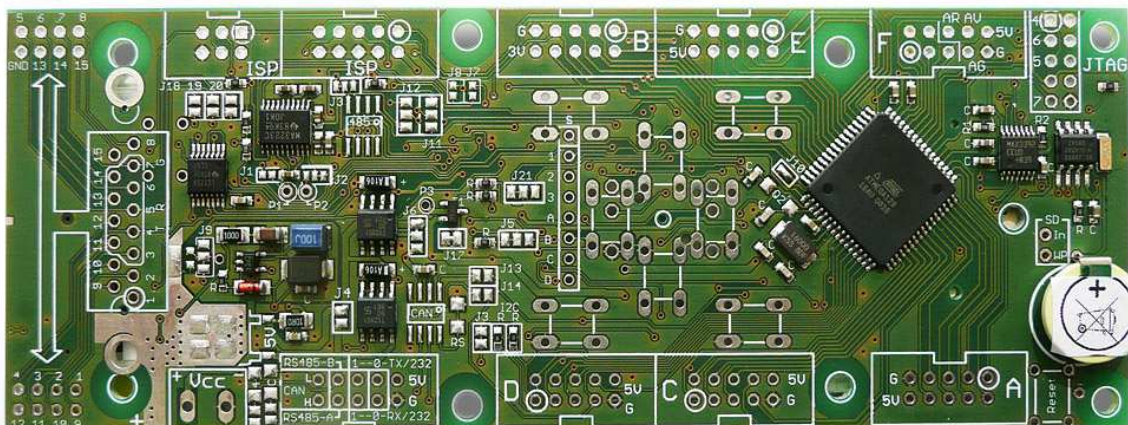
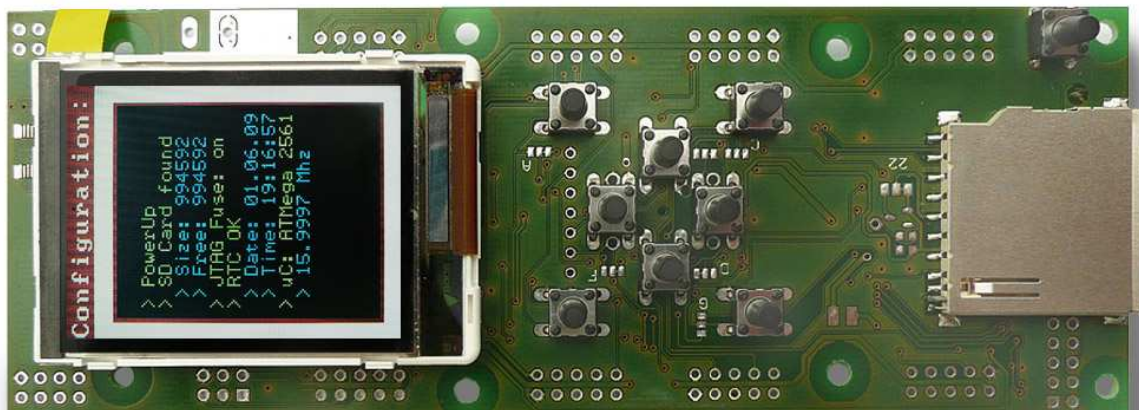


# Zusatzhandbuch für das Komplettmodul D071x mit ATMega Controller

V 1.27  
23. Oktober 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.**

## Inhaltsverzeichnis

Inhaltsverzeichnis .....	2
Lieferumfang.....	6
Erste Inbetriebnahme.....	7
Quick Guide: Häufig genutzte Umkonfigurierungen .....	9
Anschluss Programmieradapter.....	10
Spannungsversorgung.....	11
Batteriebetrieb .....	11
Benötigte Spannungen .....	12
Spannungsversorgung über die Port-Anschlüsse.....	12
„Großer“ Spannungsregler .....	13
Exkurs: Kühlflächen.....	14
Anschlusspads und Jumper:.....	15
Die Anschlussmöglichkeiten.....	16
Wannenstecker .....	16
15-poliger Standard D-Sub-Stecker .....	17
9-poliger D-Sub-Stecker.....	19
Verbinden der D-Sub-Stecker mit der seriellen Schnittstelle UART1 .....	20
Anschlüsse Schnittstellen (RS232, RS485, CAN).....	21
Die Eingabemöglichkeiten .....	22
Taster .....	22
Joystick.....	23
Folientastatur.....	24
Beschaltung des Tastenbereichs:.....	27
Reset-Taster .....	29
Taktfrequenz / Quarze.....	30
Systemtakt.....	30
Uhrenquarz für den internen Real Time Counter.....	31
Schnittstellen.....	32
SPI Bus:.....	32
I <sup>2</sup> C / TWI – Zweidraht-Interface.....	33
RS 232-Interface.....	34
RS-485 Interface.....	37
RS485-Abschlusswiderstand.....	38
RS485 Pullup / Pulldown.....	39
CAN-Bus Interface.....	40
CAN-Abschlusswiderstand.....	42
Analoge Eingänge (Analog-Digital-Wandler).....	43
Schalten der Beleuchtung durch den Mikrocontroller .....	45
Schalten der Displaybeleuchtung.....	45
Abschaltung von Verbrauchern.....	46
Echtzeituhr.....	48
SD Kartenmodul.....	50
Wichtiges zur Vermeidung von Datenverlust:.....	50
Warum ein bidirektionaler Pegelwandler mit Tristate Outputs? .....	51
Beschaltung SD Kartenmodul .....	52
Konflikt bei Konfiguration des SPI Bus .....	53
Formatierung der SD Karte .....	54

Abschaltung der Stromversorgung der SD Karte .....	55
Erkennung Schreibschutz / Karte eingesetzt .....	56
Die Software zum Ansprechen der SD Karte.....	57
Zugriff auf die Karte mit Bascom Basic.....	57
Schnellstart mit Bascom .....	58
Links:.....	62
Einsatz im Gehäuse GBS600 / GBS601 .....	63
Übersicht Jumper / Lötbrücken .....	67
Übersicht Portbelegung .....	68
Technische Daten Display-Modul-Bausätze: .....	69
Maßzeichnung.....	70
Oft gestellte Fragen (FAQ): .....	71
Ich kann die Ports F4 bis F7 nicht benutzen.....	71
Ich kann die Port D0 und D1 nicht benutzen .....	71
Ich kann die Ports D5 und D6 nicht benutzen .....	71
Ich kann nicht auf die Echtzeituhr zugreifen.....	71
Ich kann nicht auf die SD Karte zugreifen.....	71
Ich kann keine Analogmessungen durchführen.....	71
Ich habe ein RS232 Kabel an den D-Sub-9 Stecker angeschlossen, bekomme aber keine Verbindung.....	71
Ich kann CAN nicht nutzen.....	72
Das Board funktioniert, aber nach einer Weile blinkt die Beleuchtung und mein Programm stürzt ab oder das Board resettet sich selbstständig.....	72
Das Display spinnt gelegentlich .....	72
Ich kann plötzlich nicht mehr auf den Controller zugreifen .....	72
Einzelne Ports funktionieren nicht.....	72
Ich möchte den Realtime-Counter (z.B. für die Realisation einer internen Uhr) verwenden, aber dieser zählt nicht, bzw. der entsprechende Timer 0 bzw. Timer 2 (je nach Controller) arbeitet nicht oder nur manchmal.....	73
Anhang: Zusatzinfos RS232 und RS485.....	75
Anhang: Dimmen der Displaybeleuchtung mittels PWM.....	80
Anhang: SD-Karte – Warum aufwändige Beschaltung? .....	83
Anhang: Leitungsterminierung.....	86
Wannenstecker / Pins im Standardgehäuse GBS600 .....	89
Anhang: Abnehmen/Entfernen des Displays.....	90
Anhang Errata: .....	92
a) Spannungsabschaltung des bidirektionalen Pegelwandlers für SD Karten .....	<b>Fehler!</b>
<b>Textmarke nicht definiert.</b>	
b) Joystick-Beschaltung.....	92
c) Reset-Taster .....	92
Haftung, EMV-Konformität .....	93

Herzlichen Glückwunsch zum Erwerb des D074-Moduls inklusive 2.1“ TFT-Farbdisplay.

Dies ist ein multifunktionales Modul, welches in dieser Funktionsfülle, Flexibilität und Kompaktheit seinesgleichen sucht. Es ist praxisingerecht und sofort einsetzbar in Klein- und mittleren Serien, der Industrie sowie in Forschung und Entwicklung.

Zusammen mit dem BS600 Gehäuse sowie der Folientastatur erhalten Sie kostengünstig ein selbst als Prototyp professionell wirkendes Gerät (siehe Foto rechts mit einer CD als Größenvergleich). Rechnen Sie einmal durch, wie viel Sie eine solche Einzelanfertigung als Prototyp kosten würde.



Ab einer Stückzahl von 50 Platinen (darunter wäre es wirtschaftlich vermutlich unsinnig) ist übrigens auch eine komplett an Ihre Bedürfnisse hin abgewandelte Version der D071x Platine möglich. Sprechen Sie uns bei Interesse an. Wir entwickelnd und produzieren seit Jahren elektronische Schaltungen aller Art für Kunden aus allen Branchen.

### Optionen:

Je nach bestellter Ausstattung kann das Modul folgende Optionen beinhalten:

- Mikrocontroller mit 128 / 256 KByte Programmspeicher, 4/8 KByte RAM
- Transflectives (d.h. auch bei hellem Umgebungslicht ablesbar) Farb-TFT mit 176x132 Pixeln Auflösung
- 2 x RS232
- 1 x RS485 halbduplex
- CAN BUS
- I<sup>2</sup>C BUS
- SPI BUS
- Gepufferte Echtzeituhr
- Separate Uhrenquarz für TOSCI/TOSC2 (bei Echtzeituhr nicht notwendig)
- Externer Quarz mit 14,7456 Mhz oder 16 Mhz
- SD Karten-Controller
- Spannungseingang 3 bis 18 Volt statt 3 bis 12 Volt
- Steuerung über Tasten oder Joystick
- Elegantes Handgehäuse und Folientastatur

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

Diese Anleitung zeigt Ihnen die Anschlussbelegung des Boards und gibt ein paar Tipps zu diesem Board. Verschiedene Sektionen des Boards werden der besseren Übersicht halber schematisch dargestellt – die zum Verständnis nicht notwendigen Schaltungsteile werden hier nicht gezeigt. **Der komplette Schaltplan des Boards inkl. aller Zusatzoptionen findet sich auf der beigefügten CD** (nur für Kunden, nicht im Internet verfügbar).

Damit im Handbuch verwendeten Bezeichnungen links, rechts, oben und unten mit Ihrer Platinenansicht übereinstimmen, legen Sie bitte die Platine so vor sich hin, dass das Display unten liegt und sich die Bauteile oben befinden – die Beschriftung auf der Platine sollte lesbar sein.

**Zu den Fotos:** Die Fotos dieses Handbuchs sind von 2 unterschiedlichen Platinenrevisionen erstellt worden. Die beiden Versionen sind zu 99,9% identisch, aber es gibt ein paar Modifikationen, hauptsächlich am Bestückungsdruck. Die Modifikationen werden Ihnen u.U. gar nicht auffallen – aber es sollte doch zumindest erwähnt werden, um Sie nicht zu verwirren.

**VORSICHT:** Wenn eine Passage in diesem Manual rot gedruckt ist, sollten Sie sie besonders beachten, um spätere Probleme oder Beschädigungen zu vermeiden.

**Wichtiger Tipp vorab:** Überfliegen Sie nun als erstes einmal das komplette Handbuch, um einen Überblick über die verschiedene Optionen und das, was Sie erwartet, zu erhalten.

Wenn Sie sehr neugierig sind, schließen Sie dann vor Durcharbeiten des Handbuchs einmal eine Spannung an den Stecker Vcc an (siehe Kapitel „Erste Inbetriebnahme“) um Board und Display in Aktion zu sehen. **Sie sollten aber mit dem Einlöten oder dem Konfigurieren warten, bis Sie das gesamte Handbuch einmal komplett durchgelesen haben.**

**Tipp:** Sollte Ihr Board nicht korrekt arbeiten, so kann dies an einer unzureichend stabilisierten Spannungsversorgung liegen (i.d.R. billiges Steckernetzteil). Probieren Sie in einem solchen Fall als erstes eine Versorgung mit einer guten, stabilisierten Spannung aus (im Zweifelsfall erst einmal per Batterie) um ein Problem mit der Spannungsversorgung auszuschließen.

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 Mikrocontrollers sein können. Wir verweisen hier auf das umfangreiche Datenblatt des Atmel-Controllers (mehrere hundert Seiten in Englisch) sowie die diversen Foren im Internet.



## Lieferumfang

### Standard-Lieferumfang:

- 1 x bestückte Platine mit ATmega Prozessor etc.
- 1 x 2.1“ Farb TFT Display
- 9 x Taster (inkl. Reset)
- 2-reihige Pfostenstecker (RM 2,54)
- 1 x ISP Buchse (2x5, RM 2,54).
- Software, Dokumentation

### Optionale Ausstattung:

Standardmäßig wird das Board mit einem ATmega 128 ausgeliefert. Gegen Aufpreis (siehe Online-Shop) liefern wir auch den pin-kompatiblen Typ:

- ATmega 2561 (256 KByte Programmspeicher, 8 KByte RAM, 4 KByte Eeprom etc.) oder
- AT90CAN128 (ähnlich ATmega128 aber mit integrierter CAN-Schnittstelle).

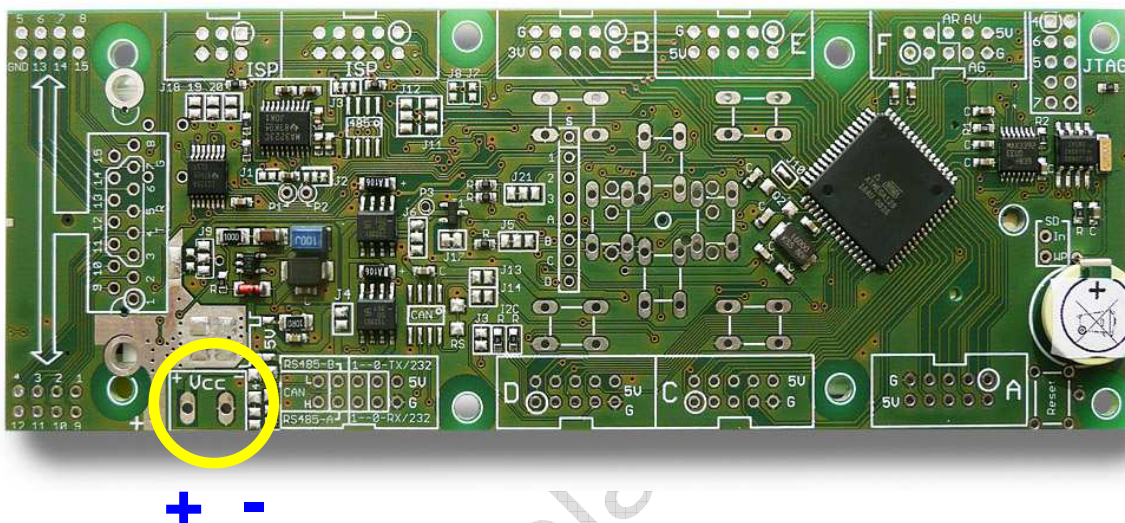
### Gegen Aufpreis ist weiterhin erhältlich:

- externer Quarz mit 14,7456 MHz oder 16,0 Mhz;
- Uhrenquarz 32 KHz für internen RTC
- Echtzeituhr mit Lithiumzelle als Pufferung (immer die korrekte Uhrzeit und Datum für mehrere Jahre auch bei Ausfall der Versorgungsspannung)
- SD Kartenelektronik und -halter
- RS232-Transceiver
- RS485-Transceiver nach Wunsch als 5V oder 3-5V Typen
- CAN-Bus-Transceiver
- 5V Spannungsregler für Eingangsspannungen >12 Volt
- Abschaltungsoption für Verbraucher
- Joystick statt Taster
- Folientastatur statt Taster
- Gehäuse
- Programmieradapter
- Wannenstecker-Set statt Pfostensteckern

## Erste Inbetriebnahme

Legen Sie eine Spannung von 5 bis 12 Volt an den Anschluss VCC (unten links auf der Platine). Der linke Anschluss ist Plus, der rechte Anschluss ist Masse (GND).

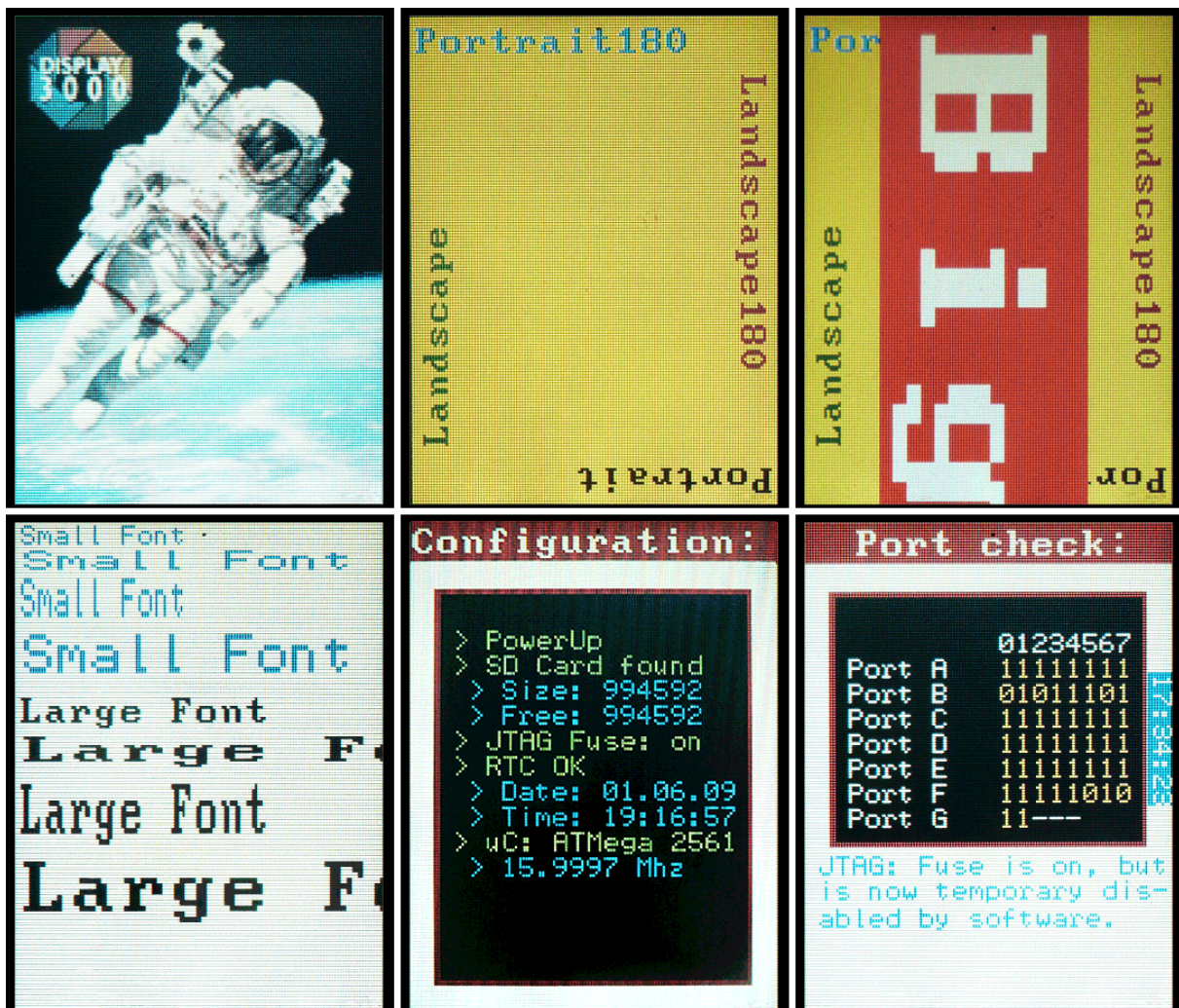
Anmerkung: Die unten gezeigte Platine zeigt bereits etliche optionale Bauteile, die Sie evtl. nicht bestellt haben und ist daher bzgl. der Bestückung nicht identisch zu der Ihnen vorliegenden Platine.



Der Controller wurde zu Testzwecken bereits vorprogrammiert und zeigt Ihnen auf dem Bildschirm ein Testprogramm (Grafik, Fonts, Komponenten – siehe Foto auf der nachfolgenden Seite) an.

Bildschirme sind immer schwierig zu fotografieren. In Wirklichkeit sieht die Bildschirmdarstellung wesentlich schärfer und brillanter aus als auf den Fotos.

**Das Modul wird von uns mit bereits montiertem Display geliefert. Wenn Sie das Display abnehmen müssen oder möchten, lesen Sie bitte unbedingt vorher das entsprechende Kapitel im Anhang (ab Seite 90).**



Das Testprogramm zeigt u.a. an, ob eine SD Karte erkannt wird (Speichergröße der Karte sowie der freie Speicher) und ob sich eine Echtzeituhr auf dem Board befindet. Die angezeigte Uhrzeit und das Datum der Echtzeituhr sind nicht korrekt – die Uhr muss von Ihnen noch einmalig gesetzt werden. Die Taktfrequenz des Boards wird dargestellt (Schätzwert!).

SD Karte: Wenn Sie eine SD Karte einlegen, wird diese erkannt, die Größe der Karte angezeigt und es wird zum Test eine Datei auf die Karte geschrieben: „README.TXT“. Sollte das Programm bei der Anzeige „SD Card found“ stehenbleiben, so wird Ihre Karte zwar erkannt, aber nicht ordnungsgemäß initialisiert. Entweder ist sie defekt oder muss neu formatiert werden (siehe hierzu die Tipps und Anmerkungen auf Seite 54).

Weiterhin wird der Status der JTAG Fuse gezeigt. Diese ist für den Ablauf des Programms nicht weiter relevant, da wir JTAG per Software deaktivieren – dies ist für den Zugriff auf das SD Kartenmodul notwendig. Wenn Sie keinen JTAG Debugger besitzen und somit JTAG nicht benötigen, können Sie diese Fuse auch komplett deaktivieren.

Beim abschließenden Port-Check des Programms können Sie z.B. testweise die Taster in ihre Kontakte einstecken (diese halten auch ohne Löten in der Platine) und erkennen, welche Portzustände sich auf bei Betätigung der Taster ändern: Das Drücken eines Tasters z.B. lässt den jeweils damit verbundenen Port von „1“ auf „0“ wechseln.



## 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 UART<sub>1</sub> (z.B. des Sub-D-Steckers am Rand):

- Jumper J18, J19, J20 schließen (siehe Seite 20)
- Jumper J7 und J8 schließen (siehe Seite 35)
- UART<sub>0</sub> kann nicht so einfach genutzt werden, da diese Ports auch vom ISP Programmieradapter genutzt werden (mehr dazu im entsprechenden RS232 Kapitel)

### 2) Komplettabschaltung nutzen:

- Jumper J17 Öffnen
- Jumper J6 umkonfigurieren (siehe Seite 43)
- Nutzung: PortE.3 muss auf Ausgang geschaltet sein.
- PortE.3=1 : Versorgung div. Verbraucher wird abgeschaltet
- PortE.3=0 : Versorgung div. Verbraucher wird eingeschaltet
- Nach dem Aufwecken müssen Display und SD Karte neu initialisiert werden, denn sie waren ja ohne Versorgungsspannung.

### 3) Displaybeleuchtung schalten:

- Jumper J9 muss umkonfiguriert werden (siehe Seite 45)
- Schalten mit Port B7; Dimmen mit PWM Kanal C (*Pwm1c*)

### 4) Folientastatur: Den roten Taster links unten zum Ein-/Abschalten per Interrupt nutzen:

- Tasterjumper D muss umkonfiguriert werden (Siehe Seite 28). Der Taster liegt dann an Port E7 – dies ist ein Interrupt-Port, welcher zum Aufwecken genutzt werden kann.

### 5) Analoge Eingänge nutzen

- AVcc und AGND müssen versorgt werden (siehe Seite 43)
- ARef festlegen (entweder interne Spannung oder Verbinden mit 5V)
- 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)

### 6) Vorhandener Uhrenquarz

- Tasterjumper TJ\_B sollte umkonfiguriert werden (siehe auch Seite 31)

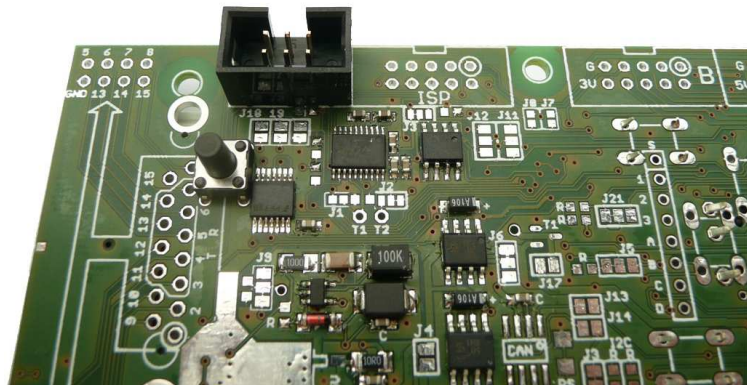
## Anschluss Programmieradapter

Um den auf der Platine befindlichen Mikrocontroller zu programmieren, brauchen Sie neben der Programmiersoftware (mehr dazu: siehe Programmierhandbuch) auch einen Programmieradapter. Dieser Programmieradapter sorgt dafür, dass die von Ihnen auf dem PC programmierte Software in den Mikrocontroller überspielt wird.

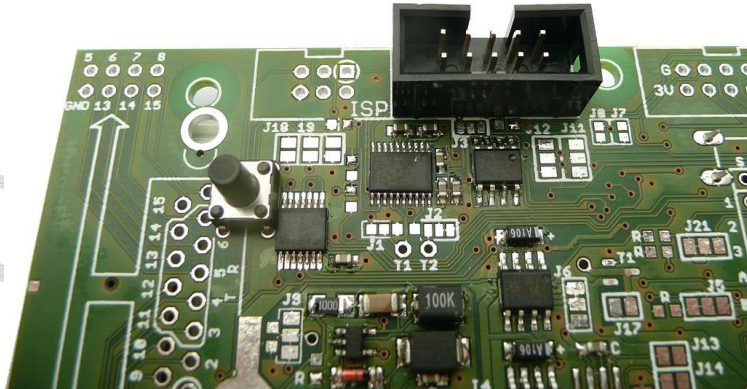
In der Regel nutzt man hierfür einen ISP-Programmer, der nicht Bestandteil des Lieferumfanges ist. Dieser ISP-Programmer verbindet Ihren PC mit diesem ATmega- Board.

Sie können entweder einen 6-poligen oder einen 10-poligen ISP Programmierstecker nutzen, das Board ist für beide Varianten vorbereitet.

Auf dem oberen Foto ist die Position des 6-poligen Stecker zu erkennen, der 10-polige Stecker liegt rechts daneben (siehe mittleres Foto). Bei beiden muss die Öffnung des Steckers nach außen zeigen.



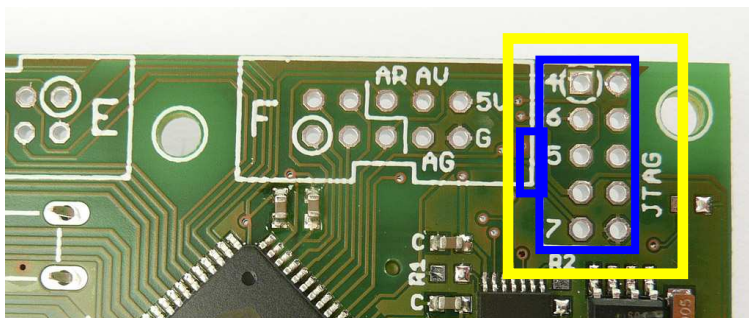
Wenn Sie mittels JTAG programmieren möchten oder die Ports F5, F6, F7 benötigen (siehe auch Kapitel Analogeingänge auf Seite 43), so stehen diese Kontakte am Stecker ganz außen rechts neben Port F zur Verfügung (siehe unteres Foto).



Allerdings können Sie hier aus Platzgründen keinen Wannenstecker einlöten, sondern müssen mit einer Stiftleiste vorlieb nehmen.

Beim Einlöten der Stiftleiste am JTAG Stecker achten Sie bitte darauf, dass die äußersten Lötunkte keinen Kontakt mit dem Metallgehäuse des SD-Karten-Slot (sofern vorhanden) auf der Rückseite bekommen – dieses liegt nämlich an Masse.

Die „Nase“ des JTAG Adaptersteckers muss beim Einstecken dann nach links zeigen (siehe blaue Markierung).





## **Benötigte Spannungen**

Der Mikrocontroller benötigt laut Datenblatt bei voller Geschwindigkeit (also 16 Mhz) eine Mindestspannung von 4,5 Volt. Mit sinkender Taktrate erlaubt das Datenblatt auch eine niedrigere Spannung. Der Erfahrung nach arbeitet der Controller auch bei 3,5 Volt mit 16 Mhz – allerdings wird er hier außerhalb der Spezifikation betrieben und ein sicherer Betrieb wird nicht garantiert.

Die sonstigen Bauteile benötigen folgende Spannungen:

- Display: 3 Volt (3 Volt Spannungsregler befindet sich auf dem Board)
- SD Karte: (3 Volt Spannungsregler befindet sich auf dem Board)
- RS232: 3,3 bis 5 Volt
- RS485: Je nach bestellten Transceiver: entweder 5 Volt oder 3 bis 5 Volt
- CAN: 5 Volt
- Echtzeituhr: 3 bis 5 Volt
- Booster für die Displaybeleuchtung: 3 bis 5 Volt

Im Lieferzustand werden Mikrocontroller, die Schnittstellen-Transceiver (RS232, RS485, CAN), sowie der Booster für die Beleuchtung mit 5 Volt betrieben.

## **Spannungsversorgung über die Port-Anschlüsse**

Selbstverständlich können Sie auch über die Portanschlüsse 5 Volt zufügen. An allen Port-Steckern mit Ausnahme von Port B liegen GND und 5V an (Port B: 3V) – Sie können dort also auch GND und 5V einspeisen.

Für die elektrisch beste Versorgung nehmen Sie in diesem Falle bitte den 5V Anschluss von Port D oder am 10er ISP Stecker (Pin 2: 5V; Pin 4,6,8,10: GND)

### **Achtung:**

Diese Spannung liegt nun hinter dem Spannungsregler an. Sie dürfen hier nicht mehr als 5V einspeisen, ansonsten werden u.U. Bauteile beschädigt.

### **Achtung:**

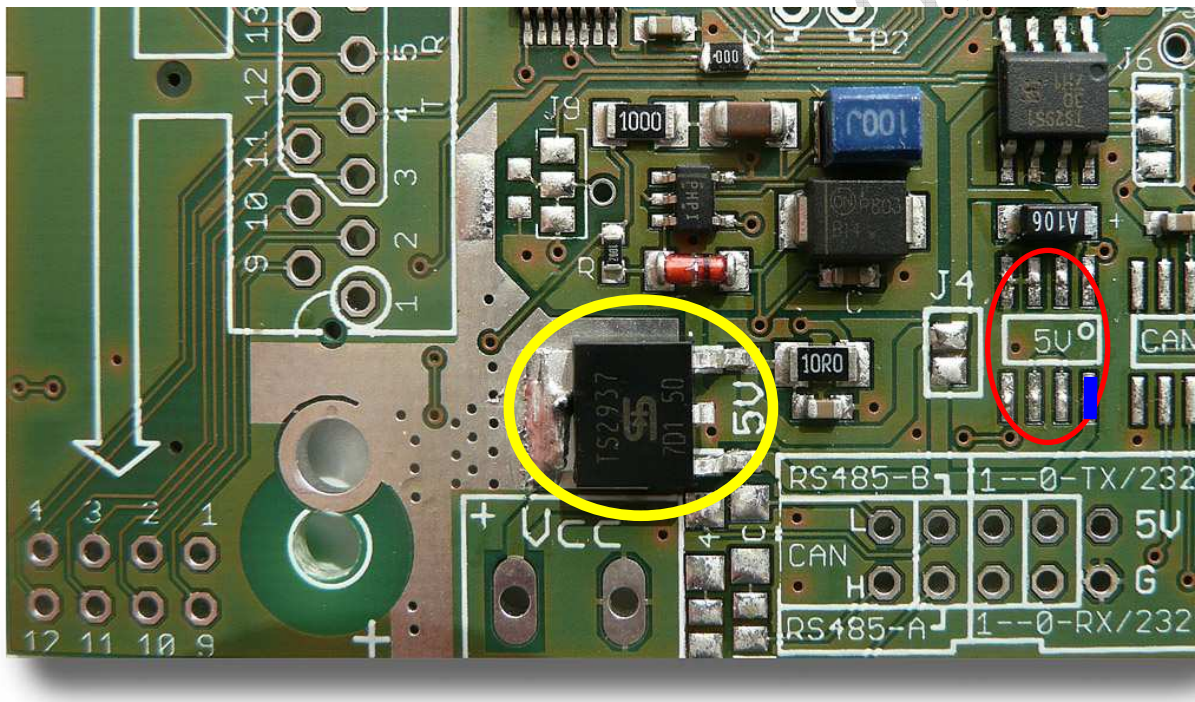
Sie dürfen bei einer solchen Einspeisung niemals gleichzeitig eine Spannung am „normalen“ VCC Eingang anschließen (also z.B. 4,5 Volt an Vcc und gleichzeitig 5 Volt an Port D einspeisen). In einem solchen Fall kann der Spannungsregler zerstört werden und liefert bei einem späteren Anschluss einer höheren Spannung (z.B. 12 Volt) dann direkt 12 Volt an alle Bausteine an – das Board wird dann sofort zerstört. Wenn Sie wissen, dass dies (doppelter Anschluss von Spannungen) in Ihrer Systemumgebung u.U. vorkommen kann, können Sie der Zerstörung des Reglers vorbeugen, indem Sie auf das Feld des Jumper 4 (siehe vorherige Seite) eine SMD Diode löten. Die Anode (also der Strich auf der Diode) muss nach unten, Richtung Platinenrand zeigen.

## „Großer“ Spannungsregler

Auf der Platine nutzen wir Linearregler, die die überschüssige Spannung in Wärme verbrennen. Im Gegensatz hierzu gibt es Schaltregler, die jedoch komplexer aufzubauen sind, teurer sind, wesentlich mehr Platz benötigen und evtl. auch Störungen verursachen.

Der Standard-5 Volt Spannungsregler im SO8 Gehäuse (siehe Foto: rote Markierung) erlaubt einen Spannungseingang von bis zu ca. 12 Volt mehr „schafft“ er aufgrund der fehlenden Kühlmöglichkeit nicht.

Um das Board mit einer höheren Spannung versorgen zu können, kann optional ein „größerer“ Regler verbaut werden, der dann eine höhere Eingangsspannung (bis ca. 18 Volt) ermöglicht. Die Platzierung dieses optionalen Spannungsregler (SMD Gehäuse TO252) erkennen Sie im folgenden Foto direkt über dem Stromanschlussstecker (gelbe Markierung).



Wenn Sie einen solchen größeren Regler einlöten, entfernen Sie bitte den Standard-Spannungsregler (rote Markierung) auf dem Board. Alternativ: Entlöten Sie Pin 8 (blaue Markierung) und heben diesen hoch, damit er keine Verbindung mehr hat. **Es dürfen nicht zwei Regler parallel in Betrieb sein.**

### Exkurs: Kühlflächen

Warum „schafft“ der kleine Regler keinen Eingang von 14 Volt bei ca. 70mA Verbrauch des Moduls, obwohl doch laut Datenblatt 150mA erlaubt sind? Dazu ein wenig Theorie:

Angenommen Sie führen 14 Volt zu, der Regler soll auf 5 Volt herunterregeln und die Schaltung benötigt 70mA.

- Dann erhalten wir eine Verlustleistung  $P_D$  von  $(14\text{Volt} - 5\text{Volt}) \times 0,07\text{A} = 630\text{mW}$
- Gleichzeitig muss auch der Eigenverlust des Reglers hinzugerechnet werden (Strom über GND üblicherweise ca. 12mA) – also  $14\text{ Volt} \times 0,012\text{A} = 0,168\text{mW}$
- Die Gesamtverlustleistung  $P_D$  beträgt also rund 800mW.

Angenommen wir erlauben dem Regler eine max. Junction-Temperatur von  $125^\circ\text{C}$  und nehmen eine maximale Umgebungstemperatur von  $40^\circ$  an – so erhalten wir  $T_R(\text{max})$  von  $85^\circ\text{C}$ .

$$\theta_{(j-a)} = T_R(\text{max}) / P_D \text{ also: } 85^\circ\text{C} / 0,8\text{W} = 106^\circ\text{C/W}$$

Der Regler im S08 Gehäuse erlaubt laut Datenblatt ohne Kühlkörper  $160^\circ\text{C/W}$  oder mehr. Sobald der errechnete Wert  $\theta_{(j-a)}$  unter diese  $160^\circ\text{C/W}$  fällt, wird der Regler zu heiß und irgendwann in die Notabschaltung gehen. Eine Kühlung ist beim S08 Gehäuse nicht möglich.

Nach obiger Rechnung ist es also NICHT möglich, das Modul mit dem „kleinen“ 5 Volt Spannungsregler und mit 14 Volt zu betreiben. Ein Nachrechnen ergibt: der Regler im S08 Gehäuse erlaubt ohne Kühlkörper maximal entweder:

- Bei 14 Volt und 70mA eine max. Umgebungstemp. von  $-3^\circ\text{C}$
- bei 14 Volt Eingangsspannung einen Dauerstrom von max. 44mA
- bzw. für eine Last von 70mA eine max. Eingangsspannung von ca. 11,2 Volt (Anmerkung: da trotzdem eine minimale Kühlung über die Anschlussleitungen stattfindet sind ca. 12 Volt noch so gerade OK).

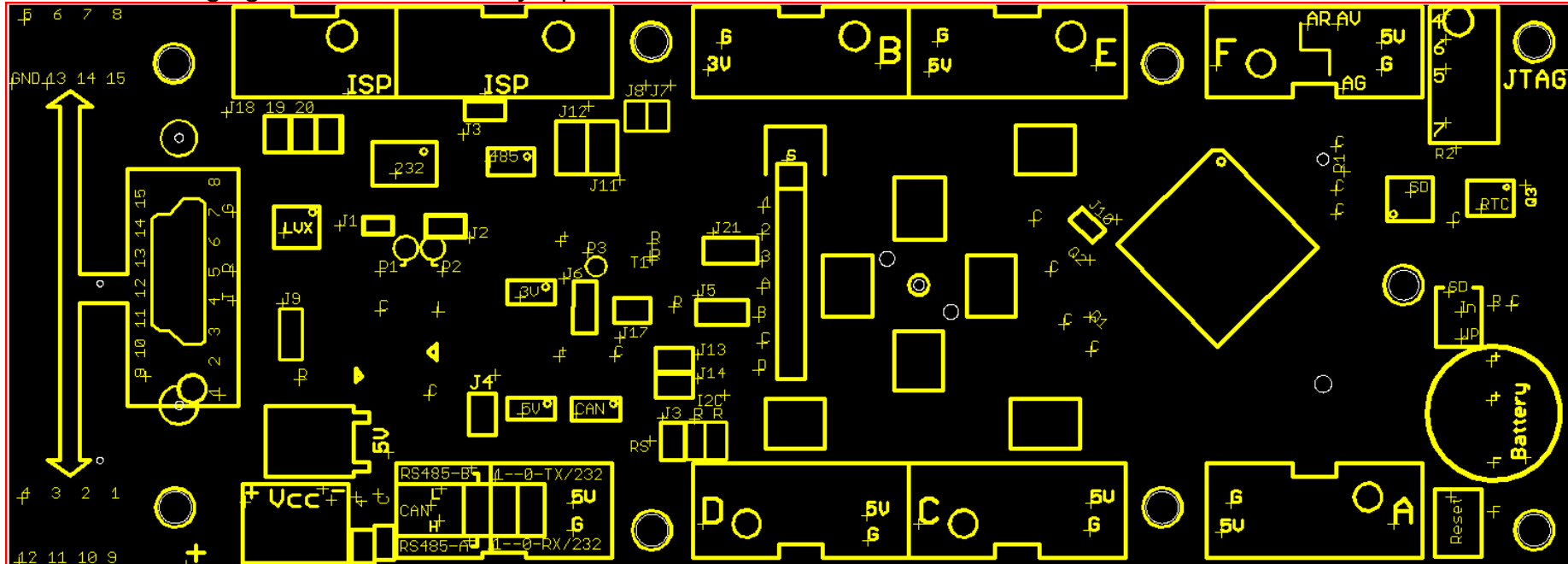
Alternative 1: ein Regler im TO220 Gehäuse erlaubt  $60^\circ\text{C/W}$ ; ein TO-263 Gehäuse immerhin noch  $80^\circ\text{C/W}$  – alles ohne externen Kühlkörper. Dies wären also evtl. sinnvolle Alternativen. Leider sind dies keine SMD Bauteile und sie nehmen zudem in der Höhe entsprechend viel Platz weg – für unseren Zweck zu viel.

Alternative 2 - Leiterbahnfläche als Kühlkörper: Ein alternativer kleinformatiger Regler im TO252 Gehäuse hat (ohne Kühlkörper) einen Wärmewiderstand von  $150^\circ\text{C/W}$ . Die erlaubten Werte steigen also gegenüber dem Regler im S08 Gehäuse nicht sonderlich an. Aber dies gilt nur auf den ersten Blick, denn hier können wir, da es sich um ein SMD Bauteil mit Kühlfläche handelt, die Kupferfläche der Platine als Kühlkörper mitbenutzen. Als Faustregel kann man sagen:  $5\text{cm}^2$  Leiterbahnfläche als Kühlkörper (ohne Lötstopplack!) senken  $\theta_{(j-a)}$  von  $150^\circ\text{C/W}$  auf ca. die Hälfte ab.

Auf unserer Platine D071x nutzen wir beidseitige Flächen als Kühlkörper, die jeweils miteinander mit vielen Durchkontaktierungen thermisch miteinander verbunden sind (die Wärmedurchlässigkeit des FR4 Materials ist vernachlässigbar und kann als nicht gegeben angesehen werden). Wir schätzen hier einen Wärmewiderstand von ca.  $90\text{-}100^\circ\text{C/W}$ . Rechnerisch sind somit bei max.  $40^\circ\text{C}$  Umgebungstemperatur ca. 16 Volt als maximale Eingangsspannung möglich. Bei  $25^\circ\text{C}$  Umgebungstemperatur sind es sogar ca. 18 Volt.

## Anschlusspads und Jumper:

Die Anschlussbelegung der Stecker sowie der Jumper ist im untenstehenden Bild zu ersehen.



Pin 1 ist jeweils mit einem Kreis im Bestückungsdruck gekennzeichnet. Um Ihnen die Arbeit mit extern angeschlossenen Bausteinen zu erleichtern, haben wir an jedem Port-Stecker an Pin 10 auch nochmals die Spannungsversorgung 5V und an Pin 9 GND gelegt.

**Ausnahme:** Port B, hier liegen statt 5 Volt immer 3,0 Volt an.

Die anderen Pins der Wannenstecker sind identisch belegt (hier Beispiel Port A)

Pin 1: PortA 0

Pin 2: PortA 1

Pin 3: PortA 2

Pin 4: PortA 3

Pin 5: PortA 4

Pin 6: PortA 5

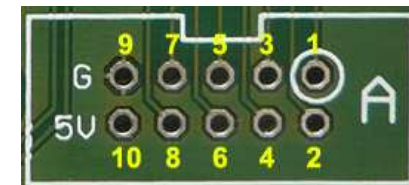
Pin 7: PortA 6

Pin 8: PortA 7

Pin 9: GND

Pin 10: Vcc

**Tip:** Wenn Sie externe Geräte zum Datenaustausch mit dem Board verbinden möchten, lesen Sie auch das Kapitel zur Terminierung im Anhang dieses Manuals.



Verschiedene Optionen des Boards belegen einige der vorhandenen Ports. Diese Ports sind dann u.U. für Sie nicht anderweitig nutzbar. Details hierzu entnehmen Sie bitte der Tabelle Portbelegung am Ende des Dokuments (Seite 67).

**Besonderheiten:** Die Stecker für Port F und Port A werden um 180° gedreht eingelötet (anders als bei den anderen Stecker sind hier die Öffnungen also nach innen gerichtet).

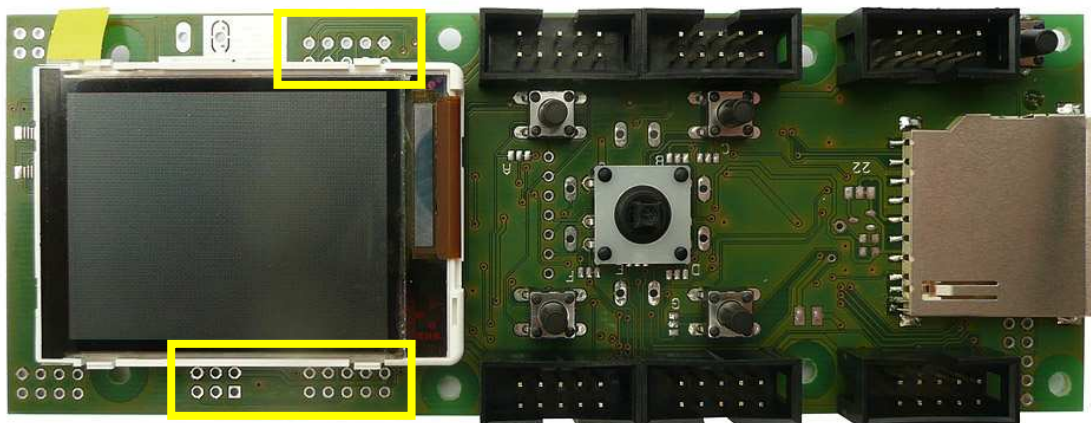
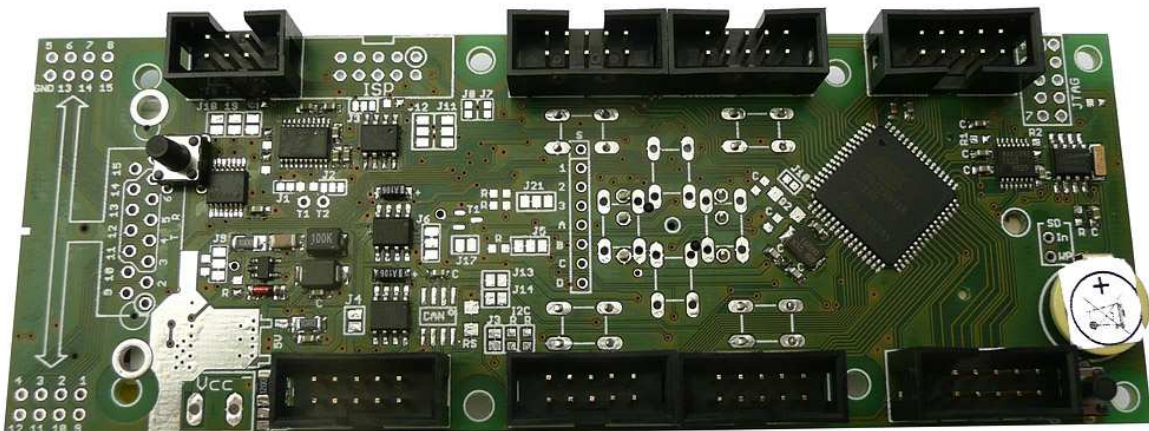
Am Stecker für Port F sind nur F0-F4 angelegt, die restlichen Pads werden belegt durch ARef, AGND und AVCC. Sofern Sie auch Port F5-F7 benötigen, greifen Sie diese bitte am nebenstehenden JTAG Stecker ab (die Pads F4-F7 liegen dort ebenfalls an und sind mit „4, 5, 6, 7“ gekennzeichnet).

Die Bedeutung der verschiedenen Jumper, der Lieferzustand sowie die Position der Jumper entnehmen Sie bitte der Tabelle Jumperbelegung am Ende des Dokuments ab Seite 67.

## Die Anschlussmöglichkeiten

### Wannenstecker

Das Board ist für den Anschluss von Wannensteckern vorbereitet. Diese können sowohl von unten als auch (größtenteils) von oben eingelötet werden. Da jedoch ein solches Modul wegen des Displays meist hinter einer Frontplatte oder einem Gehäuse verbaut wird, ist die vorgesehene Einbaurichtung unten (Bauteilseite, wie auf dem oberen Foto zu sehen).

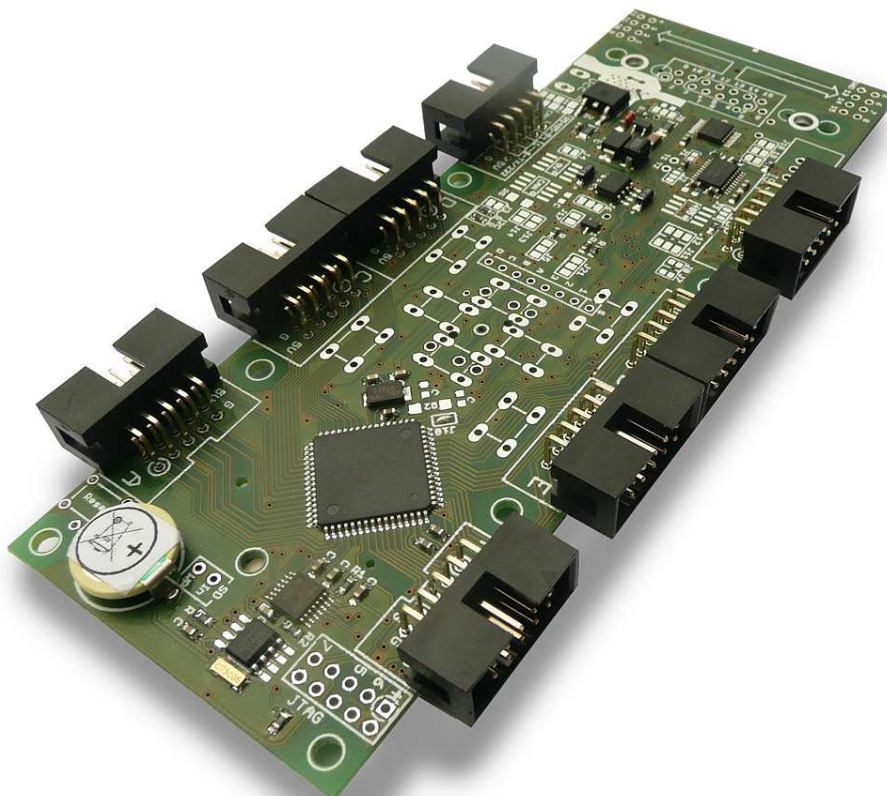
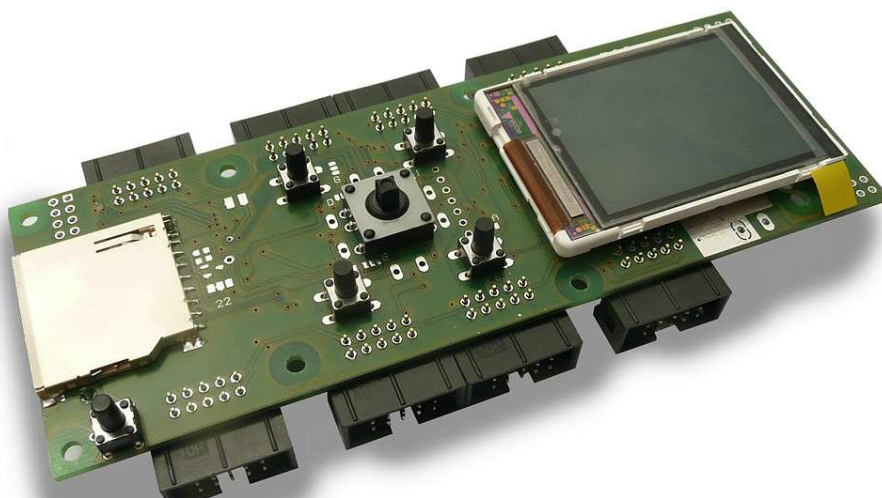




Der Einbau auf der Displayseite (siehe Foto oben) ist ebenfalls eine Alternative, jedoch reicht der Platz direkt über- und unterhalb des Displays nicht für einen Wannenstecker aus.

Achtung: Wenn Sie den Einbau in unser Gehäuse GBS600 planen: Im Gehäuse mit Batteriefach (GBS601) können wegen des Aufbaus des Batteriefachs nicht an allen Stellen Stecker verbaut werden – das Gehäuse ließe sich dann nicht mehr schließen. **Bitte prüfen Sie zuerst die Einbausituation in dem Sie die Stecker nur einstecken und erst einmal nicht verlöten – eine Tabelle mit der Info was wann wo passt, finden Sie im Anhang auf Seite 89.**

Sonstige Alternative: Auch gewinkelte Wannenstecker können eingesetzt werden und machen je nach Einbausituation Sinn. **Achten Sie dann aber darauf, dass der Stecker für Port F und Port A aufgrund der um 180° gedrehten Kontaktbelegung anders belegt ist.**

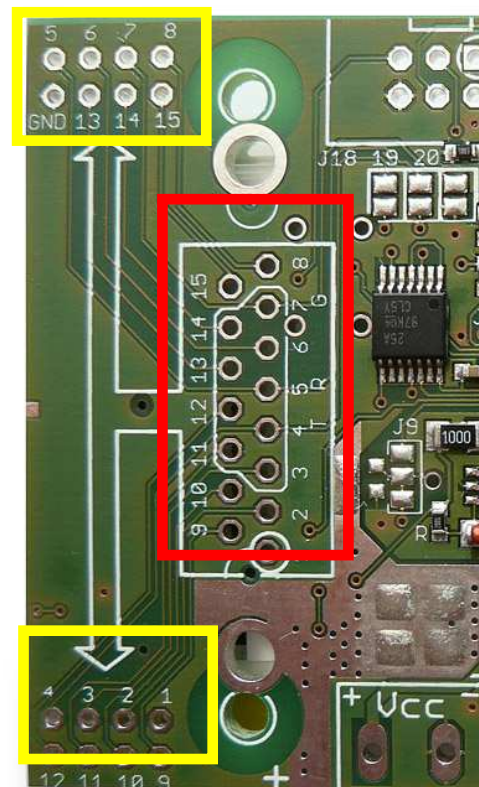


## 15-poliger Standard D-Sub-Stecker

Anmerkung: Der im Folgenden bei D-Sub genutzte Begriff *Stecker* gilt übergreifend als Synonym und kann natürlich genauso auch eine *Buchse* sein.

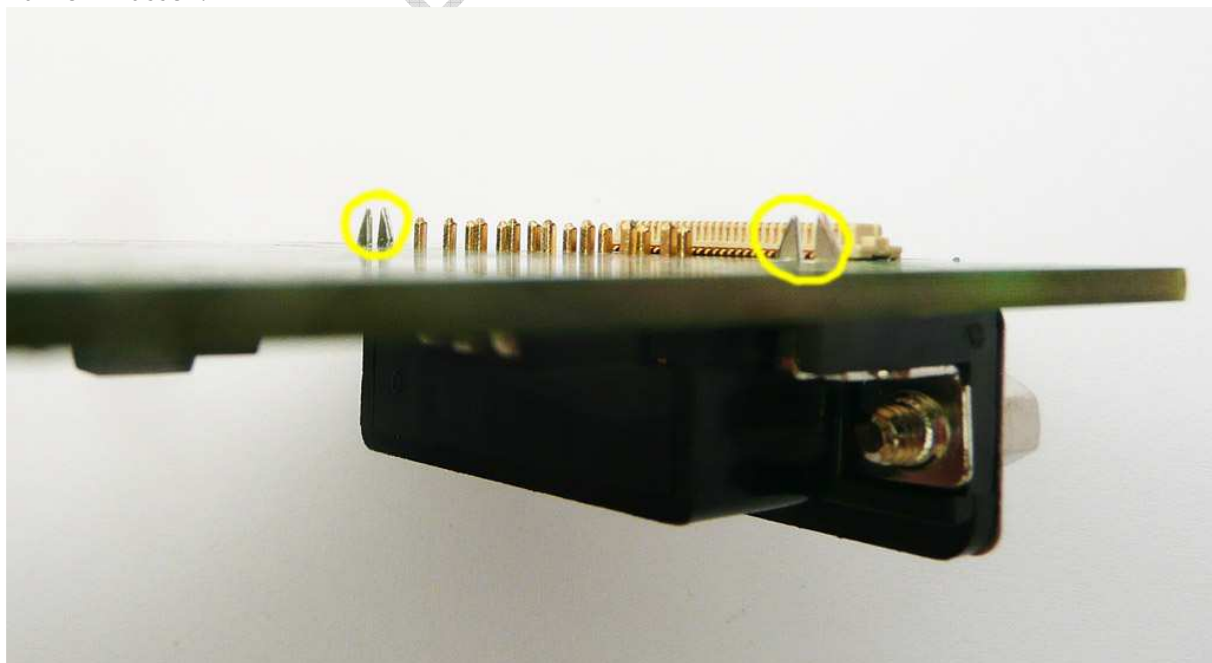
Am linken Rand bietet das Modul die Option einen D-Sub-Stecker (oder Buchse) einzustecken (siehe Foto: rote Markierung). Dieser Stecker ist noch nicht mit Ports verbunden. Sie können diesen frei belegen. Hierzu finden Sie oberhalb und unterhalb des Steckers freie Pads, die mit der Nummerierung des Steckers konform gehen (gelbe Markierung). Wenn Sie also Pin 9 des Steckers mit Port A5 verbinden möchten, verbinden Sie mittels eines Kabels den Port A5 mit dem Lötpad am Rand der Platine, welches mit „9“ gekennzeichnet ist.

Optional können Sie die Signale Tx und Rx einer RS232 Schnittstelle direkt an den Stecker verbinden (siehe Abschnitt „Verbinden der D-Sub-Stecker mit der seriellen Schnittstelle UART1“ weiter unten) um – diese beiden Signale sowie GND sind bereits vorbereitet.



### Evtl. notwendiges Kürzen der Halteklammern

Die Halteklammern der D-Sub Stecker sind u.U. zu lang und stören den korrekten Sitz des Displays. In diesem Fall sollten Sie diese Halteklammern mit einem Seitenschneider vor oder nach dem Einlöten etwas kürzen (d.h. das abstehende Ende abknipsen). Die Lötanschlüsse der Steckerkontakte sind i.d.R. kürzer und stören das Display nicht, so dass Sie diese nicht kürzen müssen.

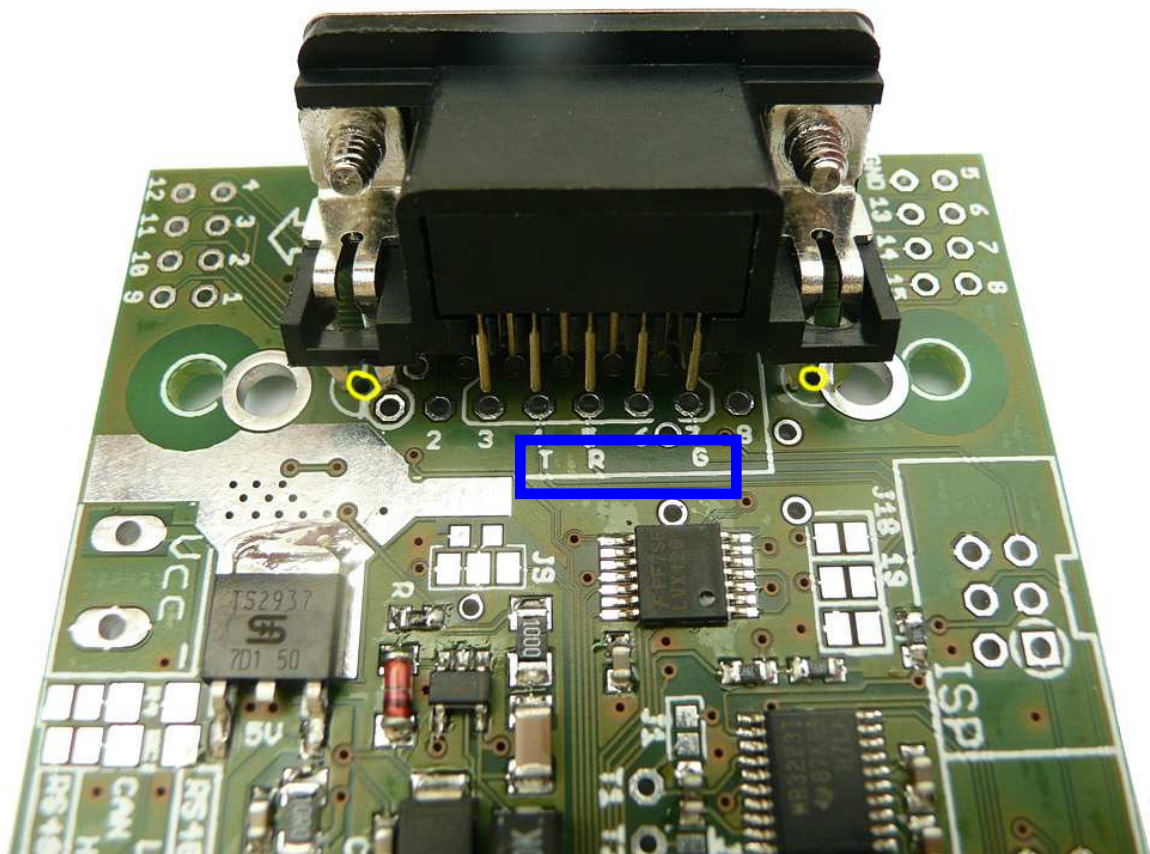


## 9-poliger D-Sub-Stecker

Optional können Sie statt des 15-poligen Steckers auch einen 9-poligen D-Sub-Stecker (oder Buchse) einsetzen. Für die korrekte Positionierung finden Sie einen entsprechenden Aufdruck auf der Platine (siehe Foto). Dieser Stecker hat den Vorteil, dass er einen schnellen Aufbau einer seriellen Verbindung erlaubt, denn der 9-polige D-Sub-Stecker ist bereits an den richtigen Stellen mit den seriellen Signalen RX und TX sowie GND verbunden. Lediglich 3 Jumper müssen dafür geschlossen werden (siehe Abschnitt „Verbinden der D-Sub-Stecker mit der seriellen Schnittstelle UART1“ weiter unten). An den im Foto unten mit „T“, „R“ und „G“ gekennzeichneten Stellen liegen dann die Signale Tx, Rx und GND an. Die Beschaltung entspricht dem Standard, so dass Sie das Mikrocontrollermodul direkt per seriellm Kabel mit einem PC verbinden können.

Ein Kompromiss musste jedoch eingegangen werden, um einen 9-poligen DSub-Stecker im Feld eines 15-poligen DSub-Steckers zu ermöglichen: Für die Halteklammern des 9-poligen Steckers konnten keine Bohrungen vorgesehen werden, denn diese liegen im Bereich der Löt pads des 15-poligen Steckers. Wenn Sie sicher sind, keinen 15-poligen Stecker zu benötigen, können Sie die Löcher für die Halteklammern selbst schnell mit einem 3mm Bohrer bohren. Wir haben diese beiden Stellen bereits gekennzeichnet und mit einem 0,8mm Loch vorgebohrt (siehe im Foto die beiden gelben Kreise). Beachten sie bitte auch den Abschnitt zum evtl. notwendigen Kürzen der Halterung auf der vorherigen Seite.

Alternativ können Sie natürlich die beiden Metallhalterungen des Steckers einfach abknipsen.

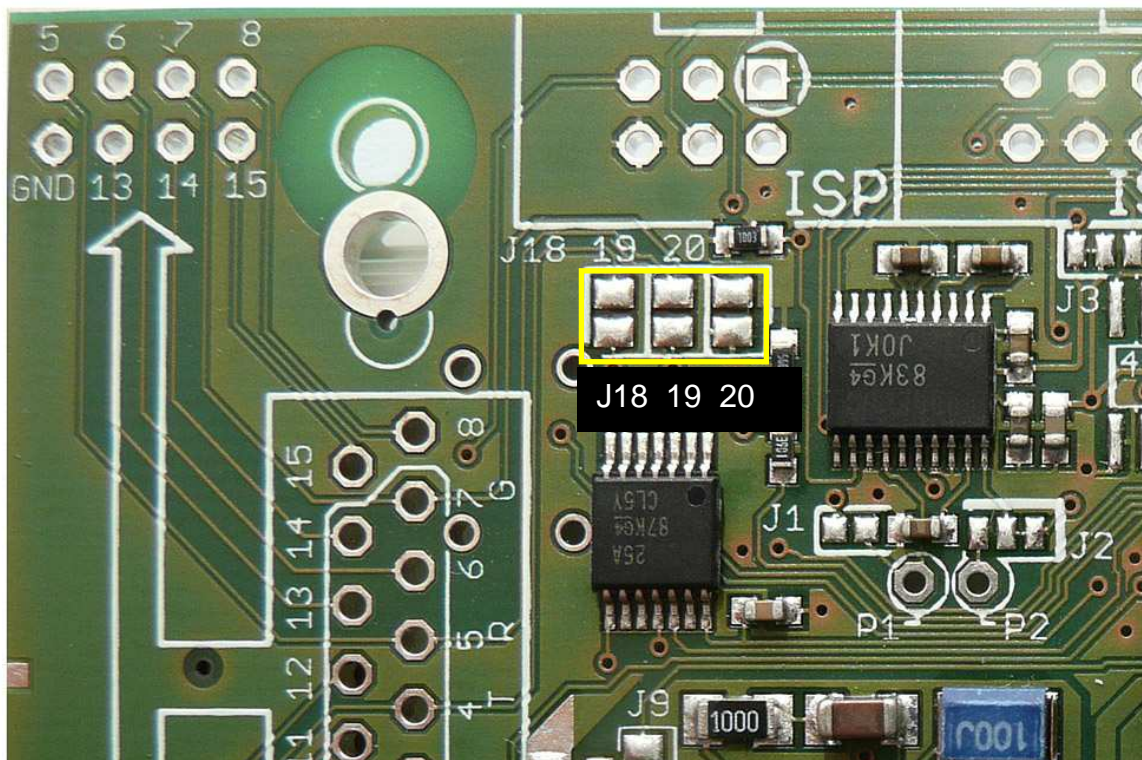


Bei der Nutzung des 9-poligen D-Sub-Steckers stimmt natürlich die Nummerierung am Rand der Kontakteleiste nicht mehr. In der folgenden Tabelle ist die korrekte Belegung aufgelistet:

Beschriftung am Platinenrand	Belegung an D-Sub-15	Belegung an D-Sub-9
1	1	-
2	2	-
3	3	1
4 (evtl. Tx)	4	2
5 (evtl. Rx)	5	3
6	6	4
7 (evtl. GND)	7	5
8	8	-
9	9	-
10	10	-
11	11	6
12	12	7
13	13	8
14	14	9
15	15	-
GND	-	-

### Verbinden der D-Sub-Stecker mit der seriellen Schnittstelle UART1

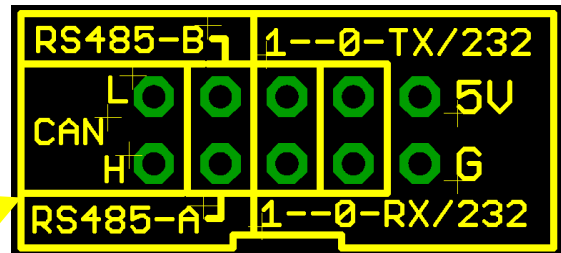
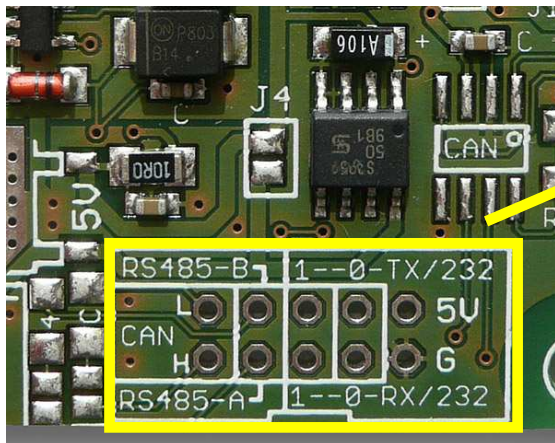
Lediglich drei Jumper (Nr. 18, 19 und 20) müssen Sie schließen um den DSUB connector mit den High-Level RS232 Signalen der seriellen Schnittstelle UART1 zu verbinden.



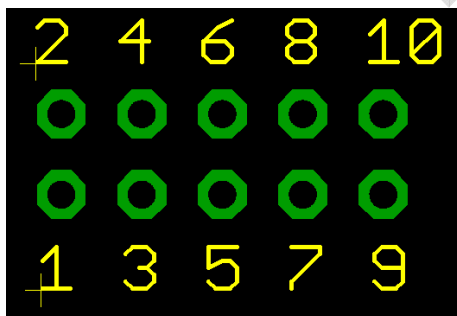
### Anschlüsse Schnittstellen (RS232, RS485, CAN)

Rechts neben dem Anschluss für die Versorgungsspannung befindet sich ein Verbindungsfeld für einige der möglichen Schnittstellen:

- UART0 (RS232 High Pegel Rx, Tx)
- UART1 (RS232 High Pegel Rx, Tx)
- RS485 (A & B)
- CAN Bus (H & L)



Der besseren Übersicht halber (die Beschriftung am Verbindungsfeld ist aufgrund des begrenzten Platzes sehr gedrängt) finden Sie nachfolgend die Belegung dieses Felds:



Pad	1	2	3	4	5	6	7	8	9	10
<b>Belegung</b>	CAN H	CAN L	RS485 A	RS485 B	RS232 RXI	RS232 TXI	RS232 RX0	RS232 TX0	GND	5V

**Achten Sie darauf, dass an diesen Leitungen Signale mit negativen Spannungen oder auch Spannungen von 12V und mehr anliegen können. Ein unvorsichtiger Umgang (z.B. eine Verbindung von diesen Löt pads zu einem Port des Controllers) kann das gesamte Board beschädigen oder zerstören. Diese Leitungen dürfen nur zu seriellen High-Pegel-Leitungen anderer Systemen geführt werden.**

**Achtung:** An Tx0 / TxI können Sie ohne Anschluss einer Gegenstelle keine Signale messen, da der RS232 Transceiver ohne Gegenstelle in den automatischen Shutdown geht.

## Die Eingabemöglichkeiten

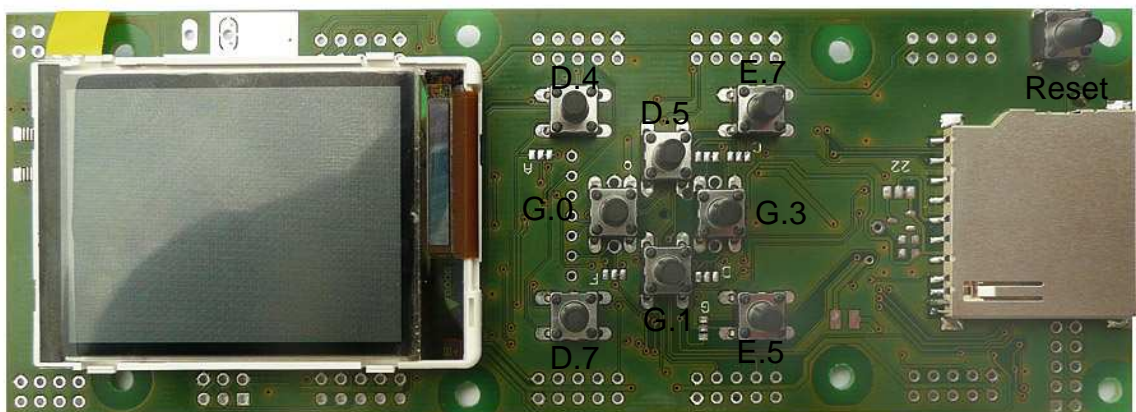
### Taster

Die Taster des Moduls sind bereits mit verschiedenen Ports (meist G und D) verbunden und ziehen die entsprechende Leitung beim Drücken auf Masse. **Wenn Sie die Taster (oder die optionale Folientastatur) nutzen möchten, so achten Sie darauf, dass Sie diese Tasterports nur als Eingang und nicht versehentlich als Ausgang definieren – Sie würden dann beim Drücken eines Tasters unweigerlich einen Kurzschluss verursachen und diesen Port zerstören.**

Die richtige Vorgehensweise ist hier:

- 1) Tasterports als Eingang definieren
- 2) Die internen Atmel Pullup-Widerstände dieser Tasterports einschalten

Somit liegen die Taster zwar an einem Potential von 5V, jedoch immer nur über die internen Pullup-Widerstände. Der abgefragte Eingang zeigt also ein high (1) an. Beim Drücken des Tasters wird nun dieser Eingang gegen Masse gezogen und beim Abfragen dieses Ports zeigt dieser ein Low (0). Das Ziehen gegen Masse bedeutet hier keine Gefahr, denn der Pullup-Widerstand erlaubt nur einen sehr geringen Strom.

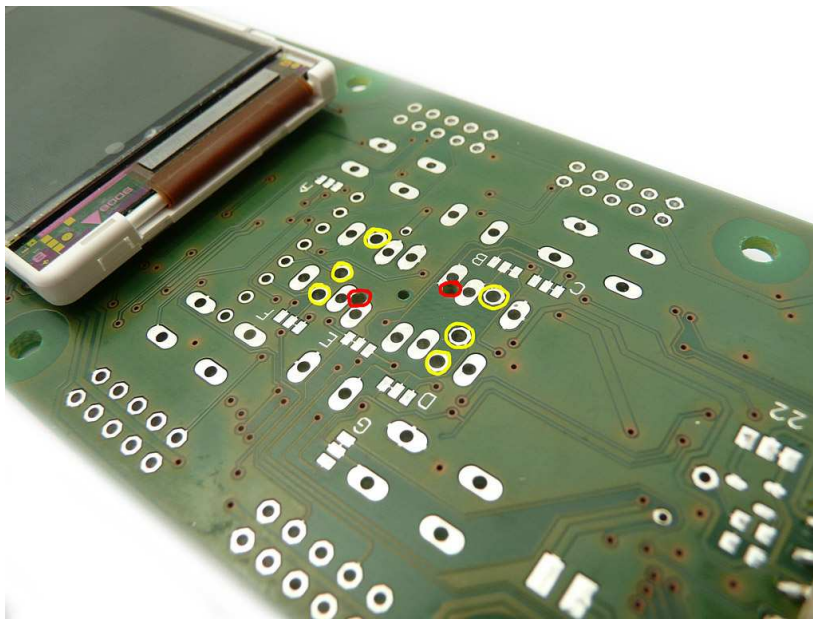


Die oben gezeigte Beschriftung zeigt die Tasterbelegung im Lieferzustand. Bei einem solch multifunktionalen Board ist es jedoch wahrscheinlich, dass Sie einen bestimmten Port für eine andere Aufgabe benötigen. Jeder einzelne Taster kann daher alternativ einem anderen Port zugewiesen werden. Mehr hierzu erfahren Sie am Ende dieses Kapitels (Seite 28).

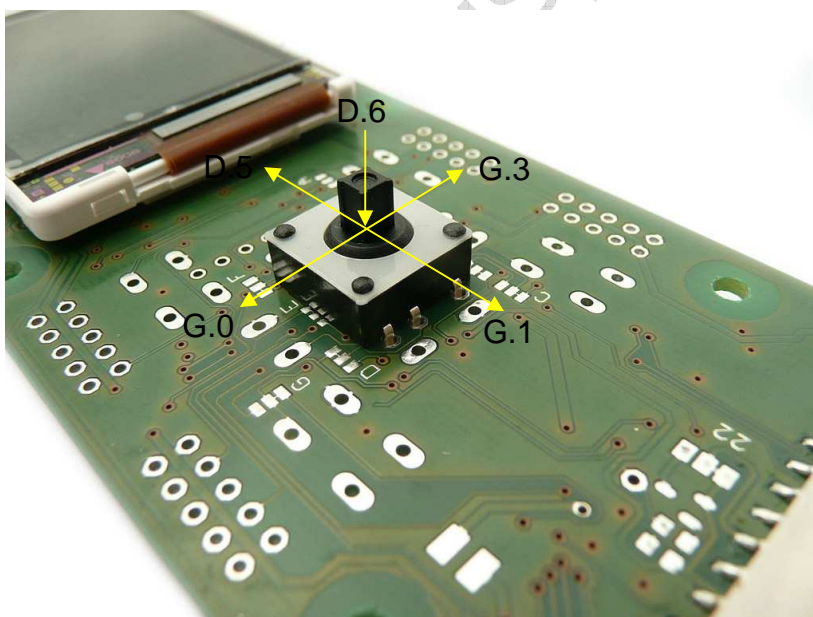
**Tipp:** Die Taster müssen (zumindest anfangs) nicht von Ihnen eingelötet werden. Sie halten in der Platine nach dem Einstecken auch ohne Löten und auch der Kontakt ist i.d.R. einwandfrei. Vor allem, wenn Sie die Platine im Gehäuse nutzen, können Sie die Taster auch erst einmal unverlötet belassen. Dies hat den Vorteil, dass Sie sie auch jederzeit einfach wieder entnehmen können, wenn Sie sich doch für eine Folientastatur entscheiden (evtl. bereits gebohrte Löcher im Gehäuse sind nicht schlimm, denn sie werden ja dann von der Folientastatur verdeckt).

## Joystick

Alternativ zu den mittleren vier Tastern können Sie auch einen Joystick einsetzen, die Portverbindungen der Richtungen jedoch nicht identisch zu den Tastern. Die Position dieses Joysticks wird im Folgenden gezeigt. Die gelben Markierungen zeigen die 6 Kontaktierungspads und die beiden roten Markierungen zeigen die Löcher für die Positionierstifte des Joysticks. Auch der Joystick kann erst einmal ohne Einlöten verwendet werden. Dies erleichtert eine spätere Änderung auf eine alternative Eingabemöglichkeit.



Da der Joystick an den gleichen Leitungen wie die Taster liegt, ist auch hier eine Umkonfigurierung der genutzten Ports möglich. Zusätzlich zum Joystick können die 4 Taster an den Außenpositionen genutzt werden. Der Joystick besitzt zusätzlich zu den 4 Richtungspositionen noch eine Druckfunktion (Push) – diese liegt standardmäßig auf Port D6.



Tipp: Joystick und Taster müssen (zumindest anfangs) nicht von Ihnen eingelötet werden. Sie halten in der Platine auch ohne Löten und auch der Kontakt ist i.d.R. einwandfrei. Vor allem, wenn Sie die Platine im Gehäuse nutzen, können Sie die Taster auch erst einmal unverlötet belassen. Dies hat den Vorteil, dass Sie sie auch jederzeit einfach wieder entnehmen können, wenn Sie sich später doch für eine Folientastatur entscheiden (evtl. gebohrte

Löcher werden ja von der Folientastatur verdeckt).

### **Folientastatur**

Speziell zum Gehäuse GBS600 bzw GBS601 haben wir eine Folientastatur entwickelt und produzieren lassen.

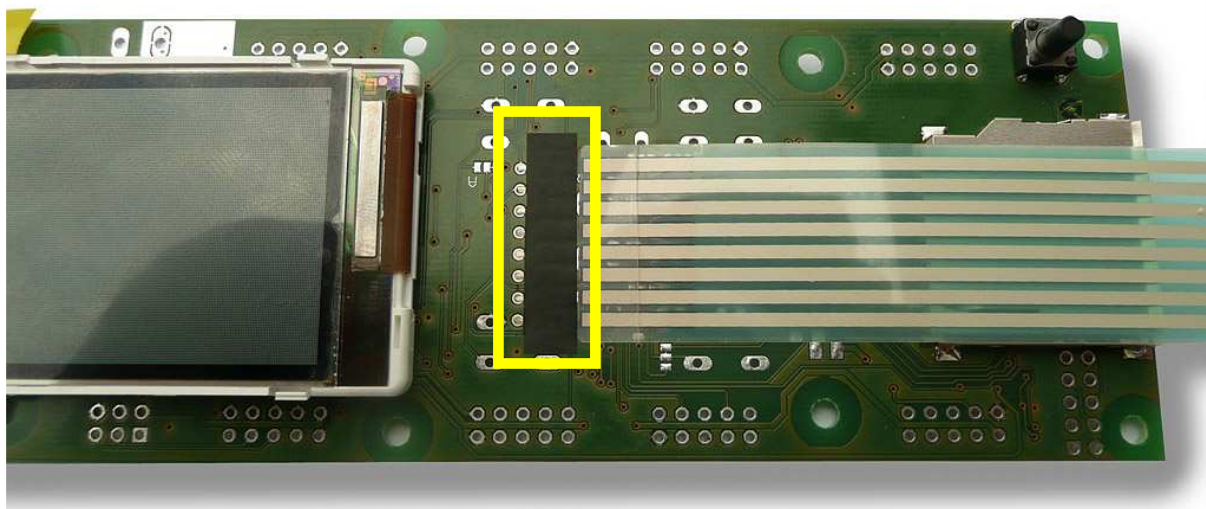
Übrigens: Ab ca. 50-100 Tastaturen macht Sinn, eine eigene Tastatur fertigen zu lassen. Ab ca. 30 Stück kann die vorhandene Tastatur auch speziell für Sie mit einer geänderten Bedruckung und/oder anderen Farben nachproduziert werden (andere Beschriftung der Tasten, Firmenlogo etc.). Sprechen Sie uns an!

Für den Anschluss dieser Tastatur löten Sie bitte die Folienkabelbuchse wie unten gezeigt ein. Das Folienkabel der Tastatur wird dann einfach eingesteckt. Tipp zum einfachen Einstecken: Legen Sie das Folienkabel auf der Platine auf und schieben das Ende in den Stecker, dann schieben/drücken Sie das Kabel mit dem Daumen in den Stecker herein.



Die 8 Leitungen der Folientastatur sind in dieser Reihenfolge mit S, 1, 2, 3, A, B, C, D bezeichnet.

Auf dem unteren Bild befindet sich die Leitung „S“ ganz unten und die Leitung „D“ ganz oben.





Die Belegung der Folientastatur ist als Matrix wie folgt ausgeführt:

	<b>S</b>	<b>I</b>	<b>2</b>	<b>3</b>
<b>D</b>		1	2	3
<b>C</b>		4	5	6
<b>B</b>		7	8	9
<b>A</b>	Shift		0	Enter

Warum eine Matrix: Ohne Matrixverschaltung müssten wir für die 12 Taster der Tastatur insg. 12 Ports opfern – so reichen für 11 Tasten 7 Ports aus (Leitungen A,B,C,D,1,2,3). Die Taste Shift erhält noch eine eigene Leitung (S).

### Das Abfragen der Matrix:

Das Abfragen einer Matrix ist programmtechnisch komplexer als bei Einzeltastern.

Zum besseren Verständnis ist nebenstehend die Tastatur nochmals mit den zugehörigen Portverknüpfungen abgebildet. Der standardmäßig genutzte Port ist größer gedruckt. Ein eventuell daneben / darunter gezeigter kleiner gedruckter Port ist eine mögliche Alternative, die durch Umkonfigurierung der Taster-Jumper genutzt werden kann (mehr hierzu auf der übernächsten Seite).

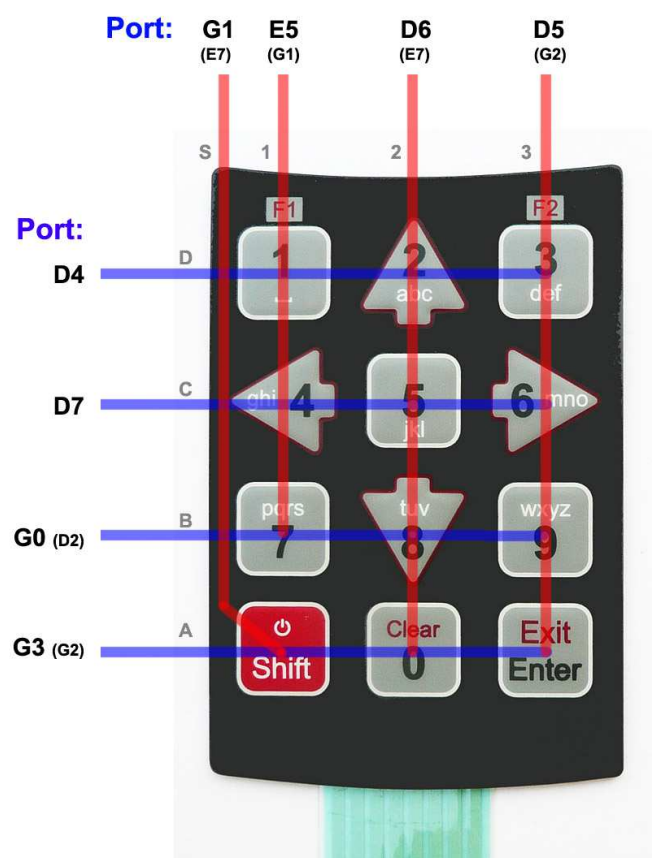
Nachfolgend finden Sie ein Beispiel für das Prinzip dieser Abfrage:

1) Alle Ports für die Tasterabfrage werden zuerst einmal auf Eingang (**Achtung: NICHT AUF AUSGANG – dies würde den betreffenden Port zerstören!**) geschaltet. Alle Pullup-Widerstände werden eingeschaltet.

2) Nun wird die Reihe D (also Port D4) als Ausgang umkonfiguriert und auf „0“ gesetzt.

3) Nacheinander werden die Eingänge der Leitungen 1, 2, 3 geprüft (also Port E5, D6, D5). Ist eine dieser Leitungen „0“, so kann die Taste aus der Matrix abgelesen werden. Die Spalte „S“ (Port G1) muss nur bei Reihe „A“ (Port G3) geprüft werden, hier ist die einzig mögliche Verbindung. Nach der Abfrage darf nicht vergessen werden, den auf Ausgang geschalteten Port wieder auf Eingang mit aktivem Pullupwiderstand zu setzen.

4) Das Gleiche (also Konfigurierung als Ausgang und Abfragen der Spalten) passiert nun für die Reihen C,(Port D7), B (Port G0), A (Port G3) bevor wieder mit „D“ begonnen wird.



Beispiel:

Reihe D: Port D4 auf Ausgang und auf Low; Abfragen von Port (Pin) E5 (=Spalte 1) zeigt eine „0“: Dies bedeutet: die Taste „1“ wurde gedrückt, denn beim der Eingangsport E5 mit seinem aktiven Pullupwiderstand wird normalerweise ein Highpegel gemessen. Wenn er jedoch Low ist, muss der Port gegen Masse gezogen worden sein.

Reihe C: Port D7 auf Ausgang und auf Low; Abfragen von Spalte 2 (Port D6) ergibt eine „0“: d.h. die Taste „5“ wurde gedrückt.

Reihe B: Port G0 auf Ausgang und auf Low; Abfragen der Spalten 1 bis 3 (E5, D6, D5) zeigt: alle Leitungen sind auf High (1): d.h. keine Taste wurde gedrückt.

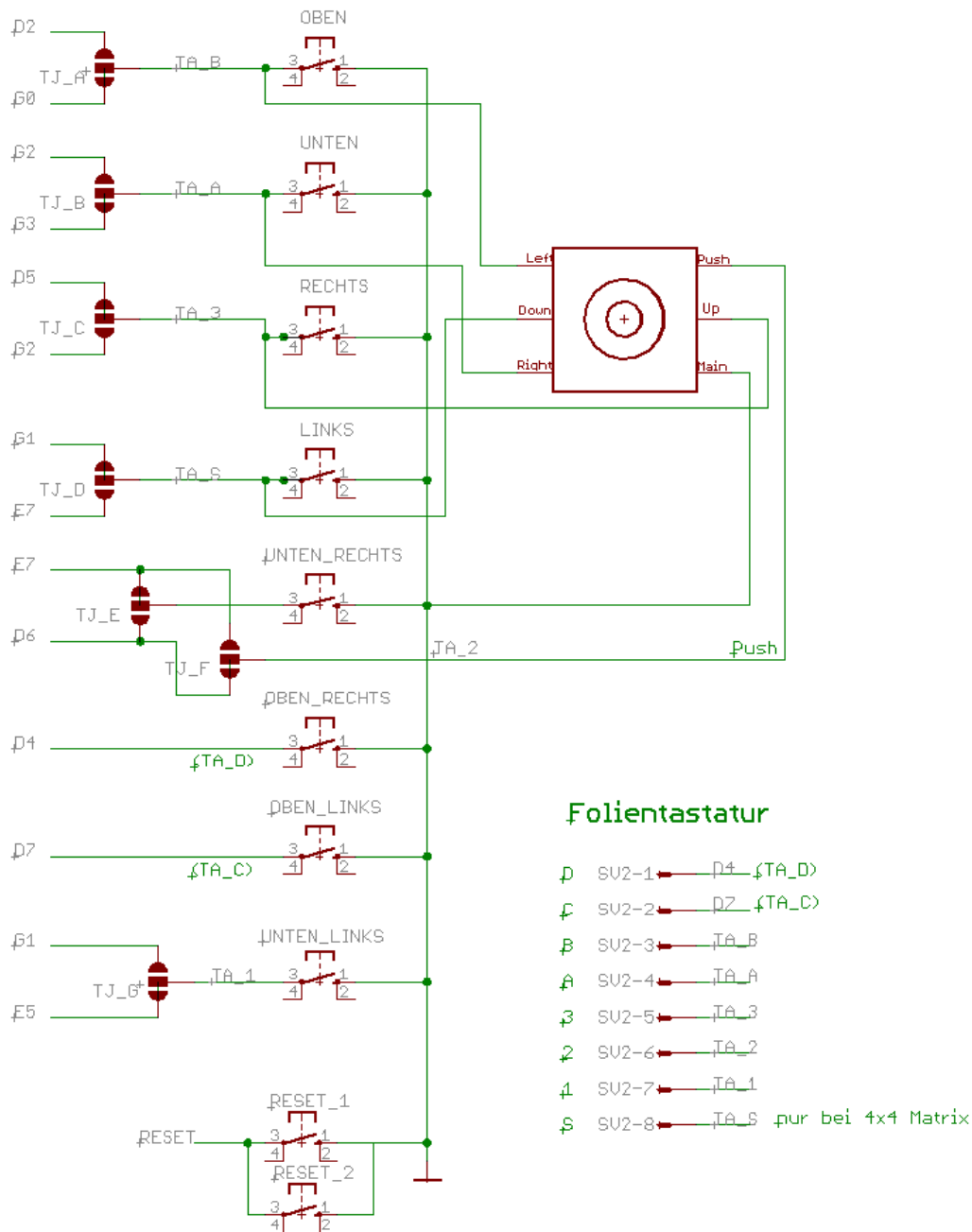
Reihe A: Port G3 auf Ausgang und auf Low; Abfragen von Spalte S (Port G1) ergibt eine „0“; Außerdem ergibt Spalte 3 (Port D5) eine „0“: d.h. die Taste „Shift“ sowie die Taste „Enter“ wurden gedrückt – der Anwender wünscht also einen Abbruch (Shift-Enter = Exit)

Zusammenfassung: Die auf Ausgang geschalteten Ports dienen dazu, der Reihe nach die auf High liegenden Spalten gegen Masse zu ziehen, wenn der betreffende Taster gedrückt wurde. Dieser Wechsel von High nach Low kann dann detektiert werden.

**Sie dürfen NIEMALS einen Tasterport als Ausgang definieren und auf High schalten**, denn ansonsten würde es bei gedrücktem Taster zu einem internen Kurzschluss kommen und die beiden betreffenden Ports (der Ausgang mit High sowie gegen Masse ziehende Port) würden zerstört.

**Ganz wichtig:** High für Taster etc. wird immer nur so erzeugt: Port als Eingang definieren und den (Atmel internen) Pull-Up-Widerstand einschalten. Der Pullup-Widerstand sorgt für einen definierten High-Pegel und kann gefahrlos gegen Masse gezogen werden, denn es fließt nur ein sehr geringer Strom.

## Beschaltung des Tastenbereichs:

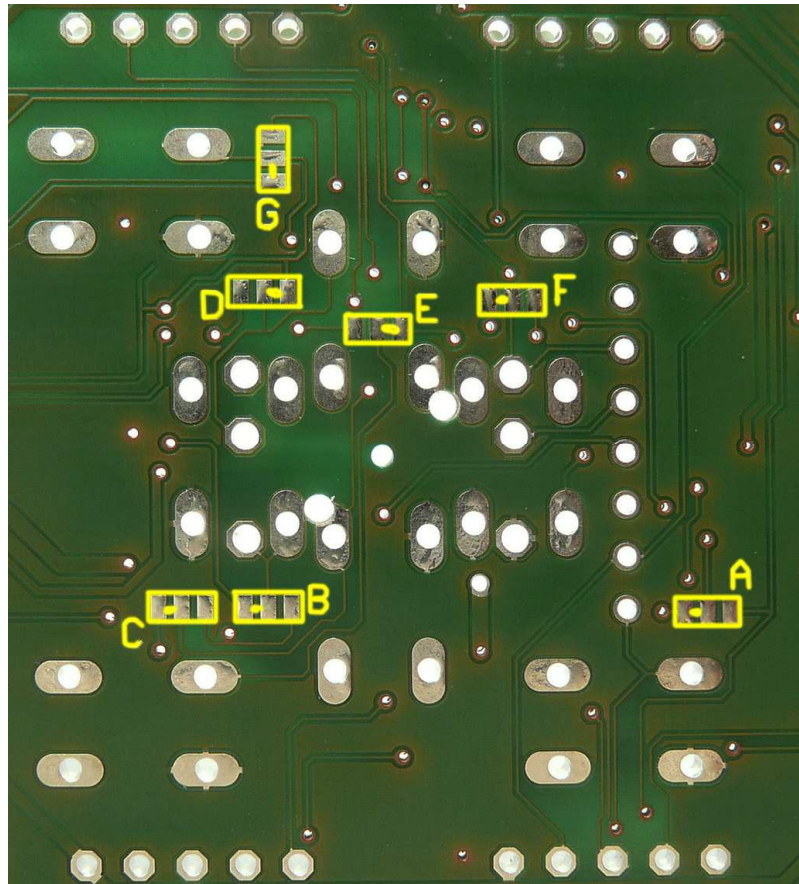


Die Standardbelegung der Jumper ist immer durch die Verbindung innerhalb des Jumpers ersichtlich (Beispiel: T\_J\_A kann den Taster „oben“ entweder an Port D2 oder G0 verbinden. Standardmäßig ist die Verbindung an G0 geschlossen).

**Achtung:** Durch einen Fehler in der Bauteillibrary entsprechen die genutzten Ports der Joystick-Richtungen *Oben*, *Unten*, *Rechts*, *Links* nicht den Tastern der gleichen Richtung. Die Tabelle auf der folgenden Seite verdeutlicht die Belegung.

## Umkonfigurierung der benutzten Taster-Ports

Unter Umständen werden Sie einen der von uns für den Tasterbereich vorgesehenen Ports für eine andere Aufgabe benötigen. Daher haben wir für einige Taster einen alternativen Port vorgesehen. Mittels der Tasterjumper TJA bis TJG können die Taster umkonfiguriert werden. Diese Jumper sowie deren Beschriftung finden Sie auf der Platinenoberseite (Displayseite). Bei diesen Jumpers muss immer das mittlere Feld mit einem der daneben liegenden Felder verbunden werden. Soll eine Verbindung geändert werden, trennen Sie diese innerhalb des Jumperfelds mit einem Messer auf und verbinden dann das mittlere mit dem anderen Feld durch einen Tropfen Lötzinn. Die Verbindungen im Lieferzustand sind gelb markiert.



Es ist nicht möglich, jedem Taster einen anderen Port zuzuweisen, einige Alternativports werden auch für mehrere Taster verwendet.

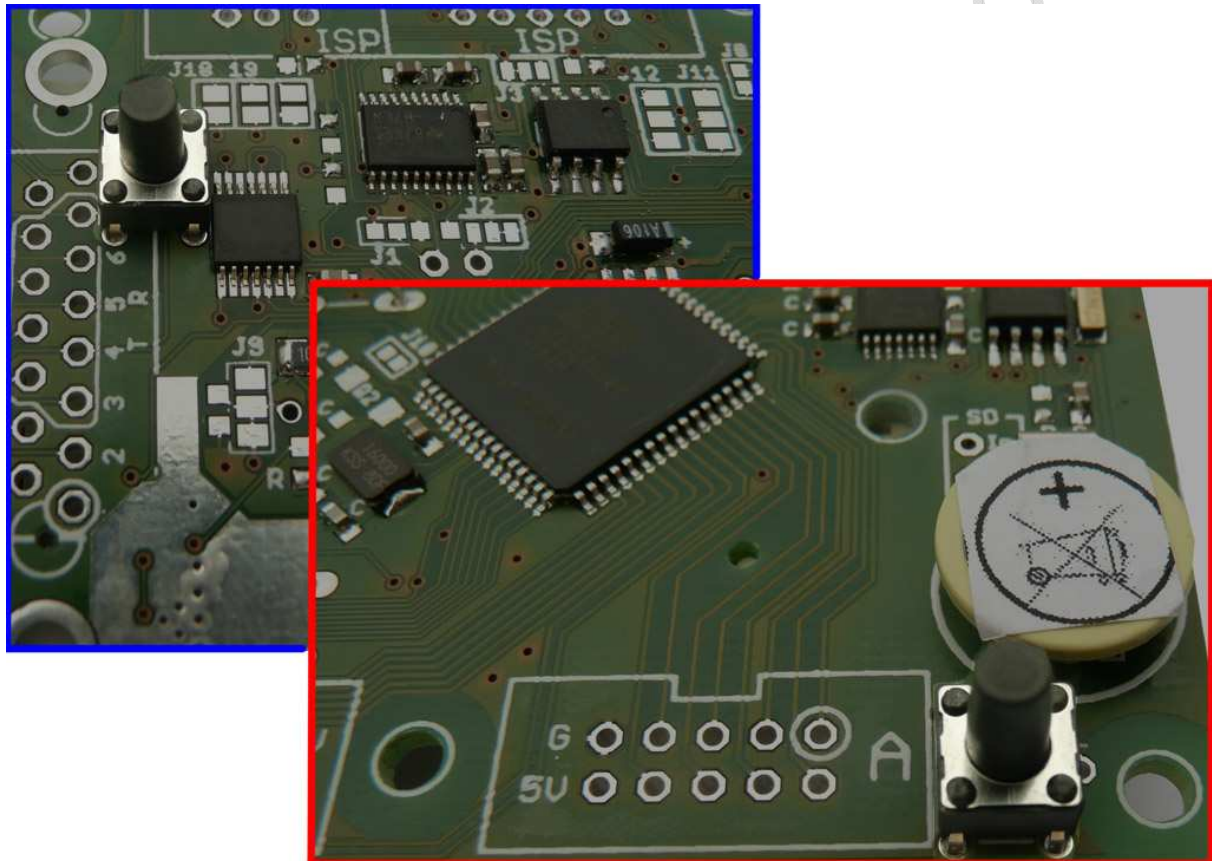
Taste	Joystick	Folientastatur (Reihe/Spalte)	Port Lieferzustand	Möglicher alternativer Port	Zuständiger Jumper
Oben	Links	Reihe B	G0	D2	TJ_A
Unten	Rechts	Reihe A	G3	G2	TJ_B
Links	Unten	Spalte S	G1	E7	TJ_D
Rechts	Oben	Spalte 3	D5	G2	TJ_C
-	Push	Spalte 2	D6	E7	TJ_F
Oben links	-	Reihe C	D7	-	-
Oben rechts	-	Reihe D	D4	-	-
Unten links	-	Spalte I	E5	G1	TJ_G
Unten rechts	-	-	E7	D6	TJ_E

**Wichtig:** Tasterjumper TJ\_B sollte umkonfiguriert werden, wenn Sie einen Uhrenquarz nutzen (evtl. auch bei Nutzung der Echtzeituhr – siehe auch Seite 31 unten).

## Reset-Taster

Das Board D071x bietet zwei mögliche Positionen für einen Reset-Taster. Einmal unten rechts neben dem Anschluss für Port A (roter Kasten). Dieser Taster kann von beiden Seiten auf die Platine gesteckt werden. Damit kann er sowohl über die Fläche des Tastenfelds, als auch von unten (z.B. durch ein Loch im Gehäuseboden) bedient werden.

Die alternative Reset-Taste kann am linken Rand eingesetzt werden. Da sich die Position unterhalb des Displays befindet ist hier natürlich nur eine Bedienung von unten möglich. Achten Sie hier bitte darauf, dass Sie die Drahtenden des Tasters nach dem Einlöten auf der anderen Seite abknipsen, da sie ansonsten evtl. das Display stören.



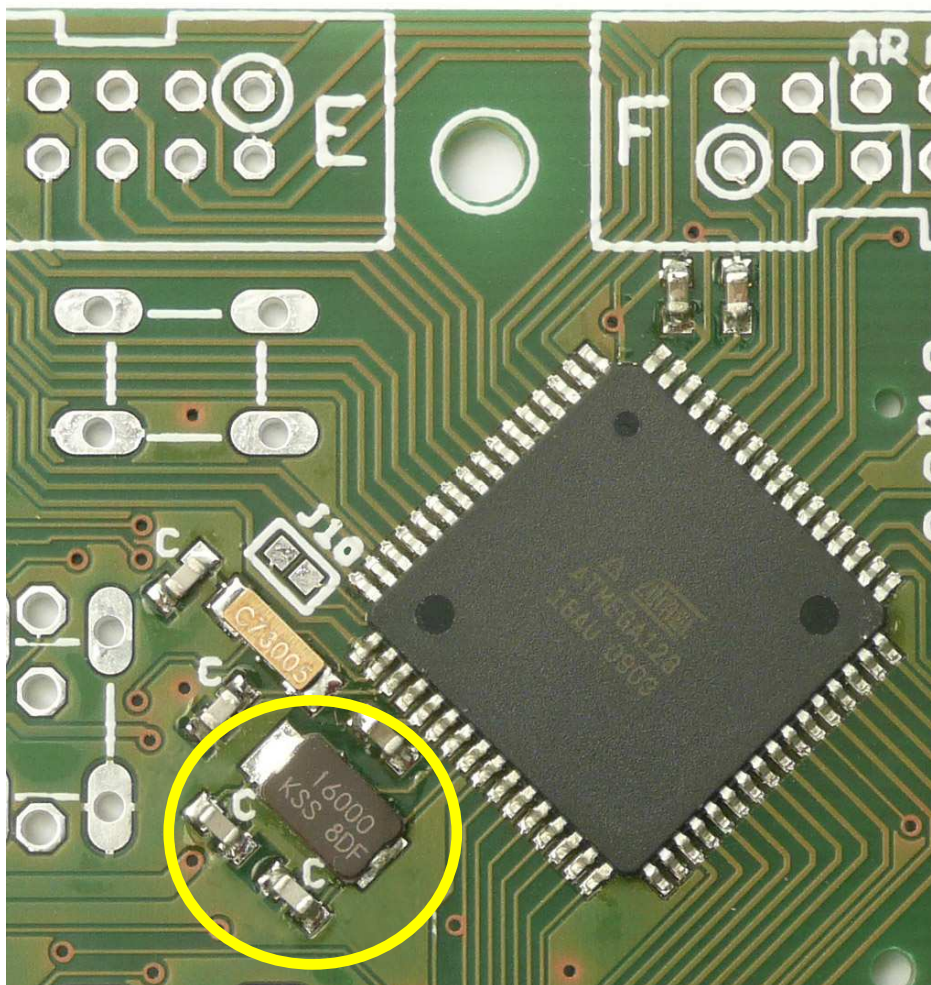
**Für die Reset-Taster-Position wie im blauen Kasten abgebildet, beachten Sie bitte den Errata-Beitrag auf Seite 92.**

## Taktfrequenz / Quarze

### Systemtakt

Der Prozessor ist aktuell auf 8 MHz Takt eingestellt (sofern Sie nicht direkt das Update auf 14,7456 MHz oder 16 MHz mitbestellt haben).

Auf Wunsch können Sie die Geschwindigkeit des Controllers erhöhen, indem Sie an die gekennzeichneten Stellen einen SMD-Quarz sowie 2x 15pF SMD-Keramikkondensatoren (Bauforn 603 einlöten (siehe Foto). Achtung: Der Platz für den Quarz ist begrenzt. Der SMD Quarz sollte eine Größe von 5x3 mm haben.



Vergessen Sie dann nicht, das entsprechende Fusebit für die Geschwindigkeit umzuprogrammieren. Lieferzustand ist **0100** (8 MHz intern). Programmieren Sie mit externem 14,7456 Mhz oder 16MHz Quarz (**und nur dann, wenn dieser eingelötet wurde**): **1111**; alle anderen Einstellungen oder ein fehlender Quarz führen u.U. zu einem nicht funktionierenden Board.

**Spielen Sie nicht grundlos mit den Einstellungen der Fuses herum – damit machen Sie u.U. Ihr Board unbrauchbar! Fragen Sie uns im Zweifelsfall vorher. Ein Anruf genügt.**

## Uhrenquarz für den internen Real Time Counter

Möchten Sie den Atmel-internen Real Time Counter (nicht zu verwechseln mit einer Real Time Clock, beides mit RTC abgekürzt) nutzen, so benötigen Sie hierfür entweder einen Uhrenquarz oder einen entsprechenden Takt. Auf unserer Platine ist ein Feld für den 32,768 KHz Quarz (Achtung: max. 4 x 1,3mm) sowie 2 Stück 15pF Kondensatoren (BF 603) vorgesehen. Der Real Time Counter ermöglicht dann durch entsprechende Dimensionierung des Teilers einen genauen internen Takt unabhängig vom Systemtakt.

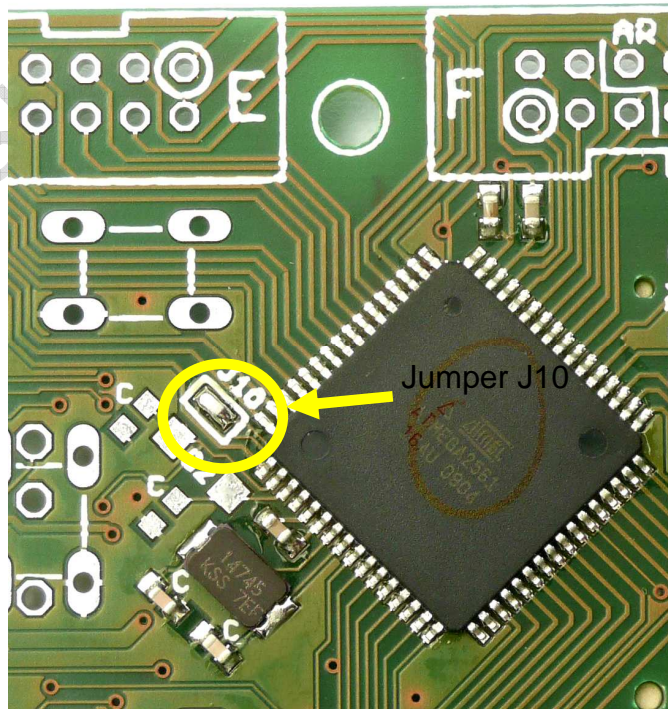
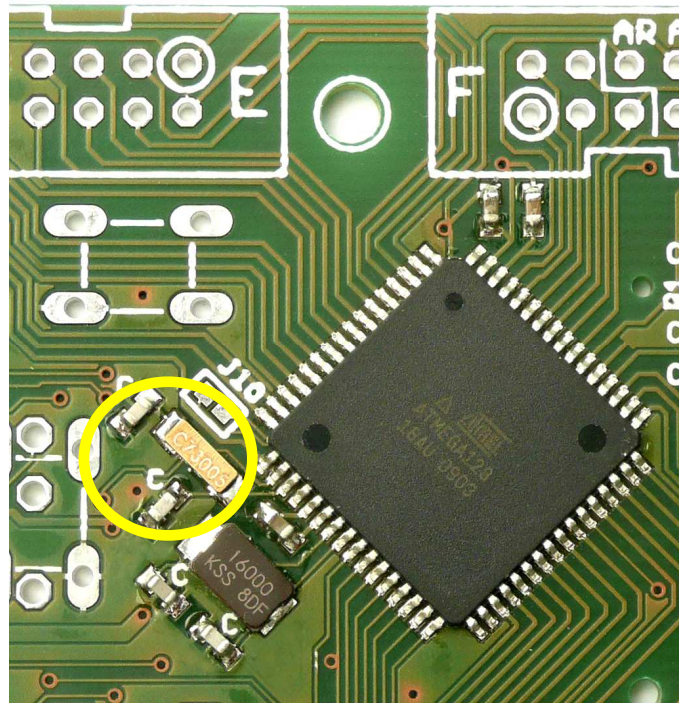
Wenn die Echtzeituhr UND einen ATmega2561 oder AT90CAN128 mitbestellt haben, so können Sie auf den

separaten Uhrenquarz verzichten: die Echtzeituhr kann einen Takt ausgeben, der am Eingang des Real Time Counters eingespeist werden kann: Dazu müssen Sie lediglich J10 mit einem Tropfen Lötzinn schließen (der Jumper ist im Lieferzustand i.d.R. offen – aber wenn Sie Glück haben, wurde er von uns während der Konfiguration Ihres Boards bereits geschlossen).

Achtung: In Ihrer Software sollten Sie dem Atmel dann noch mitteilen, dass ein externer Takt vorliegt: Bit EXCLK im Register ASSR muss auf 1 gesetzt werden (z.B. Bascom: „Set Assr.exclk“)

Der RTC des ATMega128 sollte laut Datenblatt nicht von einem externen Takt gesteuert werden und braucht daher einen eigenen Uhrenquarz wenn Sie den Real Time Counter nutzen möchten.

Wenn Sie den Uhrenquarz an TOSCI / TOSC2 nutzen möchten (müssen), dann können Sie jedoch Port G3 nicht mehr nutzen – hier ist dann eine **Umkonfigurierung** des Tasters „Unten“ von G3 auf G2 notwendig (**Taster-Jumper: TJ\_B**).



## Schnittstellen

Zur Kommunikation mit anderen Geräten stehen auf der Platine Stecker für 2 x RS232-Verbindungen, einer RS485- sowie dem CAN-Bus (nur bei Controller AT90CAN128) zur Verfügung. Als Ergänzung kommen noch I<sup>2</sup>C sowie SPI hinzu.

### **SPI Bus:**

Der SPI Bus ist ein serieller Highspeed-Bus, der auf unserem Board vom Display und der SD Karte genutzt wird. Hier gibt es 3 gemeinsame Leitungen:

- Data In
- Data Out
- Clock

sowie weitere je nach angeschlossenem Gerät notwendige Leitungen. Jeder Teilnehmer am SPI Bus hat eine individuelle CS Leitung (Chip Select), die dem Gerät mitteilt, dass die nun auf dem BUS ankommenden Signale für ihn bestimmt sind. Signale ohne aktives CS Signal werden ignoriert, da sie für einen anderen Teilnehmer bestimmt sind.

SPI erlaubt sehr hohe Geschwindigkeiten, aber i.d.R. nur sehr kurze Leitungsverbindungen.

Der verwendete Mikrocontroller nutzt ein sogenanntes Hardware-SPI, dies steigert die mögliche Datenrate. Die Datenleitungen sowie die Clock-Leitung ist bei Hardware-SPI fest vorgegeben.

Hier sind genutzt:

- Port B1: Clock-Leitung
- Port B2: Data Out (MOSI)
- Port B3: Data In (MISO)

Diese drei Ports sollten Sie (es sei denn für die Nutzung bei einer weiteren SPI-Peripherie) nicht für andere Zwecke nutzen.



## I<sup>2</sup>C / TWI – Zweidraht-Interface

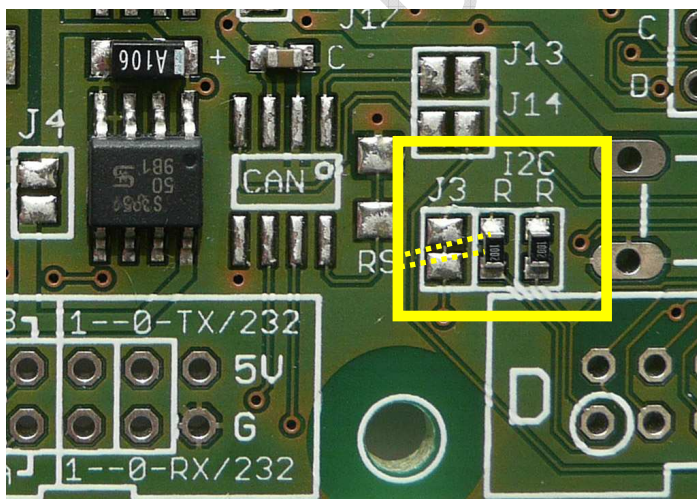
Der ATmega bietet u.a. auch ein TWI, was nichts anderes bedeutet als Two-Wire-Interface. Ein bekannter Vertreter eines TWI ist z.B. I<sup>2</sup>C (ausgesprochen: I square C) – elektrisch sind beide identisch. Lediglich aus lizenzrechtlichen Gründen (Die Rechte an I<sup>2</sup>C liegen bei Philips) benennt Atmel seine I<sup>2</sup>C Schnittstelle TWI.

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 I<sup>2</sup>C-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.

Dieses TWI (quasi identisch zu I<sup>2</sup>C) liegt bei dem vorliegenden Controllern an den Ports D0 und D1. Dies bedeutet, über die Ports D0 und D1 findet der I<sup>2</sup>C Datenverkehr statt und diese Ports können nicht für andere Aufgaben genutzt werden, solange der I<sup>2</sup>C Bus genutzt werden soll.

Der I<sup>2</sup>C Bus sieht vor, dass die beiden Leitungen des Bus von einem Mastermodul mit Pull-Up Widerständen versehen werden. Diese sind bei unseren Board standardmäßig vorhanden und aktiv! Sind bereits andere Master im Bus vorhanden, müssen die Pullup-Widerstände deaktiviert werden. Dazu brauchen Sie diese aber nicht auszulöten. Es reicht, einen vorbereiteten Jumper zu Öffnen:



Im gelb markierten Feld erkennen Sie die beiden I<sup>2</sup>C Pullup-Widerstände (je 10 KOhm) sowie Jumper J3, welcher standardmäßig geschlossen ist und die beiden Widerstände mit der 5V Versorgungsspannung verbindet.

**Solange der Jumper geschlossen ist, sind die Ports D0 und D1 für Sie u.U. nicht anderweitig nutzbar! Achtung: die Kommunikation mit der optionalen Echtzeituhr benötigt I<sup>2</sup>C.**

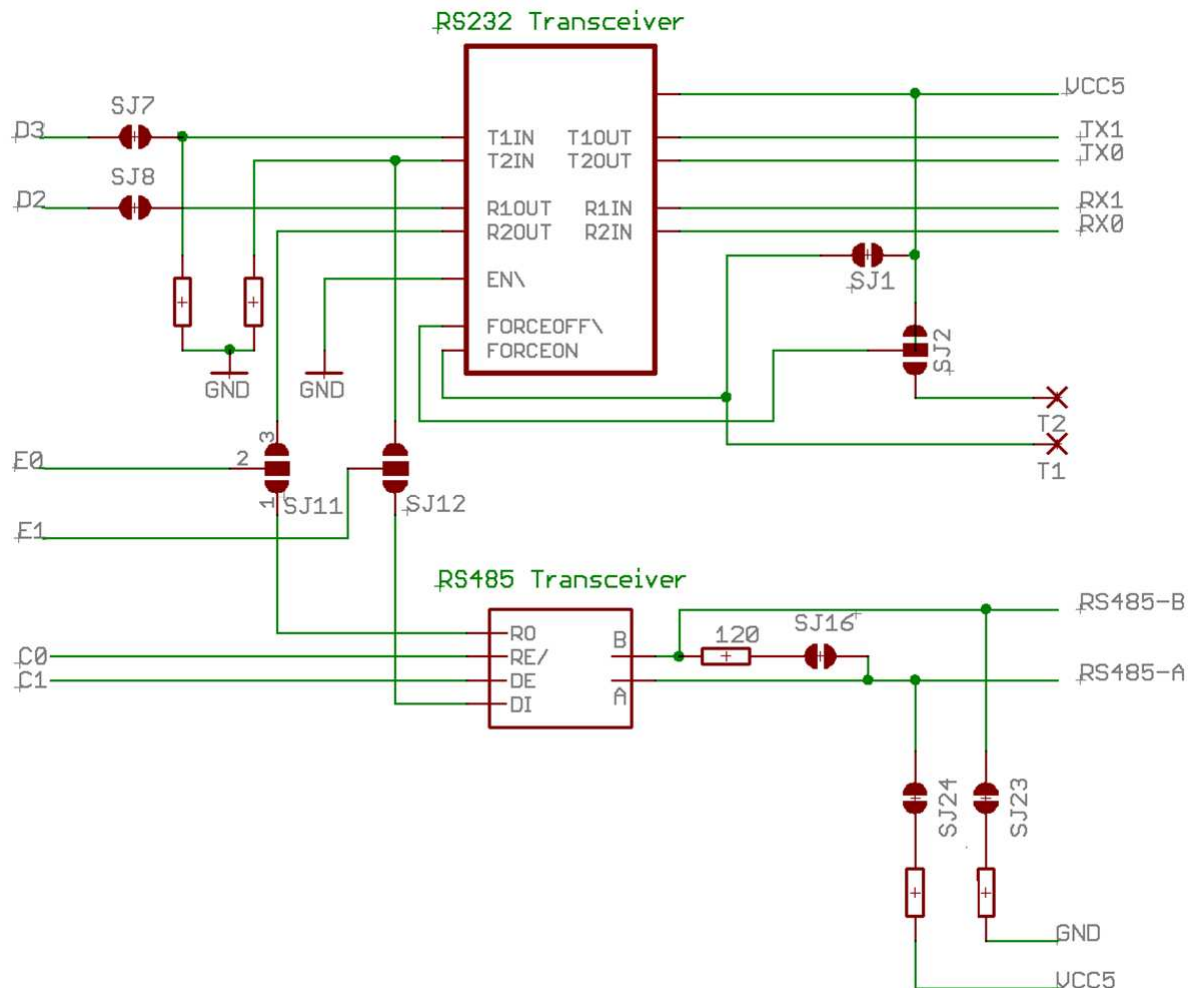
Wenn Sie den Jumper Öffnen (die dünne Leiterbahn mit einem feinen

Messer oder Skalpell durchtrennen – siehe gestrichelte Linie), sind die Ports D0 und D1 nicht mehr über die Widerstände mit 5V verbunden. Aber Achtung: D0 und D1 sind nun noch über 2x10 KOhm miteinander verbunden. Wenn dies stören sollte, müssen Sie noch einen der beiden Widerstände auslöten.

## Serielle Schnittstellen

Die ATmega 128 / 2561 und der AT90CAN128 bieten zwei serielle Schnittstellen: UART0 und UART1, deren Zuweisung von Ihnen konfiguriert werden kann.

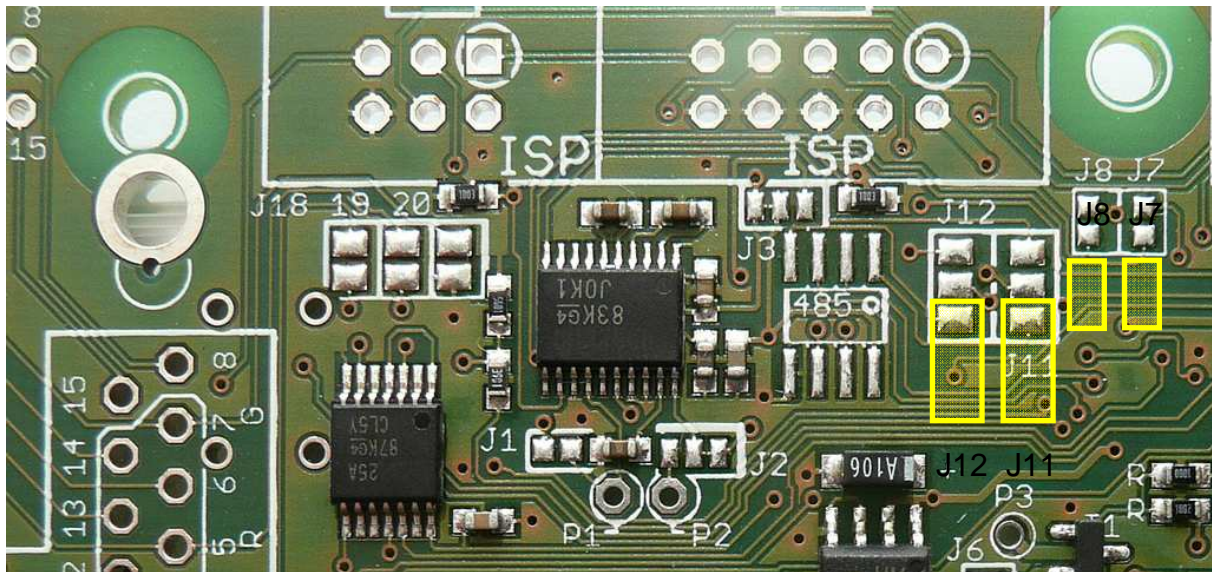
Der Anschluss der RS232 und RS485 Schnittstelle wurde wie folgt realisiert:



## RS 232-Interface

**Wichtige Vorabinfo:** Wenn Sie die RS232 Schnittstelle (UART 1) sofort nutzen möchten, müssen Sie zuvor die Jumper J7 und J8 schließen. Möchten Sie einen SubD Stecker in den vorgesehenen Platz einlöten und nutzen, müssen Sie zuvor noch die Jumper J18, J19, J20 schließen.

Die Pins für die serielle Schnittstelle 0 des ATmega-Controllers werden auch für das ISP-Programmierinterface genutzt – beide kämen sich bei einer gleichzeitigen Nutzung in die Quere, das der RS232-Pegelwandler die Receive-Leitung ohne Signal permanent auf High zieht. Daher sind die Jumper für die serielle Schnittstelle 0 (Port E0 und E1) des ATmegas standardmäßig offen und mit keinem Pegelwandler verbunden (also nur mit dem ISP-Stecker verbunden). Durch Schließen des Jumpers können Sie die UART0 an den RS232 oder den RS485 Pegelwandler legen. Der RS232 Pegelwandler ist über 2 Jumper mit UART1 verbunden (D2 und D3). Auch diese Jumper (J7 und J8) müssen Sie zuerst schließen – im Lieferzustand sind sie offen.



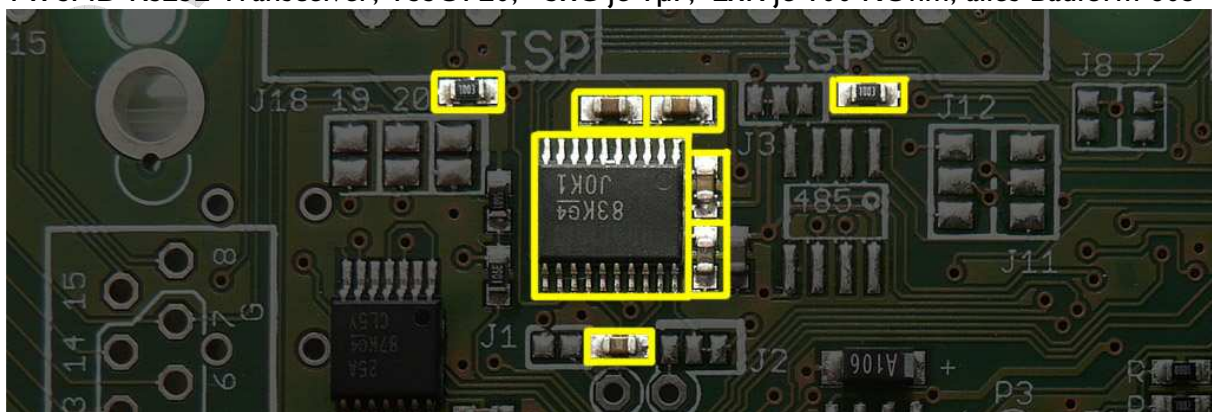
Da für die **UART0** eine gleichzeitige Beschaltung von RS232 und RS485 Pegelwandler nicht möglich ist, wird die jeweilige Auswahl mittels **J11 und J12** festgelegt – hier legen Sie also fest, mit welchem der beiden Bausteine die Ports E0 und E1 verbunden werden. Standardmäßig sind alle Jumper offen. Möchten Sie die Auswahl auf RS232 oder RS485 festlegen, so schließen Sie bitte jeweils das mittlere Feld mit dem unteren (RS485) bzw. das mittlere Feld mit dem oberen Feld (RS232).

Die serielle Schnittstelle I (**UART1**) steht durch Schließen der Jumper **J7 und J8** als High-Pegel an den Pads RxI und TxI zur Verfügung.

Wenn Sie kein High-Pegelinterface brauchen, sondern z.B. lediglich 2 Controllermodule über die Schnittstelle 0 miteinander verbinden möchten, so können Sie dies natürlich ohne Änderung der Jumperfelder durchführen (die Verbindung muss dann lediglich für eine evtl. Neuprogrammierung über den ISP-Programmieradapter getrennt werden).

Das Board D071x wird standardmäßig ohne RS232 Transceiver geliefert. Dieses ist eine Zusatzoption und muss von Ihnen mitbestellt werden. Sofern Sie den Transceiver nachrüsten möchten, benötigen Sie dazu (genaue Bauteilbezeichnungen siehe Schaltplan auf Ihrer CD):

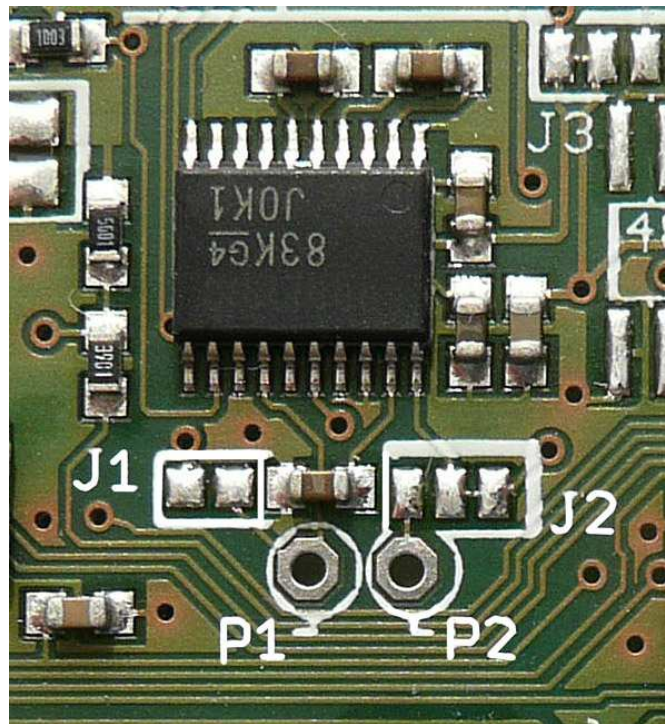
1 x SMD RS232 Transceiver, TSSOP20, 5xC je 1µF; 2xR je 100 KOhm, alles Bauform 603



Unterhalb des RS232 Pegelwandlers erkennen Sie zwei Jumper J1 / J2 sowie 2 Löt pads, die mit P1 und P2 gekennzeichnet sind.

Diese Felder wurden zu einer optionalen Konfigurierung des RS232 Transceivers vorgesehen. Im Lieferzustand des D071x ist der Pegelwandler in der Lage, selbstständig den Shutdown-Modus einzuleiten (also Energie zu sparen), wenn er nicht benötigt wird (z.B. wenn kein weiteres RS232 System angeschlossen ist).

Mit den Jumpfern 1 und 2 sowie den Feldern P1 und P2 (die Sie durch Einlöten von Leitungen nach Belieben mit gewünschten Ports verbinden können) können Sie das Shutdown-Verhalten des

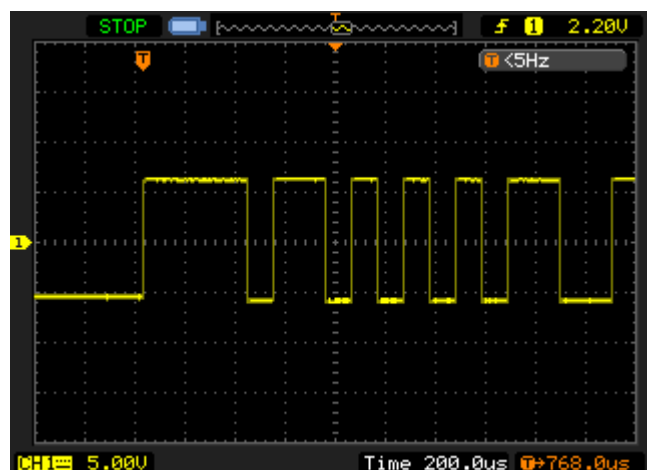


Transceivers beeinflussen. Bitte konsultieren Sie zu den verschiedenen Optionen das Datenblatt des Transceivers auf unserer CD sowie auch den D071x Schaltplan auf der CD.

**Zu beachten:** Der automatische Shutdown des Transceivers sorgt dafür, dass Sie ohne Anschluss eines Gegengerätes kein Ausgangssignal an Tx messen können – das Signal beträgt dann 0 Volt, Wenn sie trotzdem (also ohne Anschluss einer Gegenstelle) das Ausgangssignal von Tx1 messen möchten, so legen Sie an Rx1 einfach 5V an. Erst dann erwacht der RS232 Transceiver aus seinem Auto-Shutdown zum Leben.

Unser Board gibt auf der seriellen High-Pegel Schnittstelle am Signal Tx ein Signal von ca. -7V und +7V aus (siehe nebenstehendes Oszilloskop-Bild) – allerdings nur, wenn eine Gegenstelle angeschlossen wurde – ohne vorhandene Gegenstelle messen Sie hier nichts).

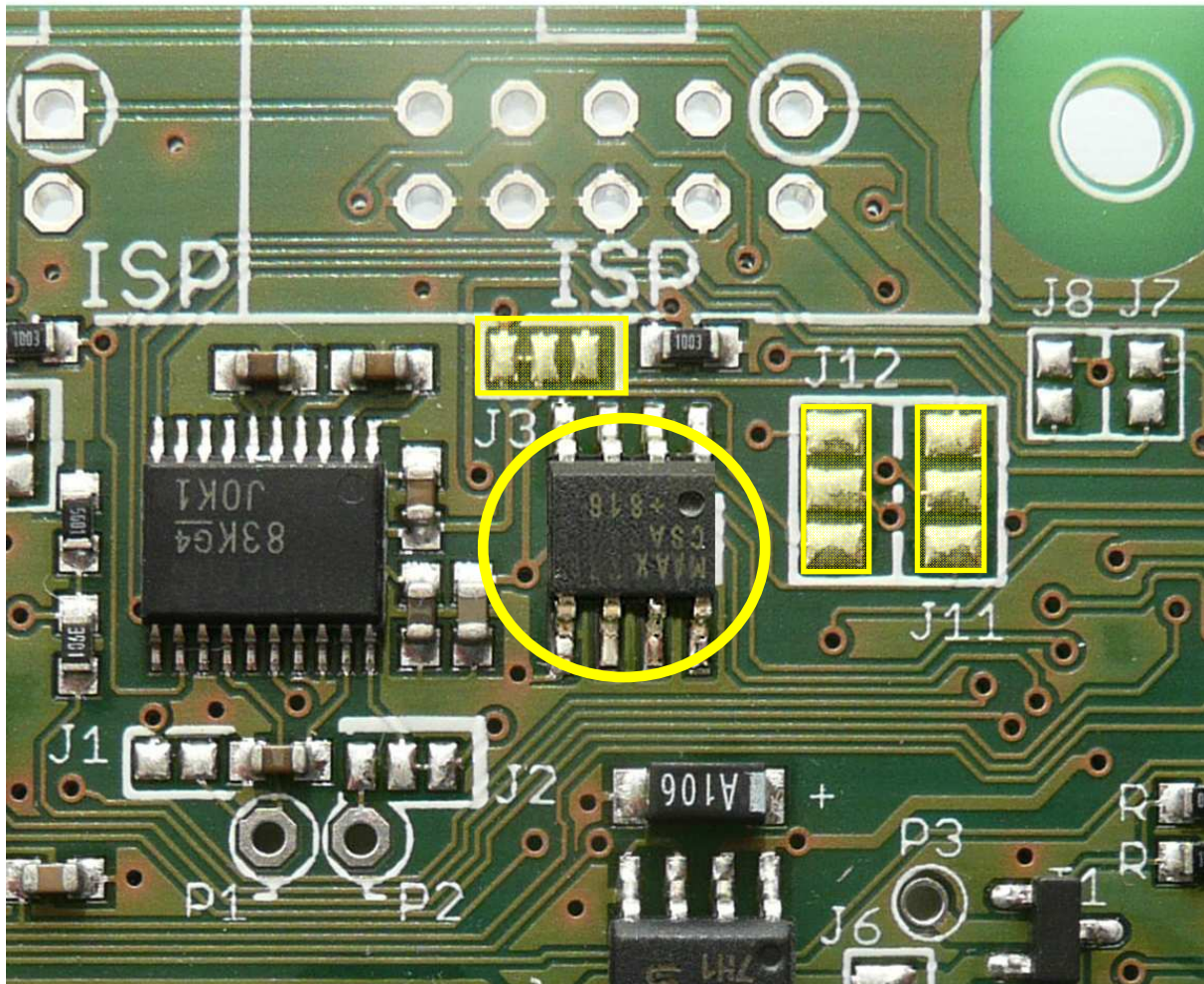
Auch wenn üblicherweise (z.B. PC) Signale von +12V und -12V genutzt werden, liegen die Signale unseres Boards noch innerhalb der RS232 Spezifikation (siehe ab Seite 75 ff.) und werden problemlos von der Gegenstelle erkannt. Selbstverständlich kann das D071x höhere Spannungen als 5V als Eingangssignal an Rx verwerten (aber nur an Rx0 / Rx1 !! – an den normalen Ports des Mikrocontrollers dürfen nur max. Spannungen in Höhe der Controller-Spannung anlegen (i.d.R. also +5 Volt / GND).



Anmerkung: Auf Seite 75 ff. erfahren Sie mehr generelle Details über RS-232 und RS485.

## RS-485 Interface

Das D071x-Board wird standardmäßig ohne RS485-Transceiver geliefert. Sie müssen diesen bei Bedarf nachrüsten, wenn sie das Board nicht direkt mit dieser Option bestellen. Dazu löten Sie an die nachfolgend gezeigte Position (I) z.B. einen Max485 Chip (S08) ein.



Rechts neben dem RS485 Transceiver erkennen Sie zwei Jumper-Felder (J11 und J12). Damit hat es folgende Bewandnis: Der ATmega nutzt für die Kommunikation per RS232 oder RS485 die gleichen Ports (hier Rx0 und Tx0 – E.0 und E.1) – also die serielle Schnittstelle 0.

Da eine gleichzeitige Beschaltung von RS232 und RS485 Pegelwandler nicht möglich ist, wird die jeweilige Auswahl mittels J11 und J12 festgelegt – hier legen Sie also fest, mit welchem der beiden Bausteine die Ports E0 und E1 verbunden werden. Standardmäßig sind die Jumper offen. Möchten Sie die Auswahl auf RS232 oder RS485 festlegen, so schließen Sie bitte jeweils das mittlere Feld mit dem unteren (RS485) bzw. das mittlere Feld mit dem oberen Feld (RS232).

Der RS485 Transceiver ist ein Halbduplex-Chip. Um beim Halbduplexbetrieb die korrekte Umschaltung zwischen Senden und Empfangen durchzuführen, können Sie an diesem Chip die Sende- und Empfangseinheit getrennt ein- und ausschalten:

RE (Receive enable) des Max485 ist mit C.0 des Mikrocontrollers verbunden;

DE (transfer enable) mit C.1 des Mikrocontrollers verbunden.

### RS485 Betrieb bei weniger als 5 Volt

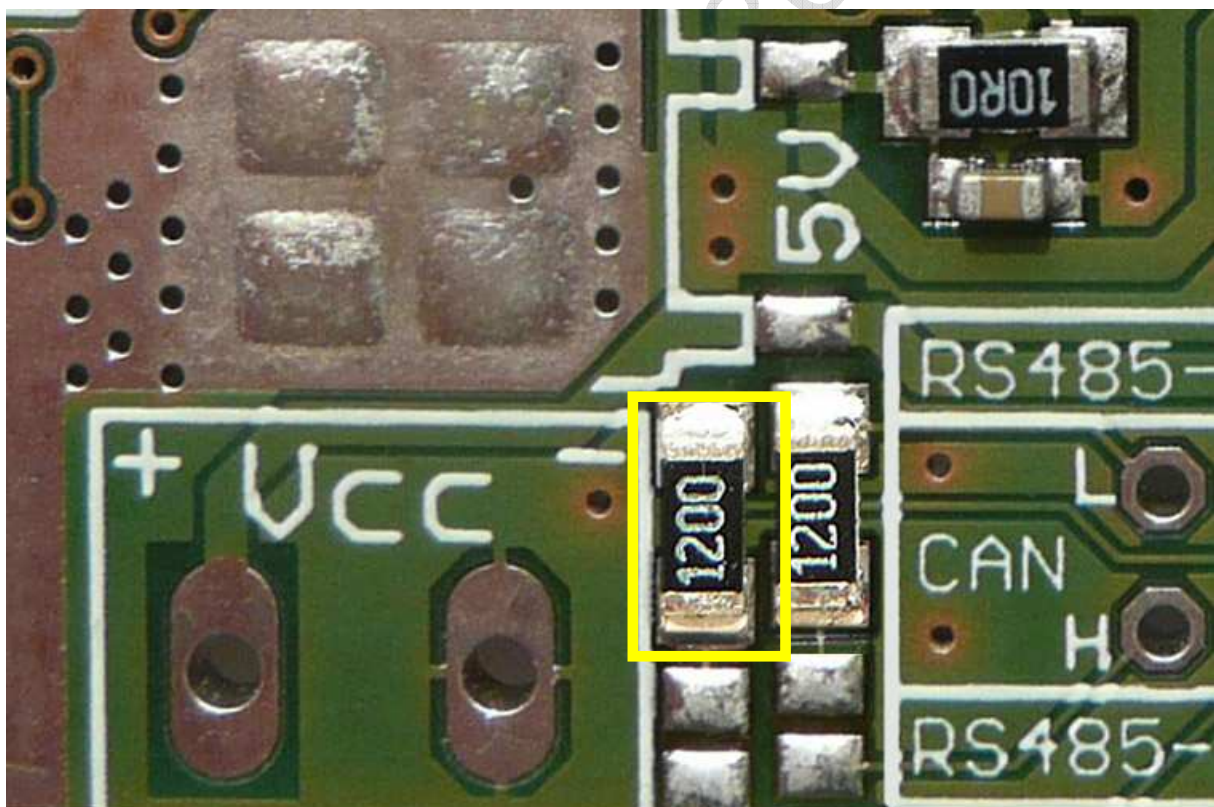
Wenn Sie vorhaben, auch das D071x als batteriebetriebenes Gerät mit einer Spannung von weniger als 4,5 Volt zu betreiben, dann ist evtl. die 5 Volt-Variante des RS485 Transceivers nicht geeignet. Alternativ können Sie die Platine auch mit einem (leider teuren) Typen bestücken lassen, der von 3 bis 5 Volt betrieben werden kann.

### Abschaltung der Spannungsversorgung des RS485 Transceivers

Links oberhalb des Transceiver-Chips erkennen Sie einen weiteren Jumper J3. Dieser erlaubt das Wechseln der Versorgungsspannung des RS485 Transceivers von Dauer-5 Volt (Lieferzustand) auf die optionale abschaltbare 5-Volt Spannungsversorgung. Dazu trennen Sie die Verbindung zwischen dem linken und dem mittleren Feld auf und verbinden mit einem Tropfen Lötzinn das mittlere Feld mit dem rechten Feld.

### RS485-Abschlusswiderstand

Bei RS485 kann es für die Endgeräte am Bus notwendig sein, einen 120 Ohm Bus-Abschlusswiderstand einzuschalten. Auf dem folgenden Foto erkennen Sie die Position des optionalen Widerstands. Mit der unter dem Widerstand liegenden Lötbrücke (Lieferzustand offen) schalten Sie den Widerstand auf den Bus auf, bzw. trennen ihn ab.



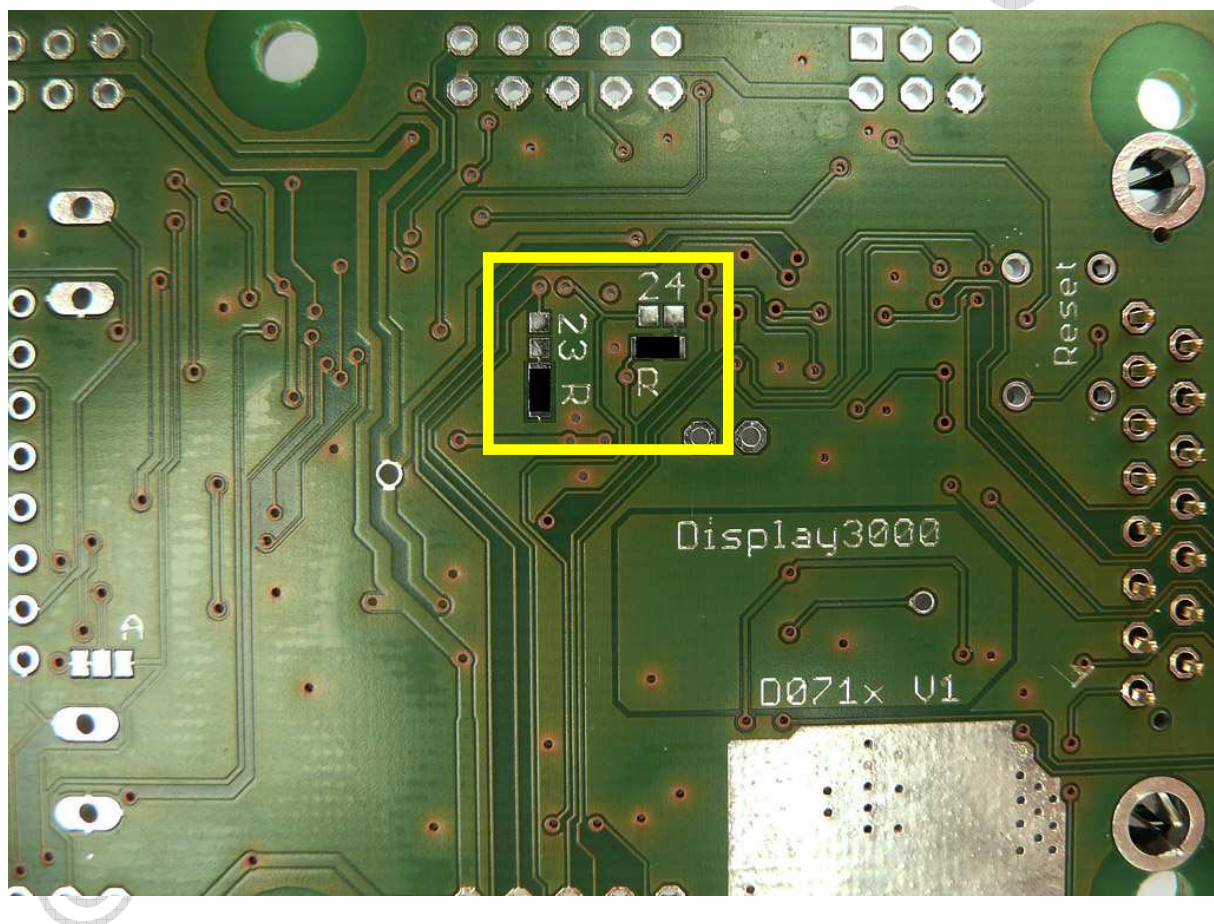
Auf dem Foto sind 2 Widerstände zu erkennen. Der linke Widerstand ist für den RS485-Bus; der rechte Widerstand ist für den CAN-Bus vorbereitet. Unterhalb der jeweiligen Widerstandsfelder liegt der zuständige Jumper. Durch Schließen des Jumpers (Lieferzustand: offen) wird der Widerstand auf den Bus geschaltet.

## RS485 Pullup / Pulldown

Eventuell (vermutlich eher selten) kann es notwendig sein, dass das D071x das RS485 Masterboard werden soll. Dann ist es notwendig, die RS485 Leitung mit einem Pullup-Widerstand (Leitung A) bzw. einem Pulldown-Widerstand (Leitung B) versehen zu müssen.

Auch dies kann mit dem D071x Board realisiert werden. Hierzu gibt es zwei, unterhalb des Displays versteckte Jumper sowie Plätze für die notwendigen Widerstände. Jumper 23 schalten den Pulldown-Widerstand gegen Masse und Jumper 24 schaltet den Pullup-Widerstand an 5V. Diese Widerstände sind nicht im Lieferumfang enthalten und müssen von Ihnen nachgelötet werden.

Dazu entfernen Sie vorher das Display wie im Kapitel Display beschrieben.



Anmerkung: Auf Seite 75 ff. erfahren Sie mehr generelle Details über RS-232 und RS485.

Gute weiterführende Informationen zu RS485 erhalten Sie hier:

[http://www.interfacebus.com/Design\\_Connector\\_RS485.html](http://www.interfacebus.com/Design_Connector_RS485.html)

[http://www.embeddedsys.com/subpages/resources/images/documents/microsys\\_art\\_RS485.pdf](http://www.embeddedsys.com/subpages/resources/images/documents/microsys_art_RS485.pdf)

## CAN-Bus Interface

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

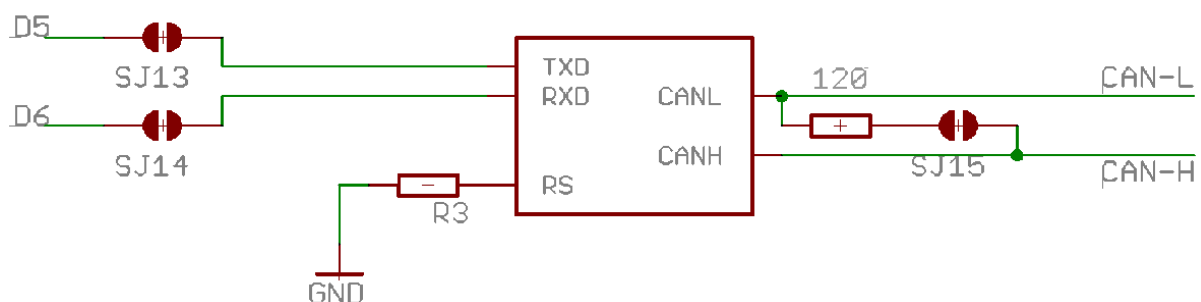
Mit einem CAN-Bus lassen sich quasi beliebig viele Module miteinander unkompliziert vernetzen. Die Länge der möglichen Verbindungsleitung ist abhängig von der Geschwindigkeit der Datenübertragung. Die maximale theoretische Leitungslänge beträgt bei

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.

Um Missverständnisse zu vermeiden: Der CAN Transceiver ist nur dann für Sie nutzbar, wenn Sie einen AT90CAN128 Controller im System bestellt haben. Ansonsten können Sie dieses Kapitel überspringen.

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 NXP PCA82C250 (5 Volt) – (ab Herbst 2011 den Nachfolger TJA1050 – siehe blauen Text auf der nächsten Seite).



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

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! Da er in der Regel gebrückt wird, ist dieses Feld von uns mit einer Leiterbahn geschlossen worden. Wenn Sie einen Widerstand einlöten möchten, durchtrennen Sie bitte vorher die Leiterbahn zwischen den beiden Löt pads für den Widerstand R3 (liegt rechts neben dem CAN Transceiver). Sie müssen je nach Geschwindigkeit den entsprechenden Widerstand einlöten. Das Feld für den Widerstand RS liegt direkt rechts neben dem CAN Transceiver.

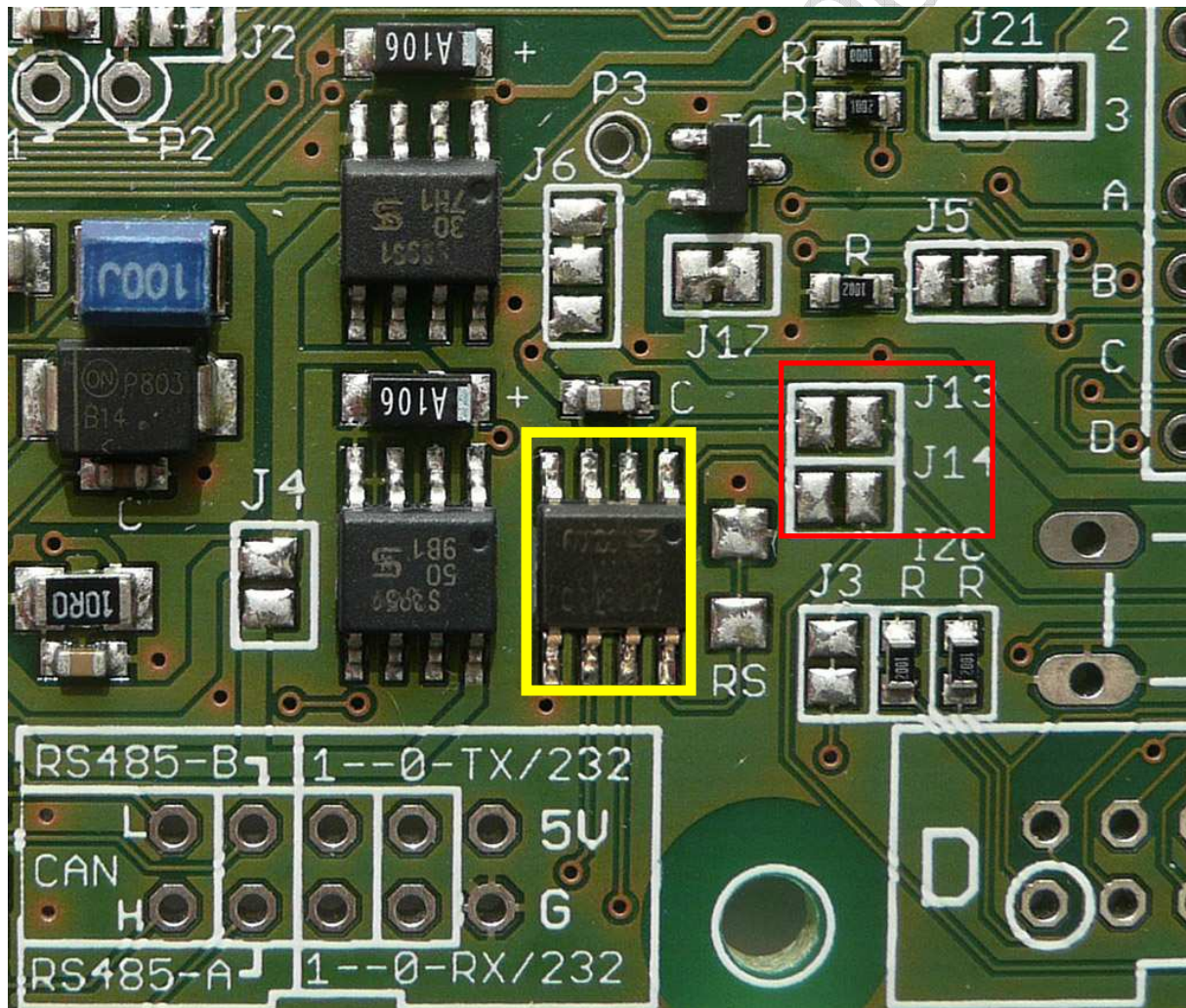


Anhaltswerte für den Widerstand RS:

Highspeed: 0 Ohm – also brücken (Lieferzustand)  
 1MBit/s 1 KOhm  
 100kBit/s: 47 KOhm

**Anmerkung: ab ca. Herbst 2011 wechseln wir den Transceiver auf den pinkompatiblen Nachfolgetyp TJA1050 ebenfalls vom Hersteller NXP. Dort ist der Widerstand R3 (RS) nicht mehr notwendig. Die Leitung liegt dann gegen Masse.**

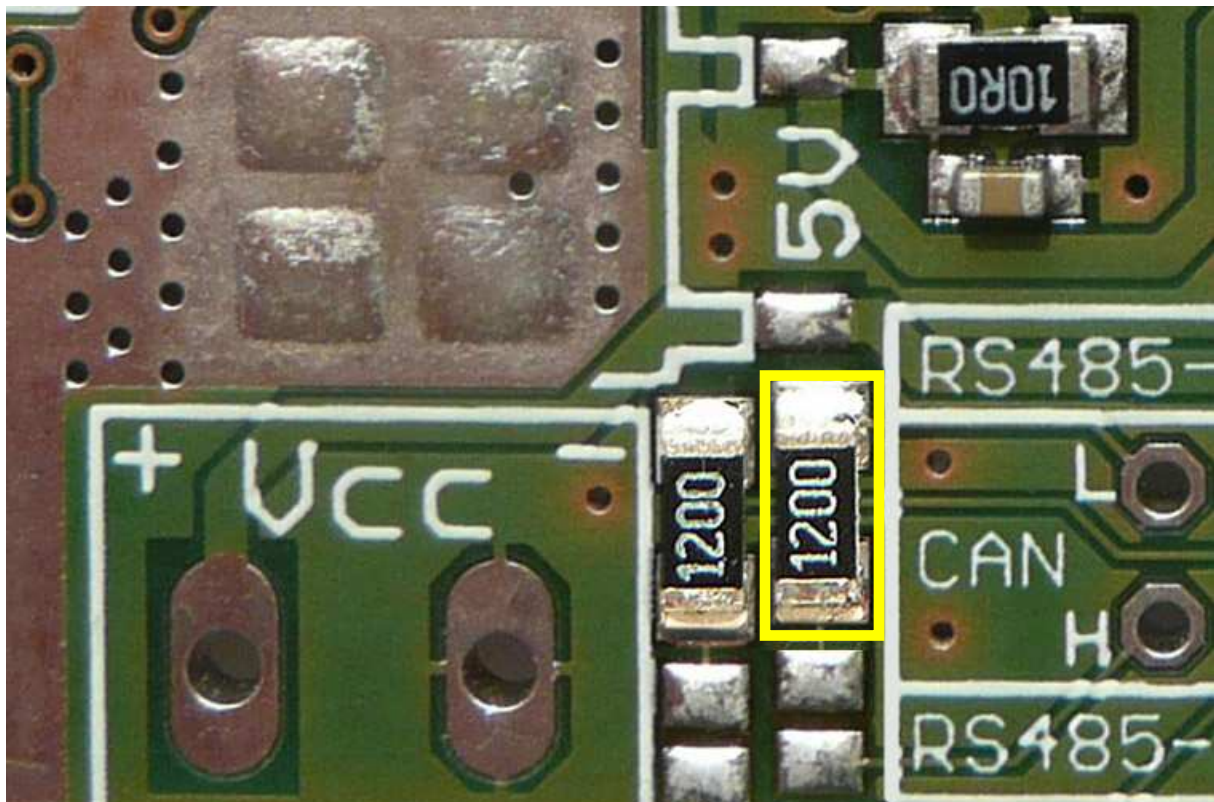
Der CAN Transceiver ist mit den Ports D5 und D6 des Mikrocontrollers verbunden. Diese sind beim AT90CAN128 als die CAN Ports definiert. Die beiden Brücken J13 und J14 sind noch geöffnet, d.h. der CAN Transceiver ist NICHT mit dem Mikrocontroller verbunden. Sie müssen diese beiden Brücken J13 und J14 jeweils noch mittels etwas Lötzinn schließen (im Foto unten rot gekennzeichnet).



Der CAN Transceiver ist standardmäßig mit der VCC5\_2 Leitung verbunden, d.h. er kann, wenn die Abschaltoption mitbestellt wurde, vom Mikrocontroller zur Energieeinsparung komplett abgeschaltet werden.

### CAN-Abschlusswiderstand

Beim CAN-Bus kann es für die Endgeräte am Bus notwendig sein, einen 120 Ohm Bus-Abschlusswiderstand (d.h. auf beiden Seiten des Bus notwendig) einzuschalten. Auf dem folgenden Foto erkennen Sie die Position des optionalen Widerstands. Mit der unter dem Widerstand liegenden Lötbrücke (Lieferzustand offen) schalten Sie den Widerstand auf den Bus auf, bzw. trennen ihn ab.



Auf dem Foto sind 2 Widerstände zu erkennen. Der linke Widerstand ist für den RS485-Bus; der rechte Widerstand ist für den CAN-Bus vorbereitet. Unterhalb der jeweiligen Widerstandsfelder liegt der zuständige Jumper. Durch Schließen des Jumpers (Lieferzustand: offen) wird der Widerstand auf den Bus geschaltet.

## Analoge Eingänge (Analog-Digital-Wandler)

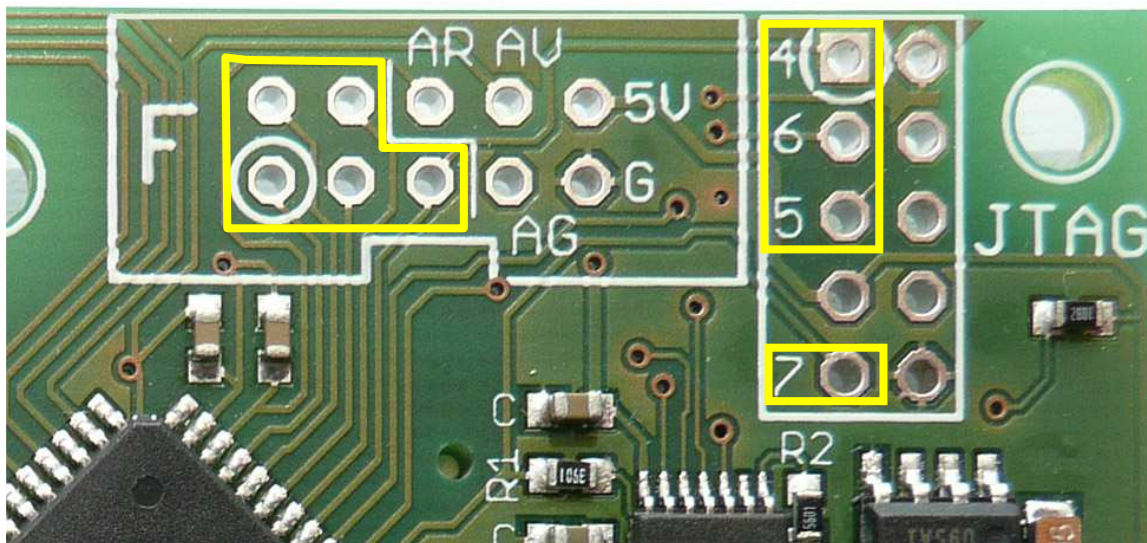
Einige Eingänge des Prozessors können analoge Werte mit einer Auflösung von bis zu 10 Bit auswerten (Port F0 bis F7). 10 bit Auflösung bedeutet, das z.B. ein Signal von 0-5 Volt mit einer Auflösung von 1024 Stufen ( $5 / 1024 = \text{ca. } 5\text{mV}$ ) ausgewertet werden kann (also 0,1%).

Wer sehr exakte Werte braucht, der findet am ATmega128 noch drei weitere Eingänge für die Spannungsversorgung der Wandler sowie der Referenzspannung: AGnd, AVcc, ARef.

Wenn der Analogwandler mit keine hochgenaue externe Versorgung benötigt, reicht es, AGND mit GND zu verbinden und AVCC mit VCC (5 Volt Versorgung). Die Anschlüsse für ARef, AGND und AVCC liegen am Stecker für Port F an.

**Am Portstecker für Port F liegen nur die Ports F0 bis F4 an.** Die Ports für F4 bis F7 liegen am daneben liegenden JTAG Stecker (F4 ist also doppelt vorhanden), da JTAG die Ports F4 bis F7 sowieso benötigt.

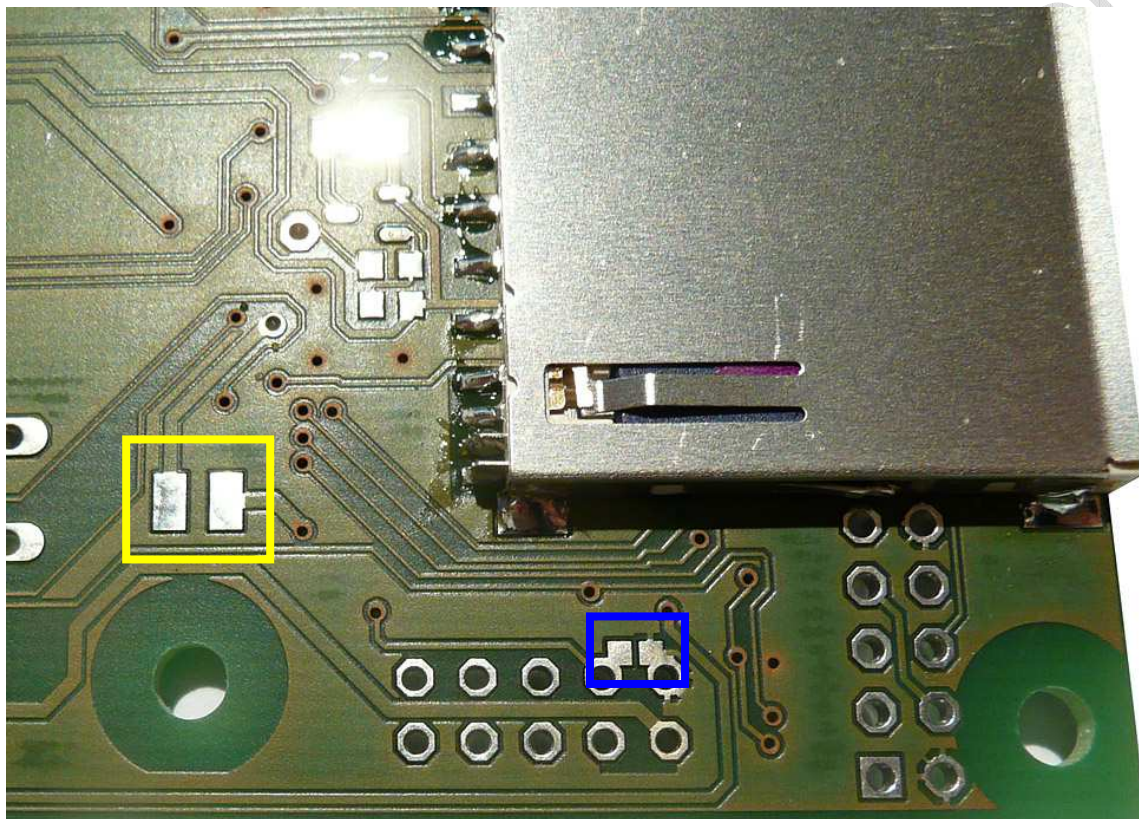
Die nun nicht von F5, F6, F7 belegten Kontakte am Portstecker F wurden von uns mit den Anschlüssen für ARef, AGND und AVcc belegt (mit AR, AB und AV beschriftet).



**AVcc und AGND sind im Lieferzustand NICHT an 5V und GND angeschlossen.** Wenn Sie analoge Werte messen möchten, so müssen Sie hier entweder die notwendige Spannung für den Analog-Digitalwandler anschließen oder diese Pads mit 5V bzw. GND verbinden.

Hierfür haben wir auf der **Rückseite** der Platine (Displayseite) zwei Felder eingefügt:

1. Eine Lötbrücke zum Verbinden von AGND mit GND. Dies geschieht mit Hilfe der blau markierten Brücke. Bitte schließen Sie diese mit einem Tropfen Lötzinn wenn Sie AGND mit GND verbinden möchten.
2. Ein Feld zum Verbinden von AVcc mit Vcc (gelb markiert). Dieses Feld ist etwas größer, da es auch das Auflöten einer SMD Induktivität erlaubt (dieser Tiefpass sorgt für eine sauberere Spannung an AVcc und somit für genauere Messergebnisse). Da hier nur ein geringer Strom fließt, kann hier eine entsprechend flach aufbauende Induktivität (z.B. 10  $\mu$ H, BF 805; Farnell Artikel Nr. 1669563) verbaut werden. Wenn keine Induktivität notwendig oder zur Hand ist, kann das Feld auch einfach nur gebrückt werden.



ARef ist der Eingang für die Referenzspannung des internen Analog-Digitalwandlers. Für die Messung kann der Mikrocontroller entweder interne Referenzspannungen zur Grundlage nehmen oder aber eine externe Spannung. ARef kann entweder mit der normalen Versorgungsspannung verbunden sein (dann verbinden Sie ARef einfach mit Vcc oder AVcc – der Nachteil ist ein höheres Rauschen bei den Messergebnis) oder auch entsprechend gefiltert werden.

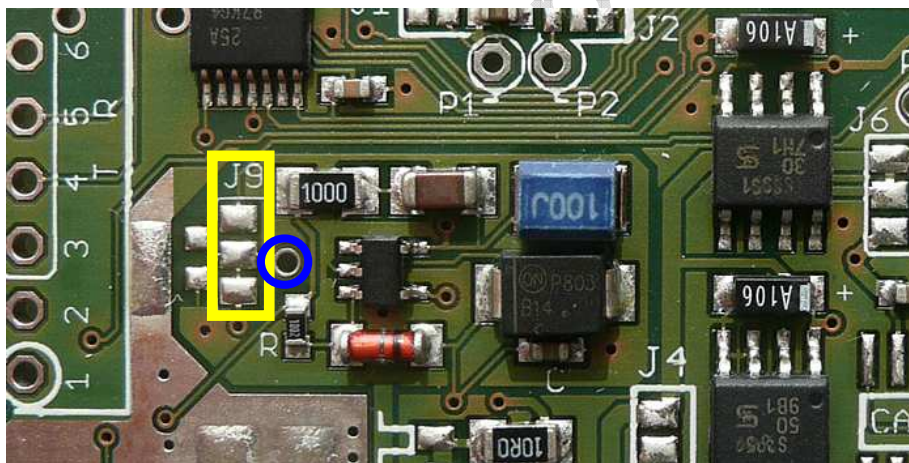
## Schalten der Beleuchtung durch den Mikrocontroller

Da die Lebensdauer der Displaybeleuchtung endlich ist (ca. 5000 Betriebsstunden bis zum Abfall der Ausgangshelligkeit auf 70%) und sich diese zudem mit über 50% am Energiebedarf des Moduls verantwortlich zeigt, macht es Sinn, die Beleuchtung bei Nichtgebrauch abzuschalten oder herunterzuregulieren. Praktischerweise kann auch der Mikrocontroller die Beleuchtung steuern (z.B. auf 50% dimmen, wenn über 1 Minute keine Bedienung mehr stattgefunden hat, nach weiteren 4 Minuten ohne Bedienung wird die Beleuchtung dann komplett abgeschaltet – sobald eine Taste betätigt wird, schaltet die Beleuchtung sich wieder ein).

Der Booster auf unserem Board kümmert sich selbstständig um die Spannungserzeugung für die Displaybeleuchtung im Bereich der Eingangsspannung von 2,5 bis 5 Volt. Standardmäßig ist das Modul auf Dauerlicht geschaltet. Über einen Lötjumper kann dies jedoch geändert werden. Öffnen Sie die Default-Verbindung zwischen dem mittleren und dem unteren Feld und verbinden dann das mittlere mit dem oberen Feld.

### Schalten der Displaybeleuchtung

In Ihrer Software müssen Sie nun Port B.7 als Ausgang definieren. Wird B7 auf High geschaltet, leuchtet das Display, ist B7 auf Low, wird die Beleuchtung abgeschaltet.. Der Port B7 ist nun natürlich nicht mehr anderweitig von Ihnen zu nutzen – falls Sie B7 jedoch benötigen, können Sie an dem blau markierten Lötpad eine Leitung zu einem beliebigen anderen Port legen – das mittlere Feld von J9 darf dann weder nach oben oder unten geschlossen sein.



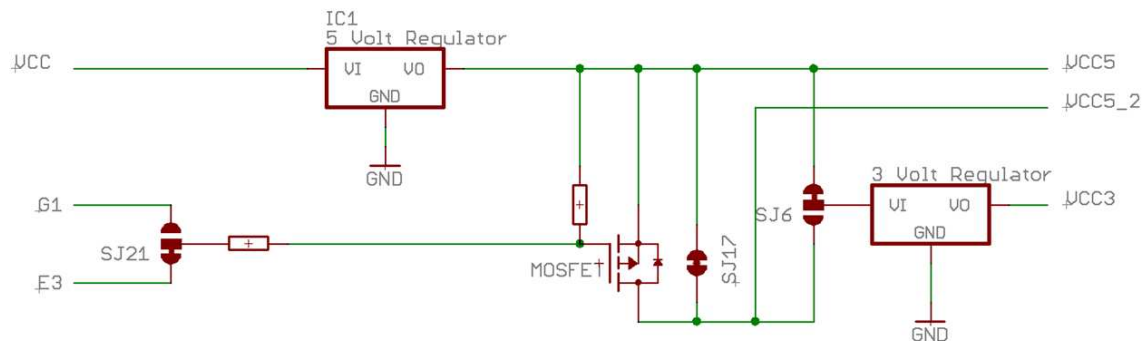
Beachten Sie: in unseren Beispielprogrammen wird Port B auch für die Displayansteuerung genutzt. Port B.7 war bislang ungenutzt und muss, damit es nun als Ausgang genutzt werden kann, auch entsprechend definiert werden.

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.

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. Der Mikrocontroller kann dies quasi nebenher automatisch erledigen, ohne dass hierzu Prozessorzeit verschwendet wird. Mehr zu PWM und der Nutzung finden Sie im Anhang: Kapitel *Dimmen der Displaybeleuchtung mittels PWM* auf Seite 80.

## Abschaltung von Verbrauchern

Als zusätzliche Option können Sie, gesteuert durch den Mikrocontroller, mittels eines MOSFETs ausgewählte Verbraucher von der Spannungsversorgung abklemmen.



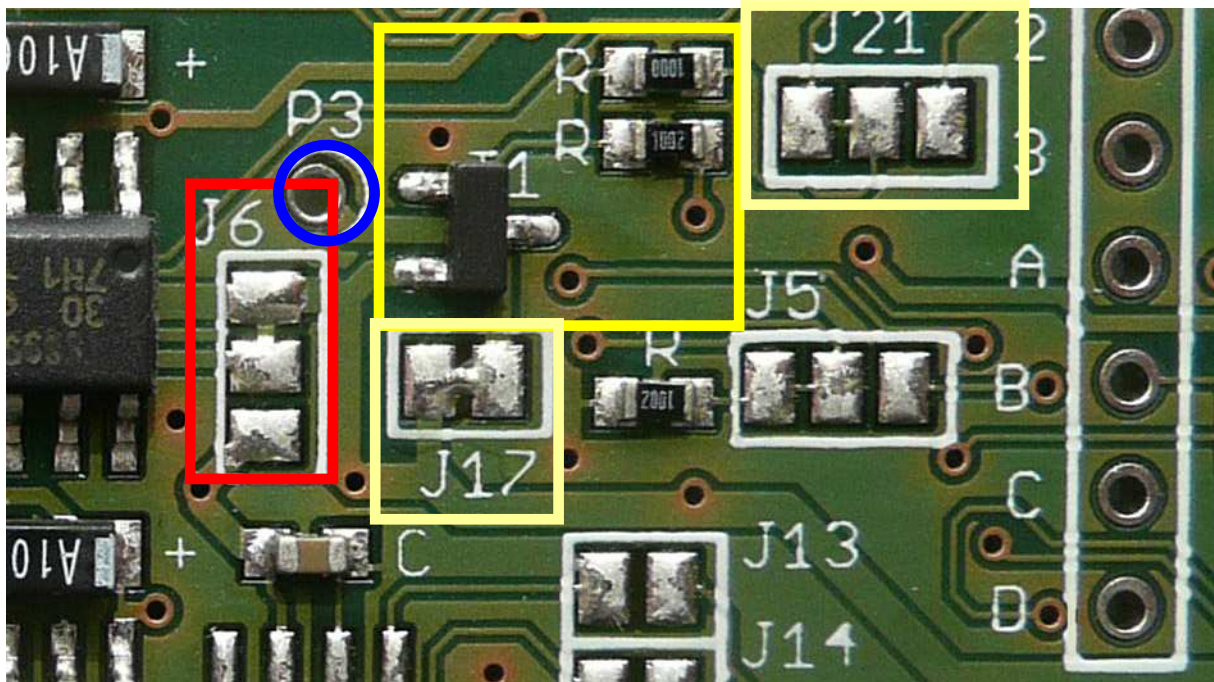
Dieser MOSFET schaltet die generelle Spannungsversorgung (im Schaltplan VCC5 genannt) für bestimmte Bauteile auf einer 2. Ebene ab (dort VCC5\_2 genannt).

Der MOSFET wird mittels Port E3 geschaltet. Durch Umkonfigurierung des Jumpers J21 (siehe Foto auf der folgenden Seite) kann der MOSFET alternativ von Port G1 geschaltet werden. Um den MOSFET durchzuschalten, muss der entsprechende Port (E3 oder G1) als Ausgang geschaltet werden und auf Low gezogen werden (Low=Einschalten, High=Ausschalten).

Die abzuschaltenden Bereiche sind:

- RS485 Transceiver - durch Jumper J3 an Vcc5 oder Vcc5\_2 zu legen
- CAN Transceiver (liegt immer an Vcc5\_2)
- Beim RS232 Transceiver ist eine Abschaltung wegen des integrierten Shutdown-Modus nicht notwendig
- ~~5-Volt-Zweig des SD-Karten-Pegelwandlers~~ (nicht mehr möglich und auch nicht notwendig - siehe auch Seiten 52 und 92)
- Die SD Karte kann mit einem separaten Transistor abgeschaltet werden (mehr hierzu auf Seite 55)
- 3 Volt Spannungsregler (und somit dann auch Display, Display-Pegelwandler und SD Karte) – durch Jumper J6 an Vcc5 oder Vcc5\_2 zu legen
- Zusätzlich ist noch ein Lötpad P3 (blaue Markierung auf dem Foto der nächsten Seite) mit der geschalteten Spannung versorgt, so dass Sie hier auch eigene Verbraucher anschließen können.

**Bitte beachten Jumper J17:** Wenn Sie die Abschaltoption nicht mitbestellt haben oder temporär nicht benötigen (bzw. programmieren möchten), so muss Jumper J17 geschlossen sein (**Lieferzustand ist geschlossen – wenn Sie die Abschaltfunktion nutzen möchten, so müssen Sie den Jumper auftrennen**).



Standardmäßig liegt der Eingang des 3-Volt Reglers immer am Ausgang des 5 Volt Reglers, ist also permanent in Betrieb. Soll auch der 3 Volt Spannungsregler (Display, SD Karte) geschaltet werden, **so ist hierfür Jumper J6 (rote Markierung) umzukonfigurieren.**

Vom 3 Volt Regler werden folgende Module gespeist:

- Display
- Displayelektronik
- SD Karte
- 3 Volt Pad an Stecker für Port B

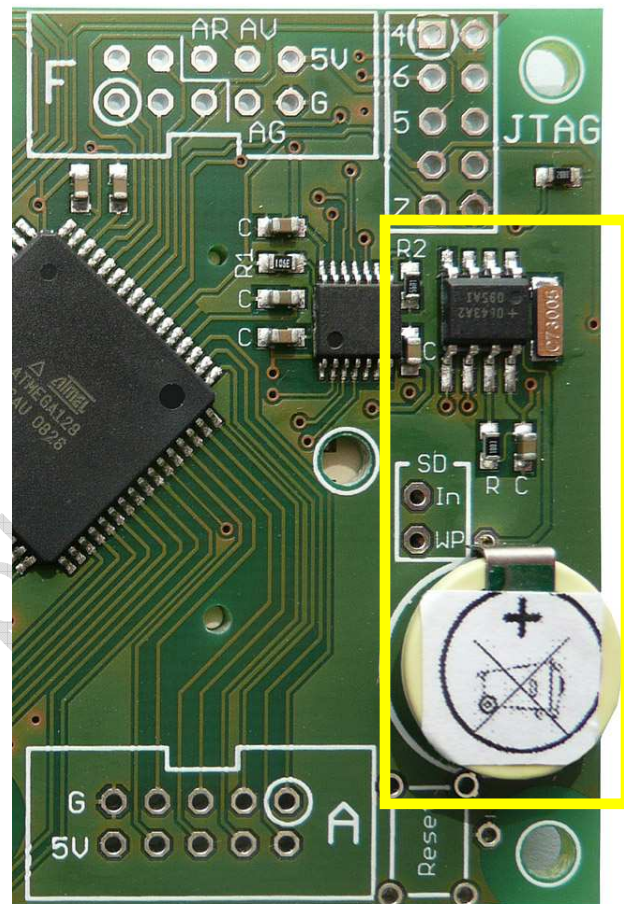
Wenn der 3 Volt Regler abgeschaltet wurde, so müssen die o.g. Bausteine nach einem Einschalten erst neu initialisiert werden, ansonsten werden sie nicht ansprechbar sein.

Bei Betrieb einer SD Karte sollten Sie unbedingt alle offenen Dateien vor dem Abschalten schließen, ansonsten ist ein Datenverlust wahrscheinlich.

## Echtzeituhr

Für eine Mikrocontroller-Lösung kann es sinnvoll sein, eine unabhängig laufende Echtzeituhr zu integrieren: auch nach einem Reset, einem Stromausfall (und sei es nur durch Batteriewechsel) oder Neuprogrammierung kann somit bei einem Neustart und auch während des Programmablaufs immer die aktuelle Uhrzeit sowie das aktuelle Datum abgerufen werden. Es gibt Anwendungen, wo dies nicht notwendig ist – alle Anwendungen wo jedoch Uhrzeit oder Datum genutzt werden (evtl. auch nur um auf der SD Karte geschriebene Dateien oder Datensätze zuordnen zu können), brauchen eine solche Echtzeituhr.

Das D071x Board bietet eine solche Echtzeituhr zusammen mit einer Miniatur-Lithiumbatterie als Option an (die gelbe Markierung zeigt die Echtzeituhr mit eigenem SMD Uhrenquarz, Lithium-Batterie sowie weitere notwendige Beschaltung). Die Batterie übernimmt die Versorgung der Echtzeituhr, wenn das Controllerboard nicht mehr versorgt wird. Wenn keinerlei Versorgungsspannung bereit steht, reicht die Batterie für ca. 2 Jahre aus ... Sobald die Versorgungsspannung wieder zur Verfügung steht, wird die Lithium-Batterie nicht belastet und kann für mind. 10 Jahre eine Pufferung bei Ausfall der Versorgungsspannung übernehmen. Interessante Zusatzfunktion: Der Echtzeituhrbaustein enthält zusätzlich noch 56 Byte RAM, die Sie nach Belieben beschreiben können (beliebig oft – RAM hat im Gegensatz zu einem Eeprom oder Flash keine begrenzte Lebensdauer). Auch dieses RAM ist gepuffert – behält seinen Inhalt also auch Unterbrechung der Spannungsversorgung. Das ist sehr praktisch, denn Sie können so z.B. während des Programmablaufs permanent Konfigurations- und Zustandsdaten in diesem RAM ablegen – bei einem Neustart kann das Mikrocontrollermodul dann daraufhin feststellen, was die letzten Aktionen waren und evtl. sogar sein Programm an der Abbruchstelle fortsetzen.

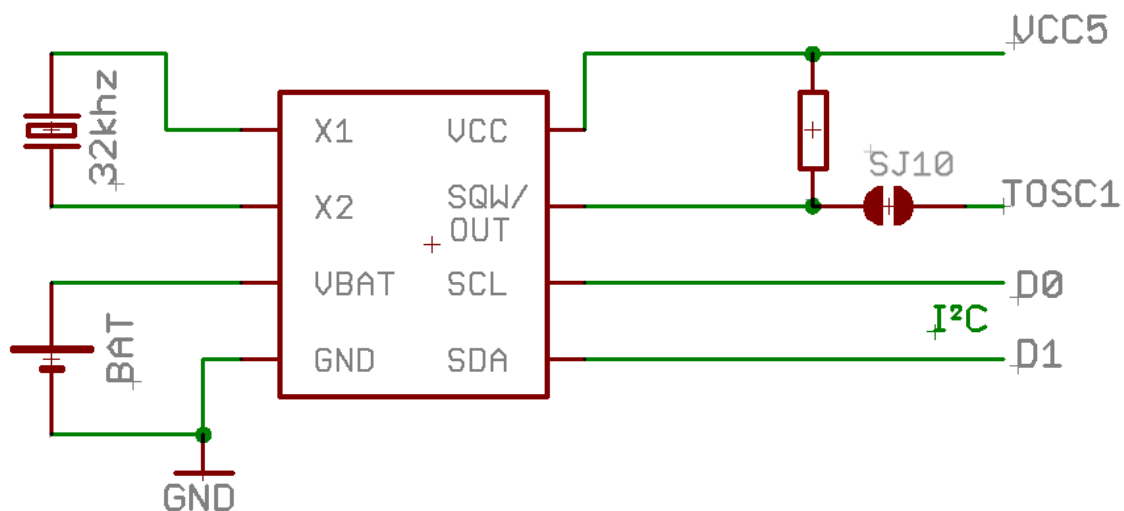




Der Uhrenbaustein kann von Ihnen so konfiguriert werden, dass am Ausgang RTC\_Clock ein Clocksignal ausgeht, welches auf dem D071x an TOSCI des Mikrocontrollers gelegt wurde.

Mit dem Schließen von J10 (dieser liegt direkt neben dem Mikrocontroller) verbinden Sie TOSCI mit dem Clock-Signal der Echtzeituhr. Hiermit ersparen Sie sich einen eigenen Uhrenquarz direkt am Mikrocontroller, denn nun erhält der Mikrocontroller das Signal der Echtzeituhr (Anmerkung: das Signal ist programmierbar zwischen 32.768 Khz und 1Hz).

**Achtung:** Obiges gilt nur für ATmega2561 und AT90CAN128. Ein ATmega128 kann nicht mit einem externen Takt versehen werden. Hier ist ein eigener Uhrenquarz zwischen TOSCI und TOSC2 notwendig ! Port G3 ist dann mit einem ATmega128 nicht mehr nutzbar (siehe auch Seite 31) – auch Tasterjumper TJ\_B sollte dann umkonfiguriert werden, denn ansonsten ist eine Taste mit G3 verbunden (nach Umkonfigurieren von TJ\_B liegt der Taster dann an G2).



## SD Kartenmodul

Mit dieser Erweiterung können Sie zukünftig SD Karten als externe Speichereinheit für Ihren Mikrocontroller nutzen.

### WICHTIG:

- 1) SD-Karten sind kein RAM sondern Flash-Speicher. Jede Speicherzelle der Karte hat eine Lebensdauer von ca. 100.000 Schreibzyklen. Angeblich soll so eine Karte selber dafür sorgen, dass die Speicherzellen gleichmäßig beschrieben (verbraucht) werden (d.h. , wenn Sie eine Karte immer nur zu 10% beschreiben, demnach 1 Million Schreibzugriffe möglich wären). Trotzdem: Bei einer falschen Programmierung (Schleife mit permanentem Schreibzugriff) ist so eine Karte somit binnen 1 Sekunde zu zerstören. SD Karten sind also nur geeignet, um Daten längerfristig abzuspeichern bzw. als Datenlogger, nicht aber um ständig und permanent Daten dort upzudaten.
- 2) Die Software für die eigentliche Kommunikation mit der SD Karte stammt nicht von uns. Bitte sprechen Sie den Autor der Software-Treiber bei Fragen an.
- 3) Es gibt sicher tausende unterschiedlicher SD Karten am Markt. Nicht jede wird einwandfrei mit dem Mikrocontroller zusammenarbeiten wollen. Manche sind sehr langsam, obwohl sie als High-Speed ausgezeichnet wurden. Wenn also eine Karte nicht wie gewünscht funktioniert, probieren Sie andere Karten aus. Wir können hier nicht helfen.
- 4) Die übliche Software kann nur mit Dateinamen bis 8 Zeichen Länge arbeiten. Speichern Sie also nur Dateinamen ab, die nicht mehr als 8 Buchstaben haben (+3 für den Extender). Beispiel: „Logo1.bin“ ist OK, „Hintergrundbild.bin“ ist nicht OK.

### Wichtiges zur Vermeidung von Datenverlust:

Wenn Sie auf die Karte Schreiben möchten, so führt ein Reset, Abschalten oder Stromausfall etc. dazu, dass eine evtl. noch geöffnete Datei später nicht mehr lesbar ist, die Dateigröße nicht stimmt bzw. Cluster verloren gingen, weil sie noch nicht in die FAT eingetragen wurden.

Daher: Vor dem Abschalten müssen zum Schreiben geöffnete Dateien immer geschlossen werden. Daher sind u.E. zwei Maßnahmen sinnvoll:

Wenn kleinere Datenmengen eher selten bzw. mit Abständen geschrieben werden sollen:  
Datensätze zuerst im RAM anlegen und dann alle x Minuten die Daten auf einen Schlag abspeichern oder wenn z.B. 512 Byte zum Schreiben „eingesammelt“ wurden. Dazu vorher die Datei öffnen und direkt danach wieder schließen. Das Gleiche gilt beim „Anhängen“ von Daten. Auf diese Weise verliert man nur die Daten seit dem letzten Abspeichern – es sei denn, der Stromausfall passiert genau während des Schreibzugriffs – dann haben Sie Pech gehabt (evtl. Strategie: 2 Dateien gleichzeitig führen und nacheinander Beschreiben).

### Wenn große Datenmengen geschrieben werden sollen

Dies tritt i.d.R. nicht permanent auf, denn dies würde (s.o.) schnell zu einem Erreichen der 100.000 Schreibzyklen führen. Daher gilt dann auch hier: Datei Öffnen – Daten ablegen/anhängen – Datei Schließen

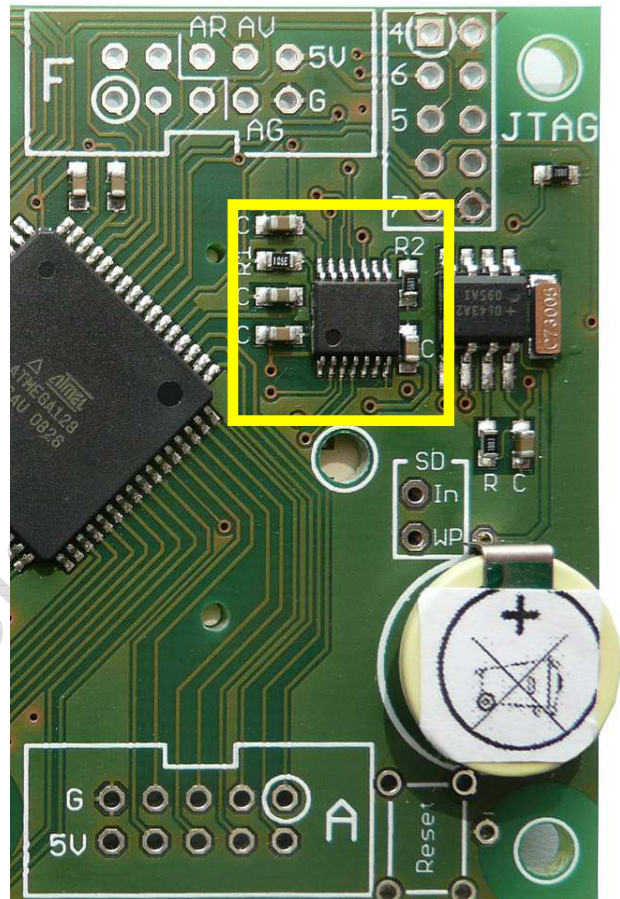
Die SD-Kartenerweiterung erfüllt mehrere Aufgaben:

- Steckkontakt für SD-Karten
- Bidirektionaler Pegelwandler von 3,3-5 Volt (Mikrocontroller) zu 3 Volt (SD-Karte) und zurück
- Abkoppelung des kompletten Moduls vom SPI Bus durch Tristate-Ausgänge

### **Warum ein bidirektionaler Pegelwandler mit Tristate Outputs?**

Eine SD-Karte benötigt eine Spannung von 2,7 bis 3,6 Volt – aber je nach Nutzung schwankt der Strombedarf der Karte von wenigen mA bis zu (kurzfristig) 100mA. Damit ist eine 3 Volt Spannungsregelung, wie sie auf dem D071x vorhanden ist, obligatorisch.

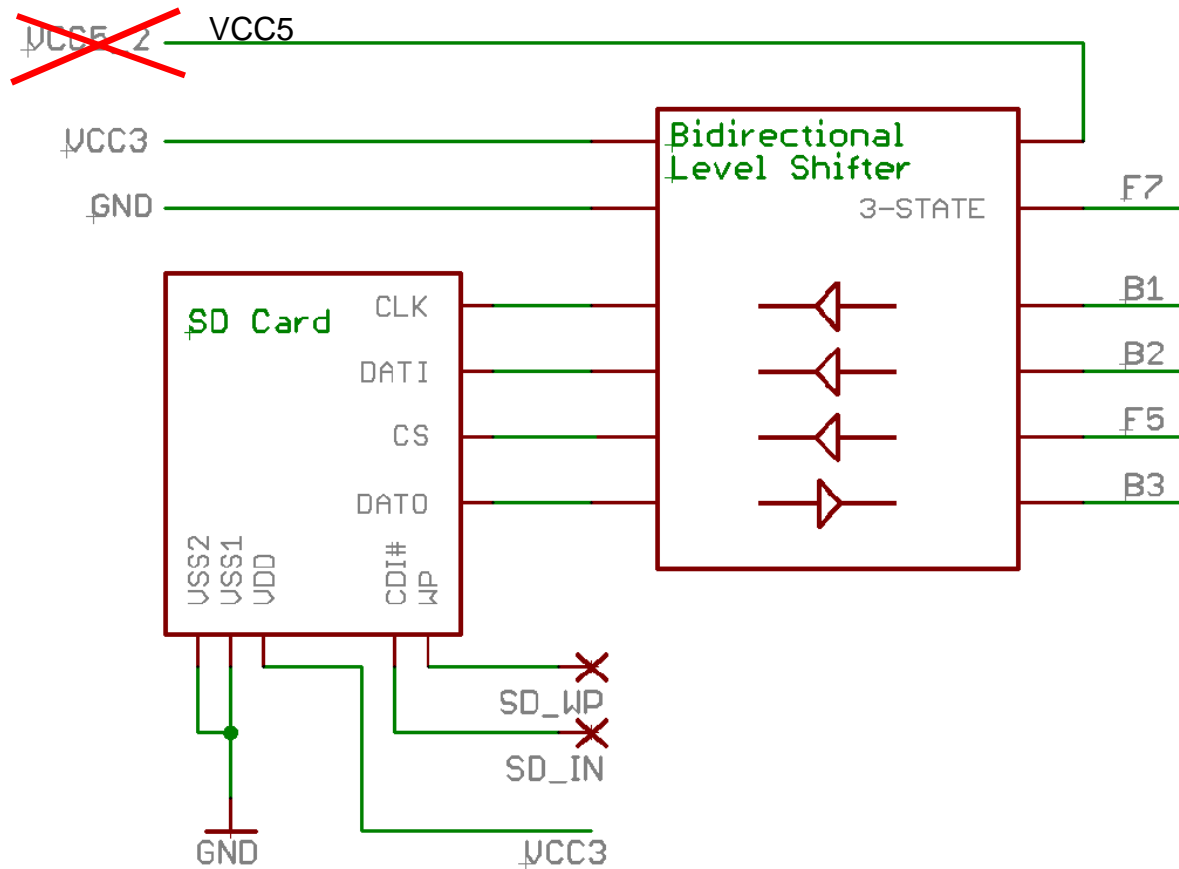
Die direkte Verbindung zwischen eine 5 Volt Mikrocontroller und einer SD Karte verbietet sich, da diese durch den 5 Volt Pegel der Signale zerstört würde. Die Signalpegel der Datenleitung müssen also von 5 Volt auf 3 Volt (und umgekehrt) umgesetzt werden. Bei einem 3 Volt-Mikrocontroller könnten Sie diesen natürlich auch direkt anschließen – allerdings hätten Sie immer noch das Problem, das die Karte am SPI Bus das Display stören würde (auch bei Inaktivität der Karte). Aus diesem Grunde findet sich auf dem D071x ein bidirektionaler Pegelwandler (d.h. er wandelt 3 Volt in 5 Volt Signale und umgekehrt), der durch die Activate-Leitung (Port F7) komplett vom Bus weggeschaltet werden kann (gelbe Markierung zeigt den Pegelwandler nebst div. weiterer notwendiger Beschaltung).



### **Tristate-Ausgänge**

Neben dem Vorhandensein der Spannungsversorgung MUSS die Leitung *Activate* (Port F7) an High liegen, damit die SD Karte Daten erhält bzw. sendet. Erst die Leitung *Activate* schaltet den bidirektionalen Pegelwandler der SD Karte ein. Ein Low schaltet den Pegelwandler auf Tristate. Dies ist notwendig, da ansonsten die SD Karte u.U. den Bus blockiert, auch wenn sie gerade gar nicht angesprochen wird.

### Beschaltung SD Kartenmodul



Die SD Kartenansteuerung benötigt neben den drei SPI Ports Clock, Mosi und Miso (B1, B2, B3) noch zwei weitere Ports des Mikrocontrollers: F5 für die CS Leitung sowie F7 für die Aktivierung/Deaktivierung des Pegelwandlers.

### **Konflikt bei Konfiguration des SPI Bus**

SPI Geräte liegen üblicherweise zwar am gleichen Bus, können aber durchaus unterschiedliche Datenformate erwarten. Im Falle eines unserer Farb TFTs und der SD Karte liegt dieser Fall z.B. vor:

TFT: Polarity = Low, Phase = 0

SD Karte : Polarity = High, Phase = 1

Zudem darf die Karte meist (dies ist Kartenabhängig) nicht ganz so schnell mit Daten versorgt werden, wie das Display.

Dies bedeutet, Sie müssen vor einem Zugriff auf die Karte bzw. Display nicht nur *Activate* (Port F7) entsprechend schalten sondern auch den SPI Bus umkonfigurieren!

Im Bascom Beispiel haben wir dies z.B. bereits für Sie durchgeführt. Dort gibt es zwei Routinen *Activate\_Display* und *Activate\_SD* die Sie einfach nur entsprechend aufrufen müssen. Tipp: Sie können z.B. *Activate\_Display* zusammen mit einer Abfrage, ob die Aktivierung überhaupt notwendig ist, auch in der Routine *LCD\_Window* unterbringen – dann wird das Display wenn notwendig automatisch aktiviert.

## Formatierung der SD Karte

Wenn die Karte beim ersten Test korrekt angesprochen wird, brauchen Sie sich um nichts zu kümmern. Wenn Sie auch nach einem Formatierungsversuch unter Windows nicht korrekt erkannt wird bzw. nicht angesprochen werden kann, versuchen Sie, die Karte mit einem speziellen Tool für SD Karten zu formatieren.

Unter folgendem Link:

[http://panasonic.co.jp/pavc/global/cs/sd/download/sd\\_formatter.html](http://panasonic.co.jp/pavc/global/cs/sd/download/sd_formatter.html)

erhalten Sie von Panasonic das Programm SD Formatter, mit welchem Sie ihre SD Karte vor der Nutzung formatieren sollten. Aus Copyright-Gründen können wir hier nur den Link posten und Ihnen das Tool leider nicht direkt übermitteln.

Das Handbuch zu dieser Software erhalten Sie hier:

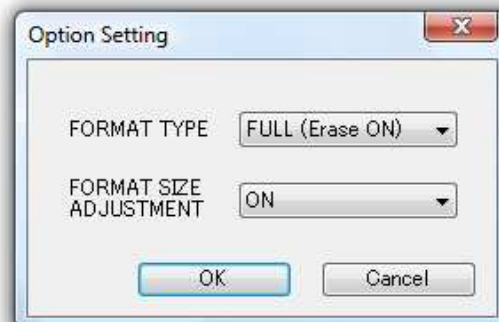
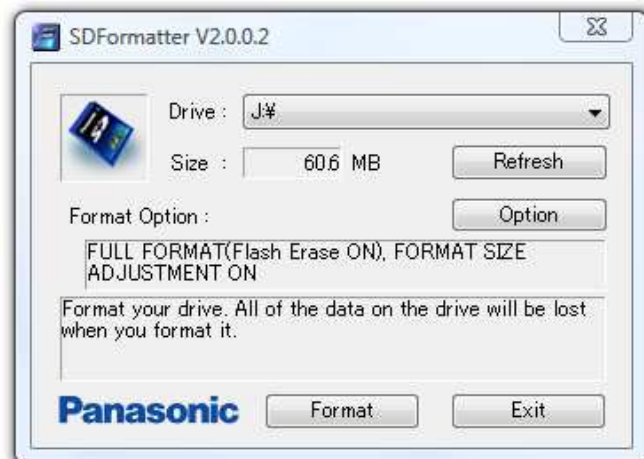
<http://panasonic.jp/support/global/cs/sd/download/ftp/manual2007e.pdf>

Als Backup falls der obige Link einmal nicht funktionieren sollte: Programm und Handbuch befinden sich auch hier: <http://www.sdcard.org/about/downloads/>

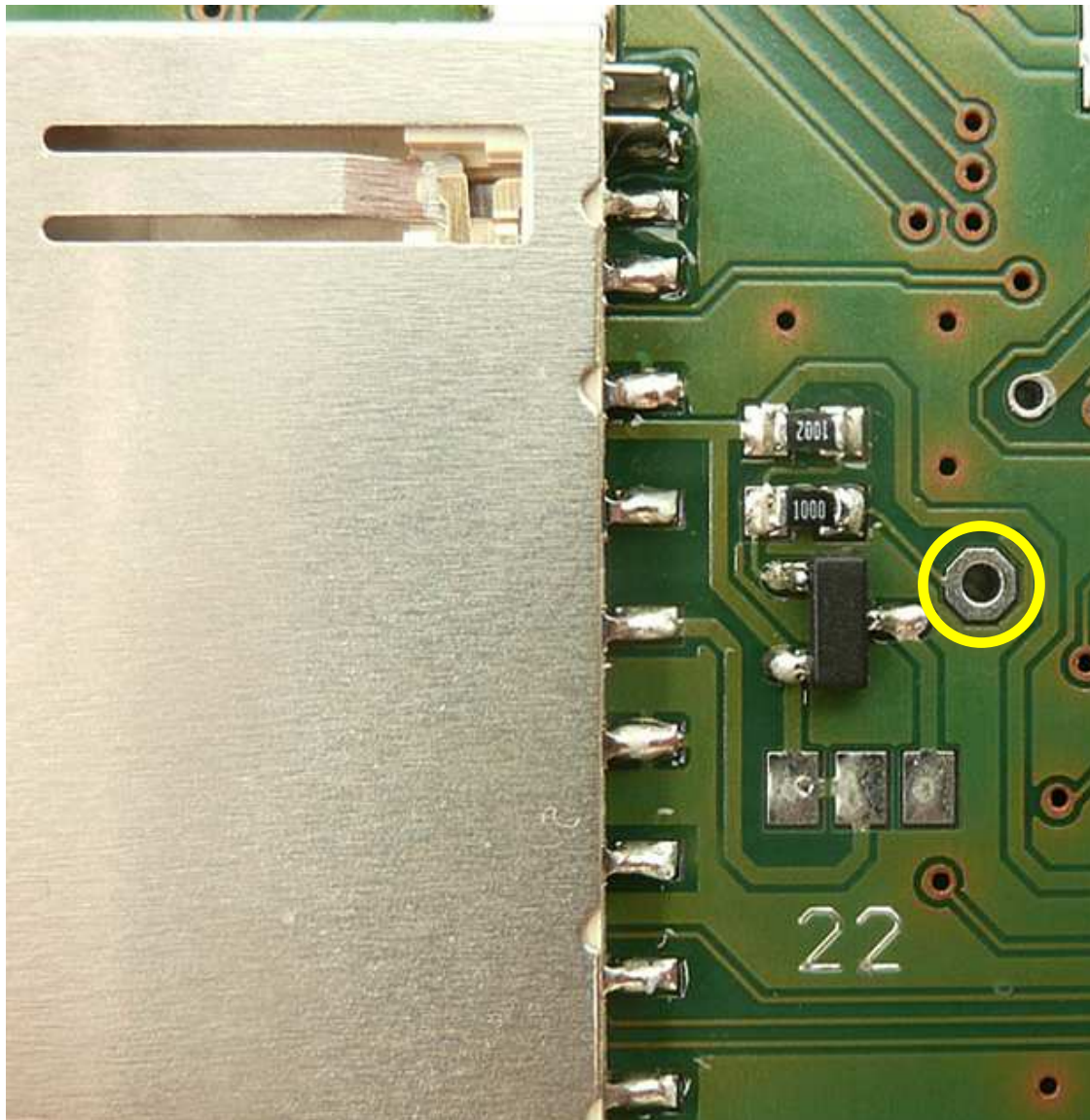
Tipp: Es reicht oft schon, einfach nur die Grundeinstellung zu nutzen:

- 1) Quickformat und bei
- 2) Format Size Adjustment: Off

**Tipp:** Es hat auch bereits Karten gegeben, die von allen möglichen Geräten erkannt wurden, aber nicht vom Mikrocontroller. Alle Formatierungsversuche unter Windows und mit dem Panasonic-Tool waren erfolglos. **Als Lösung hat sich hier eine Formatierung mittels einer Digitalkamera herausgestellt** (irgendwo in den Tiefen der Menüstruktur bietet jede Kamera die Funktion „Formatieren“ an).



## Abschaltung der Stromversorgung der SD Karte



Für den Fall, dass Sie bei einem batteriebetriebenen Gerät die SD-Karte, die ja doch eine Menge Strom benötigen kann, einzeln abschalten möchten, haben wir auf der Oberseite der Platine (Displayseite) eine Vorkehrung für eine Erweiterung getroffen. Mittels Jumper 22 können Sie festlegen, ob die Karte direkt an 3 Volt geschaltet sein soll, oder ob sie über den optionalen MOSFET abgeschaltet werden soll. Hierzu sind dann noch 2 Widerstände und ein P-Channel MOSFET notwendig (dieser muss bei  $V_{GS}$  3 Volt oder niedriger Durchschalten können; Gehäuse: SOT23). Der notwendige Steuerimpuls muss dann von Ihnen zu einem beliebigen Punkt (Port) hin verdrahtet werden (von dem freien Löt-pad am MOSFET = gelbe Markierung).

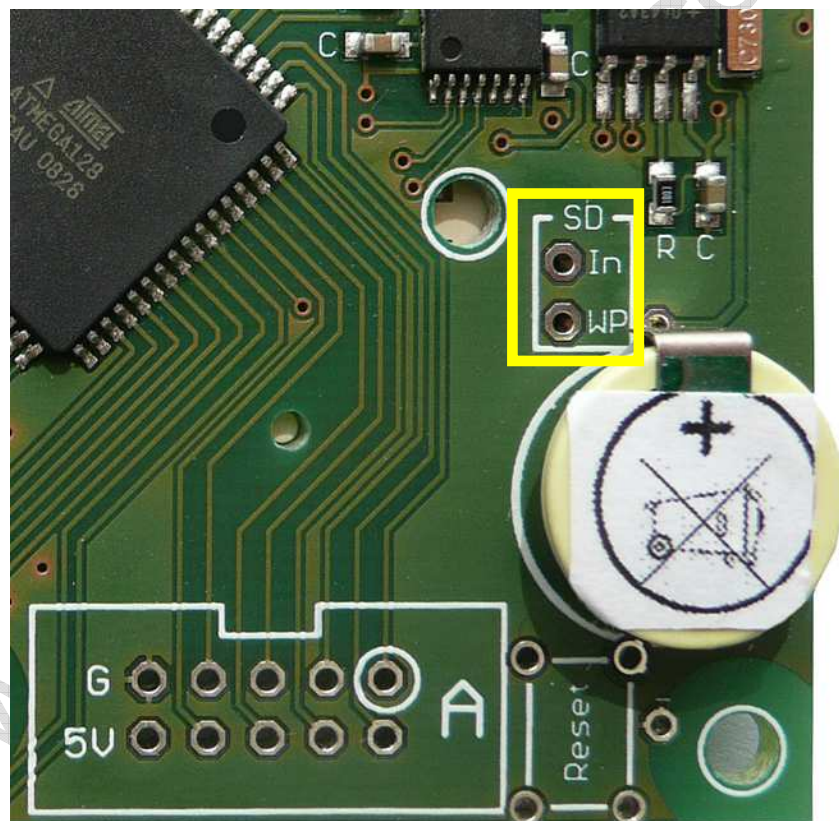
## Erkennung Schreibschutz / Karte eingesetzt

Hin und wieder kann es notwendig sein, sofort zu erkennen, ob eine Karte schreibgeschützt ist oder ob sie überhaupt eingesetzt wurde (oder evtl. auch zwischendurch entfernt wurde). Dies ist softwaremäßig nicht zu realisieren. Daher bietet der SD-Kartenhalter zwei Signale über zwei mechanische Schalter. Diese beiden Kontakte wurden von uns an die unten gezeigten Löt pads IN (Karte eingesetzt?) und WP (WriteProtect?) gelegt.

Der Schreibschutz-Schieber an der SD-Karte wird nicht von der Karte selbst abgefragt. Es ist kein interner Schalter, sondern lediglich ein Kunststoffschieber ohne jegliche Funktion für die Karte selbst. Sie können somit die Karte immer beschreiben, egal in welcher Position sich der Schalter befindet.

Diese Pads sind noch zu keinem Port verdrahtet – Sie müssen dies mittels einer einzulötenden Leitung realisieren. Die Kontakte der Karte schalten den entsprechenden Port gegen Masse wenn eine Karte eingesetzt ist bzw. wenn die Karte KEINEN Schreibschutz hat.

**ACHTUNG:** Diese Ports zur Abfrage von Schreibschutz und Kartenerkennung dürfen Sie nur als Eingang programmieren (mit gesetzten Pullup-Widerständen).



Wenn Sie diese Ports versehentlich als Ausgang programmieren, werden sie gegen Masse verbunden und somit kurzgeschlossen ... die Ausgangstransistoren brennen in kürzester Zeit durch – der Port ist dann für immer unbrauchbar.



## **Die Software zum Ansprechen der SD Karte**

Es gibt verschiedene Protokolle zum Auslesen und Beschreiben einer SD-Karte. Einige sind nicht für die Öffentlichkeit protokolliert und bedürfen einer Lizenzierung. Ein freier Zugriff ist allerdings mittels des SPI-Bus möglich. Dies ist zwar die langsamste Zugriffsart, aber für einen Mikrocontroller reicht die Geschwindigkeit in der Regel aus.

Die SD-Karte kann als Speichereinheit sektorweise beschrieben oder gelesen werden - dies ist per Software noch relativ einfach zu lösen aber oft doch unpraktisch, da man ein „richtiges“ Dateisystem gewohnt ist. So möchte man z.B. Daten von der Karte Lesen, die vorab mittels eines PCs auf sie geschrieben wurde, oder man möchte mit dem PC Daten auslesen, die der Mikrocontroller auf die Karte geschrieben hat. Dann geht kein Weg daran vorbei, das FAT Dateisystem der Karte zu nutzen. Dieses FAT Dateisystem muss der Mikrocontroller jedoch verwalten und das ist aufwändig und speicherintensiv. Ein kleiner Mikrocontroller mit wenig RAM wird hier u.U. bereits Probleme bekommen, Mind. 1 KByte RAM sollten schon verfügbar sein. Beim Lesen von der Karte und beim Schreiben auf die Karte gilt grundsätzlich: je größer der Puffer (Cache) im RAM hierfür, desto schneller ist dieser Vorgang.

Wie schon erwähnt, haben wir selbst haben keinerlei eigene Software für den Zugriff auf SD Karten programmiert. Nachfolgend geben wir Ihnen ein paar Hinweise und Hilfestellung zur Nutzung der SD Karten Treiber..

Wenn Sie Ihre Karte nicht ansprechen können, sind wir jedoch der falsche Ansprechpartner hierfür. Wir kennen diese Software nicht gut genug um Ihnen helfen zu können. Nutzen Sie dann bitte die diversen Internetforen oder bei Bascom auch das Supportforum des Herstellers unter [www.mcselec.com](http://www.mcselec.com)

## **Zugriff auf die Karte mit Bascom Basic**

Mit Bascom wird bereits AVR DOS mitgeliefert, welches den Zugriff auf die SD Karte erlaubt. Dieses stammt von Herrn Voegel. Seine Internetseite rufen Sie unter folgendem Link auf: <http://members.aon.at/voegel/>. 4 KByte Programmspeicher und ca. 1 K Ram werden hier mindestens benötigt.

**Beachten Sie bitte, das AVR DOS nur für den privaten Gebrauch lizenziert ist. Wenn Sie kommerzielle Applikationen erstellen, dann müssen Sie AVR DOS unter obiger Internetseite lizenzieren.** Eine Company-Lizenz kostet jedoch lediglich 99,95 Euro und ist somit erschwinglich. Unter dieser Internetseite erhalten Sie zudem alle wichtigen Informationen rund um die in Bascom integrierten Kommandos.

Für einen Test von AVR DOS mit einem unserer Grafikmodule erhalten Sie mit der Karte das Programm SDCardtest.bas sowie die dazugehörige Grafikdatei Grafik1.bin und Grafik2.bin, welche Sie bitte beide auf eine SD Karte kopieren (vorher evtl. Ix mit dem Panasonic Formatter wie weiter oben beschrieben, formatieren).

Zu allen weiteren Befehlen konsultieren Sie bitte die Webseiten zu AVR DOS bzw. die Onlinehilfe von Bascom. Und nochmals: Denken Sie immer daran, dass Sie jede Speicherzelle einer Flash Karte ca. 100.000 mal beschreiben können. Also programmieren Sie bitte keine Schleifen, die permanent mehrere Male pro Sekunde Daten auf die SD-Karte schreiben.

### **Schnellstart mit Bascom**

Im Verzeichnis \Softwarebeispiele finden Sie ein Unterverzeichnis für das D071x welches eine Version des Portcheck-Programms beinhaltet, welches auch auf die SD Karte zugreift. Sollte es fehlen, senden Sie uns eine kurze Mail (bitte mit Angabe der Kundennummer, welches sich auf Ihrer Rechnung oder Auftragsbestätigung befindet) und wir senden es Ihnen zu.

Für unserer 2.1“ Display benötigt diese Software neben den SD Karten-Routinen noch die üblichen 3 Librarydateien von Ihrer CD:

GLCD21\_Display3000.bas  
GLCD21\_fonts.bas  
Init21\_Display3000.bas

In Init21\_Display3000.bas nehmen Sie bitte noch eine Änderung vor: damit wir die Daten auch schnell genug aus der SD Karte lesen können, müssen wir ein Array definieren, welches alle Pixel einer kompletten Zeile beinhaltet. Maximal können dies 176 Pixel sein – jedes Pixel benötigt 2 Byte – das macht dann insg. 352 Byte als benötigter Pufferspeicher im RAM. Das Array gibt es schon, aber es muss vergrößert werden.

Also: bitte am Anfang dieser Datei die Arraygröße ändern in:  
**Dim Data\_array(352) As Byte**

Grundsätzlich geht der Zugriff mit Bascom wie folgt vonstatten:

- 1) Mit Config\_MMC.bas wird die SD Karte eingelesen, Config\_AVR-DOS.BAS lädt dann den Support für die meisten DOS Kommandos nach.
- 2) Config\_MMC.bas MUSS von Ihnen noch an Ihre aktuelle Hardware angepasst werden.
- 2) Mit Open # wird eine bestimmte Datei eröffnet oder angelegt.
- 3) Mittels Get# und Put# etc. werden Daten gelesen bzw. geschrieben.
- 4) Close # schließt die Datei wieder. Dies ist wichtig uns sollte niemals unterbleiben – also nicht einfach den Controller vom Strom nehmen. Alles weitere erfahren Sie auf den Internetseiten von Herrn Vögel oder in der Bascom-Online-Hilfe.

### **Auflistung der Bascom-Befehle, die mit AVR DOS zusammenarbeiten:**

#### **Disk/Directory:**

[InitFileSystem](#) (<Partition#>)  
[DiskSize](#)  
[DiskFree](#)  
[Kill](#) <FileName>  
[Dir](#) ([ <FileName> ])  
[FileLen](#) ([ <FileName> ])  
[FileDateTime](#) ([ <FileName> ])  
[FileDate](#) ([ <FileName> ])  
[FileTime](#) ([ <FileName> ])  
[GetAttr](#) ([ <FileName> ])  
Name <OldFileName> As <NewFileName>  
[ChDir](#) <PathName>])  
[MkDir](#) <PathName>])  
[Rmdir](#) <PathName>])

#### **File create, open, read, write, close:**

[FreeFile](#)  
[Open](#) <FileName> For Input/Output/Append/Binary as #<File#>  
[Close](#) <File#>  
[Flush](#) [<File#>]  
[Print](#) #<File#>, Variable1; Variable2; ...  
[Write](#) #<File#>, Variable1, Variable2, ...  
[Input](#) #<File#>, Variable1, Variable2, ...  
[Line Input](#) #<File#>, StringVariable  
[Get](#) #<File#>, <Variable> [,<Position>]  
[Put](#) #<File#>, <Variable> [,<Position>]  
[Seek](#) #<File#> [,<New Position>]

#### **Infos about opened files:**

[EOF](#) (#<File#>)  
[LOC](#) (#<File#>)  
[LOF](#) (#<File#>)  
[FileAttr](#) (#<File#>)

#### **Others:**

[Bload](#) <FileName>, <SRAM-Address>  
[Bsave](#) <FileName>, <SRAM-Address>, <Length>

Mehr hierzu erfahren Sie auch in der Bascom-Hilfe. Suchen Sie einfach nach „AVR-DOS“.

### **Zugriff mit C (WinAVR)**

Hier gibt es verschiedene Projekte im Internet, die den Zugriff auf eine SD Karte erlauben.

Wir haben das Beispiel von Herrn Klabunde an unser Board angepasst und stellen es Ihnen auf Anfrage zur Verfügung. Herr Klabunde hat hierzu seine Genehmigung erteilt. Grundsätzlich funktioniert es jedoch genauso wie im Bascom Code gezeigt. **Auch für C gilt hier:**

Vor dem Ansprechen der SD Karte:

- SPI Schnittstelle definieren (falls zuvor z.B. für das Display geändert)
- die Activate Leitung auf High setzen

Nach dem Ansprechen der SD Karte

- Activate Leitung auf Low
- vor dem Ansprechen des Displays: SPI Schnittstelle definieren

### **Datenblöcke**

Die SD-Karte kann als Speichereinheit blockweise beschrieben oder gelesen werden - dies ist per Software noch relativ einfach zu lösen. Oft möchte man jedoch Daten von der Karte Lesen, die vorab mittels eines PCs auf sie geschrieben wurde, oder man möchte mit dem PC Daten auslesen, die der Mikrocontroller auf die Karte geschrieben hat. Dann geht kein Weg daran vorbei, das FAT Dateisystem der Karte zu nutzen. Dies muss der Mikrocontroller jedoch verwalten und das ist aufwändig und speicherintensiv. Ein kleiner Mikrocontroller mit wenig RAM wird hier u.U. bereits Probleme bekommen, 2 KByte RAM sollten schon verfügbar sein. Beim Lesen von der Karte und beim Schreiben auf die Karte gilt grundsätzlich: je größer der Puffer (Cache) im RAM hierfür, desto schneller ist dieser Vorgang.

**Wenn Sie genug Speicher zur Hand haben, lesen/schreiben Sie immer gleichzeitig 512 Byte (=1 Sektor). Dies geht wesentlich schneller, als einzelne Bytes zu übertragen.**

## **Nutzung der SD Karte ab Ablage für Grafikdaten zusammen mit unseren Farb-TFTs**

Die Ablage von größeren Grafikdaten auf einer SD-Karte bietet sich an, kostet doch dieser Speicherplatz verhältnismäßig wenig. Allerdings gibt es hier bei Nutzung der schnellen Hardware-SPI-Schnittstelle ein grundsätzliches Problem, welches umgangen werden muss:

Fakt ist:

- Es ist grundsätzlich möglich, mehrere Busteilnehmer am gleichen SPI Bus zu betreiben, in diesem Fall also Display und SD Karte
- Der Mikrocontroller hat weniger RAM als die einzuladende Bitmapdatei an Datenmengen hat (eine Vollgrafik bei z.B. 176x132 Pixeln benötigt 45 KByte)
- Somit muss die große Datenmenge in mehreren Schüben eingelesen und an das Display geschickt werden.
- Am SPI Bus (und somit an Mosi und Clock) hängen Display und SD Karte gleichzeitig.
- Display und Karte haben zwar eigene CS-Signale
- Zur Festlegung des Bildbereiches, wird dieser mittels LCD\_Bitmap vorab definiert

Angenommen, wir möchten 45 x 1 KByte Bilddaten einlesen und diese dann jedes mal an das Display senden..... so führt dies zu einem Problem. Wir müssen, bevor wir zum 2. Mal die SD Karte ansprechen, die CS-Leitung des Displays auf 1 setzen (also deaktivieren), damit die Kommunikation mit der SD Karte nicht vom Display als Daten angenommen werden und als falsche Pixel erscheinen. Leider bricht jedoch das Display dann auch die aktuelle Funktion ab (wir hatten ja einen Bildbereich definiert, der nun nach und nach mit Pixeln gefüllt werden sollte). Nach dem Lesen des zweiten 1KByte-Blocks an Daten können wir nicht dort fortsetzen, wo das letzte Pixel gesetzt wurde. Als Ausweg böte sich an, 45 KByte RAM für das Einlesen zur Verfügung zu stellen, aber dies ist in der Regel viel zu aufwändig.

### **Hier hilft nur ein Workaround:**

Eine eigene Bitmap-Routine für die Übernahme von Datenblöcken. Als Beispiel finden Sie auf der CD das Programm SDCardtest.bas, welches eine Bitmapgrafik inkl. dem Abruf von Daten der SD Karte automatisch zeilenweise durchführt – d.h. es wird immer nur eine Zeile an Daten angefordert und an das Display geschickt. LCD\_Window wird also bei einer 100 Pixel hohen Grafik auch 100x aufgerufen. Die Y-Position ist dann jedes Mal um 1 erhöht, das Fenster jedes Mal auch nur 1 Pixel hoch.

Eines können Sie jedoch mit einer solchen Routine nicht mehr nutzen: Komprimierte Grafiken – durch den zeilenweisen Aufbau mit einem Abruf von genau so vielen Pixeln wie die Zeile breit ist. ist diese Option nicht mehr nutzbar.

Anmerkung: Theoretisch könnte die Routine natürlich erweitert werden, so dass auch die Nutzung von komprimierten Daten auf der SD Karte möglich wäre. Allerdings ist der Speicher auf der SD Karte in der Regel so reichlich bemessen, dass man die Komprimierung hier auch weglassen kann.

Umgekehrt ist dies übrigens dies kein Problem, da unser SD-Kartenmodul mit Tristate-Eingängen und -Ausgängen versehen ist, und die SD Karte einfach nach Erledigung eines Tasks komplett vom Bus abgekoppelt werden kann.

**Links:****Generell:**

<http://www.sdcard.org/home>

[http://www.sdcard.org/about/memory\\_card/pls/Simplified\\_Physical\\_Layer\\_Spec.pdf](http://www.sdcard.org/about/memory_card/pls/Simplified_Physical_Layer_Spec.pdf)

<http://www.chipcatalog.com/Doc/BC6DC20FC60BAC7595E5D0419DB6126E.pdf>

**AVR Bascom:**

<http://members.aon.at/voegel/>

<http://www.mcselec.com/>

<http://staff.ltam.lu/feljc/electronics/bascom/>

**Links zu C-Code für SD Karten:**

[http://elm-chan.org/docs/mmc/mmc\\_e.html](http://elm-chan.org/docs/mmc/mmc_e.html)

[http://www.ulrichradig.de/site/atmel/avr\\_mmc/d/pdf/hitachi\\_hb28b128mm2.pdf](http://www.ulrichradig.de/site/atmel/avr_mmc/d/pdf/hitachi_hb28b128mm2.pdf)

[http://www.embedded-os.de/index.html?pcfath\\_port.htm](http://www.embedded-os.de/index.html?pcfath_port.htm)

<http://www.avrfreaks.net/index.php?module=Freaks%20Files&func=viewFile&id=1436&showinfo=1>

<http://www.efsl.be/>

<http://www.avrfreaks.net/index.php?module=Freaks%20Files&func=viewFile&id=2251&showinfo=1>

<http://www.roland-riegel.de/sd-reader/doc/>

<http://www.cc5x.de/MMC/>

<http://dfn.dl.sourceforge.net/sourceforge/efsl/manual-0.2.8.pdf>

## Einsatz im Gehäuse GBS600 / GBS601

Das Modul D071x wurde auf das Handgehäuse hin maßgeschneidert, d.h. es passt exakt in die Aussparungen und auf die Befestigungsbolzen.

Folgendes gibt es noch zu beachten:

- a) Wenn Sie das SD Kartenmodul mitbestellt haben, so ist der untere mittlere Befestigungsbolzen im Weg: Genau dort liegt der SD Kartenhalter. Der Haltebolzen kann mit einem Seitenschneider einfach abgeknipst werden (siehe Foto). Leider ließ sich der Kartenhalter nicht anders positionieren. Dies ist aber nicht weiter dramatisch, da die Platine am unteren Ende noch durch zwei weitere Schrauben gehalten wird – drei Schrauben sind schon Luxus.

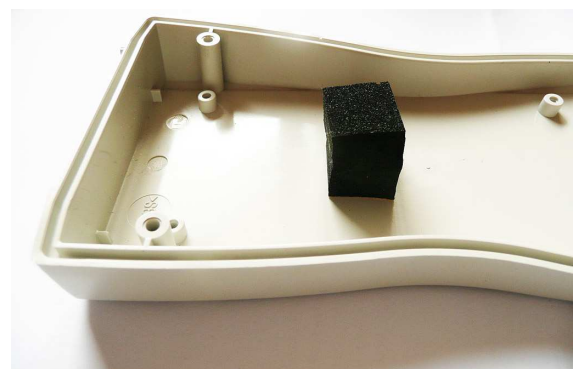


- b) Am oberen Ende bietet das Gehäuse in der Oberschale leider keine Befestigungsbolzen. Mit den Schrauben am unteren Ende ist die Platine bereits gut befestigt, sie sollte trotzdem (Gehäuse ohne Batteriefach) sicherheitshalber am oberen Ende noch fixiert werden. Hier gibt es nun (neben Einkleben) noch zwei Möglichkeiten der Befestigung:

- a. Entweder befestigen Sie an den Schraubbolzen in der Unterschale zwei Stehbolzen, die die Platine in der Oberschale fixieren (siehe Foto) oder



- b. Sie kleben in die Unterschale etwas Moosgummi, welches die Platine beim Schließen der beiden Gehäusehälften nach oben gegen die obere Gehäusehälfte drückt (siehe Foto).



**Bei einem Gehäuse mit Batteriefach ist diese Maßnahme nicht unbedingt notwendig – hier wird die Platine durch das Batteriefach gehalten.**

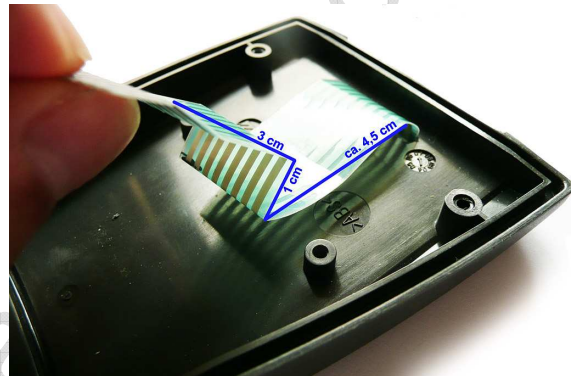
© [www.Display3000.com](http://www.Display3000.com)



- c) Wenn Sie die Folientastatur einsetzen möchten, so müssen Sie noch den Durchlass für das Folienkabel in die Oberschale einfräsen. Am besten gelingt dies durch das Bohren einiger nebeneinander liegender Löcher mit ca. 2-5 mm Durchmesser), die Sie dann mit einem Messer oder Feile oder Laubsäge etc. zu einem Schlitz erweitern. Dieser Bereich wird später komplett von der selbstklebenden Folientastatur verdeckt und ist unsichtbar sowie wasserdicht.



- d) Bei Einsatz der Folientastatur ist es sinnvoll, das Folienkabel der Tastatur zwei Mal zu knicken (Zickzack / Ziehharmonika) – dann lässt es sich am besten einbauen. Die beiden empfohlenen Knickstellen liegen (gemessen vom freien Ende) bei 3cm und 4cm (siehe Foto). Nutzen Sie nach Möglichkeit den von uns angebotenen extraflachen FPC Stecker – die üblichen erhältlichen Stecker (z.B. Molex etc.) sind leider zu hoch!

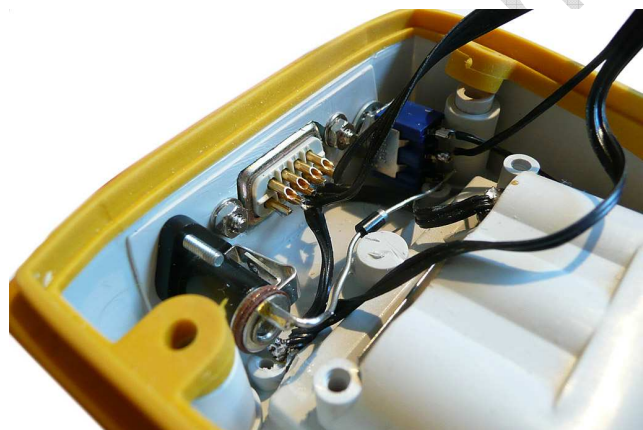


- e) Für eine SD Karte oder einzusetzenden D-Sub-Stecker müssen Sie noch die Aussparungen in das Gehäuse sägen / fräsen. Wir empfehlen hier:
- a. Für die SD-Karte reicht es, einfach nur am Gehäuserand mit einer Feile ein paar mm abzufeilen. Dies geht schnell und sieht sauber aus.



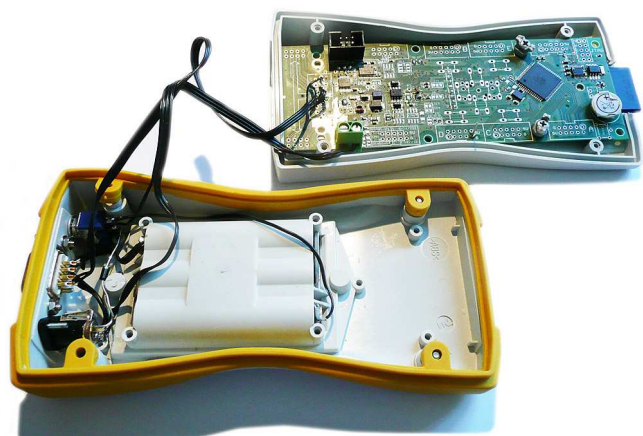
- b. Für die D-Sub-Stecker empfehlen wir den Einsatz eines außen montierbaren Steckers (oder Buchse), der mittels eines Mutter/Bolzen-Satz (z.B. Farnell Artikel Nr. 1217993) befestigt wird (siehe Fotos rechts, hier mit Schalter und Ladebuchse).

Hier ist das Ausschneiden der Öffnung nicht kritisch, da sie später von der Steckerblende verdeckt wird. Zudem hat der Stecker somit einen mechanisch festen Halt. Im nebenstehenden Beispiel sind die Stecker in der Unterschale angebracht, genauso gut ist auch in der Oberschale genug Platz. Das Gehäuse bietet zudem noch genügend Platz für eine kleine Zusatzplatine für spezifische Aufgaben. Dies können z.B. sein: Ladebeschaltung, DC Booster, USB Beschaltung, Optokoppler, Funkmodule etc. Gerne planen und entwickeln wir notwendige Zusatzplatinen für Sie. Ab einer Stückzahl von 50 Platinen (darunter wäre es wirtschaftlich vermutlich unsinnig) ist übrigens auch eine komplett an Ihre Bedürfnisse hin abgewandelte Version der D071x Platine möglich. Sprechen Sie uns bei Interesse an.



- c. Die entsprechenden Verbindungen führen Sie dann mittels kurzer Kabel durch (entweder direkt auf die nötigen Ports oder auf die D-Sub Kontakte auf der Platine (Beispiel siehe Foto rechts)).

Selbstverständlich können Sie auch direkt auf die Platinen zu löten Stecker zusammen mit dem Gehäuse einsetzen (wie auf Seite 19 zu sehen) – allerdings müssen Sie dann einen relativ großen Gehäuseausschnitt vorsehen damit der anzuschließende Gegenstecker auch passt.



## Übersicht Jumper / Lötbrücken

Auf den vorhergehenden Seiten wurden die verschiedenen Felder mit Jumpfern / Lötbrücken bereits angesprochen. Nachfolgend finden Sie nochmals eine Übersicht über diese Felder:

Jumper	Positionierung	Siehe Seite	Erläuterung
TJ_A	Displayseite, bei den Tastern	27	Wählt Tasterzuweisung zwischen <b>G0 (default)</b> und D2
TJ_B	Displayseite, bei den Tastern	27	Wählt Tasterzuweisung zwischen G2 und <b>G3 (default)</b>
TJ_C	Displayseite, bei den Tastern	27	Wählt Tasterzuweisung zwischen <b>D5 (default)</b> und G2
TJ_D	Displayseite, bei den Tastern	27	Wählt Tasterzuweisung zwischen <b>G1 (default)</b> und E7
TJ_E	Displayseite, bei den Tastern	27	Wählt Tasterzuweisung zwischen <b>E7 (default)</b> und D6
TJ_F	Displayseite, bei den Tastern	27	Wählt Tasterzuweisung zwischen E7 und <b>D6 (default)</b>
TJ_G	Displayseite, bei den Tastern	27	Wählt Tasterzuweisung zwischen G1 und <b>E5 (default)</b>
J5	Bauteilseite Mitte		Levelshifter Display. Jumper wählt zwischen <b>immer ein (default)</b> und schaltbar über Port F6
J6	Bauteilseite Mitte, neben d. 3V Regler	47	Legt fest, ob 3 Volt Spannungsregler abgeschaltet werden darf. <b>Default: immer ein</b>
J7	Bauteilseite zwischen PortB und ISP	35	Verbindet Port D2 mit TIn des RS232 Pegelwandlers. <b>Default: offen</b>
J8	Bauteilseite zwischen PortB und ISP	35	Verbindet Port D3 mit ROut des RS232 Pegelwandlers. <b>Default: offen</b>
J9	Bauteilseite links (Booster-Bereich)	45	Dieser Jumper legt fest, ob die Beleuchtung <b>immer aktiv (default)</b> ist oder vom Mikrocontroller (Port B7) geschaltet werden kann.
J10	Bauteilseite direkt am Controller oben links	31 & 49	Verbindet TOSCI mit dem Clock-Ausgang der Echtzeituhr. Nur nutzbar, wenn kein Uhrenquarz verbaut wurde und wenn die optionale Echtzeituhr verbaut ist <b>Default ist offen.</b>
J11	Bauteilseite oben, unterhalb des I0er ISP Steckers	35 & 37	Legt fest, ob E1 mit dem RS232 oder dem RS485 Transceiver verbunden werden soll. <b>Default: mit keinem Transceiver verbunden</b>
J12	Bauteilseite oben, unterhalb des I0er ISP Steckers	35 & 37	Legt fest, ob E0 mit dem RS232 oder dem RS485 Transceiver verbunden werden soll. <b>Default: mit keinem Transceiver verbunden</b>

J13	Bauteilseite, rechts neben dem CAN Transceiver	40 & 43	Verbindet CAN-TxD mit D5, <b>Default: Offen</b> . Muss bei Verwendung von CAN geschlossen werden
J14	Bauteilseite, rechts neben dem CAN Transceiver	40 & 43	Verbindet CAN-RxD mit D6, <b>Default: Offen</b> . Muss bei Verwendung von CAN geschlossen werden
J15	Bauteilseite unten, rechts zw. VCC Stecker und serieller Verbindungsstecker	42	Schaltet beim Schließen den 120 Ohm Abschlusswiderstand für CAN ein. <b>Default: Offen</b>
J16	Bauteilseite unten, links zw. VCC Stecker und serieller Verbindungsstecker	38	Schaltet beim Schließen den 120 Ohm Abschlusswiderstand für RS485 ein. <b>Default: Offen</b>
J17	Bauteilseite Mitte, unterhalb des MOSFET	47	Überbrückt den optionalen MOSFET zum Schalten verschiedener Verbraucher. <b>Default: Geschlossen</b>
J18	Bauteilseite oben links	20	Verbindet Rx mit dem D-Sub9-Stecker. <b>Default: Offen</b>
J19	Bauteilseite oben links	20	Verbindet Tx mit dem D-Sub9-Stecker. <b>Default: Offen</b>
J20	Bauteilseite oben links	20	Verbindet GND mit dem D-Sub9-Stecker. <b>Default: Offen</b>
J21	Bauteilseite Mitte	47	Legt fest, ob der Port für die Schaltung der Verbraucher über G1 oder E3 geschaltet wird. <b>Default ist E3</b>

## Übersicht Portbelegung

Bitte nutzen Sie hierfür die separate Excel-Tabelle *Portbelegung DS071x.XLS*

## Technische Daten Display-Modul-Bausätze:

### Artikel D071x:

<b>Maße:</b>	146 x 54 mm ca. 8 mm hoch (mit Display, ohne Taster)
<b>Gewicht:</b>	ca. 35-40 Gramm (schwankt je nach Ausstattung)
<b>Versorgungsspannung:</b>	3 bis 12 Volt Gleichstrom; optional bis 18 Volt (bitte Kapitel über benötigte Spannungen beachten)
<b>Mikrocontroller:</b>	Atmel ATmega 128 (Option: ATmega2561, AT90CAN128 <sup>*1 *2</sup> ) 128 KByte Programmspeicher (256 KByte <sup>*2</sup> ) 4 KByte RAM (8 KByte <sup>*2</sup> ) 2 KByte Eeprom max. 16 MHz Takt (je nach bestellter Option)
<b>Display:</b>	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:	mind. 5 Jahre
Lebensdauer Hintergrundbeleuchtung:	mind. 5000 Stunden (=Helligkeit ist auf 70% zurückgegangen). Tipp: die Lebensdauer steigt beträchtlich bei gedimmtem Betrieb
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 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)

### Optional gegen Aufpreis ist dieses Board verfügbar mit:

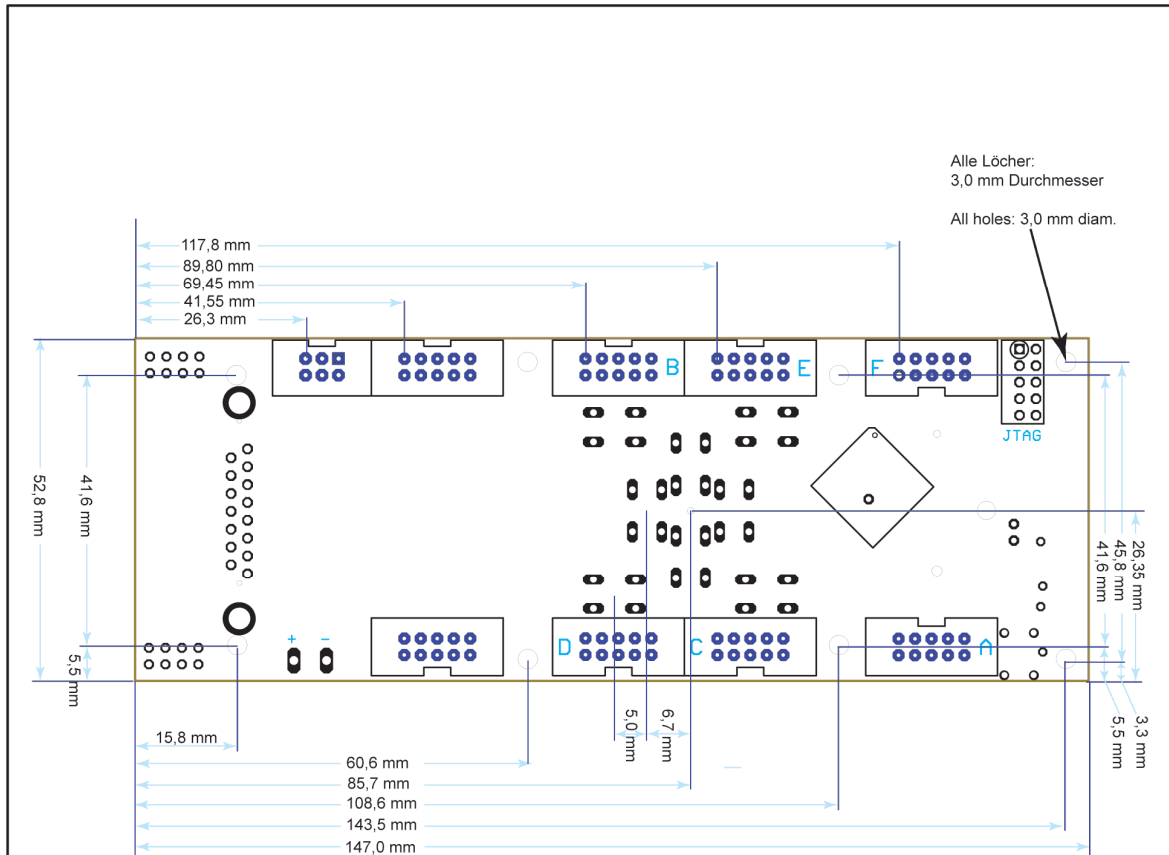
<sup>\*1</sup>) ATmega2561 (256 KByte Flash, 8 KByte RAM, 8 KByte Eeprom)

<sup>\*2</sup>) AT90CAN128 (wie Atmega128 aber mit CAN-Schnittstelle)

### Der Atmel Mikrocontroller wurde von uns gegenüber der Standardauslieferung wie folgt umprogrammiert:

- ATmega128: **Mega103 Mode** wurde abgeschaltet
- ATmega2561 und AT90CAN128: Fuse **Clockdiv18** wurde abgeschaltet
- alle: Fuse für die **Taktfrequenz** steht entweder auf 0100 (8 MHz intern) oder, wenn Sie einen externen Quarz mitbestellt haben auf 1111
- Fuse **Preserve EEPROM** when chip erase wurde eingeschaltet
- Fuse **Brown Out** wurde auf 2,7 Volt gesetzt
- Der Programmspeicher wurde zu Testzwecken bereits vorprogrammiert.

## Maßzeichnung (siehe auch separate PDF Datei auf der CD)



Bauteilseite der Platine / Component side of PCB

Alle blauen Pads sind zueinander im 2,54mm Raster angeordnet

All blue pads are placed in 0.1" (2,54mm) frame

Make sure you print this PDF 1:1 - deselect automatic resizing in printing options

Bitte diese PDF 1:1 drucken. Die automatische Größenanpassung in den Druckoptionen muss abgeschaltet sein.

### D071x - Drawing 1:1

Measurement: mm (1 inch equals 25,40 mm)

by: Display3000

Rev.  
1

Date: 22.06.2010

Sheet: 1/1

## Oft gestellte Fragen (FAQ):

### ***Ich kann die Ports F4 bis F7 nicht benutzen***

Port F4 bis F7 stehen nicht zur Verfügung wenn JTAG eingeschaltet ist. Sollten Sie einen JTAG-Zugriff nicht benötigen, so können Sie **Fusebit F:** auf **JTAG disabled** ändern und somit auch die Ports F4, F5, F6, und F7 nutzen (Tipp: alternativ kann JTAG auch temporär während der Laufzeit per Software ein- oder ausgeschaltet werden).

### ***Ich kann die Port D0 und D1 nicht benutzen***

Diese Ports können Sie nur nutzen, wenn Sie kein I<sup>2</sup>C benötigen. I<sup>2</sup>C ist notwendig, wenn Sie die Echtzeituhr ansprechen möchten. Brauchen Sie kein I<sup>2</sup>C, muss Jumper J3 geöffnet werden (im Lieferzustand geschlossen). Siehe auch Seite 33.

### ***Ich kann die Ports D5 und D6 nicht benutzen***

Vermutlich haben Sie einen AT90CAN128 Controller und haben die Jumper J13 und J14 geschlossen. D5 und D6 sind die Datenleitungen für den CAN Transceiver und können nicht zugleich mit CAN genutzt werden. Brauchen Sie kein CAN, öffnen Sie die Jumper J13 und J14 (Lieferzustand: offen).

### ***Ich kann nicht auf die Echtzeituhr zugreifen***

Bitte prüfen Sie, ob der Jumper J3 geschlossen ist (Lieferzustand geschlossen). Dieser verbindet die danebenliegenden I<sup>2</sup>C-Pullup-Widerstände mit der Versorgungsspannung.

### ***Ich kann nicht auf die SD Karte zugreifen***

Bitte prüfen Sie, ob JTAG abgeschaltet wurde. Der Zugriff auf die SD Karte benötigt Port F5 und F7 – diese sind nur per abgeschalteter JTAG Fuse nutzbar. JTAG kann auch softwaremäßig temporär ein- und abgeschaltet werden (siehe Datenblatt des Controllers).

### ***Ich kann keine Analogmessungen durchführen***

Vermutlich haben Sie nicht die Leitungen AVcc, AGND und evtl. ARef verbunden. Bitte lesen Sie das Kapitel *Analogeingänge* auf Seite 43.

Funktionieren die Analogeingänge 0 bis 3 aber die Eingänge 4 bis 7 nicht, lesen Sie bitte weiter oben den FAQ Abschnitt „Port F4 bis F7 stehen nicht zur Verfügung“.

### ***Ich habe ein RS232 Kabel an den D-Sub-9 Stecker angeschlossen, bekomme aber keine Verbindung***

- 1) Sind die Jumper J18, J19, J20 geschlossen?
- 2) Nutzen Sie die UART1 (Rx1/Tx1) in der Software?
- 3) Nutzen Sie ein „normales“ 1:1 beschaltetes RS232 Kabel? Mit einem sog. Nullmodemkabel funktioniert es nicht.
- 4) Sind die Jumper J7 / J8 geschlossen ?
- 5) Der RS232 Transceiver geht in den Shutdown-Modus wenn keine Gegenstelle vorhanden ist: Es muss tatsächlich auch ein Pegel an Rx anliegen (d.h. es muss eine Gegenstelle vorhanden sein), damit der Transceiver erwacht und auf Tx Daten senden kann!

### **Ich kann CAN nicht nutzen**

- 1) Haben Sie das Board mit AT90CAN128 Mikrocontroller bestellt?
- 2) Sind die Jumper D5 und D6 geschlossen?
- 3) Ist der Abschlusswiderstand zugeschaltet (wenn notwendig)?

### **Das Board funktioniert, aber nach einer Weile blinkt die Beleuchtung und mein Programm stürzt ab oder das Board resettet sich selbstständig.**

Unter Umständen wird der Spannungsregler zu heiß und geht in die Notabschaltung. Bitte lesen Sie das Kapitel *Spannungsregler* auf Seite 13. Denken Sie auch daran, dass bei einer größeren Belastung (z.B. durch weitere externe Verbraucher am 5V oder 3V Zweig) die errechneten Grenzwerte nicht mehr gelten. Verbauen Sie entweder einen belastbareren Regler oder reduzieren Sie die Eingangsspannung.

### **Das Display „spinnt“ gelegentlich**

Der glänzende Bereich am unteren Rand des Displays ist der Displaycontroller. Dieser ist lichtempfindlich. Verbauen Sie das Modul so, dass dieser Bereich verdeckt ist, oder kleben ihn mit einem dunklen Klebeband ab.

### **Ich kann plötzlich nicht mehr auf den Controller zugreifen**

Dies kann verschiedene Ursachen haben:

- 1) Zerstörung des Controllers durch elektrostatische Aufladung: Wenn Sie elektrostatisch aufgeladen sind und die Platine anfassen, gibt es u.U. ein Spannungspotential von mehreren zehntausend Volt. Die Ströme sind zwar minimal und Sie spüren dies evtl. noch nicht einmal, aber der Controller oder andere Bausteine sind hier empfindlich und können beschädigt werden. Auch wenn es eintausend Mal gut gegangen ist – irgendwann erwischt es jeden! Eine gute Strategie ist es, einmal kurz eine geerdete Masse zu berühren, bevor man die Schaltung berührt (z.B. Erdung am RS232 Stecker oder USB Stecker des PCs, Heizkörper etc.). In der Regel reicht es, dies einmal nach dem Hinsetzen zu tun, da der Körper speziell durch Herumlaufen (Gummisohlen etc.) statisch auflädt.
- 2) Falsch programmierte Fuse: An den Fuses sollte man nicht unwissend oder mit Halbwissen herumspielen. Durch falsche Einstellungen können Sie den Mikrocontroller unzugänglich schalten – er muss dann von uns kostenpflichtig ausgetauscht werden. Wenn Sie sich über eine Fuse-Einstellung nicht sicher sind, rufen Sie doch vor einer Umstellung einfach bei uns an.

### **Einzelne Ports funktionieren nicht**

Entweder möchten Sie einen Port nutzen, der eine alternative Beschaltung hat (bitte in der Exceldatei Portnutzung prüfen) oder eine falsche Beschaltung hat den Port durchbrennen lassen.

Wenn letzteres vorliegt, ist dieser Port nicht mehr zu retten (oft ist ein als Ausgang durchgebrannter Port aber noch als Eingang zu nutzen und umgekehrt) und Sie müssen einen Alternativport nutzen – es gibt ja zum Glück genug Ports.



**Ich möchte den Realtime-Counter (z.B. für die Realisation einer internen Uhr) verwenden, aber dieser zählt nicht, bzw. der entsprechende Timer 0 bzw. Timer 2 (je nach Controller) arbeitet nicht oder nur manchmal**

- 1) Bei Vorhandensein der Echtzeituhr:
  - a. Ist Jumper J10 geschlossen?
  - b. Ist der Uhrenbaustein so programmiert, dass er ein Ausgangssignal auf der Leitung SQW/Out sendet (mit dem Oszilloskop messen)? Evtl. stimmt die programmierte Ausgangsfrequenz nicht?
- 2) Echtzeituhr vorhanden, kein Uhrenquarz am Controller verbaut, J10 geschlossen, 32 Khz Signal an SQW/Out ist vorhanden:
  - a. Controller ist ein ATmega128: diese Konfiguration funktioniert nicht
  - b. Ansonsten (Controller ist ein ATmega2561 oder AT90CAN128): wurde das Register ASSR in Ihrer Software gesetzt? Dies ist notwendig um dem Controller das Vorhandensein eines externen Takts mitzuteilen.
- 3) Keine Echtzeituhr vorhanden:
  - a. Ist überhaupt ein Uhrenquarz mit 2x Kondensatoren vorhanden (siehe Seite 31)? Jumper 10 darf dann nicht geschlossen sein !
- 4) Uhrenquarz vorhanden, Jumper J10 ist offen:
  - a. Tasterjumper TJ\_B muss umkonfiguriert werden (siehe auch Seite 31). Dies ist notwendig um den Oszillator nicht zu stören.

Hersteller:  
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](http://www.display3000.com)

Autor dieses Manuals: Peter Küsters.

© **aller Informationen: Peter Küsters**

© www.Display3000.com

## Anhang: Zusatzinfos RS232 und RS485

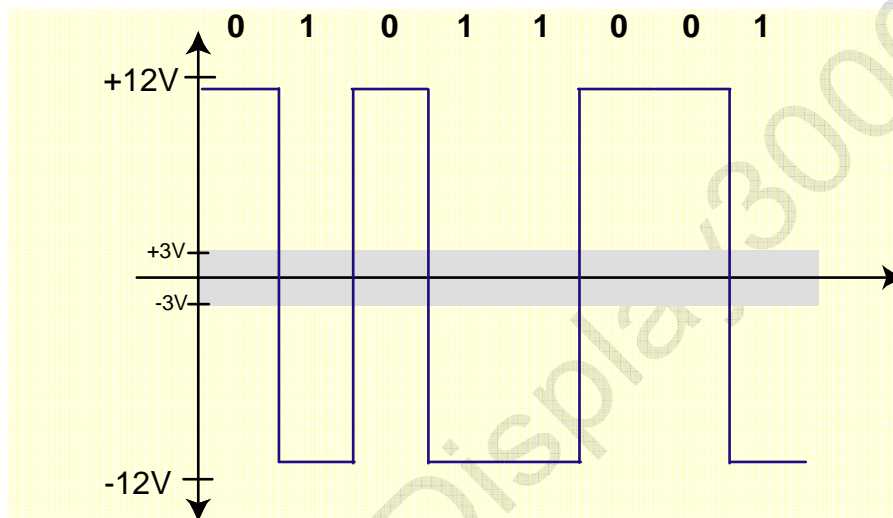
### RS232

RS-232 ist eine Kommunikationsschnittstelle welches die Daten Bit für Bit auf zwei High-Pegel-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)

Anmerkung: Der von uns genutzte Transceiver ist für einen Spannungsbereich von 3 bis 5 Volt nutzbar und erzeugt eine Spannung von +5V und -5V – was für RS232 ausreichend ist. Empfangen darf der Transceiver selbstverständlich Signale von -25V bis +25V.

Das Zeichen „Y“ (dezimal 89 – binär 01011001) wird wie folgt übertragen:



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.

Ein handelsüblicher PC arbeitet mit ca. +12 Volt und -12 Volt.

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.

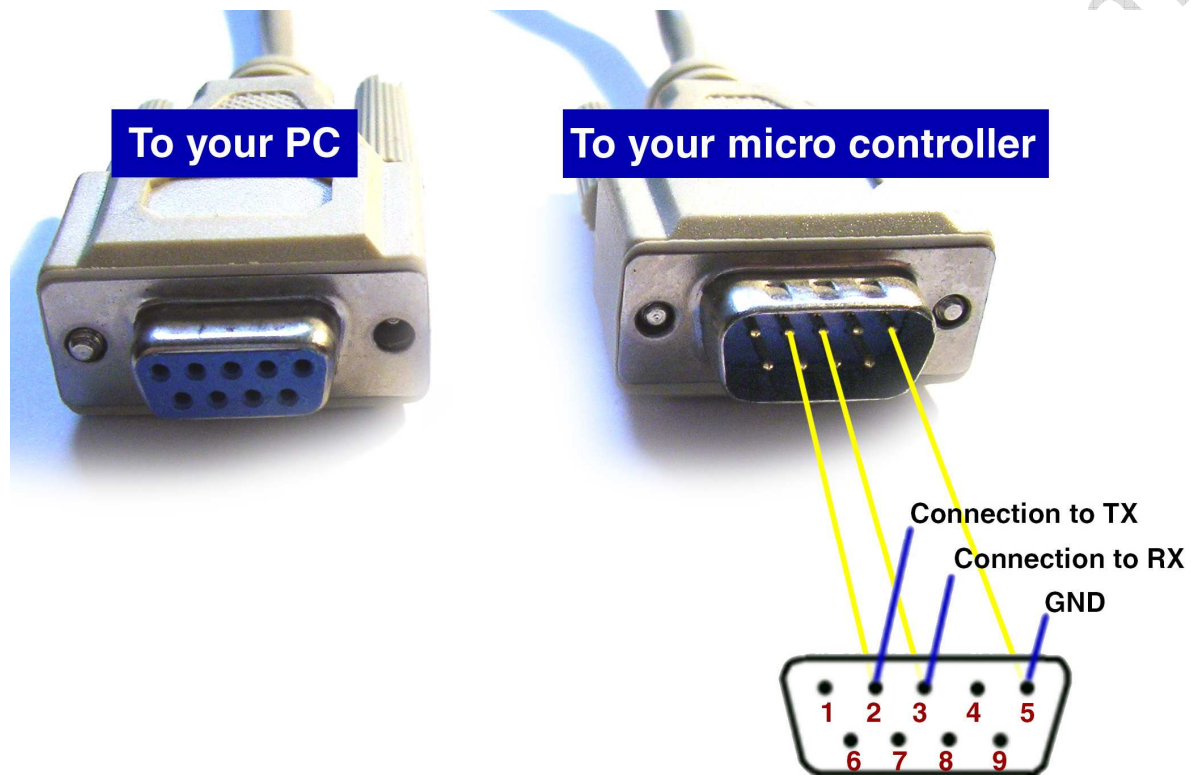
RS-232 ist ein asynchrones Protokoll, d.h. es wird keine separate Clock-Information mit übertragen. Sowohl die Sende-, wie auch die Empfangsstation müssen die Geschwindigkeit kennen (als Baud-Rate bezeichnet).

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

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

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

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



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

Der ATmega auf dem Board bietet zwei unabhängige RS-232 Interfaces. Eines davon (serielle Schnittstelle Nr. 1 – also Rx1 / Tx1) kann ohne Einschränkung genutzt werden – die Ports der seriellen Schnittstelle Nr. 0 (also Rx0 / Tx0) werden jedoch auch für die ISP-Programmierung genutzt. Bei gleichzeitigem Anschluss eines RS232 Pegelwandlers ist die ISP-Schnittstelle nicht mehr nutzbar. Die Jumper J11 und J12 legen daher fest, ob diese Ports des Controller mit einem RS232- oder RS485 Pegelwandler verbunden werden sollen oder nicht. Standard ist: Jumper offen, verbunden nur mit ISP; Jumper geschlossen: Controller-Ports verbunden mit Serieller Schnittstelle und ISP-Stecker, ISP funktioniert dann nicht mehr. Wenn Sie in einer Testphase beides öfters benötigen, bietet es sich an, die Jumper mittels eines Schalters zu Öffnen oder zu Schließen.

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

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

Unter MS-Windows® 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. Achten Sie auf die korrekten Werte des genutzten Quarzes.

```
\sample program RS232 output
$regfile = "m128def.dat"
$crystal = 16000000           'change if needed
$baud1 = 9600

Open "COM2:" For Binary As #1   'Com2: für Rx1/Tx1
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.

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

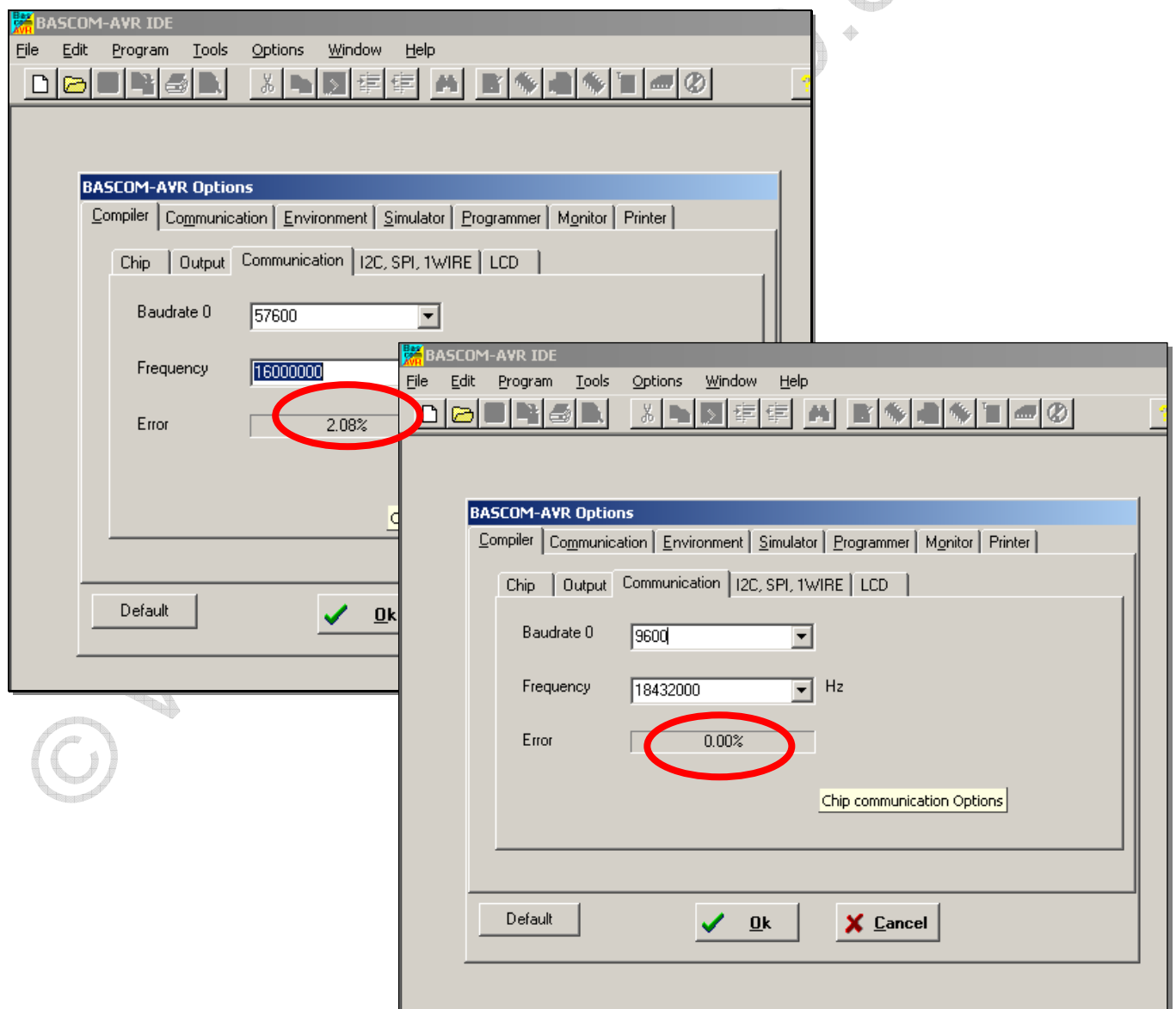
Alles über 16 MHz betreibt den Mikrocontroller jedoch über seiner Spezifikation von 16 MHz – d.h. Sie übertakten ihn. Normalerweise führt eine solch geringe Übertaktung noch zu keinem Problem, trotzdem geschieht dies immer auf eigenes Risiko. Übrigens treten evtl. Programmfehler als erstes beim Schreiben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

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

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

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

Screenshots:



### In Kürze:

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

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

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

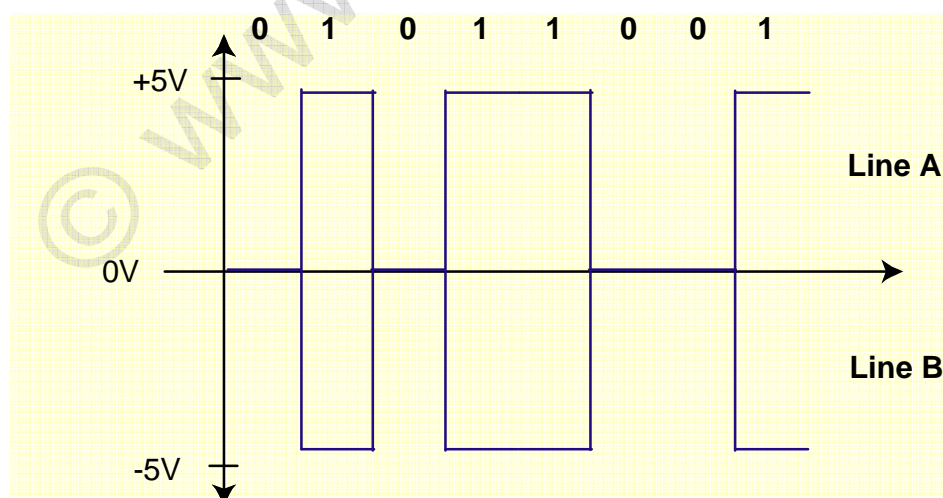
### RS-485 Interface

Das RS-485-Interface wird von Ihrer Software aus genauso wie die RS-232 Schnittstelle angesprochen. Daher gilt das oben besprochene auch ausnahmslos (mit Ausnahme der Spannungen und der Kabelverbindung) für die RS-485 Schnittstelle. Diese Schnittstelle unterscheidet sich lediglich durch das genutzte Spannungsniveau:

Oben wurde bereits erwähnt, dass RS-232 ein Kommunikationsstandard ist, welches die Daten Bit für Bit auf 2 Signal-Level sendet:

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

Die RS-232-Verbindung hat eine Masseleitung und eine Datenleitung, welche Spannungen zwischen -25V und +25V gegen Masse aufweisen kann. Der Nachteil bei dieser Signalübertragung ist die Anfälligkeit für Störungen. Ein Störimpuls ist bei einer langen Leitung schnell eingefangen, wird als Bit erkannt und führt zu einer fehlerhaften Übertragung. Mit RS-485 wurde eine andere Übertragungsart gewählt. Die Datenleitungen in eine Richtung heißen hier A und B. Ein Bit wird hier durch eine Pegeländerungen auf beiden Leitungen gekennzeichnet. Bei einem Low-Signal liegt auf der Leitung kein Pegel an, bei einem High-Signal liegt auf der A-Leitung ein +5Volt-Signal, auf der B-Leitung ein -5Volt-Signal an. Da es sehr unwahrscheinlich ist, dass ein Störsignal auf einer Leitung ein positives und auf der anderen ein negatives Störmuster hinterlässt, ist diese Signalübertragung wesentlich störungsempfindlicher als eine RS232-Verbindung.



Einen Stecker-Standard wie bei RS-232 gibt es bei RS-485 nicht, hier wird einfach mit einem 2-adrigen Kabel gearbeitet (Halbduplex) oder bei Vollduplex mit einem 4-adrigen Kabel.

## Anhang: Dimmen der Displaybeleuchtung mittels PWM

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

Zuerst etwas (vereinfachte) Theorie:

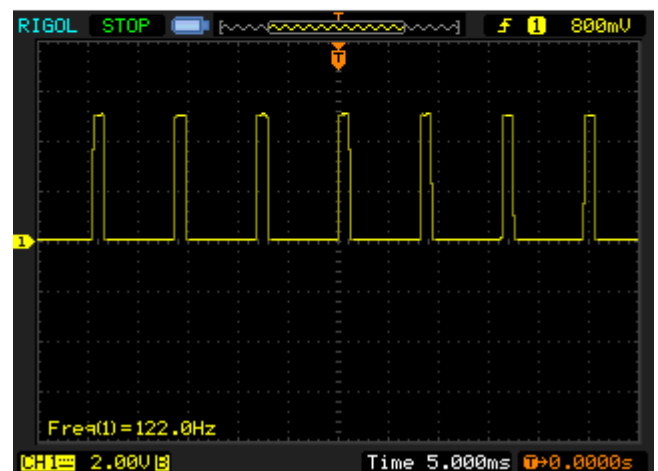
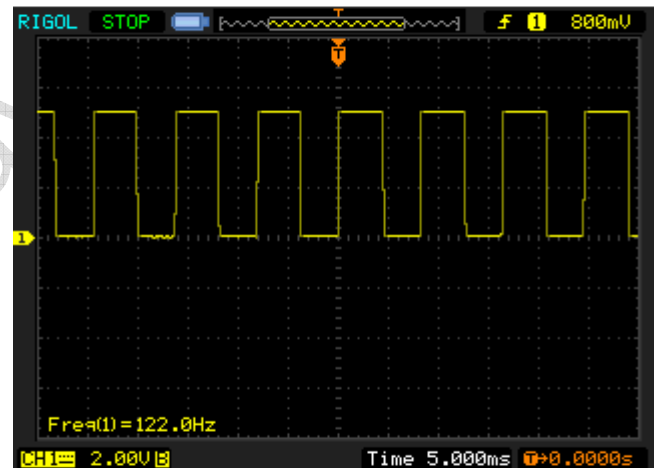
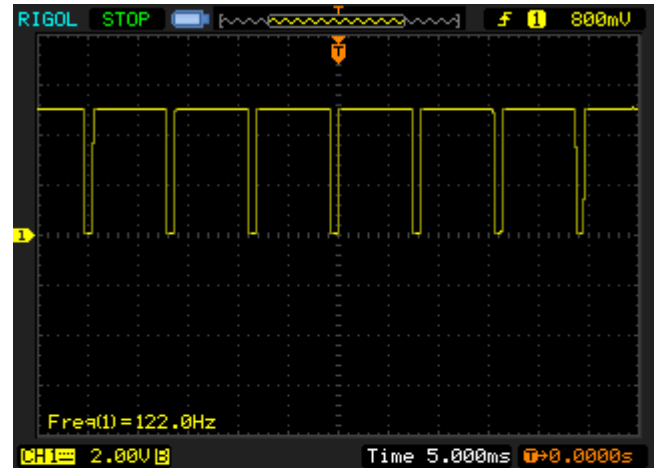
Wenn Sie z.B. eine Leuchtdiode binnen 1 Sekunde 5 Mal für 0,1 Sekunden ausschalten und dann wieder für 0,1 Sekunden einschalten, sehen Sie zuerst einmal ein nerviges Flackern. Zudem war aber die LED die Hälfte der Zeit abgeschaltet und hat daher in der Gesamtzeit auch nur die Hälfte an Lichtenergie abgegeben.

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 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. 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.

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 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. 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
    Pwmlc = I
    Waitms 10
Next I

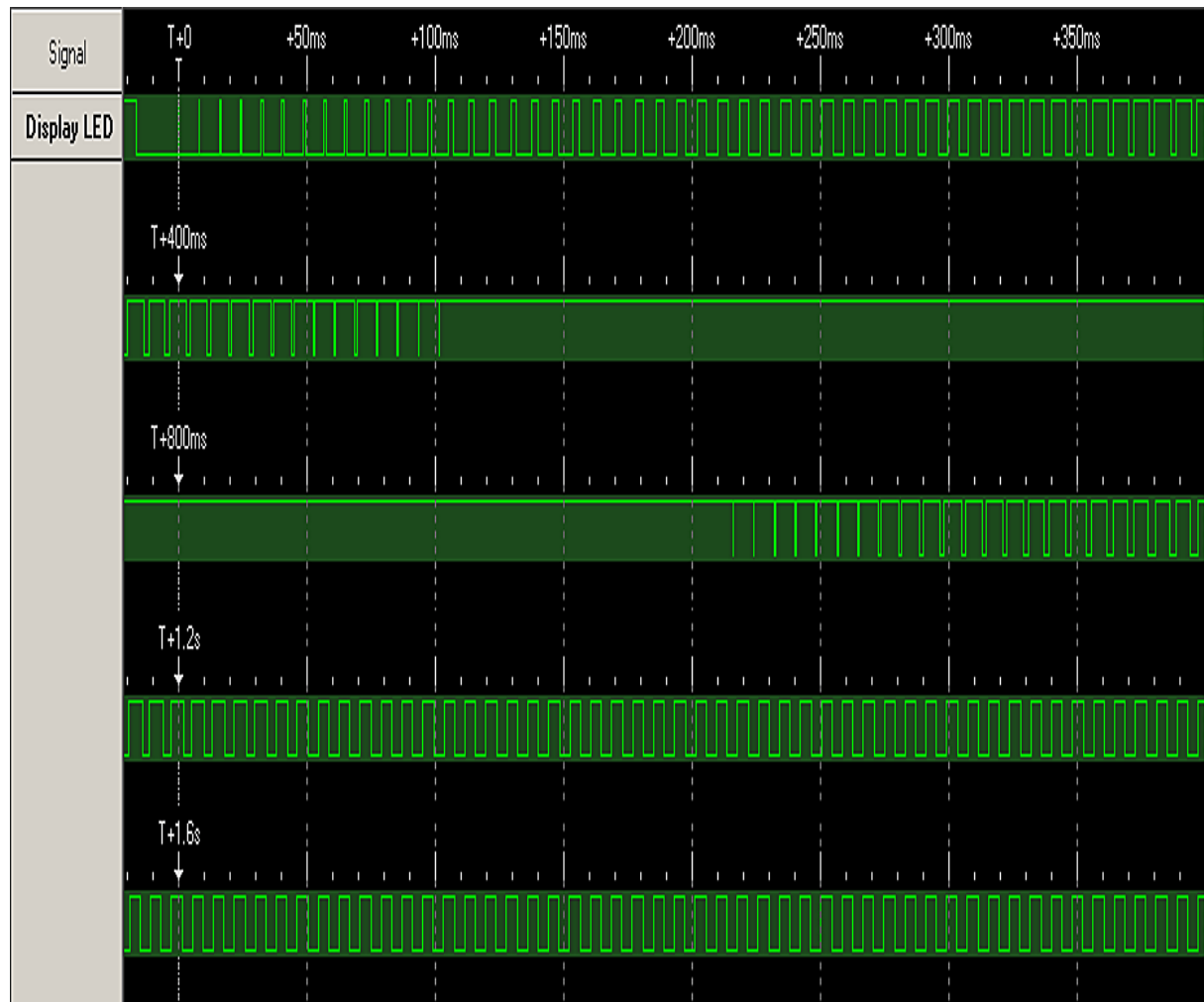
Waitms 500

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

End
```

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

Zur Verdeutlichung haben wir zeigen wir das Ausgangssignal des Mikrocontrollers noch mit einem Logikanalyzer 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 Ressourcen.

## Anhang: SD-Karte – Warum aufwändige Beschaltung?

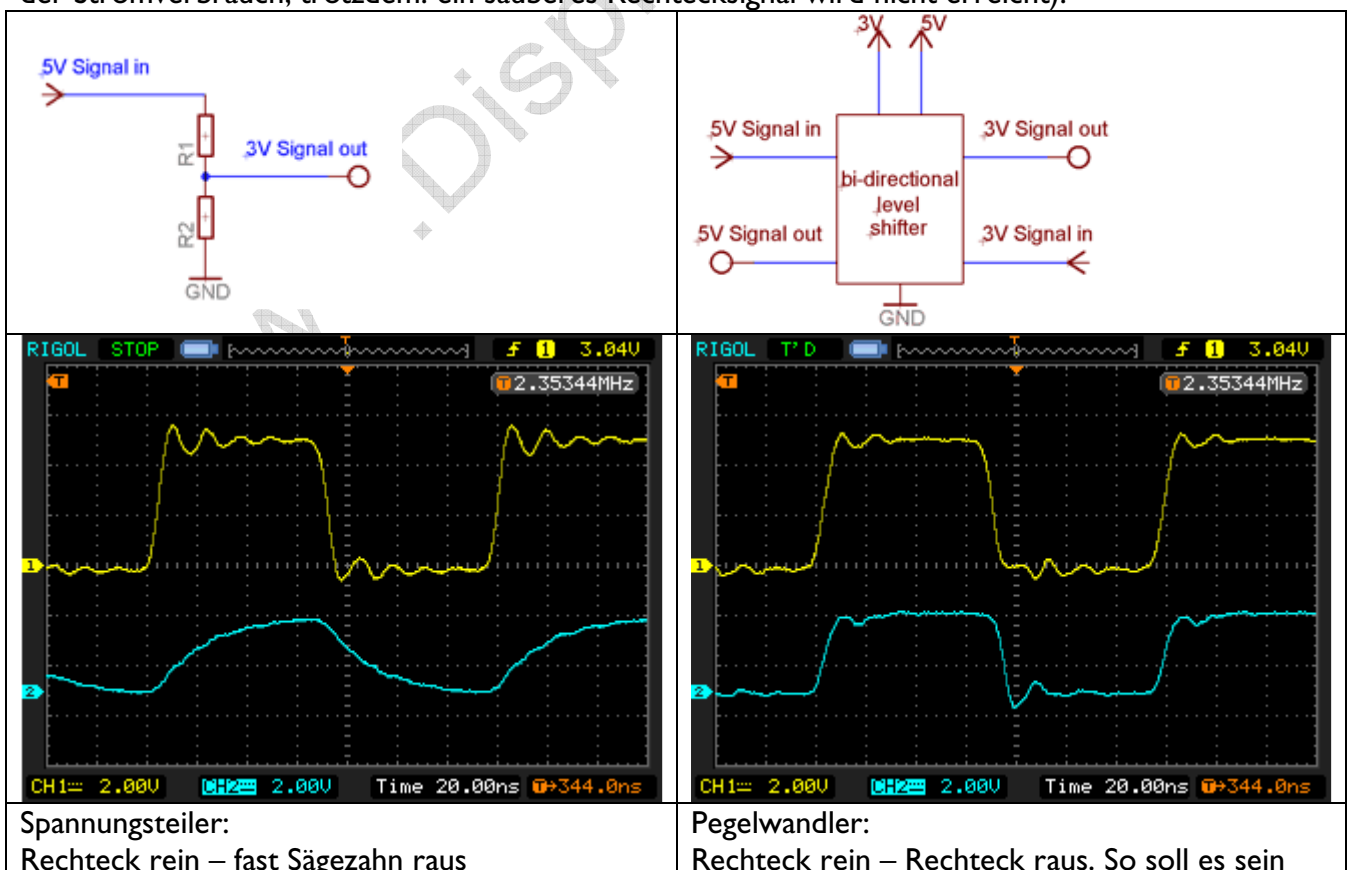
Eine SD-Karte benötigt eine Spannung von 2,7 bis 3,6 Volt – aber je nach Nutzung schwankt der Strombedarf der Karte von wenigen mA bis zu (kurzfristig) 100mA. Damit scheidet ein Spannungsteiler aus Widerständen schon aus, denn hier muss die Last schon gleich bleibend sein. Alternativ bieten sich hier evtl. mehrere Dioden an, die jeweils 0,7 Volt Spannungsabfall zeigen – aber eine professionelle Lösung ist dies nicht. Besser und langfristig sinnvoll ist also ein dezidiertes 3 Volt Spannungsregler.

### Pegelwandlung

Die direkte Verbindung zwischen eine 5 Volt Mikrocontroller und einer SD Karte verbietet sich, da diese durch den 5 Volt Pegel der Signale zerstört würde. Die Signalpegel der Datenleitung müssen also von 5 Volt auf 3 Volt umgesetzt werden. Bei einem 3 Volt Mikrocontroller könnten Sie diesen natürlich auch direkt anschließen – Sie können allerdings trotzdem unsere Karte nutzen und von den Vorteilen der Tristate-Ausgänge profitieren.

### Pegelwandlung (5V -> 3V) durch Spannungsteiler (2 Widerstände)

Hier könnte man nun meinen, dass zwischen jedem Ausgang des Mikrocontrollers und dem entsprechenden Signaleingang der Karte ein Spannungsteiler aus 2 Widerständen geschaltet werden könnte. Nun, bei langsamen Signalen ist dagegen auch nichts einzuwenden, aber sobald die Daten etwas schneller fließen sollen, wird es kritisch. Das nachfolgende Bild zeigt den Signalpegel hinter dem Spannungsteiler (so sieht dann also das 3 Volt Signal aus, das die Karte empfängt) – von einem Rechtecksignal ist hier keine Rede mehr (hier 2 Widerstände 5,6K und 3,9K – mit geringeren Werten wird das Signal ein wenig besser, dafür steigt jedoch der Stromverbrauch; trotzdem: ein sauberes Rechtecksignal wird nicht erreicht).

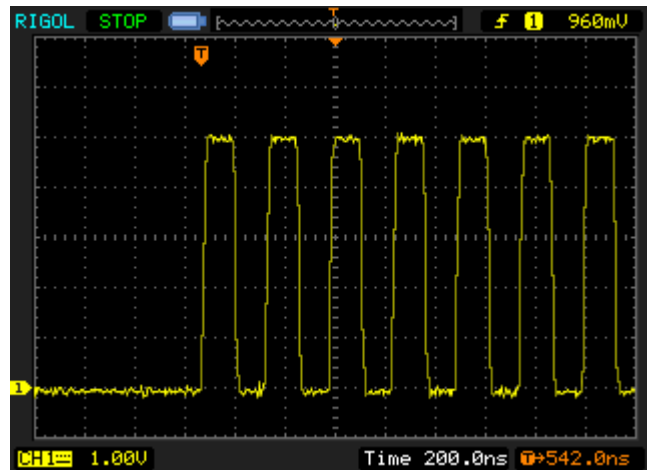


## Pegelwandlung durch Transistoren

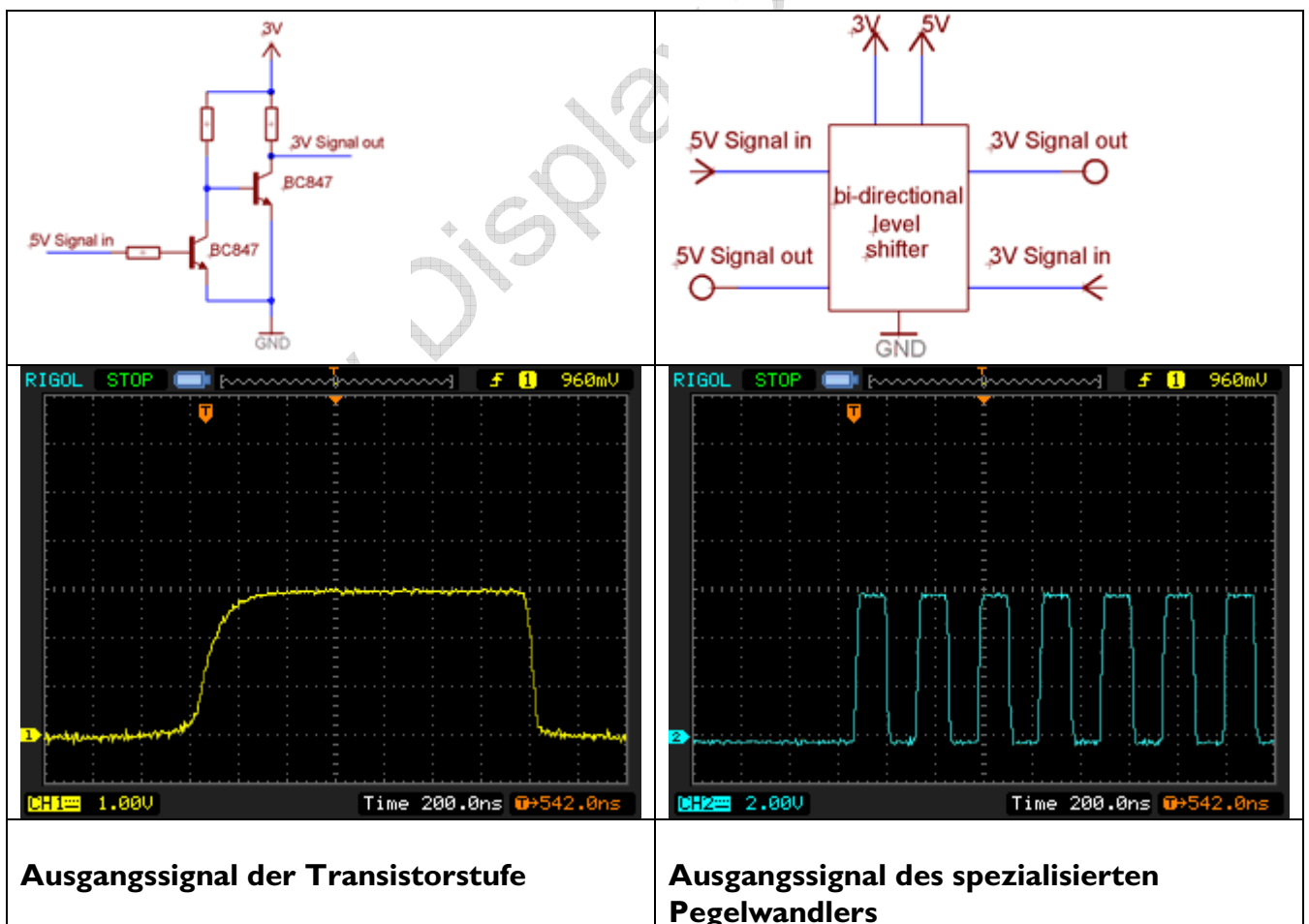
Jetzt wenden Sie vielleicht ein, dass man ja auch mittels zweier Transistoren den Pegel von 5 Volt auf 3 Volt senken kann. Das stimmt wohl, aber sie werden Schwierigkeiten haben, mit Transistoren einen Pegelwandler aufzubauen, der einen schnellen Datenverkehr zulässt. Die im Internet kursierende Schaltung mit BC547 / BC847 erlaubt dies sicher nicht.

Das nebenstehende Bild zeigt das Eingangssignal. Unten dann sehen Sie Ausgangssignal des Transistorpegelwandlers sowie des spezialisierten Pegelwandlers.

**Deutlich erkennbar: Transistoren sind einfach nicht schnell genug für schnelle Pegeländerungen und damit für einen Pegelwandler unbrauchbar (statt der Clock-Signale kommt nur ein einzelnes langes Signal an, da die Transistoren nicht schnell genug schalten).**



Eingangssignal 5V



### **Pegelwandlung in die umgekehrte Richtung (von 3 Volt auf 5 Volt)**

Die SD Karte empfängt ja nicht nur Daten, sie soll sie auch an den Mikrocontroller senden. Da die Karte nur mit 3 Volt betrieben wird, beträgt natürlich auch das Ausgangssignal der Karte 3 Volt. Theoretisch könnte man nun diese Datenleitung direkt an den Controller anschließen, 3 Volt reichen in der Regel aus, um als High-Signal erkannt zu werden.

Aber Achtung – Böse Falle! Bei einer solchen Beschaltung müssen Sie sehr vorsichtig sein. Wenn auch nur einmal an dieser Leitung 5 Volt anliegen (sei es, dass noch ein anderer Bussteilnehmer auf der Datenleitung etwas sendet, sei es, dass sie vom Controller versehentlich auf High geschaltet wird), dann liegen die 5 Volt direkt an der SD Karte an ..... und zerstören sie vermutlich.

Daher kann eine langfristige Lösung nur ein Pegelwandler sein, der sowohl die 5 Volt-Signale des Controllers in ein 3 Volt-Signal für die SD Karte wandelt, aber auch das 3 Volt Datensignal der Karte in ein 5 Volt-Signale für den Controller überträgt. Dies erledigt ein bidirektionaler Pegelwandler.

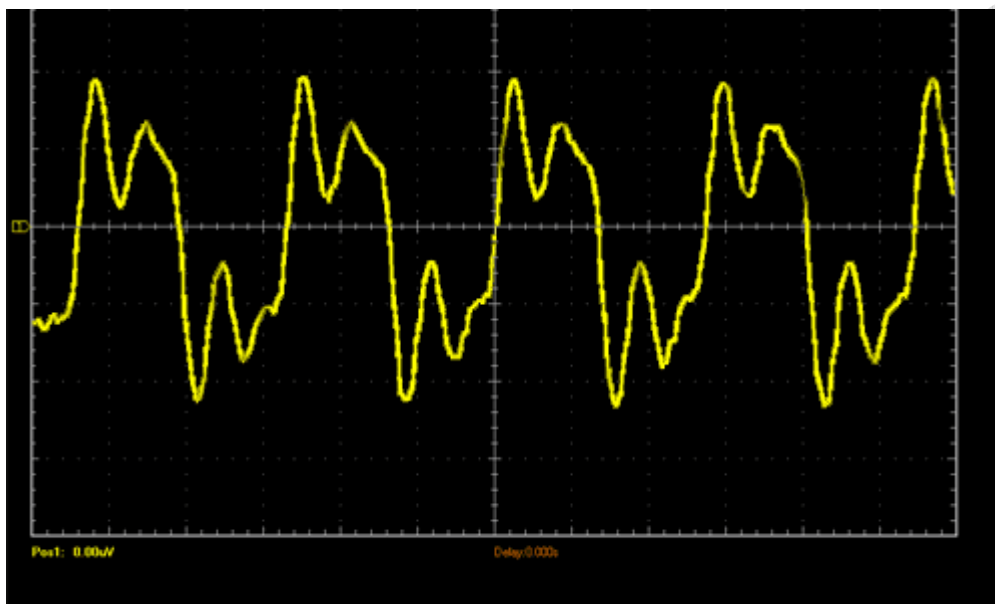
### **Tristate-Ausgänge – Wichtig !**

Die Welt besteht in der Regel nicht nur aus Mikrocontroller und SD-Karte. Meist gibt es noch andere Busteilnehmer wie z.B. ein Farbdisplay, ein Touchscreen-Controller oder andere Bausteine. Wenn ein solcher Baustein „gemein“ ist (und bei SD Karten ist dies u.U. der Fall), so behält eine Datenleitung des Teilnehmers nach der Übertragung einfach einen aktiven High-Pegel. Damit bricht der gesamte Busverkehr zusammen, denn diese Datenleitung kann nun nicht mehr von anderen Teilnehmern genutzt werden. Daher haben wir der Karte einen Pegelwandler mit Tristate-Kanälen spendiert: Durch die „Active-Leitung“ wird die Karte an den Bus komplett an- oder abgekoppelt und kann somit weder stören noch gestört werden.

## Anhang: Leitungsterminierung

Externe Beschaltungen, die bei höheren Frequenzen betrieben werden sollen, zeigen öfters Probleme beim Datenempfang. Bei dem verwendeten Controller können Sie aufgrund der Flankensteilheit u.U. nur Verbindungskabel von 10-20 cm Länge nutzen, bei Längen darüber hinaus sorgen die Reflektionen am Device-Ende dafür, dass das Signal gestört und unbrauchbar wird und nicht mehr korrekt ausgewertet werden kann. Bei langsamen Signalfrequenzen ist dies i.d.R. unproblematisch, aber bei schnell übertragenen Datenmengen gibt es dann Problem.

Das nachstehende Bild zeigt ein solches Beispiel (Ausgangssignal war ein Rechtecksignal, gemessen wurde auf Empfängerseite nach ca. 50 cm Flachbandkabel).



Statt dem einstmals sauberen Rechtecksignal wird dem Empfänger nun ein Signal überreicht, welches – nunja ... suboptimal ist. Ob hier noch eine korrekte Datenauswertung stattfinden kann ist fraglich.

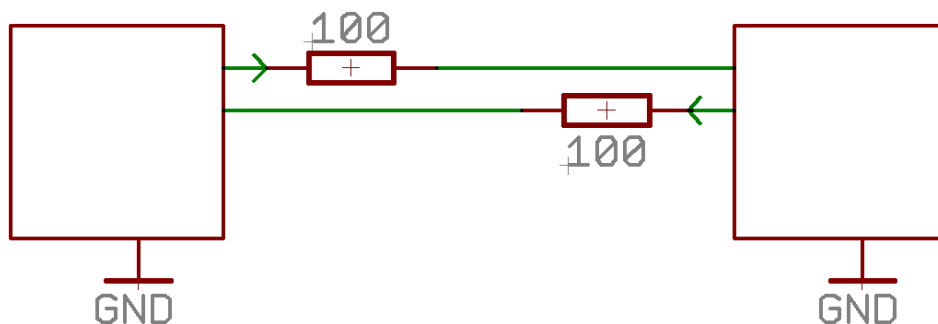
Abhilfe schaffen hier zwei Optionen

### Option 1)

Sämtliche Signale erhalten auf Senderseite einen Serienwiderstand in Höhe des Wellenwiderstands des Flachbandkabels. Idealerweise wird dies experimentell mit einem Oszilloskop ermittelt, denn jede Konfiguration und jedes Kabel benötigt einen anderen Widerstand. Der korrekte Wert wird dann in der Regel zwischen 80 und 120 Ohm liegen. Als Daumenregel kann man annehmen, dass Flachbandkabel einen Wellenwiderstand von 100 Ohm aufweist.

#### Mikrocontrollermodul

#### weiteres Modul

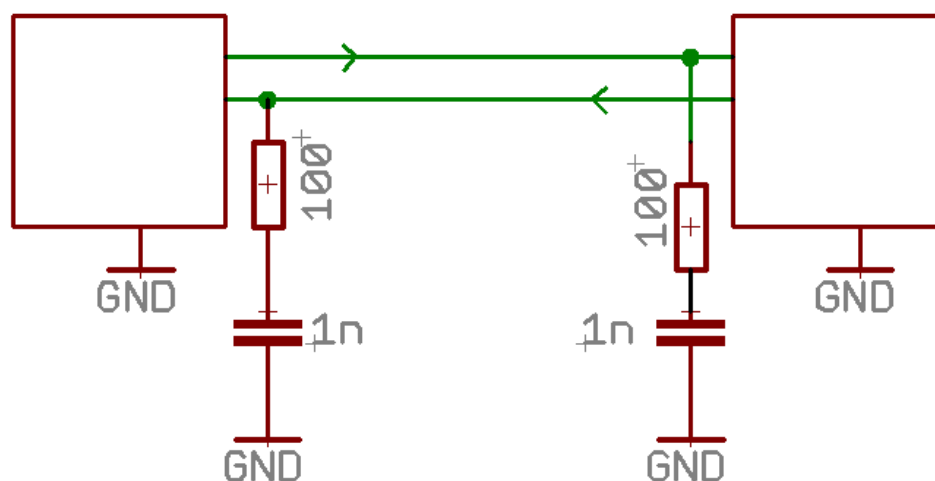


### Option 2)

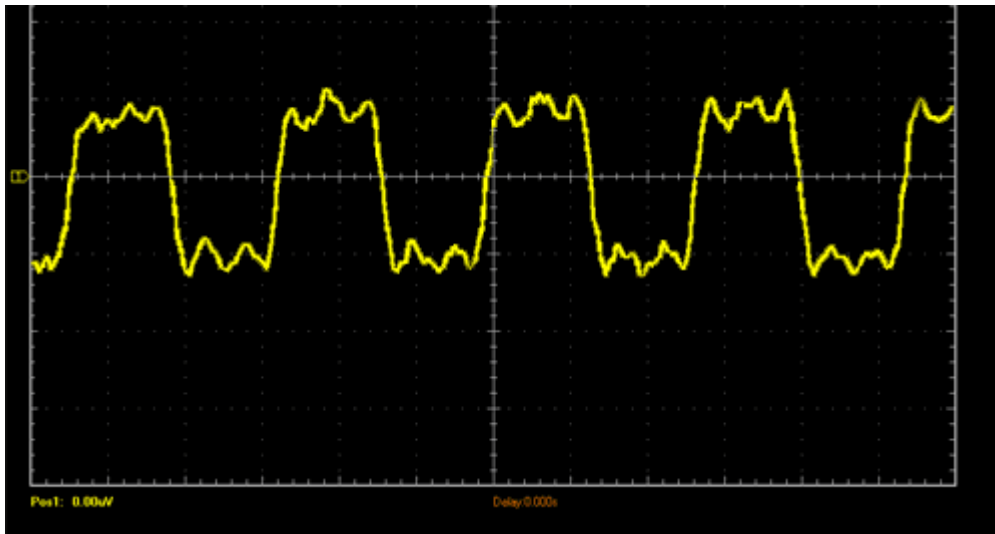
Jedes Signal wird am Device-Ende (also am Empfänger) mit einer AC Terminierung mit Widerstand und Kondensator terminiert um somit die Impedanz des Systems an das Kabel anzupassen.

#### Mikrocontrollermodul

#### weiteres Modul



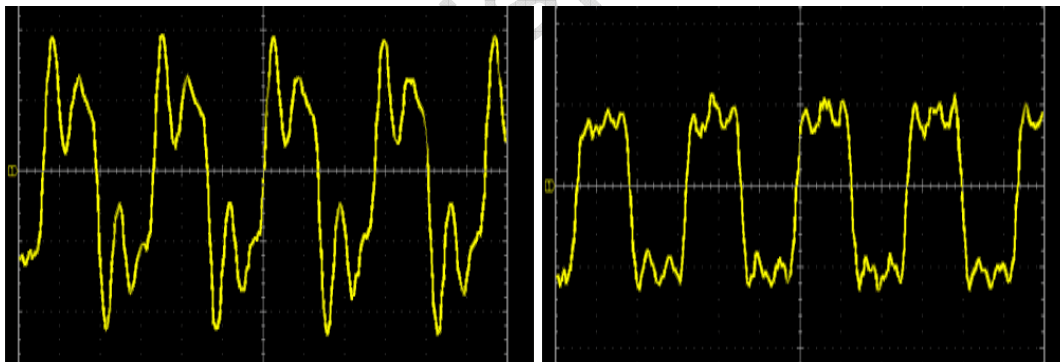
Unten erkennt man, wie gut das exakt gleiche Signal (also ebenfalls nach 50 cm Flachbandkabel) aussehen kann, wenn es mit einer Leitungsterminierung am Empfänger versehen wird. Zwar gibt es noch ein paar Einschwinger, aber die Einsen und Nullen sind klar zu erkennen und auszuwerten.



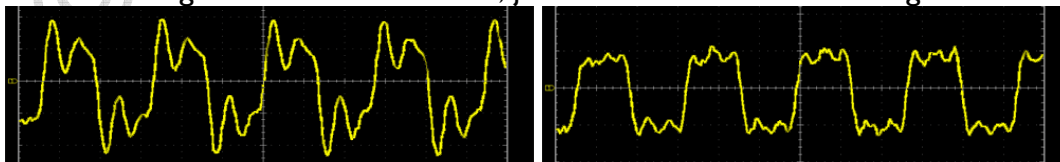
Die genauen Daten müssen je nach Umgebung experimentell ermittelt werden. Als Anhaltspunkt sind jedoch Werte von 1nF bzw. 100 Ohm gut geeignet.

Diese Terminierung auf Empfängerseite hat gegenüber den Serienwiderständen zudem noch den Vorteil, dass sie wie ein Tiefpassfilter wirkt und eingefangene Störimpulse, die ansonsten stören könnten, unterdrückt. Zudem wird das Kabel selbst weniger Störungen ausstrahlen (Stichwort EMV).

Zum Vergleich die beiden Oszilloskopbilder nochmals nebeneinander. Gezeigt wird das identische Signal auf identischem Kabel: Links ohne Terminierung, Rechts mit Terminierung.



Nochmal die gleichen Bilder wie oben, jetzt aber in anderer Skalierung:





## Wannenstecker / Pins im Standardgehäuse GBS600

Das Modul D071x wurde auf das angebotene Standardgehäuse BS600 / BS601 hin maßgeschneidert. In der Regel wird das Modul ins Gehäuse verbaut, die nach außen führenden Stecker zur Kontaktierung mit externer Peripherie werden verbaut und dann mit Kabeln an die jeweiligen Ports kontaktiert. Gelegentlich möchte man aber eventuell ein D071x Board inklusive Port-Stecker im Gehäuse verbauen. Das ist jedoch aufgrund des Platzbedarfs der Wannenstecker nicht immer möglich. Gerade bei der Version mit Batteriefach bleibt an etlichen Stellen nicht viel Platz, so dass Sie hier keine Platine mit aufgelöteten Steckern unterbringen können.

Probieren Sie die Platzverhältnisse immer vorher aus, in dem Sie die Stecker nicht festlöten und ausprobieren, ob Sie die beiden Gehäusehälften noch schließen können. Die nachfolgende Tabelle hilft vorab.

Zusammenbau als <b>Pultgehäuse</b> (=schräg; unten flach, oben höher)				
Portstecker	Ohne Batteriefach		Mit Batteriefach	
	Wannenstecker	Pins	Wannenstecker	Pins
A	(*2)		(*2)	
B				
C				
D				
E				
F	(*2)		(*2)	
JTAG				
ISP 6				
ISP 10			(*3)	
RS485,CAN,etc.			(*3)	
Zusammenbau als <b>Flachgehäuse</b> (oben und unten gleich hoch)				
Portstecker	Ohne Batteriefach		Mit Batteriefach	
	Wannenstecker	Pins	Wannenstecker	Pins
A	(*1)			
B				
C				
D				
E				
F	(*1)			
JTAG				
ISP 6				
ISP 10			(*3)	
RS485,CAN,etc.			(*3)	

\*1 = wenn am Wannenstecker eine kleine Ecke abgefeilt wird

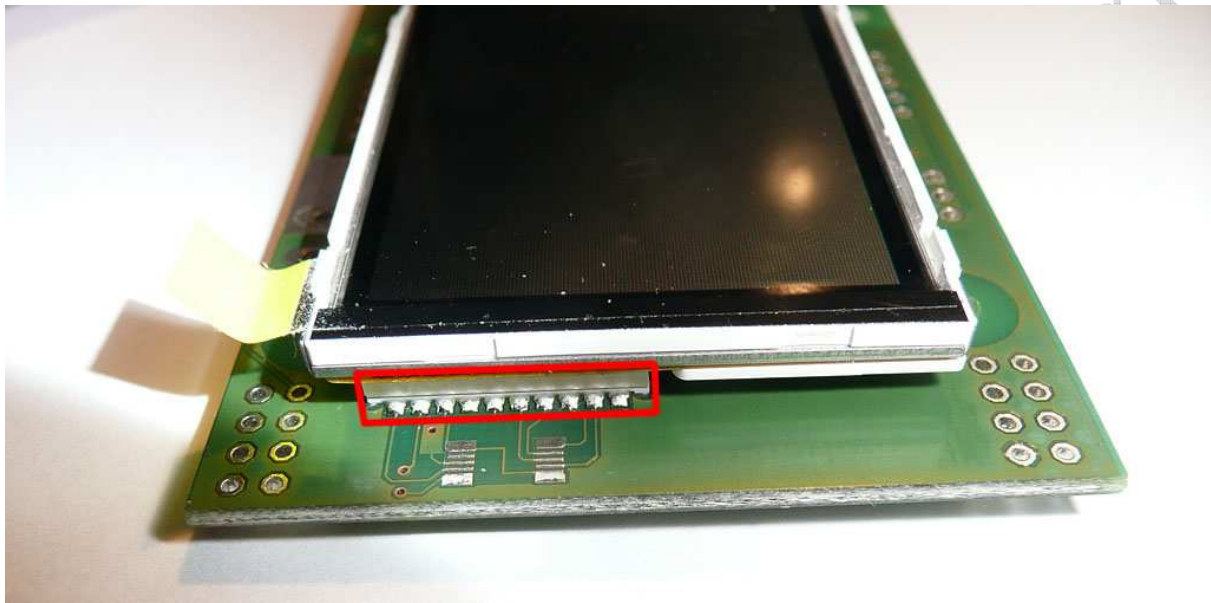
\*2 =in der Unterschale des Gehäuses müssen zwei nicht benötigte Haltebolzen mit z.B. einem Seitenschneider abgeknipst werden

\*3 = nur mit Dekordichtung

## Anhang: Abnehmen/Entfernen des Displays

Beim Abnehmen des Displays von den Modulen D071x, D074 und D012 ist etwas Sorgfalt notwendig, ansonsten kann das Display beschädigt werden.

**Hintergrund:** Das Display ist mit einem speziellen Verbindungsstecker mit der Platine verbunden. Hierzu sind auf Platine und Stecker je eine SMD Connector-Hälfte verbaut. Das nachfolgende Bild zeigt den eigentlichen Displaystecker.



**Das Problem:** Die auf der Platine angebrachte Steckerhälfte ist unkritisch, jedoch ist das Gegenstück des Displays auf einer flexiblen Leiterbahn angebracht. Beim unvorsichtigen Abziehen des Displays gibt die Leiterbahn teilweise nach. Das hat zur Folge, dass der Stecker beim Abziehen nicht gleichmäßig belastet wird und verkantet. Durch das Verkanten können sich die beiden Steckerhälften nicht ordnungsgemäß voneinander lösen und Sie wenden mehr Kraft als notwendig auf ... und reißen u.U. schlussendlich die Stecker-Hälfte vom Folienkabel des Displays ab.

### Wie ist es richtig?

Die beiden Steckerhälften zwischen Platine und Display müssen ziemlich genau im 180° Winkel geschlossen und geöffnet werden. Ein Verkanten beim Schließen (zusammendrücken) oder beim Öffnen kann ihn beschädigen.

**Das Schließen** der beiden Hälften ist einfach: achten Sie darauf, dass beide Steckerhälften plan aufeinander liegen und drücken dann Display und Platine mit Zeigefinger und Daumen zusammen, bis der Stecker geschlossen ist.

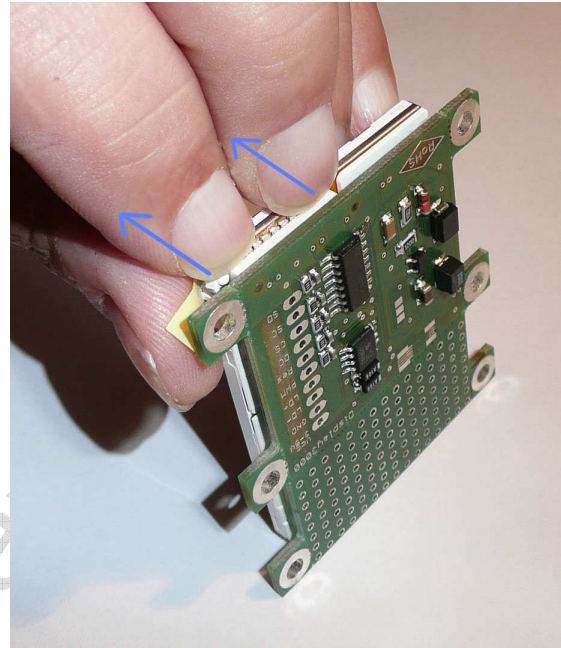
### Das Öffnen des Displaysteckers

Das Öffnen des Displaysteckers ist etwas trickreicher, denn die oben angesprochenen problematischen Aspekte gilt es zu vermeiden:

Nehmen Sie am besten Ihre Fingernägel zur Hilfe. Sie brauchen hierfür nur zwei Finger. Zwei! Mit nur einem Fingernagel üben Sie die Kraft nur punktuell aus und werden die Verbindung u.U. beschädigen. Die nachfolgenden Abbildungen zeigen wie es geht.

Beim ersten Mal sollten Sie dabei das Modul in die linke Hand nehmen und so halten, dass das Display senkrecht steht und das Display zu Ihnen hin zeigt. Der Displaystecker befindet sich dann oben rechts.

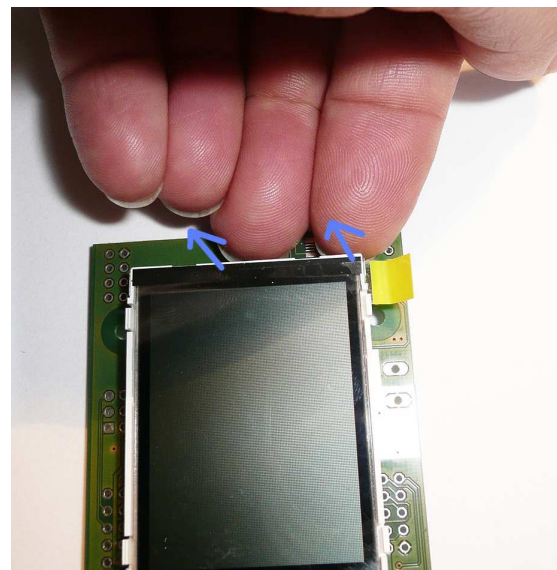
Zum Entfernen des Displays nehmen Sie Zeigefinger und Mittelfinger der rechten Hand und schieben die Spitzen der Fingernägel der beiden Finger bis an den Kunststoffkörper des Steckers; die Fingernägel **müssen** den Kunststoff berühren (siehe nebenstehendes Foto).



Nun ziehen Sie das Display einfach mit diesen beiden **Fingernägeln** zu sich heran (die linke Hand hält das Modul fest). Achten Sie darauf, dass Sie nicht mit den Fingerkuppen am Gehäuse des Displays ziehen – nur die auf den Löt-punkten aufliegenden Fingernägel dürfen den Zug ausüben. Die beiden Fingernägel sorgen so zum einen für einen gleichmäßigen Zug und verhindern das übermäßige Verkanten des Steckers und drücken zudem beim Abziehen die Lötkontakte des Steckers auf das Folienkabel des Displays und verhindern das evtl. Abreißen.

Wenn Sie es einmal können, sind Sie auch in der Lage das Display vom liegenden Modul abzuziehen (siehe nebenstehendes Foto).

Denken Sie bitte daran: der Displaystecker ist nicht für häufige Steckvorgänge konstruiert. Nehmen Sie also bitte das Display nicht zu oft ab, um das Ausleiern zu vermeiden. Wenn ein Stecker dann zu leicht aufspringt, müssen sie ihn evtl. mit je einem Tropfen Klebstoff (z.B. Uhu) an jedem Ende fixieren.



## Anhang Errata:

### a) Joystick-Beschaltung

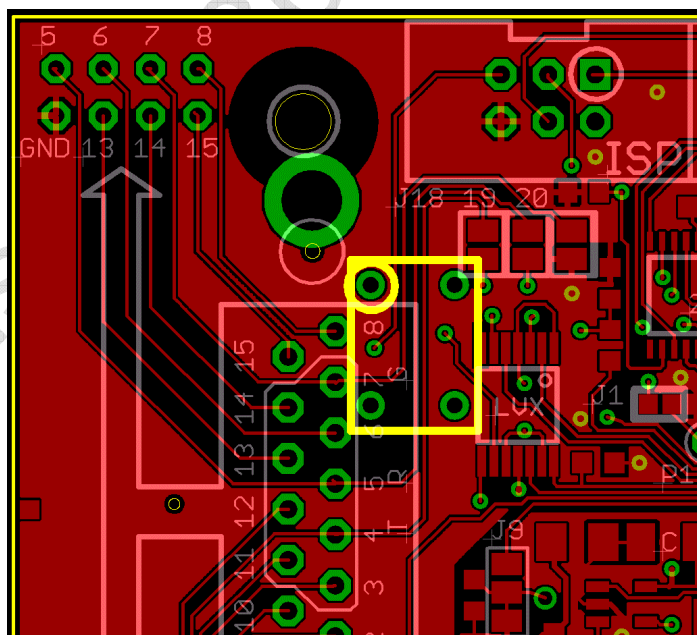
Für den Joystick wurde versehentlich ein falsches Library-Layout gewählt, die Belegung des Joysticks korrespondiert nicht mit den Tastern (d.h. *Taster hoch* ist nicht mit dem gleichen Port belegt wie *Joystick-hoch*).

In den Handbüchern Version 1.0 bis 1.1 war dies falsch abgedruckt, ab Handbuch V 1.11 ist diese abweichende Belegung dokumentiert. Die Tabelle auf Seite 28 zeigt die korrekte Belegung aller Eingabemöglichkeiten.

### b) Reset-Taster

Es gibt zwei mögliche Positionen für einen Reset-Taster: neben dem optionalen DSUB Stecker sowie am unteren Ende der Platine (neben dem Stecker für Port A). Der Reset-Taster am DSUB-Stecker ist versehentlich nicht mit Masse verbunden, d.h. er kann die Reset-Leitung nicht an Masse ziehen.

Abhilfe: Wenn nicht die alternative Position des Reset-Tasters in Frage kommt, dann müssen Sie noch den gelb eingekreisten Kontakt mit Masse verbinden. Sie können dazu ein wenig vom umgebenden Lötstopplack wegkratzen und dann den Kontakt mit Masse verbinden oder einfach nur eine kurze Leitung an den nächsten Massekontakt legen.



## **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.