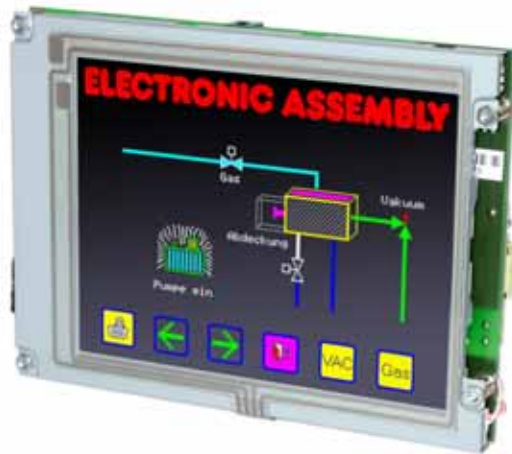


# 5,7" COLOR LCD-BEDIENEINHEIT INKL. TOUCH PANEL

mit Port für  
Dotmatrix LCD



EA KIT320F-8LWTP  
Abm. 155 x 115mm

## TECHNISCHE DATEN

- \* 5,7" COLOR LCD GRAFIKDISPLAY MIT DIVERSEN GRAFIKFUNKTIONEN UND FONTS
- \* 320x240 PIXEL, 16 FARBEN MIT CFL-BELEUCHTUNG
- \* FONT ZOOM VON ca. 2mm BIS ZU ca. 80mm, in 90° SCHRITTEN DREHBAR
- \* VERSORGUNG +5V±2% @ 320mA ODER +9..35V OPTIONAL
- \* RS-232 ODER OPTIONAL RS-422 MIT BAUDRATEN 1.200..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
- \* 1380 MAKROS PROGRAMMIERBAR (FLASH MIT 512kB)
- \* TEXT UND GRAFIK MISCHEN, BLINKATTRIBUTE, INVERS BLINKEN
- \* BELEUCHTUNG PER SOFTWARE SCHALTBAR (AUS, EIN, HALBE HELLEIGKEIT)
- \* ANALOGESTOUCH PANEL: VARIABLES RASTER MIT z.B. 10x8 FELDERN
- \* FREI DEFINIERBARE TASTEN UND SCHALTER
- \* MENÜS UND BARGRAPH PER TOUCH EINSTELLBAR
- \* 8 DIGITALE EIN- UND 8 DIGITALE AUSGÄNGE
- \* ZWEI ANALOGEINGÄNGE KOMFORTABEL PROGRAMMIERBAR
- \* DOTMATRIXDISPLAY ALS ZWEITANZEIGE DIREKT ANSCHLIESSBAR

## BESTELLBEZEICHNUNG

BEDIENEINHEIT COLOR 5,7", TOUCH PANEL, 320x240 DOTS, RS-232 **EA KIT320F-8LWTP**  
WIE VOR, JEDOCH OHNE TOUCH PANEL **EA KIT320F-8LW**

VERSORGUNG +9..35V = STATT +5V = **EA OPT-9/35V**

RS-422 SCHNITTSTELLE STATT RS-232 **EA OPT-RS4224**

OPTOKOPPLER ONBOARD FÜR 8 EIN- UND 8 AUSGÄNGE **EA OPT-OPTO16**

ALUMINIUM EINBAUBLENDE, MATT-SCHWARZ ELOXIERT **EA 0FP320F-8SW**

KABEL 1,5m FÜR ANSCHLUSS AN 9-POL. SUB-D (RS-232 FEMALE) **EA KV24-9B**

**ELECTRONIC  
ASSEMBLY** GMBH

ZEPPELINSTRASSE 19 · D-82205 GILCHING  
TEL 08105/778090 · FAX 08105/778099 · <http://www.lcd-module.de>

Documentation of revision				
Date	Type	Old	New	Reason / Description
5.2.04	Simulator	F1: help function	Shift-F1: help function	continous terminal functionality F1..F10
5.2.04	Data sheet		Page 6: Extension for analogue input description Page 22: Insert command for calibration	
6.2.04	Data sheet		Page 9: Insert "Create own key form", "Use bitmap as a key", "Radio groups"	
8.10.10	Display	EA KIT320F-8C EA KIT320F-8CTP	EA KIT320F-8LW EA KIT320F-8LWTP	Backlight changed from CFL into LED type

## INHALT

ALLGEMEINES .....	3
HARDWARE, STECKERBELEGUNG .....	4
EIN-, AUSGÄNGE UNDTASTATURANSCHLUSS .....	5~7
TOUCHPANEL .....	8, 9
MAKROPROGRAMMIERUNG .....	10, 11
ZEICHENSÄTZE .....	12, 13
FARBEN UNDFARBATTRIBUTE .....	14
BEFEHLSSYNTAX, PROGRAMMIERBEISPIEL .....	16
BEFEHLE INTABELLENFORM .....	17~22
RÜCKANTWORTEN DES BEDIENPANELS .....	23
ABMESSUNGEN .....	24

### ALLGEMEINES

EA KIT320F 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 1200 Baud bis zu 115200 Baud ausgewählt werden. Handshakeleitungen RTS und CTS stehen zur Verfügung.

Datenformat: 

### TOUCH PANEL

Die Version EA KIT320F-8LWTP ist 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 sind bereits 14 Zeichensätze fest vorhanden zudem können bis zu 10 eigene Zeichensätze eingebunden werden. Jeder Zeichensatz kann wiederum 2- bis 8-fach gezoomt werden. Mit dem größten Zeichensatz lassen sich somit bildschirmfüllende Worte und Zahlen darstellen.

### ZUBEHÖR

#### Kit-Compiler für Windows zur Makroerstellung

Zur Makroprogrammierung ist eine Software von ELECTRONIC ASSMBLY LCD-Tools<sup>\*)</sup> erhältlich (kostenfrei). Sie enthält einen Kit-Editor, Kit-Compiler, sowie Beispiele und Fonts (für PC-Win). Der Kit-Compiler übersetzt die in eine Textdatei eingegebenen Befehle in einen für die Bedieneinheit lesbaren Code und programmiert diesen dauerhaft ins FLASH PROM.

#### Simulator-Software

Für den ersten Kontakt mit dem Display empfehlen wir die Installation des Simulators auf einem Windows-PC. Alle Funktionen können so direkt am PC simuliert werden.

Auch später während der Softwareentwicklung ist der Simulator ein hilfreiches Utility, welches übrigens kostenfrei im Internet zur Verfügung steht.

#### 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> im Internet unter <http://www.lcd-module.de/deu/touch/touch.htm>

# EA KIT320F-8 COLOR

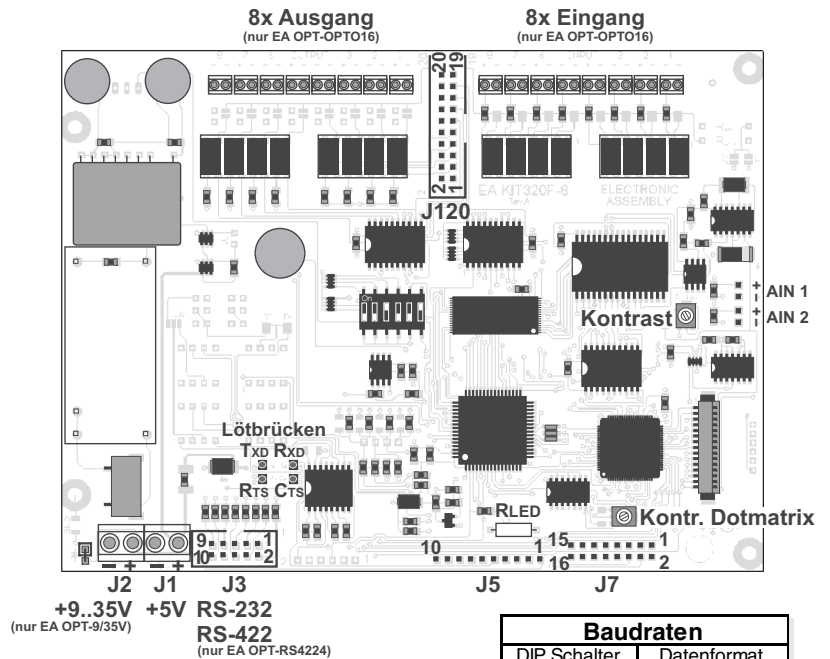
## ELECTRONIC ASSEMBLY

### 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.

Baudraten			
DIP Schalter			Datenformat
1	2	3	8,N,1
ON	ON	ON	1200
OFF	ON	ON	2400
ON	OFF	ON	4800
OFF	OFF	ON	9600
OFF	ON	OFF	19200
OFF	ON	OFF	38400
ON	OFF	OFF	57600
OFF	OFF	OFF	115200

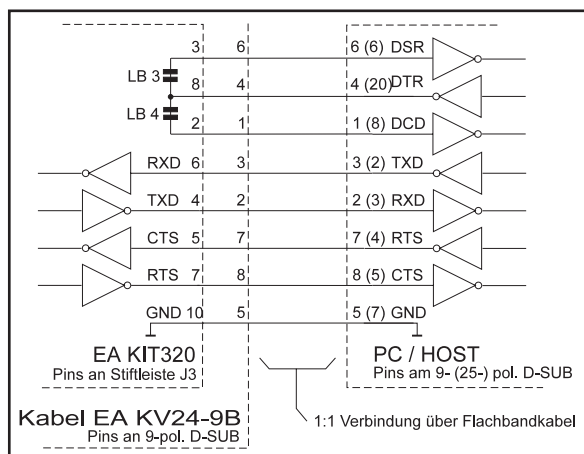
### RS-232/RS-422

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

An der Lötäugenleiste 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 TXD, RxD, RTS, CTS geöffnet werden!

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



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

Technische Änderungen vorbehalten. Wir übernehmen keine Haftung für Druckfehler.

### EIN-UND AUSGÄNGE

Alle Bedieneinheiten EA KIT320F-8 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 auto-matische 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 DISK320F.

**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.

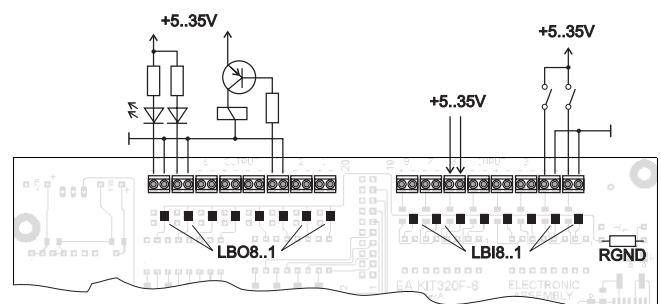
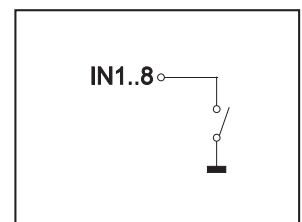
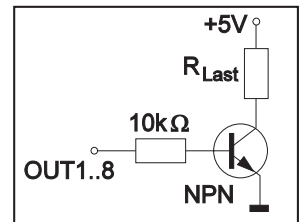
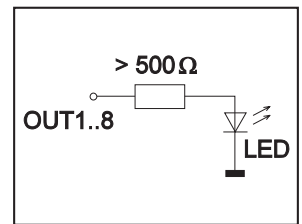
An allen 8 Eingängen können direkt Spannungen von 5..35V angelegt werden. Pegel über 4V werden als H-Pegel erkannt, Pegel unter 2V gelten als L-Pegel. Spannungen zwischen 2 und 4V sind undefiniert.

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 LBI1..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



### ANSCHLUSS FÜR DOTMATRIX-DISPLAY

An den Lötäugen J7 kann 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)

### ANALOGGEINGÄNGE AIN1 UND AIN2 SEITE 22

Zur Spannungsmessung stehen 2 Analogeingänge mit einer Eingangsempfindlichkeit von 0..+250mV zur Verfügung. Jeder Eingang hat einen Massebezug zu GND und einen Eingangswiderstand von 10k $\Omega$ .

Die Auflösung beträgt 10 Bit, was in etwa einem 3st. DVM entspricht. Die Grundgenauigkeit nach Abgleich liegt bei ca. 0,5%. Bitte beachten Sie, dass nur positive Spannungen angeschlossen werden dürfen!

#### Abgleich

Die Eingänge sind nicht abgeglichen. Eine Abgleichprozedur kann wie folgt aussehen:

- 1.) Anlegen einer definierten Spannung im Bereich von 150-250mV (Beispiel: 200mV, AIN1)
- 2.) Befehl zum Analogabgleich senden (siehe Seite 22). Im Beispiel: "ESC V @ 1 200.0 NUL".

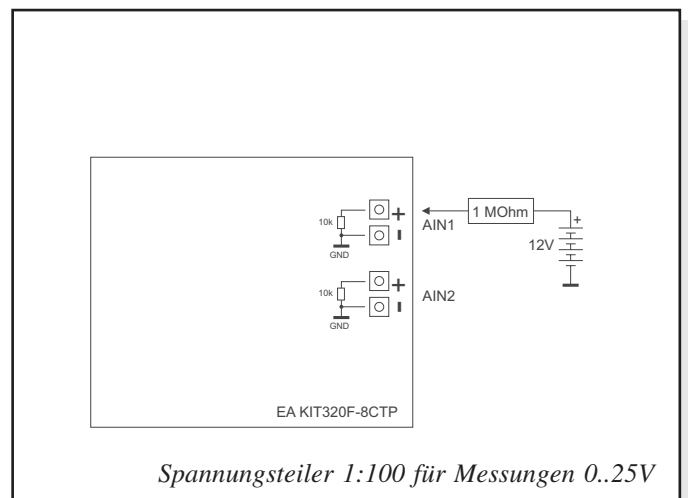
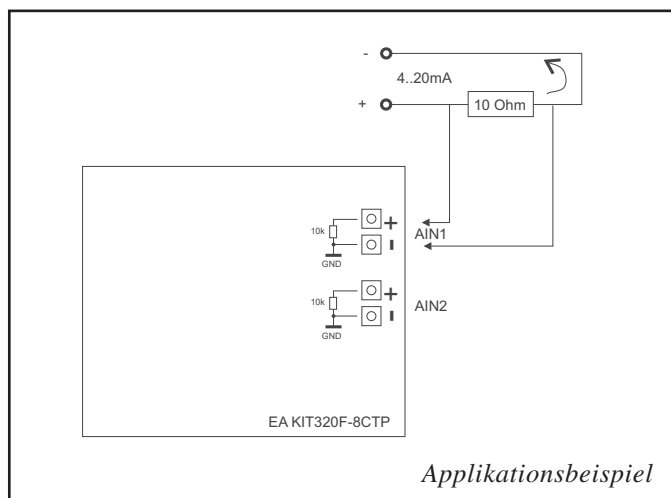
Die erwähnte Befehlsfolge kann natürlich auch in einem Touchmakro hinterlegt sein, welches auf Touchberührung ausgeführt wird.

#### Messungen

Die Messungen können gezielt über die RS-232/RS-422 angefordert oder auch direkt auf dem Display dargestellt werden (als Ziffernfolge oder Bargraph in unterschiedlichsten Größen und Farben).

Die direkte Darstellung der Messwerte erfolgt am einfachsten über ein Prozessmakro oder eines der Analogmakros (z.B. Ausführung bei jeder Änderung des Analogwertes an AIN1 bzw. AIN2, oder Ausführung bei Über- bzw. Unterschreiten eines Limits).

Für die direkte Darstellung am Display sind die Eingänge individuell skalierbar. Die Skalierung erfolgt über eine Definition an 2 Spannungswerten (Wert1 > Anzeige1, Wert2 > Anzeige2). Der Anzeigebereich beträgt maximal 0 bis +/-9999,9.



### EXTERNE MATRIX-TASTATUR

Am Steckeranschluss J120 kann eine Matrix-Tastatur (einzelne Tasten bis zur 8x8 Matrix) angeschlossen werden. Mit dem Befehl 'ESCY 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.

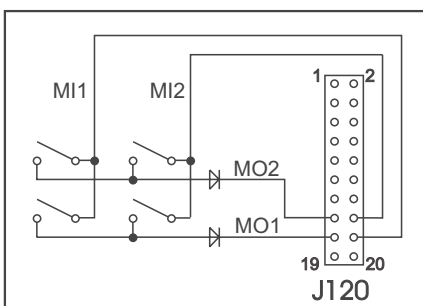
Hinweis Falls die Handshakleitung CTS das Senden nicht erlaubt, werden bis zu 8 Tastendrucke im Tastenpuffer zwischengespeichert. Ist der Puffer voll, können ältere Tastendrucke verloren gehen.

#### Bestimmung der Tastenummer:

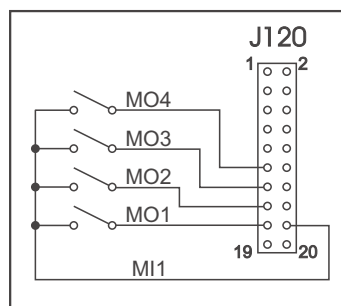
**TastenNr = (AusgangNr - 1) \* AnzahlEingänge + EingangNr** (Ausgang = MOx, Eingang = MIx).

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

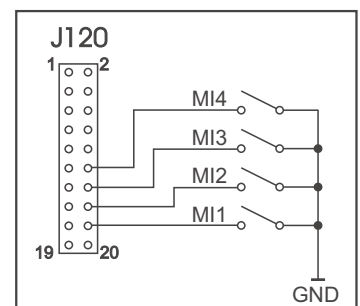
- Variante 1: Mit dem Befehl 'ESCY M 2 2 ..' werden die 4 Tasten als 2x2 Matrix definiert. Die Tasten werden an 2 Eingänge (MI1, MI2) und 2 Ausgänge (MO1, MO2) angeschlossen. Die Ausgänge sind hier mit Dioden voneinander entkoppelt um Doppeltastendrucke erkennen zu können. Es stehen weiterhin 6 Eingänge und 6 Ausgänge als Port-Anschlüsse zur Verfügung.
- Variante 2: Mit dem Befehl 'ESCY 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 'ESCY 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 KIT320F-8LWTP)

Die Version EA KIT320F-8CTP wird 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 20). 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:

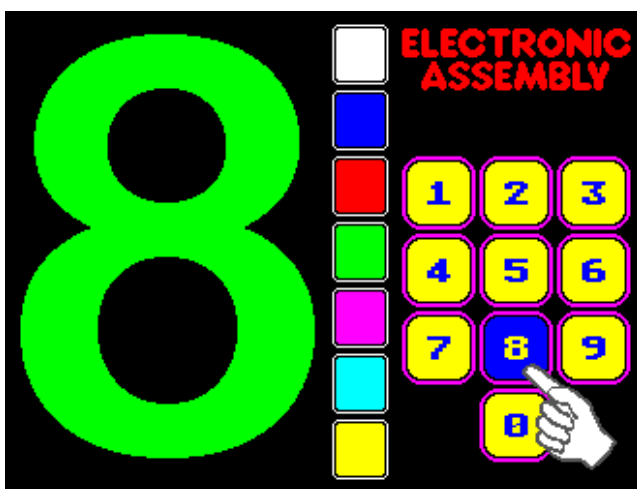
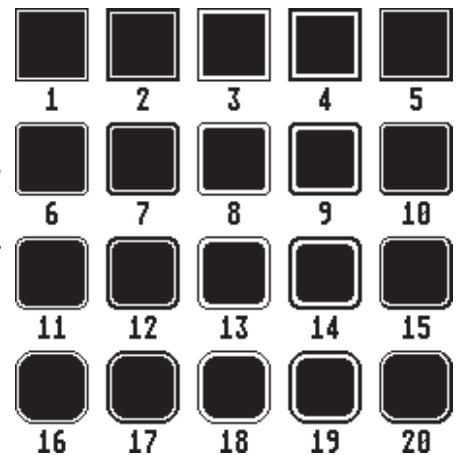
1. Beim Einschalten Touch berühren und gedrückt halten. Nach Erscheinen der Meldung "*touch adjustment ?*" den Touch wieder loslassen (alternativ den Befehl 'ESC @' senden).
2. Innerhalb 1 Sekunde den Touch nochmals für mindestens 1 Sekunde berühren.
3. Den Anweisungen zum Abgleich folgen (2 Punkte *Linksoben*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

### RAHMEN UND TASTENFORMEN

Mit den Befehlen *Rahmenbox zeichnen* sowie beim Zeichnen von Touchtasten kann ein Rahmentyp eingestellt werden. Es stehen dabei 20 interne Rahmentypen (nr.: 1..20) zur Verfügung; zudem können bis zu 20 eigene Rahmentypen (nr.: 21..40) eingebunden werden.

Diese Rahmen können in beliebiger Größe durch Angabe der Rechteck-Koordinaten gezeichnet werden. Die Rahmen sind aus 4 Teilen aufgebaut: Die Ecken (bei Typ 1-5 nicht Vorhanden), der äußere, der innere Rahmen sowie die Füllung.

Jedem dieser Teilbereiche kann eine eigene Farbe inkl. Attribut zugewiesen werden. Somit ist es möglich das z.B. beim Berühren des Touches nur ein bestimmter Teil der Touchtaste invertiert wird.

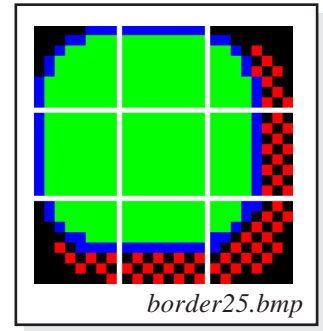




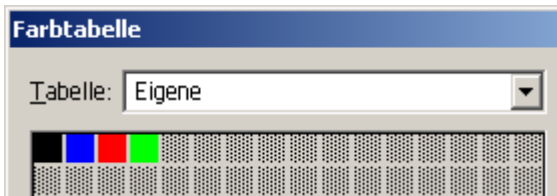
### EIGENETASTENFORMEN

Über den Kitcompiler *k320comp.exe* können eigene Rahmentypen eingebunden werden (siehe K320COMP.DOC auf der Diskette EA DISK320F: Compileranweisung "BORDER").

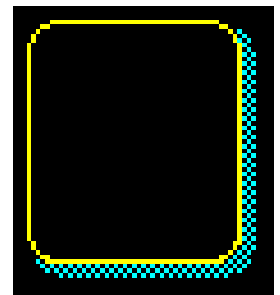
Jeder dieser zusätzlichen Rahmen wird als 16-farbiges Windows-BMP mit 24x24 Pixel hinterlegt, wobei nur die ersten 4 Farben genutzt werden (Aufbau: 9 Segmente mit je 8x8 Pixel: 4x Ecken, 4x Mittelstücke, 1x Füllung). Die Skalierbarkeit wird durch Wiederholung der 8x8 Pixel grossen Mittelstücke/Füllung erreicht. Deshalb muss die Grösse von 8x8 Pixel eingehalten werden.



*border25.bmp*



Die ersten 4 Farbeinträge dieser 16-farbigen BMPs entsprechen den 4 einstellbaren Rahmenfarben. Auf der Diskette EA DISK320F sind einige Beispielrahmen im Verzeichnis *'DISK320F\BITMAPS\border'* vorhanden



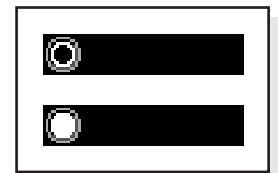
*border25:  
50x56 Pixel Grösse*

### BITMAPS ALSTASTEN

Ausser den Rahmentypen, die in der Grösse frei skalierbar sind, gibt es noch die Möglichkeit beliebige Bitmaps als Touch-Tasten oder -Schalter zu verwenden. Über den Kitcompiler *k320comp.exe* können bis zu 127 eigene Buttons eingebunden werden (siehe K320COMP.DOC auf der Diskette EA DISK320F: Compileranweisung "BUTTON").

Ein Button besteht immer aus zwei gleich grossen 16-farbigen Windows-BMPs (ein Bitmap für die gedrückte Touchtaste und ein Bitmap für normale Darstellung der Touchtaste). Die aktive Fläche der Touchtaste ergibt sich automatisch aus der Grösse der Button-Bitmaps.

Werden Buttons nachträglich beschriftet, so muss das Farbattribut SOLID gesetzt sein um ein Überschreiben des Textes beim Drücken zu verhindern.



*RadioBlack75x15\_0.bmp  
RadioBlack75x15\_1.bmp*

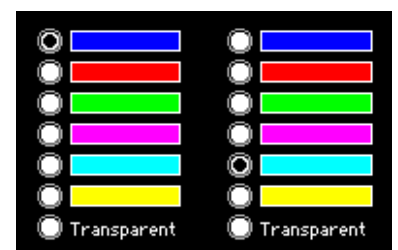


Die Farbpalette der Button-Bitmaps wird nicht zur Darstellung verwendet. Die Buttons werden immer in den 16 KIT-Grundfarben gezeichnet.

Auf der Diskette EA DISK320F sind einige Beispielbuttons im Verzeichnis *'DISK320F\BITMAPS\button'* vorhanden.

### SCHALTER IN GRUPPEN (RADIO GROUP)

Touch-Schalter ändern ihren Zustand bei jeder Berührung von EIN in AUS und umgekehrt. Mehrere Touchschalter können zu einer Gruppe zusammengefasst werden (Befehl: 'ESC A R nr'). Wird nun ein Touch-Schalter innerhalb einer Gruppe 'nr' eingeschaltet, dann werden automatisch alle andern Touch-Schalter dieser Gruppe ausgeschaltet. Es ist also automatisch immer nur ein Schalter gesetzt.



*Beispiel von zwei Radio-Gruppen mit Touch-Schaltern*

Technische Änderungen vorbehalten. Wir übernehmen keine Haftung für Druckfehler.

### MAKRO PROGRAMMIERUNG

Einzelne oder mehrere Befehlsfolgen können als sog. Makros zusammengefasst und im Daten-Flash fest abgespeichert werden. Diese können dann mit den Befehlen *Makro ausführung* 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 Matrixtastatur wird an den Ein- und Ausgängen angeschlossen; es kann max. eine 8x8 Matrixtastatur angeschlossen werden.

#### Analog Makro (0..19)

automatischer Start bei Änderung des Analogwertes AIN1 oder AIN2 oder per Befehl 'ESC MV xx'. Siehe Tabelle nebenan:

#### Prozess Makro (0..255)

automatischer Start in bestimmten Zeitintervallen (0,1s bis 25s) oder per Befehl 'ESC MC xx'. Bis zu 16 unabhängige Prozesse können mit dem Befehl 'ESC MD ..' definiert werden. Prozess-Makros werden nicht durch andere Befehle unterbrochen.

#### Power-On-Makro

Start nach dem Einschalten. Hier kann man zB. 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. Bei Verwendung des Simulators muss die Datei *Disk320f\kit320f-8.bin* gelöscht werden.

Analog Makros		
Makro Nr.		Start des Makros bei
AIN1	AIN2	
0	10	jeder Änderung des Analogwertes
1	11	fallendem Analogwert
2	12	steigendem Analogwert
3	13	kleiner unterer Grenzwert
4	14	größer unterer Grenzwert
5	15	kleiner oberer Grenzwert
6	16	größer oberer Grenzwert
7	17	Ausserhalb beider Grenzwerte
8	18	Innerhalb beider Grenzwerte
9	19	kleiner als anderer Analogkanal

### ERSTELLEN INDIVIDUELLER MAKROS UND BILDER

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

- einen PC mit serieller Schnittstelle z.B. COM1 oder COM2 und ca. 1MB Platz auf der Festplatte
- die Software ELECTRONIC ASSMBLY LCD-Tools<sup>\*)</sup>; sie ist kostenfrei auf der Webseite verfügbar und enthält einen Kit-Editor, Kit-Compiler, sowie Beispiele und Fonts (für PC-Win)

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 über den Kit-Editor definiert, startet man über F5 den Compilerlauf. Dieser erzeugt eine Datei DEMO.DF. Ist das Display an den PC angeschlossen, wird diese Datei automatisch in das DatenFlash des Displays gebrannt.

Eine ausführliche Beschreibung zur Programmierung der Makros finden Sie zusammen mit Beispielen in der Hilfefunktion der ELECTRONIC ASSEMBLY LCD-Tools<sup>\*)</sup> Software.

### SCHREIBSCHUTZ FÜR MAKROPROGRAMMIERUNG

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.

### 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 U I" oder aus einem Makro heraus. Verwendet werden können alle Bilder im Windows BMP-Format (nur monochrome und 16 farbige Bilder). Die Erstellung und Bearbeitung erfolgt über Standardsoftware wie z.B. Windows Paint oder Photoshop.

<sup>\*)</sup> im Internet unter <http://www.lcd-module.de/deu/touch/touch.htm>

### INTEGRIERTE FONTS

Es sind standardmäßig 6 monospaced, 6 proportionale Zeichensätze und 1 grosser Ziffernfont integriert. Im Terminal können nur monospaced Zeichensätze verwendet werden. Die proportionalen Zeichensätze im Grafikmode können pixelgenau platziert werden und ergeben ein schöneres Schriftbild. Gleichzeitig benötigen sie weniger Platz auf dem Bildschirm (z.B. schmales "i" und breites "W").

Jeder Zeichensatz kann 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. Es lässt sich auch mehrere verschiedene Schriftgrößen gemeinsam darstellen.

Font		Zeichen höhe	Zeilen x Zeichen	Größe in Pixel	ASCII- Bereich	Bemerkung
Nr.	Name					
1	FONT4x6	2,1 mm	40 x 80	4 x 6	32 - 158	Microschrift, monospaced
2	FONT5x6	2,1 mm	40 x 64	5 x 6	32 - 158	Minischrift, monospaced
3	FONT6x8	2,8 mm	30 x 53	6 x 8	32 - 255	erweiterter ASCII-Code ohne Rahmensymbole (176-223) monospaced
4	FONT8x8	2,8 mm	30 x 40	8 x 8	32 - 255	
5	FONT7x12	4,3 mm	20 x 45	7 x 12	32 - 255	
6	FONT8x16	5,7 mm	15 x 40	8 x 16	32 - 255	
7 : 16						10 frei definierbare Zeichensätze
17	GENEVA12	4,3 mm	20 x 32	<10 x 12	32 - 168	Geneva, proportional
18	CHICAGO16	5,7 mm	15 x 26	<12 x 16	32 - 168	Chicago, proportional
19	TIMES20	7,2 mm	12 x 17	<18 x 20	32 - 168	Times, proportional
20	TIMES26	9,3 mm	9 x 13	<24 x 26	32 - 168	
21	SWISS28	10,0 mm	8 x 11	<27 x 28	32 - 168	Swiss, proportional
22	SWISS38	13,6 mm	6 x 10	<33 x 38	32 - 168	
23	BIGZIF57	20,5 mm	4 x 8	<37 x 57	43 - 58	Große Ziffern '0'-'9' und '+,-,:'

werden. Es lässt sich auch mehrere verschiedene Schriftgrößen gemeinsam darstellen.

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

### EXTERNE FONTS

Die Makroprogrammierung erlaubt die Einbindung von weiteren 10 Fonts (7..16). Es können alle nur erdenklichen Schriften mit bis zu 255x240 Pixeln Größe mit einem Texteditor erstellt und über den Kitkompiler *KCOMP320.EXE* einprogrammiert werden.

### EINSTELLMÖGLICHKEITEN BEI DER AUSGABE VON ZEICHENKETTEN

Während der Ausgabe von Zeichenketten können Farben und Attribute lokal umgestellt werden, die globale Farbeinstellung mit 'ESC FZ' wird hierbei nicht verändert.

Die Kommandos beginnen mit dem Zeichen '\ ' Backslash gefolgt von einer Ziffer/Grossbuchstabe.

Nebenan ist die Übersichtstabelle mit allen Einstellmöglichkeiten innerhalb einer Zeichenkette.

Einstellungen innerhalb einer Zeichenkette			
CMD	Bemerkung	CMD	Bemerkung
	Neue Zeile beginnen	V0	Farbe 0: Schwarz
@...@	Text zwischen zwei '@' blinkt invers (Schrift- und Hintergrundfarbe)	V1	Farbe 1: Blau
~...~	Text zwischen zwei '~' blinkt AN/AUS (Schrift- und Hintergrundfarbe)	V2	Farbe 2: Rot
\j	Zeichen 'j' darstellen	V3	Farbe 3: Grün
\@	Zeichen '@' darstellen	V4	Farbe 4: Magenta
\~	Zeichen '~' darstellen	V5	Farbe 5: Cyan
\	Zeichen '\ ' darstellen	V6	Farbe 6: Gelb
V	Attribut- bzw. Farb- Einstellungen wirken auf die Schrift (=default)	V7	Farbe 7: Weiß
H	Attribut- bzw. Farb- Einstellungen wirken auf den Hintergrund	V8	Farbe 8: Dunkelgrau
VL	Attribut 0: Attribut löschen	V9	Farbe 9: Orange
VS	Attribut 16: SOLID (nicht invertierbar)	VA	Farbe 10: Violett
V	Attribut 32: INVERS Blinken	VB	Farbe 11:
VO	Attribut 48: AN/AUS Blinken	VC	Farbe 12:
		VD	Farbe 13: Gelbgrün
		VE	Farbe 14: Hellblau
		VF	Farbe 15: Hellgrau
		VT	Farbe 64: Transparent

### ZEICHENTABELLEN

Die unteren Tabellen zeigen die vorhandenen Zeichen.

+ 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
\$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)		€	ü	é	â	ä	á	ã	ç	ê	ë	è	ï	î	ì	ñ
\$90 (dez: 144)		É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü				

Zeichencodes für monospaced Fonts 1 und 2

+ 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
\$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)		€	ü	é	â	ä	á	ã	ç	ê	ë	è	ï	î	ì	ñ
\$90 (dez: 144)		É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü				
\$A0 (dez: 160)		á	í	ó	ú	ñ	Ñ	á	o	¿	¼	½	¾	¿	»	
\$B0 (dez: 176)																
\$C0 (dez: 192)																
\$D0 (dez: 208)																
\$E0 (dez: 224)		α	β	Γ	π	Σ	ο	μ	τ	Ϛ	θ	φ	δ	φ	φ	ε
\$F0 (dez: 240)		≡	±	≥	≤	∫	∫	÷	≈	°	•	•	√	∞	∞	-

Zeichencodes für monospaced Fonts 3 - 6

### SCHRIFTBILD

Das untere Hardcopy zeigt die 13 internen Schriften



+ 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
\$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)		€	ü	é	â	ä	á	ã	ç	ê	ë	è	ï	î	ì	ñ
\$90 (dez: 144)		É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü				
\$A0 (dez: 160)		á	í	ó	ú	ñ	Ñ	á	o							
\$B0 (dez: 176)																
\$C0 (dez: 192)																
\$D0 (dez: 208)																
\$E0 (dez: 224)		β														
\$F0 (dez: 240)										o						

Zeichencodes für proportionale Fonts 17 - 22

+ 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	:				

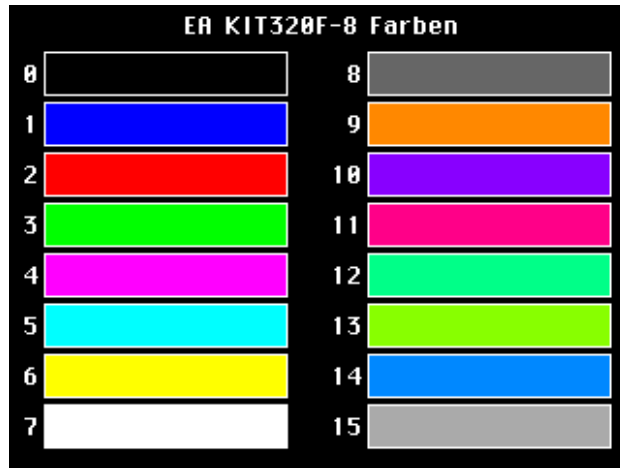
Zeichencodes für Font 23 (BigZif57)

# EA KIT320F-8 COLOR

## ELECTRONIC ASSEMBLY

### DARSTELLBARE FARBEN

Im EA KIT320F-8 sind 16 Farben (0..15) zur Ausgabe von Zeichenketten und Grafikbefehlen einstellbar. Die Farbnummer 64=Transparent hat dabei eine Sonderstellung, wird z.B. TRANSPARENT als Hintergrund für Zeichenketten eingestellt so werden die Buchstaben ohne Hintergrund gezeichnet d.h. der vorhandene Hintergrund bleibt erhalten.



Farben	
Nr.	Bezeichnung
0	Schwarz
1	Blau
2	Rot
3	Grün
4	Magenta
5	Cyan
6	Gelb
7	Weiß
8	Dunkelgrau
9	Orange
10	Violett
11	
12	
13	Gelbgrün
14	Hellblau
15	Hellgrau
64	Transparent

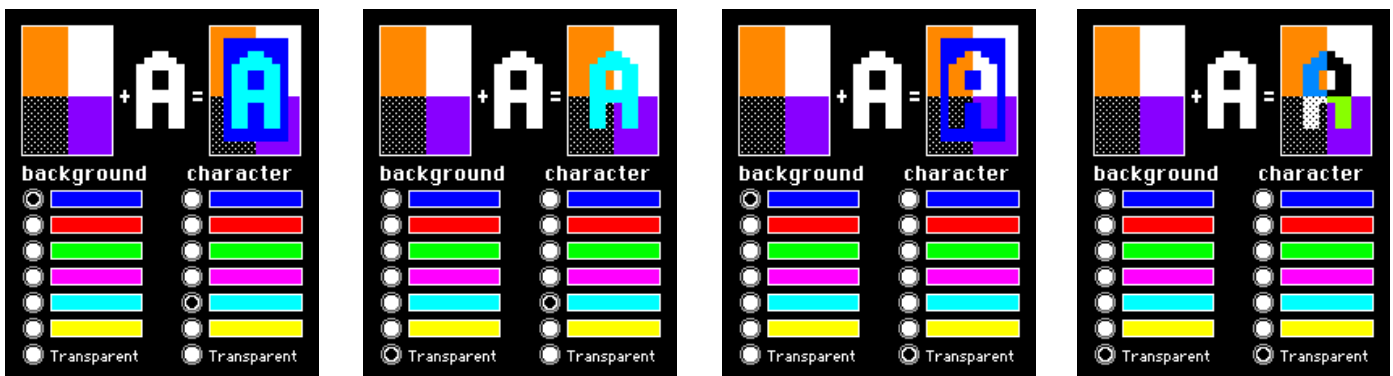
Die unsinnige Einstellung Vorder- und Hintergrundfarbe auf TRANSPARENT zu stellen (dh. nichts würde sichtbar sein) hat zur Folge das die zu zeichnenden Pixel invertierend wirken (=Komplementärfarbe). Wird das gleiche ein zweites mal an derselben Stelle ausgegeben so wird die ursprüngliche Grafik wiederhergestellt.

### FARB-ATTRIBUTE

es gibt 3 mögliche Attribute die eingestellt werden können. Der Wert des gewünschten Attributs wird beim Einstellen der Farbe zu der Farbnummer addiert. Es ist immer nur ein Attribut erlaubt, mehrere Attribute sind nicht kombinierbar!

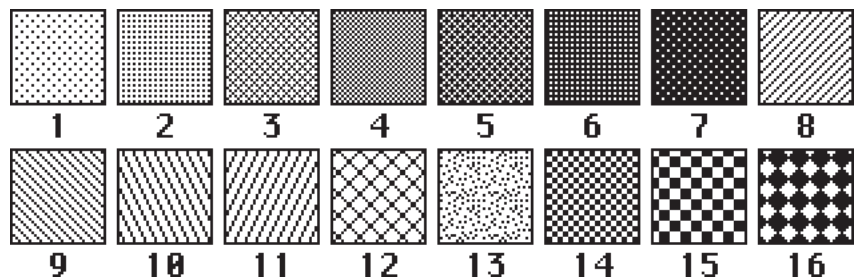
Farbattribute (zur Farbe addieren)		
Wert	Bezeichnung	Bemerkung
+ 0	kein Attribut	
+ 16	SOLID	Farben sind nicht invertierbar
+ 32	BLINKINVERS	Farben blinken invers. Invers-Farbe mit dem Befehl 'ESC QE' einstellbar
+ 48	BLINKONOFF	Farben blinken EIN/AUS. AUS-Farbe mit Befehl 'ESC QF' einstellbar

Nachfolgend einige Beispiele zur Darstellung des Zeichens 'A' auf einen vorhandenen Hintergrund.



### 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.



Technische Änderungen vorbehalten. Wir übernehmen keine Haftung für Druckfehler.

THIS PAGE IS INTENTIONALLY LEFT BLANK

### 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).

### 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%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)											



### 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.

Die Grösse des benutzbaren Terminalfensters kann frei definiert werden.

**Achtung:** Mit Grafikbefehlen kann der Inhalt des Terminalfensters überschrieben werden z.B. Löschen des Grafikbildschirms mit 'ESC DL'.

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 '##'.

EA KIT320F-8: Befehlstabelle für den Terminal Betrieb										nach Reset
Befehl	Codes				Anmerkung					
<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
Cursorposition sichern			S	die aktuelle Cursorposition wird gesichert						
Cursorposition restoren			R	die letzte gesicherte Cursorposition wird wieder hergestellt						
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						
Terminal sichtbar			E	Terminal Anzeige ist wieder sichtbar;					Ein	
<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>										
Terminal Farbe einstellen	ESC	F	T	vf	hf	Farbe für den Terminal Betrieb einstellen. vf= Schriftfarbe; hf= Hintergrundfarbe				7 / 0
Font einstellen	ESC	T	F	n1	Font Nr. n1 für Terminal Betrieb einstellen. (nur monospaced Fonts)					
zus. Zeilenabstand			Y	n1	für den aktuellen Font werden zusätzlich n1 Pixel als Zeilabstand 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	
<b>Einstellungen für das externe Dotmatrix-Modul (optional an J7)</b>										
Dotmatrixmodul initialisieren	ESC	T	D	n1	n2	Ein externes Dotmatrix-Display (HD44780 kompatibel) initialisieren n1 = anzahl Zeichen; n2 = anzahl Zeilen				

# EA KIT320F-8 COLOR

## ELECTRONIC ASSEMBLY

### ALLE BEFEHLE AUF EINEN BLICK

Die nachfolgenden Befehlstabellen geben einen Überblick über die eingebauten Funktionen des EA KIT320F-8.

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 diese Einstellungen durch Erstellen eines Power-On-Makros überschrieben werden können.

EA KIT320F-8: Befehlstabelle 1							nach Reset	
Befehl	Codes			Anmerkung				
<b>Display-Befehle (Wirkung auf das gesamte Display)</b>								
Displayfarben einstellen	ESC	F	D	vf	hf		Farbe für Display und Bereiche: vf=Vordergrundfarbe; hf=Hintergrundfarbe	7
Display löschen	ESC	D	L				Displayinhalt löschen; alle Pixel mit Hintergrundfarbe füllen	
Display füllen			S				Displayinhalt füllen; alle Pixel mit Vordergrundfarbe füllen	
Display mit Farbe füllen			F	n1			Displayinhalt mit Farbe n1 füllen	
Display invertieren			I				Alle Farben ohne SOLID-Attribut invertieren	
Display ausschalten			A				Displayinhalt wird unsichtbar bleibt aber erhalten, Befehle weiterhin möglich	
Display einschalten			E				Displayinhalt wird wieder sichtbar	Ein
Display Update			U	n1		n1=0: Displayausgaben sind unsichtbar (werden aber weiterhin ausgeführt) n1=1: Displayausgaben sind sofort sichtbar n1=2: Displayinhalt auffrischen (vorherige Ausgaben werden sichtbar)	1	
<b>Rechteckige Bereiche verändern / zeichnen</b>								
Bereich löschen	ESC	R	L	xx1 yy1	xx2 yy2		Einen Bereich von xx1,yy1 nach xx2,yy2 mit Display-Hintergrundfarbe füllen	
Bereich füllen			S	xx1 yy1	xx2 yy2		Einen Bereich von xx1,yy1 nach xx2,yy2 mit Display-Vordergrundfarbe füllen	
Bereich mit Farbe füllen			F	xx1 yy1	xx2 yy2	n1	Einen Bereich von xx1,yy1 nach xx2,yy2 mit Farbe n1 füllen	
Bereich invertieren			I	xx1 yy1	xx2 yy2		Alle Farben, ohne SOLID-Attribut, von xx1,yy1 nach xx2,yy2 invertieren	
Bereich m. Füllmuster			M	xx1 yy1	xx2 yy2	n1	Bereich von xx1,yy1 nach xx2,yy2 mit Muster n1 in den Displayfarben zeichnen	
Boxfarben einstellen	ESC	F	O	vf hf rf		Farbe für Box: vf=Musterfarbe; fh=Muster-Hintergrundfarbe; rf=Rahmenfarbe	7,0,7	
Box zeichnen	R	O	xx1 yy1	xx2 yy2	n1	Rechteck von xx1,yy1 nach xx2,yy2 mit Füllmuster n1 zeichnen		
Rahmenfarben einstellen	F	R	f1 f2 f3	f4		Farbe für Rahmen: f1=Ecken; f2=Rahmen aussen; f3=Rahmen Innen; f4=Füllung	7070	
Rahmenbox zeichnen	ESC	R	T	xx1 yy1	xx2 yy2	n1	Eine Rahmenbox Typ n1 von xx1,yy1 nach xx2,yy2 zeichnen	
<b>Geraden und Punkte zeichnen</b>								
Geradenfarben einstellen	ESC	F	G	vf hf		Farbe für Geraden einstellen: vf = Linienfarbe; hf = Musterhintergrundfarbe;	7,64	
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	0
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);	1
Muster			M	n1			Geraden-/Punkt-Muster Nr. n1 einstellen; 0 = kein Muster verwenden	0
<b>Befehle zur Ausgabe von Zeichenketten</b>								
Textfarben einstellen	ESC	F	Z	vf hf		Farbe für Zeichenketten einstellen: vf = Schriftfarbe; hf = Hintergrundfarbe;	7	
Zeichenkette ausgeben	ESC	Z	L	xx1 yy1	Text ...	NUL	Zeichenkette (...) an xx1,yy1 ausgegeben. 'NUL' (\$00), 'LF' (\$0A) oder 'CR' (\$0D) = Zeichenkettenende; mehrere Zeilen werden durch das Zeichen ' ' (\$7C) getrennt Texte die zwischen zwei '-' (\$7E) Zeichen stehen blinken An/Aus; Texte die zwischen zwei '@' (\$40) Zeichen stehen blinken invertierend;	
L: Linksbündig, C: Zentriert R: Rechtsbündig			C	xx1 yy1				
Font einstellen			F	n1			Font mit der Nummer n1 einstellen	5
Font-Zoomfaktor			Z	n1 n2			n1 = X-Zoomfaktor (1x..8x); n2 = Y-Zoomfaktor (1x..8x)	1
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°;	0		
Text-Muster	M	n1			Text mit Muster Nr. n1 verknüpfen; 0 = Text mit keinem Muster verknüpfen	0		
Zeichenkette für Terminal	ESC	Z	T		Text ...	Befehl um eine Zeichenkette in einem Makro an das Terminal ausgeben zu können		
<b>Befehle für alle Bilder</b>								
Bild laden	ESC	U	L	xx1 yy1	daten ...		Bild von RS-232 nach xx1,yy1 laden; daten=Bild im *.BH7 Format; Diese Daten werden mit dem Programm BMP2BH7.EXE (auf der EA DISK320) erzeugt.	
internes Bild laden			I	xx1 yy1	nr		internes Bild mit der nr (0..255) aus dem EEPROM nach xx1,yy1 laden	
Hardcopy senden			H				Ein komplettes Bild wird im Windows BMP-Format angefordert. Zuerst wird der Bildheader und dann die eigentlichen Bilddaten über RS232 gesendet (77878 Bytes).	
<b>Befehle für monochrome Bilder</b>								
Monochrombild Farben	ESC	F	U	vf hf		Bildfarbe für monochrome Bilder vf = Vordergrundfarbe; hf = Hintergrundfarbe;	7,0	
Monochrombild Zoom	ESC	U	Z	n1 n2		n1 = X-Zoomfaktor (1x..8x); n2 = Y-Zoomfaktor (1x..8x)	1	
Monochrombild Winkel			W	n1			Ausgabewinkel: n1=0: 0°; n1=1: 90°; n1=2: 180°; n1=3: 270°;	0
Monochrombild Muster			M	n1			Bild mit Muster Nr. n1 verknüpfen; 0 = Bild mit keinem Muster verknüpfen	0
<b>Befehle für Farbbilder (16 Farben)</b>								
Farbbild Attribut	ESC	U	A	n1		n1=0 kein Attribut für Farbbilder; n1=16 Farbbild ist nicht invertierbar (SOLID) n1=32 Farbbild blinkt An/Aus; n1=48 Farbbild blinkt invertierend;	16	
Farbbild Palette			P	n1		n1=0: Bildpalette wird nicht benutzt d.h. das Bild wird mit Farben 0..15 dargestellt Bis zu 3 Bilder mit unterschiedlichen Paletten n1=1..3 können dargestellt werden	1	
Farbbild Transparenz			T	n1			n1=0 Bild mit allen 16 Farben Rechteckig darstellen; n1=1 die Farbe der linken oberen Ecke des Bildes wird als Transparentfarbe definiert und nicht dargestellt.	0

Technische Änderungen vorbehalten. Wir übernehmen keine Haftung für Druckfehler.

### EA KIT320F-8: Befehlstabelle 2

Befehl	Codes		Anmerkung				nach Reset				
<b>Bargraph Befehle</b>											
Bargraph Farben einstellen	ESC	F	B	vf	hf	rf	Farben für den Bargraph einstellen. vf = Vordergrundfarbe; hf = Hintergrundfarbe; rf = Rahmenfarbe  Bargraph nach L(inks), R(echts), 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	7,0,7  kein Bar definiert			
Bargraph definieren			R	nl	xx1	yy1			xx2	yy2	aw
Bargraph aktualisieren	ESC	B	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 seriellen Schnittstelle senden						
Bargraph löschen	ESC	D	n1	n2	Die Definition des Bargraph mit der Nummer n1 wird ungültig. War der Bargraph als Eingabe mit Touch definiert so wird auch dieses Touchfeld gelöscht. n2=0: Bar bleibt weiterhin sichtbar; n2=1: Bar wird mit Display-Hintergrundfarbe gelöscht						
<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.		1, 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			
Farbpalette wiederherstellen			P	n1	n1=0: Die Farbpalette wird nicht wiederhergestellt. n1=1: Die Farbpalette wird immer wiederhergestellt. n1=2: Die Farbpalette wird nur bei Vollbildern (320x240 Pixel) wiederhergestellt.				2		
<b>Blinkbereich / Attribut Befehle</b>											
Blinkzeit einstellen	ESC	Q	Z	n1			Einstellen der Blinkzeit n1= 1..15 in 1/10s; 0=Blinkfunktion deaktivieren		6		
Blinkattribut On / Off			O	xx1	yy1	xx2	yy2	Blinkbereich (An/Aus) von xx1,yy1 bis xx2,yy2 setzen (BLINKONOFF-Attribut)			
Blinkfarbe OFF einstellen			F	fh	Einstellen der Hintergrundfarbe beim EIN /AUS blinken						
Blinkattribut Invertierend			I	xx1	yy1	xx2	yy2	Invertierenden Blinkbereich von xx1,yy1 bis xx2,yy2 setzen (BLINKINVERS-Attribut)			
Invertierfarbe einstellen			E	f1	f2	Der Farbe f1=0..15 wird als Invertierfarbe f2=0..15 zugewiesen					
Attribut Solid			S	xx1	yy1	xx2	yy2	Bereich von xx1,yy1 bis xx2,yy2 definieren der nicht Invertierbar ist (SOLID-Attribut)			
Attribute löschen			L	xx1	yy1	xx2	yy2	Löscht das BLINKxxx- oder SOLID-Attribut von xx1,yy1 bis xx2,yy2			
Farbpalette rotieren	R	f1	f2	n1	n2	Die Farbnummern von f1 bis f2 (0..15) innerhalb der Farbpalette n1=0..3 alle n2/10s rotieren (n2=0: Rotieren AUS). Es werden nur Farben mit dem SOLID-Attribut rotiert.					
<b>Einstellungen für Menübox / Touchmenü</b>											
Menü Farben einstellen	ESC	F	N	vf	hf	rf	Farben für die Menüs / Touchmenüs einstellen. vf = Schriftfarbe; hf = Hintergrundfarbe; rf = Rahmenfarbe		7,0,7		
Menü-Font einstellen			F	n1	Font mit der Nummer n1 für Menüdarstellung einstellen				5		
Menüfont-Zoomfaktor			Z	n1	n2	n1 = X-Zoomfaktor (1x..8x); n2 = Y-Zoomfaktor (1x..8x)				1,1	
zus. Zeilenabstand			Y	n1	zwischen zwei Menüeinträgen n1 Pixel als zusätzlichen Zeilenabstand einfügen						
Menü-Winkel			W	n1	Menüdarstellung Winkel: n1=0: 0°; n1=1: 90°; n1=2: 180°; n1=3: 270°;				0		
Touchmenü-Automatik			T	n1	n1=1: Touchmenü öffnet automatisch; n1=0: Touchmenü öffnet nicht automatisch stattdessen wird die Anforderung 'ESC T 0' zum Öffnen über RS-232/422 an den Hostrechner gesendet, dieser kann dann mit 'ESC N T 2' das Touchmenü öffnen.				1		
<b>Menübox Befehle (Steuerung mit Tasten nicht per Touch)</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			N	Der nächste Eintrag wird invertiert oder bleibt am Ende stehen							
vorheriger Eintrag			P	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							

# EA KIT320F-8 COLOR

## ELECTRONIC ASSEMBLY

### BEFEHLE ZUR UNTERSTÜTZUNG DESTOUCHPANELS

EA KIT320F-8: Befehle für das Touch-Panel										nach			
Befehl	Codes				Anmerkung					Reset			
<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-Fon in der Touch-Taste plaziert 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-Fon in der Touch-Taste plaziert 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/Menumakro+(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. Das erste Zeichen bestimmt die Richtung in der das Menü aufklappt (R=rechts L=links O=oben U=Unten). Die einzelnen Einträge sind durch Zeichen ' ' (\$7C,dez:124) getrennt. z.B. "UTaste 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 Farben	ESC F	E	f1	f2	f3	f4	Farben für den Rahmen von Touch-Tasten/Schaltern einstellen. f1=Ecken; f2=Rahmen aussen; f3=Rahmen Innen; f4=Füllung						
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 zugeordneten Return-Code wird manuell Invertiert									
Touch-Schalter einstellen		X	Code	Zustand des Schalters (Aus=0; Ein=1) wird über die serielle Schnittstelle gesendet									
Radiogroup für Schalter		P	Code n1	Zustand des Schalters wird per Befehl geändert n1=0=Aus; n1=1=Ein.									
Touch-Bereich Löschen		R	n1	Innerhalb einer Gruppe ist immer nur 1 Schalter aktiv, alle anderen werden deaktiviert n1=0: neu definierte Schalter gehören keiner Gruppe an. n1=1..255: neu definierte Schalter gehören der Gruppe mit der Nummer n1 an.							0		
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 mit Display-Hintergrundfarbe gelöscht.									
Touch-Abfrage Ein/Aus		V	xx1 yy1 n1	Touchbereich der die Koordinaten xx1,yy1 umschließt aus der Touchabfrage entfernen n1=0: Bereich bleibt sichtbar; n1=1: Bereich mit Display-Hintergrundfarbe löschen									
		Q	n1	das automatische Senden eines neuen Bargraphwertes per Toucheingabe 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 Farbe	ESC F	A	vf	hf	Farbe für die Touchtastenbeschriftung einstellen. vf=Schriftfarbe; hf=Hintergrundfarbe						vf=7 hf=0		
Beschriftungs Font	ESC A	F	n1	Font mit der Nummer n1 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		

Technische Änderungen vorbehalten. Wir übernehmen keine Haftung für Druckfehler.

# COLOR EA KIT320F-8

## ELECTRONIC ASSEMBLY

### BEFEHLE FÜR MAKROS, PORTS UND SONSTIGES

EA KIT320F-8: Befehle für Makros, Ports und Sonstiges										nach Reset		
Befehl	Codes			Anmerkung								
<b>Makro Befehle</b>												
Normal 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)		
Analog Makro ausführen			V	n1						Das Analog-Makro mit der Nummer n1 (0..19) aufrufen (max. 7 Ebenen)		
Prozess Makro ausführen			C	n1						Das Prozess-Makro mit der Nummer n1 (0..255) aufrufen (max. 7 Ebenen)		
Makros sperren	ESC	M	L	typ	n1	n2					Die Makros vom typ='N','T','P','B','M','X','V' oder 'C' (typ='A' alle Makrotypen) werden von der Nummer n1 bis n2 gesperrt, d.h. bei Aufruf nicht mehr ausgeführt.	
Makros freigeben			U	typ	n1	n2					Die Makros vom typ='N','T','P','B','M','X','V' oder 'C' (typ='A' alle Makrotypen) werden von der Nummer n1 bis n2 freigegeben, d.h. bei Aufruf wieder ausgeführt.	
<b>automatische (Normal-) Makros</b>												
Makro mit Verzögerung	ESC	M	G	n1	n2					Das (Normal-)Makro mit der Nummer n1 (0..255) in n2/10s aufrufen. Die Ausführung wird durch andere Befehle (z.B durch RS-232, Touch-, Port-, Bit-, Matrixmakros) gestoppt.		
autom. Makros einmal			E	n1	n2	n3					Makros n1..n2 automatisch einmal abarbeiten; n3=Pause in 1/10s. Die Ausführung wird durch andere Befehle (z.B durch RS-232, Touch-, Port-, Bit-, Matrixmakros) gestoppt.	
autom. Makros zyklisch			A	n1	n2	n3					Makros n1..n2 automatisch zyklisch abarbeiten; n3=Pause in 1/10s. Die Ausführung wird durch andere Befehle (z.B durch RS-232, Touch-, Port-, Bit-, Matrixmakros) gestoppt.	
autom. Makros pingpong			J	n1	n2	n3					Makros autom. von n1..n2..n1 (PingPong) abarbeiten; n3=Pause in 1/10s. Die Ausführung wird durch andere Befehle (durch RS-232, Touch-,Port-,Bit-, Matrixmakros) gestoppt.	
<b>Makro Prozesse</b>												
Makroprozess definieren	ESC	M	D	nr	typ	n3	n4	zs			Ein Makroprozess mit der Nummer nr (1..16) wird definiert (1=höchste Priorität). Die Prozessmakros n3 bis n4 werden nacheinander alle zs/10s ausgeführt. typ: 1=einmal; 2=zyklisch; 3=pingpong n3..n4..n3	
Makroprozess Zeitintervall			Z	nr	zs						Dem Makroprozess mit der Nummer nr (1..16) wird eine neue Zeit zs in 1/10s zugeordnet. Ist die Zeit zs=0 so wird die Ausführung angehalten.	
Makroprozesse anhalten			S	n1						Alle Makroprozesse werden mit n1=0 gestoppt und n1=1 wieder gestartet. z.B. um Einstellungen und Ausgaben über die RS-232 Schnittstelle ungestört auszuführen	1	
<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-Beleuchtung n1=0: AUS; n1=1: EIN; n1=2: Halbe Helligkeit;	1	
Summer Ein / Aus			S	n1						n1=1..255: Summerfür n1 1/10s lang Ein	AUS	
<b>Sonstige-Befehle</b>												
Farbe neu definieren	ESC	F	P	n1	r16	g16	b16				Der Farbe n1=0..15 neue RGB-Werte (0..15) zuweisen.	
Kontrast increment	ESC	P	I								Der Kontrast wird um einen Step erhöht	
Kontrast decrement			D								Der Kontrast wird um einen Step verringert	
Kontrast einstellen			S	n1								Der Kontrast wird auf den Wert n1=0..40 gesetzt.
Warten (Pause)	ESC	X	n1								n1 Zehntel-Sekunden abwarten bevor der nächste Befehl ausgeführt wird.	
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)		
Version senden			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		

# EA KIT320F-8 COLOR

## ELECTRONIC ASSEMBLY

### BEFEHLE FÜR DIE ANALOGEINGÄNGE

Die nachfolgende Befehltabelle gibt einen Überblick über die Möglichkeiten der beiden Analogeingänge AIN1 und AIN2. Die Auflösung beträgt 10-Bit, bei 0..+250mV eingangsempfindlichkeit (in etwa wie ein 3 stelliges DVM).

EA KIT320F-8: Befehle für die Analogeingänge										nach Reset					
Befehl	Codes				Anmerkung										
<b>Befehle für Analogeingänge</b>															
Analog-Abfrage Ein/Aus	ESC	V	A	n1			Der automatische Scan der Analogkanäle wird n1=0: deaktiviert; n1=1: aktiviert				1				
Analogwert senden			D	nr			Der Wert in [mV] vom Analogkanal nr=1..2 wird auf der RS-232/RS-422 gesendet								
Bereiche / Grenzen für Analog-Makros			K	nr	n1	n2	n3	Zwei Grenzen für Analogkanal nr=1..2 einstellen. Mit Hilfe dieser Grenzen können bei Über- und/oder Unterschreiten diverse Analog-Makros automatisch gestartet werden. n1=untere Grenze in [mV]; n2=obere Grenze in [mV]; n3=Hyterese in [mV].				0			
Bargraph für Analogeingang definieren	ESC	V	R L O U	nr	bn	xx1	yy1	xx2	yy2	aw	ew	typ	mst	Bargraph für Analogkanal nr=1..2 nach L(inks), R(rechts), O(ben), U(nten) mit der Nr. bn definieren. xx1,yy1,xx2,yy2 sind das umschließende Rechteck des Bargraphs. aw,ew sind die Werte für 0% und 100% in [mV] typ=0: Balken; typ=1: Balken im Rechteck; mst=Balkenmuster; typ=2: Strich; typ=3: Strich im Rechteck; mst= Strichbreite	kein Bar definiert
Bargraph aktualisieren	ESC	V	B	nr			Alle definierten Bargraphs für Analogkanal nr=1..2 aktualisieren								
<b>Benutzerwerte - Formatierte Textausgabe</b>															
Analogabgleich	ESC	V	@	nr	String	NUL		Der Abgleich erfolgt folgendermassen: 1.) Definierte Spannung an AIN1 (Kanal1) oder AIN2 (Kanal2) anlegen. 2.) Befehl unter Angabe des Analogkanals nr=1..2 und "String" ausführen; wobei "String" aus einer ASCII-Zeichenkette wie z.B. "200.0" besteht				nicht kalibriert			
Benutzerwerte einstellen			E	nr	Format String...	NUL		Benutzerwerte für Analogkanal nr=1..2 einstellen. Jeweils 2 Analogwerten (0..200mV) ein Benutzerwert (max. 5 Stellen + Dezimalpunkt + evtl. Vorzeichen) zugeordnet. Format String: mV1=Benutzerwert1;mV2=Benutzerwert2. 'NUL' (\$00) = Stringende Beispiel: Anzeige soll bei 200 mV "-123.45" und bei 100mV "0.00" sein Format String: "200=-123.45;100=0"				wird 0=0; 200=200.0			
Benutzerwert senden			S	nr	aktuellen Benutzerwert für Analogkanal nr=1..2 über RS-232/RS-422 senden										
Benutzerwert an Terminal Benutzerwert ausgeben			T G	nr	xx1	yy1	aktuellen Benutzerwert für Analogkanal nr=1..2 zum Terminal ausgeben aktuellen Benutzerwert für Analogkanal nr=1..2 an xx1,yy1 ausgeben								
Benutzerwert Farbe	ESC	F	V	nr	vf	hf	Farbe für Analogkanal nr=1..2 einstellen. vf=Schriftfarbe; hf=Hintergrundfarbe				7 / 0				
Benutzerwert Font	ESC	V	F	nr	n1	Font für Analogkanal nr=1..2 mit der Nummer n1 einstellen					5				
Benutzerwert Zoom			Z	nr	n1	n2	Zoomfaktor für Analogkanal nr=1..2 einstellen. n1=X-Zoom 1x..8x; n2=Y-Zoom 1x..8x				1,1				
Benutzerwert Muster			M	nr	n1	Text für Analogkanal nr=1..2 mit Muster n1 verknüpfen; 0=mit keinem Muster verknüpfen						0			
Benutzerwert Winkel			W	nr	n1	Analogkanal nr=1..2 Schriftwinkel: n1=0: 0°; n1=1: 90°; n1=2: 180°; n1=3: 270°;						0			

Technische Änderungen vorbehalten. Wir übernehmen keine Haftung für Druckfehler.

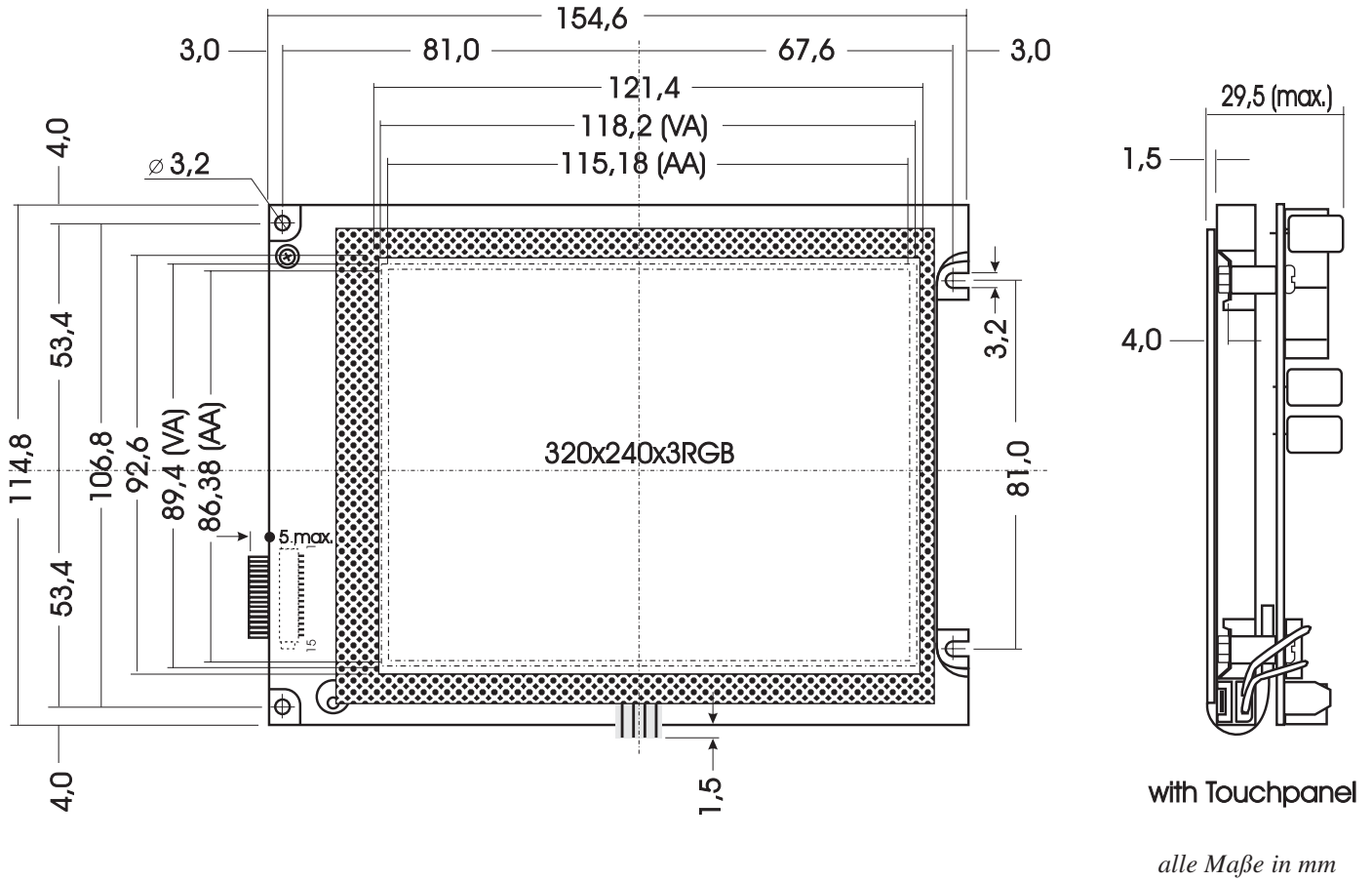
### ANTWORTEN DES EAKIT320F-8 ÜBER DIE SERIELLE SCHNITTSTELLE

Die nachfolgende Tabelle enthält alle Antworten des EA KIT320F-8 über die serielle Schnittstelle. Die Antworten können automatisch oder per Befehl kommen. Zudem können mit dem Befehl 'ESC SB ...' beliebige Zeichenfolgen über die serielle Schnittstelle gesendet werden.

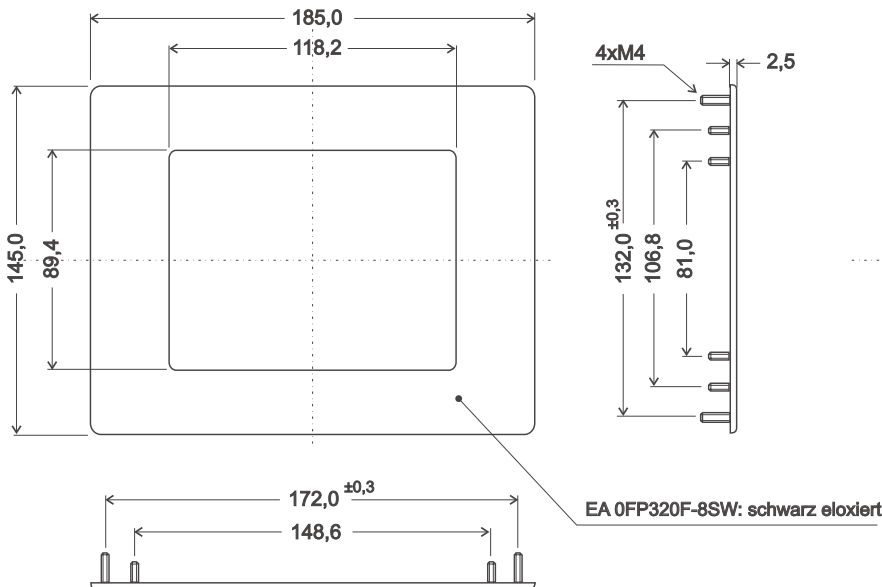
Antworten des EA KIT320F-8 über die serielle Schnittstelle									
Kennung		anz	daten				Anmerkung		
<b>automatische Antworten vom KIT320</b>									
ESC	A	1	code				Antwort vom Analogen Touchpanel wenn eine Taste/Schalter gedrückt wurde. code = down oder up Code der Taste/Schalter. Es wird nur gesendet 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	T	0					Falls das automatische Öffnen eines Touchmenüs deaktiviert ist (siehe Befehl 'ESC T n1'), so wird diese Anforderung an den Hostrechner gesendet. Dieser kann dann das Touchmenü mit dem Befehl 'ESC N T 2' öffnen.		
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 Eingans nr		
ESC	D	3	nr	LO-byt wert	HI-byt wert			Nach dem Befehl 'ESC S D nr' wird der aktuelle Analogwert vom Analogeingang nr=1 oder 2 gesendet. (wert = 0..2300 entspricht 0..230mV)	
ESC	V	anz	nr	Zeichenkette Benutzerwert ...				Nach dem Befehl 'ESC V S nr' wird der aktuelle Analogwert vom Analogeingang nr=1 oder 2 als formatierter Benutzerwert gesendet (Stringlänge = anz-1).	
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			+ 77876 Bytes Bilddaten				Nach dem Befehl 'ESC H' wird ein Hardcopy mit 77878 Bytes gesendet. Das ist ein 256 Farben BMP-Bild mit 320x240 Pixel. Die ersten beiden Bytes des BMP-Bildes beginnen 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 beginnen immer mit 'EA'	

# EA KIT320F-8 COLOR

## ABMESSUNGEN

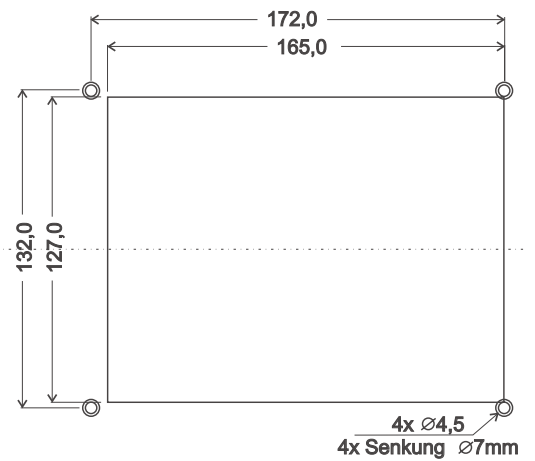


## FRONTPANEL EA 0FP320F-8SW



## PANEL CUT OUT

(passt auch für s/w Version EA KIT320-8CTP)



alle Maße in mm