

# LOW COST GRAFIKEINHEIT 120x32 MIT 3 FONTS, ZOOM UND LED-BACKLIGHT

Text+Grafik mischen

Füllmuster

Font Zoom

Bild Download



EA GE120B-5NV24  
Abmessung: 68x39x18 mm



EA GE120-5NV24  
Abmessung: 68x39x16 mm

## TECHNISCHE DATEN

- \* 120x32 PIXEL POSITIV MIT GELB/GRÜNER LED-BELEUCHTUNG
- \* AUCH BLAU NEGATIV MIT WEISSER LED-BELEUCHTUNG
- \* 3 VERSCHIEDENE FONTS INTEGRIERT
- \* ZOOM FUNKTION ALLER FONTS (2-, 3- UND 4-FACH)
- \* PROGRAMMIERUNG ÜBER DIVERSE EINGebaUTE GRAFIKFUNKTIONEN:
- \* GERADE, PUNKT, BEREICH, UND/ODER/EXOR, FÜLLMUSTER...
- \* TEXT UND GRAFIK MISCHEN
- \* 8 FREI DEFINIERBARE ZEICHEN
- \* ANSTEUERUNG ÜBER RS-232 / CMOS-PEGEL
- \* BAUDRATE PROGRAMMIERBAR VON 300 BIS 9.600 BAUD
- \* KEINE TIMINGPROBLEME BEI SCHNELLEM BUSSYSTEM
- \* 8 DIGITALE AUSGÄNGE ZUR FREIEN VERWENDUNG
- \* +5V / 100mA GELB/GRÜN BZW. 50mA BLAU-WEISS

## ZUBEHÖR

- \* DISKETTE FÜR PC MIT KONVERTIERSOFTWARE FÜR WINDOWS-BMP GRAFIKEN: **EA DISK9719**

## BESTELLBEZEICHNUNG

GRAFIKEINHEIT 120x32, POSITIV GELB/GRÜN, RS-232      **EA GE120-5NV24**  
 GRAFIKEINHEIT 120x32, NEGATIV BLAU-WEISS, RS-232      **EA GE120B-5NV24**

### ALLGEMEINES

Das Grafik-LCD EA GE120-5NV24 ist für kleine bis mittlere Stückzahlen konzipiert. Aufgrund seiner kleinen Außenabmessungen, dem sehr guten Supertwistkontrast und der einfachen Programmierung ist es innerhalb weniger Stunden möglich, an nahezu jedes Prozessorsystem ein informativen und optisch ansprechenden Bildschirm anzuschließen. Die Ansteuerung erfolgt über die Standard Schnittstelle RS-232. Das Display enthält komplette Grafikroutinen zur Displayausgabe sowie verschiedenste Schriftgrößen.

Die Programmierung erfolgt über hochsprachenähnliche Grafikbefehle; die zeitraubende Programmierung von Zeichensätzen und Grafikroutinen entfällt hier völlig. Doch nicht nur der Entwicklungsaufwand reduziert sich drastisch. Auch in der Serie sind die folgende Vorteile spürbar:

- keine Timingprobleme bei schnellem Prozessorbus
- keine Speicherplatzprobleme (Arbeitsspeicher und Speicher für den Zeichensatz v.a. bei  $\mu C$ )
- keine zeitaufwendigen Grafikberechnungen welche die Prozessorgeschwindigkeit belasten.

Es sind keine Treiber, Dekoder oder Portbausteine erforderlich. Im einfachsten Fall erfolgt die Displayansteuerung über nur 1 Leitung Rx/D.

### HARDWARE

Das Display ist für +5V Betriebsspannung ausgelegt. Die Datenübertragung erfolgt seriell asynchron im RS-232 Format mit echten V.24 Pegeln oder über 5V CMOS Pegel. Das Übertragungsformat ist fest auf 8 Datenbits, 1 Stopbit, no Parity eingestellt. Die Baudrate kann über 2 Lötbrücken von 300 Baud bis zu 9.600 Baud ausgewählt werden. Die Handshakeleitung RTS steht zur Verfügung. Bei kleinen Datenmengen ist eine Auswertung nicht erforderlich.

Datenformat:



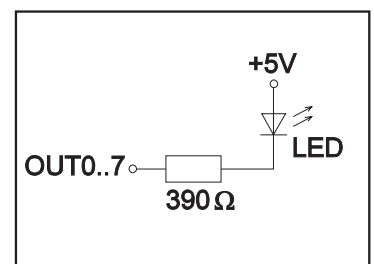
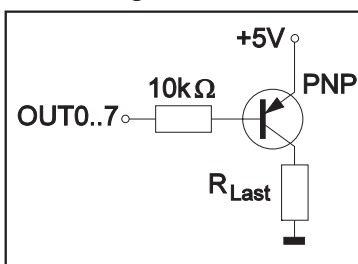
### SOFTWARE

Die Programmierung der Grafikeinheit erfolgt über Befehle wie z.B. Zeichne ein Rechteck von (0,0) nach (20,15). Der Ursprung liegt im linken oberen Eck des Displays. Über die serielle Schnittstelle müssen somit folgende Bytes gesendet werden: \$52 \$00 \$00 \$14 \$0F. Texte lassen sich ebenso pixelgenau plazieren. Das Mischen von Text und Grafik ist jederzeit möglich. Es können 2 verschiedene Zeichensätze verwendet werden. Jeder Zeichensatz kann wiederum 2-, 3- und 4-fach gezoomt werden.

### DIGITALE AUSGÄNGE OUT0..7 (ANSCHLUSS J2)

Mit dem Befehl "Y n1 n2" kann jeder Pin OUT0..7 auf H- oder L-Pegel geschaltet werden; er ist mit einem Open-Drain Ausgang vergleichbar, welcher keinen internen Pull-up Widerstand besitzt. Strom kann also nur bei L-Pegel fließen. Jeder Pin kann max. 10mA liefern, alle Pins zusammen dürfen mit nicht mehr als 26mA belastet werden (z.B. 2x10mA und 1x6mA).

Es ist somit möglich mit einem Ausgang direkt eine LED zu schalten. Größere Ströme können durch Verwendung eines externen Transistors geschaltet werden. Nach dem Power-On bzw. Power-Save-Mode liegen alle Ausgänge auf H-Pegel.



## ELECTRONIC ASSEMBLY

### INTEGRIERTE FONTS

Im der Grafikeinheit EA GE120-5NV24 sind bereits 3 Zeichensätze integriert (Font1: 4x6 Pixel; Font2: 6x8 Pixel und Font3: 8x16 Pixel). Jeder Zeichensatz kann in 1-, 2-, 3- oder 4-facher Höhe verwendet werden. Unabhängig davon läßt sich auch die Breite verdoppeln, verdreifachen oder vervierfachen. Zusätzlich können 8 eigene Zeichen definiert werden, die solange erhalten bleiben, bis die Versorgungsspannung abgeschaltet wird. (Siehe Befehl 'E').

Jedes Zeichen kann pixelgenau plaziert werden. Texte und Grafiken können beliebig gemischt dargestellt werden. Auch mehrere verschiedene Schriftgrößen lassen sich gemeinsam darstellen.

+ 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)	0	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
\$50 (dez: 80)	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_

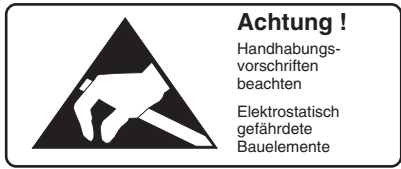
Font 1

+ 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)	0	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
\$50 (dez: 80)	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
\$60 (dez: 96)	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
\$70 (dez: 112)	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
\$80 (dez: 128)									¡	¢	£	¤	¥	¦	§	¨
\$90 (dez: 144)	©	ª	«	¬	­	®	¯	°	±	²	³	´	µ	¶	·	¸
\$A0 (dez: 160)	¹	º	»	¼	½	¾	¿									
\$B0 (dez: 176)	¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	­	®	¯	°
\$C0 (dez: 192)	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿	
\$D0 (dez: 208)									¡	¢	£	¤	¥	¦	§	¨
\$E0 (dez: 224)	©	ª	«	¬	­	®	¯	°	±	²	³	´	µ	¶	·	¸
\$F0 (dez: 240)	¹	º	»	¼	½	¾	¿									

Font 2  
(Codes 159..255 nur ab Rev.B)

+ 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)	0	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
\$50 (dez: 80)	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
\$60 (dez: 96)	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
\$70 (dez: 112)	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
\$80 (dez: 128)									¡	¢	£	¤	¥	¦	§	¨
\$90 (dez: 144)	©	ª	«	¬	­	®	¯	°	±	²	³	´	µ	¶	·	¸

Font 3  
(nur ab Rev.B)



### ALLE GRAFIKFUNKTIONEN AUF EINEN BLICK

Die Grafikeinheit läßt sich über diverse eingebaute Befehle programmieren. Jeder Befehl beginnt mit einem Befehlsbuchstaben, gefolgt von einigen Parametern.

Befehlstabelle EA GE120-5NV24								
Befehl							Anmerkung	
<b>Funktionen zur Textausgabe</b>								
Text-Modus	T	R	n1	mst				Zeichenkette nach Rechts schreiben; n1: Verknüpfungsmodus für Text n1: 1=setzen; 2=löschen; 3=invers; 4=Replace; 5=Invers Replace; mst: Muster Nr. 0..7 verwenden;
Font einstellen	F	n1	n2	n3				Font Nr. n1 einstellen; n1=1:4x6 Font; n1=2:6x8 Font n2+n3=Zoomfaktor (1..4); n2=X-Faktor; n3=Y-Faktor;
ASCII-Zeichen setzen	A	x1	y1	n1				Das Zeichen n1 wird an Koordinate x1,y1 gesetzt. (Bezug links oben)
Zeichenkette ausgeben	Z	x1	y1	...	NUL			Eine Zeichenkette (...) an x1,y1 ausgeben; Zeichen 'NUL' (\$00)=Ende
Zeichen definieren	E	n1	daten ...					n1=Zeichen Nr.; daten=Anzahl Bytes je nach akt. Font
<b>Grafik-Befehle mit Verknüpfungsmodus</b>								
Grafik-Modus	V	n1						n1: 1=setzen; 2=löschen; 3=invers; 4=Replace; 5=Invers Replace;
Punkt setzen	P	x1	y1					Ein Pixel an die Koordinaten x1, y1 setzen
Gerade zeichnen	G	x1	y1	x2	y2			Eine Gerade von x1,y1 nach x2,y2 zeichnen
Gerade weiter zeichnen	W	x1	y1					Eine Gerade vom letzten Endpunkt bis x1, y1 zeichnen
Rechteck zeichnen	R	x1	y1	x2	y2			Ein Rechteck zeichnen; x1,y1,x2,y2 = Gegenüberliegende Eckpunkte
Bereich m. Füllmuster	M	x1	y1	x2	y2	mst		Ein Bereich mit Muster mst (0..7) zeichnen; x1,y1,x2,y2 = Eckpunkte
<b>sonstige Grafik-Befehle</b>								
Display löschen	D	L						Gesamten Displayinhalt löschen (auf weiß setzen);
Display invertieren	D	I						Gesamten Displayinhalt invertieren;
Display füllen	D	S						Gesamten Displayinhalt füllen; (auf schwarz setzen);
Bereich löschen	L	x1	y1	x2	y2			Einen Bereich löschen; x1,y1,x2,y2 = Gegenüberliegende Eckpunkte
Bereich invertieren	I	x1	y1	x2	y2			Einen Bereich invertieren; x1,y1,x2,y2 = Gegenüberliegende Eckpunkte
Bereich füllen	S	x1	y1	x2	y2			Einen Bereich füllen; x1,y1,x2,y2 = Gegenüberliegende Eckpunkte
Box zeichnen	O	x1	y1	x2	y2	mst		Ein Rechteck mit Füllmuster mst (0..7) zeichnen; (immer Replace)
Bildbereich Uploaden	U	x1	y1	daten ...				Einen Bildbereich nach x1,y1 laden; daten des Bildes siehe Bildaufbau
<b>Kontroll-Befehle</b>								
Displaykontrast	K	n1						n1: 0..20 Kontrast für Display einstellen
Ausgang setzen	Y	n1	n2					n1=0..7: OUT n1 rücksetzen (n2=0); setzen (n2=1); invertieren (n2=2) n1=8: Alle 8 Ausgänge entsprechend n2 (=8-Bit Binärwert) einstellen
Display Reset (Rev.B)	D	R						Display wird rückgesetzt. Der Inhalt muss neu aufgebaut werden.
LED Ein/Aus (Rev.B)	D	Y	n1					LED-Beleuchtung n1=0: AUS; n1=1: EIN; n1=2: LED toggeln; n1=3..255: LED für n1 Zehntel Sek.. lang einschalten

### PARAMETER

Der High-Level Grafikkontroller läßt sich über diverse eingebaute Befehle programmieren. Jeder Befehl beginnt mit einem Befehlsbuchstaben, gefolgt von 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. Carriage Return.

**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; gültig sind Werte von 0..119 für x- bzw. 0..31 für y-Koordinaten.  
Beispiel: x1 = 10 (dez.) = \$0A

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

### PROGRAMMIERBEISPIEL

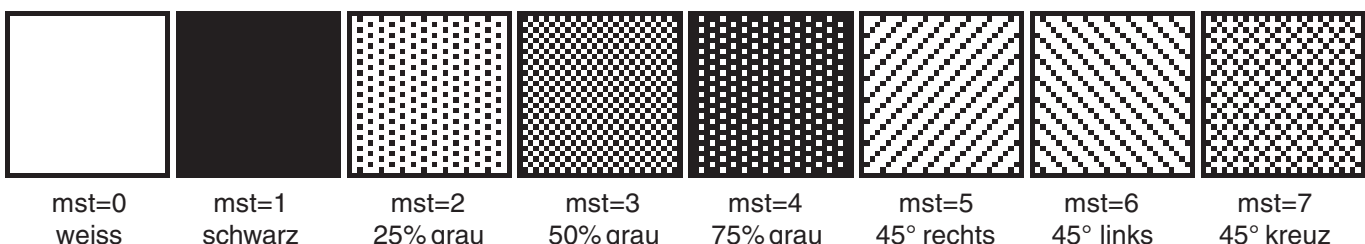
In der nachfolgenden Tabelle ist ein Beispiel zu sehen, welches die Zeichenkette "Test" an den Koordinaten 7,3 ausgibt.

Beispiel	Auszugebende Codes							
	Z	BEL	ETX	T	e	s	t	NUL
in ASCII	Z	BEL	ETX	T	e	s	t	NUL
in Hex	\$5A	\$07	\$03	\$54	\$65	\$73	\$74	\$00
in Dezimal	90	7	3	84	101	115	116	0
für Turbo-Pascal	write(aux, 'Z', chr(7), chr(3), 'Test', chr(0));							
für 'C'	fprintf(stdaux, "%c%c%c%c%s%c", 'Z', 7, 3, "Test", 0);							
für Q-Basic	OPEN "COM1:1200,N,8,2,BIN" FOR RANDOM AS #1 PRINT #1,"Z"+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.

#### A x1 y1 n1

Ein Zeichen **n1** wird an die Koordinate **x1,y1** unter Beachtung des eingestellten Fonts 'F' und des Textmodus 'T' (setzen / löschen / invertieren / replace / invers replace / Füllmuster) ausgegeben. Der Ursprung (0,0) liegt im linken oberen Eck des Displays. Die Koordinatenangaben beziehen sich auf das linke obere Eck des Zeichens. Achtung: Font Nr. 1 zeigt nur Großbuchstaben.

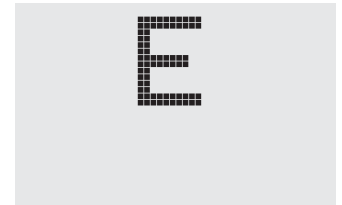
Beispiel: \$41 \$13 \$02 \$45

Zeichen 'E' wird an Koordinate 19,2 ausgegeben.

Eingestellter Font: 6x8 mit 2-facher Breite und 2-facher Höhe

Textmodus: Replace und Muster Schwarz

#### ASCII-Zeichen setzen



#### D L/I/S

Der gesamte Displayinhalt wird **L**=gelöscht (weiss), **I**=invertiert oder **S**=gefüllt (schwarz)

Beispiel: \$44 \$49

invertiert den gesamten Displayinhalt

#### Display Befehl

#### D Y n1

#### (ab Rev. B) Beleuchtung Ein-/Aus schalten

Die 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 \$59 \$4C \$64

nach diesem Befehl leuchtet die Hintergrundbeleuchtung 10s lang.

#### E n1 daten Zeichen definieren

Es ist möglich 8 Zeichen selbst zu definieren. Diese Zeichen haben dann die ASCII Codes 1 bis 8 und bleiben bis zum Abschalten der Versorgungsspannung in einem internen RAM-Bereich erhalten. Achtung! Es kann ein ASCII-Code-Nr. immer nur in einer Fontgröße definiert werden.

Beispiel 1:

Mit dem Befehl

\$45 \$03

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

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

Beispiel 2:

Mit dem Befehl

\$45 \$02

\$00 \$00 \$00 \$FF \$00 \$00 \$00 \$0

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

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

	BYTE NR.					
	1	2	3	4	5	6
Bit 0						
Bit 1						
Bit 2						
Bit 3						
Bit 4						
Bit 5						
Bit 6						
Bit 7						

	BYTE NR.							
	1	2	3	4	5	6	7	8
Bit 0								
Bit 1								
Bit 2								
Bit 3								
Bit 4								
Bit 5								
Bit 6								
Bit 7								
	9	10	11	12	13	14	15	16
	BYTE NR.							

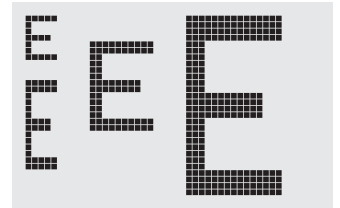
**F n1 n2 n3**

Es wird der Font mit der Nr. **n1** (1=4x6 nur Großbuchstaben; 2=6x8; 3=8x16) eingestellt. Ausserdem wird ein Vergrößerungsfaktor (1..4-fach) für die Breite **n2** und für die Höhe **n3** getrennt eingestellt.

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

ab sofort ist der 6x8- Font mit 3-facher Breite und 4-facher Höhe eingestellt.

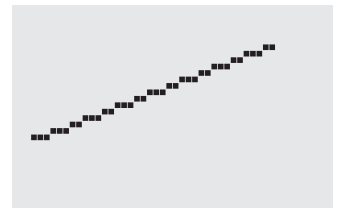
Im Bild nebenan ist das Zeichen 'E' aus dem 6x8 Font mit unterschiedlichen Vergrößerungen dargestellt.

**Font einstellen****G x1 y1 x2 y2**

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

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

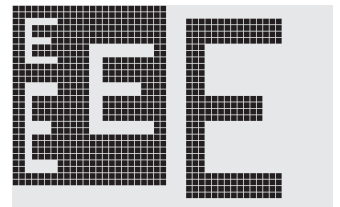
Es wird eine Gerade von 3,20 nach 50,6 gezeichnet.

**Gerade zeichnen****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: \$49 \$00 \$00 \$17 \$1B

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

**Bereich invertieren****K n1**

Mit dem Befehl K n1 kann der Kontrast des Displays verändert werden. Der Standardwert nach dem Einschalten liegt bei n1=8; n1=9..20 läßt das Display dunkler werden, n1=0..7 sorgt für einen schwächeren Kontrast.

Beispiel: \$4B \$14

Der Kontrast wird auf den max. Wert 20 eingestellt (Display ist "schwarz").

**Displaykontrast einstellen****L x1 y1 x2 y2**

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

Beispiel:

\$44 \$53

\$4C \$06 \$04 \$28 \$19

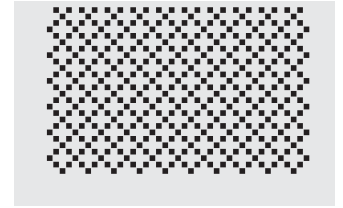
Zuerst wird das Display mir 'D', 'S' gefüllt und dann der Bereich von 6,4 nach 40,25 gelöscht .

**Bereich löschen****M x1 y1 x2 y2 mst****Bereich mit Füllmuster**

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 'V' (setzen / löschen / invertieren / replace / invers replace) gezeichnet.

Beispiel: \$4D \$05 \$01 \$2D \$1A \$07

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



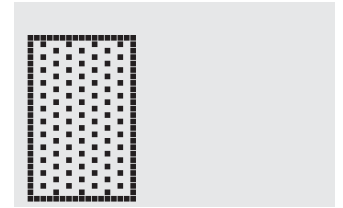
**O x1 y1 x2 y2 mst**

**Box zeichnen**

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 'R' Rechteck zeichnen. Achtung: Eine Box muß mindestens 3 Pixel breit und 3 Pixel hoch sein!

Beispiel: \$4F \$02 \$05 \$12 \$1E \$02

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



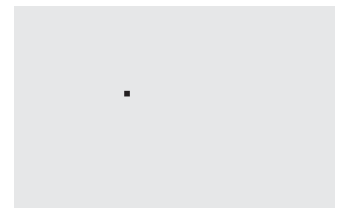
**P x1 y1**

**Punkt setzen**

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

Beispiel: \$50 \$0D \$11

setzt den Pixel an der Koordinate 17,13.



**R x1 y1 x2 y2**

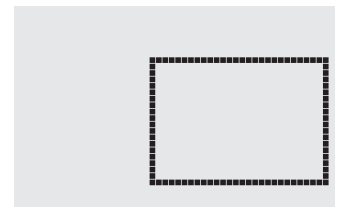
**Rechteck zeichnen**

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. Achtung: Ein Rechteck muß mindestens 3 Pixel breit und 3 Pixel hoch sein!

Vergleiche 'O' Rundeck zeichnen.

Beispiel: \$52 \$15 \$08 \$30 \$25

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



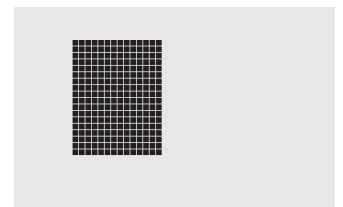
**S x1 y1 x2 y2**

**Bereich füllen**

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

Beispiel: \$53 \$09 \$05 \$16 \$16

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





## ELECTRONIC ASSEMBLY

### T R n1 mst

Der Verknüpfungsmodus **n1** und das Muster **mst** wird für die beiden Textfunktionen ASCII-Zeichen setzen 'A' und Zeichenkette ausgeben 'Z' eingestellt.

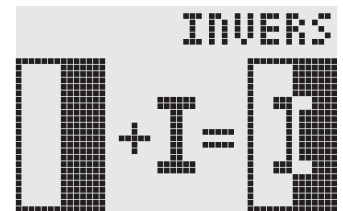
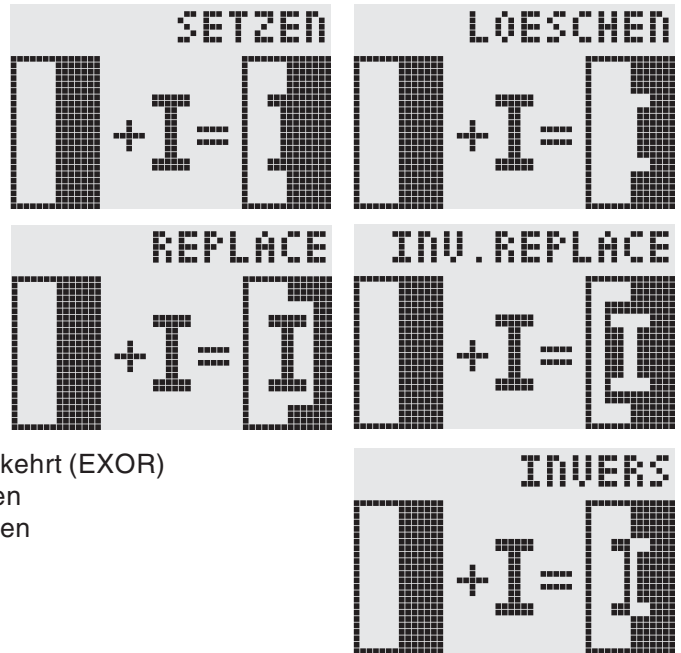
Beispiel: \$54 \$52 \$03 \$03

stellt den Verknüpfungsmodus für folgende Textfunktionen auf graue Zeichen (Muster 3 = 50%Gru) 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

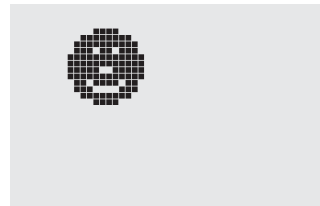
### Text-Modus einstellen



### U x1 y1 daten

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 erzeugt aus monochromen Windows-Bitmap-Grafiken die Bilddaten inkl. der Angabe von Breite und Höhe.



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													
BYTE NR.		13	14	15	16	17	18	19	20	21	22	23	24

Beispiel: \$55 \$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.

### V n1 Grafik-Modus einstellen

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

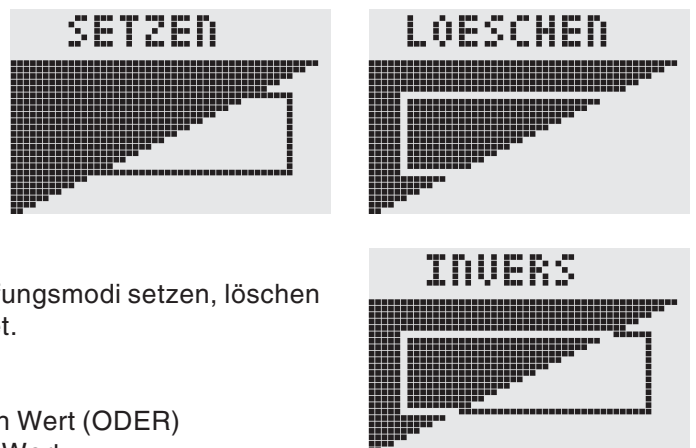
Beispiel: \$56 \$03

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 'M'
- 5=invers replace: Hintergrund füllen, Pixel löschen; nur Bereich mit Füllmuster 'M'



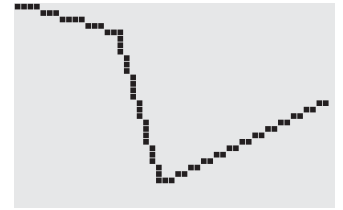
### W x1 y1

### Gerade weiterzeichnen

Zieht eine Gerade vom zuletzt gezeichneten Geradenende bzw. Punkt (siehe Seite 3 Last xy) bis nach **x1,y1** unter Beachtung des eingestellten Grafik-Modus 'V'

Beispiel:

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



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

### Y n1 n2

### Ausgänge einstellen

Ändert den Port (**n1**: 0..7 = OUT0..7) auf den Wert **n2** (0=L-Pegel; 1=H-Pegel; 2=Port invertieren). Wenn **n1= 8**, werden alle Ausgänge OUT0..7 als Binärwert **n2** ausgegeben; OUT0: LSB, OUT7: MSB. Siehe Applikation auf Seite 3.

Beispiel: \$59 \$02 \$01

schaltet OUT2 auf H-Pegel

### Z x1 y1 ASCII... NUL

### Zeichenkette schreiben

Schreibt an die Koordinate **x1,y1** die Zeichenkette **ASCII...** unter Beachtung des eingestellten Textmodus 'T' (setzen / löschen / invertieren / replace / invers replace / Füllmuster/ Richtung). Die Zeichenkette muß mit **NUL** (\$00) abgeschlossen werden. Der Ursprung (0,0) liegt im linken oberen Eck des Displays. Die Koordinaten-angaben beziehen sich auf das linke obere Eck des Zeichens.

Beispiel: \$5A \$06 \$0B \$54 \$65 \$73 \$74 \$00

schreibt an die Koordinate 6,11 die Zeichenkette "Test". Eingestellter Font: 8x16 mit normaler Breite und Höhe  
Textmodus: Schreibrichtung nach Rechts, Verknüpfung Replace mit Muster Schwarz



# EA GE120-5NV24

## ELECTRONIC ASSEMBLY

### EA GE128-6N3V24 LOWCOST GRAFIK EINHEIT 128x64, 3 FONTS, RS-232

<b>Abmessungen</b>	84 x 60 x 25mm
<b>Display</b>	128x64, Supertwist, LED- beleuchtet
<b>Betriebsspannung</b>	+5V / 150mA
<b>Anschluß</b>	10 Pins, Raster 2,54mm
<b>Arbeitstemperatur</b>	0 ... +70°C
<b>Lagertemperatur</b>	-20 ... +70°C



- \* 128x64 Pixel Supertwist-LCD mit LED-Beleuchtung grün
- \* 3 FONTS von 2,6mm (10x32 Zeichen) über 6,2mm (4x16) bis 25 mm (1x5)
- \* GERADEN ZEICHNEN, BEREICHE LÖSCHEN UND INVERTIEREN
- \* AUTOMATISCH SKALIERENDE BARGRAPHFUNKTION
- \* VERSORGUNGSSPANNUNG: +5V / typ. 150mA
- \* BAUDRATEN 1200,2400,4800,9600,19200BD

#### Bestellbezeichnung

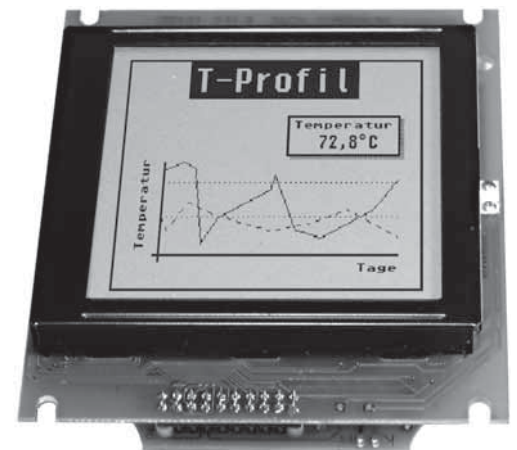
Grafikeinheit 128x64 mit LED-Bel.

EA GE128-6N3V24

Kabel mit 9-pol. D-SUB Stecker (female) EA KV24-9B

### EA GE128-7KV24 LOWCOST GRAFIK EINHEIT 128x128, 3 FONTS, RS-232

<b>Abmessungen</b>	85 x 100 x 25mm
<b>Display</b>	128x128, Supertwist, LED- beleuchtet
<b>Betriebsspannung</b>	+5V / 500mA
<b>Anschluß</b>	10 Pins, Raster 2,54mm
<b>Arbeitstemperatur</b>	0 ... +50°C
<b>Lagertemperatur</b>	-20 ... +70°C



- \* 128x128 Pixel Supertwist-LCD mit LED-Beleuchtung grün
- \* 3 FONTS von 2,2mm (21x32 Zeichen) über 5,2mm (8x16) bis zu 21 mm (2x5)
- \* GERADEN ZEICHNEN, BEREICHE LÖSCHEN UND INVERTIEREN
- \* AUTOMATISCH SKALIERENDE BARGRAPHFUNKTION

VERSORGUNGSSPANNUNG: +5V / typ. 500mA

\* BAUDRATEN 1200,2400,4800,9600,19200BD

#### Bestellbezeichnung

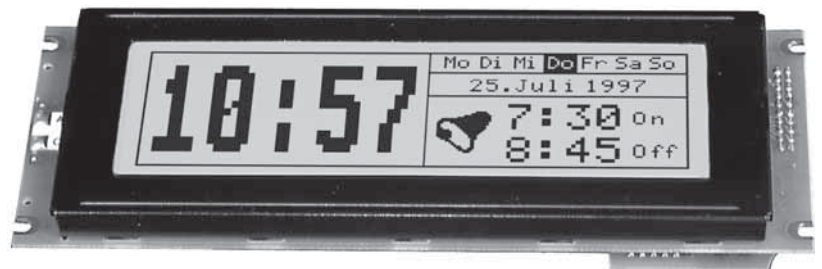
Grafikeinheit 128x128 mit LED-Bel.

EA GE128-7KV24

Kabel mit 9-pol. D-SUB Stecker (female) EA KV24-9B

### EA GE240-6K2V24 LOWCOST GRAFIK EINHEIT 240x64, 3 FONTS, RS-232

<b>Abmessungen</b>	180 x 65 x 25mm
<b>Display</b>	240x64, STN, LED-Bel.
<b>Betriebsspannung</b>	+5V / 500mA
<b>Anschluß</b>	10 Pins, Raster 2,54mm
<b>Arbeitstemperatur</b>	0 ... +50°C
<b>Lagertemperatur</b>	-20 ... +70°C



- \* 240x64 Pixel Supertwist-LCD mit LED-Beleuchtung grün
- \* 3 FONTS von 2,6mm (10x60 Zeichen) über 6,4mm (4x21) bis 25 mm (1x10)
- \* GERADEN ZEICHNEN, BEREICHE LÖSCHEN UND INVERTIEREN
- \* VERSORGUNGSSPANNUNG: +5V / typ. 500mA
- \* BAUDRATEN 1200,2400,4800,9600,19200BD

#### Bestellbezeichnung

Grafikeinheit 240x64 mit LED-Bel.

EA GE240-6K2V24

Kabel mit 9-pol. D-SUB Stecker (female)

EA KV24-9B

Frontrahmen (Fenster 131,0x38,0mm)

EA 017-10UKE

# EA GE120-5NV24

## BAUDRATEN

Die Baudrate läßt sich über 2 Lötbrücken *Baud 0* und *Baud 1* einstellen. Im Auslieferungszustand (Baud 0 geschlossen) sind 9.600 Baud eingestellt. Das Datenformat ist fest eingestellt auf 8 Datenbits, 1 Stopbit, keine Parität. Bitte beachten Sie, daß der interne Datenpuffer lediglich 20 Byte (Rev.B 80 Byte) umfaßt. Beim Senden größerer Datenmengen sollte unbedingt die Handshakeleitung RTS abgefragt werden: +5V Pegel = Daten können angenommen werden; 0V Pegel = Display ist Busy. Sind die Pegellötbrücken Richtung *lo* geschlossen, liefert der Pin RTS statt dessen 0V für Ready und +5V für Busy.

Baudraten		
Lötbrücken		Datenformat 8,N,1
Baud 0	Baud 1	
zu	zu	1200
offen	zu	2400
zu	offen	9600
offen	offen	19200

## PINBELEGUNG

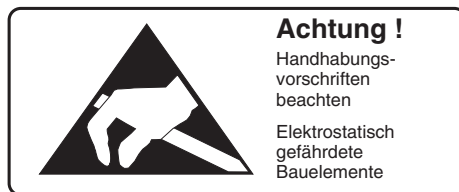
Über eine 10-pol. Stiftleiste wird die Versorgungsspannung (+5V) und die RS-232 Daten eingespeist. Das Display erwartet "echte" RS-232 Pegel<sup>\*)</sup>, wenn die beiden Pegellötbrücken *hi* nach oben hin geschlossen sind (Auslieferungszustand, siehe Zeichnung). Werden die Lötbrücken nach unten hin geschlossen (*lo*), kann das Display direkt über einen uC Ausgang angesteuert werden (5V- Pegel).

RS-232 Anschluß J1			
Pin	Symbol	In/Out	Funktion
1	VDD	-	+ 5V Versorgung
2	DCD	-	Brücke nach DTR
3	DSR	-	Brücke nach DTR
4	NC	-	kein Anschluß
5	NC	-	kein Anschluß
6	RxD	In	Receive Data
7	RTS	Out	Request To Send
8	DTR	-	siehe Pin 2, Pin 3
9	NC	-	kein Anschluß
10	GND	-	0V Masse

Anschluß J2			
Pin	Symbol	In/Out	Funktion
1	VDD	-	+ 5V Versorgung
2..9	OUT0..7	Out	Ausgang
10	GND	-	0V Masse

An der Lötäugenleiste J2 liegen die digitalen Ausgänge OUT0..OUT7 zur weiteren Verwendung an. Eine nähere Beschreibung finden Sie auf der Seite 3.

<sup>\*)</sup> Die RS-232 Spezifikation definiert -12V für den Ruhepegel und +12V für aktiv high. Aus technischen Gründen liefert der Pin 7 (RTS) jedoch 0 und +5V Pegel, was jedoch für 99% aller Schnittstellenbausteine einwandfrei als low und high erkannt wird. Sollte es doch einmal zu Problemen in der Kommunikation kommen, empfehlen wir den Einsatz eines externen Schnittstellenbausteins (z.B. MAX 202, Lötbrücken auf *lo*).



## ABMESSUNGEN

alle Maße in mm

