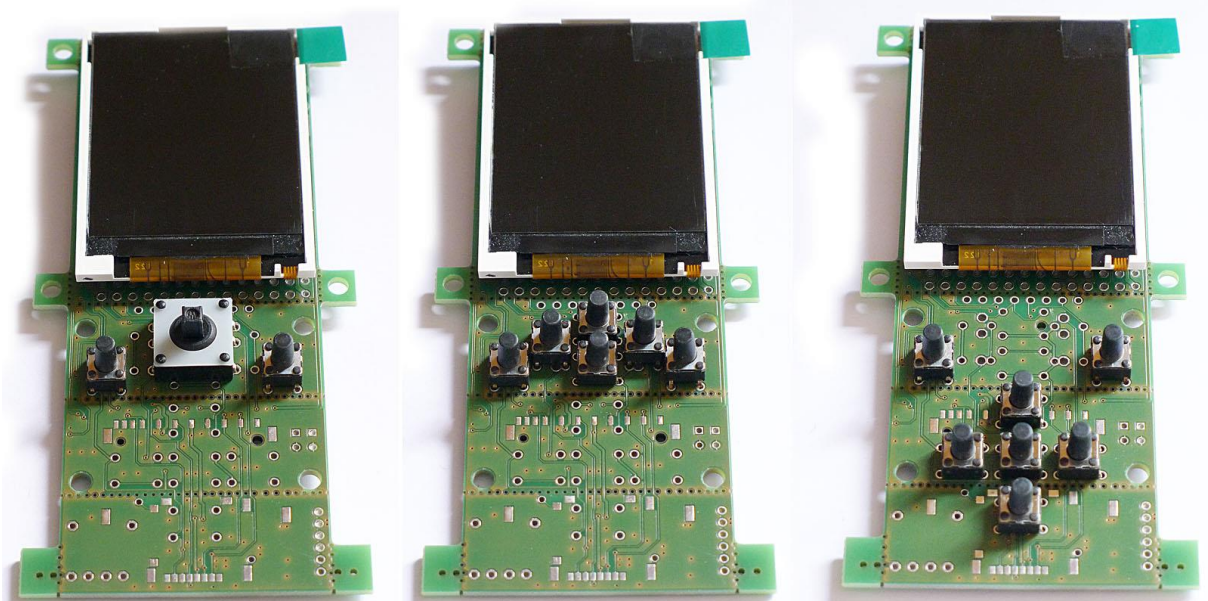


Zusatzhandbuch für das Komplettmodul D074+ mit ATMega Controller

V 0.40
6. Juli 2017



© 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
Einleitung.....	4
Optionen D074+:	4
Überblick Unterschied D074 Plus gegenüber D074 (2008 bis Mitte 2017).....	5
Optionsmatrix D074s / D074x / D074s+ / D074x+	6
Lieferumfang.....	8
Erste Inbetriebnahme	9
Quick Guide: Häufig genutzte Umkonfigurierungen	12
Anschluss Programmieradapter	13
Spannungsversorgung.....	14
Batteriebetrieb	14
USB Betrieb	14
Benötigte Spannungen	14
Spannungsregler	15
Kleiner Exkurs: Kühlflächen	Fehler! Textmarke nicht definiert.
Anschlusspads und Jumper:.....	16
Die Anschlussmöglichkeiten.....	18
Stiftleisten	18
Anschlüsse Schnittstellen (RS232, RS485, CAN).....	19
Die Eingabemöglichkeiten	20
Joystick & Taster.....	20
Folientastatur FT02.....	22
Nutzung von Matrix Tastaturen und anderen Tastaturen.....	23
Taktfrequenz / Quarze.....	24
Systemtakt.....	24
Uhrenquarz für den internen Real Time Counter	25
Schnittstellen.....	26
SPI Bus:.....	26
I ² C / TWI – Zweidraht-Interface	27
Serielle Schnittstellen.....	28
RS 232-Belegung.....	29
RS-485 Interface	31
USB	35
CAN-Bus Interface.....	37
Analoge Eingänge (Analog-Digital-Wandler).....	39
Schalten der Beleuchtung durch den Mikrocontroller	40
Schalten der Displaybeleuchtung.....	40
Stromverbrauch.....	40
Reduzierung des Stromverbrauchs	42
Abschaltung des 3V Spannungsreglers	42
Abschaltung durch MOSFet	43
Echtzeituhr.....	45
SD Kartenmodul.....	48
Wichtiges zur Vermeidung von Datenverlust:.....	48
Warum ein bidirektionaler Pegelwandler mit Tristate Outputs?	49
Beschaltung SD Kartenmodul	50
Alternative SD Kartenpositionen.....	50

Konflikt bei Konfiguration des SPI Bus	51
Formatierung der SD Karte	52
Einsatz im Gehäuse GBS400 / GBS401	53
Übersicht Jumper / Lötbrücken D074+	56
Übersicht Portbelegung	56
Technische Daten Display-Modul-Bausätze:	57
Maßzeichnung.....	59
Oft gestellte Fragen (FAQ):	60
Ich möchte das Board in ein eigenes Gehäuse bauen. Dazu müsste das Display versetzt betrieben werden. Wie kann ich das bewerkstelligen?.....	60
Ich kann die Ports F4 bis F7 nicht benutzen.....	60
Ich kann die Port D0 und D1 nicht benutzen	60
Ich kann die Ports D5 und D6 nicht benutzen	60
Ich habe Probleme bei der Nutzung der Ports E0 und E1	60
Ich kann nicht auf die Echtzeituhr zugreifen	60
Ich kann nicht auf die SD Karte zugreifen.....	60
Ich kann keine Analogmessungen durchführen.....	60
Ich messe an den Rx / Tx Leitungen von RS232 keine Spannung.....	60
Ich habe Probleme bei der Nutzung von RS232.....	61
Ich kann CAN nicht nutzen.....	61
Das Board resettet sich nach einer Weile.....	61
Das Display „spinnt“ gelegentlich	61
Ich kann plötzlich nicht mehr auf den Controller zugreifen	61
Einzelne Ports funktionieren nicht.....	61
Probleme mit Timer 0 bzw. Timer 2	62
Beim Einsatz im Gehäuse BS400 sitzt die Platine irgendwie schief.	62
Anhang:.....	64
Verkleinern des D074.....	64
Zusatzinfos RS232 und RS485	66
Dimmen der Displaybeleuchtung mittels PWM	71
USB Treiber	75
Nutzung des SD Kartenmoduls.....	78
Zugriff auf die Karte mit Bascom Basic	78
Schnellstart mit Bascom	79
Zugriff mit C (WinAVR).....	81
Nutzung der SD Karte ab Ablage für Grafikdaten zusammen mit unseren Farb-TFTs ..	82
Links:	83
SD-Karte – Warum aufwändige Beschaltung?.....	84
Leitungsterminierung	87
Standardgehäuse GBS400	90
Gehäusevarianten.....	90
Der Einbau der D074 Platine inkl. Stecker / Pins	91
Abnehmen/Entfernen des Displays	92
Das Schließen des Displaysteckers.....	92
Das Öffnen des Displaysteckers	93
Versetzen des Displays / Verlängerung Displayleitung.....	94
Errata:.....	95
Haftung, EMV-Konformität	96

Einleitung

Herzlichen Glückwunsch zum Erwerb des D074+ (Plus) Moduls inklusive 2“ TFT-Farbdisplay.

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

Zusammen mit dem von uns gefrästen BS400 Gehäuse sowie der von uns entwickelten Standard-Folientastatur erhalten Sie schnell und kostengünstig ein selbst als Prototyp professionell wirkendes Gerät (siehe Foto rechts). Für Kleinserien oder Prototypen ist eine eigene Entwicklung ansonsten unerschwinglich.

Ab einer Stückzahl von 50 Platinen (darunter wäre es wirtschaftlich vermutlich unsinnig) ist übrigens auch eine komplett an Ihre Bedürfnisse hin abgewandelte Version der D074+ Platine oder auch einer Folientastatur mit abweichender Bedruckung möglich. Sprechen Sie uns bei Interesse an. Wir entwickeln und produzieren seit Jahren elektronische Schaltungen aller Art für Kunden aus allen Branchen.



Optionen D074+:

Je nach bestellter Ausstattung kann das Modul folgende Optionen beinhalten:

- Mikrocontroller mit 128 / 256 KByte Programmspeicher, 4/8 KByte RAM
- Helles IPS Farb-TFT mit 220x176 Pixeln Auflösung
- 1x RS232 (2x UART)
- 1 x RS485 halbduplex
- USB Anschluss
- CAN BUS
- I²C BUS
- SPI BUS
- Batteriegepufferte Echtzeituhr
- 2x MOSFet zur Schaltung höherer Lasten (bis 2 A)
- Abschaltung aller Komponenten auf dem Board (außer dem Mikrocontroller) per MOSFet um einen möglichst stromsparenden Batteriebetrieb zu gewährleisten
- Separater SMD Uhrenquarz für TOSC1/TOSC2
- Externer SMD Quarz mit 14,7456 Mhz oder 16 Mhz (auf Wunsch auch andere)
- SD Karten-Connector mit entsprechender Anschlusselektronik
- Spannungsregler für den Anschluss von Spannungen bis 18 Volt
- Steuerung über Taster oder 5-Wege Joystick
- Elegantes Handgehäuse und Folientastatur

Achtung: Dieses Handbuch ist eine Vorabversion und enthält an einigen wenigen Stellen noch alte Fotos oder noch zu bearbeitende Textelemente. **Diese noch zu bearbeitenden Bereiche wurden in dieser PDF gelb markiert.** Beachten Sie sie bitte mit Vorsicht, da die dort getroffenen Aussagen u.U. nicht oder nur teilweise beim neuen D074+ zutreffen.

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

Überblick Unterschied D074 Plus gegenüber D074 (2008 bis Mitte 2017)

Da wir das bisherige Display nicht mehr beziehen konnten, mussten wir eine neue Lösung entwickeln. Ab Mai 2017 ersetzt das neue Modul D074+ das bisherige Modell D074 (2010 bis 2017). In Einzelstücken (z.B. zur Ersatzteilversorgung) ist das alte Modul noch erhältlich – bitte sprechen Sie uns darauf an, wenn Sie dieses noch benötigen.

Wir haben uns entschieden, für das neue Display ein neues D074 Modul zu entwickeln, welches elektrisch und mechanisch identisch zum alten Modul bleibt. Für die Anwender unserer bisherigen D074-Lösung bieten wir so Kontinuität, Investitionssicherheit und Zukunftssicherheit. Im Einzelnen (bitte beachten Sie auch die Optionsmatrix auf der nächsten Seite):

- Die Maße und Bohrlöcher bleiben unverändert.
 - Die Belegung und Position der Steckerleisten ist ebenfalls unverändert, d.h. bisher genutzte Huckepackplatinen / Zwischenplatinen können weiter eingesetzt werden.
 - Die integrierte Elektronik wurde wo es sinnvoll war modernisiert und verkleinert, aber die Portbelegung und somit die Software-Ansteuerung ist identisch geblieben.
 - Das Display hat fast die gleichen Maße, bietet aber nun eine höhere Auflösung von 176x200 Pixeln.
 - Das neue Display ist ein IPS Display, es ist (wenn notwendig – sonst dimmbar) um ein mehrfaches heller als das alte Display und wesentlich brillanter in der Farbdarstellung.
 - Die Displaylibrary wurde dahingehen angepasst, das das neue Display von Ihrer vorhandenen Software direkt weitergenutzt werden kann. Durch die höhere Auflösung des neuen Displays, werden Sie jedoch die X- und Y-Positionen in der Software sowie die Bitmap-Auflösung von evtl. genutzten Bitmaps (Icons etc.) anpassen müssen, wenn Sie den gesamten Bildschirmbereich ausnutzen möchten.
 - Fast sämtliche bisher auf dem unteren Bereich des D074x verfügbaren Optionen sind nun durch Nutzung kleinerer Bauteile auf die obere Hälfte des Moduls gewandert, d.h. das kürzere D074s kann nun ebenfalls so gut wie alle Optionen nutzen. Einzige Ausnahme: die beiden MOSFet-Schalter bleiben aus Platzgründen auf der unteren Hälfte und sind somit nur mit einem D074x+ und nicht mit einem D074s+ nutzbar.
 - Das D074s+ und D074x+ hat nun die Option eines Micro-SD Karten-Connectors am oberen Rand; Das D074x+ kann alternativ auch entweder ein Mikro-SD Karten und/oder (wie bisher) einen SD-Karten-Connector am unteren Rand der Platine erhalten.
 - USB als Option ist nun nur noch am oberen Rand nutzbar (dafür auch beim verkürzten D074s+ Modul).
 - Das neue Display besitzt den gleichen Stecker wie das alte Display, d.h. Sie können auf Wunsch auch Ihre „alten“ Module nachträglich mit dem neuen Display ausstatten um durchgehend die gleiche Software einsetzen zu können.
- Wichtig: Stecken Sie bitte das neue Display bitte nicht einfach auf ein altes D074! Sie können es verwenden, aber dazu ist zuerst eine kleine, schnell durchzuführende Hardwaremodifikation am Display notwendig. Ohne diese Modifikation beschädigen Sie das Display!** Wir werden eine kurze Anleitung erstellen, wie Sie das Display selber umrüsten können. Selbstverständlich können Sie einzelne neue Displays von uns auch inkl. dieser Modifikation (ohne Aufpreis) erhalten.
- Der eine oder andere Anwender eines D072 hätte lieber gerne die 6-Taster Lösung des D072 am Nachfolger D074 gesehen (welches hier Joystick +2 Taster angeboten hat). Mit dem neuen D074+ haben wir diese Wünsche erhört und bieten beide Optionen zur Auswahl an: 6 Taster oder Joystick + 2 Taster.

Optionsmatrix D074s / D074x / D074s+ / D074x+

Das Modul D074x (extended) besteht aus einem verlängerten D074s Modul (mehr hierzu siehe Anhang Seite 64). Die Grundbeschaltung ist identisch, das D074x ist jedoch länger und bietet dadurch weitere Optionen. Die folgende Tabelle listet die wichtigsten Unterschiede zwischen den beiden Varianten, auch im Vergleich mit den beiden alten Modulen auf:

Mögliche Option	D074s (2010-2017)	D074x (2010-2017)	D074s+ (ab 2017)	D074x+ (ab 2017)
Joystick +2 Taster, Folientastatur	✓	✓	✓	✓
6 Taster am oberen Bereich	✗	✗	✓	✓
Zusätzl. 5 Taster im unteren Platinenbereich	✗	✗	✗	✓
Displaybeleuchtung schaltbar und dimmbar	✓	✓	✓	✓
RS485	✓	✓	✓	✓
CAN-BUS	✓	✓	✓	✓
I ² C	✓	✓	✓	✓
RS232	✓	✓	✓	✓
Verschiedene Controller-Optionen; verschiedene Quarze, Uhrenquarz etc.	✓	✓	✓	✓
5V Spannungsregler	✗	✓	✓	✓
MOSFet Schaltung	✗	✓	✗	✓
Micro SD Kartenmodul am oberen Rand	✗	✗	✓	✓
Micro SD Kartenmodul am unteren Rand	✗	✗	✗	✓
SD Kartenmodul am unteren Rand	✗	✓	✗	✓
USB	✗	✓ (unten)	✓ (oben)	✓ (oben)
Echtzeituhr batteriegepuffert	✗	✓	✓	✓
6-poliger JST-Stecker zum Anschluss von An- schlussleitungen	✗	✗	✗	✓
Einbau in BS400 Gehäuse möglich	✓ (*)	✓	✓ (*)	✓
(*) nur 2 Schraubbefestigungen möglich, ein zusätzl. Klebepunkt am oberen Rand oder alternativ ein stützendes Element in der Unterschale (z.B. etwas Moosgummi) wird empfohlen				

Das Modul D074x+ kann nachträglich jederzeit auf ein D074s+ verkleinert werden (der Bereich ist vorperforiert – siehe auch Seite 64).

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 beigelegten 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 sich auf der Unterseite und am linken Rand und sich die Bauteile oberhalb befinden. Die Platine liegt also im Querformat vor Ihnen – die Beschriftungen auf der Platine sollten lesbar sein.

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. Vor allem sollten Sie nicht einfach die Pin-Reihen einlöten, sofern Sie vorhaben, das Modul in ein Gehäuse einzubauen. Lesen Sie zuerst die entsprechenden Kapitel.**

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 unser separates 75-seitiges 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.

Zu den Fotos in diesem Manual: Damit das Manual zum Produkt-Release verfügbar ist, basieren sämtliche Fotos auf einem handgelöteten Exemplar, evtl. optisch nicht perfekte Lötstellen bitten wir zu entschuldigen.

Aufgrund der unterschiedlichen Bestückungsvarianten je nach bestellter Option gibt es vom D074+ Modul über 10.000 mögliche Varianten. Wir haben das anhand älterer Jahre auch ausgewertet: in der Tat gib es so gut wie niemals eine identische doppelte Konfiguration d.h. jeder Kunde bestellt tatsächlich so individuell, dass es keine Häufung einzelner Konfigurationskombinationen gibt. Aus diesem Grunde entsprechen die Fotos natürlich auch nicht Ihrer individuellen D074+ Konfiguration, d.h. hier im Manual sind meistens mehr Optionen verbaut, als Sie sie auf Ihrem Board finden.

Dieses Handbuch zeigt i.d.R. Bilder des Moduls D074x+ (extended). Die Angaben für das Modul D074s+ sind identisch mit Ausnahme der auf der Verlängerung angebrachten Optionen.

Lieferumfang

Standard-Lieferumfang:

- 1 x bestückte Platine mit ATmega Prozessor etc.
- 1 x 2“ Farb TFT Display
- 6 x Taster
- 2-reihige Pfostenstecker (RM 2,54)
- Software, Dokumentation

Optionale Ausstattung

Standardmäßig wird das Board mit einem ATmega128A 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 SMD-Quarz mit 14,7456 MHz oder 16,0 Mhz;
- SMD-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). Die Batterie wird lose, d.h. nicht eingelötet geliefert für den Fall, dass Sie auf diesen Platinenbereich noch zugreifen müssen.
- SD Kartenelektronik und -halter (Mikro SD oder Standard-SD-Karte)
- RS232-Transceiver
- USB Anschluss
- RS485-Transceiver als 3-5V Typen
- CAN-Bus-Transceiver (zusammen mit dem Controller AT90CAN128)
- 5V Spannungsregler
- MOSFET-Abschaltungsoption für Verbraucher auf dem Board
- MOSFET-Option für eine weitere Schaltoption (z.B. zum Laden von Akkus etc.)
- Joystick
- Folientastatur
- Gehäuse
- Programmieradapter

Erste Inbetriebnahme

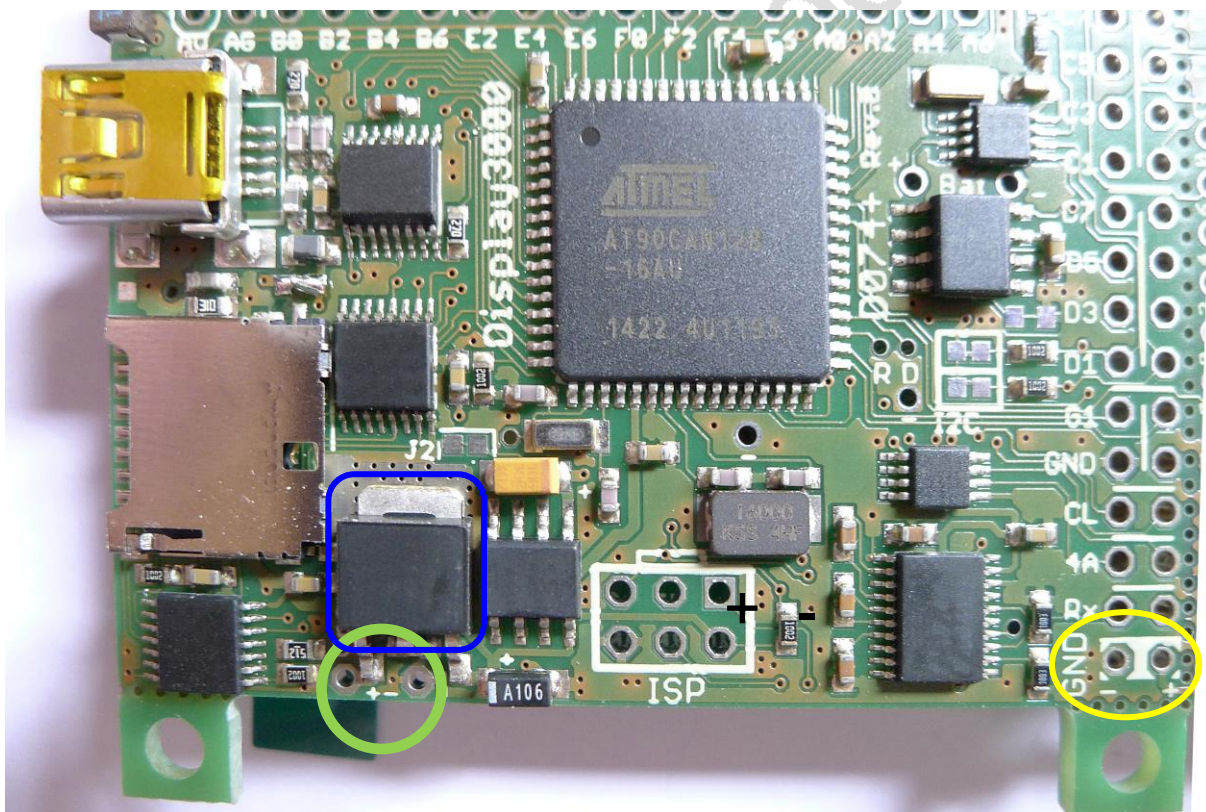
Modul ohne Spannungsregler:

Legen Sie eine Spannung von 5 Volt an die beiden mit „- 5V +“ gekennzeichneten Anschlüsse der Kontaktleiste (unten am Platinenrand, gelber Kreis). Minus bedeutet hier GND, der linke Anschluss ist somit für Masse/GND vorgesehen, der rechte Anschluss für +5V.

Modul mit 5V Spannungsregler (im Bild im blauem Rechteck):

Sofern Sie die D074 Platine inkl. des optionalen 5V Spannungsreglers erworben haben, legen Sie bitte eine Spannung von ca. 5V bis 18V an den Anschluss VCC an (großer grüner Kreis). Das linke Lötpad (mit + gekennzeichnet) erhält die positive Spannung, das rechte Pad 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.

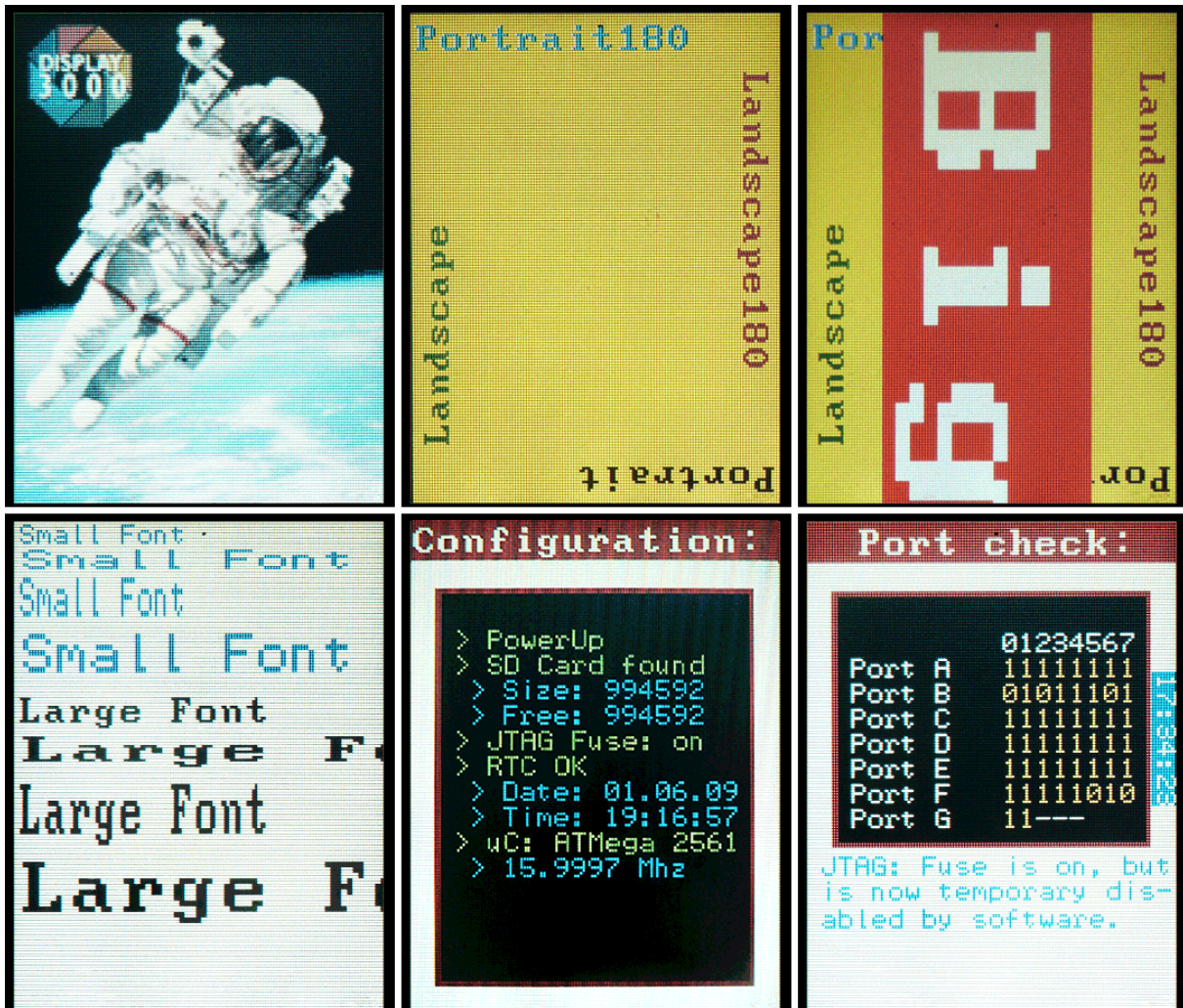


Der Mikrocontroller wurde von uns zu Testwecken bereits vorprogrammiert und zeigt Ihnen auf dem Bildschirm ein Testprogramm (Grafik, Fonts, Komponenten – siehe Foto auf der nachfolgenden Seite) an. Anmerkung: Die notwendigen Fusebits wurden von uns ebenfalls bereits gesetzt, hier sollten Sie erst einmal keine Änderungen vornehmen.

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 92).

Bilder zeigen noch das alte Display



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 Uhrzeit und das Datum der Echtzeituhr sind nicht korrekt – die Uhr muss von Ihnen noch einmalig gesetzt werden. Die Taktfrequenz des Boards wird dargestellt (ungenau da 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“ länger als 30 Sekunden 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 52).

Weiterhin wird der Status der JTAG Fuse gezeigt. Diese ist für den Ablauf des Programms nicht weiter relevant, da wir im Demoprogramm 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 und Joystick einstecken (diese halten auch ohne Löten in der Platine und haben auch oft Kontakt) 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.

© www.Display3000.com

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 in diesem Handbuch.

1) Nutzung der seriellen Schnittstelle UART_I des μ C (also Port D2/D3):

- Die UART_I kann entweder per RS485 oder USB genutzt werden.
- UART_I zu USB ist standardmäßig konfiguriert.

Anmerkung: UART₀ (E0/E1) ist standardmäßig mit dem RS232 Transceiver verbunden, Rx/Tx (RS232) ist also an den Pads RX/TX verfügbar.

Zum Umschalten der UART_I auf RS485 ist notwendig:

- Jumper J11 umkonfigurieren
- Jumper J12 umkonfigurieren

2) Stromersparung bei Batteriebetrieb:

- Siehe Seite 42
- Anmerkung: Nach einem Abschalten und Wiederaufwecken müssen Display und SD Karte neu initialisiert werden, denn sie waren ja ohne Versorgungsspannung.

3) Displaybeleuchtung schalten:

- Schalten mit Port B7; Dimmen mit PWM Kanal C (*Pwm1c*)

5) Analoge Eingänge nutzen

- AVcc und AGND sind standardmäßig mit VCC und GND verbunden (AVcc über eine 10 μ H Induktivität). Sofern Sie hier separat eine eigene (besser gefilterte) Spannung anlegen möchten, müssen die Jumper J3 und J4 geöffnet werden (siehe Seite 39)
- ARef festlegen (entweder per Software auf interne Referenzspannung oder Verbinden den Pads ARef mit der 5V Versorgung)
- 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) Nutzung des 38 Khz Takts der Echtzeituhr (sofern vorhanden) für den Atmel-internen Real Time Counter RTC

- Jumper J1 muss geschlossen werden (Seite 25 und Seite 46)

7) Verkleinern des Moduls D074x+ (Seite 64)

- Die Montage„ohren“ können durch eine Perforation leicht von Ihnen abgebrochen werden, sind aber trotzdem so stabil, dass sie das Modul sicher halten.
- Die Platine selbst ist ebenfalls an zwei Stellen vorperforiert und erlaubt somit eine Verkleinerung, wenn die Einbausituation keine andere Wahl lässt.

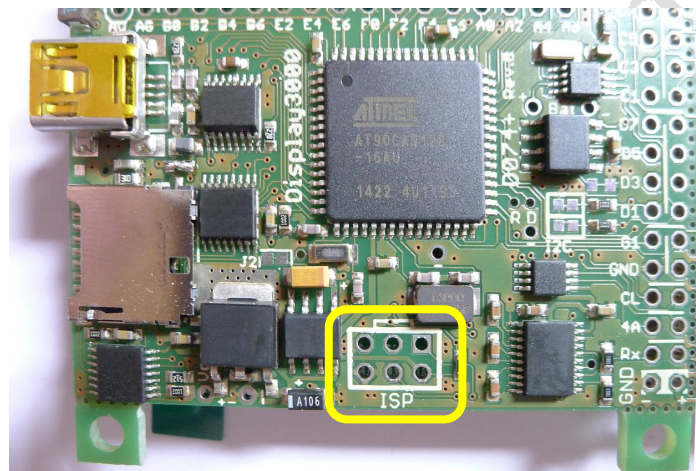
Anschluss Programmieradapter

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 und kompilierte Software in den Mikrocontroller überspielt wird.

In der Regel nutzt man hierfür einen ISP-Programmer (nicht Bestandteil des Lieferumfanges dieser Platine). Dieser ISP-Programmer verbindet Ihren PC mit diesem ATmega- Board.

Sie können auf dieser Platine NUR einen 6-poligen ISP Programmierstecker nutzen. Sofern Sie nur einen Programmieradapter mit 10-poligem Stecker besitzen, benötigen Sie noch einen Adapter (senden wir Ihnen auf Nachfrage gratis zu).

Sie können auf der Platine entweder 2x3er Pins oder einen 2x3er Wannenstecker auflöten. Die Platine ist eigentlich für 2x3 Pins vorgesehen; aber notfalls ist auch ein Wannenstecker zu nutzen. Aus Platzgründen war es jedoch leider nicht vermeidbar, dass ein Wannenstecker auf 2 Bauteilen aufliegen würde. Wenn Sie dies nicht stört, können Sie problemlos den Wannenstecker verlöten. Die Nase des Programmiersteckers muss beim Aufstecken Richtung Mikrocontroller zeigen.



Tip: Wenn Sie das Modul in Serie einsetzen möchten, so programmieren Sie es vermutlich nur einmal. Dafür können Sie sich das Einlöten des ISP Steckers sparen. Setzen Sie in Ihre ISP-Programmierbuchse am Ende des Flachbandkabels eine 2x3 Pinleiste und setzen Sie diese zum Programmieren in das D074+ ein und verkanten diese dann mit dem Daumen. So erhält der Programmieradapter Kontakt und Sie können das Board ohne Lötarbeiten am ISP Port programmieren. **Tip2:** über diese 2x3er Leitung können Sie zudem auch die Versorgungsspannung zuführen, d.h. zum Programmieren müssen Sie nur diesen einen Stecker einstecken (so programmieren wir übrigens auch alle Boards vor, die unser Haus verlassen).

WICHTIGER Tipp: Wenn Sie die Montage in einem Gehäuse vorhaben, stecken Sie vorgesehene Pins/Stecker erst einmal in die Platine ein und probieren Sie so vor! dem Einlöten, ob sich das Gehäuse noch schließen lässt.

JTAG: Wenn Sie mittels JTAG programmieren möchten, müssen Sie sich selber einen kleinen Adapter bauen. Auf der kleinen Platine war aus Platzgründen leider kein JTAG Stecker unterzubringen. Alle für JTAG notwendigen Ports (Reset, F4, F5, F6, F7, GND, 5V) stehen an den Kontaktleisten zur Verfügung. Auf Nachfrage senden wir Ihnen gerne Detailinfos zu.

Spannungsversorgung

Für alle Platinenvarianten gilt: für die Versorgung des Displays und evtl. der SD Karte ist immer ein 3,0 Volt Spannungsregler auf der Platine verbaut. Der Mikrocontroller selbst und die weitere Peripherie benötigt eine Spannung von 4,5 bis 5 Volt (bei geringerer Geschwindigkeit ist auch eine geringere Spannung möglich).

Die vorliegende Platine ist je nach Bestellung mit oder ohne einen 5V Spannungsregler ausgestattet. Sofern Sie die Platine inklusive 5V Spannungsregler bestellt haben, können Sie testweise an dem Versorgungseingang VCC eine Gleichspannung von 3,5 bis 20 Volt anlegen. Für den Normalbetrieb sollten Sie eine Spannung von mind. 5,6 Volt bis max. ca. 18 Volt zuführen. Mehr zu den Bauteilen und deren notwendiger Spannungsversorgung erfahren Sie auf der nächsten Seite.

Batteriebetrieb

Wenn Sie direkt 5,0 Volt oder weniger (z.B. bei Batteriebetrieb) zuführen möchten, so stehen Ihnen hierfür die Felder 5V/GND an der Kontaktleiste zur Verfügung. Achten Sie bitte darauf, dass Sie NIEMALS eine Spannung an diesen Feldern anlegen, wenn gleichzeitig noch über den VCC/GND Anschluss eine Spannung angelegt wird (mehr dazu weiter unten).

USB Betrieb

Beim Betrieb über den USB Anschluss erhält das Board seine 5V Versorgungsspannung über den USB Anschluss (über eine Schottky-Diode entkoppelt).

Achtung bei Akku-/Batteriebetrieb: Vermutlich haben Sie die Batterie an den Pads +- 5V angeschlossen. Dort würden jedoch beim Anschluss an USB nun 5V anliegen und einen evtl. angeschlossenen Akku beschädigen. Hier müssen Sie entweder den Akku mit einer Schottkydiode entkoppeln, so dass die 5V nicht am Akku anliegen können oder Sie trennen die 5V Versorgung von der USB Buchse auf (siehe Seite xx).

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 noch mit 16 Mhz – allerdings wird er hier außerhalb der Spezifikation betrieben und ein sicherer Betrieb in allen Temperaturbereichen wird nicht garantiert.

Die sonstigen Bauteile benötigen folgende Spannungen:

- Display: 3 Volt (3 Volt Spannungsregler befindet sich auf dem Board)
- SD Karte: 2,5 bis 3,3 Volt (3 Volt Spannungsregler befindet sich auf dem Board)
- RS232: 3,3 bis 5 Volt
- RS485: 3 bis 5 Volt
- CAN: 4,5 bis 5 Volt
- Echtzeituhr: 3 bis 5 Volt
- Booster für die Displaybeleuchtung: 2 bis 5 Volt

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

Spannungsregler

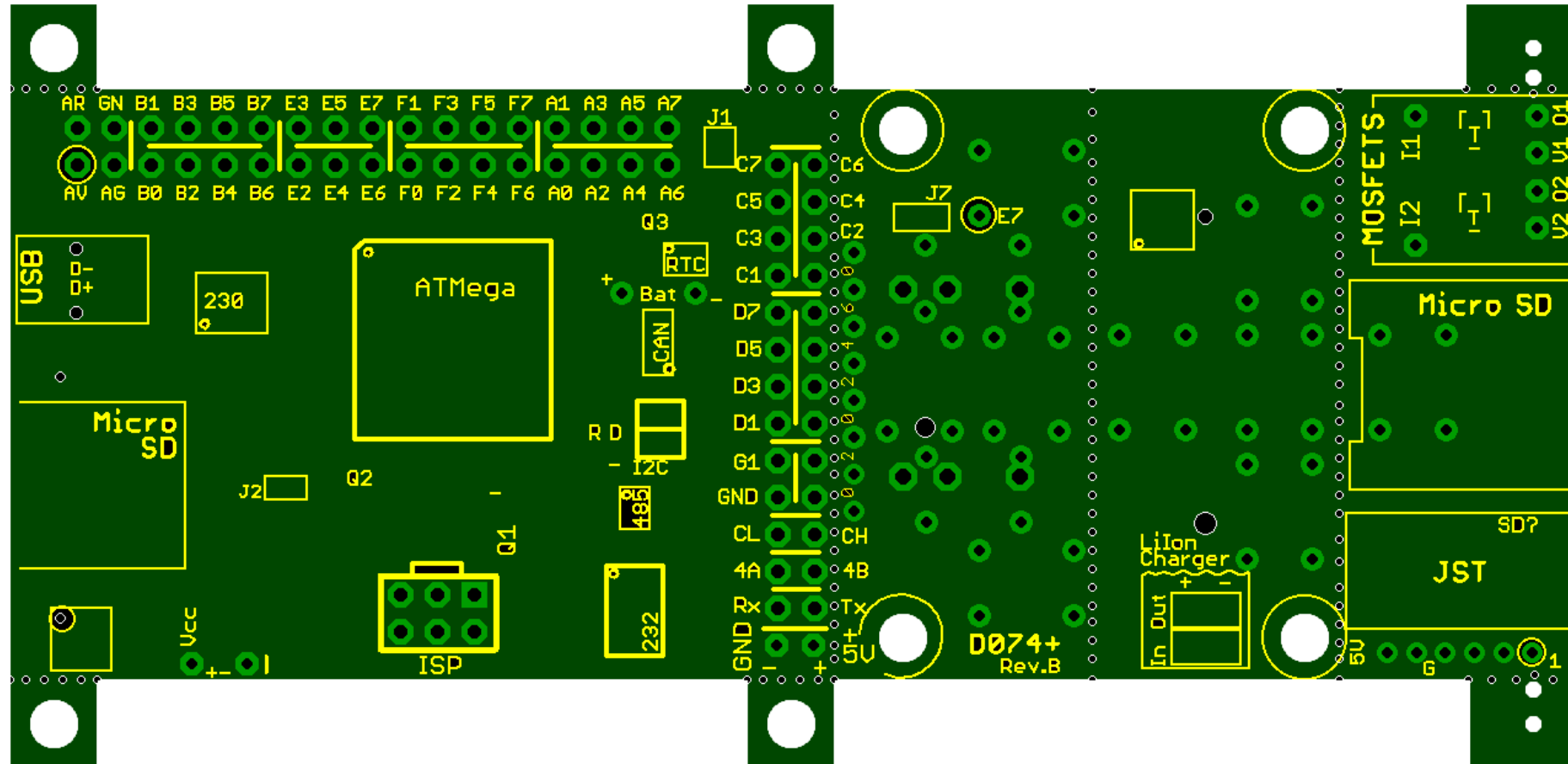
Auf der Platine nutzen wir Linearregler, die die überschüssige Spannung in Wärme verbrennen. Im Gegensatz hierzu gibt es Schaltregler, die jedoch komplexer aufzubauen sind, teurer sind, mehr Platz benötigen (den wir nicht haben) und evtl. auch Störungen verursachen.

Für das Board würde eigentlich ein „kleiner“ SMD-Spannungsregler (im Gehäuse S08) ausreichen, aufgrund der dann nicht möglichen, aber notwendigen Wärmeabfuhr, wäre allerdings nur ein Anschluss von max. 11 Volt möglich, darüber hinaus würde der Baustein zu heiß. Um das Board mit einer höheren Spannung versorgen zu können, wurde ein größerer Regler vorgesehen, der einen Teil der Platinenoberfläche als Kühlfläche nutzt. So kann das Board langfristig mit bis zu 14 Volt versorgt werden, kurzfristig und im nicht-eingebautem Zustand sind auch 18 Volt möglich.

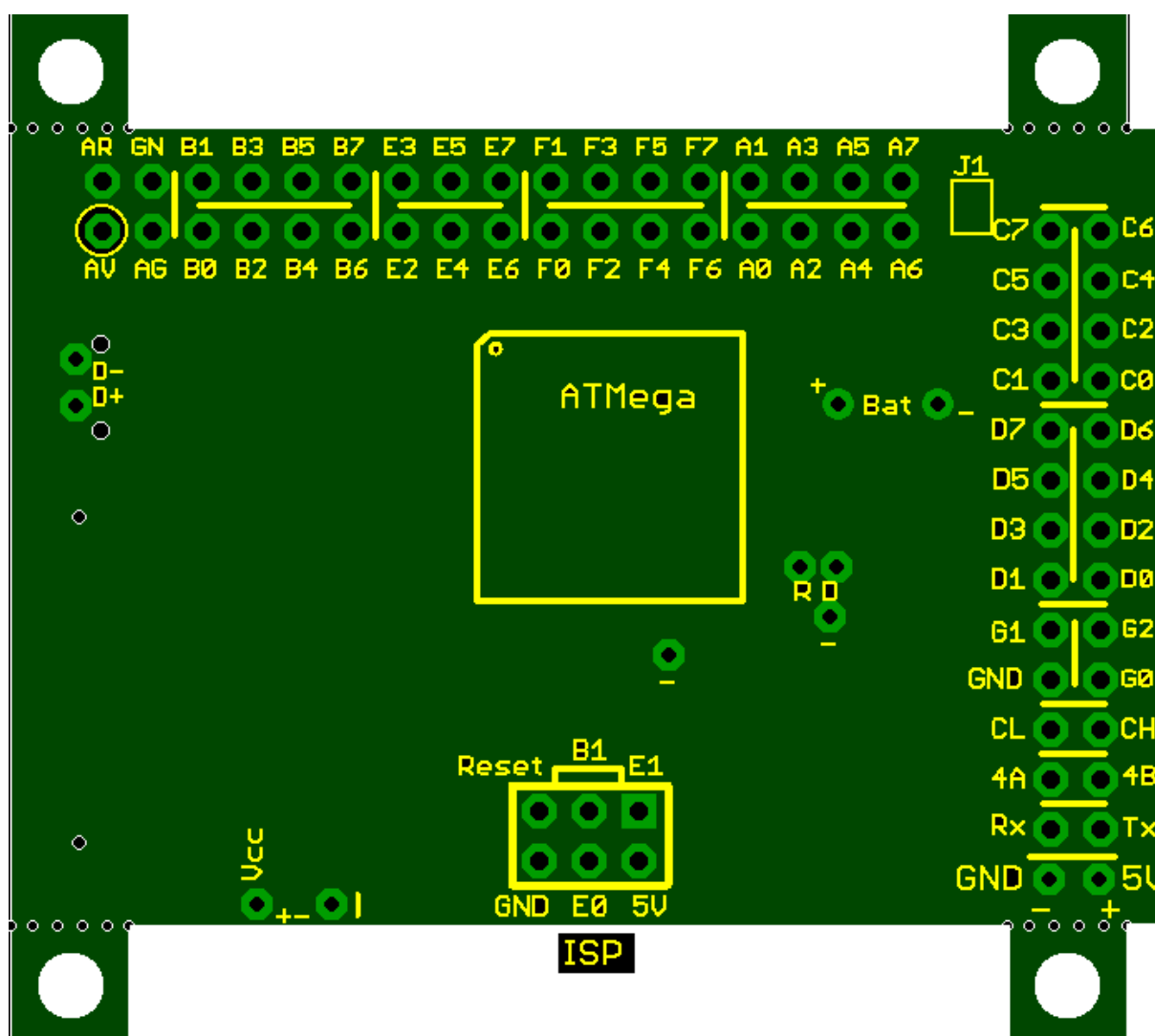
Tipp: Für maximale Effizienz und minimale Wärmeentwicklung empfehlen wir, bei höheren Eingangsspannungen einen DCDC-Schaltregler davorzuschalten. Diesen mit z.B. 5V oder 6V Ausgangsspannung schließen Sie dann vor dem 5V Spannungsregler auf dem Board an. Dieser 5V Längsregler eliminiert dann weitgehend die Störungen des Schaltreglers.

Anschlusspads und Jumper:

Die Anschlussbelegung der Stecker sowie der Jumper ist im untenstehenden Bild zu sehen. Dies entspricht der Bedruckung Ihrer Boards.



Die einzelnen Ports sind jeweils als Block zusammengefasst und auch entsprechend als Block markiert. Tipp: Wenn Sie externe Geräte zum Datenaustausch mit dem Board verbinden möchten, lesen Sie auch das Kapitel zur Terminierung im Anhang dieses Manuals. Auf der nächsten Seite wird der Bereich der beiden Port-Leisten nochmals vergrößert und abgedruckt.



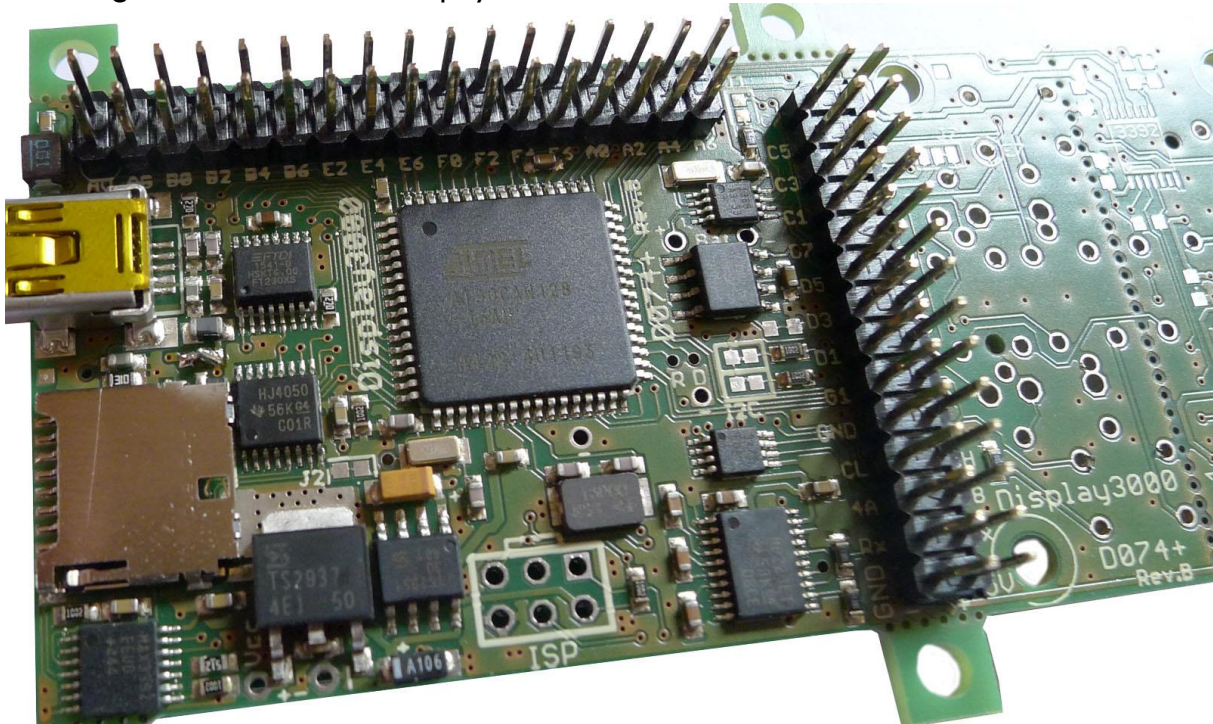
Verschiedene Optionen des Boards belegen einige der vorhandenen Mikrocontroller-Ports. Diese Ports sind dann u.U. für Sie nicht anderweitig nutzbar; manche Optionen können optional an andere Ports gelegt werden. Details hierzu entnehmen Sie bitte der Tabelle Portbelegung am Ende des Dokuments (Seite 56).

Die Bedeutung der verschiedenen Jumper (die sich teilweise auf der Unterseite – also der Displayseite – befinden, der Lieferzustand sowie die Position der Jumper entnehmen Sie bitte der Tabelle Jumperbelegung am Ende des Dokuments ab Seite 56.

Die Anschlussmöglichkeiten

Stiftleisten

Das Board ist für den Anschluss von Stiftleisten vorbereitet. Diese werden auf der Bauteilseite eingesteckt und von der Displayseite aus verlötet.



Achtung: Wenn Sie den Einbau in unser Gehäuse GB400 oder GBS401 planen (natürlich gilt dies auch für alle anderen Gehäuse):

- Probieren Sie vor dem Einlöten evtl. Pins, Steckverbinder etc. aus, ob sich Ihr Gehäuse dann noch schließen lässt.
- BS400 Gehäuse.** Vermutlich werden Sie sowieso nur Kabel direkt in den Pads anlöten. Aber sicherheitshalber: Schauen Sie sich den Sitz des montierten Display vor dem Einlöten der Pins auf der Platine an – es liegt am unteren Rand auf der Platine auf und hat am oberen Rand ca. 2 mm Abstand zur Platine. Diese „schräge“ Montage ist für einen perfekten Sitz im Gehäuse notwendig (Anmerkung: wenn Sie das Gehäuse nicht gleichzeitig mitbestellt haben, liefern wir das Display allerdings mit einem 2mm dicken Schaumklebestreifen um eine Montage parallel zur Platine zu ermöglichen – den sollten Sie dann vor Einbau ins Gehäuse entfernen).

Wenn Sie vorhaben Pins in die Platine einzulöten und die Platine dann im BS400 Gehäuse zu verbauen, dann dürfen Sie diese Pins am unteren Displayrand nicht komplett durchstecken und verlöten (siehe auch Bilder auf Seite 54). Das Display liegt dann nicht mehr auf der Platine, sondern auf den Pins auf – der Abstand ist zu groß. Lösung: Entweder knipsen Sie nach dem Verlöten die überstehenden Pins ab oder Sie verlöten sie so, dass sie die Pins nicht komplett einstecken, sondern nur soweit, dass sie die Spitzen gerade noch nicht aus den Löt pads schauen (das Löt zinn fließt trotzdem in die durchkontaktierten Löt pads und sorgt für einen perfekten Kontakt).

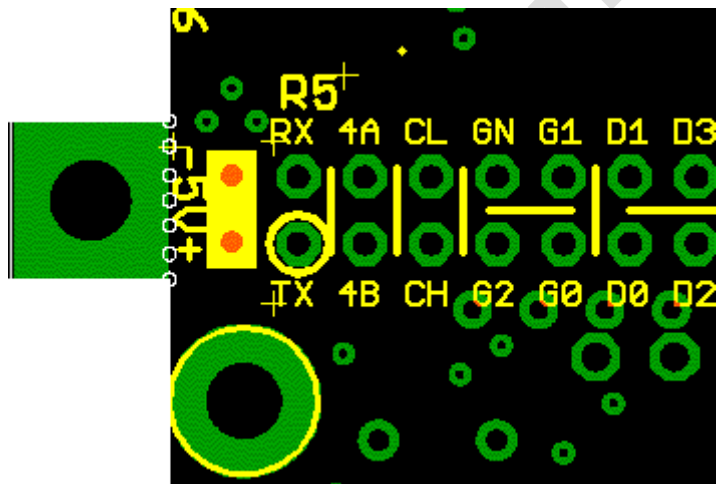
Bitte prüfen Sie zuerst die Einbausituation in dem Sie die Stecker nur einstecken und erst einmal nicht verlöten – eine Tabelle mit der Info was wann wo passt, finden Sie im Anhang auf Seite 90.

Anschlüsse Schnittstellen (RS232, RS485, CAN)

Oberhalb der beiden Pads für 5V und GND befinden sich Pads für:

- RS232 High Pegel Rx, Tx – ansprechbar per UART0
- RS485 (die Leitungen A & B sind mit 4A und 4B gekennzeichnet) - ansprechbar per UART1
- CAN Bus (die Leitungen H & L sind mit CL und CH gekennzeichnet) – nur nutzbar mit einem AT90CAN128 als Mikrocontroller
- Eine weitere GND-Leitung ist hier am Pad „GN“ abgreifbar

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.



Tipp 1: An RX / TX können Sie ohne Anschluss einer Gegenstelle keine Signale messen, da der RS232 Transceiver ohne erkannte Gegenstelle in den automatischen Shutdown geht.

Tipp 2: Ziehen Sie zur Nutzung der RS232 Schnittstelle den Programmieradapter ab, dieser nutzt die gleichen Ports und könnte stören.

Hinweis: Die Low-Level- (TTL) Pegel-Signale der beiden UART Schnittstellen des Mikrocontrollers liegen an:

UART0

Rx: Port E0

Tx: Port E1

UART1

Rx: Port D2

Tx: Port D3

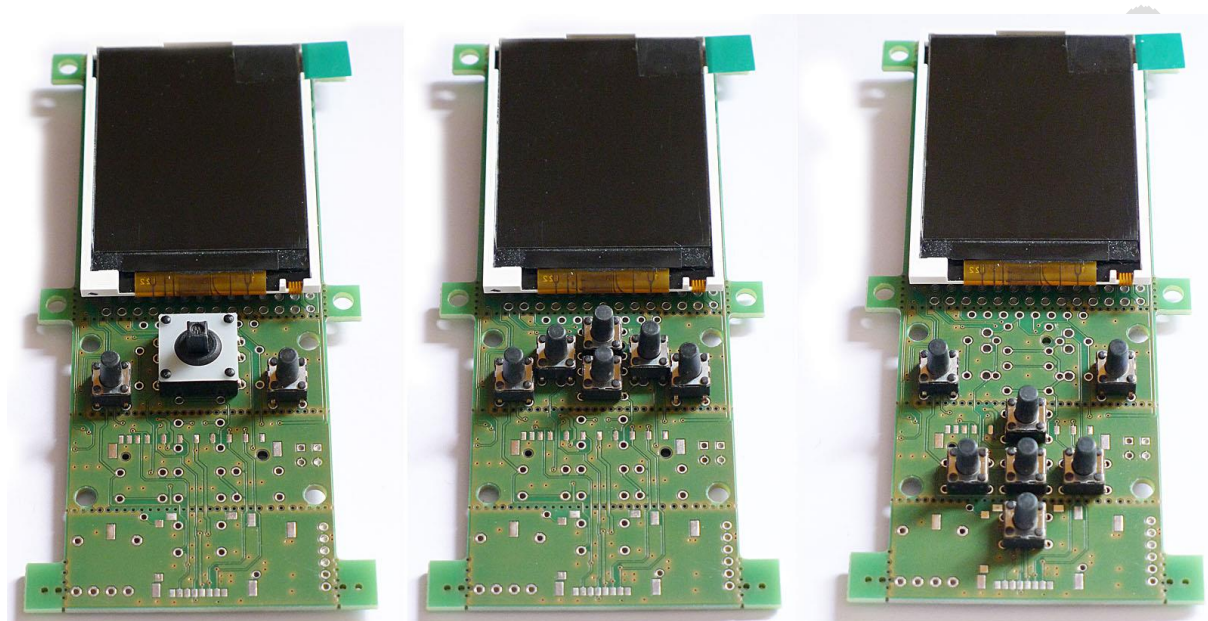
Die Eingabemöglichkeiten

Joystick & Taster

Das Modul verschiedene Optionen an Bedienelementen. Diese können sein:

- Joystick + 2 Taster direkt unterhalb des Displays
- 6 Taster direkt unterhalb des Displays
- 5 Taster im unteren Bereich sowie 2 Taster unterhalb des Displays.

Das folgende Bild visualisiert diese drei Möglichkeiten.



Selbstverständlich steht Ihnen jede andere Kombination zur Verfügung. Die Taster sind bereits fest mit Ports am Mikrocontroller verdrahtet.

Durch Betätigen eines dieser Bedienelemente wird der entsprechend angeschlossene Port an Masse gezogen. Auf dem D074+ werden alle Joystick/Taster-Ports über einen 470 Ohm Widerstand an Masse gezogen – durch diesen Widerstand kann eine versehentliche Fehlprogrammierung somit den Port nicht zerstören.

Die richtige Vorgehensweise zur Abfrage ist hier:

- 1) Tasterport als Eingang definieren
- 2) Den internen Atmel Pullup-Widerstand dieses Tasterports einschalten

Somit liegen die Leitung zwar an einem Potential von 5V, jedoch immer nur über den internen Pullup-Widerstand. Der abgefragte Eingang zeigt also ein High (1) an. Beim Drücken des Tasters oder Bewegung des Joysticks wird nun dieser Eingang gegen Masse gezogen und beim Abfragen dieses Ports zeigt dieser ein Low (0).

Tipp: Die Taster / Joystick 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

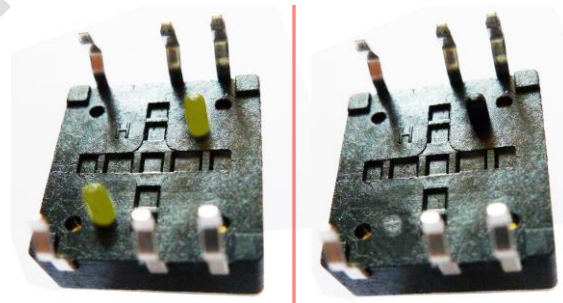
wieder entnehmen können, wenn Sie sich doch für eine Folientastatur entscheiden (evtl. bereits gebohrte Löcher im Gehäuse sind nicht schlimm, denn sie werden ja dann von der Folientastatur verdeckt).

Der Anschluss der Tasten bzw. des Joysticks:

Bedienelement	Genutzter Port Taster+Joystick	Ausnahme Joystick
Ganz linke Taste:	E7	-
hoch:	C3	
runter:	C4	C2
rechts:	C6	
links:	C2	C4
Push:	C5	
Ganz rechte Taste:	C7	-

Wichtig: Um Ihnen im Tastenbereich die Wahl zwischen 6x Tastern oder Joystick + 2 Taster zu lassen, waren aufgrund der eingeschränkten Platzverhältnisse zwei Kompromisse notwendig:

- Beim Joystick mussten die beiden Belegungen C2 und C4 getauscht werden, da wir ansonsten nicht Joystick und die optionale Tasterbelegung im kleinen Feld unterhalb des Display hätten unterbringen können.
- Sie müssen vor dem Einsetzen beim Joystick mit einem Seitenschneider oder Messer eine der beiden Positionierungshilfen auf der Unterseite abknippen/abschneiden, denn an dieser Stelle darf auf der Platine kein Loch sein. Das Bild rechts zeigt (links) die beiden Positionierstifte gelb markiert. Auf dem rechten Bild ist dann der untere linke Stift abgeknipst worden.



Die Taste oben ganz links außen wurde mit Absicht an Port E7 gelegt. E7 ist ein Interrupt-Port und erlaubt das Aufwecken des Mikrocontrollers aus einem Sleep-Modus. Vorteil bei Batteriebetrieb: das gesamte Modul kann sich bei Nichtbenutzung im Sleep-Modus befinden und kann trotzdem per Tastendruck „aufgeweckt“ werden. Über den Port C wäre dies nicht möglich gewesen.

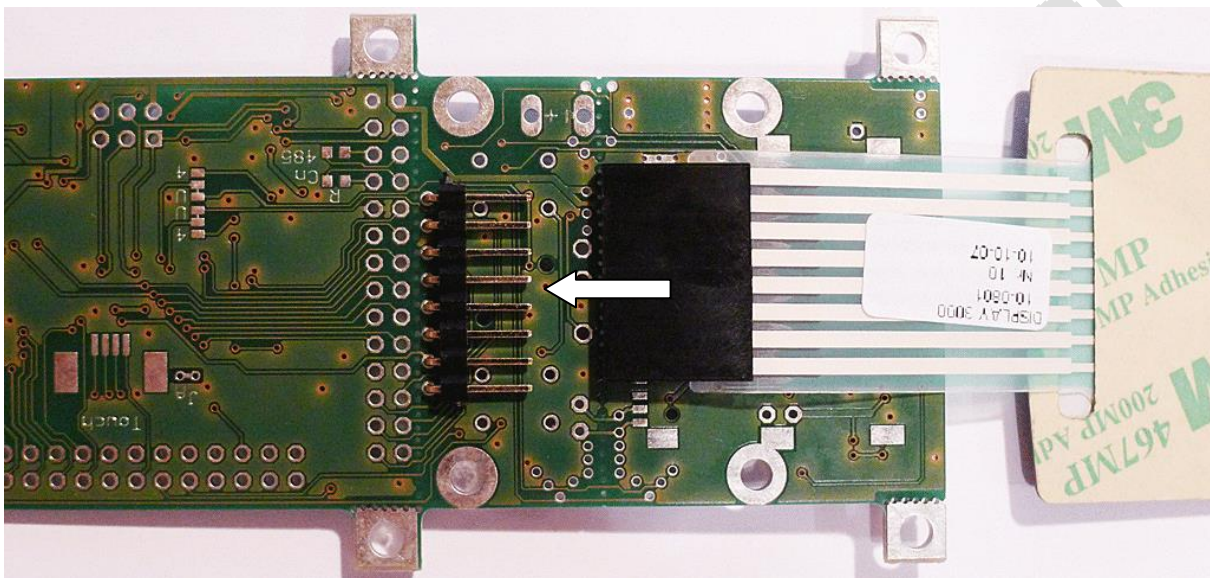
Wichtig für D074+s Nutzer: Bitte lesen Sie die Rubrik *Errata* am Ende des Manuals bzgl. der Joystick/Taster. (D074+x können das ignorieren)

Folientastatur FT02

Speziell zum Gehäuse GBS400 bzw GBS401 haben wir eine Folientastatur entwickelt und produzieren lassen. Die Folientastatur besitzt ein doppelt laminiertes, integriertes Fenster und ist komplett wasserdicht. Die Bedruckung der Folie ist von der Rückseite her geschehen, d.h. die Bedruckung kann nicht abgerieben werden. Für eine leichtere Eingabe von Zahlen, wurden auf die Tastatur auch die Zahlen von 0-9 gedruckt, die so je nach Abfrageprogramm auch direkt eingegeben werden können (0-4 direkt; 5-9 über gedrückte Shift-Taste).

Zum Anschluss dieser Tastatur löten Sie bitte den mitgelieferten gewinkelten Stecker auf der Displayseite der Platine ein. Der Stecker der Folientastatur wird dann einfach eingesteckt.

Wichtig für D07+s Nutzer: Bitte lesen Sie die Rubrik *Errata* am Ende des Manuals.



Jede Taste der Folientastatur wurde hier an einen eigenen Port gelegt. Eine Matrixverschaltung von 7 Tasten hätte die Programmierung nur verkompliziert und trotzdem keine signifikante Port-Einsparung gebracht.

Bilder zeigen noch alte Version

Die Folientastatur wurde identisch zu den Tasten/Joystick beschaltet:

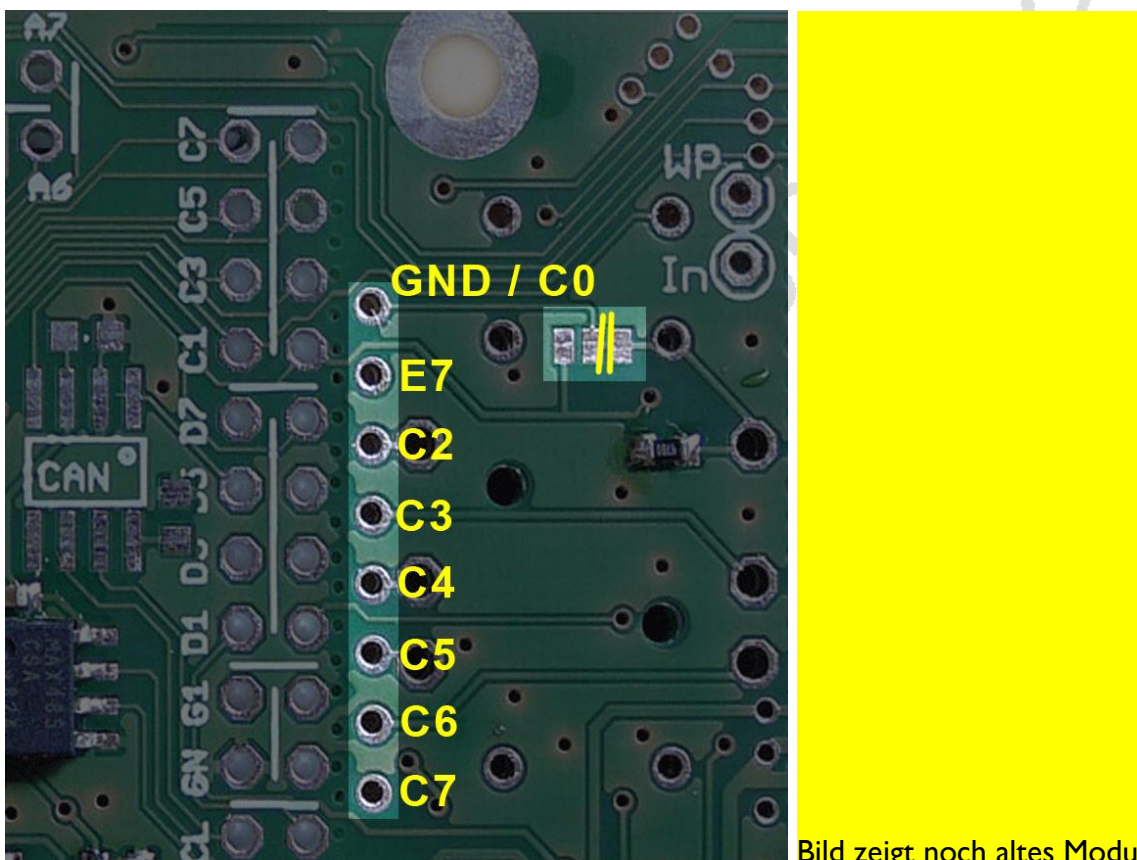
Folientastatur	Genutzter Port
Links (1/5)	C2
Hoch (2/6)	C3
Runter (4/8)	C4
Rechts (3/7)	C6
OK (0/9)	C5
Power / Shift	E7
C	C7



Nutzung von Matrixtastaturen und anderen Tastaturen

Im Lieferzustand ist werden die Tasten der Folientastatur FT400 beim Drücken gegen Masse gezogen (über einen 470 Ohm Schutzwiderstand). Sie können an unser Board jedoch an jeder unserer Folientastaturen (FT600, FT03 (mit 3xLED), FT04) anschließen, d.h. also auch Matrixtastaturen. Für Matrixtastaturen ist dann der Jumper J7 umzukonfigurieren. Damit wird das Pad, das aktuell über den Schutzwiderstand an Masse liegt, nun mit Port C0 verknüpft. Somit sind alle 8 Pads des Folientastatursteckers beliebig zu schalten, d.h., Sie können damit nun alle Tastaturen oder Folientastaturen mit bis zu 8 Leitungen steuern (d.h. max. 4x4, d.h. 16 Tasten abfragen).

Das nachfolgende Foto zeigt zum einen die Belegung des Folientastatursteckers (von der Bauteilseite aus) sowie den o.g. Jumper J7.



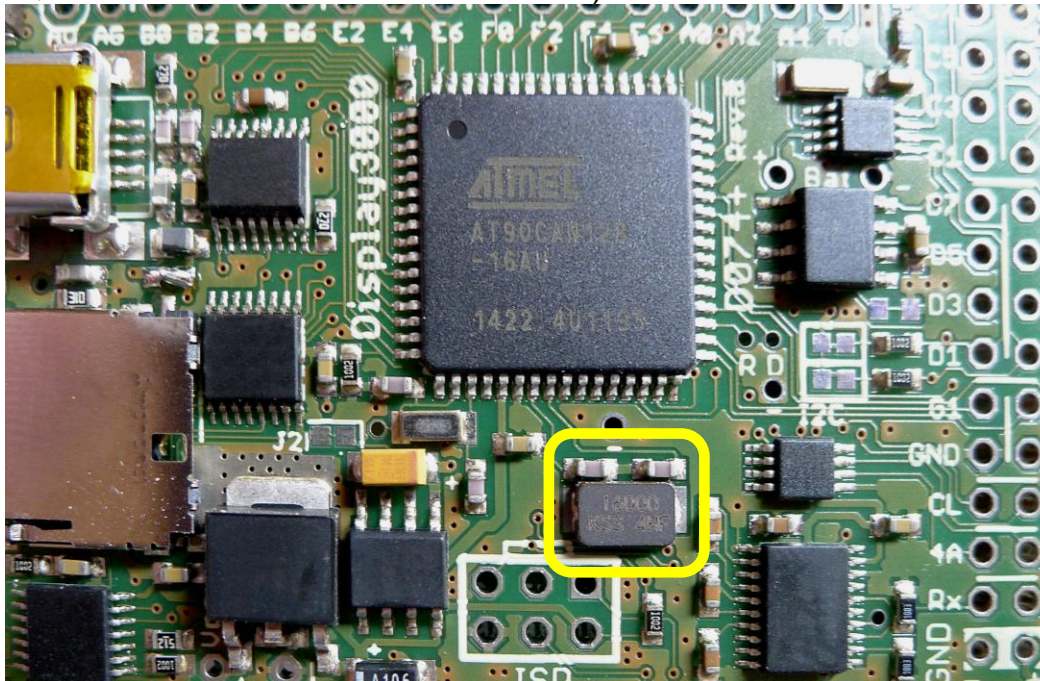
Standardmäßig ist das mittlere Pad des Jumpers mit dem rechten Pad verbunden. Der obere Anschluss des Folientastatursteckers liegt somit an Masse (über 470 Ohm). Für eine Umkonfigurierung durchtrennen Sie die Verbindung zwischen dem mittleren und dem rechten Pad und verbinden dann mit einem Tropfen Lötzinn das mittlere mit dem linken Pad. Das obere Pad des Folientastatursteckers liegt dann an C0.

Wichtiges zu obiger Umkonfiguration und RS485 Betrieb: C0 wird auch beim Einsatz des RS485-Transceivers zur Halb-Duplex-Umschaltung genutzt. Sie müssen hier programmtechnisch Vorkehrungen treffen oder die RS485 Leitung umlegen (Siehe Seite xxx). Solange Sie die vorgesehene Folientastatur verwenden, betrifft Sie dies aber nicht, denn hier ist eine Umkonfigurierung nicht notwendig.

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 Miniatur-SMD-Quarz sowie direkt darüber 2x 15pF SMD-Keramikcondensatoren (Bauform 603 einlöten (siehe Foto). Achtung: Der Platz für den Quarz ist begrenzt. Der SMD Quarz sollte eine Größe von max. 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.

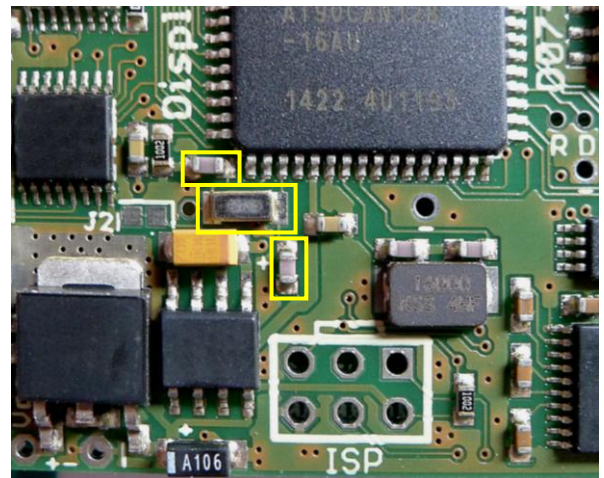
Warnung: spielen Sie nicht grundlos mit den Einstellungen der Fuses herum – damit machen Sie u.U. Ihr Board unbrauchbar! Fragen Sie uns im Zweifelsfall vorher. Ein Anruf genügt.

Uhrenquarz für den internen Real Time Counter

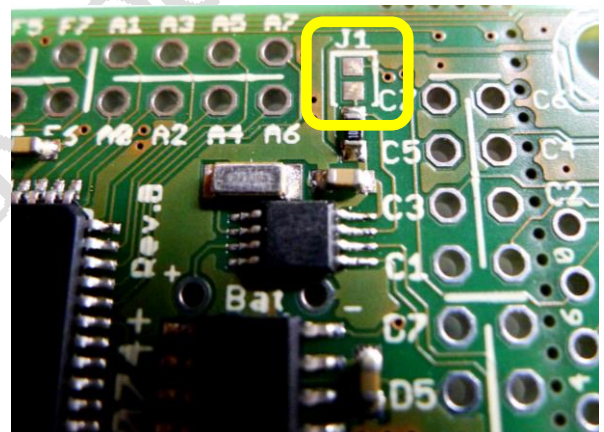
Möchten Sie den Atmel-internen Real Time Counter (nicht zu verwechseln mit einer Real Time Clock, beides mit RTC abgekürzt) nutzen, so benötigen Sie hierfür entweder einen Uhrenquarz oder einen entsprechenden Takt.

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 den Jumper J_1 (oberhalb der Echtzeituhr) mit einem Tropfen Löt-zinn 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 ATMega128a sollte laut Datenblatt nicht von einem externen Takt gesteuert werden und braucht daher einen eigenen Uhrenquarz wenn Sie den Real Time Counter nutzen möchten.

Schnittstellen

Zur Kommunikation mit anderen Geräten stehen auf der Platine Stecker für 1 x RS232-, einer RS485- Verbindung sowie dem CAN-Bus (nur bei Controller AT90CAN128) zur Verfügung. Als Ergänzung kommen noch I²C 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)

SPI wird auf unserem Board zur Ansteuerung des Displays sowie für die optionale SD-Karte verwendet. Diese drei o.g. Ports sollten Sie daher (es sei denn für die Nutzung bei einer weiteren SPI-Peripherie) nicht für andere Zwecke nutzen.

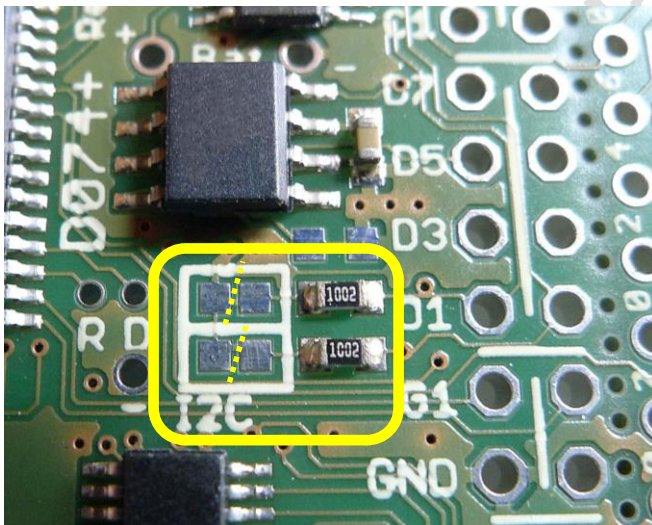
I²C / TWI – Zweidraht-Interface

Der ATmega bietet u.a. auch ein TWI, was nichts anderes bedeutet als Two-Wire-Interface. Ein bekannter Vertreter eines TWI ist z.B. I²C (ausgesprochen: I square C) – elektrisch sind beide identisch. Lediglich aus lizenzrechtlichen Gründen (Die Rechte an I²C liegen bei Philips) benennt Atmel seine I²C 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 I²C-Bus ist auch die einfache Ansteuerung. Da keine festen Taktzeiten eingehalten werden müssen, können sowohl langsame als auch sehr schnelle Busteilnehmer, Chips und Programmiersprachen eingesetzt werden. Soviel zur Einführung – weitere Informationen hält das Internet in Hülle und Fülle bereit.

Dieses TWI (quasi identisch zu I²C) liegt bei dem vorliegenden Controllern an den Ports D0 und D1. Dies bedeutet, über die Ports D0 und D1 findet der I²C Datenverkehr statt und diese Ports können nicht für andere Aufgaben genutzt werden, solange der I²C Bus genutzt werden soll.



Der I²C 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 I²C-Master im Bus vorhanden, müssen die Pullup-Widerstände deaktiviert werden. Dazu brauchen Sie diese aber nicht auszulöten. Es reicht, die beiden vorbereiteten Lötbrücken zu öffnen (auf der Platine mit I2C gekennzeichnet):

Im gelb markierten Feld erkennen Sie die beiden I²C Pullup-Widerstände (je 10 KOhm) sowie die beiden links danebenliegenden Lötbrücken, welche standardmäßig geschlossen sind und die beiden Widerstände jeweils mit der 5V Versorgungsspannung verbinden.

benliegenden Lötbrücken, welche standardmäßig geschlossen sind und die beiden Widerstände jeweils mit der 5V Versorgungsspannung verbinden.

Solange die Lötbrücken geschlossen ist, sind die Ports D0 und D1 für Sie u.U. nicht anderweitig nutzbar! **Achtung:** die Kommunikation mit der optionalen Echtzeituhr benötigt I²C.

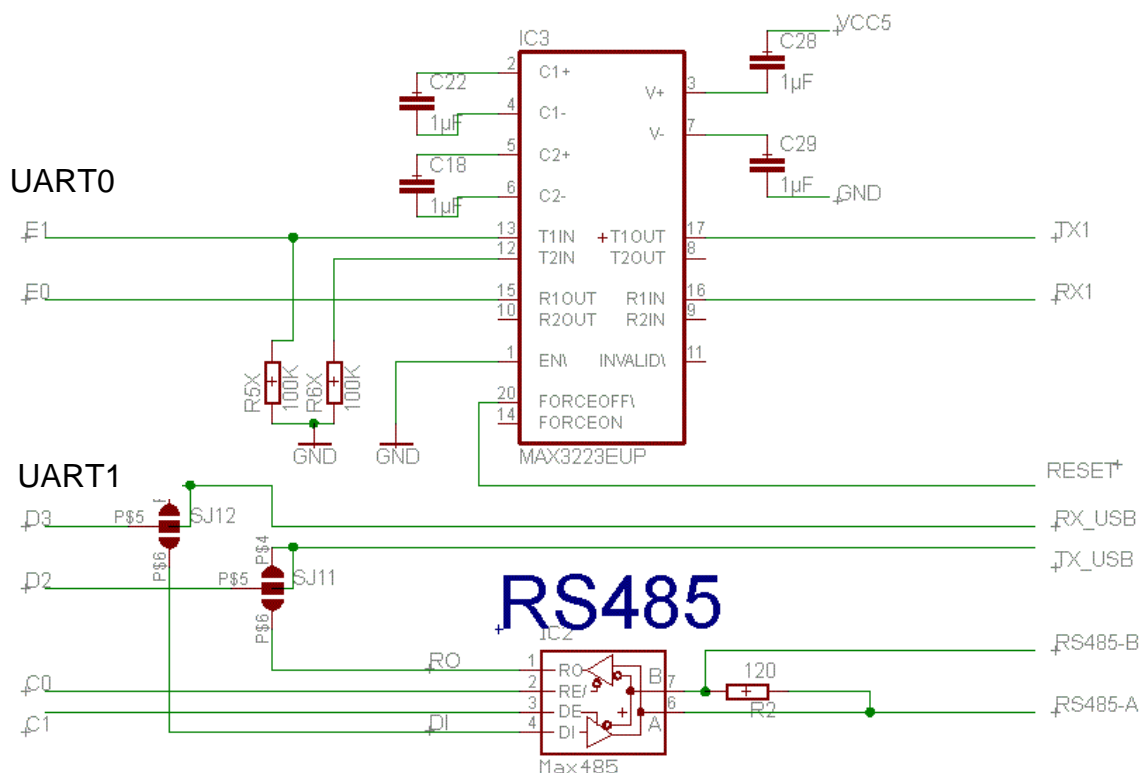
Wenn Sie den Lötbrücken öffnen (die dünnen Leiterbahnen mit einem feinen Messer oder Skalpell durchtrennen – siehe gestrichelte Linie), sind die Ports D0 und D1 nicht mehr über die Widerstände mit 5V verbunden. Sie können die Lötbrücken jederzeit wieder mit einem Tropfen Lötzinn schließen.

Serielle Schnittstellen

Die ATmega 128 / 2561 und der AT90CAN128 bieten zwei serielle Schnittstellen: UART0 und UART1, die standardmäßig von uns vorbelegt wurden und deren Zuweisung von Ihnen teilweise umkonfiguriert werden kann.

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

RS232

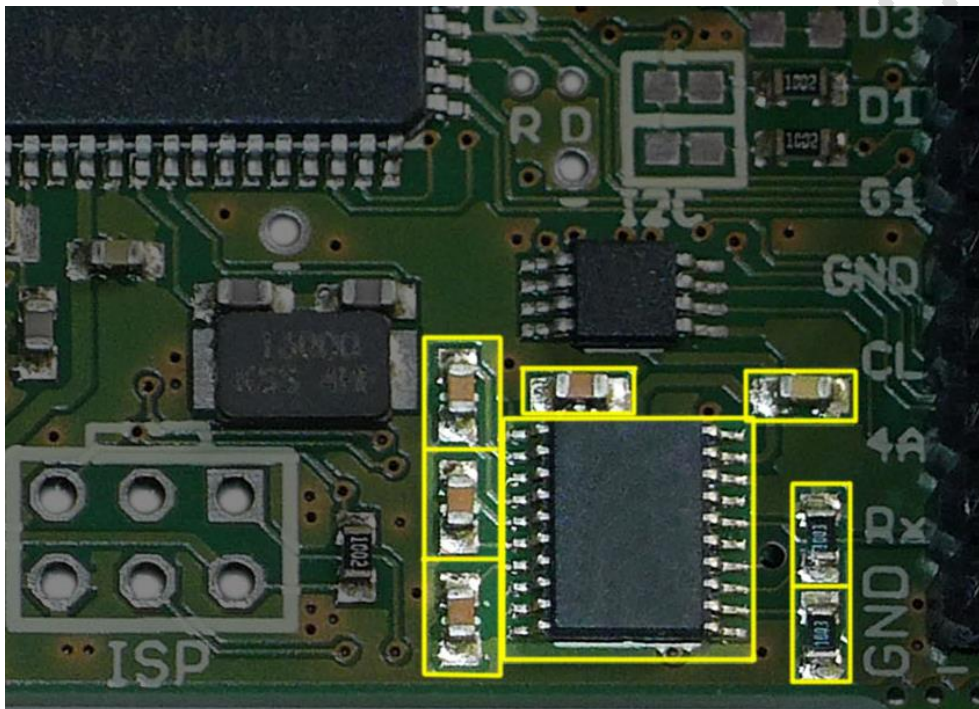


Jumper J11 und J12 sind standardmäßig mit TX_USB bzw. RX_USB verbunden und müssen für die Nutzung mit RS485 umkonfiguriert werden. Diese beiden Jumper liegen auf der Unterseite der Platine unterhalb des Displays. Für den Zugriff muss das Display abgenommen werden. Um das Display nicht zu beschädigen, lesen Sie bitte unbedingt das Kapitel *Abnehmen des Displays* auf Seite 92.

RS 232-Belegung

Die Pins für die serielle Schnittstelle 0 des ATMega-Controllers werden auch für das ISP-Programmierinterface genutzt – beide kämen sich normalerweise bei einer gleichzeitigen Nutzung in die Quere, da der RS232-Pegelwandler die Receive-Leitung ohne Signal permanent auf High zieht. Auf diesem Board wird der RS232 Transceiver beim Programmiervorgang jedoch automatisch abgeschaltet, so dass hier keine weiteren Maßnahmen zu beachten sind.

Das Board D074 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 und -Werte siehe Schaltplan auf Ihrer CD): 1 x SMD RS232 Transceiver, TSSOP20; 5xKondensatoren; 2xWiderstände, alles Bauform 603.



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, bzw. Sie den RS232-Transceiver aufgeweckt haben (siehe vorhergehender Absatz) – ansonsten messen Sie hier nichts).

Auch wenn bei RS232 oft (z.B. PC) Signalpegel von +12V und -12V genutzt werden, liegen die Signale unseres Boards noch innerhalb der RS232 Spezifikation

(siehe ab Seite 64 ff.) und werden problemlos von der Gegenstelle erkannt. Selbstverständlich kann das D074 am Rx-Eingang alle bei RS232 erlaubten Signalpegel verwerten (aber nur an RX – an den normalen Ports des Mikrocontrollers dürfen nur max. Spannungen in Höhe der Controller-Spannung anlegen (i.d.R. also +5 Volt / GND)).

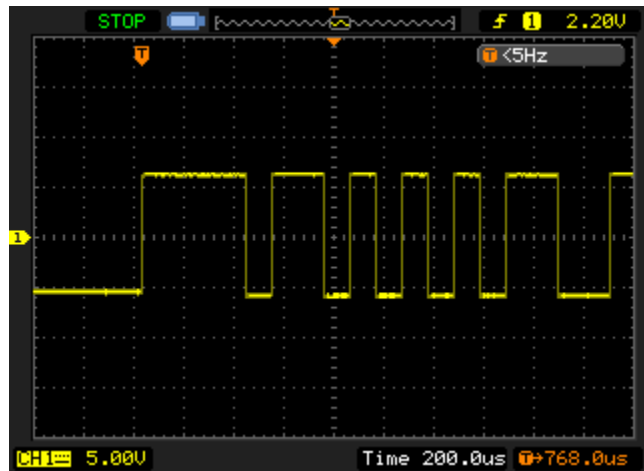
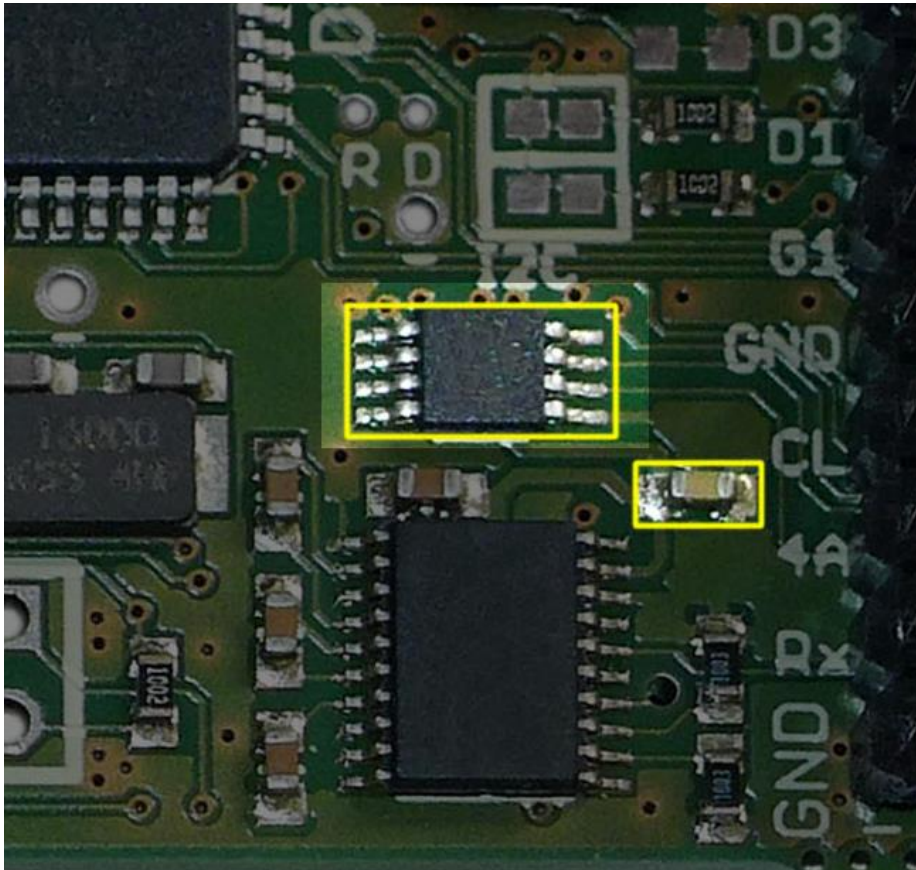


Bild erneuern

Anmerkung: Auf Seite 64 ff. erfahren Sie mehr generelle Details über RS-232 und RS485.

RS-485 Interface

Das D074-Board wird standardmäßig ohne RS485-Transceiver geliefert. Wenn Sie RS485 benötigen, bestellen Sie diese Option bitte direkt mit – dann sind alle Bauteile bereits von uns eingelötet worden, ansonsten müssen Sie die Bauteile selber besorgen und einlöten. Dazu löten Sie an die nachfolgend gezeigte Position (I) z.B. einen Max485 Chip im SMD-Gehäuse (S08) ein.



Auf der Platinenrückseite (also auf der Displayseite) finden Sie im Bereich des RS485 Transceivers zwei Jumper-Felder (**J11 und J12**), die jeweils mit „4“ und „U“ gekennzeichnet sind. Für den Zugriff muss das Display abgenommen werden. Um das Display nicht zu beschädigen, lesen Sie bitte unbedingt das Kapitel *Abnehmen des Displays* auf Seite 92.

Mit diesen Jumpers hat es folgende Bewandnis: Wir haben die 2. Serielle Schnittstelle (UART1) des Mikrocontrollers wahlweise für die USB-Kommunikation oder die Kommunikation per RS485 ausgeführt. Diese beiden Jumperfelder verbinden die RxI und TxI Leitungen (D2 und D3) entweder mit dem USB-Baustein oder mit dem RS485-Transceiver. Standardmäßig besteht eine Verbindung mit dem USB-Baustein. Das erste nachfolgende Foto zeigt den Lieferzustand: das jeweils mittlere Feld des Jumpers ist standardmäßig mit dem mit „U“ (für USB) gekennzeichneten Feld verbunden.

Wenn Sie stattdessen RxI und TxI des Controllers mit RS485 verbinden möchten, trennen Sie mit einem feinen Skalpell die bestehende Leitung innerhalb des Jumpers auf (jeweils an der mit „U“ gekennzeichneten Stelle) und verbinden dann (z.B. mit einem Tropfen Lötzinn) jeweils das mittlere Feld mit dem mit „4“ gekennzeichneten Feld (2. Foto).

Eine gleichzeitige Beschaltung der UART1 mit RS485 und USB ist nicht möglich.

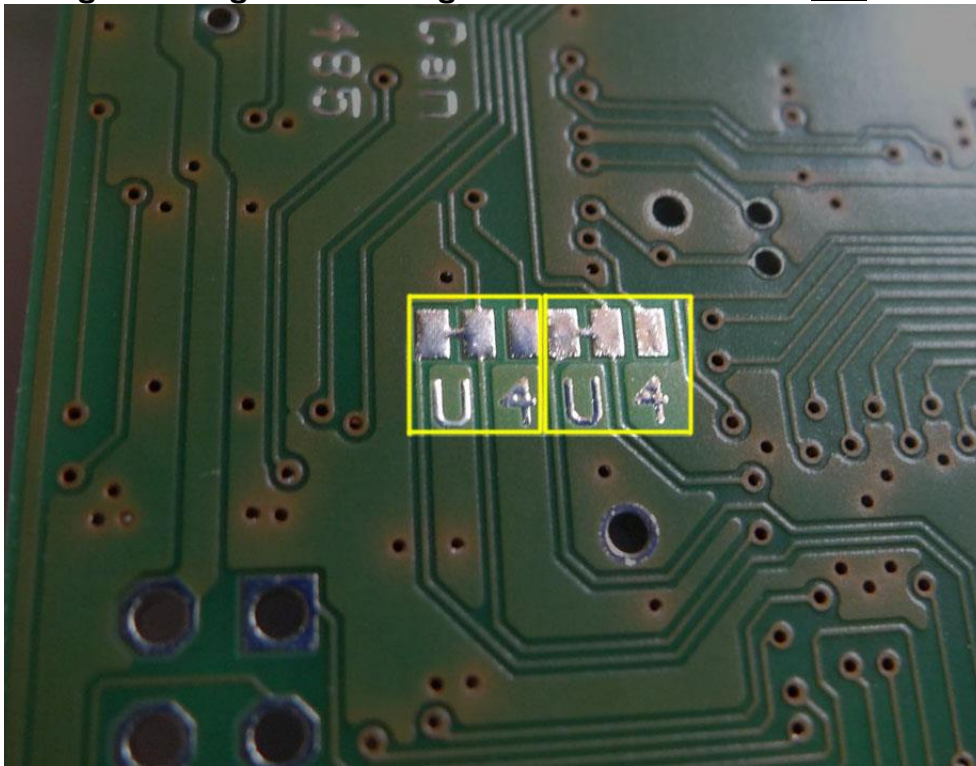
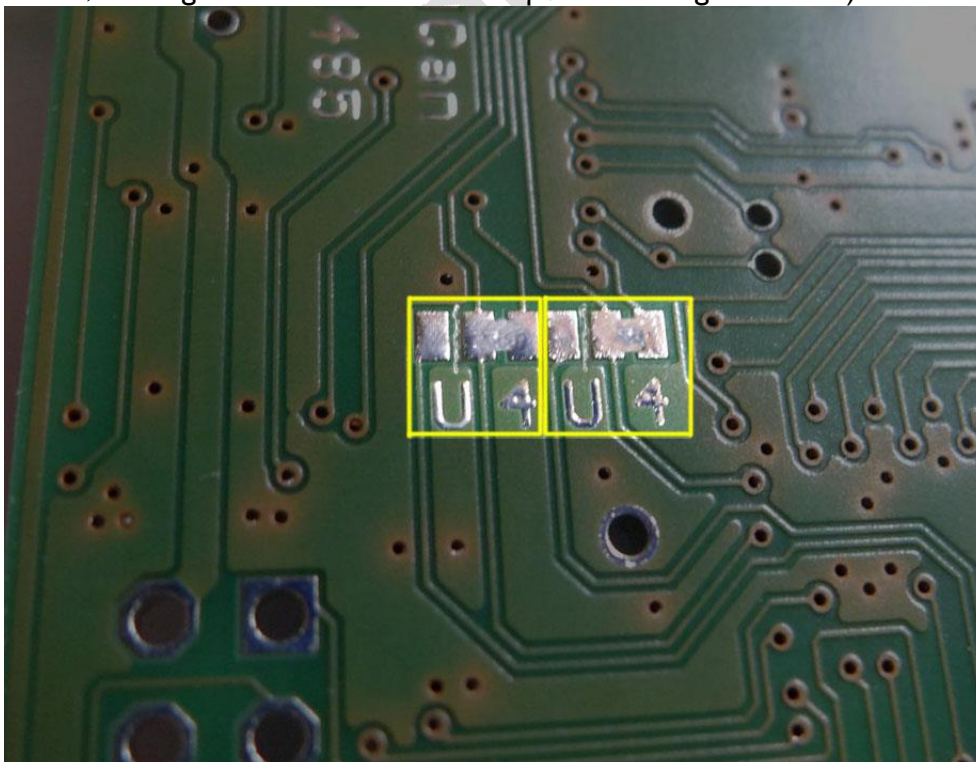


Foto oben: Konfiguriert für USB (Lieferzustand) – das jeweils mittlere Feld der Jumper ist mit „U“ verbunden

Foto unten: Manuell für RS485 umkonfiguriert (Leitung Mitte <> U mit einem Skalpell aufgetrennt; Leitung Mitte <> 4 mit einem Tropfen Lötzinn geschlossen)



RS485 ist bei einem differentiellen Leitungspaar grundsätzlich nur ein Halbduplex-Betrieb möglich, denn man kann nicht zur selben Zeit auf demselben differentiellen Leitungspaar gleichzeitig senden und empfangen. Full-duplex Verbindungen bedürfen zweier differentieller Leitungspaare.

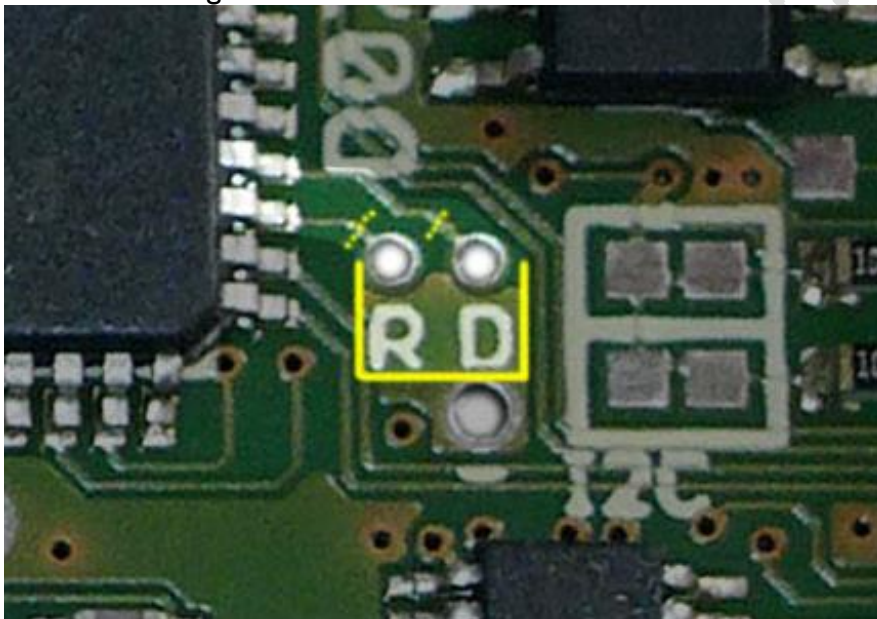
Der RS485 Transceiver auf unserem Board ist daher ein Halbduplex-Chip. Um beim Halbduplexbetrieb die korrekte Umschaltung zwischen Senden und Empfangen durchzuführen, können Sie an diesem Chip die Sende- und Empfangseinheit getrennt ein- und ausschalten:

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

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

Achtung: Die beiden Ports C.0 und C.I für RE / DE werden auch für den Tastenbereich verwendet. Sie sollten somit während einer Übertragung die beiden Tasten neben dem Joystick nicht verwenden, da dies die Übertragung stören würde (ein Druck auf die Taste würde die Leitung – über den 470 Ohm Sicherheitswiderstand – gegen Masse ziehen). Die Nutzung der optionalen Folientastatur FT02 beeinträchtigt den RS485 Betrieb nicht, da hier die Leitungen C0 und CI nicht genutzt werden.

Alternativ haben wir vorgesehen, dass Sie die Leitungen für RE und DE an einen beliebigen anderen Port legen können.



Dazu gibt es zwei Löt pads die mit R (für RE) und D (für DE) gekennzeichnet sind. Um die Leitungen RE und DE anderweitig nutzen zu können müssen Sie:

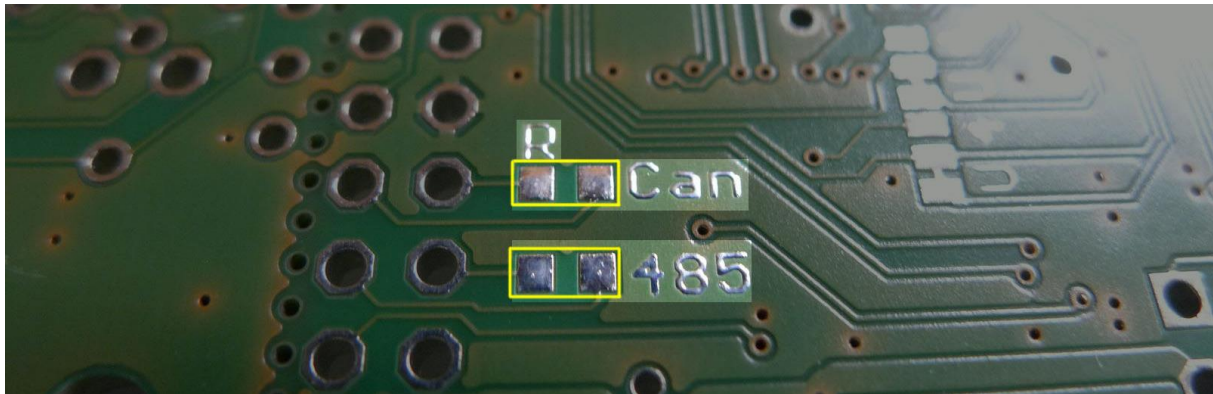
- a) Die beiden gestrichelt gezeichneten Leiterbahnen an den beiden Löt pads durchtrennen. Dies trennt die bisherige Verbindung der Ports C0 und CI zum RS485 Transceiver auf. Keine Sorge, die beiden Ports C0 und CI sind weiterhin mit ihren Pads auf der Port-Steckerleiste verbunden.
- b) Eine manuelle Verbindung von den beiden Löt pads R und D zu zwei beliebigen anderen freien Port durchführen, mit denen Sie dann in Ihrer Software die Halbduplex-Steuerung durchführen.

RS485 Betrieb bei weniger als 5 Volt

Auf dem D074+ wird ausschließlich ein Baustein verwendet, der bereits ab 3 Volt arbeitet. Im Gegensatz zum alten D074 Modul gibt es hier keine Einschränkung mehr.

RS485-Abschlusswiderstand

Bei RS485 kann es für die Endgeräte am Bus notwendig (also nicht für jedes Gerät, sondern nur die Geräte an den beiden Enden des Bus) sein, einen 120 Ohm Bus-Abschlusswiderstand einzuschalten. Auf dem folgenden Foto erkennen Sie die Position des optionalen Widerstands (auf der Displayseite, mit 485 bezeichnet). Dort kann dann ein 120 Ohm Widerstand in Bauform 603 (Bauform 805 passt auch) eingelötet werden. Alternativ können Sie diesen natürlich auch als bedrahteten Widerstand direkt an Ihrer Verkabelung anbringen.



Das Display müssen Sie hierfür zuerst entfernen. Infos zum Abnehmen des Display erhalten Sie auf Seite 92.

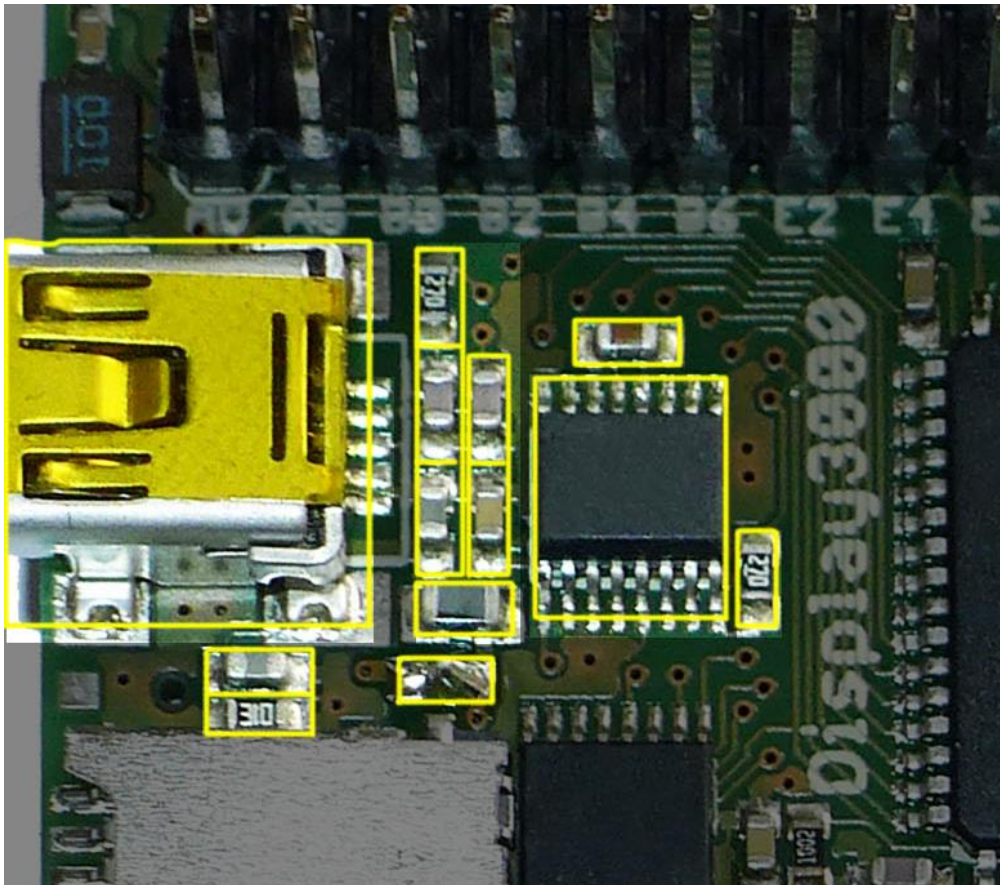
Anmerkung: Auf Seite 64 ff. erfahren Sie mehr generelle Details über RS-232 und RS485.

Gute weiterführende Informationen zu RS485 erhalten Sie hier:

http://www.interfacebus.com/Design_Connector_RS485.html

http://www.embeddedsys.com/subpages/resources/images/documents/microsys_art_RS485.pdf

USB



Das Modul D074+ wird optional mit einer integrierten USB Beschaltung und einer Mini USB Buchse geliefert. Somit ist der Anschluss an einen beliebigen PC möglich. Der Hersteller des genutzten USB-Chips (FTDI) stellt Treiber für div. Windows-Systeme, Unix und MAC OS bereit, so dass Sie fast jeden beliebigen PC mit dem D074+ verbinden können. Der Mikrocontroller spricht den USB Chip per RS232 über die UART1 an. Zusammen mit den Treibern des Chipherstellers bietet der USB-Chip dann eine virtuelle serielle Schnittstelle am PC an; über (z.B. COM2 oder COM5) können Sie so dann Daten zwischen Mikrocontroller und PC austauschen. Die Kommunikation ist dann quasi identisch zu einem Rechner mit RS232 Schnittstelle – die es aber leider heutzutage nur noch in wenigen PCs standardmäßig gibt.

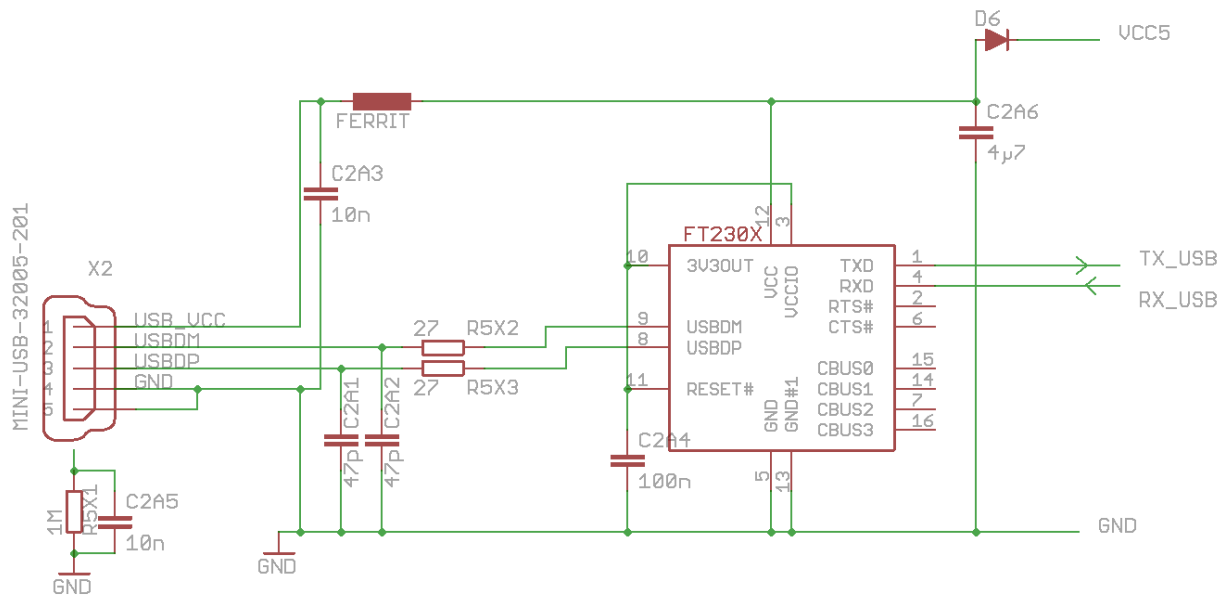
Mögliche Einsatzgebiete

- Debugging (Ausgabe von Statusmeldungen; Darstellung auf dem PC in einem Terminalprogramm)
- Abrufen von im Mikrocontroller gespeicherten Protokoll- oder Statusdaten
- Einspielen eines neuen Programms (per Bootloader)
- Änderung von Systemvorgaben

Praktischer Nebeneffekt: Die 5V Spannungsversorgung vom PC, die über USB abgegriffen wird, erlaubt einen direkten Betrieb des D074+ Moduls über USB, d.h. es ist dann keine weitere Spannungsversorgung notwendig.

Das Controllerboard selbst wird mittels einer Schottky-Diode von der USB-Spannungsversorgung entkoppelt.

USB-Seriell



Wie weiter oben beschrieben (Kapitel RS485) ist für die Kommunikation per RS485 oder USB UART1 des Mikrocontrollers zuständig. Wie zwei Seiten vorher beschrieben wurde, können Sie entweder mit RS485 oder USB kommunizieren. Die Jumper sind standardmäßig für USB konfiguriert, müssen hier also nicht geändert werden.

Beim erstmaligen Anstecken des D074x+ an USB werden bei Windows die Treiber installiert. In der Regel bringt das Windows-System die Treiber bereits mit und installiert sie nach dem Anstecken automatisch. Falls nicht, oder falls Sie ein anderes Betriebssystem nutzen, müssen Sie die Treiber von den Herstellerseiten downloaden und einmal selbst installieren. Mehr hierzu und zur Änderung der Parameter erhalten Sie im Anhang USB-Treiber auf Seite 75.

CAN-Bus Interface

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

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

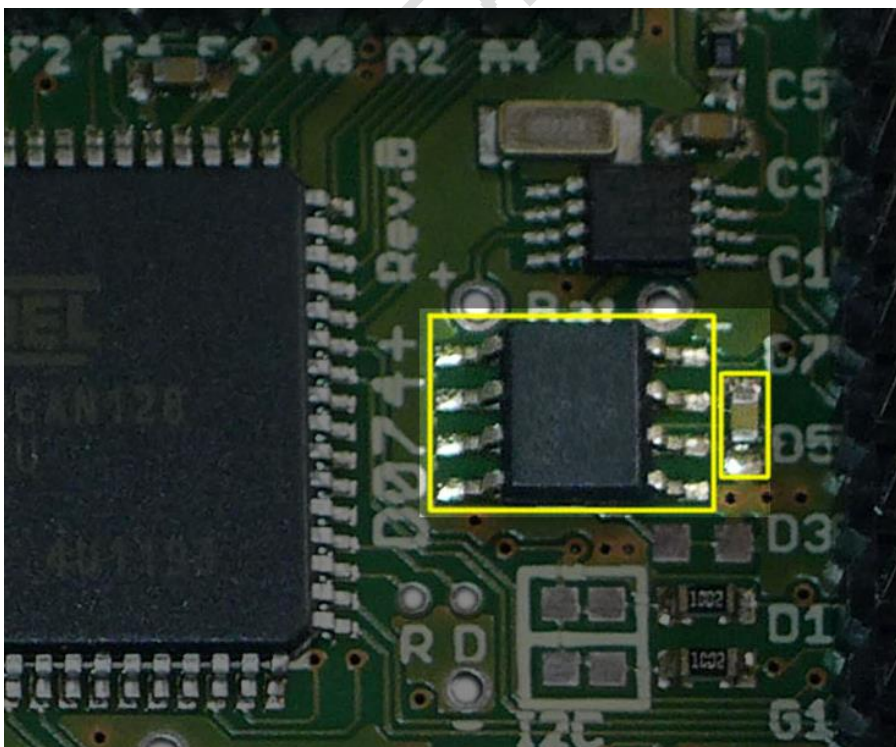
1 MBit/s:	40 m
500 KBit/s:	100 m
125 KBit/s:	500 m
50 KBit/s:	1 km

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

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 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 Nexperia TJA1050 (vormals NXP davor Philips).

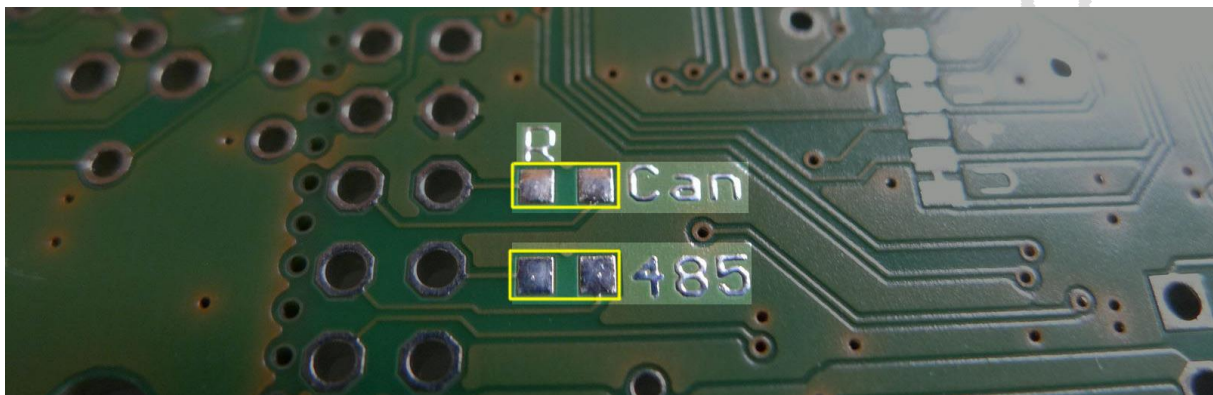
Wenn Sie die Platine mit CAN Controller AT90CAN128 bestellt haben, so löten wir zusätzlich den hierfür benötigten High Speed CAN Transceiver mit ein.



Der CAN Transceiver ist fest mit den Ports D5 und D6 des Mikrocontrollers verbunden. Auch wenn Sie erst einmal keinen CAN Bus nutzen, ist daher bei Vorhandensein des CAN Transceivers (nur wenn Sie das Board mit einem AT90CAN128 bestellt haben) die anderweitige Nutzung der Ports D5 und D6 nicht empfehlenswert.

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. Dieser Widerstand ist nicht im Lieferumfang enthalten. Für den Zugriff auf dieses Feld muss das Display abgenommen werden. Um das Display nicht zu beschädigen, lesen Sie bitte unbedingt das Kapitel *Abnehmen des Displays* auf Seite 92.



Der Widerstand verbindet die beiden Leitungen H und L miteinander. Selbstverständlich können Sie auch einen bedrahteten 120 Ohm Widerstand direkt am CAN-Bus-Kabel einlöten.

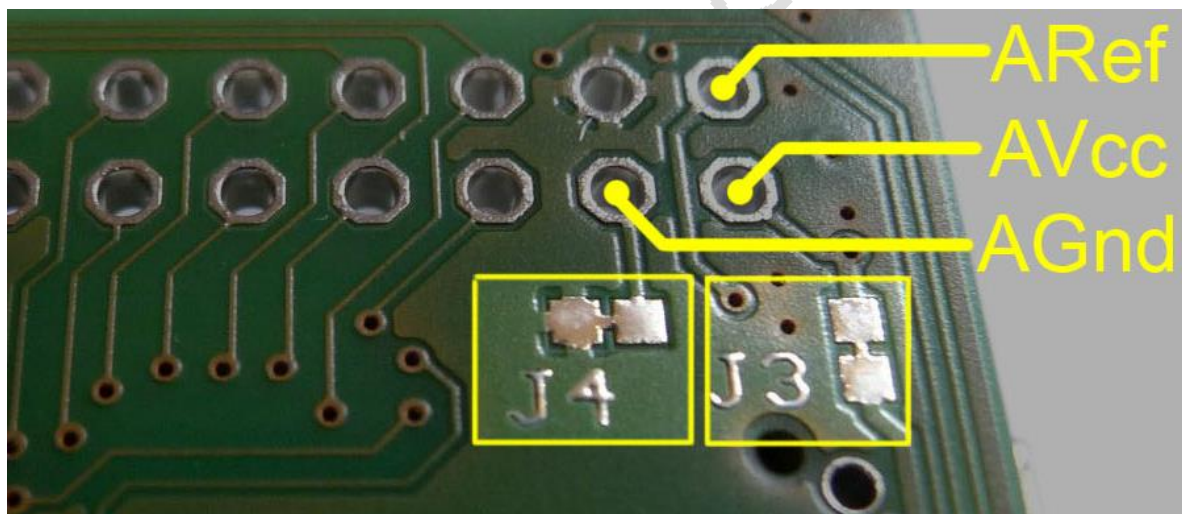
Analoge Eingänge (Analog-Digital-Wandler)

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 Mikrocontroller 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 über eine 10 μH Induktivität mit VCC (5 Volt Versorgung). Diese Verbindungen haben wir bereits vordefiniert.

Wenn Sie AVCC und AGND separat kontaktieren möchten, so müssen Sie hierzu zwei (im Lieferzustand geschlossene) Brücken zu VCC5 und GND öffnen. Diese beiden Brücken befinden sich auf der Rückseite der Platine unterhalb des Displays und sind mit J3 und J4 beschriftet. Sie müssen somit zuerst das Display entfernen. Um das Display nicht zu beschädigen, lesen Sie bitte unbedingt das Kapitel *Abnehmen des Displays* auf Seite 92.



Die Brücke J4 schließt AGND zu GND.

Die Brücke J3 schließt AVCC zu VCC5.

Beide Brücken sind im Lieferzustand geschlossen. Nach dem Öffnen können Sie eigene Spannungen an die Pads ARef und AVcc anlegen (unabhängig von der Jumper-Konfiguration wird die AVcc-Leitung des μC immer über eine 10 μH Induktivität und einen 100nF gefiltert).

ARef ist der Eingang für die Referenzspannung des internen Analog-Digitalwandler. 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. Auch die Aref-Leitung hat direkt vor dem μC einen 100nF erhalten.

Schalten der Beleuchtung durch den Mikrocontroller

Da die Lebensdauer der Displaybeleuchtung endlich ist (ca. 20000 Betriebsstunden bis zum Abfall der Ausgangshelligkeit auf 70%) und sich diese zudem mit u.U. (je nach Helligkeit) über 90% am Energiebedarf des Moduls verantwortlich zeigt, macht es Sinn, die Beleuchtung bei Nichtgebrauch abzuschalten oder herunterzuregeln. Praktischerweise kann auch der Mikrocontroller die Beleuchtung steuern. Beispiel: auf 50% dimmen, wenn über 1 Minute lang keine Bedienung mehr erfolgt ist, nach weiteren 4 Minuten wird die Beleuchtung dann komplett abgeschaltet – sobald eine Taste betätigt wird, schaltet die Beleuchtung sich wieder mit 100% ein).

Der Booster auf der Displayplatine kümmert sich selbstständig um die Spannungserzeugung für die Displaybeleuchtung im Bereich der Eingangsspannung von 2 bis 5 Volt. Die Beleuchtung wird über den Port B7 geschaltet.

Schalten der Displaybeleuchtung

In Ihrer Software müssen Sie 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.

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 71.

Das Display des neuen D074+ ist ca. 4 mal heller als das des Vorgängermoduls D074 (ohne '+'). Allerdings geht dies zu Lasten des Stromverbrauchs. Daher sollten Sie die Displaybeleuchtung bei Batteriebetrieb dimmen bzw. bei Nichtbenutzung abschalten. Bei einer PWM von 20% benötigt die Displaybeleuchtung nur noch 1/5 des Stroms und ist dann in etwa so hell wie das alte Display.

Stromverbrauch

Der Stromverbrauch des D074+ Board ist von verschiedenen Faktoren abhängig. Während die Variation bei einem Betrieb mit einer festen Spannungsversorgung (z.B. externes Netzteil etc.) zu vernachlässigen ist, ist die Beachtung des Standby-Verbrauchs für ein batteriebetriebenes Gerät durchaus wichtig.

Grundsätzlich verbraucht der Mikrocontroller (bei 16Mhz) bei Betrieb mit 5 Volt rund 35 mA. Einige andere Verbraucher besitzen einen Standby-Modus bei Nichtbenutzung. So ist es z.B. kein Problem, einen RS232 Transceiver auf dem Board zu belassen, denn der von uns verbaute Baustein verbraucht bei Nichtbenutzung lediglich 1µA.

Mikrocontroller	16 Mhz	8 Mhz	Sleep Mode / Standby
ATMega128A	30-35 mA	20-25 mA	Je nach Modus
ATMega2561	30-35 mA	20-25 mA	Je nach Modus
AT90CAN128	30-35 mA	20-25 mA	Je nach Modus
Display / Displayelektronik (3V über Regler)		2,5 mA	30µA
Displaybeleuchtung	VCC 3V-5V		
Displaybeleuchtung 100%		85mA	0,1µA
Displaybeleuchtung 50%		43mA	0,1µA
Displaybeleuchtung 25%		22mA	0,1µA
Displaybeleuchtung 10%		8,5mA	
Displaybeleuchtung 0% (Licht aus)		0,1µA	0,1µA
Diverse Optionen des D074+		Während der Nutzung	Ruhestrom / Standby
SD Karte (3V über Regler)		während des Schreibens Peaks bis 100mA	200µA bis 1mA je nach Karte
Echtzeituhr		325µA	200µA
RS485 Transceiver		1,6µA beim Empfang Senden: je nach Last	1,6µA
RS232 Transceiver		Je nach Last ab 300µA	1µA
CAN Transceiver		Je nach Last am CAN Bus	n/a
3V Spannungsregler (Eigenverbrauch)		Je nach Last 100µA bis 8mA	100µA

Typischer Verbrauch bei ATMega2561, VCC 5 Volt:

Board, 16 Mhz, inkl. Display, RTC, 100% Displaybeleuchtung: ~ mA

Board, 16 Mhz, inkl. Display, RTC, 50% Displaybeleuchtung: ~ mA

Board, 16 Mhz, inkl. Display, RTC, 20% Displaybeleuchtung: ~ mA

Board, 16 Mhz, inkl. Display, RTC, 10% Displaybeleuchtung: ~ mA

Board, 16 Mhz, inkl. Display, RTC, ohne Displaybeleuchtung: ~ mA

Board, 16 Mhz, inkl. Display, RTC, ohne Displaybeleuchtung, Standby: ~ mA

Typischer Verbrauch bei ATMega2561, VCC 3,7 Volt:

Board, 16 Mhz, inkl. Display, RTC, 100% Displaybeleuchtung: ~ mA

Board, 16 Mhz, inkl. Display, RTC, 50% Displaybeleuchtung: ~ mA

Board, 16 Mhz, inkl. Display, RTC, 20% Displaybeleuchtung: ~ mA

Board, 16 Mhz, inkl. Display, RTC, 10% Displaybeleuchtung: ~ mA

Board, 16 Mhz, inkl. Display, RTC, ohne Displaybeleuchtung: ~ mA

Board, 16 Mhz, inkl. Display, RTC, ohne Displaybeleuchtung, Standby: ~ mA

Reduzierung des Stromverbrauchs

Gerade beim Batteriebetrieb wird es Ihr Bestreben sein, den Stromverbrauch während der Nichtnutzung auf ein Minimum zu senken. Je nach Ausstattung Ihres Boards verbrauchen jedoch div. Verbraucher zu viel Strom, so dass es u.U. sinnvoll ist, diese softwaregesteuert einfach komplett von der Stromversorgung zu trennen. Der Mikrocontroller benötigt je nach Standby/Sleep-Modus unterschiedlich viel Strom (bis zu wenigen μA) und etliche weitere Bauteile gehen bei Nichtgebrauch (z.B. der RS232 Transceiver, Displaybeleuchtung, Echtzeituhr etc.) in den Standby-Modus.

Einen relativ hohen Verbrauch hat vor allem der obligatorische 3V Spannungsregler mit den angeschalteten Verbrauchern Display / Displaybeleuchtung und SD Karte. Allerdings zeigt die Tabelle auf Seite 40, dass auch der optionale CAN Transceiver für ein batteriebetriebenes Gerät relativ viel Strom verbraucht. Sollten Sie kein CAN auf dem Board haben, können Sie auf die Abschaltungsfunktion per MOSFet (siehe nächstes Kapitel) verzichten – dann schalten Sie lediglich den 3V Spannungsregler über einen freien Port des Mikrocontrollers (siehe übernächster Absatz).

Stromreduktion Displaybeleuchtung: Einen hohen Einfluss auf den Stromverbrauch hat die Displaybeleuchtung. Diese benötigt je nach Beleuchtungsstärke max. rund 85mA. Die Displaybeleuchtung kann über Port B7 abgeschaltet bzw. per PWM gedimmt werden. Bei einem Pulsverhältnis von 10% braucht sie nur noch 1/10 des Stroms, dann also nur unter 9mA. Komplett abgeschaltet benötigt die Beleuchtung auch keinen Strom mehr. Mehr dazu erfahren sie im Anhang im Kapitel Dimmen der Displaybeleuchtung auf Seite 71.

Die größten Verbraucher im Standby:

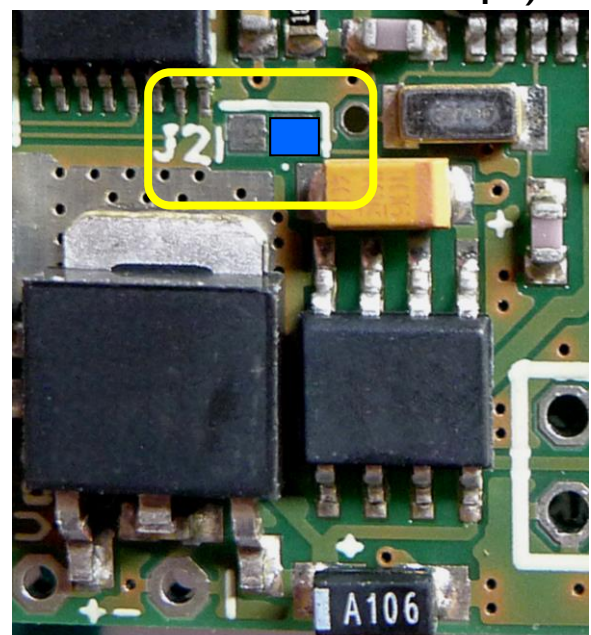
- 3V Spannungsregler (mit Verbrauchern Display und SD Karte) ($> 1\text{mA}$ Eigenverbrauch)
- CAN Chip (bis 14 mA)

Abschaltung des 3V Spannungsreglers (Verbrauch dann nur noch ca. $310\mu\text{A}$)

Der 3V Regler kann relativ leicht in den Standby-Modus versetzt werden. Hierzu öffnen Sie mit einem Skalpell oder feinen Messen die dünne Leiterbahn in der Mitte des Jumpers J2 und verbinden das im nebenstehenden Foto blau markierte Feld des Jumpers durch eine Verbindung (also durch eine Leitung) mit einem beliebigen Port.

Schalten Sie per Software den Port auf „Ausgang“. Ein „Low“ auf dem Port schaltet den 3V Spannungsregler ein, ein „High“ versetzt ihn in den Standby-Modus.

Tipp: Beachten Sie auch die Alternative am Ende der übernächsten Seite.

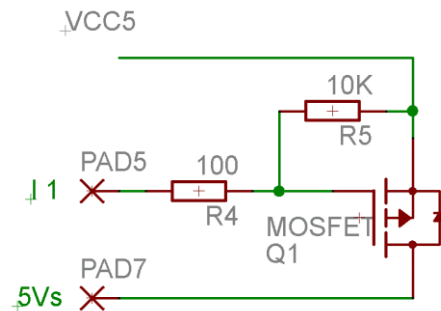


Achtung: mit dem 3V Regler werden dann auch Display und SD Karte abgeschaltet. Alle Schreibvorgänge auf der SD Karte müssen vorab beendet sein und alle geöffneten Dateien auf der SD Karte müssen zuvor geschlossen werden – ansonsten ist mindestens die aktuelle Datei verloren, evtl. aber auch das ganze Dateisystem (FAT) durcheinander. SD Karte und Display müssen nach dem Einschalten des 3V Reglers zuerst neu initialisiert werden.

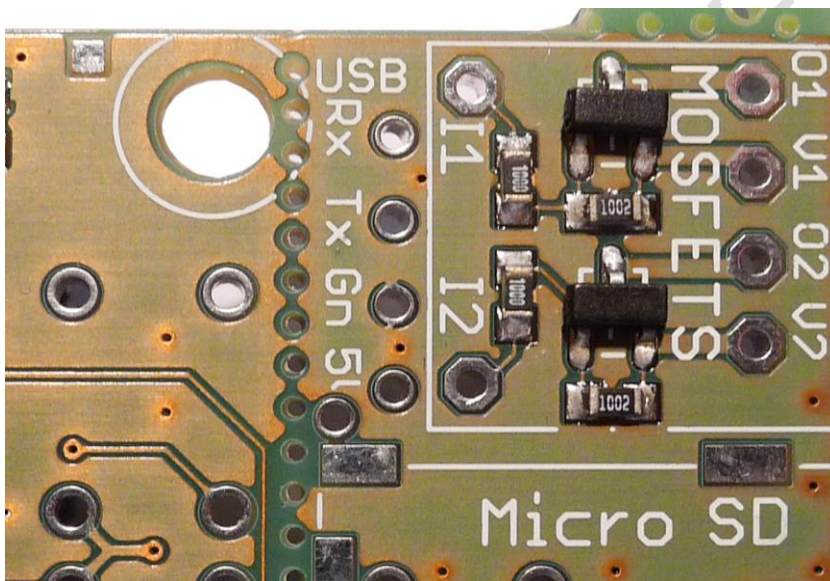
Abschaltung durch MOSFet

Das D074x-Board wird optional mit einem MOSFet geliefert, welches es erlaubt, verschiedene Verbraucher per Software von der Stromversorgung abzuklemmen.

Der MOSFet wird über das mit „I1“ gekennzeichnete Lötpad angesteuert. Es legt dann an das Pad „5Vs“ die geschaltete Spannung an.



Die Position des MOSFets und der Widerstände erkennen Sie im nachfolgenden Foto.



Altes Foto von Rev. A – Austauschen!

Um einen der beiden MOSFet zu schalten, legen Sie an I1 oder I2 (Input) einen beliebigen Port an. V1 bzw. V2 erhält die Spannungszufuhr, O1 bzw. O2 (Output) sind dann die geschalteten Ausgänge der MOSFets. Standardmäßig werden dort MOSFets für max. 12V und 4A verbaut. Sollten Sie andere Anforderungen haben, sprechen Sie uns bitte an.

Um den MOSFET durchzuschalten, muss der entsprechende Port als Ausgang geschaltet werden und auf Low gezogen werden (Low= einschalten, High= ausschalten).

Abtrennen des CAN Transceivers von der Versorgungsspannung (und Ein-/Ausschalten per MOSFet).

bis einschließlich Board Rev. B:

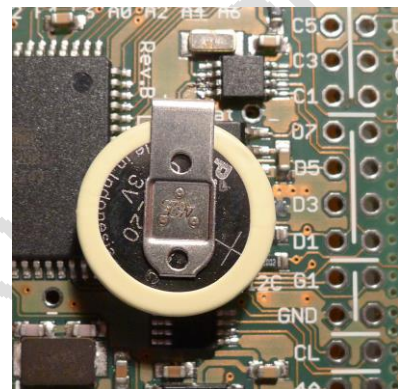
Ein Beinchen (Pin Nr. 3) am CAN Transceiver ablöten und hochheben und dort eine Leitung anlöten und zum MOSFet legen.

ab Board Rev. C:

eine Leiterbahn mit der bestehenden 5V Versorgung durchtrennen und an dem vorhandenen Lötpad über dem CAN Transceiver eine Leitung zum MOSFet legen.

Ein Foto Bild folgt, sobald Rev. C vorliegt.

Anmerkung: beide o.g. Work-Arounds setzen voraus, dass Sie vorher eine evtl. vorhandene Batterie der Echtzeituhr auslöten, denn diese liegt im eingelöteten Zustand über diesem Leiterplattenbereich. Sollten Sie vorher schon wissen, dass diese „Operation“ notwendig sein sollte, so löten Sie die Batterie erst danach ein. Aus diesem Grunde liefern wir die Batterie der Echtzeituhr auch noch nicht eingelötet mit.



Alternatives Schalten des 3V Reglers

Statt den 3V-Regler auf Standby zu schalten wie oben auf der vorletzten Seite beschrieben (310µA Verbrauch) können Sie ihn natürlich durch den MOSFet ebenfalls komplett von der Stromversorgung abtrennen.

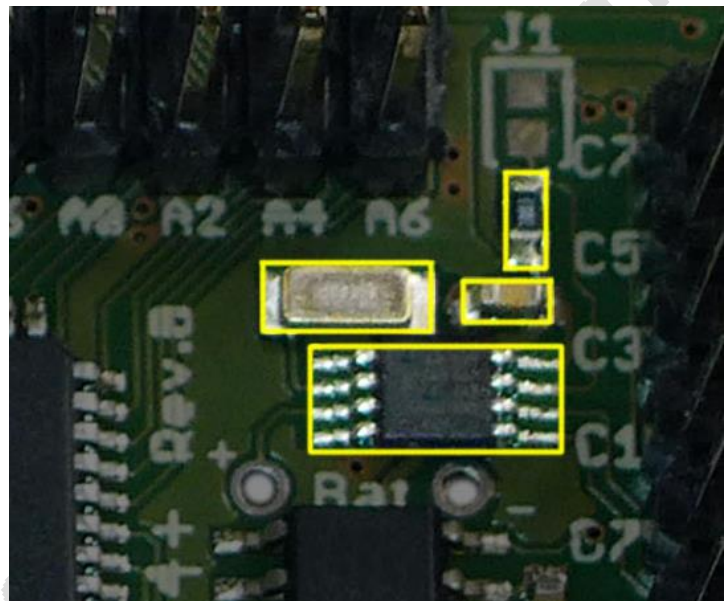
Der 3V Spannungsregler im SO8-Gehäuse befindet sich zwischen dem ISP Stecker und dem optionalen 5V Spannungsregler.

Am einfachsten ist es, das Beinchen von Pin 8 des Reglers abzulöten und nach oben zu biegen sowie dann von Pin 8 eine Leitung zum geschalteten Spannung am MOSFet s zu legen .

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 D074+ Board bietet eine solche Echtzeituhr zusammen mit einer Miniatur-Lithiumbatterie als Option an. Aus Platzgründen befinden sich sämtliche Bauteile unterhalb der Batterie. Das nebenstehende Foto zeigt das Board ohne Batterie. Damit Sie auf diesen Platinenbereich noch zugreifen können, liefern wir die Batterie lose mit, d.h. Sie müssen sie selber einlöten.



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 von uns verwendete Echtzeituhrbaustein enthält zusätzlich noch frei nutzbare 56 Byte RAM, die Sie nach Belieben beschreiben können (beliebig oft! – RAM hat im Gegensatz zu einem Eeprom oder Flash keine begrenzte Lebensdauer).

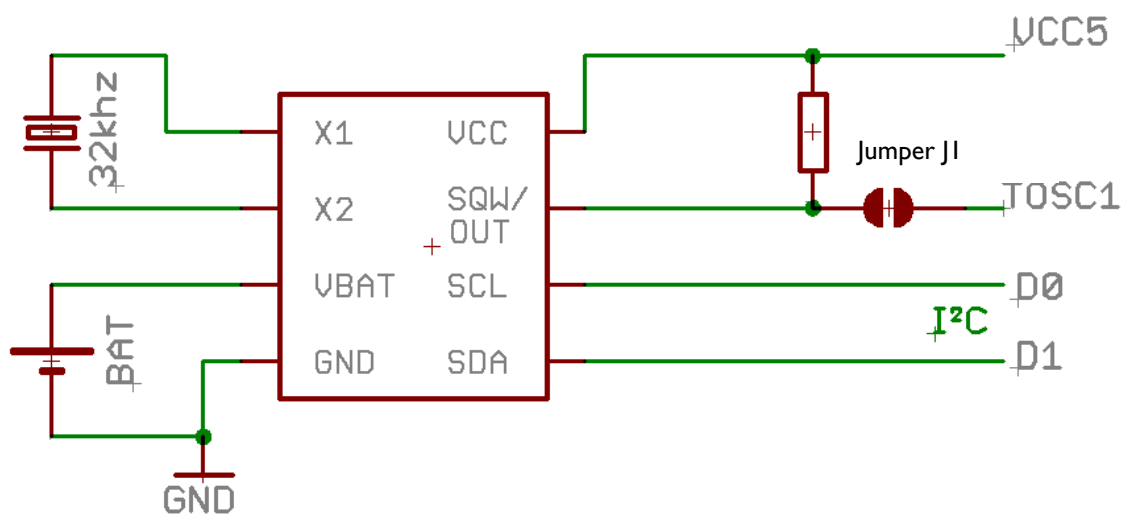
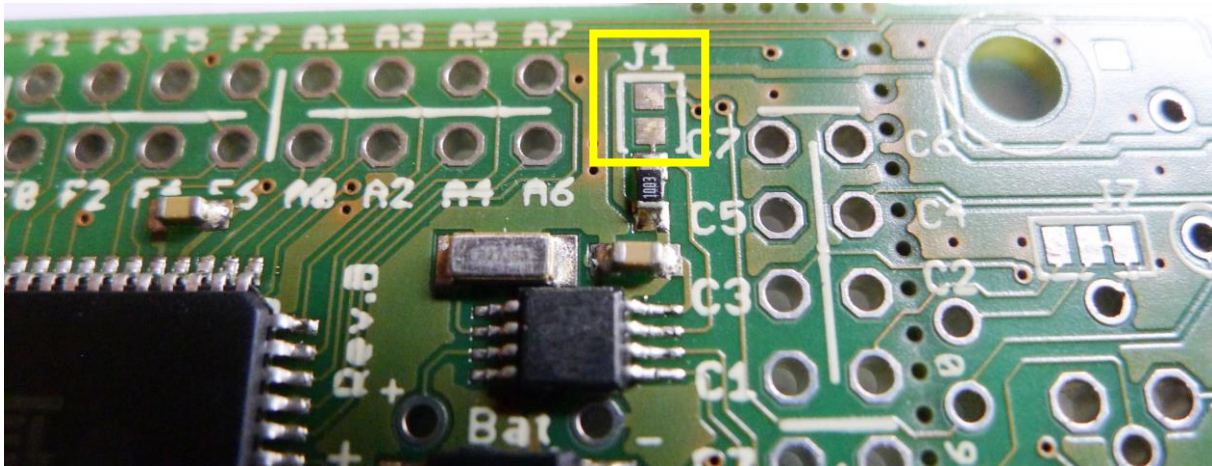
Auch dieses RAM ist gepuffert – behält seinen Inhalt also auch Unterbrechung der Spannungsversorgung. Das ist sehr praktisch, denn Sie können so z.B. während des Programmablaufs permanent Konfigurations- und Zustandsdaten in diesem RAM ablegen – bei einem Neustart kann das Mikrocontrollermodul dann daraufhin 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 D074 an TOSCI des Mikrocontrollers gelegt wurde.

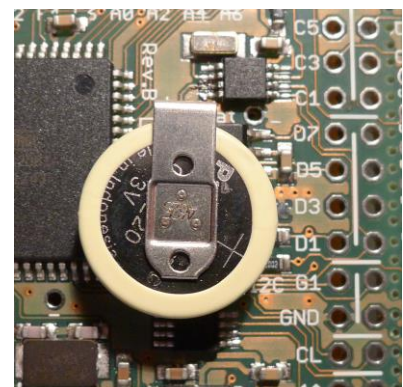
Mit dem Schließen des Jumpers „J1“ (dieser liegt zwischen ISP Programmstecker und Mikrocontroller) verbinden Sie TOSCI mit dem Clock-Signal der Echtzeituhr. Hiermit ersparen Sie sich einen eigenen Uhrenquarz direkt am Mikrocontroller, denn nun erhält der Mikrocontroller das Signal der Echtzeituhr (Anmerkung: 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, der RTC Jumper sollte also offen sein.
Bei einem ATmega128 ist dann ein eigener Uhrenquarz zwischen TOSC1 und TOSC2 notwendig!

Port G3 ist dann mit einem ATmega128 nicht mehr nutzbar (siehe auch Seite 25).



Das neue D074+ ist für verschiedene Batteriegrößen als Pufferbatterie geeignet. Grundsätzlich liegt der Pluspol der Batterie immer links unterhalb des RTC-Chips; das Pad ist mit einem „+“ gekennzeichnet (siehe Foto auf der nächsten Seite, dort rot eingekreist). Direkt rechts daneben befindet sich ein Lötpad, welches mit „-“ gekennzeichnet ist – hier könnten Sie z.B. eine Verkabelung zu einer größeren Lithiumbatterie durchführen (die u.U. mehrere Jahrzehnte puffern kann). Ansonsten gibt es für einlötbare Batterien mit Lötflähen neben dem o.g. obligatorischen Pad für den Pluspol zwei entsprechend der unterschiedlichen Batteriegröße platzierte Pads für den



Minuspole auf der Platine. D.h. Sie stecken die den Pluspol der Batterie in das mit „+“ gekennzeichnete Feld und den Minuspole je nach Größe der Batterie in eines der anderen beiden mit „-“ gekennzeichneten Felder. Die beiden möglichen Batterietypen zum direkten Aufstecken auf das Board haben einen Durchmesser von ca. 13 mm (Abstand der Lötpins 10-11mm) mit 48mAh Kapazität oder alternativ mit ca. 20 mm Durchmesser (Abstand der Lötpins ca. 15,2 mm) und 225 mAh Kapazität.

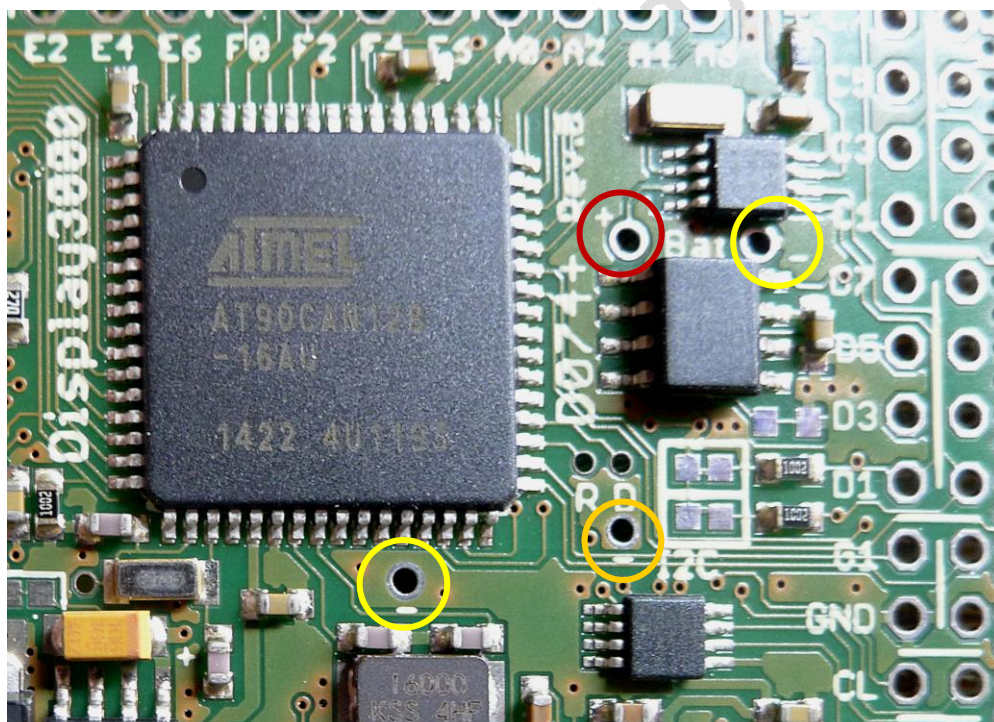
Das folgende Bild zeigt die Position des Löt pads für den Pluspol (rot) sowie die drei möglichen Pads für den Minuspole (gelb bzw. orange). Die mitgelieferte Batterie (Knopfzelle mit Lötbeinchen) ist für den roten und den orangenen Löt punkt vorgesehen. Der Minuspole der möglichen größeren Alternativzelle mit 20mm Durchmesser wird dann mit dem unteren linken gelb markierten Löt pad verlötet.

Anmerkung: Der Minus-Pole der Batterie liegt an Masse, d.h. selbstverständlich können Sie für den Minuspole einer RTC-Batterie jede beliebige Masseverbindung nutzen – sie müssen also nicht zwangsweise einen der gelben/orangenen Löt punkte nehmen. Das rot umrandete Löt pad für den Pluspol der Batterie ist allerdings obligatorisch.

Beispieltypen der Batterien:

13mm Durchmesser: Panasonic BR-1225/HCN (wie Foto auf der vorigen Seite)

20mm Durchmesser: Panasonic CR-2032/HSN



SD Kartenmodul

Mit dieser Erweiterung können Sie zukünftig SD oder Micro 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 vorhandene Software kann oft 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

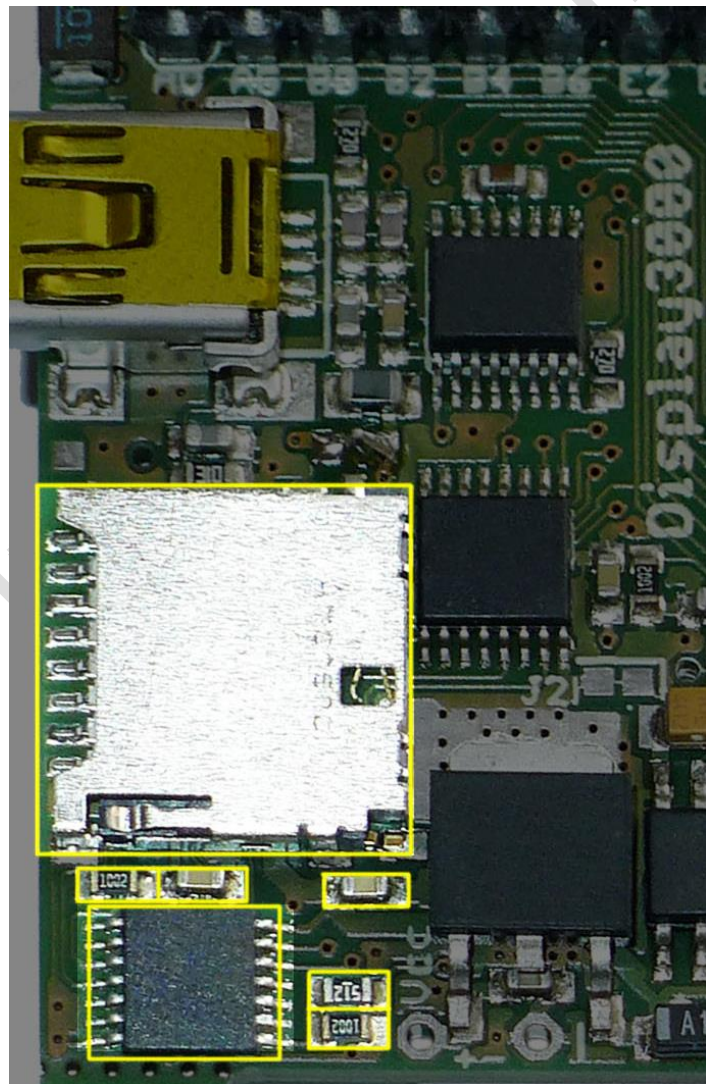
Warum ein bidirektionaler Pegelwandler mit Tristate Outputs?

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 D074+ vorhanden ist, obligatorisch.

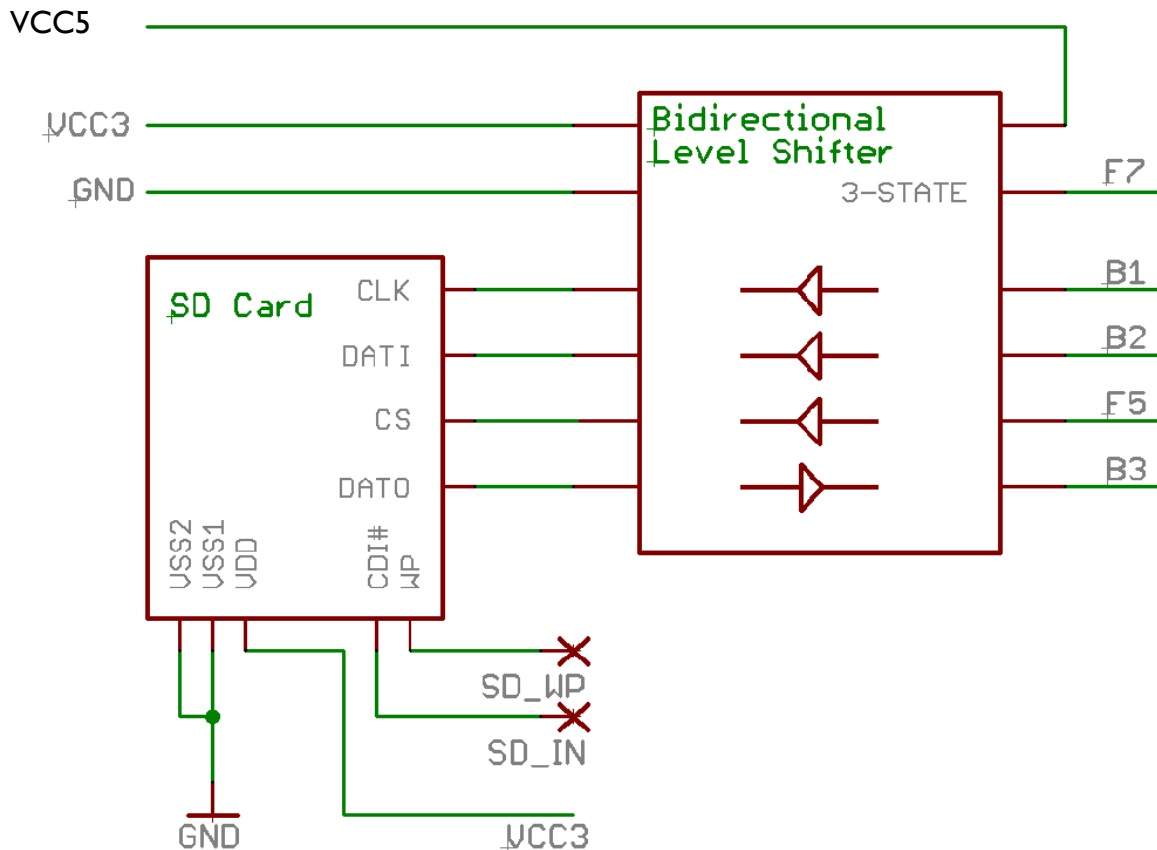
Die direkte Verbindung zwischen eine 5 Volt Mikrocontroller und einer SD Karte verbietet sich, da diese durch den 5 Volt Pegel der Signale zerstört würde. Die Signalpegel der Datenleitung müssen also von 5 Volt auf 3 Volt (und umgekehrt) umgesetzt werden. Bei einem 3 Volt-Mikrocontroller könnten Sie 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 D074 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 Beschriftung).

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.

Alternative SD Kartenpositionen

Fotos einfügen

Aufgrund der Kundenrückmeldungen durch unterschiedliche Einsatzszenarien ermöglichen wir nun auf dem neuen D074+ vier verschiedene SD-Kartenpositionen.

Diese können sein:

1. Micro-SD am oberen Platinenrand (neben USB) als Push-Pull-Ausführung
2. Micro-SD am unteren Platinenrand auf der Bauteilseite als Push-Push-Ausführung
3. Micro-SD am unteren Platinenrand auf der Displayseite als Push-Push-Ausführung
4. Normale SD am unteren Platinenrand auf der Displayseite als Push-Pull-Ausführung

Wir können Ihnen auf Anfrage mehrere Positionen mit dem SD-Karten-Connector bestücken, so dass Sie bei Prototypen dann flexibel sein können. Bitte beachten Sie aber, dass Sie immer nur eine Karte gleichzeitig betreiben können, d.h. die anderen Slots müssen dann leer sein. Theoretisch könnten Sie mit einer kleinen Modifikation (die wir Ihnen auf Wunsch gerne mitteilen) mit dem D074+ auch 2 Karten gleichzeitig betreiben (Micro SD oben und eine weitere in einer der drei Positionen unten). Lesen und Schreiben können Sie jedoch immer nur eine Karte gleichzeitig. Sobald Sie auf die zweite Karte zugreifen möchten, müssen zuerst alle Dateien auf der anderen Karte geschlossen werden und Sie müssen die zweite Karte initialisieren, damit die Software das abweichende Dateisystem erkennt.

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 u.U. (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 die SD Karte an den SPI Bus anschalten bzw. trennen (für Zugriff auf das Display) sondern auch den SPI Bus umkonfigurieren! Die SD Karte wird an den Bus mittels Port F7 angeschlossen/getrennt.

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

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:

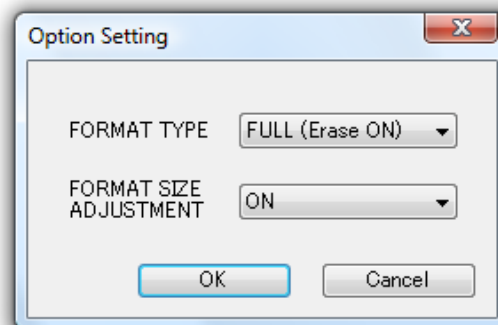
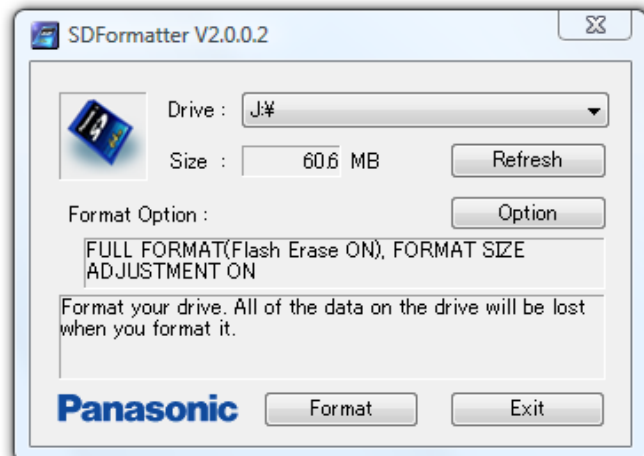
<http://panasonic.jp/support/global/cs/sd/download/ftp/manual2007e.pdf>

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).

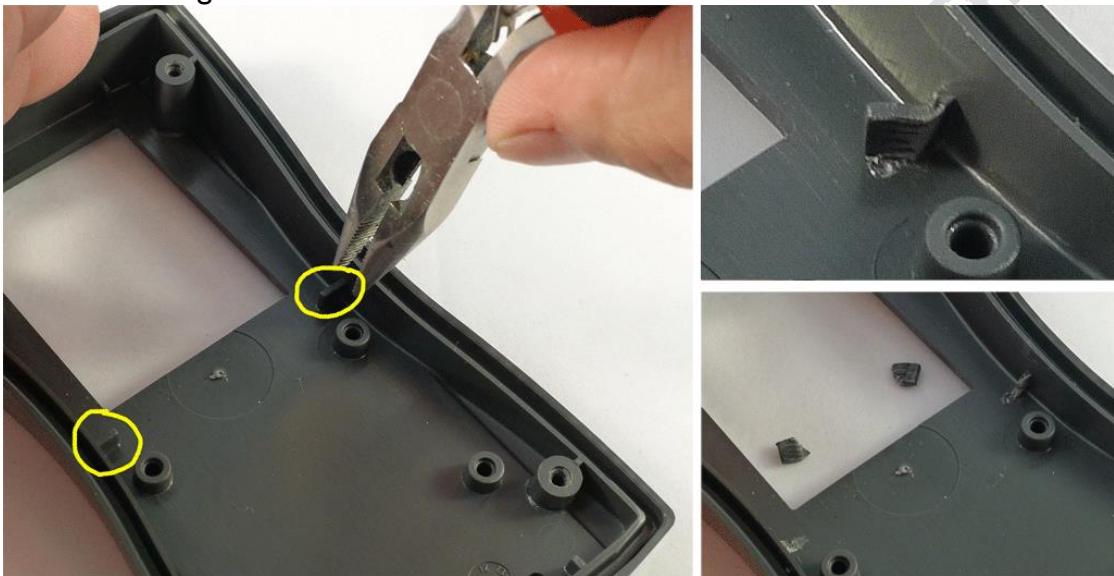


Einsatz im Gehäuse GBS400 / GBS401

Das Modul D074+ 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) Im Gehäuse versperren innen zwei kleine Stege den Platz für das Display. Diese lassen sich mit einer Zange einfach seitwärts wegdrehen (Zange ansetzen und 1x hin und her drehen). Das ist eine Arbeit von 5 Sekunden und keine Angst: Sie können nichts beschädigen.



- b) Der Durchbruch für die Folientastatur ist bereits standardmäßig erfolgt. Eine Gehäuseoberschale ohne diesen Ausschnitt für die Folientastatur ist aktuell nicht erhältlich (ab einer Bestellung von ca. 50 Stück können wir aber auch diese liefern). Bei gewünschter Nutzung ohne Folientastatur kleben Sie den Ausschnitt am besten zu (z.B. mit einem Aufkleber Ihres Firmenlogo etc.).
- c) Die mittleren Montage„ohren“ des D074x+ müssen für den Einsatz im Gehäuse abgebrochen werden, die oberen und unteren „Ohren“ können i.d.R. verbleiben. Beim D074s+ müssen die unteren Montageohren abgebrochen werden. Diese Montagehalter sind vorperforiert und lassen sich mit einer Zange leicht abbrechen.

Weitere Infos und Bilder zu USB / Micro SD !!!! fehlen

d) Für eine SD Karte oder einen USB Stecker müssen Sie noch die Aussparungen in das Gehäuse sägen / fräsen. Wir empfehlen hier:

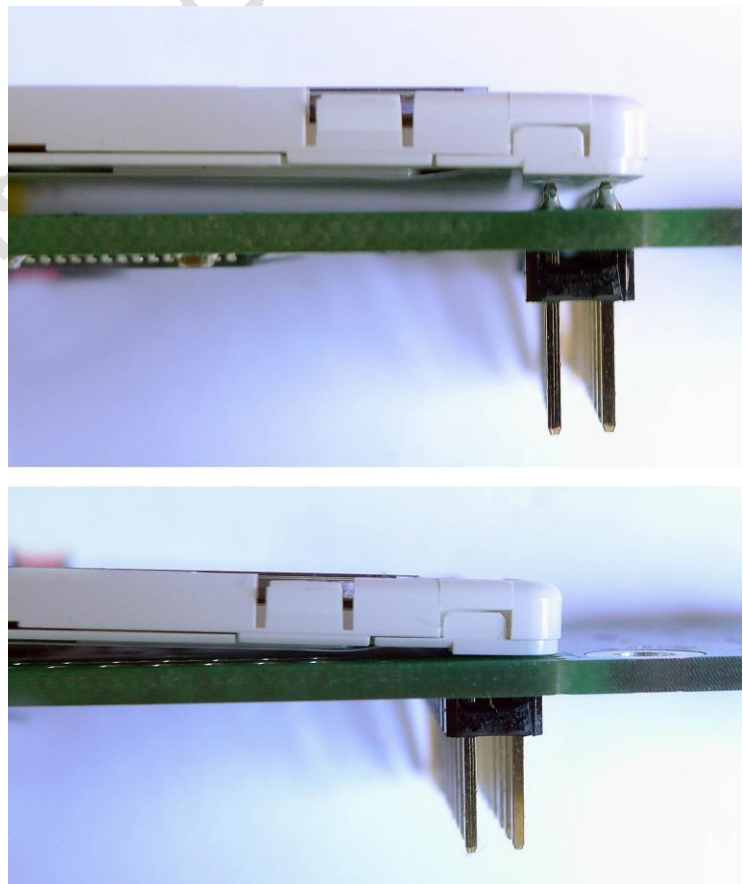
- a. Für die SD-Karte reicht es, einfach nur am Gehäuserand der Oberschale mit einer Feile ein paar mm abzufeilen. Dies geht schnell und sieht sauber aus.
- b. Auch der Durchbruch für den USB Stecker erfolgt direkt vom Gehäuserand aus, was das Fräsen, Einsägen oder Feilen wesentlich vereinfacht. (Das obige Foto zeigt noch ein BS600 Gehäuse, das Prinzip beim BS400 Gehäuse ist jedoch das gleiche).

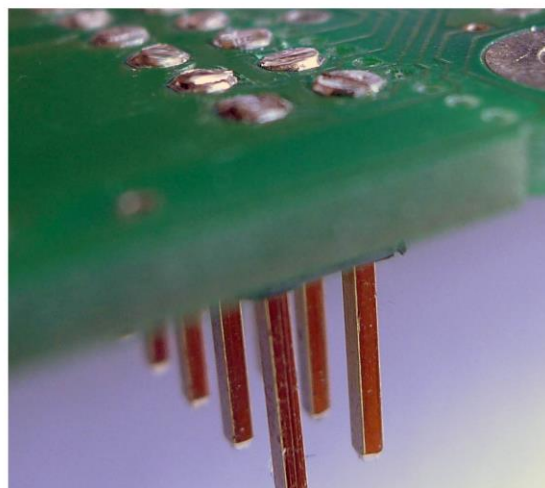
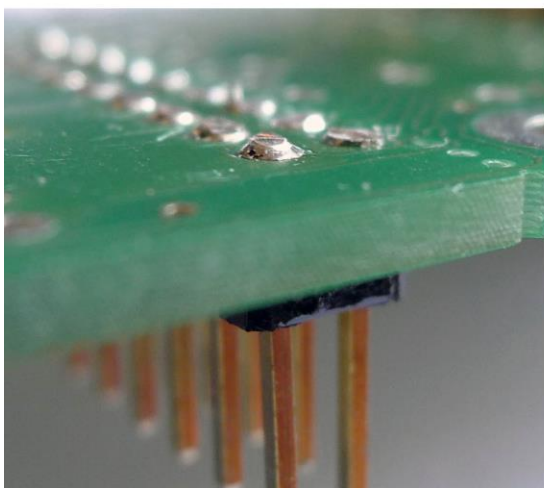
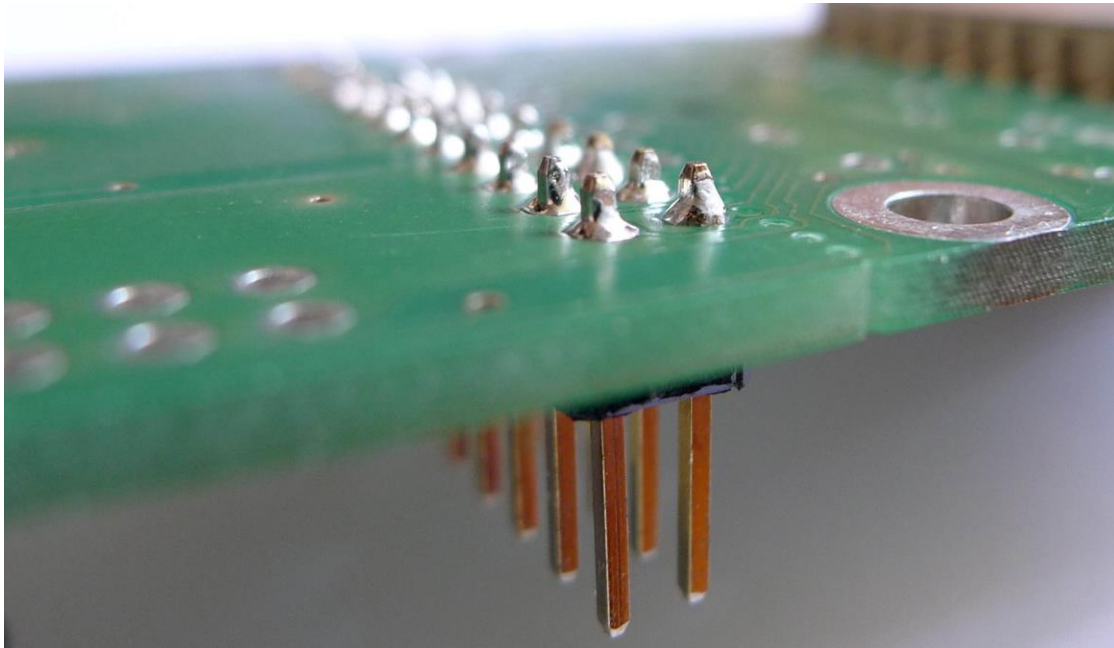


Anmerkung: Ab einer Bestellung von ca. 50 Gehäusen können wir die Schalen auch mit beliebigen Einfräsungen / Bohrungen etc. versehen.

Aktualisieren für alle SD-Karten Optionen, neue Fotos

- e) Wenn Sie Pins oder Kabel in die Portstecker für Port A oder Port F einlöten und unser Gehäuse BS400 nutzen, so stellen Sie bitte sicher, dass diese nicht das Display berühren, wenn dieses aufgesetzt wurde. Das Display liegt beim Einbau schräg und würde diese Löt-pins berühren – hier kann u.U. eine Beschädigung des μ C Boards auftreten, denn das Display hat an dieser Stelle 3V und GND offen liegen, Um das zu vermeiden, müssen Sie die Pins von Port A und F (am besten vor dem Verlöten) kürzen (abknipsen). Sicherheitshalber empfehlen wir Ihnen, über diese Löt-punkte noch einen Streifen Isolier-band zu kleben.





Oben: Üblich, aber für den Einsatz im Gehäuse nicht möglich

Unten links: abgeknipst, aber die Pins stehen noch ein wenig zu viel heraus. Besser nochmal nacharbeiten, so dass die Pins so wie unten rechts aussehen. Oder noch besser: knipsen Sie die Pins vor dem Einlöten bündig mit der Platine ab (wie unten beschrieben).

Unten rechts: OK

Tipp – Ändern Sie die Reihenfolge: Wenn Sie eine Pinreihe einlöten möchten, löten Sie nur je einen Kontakt an den beiden Rändern ein um die Pinreihe zu stabilisieren. Dann knipsen Sie alle Pins soweit wie möglich ab und löten diese danach erst ein. Da das Lötzinn in das durchkontaktierte Lötäuge fließt, haben die Pins weiterhin guten Kontakt und stabilen Halt. Hierdurch bekommen Sie wunderschöne flache Lötunkte und das Abknipsen geht leichter von der Hand.

Übersicht Jumper / Lötbrücken D074+

Auf den vorhergehenden Seiten wurden die verschiedenen Felder mit Jumpfern / Lötbrücken bereits angesprochen. Nachfolgend finden Sie nochmals eine Übersicht über diese Felder:. Zur Positionierung gilt: Legen Sie das Board quer mit dem Display auf der linken Seite.

Jumper	Positionierung	Siehe Seite	Erläuterung
J8+9 (I2C)	Bauteilseite: rechts neben dem Mikrocontroller. Die Jumper sind mit „I2C“ beschriftet	27	Verbindet die I ² C-Pullupwiderstände mit 5V. Öffnen ist sinnvoll sofern I ² C nicht benötigt wird (also auch keine Echtzeituhr) und die Ports D0 und D1 genutzt werden sollen oder wenn bereits andere angeschlossene I ² C Komponenten mit Pullup-Widerständen ausgestattet sind. Default: geschlossen
J2	Bauteilseite: links unterhalb des Mikrocontrollers	42	Standby-Möglichkeit für den 3V Spannungsregler. Default: geschlossen
J1	Bauteilseite, rechts neben Port A6 / A7	25	Verbindet TOSCI mit dem Clock-Ausgang der Echtzeituhr. Nur nutzbar, wenn kein Uhrenquarz verbaut wurde und wenn die optionale Echtzeituhr verbaut ist Default ist offen.
J7	Bauteilseite: unbeschriftet. links neben den Pads „WP“ und „In“;	23	Jumper fungiert als Umschalter. Per Default werden die gemeinsamen Kontakte der Taster über einen 470 Ohm Widerstand mit Masse verbunden. Umgeschaltet mit Port C.0 verbunden (notwendig z.B. bei Nutzung einer 3x4 oder 4x4 Matrixtastatur). Default verbunden mit gemeinsamer Taster-Masse (GND über 470 Ohm)
J5	Displayseite oberhalb des ISP Steckers (Beschriftet mit 4 / U)	32	Legt fest, ob D2 (RX1) mit USB oder dem RS485 Transceiver verbunden werden soll. Default: USB
J6	Displayseite oberhalb des ISP Steckers (Beschriftet mit 4 / U)	32	Legt fest, ob D3 (TX1) mit USB oder dem RS485 Transceiver verbunden werden soll. Default: USB
J3	Displayseite oberer Rand	39	Ein Öffnen trennt AVCC von der Verbindung zu VCC5. Default: geschlossen
J4	Displayseite oberer Rand	39	Ein Öffnen trennt AGND von der Verbindung zu GND. Default: geschlossen

Übersicht Portbelegung

Bitte nutzen Sie hierfür die separate Excel-Tabelle auf Ihrer CD: *Portbelegung DS074+.XLS*

Technische Daten Display-Modul-Bausätze:

Artikel D074+:

Maße: 108 x 50 mm (108x38mm ohne Montageohren)
ca. 8 mm hoch (mit Display, ohne Taster)

Gewicht: ca. 35-40 Gramm (schwankt je nach Ausstattung)

Versorgungsspannung mit 5V Spannungsregler:
3 bis 18 Volt Gleichstrom

Versorgungsspannung ohne 5V Spannungsregler:
3 bis 5 Volt Gleichstrom
(bitte Kapitel über benötigte Spannungen beachten)

Mikrocontroller: Atmel ATmega128a
(Option: ATmega2561, AT90CAN128 ^{*1} ^{*2})
128 KByte Programmspeicher (256 KByte ^{*2})
4 KByte RAM (8 KByte ^{*2})
2 KByte Eeprom
max. 16 MHz Takt (je nach bestellter Option)

Display: 220 x 176 Pixel, 65.536 Farben
Aktive diagonale Fläche: 2" (51 mm)
Hintergrundbeleuchtung: 3 weiße LED
Luminanz: 350 cd/m²
Dot Pitch H & V: 0,18 mm
Kontrast Verhältnis: 500
Vorgesehene Betrachtungsrichtung: 12 Uhr (Controller: 12 Uhr)
Lebensdauer: mind. 100.000 Stunden
Lebensdauer LED-Beleuchtung (25°C): mind. 20.000 Stunden (=Helligkeit ist auf 70% zurückgegangen). Tipp: die Lebensdauer steigt beträchtlich bei gedimmtem Betrieb (50% PWM = doppelte Lebensdauer)

Temperaturbereich Display Lagerung: -30°C bis +80°C

Temperaturbereich Display Betrieb: -20°C bis +70°C

Optional gegen Aufpreis ist dieses Board verfügbar mit:

^{*1}) ATmega2561 (256 KByte Flash, 8 KByte RAM, 8 KByte Eeprom)

^{*2}) AT90CAN128 (quasi wie Atmega128 aber mit CAN-Schnittstelle)

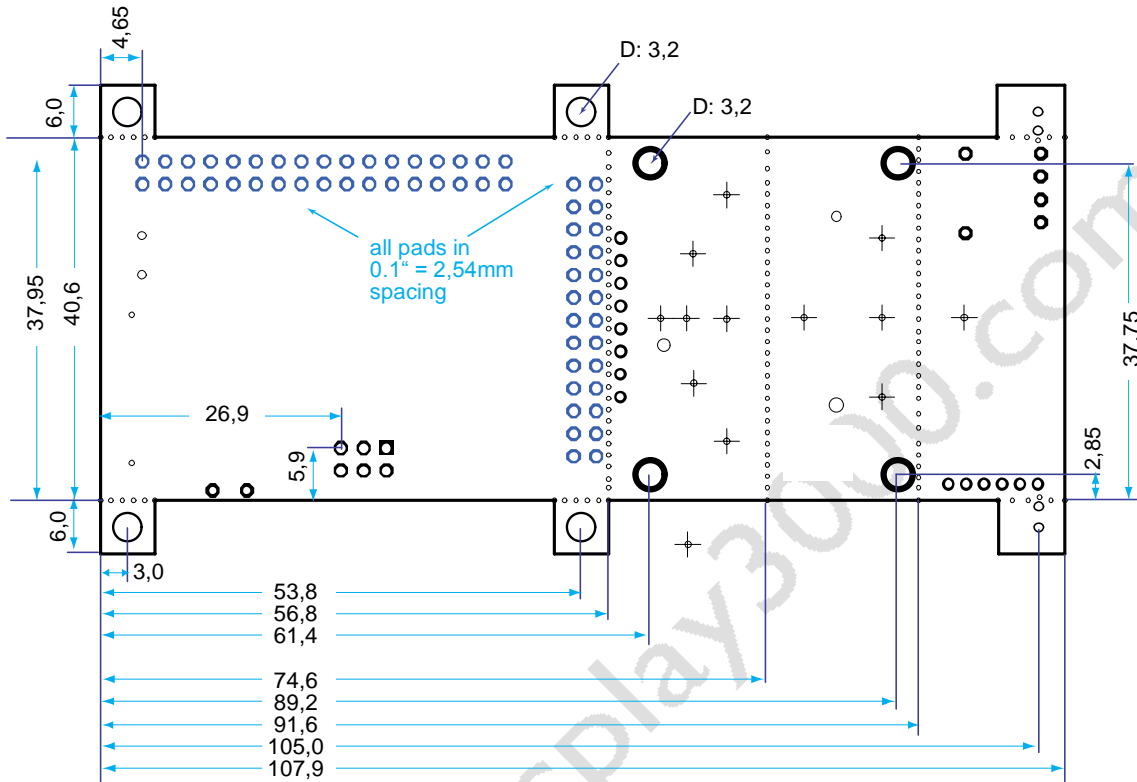
Der Atmel Mikrocontroller wurde von uns gegenüber der Standardauslieferung wie folgt umprogrammiert:

- ATMega128A: **Mega103 Mode** wurde abgeschaltet
- ATMega2561 und AT90CAN128: Fuse **Clockdiv8** 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 mit einem Demoprogramm vorprogrammiert, welches Controller, Display, Echtzeituhr und SD Karte testet. Den Source Code hierfür (Bascom) erhalten Sie auf Anfrage von uns.

Maßzeichnung (siehe auch separate PDF im 1:1 Maßstab auf der CD)



Oft gestellte Fragen (FAQ):

Ich möchte das Board in ein eigenes Gehäuse bauen. Dazu müsste das Display versetzt betrieben werden. Wie kann ich das bewerkstelligen?

Wir bieten hierfür eine kleine Adapterplatine an. Damit können Sie das Display auch vom Board abgesetzt betreiben. Im Anhang finden Sie hierzu ein Kapitel mit mehr Informationen.

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 I²C benötigen. I²C ist z.B. notwendig, wenn Sie die optionale Echtzeituhr ansprechen möchten. Brauchen Sie kein I²C, können Jumper J1 und J2 geöffnet werden (im Lieferzustand geschlossen). Siehe auch Seite 27.

Ich kann die Ports D5 und D6 nicht benutzen

Vermutlich haben Sie einen AT90CAN128 Controller. D5 und D6 sind die Datenleitungen für den CAN Transceiver und können nicht zugleich mit CAN genutzt werden.

Ich habe Probleme bei der Nutzung der Ports E0 und E1

Haben Sie Ihren Programmieradapter abgezogen? Dieser nutzt die gleichen Ports E0 und E1 und stört evtl. die Nutzung dieser Ports. Also: Programmieradapter bitte entfernen.

Ich kann nicht auf die Echtzeituhr zugreifen

Bitte prüfen Sie, ob die I²C Jumper geschlossen sind (Lieferzustand geschlossen). Diese verbinden die danebenliegenden I²C-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 oder die Nutzung von Aref in der Software nicht definiert. Bitte lesen Sie das Kapitel *Analogeingänge* auf Seite 39.

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 messe an den Rx / Tx Leitungen von RS232 keine Spannung

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! Zu Testzwecken können Sie an Rx z.B. 5V anlegen.

Ich habe Probleme bei der Nutzung von RS232

Haben Sie Ihren Programmieradapter abgezogen? Dieser nutzt die gleichen Ports RX0 und TX0 und stört evtl. die Nutzung von RS232. Also: Programmieradapter bitte entfernen.

Ich kann CAN nicht nutzen

- 1) Haben Sie das Board mit AT90CAN128 Mikrocontroller bestellt?
- 2) Haben sie einen Abschlusswiderstand eingelötet (nur wenn notwendig)?

Das Board resettet sich nach einer Weile

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 5V Spannungsregler zu heiß und geht in die Notabschaltung. Bitte lesen Sie das Kapitel *Spannungsregler* auf Seite 15. 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 untere Rand des Displays ist der Displaycontroller. Dieser ist lichtempfindlich. Er ist standardmäßig mit dunklem Klebeband verdeckt. Verbauen Sie das Modul so, dass dieser Bereich verdeckt ist, und prüfen Sie die Abdeckung.

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, USB Stecker des PCs, Heizkörper, in der Werkstatt die Masse des Oszilloskops 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.

Probleme mit Timer 0 bzw. Timer 2

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 J1 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, J1 geschlossen, 32 Khz Signal an SQW/Out ist vorhanden:
 - a. Controller ist ein ATmega128: die obige Konfiguration funktioniert hier 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 evtl. ein Uhrenquarz mit 2x Kondensatoren vorhanden (siehe Seite 25)? Jumper 1 darf dann nicht geschlossen sein !

Beim Einsatz im Gehäuse BS400 sitzt die Platine irgendwie schief.

Wichtig für einen guten Sitz im Gehäuse ist, dass das Display am unteren Rand exakt auf der Platine aufliegt (von der Seite betrachtet, muss es leicht schräg sitzen). Wenn Sie also in der unteren Pad-Reihe (Rx/Tx, Port G, Port D, Port C) Kabel oder Pins eingelötet haben, so schauen die i.d.R. auf der anderen Platinenseite zu weit heraus und drücken dort das Display am unteren Rand zu weit nach oben. Knipsen Sie die überstehenden Pins dort bis zum Platinenrand hin ab, dann passt auch alles (siehe Seite 54).

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

Autor dieses Manuals: Peter Küsters.

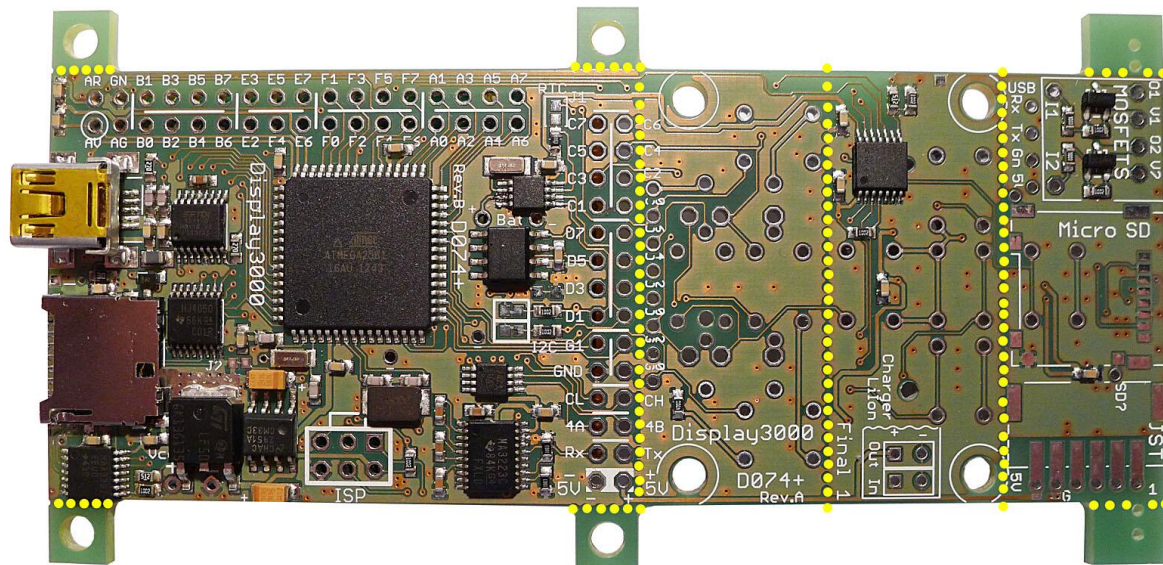
© aller Informationen: Peter Küsters

© www.Display3000.com

Anhang:

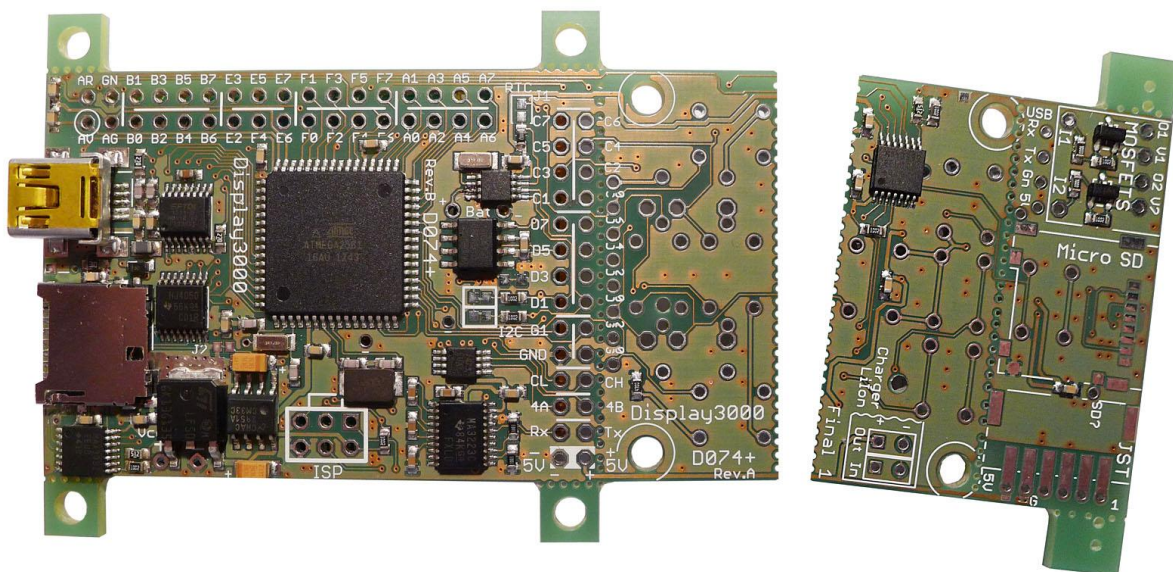
Verkleinern des D074

Das D074x+ Board besitzt (neben den Montagehaltern) noch drei abtrennbare Bereiche, die von uns bereits vorperforiert wurden, um ein leichteres Abtrennen zu ermöglichen. Beim D074s+ fehlen die beiden unteren (bzw. auf dem nachstehenden Foto ganz rechten) Bereiche bereits, hier gibt es dann nur noch einen Bereich (Joystick/Taster) zum Abtrennen. Ansonsten sind die beiden Module identisch. Die Perforationen sind nachstehend gelb gepunktet dargestellt.



Beim D074x+ können zum einen der Optionsbereich für die unteren SD-Kartenhalter sowie für Mosfets und JST-Stecker abgetrennt werden; eine weitere Verkleinerung ist möglich durch das Abtrennen des Bereichs für das User-Interface (Taster/Joystick).

Beim D074s+ kann nur der Bereich für Taster/Joystick abgetrennt werden.



Wenn Sie genauer hinsehen, erkennen Sie, dass zwischen den Perforationsbohrungen Leiterbahnen verlaufen. Irgendwie müssen die Optionen am unteren Rand (auf dem Foto rechten

Rand) ja verbunden werden. Daher müssen diese vor dem Abknicken der Platinenbereiche durchtrennt werden. Ansonsten kann es passieren, dass Sie Leiterbahnen aus dem zu behaltendem Platinenebereich mit abziehen.

Zum Abtrennen ritzen Sie bitte zuerst mit einem scharfen Messer oder Skalpell beide Platinenseiten ein. Hierzu fahren Sie einfach vorsichtig, aber mit genügend Druck der Perforation nach. Wenn Sie jede Platinenseite 2-3x nachgeschnitten haben, können Sie sie sauber abbrechen. Bewegen Sie hierzu die beiden Platinenhälften langsam 1-2x hin und her bis Sie sie getrennt haben.

Tipp: Unserer Erfahrung nach ist es am einfachsten, wenn Sie den nicht mehr benötigten Teil so in einen Schraubstock einspannen, dass die Perforation noch 5mm herausragt. Dann drücken Sie mit den Daumen auf den Platinenbereich darüber und knicken die Platine somit ab. Sobald Sie die Platine um ca. 15° geknickt haben, knicken Sie sie in die andere Richtung um eine sauberere Kante zu erhalten.

Drücken Sie nicht mit den Daumen auf vorhandene Chips, sondern auf unbelegte Platinenbereiche.

Zusatzinfos RS232 und RS485

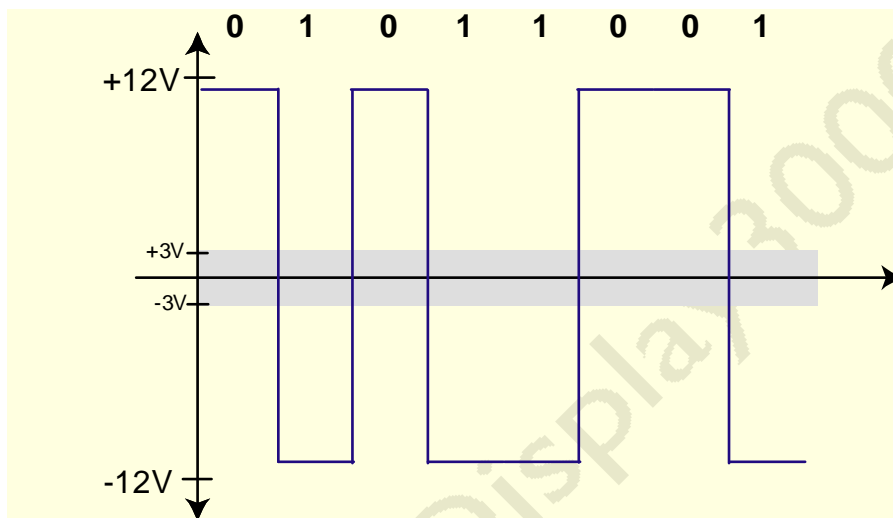
RS232

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.

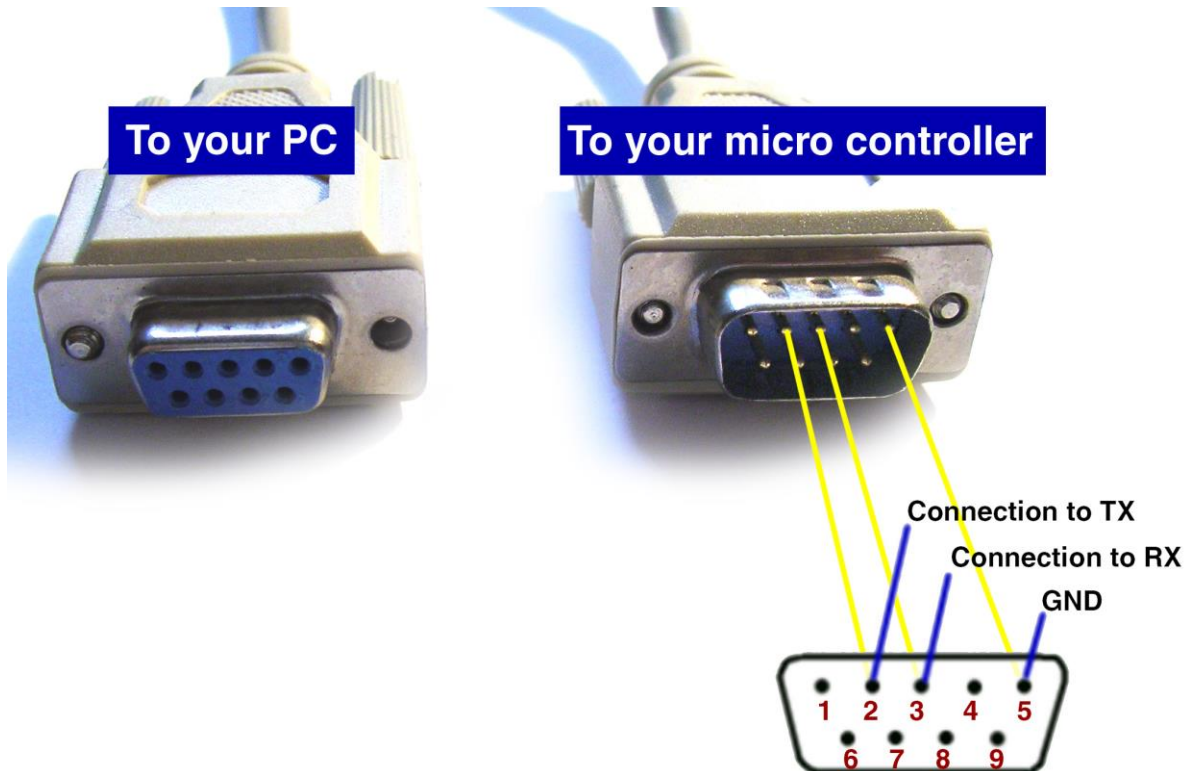
RS-232 ist ein asynchrones Protokoll, d.h. es wird keine separate Clock-Information mit übertragen. Sowohl die Sende-, wie auch die Empfangsstation müssen die Geschwindigkeit kennen (als Baud-Rate bezeichnet).

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 serielle Interfaces. Eines davon (UART1 serielle Schnittstelle Nr. 1 – also Rx1 / Tx1) kann ohne Einschränkung genutzt werden – die Ports der UART0 seriellen Schnittstelle Nr. 0 (also Rx0 / Tx0) werden jedoch auch für die ISP-Programmierung genutzt. Daher sollte der Programmieradapter beim D074x für die Benutzung der UART0 abgezogen werden. Info: Der RS232 Pegelwandler ist auf diesem Board lediglich UART0 angeschlossen.

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 erhält.

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 "COM1:" For Binary As #1   'Com1: für Rx0/Tx0
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.

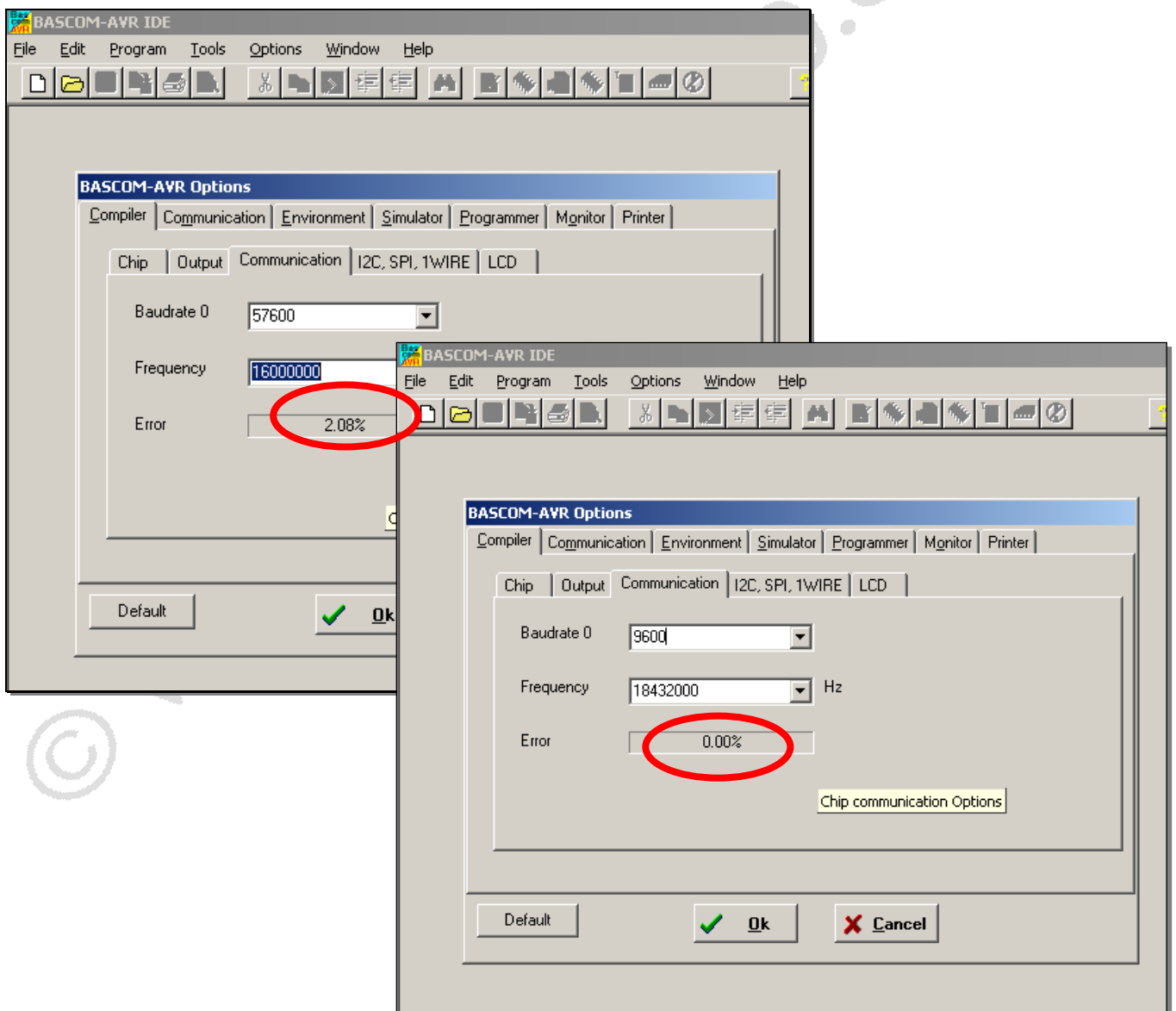
Alles über 16 MHz betreibt den Mikrocontroller jedoch über seiner Spezifikation von 16 MHz – d.h. Sie übertakten ihn. Normalerweise führt eine solch geringe Übertaktung noch zu keinem Problem, trotzdem geschieht dies immer auf eigenes Risiko. Übrigens treten evtl. Programmfehler als erstes beim Schreiben 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 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-422/485 Interface

Der Unterschied zwischen RS422 und RS485 ist im Wesentlichen: RS422 erlaubt einen Sender und bis zu 10 Empfänger am Bus, während bei RS485 bis zu 32 Devices (also Sender/Empfängerpaare) den Bus belegen dürfen.

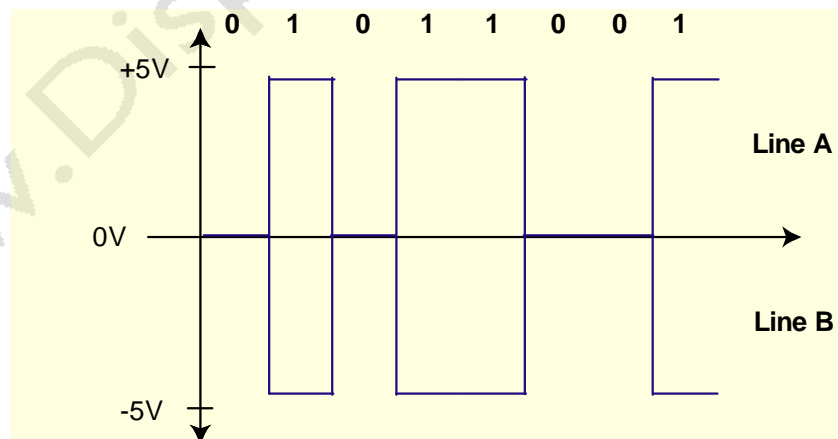
Das RS-485-Interface wird von Ihrer Software aus genauso wie die RS-232 Schnittstelle angesprochen. Daher gilt das oben bei RS232 besprochene auch ausnahmslos (mit Ausnahme der Spannungen und der Kabelverbindung) für die RS-485 Schnittstelle. Diese Schnittstelle unterscheidet sich lediglich durch das genutzte Spannungsniveau:

Oben wurde bereits erwähnt, dass RS-232 ist 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 symmetrische 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, für die Übertragung wird also ein symmetrisches differentielles System verwendet.

Auf diese Weise liefert RS-422/RS-485 all die Vorteile, die mit differentiellen Signalen in Verbindung gebracht werden können (Rauschunterdrückung, Verminderung elektromagnetischer Interferenzen, geringer Leistungsverbrauch, höhere Busgeschwindigkeiten...). Diese Vorteile resultieren daraus, dass es sehr unwahrscheinlich ist, dass ein Störsignal auf einer Leitung ein positives und auf der anderen ein negatives Störmuster hinterlässt.

Einen Stecker-Standard wie bei RS-232 gibt es bei RS-422/RS-485 nicht, hier wird einfach mit einer 2-adrigen Leitung gearbeitet (Halbduplex) oder bei Vollduplex mit einer 4-adrigen Leitung.

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 ... und auch nur 50% an Energie verbraucht.

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 beiden Oszilloskop-Darstellungen auf der folgenden Seite verdeutlichen dies.

Die Frequenz liegt bei rund 150 Herz; die gelbe Kurve zeigt das PWM Steuersignal des Mikrocontrollers für das Display (also Port B7), die orangene Kurve zeigt den tatsächlichen Stromverbrauch der Beleuchtung (die Stromsonde rauscht, daher das verrauschte Signal).



Das obere Bild zeigt eine PWM von 90%, d.h. die Beleuchtung ist in 90% der Zeit eingeschaltet. In 90% der Zeit benötigt das Display rund 80mA, in 10% keinen Strom.

Das untere Bild zeigt eine PWM von 10%. Die Beleuchtung ist also nur 10% der Zeit aktiv und leuchtet entsprechend dunkel – braucht aber auch nur 1/10 der Energie.



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 und den Stromverbrauch anpassen.

Für das hier verbaute Display gilt in etwa:

Der Stromverbrauch bei 100% Beleuchtung beträgt ca. 85mA. Durch eine PWM mit 10% können Sie den Stromverbrauch der Beleuchtung auf 1/10, also rund 9mA reduzieren. Anmerkung: Diese 10% entsprechen in etwa der Helligkeit des alten Displays, welches bis Mitte 2017 verbaut wurde.

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 Timer I = Pwm , Pwm = 8 , Compare C Pwm = Clear Up , Compare C Pwm =
Clear Down , Prescale = 256

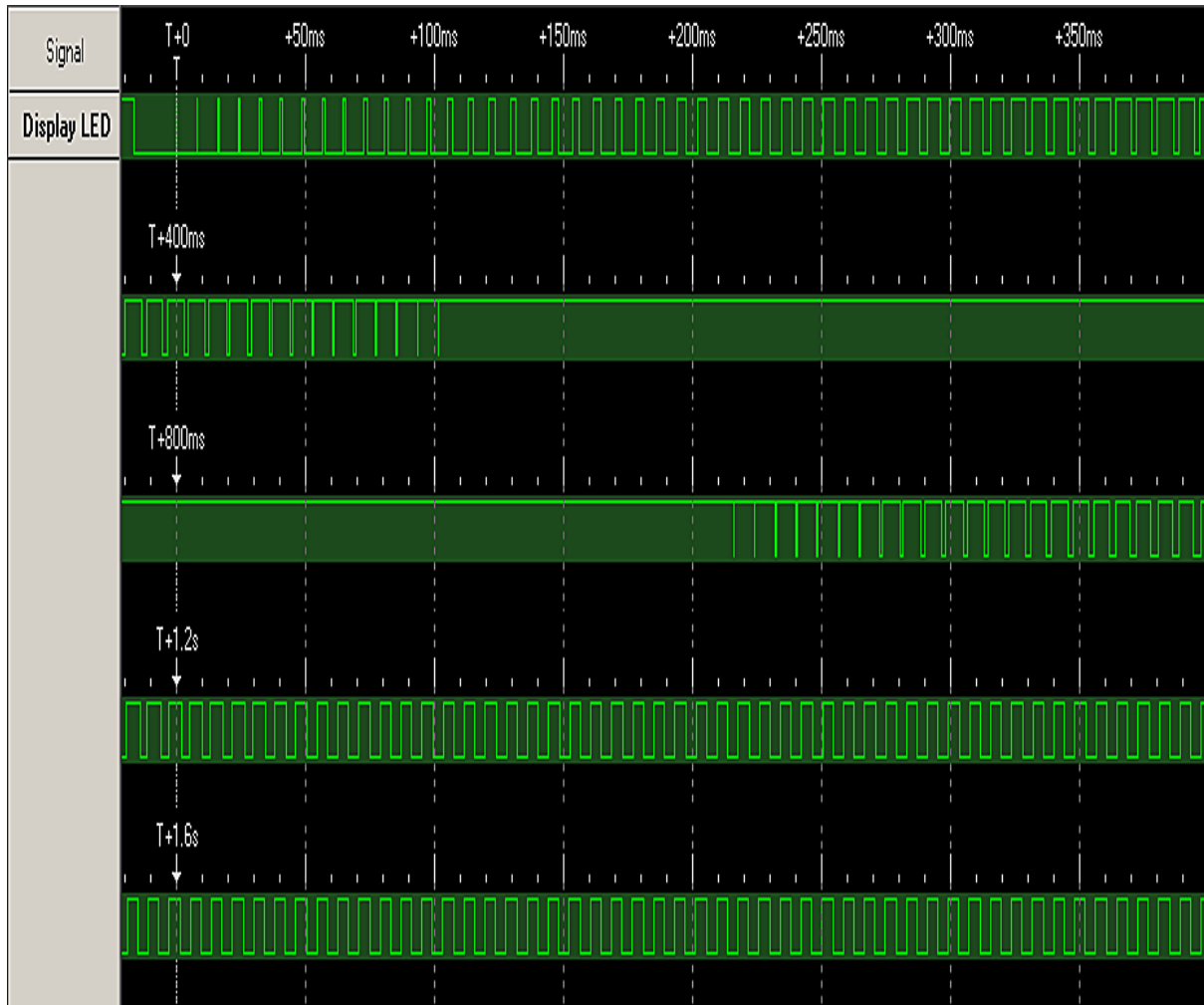
For I = 0 To 255 Step 5
  PwmIc = I
  Waitms 10
Next I

Waitms 500

For I = 255 To 130 Step -5
  PwmIc = I
  Waitms 10
```

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

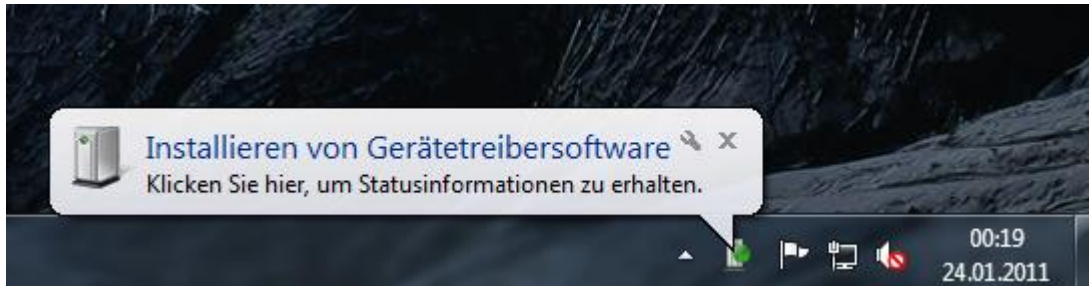
Zur Verdeutlichung haben wir zeigen wir das Ausgangssignal des Mikrocontrollers noch mit einem Logikanalyzer aufgezeichnet. Die gesamte Aufzeichnung ist 2 Sekunden lang, in jeder Zeile sind 400ms abgebildet. Sehr schön ist zu erkennen, wie das Tastverhältnis von Hell zu Dunkel sich mit fortlaufender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



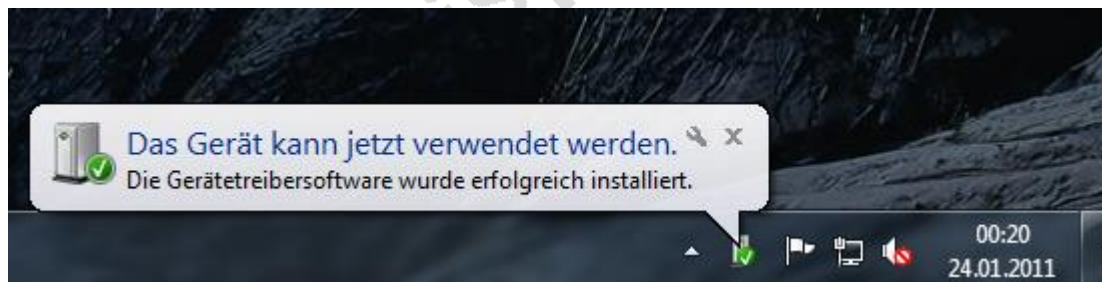
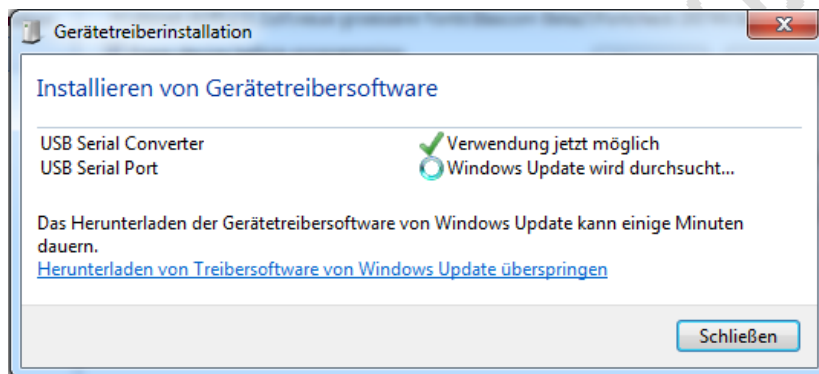
Die ersten 510 ms werden benötigt, um die Helligkeit in 51 Schritten von 0 auf 100% zu steigern – nach jedem Schritt folgt eine Wartezeit von 10 ms. Dann folgt eine Pause von 500ms (Waitms 500), in der das Display mit voller Stärke leuchtet. Dann wird das Display in 25 Schritten auf eine Leuchtstärke 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.

USB Treiber

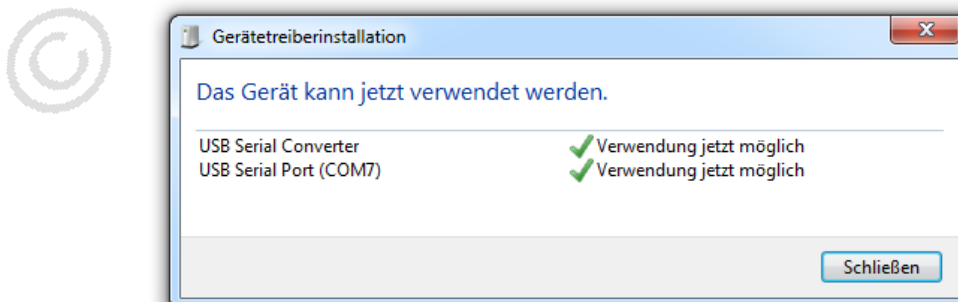
Wenn Sie das D074 Modul zusammen mit der USB Option bestellt haben, dann wird in der Regel beim erstmaligen Einstecken des Boards in den Windows-Rechner der notwendige Treiber für den **USB Chip FTDI FT230/231** installiert (Bilder zeigen ein Windows 7-System):



etwas warten (beim Anklicken des Icons erscheint zudem folgende Dialogbox)

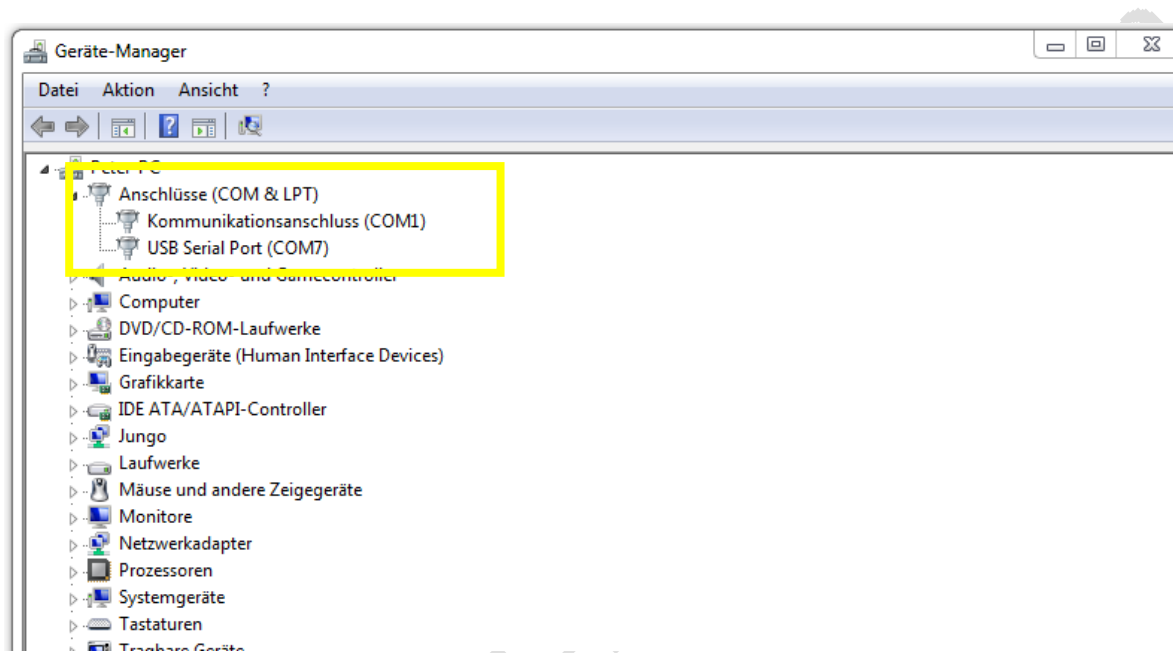


Evtl. erscheint auch folgende Box:



Nun stellt Ihnen das Betriebssystem eine virtuelle COM-Schnittstelle zur Verfügung. Sollte dies nicht funktionieren oder Sie ein anderes Betriebssystem (Linux, Mac) nutzen, so erhalten Sie die verfügbaren Treiber für einen virtuellen Com Port (VCP) auf der Seite des Chip-Herstellers: <http://www.ftdichip.com/Drivers/VCP.htm>

Unter Windows (*Start-Systemsteuerung-System-Sicherheit-Geräte-Manager*) sieht dies dann wie im nachfolgenden Bild aus (Voraussetzung: D074+ ist angesteckt und der Treiber wurde installiert). Der gewählte COM-Port kann übrigens frei gewählt werden (siehe nächste Seite).



Welcher COM-Port nun belegt wurde, können Sie im Geräte-Manager feststellen. Dort können Sie die Baudrate etc. festlegen. Die übliche Einstellung *9600 Baud, 8 Bit, 1 Stoppbit, keine Parität* sollten Sie bis zum ersten gelungenen Test erst einmal nicht verändern.

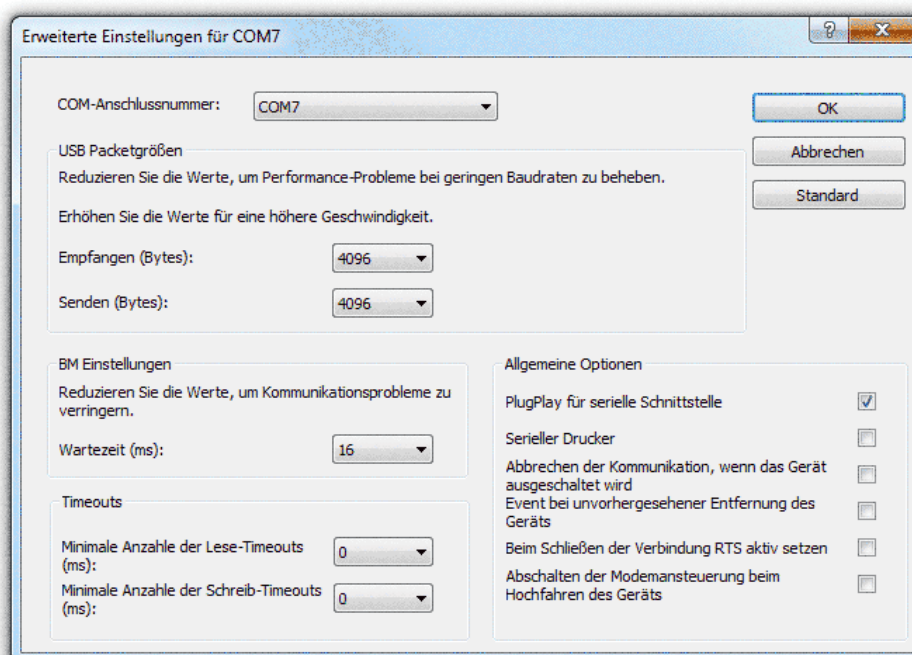
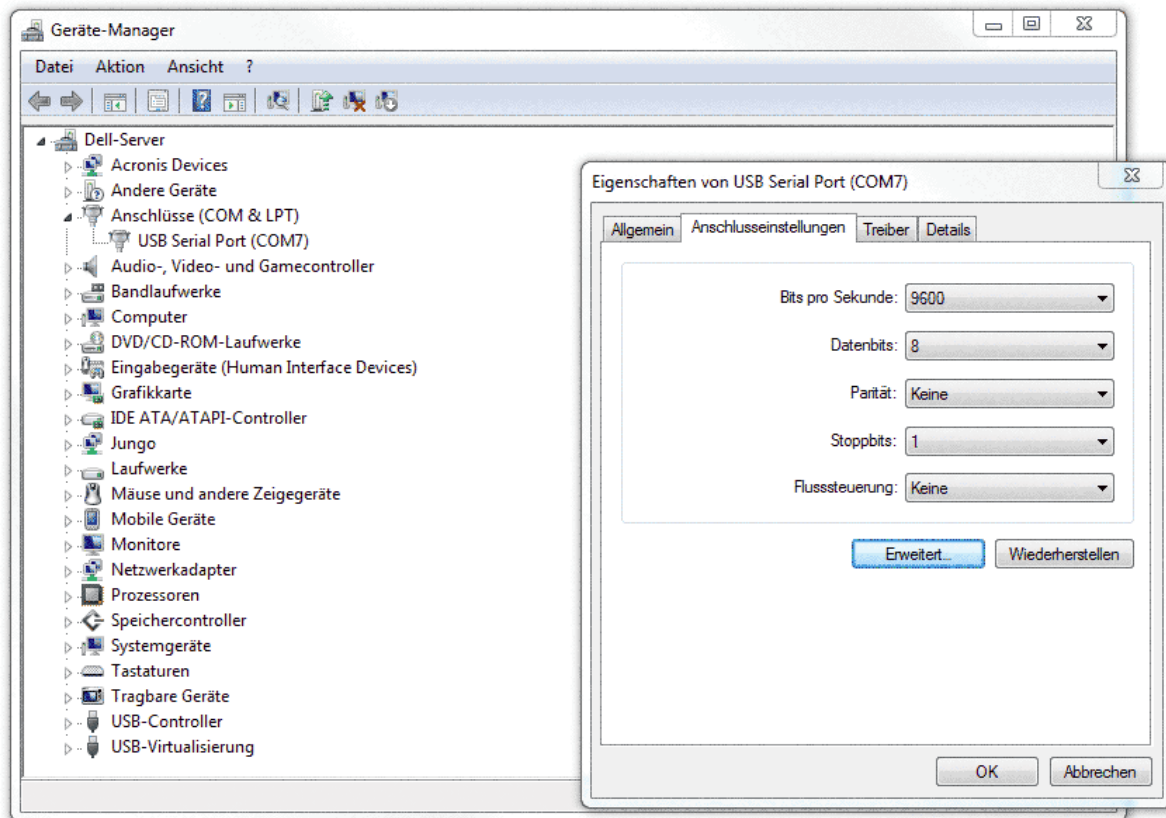
Anmerkung: Windows wählt hier automatisch erst einmal einen COM-Port vor (hier im Beispiel COM7). Dies kann bei jedem Rechner, abhängig von der Rechnerkonfiguration eine andere COM-Schnittstelle sein. Genauso kann es passieren, dass nach dem Anstecken an einen anderen, danebenliegenden USB Port von Windows automatisch eine andere COM-Schnittstelle gewählt wird. **Im Klartext: Sie können/sollten sich niemals sicher sein, dass eine bestimmte COM-Schnittstelle gewählt wird.**

Tipp: Wenn Sie eine eigene Software zur Kommunikation nutzen, so ist es empfehlenswert, dass die Software beim Programmstart alle verfügbaren COM-Ports der Reihe nach abscaant und auf ein Feedback vom D074+ wartet um dann daraufhin die COM-Schnittstelle festzulegen. Dies reduziert Fehlerquellen, Rückfragen und Anwender-Eingriffe.

Die folgende Seite illustriert, wie man die COM Schnittstelle und deren Parameter ändern kann.

Im Gerätemanager nach Doppelklick auf den Eintrag *USB Serial Port (COMx)* und unter dem Button „Erweitert“ legen Sie den gewünschten COM-Port fest. Das System merkt sich i.d.R. den genutzten USB Port, d.h. beim Wiederanstecken an den gleichen USB Port wird die gleiche virtuelle COM-Schnittstelle vergeben.

Wichtig: Das Anstecken des Moduls an einem anderen (z.B. danebenliegendem) USB Port resultiert unter Umständen in einer anderen COM-Schnittstelle (z.B. COM8).



Nutzung des SD Kartenmoduls

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

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. 1x 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 werden in Kürze ein vorhandenes Beispiel an unser Board anpassen und Ihnen dann auf Anfrage zur Verfügung stellen. 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 Ansprechend 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.chipcatalog.com/Doc/BC6DC20FC60BAC7595E5D0419DB6126E.pdf>

AVR Bascom:

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

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

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

Links zu C-Code für SD Karten:

http://elm-chan.org/docs/mmc/mmc_e.html

http://www.ulrichradig.de/site/atmel/avr_mmc/d/pdf/hitachi_hb28b128mm2.pdf

http://www.embedded-os.de/index.html?pcfat_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>

SD-Karte – Warum aufwändige Beschaltung?

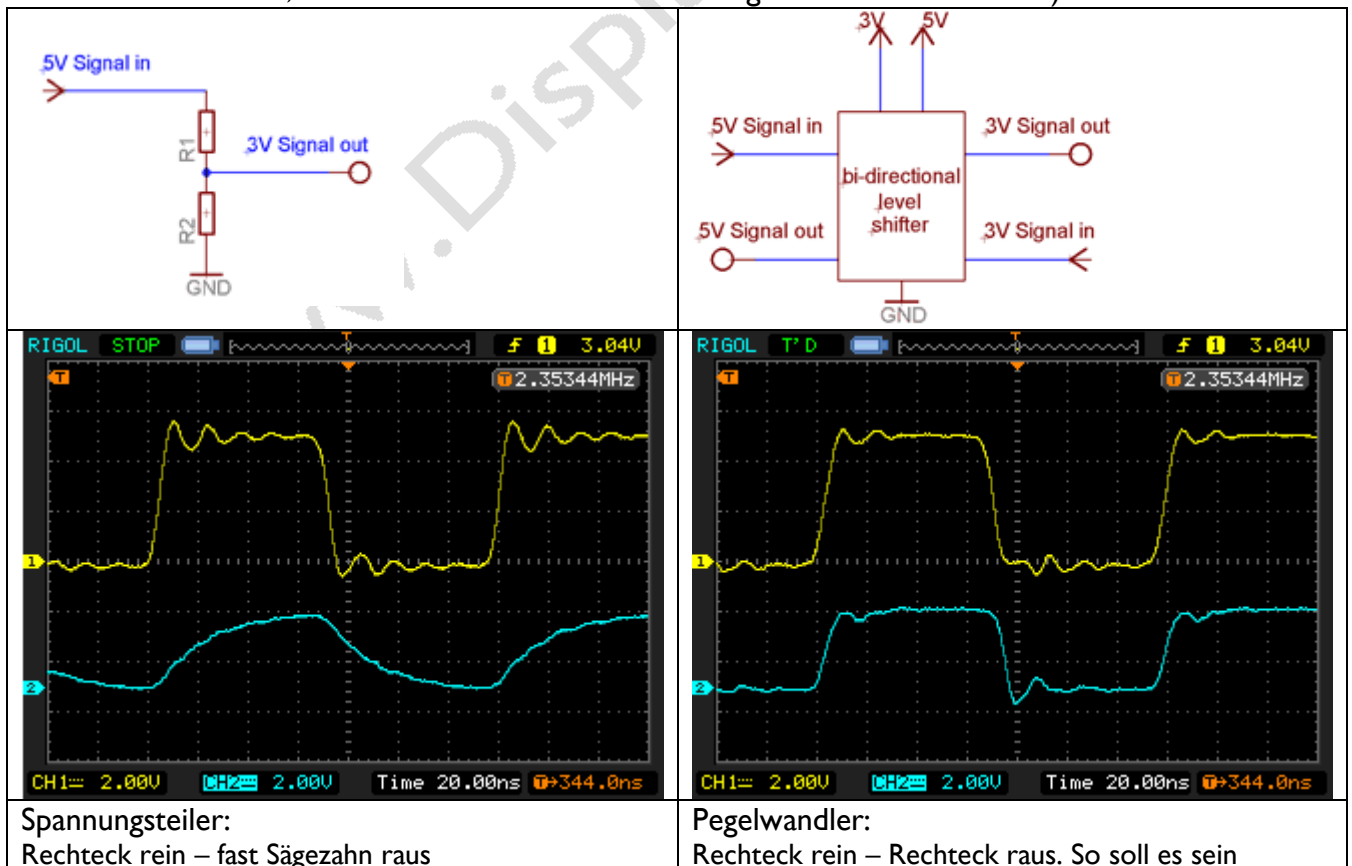
Eine SD-Karte benötigt eine Spannung von 2,7 bis 3,6 Volt – aber je nach Nutzung schwankt der Strombedarf der Karte von wenigen mA bis zu (kurzfristig) 100mA. Damit scheidet ein Spannungsteiler aus Widerständen schon aus, denn hier muss die Last schon gleich bleibend sein. Alternativ bieten sich hier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

Pegelwandlung

Die direkte Verbindung zwischen einer 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).

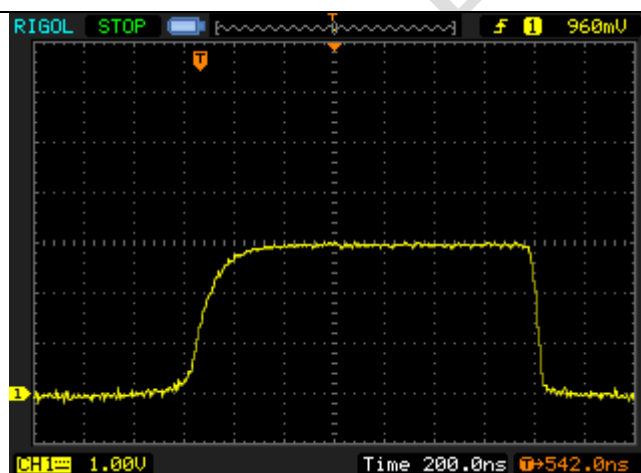
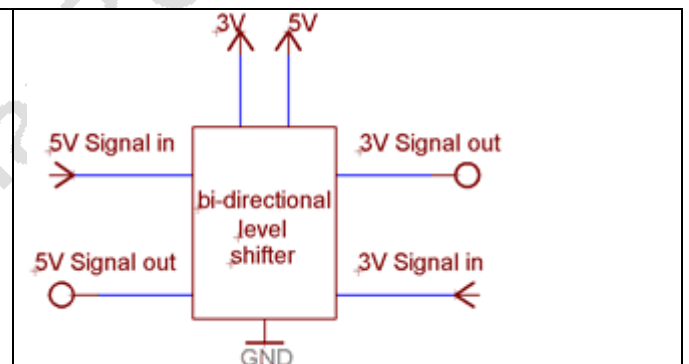
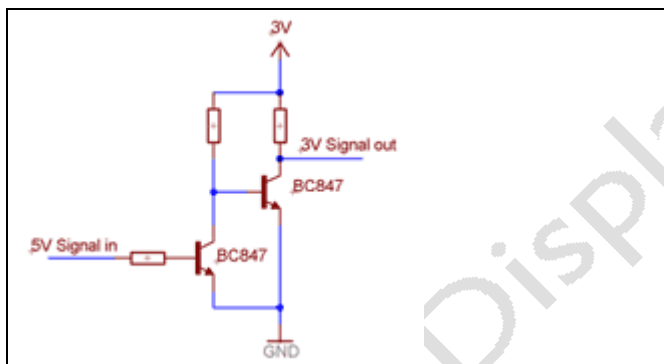
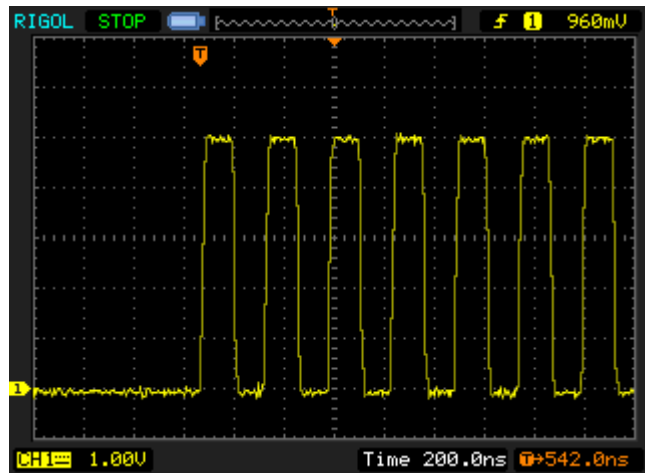


Pegelwandlung durch Transistoren

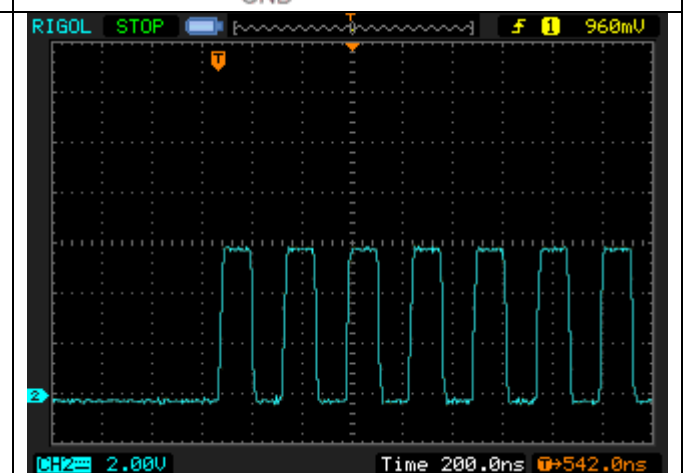
Jetzt wenden Sie vielleicht ein, dass man ja auch mittels zweier Transistoren den Pegel von 5 Volt auf 3 Volt senken kann. Das stimmt wohl, aber sie werden Schwierigkeiten haben, mit Transistoren einen Pegelwandler aufzubauen, der einen schnellen Datenverkehr zulässt. Die im Internet kursierende Schaltung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



Ausgangssignal der Transistorstufe



Ausgangssignal des spezialisierten Pegelwandlers

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 Busteilnehmer 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.

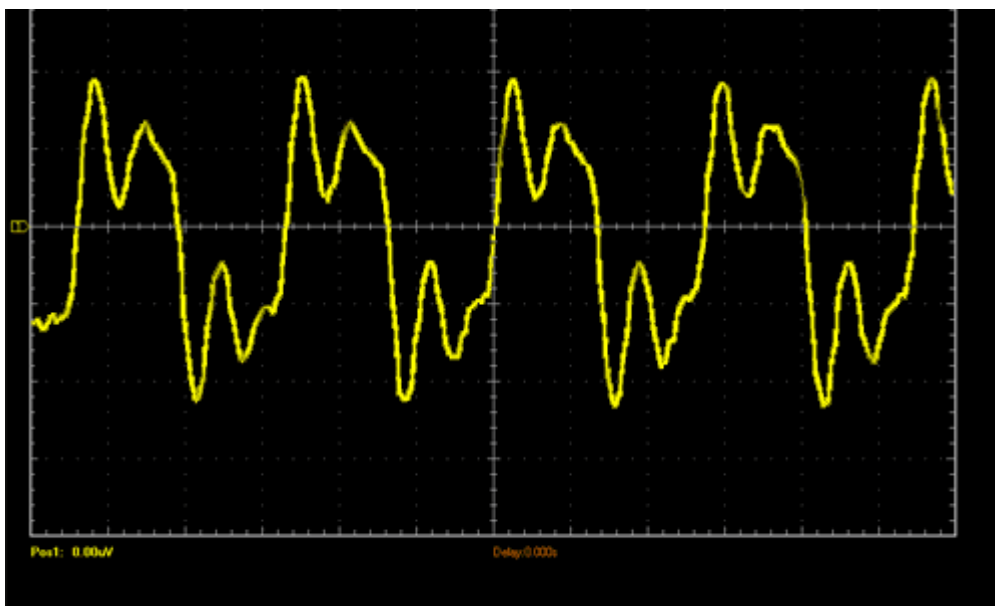
Tristate-Ausgänge – Wichtig !

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.

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).

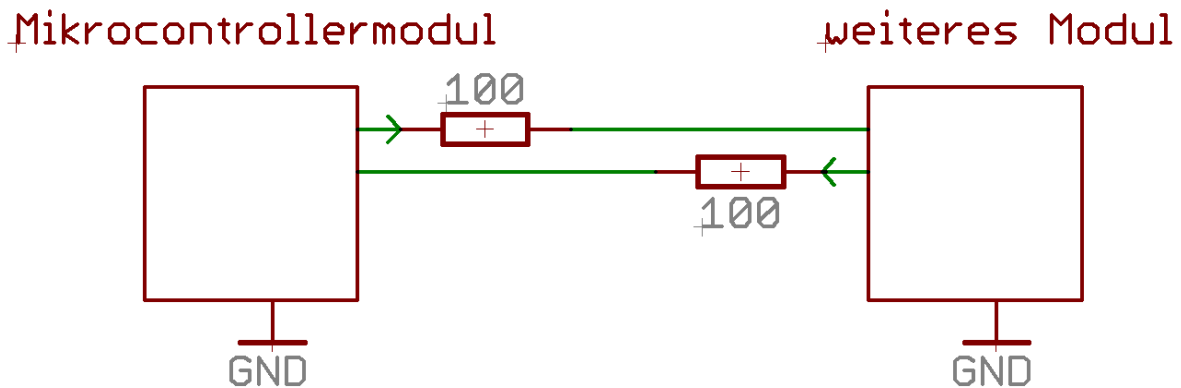


Statt dem einstmals sauberen Rechtecksignal wird dem Empfänger nun ein Signal überreicht, welches – nunja ... suboptimal ist. Ob hier noch eine korrekte Datenauswertung stattfinden kann ist fraglich.

Abhilfe schaffen hier zwei Optionen

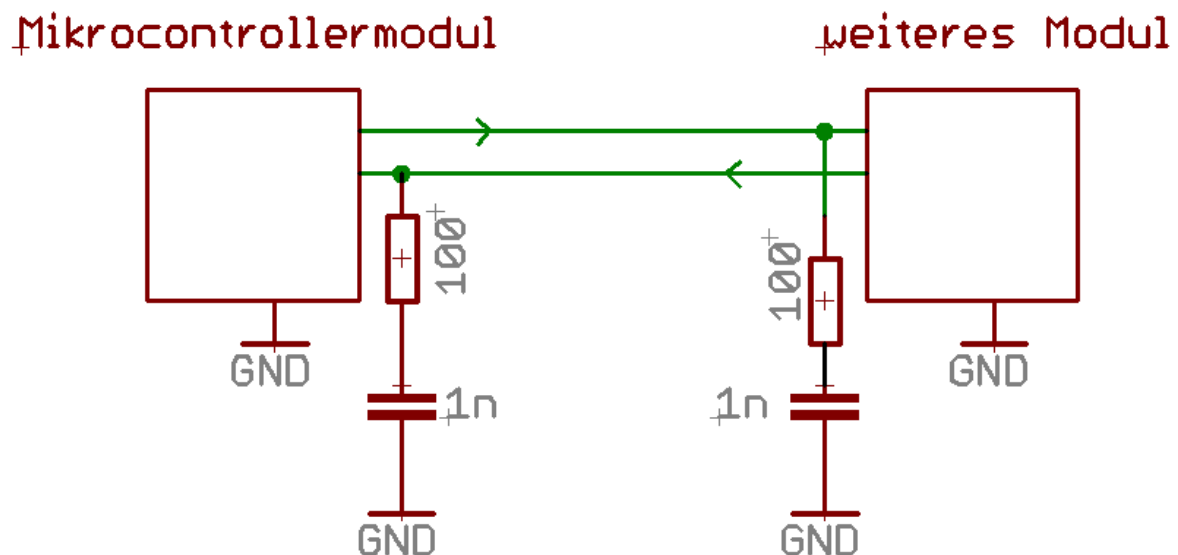
Option 1)

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.

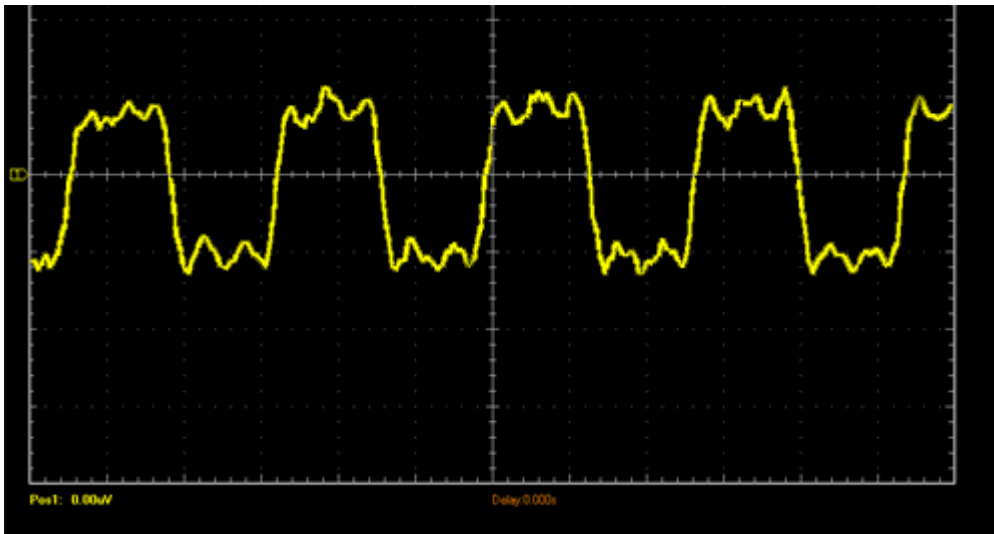


Option 2)

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.



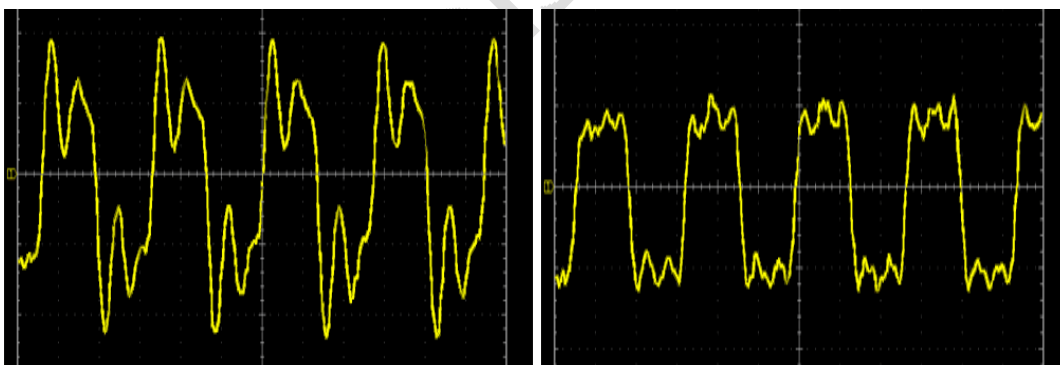
Unten erkennt man, wie gut das exakt gleiche Signal (also ebenfalls nach 50 cm Flachbandkabel) aussehen kann, wenn es mit einer Leitungsterminierung am Empfänger versehen wird. Zwar gibt es noch ein paar Einschwinger, aber die Einsen und Nullen sind klar zu erkennen und auszuwerten.



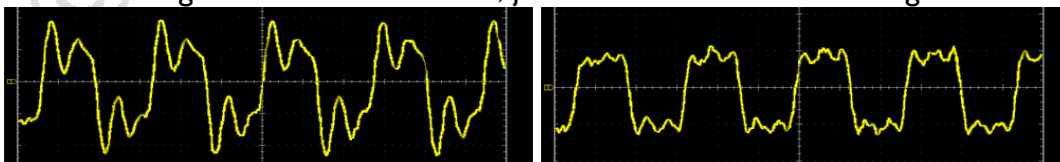
Die genauen Daten müssen je nach Umgebung experimentell ermittelt werden. Als Anhaltspunkt sind jedoch Werte von 1nF bzw. 100 Ohm gut geeignet.

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:



Also: Geräte, die mit längeren Leitungen (>50cm) angeschlossen werden müssen u.U. auf **Empfängerseite** terminiert werden.

Standardgehäuse GBS400

Das Modul D074 wurde auf das angebotene Standardgehäuse BS400 / BS401 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.

Gehäusevarianten

Das nachfolgende Foto zeigt die beiden nutzbaren Varianten des Gehäuses. Sie können die beiden Gehäuseschalen des BS400 so zusammenfügen, dass sie ein pultförmiges Gehäuse erhalten (oben) oder alternativ ein flaches, an beiden Enden gleich hohes Gehäuse (unten).



Der Platz im Gehäuse ist vor allem bei der Wahl inkl. Batteriefach relativ begrenzt. Zusammen mit einer über die ganze Länge verbauten Platine ergeben sich u.U. Platzprobleme, so dass wir zumindest für die Erstbestellung eine Bestellung inkl. der Dekordichtung empfehlen, die die nutzbare Höhe im Innern um einige mm vergrößert.

Gehäuse ohne Batteriefach

Zusammenbau als **Pultgehäuse** (=schräg; unten flach, oben höher):

Nur mit Dekordichtung möglich

Zusammenbau als **Flachgehäuse** (oben und unten gleich hoch):

Auch ohne Dekordichtung möglich

Gehäuse mit Batteriefach

Zusammenbau als **Pultgehäuse** (=schräg; unten flach, oben höher):

Nur mit Dekordichtung möglich

Zusammenbau als **Flachgehäuse** (oben und unten gleich hoch):

Nur mit Dekordichtung möglich

Der Einbau der D074 Platine inkl. Stecker / Pins

Gelegentlich möchte man aber eventuell ein D074 Board inklusive Port-Stecker im Gehäuse verbauen. Das ist jedoch aufgrund des Platzbedarfs der Pins meist nicht möglich.

Prüfen Sie bitte genau, ob die gewünschte Konfiguration passt, bevor Sie anfangen zu Löten.

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.

BS400 = ohne Batteriefach

BS401 = mit Batteriefach

Mit neuem Modul nochmal testen und updaten

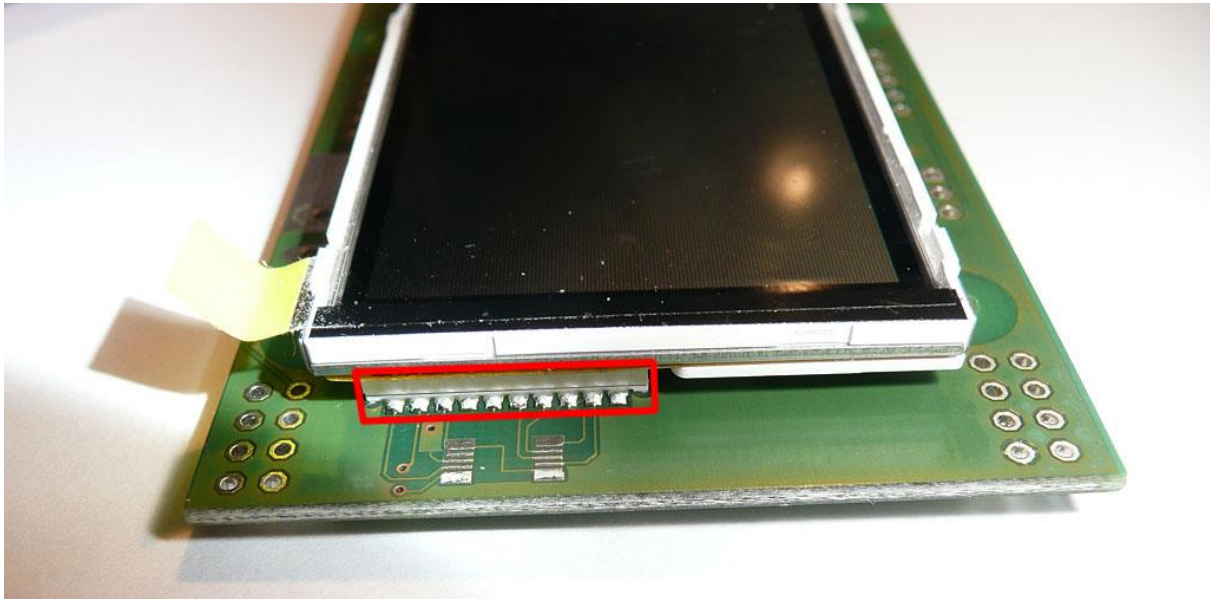
Zusammenbau als Pultgehäuse (=schräg; unten flach, oben höher)				
Bauteile	Mit Dekordichtung		Ohne Dekordichtung	
	BS400	BS401	BS400	BS401
D074 Board	Ja	Ja	Ja	Nein
ISP Pins	Ja	Ja	Ja	-
Port-Pins A,B,E,F,...	Ja	Zum Teil	Ja	-
Port-Pins C,D,...	Ja	Nein	Ja	-
SD Karte	Ja	Ja	Ja	-
USB Stecker	Ja	Ja	Ja	-
5V Spannungsregler	Ja	Nein	Ja	-
Echtzeituhr+Batt	Ja	Ja	Ja	-

Zusammenbau als Flachgehäuse (=beide Seiten gleich hoch)				
Bauteile	Mit Dekordichtung		Ohne Dekordichtung	
	BS400	BS401	BS400	BS401
D074 Board	Ja	Ja	Ja	Nein
ISP Pins	Ja	Ja	Ja	-
Port-Pins A,B,E,F,...	Ja	Zum Teil	Ja	-
Port-Pins C,D,...	Ja	Nein	Ja	-
SD Karte	Ja	Ja	Ja	-
USB Stecker	Ja	Ja	Ja	-
5V Spannungsregler	Ja	Nein	Ja	-
Echtzeituhr+Batt	Ja	Ja	Ja	-

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 Connectorhälfte 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.

Neues Foto einfügen

Das Schließen des Displaysteckers

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.

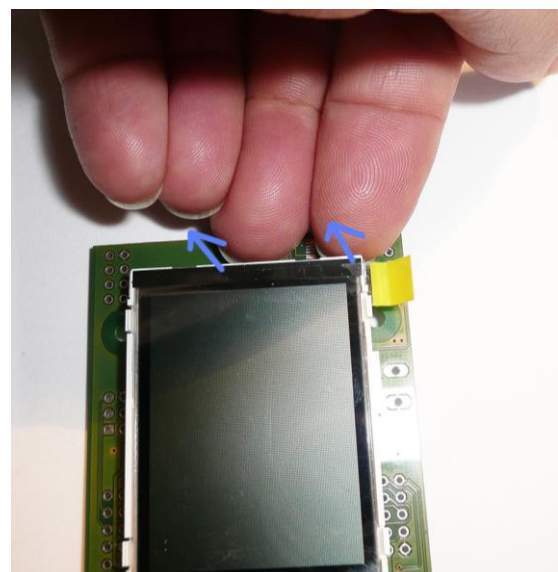
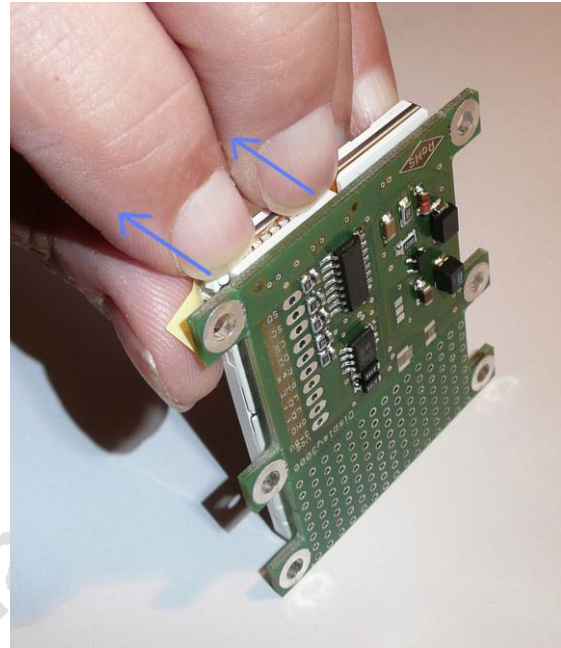
Zum Entfernen des Displays nehmen Sie Zeigefinger und Mittelfinger der rechten Hand und schieben die Spitzen der Fingernägel der beiden Finger bis an den Kunststoffkörper des Steckers; die Fingernägel **müssen** den Kunststoff berühren (siehe nebenstehendes Foto).

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.

Alternativ zu den Fingernägeln hat sich übrigens eine SD Karte als ideal erwiesen. Sie hat die richtige Dicke: einfach in den Spalt stecken und den Stecker dann mit der SD Karte aufhebeln.



Versetzen des Displays / Verlängerung Displayleitung

Das D074x Board ist darauf vorbereitet, dass auch von der Platine entfernt zu betreiben. Sofern die baulichen Gegebenheiten eine Versetzung des Displays notwendig macht, kontaktieren Sie uns bitte bzgl. einer kleinen Adapterplatine.

Die D074x-Platine zeigt am oberen Rand eine Reihe mit 10 Löt pads, an welchen die Display-signale anliegen. Die Kontaktierung geschieht 1:1 zu unserer Adapterplatine. Das Rastermaß beträgt 2,00 mm – für dieses Rastermaß gibt es auf dem Markt auch Stecker/Buchsen, ansonsten lassen sich die 10 notwendigen Leitungen auch schnell mit üblichem Flachbandkabel verbinden.

Legen Sie die Leitungen genauso, wie auf der nachfolgenden Zeichnung zu erkennen. Ein falscher Anschluss kann zur Zerstörung des Displays führen.

Wichtige Hinweise:

Die Display-signale sind relativ schnell und steilflankig. Zusätzliche Leitungen werden u.U. zu Störungen beim Displaybetrieb führen (Reflexionen, EMV Einflüsse). In diesem Fall müssen die Leitungen u.U. noch am Display passiv terminiert werden. Versuchen Sie die maximale Kabellänge auf unter 20cm (besser 10cm) zu halten.

Aufgrund der schnellen und steilflankigen Signale ist das Modul nach dem Versetzen des Displays zudem u.U. nicht mehr CE konform. Hier können evtl. in die Datenleitungen eingeschleifte Serienwiderstände (z.B. 100 Ohm) zwischen Platine und Verlängerungsleitung helfen. Evtl. Zusatzmessungen müssen durch Sie durchgeführt werden.

Errata:

~~Noch nichts bekannt ☹~~

Leider doch:

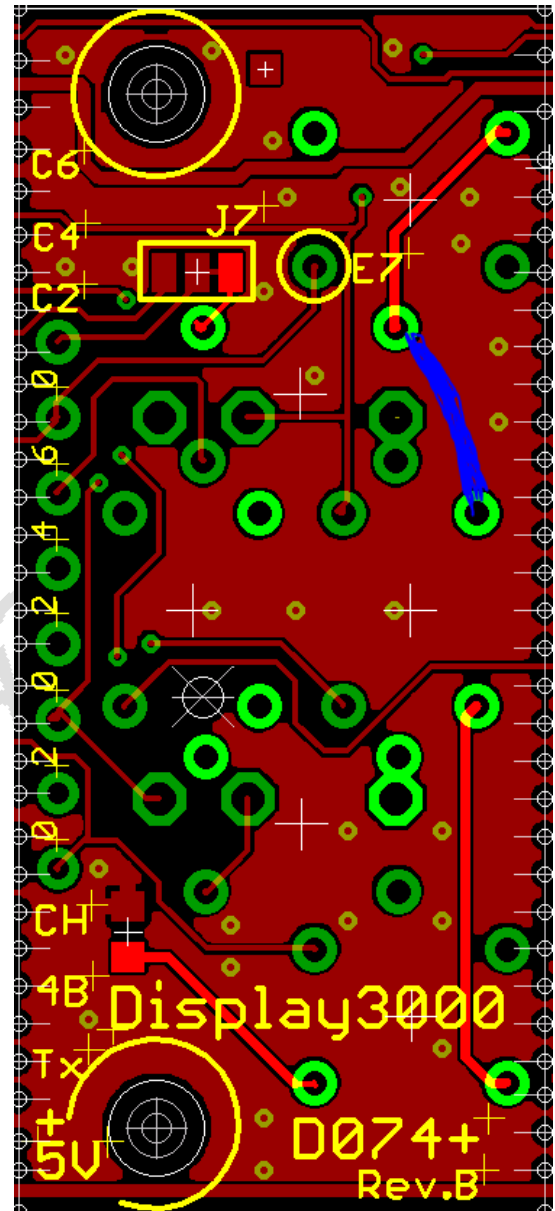
D074s+ Benutzung der Taster/Folientastatur

Ein D074s+ der Rev.B braucht für die Nutzung der Taster oder der Folientastatur noch eine manuell zu legende Leitung. Diese wird beim Abtrennen des unteren Bereichs eines D074x+ aufgetrennt und muss manuell gelegt werden.

Bei einem D074x+ ist alles in Ordnung, hier muss keine Leitung gelegt werden.

Das nachfolgende Bild vom Layout zeigt die betreffenden Pads in Gelb (also leuchtend). Sie müssen von irgendeinem der oberen 4 Pads zu irgendeinem der 9 unteren Pads eine Verbindung legen. Als Beispiel haben wir eine der möglichen Verbindungen blau markiert.

Ab Rev.C wird der Fehler behoben sein.



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.