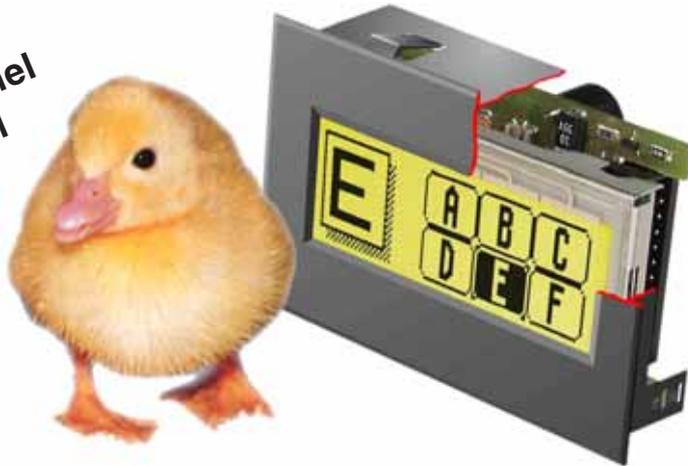


# BEDIENEINHEIT MIT FONTS, GRAFIKBEFEHLEN UND MAKROS

Touch Panel  
optional



EA KIT120-5LEDTP  
Abmessungen 77x54mm

## TECHNISCHE DATEN

- \* LCD GRAFIKDISPLAY MIT DIVERSEN GRAFIKFUNKTIONEN UND FONTS
- \* 120x32 PIXEL MIT LED-BELEUCHTUNG GN/GB
- \* FONT ZOOM VON ca. 3mm ÜBER ca. 8mm BIS ZU ca. 20mm
- \* SNAP-IN GEHÄUSE FÜR EXTREM EINFACHEN EINBAU
- \* LED-HINTERGRUNDBELEUCHTUNG ÜBER SOFTWAREBEFEHL SCHALTBAR
- \* VERSORGUNGSSPANNUNG 5V /typ. 23mA (LED EIN: 180mA) ODER 9..36V OPTIONAL
- \* RS-232 MIT BAUDRATEN 1.200..115.200 BD
- \* **PIXELGENAUE** POSITIONIERUNG BEI ALLEN FUNKTIONEN
- \* PROGRAMMIERUNG ÜBER HOCHSPRACHENÄHNLICHE BEFEHLE:
- \* GERADE, PUNKT, BEREICH, UND/ODER/EXOR, BARGRAPH...
- \* BIS ZU 256 BITMAP BILDER/ICONS IM SYSTEM SPEICHERBAR
- \* BIS ZU 256 MAKROS PROGRAMMIERBAR (32kB INKL. FONTS UND BILDER)
- \* TEXT UND GRAFIK MISCHEN
- \* 5 DIGITALE EIN- UND 5 AUSGÄNGE
- \* BETRIEBSTEMPERATUR -20..+70°C TEMPERATURKOMPENSATION ON-BOARD

## ZUBEHÖR

- \* INTEGRIERTES TOUCH PANEL MIT 5x2 FELDERN (ENTSPIEGELT, KRATZFEST)
- \* DISKETTE FÜR MAKROPROGRAMMIERUNG (PC-DOS): **EA DISK240**

## BESTELLBEZEICHNUNG

120x32 DOTS MIT LED-BELEUCHTUNG, GB/GN  
 120x32 DOTS MIT TOUCH PANEL, LED-BEL., GB/GN  
 VERSORGUNG 9..36V STATT 5V  
 OPTOKOPPLER FÜR DIE 5 EIN- UND AUSGÄNGE  
 KABEL (1,5m) FÜR 9-POL. SUB-D SOWIE DER IO-PORTS

**EA KIT120-5LED**  
**EA KIT120-5LEDTP**  
**EA OPT-9/36V**  
**EA OPT-OPTO10**  
**EA KV24-9B10**

### ALLGEMEINES

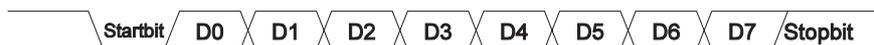
EA KIT120 ist eine komplett aufgebaute Steuer- und Bedieneinheit mit diversen eingebauten Funktionen. Das kompakt aufgebaute Display bietet zusammen mit dem sehr guten Supertwistkontrast eine sofort einsetzbare Einheit. Die Ansteuerung erfolgt über die Standard Schnittstelle RS-232. Die Grafikeinheit 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. Der Einbau erfolgt durch einfaches Einstecken in den Frontplattendurchbruch.

### HARDWARE

Die Grafikeinheit ist für +5V Betriebsspannung ausgelegt. Optional ist eine Versorgung mit 9..36V möglich. Die Datenübertragung erfolgt seriell asynchron im RS-232 Format. Das Übertragungsformat ist fest auf 8 Datenbits, 1 Stopbit, no Parity eingestellt. Die Baudrate kann mit Hilfe eines PC's von 1200 Baud bis zu 115.200 Baud eingestellt werden. Handshakeleitungen RTS und CTS stehen zur Verfügung.

Datenformat:



### TOUCH PANEL

Die Version EA KIT120-5LEDTP ist mit einem integrierten Touch Panel ausgerüstet. Durch Berühren des Displays können hier Eingaben gemacht werden. Die Beschriftung der "Tasten" ist flexibel und auch während der Laufzeit änderbar (verschiedene Sprachen, Icons). Das Zeichnen der einzelnen "Tasten", sowie das Beschriften oder Zusammenfassen mehrerer Felder 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** plazieren. Das Mischen von Text und Grafik ist jederzeit möglich. Es können bis zu 16 verschiedene Zeichensätze verwendet werden. Jeder Zeichensatz kann wiederum 2- bis 4-fach gezoomt werden. Mit dem größten Zeichensatz 16x8 lassen sich somit bei 2-fach Zoom (=16x32) bildschirmfüllende Worte und Zahlen darstellen.

### ZUBEHÖR

#### Diskette zur Makroerstellung und zur Baudrateneinstellung

Zur Makroprogrammierung ist eine Diskette EA DISK240<sup>\*)</sup> erforderlich. Diese übersetzt die in eine Textdatei eingegebenen Befehle in einen für die Grafikeinheit lesbaren Code und brennt diesen dauerhaft ins EEPROM.

Falls eine andere Baudrate als die im Auslieferungszustand eingestellten 9.600 Baud gewünscht werden, ist die Diskette ebenso erforderlich.

#### Kabel für PC

Für die einfache Anbindung an PC's (Makroprogrammierung) liefern wir als Zubehör ein ca. 1,5m langes Kabel mit 9-pol. SUB-D Stecker (female) EA KV24-9B10. Einfach an die COM 1 oder COM 2 anstecken und loslegen. Ausserdem ist bei diesem Kabel ein 10-polige Buchsenleiste für die 5 Ein- und Ausgänge (mit C-MOS Pegel) vorhanden.

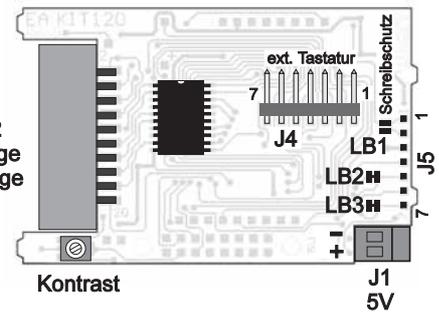
<sup>\*)</sup> auch im Internet unter <http://www.lcd-module.de/deu/disk/disk240.zip>

## ELECTRONIC ASSEMBLY

### EXTERNETASTATUS (NUR EA KIT120-5LED)

Am Steckanschluss J4 kann eine Tastatur (einzelne Tasten bis zur 5x2 Matrix-Tastatur) angeschlossen werden. Die angeschlossenen Tasten werden dabei per Software entprellt. Bitte beachten Sie, daß der Anschluß einer externen Tastatur nur bei der Version EA KIT120-5LED ohne integriertem Touch Panel und ohne zusätzliche Optionen (OPT-OPTO10= bzw. OPT-9/36V) möglich ist.

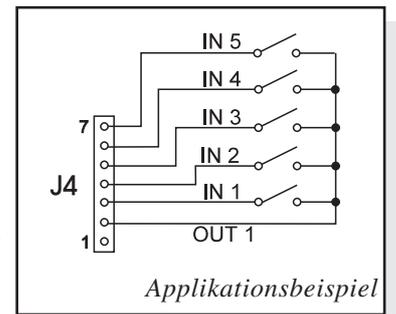
Jede Taste wird zwischen einem Ausgang und einem Eingang geschaltet. Die Eingänge sind mit 50kΩ Pullup abgeschlossen. An jeden Ausgang können bis zu 5 Tasten angeschlossen werden.



Tastaturanschluß J4		
Pin	Symbol	Funktion
1	OUT 2	Ausgang Zeile 2
2	OUT 1	Ausgang Zeile 1
3	IN 1	Eingang Spalte 1
4	IN 2	Eingang Spalte 2
5	IN 3	Eingang Spalte 3
6	IN 4	Eingang Spalte 4
7	IN 5	Eingang Spalte 5

#### Senden der Tastendrücke

Bei jedem Druck einer Taste wird die dazugehörige Tastennummer (1..10) gesendet. Das Loslassen der Taste wird nicht gesendet. Soll auch das Loslassen gesendet werden, so kann das über die Definition des Touch Makros Nr.0 realisiert werden. Der automatische Tastaturscan läßt sich über den Befehl "ESC T A 0" deaktivieren.



Die Tastennummer kann folgendermaßen bestimmt werden: **(Ausgang -1) \* 5 + Eingang** (Ausgangs: die Zahl 1 oder 2, Eingang: zwischen 1 und 5).

Hinweis Falls die Handshakeleitung (z.B. CTS) das Senden nicht erlaubt, können Tastendrücke verloren gehen.

### TOUCH PANEL (NUR EA KIT120-5LEDTP)

Die Version EA KIT120-5LEDTP wird mit einem integrierten Touch Panel mit 10 Feldern geliefert. Die Grafikeinheit unterstützt dieses Touch Panel mit komfortablen Befehlen. So können z.B. mehrere Touch-Felder zu einer großen Gesamt-Taste zusammengefasst, die Taste gezeichnet und eine Beschriftung der Taste erfolgen. Ebenso kann dieser eben definierten Taste ein Return-Code (1..255) zugewiesen werden. Wird der Return-Code 0 zugewiesen, so ist die Taste deaktiviert und wird bei Betätigung nicht gemeldet.

Beim Berühren der Touch-Tasten können diese automatisch invertiert werden und ein Summer signalisiert die Berührung. Gleichzeitig wird das interne Touch Makro mit der Nummer des Return-Codes gestartet oder, falls kein Touch Makro definiert ist, der Return-Code der Taste über die serielle Schnittstelle gesendet.

#### Beispiel:

Definieren einer Taste von Feld 4 bis 10, mit dem Return-Code 65='A' und dem Text "STOP". Anmerkung: Vor der Definition einzelner Tasten sollten alle Felder durch "ESC T R" deaktiviert sein.



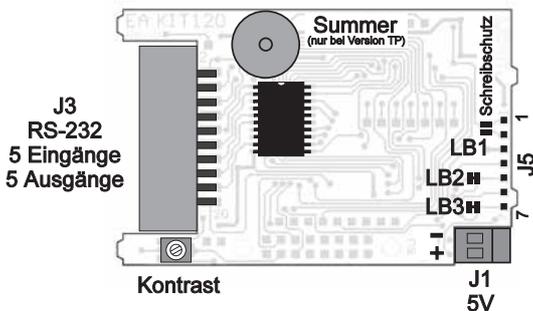
Beispiel	Auszugebende Codes										Bemerkung		
für Compiler	#TH 4, 10, 'A', 2, "STOP"										Die Endekennung 0 wird hier nicht angegeben! die Punkte '.' stehen für nicht darzustellende ASCII-Zeichen		
als ASCII	ESC	T	H	.	.	A	.	S	T	O		P	.
in Hex	\$1B	\$54	\$48	\$04	\$0A	\$41	\$02	\$53	\$54	\$4F		\$50	\$00
in Dezimal	27	84	72	4	10	65	2	83	84	79		80	0
Befehlskennung	Einleitung Touch-Befehl	horizontale Beschriftung linke oberes Touchfeld	rechtes untere Touchfeld	Return Code	Taste zeichnen mit Rahmen							Text Ende Kennung	



### VERSORGUNG/RS-232 ANSCHLUSS

In der Standardausführung wird die Versorgungsspannung von +5V über die Schraubklemme J1 eingespeist. Alternativ können die 5V auch über J3 (Pin1 und Pin 10) angeschlossen werden.

**Achtung:** Unbedingt auf die richtige Polarität achten!



An der 20 poligen Stiftleiste J3 ist der Anschluss für die RS-232 Schnittstelle ( $\pm 12V$ ) sowie der 5 Ein- und Ausgänge (5V C-MOS Pegel). Die Baudrate ist nach Auslieferung auf 9600 Baud voreingestellt. Sie läßt sich über das Programm KITBAUD.EXE (erhältlich

auf EA DISK240\*) konfigurieren, dazu muß das KIT an COM1 oder COM2 angeschlossen sein und die neue Baudrate als Parameter übergeben werden (z.B.

5V RS-232 Anschluss J5			
Pin	Symbol	In/Out	Funktion
1	WP	In	EEProm Schreibschutz
2	VDD		+5V Versorgung
3	GND		0V Masse
4	TXD5	Out	Transmit data CMOS
5	RXD5	In	Receive data CMOS
6	RTS5	Out	Request To Send CMOS
7	CTS5	In	Clear To send CMOS

KITBAUD 19200). Folgende Baudraten sind einstellbar: 1200, 2400, 4800, 9600, 19200, 38400, 56700 und 115200.

Bitte beachten Sie, daß der interne Datenpuffer lediglich 32 Byte umfaßt. Deshalb müssen 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.

Werden die RS-232 Daten als 5V Pegel an J5 eingespeist, müssen unbedingt die Lötbrücken LB2 und LB3 geöffnet werden.

### SCHREIBSCHUTZFÜR MAKROPROGRAMMIERUNG

Durch Schließen der Lötbrücke LB1 (Verbindung WP-VDD) läßt sich ein versehentliches Überschreiben der einprogrammierten Makros, Bilder und Fonts verhindern. Auch die Baudrate ist dann nicht mehr einstellbar.

### EIN-UND AUSGÄNGE

Das KIT wird mit 5 digitalen Ein- und 5 Ausgängen geliefert (5V CMOS Pegel, nicht potentialfrei). Der Anschluß erfolgt an der 20-poligen Steckerleiste J3.

**5 Ausgänge** Jede Leitung kann per Befehl "ESC Y W" individuell angesteuert werden. Pro Leitung kann ein Strom von max. 6mA geschaltet werden.

**5 Eingänge** Die Eingänge können direkt über die serielle Schnittstelle abgefragt und ausgewertet werden ("ESC Y R"). Jede Änderung der Logikpegel 0V oder 5V an den Eingängen kann ein internes Portmakro starten. Durch die Kombinationsmöglichkeit von 5 Leitungen sind somit bis zu 32 Portmakros ansprechbar. Jedes dieser Portmakros kann seinerseits den Bildschirminhalt ändern oder auch einen Ausgang schalten. Damit können vielfältige Steuerungsaufgaben gelöst werden. Für die Erstellung der Portmakros benötigt man einen PC und die Diskette EA DISK240. Eine genauere Beschreibung dazu lesen Sie auf der Seite 6. Die automatische Portabfrage läßt sich mit dem Befehl "ESC Y A 0" deaktivieren.

**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 wird dieser als High (ca. 50 kOhm PullUp) ausgewertet.

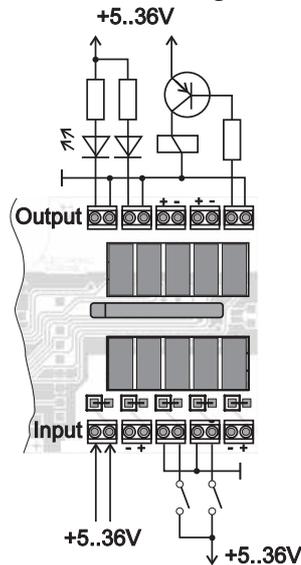
RS-232 / PORT Anschluß J3					
Pin	Symbol	In/Out	Funktion	EA KV24-9B10	
1	VDD	-	+5V Versorgung	Pin	Pin
2	DCD	-	Brücke nach DTR	1	
3	DSR	-	Brücke nach DTR	6	
4	TxD	Out	Transmit Data	2	
5	CTS	In	Clear To Send	7	
6	RxD	In	Receive Data	3	9-pol. SUB-D
7	RTS	Out	Request To Send	8	
8	DTR	-	siehe Pin 2, Pin 3	4	
9	-	-	NC	9	
10	GND	-	0V Masse	5	
11	IN1	In	CMOS-Eingang 1		1
12	OUT1	Out	CMOS-Ausgang 1		2
13	IN2	In	CMOS-Eingang 2		3
14	OUT2	Out	CMOS-Ausgang 2		4
15	IN3	In	CMOS-Eingang 3		5
16	OUT3	Out	CMOS-Ausgang 3		6
17	IN4	In	CMOS-Eingang 4		7
18	OUT4	Out	CMOS-Ausgang 4		8
19	IN5	In	CMOS-Eingang 5		9
20	OUT5	Out	CMOS-Ausgang 5		10

\*) auch im Internet unter <http://www.lcd-module.de/disk/disk240.zip>

## ELECTRONIC ASSEMBLY

### EIN-UND AUSGÄNGE MIT OPTOKOPPLERN (AUSFÜHRUNG MIT EA OPT-OPTO10)

Das KIT ist optional mit Optokoppler Ein- und Ausgängen lieferbar (EA OPT-OPTO10). Alle Ein- und Ausgänge sind sowohl von der restlichen Elektronik als auch untereinander isoliert. Der Anschluß erfolgt über 10 einzelne Schraubklemmen.

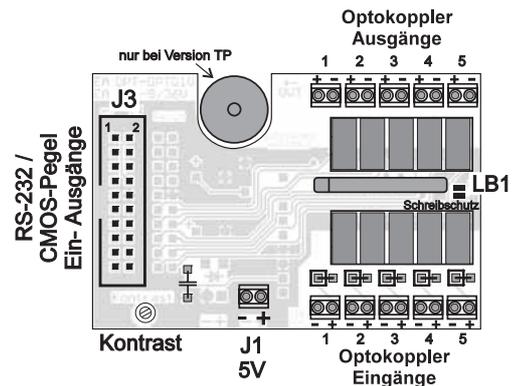


#### 5 Optokoppler Ausgänge

Als Ausgang ist jeweils der Collector und Emitter eines Transistors an den Schraubklemmen herausgeführt. Jeder Ausgang kann max. 10mA schalten. Beachten Sie die Polartität und den lastabhängigen Spannungsabfall des Transistors von 0,6..5V.

#### 5 Optokoppler Eingänge

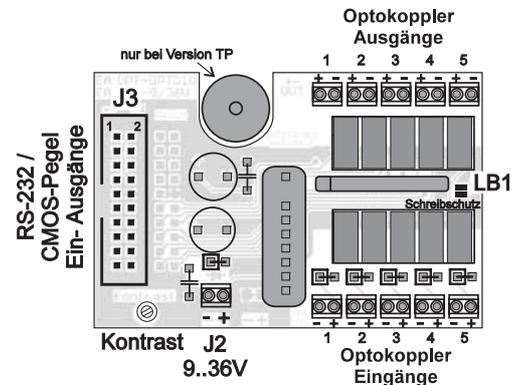
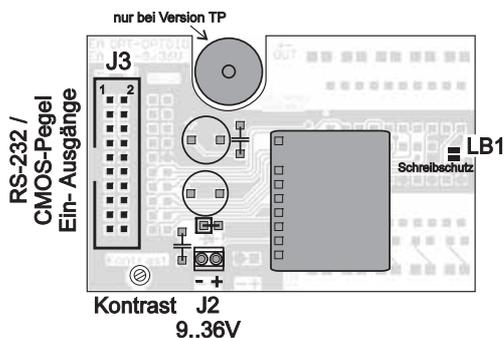
An alle 5 Eingänge können direkt Spannungen von 4..36V angelegt werden. Spannungen über 4V werden als H-Pegel erkannt, Spannungen unter 2V gelten als L-Pegel. Spannungen zwischen 2 und 4V sind undefiniert. Beachten Sie die Polarität der Eingänge (LEDs).



**Anmerkung:** Die Optokoppler invertieren die Eingangslogik (alle Eingänge offen: Portmakro 31). Hier empfiehlt es sich (z.B. im Power-On-Makro) mit dem Befehl "ESC Y I 1" die Eingänge invertiert auszuwerten (alle Eingänge offen: Portmakro 0).

### EA OPT-9/36V: VERSORGUNGSSPANNUNG 9-36V

Liegt die Version für 9 bis 36V Versorgungsspannung vor (EA OPT-9/36V), so erfolgt die Stromversorgung über J2.



### GRUNDEINSTELLUNGEN

Nach dem Einschalten bzw. nach einem manuell ausgelösten Reset werden die nebenstehenden Register auf einen bestimmten Wert voreingestellt. Beachten Sie bitte, daß alle Einstellungen durch Erstellen eines Power-On-Makros (Normal-Makro Nr.0) überschrieben werden können.



Grundeinstellungen		
Register	Befehl	nach Power-On / Reset
Text-Modus	ESC L	setzen,schwarz
Terminal Font	ESC FT	Font 3, kein Zoom
Cursor	ESC QC	ein
Blinkzeit	ESC QZ	0,6 sek.
Selbst definierte Zeichen	ESC E	undefiniert
Grafik-Modus	ESC V	setzen
Grafik Font	ESC F	Font 3, kein Zoom
Last xy	ESC W	(0;0)
Bargraph 1..4	ESC B	undefiniert
Selekt/Deselekt	ESC K	selektiert
LED-Beleuchtung	ESC YL	aktiviert
Ausgänge OUT1..5	ESC Y	H-Pegel / offen

## MAKROPROGRAMMIERUNG

Einzelne oder mehrere Befehlsfolgen können als sog. Makros zusammengefasst und im EEPROM fest abgespeichert werden. Diese können dann mit den Befehlen *Makro ausführe* gestartet werden. Es gibt 4 verschiedene Makrotypen:

### Touch Makro (1..255)(0)

Start bei Berührung eines Touchfeldes (nur bei Versionen mit Touch Panel TP) oder bei Betätigung einer ext. angeschlossenen Taste/Matrixtastatur. Das Touch Makro Nr.0 hat eine Sonderstellung: Beim Loslassen einer x-beliebigen Taste wird das Touch Makro Nr.0 gestartet.

### Port Makro (0..31)(32..255)

Start bei einer Änderung des Logikpegels an den Eingänge IN 1..5. Die restlichen Portmakros (32..255) stehen frei zur Verfügung und können per Befehl 'ESC M P nr' gestartet werden.

### Normal Makro (1..255)

Start per Befehl 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).

### Power-On-Makro = Normal Makro (0)

Das Normal Makro Nr.0 hat eine Sonderstellung: es wird automatisch nach dem Einschalten ausgeführt. Hier kann man z.B. den Cursor abschalten und einen Startbildschirm definieren.

## 256 BILDER FEST ABGELEGT

Um Übertragungszeiten der seriellen Schnittstelle zu verkürzen, oder auch um Speicherplatz im Prozessorsystem zu sparen, können bis zu 256 Bilder im internen EEPROM abgelegt werden. Der Aufruf erfolgt über den Befehl "ESC U E" über die serielle Schnittstelle oder aus einem Touch-/Port-/Normal-Makro heraus. Verwendet werden können alle monochromen Bilder im Windows BMP Format. Die Erstellung und Bearbeitung erfolgt über Standardsoftware wie z.B. Windows Paint.

## ERSTELLEN INDIVIDUELLER MAKROS

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

- die Diskette EA DISK240<sup>\*)</sup>; sie enthält einen Compiler, Beispiele und Fonts
- einen PC mit serieller Schnittstelle COM1 oder COM2, mit ca. 500kB Platz auf der Festplatte
- einen Texteditor wie z.B. WordPad, Norton Editor 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 EEPROM-Datei DEMO.EEP, welche dann automatisch mit der eingetragenen Baudrate in das Display-EEPROM 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 vielen Beispielen auf der Diskette EA DISK240<sup>\*)</sup> unter dem Namen DOKU.DOC (für WORD) bzw. DOKU.TXT (DOS).

```

;Makro Demo
KIT120-5                ; Kittyp festlegen
COM2: 115200            ; KIT ist an COM2 angeschlossen,
                       ; Übertragung mit 115.200 Baud

;-----
;Konstanten definieren
AUS = 0
EIN = 1
FONT4x6 = 1
FONT5x6 = 2
FONT6x8 = 3
FONT8x8 = 4
FONT8x16 = 5
;-----
;Fonts einbinden
Font: FONT4x6, 32, 95  INTERN4x6
Font: FONT5x6, 32, 158 INTERN5x6
Font: FONT6x8, 32, 158 INTERN6x8
Font: FONT8x8, 32, 158 INTERN8x8
Font: FONT8x16, 32, 158 INTERN8x16
;-----
Makro: 0                ; Power-On/Reset Makro
#QC EIN                 ; Cursor sichtbar
#FT FONT8x16            ; Terminalfont einstellen
#UL 0, 20, <EA2.BMP>   ; ELECTRONIC ASSEMBLY Logo

```

<sup>\*)</sup> auch im Internet unter <http://www.lcd-module.de/deu/disk/disk240.zip>

### INTEGRIERTE FONTS

Nr.	Zeichen höhe	Zeilen x Zeichen	Größe in Pixel	ASCII-Bereich	Frei def. ASCII-Codes	Bemerkung
1	3,3 mm	5 x 30	4 x 6	32 - 95	1..16	Microschrift
2	3,3 mm	5 x 24	5 x 6	32 - 158	1..12	Minischrift
3	4,5 mm	4 x 20	6 x 8	32 - 158	1..10	Normalschrift
4	4,5 mm	4 x 15	8 x 8	32 - 158	1..8	Fettschrift
5	7,8 mm	2 x 15	8 x 16	32 - 158	1..4	Großschrift

Im EA KIT120-5 sind standardmäßig 5 Zeichensätze integriert. Jeder Zeichensatz kann in 1- bis 4-facher Höhe verwendet werden. Unabhängig davon läßt sich auch die Breite verdoppeln bis vervierfachen.

Zusätzlich können, je nach Font, bis zu 16 eigene Zeichen definiert werden die solange erhalten bleiben, bis die Versorgungs-spannung abgeschaltet wird. (Siehe Befehl ESC E).

Jedes Zeichen kann **pixelgenau** plaziert 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 eine 90° Drehung (vertikaler Einbau des Displays) ist möglich.

Die Makroprogrammierung erlaubt die Einbindung von weiteren 11 Fonts, sowie die komplette Umgestaltung der einzelnen Zeichen.

Durch einen Fonteditor auf der Diskette EA DISKFONT1520 können alle nur erdenklichen Schriften mit bis zu 16x16 Pixeln Größe erstellt und einprogrammiert werden.

### TIP: SCHRIFTEFFEKTE

Mit dem Befehl 'ESC L' TEXT-Modus (Verknüpfung, Muster) können bei grossen Schriften interessante Effekte durch Überlagerung (mehrmaliges versetztes Schreiben eines Wortes) erzielt werden.

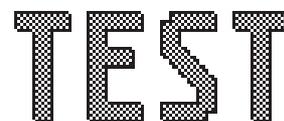


Originalschrift 8x16 mit ZOOM 2 an Position 0,0 mit Muster Schwarz

Durch Überlagerung (EXOR) an Pos.1,1 entstandene "Outline Schrift"



Nochmalige Überlagerung (EXOR) der "Outline Schrift" an Pos.2,2. führt zu einer "Outline Schrift mit Füllung"



Überlagerung (ODER) mit Muster 50% Grau der "Outline Schrift" an Pos.0,0. führt zu einer "Schrift mit Musterfüllung"

+ Lower	\$0	\$1	\$2	\$3	\$4	\$5	\$6	\$7	\$8	\$9	\$A	\$B	\$C	\$D	\$E	\$F
Upper	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)
\$20 (dez: 32)		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
\$30 (dez: 48)		0	1	2	3	4	5	6	7	8	9	:	;	<	=	>
\$40 (dez: 64)		@	A	B	C	D	E	F	G	H	I	J	K	L	M	N
\$50 (dez: 80)		P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^

Font 1: 4x6

+ Lower	\$0	\$1	\$2	\$3	\$4	\$5	\$6	\$7	\$8	\$9	\$A	\$B	\$C	\$D	\$E	\$F
Upper	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)
\$20 (dez: 32)		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
\$30 (dez: 48)		0	1	2	3	4	5	6	7	8	9	:	;	<	=	>
\$40 (dez: 64)		@	A	B	C	D	E	F	G	H	I	J	K	L	M	N
\$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	ë	ë	ë	ë	ë	ë	ë	ë	ë	ë	ë	ë	ë
\$90 (dez: 144)		ê	ê	ê	ê	ê	ê	ê	ê	ê	ê	ê	ê	ê	ê	ê

Font 3: 6x8

+ Lower	\$0	\$1	\$2	\$3	\$4	\$5	\$6	\$7	\$8	\$9	\$A	\$B	\$C	\$D	\$E	\$F
Upper	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)
\$20 (dez: 32)		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
\$30 (dez: 48)		0	1	2	3	4	5	6	7	8	9	:	;	<	=	>
\$40 (dez: 64)		@	A	B	C	D	E	F	G	H	I	J	K	L	M	N
\$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	ë	ë	ë	ë	ë	ë	ë	ë	ë	ë	ë	ë	ë
\$90 (dez: 144)		ê	ê	ê	ê	ê	ê	ê	ê	ê	ê	ê	ê	ê	ê	ê

Font 5: 8x16

## ALLE BEFEHLE AUF EINEN BLICK

Nach dem Einschalten blinkt der Cursor in der ersten Zeile und das Display ist empfangsbereit. Alle ankommenden Zeichen werden als ASCII's dargestellt (Ausnahme: CR, LF, FF, ESC). Der Zeilenvorschub erfolgt automatisch oder durch das Zeichen 'LF'. Ist das Display voll, so wird der Cursor wieder auf Position 1,1 gesetzt. Beim Zeichen 'FF' (Seitenvorschub) wird das Display gelöscht.

Alle zusätzlichen Befehle (Cursor positionieren, Font einstellen, Grafikbefehle, ...) werden über das ESCAPE-Zeichen (dez 27/ hex 1B) eingeleitet.

Befehlstabelle für EA KIT120										
Befehl	Codes					Anmerkung				
<b>Befehle für den Terminal Betrieb</b>										
Formfeed FF (dez:12)	^L									Bildschirm wird gelöscht und der Cursor nach Pos. (1,1) gesetzt
Carriage Return CR(13)	^M									Cursor ganz nach links zum Zeilenanfang
Linefeed LF (dez:10)	^J									Cursor 1 Zeile tiefer, falls Cursor in letzter Zeile dann auf 1. Zeile setzen
Cursor On / Off	ESC	Q	C	n1						n1=0: Cursor ist unsichtbar; n1=1: Cursor blinkt (invers 6/10s);
Cursor positionieren	ESC	O	n1	n2						n1=Spalte; n2=Zeile; Ursprung links oben ist (1,1)
Terminal Font einstellen	ESC	F	T	n1						n1=1: Font Nr. n1 (1..16) für Terminal Betrieb einstellen
<b>Befehle zur Textausgabe</b>										
Text-Modus	ESC	L	n1	mst						Modus n1: 1=setzen; 2=löschen; 3=invers; 4=Replace; 5=Invers Replace; mst: Muster Nr. 0..7 verwenden;
Font einstellen	ESC	F	n1	n2	n3					Font mit der Nummer n1 (1..16) einstellen; n2=X- n3=Y-Zommfaktor (1x..4x);
Zeichenkette horizontal ausgeben	ESC	Z	L Z R	x1	y1	Text ...	NUL			Eine Zeichenkette (...) an x1,y1 ausgeben. 'NUL' (\$00)=Zeichenkettenende; Mehrere Zeilen werde durch das Zeichen ' ' (\$7C, dez: 124) getrennt; 'L':= Linkbündig an x1; 'Z':= Zentriert an x1; 'R':= Rechtsbündig an x1; y1 ist immer die Oberkannte der Zeichenkette
Zeichenkette 90° gedreht (vertikal) ausgeben	ESC	Z	O M U	x1	y1	Text ...	NUL			Eine Zeichenkette (...) um 90° gedreht an x1,y1 ausgeben; 'NUL' (\$00)=Ende; Mehrere Zeilen werde durch das Zeichen ' ' (\$7C, dez: 124) getrennt; 'O':= Oben-Bündig an y1; 'M':= Mittig an y1; 'U':= Unten-Bündig an y1; x1 ist immer die Rechte Kannte der Zeichenkette
Zeichen definieren	ESC	E	n1					daten ...		n1=Zeichen Nr.; daten=Anzahl Bytes je nach akt. Font
<b>Befehle zum Zeichnen</b>										
Grafik-Modus	ESC	V	n1							Zeichenmodus einstellen für die Befehle: 'Punkt setzen', 'Gerade zeichnen', 'Rechteck', 'Rundeck' und 'Bereich mit Füllmuster' n1: 1=setzen; 2=löschen; 3=invers; 4=Replace; 5=Invers Replace;
Punkt setzen	ESC	P	x1	y1						Ein Pixel an die Koordinaten x1, y1 setzen
Gerade zeichnen	ESC	G	x1	y1	x2	y2				Eine Gerade von x1,y1 nach x2,y2 zeichnen
Gerade weiter zeichnen	ESC	W	x1	y1						Eine Gerade vom letzten Endpunkt bis x1, y1 zeichnen
<b>Rechteck Befehle</b>										
Rechteck zeichnen	ESC	R	R	x1	y1	x2	y2			Ein Rechteck (Rahmen) von x1,y1 nach x2,y2 zeichnen
Rundeck zeichnen			N	x1	y1	x2	y2			Ein Rechteck mit runden Ecken von x1,y1 nach x2,y2 zeichnen
Bereich löschen			L	x1	y1	x2	y2			Einen Bereich von x1,y1 nach x2,y2 löschen (alle Pixel aus)
Bereich invertieren			I	x1	y1	x2	y2			Einen Bereich von x1,y1 nach x2,y2 invertieren (alle Pixel umkehren)
Bereich füllen			S	x1	y1	x2	y2			Einen Bereich von x1,y1 nach x2,y2 füllen (alle Pixel ein)
Bereich m. Füllmuster			M	x1	y1	x2	y2	mst		Einen Bereich von x1,y1 nach x2,y2 mit Muster mst (0..7) zeichnen
Box zeichnen			O	x1	y1	x2	y2	mst		Ein Rechteck mit Füllmuster mst (0..7) zeichnen; (immer Replace)
Rundbox zeichnen			J	x1	y1	x2	y2	mst		Ein Rundeck mit Füllmuster mst (0..7) zeichnen; (immer Replace)
<b>Bitmap Bilder Befehle</b>										
Bild aus EEPROM	ESC	U	E	x1	y1	nr				internes Bild mit der nr (0..255) aus dem EEPROM nach x1,y1 laden
Bild laden			L	x1	y1		daten ...			Ein Bild nach x1,y1 laden; daten des Bildes siehe Bildaufbau
Hardcopy senden			H	x1	y1	x2	y2			Es wird ein Bild angefordert. Zuerst werden die Breite und Höhe in Pixel und dann die eigentlichen Bilddaten über RS232 gesendet.

Bargraph Befehle													
Bargraph definieren	ESC	B	R L O U	nr	x1	y1	x2	y2	aw	ew	mst	Bargraph nach L(inks), R(echts), O(ben), U(nten) mit der 'nr' (1..4) definieren. x1,y1,x2,y2 sind das umschließende Rechteck des Bargraphs. aw,ew sind die Werte für 0% und 100%. mst=Muster (0..7)	
Bargraph zeichnen				nr	wert							Den Bargraph mit der Nummer nr (1..4) auf den neuen Benutzer- 'wert' setzen	
Tastatur / Touch-Panel Befehle													
Touch-Taste mit horizontaler Beschriftung definieren	ESC	T	H	f1	f2	Ret. Code	Form	Text ...	NUL	Die Touch-Felder f1 bis f2 (gegenüberliegenden Eckfelder), werden zu einer Touch-Taste mit dem Rückgabewert 'Ret. Code' (=1..255) zusammengefasst (Ret.Code=0 Touch-Taste nicht aktiv). 'Form': Touch-Taste (=0 nichts; =1 löschen; =2 mit Rahmen) zeichnen 'Text': es folgt eine Zeichenkette die zentriert mit dem akt. Font in der Touch-Taste plaziert wird, mehrzeilige Texte werden mit dem Zeichen ' ' (\$7C, dez: 124) getrennt; Zeichen NUL (\$00) = Zeichenkettenende			
Touch-Taste mit vertikaler (90° gedreht) Beschriftung definieren			V										
Touch-Tasten (P)Reset			P								Alle Touch-Tasten werden aufsteigend aktiviert (Felder mit Code 1..10)		
Touch-Tasten Reaktion			R								Alle Touch-Tasten werden deaktiviert (alle Felder mit Code 0)		
			I	n1							n1=0: kein invertieren beim Berühren der Touch-Taste n1=1: Touch-Taste wird beim Berühren automatisch invertiert		
			S	n1							n1=0: kein Summer beim Berühren einer (Touch-)Taste n1=1: Summer piepst kurz beim Berühren einer (Touch-)Taste		
Touch-Taste Invertieren			M	n1							Die Touch-Taste mit dem Return-Code n1 wird manuell Invertiert		
Taste manuell abfragen			W								Die momentan gedrückte (Touch-)Taste wird auf der RS-232 gesendet		
Tasten-Abfrage Ein/Aus			A	n1							Tastaturabfrage wird n1=0:deaktiviert; n1=1:aktiviert, Tastendrucke werden automatisch gesendet; n1=2:aktiviert, Tastendrucke werden nicht gesendet (mit ESC T W abfragen)		
Kontroll- / Definitions-Befehle													
Automatisch blinkender Bereich (Cursor-Funktion)	ESC	Q	D	x1	y1	x2	y2	Definiert einen Blinkbereich x1,y1 bis x2,y2; Blinkfunktion aktivieren					
			Z	n1	Einstellen der Blinkzeit n1= 1..15 in 1/10s; 0=Blinkfunktion deaktivieren								
			C	n1	Automatisch blinkender Bereich als Cursor für den Terminal Betrieb n1=0: Blinkfunktion deaktivieren; n1=1: Blinkfunktion aktivieren (Invers, 6/10s)								
Selekt / Deselekt	ESC	K	S	adr	Kit mit Adresse n1 aktivieren (n1=255: alle)								
			D	adr	Kit mit Adresse n1 deaktivieren (n1=255: alle)								
			A	adr	Neue Adresse adr zuweisen (z.B. im Power-On Makro)								
Warten (Pause)	ESC	X	n1	n1 Zehntel-Sekunden abwarten bevor der nächste Befehl ausgeführt wird.									
Summer Ein / Aus	ESC	J	n1	n1=0:Summer Aus; n1=1:Summer Ein; n1=2..255:für n1 1/10s lang Ein									
Bytes senden	ESC	S	anz	daten ...				Es werden anz (1..255; 0=256) Bytes auf der RS-232 gesendet daten ... = anz Bytes (z.B Ansteuerung eines externen seriellen Druckers)					
Displayreset	ESC	D	R	Ein Reset der Displaytreiber wird ausgelöst, der Controller arbeitet weiter									
Port-Befehle													
Output-Port schreiben	ESC	Y	W	n1	n2	n1=0: Alle 5 Ausgabe-Ports entsprechend n2 (=5-Bit Binärwert) einstellen n1=1..5: Ausgabe-Port n1 rücksetzen (n2=0); setzen (n2=1); invertieren (n2=2)							
Eingabe-Port lesen			R	n1	n1=0: Alle 5 Eingabe-Ports als 5-Bit Binärwert einlesen n1=1..5: 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								
Eingabe-Port invers			I	n1	Der Eingabe-Port wird n1=0: normal; n1=1: invertiert ausgewertet								
LED-Bel. Ein/Aus			L	n1	LED-Beleuchtung n1=0: LED AUS; n1=1: LED EIN; n1=2: LED INVERTIEREN; n1=3..255: Beleuchtung für n1 Zehntel Sek.. lang einschalten								
Makro Befehle													
Makro ausführen	ESC	M	N	n1	Das (Normal-)Makro mit der Nummer n1 aufrufen (max. 7 Ebenen)								
Touch Makro ausführen			T	n1	Das Touch-Makro mit der Nummer n1 aufrufen (max. 7 Ebenen)								
Port Makro ausführen			P	n1	Das Port-Makro mit der Nummer n1 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						
Display-Befehle (Wirkung auf das gesamte Display)													
Display löschen	ESC	D	L	Displayinhalt löschen (alle Pixel aus)									
Display invertieren	ESC	D	I	Displayinhalt invertieren (alle Pixel umkehren)									
Display füllen	ESC	D	S	Displayinhalt füllen (alle Pixel ein)									
Display ausschalten	ESC	D	A	Displayinhalt wird unsichtbar bleibt aber erhalten, Befehle weiterhin möglich									
Display einschalten	ESC	D	E	Displayinhalt wird wieder sichtbar									

## PARAMETER

Die Grafikeinheit läßt sich über diverse eingebaute Befehle programmieren. Jeder Befehl beginnt mit ESC gefolgt von einem oder zwei Befehlsbuchstaben und einigen Parametern. Alle Befehle und deren Parameter wie Koordinaten und sonstige Übergabewerte werden immer als Bytes erwartet. Dazwischen dürfen keine Trennzeichen z.B. Leerzeichen oder Kommas verwendet werden. Die Befehle benötigen auch **kein Abschlussbyte** wie z.B. Carrige Return (außer Zeichenkette: \$00).

**A..Z, L/R/O/U** ..... Alle Befehle werden als ASCII-Zeichen übertragen.  
Beispiel: G= 71 (dez.) = \$47 leitet den Geraden-Befehl ein.

**x1, x2, y1, y2** ..... Koordinatenangaben werden mit 1 Byte übertragen.  
Beispiel: x1= 10 (dez.) = \$0A

**ESC** ..... 1 Byte: 27(dez.) = \$1B

**n1,n2,nr,aw,ew,wert,mst,ret, frm,daten** ..... Nummernwerte werden mit 1 Byte übertragen.  
Beispiel: n1=15(dez.) = \$0F

## PROGRAMMIERBEISPIEL

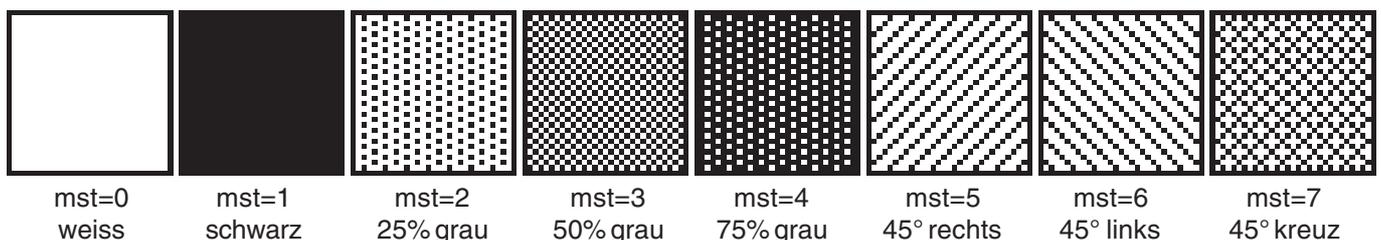
In der nachfolgenden Tabelle ist ein Beispiel zu sehen welches die Zeichenkette "Test" linksbündig an den Koordinaten 7,3 ausgibt.

Beispiel	Auszugebende Codes									
	ESC	Z	L	BEL	ETX	T	e	s	t	NUL
in ASCII										
in Hex	\$1B	\$5A	\$4C	\$07	\$03	\$54	\$65	\$73	\$74	\$00
in Dezimal	27	90	76	7	3	84	101	115	116	0
für Turbo-Pascal	write(aux, chr(27), 'Z', 'L', chr(7), chr(3), 'Test', chr(0));									
für 'C'	fprintf(stdaux, "\x1BZL%c%c%s\x00", 7, 3, "Test");									
für Q-Basic	OPEN "COM1:9600,N,8,1,BIN" FOR RANDOM AS #1 PRINT #1,CHR\$(27)+"ZL"+CHR\$(7)+CHR\$(3)+"Test"+CHR\$(0)									

## MUSTER

Bei diversen Befehlen kann als Parameter ein Mustertyp (mst = 0..7) eingestellt werden. So können rechteckige Bereiche, Bargraphs und sogar Texte mit unterschiedlichen Mustern verknüpft und dargestellt werden.

Folgende Füllmuster stehen dabei zur Verfügung:

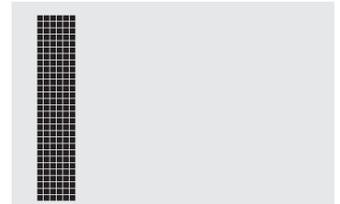


**BESCHREIBUNG DER EINZELNEN GRAFIKFUNKTIONEN**

Auf den nächsten Seiten befindet sich eine detaillierte alphabetisch sortierte Beschreibung zu jeder einzelnen Funktion. Als Beispiel wird jeweils ein vergrößerter Bildausschnitt von 50x32 Pixeln als Hardcopy gezeigt der den Displayinhalt nach Ausführung des Befehls darstellt. In den Beispielen sind die zu übertragenden Bytes als Hex-Werte abgebildet.

**ESC B L/R/O/U nr x1 y1 x2 y2 aw ew mst Bargraph definieren**

Es können bis zu 4 Bargraphs (**nr**=1..4) definiert werden, welche nach **L**=links, **R**=rechts, **O**=oben oder **U**=unten ausschlagen können. Der Bargraph beansprucht bei Vollausschlag einen Bereich mit den Koordinaten **x1,y1** bis **x2,y2**. Mit dem Anfangswert (kein Ausschlag) **aw** (=0..254) und dem Endwert (Vollausschlag) **ew** (=0..254) wird der Bargraph skaliert. Der Bargraph wird immer im Inversmodus mit dem Muster **mst** gezeichnet: Der Hintergrund bleibt somit in jedem Fall erhalten. (Achtung! Nach diesem Befehl ist der Bargraph nur definiert, am Display ist er aber noch nicht zu sehen).



Beispiel:

```
$1B $42 $4F $01 $04 $02 $09 $1E $04 $14 $01
```

Es wird der Bargraph Nr. 1 der nach oben ausschlägt definiert. Bei Vollausschlag nimmt er einen Bereich von den Koordinaten 4,2 bis 9,30 ein. Anfangs- und Endwert entspricht einer 4..20 mA Anzeige. (Das Bild zeigt den Bargraph im Vollausschlag wie er mit \$42 \$01 \$14 dargestellt wird)

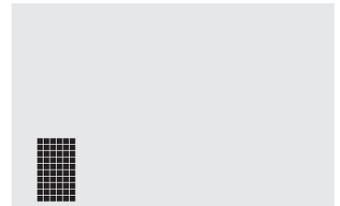
**ESC B nr wert**

Der Bargraph mit der Nummer **n1** (1..4) wird auf den neuen Wert eingestellt (**aw** <= **wert** <= **ew**). Ist **wert** > **ew** dann wird Endwert **ew** angezeigt. Der Bargraph muss vorher definiert worden sein (siehe oben).

Beispiel:

```
$1B $42 $01 $0A
```

Der im oberen Beispiel definierte Bargraph Nr. 1 wird auf den Wert 10 gestellt.

**Bargraph zeichnen****ESC D L**

Der gesamte Displayinhalt wird gelöscht (weiss)

Beispiel: \$1B \$44 \$4C

**Displayinhalt löschen****ESC D I**

Der gesamte Displayinhalt wird invertiert.

Beispiel: \$1B \$44 \$49

**Displayinhalt invertieren****ESC D S**

Der gesamte Displayinhalt wird gefüllt (schwarz)

Beispiel: \$1B \$44 \$53

**Displayinhalt füllen****ESC D A**

Der Displayinhalt wird ausgeschaltet (unsichtbar). Ausgaben sind auch im ausgeschalteten Zustand möglich.

Beispiel: \$1B \$44 \$41

Nach diesem Befehl ist der Displayinhalt nicht mehr sichtbar.

**Display Ausschalten****ESC D E**

Der Displayinhalt wird wieder eingeschaltet (sichtbar).

Beispiel: \$1B \$44 \$41

Nach diesem Befehl ist der Displayinhalt wieder sichtbar.

**Display Einschalten**

## ESC E n1 daten

Es ist möglich bis zu 16 Zeichen selbst zu definieren (je nach Fontgröße). Diese Zeichen haben dann die ASCII Codes 1 bis max.16 und bleiben bis zum Abschalten der Versorgungsspannung in einem 64 Byte großen unsichtbaren Bildschirm-RAM erhalten. Bei einem 4x6 Font können bis zu 16 Zeichen definiert werden, bei einem 8x16 Font bis zu 4 Zeichen. Achtung! Sollen mehrere Zeichen aus unterschiedlichen Fonts definiert werden, so ist darauf zu achten daß z.B. ein Zeichen mit Code 1 vom 8x16 Font denselben Platz im RAM benötigt wie die Zeichen mit den Codes 1 bis 4 vom 4x6 Font (siehe Tabelle nebenan)!

Beispiel 1:

\$1B \$45 \$01

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

definiert einen Pfeil nach oben für ASCII-Nr. 1 bei eingestelltem 6x8 Zeichensatz.

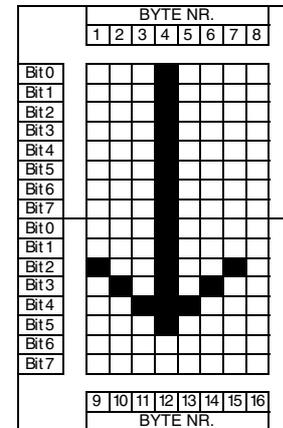
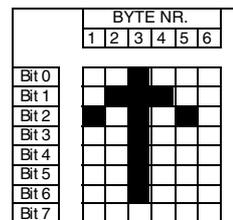
Beispiel 2:

\$1B \$45 \$02

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

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

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



Selbstdefinierbare Zeichen (Code)	
4x6	8x16
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12
13	13
14	14
15	15
16	16

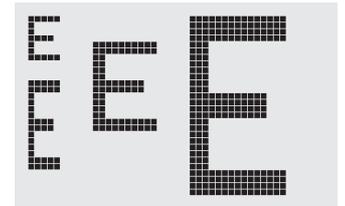
## ESC F n1 n2 n3

Es wird der Font mit der Nr. **n1** eingestellt. Ausserdem wird ein Vergrößerungsfaktor (1..4-fach) für die Breite **n2** und für die Höhe **n3** getrennt eingestellt.

Beispiel: \$1B \$46 \$02 \$03 \$04

ab sofort ist der 6x8- Font mit 3-facher Breite und 4-facher Höhe eingestellt. Im Bild nebenan ist das Zeichen 'E' aus dem 6x8 Font mit unterschiedlichen Vergrößerungen dargestellt.

## Font einstellen



## ESC F T n1

## Terminal-Font einstellen

Es wird der Font mit der Nr. **n1** für das Terminal eingestellt. Der Font für das Terminal wird immer ohne Zoom und im REPLACE Modus benutzt.

Beispiel: \$1B \$46 \$54 \$03

ab sofort ist der 6x8 Font als Terminalfont eingestellt.

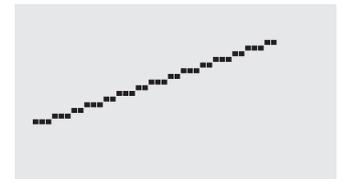
## ESC G x1 y1 x2 y2

## Gerade zeichnen

Eine Gerade wird von den Koordinaten **x1,y1** nach **x2,y2** unter Beachtung des eingestellten Grafikmodus 'V' (setzen / löschen / invers) gezeichnet.

Beispiel: \$1B \$47 \$03 \$14 \$28 \$06

Es wird eine Gerade von 3,20 nach 50,6 gezeichnet.



## ESC H x1 y1 x2 y2

## Hardcopy vom Displayinhalt erstellen

Der Bereich von der linken oberen Ecke **x1,y1** bis zu rechten unteren Ecke **x2,y2** wird angefordert. Das Kit sendet daraufhin sofort die Breite und Höhe des Bildausschnittes und danach die Bilddaten. Zum Aufbau der Bilddaten siehe den Befehl Bild Upload **ESC U L**'.

Beispiel: \$1B \$48 \$00 \$00 \$1F \$0F

und sofort wird der linke obere Teil des Bildschirms mit der Größe 32 x 16 Pixel über RS-232 gesendet.

## ESC J n1

## Summer manuell Ein-/Ausschalten

Der Summer wird **n1=0** ausgeschaltet, **n1=1** dauerhaft eingeschaltet oder mit **n1=2..255** für n1/10 Sekunden lang eingeschaltet (nur bei den Versione mit Touchpanel EA KIT120-5LEDTP).

Beispiel: \$1B \$4A \$0A

nach diesem Befehl ertönt der Summer 1s lang.

## ELECTRONIC ASSEMBLY

### ESC K A adr

### Adresse zuweisen

Dem KIT wird die Adresse **adr** (0..254) zugewiesen. Dieser Befehl befindet sich am besten im Power-On Makro.

Beispiel: \$1B \$4B \$41 \$01

Das KIT kann ab sofort unter der Adresse \$01 angesprochen werden.

### ESC K S/Dadr

### KIT240 (de)selektieren

Das KIT mit der Adresse **adr** (0..254) wird **S**=selektiert oder **D**=deselektiert; Die Adresse 255=\$FF ist eine Masteradresse mit der alle KITs angesprochen werden.

Beispiel: \$1B \$4B \$44 \$01

alle Befehle werden für das KIT mit der Adresse \$01 ab sofort ignoriert.

### ESC L n1 mst

### Text-Modus einstellen

Der Verknüpfungsmodus **n1** und das Muster **mst** wird für die Textfunktion Zeichenkette ausgegeben **ESC Z** eingestellt.

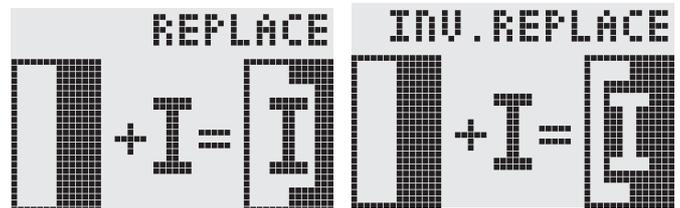
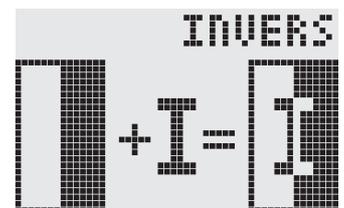
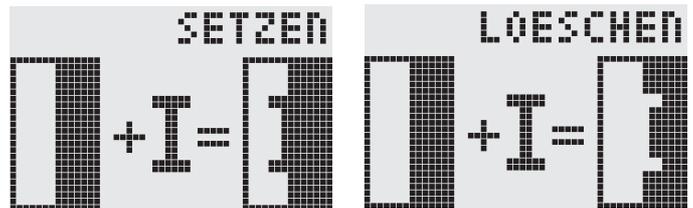
Beispiel:

\$1B \$4C \$03 \$03

stellt den Verknüpfungsmodus für alle folgenden Textfunktionen auf graue Zeichen (Muster 3 = 50%Grau) invertiert mit dem Hintergrund.

Verknüpfungsmodus n1:

- 1 = setzen: schwarze Pixel ohne Rücksicht auf den vorigen Wert (ODER)
- 2 = löschen: weißes Pixel ohne Rücksicht auf den vorigen Wert
- 3 = invers: aus schwarzen Pixeln werden Weiße und umgekehrt (EXOR)
- 4 = replace: Hintergrund löschen und schwarze Pixel setzen
- 5 = invers replace: Hintergrund füllen und weiße Pixel setzen



### ESC M N n1

### Makro aufrufen

Das (Normal) Makro mit der Nummern **n1** (0..255) wird aufgerufen.

Beispiel: \$1B \$4D \$4E \$0F ; Nach diesem Befehl wird das Makro Nummer 15 gestartet.

### ESC M T n1

### Touchmakro aufrufen

Das Touch-Makro mit der Nummern **n1** (0..255) wird aufgerufen.

Beispiel: \$1B \$4D \$4E \$0F ; Nach diesem Befehl wird das Touch-Makro Nummer 15 gestartet.

### ESC M P n1

### Portmakro aufrufen

Das Port-Makro mit der Nummer **n1** (0..255) wird aufgerufen.

Beispiel: \$1B \$4D \$4E \$0F ; Nach diesem Befehl wird das Port-Makro Nummer 15 gestartet.

### ESC M A/J n1 n2 n3

### Makros automatisch ausführen

Die Normal-Makros mit den Nummern **n1** bis **n2** werden automatisch alle **n3/10** Sekunden aufgerufen.

**A**=zyklischer Aufruf (z.B. 1,2,3,4,1,2,3,4 usw.); **J**=Pingpong Aufruf (z.B. 1,2,3,4,3,2,1,2,3,4 usw.).

Die automatische Ausführung wird beendet:

- wenn ein Zeichen von der RS-232 Schnittstelle empfangen wird.
- eine Touchberührung automatisch ein Touchmakro ausführt.
- oder eine Eingangsänderung ein Portmakro ausführt

Beispiel: \$1B \$4D \$41 \$01 \$03 \$05

Die Makros mit den Nummern 1, 2 und 3 werden automatisch mit einer Pause vom 1/2 Sekunde ausgeführt.

### ESC O n1 n2

### Cursor positionieren

Der Cursor wird für den Terminal-Betrieb auf Spalte **n1** und Zeile **n2** gesetzt. Der Ursprung links oben ist (1,1).

Beispiel:

```
$1B $4F $03 $05
```

setzt den Cursor auf die 3. Spalte in Zeile 5.

### ESC P x1 y1

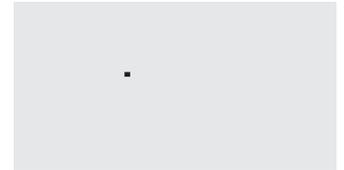
### Punkt setzen

Ein Pixel wird an der Koordinate **x1,y1** unter Beachtung des eingestellten Grafikmodus 'ESC V' (setzen / löschen / invertieren) gesetzt.

Beispiel:

```
$50 $11 $0D
```

setzt den Pixel an der Koordinate 17,13.



### ESC Q C n1

### Cursor EIN/AUS

**n1=1**: der Cursor wird eingeschaltet, er blinkt an der aktuellen Zeichenposition im Terminal.

**n1=0**: der Cursor wird ausgeschaltet.

Beispiel:

```
$1B $51 $43 $01
```

Der Cursor wird eingeschaltet.

### ESC Q D x1 y1 x2 y2

### Blinkbereich definieren

Der Bereich von der linken oberen Ecke **x1,y1** bis zur rechten unteren Ecke **x2,y2** wird als automatisch invertierender Bereich festgelegt und die Blinkfunktion wird gestartet. Der Terminal-Cursor wird dadurch deaktiviert.

Beispiel:

```
$1B $51 $44 $00 $0F $07 $10
```

Definiert den Blinkbereich von 0,15 nach 7,16.

### ESC Q Z n1

### Blinkzeit einstellen

Stellt die Blinkzeit auf **n1** (=1..15) zehntel Sekunden ein. Bei **n1= 0** wird die Blinkfunktion deaktiviert und der Original Bildschirm wieder hergestellt.

Beispiel:

```
$1B $51 $5A $03
```

stellt die Blinkzeit auf 0,3 Sekunden ein.

### ESC R R x1 y1 x2 y2

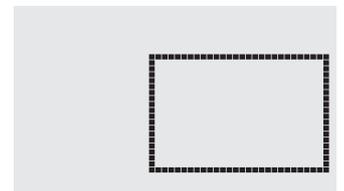
### Rechteck zeichnen

Ein Rechteck wird von der linken oberen Ecke **x1,y1** bis zur rechten unteren Ecke **x2,y2** unter Beachtung des eingestellten Grafikmodus 'V' (setzen / löschen / invers) gezeichnet. Der Inhalt des Rechtecks wird dabei nicht verändert. Vergleiche 'ESC R O' Box zeichnen.

Beispiel:

```
$1B $52 $52 $15 $08 $30 $25
```

zeichnet ein Rechteck von 21,8 nach 48,37.



### ESC R N x1 y1 x2 y2

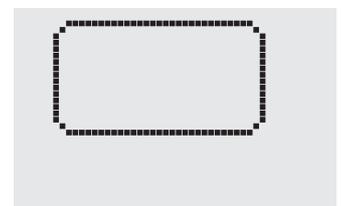
### Rundeck zeichnen

Ein Rechteck mit abgerundeten Ecken wird von der linken oberen Ecke **x1,y1** bis zur rechten unteren Ecke **x2,y2** unter Beachtung des eingestellten Grafikmodus 'V' (setzen / löschen / invers) gezeichnet. Der Inhalt des Rundecks wird nicht verändert. Vergleiche 'ESC R J' Rundbox zeichnen.

Beispiel:

```
$1B $52 $4E $06 $02 $26 $13
```

zeichnet ein Rundeck von 6,2 nach 38,19.



## ELECTRONIC ASSEMBLY

**ESC R L x1 y1 x2 y2**

Der Bereich von der linken oberen Ecke **x1,y1** bis zur rechten unteren Ecke **x2,y2** wird gelöscht.

Beispiel:

\$1B \$44 \$53 \$1B \$52 \$4C \$06 \$04 \$28 \$19

das Display wird mit **ESC D S** gefüllt und dann von 6,4 nach 40,25 gelöscht .

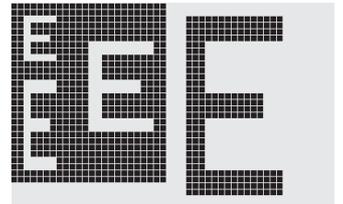
**Bereich löschen****ESC R I x1 y1 x2 y2**

Der Bereich von der linken oberen Ecke **x1,y1** bis zur rechten unteren Ecke **x2,y2** wird invertiert (aus schwarzen Pixeln werden Weiße und umgekehrt).

Beispiel:

\$1B \$52 \$49 \$00 \$00 \$17 \$1B

invertiert bei vorhandenem Displayinhalt aus dem Beispiel "Font einstellen" den Bereich von 0,0 nach 23,27.

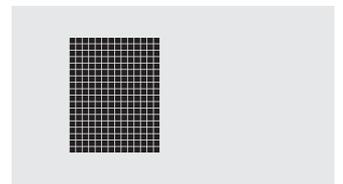
**Bereich invertieren****ESC R S x1 y1 x2 y2**

Der Bereich von der linken oberen Ecke **x1,y1** bis zur rechten unteren Ecke **x2,y2** wird gefüllt (auf schwarze Pixel gesetzt).

Beispiel:

\$1B \$52 \$53 \$09 \$05 \$16 \$16

setzt den Bereich von 9,5 nach 22,22 auf schwarz.

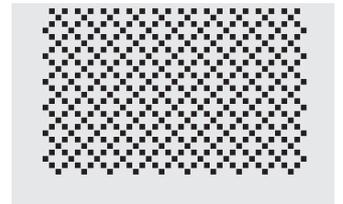
**Bereich füllen****ESC R M x1 y1 x2 y2 mst**

Ein rechteckiger Bereich wird von der linken oberen Ecke **x1,y1** bis zur rechten unteren Ecke **x2,y2** mit dem Muster **mst** unter Beachtung des eingestellten Grafikmodus 'ESC V' (setzen/löschen/invertieren/replace/invers replace) gezeichnet.

Beispiel:

\$1B \$52 \$4D \$05 \$01 \$2D \$1A \$07

zeichnet das Muster 7=45°Kreuz von 5,1 nach 45,26.

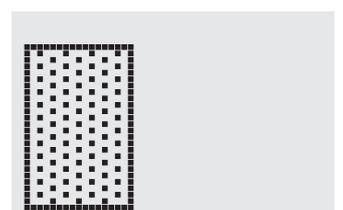
**Bereich mit Füllmuster****ESC R O x1 y1 x2 y2 mst**

Ein Rechteck wird von der linken oberen Ecke **x1,y1** bis zur rechten unteren Ecke **x2,y2** mit dem Muster **mst** gezeichnet. Der Hintergrund der Box wird dabei gelöscht. Vergleiche 'ESC R R' Rechteck zeichnen.

Beispiel:

\$1B \$52 \$4F \$02 \$05 \$12 \$1E \$02

zeichnet eine Box von 2,5 nach 18,30 mit dem Muster 2=25%Grau.

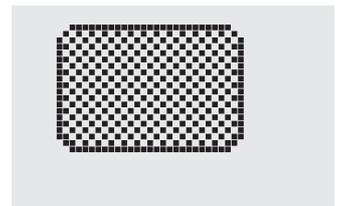
**Box zeichnen****ESC R J x1 y1 x2 y2 mst**

Ein Rechteck mit abgerundeten Ecken wird von der linken oberen Ecke **x1,y1** bis zur rechten unteren Ecke **x2,y2** mit dem Muster **mst** gezeichnet. Der Hintergrund wird dabei gelöscht. Vergleiche 'ESC R N' Rundeck zeichnen.

Beispiel:

\$1B \$52 \$4A \$07 \$03 \$23 \$16 \$03

zeichnet eine Rundbox von 7,3 nach 35,22 mit dem Muster 3=50%Grau.

**Rundbox zeichnen****ESC S anz daten...**

Die nachfolgenden **anz** (1..255, 0=256) Bytes werden auf der seriellen Schnittstelle ausgegeben.

Beispiel:

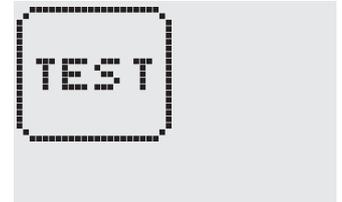
\$1B \$53 \$04 \$54 \$45 \$53 \$54

Das Wort 'TEST' wird über die RS-232C Schnittstelle gesendet.

**Bytes über RS-232 senden**

### ESC T H/Vf1 f2 ret frm text... NUL Touch-Taste definieren

Ein Touch-Taste wird definiert und mit dem aktuellen Font beschriftet. **H**=horizontale oder **V**=vertikale Berschriftung (90° gedreht). Mehrere Touch-Felder können als eine einzige Touch-Taste zusammengefasst werden **f1**=linkes obere Touchfeld, **f2**=rechtes untere Touchfeld der neuen Touchtaste. Diese Touchtaste wird mit **ret** ein Returncode zugewiesen (1..255). Beim Berühren der Touchtaste wird dann das Touchmakro mit der Nummer **ret** aufgerufen oder, falls kein Touchmakro definiert ist, dieser Returncode über die RS232 gesendet. Mit **frm** wird die Darstellung der Touchtaste festgelegt (frm=0: nichts zeichnen; frm=1: Touchtaste löschen; frm=2: Touchtaste löschen und mit Rahmen zeichnen). **text...**=Zeichenkette mit der Beschriftung (wird immer in der Touchtaste zentriert). Die Beschriftung kann auch mehrzeilig sein, die einzelnen Zeilen werden durch das Zeichen '|' (=7C) getrennt. Die Zeichenkette muss mit **NUL**= \$00 beendet werden. Siehe Beispiel auf Seite 3.



#### Beispiel 1 Horizontale Touchtaste:

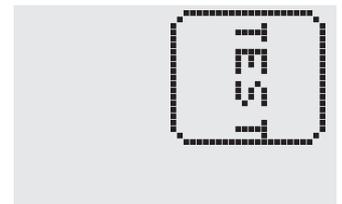
\$1B \$54 \$48 \$01 \$01 \$41 \$02 \$54 \$45 \$53 \$54 \$00

definiert eine horizontale Touchtaste (nur Feld Nr. 1) mit dem Returncode 65='A'. Die Touchtaste wird mit Rahmen gezeichnet und mit dem Wort 'TEST' beschriftet.

#### Beispiel 2 Vertikale Touchtaste:

\$1B \$54 \$56 \$02 \$02 \$42 \$02 \$54 \$45 \$53 \$54 \$00

definiert eine vertikale Touchtaste (nur Touchfeld Nr. 2) mit dem Returncode 66='B'. Die Touchtaste wird mit Rahmen gezeichnet und mit dem Wort 'TEST' beschriftet.



### ESC T P/R Touchfelder Vorbelegen/Reset

Alle 10 Touchfelder werden mit **P**=aufsteigendem Returncode belegt (1..10) oder **R**=rückgesetzt alle Touchfelder erhalten den Returncode 0 d.h sie sind deaktiviert.

Beispiel: \$1B \$54 \$52

Alle Touchfelder sind nach diesem Befehl deaktiviert und werden nicht mehr erkannt.

### ESC T I/S n1 Touchtasten Reaktion

Mit diesem Befehlen wird die automatische Reaktion des Touchpanels beim Berühren eingestellt. Es können beide Reaktionen gleichzeitig aktiviert werden.

**I**=automatisches Invertieren beim Berühren der Touchtaste **n1**=0: AUS oder **n1**=1: EIN.

**S**=automatischer Signalton beim Berühren **n1**=0: AUS oder **n1**=1: EIN

Beispiel: \$1B \$54 \$49 \$01

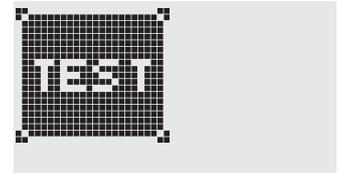
Nach diesem Befehl ertönt der Summer beim Berühren einer Touchtaste.

### ESC T M ret Touchtaste manuell invertieren

Die Touchtaste mit dem Returncode **ret** kann mit diesem Befehl manuell invertiert werden.

Beispiel: \$1B \$54 \$4D \$41

Die Touchtaste aus obigen Beispiel mit dem Returncode 65='A' wird invertiert.



### ESC T A n1 (Touch)Tastenabfrage Ein/Aus

Die (Touch)Tastenabfrage wird mit diesem Befehl eingestellt:

**n1**=0: Tastenabfrage ist komplett abgeschaltet: keine Touchmakros, keine manuelle Tastenabfrage möglich.

**n1**=1: Tastenabfrage ist aktiv: Tastendrucke lösen Touchmakros aus oder werden über RS232 gesendet.

**n1**=2: Tastenabfrage ist aktiv: Tastendrucke lösen Touchmakros aus, müssen manuell abgefragt werden.

Beispiel: \$1B \$54 \$41 \$02

Die (Touch)Tastenfrage wird aktiviert, die Tastendrucke werden nicht automatisch über RS232 gesendet, sie müssen manuell mit dem Befehl **ESC T W** angefordert werden.

### ESC T W Touchtaste manuell abfragen

Der Returncode der momentan gedrückten Touchtaste wird auf der RS232 gesendet.

Beispiel: \$1B \$54 \$57

## ELECTRONIC ASSEMBLY

### ESC U E x1 y1 n1

### Bild aus EEPROM

Das gespeicherte Bild im EEPROM mit der Numer **n1** (0..255) wird an die Koordinate **x1,y1** geladen.

Beispiel:

```
$1B $55 $45 $02 $03 $0E
```

Das Bild Nummer 14 aus dem EEPROM wird an Koodinate 2,3 angezeigt.

### ESC U L x1 y1 daten...

### Bild Upload

Ein Bild wird an die Koordinate **x1,y1** geladen.

daten..:

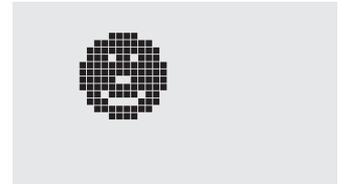
- 1 Byte für die Bildbreite in Pixeln
- 1 Byte für die Bildhöhe in Pixeln
- Bilddaten: Anzahl = ((Höhe+7) / 8) \* Breite Bytes.

1 Byte steht für 8 senkrechte Pixel am Bildschirm; 0=weiß, 1=schwarz;  
LSB: oben, MSB: unten; Das Bild ist von links nach rechts abgelegt.  
Das Programm BMP2BLV.EXE auf der als Zubehör erhältlichen Diskette EA DISK240 erzeugt aus monochromen Windows-Bitmap- Grafiken (\*.BMP) die Bilddaten inkl. der Angabe von Breite und Höhe.

Beispiel:

```
$1B $55 $4C $09 $04 $0C $0C
$F0 $FC $FE $FE $F7 $BF $BF $F7 $FE $FE $FC $F0
$00 $03 $07 $06 $0D $0D $0D $0D $06 $07 $03 $00
```

lädt das nebenstehende Bild an die Koordinate 9,4.



	BYTE NR											
	1	2	3	4	5	6	7	8	9	10	11	12
Bit 0												
Bit 1												
Bit 2												
Bit 3												
Bit 4												
Bit 5												
Bit 6												
Bit 7												
Bit 0												
Bit 1												
Bit 2												
Bit 3												
Bit 4												
Bit 5												
Bit 6												
Bit 7												
	13	14	15	16	17	18	19	20	21	22	23	24
	BYTE NR											

### ESC V n1

### Grafik-Modus einstellen

Einstellen des Verknüpfungsmodus **n1** für folgende Grafikfunktionen: Punkt setzen ESC P, Gerade zeichnen ESC G, Gerade weiter zeichnen ESC W, Rechteck zeichnen ESC R R, Rundeck zeichnen ESC R N, Bereich mit Füllmuster ESC R M.

Beispiel:

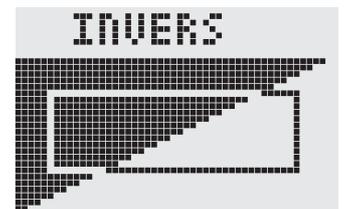
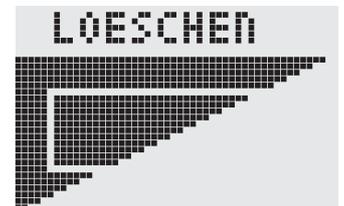
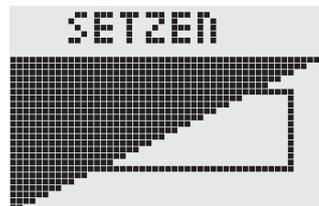
```
$1B $56 $03
```

stellt den Verknüpfungsmodus auf invers.

Als Beispiel wird nebenan ein Rechteck mit den Verknüpfungsmodi setzen, löschen und invers auf einen vorhandenem Hintergrund gezeichnet.

Verknüpfungsmodus n1:

- 1=setzen: schwarze Pixel ohne Rücksicht auf den vorigen Wert (ODER)
- 2=löschen: weißes Pixel ohne Rücksicht auf den vorigen Wert
- 3=invers: aus schwarzen Pixeln werden Weiße und umgekehrt (EXOR)
- 4=replace: Hintergrund löschen und Pixel setzen; nur Bereich mit Füllmuster 'mst'
- 5=invers replace: Hintergrund füllen, Pixel löschen; nur Bereich mit Füllmuster 'mst'



### ESC W x1 y1

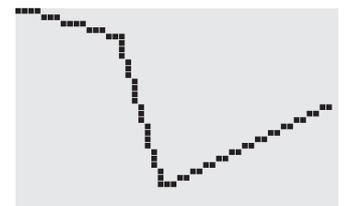
### Gerade weiterzeichnen

Zieht eine Gerade vom zuletzt gezeichneten Geradenende bzw. Punkt bis nach **x1,y1** unter Beachtung des eingestellten Grafik-Modus 'V'

Beispiel:

```
$1B $47 $00 $00 $10 $04
$1B $57 $16 $1B
$1B $57 $30 $0F
```

Zuerst wird eine Gerade von 0,0 nach 16,4 gezeichnet. Dann weiter nach 22,27 und nach 48,15.



### ESC X n1

### Warten / Pause

Mit diesem Befehl wird das KIT für **n1**/10 Sekunden angehalten.

Beispiel:

\$1B \$58 \$0A

Nach diesem Befehl wartet das KIT eine Sekunde bevor der nächste Befehl abgearbeitet wird.

### ESC Y R n1

### Eingabe-Port lesen

Liest den Eingangs-Port (**n1**=1..5 = IN1..IN5) ein. Wenn **n1**=0, werden alle Eingänge als 5-Bit Binärwert eingelesen (MSB:IN5...In1:LSB); Siehe Applikation auf Seite 5. Achtung: Die Optokoppler invertieren die Eingangslogik (Eingang offen: 1). Der Befehl "ESC Y I 1" stellt dies richtig (Eingang offen: 0).

Beispiel:

\$1B \$59 \$52 \$03

liest den Port IN3 ein. Ergebnis wird über RS232 gesendet.

### ESC Y W n1 n2

### Ausgabe-Port schreiben

Ändert den Ausgabe Port (**n1**=1..5 = OUT1..OUT5) auf den Wert **n2** (0=L-Pegel; 1=H-Pegel; 2=Port invertieren). Wenn **n1**=0, werden alle Ausgänge als Binärwert **n2** (MSB:OUT5...OUT1:LSB) ausgegeben; Siehe Applikation auf Seite 5.

Beispiel:

\$1B \$59 \$57 \$02 \$01

schaltet den Ausgabe Port OUT2 auf H-Pegel.

### ESC Y A n1

### automatische Portabfrage EIN/AUS

Jede Änderung am Eingabeport (5-Bit Binärwert IN5..IN1) kann ein Portmakro (0..31) aufrufen. Mit diesem Befehl wird die automatische Portabfrage **n1**=1 aktiviert oder mit **n1**=0 deaktiviert. Nach dem Einschalten wird der aktuelle Portzustand gelesen und sofort das dazugehörige Portmakro ausgeführt.

Beispiel:

\$1B \$59 \$41 \$01

Die automatische Portabfrage wird aktiviert und das anliegende Portmakro wird ausgeführt.

### ESC Y I n1

### Eingabe-Port invers

Mit diesem Befehl kann die Logik des Eingabe-Ports umgekehrt werden (**n1**=0 normal oder mit **n1**=1 invers). Sinnvoll z.B. bei den Optokoppler Eingängen.

Beispiel:

\$1B \$59 \$49 \$01

Die Logik Eingabe-Ports wird invertiert.

### ESC Y L n1

### LED-Beleuchtung Ein-/Ausschalten

Die LED Hintergrundbeleuchtung wird **n1**=0 ausgeschaltet, **n1**=1 dauerhaft eingeschaltet; **n1**=2 umgekehrt: EIN->AUS bzw. AUS->EIN oder mit **n1**=3..255 für **n1**/10 Sekunden lang eingeschaltet .

Beispiel:

\$1B \$4A \$64

nach diesem Befehl leuchtet die LED-Hintergrundbeleuchtung 10s lang.

## ELECTRONIC ASSEMBLY

**ESC Z L/Z/R x1 y1 text... NUL****Zeichenkette horizontal**

Schreibt die Zeichenkette **text...**, **L**=Linksbündig, **Z**=Zentriert oder **R**=Rechtsbündig an der Koordinate **x1** unter Beachtung des eingestellten Textmodus **ESC L**. Es können auch mehrzeilige Texte ausgegeben werden, die einzelnen Zeilen sind durch das Zeichen '|' (=\$7C) getrennt. Die Zeichenkette muss mit **NUL**= \$00 beendet werden. Die Koordinate **y1** ist die Oberkante der 1. Zeile.

Beispiel 1:

```
$1B $5A $4C $00 $00 $4C $65 $66 $74 $7C $4F $6B $00
```

schreibt an 0,0 linksbündig den Text "Left|Ok"

Beispiel 2:

```
$1B $5A $5A $19 $00 $43 $65 $6E $74 $65 $72 $7C $4F $6B $00
```

schreibt an 25,0 zentriert "Center|Ok"

Beispiel 3:

```
$1B $5A $52 $31 $00 $52 $69 $67 $68 $74 $7C $4F $6B $00
```

schreibt an 49,0 rechtsbündig "Right|Ok"




**ESC Z O/M/U x1 y1 text... NUL****Zeichenkette vertikal**

Schreibt die Zeichenkette **text...** um 90° gedreht, **O**=Obenbündig, **M**=Mittig oder **U**=Untenbündig an der Koordinate **y1** unter Beachtung des eingestellten Textmodus **ESC L**. Es können auch mehrzeilige Texte ausgegeben werden, die einzelnen Zeilen sind durch das Zeichen '|' (=\$7C) getrennt. Die Zeichenkette muss mit **NUL**= \$00 beendet werden. Die Koordinate **x1** ist die rechte Kante der 1. Zeile.

Beispiel 1:

```
$1B $5A $4F $31 $00 $54 $6F $70 $7C $4F $6B $00
```

schreibt an 49,0 obenbündig "Top|Ok"

Beispiel 2:

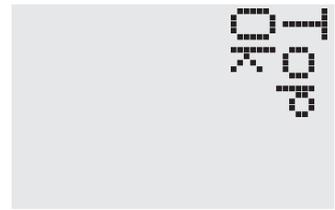
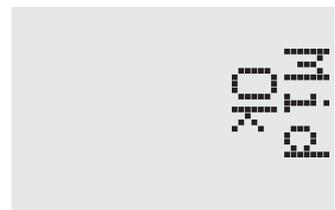
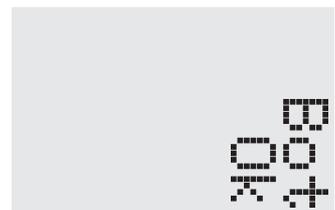
```
$1B $5A $4D $31 $0F $4D $69 $64 $7C $4F $6B $00
```

schreibt an 49,15 mittig "Mid|Ok"

Beispiel 3:

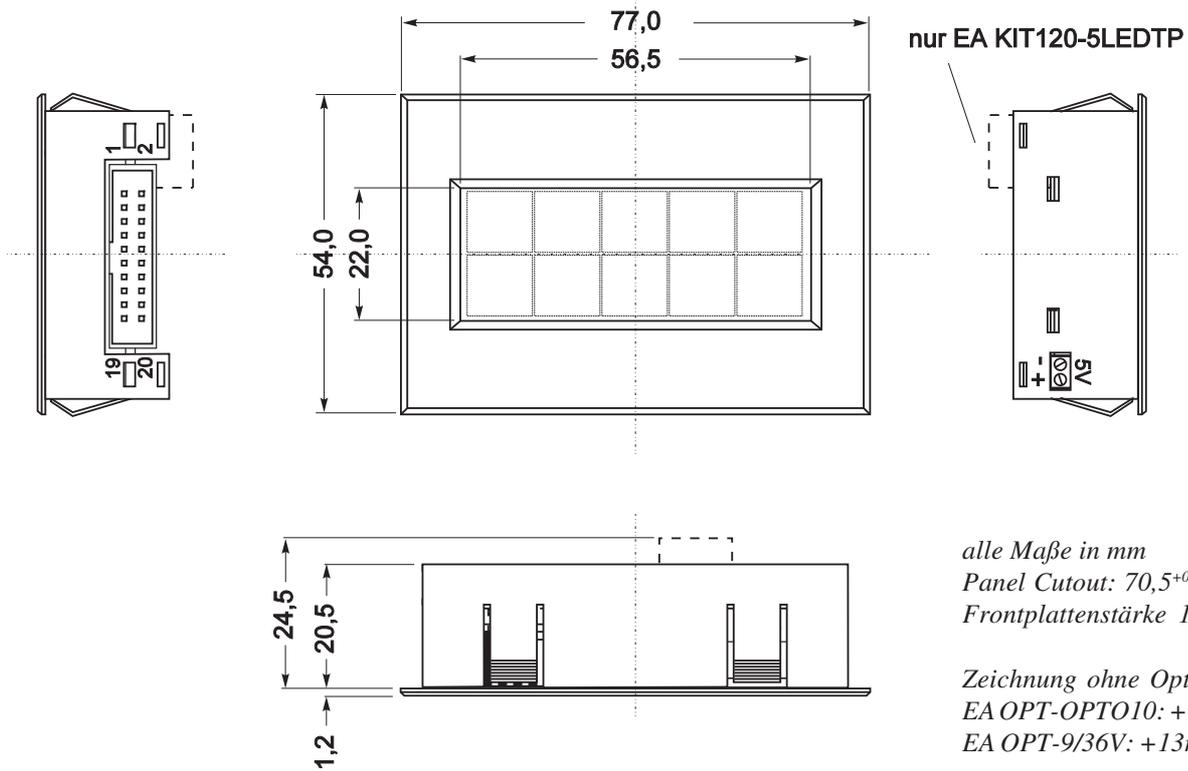
```
$1B $5A $55 $31 $1F $42 $6F $74 $7C $4F $6B $00
```

schreibt an 49,31 untenbündig "Bot|Ok"

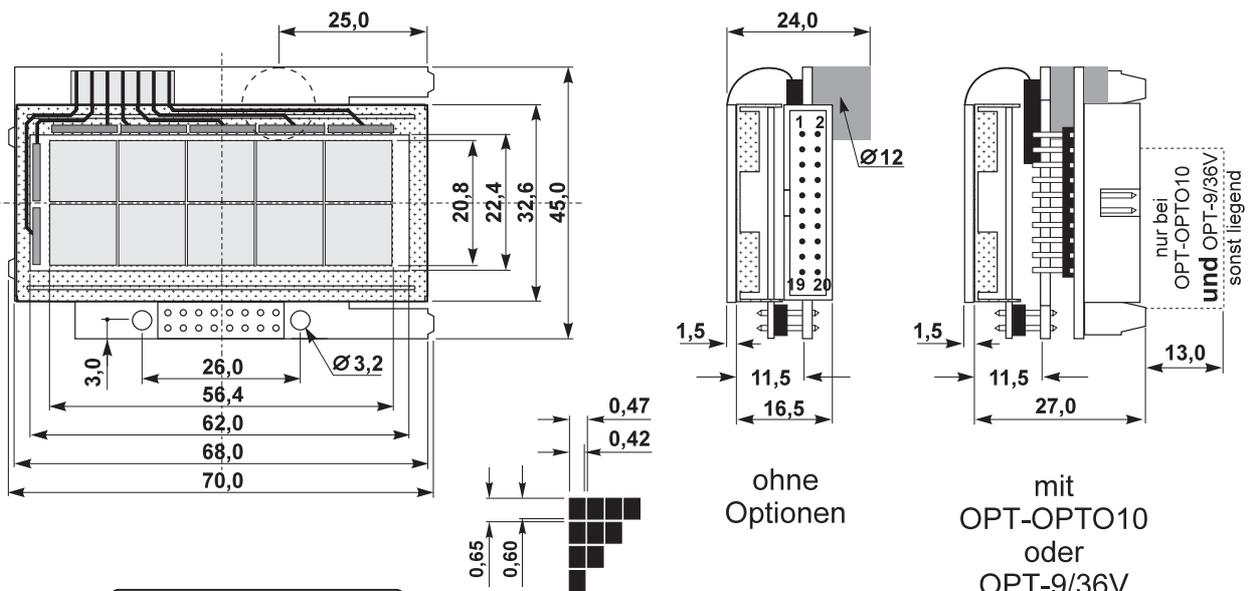
# EA KIT120-5

## ABMESSUNGEN



## ABMESSUNGEN OHNE GEHÄUSE

In manchen Applikationen mag es sinnvoll sein, die gesamte Elektronik aus dem Gehäuse zu nehmen. Achten Sie in diesem Fall bitte besonders auf ESD-gerechtes Handling!



alle Maße in mm