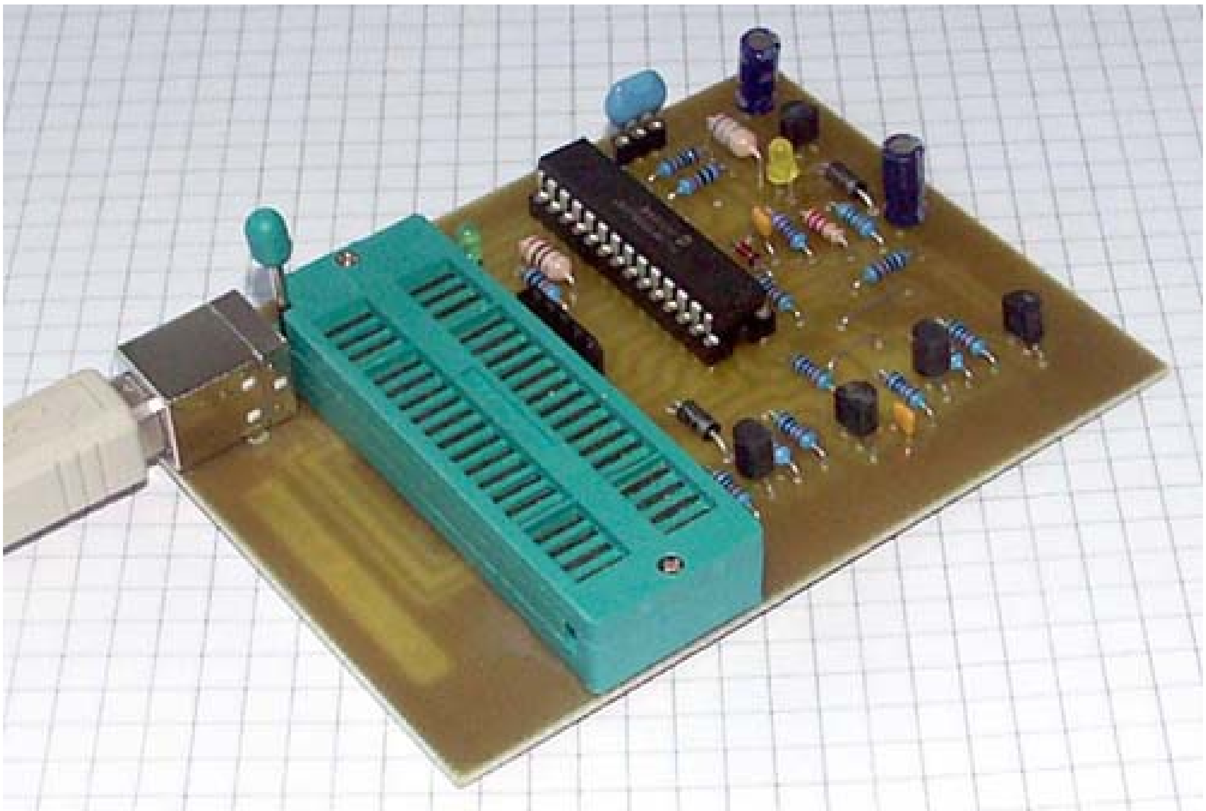
Manual for the Brenner8 (& 9) PIC-programmer



Author: sprut
Date: 23.05.2015

1 Table of contents

2 List of Figures

3 List of Tables

4 TERMS OF USE:

4.1 US-Burn for Windows / Firmware

THIS SOFTWARE CAN BE USED WITHOUT PAYING ANY LICENCE FEE FOR PRIVATE AND COMMERCIAL USE.

THE SOFTWARE IS PUBLISHED “AS IS". FÜR DIE EINHALTUNG ZUGESICHERTER EIGENSCHAFTEN ODER FÜR SCHÄDEN, DIE DURCH DEN EINSATZ ENTSTANDEN SEIN KÖNNTEN, ÜBERNIMMT DER AUTOR KEINERLEI HAFTUNG. YOU USE THIS SOFTWARE ON OWN RISK!

4.2 usburn for 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 Introduction

5.1 History

The first version of the Brenner8 was developed 2006 as a cost-efficient self made programmer with USB-interface. Cheap industrial produced programmers where not available at this time.

In the mean time the situation has changed. There are some industrial produced programmers on the marked for acceptable money. Some even can be integrated into the Microchip-developement-environment. There is no need to build a DIY-programmer anymore.

I will not spend much time into future improvement of my programmer-function. The Version V1.14 of USBurn has no additional functionality. Its only new feature is the use of a different USB-driver. This makes it much easier to use existing Brenner8/9 on modern Windows-Versions.

5.2 The Brenner8/9

The Brenner8/9 was designed to program PIC-Microchip-microcontrollers with flash-program memory.

The software reads Intel-Hex-files as they are produced e.g. by the software MPLAB, and burns them into the flash memory of the PICs. EEPROM- and configuration-data (if included in the HEX-file) are burned as well into the PIC.

For the use of Brenner8/9 one needs:

- the Brenner8 or Brenner9 itself
- the firmware for the control PIC of the Brenner8/9
- a PC with USB-interface and USB-cable
- WindowsXP (nt/2k/Vista32) and the Windows-software: US-Burn
- or Linux with libusb and the Linux program usburn
- the Database-files
- some PICs need an ICSP-adapter

The Brenner8/9 and its software will continuously be improved. This handbook is based on:

- Brenner8 Rev.5 with firmware Fw 0.16 and 0.17
- Brenner9 Rev.0 with firmware Fw 3.16
- USBurn V1.14 for Windows
- usburn V0.4 for Linux
- Database30

5.3 Supported PIC-types

The Brenner8 was designed, to burn all PIC-Microcontrollers with Flash-program memory and 5V operating voltage. These are:

- all PIC18Fxxx und PIC18Fxxxx (no PIC18FxxJxx or PIC18FxxKxx)
- all PIC16Fxx und PIC16Fxxx
- all PIC12Fxxx
- all PIC10Fxxx
- all dsPIC30Fxxxx

By the use of a special adapter this additional 3,3V PICs can be programmed by the Brenner8:

- all PIC18FxxKxx

The Brenner9 was designed, to burn all PIC-Microcontrollers with Flash-program memory and 3.3V operating voltage. These are:

- all PIC18FxxJxx
- all PIC24FJxxxx
- all PIC24HJxxxx
- all dsPIC33Fxxxx

The following table lists all types:

Types in parenthesis can not be programmed directly by the Brenner8. There maximum operating voltage is not high enough for use with Brenner8.

PIC18FxxKxx can be programmed by the Brenner8, if a special adapter is used, to reduce the voltage level.

supported members of PICxxF/dsPICxxF -series
-- PIC-database V.24 (06/05/2010)

Name	Pins	Prog [kB]	EEPRM [B]	TMR	Capt -ure	PWM CMP	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	1	-/-	-/-	-	-	4	6
12F617	8	3	-	3	1	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	1	-/-	-/-	-	-	4	6
12F1822	8	3	256	3	1	1	1/1	1/1	-	-	4	6
16F54	18	-	-	1	-	-	-/-	-/-	-	-	-	12
16F57	28	2	-	1	-	-	-/-	-/-	-	-	-	20
16F59	40	2	-	1	-	-	-/-	-/-	-	-	-	32
16F72	28	3	-	3	1	1	-/1	1/1	-	-	5	22
16F73	28	6	-	3	2	2	1/1	1/1	-	-	5	22
16F74	40	6	-	3	2	2	1/1	1/1	-	-	8	33
16F76	28	11	-	3	2	2	1/1	1/1	-	-	5	22
16F77	40	11	-	3	2	2	1/1	1/1	-	-	8	33
16F84	18	1	64	1	-	-	-/-	-/-	-	-	-	13
16F87	18	6	256	3	1	1	1/1	1/1	-	-	-	16
16F88	18	6	256	3	1	1	1/1	1/1	-	-	7	16
16F505	14	1	-	1	-	-	-/-	-/-	-	-	-	12
16F506	14	1	-	1	-	-	-/-	-/-	-	-	3	12
16F526	14	1	64	1	-	-	-/-	-/-	-	-	3	12

Name	Pins	Prog [kB]	EEPRM [B]	TMR	Capt-ure	PWM CMP	UART /SPI	SSP I2C	CAN	USB	ADC	IO-Pins
16F610	14	1	-	2	-	-	-/-	-/-	-	-	-	12
16F616	14	3	-	3	1	1	-/-	-/-	-	-	8	12
16F627	18	1	128	3	1	1	1/-	-/-	-	-	-	16
16F628	18	3	128	3	1	1	1/-	-/-	-	-	-	16
16F630	14	1	128	2	-	-	-/-	-/-	-	-	-	12
16F631	20	1	128	2	-	-	-/-	-/-	-	-	-	18
16F636	14	3	256	2	-	-	-/-	-/-	-	-	-	12
16F639	20	3	256	2	-	-	-/-	-/-	-	-	-	12
16F676	14	1	128	2	-	-	-/-	-/-	-	-	8	12
16F677	20	3	256	2	-	-	-/1	1/1	-	-	12	18
16F684	14	3	256	3	1	1	-/-	-/-	-	-	8	12
16F685	20	6	256	3	1	1	-/-	-/-	-	-	12	18
16F687	20	3	256	2	-	-	1/1	1/1	-	-	12	18
16F688	14	6	256	2	-	-	1/-	-/-	-	-	8	12
16F689	20	6	256	2	-	-	1/1	1/1	-	-	12	18
16F690	20	6	256	3	1	1	1/1	1/1	-	-	12	18
16F716	18	3	-	3	1	1	-/-	-/-	-	-	4	13
16F720	20	3	-	3	1	1	1/1	1/1	-	-	12	18
16F721	20	6	-	3	1	1	1/1	1/1	-	-	12	18
16F722	28	3	-	3	2	2	1/1	1/1	-	-	11	25
16F723	28	6	-	3	2	2	1/1	1/1	-	-	11	25
16F724	40	6	-	3	2	2	1/1	1/1	-	-	14	36
16F726	28	11	-	3	2	2	1/1	1/1	-	-	11	25
16F727	40	11	-	3	2	2	1/1	1/1	-	-	14	36
16F737	28	6	-	3	3	3	1/1	1/1	-	-	11	25
16F747	40	6	-	3	3	3	1/1	1/1	-	-	14	36
16F767	28	11	-	3	3	3	1/1	1/1	-	-	11	25
16F777	40	11	-	3	3	3	1/1	1/1	-	-	14	36
16F785	20	3	256	3	1	2	-/-	-/-	-	-	12	18
16F818	18	1	128	3	1	1	-/1	1/1	-	-	5	16
16F819	18	3	256	3	1	1	-/1	1/1	-	-	5	16
16F870	28	3	64	3	1	1	1/-	-/-	-	-	5	22
16F871	40	3	64	3	1	1	1/-	-/-	-	-	8	33
16F872	28	3	64	3	1	1	-/-	-/1	-	-	5	22
16F873	28	6	128	3	2	2	1/-	-/1	-	-	5	22
16F874	40	6	128	3	2	2	1/-	-/1	-	-	8	33
16F876	28	11	256	3	2	2	1/-	-/1	-	-	5	22
16F877	40	11	256	3	2	2	1/-	-/1	-	-	8	33
16F882	28	3	128	3	2	2	1/1	1/1	-	-	11	25
16F883	28	6	256	3	2	2	1/1	1/1	-	-	11	25
16F884	40	6	256	3	2	2	1/1	1/1	-	-	14	36
16F886	28	11	256	3	2	2	1/1	1/1	-	-	11	25
16F887	40	11	256	3	2	2	1/1	1/1	-	-	14	36
16F913	28	6	256	3	1	1	1/1	1/1	-	-	5	25
16F914	40	6	256	3	2	2	1/1	1/1	-	-	8	36
16F916	28	11	256	3	1	1	1/1	1/1	-	-	5	25
16F917	40	11	256	3	2	2	1/1	1/1	-	-	8	36
16F946	64	11	256	3	2	2	1/1	1/1	-	-	8	54

Name	Pins	Prog [kB]	EEPRM [B]	TMR	Capt -ure	PWM CMP	UART /SPI	SSP I2C	CAN	USB	ADC	IO-Pins
16F1823	14	3	256	3	1	1	1/1	1/1	-	-	8	12
16F1824	14	6	256	5	4	4	1/1	1/1	-	-	8	12
16F1825	14	11	256	5	4	4	1/1	1/1	-	-	8	12
16F1826	18	3	256	3	1	1	1/1	1/1	-	-	12	16
16F1827	18	6	256	5	4	4	1/2	2/2	-	-	12	16
16F1828	20	6	256	5	4	4	1/1	1/1	-	-	12	18
16F1829	20	6	256	5	4	4	1/2	2/2	-	-	12	18
16F1933	28	6	256	5	5	5	1/1	1/1	-	-	11	25
16F1934	40	6	256	5	5	5	1/1	1/1	-	-	14	36
16F1936	28	11	256	5	5	5	1/1	1/1	-	-	11	25
16F1937	40	11	256	5	5	5	1/1	1/1	-	-	14	36
16F1938	28	22	256	5	5	5	1/1	1/1	-	-	11	25
16F1939	40	22	256	5	5	5	1/1	1/1	-	-	14	36
16F1946	64	11	256	5	5	5	2/2	2/2	-	-	17	53
16F1947	64	22	256	5	5	5	2/2	2/2	-	-	17	53
16F84A	18	1	64	1	-	-	-/-	-/-	-	-	-	13
16F627A	18	1	128	3	1	1	1/-	-/-	-	-	-	16
16F628A	18	3	128	3	1	1	1/-	-/-	-	-	-	16
16F648A	18	6	256	3	1	1	1/-	-/-	-	-	-	16
16F873A	28	6	128	3	2	2	1/-	-/1	-	-	5	22
16F874A	40	6	128	3	2	2	1/-	-/1	-	-	8	33
16F876A	28	11	256	3	2	2	1/-	-/1	-	-	5	22
16F877A	40	11	256	3	2	2	1/-	-/1	-	-	8	33
(16LF720)	20	2	-	3	1	1	1/1	1/1	-	-	12	18
(16LF721)	20	4	-	3	1	1	1/1	1/1	-	-	12	18
(16LF722)	28	2	-	3	2	2	1/1	1/1	-	-	11	25
(16LF723)	28	4	-	3	2	2	1/1	1/1	-	-	11	25
(16LF724)	40	4	-	3	2	2	1/1	1/1	-	-	14	36
(16LF726)	28	8	-	3	2	2	1/1	1/1	-	-	11	25
(16LF727)	40	8	-	3	2	2	1/1	1/1	-	-	14	36
(16LF1823)	14	2	256	3	1	1	1/1	1/1	-	-	8	12
(16LF1824)	14	4	256	5	4	4	1/1	1/1	-	-	8	12
(16LF1825)	14	8	256	5	4	4	1/1	1/1	-	-	8	12
(16LF1826)	18	2	256	3	1	1	1/1	1/1	-	-	12	16
(16LF1827)	18	4	256	5	4	4	1/2	2/2	-	-	12	16
(16LF1828)	20	4	256	5	4	4	1/1	1/1	-	-	12	18
(16LF1829)	20	4	256	5	4	4	1/2	2/2	-	-	12	18
(16LF1933)	28	4	256	5	5	5	1/1	1/1	-	-	11	25
(16LF1934)	40	4	256	5	5	5	1/1	1/1	-	-	14	36
(16LF1936)	28	8	256	5	5	5	1/1	1/1	-	-	11	25
(16LF1937)	40	8	256	5	5	5	1/1	1/1	-	-	14	36
(16LF1938)	28	16	256	5	5	5	1/1	1/1	-	-	11	25
(16LF1939)	40	16	256	5	5	5	1/1	1/1	-	-	14	36
(16LF1946)	64	8	256	5	5	5	2/2	2/2	-	-	17	53
(16LF1947)	64	16	256	5	5	5	2/2	2/2	-	-	17	53

Name	Pins	Prog [kB]	EEPRM [B]	TMR	Capt-ure	PWM CMP	UART /SPI	SSP I2C	CAN	USB	ADC	IO-Pins
18F242	28	16	256	4	2	2	1/-	1/-	-	-	5	23
18F248	28	16	256	4	1	1	1/-	1/-	1	-	5	23
18F252	28	32	256	4	2	2	1/-	1/-	-	-	5	23
18F258	28	32	256	4	1	1	1/-	1/-	1	-	5	23
18F442	40	16	256	4	2	2	1/-	1/-	-	-	8	34
18F448	40	16	256	4	2	2	1/-	1/-	1	-	8	34
18F452	40	32	256	4	2	2	1/-	1/-	-	-	8	34
18F458	40	32	256	4	2	2	1/-	1/-	1	-	8	34
18F1220	18	4	256	4	1	1	1/-	-/-	-	-	7	16
18F1230	18	4	128	2	-	-	1/-	-/-	-	-	4	16
18F1320	18	8	256	4	1	1	1/-	-/-	-	-	7	16
18F1330	18	8	128	2	-	-	1/-	-/-	-	-	4	16
18F2220	28	4	256	4	2	2	1/1	1/1	-	-	10	25
18F2221	28	4	256	4	2	2	1/1	1/1	-	-	10	25
18F2320	28	8	256	4	2	2	1/1	1/1	-	-	10	25
18F2321	28	8	256	4	2	2	1/1	1/1	-	-	10	25
18F2331	28	8	256	4	2	2	1/1	1/1	-	-	5	24
18F2410	28	16	-	4	2	2	1/1	1/1	-	-	10	25
18F2420	28	16	256	4	2	2	1/1	1/1	-	-	10	25
18F2423	28	16	256	4	2	2	1/1	1/1	-	-	10	25
18F2431	28	16	256	4	2	2	1/1	1/1	-	-	5	24
18F2450	28	16	-	3	1	1	1/-	-/-	-	1	10	23
18F2455	28	24	256	4	2	2	1/1	1/1	-	1	10	24
18F2458	28	24	256	4	2	2	1/1	1/1	-	1	10	24
18F2480	28	16	256	4	1	1	1/1	1/1	1	-	8	25
18F2510	28	32	-	4	2	2	1/1	1/1	-	-	10	25
18F2515	28	48	-	4	2	2	1/1	1/1	-	-	10	25
18F2520	28	32	256	4	2	2	1/1	1/1	-	-	10	25
18F2523	28	32	256	4	2	2	1/1	1/1	-	-	10	25
18F2525	28	48	1024	4	2	2	1/1	1/1	-	-	10	25
18F2550	28	32	256	4	2	2	1/1	1/1	-	1	10	24
18F2553	28	32	256	4	2	2	1/1	1/1	-	1	10	24
18F2580	28	32	256	4	1	1	1/1	1/1	1	-	8	25
18F2585	28	48	1024	4	1	1	1/1	1/1	1	-	8	25
18F2610	28	64	-	4	2	2	1/1	1/1	-	-	10	25
18F2620	28	64	1024	4	2	2	1/1	1/1	-	-	10	25
18F2680	28	64	1024	4	1	1	1/1	1/1	1	-	8	25
18F2682	28	80	1024	4	1	1	1/1	1/1	1	-	8	25
18F2685	28	96	1024	4	1	1	1/1	1/1	1	-	8	25
18F4220	40	4	256	4	2	2	1/1	1/1	-	-	13	36
18F4221	40	4	256	4	2	2	1/1	1/1	-	-	13	36
18F4320	40	8	256	4	2	2	1/1	1/1	-	-	13	36
18F4321	40	8	256	4	2	2	1/1	1/1	-	-	13	36
18F4331	40	8	256	4	2	2	1/1	1/1	-	-	9	36
18F4410	40	16	-	4	2	2	1/1	1/1	-	-	13	36
18F4420	40	16	256	4	2	2	1/1	1/1	-	-	13	36
18F4423	40	16	256	4	2	2	1/1	1/1	-	-	13	36
18F4431	40	16	256	4	2	2	1/1	1/1	-	-	9	36
18F4450	40	16	-	3	1	1	1/-	-/-	-	1	13	34
18F4455	40	24	256	4	2	2	1/1	1/1	-	1	13	35
18F4458	40	24	256	4	2	2	1/1	1/1	-	1	13	35
18F4480	40	16	256	4	2	2	1/1	1/1	1	-	11	36

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

Name	Pins	Prog [kB]	EEPRM [B]	TMR	Capt-ure	PWM CMP	UART /SPI	SSP I2C	CAN	USB	ADC	IO-Pins
(30F1010)	28	6	-	2	-	1	1/1	-/1	-	-	6	21
30F2010	28	12	1024	4	4	2	1/1	-/1	-	-	6	20
30F2011	18	12	-	4	2	2	1/1	-/1	-	-	8	12
30F2012	28	12	-	4	2	2	1/1	-/1	-	-	10	20
(30F2020)	28	12	-	4	-	2	1/1	-/1	-	-	8	21
(30F2023)	44	12	-	4	-	2	1/1	-/1	-	-	12	35
30F3010	28	24	1024	7	4	2	1/1	-/1	-	-	6	20
30F3011	40	24	1024	7	4	4	2/1	-/1	-	-	9	30
30F3012	18	24	1024	4	2	2	1/1	-/1	-	-	8	12
30F3013	28	24	1024	4	2	2	2/1	-/1	-	-	10	20
30F3014	40	24	1024	4	2	2	2/1	-/1	-	-	13	30
30F4011	40	48	1024	7	4	4	2/1	-/1	1	-	9	30
30F4012	28	48	1024	7	4	2	1/1	-/1	1	-	6	20
30F4013	40	48	1024	7	4	4	2/1	-/1	1	-	13	30
30F5011	64	66	1024	7	8	8	2/2	-/1	2	-	16	52
30F5013	80	66	1024	7	8	8	2/2	-/1	2	-	16	68
30F5015	64	66	1024	7	4	4	1/2	-/1	1	-	16	52
30F5016	80	66	1024	7	4	4	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	7	8	8	2/2	-/1	1	-	16	52
30F6010A	80	144	4096	7	8	8	2/2	-/1	2	-	16	68
30F6011A	64	132	2048	7	8	8	2/2	-/1	2	-	16	52
30F6012A	64	144	4096	7	8	8	2/2	-/1	2	-	16	52
30F6013A	80	132	2048	7	8	8	2/2	-/1	2	-	16	68
30F6014A	80	144	4096	7	8	8	2/2	-/1	2	-	16	68
30F2011es	18	12	-	4	2	2	1/1	-/1	-	-	8	12
30F2012es	28	12	-	4	2	2	1/1	-/1	-	-	10	20
30F6010es	80	144	4096	5	8	1	2/2	-/1	2	-	16	68

Name	Pins	Prog [kB]	EEPRM [B]	TMR	Capt -ure	PWM CMP	UART /SPI	SSP I2C	CAN	USB	ADC	IO-Pins
18F13K22	20	8	256	4	1	1	1/1	1/1	-	-	12	18
18F13K50	20	8	256	4	1	1	1/1	1/1	-	1	9	15
18F14K22	20	16	256	4	1	1	1/1	1/1	-	-	12	18
18F14K50	20	16	256	4	1	1	1/1	1/1	-	1	9	15
18F23K20	28	8	256	4	2	2	1/1	1/1	-	-	10	25
18F23K22	28	8	256	4	2	2	2/2	2/2	-	-	17	25
18F24K20	28	16	256	4	2	2	1/1	1/1	-	-	10	25
18F24K22	28	16	256	4	2	2	2/2	2/2	-	-	17	25
18F25K20	28	32	256	4	2	2	1/1	1/1	-	-	10	25
18F25K22	28	32	256	7	5	5	2/2	2/2	-	-	17	25
18F26K20	28	64	1024	4	2	2	1/1	1/1	-	-	10	25
18F26K22	28	64	1024	7	5	5	2/2	2/2	-	-	17	25
18F43K20	40	8	256	4	2	2	1/1	1/1	-	-	13	36
18F43K22	40	8	256	4	2	2	2/2	2/2	-	-	28	36
18F44K20	40	16	256	4	2	2	1/1	1/1	-	-	13	36
18F44K22	40	16	256	4	2	2	2/2	2/2	-	-	28	36
18F45K20	40	32	256	4	2	2	1/1	1/1	-	-	13	36
18F45K22	40	32	256	7	5	5	2/2	2/2	-	-	28	36
18F46K20	40	64	1024	4	2	2	1/1	1/1	-	-	13	36
18F46K22	28	64	1024	7	5	5	2/2	2/2	-	-	28	25
18LF13K22	20	8	256	4	1	1	1/1	1/1	-	-	12	18
18LF13K50	20	8	256	4	1	1	1/1	1/1	-	1	9	15
18LF14K22	20	16	256	4	1	1	1/1	1/1	-	-	12	18
18LF14K50	20	16	256	4	1	1	1/1	1/1	-	1	9	15
18LF23K22	28	8	256	4	2	2	2/2	2/2	-	-	17	25
18LF24K22	28	16	256	4	2	2	2/2	2/2	-	-	17	25
18LF25K22	28	32	256	7	5	5	2/2	2/2	-	-	17	25
18LF26K22	28	64	1024	7	5	5	2/2	2/2	-	-	17	25
18LF43K22	40	8	256	4	2	2	2/2	2/2	-	-	28	36
18LF44K22	40	16	256	4	2	2	2/2	2/2	-	-	28	36
18LF45K22	40	32	256	7	5	5	2/2	2/2	-	-	28	36
18LF46K22	28	64	1024	7	5	5	2/2	2/2	-	-	28	25

293 PIC-Types

By continuous updating of firmware and software the list of supported types can be kept up-to-date.

All 14&16-bit-core-PICs with DIL-housing (8 / 14 / 18 / 28 / 40 Pin-DIL-housing) can be programmed within the 40-pin test socked of the Brenner8. All other types have to be hooked up to the ICSP-connector of the Brenner8/9 via an ICPS-adapter.

The dsPIC30Fxxxx (and the PIC10Fxxx) have to be connected via ICPS-adapters too. They don’t fit into the on-board test socked.
Only the Brenner8P/Brenner8miniP is able to program the dsPIC30F-types.

By an additional Adapter, the Brenner8P can be converted into a Brenner9 to program the following 3,3V-PICs:

- PIC18FxxJxx
- PIC24xxxx
- dsPIC33Fxxxx

5.4 The PIC1xF5x und PIC1xF5xx -Problem

Several 10-bit-core-PICs are (because of historical reasons) not labeled PIC10Fxxx. However, in USBurn has to be selected the PIC-family PIC10F to program these PICs.

These are all PIC1xFxxx-types, with the letter 5 directly behind the F in the name. The following table lists all this types:

Table 1 PIC1xF5...

Name	Pins	Fits in Testsocket?
PIC12F508	8	Yes
PIC12F509	8	Yes
PIC12F510	8	Yes
PIC12F519	8	Yes
PIC16F54	18	Yes
PIC16F57	28	**No**
PIC16F59	40	**No**
PIC16F505	14	Yes
PIC16F506	14	Yes
PIC16F526	14	Yes

All this types can be programmed via ICSP, but some can be programmed in the test socked of the Brenner8 as well (see table).

In USBurn the PIC-family PIC10Fxxx is selected, then immediately **ICSP/18-Pins** is selected too, and this selection is locked by default. But the most of thePIC1xF5x-PICs need **8/14/20-Pins**.

To use this socked-setting, one has first to check **I know, that several PIC-Types with DIL-housings require ICSP-connection** at the page **Options - PIC programming options**.

After this was done, **ICSP/18-Pins** will still be selected automatically, but it will not be locked. The user can then select **8/14/20-Pins** manually.

However, PIC16F57 and PIC16F59 can not be programmed inside the test socked!

6 Build the hardware

6.1 Variants

Multiple versions of the Brenner8 exist, but all use the same firmware and software. Not all versions can be used for all jobs:

Table 2 Brenner8-versions

Type \ can program	PIC in test socket	PICs via ICSP-adapter	PICs via ICSP in circuit	dsPIC30Fxxxx
Brenner8	X	X	-	-
Brenner8-P	**X**	**X**	**X**	**X**
Brenner8mini	-	X	-	-
Brenner8mini-P	-	X	X	X

In the following text describes the Brenner8 except otherwise noted.

6.1.1 Brenner8

The “classic” Brenner8 has one 40-pin on-board test socket and one 5-pin ICSP-connector. Its primary use is to program PICs in the test socked.
With an ICSP-adapter PICs can be connected to the Brenner8 and programmed too.

The “classic” Brenner8 was not designed to program PIC in circuit via ICSP-cable. Its outputs are not powerful enough to drive other electric parts beside the PIC (what happens often during in circuit programming). Only capacitors up to 100µF (combined) are allowed in parallel to Vss and Vdd pins.

6.1.2 Brenner8-P

The Brenner8-P has all positive qualities of the Brenner8 but a more powerful Vdd-output for ICSP. Thus it can program PICs in circuit via an ICSP-cable. (As long the circuit was designed for this purpose.)

6.1.3 Brenner8mini

The Brenner8mini is reduced to ICSP-functionality only; it has no on-board test socked. PICs can be connected and programmed via ICSP-adapters only.

Like the “classic” Brenner8 the Brenner8mini was not designed to program PIC in circuit via ICSP-cable. Its outputs are not powerful enough to drive other electric parts beside the PIC (what happens often during in circuit programming). Only capacitors up to 100µF (combined) are allowed in parallel to Vss and Vdd pins.

6.1.4 Brenner8mini-P

The Brenner8mini-P is a Brenner8mini with powerful Vdd- output for ICSP. Thus it can program PICs in circuit via an ICSP-cable. (As long the circuit was designed for this purpose.)

6.2 Revisions

The development of the Brenner8 goes on continuously to improve its capabilities. Every hardware modification gets its own revision-number. New revisions sometimes need newer firmware versions to use its new capabilities. But old hardware will always run with the new firmware too.

6.2.1 Revision 0 (2006)

That was the 1st version of the Brenner8. It had still some serious problems. The 8/14-Pin-PICs could not be programmed in the on board test socket, the Vpp voltage was sometimes too low, the zener diode voltage had large tolerances and the Vpp voltage level measurement had a too limited range. A Rev. 0 can be upgraded easily to the Rev. 1 standard. I suggest to du this!

6.2.2 Revision 1 (2006)

With the design of the Rev 1 the substantial problems of the Rev 0 where solved.

- The replacement of D2 with a shottky diode made it possible to program 8/14-Pin-PICs in the test socket.
- The quality of the zener voltage was improved by change of R16.
- The Vpp measuring range was extended by change of R5.
- By the replacement of D1 with a shottky diode the Vpp generation becomes more efficient.

The Brenner8-P, Brenner8mini and Brenner8mini-P were “born” at this time. They got immediately revision number 1.
The revision 1 needs firmware 0.5 or later. The two P-versions need at least the firmware V 0.5a.

6.2.3 Revision 2 (2006)

With the introduction of the Bootloader, I introduced revision 2 of the Brenner8. The only change of hardware is a Jumper to activate the Bootloader. Since this activation can take place normally also via software (and in an emergency a piece of wire), a change of Rev. 1 on Rev. 2 not really necessarily. A revision 2 for the Brenner8mini and Brenner8mini-P don’t exist.
The revision 2 needs firmware 0.6 and Bootloader 1.

6.2.4 Revision 3 (2006)

This revision is to give an adequate ICSP support to the Brenner8P. An additional reset transistor (Q8) and a MCLR pull up diode (D4) was necessary. The normal Brenner8 gets only the diode D4.
With the reset transistor PICs can be burned also in circuits with a low impedance (few kiloohms) pull up resistor at the MCLR pin. The diode D4 makes it possible, to run a PIC in a test board or in an application circuit board.
Clearly the software development is easier, since the Brenner8 does not have to be separated for the test of the software no more from the test board. For burning PICs in the test socket or via a simple ICSP adapter the revision 3 does not makes any changes. The revision 3 needs the firmware V 0.8 or later.

6.2.5 Revision 4 (2007)

Some PIC-types with large flash-memory (e.g. PIC18F4682) could not be erased in some situations. The root of the problem was a short but unusual-high Vdd-power consumption (out of specs!) of these PICs during bulk erase.
To solve this problem, all unbuffered Vdd-outputs of all Brenner8-versions get additional buffer capacitors.

The Brenner8 gets 2 capacitors, while the Brenner8P and the Brenner8mini get 1 capacitor. No changes are necessary for the Brenner8miniP.

All Brenner8-versions of the revisions 2 and 3 can be updated easily. I suggest the use of ceramic capacitors of 100nF…470nF. Thy can be mounted easily at the PCB-side of the Brenner
The Brenner8 and the Brenner8P get one capacitor between the pins 31 and 32 of the 40-pin test socket. Brenner8 and Brenner8mini get one capacitor between pins 2 and 3 of the ICSP connector. The following picture shows a modified Brenner8.

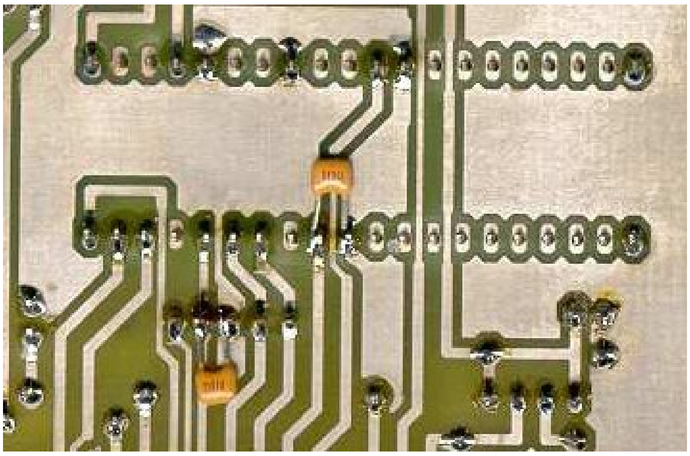


Figure 1 PCB-side of a modified Brenner8

6.2.6 Revision 5 (2008)

For improved reliability Brenner8P/Brenner8miniP gets the additional capacitor between pin2 and pin 3 of the ICSP-connector.

Other changes are necessary, if Brenner8P/Brenner8miniP-Silviu should be converted into a Brenner9 by the use of an adapter.

The Brenner8P rev4 (and Brenner8miniP-Silviu) has a rest-transistor Q8. Its original type was BC338. This type has to be replaced by a high-frequency transistor with small base-collector-capacity. (e.g. BF595 or BF199)
Please pay attention to the fact, that most HF-transistors have a different pinout (B-E-C) than the BC338 (E-B-C) !

The Brenner8P rev4 has a resistor R19 of 330 Ohm resistivity. (is Brenner8miniP-Silviu this resistor is labeled R13). This resistor has to be replaced by a resistor of 100 Ohm.

The 10nF-capacitors C6 and C7 of the Brenner8P have to be replaced by 1nF-capacitors. (In Brenner8miniP-Silviu this is C6 only.)

6.3 18FxxKxx-Adapter for Brenner8P

The programming-adapter (see the attachment of this document) makes it possible to program PIC18FxxKxx-types with the Brenner8P. These PICs are limited to 4V at all pins. The adapter works as voltage level shifter.

The adapter has 3 jumpers. It is not allowed to close all 3 jumpers at the same time! To program PIC18FxxKxx the jumpers JP1 and JP3 have to be closed.

The 5-pin male connector of the adapter is plugged into the female ICSP-connector of the Brenner8P (pin1 into pin5 and so on, look at the schematic)
The female connector of the adapter is now the new ICSP-connector to connect PIC18FxxKxx-PICs.

Inadvertently I programmed several PIC18FxxKxx in Brenner8P without the adapter. It worked, and I have not destroyed any PIC. However, this is risky and beyond the specs of the PICs. I suggest using the adapter.

6.4 Variants of Brenner9

While the Brenner8 is good for 5V-PICs the Brenner9 was designed for 3.3V-PICs.

The 3.3V-PICs don't need a high programming voltage (Vpp). Thus the design of the Brenner9 is much simpler then the Brenner8.
Three versions of the Brenner9 exist:

- Brenner8P with Adapter
- Brenner9N
- Brenner9L

The Brenner9N and Brenner8P+Adapter are using the same firmware and bootloader. The Brenner9L needs modified firmware and bootloader.

6.4.1 Brenner8P+Adapter

It can be used the Brenner8P or Brenner8miniP-Silviu Revision 5.

A special adapter - connected to the ICSP-connector of the Brenner8P - works as level shifter from 5V to 3,3V. Thus the Brenner8-hardware can be used to program 3.3V-PICs. The adapter is describes on the Brenner0-homepage. It is not identical to the 18FxxKxx-Adapter.

Into the Brenner8 has to be loaded the Brenner9-firmware. A combined Brenner8&9-firmware don't exist. It would not fit into the control-PICs memory.

6.4.2 Brenner9N

The Brenner9N is nothing else then a Brenner8P without Vpp-boost-converter but with integrated 5V-3.3V-adapter.

6.4.3 Brenner9L

The Brenner9L employs a low-power PIC18LF2450 as control-PIC. It works with an operating voltage of 3.3V. Thus the PIC can directly generate the 3.3V-signals, and a adapter is not necessary.
A solid-state-voltage-regulator generates the 3.3V for the PIC18LF2550.

The reduced voltage results in a reduced maximum clock speed of the PIC. Consequently a modified firmware and a modified bootloader have to be used inside the Brenner9L.

6.5 Printed board

Although it is surely possible to develop the Brenner8 on a hole-raster plate I recommend the photo-chemical production of a printed circuit board nevertheless. The layout made available by me is relatively simple. It doesn’t has very small structures, and only a one-sided board is needed.

The layout is for instance 75mm x 100mm largely. Board material of this size is commercial. Thus annoying cutting of the printed circuit board material is not necessary.

The layout of the Brenner8mini-P is also single-sided, but somewhat more difficult. The board measure is 83mm x 43mm.

On the Brenner8-Hompage also a layout suggestion for a Brenner8 on a hole-raster plate is available.

6.6 Component parts

After etching and drilling the board, it is equipped as usual. One starts it with the wires, follows the flat construction units (resistances, diodes), then the 28-pin-IC-socket, and in the end all "bulky" parts.

The values of the resistors and capacitors are not uncritical. The values may deviate by 25% from the values indicated in the circuit diagram. The Vpp-voltage-divider-resistors R4 & R5 are an exception. In order to simplify the later calibration of the programmer there values should not be deviated from the values of the circuit diagram.

When soldering the test socket, the lever of the socket should be in "openly" position, in order to avoid a misjudged sockets contact.

If other transistor types are used, one has to pay attention to the pin allocation of these types. All original types have the connection sequence E-B-C except the BF959.

As diodes all shottky diodes that can handle 30V and 100mA are suitable. I recommend the BAT43.

6.7 Burn the bootloader and firmware

The Brenner8 is controlled by a PIC18F2550 microcontroller. The software that runs in this chip is called the firmware of the Brenner8

Before the Brenner8 can be used, the bootloader and the firmware have to be programmed into the control-PIC. Therefore a PIC-Programmer is needed. Any programmer that supports the PIC18F2550 can be used, e.g. the Brenner5 or a different Brenner8.

This programmer is used to program the bootloader into the control-PIC. Then the PIC is plugged into the Brenner8/9, and the Brenner is connected to the PC. After the drivers are installed, the firmware is uploaded into the Brenner8/9 by use of USBurn (under Windows) or usburn (under Linux).

6.8 Clock source

The burner can be operated with a ceramic resonator as well as with a quartz crystal. If a quartz crystal is used, then two load capacitors are to be used for the quartz (C2 & C3). But if a ceramic resonator is used, then the two capacitors are unnecessary. The resonator is mounted on the board at the location of the two capacitors. In the Brenner8 the middle pin of the resonator fits into the additional ground connection between the capacitors locations. In the Brenner8mini the resonator is mounted into the two solder pad for C2 and lower solder pad for C3.

As frequency for the resonator/quartz 20 MHz are intended. Unfortunately 20MHz-resonators are not easy to procure. The employment of other types is possible, if the following is considered:

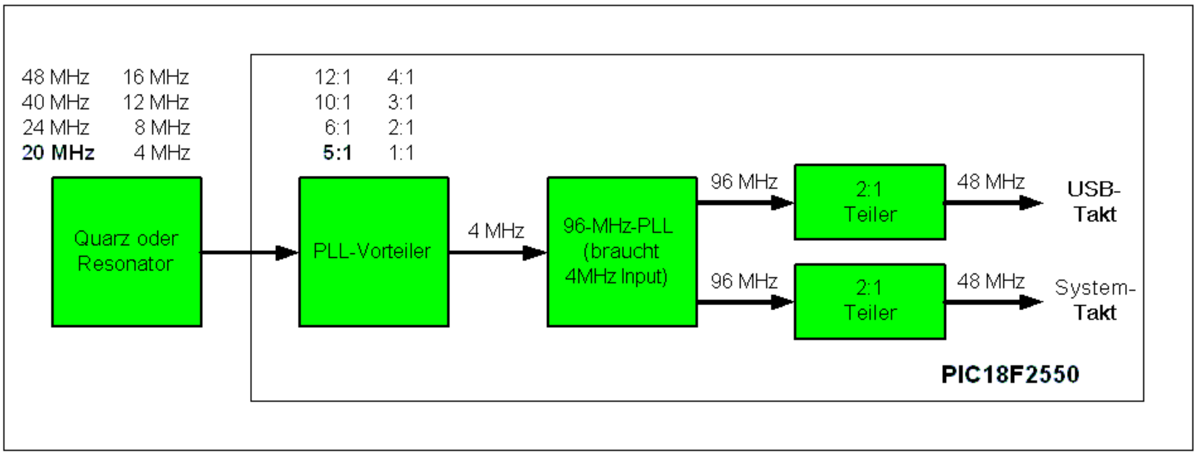


Figure 2 Clock generation inside the controlling PIC

By default the quartz crystal frequency is divided inside the PIC first by a 5:1 frequency divider (prescaler / PLL-Vorteiler) to 4 MHz. These 4 MHz are used to generate 96 MHz with a PLL circuitry. This 96 MHz clock is the base for the USB clock (2:1 division) and the PIC clock (2:1 division).

20 MHz is not the one and only possible input frequency.
The prescaler divider ratio can be changed from 5:1 to 12:1, 10:1, 6:1, 4:1, 3:2, 2:1 and 1:1. Thus a resonator/quartz with 48 MHz, 40 MHz, 24 MHz, 16 MHz, 12 MHz, 8

MHz and 4 MHz can be used too. One must change only the prescaler setup in the configuration of the controller PIC.

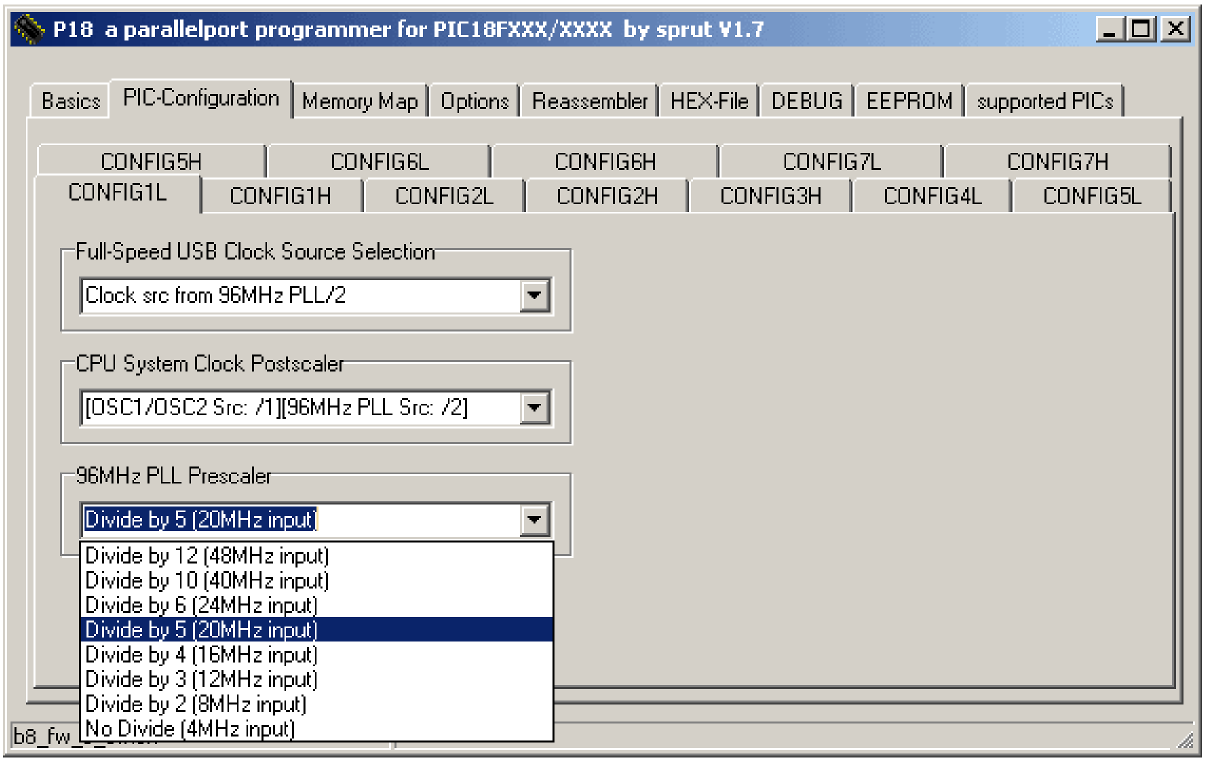


Figure 3 Resonator/Crystal-Selection for the controlling PIC

The prescaler ratio is defined in the PIC configuration, which is defined in the bootloader-HEX-file. With suitable programming software, like P18 or US-Burn, this configuration can be manipulated before the bootloader is burned into the control PIC.

After the HEX file was loaded into the programming software the option **Config from HEX-File** has to be deactivated on the **Basics** tab sheet.
Now the tab sheet "**PIC Configuration**" has to be selected and there the sub tab sheet "**CONFIG1L**". Here now one can find all clock setups of the PIC. The option "**96 MHz PLL Prescaler**" is the important one, which is by default **Divide by 5 (20MHz input)**.
This option can be changed to the desired resonator/quartz frequency. E.g. the setting **Divide by 2 (8MHz input)** permits e.g. the use of an 8 MHz ceramic(s) resonator.

6.8.1 Clock for the Brenner9L

The Brenner9L uses internally a system clock of 16 MHz (instead of 48 MHz). Consequently the frequency divider (to divide the 96 MHz-PLL-clock) is set to a divider ratio of 6:1 (instead of 2:1) (CPU System Clock Postscaler).

For the Brenner9 are modified bootloader and firmware available.

6.9 Test of operation

After the assembly of the burner and installation of the control PICs the burner has to be attached to a running PC. The green LED lights up immediately, followed by the yellow LED. In the same order both LEDs go off after 0.5 seconds. At the cathode of the diode D1 one can measure a voltage between 10 V and 20 V. If the blinking of the LED is missing and is the voltage at the cathode of D1 is below 6V, then the firmware was not correctly burned into the control PIC or the oscillator don't works.

If the USB driver is not yet installed, then Windows announces itself, and requires the installation of the driver, which is described in another chapter. If Windows reports that an USB device did not announce itself correctly, and therefore can not be used, then e.g. a quartz crystal with wrong frequency could have been installed.

The software US Burn offers the possibility, to test the operating voltage Vdd, the programming voltage Vpp as well as the data and the clock line individually by switching them on and off (**Options – Hardware).** For better orientation thereby the green LED as well as Vdd and the yellow LED as well as Vpp are switched. Depending upon selected IC socket (**Basic**) the voltages are switched to other pins of the 40-pin of IC socket:

Table 3 Signals at the test socked

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

With selected **8-/14-Pin**-socket, a voltage of less then 0,25V remains at the pin 40. At the 5-pin ICSP connector the voltages are in accordance with my standard, if the IC socket **18 Pins / ICSP** was selected:

Table 4 Signals at the ICSP-connector

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

Vdd should reach a level of at least 4.5 V, SDATA and SCLK should be over 4V. These levels depend on the voltage in the USB bus. To prevent an undervoltage situation the Brenner8 should not be attached to a passive USB hub.

The level of Vpp lies between 10V and 20V. It is adjusted by a calibration to a certain desired value. This calibration is described in a later chapter, and should be absolutely accomplished before the first target PIC is connected to the programmer.

Before the calibration is finished, no PIC may be attached to the test socket or to the ICSP connector. It could be damaged by overvoltage.

6.10 ICSP-Adapter

All 14-bit- and 16-bit-core PICs (PIC12F6xx, PIC16Fxxx and PIC18Fxxxx) with DIL-housing can be programmed in the on-board test socket (except PIC18F2331 and PIC18F2431). 8-/14-/18-/28- and 40-Pin DIL-types have to be plugged into the socket that pin 1 of the PIC sits inside pin 1 of the socket.

PICs in other housings (and in future dsPIC30Fxxx- and PIC10Fxxx-types as well) have to be connected by an adapter to the ICSP-connector.
The Brenner8mini has no on-board test socket. All PIC types have here to be hooked up to the ICSP connector.

During programming a target-PIC is connected to the programmer by only 5 signal lines (wires) even if all Pins of the PIC are in a socket. This is because the PIC is programmed using a serial communication - In Circuit Serial Programming (ICSP).

ICSP needs the following connections:

1. about +12V-programming voltage (Vpp)
2. +5V- operational voltage (Vdd)
3. Ground/return (Vss)
4. data line (Data / PGD)
5. clock line (Clock / PGC)

These 5 signals are to be connected to the following pins of the target PIC:

Table 5 The ICSP-connector

Nr.	signal description	signal name	pin name
1	+12V-programming voltage	Vpp	MCLR/Vpp (reset)
2	+5V-supply voltage	Vdd	Vdd
3	Ground line	Vss	Vss
4	Data line	Data	PGD (mostly RB7)
5	Clock line	Clk	PGC (mostly RB6)

PICs in large housings have multiple pins for Vdd and Vss. The producer suggests using all pins during programming. By tests I found out, that sometimes a single Vdd and a single Vss pin is enough. Other types need Vdd and Vss at all Vdd and Vss pins. Some types need Vdd and Vss at the AVdd and AVss pins as well.
If the on board test socked is used, the programmer connects the right pins. If an ICSP-adapter is in use, the correct connection is described in the handbook for the universal ICSP-adapter (http://www.sprut.de/electronic/pic/icsp/icsp.htm#universell).

All 5 ICSP-signals are available at the ICSP-connector to hook up a target PIC. An example for the use of the ICSP connector is an ICSP-adapter for PICs. Such an adapter is an additional test socket, which can be connected to the programmer via the ICSP connector.

6.10.1 Basic rules for ICSP-Adapters

Isn’t it a waste of time to talk about a simple cable? NO IT IS NOT!

CLK-Shielding
In the allocation of the cable there is a weak point. The clocking line (CLK, PGC) is very sensitive to disturbing nearby signals (cross coupling) and must be shielded from the other lines. In addition now no complete shielding is necessary, but a separate ground wire between CLK and DATA is at least necessary. For this reason I always use ribbon cable with 6 wires. Two of them are with ground wires: one between Vdd and DATA and another one between DATA and CLK. That is to be seen clear in the following circuit diagrams.

Naturally there are also different solutions, in order to avoid a cross coupling on the CLK line, e.g. one can draw the CLK line separately from the remaining cable. There I do not want to make further regulations as long CLK is protected against cross coupling.

If this simple rule is ignored, even the Autodetect function of the burner often fails. An error free programming of a target PIC becomes impossible.

How long can be an ICSP-cable?
It should be as long as necessary and as short as possible. Who wants to adapt only an additional socket, gets along with 10 cm. For real ICSP (to burn a PIC in the application circuit) 20 cm should also be sufficient. If CLK is properly protected, 1/2 meter should not be a problem. To long cables and cables without CLK shielding are very often the cause for burning problems.

In follow some examples of ICSP adapters.

6.10.2 ICSP-Adapter for PIC12F6xx

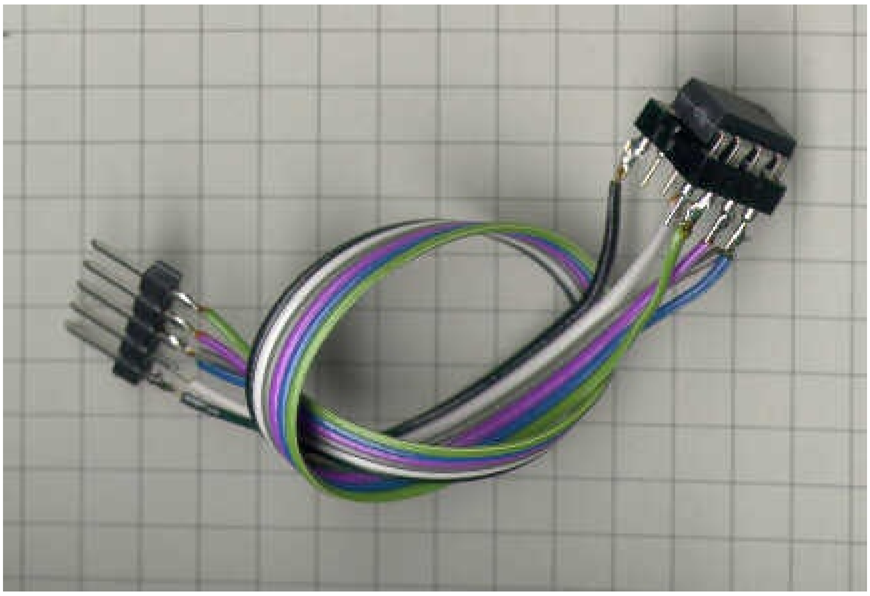


Figure 4 ICSP-Adapter for PIC12F6xx

The above photo shows an ICSP adapter for PIC12F6xx as it is needed for the Brenner8mini (the Brenner8 can burn the PIC12F6xx in the on board test socket.) It consists only of a 5-pin header (which is to be plugged into the ICSP connector of the burner) an 8-pin DIL socket for the PIC and a 6-wire ribbon cable. The following illustration shows the circuit diagram:

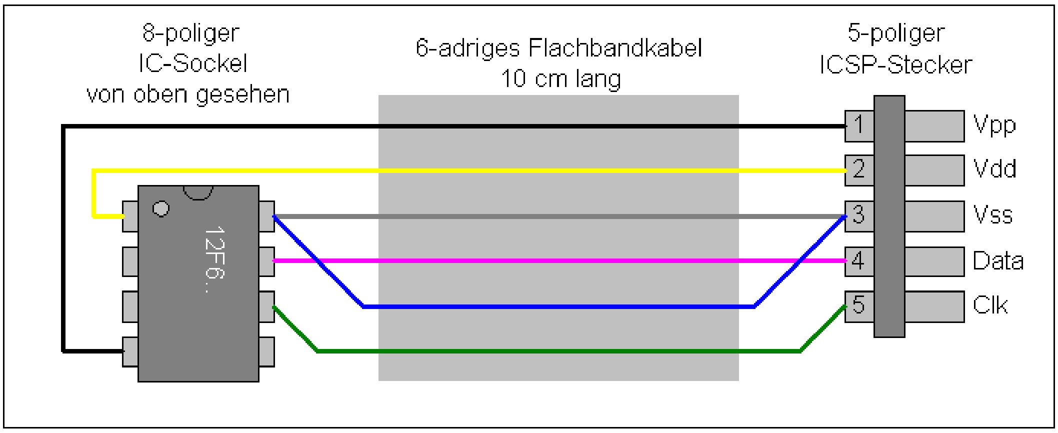


Figure 5 ICSP-Adapter for PIC12F6xx – circuit diagram

A 6-wire ribbon cable is used, because an additional ground wire is needed to shield the clocking line (Clk, PGC) from the other lines. That is not luxury, but for stable work of the burner necessary.

This adapter is NOT suitable for PIC10F2xx in 8-pin DIL housing, since these have a different pin allocation.

6.10.3 ICSP-Adapter for PIC in PLCC-Housing

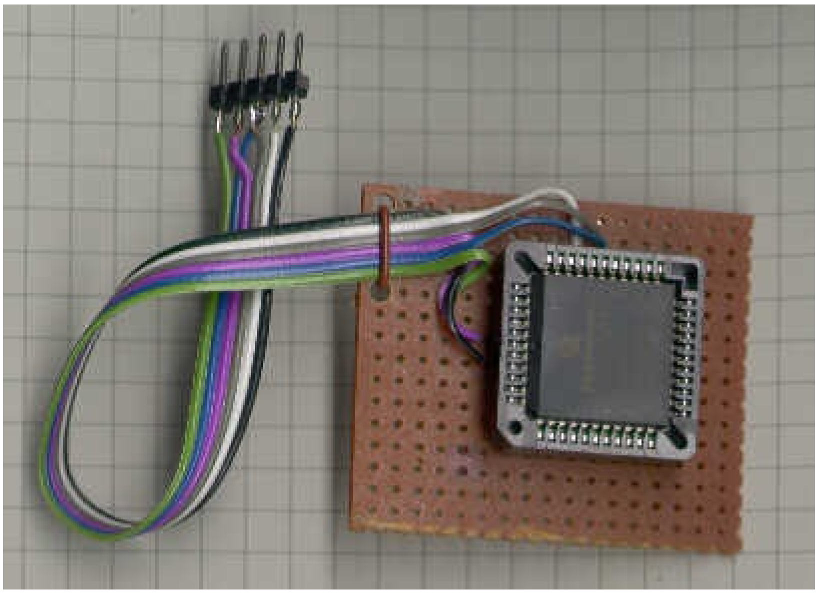


Figure 6 Adapter for PLCC-44

The above photo shows an ICSP adapter for PICs in PLCC-44-housing (a PIC16F87xA is shown).
The adapter consists only of a 5-pin header (which is to be plugged into the ICSP connector of the burner), a hole raster circuit board, a PLCC-44-socket for the PIC and a 6-wire ribbon cable. The following illustration shows the circuit diagram:

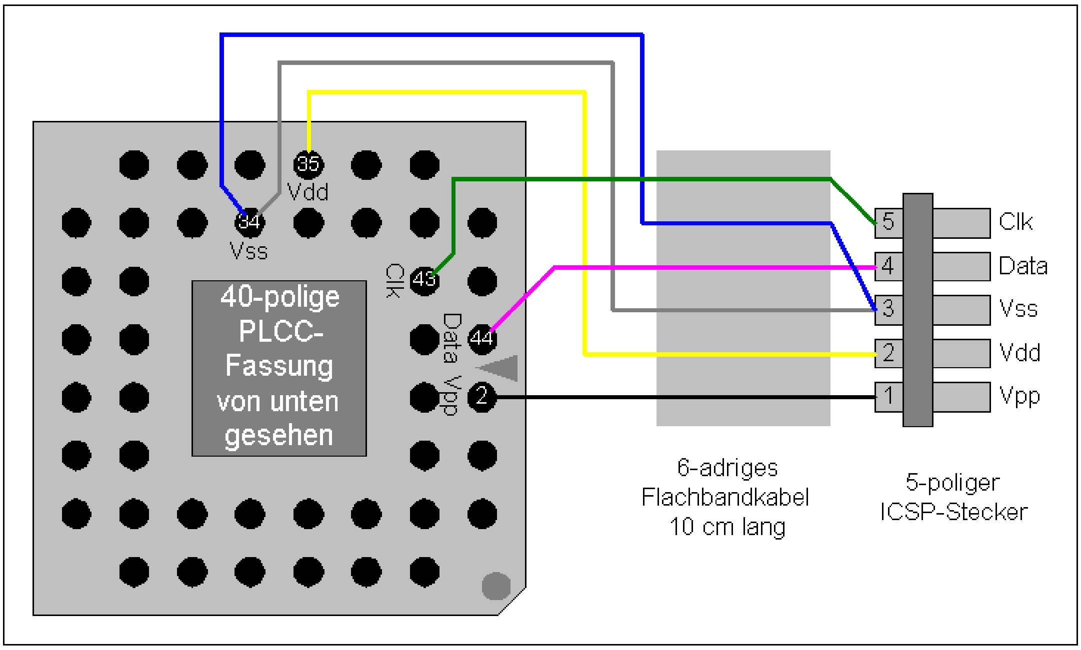


Figure 7 Adapter for PLCC-44 - circuit diagram

All statements I the made for the PIC12F6xx-adapter apply also here.

Microchip recommends to attach generally all Vss pins and all Vdd pins. For some PICs it is necessary, but not for all. Thus this simplified adapter doesn't works with all 44-pin-PICs.
I suggest to follow the Microchip recommendation for all future adapters.

6.10.4 Universal Programming Adapter for DIL-PICs

This adapter can be used to program any PIC-controller (if supported by software) in DIL housing. Before the adapter can be used for a specific PIC-type some wires (5...7) have to be installed (plugged) in the adapter. This has to be done very carefully to prevent damage to the target PIC. Details can be found in the handbook for the adapter.
For more information see http://www.sprut.de/electronic/pic/icsp/icsp.htm#universell

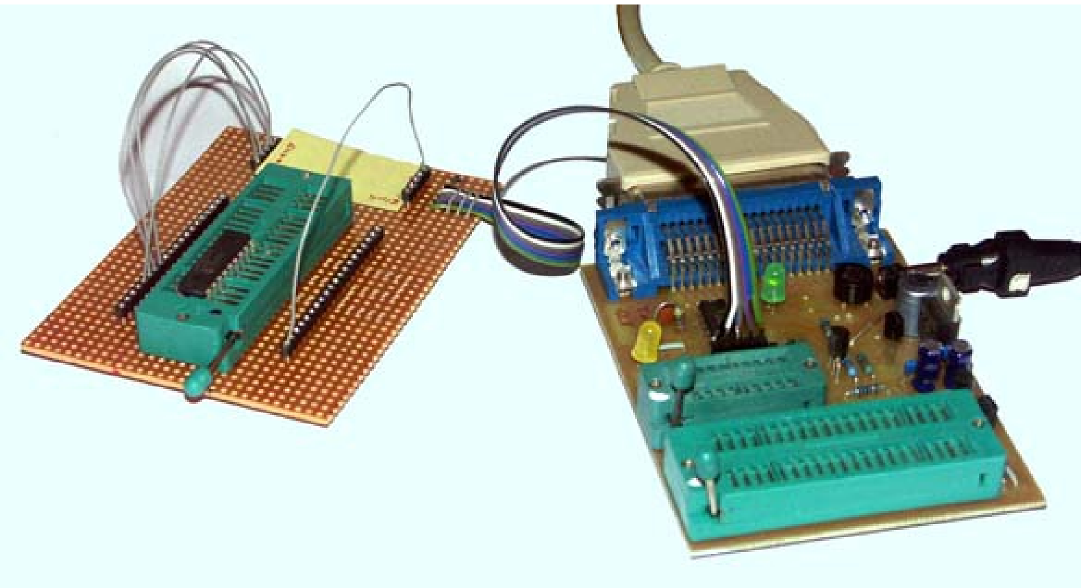


Figure 8 Universal Programming Adapter (here with a Brenner5)

6.11 ICSP- PIC-programming in the circuit

Only the Brenner8-P and the Brenner8mini-P are suitable to program a PIC inside the application circuit board.

A further application of the ICPS connection is to program a PIC, which is installed into its application circuit already.

Therefore the application circuit and the programmer have a 5-pin ICSP connector. Programmer and application circuit are connected by a cable and the target PIC is programmed „at home". There is no need to move the PIC between the circuit and the programmer anymore, and PICs in SMD-housings can only be programmed in this way.

Microchip uses on its test boards a Western-socket with 6 pins, which is marked with ICD or ICSP. The first 5 pins of this connection correspond to the 5-Pins of my ICSP connector; the 6th Pin of the Western socket is reserved.

6.11.1 Design of ICSP-ready circuits

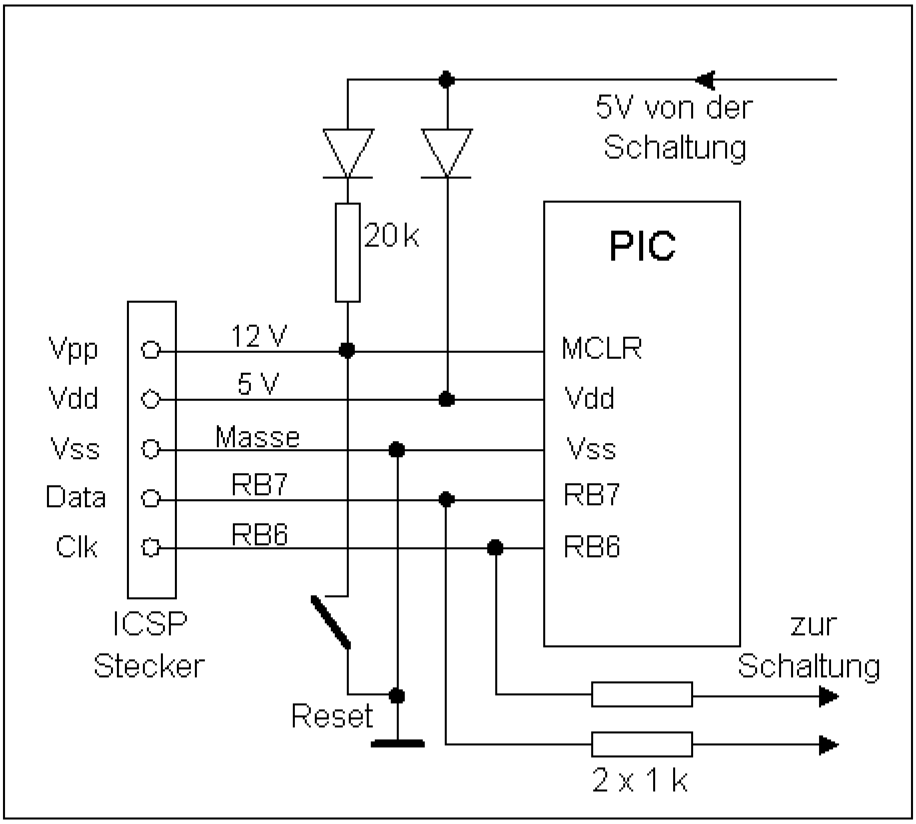


Figure 9 ICSP-ready circuit

The 5 PIC-ICSP-pins, which must be attached to programmer, do not exclusively reserved for programming use. They also are used in the application circuit for other functions.
In order to prevent interference between programming function and application function, some things are to be considered during the development of the application circuit:

6.11.1.1 Programming voltage MCLR/Vpp

This connection is the trickiest one.

Older Versions of Brenner8 (up to rev 2) pull this pin by a 10kOhm resistor down to ground (Vss) or via a switching transistor to Vpp (12V). (Newer Brenner8 versions pull it down by a switching transistor instead of a resistor.)
In the application circuit the same pin is usually connected by a pull-up-resistor to Vdd (5V) and can be grounded for a short time by a reset push button. It is clear, that this push button don't has to be used during the programming of the target PIC.

But the in circuit pull-up resistor is a problem too. This resistor is the only separation between the 5V of the user circuit and the 12V programming voltage from the programmer. A diode has to be used in line with the resistor to prevent overvoltage problems in the user application circuit.

If an older Brenner8 with pull down resistor is used, then the pull-up resistor in the application circuit has to be at leas 20 times larger in value. That means a 200 kOhm resistor has to be used as pull-up resistor.

Helpful is the use of a diode between the PICs Vdd-Pin and the +5V-rail of the application circuit.

For more modern Brenner8 versions (from rev. 3) with pull down transistors more usual pull-up resistor values (e.g. 10 kOhm) can be used.

6.11.1.2 Operational voltage Vdd

During programming the burner feeds the PIC with the necessary supply voltage (Vdd = 5V). As long the target PIC is the only load for Vdd on the circuit board the Vdd-output of the burner can be connected directly to the Vdd-pin of the target PIC. Before the burner is connected to the application circuit board the applications power supply has to be disconnected.

Are beside the target PIC other Vdd-loads in the application circuit board, then the Vdd-output of the programmer can be overloaded. The Brenner8 can generate only some milliampere, the Brenner8P delivers up to 100mA.

To separate programmer and target PIC from the rest of the application circuit board a shottky diode or a jumper can be used between Vdd-pin of the PIC and 5V-rail of the application.

The disadvantage of a shottky diode (compared to the jumper) is the reduction of the Vdd voltage level of the PIC by 0.2V. For the most applications this is not critical. But if e.g. the ADC of the PIC uses Vdd as reference voltage, a jumper may be the better solution.
During the programming process the jumper should be open.

PICs in large housings have multiple Vdd-pins. All this pins have to be connected. In application circuits this is anyway the normal situation.

6.11.1.3 Ground/return Vss

This is the only simple connection between burner and target PIC. Vss-pin of the programmer and Vss of the target are to be connected.

PICs in large housings have multiple Vss-pins. All this pins have to be connected. In application circuits this is anyway the normal situation.

6.11.1.4 Clock- and Data-connection (PGC und PGD)

PGD are PGC at the most PIC-types the port-pins RB6 and RB7.

If these pins are not really-really needed in the application circuit, one should exclusively use this pins for the ICSP-interface only.

But is the pins are needed in the application, they should be connected directly to the ICSP-connector but to the application via resistors of at least 1kOhm resistivity.
If such resistors are not acceptable for the application, then jumpers have to be used instead. For programming the jumpers have then to be removed, to insolate the application circuit from the PIC.

7 Driver installation

After the Brenner8 is assembled and the firmware is burned into the control PIC, it has to be installed in Windows.

Previous versions of the Brenner8 needed the *Microchip Custom Driver*. But this driver is troublesome on modern Windows versions. Because of this I switched over to the **libusb**-driver.

The **libusb**-driver can be downloaded from http://www.libusb.org and is included in the US-Burn software package as well. Because US-Burn is needed anyway, one should download the US-Burn-ZIP-file from my homepage …

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

… and un-zip the package into a folder on the local harddrive. The driver is then located in the subfolder **\driver**.

Users of a 32-bit-Windows launch there **installer_x86.exe**, Everybody else launch **installer_x64.exe** instead. This will install the libusb-driver.

In the next step we have to tell Windows, that it should use this driver for the Brenner8/9. To do this right-click on the file **sprut_device_libusb.inf** and choose **install** from the context-menu.

If now the Brenner8/9 becomes connected to the USB-connector of the PC, then Windows should automatically use the libusb-driver. This can be checked by the use of the Device-Manager. There the Brenner8/9 is now listed as “Sprut Device”.

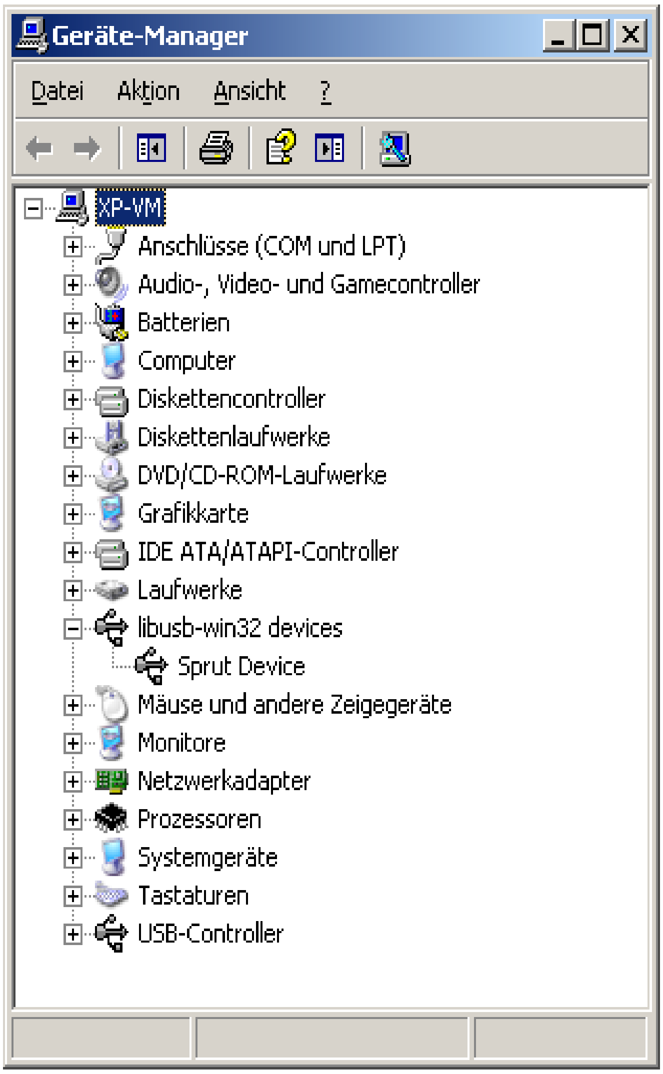


Figure 10: Device manager

Ready. US-Burn can now be used. But don't forget to calibrate the Brenner8 before you connect it to any target PIC.

8 Calibration

After the assembly of the Brenner8, the programming of the control PIC and the installation of the USB-driver the Vpp-generation of the Brenner8 has to be calibrated.

A non-calibrated Brenner8 can destroy every target-PIC in shortest time! It can produce Vpp-voltages of 25V and above. No PIC survives this!

A small boost-converter is used in the Brenner8 to generate the programming voltage Vpp. The voltage level can be controlled by software. Such every target-PIC gets its ideal programming voltage.
This can function only if the control PIC of the Brenner8 can measure the Vpp level precisely.

The programming voltage is measured by the control PIC via a voltage-divider and compared with a reference voltage (from a zener-diode). During calibration Z-voltage and voltage-divider-ratio are determined.

For the calibration is needed:

- Brenner8
- US-Burn-software
- voltmeter

8.1 Preparation

Connect the Brenner8 with the PC.
Start the US-Burn software on the PC.
In US-Burn select **Options** – **Hardware**.

The following adjustments are done in the „Box“ **programming voltage Vpp calibration** in 3 steps:

1. setting the Z-voltage
2. setting the voltage-divider-ratio
3. automatic adjustment of offset and gain

8.2 Step No. 1: Z-Voltage

The Brenner8 contains a 3,3V-Z-diode. Because typical Z-diodes have tolerances of up to 10%, the real Z-voltage has to be measured.
A voltmeter has to be connected between GND/Vss and Pin 5 of the control-PIC. The latest Brenner8-revisions have a test-point LSP2 for the Z-voltage determination.
The measured voltage is entered into the **Z-voltage** field of US-Burn. The both arrow-buttons can be used to adjust the value between 2V and 4V in 0,01V-steps.
To apply the modified Z-voltage setting, the **apply**–button has to be used.
(In the **BASIC**–page of US-Burn is now a calibration value shown, that should be close to 1.)

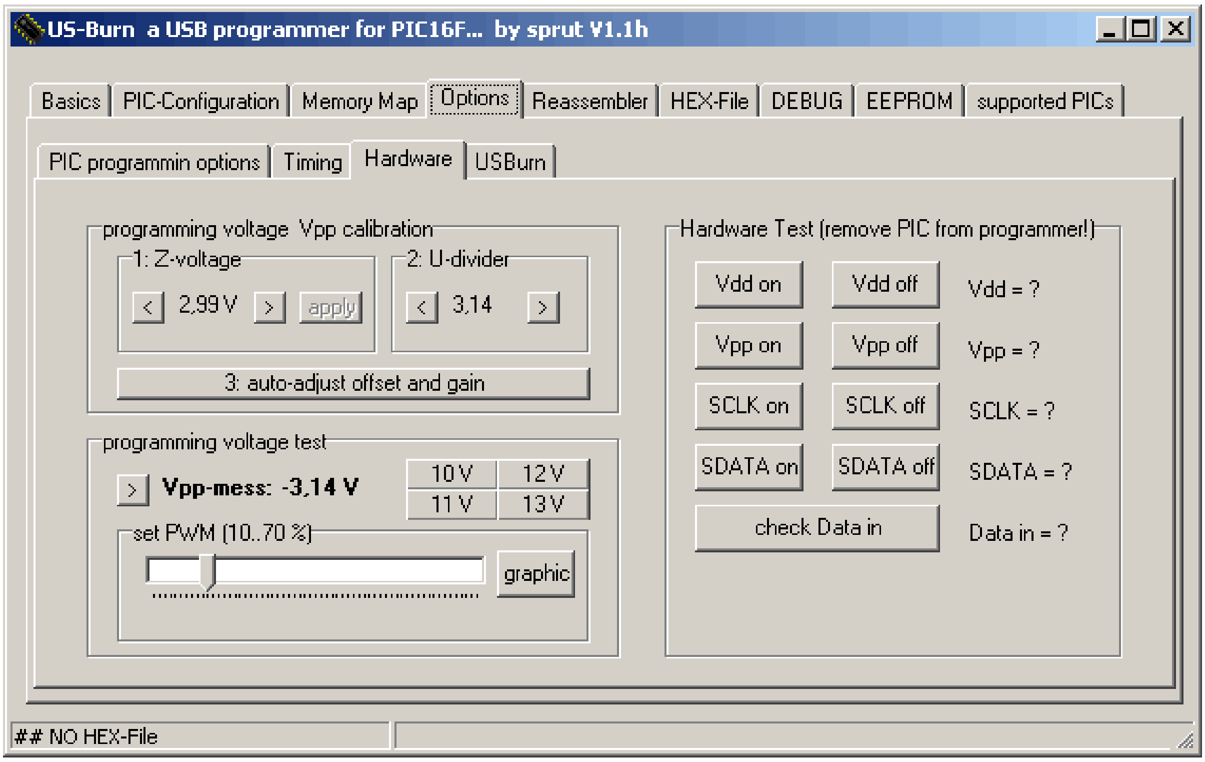


Figure 11 US-Burn - Options-Hardware

8.3 Step No. 2: Voltage divider

The Vpp-voltage-divider is made from the both resistors R4 and R5. There default divider ratio is 3.14. The real value may differ a bit and has to be determined now. The setting is done in the **U-divider**-field.

The voltmeter has to be connected between the cathode of D1 and Vss (alternative via C5). The latest Brenner8-revisions have a test-point LSP1 for the voltage determination.
The sliding control **set PWM** is used to adjust the voltage to about 13V (check with voltmeter). (The voltage should be high but has not to exceed 14V!)

The Brenner8 measures the voltage too, and indicates its result as **Vpp-mess**. By adjusting the **U-divider** -ratio (use the arrow-buttons) the Brenner8-measure-result is approximated to the voltmeter-result as close as possible.

8.4 Step No. 3: regulator set-up

The final step doesn't require a voltmeter anymore. Simply click on **auto-adjust offset and gain**. The Brenner8 starts an autocalibration process, which takes about 6 seconds to be completed.

During this autocalibration high Vpp-voltages are generated. Because of that no target-PIC has to be in an on-board test socked or connected to the ICSP-connector.

(On the **BASIC** –page US-Burn indicates some numbers with could be helpful in case of fault analysis. If during the calibration there was sprinted ## **Vpp-**

Regulation-Mode1 can not be used!!, then firmware 0.17 can not be used. Use firmware 0.16 instead.)

To test the calibration now the buttons **10V**, **11V**, **12V** and **13V** can be used to set Vpp to the desired values. The typical acceptable error is about 0.3V.

8.5 Ready

Calibration data is stored in the Brenner8. A backup copy of this data is stored in the **usburn.ini**.

8.6 Error diagnostic

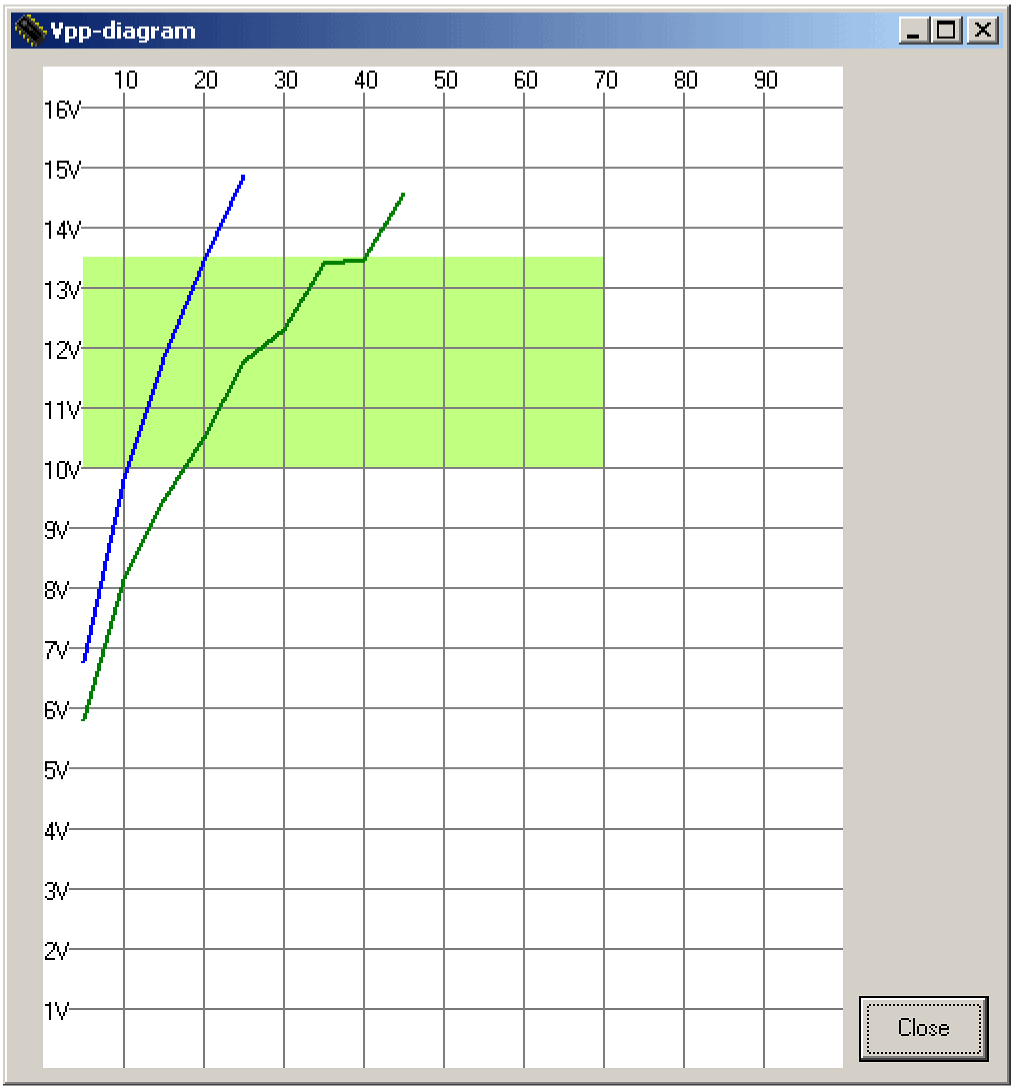


Figure 12 Vpp-diagram - normal

In the field **programming voltage test** is the button **graphic**. If one clicks on this button, then a drawing of the Vpp output voltage levels us generated (needs some seconds). The figure shows the voltages Vpp-on (green line) and Vpp-off (blue line) for different duty cycles of the boost converter.

During the drawing process high Vpp-voltages are generated. Because of that no target-PIC has to be in an on-board test socked or connected to the ICSP-connector.

Only voltages up to 15V can be measured by the Brenner8 and are shown in the diagram. The light green area is the normal working area of the boost converter. The both voltage-lines have to cross this area from bottom to top, like seen in the figure above.

The following diagram shows a bad boost converter. The voltages are to low. This is probably a hardware error, e.g. a wrong type of the diode D1.

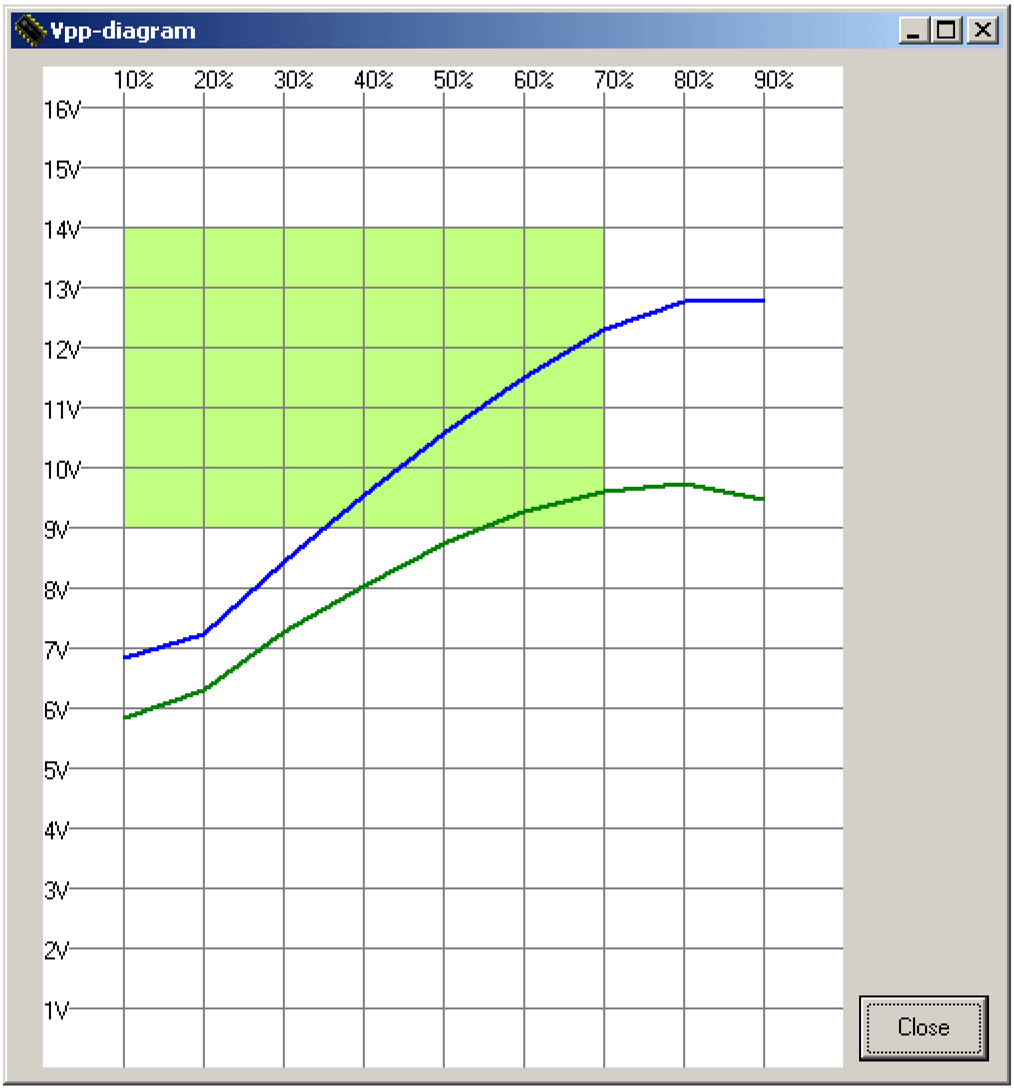


Figure 13 Vpp-diagram - undervoltage

9 Indicator LEDs

The Brenner8 has two LEDs: 1x green, 1x yellow. They indicate the operation mode of the Brenner8:

9.1 Power-up and programmer selection

The green LED lights up immediately after the Brenner8 is connected to the USB-port of the PC. The yellow LED follows 0.25 seconds later.
In same order the LEDs go off after they lit 0.5 second.

If USBurn is launched, then it will look for a programmer connected to the PC. If it detects a Brenner8 or Brenner9, then it will flash both LEDs 2 times. If multiple programmers are attached to the PC and the user selects a different programmer for USBurn, then both LEDs of the new selected programmer will flash 2 times.

9.2 The yellow LED

During normal operation the yellow LED lights up during excess to the target PIC.

9.3 The green LED

9.3.1 Mode1 (firmware 0.17 and later)

Firmware 0.17 supports a more accurate Vpp-stabilization, called mode1. If mode1 becomes active, then the green LED will light up. The LED brightness depends on the Vpp-voltage level. This is not an error-indication any more.

9.3.2 Malfunction (up to firmware 0.16)

If during normal operation the green LED lights up and stays on, then a malfunction of the programming-voltage-generation was detected. For safety reasons the Brenner8 has switched off the programming voltage totally.

The Brenner8 has to be disconnected from the PC and the reconnected to reset the firmware. The cause for the malfunction has to be removed by calibration of the Brenner8.

9.4 Bootloader

Starts the Bootloader, then booth LEDs are on continuously.

10 US-Burn

US-Burn is a Windows program that loads and analyses a HEX-file and transfers its data via USB to the Brenner8
In addition is offers the following features:

- Configuration of the target PIC
- HEX-file viewer
- Disassembler
- EEPROM-data-viewer
- Calibration of the Brenner8
- Upload of new firmware into the Brenner8

10.1 Requirements for use of US-Burn

10.1.1 Software

US-Burn runs under Win98/ME/NT/2000 and XP. But I test the software only under Win2000 and WinXP. I recommend WinNT/2000/XP.

10.1.2 Data

US-Burn accepts only the Intel-Hex8-format. The programming development (e.g. MPLAB) has to generate such HEX-files (MPLAB by default).

10.1.3 Hardware

US-Burn needs a USB-PIC programmer from the Brenner8-family (www.sprut.de/electronic/pic/projekte/brenner8/).

- Brenner8
- Brenner8P
- Brenner8mini
- Brenner8miniP

Brenner8(P): my standard USB programmer
http://www.sprut.de/electronic/pic/projekte/brenner8/index.htm

Brenner8mini(P) a simplified USB-programmer for ICSP
http://www.sprut.de/electronic/pic/projekte/brenner8mini/index.htm

10.2 Installation

US-Burn runs under Win98/ME/NT/2000 and XP.

The Program US-Burn can be downloaded as zip-file with the name usburnxx.ZIP. This archive contains the following files:

- usburnxx.exe the main program
- readme.txt short instruction

- usburn.hlp the help-file
- mpusbapi.dll DLL-file for USB (from Microchip)
- Picdef3.dll DLL-file for PIC-database

and in the subfolder \DB :

- picdef03.dat PIC-database file
- setdef03.dat PIC-database file
- fildef03.dat PIC-database file
- cfgdef03.dat PIC-database file
- cekdef03.dat PIC-database file
- texdef03.dat PIC-database file

and the folder:

- Driver with the USB-driver (from Microchip)

The 6 PIC- database files are together the **PIC-Database**.

Before US-Burn can be used all files and directories have to be copied from the ZIP-file into one directory. At the first use the software will create a file named **usburn.ini** in its directory. In this file it stores several parameters and settings. US-Burn don't writes anything into the registry

To use the Brenner8 the USB-driver (from the sub-directory **Driver**) has to be installed. This can be done by executing **mchpusb.inf** in this directory or is done automatically if the Brenner8 is connected to the PC for the first time. (see chapter before)

10.3 Quick start

- Un-zip everything from the ZIP-file into one directory
- Connect the Brenner8 to the PC
- Install the driver
- Start the program US-Burn
- Put target PIC into the testsocket
- select PIC-socket and PIC-family
- click ‚**Identify PIC in Programmer**‘
- load Hex-File in US-Burn: ‚**Select HEX-File as source**’
- change config-setting if necessary
- click ‚**Write HEX-File into PIC**‘
- finished

10.4 Programming of PICs - basics

PICs are burned via a serial data link (ICSP – In Circuit Serial Programming) that contains a clock line and one data line. In addition a 5V-supply line, a 12V (more or less) program voltage line and – of course – a ground line is needed.

The programmer contains a 5-pin ICSP-connector that carries all 5 signals. A program adapter (with test socket) or an ICSP-capable circuitry can be hooked up to this connector via cable.

The Brenner8 carries a 40-pin test socket for 14-bit-core (PIC16Fxxx) and 16-bit-core (PIC18Fxxxx) PICs in DIL housings. The PICs have to be plugged into the test socked that pin 1 of the PIC sits in Pin 1 of the socket.

PICs of the dsPIC30xxxx family (and in future 12-bit core-PICs) can not be programmed in this test socket! An ICSP-capable programming adapter has to be used for these types.

10.5 Use of the software

At the program start US-Burn opens the USB-driver, searches for and finds the Brenner8 and indicates the program window. If any problem is detected, an error message is indicated.

10.5.1 Missing USB-driver

The program needs the microchip USB-driver (in the ZIP file included). If the driver is missing, an error message is indicated.
It is not possible to use the software until the driver is installed.

10.5.2 Missing Brenner8

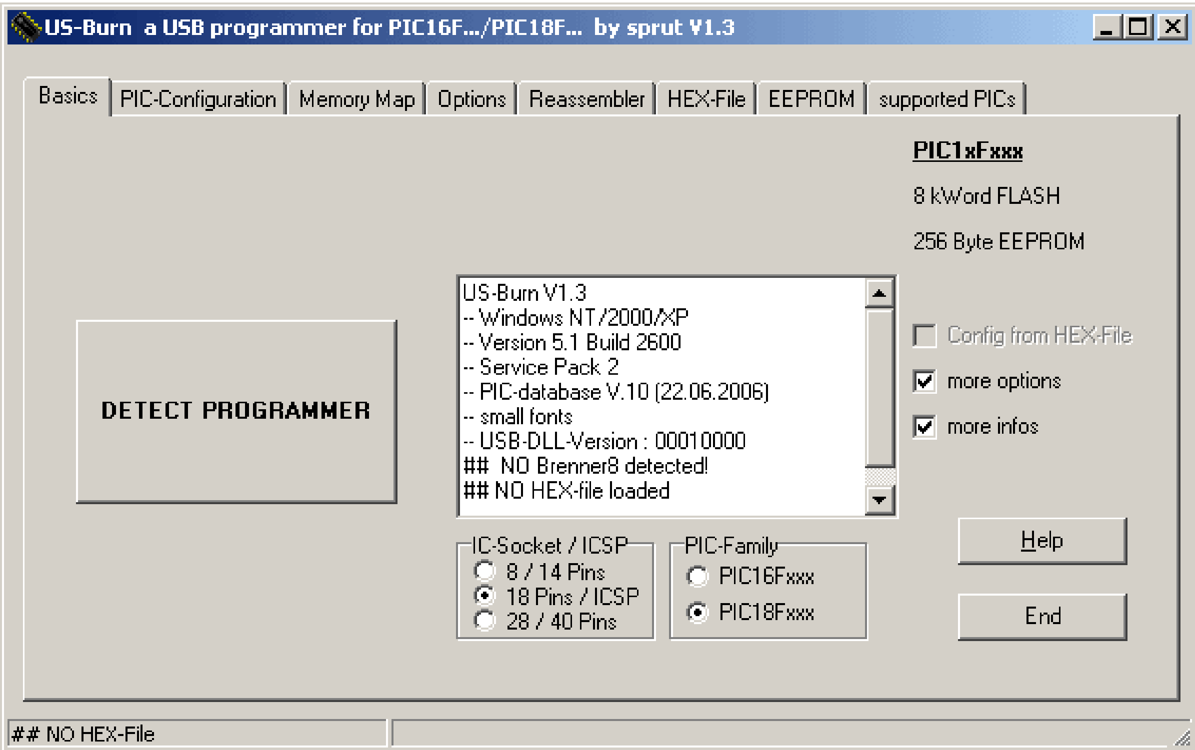


Figure 14 error message: missing programmer

At program start US-Burn looks for a Brenner8 programmer connected to the PC, because of this the programmer should be connected to the PC before this point of time. If no programmer is detected, this error is reported (see figure above).

After a click on auf “**DETECT PROGRAMMER**” the software tries again to detect a programmer. If the programmer is now detected (because e it was connected to the PC in the meantime) the normal program window is shown.

There is a simplified and a complete program window available. The simple window shows only the essential functions.

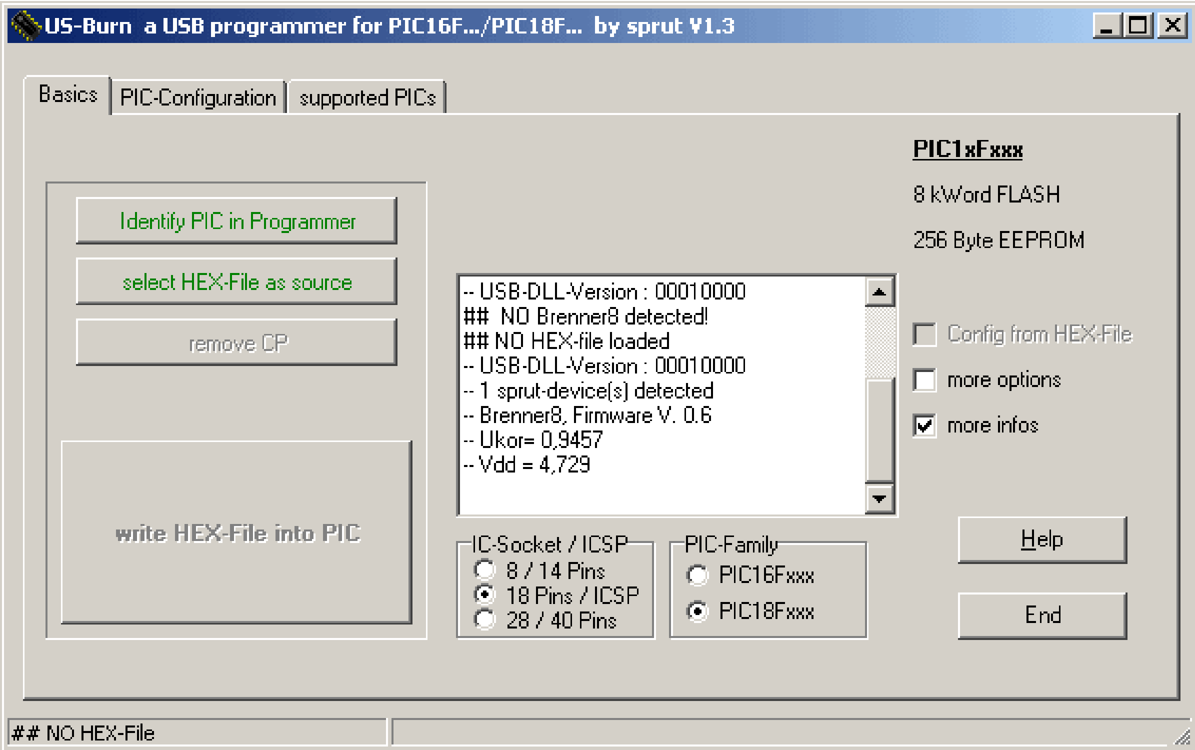


Figure 15 simplified program window

Advanced users can select **‚more options’** to see all functions of US-Burn.

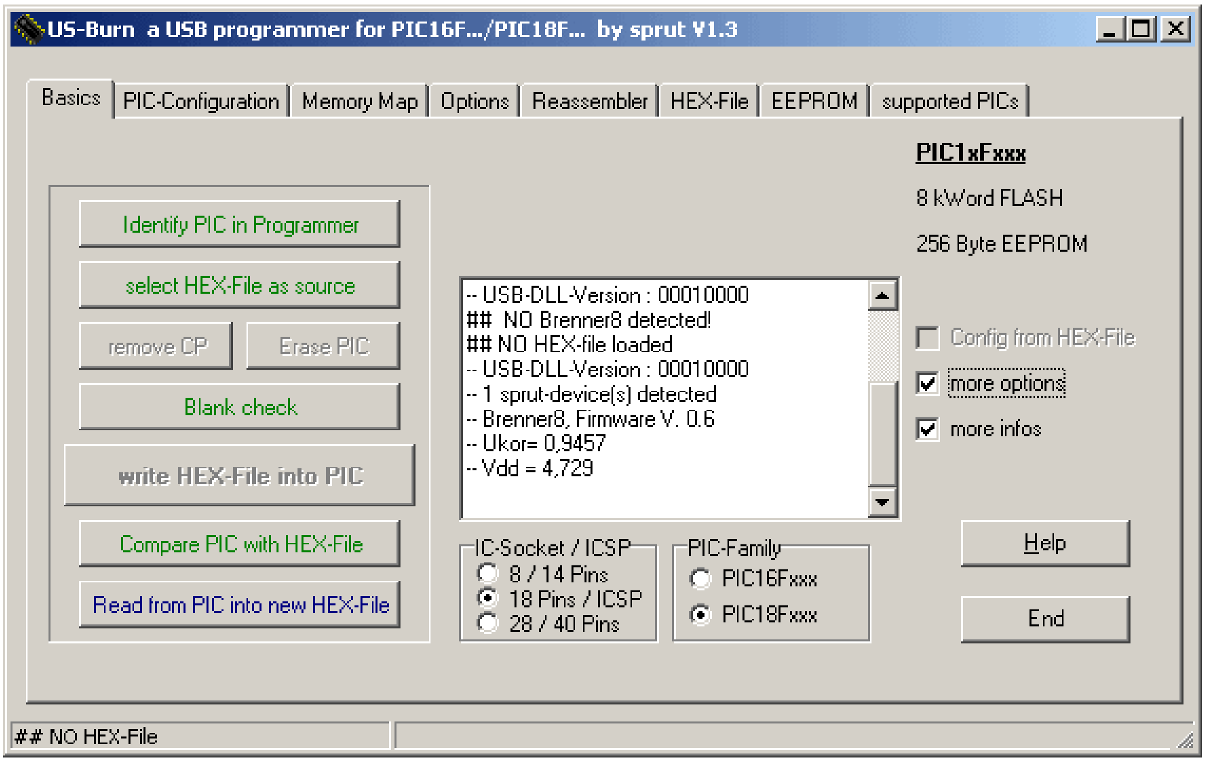


Figure 16 complete program window - no PIC detected till now

On the **Basics**-page is a log window to report more and less important things. All lines in this window that start with '##' are warnings or error messages, and should not be ignored.

The program is talkative. The user can suppress less important messages by deactivate the "**more infos**"-option.

The default background color of the log window is white. Successful and failed programming tries will change the color to light green or light red.

As long as the correct type of the target PIC is not determined the software will not allow any writing access to the target PIC. Thus some buttons on the program window are disabled.

10.5.3 Select PIC-Type

All information about programming strategy and PIC configuration settings are stored in 6 PIC-data-files (the PIC database) which are contained in the ZIP-file. If US-Burn can not find this data files, then it puts out an error message e.g.:
missing PIC-database: picdef03.dat
As long US-Burn can not detect the files, the software can not work.

Now it is high time to plug a target PIC into the on board test socket or to connect a target PIC to the Brenner8 via ICSP. After that the case size and the PIC-family has to be selected in the select boxes **IC-Socket/ICSP** and **PIC-Family**. If the target PIC is connected via ICSP, then **18-pin/ICSP** has to be selected independent of the real case size of the target PIC.
The family-selection is related to the internal design of the PIC and not so much to the real type-name. Because of this: if 14-bit PICs of the series PIC12F6xx are used,

then PIC16Fxx has to be selected as family.

With wrong setting the PIC can not be identified or programmed by the programmer. It may be possible (but not probable) that a PIC can be damaged with wrong settings.

US-Burn supports more then 170 PIC types. After a click on **Identify PIC in Programmer** the target PIC type will be determined. If the PIC type can not be identified (because programmer or PIC is defective) no writing or erasing functions can be used. This is for safety reasons.

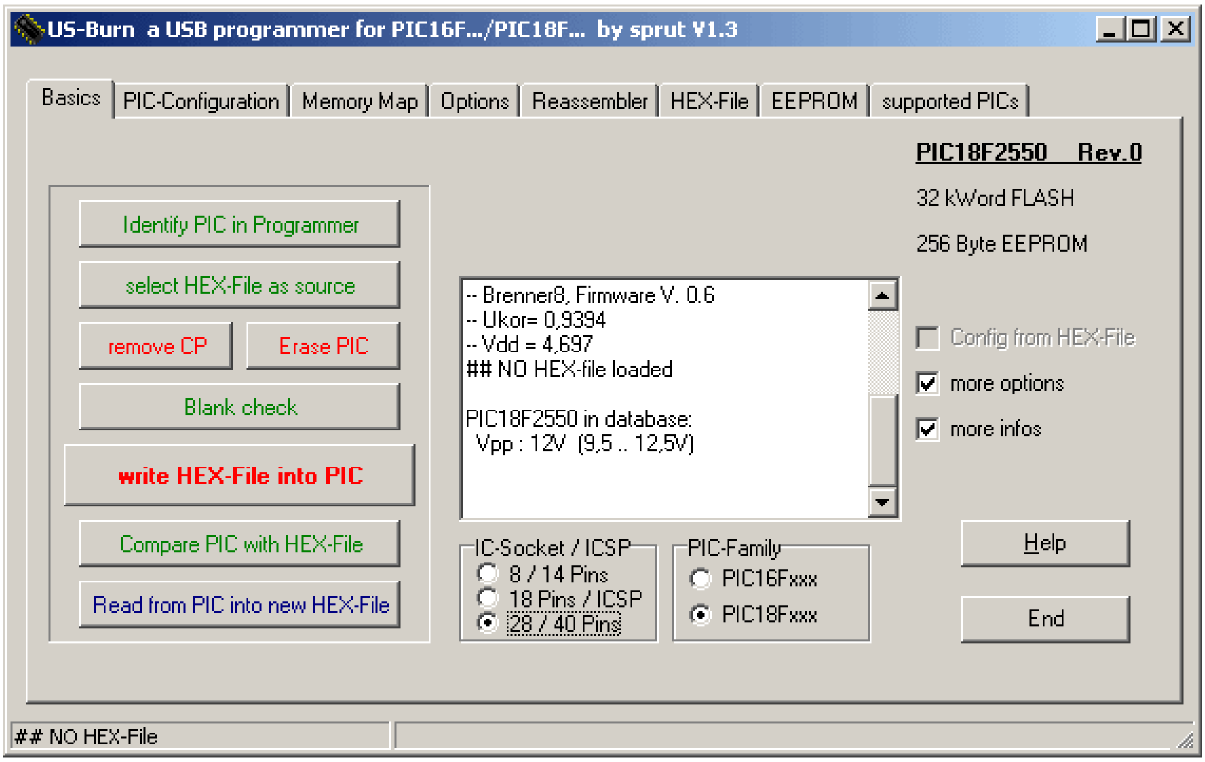


Figure 17 complete main program window – PIC type identified

The **Identify PIC in Programmer** – function detects an active copyprotection (codeprotected) of program memory (FLASH) or data memory (EEPROM) of the target PIC. **A copyprotected PIC can neither be programmed nor correctly read or compared**. **Compare** and **Read** –functions have to fail in this case. The **Read**-function read only zeros from a copyprotected memory area. Copyprotection can only be removed by the **remove CP** – function. This function will also erase the whole PIC memory.

If the PIC-type is detected, then the default programming voltage of the PIC will be indicated in the log-window. The Brenner8 will be adjusted to the default programming voltage automatically.

10.5.4 Load the Hex-File

US-Burn needs a HEX-file, which contains all the data that has to be programmed into the target PIC. The **Select HEX-File as source** button opens a dialog box where the HEX-file can be selected. US-Burn opens always the last used directory as default.

++ATENTION++
The hex-file often includes configuration-information for the PIC. If US-Burn detects such information, then it will read this information while the hex-file will be selected. If you don't like to use this information, then you can modify it manually after you have unchecked the **Config from HEX-File** option.

10.5.5 Configuration of the PIC

If the hex-file don't contains configuration-information or if one likes to modify the configuration-information, then this can be done manually. Therefore the checkbox Config **from HEX-File** has to be unchecked. Then all for this PIC-type possible configurations are available at the **PIC-Configuration** page and can be changed manually.
How this page looks like depends on the detected PIC-type. Before a PIC-type was detected, this page will be empty.
Gray shaded fields can not be changed.

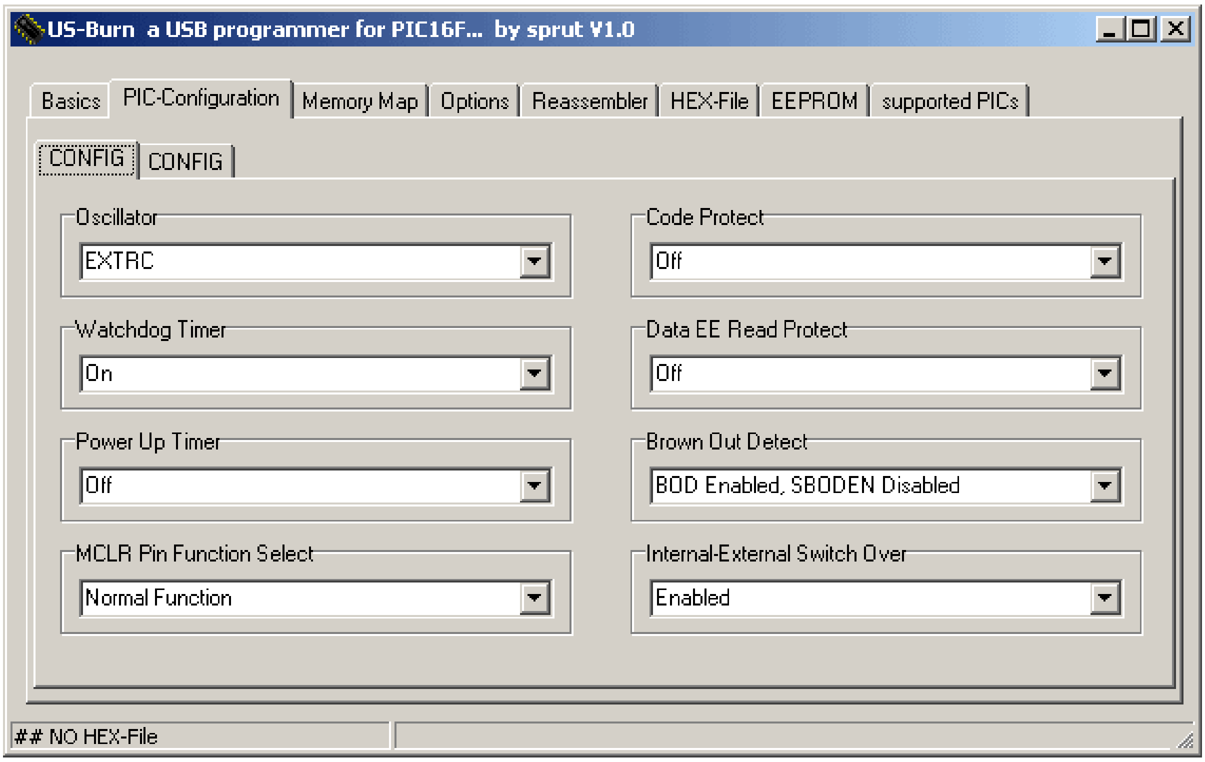


Figure 18 Configuration window (example)

A description of many configuration options can be found at www.sprut.de/electronic/pic/config/config.htm.
The original data sheet of the PIC contains the complete description of the configuration.

In one tries to program a PIC without any configuration information, then the PIC will mostly not work. The default settings of the configuration are normally not practically usefully. Often the external RC-oscillator and the watchdog-timer are activated.

10.5.6 ID-Information of the PIC

Inside many PICs can be stored an up to 8 digits long ID-information. This information has to be included in the hex-file.
The ID-information has no influence on the operation of the PIC, but it can e.g. be used for a serial number.

10.5.7 Burn the PIC

A click on the button **Write HEX-File into PIC** starts the burn process. Its progress will be indicated with gauges.

In no hex-file was selected, then US-Burn will now ask for one. If a hex file was selected earlier, then US-Burn will read this now again and then burn into the PIC. After the burning process is finished, the content of the PIC will be checked for correctness and the result indicated in the log window.

Before US-Burn starts to burn the PIC, it erases the whole PIC except this was forbidden (see Options).

10.5.8 Compare PIC and HEX-File

After a click on **Compare PIC with HEX-File** the content of the PIC will be read and compared with the selected hex-file. If the configuration was changed manually, then the changed configuration will be used (except a modified OSCCALL value).
The result of the compare process will be indicated in the log window.

PICs with activated codeprotection can not be compared (many errors would be indicated). PICs with a changed OSCALL will have virtually one error.

10.5.9 Erase PIC

After a click on **Erase PIC** the FLASH-program memory of the PIC will be erased. In many (but not all) PICs all other memory (EEPROM, ID, config) will be erased too.
After a click on **remove CP t**he whole PIC (all memory) will be erased.
US-Burn will erase memory anyway before it programs new data into a PIC. Thus the erase function is normally not necessary.

10.5.10 Blank Check of PIC

After a click on **Blank Check** the software checks if the PIC is blank (erased).
US-Burn will erase memory anyway before it programs new data into a PIC. Thus the erase function is normally not necessary.

The function may be useful, if used PICs are handed over to other persons, and you like to be sure not to give away your program code by accident.

10.5.11 Read out the PIC

After a click on **Read from PIC into new HEX-File** all PIC memory (FLASH-program-, EEPROM-data, config- and id-memory) will be read and written into a new hex-file.
(Reading a codeprotected PIC results in nonsense.)

To copy a PIC into a new PIC, the new saved hex-file has to be read again with the

select hex-File as source option. Then it can be burned into a new PIC.

10.5.12 Remove code protection

After a click on remove **CP** the codeprotection of a PIC will be removed **and all memory of the PIC will be erased**.
If a PIC is codeprotected or not can be tested by with the Identify **PIC in Programmer**-function.

10.6 Additional features

10.6.1 Graphic memory indication

US-Burn has a graphic memory display. It shows how much of the available memory would be occupied by the hex-file-data.

The following display-options can be selected::
1. content of the hex-file
2. content of the PIC
3. areas in the PIC that would have to be programmed
4. errors (differences between hex-file and real PIC-data)

The options 2 ... 4 show useful data only after the PIC was red or programmed.

The Flash-memory is visualized as up to 8 graphic blocks. Depending on the PIC type, one block represents a different amount of memory. If the mouse cursor moves over a block, then the address-area of this block is indicated.

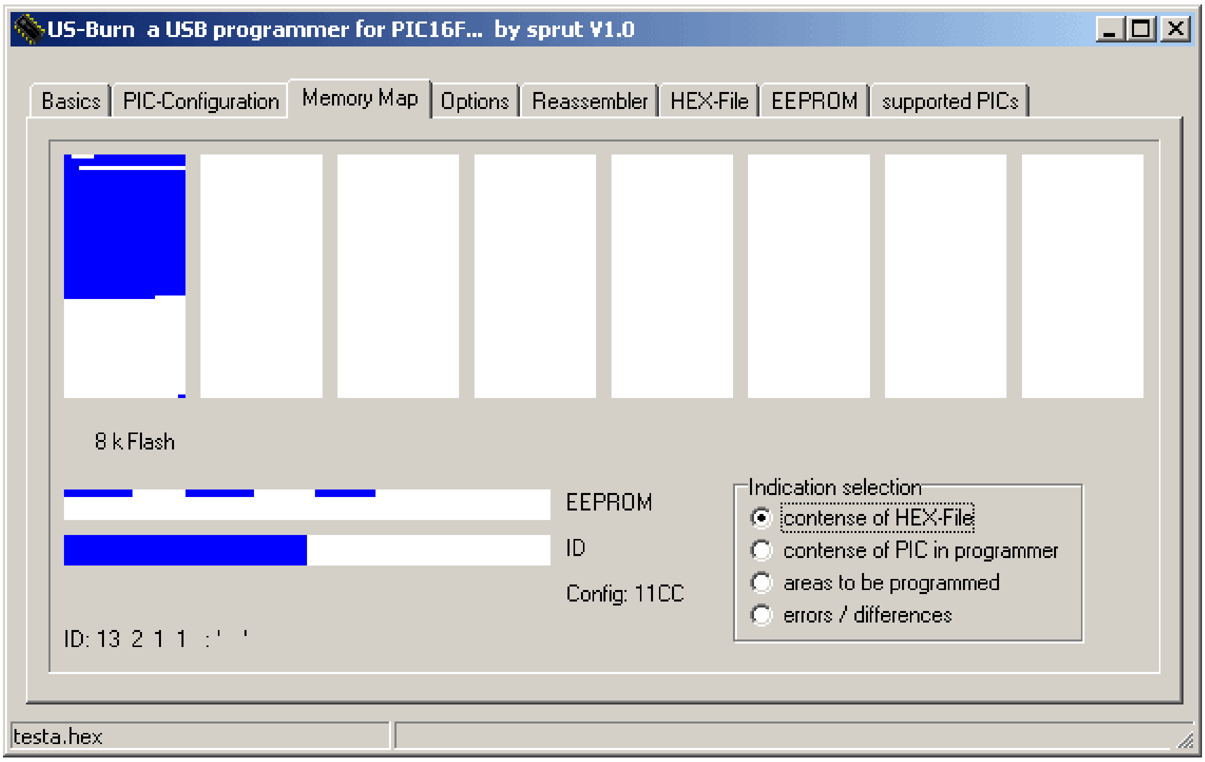


Figure 19 graphical memory display - hex file is loaded

The graphic memory display shows the used memory space and the available memory space of the detected PIC. Free space is white, used space is blue.

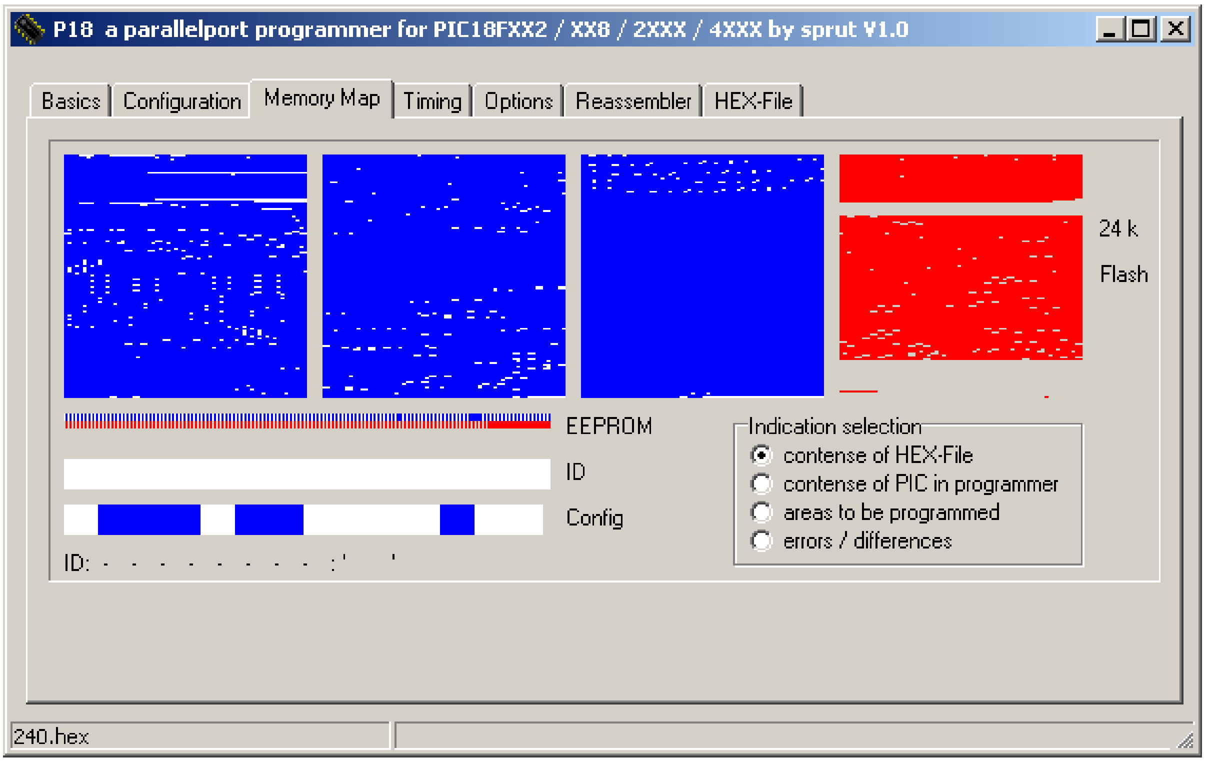


Figure 20 graphical memory display - loaded data is to large for the selected PIC

If parts of the hex-file-date don't fit into the real existing memory space of the PIC, then the non-fitting data is displayed in red.

If the compare-function identifies errors after the PIC was programmed, then these errors are displayed in red too.

10.6.2 Disassembler

If a hex-file is selected, then it will automatically be disassembled and the result can be seen in the disassembler window.

If the disassembler detects errors inside the code (unknown command codes, jumps into the nowhere...) then the number of such errors will be displayed in the main log-window.

To separate code from data is not a simple task. To be on the safe side, the disassembler even tries to disassemble memory areas that seem to contain data only. In this areas code-errors are ignored an every “code” line ends with a “DW 0x…”-comment.

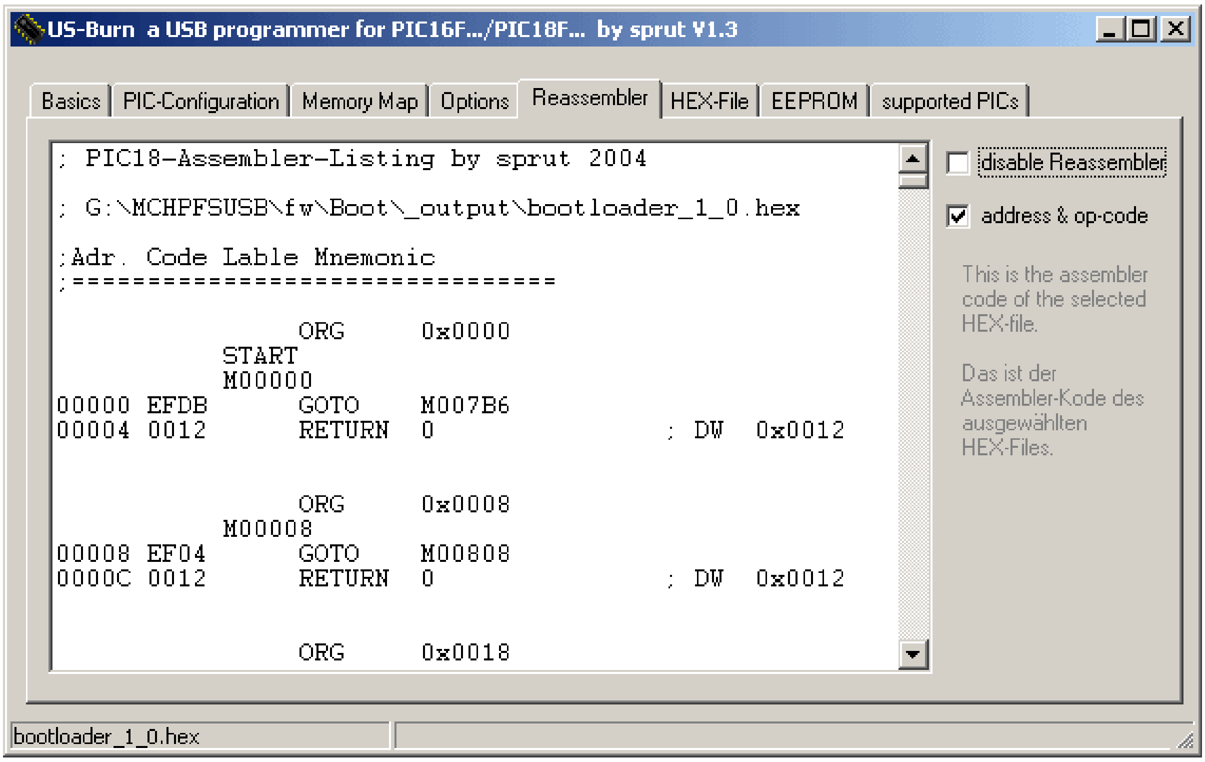


Figure 21 Disassembler window

The integrated disassembler can be disabled (**disable Disassembler**). An activated disassembler would cause additional processing delays during the load of HEX-files and during the flash-process.

An active disassembler may cause error messages during the hex-file loading process. This happens if the hex-file contains “strange” code. However, this will have no negative influence on the “burn”-process.

I suggest to disable the disassembler during normal work.

10.6.3 HEX-File viewer

After a HEX-file was selected, its content can be seen in the HEX-file-window.

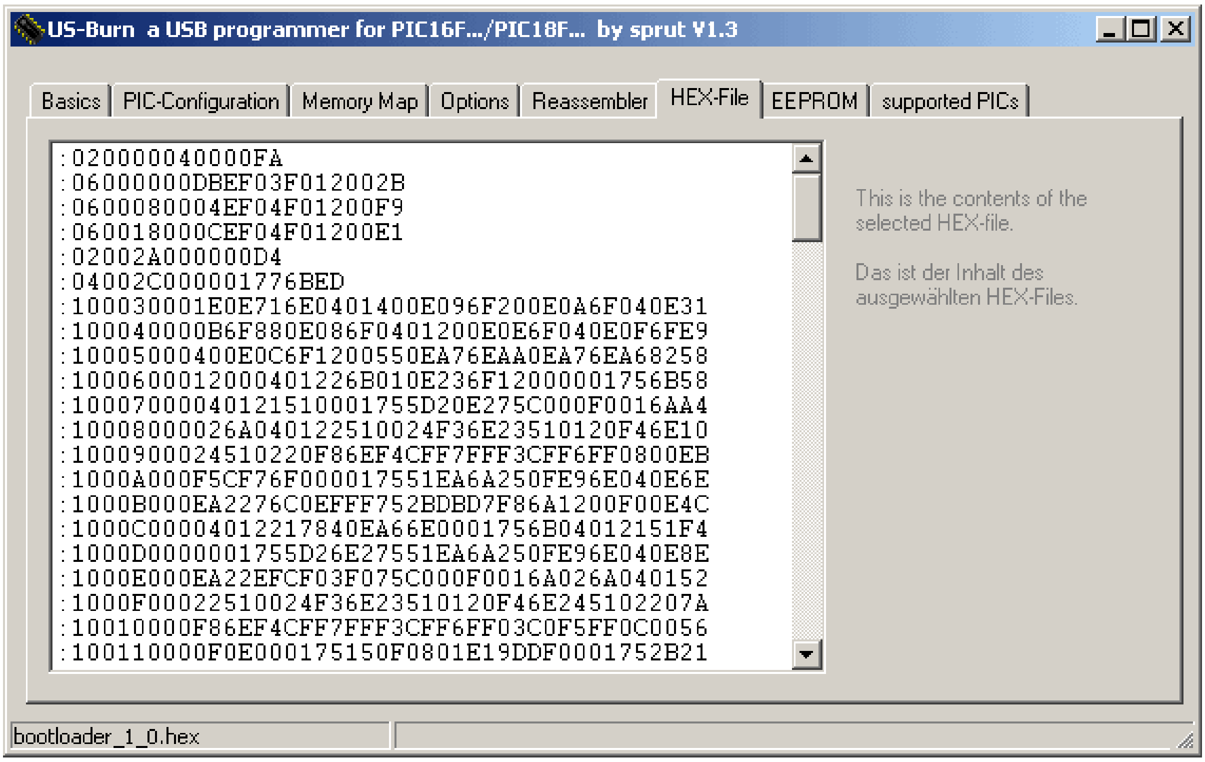


Figure 22 Hex-File viewer

10.6.4 EEPROM-viewer

After a HEX-file was selected, the EEPROM-data of this HEX-file is shown in the EEPROM-window. The window will show EEPROM-cells only, which are really contained in the HEX-file.
The window will show the data in hexadecimal format and as ASCII-symbols.

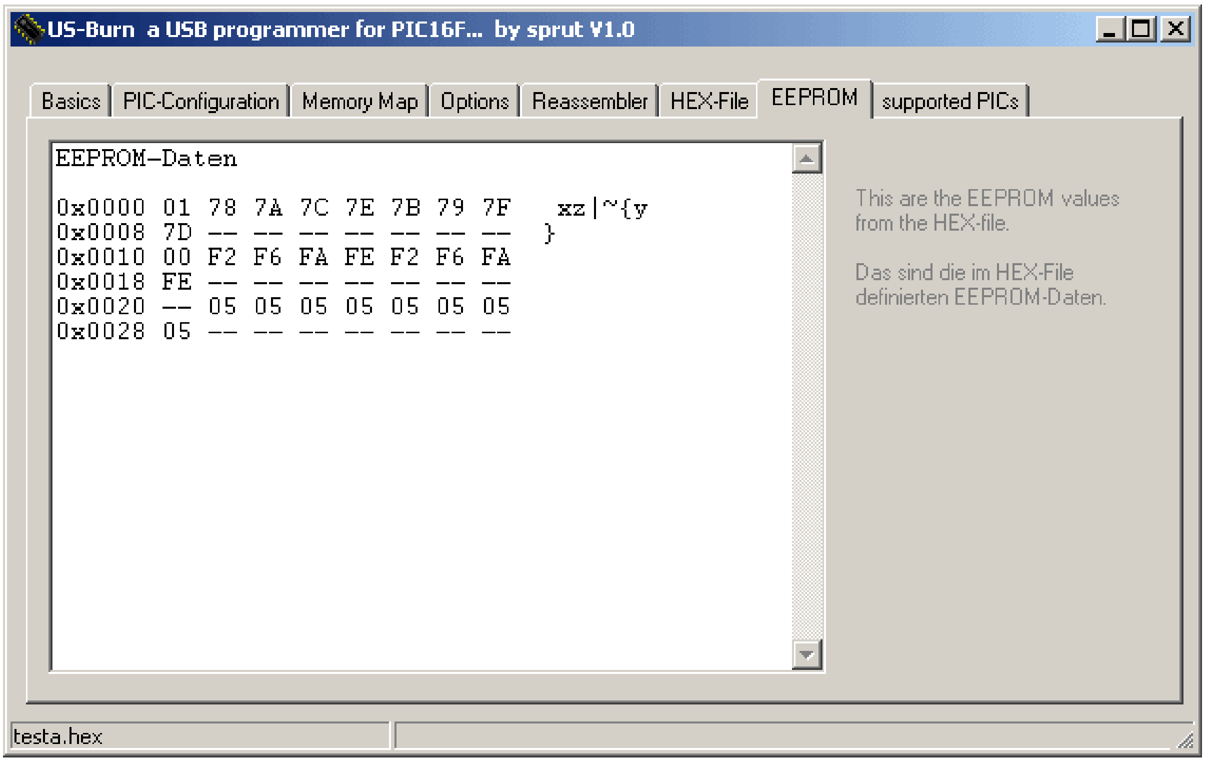


Figure 23 EEPROM viewer

10.6.5 Options

Some options are active by default at program launch. At the option-window one can switch them off or on. But this is not necessary under normal conditions.
Changes selections will be stored.

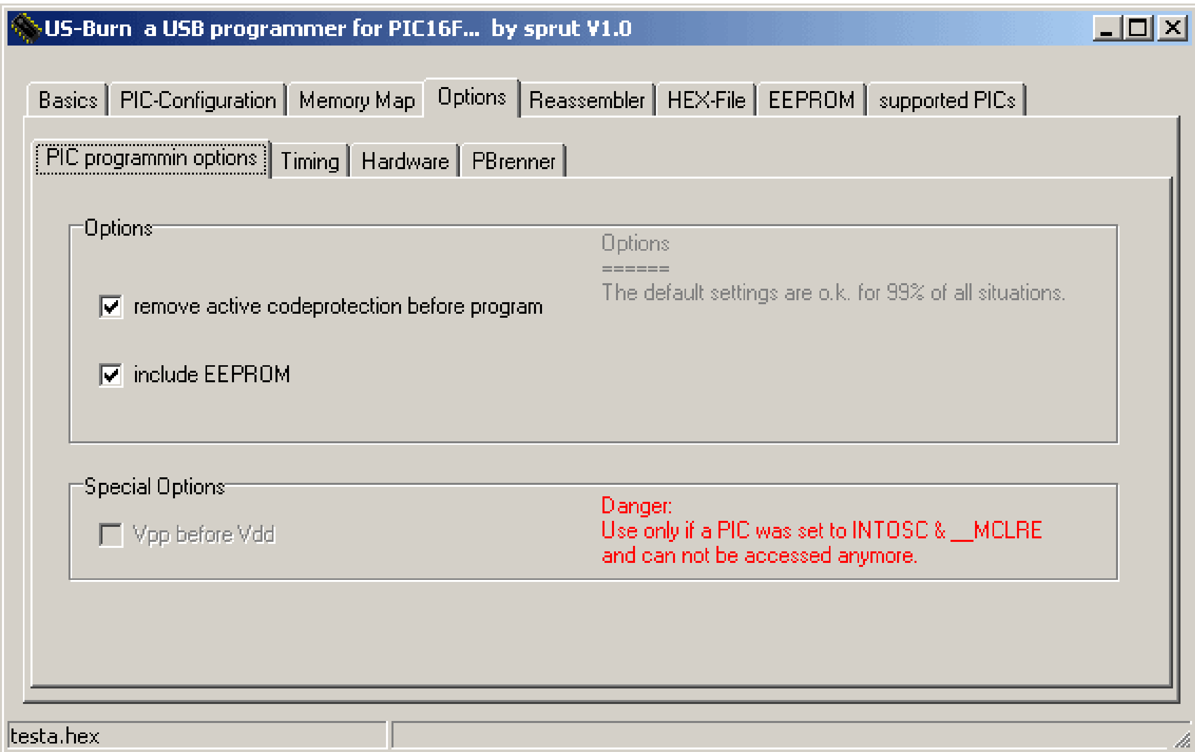


Figure 24 Options window

10.6.5.1 remove active codeprotection before program

If this feature is activated, then the software will test the PIC for codeprotection (CP) at the begin of each programming process. If CP is active, ten CP will be disabled before the new HEX-file is flashed into the target PIC.

This feature should be activated.

10.6.5.2 Vpp before Vdd (not supported jet)

The target PIC is fed with two voltages: Vdd and Vpp. During power up they are switched on in a sequence: Vdd fist and then Vpp.
If this option is activated, then the order is changed. This is a bit risky, but it may be the last chance if you stepped into the INTOSC-MCLR-trap. If a PIC was programmed with **INTOSC** enabled and **MCLR-Pin enabled** disabled, then it can happen that this PIC can not be programmed anymore. In this case - and only in this case - you may activate this feature and try to erase the PIC. If this was successful, then disable this feature immediately.

During program launch this feature is always disabled for safety reasons.

10.6.6 Timing

The Timing-modification-functions are disabled in this program version.

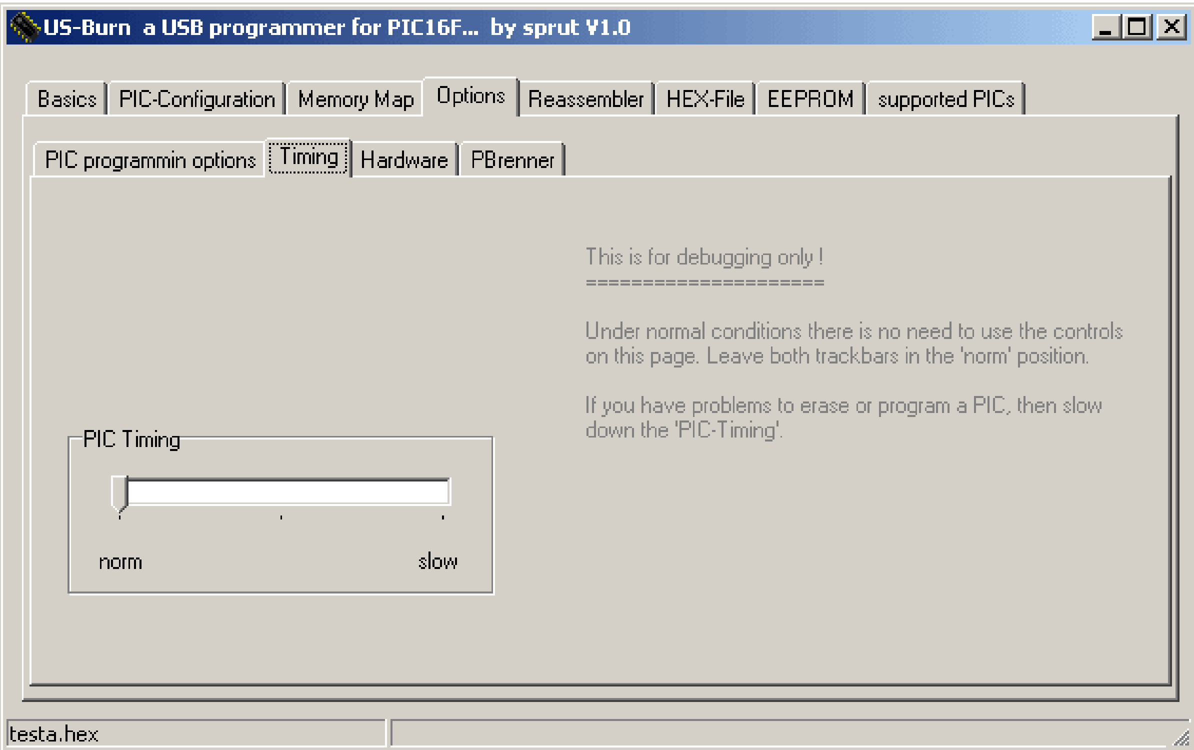


Figure 25 Timing window

10.7 Hardware setup

The **‚Options'-‚Hardware'**-window can be used to calibrate the programming voltage source of the programmer and to check the function of each signal line between

programmer and target PIC.
During calibration and test no target PIC should be connected to the programmer or plugged into the test socked.

10.8 Calibration of then programming voltage Vpp

The calibration of the Brenner8 is explained in an earlier chapter.

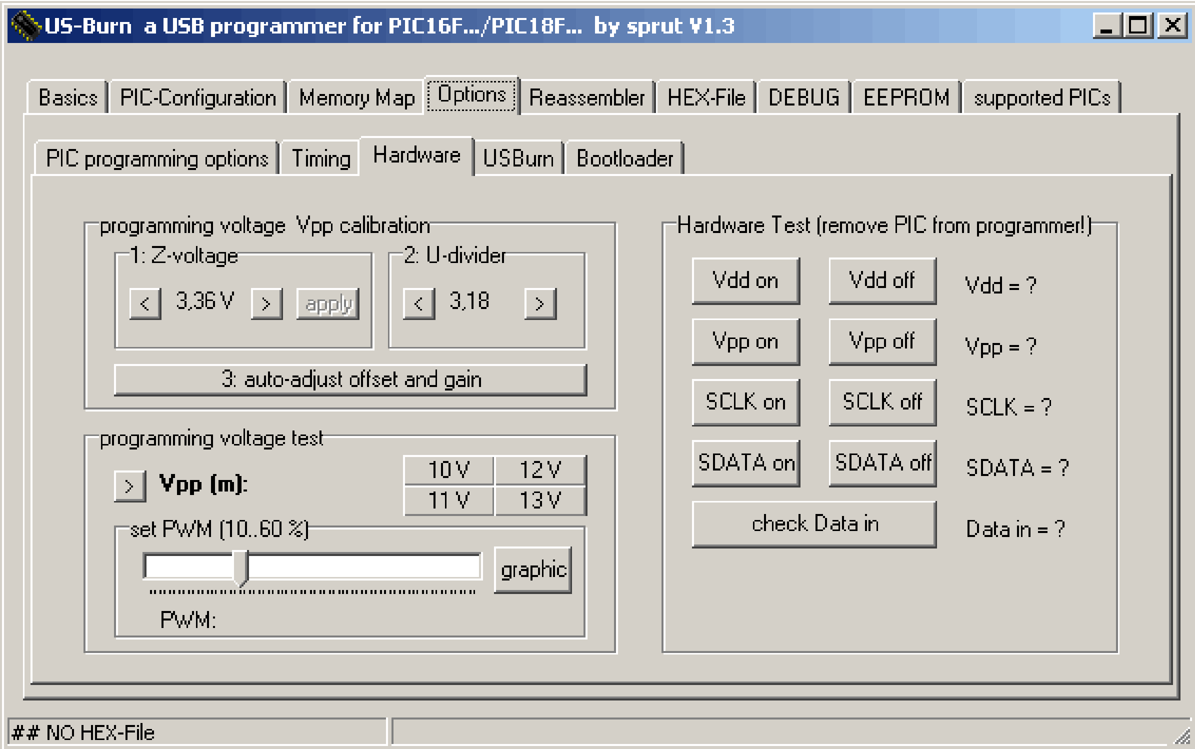


Figure 26 Hardware setup

10.9 OSCCAL-Editor

Some PICs have stored a calibration value (OSCCAL) for the internal oscillator. It may be stored as part of a RETLW command in the last flash program memory cell.

USBurn cares for this value and prevents a data loss during reprogramming of the target PIC. But nothing is 100% safe. If the data was lost or if the user likes to change this value, then the OSCCAL-editor can be used to do this.

The OSCCAL is an integer number between 0 and 255. In the most cases it will be close to 128.

The ‚**old OSCCAL from PIC'**-panel shows the value stored in the PICs. This value is located in the flash memory, but some software can write a backup value into the EEPROM.
The following figure shows an example. The flash contains the OSCCAL-value 136 while no value was found in the EEPROM. This is a normal scenario.

In the ‚**source for OSCCAL**'-panel the user can select the source for the new OSCCAL-value. By default ‚**keep old OSCCAL**' is selected. With this setting the

value from the flash would be used.
But the user can use the value from the HEX-file (what normally makes no sense at all) or a manual selected value instead.

If the user likes to use the backup value from the EEPROM, then he has to select **manual selected OSCCAL** and to click on **get it from EEPROM**.

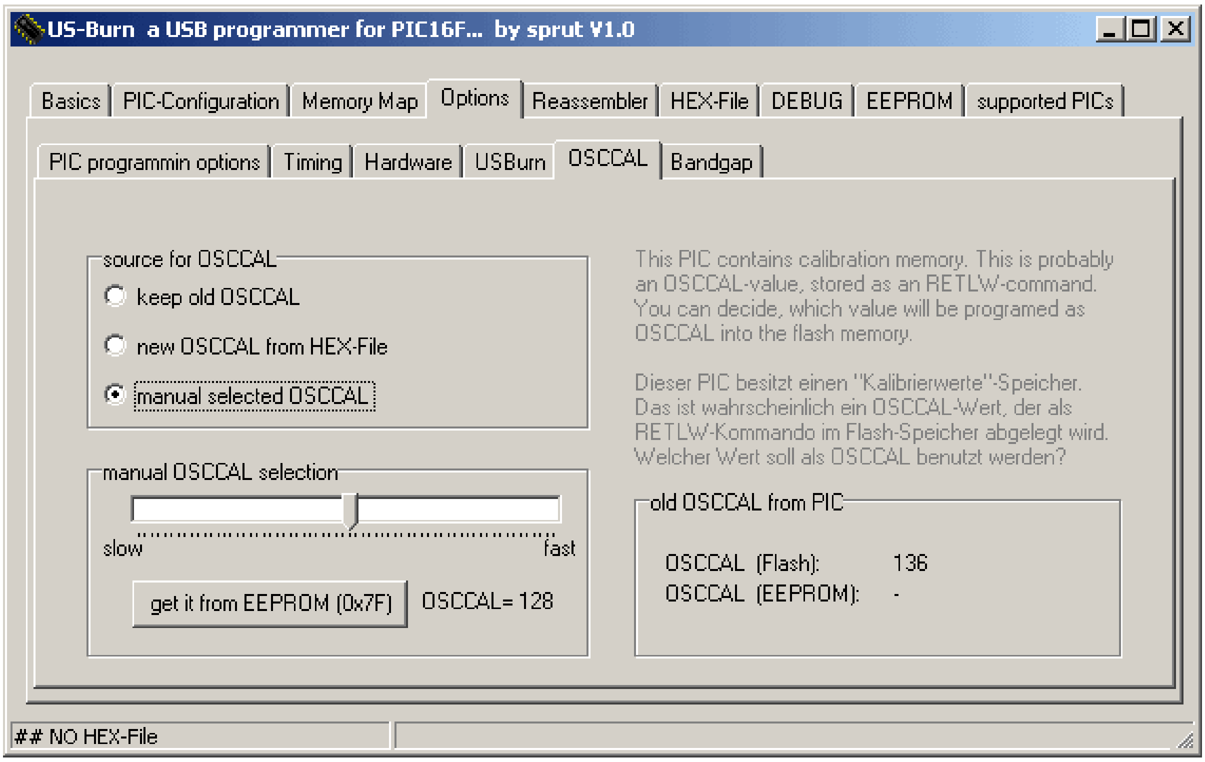


Figure 27 OSCCAL editor

10.10 Bandgap-Editor

Some PICs have stored a calibration value for an internal bandgap voltage source. This calibration value can be part of the configuration of the PIC. USBurn cares for this calibration data and prevents a loss of the data during reprogramming. However, there is a little chance that the data may get los, and it can happen that the user likes to change this BG-value.

The Bandgap-editor can be used to define a new BG-value.

After a click on '**identify PIC in Programmer'** the actual BG-value will be red from the target PIC. The position of the BG-ruler shows this value (from low to high). The ruler can be moved by the user to select a different BG-value. During the next programming step (**write HEX-file into PIC**) the modified value will be written into the target PIC.

If after the programming the **Compare PIC with HEX-File** button is clicked, then 1 single configuration error would be reported. This is normal, because the BG-configuration-value was manually changed.

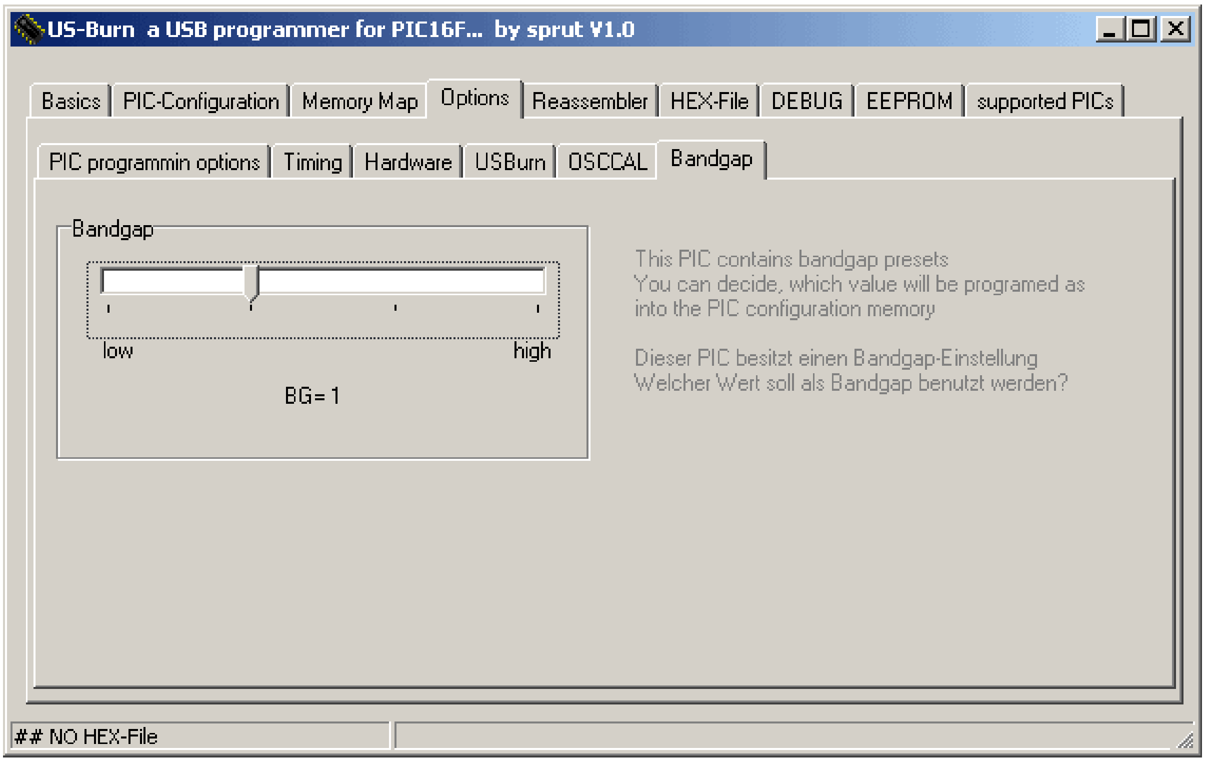


Figure 28 Bandgap editor

10.11 Possible Problems and solutions

10.11.1 What are all this options in the PIC-Configuration-window?

Many of this options are explained (in German language) on my homepage www.sprut.de/electronic/pic/config/config.htm.
The datasheets (in English language available on the Microchip homepage) contain descriptions of all options in detail.

Error message: „file to large or corrupt“ during hex-file-load or programming of the PIC
This error is caused by the disassembler on 16-bit operating systems (e.g. Win98) if large hex-files are used. This error message has no negative influence on the programming process. However, I would suggest to disable the disassembler.

10.12 Speed of the Brenner8

The time needed to program a PIC varies from type to type. New types can often be programmed faster the older types.

The following table lists some examples. The duration is the time to program the whole flash-program memory and EEPROM-data memory and to check both memory areas for correctness.

Table 6 Time to program a PIC

Type	Flash-Cells	EEPROM-Bytes	Duration [seconds]
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

10.13 US-Burn removal

To remove US-Burn from your PC you have to erase the following files:

- xxx\usburn*.exe
- xxx\mpusbapi.dll
- xxx\mpusbapi.dll
- xxx\picdef3.dll
- xxx*03.dat (the 6 files of the PIC-database)
- xxx\usbrn.hlp
- xxx\usburn.ini

and to deinstall the USB-driver.

The prefix ‚xxx\‘ has to be replaced by the real installation directory path of US-Burn.

10.14 Known problems

1)
The type-identification fails for the exotic PIC16F83.
USBurn don’t separate between PIC16F636 and 16F639. (no real problem)

11 Bootloader

From time to time a new version of the firmware will be published to fix errors and to integrate new features. The bootloader is a simple to use tool to load the new firmware into the PIC.

The bootloader is small piece of software, which has to be programmed into a special area of the control PIC of the Brenner8. To Program it into the control PIC am a PIC programmer is needed. This can be a Brenner5 (with Windows-Software P18) or a Brenner8 (Firmware 0.5 or later; software US-Burn V1.2 or later).

The bootloader is available in a separate hex-file or in a combination with the Brenner8-firmware.

From now I assume that the bootloader is programmed in the control PIC.

11.1 Start the Bootloader

The bootloader is not needed during the normal operation of the programmer. It stays inactive.
But if new firmware has to be loaded into the USB4all, then the bootloader has to be activated. There are two ways to do this:

- Activation by software USBurn
- Activation by jumper JP1

11.1.1 Activate the Bootloader with US-Burn

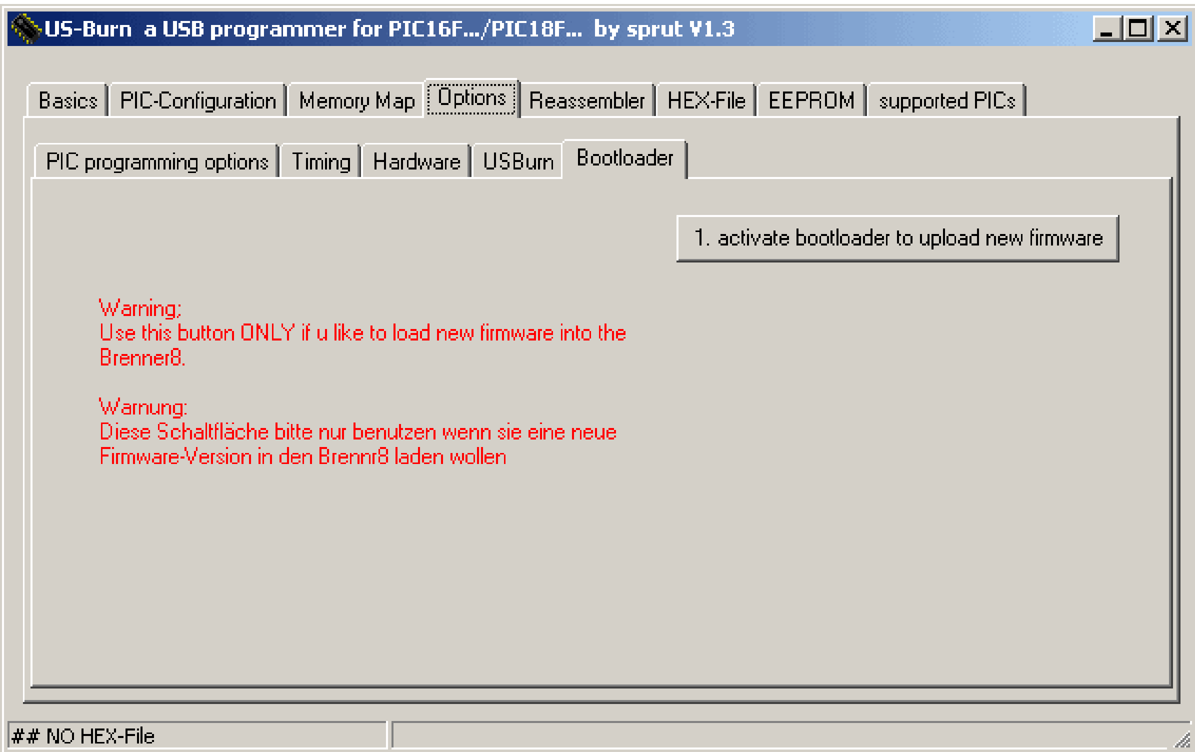


Figure 29 Activation of the bootloader by US-Burn

To activate the bootloader by software open the page **Options-Bootloader** and click

The software marks the old firmware an non-valid and reboots the programmer. This will need about 2 seconds. The bootloader detects, that the firmware is not valid anymore and becomes active.
Both LEDs light up.

11.1.2 Activate the Bootloader by the Jumper JP1

First disconnect the device from the PC. Then close the jumper JP1 and connect the device with the PC. Now the bootloader is active and the jumper can be removed.

If you hardware is missing the JP1, then simply connect pin 1 of the control-PIC with ground (Vss) while you reconnect the device with the PC.

The bootloader becomes active and both LED light up.
Now start USBurn.

11.1.3 Upload new Firmware into the Brenner8

If USBurn detects an active bootloader, then it will show this window.

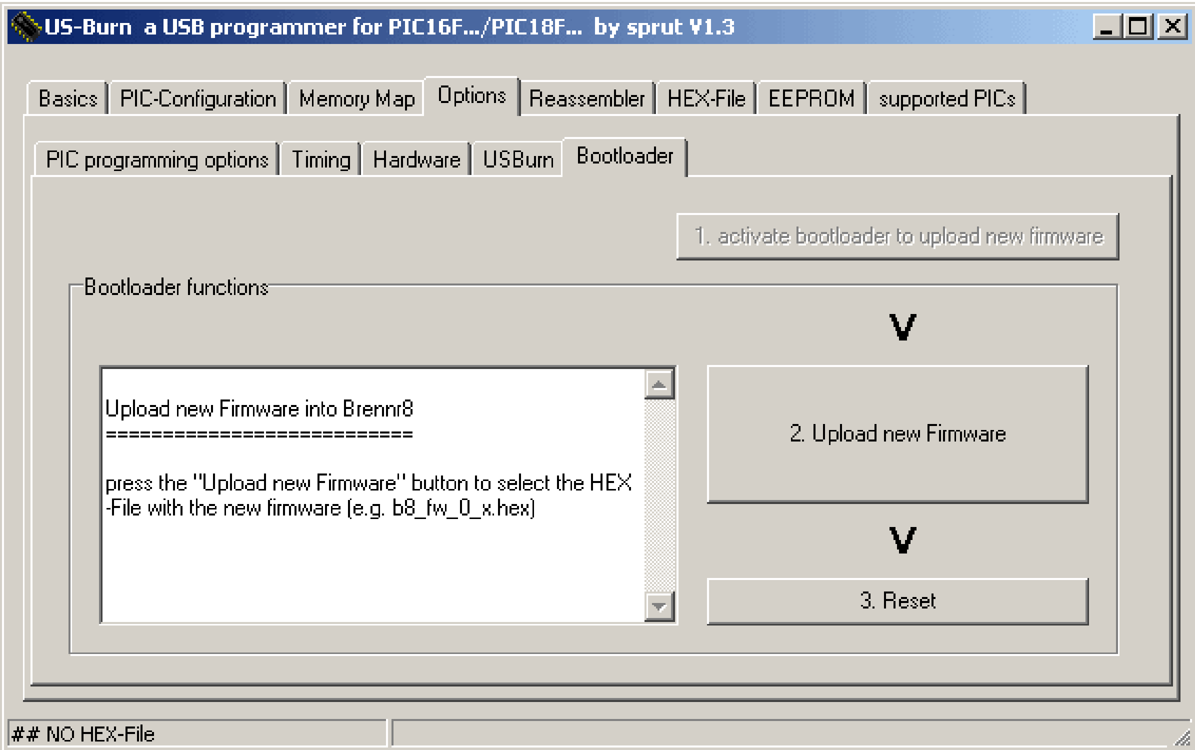


Figure 30 Loading new firmware into the Brenner8

Then click on the button **2. Upload new Firmware**. A file selection window opens. The user has to select the correct HEX-file with the new firmware. If this was done, then the software will:

- Load the HEX-file,
- Flash the new firmware into the control PIC of the programmer,
- Check the flashed firmware for correctness and
- Mark the new firmware as valid.

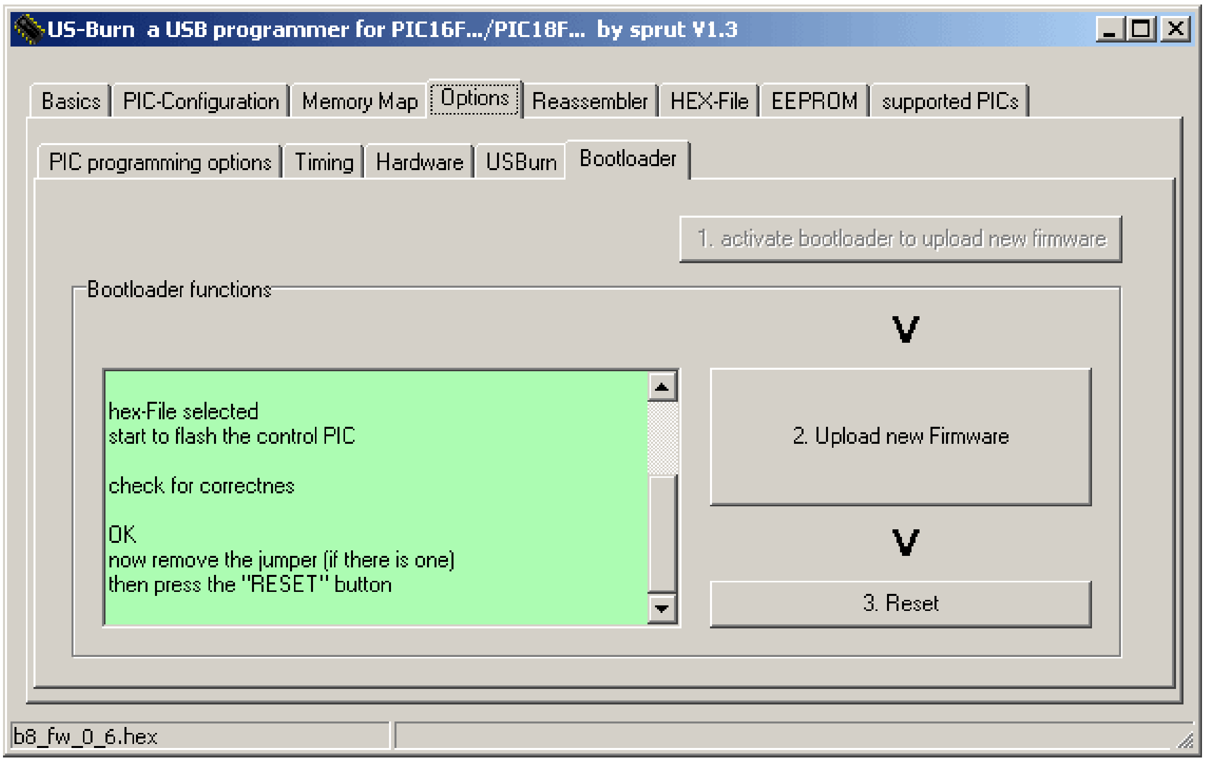


Figure 31 new firmware successfully loaded

If you forgot to remove the jumper JP1, then please do it now.

A click on the button **3. Reset** restarts the programmer. The programmer will be operational after 2 seconds.

The bootloader will not change the calibration data. A recalibration is not needed.

11.1.4 If you loaded a wrong HEX-file into the Brenner8 …

The bootloader can not know, if the selected HEX-file is really a valid firmware. If by accident a wrong HEX-file was selected, then the bootloader will flash it into the programmer. Of course then the programmer will not be operational anymore, but nothing is lost (except the old firmware).

The bootloader will not be damaged, and the user can always use the jumper JP1 to activate the bootloader. Then you can try again to select the correct HEX-file.

12 Attachments

The latest versions of schematics and layouts are available from www.sprut.de.

12.1 Brenner8 - schematic

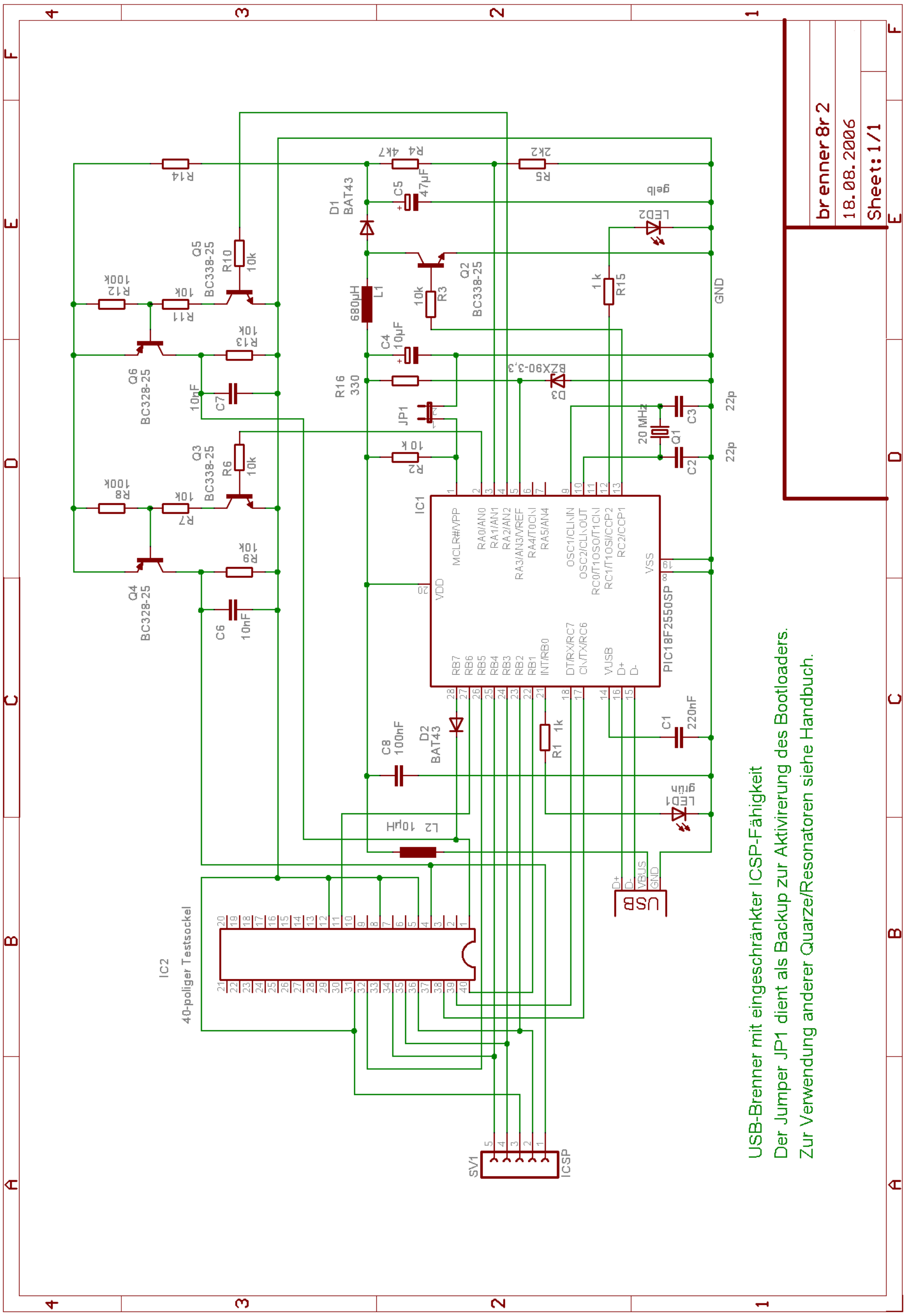


Figure 32 Schematic of Brenner8 (revision 2))

12.2 Brenner8 - Components side of the PCB

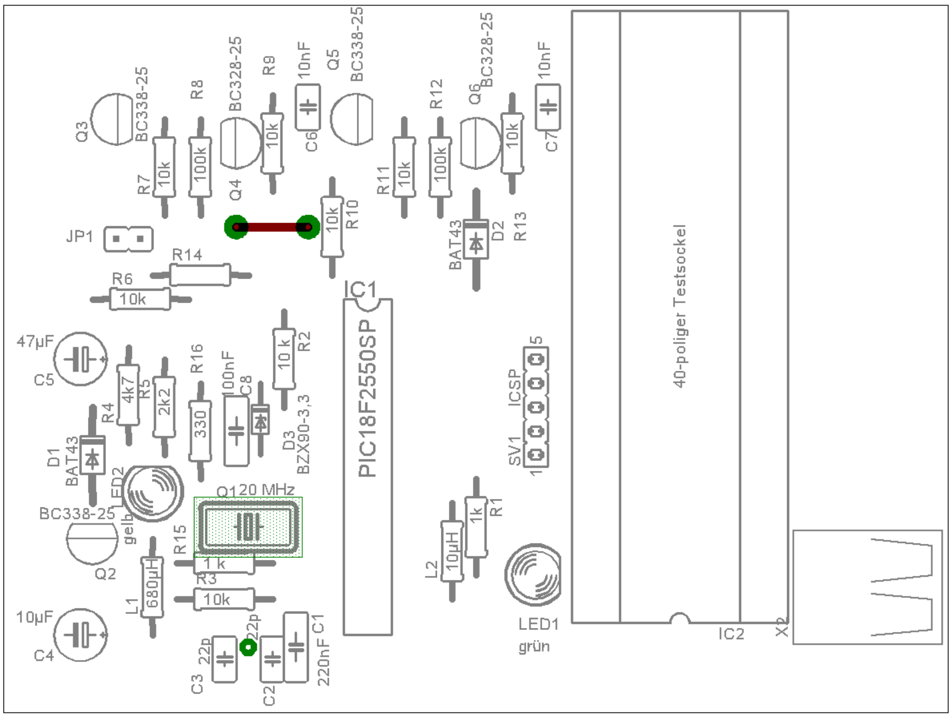


Figure 33 Component side of Brenner8

12.3 Brenner8 – PCB-layout

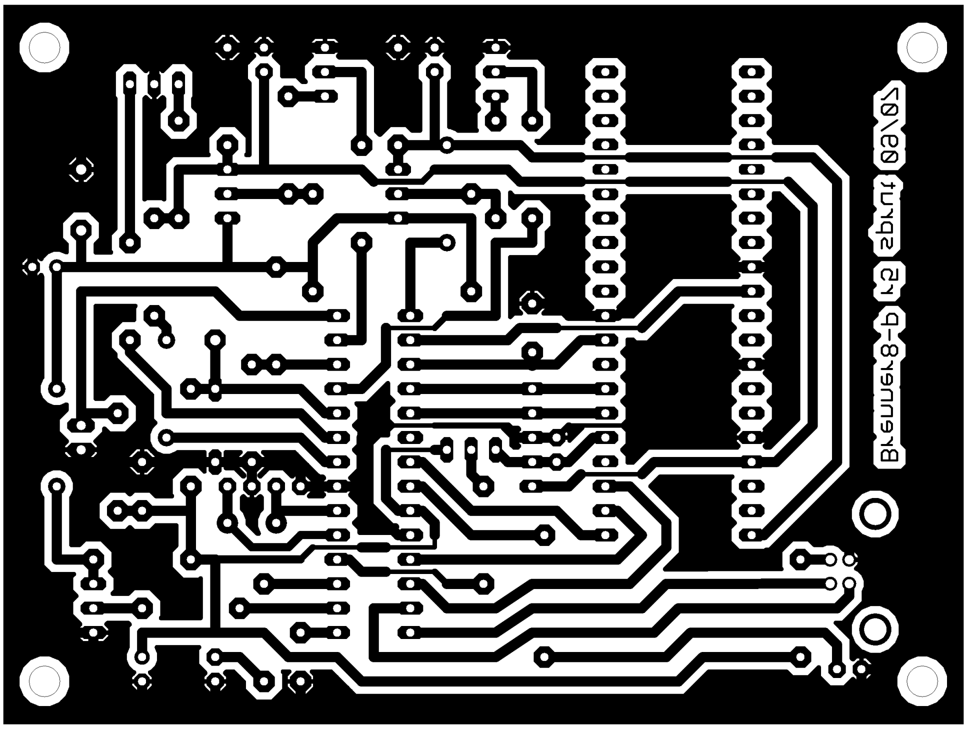


Figure 34 Layout of Brenner8, 75mm x 100mm (not in scale

12.4 Brenner8 - part list

Partlist für Brenner8 R1

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

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.

12.5 Brenner8mini-P Schematic

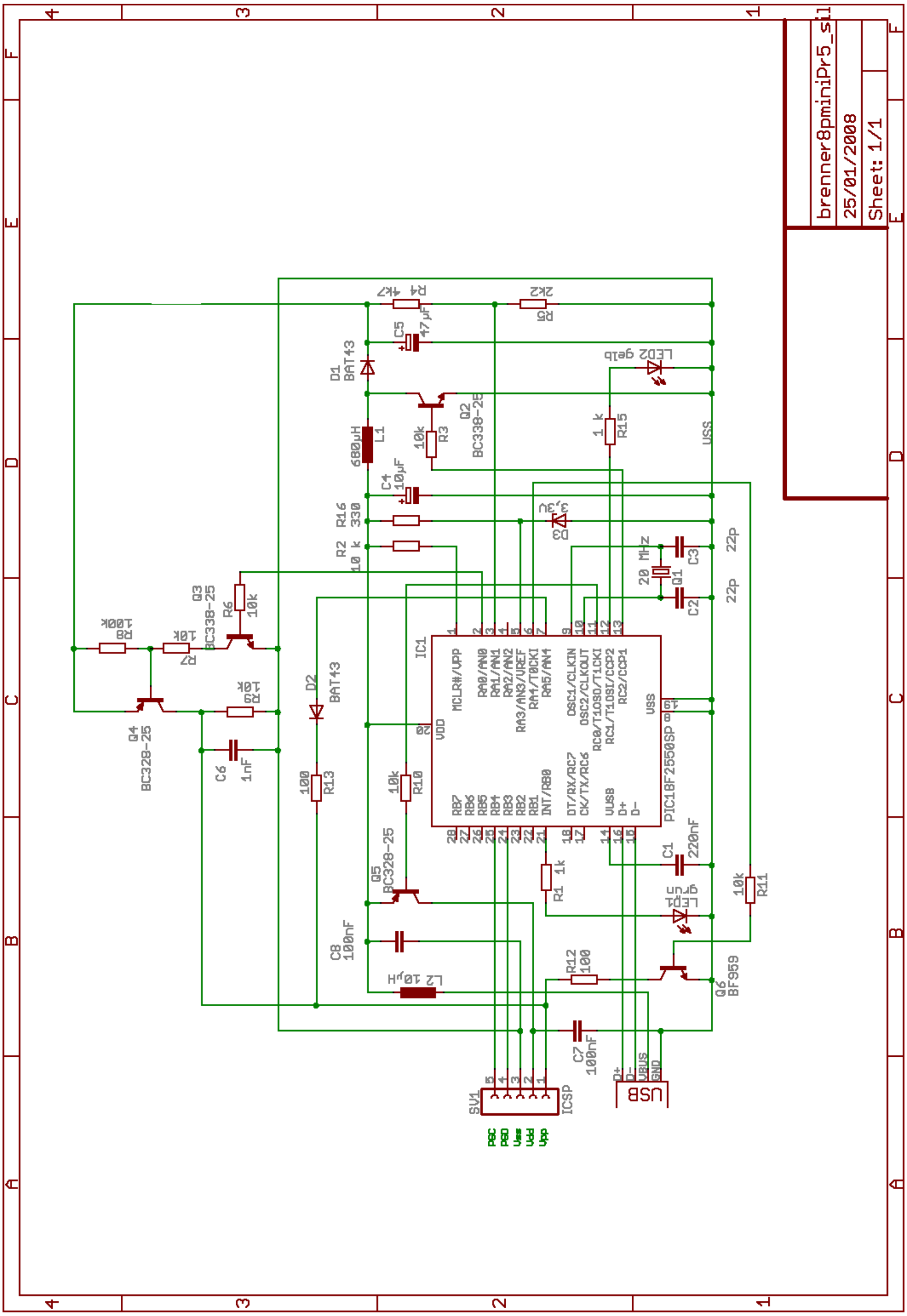


Figure 35 Schematic of Brenner8miniP (Rev. 5 Silviu)

12.6 Brenner8mini-P - Components side of the PCB

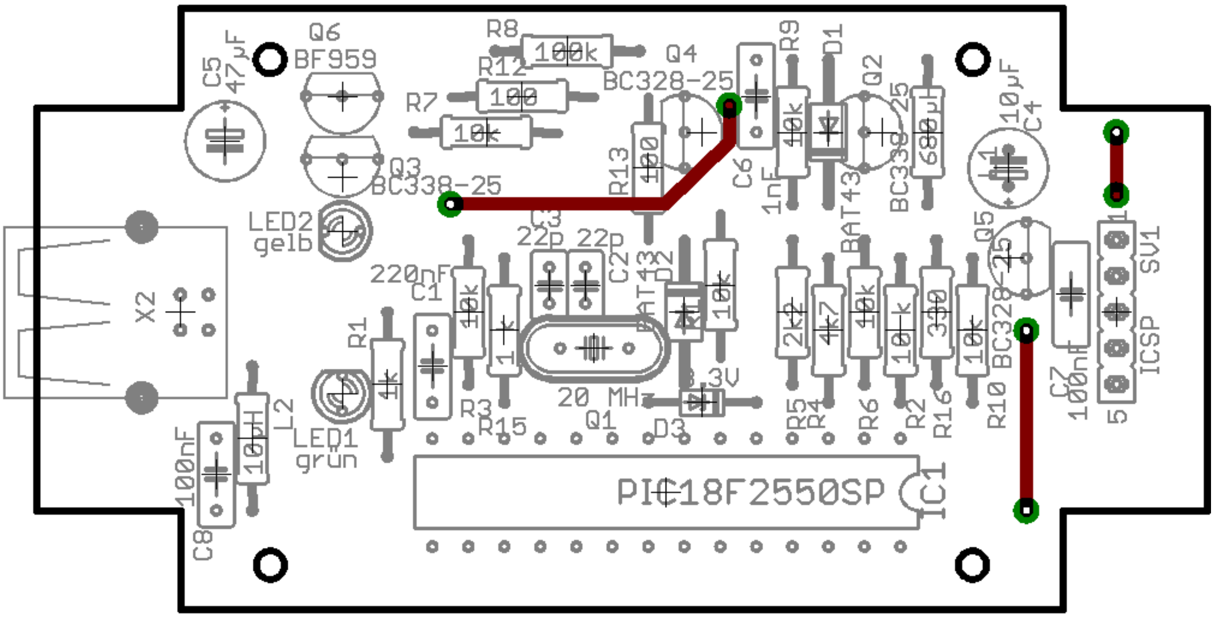


Figure 36 Component side of Brenner8miniP

12.7 Brenner8mini-P – PCB-layout

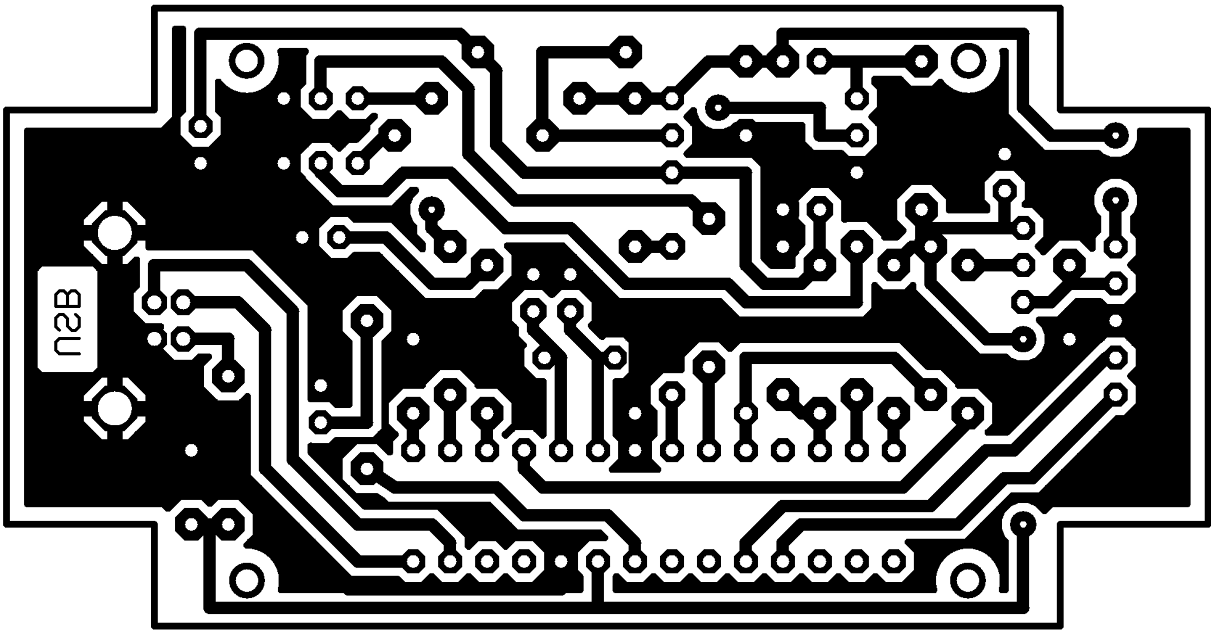


Figure 37 Layout of Brenner8mini-P, 83mm x 43mm (not in scale

12.8 Brenner8mini-P - part list

```
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
```

12.9 18FxxKxx-Adapter - Schematic

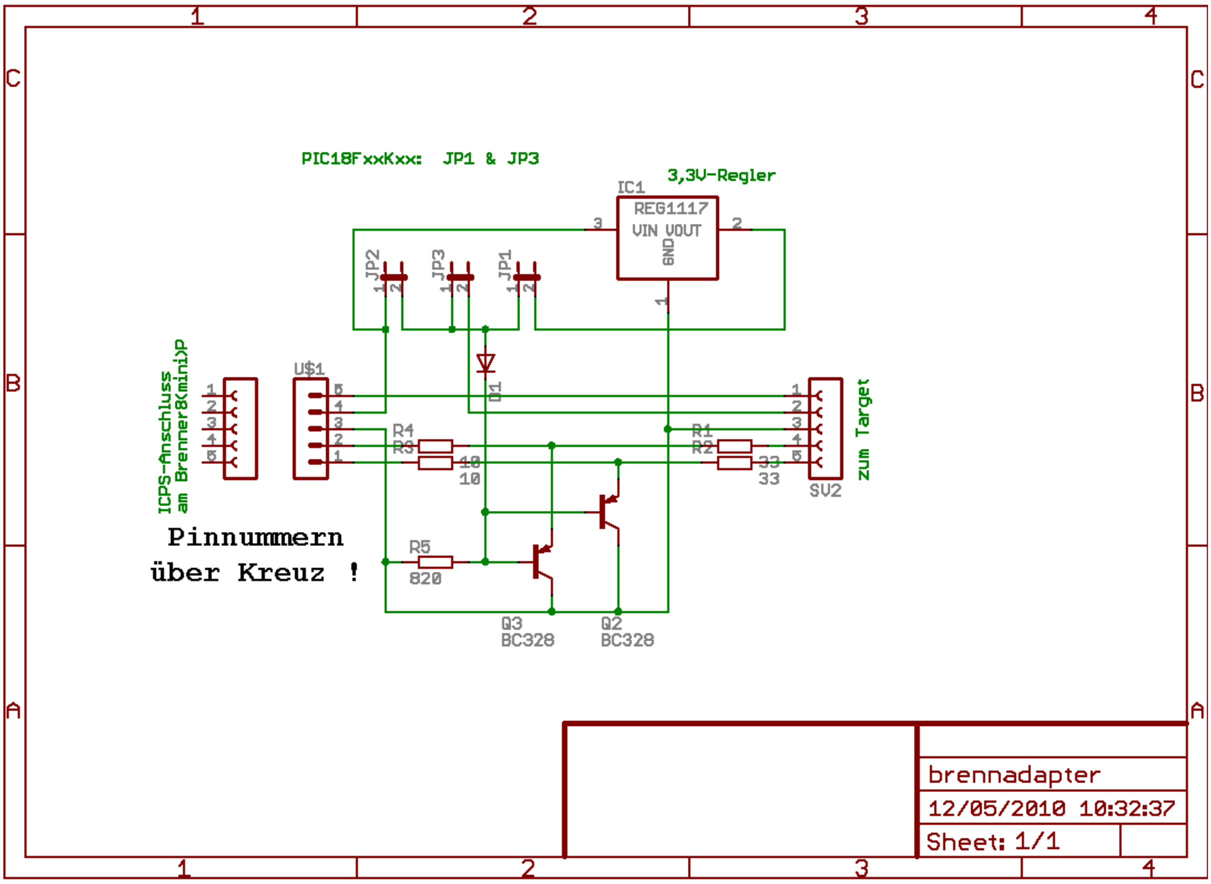


Figure 38 Schematic of the Adapter

12.10 18FxxKxx-Adapter - Layout

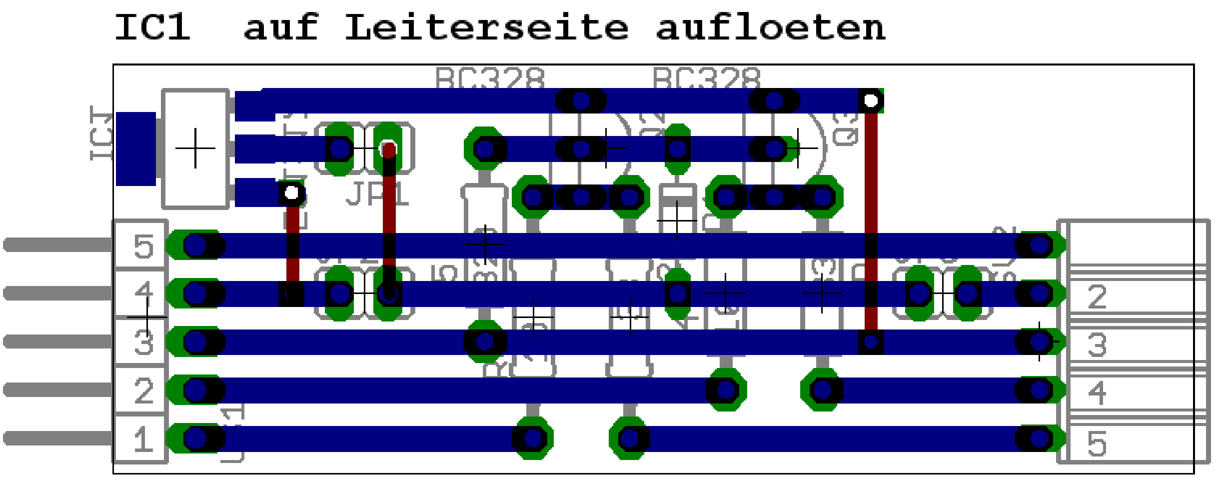


Figure 39 Layout of the Adapter

12.11 18FxxKxx-Adapter - Components side of the PCB

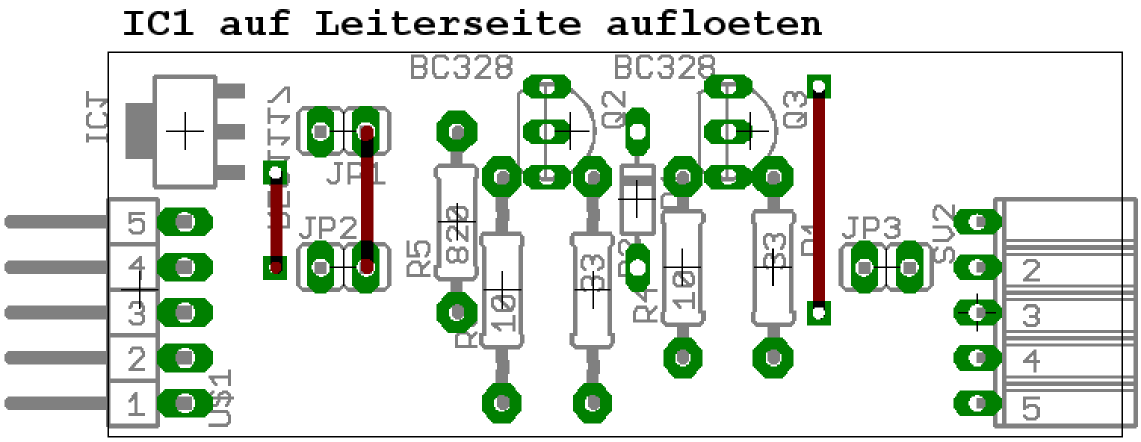


Figure 40 Components side of the Adapter

12.12 18FxxKxx-Adapter - List of Parts

```
Partlist für Brennadapter

Part      Value          Reichelt
=====================================
D1        1N4001         1N 4001
IC1       REG1117        LT 1117 CST-3.3
JP1,2,3   Jumper         SL 1x36G 2,54
Q2, Q3    BC328          BC328-25
R1, R2    33             1/4W 33
R3, R4    10             1/4W 10
R5        820            1/4W 820
SV2       ICSP out       BL 1X20W 2,54
U$1       ICSP in        SL 1x36W 2,54


Es werden 2 Jumper (Reichelt: JUMPER 2,54GL RT) benötigt.

JP1, 2, 3 sind aus einer Stiftleiste (SL 1x36G 2,54) zu fertigen.

Die Buchse SV2 ist auf 5 Pins zu kürzen.
Der Stecker U$1 ist auf 5 Pins zu kürzen.

Alle Widerstands- und Kapazitätswerte sind unkritisch (25%).

Es muss mindestens eine Firmware V 0.15 verwendet werden.

IC1 ist ein 3,3V-low-drop-Regler im SOT223-Gehäuse.
```