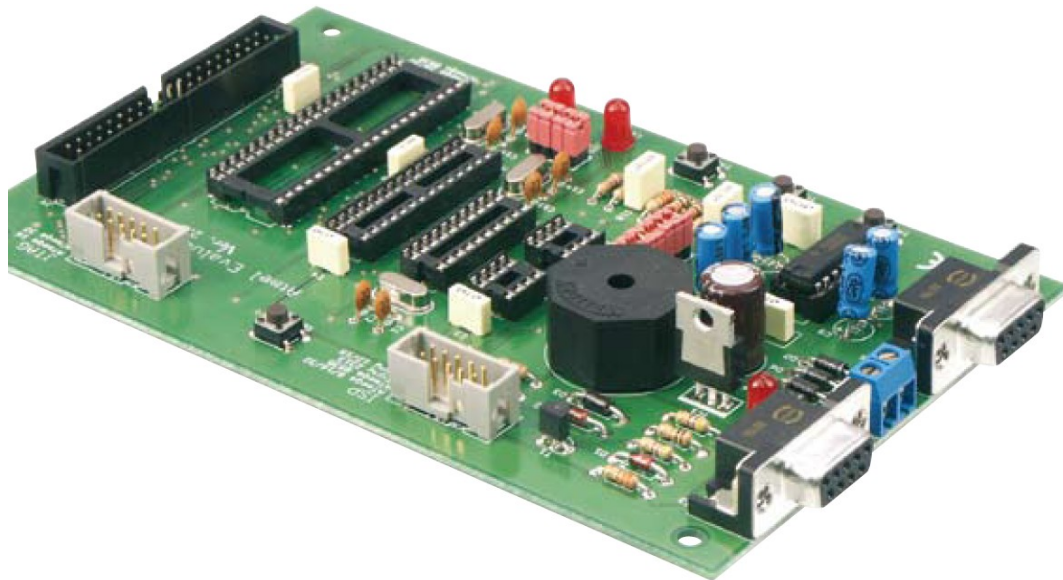


Mikrocontroller Einstieg

ATmega 8



BASCOM AVR

```

1 ' Atmega 8
2 ' Entwicklungsboard Neuhold
3 ' Version 0
4 '@ hpt 12.1.2012
5 ' Testen des 2 zeiligen - Displays 16217SY-PY
6 ' 2 x 16 Zeichen (groß grün)
7 ' auf Lochraster-Zusatzplatine
8
9 ' Achtung: Hintergrundbeleuchtung Pins vor 1.2 (nicht angeschlossen)
10 ' Masse >> R/V Display Pin5
11 ' Port PC1 >> J4/Pin2 >> Enable Display Pin6
12 ' Port PC2 >> J4/Pin3 >> R/S Display Pin4
13 ' Port PC13 >> J4/Pin13 >> DB4 Display Pin11
14 ' Port PC14 >> J4/Pin14 >> DB5 Display Pin12
15 ' Port PC15 >> J4/Pin15 >> DB6 Display Pin13
16 ' Port PC16 >> J4/Pin16 >> DB7 Display Pin14
17
18 *****
19 $regfile = "M8def.dat"
20 $crystal = 16000000 '16Mhz
21 *****
22 Config Lcdpin = Pin , Db4 = PORTB.2 , Db5 = PORTB.3 , Db6 = PORTB.4 , Db7 = PORT
23 Config Lcd = 16 * 2
24 'Config Portb = Output
25 ' *****
26 ' Testroutine
27 Cursor Off Noblink
28 Cls
29 Lcd " Test"
30 Lowerline
31 Lcd " V.0 "
32 Wait 1
33 Cls
34 ***** Hauptprogramm *****
35 Do
36 Locate 1 1
37 Lcd "Displaytest HPT"
38 Locate 2 1
39 Lcd "1234567890123456"
40 Loop
41 End
42
43
44
45
46
47
48
49
    
```

Chip Pinbelegung

Gehäuse: DIP28 C:\... Chip suchen

Suchen Pin HL löschen

Package: DIP28, Pin: 5

PD3 - Generic IO pin PD3
 INT1 - External Interrupt source 1 to the MCU. See the interrupt description for further details and how to enable the source.

Einstieg in die faszinierende Welt der Mikrocontroller

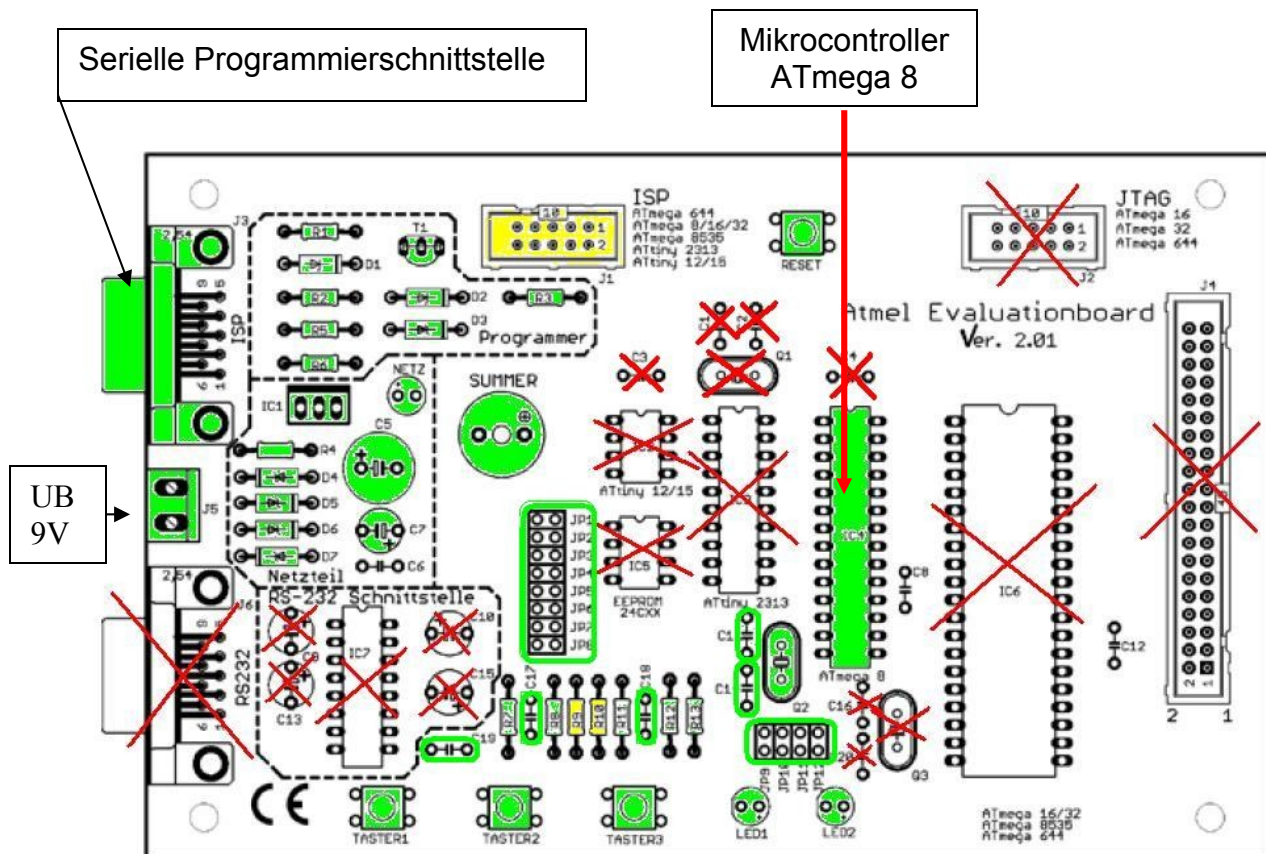
Für eine praktische Einführung in die Grundlagen und komplexen Zusammenhänge in die Thematik wurde ein moderner und zeitgemäßer Atmel AVR – (Risc) Prozessor sowie die schnell erlernbare Programmiersprache BASCOM gewählt.

1. Hardware

Für die Entwicklung und das Austesten einfacher Projekte, wird eine Lern- und Experimentier-Leiterplatte bestückt und aufgebaut. Diese Entwicklungsplatine, zum Anschluss (über ein serielles Kabel) an den PC ermöglicht die direkte Programmierung verschiedener Atmel- Mikrocontroller hier z.B. **ATmega8** .

Aufgrund der vorhandenen Elemente wie Taster, LEDs, Summer und serieller Schnittstelle ist eine einfache und rasche Anwendungsentwicklung und deren Austestung möglich. Über Jumper lassen sich Komponenten an – und wegschalten. Erweiterungen sind über eine 40 polige Steckverbindung J4 möglich.

Betriebsspannung **9 Volt** (Gleich- oder Wechselspannung)



Minimalbestückung für ATmega 8

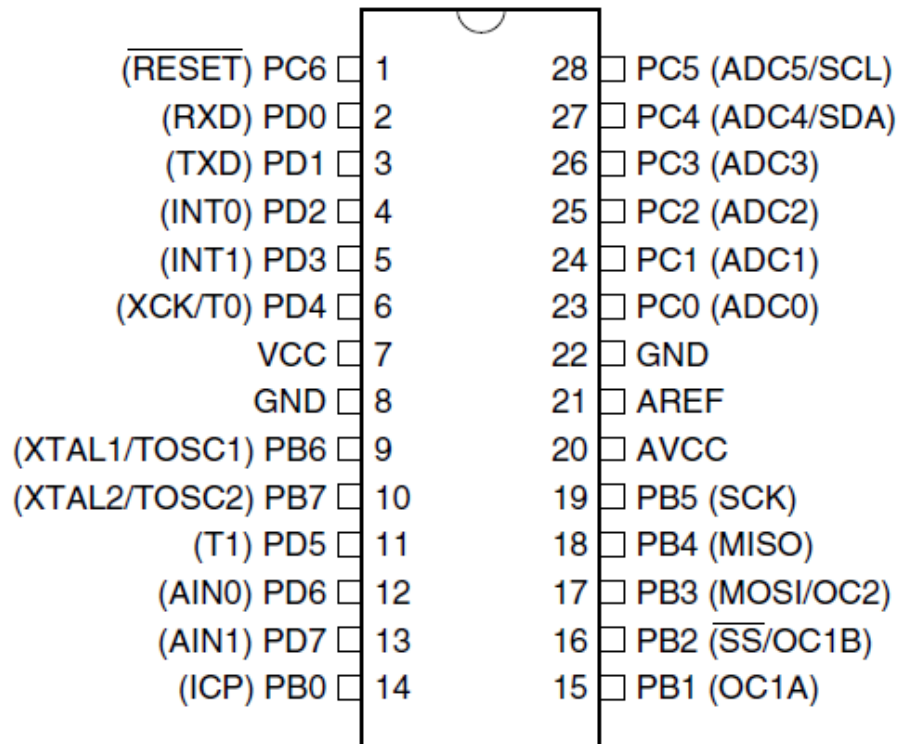
Vor Inbetriebnahme des Testboards unbedingt auf Kurzschlüsse prüfen. Zuerst ohne Prozessor testen und die Betriebsspannung von 5V prüfen!

1.1 Der Mikrocontroller



**8-bit AVR[®]
with 8K Bytes
In-System
Programmable
Flash**

ATmega8



Der (28-polige) ATmega8 besitzt neben einen 8 kB großen Flash als Programmspeicher, noch 512 Byte EEPROM und 1 kByte RAM.

Die meisten Anschlüsse des ATmega8 sind als Ports ausgeführt. Neben der Funktion als digitale Ein- oder Ausgang, sind die einzelnen Pins auch noch mit anderen Funktionen belegt, die man bei Bedarf verwenden kann.

Auch AD- Wandler (ADC0....ADC5), eine serielle Schnittstelle, eine SPI- und eine I²C- Schnittstelle sind auf diesem Chip zu finden.

Als Takt verwenden wir einen 16MHz Quarz. Dieser könnte auch weggelassen werden, da der Prozessor dann mit einem 1MHz internen Taktgeber arbeitet.

Der Controller der AVR- Reihe ist mit der so genannten Risc- Struktur ausgestattet. Dies bedeutet, dass für die Ausführung eines Maschinenbefehls, meist nicht mehr als 1 Systemtakt benötigt wird.

Durch ISP (In System Programming) kann der Baustein direkt auf der fertig aufgebauten Schaltung mit Hilfe eines einfachen Programmieradapters programmiert werden.

2. Software :

Außer der Hardware, benötigen wir noch eine (Programm-) Entwicklungsumgebung auf einem PC. BASCOM , eine BASIC- Programmiersprache, mit einem Compiler, welche Basic in eine für den Mikrocontroller verständliche Sprache umsetzt. BASCOM mit einem Editor, einen Simulator und einer Brenn- Programmiersoftware steht bis zu einer Programmgröße von 4kByte gratis im Internet zur Verfügung.

Programm Entwicklung besteht aus folgenden Schritten:

2.1 Programm Entwicklung in BASIC (BASCOM-AVR)

2.2 Maschinencode Compilierung (Bascom-AVR)

2.3 Programm Download auf Target Board (Pony-Prog)

Danach wird getestet und eventuell weiterentwickelt.

2.1.Programmentwicklung in Basic

Für die Programmierung des Mikrokontrollers gibt es verschiedene Möglichkeiten. Wir wollen das Programm "Bascom AVR" (Gratisdownload unter <http://www.mcselec.com/>) mit serieller Programmierschnittstelle benutzen. Benötigt wird ein PC mit einer seriellen Schnittstelle, und die Programmierumgebung, mit der die Programme geschrieben und compiliert werden können. Besitzt der PC (oder Notebook) keine serielle Schnittstelle, so benötigt man einen USB-ISP Programmieradapter.

- Zu Beginn wird "Bascom AVR" gestartet und im Menü "Datei" mittels "neu" ein neues Programm- Fenster erstellt.


```

1 'Evaluationsboard V 2.01
2 'HPT 11.1.2012
3 $regfile = "m8def.dat"           ' Controllerauswahl
4 $crystal = 16000000             ' Quarzfrequenz
5
6 Config PORTD = Output          ' PinD.5=LED1, PinD.6=LED2, PinD.7=Summer
7 Config PIND 2 = Input          '=Taster 1
8 Config PIND 3 = Input          '=Taster 2
9 Config PIND 4 = Input          '=Taster 3
10
11 Anfang:
12 If PIND 2 = 1 Then Set PORTD.5 ' Taster 1 gedrückt = LED1
13 If PIND 3 = 1 Then Toggle PORTD.6 ' Taster 2 gedrückt = LED2 ein - aus
14 If PIND 4 = 1 Then Sound PORTD.7 , 200 , 450 ' pin, Länge, Periodendauer
15 Waitms 250
16 Reset PORTD.5                 ' Ports zurücksetzen
17 Goto Anfang
18
19 End
20

```

- Das Programm wird geschrieben und gut dokumentiert !

2.2 Compilierung

Danach wird Menü >> Programmieren >> Compilieren oder mit Symbol  kompiliert und dabei ein xxx.hex und eine xxx.bin Datei erstellt.



2.3 Anleitung PONY-PROG

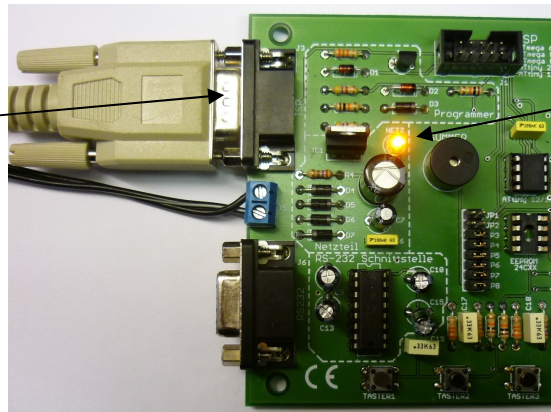
Damit die compilierten Daten in den Controller geschrieben werden können, benötigen wir noch PONYPROG (Gratisdownload <http://www.lancos.com/prog.html>).

Voraussetzungen: eine xxx.hex oder xxx.bin Datei.

Z.B. durch programmieren und compilieren mit BASCOM.

Hardware :

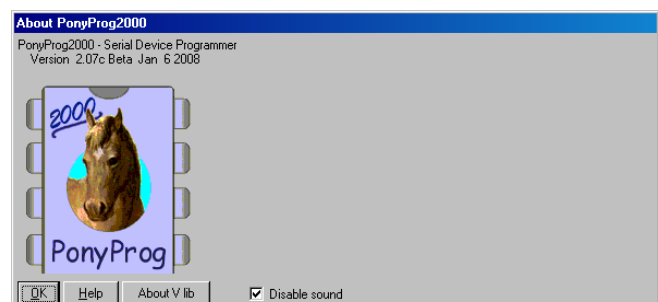
Serielle Schnittstelle



Betriebsspannung

Ein Controller
(hier ein ATtiny)

Wir starten das Programm
Pony-Prog 2000 (event. in deutsch)



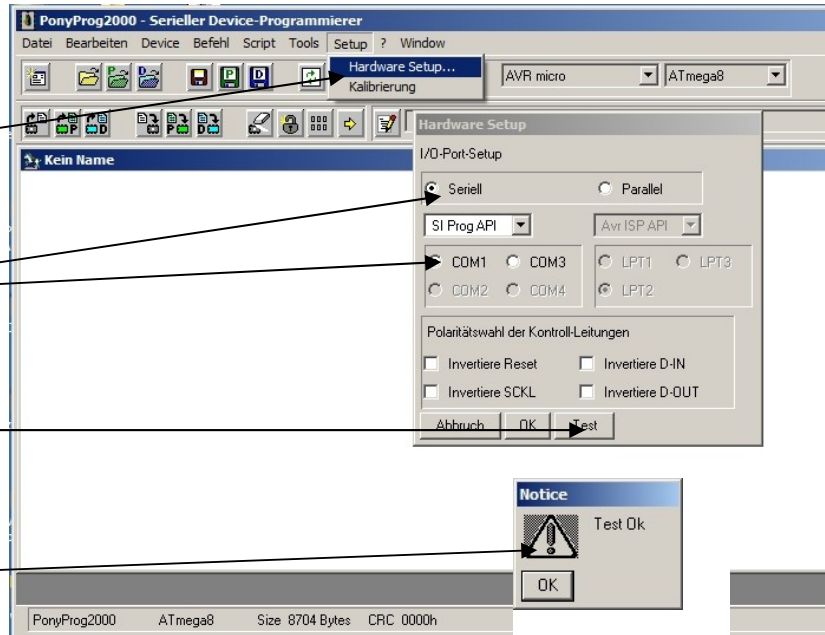
Nach dem Start von Pony Prog:

- 1.
- >> Setup
- >> Hardware Setup

1.1 Einstellungen treffen

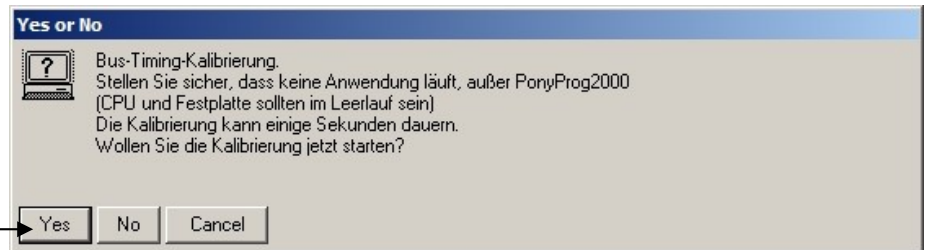
1.2 Testen

1.3 Bestätigung
Wenn alles OK



- 2.
- >> Setup
- >> Kalibrierung

Yes



Danach sollt dieses Feld erscheinen

Bestätigen OK

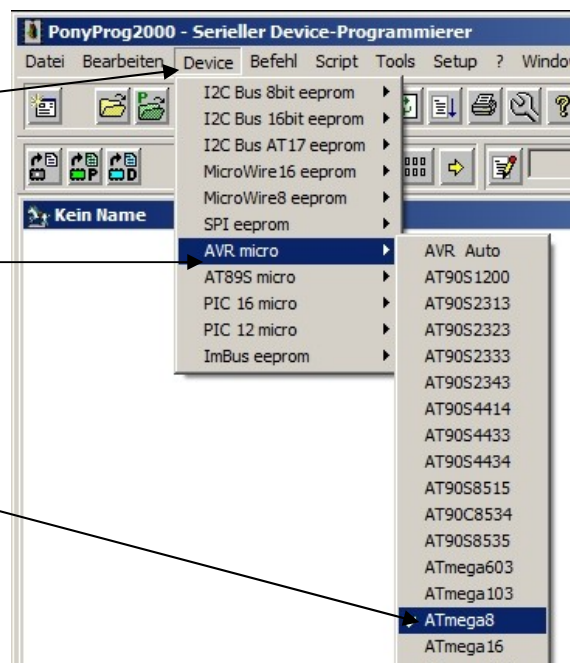


3.
Den zu programmierenden Baustein auswählen

3.1 >> Device

3.2 AVR Micro

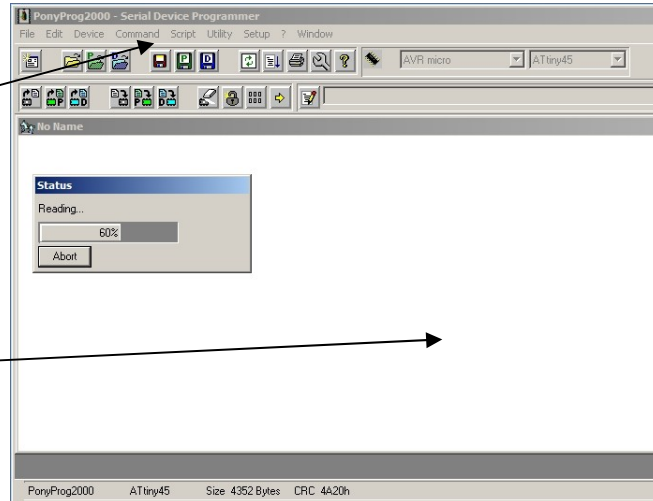
3.3 Controller (Beispiel...)



4. Erster Test, ob Verbindung zum Controller möglich ist.

>> Befehl
>> lese alles

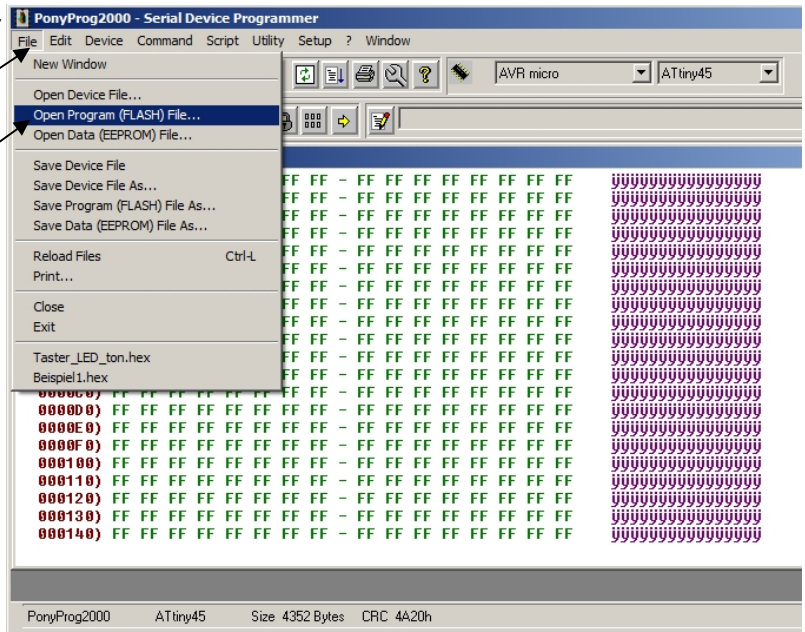
Danach sollten in diesem Fenster Inhalte des Flash-Speichers sichtbar werden.



5. Beschreiben des Flash-Controller

5.1 Eine xxx.hex oder xxx.bin Datei (die vorher in BASCOM erzeugt wurde, laden)

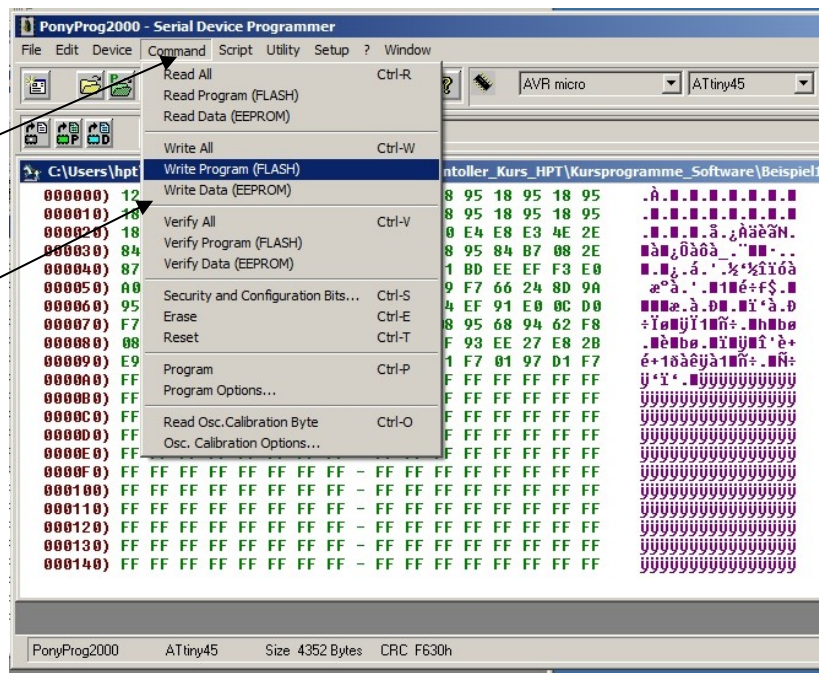
>> Datei
>> öffne (flash) Datei



5.2 Schreiben auf den Chip:

>> Befehl
>> Schreibe Programm (flash)

Dann sollte das Programm im Chip sein !



6.

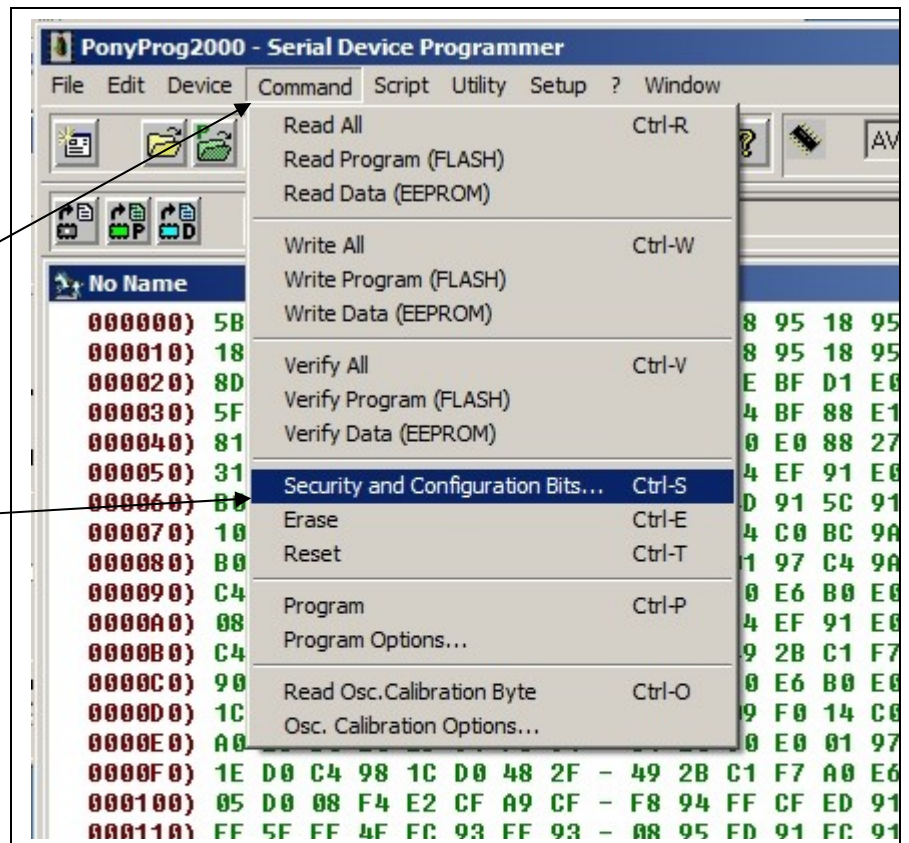
FUSEBITS

(ACHTUNG: Nur verstellen wenn man weiß wie und was !!! Zerstörung möglich !!!)

Kontrolle durch:

>> Befehl (Command)

>> Security und Configurations Bits



6.1

Beispiel: Einstellungen bei einem ATmega 8 mit 16 MHz

Bei PonyProg entspricht ein gesetztes Häkchen einer 0 im Datenblatt !



siehe Fuse Kalkulator (Simulator) <http://www.engbedded.com/fusecalc/>

3. Projekte

3.1 Beispiel : Ausgang – LED

```
' I/O Port umschalten
```

```
' © HPT 5.3.2012
```

```
$regfile = "m8def.dat"
```

```
'Die Anweisung bestimmt Controllertyp, hier AVR Mega 8
```

```
$crystal = 16000000
```

```
'Die Frequenz des verwendeten Quarzes
```

```
Config PinD.5 = Output
```

```
'Pin 5 von Port D wird als Ausgang konfiguriert
```

```
Anfang:
```

```
PortD.5 = 1
```

```
'Pin wird auf High, also 5V geschaltet
```

```
Waitms 100
```

```
PortD.5 = 0
```

```
'Pin wird auf Low, also 0V geschaltet
```

```
Waitms 100
```

```
Goto Anfang
```

```
End
```

Erweitern sie selbstständig auf: Led-Blitzer; Led binär zählen; Lauflicht; Einfache Ampel

3.2. Beispiel: Testen von Eingängen und Ausgängen

```
'Evaluationsboard V 2.01
```

```
'HPT 5.3.2012
```

```
$regfile = "m8def.dat"
```

```
'Controllerauswahl
```

```
$crystal = 16000000
```

```
'Quarzfrequenz
```

```
Config Portd = Output
```

```
'PinD.5=LED1, PinD.6=LED2, PinD.7=Summer
```

```
Config Pind.2 = Input
```

```
'=Taster 1
```

```
Config Pind.3 = Input
```

```
'=Taster 2
```

```
Config Pind.4 = Input
```

```
'=Taster 3
```

```
do
```

```
  If Pind.2 = 1 Then Set Portd.5
```

```
  ' Taster 1 gedrückt = LED1
```

```
  If Pind.3 = 1 Then Toggle Portd.6
```

```
  ' Taster 2 gedrückt = LED2 ein - aus
```

```
  If Pind.4 = 1 Then Sound Portd.7 , 200 , 450
```

```
  ' pin, Länge, Periodendauer.
```

```
  Waitms 250
```

```
  Reset Portd.5
```

```
  ' Port zurücksetzen
```

```
loop
```

```
End
```

```
'Programm-Ende
```

3.3 Beispiel : LCD

```
'Display Anschlüsse auf Zusatzprint von hpt über Stecker J4
```

```
Config Lcdpin=Pin,Db4 = Portb.3,Db5 = Portb.2,Db6 = Portc.2,Db7 = Portc.1,E = Portb.4,Rs = Portb.5
```

```
Config Lcd = 16 * 2
```

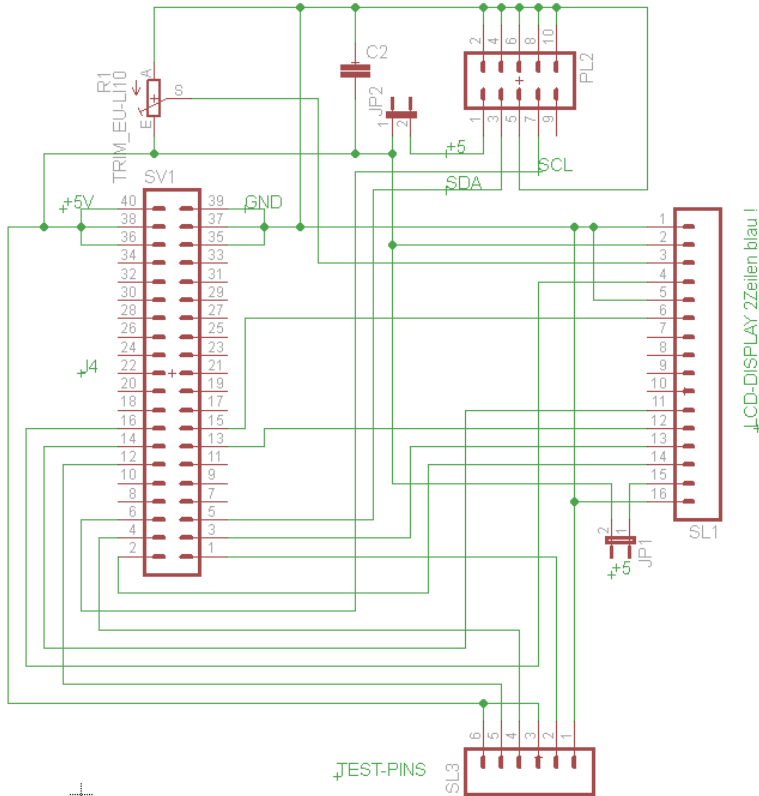
```
Cursor Off Noblink
```

```
Cls
```

```
Lcd " Test"
```

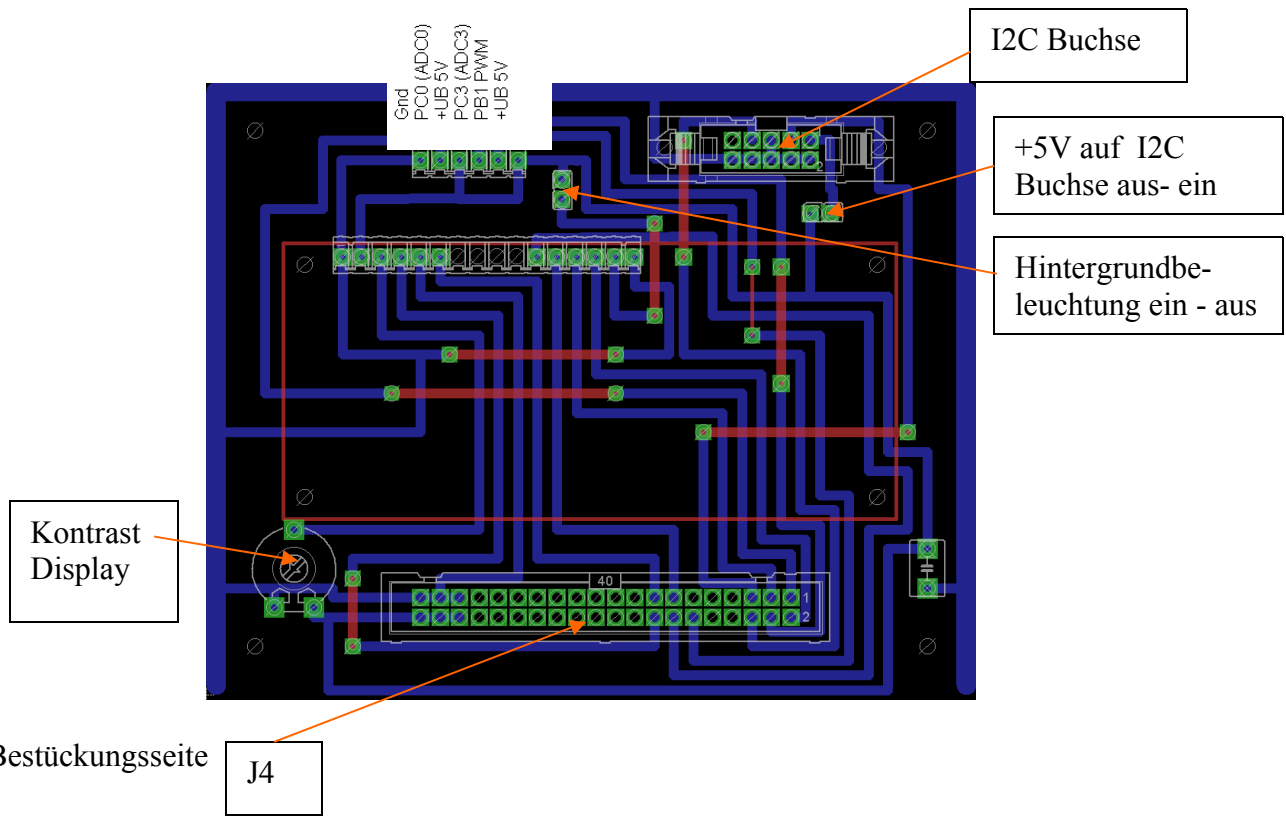
4. Erweiterungen

2 zeiliges Display (blau) - Erweiterung zu Entwicklungsboard mit I2C Buchse



Display-Anschlüsse:
Pin Disp. // J4

1 GND	>>	Gnd
2 +5V	>>	+5V
3 Vo	>>	+Helligk.
4 (RS)	>>	16 (PB.5)
5 (R/W)	>>	GND
6 (E)	>>	15(PB.4)
11 (DB4)	>>	14(PB.3)
12 (DB5)	>>	13(PB.2)
13 (DB6)	>>	3 (PC.2)
14 (DB7)	>>	2 (PC.1)
15 (L-)	>>	GND
16 (L+)	>>	+Hinterg.



5. Anhang

Variable:

Variable müssen am Anfang eines Programmes mit DIM initialisiert werden. Sie sollten gut überdacht (Geschwindigkeit und Größe) und ausgewählt werden.

Folgende Typen stehen zur Verfügung:

Typ	Wertebereich	Speicherbedarf
Bit	0 /1	1 Bit
Byte	0...255	1 Byte
Integer	-32768....+32767	2 Byte
Word	0...65535	2 Byte
Long	-2147483648.....2147483647	4 Byte
Single	$1,5 * 10^{-45}$ $3,4 * 10^{48}$	4 Byte
Double	$5 * 10^{-324}$ $1,7 * 10^{308}$	8 Byte
String * n	1.....253 Zeichen	Max. 254 bytes

Folgende Anweisungen werden von BASCOM unterstützt:

Bedingungen und Strukturen:

IF, THEN, ELSE, ELSEIF, END IF, DO, LOOP, WHILE, WEND, UNTIL, EXIT DO, EXIT WHILE, FOR, NEXT, TO, DOWNT0, STEP, EXIT FOR, ON .. GOTO/GOSUB, SELECT, CASE.

Eingabe and Ausgabe:

PRINT, INPUT, INKEY, PRINTEX, INPUTEX, LCD, UPPERLINE, LOWERLINE, DISPLAY ON/OFF, CURSOR ON/OFF/BLINK/NOBLINK, HOME, LOCATE, SHIFTLCD LEFT/RIGHT, SHIFTCURSOR LEFT/RIGHT, CLS, DEFLCDCHAR, WAITKEY, INPUTBIN, PRINTBIN, LCDHEX, OPEN, CLOSE, DEBOUNCE, SHIFTLIN, SHIFTOUT.

Numerische Funktionen

AND, OR, XOR, INC, DEC, MOD, NOT, ABS, BCD.

I2C

I2CSTART, I2CSTOP, I2CWBYTE, I2CRBYTE, I2CSEND and I2CRECEIVE.

1WIRE

1WWRITE, 1WREAD, 1WRESET.

SPI

SPIINIT, SPIIN, SPIOUT.

Interrupt Programmierung

ON INT0/INT1/TIMER0/TIMER1/SERIAL, RETURN, ENABLE, DISABLE, PRIORITY SET/RESET, COUNTERx, CAPTUREx, INTERRUPTS, CONFIG, START, LOAD.

Bit Verarbeitung

SET, RESET, ROTATE, BITWAIT.

Variablen

DIM, BIT, BYTE, INTEGER, WORD, LONG, SINGLE, STRING, DEFBIT, DEFBYTE, DEFINT, DEFWORD.

Verschiedenes

REM, ', SWAP, END, STOP, CONST, DELAY, WAIT, WAITMS, GOTO, GOSUB, POWERDOWN, IDLE, DECLARE, CALL, SUB, END SUB, MAKEDEC, MAKEBCD, INP, OUT, ALIAS, DIM, ERASE, DATA, READ, RESTORE, INCR, DECR, PEEK, POKE, CPEEK, GETRC5.

Compiler Anweisungen

\$INCLUDE, \$NOINIT, \$BAUD and \$CRYSTAL, \$OBJ, \$SERIALINPUT, \$SERIALOUTPUT, \$ROMSTART, \$RAMSIZE, \$RAMSTART, \$MONSTART, \$IRAMSTART, \$DEFAULT XRAM, \$ASM-\$END ASM, \$LCD.

Bedingte Kompilierung

#IF, #ELSE, #ENDIF

String Verarbeitung

STRING, SPACE, LEFT, RIGHT, MID, VAL, HEXVAL, LEN, STR, HEX, LCASE, UCASE

Tipps für Rechenoperationen:

BASCOM kann nur eine Rechenoperation pro Zeile verarbeiten !!!

a = b * 2 + c

funktioniert so nicht.

Stattdessen müssen die Operationen aufgeteilt werden:

a = b * 2

a = a + c

6. Links:

Bascom AVR (Gratisdownload)	http://www.mcselec.com/
PONYPROG (Gratisdownload)	http://www.lancos.com/prog.html
Atmega8 (Datenblatt)	http://www.atmel.com/Images/doc2486.pdf
Fuse Kalkulator (Simulator)	http://www.engbedded.com/fusecalc/
Atmel Studio5	http://www.atmel.com/microsite/avr_studio_5/default.aspx
Bascom Forum	http://bascom-forum.de/forum.php
Grundlegendes AVR-Tutorial	http://www.rn-wissen.de/index.php/AVR-Einstieg_leicht_gemacht
AVR-Kurs	http://halvar.at/elektronik/kleiner_bascom_avr_kurs/

Bücher:

Einfacher Einstieg in die Elektronik mit AVR-Mikrocontroller und Bascom
(Stefan Hoffmann) ISBN 978-3-8391-8430-1

Programmieren der AVR Risc Mikrocontroller
(Claus Kühnel) ISBN 3 –907857-04-6

Basiskurs Bascom-AVR
(Burkhard Kainka) ISBN 978-3-89576-238-3

Videos:

Elektronikboard.de <http://www.youtube.com/user/DasElektronikBoard/videos>