

Zusatzhandbuch für das Komplettmodul

D072

mit Atmel Mikrocontroller:

ATMega128A oder **ATMega2561** oder **AT90CAN128**

Version 2.11 (für Board D072 – V10)

Stand: 22. August 2012



Wichtig!

Die Verfügbarkeit des Moduls D072 läuft aus. Bestandskunden können dies noch jederzeit nachbestellen, da wir genügend Module auf Lager haben. Ihre Ersatzteilversorgung oder Verfügbarkeit für Kleinserien ist also nicht gefährdet.

Neukunden sollten jedoch unbedingt das Nachfolgemodul D074s oder das etwas längere D074x wählen.

Vorhandene D072 Software läuft ohne Änderung auch auf dem D074 (Ausnahme: Abfrage der Tasten ist nun Port C statt Port D).

© 2012 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 ATmega-Moduls D072.

Dieses Modul beinhaltet neben einem Atmel Controller (je nach geordeter 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 ☺



Vorderansicht

Montagehalter und Tasteneinheit kann schnell und einfach entfernt werden.



Rückansicht

weitere Bauteile befinden sich auf der anderen Platineseite. Am oberen Rand ist hier das Display aufgesteckt. Jeder Port ist deutlich sichtbar beschriftet.



Nach dem (kinderleichten) Entfernen der Montagehalterungen und der Tasteneinheit ist das Gesamtmodul lediglich so groß wie das Display selbst

(c) 2007 www.Display3000.com

Sie erkennen rechts und links vom Modul noch Befestigungen mit Montagebohrungen. Diese 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 42).

Übrigens: nachdem wir oft danach gefragt wurden, haben wir das Testprogramm, mit welchem das Modul ausliefern mit auf die CD kopiert: Portcheck im Verzeichnis der Bascom-Basic-Programme.

Sie müssen diesen Bausatz noch vervollständigen, indem Sie das Display aufstecken sowie die notwendigen Kabel und Steckverbinder anlöten. Haben Sie ein Modul D072 erworben, werden von Ihnen zudem noch die Taster eingesteckt und eingelötet.

Übrigens: Wenn Sie Ihren TFT-Monitor, der vor Ihnen auf dem Schreibtisch steht nehmen und auf den Boden werfen ist er defekt. Das gleiche passiert mit unseren TFT-Displays. Gehen Sie daher sorgsam damit um. Fällt Ihnen das Display hin oder drücken Sie zu stark auf die Frontscheibe, wird es schnell zerstört.

Stecken Sie das Display erst zum Schluss auf. Tipp: Wir liefern nur getestete Neuware – das gilt auch für das Display. Belassen Sie die Schutzfolie möglichst lange auf dem Display – es schützt vor Kratzern während der Anschluss- und Einbauphase. Da die Folie kristallklar ist, stört sie (bis auf die leuchtend gelbe Lasche zum Abziehen, die sich aber auch einfach abschneiden lässt) nicht weiter.

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 75-seitige Programmierhandbuch auf der CD.

Noch eine Anmerkung, damit keine Verwirrung aufkommt: einige Fotos stammen noch vom Vorgängermodul dieses D072; kleine Details sehen also evtl. etwas anders aus – für wichtige Details sind natürlich aktuelle Fotos verwendet worden.

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. Aber gerne geben wir Ihnen einen Tipp, wenn wir können.

ACHTUNG:

- 1) Stecken Sie niemals das Display auf oder nehmen es ab, solange Spannung am Modul anliegt.
- 2) Kontrollieren Sie vor Anlegen der Spannung, ob Sie nicht versehentlich den Displastercker um eine Position versetzt aufgesteckt haben – dann würde das Display beim Anlegen der Spannung zerstört werden.
- 3) 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.

Neu in V10 gegenüber V9:

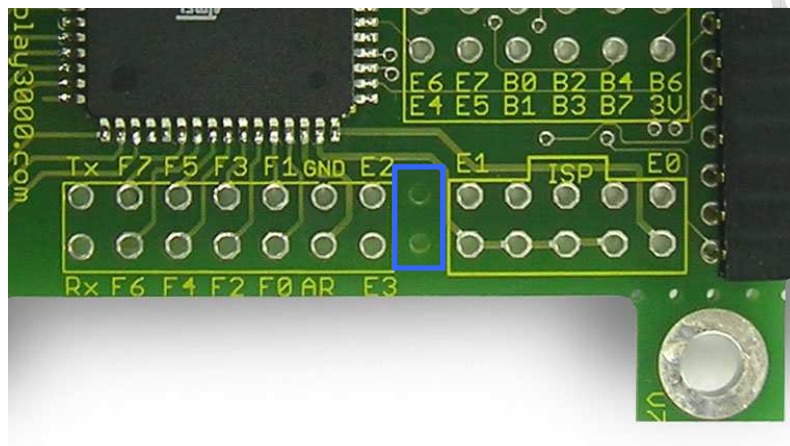
- CAN Transceiver mit auf der Platine bei Bestellung mit einem AT90CAN128 Transceiver.
- 2 bislang unbelegte Pads neben dem ISP Stecker sind nun von CAN L und CAN H belegt (wenn AT90CAN128 vorhanden, ansonsten weiterhin unbelegt)
- Einzel konfigurierbare Abschaltungsoptionen zum Stromsparen
- Tristate-Ausgänge für das Display – komplettes Wegschalten vom SPI Bus möglich
- Der RS232 Transceiver ist standardmäßig nicht mit dem ATmega verbunden. Dies erlaubt den Einsatz der Ports D2 und D3. Zwei, mit einem Tropfen Lötzinn zu schließende, Lötbrücken erlauben jedoch eine schnelle Verbindung.

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 D072 nur auf, und gelötet wird stattdessen auf der Experimentierplatine – d.h. Ihr Modul ist auch nach vielem Probieren noch wie neu.



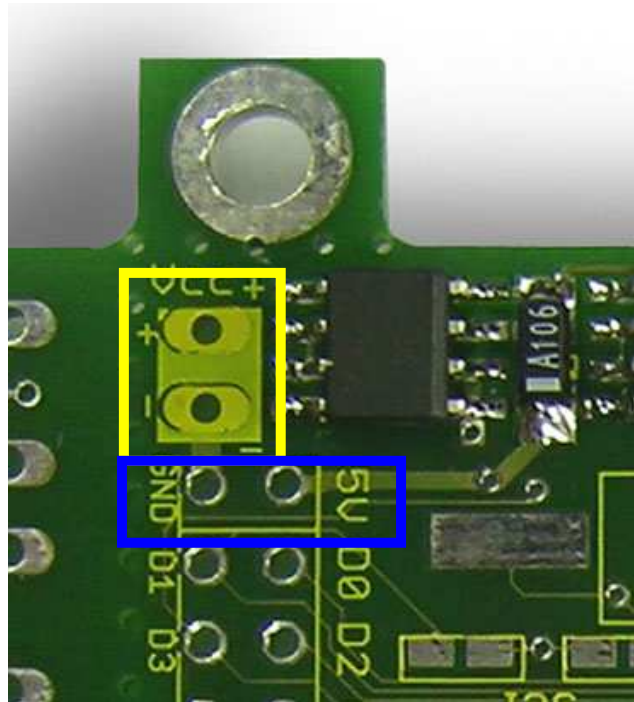
Wenn Sie, wie auf Seite 13 gezeigt, den ISP-Stecker Ihres Programmiergerätes direkt mit dem Controllermodul verbinden möchten, dann dürfen Sie keine komplette Steckerleiste einsetzen, sondern müssen diese vorher in einen Stecker mit 2x5 und einen Stecker mit 2x7 Kontakten trennen (oder vorher mit einer Zange die Pins aus dem Kunststoffkörper ziehen)

– 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. Der blaue Bereich sollte dann also freibleiben.

Neu in Version V10: Bislang blieben diese beiden oben genannten Pads unbelegt, mit dieser neuen Version 10 liegen an diesen beiden Pads (d.h. im blauen Bereich) nun CAN L und CAN H an. Dies betrifft Sie jedoch nur, wenn Sie ein Modul D072 mit AT90CAN128 Mikrocontroller und mit CAN Transceiver erhalten haben.

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 (**zur max. Spannung lesen Sie bitte unbedingt auch den Hinweis auf Seite 43**) anlegen. Der Prozessor und die RS-232 Schnittstelle wird mit 5 Volt betrieben, die Displayelektronik mit 3 Volt. Die Spannungsregelung 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). Wenn Sie für die Analogeingänge die interne Referenzspannung nutzen möchten, sollten Sie jedoch trotzdem mind. 6 Volt an den Eingang anlegen um eine stabile 5 Volt Versorgung zu erhalten.

ACHTUNG:

Der 5-Volt Spannungsregler kann max. 160mA liefern und wird bereits mit ca. 50-65mA belastet (aus diesem Stromzweig wird auch die ca. 14-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 **maximal** 5V an (minimal ca. 4,5 Volt).

Legen Sie dann aber niemals eine höhere Spannung als 5 Volt an – eine Zerstörung der Displaybeleuchtung und des Controllers kann die Folge sein.

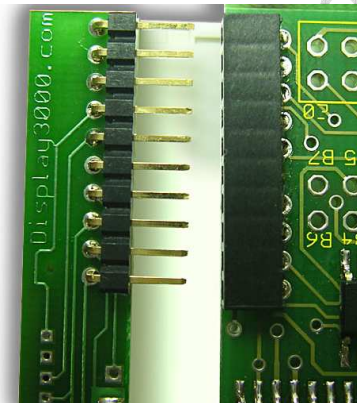
Inbetriebnahme

1. Stecken Sie die Pfostenstecker in die vorgesehenen Löt pads und verlöten Sie sie. Die Stecker werden von der Bauteilseite aus eingesteckt.
2. Löten Sie die Taster ein (von der Displayseite, also der Moduloberseite aus!)
3. Stecken Sie das Display auf und legen Sie eine Spannung an die beschrifteten Kontakte Vcc und GND an. Jedes Modul und jedes Display wurde vor der Auslieferung getestet. Auf dem Modul befindet sich noch das Testprogramm, welches nun startet.

Das Display wird wie folgt montiert:

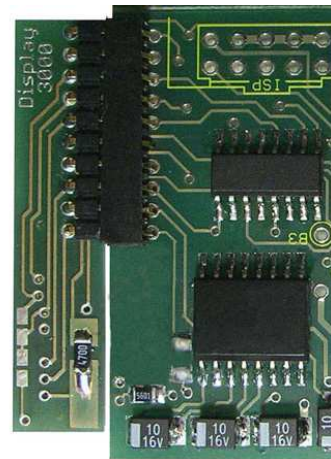
1) Legen Sie das Display vor sich auf den Tisch. Dann legen Sie die Modulplatine auf das Display, so dass sich Stecker und Buchse gegenüber liegen

Anmerkung: Ein Blick auf die Verbindung Display-Displayplatine zeigt einen Spezialstecker im Innern, der die Platine mit dem Display verbindet. Bei stärkerer Belastung (z.B. Fallenlassen des Displays) kann dieser Stecker aufspringen und muss dann von Ihnen wieder zusammengedrückt werden. Wenn Sie auf Nummer sicher gehen möchten, **fixieren Sie die kleine Steckplatine mit dem Display (bzw. die beiden Steckerhälften) durch einen Tropfen Klebstoff** auf der linken Seite (rechts ist die Platine ja bereits durch uns fixiert worden).



2) Halten Sie mit Daumen und Zeigefinger der linken Hand die kleine Platine des Displays fest und mit der rechten Hand die Modulplatine. Dann schieben Sie Stecker und Buchse zusammen.

Achten Sie immer darauf, den Stecker nicht versehentlich um einen Kontakt zu versetzen – bei Anlegen einer Spannung würde das Display u.U. zerstört.



3) Fertig

Diese Steckverbindung hat den Vorteil, dass Sie das Display jederzeit weiter entfernt montieren und mittels eines Kabels die Verbindung verlängern können. Stecker und Buchse hierzu sind bei uns im Shop erhältlich. Beachten Sie aber, dass das Display durch eine Verbindung per Kabel schneller gestört werden kann. 20-30 cm haben wir bereits erfolgreich getestet.

Wenn Sie das Display wieder abnehmen möchten, führen Sie die obige Anleitung in umgekehrter Reihenfolge durch.

Quick Guide: Häufig genutzte Umkonfigurierungen

Durch die vielen möglichen Optionen ist das Handbuch recht umfangreich geworden. Hier ist die Gefahr groß, dass jemand eine notwendige Umkonfigurierung übersieht. Daher haben wir für in der Praxis am häufigsten vorkommenden Umkonfigurierungen nachfolgend eine Kurzanweisung aufgeführt. Diese zeigt die notwendigen Änderungen inkl. Verweis auf eine ausführlichere Erläuterung im Handbuch.

1) [Nutzung der seriellen Schnittstelle UART1](#) (RS232 - High Pegel)

- Bei Nutzung als TTL-Pegel an Port D2 und D3 (RX1/TX1): keine Änderung notwendig
- Für Nutzung als RS232 (High Pegel): Jumper J6, J7 schließen (siehe Seite **Fehler! Textmarke nicht definiert.**)
- **UART0** (Port D0 und D1) kann nicht so einfach genutzt werden, da diese Ports auch vom ISP Programmieradapter genutzt werden (mehr dazu im entsprechenden RS232 Kapitel)

2) [Komplettabschaltung nutzen](#) (sofern diese Option mitbestellt wurde):

- Jumper JP Öffnen
- Die Jumper J2, J3, J11, legen die abzuschaltenden Bereiche fest (siehe Seite 15 ff.)
- Nutzung: PortG.I muss auf Ausgang programmiert werden.
- PortG.I=1 : Abgeschaltet
- PortG.I=0 : Eingeschaltet
- Nach dem Aufwecken müssen das Display und eine evtl. angeschlossene SD Karte neu initialisiert werden, denn sie waren ja ohne Versorgungsspannung.

3) [Displaybeleuchtung schalten/dimmen](#) (sofern diese Option mitbestellt wurde):

- Jumper J10 muss geöffnet werden (siehe Seite 17 ff.)
- Port B7 muss auf Ausgang programmiert sein. Schalten mit Port B7; Dimmen mit PWM Kanal C (*Pwm1c*)

4) [Analoge Eingänge](#) nutzen

- AGND mit GND verbinden (Jumper J9)
- ARef festlegen (entweder interne Spannung per Software programmieren oder Verbinden mit 5V durch Jumper J8)
- Die ersten vier AD Wandler (Port F0 bis F3) sind direkt nutzbar. Für die Nutzung der restlichen vier Wandler (Port F4 bis F7) muss JTAG in den Fuses abgeschaltet werden (Alternativ: JTAG per Software zur Laufzeit abschalten)

5) [CAN BUS Nutzung](#) (siehe Seite 25)

- Widerstand RS festlegen (i.d.R. reicht 0 Ohm, d.h. eine Lötbrücke durch einen Tropfen Lötzinn auf das Pad RS)
- J2 konfigurieren
- Tasterjumper J1C und J1D umkonfigurieren, da ansonsten die Taster links/rechts mit den CAN-Ports verbunden sind und nicht mehr nutzbar sind.

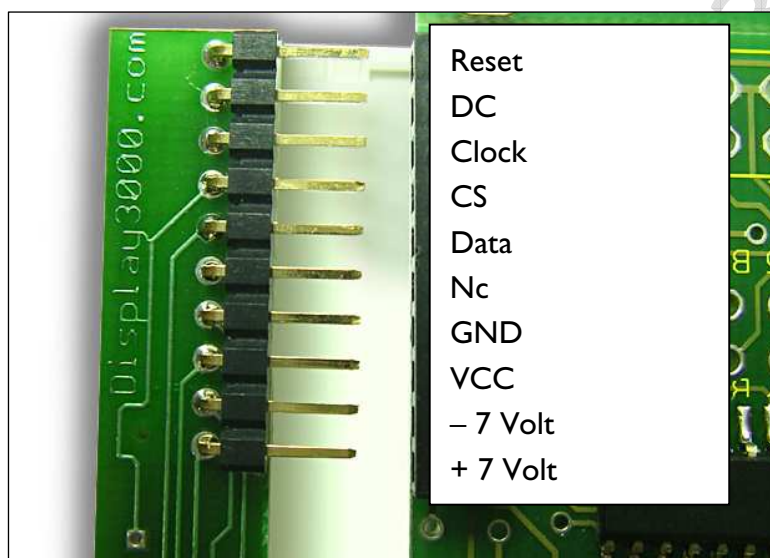
Der Prozessor ist zu Testzwecken bereits vorprogrammiert und zeigt Ihnen auf dem Bildschirm den Status der verfügbaren Ports an. Anmerkung: Port F4 bis F7 zeigen hier „0“. **Wir liefern die Boards mit aktivierter JTAG-Schnittstelle aus, d.h. die Ports F4 bis F7 sind erst nach Deaktivierung nutzbar. Tipp: die temporäre JTAG-Deaktivierung (und -Aktivierung) ist auch per Software zur Programmlaufzeit möglich (siehe Datenblatt des Mikrocontrollers).**

Programmierschnittstelle

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 Board und erlaubt das Überspielen Ihres am PC erstellten Programms. Mehr zum Anschluss des ISP-Programmiers adaptors erfahren Sie auf Seite 13.

Die Anschlüsse des Displays

Im separaten Programmierhandbuch des Displays wird die Ansteuerung des Displays sowie der Steuerleitungen im Einzelnen erläutert. Sie können das Display also auch an andere Systeme oder eigene Platinen anschließen. Die Belegung der Steuerleitungen ist im folgenden Bild zu erkennen:

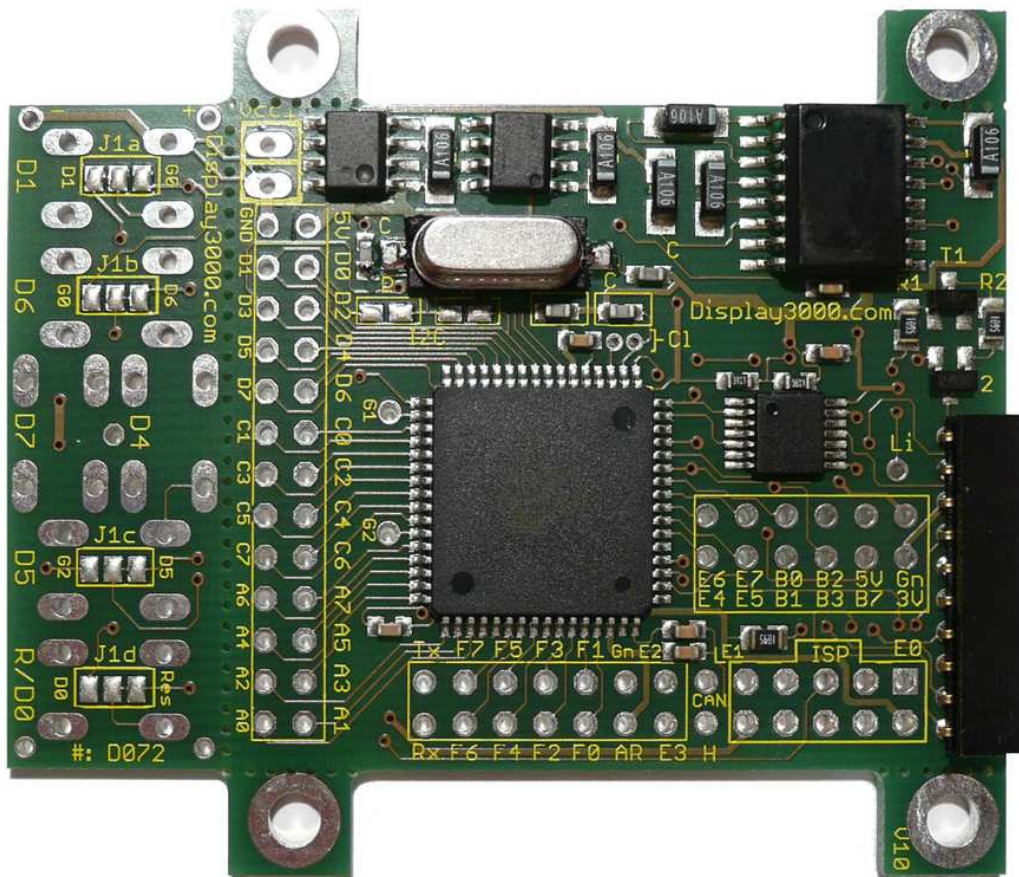


Das Modul bietet einige Sonderoptionen, die in den nachfolgenden Kapiteln ebenfalls beschrieben werden. **Wenn Sie diese Optionen nicht mitbestellt haben, müssen Sie sie erst nachrüsten, bevor diese nutzbar sind.**

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
8. Stromsparmmodus (konfigurierbare Abschaltung verschiedener Verbraucher auf dem Board)

Die Löt pads des Moduls D072

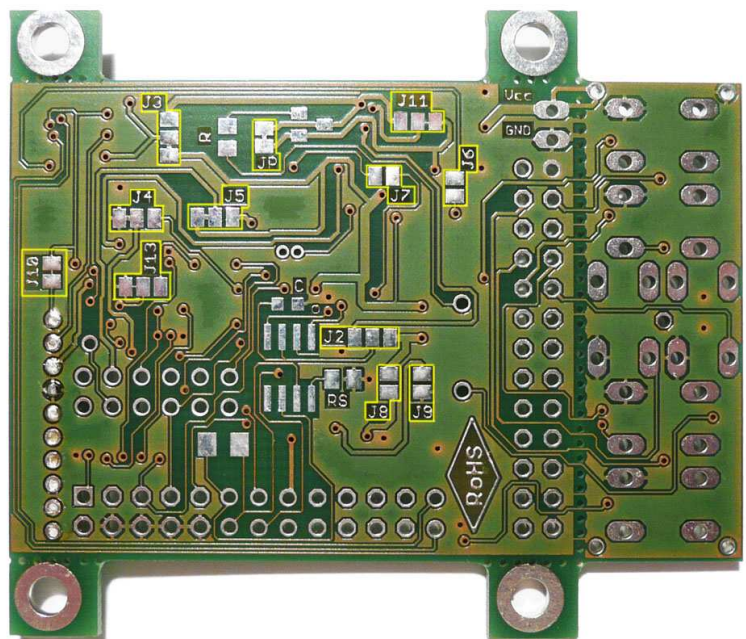
Das Modul wird mit Bestückungsdruck ausgeliefert, so dass die Portbelegung diesem Druck jederzeit entnommen werden kann.



Es gibt zudem auf der Platinenvor- derseite und -rückseite diverse Lötbrücken für unterschiedliche Einsatzgebiete. Jede dieser Brücke ist mit einer entsprechenden Nummer gekennzeichnet und wird im Verlauf des Manuals beschrieben. Am Ende des Manuals (ab Seite 31) finden Sie eine Tabelle mit einer Auflistung und Zusammenfassung der einzelnen Lötbrücken.

Ein Aspekt vorab:

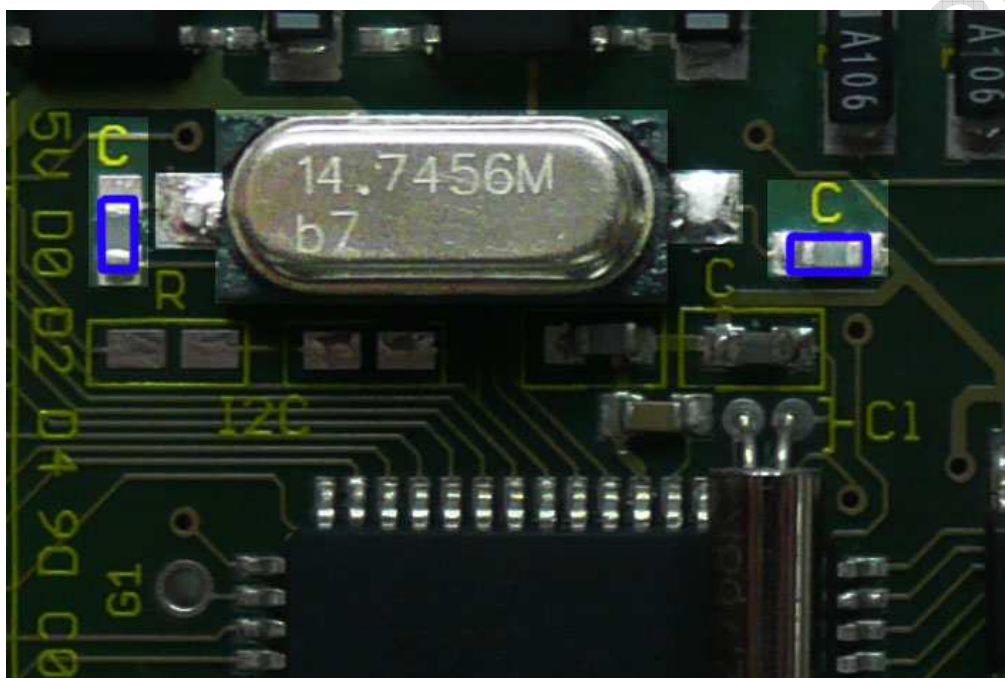
Für die **Nutzung der RS232 Schnittstelle** müssen Sie zuerst zwei Lötbrücken schließen. Bitte beachten Sie dazu **Seite 31**.



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 rechts und links des Quarzes – die beiden Felder für die Kondensatoren 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 erhalten (Bauform 603 – Achtung! Das ist sehr klein).



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 daher nicht mit den Einstellungen der Fuses herum! Rufen Sie im Zweifelsfall lieber an.**

Übertakten

Wenn Sie die Fuse für den Takt auf 1111 gesetzt haben, können Sie jeden beliebigen Quarz mit einem Takt > 8Mhz einsetzen. Der Controller ist offiziell auf 16 MHz limitiert, kann aber oft mit bis zu 20 MHz betrieben werden.

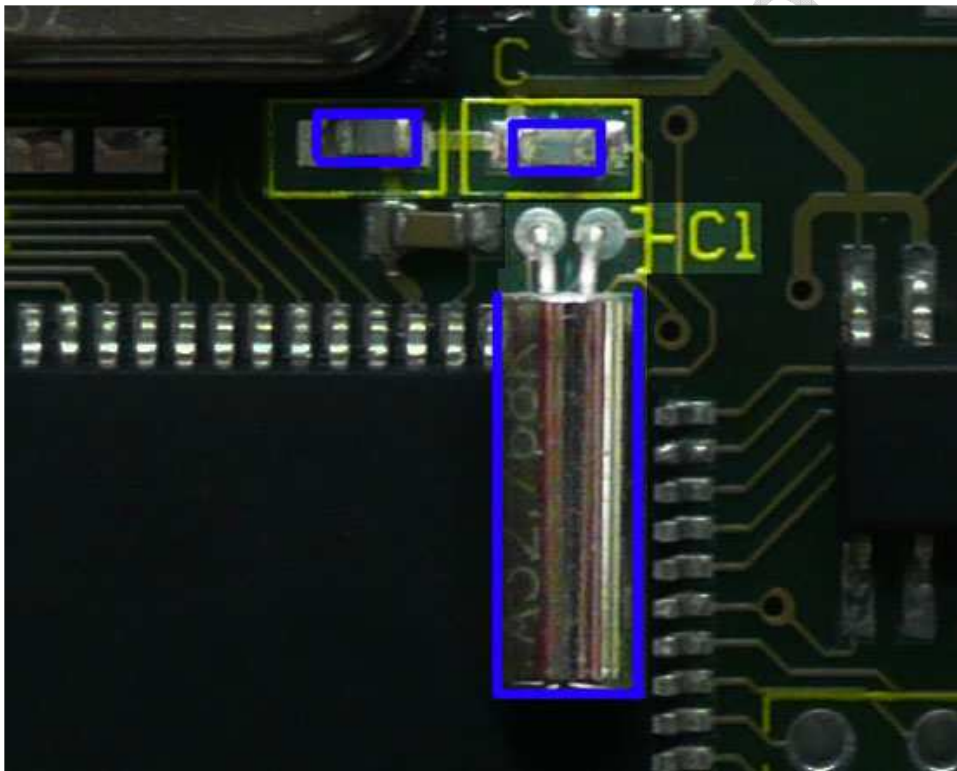
Aber Vorsicht: Der empfindlichste Bereich im Controller ist das eingebaute Eeprom. Während 10% Frequenzerhöhung in der Regel unproblematisch ist, können höhere Frequenzen dazu führen, dass als erstes das Eeprom nicht korrekt geschrieben oder gelesen wird – während alle anderen Bereiche des Controllers 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 24.

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 (ideal für die Realisierung einer Uhr) 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 mit „C1“ gelb markierten Pads gelötet. Unser Tipp: Klappen Sie die beiden Anschlussdrähte des Quarzes vor dem Einlöten einfach um 90° um, so dass er auf dem Controller zum Liegen kommt (siehe auch Foto). 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 Betrieb des Uhrenquarzes noch zwei Kondensatoren von 22pF. Daher haben wir hier zwei Felder vorbereitet: Direkt oberhalb der beiden Löt pads für den Uhrenquarz erkennen Sie die mit dem gelben Kasten gekennzeichnete Lötfelder. Diese sind für 2 Stück SMD Kondensatoren (22pF, BF 603) vorbereitet. Sie brauchen nur bei Verwendung eines ATmega2561 oder AT90CAN128 bestückt werden.

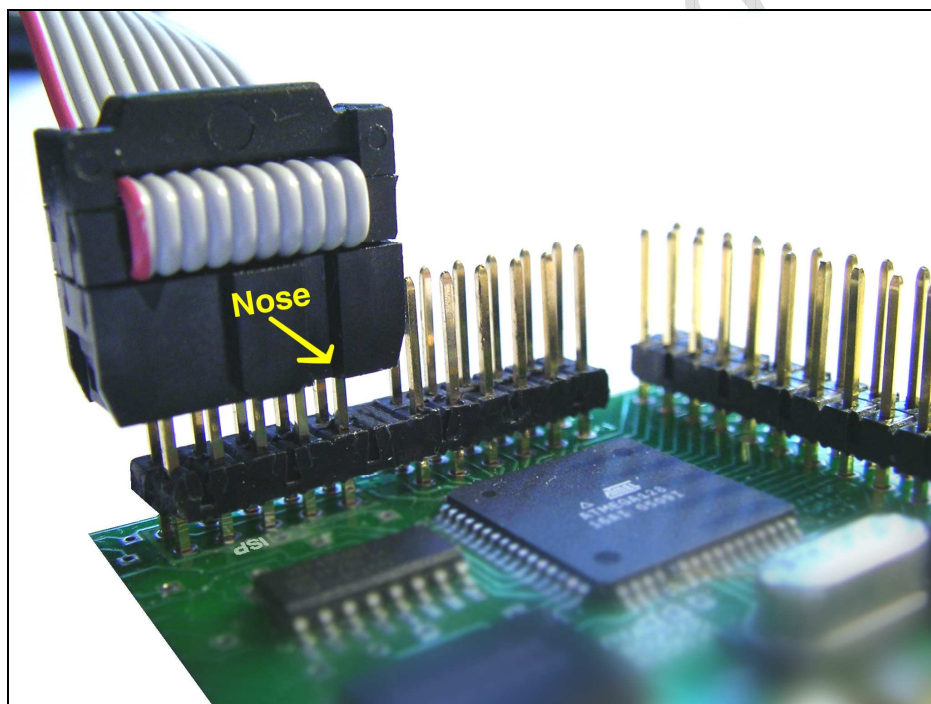
Der ISP-Stecker

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.

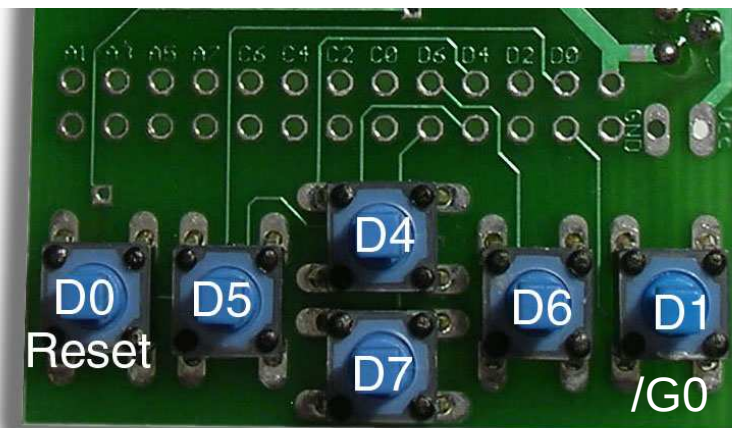
Bitte stecken Sie ihren ISP-Programmer (bei uns im Shop unter www.shop.display3000.com in drei Anschlussvarianten erhältlich: Parallel, Seriell, USB) so wie im folgenden Bild gezeigt auf. Der Stecker mit seinen 10 Kontakten wird auf die Leiste gesteckt, die ebenfalls 2 Reihen mit je 5 Steckern zeigt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

Die „Nase“ des Steckers muss in Richtung der Chips zeigen.

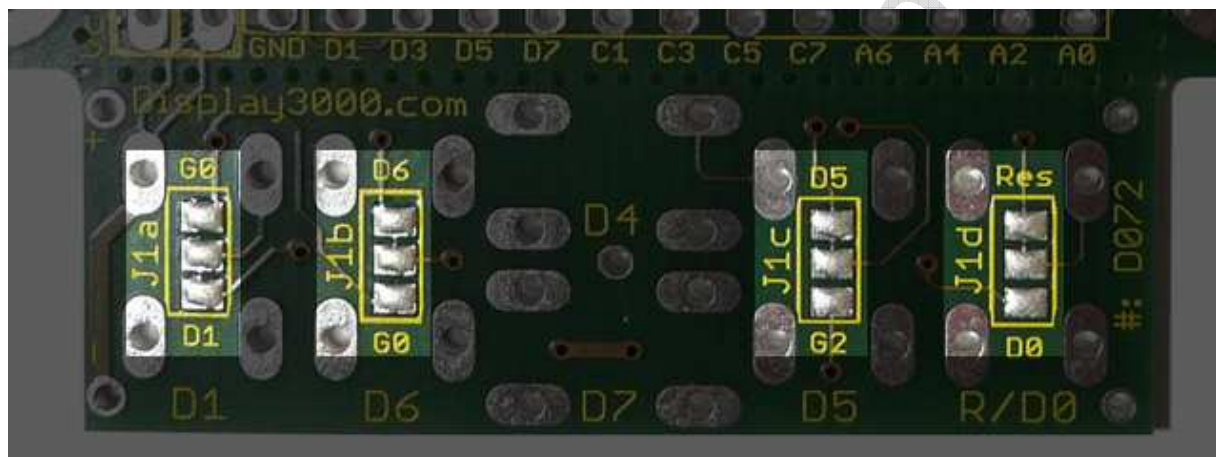
Um Ihnen das Erinnern leichter zu machen, haben wir zudem auf die Platine „ISP“ gedruckt sowie die Nase des Steckers stilisiert. Die Nase soll immer in Richtung der Beschriftung ISP zeigen.



Die Tasten des D072



Die Tasten vom Modul D072 sind bereits mit Port D verbunden (Belegung siehe Foto links). Die Belegung der jeweils äußeren beiden Taster kann durch eine Lötbrücke (J1a, J1b, J1c, J1d) geändert werden.



Die (von der Displayseite aus gesehene) rechte Taste ist mit Port D.I verbunden. Beim AT-Mega erlauben die meisten Pins eine Mehrfachnutzung – so auch der Port D.I, 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.I auf G.0 geändert werden.

Zur Änderung einer Taste trennen Sie bitte auf der Bauteilseite der Platine die dort bestehende Leiterbahnbrücke zwischen dem mittleren und dem unteren Feld auf (gekennzeichnet mit D.I) und schließen dann die beiden Lötfelder mittleres Feld mit oberem Feld (G.0) mit einem Tropfen Lötzinn.

Ähnliches ist auch mit den anderen 3 äußeren Tastern möglich. Die 2. Taste von links kann ebenfalls alternativ von D6 auf G0 geschaltet werden. Die 2. Taste von rechts kann entweder zu Port D5 oder G2 verbunden werden. Die rechte Taste (Reset) kann statt mit Reset auch mit D0 verbunden werden.

Notwendige Umkonfiguration bei CAN Controller: Wenn Sie ein Modul mit **AT90CAN128** und **CAN-Transceiver** erworben haben, dann sind Port D5 und D6 mit dem CAN-Transceiver verbunden (dies sind die CAN-Rx und CAN-Tx Leitungen des Microcontrollers). Sie sollten dann diese beiden Tasten (sofern benötigt) auf G0 und G2 (s.o.) umkonfigurieren, ansonsten erhalten Sie hier Fehlfunktionen.

Abschalt-Option zur Stromreduzierung

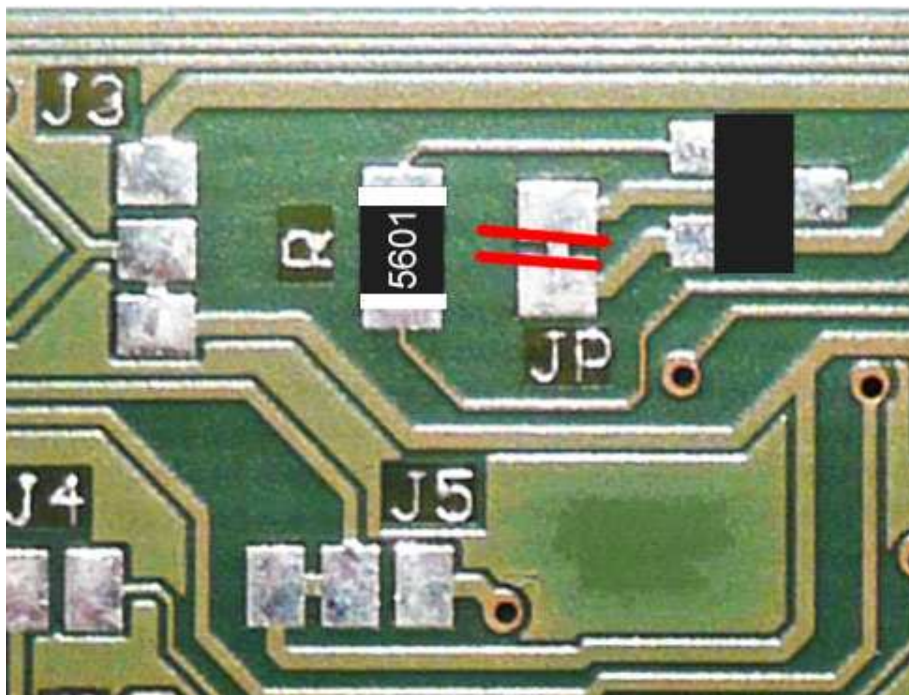
Es gibt Situationen, in denen der Mikrocontroller in den Standby-Modus geschaltet werden soll um Strom zu sparen. Solange jedoch noch die andere Peripherie auf dem Board weiterhin an der Versorgungsspannung verbleibt, verbraucht dieser auch weiterhin Strom, so dass sich nur ein eingeschränkter Stromspareffekt einstellen kann. Gerade bei einer batteriebetriebenen Lösung ist dies ärgerlich. Daher können Sie optional eine Erweiterung bestellen, durch die der Mikrocontroller sämtliche Peripherie kontrollieren und abschalten kann.

Durch je einen Jumper (Lötbrücke) legen Sie für jede einzelne Komponente fest, ob Sie vom Mikrocontroller abgeschaltet werden kann.

Im Einzelnen sind dies:

- 3 Volt Spannungsregler und somit Display inkl. Displayelektronik (jedoch nicht die Displaybeleuchtung) (Jumper J11)
- RS232 Transceiver, Booster für die Displaybeleuchtung (Jumper J3)
- CAN Transceiver (nur bei AT90CAN128 Mikrocontroller) (Jumper J2)

Um das Board mit dieser Option nachträglich auszustatten löten Sie an die beiden unten gezeigten Positionen einen Transistor (BC 857A) sowie einen 5,6K Widerstand ein. Durch Durchtrennen des Feldes JP können nun die obigen Elemente des Board von der Versorgung abgeklemmt werden. Der Transistor schaltet nun die o.g. Elemente.



Damit kann der Stromverbrauch des gesamten Boards auf ein Minimum gedrückt werden. Nach der Abschaltung liegt es lediglich noch an Konfiguration und Software, wie viel das Board noch an Strom benötigt.

Zu den unterschiedlichen Power-Modi erfahren Sie mehr im Datenblatt des Mikrocontrollers unter dem Kapitel „Power Management and Sleep Modes“. Dort erfahren Sie auch mehr über alle weiteren sinnvollen Maßnahmen zur Stromreduzierung (z.B. ADC abschalten, Pull-up-Widerstände abschalten etc.).

Wenn Sie das Display nach einer Abschaltung wieder ansprechen möchten, so ist es allerdings notwendig, dass Sie es vorher neu initialisieren (LCD_Init).

Nach Durchtrennen der Brücke JP schalten Sie 3Volt Spannungsregler, Display, RS232 und Powerbooster mittels des Ports G1 an und ab.

Port G.1 auf 0 = Einschalten, durch

Port G.1 = 1 = Abschalten.

Standardmäßig sind jedoch alle Elemente des Board an eine permanente Versorgung angeschlossen. Sie können nun für jedes einzelne ein Brückenfeld (Jumper 11, Jumper 3, Jumper 2) umkonfigurieren (siehe vorherige Seite, bzw. ab Seite 31) um es bei Bedarf abschalten zu können.

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 10.000 bis 20.000 Stunden haben (also bei Dauerbetrieb ca. 1-2 Jahre), macht dies u.U. ebenfalls Sinn für Projekte, die lange Zeit ohne weitere Beachtung vor sich hin arbeiten sollen (Anmerkung: Lebensdauer bedeutet hier: Helligkeit ist auf 50% des Anfangswertes abgefallen).

Aufgrund der für die Displaybeleuchtung notwendigen hohen Spannung (mindestens 11 Volt, auf unserem Board sogar ca. 14-16 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 ohne diese Schaltoption bestellt haben, so müssen Sie diese Option selber nachrüsten.

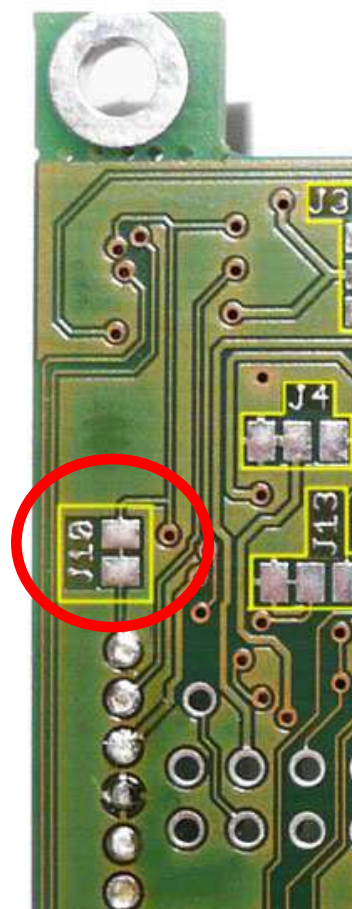
Im Lieferzustand besteht eine direkte Leitung zwischen dem PowerBooster und dem Display mit der integrierten Beleuchtung. Die Displaybeleuchtung ist von der übrigen Elektronik des Displays separiert und kann daher getrennt geschaltet werden.

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

Dazu trennen Sie die im folgenden Foto mit J10 gekennzeichnete Brücke mit einem scharfen Messer auf.

Schalten der Beleuchtung mit einem Schalter:

Löten Sie an die beiden Löt pads rechts und links der durchtrennten Leiterbahn im Jumper J10 ein Kabel und an dessen Ende einen Schalter und die Beleuchtung kann von Ihnen mittels dieses Schalter manuell ein- oder ausgeschaltet werden.



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 Sie diese nicht direkt anschließen – wir brauchen dazwischen noch zwei Transistoren als elektronische Schalter (zwei Transistoren da die LEDs über eine positive und negative Spannung versorgt werden: + 7Volt und -7 Volt). 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 J10 auf der Platinenrückseite so auf**, wie auf der vorherigen Seite beschrieben.

Dann löten Sie am oberen Rand der Platinenvorderseite die **beiden Transistoren** wie auf dem Foto gezeigt auf. Wichtig: **der NPN-Transistor (BC847) ist mit T1 gekennzeichnet und der PNP-Transistor (BC857) mit T2**. 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.

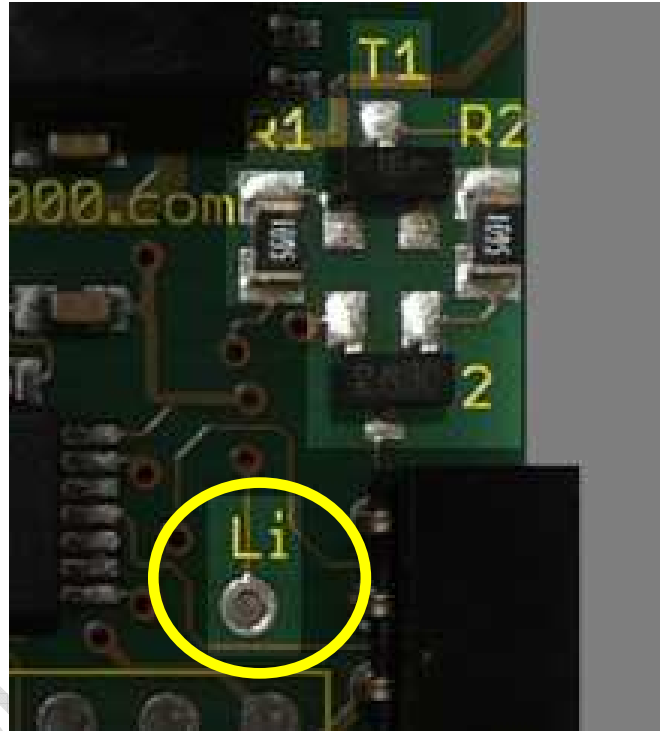
Ändern Sie daher in den Bascom-Beispielen die Zeile **Ddrb = &B01100110** ab in **Ddrb = &B11100110** (wenn dort nicht sowieso schon &B11111111 steht). Damit wird Port B.7 als Ausgang bereitgestellt (d.h. 1 bedeutet Output, 0 bedeutet Input). Mit dem Befehl **Portb.7 = 1** oder **Portb.7=0** schalten 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:

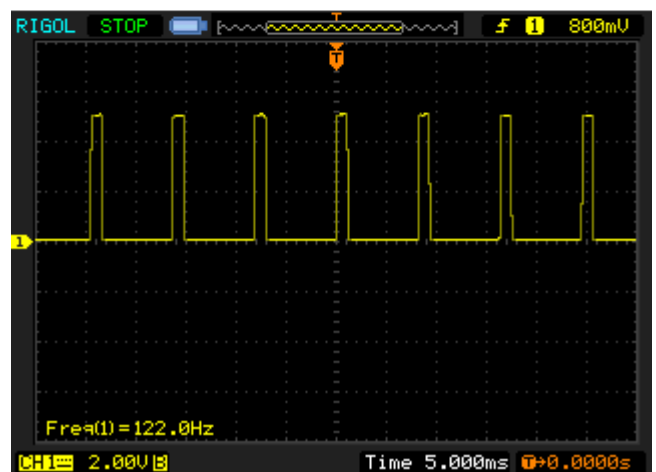
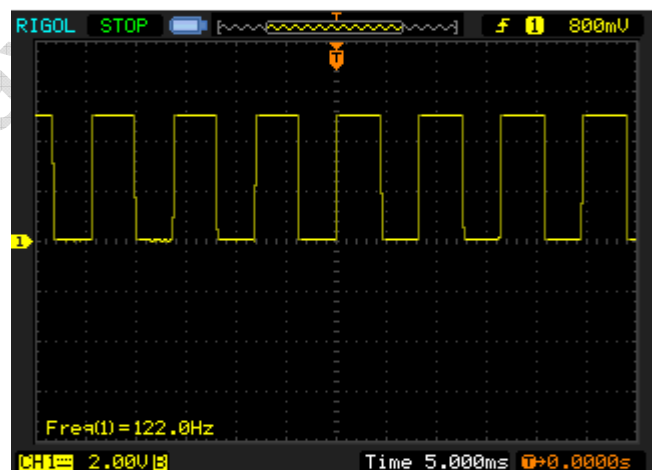
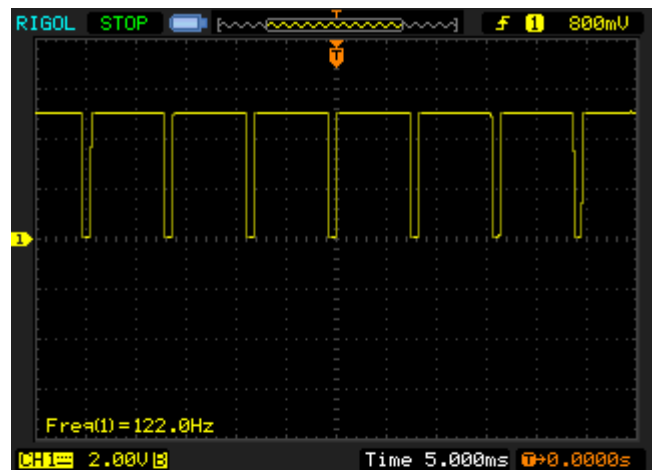
Wenn Sie z.B. eine Leuchtdiode binnen 1 Sekunde 5 Mal für 0,1 Sekunden ausschalten und dann wieder für 0,1 Sekunden einschalten, sehen Sie zuerst einmal ein Flackern. Zudem war aber die LED die Hälfte der Zeit abgeschaltet und hat daher in der Gesamtzeit auch nur die Hälfte an Lichtenergie abgegeben.

Angenommen, Sie beschleunigen diese Rate auf eine Ein- und Ausschalzeit 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 Beispiel 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. Man beachte: die Frequenz bleibt immer dieselbe. In der Mitte das 1:1 Verhältnis (d.h. die LED ist in ca. 50% der Zeit abgeschaltet und würde mittelmäßig hell leuchten); oben 1:10 (Die LED wäre meist eingeschaltet, also sehr hell); unten ca. 7:1 (LED meist abgeschaltet, also recht dunkel).



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.

Hinweis: Bestimmte Frequenzen führen zu einem „flimmern“ des Displays – sollten Sie dieses Flimmern wahrnehmen, variieren Sie die Frequenz entsprechend.

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 Einschaltet zu Ausgeschaltet verändert. Den Rest erledigt die Hardware nebenher – sie brauchen also in Ihrer Software keinerlei Ressourcen 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:

```
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

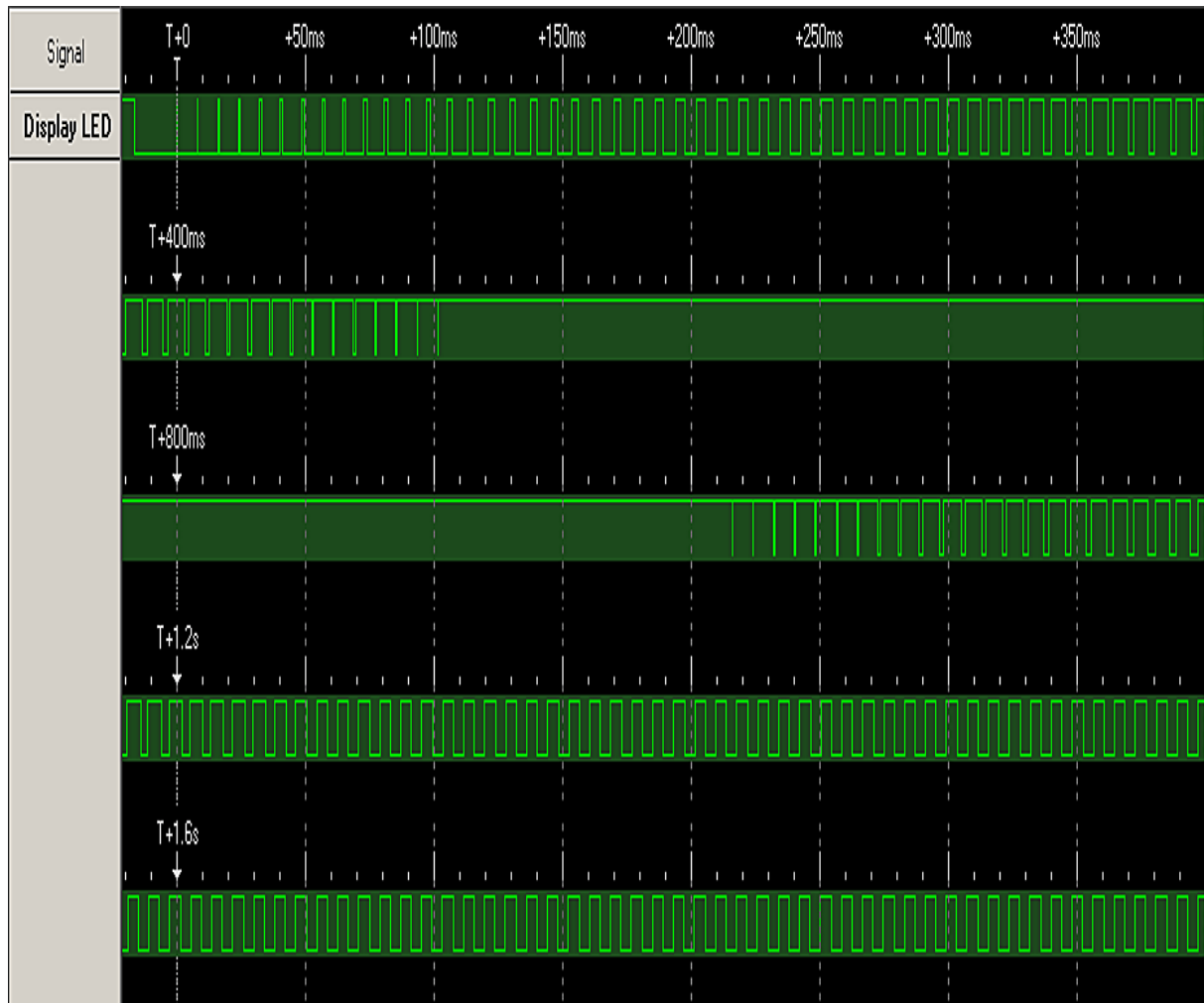
For I = 255 To 130 Step -5
    Pwm1c = I
    Waitms 10
Next I

End
```

Dann endet das Programm. **Preisfrage: Was passiert nach dem Programmende mit der Displaybeleuchtung?** Wird sie abgeschaltet, bleibt sie bei 50% oder wieder bei 100%? Antwort auf der nächsten Seite.

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 das Ausgangssignal des Mikrocontrollers noch mit einem Logikanalyzer aufgezeichnet – es ist die Aufzeichnung des Programms von der letzten Seite. Die gesamte Aufzeichnung ist 2 Sekunden lang, in jeder Zeile sind 400ms abgebildet. Sehr schön ist zu erkennen, wie sich das Tastverhältnis von Hell zu Dunkel 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 Ressourcen.

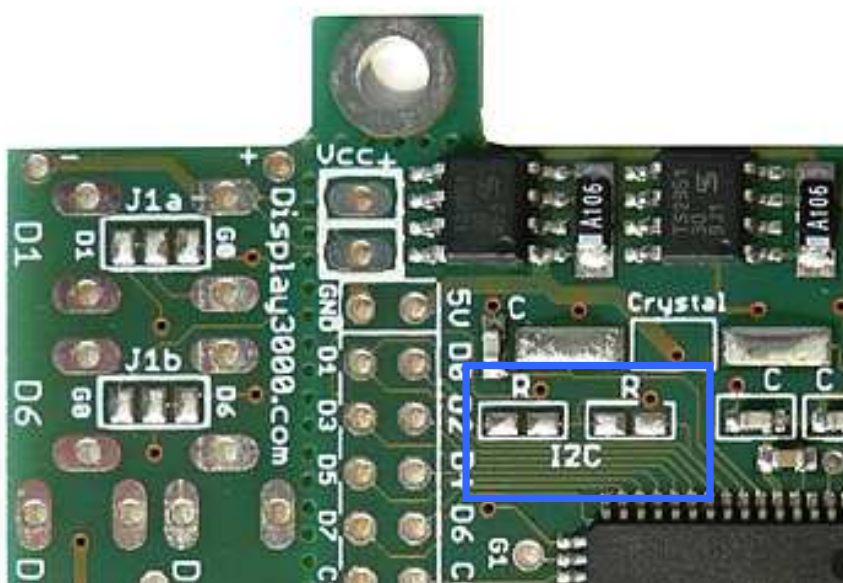
I²C / TWI – Zweidraht-Interface

Der ATmega bietet u.a. auch sein TWI, was nichts anderes bedeutet als Two-Wire-Interface. Ein bekannter 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 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 D072 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 D072 als Masterboard fungieren soll und es im System noch keine weiteren I²C Bus-Teilnehmer mit Pullup-Widerständen gibt, so können diese direkt im D072-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 603) eingelötet werden kann. Die Löt pads stehen dort zur Verfügung. Mehr ist (außer Software) für I²C nicht notwendig.



CAN-Bus Interface

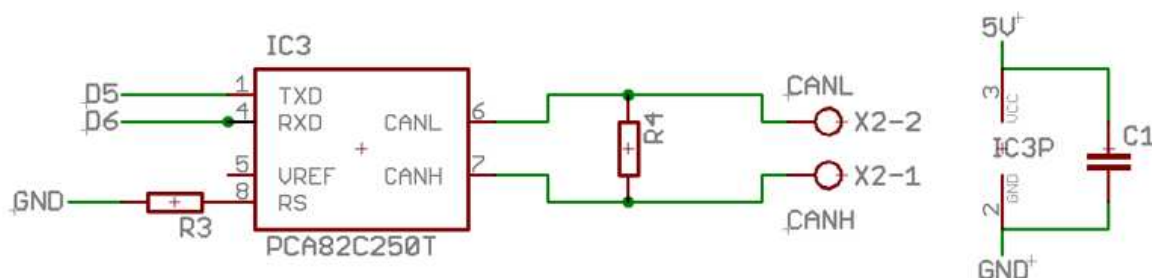
Der CAN-Bus (Controller Area Network) ist ein sog. Feldbus. Es handelt sich hier um ein asynchrones, serielles Bussystem.

Mit einem CAN-Bus lassen sich quasi beliebig viele Module miteinander unkompliziert vernetzen. Die Länge der möglichen Verbindungsleitung ist abhängig von der Geschwindigkeit der Datenübertragung. Die maximale theoretische Leitungslänge beträgt bei

1 MBit/s	40 m
500 KBit/s	100 m
125 KBit/s	500 m
50 KBit/s	1 km

Der CAN Bus kann von Ihnen nur genutzt werden, wenn Sie ein Controllermodul mit einem AT90CAN128 bestellt haben. Dieser Controller ist weitgehend kompatibel mit einem AT-Mega128, bietet jedoch zusätzlich noch einen integrierten CAN-Bus-Controller.

Zusätzlich zu diesem Controller ist jedoch in der Regel noch ein Transceiver notwendig. Dieser Transceiver ist notwendig, um das System an einen CAN Bus anschließen zu können. Der Transceiver ist der Schnittstellenbaustein, der es dem CAN-Bus-Controller (im Mikrocontroller) erlaubt, auf den CAN-Bus zuzugreifen. Wir nutzen den Quasi-Standard für High-Speed Systeme nach ISO 11898-2, den Philips PCA82C250.



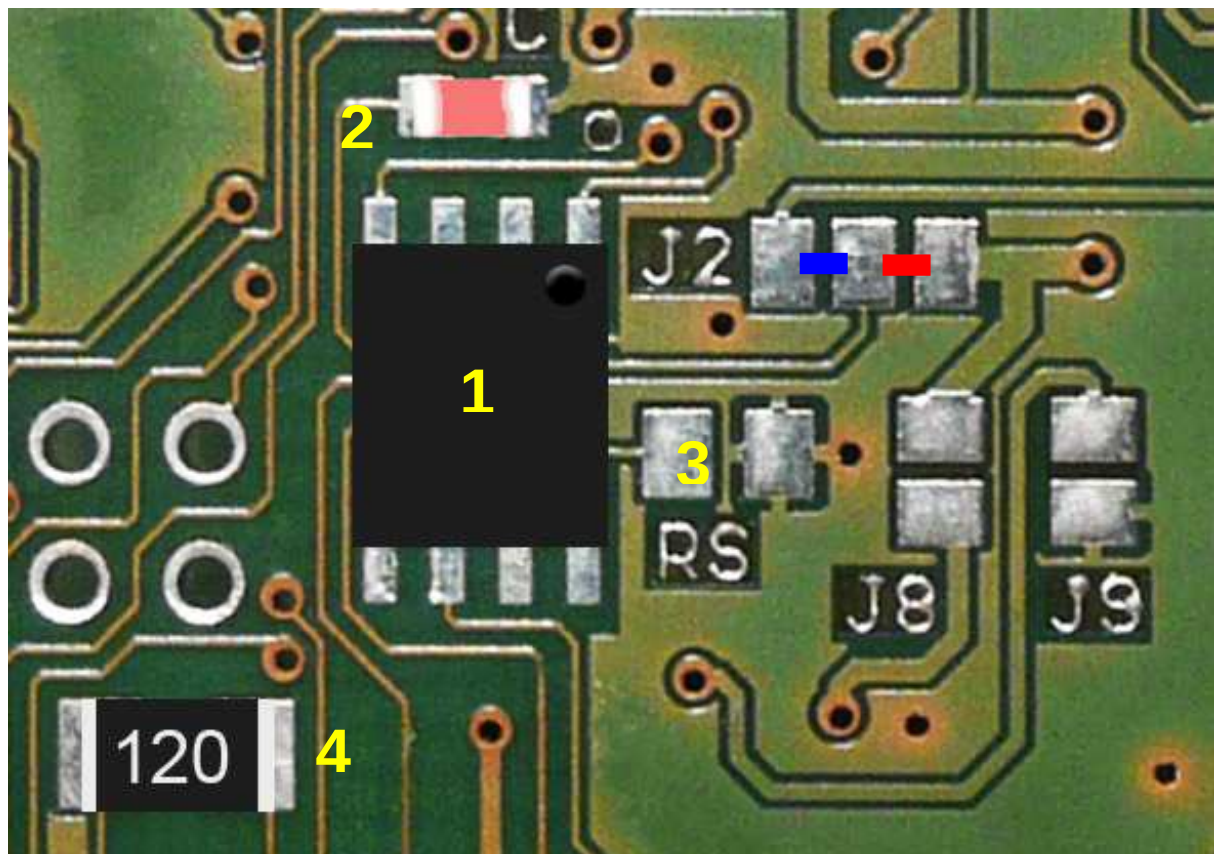
Wenn Sie die Platine mit CAN Controller AT90CAN128 bestellt haben, so löten wir zusätzlich den hierfür benötigten High Speed CAN Transceiver 82C250 mit ein.

Info zu den Widerständen: R4 ist der (unbedingt notwendige) Bus-Abschlusswiderstand (d.h. auf beiden Seiten des Bus notwendig) und wird laut Datenblatt mit 124 Ohm dimensioniert – aber 120 Ohm tun es auch. (siehe auch Foto auf der nächsten Seite).

R3 (der RS-Widerstand) muss von Ihnen je nach gewünschter Bus-Geschwindigkeit dimensioniert werden. Konsultieren Sie hierzu das Datenblatt bzw. die *Application Note* des PCA82C250 Transceivers auf unserer CD. **Dieser Widerstand ist von uns noch nicht eingelötet ! Sie müssen je nach Geschwindigkeit hier entweder eine Lötbrücke setzen oder den entsprechenden Widerstand einlöten.**

Anhaltswerte für den Widerstand RS:

Highspeed:	0 Ohm – also brücken (z.B. ein Tropfen Lötzinn)
1MBit/s	1 KOhm
100kBit/s:	47 KOhm



Legende:

- (1) = CAN Transceiver
- (2) = Kondensator 100nF
- (3) = RS Widerstand (muss von Ihnen gewählt und eingelötet werden - siehe vorherige Seite)
- (4) = 120 Ohm Bus-Abschlusswiderstand

Der CAN Transceiver ist mit den Ports D5 und D6 des Mikrocontrollers verbunden. Diese sind beim AT90CAN128 als die CAN Ports definiert. Diese beiden Ports werden standardmäßig jedoch auch durch zwei der Taster genutzt, die Sie jedoch umkonfigurieren können und bei Nutzung des CAN Bus auch sollten. Mehr hierzu erfahren Sie auf Seite 14).

Mit Lötbrücke J2 legen Sie fest, ob der CAN Transceiver durch den Mikrocontroller von der Versorgungsspannung abgetrennt werden kann oder nicht (siehe Seite 15).

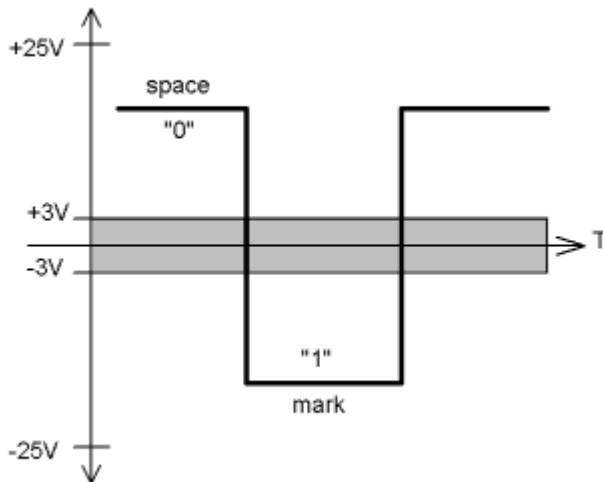
Standardmäßig sollte das mittlere und das rechte Feld miteinander verbunden sein (rot). Sollte dies nicht der Fall sein, holen Sie dies bitte noch mit einem Lötkolben und einem Tropfen Lötzinn nach – ansonsten erhält der Transceiver überhaupt keine Versorgungsspannung.

Für die Abschaltoption müssen dann das mittlere und das rechte Feld getrennt und dafür das mittlere mit dem linken Feld verbunden werden (blau).

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.

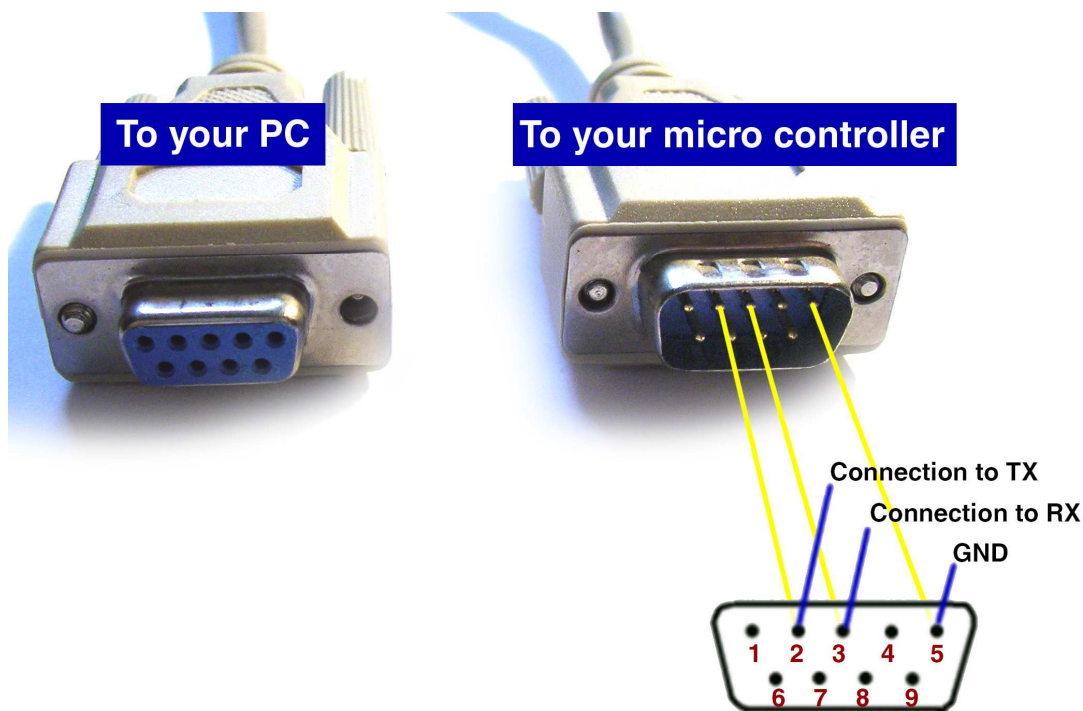
RS-232 ist ein asynchrones Protokoll, d.h. es wird keine separate Clock-Information mit übertragen. Sowohl die Sende-, wie auch die Empfangsstation müssen die daher die genaue Geschwindigkeit kennen (als Baud-Rate bezeichnet).

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 seriell 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. **Achtung: Um RS232 nutzen zu können, müssen Sie zuerst 2 Lötbrücken schließen. Mehr dazu auf Seite 32.**

Der RS-232-Interface-Chip wiederum ist an die beiden Anschlüsse RX und TX angeschlossen.

Achtung:

Die Löt pads 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.

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 enthält.

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 Schnittstellen-
ausgang 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 aber 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.

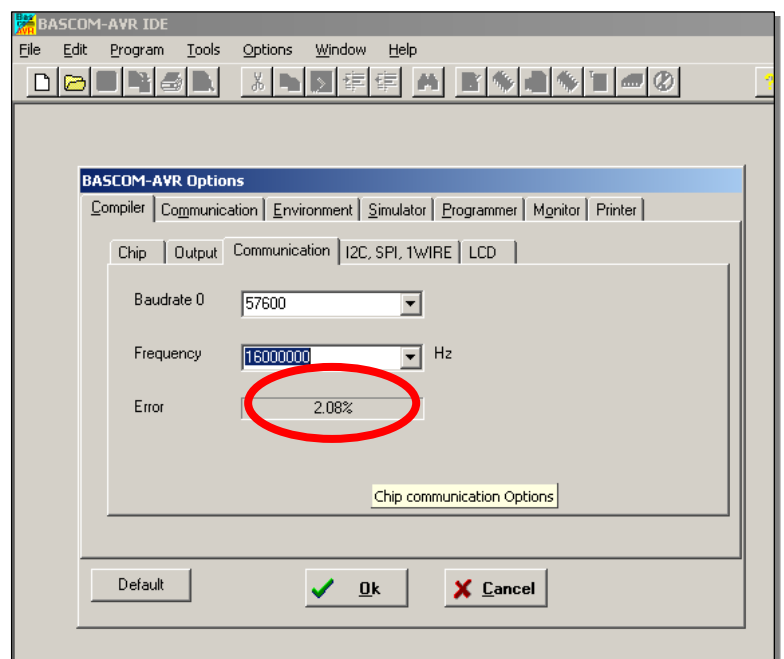
c) Sie müssen dem Compiler mitteilen, welche Taktfrequenz am Controller anliegt, sonst wird die Ermittlung des notwendigen Teilers für die Berechnung der Baudrate nicht korrekt durchgeführt. In Bascom führen Sie dies mit dem Befehl **\$crystal = 8000000** am Anfang des Programms durch at the beginning (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 (wir senden Ihnen auf Wunsch gerne die Excel-Tabelle zur Berechnung zu).

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** (siehe Foto links)



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 den eingebauten RTC (Real Time Counter) für eine Uhr nutzen, sollten Sie sowieso an die vorbereitete Stelle den separaten Uhrenquarz einsetzen – dann ist die Uhr unabhängig von der Taktfrequenz.

Stromverbrauch:

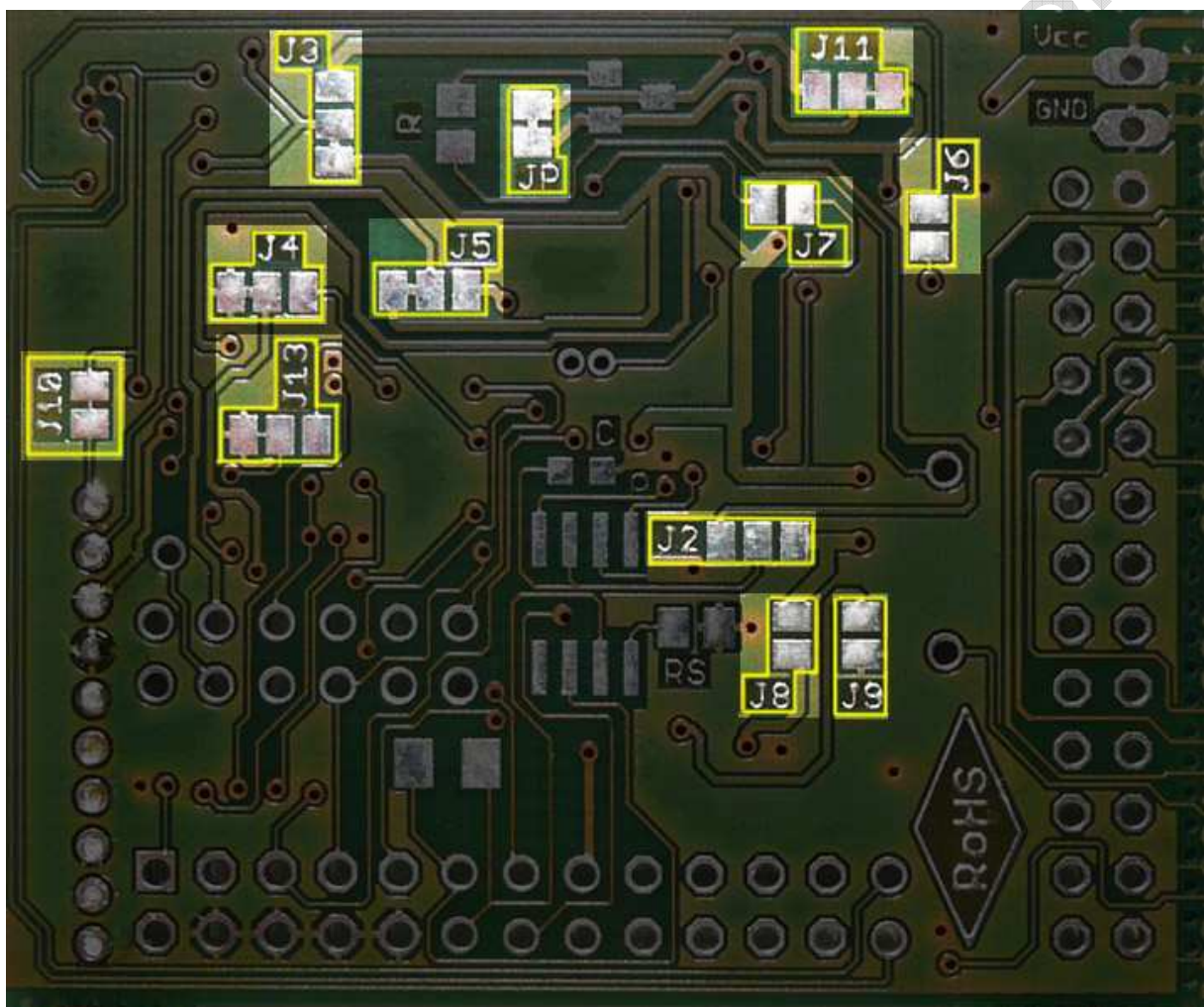
Ein weiterer Aspekt soll noch erwähnt werden, denn er ist für batteriebetriebene Geräte u.U. relevant: Je höher die Taktfrequenz des Mikrocontrollers, desto höher ist auch sein Stromverbrauch (siehe auch Seite 43).

Lötbrücken

Diverse Lötbrücken auf der Platine vereinfachen die Umkonfiguration für besondere Einsatzzwecke. Jede dieser Brücke ist mit einer entsprechenden Nummer gekennzeichnet und wird nachfolgend beschrieben.

Die Lötbrücken auf der Vorderseite J1a, J1b, J1c, J1d im Tasterbereich wurden bereits im Kapitel Taster beschrieben. Sämtliche weitere Brücken befinden sich auf der Rückseite. Um diese zu modifizieren ist es notwendig, das Display vorher vom Stecker abzuziehen.

Dort erkennen Sie dann die Lötbrücken J2 bis J13 sowie JP.



Lötbrücke CAN (J2):

Wenn Sie einen AT90CAN128 Mikrocontroller bestellt haben, dann legen Sie mir diesem Jumper fest, ob der CAN Transceiver durch die Abschaltoption (siehe Seite 15) mit von der Versorgungsspannung abgeklemmt werden soll.

Lötbrücke B6/GND (J3):

Standardmäßig wird der RS232 Transceiver und DCDC Booster für die Displaybeleuchtung immer mit 5V versorgt. Durch Umkonfiguration des Jumperfelds ist hier festzulegen, ob die Abschaltoption (siehe Seite 15) hier wirksam werden soll. Dazu gilt es, die Verbindung vom mittleren Feld zum unteren Feld aufzutrennen und eine neue Verbindung (LötKolben und ein Tropfen Lötzinn) zwischen dem mittleren und dem oberen Feld herzustellen.

Lötbrücke B6/GND (J4):

Da B6 auch für die Ansteuerung (DC-Data/Command) des Displays benötigt wird, wird eine weitere Nutzung durch andere Peripherie eher unwahrscheinlich sein. Daher haben wir das Pad B6 am inneren Connector standardmäßig mit Masse (GND) belegt – dort liegt also im Lieferzustand Masse an. Wenn das Feld für B6 benötigt wird, kann durch Auftrennen der Verbindung zwischen dem linken und dem mittleren Feld sowie der Verbindung (z.B. durch einen Klecks Lötzinn) des mittleren und des rechten Feldes, dem bisherigen Lötpad B6 nun auch der Port B6 zugewiesen werden.

Lötbrücke B4/Vcc 5Volt (J5):

Da B4 auch für die Ansteuerung (Reset) des Displays benötigt wird, wird eine weitere Nutzung durch andere Peripherie eher unwahrscheinlich sein. Daher haben wir das Pad B4 am inneren Connector standardmäßig mit 5 Volt (Vcc) belegt – dort liegt also im Lieferzustand 5 Volt an. Wenn das Feld für B4 benötigt wird, kann durch Auftrennen der Verbindung zwischen dem linken und dem mittleren Feld sowie der Verbindung (z.B. durch einen Klecks Lötzinn) des mittleren und des rechten Feldes, dem bisherigen Lötpad B4 nun auch der Port B4 zugewiesen werden.

Lötbrücke D3 (RS232) (J6):

Lötbrücke D2 (RS232) (J7):

Da der RS232-Transceiver an die Ports D2 und D3 angeschlossen ist (RxI und TxI des Mikrocontrollers), würde der Transceiver diese Ports beeinflussen, selbst wenn kein RS232-Empfang oder Sendebetrieb stattfindet. 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.

Daher wurde der Transceiver durch diese beiden Brücken abgekoppelt. **Das bedeutet, wenn Sie RS232 nutzen möchten, so müssen Sie J6 und J7 zuerst schließen.**

D.2 und D.3 stehen trotzdem immer an der Pfostenleiste zur Verfügung, lediglich die Verbindung zum RS232 Chip wird durch die Brücken geschlossen.

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 Sie im Lieferumfang RS232 nicht nutzen können und nach dem Schließen der Brücken D.2 und D.3 für Sie nicht als „normale“ Ports nutzbar sind.

Lötbrücke ARef bridge (J8):

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) – dafür haben wir Vorbereitungen getroffen:

Der Eingang ARef (Auf der Platine mit AR gekennzeichnet, zwischen F0 und E3) kann über die Lötbrücke (J8) mit Vcc (5V) verbunden werden. Diese Lötbrücke ist standardmäßig NICHT geschlossen. Sollten Sie die ARef für das Anlegen einer eigenen Referenzspannung benötigen, so muss die Brücke offen sein. Solange Sie diese Brücke geschlossen ist, 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 bitte Ihre eigene Referenzspannung an.

Hinweis für Experten: Bitte nutzen Sie nicht die interne Referenzspannung des Controllers, solange Sie ARef mit Vcc verbunden haben.

Aref hat einen eigenen 100nF Kondensator, der bereits von uns eingelötet wurde.

Lötbrücke AGND bridge (J9):

Ähnliches wie die Vcc Versorgung für Aref gilt für GND von Aref. Auf Wunsch können Sie, durch Schließen der Brücke J9 hier AGND (Das Lötpad direkt über Aref, also zwischen F1 und E4 auf die Standard-Masse legen.

Lötbrücke Licht (J10):

Solange diese Brücke geschlossen ist, arbeitet die Beleuchtung immer. Mit Öffnen dieser Brücke und Einlöten zweier Transistoren und zweier Widerstände können Sie die die Beleuchtungssteuerung (Schalten/Dimmen) mit dem Mikrocontroller vornehmen. Mehr hierzu auf Seite 17.

Lötbrücke 3V Spannungsregler mit Display und Displayelektronik (J11):

Solange das mittlere und das rechte Feld diese Brücke geschlossen ist, ist der 3V Spannungsregler ständig mit 5V verbunden. Durch Öffnen dieser Verbindung und Schließen der linken mit dem mittleren Feld kann nach Aufrüsten mit der Abschaltoption (siehe Seite 15) der Regler und das Display vom Mikrocontroller zu Stromsparszwecken abgeschaltet werden.

Lötbrücke (J12): Nicht belegt

Lötbrücke Tristate vom Display-Pegelwandler aktivieren (J13):

Der Pegelwandler für das Display ist permanent auf die SPI Schnittstelle aufgeschaltet. Es mag Situationen geben, wo andere Busteilnehmer das Display u.U. irritieren können. Dann lassen sich mittels Port E6 die Tristateausgänge des Pegelwandlers aktivieren und somit das Display komplett vom SPI Bus wegschalten. Hierzu muss die Brücke umkonfiguriert werden (Auftrennen Links-Mitte und Schließen von Mitte-Rechts). Mit E6 auf Low schaltet sich das Display auf den SPI Bus auf, mit E6 auf High wird das Display vom SPI Bus abgeklemmt.

Lötbrücke Abschaltoption (JP):

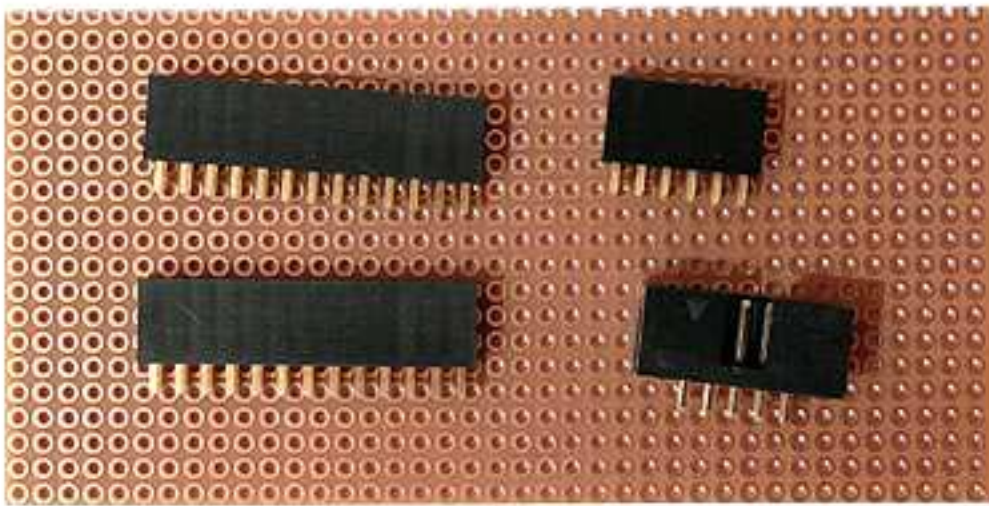
Solange diese Brücke geschlossen ist, werden sämtliche Elemente des Board von der vorhandenen Spannung versorgt. Durch Aufrüsten der Abschaltoption (siehe Seite 15) können dann durch Auftrennen der Brücke JP Bauteile vom Mikrocontroller zu Stromsparszwecken abgeschaltet werden.

Die zusätzlich verfügbare Hardware

Zubehör: Mini-Lochrasterplatine (P005):

Diese einseitige Platine mit den Maßen 100x50 mm passt hinter das Displaymodul. Es hat etwa die Breite des Moduls, ist aber länger und kann mittels eines Cutter-Messers schnell auf die benötigten Maße gekürzt werden.

Mit der Platine werden 3 Steckerleisten sowie ein ISP-Stecker geliefert. Das D072 Modul wird somit einfach aufgesteckt und kann durch eigene Beschaltungen erweitert werden. Dadurch sind Sie in der Lage ein Gesamtmodul mit weiteren Bauteilen (Relais, Transistoren etc.) zu erstellen.

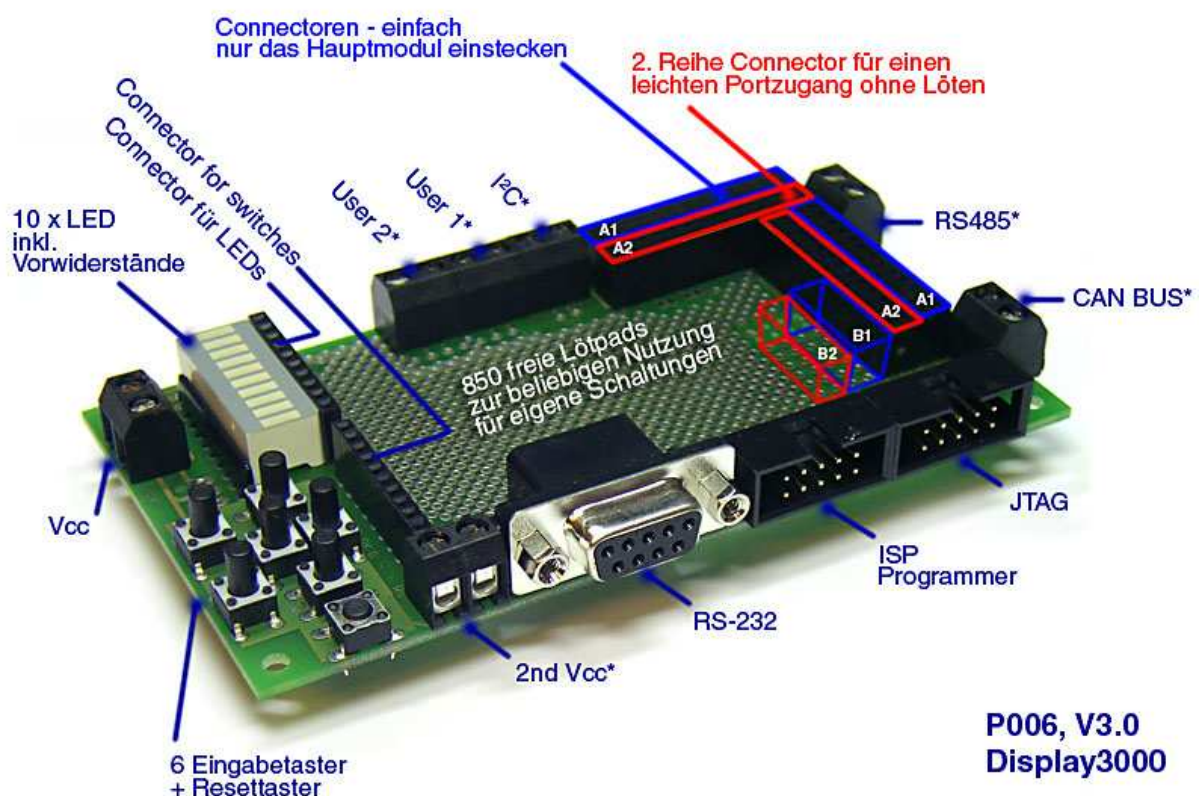


Zubehör: große Entwicklungsplatine (P006)

Ideal für:

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

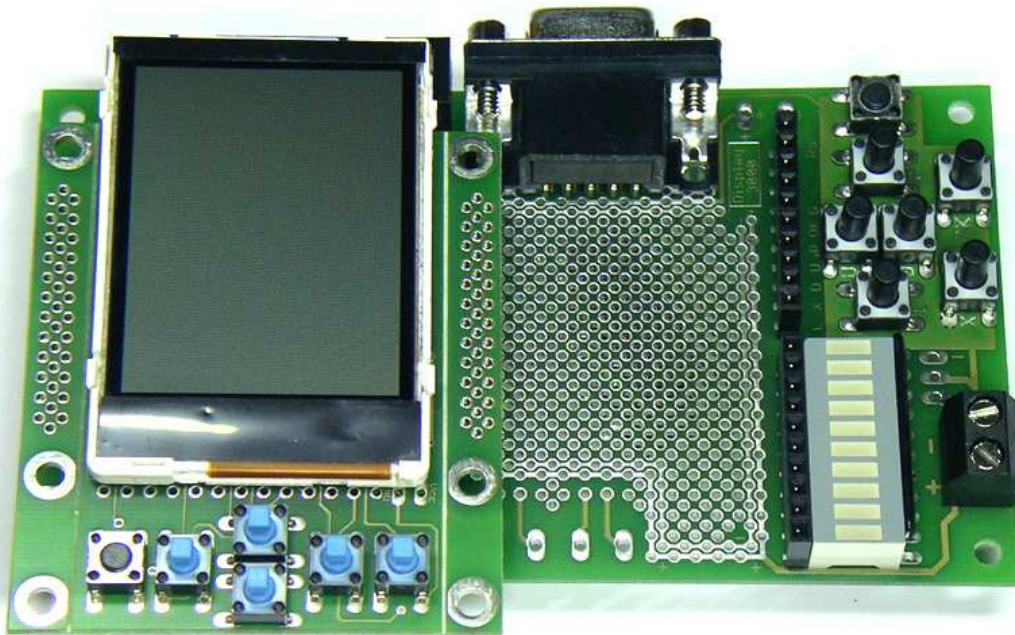
Diese Platine beinhaltet 6 Taster sowie einen Reset-Taster, 10 Leuchtdioden, einen Standard-RS232 Connector, einen ISP Stecker, einen JTAG-Stecker, sowie 4 Steckerleisten, einen Stekkontakt für die Spannungsversorgung sowie einen vorbereiteten Platz für einen größeren Spannungsregler. Über 850 frei verfügbare Lötunkte 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 D072 (hier noch ein Bild einer Vorgängerversion des D072) 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 Original Atmel ISP MK2
		

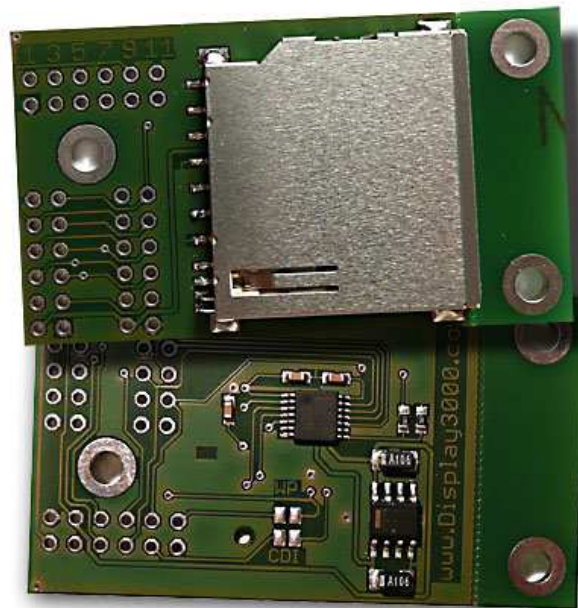
Am flexibelsten ist im Grunde der parallele Programmieradapter – leider bieten immer weniger Rechner standardmäßig eine parallele Schnittstelle an. Ein USB-Parallelumsetzer funktioniert übrigens nicht mit einem parallelen ISP-Programmer.

Wir empfehlen den original Atmel ISP MK2 Programmieradapter, den wir günstig anbieten. Er erlaubt auch die Programmierung von weiteren Atmel Controllertypen (z.B. AVR32 oder XMeta) und ist zudem **wesentlich schneller** als die seriellen oder parallelen Adapter. Es macht schon einen Unterschied, ob Sie bei der Programmentwicklung und dem durchs Testen häufigen Übertragung 20 x 10 Sekunden oder 20 x 90 Sekunden warten müssen.

Zubehör: SD-Kartenadapterkarte P001

Für den Anschluss einer SD Karte als Quasi-Speichererweiterung für z.B. Bilder oder zum Ablegen von Daten (z.B. Datenlogger etc.) ist diese Karte entwickelt worden. Sie beinhaltet optional einen 3V Spannungsregler sowie einen bidirektionalen Pegelwandler 5V->3V und 3V->5V mit Tristate-Ausgängen (wichtig!).

Wir liefern Softwarebeispiele in für Bascom Basic mit, für C gibt es div. vorhandene Bibliotheken im Internet. Wir erläutern detailliert in unserem Handbuch dieses Moduls die ansonsten bei der Nutzung von SD Karten problematische Zusammenarbeit mit anderen SPI-Bus-Teilnehmern (z.B. das TFT Display) und wie dieses Modul solche Probleme umgeht.

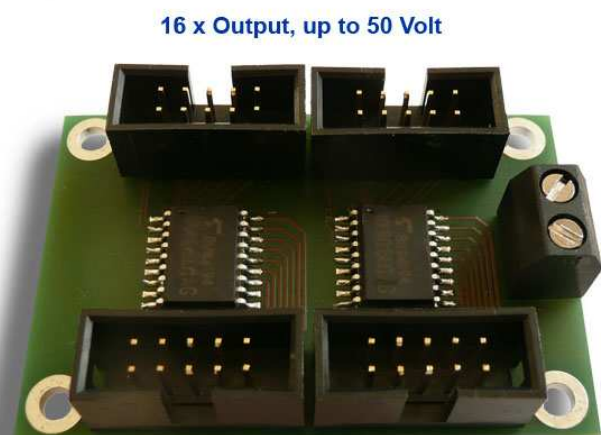


Zubehör: Treibermodul P017

Der Mikrocontroller kann über seine Ports lediglich Signale mit 5V Spannung abgeben – der Strom darf max. ca. 20 mA betragen. Für viele Steuerungen ist dies zu knapp bemessen.

Dieses Modul erlaubt den leichten Anschluss von stärkeren Verbrauchern wie z.B. Relais, Glühlampen, Pumpen etc.

Das Treibermodul P017 ist mit 8 oder 16 Ausgängen verfügbar



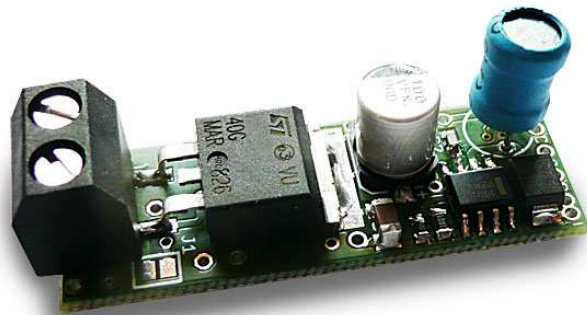
16 x Output, up to 50 Volt

16 x Input from Micro Controller (0.7 - 2 mA)

Zubehör: Platine mit Kfz-Versorgung P002

Spannungsversorgung beim Betrieb im Kfz

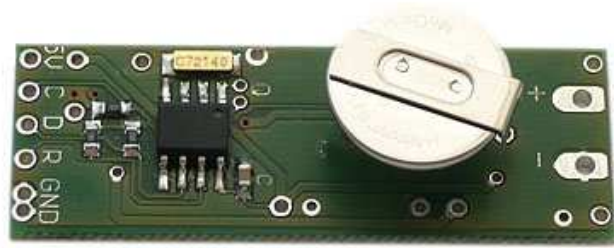
In einem Kraftfahrzeug liegt bei laufendem Motor eine Spannung von 13,8 Volt an. Für den Dauerbetrieb an unserem Modul ist dies u.U. zu viel (denn hier muss dann der interne Spannungsregler $13,8-5V \times 65mA = 0,6 \text{ Watt}$ in Hitze verwandeln und das ist für den Dauerbetrieb im Kfz zu viel) – mehr dazu erfahren Sie auf der übernächsten Seite. Hinzu kommt, dass in einem Fahrzeug kurzzeitige Spitzenspannungen von über 40 Volt auftreten können. Solche energiereichen Impulse können angeschlossene Elektronik zerstören. Daher muss Elektronik im Kfz immer so abgesichert werden, dass hier mehrere 100 Watt an kurzfristiger Energie vernichtet werden können.



Diese Platine P002 wird mit einem speziellen Kfz-Suppressor-Chip ausgeliefert, der hohe Spannungen erkennt und kurzzeitig bis zu 40 Volt und 50 Ampere vernichten kann. Zudem ist auf dieser Platine ein Schaltregler (ca. 55 KHz) implementiert, der eine 5 Volt Spannung bereitstellt, sich dabei aber nur unmerklich erwärmt und hocheffizient arbeitet statt wie ein normaler Längsregler die überschüssige Spannung in Wärme umzuwandeln.

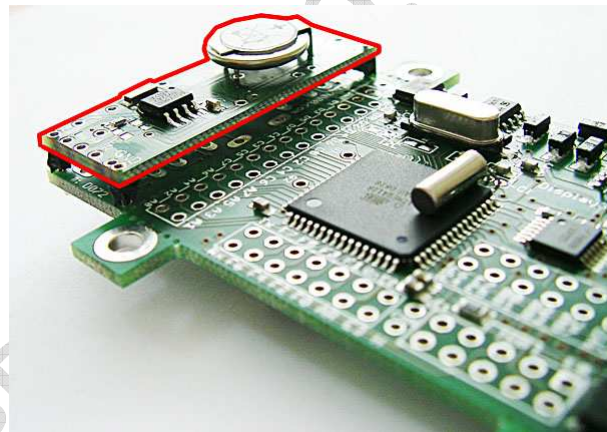
Zubehör: Echtzeituhr

Dies ist eine preisgünstige Möglichkeit, das Modul D072 um eine Echtzeituhr zu erweitern. Die Uhr läuft auch ohne Stromversorgung weiter, wenn sie durch eine Batterie gepuffert wird. Zusätzlich zu Datum und Uhrzeit bietet dieses Modul noch gepufferte 56 Byte RAM an, d.h. diese Daten bleiben bei Stromausfall oder Abschaltung des Systems vorhanden (sehr praktisch für Konfigurationsdaten oder sich oft ändernde Informationen/Parameter, die bei einem Neustart des Programms wichtig sind). Da dieser 56 Byte-Speicher ein RAM Speicher ist, kann er beliebig oft beschrieben werden (anders als ein Flash-Speicher oder ein Eeprom).



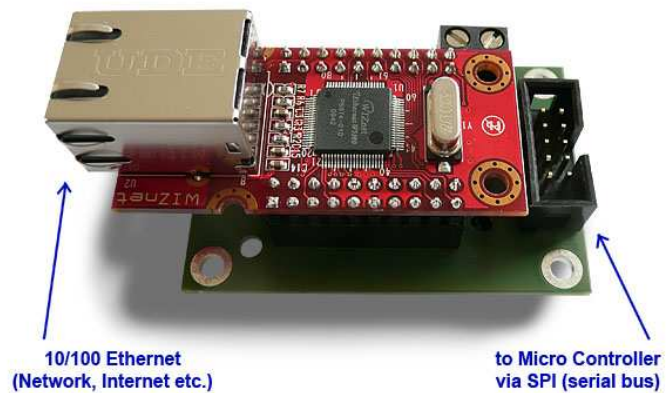
Die Lithiumbatterie hat eine Lebensdauer von mind. 10 Jahren und kann den Baustein bei fehlender Spannung für ca. 2 Jahre versorgen (solange Spannung anliegt, wird die Batterie nicht belastet).

Das Modul kann auf dem D072 befestigt werden (siehe nebenstehendes Foto) und braucht somit kaum zusätzlichen Platz.



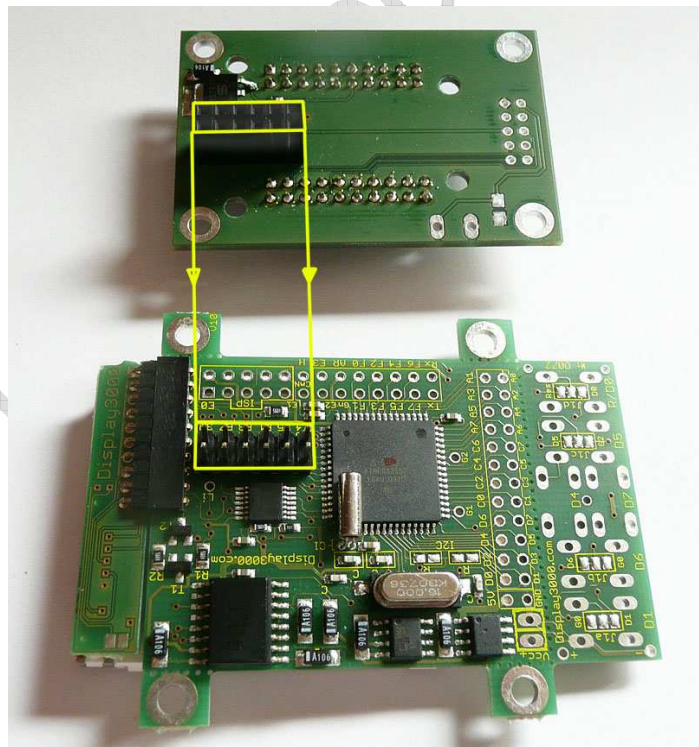
Zubehör: Wiznet Netzwerkmodul

Mit dem P020 Netzwerkmodul schließen Sie Ihren Mikrocontroller an ein Ethernet-Netzwerk an und realisieren hiermit z.B. einen Webserver auf Atmel-Basis oder rufen z.B. Parameter ab. Hiermit steuern Sie dann auf Wunsch Ihre Grafikmodule aus der Ferne. Das Modul kann über den Wannenstecker mit jedem Mikrocontrollersystem mit SPI Bus verbunden werden.



Das Netzwerkmodul P020 hält auf der Unterseite einen zusätzlichen Stecker zum direkten Aufstecken auf das D072 Modul bereit.

Somit vermeiden Sie die ansonsten notwendige externe Verkabelung: einfach Aufstecken und Ihr D072 Modul ist netzwerkfähig.



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.
- c) **Komplettabschaltung RS232 und Display:** Wenn Sie den Stromverbrauch auf ein absolutes Minimum bringen möchten, so können Sie das Modul inkl. einer Option zur Komplettabschaltung des Displays und des RS232 Moduls erwerben. Wenn Sie dann Ihren Mikrocontroller noch in den Sleep-Modus versetzen, ist ein Stromverbrauch von wenigen μA möglich, so dass sich dann auch batteriebetriebene Handgeräte entwickeln lassen.

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.) – ohne Spannungsreglerverluste, d.h. Eingangsspannung: ca. 5 Volt

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

Sleep Modus (Display, Beleuchtung & RS232: aus)	Full Speed (16 Mhz) (Display, Beleuchtung & RS232: aus)	Full Speed (16 Mhz) alles an
Abhängig von der Häufigkeit des Aufwachens: ab 100 μA	20 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:

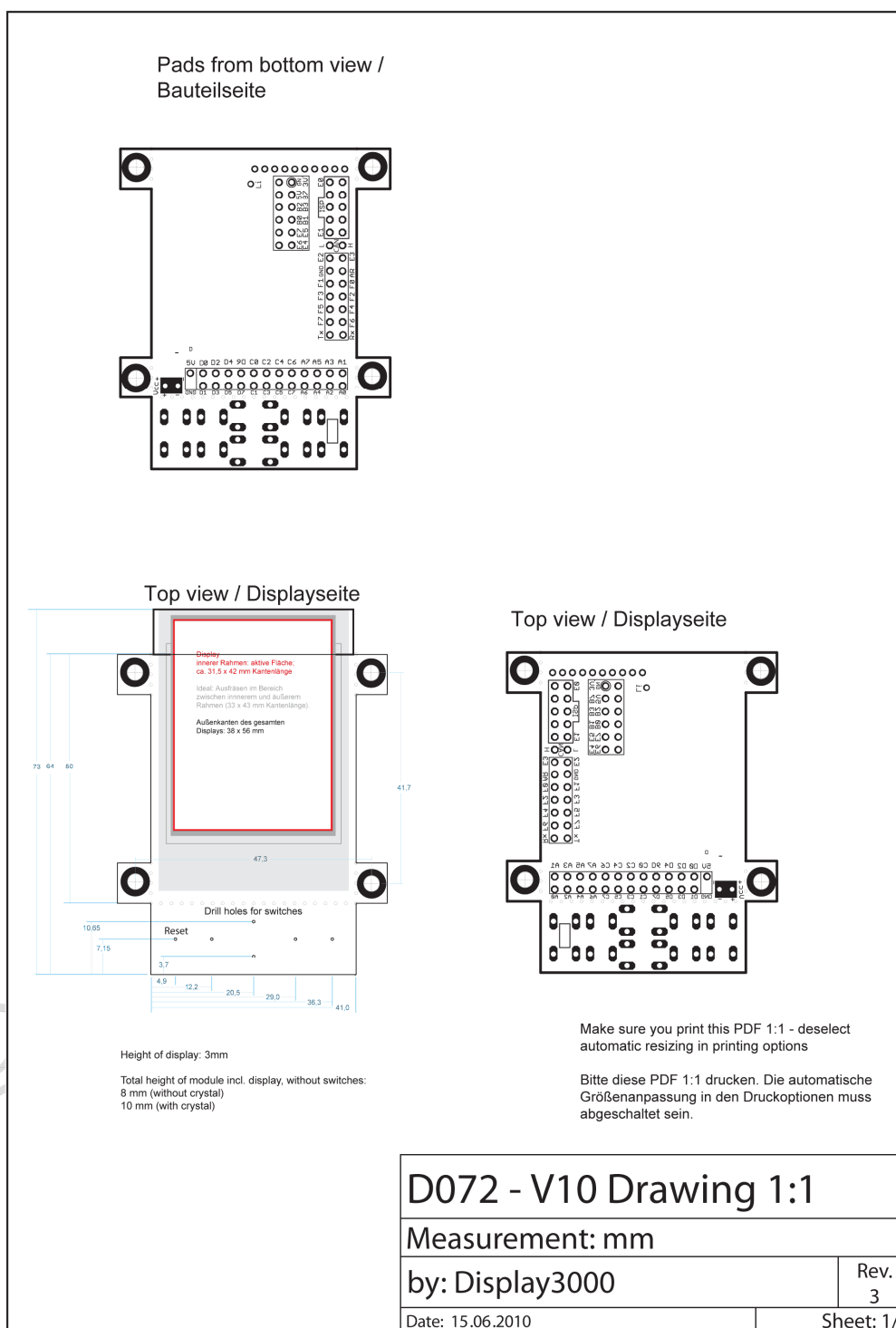
- a) Reduzierung des Stromverbrauchs durch Dimmen / Abschalten der Displaybeleuchtung (siehe auch Seite 43).
- b) Und/oder: Nutzung eines zusätzlichen externen Spannungsreglers mit Kühlkörper (z.B ein 7808, der dem Modul dann 8 Volt zur Verfügung stellt).
- c) Nutzung eines Schaltreglers – siehe hierzu nächster Absatz

Betrieb im Kfz

Bitte beachten Sie die oben unter Zubehör erwähnte Zusatzplatine P002.

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.



Technische Daten Display-Modul-Bausätze:

Artikelnummer D072:

Maße (Breite x Länge inkl. Display):

Mit Halterung, <u>und</u> Tastenfeld:	ca. 54 x 73 mm
Mit Halterung, <u>ohne</u> Tastenfeld:	ca. 54 x 59 mm
Ohne Halterung, <u>mit</u> Tastenfeld:	ca. 41 x 73 mm
Ohne Halterung, <u>ohne</u> Tastenfeld:	ca. 41 x 59 mm
Höhe:	ca. 8 mm inkl. Display

Versorgungsspannung:

4,5 bis 20* Volt Gleichspannung
(* max. Spannung siehe Seite 43)

Controller (je nach bestellter Ausstattung):

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

Technische Daten zum Display:

Display:

176 x 132 Pixel, 65.536 Farben
Aktive diagonale Fläche: 2,1" (53 mm)

Hintergrundbeleuchtung:

3 weiße LED, Color Ranking 5,6

Dot Pitch H & V:

0,237 mm

Vorgesehene Betrachtungsrichtung:

6 Uhr (Controller: 12 Uhr)

Lebensdauer im Betrieb:

5 Jahre (23 Stunden pro Tag x 365 Tage/Jahr)

Lebensdauer Hintergrundbeleuchtung:

mind. 5.000 Stunden (bei 15mA; wir betreiben es mit 12 mA = ca. 10.000 Stunden)

Temperaturbereich Display Lagerung:

-30°C bis +80°C

Temperaturbereich Display Betrieb:

-10°C bis +55°C

Temp. Bereich Display erweiterter Betrieb: -20°C bis -10°C und +55°C bis +65°C *

* = evtl. mit reversiblen (d.h. temporärem) Ausfall (z.B. dunkler Displaydarstellung)

Contrast Ratio Reflective Mode:

Min: 6; Typ: 11 (ohne Hintergrundbeleuchtung)

Contrast Ratio Transmissive Mode:

Min: 40; Typ: 70 (mit Hintergrundbeleuchtung)

Die Mikrocontroller werden von uns gegenüber der Standardauslieferung des Herstellers ATMEL wie folgt umprogrammiert:

ATmega128A: Die ATmega103-Kompatibilitäts-Fuse wird deaktiviert

Alle: Brown-Out wird auf 2,7 Volt gesetzt

Alle: Die Fuses für die Taktquelle sind je nach gelieferter Konfiguration entweder auf 8 Mhz intern oder 16 Mhz Quarz extern eingestellt.

Alle: „Eeprom Inhalt bei Reset nicht löschen“ ist eingeschaltet

ATmega2561 und AT90CAN128: Fuse Clockdiv8 wird abgeschaltet

Der Programmspeicher wurde zu Testzwecken bereits mit einem Demoprogramm vorprogrammiert.

Sollten Sie den JTAG-Zugriff nicht benötigen, so können Sie **Fusebit F:** auf **JTAG disabled** setzen. Port F4 bis F7 stehen nicht zur Verfügung, solange die JTAG-Option eingeschaltet ist. Auch eine temporäre JTAG-Deaktivierung per Software ist möglich.

Internes Eeprom und Brown-Out Fusebits:

Uns ist aufgefallen, dass der Mikrocontroller in bestimmten Grenzbereichen Zellen des internen Eeproms überschreibt. Dies tritt auf, wenn die Versorgungsspannung des Controllers langsam zusammenbricht (z.B. weil Sie ein Netzteil abschalten – dann fällt die Spannung langsam von Vcc auf 0 Volt ab). Dieses Phänomen kann auftreten, sobald die Spannung unter einen Wert von ca. 2,0 bis 2,5 Volt fällt. Eine Abhilfe bietet hier der integrierte Brown-Out Detector des Displays. Sobald die Versorgungsspannung des Controllers unter einen bestimmten Wert fällt, wird die Reset-Leitung aktiviert und der Controller stellt die weitere Arbeit ein. **Daher ist es sinnvoll, beim Mikrocontroller einen Brown-Out-Wert von 2,7 Volt einzustellen und Brown-Out zu aktivieren.**

Lieferant:

Speed IT up
Inhaber Peter Küsters
Wekeln 39
47877 Willich
Telefon: (0 21 54) 88 27 5-10
Telefax: (0 21 54) 88 27 5-22

Weitere Informationen und Updates: www.display3000.com

Autor dieses Manuals: Peter Küsters.

© **aller Informationen: 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.