

# 5,7" LCD-BEDIENEINHEIT INKL. TOUCH PANEL

mit Port für  
Dotmatrix LCD



EA KIT320-8CTP  
Abm. 153x120mm

## TECHNISCHE DATEN

- \* 5,7" LCD GRAFIKDISPLAY MIT DIVERSEN GRAFIKFUNKTIONEN UND FONTS
- \* 320x240 PIXEL MIT CFL-BELEUCHTUNG BLAUNEGATIV (EMPFOHLEN)
- \* 320x240 PIXEL MIT LED-BELEUCHTUNG WEISS, SCHWARZE SCHRIFT
- \* FONT ZOOM VON ca. 2mm BIS ZU ca. 80mm, in 90° SCHRITTEN DREHBAR
- \* VERSORGUNG +5V±2% @ 500mA (CFL) / 400mA (LED) ODER +9..35V OPTIONAL
- \* RS-232 ODER OPTIONAL RS-422 MIT BAUDRATEN 2.400..115.200 BD
- \* **PIXELGENAUE** POSITIONIERUNG BEI ALLEN FUNKTIONEN
- \* GERADE, PUNKT, BEREICH, UND/ODER/EXOR, BARGRAPH...
- \* CLIPBOARD FUNKTIONEN, PULL-DOWN MENÜS
- \* BIS ZU 256 BILDER INTERN SPEICHERBAR
- \* BIS ZU 1024 MAKROS PROGRAMMIERBAR (FLASH MIT 480kB)
- \* TEXT UND GRAFIK MISCHEN, BLINKATTRIBUTE, INVERS BLINKEN
- \* BELEUCHTUNG PER SOFTWARE SCHALTBAR (AUS, EIN, HALBE HELLGHEIT)
- \* ANALOGES TOUCH PANEL: VARIABLES RASTER MIT z.B. 10x8 FELDERN
- \* FREI DEFINIERBARE TASTEN UND SCHALTER
- \* MENÜS UND BARGRAPH PER TOUCHEINSTELLBAR
- \* DOTMATRIXDISPLAY ALS ZWEITANZEIGE DIREKT ANSCHLIESSBAR

## BESTELLBEZEICHNUNG

320x240 DOTS 5,7" MIT CFL-BELEUCHTUNG, BLAUNEGATIV

WIE VOR, JEDOCH OHNE TOUCH PANEL

320x240 DOTS, WEISSE LED-BELEUCHTUNG, POSITIV MODE, FSTN

WIE VOR FSTN, JEDOCH OHNE TOUCH PANEL

**EA KIT320-8CTP**

**EA KIT320-8C**

**EA KIT320-8LWTP**

**EA KIT320-8LW**

## OPTIONEN/ZUBEHÖR

VERSORGUNG +9..35V = STATT +5V =

RS-422 SCHNITTSTELLE STATT RS-232

OPTOKOPPLER ONBOARD FÜR 8 EIN- UND 8 AUSGÄNGE

ALUMINIUM EINBAUBLENDE, MATT-SCHWARZ ELOXIERT

ALUMINIUM EINBAUBLENDE, BLAU ELOXIERT

KABEL 1,5m FÜR ANSCHLUSS AN 9-POL. SUB-D (RS-232 FEMALE)

DISKETTE FÜR MAKROPROGRAMMIERUNG (PC-WIN95)

**EA OPT-9/35V**

**EA OPT-RS4224**

**EA OPT-OPT016**

**EA 0FP320-8SW**

**EA 0FP320-8BL**

**EA KV24-9B**

**EA DISK320**

**ELECTRONIC  
ASSEMBLY** GMBH

LOCHHAMER SCHLAG 17 · D-82166 GRÄFELFING  
TEL 089/8541991 · FAX 089/8541721 · <http://www.lcd-module.de>

### ALLGEMEINES

EA KIT320 ist eine komplett aufgebaute Steuer- und Bedieneinheit mit diversen eingebauten Funktionen. Das kompakt aufgebaute Display bietet zusammen mit dem sehr guten Supertwistkontrast eine sofort einsetzbare Einheit. Die Ansteuerung erfolgt über die Standard Schnittstelle RS-232 oder RS-422. Die Bedieneinheit enthält neben kompletten Grafikroutinen zur Displayausgabe auch verschiedenste Schriften.

Die Programmierung erfolgt über hochsprachenähnliche Grafikbefehle; die zeitraubende Programmierung von Zeichensätzen und Grafikroutinen entfällt hier völlig. Die simple Verwendung von Makros und die Eingabemöglichkeit über Touchpanel machen es zu einem richtigen Power Display.

### HARDWARE

Die Bedieneinheit ist für +5V Betriebsspannung ausgelegt. Optional ist eine Versorgung mit 9..35V möglich. Die Datenübertragung erfolgt seriell asynchron im RS-232 oder RS-422 Format. Das Übertragungsformat ist fest auf 8 Datenbits, 1 Stopbit, no Parity eingestellt. Die Baudrate kann über DIP-Schalter von 2.400 Baud bis zu 115.200 Baud ausgewählt werden. Handshakeleitungen RTS und CTS stehen zur Verfügung.

Datenformat:



### TOUCHPANEL

Die Versionen EA KIT320-8CTP und -8LWTP sind mit einem integrierten Touch Panel ausgerüstet. Durch Berühren des Displays können hier Eingaben gemacht und Einstellungen per Menü oder Bargraphs getätigt werden. Die Beschriftung der "Tasten" ist flexibel und auch während der Laufzeit änderbar (verschiedene Sprachen, Icons). Das Zeichnen der einzelnen "Tasten", sowie das Beschriften wird von der eingebauten Software komplett übernommen.

### SOFTWARE

Die Programmierung der Bedieneinheit erfolgt über Befehle wie z.B. *Zeichne ein Rechteck von (0,0) nach (64,15)*. Es ist keine zusätzliche Software oder Treiber erforderlich. Zeichenketten lassen sich **pixelgenau** platzieren. Blinkattribute können beliebig oft vergeben werden - auch für Grafiken. Das Mischen von Text und Grafik ist jederzeit möglich. Es können bis zu 16 verschiedene Zeichensätze verwendet werden. Jeder Zeichensatz kann wiederum 2- bis 8-fach gezoomt werden. Mit dem größten Zeichensatz lassen sich somit bildschirmfüllende Worte und Zahlen darstellen.

### ZUBEHÖR

#### Diskette zur Makroerstellung

Zur Makroprogrammierung ist eine Diskette EA DISK320 erforderlich<sup>\*)</sup>. Diese übersetzt die in eine Textdatei eingegebenen Befehle in einen für die Bedieneinheit lesbaren Code und programmiert diesen dauerhaft ins FLASH PROM.

#### Kabel für PC

Für die einfache Anbindung an PC's (speziell zur Makroprogrammierung) liefern wir optional ein ca. 1,5m langes Kabel mit 9-pol. SUB-D Stecker (female) EA KV24-9B. Einfach an die COM 1 oder COM 2 anstecken und loslegen. Hinweis: Das Kabel ist nicht für die RS-422 Version EA OPT-RS4224 geeignet.

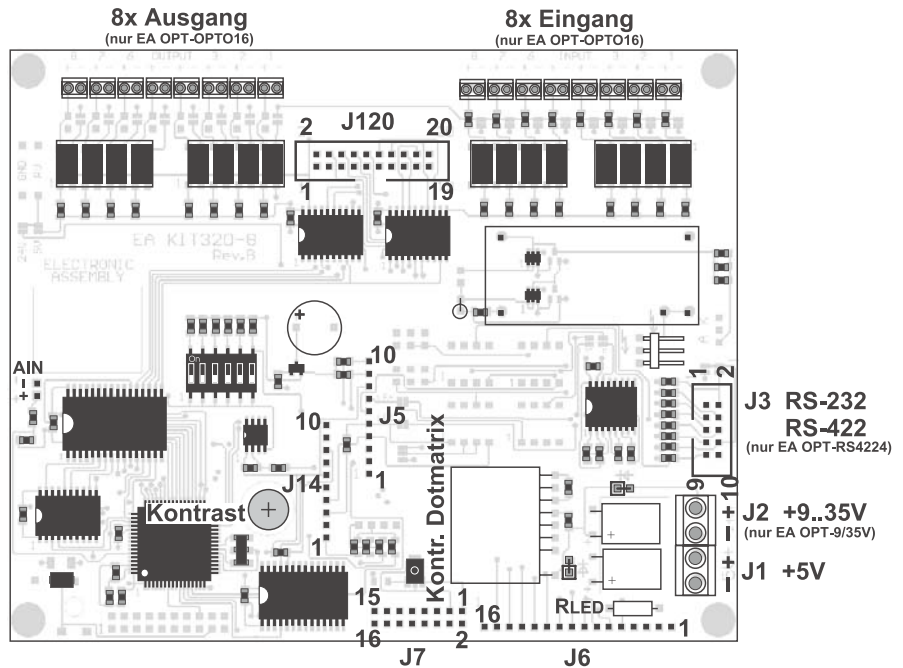
<sup>\*)</sup> auch im Internet unter <http://www.lcd-module.de/deu/disk/disk320.zip>

## ELECTRONIC ASSEMBLY

### VERSORGUNGSSPANNUNG

In der Standardausführung wird die Versorgungsspannung von +5V über die Schraubklemme J1 eingespeist. Liegt die Version für 9..35V (EA OPT-9/35V) vor, so erfolgt die Stromversorgung über J2.

**Achtung:** Unbedingt auf die richtige Polarität achten! Eine auch noch so kurzzeitige Verpolung kann zur sofortigen Zerstörung der gesamten Bedieneinheit führen.



Ansicht von hinten

### BAUDRATEN

Die Baudrate läßt sich über die linken 3 DIP-Schalter einstellen. Im Auslieferungszustand sind 9.600 Baud eingestellt (DIP 3 ON). Bitte beachten Sie, dass der interne Datenpuffer lediglich 128 Byte umfaßt. Deshalb muss unbedingt die Handshakeleitung RTS abgefragt werden (+10V Pegel: Daten können angenommen werden; -10V Pegel: Display ist Busy). Das Datenformat ist fest eingestellt auf 8 Datenbits, 1 Stopbit, keine Parität.

DIP Schalter			Baudraten
1	2	3	Datenformat
ON	ON	ON	8,N,1
OFF	ON	ON	2400
ON	OFF	ON	4800
OFF	OFF	ON	9600
ON	ON	OFF	19200
OFF	ON	OFF	38400
ON	OFF	OFF	57600
OFF	OFF	OFF	115200

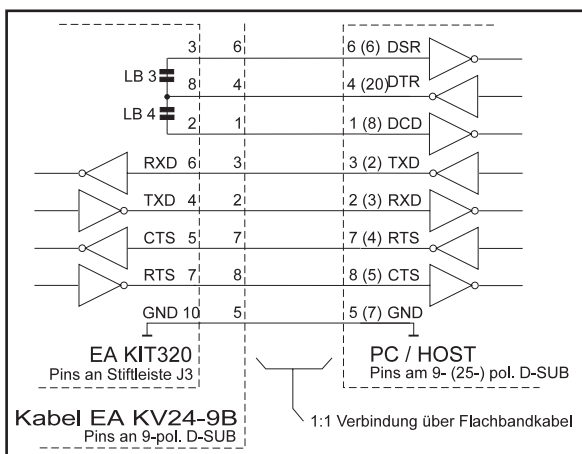
### RS-232/RS-422

Standardmäßig wird die Bedieneinheit mit einer RS-232 Schnittstelle ausgeliefert. Die Stiftleiste J3 hat dann die Pinbelegung wie in der Tabelle links abgebildet. J3 ist im Raster 2,54mm ausgeführt. Wird die Bedieneinheit zusammen mit der Option EA OPT-RS4224 bestellt, sind spezielle RS-422 Treiber bestückt. Damit ist die Pinbelegung in der Tabelle rechts gültig.

RS-232 Anschluß J3			
Pin	Symbol	In/Out	Funktion
1	VDD	-	+5V Versorgung
2	DCD	-	Brücke nach DTR
3	DSR	-	Brücke 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
10	GND	-	0V Masse

RS-422 Anschluß J3		
Pin	Symbol	Funktion
1	VDD	+5V Versorgung
2	Data In-	Receive Data
3	Data In+	Receive Data
4	Data Out-	Transmit Data
5	Data Out+	Transmit Data
6	HS In-	Handshake
7	HS In+	Handshake
8	HS Out-	Handshake
9	HS Out+	Handshake
10	GND	0V Masse

An der Löttaugenleiste J5 stehen übrigens die gleichen seriellen Daten mit 5V Pegeln und TTL-Logik zur Verfügung. Diese Pegel sind für den direkten Anschluß an einen µC geeignet. Bei Verwendung dieser Signale müssen dann die 4 Lötbrücken LB1, LB2, LB5 und LB6 geöffnet werden!



Erweiterung J5			
Pin	Symbol	In/Out	Funktion
1	VU	-	9..35V Versorgung
2	VDD	-	+5V Versorgung
3	GND	-	0V, Masse
4	TxD5	Out	Transmit Data
5	RxD5	In	Receive Data
6	RTS5	Out	Request To Send
7	CTS5	In	Clear To Send
8	RESET	In	L: Reset
9	SCL	Out	I2C Bus, Clock
10	SDA	In/Out	I2C Bus, Data

## EIN- UND AUSGÄNGE

Alle Bedieneinheiten EA KIT320 werden mit 8 digitalen Ein- und 8 Ausgängen (5V CMOS Pegel, nicht potentialfrei) geliefert.

### 8 Ausgänge

Jede Leitung kann per Befehl "ESC Y W" individuell angesteuert werden. Pro Leitung kann ein Strom von max. 6mA geschaltet werden. Es ist somit möglich, mit einem Ausgang direkt eine LED (low current) zu schalten. Größere Ströme können mittels externen Transistors verstärkt werden.

### 8 Eingänge

Die Eingänge können direkt über die serielle Schnittstelle abgefragt und ausgewertet werden ("ESC Y R"). Zusätzlich ist es möglich, bei Änderungen an den Eingängen ein Bit-/Portmakro automatisch aufzurufen. Die automatische Portabfrage läßt sich mit dem Befehl "ESC Y A 0" deaktivieren.

**Portmakros:** durch die binäre Kombination von 8 Eingängen sind bis zu 256 Portmakros ansprechbar.

**Bitmakros** wirken jeweils nur auf einen Eingang. Bitmakro 1..8 wird bei Änderung auf HIGH-Pegel an einem der Eingänge 1..8 aufgerufen. Bitmakro 9..16 werden bei Änderung auf LOW-Pegel aufgerufen.

Bei jeder Änderung des Eingangports werden zuerst die Bitmakros und dann das Portmakro ausgeführt. Ist kein Makro definiert so wird der neue Portzustand über die RS232/RS422 gesendet. Jedes der Makros kann den Bildschirminhalt ändern oder Ausgänge schalten. Damit können vielfältige Steuerungsaufgaben gelöst werden. Für die Erstellung der Makros benötigt man einen PC und die Diskette EA DISK320.

**Anmerkung:** Die Logik ist für langsame Vorgänge ausgelegt; d.h. mehr als 3 Änderungen pro Sekunde können nicht mehr sinnvoll ausgeführt werden. Falls ein Eingang offen ist, so ist dieser High (interner 100 kOhm PullUp).

## EIN- UND AUSGÄNGE ÜBER OPTOKOPPLER (EA OPT-OPTO16)

Die Ein- und Ausgänge können optional mit Optokopplern ausgestattet werden (EA OPT-OPTO16). Die Ein- und Ausgänge sind dann sowohl von der restlichen Elektronik, als auch untereinander isoliert. Der Anschluß erfolgt über 16 einzelne Schraubklemmen.

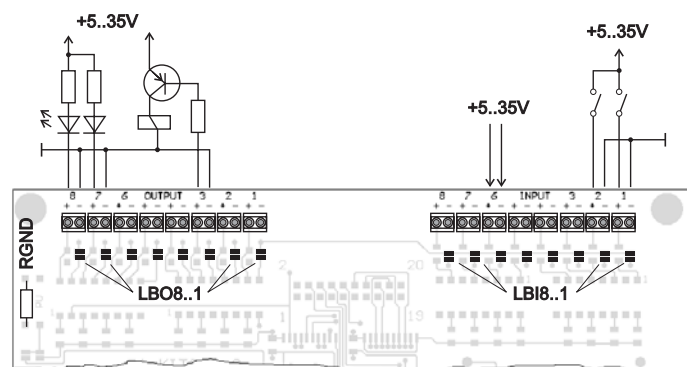
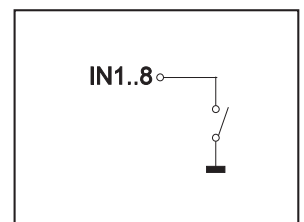
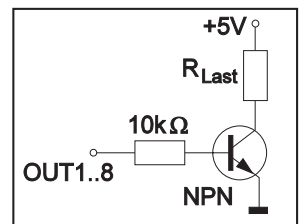
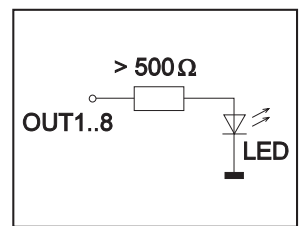
An allen 8 Eingängen können direkt Spannungen von 5..35V angelegt werden. Pegel über 4V werden als H-Pegel erkannt, Pegel unter 2V gelten als L-Pegel. Spannungen zwischen 2 und 4V sind undefiniert.

Als Ausgang ist jeweils der Kollektor und Emitter eines Transistors an den Schraubklemmen herausgeführt. Jeder Ausgang kann max. 10mA schalten.

**Hinweis:** Der Minuspol jeder Schraubklemme kann durch Schließen der Lötbrücken LB11..8 bzw. LBO1..8 zusammengeschaltet werden. Zusätzlich können diese Lötbrücken auf die Systemmasse GND gelegt werden (0Ω Brücke RGND einlöten).

**Anmerkung:** Die Optokoppler invertieren die Eingangslogik (alle Eingänge offen: Portmakro 255). Hier empfiehlt es sich (z.B. im Power-On-Makro) mit dem Befehl "ESC Y I 1" die Eingänge invertiert auszuwerten (d.h. alle Eingänge offen: Portmakro 0).

Ein- und Ausgänge J120					
Pin	Symbol	Funktion	Pin	Symbol	Funktion
1	VDD	+5V Versorgung	2	GND	0V, Masse
3	OUT1 / MO8	Port Ausgang 1 Matrix Ausgang 8	4	IN1 / MI8	Port Eingang 1 Matrix Eingang 8
5	OUT2 / MO7	Port Ausgang 2 Matrix Ausgang 7	6	IN2 / MI7	Port Eingang 2 Matrix Eingang 7
7	OUT3 / MO6	Port Ausgang 3 Matrix Ausgang 6	8	IN3 / MI6	Port Eingang 3 Matrix Eingang 6
9	OUT4 / MO5	Port Ausgang 4 Matrix Ausgang 5	10	IN4 / MI5	Port Eingang 4 Matrix Eingang 5
11	OUT5 / MO4	Port Ausgang 5 Matrix Ausgang 4	12	IN5 / MI4	Port Eingang 5 Matrix Eingang 4
13	OUT6 / MO3	Port Ausgang 6 Matrix Ausgang 3	14	IN6 / MI3	Port Eingang 6 Matrix Eingang 3
15	OUT7 / MO2	Port Ausgang 7 Matrix Ausgang 2	16	IN7 / MI2	Port Eingang 7 Matrix Eingang 2
17	OUT8 / MO1	Port Ausgang 8 Matrix Ausgang 1	18	IN8 / MI1	Port Eingang 8 Matrix Eingang 1
19	GND	0V, Masse	20	VDD	+5V Versorgung



## EXTERNE MATRIX-TASTATUR

Am Steckeranschluss J120 kann eine Matrix-Tastatur (einzelne Tasten bis zur 8x8 Matrix) angeschlossen werden. Mit dem Befehl 'ESC Y M n1 n2 n3' werden die Anzahl der verwendeten Ein- und Ausgänge der Ports (n1,n2=1..8) definiert und die Tastenentprellung (n3=0..7 in 50ms Schritten) festgelegt. Bitte beachten Sie, dass bei Anschluß einer externen Tastatur die digitalen Eingänge um die Anzahl n1, und die Ausgänge um die Anzahl n2 reduziert werden.

Jede Taste wird i.d.R. zwischen einen Ausgang und einen Eingang geschaltet. Jeder Eingang ist mit einem ca. 100kΩ Pullup abgeschlossen. Um Doppeltastendrucke zu erkennen, müssen die Ausgänge voneinander entkoppelt werden. Dies geht am besten mit Schottky-Dioden (z.B. BAT 43).

### Senden der Tastendrucke

Bei jedem Druck einer Taste (Tastenummer 1..64) wird das dazugehörige Matrix-Makro ausgeführt, oder wenn kein Makro definiert ist, die Tastenummer mit Kennbuchstaben gesendet. Das Loslassen der Taste wird nicht gesendet. Soll auch das Loslassen gesendet werden, so kann das über die Definition des Matrix Makros Nr.0 realisiert werden.

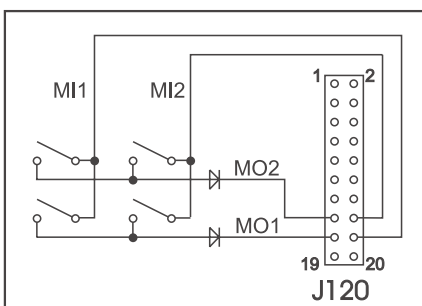
Hinweis Falls die Handshakleitung CTS das Senden nicht erlaubt, werden bis zu 8 Tastendrucke im Tastenpuffer zwischengespeichert. Ist der Puffer voll, können ältere Tastendrucke verloren gehen.

### Bestimmung der Tastenummer:

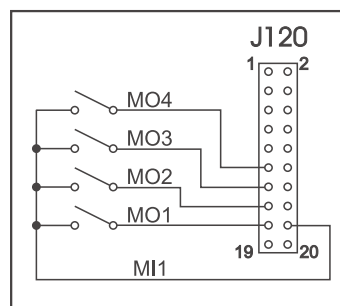
**TastenNr = (AusgangNr - 1) \* AnzahlEingänge + EingangNr** (Ausgang = MOx, Eingang = MIx).

### Beispiel: Anschluss von 4 Tasten in 3 Variationen

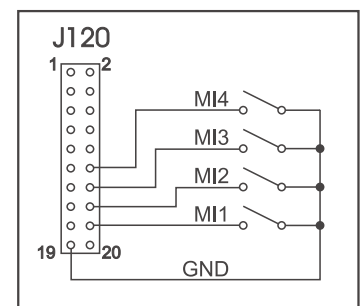
- Variante 1: Mit dem Befehl 'ESC Y M 2 2 ..' werden die 4 Tasten als 2x2 Matrix definiert. Die Tasten werden an 2 Eingänge (MI1, MI2) und 2 Ausgänge (MO1, MO2) angeschlossen. Die Ausgänge sind hier mit Dioden voneinander entkoppelt um Doppeltastendrucke erkennen zu können. Es stehen weiterhin 6 Eingänge und 6 Ausgänge als Port-Anschlüsse zur Verfügung.
- Variante 2: Mit dem Befehl 'ESC Y M 1 4 ..' werden die 4 Tasten als 1x4 Matrix definiert. Die Tasten werden an 4 Ausgänge (MO1..MO4) angeschlossen und über den Eingang MI1 eingelesen. Es stehen weiterhin 7 Eingänge und 4 Ausgänge als Port-Anschlüsse zur Verfügung.
- Variante 3: Wird nur ein Ausgang benutzt (4x1 Matrix), so können die Tasten auch gegen Masse geschaltet werden und direkt an den Eingänge eingelesen werden (= 4x0 Matrix). Mit dem Befehl 'ESC Y M 4 0 ..' werden die 4 Tasten an den 4 Eingängen (MI1..MI4) definiert. Es stehen weiterhin 4 Eingänge und alle 8 Ausgänge als Port-Anschlüsse zur Verfügung.



Variante 1: 2x2 Matrix



Variante 2: 1x4 Matrix



Variante 3: 4x0 Matrix

### TOUCH PANEL (NUR EA KIT320-8xxTP)

Die Versionen EA KIT320-8CTP und -8LWTP werden mit einem analogen resistiven Touchpanel geliefert. Bis zu 80 Touchbereiche (Tasten, Schalter, Menüs, Bargraph eingaben), können gleichzeitig definiert werden. Die Bedieneinheit unterstützt dieses Touch Panel mit komfortablen Befehlen (Seite 12). Beim Berühren der Touch-Tasten können diese automatisch invertiert werden und ein Summer signalisiert die Berührung. Der definierte Return-Code der Taste wird über die serielle Schnittstelle gesendet oder es wird, statt dessen ein internes Touch Makro mit der Nummer des Return-Codes gestartet.

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	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80

Vordefinierte TouchFelder fx für die Befehle 'ESC AC' und 'ESC AG'

### TOUCHPANELABGLEICH

Das Touchpanel ist bei Auslieferung abgeglichen und sofort einsatzbereit. Durch Alterung und Abnutzung kann es nötig sein, dass das Touchpanel neu abgeglichen werden muss.

#### Abgleichprozedur:

1. Beim Einschalten Touch berühren und gedrückt halten. Nach Erscheinen der Meldung "touch adjustment ?" den Touch wieder loslassen (alternativ den Befehl 'ESC @' senden).
2. Innerhalb 1 Sekunde den Touch nochmals für mindestens 1 Sekunde berühren.
3. Den Anweisungen zum Abgleich folgen (2 Punkte *Linksoben* und *Rechtunten* betätigen).

### ANSCHLUSS FÜR DOTMATRIX-DISPLAY

An den Lötäugen J6 und J7 kann maximal ein externes Dotmatrix-Modul (mit HD44780 oder kompatibel) mit 1x8 bis zu 4x20 bzw. 2x40 Zeichen angeschlossen werden. Dieses Dotmatrix-Display wird sehr komfortabel über die Terminal-Befehle 'ESC T xx' angesprochen. Alternativ stehen Befehle zur Verfügung ('ESC L xx') um den HD44780 direkt anzusteuern. Ein Poti für die Kontrasteinstellung ist bereits bestückt. Für eine evtl. LED-Hintergrundbeleuchtung kann ein passender Vorwiderstand  $R_{LED}$  bestückt werden.

Dotmatrix-Anschluss J6 + J7			
Pin	Symbol	Pegel	Beschreibung
1	VSS	L	Versorgung 0V, GND
2	VDD	H	Versorgung +5V
3	VEE	-	Displayspannung 0V-5V
4	RS	H/L	Register Select
5	R/W	H/L	H: Read / L: Write
6	E	H	Enable
7	D0	H/L	Datenleitung 0 (LSB)
8	D1	H/L	Datenleitung 1
9	D2	H/L	Datenleitung 2
10	D3	H/L	Datenleitung 3
11	D4	H/L	Datenleitung 4
12	D5	H/L	Datenleitung 5
13	D6	H/L	Datenleitung 6
14	D7	H/L	Datenleitung 7
15	A	-	Anode für LED (RLED)
16	K	L	Kathode f. LED (=VSS)

### MAKROPROGRAMMIERUNG

Einzelne oder mehrere Befehlsfolgen können als sog. Makros zusammengefasst und im Daten-Flash fest abgespeichert werden. Diese können dann mit den Befehlen *Makro ausführend* gestartet werden. Es gibt verschiedene Makrotypen:

#### Normal Makro (0..255)

Start per Befehl 'ESC MN xx' über serielle Schnittstelle oder von einem anderen Makro aus. Es können auch mehrere hintereinander liegende Makros automatisch zyklisch aufgerufen werden (Movie, sich drehende Sanduhr, mehrseitiger Hilfetext). Diese automatischen Makros werden solange abgearbeitet bis ein Befehl über die RS-232 empfangen wird, oder ein anderes Makro mit entsprechendem Return-Code ausgelöst wird (z.B. Touch-, Port-, Matrixmakro).

#### Touch Makro (1..255)

Start beim Berühren/Loslassen eines Touchfeldes (nur bei Versionen mit Touch Panel TP) oder per Befehl 'ESC MT xx'.

#### Menü Makro (1..255)

Start bei Auswahl eines Menüeintrages oder per Befehl 'ESC MM xx'.

#### Bit Makro (1..8) bzw. (9..16)

Start bei Anlegen/Änderung einer Spannung an einzelnen Eingängen IN 1..8 (Bitweise) oder per Befehl 'ESC MB xx'. Die Bit-Makros 1..8 reagieren auf steigende Flanke, Bit-Makros 9..16 auf die fallende Flanke der Eingänge 1..8.

Port Makro (0..255)

Start bei Anlegen/Änderung einer Spannung an den 8 Eingängen IN 1..8 (binär kombiniert) oder per Befehl 'ESC MP xx'.

Matrix Makro (0..64)

Matrix-Makro 1..64: Start beim Drücken einer Taste oder per Befehl 'ESC MX xx'.  
Matrix-Makro 0: Start beim Loslassen wenn keine Taste mehr gedrückt ist oder per Befehl.  
Die Matrixastatur wird an den Ein- und Ausgängen angeschlossen; es kann max. eine 8x8 Matrixastatur angeschlossen werden.

Power-On-Makro

Start nach dem Einschalten Power-On. Hier kann man zB. den Cursor abschalten und einen Startbildschirm definieren.

Reset-Makro

Start nach einem externen Reset oder nach einem Spannungseinbruch unter 4,7V (VDD-VSS).

Watchdog-Makro

Start nach einem Fehlerfall (z.B. Absturz)

**Achtung:** Wird im Power-On-, Reset- oder Watchdog-Makro eine Endlosschleife programmiert, ist das Display nicht mehr ansprechbar. In diesen Fall hilft nur noch: DIP-Schalter 5 auf ON, Power off, Power on und dann DIP 5 wieder auf off. Jetzt müssen die Makros wieder neu eingespielt werden.

**256 BILDER FEST ABGELEGT**

Um die Übertragungszeiten der seriellen Schnittstelle zu verkürzen, oder auch um Speicherplatz im Prozessorsystem zu sparen, können bis zu 256 Bilder im internen Daten-Flash abgelegt werden. Der Aufruf erfolgt über den Befehl "ESC UI" oder aus einem Makro heraus. Verwendet werden können alle Bilder im Windows BMP-Format (nur monochrome Bilder). Die Erstellung und Bearbeitung erfolgt über Standardsoftware wie z.B. Windows Paint oder Photoshop (nur schwarz/weiss = 1 Bit).

**ERSTELLEN INDIVIDUELLER MAKROS**

Um nun Ihre speziellen Makros erstellen zu können, benötigen Sie folgende Hilfsmittel:

- die Diskette EA DISK320<sup>\*)</sup>; sie enthält einen Compiler, Beispiele und Fonts
- einen PC mit serieller Schnittstelle COM1 oder COM2
- einen Texteditor wie z.B. WordPad, Textpad o.ä.

Um eine Befehlsfolge als Makro zu definieren, werden alle Befehle auf dem PC in eine Datei z.B. DEMO.KMC geschrieben. Hier bestimmen Sie, welche Zeichensätze eingebunden werden und in welchen Makros welche Befehlsfolgen stehen sollen.

Sind die Makros definiert, startet man das Programm C:>KITCOMP DEMO.KMC. Dieses erzeugt eine Daten-Flash-Datei DEMO.DF, welche dann automatisch mit der eingetragenen Baudrate in das Daten-Flash gebrannt wird. Dieser Vorgang dauert nur wenige Sekunden und sofort danach können die selbstdefinierten Makros genutzt werden. Eine ausführliche Beschreibung zur Programmierung der Makros finden Sie zusammen mit Beispielen auf der Diskette EA DISK320<sup>\*)</sup> unter dem Namen DOKU.DOC (für WORD) bzw. DOKU.TXT (DOS).

**SCHREIBSCHUTZ FÜR MAKROPROGRAMMIERUNG**

Platine ab Rev.C: Wird der DIP-Schalter 6 (write enable) auf OFF gestellt läßt sich ein versehentliches Überschreiben der einprogrammierten Makros, Bilder und Fonts verhindern. Zum erneuten Programmieren muss dieser DIP-Schalter vorher wieder auf ON gestellt werden.

(Achtung: Bei alten Platinenversionen Rev.A und Rev.B (vor 2002) war die Bedeutung des DIP-Schalter 6 umgekehrt (write protect); ON=Schreibschutz; OFF=Programmieren möglich).

<sup>\*)</sup> auch im Internet unter <http://www.lcd-module.de/deu/disk/disk320.zip>

internur Font 1: 4x6 monospaced

+ Lower Upper	\$0 (0)	\$1 (1)	\$2 (2)	\$3 (3)	\$4 (4)	\$5 (5)	\$6 (6)	\$7 (7)	\$8 (8)	\$9 (9)	\$A (10)	\$B (11)	\$C (12)	\$D (13)	\$E (14)	\$F (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)		`	a	b	c	d	e	f	g	h	i	j	k	l	m	n
\$70 (dez: 112)		p	q	r	s	t	u	v	w	x	y	z	{	}		
\$80 (dez: 128)		ç	ü	é	â	ä	à	ç	ê	ë	è	ì	í	î	ï	ñ
\$90 (dez: 144)		é	æ	œ	ô	ö	ò	ù	û	ü	ç	£	¥	β	f	

internur Font 2: 5x6 monospaced

+ Lower Upper	\$0 (0)	\$1 (1)	\$2 (2)	\$3 (3)	\$4 (4)	\$5 (5)	\$6 (6)	\$7 (7)	\$8 (8)	\$9 (9)	\$A (10)	\$B (11)	\$C (12)	\$D (13)	\$E (14)	\$F (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)		`	a	b	c	d	e	f	g	h	i	j	k	l	m	n
\$70 (dez: 112)		p	q	r	s	t	u	v	w	x	y	z	{	}		
\$80 (dez: 128)		ç	ü	é	â	ä	à	ç	ê	ë	è	ì	í	î	ï	ñ
\$90 (dez: 144)		é	æ	œ	ô	ö	ò	ù	û	ü	ç	£	¥	β	f	

internur Font 3: 6x8 monospaced

+ Lower Upper	\$0 (0)	\$1 (1)	\$2 (2)	\$3 (3)	\$4 (4)	\$5 (5)	\$6 (6)	\$7 (7)	\$8 (8)	\$9 (9)	\$A (10)	\$B (11)	\$C (12)	\$D (13)	\$E (14)	\$F (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)		`	a	b	c	d	e	f	g	h	i	j	k	l	m	n
\$70 (dez: 112)		p	q	r	s	t	u	v	w	x	y	z	{	}		
\$80 (dez: 128)		ç	ü	é	â	ä	à	ç	ê	ë	è	ì	í	î	ï	ñ
\$90 (dez: 144)		é	æ	œ	ô	ö	ò	ù	û	ü	ç	£	¥	β	f	
\$A0 (dez: 160)		á	í	ó	ú	ñ	ñ	¿	¿	¼	½	¾	i	«	»	
\$B0 (dez: 176)		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
\$C0 (dez: 192)		L	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
\$D0 (dez: 208)		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
\$E0 (dez: 224)		α	β	Γ	π	Σ	σ	μ	τ	ϕ	θ	Ω	δ	φ	ε	π
\$F0 (dez: 240)		≡	±	≥	≤	Γ	J	÷	≈	•	•	•	•	•	•	•

internur Font 4: 8x8 monospaced

+ Lower Upper	\$0 (0)	\$1 (1)	\$2 (2)	\$3 (3)	\$4 (4)	\$5 (5)	\$6 (6)	\$7 (7)	\$8 (8)	\$9 (9)	\$A (10)	\$B (11)	\$C (12)	\$D (13)	\$E (14)	\$F (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)		`	a	b	c	d	e	f	g	h	i	j	k	l	m	n
\$70 (dez: 112)		p	q	r	s	t	u	v	w	x	y	z	{	}		
\$80 (dez: 128)		ç	ü	é	â	ä	à	ç	ê	ë	è	ì	í	î	ï	ñ
\$90 (dez: 144)		é	æ	œ	ô	ö	ò	ù	û	ü	ç	£	¥	β	f	
\$A0 (dez: 160)		á	í	ó	ú	ñ	ñ	¿	¿	¼	½	¾	i	«	»	
\$B0 (dez: 176)		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
\$C0 (dez: 192)		L	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
\$D0 (dez: 208)		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
\$E0 (dez: 224)		α	β	Γ	π	Σ	σ	μ	τ	ϕ	θ	Ω	δ	φ	ε	π
\$F0 (dez: 240)		≡	±	≥	≤	Γ	J	÷	≈	•	•	•	•	•	•	•

internur Font 5: 7x12 monospaced

+ Lower Upper	\$0 (0)	\$1 (1)	\$2 (2)	\$3 (3)	\$4 (4)	\$5 (5)	\$6 (6)	\$7 (7)	\$8 (8)	\$9 (9)	\$A (10)	\$B (11)	\$C (12)	\$D (13)	\$E (14)	\$F (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)		`	a	b	c	d	e	f	g	h	i	j	k	l	m	n
\$70 (dez: 112)		p	q	r	s	t	u	v	w	x	y	z	{	}		
\$80 (dez: 128)		ç	ü	é	â	ä	à	ç	ê	ë	è	ì	í	î	ï	ñ
\$90 (dez: 144)		é	æ	œ	ô	ö	ò	ù	û	ü	ç	£	¥	β	f	
\$A0 (dez: 160)		á	í	ó	ú	ñ	ñ	¿	¿	¼	½	¾	i	«	»	
\$B0 (dez: 176)		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
\$C0 (dez: 192)		L	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
\$D0 (dez: 208)		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
\$E0 (dez: 224)		α	β	Γ	π	Σ	σ	μ	τ	ϕ	θ	Ω	δ	φ	ε	π
\$F0 (dez: 240)		≡	±	≥	≤	Γ	J	÷	≈	•	•	•	•	•	•	•

internur Font 6: 8x16 monospaced

+ Lower Upper	\$0 (0)	\$1 (1)	\$2 (2)	\$3 (3)	\$4 (4)	\$5 (5)	\$6 (6)	\$7 (7)	\$8 (8)	\$9 (9)	\$A (10)	\$B (11)	\$C (12)	\$D (13)	\$E (14)	\$F (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)		`	a	b	c	d	e	f	g	h	i	j	k	l	m	n
\$70 (dez: 112)		p	q	r	s	t	u	v	w	x	y	z	{	}		
\$80 (dez: 128)		ç	ü	é	â	ä	à	ç	ê	ë	è	ì	í	î	ï	ñ
\$90 (dez: 144)		é	æ	œ	ô	ö	ò	ù	û	ü	ç	£	¥	β	f	
\$A0 (dez: 160)		á	í	ó	ú	ñ	ñ	¿	¿	¼	½	¾	i	«	»	
\$B0 (dez: 176)		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
\$C0 (dez: 192)		L	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
\$D0 (dez: 208)		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
\$E0 (dez: 224)		α	β	Γ	π	Σ	σ	μ	τ	ϕ	θ	Ω	δ	φ	ε	π
\$F0 (dez: 240)		≡	±	≥	≤	Γ	J	÷	≈	•	•	•	•	•	•	•



## ELECTRONIC ASSEMBLY

### INTEGRIERTE UND EXTERNE FONTS

Es sind standardmäßig 6 monospaced Zeichensätze integriert, die sowohl im Terminal als auch im Grafikmodus verwendet werden können. Jeder Zeichensatz kann zudem bei der Grafikausgabe von 1- bis 8-facher Höhe verwendet werden. Unabhängig davon lässt sich auch die Breite verdoppeln bis verachtfachen.

Jedes Zeichen kann **pixelgenau** platziert werden. Text und Grafik kann beliebig gemischt dargestellt werden. Auch mehrere verschiedene Schriftgrößen lassen sich gemeinsam darstellen.

Jeder Text lässt sich linksbündig, rechtsbündig und zentriert ausgeben. Auch Drehungen in 90° Schritten (z.B. für vertikalen Einbau des Displays) sind möglich.

Die Makroprogrammierung erlaubt die Einbindung von weiteren 10 Fonts. Es sind auch proportionale Zeichensätze möglich (nur im Grafikmode), die ein schöneres Schriftbild ergeben und weniger Platz auf dem Bildschirm benötigen. Es können alle nur erdenklichen Schriften mit bis zu 255x240 Pixeln Größe mit einem Texteditor erstellt und über den Kitcompiler einprogrammiert werden.

z.B. externer Font 10: GENEVA15.FXT proportional

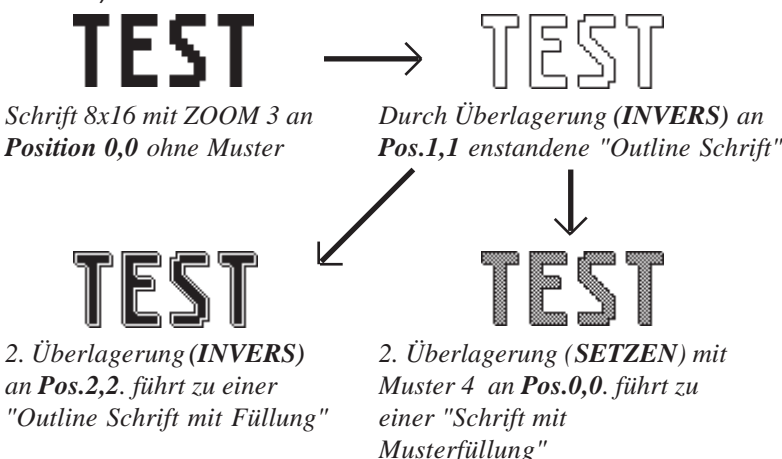
+ Lower Upper	\$0 (0)	\$1 (1)	\$2 (2)	\$3 (3)	\$4 (4)	\$5 (5)	\$6 (6)	\$7 (7)	\$8 (8)	\$9 (9)	\$A (10)	\$B (11)	\$C (12)	\$D (13)	\$E (14)	\$F (15)
\$20 (dez: 32)		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
\$30 (dez: 48)	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
\$40 (dez: 64)		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
\$50 (dez: 80)	P	Q	R	S	T	U	V	W	X	Y	Z	[	]	^	_	
\$60 (dez: 96)	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
\$70 (dez: 112)	p	q	r	s	t	u	v	w	x	y	z	{	}			
\$80 (dez: 128)	Ä	Å	Ç	É	Ë	Ï	Ñ	Ö	Ü	á	â	ã	ä	å	ç	é
\$90 (dez: 144)	ê	ë	í	î	ï	ñ	ó	ô	õ	ö	ù	ú	û	ü		
\$A0 (dez: 160)	†	°	¢	£	§	•	¶	ß	®	©	™	'	¨	≠	Æ	Ø
\$B0 (dez: 176)	∞	±	≤	≥	¥	µ	∂	Σ	Π	π	∫	∑	∏	Ω	æ	ø
\$C0 (dez: 192)	¿	¡	¬	✓	ƒ	≈	Δ	«	»	...	À	Á	Ã	Ö	œ	
\$D0 (dez: 208)	-	-	"	"	'	'	+	◊	ÿ							
\$E0 (dez: 224)																
\$F0 (dez: 240)																

z.B. externer Font 7: CHICAGO.FXT proportional

+ Lower Upper	\$0 (0)	\$1 (1)	\$2 (2)	\$3 (3)	\$4 (4)	\$5 (5)	\$6 (6)	\$7 (7)	\$8 (8)	\$9 (9)	\$A (10)	\$B (11)	\$C (12)	\$D (13)	\$E (14)	\$F (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)	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
\$70 (dez: 112)	p	q	r	s	t	u	v	w	x	y	z	{	}			Δ
\$80 (dez: 128)	Ç	ü	é	â	ä	à	ã	ç	ê	ë	è	í	î	ï	Ä	Å
\$90 (dez: 144)	É	æ	Æ	ô	ö	ò	û	ÿ	Ö	Ü	ç	£	¥	ß	f	
\$A0 (dez: 160)	á	í	ó	ú	ñ	Ñ	á	ä	¿	½	¼	¡	«	»		
\$B0 (dez: 176)	ã	õ	ø	ø	œ	œ	À	Ã	Ö	¨	'	†	¶	©	®	™
\$C0 (dez: 192)	ij	Y	X	1	7	T	n	l	T	n	U	'	∫	∏	∑	
\$D0 (dez: 208)	0	U	9	z	7	W	l	7	0	9	4	§	^	∞		
\$E0 (dez: 224)	α	β	Γ	π	Σ	Ω	μ	τ	ϕ	θ	Ω	δ	φ	ϕ	€	Π
\$F0 (dez: 240)	≡	±	≥	≤	∫	J	÷	≈	°	*	.	√	²	³	-	

### TIP: SCHRIFTEFFEKTE

Mit dem Befehl 'ESC ZV' Verknüpfung und 'ESC ZM' Muster, können bei grossen Schriften interessante Effekte durch Überlagerung (mehrmaliges versetztes Schreiben eines Wortes) erzielt werden.



### SCHRIFTBILD

Die untere Hardcopy eines Bildinhaltes zeigt die 6 internen sowie 8 externe Schriften.



## ALLE BEFEHLE AUF EINEN BLICK

EA KIT320: Befehlstabelle 1							nach	
Befehl	Codes			Anmerkung			Reset	
<b>Befehle zur Ausgabe von Zeichenketten</b>								
Zeichenkette ausgeben L: Linksbündig, C: Zentriert R: Rechtsbündig	ESC	Z	L	xx1	yy1	Text NUL	Eine Zeichenkette (...) an xx1,yy1 ausgegeben. 'NUL' (\$00), 'LF' (\$0A) oder 'CR' (\$0D) = Zeichenkettenende; Mehrere Zeilen werden durch das Zeichen ' ' (\$7C) getrennt;	
			C			...	Texte die zwischen zwei '-' (\$7E) Zeichen stehen blinken An/Aus; Texte die zwischen zwei '@' (\$40) Zeichen stehen blinken Invertierend;	
Font einstellen	ESC	Z	R	n1			Font mit der Nummer n1 (1..16) einstellen	
Font-Zoomfaktor			Z	n1	n2		n1 = X-Zoomfaktor (1x..8x); n2 = Y-Zoomfaktor (1x..8x)	
zus. Zeilenabstand			Y	n1			zwischen zwei Textzeilen n1 Pixel als zusätzlichen Zeilenabstand einfügen	
Text-Winkel			W	n1			Text-Ausgabewinkel: n1=0: 0°; n1=1: 90°; n1=2: 180°; n1=3: 270°;	
Text-Verknüpfungsmodus			V	n1			Modus n1: 1=setzen; 2=löschen; 3=invers; 4=Replace; 5=Invers Replace;	
Text-Muster			M	n1			Text mit Muster Nr. n1 verknüpfen; 0 = Text mit keinem Muster verknüpfen	
Zeichenkette für Terminal			ESC	Z	T		Text ...	Befehl um eine Zeichenkette in einem Makro an das Terminal ausgeben zu können
<b>Geraden und Punkte zeichnen</b>								
Rechteck zeichnen	ESC	G	R	xx1	yy1	xx2 yy2	Vier Geraden als Rechteck von xx1,yy1 nach xx2,yy2 zeichnen	
Gerade zeichnen			D	xx1	yy1	xx2 yy2	Eine Gerade von xx1,yy1 nach xx2,yy2 zeichnen	
Gerade weiter zeichnen			W	xx1	yy1		Eine Gerade vom letzten Endpunkt bis xx1, yy1 zeichnen	
Punkt zeichnen			P	xx1	yy1		Ein Punkt an die Koordinaten xx1, yy1 setzen	
Punktgröße / Liniendicke			Z	n1	n2		n1 = X-Punktgröße (1..15); n2 = Y-Punktgröße (1..15);	
Verknüpfungsmodus			V	n1			Zeichenmodus einstellen n1: 1=setzen; 2=löschen; 3=invers;	
Muster			M	n1			Geraden-/Punkt-Muster Nr. n1 einstellen; 0 = kein Muster verwenden	
<b>Rechteckige Bereiche verändern / zeichnen</b>								
Bereich löschen	ESC	R	L	xx1	yy1	xx2 yy2	Einen Bereich von xx1,yy1 nach xx2,yy2 löschen (alle Pixel aus)	
Bereich invertieren			I	xx1	yy1	xx2 yy2	Einen Bereich von xx1,yy1 nach xx2,yy2 invertieren (alle Pixel umkehren)	
Bereich füllen			S	xx1	yy1	xx2 yy2	Einen Bereich von xx1,yy1 nach xx2,yy2 füllen (alle Pixel ein)	
Bereich m. Füllmuster			M	xx1	yy1	xx2 yy2	n1	Einen Bereich von xx1,yy1 nach xx2,yy2 mit Muster n1 zeichnen (immer setzen)
Box zeichnen			O	xx1	yy1	xx2 yy2	n1	Ein Rechteck von xx1,yy1 nach xx2,yy2 mit Füllmuster n1 zeichnen; (immer Replace)
Rahmen zeichnen			R	xx1	yy1	xx2 yy2	n1	Einen Rahmen Typ n1 von xx1,yy1 nach xx2,yy2 zeichnen (immer setzen)
Rahmenbox zeichnen			T	xx1	yy1	xx2 yy2	n1	Eine Rahmenbox Typ n1 von xx1,yy1 nach xx2,yy2 zeichnen; (immer Replace)
<b>Bitmap Bilder Befehle</b>								
Bild aus Clipboard	ESC	U	C	xx1	yy1		Der akt. Clipboardinhalt wird mit allen Bildattributen nach xx1,yy1 geladen	
internes Bild laden			I	xx1	yy1	nr	internes Bild mit der nr (0..255) aus dem EEPROM nach xx1,yy1 laden	
Bild laden			L	xx1	yy1	daten ...	Ein Bild nach xx1,yy1 laden; daten des Bildes siehe Bildaufbau	
Bild-Zoomfaktor			Z	n1	n2		n1 = X-Zoomfaktor (1x..8x); n2 = Y-Zoomfaktor (1x..8x)	
Bild-Winkel			W	n1			Ausgabewinkel: n1=0: 0°; n1=1: 90°; n1=2: 180°; n1=3: 270°;	
Bild-Verknüpfungsmodus			V	n1			Modus n1: 1=setzen; 2=löschen; 3=invers; 4=Replace; 5=Invers Replace;	
Bild-Muster			M	n1			Bild mit Muster Nr. n1 verknüpfen; 0 = Bild mit keinem Muster verknüpfen	
Bild-Blinkattribut	B	n1			n1=0 Bild-Attribut blinken Aus; n1=1 Bild blinkt An/Aus; n1=2 Bild blinkt Invertierend;			
Hardcopy senden	ESC	H				Es wird ein komplettes Bild im Windows BMP-Format angefordert. Zuerst wird der Bild- header und dann die eigentlichen Bilddaten über RS232 gesendet (9662 Bytes).		
<b>Display-Befehle (Wirkung auf das gesamte Display)</b>								
Display löschen	ESC	D	L				Displayinhalt löschen (alle Pixel aus)	
Display invertieren			I				Displayinhalt invertieren (alle Pixel umkehren)	
Display füllen			S				Displayinhalt füllen (alle Pixel ein)	
Display ausschalten			A				Displayinhalt wird unsichtbar bleibt aber erhalten, Befehle weiterhin möglich	
Display einschalten			E				Displayinhalt wird wieder sichtbar	
Display Update			U	n1				n1=0: Displayausgaben sind nicht mehr sichtbar (werden aber weiterhin ausgeführt) n1=1: Displayausgaben sind sofort sichtbar n1=2: Displayinhalt auffrischen (vorherige Ausgaben werden sichtbar)
								1
<b>Makro Befehle</b>								
Makro ausführen	ESC	M	N	n1			Das (Normal-)Makro mit der Nummer n1 (0..255) aufrufen (max. 7 Ebenen)	
Touch Makro ausführen			T	n1			Das Touch-Makro mit der Nummer n1 (0..255) aufrufen (max. 7 Ebenen)	
Port Makro ausführen			P	n1			Das Port-Makro mit der Nummer n1 (0..255) aufrufen (max. 7 Ebenen)	
Bit Makro ausführen			B	n1			Das Bit-Makro mit der Nummer n1 (1..16) aufrufen (max. 7 Ebenen)	
Menü Makro ausführen			M	n1			Das Menü-Makro mit der Nummer n1 (0..255) aufrufen (max. 7 Ebenen)	
Matrix Makro ausführen			X	n1			Das Matrix-Makro mit der Nummer n1 (0..64) aufrufen (max. 7 Ebenen)	
autom. Makro zyklisch			A	n1	n2	n3		Makros n1..n2 automatisch zyklisch abarbeiten; n3=Pause in 1/10s
autom. Makro pingpong	J	n1	n2	n3		Makros autom. von n1..n2..n1 (PingPong) abarbeiten; n3=Pause in 1/10s		
<b>Blinkbereichs-Befehle</b>								
Blinkbereich definieren	ESC	Q	B	xx1	yy1	xx2 yy2	Definiert einen Blinkbereich (An/Aus) von xx1,yy1 bis xx2,yy2	
Invertierender Blinkbereich			I	xx1	yy1	xx2 yy2	Definiert einen invertierenden Blinkbereich von xx1,yy1 bis xx2,yy2	
Blinkattribut löschen			L	xx1	yy1	xx2 yy2	Löscht das Blinkattribut von xx1,yy1 bis xx2,yy2	
Blinkzeit einstellen			Z	n1			Einstellen der Blinkzeit n1= 1..15 in 1/10s; 0=Blinkfunktion deaktivieren	

### GRUNDEINSTELLUNGEN

Nach dem Einschalten bzw. nach einem Reset werden einige Funktionen auf einen bestimmten Wert voreingestellt (siehe letzte Spalte 'nach Reset' in der Tabelle). Beachten Sie bitte, daß alle Einstellungen durch Erstellen eines Power-On-Makros überschrieben werden können.

EA KIT320: Befehlstabelle 2										nach				
Befehl	Codes			Anmerkung						Reset				
<b>Bargraph Befehle</b>														
Bargraph definieren	ESC	B	R L O U	n1	xx1	yy1	xx2	yy2	aw	ew	typ	mst	Bargraph nach L(inks), R(rechts), O(ben), U(nten) mit der Nr. n1 definieren. xx1,yy1,xx2,yy2 sind das umschließende Rechteck des Bargraphs. aw,ew sind die Werte für 0% und 100%. typ=0: Balken; typ=1: Balken im Rechteck; mst=Balkenmuster; typ=2: Strich; typ=3: Strich im Rechteck; mst= Strichbreite	n1 kein Bar definiert
Bargraph aktualisieren			A	n1	wert	Den Bargraph mit der Nummer n1 auf den neuen Benutzer-'wert' setzen und zeichnen.								
Bargraph neu zeichnen			Z	n1	Den Bargraph mit der Nummer n1 komplett neu zeichnen									
Bargraphwert senden			S	n1	Den aktuellen Wert des Bargraph Nr. n1 auf der serilen Schnittstelle senden									
<b>Clipboard Befehle (Zwischenspeicher für Bildbereiche)</b>														
Clipboard Nr. wählen	ESC	C	N	n1	2 Clipboards stehen zur Verfügung, mit n1= (1,2) wird das akt. Clipboard ausgewählt.						n1, leer			
Displayinhalt sichern			B	Der gesamte Displayinhalt wird als Bildbereich ins Clipboard kopiert										
Bereich sichern			S	xx1	yy1	xx2	yy2	Der Bildbereich von xx1,yy1 bis nach xx2,yy2 wird ins Clipboard kopiert						
Bereich restaurieren			R	Der Bildbereich im Clipboard wird wieder ins Display an die ursprüngliche Stelle kopiert										
Bereich kopieren			K	xx1	yy1	Der Bildbereich im Clipboard wird ins Display nach xx1,yy1 kopiert								
<b>Menü / Popup Befehle</b>														
Menü definieren und Darstellen	ESC	N	D	xx1	yy1	nr	Text ...	NUL	Ein Menü wird ab der Ecke xx1,yy1 mit dem akt. Menüfont gezeichnet. nr:= aktuell invertierter Eintrag (z.B: 1 = 1. Eintrag) Text:= Zeichenkette mit den Menüeinträgen. Die einzelnen Einträge sind durch Zeichen " " (\$7C,dez:124) getrennt z.B. "Eintrag1 Eintrag2 Eintrag3" Der Hintergrund des Menüs wird automatisch gesichert. Ist bereits ein Menü definiert, wird dieses automatisch abgebrochen+entfernt.					
nächster Eintrag vorheriger Eintrag			N P	Der nächste Eintrag wird invertiert oder bleibt am Ende stehen Der vorherige Eintrag wird invertiert oder bleibt am Anfang stehen										
Menüende / Senden			S	Das Menü wird vom Display entfernt und durch den ursprünglichen Hintergrund ersetzt der aktuelle Eintrag wird als Nummer (1..n) gesendet (0=kein Menü dargestellt)										
Menüende / Makro			M	n1	Das Menü wird vom Display entfernt und durch den ursprünglichen Hintergrund ersetzt. Für Eintrag 1 wird das Menü-Makro n1 aufgerufen, für Eintrag 2 Menü-Makro nr+1 usw.									
Menüende / Abbrechen			A	Das Menü wird vom Display entfernt und durch den ursprünglichen Hintergrund ersetzt										
Menu-Font einstellen			F	n1	Font mit der Nummer n1 (1..16) für Menüdarstellung einstellen								5	
Menüfont-Zoomfaktor zus. Zeilenabstand			Z Y	n1	n2	n1 = X-Zoomfaktor (1x..8x); n2 = Y-Zoomfaktor (1x..8x) zwischen zwei Menüeinträgen n1 Pixel als zusätzlichen Zeilenabstand einfügen							1,1	
Menu-Winkel			W	n1	Menüdarstellung Winkel: n1=0: 0°; n1=1: 90°; n1=2: 180°; n1=3: 270°;								0	
<b>Port-Befehle</b>														
Ausgabe-Port schreiben	ESC	Y	W	n1	n2	n1=0: Alle 8 Ausgabe-Ports entsprechend n2 (=8-Bit Binärwert) einstellen n1=1..8: Ausgabe-Port n1 rücksetzen (n2=0); setzen (n2=1); invertieren (n2=2)					Ports 1-8=0			
Eingabe-Port lesen			R	n1	n1=0: Alle 8 Eingabe-Ports als 8-Bit Binärwert einlesen n1=1..8: Eingabe-Port <n1> einlesen (1=H-Pegel=5V, 0=L-Pegel=0V)									
Port Scan Ein/Aus			A	n1	Der automatische Scan des Eingabe-Port wird n1=0: deaktiviert; n1=1: aktiviert								1	
Eingabe-Port invers			I	n1	Der Eingabe-Port wird n1=0: normal; n1=1: invertiert ausgewertet								0	
Matrix-Tastatur			M	n1	n2	n3	Festlegung einer externen Matrix-Tastatur an den Ein- und Ausgängen n1=Anzahl Eingänge (1..8); n2=Anzahl Ausgänge (0..8); n3= Entprellung (0..7)					0		
Beleuchtung Ein/Aus/Halb			L	n1	CFL/LED-Beleuchtung n1=0: AUS; n1=1: EIN; n1=2: Halbe Helligkeit;								1	
<b>Sonstige-Befehle</b>														
Warten (Pause)	ESC	X	n1	n1 Zehntel-Sekunden abwarten bevor der nächste Befehl ausgeführt wird.										
Summer Ein / Aus	ESC	J	n1	n1=1..255: Summer für n1 1/10s lang Ein							AUS			
Bytes senden	ESC	S	B	anz	daten ...					Es werden anz (=1..255) Bytes auf der RS-232/RS-422 gesendet daten ... = anz Bytes (z.B Ansteuerung eines externen seriellen Druckers)				
Analogwert senden	ESC	S	D	Der aktuelle Wert vom Analogeingang AIN wird auf der RS-232/RS-422 gesendet										
Version senden	ESC	S	V	Es wird die Software Versions-Nr. + Datum als String auf der RS-232/RS-422 gesendet										
Befehle an HD44780 *)	ESC	L	B	anz	daten ...					Es werden anz (=1..255) Befehle an das ext. Dotmatrixmodul mit HD44780 gesendet.				
Daten an HD44780 *)			D	anz	daten ...					Es werden anz (=1..255) Daten an das ext. Dotmatrixmodul mit HD44780 gesendet				
EEPROM lesen	ESC	E	R	adr	anz	Vom internen Benutzer EEPROM ab Adresse adr werden anz (=1..255) Bytes angefordert und über die RS-232/RS-422 gesendet.								
EEPROM schreiben			W	adr	anz	daten ...					In das interne Benutzer EEPROM ab Adresse adr werden anz (=1..255) Bytes geschrieben. daten ... = anz Bytes			
I2C-Bus lesen	ESC	I	R	adr	anz	Von dem Baustein am I2C-Bus mit der Device Adresse adr werden anz (=1..255) Bytes angefordert und über die RS-232/RS-422 gesendet.								
I2C-Bus schreiben			W	adr	anz	daten ...					Auf dem I2C-Bus für den Baustein mit der Device Adresse adr werden anz (=1..255) Bytes gesendet. daten ... = anz Bytes			

\*) Befehl erst ab Version 1.1 vorhanden

EA KIT320: Befehle für das Touch-Panel										nach Reset				
Befehl	Codes		Anmerkung											
<b>Touch: Bereiche definieren</b>														
Touch-Taste definieren  (Taste ist gedrückt solange der Touch berührt wird)	ESC	A	C	f1	f2	down Code	up Code	Text ...	NUL	'C': Die Touch-Felder f1 bis f2 werden zu einer Taste definiert. 'T': Der Bereich von xx1,yy1 nach xx2,yy2 wird als Taste definiert. 'U': Das Bild Nr. n1 wird nach xx1,yy2 geladen und als Taste definiert. 'down Code':(1-255) Rückgabe/Touchmakro beim Drücken der Taste. 'up Code': (1-255) Rückgabe/Touchmakro beim Loslassen der Taste. (down-/up-Code = 0 drücken/loslassen wird nicht gemeldet). 'Text': es folgt eine Zeichenkette die zentriert mit dem akt. Touch-Font in der Touch-Taste platziert wird, mehrzeilige Texte werden mit dem Zeichen ' ' (\$7C, dez: 124) getrennt; 'NUL': (\$00) = Zeichenkettenende				
			T	xx1	yy1	xx2	yy2	down Code	up Code			Text ...	NUL	
			U	xx1	yy1	n1	down Code	up Code	Text ...			NUL		
Touch-Schalter definieren  (Zustand der Schalter toggelt nach jeder Berührung)	ESC	A	G	f1	f2	down Code	up Code	Text ...	NUL	'G': Die Touch-Felder f1 bis f2 werden zu einem Schalter definiert. 'K': Der Bereich von xx1,yy1 nach xx2,yy2 wird als Schalter definiert. 'J': Das Bild n1 wird nach xx1,yy2 geladen und als Schalter definiert. 'down Code': (1-255) Rückgabe/Touchmakro beim Einschalten. 'up Code': (1-255) Rückgabe/Touchmakro beim Ausschalten. (down-/up-Code = 0 Ein-/Ausschalten wird nicht gemeldet). 'Text': es folgt eine Zeichenkette die zentriert mit dem akt. Touch-Font in der Touch-Taste platziert wird, mehrzeilige Texte werden mit dem Zeichen ' ' (\$7C, dez: 124) getrennt; 'NUL': (\$00) = Zeichenkettenende				
			K	xx1	yy1	xx2	yy2	down Code	up Code			Text ...	NUL	
			J	xx1	yy1	n1	down Code	up Code	Text ...			NUL		
Touch-Taste mit Menüfunktion definieren	ESC	A	M	xx1	yy1	xx2	yy2	down Code	up Code	mnu Code	Text ...	NUL	Der Bereich xx1,yy1 nach xx2,yy2 wird als Menü-Taste definiert. 'down Code':(1-255)Rückgabe/Touchmakro beim Drücken. 'up Code':(1-255) Rückgabe/Touchmakro beim Menü-Abbruch 'mnu Code':(1-255) Rückgabe/Menümakro+(EintragsNr-1) nach Auswahl eines Menü-Eintrages. (down-/up-Code = 0 Aktivieren / Abbruch des Menüs wird nicht gemeldet). 'Text':= Zeichenkette mit dem Menü-Tastentext und den Menüeinträgen. Die einzelnen Einträge sind durch Zeichen ' ' (\$7C,dez:124) getrennt z.B. "Taste Eintrag1 Eintrag2 Eintrag3" Der Tastentext wird mit dem akt. Touchfont und die Menü-Einträge mit dem akt. Menüfont gezeichnet. Der Hintergrund des Menüs wird automatisch gesichert.	
Zeichenbereich definieren	ESC	A	D	xx1	yy1	xx2	yy2	n1	Ein Zeichenbereich wird definiert. Innerhalb der Eck-Koordinaten xx1,yy1 und xx2,yy2 kann dann mit der Strichstärke n1 gezeichnet werden.					
Freien Touchbereich def.*)	ESC	A	H	xx1	yy1	xx2	yy2	Ein frei benutzbarer Touchbereich wird definiert. Touchaktionen (down, up und drag) innerhalb der Eck-Koordinaten xx1,yy1 und xx2,yy2 werden über RS232 gesendet.						
Bar per Touch einstellbar	ESC	A	B	nr	Der Bargraph mit der Nr. n1 wird zur Eingabe per Touchpanel definiert.									
<b>Touch: Einstellungen</b>														
Touch- Rahmen	ESC	A	E	n1	mit n1 wird der Rahmentyp für die Darstellung von Touch-Tasten/Schaltern eingestellt		1							
Touch-Tasten Reaktion			I	n1	automatisches Invertieren beim Berühren der Touch-Taste: n1=0=AUS; n1=1=EIN;		1							
Touch-Taste Invertieren			S	n1	Summer piepst kurz beim Berühren einer Touch-Taste: n1=0=AUS; n1=1=EIN		1							
Touch-Schalter abfragen			N	Code	Die Touch-Taste mit dem zugeordnetem Return-Code wird manuell Invertiert									
Touch-Schalter einstellen			X	Code	Der Zustand des Schalters (Aus=0; Ein=1) wird über die serielle Schnittstelle gesendet.									
Touch-Bereich Löschen			P	Code	n1	Der Zustand des Schalters wird per Befehl geändert n1=0=Aus; n1=1=Ein.								
Barwert senden Ein/Aus			L	Code	n1	Der Touchbereich mit dem Return-Code (Code=0: alle Touchbereiche) wird aus der Touchabfrage entfernt. Mit n1=0 bleibt der Bereich am Display sichtbar, mit n1=1 wird der Bereich im Display gelöscht.								
Touch-Abfrage Ein/Aus			Q	n1	das automatische Senden eines neuen Bargraphwertes per Topucheingabe wird n1=0:deaktiviert; n1=1:aktiviert;		1							
	A	n1	Touchabfrage wird n1=0:deaktiviert; n1=1:aktiviert;		1									
<b>Touch: Beschriftungs-Font</b>														
Beschriftungs Font	ESC	A	F	n1	Font mit der Nummer n1 (1..16) für Touchtastenbeschriftung einstellen		5							
Beschriftungs-Zoomfaktor			Z	n1	n2	n1 = X-Zoomfaktor (1x..8x); n2 = Y-Zoomfaktor (1x..8x)		1,1						
zus. Zeilenabstand			Y	n1	zwischen zwei Textzeilen n1 Pixel als zusätzlichen Zeilenabstand einfügen									
Beschriftungs-Winkel			W	n1	Text-Ausgabewinkel: n1=0: 0°; n1=1: 90°; n1=2: 180°; n1=3: 270°;		0							

*\*) Befehl erst ab Version 1.1 vorhanden*

EA KIT320: Befehlstabelle für den Terminal Betrieb										nach	
Befehl	Codes					Anmerkung				Reset	
<b>Befehle für den Terminal Betrieb</b>											
FF: Formfeed (dez:12)	^L						Der Terminalinhalt wird gelöscht und der Cursor nach Pos. (1,1) gesetzt				
CR: Carriage Return (d:13)	^M						Cursor ganz nach links zum Zeilenanfang				
LF: Linefeed (dez:10)	^J						Cursor wird in die nächste Zeile gesetzt				
Cursor positionieren	ESC	T	P	n1	n2		n1=Spalte; n2=Zeile; Ursprung links oben ist (1,1)			1,1	
Cursor On / Off			C	n1			n1=0: Cursor ist unsichtbar; n1=1: Cursor blinkt;			1	
Terminal-Modus			M	n1			n1=0: Clear-Modus; n1=1: Overwrite-Modus; n1=2: Scroll-Modus			2	
Autom. Zeilenumbruch			Z	n1			Der automatische Zeilenumbruch wird n1=1 Ein- oder n1=0 Ausgeschaltet			1	
Terminal unsichtbar			A					Terminal Anzeige ist nicht sichtbar; Ausgaben werden weiterhin ausgeführt			Ein
Terminal sichtbar	E					Terminal Anzeige ist wieder sichtbar;					
<b>Terminal-Ausgaben umleiten</b>											
Terminal unterdrücken	ESC	T	N			ASCII-Zeich., FF, CR, LF werden unterdrückt. Befehle (ESC T) werden ausgeführt					
Terminalausgabe intern			I			Alle Terminal-Ausgaben/Befehle wirken auf das interne Terminal des EA KIT320				intern	
Terminalausgabe extern			X			Alle Terminal-Ausgaben/Befehle wirken auf das externe Dotmatrix Modul					
<b>Einstellungen für das interne Terminal</b>											
Font einstellen	ESC	T	F	n1			Font Nr. n1 (1..16) für Terminal Betrieb einstellen. (nur monospaced Fonts)		5		
zus. Zeilenabstand			Y	n1			für den aktuellen Font werden zusätzlich n1 Pixel als Zeilenabstand definiert				
Fenster definieren			W	xx1	yy1	xx2	yy2	w	Die Terminal Ausgabe erfolgt nur innerhalb des Fensters xx1,yy1 (=linke obere Ecke) und xx2,yy2 (=rechte untere Ecke); xx=0..319; yy=0..239; w=Winkel (0=0°; 1=90°; 2=180°; 3=270°) der Terminaldarstellung		0..319 0..239
<b>Einstellungen für das externe Dotmatrix-Modul (optional an J6 oder J7)</b>											
Dotmatrixmodul initialisieren	ESC	T	D	n1	n2			Ein externes Dotmatrix-Display (HD44780 kompatibel) initialisieren n1 = anzahl Zeichen; n2 = anzahl Zeilen			

Antworten des EA KIT320 über die serielle Schnittstelle										
Kennung	anz	daten				Anmerkung				
<b>automatische Antworten vom KIT320</b>										
ESC	A	1	code			Antwort vom Analogen Touchpanel wenn eine Taste/Schalter gedrückt wurde. code = down oder up Code der Taste/Schalter. Es wird nur gesendet wenn kein Touch-Makro mit der Nr. code definiert ist !				
ESC	N	1	code			Nach dem Auswählen eines Menüeintrages per Touch wird der ausgewählte Menüeintrag code gesendet. Es wird nur gesendet wenn kein Menü-Makro mit der Nr. code definiert ist !				
ESC	P	1	wert			Nach Änderung des Eingangs-Port wird der neue 8-Bit Wert gesendet Automatischer Port-Scan muß aktiviert sein siehe Befehl 'ESC Y A n1' Es wird nur gesendet wenn kein Port-Makro mit der Nr. wert definiert ist !				
ESC	M	1	nr			Nach Erkennen eines Tastendruckes der extern Matrix-Tastatur wird die neu gedrückte Tastennummer nr gesendet. Es wird nur gesendet wenn kein Matrix-Makro mit der Nr. nr definiert ist !				
ESC	B	2	nr	wert			Nach dem Einstellen eines Bargraph per Touch wird der aktuelle wert des Bars mit der nr gesendet. Barwert Senden muß aktiviert sein siehe Befehl 'ESC A Q n1'.			
ESC	H	5	typ	xLO	xHI	yLO	yHI	*) Bei einem freien Touchbereich-Ereignis wird folgendes gesendet: typ=0 ist Loslassen; typ=1 ist Berühren; typ=2 ist Draggen innerhalb des freien Touchbereiches an den Koordinaten x,y (16-Bit Werte)		
<b>Antworten nur nach Anforderung per Befehl</b>										
ESC	N	1	nr			Nach dem Befehl 'ESC N S' wird der aktuell ausgewählte Menüeintrag gesendet. nr=0: kein Menüeintrag ist ausgewählt.				
ESC	B	2	nr	wert			Nach dem Befehl 'ESC B S n1' wird der aktuelle Wert Bars mit der Nr. nr gesendet.			
ESC	X	2	code	wert			Nach dem Befehl 'ESC A X' wird der aktuelle Zustand des Touch-Schalters mit dem Return-Code code gesendet. wert = 0 oder 1			
ESC	Y	2	nr	wert			Nach dem Befehl 'ESC Y R' wird der angeforderte Eingangs-Port gesendet nr=0: wert ist ein 8-Bit Binärwert aller 8 Eingänge nr=1..8: wert ist 0 oder 1 je nach Zustand des Eingangs nr			
ESC	D	2	LO-byt wert	HI-byt wert			Nach dem Befehl 'ESC S D' wird der aktuelle Analogwert (max. 1/2 VDD) vom Eingang AIN gesendet. (wert = 0..1023 entspricht ca. 0..2,5V)			
ESC	E	anz	daten ...				Nach dem Befehl 'ESC E R adr anz' werden die angeforderten Bytes aus dem Benutzer EEPROM gesendet.			
ESC	I	anz	daten ...				Nach dem Befehl 'ESC I R adr anz' werden die angeforderten Bytes I2C-Bus gesendet.			
<b>Antworten ohne ESC und Längenangabe (anz)</b>										
B	M	+ 9660 Bytes Bilddaten				Nach dem Befehl 'ESC H' werden 9662 Bytes (=320x240 BMP-Bild) gesendet. Die ersten beiden Bytes des BMP-Bildes beginen immer mit 'BM'				
E	A	Zeichenkette ..		NUL		Nach dem Befehl 'ESC S V' wird die Version der KIT-Firmware als Zeichenkette gesendet (Endekennung ist das Zeichen NUL = \$00). Die ersten beiden Bytes der Zeichenkette beginen immer mit 'EA'				

### TERMINAL-BETRIEB

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

Das Terminal besitzt eine eigene Ebene zur Darstellung und ist somit völlig unabhängig von den Grafikausgaben; zudem kann die Grösse des benutzbaren Terminalfensters frei definiert werden. Wird z.B. der Grafikbildschirm mit 'ESC DL' gelöscht, so beeinflusst das nicht den Inhalt des Terminalfensters (die Terminalebene wird mit 'FF' gelöscht).

Das Zeichen '#' wird als Escape-Zeichen benutzt (siehe unten) und ist somit nicht direkt im Terminal darstellbar. Soll das Zeichen '#' im Terminal ausgegeben werden, so muß es doppelt gesendet werden '##'.

### BEFEHLSÜBERGABE/PARAMETER

Die Bedieneinheit läßt sich über diverse eingebaute Befehle programmieren. Jeder Befehl beginnt mit ESCAPE gefolgt von einem oder zwei Befehlsbuchstaben und einigen Parametern.

Es gibt zwei Möglichkeiten Befehle zu senden:

#### 1. ASCII-Modus

- Das Escape-Zeichen entspricht dem Zeichen '#' (hex: \$23, dez: 35).
- Die Befehlsbuchstaben folgen direkt im Anschluss an das '#' Zeichen.
- Die Parameter werden im Klartext (mehrere ASCII Ziffern) mit einem nachfolgenden Trennzeichen (z.B. das Komma ',') gesendet.
- Zeichenketten (Texte) werden direkt ohne Anführungsstrichen geschrieben und mit CR (hex: \$0D), oder LF (hex: \$0A) abgeschlossen.

#### 2. Binär-Modus

- Das Escape-Zeichen entspricht dem Zeichen ESC (hex: \$1B, dez: 27).
- Die Befehlsbuchstaben werden direkt gesendet.
- Die Koodinaten xx und yy werden als 16-Bit Binärwerte (zuerst das LOW-Byte dann das HIGH-Byte) gesendet.
- Alle anderen Parameter werden als 8-Bit Binärwert (1 Byte) gesendet.
- Zeichenketten (Texte) werden mit CR (hex: \$0D), LF (hex: \$0A) oder NUL (hex: \$00) abgeschlossen.

Im Binär-Modus dürfen keine Trennzeichen z.B. Leerzeichen oder Kommas verwendet werden. Die Befehle benötigen auch **kein Abschlussbyte** wie z.B. Carrige Return (außer Zeichenkette: \$00).

## ELECTRONIC ASSEMBLY

### PROGRAMMIERBEISPIEL

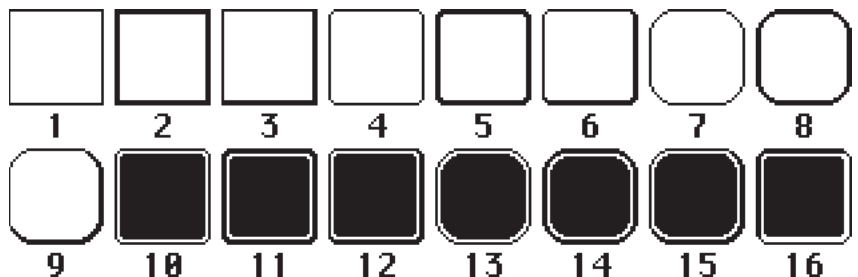
In dem nachfolgenden Beispiel ist zu sehen wie die Zeichenkette "Test" linksbündig an den Koordinaten 117,32 ausgegeben werden kann.

Beispiel	Auszugebende Codes im ASCII-Modus	
für Terminal	#ZL117, 32,Test	<Return>
für Turbo-Pascal	write(aux, '#ZL117, 32,Test', chr(13) );	
für 'C'	fprintf(stdaux, "#ZL%d,%d,%s\x0D", 117, 32, "Test");	
für Q-Basic	OPEN "COM1:9600,N,8,1,BIN" FOR RANDOM AS #1 PRINT #1,"#ZL117,32,Test"+CHR\$(13)	

Beispiel	Auszugebende Codes im Binär-Modus											
in ASCII	ESC	Z	L	u	NUL	space	NUL	T	e	s	t	NUL
in Hex	\$1B	\$5A	\$4C	\$75	\$00	\$20	\$00	\$54	\$65	\$73	\$74	\$00
in Dezimal	27	90	76	117	0	32	0	84	101	115	116	0
für Turbo-Pascal	write(aux, chr(27), 'Z', 'L', chr(117), chr(0), chr(32), chr(0), 'Test', chr(0));											
für 'C'	fprintf(stdaux, "\x1BZL%c%c%c%c%c%s\x00", 117, 0, 32, 0, "Test");											
für Q-Basic	OPEN "COM1:9600,N,8,1,BIN" FOR RANDOM AS #1 PRINT #1,CHR\$(27)+"ZL"+CHR\$(117)+CHR\$(0)+CHR\$(32)+CHR\$(0)+"Test"+CHR\$(0)											

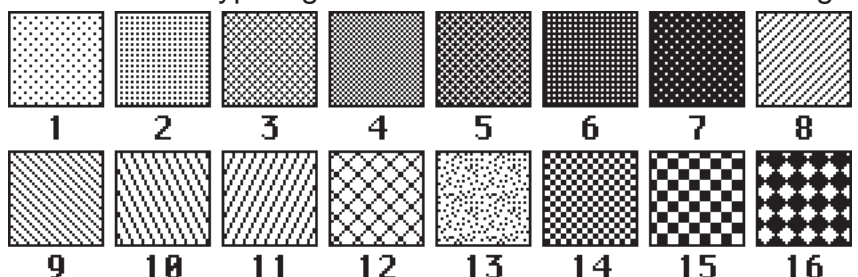
### RAHMEN UND TASTENFORMEN

Mit den Befehlen *Rahmen /Rahmenbox zeichnen* sowie beim Zeichnen von Touchtasten kann ein Rahmentyp eingestellt werden. Es stehen dabei 16 interne Rahmentypen zur Verfügung; zudem können eigene Rahmentypen über den Kitcompiler eingebunden werden.



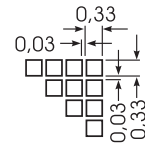
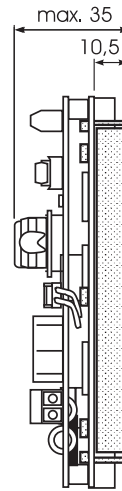
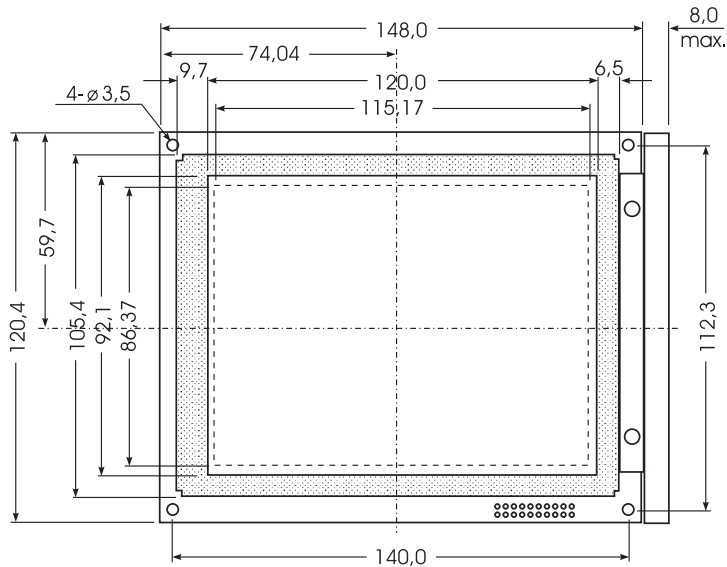
### MUSTER

Bei diversen Befehlen kann als Parameter ein Mustertyp eingestellt werden. So können rechteckige Bereiche, Bargraphs und sogar Texte mit unterschiedlichen Mustern gefüllt werden. Dabei stehen 16 interne Füllmuster zur Verfügung, zudem können eigene Füllmuster über den Kitcompiler eingebunden werden.



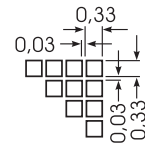
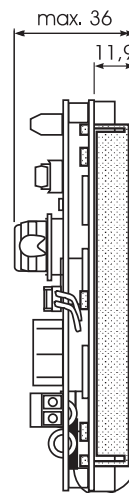
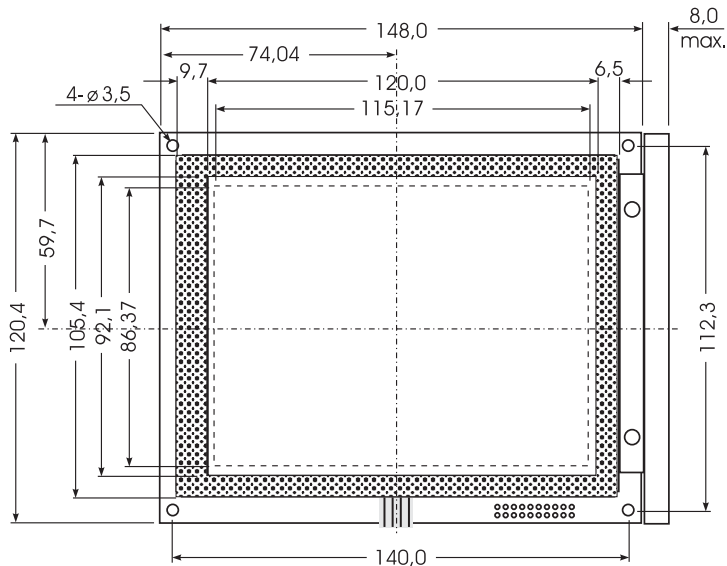
# EA KIT320-8

## ABMESSUNGEN OHNE TOUCH PANEL



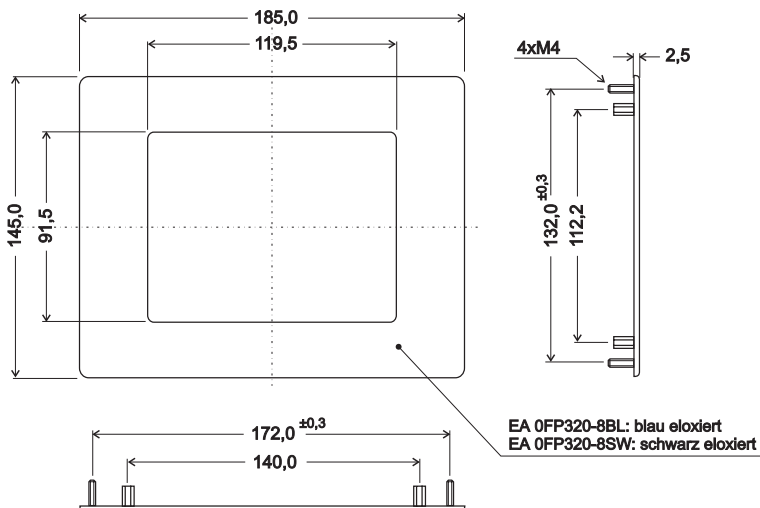
alle Maße in mm

## ABMESSUNGEN MIT TOUCH PANEL

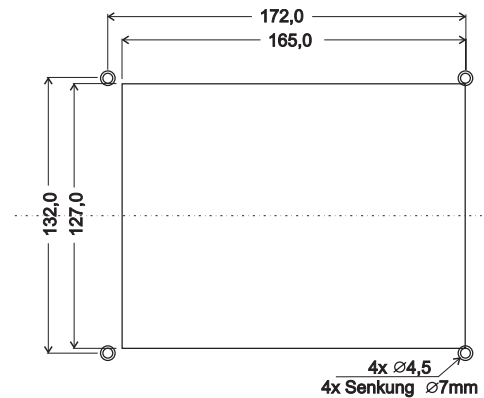


alle Maße in mm

## FRONTPANEL EA 0FP320-8



## PANEL CUT OUT



alle Maße in mm