren

Jan Grosser
email@jan-grosser.de

Technik-Club, Bonn, 05.03.2013

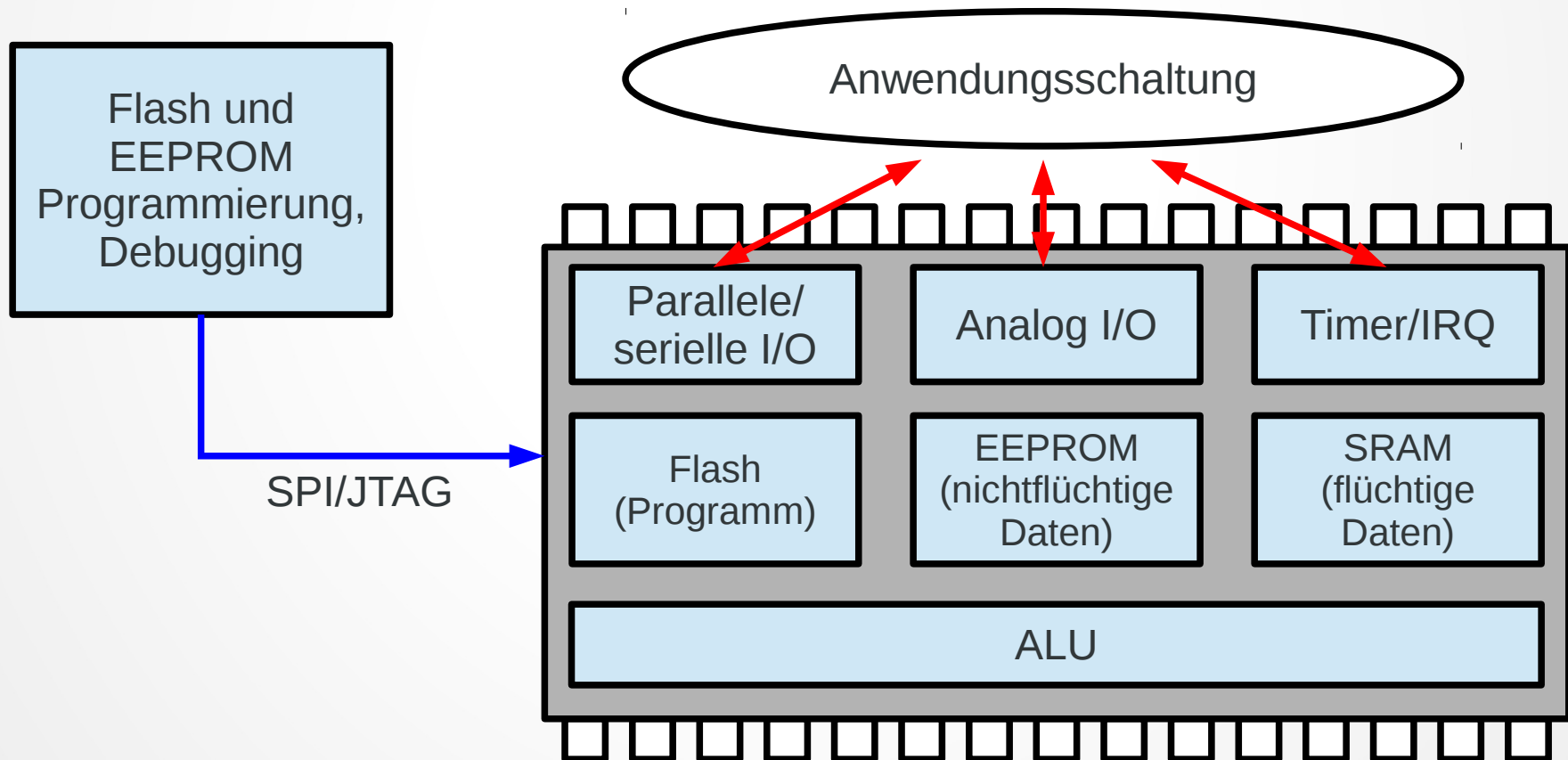


Inhalt

- Mikrocontroller
- Assembler
- Was wird benötigt?
 - Hardware
 - Software
- Beispiel-Workflow (simple-io.asm)
- Probleme/Lösungen
- Lesenswert

Mikrocontroller

- μ C sind leistungsfähige Mini-Computer
- Flexibel, da re-programmierbar



Assembler

- Menschenlesbare Darstellung von Maschinensprache (Mnemos statt Opcode)
- Gegenteil: Generalisierte Hochsprache wie bspw. C
- Vorteile
 - Direkte Programmierung der Ziel-Hardware
 - Effizienter als Hochsprachen
- Nachteile
 - Realisierung komplexer Funktionen aufwändig
 - Schlechte Portierbarkeit

Was wird benötigt?

- Hardware
 - Evaluations-Board
 - Programmer
 - Steckbrett, Elektronik-Kleinkram
- Software
 - Compiler
 - Programmier-Software für μC

Hardware: Evaluationsboard

- **ATMEL Evaluations-Board** für < 30€ bei www.pollin.de
- PoC/Entwicklung
- Grundl. Beschaltung (z.B. Power-Versorgung)
- Peripherie (Schalter, LEDs, RS232, ...)
- Sockel für μC oder festverlöteter μC
- Eigene Schaltung läßt sich verbinden

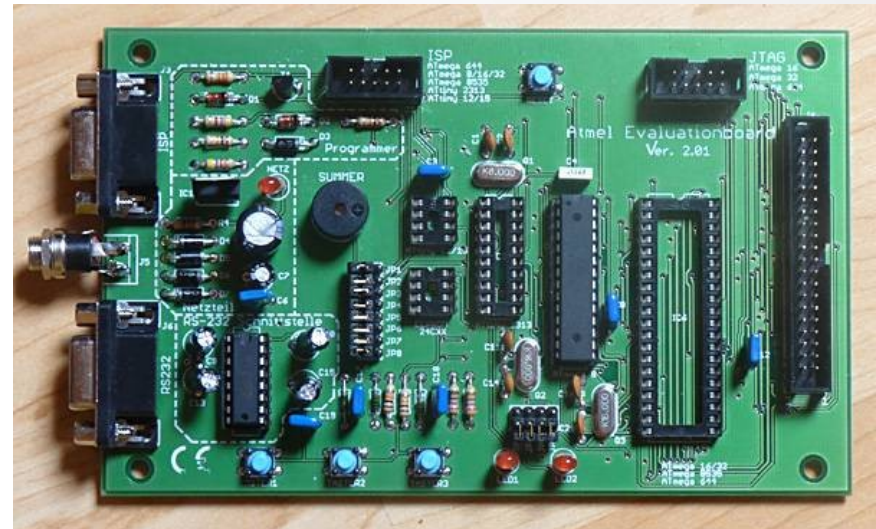
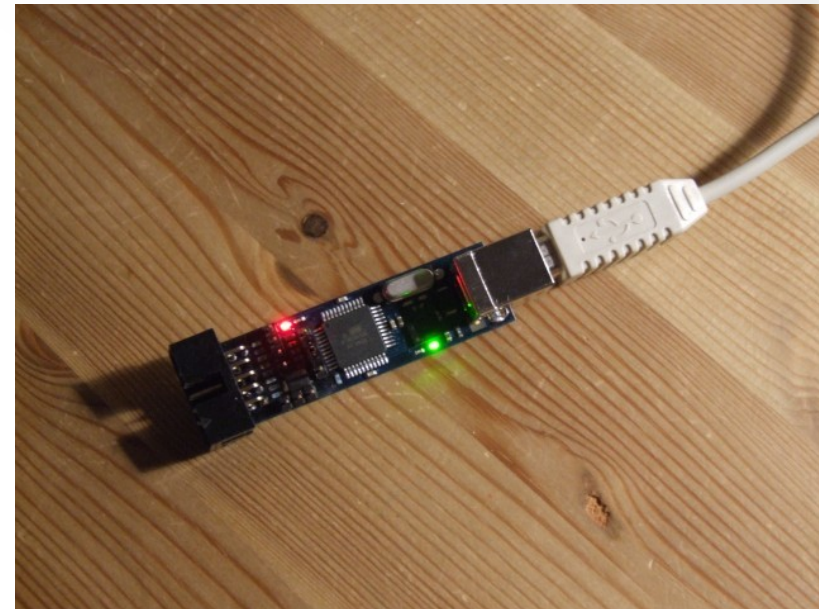


Bild: L. Höll (www.mydarc.de)

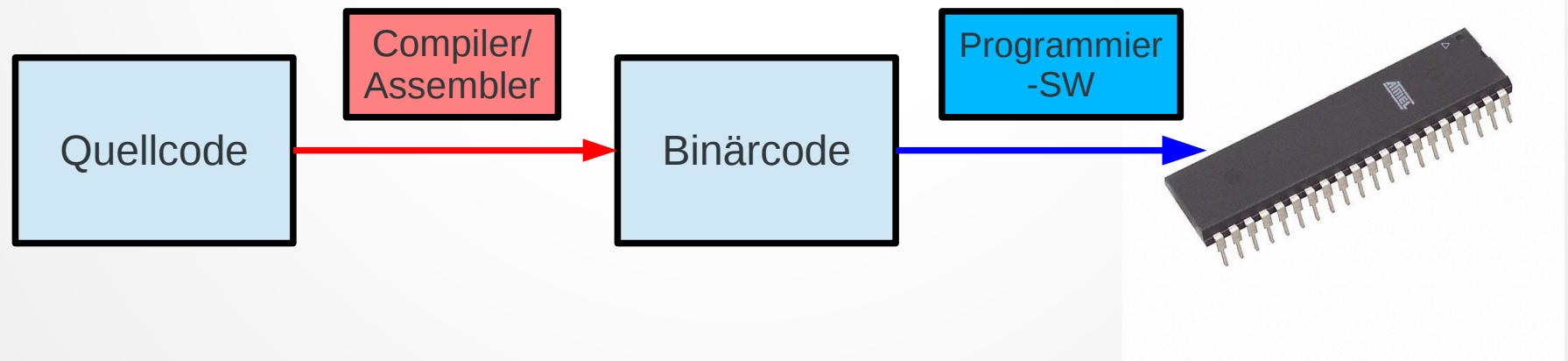
Hardware: Programmer

- Embedde Projects **USBprog** für < 30€
- Kompatibel AVRISP mk2
- Verbindung PC <--> μ C
- Schreiben und Lesen des Flash und EEPROM
- Frei programmierbar, d.h. vielfältig einsetzbar
 - In-System-Programmer/Serial Programming IF
 - JTAG Programmer/Debugger
 - USB-RS232-Wandler
 - ...



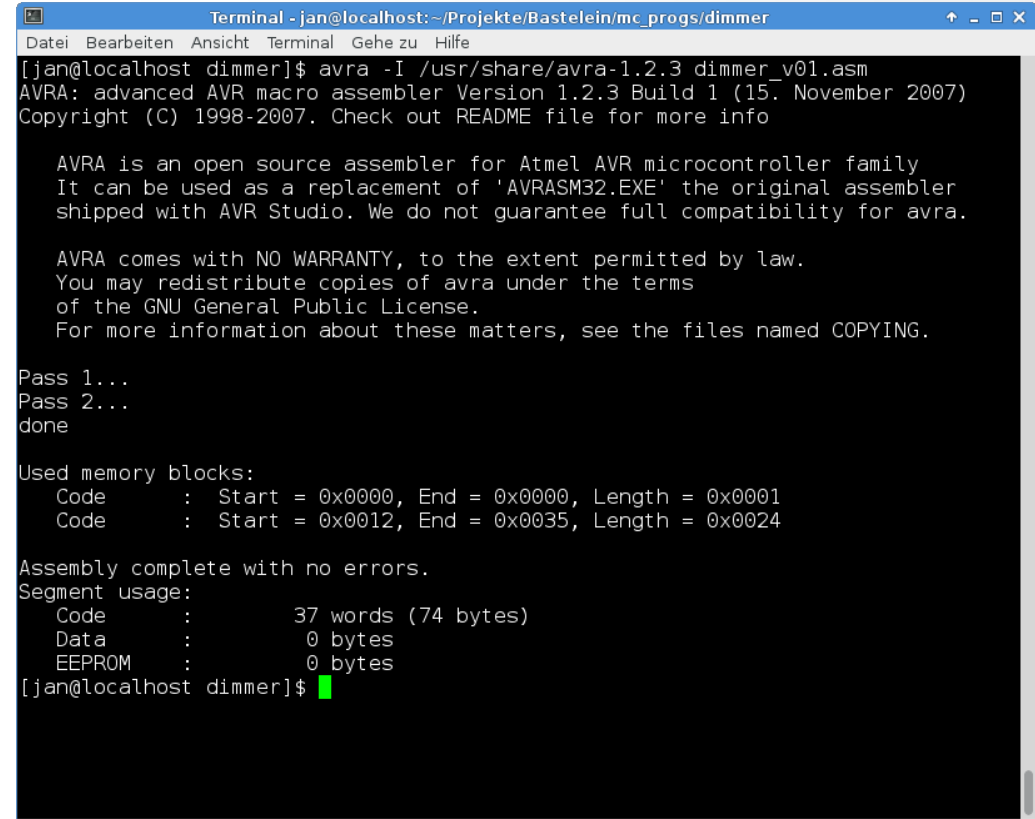
Software: Compiler

- Atmel AVR RISC μ C in mehreren Sprachen programmierbar:
 - Assembler
 - C
 - BASIC
 - ...



Software: Compiler

- **AVRA**: Open Source Assembler für Atmel μ C
- Kompatibel zu Atmels AVRASM
- Erzeugt Maschinencode
- Kommandozeilen-Programm



```
Terminal - jan@localhost: ~/Projekte/Bastelein/mc_progs/dimmer
Datei Bearbeiten Ansicht Terminal Gehe zu Hilfe
[jan@localhost dimmer]$ avra -I /usr/share/avra-1.2.3 dimmer_v01.asm
AVRA: advanced AVR macro assembler Version 1.2.3 Build 1 (15. November 2007)
Copyright (C) 1998-2007. Check out README file for more info

AVRA is an open source assembler for Atmel AVR microcontroller family
It can be used as a replacement of 'AVRASM32.EXE' the original assembler
shipped with AVR Studio. We do not guarantee full compatibility for avra.

AVRA comes with NO WARRANTY, to the extent permitted by law.
You may redistribute copies of avra under the terms
of the GNU General Public License.
For more information about these matters, see the files named COPYING.

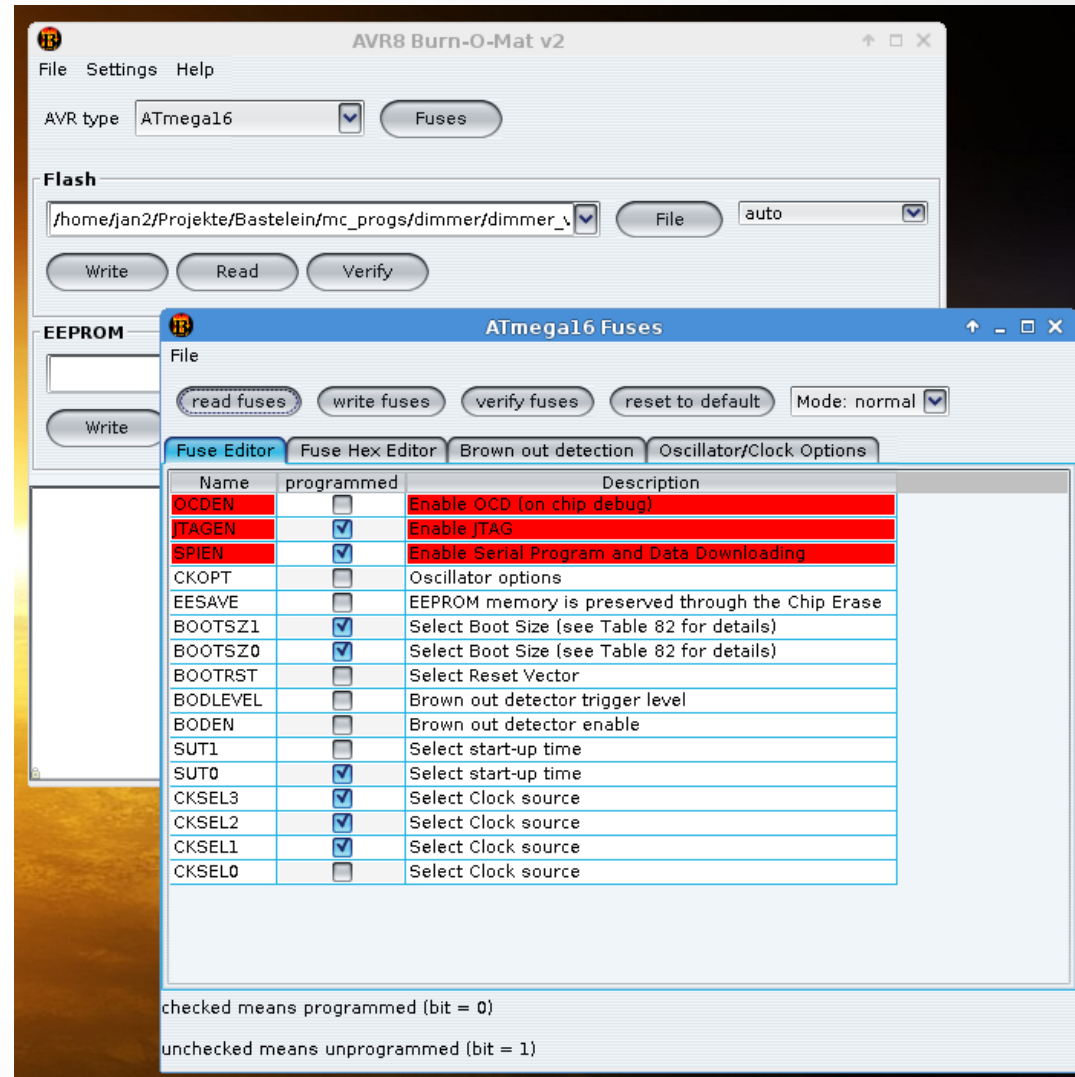
Pass 1...
Pass 2...
done

Used memory blocks:
Code      : Start = 0x0000, End = 0x0000, Length = 0x0001
Code      : Start = 0x0012, End = 0x0035, Length = 0x0024

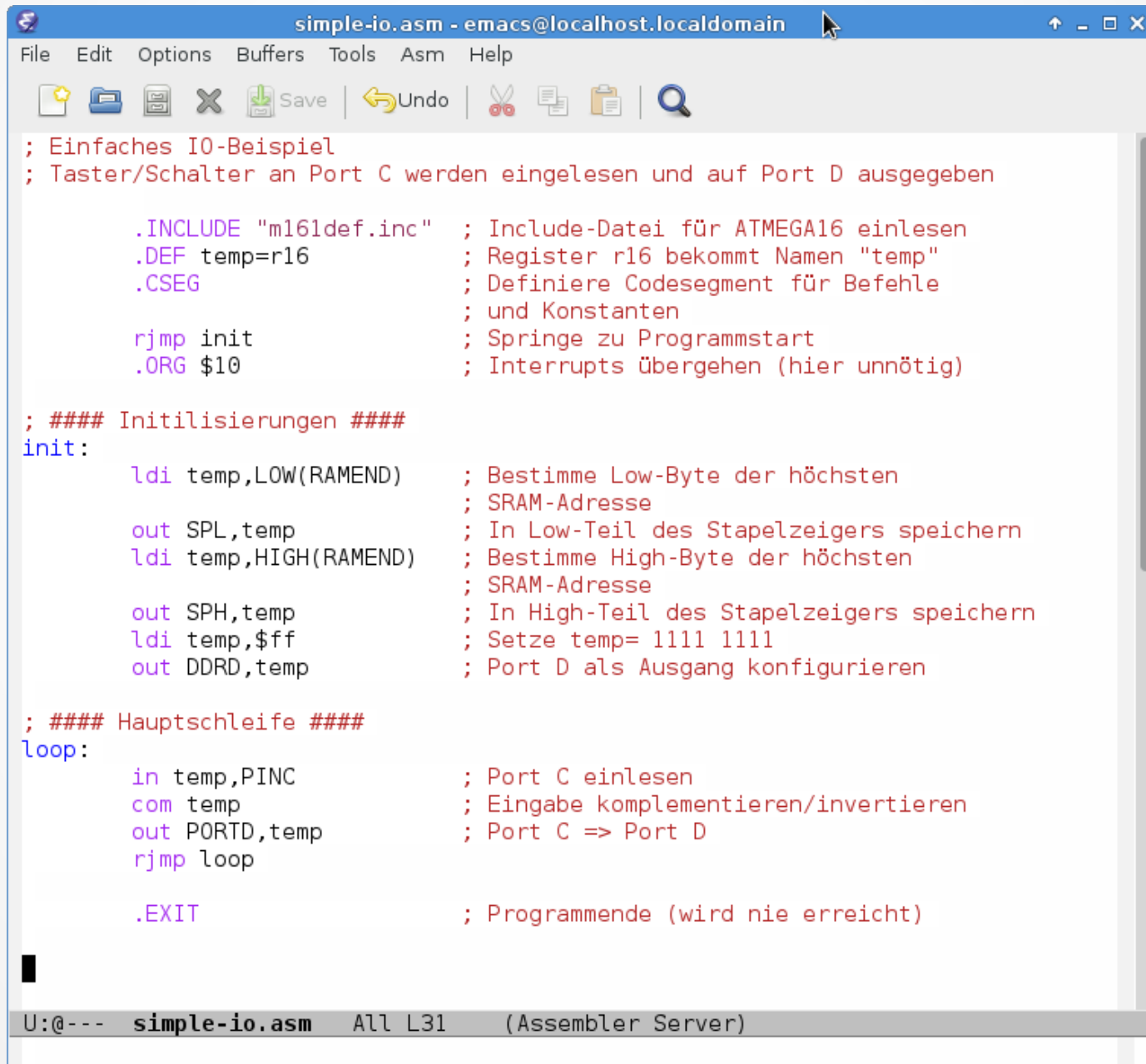
Assembly complete with no errors.
Segment usage:
Code      :          37 words (74 bytes)
Data      :           0 bytes
EEPROM    :           0 bytes
[jan@localhost dimmer]$
```

Software: Programmier-SW

- **Avrdude**: Open Source Programmer
- **AVR8 Burn-O-Mat** GUI für avrdude
- Hilft insbes. beim Setzen der Fuses



Beispiel-Workflow: Quellcode



```
; Einfaches IO-Beispiel
; Taster/Schalter an Port C werden eingelesen und auf Port D ausgegeben

.INCLUDE "m161def.inc" ; Include-Datei für ATMEGA16 einlesen
.DEF temp=r16          ; Register r16 bekommt Namen "temp"
.CSEG                 ; Definiere Codesegment für Befehle
                     ; und Konstanten
rjmp init             ; Springe zu Programmstart
.ORG $10              ; Interrupts übergehen (hier unnötig)

; ##### Initilisierungen #####
init:
    ldi temp,LOW(RAMEND) ; Bestimme Low-Byte der höchsten
                        ; SRAM-Adresse
    out SPL,temp         ; In Low-Teil des Stapelzeigers speichern
    ldi temp,HIGH(RAMEND) ; Bestimme High-Byte der höchsten
                        ; SRAM-Adresse
    out SPH,temp         ; In High-Teil des Stapelzeigers speichern
    ldi temp,$ff         ; Setze temp= 1111 1111
    out DDRD,temp        ; Port D als Ausgang konfigurieren

; ##### Hauptschleife #####
loop:
    in temp,PINC         ; Port C einlesen
    com temp             ; Eingabe komplementieren/invertieren
    out PORTD,temp       ; Port C => Port D
    rjmp loop

.EXIT                  ; Programmende (wird nie erreicht)
```

U:@--- simple-io.asm All L31 (Assembler Server)

Beispiel-Workflow: Assemblieren

```
Terminal - jan@localhost:~/Projekte/Bastelein/mc_progs/simple-io
Datei Bearbeiten Ansicht Terminal Gehe zu Hilfe
[jan@localhost simple-io]$ avra -I /usr/share/avra-1.2.3 simple-io.asm
AVRA: advanced AVR macro assembler Version 1.2.3 Build 1 (15. November 2007)
Copyright (C) 1998-2007. Check out README file for more info


AVRA is an open source assembler for Atmel AVR microcontroller family
It can be used as a replacement of 'AVRASM32.EXE' the original assembler
shipped with AVR Studio. We do not guarantee full compatibility for avra.

AVRA comes with NO WARRANTY, to the extent permitted by law.
You may redistribute copies of avra under the terms
of the GNU General Public License.
For more information about these matters, see the files named COPYING.

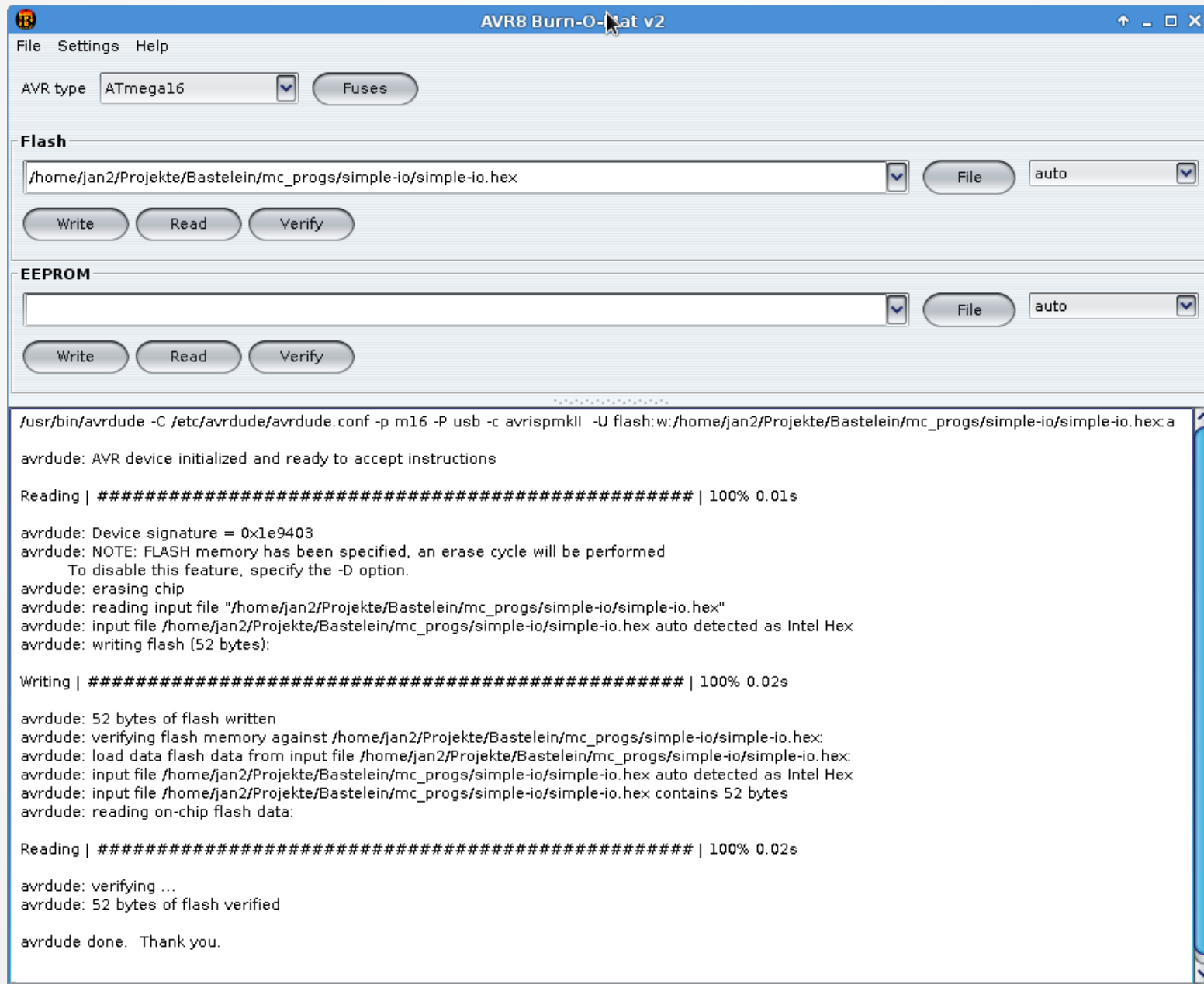
Pass 1...
Pass 2...
done

Used memory blocks:
  Code      : Start = 0x0000, End = 0x0000, Length = 0x0001
  Code      : Start = 0x0010, End = 0x0019, Length = 0x000A

Assembly complete with no errors.
Segment usage:
  Code      :      11 words (22 bytes)
  Data      :       0 bytes
  EEPROM    :       0 bytes
[jan@localhost simple-io]$ ls -l
simple-io.asm
simple-io.asm~
simple-io.cof
simple-io.eep.hex
simple-io.hex
simple-io.obj
[jan@localhost simple-io]$
```



Beispiel-Workflow: Flashen



Probleme/Lösungen: AVRA

- Die mit AVRA mitgelieferten Include-Dateien in /usr/share/avra-1.2.3 funktionieren nicht
- Das sind die Includes für Atmels AVRASM², AVRA braucht aber die von AVRASM¹.
- Richtige Includes kann man sich aus Installation von Atmels AVR Studio „klauen“ oder [hier](#) als ZIP herunterladen

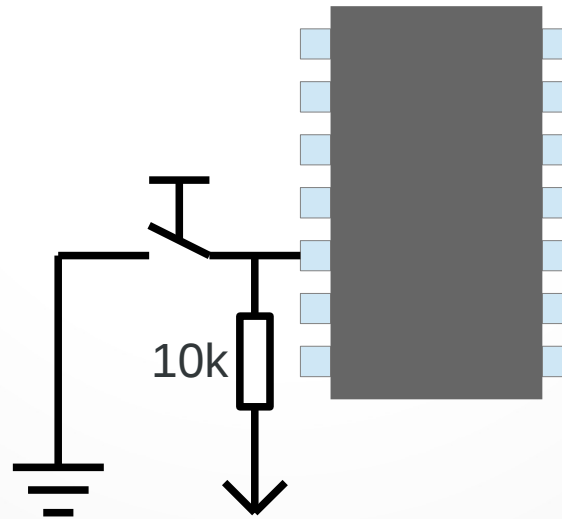
Probleme/Lösungen: Fuses

- Falsche Wahl der Fuses
 - CKSEL0 ... 3 falsch gesetzt => μ C taktet nicht
 - Deaktivierung von SPI => keine Programmierung mehr möglich
 - JTAG aktiv => Entspr. Pins nicht als I/O nutzbar

Fuse	Beschreibung
OCDEN	On-Chip-Debug: Kein kompletter Power-Down
JTAGEN	JTAG Debugging IF (de)aktivieren
SPIEN	Serial Programming IF (de)aktivieren
CKSEL0 ... 3	Clock Source: <ul style="list-style-type: none">• Externer Quarz• Interner oder externer RC Oszillator• Externes Clock Signal

Probleme/Lösungen: I/O

- Input-Pins sollten nicht „offen“ sein
 - Nutzung interner Pull-Up-Widerstand (PUD-Bit in SFIOR Register)
 - Nutzung externer Pull-Up/Down-Widerstand



Probleme/Lösungen: I/O

- Taster an Ports „entprellen“

- SW-Entprellung
- HW-Entprellung (z.B. MC 14490)

```
; Warte auf fallende Flanke an PIND0 (Taste drücken)
loop:  sbic PIND,PIND0          ; Überspringe nächsten
                                           ; Befehl wenn PIND0 low
                                           ; Herausspringen
        rjmp tast2

; Führe Funktion aus, die mit Tastendruck getriggert wird
inc counter          ; Inkrementiere Counter

rcall pause          ; Verzögerung

; Warte nun auf steigende Flanke (Taste loslassen)
tast1:  sbis PIND,PIND0          ; Überspringe nächsten
                                           ; Befehl wenn PIND0 high
                                           ; Wiederhole Schleife
        rjmp tast1

tast2:  rjmp loop                ; Programm-Schleife

; =====

; Verzögerung durch Hochzählen eines 16Bit-Registers
pause:  push r24                 ; Rette Register
        push r25                 ; Rette Register
        clr r24                  ; Low Byte 16Bit Zählreg.
        clr r25                  ; High Byte
pause1:  sbiw r24,1               ; dekrementiere Zählreg.
        brne pause1             ; Gehe pause1 solange !=0
        pop r25                  ; Register zurücksetzen
        pop r24                  ; Register zurücksetzen
        ret                     ; Rücksprung
```

Lesenswert

- Bücher
 - G. Schmitt, Mikrocomputertechnik mit Controllern der Atmel AVR-RISC-Familie, Oldenbourg
- WWW
 - <http://www.mikrocontroller.net/articles/AVR-Tutorial>
 - http://www.avr-asm-tutorial.net/avr_de/beginner/