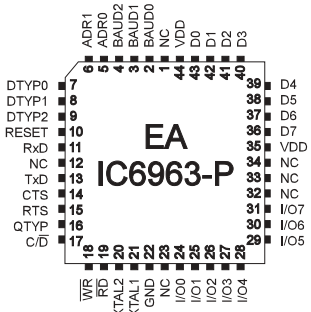
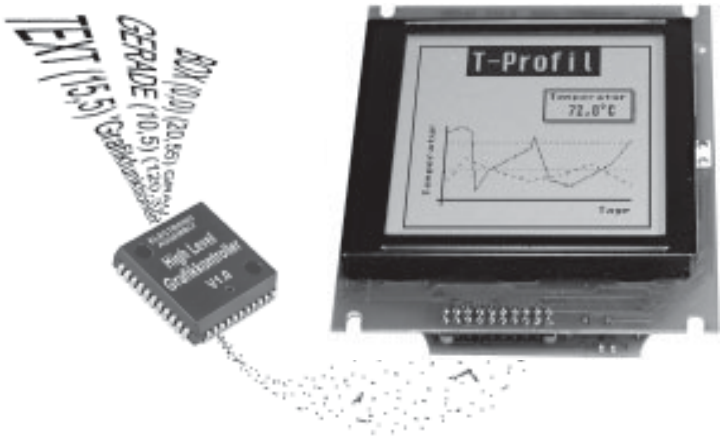
EA IC6963

HIGH-LEVEL GRAFIKKONTROLLER FÜR DISPLAYS MIT T6963



PLCC44J



TECHNISCHE DATEN

* FÜR LC GRAFIKDISPLAYS MIT T6963 z.B. 240x64, 240x128, 128x128 etc.
* KEINE TIMINGPROBLEME MEHR BEI SCHNELLEM BUSSYSTEM
* PIXELGENAUE POSITIONIERUNG BEI ALLEN FUNKTIONEN
* PROGRAMMIERUNG ÜBER HOCHSPRACHENÄHNLICHE BEFEHLE:
* GERADE, PUNKT, BEREICH, UND/ODER/EXOR, BARGRAPH...
* 3 VERSCHIEDENE FONTS INTEGRIERT
* ZOOM FUNKTION (2- BIS 8-FACH) ALLER FONTS
* BIS ZU 21 FREI DEFINIERBARE ZEICHEN
* TEXT UND GRAFIK MISCHEN
* 6 CLIPBOARD-FUNKTIONEN, CURSORFUNKTIONEN
* ANSTEUERUNG ÜBER RS-232 / CMOS-PEGEL
* DIREKTER ANSCHLUß VON MAX232 O.Ä. MÖGLICH
* BAUDRATEN 1200, 2400, 9600 BIS ZU 115200 BAUD
* BELASTET NICHT DAS PROZESSORSYSTEM
* NUR MAX. 4 EXTERNE BAUTEILE ERFORDERLICH
* 8 DIGITALE EIN-/AUSGÄNGE ZUR FREIEN VERWENDUNG

BESTELLBEZEICHNUNG

HIGH-LEVEL GRAFIKKONTROLLER FÜR LCD´S MIT T6963	**EA IC6963-PGH**
GRAFIKDISPLAY MIT 240x128 PIXEL, CFL-BELEUCHTUNG	**EA P240-7KC**
GRAFIKDISPLAY MIT 240x128 PIXEL, LED- BELEUCHTUNG	**EA P240-7KLED**
GRAFIKDISPLAY MIT 128x128 PIXEL, LED-BELEUCHTUNG	**EA P128-7KLED**
GRAFIKDISPLAY MIT 240x64 PIXEL, LED-BELEUCHTUNG	**EA P240-6K2LED**
KONVERTIERPROGRAMM FÜR BMP-BILDER (PC-DOS)	**EA DISKIC-1**

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

ALLGEMEINES

Der High-Level Grafikkontroller EA IC6963 versteht sich als Bindeglied zwischen Ihrem Prozessorsystem und dem Grafikdisplay. Die Ansteuerung erfolgt über eine serielle asynchrone Schnittstelle RS-232. Der Grafikkontroller 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:

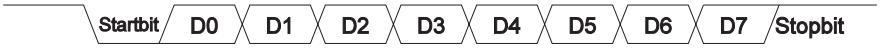
- keine Timingprobleme bei schnellem Prozessorbus
- keine Speicherplatzprobleme (Arbeitsspeicher und Speicher für den Zeichensatz v.a. bei µC)
- keine zeitaufwendigen Grafikberechnungen welche die Prozessorgeschwindigkeit belasten.

Auch die Hardwareanbindung ist denkbar einfach. Das Display und der Hauptprozessor lassen sich direkt anschließen. Es sind keine Treiber, Dekoder oder Portbausteine erforderlich. Im einfachsten Fall erfolgt die Displayansteuerung über nur 1 Leitung RxD. Lediglich 2 bis maximal 4 externe Bauteile sind erforderlich: ein Quarz mit 2 Kondensatoren und ein Reset-Kondensator. Arbeiten Sie mit einem 8051-kompatiblen System, dann benötigen Sie sogar **keine externen Bauteile** mehr. Der Takt und der Reset kann dann vom Hauptprozessor übernommen werden.

HARDWARE

Das System ist für +5V Betriebsspannung ausgelegt. Die Datenübertragung erfolgt seriell asynchron im RS-232 Format mit CMOS Pegeln. Das Übertragungsformat ist fest auf 8 Datenbits, 1 Stopbit, no Parity eingestellt. Die Baudrate kann über 3 Pins von 1200 Baud bis zu 115200 Baud ausgewählt werden. Handshakeleitungen RTS und CTS stehen zur Verfügung. Bei kleinen Datenmengen ist eine Auswertung nicht erforderlich (ein 56 Byte großer Datenpuffer ist integriert).

Datenformat:



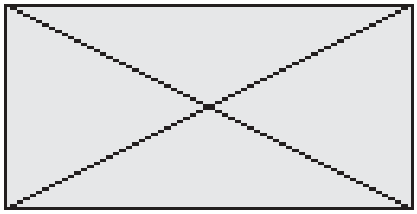
Zusätzlich sind 8 I/O-Ports zur freien Verwendung vorhanden. Diese können sowohl als Aus- als auch als Eingänge individuell geschaltet werden. Mögliche Anwendungen dafür ist das Schalten einer Hintergrundbeleuchtung oder das Einlesen von einer oder mehreren Tasten.

SOFTWARE

Die Programmierung des High-Level Grafikkontrollers 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 somit 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 bis zu 3 verschiedene Zeichensätze verwendet werden. Jeder Zeichensatz kann wiederum 2- bis 8-fach gezoomt werden. Mit dem größten Zeichensatz 8x16 lassen sich somit bei 8-fach Zoom (=64x128) bildschirmfüllende Worte und Zahlen darstellen.

TESTMODE

Solange der Pin 15 (RTS) nach dem Power-On oder Reset auf GND liegt, befindet sich der Grafikkontroller im Testmode. Auf dem angeschlossenen Display wird ein blinkendes Rechteck mit Kreuz dargestellt. Wird die Verbindung von Pin 15 (RTS) zu GND aufgehoben, dann kehrt der Grafikkontroller zum Normalbetrieb zurück.

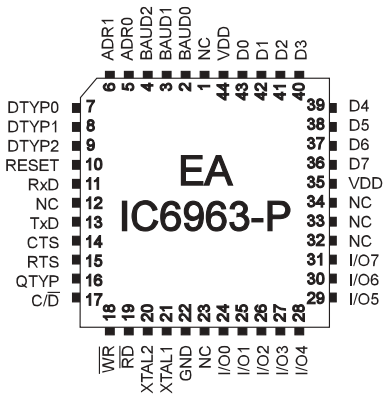


TECHNISCHE DATEN

Symbol	Parameter	Valid for	Condition	Min	Max	Units
VDD	Power Supply	VDD	11,059 MHz	4	6	V
ICC	Power Supply Current	VDD, Controller is busy	11,059 MHz		25	mA
ICC	Power Supply Current	VDD, Controller is ready	11,059 MHz		6.5	mA
VIL	Input Low Voltage	RESET, I/O0..7, Baud0..2, Adr0..1, RxD, CTS		-0.5	0.2*VDD-0.1	V
VIH	Input High Voltage	I/O0..7, Baud0..2, Adr0..1, RxD, CTS		0.2*VDD+0.9	VDD+0.5	V
VIHR	Input High Voltage Reset	RESET		0.7*VDD	VDD+0.5	V
VOL	Output Low Voltage	I/O0..7	IOL=3.2mA		0.45V	V
IIL	Logical 0 Input Current	Baud0..2, Adr0..1, RxD, CTS	VIN=0.45V		-50	µA
ITL	Logical 1 to 0 Transition Current	Baud0..2, Adr0..1, RxD, CTS	VIN=2V		-650	µA
ILI	Input Leakage Current	I/O0..7	0.45<VIN<VD		±10	µA
CIO	Pin Capacitance	RESET, I/O0..7, Baud0..2, Adr0..1, RxD, CTS	1 MHz, 25°C		10	pF
IOL	Output Low Current	I/O0..7	per line		10	mA
IOP	Output Low Current	I/O	port		26	mA
TRSTH	RESET Pulse Width	RESET		10		ms
RRST	RESET Pull Down Resistor	RESET		50	300	kOhm
TOP	Operating Temperature			0	+70	°C
FOSC	Oscillator Frequency	XTAL1, XTAL2		0	20	MHz

Werte gelten wenn nicht anders angegeben für T_a = 0..+70°C und VDD= 5,0V ±20%.

PINBELEGUNG



Pin Beschreibung				
Pin	**Bezeichnung**	**In/Out**	**Pegel**	**Beschreibung**
1	NC			nicht beschalten!
2,3,4	BAUD0..2	In	lo	Baudrateneinstellung
5,6	ADR0..1	In	lo	Adresseinstellung
7,8,9	DTYP	In	lo	Einstellung Displaytyp
10	RESET	In	hi	setzt Kontroller in den Ausgangszustand
11	RxD	In	lo	RS-232 Empfangsleitung
12	NC			nicht beschalten!
13	TxD	Out	lo	RS-232 Sendeleitung
14	CTS	In	lo	lo: evtl. anstehende RS-232 Daten werden gesendet; hi: evtl. anstehende RS-232 Daten werden unterdrückt
15	RTS	Out	lo	lo: zeigt an, daß RS-232 Daten empfangen werden können; hi: es können keine RS-232 Daten angenommen werden
16	QTYP	In	lo	Quarztyp: lo=11,0592 MHz; hi=18,432 MHz Quarz
17	C/D	Out	hi/lo	Display: hi = Befehle; lo = Daten
18	WR	Out	lo	Display: Daten/Befehle schreiben
19	RD	Out	lo	Display: Daten/Befehle lesen
20	XTAL2	Out		Systemoszillator 11,0592 oder 18,432 MHz (PIN16:QTYP)
21	XTAL1	In		Systemoszillator bzw. Einspeisung ext. Systemtakt
22	GND	GND	lo	Versorgungsspannung 0V
23	NC			nicht beschalten!
24..31	IO0..7	I/O	hi/lo	8 Ein-/Ausgänge zur freien Verwendung
32,33,34	NC			nicht beschalten!
35	VDD	VDD	hi	Versorgungsspannung +5V
36..43	D7..0	I/O	hi/lo	Display: 8 Datenleitungen
44	VDD	VDD	hi	Versorgungsspannung +5V

BAUDRATEN

Je nach verwendetem Systemtakt (Quarz, Keramikschwinger) können diverse Baudraten für die RS-232 Datenübertragung eingestellt werden. Das erfolgt durch Verbinden der Pins BAUD0..2 und QTYP mit VDD oder GND-Pegel. Die dadurch programmierten Baudraten entnehmen Sie bitte der Tabelle nebenan (0:GND, 1:VDD).

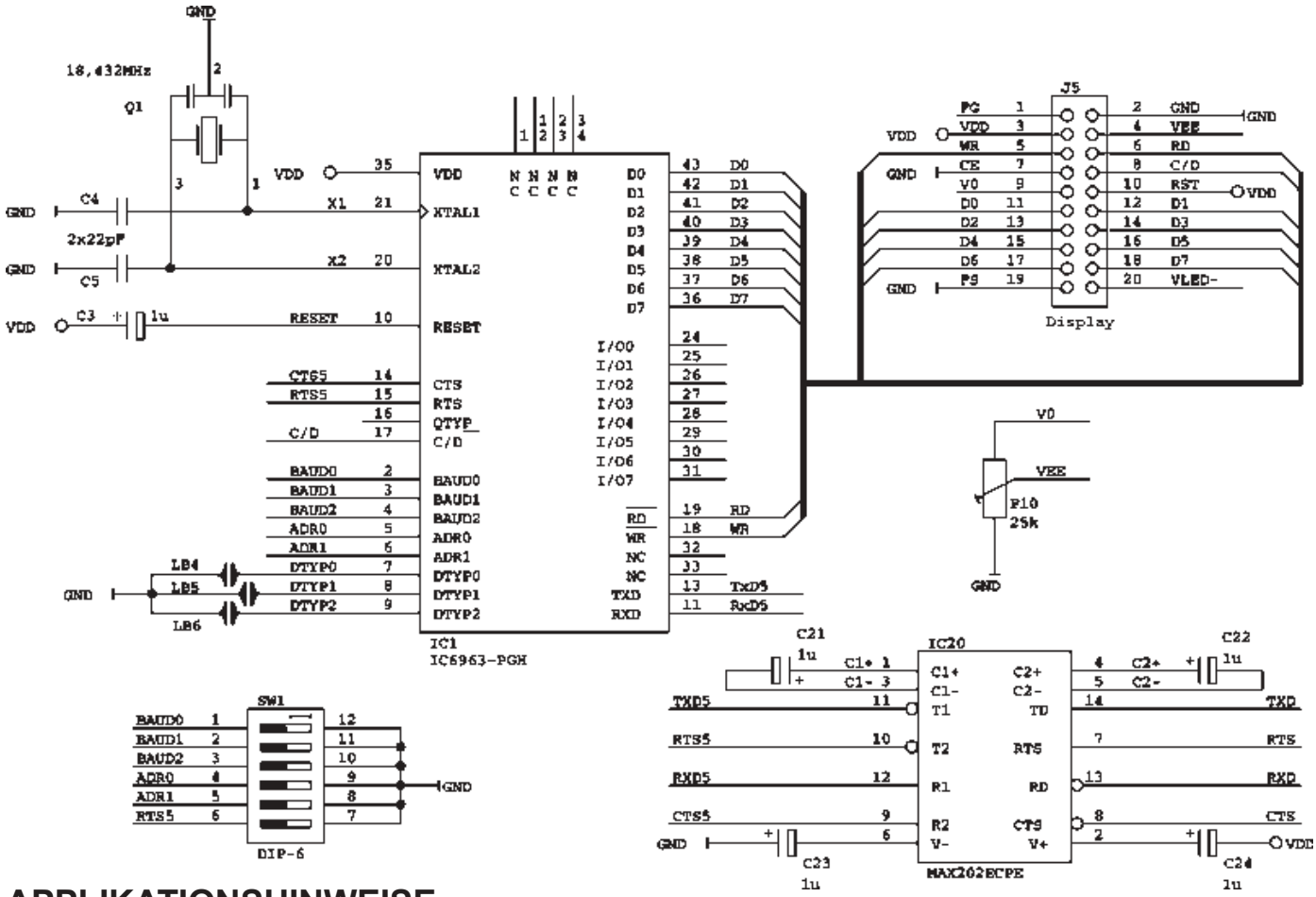
Baudratentabelle				
Baud 2	Baud 1	Baud 0	QTYP = 0 11,0592 MHz	QTYP = 1 18,432 MHz
0	0	0	1200	1200
0	0	1	2400	2400
0	1	0	4800	4800
0	1	1	9600	9600
1	0	0	19200	19200
1	0	1	38400	38400
1	1	0	57600	57600
1	1	1	115200	115200

DISPLAYTYPEN

Durch Verbinden der Pins DTYP0..2 mit VDD oder GND-Pegel können 7 Standard Displaytypen eingestellt werden (0: GND, 1: VDD). Mögliche Displays von ELECTRONIC ASSEMBLY entnehmen Sie bitte der Tabelle nebenan. Andere Displaygrößen lassen sich mit dem Befehl '!' (siehe Seite 15) einstellen.

Dtyp 2	Dtyp 1	Dtyp 0	Auflösung	Display z.B.
0	0	0	128 x 64	
0	0	1	128 x 112	EA 7128-6,8KEL
0	1	0	128 x 128	EA P128-7KLED
0	1	1	160 x 128	EA 7160-7KLED
1	0	0	240 x 40	EA VK-5343LED
1	0	1	240 x 64	EA P240-6K2LED
1	1	0	240 x 128	EA P240-7KLED

APPLIKATIONSBEISPIEL



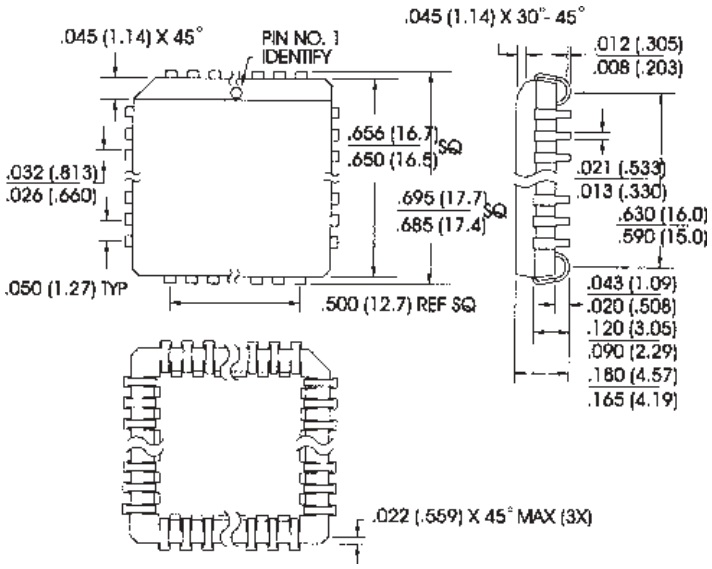
APPLIKATIONSHINWEISE

Um eine störungsfreien Betrieb zu gewährleisten sollten bei der Leiterplattenentflechtung folgende Regeln beachtet werden:

- Achten Sie auf eine saubere Masseführung in Ihrem Layout (keine Masseschleifen)
- Die Versorgungsspannung sollte über verbreiterte Leiterbahnen sternförmig verteilt werden. Am besten ist natürlich eine Platine mit speziellen Versorgungslayern.
- Bauteile bzw. Baugruppen mit erhöhter oder stark schwankender Stromaufnahme benötigen völlig eigene Versorgungsleitungen. Diese sollten vom Rest der Elektronik entkoppelt sein (Filter verwenden). Auch die LED-Beleuchtung des Displays sollte separat versorgt werden.
- Sehen Sie Blockkondensatoren an allen aktiven Bauteilen vor.
- Leitungen mit hochfrequenten Signalen bzw. steilen Flanken so kurz wie möglich halten (XTAL1 und XTAL2 !)

ABMESSUNGEN EA IC6963-PGH

Gehäuse: PLCC44J; alle Maße in Inch (mm)



GRUNDEINSTELLUNGEN

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

Achtung !

Handhabungs-vorschriften beachten

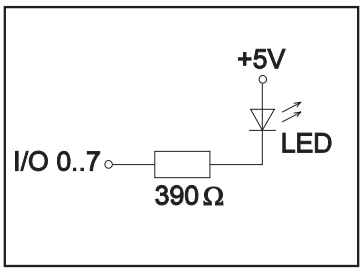
Elektrostatisch gefährdete Bauelemente

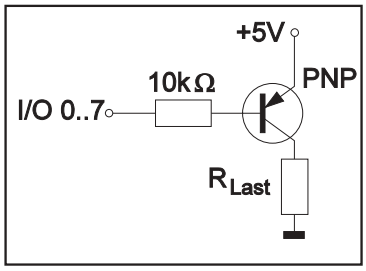
DIGITALE EIN-/ AUSGÄNGE IO 0..7

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.

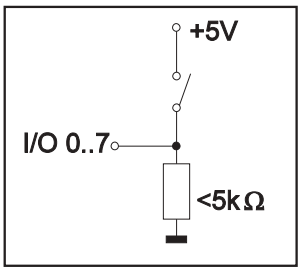
Beschaltung als Ausgang

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.

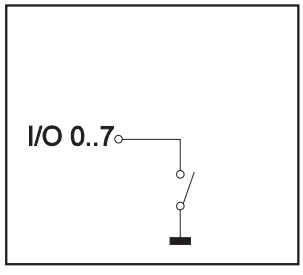




Beschaltung als Eingang



Am Eingang dürfen Spannungspegel zwischen -0,5V und +0,2V*VDD-0,1V anliegen. Der Leckstrom beträgt max. ±10µA. Die Schaltschwellen entnehmen Sie bitte den technischen Daten auf der Seite 3. 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 auf der Seite 15

INTEGRIERTE FONTS

Im High-Level Grafikkontroller EA IC6963 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	

Font 2: 6x8

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

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

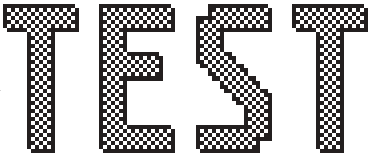
TIP: SCHRIFTEFFEKTE

Mit dem Befehl 'T' TEXT-Modus (Verknüpfung, Muster) können interesannte 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

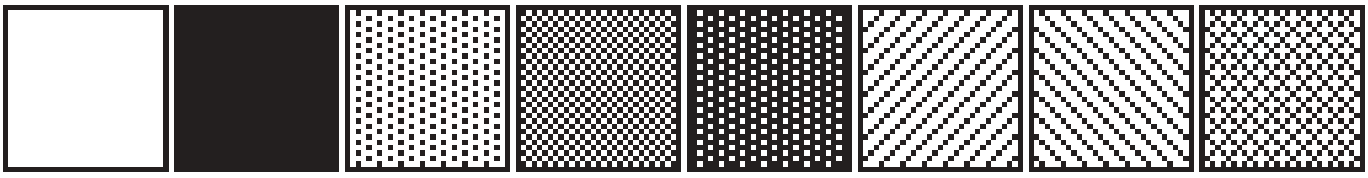
In der nachfolgenden Tabelle ist ein Beispiel zu sehen welches die Zeichenkette "Test" an den Koordinaten 7,3 ausgibt.

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:



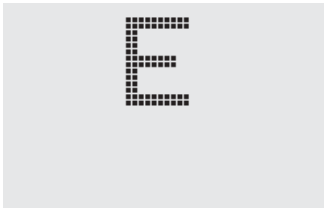
mst=0 weiss	mst=1 schwarz	mst=2 25% grau	mst=3 50% grau	mst=4 75% grau	mst=5 45° rechts	mst=6 45° links	mst=7 45° kreuz

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

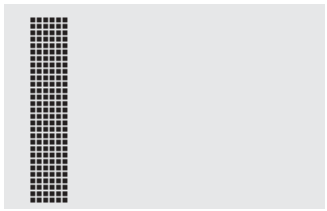
Zeichen 'E' wird an Koordinate 19,2 ausgegeben.

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!*

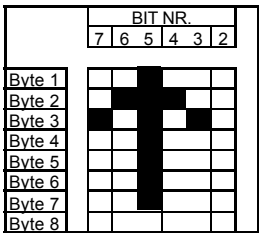
E n1 daten — Zeichen definieren

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) !

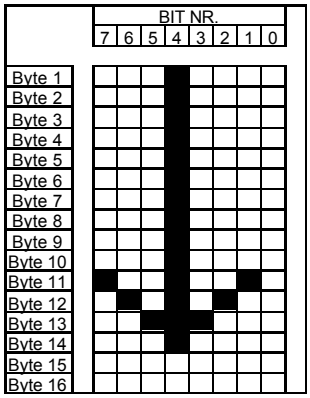
Selbstdefinierbare Zeichen (Code)

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

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.



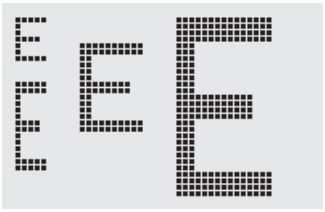
Beispiel 2:
$45 $02
$10 $10 $10 $10 $10 $10 $10 $10 $10 $10 $92 $54 $38 $10 $00 $00
definiert einen Pfeil nach unten für ASCII-Nr. 2, bei eingestelltem 8x16 Zeichensatz.



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.

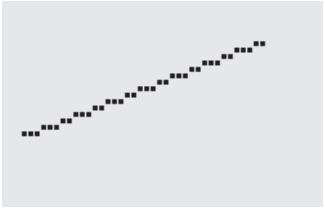
Beispiel: $46 $02 $03 $04
ab sofort ist der 6x8- Font mit 3-facher Breite und 4-facher Höhe eingestellt.
Im Bild nebenan ist das Zeichen 'E' aus dem 6x8 Font mit unterschiedlichen Vergrösserungen dargestellt.



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

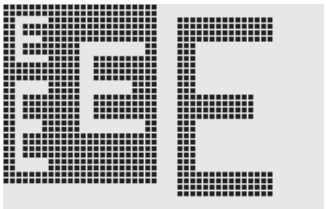
Der Bereich von der linken oberen Ecke **x1,y1** bis zu rechten unteren Ecke **x2,y2** wird angefordert. Der Grafikchip sendet daraufhin sofort die Breite und Höhe des Bildausschnittes und danach die Bilddaten. Zum Aufbau der Bilddaten siehe den Befehl Bild Upload 'U'.

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.

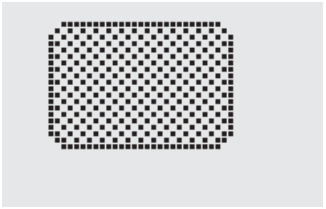
I x1 y1 x2 y2 — Bereich invertieren

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.

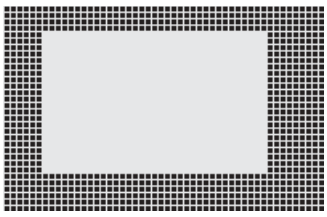
K S/D n1 Grafikkontroller (de)selektieren

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

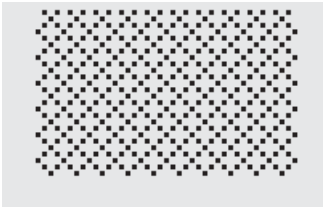


Der Bereich von der linken oberen Ecke **x1,y1** bis zur rechten unteren Ecke **x2,y2** wird gelöscht.

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

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

M x1 y1 x2 y2 mst — Bereich mit Füllmuster

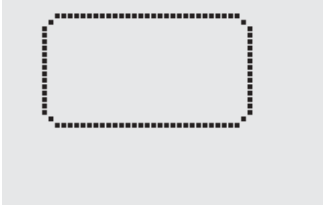


Ein rechteckiger Bereich wird von der linken oberen Ecke **x1,y1** bis zur rechten unteren Ecke **x2,y2** mit dem Muster **mst** unter Beachtung des eingestellten Grafikmodus 'V' (setzen / löschen / invertieren / replace / invers replace) gezeichnet.

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

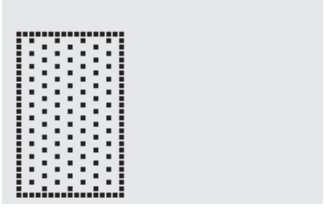


Ein Rechteck mit abgerundeten Ecken wird von der linken oberen Ecke **x1,y1** bis zur rechten unteren Ecke **x2,y2** unter Beachtung des eingestellten Grafikmodus 'V' (setzen / löschen / invers) gezeichnet. Der Inhalt des Rundecks wird nicht verändert. Vergleiche 'J' Rundbox zeichnen.

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 $0D $11

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!

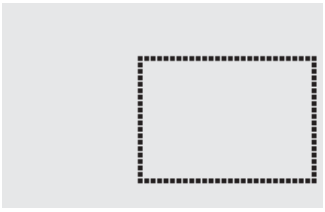
Beispiel: $51 $43 $00

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.

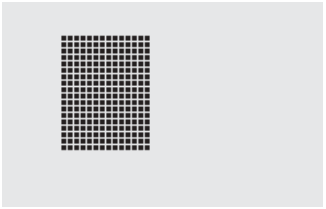


**) Der Befehl Q C mst benötigt ein Display-RAM mit mindestens 8kB. Bei Displays mit kleinerem RAM (z.B. 2kB) kann dieser Befehl nicht verwendet werden!*

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.

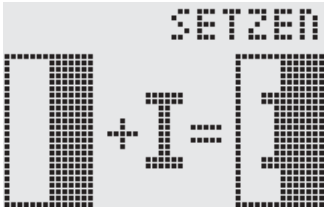


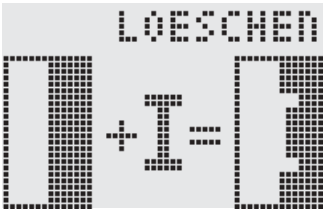
T L/R/O/U n1 mst — Text-Modus einstellen

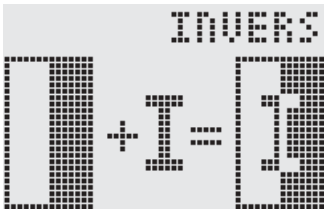
Der Verknüpfungsmodus **n1** und das Muster **mst** wird für Textfunktionen ASCII-Zeichen setzen 'A' und Zeichenkette ausgeben 'Z' eingestellt. Für den Befehl Zeichenkette ausgeben 'Z' wird außerdem die Schreibrichtung angegeben: **L**=links, **R**=rechts, **O**=oben und **U**=unten.

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

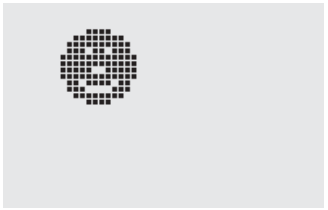
U x1 y1 daten — Bild Upload

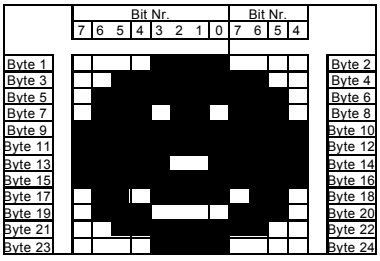
Ein Bild wird an die Koordinate **x1,y1** geladen.

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.

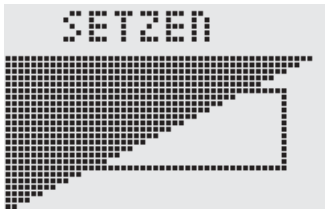
Beispiel: $55 $09 $04 $0C $0C
$0F $00 $3F $C0 $7F $E0 $76 $E0 $FF $F0 $FF $F0
$F1 $F0 $FF $F0 $6F $60 $70 $E0 $3F $C0 $0F $00

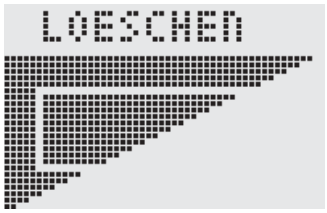
lädt das nebenstehende Bild an die Koordinate 9,4.





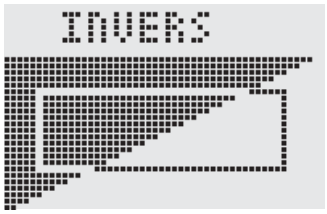
V n1 — Grafik-Modus einstellen





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.

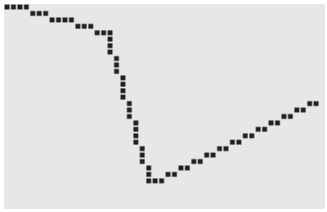


Als Beispiel wird nebenan ein Rechteck mit den Verknüpfungsmodi setzen, löschen und invers auf einen vorhandenem Hintergrund gezeichnet.

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 5.

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 6.

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

EA IC6963

! n1 n2 adrLO adrHI — Display einstellen

Mit diesem Befehl kann auch ein Display eingestellt werden welches nicht über DTYP0..2 (Seite 5) programmierbar ist. Mit **n1** wird die Breite des Displays und mit **n2** die Höhe eingestellt. Gültige Werte sind für **n1** sind 64..256 (bei 256 Pixeln Breite programmieren Sie n1=0) und für **n2** 16..128. Außerdem kann die Bildstartadresse **adrLO** und **adrHI** (16-Bit) eingestellt werden (normalerweise immer $0000).

Beispiel: $21 $64 $32 $00 $00

Ein Display mit 100 Pixel in der Breite und 50 Pixel Höhe wird eingestellt

? — Displaytyp abfragen

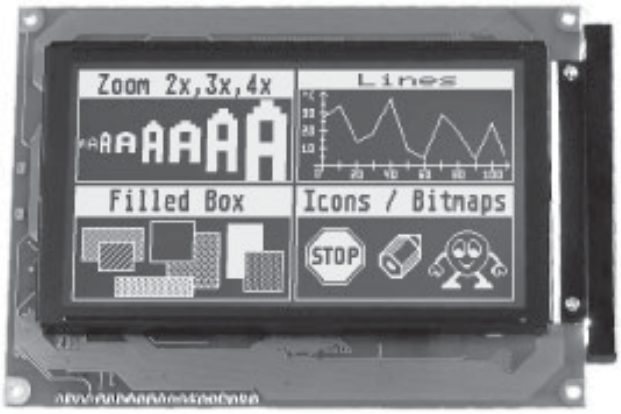
Die Auflösung des Displays und die Art des Bildaufbaus wird abgefragt.

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.

FERTIG AUFGEBAUTE GRAFIKEINHEITEN MIT IC6963
240x128 - 128x128 - 240x64

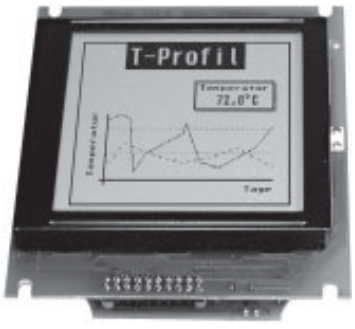
EA GE240-6KV24



EA GE240-7KCV24

TECHNISCHE DATEN

* 240x128 PIXEL MIT CFL-BEL. BLAU NEGATIV
* 240x128 PIXEL MIT LED-BELEUCHTUNG GN/GB
* 128x128 PIXEL MIT LED-BELEUCHTUNG GN/GB
* 240x64 PIXEL MIT LED-BELEUCHTUNG GN/GB
* 3 FONTS (ZOOM) VON ca. 2mm ÜBER ca. 5mm BIS ZU ca. 50mm
* VERSORGUNGSSPANNUNG: +5V / 500..1000mA
* NEGATIVE SPANNUNG FÜR KONTRAST WIRD ON-BOARD ERZEUGT
* CFL-WANDLER ON-BOARD (EA GE240-7KCV24)
* "ECHTE" RS-232 PEGEL ±10V
* RS-232 BAUDRATEN 1200..115200 BD



EA GE128-7KV24

ZUBEHÖR

* KABEL FÜR ANSCHLUß AN 9-POL. SUB-D (FEMALE): **EA KV24-9B**
* DISKETTE MIT BEISPIELPROGRAMMEN FÜR PC-DOS: **EA DISKIC-1**

BESTELLBEZEICHNUNG

GRAFIKEINHEIT 240x128 MIT CFL-BELEUCHTUNG, 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**

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

ELECTRONIC ASSEMBLY GMBH