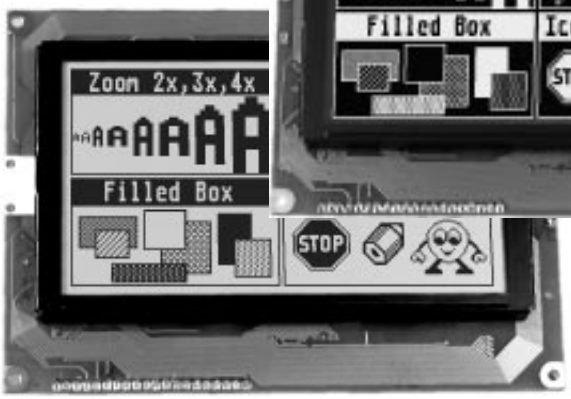
KOMPLETTEINHEIT MIT 3 FONTS UND INTELLIGENTEM KONTROLLER

128x128:
EA GE128-7KV24

240x128: EA GE240-6KV24

240x128:
EA GE240-7KV24



240x128:
EA GE240-7KCV24

TECHNISCHE DATEN

* LCD GRAFIKDISPLAYS MIT HIGH-LEVEL-GRAFIKKONTROLLER EA IC6963
* 240x128 PIXEL MIT CFL-BEL. BLAU NEGATIV, ABM: 151 x 104 x 25 mm
* 240x128 PIXEL MIT LED-BELEUCHTUNG GN/GB, ABM: 144 x 104 x 25 mm
* 128x128 PIXEL MIT LED-BELEUCHTUNG GN/GB, ABM: 85 x 100 x 25 mm
* 240x64 PIXEL MIT LED-BELEUCHTUNG GN/GB, ABM: 180 x 65 x 25 mm
* 3 FONTS (ZOOM) VON ca. 2mm ÜBER ca. 5mm BIS ZU ca. 50mm
* VERSORGUNGSSPANNUNG: +5V / 500..1000mA
* RS-232 BAUDRATEN 1200..115200 BD
* PIXELGENAUE POSITIONIERUNG BEI ALLEN FUNKTIONEN
* PROGRAMMIERUNG ÜBER HOCHSPRACHENÄHNLICHE BEFEHLE:
* GERADE, PUNKT, BEREICH, UND/ODER/EXOR, BARGRAPH...
* BIS ZU 21 FREI DEFINIERBARE ZEICHEN
* TEXT UND GRAFIK MISCHEN
* 6 CLIPBOARD-FUNKTIONEN, CURSORFUNKTIONEN

ZUBEHÖR

* KABEL FÜR ANSCHLUß AN 9-POL. SUB-D (FEMALE): **EA KV24-9B**
* DISKETTE MIT KONVERTIERPROG. FÜR BMP-BILDER (PC-DOS): **EA DISKIC1**
* DIP-SCHALTER Z.B. ZUR EINSTELLUNG DER BAURATE: **EA OPT-DIP6**

BESTELLBEZEICHNUNG

GRAFIKEINHEIT 240x128 MIT CFL-BEL., BLAU NEGATIV	**EA GE240-7KCV24**
GRAFIKEINHEIT 240x128 MIT LED-BELEUCHTUNG GN/GB	**EA GE240-7KV24**
GRAFIKEINHEIT 128x128 MIT LED-BELEUCHTUNG GN/GB	**EA GE128-7KV24**
GRAFIKEINHEIT 240x64 MIT LED-BELEUCHTUNG GN/GB	**EA GE240-6KV24**

ELECTRONIC ASSEMBLY GMBH
LOCHHAMER SCHLAG 17 · D-82166 GRÄFELFING
TELEFON 089/8541991 · TELEFAX 089/8541721

GRAFIKEINHEITEN

ALLGEMEINES

Die Grafik-LCD´s mit EA IC6963 sind komplett aufgebaute Grafikeinheiten mit diversen eingebauten Funktionen. Aufgrund der kleinen Außenabmessungen, dem sehr guten Supertwistkontrast und der einfachen Programmierung ist es innerhalb weniger Stunden möglich, an nahezu jedes Prozessorsystem einen informativen und optisch ansprechenden Bildschirm anzuschließen. Die Ansteuerung erfolgt über eine Standard Schnittstelle RS-232. Das Display enthält komplette Grafikroutinen zur Displayausgabe sowie verschiendenste 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:

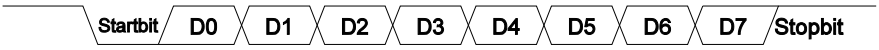
- keine Timingprobleme bei schnellem Prozessorbus
- keine Speicherplatzprobleme (Arbeitsspeicher und Speicher für den Zeichensatz v.a. bei µC)
- keine zeitaufwendigen Grafikberechnungen welche die Prozessorgeschwindigkeit belasten.

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

HARDWARE

Die Displays sind für +5V Betriebsspannung ausgelegt. Die Datenübertragung erfolgt seriell asynchron im RS-232 Format mit echten V.24 Pegeln (±10V) oder über 5V CMOS Pegeln. Das Übertragungsformat ist fest auf 8 Datenbits, 1 Stopbit, no Parity eingestellt. Die Baudrate kann über 3 Lötbrücken von 1200 Baud bis zu 115.200 Baud ausgewählt werden. Handshakeleitungen RTS und CTS stehen zur Verfügung. Bei kleinen Datenmengen ist eine Auswertung nicht erforderlich.

Datenformat:



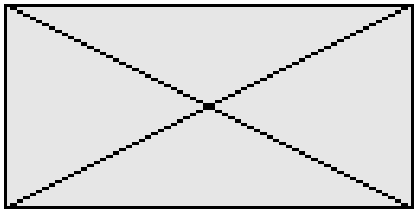
Zusätzlich sind an der Lötaugenleiste J3 8 I/O-Ports zur freien Verwendung vorhanden. Diese können sowohl als Aus- als auch als Eingänge individuell geschaltet werden. Mögliche Anwendungen dafür sind das Schalten eines Transisitors/Relais (IL_{max}=10mA) oder das Einlesen von Tasten/Schaltern.

SOFTWARE

Die Programmierung der Grafikeinheit erfolgt über Befehle wie z.B. Zeichne ein Rechteck von (0,0) nach (64,15). Der Ursprung liegt im linken oberen Eck des Displays. Über die serielle Schnittstelle müssen dazu folgende Bytes gesendet werden: $52 $00 $00 $40 $0F. Zeichenketten lassen sich ebenso pixelgenau plazieren. Das Mischen von Text und Grafik ist jederzeit möglich. Es können 3 verschiedene Zeichensätze verwendet werden. Jeder Zeichensatz kann wiederum 2- bis 8-fach gezoomt werden. Mit dem größten Zeichensatz 16x8 lassen sich somit bei 8-fach Zoom (=128x64) bildschirmfüllende Worte und Zahlen darstellen.

TESTMODE

Solange die Lötbrücke 6 (Pin RTS5) nach dem Power-On oder Reset geschlossen ist (mit GND verbunden), befindet sich das Display im Testmode: es blinkt ein Rechteck mit zwei diagonalen Linien. Wird die Lötbrücke geöffnet, dann kehrt das Display in den Normalbetrieb zurück. Das Testbild ist aber immer noch zu sehen.



INTEGRIERTE FONTS

In jeder Grafikeinheit sind 3 Zeichensätze integriert. Jeder Zeichensatz kann in 1- bis 8-facher Höhe verwendet werden. Unabhängig davon läßt sich auch die Breite verdoppeln bis verachtfachen.

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

Font 1: 4x6

Upper \ + Lower	$0 (0)	$1 (1)	$2 (2)	$3 (3)	$4 (4)	$5 (5)	$6 (6)	$7 (7)	$8 (8)	$9 (9)	$A (10)	$B (11)	$C (12)	$D (13)	$E (14)	$F (15)
$20 (dez: 32)		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
$30 (dez: 48)	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
$40 (dez: 64)	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
$50 (dez: 80)	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
$60 (dez: 96)	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
$70 (dez: 112)	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	⌂
$80 (dez: 128)	Ç	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	ì	Ä	Å
$90 (dez: 144)	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	¢	£	¥	ß	

Zusätzlich können, je nach Font, bis zu 21 eigene Zeichen definiert werden, die solange erhalten bleiben, bis die Versorgungsspannung abgeschaltet wird. (Siehe Befehl 'E').

Font 2: 6x8

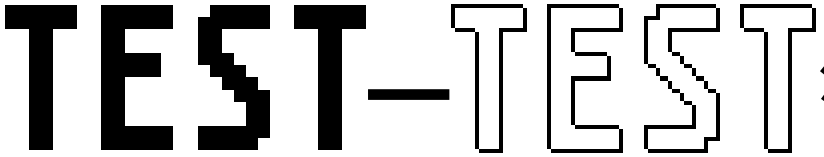
Jedes Zeichen kann pixelgenau plaziert werden. Text und Grafik kann beliebig gemischt dargestellt werden. Auch mehrere verschiedene Schriftgrößen lassen sich gemeinsam darstellen.

Upper \ + Lower	$0 (0)	$1 (1)	$2 (2)	$3 (3)	$4 (4)	$5 (5)	$6 (6)	$7 (7)	$8 (8)	$9 (9)	$A (10)	$B (11)	$C (12)	$D (13)	$E (14)	$F (15)
$20 (dez: 32)		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
$30 (dez: 48)	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
$40 (dez: 64)	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
$50 (dez: 80)	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
$60 (dez: 96)	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
$70 (dez: 112)	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	⌂
$80 (dez: 128)	Ç	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	ì	Ä	Å
$90 (dez: 144)	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	¢	£	¥	ß	

Font 3: 8x16

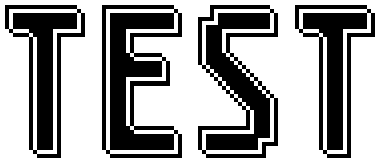
TIP: SCHRIFTEFFEKTE

Mit dem Befehl 'T' TEXT-Modus (Verknüpfung, Muster) können interessante Effekte bei grossen Schriften durch Überlagerung (mehrmaliges versetztes Schreiben eines Wortes) erzielt werden.

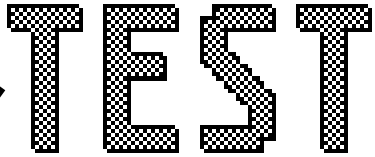


*Orginalschrift 8x16 mit ZOOM 3 an **Position 0,0** mit Muster Schwarz*

*Durch Überlagerung **(EXOR)** an **Pos.1,1** enstandene "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 Schrfit" an **Pos.0,0**. führt zu einer "Schrift mit Musterfüllung"*

Befehlstabelle EA IC6963

Befehl											Anmerkung
Funktionen zur Textausgabe											
Text-Modus	T	R L O U	n1	mst							R/L/O/U: Zeichenkette nach (R)echts,(L)inks,(O)ben, (U)nten schreiben; n1: Verknüpfunsmodus für Textausgabe 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=3:8x16 Font n2+n3=Zoomfaktor (1..8); 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
Grafik-Befehle mit Verknüpfungsmodus											
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
Rundeck zeichnen	N	x1	y1	x2	y2						Ein Rechteck mit runden Ecken zeichnen; x1,y1,x2,y2 = Eckpunkte
Bereich m. Füllmuster	M	x1	y1	x2	y2	mst					Ein Bereich mit Muster mst (0..7) zeichnen; x1,y1,x2,y2 = Eckpunkte
sonstige Grafik-Befehle											
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)
Rundbox zeichnen	J	x1	y1	x2	y2	mst					Ein Rundeck mit Füllmuster mst (0..7) zeichnen; (immer Replace)
Bargraph zeichnen	B	nr	wert								Den Bargraph mit der ´nr´ (1..8) auf den neuen Benutzer-´wert´ setzen
Bildbereich Uploaden	U	x1	y1	daten ...							Einen Bildbereich nach x1,y1 laden; daten des Bildes siehe Bildaufbau
Kontroll- / Definitions-Befehle											
Bargraph definieren	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..8) definieren. x1,y1,x2,y2 sind das umschließende Rechteck des Bargraphs. aw,ew sind die Werte für 0% und 100%. mst=Muster (0..7)
Clipboard-Befehle *) (Zwischenspeicher für Bildbereiche)	C	B									Der gesamte Displayinhalt wird als Bildbereich ins Clipboard kopiert
		S	x1	y1	x2	y2					Der Bildbereich von x1, y1 bis nach x2, y2 wird ins Clipboard kopiert
		R									Der Bildbereich im Clipboard wird wieder ins Display zurückkopiert
		K	x1	y1							Der Bildbereich im Clipboard wird ins Display nach x1, y1 kopiert
		H									Der Bildbereich im Clipboard wird als Hardcopy über RS232 gesendet
		L	daten...								Einen Bildbereich ins Clipboard laden; daten des Bildes siehe Bildaufbau
Automatisch blinkender Bereich (Cursor-Funktion)	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
		M	I								Invers-Modus (Blinkbereich wird invertiert); Blinkfunktion aktivieren
			mst								Clipboard-Modus*) mst=Muster(0..7) des Blockcursors; Blinken aktivieren
Selekt / Deselekt Grafikkontroller	K	S	n1								Kontroller mit Adresse n1 (n1=0..3; n1=255: alle) aktivieren
		D									Kontroller mit Adresse n1 (n1=0..3; n1=255: alle) deaktivieren
I/O-Port schreiben	Y	n1	n2								n1=0..7: I/O-Port n1 rücksetzen (n2=0); setzen (n2=1); invertieren (n2=2) n1=8: Alle 8 I/O-Ports entsprechend n2 (=8-Bit Binärwert) einstellen
Display einstellen	!	n1	n2	LO	HI						Ein anders Display kann eingestellt werden. n1=X-Auflösung (64..240); n2=Y-Auflösung (16..128); LO, HI 16-Bit Bildstartadresse (normal $0000)
Sende-Befehle											
Hardcopy	H	x1	y1	x2	y2						Es wird ein Bereich als Bild angefordert. Zuerst werden die Breite und Höhe in Pixel und dann die eigentlichen Bilddaten über RS232 gesendet.
I/O-Port lesen	X	n1									n1=0..7: I/O-Port <n1> einlesen (1=H-Pegel=5V, 0=L-Pegel=0V) n1=8: Alle 8 I/O-Ports I/O0..I/O7 als 8-Bit Binärwert einlesen
Displaytyp abfragen	?										mit diesem Befehl wird der Displaytyp abgefragt. Zurückgesendet werden 3 Bytes: X-Auflösung, Y-Aufl., 'H' (z.B. 240, 64 (Pixel), horizontales Bild)

*) *Alle Clipboardbefehle benötigen ein Display-RAM mit mindestens 8kB. Bei Displays mit kleinerem RAM (z.B. 2kB) können die Clipboardbefehle nicht verwendet werden!*

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 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

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

PROGRAMMIERBEISPIEL

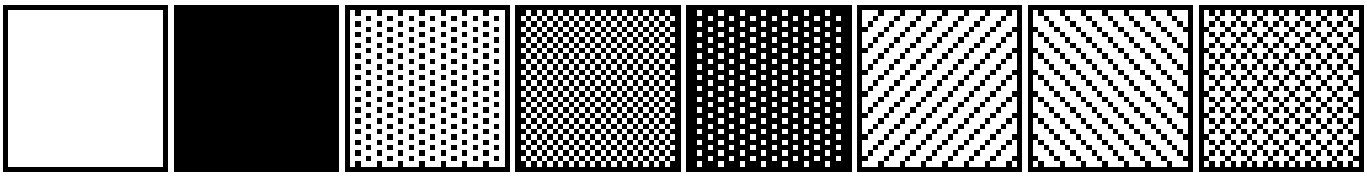
In der nachfolgenden Tabelle ist ein Beispiel zu sehen welches die Zeichenkette "Test" an den Koordinaten 7,3 ausgibt.

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

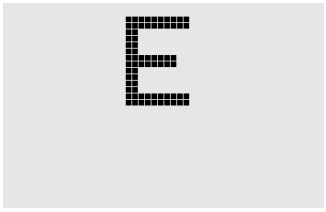


mst=0	mst=1	mst=2	mst=3	mst=4	mst=5	mst=6	mst=7
weiss	schwarz	25% grau	50% grau	75% grau	45° rechts	45° links	45° kreuz

BESCHREIBUNG DER EINZELNEN GRAFIKFUNKTIONEN

Auf den nächsten Seiten befindet sich eine detaillierte alphabetisch sortierte Beschreibung zu jeder einzelnen Funktion. Als Beispiel wird jeweils ein vergößerter 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 — ASCII-Zeichen setzen



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.

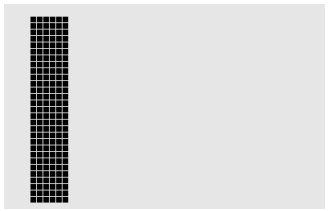
Beispiel: $41 $13 $02 $45

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

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

Textmodus: Replace und Muster Schwarz

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

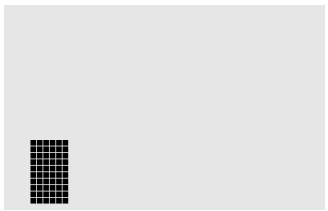


Es können bis zu 8 Bargraphs (**nr**=1..8) definiert werden, welche nach **L**=links, **R**=rechts, **O**=oben oder **U**=unten ausschlagen können. Der Bargraph beansprucht bei Vollauschlag einen Bereich mit den Koordinaten **x1**,**y1** bis **x2**,**y2**. Mit dem Anfangswert (kein Ausschlag) **aw** (=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 dem Ausführen dieses Befehles ist der Bargraph nur definiert, am Display ist er aber noch nicht zu sehen).

Beispiel: $42 $4F $01 $04 $02 $09 $1E $04 $14 $01

Es wird der Bargraph Nr. 1 der nach oben ausschlägt definert. Bei Vollauschlag 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 Vollauschlag wie er mit $42 $01 $14 dargestellt wird)

B nr wert — Bargraph zeichnen



Der Bargraph mit der Nummer **n1** (1..8) 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: $42 $01 $0A

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

C B*) — Displayinhalt ins Clipboard sichern

kopiert den gesamten Displayinhalt in das Clipboard (Zwischenspeicher).

Beispiel: $43 $42

sichert den gesamten Displayinhalt für ein späteres Wiederherstellen des Bildschirms ins Clipboard. Der Displayinhalt wird dabei nicht verändert.

**) Alle Clipboardbefehle benötigen ein Display-RAM mit mindestens 8kB. Bei Displays mit kleinerem RAM (z.B. 2kB) können die Clipboardbefehle nicht verwendet werden!*

C S x1 y1 x2 y2*) — Bereich ins Clipboard sichern

kopiert einen Bereich von der linken oberen Ecke **x1,y1** bis zur rechten unteren Ecke **x2,y2** in das Clipboard (Zwischenspeicher).

Beispiel: $43 $53 $00 $00 $17 $1B

sichert den Bereich von 0,0 nach 23,27 für ein späteres Wiederherstellen des Bildschirms. Der Displayinhalt wird nicht verändert.

C R*) — Bereich wiederherstellen

kopiert den zuletzt gespeicherten Bereich vom Clipboard (Zwischenspeicher) in das Display zurück. Ziel: Ursprüngliche Koordinaten.

Beispiel: $43 $52

stellt den zuletzt gesicherten Bereich wieder her.

C K x1 y1*) — Bereich vom Clipboard kopieren

kopiert den zuletzt gespeicherten Bereich im Clipboard (Zwischenspeicher) an eine neue Position **x1,y1** des Displays.

Beispiel: $43 $4B $0A $20

kopiert den zuletzt gesicherten Bereich an die Koordinate 10,32.

C L daten*) — Bild ins Clipboard laden

lädt die nun folgenden Daten ins Clipboard (Zwischenspeicher).

Beispiel: $43 $4C daten wie bei Befehl ´U´ Upload

Damit kann ein Bild auch mit niedriger Baudrate (langsam) in einen unsichtbaren Bereich geladen werden und "plötzlich" durch den Befehl ´C´, ´K´ an eine oder mehrere Stellen zur Anzeige gebracht werden.

C H*) — Bild aus dem Clipboard als Hardcopy senden

fordert die Daten aus dem Clipboard (Zwischenspeicher) an. Funktion ähnlich dem Befehl ´H´ Hardcopy.

Beispiel: $43 $48

und sofort wird das Bild im Clipboard über RS-232 gesendet.

D L/I/S — Display Befehl

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

Beispiel: $44 $49

invertiert den gesamten Displayinhalt

**) Alle Clipboardbefehle benötigen ein Display-RAM mit mindestens 8kB. Bei Displays mit kleinerem RAM (z.B. 2kB) können die Clipboardbefehle nicht verwendet werden!*

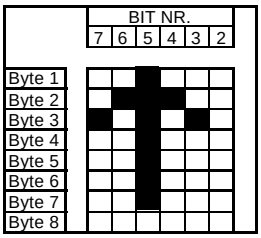
E n1 daten — Zeichen definieren

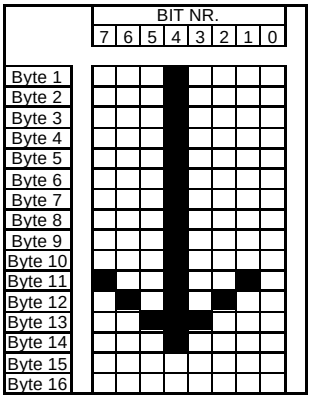
Es ist möglich bis zu 21 Zeichen selbst zu definieren (je nach Fontgröße). Diese Zeichen haben dann die ASCII Codes 1 bis max.21 und bleiben bis zum Abschalten der Versorgungsspannung in einem 128 Byte großen unsichtbaren Bildschirm-RAM erhalten. Bei Font 1 können bis zu 21 Zeichen definiert werden, bei Font 2 noch 16 Zeichen und beim größten Font 3 immmer noch 8 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 3 vom 4x6 Font (siehe Tabelle nebenan) !

Selbstdefinierbare Zeichen (Code)

4x6	6x8	8x16
1	1	1
2		
3	2	
4	3	2
5	4	
6		
7	5	3
8	6	
9	7	4
10	8	
11		
12	9	5
13	10	
14		
15	11	6
16	12	
17		
18	13	7
19	14	
20	15	8
21	16	

Beispiel 1:
$45 $01
$20 $70 $A8 $20 $20 $20 $20 $00
definiert einen Pfeil nach oben für ASCII-Nr. 1 bei eingestelltem 6x8 Zeichensatz.



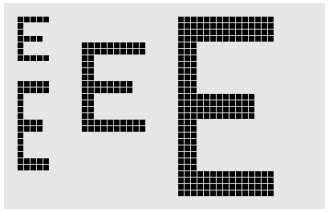


Beispiel 2:
$45 $02
$10 $10 $10 $10 $10 $10 $10 $10 $10 $10 $92 $54 $38 $10 $00 $00
definiert einen Pfeil nach unten für ASCII-Nr. 2, bei eingestelltem 8x16 Zeichensatz.

F n1 n2 n3 — Font einstellen

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

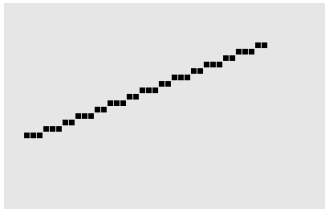
Beispiel: $46 $02 $03 $04
ab sofort ist der 6x8- Font mit 3-facher Breite und 4-facher Höhe eingestellt.
Im Bild nebenan ist das Zeichen 'E' aus dem 6x8 Font mit unterschiedlichen Vergrösserungen dargestellt.



G x1 y1 x2 y2 — Gerade zeichnen

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.



H x1 y1 x2 y2 — Hardcopy vom Displayinhalt erstellen

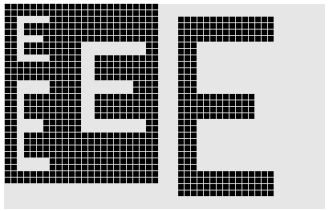
Der Bereich von der linken oberen Ecke **x1,y1** bis zu rechten unteren Ecke **x2,y2** wird angefordert. Der Grafikchip sendet daraufhin sofort die Breite und Höhe des Bildausschnittes und danach die Bilddaten. Zum Aufbau der Bilddaten siehe den Befehl Bild Upload 'U'.

Beispiel: $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.

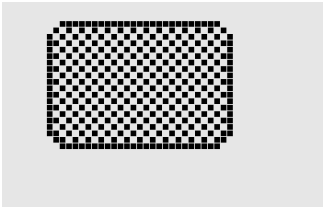
I x1 y1 x2 y2 — Bereich invertieren

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 Beipiel "Font einstellen" den Bereich von 0,0 nach 23,27.



J x1 y1 x2 y2 mst — Rundbox zeichnen



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 'N' Rundeck zeichnen.

Beispiel: $4A $07 $03 $23 $16 $03

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

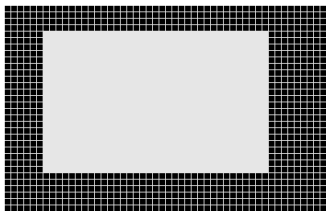
K S/D n1 — Grafikkontroller (de)selektieren

Der Grafikkontroller mit der Hardwareadresse **n1** (0..3) wird **S**=selektiert oder **D**=deselektiert; Die Adresse 255=$FF ist eine Masteradresse mit der alle Grafikkontroller angesprochen werden. Die Adresseinstellung erfolgt per Hardware (Pins ADR0/1 siehe Seite 3).

Beispiel: $4B $44 $00

alle Befehle werden für den Grafikkontroller mit der Adresse $00 ab sofort ignoriert.

L x1 y1 x2 y2 — Bereich löschen

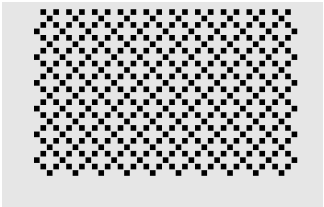


Der Bereich von der linken oberen Ecke **x1,y1** bis zur rechten unteren Ecke **x2,y2** wird gelöscht.

Beispiel: $44 $53 $4C $06 $04 $28 $19

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

M x1 y1 x2 y2 mst — Bereich mit Füllmuster

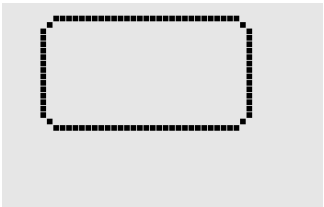


Ein rechteckiger Bereich wird von der linken oberen Ecke **x1,y1** bis zur rechten unteren Ecke **x2,y2** mit dem Muster **mst** unter Beachtung des eingestellten Grafikmodus 'V' (setzen / löschen / invertieren / replace / invers replace) gezeichnet.

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

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

N x1 y1 x2 y2 — Rundeck zeichnen

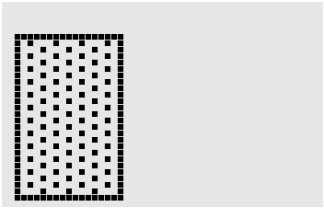


Ein Rechteck mit abgerundeten Ecken wird von der linken oberen Ecke **x1,y1** bis zur rechten unteren Ecke **x2,y2** unter Beachtung des eingestellten Grafikmodus 'V' (setzen / löschen / invers) gezeichnet. Der Inhalt des Rundecks wird nicht verändert. Vergleiche 'J' Rundbox zeichnen.

Beispiel: $4E $06 $02 $26 $13

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

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.

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 $11 $0D

setzt den Pixel an der Koordinate 17,13.

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 automatischer Blinkbereich festgelegt. Zugleich wird die Blinkfunktion gestartet. Hiermit kann z.B bei Eingaben ein "Cursor" nachgebildet werden.

Beispiel: $51 $44 $00 $0F $07 $10

Definiert den Blinkbereich von 0,15 nach 7,16. (Simulation eines Unterstrich Cursors für den 8x16 Font mit einem Zeichen an der Position 0,0).

Q Z n1 — Blinkzeit einstellen

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

Beispiel: $51 $5A $05

stellt die Blinkzeit auf ½ Sekunde ein.

Q M I — Blinkmodus Invers

Der definierte Blinkbereich wird zyklisch mit der eingestellten Blinkzeit automatisch invertiert. Zugleich wird die Blinkfunktion gestartet.

Beispiel: $51 $49

Der Blinkmodus invers wird eingestellt.

Q M mst*) — Blinkmodus Blockcursor

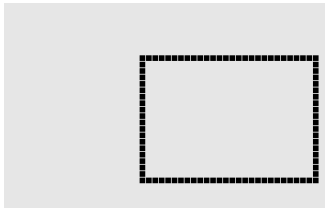
Der definierte Blinkbereich wird im Clipboard gesichert. Mit der eingestellten Blinkzeit wird zyklisch zwischen dem Orginal Bereich und dem Muster mst (=0..7) umgeschaltet. Dadurch kann z.B ein Blockcursor simuliert werden (mst=1 schwarz) oder ein blinkendes Wort angezeigt werden (mst=0 weiss). Zugleich wird die Blinkfunktion gestartet. Die Clipboardbefehle können dann nicht mehr verwendet werden!

Beispiel: $51 $43 $00

Der Blinkmodus Blockcursor mit dem Muster weiss wird eingestellt. Dadurch wird erreicht, daß der eingestellte Bereich auf weissem Hintergrund blinkt.

R x1 y1 x2 y2 — Rechteck zeichnen

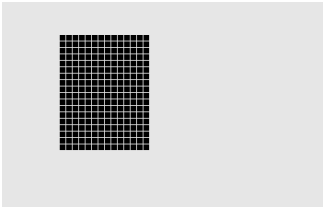
Ein Rechteck wird von der linken oberen Ecke **x1,y1** bis zur rechten unteren Ecke **x2,y2** unter Beachtung des eingestellten Grafikmodus 'V' (setzen / löschen / invers) gezeichnet. Der Inhalt des Rechtecks wird dabei nicht verändert.
Vergleiche 'O' Rundeck zeichnen.



Beispiel: $52 $15 $08 $30 $25
zeichnet ein Rechteck von 21,8 nach 48,37.

*) *Der Befehl Q M mst benötigt ein Display-RAM mit mindestens 8kB. Bei Displays mit kleinerem RAM (z.B. 2kB) kann dieser Befehl nicht verwendet werden!*

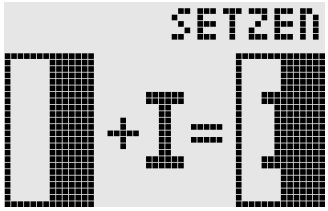
S x1 y1 x2 y2 — Bereich füllen

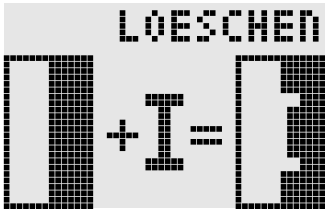


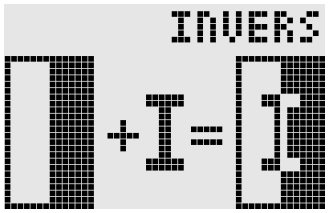
Der Bereich von der linken oberen Ecke **x1,y1** bis zur rechten unteren Ecke **x2,y2** wird gefüllt (auf schwarze Pixel gesetzt).

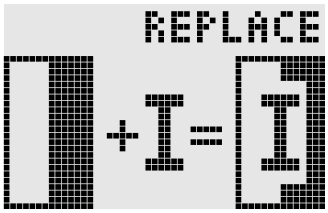
Beispiel: $53 $09 $05 $16 $16
setzt den Bereich von 9,5 nach 22,22 auf schwarz.

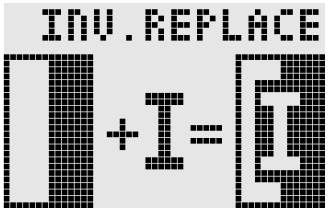
T L/R/O/U n1 mst — Text-Modus einstellen











Der Verknüpfungsmodus **n1** und das Muster **mst** wird für Textfunktionen ASCII-Zeichen setzen 'A' und Zeichenkette ausgeben 'Z' eingestellt. Für den Befehl Zeichenkette ausgeben 'Z' wird außerdem die Schreibrichtung angegeben: **L**=links, **R**=rechts, **O**=oben und **U**=unten.

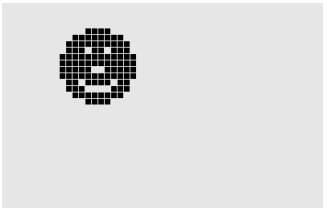
Beispiel: $54 $52 $03 $03

stellt den Verknüpfungsmodus für alle folgenden Textfunktionen auf graue Zeichen (Muster 3 = 50%Grau) invertiert mit dem Hintergrund, Zeichenketten werden nach rechts geschrieben.

Verknüpfungsmodus n1:

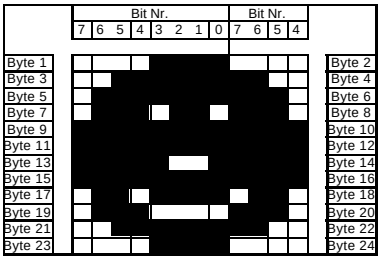
1 = setzen: schwarze Pixel ohne Rücksicht auf den vorigen Wert (ODER)
2 = löschen: weißes Pixel ohne Rücksicht auf den vorigen Wert
3 = invers: aus schwarzen Pixeln werden Weiße und umgekehrt (EXOR)
4 = replace: Hintergrund löschen und schwarze Pixel setzen
5 = invers replace: Hintergrund füllen und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

U 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 = ((Breite+7) / 8) * Höhe Bytes.
1 Byte steht für 8 waagrechte Pixel am Bildschirm; 0=weiß, 1=schwarz;
MSB: links, LSB: rechts; das Bild ist von oben nach unten abgelegt.
Das Programm BMP2BLH.EXE auf der als Zubehör erhältlichen Diskette EA DISKIC1 erzeugt aus monochromen Windows-Bitmap- Grafiken (*.BMP) die Bilddaten inkl. der Angabe von Breite und Höhe.



Beispiel: $55 $09 $04 $0C $0C
$0F $00 $3F $C0 $7F $E0 $76 $E0 $FF $F0 $FF $F0
$F1 $F0 $FF $F0 $6F $60 $70 $E0 $3F $C0 $0F $00

lädt das nebenstehende Bild an die Koordinate 9,4.

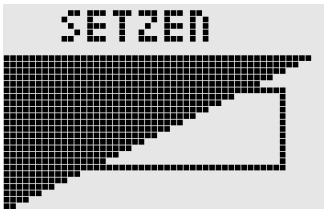
? — Displaytyp abfragen

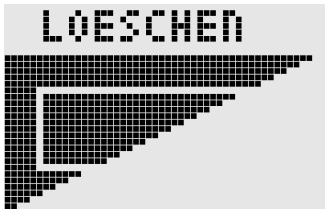
Die Auflösung des Displays und die Art des Bildaufbaus wird abgefragt.

Beispiel: $3F

Nach diesem Befehl wird zuerst die X- und Y-Auflösung und dann die Art des Bildaufbaus ('H') für die horizontale Organisation über die RS-232 Schnittstelle gesendet.

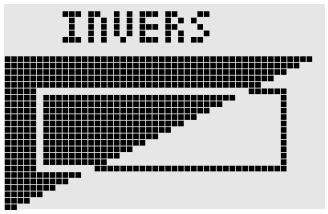
V n1 — Grafik-Modus einstellen





Einstellen des Verknüpfungsmodus **n1** für folgende Grafikfunkionen: 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.

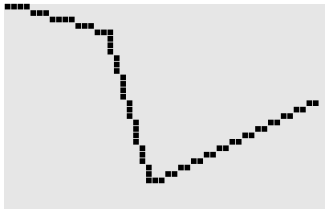


Als Beispiel wird nebenan ein Rechteck mit den Verknüpfungsmodi setzen, löschen und invers auf einen vorhandenem Hintergrund gezeichnet.

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

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:
$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.

X n1 — I/O Port lesen

Liest einen Port (**n1**: 0..7 = I/O: 0..7) ein. Wenn **n1** = 8, werden alle I/O 0..7 als Binärwert eingelesen; I/O 0: LSB, I/O 7: MSB. Siehe Applikation auf Seite 13.

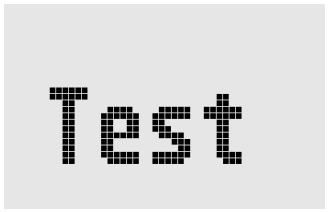
Beispiel: $58 $02
liest den Pegel an I/O 2 ein und sendet bei L-Pegel ein $00 und bei H-Pegel ein $01 über RS-232

Y n1 n2 — I/O Port einstellen

Ändert den Port (**n1**: 0..7 = I/O: 0..7) auf den Wert **n2** (0=L-Pegel; 1=H-Pegel; 2=Port invertieren). Wenn **n1**= 8, werden alle I/O 0..7 als Binärwert **n2** ausgegeben; I/O 0: LSB, I/O 7: MSB. Siehe Applikation auf Seite 13.

Beispiel: $59 $02 $01
schaltet den Port I/O 2 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.

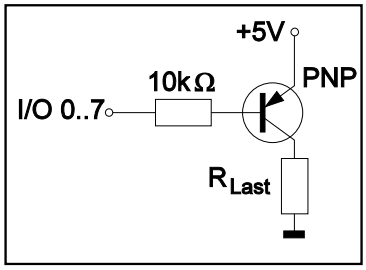
Beispiel: $5A $06 $0B $54 $65 $73 $74 $00
schreibt an die Koordinate 6,11 die Zeichenkette "Test". Eingesteller Font: 8x16 mit normaler Breite und Höhe Textmodus: Schreibrichtung nach Rechts, Verknüpfung Replace mit Muster Schwarz

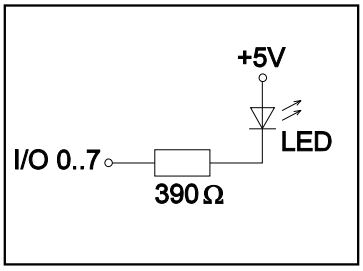
DIGITALE EIN-/ AUSGÄNGE IO 0..7

8 Pins am High-Level Grafikkontroller können als frei programmierbare Ein- und Ausgänge verwendet werden. Auch ein gemischter Betrieb von z.B. 3 Ausgängen und 5 Eingängen ist möglich.

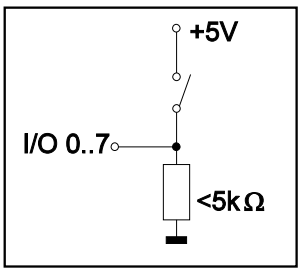
Beschaltung als Ausgang

Mit dem Befehl "Y n1 n2"[1)] kann jeder Pin IO 0..7 auf H- oder L-Pegel geschaltet werden; Strom kann nur bei L-Pegel fließen (interner Pullup). 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. nach einem RESET liegen alle Ausgänge auf H-Pegel.

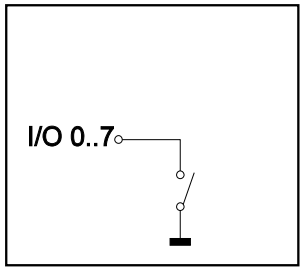




Beschaltung als Eingang



Am Eingang dürfen Spannungspegel zwischen -0,5V und +0,2V*VDD-0,1V anliegen. Der Leckstrom beträgt max. ±10µA. Die Schaltschwellen für den L-Pegel liegt bei <0,2*VDD-0,1V und für den H-Pegel bei >0,2*VDD+0,9V. Mit dem Befehl "X n1"[1)] kann jeder Pin IO 0..7 eingelesen werden. Der Spannungspegel muß während des gesamten Einlesevorgangs stabil sein. Eine Entprellfunktion ist nicht eingebaut.



[1)] *eine Befehlsbeschreibung finden Sie nebenan auf der Seite 12*

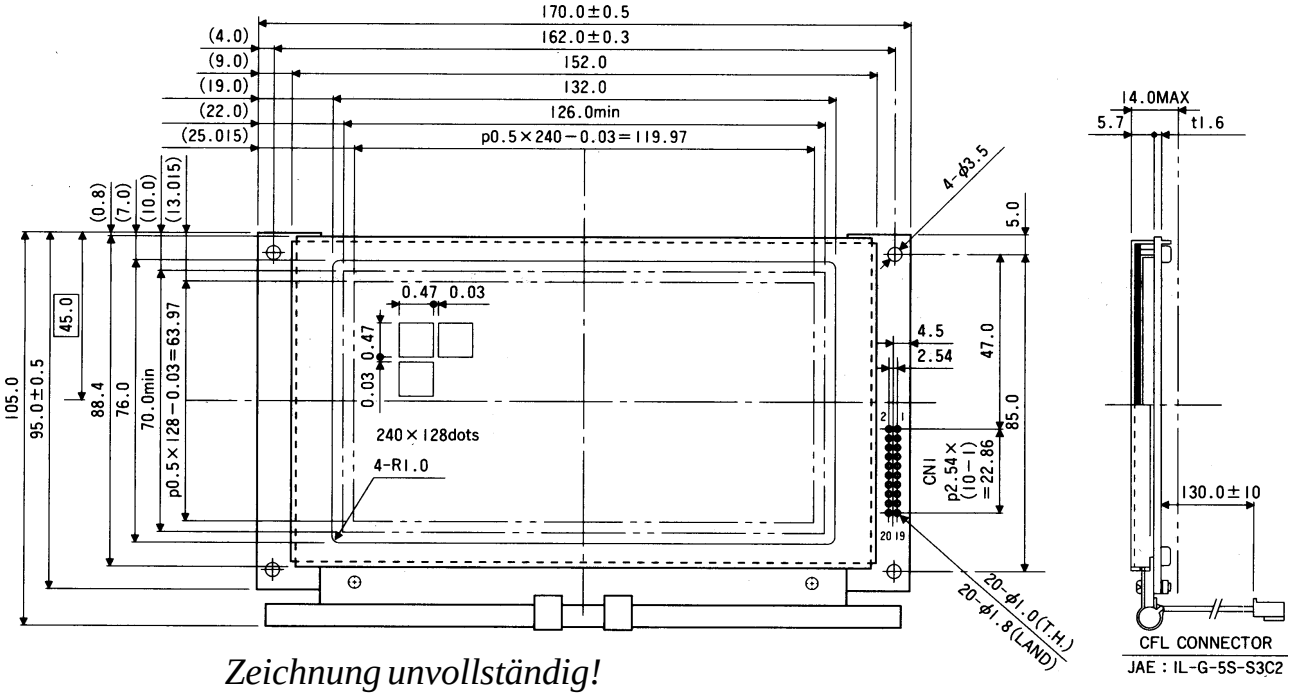
GRUNDEINSTELLUNGEN

Nach dem Einschalten bzw. nach einem manuell ausgelösten Reset werden die nebenstehenden Register auf einen bestimmten Wert voreingestellt.

Grundeinstellungen		
Register	**Befehl**	**nach Power-On / Reset**
Text-Modus	T	rechts,setzen,schwarz
Grafik-Modus	V	setzen
Font	F	6x8
Fontfaktor Breite/Höhe	F	1/1
Last xy	W	(0;0)
Selbst definierte Zeichen	E	undefiniert
Bargraph 1..8	B	undefiniert
High-Level Grafikkontroller	K	selektiert
Blinkbereich	QD	(0;0)
Blinkmodus	QC	invers
Blinkzeit	QZ	0,6 sek.
Clipboard	C	leer
Ein-/ Ausgänge I/O0..7	Y	H-Pegel

EA GE240-7K2CV24

240x128 MIT CFL-BEL., BLAU NEGATIV

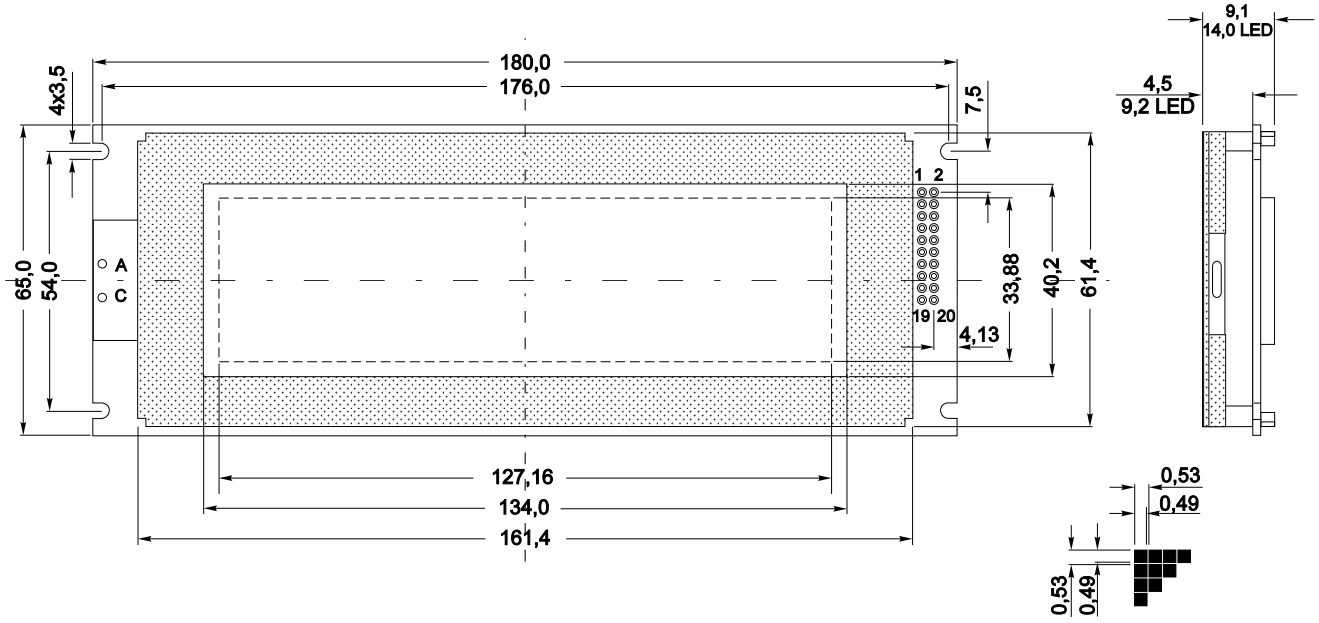


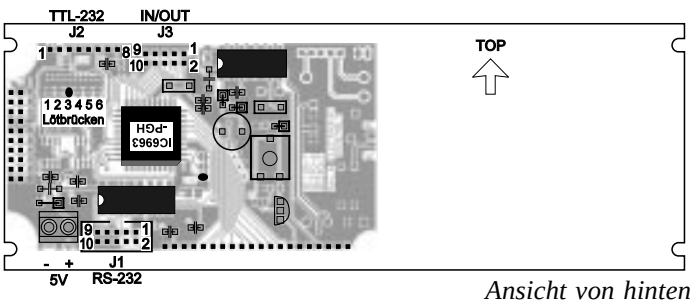
Zeichnung unvollständig!
Bauhöhe max. 30mm!

EA GE240-6KV24

240x64 MIT LED-BELEUCHTUNG GN/GB

Zubehör: Frontrahmen EA 017-10UKE



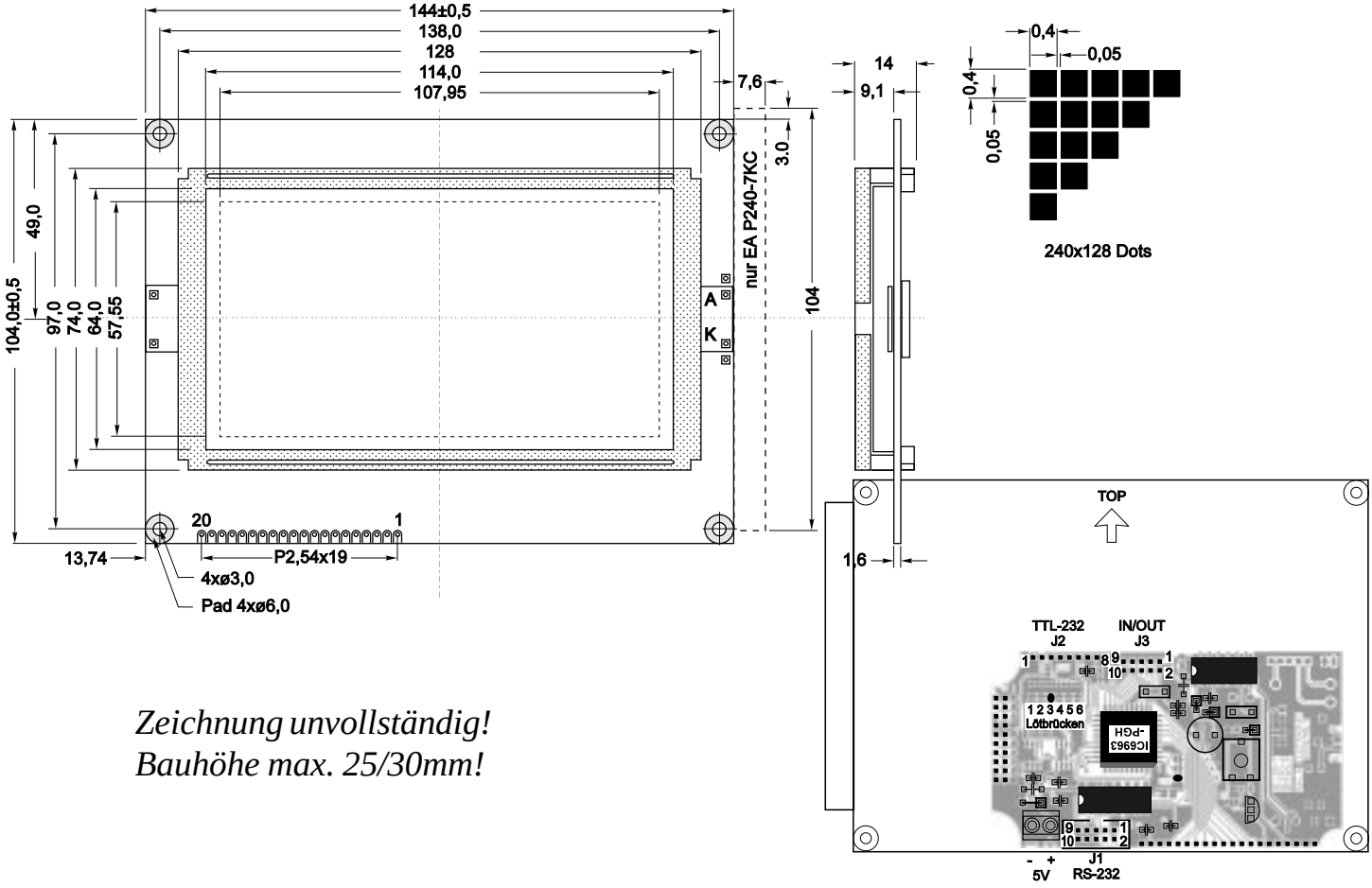


Ansicht von hinten

Zeichnung unvollständig!
Bauhöhe max. 25mm!

EA GE240-7KCV24
EA GE240-7KV24

240x128 MIT CFL-BEL., BLAU NEGATIV
240x128 MIT LED-BELEUCHTUNG GN/GB

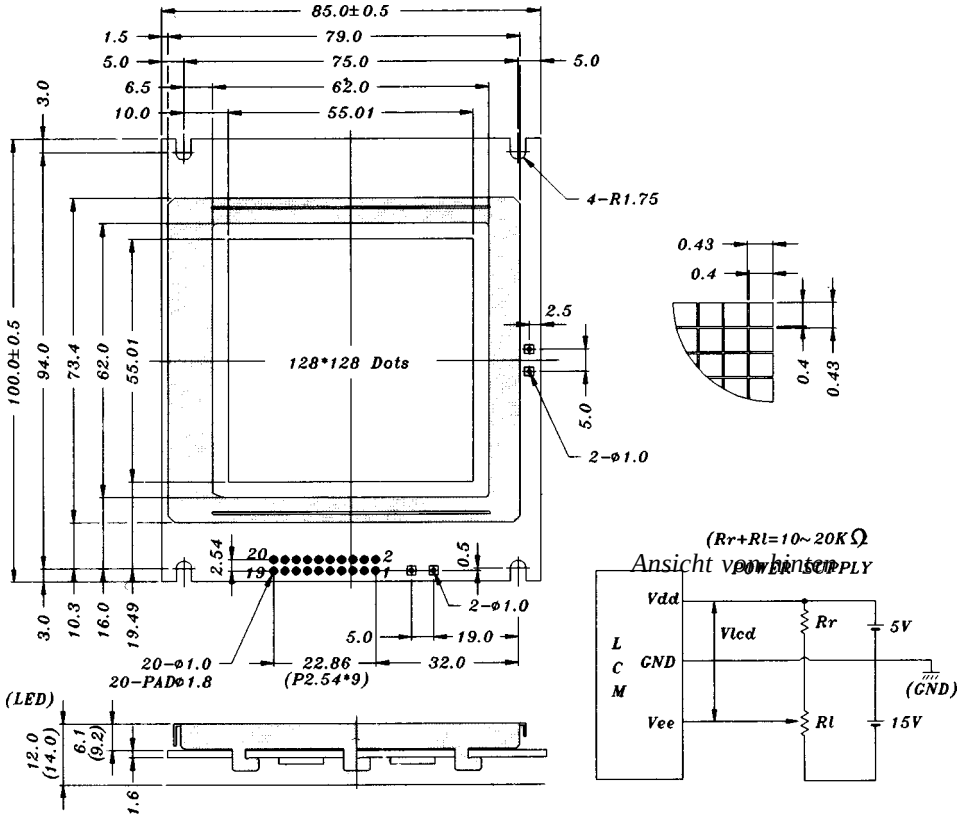


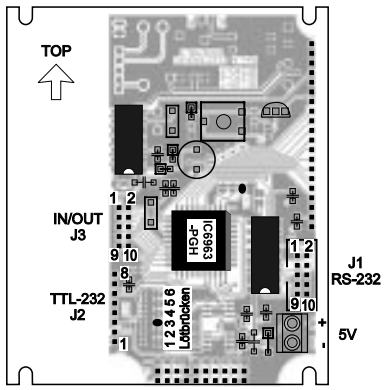
Zeichnung unvollständig!
Bauhöhe max. 25/30mm!

Ansicht von hinten

EA GE128-7KV24

128x128 MIT LED-BELEUCHTUNG GN/GB





Ansicht von hinten

Zeichnung unvollständig!
Bauhöhe max. 25mm!

GRAFIKEINHEITEN

BAUDRATEN

Die Baudrate läßt sich über die linken 3 Lötbrücken einstellen. Im Auslieferungszustand sind 9.600 Baud eingestellt. Bitte beachten Sie, daß der interne Datenpuffer lediglich 56 Byte umfaßt. Beim Senden größerer Datenmengen sollte unbedingt die Handshakeleitung RTS abgefragt werden (+10V Pegel: Daten können angenommen werden; -10V Pegel: Display ist Busy). Das Datenformat ist fest eingestellt auf 8 Datenbits, 1 Stopbit, keine Parität.

Baudraten			
Lötbrücken			Datenformat 8,N,1
1	2	3	
zu	zu	zu	1200
offen	zu	zu	2400
zu	offen	zu	4800
offen	offen	zu	9600
zu	zu	offen	19200
offen	zu	offen	38400
zu	offen	offen	57600
offen	offen	offen	115200

ADRESSIERUNG

Bis zu 4 Displays können an einer seriellen Schnittstelle adressiert betrieben werden. Die jeweilige Adresse wird über die Lötbrücken 4 und 5 eingestellt.

Adressen		
Lötbrücke		Adresse
4	5	
zu	zu	1
zu	offen	2
offen	zu	3
offen	offen	4

Achtung! Da beim einfachen Parallelschalten der Handshakeleitungen RTS bzw. der Sendeleitungen TxD zwei Ausgänge gegeneinander arbeiten würden, muß durch eine zusätzliche Hardware sichergestellt werden, daß es zu keinem Datencrash kommen kann. Sinnvoll ist z.B. eine Verknüpfung über ODER-Logik bei RTS bzw. über UND-Logik bei TxD.

DIP-SCHALTER EA OPT-DIP6

Die Einstellung der Baudrate und Adresse über DIP-Schalter schont nicht nur die Platine, sondern bietet diese Möglichkeit auch wenn kein Lötkolben in der Nähe ist. Natürlich läßt sich hierüber auch jederzeit der Testmode aktivieren und deaktivieren (DIP-Schalter Nr. 6). EA OPT-DIP6 ist als optional erhältlich (Bitte bei der Bestellung angeben).

PINBELEGUNG

Die Versorgungsspannung +5V wird über zwei Schraubklemmen eingespeist. An der Stiftleiste J1 werden die RS-232 Daten mit "echten" ±10V Pegeln erwartet. J2 ist für RS-232 Daten mit 5V Pegeln zum direkten Anschluß an einen µC konzipiert. Bei Verwendung von J2 müssen die Lötbrücken "C" und "R" geöffnet oder der Baustein 232 entfernt werden! Wird die Kontrastspannung V0 an J2 eingespeist, so muß die Lötbrücke von "232" auf "Ext" umgestellt werden.

Am Anschluß J3 stehen die 8 Ein-Ausgänge I/O1..8 zur Verfügung. Eine genauere Beschreibung der Ein- und Ausgänge finden Sie auf der Seite 13.

RS-232 Anschluß J1			
Pin	Symbol	In/Out	Funktion
1	VDD	-	+ 5V Versorgung
2	DCD	-	Brücke nach DTR
3	DSR	-	Brücke nach DTR
4	TxD	Out	Transmit Data
5	CTS	In	Clear To Send
6	RxD	In	Receive Data
7	RTS	Out	Request To Send
8	DTR	-	siehe Pin 2, Pin 3
9	V0	Out	ca. -9V Displayspg.
10	GND	-	0V Masse

Anschluß J2			
Pin	Symbo	In/Out	Funktion
1	GND	-	0V Masse
2	VDD	-	+ 5V Versorgung
3	V0	In	ca. -9V Displayspg.
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
8	RESET	In	Reset Display

Anschluß J3			
Pin	Symbol	In/Out	Funktion
1	VDD	-	+ 5V Versorgung
2..9	I/O0..7	In/Out	Ein-/Ausgang
10	GND	-	0V Masse

Achtung !
Handhabungs-vorschriften beachten

Elektrostatisch gefährdete Bauelemente

LOCHHAMER SCHLAG 17 · D-82166 GRÄFELFING
TELEFON 089/8541991 · TELEFAX 089/8541721

ELECTRONIC ASSEMBLY GMBH