

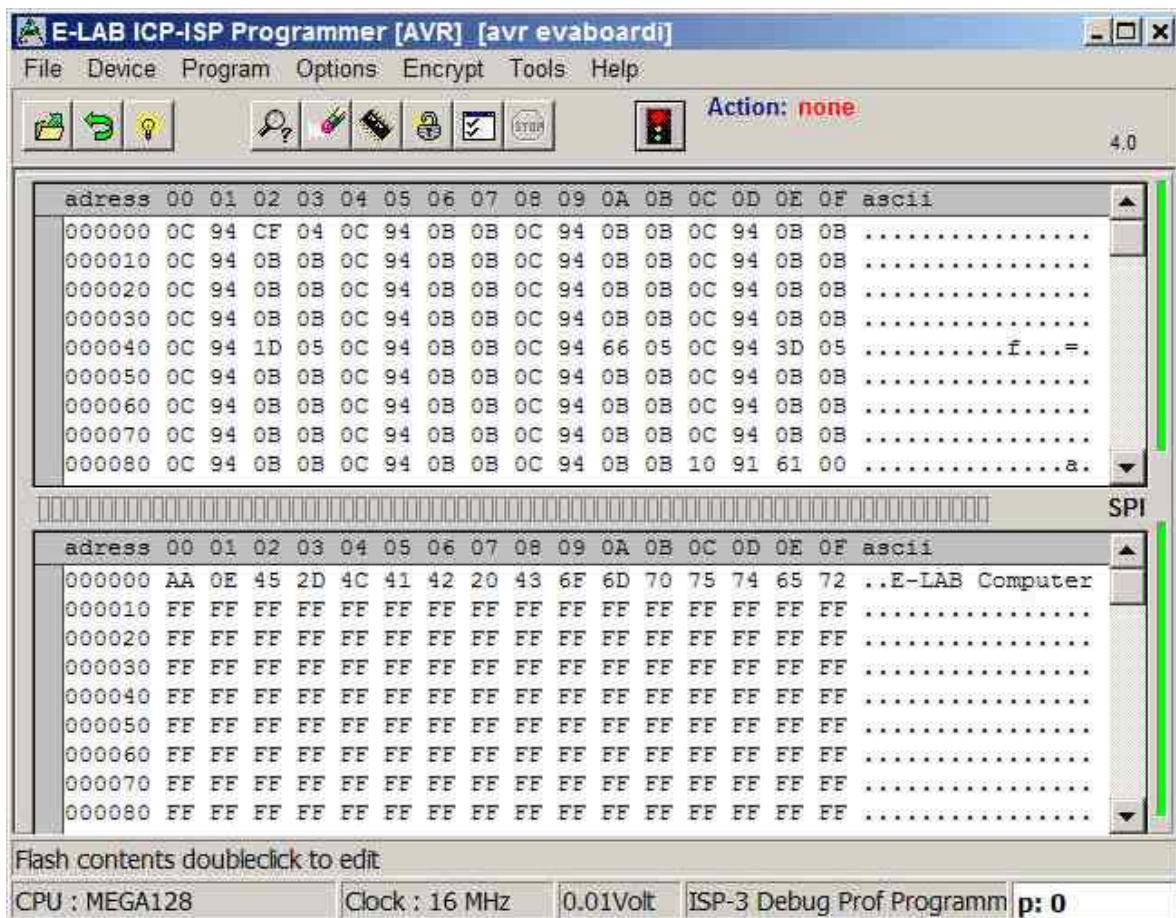


Benutzer Handbuch ISP3-X Programmer

# ISP3-X mit AES encryption

## In-Circuit Programmer

© Copyright 1998-2018 by E-LAB Computers



WEB: [www.e-lab.de](http://www.e-lab.de) Tel: 07268/9124-0 Fax: 07268/9124-24



## Inhaltsverzeichnis

<b>ÜBERSICHT</b> .....	<b>3</b>
<b>EIGENSCHAFTEN</b> .....	<b>3</b>
<b>ANSCHLÜSSE</b> .....	<b>4</b>
<b>SOFTWARE</b> .....	<b>5</b>
<b>AVRProg</b> .....	<b>5</b>
USB Stromversorgung.....	7
Aufruf Optionen .....	8
Kommando Zeilen Parameter .....	8
Knopf-Leiste .....	10
Funktionen der Buttons und Menues.....	11
AES PAC Files .....	24
Projekt Files erstellen .....	25
Programmieren mit AVRprog .....	26
Status, Fehleranzeige und Probleme .....	26
<b>PackProg</b> .....	<b>27</b>
Buttons und Menus.....	27
Programmieren im Transparent Modus .....	29
Kommando Zeilen Parameter .....	30
Telnet Interface.....	31
<b>EXTERNE HARDWARE</b> .....	<b>33</b>
<b>Diverse Adapter</b> .....	<b>33</b>
<b>USB TREIBER</b> .....	<b>34</b>
<b>Unsignierte Treiber installieren</b> .....	<b>35</b>
<b>STROMVERSORGUNG DES ZIELSYSTEMS</b> .....	<b>38</b>
<b>MULTIPLE PROGRAMMER</b> .....	<b>38</b>
<b>FIRMWARE UPDATE</b> .....	<b>39</b>
<b>ANHANG</b> .....	<b>41</b>
<b>XMega</b> .....	<b>41</b>
<b>TINY4-5-9-10-20</b> .....	<b>42</b>
<b>UPDI Tiny und UPDI Mega</b> .....	<b>43</b>
<b>Chipcon</b> .....	<b>44</b>
Chipcon Evaluation Boards .....	45
<b>AT89LPxx</b> .....	<b>47</b>
<b>Serial Flash (SPI-Flash)</b> .....	<b>48</b>
<b>WEITERE E-LAB PROGRAMMER</b> .....	<b>49</b>

---

## ÜBERSICHT

In-Circuit Programmierung (ISP oder JTAG) ist die Technik der Zukunft für Klein- und Mittel-Serien elektronischer Baugruppen mit Prozessoren. Vor allen Dingen bei SMD-Bauteilen stellt sich das Problem der Programmierung, da hier eine Vielfalt von teuren und speziellen Adaptern notwendig ist. Ein weiterer Vorteil von ISP/JTAG im Zusammenhang mit dem AVR ist die fast beliebige Reprogrammierbarkeit der CPU's .

Der vorliegende ISP-Programmierer zeichnet sich durch kompakte Grösse, komfortable Software und einfache Handhabung aus. Dieses Handbuch gilt nur für die **ISP3-X** Programmierer Version.

---

## EIGENSCHAFTEN

- Anschluss an den PC über die USB Schnittstelle
- Arbeitet im Full-Speed USB-2 Modus
- Kann auch an USB-1 Ports und HUBs angeschlossen werden
- Kein Netzteil erforderlich. Gerät wird von der Schnittstelle des PC's (USB) versorgt.
- Automatische Anpassung an die Spannung der Ziel-CPU (1.8-5.5Volt)
- Komfortable und umfangreiche Software
- Software läuft unter XP, Vista und Windows7,8 / 32 und 64bit.
- Kleines, leichtes und handliches Gerät ca. 100x55x20mm
- Unterstützt alle SPI, JTAG, PDI , TPI und UPDI programmierbaren AVR's
- Unterstützt alle SPI programmierbare AT89Sxx Typen
- Unterstützt die ChipCon CC1110, CC2510 und CC2430 Familie
- Unterstützt die Serielle Flash (SPI-Flash) Familien AT25DFxxx, S25FLxxx und SST25VFxxx
- Unterstützt die Atmel AT89LPxx Familie
- Extrem schnell, programmiert einen Mega128 in **3sec** (JTAG) und einen Tiny44 in **1sec** (16MHz)
- Programmierbare Ausgangsspannung (Speisung) für das Zielsystem. 1.8..5.5Volt 30mA..300mA
- Selbst Update mit neuer Firmware vom PC aus.
- **SPI, JTAG, TPI , PDI** und **UPDI** programming Support für die AVR's
- Mit dem enthaltenen Tool **PackProg** auch für die Klein-Serien Fertigung geeignet.
- Hervorragend auch für Produktions Programmierung geeignet durch extrem sichere AES Verschlüsselung
- Verarbeitet auch **AES** verschlüsselte Projekt Files (PAC) onBoard
- Stückzahlen Begrenzung bei AES-PAC Files voreinstellbar
- Diverse Adapter als Option erhältlich

## ANSCHLÜSSE

Windows-XP, Vista oder Windows7/8 ist Voraussetzung. Ein USB1 oder USB2 Port ist erforderlich. Mit einem USB-2 Port arbeitet das Gerät mit Full Speed. Mit einem USB-1 Port erhöht sich die Programmierzeit etwas.

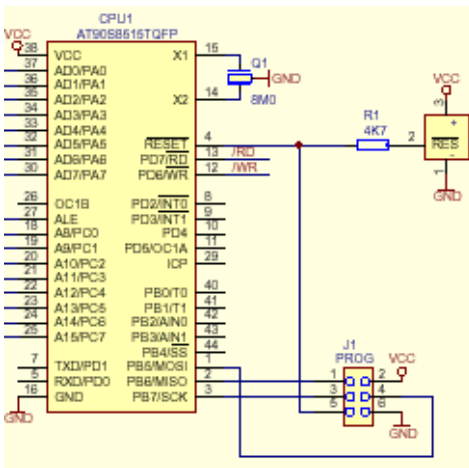
Das mitgelieferte USB Kabel muss am PC angeschlossen werden. Die zugehörigen USB Treiber müssen einmal installiert werden. Bitte lesen Sie dazu das Kapitel „USB-Treiber“ im Anhang.

Die interne Spannung beträgt ca. 3,3Volt. Es dürfen keine niederohmigen Lasten an den Programmierpins der CPU angeschlossen sein. Kondensatoren > 100pF sind ebenfalls nicht erlaubt, am Reset-Pin max. 100nF!! Beim XMEga darf kein Kondensator am Reset angeschlossen sein. Pullup minimal 100kOhm.

Die Belegung des 6-poligen Steckverbinders (0.1inch male dual inline) entspricht einer Empfehlung von Atmel. Auf die entsprechende Pfostenleiste (Target) gesehen, ergibt sich folgende Beschaltung:

CPU MISO(TxD)	<b>01 20</b>	VCC	PDI_DAT	<b>01 20</b>	VCC (3.3V)
CPU SCLK	<b>03 40</b>	CPU MOSI(RxD)		<b>03 40</b>	
/RES	<b>05 60</b>	GND	RESET/PDI_CLK	<b>05 60</b>	GND

**TopView AVR SPI Stecker** **TopView XMEGA PDI Stecker**



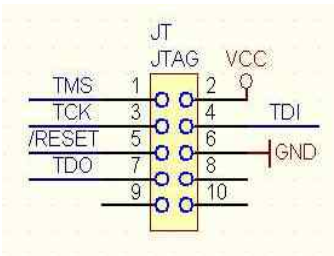
Der Pin1 der 6poligen Buchse des Programmers ist durch ein kleines Dreieck auf der Vorderseite markiert. Ein **verpoltes** Einstecken der Buchse des Programmers auf die Pfostenleiste des Targets kann zur **Zerstörung** des Programmierers führen!

Die zulässige Betriebsspannung des Targets während des Programmierens muss zwischen 1.8 und 5.5 Volt liegen.

Andauernde **Kurzschlüsse** auf einem oder mehreren der 4 Steuerpins führt zur Zerstörung des Gerätes bzw. seiner internen AVR CPU. Deshalb dürfen nur elektrisch einwandfreie Boards an den Programmer angeschlossen werden.

### JTAG Programmierung

Zur JTAG Programmierung der Target CPU muss das 6-polige Kabel gegen ein 10-poliges ausgetauscht werden. Da der den gleichen Stecker sowohl für ISP als auch für JTAG Programmierung verwendet, weicht der JTAG Stecker auf dem Zielsystem von der Original Atmel JTAG Steckerbelegung ab.



Nebenstehend ist die Steckerbelegung, wie sie das Zielsystem verwenden muss, wenn über die JTAG Schnittstelle der Ziel-CPU programmiert werden soll. Bitte beachten, dass im Gegensatz zum Atmel Stecker hier auch das **/RESET** Signal an die Target CPU angeschlossen werden muss.

TPI_DAT	<b>01 20</b>	VCC (5V)
TPI_CLK	<b>03 40</b>	
/RES	<b>05 60</b>	GND

### Tiny 4, 5, 9, 10, 20 Programmierung

Diese Tinsys werden mit 5V über drei Pins programmiert. Der Programmier Mode nennt sich TPI. Da der Programmier Stecker auch für den SPI Mode verwendet wird, weicht dieser von der Atmel Belegung ab. Die Last am TPI\_DAT Pin darf nicht kleiner als 80kOhm sein.

## SOFTWARE

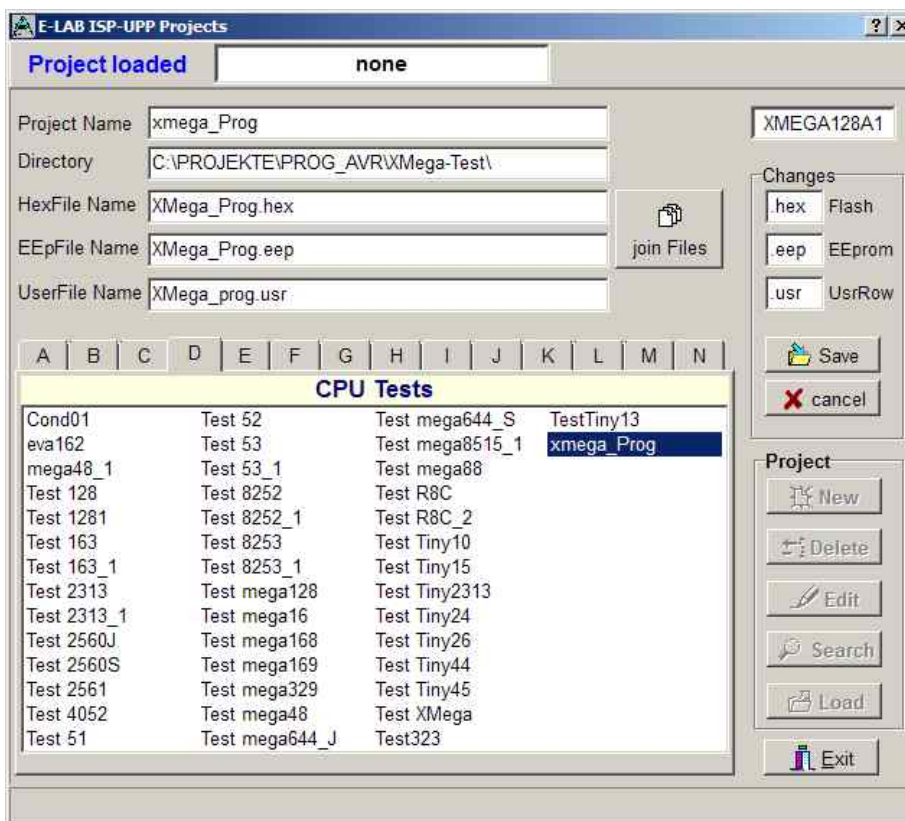
Zur Steuerung des ISP-X Programmers werden zwei Programme geliefert. Das erste nennt sich **AVRProg.exe** und ist das umfangreichere. Es wird benutzt um Projekte zu erzeugen, die Daten anzusehen und zu editieren, sowie die Fuse- und Lock Bits auszulesen / zu editieren. Weiterhin können damit die Programmer Setup Optionen gesetzt werden (z.B. Stromversorgung) und auch das Zielsystem kann damit programmiert und überprüft werden. Zusätzlich erlaubt es gepackte und verschlüsselte Programmierdateien zu erstellen.

Das zweite Programm nennt sich **PackProg.exe** und kann nur mit den zuvor durch die mit AVRprog erzeugten gepackten oder verschlüsselten Dateien benutzt werden. Dieses Programm ist weiter unten beschrieben und eher für die Benutzung in der Produktion gedacht.

## AVRProg

AVRProg kann direkt aus der AVRco IDE (PED32) gestartet werden. Dann werden alle Einstellungen und Dateien direkt aus der IDE übernommen. Alternativ kann der Aufruf über den Eintrag im Startmenu erfolgen. In letzterer Betriebsart muss ein existierendes Projekt geladen oder ein neues angelegt werden.

**Existierendes Projekt laden:** ein existierendes Projekt wird durch Doppelklick auf den entsprechenden Tabelleneintrag geöffnet und geladen. Alternativ einfacher Klick und Load Schaltfläche. Dadurch werden alle zugehörigen Einstellungen übernommen.



Anzeige des im Auswahlfenster geklickten Projekts mit CPUtyp und zugehörigem Projekt Pfad.

Das Flash Hexfile mit File Extension.  
EEProm File mit Extension.  
XMega UserRow + Extension

Abspeichern nach Änderungen im Projekt oder neuem Projekt (A..N dienen nur eigenen Gruppierungen der Projekte)

Kommandos:  
Neues Projekt erstellen

Projekt löschen.

Vorhandenes Projekt ändern

Projekt im Netzwerk suchen

Projekt laden.

Verlassen dieses Dialogs.

Zu jedem Projekt kann auch ein Kommentar abgelegt werden. Siehe dazu „Kommentar“-Dialog.

**Anlegen eines neuen Projekts:** Auswahl der gewünschten Speicher Seite im Tabbed Notebook. Button **New** klicken. Im Feld **Project Name** den gewünschten Namen eintragen. Das Feld **Directory** klicken. In dem erscheinenden Dialog die gewünschte directory auswählen Jetzt erscheint der Dialog für die File-Extensions und File-Typen. Gewünschte Typen auswählen. Mit dem nun erscheinenden File-Dialog den Hex-File Namen auswählen. Zuletzt muss noch die CPU aus dem CPU-Select Dialog ausgewählt werden. Die bis zu diesem Punkt erfolgten Angaben sind muss Angaben. Der folgende Dialog für ein evtl. vorhandenes EEPROM File kann ignoriert werden. Das neue Projekt wird mit dem **Save** Button abgespeichert.



## ISP3-X portabler In-Circuit Programmer

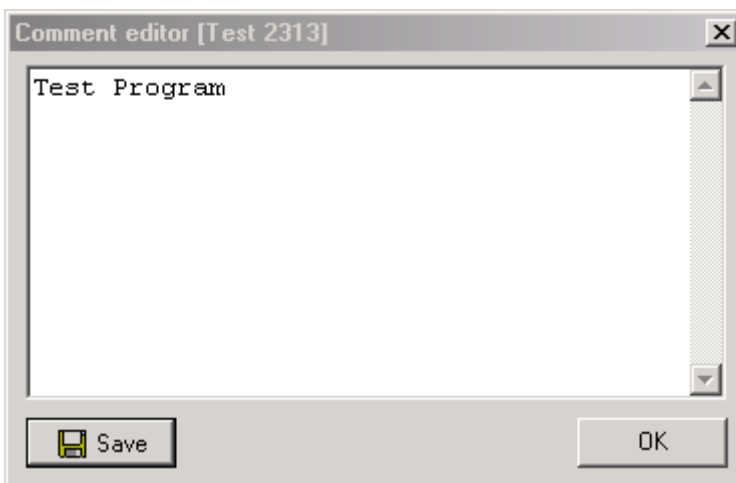


**Dialog für die File Typen** eines Projekts. Diese Parameter sind Projekt bezogen und müssen für jedes neue Projekt separat eingestellt werden.

**Ändern eines vorhandenen Projekts:** Button **Edit** klicken. Durch einen Klick auf die jeweiligen Felder wird der zugehörige Dialog aufgerufen. Nach den Änderungen das Projekt mit dem **Save** Button abspeichern.

Soll z.B. nur ein Projekt von einer Seite Notebooks in eine andere Seite verschoben werden, dann muss mit dem Edit Button der Edit Modus gestartet werden. Dann wird die gewünschte Notebook Seite eingestellt und anschliessend mit dem Save Button die Änderung gespeichert.

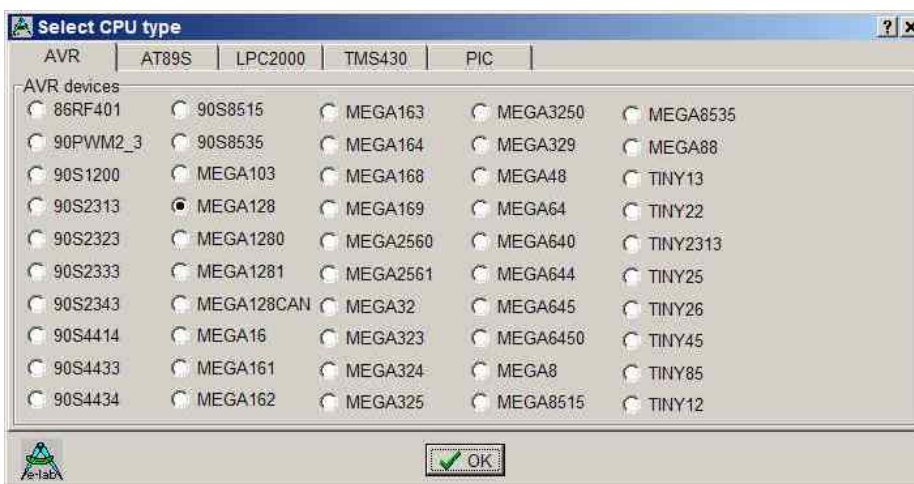
### Kommentar Editor



Ein Klick mit der rechten Maustaste auf einen Eintrag im Projekt Auswahl Dialog öffnet den zugehörigen Editor entweder im read-only mode oder zum Editieren.

Hier können Kommentare zu den jeweiligen Projekten angezeigt oder abgelegt werden.

### Dialog zur CPU Auswahl



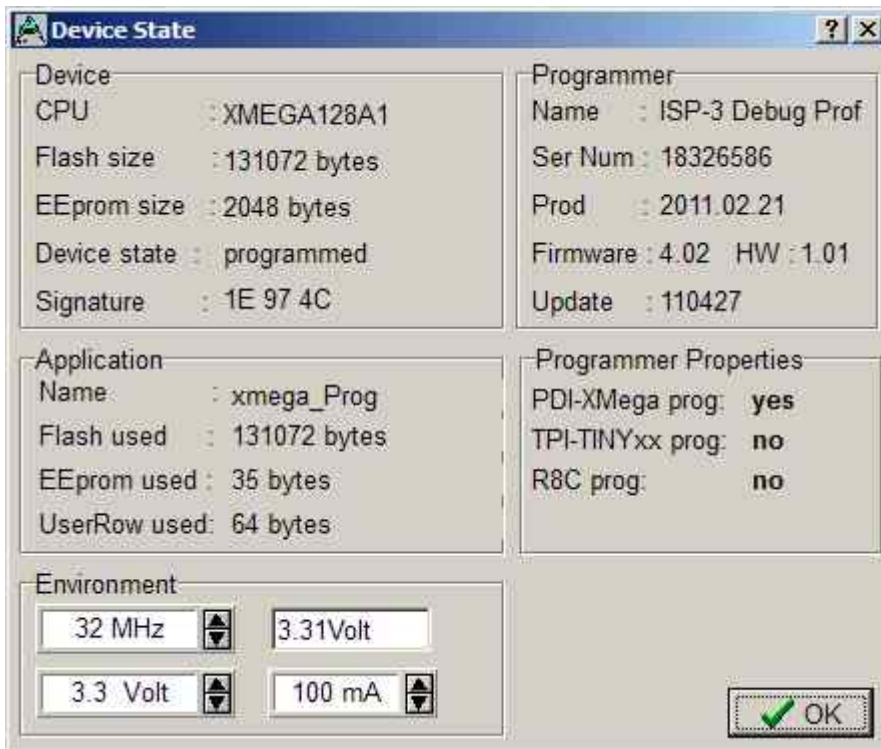
Wenn ein Projekt neu erstellt oder editiert werden muss, dann muss auch der CPU Typ vorgegeben werden. Dazu dient dieser Auswahl Dialog.

Die CPU Typen LPC, TMS und PIC sind zur Zeit noch **nicht** implementiert.

## Device state dialog



Nach Auswahl der CPU muss die Taktfrequenz angegeben werden. Der entsprechende Dialog wird mit diesem Button geöffnet. Hier werden die Parameter des vom PC geladenen Projekts und die Parameter des geladenen hex-Files angezeigt.



Editierbare Parameter befinden sich in der **Environment** Gruppe

Die Taktfrequenz bestimmt die SPI Prog. Geschwindigkeit. Ist diese zu hoch, kann die Programmierung fehlschlagen.

Für **JTAG, PDI, TPI** Programmierung ist der Wert egal, sollte aber dennoch korrekt gesetzt werden.

Die Spannung rechts daneben ist der Wert den der ISP-Programmer auf dem Target Board misst, falls dieses angeschlossen ist.

Die unteren beiden Felder für Spannung und Strom sind nur bei den Programmer Typen sichtbar, die die Möglichkeit besitzen, das Zielsystem mit Strom zu versorgen. Der ISP3 gehört dazu.

Das Strom Feld dient zum Einstellen des maximal zulässigen Stroms. 0.0mA sperrt die ISP-interne Versorgung des Ziel Systems. Die anderen Werte geben die interne Versorgung frei. Wenn ein Strom > 0mA eingestellt ist, kann die gewünschte zugehörige Spannung eingestellt werden. Wenn freigegeben, versorgt der ISP Programmer das Ziel System mit der eingestellten Spannung (nur im programming status). Der Strom für das Zielsystem wird auf den Vorgabe Wert begrenzt.

Bei den XMegas wird die Spannung auf max. 3.6V begrenzt  
Bei einigen TINYS ist die minimale Versorgungsspannung auf 5V begrenzt.

Im Feld **Application** wird das aktuelle, im PC geladene Projekt angezeigt.

Im Feld **Programmer** werden die Daten des angeschlossenen Programmers angezeigt, Programmer Typ, Firmware Revision und letzter Firmware Update. Hierbei ist der letzte Punkt wichtig, denn alle Update Files beginnen mit dem Erstellungs-Datum des Updates **yy-mm-dd**. Im obigen Dialog ist das 2011, Apr, 27.

Damit lässt sich leicht feststellen, ob ein vom Internet heruntergeladenes Update neuer ist, als das im ISP3 installierte.

Mehr Informationen über Firmware Updates finden Sie im Kapitel **Firmware Update** am Ende dieses Manuals

## USB Stromversorgung

Der ISP3-X besitzt einen internen Step-Up Regler, der aus der USB Spannung (4..4,8V) etwa 5,6V erzeugt um das Zielsystem versorgen zu können. Deshalb ist keine externe Versorgung des Zielsystems notwendig, solange die Last durch den USB Port bzw. Hub bereitgestellt werden kann. Weiter unten wird die Stromversorgung noch genauer beschrieben.



## Aufruf Optionen

### Start durch den Windows Explorer

Wenn Sie im Windows Explorer die File-Extension **\*.ispe** mit **AVRprog.exe** verknüpfen, ruft ein Doppel-Klick auf ein Projekt-INI File xxx.ispe im Explorer automatisch das Programmer Programm auf, welches nun wiederum automatisch das komplette angeklickte Projekt lädt.

### Start innerhalb der E-LAB IDE PED32

Der Programm Aufruf ist fertig installiert in die IDE.

## Kommando Zeilen Parameter

Grundsätzlich können bei allen Aufrufen von AVRprog auch Schalter mitgegeben werden. Dies sind:

- USB2** Es werden nur Geräte mit USB2 gesucht
- e** Automatisches Device löschen
- p** Automatischer Programmier Start
- r** Automatischer Target Run
- c** Programm verlassen
- s** Es werden keine visuellen Fehlermeldungen generiert  
Statt dessen werden die Fehlermeldungen in eine Datei geschrieben „ICPISP.err“. Diese Datei befindet sich dann im angegebenen Project Pfad bzw. im Programmverzeichnis, falls die Parameter kein Programmverzeichnis enthalten
- u0** Es wird ein standard Pack File generiert
- u1** Es wird ein encrypted Pack File generiert  
Diese Dateien befinden sich dann im angegebenen Project Pfad.
- gProgSerNum** Werden mehrere Programmer gefunden, dann wird der Programmer mit der Serien Nummer **ProgSerNum** benutzt.

Die Reihenfolge der Schalter spielt dabei keine Rolle. Die interne Verarbeitung erfolgt jedoch in obiger Reihenfolge. Die Schalter müssen durch ein Leerzeichen getrennt sein. Innerhalb der Schalter darf kein Leerzeichen enthalten sein.

Beispiel: **C:\pppp\AVRProg.exe ProjectName -USB2 -p -c**

Ist der Schalter -p aktiv, so ist ein vorheriges Löschen (-e) unnötig, da ein Programmiervorgang automatisch auch ein Löschvorgang beinhaltet.

Ist der Schalter -c aktiv, so ist ein vorheriges Target Run unnötig, da ein Programm Ende auch den Target freigibt.

Für **ProjectName** gibt es 2 Möglichkeiten. Es kann entweder der komplette Pfad und der Namen des gewünschten Steuerfiles „ISPE“ angegeben werden Beispiel: “C:\files\hex\myprog.ispe”  
Oder nur der Namen des Projekts Beispiel: “myprog”



## Return Codes

0	dsOk	Operation erfolgreich beendet
1	dsPwrDown	Target Spannung fehlt
2	dsPwrErr	Target Spannung zu hoch oder zu niedrig
3	dsFalseTyp	Falsche CPU ID gefunden
4	dsProtect	Target ist über Fuses geschützt
5	dsNotEmpty	Target ist nach einem Erase nicht leer
6	dsVerifyErr	Target bzw. Programmer meldet einen Verify Fehler während des Programmierens
7	dsFileError	N.A.
8	dsTimeoutErr	USB Schnittstelle meldet einen Timeout
9	dsCommError	Kommunikations Problem mit dem Programmer
10	dsNoProg	Programmer nicht gefunden
11	dsNoProj	Projekt nicht gefunden
12	dsFwLost	Programmer meldet eine defekte Firmware
13	dsNotFound	File, z.B. Hexfile, wurde nicht gefunden
14	dsCalReq	Programmer meldet Verlust oder illegale Kalibration

## Netzwerke

Bei Netzwerken, die mit DOS Filenamen arbeiten, kommt es zu Problemen mit der Projekt File Extension **.ispe** weil hier nur drei Buchstaben erlaubt sind. Durch einen Eintrag in das INI-File des Programmers selbst **ISPICP.ini** kann die Extension der Projekt Files selbst bestimmt werden.

Beispiel:

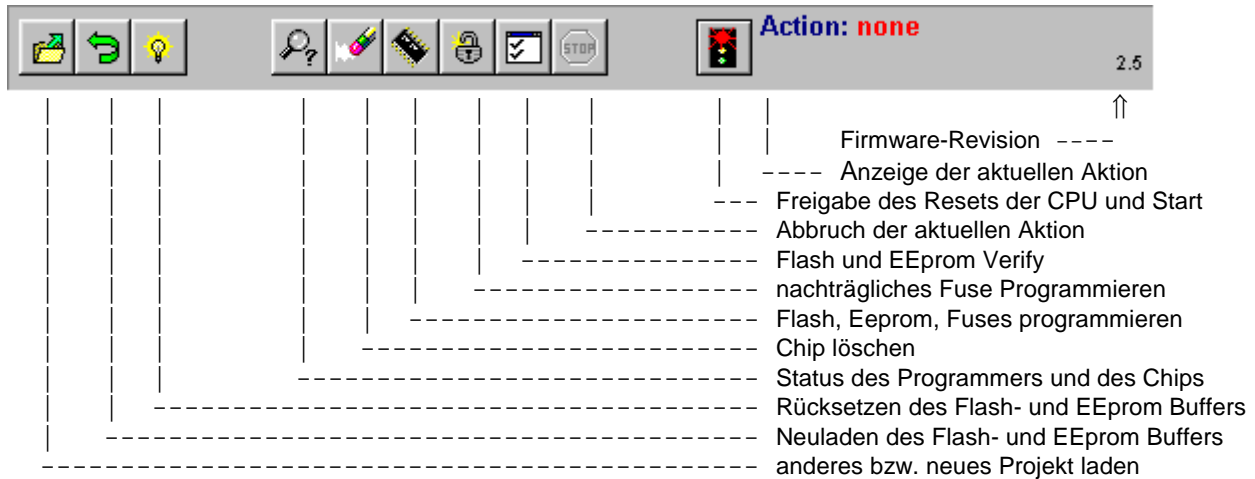
```
[Settings]  
ProjExt=.isp
```

## Alternative Programmerstellung

Wird mit dem vorliegenden Programm ein sog. **PACK** File erstellt, so kann dieses auch mit dem Pack Programmer Programm **PackProg.exe** verarbeitet werden. Dieses Programm ist Bestandteil des Programmer Packets.

## Knopf-Leiste

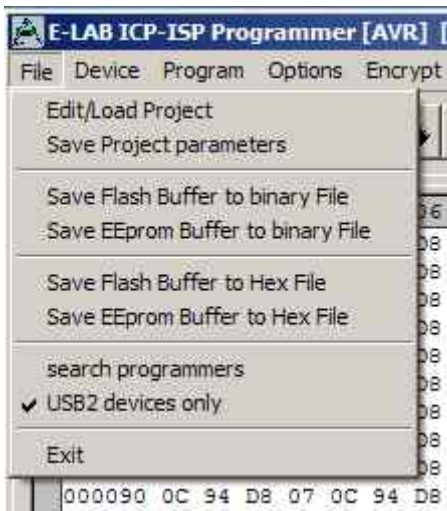
Zur schnellen Bedienung mit der Maus besitzt das Programm eine Leiste mit sog. Speed-Buttons. Diese erlauben ein schnelles Arbeiten ohne Benutzung der Menues.



## Funktionen der Buttons und Menues

Im Normalfall braucht man die Menues nicht zu benutzen. Alle Standard Operationen können mit den sogenannten Speedbuttons mittels Mausklicks gestartet werden. Spezielle Operationen sind nur in den Menues zu finden.

### File Menue



**Edit/Load Project** öffnet den Projekt Dialog. Ein neues Projekt kann angelegt oder ein vorhandenes geöffnet werden.

**Save Project parameters** speichert die aktuellen Parameter in das isp - File.

**Save Flash Buffer to binary File** speichert den Flash-Buffer in ein binär File. Dazu wird ein entspr. File-Dialog geöffnet.

**Save EEPROM Buffer to binary File** speichert den Flash-Buffer in ein binär File. Dazu wird ein entspr. File-Dialog geöffnet.

**Save Flash Buffer to Hex File** speichert den Flash-Buffer in ein hex File. Dazu wird ein entspr. File-Dialog geöffnet.

**Save EEPROM Buffer to Hex File** speichert den Flash-Buffer in ein hex File. Dazu wird ein entspr. File-Dialog geöffnet.

**Search Programmers** ist eine Support Funktion die die aktuelle Programmer Verbindung trennt und anschliessend versucht alle angeschlossenen Programmer zu finden.

**USB2 devices only** verbietet das globale Suchen nach angeschlossenen Programmern und sucht nur nach USB2 Typen. Das verhindert extrem lange timeouts beim COMport suchen, das besonders mit auf dem PC installierten Bluetooth virtuellen Comport mehrere Minuten dauern kann.



Der **Projekt-Open** Button eröffnet den Projekt-Dialog zum Laden, Neuerstellen oder Verändern eines Projekts.



Der **Reload** Button lädt die schon zuvor geladenen Dateien erneut.

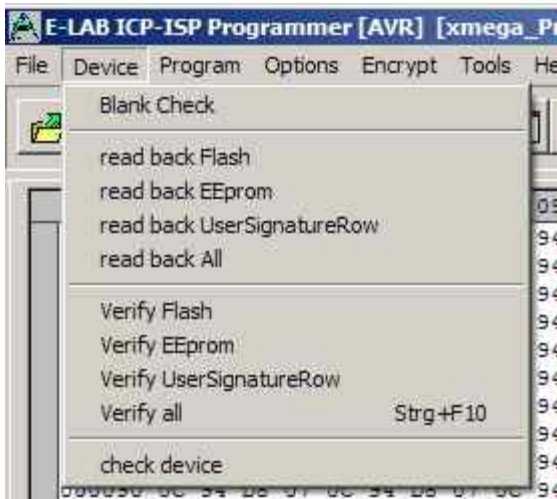


Der **Flush** Button setzt den Flash-Buffer als auch den EEPROM-Buffer komplett auf \$FF



Der **DeviceState** Button öffnet den Device State Dialog. Beschreibung weiter oben.

## Device Menue



**Blank Check** prüft das Target (CPU) auf unprogrammiert bzw. leer. Ist das Target lesegeschützt, so erfolgt eine entsprechende Fehler-Meldung.

**Read back Flash** sofern das Target nicht lesegeschützt ist, wird der Flash Inhalt aus der CPU in den Flash-Buffer zurückgelesen.

**Read back EEprom** sofern das Target nicht lesegeschützt ist, wird das EEprom in den EEprom-Buffer zurückgelesen.

**Read back UserSignatureRow** sofern das Target nicht lesegeschützt ist, wird der UserRow Speicher in den UserRow-Buffer zurückgelesen. Nur XMega.

**read back All** sofern das Target nicht lesegeschützt ist, wird das Flash als auch das EEprom in den jeweiligen Buffer zurückgelesen. Bei XMegas wird auch die UserSignatureRow zurückgelesen.

**Verify Flash** sofern das Target nicht lesegeschützt ist, wird der Flash Inhalt der CPU mit dem Flash-Buffer verglichen. Bei Differenzen erfolgt eine Fehlermeldung.

**Verify EEprom** sofern das Target nicht lesegeschützt ist, wird der EEprom Inhalt der CPU mit dem EEprom-Buffer verglichen. Bei Differenzen erfolgt eine Fehlermeldung.

**Verify UserSignatureRow** sofern das Target nicht lesegeschützt ist, wird der UserRow Inhalt der CPU mit dem UserRow-Buffer verglichen. Bei Differenzen erfolgt eine Fehlermeldung. Nur XMega.

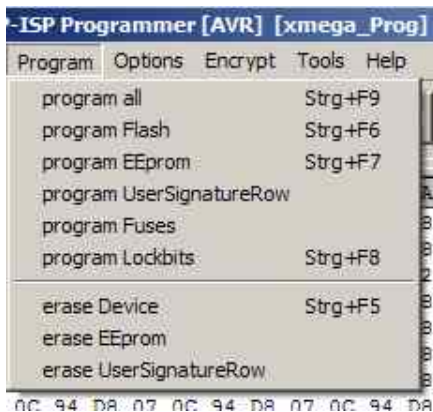
**Verify all** sofern das Target nicht lesegeschützt ist, wird der Flash und EEprom Inhalt der CPU mit dem jeweiligen Buffer verglichen. Bei Differenzen erfolgt eine Fehlermeldung. Bei XMegas wird auch die UserSignatureRow verglichen.

**check device** prüft den Programmer als auch das Target. Evtl. auftretende Probleme werden angezeigt. Wenn CPU auslesegeschützt ist kann allerdings nur festgestellt werden, ob sie reagiert (ID = 00 01 02) oder nicht. Der weiter oben beschriebene **Device State** Dialog wird angezeigt.



Der **check** Button hat die gleiche Aktion zur Folge wie „check Device“ im obigen Device Menue.

## Program Menue



**program all** programmiert das Flash, das EEprom als auch die Fuses und Lockbits wie unter Options eingestellt.

**program flash** programmiert nur das Flash

**program EEprom** programmiert nur das EEprom (nicht im JTAG Mode)

**program UserSignaturRow** programmiert nur die UserRow (XMega)

**program Fuses** programmiert die Fuses wie unter Options eingestellt

**program Lockbits** programmiert nur die Lockbits

**erase Device** löscht das ganze Chip, nicht aber die Fuse Bits

**erase EEprom** löscht das EEprom auf \$FF. (nicht im JTAG Mode)

**erase UserSignaturRow** löscht die UserRow auf \$FF. (XMega)



Mit dem **Erase** Button wird das ganze Chip gelöscht incl. der Lock Bits. Die Fusebits werden weder gelöscht noch verändert.



Mit dem **Programm** Button wird das ganze Chip gelöscht incl. der Lock Bits und anschliessend neu programmiert. Die Fuse und Lock Bits werden wie unter Options eingestellt, behandelt.



Mit dem **Security** Button wird einem schon programmierten Device, das noch nicht geschützt ist, die Lockbits gesetzt, wie unter Options eingestellt.



Mit dem **Verify** Button kann das Target mit den Buffern verglichen werden, vorausgesetzt es ist nicht auslesegeschützt.



Der **Stop** Button bricht eine gerade laufende Aktion ab.



Nach dem Programmieren bleibt der während der einzelnen Aktionen an die CPU angelegte Reset erhalten. Die CPU kann jetzt, ohne den Programmer abzuziehen, durch Klicken der **Run** Taste gestartet werden.

Wenn die Option **autorelease target** im Option Dialog angegeben ist, wird der Reset immer nach einem Programmier Zyklus automatisch freigegeben

## Options Menue

Sämtliche Fuse Bits und Lock Bits, Reset Polarität etc. sowie das gesamte Verhalten des Programmers und seine zusätzlichen Optionen müssen zumindest einmal pro Projekt eingestellt werden. Dazu gibt es zwei Dialoge: den Options Dialog und den Target Options Dialog. Diese werden über das untenstehende Menue aufgerufen.



**Programmer options** startet den Options Dialog wo die Fuse und Lock bits editiert bzw. gesetzt werden können/müssen. Ebenso sind hier weitere Einstellungen zu finden.

**Target options** startet den Target Options Dialog wo mächtige zusätzliche Funktionen freigeschaltet und eingestellt werden können

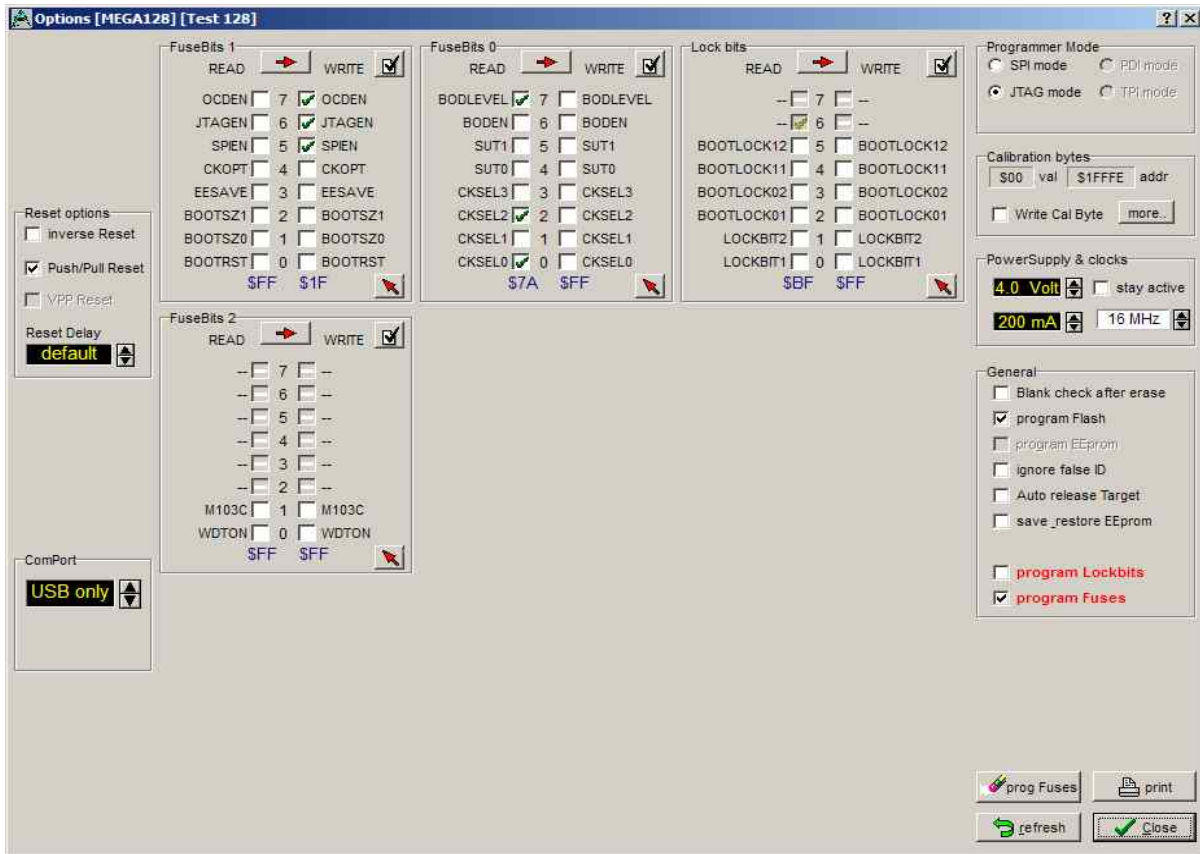
**Download new Firmware** ist die Selbst-Programmierung bzw. Firmware Update. Für weitere Info lesen Sie bitte das Kapitel **Firmware Update** am Ende dieses Handbuchs.

**calibrate programmer** wird beim UPP1-X nicht gebraucht.

unterstützt werden SD Karten mit 2..32 GB. Auch SDHC Typen. Bitte nur Karten mit 4-fach oder schneller benutzen! Für größtmögliche Geschwindigkeit ist die Cluster Größe wichtig. Gute Werte sind: bis 4GB Größe -> 4kB, 8GByte Größe -> 8kB, 16GByte Größe -> 16kB, 32GByte Größe -> 32kB

## Programmer Options Dialog

Mit dem Options-Dialog wird das Verhalten des Programmiers beim Löschen bzw. programmieren des Targets eingestellt. Die Optionen sind zum Teil von den Hex-Files und zum Teil von dem eingestellten CPU-Typ abhängig.



Wurde z.B. kein EEPROM-Hexfile geladen, ist unter General das aktivieren des „program EEPROM“ Feldes nicht möglich.

Hat die CPU keine „Fuse bits“ so sind die Felder **Fuse bits** gesperrt, ansonsten sind die vorhandenen Bits zum Ändern freigegeben. Die Bedeutung der Fuse Bits ist den CPU Manuals von Atmel zu entnehmen

Im Feld **General** wird das allgemeine Verhalten eingestellt. Der Punkt **blank check after erase** ist nur im Zweifelsfall und für Testzwecke sinnvoll. Er sollte ausgeschaltet sein.

**Program Flash** und **program EEPROM** sind im Normalfall beide aktiviert. Ist kein EEPROM Hexfile vorhanden, wird „program EEPROM“ automatisch deaktiviert.

**Ignore false ID** schaltet die Fehlermeldung bei falscher Device-ID ab. Sollte nicht aktiv sein.

**Auto release Target** gibt den Reset der CPU nach der Programmierung automatisch wieder frei.

**Program Fuses** sollte immer aktiv sein und ist nur in Ausnahmefällen zu deaktivieren.

**Program Lockbits** sollte auch immer aktiv sein. Hat aber keine Bedeutung, wenn keins der Lockbits aktiviert ist.

Im Feld **Security** werden die Lockbits für den Ausleseschutz eingestellt. **Lockbit0** oder **Lockbit1** machen für sich allein nur wenig Sinn. Der komplette Schutz wird nur erzielt, wenn beide Lockbits aktiviert sind. Die BootLock Bits sollten nur aktiviert werden, wenn der Boot im AVR auch aktiv ist bzw. als solcher benutzt wird. Die genaue Funktion der Lockbits ist im Datenbuch des jeweiligen Prozessors nachzulesen.

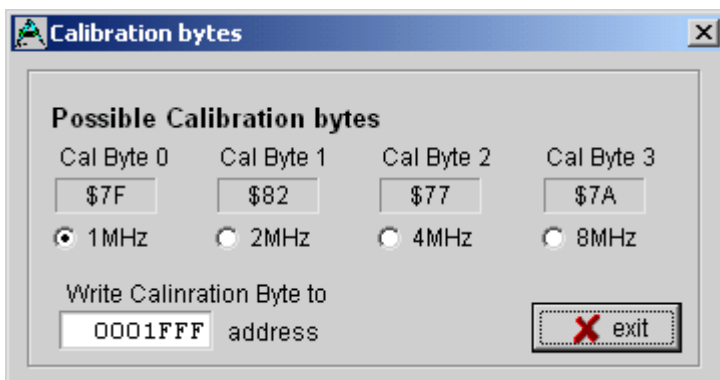
In den Feldern **fuseBits...** kann, falls die CPU diese Funktion bietet, z.B. mega128, die Länge des internen Resets eingestellt werden. Bei anderen Typen ist hier z.B. die Freigabe des RC-Oszillators, Brown-Out usw. vorgesehen. Da jede CPU diese Bits unterschiedlich interpretiert und die Lage der Bits ebenfalls verschieden ist, kann hier keine weitere Angabe gemacht werden. Entsprechende Informationen über die exakte Funktion dieser Bits bitte ebenfalls in den Datenbüchern nachlesen. Ein grüner Haken bedeutet, dass diese Option geschaltet bzw. aktiviert ist (bit = low).

**Achtung:** einige CPUs besitzen eine **SPIEN** Fuse. Diese Fuse ist normalerweise bei SPI-Low-Voltage-Programmierung unwichtig. Es gibt jedoch einige Ausnahmen, z.B. beim Tiny12. Bei diesen Typen muss die SPIEN Fuse aktiviert sein, sonst kann auf die CPU nicht mehr zugegriffen werden.

Die **SPIEN** Fuse kann im **JTAG Mode** immer geschrieben werden. Wenn **SPIEN** und **JTAGEN** ausgeschaltet sind, besteht kein Zugriff mehr. Beachten Sie, dass JTAG ganz bestimmte Pins benötigt, die dann nicht mehr für Standard Funktionen zur Verfügung stehen.

Die boolsche Darstellung mit den farbigen Haken ist eindeutig, wenn es um einzelne Fuses geht. Bei kombinierten Fusebits bzw. Bitfeldern, wie z.B. den Oscillator bits, ist eine binäre Darstellung aufschlussreicher. Mit diesem Button kann die Darstellung umgestellt werden.

CPUs die einen internen RC-Oszillator besitzen, haben oft auch eine oder mehrere read-only fuses „Calibration byte“, das dann im Feld **Calibration Byte** angezeigt wird. Dieses Byte kann vom Programm/Applikation benutzt werden, um den RC-Oszillator abzustimmen. Da dieses Byte in jeder einzelnen CPU unterschiedlich ist, muss es individuell dem Programm mitgeteilt werden. Mit der Checkbox **WriteCal Byte** wird dem Programmierer mitgeteilt, dass dieses Fuse-Byte beim Programmieren ausgelesen werden und ins Flash geschrieben werden muss. Die Ziel-Adresse für dieses Schreiben wird mit dem untenstehenden Dialog „Calibration bytes“ eingestellt. Je nach CPU Typ werden bis zu 4 Calibration Bytes bereitgestellt, die



aus der CPU ausgelesen werden. Jedem Byte ist ein bestimmter RC-Oszillator zugeordnet. Die Wahl dieses Bytes (Radio Button) muss deshalb mit der Einstellung der CLKSEL-Fuses übereinstimmen. Im Editfeld „address“ muss die Ziel-Adresse für dieses Byte im Flash eingestellt werden. Bei CPUs bis 64kByte Flash kann dies jede Adresse > \$0000 sein. Bei CPUs > 64kB Flash (mega128) können nur geradzahlige Adressen > \$0000 verwendet werden.

Bei modernen AVRs ist beim Standard RC-Oszillator kein Calibration Byte notwendig.

Diese laden dieses Byte automatisch aus dem dafür vorgesehenen EEPROM Speicher in das OSCAL Register.

**Achtung:** Alle Fuse und Lockbits sind low active. D.h. wenn im Datenblatt für ein bestimmtes bit eine „0“ steht, muss in dem zugehörigen Feld im Dialog ein Haken sein. Ein log 1 bit wird deshalb durch ein leeres Feld dargestellt. Atmel verwendet für Fuse und Lockbits immer negative Logik!

Im Feld **ComPort** wird die zu verwendende Schnittstelle ausgewählt. Für den ISP3-X ist die Einstellung „USB only“ korrekt. Werden an diesen PC jedoch auch noch V24 Programmierer Typen angeschlossen, so empfiehlt sich die Einstellung „Automatic“.

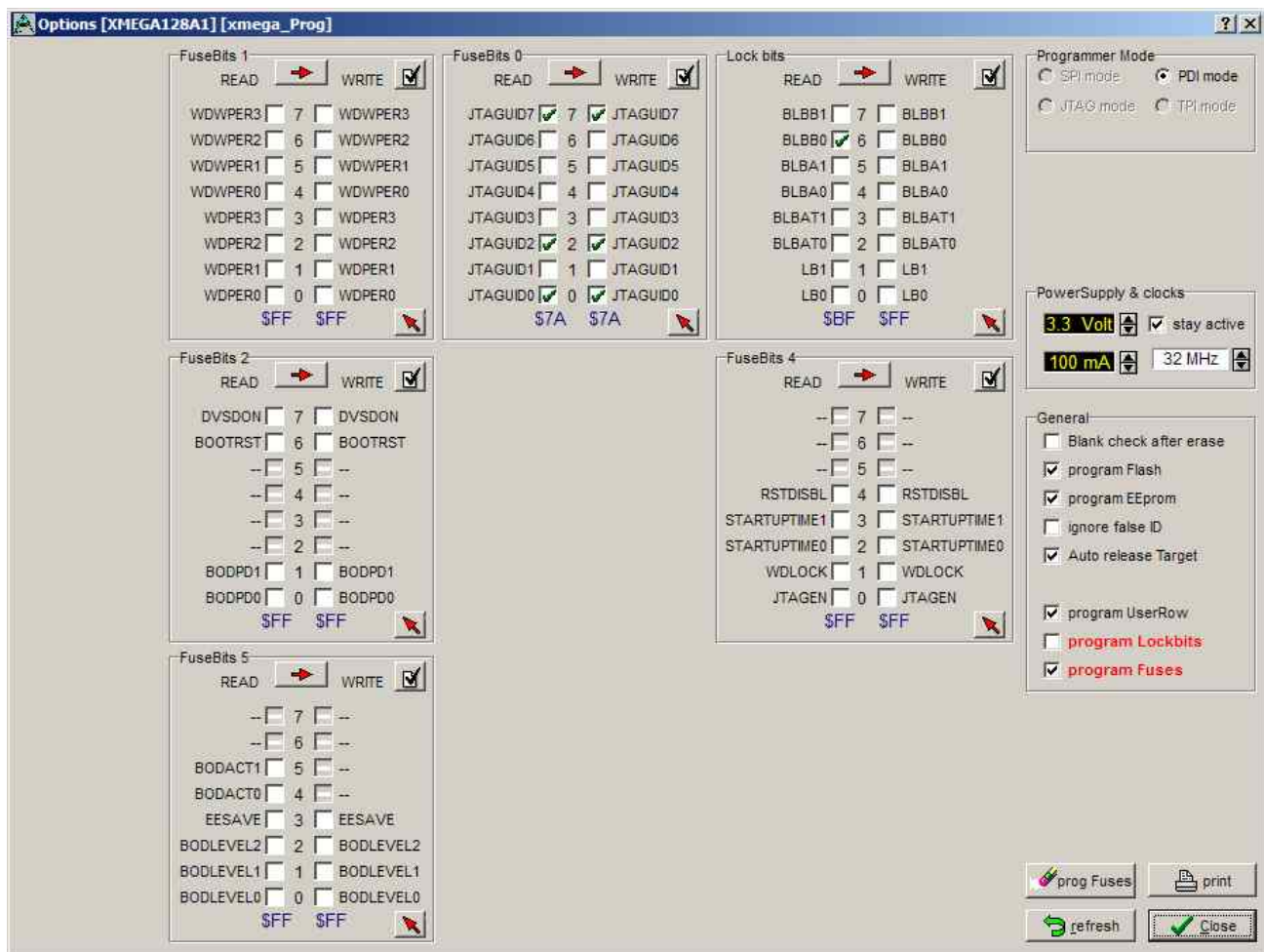


## ISP3-X portabler In-Circuit Programmer

In dem Feld **Reset options** wird das Steuerverhalten des ISP gegenüber der Target CPU eingestellt. Im Normalfall sind alle drei Einstellmöglichkeiten inaktiv bzw. default. Für spezielle Target-Hardware kann z.B. der Reset gegenüber dem Normalfall invertiert werden. Push/Pull Reset **off** ist nur sinnvoll, wenn auf den Reset-Eingang des Targets noch weitere Elektronik zugreifen kann. Diese ist dann jedoch mit einem Widerstand zu entkoppeln. Das Reset-Delay (die Zeit wie lange der Reset inaktiv sein muss, bevor er wieder aktiviert werden kann), kann nach Wunsch verlängert werden. Nur für spezial Fälle.

Das Feld **Programmer Mode** ist nur sichtbar, wenn die gewählte CPU entweder SPI, JTAG, TPI oder PDI Programmierung unterstützt. In diesem Fall kann hier der Modus eingestellt werden.

## XMega Options Dialog



Einige Optionen sind bei den XMegas nicht möglich oder nicht sinnvoll. Der obenstehende Dialog ist typisch für einen XMEGA. Zu beachten ist dass es hier eine Checkbox **program UserRow** gibt.

Die Fuse **JTAGEN** kann immer unprogrammed sein, da hier das JTAG Interface nie benutzt wird sondern immer das PDI Interface.

Manche Kombinationen in den Fusebits1, Fusebits2 und Fusebits5 sind laut Datenblatt illegal und sollten vermieden werden. Es kann sonst zu unerwünschtem Verhalten der CPU kommen.

Die max. einstellbare Ausgangs Spannung bei den XMegas ist 3.6Volt.



## Power Supply

Das **Power Supply** Feld dient zum Einstellen des maximal zulässigen Stroms. 0.0mA sperrt die ISP-interne Versorgung des Ziel Systems. Die anderen Werte geben die interne Versorgung frei. Wenn ein Strom > 0mA eingestellt ist, kann die gewünschte zugehörige Spannung eingestellt werden. Wenn freigegeben, versorgt der ISP Programmer das Ziel System mit der eingestellten Spannung.

Die Strombegrenzung ist einstellbar zwischen 30mA und 300mA. Der Strom für das Zielsystem wird auf diesen Vorgabe Wert begrenzt. Achtung: PC-gespeiste externe USB-HUBs können oft nur 100mA liefern.

Die aktivierte Versorgung wird normalerweise nach dem Programmiervorgang ausgeschaltet. Soll die Spannung hier weiterhin erhalten bleiben, so ist die Checkbox **stay active** zu aktivieren. Wenn jetzt noch dazu **auto release target** aktiviert ist, läuft das geprüfte System an und es kann zur Laufzeit getestet werden.

Die veränderbaren Fuse- und Lock Bits werden jeweils auf der rechten Seite in der **Write Spalte** angezeigt und editiert. Die **Read Spalten** können mit dem Refresh Button jederzeit aktualisiert werden. Dazu werden die Lock und Fuse bits aus der CPU ausgelesen, soweit möglich.

Der Button **Erase Fuses** ist sehr nützlich zum löschen von illegalen Fuses, die durch eine Fehl-Programmierung entstanden sein können. Die Löschung erfolgt durch Überschreiben der Fuses mit den gewünschten Werten.

### Achtung (SPI mode):

Manche CPUs besitzen interne RC-Oszillatoren oder die Möglichkeit eines externen RC-Oszillators. Diese Optionen werden über die Fuse-Bits eingestellt. Manchmal ist es auch möglich einen externen Low-Frequency Quartz vorzugeben. Hierbei ist folgendes zu beachten:

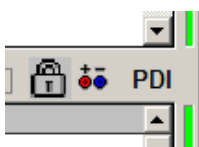
1. Interner RC-Oszillator.  
Hier ist die Standard Frequenz ca. 1MHz. Dadurch muss im Programmer auch 1MHz als CPU-Frequenz eingestellt werden, ansonsten ist mit Fehlern zu rechnen. Die Frequenz ist nominal 1MHz. Bei einer CPU-Spannung von 3Volt ist von ca. 500kHz auszugehen.
2. Externer RC-Oszillator.  
Wird dieser Mode aktiviert, muss auch unbedingt eine solche Schaltung angeschlossen werden, sonst kann nach dem Programmieren nicht mehr auf die CPU zugegriffen werden.
3. Low Frequency Crystal.  
Wird dieser Mode aktiviert, muss auch unbedingt ein 32kHz Quartz angeschlossen werden, sonst kann nach dem Programmieren nicht mehr auf die CPU zugegriffen werden.

Zu beachten ist weiterhin, dass beim Programmieren die aktuelle Fuse Einstellung gilt. Die neu programmierte wird frühestens nach dem Programmier Vorgang gültig. Manche Fuses werden auch erst nach einem PowerDown aktiv.

Bei falsch programmierten Oszillator Fuses kann es sehr gut sein, dass anschliessend der externe Oszillator umgebaut werden muss, um wieder über den Programmer auf die CPU zugreifen zu können.

Im JTAG, TPI und PDI Mode bestehen jedoch keinerlei Restriktionen, hier kann immer programmiert werden solange die CPU Spannung hat. Ein Oszillator ist dazu nicht erforderlich. Bedingung ist allerdings dass bei JTAG die JTAGEN Fuse aktiviert ist und auch so bleibt.

Der Vorgabe Status der Lockbits und die Versorgung des Targets durch den ISP werden im Hauptprogramm auch optisch angezeigt.

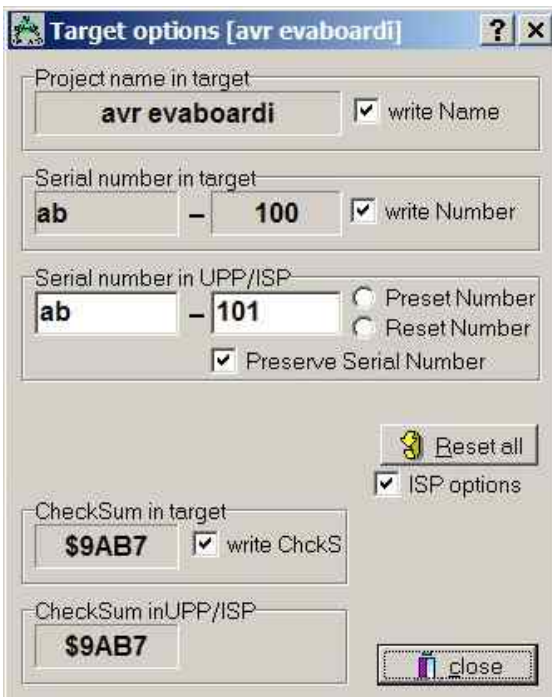


Das Schloss bedeutet hier, dass für das Projekt der Ausleseschutz (Lockbits) für die Ziel CPU aktiviert ist. Das Netzteil Zeichen bedeutet, dass der ISP (für das momentan im PC geladene Projekt) das Target System mit Strom/Spannung versorgen muss.

## Dialog Target Options

Dieser Dialog hat zwei Aufgaben:

1. Speichern/Programmieren von zusätzlichen Daten ins Flash des Targets.
2. Serien Nummern pflegen.



Im Titel steht der Name des im PC geladenen Projekts. Falls ein Projektname im Target abgespeichert ist und dieser lesbar ist, wird er in diesem Feld angezeigt. Der CheckButton **write Name** zeigt den Status der Option an.

**Serial number in target** zeigt die ausgelesene Seriennummer im Target an, falls vorhanden. Der Check-Button **write Number** zeigt den Status dieser Option an.

**Serial number in UPP/ISP** zeigt die ausgelesene Seriennummer im ISP bzw. PC an. Diese wird grundsätzlich bei jedem Programmiervorgang um eins erhöht.

Mit **Preserve Serial Number** wird die im Target vorhandene Serien Nummer wieder verwendet.

**Reset all** setzt alle Optionen zurück, auch die Serien Nummer.

**ISP options** ist die generelle Freigabe dieser Optionen.

**Checksum in target** zeigt die dort evtl. vorhandene Flash Checksumme an, der Check-Button **write ChkSum** den Status dieser Option.

**Checksum in UPP/ISP** zeigt die im PC vorhandene Flash Checksumme an.

## Speichern der Parameter im Flash

Vorausgesetzt es ist noch ausreichend Platz in dem Flash Bereich der Target-CPU, so können hier am oberen Ende des Flashs Speichers zusätzliche Informationen abgelegt werden. Mit den Optionen-Schaltern (CheckBox) des obigen Dialogs werden die einzelnen Optionen freigeschaltet oder gesperrt.

Diese Optionen werden auch in das INI-File des Projekts abgelegt. Die Optionen werden nach jedem Programmiervorgang am Ende des Flash abgelegt.

Bei jedem Aufruf dieses Dialogs wird versucht, falls angeschlossen, aus dem Target die Istwerte auszulesen. Das letztere kann natürlich nur funktionieren, wenn die Target-CPU nicht auslesegeschützt ist. Das Programm in der Target CPU hat immer Zugriff auf diese Daten.

## Projekt Name ins Flash

Durch Anklicken (aktivieren) der CheckBox **write Name** wird der ISP angewiesen den Projekt Namen in das Flash zu programmieren, d.h. am Ende eines Programmiervorgangs. Das deaktivieren dieser CheckBox schaltet diese Programmierung wieder aus.

## Serien Nummer ins Flash

Das Aktivieren der CheckBox **write Number** weist den Programmer an, die Serien Nummer ins Flash zu programmieren. Der Integer Teil dieser Nummer wird anschliessend inkrementiert. Die Serien Nummer besteht aus 2 Teilen:

1. Zwei beliebige Zeichen im Feld **Serial Number in UPP/ISP**. Diese werden unverändert übernommen.
2. Aus einer Zahl im Feld rechts daneben. Diese Zahl kann mit **Reset Number** entweder auf 0 zurückgesetzt oder mit **Preset** vorbesetzt werden.

Die Anzahl der bis jetzt programmierten CPUs kann hier indirekt abgelesen werden.

## Preserve Serial Number

Wurde die Serien Nummer freigegeben, so kann mit dieser Option verhindert werden, dass eine neue Nummer ins Flash geschrieben wird. Ist das Target nicht protected, dann wird dessen interne Nummer ausgelesen und zur Neu-Programmierung des Flash wiederverwendet. Damit kann sichergestellt werden, dass sich die Nummer nicht ändert, trotz einer Neuprogrammierung.

## Checksumme ins Flash

Die aus dem Flash-Hexfile generierte 16bit Checksumme wird im Feld **CheckSum in ICP/ISP** angezeigt. Dieser Wert kann mit der CheckBox **write CheckSum** in die Target CPU abgelegt werden. Die Checksumme enthält nur Werte des Flash-Hexfiles. Die vom ISP zusätzlich abgelegten Werte und evtl. vorhandene, vom File nicht betroffene Speicherteile werden nicht berücksichtigt.

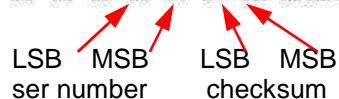
## Reset All

Mit diesem Button werden alle Werte und Optionen zurückgesetzt.

Bemerkung: alle Änderungen werden sofort gespeichert. Sie wirken sich allerdings erst beim nächsten Programmiervorgang aus.

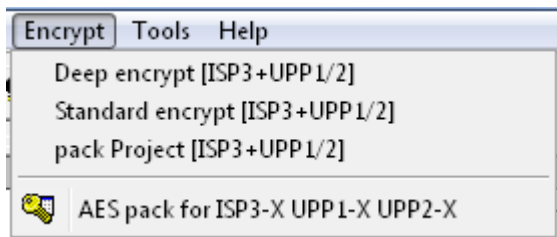
Das Anwendungs Programm kann auf diese Parameter zugreifen. Die Parameter sind in den letzten 16 bzw. 32 Bytes des Flash abgelegt. Der Projekt Name ist durch ein vorangestelltes „proj“ gekennzeichnet, die Serien Nummer durch ein „ser#“. Die Checksumme steht immer auf den 2 letzten Bytes des Flash. Reihenfolge der Serial Number und der Checksum ist loByte/hiByte. Beispiel:

```
01FFEO FF FF 70 72 6F 6A 0D 61 76 72 20 65 76 61 62 6F ..proj.avr evabo
01FFFO 61 72 64 69 BA 00 73 65 72 23 61 62 65 00 B7 9A ardi...ser#abe...
```





## Encrypt Menu



Diese Menu-Punkte eröffnen einen Encrypt oder Pack Dialog. Mit Hilfe eines dieser Dialoge können **verschlüsselte** oder **gepackte** binäre Dateien erstellt werden, die nur noch durch das beigefügte Programm **PackProg.exe** gelesen und verarbeitet werden können. Alle drei File Typen benötigen beim ISP3-X als Ziel Programmer das Programm PackProg.exe

Für die Typen ISP3-X, UPP1-X und UPP2-X sollte nur pack Project [ISP3+UPP1/2] oder AES pack for ISP3-X UPP1-X UPP2-X verwendet werden

Bei **gepackten** Dateien bzw. Projekten besteht nur ein geringer Schutz. Alle Hex Files und auch die Programmier Vorgaben, Fuses etc. werden in eine binäre Datei geschrieben so dass der Empfänger eigentlich keine weiteren Programmier Kenntnisse haben muss und auch keinerlei Einstellungs Veränderungen vornehmen kann oder muss.

Diese **Verschlüsselung** (encrypt) erweitert eine gepackte Datei dahingehend, dass Programm Dateien rund um die Welt verschickt werden können, ohne dass der Empfänger oder andere die Möglichkeit haben, die Datei zu disassemblieren oder anderweitig ein sogenanntes „Re-Engeneering“ anzuwenden.

Weiterhin besteht die Möglichkeit, das encryptete File mit einem Passwort zu versehen, so dass nur der richtige ISP3-X Programmer diese Datei verarbeiten kann. Ein Abzweigen von Produkten für dunkle Zwecke wird damit ziemlich erschwert.

**Deep** encrypted Files können nur noch mit dem Programm PackProg.exe verarbeitet werden.

**Standard** encrypted Files können direkt von den UPP Programmern geladen werden. Zum Verarbeiten mit dem ISP3-X wird zwingend das Programm PackProg benötigt.

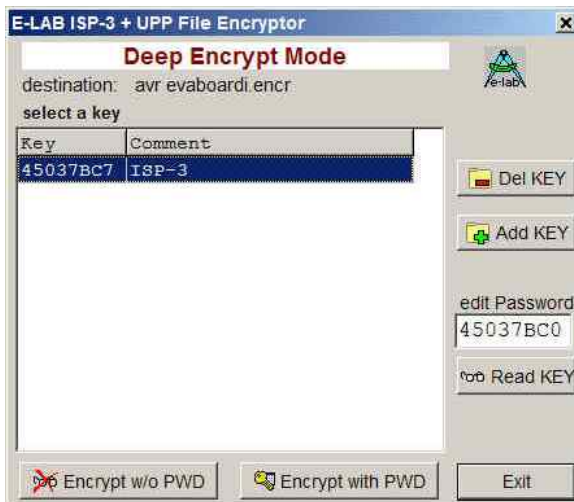
**Packed** Files können direkt von den UPP Programmern geladen werden. Zum Verarbeiten mit dem ISP3-X wird zwingend das Programm PackProg benötigt.

### **AES pack for ISP3-X UPP1-X UPP2-X**

Hier wird grundsätzlich das absolut sichere AES 128bit Verschlüsselungs Verfahren angewendet. Ist als Ziel Programmer einer der X-Typen vorgesehen, ist das AES allen anderen vorzuziehen.

## Deep encrypt Modus

Dieser Modus sollte für die neueren Programmer (ISP3-X, UPP1-X, UPP2-X) nicht verwendet werden, da diese das wesentlich bessere **AES** Verfahren verwenden. Diese Option generiert ein gepacktes und **extrem verschlüsseltes** Projekt, das nur mit dem Programm **PackProg.exe** gelesen und verarbeitet werden kann. Als Option kann hierbei noch ein Passwort angegeben werden, so dass eine Verarbeitung nur mit dem Programmer möglich ist, von dem das Passwort generiert wurde. Der Menu Punkt eröffnet untenstehenden Dialog:



Neue Passwörter können mit dem *Add Key* Button hinzugefügt, vorhandene mit dem *Del Key* Button gelöscht werden.

Mit dem *Encrypt with PWD* Button wird ein Passwort geschütztes File erstellt. Ein Passwort Schutz bindet das generierte File absolut an den Programmer der dieses Passwort generiert hat!

Mit dem *Encrypt w/o PWD* Button wird ein verschlüsseltes aber nicht Passwort geschütztes File generiert und gespeichert.

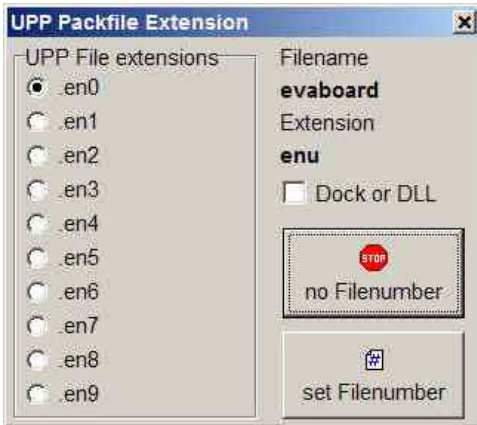
Die Generierung eines ISP3-X Passworts wird auch im Abschnitt Arbeiten mit dem ISP3 beschrieben.

Ist der Ziel Programmer schon an diesem PC angeschlossen, so kann auch hier mit dem Button *Read KEY* dessen Passwort ermittelt werden.

Dieser File Typ kann nur mit dem Programm PackProg.exe verarbeitet werden.

## Standard encrypt Modus

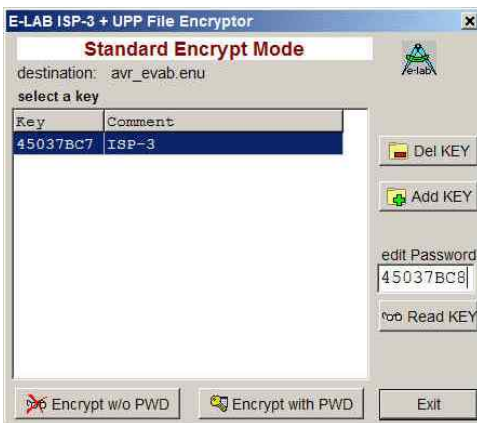
Diese Option generiert ein gepacktes und **gut verschlüsseltes** Projekt, das nur mit dem Programm **PackProg.exe** in Verbindung mit dem ISP3-X gelesen und verarbeitet werden kann. Die UPP Programmer können diese Dateien aber auch direkt laden und verarbeiten. Als Option kann hierbei noch ein Passwort angegeben werden, so dass eine Verarbeitung nur mit dem Programmer möglich ist, von dem das Passwort generiert wurde. Der Menu Punkt eröffnet untenstehenden Dialog:



Hiermit wird die File Extension (Endung) festgelegt. Die Endung kann entweder eine Nummer enthalten, also en0..en9 zur besseren Verarbeitung in den UPP Programmern oder ein einfaches \*.enu sein. Die Auswahl erfolgt mit den zwei File Nummern Buttons.

Die Checkbox *Dock or DLL* ist für den ISP3-X irrelevant.

Wurde die Extension Option gewählt, öffnet sich unten stehender Dialog:



Dieser Dialog dient in erster Linie dazu zu bestimmen, ob ein Passwort vergeben werden soll, was dann das generierte File ausschliesslich an den Programmer bindet der dieses Passwort gebildet hat. Wird ein Passwort benötigt, dann muss es aus dem Listenfeld ausgewählt werden.

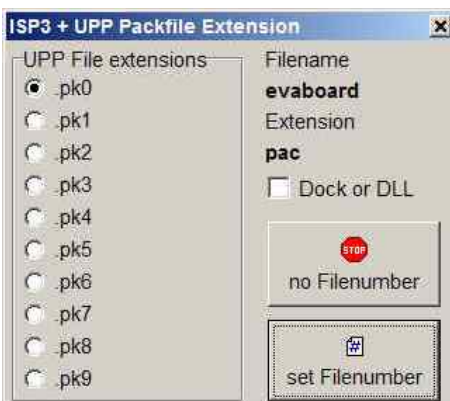
Diese Option wird durch den Button *Encrypt with PWD* gewählt bzw. mit dem Button *Encrypt w/o PWD* gesperrt.

Ist der Ziel Programmer schon an diesem PC angeschlossen, so kann auch hiermit das Passwort ermittelt werden. Mit dem Button *Read KEY* wird das Passwort aus dem Programmer herausgelesen und angezeigt. Mit dem Button *Add Key* kann das Passwort dem Passwort Pool hinzugefügt werden.

Dieser Datei Typ kann mit dem Programm PackProg.exe und auch direkt mit den UPP Programmern verarbeitet werden.

## pack Project Modus

Diese Option generiert ein gepacktes aber **unverschlüsseltes** Projekt, das nur mit dem Programm **PackProg.exe** in Verbindung mit dem ISP3-X gelesen und verarbeitet werden kann. Die UPP Programmer können diese Dateien aber auch direkt laden und verarbeiten. Es ist hierbei kein Passwort möglich und somit auch keine Bindung an einen speziellen Programmer.



Der Menu Punkt eröffnet nebenstehenden Dialog:

Hiermit wird die File Extension (Endung) festgelegt. Die Endung kann entweder eine Nummer enthalten, also pk0..pk9 zur besseren Verarbeitung in den UPP Programmern oder ein einfaches \*.pac sein. Die Auswahl erfolgt mit den zwei File Nummern Buttons.

Die Checkbox *Dock or DLL* ist für den ISP3-X irrelevant.

Dieser Datei Typ kann mit dem Programm PackProg.exe und auch direkt mit den UPP Programmern verarbeitet werden.

## Vorgehensweise

1. Erstellen Sie ein Projekt wie sonst auch üblich oder laden Sie ein vorhandenes Projekt.
2. Nehmen Sie die gewünschten Einstellungen vor, die für dieses Projekt vorgesehen sind.
3. Wählen Sie eines der drei Ziel File Typen aus, stark verschlüsselt, mässig verschlüsselt oder nur gepackt.
4. Bei den beiden verschlüsselten File Typen können Sie noch ein Passwort vorgeben, so dass das generierte File an einen ganz bestimmten Programmer gebunden ist.
5. Klicken Sie den Button „Encrypt“ bzw. die File Number Buttons.
6. Kopieren Sie die generierte Datei auf einen Datenträger
7. Falls der Empfänger das Programm *“PackProg.exe“* noch nicht besitzt, kopieren Sie diese ebenfalls auf den Datenträger.
8. Versenden Sie den Datenträger an den Empfänger.

Das Versenden kann natürlich auch per email erfolgen. In diesem Fall ist es der sicherste Weg, alle Dateien zu „zippen“.

## Verschlüsselung

Zum Verschlüsseln kommt ein sehr sicherer Algorithmus zum Einsatz. Leider muss man sagen, dass es bis jetzt noch keinen solchen gibt, der noch nicht geknackt worden ist. Allerdings ist hier der Aufwand so gross, dass es einfacher ist, das Programm selbst neu zu schreiben. Und damit hat es seinen Zweck auch erfüllt ☺

## Passwort

Das Programm *“PackProg.exe“* generiert auf Wunsch auf dem Zielsystem ein Passwort, das aus dem angeschlossenen ISP3-X, UPP1-X oder UPP2-X Programmer ausgelesen wird. Dieses Passwort gilt nur für diesen Programmer und ist grundsätzlich nicht auf andere Programmer übertragbar. Damit wird sichergestellt dass nur mit diesem Programmer programmiert werden kann.

## Passwort Erstellung

1. Der Chip/Board Programmierer installiert das Programmiersystem *PackProg* auf dem Rechner, der in Zukunft zum Programmieren benutzt werden soll. Dazu müssen die Datei *“PackProg.exe“* in das gewünschte Directory kopiert werden.
2. Das Programm *“PackProg.exe“* wird aufgerufen.
3. Der ISP3-X Programmer muss an diesem PC angeschlossen sein.
4. Das Passwort (Anzeige über „Setup/request Password“) muss dem Hersteller der Programmier Dateien übermittelt werden.
5. Der Hersteller trägt das Passwort im Programm *„AVRprog.exe“* auf seinem System ein:
  - a. Programm Aufruf von *„AVRprog.exe“*
  - b. Über den Menu-Punkt „Encrypt/PackProg Encrypt“ den Encrypt Dialog aufrufen.
  - c. Den Button „Add“ klicken.
  - d. Einen beliebigen Kommentar oder Namen in das linke Feld eintragen.
  - e. Das übermittelte Passwort in das rechte Feld eintragen.
  - f. Die Eingaben mit dem Button „Add“ abspeichern.
6. Bei allen Dateien, die an diesen Empfänger mit diesem Passwort gesendet werden sollen, muss vor dem klicken des Encrypt-Buttons sichergestellt sein, dass das korrekte Passwort eingestellt ist. Alternative siehe Punkt 7
7. Mit dem Button „Public“ wird kein Passwort vergeben und alle Empfänger, die das Programm *“PackProg.exe“* besitzen, können diese Datei verarbeiten.

## Programmer

Das Programm *“PackProg.exe“* unterstützt alle Programmer der Typen *ISP3*, *UPP1* und *UPP2*. Weitere Informationen dazu im nächsten Abschnitt.

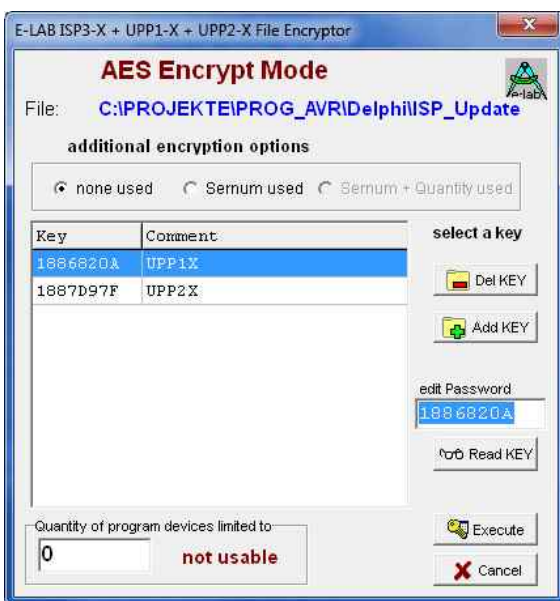
## AES PAC Files

Sollen absolut sichere PAC Files für die Typen ISP3-X, UPP1-X oder UPP2-X („X“ Programmer) erstellt werden, so müssen diese mit der **AES Verschlüsselung** erzeugt werden. AES ist eine absolut sichere Verschlüsselung die nicht „geknackt“ werden kann. Da die Entschlüsselung nur **in** den „X“ Programmer Typen erfolgen kann, ist damit auch ein Mithören (sniffing) auf den USB Leitungen ausgeschlossen.

Weiterhin bietet der AES Mode den Vorteil ein PAC File für einen ganz **bestimmten Programmer**, unter Einbeziehung dessen Serien Nummer, zu erstellen das auf keinem anderen Programmer benutzbar ist. Noch weitergehend ist das Feature, dass hier auch eine **Stückzahlen Begrenzung** angegeben werden kann. Somit lassen sich heimliche „schwarze“ Produktionen absolut unterbinden.

Daher sollte zumindest für ausser-haus Produktion die AES Verschlüsselung verwendet werden. Für in-haus genügt aber auch das „normale“ PAC File.

- |   |                                       |                   |
|---|---------------------------------------|-------------------|
| 1 | pack Project [ISP3+UPP1/2]            | Standard PAC File |
| 2 | pack Project for ISP3-X UPP1-X UPP2-X | AES PAC File      |



Zum erzeugen eines AES PAC ist es unabdingbar, dass ein „X“ Programmer am PC angeschlossen ist.

Der Dialog zum erzeugen von AES PAC Files gestattet drei unterschiedliche Modi auszuwählen.

- Standard Encryption (only AES used)
- Encryption mit Passwort (Sernum used). Hier wird die Seriennummer des Ziel Programmers benötigt. Diese wird in die Verschlüsselung integriert und vom Ziel Programmer geprüft. Der Ziel Programmer wird hierzu nicht benötigt, nur seine Serien Nummer.
- Stückzahl Begrenzung (Sernum+Quantity used). Hier wird ebenfalls die Seriennummer des Ziel Programmers benötigt. Weiterhin wird hier die maximal mögliche zu programmierende Stückzahl vorgegeben. Wird diese im Betrieb erreicht, dann weigert sich der Programmer weitere Targets zu programmieren.



Die AES Eigenschaften eines PAC Files können auf dem Ziel Programmer im Tool PackProg über „File Info“ abgefragt werden.

### **Achtung:**

AES encrypted PAC Files können auch von jedem „alten“ Programmer Typ (ISP3-USB, UPP1-USB, UPP2-USB) abgearbeitet werden. Da diese jedoch nicht in der Lage sind solche Files zu entschlüsseln wird natürlich nur Nonsens programmiert.



## Projekt Files erstellen

Die vorherigen Kapitel zeigten zwei Arten um Pack Prog Projekt Files zu erstellen. Entweder durch direkten Download auf die SD Karte im UPP oder durch ein Flash Laufwerk im PC.

Es wurde angemerkt, dass der Standard UPP1 nur ein einziges Projekt im portable mode benutzen kann, da es keine Möglichkeit gibt zwischen mehreren Projekten auszuwählen.

Deswegen gibt es die **Version S** und die **Version D** des UPP1. Version ‚S‘ hat einen Drehschalter an der Unterseite, der die Auswahl von 10 gespeicherten Projekten ermöglicht. Die Version ‚D‘ unterstützt zusammen mit einer Docking Station ebenfalls 10 Projekte.

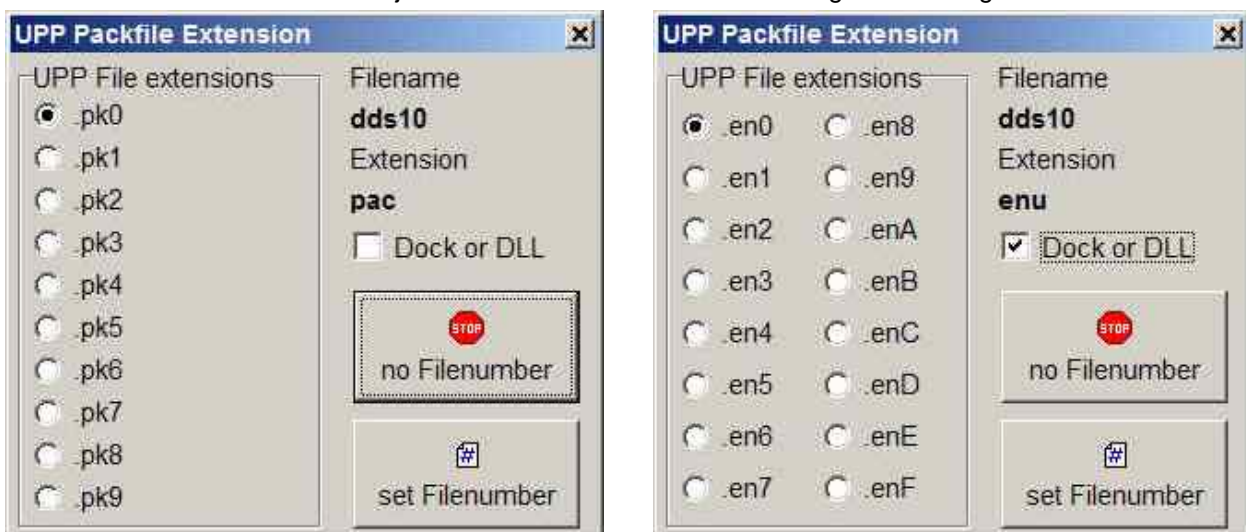
Nun ergibt sich das Problem, dass die Zuordnung der Projekte auf der SD Karte zur Schalterstellung nicht statisch ist. Es ist abhängig von den FAT Einträgen auf der SD Karte. Nur wenn man eine Karte komplett löscht dann entspricht die Reihenfolge des Speicherns der Schalter Position.

Wenn nun jedoch Files gelöscht/neu geschrieben/upgedatet werden kann sich die Reihenfolge drastisch ändern. Folglich müssten nach jeder Änderung die Files in der neuen Reihenfolge in den UPP File-Dialog des PC Programms kopiert und umsortiert werden. Unterlässt man das, ergeben sich größte Probleme bei der portablen Programmierung,

Um diesem Missstand abzuwehren, besteht die Möglichkeit einer absoluten Relation zw. File/Projekt und Schalterstellung des UPP1/S.

Standardmäßig haben alle Pack Files die File Extension **\*.pac** und die verschlüsselten Files die Extension **\*.enu**. Um eine feste Zuordnung eines solchen Projekts zu einer Schalterstellung vorzugeben können die Zahlen 0 bis 9 angehängt werden

Wenn ein **PAC** or **ENU** File /Projekt erstellt werden soll öffnet sich folgender Dialog:



**no Filenumber** wählt die Erstellung eines Standard \*.pac / \*.enu Projekt. **set Filenumber** wählt die Erstellung mit der auf der linken Seite angegebenen Extension. **.pk0, .pk1, ...** oder **.en0, .en1, ...** Dadurch wird die feste Zuordnung zu einer Schalterstellung unabhängig von der Lage des Projekts auf der SD Karte vorgegeben.

Der UPP weist Files mit **Extension Nummern** zurück, wenn diese schon auf der SD Karte mit der gleichen Nummer existieren. Folgende Dateien können co-existieren:

```
DDS10.pac
DDS10.enu
DDS10.pk0
DDS10.en1          etc.
```



## Programmieren mit AVRprog



Das interaktive Programmieren wird mit diesem Button der Programmer Oberfläche gestartet. Das ganze Chip wird gelöscht incl. der Lock Bits und anschliessend neu programmiert. Die Fuse und Lock Bits werden wie unter Options eingestellt, behandelt.

## Status, Fehleranzeige und Probleme

Mögliche Fehlermeldungen des Programmiersystems (AVRprog PC):

### ProgBox not found

- Der Programmer ist nicht am PC angeschlossen
- Der COM-Anschluss des PC's ist durch ein anderes Programm belegt (gilt nicht bei USB).
- Es sind nicht alle Leitungen der 9poligen SUB-D Verbindung PC-Programmer angeschlossen  
Bitte lesen Sie hierzu auch den Abschnitt **Anschlüsse**

### Target Power down

- Der 6polige Stecker des Programmers ist nicht in das Ziel-System eingesteckt
- Das Zielsystem ist stromlos oder dessen Spannung ist zu niedrig ( < 3Volt)
- Das Projekt Setup erwartet, dass der ISP das Target mit Spannung versorgt aber die Stromaufnahme des Targets ist zu hoch.

### Device not responding

- Spannung des Targets zu gross oder zu klein (siehe weiter unten)
- Chip defekt
- Reset kommt nicht an der CPU an
- CPU-Oszillator schwingt nicht (SPI Mode).

### Wrong Device ID

- Spannung des Targets zu gross oder zu klein (siehe weiter unten)
- Chip defekt

Es ist möglich, dass eine Device-ID zerstört ist. Das Chip arbeitet jedoch fehlerfrei. Wenn es der richtige CPU-Typ ist, kann trotzdem weiterprogrammiert werden.

Alle o.a. Fehler können auch von einem defekten Programmierer herrühren. Eine Reset Beschaltung kann auch die Ursache sein. Dann Push-Pull Reset verwenden.

Falls eine CPU beim programmieren eine **falsche ID** meldet, kann trotzdem mit Löschen, Programmieren etc. fortgefahren werden.

### Bluetooth Interfaces

Viele dieser PC-Treiber emulieren ein virtuelles COMport in dem PC. Wenn die Interface Einstellung der Programmer Software in *Programmer Options* den Automatic Mode vorgibt, werden vor den USB Treibern auch alle COMports gescannt. Das kann zu unerwartet langen Wartezeiten führen bis der Bluetooth Treiber mit einem Timeout zurückkehrt. In diesem Fall sollte die Programmer Port Einstellung auf **USB only** gesetzt werden.

## PackProg

Das Haupt Programm des E-LAB Programmier Systems, das **AVRprog.exe**, kann für alle Arbeiten benutzt werden.:

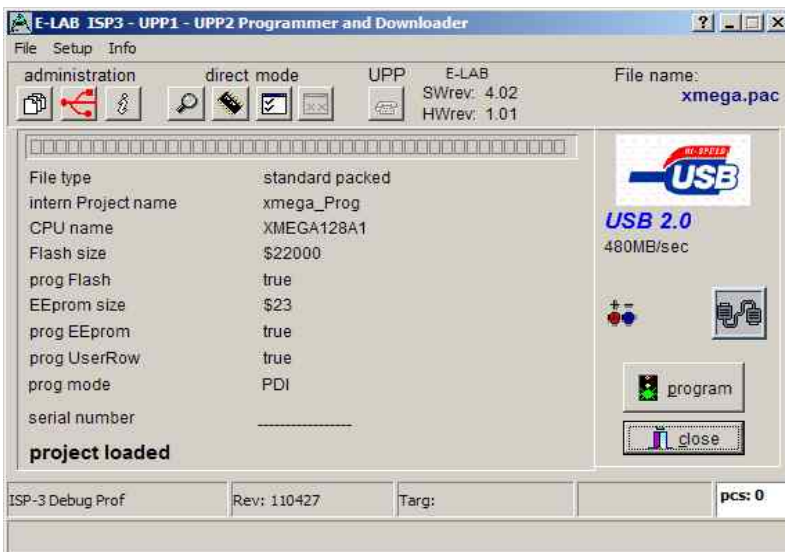
1. für das Erstellen von Projekten mit Fuse und Lockbits, Definition des Flash und EEPROM Files etc
2. für das direkte inCircuit Programmieren von Chips mit allen Programmier Typen
3. für das indirekte inCircuit Programmieren (mittels Flash Karte) durch die UPP Programmier Typen
4. für das Erstellen von gepackten oder encrypteten Projekten
5. für den Download von gepackten Projekten in die Flash Karte der UPP Programmier Typen
6. für das Ablegen von gepackten Projekten in die Flash Karte im Flash Laufwerk des PCs.

Die meisten dieser Funktionen sind für die Produktion und den Service nicht erforderlich, mehr noch, sie stören u.U. hierbei und können zu Fehlern führen.

Deshalb gibt es das reine Programmier Tool **PackProg.exe** für die Programmier Typen ISP3, UPP1 und UPP2. Dieses Tool erlaubt nur das Programmieren von Chips und bei den UPP Typen den Download von gepackten Projekten in den Programmer. Ausserdem kann es als einziges Tool die „deep encrypted“ Projekte verarbeiten. #

## Buttons und Menus

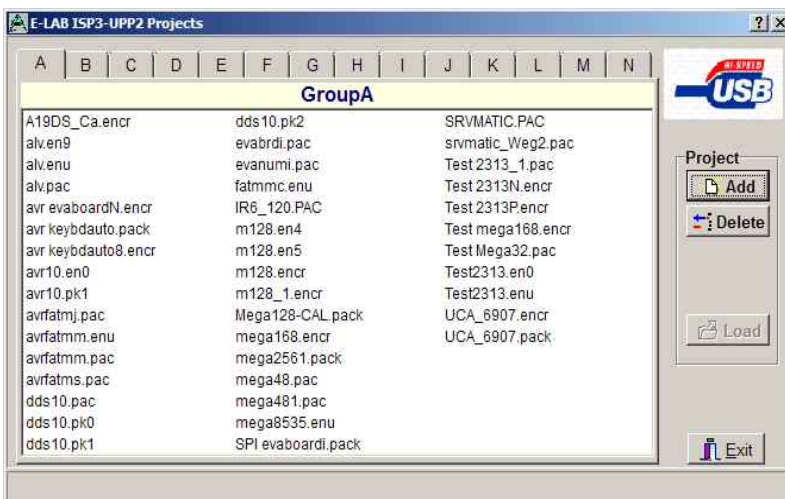
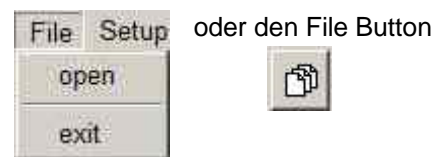
### Projekt Importieren



PackProg arbeitet mit einem Projekt Pool ähnlich wie das Hauptprogramm AVRprog. Es werden hier aber nur gepackte oder encryptete Projekte registriert.

Das bedeutet dass ein neues Projekt zuerst einmal registriert werden muss bevor man mit diesem arbeiten kann.

Neue Projekte werden durch die Projekt Verwaltung importiert, entweder über das File Menu



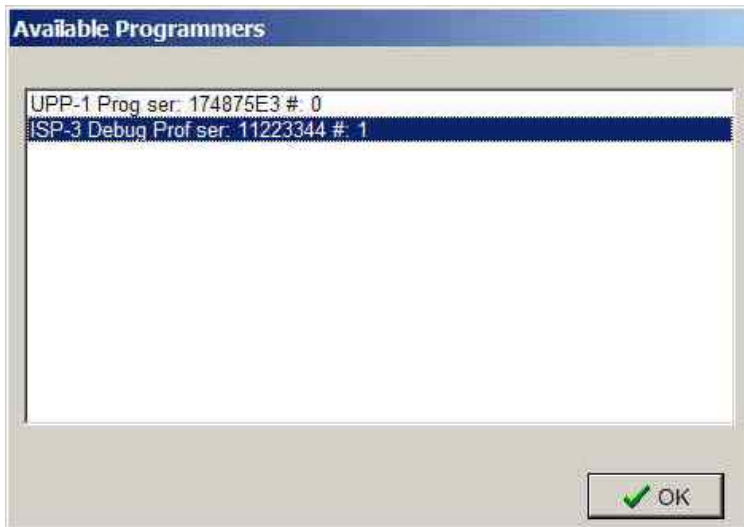
Das eröffnet den Projekt Dialog

Mit dem Add Button kann jetzt ein Projekt dem Pool hinzugefügt werden.

Mit dem Delete Button kann ein vorhandenes Projekt aus dem Pool gelöscht werden.

Mit dem Load Button oder einem Doppel Klick auf einen Eintrag wird das gewählte Projekt geladen.

## Programmer suchen



Um jetzt mit dem gewählten und geladenen Projekt arbeiten zu können muss der Programmer gesucht werden.

Dies geschieht mit dem Check-USB Button.

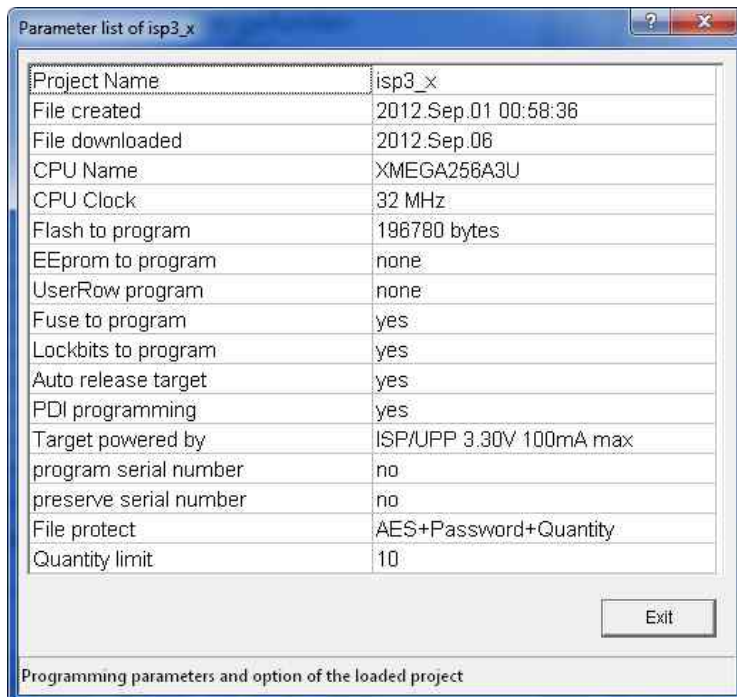


Wird ein Programmer gefunden so wird dies im Hauptfenster angezeigt.

Werden mehrere Programmer gefunden wird dies im nebenstehenden Dialog angezeigt.

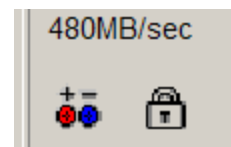
Da immer nur ein Programmer zu einer Zeit gesteuert werden kann, muss einer ausgewählt und bestätigt werden.

## Projekt Prüfen



Mit dem Info Button können die wesentlichen Parameter und Eigenschaften des aktuell im PC geladenen Projekts angezeigt werden.

Zwei Projekt Informationen werden auch im Hauptfenster grafisch angezeigt:



Das Batterie Symbol ist sichtbar wenn der Programmer das Target mit Spannung versorgen muss.

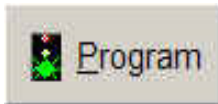
Das Schloss Symbol zeigt an dass der Programmer das Target auslese schützen muss.

## Programmieren im Transparent Modus

Beim Transparent Modus wird der Programmer direkt vom PC aus gesteuert und mit Daten versorgt. Die drei möglichen Operationen sind dabei:



**Device Check.** Mit diesem Button wird der Programmer und das zu programmierende Chip geprüft. Dazu zählen das Testen der Betriebsspannung und das Auslesen und Prüfen der Chip-ID, falls vorhanden.



**Program Chip.** Dieser Button startet die Komplette Programmierung des Chips. Dazu gehören das Flash, das EEprom, die Fusebits, die Lockbits und ggf. eine Serien Nummer. Was im einzelnen ausgeführt wird steht in dem Pack oder Encrypt File (Project).



**Verify Chip.** Verifiziert das Flash nach einem Programmier Vorgang, und falls vorhanden und freigegeben auch das EEprom. Der Vorgang kann natürlich nur durchgeführt werden wenn das Chip nicht auslesegeschützt ist.

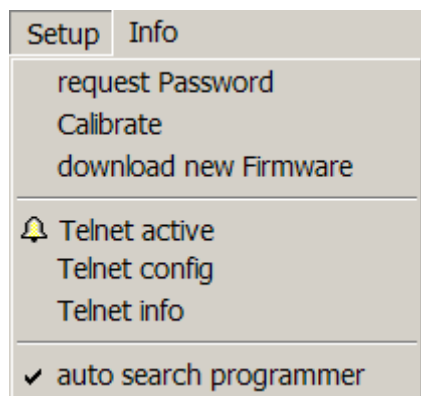
Wurde bei der Projekt Erstellung (Pack oder Encrypted File) die Serien Nummer Programmierung freigegeben so wird die aktuelle Nummer angezeigt. Mit einem Doppelklick auf dieses Feld lässt sich diese hier editieren. Beim Programmieren wird diese Nummer im Flash mit abgespeichert und anschliessend inkrementiert.

Ausnahme: „preserve serial number“, hier wird die im Target vorhandene Nummer verwendet.

prog EEprom	false	prog EEprom	false
prog mode	SPI	prog mode	SPI
serial number	8899AAB2CCDDEE3C	serial number	88 99 AA B2 CC DD EE 3C
<b>Target protected</b>		<b>Target protected</b>	
JPP-1 Prog	Rev: 061211	Targ: 3.28V/0mA	
Infos about the actual loaded project		Infos about the actual loaded project	

## Setup

Zum Erstellen von encrypted oder deep encrypted Projekt Files muss das Passwort des Programmers dem Ersteller der Datei mitgeteilt werden.



Dazu muss im Setup Menu der Dialog **request Password** aufgerufen werden. In diesem Dialog erscheint nun das Passwort des aktuellen Programmers:



Die **Kalibrierung** des Programmers, falls erforderlich, erfolgt in exakt der gleichen Weise wie weiter oben schon beschrieben. Da der Programmer nach dem Einschalten automatisch seine Kalibrierung prüft und ggf. eine Neu-Kalibrierung anfordert ist eine Kalibrierung auch nur in diesem Fall sinnvoll.

Der Download einer neuen **Firmware** erfolgt ebenfalls exakt in der gleichen Weise wie weiter unten beschrieben.

Mit der aktivierten Option **auto search programmer** wird immer nach dem Start des Programms PackProg zuerst das System nach einem angeschlossenen und aktiven ISP3 oder UPP Programmiers abgesehen. Wird ein Programmer gefunden, so wird das sofort hier angezeigt:



## ISP3-X portabler In-Circuit Programmer

serial number	-----
<b>UPP-1 Prog found</b>	
UPP-1 Prog	Rev: 061211
Project: do program EEprom	

### Kommando Zeilen Parameter

Grundsätzlich können bei allen Aufrufen von PackProg auch Schalter mitgegeben werden. Dies sind:

- p Automatischer Programmier Start
  - v Verify target
  - f Verify flash only
  - u Check programmer
  - t Enable Telnet Server
  - c Programm verlassen
  - d Programmer bestimmen, wenn mehrere angeschlossen **-dProgSerNum**
  - s Es werden keine visuellen Fehlermeldungen generiert  
Statt dessen werden die Fehlermeldungen in die Datei „ISP\_UPP.err“ im Programmverzeichnis geschrieben
- “**Filename**“ lädt automatisch das angegebene Pack-File.

Die Reihenfolge der Schalter spielt dabei keine Rolle. Die Schalter müssen durch ein Leerzeichen getrennt sein. Innerhalb der Schalter darf kein Leerzeichen enthalten sein.

Beispiel: **C:\pppp\PackProg.exe "abc.pac" -d12345678 -p -c**

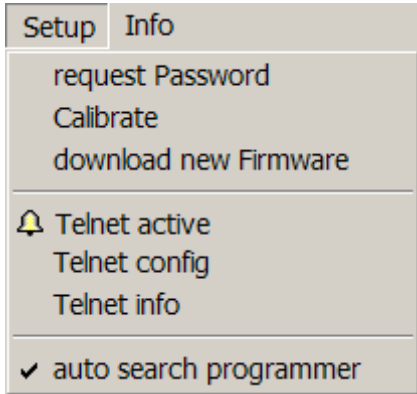
### Return Codes

Folgende Return-Codes können zur Steuerung von Batch-Files benutzt werden:

0	dsOk	Operation erfolgreich beendet
1	dsPwrDown	Target Spannung fehlt
2	dsPwrErr	Target Spannung zu hoch oder zu niedrig
3	dsFalseTyp	Falsche CPU ID gefunden
4	dsProtect	Target ist über Fuses geschützt
5	dsNotEmpty	Target ist nach einem Erase nicht leer
6	dsVerifyErr	Target bzw. Programmer meldet einen Verify Fehler während des Programmierens
7	dsFileError	N.A.
8	dsTimeOutErr	USB Schnittstelle meldet einen Timeout
9	dsCommError	Kommunikations Problem mit dem Programmer
10	dsNoProg	Programmer nicht gefunden
11	dsNoProj	Projekt nicht gefunden
12	dsFwLost	Programmer meldet eine defekte Firmware
13	dsNotfound	File, z.B. Hexfile, wurde nicht gefunden
14	dsCalReq	Programmer meldet Verlust oder illegale Kalibration

## Telnet Interface

Zur einfachen Fernsteuerung des Programmers aus anderen Programmen heraus besitzt **PackProg** ein Telnet Interface. Hiermit können andere Applikationen über Telnet Strings den Programmer komplett fernsteuern. PackProg arbeitet dann als Telnet Server.



Der Menu Punkt schaltet den PackProg Telnet Server ein oder aus.

Der aktive Telnet Server wird im Hauptfenster dargestellt:

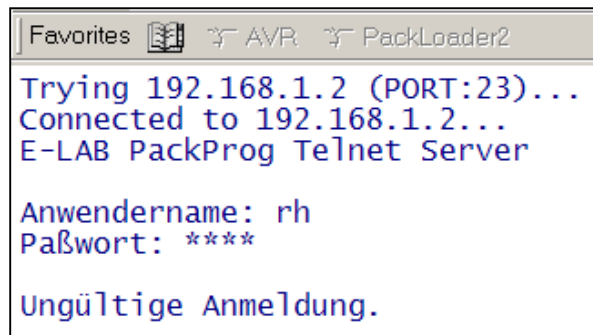


## Telnet Konfiguration

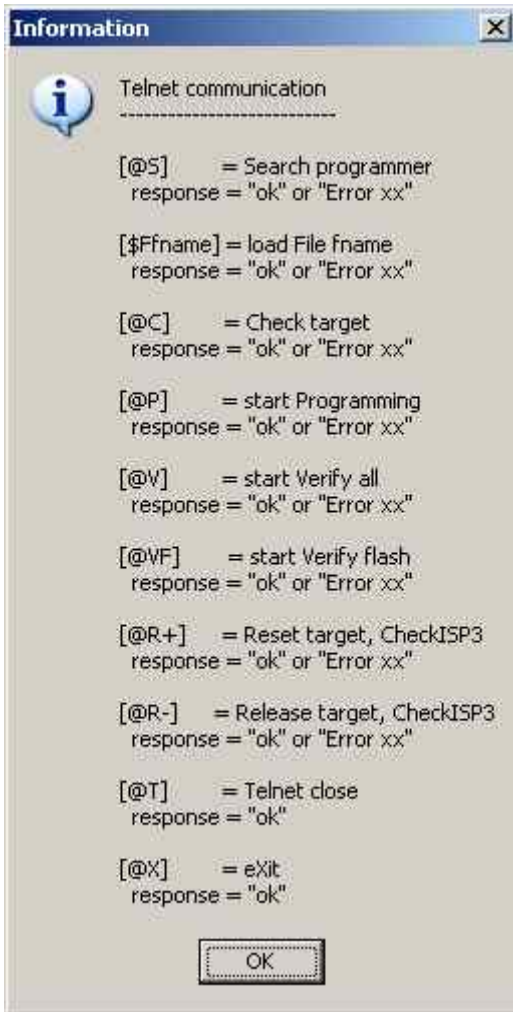
Der Telnet Server benötigt eine USER ID und ein Passwort. Nimmt ein Telnet Client (andere Applikation oder PC) mit dem Server Kontakt auf, dann muss dieser Client sich mit dieser ID und Passwort anmelden. Damit wird vermieden, dass unerwünschte Telnet Clients die Fernsteuerung übernehmen oder beeinflussen können.



Eine ungültige Anmeldung mit einem Telnet Tool sieht so aus:



Mit dem Menu Punkt Telnet Info erhält man eine Kommando Liste.



Mögliche Fehlermeldungen sind

1. PowerDown Target hat keine Spannung
2. Voltage Target Spannung zu hoch oder zu niedrig
3. FalseTyp Target hat die falsche ID oder ID nicht lesbar
4. Protected Bei einem separaten Verify
5. notEmpty Beim Programmieren
6. Verify Verify Error beim Programmieren
7. File File nicht gefunden, Kommando \$F
8. TimeOut Kommunikations Timeout
9. Comm Kommunikations Problem
10. noProg Programmer nicht gefunden
11. noProj Kein Projekt ausgewählt
12. Firmware Programmer hat Firmware verloren



## EXTERNE HARDWARE

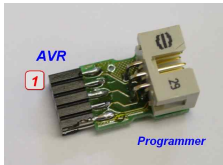
### Schutz des Programmers gegen externe Kurzschlüsse und Überspannung

Alle neueren Programmer Typen sind gegen externen Dauerkurzschluss geschützt. Der Überspannungsschutz ist bei so einem Gerät natürlich nur in Grenzen möglich.

Die Schutzeinrichtung besteht für jede Steuerleitung aus einem Serien Widerstand mit **220Ohm** und einer Überspannungs Schutzdiode mit ca. 6Volt, wobei die Diode zwischen Widerstand und internem CPU-Pin gegen 0Volt liegt. Bedingt durch diese Anordnung kann es zu Problemen beim Programmieren kommen, wenn das Zielsystem eine oder mehrere Leitungen niederohmig abschliesst, d.h. zu stark belastet. Dazu gehören natürlich auch dynamische Lasten wie z.B. Kondensatoren.

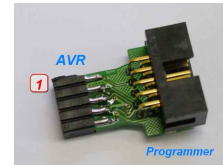
Grundsätzlich sollte beim Schaltungsdesign deshalb besonders darauf geachtet werden. Lasten kleiner 2kOhm und Kondensatoren grösser 100pF sollten nicht an die Programmierleitungen angeschlossen werden. Das gilt für die SPI-Programmierung genauso wie für die JTAG-Programmierung.

## Diverse Adapter



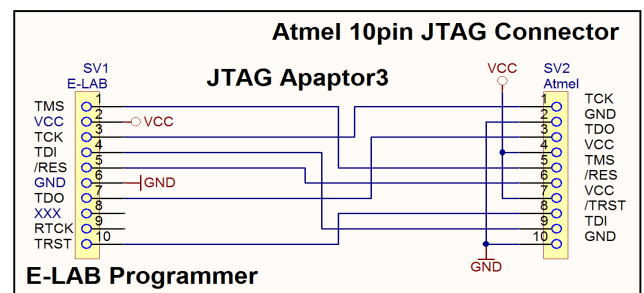
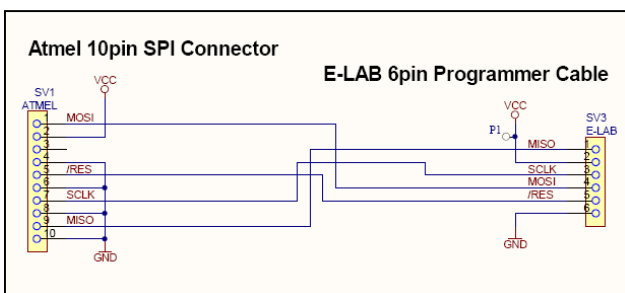
AVR  
1  
Programmer

Als Option ist ein Adapter von E-LAB 6pin SPI auf Atmel 10pin SPI erhältlich #2125



AVR  
1  
Programmer

Als Option ist ein Adapter von E-LAB 10pin JTAG auf Atmel 10pin JTAG erhältlich #2127





## USB TREIBER

### Windows Kompatibilität

Um die USB Programmer mit einem Windows System betreiben zu können ist zumindest Win98SE Voraussetzung. Win95, NT3/4 und Standard Win98 unterstützen kein USB. Bei Win98SE ist ebenfalls Vorsicht geboten. Nicht alle Versionen können USB sauber handhaben, weshalb E-LAB die Funktion der USB Programmer auch unter Win98SE nicht garantieren kann. Für die USB Versionen ist deshalb nur XP, Vista und Windows7 möglich. Es werden die 32bit und 64bit Versionen unterstützt. Grundsätzlich sollten Hard/Software Entwicklung Systeme nur mit XP oder neuer betrieben werden. Diese Systeme sind sehr stabil und haben nicht das Problem der beschränkten Ressourcen bzw. Handles wie alle anderen.

### Treiber Speicherort

E-LAB Programmer mit USB2-Interface brauchen, wie alle nicht-Standard USB-Devices, einen speziellen USB2-Treiber. Dieser Treiber ist Bestandteil der Installation.

Für alle E-LAB USB2-Typen wird ein gemeinsamer, spezieller Treibersatz mitgeliefert, der sich in einem Unterverzeichnis des Installations Verzeichnis befindet:

..\USB2\_Driver\\*. \*                      Treiber für alle USB2 Programmer **ISP3**, **UPP-1** und **UPP-2**

In der **AVRco** Installation befinden sich diese Subdirectories unter dem Verzeichnis

..\AVRco\.

### Treiber Installation

Die notwendige Installation der Treiber auf einem PC ist relativ einfach und geht problemlos vor sich. Mit der Installation des Programmer Pakets wird automatisch der richtige Treiber installiert. Ein manuelles Eingreifen ist nicht erforderlich.

Es ist hierbei ist jedoch ein bestimmtes Verfahren einzuhalten.

1. Programmer vom PC abziehen (wichtig)
2. Computer einschalten und warten bis das System arbeitsbereit ist.
3. Das normale Programmer Install Programm „*ISP\_ICPinst.exe*“ ausführen. (**nicht** für AVRco)
4. Die folgenden Windows Abfragen betreffend des USB Treibers sind jeweils mit ja zu beantworten
5. Den USB Programmer an einen freien USB-Port des PCs anschliessen.
6. Windows bemerkt jetzt ein neues unbekanntes USB Device und registriert dieses.
7. Die Software ist jetzt arbeitsbereit.

## Unsignierte Treiber installieren

Um unsignierte, also nicht von Microsoft zertifizierte, Treiber unter Windows8/64 zu installieren, muss dies vor dem eigentlichen Hochfahren des Systems freigegeben werden mit „cmd shutdown /r /o“. Windows führt dann einen Neustart durch.

Dieses betrifft den AVRco USB Treiber als auch den AVRprog USB Treiber.

### Schritt 1:

Wenn Windows 8 hochgefahren ist, so muss man zuerst einmal in das Boot-Options-Menü starten. Hierzu drückt man im Windows 8 Desktop einmal die Tastenkombination **Win+R**, um den “Ausführen”-Dialog zu öffnen.



### Schritt 2:

Im Options-Menü auf “Problembehandlung” klicken



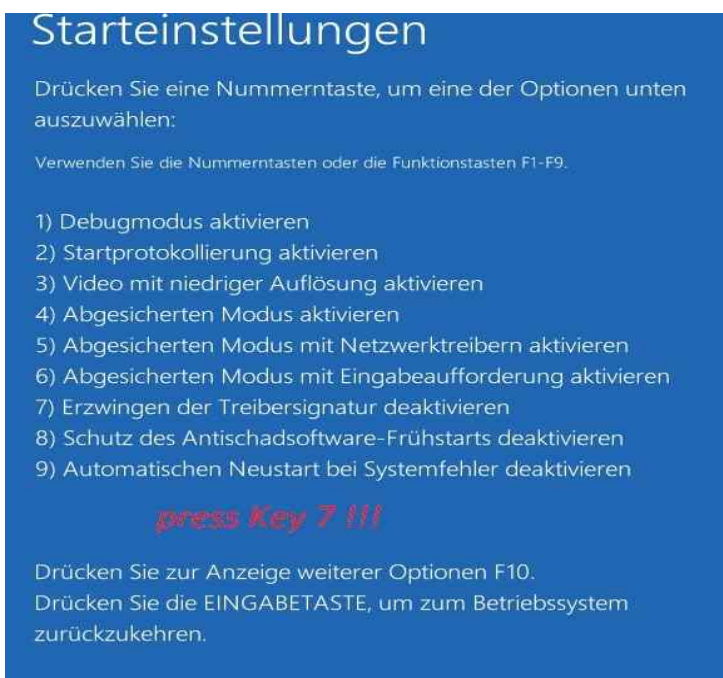
### Schritt 3:

auf “Erweiterte Optionen” klicken



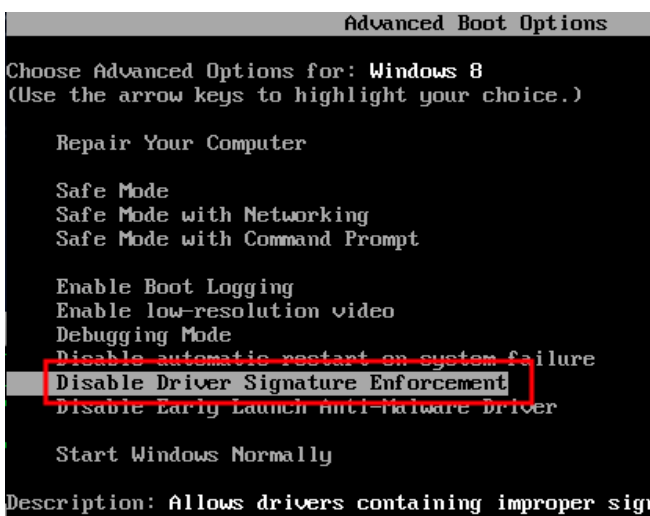
## Schritt 4:

auf  
"Starteinstellungen"  
klicken.



## Schritt 5:

Darin nun die Startauswahl mit der Taste 7 bestätigen.



## Schritt 6:

auswählen von **Disable Driver Signature Enforcement** und mit der Enter Taste Windows neu starten.

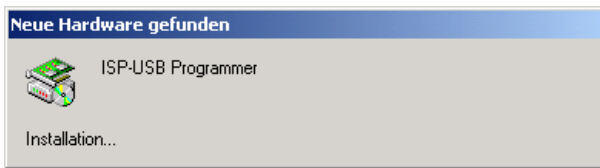
Bitte beachten dass bei WIN8/64 die betreffenden EXEs im WIN7 Mode ausgeführt werden müssen.

Treiber ohne Fehlermeldung installiert werden.  
Beachten Sie, dass die enthaltenen .exe Files im Windows 7 Kompatibilitäts Modus gestartet werden müssen!

Jetzt kann der AVRco bzw. der AVRprog

## Zusätzliche Programmer installieren

Die E-LAB Programmer werden individuell in Windows registriert. Wurde der allererste Programmer registriert kennt Windows jetzt zumindest diese Version (ISP oder UPP) der Programmer.



Weitere Geräte werden in aller Regel kurz mit nebenstehendem Dialog angezeigt und dann automatisch registriert. Je nach System Einstellung des Windows kann dann noch der Pfad für den Treiber abgefragt werden, der dann entsprechend anzugeben ist.

## Programmer Treiber de-installieren

Im allgemeinen macht es wenig Sinn, die Treiber für die E-LAB Programmer zu de-installieren, da diese nur bei Bedarf temporär geladen werden und damit das System nicht belasten. Sollte es trotzdem notwendig sein, die USB Treiber zu de-installieren, darf das nicht manuell geschehen.

Derjenige Treiber, der de-installiert werden soll, dessen Programmer muss angeschlossen sein. Dann wird der Geräte Manager von Windows aufgerufen und mit dessen Hilfe wird der Treiber de-installiert. Andere Windows Utilities bieten ein sogenanntes Deaktivieren von Treibern an, welches aber mit einer De-Installation nichts zu tun hat.

## STROMVERSORGUNG DES ZIELSYSTEMS

Alle USB Programmer können die Target CPU mit einer voreinstellbaren Spannung versorgen.

Da ein USB-Port des PC auch mindestens 100mA/5Volt= liefert, kann diese Versorgung benutzt werden. Hier ist jedoch zu beachten, dass sehr oft diese nominellen 5V gar keine 5V sind, sondern z.B. 4.6V oder noch weniger. Bedingt durch den im Gerät vorhandenen programmierbaren Spannungsregler ist ein weiterer Spannungsverlust von ca. 0.2Volt zu erwarten. Um trotzdem  $\geq 5V$  am Ausgang zu erreichen haben die Programmer ein step-up Spannungskonverter eingebaut, so dass dies kein Problem sein sollte.

Weiterhin ist zu beachten, dass ein USB HUB max. 100mA Strom liefert, wenn er nicht ein eigenes Netzteil hat. Daher kann der max. entnehmbare Strom hier enorm reduziert werden. Wird der Programmer direkt an den PC angeschlossen, kann der Programmer bis zu 300mA liefern.

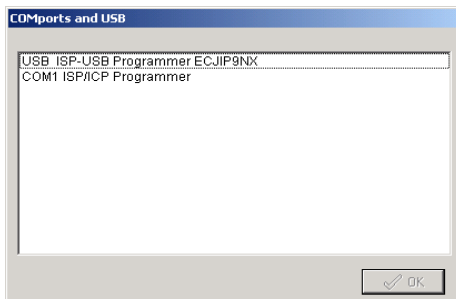
01	20
03	40
05	60
07	80
09	100

Der Programmier Anschluss der ISP3-X Geräte bietet auch einen extra Pin für externe Strom Versorgung. Dieser kann mit einem Netzteil verbunden werden mit ca. 5..9V/DC 500mA.

Pin6 Ground

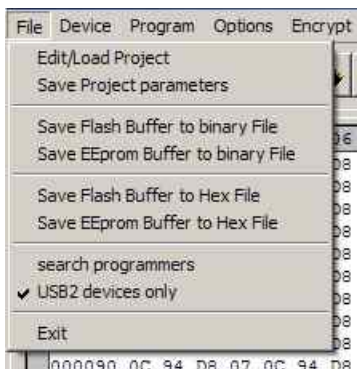
Pin10 kann als externer positiver Strom Versorgungs Eingang verwendet werden.

## MULTIPLE PROGRAMMER



Werden mehrere USB Programmer gefunden oder zumindest ein serieller und ein USB Typ, so wird der nebenstehende Auswahl Dialog angezeigt, bevor man mit dem System arbeiten kann. Grundsätzlich wird aber nur max. ein serieller Programmer gesucht, weitere werden ignoriert.

Weiterhin kann immer nur mit einem Programmer gleichzeitig gearbeitet werden. Mehrfach (Gang) programmieren ist nicht möglich.



Werden Programmer hinzugefügt oder abgezogen, ist es möglich sich mit dem Menu Punkt *File/search programmers* zu jeder Zeit einen Überblick über die angeschlossenen Geräte zu verschaffen.

## FIRMWARE UPDATE

Neuere Programmier Geräte Typen unterstützen das Downloaden einer neuen Firmware in den Programmer. Ob das vorliegende Gerät dies auch kann, erfahren Sie, indem Sie den **Device Status** Dialog aufrufen. Hier erscheint dann in der Zeile Update eine Zahl, die das aktuelle Firmware Datum wiedergibt, was auch ein Teil des Datei Namens darstellt.

Bei einem Firmware update des Programmers wird, abgesehen von einem kleinen sog. Bootsektor, dessen komplettes Programm gelöscht und mit einem neuen Programm beschrieben. Neue Firmware für die einzelnen Programmer können von [www.e-lab.de](http://www.e-lab.de) herunter geladen werden. Diese Programme sind nicht auf dem PC ausführbar, sondern müssen in das Programmiergerät geladen werden.

ISP3-X Standard Datei: [yymmdd\\_ISP3\\_X.pupd](#) oder vom WEB in Datei: [ISP3X\\_update.zip](#)

ISP3-X ICE Std Datei: [yymmdd\\_ISP3X\\_Deb\\_Std.pupd](#) oder vom WEB in Datei: [ISP3X\\_Deb\\_Std.zip](#)

ISP3-X ICE Prof Datei: [yymmdd\\_ISP3X\\_Deb\\_Prof.pupd](#) oder vom WEB in Datei: [ISP3X\\_Deb\\_Prof.zip](#)

Dies geschieht mit Hilfe des Menu-Punkts **Options/Download new Firmware**. Beachten Sie dazu bitte, dass alle Firmware Dateien sich im Ordner **ISP\_Updates** unterhalb der Home Directory des Programmers befinden müssen, ansonsten werden sie nicht gefunden. Mit obigen Menu wird unten stehender Dialog geöffnet.



Die Gruppe „File state“ enthält den Status der geladenen Datei, wie z.B. Dateinamen, Dateigröße und Infos über die Ziel-CPU.

Die Gruppe „Target state“ enthält den Status des Programmers.

Wenn die Ampel grün ist, kann mit dem Herunterladen bzw. Programmieren begonnen werden. Der senkrechte Balken zeigt den Fortschritt beim Download an.



Mit dem „Open“ Button wird ein Dialog geöffnet, der alle in Frage kommenden Firmware Updates anzeigt. Die Datei Namen beginnen mit dem Erstellungsdatum und sind deshalb gut zu identifizieren. Ein Doppelklick lädt die Datei in den Download Buffer.



Mit dem Comport Button wird die Verbindung mit dem Downloader im Programmiergerät aufgenommen. Der Downloader überprüft, ob die Datei zu dem Programmiergerät passt. Wenn das nicht der Fall ist, erfolgt eine Fehlermeldung.



Wenn die Datei erfolgreich geladen wurde und zu dem Programmer passt, kann mit dem DownLoad Button das Programmieren des Programmers gestartet werden. Dieser Vorgang kann in dem rechten Bargraph beobachtet werden.



Sollte der Download einmal „hängen“ bleiben, so kann mit dem Stop Button der Vorgang abgebrochen werden. Bei Problemen evtl. die externe Spannung des ISP erhöhen oder erniedrigen.



## ISP3-X portabler In-Circuit Programmer

Nach erfolgreichem Update kann eine evtl. auftretende USB Fehlermeldung von Windows ignoriert werden. Auf jeden Fall muss der Programmer aber vom PC einmal abgezogen werden.

Eine fehlerhafte oder unvollständige Programmierung wird beim Power-On des Gerätes festgestellt. Der Programmer fordert dann einen neuen Download an.

Ein **Firmware Download** kann auch mit dem RESET Taster erzwungen werden. Diese RESET Taste befindet auf der hinteren Frontseite des Gehäuses.



1. USB Buchse
2. Micro-SD Karten Schacht (nicht bei ISP3)
3. RESET Taste

Diese Taste hat zwei Funktionen:

1. ein kurzer Druck auf diese Taste führt einen **Hardware Reset** aus.
2. ein längerer Druck auf diese Taste (>1sec) führt ebenfalls ein **Hardware Reset** aus mit anschliessendem Einsprung in den **Zwangs Download** Mode wie oben beschrieben.



---

## ANHANG

### XMega

**Achtung:**

Die RESET Leitung auf dem Target darf nicht mit Kondensatoren belastet sein, da hier der PDI-CLK mit mehr als 1MHz darüber geht. Ein Jumper der den RESET Kondensator beim programmieren abklemmt hilft hier.

XMega PDI speed

high speed

low speed

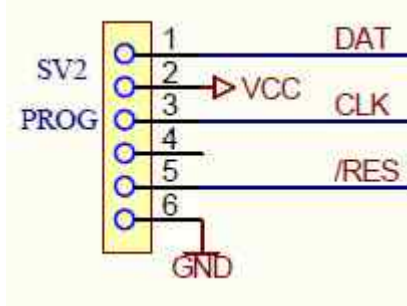
Bei älteren XMegas, z.B. XMega128A1, oder langen Programmierleitungen kann es zu Fehlern kommen. Atmel empfiehlt hierbei die Clockrate des PDI Interfaces herabzusetzen. Dies geschieht unter "Options/Programmer options"

Ein spezieller XMega Adapter ist beim ISP3-X nicht erforderlich.

## TINY4-5-9-10-20

Diese Tins werden mit 5V über drei Pins programmiert.

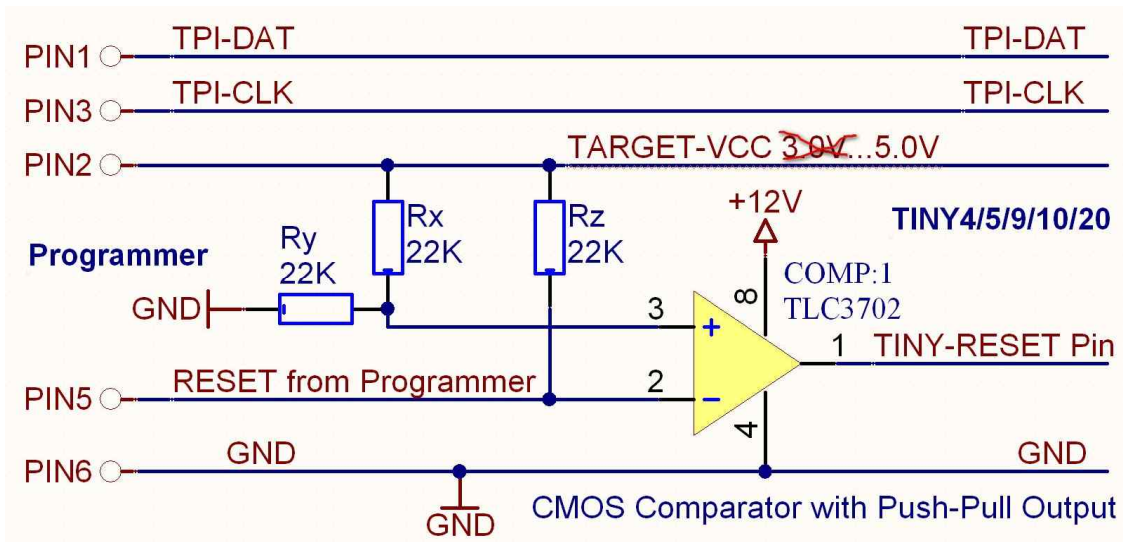
Der Programmier Mode nennt sich TPI. Da der Programmier Stecker auch für den SPI Mode verwendet wird, weicht dieser von der Atmel Belegung ab. Die Last am TPI\_DAT Pin sollte nicht kleiner als 80kOhm sein.



Hier die Verbindung vom Programmier zu einem TPI TINY.

Weil der RESET Pin dieser TINYS durch eine Fuse abgeschaltet werden kann, ist ein normal re-programming dieser Chips normalerweise nicht möglich. Dazu muss man in den high-voltage programmier Mode schalten. Glücklicherweise ist dabei der Programmier-Algorithmus derselbe mit der Ausnahme dass an den RESET Pin 12V angelegt werden müssen.

Hier ist eine einfache Schaltung wobei der RESET Ausgang des Programmier die 12V schaltet:

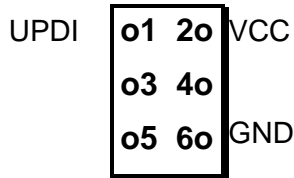


## UPDI Tiny und UPDI Mega

### Basics

Das neueste Atmel Programmier Interface ist eine Ein-Draht (one-wire) Version die nur einen Pin der Chips belegt. Dieses Interface ist sehr schnell, benötigt aber einen speziellen Algorithmus. Dieser wird in den X-Versionen der E-LAB Programmer unterstützt.

Die Belegung des Programmiersteckers ist untenstehend. Hierbei ist zu beachten dass die UPDI Leitung nicht nieder-ohmig belastet werden darf.



**TopView TINY UPDI Stecker**

Ein gewisser Nachteil dieses single-wire Systems ist dass der UPDI Pin der Chips über Fuses zum normalen IO-Pin programmiert werden kann. Das hat zur Folge dass so ein Chip dann nicht mehr auf normalem Weg neu programmiert werden kann! Das trifft allerdings für die UPDI Megas nicht zu.

In diesem Fall kann das nur mit einem 12V Impuls am UPDI Pin wieder zurückgesetzt werden. Dazu ist allerdings ein spezieller Algorithmus nötig. Dieser wird in den Programmern auch unterstützt. Da hier alles zusammen passen muss, timings und Spannungsverläufe, ist eine spezielle externe Hardware nötig.

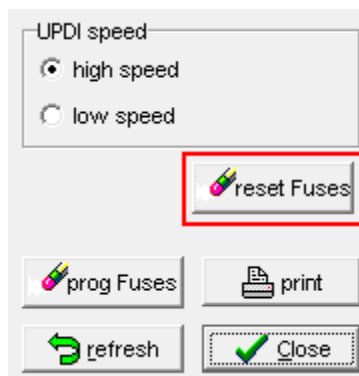
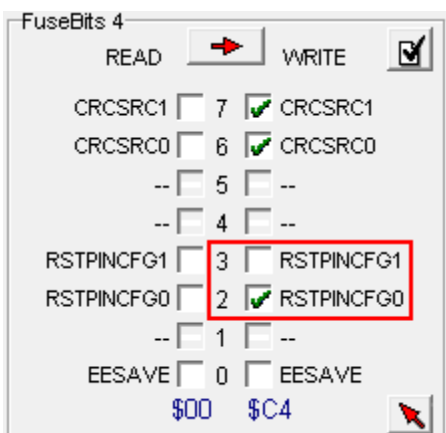
Dieser HVP Adapter ist von E-LAB erhältlich. Er braucht keine weitere Spannungs Versorgung.



Ist das 12V (HVP) Reset notwendig dann kann mit angeschlossenem HVP Adapter diese in Frage kommende Fuse zurückgesetzt werden.

Dazu ist allerdings zu beachten dass am UPDI des Chips evtl . angeschlossene Bauteile auch 12V tolerant sein müssen und keine grosse Last angeschlossen ist.

Mit untenstehendem Dialog im AVRprog.exe kann dieser RESET durchgeführt werden.



Dieser Button ist nur sichtbar wenn mit UPDI programmiert wird und der HVP Adapter gefunden wird.



## Chipcon

### Allgemeines

Die aktuelle ISP3-X Firmware unterstützt auch die in-Circuit programmierbaren ZigBee Chip Familie CC1110, CC2510 und CC2430 von TI-Chipcon. .

Beim Erstellen eines neuen Projektes für den CC2430 muss neben dem Flash Hexfile auch die CPU ausgewählt werden (CC2430-F32, CC2430-F64 oder CC2430-F128).

Die Ausgangsspannung für das Target kann hier zwischen 2.7V und 3.6V eingestellt werden, wenn der Programmer das Target versorgen soll. Der korrekte CPU Clock muss vorgegeben werden, entweder 16MHz oder 32MHz.

Fusebits existieren keine aber ein Lockbit Block. Die genaue Bedeutung der Lockbits muss dem Datenblatt des CC2430 entnommen werden.

### IEEE Adresse



Da der CC2430 ein ZigBee Device ist muss der Controller eine feste Adresse haben (IEEE), vergleichbar mit einer MAC Adresse bei Ethernet. Chipcon hat festgelegt dass diese Adresse (8 Bytes) auf den letzten 8 Bytes des Flash liegen soll. Beim F128 wäre dies \$1FFF8..\$1FFFF.

Das MSB liegt dabei auf der unteren Adresse, LSB auf der obersten.

Die Programmiersoftware unterstützt dies indem diese Adresse vorgegeben werden kann und die Programmier Software diese nach jedem Programmier Durchgang automatisch inkrementiert.

Mit **Preserve Serial Number** wird die im Target vorhandene Serien Nummer wieder verwendet, vorausgesetzt das Chip ist noch auslesbar.

Mit **LSB Serial Number Order** wird Serien Nummer beginnend mit dem LSB aufsteigend im Flash abgelegt, ansonsten beginnend mit dem MSB.

Diese Optionen können im Dialog **Options/Target Options** vorgegeben und auch editiert werden.

### Belegung des Programmiersteckers

Es wird das beiliegende 6polige Kabel verwendet. Die Signal Belegung des Kabelsteckers bzw. der Buchse auf dem Target System ist wie untenstehend vorzunehmen.

N.C	01 20	VCC
(P2_2) CLK	03 40	DATA (P2_1)
/RESET	05 60	GND

TopView Buchse auf dem Target

#### Achtung!

Der PIN2 muss mit der Target Versorgung (VCC) verbunden werden und PIN6 mit Ground des Targets. PIN5 muss an den Reset PIN, PIN3 an den Debug Clock (P2\_2) und PIN4 an den DATA PIN (P2\_1) des CC2430 angeschlossen werden.

## Verify

Während des Programmiervorgangs der in 1kB Blöcken erfolgt, wird nach jedem Block automatisch ein Verify dieses Blockes durchgeführt und im Fehlerfall mit einer entspr. Meldung abgebrochen.

## Chip interne ID



Diese ID Nummer wird aus dem Chip ausgelesen und hier dargestellt. Das erste Byte ist immer Null und ohne Bedeutung. Das zweite repräsentiert die CPU Revision (\$01) und das dritte Byte die Chip-ID, \$85 für die CC2430 Familie.

### **Achtung**

Eine falsche Chip-ID kann diverse Ursachen haben, defekter Programmer, defektes Kabel, schlechte oder falsche Kontaktierung zum Target oder auch ein defektes Chip.

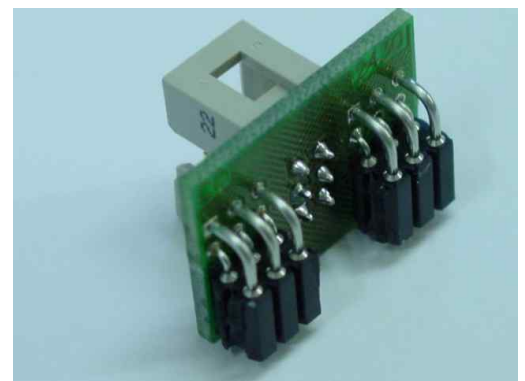
Eine falsche ID sollte niemals ignoriert werden sondern der Grund dafür muss eingehend untersucht werden.

## Chipcon Evaluation Boards

### CC2430DB

Dieses Board hat zwei 6pol Pfosten, die es erlauben einen externen Programmer anzuschliessen. Da der ISP3-X Programmer deshalb nicht direkt auf dieses Board gesteckt werden kann, gibt es als optionales Zubehör für den ISP3-X ein kleines Adapter Board das separat bestellt werden muss. (ord #2109)

Dieser Adapter ist wie untenstehend zu sehen auf das Evaluation Board zu stecken.



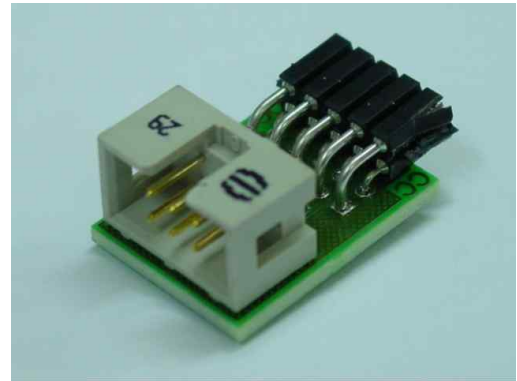


## ISP3-X portabler In-Circuit Programmer

### Chipcon Evaluation Board SOC-BB

Dieses Board hat einen 10pol Pfosten für den Anschluss eines externen Programmers. Da der ISP3-X Programmer nicht direkt auf dieses Board gesteckt werden kann, gibt es als optionales Zubehör für den ISP3-X das kleine Adapter Board **SOC-BB Adaptor** das separat bestellt werden muss. (ord #2123)

Dieser Adapter ist wie untenstehend zu sehen auf das Evaluation Board zu stecken.



## AT89LPxx

### Allgemeines

Die aktuelle ISP3-X Firmware unterstützt auch die in-Circuit programmierbaren AT89LP2052 und 4052 von Atmel. Der Unterschied zu den AVR's und 89Sxx Typen besteht in der Programmier Beschaltung der CPUs beim in-circuit Programmieren.

### Belegung des Programmiersteckers

Es wird das beiliegende 10polige Kabel verwendet. Die Signal Belegung des Kabelsteckers bzw. der Buchse auf dem Target System ist wie untenstehend vorzunehmen.

MISO	<b>01 20</b>	VCC
SCLK	<b>03 40</b>	MOSI
/RESET	<b>05 60</b>	GND
SS	<b>07 80</b>	N.C.
N.C.	<b>09100</b>	N.C.

#### Achtung!

Der PIN2 muss mit der Target Versorgung (VCC) verbunden werden und PIN6 mit Ground des Targets. PIN5 muss an den Reset pin, PIN3 an den SCLK, PIN4 an den MOSI pin, PIN1 an den MiSO pin und PIN7 an den SS PIN angeschlossen werden.

**TopView Buchse auf dem Target**



## Serial Flash (SPI-Flash)

### Allgemeines

Die aktuelle ISP3-X Firmware unterstützt auch die in-Circuit programmierbaren SPI-Flash Bausteine der Typen AT25DFxxx, S25FLxxx und SST25VFxxx.

### Belegung des Programmiersteckers

Es wird das beiliegende 6polige Kabel verwendet. Die Signal Belegung des Kabelsteckers bzw. der Buchse auf dem Target System ist wie untenstehend vorzunehmen.

MISO/SO	<b>o1 2o</b>	VCC (3.3V)
SCLK	<b>o3 4o</b>	MOSI/SI
/CS-SELECT	<b>o5 6o</b>	GND

### TopView Buchse auf dem Target



## Weitere E-LAB Programmer



### UPP1-X

Der Programmer UPP1-X kann sowohl vom PC aus gesteuert werden als komplett PC unabhängig (portabel). Im portable Mode werden die Projekte auf seine microSD Karte gespeichert.

Im portablen Mode wird das Gerät durch einen internen Li-Po Akku gespeist.

Der Anschluss an den PC erfolgt über USB2.



### UPP1-XS

Der Programmer UPP1-XS kann sowohl vom PC aus gesteuert werden als komplett PC unabhängig (portabel). Im portable Mode werden bis zu 10 Projekte auf seine microSD Karte gespeichert.

Die Projekt Auswahl erfolgt über die Up/Down Tasten mit Hilfe des LED Displays. Der aktuelle Projekt Name kann über das Display in Laufschrift angezeigt werden.

Im portablen Mode wird das Gerät durch einen internen Li-Po Akku gespeist.

Der Anschluss an den PC erfolgt über USB2.



### UPP2-X

Das Flaggschiff unter den portablen E-LAB Programmern.

Auf der Flashkarte können bis zu 64 Projekte abgelegt werden.

Der Anschluss an den PC erfolgt über USB2.

Im portablen Mode wird das Gerät durch einen internen Li-Po Akku gespeist.

Hervorragend für den Service und auch die Kleinserien Produktion geeignet.



## UPP1-P und UPP1-PR

Die Produktions Ausführung des UPP1-X.  
Im Hutschienen Gehäuse.  
Steuerung vom PC über USB oder  
**Fernsteuerung über DLL**  
Version P oder Version PR





# Notizen

---

©1998-2018 **E-LAB Computers**  
Grombacherstr. 27  
D74906 Bad Rappenau

Tel. 07268/9124-0  
Fax. 07268/9124-24

Internet: [www.e-lab.de](http://www.e-lab.de)  
e-mail: [info@e-lab.de](mailto:info@e-lab.de)

---