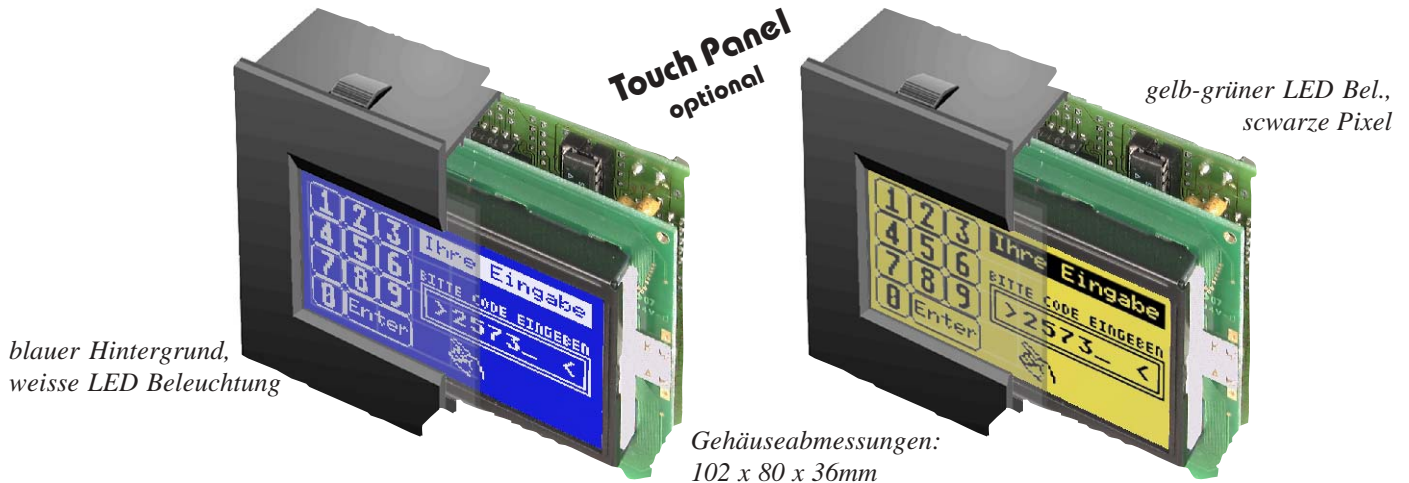


# BEDIENEINHEIT 128x64 MIT ZEICHENSÄTZEN UND GRAFIKBEFEHLEN



## TECHNISCHE DATEN

- \* 128x64 PIXEL, SCHWARZE PIXEL MIT LED-BELEUCHTUNG (GELB-GRÜN)
- \* oder WEIÑE PIXEL (WEIÑE LED) MIT BLAUEM HINTERGRUND
- \* IM SCHWARZEN SNAP-IN GEHÄUSE MIT ENTSPIEGELTER SCHEIBE
- \* TEXTMODE MIT SCROLLFUNKTION UND CR/LF AUSWERTUNG
- \* 10 FONTS VON 3,1mm (8x32 ZEICHEN) ÜBER 4,2mm (8x21) BIS 16,6mm (2x8)
- \* GERADEN ZEICHNEN, BEREICHE LÖSCHEN UND INVERTIEREN
- \* ANSCHLUSS FÜR 8x8 MATRIXTASTATUR (NUR EA KIT128-V24)
- \* VERSORGUNGSSPANNUNG: +5V / typ. 400mA (BLAU 160MA)
- \* BAUDRATEN 1200,2400,4800,9600, 19200BD
- \* VERSION -GXT: ANZEIGE VON BIS ZU 256 FESTTEXTEN ODER FESTGRAFIKEN AUS DEM FLASH
- \* VERSION -GXT: AUSWAHL ÜBER 8 LEITUNGEN (8 SCHRAUBKLEMMEN)

## BESTELLBEZEICHNUNG

FÜR V.24/RS-232 MIT INTEGR. TOUCH PANEL 8x4	EA KIT128-V24TP
WIE OBEN, MIT WEISSER LED, BLAUEM HINTERGRUND	EA KIT128-V24TPB
GRAPHIC UNIT FÜR V.24/RS-232 MIT TASTATURANSCHLUß	EA KIT128-V24
WIE OBEN, MIT WEIÑER LED, BLAUEM HINTERGRUND	EA KIT128-V24B
TEXT-/GRAFIKANZEIGE FÜR 256 TEXTE / GRAFIKEN	EA KIT128-GXT
WIE OBEN, MIT WEIÑER LED, BLAUEM HINTERGRUND	EA KIT128-GXTB
24V VERSORGUNG (+9..+35V) z.B. BEI SPS ANWENDUNG	EA OPT-9/35V
RS-422 INTERFACE STATT RS-232	EA OPT-RS4224
KABEL MIT SUB-D STECKER ZUM ANSCHLUß AN PC	EA KV24-9B
HILFSPROGRAMME (PC) FÜR MAKROS UND -GXT	EA DISK9719

WEGWEISER	gültig für Version	Seite
Bestellbezeichnungen .....	alle .....	1
Versionen / Unterschiede .....	alle .....	3
Terminal, Befehle .....	alle .....	4,5
Makros .....	alle .....	5
Zeichensätze, selbstdefinierte Zeichen .....	alle .....	6
Anschluß an RS-232C / V.24 .....	-V24, -V24TP .....	8
Tastaturanschluß 8x8 .....	-V24 .....	9
Touch Panel 8x4 .....	-V24TP .....	9
Festtextversion .....	-GXT .....	10
Compiler für Makros + -GXT .....	alle .....	11
Zusätzliche Ausgänge .....	alle .....	11
Abmessungen .....	alle .....	12
Lötbrücken / Reset .....	alle .....	12
Website .....	<a href="http://www.lcd-module.de/deu/touch/touch.htm">http://www.lcd-module.de/deu/touch/touch.htm</a>	

## EINFÜHRUNG

Die Displaybaugruppe EA KIT128 bildet eine kompakte und extrem leicht zu programmierende Anzeigeeinheit. Es können auf dem Display gleichzeitig mit komfortabelsten Befehlen verschiedene Schriften in unterschiedlichen Größen dargestellt werden. Grafikelemente wie Firmenlogos, Icons und Symbole können schnell in Grafikprogrammen wie z.B. Windows Paint erstellt und angezeigt werden. Auch Meßkurven, Diagramme und Tendenzanzeigen sind mit den vielfältigen Grafikbefehlen schnell zu erstellen.

Hervorzuheben ist auch der standardmäßig enthaltene kyrillische Zeichensatz in 2 Größen, sowie die Anschlußmöglichkeit einer Matrixtastatur bzw. eines Touchpanels (den Tastatureingang besitzen nur die Versionen -V24 und -V24B).

## INBETRIEBNAHME, SCHNELLSTART

Bitte beachten Sie vor dem Anschluß des Moduls die auf der Rückseite des Datenblattes befindlichen Hinweise zum Betrieb und zur Handhabung!

Als Versorgungsspannung für das Modul werden +5V / max. 410mA benötigt. Diese wird über 2 Schraubklemmen eingespeist. Nach dem Anlegen der Versorgungsspannung blinkt links oben der Cursor. Der Kontrast kann mit einem Schraubendreher über das eingebaute Potentiometer nachgestellt werden.

Nun wird das Display mit dem Hauptrechner verbunden (Anschluß siehe Seite 7). Für die Verbindung des Displays EA KIT128 liefert ELECTRONIC ASSEMBLY als Zubehör ein fertig konfektioniertes Kabel EA KV24-9B, welches direkt an den 9-pol. SUB-D Stecker (z.B. COM1) eines PCs angesteckt wird. Das Kabel wird am Display so eingesteckt, daß es zur Innenseite zeigt.

Bei korrekter Verbindung lassen sich sofort Texte auf dem Display anzeigen (z.B. mit Hilfe des unter der Bestellbezeichnung EA DISK9719 lieferbaren PC-Programms TERMINAL.EXE. Wenn der DOS Befehl COPY xxxxx COM1 verwendet wird, sollte zuvor die Schnittstelle mit MODE COM1:12,N,8,2 initialisiert werden). Cursorsteuerungen wie z.B. CR-LF können verwendet werden. Das Umschalten auf die verschiedenen Schriftgrößen oder die Verwendung von Grafikbefehlen ist auf den Seiten 6 und 7 beschrieben.

### ÜBERSICHT, UNTERSCHIEDE

#### **EA KIT128-V24TP mit RS-232C Schnittstelle und Touchpanel**

Diese Grafikeinheit wird mit einer durchsichtigen Touch Tastatur mit 32 Feldern geliefert. Durch Berühren des Displays können somit Daten ausgegeben oder Aktionen ausgelöst werden. Der eingebaute Summer quittiert die Tastendrucke. Über das Kabel EA KV24-9B ist die Grafikeinheit direkt an eine serielle Schnittstelle eines PC's anschließbar. Die Baudrate ist einstellbar (1200, 2400, 4800, 9600, 19200).

#### **EA KIT128-V24 mit RS-232C Schnittstelle und Tastaturabfrage**

Diese Grafikeinheit ist über das Kabel EA KV24-9B direkt an eine serielle Schnittstelle eines PC's anschließbar. Die Baudrate ist einstellbar (1200, 2400, 4800, 9600, 19200). Eine Tastatur (max. 8x8 Matrix) kann angeschlossen werden.

#### **EA KIT128-GXT Text- und Grafikkontroller mit 8 Eingängen und RS-232C**

Als Festtext- bzw. Festbildspeicher können bis zu 256 Texte/Bilder über 8 Leitungen abgerufen werden. Die Texte/Bilder werden im FLASH gespeichert. Die Leitungen sind High-aktiv und besitzen interne Pull-Down's (100 k $\Omega$ ), Vorwiderstände und Z-Dioden. Es ist somit möglich, die Eingänge über potentialfreie Schalter zu bedienen. Aber auch der direkte Anschluß an +5V oder +24V (SPS) Pegel ist erlaubt. Acht zusätzliche Ausgänge können für Steuerungszwecke genutzt werden. Für das EA KIT128-GXT wird die Diskette EA DISK9719 benötigt.

### EA DISK9719, HILFSPROGRAMME FÜR DOS

Unter der Bestellbezeichnung EA DISK9719 ist eine Diskette für PC's lieferbar, welche Hilfsprogramme zum Entwickeln enthält. Geliefert werden ein Terminalprogramm für DOS, ein Fonteditor mit Beispiel-Zeichensätzen, sowie ein Konvertierprogramm zum Umwandeln der Fonts in das Zeichensatzformat für das EA KIT128. Außerdem befindet sich auch ein Konvertierprogramm zum Umwandeln von Windows Bitmap-Grafiken (\*.BMP) in das Bildformat für das EA KIT128 auf der Diskette. Der Compiler zum Erstellen der Makros und Scripts, bzw. der Festtexte und Festgrafiken befindet sich ebenfalls auf der Diskette.

### TOUCH PANEL 8x4

Die Versionen EA KIT128-V24TP und -422TP werden mit einer durchsichtigen Touch Tastatur mit 32 Feldern geliefert. Durch Berühren des Displays können somit Daten ausgegeben oder Aktionen ausgelöst werden. Der eingebaute Summer quittiert die Tastendrucke.

Eine Kombination von EA KIT128-GXT mit dem Touchpanel ist nicht möglich. Bitte prüfen Sie in diesem Fall den Einsatz von EA KIT160-6LEDTP in gleicher Baugröße.

### EA OPT-9/35V

Alle Versionen mit der Option EA OPT-9/35V können mit einer unstabilierten Versorgung zwischen 9V und 35V betrieben werden. Beachten Sie hierbei bitte, daß die Spannung an allen Eingängen des Displays 5V nicht überschreiten darf (Ausnahme: EA KIT128-GXT (4-50V Eingänge) und alle RS-232C Pegel mit  $\pm 12V$  Pegeln).

## TEXT / GRAFIK / BEFEHLSVORRAT

Nach dem Einschalten blinkt der Cursor in der ersten Zeile und das Display ist empfangsbereit. Alle ankommenden Zeichen werden als ASCII's dargestellt (Ausnahme: CR, LF, FF, ESC). Der Zeilenvorschub erfolgt automatisch oder durch das Zeichen 'LF'. Ist das Display voll, scrollt der Bildschirm nach oben. Beim Zeichen 'FF' (Seitenvorschub) wird das Display gelöscht.

Alle zusätzlichen Befehle (Cursor positionieren, Font einstellen, Grafikbefehle, ...) werden über das ESCAPE-Zeichen (dez 27/hex 1B) eingeleitet. Danach folgt ein Befehlsbuchstabe (Groß- und Kleinbuchstaben sind gleichwertig) und evtl. einige Parameter. Die Übergabe der Parameter erfolgt jeweils als 8-Bit Binärwert; z.B. entsprechen die Koordinaten x=65 und y=51 den ASCII-Zeichen 'A' bzw. '3'

Befehlstabelle (alle Versionen)											
Befehl	Codes					Anmerkung					
<b>Befehle für den Textmodus</b>											
Formfeed FF (dez:12)	^L					Bildschirm wird gelöscht und der Cursor nach Pos. (1,1)					
Carriage Return CR (13)	^M					Cursor ganz nach links					
Linefeed LF (dez:10)	^J					Cursor 1 Zeile tiefer mit evtl. scrollen					
Cursor On / Off	ESC	C	n1			n1=0: Cursor ist unsichtbar; n1=1: Cursor blinkt					
Cursor positionieren	ESC	O	n1	n2		n1=Spalte; n2=Zeile; Ursprung links oben ist (1,1)					
Inverse Schrift	ESC	J	n1			n1=0: normale Schrift; n1=1: Inverse Schrift					
Autom. Zeilenumbruch	ESC	W	n1			n1=0: Automatischer Zeilenumbruch AUS; n1=1: EIN					
<b>Grafikbefehle</b>											
Zeichen-Modus	ESC	V	n1			n1: 1=setzen; 2=löschen; 3=invers; 4=Replace; 5=Invers Replace;					
ASCII-Zeichen setzen	ESC	A	x1	y1	n1	Das Zeichen n1 wird an Koordinate gesetzt.					
Zeichenkette ausgeben	ESC	Z	x1	y1	...	NUL	Die Zeichenkette (...) ausgeben; Zeichen 'NUL' (Binär 0) = Ende				
Punkt setzen	ESC	P	x1	y1		Koordinaten (0,0) bis (127,63)					
Gerade zeichnen	ESC	G	x1	y1	x2	y2	Anfangs- und Endkoordinaten (0,0) bis (127,63)				
Gerade weiter zeichnen	ESC	T	x1	y1		Eine Gerade vom letzten Endpunkt bis x1, y1 zeichnen					
Rechteck zeichnen	ESC	R	x1	y1	x2	y2	Gegenüberliegende Eckpunkte (0,0) bis (127,63)				
Bereich löschen	ESC	L	x1	y1	x2	y2	Gegenüberliegende Eckpunkte (0,0) bis (127,63)				
Bereich invertieren	ESC	I	x1	y1	x2	y2	Gegenüberliegende Eckpunkte (0,0) bis (127,63)				
Bereich füllen	ESC	S	x1	y1	x2	y2	Gegenüberliegende Eckpunkte (0,0) bis (127,63)				
Bargraph definieren	ESC	-	L R O U	Nr	x1	y1	x2	y2	AW	EW	Einen Bar nach L(inks),R(echts),O(ben),U(nten) mit der 'Nr' (1..8) definieren. x1,y1,x2,y2 sind das Umschliessende Rechteck des Bargraphs. AW, EW sind die Werte für 0% und 100%.
Bargraph zeichnen	ESC	-	Nr	Wert		Den Bargraph mit der 'Nr' (1..8) auf den neuen Benutzer-'Wert' setzen					
Bildbereich laden	ESC	B	x1	y1	Daten	Einen Bildbereich nach x1,y1 laden; Daten des Bildes siehe Bildaufbau					
<b>Sonstige Steuerbefehle</b>											
Font einstellen	ESC	F	n1			n1=1: Font Nr. n1 (1..8) einstellen					
Breitschrift	ESC	E	n1			n1=0: normale Schrift; n1=1: breite Schrift					
Zeichen definieren	ESC	U	n1	Daten		n1=Zeichen Nr.; Daten=Anzahl Bytes je nach eingest. Font					
Display Befehle	ESC	D	n1			n1=0:Aus;n1=1:Ein;n1=2:Lösch.;n1=3:Invert.;n1=10..20:LED-Helligkeit					
Beleuchtung Ein / Aus	ESC	H	n1			n1=0: Bel. Aus; n1=1: Bel. Ein; n1=2..255: für n1 sek. lang Ein					
Selekt Kontroller	ESC	K	S	n1		Kontroller mit Adresse n1 (n1=0..254; n1=255: alle) aktivieren					
Deselekt Kontroller	ESC	K	D	n1		Kontroller mit Adresse n1 (n1=0..254; n1=255: alle) deaktivieren					
Selekt/Desel. Adresse	ESC	K	A	n1		Neue Adresse n1 für Kontroller definieren (z.B im Power-On Makro)					
Ausgabeport einstellen	ESC	Y	n1	n2		n1=0: Alle 8 Ausgänge entsprechend n2 (=8-Bit Binärwert) einstellen n1=1..8: Ausg. n1 rücksetzen (n2=0); setzen (n2=1); invertieren (n2=2)					
Warten (Pause)	ESC	!	n1			n1 Zehntel-Sekunden abwarten bevor die nächste Ausgabe stattfindet					
Status Check	ESC	?	C	n1		n1=0:Meldung auf RS232/422: 'O'=OK; 'E' od. keine Reaktion=ERROR n1=1..8: OUT1..8 wird invertiert=OK; keine Reaktion=ERROR					
<b>Befehle für Makros</b>											
Script ausführen	ESC	Q	n1			Script n1 abarbeiten (max. 2 Ebenen)					
Makro ausführen	ESC	N	n1			Makro n1 abarbeiten (max. 2 Ebenen)					
Makros autom. ausführ.	ESC	X	n1	n2	n3	Makros von n1 bis n2 zyklisch abarbeiten (n3=Pause in 1/10 s)					

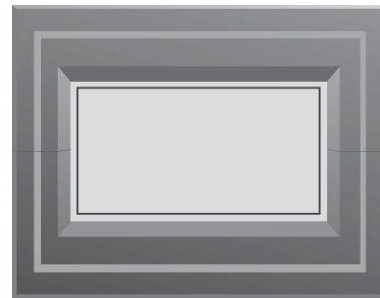
zusätzliche Befehle für -V24							
Matrixtastatur abfragen	ESC	M	n1				n1=0: Automatische Abfrage AUS n1=1: Automatische Abfrage Gesamtzustand EIN n1=2: Automatische Abfrage Einzeltastenmodus EIN n1=3: Aktuellen Zustand der Matrixtastatur senden
Matrixtastatur ausmaskieren	ESC	M	4	n1			Um mit dem Befehl 'ESC Y' Ausgänge benutzen zu können, dürfen diese nicht mehr für die Matrixtastaturabfrage benutzt werden. Eine '0' in den 8-Bits des Parameter n1 deaktiviert den jeweiligen Ausgang.
Bild senden	ESC	>					Es wird der gesamte Bildinhalt gesendet. Zur Kennzeichnung wird zuerst ein 'B' gesendet. Danach folgen die Breite und Höhe in Pixel und die eigentlichen Bilddaten
Bytes senden	ESC	=	n1	...			es werden n1 nachfolgende Bytes auf der V.24 gesendet (z.B zum Ansteuern einer externen seriellen Druckers mit RS232-C Schnittstelle)
Ausgabeport einlesen	ESC	?	Y	n1			n1=0: Alle 8 Ausgänge OUT1..8 als 8-Bit Binärwert einlesen n1=1..8: Ausgang OUT<n1> einlesen (1=high=5V, 0=low=0V)
Befehle zur Unterstützung des Touch-Panels (EA TOUCH8x4-A)							
Touch-Taste definieren	ESC	*	f1	f2	Ret. Code	Form	Anz. Text
Touch-Tasten Reaktion	ESC	*	A B C D				
Touch-Taste Invertieren	ESC	*	I		Ret. Code		
Touch-Tasten Reset	ESC	*	R				
Summer Ein / Aus	ESC	.	n1				

zusätzliche Befehle für -GXT							
Bild senden	ESC	>					Es wird der gesamte Bildinhalt gesendet. Zur Kennzeichnung wird zuerst ein 'B' gesendet. Danach folgen die Breite und Höhe in Pixel und die eigentlichen Bilddaten
Bytes senden	ESC	=	n1	...			es werden n1 nachfolgende Bytes auf der V.24 gesendet (z.B zum Ansteuern einer externen seriellen Druckers mit RS232-C Schnittstelle)
Ausgabeport einlesen	ESC	?	Y	n1			n1=0: Alle 8 Ausgänge OUT1..8 als 8-Bit Binärwert einlesen n1=1..8: Ausgang OUT<n1> einlesen (1=high=5V, 0=low=0V)

### BEISPIEL ZUR BEFEHLSANWENDUNG

Im folgenden Beispiel wird für verschiedene Programmiersprachen ein Rechteck auf dem Display gezeichnet.

Beispiel	Auszugebende Codes					
für Compiler	#R 0, 0, 127, 63					
in Hex	\$1B	\$52	\$00	\$00	\$7F	\$3F
in Dezimal	27	82	0	0	127	63
für Turbo-Pascal	write(aux, chr(27), 'R', chr(0), chr(0), chr(127), chr(63));					
für 'C'	fprintf(stdaux, "%c%c%c%c%c%c", 27, 'R', 0, 0, 127, 63);					
für Q-Basic	OPEN "COM1:1200,N,8,2,BIN" FOR RANDOM AS #1 PRINT #1,CHR\$(27)+"R"+CHR\$(0)+CHR\$(0)+CHR\$(127)+CHR\$(63)					



### MAKROS

Mehrere Befehle können zu einem Makro zusammengefasst werden und über den Befehl *Makro ausführen* gestartet werden. Die Makros werden mit Hilfe eines Compilers erstellt (Diskette EA DISK9719 erforderlich) und in das FLASH des EA KIT128 programmiert.

Das sog. Power-On-Makro wird z.B. sofort nach dem Anlegen der Versorgungsspannung automatisch ausgeführt. Hier können z.B. Befehle stehen wie *Cursor aus* (ESC C 0), *Font 3 einstellen* (ESC F 3), *Baudrate 9600 Baud* (ESC + 3'). Eine kurze Beschreibung dazu befindet sich auf der Seite 11.

## ZEICHENSÄTZE

In der Displayeinheit EA KIT128 sind 10 Zeichensätze vorhanden. Alle Zeichensätze werden im Textmode und im Grafikmode unterstützt. Im Grafikmode (Befehle 'ESC A' und 'ESC Z') können die Zeichen pixelweise positioniert werden, dabei bezieht sich die Koordinatenangabe auf die linke obere Ecke.

Nr.	Zeichenhöhe	Zeilen x Zeichen	Größe in Pixel	ASCII-Bereich	Frei def. ASCII-Code(s)	Bemerkung
1	4,2 mm	8 x 21	6 x 8	32 - 255	0..9	erweiterter ASCII-Code
2	4,2 mm	8 x 16	8 x 8	32 - 255	0..7	erweiterter ASCII-Code
3	8,3 mm	4 x 16	8 x 16	32 - 255	0..3	erweiterter ASCII-Code
4	3,1 mm	8 x 32	4 x 6	32 - 158	0..15	Minischrift
5	3,1 mm	8 x 25	5 x 6	32 - 158	0..11	Minischrift
6	16,6 mm	2 x 8	16 x 32	42 - 58	0	Große Ziffern '0'-'9', '*', '-./'
7	16,6 mm	2 x 8	16 x 32	32 - 127	0	ASCII-Code
8	25 mm	1 x 5	24 x 48	45 - 58	-	Sehr große Ziffern '0'-'9', '-./'
9	4,2 mm	8 x 21	6 x 8	64 - 128	0..9	Kyrillische Zeichen
10	8,3 mm	4 x 16	8 x 16	64 - 128	0..3	Kyrillische Zeichen



## SELBSTDEFINIERTER ZEICHEN

Es ist möglich, für einen Zeichensatz bis zu 16 Zeichen selbst zu definieren (je nach Zeichengröße). Diese Zeichen haben dann die ASCII Codes 0..15 und bleiben bis zum Abschalten der Versorgungsspannung in einem 64 Byte großem internen RAM-Bereich erhalten. **Achtung:** Werden eigene Zeichen definiert, so darf der Bargraph-Befehl nicht mehr ausgeführt werden.

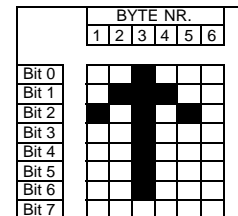
*Beispiel 1:*

Mit dem Befehl

**ESC U \$03**

**\$04 \$02 \$7F \$02 \$04 \$00**

wird für ASCII-Nr. 3, bei eingestelltem 6x8 Zeichensatz, ein Pfeil nach oben definiert.



*Beispiel 2:*

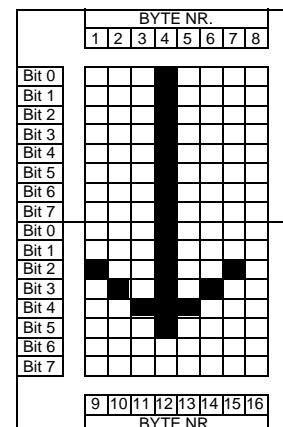
Mit dem Befehl

**ESC U \$02**

**\$00 \$00 \$00 \$FF \$00 \$00 \$00 \$00**

**\$04 \$08 \$10 \$3F \$10 \$08 \$04 \$00**

wird für ASCII-Nr. 2, bei eingestelltem 8x16 Zeichensatz, ein Pfeil nach unten definiert.



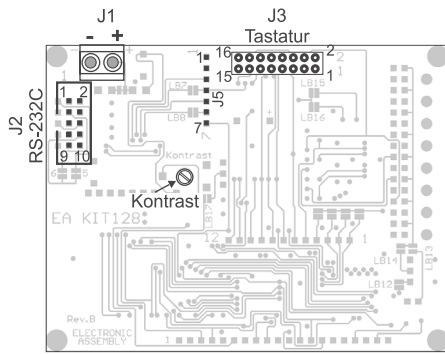
+ Lower	\$0	\$1	\$2	\$3	\$4	\$5	\$6	\$7	\$8	\$9	\$A	\$B	\$C	\$D	\$E	\$F
Upper	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)
\$20 (dez: 32)		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
\$30 (dez: 48)	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
\$40 (dez: 64)	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	
\$50 (dez: 80)	p	q	r	s	t	u	v	w	x	y	z	[	\	]	^	_
\$60 (dez: 96)	`	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~
\$70 (dez: 112)	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~
\$80 (dez: 128)	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~
\$90 (dez: 144)	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~
\$A0 (dez: 160)	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~
\$B0 (dez: 176)	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~
\$C0 (dez: 192)	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~
\$D0 (dez: 208)	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~
\$E0 (dez: 224)	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~
\$F0 (dez: 240)	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~

Font 3

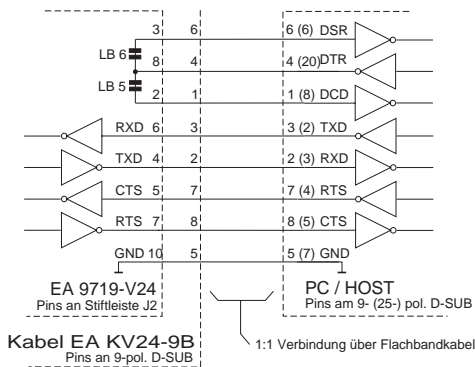
+ Lower	\$0	\$1	\$2	\$3	\$4	\$5	\$6	\$7	\$8	\$9	\$A	\$B	\$C	\$D	\$E	\$F
Upper	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)
\$20 (dez: 32)											x	+	,	-	.	÷
\$30 (dez: 48)	0	1	2	3	4	5	6	7	8	9	:					

Font 6

### V.24/RS-232C ANSCHLUSS



Ansicht von hinten



Die Displayeinheit EA KIT128 wird für den Anschluß an eine bidirektionale serielle Schnittstelle mit echten RS-232C Pegeln ( $\pm 10V$ ) ausgeliefert. Auf die 10-pol. Stiftleiste J2 kann direkt das als Zubehör erhältliche Kabel EA KV24-9B aufgesteckt werden. Dieses Kabel besitzt auf der Gegenseite einen 9-pol. D-SUB Stecker, passend für den Anschluß an die COM1 eines PC's.

Das Modul stellt die Handshakeleitungen RTS und CTS zur Verfügung. Durch einen -10V Pegel am Pin RTS zeigt das Modul an, daß solange keine weiteren Daten mehr gesendet werden dürfen, bis RTS wieder +10V Pegel zeigt. Andererseits unterdrückt das EA KIT128 die Datenausgabe falls am Pin CTS ein -10V Pegel anliegt. In diesem Fall können jedoch evtl. Tastaturbetätigungen verloren gehen.

Die Datenübertragungsparameter sind werkseitig auf 8 Datenbits, 1 Stopbit und 1200 Baud eingestellt. Durch Schließen der Lötbrücke 1 bis 4 kann die Baudrate auf 2400, 4800, 9600 oder 19200 Baud erhöht werden. Eine Beschreibung der Lötbrücken befindet sich auf der letzten Seite.

**Achtung:** Es darf immer nur max. 1 Lötbrücke geschlossen sein. Niemals dürfen 2 Lötbrücken zugleich gesetzt werden!

An der 7-poligen Lötleiste J5 können die CMOS Pegel der seriellen Schnittstelle benutzt werden, um sie z.B. direkt an ein Mikroprozessor anzuschließen. **Achtung:** Der RS-232 Pegelwandler SP232 (oder ähnlich) muß dann durch Öffnen der Lötbrücken LB7 und LB8 deaktiviert werden! Der Stromverbrauch beträgt typ. 45mA, und mit eingeschalteter LED-Beleuchtung typ. 300mA (bzw. 160mA die blaue Version)

RS-232C Anschluß J2			
Pin	Symbol	In/Out	Funktion
1	VDD	-	+ 5V Versorgung
2	DCD	-	über LB5 nach DTR
3	DSR	-	über LB6 nach DTR
4	TxD	Out	Transmit Data
5	CTS	In	Clear To Send
6	RxD	In	Receive Data
7	RTS	Out	Request To Send
8	DTR	-	siehe Pin 2, Pin 3
9	NC	-	frei
10	GND	-	0V Masse

CMOS-Pegel Anschluß J5		
Pin	Symbol	Funktion
1	VDD	+ 5V Versorgung
2	GND	0V Masse
3	TxD	Transmit Data
4	RxD	Receive Data
5	RTS	Request To Send
6	CTS	Clear To Send
7	RESET	H: Reset

## TASTATURBETRIEB (NUR EA KIT128-V24)

Am Steckanschluss J3 kann eine Tastatur (einzelne Tasten bis zur 8x8 Matrix-Tastatur) angeschlossen werden. Die angeschlossenen Tasten werden dabei per Software entprellt. Bitte beachten Sie, daß die Tastaturfunktion nur von der V.24 Version ohne Touchpanel unterstützt wird.

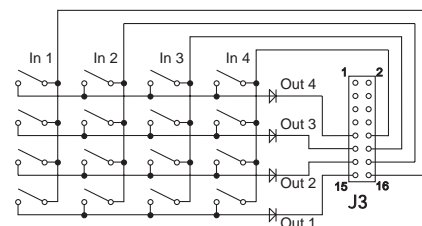
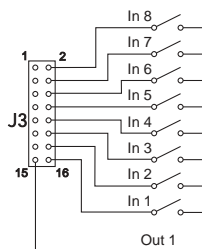
Die Taste wird zwischen einem Ausgang und einem Eingang geschaltet. Jeder Eingang ist mit einem ca. 100kΩ Pullup abgeschlossen. An jeden Ausgang können bis zu 8 Tasten angeschlossen werden.

Um Doppeltastendrücke zu erkennen, müssen die Ausgänge voneinander entkoppelt werden. Dies geht am besten mit Schottky-Dioden (z.B. BAT 43). Bei Mehrfach-Tastendrücken (>2) muß jede Taste mit einer eigenen Diode entkoppelt werden.

Matrix - Tastaturanschluß J3					
Pin	Symbol	Funktion	Pin	Symbol	Funktion
1	OUT 8	Ausgang Zeile 8	2	IN 8	Eingang Spalte 8
3	OUT 7	Ausgang Zeile 7	4	IN 7	Eingang Spalte 7
5	OUT 6	Ausgang Zeile 6	6	IN 6	Eingang Spalte 6
7	OUT 5	Ausgang Zeile 5	8	IN 5	Eingang Spalte 5
9	OUT 4	Ausgang Zeile 4	10	IN 4	Eingang Spalte 4
11	OUT 3	Ausgang Zeile 3	12	IN 3	Eingang Spalte 3
13	OUT 2	Ausgang Zeile 2	14	IN 2	Eingang Spalte 2
15	OUT 1	Ausgang Zeile 1	16	IN 1	Eingang Spalte 1

### Senden der Tastendrücke

**Modus 0:** Mit dem Befehl "ESC M 0" wird die automatische Tastaturabfrage ausgeschaltet. Die Tastendrücke werden nicht mehr automatisch gesendet. Die Tasten können jedoch per Befehl "ESC M 3" abgefragt werden. Nach diesem Befehl wird der aktuelle Zustand der Tastatur wie im Modus 2 gesendet.



**Modus 1:** Mit dem Befehl "ESC M 1" wird die automatische Abfrage für den Gesamtzustand der Tasten aktiviert. Bei jeder Änderung, Drücken und Loslassen einer Taste, werden alle noch gedrückten Tasten gesendet. Die Tasten werden wie folgt übertragen: Der Kennbuchstabe "T" (\$54), gefolgt von den Tastennummern (1..64) aller gedrückten Tasten. Als Endekennung folgt das Zeichen "NUL" (\$00).

**Modus 2:** Mit dem Befehl "ESC M 2" wird die automatische Abfrage für Einzeltasten aktiviert. Bei jedem Druck einer Taste wird die dazugehörige Tastennummer (1..64) gesendet. Das Loslassen der Taste wird nicht gesendet.

Die Tastennummer kann folgendermaßen bestimmt werden: **Tastenummer = (Ausgang - 1) \* 8 + Eingang** (Ausgang, Eingang: eine Zahl zwischen 1 und 8)

**Achtung !** Falls die Handshakleitung (z.B. CTS) das Senden nicht erlaubt, können die Tastendrücke verloren gehen.



## ELECTRONIC ASSEMBLY

### TOUCH-PANEL 8x4 (NUR EA KIT128-V24TP)

Für die Version EA KIT128-V24TP ist anstelle einer externen 8x8 Tastatur, eine durchsichtige Touch-Tastatur mit 32 Feldern eingebaut. Der Controller unterstützt dieses Touchpanel mit komfortablen Befehlen. So können mehrere Touch-Felder zu einer großen Gesamt-Taste zusammengefasst, die Taste gezeichnet und einText(e) in der Taste zentriert werden. Ebenso kann dieser Gesamt-Taste ein Return-Code (1..255) zugewiesen werden. Wird der Return-Code 0 zugewiesen so ist die Taste deaktiviert und wird bei Betätigung nicht gemeldet.

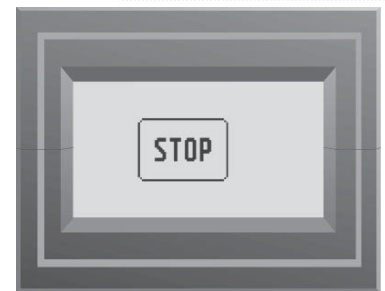
Beim Berühren der Touch-Tasten können diese automatisch invertiert werden und/oder ein Summer die Berührung signalisieren (an OUT2 (Pin 13, J3) liegt für  $\frac{2}{10}$  Sek. ein L-Pegel an). Gleichzeitig wird der definierte Return-Code der Taste über die serielle Schnittstelle gesendet oder es kann ein internes SCRIPT mit der Nummer des Return-Codes abgearbeitet werden (dazu muß als Dummy-Befehl das Kommando 'ESC :' am Anfang dieses Scripts stehen).

#### Beispiel:

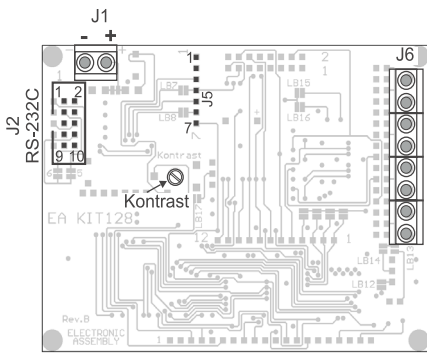
Definieren einer Taste von Feld 11 bis 21, mit dem Return-Code 65='A' und dem Text "STOP"

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32

Beispiel	Auszugebende Codes										Bemerkung		
für Compiler	#* 11, 21, 'A', 2, "STOP"										Die Anzahl der Texte wird hier nicht angegeben !		
als ASCII	ESC	*	.	.	A	.	.	S	T	O	P	.	die Punkte '.' stehen für nicht darzustellende ASCII-Zeiche
in Hex	\$1B	\$2A	\$0B	\$15	\$41	\$02	\$01	\$53	\$54	\$4F	\$50	\$00	
in Dezimal	27	42	11	21	65	2	1	83	84	79	80	0	
Befehlskennung	Einleitung Touch-Befehl	linke oberes Touchfeld	rechtes untere Touchfeld	Return Code	Taste zeichnen mit Rahmen ein 1-zeiliger Text							Text Ende Kennung	



## FESTTEXT-/GRAFIKVERSION EA KIT128-GXT



Ansicht von hinten

Die Displayeinheit EA KIT128-GXT wird über 8 digitale Eingänge gesteuert. Jede Änderung an einem der Eingänge startet ein ins FLASH programmiertes Script. Die Eingänge sind über 8 Schraubklemmen zu erreichen. Hier können sowohl einfache Schalter (Schließer) angeschlossen werden, als auch Spannungen zwischen 0 und +50V. Für die Erstellung der Scripts benötigt man einen PC und die Diskette EA DISK9719.

Auswahl J6		
Pin	Symbol	Funktion
D0	LSB	Bit 0
D1		Bit 1
D2		Bit 2
D3		Bit 3
D4		Bit 4
D5		Bit 5
D6		Bit 6
D7	MSB	Bit 7

### 1.) Anschluß an 8 Schalter

Der jeweilige Schalter wird zwischen dem Pluspol der Stromversorgung und dem Eingang angeschlossen. Ein offener Eingang bzw. offener Schalter wird als "0" gewertet: Ist kein Schalter angeschlossen, bzw.

sind alle Schalter offen, so wird das Script Nr. 0 ausgeführt. Sind alle 8 Schalter geschlossen ("1"), so wird das Script Nr. 255 ausgeführt.

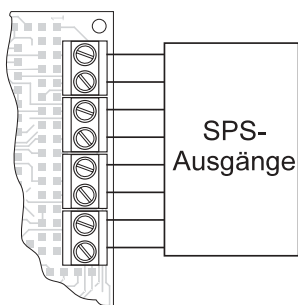
### 2.) Anschluß an SPS oder CMOS-Pegel

Die 8 Eingänge werden direkt mit dem jeweiligen Spannungsausgang verbunden. Eine gemeinsame Masse (Minuspol) ist erforderlich. Liegt keine Spannung an, wird das Script Nr. 0 ausgeführt, liegt eine Spannung mit mindestens 4V an, wird das Script Nr. 255 ausgeführt.

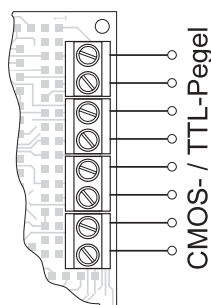
Alle Befehle und Makros können auch über die serielle Schnittstelle RS-232C aufgerufen werden.

## APPLIKATIONSBEISPIELE

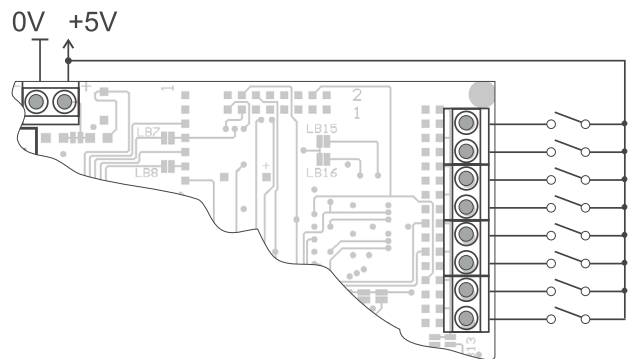
Die 3 häufigsten Anschlußvarianten für SPS-Ansteuerung, für die direkte Anbindung an Logikpegel und für den Anschluß von einfachen Schließern sind unten abgebildet.



Anschluß an SPS



Ansteuerung über CMOS-Pegel



Anschluß an 8 Schalter

## ERSTELLEN DER MAKROS

Einzelne oder mehrere Befehlsfolgen können als sog. Makros oder Scripts zusammengefasst und im FLASH fest abgespeichert werden. Diese können dann mit den Befehlen *Makro ausführen* / *Script ausführen* gestartet werden. Bei der Version EA KIT128-GXT können die Scripts auch über die 8 Eingänge aufgerufen und auf dem LC-Display angezeigt werden. Um nun Ihre speziellen Makros erstellen zu können, benötigen Sie folgende Hilfsmittel:

- die Diskette EA DISK9719; sie enthält einen Compiler, Beispiele und Fonts für DOS
- einen PC mit serieller Schnittstelle COM1, möglichst 386/486 mit ca. 1MB Platz auf der Festplatte

## ERSTELLEN DER FLASH-DATEI MIT INDIVIDUELLEN MAKROS

Um eine Befehlsfolge als Script zu programmieren, werden alle Befehle auf dem PC in eine Datei geschrieben. Mit der Dateiendung (Extension) wird die Scriptnummer bestimmt (\*.000 bis \*.255).

# ELECTRONIC ASSEMBLY

Sind alle benötigten Scriptdateien erstellt, startet man das Programm COMPILER.EXE. Dieses erzeugt eine EPROM-Datei \*.EPR, welche dann in das FLASH auf dem Display gebrannt wird.

Während der Erstellung der Makrodateien lassen sich einzelne Scripts durch Anschluß des EA KIT128 an die serielle Schnittstelle des PCs testen, ohne immer wieder auf's neue das FLASH programmieren zu müssen. Eine ausführliche Beschreibung zur Programmierung der Makros finden Sie auf der Diskette EA DISK9719 unter dem Namen DOKU.DOC (für WORD) bzw. DOKU.TXT (DOS).

## DEFINITIONEN

### Makro

Ein Makro besteht aus einer Befehlsfolge, welche resident im FLASH gespeichert ist. Es kann über den Befehl *Makro ausführen* aufgerufen werden. Es ist für immer wiederkehrende Befehlsfolgen gedacht; z.B. Display löschen, einen Rahmen zeichnen und das Firmenlogo darstellen. In einem Makro darf wiederum ein Makro aufgerufen werden. Dieses Makro darf dann kein weiteres Makro mehr starten.

Es können bis zu 100 Makros angelegt werden (Dateiendungen \*.N00 bis \*.N99). Die Makronummerierung darf auch Lücken aufweisen.

Das Makro \*.N00 hat eine Sonderstellung: Es wird automatisch beim Anlegen der Versorgungsspannung ausgeführt, kann aber auch per Befehl aufgerufen werden (Power-On Makro).

**Achtung:** Wird im Power-On-Makro eine Endlosschleife programmiert, ist das Display nicht mehr ansprechbar. Die Ausführung des Power-ON Makros und des eigestellten Scriptmakros (-GXT) kann verhindert werden, wenn beim Einschalten/Reset der Pin 2 (GND) mit Pin 5 (RTS-CMOS) an J5 gebrückt sind (ab Version V1.5). Jetzt müssen die Fonts und Makros wieder neu eingespielt werden.

### Script

Ein Script ist wie ein Makro aufgebaut. Es kann jedoch bei der Version EA KIT128-GXT zusätzlich über externe Digitalpegel gestartet werden. Nach jeder Änderung an den Eingängen E0..7 wird das zugehörige Script ausgeführt. Ebenso kann es im Gegensatz zu Makros bei den Versionen mit EA TOUCH8X4 über das Touchpanel gestartet werden. Es können bis zu 256 Scripts angelegt werden (\*.000 bis \*.255). Die Scriptnummerierung darf auch Lücken aufweisen.

### Projektname

Ein Projektname darf aus bis zu 8 Zeichen (DOS-üblich) bestehen. Wenn ein Projekt z.B. "DEMO" genannt wird, müssen die dazugehörigen Scriptdateien DEMO.000 bis DEMO.255 und alle Makros DEMO.N00 bis DEMO.N99 heißen.

Jedem Projekt muß eine eigene Initialisierungsdatei zugeordnet werden, welche je nach Projekt z.B. DEMO.INI heißt. Die vom Compiler erstellte EPROM-Datei bekommt den Namen DEMO.EPR.

## ZUSÄTZLICHE AUSGÄNGE

Die nicht genutzten Ausgänge für die Tastaturabfrage (J3, OUT1..OUT8) können in der Version EA KIT128-GXT als zusätzliche Ausgänge verwendet werden (C-MOS Pegel). Bei den Versionen EA KIT128-V24 und EA KIT128-V24TP können die zusätzlichen Ausgänge erst dann benutzt werden, wenn einzelne Ausgänge von der Tastaturabfrage ausgenommen wurden (Befehl "ESC M 4 n1").

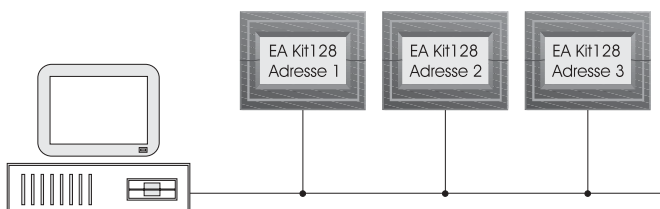
Mit dem Befehl "ESC Y" können die 8 Ausgänge geschaltet werden (siehe Befehlstabelle auf Seite 4). Jeder Ausgang kann als H-Pegel 0,25mA Strom liefern (Ausgangswiderstand typ. 20kΩ) und als L-Pegel bis zu 20mA.

**Achtung:** Die Lötbrücken LB1..LB4 zur Baudrateneinstellung liegen an den Ausgängen OUT1..OUT4. Falls eine Lötbrücke geschlossen ist (bei 2400, 4800, 9600 oder 19200 Baud) kann der entsprechende Ausgang nicht mehr verwendet werden! Es empfiehlt sich die Einteilung der Baudrate per Software (Seite 5 unten).

## ADRESSIERUNG MEHRERER EA KIT128 ÜBER EINE SCHNITTSTELLE

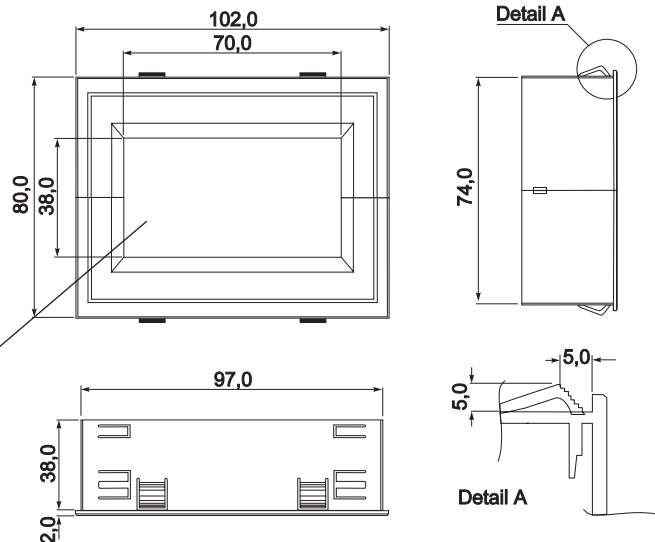
Mit dem Befehl "Selekt / Deselekt" können mehrere EA KIT128 an einer Schnittstelle adressiert betrieben werden. Die jeweilige Kontrolleradresse kann mit dem Power-On Makro und dem Befehl "ESC K A n1" eingestellt werden.

**ACHTUNG !** Die Ausgangssignale TXD, RTS dürfen nicht parallel geschaltet werden. Falls sie benötigt werden (Handshake, Tastaturabfrage), müssen diese Signale mit einer externer Logik verknüpft werden.



# EA KIT128 REV.B

## ABMESSUNGEN

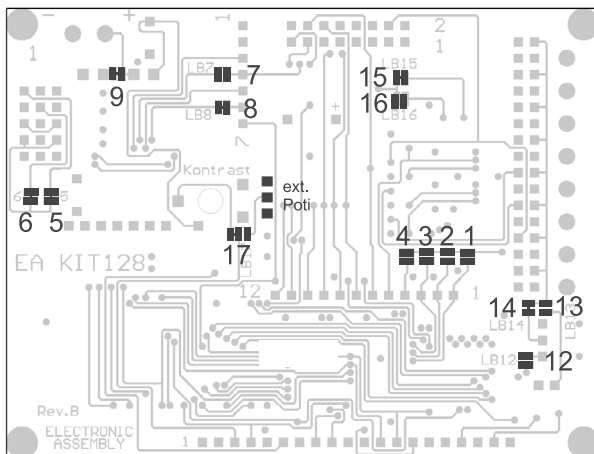


entpiegelte Scheibe oder  
Touch Panel (EA TOUCH8X4)

alle Maße in mm  
Frontplattendurchbruch 97,5x75,0 mm

## LÖTBRÜCKEN / RESET

Auf der Platinenrückseite befinden sich einige Lötbrücken zur Anpassung der EA KIT128 an bestimmte Anforderungen. Diese sind bei Auslieferung bereits richtig gesetzt und müssen nicht mehr verändert werden. Die Baudrate kann mit den Lötbrücken LB1-4 verändert werden (Voreinstellung 1200 Baud).



Lötbr.	Bezeichnung	Beschreibung	BAUD				
			LB4	LB3	LB2	LB1	
1-4	Baudrate	Hier wird die Baudrate eingestellt. Es darf max. 1 Lötbrücke gesetzt sein! Bei Auslieferung sind diese offen.	1200 2400 4800 9600 19200	auf zu	auf zu	auf zu	auf zu
5	DTR - DCD	Gewisse oft benötigte Verbindungen an der RS-232C Schnittstelle werden hier über Lötbrücken hergestellt. Bei Auslieferung sind diese geschlossen.					
6	DTR - DSR						
7	RXD	Diese Lötbrücken müssen bei Benutzung der 5V-TTL Pegel an J5 für die RS232 Schnittstelle offen sein.					
8	CTS	Bei Auslieferung sind diese geschlossen.					
9	DC / DC	Diese Lötbrücke muß bei der 5V Version geschlossen und bei der 24V Version (Option 9/35V) offen sein !					
12	Prg. Disable	Durch Öffnen der LB 12 kann ein versehentliches Umprogrammieren des FLASH verhindert werden.					
13,14	Z-Dioden	Nur EA KIT128-GXT: Die Z-Dioden werden auf 0V (LB 14 zu) oder auf +5V bezogen (LB 13 zu, Z-Dioden umdrehen !)					
15,16	Pullup/down	Das Widerstandsnetzwerk am Eingang wird auf 0V (LB 15 zu) oder auf +5V bezogen (LB 16 zu)					
17	ext. Poti	Durch umlegen dieser Lötbrücke kann das interne Poti deaktiviert werden und ein externes Poti (25k) benutzt werden.					

Um während des Betriebs erkennen zu können ob das Display noch einwandfrei funktioniert, empfiehlt es sich in kürzeren Zeitabständen mit dem Befehl 'ESC ? C n1' einen Ausgang n1 zu invertieren. Wenn der entsprechende Ausgang n1 invertiert wird, kann von einer korrekten Funktion des Displays ausgegangen werden. Reagiert der Ausgang n1 nicht, muß über einen High-Pegel (>2µs) am Anschluß "Reset" (J5 Pin 7) ein Neustart ausgelöst, und das Display neu beschrieben werden. In einigen Fällen reicht es auch einen Softwarereset durch die Befehlsfolge 'ESC & F 0 0 0 0 J' auszulösen.

## HINWEISE ZUR HANDHABUNG UND ZUM BETRIEB

- Zur elektrischen Zerstörung des Moduls kann führen: Verpolung oder Überspannung der Stromversorgung, Überspannung oder Verpolung bzw. statische Entladung an den Eingängen, Kurzschließen der Ausgänge.
- Vor der Demontage des Moduls muß unbedingt die Stromversorgung abgeschaltet sein. Ebenso müssen alle Eingänge stromlos sein.
- Das Display, der Touchscreen und die Frontscheibe bestehen aus Kunststoff und dürfen nicht mit harten Gegenständen in Berührung kommen. Die Oberflächen können mit einem weichen Tuch ohne Verwendung von Lösungsmitteln gereinigt werden.
- Das Modul ist ausschließlich für den Betrieb innerhalb von Gebäuden konzipiert. Für den Betrieb im Freien müssen zusätzliche Vorkehrungen getroffen werden. Der maximale Temperaturbereich von 0..+50°C darf nicht überschritten werden. Bei Einsatz in feuchter Umgebung kann es zu Funktionsstörungen und zum Ausfall des Moduls kommen. Das Display ist vor direkter Sonneneinstrahlung zu schützen.

