



Benutzer Handbuch UPP2-X Programmer

# UPP2-X mit AES encryption

## Portable Programmer

© Copyright 1998-2018 by E-LAB Computers

The screenshot shows the E-LAB ICP-ISP Programmer software interface. The window title is "E-LAB ICP-ISP Programmer [AVR] [avr fatmmc]". The menu bar includes File, Device, Program, Options, Encrypt, and Help. The toolbar contains icons for file operations, a lightbulb, a red stop sign, a magnifying glass, a USB icon, a chip icon, a lock icon, a checkmark icon, and a stop sign icon. The status bar shows "Action: none" and "2.9".

The main display area is divided into two sections:

**AVR Memory:**

adress	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	ascii
000000	0C	94	48	3C	0C	94	1E	45	0C	94	1E	45	0C	94	1E	45	..H<...E...E...E
000010	0C	94	1E	45	0C	94	1E	45	0C	94	1E	45	0C	94	1E	45	...E...E...E...E
000020	0C	94	1E	45	0C	94	1E	45	0C	94	1E	45	0C	94	1E	45	...E...E...E...E
000030	0C	94	1E	45	0C	94	1E	45	0C	94	1E	45	0C	94	1E	45	...E...E...E...E
000040	0C	94	AA	3C	0C	94	1E	45	0C	94	EF	3C	0C	94	C6	3C	...<...E...<...<
000050	0C	94	1E	45	0C	94	1E	45	0C	94	1E	45	0C	94	1E	45	...E...E...E...E
000060	0C	94	1E	45	0C	94	1E	45	0C	94	1E	45	0C	94	1E	45	...E...E...E...E
000070	0C	94	1E	45	0C	94	1E	45	0C	94	1E	45	0C	94	1E	45	...E...E...E...E
000080	0C	94	1E	45	0C	94	1E	45	0C	94	1E	45	0E	94	0A	40	...E...E...E...@

**SPI Memory:**

adress	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	ascii
000000	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	.....
000010	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	.....
000020	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	.....
000030	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	.....
000040	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	.....
000050	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	.....
000060	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	.....
000070	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	.....
000080	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	.....

At the bottom of the window, the status bar displays: CPU : MEGA128, Clock : 15000000 Hz, 5.08Volt, UPP-USB Programmer.

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



## Inhaltsverzeichnis

<b>EINLEITUNG</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 Spannungsversorgung.....	8
Aufruf Optionen und Command Line Parameter.....	8
Knopf-Leiste .....	10
Funktionen der Buttons und Menues .....	10
AES PAC Files .....	24
Interaktives Programmieren .....	25
Erstellen von Projekten.....	25
Programmieren von SD Karte .....	26
Status, Fehleranzeige und Probleme .....	27
<b>PackProg</b> .....	<b>28</b>
Buttons und Menues.....	28
Kommando Zeilen Parameter .....	33
Return Codes .....	33
Telnet Interface .....	34
Arbeiten Stand Alone.....	36
<b>EXTERNE HARDWARE</b> .....	<b>40</b>
Schutz des Programmers.....	40
Verschiedenen Adapter .....	40
<b>MULTIPLE PROGRAMMER</b> .....	<b>41</b>
<b>BATTERIE BETRIEB</b> .....	<b>41</b>
<b>TARGET POWER SUPPLY</b> .....	<b>41</b>
<b>USB TREIBER</b> .....	<b>42</b>
Unsignierte Treiber installieren.....	43
<b>FIRMWARE UPDATE</b> .....	<b>46</b>
<b>ANHANG</b> .....	<b>48</b>
<b>XMega</b> .....	<b>48</b>
<b>TINY4-5-9-10-20</b> .....	<b>48</b>
<b>UPDI Tiny und UPDI Mega</b> .....	<b>49</b>
<b>Chipcon</b> .....	<b>50</b>
Chipcon Evaluation Boards .....	52
<b>AT89LPxx</b> .....	<b>53</b>
Serial Flash (SPI-Flash).....	54
<b>WEITERE E-LAB PROGRAMMIERGERÄTE</b> .....	<b>55</b>

---

## EINLEITUNG

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 ist die fast beliebige Reprogrammierbarkeit der CPU's .

Der vorliegende UPP-Programmer zeichnet sich durch kompakte Grösse, komfortable Software und einfache Handhabung aus. Dieses Handbuch gilt nur für die **UPP2-X** Programmer Version

---

## EIGENSCHAFTEN

- Anschluss an den PC über die USB Schnittstelle: USB2, USB1 oder Hub
- Kein Netzteil erforderlich. Gerät wird von dem internen Akku, der Schnittstelle des PC's (USB) oder vom Target versorgt.
- Automatische Anpassung an die Spannung der Ziel-CPU (1.8-5.5Volt)
- Strom Aufnahme 50..70mA wenn Speisung durch das Target System erfolgt
- Komfortable und umfangreiche Software. Bis zu 63 Projekte können auf der internen SD Karte gespeichert werden
- Software läuft unter XP, Vista und Windows7,8 / 32 und 64bit
- Kleines und handliches Gerät ca. 100x55x20mm
- Unterstützt alle **SPI, JTAG, PDI, TPI** und **UPDI** programmierbaren AVR's
- Unterstützt alle SPI programmierbare AT89Sxx Typen
- Unterstützt die Atmel AT89LPxx Familie
- Unterstützt die ChipCon CC1110, CC2510 und CC2430 Familie
- Unterstützt die Serielle Flash (SPI-Flash) Familien AT25DFxxx, S25FLxxx und SST25VFxxx
- Programmierbare Ausgangsspannung (Speisung) für das Zielsystem 30mA..300mA
- Extrem schnell - programmiert einen Mega128 in **3sec** (JTAG) und einen Tiny44 in **1sec** (16MHz)
- Selbst Update mit neuer Firmware vom PC aus.
- Hervorragend für Reparatur und Service Programmierung geeignet.
- unterstützt auch AES verschlüsselte Projekt Files ((PAC Files)
- unterstützt mit AES-PAC files eine Limitierung der möglichen Programmierzyklen
- Umfangreiches und komfortables grafisches Menu System auf dem Programmer Display. Anzeige von Projekten, Directories (Folders), Projekt Eigenschaften und aktuelle Target Parameter.
- micro-SD Karte enthalten. FAT16 bis 2GB und FAT32 bis 32GB werden unterstützt.
- Interner **Li-PO Akku** mit 700mAh, optional mit 1500mAh
- Externes Schaltnetzteil mit 5Volt/1000mA enthalten
- Auto Adapter für Zigarettenanzünder vorhanden
- Option Adapter von Atmel 6pol SPI zu Atmel 10pol Programmierstecker (nicht enthalten)
- Option Adapter von E-LAB JTAG zu Atmel JTAG Programmierstecker (nicht enthalten)
- Option Adapter von UPP2-X auf ChipCon CC2430 Evaluation Board (nicht enthalten)
- Option Adapter von UPP2-X auf ChipCon SOC\_BB Evaluation Board (nicht enthalten)

### Achtung

Neue Flash Karten sollten vor dem ersten Gebrauch unbedingt im PC formatiert werden. Damit wird sicher gestellt, dass diese auch dann wirklich im PC und im Programmer gelesen werden können.

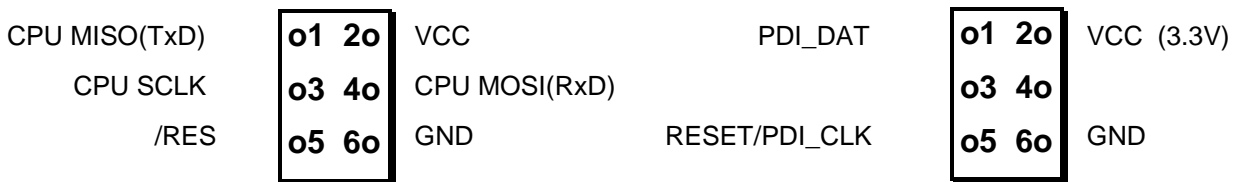
## ANSCHLÜSSE

XP, Vista oder Windows7/8 ist Voraussetzung. Ein USB-1 oder USB2 Port ist erforderlich. 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.**

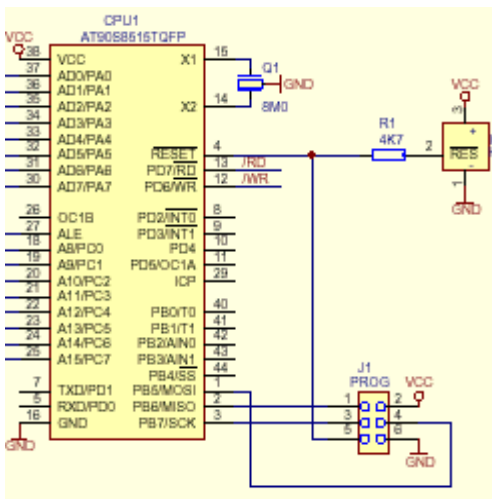
Die interne Spannung beträgt ca. 3.3Volt. Es dürfen keine niederohmigen Lasten an den Programmierpins der CPU angeschlossen sein. Kondensatoren sind ebenfalls nicht erlaubt, am Reset-Pin max. 100nF!! Beim X Mega darf kein Kondensator am Reset angeschlossen sein. Pullup minimal 100kOhm.

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:



TopView AVR SPI Stecker

TopView XMEGA PDI Stecker

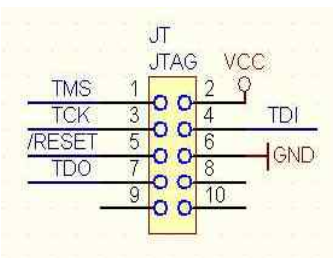


Der Pin1 der 6poligen Buchse des Programmers ist durch ein kleines Dreieck auf der Vorderseite markiert. Ein **verpoltes** Einstecken der Buchse des Programmers auf die Pfostenleiste des Targets **kann zur Zerstörung** des Programmierers führen!

Die zulässige Betriebsspannung des Targets während des Programmierens muss zwischen 1.8 und 5.5 Volt liegen. Bei Spannungen unter 3.3V muss der UPP durch den PC oder externes Netzteil versorgt werden.

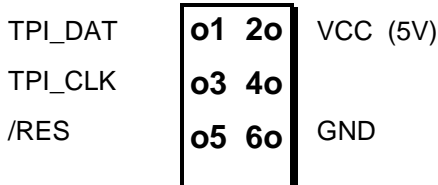
Obwohl diese Leitungen gegen Überlast geschützt sind, können andauernde **Kurzschlüsse** auf mehreren der 4 Steuerepins zur Zerstörung des Gerätes bzw. seiner internen AVR CPU führen. Deshalb dürfen nur elektrisch einwandfreie Boards an den Programmer angeschlossen werden.

### JTAG Programmierung



Zur JTAG Programmierung der Target CPU muss das 6-polige Kabel gegen ein 10-poliges ausgetauscht werden. Da der Programmer 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 sollte.



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

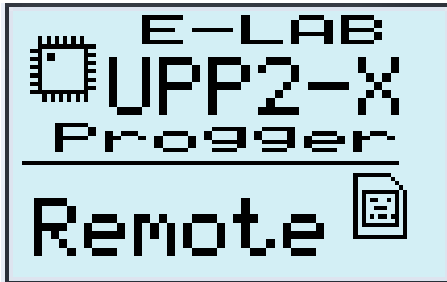
Diese Tinsys werden mit 5V über drei Pins programmiert. Der Programmier Mode nennt sich TPI. Da der Programmier Stecker auch für den SPI Mode verwendet wird, weicht dieser von der Atmel Belegung ab.

Die Last am TPI\_DAT Pin darf nicht kleiner als 80kOhm sein.

## SOFTWARE

Zur Steuerung des UPP2-X Programmers sind 2 Programme vorgesehen. Das erste ist **AVRProg.exe**, welches das umfangreichere ist. Es kann zum Erstellen einer Projekts, zum Ändern der Daten und der Fuse Bits, Lock Bits und der Programmer Setup Optionen benutzt werden. Ebenso kann die Stromversorgung eingestellt und das Zielsystem verifiziert werden oder es können gepackte und verschlüsselte Programmier Files (zur Weitergabe an ausführende Dienstleister) erstellt werden.

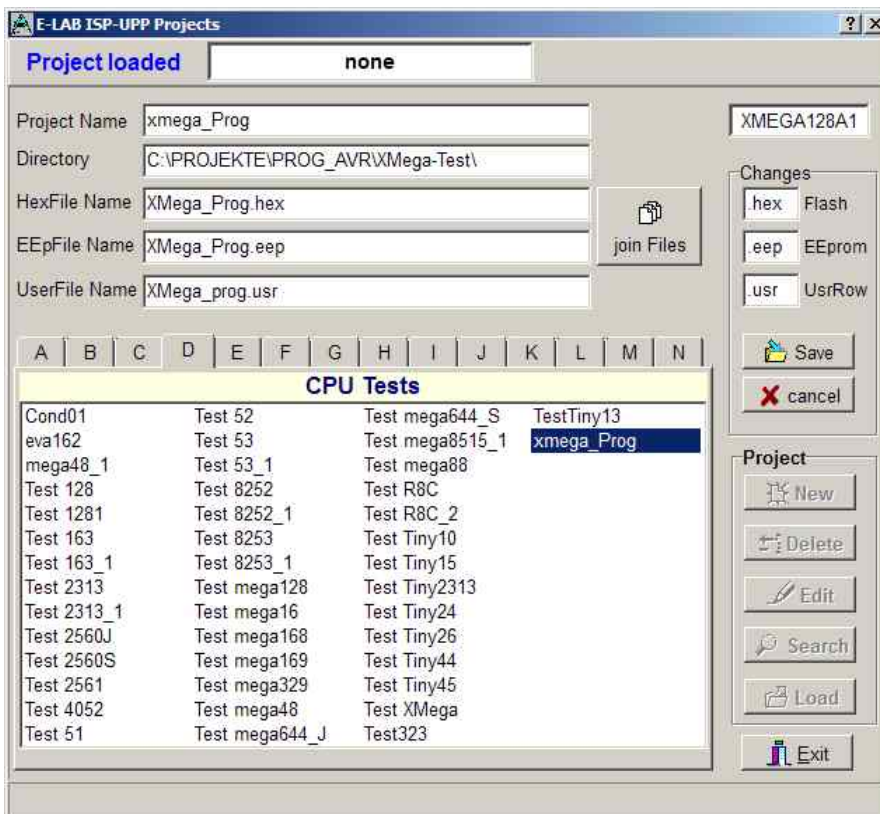
Das zweite Programm nennt sich **PackProg.exe** und kann lediglich zum Programmieren und zum Verifizieren von gepackten oder verschlüsselten Files benutzt werden, die zuvor mit AVRProg erstellt wurden. Dieses Programm bestens für Produktion geeignet und wird an späterer Stelle beschrieben.



Der Programmer zeigt diese Anzeige wenn er vom PC gesteuert wird

## AVRProg

AVRProg kann aus der AVRco IDE (PED32) aufgerufen werden. In diesem Mode werden alle Projekt Einstellungen direkt aus der IDE eingetragen. Ein weitere Möglichkeit ist der direkte Aufruf aus dem Windows Start Menu. Bei diesem Modus muss ein Projekt geöffnet werden oder es kann ein neues Projekt erstellt werden. Aus diesem Grund wird der Projekt Auswahl Dialog angezeigt:



### Kommandos

- Neues Projekt erstellen
- Projekt löschen.
- Vorhandenes Projekt ändern
- Projekt im Netzwerk suchen
- Projekt laden.
- Verlassen des Auswahl Dialogs.



## UPP2-X Portabler In-Circuit Programmer mit GLCD

### Öffnen eines existierenden Projekts:

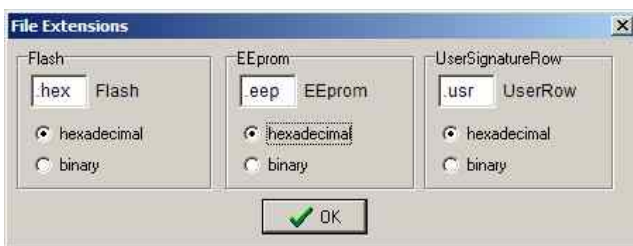
Ein vorhandenes Projekt wird mit einem Doppelklick auf den Eintrag oder mit einem einfachen Klick auf den Eintrag und einem weiteren Klick auf den Load-Button geladen. Dazu gehören alle projekt-relevanten Parameter, so dass sofort mit dem Download bzw. Programmieren begonnen werden kann.

### Anlegen eines neuen Projekts:

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.

### Dialog für die File Typen eines Projekts:

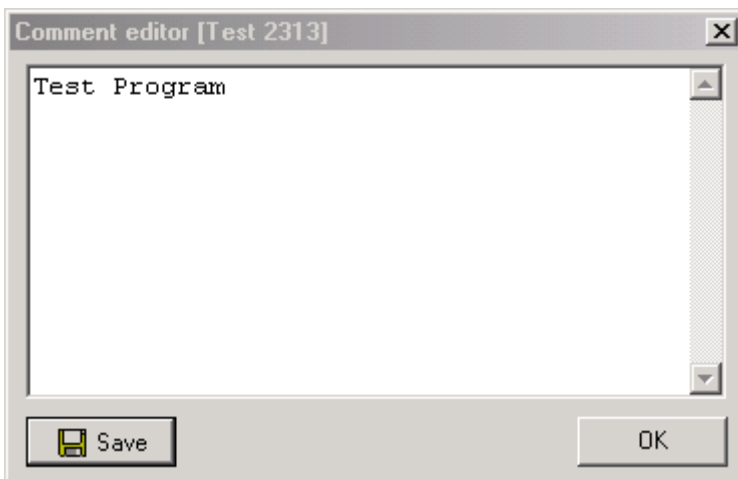
Diese Parameter sind Projekt bezogen und müssen für jedes neue Projekt separat eingestellt werden.



**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

Zu jedem Projekt kann auch ein Kommentar abgelegt werden. Siehe dazu „Kommentar“-Dialog. Zur besseren Übersicht können die Projekte mit drag and drop auf die einzelnen Seiten des Notebooks verteilt werden.

### Kommentar Editor



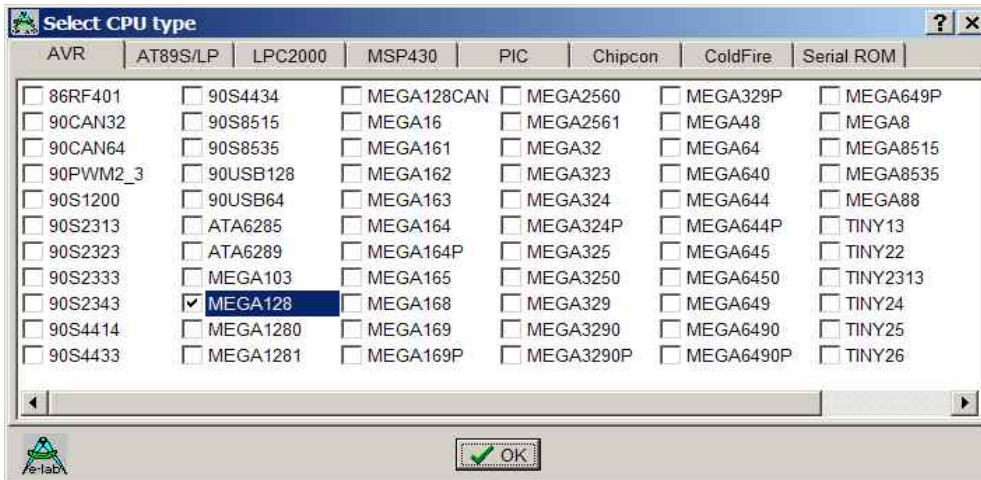
Ein Klick mit der rechten Maustaste auf einen Eintrag im Projekt Auswahl Dialog öffnet den zugehörigen Editor entweder im read-only mode oder zum Editieren.

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.

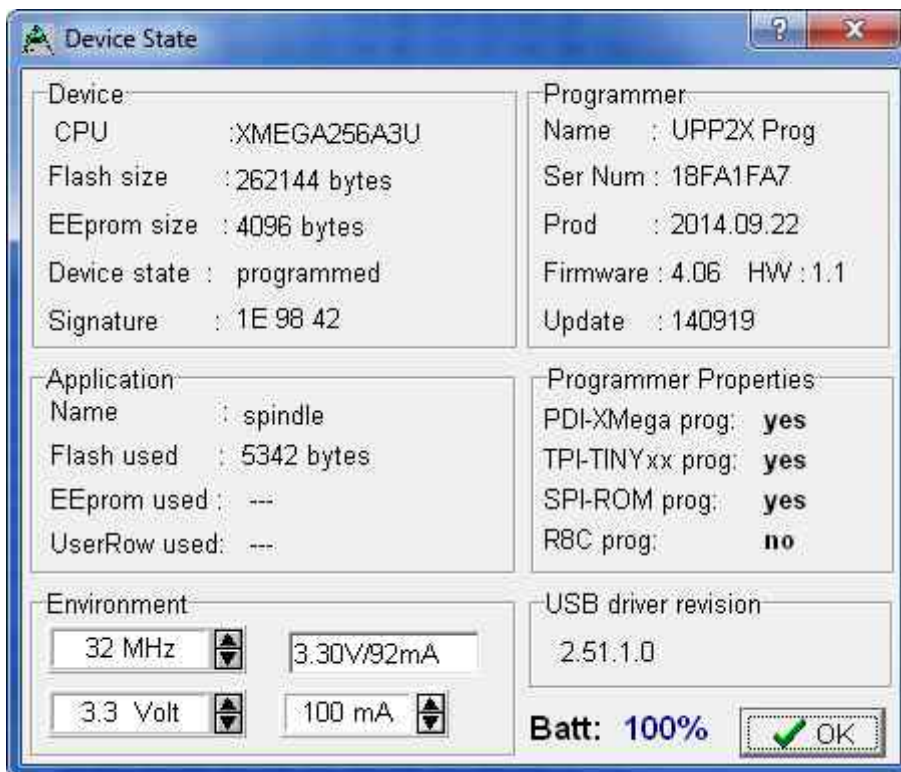


Einige CPU Typen LPC, TMS und PIC sind zur Zeit noch **nicht** implementiert.

## Dialog Device State



Nachdem die CPU ausgewählt wurde muss die Taktfrequenz der CPU mit diesem Button festgelegt werden. Der Dialog Device State zeigt die Daten des geladenen Projekts bzw. der im PC-Programm geladenen Hex-Files an.



Veränderbare Parameter befinden sich in der Environment box.

Die Clock-Frequenz der Ziel-CPU muss angegeben werden. Soll das Projekt bzw. CPU im JTAG Mode programmiert werden, ist diese Frequenz Angabe ohne Bedeutung. Der Parameter sollte aber auch hier korrekt besetzt sein. Im **SPI** Mode bestimmt der Clock die SPI Geschwindigkeit beim programmieren. Eine zu hohe Angabe führt zu Problemen bei der Programmierung. Der Parameter Clock kann jederzeit nach Bedarf geändert werden.

Die rechts daneben sichtbare Spannung ist der Wert den der

UPP-Programmer auf dem Target Board misst, falls dieses angeschlossen ist. Die Spannung bestimmt die Länge des Programmier Impulses bei der Programmierung und bestimmt dadurch die totale Programmierzeit.

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 UPP2-X gehört dazu.

Das Strom Feld dient zum Einstellen des maximal zulässigen Stroms. 0.0mA sperrt die UPP-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 UPP Programmer das Ziel System mit der eingestellten Spannung (nur im programming status). Der Strom für das Zielsystem wird auf den Vorgabe Wert begrenzt.



## UPP2-X Portabler In-Circuit Programmer mit GLCD

Die Versorgungsspannung ist für XMegas auf 3,6V begrenzt.  
Bei einigen TINYS ist die minimale Versorgungsspannung auf 5V limitiert.

Die Abteilung **Applikation** zeigt das aktuell in den PC geladene Projekt.

Die **Programmer** Abteilung zeigt Informationen über das angeschlossene Programmiergerät: Typ, Seriennummer, Herstellungsdatum, Firmware Revision und das letzte Firmware Update. Letzterer ist wichtig weil alle Updates Files mit dem Erstellungs-Datum yy-mm-dd beginnen. Im obigen Beispiel ist das Herstellungsdatum also 2011, Feb., 21.

So kann man leicht herausfinden ob ein aus dem Web frisch heruntergeladener Update File neuer als der aktuell schon auf den Programmer downgeladene ist.

Das Download eines Updates ist weiter unten im Kapitel Firmware Update beschrieben.

## USB Spannungsversorgung

Zur Versorgung des Zielsystems hat der UPP2-X einen internen Step-Up Converter der aus der USB Spannung (4..4.8V) eine Spannung von ca. 5.6V erzeugt. Damit lässt sich auch ohne externes Netzteil das Zielsystem versorgen, solange der interne Akku, der Port oder Hub nicht überlastet wird. Genauere Informationen über die Stromversorgung finden Sie weiter unten im Kapitel **Power Supply**.

## Aufruf Optionen und Command Line Parameter

### 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

Die aktuell Version dieser Software ist in der IDE integriert.

### 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 bei allen Typen
- p** Automatischer Programmier Start bei allen Typen
- r** Automatischer Target Run bei allen Typen
- c** Programm verlassen bei allen Typen
- 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 Seriennummer **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 File Namen 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.

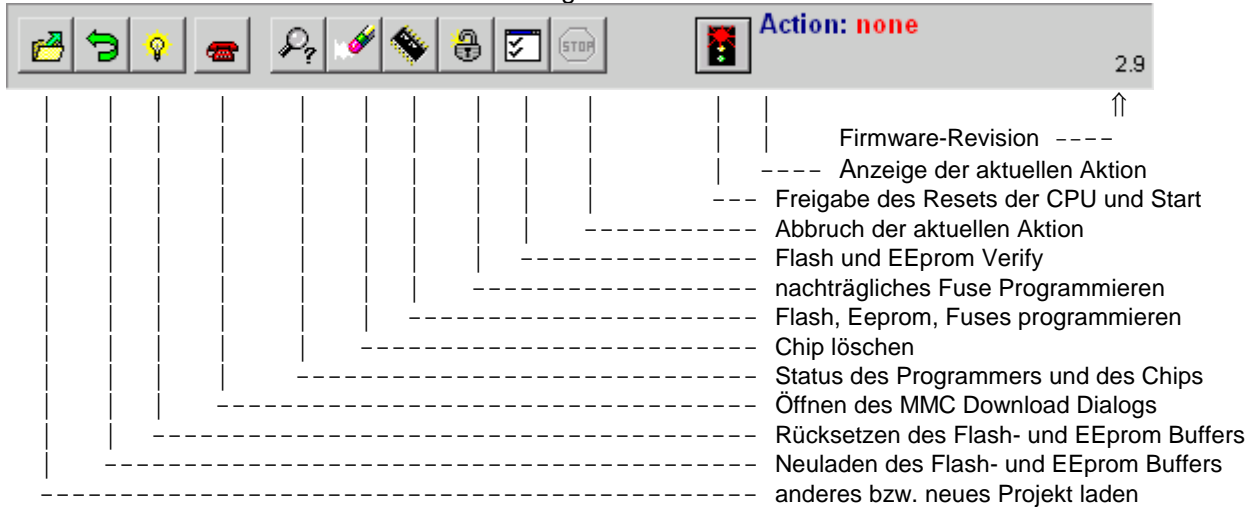
Beispiel:  
[Settings]  
ProjExt=.isp

## Alternative Möglichkeit

Es kann auch ein Pack File erzeugt werden, der alle Projekt Parameter enthält. Dieser kann dann mit dem Programm **PackProg.exe** verarbeitet oder auf die UPP SD-Karte zur Benutzung im Stand Alone Mode kopiert werden. PackProg ist im Software Paket enthalten und wird weiter unten beschrieben. Diese Option ist sehr nützlich wenn man die Files zum Programmieren außer Haus geben will.

## 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.

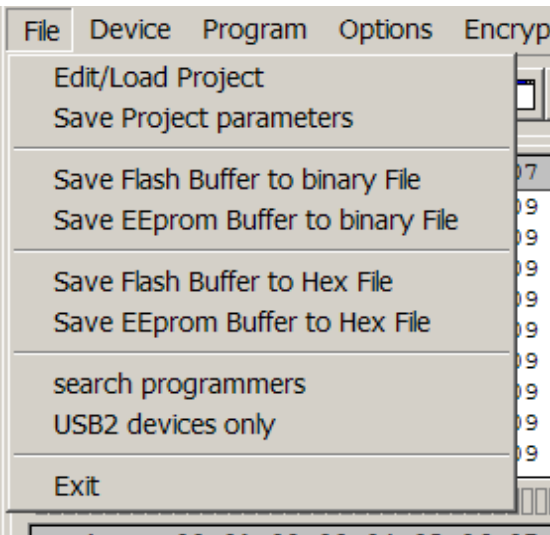


Der Download Button überträgt Projekte auf die SD Karte

## Funktionen der Buttons und Menues

Im Normalfall braucht man die Menues nicht zu benutzen. Alle Standard Operationen können mit den sogenannten Speedbuttons mittels Mausclicks 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 Programmiers** ist eine Support Funktion die die aktuelle Programmier Verbindung trennt und anschliessend versucht alle angeschlossenen Programmiers zu finden.

**USB2 devices only** verbietet das globale Suchen nach angeschlossenen Programmiers 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.

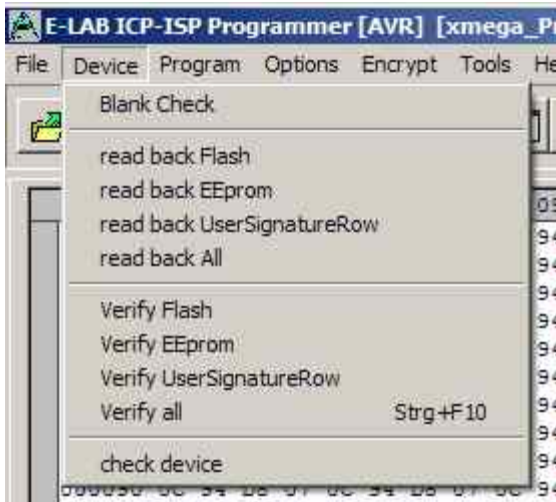


Der **Flush** Button setzt den Flash-Buffer als auch den EEPROM-Buffer komplett auf \$FF



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 und EEPROM und beim Xmega auch die UserSignatureRow in den jeweiligen Buffer 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.

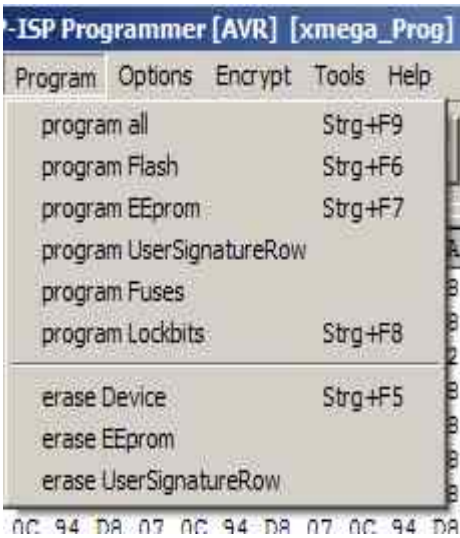
**Verify all** sofern das Target nicht lesegeschützt ist, wird der Flash und EEPROM Inhalt der CPU mit dem jeweiligen Buffer verglichen. Bei Differenzen erfolgt eine Fehlermeldung. Bei XMegas wird auch die UserSignatureRow verglichen.

**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 JTAG mode)

**program UserSignatureRow** 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 nur das EEPROM. (nicht im JTAG mode)

**erase UserSignatureRow** 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.

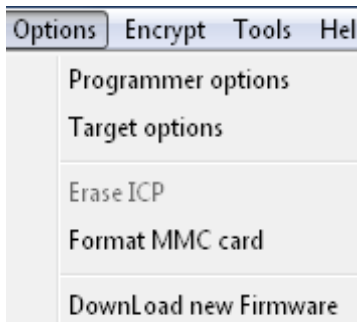


Nach dem Programmieren bleibt der während der einzelnen Aktionen an die CPU angelegte Reset erhalten. Die CPU kann jetzt, ohne den Programmer abzuziehen, durch Klicken der **Run** Taste gestartet werden.

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

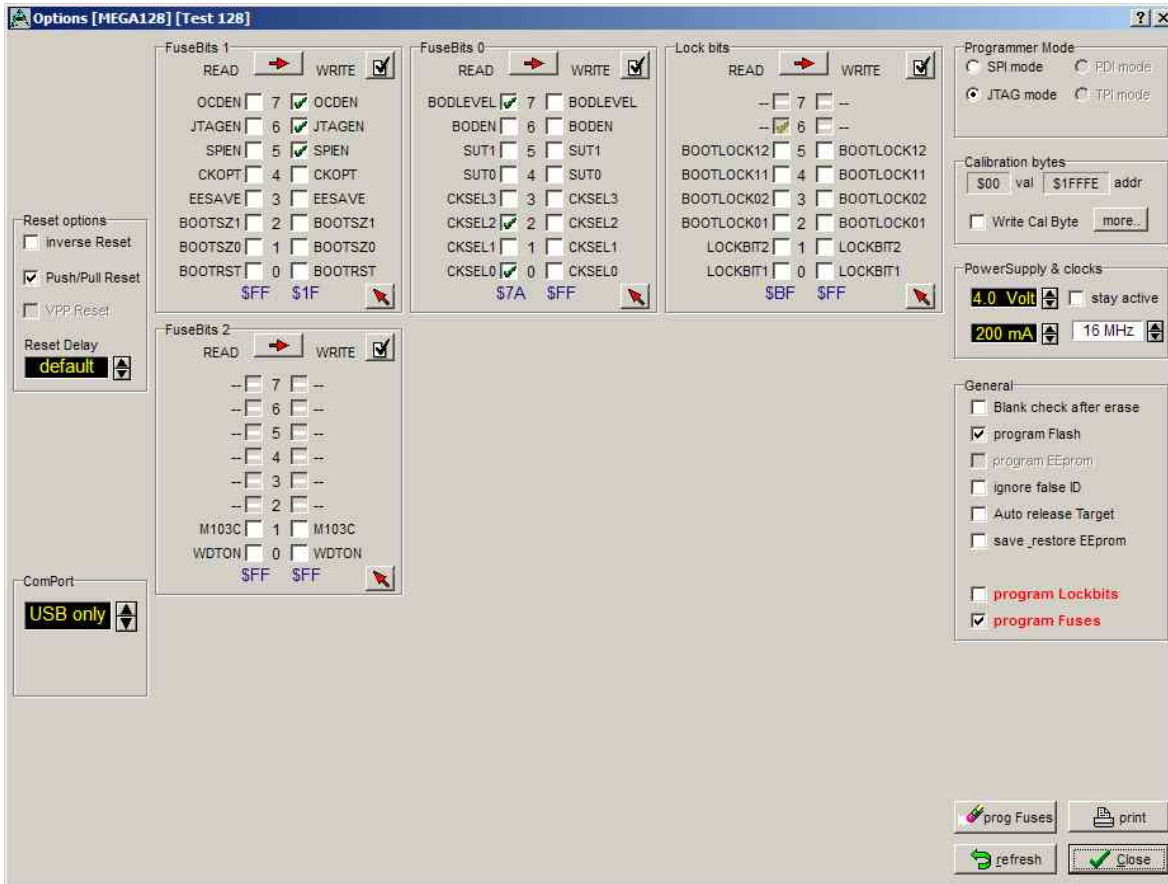
**Format MMC card** startet ein Windows Utility das die MMC/SD Karte im Flash Drive des PCs formatiert.

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

SD-cards der Grössen 2..32GByte werden unterstützt. Auch SDHC Typen. Bitte nur speed grades 4 und höher benutzen. Für höchste Geschwindigkeit ist die Cluster Grösse wichtig. Gute Werte sind: bis 4GByte size -> 4kB, 8GByte size -> 8kB, 16GByte size -> 16kB, 32GByte size -> 32kB

## Programmer Options Dialog

Mit dem Options-Dialog wird das Verhalten des Programmers 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 ist das Feld Fuse bits gesperrt, ansonsten sind die vorhandenen Bits zum Ändern freigegeben. Die Bedeutung der Fuse Bits ist dem CPU Manual 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.

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

**Program Fuses** gibt das Programmieren der Fuses frei und sollte unbedingt immer aktiviert sein. Nur für spezielle Zwecke kann es Sinn machen, das Fuse Programmieren zu unterbinden.

**Program Lockbits** gibt das Programmieren der Lockbits frei.

Im Feld **Security** werden die Lockbits für den Ausleseschutz eingestellt. **Program locked** ist nur wenig sinnvoll. Im Normalfall ist entweder kein Schutz mit **no locks** oder der komplette Schutz mit **program + verify locked** einzustellen. Die genaue Funktion der Fuses ist im Datenbuch des jeweiligen Prozessors nachzulesen.





## UPP2-X Portabler In-Circuit Programmer mit GLCD

Im Feld **FuseBit** 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).

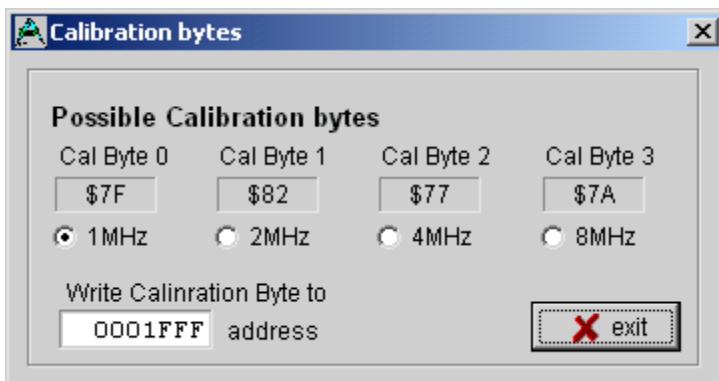
**Achtung:** einige CPUs besitzen eine **SPIEN** Fuse. Dies ist normalerweise 'don't care' für SPI (low voltage) Programmierung. Aber es gibt Ausnahmen, wie z.B. der Tiny12. Hier muss die SPIEN Fuse aktiviert werden. Andernfalls kann nicht mehr auf den Chip zugegriffen werden.

Im **JTAG Mode** ist die **SPIEN** Fuse immer programmierbar. Wenn **SPIEN** und **JTAGEN** beide disabled sind ist die CPU ebenso nicht mehr programmierbar. Beachten Sie, dass JTAG einige I/O-Pins für die Programmierung benutzt.

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.

### Calibration Bytes Dialog

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 (Checkbox) 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 AVR's ist beim Standard RC-Oszillator kein Calibration Byte notwendig. Diese laden dieses Byte automatisch aus dem dafür vorgesehenen EEPROM Speicher in das OSCAL Register.

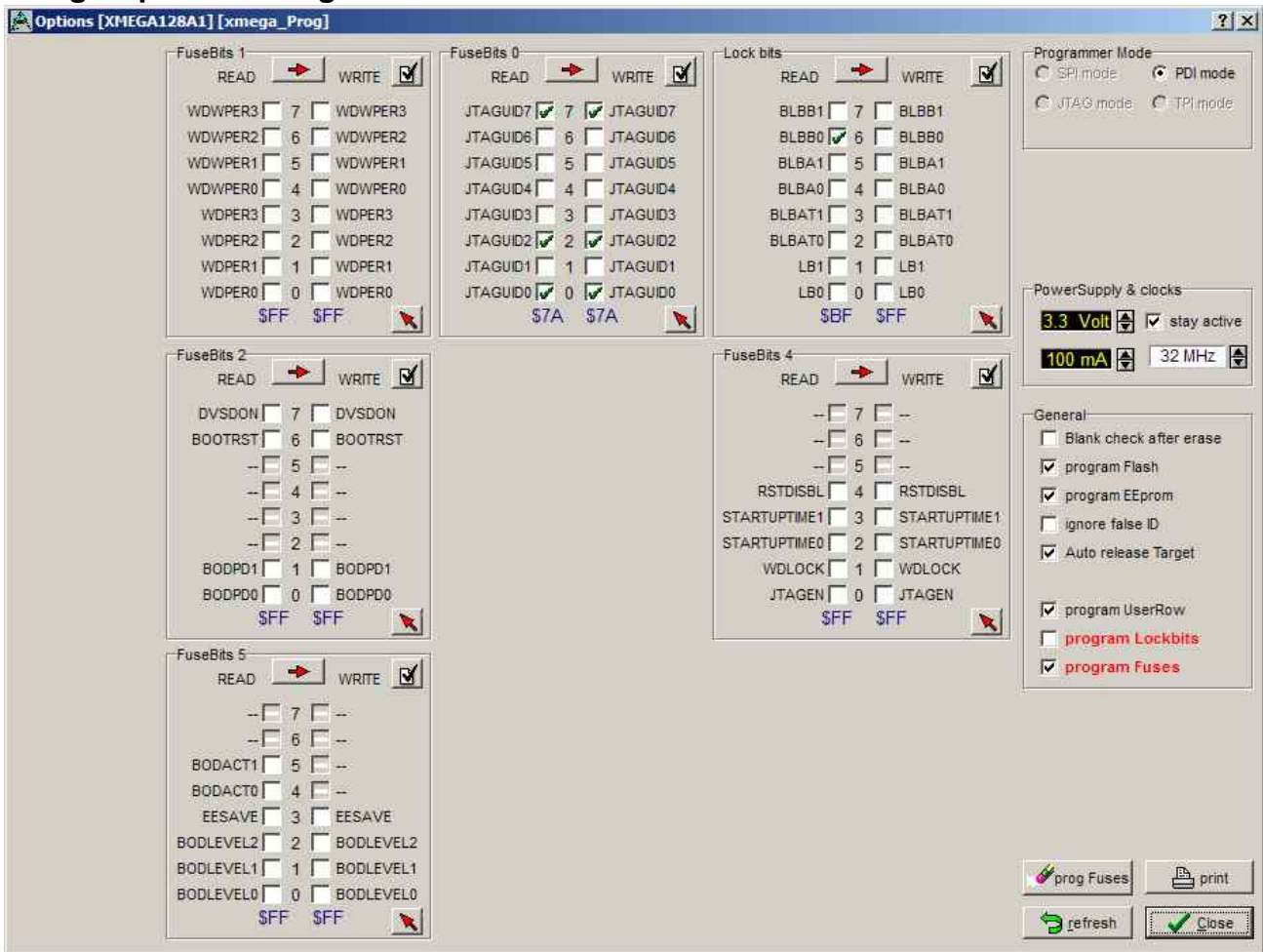
**Achtung:** Alle Fuse und Lockbits sind low active. D.h. wenn im Datenblatt für ein bestimmtes bit eine „0“ steht, muss in dem zugehörigen Feld im Dialog ein Haken sein. Ein log 1 bit wird deshalb durch ein leeres Feld dargestellt. Atmel verwendet für Fuse und Lockbits immer negative Logik!

Im Feld **ComPort** wird die zu verwendende serielle Schnittstelle eingestellt, wenn automatic auf "AUS" eingestellt wird. Ist die Automatik "EIN", so werden alle acht möglichen Schnittstellen abgefragt, bis der Programmierer gefunden ist. Für den UPP Programmer sollte hier **USB** eingestellt werden.

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 Spezialfälle.

Das Feld **Programmer Mode** ist nur sichtbar, wenn die gewählte CPU sowohl SPI als auch JTAG Programmierung unterstützt. In diesem Fall kann hier der Modus eingestellt werden.

## XMega Options Dialog



Einige Fuse 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. SupplyVoltage ist beim XMEGA auf 3.6Volt begrenzt.



## UPP2-X Portabler In-Circuit Programmer mit GLCD

### Power Supply

Das **Power Supply** Feld dient zum Einstellen des maximal zulässigen Stroms. 0.0mA sperrt die UPP-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 UPP Programmer das Ziel System mit der eingestellten Spannung (nur im programming status). Der Strom für das Zielsystem wird auf den Vorgabe Wert begrenzt.

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 **Program 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 bei der SPI-Programmierung 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 verändert sich der Wert um 2..5%.
2. Externer RC-Oszillator.  
Wird diese Fuse 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 diese Fuse aktiviert, muss auch unbedingt ein 32kHz Quartz angeschlossen werden, sonst kann nach dem Programmieren nicht mehr auf die CPU zugegriffen werden.

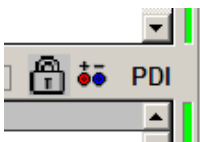
Zu beachten ist weiterhin, dass beim Programmieren die aktuelle Fuse Einstellung gilt. Die neu programmierte wird frühestens nach dem Programmier Vorgang gültig. Manche Fuses werden auch erst nach einem PowerDown aktiv.

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.

Die neueren AVRs kommen mit internem RC-Oszillator aktiviert ab Werk. Hier muss der erste Zugriff auch mit 1MHz! Mit einer 3Volt Versorgung muss dieser Wert 900kHz sein. Dies erfolgt automatisch durch den Programmer. Dabei werden mit 32kHz zuerst alle Fuses auf den gewünschten Wert gesetzt. Die CPU schaltet danach z.B. auf den externen Oszillator/Quarz um, der natürlich vorhanden sein muss. Die Programmierung erfolgt jetzt mit der eingestellten Geschwindigkeit, z.B. 8MHz.

Im **JTAG** 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 die JTAGEN Fuse aktiviert ist und auch so bleibt.

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

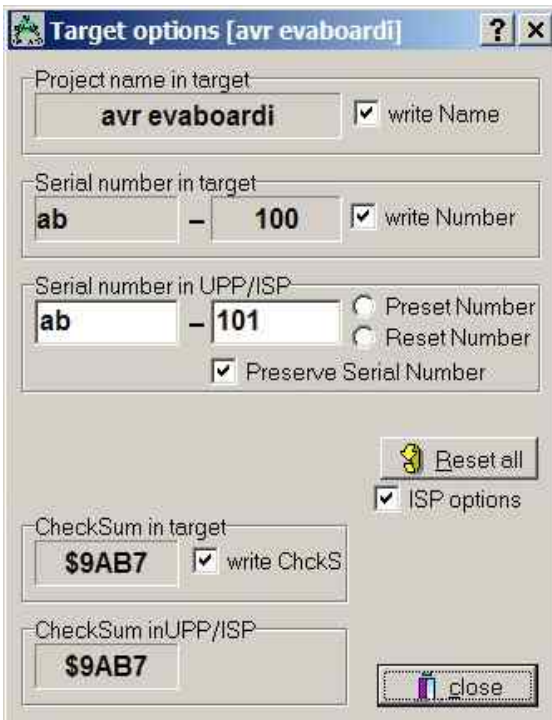


Das Schloss bedeutet hier, dass für das Projekt der Ausleseschutz (Lockbits) für die Ziel CPU aktiviert ist. Das Netzteil Zeichen bedeutet, dass der UPP (für das momentan im PC geladene Projekt) das Target System mit Strom/Spannung versorgen muss.

## 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 Daten im Flash

Vorausgesetzt es ist noch ausreichend Platz in dem Flash Bereich der Target-CPU, so können hier am oberen Ende des Flashs Speicher 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 ICP/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 abgelesen werden.



## UPP2-X Portabler In-Circuit Programmer mit GLCD

### 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 UPP 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
01FFF0 61 72 64 69 BA 00 73 65 72 23 61 62 65 00 B7 9A ardi..ser#abe...
```

LSB    MSB            LSB    MSB  
      ↑            ↑            ↑            ↑  
ser number            checksum

### **Achtung:**

Wenn aus der Flash Karte heraus programmiert wird, dann wird die Serial Number jedes mal auf die Karte geschrieben nach jedem erfolgreichen Programmier Zyklus. Das bedeutet Stress für die Karte und kann irgendwann zum Ausfall dieser führen. Um einen frühzeitigen Ausfall zu vermeiden, sollten die neueren und besseren Karten Typen mit dem Attribut **x80** eingesetzt werden. Diese sind wesentlich schneller und bieten eine 10 längere Lebenszeit bezogen auf die Schreib Zyklen.



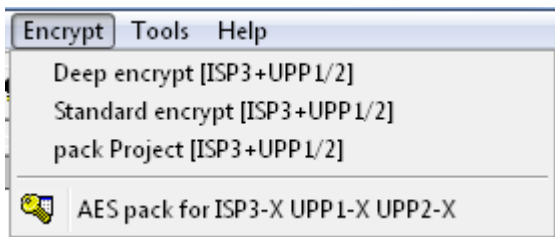
## Projekt Download

Der UPP Programmer hat mehrere Arbeits Modi. Während der PC-direkt Mode keine MMC Flash Karte benötigt, brauchen alle anderen Modi eine mit einem Project geladene MMC Karte. Es gibt dazu mehrere Möglichkeiten ein Projekt in eine MMC/SD Karte zu schreiben.

## Flash Karten Reader/Writer

Dieses Gerät meldet sich beim PC als zusätzliches Laufwerk an.

Mit dem E-LAB Steuerprogramm wird eine gepackte oder verschlüsselte Projekt Datei gebildet. Dazu muss das gewünschte Projekt geladen werden. Dann wird das Encrypt Menu geöffnet und der Punkt **pack Project** oder **Standard encrypt** angeklickt. Jetzt öffnet sich ein File Dialog zum speichern der Datei.



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

Wenn jetzt das MMC Laufwerk (reader/writer) gewählt wird, speichert Windows die gepackte Datei auf die MMC Karte.

Alternativ kann diese Datei auch auf ein anderes Medium, z.B. Floppy, abgespeichert werden. Oder sie kann als Attachment einer Mail verschickt werden.

Bitte beachten: Die mit einer **encrypt** Optionen erzeugten Dateien können niemals direkt mit den Programmern verarbeitet werden. Hierzu ist immer das Programm **PackProg.exe** notwendig.

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-Engineering“ anzuwenden.

Weiterhin besteht die Möglichkeit, das encryptete File mit einem Passwort zu versehen, so dass nur der richtige UPP 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-USB wird zwingend das Programm PackProg benötigt.

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

**AES pack** 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

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.

Diese **Verschlüsselung** hat die Aufgabe, 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.

Die verschlüsselte Datei (\*.enu oder \*.en#) enthält die Seriennummer des UPP Gerätes, für das diese Datei bestimmt ist. Damit wird auch sichergestellt, dass nur dieser UPP diese Datei verarbeiten kann.

Um diese Datei generieren zu können muss dem Ersteller der Datei und damit auch AVRprog.exe die Seriennummer des UPP bekannt sein.

Diese Nummer kann direkt festgestellt werden wenn der Ziel-UPP am diesem PC angeschlossen ist. Die Taste **Read Key** drücken und im Feld **edit Password** erscheint die interne Seriennummer des UPP. Diese kann jetzt mit der Taste **Add Key** der Liste hinzugefügt werden.

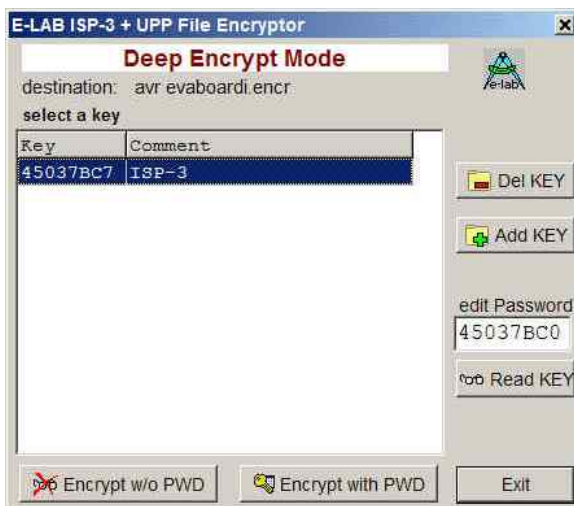
Wenn die Seriennummer unbekannt ist, muss der Besitzer des Geräts diese mit **PackProg** auslesen. Sie ist im **setup - request password** Dialog von PackProg zu finden. Der Ersteller des encrypted File muss diese Nummer im Feld **edit Password** eintragen und sie mit der Taste **Add Key** der Liste hinzuzufügen.

Zum Erstellen der verschlüsselten Datei muss jetzt aus der Liste der gewünschte UPP ausgewählt werden (blauer Balken). Mit dem Button **Encrypt with PWD** wird das File gebildet und gespeichert.

Wenn eine Bindung der Datei an einen bestimmten UPP nicht gewünscht ist, dann kann mit der Taste **Encrypt w/o PWD** ein File erstellt werden, das zwar verschlüsselt ist, aber kein Passwort besitzt. Dieses File ist dann auf jedem UPP verarbeitbar.

Abhängig welche File Erweiterung gewählt wurde, hat das generierte File jetzt entweder die Erweiterung **\*.enu** oder **\*.en0** oder **\*.en1** etc

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 verschlüsselt aber nicht Passwort geschütztes File generiert und gespeichert.

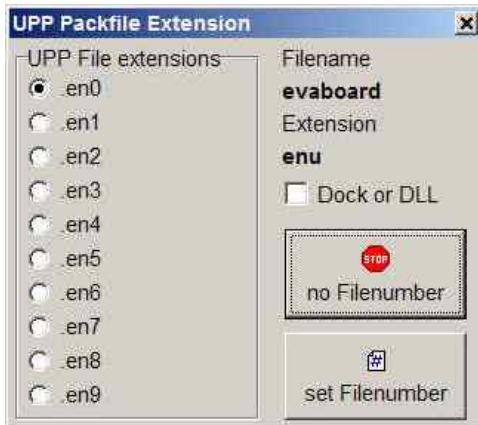
Die Generierung eines UPP2-X Passworts wird auch weiter unten im Kapitel PackProg Tool beschrieben.

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.

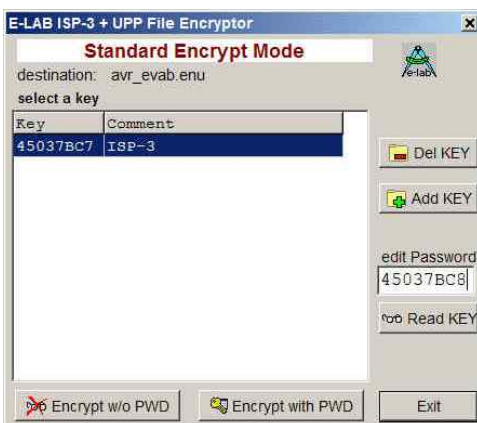
## Standard encrypt Modus

Diese Option generiert ein gepacktes und **gut verschlüsseltes** Projekt, das nur mit dem Programm **PackProg.exe** in Verbindung mit dem UPP2-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* bestimmt beim UPP2-X ob mehr als 10 Projekte auf der Flashkarte sein dürfen.



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.

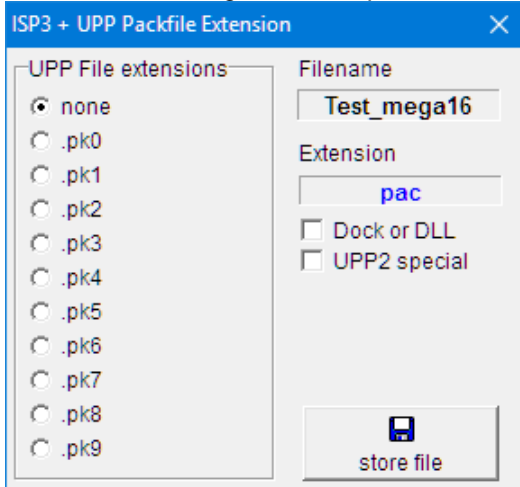
Diese Option wird durch den Button *Encrypt with PWD* gewählt bzw. mit dem Button *Encrypt w/o PWD* gesperrt.

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-USB 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:

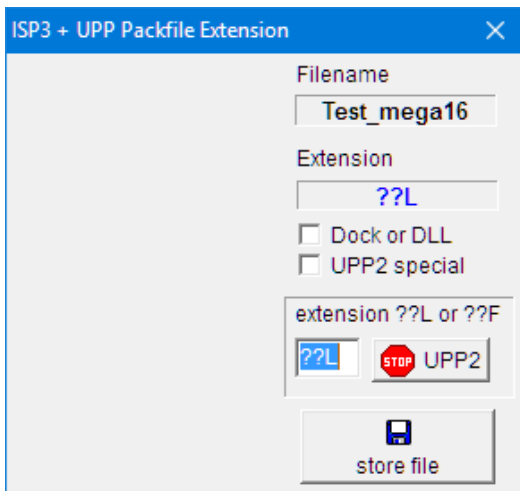
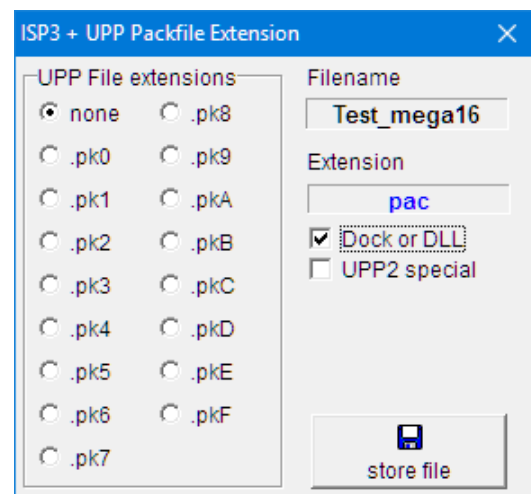
Hiermit wird die File Extension (Endung) festgelegt. Die Endung kann entweder eine Nummer enthalten, also pk0..pk9 zur besseren Verarbeitung in den UPP Programmern oder ein einfaches \*.pac sein. Die Auswahl erfolgt mit den zwei File Nummern Buttons.

Die Checkbox *Dock or DLL* bestimmt dass beim UPP2-X mehr als 10 Projekte auf der Flashkarte sein dürfen.

Wurde die Dock Option gewählt, öffnet sich neben stehender Dialog:

Mit der UPP Version „D“ mit Docking Station können mehr als 10 Projekte gespeichert und ausgewählt werden.

Diese Datei Typen können mit dem Programm PackProg.exe und auch direkt mit den UPP Programmern verarbeitet werden.



Mit der Checkbox **UPP2 special** können fast beliebige File Extensions vorgegeben werden, z.B. \*.axL oder \*.3zL. Diese Files lassen sich aber dann nur mit dem UPP2-X Programmer verarbeiten.

## Allgemeine 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 UPP 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 UPP2-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.

Abhängig von der gewählten Datei Erweiterung bekommt das erstellte File die Erweiterung *\*.enu* oder *\*.en0* or *\*.en1* etc.

## Programier Geräte

Das Programm *“PackProg.exe“* unterstützt alle Programmer der Typen *ISP3-USB*, *UPP1* und *UPP2*. Weitere Informationen im Kapitel **Tool Packprog**. Dieses Handbuch sollte auch der Empfänger der Dateien erhalten.



## 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.

<b>1</b>	pack Project [ISP3+UPP1/2]	Standard PAC File
<b>2</b>	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.

- A. Standard Encryption (only AES used)
- B. 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.
- C. 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 Nonsense programmiert.

## Interaktives Programmieren



Das interaktive Programmieren wird mit diesem Button der Programmer Oberfläche gestartet.

Dabei verhält sich der UPP Programmer wie sein kleiner Bruder, der ISP3-USB. Das ganze Chip wird gelöscht incl. der Lock Bits und anschliessend neu programmiert. Die Fuse und Lock Bits werden wie unter Options eingestellt, behandelt.

Da hier der PC und das Steuerprogramm alle relevanten Daten unter Kontrolle hat, ist die MMC Karte hier nicht notwendig und wird auch nicht gebraucht.

## Erstellen von Projekten

Auf den vorhergehenden Seiten wurden die zwei Möglichkeiten aufgezeigt wie die UPP Projekt Files erstellt werden können. Entweder durch direkten Download auf die MMC Karte des UPP oder durch ein Flash Laufwerk des PCs.

Dabei wurde darauf hingewiesen, dass die Standard UPP Version nur ein File im portable Mode benutzen kann, und zwar mangels Auswahl Möglichkeiten am UPP selbst.

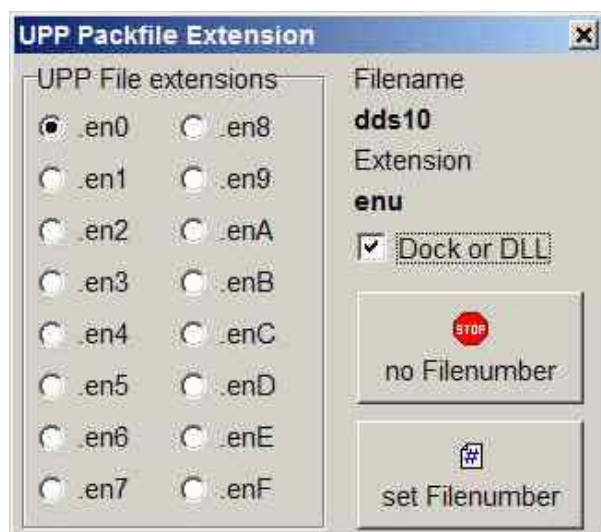
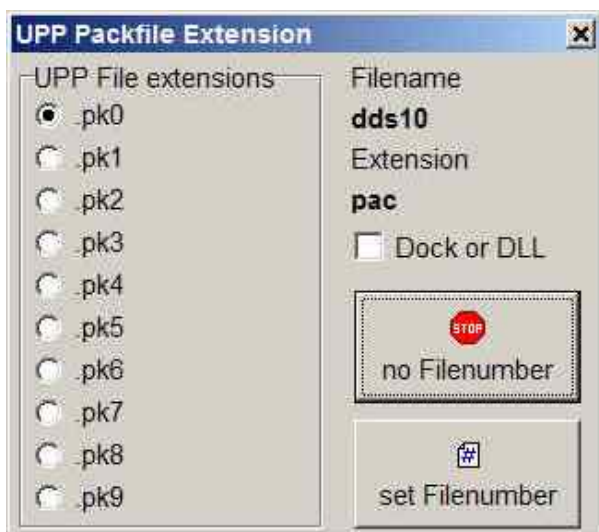
Diesen Umstand berücksichtigen die **Version S** und **Version D** des UPP1, indem die Version „S“ auf der Unterseite einen Drehschalter besitzt, der die Auswahl von einem aus 10 geladenen Projekten erlaubt. Bei der Version „D“ kann ebenfalls über die Docking Station eins von 16 Projekten angewählt werden.

Dabei stellt sich das Problem, dass die Zuweisung von Projekten auf der MMC Karte an die jeweilige Schalterstellung nicht direkt beeinflusst werden kann. Diese ist abhängig von der Reihenfolge der FAT16 Directory Einträge auf der MMC Karte. Wird eine Karte komplett gelöscht und werden dann nacheinander Files auf die Karte geschrieben, werden diese auch in dieser Reihenfolge dem Schalter zugewiesen. Werden dann aber Files gelöscht und/oder überschrieben, dann kann sich diese Reihenfolge erheblich ändern. Das hat zur Folge, dass in diesem Fall die neue Reihenfolge aus dem UPP File-Dialog des PC Programms notiert werden muss. Wenn hier nicht sorgfältig gearbeitet wird kann es dann zu erheblichen Problemen kommen.

Aus diesem Grund bieten die beiden Download Funktionen die Möglichkeit jeder Datei einen festen Platz bzw. einen festen Bezug zur Schalterstellung zuzuweisen.

Normalerweise haben alle UPP Pack Files die Datei Erweiterung **\*.pac** und die Encrypted Versionen die Datei Erweiterung **\*.enu**. Um diesen Dateien einen festen Listenplatz zuzuweisen gibt es die Möglichkeit eine Nummerierung in die jeweilige Datei Endung einzubauen, die den UPP Programmer dazu zwingt dieser Datei einer Schalterstellung zuzuweisen.

Beim Erstellen bzw. Download von **PAC** und **ENU** Dateien öffnet sich einer dieser beiden Dialoge:





## UPP2-X Portabler In-Circuit Programmer mit GLCD

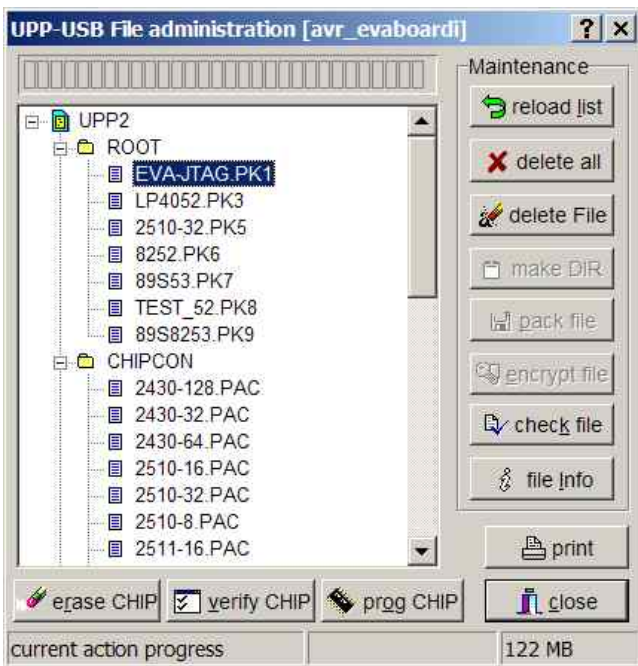
Wird der Button **no Filenumber** gedrückt, wird ein normales \*.pac bzw. \*.enu File generiert. Wird der Button **set Filenumber** gedrückt, dann bekommt ein encrypted File die Erweiterung \*.en0 und ein gepacktes File die Endung \*.pk0. Der letzte Buchstabe wird durch das Radio-Check Feld bestimmt und definiert wiederum den Platz in der internen File Liste des UPP1 und damit die Drehschalter Position.

Der UPP weist Dateien mit **Endungsnummern**, die schon auf der Flashkarte existieren, zurück. Es können jedoch z.B. folgende Files co-existieren:

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

### Programmieren von SD Karte

Weiterhin gibt es auch noch die Möglichkeit über den PC zu programmieren, aber dabei das Projekt in der MMC Karte zu benutzen. Dazu wird mit dem Download Button  der Download Dialog geöffnet.

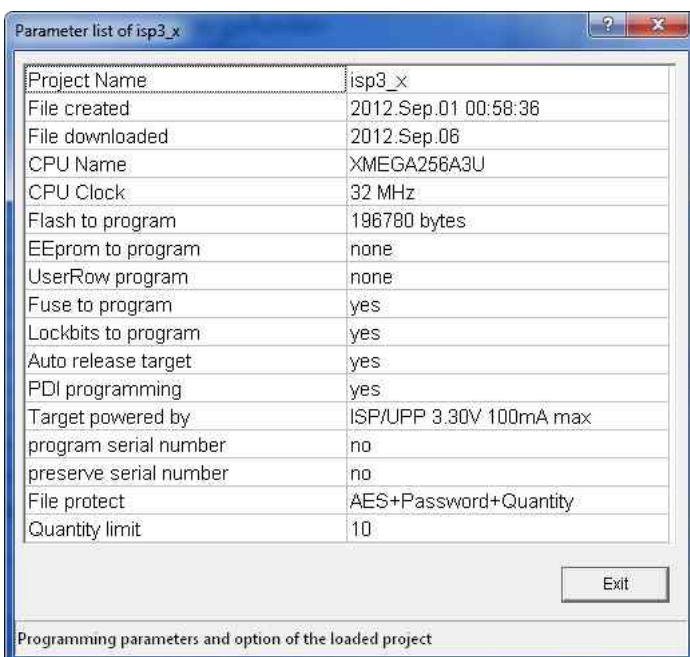


Das Projekt in der MMC Karte wird durch einen Click in das Listenfenster ausgewählt. Damit werden die erase, verify und prog Buttons freigegeben.

Jetzt kann mit dem **prog Chip** Button die CPU programmiert werden.

Ist das Chip nicht auslese geschützt kann optional mit dem **verify Chip** Button die CPU getestet werden.

Die aktuelle File Liste der MMC Karte kann mit dem **Print** Button als Hardcopy ausgedruckt werden



Mit dem **File Info** Button können relevante Parameter des auf der MMC Karte ausgewählten Projekts angezeigt werden

## Status, Fehleranzeige und Probleme

Mögliche Fehlermeldungen des Programmiersystems (AVRprog PC):

### Programmer 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 das Netzteil ist nicht angeschlossen oder 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. Auf jeden Fall sollte die Programmer Port Einstellung auf **USB only** gesetzt werden.

## Meldungen im Stand Alone Betrieb

Ohne angeschlossenen PC arbeitet der UPP Programmer nur mit seiner MMC Flash Karte. Hierbei wird der Status optisch (Display und LED) und akustisch angezeigt. Weiteres unter *Stand-Alone Betrieb*.

## Harter Reset

In beiden Modi, PC-gesteuert und Stand-alone, kann durch gleichzeitiges Drücken der beiden Pfeil Tasten ein Hardware Reset im Programmer ausgelöst werden.

Der UPP2-X besitzt auch eine RESET Taste am unteren Ende des Gehäuses

## Aufwecken des UPP2-X

Durch kurzes Drücken der RESET Taste startet das Gerät.

## Lade Zustand

Im Grundzustand kann der Akku Stand durch einen Druck auf die C Taste festgestellt werden.

## Ausschalten

Durch gleichzeitiges Drücken der **S** und der **C** Taste schaltet sich das Gerät aus.





## PackProg

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

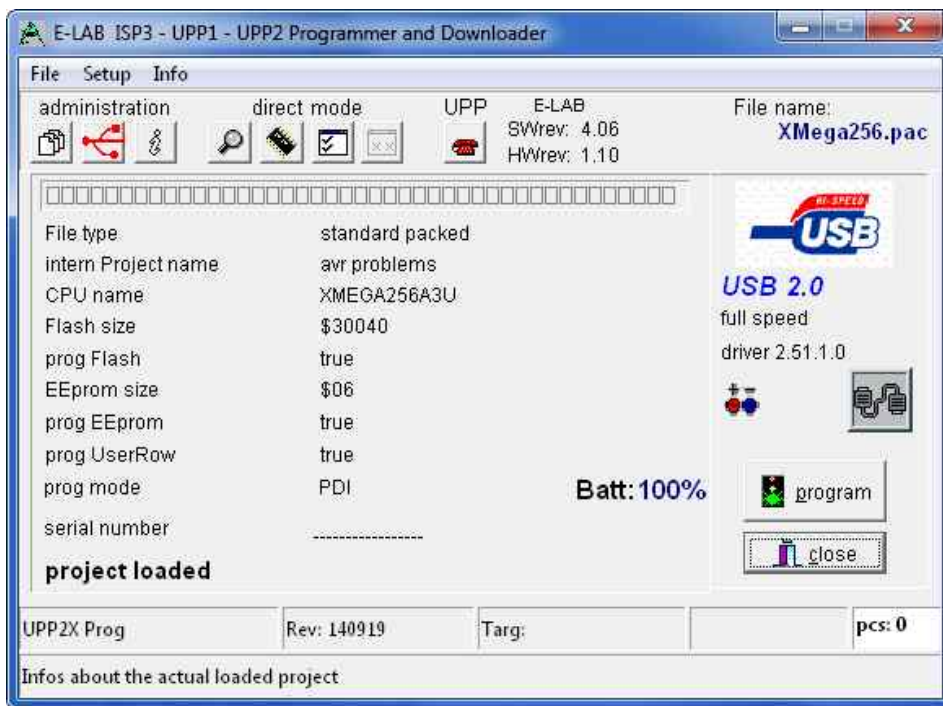
1. für das Erstellen von Projekten mit Fuse und Lockbits, Definition des Flash und EEprom Files etc
2. für das direkte inCircuit Programmieren von Chips mit allen Programmer Typen
3. für das indirekte inCircuit Programmieren (mittels Flash Karte) durch die UPP Programmer Typen
4. für das Erstellen von gepackten oder encrypteten Projekten
5. für den Download von gepackten Projekten in die Flash Karte der UPP Programmer 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 Programmer Typen ISP3, UPP2 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 einzige Tool die „deep encrypted“ Projekte verarbeiten.

## Buttons und Menues

### Projekt Importieren



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

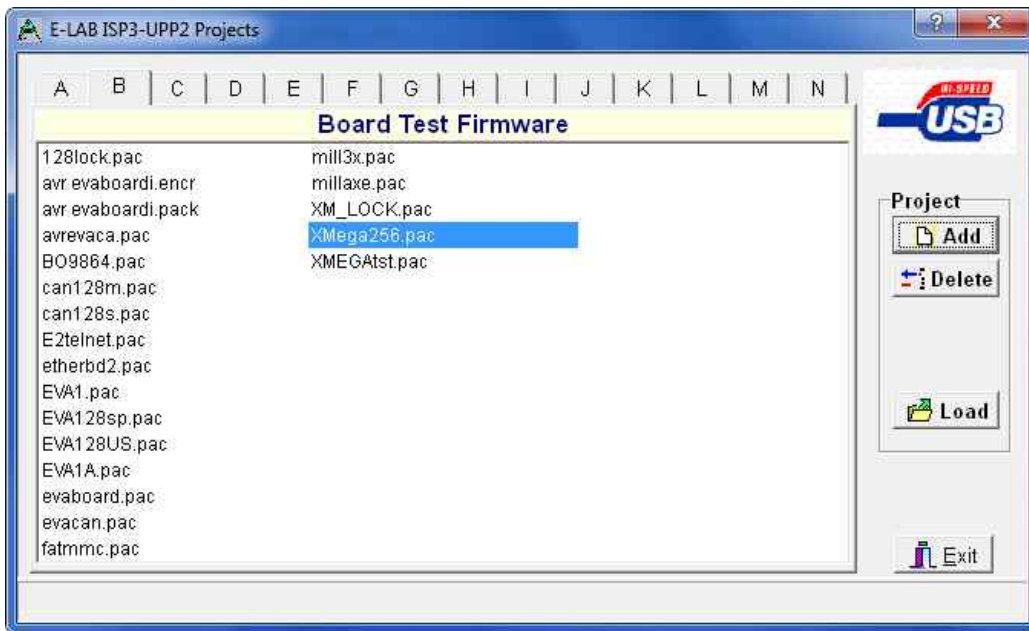
Das bedeutet dass ein neues Projekt zuerst einmal registriert werden muss bevor man mit diesem arbeiten kann.

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



Das eröffnet folgenden Projekt Dialog



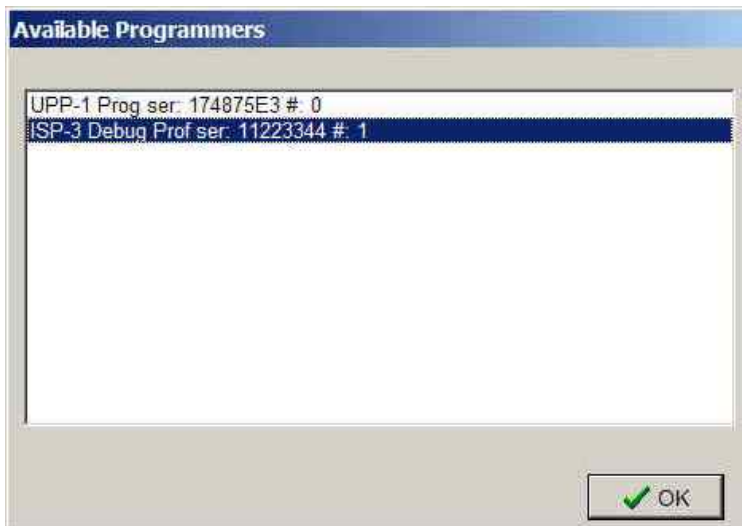


Mit dem Add Button kann jetzt ein Projekt dem Pool hinzugefügt werden.

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

Mit dem Load Button oder einem Doppel Click 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.



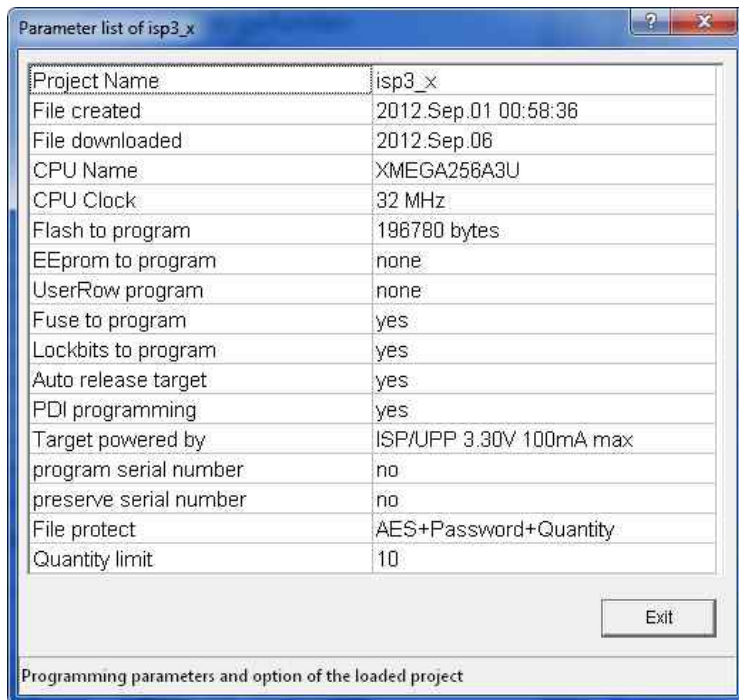
Dies geschieht mit dem Check-USB Button.

Wird ein Programmer gefunden so wird dies im Hauptfenster angezeigt.

Werden mehrere Programmer gefunden wird dies im nebenstehenden Dialog angezeigt.

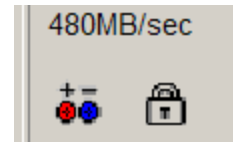
Da immer nur ein Programmer zu einer Zeit gesteuert werden kann, muss einer ausgewählt und bestätigt werden.

## 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.

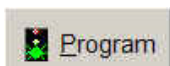
Das Schloss Symbol zeigt an dass der Programmer das Target auslese schützen muss.

## Programmieren im direkt Mode

Beim Transparent Modus wird der Programmer direkt vom PC aus gesteuert und mit Daten versorgt, ohne dass die Flash Karte des Programmers dabei beteiligt ist. 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.

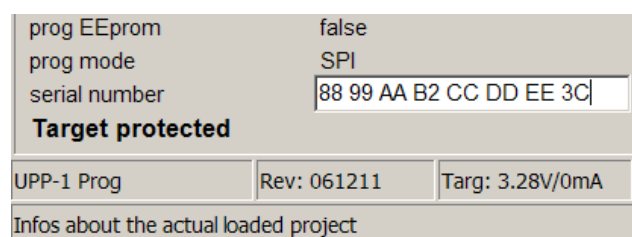
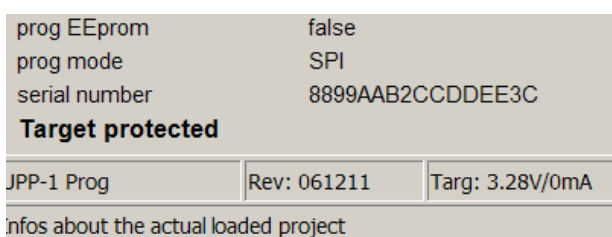
Was im einzelnen ausgeführt wird steht in dem Pack oder Encrypt File (Project)

Da das Steuerprogramm alle notwendigen Daten kennt und im Zugriff hat, ist die SD Karte nicht notwendig und wird hier nicht benutzt.



**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.

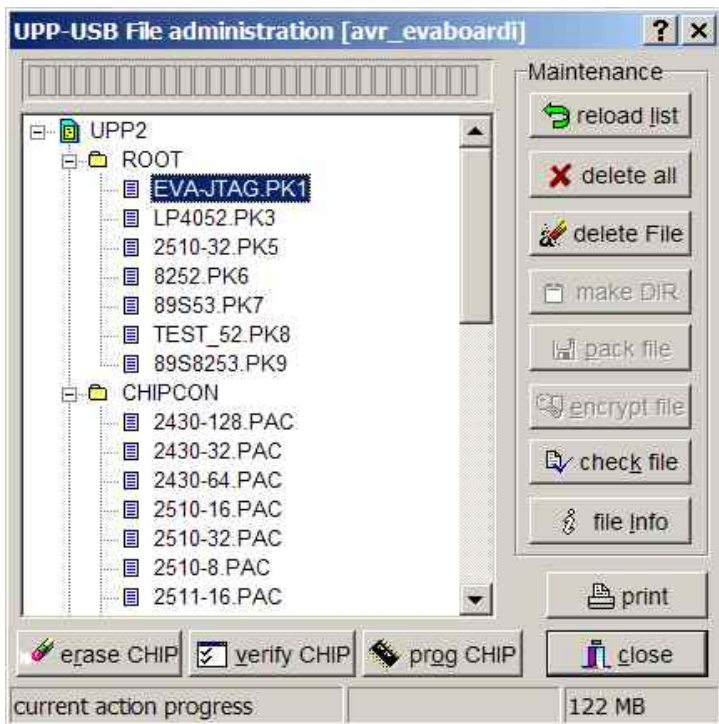
Wurde bei der Projekt Erstellung (Pack oder Encrypted File) die Serien Nummer Programmierung freigegeben so wird die aktuelle Nummer angezeigt. Mit einem Doppelclick auf dieses Feld lässt sich diese hier editieren. Beim Programmieren wird diese Nummer im Flash mit abgespeichert und anschliessend inkrementiert.



## Programmieren von SD Karte

der Download Button liest die SD Karte und öffnet folgendes Menu:

Deep encrypted Projekte können nicht auf der Flash Karte abgespeichert werden. Sie müssen im Direkt Modus verarbeitet werden.



Das aktuell im PC geladene Projekt erscheint ganz oben in der blauen Kopfleiste.

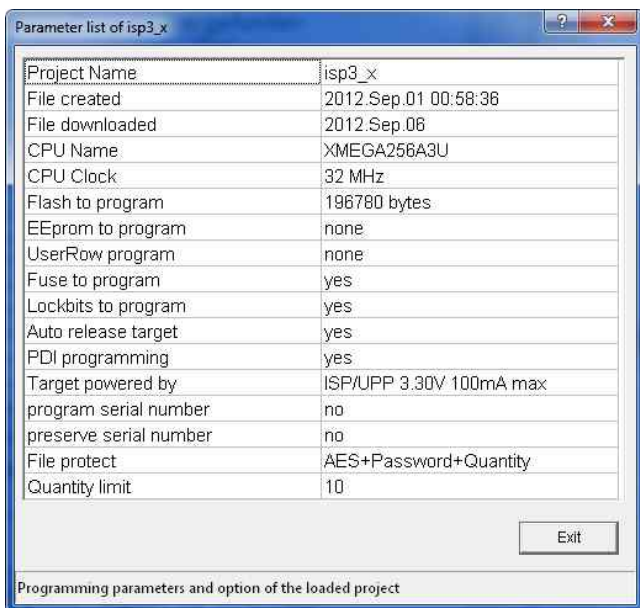
Eventuell in der MMC/SD Karte vorhandene Projekte werden im Listenfeld angezeigt.

Mit **check file** und **file Info** kann der Download auf der Karte noch einmal überprüft werden.

Das Projekt in der MMC Karte wird durch einen Click in das Listenfenster ausgewählt. Damit werden die erase, verify und prog Buttons freigegeben.

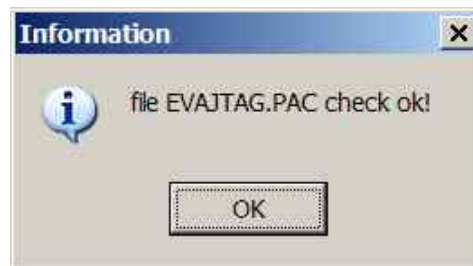
Jetzt kann mit dem **prog CHIP** Button die CPU programmiert werden. Ist das Chip nicht auslese geschützt kann optional mit dem **verify CHIP** Button die CPU getestet werden.

Die aktuelle File Liste der MMC Karte kann mit dem **Print** Button als Hardcopy ausgedruckt werden



Mit dem **file Info** Button können relevante Parameter des auf der MMC Karte ausgewählten Projekts angezeigt werden

Eine File Prüfung durch den **check file** Button sollte immer einen Ok Dialog ausgeben:

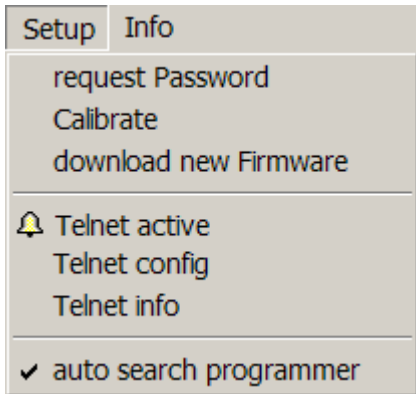




## UPP2-X Portabler In-Circuit Programmer mit GLCD

### Setup

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

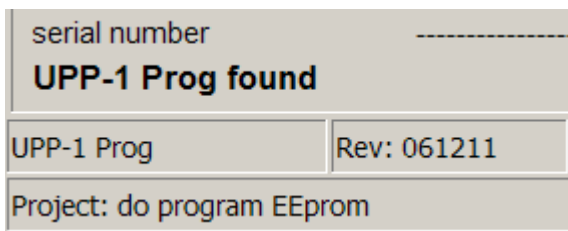


Dazu muss im Setup Menu der Dialog **request Password** aufgerufen werden. In diesem Dialog erscheint nun das Passwort des aktuellen Programmers:



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 Programmier gefunden, so wird das sofort hier angezeigt:





## 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\AVRProg.exe** ProjectName **-p -c**

**Achtung:** Die Kommando Zeilen Parameter unterstützen nur den Transparent Modus. Ein Zugriff auf Projekte von der Flash Karte ist nicht möglich!

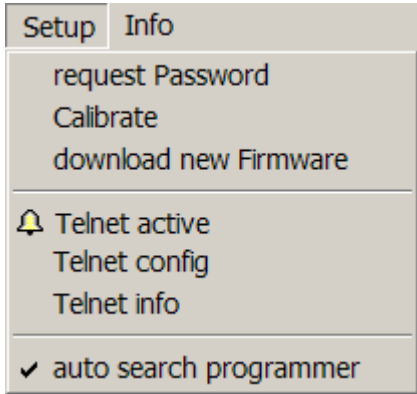
## Return Codes

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



## 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:

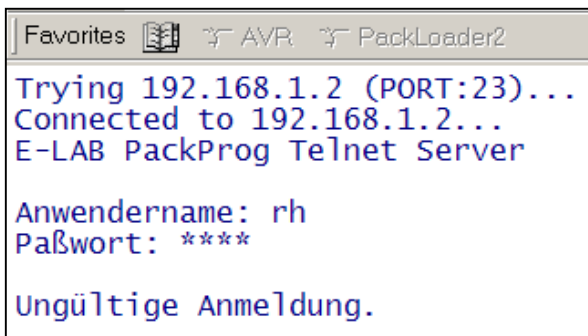


## Telnet config

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.

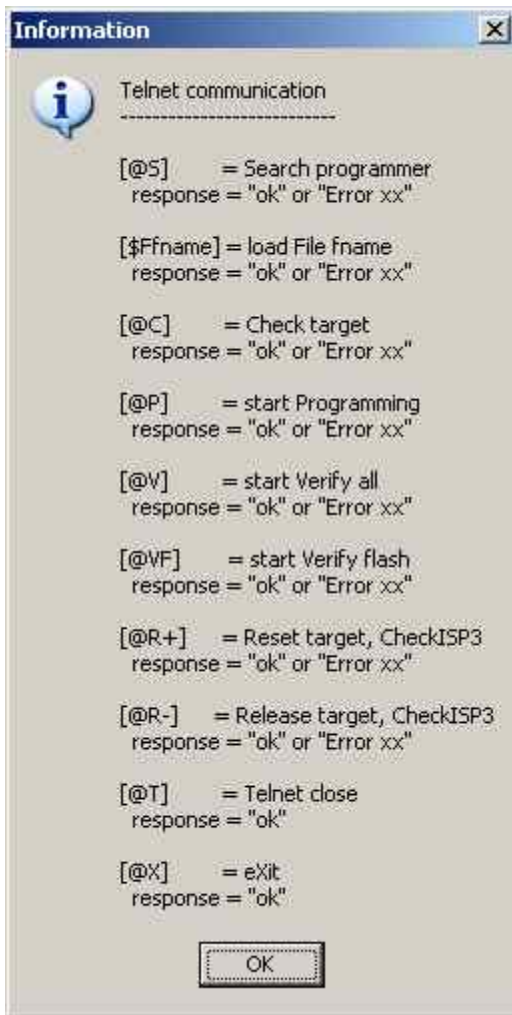


Eine ungültige Anmeldung mit einem Telnet Tool sieht so aus:



Im Telnet Mode arbeitet PackProg im Transparent Mode!

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 tief    |
| 3. 'FalseTyp | Target hat falsche oder unleserliche ID |
| 4. Protected | bei Verify                              |
| 5. notEmpty  | beim Programmieren                      |
| 6. Verify    | Verify Fehler beim Programmieren        |
| 7. File      | Datei nicht gefunden bei Kommando \$F   |
| 8. TimeOut   | Timeout bei Kommunikation               |
| 9. Comm      | Problem bei Kommunikation               |
| 10. noProg   | Programmer nicht gefunden               |
| 11. noProj   | kein Project ausgewählt                 |
| 12. Firmware | Programmer hat Firmware verloren        |

## Arbeiten Stand Alone

Das Gerät besitzt vier Tasten, **up**, **down**, **select** und **clear**. Ein grafisches Display dient der Benutzer Führung. Nach dem Power-On erscheint die Bereitschafts Meldung auf dem Display.



Das Flash Karten Symbol mit dem ? zeigt an, dass keine oder eine defekte Flash Karte gefunden wurde. Wenn jetzt die Select Taste gedrückt erfolgt die Aufforderung zum Einlegen der Flash Karte.



## Setup



Das Flash Karten Symbol zeigt dass die Karte vorhanden und ok ist. Mit der down Taste kann jetzt der Auswahlpfeil auf **Setup** gesetzt und dieses mit der select Taste angewählt werden.



Im Setup dialog können nun die Display Einstellungen verändert oder das System geprüft werden.

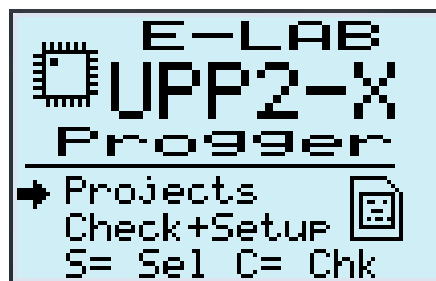
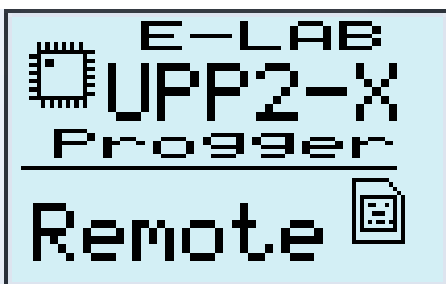
Mit der select Taste kann der **System Check** aufgerufen werden.

Dieser Prüft die Kalibrierung, die Flashkarte und alle Dateien auf der Karte.



Mit der select Taste kann der **Kontrast** oder **Helligkeit** ausgewählt und dann mit der up oder down Taste verändert werden. Die veränderten Werte können dann mit der select Taste abgespeichert oder mit der clear Taste verworfen werden.

Bei der Programmer Type **UPP2-X** sehen die Start Screens etwas anders aus:



## Reset

Durch gleichzeitiges Betätigen der up und down Taste wird ein Hardware Reset im Programmer durchgeführt.

## Start-Up

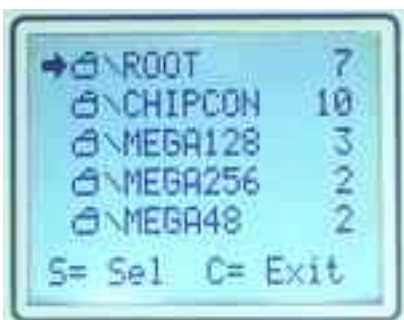
Da der UPP2-X mit einem internen Akku ausgerüstet ist, wird dieser immer nach dem Einschalten im Stand-alone Mode geprüft. Ist der Ladezustand des Akkus zu gering erscheint diese Anzeige:



Hier sollte unbedingt über die USB Buchse aufgeladen werden, entweder über den PC oder eines der enthaltenen Ladegeräte.

## Projects

Wird im Hauptmenu (Bereitschafts Anzeige) mit der select Taste der Dialog **Projects** aufgerufen so erscheint nach einem kurzen Speicher Test der Projekte Auswahl Dialog.



Wie nebenstehend ersichtlich ist, können mehrere Directories (Folders) existieren. Zumindest der ROOT Folder ist immer vorhanden. Innerhalb dieser Folder befinden sich die gespeicherten Projekte. Zeigt der Auswahlpfeil auf einen Folder, dann kann dieser mit der select Taste geöffnet werden und die enthaltenen Projekte werden aufgelistet.

Ein geöffneter Folder kann mit der clear Taste wieder geschlossen werden. Die Zahl neben dem Folder zeigt die Anzahl der in diesem Folder enthaltenen Projekte an. Sind mehr als 5 Folder vorhanden, so kann mit den up/down Tasten gescrollt werden.

Der UPP2 Programmer kann **eine** Directory Tiefe nutzen, das heisst sub-Directories in diesen Foldern sind nicht zulässig. Insgesamt können bis zu 64 Projekte und Folder verarbeitet werden.



Wurde mit der select Taste ein Folder geöffnet, so erscheint ein Dialog in dieser Art. Mit den up/down Tasten kann der Auswahl Pfeil auf- und abwärts bewegt werden. Sind mehr als 4 Projekte in diesem Folder gespeichert, dann wird mit den up/down Tasten auch gescrollt.

Mit der clear Taste wird der Folder wieder geschlossen.

Mit der select Taste wird das Projekt geladen, auf das der Auswahl Pfeil zeigt.

## Operations Auswahl



Die wesentlichen Parameter des aktuell geladenen Projekts werden dann in diesem Dialog angezeigt. Ganz oben steht der Datei Name. Rechts darunter wird die Spannung des Targets angezeigt und falls dieses vom Programmer mit Spannung versorgt wird, wird auch dessen aktuelle Stromaufnahme angezeigt.

Mit der clear Taste kehrt das Programm wieder in die Projekt Auswahl zurück.

Mit der select Taste kann eine von 4 Funktionen für dieses Projekt ausgewählt werden.





# UPP2-X Portabler In-Circuit Programmer mit GLCD

## Project Info

```
Project:
ABC.PAC   PDI
TARG XMEGA256A3U
FLASH: yes  Power
EEPRM: yes  none
USROW: yes
```

In den zwei Project Info Dialogen wird der Projekt bzw. Dateiname angezeigt. Daneben wird der Programmier Mode angezeigt, hier ist das PDI. Darunter befindet sich die Info über den CPU/Device Typ und was programmiert werden soll. Daneben wird angezeigt ob und was der Programmer an Spannung und maximalem Strom ausgeben soll.

```
Project:
ABC.PAC PDI
LOCK : yes
File Protection:
AES+PWD+Quantity
123 pcs left
```

Hier werden diverse Optionen angezeigt, z.B. AES Verschlüsselung und Stückzahlen Begrenzung.

Wenn der UPP2 eine Spannung an das Target ausgeben muss, so prüft er diese auch. Soll z.B. 4.0Volt ausgegeben werden und das Target hat durch eine eigene Versorgung 5.0Volt dann erscheint beim Programmieren, Verifizieren oder Löschen

eine Warnung: **Power Error.**

Diese kann mit der select Taste ignoriert werden und die Operation wird fortgesetzt.

## Program



Wurde mit der select Taste die Programmierung gestartet, so erfolgen zuerst ein paar Prüfungen. Die notwendige Stromversorgung wird geprüft und auch die ID des Prüflings, soweit vorhanden. Dann erfolgt ein Kommunikations Test. War alles in Ordnung dann erscheint die Programming Anzeige mit einem Fortschritts Balken, wobei zuerst das Flash angezeigt wird und dann das EEprom, falls vorhanden.

Treten Fehler auf, so werden diese entsprechend angezeigt, ansonsten erscheint obige ok Anzeige.

## Verify



Wurde mit der select Taste ein Verify gestartet, so erfolgen zuerst ein paar Prüfungen. Die notwendige Stromversorgung wird geprüft und auch die ID des Prüflings, soweit vorhanden. Dann erfolgt ein Kommunikations Test. War alles in Ordnung dann erscheint die Verify Anzeige mit einem Fortschritts Balken, wobei zuerst das Flash angezeigt wird und dann das EEprom, falls vorhanden.

Treten Fehler auf so werden diese entsprechend angezeigt, ansonsten erscheint obige ok Anzeige.



## Erase



Wurde ein Chip Erase ausgewählt erfolgt unmittelbar nach den obligatorischen Prüfungen eine ok oder fail Anzeige.

Treten Fehler auf so werden diese entsprechend angezeigt.

## Fehler Anzeige

Bei den **program**, **verify** oder **erase** Operationen sind diese Fehlermeldungen möglich:

- Power Down** zuerst wird die Target Spannung geprüft. Diese muss > 2.7V sein
- Power Error** wenn der Programmer das Target mit Spannung versorgen muss, wird diese geprüft.
- Wrong ID** wenn das Target eine auslesbare und verwendbare ID hat, wird diese geprüft.
- Protected** ein verify wird abgebrochen wenn das Target protected ist, auch bei einem erase Fehler.
- Not Empty** ein erase, entweder direkt oder in Verbindung mit program, kann diesen Fehler anzeigen.
- Verify Error** beim programmieren oder verifizieren kann dieser Fehler auftreten.

## Akku Status



Im Idle Mode kann über die **C** Taste der aktuelle Akku Status abgefragt werden.

## Shut Down

Werden die **C** Taste und die **S** Taste gleichzeitig gedrückt, dann geht der UPP2-X aus bzw. in Powerdown.

## Externe Hardware

### Schutz des Programmers gegen externe Kurzschlüsse und Überspannung

Alle neueren Programmer Typen sind gegen externen Dauerkurzschluss geschützt. Der Überspannungsschutz ist natürlich 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.

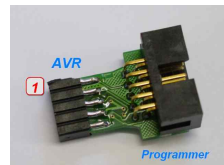
Grundsätzlich sollte beim Schaltungsdesign deshalb besonders darauf geachtet werden. Lasten kleiner 2kOhm und Kondensatoren grösser 100pF sollten nicht an die Programmierleitungen angeschlossen werden. Das gilt für die SPI-Programmierung genauso wie für die JTAG-Programmierung.

### Verschiedenen Adapter



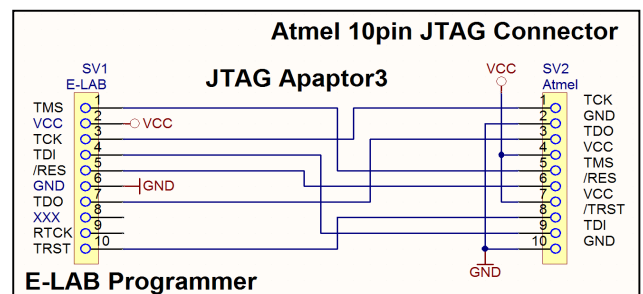
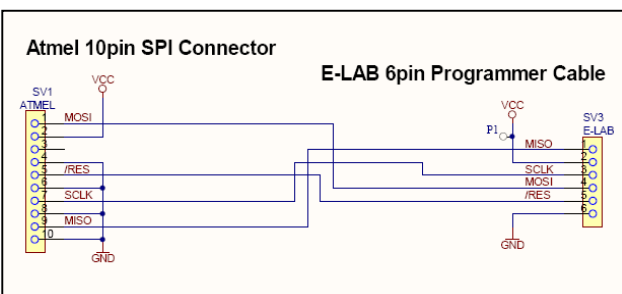
AVR  
1  
Programmer

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

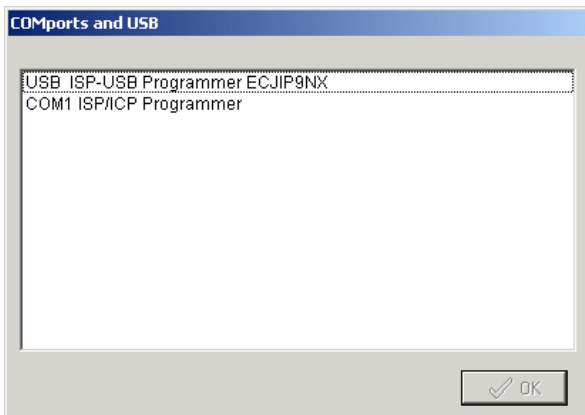


AVR  
1  
Programmer

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



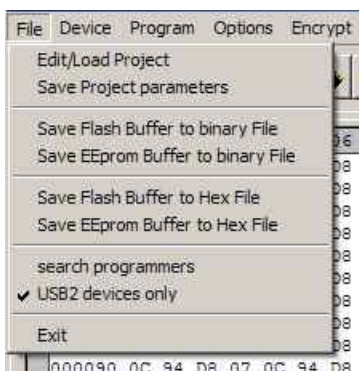
## MULTIPLE PROGRAMMER



Wenn ein PC Steuer Programm *AVRprog.exe* oder *PackProg.exe* mehrere USB Programmer findet oder wenigstens ein serieller und ein USB Programmer vorhanden sind, muss das gewünschte Programmier Gerät aus diesem Dialog ausgewählt werden.

Grundsätzlich wird nur ein serieller Programmer gesucht. Dann wird die Suche nach seriellen abgebrochen.

Unabhängig davon wie viele Programmer gefunden wurden kann immer nur einer gleichzeitig zum Programmieren benutzt werden.



Sollten weitere Programmer eingesteckt oder vorhandene entfernt werden, können dies mit dem Menüpunkt *File/search programmers* aktualisiert werden.

## BATTERIE BETRIEB

Der UPP2-X enthält einen eingebauten Akku Pack mit 3,7V/700mAh. Wenn das Gerät portabel betrieben wird, und das Target selbst keine eigene Stromversorgung hat, oder diese ist zu schwach um auch noch den Programmer versorgen zu können, kann das die interne Batterie leisten. Als Option gibt es den UPP2-X auch mit 1500mAh internem Akku.

Muss der UPP2-X das Target mit Spannung versorgen, so gewährleistet ein interner step-up Konverter eine Ausgangsspannung bis zu 5.5V, falls diese Option beim erstellen des Projekts gewählt wurde.

Der interne Akku wird wiederaufgeladen wenn der Programmer am USB Port am PC angeschlossen ist oder eines der externen Ladegeräte aktiv ist.

Im dem USB2-X Paket ist dazu ein mini AC zu USB Power Supply enthalten als auch ein Auto Konverter für den Zigaretten Anzünder (12V -> 5V USB).

## TARGET POWER SUPPLY

Alle USB Programmer können die Ziel CPU mit einer einstellbaren Spannung / Strom versorgen.

Da auch ein USB-Port oder Hub wenigstens 100mA/5V liefern, können auch diese benutzt werden. Beachten Sie, dass die nominellen 5V vielfach nur 4.8V oder weniger sind. Weiterhin hat der interne Spannungsregler einen Spannungsabfall von etwa 0.2V. Um dennoch die 5V erreichen zu können besitzen die Programmer Step-up Regler, sodass immer  $\geq 5V$  verfügbar sind.



## UPP2-X Portabler In-Circuit Programmer mit GLCD

Als weitere Option kann auch ein externes Netzteil mit 5V an die USB Buchse angeschlossen werden. Ein entsprechendes Mini-Netzteil ist Teil des Programmer Pakets. Mit dem internen Akku kann der Programmer immer bis zu 300mA liefern, solange dieser voll oder fast voll geladen ist. Dabei ist es gleichgültig ob ein externes Netzteil angeschlossen ist oder nicht.

o1	2o
o3	4o
o5	6o
o7	8o
o9	10o

Am Programmieranschluss des UPP1-X Gerätes kann auch eine externe kann an ein 5..9V/DC (bis 500mA) Netzteil angeschlossen werden.

Pin6 Ground

Pin10 can be used as the external positive supply input.

---

## 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 oder Windows7 möglich. Grundsätzlich sollten Hard/Software Entwicklungs-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.

### Windows Treiber

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

Für die 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, UPP1** und **UPP2**

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.

## Starteinstellungen

Drücken Sie eine Nummerntaste, um eine der Optionen unten auszuwählen:

Verwenden Sie die Nummerntasten oder die Funktionstasten F1-F9.

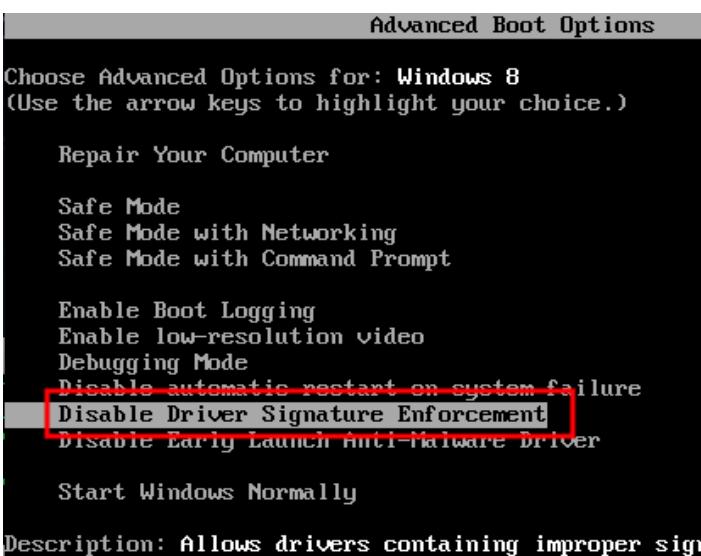
- 1) Debugmodus aktivieren
- 2) Startprotokollierung aktivieren
- 3) Video mit niedriger Auflösung aktivieren
- 4) Abgesicherten Modus aktivieren
- 5) Abgesicherten Modus mit Netzwerktreibern aktivieren
- 6) Abgesicherten Modus mit Eingabeaufforderung aktivieren
- 7) Erzwingen der Treibersignatur deaktivieren
- 8) Schutz des Antischadsoftware-Frühstarts deaktivieren
- 9) Automatischen Neustart bei Systemfehler deaktivieren

*press Key 7 !!!*

Drücken Sie zur Anzeige weiterer Optionen F10.  
Drücken Sie die EINGABETASTE, um zum Betriebssystem zurückzukehren.

## Schritt 5:

Darin nun die Startauswahl mit der Taste 7 bestätigen.



## Step6:

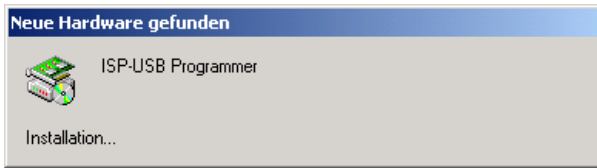
auswählen von **Disable Driver Signature Enforcement** und mit der Enter Taste Windows neu starten. Bitte beachten dass bei WIN8/64 die betreffenden EXEs im WIN7 Mode ausgeführt werden müssen.

Treiber ohne Fehlermeldung installiert werden.

Jetzt kann der AVRco bzw. der AVRprog

## Weitere 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.

## Firmware Update

Neuere E-LAB Programmier Geräte 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.

USB Version Datei: *yymmdd\_UPP2\_X.pupd* oder vom WEB in Datei: *UPP2\_Xupdate.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.

### Dialog ISP/UPP Firmware Loader



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 Doppelclick 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 Programmiergerät 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.

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 **erzwungen** werden.

Bei den UPP1-X Geräten befindet sich eine RESET Taste auf der hinteren Frontseite des Gerätes.



1. USB und Netzteil Buchse
2. Micro-SD Karten Schacht
3. RESET Taste

Diese RESET Taste hat zwei Funktionen:

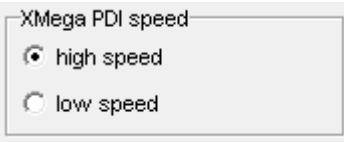
1. ein kurzer Druck auf diese Taste führt einen **Hardware Reset** aus.
2. ein längerer Druck auf diese Taste (>1sec) führt ebenfalls ein **Hardware Reset** aus mit anschliessendem Einsprung in den **Zwangs Download Mode** wie oben beschrieben.

## 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.

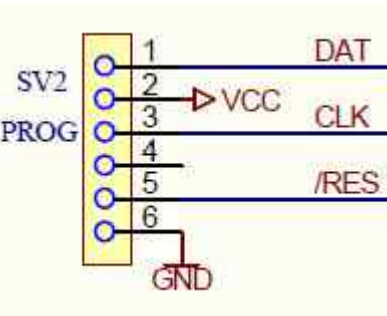


Bei älteren XMegas, z.B. XMega128A1, oder langen Programmierleitungen kann es zu Fehlern kommen. Atmel empfiehlt hierbei die Clockrate des PDI Interfaces herabzusetzen. Dies geschieht unter "Options/Programer options"

### TINY4-5-9-10-20

Diese Tinsys werden mit 5V über drei Pins programmiert.

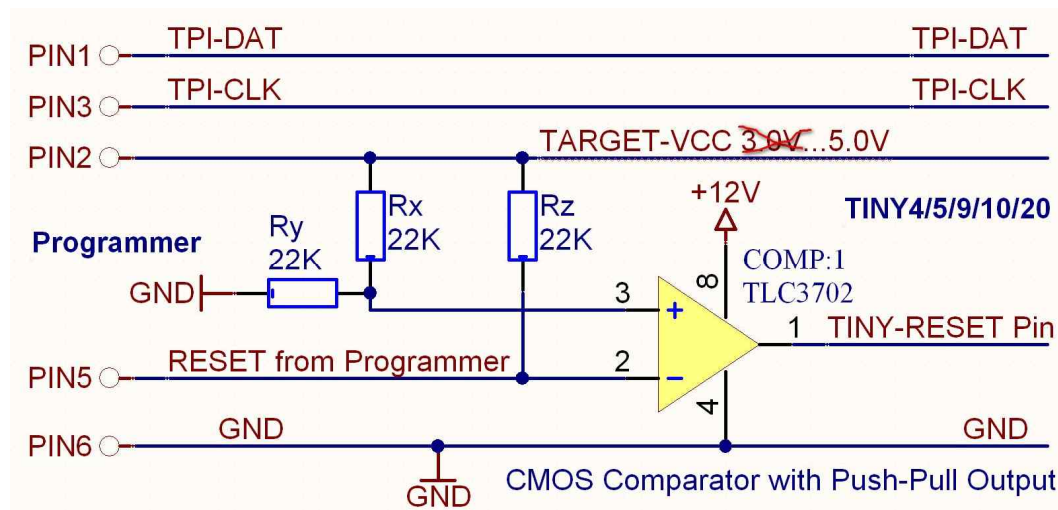
Der Programmier Mode nennt sich TPI. Da der Programmier Stecker auch für den SPI Mode verwendet wird, weicht dieser von der Atmel Belegung ab. Die Last am TPI\_DAT Pin sollte nicht kleiner als 80kOhm sein.



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-programmieren 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 n den RESET Pin 12V angelegt werden müssen.

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



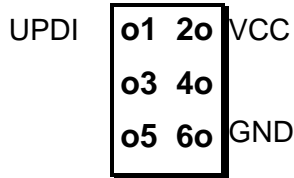


## UPDI Tiny und UPDI Mega

### 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.

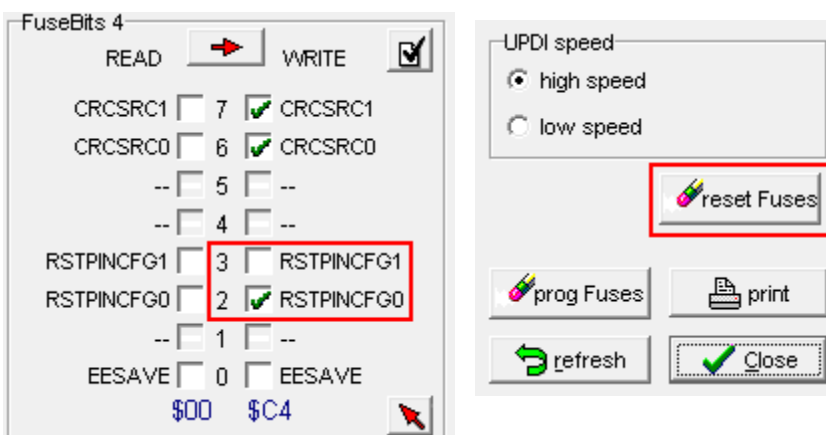
Dieser HVP Adapter ist von E-LAB erhältlich. Er braucht keine weitere Spannungs Versorgung.



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.





## Chipcon

### Allgemeines

Die aktuelle UPP 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	<b>01 20</b>	VCC
(P2_2) CLK	<b>03 40</b>	DATA (P2_1)
/RESET	<b>05 60</b>	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.

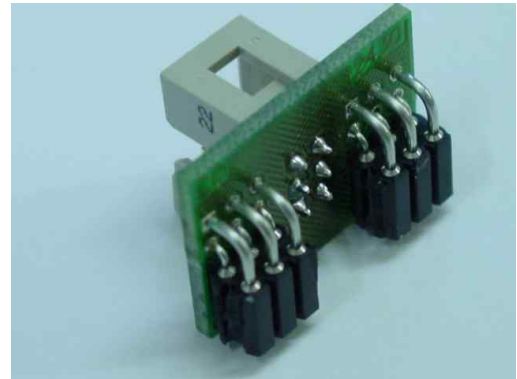
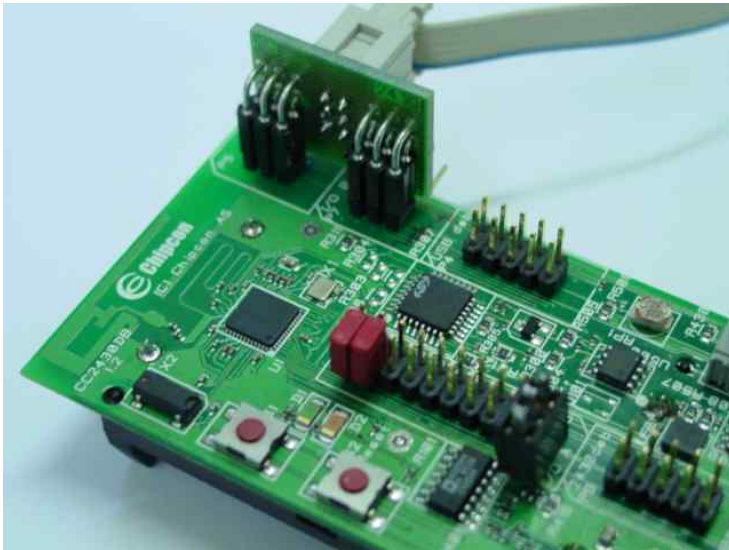
Eine falsche ID sollte niemals ignoriert werden sondern der Grund dafür muss eingehend untersucht werden.

## Chipcon Evaluation Boards

### CC2430DB

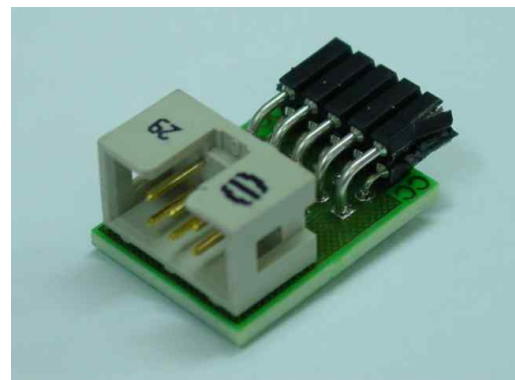
Dieses Board hat zwei 6pol Pfosten, die es erlauben einen externen Programmer anzuschliessen. Da der UPP Programmer deshalb nicht direkt auf dieses Board gesteckt werden kann, gibt es als optionales Zubehör für den UPP ein kleines Adapter Board das separat bestellt werden muss. (Ord #2109)

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



Dieses Board hat einen 10pol Pfosten für den Anschluss eines externen Programmers. Da der UPP Programmer nicht direkt auf dieses Board gesteckt werden kann, gibt es als optionales Zubehör für den UPP 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 UPP 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 PROGRAMMIERGERÄTE



### ISP3-X

Der Programmer ISP3-X arbeitet ausschliesslich mit dem PC zusammen und ist nicht portabel.

Deshalb ist er das preiswerteste Gerät unter den E-LAB Programmern.

Der Anschluss an den PC erfolgt über USB2.



### UPP1-X

Der universelle unter den portablen E-LAB Programmern.

Auf der Flashkarte können bis zu 16 Projekte abgelegt werden.

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

Der Anschluss an den PC erfolgt über USB2.

Hervorragend für den Service und auch die Kleinserien Produktion geeignet.



### 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-USB In-Circuit Programmer mit USB-2



### 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 mit Drahtfernsteuerung Version **PR**





**Notizen**





## Notizen





## 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)

---