

AVR-8-bit-Mikrocontroller
Gruppe 200 - Einsetzen von AVR-Tools
Teil 203 - AVR-ALE-Testboard



Teil 201 - Experimentierboards

- 1 Experimentierboards zum Testen und Programmieren von AVR-Mikrocontroller
 - 1.1 Mit welchen Mitteln AVR-Mikrocontroller programmiert werden
 - 1.2 Starterkit STK500
 - 1.3 Entwicklungs-Tool AVR Dragon
 - 1.4 ATM18-Controllermodul und ATM18-Testboard
 - 1.5 AVR-ALE-Testboard

Teil 202 - ISP-Programmieradapter

- 2 ISP-Programmieradapter
 - 2.1 ISP Bezogen auf die verschiedenen Schnittstellen
 - 2.1.1 Serielle Schnittstelle
 - 2.1.2 Parallele Schnittstelle
 - 2.1.3 USB-Schnittstelle
 - 2.2 CC2-AVR-Programmer alias USBprog
 - 2.2.1 Aufbau
 - 2.2.2 Arbeitsweise
 - 2.2.3 Firmware-Änderung

Teil 203 - AVR-ALE-Testboard

- 3 Beschreibung des AVR-ALE-Testboard
 - 3.1 Schaltungsaufbau
 - 3.2 Stromversorgung
 - 3.3 Einsatz verschiedener AVR-Mikrocontroller
 - 3.4 Anzahl LEDs und Tasten
 - 3.5 LCD-Interface und 20x4-LCD
 - 3.5.1 Erzeugung des Enable-Signals für das LCD
 - 3.5.2 LCD-Backlight
 - 3.6 Ansteuerung von Relais
 - 3.7 RS-232-Schnittstelle
 - 3.8 USART-Testboard-Schnittstelle

Teil 204 - AVR Studio

- 4 Einsatz des AVR Studio
 - 4.1 AVR Studio installieren
 - 4.2 Testboard und Programmer zusammenschalten
 - 4.2.1 Treiber AVRISP mkII neu installieren
 - 4.3 Starten von AVR Studio
 - 4.3.1 AVR Studio und CC2-AVR-Programmer
 - 4.3.2 Mikrocontroller-Einstellungen im AVR Studio

Teil 205 - Assembler und AVR Studio

- 5 Assembler und AVR Studio
 - 5.1 Der Übersetzer (Assembler)
 - 5.2 Ein neues Projekt erzeugen
 - 5.2.1 Der Projekt-Bereich
 - 5.2.2 Bearbeiten der Assemblerdatei
 - 5.2.3 Assemblieren des Quell-Codes
 - 5.3 Simulation des Codes
 - 5.3.1 Programmausführung im Einzelschrittverfahren
 - 5.3.2 Debugger-Stopp-Punkte
 - 5.4 Verändern des Programmtextes

AVR-8-bit-Mikrocontroller

Gruppe 200 - Einsetzen von AVR-Tools

Teil 203 - AVR-ALE-Testboard

- 5.4.1 Überwachen von Variablen
- 5.4.2 Anzeigen der Prozessordetails
- 5.4.3 Speichern des Projekts
- 5.5 Erzeugen eines weiteren ASM-Projektes im Schnelldurchgang
- 5.6 Flashen eines ASM-Programms in ein Mikrocontroller ATmega88

Teil 206 - C-Compiler und AVR Studio

- 6 CodeVisionAVR C-Compiler und AVR Studio
 - 6.1 CodeVisionAVR C-Compiler installieren
 - 6.2 Erzeugen eines C-Projektes
 - 6.2.1 Ein neues Projekt beginnen
 - 6.2.2 Ein C-Projekt generieren
 - 6.3 Einbinden von AVR Studio in den CVAVR
 - 6.4 AVR Studio Debugger für CVAVR
 - 6.5 Flashen eines C-Programms in ein Mikrocontroller ATmega88

Teil 207 - Editor - UltraEdit

- 7 Editor - UltraEdit
 - 7.1 Kopf- und Fuss-Zeile
 - 7.1.1 Einstellungen für Assembler-Programme
 - 7.1.2 Einstellungen für C-Compiler-Programme
 - 7.2. Einträge in das WORDFILE.TXT
 - 7.2.1 Grundsätzliches Format
 - 7.2.2 Syntax Befehle
 - 7.3 Wortsammlung für AVR-Assembler
 - 7.4 Wortsammlung für CodeVisionAVR C-Compiler

Hinweis

Externe Anschaltungen und Hardware-Erweiterungen werden in der **Gruppe 400 - ASM-Projekte** und in der **Gruppe 600 - C-Projekte** detailliert beschrieben.

AVR-8-bit-Mikrocontroller

Gruppe 200 - Einsetzen von AVR-Tools

Teil 203 - AVR-ALE-Testboard

Vorbemerkung

Nichts ist vollkommen - und nichts ist endgültig! So auch nicht dieses Tutorial! Deshalb bitte immer erst nach dem neuesten Datum schauen. Vielleicht gibt es wieder etwas Neues oder eine Fehlerbereinigung oder eine etwas bessere Erklärung. Wer Fehler findet oder Verbesserungen vorzuschlagen hat, bitte melden (info@alenck.de).

Immer nach dem Motto: Das Bessere ist Feind des Guten und nichts ist so gut, dass es nicht noch verbessert werden könnte.

Bild-, Beispiel-, Form- und Tabellen-Nummern sind nach folgendem Schema aufgebaut, damit bei Einfügungen/Löschungen nicht alle Nummern wieder geändert werden müssen (hier bunt dargestellt):

Darstellungsart	Abschnitt-LfdNummer: Beschreibung	allgemeines Schema
•	Bild 5.1.4-02: Daten-Adress-Raum	Benummerung eines Bildes
•	Beispiel 5.1.4-03: EEPROM-Speicherung	Benummerung eines Beispiels
•	Form 5.1.3-01: Die main-Funktion	Benummerung einer Formdarstellung
•	Tabelle 5.1.4-01: Schlüsselwörter vom CAVR	Benummerung einer Tabelle

Gravierende Änderungen gegenüber der Vorversion

1.

Völlig neue Strukturierung in **Gruppen** und **Teile**, um das Tutorial umfassend ordnen zu können. Die **Abschnitte** in den **Teilen** sind weitgehend erhalten geblieben.

Gruppenbezeichnung	Kurzbezeichnung
Gruppe 100: Technologie der AVR-8-Bit-Mikrocontroller	Technologie
Gruppe 200: Einsetzen von AVR-Tools	Tools
Gruppe 300: Arbeiten mit AVR-Assembler 3xx_Programm_yyyy	ASM-Programmierung ASM-Programm-Beispiel
Gruppe 400: AVR-ASM-Projekte 4xx_Projekt_yyyy	ASM-Projekte ASM-Projekt-Bezeichnung
Gruppe 500: CodeVisionAVR C-Compiler 5xx_Programm_yyyy	C-Programmierung C-Programm-Beispiel
Gruppe 600: AVR-C-Projekte 6xx_Projekt_yyyy	C-Projekte C-Projekt-Bezeichnung

xx steht für die laufende Nummer innerhalb des **Teils**, in dem das Programm bzw. das Projekt erscheint und **yyyy** steht für die Programm- bzw. Projekt-Kurz-Bezeichnung.

2.

Notwendige Änderungen auf Grund Neuinstallation von **Windows 7**.

3.

Windows 7 machte eine Installation von **CodeVisionAVR V2.60** als Vollversion notwendig. Daraus leiten sich auch viele Änderungen im Detail für die C-Programmierung (**Gruppe 500**) ab.

4.

Neu-Installation von **AVR Studio Vers. 4.19** unter **Windows 7**

5.

Zur Demonstration des Tools **AVR Studio** ist in **Gruppe 200** eine Trennung in **Teil 205 - Assembler und AVR Studio** und **Teil 206 - C-Compiler und AVR Studio** vorgenommen worden.

6.

ASM- und **C-Projekte** werden jeweils in eigenen Gruppen gesammelt (**Gruppe 400** für Assembler- und **Gruppe 600** für C-Projekte).

AVR-8-bit-Mikrocontroller

Gruppe 200 - Einsetzen von AVR-Tools

Teil 203 - AVR-ALE-Testboard

3 Beschreibung des AVR-ALE-Testboard

Mit den Vorstellungen des Starterkits **STK500** der Firma **ATMEL** im Hinterkopf sollte vom Aufbau des ATM18-Projekt-Equipments aus Kompatibilitätsgründen nicht grundlegend abgewichen werden, so dass schließlich eine Adaption der ATM18-Platine in Frage kam: Umwandlung der **SMD**-Technik nach **THT** (Through Hole Technology). Die getestete Version kann mit vielen Einzel-Links von der Website www.alenck.de/AVR_Gruppe_200_Tools.html herunter geladen werden.

Nachdem die Entscheidung zugunsten der herkömmlichen Platinen-Technologie gefallen war, ergab sich die Größe fast von allein: Eine Europlatine - 4x so groß wie das Original. Es sollte nämlich ein **Null-Kraft-Sockel** für die zu testenden Mikrocontroller in Dual-In-Line-Ausführung (**DIL**) verwendet werden. Das **LCD-Interface** sollte ebenfalls noch Platz finden und darüber hinaus braucht man gelegentlich "echte Relais" zur galvanischen Trennung von Schaltkreisen. Damit beginnen aber schon die Ergänzungen.

Alle bisher verwendeten Chips sind zum Glück auch noch als Dual-In-Line zu bekommen, so dass keine Schwierigkeiten auftauchten. Auch für die übrigen Bauteile ergaben sich keine Adaptions-schwierigkeiten. Äquivalenztypen lassen sich fast immer finden.

Das Testboard stellt zahlreiche Schnittstellen zu Verfügung, die über Pfostenbuchsen und Verbindungsschnüre (flexible Steckbrücken) mit der Außenwelt in Verbindung treten. Sie dienen zum Anschalten von Hardware-Erweiterungen, Sensoren, Aktoren und sonstigen Zubehör.

Tipp für die Praxis: Bei den Pfostenbuchsen und den Verbindungsschnüren sollte man genau darauf achten, dass die Durchmesser genau miteinander harmonieren. Käufliche sog. flexible Steckbrücken (z.B. von CONRAD ELECTRONIC SE) sind für normale Buchsenleisten, die für Vierkantstifte der Stiftleisten konzipiert sind, zu dünn und oft auch mit fertigungsbedingten "Wackelkontakten" behaftet. Besser ist es, sich selbst flexible Steckbrücken aus Stiften und/oder Buchsen (0,635 mm im Quadrat oder 0,7 bis 0,9 mm Durchmesser) herzustellen.

3.1 Schaltungsaufbau

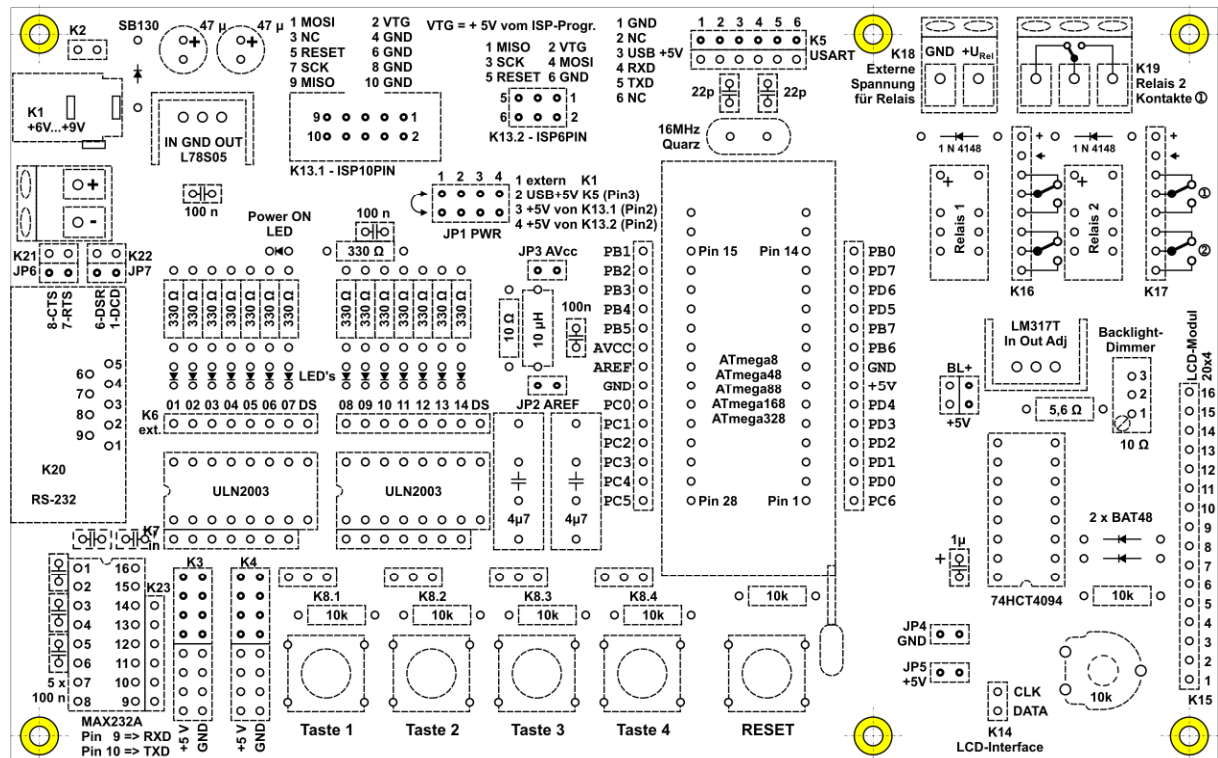


Bild 3.1-01 AVR-ALE-Testboard (Bestückung - [Bildvergrößerung](#))

AVR-8-bit-Mikrocontroller Gruppe 200 - Einsetzen von AVR-Tools Teil 203 - AVR-ALE-Testboard

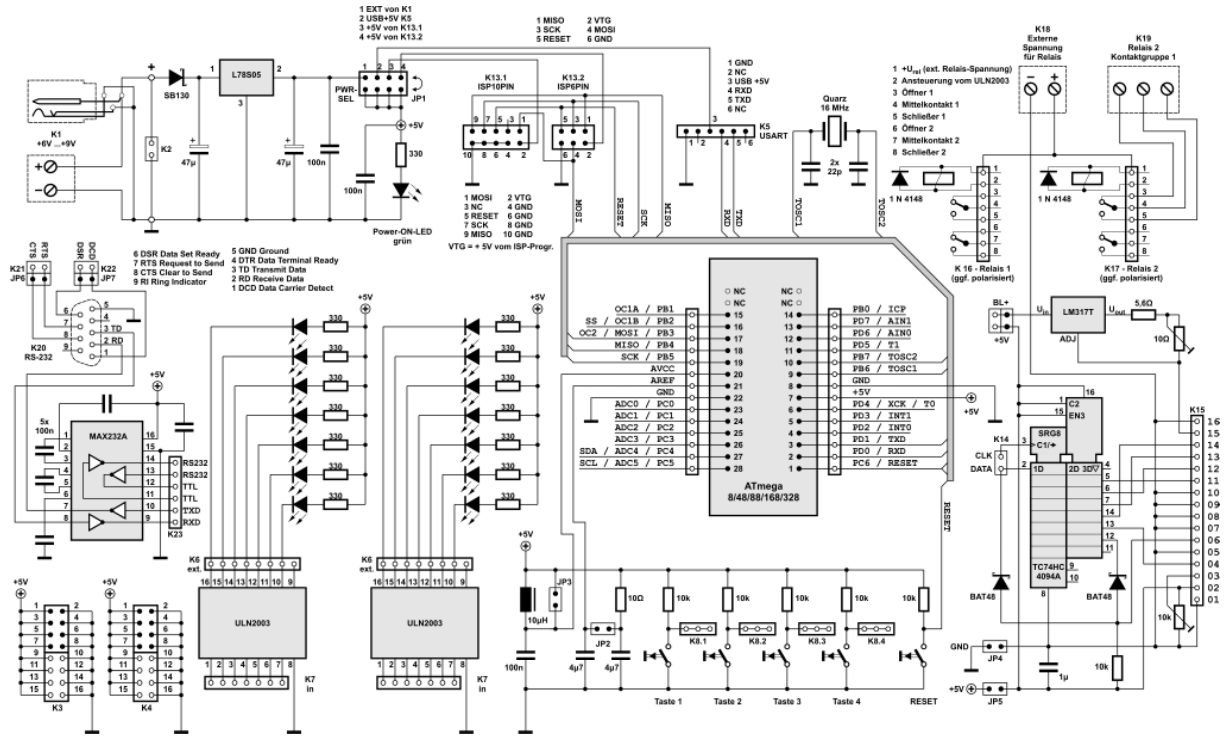


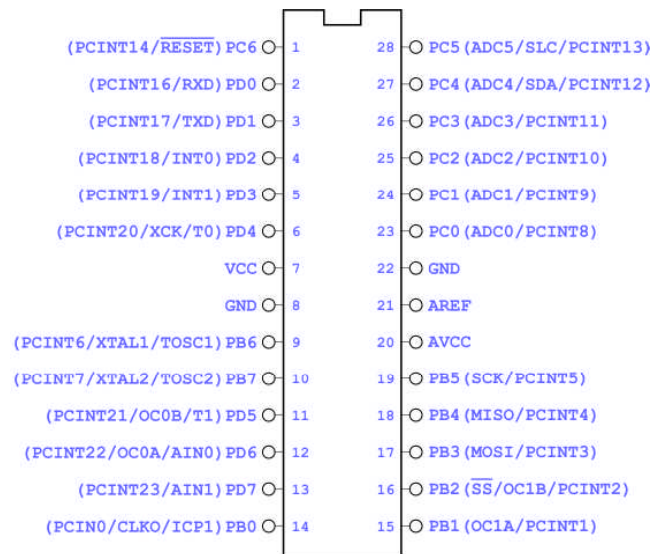
Bild 3.1-02: AVR-ALE-Testboard (Gesamtschaltbild - [Bildvergrößerung](#))

3.2 Die Stromversorgung

Für die Stromversorgung wurde in Erwartung größerer "Stromverbraucher" eine leistungsfähigere Version mit einem **L78S05** vorgesehen. Und um die Verlustleistung nicht zu hoch werden zu lassen, wurde ein Stecker-Netzteil mit einer Vorregelung auf 7,5 Volt ausgesucht.

Im Regelfall wird das Testboard mit dem geregelten Stecker-Netzteil von 7,5 V/2,4 A an **K1** betrieben. Auf der 8-poligen Stiftleiste Power-Selector (**PWR-SEL**) wird mit einem Jumper (**JP1**) die Stromversorgung des Testboards eingestellt. Bei der Stromversorgung über die Buchse **K1** wird der Jumper auf die Position 1 (**Ext**) gesetzt. Beim Setzen des Jumpers auf die Position 2 erhält das Board seine Spannung vom **K5 (USART Pin 3)** und beim Setzen des Jumpers auf die Position 3 oder 4 von den jeweiligen ISP-Schnittstellen **K13.1 (ISP10Pin, Pin 2)** bzw. **K13.2 (ISP6PIN, Pin 2)** sofern die entsprechenden Pins beschaltet sind und mit +5 Volt versorgt werden.

Wenn das Testboard mit +5 Volt versorgt ist, leuchtet die **Power ON LED** grün auf.



3.3 Einsatz verschiedener AVR-Mikrocontroller

Das Testboard ist für 8-bit-AVR-Mikrocontroller geeignet. Mit seinem Null-Kraft-Sockel ist es auf die folgenden AVR-Mikrocontroller in PDIP-Ausführung (DIL) beschränkt, die alle die gleiche Pin-Belegung aufweisen:

- ATmega8**
- ATmega48**
- ATmega88**
- ATmega168**
- ATmega328**

Bild 3.3-01: AVR-Pin-Belegung ([Bildvergrößerung](#))

AVR-8-bit-Mikrocontroller

Gruppe 200 - Einsetzen von AVR-Tools

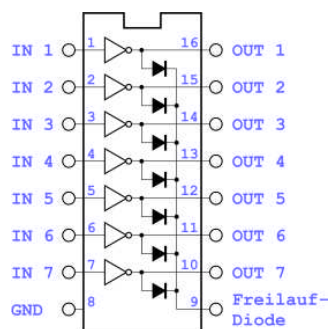
Teil 203 - AVR-ALE-Testboard

3.4 Anzahl LEDs und Tasten

Wenn man eine leere Europlatine in den Abmessungen 160 x 100 mm vor sich hat, wachsen die Begehrlichkeiten: Also wurde ein 2. Treiber **ULN2003** mit weiteren 7 LEDs und 2 weitere Tasten eingeplant. Was aber grundsätzlich fehlte, war eine **ISP10PIN**-Schnittstelle. Mit ihrer Einrichtung wurde ein Adapter von **ISP6PIN** nach **ISP10PIN** und umgekehrt gem. **Bild 2-01: ISP-Schnittstellen-Adapter** überflüssig.

Für die "Primitiv-Eingabe" sind 4 Taster und für die "Primitiv-Ausgabe" 7 rote und 7 gelbe LEDs vorhanden. Die LEDs werden von 14 Open-Collector-Ausgängen von 2 Chips [ULN2003](#) mit je 7 Leistungstreibern angesteuert. Wenn man so will, wird dabei "mit Kanonen auf Spatzen geschossen", denn der ULN2003 ist ein "richtiges Arbeitspferd" und ist für viel mehr zu gebrauchen.

Wichtige Funktion: Eine logische 1 an einem Leistungstreiber-Eingang des ULN2003 schaltet den gegenüberliegenden Open-Collector-Ausgang vom hochohmigen Zustand ("Schalter" offen) auf **GND** (d.h. "Schalter" geschlossen entsprechend 0 Volt), so dass das angeschlossene zu aktivierende Teil (z.B. die mit einem Vorwiderstand versehene LED) auf der Gegenseite mit einem positiven Spannungspotential (z.B. +5 Volt) versorgt werden muss. Die Treiber können aber auch unterschiedliche Schaltungen mit unterschiedlichen positiven Spannungen bis max. 50 Volt ansteuern. Und schließlich kann man sie noch parallel schalten, um stärkere Ströme zu ermöglichen.



- Jeder der 7 Leistungstreiber kann bis zu 0,5 A schalten und
- Ausgangsspannungen bis zu 50 V bewältigen,
- induktive Lasten können durch integrierte Freilauf-Dioden individuell geschützt werden,
- höhere Ströme können durch Parallelschaltung erzielt werden,
- die Eingänge sind TTL- und CMOS-kompatibel,
- die Ausgänge befinden sich für ein übersichtliches Layout gegenüber den Eingängen.

Bild 3.4-01: Pin-Belegung des Treibers ULN2003 ([Bildvergrößerung](#))

3.5 LCD-Interface und 20x4-LCD

Wenn das LCD 20x4 verwendet werden soll, dann werden die **CLK**-Impulse von Pin **PB1** des Mikrocontrollers mit einem Steckverbinder zum **CLK**-Pin von **K14** geleitet. Die **DATA**-Impulse gelangen von Pin **PB2** mit einem Steckverbinder zum **DATA**-Pin von **K14**. Über die Jumper **JP4 (GND)** und **JP5 (+5 V)** wird das LCD vom Testboard mit Spannung versorgt.

Mit einer Steckbrücken-Verbindung zwischen dem Plus-Kontakt von **K2** und dem Kontakt **BL+** kann die Helligkeit des LCD-Backlight ideal über den 10-Ohm-Einstellregler (Wendel-Ausführung) eingestellt werden. Mit dem 10-kOhm-Einstellregler wird der Kontrast der Zeichen auf dem LCD eingestellt. **Vorsicht:** Bei ungenügender Einstellung verschwinden die Zeichen auf dem LCD.

3.5.1 Erzeugung des Enable-Signals für das LCD

Eine Besonderheit der Schaltung ist die rechtzeitige Erzeugung des **Enable-Signals** für das LCD (**Pin06** von **K15**). Damit erst nach dem vollständigen Anliegen eines Nibbels an den Pins **Q2** bis **Q5** vom Schieberegister [TC74HC4094A](#) die Übertragung zum LCD passiert (**Pin11, Pin12, Pin13** und **Pin14** von **K15**), wird erst zu dem Zeitpunkt, an dem alle Bits im Shift-Register ihre korrekte Position erreicht haben, das Enable-Signal **E** aus dem in **Q7** erzeugten **HIGH** und einem gesonderten **DATEN**-Impuls durch ein logisches **UND** gebildet und an **Pin06** des LCD gesendet.

AVR-8-bit-Mikrocontroller Gruppe 200 - Einsetzen von AVR-Tools Teil 203 - AVR-ALE-Testboard

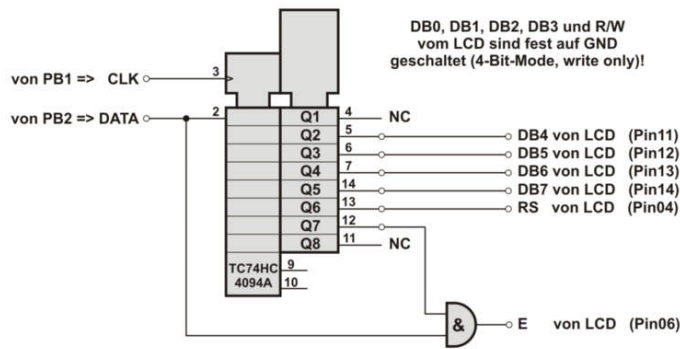
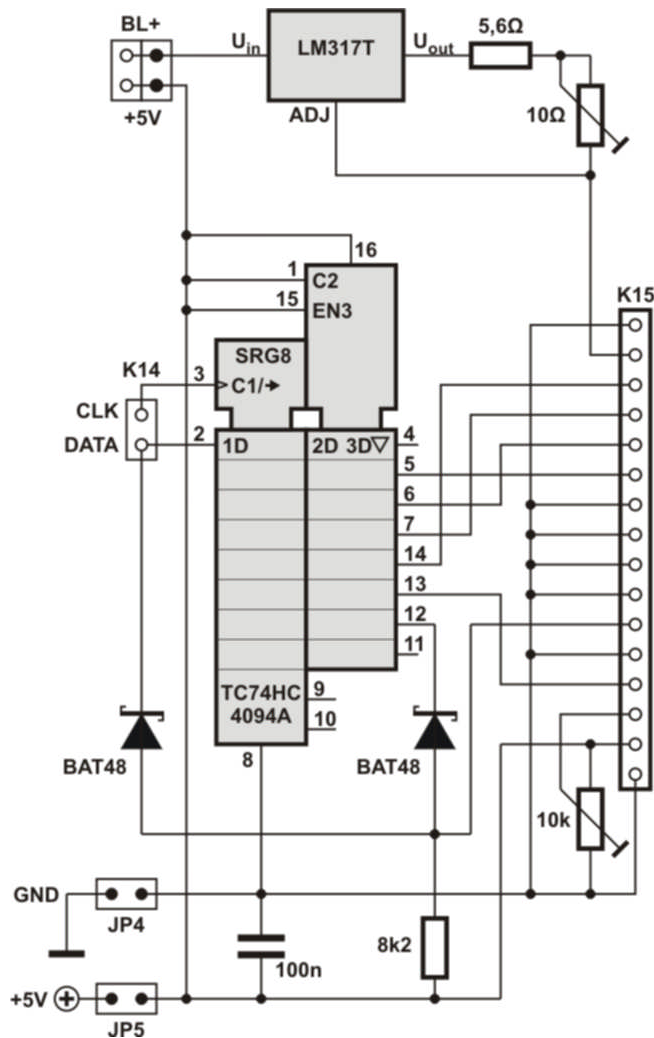


Bild 3.5.1-01: Erzeugung des Enable-Signals E ([Bildvergrößerung](#))

Dieses **UND** hat es offensichtlich "in sich". Denn wenn man die vielen Beiträge im ATM18-Forum zum Selbstbau des 2-Draht-LCD verfolgt, so wird man feststellen, dass immer wieder eine kryptische Anzeige bemängelt wird!

Auch das Original-LCD zeigte an gesondert aufgebauten Schieberegister-Schaltungen nach der ursprünglichen Bestückung ebenfalls nur durchlaufende Hieroglyphen und keine Zeilenumschaltung! Mit dem Original-LCD-Interface als auch mit dem hier dargestellten lief das LCD sowohl am Original-Testboard als auch an dem nachgebauten AVR-ALE-Testboard mit herkömmlichen Bauteilen einwandfrei!

Auch das Original-LCD zeigte an gesondert aufgebauten Schieberegister-Schaltungen nach der ursprünglichen Bestückung ebenfalls nur durchlaufende Hieroglyphen und keine Zeilenumschaltung! Mit dem Original-LCD-Interface als auch mit dem hier dargestellten lief das LCD sowohl am Original-Testboard als auch an dem nachgebauten AVR-ALE-Testboard mit herkömmlichen Bauteilen einwandfrei!



Der Grund lag wahrscheinlich in der ursprünglichen Realisation des UND-Gatters in der Form von nur einer Schottky-Diode und einem Widerstand. In alten Zeiten, als es noch keine ICs gab und man solche Schaltungen noch mit Dioden und Transistoren aufbauen musste, bestand das typische UND-Gatter für **jeden Eingang** aus einer Diode und einem gemeinsamen Pull-Up-Widerstand zwischen Ausgang und Spannungsversorgung, so dass die UND-Schaltung hier mit 2 Schottky-Dioden (BAT48) und einem Widerstand (8k2) gegen +5 Volt realisiert wurde und **es funktionierte auf Anhieb!!!**

Bild 3.5.1-02: Das LCD-Interface ([Bildvergrößerung](#))

Diese Feststellung wurde im Übrigen messtechnisch mit dem Beitrag von [franky69](#) nach einem Disput mit dem Entwickler der ansonsten großartigen Schaltung bestätigt.

Das die zweifelsfrei funktionierende Original-Schaltung auch ohne die 2. Diode läuft, ist vermutlich - aus Sicht des Autors - nur so zu erklären, dass der 4k7-Widerstand und der Durchlass- und Sperrwiderstand der Schottky-Diode BAS70 sowie die Impedanz des LCD optimal aufeinander abgestimmt sind, was mit "Bordmitteln" wohl nicht allgemein erreicht werden kann.

Wenn man sich die Ersatz-Schaltungen der TTL's anschaut, wird man auch feststellen, dass die Gatter aus einer entsprechenden Anzahl integrierter Dioden oder äquivalenter Transistoren mit mehreren Emitter-Eingängen bestehen.

3.5.2 LCD-Backlight

Für die Strombegrenzung vom LCD-Backlight wurde eine Lösung mit der für LEDs notwendigen konstant-Stromquelle (einstellbar auf den gewünschten Bereich) mittels [LM317T](#) gefunden. Einen Vor-

AVR-8-bit-Mikrocontroller

Gruppe 200 - Einsetzen von AVR-Tools

Teil 203 - AVR-ALE-Testboard

schlag, die Steuerung auch dem Mikrocontroller anzuvertrauen, ist zwar reizvoll aber für die Praxis scheint eine "Hardware-Lösung" besser geeignet zu sein. Es spart außerdem Port-Pins.

Der steuerbare Spannungsregler **LM317T** lässt sich auch als Stromregler benutzen. Seine Regelung beruht auf eine rückgekoppelte Referenzspannung von genau 1,25 V (am Pin **ADJ**). Ein Widerstand im Ausgangszweig, an dem bei dem gewünschten Treiberstrom (zwischen 80 und ca. 220 mA) genau 1,25 V abfallen stellt die Lösung dar. Da die Spannungsdifferenz von 3,75 V (zwischen den vorhandenen +5 V und der Regelspannung von +1,25 V) nicht ausreicht, die LEDs zu speisen, wurden ein Jumper und die Buchse **BL+** vorgesehen, um über **K2** (+6 bis +9 V) eine getrennte Spannungsversorgung des Backlights vornehmen zu können.

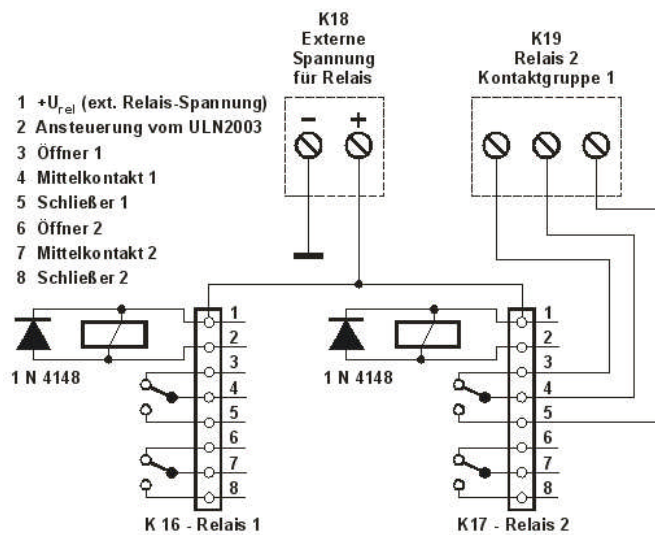
Für **80 mA** ergibt sich ein Widerstand R1 zu $1,25 \text{ V} \times 0,08 \text{ A}$ gleich 15,625 Ohm (rechnerisch!)
 Für **220 mA** ergibt sich ein Widerstand R2 zu $1,25 \text{ V} \times 0,22 \text{ A}$ gleich 5,68 Ohm (rechnerisch!)

Gewählt wurde ein Festwiderstand von **5,6 Ohm / 0,6 W** und ein Einstellregler von **10 Ohm / 0,5 W**, ergibt zusammen einen Widerstandsbereich zwischen **5,6** und **15,6 Ohm**. Damit dürfte der Bereich für den Backlight-Strom von **ca. 80 bis 220 mA** ohne Änderung (!) am LCD-Modul erfüllt werden können.

Wenn eine herkömmliche Lösung bevorzugt wird, kann der Regler LM317T auch entfallen. Die für U_{in} und U_{out} vorgesehenen Bohrungen erhalten dann eine Brücke, der Widerstand 5,6 Ohm wird ggf. erhöht (10 Ohm) und es wird der Jumper auf +5 V gesetzt.

Eine ausführliche Beschreibung des 2-Draht-LCD-Interfaces, des 20x4-LCD's und des Steuerungsprogramms siehe **Gruppe 600 - AVR-C-Projekte, Teil 602 - 2_Draht_LCD**

3.6 Ansteuerung von Relais



Die beiden zusätzlich steckbaren Relais werden extern über Klemmen mit Spannung versorgt, um Print-Relais unterschiedlicher Spannung einsetzen zu können. Natürlich können auch passende Print-Relais mit den +5 Volt der Platine versorgt werden (z.B. [TAKAMISAWA-Relais NA-5 W-K](#)). Aber Vorsicht: Wegen der Polarisierung dieser Relais musste das Layout der Platine extra angepasst werden!

Bild 3.6-01: Relais-Steuerung
[\(Bildvergrößerung\)](#)

Die Ansteuerung eines Relais erfolgt über **Pin2** von **K16** bzw. **K17** und einem Ausgang vom Treiber **ULN2003**, der im Aktionsfall (logische 1 am Eingang des Treibers) das Beinchen auf **Low** zieht und damit das Relais betätigt.

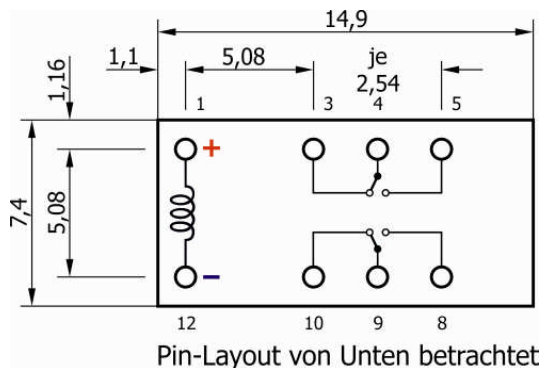


Bild 3.6-02: Gepoltes TAKAMISAWA-Relais
[\(Bildvergrößerung\)](#)

AVR-8-bit-Mikrocontroller Gruppe 200 - Einsetzen von AVR-Tools Teil 203 - AVR-ALE-Testboard

3.7 RS-232-Schnittstelle

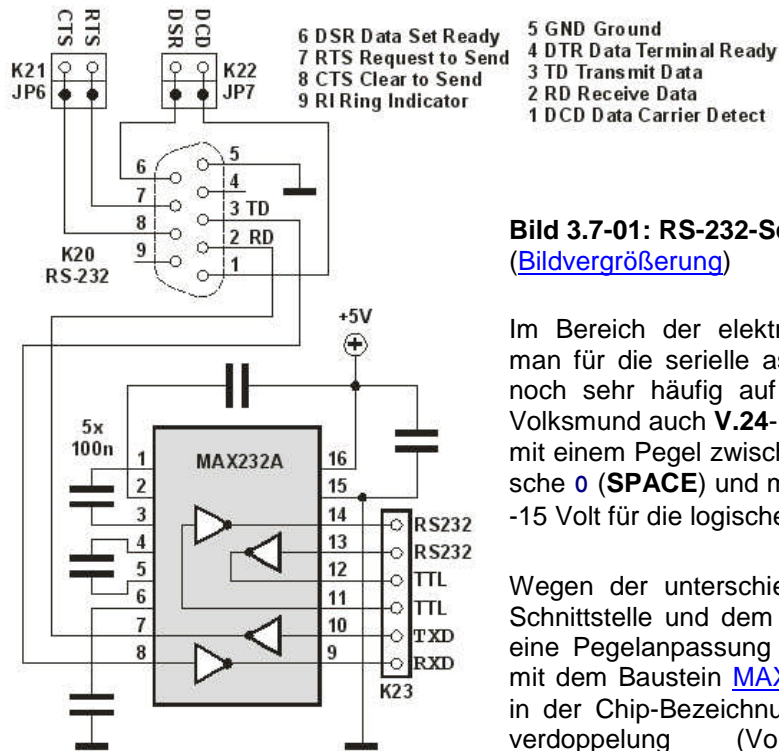


Bild 3.7-01: RS-232-Schnittstelle
([Bildvergrößerung](#))

Im Bereich der elektronischen Übertragungstechnik trifft man für die serielle asynchrone Datenübertragung immer noch sehr häufig auf die sog. **RS-232**-Schnittstelle (im Volksmund auch **V.24**-Schnittstelle genannt). Diese arbeitet mit einem Pegel zwischen +3 Volt und +15 Volt für die logische **0 (SPACE)** und mit einem Pegel zwischen -3 Volt und -15 Volt für die logische **1 (MARK)**.

Wegen der unterschiedlichen Pegelstruktur der RS-232-Schnittstelle und dem AVR-System mit seinen +5 Volt ist eine Pegelanpassung notwendig, die auf einfache Weise mit dem Baustein [MAX232A](#) erreicht werden kann. Das **A** in der Chip-Bezeichnung bedeutet, dass die Spannungsverdoppelung (Voltage Doubler) und die Spannungsinvertierung (Voltage Inverter) des Bausteins mit 4 Kondensatoren von nur 100 nF erreicht werden kann (ohne das **A** werden Elkos von mindestens 1 µF benötigt).

Die Anbindung an den Mikrocontroller erfolgt über **TXD**-Pin und **RXD**-Pin von **K23** mit den entsprechenden Pins von **K5 USART**. Empfehlenswert ist es, **Pin 1** und **Pin 6** von **K20** mit Jumper **JP7** zu überbrücken und über **K22** mit dem Treiberausgang von **MAX232A Pin 14** zu verbinden und dessen Eingang **Max232A Pin 11** auf Masse zu legen.

Pin 7 und **Pin 8** von **K20** sollten noch am Jumper **JP6** überbrückt werden, so dass sich der PC selbst von **RTS** (Request to Send) auf **CTS** (Clear to Send) rückkoppelt.

3.8 USART-Testboard-Schnittstelle

Die **USART**-Schnittstelle (USART = **U**niversal **S**ynchronous and **A**synchronous Serial **R**eceiver and **T**ransmitter) über **K5** ist eine nicht invertierte TTL-kompatible Schnittstelle, die direkt für den [TTL-232R USB-Serial-Converter](#) von der Firma **FTDI** geeignet ist.

Das **TTL-232R** ist ein Kabel, das eine einfache Verbindung zwischen einem Gerät, das mit TTL-Pegel arbeitet (z.B. das Testboard) und dem USB am PC herstellt. Die Verbindung verwendet den Chip [FT232RQ](#), der in dem Steckerteil des **USB A** integriert ist. Das 1,8 Meter Kabel endet in einem 6-poligen Steckverbinder, der mit den Signal-Leitungen Transmit (**Tx**), Receive (**Rx**), den Handshake-Leitungen **RTS#** und **CTS#** sowie der Stromversorgung (aus dem USB-Port) **+5V** und **GND** beschaltet ist. Je nach Kabel-Typ können zum 6-poligen Steckverbinder +5 Volt (TTL-232R-5V) oder +3,3 Volt (TTL-232R-3V3) durchgeschaltet sein.



RTS# und **CTS#** werden auf dem Testboard nicht verwendet.

Bild 3.8-01: [TTL-232R USB-Serial-Converter](#)

