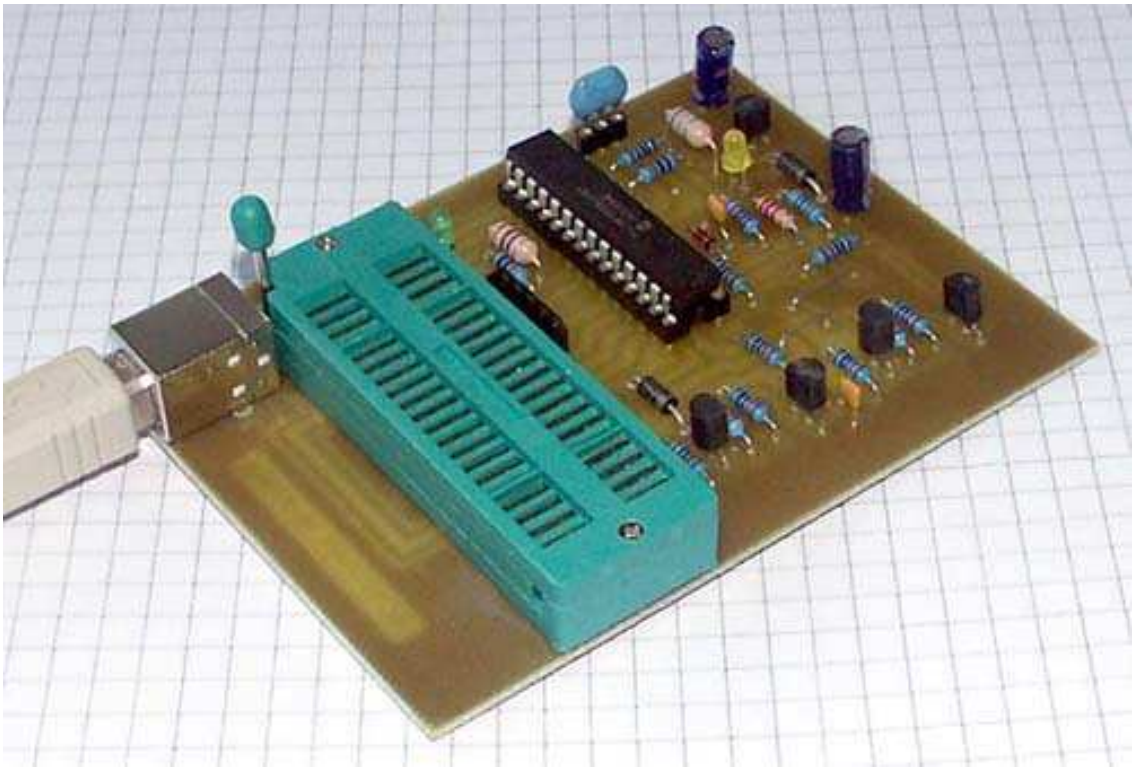
Handbuch für den Brenner8 (& 9)



Autor: sprut
Stand: 20.04.2009

1 Inhaltsverzeichnis

2 Abbildungsverzeichnis

3 Tabellenverzeichnis

4 NUTZUNGSBEDINGUNGEN:

4.1 US-Burn für Windows / Firmware

DIE SOFTWARE DARF OHNE ENTRICHTUNG EINER LIZENZGEBÜHR BENUTZT WERDEN. DAS GILT FÜR DIE PRIVATE UND GEWERBLICHE NUTZUNG.

DIE PUBLIKATION DER SOFTWARE ERFOLGT "AS IS". FÜR DIE EINHALTUNG ZUGESICHERTER EIGENSCHAFTEN ODER FÜR SCHÄDEN, DIE DURCH DEN EINSATZ ENTSTANDEN SEIN KÖNNTEN, ÜBERNIMMT DER AUTOR KEINERLEI HAFTUNG. SIE NUTZEN DIE SOFTWARE AUF EIGENE GEFAHR!

4.2 usburn für Linux

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

5 Einleitung

Der Brenner8/9 dient zum komfortablen Programmieren von PIC-Microchip-Prozessoren mit Flash-Programmspeicher.

Es liest Intel-Hex-Dateien wie sie z.B. vom Programm MPLAB erzeugt werden, und brennt diese in den Flash-Speicher des PICs. Im Hex-File enthaltene EEPROM-Daten und Konfigurationseinstellungen werden ebenfalls in den PIC gebrannt.

Zum Betrieb des Brenner8/9 benötigt man:

- den Brenner8/9 selbst
- die Firmware für den Steuer-PIC im Brenner8
- einen PC mit USB-Anschluss und USB-Kabel
- WindowsXP (nt/2k/Vista32) und das Windows-Programm: US-Burn
- oder Linux mit libusb und das Programm usburn
- die Database-Files mit Informationen für US-Burn/usburn

Der Brenner8/9 und die dazugehörige Software werden ständig weiterenwickelt. Dieses Handbuch basiert auf:

- Brenner8 Rev.5 mit Firmware Fw 0.13
- Brenner9 Rev.0 mit Firmware Fw 3.12
- USBurn V1.10 für Windows
- usburn V0.2 für Linux
- Database19

Der Brenner8/9 entstand ursprünglich 2006 als preiswertes PIC-Programmiergerät für Bastler. Der Einsatz von Parallelport-Brennern wurde durch das „Aussterben“ des Parallelports auf modernen PC-Mainboards immer schwieriger, und preiswerte industrielle Brenner mit USB-Anschluss waren nicht verfügbar.

5.1 Unterstützte PIC-Typen

Der Brenner8 wurde entworfen, um alle PIC-Microcontroller mit Flash-Programmspeicher und 5V Betriebsspannung programmieren zu können. Das sind:

- alle PIC18Fxxx und PIC18Fxxxx (keine PIC18FxxJxx oder PIC18FxxKxx)
- alle PIC16Fxx und PIC16Fxxx
- alle PIC12Fxxx
- alle PIC10Fxxx
- alle dsPIC30Fxxxx

Der Brenner9 wurde entworfen, um alle PIC-Microcontroller mit Flash-Programmspeicher und 3,3V Betriebsspannung programmieren zu können. Das sind:

- alle PIC18FxxJxx
- alle PIC24FJxxxx
- alle PIC24HJxxxx
- alle dsPIC33Fxxxx

Im Folgenden sind alle Typen für den Brenner8 aufgelistet:

supported members of PIC/dsPIC -series
-- PIC-database V.19 (17/04/2009)

Name	Pins	Prog [kB]	EEPRM [B]	TMR	ECCP /CCP	PWM mot	UART /SPI	SSP I2C	CAN	USB	ADC	IO-Pins
10F200	6	-	-	1	-/-	-	-/-	-/-	-	-	-	4
10F202	6	-	-	1	-/-	-	-/-	-/-	-	-	-	4
10F204	6	-	-	1	-/-	-	-/-	-/-	-	-	-	4
10F206	6	-	-	1	-/-	-	-/-	-/-	-	-	-	4
10F220	6	-	-	1	-/-	-	-/-	-/-	-	-	2	4
10F222	6	-	-	1	-/-	-	-/-	-/-	-	-	2	4
12F508	8	-	-	1	-/-	-	-/-	-/-	-	-	-	6
12F509	8	1	-	1	-/-	-	-/-	-/-	-	-	-	6
12F510	8	1	-	1	-/-	-	-/-	-/-	-	-	3	6
12F519	8	1	64	1	-/-	-	-/-	-/-	-	-	-	6
12F609	8	1	-	2	-/-	-	-/-	-/-	-	-	-	6
12F615	8	1	-	3	1/-	-	-/-	-/-	-	-	4	6
12F629	8	1	128	2	-/-	-	-/-	-/-	-	-	-	6
12F635	8	1	128	2	-/-	-	-/-	-/-	-	-	-	6
12F675	8	1	128	2	-/-	-	-/-	-/-	-	-	4	6
12F683	8	3	256	3	-/1	-	-/-	-/-	-	-	4	6
16F72	28	3	-	3	-/1	-	-/-	-/1	-	-	5	22
16F73	28	6	-	3	-/2	-	1/-	-/1	-	-	5	22
16F74	40	6	-	3	-/2	-	1/-	-/1	-	-	5	33
16F76	28	11	-	3	-/2	-	1/-	-/1	-	-	5	22
16F77	40	11	-	3	-/2	-	1/-	-/1	-	-	5	33
16F84	18	1	64	1	-/-	-	-/-	-/-	-	-	-	13
16F87	18	6	256	3	-/1	-	1/-	-/1	-	-	-	16
16F88	18	6	256	3	-/1	-	1/-	-/1	-	-	7	16
16F505	14	1	-	1	-/-	-	-/-	-/-	-	-	-	12
16F506	14	1	-	1	-/-	-	-/-	-/-	-	-	3	12
16F526	14	1	64	1	-/-	-	-/-	-/-	-	-	3	12
16F610	14	1	-	2	-/-	-	-/-	-/-	-	-	-	12
16F616	14	3	-	3	1/-	-	-/-	-/-	-	-	8	12
16F627	18	1	128	3	-/1	-	1/-	-/-	-	-	-	16
16F628	18	3	128	3	-/1	-	1/-	-/-	-	-	-	16
16F630	14	1	128	2	-/-	-	-/-	-/-	-	-	-	12
16F631	20	1	128	2	-/-	-	-/-	-/-	-	-	-	18
16F636	14	3	256	2	-/-	-	-/-	-/-	-	-	-	12
16F676	14	1	128	2	-/-	-	-/-	-/-	-	-	8	12
16F677	20	3	256	2	-/-	-	-/-	-/-	-	-	12	18
16F684	14	3	256	3	1/-	-	-/-	-/-	-	-	8	12
16F685	20	6	256	3	1/-	-	-/-	-/-	-	-	12	18
16F687	20	3	256	2	-/-	-	1/-	-/-	-	-	12	18
16F688	14	6	256	2	-/-	-	1/-	-/-	-	-	8	12
16F689	20	6	256	2	-/-	-	1/-	-/-	-	-	12	18
16F690	20	6	256	3	1/-	-	1/-	-/-	-	-	12	18

Name	Pins	Prog [kB]	EEPRM [B]	TMR	ECCP /CCP	PWM mot	UART /SPI	SSP I2C	CAN	USB	ADC	IO-Pins
16F716	18	3	-	3	1/-	-	-/-	-/-	-	-	4	13
16F722	28	3	-	3	-/2	-	1/-	1/-	-	-	11	25
16F723	28	6	-	3	-/2	-	1/-	1/-	-	-	11	25
16F724	40	6	-	3	-/2	-	1/-	1/-	-	-	14	36
16F726	28	11	-	3	-/2	-	1/-	1/-	-	-	11	25
16F727	40	11	-	3	-/2	-	1/-	1/-	-	-	14	36
16F737	28	6	-	3	-/3	-	1/-	-/1	-	-	11	25
16F747	40	6	-	3	-/3	-	1/-	-/1	-	-	11	36
16F767	28	11	-	3	-/3	-	1/-	-/1	-	-	11	25
16F777	40	11	-	3	-/3	-	1/-	-/1	-	-	11	36
16F785	20	3	256	3	-/1	-	-/-	-/-	-	-	12	18
16F818	18	1	128	3	-/1	-	-/-	-/1	-	-	5	16
16F819	18	3	256	3	-/1	-	-/-	-/1	-	-	5	16
16F870	28	3	64	3	-/1	-	1/-	-/-	-	-	5	22
16F871	40	3	64	3	-/1	-	1/-	-/-	-	-	8	33
16F872	28	3	64	3	-/1	-	-/-	-/1	-	-	5	22
16F873	28	6	128	3	-/2	-	1/-	-/1	-	-	5	22
16F874	40	6	128	3	-/2	-	1/-	-/1	-	-	8	33
16F876	28	11	256	3	-/2	-	1/-	-/1	-	-	5	22
16F877	40	11	256	3	-/2	-	1/-	-/1	-	-	8	33
16F882	28	3	128	3	1/1	-	1/-	-/1	-	-	11	25
16F883	28	6	256	3	1/1	-	1/-	-/1	-	-	11	25
16F884	40	6	256	3	1/1	-	1/-	-/1	-	-	14	36
16F886	28	11	256	3	1/1	-	1/-	-/1	-	-	11	25
16F887	40	11	256	3	1/1	-	1/-	-/1	-	-	14	36
16F913	28	6	256	3	-/1	-	1/-	-/1	-	-	5	25
16F914	40	6	256	3	-/2	-	1/-	-/1	-	-	8	36
16F916	28	11	256	3	-/1	-	1/-	-/1	-	-	5	25
16F917	40	11	256	3	-/2	-	1/-	-/1	-	-	8	36
16F946	64	11	256	3	-/2	-	1/-	-/1	-	-	8	54
16F1933	28	6	256	5	3/2	-	1/-	1/-	-	-	11	25
16F1934	40	6	256	5	3/2	-	1/-	1/-	-	-	14	36
16F1936	28	11	256	5	3/2	-	1/-	1/-	-	-	11	25
16F1937	40	11	256	5	3/2	-	1/-	1/-	-	-	14	36
16F84A	18	1	64	1	-/-	-	-/-	-/-	-	-	-	13
16F627A	18	1	128	3	-/1	-	1/-	-/-	-	-	-	16
16F628A	18	3	128	3	-/1	-	1/-	-/-	-	-	-	16
16F648A	18	6	256	3	-/1	-	1/-	-/-	-	-	-	16
16F873A	28	6	128	3	-/2	-	1/-	-/1	-	-	5	22
16F874A	40	6	128	3	-/2	-	1/-	-/1	-	-	8	33
16F876A	28	11	256	3	-/2	-	1/-	-/1	-	-	5	22
16F877A	40	11	256	3	-/2	-	1/-	-/1	-	-	8	33
16LF722	28	2	-	3	-/2	-	1/-	1/-	-	-	11	25
16LF723	28	4	-	3	-/2	-	1/-	1/-	-	-	11	25
16LF724	40	4	-	3	-/2	-	1/-	1/-	-	-	14	36
16LF726	28	8	-	3	-/2	-	1/-	1/-	-	-	11	25
16LF727	40	8	-	3	-/2	-	1/-	1/-	-	-	14	36
16LF1933	28	4	256	5	3/2	-	1/-	1/-	-	-	11	25
16LF1934	40	4	256	5	3/2	-	1/-	1/-	-	-	14	36
16LF1936	28	8	256	5	3/2	-	1/-	1/-	-	-	11	25
16LF1937	40	8	256	5	3/2	-	1/-	1/-	-	-	14	36

Name	Pins	Prog [kB]	EEPRM [B]	TMR	ECCP /CCP	PWM mot	UART /SPI	SSP I2C	CAN	USB	ADC	IO-Pins
18F242	28	16	256	4	-/2	-	1/-	1/-	-	-	5	23
18F248	28	16	256	4	-/1	-	1/-	1/-	1	-	5	23
18F252	28	32	256	4	-/2	-	1/-	1/-	-	-	5	23
18F258	28	32	256	4	-/1	-	1/-	1/-	1	-	5	23
18F442	40	16	256	4	-/2	-	1/-	1/-	-	-	8	34
18F448	40	16	256	4	1/1	-	1/-	1/-	1	-	8	34
18F452	40	32	256	4	-/2	-	1/-	1/-	-	-	8	34
18F458	40	32	256	4	1/1	-	1/-	1/-	1	-	8	34
18F1220	18	4	256	4	1/-	-	1/-	-/-	-	-	7	16
18F1230	18	4	128	2	-/-	1	1/-	-/-	-	-	4	16
18F1320	18	8	256	4	1/-	-	1/-	-/-	-	-	7	16
18F1330	18	8	128	2	-/-	1	1/-	-/-	-	-	4	16
18F2220	28	4	256	4	-/2	-	1/-	1/-	-	-	10	24
18F2221	28	4	256	4	-/1	-	1/-	-/-	-	-	10	24
18F2320	28	8	256	4	-/2	-	1/-	1/-	-	-	10	24
18F2321	28	8	256	4	-/1	-	1/-	-/-	-	-	10	24
18F2331	28	8	256	4	-/2	1	1/-	1/-	-	-	5	24
18F2410	28	16	-	4	-/1	-	1/-	1/-	-	-	10	25
18F2420	28	16	256	4	-/1	-	1/-	1/-	-	-	10	25
18F2423	28	16	256	4	-/1	-	1/-	1/-	-	-	10	25
18F2431	28	16	256	4	-/2	1	1/-	1/-	-	-	5	24
18F2450	28	16	-	3	-/1	-	1/-	-/-	-	-	10	22
18F2455	28	24	256	4	-/2	-	1/-	1/-	-	1	10	23
18F2458	28	24	256	4	-/2	-	1/-	1/-	-	1	10	24
18F2480	28	16	256	4	-/1	-	1/-	1/-	1	-	8	24
18F2510	28	32	-	4	-/1	-	1/-	1/-	-	-	10	25
18F2515	28	48	-	4	-/2	-	1/-	1/-	-	-	10	25
18F2520	28	32	256	4	-/1	-	1/-	1/-	-	-	10	25
18F2523	28	32	256	4	-/1	-	1/-	1/-	-	-	10	25
18F2525	28	48	1024	4	-/2	-	1/-	1/-	-	-	10	25
18F2550	28	32	256	4	-/1	-	1/-	1/-	-	1	10	23
18F2553	28	32	256	4	-/1	-	1/-	1/-	-	1	10	23
18F2580	28	32	256	4	-/1	-	1/-	1/-	1	-	8	24
18F2585	28	48	1024	4	-/1	-	1/-	1/-	1	-	8	25
18F2610	28	64	-	4	-/2	-	1/-	1/-	-	-	10	25
18F2620	28	64	1024	4	-/2	-	1/-	1/-	-	-	10	25
18F2680	28	64	1024	4	-/1	-	1/-	1/-	1	-	8	25
18F2682	28	80	1024	4	-/1	-	1/-	1/-	1	-	8	25
18F2685	28	96	1024	4	-/1	-	1/-	1/-	1	-	8	25
18F4220	40	4	256	4	1/1	-	1/-	1/-	-	-	13	36
18F4221	40	4	256	4	-/1	-	1/-	1/-	-	-	13	36
18F4320	40	8	256	4	1/1	-	1/-	1/-	-	-	13	36
18F4321	40	8	256	4	1/1	-	1/-	1/-	-	-	13	36
18F4331	40	8	256	4	-/2	1	1/-	1/-	-	-	9	36
18F4410	40	16	-	4	1/1	-	1/-	1/-	-	-	13	36
18F4420	40	16	256	4	1/1	-	1/-	1/-	-	-	13	36
18F4423	40	16	256	4	1/1	-	1/-	1/-	-	-	13	36
18F4431	40	16	256	4	-/2	1	1/-	1/-	-	-	9	36
18F4450	40	16	-	3	-/1	-	1/-	-/-	-	-	13	34
18F4455	40	24	256	4	-/1	-	1/-	1/-	-	1	13	34
18F4458	40	24	256	4	-/1	-	1/-	1/-	-	1	13	34
18F4480	40	16	256	4	1/1	-	1/-	1/-	1	-	11	36

Name	Pins	Prog [kB]	EEPRM [B]	TMR	ECCP /CCP	PWM mot	UART /SPI	SSP I2C	CAN	USB	ADC	IO-Pins
18F4510	40	32	-	4	1/1	-	1/-	1/-	-	-	13	36
18F4515	40	48	-	4	1/1	-	1/-	1/-	-	-	13	36
18F4520	40	32	256	4	1/1	-	1/-	1/-	-	-	13	36
18F4523	40	32	256	4	1/1	-	1/-	1/-	-	-	13	36
18F4525	40	48	1024	4	1/1	-	1/-	1/-	-	-	13	35
18F4550	40	32	256	4	1/1	-	1/-	1/-	-	1	13	34
18F4553	40	32	256	4	1/1	-	1/-	1/-	-	1	13	34
18F4580	40	32	256	4	1/1	-	1/-	1/-	1	-	11	36
18F4585	40	48	1024	4	1/1	-	1/-	1/-	1	-	11	36
18F4610	40	64	-	4	-/1	-	1/-	1/-	-	-	13	36
18F4620	40	64	1024	4	1/1	-	1/-	1/-	-	-	13	36
18F4680	40	64	1024	4	1/1	-	1/-	1/-	1	-	11	36
18F4682	40	80	1024	4	1/1	-	1/-	1/-	1	-	11	36
18F4685	40	96	1024	4	1/1	-	1/-	1/-	1	-	11	36
18F6310	64	8	-	4	-/3	-	2/-	1/-	-	-	12	54
18F6390	64	8	-	4	-/2	-	2/-	1/-	-	-	12	54
18F6393	64	8	-	4	-/2	-	2/-	1/-	-	-	12	50
18F6410	64	16	-	4	-/3	-	2/-	1/-	-	-	12	54
18F6490	64	16	-	4	-/2	-	2/-	1/-	-	-	12	54
18F6493	64	16	-	4	-/2	-	2/-	1/-	-	-	12	50
18F6520	64	32	1024	5	-/5	-	2/-	1/-	-	-	12	52
18F6525	64	48	1024	5	3/2	-	2/-	1/-	-	-	12	53
18F6527	64	48	1024	5	3/2	-	2/-	2/-	-	-	12	54
18F6585	64	48	1024	4	1/1	-	1/-	1/-	1	-	12	53
18F6620	64	64	1024	5	-/5	-	2/-	1/-	-	-	12	52
18F6621	64	64	1024	5	3/2	-	2/-	1/-	-	-	12	53
18F6622	64	64	1024	5	3/2	-	2/-	2/-	-	-	12	54
18F6627	64	96	1024	5	3/2	-	2/-	2/-	-	-	12	54
18F6628	64	96	1024	5	3/2	-	2/-	2/-	-	-	12	54
18F6680	64	64	1024	4	1/1	-	1/-	1/-	1	-	12	53
18F6720	64	128	1024	5	-/5	-	2/-	1/-	-	-	12	52
18F6722	64	128	1024	5	3/2	-	2/-	2/-	-	-	12	54
18F6723	64	128	1024	5	3/2	-	2/-	2/-	-	-	12	54
18F8310	80	8	-	4	-/3	-	2/-	1/-	-	-	12	70
18F8390	80	8	-	4	-/2	-	2/-	1/-	-	-	12	70
18F8393	80	8	-	4	-/2	-	2/-	1/-	-	-	12	66
18F8410	80	16	-	4	-/3	-	2/-	1/-	-	-	12	70
18F8490	80	16	-	4	-/2	-	2/-	1/-	-	-	12	70
18F8493	80	16	-	4	-/2	-	2/-	1/-	-	-	12	66
18F8520	80	32	1024	5	-/5	-	2/-	1/-	-	-	16	68
18F8525	80	48	1024	5	3/2	-	2/-	1/-	-	-	16	69
18F8527	80	48	1024	5	3/2	-	2/-	2/-	-	-	16	70
18F8585	80	48	1024	4	1/1	-	1/-	1/-	1	-	16	69
18F8620	80	64	1024	5	-/5	-	2/-	1/-	-	-	16	68
18F8621	80	64	1024	5	3/2	-	2/-	1/-	-	-	16	69
18F8622	80	64	1024	5	3/2	-	2/-	2/-	-	-	16	70
18F8627	80	96	1024	5	3/2	-	2/-	2/-	-	-	16	70
18F8628	80	96	1024	5	3/2	-	2/-	2/-	-	-	16	70
18F8680	80	64	1024	4	1/1	-	1/-	1/-	1	-	16	69
18F8720	80	128	1024	5	-/5	-	2/-	1/-	-	-	16	68
18F8722	80	128	1024	5	3/2	-	2/-	2/-	-	-	16	70
18F8723	80	128	1024	5	3/2	-	2/-	2/-	-	-	16	70

Name	Pins	Prog [kB]	EEPRM [B]	TMR	ECCP /CCP	PWM mot	UART /SPI	SSP I2C	CAN	USB	ADC	IO-Pins
30F2010	28	12	1024	3	-/2	1	1/1	-/1	-	-	6	20
30F2011	14	12	-	3	-/2	-	1/1	-/1	-	-	8	12
30F2012	28	12	-	3	-/2	-	1/1	-/1	-	-	10	20
30F3010	28	24	1024	5	-/2	1	1/1	-/1	-	-	6	20
30F3011	40	24	1024	5	-/4	1	2/1	-/1	-	-	9	30
30F3012	14	24	1024	3	-/2	-	1/1	-/1	-	-	8	12
30F3013	28	24	1024	3	-/2	-	2/1	-/1	-	-	10	20
30F3014	40	24	1024	3	-/2	-	2/1	-/1	-	-	13	30
30F4011	40	48	1024	5	-/4	1	2/1	-/1	1	-	9	30
30F4012	28	48	1024	5	-/2	1	1/1	-/1	1	-	6	20
30F4013	40	48	1024	5	-/4	-	2/1	-/1	1	-	13	30
30F5011	64	66	1024	5	-/8	-	2/2	-/1	2	-	16	52
30F5013	80	66	1024	5	-/8	-	2/2	-/1	2	-	16	68
30F5015	64	66	1024	5	-/4	1	1/2	-/1	1	-	16	52
30F5016	80	66	1024	5	-/4	1	1/2	-/1	1	-	16	68
30F6010	80	144	4096	5	-/8	1	2/2	-/1	2	-	16	68
30F6011	64	132	2048	5	-/8	-	2/2	-/1	2	-	16	52
30F6012	64	144	4096	5	-/8	-	2/2	-/1	2	-	16	52
30F6013	80	132	2048	5	-/8	-	2/2	-/1	2	-	16	68
30F6014	80	144	4096	5	-/8	-	2/2	-/1	2	-	16	68
30F6015	64	144	4096	5	-/8	1	2/2	-/1	1	-	16	52
30F6010A	80	144	4096	5	-/8	1	2/2	-/1	2	-	16	66
30F6011A	64	132	2048	5	-/8	-	2/2	-/1	2	-	16	52
30F6012A	64	144	4096	5	-/8	-	2/2	-/1	2	-	16	52
30F6013A	80	132	2048	5	-/8	-	2/2	-/1	2	-	16	68
30F6014A	80	144	4096	5	-/8	-	2/2	-/1	2	-	16	68
30F2011es	-	12	-	-	-/-	-	-/-	-/-	-	-	-	-
30F2012es	-	12	-	-	-/-	-	-/-	-/-	-	-	-	-
30F6010es	-	144	4096	-	-/-	-	-/-	-/-	-	-	-	-

Durch Aktualisierung der Firmware und der Windows-Software kann die vom Brenner8 unterstützte Typenpalette immer aktuell gehalten werden.

Durch einen Adapter und kleinere Hardwaremodifikationen kann der Brenner8P zum Brenner9 umgebaut werden, der 3,3V-PICs folgender Serien brennen kann:

- PIC18FxxJxx
- PIC24xxxx
- dsPIC33Fxxxx

Alle 14&16-Bit-Kern-PICs (PIC16Fxxxx/PIC18Fxxxx) mit DIL-Gehäuse (8 / 14 / 18 / 28 / 40 Pin-DIL-Gehäuse) können im 40-poligen Testsockel des Brenner8 programmiert werden. Alle anderen Typen sind mit einem geeigneten Adapter am ICSP-Steckverbinder anzuschließen.

Alle dsPIC30Fxxxx, PIC1xF5x, PIC1xF5xx und PIC10Fxxx müssen auch am ICSP-Steckverbinder adaptiert werden. Sie passen elektrisch nicht in den Testsockel des Brenner8.
Die dsPIC30F-Typen können nur vom Brenner8P/Brenner8miniP programmiert werden.

6 Aufbau der Hardware

6.1 Die Hardware des Brenner8

Dieser Abschnitt ist nur für technisch interessierte Nutzer des Brenner8. alle anderen Leser können zum nächste Punkt dieses Dokuments übergehen.

Es folgt eine kurze Beschreibung der Brenner8-Hardware:

Der Brenner8 besteht aus folgenden Baugruppen:

- USB-Interface
- Takterzeugung
- Referenzspannungsquelle
- Programmierspannungserzeugung
- 2 Programmierspannungsschaltern
- Testsockel

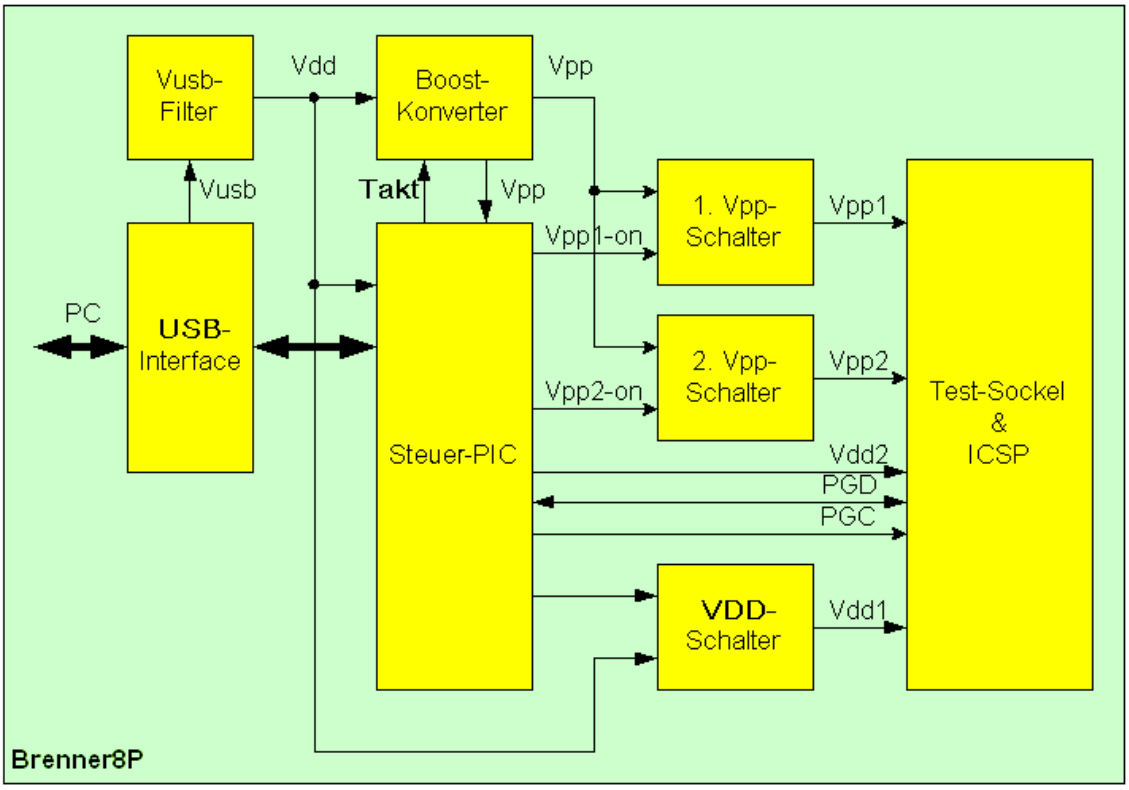


Abbildung 1 Blockschaltbild des Brenner8

6.1.1 USB-Interface

Das USB-Interface dient der Kommunikation mit dem PC wie auch der Stromversorgung des Brenners. Die USB-Buchse dient dem Anschluß an den PC. Der Kondensator am Vusb-Pin des Steuer-PIC ist Bestandteil der 3,3V-Erzeugung im PIC18F2550. Diese 3,3V werden als Signalpegel auf den USB-Leitungen verwendet.

Mit einer Spule und zwei Kondensatoren wird die VBUS-Spannung des USB-Interfaces gefiltert, und steht dann als Betriebsspannung Vdd des Brenners zur Verfügung (4.5 .. 5 V)

6.1.2 Takterzeugung

Mit einem Rsonator bzw. einem Quarz und 2 Kondensatoren wird ein 20 MHz-Takt erzeugt, aus dem im PIC18F2550 sowohl der Arbeitstakt des PIC-Kerns wie auch der USB-Takt abgeleitet werden.

6.1.3 Referenzspannung

Eine 3,3V-Z-Diode mit einem Vorwiderstand erzeugen eine stabile Spannung (ca. 3,3V) die vom Betriebsspannungspegel (und damit vom VBUS-Pegel) unabhängig ist. Sie dient als Referenz für die Messung (und damit für die korrekte Einstellung) der Programmierspannung.

6.1.4 Programmierspannungserzeugung

Die PICs benötigen zur Programmierung eine Programmierspannung von 10 .. 13V (je nach Typ). Sie wird aus der Betriebsspannung durch einen Boost-Konverter erzeugt. Der PIC18F2550 gibt am Pin 13 ein 100kHz-Rechtecksignal aus, dessen Pulsweitenverhältnis einstellbar ist.
Die an mit dem Boost-Converter erzeugte Spannung beträgt theoretisch

Vpp = 4,7V * Periode / off-zeit

wobei "Periode" die Dauer eine Rechteckschwingung ist, wärend "off-zeit" der low-Teil dieser Periode ist. Durch Änderung von "off-zeit" kann die Vpp-Spannung eingestellt werden.
In der Praxis gibt einige Seiteneffekte, die durch eine Kalibrierung kompensiert werden müssen.

Zwei Widerstände bilden einen Spannungsteiler, über den der PIC18F2550 die erzeugte Vpp-Spannung messen kann (am Pin 3).

6.1.5 2 Programmierspannungsschalter

Es gibt 2 Schalter, die Vpp zum Testsockel zuschalten können. (Der Brenner8mini(P) hat nur einen Vp-Schalter.) Da verschiedene PICs die Programmierspannung an verschiedenen Pins erwarten, muß je nach PIC-Typ der eine oder der andere Schalter aktiviert werden.
Jeder Schalter besteht aus einem npn- und einem pnp-Transistor.

6.1.6 Testsockel

Am Testsockel bzw. am ICSP-Verbinder müssen die 5 Signale Vpp, Vdd, Vss, Data, Clk angelegt werden. Mit Ausnahme von Vpp werden alle Signale direkt von Port-Pins des 18F2550 erzeugt. Je nach Gehäuse des Target-PICs werden die Signale an andere Pins gelegt.
Eine Diode ist nötig, da Pin 1 des Testsockels manchmal Vpp und manchmal Vss ist (je nach PIC). RB7 erzeugt bei Bedarf Vss-Pegel.

6.1.7 Rest

Ein pull-up-Widerstand und ein Jumper erzeugen am Pin 1 des 18F2550 high- oder low-Pegel. Erkennt der 18F2550 beim Einschalten dort low-Pegel, dann startet er den Bootloader, ansonsten startet er die Brenner8-Firmware.

6.2 Varianten des Brenner8

Es existieren mehrere Varianten des Brenner8, die aber alle mit der gleichen Software/Firmware betrieben werden.
Je nach Brennertyp eignet sich der Brenner für verschiedene Aufgaben:

Tabelle 1 Brenner8-Versionen

Typ \ brennt	PIC im Testsockel	PICs via ICSP-Adapter	PICs via ICSP in der Schaltung	dsPIC30F
Brenner8	X	X	-	-
Brenner8-P	**X**	**X**	**X**	**X**
Brenner8mini	-	X	-	-
Brenner8mini-P	-	X	X	X

Der Brenner8P und der Brenner8miniP (Silviu-Layout) lassen sich mit einem zusätzlichen Adapter zum Brennen von 3,3V-PICs verwenden (Brenner9).

Wenn nicht ausdrücklich erwähnt, dann beziehe ich mich im Text immer auf den Brenner8.

6.2.1 Brenner8

Der einfache Brenner8 hat einen 40-poligen Testfassung und einen 5-poligen ICSP-Verbinder. Er ist in erster Linie dafür gedacht PICs im Testsockel zu Brennen Am ICSP-Anschluss können über Adapter auch PICs angeschlossen und gebrannt werden.

Der klassische Brenner8 ist nicht dafür ausgelegt, PICs in einer fertig aufgebauten Schaltung via ICSP-Kabel zu brennen. Sein Vdd-Ausgang (im ICSP-Anschluss) ist nicht in der Lage, neben dem zu brennenden PIC noch weitere Baugruppen mit Spannung zu versorgen. Lediglich Kondensatoren von bis zu 100µF dürfen am zu brennenden PIC angeschlossen sein.

6.2.2 Brenner8-P

Der Brenner8-P weist alle positiven Eigenschaften des Brenner8 auf. Er hat aber einen Treibertransistor für die Vdd-Leitung im ICSP-Verbinder. Deshalb kann er auch PICs in der fertig aufgebauten Schaltung problemlos brennen, sofern die Schaltung für ICSP ausgelegt wurde.

6.2.3 Brenner8mini

Der Brenner8mini ist ein auf die ICSP-Funktion reduzierter Brenner8. Er hat keinen Testsockel. Nur am ICSP-Anschluss können über Adapter PICs angeschlossen und gebrannt werden.

Wie der klassische Brenner8 ist auch der Brenner8mini nicht dafür ausgelegt, PICs in einer fertig aufgebauten Schaltung via ICSP-Kabel zu brennen. Sein Vdd-Ausgang (im ICSP-Anschluss) ist nicht in der Lage, neben dem zu brennenden PIC noch

weitere Baugruppen mit Spannung zu versorgen. Lediglich Kondensatoren von bis zu 100µF dürfen am zu brennenden PIC angeschlossen sein.

6.2.4 Brenner8mini-P

Der Brenner8mini-P ist ein Brenner8mini mit Vdd-Treibertransistor. Deshalb kann er auch PICs in der fertig aufgebauten Schaltung problemlos brennen, sofern die Schaltung für ICSP ausgelegt wurde.

6.3 Revisionen des Brenner8

Der Brenner8 wird kontinuierlich weiterentwickelt, um seine Eigenschaften zu verbessern. Bei jeder Hardwareänderung, vergebe ich eine neue Revisionsnummer. Neue Revisionen benötigen zur Nutzung der verbesserten Fähigkeiten manchmal auch eine neuere Firmware. Aber umgekehrt funktioniert auch eine neue Firmware stets in einer alten Hardware-Revision.

6.3.1 Revision 0

Das war die Urversion des Brenner8. Er hatte noch einige ernsthafte Probleme. So war der Betrieb von 8/14-Pin-PICs im Testsockel nicht möglich, die Vpp-Spannung war manchmal zu niedrig, die Z-Diodenspannung hatte große Toleranzen und die Vpp-Spannungsmessung hatte einen zu eingeschränkten Messbereich. Eine Rev. 0 lässt sich einfach auf eine Rev. 1 umbauen, und das sollte man auch tun.

6.3.2 Revision 1

In der Rev. 1 wurden die wesentlichen Mängel der Rev. 0 beseitigt.

- Durch den Tausch von D2 gegen eine Shottky-Diode wurde der Betrieb von 8/14-Pin-PICs im Testsockel möglich.
- Durch Änderung von R16 wird die Z-Spannung verbessert.
- Durch Änderung von R5 wird der Vpp-Messbereich erweitert
- Durch den Tausch von D1 gegen eine Shottky-Diode wird die Vpp-Erzeugung leistungsfähiger.

Der Brenner8-P, Brenner8mini und Brenner8mini-P wurden zu dieser Zeit aus der Taufe gehoben. Sie trugen also gleich die Revisionsnummer 1.
Die Revision 1benötigte mindestens die Firmware 0.5. Die beiden P-Versionen benötigen wenigstens die Firmware V. 0.5a.

6.3.3 Revision 2

Mit der Einführung des Bootloaders kam auch die Revision 2.
Die einzige Hardwareänderung ist ein Jumper, mit dem sich der Bootloader aktivieren lässt. Da diese Aktivierung normalerweise auch via Software erfolgen kann, und im Notfall ein Stück Draht den Jumper ersetzen kann, ist ein Umbau von Rev. 1 auf Rev. 2 nicht nötig. Beim Brenner8mini und Brenner8mini-P verzichtete ich gleich auf die Revision 2.
Zur Revision 2 gehört eine Firmware ab V. 0.6 und der Bootloader 1, die aber auch für die Rev. 1 zu empfehlen sind.

6.3.4 Revision 3

Diese Revision soll vor allem dem Brenner8P eine vollwertige ICSP-Unterstützung bescheren. Dafür waren eine zusätzlicher Reset-Transistor (Q8) und eine MCLR pull-up-Diode (D4) nötig. Der normale Brenner8 erhält nur die Diode D4.
Mit dem Reset-Transistor können PICs auch in Schaltungen gebrannt werden, in denen ein niederohmiger (wenige Kiloohm) pull-up-Widerstand am MCLR-Pin sitzt. Die Diode D4 ermöglicht es einen PIC in einer Testplatine oder in einer fertigen Schaltung „laufen“ zu lassen. Das erleichtert deutlich die Softwareentwicklung, da der Brenner zum Test der Software nicht mehr von der Testplatine getrennt werden muss.
Für das Brennen von PICs im Testsockel oder einem einfachen ICSP-Adapter bringt die Revision 3 keine Veränderungen.
Die Revision 3 benötigt die Firmware V 0.8.

6.3.5 Revision 4

Es gab einige Probleme beim Löschen von einigen wenigen PIC-Typen mit großem Flash-Speicher wie z.B. PIC18F4682. Als Ursache stellte sich heraus, dass diese PICs während des Brennens/Löschens kurzfristig deutlich mehr als die spezifizierten 10 mA aus der Versorgungsspannung Vdd aufnahmen. Pufferkondensatoren an allen unverstärkten Vdd-Ausgängen beheben dieses Problem. Der Brenner8 erhält zwei Pufferkondensatoren, während der Brenner8P und der Brenner8mini jeweils einen Kondensator erhalten. Der Brenner8miniP benötigt keine Modifikation.

Brenner der Revisionen 2 und 3 lassen sich mit den Kondensatoren leicht nachrüsten. Ich empfehle Keramikkondensatoren von 47 .. 220 nF, die sich leicht auf der Unterseite der Platine anlöten lassen. (Meistens genügen 47 nF) Der Brenner8 und der Brenner8P erhalten einen Kondensator zwischen den Pins 31 und 32 der 40-poligen Testfassung. Der Brenner8 und der Brenner8mini erhalten einen Kondensator zwischen den Pins 2 und 3 der 5-poligen ICSP-Buchse.
Das Bild zeigt einen entsprechend modifizierten Brenner8.

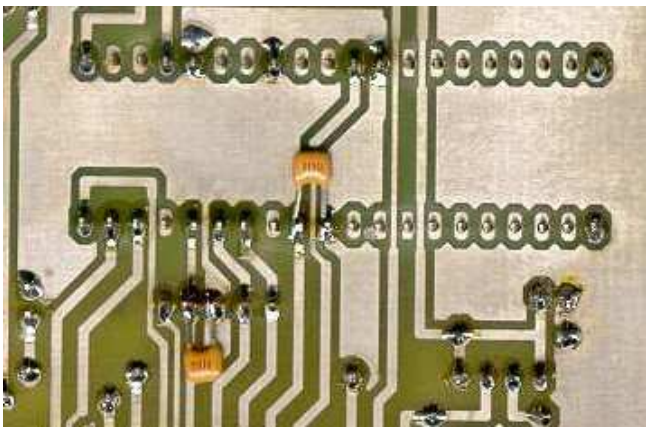


Abbildung 2 Leiterseite eines entsprechend Rev. 4 modifizierten Brenner8

Brenner8 der Revision 4 benötigen eine Firmware ab 0.8a, um alle PIC im on-board-Sockel problemlos erkennen zu können.

6.3.6 Revision 5

Es stellte sich heraus, dass auch der Brenner8P/Brenner8miniP von einem Vpp-Kondensator zwischen Pin2 und Pin 3 der ICSP-Buchse profitieren kann, also bekommt er ihn in der Revision 5. Die Einrüstung dieses Kondensators empfehle ich dringend.

Weitere Änderungen zielen lediglich auf eine leichte Umrüstbarkeit des Brenner8P/Brenner8miniP-Silviu zum Brenner9.

Der Brenner8P rev4 (und der Brenner8miniP-Silviu) hat einen Resettransistor Q8, der im Original ein BC338 war. Dieser Typ ist durch einen Transistor mit kleiner Basis-Kollector-Kapazität zu ersetzen. Gut geeignet ist der HF-Typ BF959. Beim Tausch des Transistors ist darauf zu achten, dass der BF959 (B-E-C) eine andere Anschlussbelegung hat als der BC338 (E-B-C) !

Der Brenner8P rev4 hat einen Widerstand R19 mit dem Wert von 330 Ohm (beim Brenner8miniP-Silviu ist das R13). Dieser Widerstand ist durch einen 100-Ohm-Widerstand zu ersetzen.

Die 10nF-Kondensatoren C6 und C7 des Brenner8P sind durch 1nF-Typen zu ersetzen. (Beim Brenner8miniP-Silviu betrifft das nur C6.)

6.4 Varianten des Brenner9

Wärend er Brenner8 geeignet ist 5V-PICs zu brennen, ist der Brenner9 für die 3,3V-PICs gedacht. Diese PICs haben einen engen Betriebsspannungsbereich (meist 3V ... 3,6V) und würden deshalb das Brennen mit einem 5V-Brenner nicht überleben. Ein Vorteil der 3,3V-PICs ist, dass sie keine hohe Programmierspannung benütigen, und ein Boost-Konverter (wie im Brenner8) damit entfallen kann.
Es gibt drei Varianten des Brenner9

- Brenner8 mit Adapter
- Brenner9N
- Brenner9L

Dabei arbeiten Brenner9N und Brenner8+Adapter mit der gleichen Firmware. Der Brenner9L sollte mit einem geringerem internen Takt betrieben werden, als die anderen Typen. Das erfordert eine modifizierte Firmware.
Die Firmware des Brenner9N kann im Brenner9L funktionieren, es ist aber nicht garantiert. Die Firmware des Brenner9L funktioniert auch im Brenner9N.

6.4.1 Brenner8+Adapter

Ein an den ICSP-Steckverbinder angeschlossener Adapter wandelt die 5V-Signale der Brenner8-Hardware in 3,3V-Signale. Damit kann die Brenner8-Hardware auch für 3,3V-PICs verwendet werden.
In den Brenner8 ist dafür noch die Brenner9-Firmware zu laden. Diese Firmware beherrscht das Brennen der 3,3V-PICs und benutzt den Vpp-Boost-Konverter nicht.
Voraussetzung ist ein Brenner8P oder Brenner8miniP-Silviu der Revision 5.

Eine Firmware, die sowohl die Brenner8-Funktionen wie auch die Brenner9-Funktionen beinhaltet, gibt es nicht. Sie würde nicht in den Programmspeicher des Steuer-PIC passen.

6.4.2 Brenner9N

Der Brenner9 ist prinzipiell ein Brenner8miniP, ohne Boost-Konverter aber mit integriertem 5V-3,3V-Adapter.

6.4.3 Brenner9L

Im Brenner9L arbeitet ein PIC18LF2450 als Steuer-PIC, der mit einer Betriebsspannung von 3,3V-betrieben wird. Damit kann der Steuer-PIC direkt die nötigen 3,3V-Signale erzeugen, und ein 5V/3,3V-Adapter ist überflüssig. Die Betriebsspannung des PIC18LF2450 wird mit einem 3,3V-Festspannungsregler erzeugt.
Aufgrund der verminderten Betriebsspannung, ist der maximal zulässige interne Takt der PIC18LF2450 geringer, als der normalerweise in meinen Brennern intern verwendete Takt von 48 MHz. Deshalb ist eine angepasste Firmware und auch ein angepasster Bootloader erforderlich!

6.5 Platine

Im Folgenden beziehe ich mich exemplarisch auf den Brenner8P. Für andere Brenner-Versionen gilt alles sinngemäß.

Obwohl es sicherlich möglich ist, den Brenner8 auf einer Lochrasterplatine aufzubauen, empfehle ich doch die fotochemische Herstellung einer Leiterplatte. Das von mir bereitgestellte Layout (siehe Anlage) stellt keine allzu hohen Anforderungen. Das Layout ist nicht sehr filigran, und es wird nur eine einseitige Platine benötigt.

Das Layout ist etwa 75mm x 100mm groß. Platinenmaterial dieser Größe ist handelsüblich. Damit entfällt das lästige Zuschneiden des Leiterplattenmaterials.

Das Layout des Brenner8mini-P ist auch einseitig, aber etwas filigraner. Das Platinenmaß ist 83mm x 43mm.

Für den Notfall liegt auf der Brenner8-Homepage auch ein Layoutvorschlag für einen Brenner8 auf einer Lochrasterplatine.

6.6 Bestückung

Nach dem Ätzen und Bohren wird die Platine wie üblich bestückt. Dabei startet man mit den Drahtbrücken, es folgen die flachen Bauteile (Widerstände, Dioden), dann die 28-polige Fassung, und zum Schluss alles „sperrige“.

Die Werte der Widerstände und Kondensatoren sind unkritisch. Die Werte dürfen um 25% von den im Stromlaufplan angegebenen Werten abweichen. Eine Ausnahme bilden die Vpp-Spannungsteilerwiderstände R4 & R5. Um die spätere Kalibrierung des Brenners zu vereinfachen, sollte hier nicht von den Werten des Stromlaufplans abgewichen werden.

Beim Einlöten der Testfassung sollte der Arretierheben der Fassung auf „offen“ stehen, um ein Verkannten der Fassungskontakte zu vermeiden.

Wer andere Transistortypen einsetzt, muss auf die Pinbelegung dieser Typen achten. Alle von mir vorgesehenen Typen mit Ausnahme des BF959 haben die Anschlussreihenfolge E-B-C. Beim BF959 lautet sie dagegen B-E-C.

Als Dioden eignen sich alle Shottky-Dioden mit einer Spannungsfestigkeit von mindestens 30V und einer Strombelastbarkeit von mindestens 100 mA. Ich empfehle die BAT43.

6.7 Firmware & Bootloader brennen

Im Brenner8 arbeitet ein Steuer-PIC vom Typ PIC18F2550. Der benötigt für seine Arbeit natürlich Software. Die besteht aus

- einem Bootloader und
- der eigentlichen Firmware.

Im Normalfall wird beim Einschalten des Brenner8 die Firmware aktiviert. Sie ist für alle Brennaufgeben des Brenner8 zuständig. Der Bootloader hat nur eine Aufgabe: Er kann bei Bedarf einen neue Firmware in den Brenner8 laden. Das ist natürlich bei der Erstinbetriebnahme nötig, kann aber auch nötig sein, wenn eine verbesserte Firmware für den Brenner8/9 veröffentlicht wird.

Um zunächst den Bootloader in den Steuer-PIC zu bekommen wird ein Brenner benötigt, der den PIC18F2550 brennen kann. Das kann z.B. ein Brenner5 sein, oder irgendein anderer PIC-Brenner, der den PIC18F2550 unterstützt.

Der Bootloader ermöglicht es später, unproblematisch die Firmware zu aktualisieren, und er übernimmt auch die Kontrolle über den Taktgenerator des Steuer-PIC. Der Bootloader liegt in verschiedenen Varianten für verschiedene Quarzfrequenzen vor (4, 8, 20 MHz). Man wählt einfach den passenden Bootloader aus, und brennt ihn mit einem Programmiergerät in den PIC.
(Für den Brenner9L gibt es einen speziell modifizierten Bootloader, alle anderen Brenner8/9 verwenden den Standard-Bootloader.)

Damit wird automatisch auch die Taktfrequenz für die Firmware festgelegt. Wenn man z.B. einen 8-MHz-Resonator verwenden will, dann brennt man einen 8-MHz-Bootloader in den PIC. Wird dann später die (eigentlich für 20 MHz ausgelegte) Firmware mit Hilfe des Bootloaders in den PIC geladen, dann funktioniert diese auch mit dem 8 MHz-Resonator.
Da der Bootloader sehr klein ist, kann er auch mit weniger zuverlässigen Programmiergeräten in den PIC18F2550 gebrannt werden.

Der PIC18F2550, der bisher nur den Bootloader enthält, wird nun als Steuer-PIC in den Brenner8/9 eingesetzt. Nun kann die eigentliche Firmware mit Hilfe des Bootloaders in den PIC nachgeladen werden. Die Details sind im Kapitel „Bootloader“ beschrieben.

6.8 Taktquelle

Der Brenner kann sowohl mit einem Keramikresonator wie auch mit einem Quarz betrieben werden. Wird ein Quarz verwendet, dann sind auch die beiden Lastkondensatoren für den Quarz (C2 & C3) einzusetzen. Wird dagegen ein Keramikresonator eingesetzt, dann entfallen die beiden Kondensatoren. Der Resonator wird in der Platine am Einbauort der beiden Kondensatoren eingebaut.

Im Brenner8 passt der Mittelanschluss des Resonators in den zusätzlichen Masseanschluss zwischen den Kondensatoreinbauorten.
Im Brenner8mini-Simon-Layout wird der Resonator in die beiden Lötaugen für C2 und das untere Lötauge von C3 eingesetzt. Das Brenner8miniP-Silviu-Layout sieht keinen Resonator vor, der Bastler ist hier etwas mehr gefordert.

Als Frequenz für den Resonator/Quarz ist 20 MHz vorgesehen. Leider sind 20MHz-Resonatoren nicht leicht zu beschaffen. Weiter oben wurde beschrieben, wie man mit Hilfe des Bootloaders einen anderen Takt verwenden kann. Es gibt aber auch noch einen zweiten Weg, den Brenner8 mit einem anderen Takt als 20MHz zu betreiben. Der Einsatz anderer Resonator/Quarz-Typen ist möglich, wenn folgendes beachtet wird:

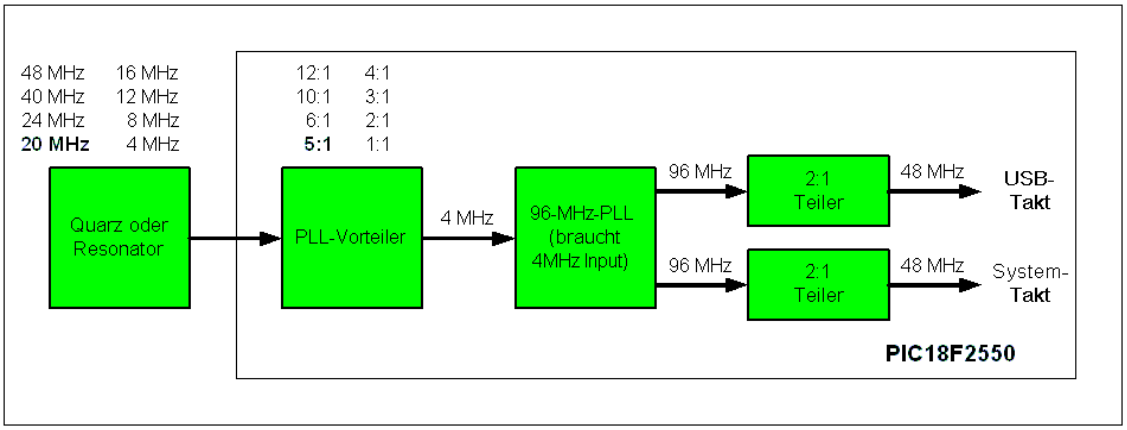


Abbildung 3 Takterzeugung im Steuer-PIC

Standardmäßig wird die Quarzfrequenz im PIC zunächst mit einem 5:1 Frequenzteiler auf 4 MHz heruntergeteilt. Aus diesen 4 MHz werden anschließend mit einer PLL 96 MHz erzeugt. Diese wiederum dient als Basis für den USB-Takt (2:1 Teilung) und den PIC-Takt (ebenfalls 2:1 Teilung).

Die 4 MHz für die PLL lassen sich natürlich nicht nur aus 20 MHz erzeugen. Da der Eingangsteiler neben dem Teilverhältnis 5:1 auch die Teilverhältnisse 12:1, 10:1, 6:1, 4:1, 3:2, 2:1 und 1:1 beherrscht, kommen auch Resonatoren/Quarze mit 48 MHz, 40 MHz, 24 MHz, 16 MHz, 12 MHz, 8 MHz und 4 MHz in Frage. Man muss nur die Vorteilereinstellung ändern.

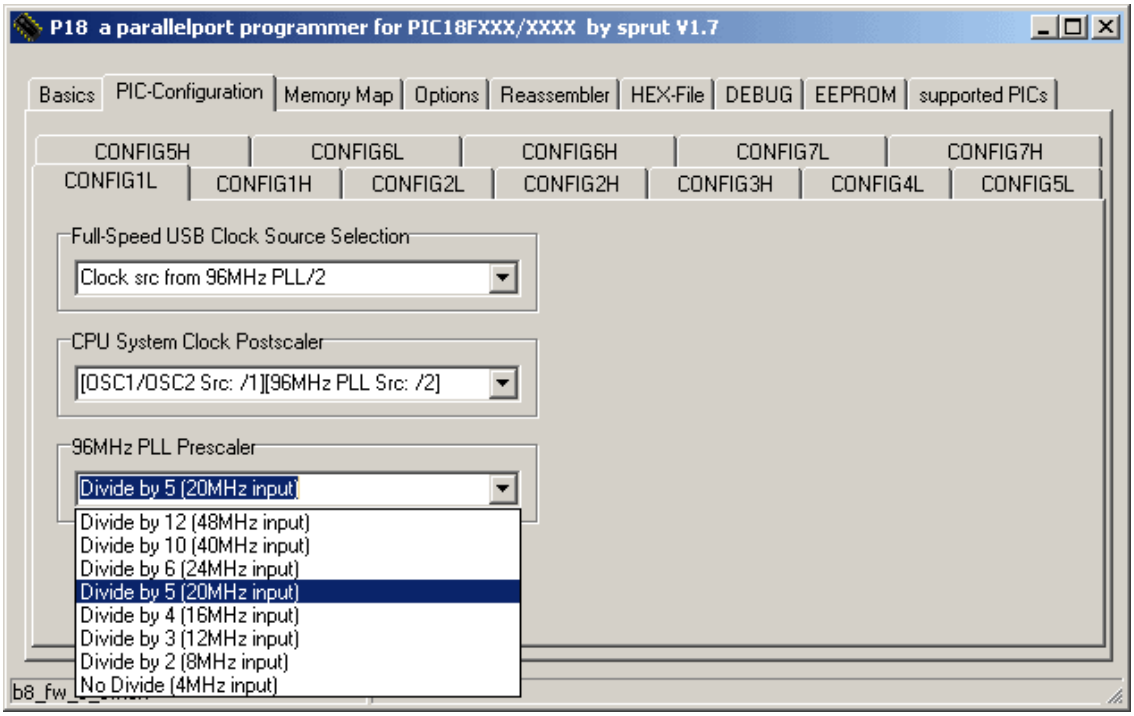


Abbildung 4 Resonator/Quarz-Einstellung für den Steuer-PIC

Die Vorteilereinstellung ist in der PIC-Konfiguration festgelegt, die im Firmware-HEX-File abgelegt ist. Mit einer geeigneten Brennsoftware, wie P18 oder US-Burn, lässt sich diese Konfigurationseinstellung vor dem Brennen des Steuer-PICs manipulieren. Dazu wird nach dem Laden des HEX-Files in der „Basics“ -Karteikarte von P18 oder US-Burn die Option „**Config from HEX-File**“ deaktiviert. Danach wechselt man auf die Karteikarte „**PIC Configuration**“. Dort wählt man die Unterkarteikarte „**CONFIG1L**“.
Hier nun findet man alle Takteinstellungen des PIC. Wichtig ist die Option „**96 MHz PLL Prescaler**“, die standardmäßig auf „**Divide by 5 (20MHz input)**“ steht. Diese Option passt man einfach der gewünschten Resonator/Quarz-Frequenz an. Die Einstellung „**Divide by 2 (8MHz input)**“ erlaubt z.B. den Einsatz eines 8 MHz Keramikresonators.

6.8.1 Takt beim Brenner9L

Der Brenner9L verwendet einen internen System-Takt von nur 16 MHz (anstelle von 48 MHz). Deshalb ist bei Ihm der Frequenzteiler, der den 96-MHz-PLL-Ausgangstakt auf den Systemtakt herunterteilt auf ein Teilverhältnis von 6:1 (anstelle von 2:1) eingestellt. (CPU System Clock Postscaler)
Alle anderen Takte (auch der externe Quarztakt) sind mit dem Brenner8 und dem Brenner9N identisch.
Für den Brenner9L gibt es an diese Taktbesonderheit angepasse Bootloader und Firmware-Versionen.

6.9 Funktionstest

Nach dem Zusammenbau des Brenners und dem Einsetzen des Steuer-PICs wird er im laufenden Betrieb an einen PC angeschlossen. Unmittelbar darauf leuchtet die grüne LED auf, gefolgt von der gelben LED. In der gleichen Reihenfolge verlöschen beide LEDs auch wieder nach jeweils 0,5 Sekunden Leuchtzeit.
An der Kathode der Diode D1 des Brenner8 kann man eine Spannung zwischen 10 V und 20 V messen. Fehlt das Aufblinken der LEDs und ist die Spannung an der Kathode von D1 < 6V, dann ist die Firmware nicht korrekt in den Steuer-PIC gebrannt, oder der Quarz schwingt nicht.

6.9.1 Funktionstest - Windows

Ist der USB-Treiber noch nicht installiert, dann meldet sich Windows, und verlangt die Installation des Treibers, die in einem anderen Kapitel beschrieben ist. Sollte Windows melden, dass sich ein USB-Device nicht korrekt angemeldet hat, und deshalb nicht benutzt werden kann, so könnte z.B. ein Quarz mit falscher Frequenz eingesetzt worden sein.

In der Brennersoftware US-Burn gibt es die Möglichkeit, zu Testzwecken die Betriebsspannung Vdd, die Programmierspannung Vpp sowie die Daten- und die Takt-Leitung einzeln ein- und auszuschalten („**Options – Hardware**“). Zur besseren Orientierung wird dabei die grüne LED zusammen mit Vdd und die gelbe LED zusammen mit Vpp geschaltet.

6.9.1.1 Spannungspegel im Brenner8

Je nach gewähltem IC-Sockel („**Basic**“) werden die Spannungen auf andere Pins des 40-poligen IC-Sockels geschaltet:

Tabelle 2 Signale am Testsockel

Signal	*Vdd (5V)*	*Vpp (13V)*	*SCLK (5V)*	*SDATA (5V)*
8-/ 14-Pin	1	4	38	39
18-Pin / ICSP	36	4	34	35
28- /40-Pin	11&32	1	39	40

Bei ausgewähltem 8-/14-Pin-Sockel, liegt am Pin 40 eine Spannung von unter 0,25V an.
Am 5-poligen ICSP-Anschluss liegen die Spannungen gemäß meinem Standard an, wenn als IC-Sockel „**18 Pins / ICSP**“ gewählt wurde:

Tabelle 3 Signale am ICSP-Anschluss

Signal	*Vdd (5V)*	*Vpp (13V)*	*SCLK (5V)*	*SDATA (5V)*	*Vss (0V)*
ICSP-Pin	2	1	5	4	3

Vdd sollte einen Pegel von mindestens 4,5 V erreichen, SDATA und SCLK sollten über 4V liegen. Diese Pegel hängen von der Spannung im USB-Bus ab.

Beim Brenner8 liegt der Pegel von Vpp zwischen 10V und 20V. Er wird später noch durch eine Kalibrierung auf einen bestimmten Sollwert justiert. Diese Kalibrierung ist

in einem späteren Kapitel beschrieben, und sollte unbedingt vor den ersten Brennversuchen durchgeführt werden.

Bevor die Kalibrierung abgeschlossen ist, darf kein PIC in den Testsockel oder an den ICSP-Verbinder angeschlossen werden. Er könnte durch Überspannung zerstört werden.

6.9.1.2 Spannungspegel im Brenner9

Am 5-poligen ICSP-Anschluss liegen die Spannungen gemäß meinem Standard an, wenn als IC-Sockel „**18 Pins / ICSP**“ gewählt wurde:

Tabelle 4 Signale am ICSP-Anschluss

Signal	*Vdd (3,3V)*	*Vpp (3,3V)*	*SCLK (3,3V)*	*SDATA (3,3V)*	*Vss (0V)*
ICSP-Pin	2	1	5	4	3

Vdd sollte einen Pegel von mindestens 3 V erreichen, Vpp, SDATA und SCLK sollten über 3V liegen. Diese Pegel hängen (beim Brenner9) von der Spannung im USB-Bus ab.

Eine Kalibrierung ist für den Brenner9 nicht erforderlich.

6.9.2 Inbetriebnahme/Funktionstest unter Linux

Wurde libusb installiert und usburn aus den Quellen mit make compiliert, dann kann der Brenner8/9 an den PC angeschlossen und usburn probehalber ohne Optionen und Parameter aufgerufen werden.

6.9.2.1 Zugriffsrechte

usburn sucht nach einem am PC angeschlossenen „sprut-device“ und versucht dessen interface zu übernehmen (claimen). Kommt es dabei zu einem Problem (function not supported) dann hat nur root Zugriff auf das Interface dieses USB-Devices. Das lässt sich später ändern, aber zunächt hilft der „su“-Befehl weiter, mit dem man vorübergehend root-Rechte erlangt.

6.9.2.2 Firmware brennen

Befindet sich im Brenner bisher nur der Bootloader, so ist die Firmware mit dem Befehl

- `usburn -f --IN name.hex`

In den Brenner zu flashen. Natürlich ist für *name.hex* der richtige Filename des Firmwarefiles einzusetzen.

6.9.2.3 Brenner8 kalibrieren

Ein neuer Brenner8 muss noch kalibriert werden. Das erledigt man mit Hilfe eines Multimeters und dem Aufruf von

- `usburn --calibration`

Genauere Erläuterungen dazu befinden sich an anderer Stelle in diesem Handbuch.

6.9.2.4 Hardwaretest

Wenn Zweifel an der korrrekten Funktion des Brenners bestehen, dann kann man alle vom Brenner erzeugten Signale durch Aufruf von

- `usburn --test`

überprüfen. Dafür ist neben etwas Zeit auch ein Multimeter erforderlich.

6.10 ICSP-Adapter

Alle 14-Bit- und 16-Bit-Kern PICs (PIC12F6xx, PIC16Fxxx, PIC18Fxxxx) mit DIL-Gehäuse lassen sich im Testsockel des Brenner8 programmieren. 8-/14-/18-/28- und 40-Pin DIL-Typen sind dabei so in den Testsockel einzusetzen, dass Pin 1 des Chips in Pin 1 der Fassung steckt.

PICs in anderen Gehäusebauformen wie auch dsPIC30Fxxx-, PIC1xF5x und PIC10Fxxx-Typen müssen über einen Adapter an den 5-poligen ICSP-Steckverbinder angeschlossen werden.
Der Brenner8mini und Brenner9 haben keinen Testsockel. Somit werden für sie auch bei DIL-PICs ICSP-Adapter benötigt.

Auch wenn der PIC beim Brennen mit allen Pins in der Fassung des Brenners steckt, elektrisch sind mit dem Brenner nur 5 Leitungen verbunden. Das ist möglich, da der PIC mit Hilfe einer seriellen Datenübertragung programmiert wird - dem In Circuit Serial Programming (ICSP).

Dazu benötigt man:

1. eine Leitung für die ca. +12V-Programmierspannung (Brenner9: 3,3V)
2. eine Leitung für die +5V-Betriebsspannung (Brenner9: 3,3V)
3. eine Masseleitung
4. eine Datenleitung
5. eine Taktleitung

Diese 5 Leitungen des Brenners werden an folgende Pins des PIC angeschlossen:

Tabelle 5 Der ICSP-Anschluss

Nr.	Leitung des Brenners	Signalbezeichnung	Pin des PIC
1	Leitung für die +12V-Programmierspannung	Vpp	MCLR/Vpp (der Reset-Anschluss)
2	Leitung für die +5V-Betriebsspannung	Vdd	Vdd
3	Masseleitung	Vss	Vss
4	Datenleitung	Data	PGD (meist RB7)
5	Taktleitung	Clk	PGC (meist RB6)

Größere PICs haben mehrere Pins für Vdd und Vss. Der Hersteller empfiehlt, alle Pins zusammen zu verwenden. Oft (aber leider nicht immer) genügt aber auch die Verwendung von jeweils einem Vss und einem Vdd-Pin.

An der ICPS-Buchse stehen also alle Signale zur Verfügung, um einen PIC zum Programmieren an den Brenner anzuschließen. Ein Beispiel für die Nutzung der ICSP-Buchse sind Adapter, mit dem eine zusätzliche Schaltkreis-Fassung an einen beliebigen Brenner angeschlossen werden kann, wenn dieser über die ICPS-Buchse verfügt.

6.10.1 Grundregeln für ICSP-Adapter

Ist es wirklich nötig, über ein einfaches Kabel Worte zu verlieren? JA ES IST NÖTIG.

CLK-Schirmung
In der Belegung des Kabels gibt es eine Schwachstelle. Die für störende Einstreuungen sehr empfängliche Takt-Leitung (CLK, PGC) muss dringend von den anderen Leitungen abgeschirmt werden. Dazu ist nun keine komplette Schirmung nötig, aber eine separate Masseleitung zwischen CLK und DATA ist wenigstens erforderlich. Aus diesem Grunde verwende ich stets 6-poliges Hosenträgerkabel mit 2 Masseleitungen: eine zwischen Vdd und DATA und eine weitere zwischen DATA und CLK. Das ist in den untenstehenden Stromlaufplänen deutlich zu sehen.

Natürlich gibt es auch andere Lösungen, um ein Übersprechen auf die CLK-Leitung zu vermeiden, z.B. kann man die CLK-Leitung vom restlichen Kabel getrennt verlegen. Man kann auf die Masseleitung zwischen Vdd und DATA auch verzichten, ihre Funktion erfüllt ja auch die Masse zwischen DATA und CLK.
Da will ich keine weiteren Vorschriften machen. Hauptsache CLK ist vor Einstreuungen geschützt.

Wer diese einfache Regel missachtet, wird feststellen, dass schon die Autodetect-Funktion der Brennersoftware nicht funktioniert. Vom Brennen ganz zu schweigen.

Wie lang darf ein ICSP-Kabel eigentlich sein?
Es sollte so lang wie nötig und so kurz wie möglich sein. Wer nur einen zusätzlichen Sockel adaptieren will, kommt mit 10 cm aus. Für das programmieren eines PIC in der fertig aufgebauten Anwendungsschaltung sollten 20 cm auch genügen. Wenn CLK ordentlich geschirmt ist, sollte aber auch 1/2 Meter kein Problem sein. Zu lange Kabel und Kabel ohne jede CLK-Schirmung führen immer wieder zu Brennproblemen.

Nachfolgend einige Beispiele für ICSP-Adapter.

6.10.2 ICSP-Adapter für PIC12F6xx

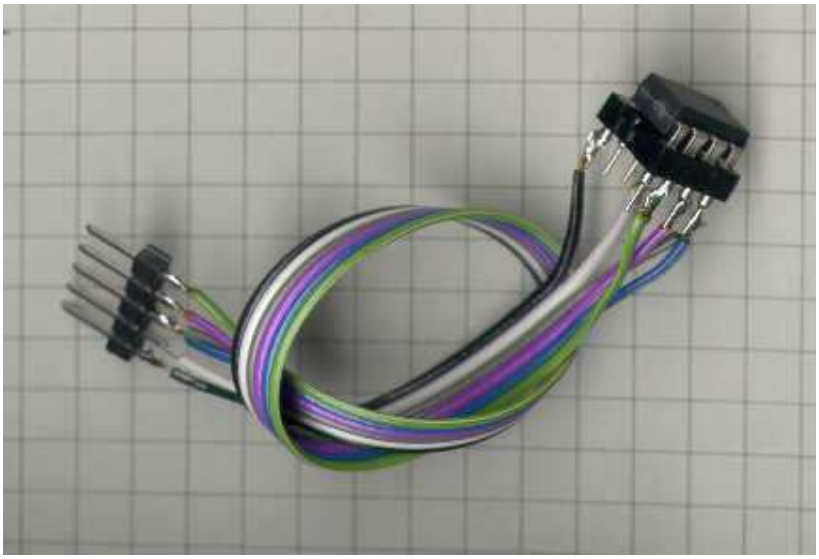


Abbildung 5 ICSP-Adapter für PIC12F6xx

Das obige Foto zeigt einen ICSP-Adapter für PIC12F6xx wie er für den Brenner8mini benötigt wird. (Der Brenner8 kann den PIC12F6xx im Testsockel brennen.) Er besteht nur aus einer 5-poligen Stiftleiste, die in die ICSP-Buchse des Brenners gesteckt wird, einer DIL-Fassung für den PIC und einem 6-adrigen Flachbandkabel. Die nachfolgende Abbildung zeigt den Stromlaufplan:

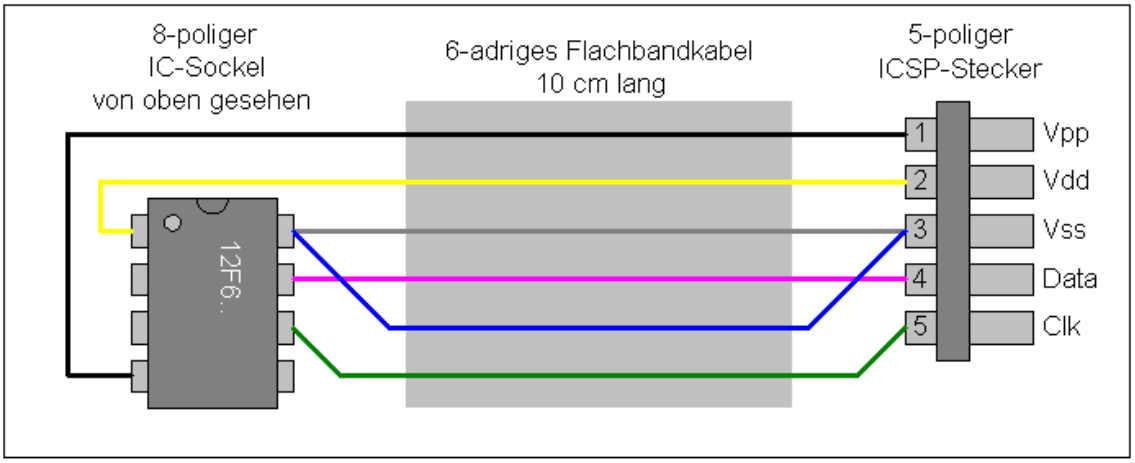


Abbildung 6 ICSP-Adapter für PIC12F6xx - Stromlaufplan

Es wird ein 6-adriges Kabel verwendet, da ich im Kabel zwei Masseleitungen (Vss) vorgesehen habe. Wichtig ist dabei vor allem die blaue Leitung. Sie schirmt die Taktleitung (Clk, PGC) von den anderen Leitungen. Das ist kein Luxus, sondern für die sichere Arbeit des Brenners nötig.

Dieser Adapter eignet sich NICHT für PIC10F2xx im 8-poligen DIL-Gehäuse, da diese eine andere Pinbelegung haben.

6.10.3 ICSP-Adapter für PIC im PLCC-Gehäuse

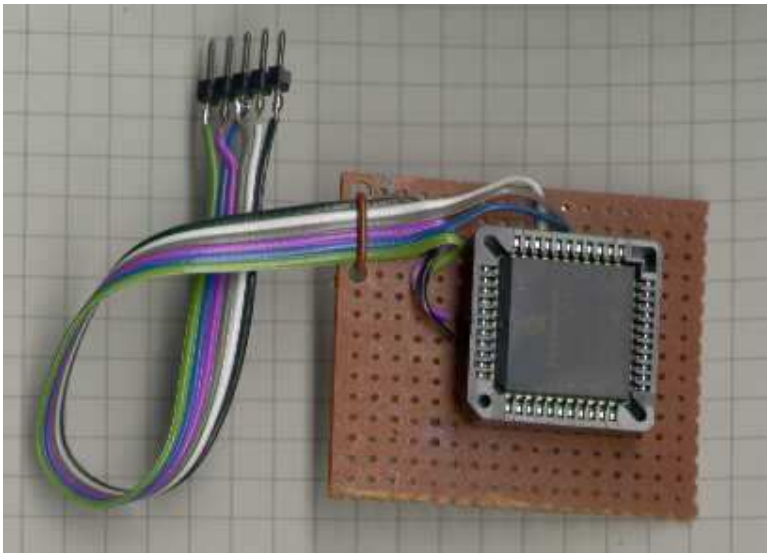


Abbildung 7 Adapter für PLCC-44

Das obige Foto zeigt einen ICSP-Adapter für PICs im PLCC-44-Gehäuse. Im konkreten Fall ist es ein PIC16F87xA-Typ.
Der Adapter besteht nur aus einer 5-poligen Stiftleiste, die in die ICSP-Buchse des Brenners gesteckt wird, einer Lochrasterplatine mit einer PLCC-44-Fassung für den PIC und einem 6-adrigen Flachbandkabel. Die nachfolgende Abbildung zeigt den Stromlaufplan:

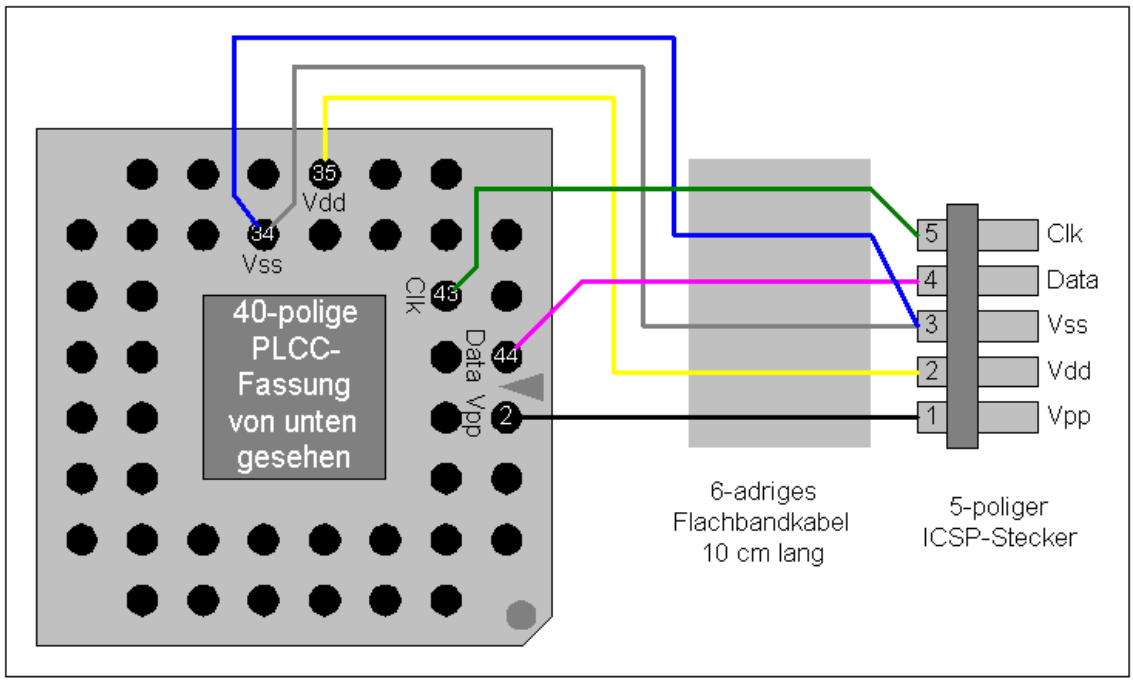


Abbildung 8 Adapter für PLCC-44 - Stromlaufplan

Die zum PIC12F6xx-Adapter gemachten Aussagen gelten auch hier.

Microchip empfiehlt generell alle Vss-Pins und alle Vdd-Pins anzuschließen. Bei vielen (aber nicht bei allen) PICs geht es auch mit nur je einem Pin. Wer aber einen Adapter neu baut, kann es ja gleich richtig machen, und der Microchip-Empfehlung folgen.

6.10.4 Universeller Programmieradapter für DIL-PICs

Dieser Adapter dient zum Programmieren beliebiger PIC-Controller im DIL-Gehäuse. Um einen PIC im Adapter zu programmieren, müssen zunächst 5 bis 7 Drahtbrücken in den Adapter eingesetzt werden. Dabei ist sorgfältig vorzugehen, da eine falsch gesetzte Brücke zur Beschädigung des Target-PICs führen kann. Wie die Drahtbrückern einzusetzen sind ist detailliert im Handbuch des Adapters beschieben.
Näheres findet sich auf http://www.sprut.de/electronic/pic/icsp/icsp.htm#universell

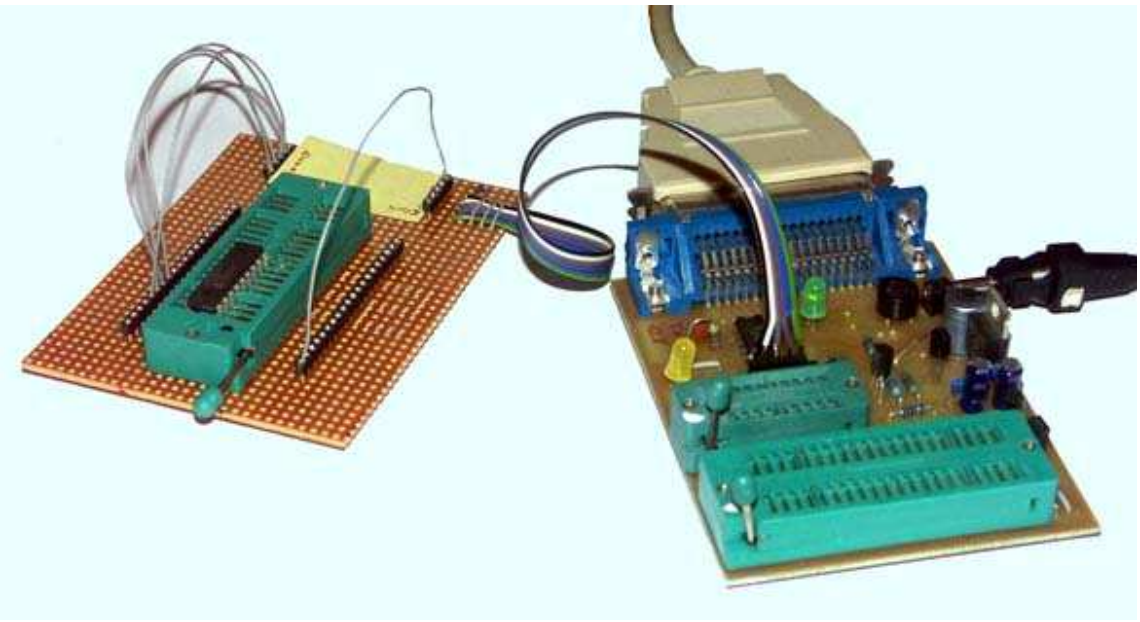


Abbildung 9 Universeller Programmieradapter (hier am Brenner5)

6.11 ICSP- Brennen in der fertigen Schaltung

Zum Brennen eines PIC in der fertig aufgebauten Schaltung eignen sich nur der Brenner8-P und der Brenner8mini-P.

Eine weitere Anwendung der ICPS-Verbindung ist das Brennen eines PIC, der bereits in seine Anwendungsschaltung eingebaut ist.

Dazu verfügt die Leiterplatte der Anwendungsschaltung des PIC auch über eine ICPS-Buchse. Brenner und Anwenderschaltung werden über ein 5-poliges Kabel miteinander verbunden, und der PIC wird "zuhause" gebrannt. Das lästige Umstecken des PIC zwischen Anwendungsschaltung und Brenner entfällt, und das komfortable Brennen von PICs im SMD-Gehäuse wird überhaupt erst möglich.

Microchip verwendet auf seiner Test-Platinen einen 6-polige Westernbuchse, die mit ICD oder ICSP beschriftet ist. Die ersten 5 Pins dieses Anschlusses entsprechen den 5-Pins meines ICSP-Anschlusses, der 6. Pin der Western-Buchse ist reserviert.

6.11.1 Entwurf einer ICSP-tauglichen Schaltung

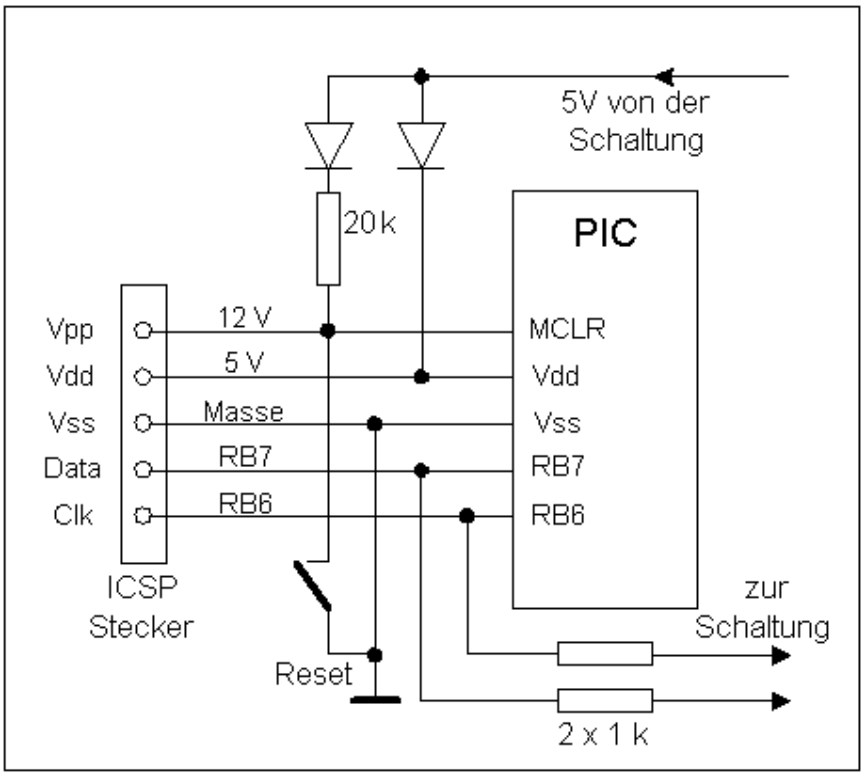


Abbildung 10 ICSP-taugliche Schaltung

Die 5 Pins, die zum ICSP an den Brenner angeschlossen werden müssen, dienen ja nicht exklusiv zum Brennen, sie werden meist auch in der Anwendungsschaltung verwendet.
Um zu verhindern, dass sich Brenner und Anwendungsschaltung gegenseitig in ihrer Funktion stören, sind einige Dinge beim Entwurf der Anwendungsschaltung zu beachten:

6.11.1.1 Programmierspannung MCLR/Vpp

Dieser Anschluss ist am schwierigsten.

Meine Brenner8 (bis Rev. 2) ziehen dieses Pin über einen 10k-Widerstand auf Masse, oder über einen Schalttransistor auf +12V.
In der Anwenderschaltung wird dieses Pin mit einem Hochziehwiderstand auf 5V gehalten, oder mit einem Resettaster kurzfristig auf Masse gelegt. Dass der Reset-Taster beim Brennen keinesfalls gedrückt werden darf ist damit klar!!

Schwierig ist aber auch der Hochziehwiderstand. Beim Brennen trennt nur dieser Widerstand die 5-V-Versorgung der Anwendungsschaltung von den 12V des Brenners. Hier muss deshalb eine Diode vor Schäden durch Überspannung schützen.

Der 5V-Hochziehwiderstand muss deutlich größer sein (mindestens 20 X) als der Widerstand, der im Brenner das MCLR-Pin mit Masse verbindet. Ansonsten kann der Brenner MCLR nicht sauber auf Masse ziehen. Wird ein Brenner8 (bis Rev. 2)

verwendet, sollte der Hochziehwiderstand etwa 200kOhm betragen. Microchip empfiehlt aber einen Wert von 40 kOhm.

Eine Alternative ist die in der Abbildung gezeigte Trennung von ICSP-Vdd von der 5V-Versorgung der Schaltung mit Dioden. Sie erlaubt die Nutzung deutlich kleinerer Hochziehwiderstände.

Ab der Revision 3 kann die P-Version des Brenner8 das MCLR-Pin mit einem Transistor aktiv nach Vss ziehen. Damit entfällt dieses Problem.

6.11.1.2 Betriebsspannung Vdd

Beim Brennen speist der Brenner den PIC mit der nötigen Betriebsspannung. Ist der PIC der einzige Spannungsverbraucher in der Anwenderschaltung, kann die +5V-Leitung des Brenners direkt mit dem Vdd-Pin des PIC verbunden werden. Vor dem Anschluss des Brenners muss dann unbedingt die normale Betriebsspannung des PIC abgeschaltet werden.

Sind neben dem PIC noch andere Bauelemente mit der 5-V-Versorgung der Anwendungsschaltung verbunden, würde der Brenner bei Brennen die gesamte Anwendungsschaltung in Betrieb nehmen. Bei größeren Schaltungen könnte das den Brenner überlasten. Eine Entkopplung mit Shottky-Dioden oder ein Jumper in der +5V-Leitung trennen dann besser die beiden potentiellen 5-V-Quellen.
Wird mit einer Shottky-Diode entkoppelt, dann ist die Vdd des PICs im Normalbetrieb ca. 0,2V kleiner als Vdd der restlichen Schaltung. Meist ist das unkritisch, aber wenn Vdd z.B. als positive Referenzspannung des ADC verwendet wird, kann es zu Messwertverfälschungen des ADC kommen. In diesem Fall ist ein Jumper der Diode vorzuziehen.

Größere PICs besitzen mehrere Vdd-Pins. Zum Programmieren sind alle diese Pins untereinander zu verbinden, was in der Anwenderschaltung in der Regel ohnehin gegeben ist.

6.11.1.3 Masseverbindung Vss

Das ist die einzige unkritische Verbindung. Normalerweise wird die Masse des Brenners direkt mit der Masse des PIC und damit auch mit der Masse der Anwenderschaltung verbunden.
Größere PICs besitzen mehrere Vss-Pins. Zum Programmieren sind alle diese Pins untereinander zu verbinden, was in der Anwenderschaltung in der Regel ohnehin gegeben ist.

6.11.1.4 Takt- und Datenleitung (PGC und PGD)

PGD und PGC sind bei den meisten PICs gleichzeitig die Port-Pins RB6 und RB7.

Wer in der Anwendungsschaltung auf diese beiden Pins verzichten kann, sollte sie exklusiv der ICSP-Schnittstelle zur Verfügung stellen. Werden die beiden Pins aber benötigt, sollten sie mit der ICSP-Buchse direkt, aber mit dem Rest der Schaltung über Widerstände von wenigstens 1 kOhm verbunden werden. Ist so ein 1 kOhm Widerstand für die Applikationsschaltung zu groß, helfen nur noch Jumper, die vor dem Brennen geöffnet werden müssen, um den PIC von der restlichen Schaltung zu trennen.

7 Treiberinstallation (nur Windows)

Nachdem der Brenner8/9 fertig aufgebaut wurde, und der Steuer-PIC die korrekte Firmware (der wenigstens der Bootloader) eingebrannt bekam, muss er nun noch unter Windows eingerichtet werden.

Der Brenner8/9 benötigt den **Microchip Custom Driver** (mpusbapi.dll). Den findet man auf der Microchip Homepage, oder im Softwarepaket US-Burn auf meiner Homepage. Da US-Burn ohnehin für den Brenner8 benötigt wird, lädt man sich das US-Burn-Zip-File von meiner Homepage,

http://www.sprut.de/electronic/soft/usburn/usburn.htm - download

und entpackt es in einen Ordner auf der lokalen Festplatte.
Der Treiber liegt dann im Unterordner **driver** .

Nun kann der Brenner8/9 an den Windows-PC angesteckt werden. Windows findet automatisch das ihm noch unbekannte USB-Gerät **PIC-Brenner8 (sprut 2006)** ausgibt.

Abbildung 11 Neue Hardware gefunden

Anschließend fordert Windows zur Treiberinstallation auf.

Abbildung 12 Hardware Assistent 1

Nach einem Klick auf **Weiter** ist man im Assistenten für die Hardwareinstallation.

Abbildung 13 Hardware Assistent 2

Hier wählt man die untere der beiden möglichen Optionen aus, um den Treiber manuell auswählen zu können. Danach klickt man auf **Weiter**.

Nun muss man den Brenner in eine Geräteklasse einordnen. Da er nicht so richtig in eine der üblichen Gruppen passt, wählt man **Andere Geräte**, und klickt auf **Weiter**.

Abbildung 14 Hardware Assistent 3

Nun kommt man zur Auswahl des Gerätetreibers. Hier klickt man auf die Schaltfläche **Datenträger...**

Abbildung 15 Gerätetreiber auswählen 1

Im folgenden Fenster muss nun der Pfad zum Verzeichnis mit dem Treiber eingestellt werden.

Abbildung 16 Pfad zum Treiber einstellen

Windows schaut in diesem Verzeichnis nach, und finden einen passenden Treiber für ein **PIC-Brenner8 (sprut)**.

Abbildung 17 Gerätetreiber auswählen 2

Im folgenden Fenster wird die Auswahl mit **Weiter** bestätigt.

Abbildung 18 Treiber kann installiert werden

Windows installiert nun den Treiber. Mit einem Klick auf **Fertig stellen** wird die Installation abgeschlossen.

Abbildung 19 Assistent fertig stellen

Von nun an findet man den Brenner8 im Gerätemanager.

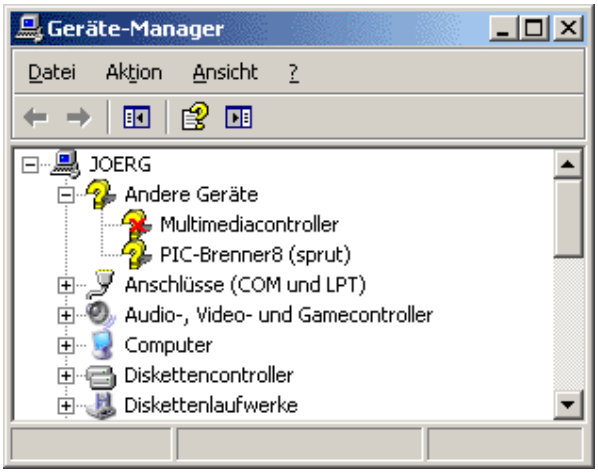


Abbildung 20 Gerätemanager

Im Gerätemanager sollte dem Betriebssystem verboten werden, den Brenner8 abzuschalten um Energie zu sparen. Ansonsten kann es passieren, dass der Brenner8 extrem langsam arbeitet.

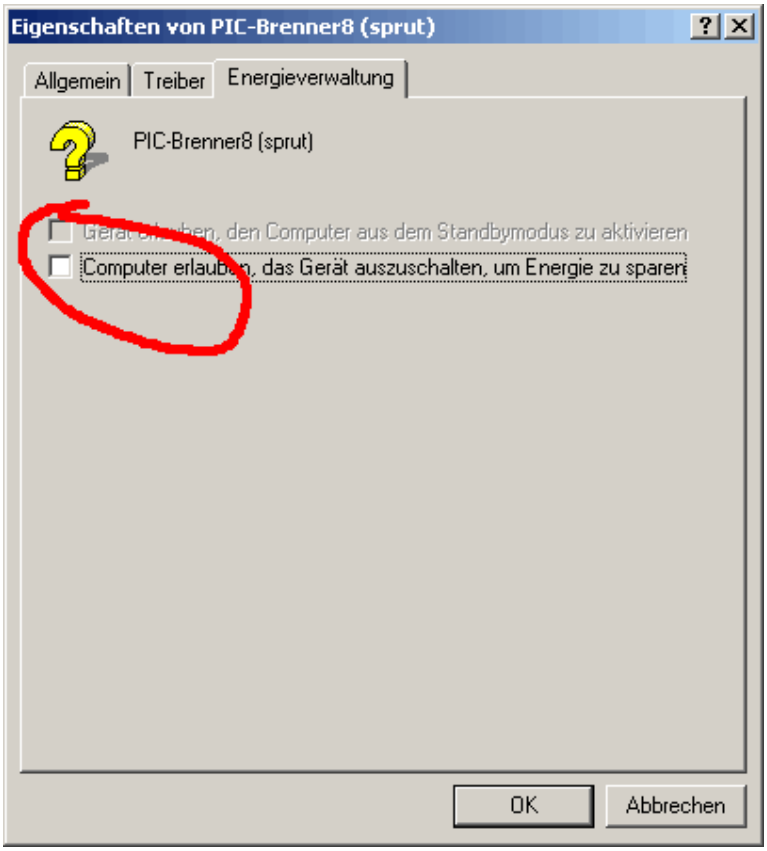


Abbildung 21 Gerätemanager - Energiesparoption

Damit ist der Brenner8 einsatzbereit.

8 Kalibrierung des Brenner8

Für den Brenner9 ist eine Kalibrierung nicht erforderlich.

Nachdem der Brenner8 fertig aufgebaut wurde, der Steuer-PIC die korrekte Firmware eingebrannt bekam und der USB-Treiber eingerichtet wurde, muss die Programmierspannungserzeugung des Brenners noch kalibriert werden.

Ein unkalibrierter Brenner8 ist in der Lage jeden PIC innerhalb kürzester Zeit zu zerstören! Es kann problemlos eine Programmierspannung von Vpp=25V erzeugt werden. Das überlebt kein PIC!!

Der Brenner8 erzeugt die Programmierspannung Vpp mit Hilfe eines kleinen Schaltreglers. Per Software kann die Höhe der Spannung variiert werden. Damit ist garantiert, dass jeder PIC die für ihn optimale Programmierspannung bekommt.

Das funktioniert aber nur optimal, wenn der Steuer-PIC des Brenners die Höhe der Programmierspannung auch genau messen kann.

Die Spannung wird im Brenner8 über einen Spannungsteiler gemessen, und mit einer Referenzspannung (von einer Z-Diode) verglichen.
Bei der Kalibrierung werden die Z-Spannung und das Spannungsteilerverhältnis ermittelt.

8.1 Kalibrierung unter Windows

Benötigt werden:

- Brenner8
- US-Burn-Software
- Voltmeter

8.1.1 Vorbereitung

Den Brenner8 am PC anschließen.
US-Burn im PC starten
In US-Burn auf die Registerkarte **Options** – **Hardware** wechseln.

Die nötigen Einstellungen werden in der „Box“ **programming voltage Vpp calibration** vorgenommen, und erfolgen in drei Schritten

1. Einstellung der Z-Spannung
2. Einstellung des Spannungsteilerverhältnisses
3. Automatische Reglereinstellung

8.1.2 Schritt Nr. 1: Z-Spannung

Im Stromlaufplan ist eine 3,3V-Z-Diode vorgesehen, aber typische Z-Dioden haben eine Toleranz von 10%. Man kann sich auf die Spannungsangabe also nicht verlassen.
Die Spannung über der Z-Diode D3 wird mit dem Voltmeter gemessen. (Am Brenner8P z.B. zwischen den Lötstiften LSP2 und LSP3.) Der Spannungswert wird

dann im Feld **Z-voltage** eingestellt. Mit den beiden Pfeil-Schaltflächen kann die Spannung zwischen 2V und 4V in 0,01V-Schritten angepasst werden.
Damit der neu eingetragene Wert wirksam wird, wird abschließend auf die Schaltfläche **apply** geklickt. Im Textfenster der **BASIC** –Registerkarte erscheint daraufhin ein Kalibrierwert, der in der Regel etwas kleiner als 1 ist.

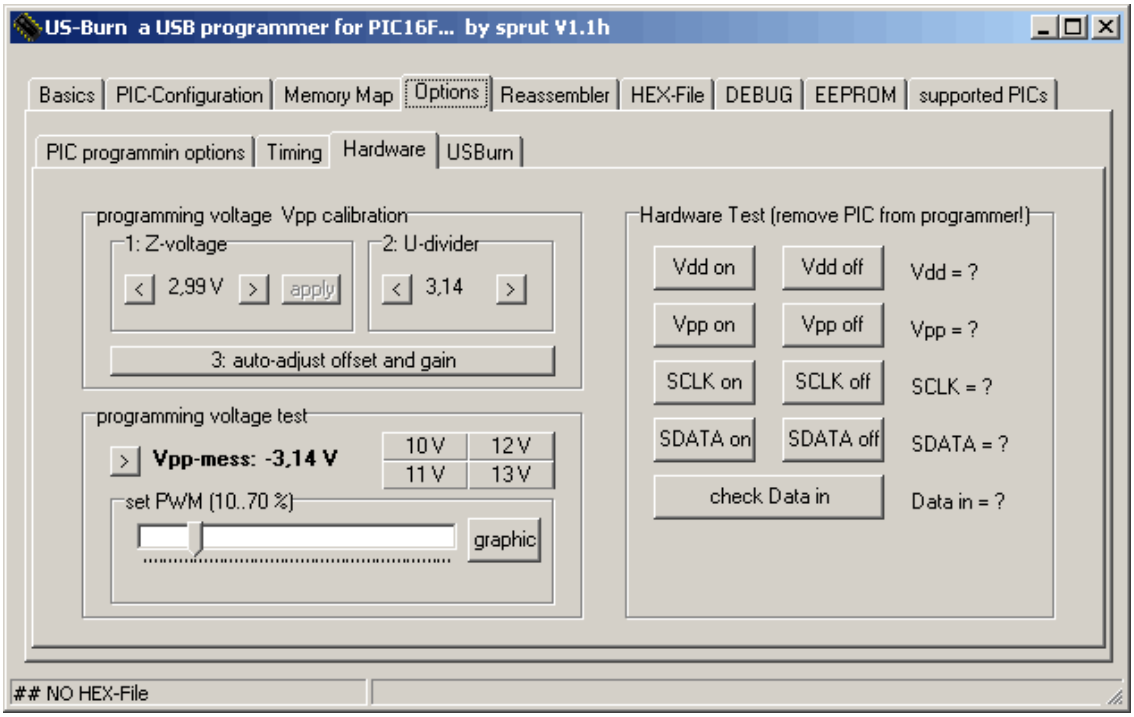


Abbildung 22 USBurn - Options-Hardware

8.1.3 Schritt Nr. 2: Spannungsteiler

Der Spannungsteiler zum Messen der Vpp-Spannung besteht aus den Widerständen R4 und R5. Sein Spannungsteilerverhältnis ist theoretisch 3,14. In der Praxis kann der Wert aber abweichen. Die Einstellung erfolgt im Feld **U-divider**.

Zur Vorbereitung schließt man das Voltmeter zwischen der Kathode von D1 und Vss an (alternativ über C5). (Am Brenner8P z.B. zwischen den Lötstiften LSP1 und LSP3.) Mit dem Schieberegler **set PWM** stellt man eine Spannung von etwa 13V (auf dem Multimeter) ein. (Die Spannung sollte möglichst hoch sein, darf aber 14V auf keinen Fall überschreiten!) In einigen USBurn-Versionen läßt sich der Schieberegler nicht bedienen, dann ist die vorhandene Spannung so zu nutzen wie sie ist.

Der Brenner8 misst die Spannung auch, und zeigt seinen Messwert als **Vpp-mess** an. Durch Ändern von **U-divider** wird nun der vom Brenner8 gemessene Spannungswert dem Messwert des Multimeters möglichst gut angenähert.

8.1.4 Schritt Nr. 3: Reglereinstellung

Abschließend muss USBurn das Verhalten des Reglers bestimmen, um später möglichst schnell genaue Spannungen einstellen zu können. Dazu klickt man einfach auf die Schaltfläche **auto-adjust offset and gain**. Der Brenner macht nun selbständig alle nötigen Messungen, was ca. 6 Sekunden dauert. Da dabei u.U. recht

hohe Spannungen erzeugt werden, darf sich während des Tests im Testsockel und am ICSP-Anschluss des Brenners kein PIC befinden.
Im Textfenster der **BASIC** –Registerkarte erscheint daraufhin einige Zahlenwerte, die im Falle einer Fehlersuche hilfreich wären.

Zur Prüfung des Ergebnisses kann man nun die Schaltflächen **10V**, **11V**, **12V** und **13V** anklicken. Daraufhin sollte eine Programmierspannung eingestellt werden, die diesen Werten etwa entspricht. Der typische Einstellfehler beträgt ca. 0,3V.

8.1.5 Fertig

Die Kalibrierdaten werden im Brenner dauerhaft gespeichert. Beim Schließen des Programms US-Burn werden zusätzlich Sicherheitskopien der Werte in der Datei **usburn.ini** gespeichert, und stehen beim nächsten Programmstart wieder zur Verfügung.

8.1.6 Fehlersuche

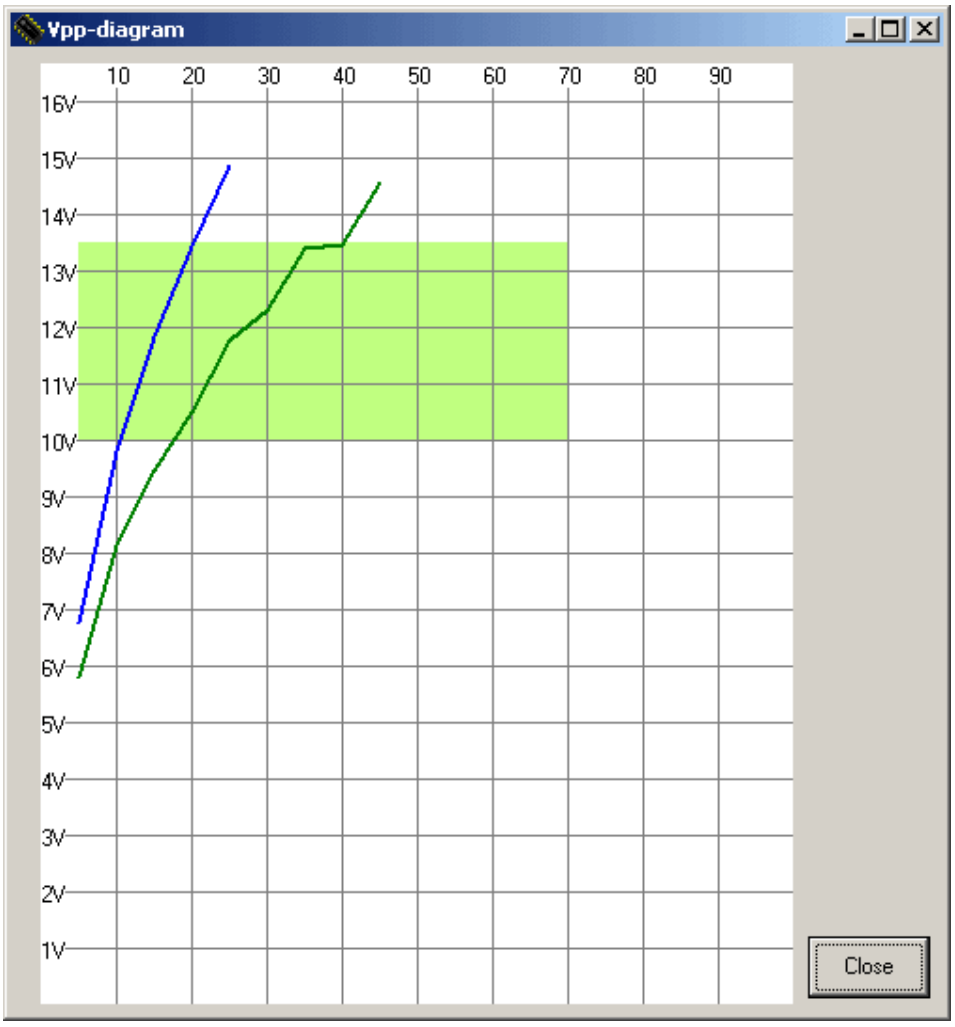


Abbildung 23 Vpp-Diagramm - normal

In der Box **programming voltage test** gibt es die Schaltfläche **graphic**. Wird sie angeklickt, dann erzeugt US-Burn eine Grafik der Reglerausgangsspannungen. Die Grafik zeigt die Vpp-Spannung für unterschiedliche Regler-Taktverhältnisse mit Vpp-

off (blaue Linie) und Vpp-on (grüne Linie). Die Erstellung der Grafik dauert wenige Sekunden.

Da dabei u.U. recht hohe Spannungen erzeugt werden, darf sich während des Tests im Testsockel und am ICSP-Anschluss des Brenners kein PIC befinden.

Da der Brenner8 nur Spannungen bis ca. 15V messen kann, werden auch keine höheren Werte im Diagramm eingetragen.
Die hellgrüne Fläche ist der normale Arbeitsbereich des Reglers. Beide Kennlinien sollten diese Fläche von unten nach oben durchqueren.

Die folgende Grafik zeigt einen Brenner8 mit einem schlechten Spannungsregler. Die erzeugten Spannungen liegen zu niedrig. Die Fehlerursache kann nur ein Hardwareproblem sein, wahrscheinlich wurde als Diode D1 ein ungeeigneter Typ ausgewählt.

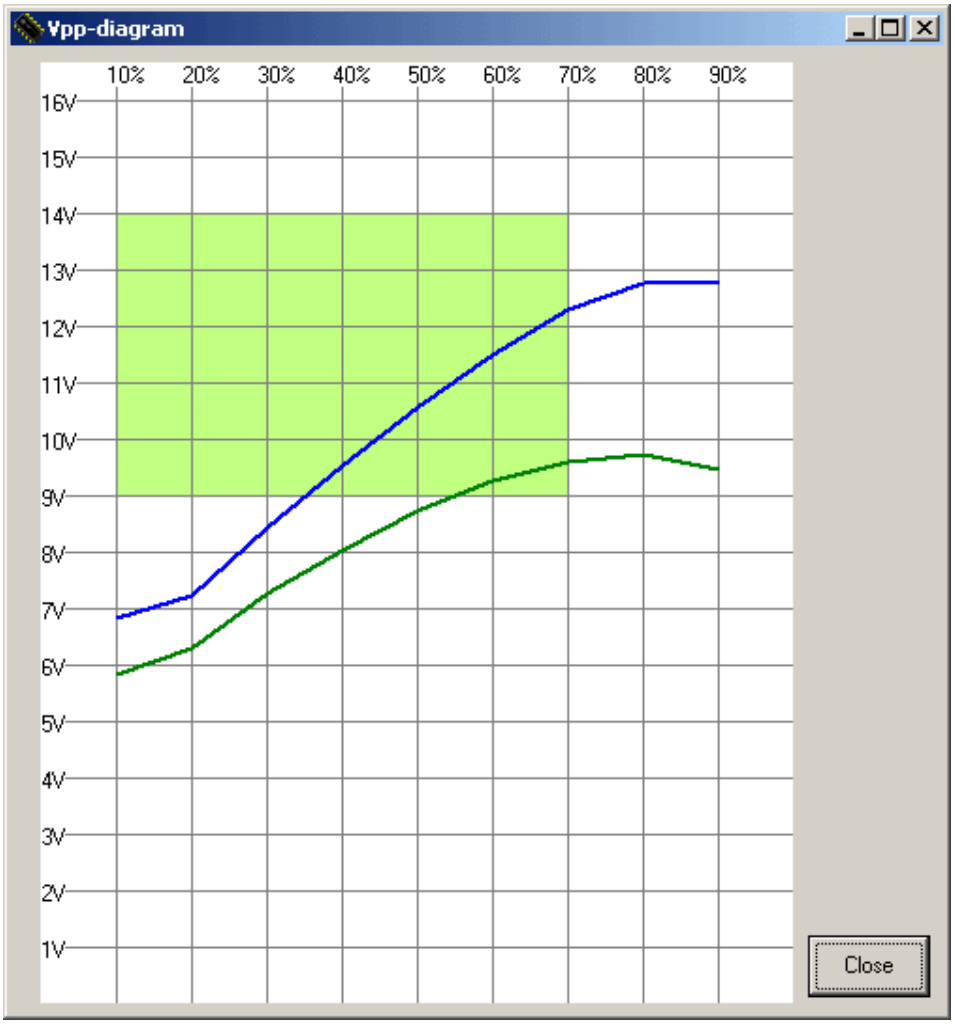


Abbildung 24 Vpp-Diagramm - Spannung zu klein

8.2 Kalibrierung unter Linux

Benötigt werden:

- Brenner8
- usburn-Software
- Voltmeter

8.2.1 Vorbereitung

Den Brenner8 am PC anschließen.

Die Kalibrierung erfolgen in drei Schritten

1. Einstellung der Z-Spannung
2. Einstellung des Spannungsteilerverhältnisses (Div)
3. Automatische Reglereinstellung

8.2.2 Schritt Nr. 1: Z-Spannung

Im Stromlaufplan ist eine 3,3V-Z-Diode vorgesehen, aber typische Z-Dioden haben eine Toleranz von 10%. Man kann sich auf die Spannungsangabe also nicht verlassen.
Die Spannung über der Z-Diode D3 wird mit dem Voltmeter gemessen. (Am Brenner8P z.B. zwischen den Lötstiften LSP2 und LSP3.)

Usburn mit der Option -k oder --calibration starten. Im Terminal zeigt usburn den intern gespeicherten Wert der Z-Spannung an. Durch Eingabe von „+“ und „-“-Zeichen (dahinter das ENTER nicht vergessen ;-)) wird der Wert geändert und der vom Multimeters gemessene Spannungswert möglichst gut angenähert.
Durch Eingabe eines „=“-Zeichens wird dieser Kalibrierschritt beendet.

8.2.3 Schritt Nr. 2: Spannungsteiler

Der Spannungsteiler zum Messen der Vpp-Spannung besteht aus den Widerständen R4 und R5. Sein Spannungsteilerverhältnis ist theoretisch 3,14. In der Praxis kann der Wert aber abweichen.

Zur Vorbereitung schließt man das Voltmeter zwischen der Kathode von D1 und Vss an (alternativ über C5). (Am Brenner8P z.B. zwischen den Lötstiften LSP1 und LSP3.)

Der Brenner8 misst die Spannung auch, und zeigt seinen Messwert als **Vpp-mess** an.
Durch Eingabe von „+“ und „-“-Zeichen (auch hier wieder mit ENTER abschließen) wird der Div-Wert geändert und Vpp-mess dem vom Multimeters gemessenen Spannungswert möglichst gut angenähert.
Durch Eingabe eines „=“-Zeichens wird dieser Kalibrierschritt beendet.

8.2.4 Schritt Nr. 3: Reglereinstellung

Abschließend muss USBurn das Verhalten des Reglers bestimmen, um später möglichst schnell genaue Spannungen einstellen zu können.
Der Brenner macht nun selbständig alle nötigen Messungen, was ca. 6 Sekunden dauert. **Da dabei u.U. recht hohe Spannungen erzeugt werden, darf sich während des Tests im Testsockel und am ICSP-Anschluss des Brenners kein PIC befinden.**
Im Terminal erscheinen daraufhin einige Zahlenwerte, die im Falle einer Fehlersuche hilfreich wären.

Zur Prüfung des Ergebnisses erzeugt der Brenner8 nun nacheinander **10V**, **11V**, **12V** und **13V** jeweils nach einem Druck auf die Enter-Taste. Daraufhin sollte jeweils die ensprechende Programmierspannung eingestellt werden. Der typische Einstellfehler beträgt ca. 0,3V.

9 Indikator LEDs

Der Brenner8 hat zwei LEDs: eine grüne und eine gelbe. An ihnen lässt sich der Betriebszustand ablesen:

9.1 Normalbetrieb

Unmittelbar nach dem Anstecken des Brenner8 an den PC leuchtet die grüne LED auf, gefolgt von der gelben LED. In der gleichen Reihenfolge verlöschen beide LEDs auch wieder nach jeweils 0,5 Sekunden Leuchtzeit.

Beim Start von USBurn (Windowsversion) blinken beide LEDs 2 mal kurz auf. Wird unter windows mit mehreren Brennern gearbeitet, und in USBurn auf einen anderen Brenner umgeschaltet, dann blinken beide LEDs des neu ausgewählten Brenners 2 hintereinander 2 mal kurz auf.

Im Normalbetrieb leuchtet dann die gelbe LED bei jedem Zugriff auf den zu brennenden PIC auf. Bei kurzen Zugriffen ist das kurze Flackern allerdings kaum zu sehen. Die grüne LED bleibt dunkel.

9.2 Havarie

Leuchtet während des normalen Betriebs die grüne LED auf (und bleibt dauerhaft an), dann gab es eine Fehlfunktion in der Programmierspannungserzeugung. Der Brenner hat daraufhin die Programmierspannung abgeschaltet, um den zu programmierenden PIC vor der Zerstörung zu schützen. Der Brenner ist vom PC zu trennen und dann wieder anzuschließen. Die Fehlerursache ist durch Kalibrierung zu beseitigen.
Es wäre natürlich logischer, wenn diese LED rot wäre, aber das möchte ich im Nachhinein nicht mehr ändern, um Verwirrung zu vermeiden.

9.3 Bootloader

Startet der Bootloader, so leuchten beide LEDs dauerhaft.

10 US-Burn for Windows

US-Burn ist ein Windowsprogramm, dass das HEX-File analysiert, und die darin enthaltenen Daten via USB-Anschluß an den Brenner8 überträgt. Darüber hinaus bietet US-Burn noch einige Extras:

- Freie Konfiguration des PIC
- HEX-File-Betrachter
- Reassembler
- EEPROM-Daten-Anzeige
- Kalibrierung des Brenner8
- Laden neuer Firmware in den PIC (mit Bootloader)

10.1 Voraussetzungen für die Nutzung von US-Burn

10.1.1 Software

US-Burn ist unter Win98/ME/NT/2000 und XP lauffähig. Ich teste die Software aber nur noch unter Win2k und WinXP.
Der Betrieb unter Vista sollte möglich sein, wird von mir aber nicht getestet.

10.1.2 Daten

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

10.1.3 Hardware

Das Programm US-Burn benötigt als Hardware den USB-Port-Brenner „Brenner8/9“ (www.sprut.de/electronic/pic/projekte/brenner8/) .

- Brenner8
- Brenner8mini
- Brenner9

Brenner8: Mein momentaner Standard für den USB-Port
http://www.sprut.de/electronic/pic/projekte/brenner8/index.htm

Brenner8mini Ein vereinfachter Brenner8 der nur über ICSP brennt
http://www.sprut.de/electronic/pic/projekte/brenner8mini/index.htm

Brenner9: Ein Brenner für 3,3V-PICs der nur über ICSP brennt
http://www.sprut.de/electronic/pic/projekte/brenner9/index.htm

10.2 Installation

US-Burn ist unter Win98/ME/NT/2000 und XP lauffähig.

Das Programm US-Burn steht als ZIP-Archiv zum Download bereit. Dieses Archiv mit dem Namen usburnxx.ZIP enthält die folgenden Dateien.

- usburnxx.exe Das Hauptprogramm in der aktuellen Version
- readme.txt Kurzanleitung
- usburn.hlp die Help-Datei
- mpusbapi.dll DLL-Datei für den USB-Zugriff (von Microchip)
- Picdef3.dll DLL-Datei für PIC-Typen-Verwaltung
- picdef03.dat PIC-Datenfile
- setdef03.dat PIC-Datenfile
- fildef03.dat PIC-Datenfile
- cfgdef03.dat PIC-Datenfile
- cekdef03.dat PIC-Datenfile
- texdef03.dat PIC-Datenfile

sowie den Ordner:

- Driver mit dem USB-Treiber (von Microchip)

Die 6 PIC-Datenfiles sind zusammen die PIC-Database.

Bevor das Programm US-Burn benutzt werden kann, ist es zusammen mit allen anderen Files des ZIP-Verzeichnisses in ein gemeinsames Verzeichnis zu kopieren. US-Burn legt beim Erststart in diesem Verzeichnis eine Datei **usburn.ini** an, in der das Programm bestimmte Einstellungen speichert. US-Burn schreibt nichts in die Registry.

Um den Brenner8 benutzen zu können ist der im Unterverzeichnis **Driver** enthaltene USB-Treiber zu installieren. Das erfolgt durch Aufruf der Datei **mchpusb.inf** in diesem Ordner, oder nach Anschluss des Brenners8 an den PC nach Anleitung von Windows. (siehe voriges Kapitel)
Es gibt zwei unterschiedliche Treiber zur Auswahl. Der Treiber im Unterordner **driver_98_me_2k_xp** ist nicht Vista-kompatibel, er unterstützt dafür aber noch 16-Bit-Windowsvarianten. Der Treiber im Unterordner **driver_2k_xp_vista** unterstützt kein 16-Bit-Windows, dafür aber Windows-Vista.

10.3 Schnellstart

- Alle Dateien aus dem gezipten File in eine Verzeichnis kopieren
- Brenner8 an PC anschließen
- Treiber installieren
- Programm US-Burn starten
- PIC in Brenner einsetzen
- Im Programmfenster PIC-Sockelgröße und PIC-Familie auswählen.
- ‚**Identify PIC in Programmer**‘ drücken
- Hex-File in US-Burn laden: ‚**Select HEX-File as source**’
- Falls nötig Config-Einstellungen anpassen
- ‚**Write HEX-File into PIC**‘ Taste drücken
- fertig

10.4 Grundlagen des PIC-Brennens

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 (ca.) 12-V-Programmierspannung und natürlich die Masseverbindung.

Im Brenner8 werden die Taktleitung, die Datenleitung und die 5V-Leitung direkt vom Steuer-PIC erzeugt.
Die Programmierspannungsleitung wird mit Transistoren geschaltet.

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.

Der Brenner8 hat einen 40-poliger Testsockel, in dem sich alle 14- und 16-Bit-PICs mit DIL Gehäuse programmieren lassen. Die PICs sind jeweils so in den Sockel einzusetzen, dass Pin1 des PICs in Pin 1 der Fassung sitzt.

Der Brenner8 wird zukünftig auch für 12-Bit-PICs und dsPIC30-Typen erweitert werden. Allerdings werden dsPICs sowie 8-Pin-12-Bit-PICs nicht im Testsockel zu brennen sein, sondern nur über ein ICSP-Kabel mit passendem Adapter

10.5 Bedienung des Programms

Beim Programmstart öffnet US-Burn den installierten USB-Treiber, sucht und findet den Brenner8 und meldet sich dann mit seinem Programmfenster. Falls dabei aber etwas nicht funktioniert, kann es zu einer Fehlermeldung kommen.

10.5.1 Fehlender USB-Treiber

Das Programm US-Burn benötigt den USB-Treiber von Microchip, der im US-Burn-ZIP-Archiv beiliegt. Falls das Programm den Treiber nicht findet, meldet es beim Start einen Fehler
Eine Arbeit mit dem Programm ist nicht möglich, bis der Treiber installiert wurde.

10.5.2 Fehlender Brenner

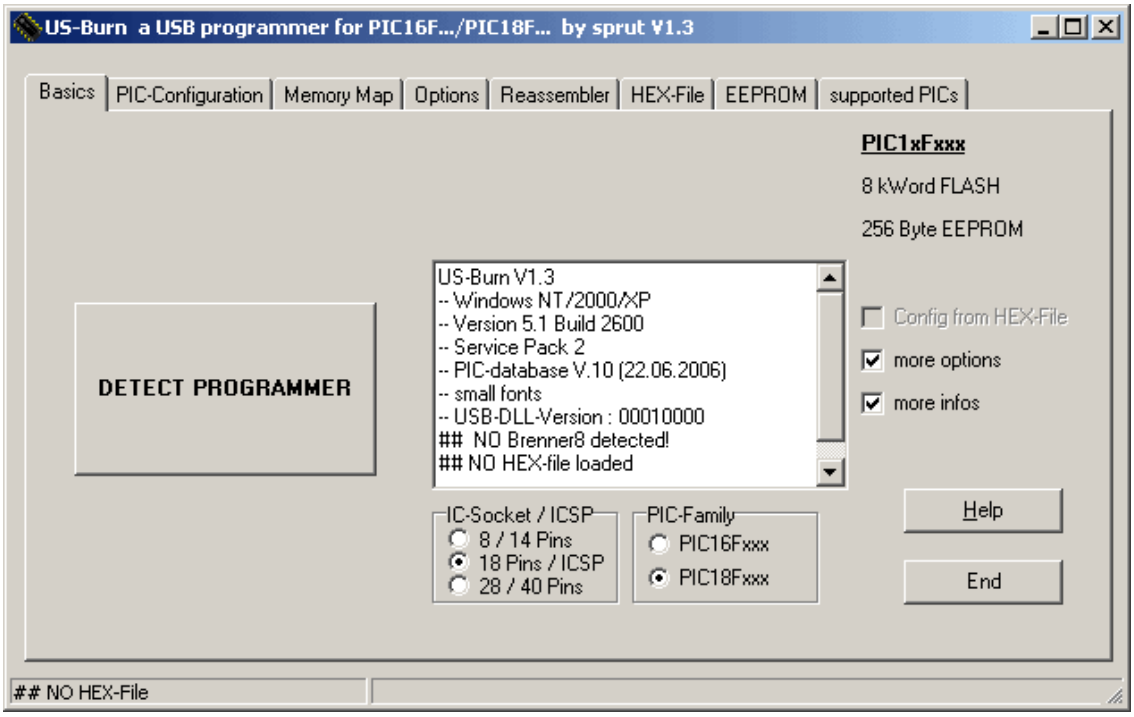


Abbildung 25 Fehlermeldung bei fehlendem Brenner

Beim Programmstart überprüft US-Burn die USB-Ports auf einen angeschlossenen Brenner8 hin ab. Deshalb sollte der Brenner zu diesem Zeitpunkt an das USB-Port angeschlossen sein. Falls das Programm keinen Brenner findet, meldet es sich mit obigem Fenster:

Nach dem Klicken auf “**DETECT PROGRAMMER**” wiederholt das Programm die Suche. Wird nun ein Brenner gefunden (weil er nachträglich angeschlossen wurde) meldet sich das normale Programmfenster.

Es gibt ein vereinfachtes und ein vollständiges Programmfenster. Das vereinfachte Programmfenster stellt nur die wirklich nötigen Funktionen zur Verfügung. Es vereinfacht dadurch die Bedienung des Programms

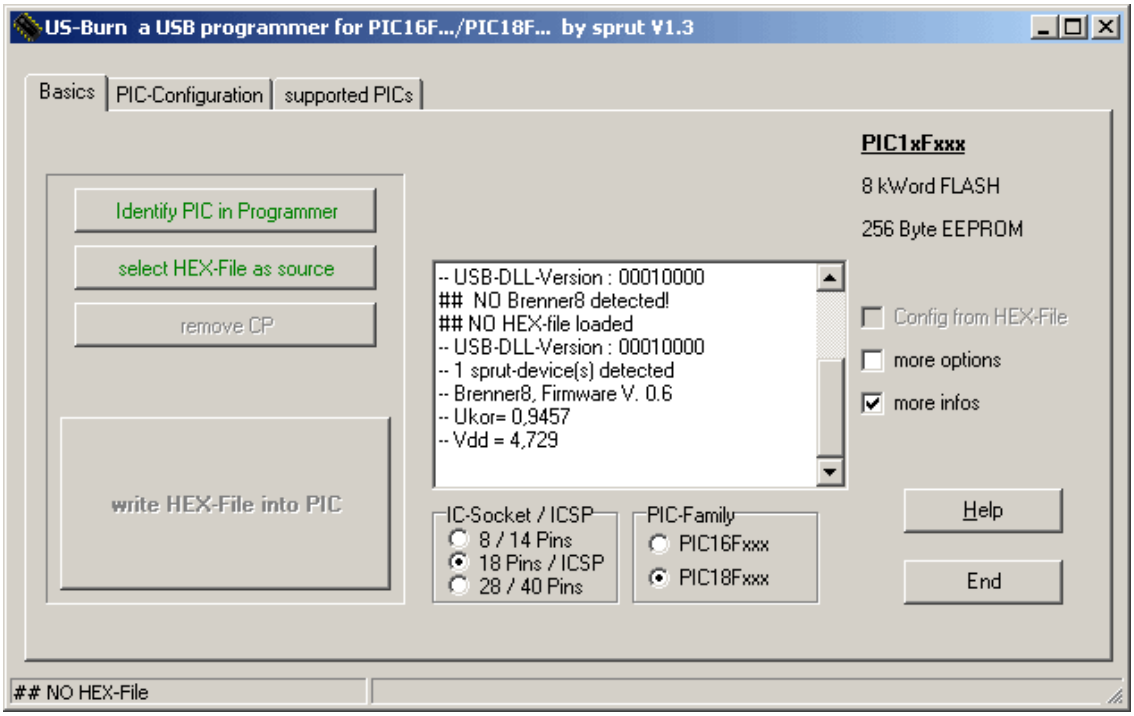


Abbildung 26 vereinfachtes Haupt-Programmfenster

Durch Anklicken des Auswahlfeldes **‚more options'** werden auch alle anderen Funktionen von US-Burn freigeschaltet.

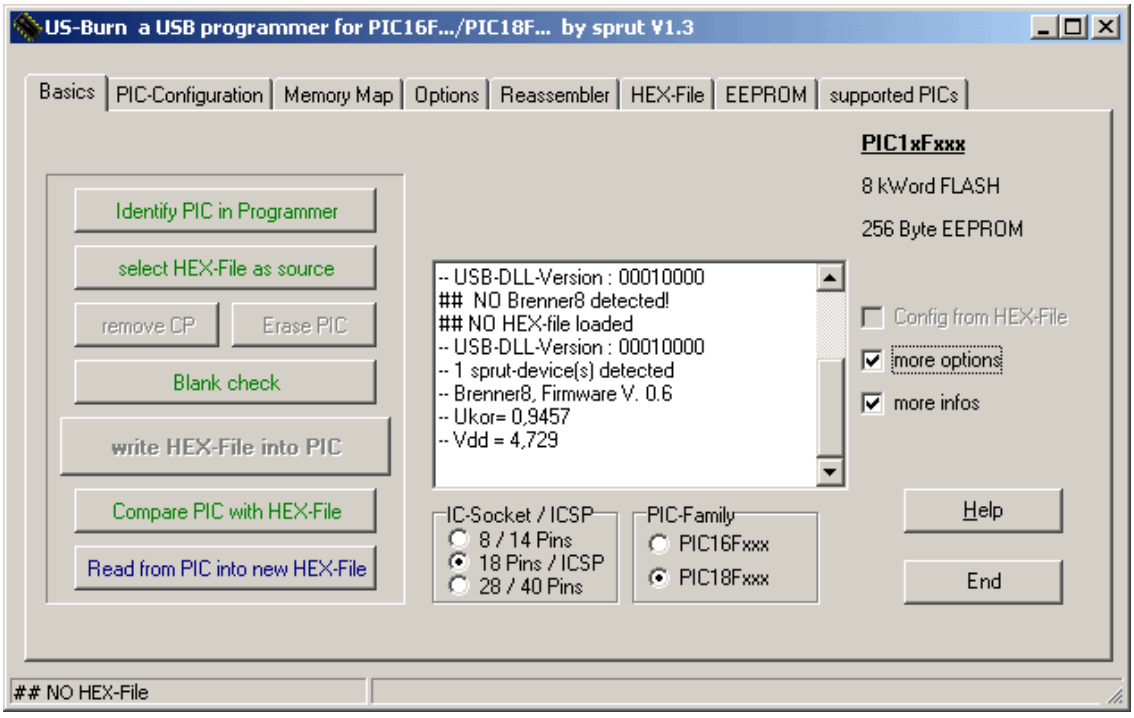


Abbildung 27 vollständiges Haupt-Programmfenster - noch kein PIC erkannt

Auf der **Basics**-Seite gibt es ein Log-Fenster, in dem US-Burn über Abläufe und Ereignisse berichtet.
Alle Zeilen die in diesem Fenster mit ‚##' anfangen stellen Warnungen oder

Fehlermeldungen dar, und sollten unbedingt beachtet werden.
Das Programm ist recht schwatzhaft. Möchte man nur die wirklich wichtigen Informationen bekommen, dann deaktiviert man die Option „**more infos**“

Die Hintergrundfarbe des Log-Fensters wechselt bei Fehlern auf hellrot und bei erfolgreichen Brennversuchen auf hellgrün.

Solange der genaue Typ des PIC-Prozessors noch nicht bekannt ist, verweigert das Programm den schreibenden Zugriff auf den PIC. Deshalb sind einige Schaltflächen noch deaktiviert.

10.5.3 PIC-Typ einstellen

Alle Informationen über Programmierstrategien und Konfigurationen der PICs sind in 6 PIC-Datenfiles (der PIC-Database) abgelegt, die im ZIP-Paket von US-Burn enthalten sind. Findet US-Burn diese Dateien nicht, so gibt es eine Fehlermeldung aus, wie z.B.:
missing PIC-database: picdef03.dat

Solange US-Burn diese Dateien nicht findet, ist es nicht funktionstüchtig.

Spätestens nun sollte man den zu programmierenden PIC in den Sockel des Brenners einsetzen, oder per ICSP-Kabel anschließen. Anschließend **MUSS** im Feld **IC-Socket/ICSP** ausgewählt werden welche Gehäusegröße der PIC hat (wenn er im Sockel sitzt) oder ob er am ICSP-Anschluss angeschlossen ist.
Wenn ein PIC mit einem ICSP-Kabel an den Brenner angeschlossen ist, dann muss immer die Einstellung 18pin/ICSP gewählt werden, egal wie groß der PIC wirklich ist.

Außerdem ist im Feld **PIC-Family** einzustellen ob es sich um einen 14-Bit-Kern PIC (PIC16Fxxx/PIC12F6xx), einen 16-Bit-Kern PIC (PIC18Fxxx) oder einen 12-Bit-Kern-PIC (PIC10Fxxx/PIC1xF5xx) handelt. Für die 14-Bit-PICs der Serien PIC12F6xx lautet die richtige Einstellung PIC16Fxx. Für alle 12-Bit-Kern-PICs (auch z.B. PIC16F54) lautet die richtige Einstellung PIC10Fxx.
Ist hier eine falsche Einstellung gewählt, dann kann der PIC nicht korrekt gebrannt werden. Es ist auch denkbar, wenn auch sehr unwahrscheinlich, dass mit einer falschen Einstellung der PIC beschädigt werden kann.

US-Burn unterstützt etwa 200 unterschiedliche PIC-Typen. Durch Drücken der Taste **Identify PIC in Programmer** ermittelt US-Burn den Typ des PIC, der im Brenner eingesetzt ist, oder via ICSP-Kabel angeschlossen wurde.
Das dient auch der Überprüfung der Funktion des Brenners. Wird der PIC nicht erkannt, dann liegt eine Fehlfunktion vor, und der PIC kann aus Sicherheitsgründen weder gelöscht noch beschrieben werden.

Nur bei der PIC-family „PIC10Fxxx“ (also den 12-Bit-Kern-PICs) funktioniert das nicht, diese müssen von der Anklicken von „Identify PIC in programmer“ in einem Auswahlfeld manuell ausgewählt werden.

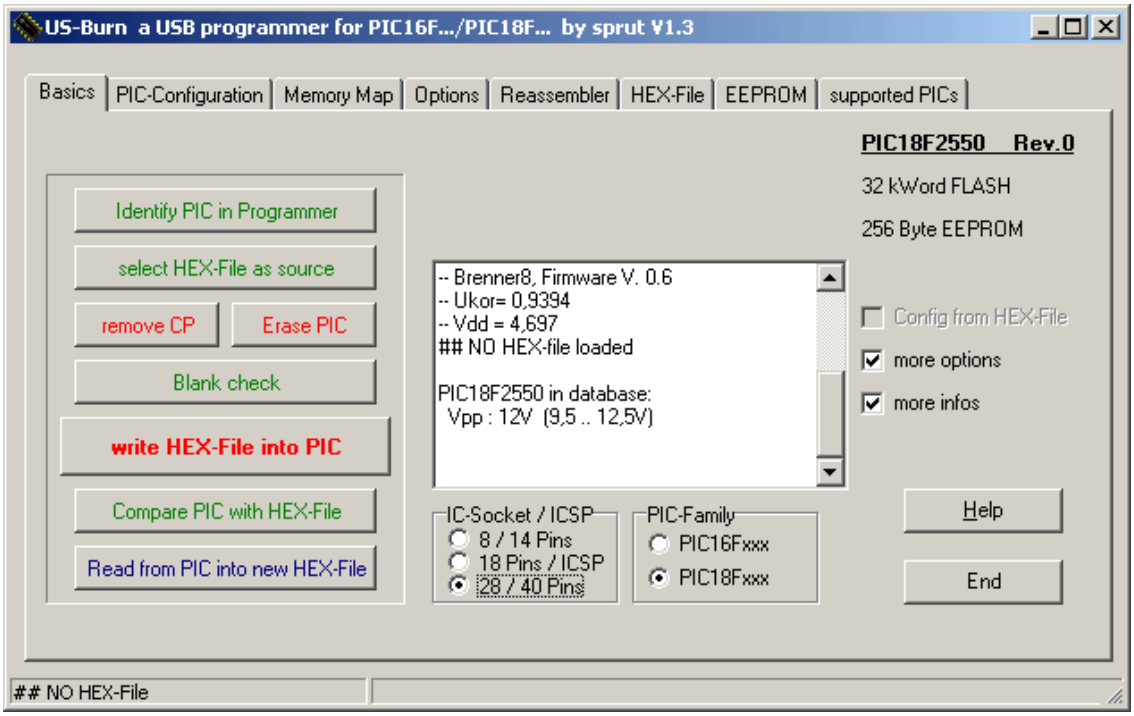


Abbildung 28 vollständiges Haupt-Programmfenster - PIC erkannt

Die **Identify PIC in Programmer** -Funktion erkennt auch, ob der Programmspeicher (FLASH) oder der Datenspeicher (EEPROM) des PIC kopiergeschützt (codeprotected) sind. **Ein kopiergeschützter PIC kann weder gebrannt, noch verglichen, noch sinnvoll ausgelesen werden.** In diesen Fällen müssen die Funktionen **Vergleichen** und **Brennen** fehlschlagen. Die **Lesen**-Funktion liest aus einem Bereich mit Codeprotection nur Nullen aus. Codeprotection lässt sich durch **remove CP** beseitigen. Dabei wird der PIC gleichzeitig gelöscht.

Wird der PIC erkannt, dann wird im Log-Fenster auch die nötige Programmierspannung zum Brennen des PIC angezeigt. Der Brenner wird automatisch so eingestellt, dass er eine korrekte Programmierspannung erzeugt.

10.5.4 Hex-File-laden

Mit dem **Select HEX-File as source**-Button (zweiter von oben) wird ein Dialogfenster geöffnet, mit dem man das in den PIC zu brennende HEX-File auswählt. Dabei öffnet US-Burn das zuletzt verwendeten Directory.

++ACHTUNG++
Oft enthält das HEX-File auch Konfigurationseinstellungen für den PIC. Findet US-Burn solche Informationen im HEX-File, werden sie beim Laden der Datei automatisch übernommen. Sollen diese Konfigurations-Daten nicht verwendet werden, kann die Konfiguration manuell verändert werden, wenn das entsprechenden Häkchen (**Config from HEX-File**) durch anklicken entfernt wird.

USBurn prüft die im HEX-File enthaltenen Konfigurationseinstellungen auf Plausibilität. Werden dabei Fehler festgestellt, erfolgt eine Warnung darüber, in welchem Konfigurationswort der Fehler auftrat. Es wird versucht den Fehler automatisch zu korrigieren und dann die Option **Config from HEX-File** deaktiviert.

Der Anwender sollte in diesem Fall die Konfigurationseinstellung mit Hilfe des **PIC-Configuration**-Fensters manuell überprüfen, bevor er den PIC brennt.

10.5.5 Konfiguration des PIC

Falls die Konfiguration des Brenners nicht schon mit dem Hex-File geladen wurde, oder wenn man die geladene modifizieren muss, kann sie von Hand eingestellt werden.
Dazu darf das Feld **Config from HEX-File** nicht aktiv sein (kein Häkchen). Alle für den jeweiligen PIC-Typ möglichen Konfigurations-Einstellungen stehen dann auf der **PIC-Configuration**-Seite zur Verfügung und können verändert werden. Das konkrete Aussehen des Konfigurationsfensters ist vom erkannten PIC-Typ abhängig. Es kann aus mehreren Unterfenstern bestehen. (Solange kein PIC-Typ erkannt wurde, ist diese Seite leer.)
Grau schattierte Felder können nicht verändert werden.

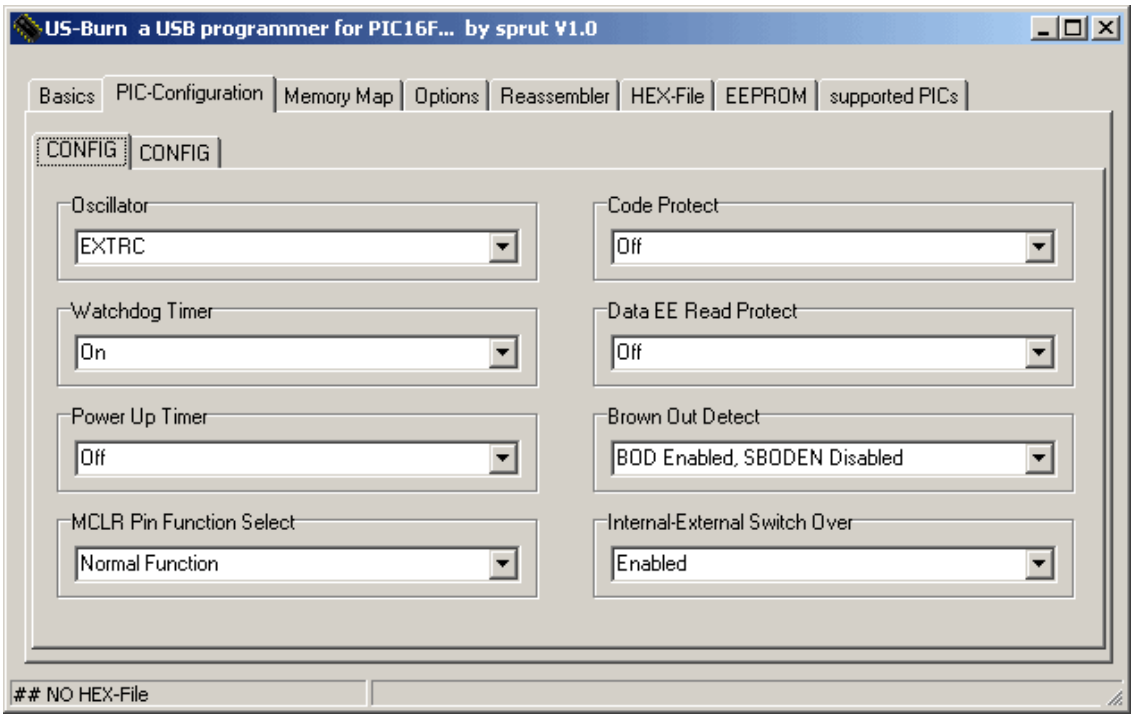


Abbildung 29 Konfigurations-Fenster

Eine Beschreibung vieler der Konfigurationsoptionen findet sich unter www.sprut.de/electronic/pic/config/config.htm.
Die vollständige Beschreibung aller Optionen findet man im Datenblatt des Herstellers.

Das Brennen eines PIC ohne jegliche Konfigurations-Einstellungen führt selten zu einem funktionstüchtigen PIC. Die Default-Einstellungen der PICs wählen meist einen externen RC-Oszillator aus und aktivieren den Watchdog-Timer.

10.5.6 ID-Information des PIC

Im PIC kann (je nach Typ) zur Identifikation eine bis zu 8-stellige ID abgelegt werden. Diese Information muss dann aber im HEX-File enthalten sein.

10.5.7 Brennen des PIC

Das Drücken auf die **Write HEX-File into PIC**-Taste startet den Brennvorgang, dessen Fortschritt mit den Fortschrittsbalken angezeigt wird.
War vorher noch kein HEX-File ausgewählt worden, so verlangt US-Burn jetzt die Auswahl eines Hex-Files. Wurde schon vorher ein HEX-File ausgewählt, so wird dieses nun noch einmal ‚frisch' eingelesen, und dann in den PIC gebrannt.
Nach dem Brennen wird das Ergebnis überprüft, und das Testergebnis im Log-Fenster ausgegeben.
Vor dem Brennen löscht US-Burn den PIC, wenn ihm das nicht verboten wurde (siehe Optionen).

10.5.8 Vergleichen des PIC mit dem HEX-File

Nach dem Drücken auf die **Compare PIC with HEX-File**-Taste wird der Inhalt des PIC mit dem geladenen HEX-File verglichen. Wurden alternative Einstellungen für die Konfiguration gewählt, so werden diese verwendet, und nicht die aus dem HEX-File. (Das gilt nicht für veränderte OSCCAL-Einstellungen)
Nach dem Vergleich wird im Log-Fenster die Anzahl der gefundenen Fehler angezeigt. Das funktioniert nicht bei PICs mit Codeprotection.

10.5.9 Löschen des PIC

Nach dem Drücken auf die **Erase PIC**-Taste wird der Inhalt des PIC gelöscht. Je nach PIC-Typ wird nicht immer alles gelöscht, in jedem Fall aber der Flash-Programmspeicher.
Wenn wirklich alles gelöscht werden soll, erreicht man das durch **remove CP.**
Da US-Burn in jedem Fall die zu programmierenden Speicherbereiche vor dem Brennen noch einmal löscht, ist das für das Brennen der PICs aber nicht nötig.

10.5.10 Blank Check des PIC

Nach dem Drücken auf die **Blank Check**-Taste wird überprüft, ob der PIC ordnungsgemäß gelöscht wurde.
Da US-Burn in jedem Fall die zu programmierenden Speicherbereiche vor dem Brennen noch einmal löscht, ist dieser Test für das Brennen der PICs nicht nötig.

Der Test ist nur nötig, falls man gebrauchte PICs vor der Weitergabe an Dritte gelöscht hat, um den eigenen Code nicht versehentlich mit weiterzugeben.

10.5.11 Auslesen des PIC

Nach dem Drücken auf die **Read from PIC into new HEX-File**-Taste wird der Inhalt des PIC ausgelesen. Dabei wird immer der gesamte Programmspeicher, der EEPROM-Datenbereich die ID und die Konfiguration ausgelesen. (Das Auslesen von PICs, die codeprotected sind führt zu unsinnigen Daten: alles Nullen.)

Die ausgelesenen Werte können danach nur als HEX-File abgespeichert werden, wobei das Programm versucht unbelegte PIC-Speicherbereiche zu erkennen und zu ignorieren.

Will man den Inhalt des PIC in andere PICs kopieren, so muss das gespeicherte HEX-File zunächst wieder manuell in US-Burn geladen werden.

10.5.12 Codeprotection entfernen

Nach dem Drücken auf die **remove CP**-Taste wird der Speicherleseschutz des PIC aufgehoben. **Dabei wird gleichzeitig der gesamte PIC gelöscht**.

Ob ein PIC durch Codeprotection geschützt ist, lässt sich durch die **Identify PIC in Programmer**-Funktion ermitteln.

10.6 Zusätzliches Anzeigefenster

10.6.1 Grafische Speicheranzeige

In US-Burn gibt es eine grafische Anzeige der Speicherauslastung des PIC

Die Darstellung lässt sich umschalten zwischen:

1. Inhalt des HEX-Files
2. Inhalt des PIC
3. im PIC zu brennende Bereiche
4. Abweichungen zwischen PIC und HEX-File

Die Punkte 2 ... 4 zeigen nur nach dem Auslesen des PIC richtige Informationen an.

Die Anzeige erfolgt als bis zu 8 Blöcke. Je nach Typ des verwendeten PICs sind diese Blöcke unterschiedlich groß. Wird der Mauszeiger auf einen Speicherblock geschoben, dann wird neben dem Mauskursor der Adressbereich des Speicherblocks angezeigt.

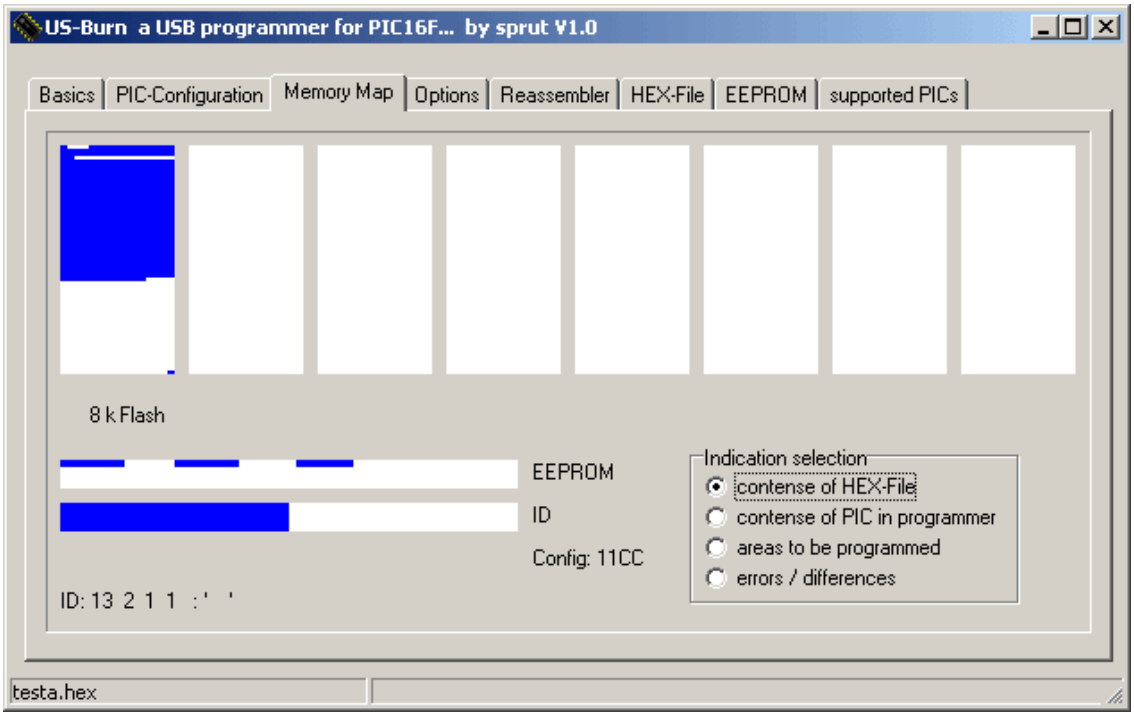


Abbildung 30 grafische Speicheranzeige mit geladenem Programm

Die grafische Speicheranzeige visualisiert den benutzten und den zur Verfügung stehenden Speicher im PIC. Die angezeigten Speicherbereiche entsprechen immer

der FLASH- und EEPROM-Größe des erkannten PICs.
Unbenutzte Speicherbereiche sind weiß dargestellt. Geladener Programmcode wird blau dargestellt.

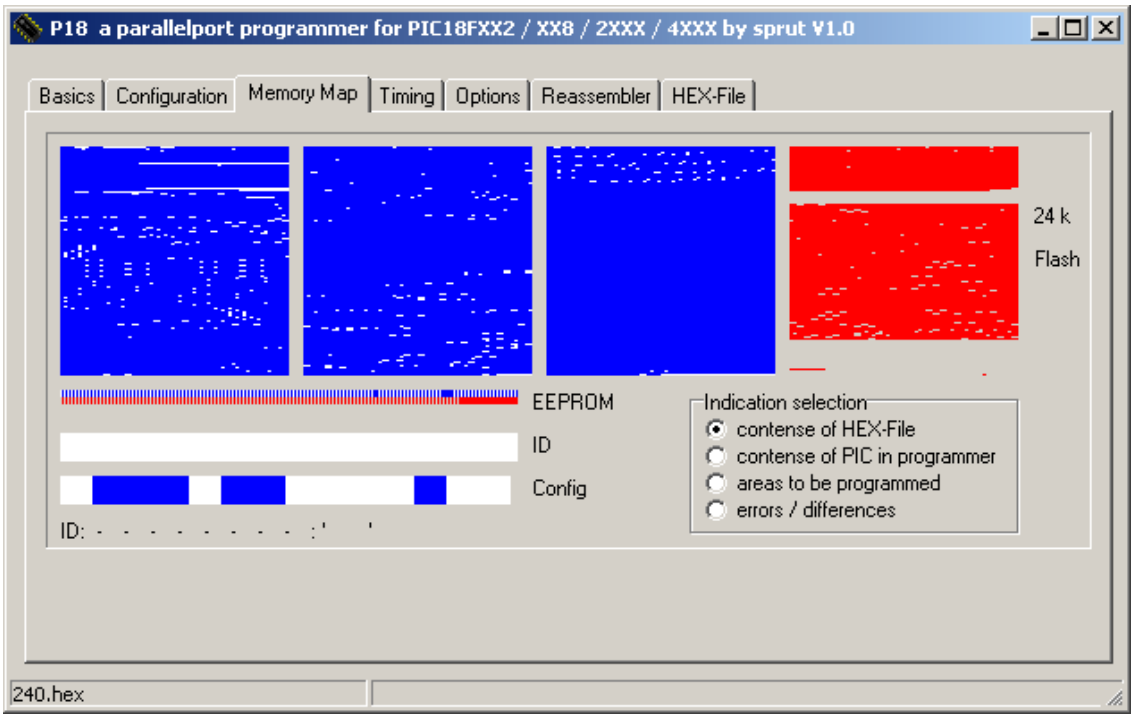


Abbildung 31 grafische Speicheranzeige - Programm und EEPROM-Daten zu groß

Falls Teile des im HEX-File enthaltenen Programmcodes oder der Daten nicht in den FLASH bzw. EEPROM des aktuellen PIC passen, werden die nicht passenden Daten rot dargestellt.

Auch die beim Brennen bzw. Vergleichen gefundenen Abweichungen zwischen HEX-File und PIC-Inhalt (z.B. Brennfehler) werden rot dargestellt.

10.6.2 Reassembler

Sobald im Haupt-Programmfenster ein HEX-File ausgewählt wurde, reassembliert (disassembliert) das Programm den Inhalt des HEX-Files. Das Ergebnis kann man sich im Reassembler-Fenster anschauen.
Findet der Reassembler dabei offensichtliche Code-Fehler (unbekannte Befehle, Sprünge ins Nirgendwo), so wird die Zahl der Fehler auch im Hauptfenster angezeigt.

Es ist für den Reassembler nicht immer einfach Programmcode und Daten von einander zu unterscheiden, insbesondere wenn Sprünge durch Berechnungen mit dem PC-Register durchgeführt werden. Speicherbereiche, die der Reassembler für Daten hält, werden auch reassembliert, allerdings werden hier Code-Fehler ignoriert und als Kommentar ein „DW 0x....“ an jede Zeile angehangen.

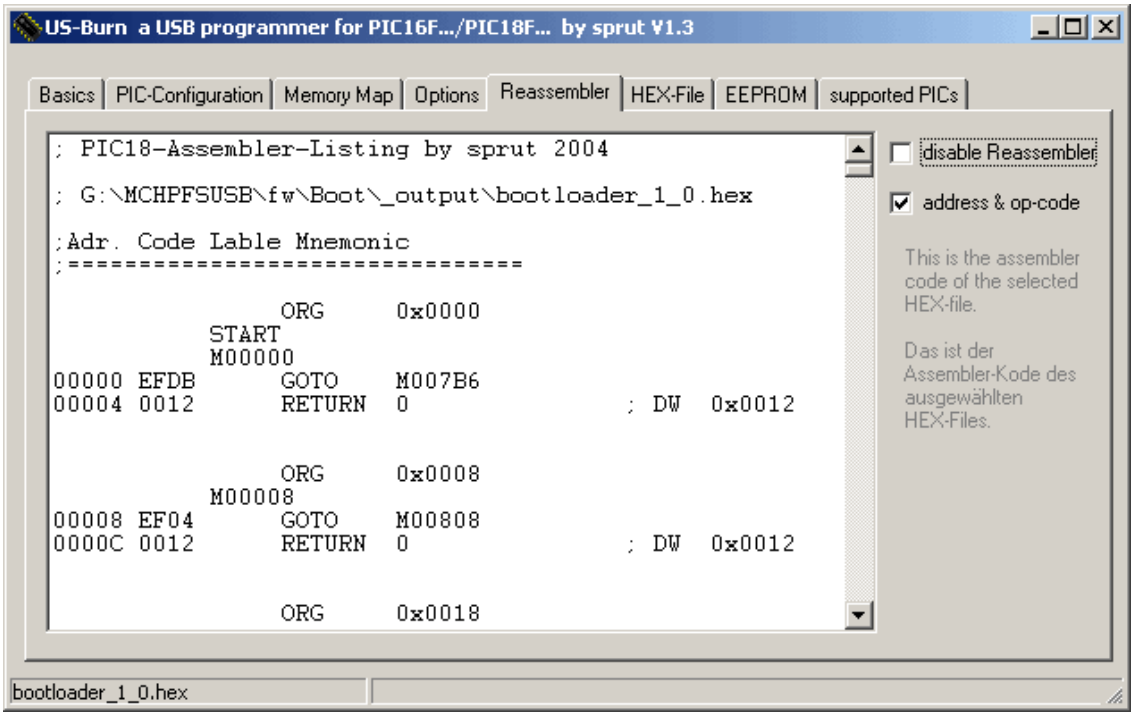


Abbildung 32 Reassembler-Fenster

Der integrierte Reassembler lässt sich abschalten (**disable Reassembler**). Da der Reassembler beim Laden eines HEX-Files wie auch beim Brennen jeweils den gesamten Code reassembliert, kann er auf langsamen PCs zu einer Verzögerung der Arbeit führen.

Der Reassembler kann zu Programmfehlermeldungen führen, falls ‚eigentümlicher' Code benutzt wird. US-Burn funktioniert aber trotzdem.
Da der Reassembler die Arbeit von US-Burn deutlich verlangsamen kann, sollte er solange er nicht benötigt wird abgeschaltet bleiben.

Unter 16-Bit-Windows (Win98/me) ist die Größe des anzeigbaren Textes auf 64kByte begrenzt. Deshalb wird unter diesen Betriebssystemen bei großen Files der reassemblierte Code nicht angezeigt.

10.6.3 HEX-File

Sobald im Haupt-Programmfenster ein HEX-File ausgewählt wurde, kann man sich dieses HEX-File im HEX-File-Fenster anschauen.
Ein Editieren des HEX-Files ist nicht möglich.

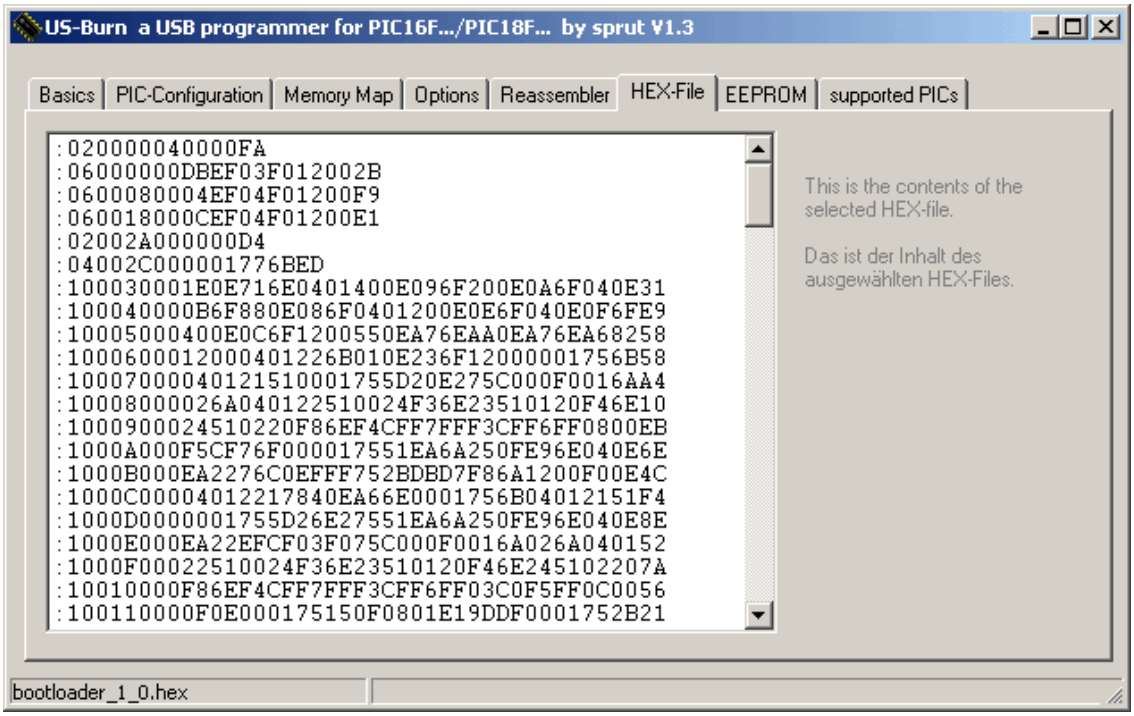


Abbildung 33 HEX-File-Fenster

Unter 16-Bit-Windows (Win98/me) ist die Größe des anzeigbaren Textes auf 64kByte begrenzt. Deshalb werden unter diesen Betriebssystemen extrem große HEX-Files nicht angezeigt.

10.6.4 EEPROM-Fenster

Sobald im Haupt-Programmfenster ein HEX-File ausgewählt wurde, kann man sich die im HEX-File enthaltenen EEPROM-Daten im EEPROM-Fenster anschauen. Es werden nur die EEPROM-Adressen angezeigt, die auch wirklich im HEX-File enthalten sind. Neben den hexadezimalen Zahlenwerten der EEPROM-Daten werden mögliche ASCII-Zeichen auch ausgegeben.
Ein Editieren der EEPROM-Daten ist nicht möglich.

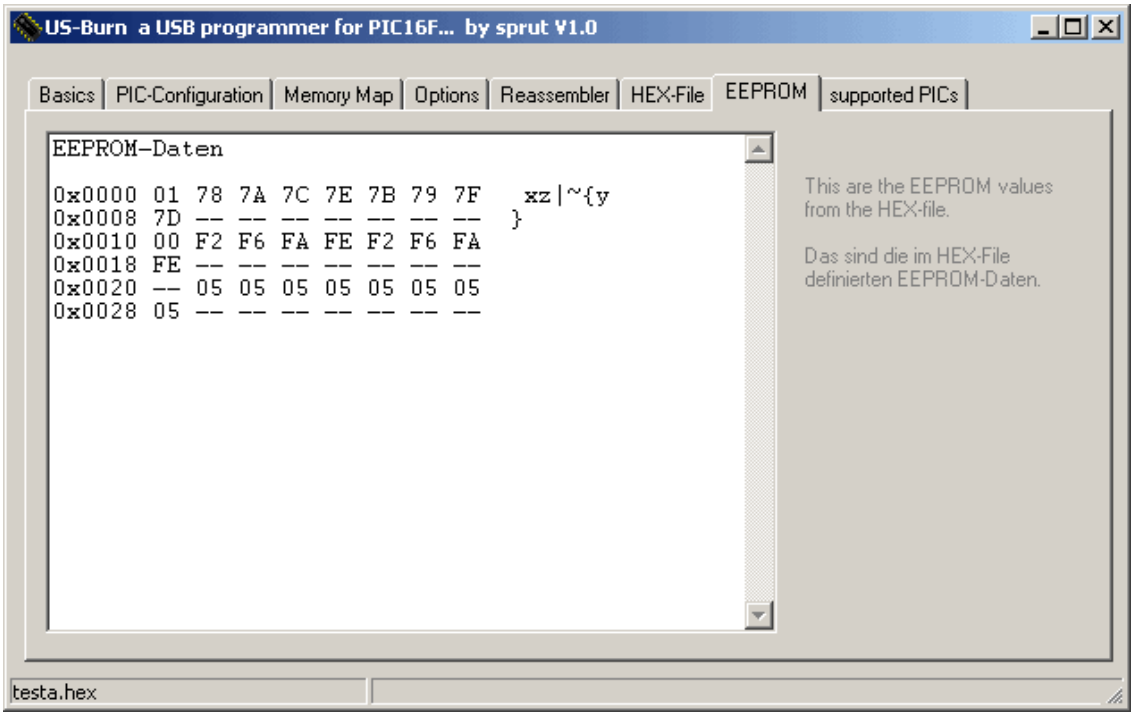


Abbildung 34 EEPROM-Fenster

10.6.5 Optionen

Einige Optionen sind beim Programmstart immer aktiviert. Im Optionen-Fenster kann man sie für an- oder abschalten. Nötig ist das normalerweise nicht.
Die gewählten Einstellungen werden gespeichert.

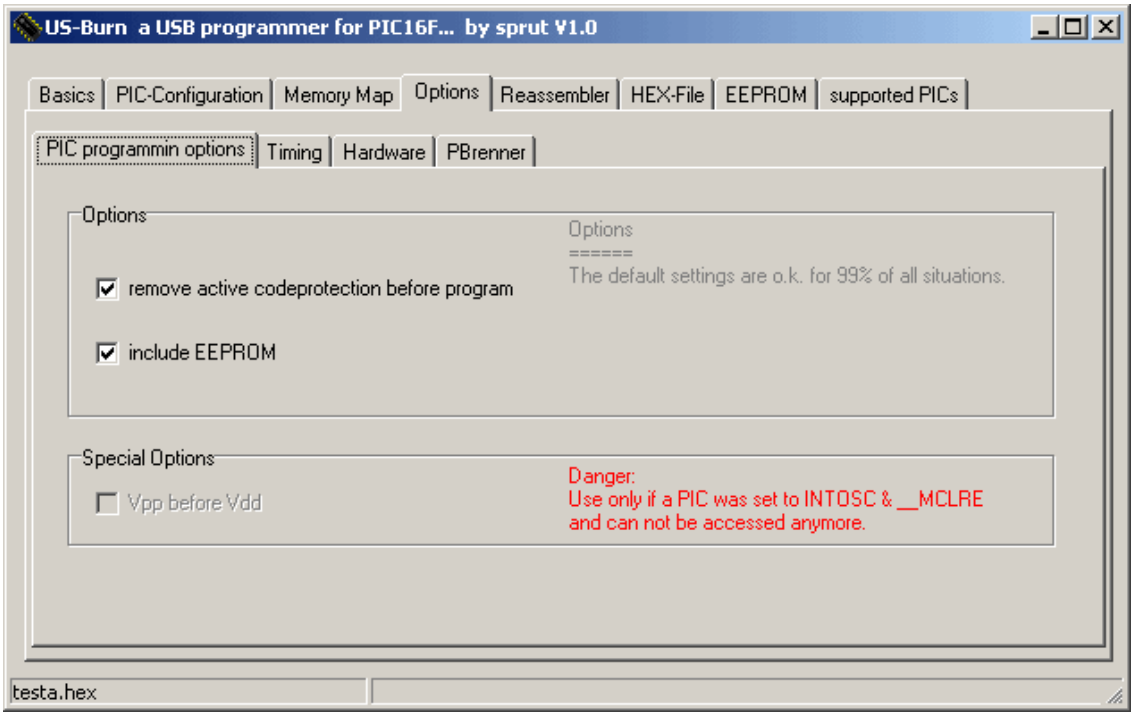


Abbildung 35 Optionen-Fenster

10.6.5.1 remove active codeprotection before program

Ist diese Option aktiviert, dann wird vor einem Brennen geprüft, ob der PIC codeprotected ist. Ist das der Fall, wird Codeprotection deaktiviert, wodurch der gesamte PIC vor dem Brennen gelöscht wird.
In den meisten Fällen ist das sinnvoll. US-Burn prüft aber nicht, ob eventuell nur Bereiche des PIC codeprotected sind, die gar nicht beschrieben werden sollen.

Hat man z.B. einen PIC, in dem sich in einem Bereich Programmcode oder Daten befinden, die geschützt sind, und man will diese Daten erhalten und zusätzlichen Code/Daten in andere Bereiche dazubrennen, dann muss die Option deaktiviert werden, oder US-Burn würde die alten Daten ohne Nachfrage löschen.

10.6.5.2 Vpp before Vdd

Diese Option vertauscht die Einschaltreihenfolge von Betriebs- und Programmierspannung des PIC. Das verstößt gegen die Beschreibung des Herstellers und ist für den PIC deshalb potenziell gefährlich. Ich betrachte das als letzte Notmaßnahme, falls man in die INTOSC-MCLR-Falle getappt ist.
Falls ein PIC mit den Optionen **INTOSC** und ohne **MCLR-Pin enabled** programmiert wurde, dann kann es passieren, dass er sich nicht mehr programmieren lässt. Dann – und nur dann – kann man diese Option aktivieren und dann versuchen den Chip zu löschen. Falls das klappt, sollte man die Option wieder deaktivieren.
Beim Programmstart ist diese Option aus Sicherheitsgründen immer deaktiviert.

10.6.6 Timing

Die Timing Einstellungen sind in der momentanen Programmversion deaktiviert.

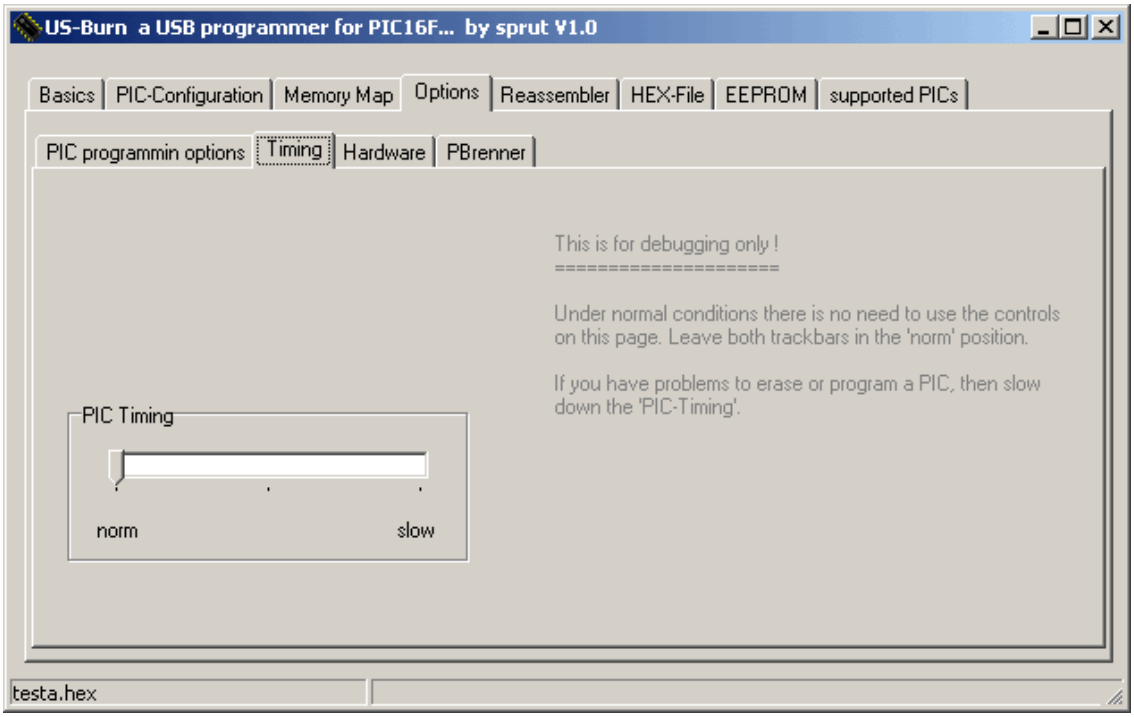


Abbildung 36 Timing Fenster

10.7 Hardware Einstellungen

Im **‚Options'-‚Hardware'**-Fenster kann man die Programmierspannungserzeugung des Brenners kalibrieren und die Funktion des Brenners testen. Zum Test der einzelnen Signalleitungen sollte kein PIC im Brenner eingesetzt sein.

10.8 Kalibrierung der Programmierspannung

Die Kalibrierung der Brenner8-Programmierspannung mit Hilfe von US-Burn ist weiter oben erklärt (Seite 42).

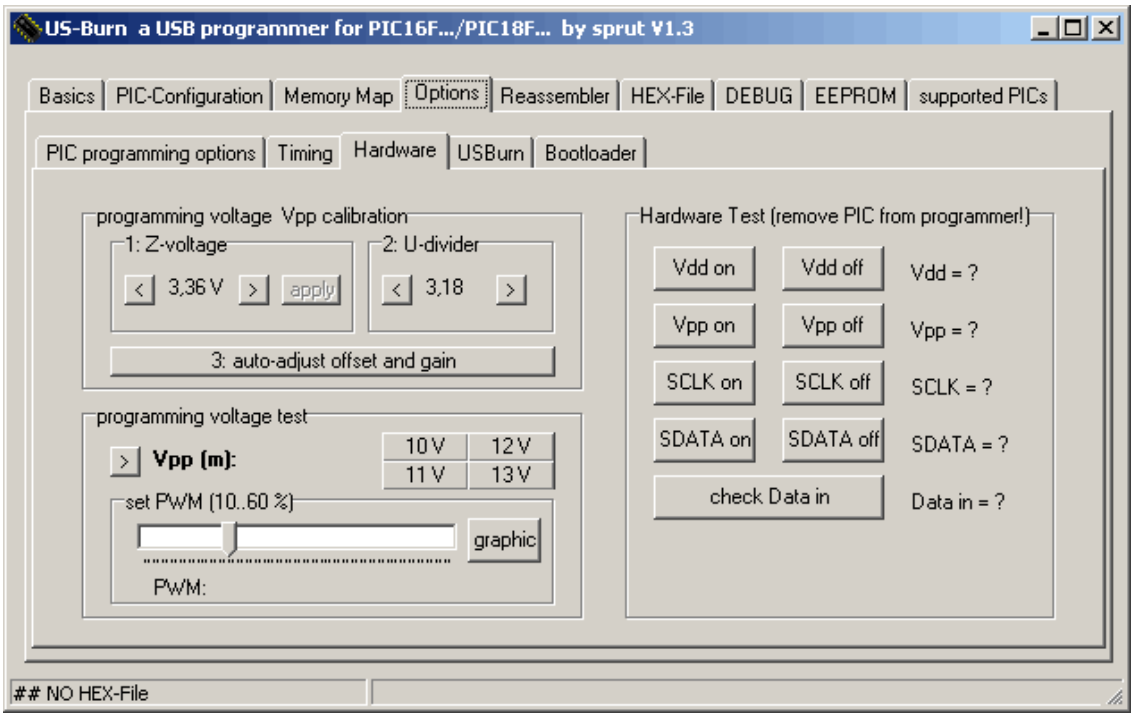


Abbildung 37 Hardware Einstellungen

10.9 Einsatz mehrerer Brenner8/9

Findet USBurn (ab V.1.9a5) beim Start mehrere am PC angeschlossene USB-Geräte vor die Brenner8 oder Brenner9 sein könnten, dann aktiviert die Software immer das USB-Device, das von Windows zuerst erkannt worden ist (USB Instanz 0). Ein zweifaches Aufblinken beider LEDs des ausgewählten Brenners zeigt dem Nutzer an, welches Programmiergerät nun aktiv ist.

Gleichzeitig erscheint im **‚Options'-‚Hardware'**-Fenster das Feld **programmer select**. Hier kann mit Hilfe eines Schiebereglers einfach auf einen anderen angeschlossenen Brenner umgeschaltet werden. Am neu ausgewählten Brenner blinken nun beide LEDs zwei mal hintereinander doppelt auf.

Ein An- oder Abstecken eines Brenners während USBurn läuft wird nicht unterstützt.

10.10 OSCCAL-Editor

Einige PICs haben einen Kalibrierwert für den internen Oszillator. Dieser OSCCAL-Wert wird im Flash-Speicher z.B. auf der Adresse 0x3FF als Teil eines RETLW-Befehls gespeichert.
US-Burn kümmert sich selbständig darum, dass beim Löschen und Neuprogrammieren solcher PICs der OSCCAL-Wert im Flash-Speicher erhalten bleibt. Man kann aber nie ausschließen, dass der Wert beim Programmieren doch einmal verloren geht. Dann kann mit dem OSCCAL-Editor ein neuer OSCCAL-Wert festgelegt werden.

Der OSCCAL ist ein Zahlenwert zwischen 0 und 255. Normalerweise liegt er etwa mittig, also bei 128. Das ‚**old OSCCAL from PIC'**-Fenster zeigt an, welche OSCCAL-Werte im PIC vorgefunden wurden. Dabei steht der richtige Wert normalerweise im Flash, aber einige Programme sind in der Lage einen neuen OSCCAL zu bestimmen, und legen ihn im EEPROM ab. (Wo er aber nur ein Backup ist.) Im unten stehenden Beispielbild ist im Flash-Speicher ein OSCCAL von 136 gefunden worden. Im EEPROM steht nichts.

Im ‚**source for OSCCAL**'-Fenster kann man entscheiden, welcher OSCCAL-Wert verwendet werden soll. Normalerweise ist ‚**keep old OSCCAL**' die richtige Einstellung. Es wird dann der im PIC–Flash vorhandene Wert benutzt. Wenn dieser aber verlorengegangen ist (wie im Beispiel), dann kann man einen OSCCAL aus dem HEX-File benutzen (falls einer im HEX-File steht), oder den OSCCAL an einem Schieberegler manuell einstellen.

Falls ein Backup-Wert des OSCCAL im EEPROM (Adresse 0x7F) stehen sollte, kann man den Schieberegler darauf einstellen lassen. (**get it from EEPROM**)

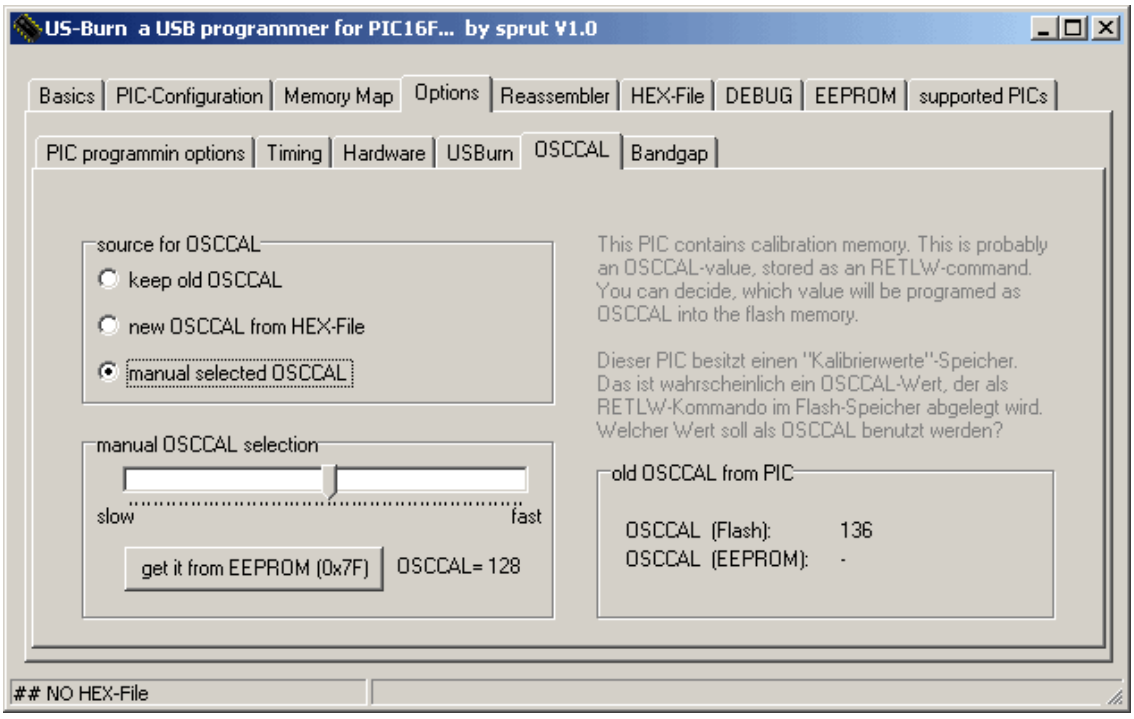


Abbildung 38 OSCCAL-Editor

10.11 Bandgap-Editor

Einige PICs haben einen Kalibrierwert für die internen Bandgap-Spannungsquelle. Dieser Bandgap-Wert wird in der Konfiguration des PICs gespeichert.
US-Burn kümmert sich selbständig darum, dass beim Löschen und Neuprogrammieren solcher PICs der Bandgap-Wert erhalten bleibt. Man kann aber nie ausschließen, dass der Wert beim Programmieren doch einmal verloren geht. Dann kann mit dem Bandgap-Editor ein neuer Bandgap-Wert festgelegt werden.

Beim Betätigen '**identify PIC in Programmer'** wird der aktuelle Bandgap-Wert aus dem PIC ausgelesen, und der Schieberegler auf diesen Wert eingestellt. Um beim nächsten Programmieren (**write HEX-file into PIC**) einen anderen Wert in den PIC zu schreiben, braucht man nur den Schieberegler zu verstellen.

Zwischen dem Verstellen des Schiebereglers und dem Programmieren des PIC darf natürlich nicht noch einmal '**identify PIC in Programmer'** betätigt werden.

Wird nach dem Programmieren noch einmal **Compare PIC with HEX-File** angeklickt, dann wird natürlich ein Fehler in der Konfiguration gemeldet, da der veränderte BG-Wert in der Konfiguration des PIC ja nicht mit dem Wert des HEX-Files übereinstimmt.

Es gibt Berichte, nach denen einige PICs bei eingestelltem BG-Wert von 3 nicht mehr laufen. Nach einer Verringerung des BG-Wertes funktionieren sie wieder. Ich konnte das aber nicht nachvollziehen.

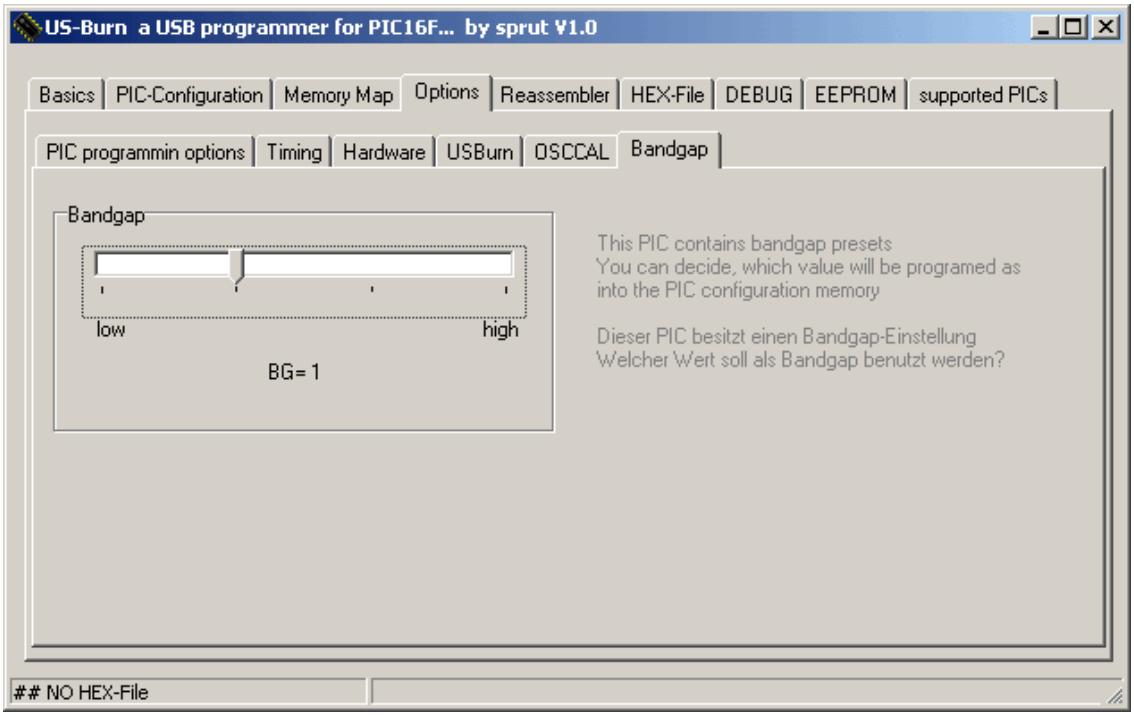


Abbildung 39 Bandgap-Editor

10.12 Reanimation

Durch eine Eigenheit einiger PIC18Fxxxx-Typen ist es möglich, dass ein PIC durch

Brennen mit einer fehlerhaften Konfiguration in einen Mode geschaltet wird, in dem er für den Brenner nicht mehr ansprechbar ist. („ICD/ICSP Port Enable bit“-Falle) Der Brenner meldet dann beim Klick auf **Identify PIC in Programmer**, dass ein PIC mit dieser ID nicht in der Database enthalten ist. Sollte dieser Effekt bei einem PIC auftreten, der anfangs einwandfrei funktioniert hat (und nur in diesem Fall !!!) kann der PIC12/16/18 mit der Reanimationsoption wieder in den Normalzustand versetzt werden. Dabei wird der PIC gelöscht. (Reanimate steht nicht für alle PIC-Typen zur Verfügung.)

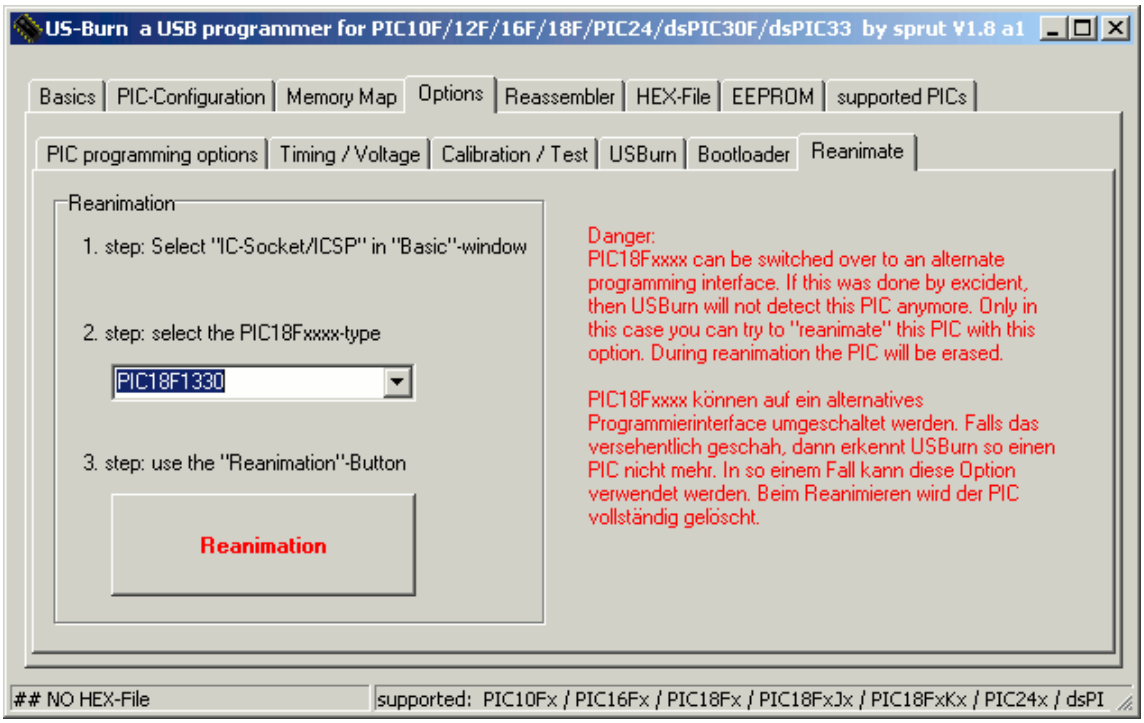


Abbildung 40 Reanimations-Fenster

Zur Reanimation muß zunächst im **Basic** Fenster die richtige Option für **IC-Socket/ICSP** ausgewählt werden, was in der Regel ohnehin schon erfolgte. Danach ist im **Option** – **Reanimation** Fenster der richtige PIC-Typ manuell auszuwählen. Daraufhin kann die Schaltfläche **Reanimation** angeklickt werden.
USBurn versucht nun den PIC zu reanimieren, schaltet zum **Basic**-fenster und startet automatisch **Identify PIC in Programmer**. Hat alles geklappt, dann wurde der PIC nun wieder erkannt. Wenn nicht, dann hat das Problem der Nichterkennung eine andere Ursache.

USBurn verhindert ab der Version V1.8 das ein PIC mit so einer fehlerhaften Konfiguration programmiert wird, so das dieses Problem hoffentlich in der Zukunft an Bedeutung verliert.

10.13 Kommandozeilenparameter

Auf vielfachen Wunsch habe ich mit der Version 1.9 wieder eine Steuerung des Programms mit Kommandozeilenparametern eingeführt. Diese Steuerung umfasst aber nur grundlegende Funktionen. Die 12-Bit-Kern-PICs werden noch nicht unterstützt.

Da US-Burn beim Programmstart die Kommandozeilenparameter nacheinander abarbeitet, ist die richtige Reihenfolge einzuhalten.

Alle Optionen werden mit „/“ eingeleitet. Innerhalb einer Option sind keine Leerzeichen erlaubt. Zwischen Optionen hat mindestens ein Leerzeichen zu stehen. Gross- und Kleinschreibung ist zu beachten. Hat eine Option einen Parameter, dann folgt der ohne Leerseichen. Ein typischer Programmaufruf könnte so aussehen:

usb19.exe /S18 /F18 /P /Ic:\testfiles\test.hex /p /w /x

Die Optionen auf der Kommandozeile lassen sich in 4 aufeinanderfolgende Gruppen einteilen:

- Mit /S, /F und /P wird der PIC-Typ und seine Verbindung zum Brenner bestimmt.
- Mit /I wird der Name des Hex-Files festgelegt (nur nötig falls /w oder /c folgen).
- Mit /p, /e, /b, /w, /c wird die eigentliche Aktion bestimmt.
- Mit /x wird das Programm beendet.

Optionen mit Parameter

/S...	(**S**OCKET)	Pin-Anzahl (nur Brenner8) oder ICSP- Anschluss
/F...	(**F**AMILY)	PIC-Familie
/I...	(**I**N)	Name des HEX-File (großes I) (für write & compare)

Optionen ohne Parameter

/P	(**P**IC)	erkenne den PIC-Typ (großes P)
/p	(remove c**p**)	entferne codeprotection vom PIC (kleines p)
/e	(**e**rase)	lösche PIC
/b	(**b**lank)	prüfe ob der PIC leer ist
/w	(**w**rite)	brenne HEX-file in den PIC
/c	(**c**ompare)	vergleiche PIC-Inhalt mit dem HEX-file
/x	(ende)	beendet das Programm und erzeugt das Log-File

/S (SOCKET) Pin-Anzahl (nur Brenner8) oder ICSP- Verbindung

Damit der Brenner den PIC **im Testsockel** des Brenner8 korrekt ansteuern kann, muss er wissen, wieviele Pins er hat oder ob er am ICSP-Verbinder angeschlossen ist. Das erfährt er durch den Parameter der **/S**-Option. Mögliche Parameter für PICs im Testsockel sind **8**, **14**, **18**, **28** und **40**. Das sind die Anzahl der Pins des PICs im DIL-Gehäuse. Ist der PIC aber am ICSP-Verbinder angeschlossen (bei Brenner8mini und Brenner9 immer der Fall), ist stets der Parameter **ICSP** zu wählen. Ich erinnere daran, dass nur PICs mit 14-Bit-Kern (PIC16F...und einige PIC12F...) und 16-Bit-Kern (PIC18F...) im Testsockel gebrannt werden können. Alle anderen benötigen den ICSP-Anschluss.
Wird anschließend mit der **/F**-Option eine PIC-Familie ausgewählt, die sich ausschließlich über den ICSP-Verbinder brennen läßt (PIC18FxxJxx, PIC24,

dsPIC30, dsPIC33), dann wird der Brenner auf ICSP eingestellt, auch wenn eine andere **/S**-Option verwendet wird.
Beispiel:

```
usb19.exe /SICSP .........
usb19.exe /S28 ..........
```

/F (FAMILY) PIC-Familie
Es gibt eine Reihe grundsätzlich unterschiedlicher PIC-Prozessorfamilien. Der Brenner benötigt die Information, zu welcher Famile der Target-PIC gehört. Diese Einstellung erfolgt mit dem Parameter der **/F**-Option. Es gibt folgende Parameter: **10**, **16**, **18**, **18J**, **24**, **30**, **33**

Parameter	Beschreibung	Typen
10	alls 12-Bit-Kern-PICs	alle PIC10Fxxx alle PIC1xF5x alle PIC1xF5xx
16	alls 14-Bit-Kern-PICs	alle PIC16Fyx außer y=5 alle PIC16Fxxx alle PIC12Fyxx außer y=5
18	16-Bit-Kern-PICs	alle PIC18Fxxx alle PIX18Fxxxx
18J	16-Bit-Kern-PICs	alle PIC18FxxJxx
24	alle PIC24-Typen	alle PIC24....
30	alle dsPIC30F-Typen	alle dsPIC30F....
33	alle dsPIC33F-Typen	alle dsPIC33F....

Beispiel:

```
usb19.exe /SICSP /F30 .........
usb19.exe /S28 /F16 ..........
```

/P (PIC-Typ) PIC-Typ bestimmen
Nachdem Sockel (/S) und Familie (/F) festgelegt wurden, dann wird mit dieser Option die automatische Typbestimmung gestartet. Erst nach korrekter Typerkennung können weitergehende Funktionen (löschen , brennen ...) genutzt werden.
Aus diesem Grunde **MUSS** die /P-Option unbedingt nach /S und /F folgen.
Beispiel:

```
usb19.exe /SICSP /F30 /P .........
usb19.exe /S28 /F16 /P ..........
```

/I (IN) Name des HEX-Files (für brennen und vergleichen)
USBurn kann ein Input-HEX-File einlesen (für /w und /c).
Diese Option legt einen Filenamen fest, der ausschließlich für das Input-File (zum Brennen mit /w und Vergleichen mit /c) verwendet wird. Befindet sich das HEX-File nicht im gleichen Verzeichnis wie USBurn, dann ist der Hex-File-Name mit dem gesamten Dateipfad anzugeben.
Beispiel:

```
usb19.exe /SICSP /F30 /P /Ic:test.hex .........
usb19.exe /S28 /F16 /P /Ic:test.hex..........
```

/p (remove cp) entferne codeprotection vom PIC
Eine aktivierte Codeprotection des PICs wird deaktiviert und dabei der Flash-Programmspeicher und der EEPROM-Datenspeicher gelöscht. Das Löschen der Konfiguration und der User-ID ist nicht bei allen PIC-Typen gewärleistet. Ein vollständiges Löschen ist aber durch die Kombination von /p und /e garantiert.
Beispiel:

```
usb19.exe /SICSP /F30 /P /Ic:test.hex /p.........
usb19.exe /S28 /F16 /P /Ic:test.hex /p..........
```

/e (erase) lösche PIC
Der PIC wird gelöscht. Bei einigen PICs funktioniert diese Option nicht, solange Codeprotection aktiviert ist. In diesem Fall sollte vor /e zusätzlich /p verwendet werden. Vor dem Brennen ist diese Option nicht nötig, da vor dem Brennen der PIC ohnehin automatisch gelöscht wird
Beispiel:

```
usb19.exe /SICSP /F30 /P /Ic:test.hex /e.........
usb19.exe /S28 /F16 /P /Ic:test.hex /p /e..........
```

/w (write) brennen des HEX-Files in den PIC
Der Inhalt einen HEX-Files (welches mit /I vorher festgelegt wurde) wird in den PIC gebrannt.
Die Option /w bewirkt automatisch ein Löschen des PICs und ein Entfernen von Codeprotection vor dem Brennen sowie ein Test des gebrannten PICs nach dem Brennen. Das Angeben der Optionen /p, /e und /c ist also nicht nötig.
Beispiel:

```
usb19.exe /SICSP /F30 /P /Ic:test.hex /w .........
usb19.exe /S28 /F16 /P /Ic:test.hex /w ..........
```

/c (compare) vergleiche den PIC mit dem HEX-File
Der Inhalt eines PIC wird mit dem Inhalt einen HEX-Files Files (welches mit /I vorher festgelegt wurde) verglichen, und Fehler angezeigt.
Beispiel:

```
usb19.exe /SICSP /F30 /P /Ic:test.hex /c .........
usb19.exe /S28 /F16 /P /Ic:test.hex /c ..........
```

/x (ende) beendet das Programm und erzeugt das Log-File
Diese Option sorgt dafür, das USBurn beendet wird. Damit alle Ausgaben des Log-Fensters danach noch zur Verfügung stehen, wird ein Log-File (**usburnlog.txt**) mit dem Inhalt des Log-Fensters erzeugt.
Um eine automatische Auswertung des Log-Files zu vereinfachen, beginnt die usburn.log-Datei mit einer Zeile in der entweder „**OK**“ oder „**FAIL**“ steht. Falls dort „FAIL“ stehen sollte, dann hat eine der Funktionen des Brenners (Erkennen des PIC-Types, Brennen, Vergleichen) zu einer Fehlermeldung geführt. Was genau passiert ist, muss man dann den Details des Log-Files entnehmen.
Wird USBurn ohne die abschließende Option /x aufgerufen, dann bleibt das

Programm aktiv, und kann ganz normal bedient werden. Ein Log-File wird dann nicht erstellt, da der Nutzer ja das Log-Fenster von USBurn einsehen kann.
Beispiel:

```
usb19.exe /SICSP /F30 /P /Ic:test.hex /w /x
usb19.exe /S28 /F16 /P /Ic:test.hex /w /x
```

10.14 Mögliche Probleme und Lösungen

10.14.1 unbekanntes Device

Symptom:
Windows erkennt nach dem anstecken des Brenners an den PC ein Device, kann es aber nicht identifizieren. Folglich kann auch kein passender Treiber installiert werden.

Ursache:
Möglicherweise stimmt der Takt des Steuer-PIC nicht. Typ des Quarzes und/oder Config-Einstellung des Steuer-PIC prüfen.
Wurde ein eigenes Platinenlayout entwickelt, dann können auch die beiden USB-Datenleitungen (D+ und D-) im Layout vertauscht worden sein.

10.14.2 Unzuverlässige Funktion

Symptom 1:
Der Brenner wird beim ersten Anstecken an den PC erkannt und der Treiber installiert. Wird der Brenner später wieder angesteckt, wird er nicht erkannt.

Symptom 2:
Der Brenner wird erkannt. Bei größeren Aktionen (Lesen, Brennen) stürzt die Brennsoftware ab. Wird die Brennsoftware gleich wieder gestartet, kommt ein Fehler 997.

Ursache:
Der 220nF-Kondensator am Vusb-Pin des steuer-PIC ist nicht angeschlossen. Kondensatoranschluss prüfen.

Symptom 3:
Die grüne LED leuchtet auf. Sodann läßt sich kein PIC mehr erkennen oder brennen. Wird der Brenner vom USB-Kabel getrennt und dann wieder angesteckt funktioniert er wieder, bis nach einiger Zeit wieder die grüne LED aufleuchtet.

Ursache:
Der Vpp-Regler erzeugt manchmal eine (scheinbare) Überspannung die eine Sicherheitsschaltung aktiviert. Der Brenner muß neu kalibriert werden.

Symptom 4:
Der Brenner8 wird erkannt, USBurn erkennt den am Brenner .angeschlossenen PIC-Typ korrekt. Aber beim Start des Brennens kommt ein Fehler 997.

Ursache:

Der Vpp-Regler des Brenner8 schafft es nicht, die für diesen PIC-Typ vorgeschriebene Programmierapnnung innerhalb einer angemessenen Zeit (ca. 1 Sekunde) genau genug einzustellen. Der Brenner muß neu kalibriert werden, in der Regel genügt es, den Schritt 3 der Kalibrierung zu wiederholen.

10.14.3 Einzelne Signale fehlen

Symptom 1:
Das Takt- oder Datensignal oder Vdd des ICSP-Protokolls fehlt.

Ursache:
Der Steuer-PIC wurde nicht fest in seine IC-Fassung gedrückt. (Kommt immer wieder vor.)

10.14.4 Falsche Z-Spannung

Symptom 1:
Die Z-Spannung beträgt nur 0.7V

Ursache:
Die Z-Diode wurde falsch herum eingelötet. Z-Diode umdrehen.

Symptom 2:
Die Z-Spannung liegt deutlich unterhalb der erwarteten Z-Dioden-Spannung.

Ursache:
Der Vorwiderstand der Z-Diode hat eine falschen (viel zu hohen) Wert. Den Widerstand prüfen.

10.14.5 Unzureichende Programmierspannung

Symptom:
Die Programmierspannung Vpp ist zu niedrig. Ihr Maximum erreicht kaum 12V. Eine Kalibrierung ist somit nicht möglich.

Ursache:
Die Diode D1 wurde eine normale Si-Diode benutzt. D1 ist gegen eine Shottky-Diode (BAT43) zu tauschen.

10.14.6 USB Error SE: 100

Wärend des Brennen erscheint ein kleines USB-Fehlerfenster, und im Log-Fenster steht „**USB Error SE: 100**“. In diesem Fall hat der Brenner nicht innerhalb einer vorgegebenen Zeit auf einen Befehl reagiert. Es kann sich um eine zusammengebrochene USB-Verbindung auf Grund eines Hardwareproblems handeln, oft liegt die Ursache aber in der Kalibrierung.
Insbesondere dann, wenn diese Meldung zu Beginn des Schreibens in den PIC oder zu Beginn des Lesens aus dem PIC auftritt, dann ist die Ursache eine zu langsame Regelung der Vpp-Programmierspannung. In diesem Fall ist der Schritt 3 der Kalibrierung zu wiederholen. Danach sollte das Problem verschwunden sein.

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

Nachzulesen unter www.sprut.de/electronic/pic/config/config.htm.
In den (englischsprachigen) Datenblättern des Herstellers sind alle Einstellungen genau erklärt.

Fehlermeldung „file to large or corrupt“ beim Einlesen eines HEX-Files oder beim Brennen
Dieser Fehler wird auf 16-Bit-Betriebssystenen (Win98/me) vom Reassembler verursacht, falls mit großen HEX-Files gearbeitet wird. Das Brennen funktioniert aber trotzdem. Um die Meldung zu unterdrücken sollte man den Reassembler abschalten

10.15 Geschwindigkeit des Brenner8

Je nach PIC-Typ variiert die Geschwindigkeit des Brennens etwas. Generell lassen sich neuere Typen schneller brennen als ältere.

Die folgende Tabelle listet exemplarisch einige PIC-Typen auf, und gibt an, wie lange es dauert den PIC vollständig (FLASH + EEPROM + ID + Config) zu brennen, und anschließend die gebrannten Daten auf Korrektheit zu prüfen:

Tabelle 6 Programmierzeiten

Typ	Flash-Zellen	EEPROM-Bytes	Dauer in Sekunden
PIC16F876	8 k	256	45
PIC16F84	1 k	64	13
PIC16F628A	2 k	128	10
PIC16F916	8 k	256	15
PIC12F629	1 k	128	6
PIC16F874A	4 k	128	6
PIC18F242	16 k	256	17
PIC18F2455	24 k	256	23
PIC18F4431	16 k	256	15
PIC18F1320	8 k	256	6

Anmerkung:
Der Brenner8/9 läuft mit „angezogener Handbremse“. Es wäre problemlos möglich, seine Geschwindigkeit zu verdreifachen. Ich befürchte für diesen Fall aber Probleme mit der Signalqualität auf „suboptimal“ aufgebauten Brennern. Vielleicht werde ich das zukünftig änden, aber der Bastler sollte die paar Sekunden Zeit haben.

10.16 US-Burn deinstallieren

Um US-Burn vollständig vom Rechner zu entfernen genügt es, folgende Dateien manuell zu löschen:

- xxx\usburn*.exe
- xxx\mpusbapi.dll
- xxx\mpusbapi.dll
- xxx\picdef3.dll
- xxx*03.dat (insgesamt 6 Dateien der PIC-Database)

- xxx\usbrn.hlp
- xxx\usburn.ini

sowie den USB-Treiber zu deinstallieren.

Dabei steht ‚xxx\' für das Installationsverzeichnis von US-Burn.

10.17 Bekannte Probleme

1)
Die PIC-Typerkennung versagt beim PIC16F83.
Zwischen PIC16F636 und 16F639 wird nicht unterschieden. (kein Problem)

11 usburn for Linux

usburn ist ein Linuxprogramm, dass das HEX-File analysiert, und die darin enthaltenen Daten via USB-Anschluß an den Brenner8 überträgt. Darüber hinaus bietet US-Burn noch einige Extras:

- Kalibrierung des Brenner8 (in Vorbereitung)
- Laden neuer Firmware in den PIC (mit Bootloader)

11.1 Voraussetzungen für die Nutzung von usburn

11.1.1 Software

US-Burn ist unter Linux mit libusb lauffähig. Ich teste die Software aber nur unter Debian 4.0.
Der Betrieb unter allen aktuellen Linuxversionen sollte möglich sein.

11.1.2 Daten

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

11.1.3 Hardware

Das Programm US-Burn benötigt als Hardware den USB-Port-Brenner „Brenner8“ oder „Brenner9“ .

- Brenner8
- Brenner8mini
- Brenner9

Brenner8: Mein momentaner Standard für den USB-Port
http://www.sprut.de/electronic/pic/projekte/brenner8/index.htm

Brenner8mini: Ein vereinfachter Brenner8 der nur über ICSP brennt
http://www.sprut.de/electronic/pic/projekte/brenner8mini/index.htm

Brenner9: Ein Brenner für 3,3V-PICs der nur über ICSP brennt
http://www.sprut.de/electronic/pic/projekte/brenner9/index.htm

11.2 Installation

Usburn erfordert libusb (http://libusb.wiki.sourceforge.net/), was man am besten durch den Paketmanager der Linuxdistribution installieren lässt.

Der Quellcode des Programm usburn steht als TAR-Archiv zum Download bereit. Dieses Archiv mit dem Namen usburnxx.ZIP enthält die folgenden Dateien.

- Quellcodedateien — Das Hauptprogramm in der aktuellen Version
- readme.txt — Kurzanleitung
- picdef03.dat — PIC-Datenfile aus der PIC-Database

- make make-File zum compilieren
- b8_handbuch.pdf dieses Handbuch

Bevor das Programm US-Burn benutzt werden kann, ist das TAR-File zu entpacken, ("**`tar xfv [ARCHIVNAME].tar`**" oder "**`tar xfvj [ARCHIVNAME].tar.gz2`**") und durch Aufrufe von make die ausführbare datei „usburn“ zu erzeugen. Je nach Distribution kann es erforderlich sein, die Pfade in make anzupassen.

Da der Zugriff auf den Brenner8&9 nicht über Kernelmodule, sondern direkt über libusb erfolgt, sind eigentlich **root**-Rechte erforderlich. Damit alle User Zugriff bekommen legt man eine Datei "**/etc/udev/rules.d/99-sprutbrenner**" an und schreibt in diese Datei:

```
SUBSYSTEM=="usb", ATTRS{manufacturer}=="sprut*",
ATTRS{product}=="PIC-Brenner*", GROUP="plugdev"
```

Damit sollte der Brenner ab dem nächsten Einstecken für alle User der plugdev-Gruppe (in der man ja grundsätzlich sein muss, wenn man usb-Geräte benutzen möchte) verfügbar sein. Getested unter Gentoo und Debian 4.0. (Danke Marcel)

11.3 Anwendung

Usburn ist ein Kommandozeilenprogramm, das durch Kommandozeilenoptionen (mit Parametern) gesteuert wird.

Usburn kennt Lang-Optionen und Kurz-Optionen mit nachfolgendem Parameter und ohne Parameter.

Lang-Optionen werden nurch „--„ (zwei Minusse) eingeleitet und bestehen aus einem Wort. Erfordert die Option einen Parameter, so ist dieser durch ein Leerzeichen von der Option abzutrennen. („**--FAMILY 18**“)

Kurz-Optionen werden durch „-„ (ein Minus) eingeleitet und bestehen aus jeweils nur einem Buchstaben. Erfordert die Kurz-Option einen Parameter, so folgt dieser dem Options-Buchstaben ohne trennendes Leerzeichen. („**-F18**“)
Mehrere parameterlose Kurz-Optionen können zusammen hinter einem „-„ stehen.

Usburn unterscheidet bei den Optionen zwischen Klein- und Großschreibung.

11.4 Optionen

Optionen ohne Parameter

-h	--help	schow the help-screen
-r	--read	read PIC-content into HEX-file
-w	--write	write HEX-file into PIC
-c	--compare	compare PIC-content with HEX-file
-e	--erase	erase PIC
-p	--remove	remove codeprotection from PIC

-i	--info	show a lot of nonnecessary information
-d	--reanimate	reanimate a PIC, that dont reacts anymore
-l	--list	list all supported PIC-types
-a	--auto	autodetect PIC-type
-b	--blank	check if the PIC is blank
-o	--boot	switch programmer into bootloader-mode
-f	--firmware	update firmware
-n	--normal	deactivates bootloader-mode
-u	--run	activate Vdd for target-PIC
-k	--calibration	Calibrate Vpp-generation of Brenner8
-t	--test	interactive test of the hardware

Optionen mit Parameter

-S	--SOCKET	PIC-Socket(Brenner8 only) or ICSP- connector
-F	--FAMILY	PIC-family (core architecture)
-H	--HEX	name of HEX-file
-I	--IN	name of input HEX-file (for write & compare)
-O	--OUT	name of output HEX-file (for read)
-P	--PIC	manual selection of the PIC-type

-h --help schow the help-screen

Diese Option bewirkt eine Auflistung aller möglichen Optionen von usburn. Diese Option kann mit allen anderen Optionen zusammen verwendet werden.

Beispiel:

```
usburn --help
usburn -h
```

-r --read read PIC-content into HEX-file

Der PIC wird ausgelesen, und sein Inhalt in eine HEX-Datei geschrieben. Standardmäßig lautet der Name der HEX-Datei „HexOut.hex“. Durch die Optionen –HEX (-H) und --OUT (-O) kann aber ein beliebiger anderer Name festgelegt werden. Wird –read in Kombination mit --write, --erase oder --remove verwendet, dann wird –read immer zuletzt ausgeführt. Man liest dann also den gelöschten bzw. neu beschriebenen PIC aus.

Ein eventuell bereits vorhandenen HEX-File gleichen Namens wird ohne Warnung überschrieben.

Beispiel:

```
usburn --read  --SOCKET ICSP  -F18 --OUT test.hex
usburn -r  -S28 -F16 -Otest.hex
```

-w --write write HEX-file into PIC

Der Inhalt einen HEX-Files wird in den PIC gebrannt. Standardmäßig lautet der Name der HEX-Datei „HexIn.hex“. Durch die Optionen --HEX (-H) und --IN (-I) kann aber ein beliebiger anderer Name festgelegt werden.

Die Option --write bewirkt automatisch ein Löschen des PICs und ein Entfernen von Codeprotection vor dem Brennen sowie ein Test des gebrannten PICs nach dem Brennen. Das Angeben der Optionen --erase, --remove und --compare ist also nicht nötig.
Beispiel:

```
usburn --write  --SOCKET ICSP  -F18 --IN test.hex
usburn -w  -S28 -F16 -Itest.hex
```

-c --compare compare PIC-content with HEX-file
Der Inhalt eines PIC wird mit dem Inhalt einen HEX-Files verglichen, und Fehler angezeigt. Standardmäßig lautet der Name der HEX-Datei „HexIn.hex“. Durch die Optionen --HEX (-H) und --IN (-I) kann aber ein beliebiger anderer Name festgelegt werden.
Normalerweise wird nur die Anzahl der Fehler angegeben. Wird aber gleichzeitig die Option --info verwendet, dann wird für jeden gefundenen Fehler die Adresse im PIC sowie Sollwert und Istwert angegeben.
Beispiel:

```
usburn --compare  --SOCKET ICSP  -F18 --IN test.hex
usburn -ic  -S28 -F16 -Itest.hex
```

-e --erase erase PIC
Der PIC wird gelöscht. Bei einigen PICs funktioniert diese Option nicht, solange Codeprotection aktiviert ist. In diesem Fall sollte --erase zusammen mit --remove verwendet werden.
Beispiel:

```
usburn --erase  --SOCKET ICSP  -F18
usburn -e  -S28 -F16
```

-p --remove remove codeprotection from PIC
Eine aktivierte Codeprotection des PICs wird deaktiviert, und dabei der Flash-Programmspeicher und der EEPROM-Datenspeicher gelöscht. Das Löschen der Konfiguration und der User-ID ist nicht bei allen PIC-Typen gewärleistet. Ein vollständiges Löschen ist aber durch die Kombination von --remove und --erase garantiert.
Beispiel:

```
usburn --erase  --remove --SOCKET ICSP  -F18
usburn -pe  -S28 -F16
```

-i --info show a lot of nonnecessary information
Usburn gibt, wärend es arbeitet, wichtige Informationen auf der Konsole aus. Durch die Option –info wird usburn extrem geschwätzig, und gibt eine Menge an Zusatzinformationen aus. Das ist in aller Regel nicht nötig, aber im Rahmen einer Fehlersuche kann diese Option hilfreich sein.
Beispiel:

```
usburn --info --compare  --SOCKET ICSP  -F18 --IN test.hex
usburn -ic  -S28 -F16 -Itest.hex
```

-d --reanimate reanimate a PIC, that dont reacts anymore

Es kommt gelegentlich vor, dass ein PIC so fehlerhaft gebrannt wird, dass er sich tot stellt und auf einenn Brenner nicht mehr reagiert. Das ist meist die Folge einer fehlerhaften Konfiguration.
In so einem Fall kann die --reanimate Option helfen, den PIC wiederzubeleben. Damit diese Option funktioniert, ist zwingen die gleichzeitige Nutzung von „--PIC“ erforderlich, da die automatische Typerkennung bei scheintoten PICs nicht funktioniert.
Beispiel:

```
usburn --reanimate --SOCKET ICSP  -F18  --PIC PIC18F2450
usburn -d  --SICSP  -F16  -PPIC16F876
```

-l --list list all supported PIC-types

Diese Option listet alle PIC-Typen auf, die die usburn-Software zusammen mit der eingesetzten Database und der Firmware des Brenners unterstützt.
Beispiel:

```
usburn --list
usburn -l
```

-b --blank check if the PIC is blank

Diese Option prüft, ob Programm- und Datenspeicher ein PIC leer sind. Diese Funktion ist im normalen Betrieb eigentlich überflüssig, kann aber wichtig sein, falls man gebrauchte PICs an dritte weitergeben möchte, und sicher sein will, dass sie keine Daten mehr enthalten.
Beispiel:

```
usburn --blank  --SOCKET ICSP  -F18
usburn --b -S28 -F16
```

-o --boot switch programmer into bootloader-mode

Diese Option aktiviert den Bootloader des Brenners. Für die normale Nutzung des Brenners ist diese Option nicht nötig.
Beispiel:

```
usburn --boot
usburn -b
```

-f --firmware upload new firmware

Diese Option aktiviert den Bootloader des Brenners und lädt eine neue Firmware in den SteuerPIC. Hat das fehlerfrei funktioniert, dann wird der Brenner wieder in den Normalmodus zurückgeschaltet.
Die Firmware ist ein HEX-file. Sein Name sollte mit der option --HEX oder --IN angegeben werden. Ansonsten wird versucht ein HEX-File namens „HexIn.hex“ zu verwenden.
Beispiel:

```
usburn --firmware  --IN fw_12.hex
usburn -f  -Ifw_12.hex
```

-n --normal deactivates bootloader-mode

Diese Option schaltet einen Brenner aus dem Bootloadermode in den normalen Brennermode. Befindet sich eine funktionsfähige Firmware im Brenner, dann wird er normal funktionieren. Ansonsten ist der Brenner dann funktionsunfähig, und kann nur durch den Bootloader-Jumper wieder in den Bootloader-Mode zurückgebracht werden.
Eigentlich ist diese Option für den Betrieb des Brenners nicht nötig, da der Bootloader nur für die Firmwareaktualisierung nötig ist (--boot) und nach erfolgreicher Firmwareaktualisierung der Bootloadermode automatisch beendet wird.
Beispiel:

```
usburn --normal
usburn -n
```

-u --run activate Vdd for target-PIC

(Nur für Brenner8P, Brenner8miniP, Brenner9. Nur mit PICs am ICSP-Anschluss.)
Diese Option bewirkt, dass der Brenner dem PIC Betriebsspannung zuschaltet, und das MCLR-Pin (Reset) auf Vdd-Pegel legt. Damit fängt der PIC an, das in ihm enthaltene Programm abzuarbeiten.
Das ist nur bei PICs sinnvoll, die sich auf einer Testplatine befinden, und mit dem Brenner über die ICSP-Schnittstelle verbunden sind.
Diese Aktivierung des PICs erfolgt, nachdem alle anderen Optionen von usburn abgearbeitet wurden. Man kann also „--run“ problemlos mit „--write“ kombinieren, dann wird erst die neue Software in den PIC gebrannt, und anschließend gleich gestartet.
Nach dem Start des PIC bleibnt usburn stehen, und wartet auf die Betätigung der „Enter/Return“ Taste . Daraufhin, wird dem PIC sie Betriebsspannung abgeschaltet und usburn beendet.
Beispiel:

```
usburn --write --run  --SOCKET ICSP  -F18 --IN test.hex
usburn -wu  -S28 -F16 -Itest.hex
```

-k --calibration starts the calibration of Brenner8

(Nur für Brenner8.)
Diese Option startet die Kalibrierung des Vpp-Boostkonverters im Brenner8.
Beispiel:

```
usburn --calibration
usburn -c
```

-t --test starts an interactive test of the hardware

Diese Option wird benutzt, um alle Signale des Brenners einzeln zu testen.
Beispiel:

```
usburn --test
usburn -t
```

-S --SOCKET PIC-Socket (Brenner8 only) or ICSP- connector

Damit der Brenner den PIC im Testsockel korrekt ansteuern kann, muss er wissen, wieviele Pins er hat oder ob er am ICSP-Verbinder angeschlossen ist. Das erfährt er durch den Parameter der –SOCKET-Option. Mögliche Parameter für PICs im Testsockel sind **8**, **14**, **18**, **28** und **40**. Das sind die Anzahl der Pins des PICs. Ist der PIC im ICSP-Verbinder angeschlossen, ist der Parameter **ICSP** zu wählen. Ich erinnere darann, dass nur PICs mit 14-Bit-Kern (PIC16F...und einige PIC12F...) und 16-Bit-Kern (PIC18F...) im Testsockel gebrannt werden können. Alle anderen benötigen den ICSP-Anschluss.
Nach dem Anstecken des Brenners, ist er so eingestellt, dass er sowohl den ICSP-anschluss wie auch 18-polige PICs im Testsockel korrekt bedienen kann. Wer also nur mit ICSP arbeitet (Brenner8mini, Brenner9), benötigt die –SOCKET-Option also nicht.
Wird mit der --FAMILY-Option eine PIC-Familie ausgewählt, die sich ausschließlich über den ICSP-Verbinder brennen läßt (PIC18FxxJxx, PIC24, dsPIC30, dsPIC33), dann wird der Brenner auf ICSP eingestellt, auch wenn eine andere --SOCKET-Option verwendet wird.
Beispiel:

```
usburn --write  --SOCKET ICSP  -F18 --IN test.hex
usburn -w  -S28 -F16 -Itest.hex
```

-F --FAMILY PIC-family (core architecture)
Es gibt eine Reihe grundsätzlich unterschiedlicher PIC-Prozessorfamilien. Der Brenner benötigt die Information, zu welcher Famile der Target-PIC gehört. Diese Einstellung erfolgt mit dem Parameter der --FAMILY-Option. Es gibt folgende Parameter: **10**, **16**, **18**, **18J**, **24**, **30**, **33**

Parameter	Beschreibung	Typen
10	alls 12-Bit-Kern-PICs	alle PIC10Fxxx alle PIC1xF5x alle PIC1xF5xx
16	alls 14-Bit-Kern-PICs	alle PIC16Fyx außer y=5 alle PIC16Fxxx alle PIC12Fyxx außer y=5
18	16-Bit-Kern-PICs	alle PIC18Fxxx alle PIX18Fxxxx
18J	16-Bit-Kern-PICs	alle PIC18FxxJxx
24	alle PIC24-Typen	alle PIC24....
30	alle dsPIC30F-Typen	alle dsPIC30F....
33	alle dsPIC33F-Typen	alle dsPIC33F....

Beispiel:

```
usburn --write  --SOCKET ICSP  -F18 --IN test.hex
usburn -w  -S28 -F16 -Itest.hex
```

-H --HEX name of HEX-file
Usburn kann ein Input-HEX-File einlesen (für --write und --compare) und ein Output-HEX-File schreiben (für –read). Die Standard-filenamen lauten

- HexIn.hex für Input
- HexOut.hex für Output

Diese Option legt einen neuen Filenamen fest, der sowohl für das Input-File (zum Brennen mit --write und Vergleichen mit --compare) wie auch für das Output-File (zum Lesen mit --read) verwendet wird.
Beispiel:

```
usburn --write  --SOCKET ICSP  -F18 --HEX test.hex
usburn -w  -S28 -F16 -Htest.hex
usburn --read  --SOCKET ICSP  -F18 --HEX  test.hex
usburn -r  -S28 -F16 -Htest.hex
```

-I --IN name of input HEX-file (for write & compare)
Usburn kann ein Input-HEX-File einlesen (für --write und --compare). Der Standard-filename lautet „**HexIn.hex**“.
Diese Option legt einen neuen Filenamen fest, der ausschließlich für das Input-File (zum Brennen mit --write und Vergleichen mit --compare) verwendet wird.
Beispiel:

```
usburn --write  --SOCKET ICSP  -F18 --IN test.hex
usburn -w  -S28 -F16 -Itest.hex
```

-O --OUT name of output HEX-file (for read)
Usburn kann ein Output-HEX-File schreiben (für --read). Der Standard-filename lautet „**HexOut.hex**“.
Diese Option legt einen neuen Filenamen fest, der ausschließlich für das Output-File (zum Lesen mit --read) verwendet wird.
Beispiel:

```
usburn --read  --SOCKET ICSP  -F18 --OUT test.hex
usburn -r  -S28 -F16 -Otest.hex
```

-P --PIC manual selection of the PIC-type
Voraussetzung für die Arbeit von usburn ist die Korrekte automatische Erkennung des PIC-Types. In zwei Situationen ist das aber unmöglich:

- beim Reanimieren eines PICs (--reanimate)
- beim Arbeiten mit 12-Bit-Kern PICs (--FAMILY 10)

In diesen Fällen muss usburn der exakte PIC-Typ mit dem Parameter der --PIC-Option mitgeteilt werden. Als Parameter ist der exakte PIC-Typ in Großschreibung ohne Gehäuse- uder Frequenzspezifische Ergänzung anzugeben. Bei low-power-PICs ist das „L“ im Bezeichner wegzulassen. Für einen „Pic16LF628A-04/P“ lautet der exakte Parameter folglich „PIC16F628A“.
Mit der option --list lassen sich alle unterstützten PIC-Typen auflisten. Dabei verwendet usburn die von ihm akzeptierten Bezeichner für PICs.
Beispiel:

```
usburn --reanimate  --PIC PIC18F2450
usburn --w  --SOCKET ICSP  -F10  -PPIC10F202
```

11.5 Brennen eines PIC

Damit usburn ein HEX-File in einen PIC brennt, benötigt das Programm folgende

Angaben

- Die Anweisung zum Brennen mit der: --write
- Die Information darüber, wid der PIC am Brenner angeschlossen ist: --SOCKET
- Die Familenzugehörigkeit des PIC: --FAMILY
- Den Namen des HEX-Files: --IN

Es folgen einige Beispiele jeweils mit langen und kurzen Optionen.

Um einen im Testsockel sitzenden PIC16F876 (28 Pins) mit dem File blink.hex zu brennen, genügt

```
usburn --write  --SOCKET 28  -FAMILY 16 --IN blink.hex
usburn -w  -S28 -F16 -Iblink.hex
```

Um einen via ICSP angeschlossenen PIC18F2450 mit dem File test.hex zu brennen, genügt:

```
usburn --write  --SOCKET ICSP  - FAMILY 18 --IN test.hex
usburn -w  -SICSP -F18 -Itest.hex
```

Um einen via ICSP angeschlossenen PIC10F202 mit dem File test.hex zu brennen, genügt:

```
usburn --write  --SOCKET ICSP  - FAMILY 10 --PIC ⏎
     PIC10F202 --IN test.hex
usburn -w  -SICSP -F10 -PPIC10F202 -Itest.hex
```

Um einen im Testsockel sitzenden PIC16F628 (18 Pins) auszulesen, und das Ergebnis in result.hex zu schreiben, genügt:

```
usburn --read  --SOCKET 18  -FAMILY 16 --OUT result.hex
usburn -r  -S18 -F16 -Oblink.hex
```

Um zu prüfen, ob ein im Testsockel sitzenden PIC16F876 (28 Pins) bereits die daten des HEX-Files blink.hex enthält, genügt:

```
usburn --compare  --SOCKET 28  -FAMILY 16 --IN blink.hex
usburn -c  -S28 -F16 -Iblink.hex
```

Um die neue Firmware fw_14.hex in den Steuer-PIC des Brenners zu laden, genügt:

```
usburn --firmware --IN fw_14.hex
usburn -f --IN fw_14.hex
```

12 Bootloader

Hin und wieder wird für den Brenner8/9 eine neue Firmware veröffentlicht. Das ist nötig, um gefundene Fehler in der Firmware zu beseitigen oder um die Möglichkeiten des Brenner8/9 zu erweitern.
Der Bootloader ermöglicht es, die Firmware des Brenner8/9 einfach und schnell zu aktualisieren.

Der Bootloader ist ein kleines Programm, das in einem bestimmten Speicherbereich des Brenner8-Steuer-PIC gebrannt wird. Dafür benutzt man ein PIC-Programmiergerät, das in der Lage ist PIC18F2550 zu programmieren. Das kann ein Brenner5 (mit der Software P18) oder ein Brenner8 (mindestens mit der Firmware V0.5 und US-Burn V1.2) sein.
Den Bootloader gibt es einzeln, oder in Kombination mit der aktuellen Firmware des Brenner8.

Im folgenden gehe ich davon aus, dass der Bootloader sich im Steuer-PIC befindet.

12.1 Benutzung des Bootloaders mit US-Burn (Windows)

Für den normalen Betrieb des Brenner8/9 wird der Bootloader nicht benötigt. Er bleibt inaktiv.
Soll aber eine neue Firmware in den Brenner8/9 geladen werden, muss man den Bootloader „wecken“. Dafür gibt es zwei Möglichkeiten

- Aktivieren des Bootloaders mit US-Burn
- Aktivieren des Bootloaders mit dem Jumper JP1

12.1.1 Aktivieren des Bootloaders mit US-Burn

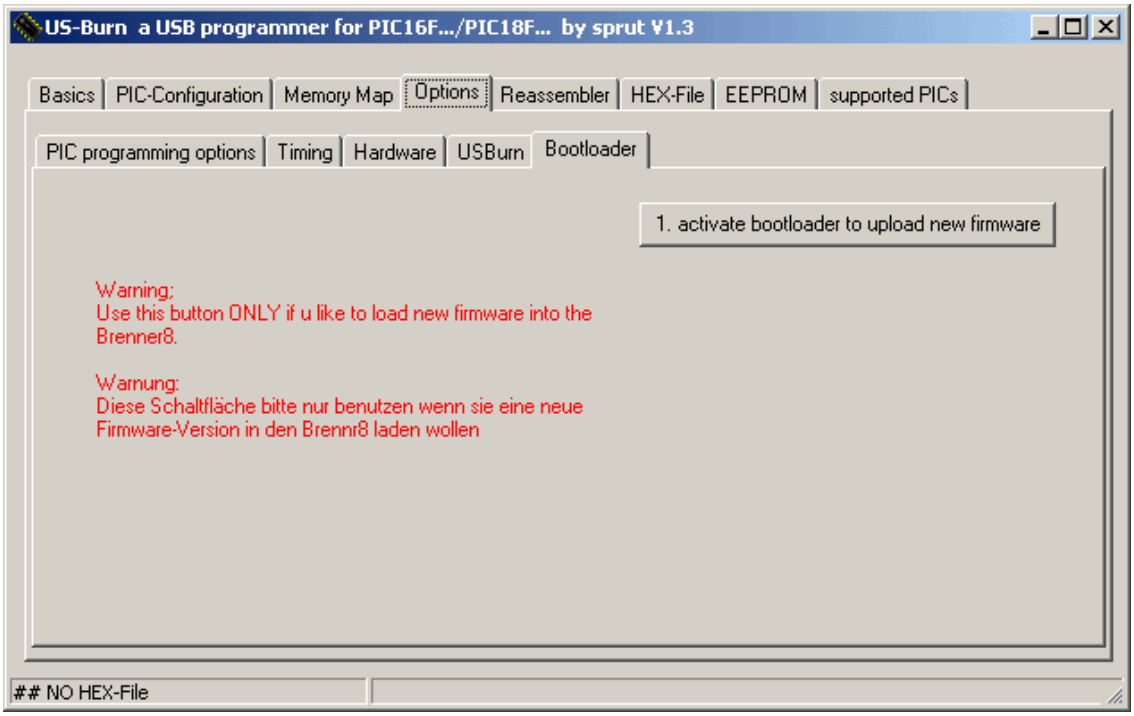


Abbildung 41 Aktivieren des Bootloaders mit US-Burn

Wenn der Brenner8 mit der alten Firmware (mindestens V0.6) noch normal funktioniert, kann man den Bootloader mit Hilfe von US-Burn (mindestens V1.3) aktivieren. Dazu klickt man im Fenster **Options-Bootloader** auf die Schaltfläche **1. aktivate bootloader to upload new firmware**.

Dadurch wird die bisher verwendete Firmware für ungültig erklärt, und der Brenner8 neu gestartet. Das dauert ca. 2 Sekunden. Beim Neustart erkennt der Bootloader des Brenner8, dass keine gültige Firmware im Steuer-PIC ist, und wird aktiv.
Beide LEDs des Brenner8 leuchten auf

12.1.2 Aktivieren des Bootloaders mit dem Jumper JP1

Sollte die Aktivierung des Bootloaders mit US-Burn aus irgendeinem Grunde nicht funktionieren, gibt es auch eine Hardware-Lösung.

Der Brenner8 ist vom PC zu trennen (USB-Kabel trennen). Dann ist der Jumper JP1 auf der Brenner8-Platine zu stecken. Auf älteren Brenner8-Versionen ohne Jumper sowie am Brenner8mini ist das Pin 1 des Steuer-PIC mit Masse zu verbinden.

Nun wird der Brenner8 wieder mit dem PC verbunden. Der Bootloader startet.
Beide LEDs des Brenner8 leuchten auf. Von nun an wird der Jumper (bzw. die Masseverbindung am Pin 1) nicht mehr benötigt.

12.1.3 Neue Firmware in den Brenner8 laden

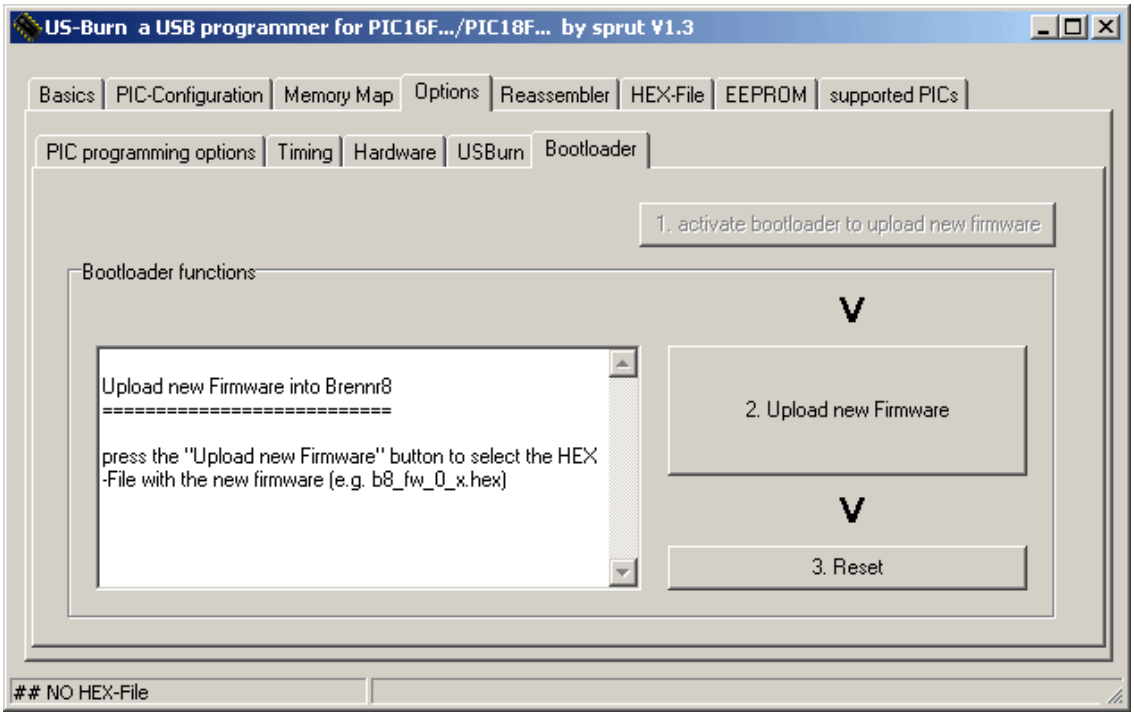


Abbildung 42 Neue Firmware in den Brenner8 laden

Hat man den Bootloader mit US-Burn aktiviert, zeigt US-Burn daraufhin das obige Fenster. Benutzte man einen Jumper, muss man US-Burn noch starten. Dann zeigt es automatisch das obige Fenster.

Der nächste Schritt ist das Klicken auf die Schaltfläche **2. Uplade new Firmware**. Daraufhin öffnet sich ein Auswahlfenster, in dem man das HEX-File mit der neuen Firmware auswählen muss. US-Burn

- lädt das HEX-File,
- brennt die neue Firmware in den Steuer-PIC,
- prüft die gebrannten Daten auf Korrektheit und
- erklärt die neue Firmware für gültig.

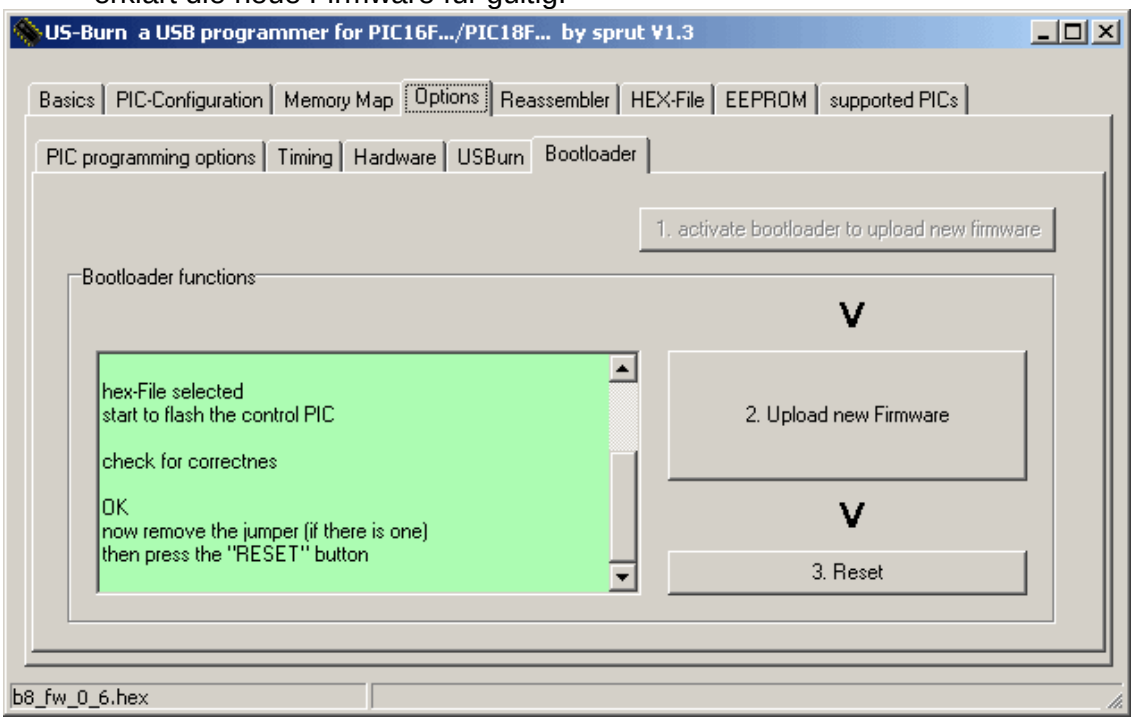


Abbildung 43 Neue Firmware in den Brenner8 geladen

Der Erfolg ist an der grünen Farbe des Info-Fensters zu erkennen.
Spätestens jetzt ist der Jumper JP1 zu entfernen.

Durch ein Klicken auf Reset wird der Brenner8 neu gestartet. Nach ca. 2 Sekunden sind der Brenner8 und US-Burn betriebsbereit.

Der Bootloader verändert NICHT die Kalibrierdaten des Brenner8. Eine Neukalibrierung ist also nicht erforderlich.

12.2 Benutzung des Bootloaders mit usburn (Linux)

Für den normalen Betrieb des Brenner8/9 wird der Bootloader nicht benötigt. Er bleibt inaktiv.
Soll aber eine neue Firmware in den Brenner8/9 geladen werden, muss man den Bootloader „wecken“. Dafür gibt es zwei Möglichkeiten

- Aktivieren des Bootloaders mit usburn (Normalfall)
- Aktivieren des Bootloaders mit dem Jumper JP1 (Backup-Methode)

12.2.1 Benutzen des Bootloaders mit usburn

Um in einen funktionierenden Brenner eine neue Firmware einzuspielen, wird usburn mit der Option --firmware bzw. -f verwendet. Außerdem ist noch mit der Option --IN bzw. -I der Name des HEX-Files anzugeben, das die Firmware enthält.
Das sieht dann z.B. so aus:

- usburn --firmware --IN fw_14.hex
- usburn -f -Ifw_14.hex

Nach dem Programmstart sucht usburn einen Brenner8/9, und schaltet ihn in den Bootloader-Mode. Das ist mit einem Reset des Brenners sowie seinem Abmelden und wieder-Anmelden am Betriebssystem verbunden, was 3..4 Sekunden dauert. Man sieht nun beide LEDs am Brenner aufleuchten.

Nun öffnet usburn das angegebene HEX-File und brennt es in den Steuer-PIC des Brenners. Anschließend wird der Steuer-PIC zur Kontrolle ausgelesen und auf Korrektheit überprüft. War alles erfolgreich, dann wird der Brenner wieder in den normalen Brenner-Modus zurückgeschaltet und usburn beendet. Man sieht beide LEDs des Brenners aufblinken.

Fertig.

Gab es beim Öffnen des HEX-files ein Problem (z.B. falscher Name) oder konnte es nicht korrekt gebrannt werden, dann beendet sich usburn. Der Brenner bleibt nun aber im Bootloader-Mode. Man kann durch einen erneuten Aufruf von usburn -f (nun z.B. mit dem richtigen Hex-File-Namen) das Firmwareupdate abschließen.

HINWEIS
Der Bootloader meldet sich nicht als „Brenner...“ im System an. Folglich erlaubt udev nicht automatisch den Zugriff für plugdev -Gruppenmitglieder. Aus diesem Grunde sollte die Firmwareaktivierung mit Root-Rechten durchgeführt werden.

Falls der Fehler beim Öffnen des HEX-files auftrat, und man den Brenner nun doch noch mit der alten Firmware nutzen will (diese befindet sich ja noch im PIC). Dann kann die alte Firmware durch Aufruf von

- usburn --normal
- usburn -n

Wieder reaktiviert werden.

Falls die Option --normal aber nach einem Brennfehler wärend des Firmwareupdates benutzt wird (sich also keine funktionsfähige Firmware im Brenner befindet) kann das zu einem unbrauchbaren Brenner führen, der sich per USB nicht mehr ansprechen läßt.
In diesem Fall ist der Jumper J1 zu stecken (oder Pin 1 des Steuer-PIC mit Vss -Masse- zu verbinden) und der Brenner vom PC zu trennen und nach ein paar Sekunden wieder anzustecken. Dadurch wird der Bootloader wieder aktiviert, und es kann mit der --firmware-Option eine Firmware eingespielt werden.

12.3 Falsches HEX-File in den Brenner8 geladen

Der Bootloader überprüft nicht, ob es sich bei dem Hex-File wirklich um eine Brenner8-Firmware handelt.
Falls man versehentlich ein falsches HEX-File ausgewählt und per Bootloader in den PIC geladen hat, ist nichts verloren (außer der alten Firmware natürlich). Der Bootloader verändert weder die Konfiguration des Steuer-PIC noch die im EEPROM gespeicherten Kalibrierdaten. Der Bootloader kann sich nicht selbst zerstören.

Der Bootloader kann in jedem Fall immer noch mit dem Jumper JP1 aktiviert werden, und dann kann man das richtige HEX-File in den Steuer-PIC brennen.

13 Anlagen

13.1 Brenner8P - Stromlaufplan

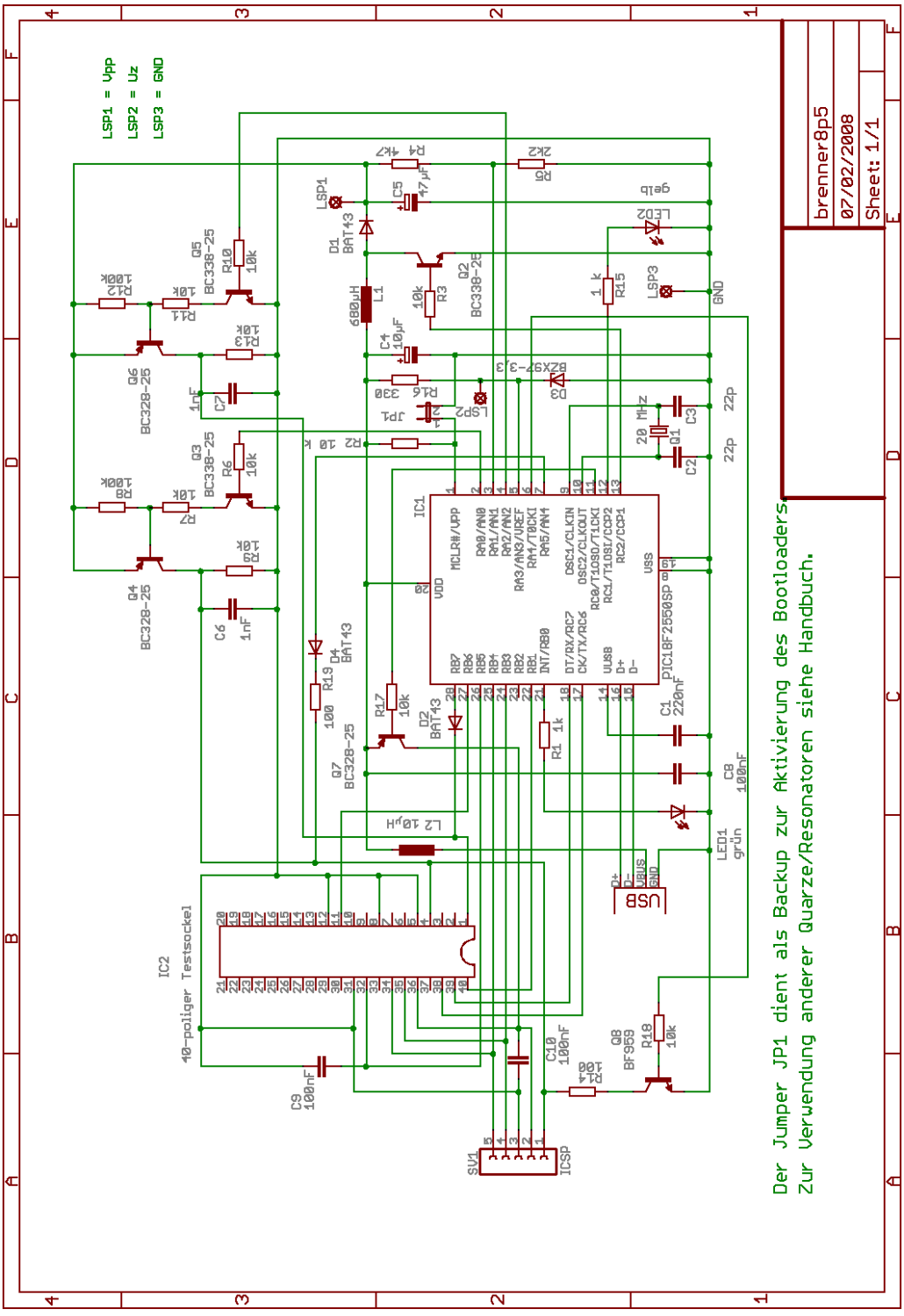


Abbildung 44 Stromlaufplan des Brenner8P (Revision 5)

13.2 Brenner8P – Bestückungsplan

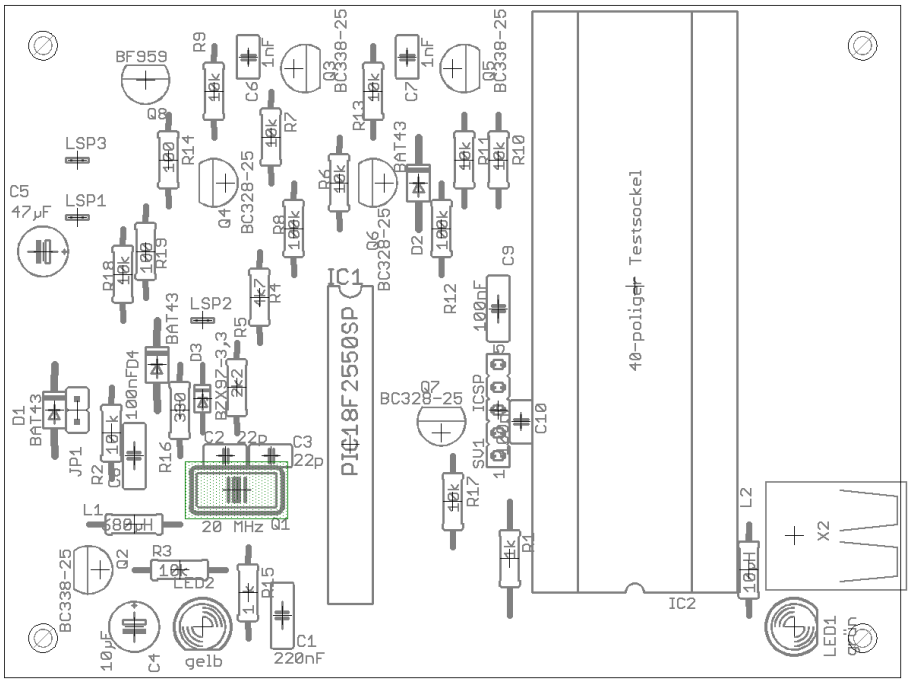


Abbildung 45 Bestückungsplan des Brenner8P

13.3 Brenner8P – Platinenlayout

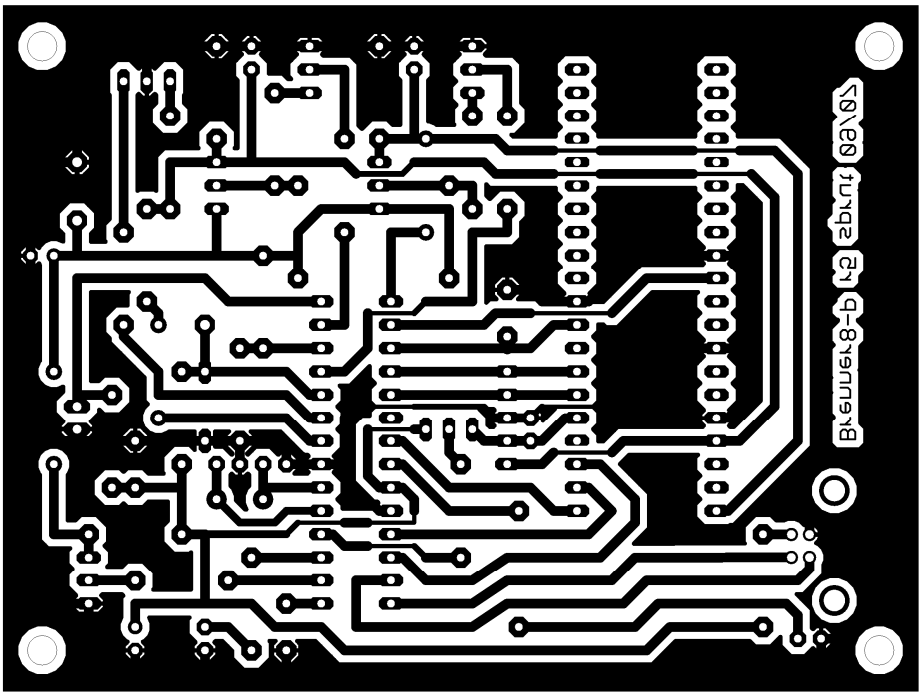


Abbildung 46 Layout der Brenner8P-Platine, 75mm x 100mm (nicht maßstabsgetreu)

13.4 Brenner8P - Stückliste

Partlist für Brenner8P R4

Part	Value	Reichelt	Conrad
C1	220nF	Z5U-5 220n	
C2, C3	22p	Kerko 22p	
C4	10µF	rad10/100	
C5	47µF	rad47/35	
C6, C7	1nF	X7R-2,5 1,0n	
C8	100nF	Z5U-5 100n	
C9	100nF	Z5U-5 100n	
C10	100nF	Z5U-2,5 100n	
D1, D2	BAT43	BAT 43	
D3	BZX97-3,3	ZF 3,3	
D4	BAT43	BAT 43	
IC1	PIC18F2550SP	PIC18F2550-I/SP	
	28-polige IC-Fassung	GS 28P-S	
IC2	40-poliger Testsockel	TEX 40	
JP1	Jumper	Jumper 2,54 RT	
L1	680µH	SMCC 680µ	
L2	10µH	SMCC 10µ	
LED1	grün	LED5mm2MAgn	
LED2	gelb	LED5mm2MAge	
LSP1	Lötstift		
LSP2	Lötstift		
LSP3	Lötstift		
Q1	20 MHz	20-HC49U-S	
Q2, Q3	BC338-25	BC338-25	
Q4	BC328-25	BC328-25	
Q5	BC338-25	BC338-25	
Q6, Q7	BC328-25	BC328-25	
Q8	BF959	BF959	
R1	1k	1/4W 1K	
R2, R3	10k	1/4W 10K	
R4	4k7	1/4W 4,7K	
R5	2k2	1/4W 2,2K	
R6, R7	10k	1/4W 10K	
R8	100k	1/4W 100K	
R9..R11	10k	1/4W 10K	
R12	100k	1/4W 100K	
R13	10k	1/4W 10K	
R14	100	1/4W 100	
R15	1 k	1/4W 1K	
R16	330	1/4W 330	
R17, R18	10k	1/4W 10K	
R19	100	1/4W 100	
SV1	ICSP	BL 1X10G 2,54	
X2	USB-B-H	USB BW	

Die Buchse SV1 ist auf 5 Pins zu kürzen. Ein 2-Pin langes Stück vom Rest ist für JP1 zu verwenden.
Alle Widerstands- und Kapazitätswerte sind unkritisch (25%) außer R4, R5.

Es muß mindestens eine Firmware V 0.5 verwendet werden.

Anstelle von Q1+C2+C3 kann auch ein 20 MHz Keramikresonator verwendet werden:

Part	Value	Reichelt	Conrad
Q1a	20 MHz	-	

Im Dokument
http://www.sprut.de/electronic/pic/projekte/brenner8/brenner8.pdf
beschreibe ich auch die Nutzung von Resonatoren mit z.B. 8 oder 12 MHz.

13.5 Brenner8mini-P Stromlaufplan

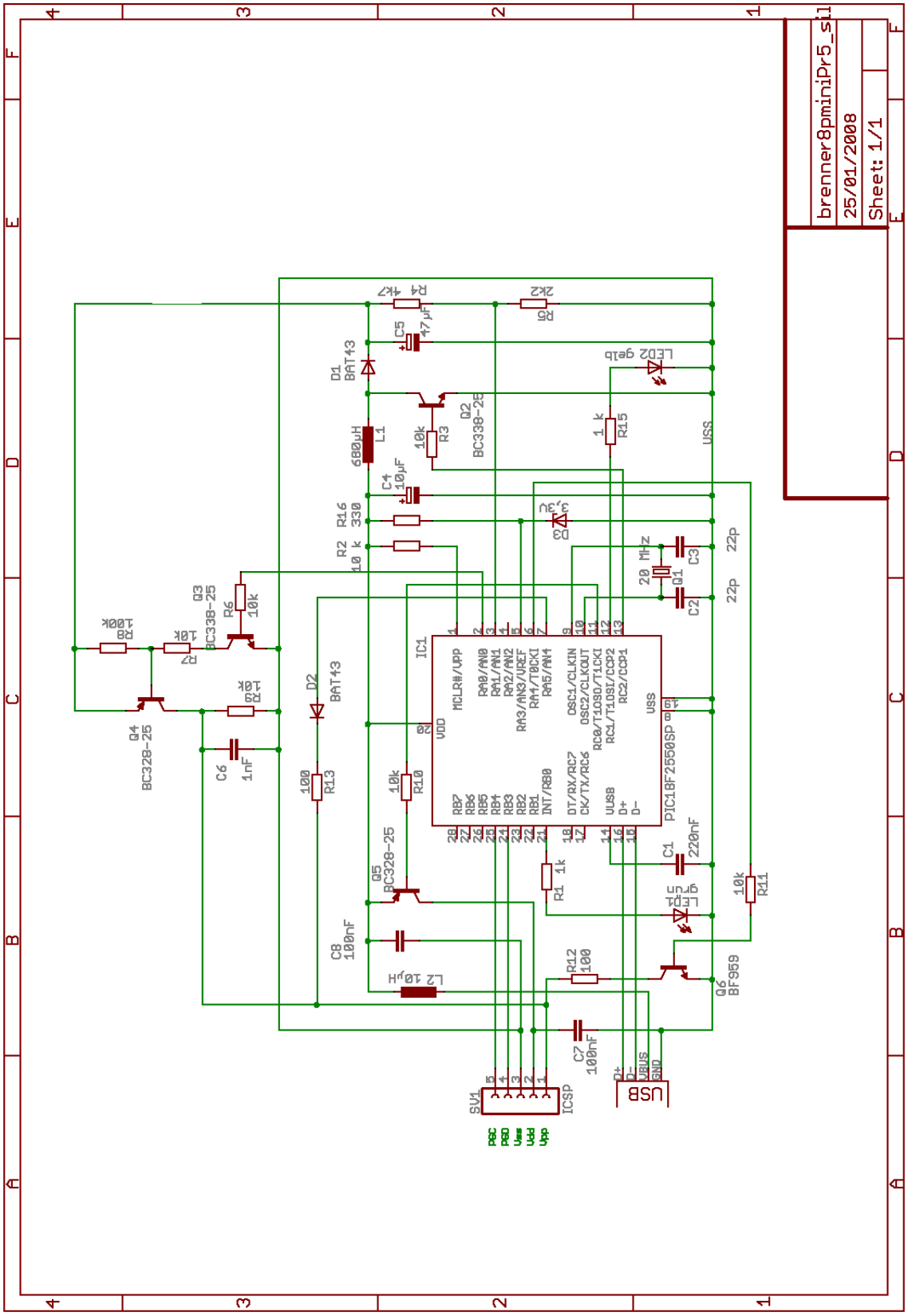


Abbildung 47 Stromlaufplan des Brenner8mini-P (Rev. 5 Silviu)

13.6 Brenner8mini-P – Bestückungsplan

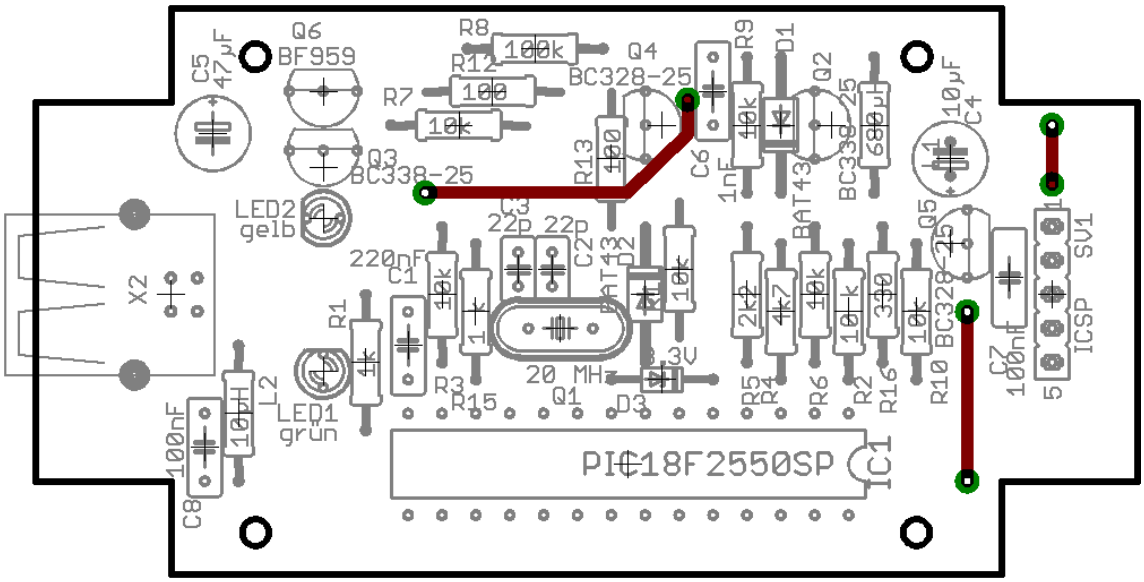


Abbildung 48 Bestückungsplan des Brenner8mini-P (Silviu)

13.7 Brenner8mini-P – Platinenlayout

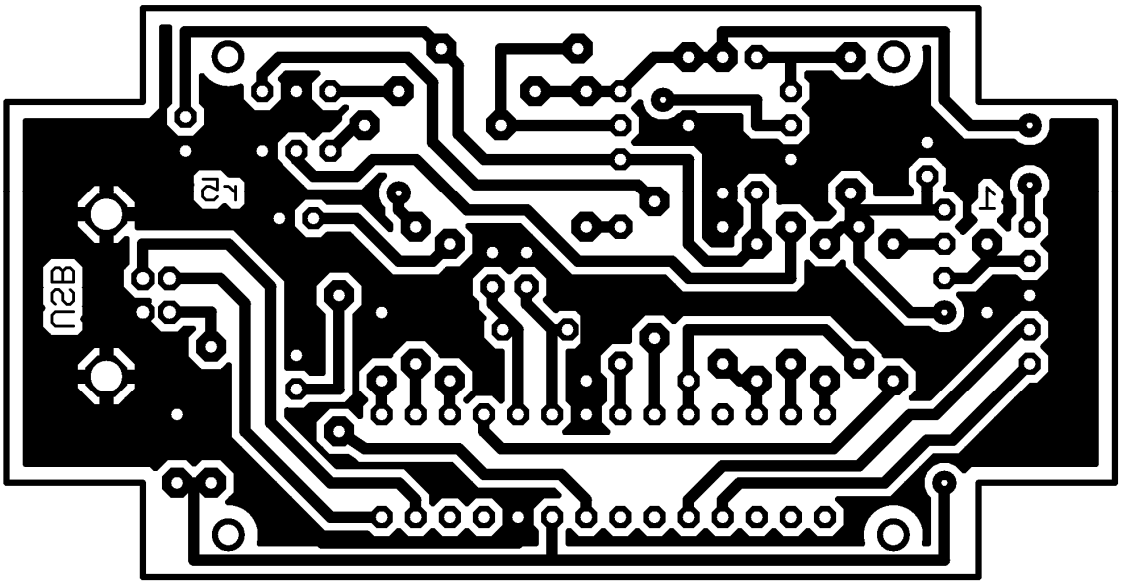


Abbildung 49 Layout des Brenner8mini-P, 83mm x 43mm (nicht maßstabsgetreu)

13.8 Brenner8mini-P – Stückliste

Partlist für Brenner8miniP r5 Silviu Layout

Part	Value	Reichelt
C1	220nF	Z5U-5 220n
C2, C3	22p	Kerko 22p
C4	10µF	rad10/100
C5	47µF	rad47/35
C6	1nF	X7R-5 1,0n
C7,C8	100nF	Z5U-5 100n
D1, D2	BAT43	BAT 43
D3	BZX90-3,3	ZF 3,3
L1	680µH	SMCC 680µ
L2	10µH	SMCC 10µ
LED1	grün	LED5mm2MAgn
LED2	gelb	LED5mm2MAge
Q1	20 MHz	20-HC49U-S
Q2, Q3	BC338-25	BC338-25
Q4	BC328-25	BC328-25
Q5	BC328-25	BC328-25
Q6	BF959	BF959
R1	1k	1/4W 1K
R2, R3	10k	1/4W 10K
R4	4k7	1/4W 4,7K
R5	2k2	1/4W 2,2K
R6, R7	10k	1/4W 10K
R8	100k	1/4W 100K
R9, R10, R11	10k	1/4W 10K
R12	100	1/4W 100
R13	100	1/4W 100
R15	1 k	1/4W 1K
R16	330	1/4W 330
SV1	ICSP	BL 1X10G 2,54
X2	USB-B-H	USB BW
IC1	PIC18F2550SP	PIC18F2550-I/SP
IC-Fassung	28-polig	GS 28P-S

Die Buchse SV1 ist auf 5 Pins zu kürzen.

Alle Widerstands- und Kapazitätswerte sind unkritisch (25%) außer R4, R5.

Es muß mindestens eine Firmware V 0.5 verwendet werden.

Anstelle von Q1+C2+C3 kann auch ein 20 MHz Keramikresonator verwendet werden:

Part	Value	Reichelt	Conrad
Q1a	20 MHz	-	

Im Dokument
http://www.sprut.de/electronic/pic/projekte/brenner8/brenner8.pdf
beschreibe ich auch die Nutzung von Resonatoren mit z.B. 8 oder 12 MHz