

LCD-Terminal mit serieller Schnittstelle

- Handbuchauszüge -

Quelle: Beckmann+Egle GmbH, (c) 1994

BELC 1001 Controller für DMC-Modul mit Festtextanzeige

1

Mit den *MiniTerminals* der Serie BELC100x können alle gängigen LCD-Character-Module an eine V24-Schnittstelle angeschlossen werden.

Die ESC-Sequenzen zur Steuerung des Displays sind an den VT100 Standard angelehnt.

Das *MiniTerminal* BELC1001 verfügt ausserdem über einen Festtextspeicher von bis zu 6k Byte, wobei der BELC1001-Controller mit max. 3 EEPROMs zu je 2kByte bestückt werden kann. Die 3 EEPROMs dienen ausschließlich der Speicherung von Festtexten. Standardmäßig ist der BELC1001-Controller mit einem EEPROM bestückt (2 weitere sind optional). Die Festtexte können über die serielle Schnittstelle und/oder über 7 TTL-Leitungen abgerufen werden.

D.h. es gibt grundsätzlich *zwei* Möglichkeiten die Festtexte abzurufen :

- a.** intern über die serielle Schnittstelle (PC) anhand einer entsprechenden ESC-Sequenz
- b.** extern über 7 TTL-Leitungen (FT0-FT6) am Stecker J4

Die Ausgabe von Zeichen auf dem Display ist unabhängig von der Festtextausgabe und kann mit dieser beliebig gemischt werden (siehe Demo-Programm).

Das *MiniTerminal* kennt *zwei* verschiedene Betriebsarten :

- 1.** Betriebsart '**Bus**'
=> liegt vor, wenn beim Einschalten Pin '**FT/Bus**' low(0) gesetzt ist
- 2.** Betriebsart '**automatischer Festtext**'
=> liegt vor, wenn beim Einschalten Pin '**FT/Bus**' high(1) gesetzt ist

Wird in dieser *MiniTerminal*-Beschreibung die Bezeichnung "Pin 'FT/Bus'" verwendet, so ist immer entweder der Pin 'FT/Bus' bei BR5 oder der Pin 8 am Stecker J4 gemeint.

zu beachten :

- Der Pin 'FT/Bus' (Pin BR5 oder Pin 8 an J4) **muss** mit Masse (Betriebsart 'Bus') oder +5 V (Betriebsart 'automatischer Festtext') verbunden werden. Dies gewährleistet, daß beim Einschalten das *MiniTerminal* auf eine der beiden Betriebsarten festgelegt wird.
- Der Pin '**FT/Bus**' (BR5) und der Pin 8 am Stecker J4 sind als elektrisch gleich zu bewerten. Dies bedeutet daß **nie** unterschiedliche Pegel an diesen beiden Pins anliegen dürfen.

zu 1. :

In der Betriebsart '**Bus**' können bis zu 16 *MiniTerminals* über die serielle Schnittstelle angesteuert werden. Die Ansteuerung erfolgt über eine ESC-Sequenz (siehe Befehl 'Adresse vorgeben'), anhand der, durch Angabe einer '*adresse*', das entsprechende *MiniTerminal* angesteuert wird. Die Zuordnung eines *MiniTerminal*s zu einer Adresse wird auf Seite 6 unter dem Begriff 'Brücken' erklärt.

Der Abruf der Festtexte über die 7 TTL-Leitungen(FT0-FT6) ist in diesem Mode **nicht möglich**. Über die serielle Schnittstelle jedoch ist der Abruf **möglich**.

zu 2. :

In der Betriebsart '**automatische Festtextanzeige**' können die gespeicherten Festtexte **zusätzlich** zur seriellen Schnittstelle auch über die Pins FT0-FT6 angewählt werden. Wie sich das *MiniTerminal* hierbei verhält wird beim Befehl 'ESC D nr' näher beschrieben.

zu beachten :

- Wenn Sie das *MiniTerminal* für '**automatischer Festtext**' konfigurieren(Pin 'FT/Bus' ist high(1)), **müssen** die Anschlüsse FT0 .. FT6 einen definierten TTL-Pegel aufweisen ! Dies bedeutet, wenn die externen Anschlüsse FT0-FT6 (J4) verwendet werden, dann **müssen** alle Anschlüsse angeschlossen werden(siehe Abb.' Ansteuerung zur Festtext-Auswahl'). Auch dann, wenn z.B. nur die Anschlüsse FT0-FT3 benötigt werden. Denn nicht angeschlossene Pins werden als high(1) interpretiert.

Serielle Schnittstelle (V24)

9600 Baud / 8 Data-Bits / no parity / 2 Stop-Bits

Die serielle Schnittstelle kann nicht gleichzeitig senden und empfangen. Sie müssen deshalb das CTS-Signal beachten, wenn Sie die Tastatur verwenden.

Wird das *MiniTerminal* mit einem PC angesteuert, kann die Schnittstelle beispielsweise in Qbasic mit folgendem Befehl initialisiert werden:

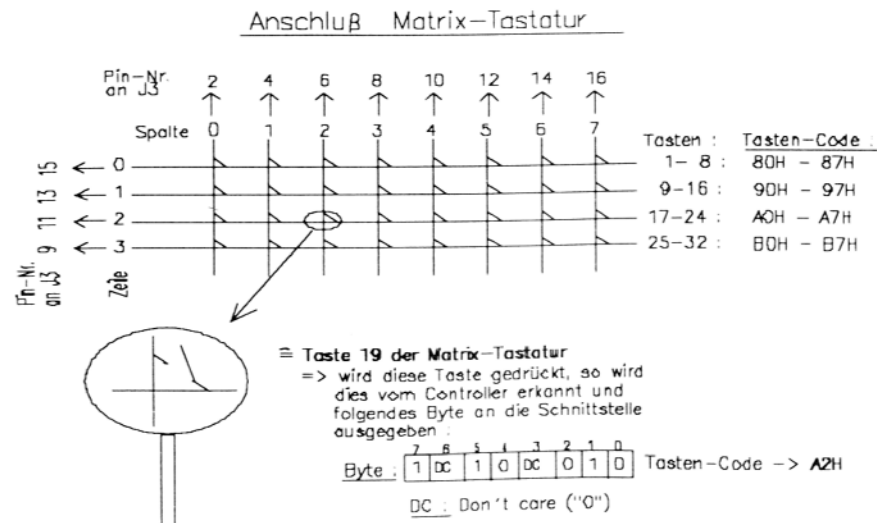
```
OPEN "COM1: 9600,N,8,2,cs0,ds0,cd0,rs" AS 1
```

Matrix-Tastatur (8 x 4 Tasten)

Die Matrix-Tastatur kann aus maximal 8 Spalten und 4 Zeilen (32 Tasten) bestehen. Sie wird am Stecker J3 angeschlossen.

Wird eine Taste gedrückt, setzt das *MiniTerminal* die CTS-Leitung und sendet den Tasten-Code. Der Tasten-Code wird an die serielle Schnittstelle weiter geleitet. Dort kann der Code durch ein Programm ausgelesen und beliebig weiterverarbeitet werden.

- Bit4.. Bit5 = Zeilen-Nummer -> da nur 4 Zeilen zu unterscheiden sind wird Bit6 nicht benutzt (DC)
- Bit0 .. Bit3 = Spalten-Nummer -> da nur 8 Spalten zu unterscheiden sind wird Bit3 nicht benutzt (DC)
- Bit7 ist immer '1'



Escape-Sequenzen zur Steuerung des Displays**3**

CR Wagen-Rücklauf / Neue Zeile 0d (hex) / 13 (dezimal)
 ESC 1B (hex) / 27 (dezimal)

ESC [ParZ ; ParS H	Cursor setzen
----------------------------	---------------

Zeile 0.. 3 / Spalte 0..39 Alle Zahlen ohne führende Nullen !

Beispiel: **PRINT #1,CHR\$(27);"[";CHR\$(2);";";CHR\$(5);"H";**
 setzt den Cursor in die 6. Spalte der 3. Zeile

ESC & #	Display löschen / Cursor Home
--------------------	-------------------------------

Beispiel: **PRINT #1,CHR\$(27);"&#";**

ESC & C	Cursor ein
--------------------	------------

Beispiel: **PRINT #1,CHR\$(27);"&C";** schaltet den Cursor ein

ESC & D	Cursor aus
--------------------	------------

Beispiel: **PRINT #1,CHR\$(27);"&D";** schaltet den Cursor aus

ESC [K	Bis zum Ende der Zeile löschen
----------------	--------------------------------

Beispiel: **PRINT #1,CHR\$(27);"[K";** löscht alle Zeichen bis zum Ende der aktuellen Zeile

Transparent-Mode :

ESC & T	in Transparent-Mode wechseln
--------------------	------------------------------

Achtung : Diesen Mode sollten nur gute Kenner des Controllers benutzen

Im Transparent-Mode hat der Host direkten Zugriff auf den Controller des Displays (HD44780 oder ähnlich)

x0 data	Write CMD (Adresse 0)
x1 data	Write Data (Adresse 1)
x2	Read CMD
x3	Read Data
x4 data	Controller auswählen (nur 40x4)
	data = x0 -> Controller 1
	data = x1 -> Controller 2
sonst	Transparent-Mode verlassen

data Zeichen in Hex
 x don't care (der High-Nipple ist ohne Bedeutung)

Beispiel: **PRINT #1,CHR\$(27);"&T";** ' Enter Transparent Mode
PRINT #1,CHR\$(0);CHR\$(2); ' 2 in Controller-Adresse 0 schreiben
PRINT #1,CHR\$(255); ' Transparent-Mode verlassen

Select Display Type :

ESC & S n	Display-Type auswählen
----------------------	------------------------

n =	0 (^@)	16 x 1
	1 (^A)	16 x 2
	2 (^B)	16 x 4
	3 (^C)	20 x 1
	4 (^D)	20 x 2
	5 (^E)	20 x 4
	6 (^F)	24 x 1
	7 (^G)	24 x 2
	8 (^H)	32 x 1
	9 (^I)	32 x 2
	10 (^J)	40 x 1
	11 (^K)	40 x 2
	12 (^L)	40 x 4

4

Beispiel: **PRINT #1,CHR\$(27);"&S";CHR\$(2);**
Initialisiert das *MiniTerminal* für ein 16 x 4 Display

Der ausgewählte Display-Typ wird im
EEPROM abgespeichert (wenn vorhanden).

Adresse vorgeben :

ESC & A adresse	Adresse vorgeben
ESC & A Z	immer selektiert (Default-Wert)

=> d.h. alle angeschlossenen *MiniTerminals* sind selektiert

adresse =	0
	1
	:
	9
	10
	11
	12
	13
	14
	15

Beispiel : **Print #1, CHR\$(27); "&A";CHR\$(1)**
=> nur *MiniTerminal* mit zugeordneter Adresse 1
wertet Daten aus

Print #1, CHR\$(27); "&AZ";
=> alle angeschlossenen *MiniTerminals* werten
Daten aus (Default-Einstellung)

Ist der Pin '**FT/Bus**' beim Einschalten low (0), geht das *MiniTerminal* in die Betriebsart '**Bus**' und interpretiert die Pins **Adr0 .. Adr3**(BR1-BR4) als Bus-Adresse.

zu beachten :

Die Pins **Adr0 .. Adr3**(BR1-BR4) sind elektrisch gleichzusetzen mit Pins **FT0 .. FT3** (J4).
D.h. nie Pins FT0 .. FT3 auf Masse legen und dann gleichzeitig Pins an BR1 - BR3 auf +5V legen.

Ist der Pin '**FT/Bus**' beim Einschalten high (1), geht das *MiniTerminal* in die Betriebsart '**automatische Festtextanzeige**' und interpretiert die Pins **FT0 .. FT6** als Festtext-Nummer.

In der Betriebsart '**Bus**' kann jedem *MiniTerminal* über die Pins '**Adr0 .. Adr3**' eine Adresse zugeordnet werden (0..15). Wird vom Host das Kommando '**Adresse vorgeben**' gesendet, wertet nur noch das *MiniTerminal* mit dieser Adresse die folgenden Daten aus.

Wird an einem nicht selektierten *MiniTerminal* eine Taste gedrückt, so wird der Code gespeichert. Er wird gesendet, sobald das *MiniTerminal* selektiert wird.

Nach dem Einschalten sind alle angeschlossenen *MiniTerminals* selektiert und werten somit Daten aus (immer selektiert).

Zeichen ändern : (permanent)

ESC & L CG	Zeichen 8x Muster	bestehendes Zeichen umdefinieren
-----------------------	-------------------	----------------------------------

- CG = Zu verwendende CG-RAM-Adresse. Es stehen 8 Adressen zur Zeichenumdefinierung zur Verfügung.
d.h. CG = Zahl von 1 - 8
- Zeichen = Zeichen, welches umdefiniert werden möchte.
d.h. hier kann ein beliebiges Zeichen des Character-Fonts angegeben werden
- 8x Muster = Hier muß das Bit-Muster des neuen Zeichens angegeben werden. Ein Zeichen besteht aus einer '5x7 Punkt-Matrix'
d.h. nur 'Bit0 - Bit4' der 7 anzugebenden Bytes sind relevant. Als 8. Byte wird immer 'chr(0)' angegeben.

Beispiel-Proramm (Qbasic) :

```
'Schnittstelle definieren
OPEN "COM1:9600,N,8,2,cs0,ds0,cd0,rs" AS 1
'Zeichen-Muster
pattern$ = CHR$(85)+CHR$(170)+CHR$(85)+CHR$(170)+CHR$(85)+CHR$(170)+CHR$(85)+CHR$(0)
INPUT "Umzudefinierendes Zeichen .:"; zch$
INPUT "Zu verwendende CG-RAM-Adresse (1-8) .:"; CGAdr%
'ESC-Sequenz zur Zeichenumdefinierung
PRINT #1, CHR$(27) + "&L";
PRINT #1, CHR$(CGAdr% - 1);
strg$ = zch$ + pattern$
I = LEN(strg$)
IF I <> 0 THEN
  FOR n = 1 TO I
    PRINT #1, MID$(strg$, n, 1);
    R$ = INPUT$(1, #1)
    IF R$ <> MID$(strg$, n, 1) THEN
      PRINT "Fehler beim Speichern": BEEP
    EXIT FOR
  END IF
NEXT n
END IF
```

Möglichkeit Zeichenumdefinierung rückgängig zu machen:

Für umzudefinierendes Zeichen (zch\$) einen Wert zwischen chr(10) - chr(18) eingeben (nicht darstellbares Zeichen) und zu löschende CG-RAM-Adresse (CGAdr%) angeben.

Escape-Sequenz zur Steuerung der Datenübertragungsgeschwindigkeit

ESC & Bn	Einstellen der Baudrate
---------------------	-------------------------

- n = 0 - 9600
 1 - 4800
 2 - 2400
 3 - 1200

Beispiel: **PRINT #1,CHR\$(27);"&B";CHR\$(2);**
 setzt die Baudrate auf 2400 Bits pro Sekunde

Diese Einstellung wird im EEPROM permanent abgespeichert.

Escape-Sequenz zum Ändern der Tasten-Codes**6**

ESC & u Byte0 Byte31	Belegung des Tasten-Codes ändern
---------------------------------------	----------------------------------

Beispiel:

```

PRINT #1,CHR$(27);"&u";
TEXT$ = "abcdefghijklmnopqrstuvwxy,,:?" ' neuer Tasten-Code
L = LEN(TEXT$)
FOR I=1 TO L
PRINT #1,MID$(TEXT$,I,1);
R$ = INPUT$(1,#1)
IF R$ <> MID$(TEXT$,I,1) THEN GOTO fehler ' Fehler beim Speichern
NEXT I
fehler:
PRINT "Fehler beim Speichern"
ende:
END

```

Der originale Tasten-Code wird, durch die dem "u" folgenden 32 Byte, der Reihe nach ersetzt.
Der originale Tasten-Codes sieht folgendermaßen aus:

```

00H - 07H    -> Taste 1 - 8
10H - 17H    -> Taste 9 - 16
20H - 27H    -> Taste 17 - 24
30H - 37H    -> Taste 25 - 32

```

Die Tasten-Codes werden wie die Festtexte, ebenfalls im EEPROM permanent abgespeichert.

zu Beachten :

Die Einstellungen der Baudrate und der Umcodiertabelle werden auch bei Spannungsausfall gehalten. Diese Einstellungen werden nur durch die o.g. Escape-Sequenzen geändert.

Kommandos für die Festtext-Anzeige

Die Festtexte sind mit 0 .. 127 durchnummeriert.
Vor der Ausgabe eines Festtextes wird das Display gelöscht.
Ein Festtext darf als Sonderzeichen CR enthalten.

Der BELC1001-Controller kann mit maximal 3 EEPROMs mit je 2kByte bestückt werden.

Festtexte auslesen : (alle)

ESC W	Festtexte über die serielle Schnittstelle ausgeben
--------------	--

Beispiel: **PRINT #1,CHR\$(27);"W";**

Der BELC1001-Controller gibt alle gespeicherten Festtexte der Reihe nach über die serielle Schnittstelle aus. Jeder Festtext wird durch eine Null abgeschlossen. Die Übertragung wird durch 2 Nullen abgeschlossen.

Festtext laden :

ESC R	Festtexte über ser. Schnittstelle laden und im EEPROM abspeichern
--------------	---

Beispiel: **PRINT #1,CHR\$(27);"R";**
 PRINT #1,"T";
 INPUT\$(1,#1)

 PRINT #1,CHR\$(0);
 PRINT #1,"K";
 INPUT\$(1,#1)

 PRINT #1,CHR\$(0);
 PRINT #1,CHR\$(0);

1. Zeichen("T") des 1. Festtextes
warten, bis das Zeichen zurückgeschickt wird

Ende des 1. Festtextes
1. Zeichen("K") des 2. Festtextes
warten, bis das Zeichen zurückgeschickt wird

Ende des n. Festtextes
Ende der Übertragung

7

Der BELC1001-Controller schickt jedes Zeichen nach dem Programmieren zurück. Wurde beim Programmieren ein Fehler festgestellt (z.B. EEPROM defekt oder Speicherplatz erschöpft), invertiert der BELC1001-Controller das Zeichen. Es ist daher sinnvoll beim Speichern von Festtexten das zurückgeschickte Zeichen aus dem Schnittstellenpuffer wieder auszulesen und auf Gleichheit mit dem abgeschickten Zeichen zu prüfen (siehe Demo-Programm Zeile 230-250).

Das Speichern der Texte dauert ungefähr 1 sek pro 100 Zeichen.

Achtung: Das Kommando **löscht** alle gespeicherten Texte !
D.h. sobald man zusätzliche Festtexte im EEPROM abspeichern möchte werden alle bisher gespeicherten Festtexte gelöscht. Es müssen somit alle seither gespeicherten Festtexte und der gewünschte zusätzliche Festtext neu abgespeichert werden.

Speicherformat : 'FT0'+Chr\$(0)+ 'FT1'+Chr\$(0)+ 'FT2'+Chr\$(0) 'FT0n'+Chr\$(0)+Chr\$(0)
(im EEPROM)

Festtext ausgeben : (spezieller)

ESC D nr	Festtext mit Nummer 'nr' anzeigen ('nr' von 0 .. 127)
-----------------	---

Beispiel: **PRINT #1,CHR\$(27);"D";CHR\$(2);**

Der BELC1001-Controller löscht das Display und zeigt den 3. Festtext an.
Dieser Befehl überschreibt die Einstellung am Stecker J4.
Ändert sich der Wert am Stecker J4, wird wieder der Festtext, der dem Wert am Stecker J4 zugeordnet ist, angezeigt. Das bedeutet eine Festtextnummer wird immer dann angewählt, wenn entweder über die serielle Schnittstelle eine Nummer vorgegeben wird oder sich am Stecker J4 die Einstellung ändert.

Hinweis: Die Festtext-Befehle können unabhängig vom Wert des Pins '**FT/Bus**' verwendet werden.
D.h. über die serielle Schnittstelle können in **beiden** Betriebsarten ('autom. FT' oder 'Bus') Festtexte angewählt werden.

allgemeine Hinweise :

- Der Trimmer P1 dient zum Einstellen des Kontrastes.
Der Kontrast **muss** für jedes LCD-Modul separat eingestellt werden (siehe Beschreibung 'Inbetriebnahme des MiniTerminals').
- Das MiniTerminal kann keine Daten über die serielle Schnittstelle empfangen, solange es ein Kommando ausführt. Dies kann, speziell bei langsamen Kommandos (z.B. Display löschen), zum Verlust von Zeichen führen.

Steckerbelegungen :**8****J3 Matrix-Tastatur TTL-Eingänge**Zuordnung der
Zeilen zu PinsZuordnung der
Spalten zu Pins

Spalte 0 ->	2	1	
Spalte 1 ->	4	3	
Spalte 2 ->	6	5	
Spalte 3 ->	8	7	
Spalte 4 ->	10	9	<- Zeile0
Spalte 5 ->	12	11	<- Zeile1
Spalte 6 ->	14	13	<- Zeile2
Spalte 7 ->	16	15	<- Zeile3

Pin nr an Stecker J3

J4 Festtext-Auswahl TTL-Eingänge ↔ Brücken

1	FT0	Nummer des Festtextes der ausgegeben	↔ BR1
2	FT1	werden soll	↔ BR2
3	FT2	(7 Bit / 0 127 binär codiert)	↔ BR3
4	FT3		↔ BR4
5	FT4		
6	FT5		
7	FT6		
8	FT/Bus	automatischer Festtext (1) / Bus (0)	↔ BR5
9	Masse		

Zeichen '↔' bedeutet 'elektrisch gleich'

J6 Host / Versorgungsspannung

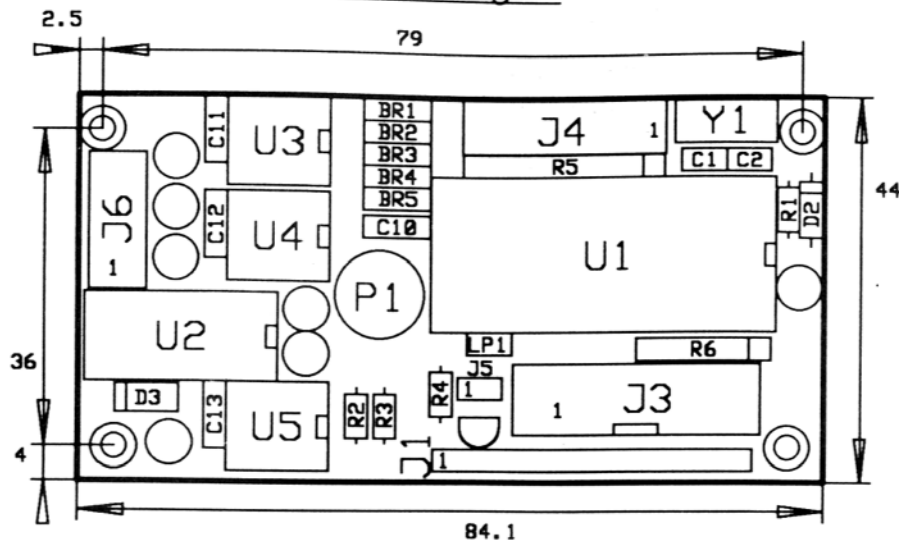
1	+ 5 V	Versorgungs-Spannung
2	TxD	V24 Ausgang
3	RxD	V24 Eingang
4	CTS	
5	Masse	
6	nicht verwendet	

Brücken :

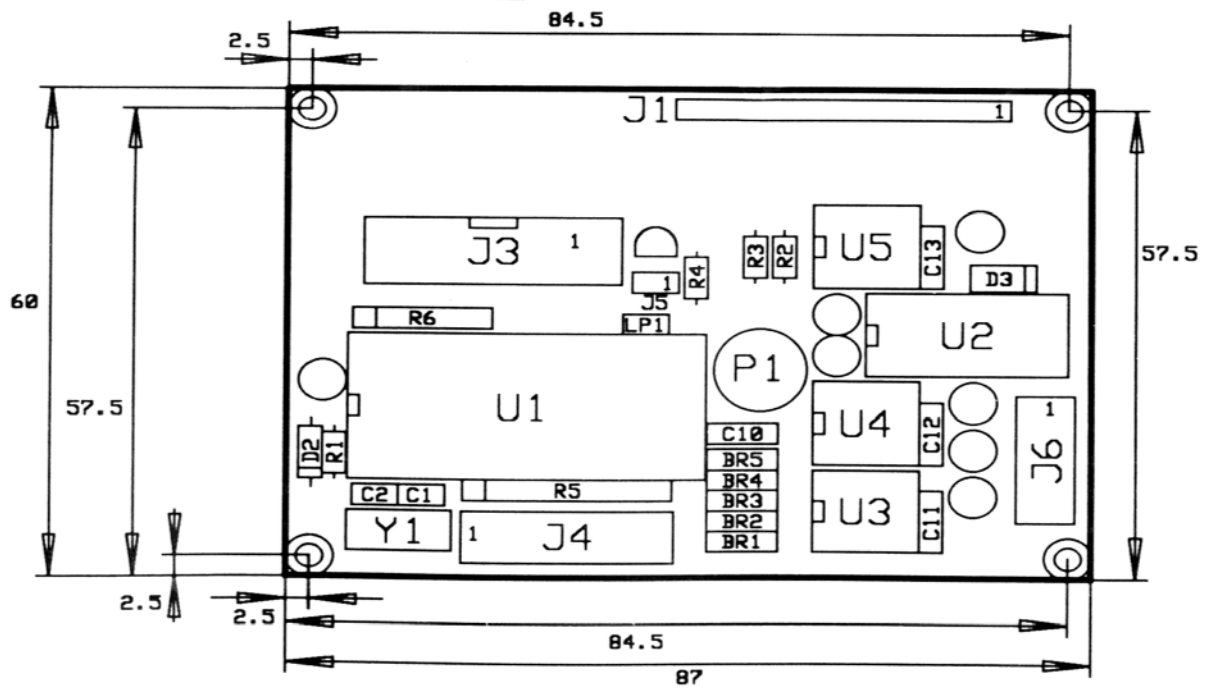
Mit Hilfe der Pins Adr0-Adr3 muss jedem angeschlossenen *MiniTerminal* eindeutig eine Adresse (0-15) zugeordnet werden. Diese Zuordnung erfolgt (extern auf der Platine) anhand der Brücken BR1-BR4, indem entweder die Brücken mit Pegel 0(Masse) oder Pegel 1(+5 Volt) verbunden werden. Ohne diese Zuordnung kann bei mehreren angeschlossenen *MiniTerminals* keine eindeutige Ansteuerung über die serielle Schnittstelle(Befehl 'Adresse vorgeben') erfolgen.

für 16 x2 LCD-Anzeigen:

9



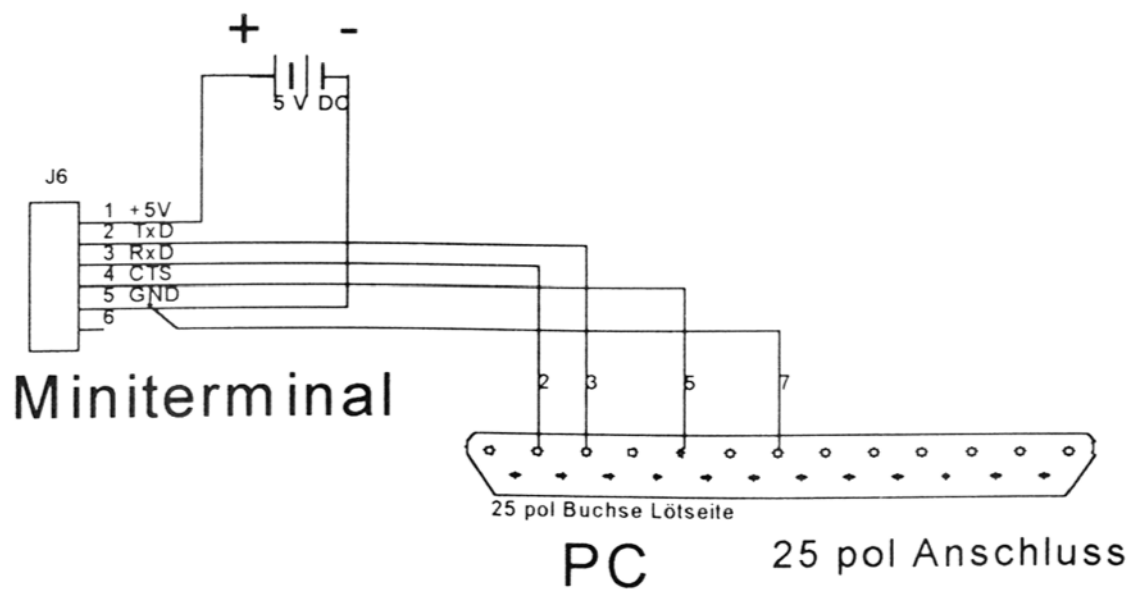
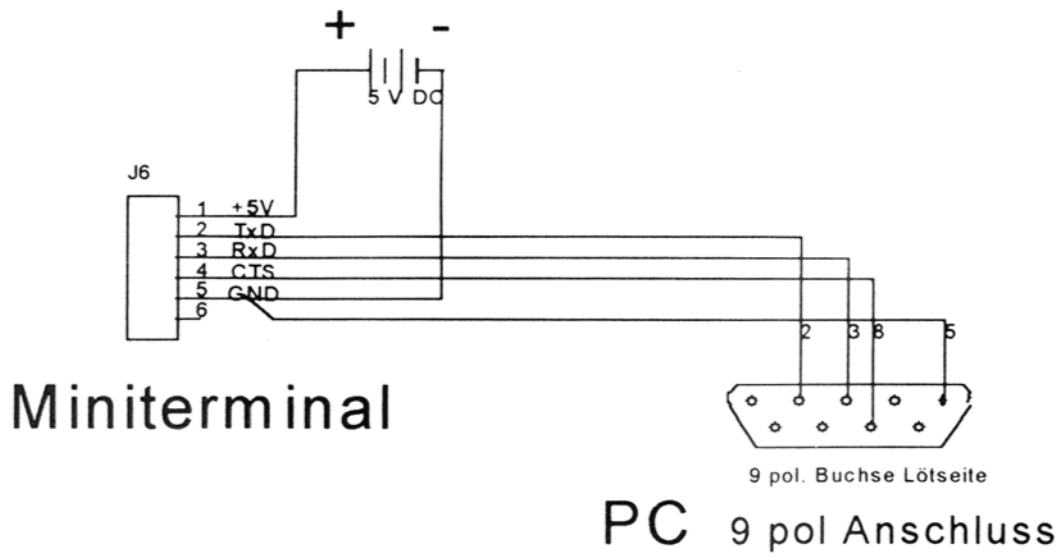
für 16x4 LCD-Anzeigen :



zusätzlich lieferbare Platinengrößen :

- 8x2 -> Maße : 58 mm x 32 mm (SMD)
- 16x1 -> Maße : 36.1mm x 80 mm
- 40x2 -> Maße : 182 mm x 33.5mm
- 40x4 -> Maße : 96 mm x 54 mm

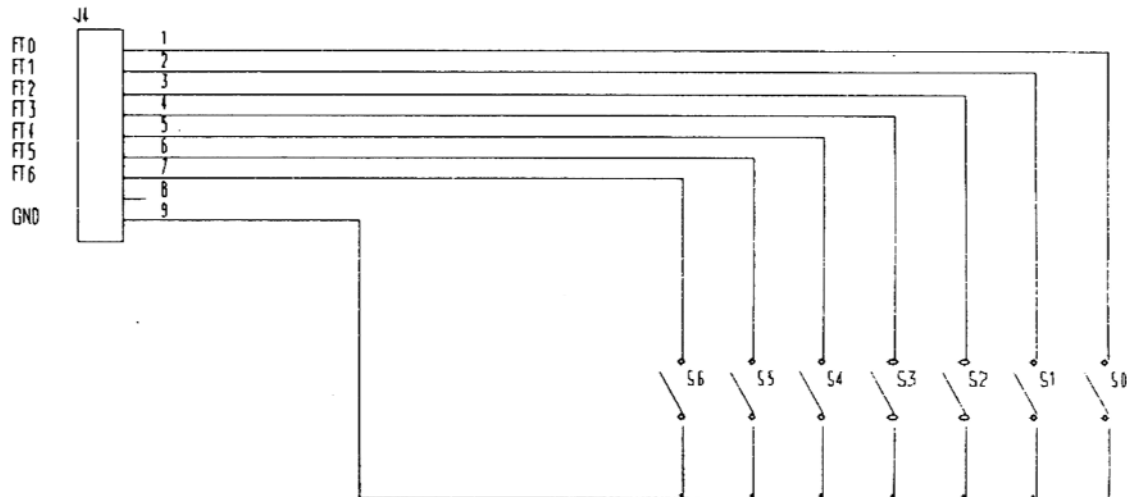
Kabel MiniTerminal <--> PC 10



Ansteuerung der Festtextauswahl :

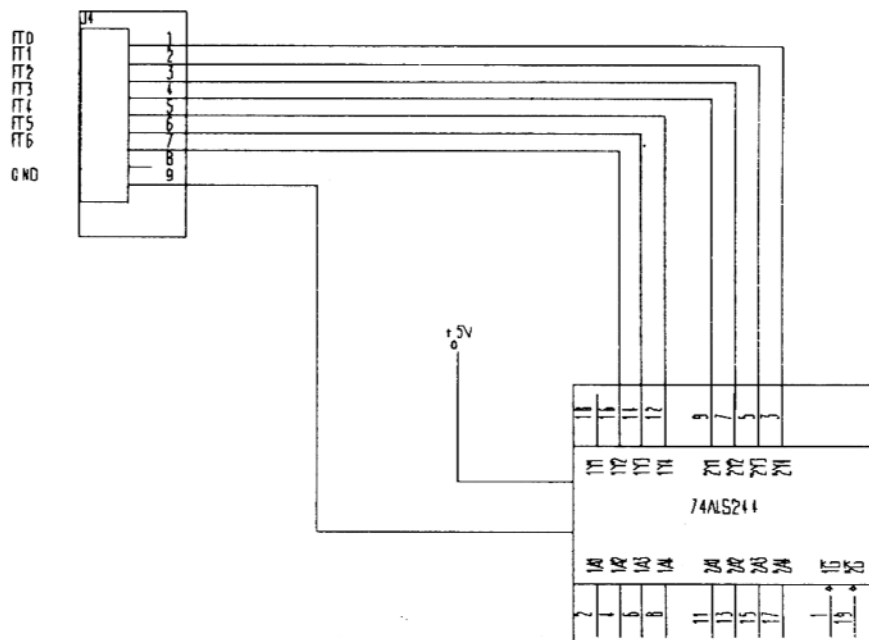
11

Kontakt – Anschluss



Bei offenem Schalter wird der Pegel am Pin als high(1) interpretiert
 Bei geschlossenem Schalter als low(0)

TTL – Anschluss



Mit den Eingängen 1A1 - 1A4 und 2A1 - 2A4 können die Anschlüsse FT0 - FT6 an J4 angesteuert werden.