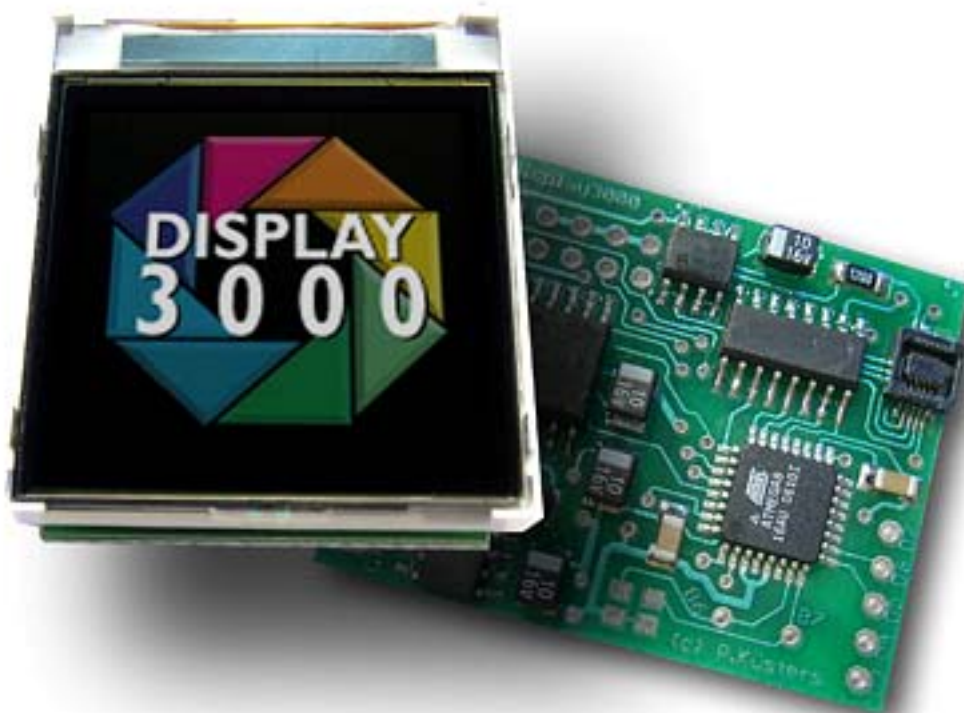


# Zusatzhandbuch für das Komplettmodul D04I mit ATMega8 Prozessor

V 1.5  
6. Oktober 2006



© 2006 by Peter Küsters

Dieses Dokument ist urheberrechtlich geschützt. Es ist nicht gestattet, dieses Dokument zu verändern und komplett oder Teile daraus ohne schriftliche Genehmigung von uns weiterzugeben, es zu veröffentlichen; es als Download zur Verfügung zu stellen oder den Inhalt anderweitig anderen Personen zur Verfügung zu stellen. Zuwiderhandlungen werden verfolgt.

Herzlichen Glückwunsch zum Erwerb des ATmega8-Moduls.

Dieses Modul beinhaltet neben der Ansteuerungselektronik für das Farbdisplay inkl. Power-Booster Technologie auch noch einen ATmega8 Prozessor sowie eine RS-232 Schnittstelle.

Dieses Modul erlaubt es Ihnen, die gesamte Ansteuerungselektronik inkl. Display auf kleinstem Raum unterzubringen.

Sie müssen diesen Bausatz noch vervollständigen, indem Sie das Display aufstecken sowie die notwendigen Kabel und Steckverbinder anlöten.

Wird das Displaymodul fest eingebaut, sollte es kein Problem geben (sofern Ihr Fertigprojekt nicht geschüttelt o. ä. wird). Beim Basteln wird das Modul zwangsläufig häufig bewegt und diese Belastung könnte auf die Dauer die Steckverbindung schädigen. Daher schlagen wir vor, dass Sie das Display am unteren Rand mit einem Tropfen Klebstoff oder einem Stück doppelseitigem Klebeband auf der Platine fixieren.

Das Display lässt sich jedoch nur dann sicher wieder entfernen, wenn Sie es nur am Kunststoffrand festkleben, Kleben Sie es jedoch auf der Rückseite fest, dann wird u.U. die Rückfolie des Displays beim Abziehen von der Klebestelle beschädigt.

Achtung: durch ein zu starkes Drücken auf die Frontscheibe des Displays kann das Display zerbrechen.

**Diese Anleitung zeigt Ihnen lediglich die Anschlussbelegung des Boards und gibt ein paar Tipps zu diesem Board. Zur eigentlichen Ansteuerung des Farbdisplays verweisen wir auf die separate Anleitung.**

Bitte haben Sie Verständnis, wenn wir keine Anlaufstation für Fragen zur generellen Programmierung dieses Prozessors sein können. Wir verweisen hier auf das umfangreiche Datenblatt des Prozessors (in Englisch) sowie die diversen Foren im Internet.

**ACHTUNG:**

- 1) Stecken Sie niemals das Display auf oder nehmen es ab, solange Spannung anliegt.
- 2) Stecken Sie das Display immer richtig herum auf (siehe Abbildungen). Wenn Sie das Display verkehrt herum aufstecken, wird es beim Anschluss an die Versorgungsspannungen unweigerlich zerstört.

**Port-Nutzung dieses Moduls – Änderung der Beispielprogramme**

Unsere Beispielprogramme sind für alle Module identisch, allerdings ist die Nutzung der Ports nicht immer die gleiche. **Daher müssen Sie für dieses Modul D041 den Anfang der Programme ändern - so wie in der rechten Box abgebildet:**

Display Verbindung	Port am integrierten Mikrocontroller	Die Bascom-Beispielprogramme sollten am Anfang folgende Zeilen zeigen:
SC	C.0	<code>Const Rs = 2</code>
SD	C.1	<code>Const CS = 3</code>
RS	C.2	<code>Const Sdata = 1</code>
CS	C.3	<code>Const Sclk = 0</code>



## Lieferumfang

### Sie bekommen geliefert:

- I x Bestückte Platine mit Prozessor ATmega 8 etc.
- I x Display
- I x Pfostenstecker (2x5, RM 2,54)
- Software, Dokumentation

## Spannungsversorgung

Die vorliegende Platine ist mit 2 Spannungsreglern ausgestattet um einen einfachen Betrieb zu ermöglichen und um eine Beschädigung des empfindlichen Displays auszuschließen.

Sie können an den Versorgungseingang [auf der Platine bezeichnet (+) und (G)] eine Gleichspannung von 5 bis 20 Volt anlegen. Der Prozessor und die RS-232 Schnittstelle wird mit 5 Volt betrieben, die Displayelektronik mit 3 Volt. Die beiden Spannungsregler sind Very-Low-Drop-Regler, so dass Sie tatsächlich auch nur 5 Volt anlegen brauchen (und nicht, wie bei den üblichen 7805 etc. mind. 6,5 Volt, um eine 5 Volt Ausgangsspannung zu erhalten).

### **ACHTUNG:**

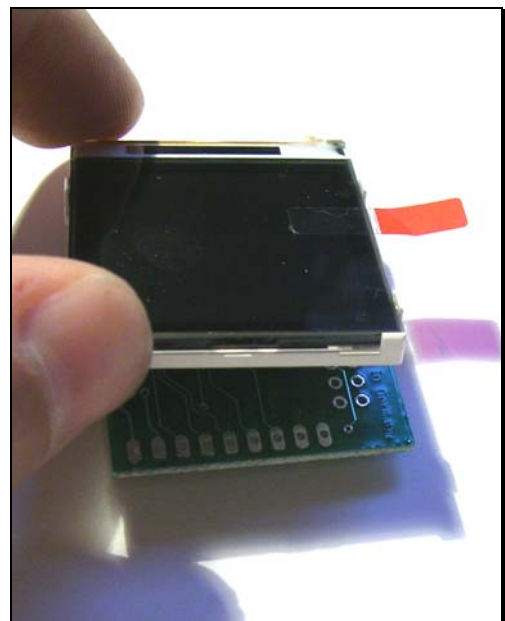
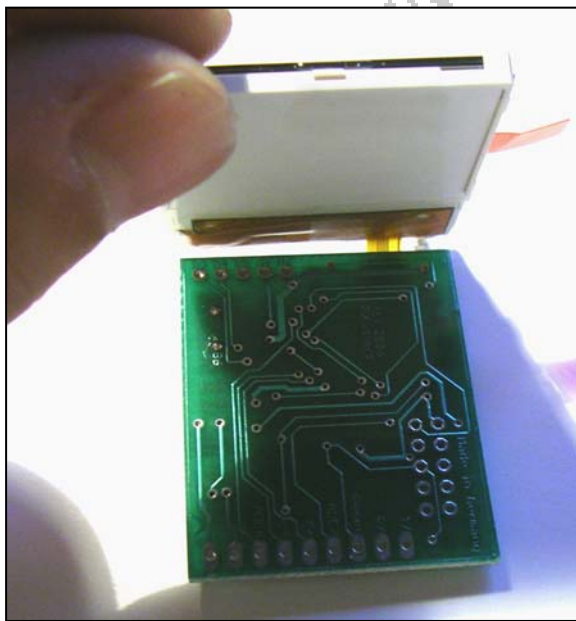
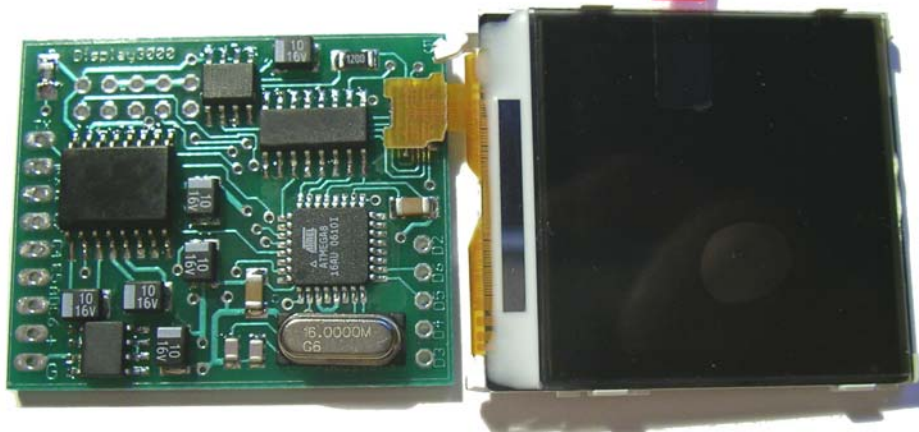
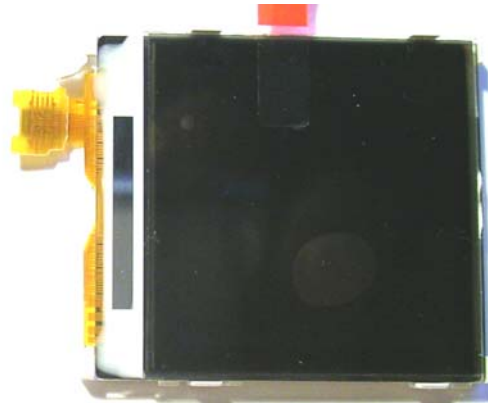
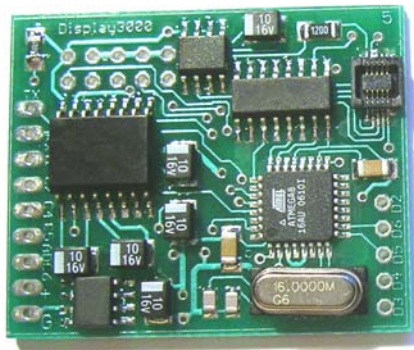
Der 5-Volt Spannungsregler kann 150mA liefern und wird bereits mit ca. 40mA belastet (aus diesem Stromzweig wird auch die ca. 8-Volt Spannung für die Displaybeleuchtung gewonnen). Daher ist es anzuraten, dass Sie von den Steckern des Moduls keine größeren Ströme anfordern. Sollte dies unvermeidlich sein, bietet es sich an, den integrierten 5-Volt-Spannungsregler (der erste Regler rechts neben dem Vcc-Anschluss) zu überbrücken und am Spannungseingang Vcc des Moduls genau 5,0 Volt anzulegen (die dann extern von einem Spannungsregler geliefert werden).

Legen Sie bei überbrücktem 5-Volt-Spannungsregler niemals eine höhere Spannung als 5 Volt an – eine Zerstörung der Displaybeleuchtung und evtl. des Prozessors kann die Folge sein.

**Teilweise werden auf den folgenden Seiten noch Fotos einer älteren Platinenrevision gezeigt. Die Ihnen vorliegende Platine unterscheidet sich jedoch lediglich durch andere Bauteile – die Lage der Löt pads und die anderen techn. Details sind identisch.**

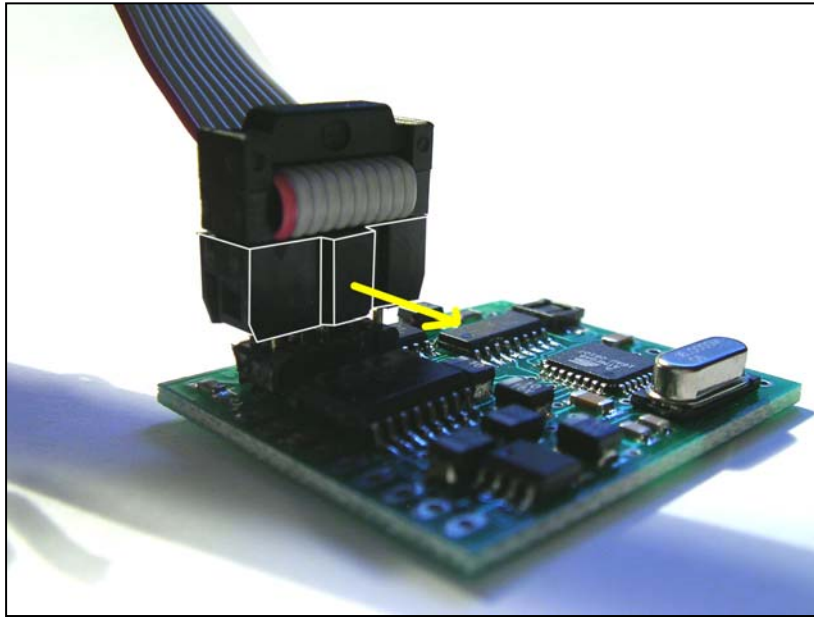
## Inbetriebnahme

Nehmen Sie das Display aus der Verpackung und legen es - wie hier zu sehen - neben die Platine und stecken dann den Stecker auf. Nun klappen Sie das Display vorsichtig nach vorne um. Beim Ablösen des Displays gehen Sie bitte in umgekehrter Reihenfolge vor. D.h. erst umklappen, dann den Stecker abziehen.



**TIPP:** Sie können das Display natürlich mit doppelseitiger Klebefolie o.ä. am Rücken der Platine befestigen. Dies ist sicher die einfachste und haltbarste Fixierung, aber Achtung: Sie können das Display dann nicht mehr entfernen – beim Versuch es vom Klebeband abzuziehen, werden Sie die Folie unter dem Display beschädigen und es somit zerstören.

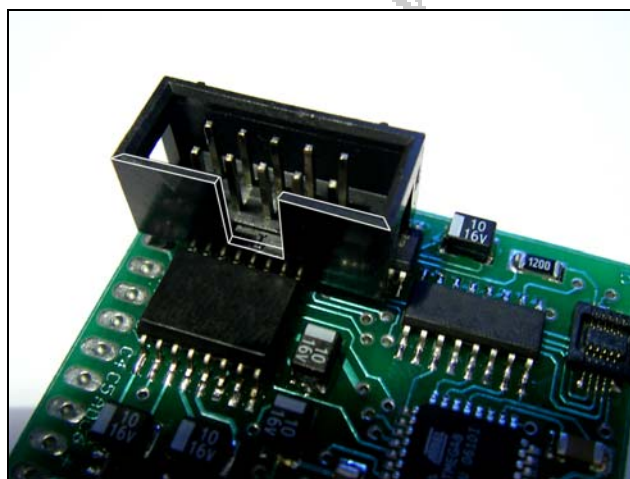
Legen Sie nun eine Spannung von 5 bis 20 Volt an die Eingänge + und G (G = Masse). Der Prozessor ist zu Testzwecken bereits vorprogrammiert und zeigt Ihnen auf dem Bildschirm ein Willkommen sowie den Status der verfügbaren Ports an.



Um das Board zu programmieren, brauchen Sie einen sog. ISP-Programmer, der nicht Bestandteil des Lieferumfangs ist. Dieser ISP-Programmer verbindet Ihren PC mit unserem D041 Board. Hierfür ist der mitgelieferte 2x5 Pfostenstecker einzulöten.

Alternativ (siehe Foto unten) können Sie hier auch den üblichen Wannenstecker einlöten, allerdings würde dessen Kunststoffwanne teilweise einen Chip

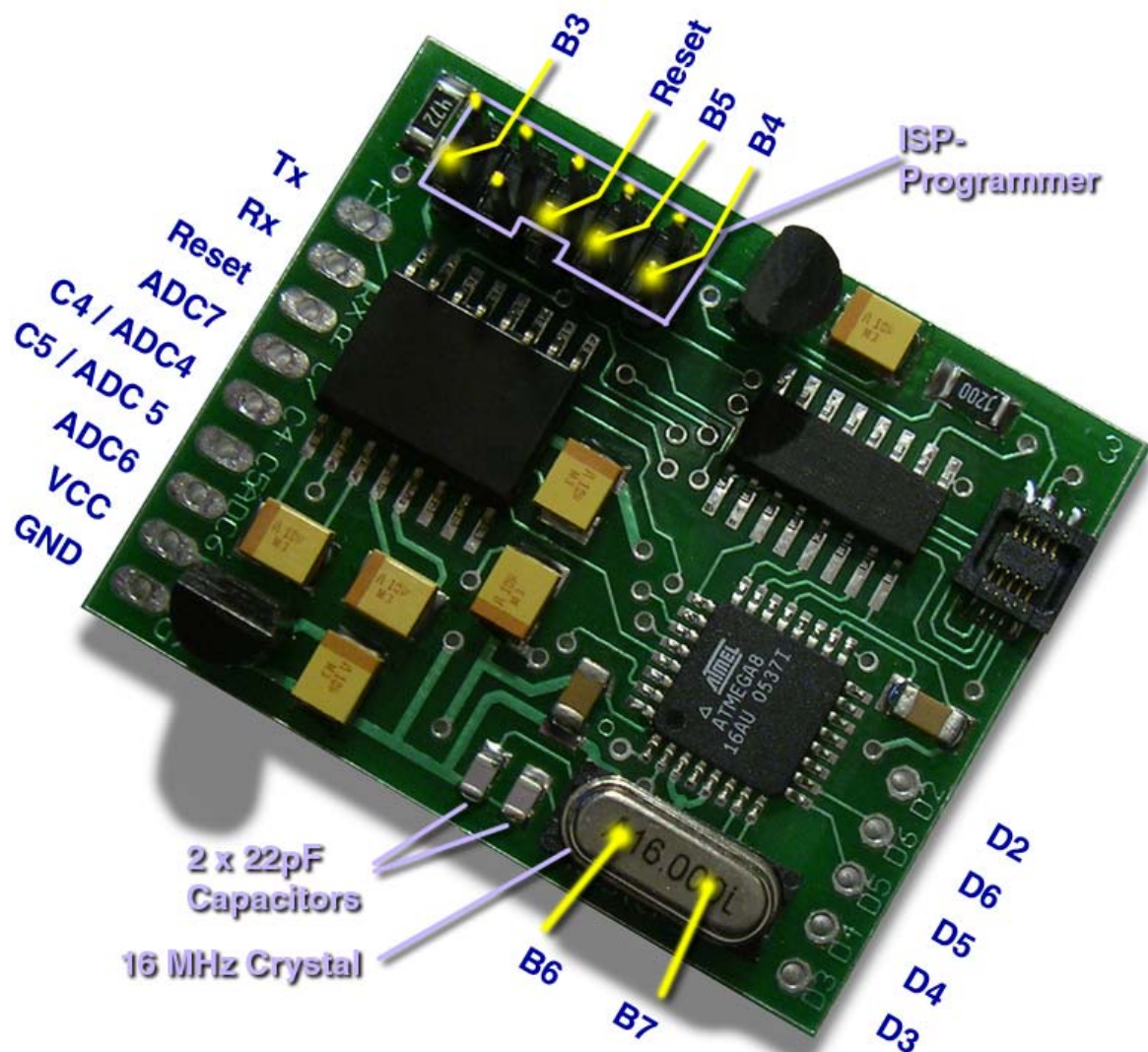
verdecken. Wir empfehlen daher einen Pfostenstecker. Achten Sie dann darauf, dass der Stecker des Programmers richtig herum eingesteckt wird: **Die Nase des Steckers muss immer Richtung Controller zeigen (siehe Foto oben).**



alternative Möglichkeit: Wannenstecker



## Pinbelegung des Moduls:



Die Anschlussbelegung der einzelnen Löt pads ist im obigen Foto zu ersehen und zudem auch auf der Platine nochmals aufgebracht (Das Foto zeigt ein älteres Modell, beim neuen Modell liegen die Pins unverändert an der gleichen Stelle).

Aufgrund des knappen Platzes haben wir nur einige Ports des Prozessors herausgeführt. Dies sind D2, D3, D4, D5, D6 sowie mit C4 und C5 zwei Ports, die sowohl als Digital als auch als Analog-Eingänge fungieren können. Weiterhin stehen Ihnen 2 reine Analog-Ports mit ADC6 und ADC7 zur Verfügung.

Sofern Sie keinen Quarz nutzen möchten, sind diese beiden Pins ebenfalls als Port zu nutzen: Port B6 und B7. Sobald Sie einen Quarz einlöten und die Fuses auf externen Quarz umstellen, stehen die Ports B6 und B7 nicht mehr zur Verfügung.

Zusätzliche steht es Ihnen frei, die Ports B3, B4 und B5 am ISP-Stecker abzugreifen, so dass Sie trotz des kleinen Boards insgesamt bis zu 14 Ein-/Ausgänge zur Verfügung haben, 4 davon auch für die Auswertung von Analogsignalen.

Wenn Sie das mit Reset beschriftete Lötpad an Masse ziehen, wird im Mikrocontroller ein Reset ausgelöst und das dort abgelegte Programm neu gestartet.

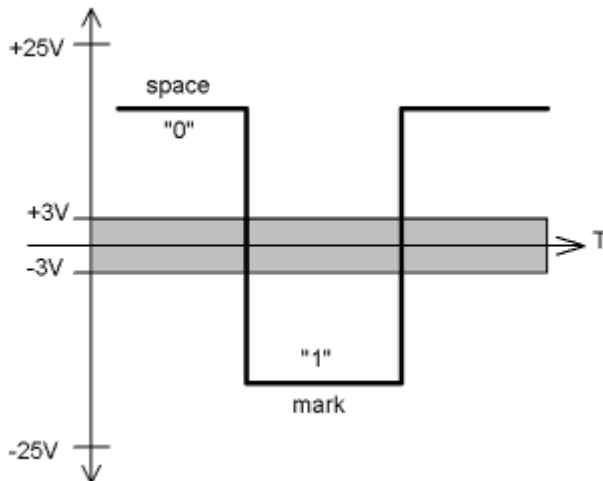
Der Prozessor ist aktuell auf 8 MHz Takt eingestellt. Auf Wunsch können Sie die Geschwindigkeit verdoppeln, in dem Sie an die gekennzeichneten Stellen einen 16 MHz Quarz sowie 2x 22pF Keramikkondensatoren einlöten. Vergessen Sie nicht, das entsprechende Fusebit für die Geschwindigkeit umzuprogrammieren. Lieferzustand ist **0100** (8 MHz intern). Programmieren Sie mit externem 16MHz Quarz (**und nur dann, wenn dieser eingelötet wurde**): **1111**; alle anderen Einstellungen oder ein fehlender Quarz führen u.U. zu einem nicht funktionierenden Board. **Spielen Sie nicht mit den Einstellungen der Fuses herum!**

**Ändern Sie NIEMALS die Fuse für C6/Reset.** Hiermit können Sie zwar den Reset-Pin als weiteren Port benutzen, dann jedoch nie wieder eine Neuprogrammierung des Controllers durchführen (dafür wird Reset benötigt). Auch ist diese Einstellung nicht mehr rückgängig zu machen (denn auch dafür würde Reset benötigt).

## RS232

RS-232 ist ein Kommunikationsprotokoll welches die Daten Bit für Bit auf 2 Signal-Level sendet:

- eine Spannung von -3 bis -25 Volt entspricht einer logischen Eins (1)
- eine Spannung von +3 bis +25 Volt entspricht einer logischen Null (0)



Wie das obige Bild zeigt, ist der Spannungsbereich von -3 bis +3 Volt undefiniert. In der Praxis ist dies jedoch nicht so. Meistens werden alle Spannungen oberhalb von +2,5 Volt als logische Eins angesehen und alle Spannungen darunter als logisch Null.

Die elektronische Spezifikation der RS-232 Verbindung ist robust – all Ausgänge müssen einem Kurzschluss widerstehen und alle Eingänge müssen ein Schmitt-Trigger Verhalten haben. Dies lässt eine RS-232-Schnittstelle am PC wesentlich weniger anfällig sein, als z.B. die Parallelschnittstelle, welche mit TTL-Level arbeitet.

RS-232 ist ein asynchrones Protokoll, d.h. es wird keine separate Clock-Information mit übertragen. Sowohl die Sende-, wie auch die Empfangsstation müssen die Geschwindigkeit kennen (als Baud-Rate bezeichnet).

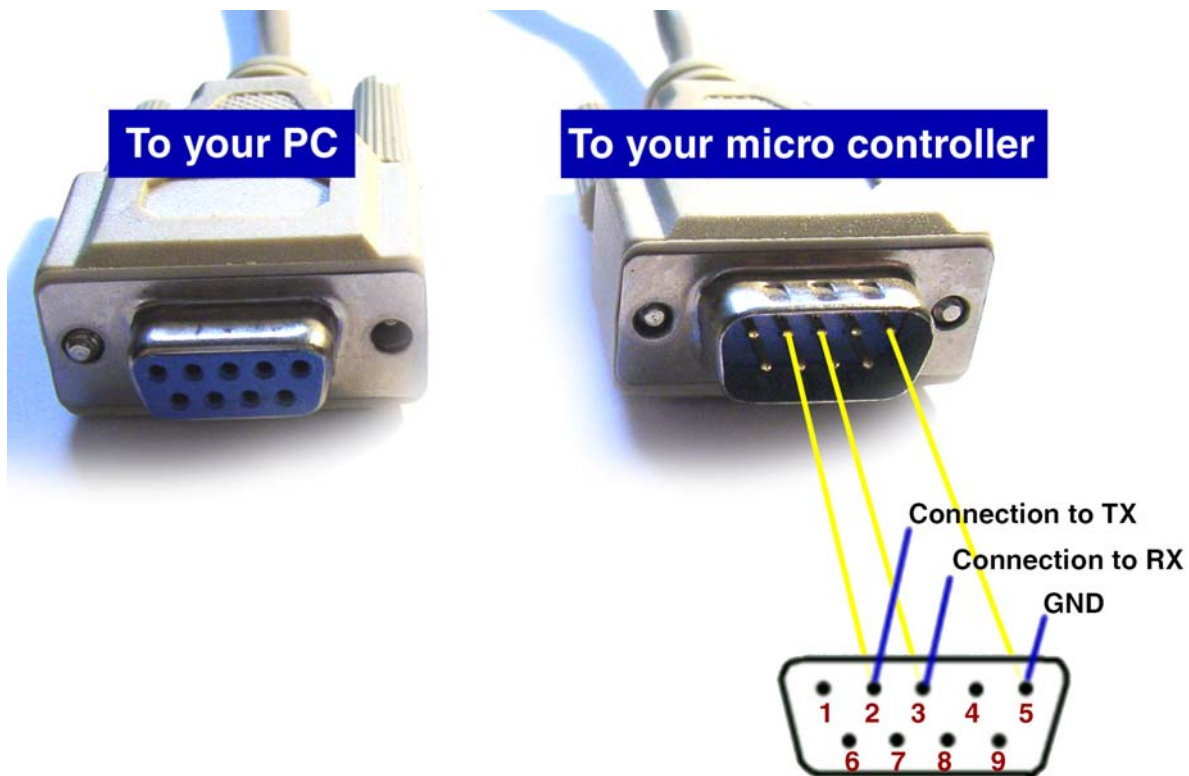
Wir nutzen hier die drei wichtigsten Signale des RS-232-Systems:

- RxD : receive data, Pin 2 am DB9 Stecker
- TxD : transmit data, Pin 3
- Masse, Pin 5

Diese Pin-Nummern entsprechen der Nummerierung am normalen DB9-Stecker am PC oder Laptop (siehe auch die Fotos auf der nächsten Seite).

Wenn Sie eine Verbindung zwischen PC und Mikrocontroller aufbauen möchten, so benötigen Sie ein übliches serielles Kabel (KEIN sog. Null-Modemkabel, da hier die Kabel gekreuzt werden) mit einem Stecker und einer Buchse. Die Seite mit der Buchse wird mit Ihrem PC verbunden, der Stecker mit dem Mikrocontroller. Das nachfolgende Bild zeigt Ihnen die notwendige Verbindung. Sollten Sie nur ein Nullmodem-Kabel zur Verfügung haben, dann müssen Sie gezeigten Verbindungen TX und RX tauschen, da diese beiden Leitungen innerhalb des Kabels getauscht sind.





Pin 2 ist definiert als Empfangskanal des PCs, daher müssen Sie hier den Sendekanal (TX) des Mikrocontrollerboards anschließen. Pin 3 entspricht dem Sendekanal des PCs, dieser wird an den Eingang des Boards (RX) angeschlossen.

Der ATmega8 bietet ein RS-232 Interface (mehr dazu im ATmega8-Datenblatt). Das Interface steht an den Ports D.0 und D.1 zur Verfügung. Diese beiden Ports sind mit dem RS-232-Interface-Chip auf dem Board verbunden, denn der Mikrocontroller kennt nur den Pegel von 5 Volt. Wie Sie oben gesehen haben, sind für RS-232 andere Spannungen notwendig, die mit diesem Chip erzeugt werden. Wenn Sie die RS-232 Leitung des PCs direkt an den Mikrocontroller anschließen würden, würde dieser vermutlich zerstört werden. Der RS-232-Interface-Chip wiederum ist an die beiden Anschlüsse RX und TX angeschlossen.

Wenn Sie das RS-232 Interface des Boards nutzen möchten, hilft Ihnen evtl. das nachfolgende Beispiel. Die Schnittstelle ist auch sehr praktisch zum Debuggen eigener Software. Mittels des Print-Befehls können Sie jederzeit z.B. den Inhalt einer Variable ausgeben um zu kontrollieren, ob diese den erwarteten Wert enthält.

Die Ausgabe des Boards über die RS-232 Schnittstelle lassen Sie sich dann mittels eines Terminal-Programms anzeigen.

Unter MS-Windows<sup>®</sup> nutzen Sie z.B. das Programm Hyperterminal, in Bascom gibt es einen eingebauten Monitor etc.

Das nachfolgende kleine Programm gibt permanent einen String auf dem Schnittstellenausgang aus – damit können Sie schnell eine funktionsfähige Verbindung aufbauen.

```
\sample program RS232 output
$regfile = "m8def.dat"
$crystal = 8000000
$baud1 = 9600

Do
  Print "Hello world"
  Wait 1
Loop
End
```

### RS232 und die Taktfrequenz / Übertakten des Boards

Wenn Sie eine größere Datenmenge übertragen möchten, oder ihre Daten fehlerfrei ankommen sollen, dann sollten Sie wissen, dass die notwendige Frequenz zur passenden Baud-Rate vom Mikrocontroller durch Teilen der Taktfrequenz des Mikrocontrollers erreicht wird. Zwei Dinge sind wichtig zu wissen:

a) Der eingebaute interne Taktgeber des Controllers ist nicht sehr genau – die Frequenz schwankt im Übrigen auch noch je nach Umgebungstemperatur. Wenn also Ihr Board ohne externen Quarz betrieben wird, sind Übertragungsprobleme zu erwarten. Besser, Sie setzen einen Quarz ein – unsere Boards sind alle dafür vorbereitet.

b) Der übliche externe 16 MHz-Quarz ist nicht optimal, denn durch die Teilung wird keine 100% korrekte Taktfrequenz erreicht. Bei kleineren Baudraten ist dies noch nicht relevant, bei höheren Baudraten macht sich dies jedoch bemerkbar. Ein idealer Quarz wäre einer mit einer Frequenz von 14.7456 MHz oder 18.432 MHz.

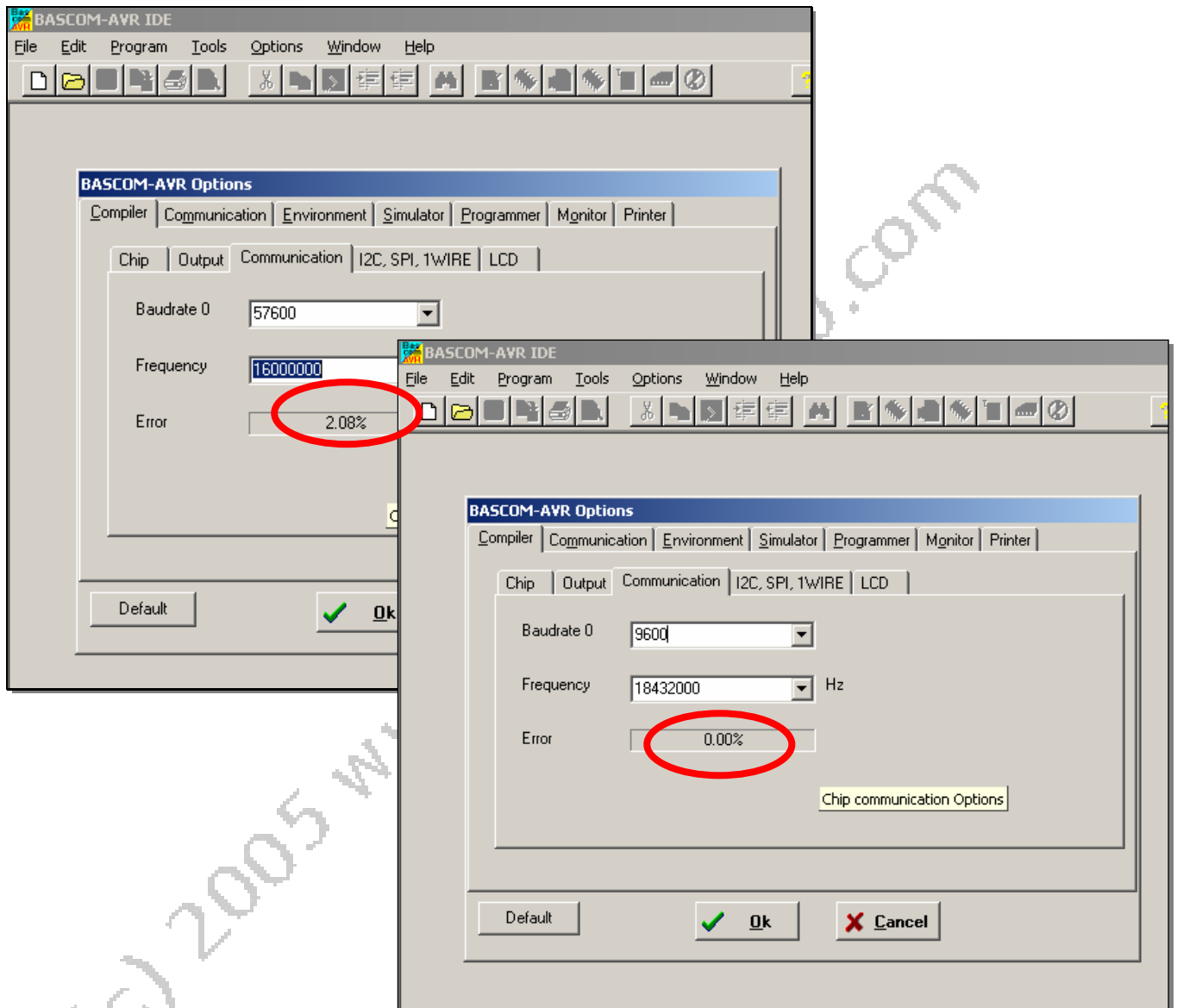
Alles über 16 MHz betreibt den Mikrocontroller jedoch über seiner Spezifikation von 16 MHz – d.h. Sie übertakten ihn. Normalerweise führt eine solch geringe Übertaktung noch zu keinem Problem, trotzdem geschieht dies immer auf eigenes Risiko. Übrigens treten evtl. Programmfehler als erstes beim Schreiben und Lesen des internen Eeproms auf. Der eigentliche Controller-Kern läuft oft auch mit über 20 MHz fehlerfrei – dann aber ist ein Zugriff auf das Eeprom nicht mehr zu empfehlen.

c) Sie müssen dem Compiler mitteilen, welche Taktfrequenz am Controller anliegt, sonst wird die Ermittlung des notwendigen Teilers für die Berechnung der Baudrate nicht korrekt durchgeführt. In Bascom führen Sie dies mit dem Befehl `$crystal = 8000000` am Anfang des Programms durch at the beginning (8000000 für 8 MHz; 16000000 für 16 MHz, 14745600 für 14.7456 MHz etc.).

Hier muss die exakte Baudrate eingegeben werden, ansonsten ist eine RS-232-Verbindung nicht möglich.

In Bascom® befindet sich übrigens ein Rechner, der Ihnen die Fehlerrate Ihrer gewählten Kombination aus Quarz und Baudrate anzeigt. Sie finden diesen unter dem Menü **Options / Compiler / Communications**.

Screenshots:



### In Kürze:

Da der interne 8 MHz Taktgeber nicht sehr akkurat ist, sollten Sie einen externen Quarz nutzen, wenn Sie RS-232 einsetzen möchten.

Ein 16 MHz-Quarz z.B. führt bei 9600 Baud zu einer Fehlerrate von 0,16%, was OK ist. Alles unter 1% Abweichung ist in der Regel akzeptabel.

14.7456 MHz oder 18.432 MHz führen zu 0,00% und sind die ideale Wahl für RS-232, leider gibt es diese Quarze nicht immer und in allen Ausführungen.

## Technische Daten Display-Modul-Bausätze:

### Artikel D041:

<b>Maße:</b>	40 x 35 mm ca. 8 mm hoch (mit Display)
<b>Versorgungsspannung:</b>	5 bis 20 Volt Gleichstrom, ca. 50 mA
<b>Prozessor:</b>	ATMega 8 8 KByte Programmspeicher 1 KByte RAM 512 Byte Eeprom max. 16 MHz Takt (Lieferumfang 8 MHz)
<b>Display:</b>	132x132 Pixel, 65.536 Farben Aktive diagonale Fläche: 1,5" (38mm)  Achtung! Es gibt 5 Modelle von diesem Display Im Zweifelsfall können Sie Displays günstig bei <a href="http://www.display3000.com">www.display3000.com</a> beziehen

**Der Prozessor wurde von uns gegenüber der Standardauslieferung des Herstellers ATMEL wie folgt umprogrammiert:**

**Fusebit:** Speed von 0001 (1MHz intern) **auf 0100 (8 MHz intern)**

**Fusebit: Preserve EEPROM** when chip erase

Eeprom und Programmspeicher wurden zu Testzwecken programmiert.

### Hersteller:

Speed IT up  
Inhaber Peter Küsters  
Wekeln 39  
47877 Willich  
Telefon: (0 21 54) 88 27 5-0  
Telefax: (0 21 54) 88 27 5-22

Weitere Informationen und Updates: [www.display3000.com](http://www.display3000.com)

Autor dieses Manuals: Peter Küsters.

© **aller Informationen: Peter Küsters**