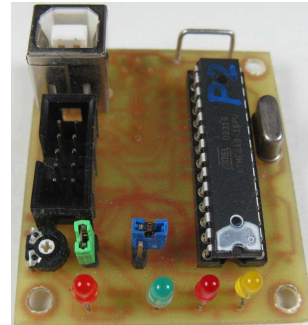


Bastelprojekt USB-Programmer für den ATMEGA8 und andere Prozessoren.

Ein Gemeinschaftsprojekt von OE5GHN Hubert und OE5VLL Erwin.

Grundlegendes:

Wenn man Mikroprozessoren wie den ATMEGA8 von Atmel (z.B. verwendet im Transistortester) selber programmieren will, z.B. um eine neue Programmversion einzuspielen oder Änderungen am bestehenden Programm vorzunehmen, dann benötigt man einen sogenannten Programmer.



Da heutige PC's oft keine seriellen bzw. parallelen Schnittstellen mehr besitzen, ist ein Programmer mit USB-Anschluss sehr sinnvoll.

OM Hubert, OE5GHN, hat auf seiner Homepage www.schorsch.at einen solchen Programmer vorgestellt. Dieser stammt von [www.ullihome.de/index.php/Hauptseite#USB AVR-Lab](http://www.ullihome.de/index.php/Hauptseite#USB_AVR-Lab).

Über diese Homepage kann man auch Bausätze bzw. Fertigeräte beziehen und sich über weitere Projekte informieren.

In diesem Artikel soll der Aufbau eines solchen Programmers, die Beschaffung der nötigen Software, die Installation derselben und die Inbetriebnahme beschrieben werden. Auch für Programmier-Anfänger geeignet.

Die ISP-Schnittstelle:

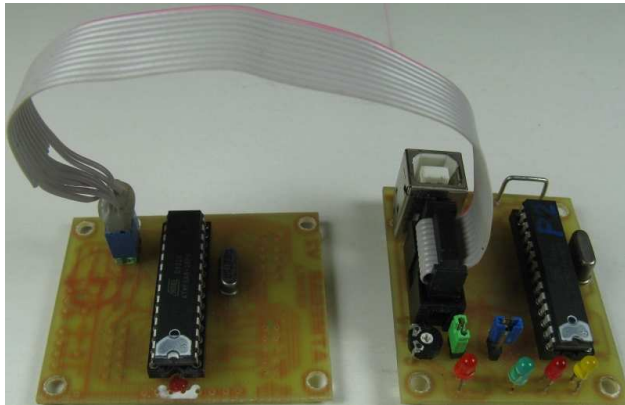
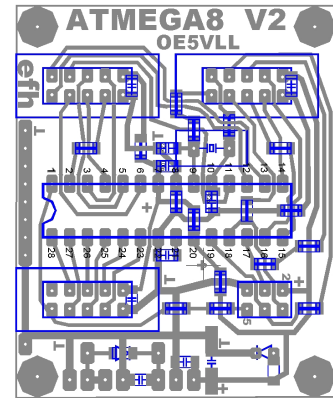
Prozessoren wie der ATMEGA8 besitzen eine ISP-Schnittstelle. Das steht für In-System-Programmierung, was bedeutet, dass solche Prozessoren programmiert werden können, ohne dass sie aus der Schaltung entfernt werden müssen, vorausgesetzt, die Schaltung verfügt über den entsprechenden Anschluss. Ist das nicht der Fall (wie z.B. beim Transistortester), dann benötigt man eine einfache Zusatzplatine (Programmierplatine), in welche man den Prozessor zum Zwecke der Programmierung einstecken kann. Diese Platine wird mittels eines Verbindungskabels mit dem Programmer verbunden.

Für die Programmierschnittstelle sind 2 Arten des Anschlusses üblich. Erstens eine 10-polige-, zweitens eine 6-polige-Pfostenleiste.

Im vorliegenden Fall besitzt der Programmer die 10-polige, die Programmierplatine aber die 6-polige Ausführung. Das bedingt zwar eine etwas umständlichere Anfertigung des Verbindungskabels, hat aber den großen Vorteil, dass das Kabel „einzigartig“ ist und somit nicht so ohne weiteres an einen falschen 10-poligen Anschluss angesteckt wird, was auch zur Zerstörung einer Schaltung führen könnte.

Die zugehörige Programmierplatine hat die Abmessungen 55 x 45 mm. Auf beiden Platinen sind übliche 28-polige IC-Sockel für die Prozessoren in DIL-Ausführung.

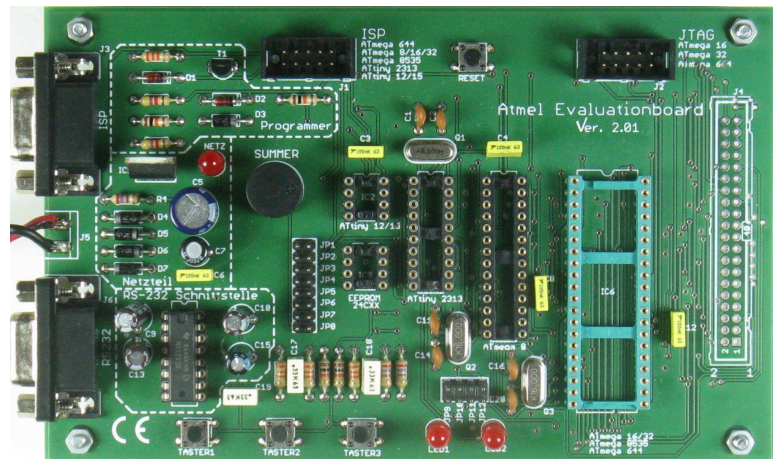
Diese Platine kann auch als Prozessorplatine für ein beliebiges Projekt verwendet werden. 10-polige Stiftleisten für die Ports können bestückt werden, sind aber nicht notwendig, wenn die Platine nur zur Programmierung verwendet wird.



Für Prozessoren mit anderen Gehäusen (z.B. ATMEGA16 im 40-poligen DIL-Gehäuse) ist eine eigene Programmierplatine nötig, wenn nicht das eigentliche Prozessorboard mit der ISP-Schnittstelle ausgestattet ist.

Werden Prozessoren im TQFP- oder anderen SMD-Gehäuseformen eingesetzt, empfiehlt sich die ISP-Schnittstelle auf jeden Fall, da für SMD-Gehäuse Adapter teuer und selten sind.

Es gibt eine preisgünstige Alternative für die Programmierplatine. Es gibt bei der Firma Neuhold-Elektronik in Graz das „Atmel Evaluation-Board Version 2.0.1 - Bausatz“, Bestell-Nr N2755 um 14,95 Euro. Dieses Board verfügt über den 10-poligen ISP-Anschluß und kann somit an den Programmer angeschlossen werden.



Atmel Evaluationboard

Damit können dann die 40-poligen ATMEGA16, ATMEGA32, ATMEGA8535 und ATMEGA644, die 28-poligen ATMEGA8, die 20-poligen ATtiny2313, die 8-poligen ATtiny12 und ATtiny15 u.A. programmiert werden. Dieses Board kann auch gut als Entwicklungsboard eingesetzt werden.

Der von OM Hubert, OE5GHN vertriebene Transistortesterbausatz (beschrieben in QSP Heft 3 / 2010) besitzt auch eine ISP-Schnittstelle, über welche die ATMEGA8 programmiert werden können.

Die Software:

Die Software teilt sich in 5 Bereiche:

1. Jene Software, welche in dem ATMEGA8 steckt, mit dem programmiert wird. Diese übernimmt auch die Kommunikation über die USB-Schnittstelle.
2. Den Treiber für den PC (für die USB-Hardware)
3. Der C-Compiler (oder auch Compiler für andere Programmiersprachen, z.B. Basic)
4. Das AVR-Studio 4, welches die „Steuerung“ für den Programmierer übernimmt.
5. Schlussendlich die Anwendersoftware, mit welcher der ATMEGA8 (oder auch ein anderer Prozessor) programmiert werden soll.

Zu Teil 1: Software des Programmers

Wer nicht schon eine andere Möglichkeit hat, den ATMEGA8 zu programmieren, tut sich schwer, die kostenlos angebotene Software in den Chip zu bekommen – und wer das kann, braucht eigentlich keinen Programmierer mehr. Hat man einen guten Freund, welcher das für einen erledigen kann, ist man auch fein raus. Für alle, welche keine Gelegenheit haben, biete ich mit der Programmierer-Software versehene ATMEGA8 preisgünstig an. Email an erwin.hackl@pc-club.at genügt für die Bestellung.

Wer sich den Programmierchip selber programmieren will, kann die nähere Vorgehensweise dazu auf der auch anderweitig sehr interessanten Homepage <http://www.wiki.ullihome.de/> nachlesen

Zu Teil 2: USB AVR Lab Tool

Die folgende Beschreibung der Installation bezieht sich auf Windows-XP. Um den Treiber installieren zu können, wird von


<http://wiki.ullihome.de/index.php/USB AVR-ISP-Download/de#Windows>

das USB AVR Lab Tool 5.09 Stand 19. 5. 2010 heruntergeladen. Es handelt sich dabei um die Datei

10403_0de_0avrlabtool_i386-win32-5_9.exe.

Dieses Programm wird anschließend gestartet und damit das USB AVR Lab Tool Version 5.09 Build 0 installiert. Der Treiber wird dabei mit installiert.

Anschließend wird der Programmierer an eine USB-Schnittstelle angeschlossen. Zuerst kommt dann die Meldung „Gerät wurde nicht erkannt „. Man lässt nun diesen automatischen suchen - der Treiber befindet sich bereits auf dem PC, braucht also nicht aus dem Internet geholt zu werden. Nach erfolgreicher Installation kann man im Geräte-Manager unter Anschlüsse (COM und LPT) einen Communications Port (z.B. COM 4) finden – es kann natürlich je nach PC eine andere Ziffer sein. Damit ist die Installation des Treibers abgeschlossen.

Adresse  <http://wiki.ullihome.de/index.php/USB-AVR-ISP-Download/de#Windows>

3.1.1 PDF
 3.2 USB-AVR ISP

Werkzeuge

- Links auf diese Seite
- Änderungen an verlinkten Seiten
- Hochladen
- Spezialseiten
- Druckversion
- Permanentlink

Produktinfos

Produktinfo AVR-USB Lab
 Treiberinstallation AVR-USB Lab
 Produktinfo AVR-USB Lab Leiterplatte

Entnehmen Sie die Produktinfos und Downloads für die anderen Produkte bitte den Artikeln im Shop.

Software

Mit der Software *USB-Lab-Tool* kann man die Firmware des USB-Labs austauschen sowie die Option: (Daher bitte vor Installation des USB Labs herunterladen und installieren).

Windows

USB AVR Lab Tool 5.09 Stand 19.05.2010
 USB AVR Lab Tool+USB AVR Lab Programmer 4.08 Stand 23.09.2009

USB AVR Lab Oszilloskop 0.6 Stand 28.12.2009

USB AVR Lab Boundary Scan Tool 1.0 Stand 05.04.2009

...

Zu Teil 3: AVR GCC C-Compiler

Wer auch vor hat, selber in C zu programmieren, sollte unbedingt vor Installation des AVRStudio4 den AVR GCC C-Compiler installieren. Der Grund dafür ist, dass dieser dann automatisch mit eingebunden wird.

Wie später unter dem Punkt „Die Programmierung von Prozessoren“ beschrieben, hier der Aufruf des C-Compilers. Nachdem beim Einstiegsfenster „New Project“ angeklickt wird, kommt nebenstehendes Fenster. Hier gibt man für ein neues Projekt den Projektnamen ein und startet dann den C-Compiler durch anklicken von „AVR GCC“.



Zu Teil 4: AVR-Studio 4

Adresse <http://www.mikrocontroller.net/articles/AVR-Studio> Wechseln zu Li

www.mikrocontroller.net

Home
AVR

- AVR-Tutorial
- AVR-GCC-Tutorial

ARM

- LPC2000
- AT91SAM7

MSP430

FPGA, CPLD & Co.

- Grundlagen zu FPGAs
 - VHDL & Co.
 - Xilinx ISE

DSP

Elektronik allgemein

- SMD Lötten
- Operationsverstärker
 - Oszilloskop

Foren

- µC & Elektronik
- Analogtechnik
- FPGA, VHDL & Co.
 - DSP
 - GCC
- Codesammlung
 - Markt
 - Platinen
- HF, Funk & Felder
 - Hausbus
- PC-Programmierung
- PC Hard- & Software
- Ausbildung & Beruf
 - Offtopic

AVR-Studio

Das **AVR-Studio** ist eine kostenlose Entwicklungsumgebung (IDE) für die Assembler-Programmierung der AVR-Mikrocontroller von Atmel. Sie besteht aus einer Projektverwaltung, einem Editor und einem Debugger, der mit dem integrierten Simulator, einem JTAG-Adapter oder einem In Circuit Emulator genutzt werden kann.

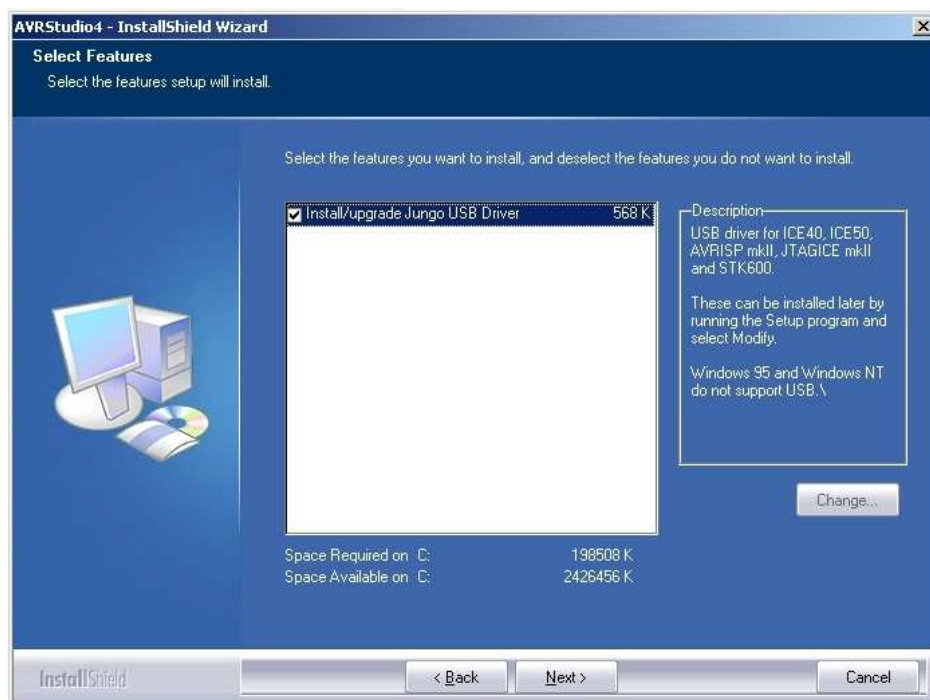
Mit dem AVR Studio kann auch in der Programmiersprache C programmiert werden. Dazu muss vor der Installation des AVR Studio der GNU C Compiler für AVRs WinAVR installiert werden.

- <http://www.atmel.com/avrstudio>

Direktlink aktuelle Version [Bearbeiten]

- AVRStudio4.18SP3.exe AVR Studio 4.18 Service Pack 3 Release Candidate (build 700, 32 MB, updated 07/10)
- AVRStudio4.18SP2.exe AVR Studio 4.18 SP2 (build 700, 28 MB, updated 02/10)
MD5:51350b5b0949fb3673ad6cb2621ec5d8
- AVRStudio4.18SP1.exe AVR Studio 4.18 SP1 (build 692, 28 MB, updated 12/09)
MD5:9c1c4bcbe7614f7c44d215e3c277d10
erzwingt zwar nicht Version 4.18 installiert zu haben, ist aber empfehlenswert
- AvrStudio4Setup.exe AVR Studio 4.18 (build 684) (116 MB, updated 11/09) MD5:f2b0fa6dbdf6e858806e58fd6ae6b82
- AvrStudio417Setup.exe AVR Studio 4.17 (build 666) (112 MB, updated 7/09)
MD5:9705a9362da76aa9779322127640a184
- AvrStudio416Setup.exe AVR Studio 4.16 (build 628) (126 MB, updated 2/09) (last version for Win98)
MD5:d1c412d7a05a9ad95486d7ea680f68e5
- aStudio4b623.exe AVR Studio 4.15 (build 623) (94 MB, updated 11/08)
- aStudio4b589.exe AVR Studio 4.14 (build 589) (89 MB, updated 4/08)
- aStudio4b528.exe AVR Studio 4.13 (build 528) (73 MB, updated 03/07)

Direkt von der ATMEL-Homepage oder von www.mikrocontroller.net/articles/AVR-Studio wird das AVRStudio4 (derzeit Version 4.18) und die drei Servicepacks heruntergeladen und auf dem Rechner installiert. Die Installation ist soweit eigentlich problemlos, wichtig ist aber, dass das Häkchen bei „install/upgrade Jungo USB Driver“ gesetzt ist, damit dieser auch mit installiert wird.



Wird der USB-Programmer dann erstmalig angeschlossen, kommt die Meldung „Assistent für das Suchen neuer Hardware. Hier wird die automatische Suche auf der Festplatte empfohlen, es braucht nicht ins Internet gegangen zu werden, da sich der Treiber bereits auf der Festplatte befindet.

Anschließend kann man in der Systemsteuerung unter System – Hardware – Geräte-Manager – Anschlüsse (COM und LPT) einen vorher nicht da gewesenen Eintrag „Communications Port (Com 5)“ finden. Die Nummer des Ports kann natürlich je nach PC eine andere sein. Diese merke man sich aber, da man sie später noch benötigt.



Zu Teil 5: Anwendersoftware

Hier gibt es mehrere Möglichkeiten:

- a.) Man kann eine fertige Software in den Prozessor einspielen – ein Beispiel hierfür ist der Transistortester.
- b.) Man ändert vorhandene Software
- c.) Man erstellt neue Software

Für die Fälle b und c sind Programmierkenntnisse erforderlich. Dieses Thema wird in diesem Beitrag nicht behandelt. Das vorgehen im Fall a wird im kommenden Kapitel beschrieben.

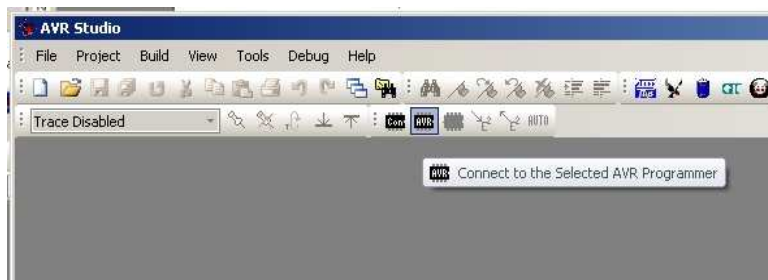
Die Programmierung von Prozessoren:

Hier wird die Programmierung eines ATMEGA8-Prozessors mit einer fertigen Software am Beispiel des Transistortesters beschrieben.

Am USB-Programmer wird die Zusatzplatine (Programmierplatine) mit dem zu programmierenden ATMEGA8 über das Programmierkabel angeschlossen. Der Programmer wird nun seinerseits an einem USB-Port des Rechners angeschlossen.

AVRStudio4 wird gestartet. Das Fenster mit „Welcome to AVRStudio4“ kann mit „Cancel“ geschlossen werden.

Wird hier auf „New Project“ geklickt und dann auf „AVR GCC“ kommt man zum C-Compiler (wenn er installiert ist).

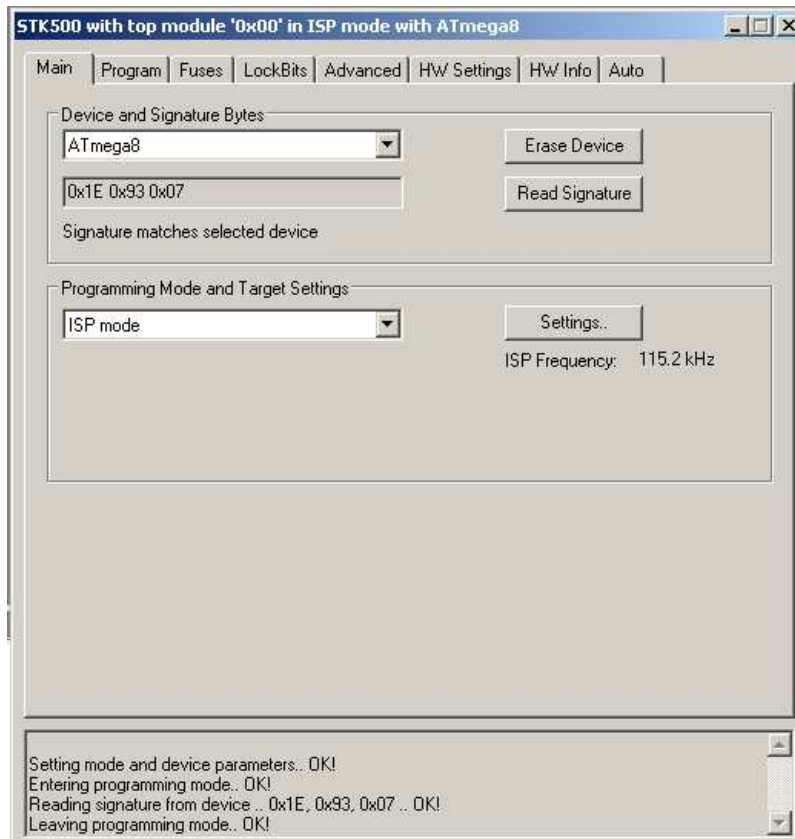


Danach wird „Connect to the selected AVR Programmer“ angeklickt (Der mittlere von drei symbolischen IC's).

Beim ersten mal sollte dann das Fenster für Auswahl des Programmers und des COM-Ports aufgehen. Als Programmer wird der STK500 (ohne + !) und der COM-Port, welcher in der Systemsteuerung aufschien, angegeben.

Wurde alles richtig angegeben, sollte anschließend das Fenster „STK500 with top module „0x00“ in ISP mode with ATmega8“ aufgehen. Dazu siehe nächstes Bild.

Hier wird unter „**Main**“ der ATmega8 als Prozessor aufgeführt.



Unter „Settings“ wird ISP Frequency 115,2 kHz ausgewählt.

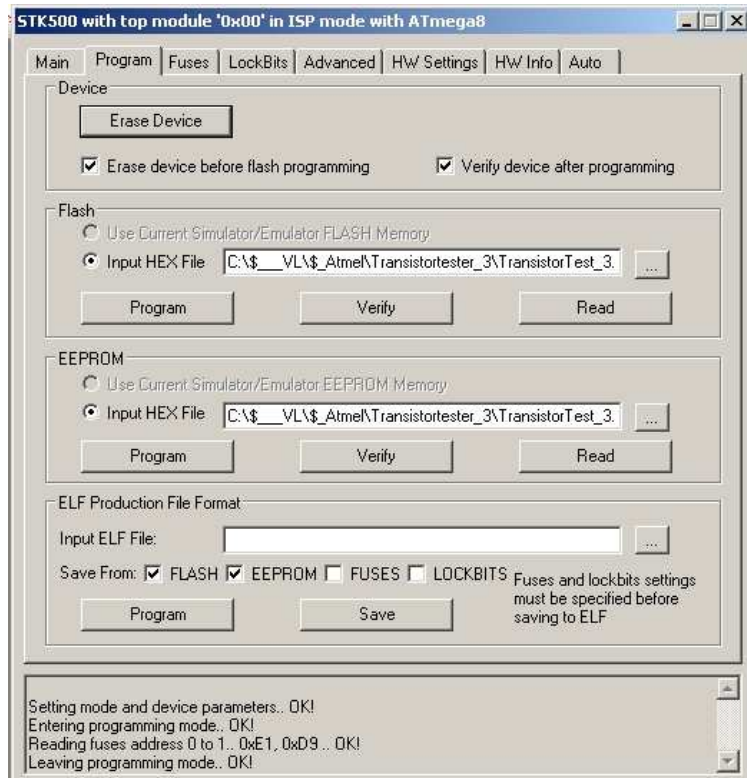
Wird „Read Signature“ angeklickt, sollte unterhalb der Prozessorsignatur der Text „Signature matches selected device“ erscheinen, was bedeutet, dass die Signatur des Prozessors dem ausgewählten Typ entspricht und somit alles bestens ist.

Unter „**Program**“ können nun die in den Prozessor zu schreibenden HEX-Files ausgewählt werden.

Die beiden Häkchen bei „Erase device before flash programming“ und bei „Verify device after programming“ sollten gesetzt sein.

In den meisten Fällen braucht nur das Flash programmiert zu werden. Somit muss nur das HEX-File unter „Flash“ ausgewählt zu werden. Ein Klick auf „Program“ und der Prozessor wird programmiert.

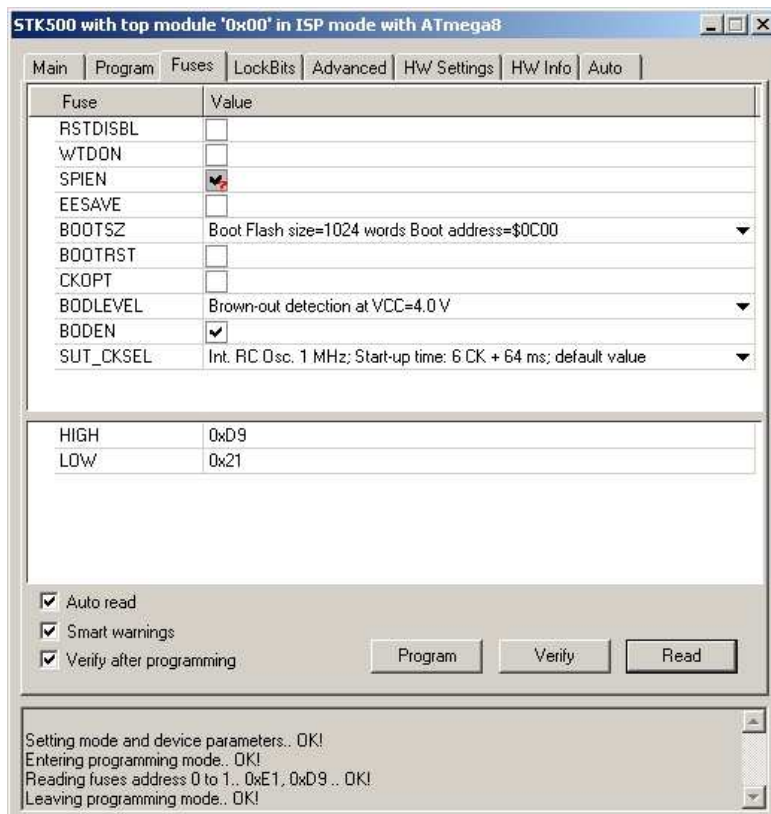
Beim Transistortester wird auch das EEPROM programmiert. Somit wird unter „EEPROM“ das zugehörige EEP-File ausgewählt, und dann über den entsprechenden „Program“-Knopf der Programmiervorgang für das EEPROM des Prozessors gestartet.



ACHTUNG:

Es ist wichtig, als erstes das Flash und erst als zweites das EEPROM zu programmieren!

Besitzt das Gerät selber eine Programmierschnittstelle (wie z.B. jener Transistortester, welcher von OE5GHN angeboten wird), so benötigt man die Zusatzplatine in dem Fall nicht, es kann gleich die Transistortesterplatine als Programmierplatine verwendet werden.



Unter „Fuses“ sollte in unserem Beispiel bei „BODLEVEL“ Vcc=4.0V eingestellt sein und unter „BODEN“ das Häkchen gesetzt sein. Ist alles richtig, ist in den unteren zwei Zeilen unter „HIGH“ 0xD9 und unter „LOW“ 0x21 zu lesen. Es können aber auch gleich hier diese Werte eingegeben werden. Beim Transistortester kann die Programmierung der Fuses auch entfallen.

ACHTUNG:

Die Einstellung der Fuses ist je nach Schaltung anders. Sie dient unter anderem der Auswahl, ob der Prozessor mit internem Taktgenerator, mit Quarz oder mit Takt von einem externen Quarzoszillator arbeiten soll.

Soll ein ATMEGA8-Chip, welcher schon z.B. für Quarz programmiert wurde, in einer Programmierplatine ohne Quarz neu programmiert werden, so funktioniert das nicht, da ihm der Takt fehlt. In diesem Fall muss ein Quarz in der Programmierplatine eingelötet sein. Sehr wohl funktioniert dies aber, wenn bei einem neuen Prozessor zuerst das Flash und dann erst die Fuses programmiert werden, da dieser dann den externen Quarz erst in der Schaltung benötigt, in der er dann später betrieben wird.

Nun wäre der Prozessor fertig, die Zusatzplatine kann abgesteckt werden, der IC entnommen und in unserem Fall in den Transistortester eingesetzt werden.

Die LED's auf der Programmer-Platine:

Die **orange LED** zeigt an dass der Programmer an die Stromversorgung angeschlossen ist.

Die **rote LED** beginnt ebenfalls mit Anschluss an die Stromversorgung zu leuchten, erlischt aber nach wenigen Sekunden. Ab diesem Moment leuchtet die **grüne LED**.

Außerdem blinkt die **rote LED** während des Programmiervorgangs.

Die **gelbe LED** beginnt zu leuchten, wenn der Programmer vom AVRStudio4 erkannt wurde.

Es ist daran gedacht, auch bei diesem Projekt einen Bausatz anzubieten, welcher beide Platinen und alle benötigten Bauteile beinhaltet.

Selbstverständlich können auch einzelne programmierte Prozessoren bezogen werden.

Bei Interesse bitte email an erwin.hackl@pc-club.at .

Viel Spaß am Eigenbau dieses Programmers wünschen Euch

OE5GHN Hubert und OE5VLL Erwin.