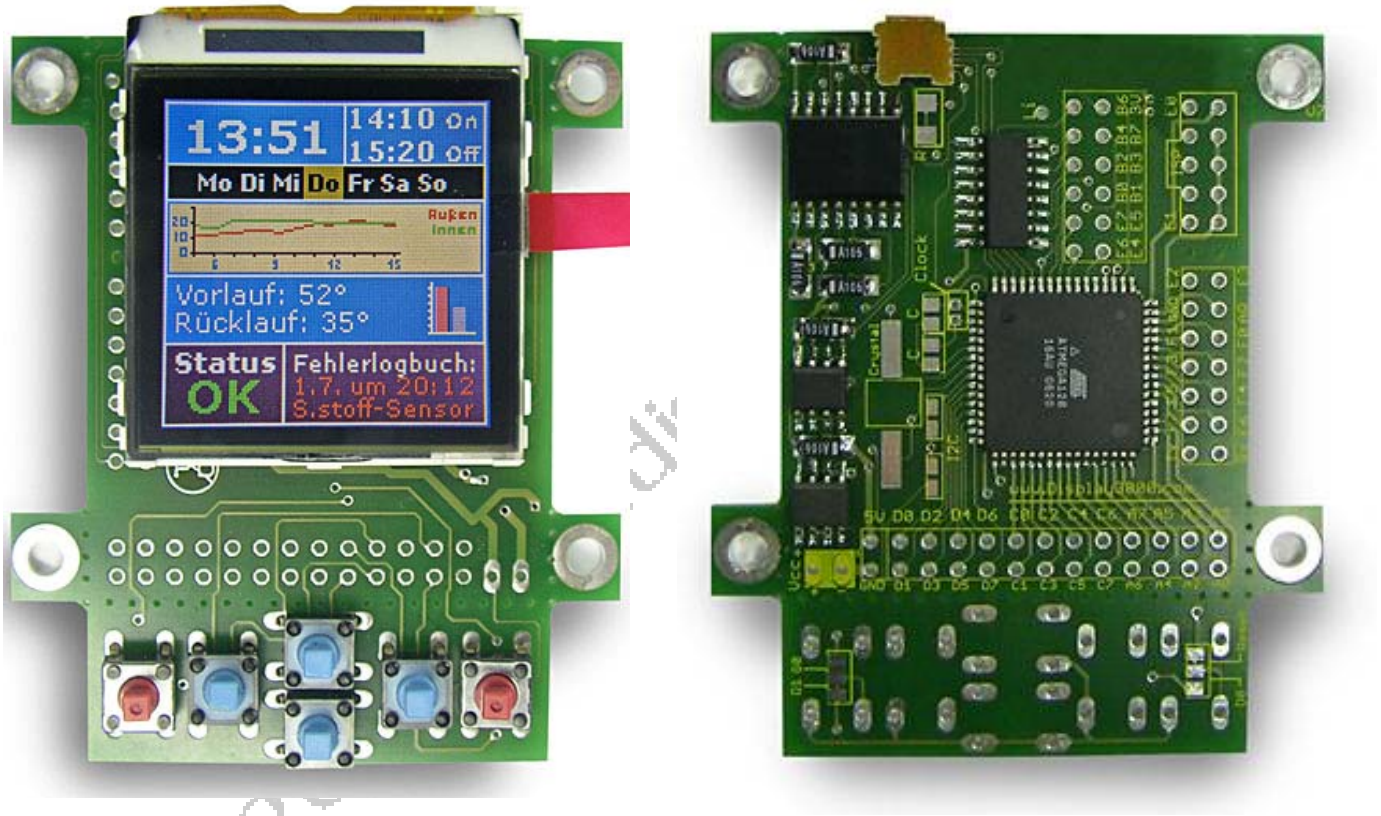
Zusatzhandbuch für das Komplettmodul

D062x

mit Atmel Mikrocontroller:
ATMega128 oder **ATMega2561** oder **AT90CAN128**

Version 2.3
Stand: 21. November 2007



Herzlichen Glückwunsch zum Erwerb des ATMega-Moduls D062x. Das „x“ steht für „Extended Version“ – das Kundenfeedback auf die Vorgängermodule hat zur ständigen Weiterentwicklung dieses Moduls geführt. Daher freuen wir uns immer über konstruktive Kritik und über positive Rückmeldungen von Ihnen.

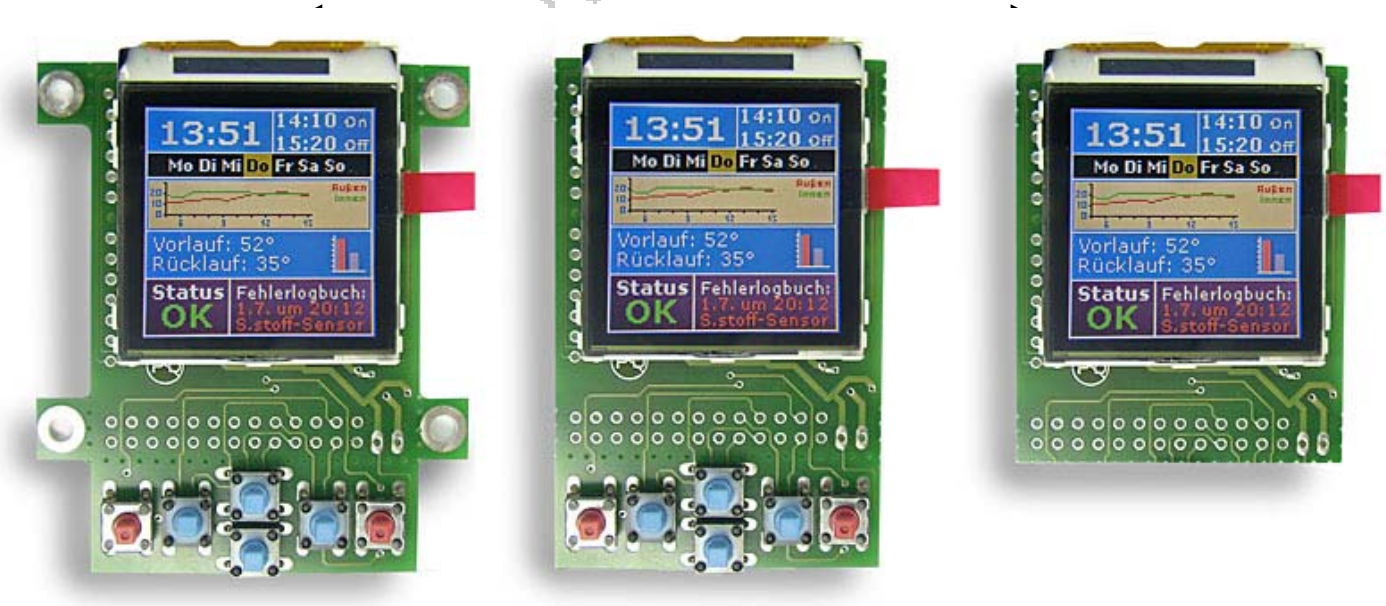
Dieses Modul beinhaltet neben einem Atmel Controller (je nach georderter Variante mit ATMega128, ATMega2561 oder AT90CAN128) noch die Ansteuerungselektronik für das Farbdisplay inkl. PowerBooster Technologie und eine RS-232 Schnittstelle. Dieses Modul erlaubt es Ihnen, die gesamte Ansteuerungselektronik inkl. Display auf kleinstem Raum unterzubringen. Wenn man sich nun vorstellt, dass dieses Modul vor 30 Jahren vermutlich eine Mondrakete hätte steuern können … na gut, vielleicht sollte man doch besser zwei Module nehmen ☺

Sie erkennen rechts und links vom Modul noch Halterungen mit Montagebohrungen. Diesen Rahmen können Sie jederzeit leicht entfernen. Das gleiche gilt für das Tastenfeld unterhalb des Displays – auch dieses kann von Ihnen leicht entfernt werden. Durch Entfernen aller Randbereiche ist das gesamte Modul kaum mehr größer wie das Display selbst. Die notwendigen Schritte zum Entfernen des Rahmens erfahren Sie ab Seite 34).

Errata: Bitte beachten Sie die Bemerkung auf Seite 9.

Übrigens: nachdem wir oft danach gefragt wurden, haben wir das Testprogramm, mit welchen das Modul ausliefern, mit auf die CD kopiert.

Sie müssen diesen Bausatz noch vervollständigen, indem Sie das Display aufstecken sowie die notwendigen Kabel und Steckverbinder anlöten. Haben Sie ein Modul D062x erworben, werden von Ihnen zudem noch die Taster eingesteckt und eingelötet.



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 einfachem Sanitär-Silikon 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.
Ein Warnung: durch ein zu starkes Drücken auf die Frontscheibe des Displays kann das Display zerbrechen.

Eine Anmerkung noch vorab, damit dies nicht zu Irritationen führt: Einige Fotos zeigen noch eine ältere Version der Platine für D062x – lassen Sie sich dadurch nicht irritieren. Alle wichtigen Aufnahmen wurden ersetzt. Wann immer Sie ein Bild des D062x ohne Montagerahmen sehen, zeigt dieses in der Regel das alte Modul.

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 das separate Programmierhandbuch.

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 am Modul anliegt.
2) Stecker 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.

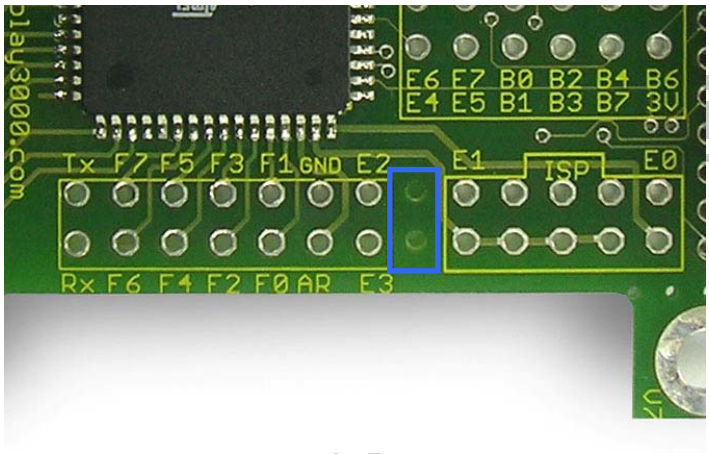
Lieferumfang

Sie bekommen geliefert:

- 1 x Modulplatine; alle Bauteile sind bereits aufgelötet
- 1 x Farbdisplay mit Connector-Platine
- 6 x Taster zum Aufstecken oder Einlöten
- Pfostenstecker zum Einlöten
- CD mit Beispielsoftware, Utility-Software und umfangreicher Dokumentation

Das Modul ist komplett gelötet. Lediglich um die Anschlussleisten müssen Sie sich selbst kümmern, denn jeder Anwender hat hier seine eigenen Wünsche. Bitte löten Sie als erstes die Pfostenstecker ein, alternativ können Sie natürlich auch Buchsenleisten oder auch direkt die notwendigen Kabel anlöten. Wenn Sie dieses Modul zum experimentieren nutzen möchten, so legen wir Ihnen unsere Experimentierplatine (z.B. P006) ans Herz. Hier stecken Sie das Hauptmodul D062 nur auf, und gelötet wird stattdessen auf der Experimentierplatine – d.h. Ihr Modul ist auch nach vielem Probieren noch wie neu.

Wichtig: Lesen Sie bitte VOR dem Einlöten der Pfostenstecker den folgenden Absatz und die nächste Seite.



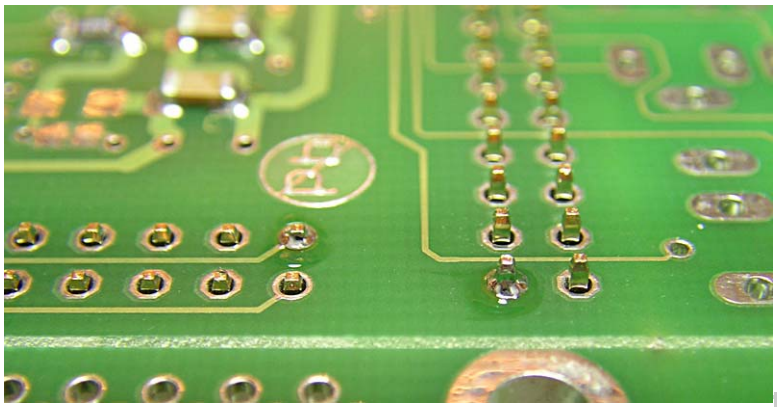
Wenn Sie, wie auf Seite 15 gezeigt, den ISP-Stecker Ihres Programmiergerätes direkt mit dem Controllermodul verbinden möchten, dann dürfen Sie keine komplette Steckerleiste einsetzen, sondern müssen diese vorher in einen Stecker mit 2x5 und einen Stecker mit 2x7 Kontakten trennen – zwischen den beiden Steckern muss dann ein Pad frei bleiben (blaues Rechteck), ansonsten hat der Kunststoffrahmen des ISP Steckers nicht genügend Platz und Sie können den Stecker nicht einstecken. Sie erkennen dies gut auf der Platine, da sich dort nur Bohrungen aber keine Lötpads befinden.

Sollten Sie zu den Menschen gehören, die ein Handbuch, wenn überhaupt, erst ganz zum Schluss lesen ☺, dann dürfen Sie sich nun darüber freuen, dass wir dies vorausgesehen haben: Die beiden Stifte, die Ihnen nun im Weg sind, konnten von Ihnen ja mangels Lötpad nicht festgelötet werden – ziehen Sie sie also einfach mit einer flachen Zange heraus.

Einlöten der Pfostenstecker

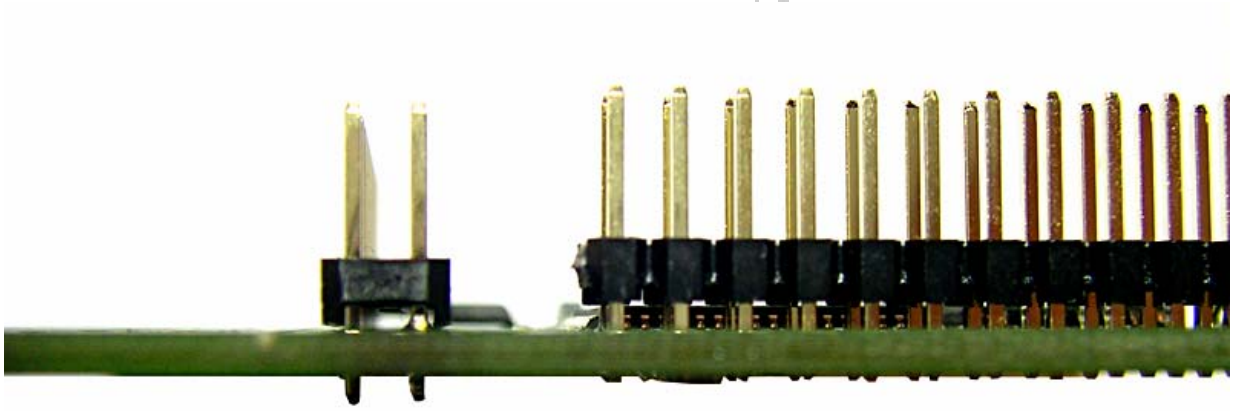
Die Lötpads der Pfostenstecker liegen unterhalb des Displays. Wir empfehlen daher, diese Stiftleisten nicht komplett in die Platine zu stecken, sondern lediglich ein wenig herausschauen zu lassen, um keinen Einfluss auf die Lage des Displays zu bekommen. Das gleiche gilt übrigens auch für das Einlöten von Kabeln.

Nachfolgend zwei Bilder zur Illustration:



Stiftleiste auf der rechten Seite: **So ist es falsch**. Diese wurde komplett durchgesteckt und dann gelötet. Dies würden wir nicht empfehlen.

Stiftleiste auf der linken Seite: **So ist es richtig.** Die Stiftleiste schaut lediglich ein wenig aus den Lötpads heraus.



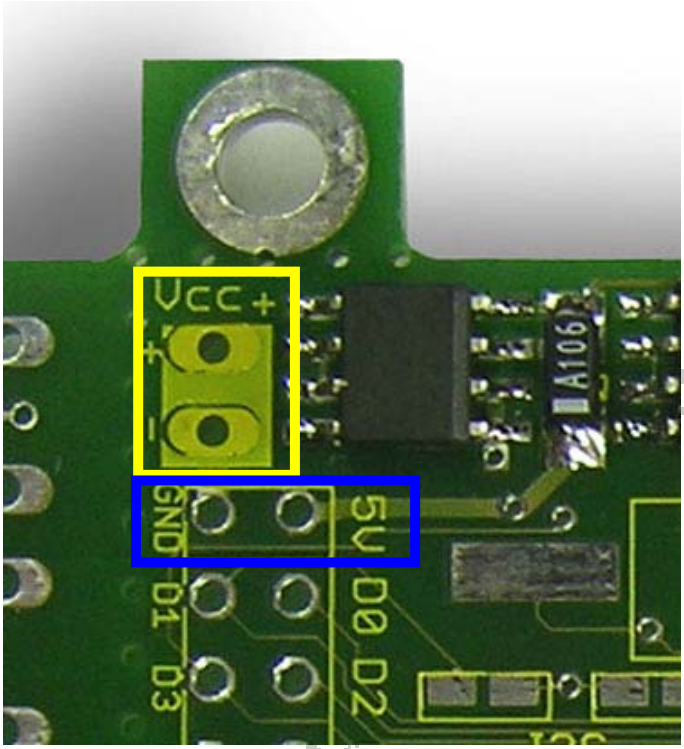
Oben: Eine Ansicht von der Seite. Links: zu weit durchgesteckt, man erkennt, wie weit die Lötstifte aus der Platine herausragen. Rechts: so ist es richtig.

Zum Einlöten einer Leiste empfehlen wir folgende Vorgehensweise:

a) Einen Leiste einstecken und Modul umdrehen (Lötpads also oben)
b) Platine leicht anheben, bis die Stifte nur noch ½ mm herausschauen (wie im oberen Bild links) und dann einen Stift einlöten.
c) Einen weiteren Stift an der gegenüberliegenden Seite festlöten.
d) Rechtwinkligen Sitz der Leiste prüfen (nach Augenschein reicht) und ggf. noch mal korrigieren.
e) Alle anderen Pins einlöten

Spannungsversorgung

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



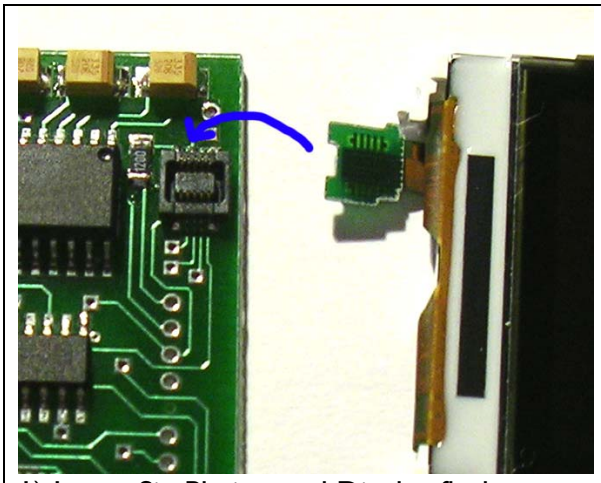
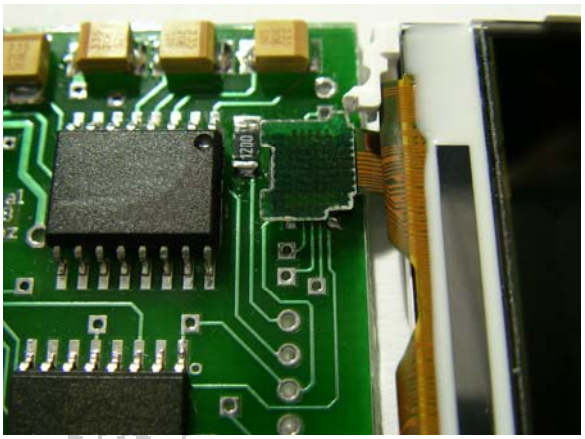
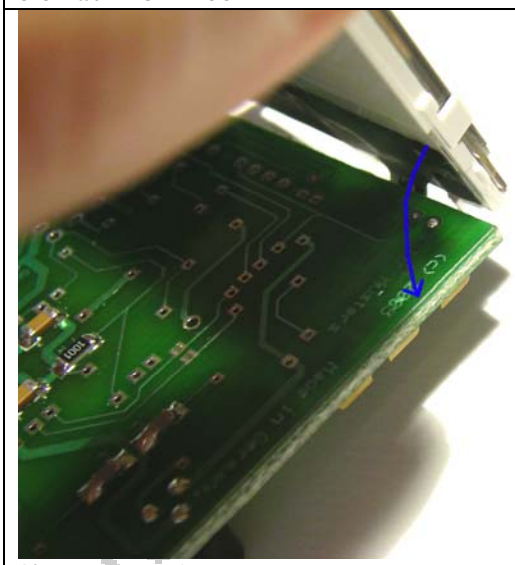
Sie können an den Versorgungseingang Vcc – GND (gelber Kasten) eine Gleichspannung von 4,5 bis 20 Volt anlegen. Der Prozessor und die RS-232 Schnittstelle wird mit 5 Volt betrieben, die Displayelektronik mit 3 Volt. Die Spannungsreglung wurde mit einer Spezialregelung ausgeführt, so dass Sie für den Betrieb 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 max. 160mA liefern und wird bereits mit ca. 50mA 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).
Alternative: Sie legen an die beiden Pads 5V und GND (blauer Kasten) eine Spannung von 5V (oder weniger) an.
Legen Sie dann aber niemals eine höhere Spannung als 5 Volt an – eine Zerstörung der Displaybeleuchtung und des Controllers kann die Folge sein.

Inbetriebnahme

1) Stecken Sie die Pfostenstecker in die vorgesehen Lötpads und verlöten Sie sie.
Die Stecker werden von der Bauteilseite aus eingesteckt (siehe auch Fotos weiter unten)

2) Verbinden Sie das Board mit dem Display wie im folgenden gezeigt und legen Sie dann eine Spannung an die beschrifteten Kontakte Vcc und GND an. Es erscheint das von uns bereits aufgespielte Testprogramm (siehe Foto auf Seite **Error! Bookmark not defined.**).

<table>
<tr>
<td> 
1) Legen Sie Platine und Display flach vor sich auf den Tisch</td>
<td> 
2) Drücken Sie den Display-Stecker auf das Gegenstück auf der Platine</td>
</tr>
<tr>
<td> 
3) Nehmen Sie vorsichtig die Platine in die linke Hand und das Display in die rechte Hand – dann klappen Sie das Display vorsichtig auf die andere Seite der Platine um</td>
<td> 
4) Fertig</td>
</tr>
</table>

Wenn Sie das Display wieder abnehmen möchten, führen Sie die obige Anleitung in umgekehrter Reihenfolge durch – falten Sie es immer erst zurück, bevor Sie den Stecker abziehen – ansonsten können Sie das Displaykabel beschädigen.

Der Prozessor ist zu Testwecken bereits vorprogrammiert und zeigt Ihnen auf dem Bildschirm (siehe Foto auf Seite **Error! Bookmark not defined.**) den Status der verfügbaren Ports an. Anmerkung: Port F4 bis F7 zeigen hier „0“, wenn das Modul für einen JTAG-Zugriff vorbereitet wurde.

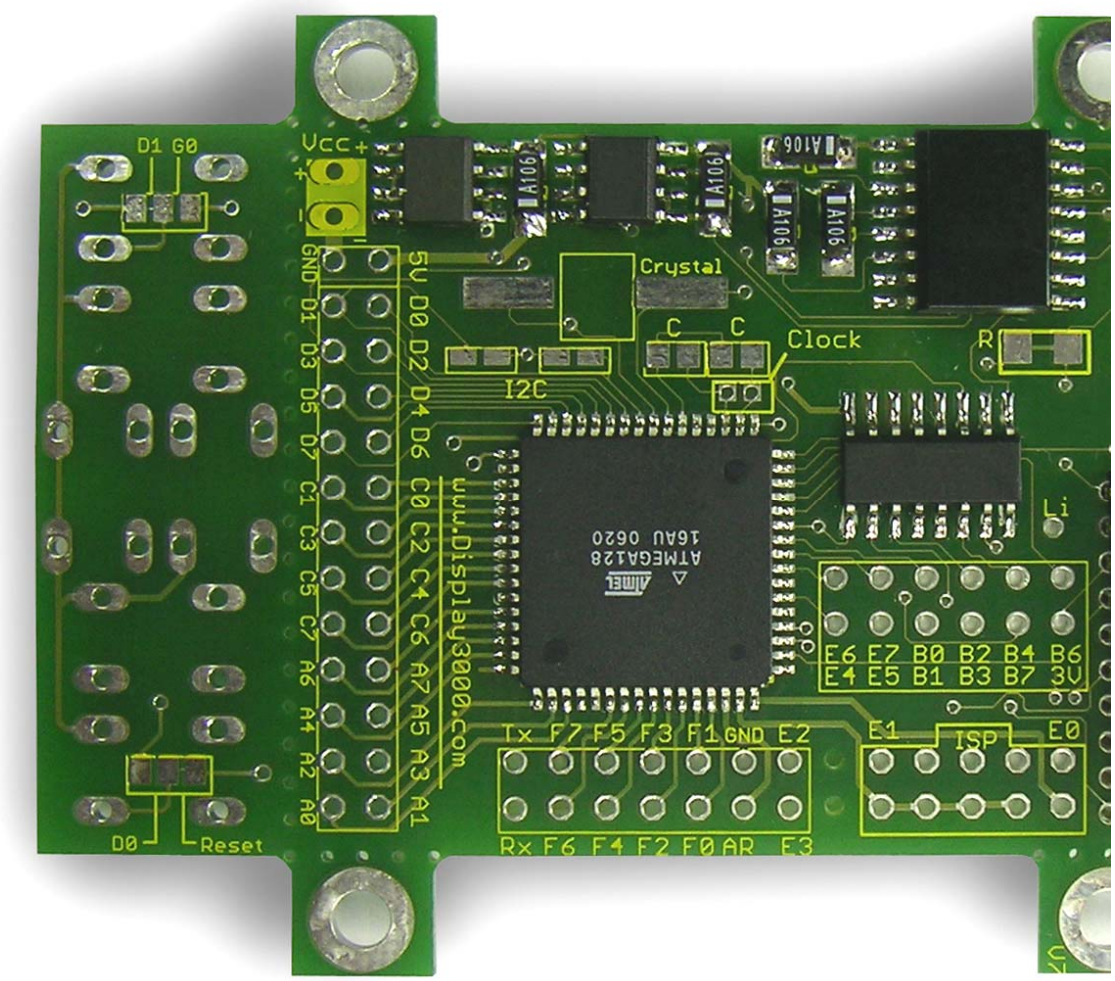
Programmierschnittstelle
Um das Board zu programmieren, brauchen Sie einen sog. ISP-Programmer, der nicht Bestandteil des Lieferumfanges ist, aber preiswert über uns erworben werden kann. Dieser ISP-Programmer verbindet Ihren PC mit unserem D062x Board und erlaubt das Überspielen Ihres am PC erstellten Programms. Mehr zum Anschluss des ISP-Programmieradapters erfahren Sie auf Seite 15.

Optionen

Das Modul bietet einige Sonderoptionen, die in den nachfolgenden Kapiteln ebenfalls beschrieben werden. **Wenn Sie diese Optionen nicht mitbestellt haben, müssen Sie sie erst nachrüsten, bevor diese nutzbar sind.**

1. Geschwindigkeit erhöhen durch Zufügen eines SMD-Quarzes
2. Zufügen eines Uhrenquarzes für den internen RTC (Real Time Counter)
3. Ein-/Ausschalten der Displaybeleuchtung durch einen Schalter
4. Steuerung der Displaybeleuchtung durch den Controller
5. Dimmen der Displaybeleuchtung durch den Controller
6. Nutzung des TWI-Interfaces des Controllers (z.B. durch Anschluss von I²C Geräten)
7. Nutzung des CAN-Bus bei einem AT90CAN128

Die Lötpads des Moduls D062x



Das Modul wird mit Bestückungsdruck ausgeliefert, so dass die Portbelegung dieser entnommen werden kann.

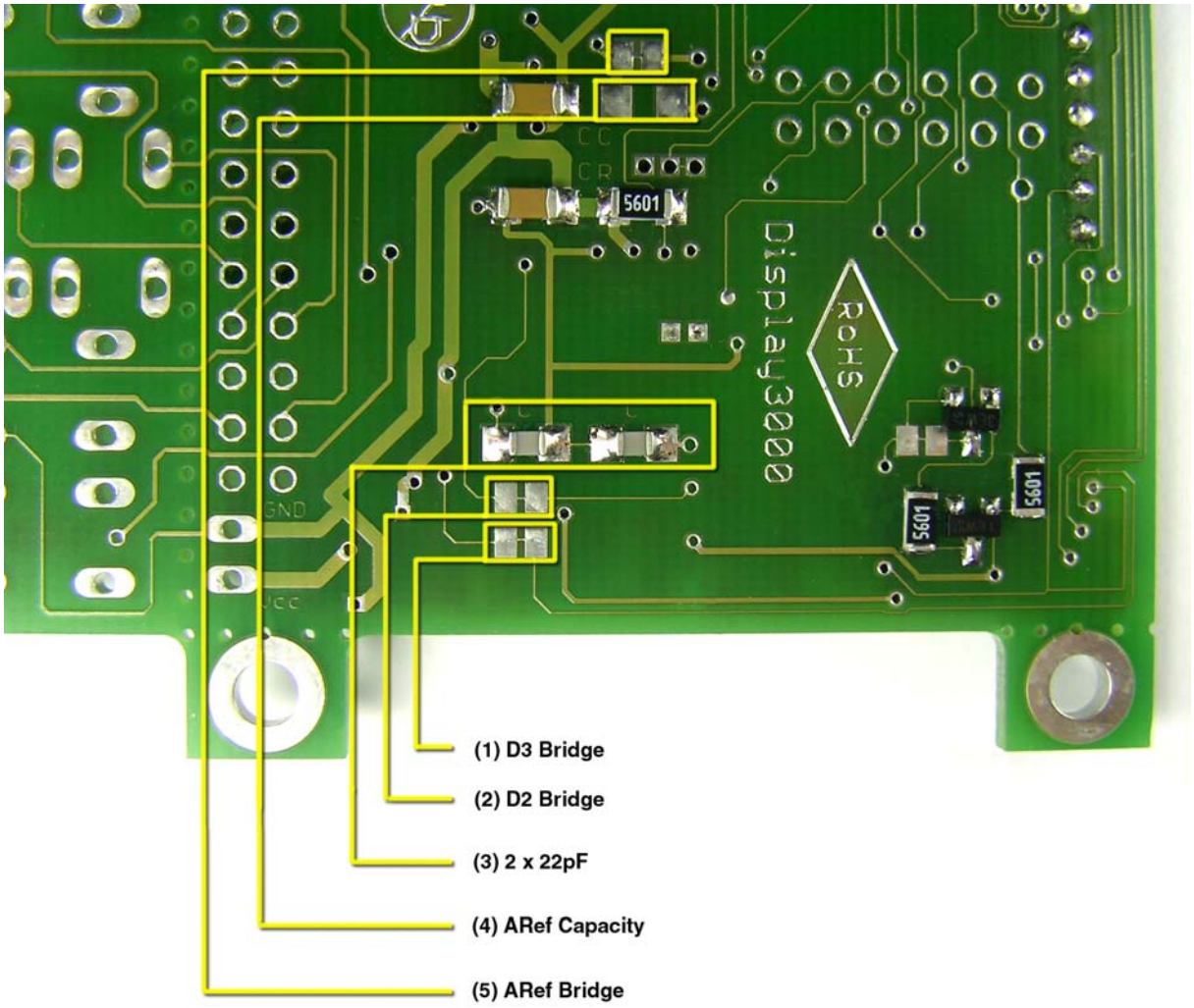
Errata für Platine V7 (siehe Beschriftung auf dem Montage-Pad rechts unten)**:**
Port B6 ist versehentlich falsch beschriftet bzw. belegt worden – hier liegt in Wirklichkeit Port B5 an, der jedoch von Ihnen nicht genutzt werden sollte, da dies die Select-Leitung des Displays ist. In zukünftigen Versionen (V8 erst ab ca. Frühjahr 2008) wird an dieser Stelle wirklich Port B6 liegen.

Das Modul beinhaltet einige Besonderheiten, die nachfolgend beschrieben werden:

Lötbrücken

Es gibt auf der Platinenrückseite vier Lötbrücken – alle sind standardmäßig geschlossen.

Die ersten drei werden hier besprochen, die vierte Lötbrücke im Kapitel Displaybeleuchtung.



Lötbrücke ARef bridge **(5):**

Einige (acht) Eingänge des Controllers können analoge Werte auswerten. Wer sehr exakte Werte braucht, der findet am ATMega128 einen spezielle Port für eine Referenzspannung: ARef. Oft wird eine externer Zuführung von ARef nicht benötigt (dann ist ARef mit der Versorgungsspannung von 5V verbunden), aber wenn doch, haben wir Vorbereitungen getroffen:

Der Eingang ARef ist über die Lötbrücke (5) bereits mit Vcc verbunden. Diese Lötbrücke ist von uns standardmäßig geschlossen. Sollten Sie die ARef für das Anlegen einer eigenen Referenzspannung benötigen, so Öffnen Sie bitte diese Lötbrücke. Das bedeutet: mit einem scharfen Messer trennen Sie vorsichtig die dünne Leiterbahn zwischen den beiden Feldern durch. Solange Sie diese Leitung nicht durchtrennen, steht zudem außen an den Pads des

ARef-Anschluss die normale 5-Volt-Versorgungsspannung für andere Nutzungen zur Verfügung – wenn die Leitung durchtrennt ist, legen Sie an diesem Pad Ihre eigene Refernzspannung an.
Hinweis für Experten: Bitte nutzen Sie nicht die interne Referenzspannung des Controllers, solange Sie ARef mit Vcc verbunden haben.

Mittels eines Lötkolbens können Sie diese Aktion jederzeit wieder rückgängig machen: Verbinden Sie lediglich mittels eines Tropfen Lötzinns die beiden Felder, zwischen denen Sie die Leiterbahn durchtrennt haben.

Das Feld ARef Capacity (4) erlaubt Ihnen zudem noch das Einlöten eines eigenen SMD-Glättungskondensators für die ARef-Leitung. Diese Felder für den Kondensator sind mit ARef und Masse verbunden.

D3. und D.2 Lötbrücken (1) und (2)
Die Ports D.2 und D.3 werden für die RS232-Kommunikation benötigt und sind mit dem RS232-chip auf der Platine verbunden. Wenn Sie also eine „echte" RS232-Verbindung (über die beiden Anschlüsse RX und TX) nutzen möchten, können Sie nicht auch gleichzeitig die Ports D.2 und D.3 nutzen.

Wenn Sie RS232 nicht nutzen möchten und die Ports D.2 und D.3 anderweitig benötigen, so ist es empfehlenswert, diese vom RS232-Chip auf der Platine abzukoppeln.
Dazu durchtrennen Sie die Leiterbahnen mit einem scharfen Messer jeweils zwischen den beiden Feldern.

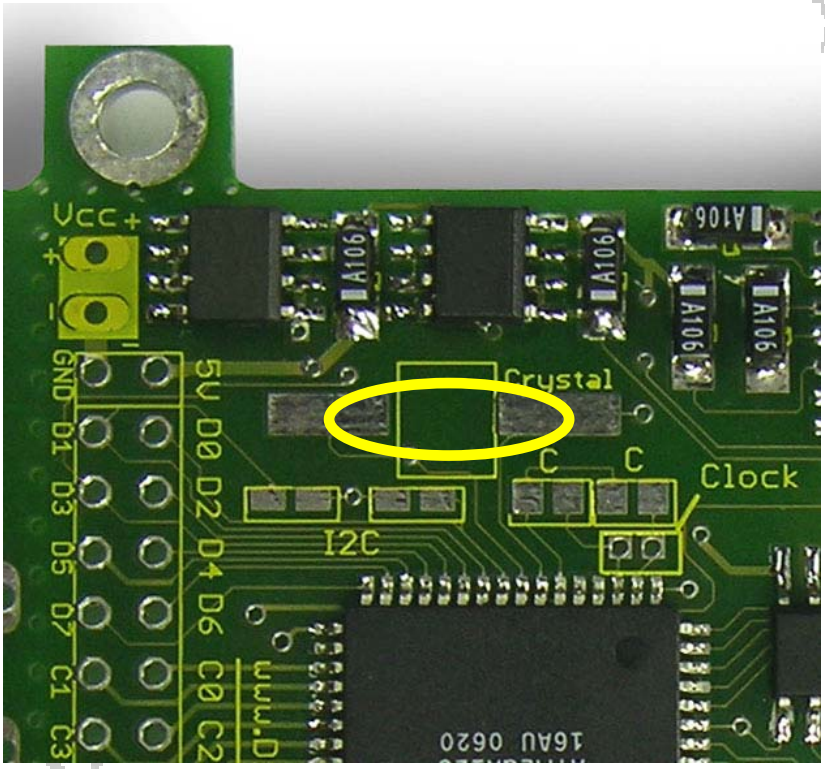
D.2 und D.3 stehen nun immer noch an der Pfostenleiste zur Verfügung, lediglich die Verbindung zum RS232 Chip ist nun unterbrochen. Mittels eines Lötkolbens können Sie diese Aktion jederzeit wieder rückgängig machen: Verbinden Sie lediglich mittels eines Tropfen Lötzinns die beiden Felder, zwischen denen Sie die Leiterbahn durchtrennt haben.

Wenn Sie weder RS232, noch D.2 und D.3 benötigen, lassen Sie einfach alles wie es ist. Sie müssen nur in Erinnerung behalten, dass D.2 und D.3 für Sie nicht als „normale" Ports nutzbar sind.

Hinzufügung eines Quarzes

Wenn Sie die Platine ohne einen zusätzlichen Quarz bestellt haben, so wird diese mit 8 MHz Taktfrequenz (interner Takt) geliefert. Sie können jederzeit die Taktfrequenz ändern, in dem Sie einen Quarz sowie zwei 22pF SMD-Kondensatoren einlöten.

Der Quarz wird auf das vorbereitete Feld (siehe Foto unten) gelötet. Die 22pF Kondensatoren liegen direkt unterhalb des Quarzes **auf der anderen Seite der Platine** (siehe Seite 10) – die beiden Felder für die Kondensatoren **(3)** sind auf der Platine übrigens mit einem „C“ gekennzeichnet). Tipp: stellen Sie bei einem separaten Erwerb sicher, dass Sie einen SMD-Quarz sowie zwei 22pF-SMD-Kondensatoren (Bauform 805 oder 1206) erhalten.



Nachdem Sie einen Quarz hinzugefügt haben, müssen Sie dies dem Mikrocontroller noch mitteilen, ansonsten wird er weiterhin mit internen 8 MHz getaktet. Dazu müssen Sie die Geschwindigkeits-Fuse von 0100 (8 MHz intern) auf 1111 (externer Quarz) umstellen.

Achtung: jede andere Einstellung als 0001, 0100, 0100 und 1111 kann dazu führen, dass Ihr Modul nicht mehr arbeitet und auch nicht mehr umgestellt werden kann! Das gleiche passiert, wenn Sie die Fuse auf 1111 umstellen, ohne einen Quarz eingelötet haben! Also: alle anderen Einstellungen oder ein fehlender Quarz führen u.U. zu einem nicht funktionierenden Board. **Warnung! Spielen Sie nicht mit den Einstellungen der Fuses herum!**

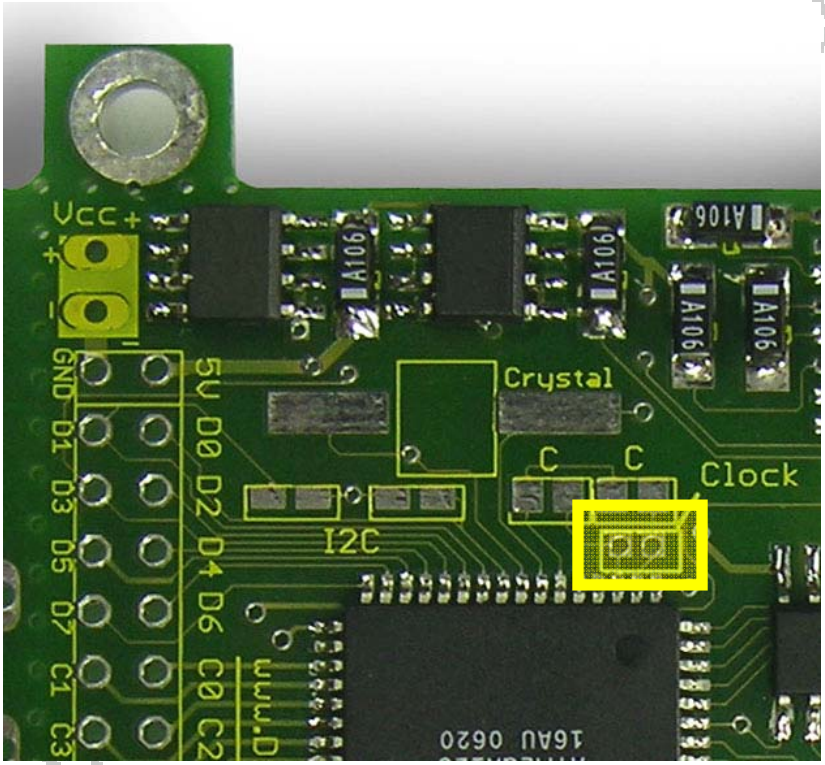
Übertakten
Wenn Sie die Fuse für den Takt auf 1111 gesetzt haben, können Sie jeden beliebigen Quarz einsetzen. Der ATMega 128 ist offiziell auf 16 MHz limitiert, kann aber oft mit bis zu 20 MHz betrieben werden.

Aber Vorsicht: Der empfindlichste Bereich im Controller ist das eingebaute Eeprom. Während 10% Frequenzerhöhung in der Regel unproblematisch ist, können höhere Frequenzen dazu führen, dass das Eeprom nicht korrekt geschrieben oder gelesen wird – während alle anderen Bereiche des Controllers noch problemlos arbeiten. Für Hobby-Zwecke ist das Übertakten OK, aber für kritische Anwendungen sollten Sie die Spezifikationen nicht überschreiten.

Die Auswahl des richtigen Quarzes ist zum einen von der benötigten Geschwindigkeit abhängig (übrigens benötigt der ATMega um so mehr Strom, je schneller er arbeiten muss), zum andern beeinflusst der Quarz auch die Berechnung der Frequenzen, die für eine fehlerfreie RS232-Verbindung benötigt werden. Mehr dazu erfahren Sie auf Seite 17.

Hinzufügung eines Uhrenquarzes für exakte Zeitmessung

Wir haben die Platine für den Einsatz eines üblichen 32.768 KHz Uhrenquarzes vorbereitet. Sie sollten einen solchen Einlöten, wenn Sie einen exakten Timer benötigen, oder den ATMega nur zu gewissen Zeiten „aufwecken“ möchten. Dies ist sehr praktisch, wenn Sie eine Batterie-Anwendung betreiben: Sie wecken ihn 1-2 x pro Sekunden zur Prüfung von Input, Daten etc. und den Rest der Zeit wird der Controller in den Idle-Mode geschickt, d.h. 99% der Zeit „schläft“ er, benötigt kaum Energie und erfüllt trotzdem seine Aufgabe. Das Feld für diesen Uhrenquarz ist auf der Platine beschriftet. Der bedrahtete Quarz wird in die beiden gelb markierten Pads gelötet. Unser Tipp: Klappen Sie den Quarz nach dem Einlöten einfach um 90° um, so dass er auf dem Controller zum Liegen kommt. Dort ist er nicht im Weg.



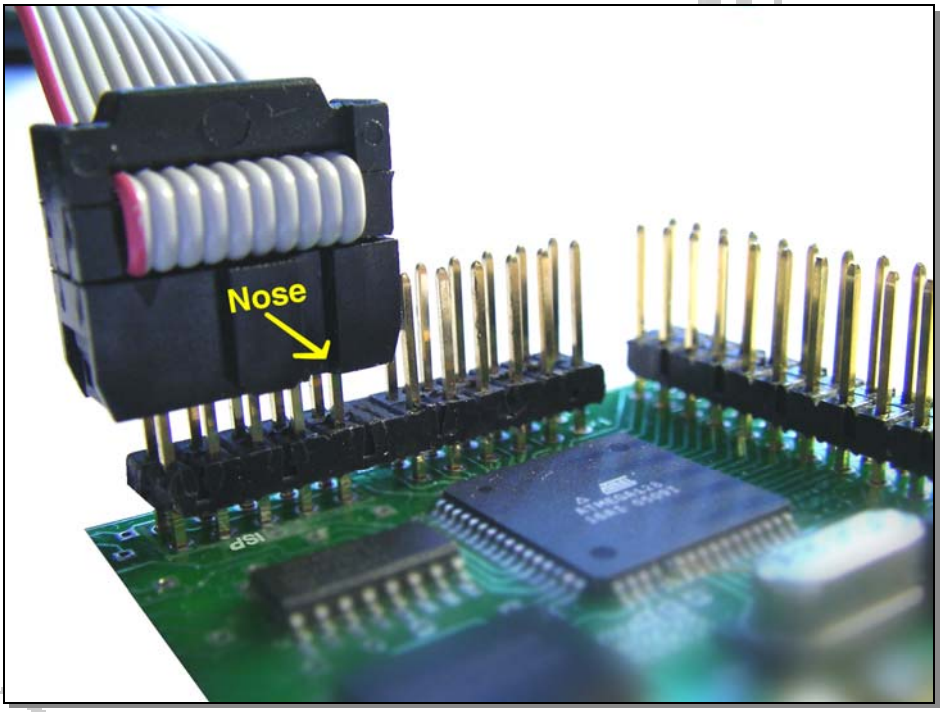
Besonderheit ATMega2561 / AT90CAN128 und Uhrenquarz:
Der ATMega2561 und der AT90CAN128 benötigt laut Datenblatt (anders als die anderen ATMegas) für den den Betrieb des Uhrenquarz noch zwei Kondensatoren von 22pF. Daher haben wir hier zwei Felder vorbereitet: Direkt oberhalb der beiden Lötpads für den Uhrenquarz erkennen Sie 4 mit „C“ gekennzeichnete Lötfelder. Diese sind für 2 Stück SMD Kondensatoren (22pF, BF 805) vorbereitet. Sie dürfen nur bei Verwendung eines ATMega2561 oder AT90CAN128 bestückt werden.

Der ISP-Stecker

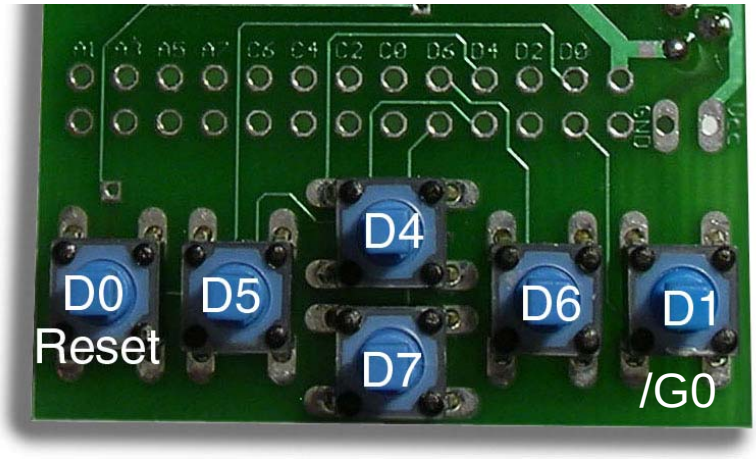
Wir haben versucht, das Modul so klein wie möglich zu gestalten und haben auf den üblichen ISP-Wannenstecker verzichtet – er hätte zu viel Platz benötigt. Stattdessen stehen die gleichen Pins in gleicher Anordnung zur Verfügung – lediglich die Kunststoffwanne des Steckers fehlt.

Bitte stecken Sie ihren ISP-Programmer (bei uns im Shop unter www.shop.display3000.com in drei Anschlussvarianten erhältlich: Parallel, Seriell, USB) so wie im folgenden Bild gezeigt auf. Der Stecker mit seinen 10 Kontakten wird auf die Leiste gesteckt, die ebenfalls 2 Reihen mit je 5 Steckern zeigt – woanders würde der ISP-Programmer auch nicht passen.
Die „Nase“ des Steckers muss in Richtung der Chips zeigen.

Um Ihnen das Erinnern leichter zu machen, haben wir zudem auf die Platine „ISP“ gedruckt sowie die Nase des Steckers stilisiert. Die Nase soll immer in Richtung der Beschriftung ISP zeigen.



Die Tasten des D062x

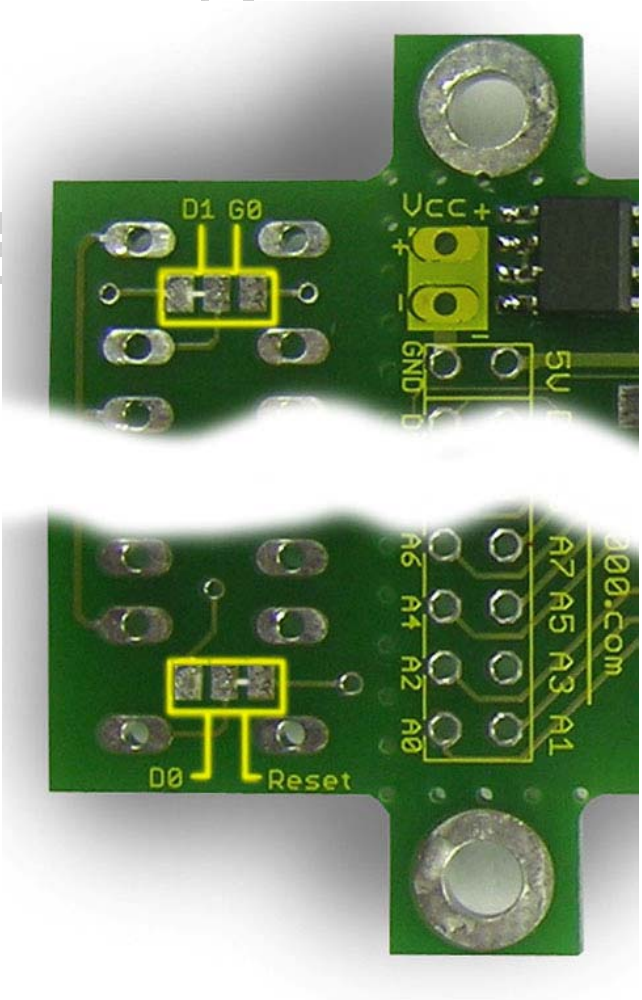


Die Taster vom Modul D062x sind bereits mit Port D verbunden (Belegung siehe Foto links). Die Belegung der linken und der rechten Taste kann durch eine Lötbrücke geändert werden.
Die linke Taste ist standardmäßig mit Reset verbunden. Braucht man Reset nicht oder benötigt eine weitere Eingabetaste, so kann diese Taste alternativ mit Port D.0 verbunden werden.

Die rechte Taste ist mit Port D.1 verbunden. Beim ATMega erlauben die meisten Pins eine Mehrfachnutzung – so auch der Port D.1, der z.B. alternativ als Datenleitung bei einer Nutzung eines I²C-Bussystems fungiert. Ist diese Nutzung geplant, so kann die Belegung dieser Taste von D.1 auf G.0 geändert werden.

Zur Änderung einer Taste trennen Sie bitte auf der Bauteilseite der Platine die dort bestehende Leiterbahnbrücke auf (gekennzeichnet mit *Reset* bzw. *D.1*) und schließen dann die danebenliegenden beiden Lötfelder mit einem Tropfen Lötzinn.

Beispiel: Die linke Taste soll mit D.0 statt mit Reset verbunden werden. Dann trennen Sie (im rechten Foto unten) die bestehende Brücke zwischen dem mittleren und dem rechten Lötfeld mit einem scharfen Messer auf und verbinden dann mittels etwas Lötzinns das linke mit dem mittleren Lötfeld. Der Aufdruck auf der Leiterbahn kennzeichnet die jeweilige Verbindung.



Schalten der Display-Beleuchtung

Gerade bei batteriebetriebenen Geräten macht sich der Stromverbrauch der Hintergrundbeleuchtung des Displays unangenehm bemerkbar. Hier wäre die Möglichkeit der Abschaltung sehr praktisch. Da die LEDs im Display eine begrenzte Lebensdauer von 20.000 bis 50.000 Stunden haben (also bei Dauerbetrieb ca. 2-5 Jahre), macht dies u.U. ebenfalls Sinn, für Projekte, die lange Zeit ohne weitere Beachtung vor sich hin arbeiten sollen.

Aufgrund der für die Displaybeleuchtung notwendigen hohen Spannung (mindestens 6,5 Volt, auf unserem Board sogar ca. 16 Volt) ist hierfür ein wenig Aufwand notwendig: entweder muss ein Schalter, oder zwei SMD Transistoren und zwei Widerstände eingelötet werden. Standardmäßig liefern wir unser Board mit Dauerbeleuchtung aus. Wenn Sie das Board nicht direkt mit Schaltoption bestellt haben, so müssen Sie diese Option selber nachrüsten.

Im Lieferzustand besteht eine direkte Leitung zwischen dem PowerBooster und dem Display mit der integrierten Beleuchtung. Die Displaybeleuchtung ist von der übrigen Elektronik des Displays separiert und kann daher getrennt geschaltet werden.

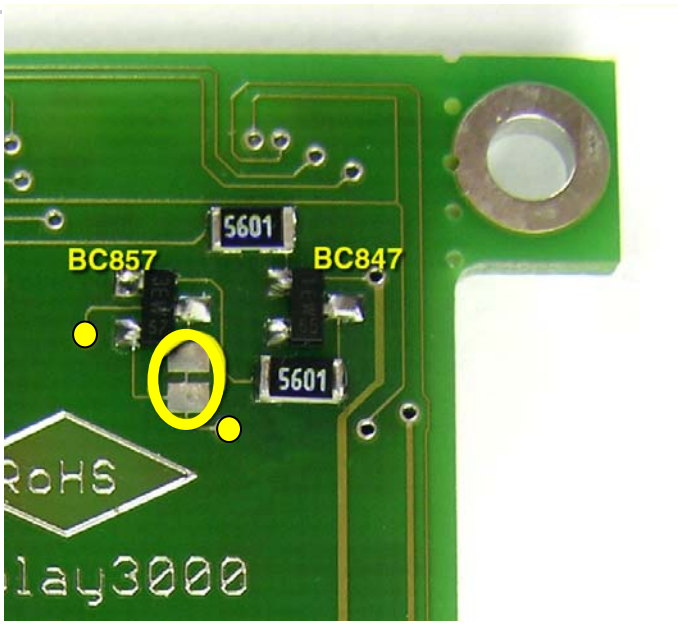
Da standardmäßig eine direkte Leitung besteht, muss diese zuerst aufgetrennt werden.

Dazu trennen Sie die im folgenden Foto im Kreis sichtbare Brücke mit einem scharfen Messer auf. Hinweis: Wenn Sie das Modul ohne die Option zum Schalten der Displaybeleuchtung bestellt haben, so fehlen die hier im Foto gezeigten Bauteile – das manuelle Schalten per Schalter ist aber natürlich trotzdem möglich.

Schalten der Beleuchtung mit einem Schalter:

Löten Sie an die beiden Lötpads rechts und links der durchtrennten Leiterbahn (im gelben Kreis) ein Kabel und an dessen Ende einen Schalter und die Beleuchtung kann von Ihnen mittels dieses Schalter manuell ein- oder ausgeschaltet werden.

Sie können dieses Kabel alternativ auch an den beiden gezeigten Durchkontaktierungen einlöten (von der Bauteilseite durchstecken und von der Rückseite her festlöten). Die beiden Durchkontaktierungen sind als gelbe Punkte gekennzeichnet.



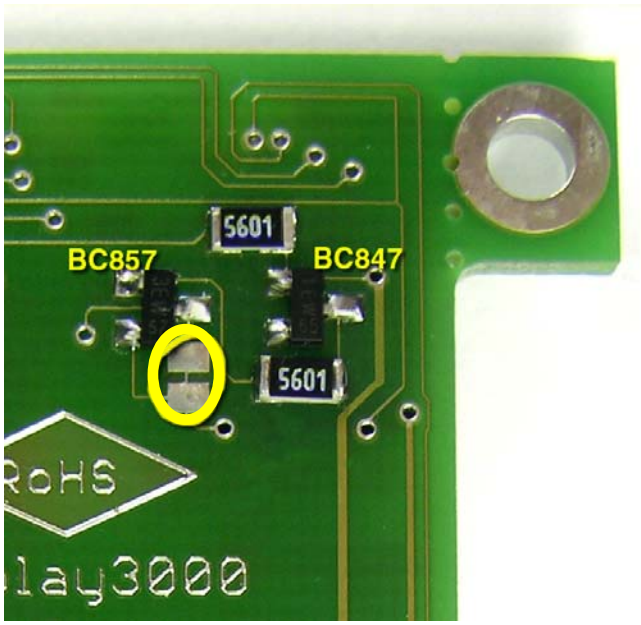
Schalten der Beleuchtung durch den Mikrocontroller

Statt eines manuellen Schalters können Sie die Beleuchtung auch automatisieren, so dass der Mikrocontroller die Beleuchtung steuert (z.B. die Beleuchtung abschaltet, wenn über 10 Minuten keine Bedienung mehr stattgefunden hat – sobald eine Taste betätigt wird, schaltet die Beleuchtung sich wieder ein).

Da die Beleuchtung eine höhere Spannung benötigt, als der Controller sie liefern kann, können wir diese nicht direkt anschließen – wir brauchen dazwischen noch Transistoren als elektronische Schalter. Genauer gesagt, brauchen Sie:
1 x SMD-PNP-Transistor BC857A (Bauform SOT 23)
1 x SMD-NPN-Transistor BC847A (Bauform SOT 23)
2 x SMD-Widerstände 5,6 KOhm (Bauform 805)

Wir haben das Board so vorbereitet, dass Sie die Beleuchtung mittels **Port B.7** steuern können. **Zuerst trennen Sie die Brücke so auf**, wie auf der vorherigen Seite beschrieben.

Dann löten Sie auf die Rückseite die **beiden Transistoren** wie auf dem Foto gezeigt auf. Wichtig: **rechts befindet sich der NPN-Transistor (BC847)** und **links der PNP-Transistor (BC857).** Als nächstes werden die beiden 5,6 KOhm Widerstände eingelötet. Das war es schon.



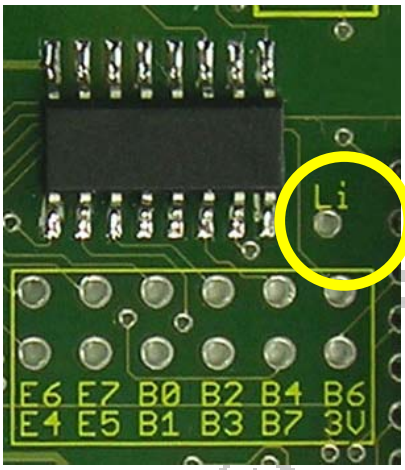
Beim Anschluss an die Stromversorgung sollte Ihr Display nun solange dunkel bleiben, bis Sie in Ihrer Software Port B.7 als Ausgang definieren und auf High schalten. Der Port B.7 ist nun natürlich nicht mehr anderweitig von Ihnen zu nutzen (siehe hierzu auch die nächste Seite).

Beachten Sie: in unseren Beispielprogrammen wird Port B auch für die Displayansteuerung genutzt. Port B.7 war bislang ungenutzt und muss, damit es nun als Ausgang genutzt werden kann, auch entsprechend definiert werden.

Ändern Sie daher in den Bascom-Beispielen die Zeile **Ddrb = &B01100110** ab in **Ddrb = &B11100110**. Damit wird Port B.7 als Ausgang bereitgestellt. Mit dem Befehl **Portb.7 = 1** oder **Portb.7=0** schalten Sie nun die Beleuchtung ein oder aus.

Änderung des Ports für die Steuerung der Displaybeleuchtung:
Wenn Sie die Beleuchtung über einen anderen Port als Port B.7 steuern möchten (z.B. weil Sie diesen PWM Kanal für eine andere Aufgabe benötigen), so gibt es hierfür eine Vorbereitung auf der Platine:

In der Nähe des oberen Rands der Platine befindet sich ein Lötpad, welches mit „Li“ beschriftet ist (siehe Foto: gelber Kreis). Wenn Sie die Verbindung von Pad B7 zum Pad „Li“ mit einem Messer kappen (andere Seite der Platine) und dann mittels eines kurzen Drahtstücks eine Verbindung vom Pad „Li“ zu einem anderen beliebigen Port herstellen, so können Sie die Displaybeleuchtung dann über diesen geänderten Port steuern.

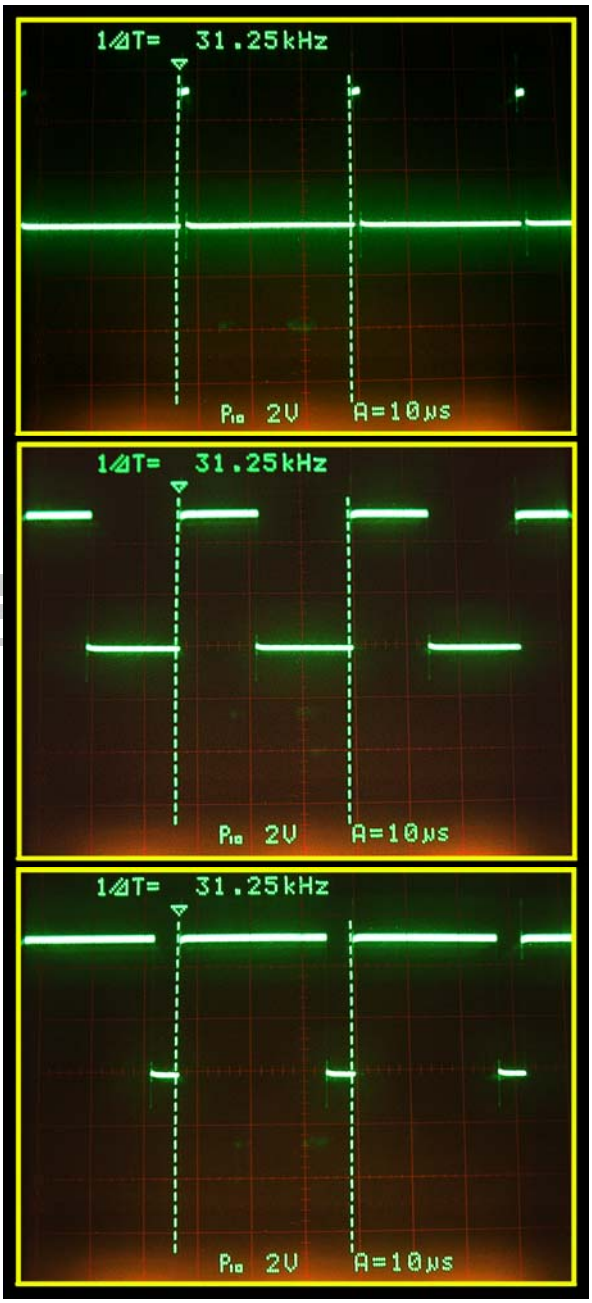


Eine weitere interessante Alternative zur Steuerung der Displaybeleuchtung ist die Ansteuerung mittels gepulster Signale. Damit lässt sich die Beleuchtung auch dimmen, d.h. in der Helligkeit anpassen. Mehr dazu im nächsten Kapitel.

Steuerung (Dimmen) der Displaybeleuchtung mittels PWM (Pulsweitenmodulation)

Die Pulsweitenmodulation wird zur Informationsübertragung und zusätzlich häufig zur Steuerung der Energieumwandlung in einem technischen System eingesetzt.

Zuerst etwas (vereinfachte) Theorie:
Wenn Sie z.B. eine Leuchtdiode binnen 1 Sekunde 5 Mal für 0,1 Sekunden ausschalten und dann wieder für 0,1 Sekunden einschalten, sehen Sie zuerst einmal ein nerviges Flackern. Zudem war aber die LED die Hälfte der Zeit abgeschaltet und hat daher in der Gesamtzeit auch nur die Hälfte an Lichtenergie abgegeben.



Angenommen, Sie beschleunigen diese Rate auf eine Ein- und Ausschaltzeit von 0,001 Sekunden, dann würden das Auge sicher kein Flackern oder Flimmern mehr registrieren – es würde jedoch eine Leuchtdiode sehen, die scheinbar nur mit halber Kraft leuchtet – kein Wunder, sie ist ja auch die Hälfte der Zeit (nämlich pro Sekunde 500 x für je eine tausendstel Sekunde abgeschaltet – und die gleiche Zeitspanne eingeschaltet).

Wenn Sie nun das Verhältnis von 1:1 im obigen Bespiel ändern auf z.B. 1:3 (also die LED ist binnen einer Sekunde 500 Mal für 0,0005 Sek. eingeschaltet und 500 Mal für 0,0015 Sek. abgeschaltet), dann würde die Helligkeit noch weiter abnehmen – im umgekehrten Fall, also wenn die LED länger ein als ausgeschaltet wäre, würde die Helligkeit zunehmen.

Dies nennt man Pulsweitenmodulation: das Tastverhältnis variiert, die Frequenz bleibt die gleiche.

Die nebenstehende Darstellung im Oszilloskop verdeutlicht dies. In der Mitte das 1:1 Verhältnis (d.h. die LED ist in ca. 50% der Zeit abgeschaltet und würde mittelmäßig hell leuchten); oben 1:15 (Die LED wäre meist abgeschaltet, also sehr dunkel); unten ca. 7:1 (LED meist eingeschaltet, also recht hell).

Durch das Verhältnis der Einschaltdauer zur Ausschaltdauer in einer definierten Zeit kann die einem Verbraucher zugeführte Leistung gesteuert werden. In unserem Fall könnte also der Mikrocontroller mittels PWM, also der schnellen Steuerung der Ein- und Ausschaltzeiten, die Helligkeit der Displaybeleuchtung variieren.

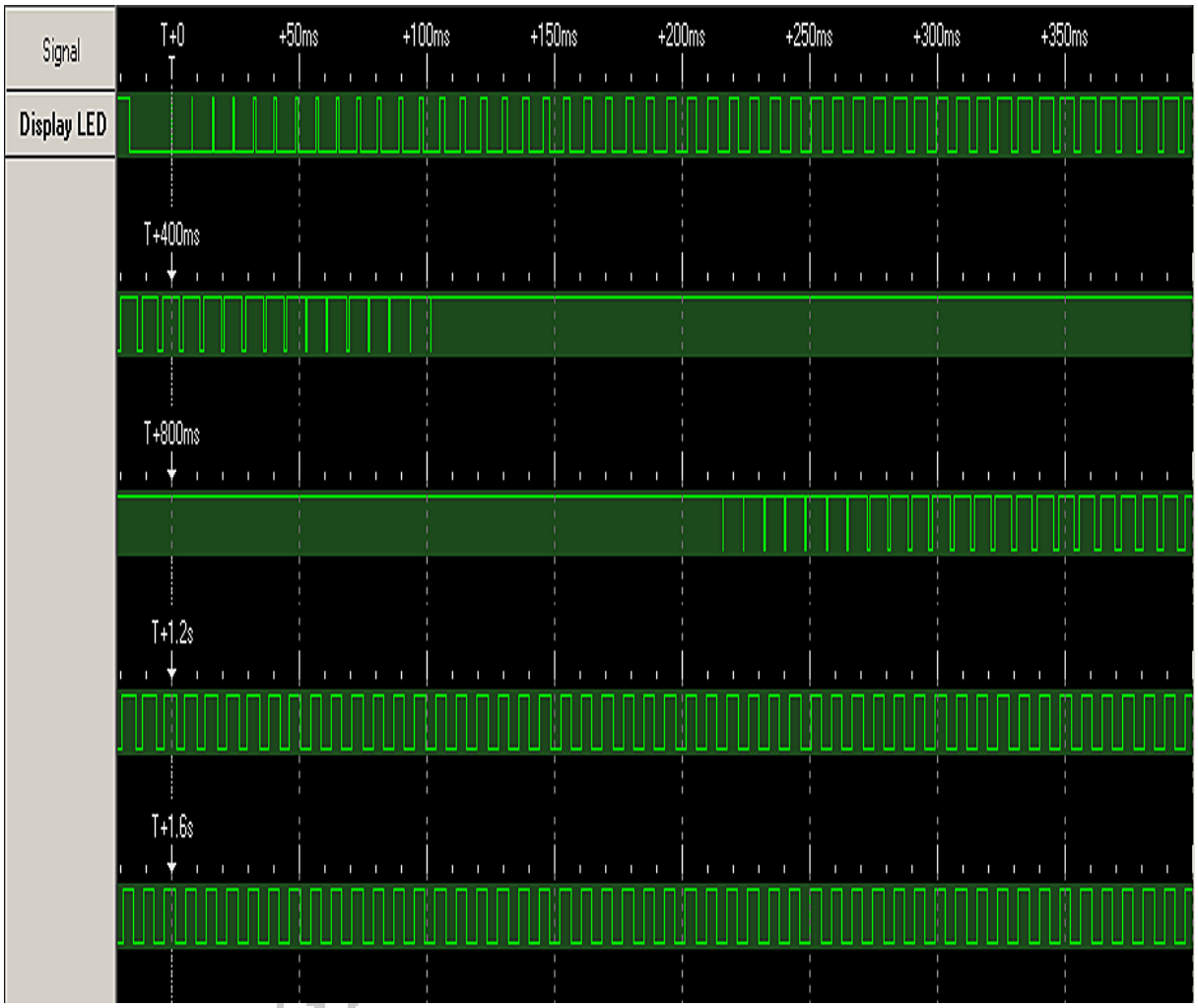
Starten Sie nun noch nicht gleich mit der Erstellung eines Programms, welches diese gepulste Ausgabe realisiert! Der auf unserem Modul befindliche Atmel-Mikrocontroller kann diese PWM-Steuerung hardwareseitig quasi nebenbei erledigen – für PWM ist extra entsprechende Hardware inkludiert. Um diese zu nutzen, ist nicht viel Programmieraufwand notwendig, lediglich durch das Setzen einiger Parameter wird der Pulsweitenmodulator gestartet; und durch Variation des genutzten Timers wird das Tastverhältnis von Eingeschaltet zu Ausgeschaltet verändert. Den Rest erledigt die Hardware quasi nebenbei – sie brauchen also in Ihrer Software keinerlei Resourcen hierfür zur Verfügung stellen.

Beispiel in Bascom zur Veranschaulichung:
Die folgenden Zeilen starten PWM auf Port B.7 und dimmen das Display langsam von 0% auf 100% hoch um dann, nach 5 Sekunden Wartezeit, die Beleuchtung auf 50% zu reduzieren. Dann endet das Programm. Preisfrage: Was passiert dann hier mit der Displaybeleuchtung? Wird sie abgeschaltet, bleibt sie bei 50% oder wieder bei 100%? Antwort auf der nächsten Seite.

```
Gosub Lcd_cls
Call Lcd_print( "Display3000" , 1 , 1 , 2 , Weiss , Dunkelrot)

Config Timer1 = Pwm , Pwm = 8 , Compare C Pwm = Clear Up , Compare C Pwm =
       Clear Down , Prescale = 256

For I = 0 To 255 Step 5
   Pwm1c = I
   Waitms 10
Next I

Waitms 500

For I = 255 To 130 Step -5
   Pwm1c = I
   Waitms 10
Next I

End
```

Zu kurzen Erläuterung: PortB.7 hängt an der Hardware für den PWM Kanal C. Mit *Pwm1c* wird das entsprechende Register mit dem gewünschten Wert beschrieben.

Zur Verdeutlichung haben wir zeigen wir das Ausgangssignal des Mikrocontrollers noch mit einem Logikananalyzer aufgezeichnet. Die gesamte Aufzeichnung ist 2 Sekunden lang, in jeder Zeile sind 400ms abgebildet. Sehr schön ist zu erkennen, wie das Tastverhältnis von Hell zu Dunkel sich mit fortlaufender Zeit verändert.



Die ersten 510 ms werden benötigt, um die Helligkeit in 51 Schritten von 0 auf 100% zu steigern – nach jedem Schritt folgt eine Wartezeit von 10 ms. Dann folgt eine Pause von 500ms (Waitms 500), in der das Display mit voller Stärke leuchtet. Dann wird das Display in 25 Schritten auf eine Leuchtstärke von ca. 50% gefahren (dies wird bei 1.25 Sekunden erreicht) und das Programm dann beendet. Nun die Antwort auf unsere Frage von der vorhergehenden Seite: Sie erkennen, obwohl das Programm beendet wurde (ca. ab Position 1,25 Sek.), arbeitet der ATMega seinen PWM-Befehl weiterhin ab (=das Display wird weiter in einem Tastverhältnis von 1:1 gepulst) und verbraucht dabei keinerlei Resourcen.

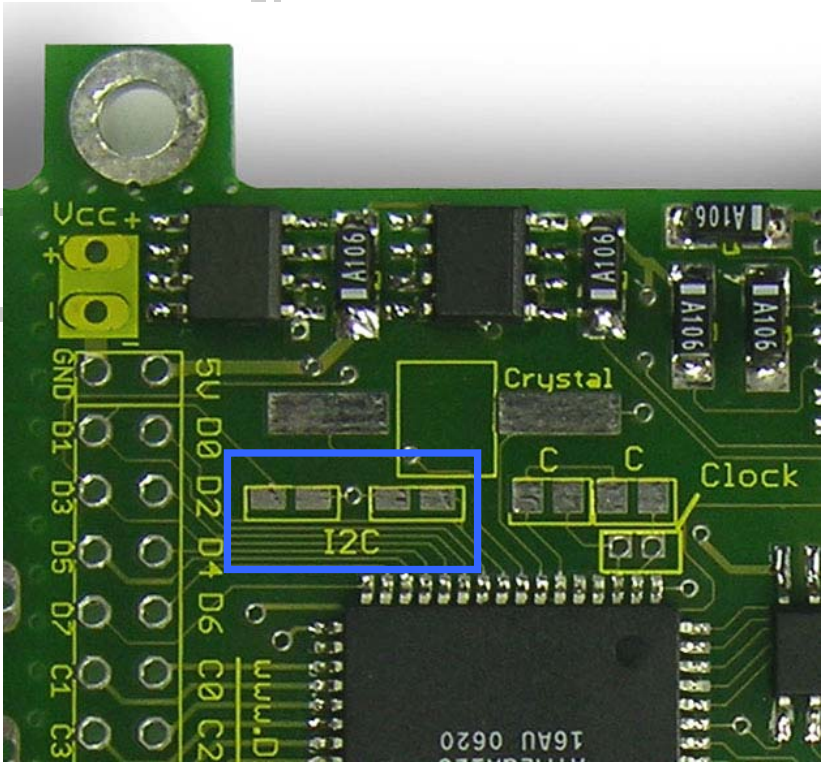
I²C / TWI – Zweidraht-Interface

Der ATMega bietet u.a. auch sein TWI, was nichts anderes bedeutet als Two-Wire-Interface. Ein bekannten Vertreter eines TWI ist z.B. I²C (ausgesprochen: I square C).

Dieses Interface wird manchmal auch 2-Draht-Bus genannt, da der Bus tatsächlich nur mit 2 bidirektionalen Leitungen auskommt (Masse und Versorgungsspannung nicht mitgerechnet). Es ist ein ist ein serieller synchroner Zweidraht-Bus, eine Leitung enthält das Clock-Signal, die andere Leitung das Datensignal.

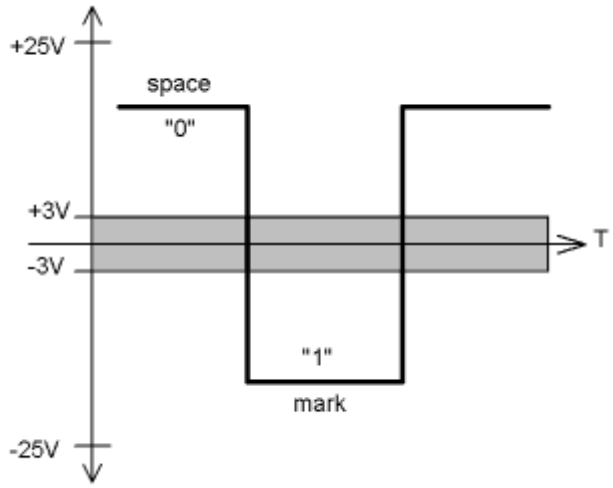
Wofür braucht man dieses? In vielen modernen elektronischen Systemen wird häufig eine Kommunikation der einzelnen Bausteine untereinander benötigt. Man will aber auch nicht viele Meter Leitungen ein System legen – ein Bus erlaubt es, dass alle Bausteine am gleichen Kabelstrang hängen und entweder miteinander oder mit einer Master-Einheit kommunizieren. Ein großer Vorteil des I2C-Bus ist auch die einfache Ansteuerung. Da keine festen Taktzeiten eingehalten werden müssen, können sowohl langsame als auch sehr schnelle Busteilnehmer, Chips und Programmiersprachen eingesetzt werden. Soviel zur Einführung – weitere Informationen hält das Internet in Hülle und Fülle bereit.

Unser Board D062 bietet an den Ports D0 und D1 den direkten Anschluss an einen I²C-Bus. Ein I²C Bus muss immer mittels Pullup-Widerständen auf ein definiertes Spannungsniveau angehoben werden. Wenn das Board D062 als Masterboard fungieren soll und es im System noch keine weiteren I²C Bus-Teilnehmer mit Pullup-Widerständen gibt, so können diese direkt im D062-Board eingelötet werden. Das folgende Foto zeigt die beiden bereits beschrifteten Positionen (blaues Rechteck), an denen jeweils ein 4,7 KOhm bis 10 KOhm-Widerstand (SMD Bauform 805) eingelötet werden kann. Die Lötpads stehen dort zur Verfügung. Mehr ist (außer Software) für I²C nicht notwendig.



Die RS232-Schnittstelle

RS-232 ist eine Schnittstelle, welche 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)



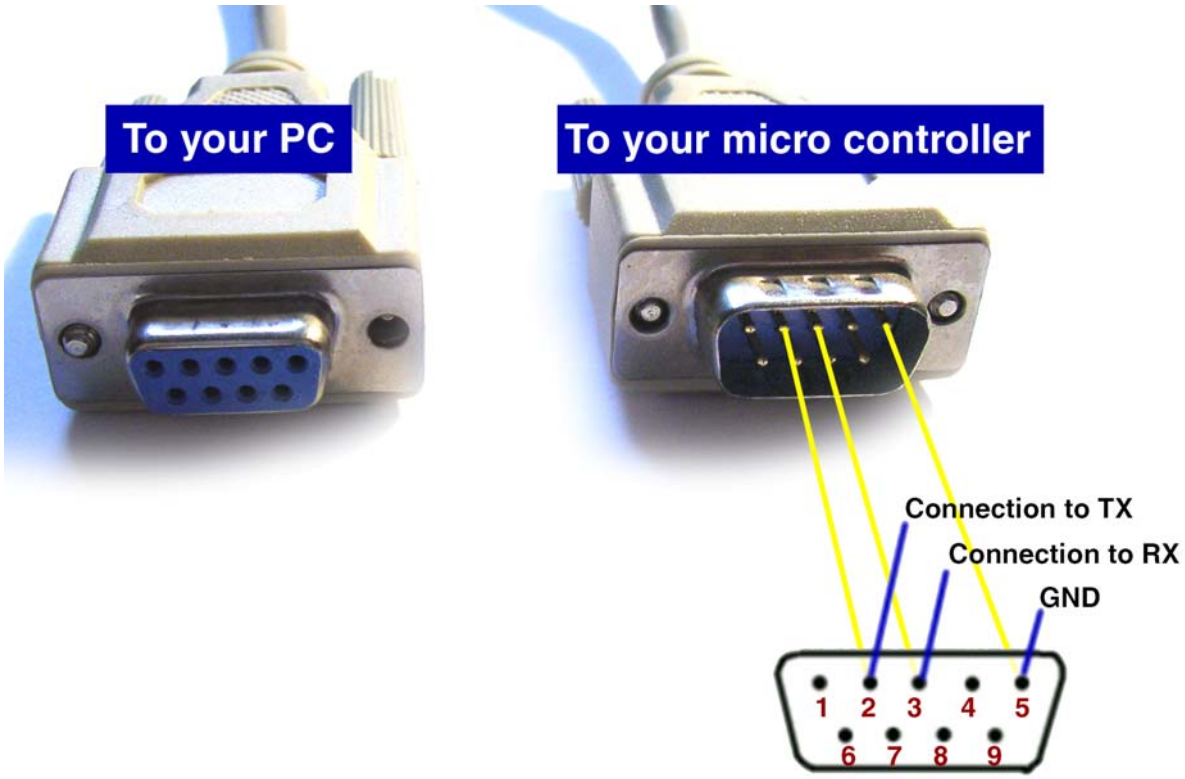
Wie das obige Bild zeigt, ist der Spannungsbereich von -3 bis +3 Volt undefiniert. In der Praxis ist dies jedoch nicht so. Meistens werden alle Spannungen oberhalb von +2,5 Volt als logische Eins angesehen und alle Spannungen darunter als logisch Null.

Die elektronische Spezifikation der RS-232 Verbindung ist robust – alle 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 daher die genaue Geschwindigkeit kennen (als Baud-Rate bezeichnet).

Wir nutzen hier die drei wichtigsten Signale des RS-232-Systems:
- RxD : receive data (Empfang), Pin 2 am DB9 Stecker
- TxD : transmit data (Senden), 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 RX- und TX-Leitungen 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 ATMega128 bietet zwei unabhängige RS-232 Interfaces, wir nutzen das Interface Nr. 1 (das andere ist das Interface 0 – wir nutzen es hier nicht, da die Ports dieser Schnittstelle gleichzeitig auch für die ISP-Programmierung genutzt werden – mehr dazu im ATMega128-Datenblatt).
Das Interface 1 steht an den Ports D.2 und D.3 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 und wie Sie oben lesen konnten, sind für RS-232 Spannungen notwendig, die nicht Mikrocontroller-tauglich sind. 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 der Platine nutzen möchten, hilft Ihnen evtl. das nachfolgende Beispiel. **Tipp:** Diese 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 enhält.
Die Ausgabe des Boards über die RS-232 Schnittstelle lassen Sie sich dann mittels eines Terminal-Programms anzeigen.
Unter MS-Windows® 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 = "m128def.dat"
$crystal = 8000000
$baud1 = 9600

Open "COM2:" For Binary As #1
Do
  Print #1 , "Hello world"
  Wait 1
Loop
Close #1
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, und Sie können das Board direkt mit einen Quarz und der korrekten Einstellung bestellen.

b) Der übliche externe 16 MHz-Quarz ist nicht optimal, denn durch die Teilung wird keine 100% korrekte für eine RS232-Baudrate notwendige 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. Es kann (muss nicht) bei 18 Mhz bereits zu Problemen führen. Übrigens treten evtl. Programmfehler als erstes beim Schreiben und Lesen des internen Eeproms auf. Der eigentliche Controller-Kern läuft oft auch mit 18 MHz fehlerfrei – dann aber ist ein Zugriff auf das Eeprom nicht mehr zu empfehlen.

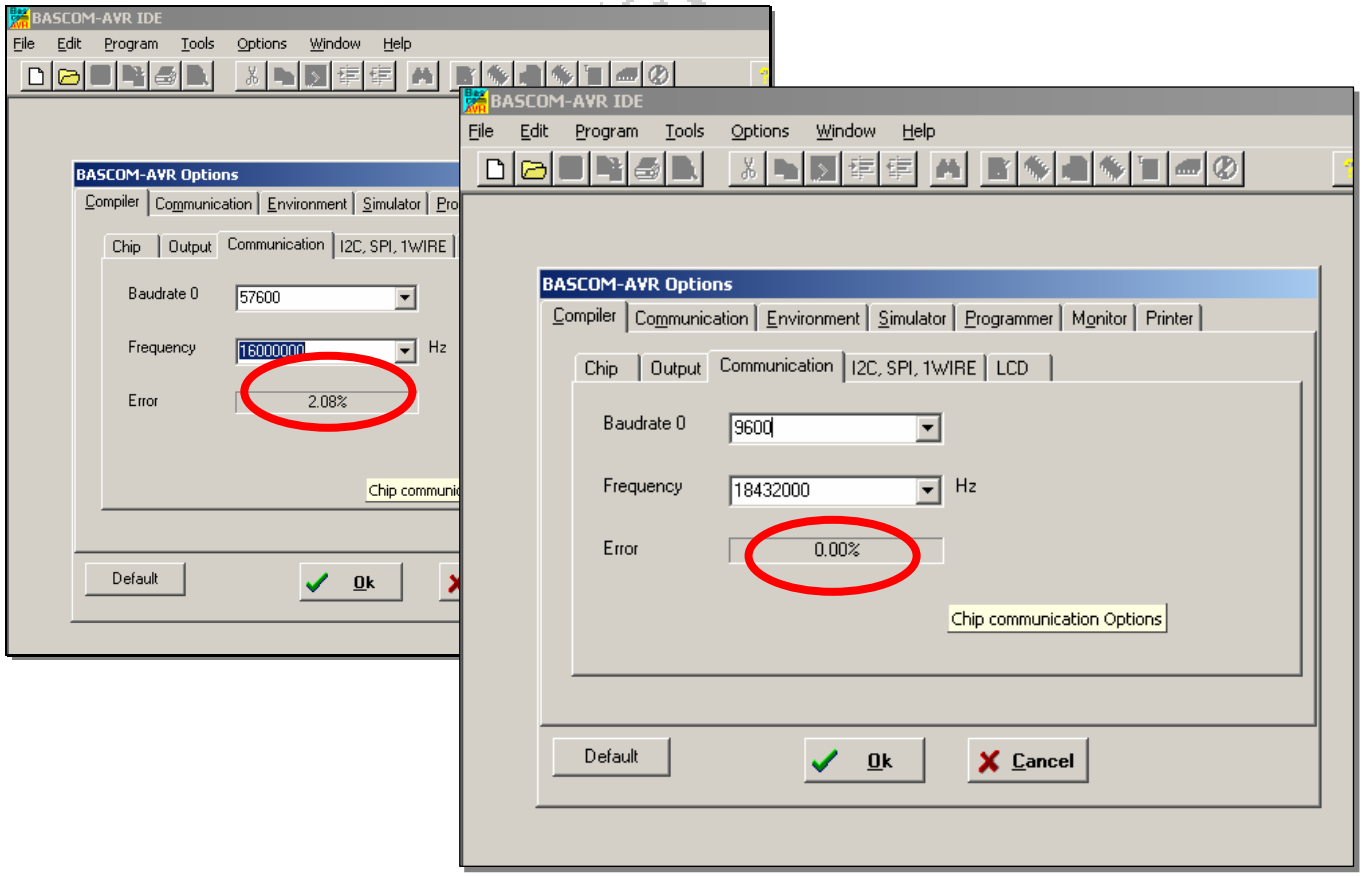
c) Sie müssen dem Compiler mitteilen, welche Taktfrequenz am Controller anliegt, sonst wird die Ermittlung des notwendigen Teilers für die Berechnung der Baudrate nicht korrekt durchgeführt. In Bascom führen Sie dies mit dem Befehl **`$crystal = 8000000`** am Anfang des Programms durch at the beginning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

Hier muss die exakte Taktfrequenz des Quarzes eingegeben werden, ansonsten ist eine RS-232-Verbindung aufgrund der falschen Baudrate nicht möglich.

Die folgende Tabelle zeigt die Fehlerquote (gerundet) aufgeschlüsselt nach gewünschter Baudrate und Taktrate des Controllers. Eine schwarze Zahl ist OK, rote Zahlen können zu einer gestörten Verbindung führen.

	Taktfrequenz des Controllers in MHz										
Baud	**1,00**	**2,00**	**4,00**	**7,373**	**8,00**	**11,059**	**14,318**	**14,746**	**16,00**	**18,432**	**20,00**
2400	0,2%	0,2%	0,2%	0,0%	0,2%	0,0%	0,0%	0,0%	-0,1%	0,0%	0,0%
4800	0,2%	0,2%	0,2%	0,0%	0,2%	0,0%	0,2%	0,0%	0,2%	0,0%	0,2%
9600	-7,0%	0,2%	0,2%	0,0%	0,2%	0,0%	0,2%	0,0%	0,2%	0,0%	0,2%
14400	8,5%	-3,5%	2,1%	0,0%	-0,8%	0,0%	0,2%	0,0%	0,6%	0,0%	-0,2%
19200	8,5%	-7,0%	0,2%	0,0%	0,2%	0,0%	-0,8%	0,0%	0,2%	0,0%	0,2%
28800	8,5%	8,5%	-3,5%	0,0%	2,1%	0,0%	0,2%	0,0%	-0,8%	0,0%	0,9%
38400	-18,6%	8,5%	-7,0%	0,0%	0,2%	0,0%	1,3%	0,0%	0,2%	0,0%	-1,4%
57600	8,5%	8,5%	8,5%	0,0%	-3,5%	0,0%	-2,9%	0,0%	2,1%	0,0%	-1,4%
76800	-18,6%	-18,6%	8,5%	0,0%	-7,0%	0,0%	-2,9%	0,0%	0,2%	0,0%	1,7%
115200	-45,7%	8,5%	8,5%	0,0%	8,5%	0,0%	-2,9%	0,0%	-3,5%	0,0%	-1,4%
230400	-72,9%	-45,7%	8,5%	0,0%	8,5%	0,0%	-2,9%	0,0%	8,5%	0,0%	8,5%
250000	-75,0%	-50,0%	0,0%	-7,8%	0,0%	-7,8%	-10,5%	-7,8%	0,0%	-7,8%	0,0%

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**.



Tipps zur Auswahl eines Quarzes:

Da der interne 8 MHz Taktgeber des Mikrocontrollers nicht sehr akkurat ist (lediglich eine RC-Kombination), sollten Sie **immer** einen externen Quarz nutzen, wenn Sie RS-232 einsetzen möchten oder wenn Sie z.B. eine Uhr mitlaufen lassen möchten.

Ansonsten ist die Wahl des Quarzes lediglich davon abhängig, welche Geschwindigkeit Sie benötigen und ob bei gewünschter RS232-Anbindung die Abweichung vom Ideal noch akzeptabel ist.

RS232 / RS485 / CAN-Bus:
14.7456 MHz oder 18.432 MHz führen zu 0,00% Fehlerabweichung und sind die ideale Wahl für RS-232, leider gibt es diese Quarze nicht immer und in allen Ausführungen. Wenn Sie dies vor dem Kauf lesen: Bestellen Sie am besten direkt den Quarz mit (Artikel Nr. Z001a für 16 MHz oder Z001b für 14,7456 MHz). Das kostet nicht viel und auch die Fuses des Moduls sind dann bereits korrekt eingestellt.

Ein 16 MHz-Quarz z.B. führt bei 9600 Baud zu einer Fehlerrate von 0,16%, was noch OK ist. Alles unter 0,5% Abweichung ist in der Regel akzeptabel.

Zeitmessungen:
Ein geradzahliger Quarz (z.B. 16 Mhz) hat allerdings auch einen Vorteil: Da alle Zeiten innerhalb des Controllers durch Herunterteilen der Taktfrequenz gewonnen werden, sind diese mit dem 16 Mhz Quarz immer gerade teilbar – der 14.7456 Mhz Quarz dagegen führt zu minimalen! Ungenauigkeiten beim Betrieb einer Uhr oder beim Messen von Zeiten. Hier muss man also abwägen, was einem wichtiger ist. Soll der ATmega die eingebaute Echtzeit-Uhr nutzen, sollten Sie sowieso an die vorbereitete Stelle den separaten Uhrenquarz einsetzen – dann ist die Uhr unabhängig von der Taktfrequenz.

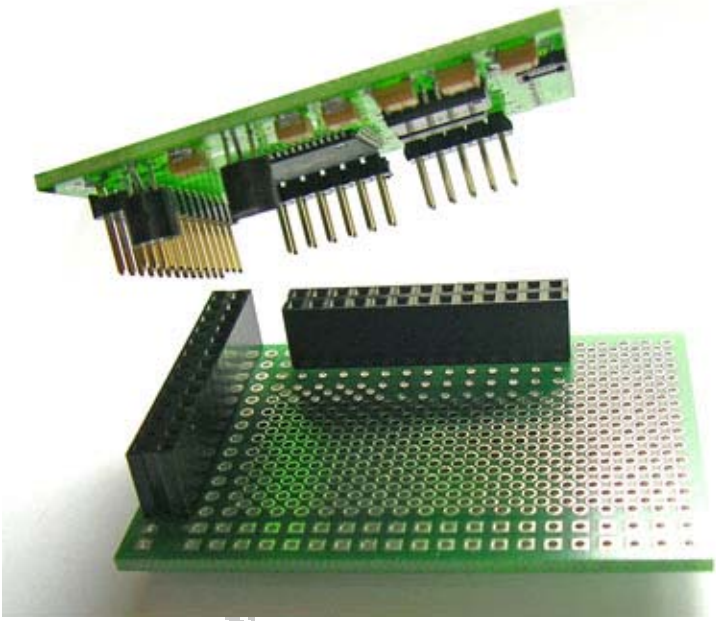
Stromverbrauch:
Ein weitere Aspekt soll noch erwähnt werden, denn er ist für batteriebetriebene Geräte u.U. relevant: Je höher die Taktfrequenz des Mikrocontrollers, desto höher ist auch sein Stromverbrauch (siehe auch Seite 35).

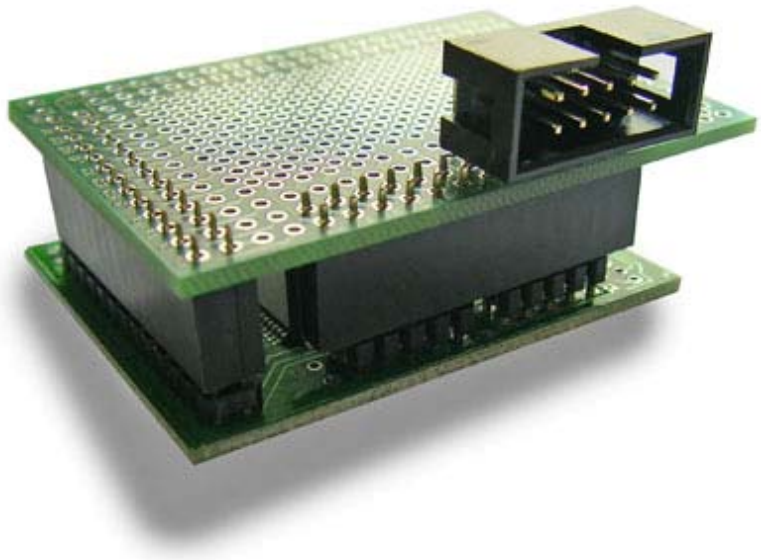
Achtung:
Die Lötpads TX und Tx sind direkt neben den Pads für den Port F angebracht. Da die Rx und Tx-Signale deutlich mehr als 5 Volt betragen können, müssen Sie hier vorsichtig sein. Wenn der Mikrocontroller direkt mit diesen Signalen in Verbindung kommt, wird er und evtl. auch das Display beschädigt oder zerstört.
Seien Sie daher bitte vorsichtig und nutzen Sie diese Signale für nichts anderes als den Anschluss an ein RS232-Kabel.

Die zusätzlich verfügbare Hardware

Zubehör: Mini-Entwicklungsplatine (P005):

Diese doppelseitige und durchkontaktierte Platine hat eine Größe von lediglich 60x51 mm und passt genau hinter das Displaymodul. Mit der Platine werden 2 Steckerleisten sowie ein ISP-Stecker geliefert. Dadurch sind Sie in der Lage ein Gesamtmodul mit weiteren Bauteilen (Relais, Transistoren etc.) zu erstellen. Der mitgelieferte ISP-Stecker wird an seinen Platz gelötet und erlaubt dann eine jederzeitige Programmierung, ohne die Platine jedes Mal vom Modul abziehen zu müssen.



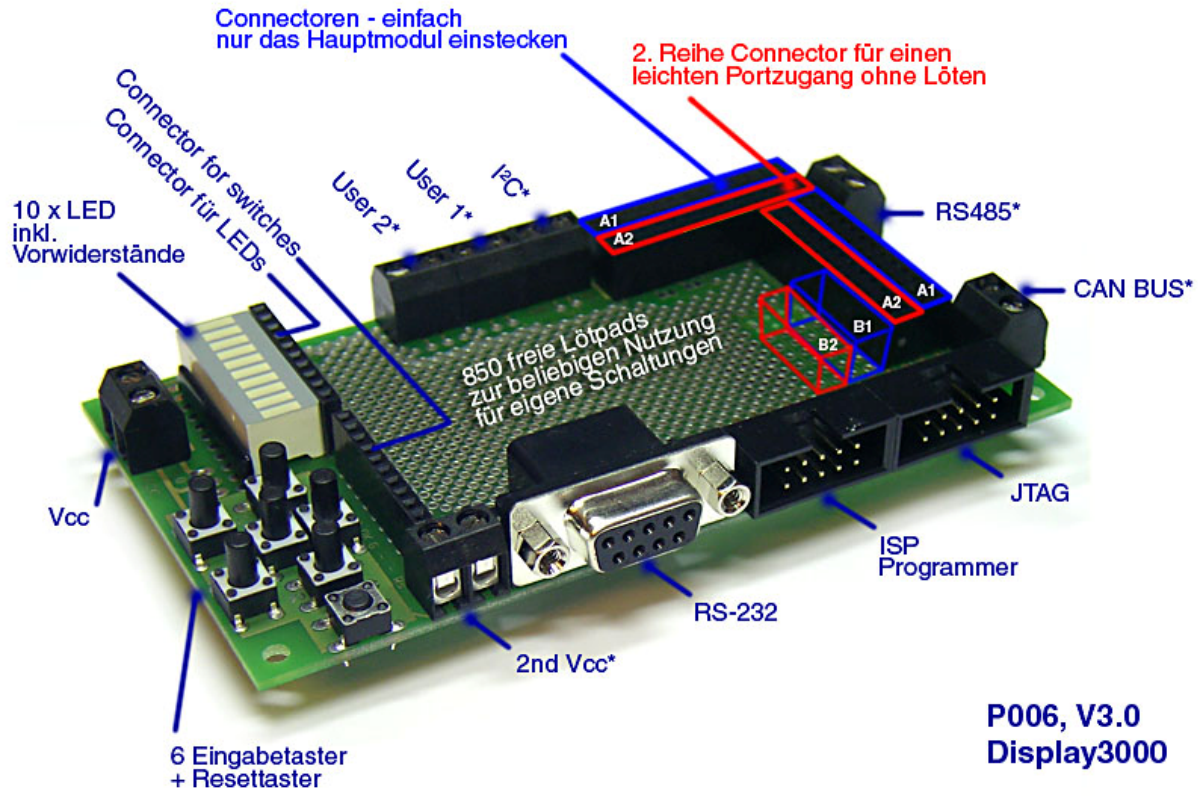


Zubehör: große Entwicklungsplatine (P006)

Ideal für:

- eigene Entwicklungen
- RS485
- RS232
- CAN-Bus
- I²C
- ISP
- JTAG

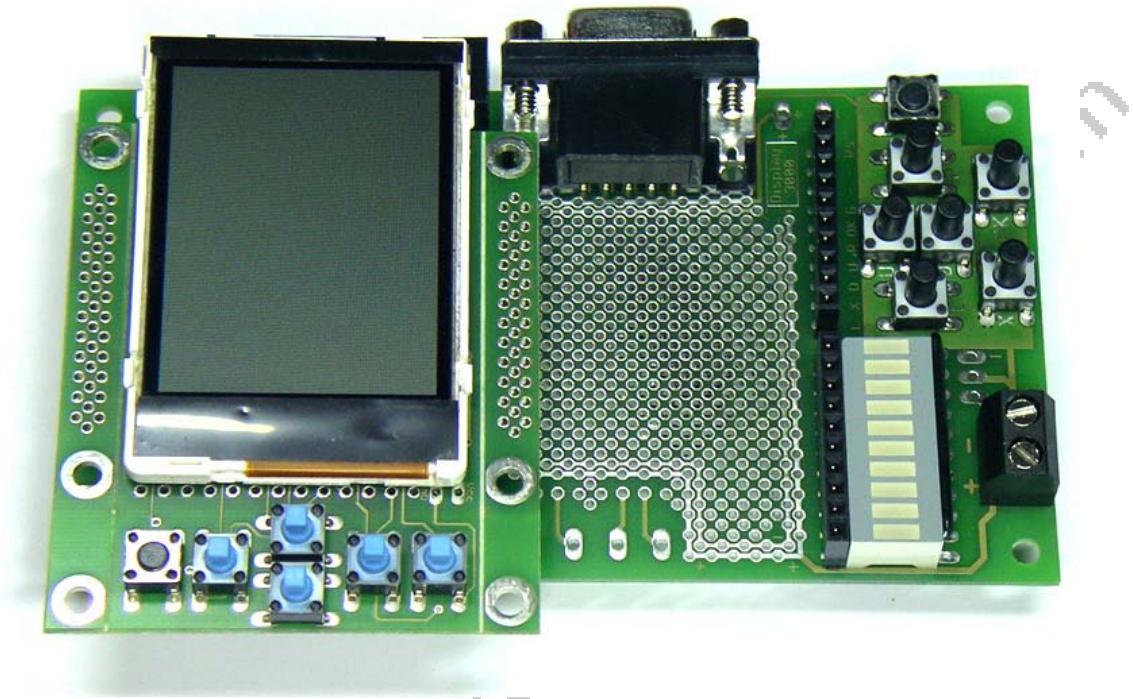
Diese Platine beinhaltet 6 Taster sowie einen Reset-Taster, 10 Leuchtdioden, einen Standard-RS232 Connector, einen ISP Stecker, einen JTAG-Stecker, sowie 4 Steckerleisten, einen Steckkontakt für die Spannungsversorgung sowie einen vorbereiteten Platz für einen größeren Spannungsregler. Über 850 frei verfügbare Lötpunkte erlauben die Erweiterung durch eigene Schaltungen. Hinzu kommen Optionen für: eigene Anschlussterminals, I²C, RS485-Chip und CAN-Bus-Treiber.



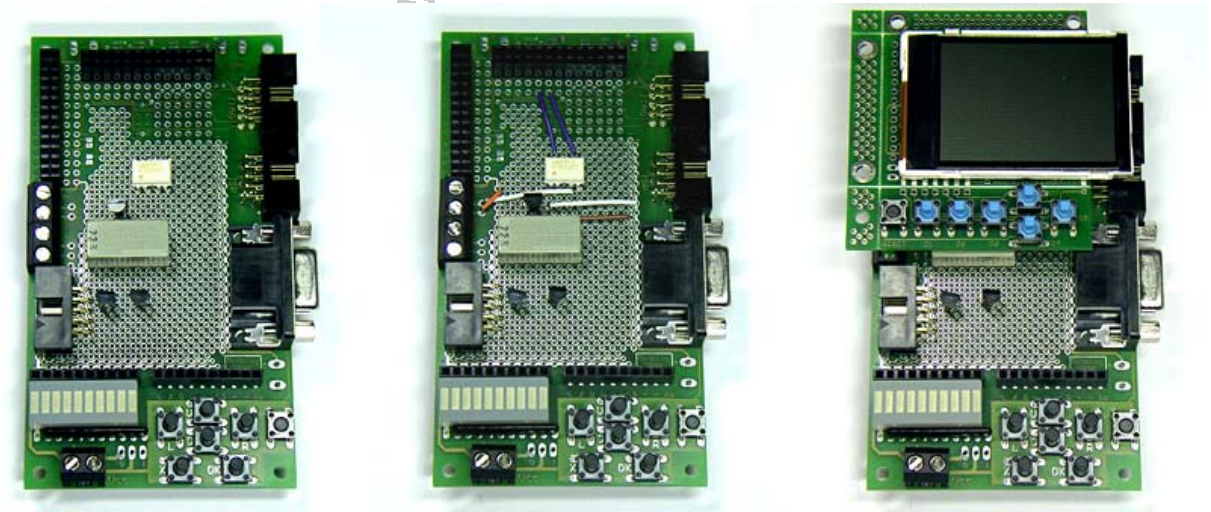
Sie erhalten diese Platine als Bausatz. Die mitgelieferten Bauteile löten Sie je nach Erfordernis ein. Das Manual zu dieser Platine zeigt Ihnen die genaue Belegung und Nutzung. Wir liefern: Platine mit RS232, JTAG und ISP Stecker. 7 Taster, 10 LEDs (BAR), Widerstandsnetzwerk, 4 Stück 2x13 Buchsenleisten, Stromanschlussbuchse, LED Connector, Tasten-Connector.

Natürlich muss dies keine Entwicklungsplatine sein – sie können sie genauso gut auch für eine endgültige Applikation einsetzen. Die Signale für ISP, JTAG, RS485, CAN-Bus, RS232 sind bereits durchgeschleift. Die Taster und die LEDs stehen zur beliebigen Kontaktierung zur Verfügung.

Und so sieht es aus, wenn das Modul (hier ein Bild des Schwestermoduls D072 mit größerem Display) auf der Platine P006 steckt:



Nutzungsbeispiel aus der Praxis (hier: 2 Optokoppler, Relais, 3 Transistoren)



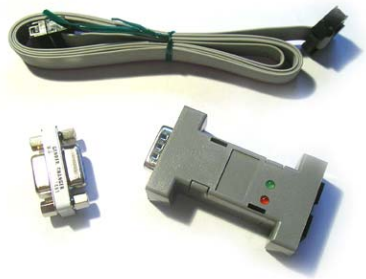
1. Bauteile platzieren

2. Verdrahten,
entweder oberhalb
und/oder unterhalb
der Platine

3. Modul aufstecken
Fertig.
Schaltungsänderungen?`
Einfach das Modul
abziehen - das Hauptmodul
bleibt wie neu.

Zubehör: ISP Programmieradapter

Ein ISP Programmieradapter ist zwingend notwendig, um das am PC kompilierte Programm in das Controllermodul zu übertragen. Die ISP-Programmer gibt es bei uns in drei verschiedenen Varianten:

Parallelschnittstelle (Druckerport – Centronics)	**Serielle Schnittstelle (RS232)**	**USB-Schnittstelle (bestehend aus USB-RS232 Adapter und seriellem Programmer**
		

Am flexibelsten ist im Grunde der parallele Programmieradapter – leider bieten immer weniger Rechner standardmäßig eine parallele Schnittstelle an. Ein USB-Parallelumsetzer funktioniert übrigens nicht mit einem parallelen ISP-Programmer (zumindest ist uns dies nicht bekannt – sollten Sie gegenteilige Erfahrung gemacht haben, wären wir über ein Feedback dankbar).

Das Entfernen des Montagerahmens / Tasteneinheit

Oft wird das gesamte Modul unterhalb einer Frontplatte montiert, daher wurde häufig nach Modulen mit Montagehaltern gefragt. Wir liefern dieses Modul daher standardmäßig mit solchen Haltern aus: rechts und links vom eigentlichen Modul befindet sich jeweils ein ca. 7mm breiter Streifen mit je zwei 3 mm Montagebohrungen. Ein genauerer Blick auf diese Halter zeigt, dass diese mit einer Perforation versehen sind. Sollten Sie aufgrund von Platzproblemen oder anderer Gründe diese Montagehalterungen entfernen wollen, so können Sie diese einfach mit einer Zange abknicken. Das gleiche gilt für den Tastenbereich unterhalb des Displays. Dieser lässt sich separat oder zusammen mit dem Rahmen entfernen – so wie es Ihre Einbausituation notwendig macht.

Das obige Foto zeigt mögliche Alternativen: einmal ohne Halterungen und einmal ohne Halterungen und Tasterfeld. Selbstverständlich ist die (nicht abgebildete) Option „ohne Taster aber mit Halterungen“ ebenfalls möglich.

Das Entfernen des Montagerahmens

Der Montagerahmen ist bereits vorperforiert und wird mit einer Zange ganz einfach abgeknickt. Entfernen Sie vor diesem Vorgang das Display, um dieses nicht versehentlich zu beschädigen.

Tipp für eine sauberere Entfernung von Montagehalterung und Tastenfeld: Ritzen Sie vor der Entfernung mit einem scharfen Messer (z.B. Teppichmesser oder Skalpell) die Platine beidseitig entlang der Perforation ein oder zwei mal an, dann lässt sie sich sehr sauber entfernen.

Das Entfernen des Tastenfelds

Die Entfernung des Tastenfeldes bedarf eines weiteren Schrittes, denn die Taster sind ja mittels Leiterbahnen bereits mit Port D des Controllers verbunden. Um eine saubere Trennung der sieben Leiterbahnen zu erreichen, sollten Sie diese mit einem scharfen Messer zuerst einmal auf Höhe der Perforation durchschneiden.

Der Stromverbrauch des Moduls

Der Stromverbrauch des Gesamtmoduls ist abhängig von verschiedenen Faktoren:

a) **Geschwindigkeit** des ATMega Controllers (je schneller, desto höher der Stromverbrauch des Moduls)

b) **Displaybeleuchtung:** Die Displaybeleuchtung benötigt selbstverständlich auch Strom um adäquat leuchten zu können. Die Displaybeleuchtung wird mit knapp 16 Volt Leerlaufspannung betrieben. Um diese zu erzeugen, wird die 5 Volt Spannung des Moduls zwei Mal verdoppelt – dadurch steigt zwangsläufig auch der Stromverbrauch der Beleuchtung um das 4-fache (irgendwo her muss die Leistung ja kommen – ein Perpetuum mobile gibt es nicht). Mit sinkender Beleuchtungsstärke, sinkt auch der Stromverbrauch des Moduls.

Die folgende Tabelle gibt Aufschluss über einige veränderbare Parameter und deren Einfluss auf den Stromverbrauch. Das Display selbst braucht übrigens nur einen vernachlässigbar geringen Strom (abgesehen von der Beleuchtung). Hier lohnt es sich nicht, dieses zeitweise abzuschalten oder weniger intensiv zu nutzen.

Verbrauch Gesamtmodul (Controller 100% aktiv, ohne Idle-Modus etc.)

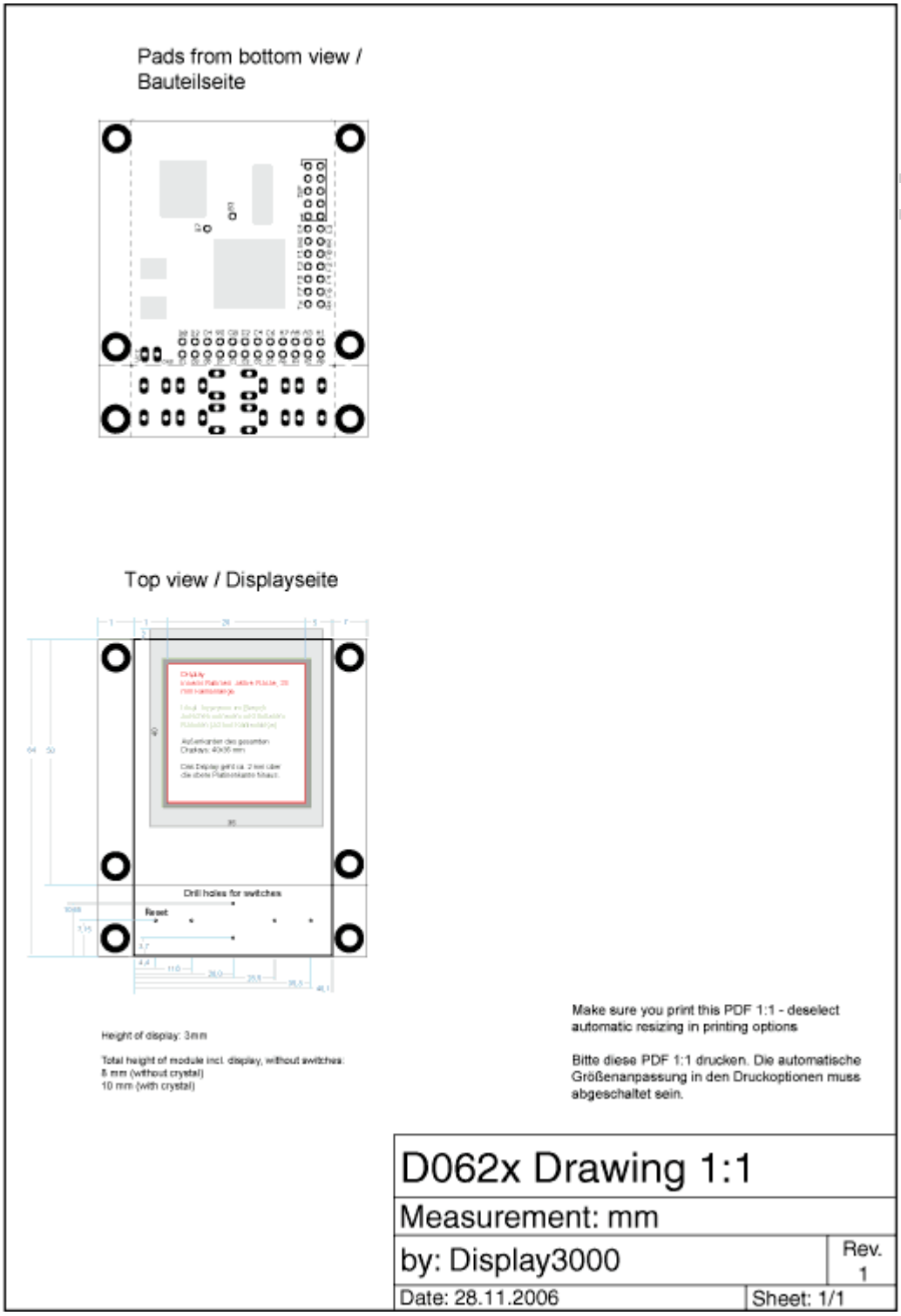
Controller-Takt	**Displaybeleuchtung 0% (aus)**	**Displaybeleuchtung 50%**	**Displaybeleuchtung 100%**
8 Mhz	19 mA	41 mA	58 mA
16 Mhz	28 mA	50 mA	65 mA

Eine Anmerkung zur Versorgungsspannung:
Der integrierte Spannungsregler ist ein typischer Längsregler der die überflüssige Spannung in Wärme umsetzen muss. Der Spannungsregler verträgt zwar 20 Volt Eingangsspannung, kann dies aber nicht auf Dauer leisten:
Bei einem angenommen Verbrauch des Moduls von 60mA müssten bei 20 Volt Eingang 15 Volt * 60 mA = 0,9 Watt an Wärme abgeführt werden. Das schafft der kleine Spannungsregler nicht lange und die integrierte Wärmesicherung schaltet ihn zur Abkühlung ab. Selbst bei einem Anschluss an 12 Volt wird er bereits sehr heiß. Wir empfehlen daher einen dauerhaften Anschluss an max. 9 Volt.
Sollten Sie eine höhere Eingangsspannung zur Verfügung haben empfehlen wir folgendes:

a) Reduzierung des Stromverbrauchs durch Dimmen / Abschalten der Displaybeleuchtung (siehe auch Seite 35).
b) Und/oder: Nutzung eines zusätzlichen externen Spannungsreglers mit Kühlkörper (z.B ein 7808, der dem Modul dann 8 Volt zur Verfügung stellt).

Maßzeichnung und Pads

Auf der CD befindet sich eine separate PDF-Datei mit der Pad-Anordnung und den Maßen. Die Datei auf der CD liegt maßstabsgetreu im Maßstab 1:1 vor, die hier nachfolgende Grafik ist etwas verkleinert. Bzgl. Port B5/B6 beachten Sie bitte auch den Errata-Eintrag auf Seite 9.



Technische Daten Display-Modul-Bausätze:

Artikelnummer D062x:

Maße (Breite x Länge):

Mit Rahmen, und Tastenfeld:	ca. 54 x 64 mm
Mit Rahmen, ohne Tastenfeld:	ca. 54 x 50 mm
Ohne Rahmen, mit Tastenfeld:	ca. 41 x 64 mm
Ohne Rahmen, ohne Tastenfeld:	ca. 41 x 50 mm
Höhe:	ca. 8 mm inkl. Display

Versorgungsspannung: 4,5 bis 20* Volt Gleichspannung
(* max. Spannung siehe Seite 35)

Prozessor (je nach bestellter Ausstattung):

	Programmspeicher	RAM	Eeprom	Takt
ATMega 128	128 KByte	4 KByte	4 KByte	Max. 16 Mhz
ATMega 2561	256 KByte	8 Kyte	4 KByte	Max. 16 Mhz
AT90CAN128	128 KByte	4 KByte	4 KByte	Max. 16 Mhz

Display: 132x132 Pixel, 65.536 Farben
Aktive diagonale Fläche: 1,5“ (38mm)

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

Fusebit DCBA: Speed von 0001 (1MHz intern) **auf 0100 (8 MHz intern)** wenn Sie einen Quarz mitbestellen, ist die Fuse bereits auf diesen Quarz eingestellt (Fusebit dann in der Regel 1111 (External crystal, High speed).
Fusebit I: Preserve EEPROM when chip erase
Fusebit P: ATMega 128 Modus (bei ATMega128)

Eeprom und Programmspeicher wurden zu Testzwecken bereits programmiert.

Sollten Sie den JTAG-Zugriff benötigen, so können Sie **Fusebit F:** auf **JTAG disabled** ändern. Port F4 bis F7 stehen nicht zur Verfügung, solange die JTAG-Option eingeschaltet ist.

Lieferant:

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

Autor dieses Manuals: Peter Küsters.

Haftung, EMV-Konformität

Wenn Sie diesen Bausatz fertig gestellt haben bzw, diese Baugruppe durch Erweiterung bzw. Gehäuseeinbau betriebsbereit gemacht haben, gelten Sie nach DIN VDE 0869 als Hersteller und sind verpflichtet, bei der Weitergabe des Gerätes alle Begleitpapiere mitzuliefern und auch Ihren Namen und Ihre Anschrift anzugeben.

Geräte, die aus Bausätzen selbst zusammengestellt werden, sind sicherheitstechnisch wie ein industrielles Produkt zu betrachten.

Derjenige, der den Bausatz zusammenbaut und in einem Gehäuse montiert, gilt als Hersteller und ist damit selbst für die Einhaltung der geltenden Sicherheits-, EMV- und Entsorgungsvorschriften verantwortlich.

Für Schäden die durch fehlerhaften Aufbau entstanden sind, direkt oder indirekt, ist die Haftung generell ausgeschlossen.

Bei der Lieferung von Fremdprodukten als auch Software gelten über diese Bedingungen hinaus die besonderen Lizenz- oder sonstigen Bedingungen des Herstellers.