

# **PBrenner V 1.8**

## **Steuersoftware für Parallelport-PIC-Brenner unter Windows**

### **Kurzanleitung**

## Inhalt:

---

1	Einleitung .....	3
2	Voraussetzungen .....	3
3	Installation.....	3
3.1	Windows95/98 .....	3
3.2	WindowsNT/2000 .....	4
4	Grundlagen der Parallelportbrenner .....	4
5	Start des Programms .....	5
5.1	Fehlender SmallPort-Treiber.....	5
5.2	Fehlender Brenner.....	5
6	Hardwareeinstellung .....	6
7	PIC-Typ einstellen.....	8
8	Hex-File-laden.....	8
9	Konfiguration des PIC .....	8
10	Brennen des PIC.....	9
11	Zusätzliche Anzeigefenster .....	9
11.1	Grafische Speicheranzeige .....	10
11.2	Hexdatei-Betrachter .....	11
11.3	Assembler-Code-Betrachter .....	12
12	mögliche Probleme und Lösungen .....	12
12.1	kann ich den PIC16F83 brennen?.....	12
12.2	kann ich den PIC16C84 brennen? .....	12
12.3	Autodetect meldet, das kein PIC gefunden wurde.....	12
12.4	Autodetect meldet einen PIC16F87x als PIC16F84 .....	13
12.5	Autodetect meldet den richtigen PIC-Typ aber das Brennen schlägt fehl .....	13
12.6	Was bewirken alle diese Einstellungen im PIC-Setup-Bereich des Programmfensters? .....	13
12.7	Unerklärliche Brennabbrüche und Fehler beim Vergleichen auf schnellen Rechnern 13	
12.8	Der Brenner wird nicht erkannt.....	13
13	Kommandozeilenparameter .....	13
14	Bekannte Probleme.....	14

---

## Abbildungen:

---

Abbildung 1	Fehlermeldung bei fehlendem Treiber .....	5
Abbildung 2	Fehlermeldung bei fehlendem Brenner .....	5
Abbildung 3	Programmfenster bei fehlendem Brenner .....	6
Abbildung 4	Setup-Fenster.....	7
Abbildung 5	Setup-Fenster - Brenngeschwindigkeit .....	7
Abbildung 6	Falsch eingestellter PIC-Typ.....	8
Abbildung 7	grafische Speicheranzeige .....	10
Abbildung 8	HEX-File-Betrachter .....	11
Abbildung 9	Assembler-Betrachter .....	12

---

# 1 Einleitung

Das Programm PBrenner dient zum komfortablen Programmieren von PIC-Microchip-Prozessoren der Serien PIC16F8xx. Es liest Intel-Hex-Dateien wie sie z.B. vom Programm MPLAB erzeugt werden, und brennt diese in den Flash-Speicher des PICs. Die Version 1.8 unterstützt folgende PIC-Prozessoren:

Typ	Programmspeicher (Worte)	RAM (Byte)	Daten-EEPROM (Byte)
PIC16F84	1024	68	64
PIC16F84A	1024	68	64
PIC16F627	1024	224	128
PIC16F628	2048	224	128
PIC16F73	4096	192	-
PIC16F74	4096	192	-
PIC16F76	8192	368	-
PIC16F77	8192	368	-
PIC16F870	2048	128	64
PIC16F871	2048	128	64
PIC16F872	2048	128	64
PIC16F873	4096	192	128
PIC16F874	4096	192	128
PIC16F876	8192	368	256
PIC16F877	8192	368	256

## 2 Voraussetzungen

Das Programm PBrenner benötigt als Hardware einen Parallelport-Brenner wie den Brenner3 ([www.sprut.de/electronic/pic/projekte/brenner3/](http://www.sprut.de/electronic/pic/projekte/brenner3/)), den Brenner5 ([www.sprut.de/electronic/pic/projekte/brenner5/](http://www.sprut.de/electronic/pic/projekte/brenner5/)) oder einen Brenner nach dem Tait-Prinzip.

PBrenner akzeptiert zu brennende PIC-Programme ausschließlich im Intel-Hex8-Format. Die verwendete PIC-Entwicklungsumgebung (z.B. MPLAB) ist dementsprechend einzustellen.

## 3 Installation

Das Programm PBrenner steht als ZIP-Archiv zum Download bereit. Dieses Archiv mit dem Namen PBRENNER.ZIP enthält die folgenden Dateien.

- PBrennerxx.exe Das Hauptprogramm in der aktuellen Version
- smport.vxd SmallPort-Treiber für Win95/98
- smport.sys SmallPort-Treiber für WinNT/2000
- readme.txt Kurzanleitung
- PBrennerxx.doc diese Word-Datei

Bevor das Programm PBrenner benutzt werden kann, ist der richtige SmallPort-Treiber in das System-Verzeichnis des Betriebssystems zu kopieren.

### 3.1 Windows95/98

Für die Arbeit unter den Consumer-Betriebssystemen Windows95/98 ist der Treiber smport.vxd in das Windows-Systemverzeichnis zu kopieren. Dieses Verzeichnis heißt standardmäßig:

C:\WINDOWS\SYSTEM\

Nach einem Windows-Neustart ist der Treiber dann installiert.

Wurde Windows unter einem anderen Pfad als C:\WINDOWS\ installiert, ist der System-Pfad dementsprechend anders.

### 3.2 WindowsNT/2000

Für die Arbeit unter den Profi-Betriebssystemen WindowsNT/2000 ist der Treiber smport.sys in das Windows-Systemverzeichnis zu kopieren. Dieses Verzeichnis heißt standardmäßig unter WinNT:

C:\WINNT\SYSTEM32\

*Dafür sind auf dem Rechner vorübergehend Administrator-Rechte nötig.*

Nach einem Windows-Neustart ist der Treiber dann installiert.

Wurde Windows unter einem Anderen Pfad als C:\WINNT\ installiert, ist der System-Pfad dementsprechend anders.

### 3.3 Windows XP

Für Windows XP habe ich noch keine lauffähigen Treiber. Somit kann die Software unter Windows XP **NICHT** genutzt werden.

## 4 Grundlagen der Parallelportbrenner

PICs werden über eine serielle Datenverbindung programmiert (ICSP – In Circuit Serial Programming), die aus einer Taktleitung und einer Datenleitung besteht.

Außerdem benötigt der PIC eine 5-V-Betriebsspannung und eine 12-V-Programmierspannung.

Der fünfte und letzte Kontakt zum PIC ist die Masseverbindung.

Einzelne PINs des PC-Druckerports sind beim Parallelportbrenner für diese Funktionen zuständig:

- D0 schreibt auf die Datenleitung
- D1 Taktleitung
- D2 schaltet die 5V ein
- D3 schaltet die 12V ein
- ACK liest von der Datenleitung

Die Leitungen D0, D1 und ACK werden oft mit einem Schaltkreis verstärkt. Dabei werden aber je nach Brenner zwei unterschiedliche Typen verwendet

- der nichtinvertierende 7407 oder
- der invertierende 7406

Die Leitungen D2 und D3 steuern elektronische Schalter an, die die 5- und 12-V-Spannungen zum PIC schalten. Als Schalter werden je nach Brennertyp

- zwei pnp-Transistoren (invertierend) oder
- ein 4066-Schaltkreis (nichtinvertierend)

verwendet.

Damit ergeben sich insgesamt 4 unterschiedliche Hardwarevarianten für den Brenner. Für die korrekte Funktion des Programms PBrenner ist es unbedingt erforderlich, die verwendete Hardwarevariante zu kennen.

Beim Erststart geht das Programm von einem angeschlossenen Brenner1 oder Brenner3 aus (7407 und pnp-Transistoren), der einem Tait-Brenner-Typ-3 entspricht. Ist ein anderer

Brenner angeschlossen, so muß nach dem Programmstart die Hardwareeinstellung im Programm korrigiert werden, bevor ein PIC in den Brenner eingesetzt wird.

Der Brenner verfügt über eine 5-polige ICSP-Steckverbindung, die alle 5 Brenner-Signale (12V, 5V, Masse, Daten, Takt) führt. Hier kann ein Adapter mit Fassung für einen PIC oder über ein Kabel ein bereits in eine Schaltung eingebauter PIC zum Brennen angeschlossen werden.

## 5 Start des Programms

Beim Programmstart öffnet PBrenner den installierten SmallPort-Treiber, sucht und findet den Brenner und meldet sich dann mit seinem Programmfenster. Falls dabei aber etwas nicht funktioniert, kann es zu einer Fehlermeldung kommen.

### 5.1 Fehlender SmallPort-Treiber

Das Programm PBrenner benötigt den SmallPort-Treiber von Alexander Weitzman, der im ZIP-Archiv beiliegt, und wie oben beschrieben in das System-Verzeichnis kopiert werden muß. Falls das Programm den Treiber nicht findet, meldet es sich mit folgendem Fenster:

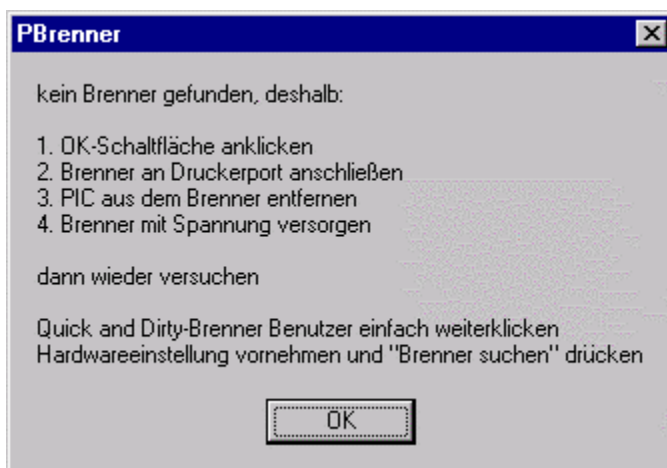


**Abbildung 1 Fehlermeldung bei fehlendem Treiber**

Eine Arbeit mit dem Programm ist nicht möglich, bis der Treiber installiert wurde.

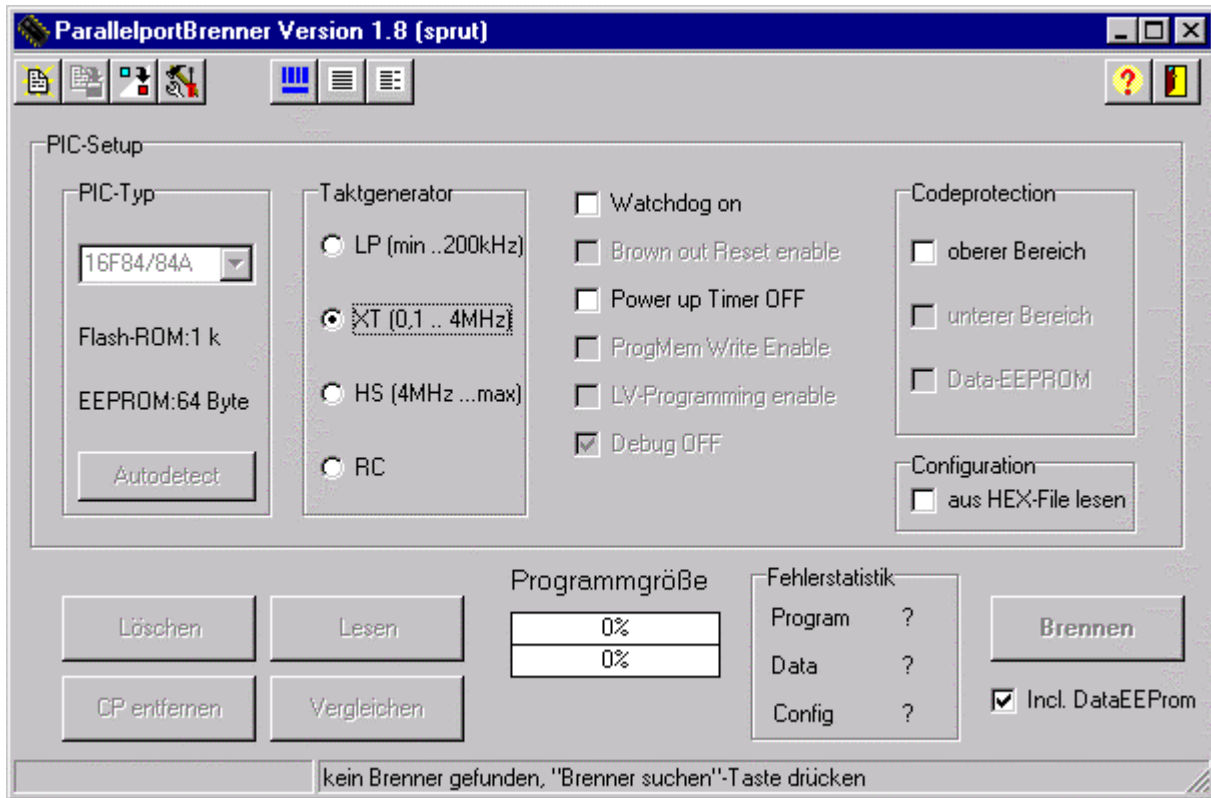
### 5.2 Fehlender Brenner

Beim Programmstart überprüft PBrenner die Druckerports auf einen angeschlossenen Brenner hin ab. Deshalb sollte der Brenner zu diesem Zeitpunkt an das Druckerport angeschlossen und mit einer Spannungsquelle verbunden sein. Falls das Programm keinen Brenner findet, meldet es sich mit folgendem Fenster:



**Abbildung 2 Fehlermeldung bei fehlendem Brenner**

Nach dem klicken auf „OK“ startet das Programm, verweigert aber die Arbeit, bis ein Brenner gefunden wurde. Das erkennt man an den deaktivierten Schaltflächen:



**Abbildung 3 Programmfenster bei fehlendem Brenner**

Der Anschluß des Brenners und seine Stromversorgung muß dann überprüft werden. Anschließend kann mit der „Brenner suchen“-Taste (3. Taste in der oberen Tastenleiste) der Brenner erneut gesucht werden.

**++ACHTUNG++**

Falls sich ein PIC in der Fassung des Brenners befindet, kann das die automatische Brennersuche behindern. Deshalb den beim Programmstart nie einen PIC im Brenner belassen.

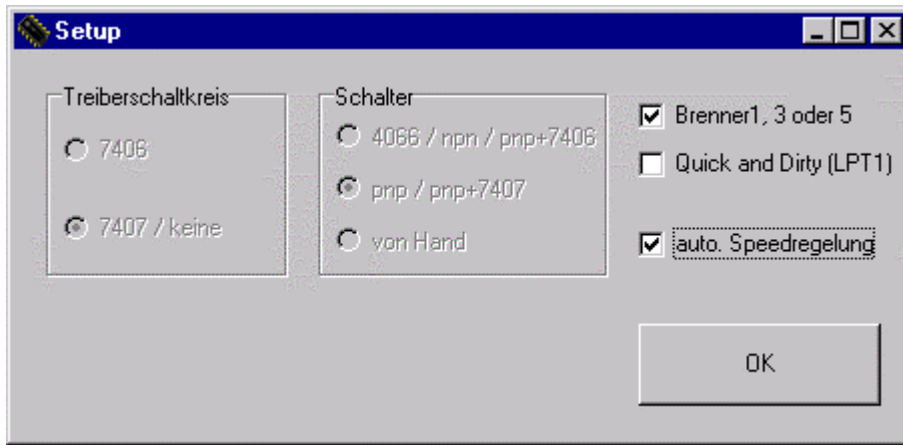
**+++Quick and Dirty-Brenner++**

„Quick and Dirty-Brenner“ (QaD) können nicht automatisch erkannt werden. Falls ein QaD-Brenner verwendet wird, so ist dieser Typ bei „Hardwareeinstellungen“ auszuwählen, und danach die „Brenner suchen“-Taste zu drücken. Das Programm glaubt dann blind an einen QaD-Brenner am Port LPT1. Das Port LPT2 wird für QaD-Brenner nicht unterstützt.

## 6 Hardwareeinstellung

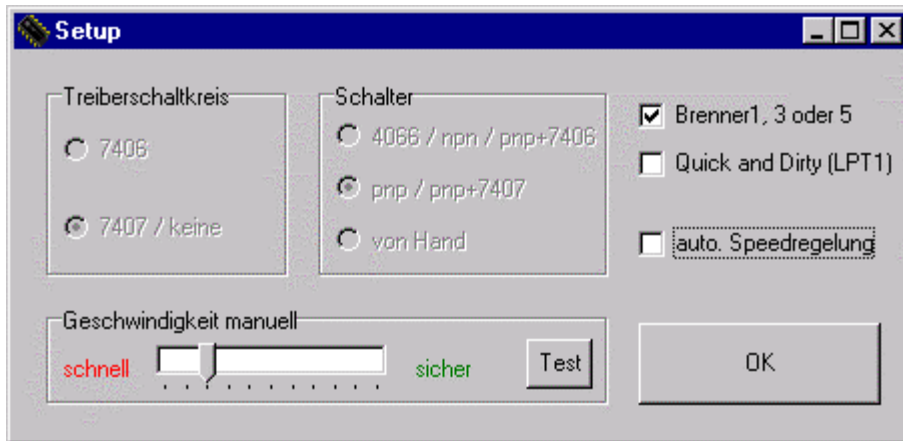
Wie oben erwähnt, gibt es unterschiedliche Varianten von Paralleport-Brennern. Im Setup-Fenster kann jeder Nutzer seinen eigenen Brennertyp auswählen. Standardmäßig ist der „Brenner1 oder 3“ eingestellt. Entfernt man das „Brenner1 oder 3“-Häkchen, lassen sich andere Treiberschaltkreise und Spannungsschalter auswählen.

Während die Auswahl des richtigen Treibers nur einen flüchtigen Blick auf die Leiterplatte erfordert, ist die Wahl des richtigen Schaltertyps schon schwieriger. Am Einfachsten ist es hier die beiden oberen Einstellungen auszuprobieren. Wenn nach dem Schließen des Setup-Fensters die Programmierspannungs-LED (meist gelb) ausgeht, hat man den richtigen Schaltertyp gefunden.



**Abbildung 4 Setup-Fenster**

Das Programm stellt eine (hoffentlich) optimale Brenngeschwindigkeit ein. Diese Geschwindigkeit ist vom PC und vom Aufbau des Brenners abhängig, aber nicht vom PIC. Wenn man das Kästchen „auto.Speedregelung“ deaktiviert, kann man manuell eine Geschwindigkeit wählen. Das ist nur bei Brennproblemen auf schnellen PCs ratsam.



**Abbildung 5 Setup-Fenster - Brenngeschwindigkeit**

Der Testbutton stellt den Wert ein, den das Programm am jeweiligen PC empfiehlt. Bei Brennproblemen kann man eine langsamere probieren. Die Einstellung bleibt für den nächsten Programmstart gespeichert.

Falls beim Brennen keinerlei Probleme auftreten, kann man den Schieberegler auf ‚schnell‘ stellen. Dadurch erhöht die Geschwindigkeit insbesondere dann, wenn nur einige Bytes im PIC geändert werden müssen.

Typische Brenngeschwindigkeit für den PIC16F84 auf einem 700-MHz-PC:

Geschwindigkeit	komplett brennen	1 Byte korrigieren
Minimalgeschwindigkeit:	60 Sekunden	15 Sekunden
Normalgeschwindigkeit:	30 Sekunden	5 Sekunden
Maximalgeschwindigkeit:	20 Sekunden	2 Sekunden

Nach dem Schließen des Setup-Fensters kann der PIC in die Fassung eingesetzt werden. Benutzer des Brenners1, des Brenners3 oder eines Tait-Brenners-Typ-3 brauchen dieses Fenster normalerweise nicht aufzurufen.

Beim Schließen des Setup-Fensters speichert PBrenner die Hardwareparameter in der Datei 'C:\PBRENNER.INI' und liest sie beim nächsten Programmstart aus dieser Datei aus. Dadurch müssen die Hardwareeinstellungen nur ein mal vorgenommen werden.

## 7 PIC-Typ einstellen

PBrenner unterstützt 15 unterschiedliche PIC-Typen, die unterschiedlich programmiert werden. Deshalb ist vor dem eigentlichen Brennen der richtige PIC-Typ auszuwählen. Am einfachsten erfolgt das durch Drücken von „Autodetect“. Alternativ kann der Typ manuell im Typ-Feld oberhalb des Autodetect-Schaltfeldes ausgewählt werden.

Die Autodetect-Funktion kann beim „alten PIC16F84“ (ohne A) ohne eindeutiges Ergebnis enden. Das meldet das Programm aber, und wählt dann doch den PIC16F84.

Vor jedem Versuch des Bedieners auf den PIC zuzugreifen (Lesen, Löschen, CP-entfernen, Brennen) überprüft das Programm die richtige Einstellung des PIC-Typs. Bei falscher Einstellung verweigert das Programm die Arbeit mit folgender Fehlermeldung:



**Abbildung 6 Falsch eingestellter PIC-Typ**

++QaD-Brenner++

Bei Quick and Dirty-Brennern ist die PIC-Typ-Abfrage nicht möglich. Der Bediener ist selbst für die richtige Einstellung verantwortlich.

## 8 Hex-File-laden

Mit dem HEX-File-Laden-Button (linker Button in der oberen Buttonleiste) wird ein Dialogfenster geöffnet, mit dem man das in den PIC zu brennende HEX-File auswählt. Nach dem Schließen dieses Dialogs zeigen Fortschrittsbalken an, wie weit dieses Programm den zu brennenden PIC füllen wird. Ist ein Balken rot, so paßt das Programm aus dem HEX-File nicht in den PIC. Ein gelber Balken bedeutet, daß der PIC vollständig gefüllt wird.

++ACHTUNG++

Manchmal enthält das HEX-File auch Konfigurationseinstellungen für den PIC. Sollen diese benutzt werden, ist **vor dem Laden** das Feld „Configuration – aus Hex-File lesen“ zu aktivieren (Häkchen). Die manuellen Konfigurationseinstellungen werden solange deaktiviert.

## 9 Konfiguration des PIC

Falls die Konfiguration des Brenners nicht schon mit dem Hex-File geladen wurde, kann sie von Hand eingestellt werden.

Dazu darf das Feld „Configuration – aus Hex-File lesen“ nicht aktiv sein (kein Häkchen). Alle für den jeweiligen PIC-Typ relevanten Einstellungen stehen zur Verfügung eine Beschreibung der Konfigurationsoptionen findet sich unter [www.sprut.de/electronic/pic/config/config.htm](http://www.sprut.de/electronic/pic/config/config.htm).



## 10 Brennen des PIC

Sollen auch Daten in den Daten-EEPROM gebrannt werden, muß das Feld „Incl. DataEEPROM“ aktiviert sein (Häkchen). Das ist beim Programmstart standardmäßig der Fall.

Das Drücken auf die Brenner-Taste bewirkt ein **automatisches erneutes Laden** der HEX-Datei und startet dann den Brennvorgang, dessen Fortschritt mit dem oberen Fortschrittsbalken angezeigt wird. Tritt ein Brennfehler auf, bricht das Programm mit einer Fehlermeldung ab. Nach einem erfolgreichen Brennen blinkt für 2 Sekunden ein grünes OK-Zeichen.

Das automatische erneute Lesen der Hex-Datei vor dem Brennen ermöglicht eine schnelle Zykluszeit beim Entwickeln eines Programms. In einem Windowsfenster entwickelt man z.B. mit MPLAB ein Programm, und speichert dabei ständig neue Programmversionen (HEX-File) unter dem selben Filenamen zum Brennen. In einem zweiten Windowsfenster läuft PBrenner, das durch einfaches drücken der „Brennen“-Taste stets die neueste Hex-File-Version lädt und brennt, ohne daß diese aufwendig in einem Dialogfenster geladen werden muß.

## 11 Zusätzliche Anzeigefenster

Im Hauptprogrammfenster befinden sich drei zusätzliche Schaltflächen, mit denen spezielle Fenster geöffnet werden können. Diese Fenster enthalten

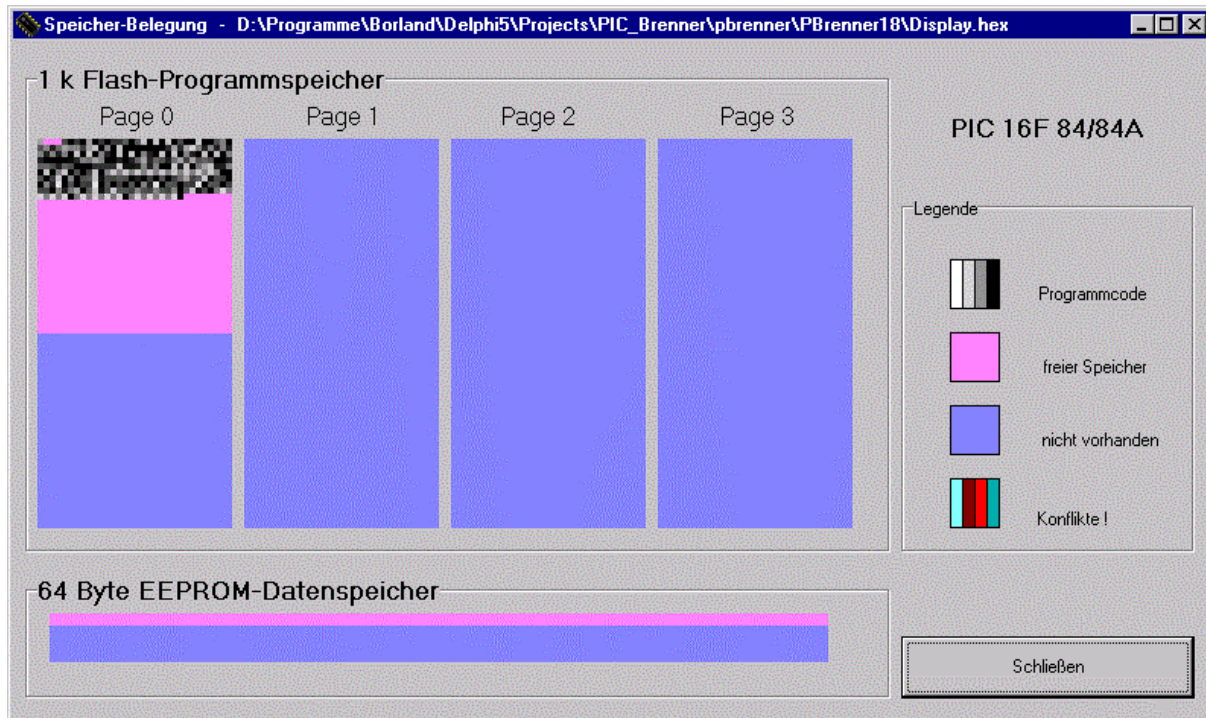
- Eine grafische Anzeige der Speicherauslastung
- Eine Anzeige des geladenen HEX-Files
- Eine Anzeige des geladenen HEX-Files als Assemblercode

Zwischen den Fenstern kann hin und her gewechselt werden. Wird im Hauptfenster ein neues HEX-File geladen, oder der Prozessortyp gewechselt, so wird ein Anzeigefenster erst dann automatisch aktualisiert, wenn man wieder zu ihm wechselt.

Es ist auch möglich mehrere Fenster des selben Typs zu öffnen.

Die dargestellten Inhalte beziehen sich immer auf das geladene HEX-File, und nicht auf den realen Inhalt des PIC. Will man sich den Inhalt eines PIC anzeigen lassen, so muß man ihn auslesen, und seinen Inhalt als HEX-File speichern. Dieses Hex-File muß dann erst wieder geladen werden, bevor die Anzeige benutzt werden kann.

## 11.1 Grafische Speicheranzeige



**Abbildung 7 grafische Speicheranzeige**

Die grafische Speicheranzeige visualisiert den benutzten und den zur Verfügung stehenden Speicher im PIC. Die angezeigten Speicherbereiche entsprechen immer einem 8-k-Flash-Programmspeicher und 256 Byte EEPROM.

Ist ein ‚kleinerer‘ PIC ausgewählt, so wird der physisch nicht vorhandene Speicher blau dargestellt. Unbenutzte Speicherbereiche sind pinkt eingefärbt. Geladener Programmcode wird in allen Graustufen (von weiß bis schwarz) dargestellt.

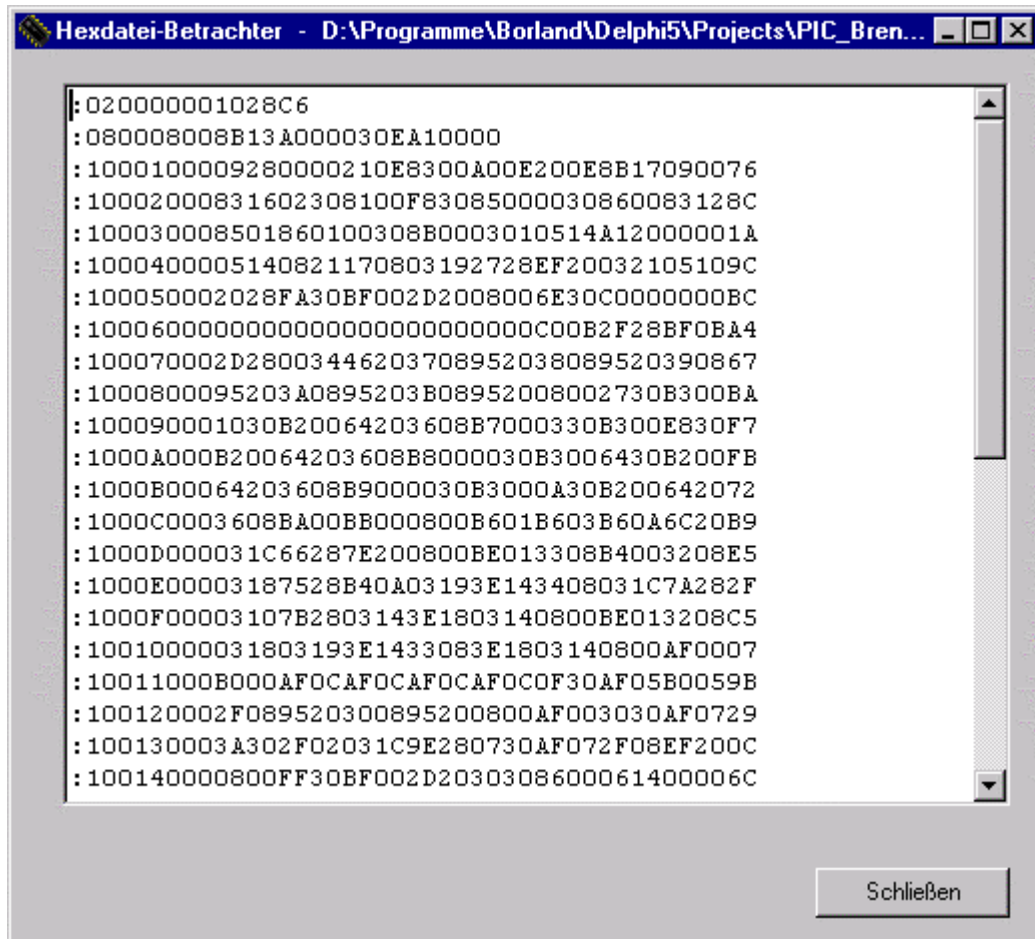
Andere Farben weisen auf Speicherkonflikte hin, z.B. auf ein Programm, das größer ist, als der Speicher des PIC.

Die oben stehende Abbildung zeigt eine HEX-File für einen PIC16F84. Dieser besitzt nur 1 k Programmspeicher, was einer halben Speicher-Page entspricht. Die anderen nicht vorhandenen 3,5 Pages sind blau eingefärbt (nicht vorhanden).

In der Page 0 ist das geladene Programm als graues Punktemuster zu erkennen. Die pinkfarbenen Bereiche stellen den noch freien Speicher dar.

Für den EEPROM enthält das geladene HEX-File gar keine Daten.

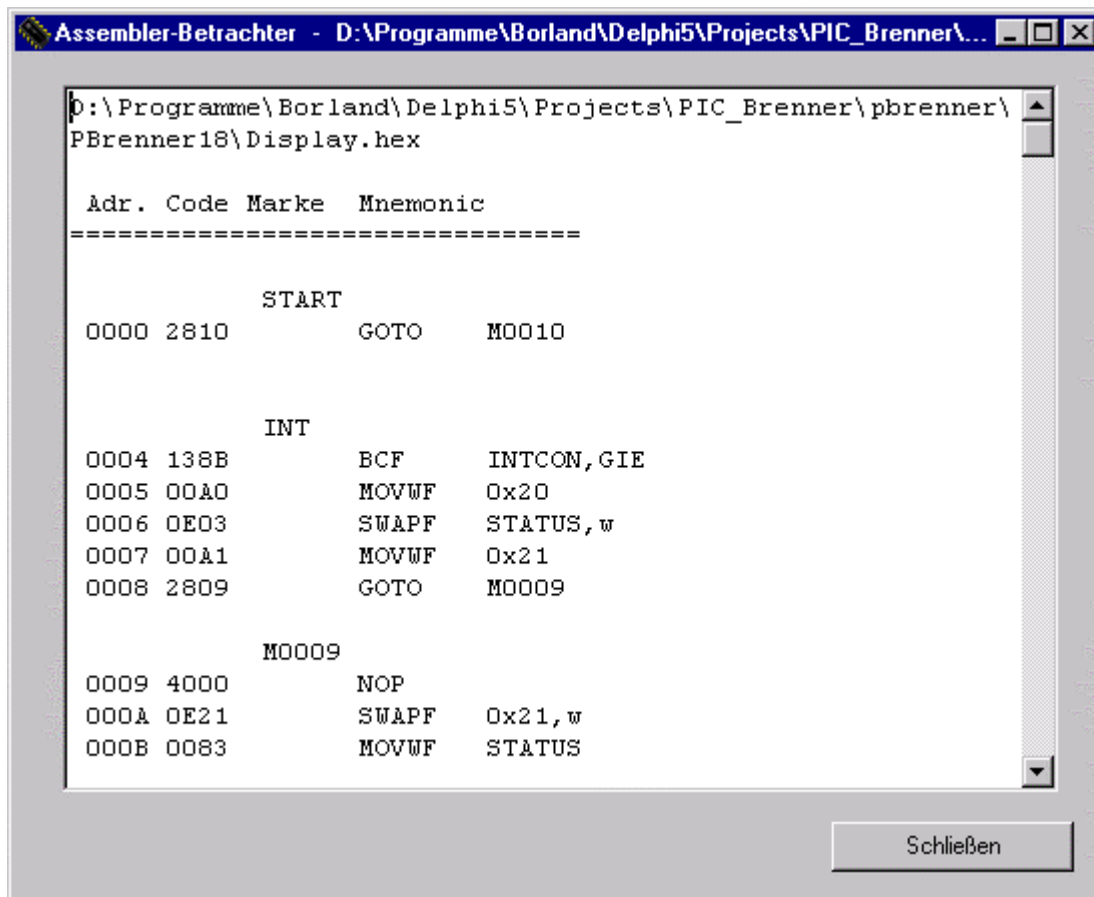
## 11.2 Hexdatei-Betrachter



**Abbildung 8 HEX-File-Betrachter**

Der HEX-File-Betrachter stellt das geladene HEX-File in einem Fenster dar

### 11.3 Assembler-Code-Betrachter



**Abbildung 9 Assembler-Betrachter**

Beim Klicken auf das „Assembler-Code anzeigen“-Feld, wird der HEX-Code des geladenen Files zunächst reassembliert. Anschließend wird er in einem Fenster dargestellt.

## 12 mögliche Probleme und Lösungen

### 12.1 kann ich den PIC16F83 brennen?

Ja. Der PIC16F83 wird vom Programm als PIC16F84 behandelt und kann somit gebrannt werden. Allerdings ist der Programmspeicher des PIC16F83 nur halb so groß wie der des PIC16F84. Der Nutzer muß deshalb darauf achten, daß das zu brennende Programm nicht zu groß ist, da das Brennprogramm das in diesem Sonderfall nicht berücksichtigt.

### 12.2 kann ich den PIC16C84 brennen?

Theoretisch ja, ich habe es aber nie probiert. Der PIC16C84 wird vom Programm wahrscheinlich unter „Protest“ als PIC16F84 eingestuft und kann somit gebrannt werden. Es läßt sich aber kaum vorhersehen, ob das Brennen der Konfiguration, das Löschen des Chips oder das Entfernen von „Code Protection“ funktionieren. Außerdem muß das Häkchen für die Power-On-Timer-Einstellung invers gesetzt werden.

Falls es jemand probiert, bitte mailt mir die gemachten Erfahrungen. (sprut@sprut.de)

### 12.3 Autodetect meldet, das kein PIC gefunden wurde

Vielleicht ist die 12-V-Spannung zu klein. Falls nicht sollte noch mal der richtige Sitz des PIC in der Fassung geprüft werden.

## 12.4 Autodetect meldet einen PIC16F87x als PIC16F84

Das ist ein Kommunikationsproblem, das auf einen unsauberen Aufbau der Hardware schließen läßt. Wahrscheinlich laufen die einzelnen Drähte des Brenneranschlußkabels eng beieinander parallel ohne Schirmung oder schirmende Masseleitungen. Vielleicht ist das Anschlußkabel des Brenners auch zu lang.

## 12.5 Autodetect meldet den richtigen PIC-Typ aber das Brennen schlägt fehl

Der PIC ist wahrscheinlich codeprotectet. Das läßt sich durch Drücken von „CP-entfernen“ beheben. Danach sollte sich der PIC wieder normal brennen lassen.

## 12.6 Was bewirken alle diese Einstellungen im PIC-Setup-Bereich des Programmfensters?

Nachzulesen unter [www.sprut.de/electronic/pic/config/config.htm](http://www.sprut.de/electronic/pic/config/config.htm).

## 12.7 Unerklärliche Brennabbrüche und Fehler beim Vergleichen auf schnellen Rechnern

Spontane Abbrüche auf Rechnern mit mehr als 600 MHz haben ihre Ursache oft in einer unsauberen Spannungserzeugung des Brenners. Wer im Brenner3 den 12-V-Spannungsregler mit Hilfe von 2 Dioden auf 13,4 V aufstockt sollte die Dioden mit einem 100 nF-Kondensator überbrücken, um solche Fehler zu vermeiden.

Softwaremäßig kann man folgendes versuchen:

- Im Setup-Fenster des Programms „auto. Speedregelung“ abschalten.
- Im dann erscheinenden „Geschwindigkeitsregler“ eine langsamere Geschwindigkeit einstellen.

Hilft das nicht :

- Masseverbindung zwischen Brenner und PC prüfen.
- Siebkondensatoren im Brenner vergrößern.

## 12.8 Der Brenner wird nicht erkannt

Das liegt meistens an einem Defekt des Brenners. Zum Test kann man folgendes probieren.

- Das Programm auf „Quick and Dirty“-Brenner einstellen
- „Brenner suchen“ Button anklicken, in den folgenden beiden Fenstern ‚OK‘ klicken
- Das Programm wieder auf den richtigen Brennertyp einstellen

Damit läßt sich die Brennererkennung überlisten, und anschließend die Funktion des Brenners testen.

# 13 Kommandozeilenparameter

Um PBrenner leichter in Entwicklungsumgebungen einzubinden, oder das Programm mit Batch-Dateien einfacher zu bedienen, unterstützt es zwei Kommandozeilenparameter:

- **/P<PIC-Typ>**  
Einstellen des zu brennenden PIC-Typs. Als <PIC-Typ> ist der Name des PIC ohne den Präfix ‚PIC‘ zu verwenden.  
Beispiel: PBrenner.exe /p16F84a  
PBrenner.exe /P16F876
- **/F<filename>**  
Laden eines bestimmten HEX-Files. Als <filename> ist der gesamte Filename inklusive

der Endung „.hex“ zu verwenden.

Beispiel: PBrenner.exe /ftestfile.hex  
PBrenner.exe /FANZEIGE.HEX

## 14 Bekannte Probleme

1)

Die Grafikkarte des PC darf nicht auf „große Schriftarten“ eingestellt sein. Ansonsten verschieben sich Bedienfelder des PBrenner-Programms.

2)

Der von einigen Compilern erzeugte HEX-Code trickst die „Programmspeicher-Füllstandsanzeige“ meines Programms aus. Die zeigt dann auch bei kleinen Programmen einen gelben Balken (100% voll) an. Das hat auf das Brennen keinen negativen Einfluß. Die ‚grafische Speicherdarstellung‘ deckt solche Tricks auf.

3)

Programmcode der außerhalb des physischen Programmspeichers des jeweiligen PIC liegt wird nicht in den PIC gebrannt. D.h. enthält ein Hex-File für den PIC16F84 Code für die nicht vorhandene Adresse 420h, wird dieser nicht automatisch auf die vorhandene Adresse 020h gebrannt.