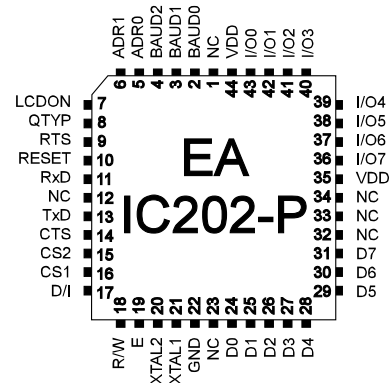


# HIGH-LEVEL GRAFIKKONTROLLER FÜR DISPLAYS MIT HD61202 (KS0108B)



PLCC44J

## TECHNISCHE DATEN

- \* FÜR LC GRAFIKDISPLAYS 128x64 MIT HD 61202 ODER KOMPATIBEL
- \* KEINE TIMINGPROBLEME BEI SCHNELLEM BUSSYSTEM
- \* PROGRAMMIERUNG ÜBER DIVERSE EINGebaUTE GRAFIKFUNKTIONEN:
- \* GERADE, PUNKT, BEREICH, UND/ODER/EXOR, BARGRAPH, FÜLLMUSTER...
- \* 3 VERSCHIEDENE FONTS INTEGRIERT
- \* ZOOM FUNKTION ALLER FONTS (2-, 3- UND 4-FACH)
- \* 4-16 FREI DEFINIERBARE ZEICHEN (JE NACH GRÖßE)
- \* TEXT UND GRAFIK MISCHEN
- \* ANSTEUERUNG ÜBER RS-232 / CMOS-PEGEL
- \* DIREKTER ANSCHLUß VON MAX232 O.Ä. MÖGLICH
- \* BAUDRATE PROGRAMMIERBAR VON 300 BIS 115.200 BAUD
- \* NUR MAX. 4 EXTERNE BAUTEILE ERFORDERLICH
- \* 8 DIGITALE I/O'S ZUR FREIEN VERWENDUNG
- \* 5V / typ. 12mA IM NORMALBETRIEB (BEI 11,059 MHz)
- \* POWER-SAVE FUNKTION: STROMVERBRAUCH max. 100µA / 5V
- \* HARDWARE CODIERUNG VON BIS ZU 4 ADRESSEN

## OPTIONEN (IN VORBEREITUNG)

- \* VERSION FÜR VERTIKALEN DISPLAYEINBAU (90° DREHUNG)
- \* VERSION IM 40-POL. DIL GEHÄUSE

## BESTELLBEZEICHNUNG

HIGH-LEVEL GRAFIKKONTROLLER 128x64 FÜR HD61202	<b>EA IC202-PGH</b>
PASSENDEN GRAFIKDISPLAY MIT HD61202, 128x64	<b>EA P128-6N3LED</b>
KERAMIKRESONATOR SMD 7,37MHz, 3 PINS INKL. C's	<b>EA KERS7M37-C</b>
GRAFIKEINHEIT MIT IC202-PGH UND RS-232C	<b>EA GE128-6N3V24</b>

### ALLGEMEINES

Der High-Level Grafikkontroller EA IC202 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 verschiedenste Schriftgrößen.

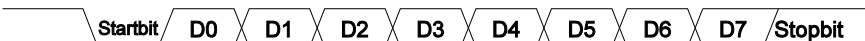
Die Programmierung erfolgt über hochsprachenähnliche Grafikbefehle; die zeitraubende Programmierung von Zeichensätzen und Grafikroutinen entfällt hier völlig. Doch nicht nur der Entwicklungsaufwand reduziert sich drastisch. Auch in der Serie sind die folgende Vorteile spürbar:

- keine Timingprobleme bei schnellem Prozessorbus
- keine Speicherplatzprobleme (Arbeitsspeicher und Speicher für den Zeichensatz v.a. bei  $\mu\text{C}$ )
- 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 150 Baud bis zu 115200 Baud ausgewählt werden. Handshakeleitungen RTS und CTS stehen zur Verfügung. Bei kleinen Datenmengen ist eine Auswertung nicht erforderlich.

Datenformat: 

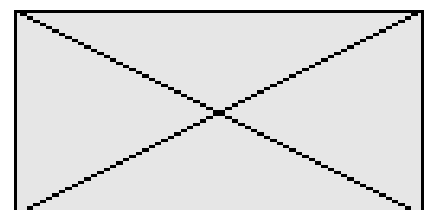
Zusätzlich sind 8 I/O-Ports zur freien Verwendung vorhanden. Diese können sowohl als Aus- als auch als Eingänge individuell geschaltet werden. Mögliche Anwendungen dafür, ist das Schalten einer Hintergrundbeleuchtung oder das Einlesen von einer oder mehreren Tasten.

### 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 3 verschiedene Zeichensätze verwendet werden. Jeder Zeichensatz kann wiederum 2-, 3- und 4-fach gezoomt werden. Mit dem größten Zeichensatz 16x8 lassen sich somit bei 4-fach Zoom (=64x32) bildschirmfüllende Worte und Zahlen darstellen.

### TESTMODE

Solange man den Pin 9 (RTS) nach dem Power-On oder Reset auf GND legt, befindet sich der Grafikkontroller im Testmode. Auf dem angeschlossenen Display wird ein blinkendes Rechteck mit Kreuz dargestellt. Wird die Verbindung von Pin 9 (RTS) zu GND aufgehoben dann kehrt der Grafikkontroller zum Normalbetrieb zurück. Das Testbild ist aber immer noch zu sehen.



### 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
ICPS	Power Supply Current Power-Save-Mode	VDD	VDD=6V		100	µA
			VDD=3V		40	µA
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+	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  $T_a = 0..+70^\circ\text{C}$  und  $VDD = 5,0\text{V} \pm 20\%$ .

### POWER-SAVE FUNKTION

Die Power-Save Funktion ermöglicht den Einsatz in batteriebetriebenen Geräten. Mit dem Befehl "Q \$01" läßt sich der High-Level-Grafikkontroller vom Normalbetrieb in den Power-Save Modus schalten. Der Stromverbrauch ohne Display reduziert sich dabei auf ca. 100µA. Da das Display weiterhin mit Spannung versorgt wird, fließen dort je nach Displaytyp, einige mA Strom. Der Displayinhalt, alle selbstdefinierten Zeichen und die Bargraphdefinitionen bleiben dadurch erhalten. Im Power-Save-Modus können keine Befehle angenommen und ausgeführt werden (RTS liegt auf H-Pegel).

Mit dem Befehl "Q \$02" wird sowohl der Kontroller, als auch das über einen zusätzlichen Transistor angeschlossene Display (siehe Applikation Seite 5) in den Power-Save Modus geschaltet. Hierbei geht der Displayinhalt jedoch verloren, alle selbstdefinierten Zeichen und die Bargraphdefinitionen bleiben aber erhalten. Auch in diesem Modus können keine Befehle angenommen und ausgeführt werden (RTS liegt auf H-Pegel).

Durch einen mindestens 10ms langen Resetimpuls (H-Pegel) wird der Grafikkontroller wieder in den Normalbetrieb geschaltet. Die internen Register werden dabei, wie in der Tabelle beschrieben, gesetzt.

Grundeinstellungen			
Register	Befehl	nach Power-On	nach Power-Save-Modus
Display Control	C	Display ein	Display ein
Text-Modus	T	setzen	setzen
Grafik-Modus	V	setzen	setzen
Font	F	6x8	6x8
Fontfaktor Breite/Höhe	F	1/1	1/1
Last xy	W	(0;0)	(0;0)
Frei definierbare Zeichen	E	undefiniert	unverändert
Bargraph 1..8	B	undefiniert	unverändert
High-Level Grafikkontroller	K	selektiert	selektiert
Ein-/ Ausgänge I/O0..7	Y	H-Pegel	H-Pegel

## PINBELEGUNG

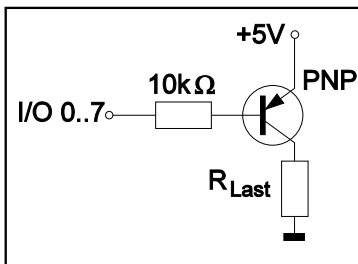
Pin Beschreibung				
Pin	Bezeichnung	In / Out	Pegel	Beschreibung
1	NC			nicht beschalten!
2,3,4	BAUD0..2	In	lo	Baudrateneinstellung
5,6	ADR0, ADR1	In	lo	Adresseinstellung
7	LCDON	Out	hi	lo: Power-Save-Modus, hi: Normalbetrieb
8	QTyp	In	lo	Quarzystyp siehe Baudrateneinstellung
9	RTS	Out	lo	lo: zeigt an, wenn RS-232 Daten empfangen werden können; hi: es können keine RS-232 Daten angenommen werden
10	RESET	In	hi	setzt Controller und das Display 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
15,16	CS2, CS1	Out	hi	Display: Auswahl linke (CS1) oder rechte (CS2) Displayhälfte
17	D/I	Out		Display: hi: Daten; lo: Befehle
18	R/W	Out		Display: hi: Daten/Befehle lesen; lo: Daten/Befehle schreiben
19	E	Out	hi	Display: Enable
20	XTAL2	Out		Systemoszillator
21	XTAL1	In		Systemoszillator bzw. Einspeisung ext. Systemtakt
22	GND	GND	lo	Versorgungsspannung 0V
23	NC			nicht beschalten!
24,25,26,27, 28,29,30,31	D0..7	I/O		Display: 8 Datenleitungen
32,33,34	NC			nicht beschalten!
35	VDD	VDD	hi	Versorgungsspannung +5V
36,37,38,39, 40,41,42,43	IO7..0	I/O		8 Ein-/Ausgänge zur freien Verwendung
44	VDD	VDD	hi	Versorgungsspannung +5V

## DIGITALE EIN-/ AUSGÄNGE IO 0..7

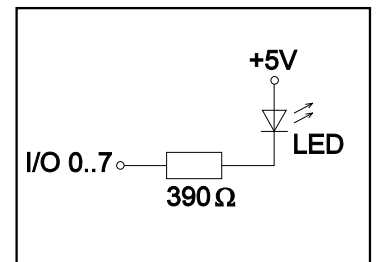
8 Pins am High-Level Grafikkontroller können als frei programmierbare Ein- und Ausgänge verwendet werden. Auch ein gemischter Betrieb von z.B. 3 Ausgängen und 5 Eingängen ist möglich.

### Beschaltung als Ausgang

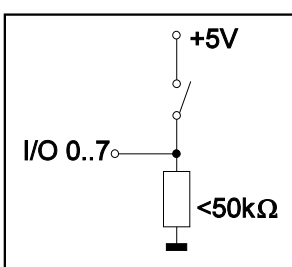
Mit dem Befehl "Y n1 n2"<sup>1)</sup> kann jeder Pin IO 0..7 auf H- oder L-Pegel geschaltet werden; er ist damit einem Open-Drain Ausgang vergleichbar,



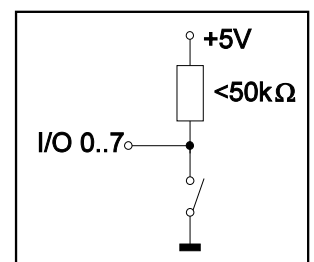
welcher keinen internen Pull-up Widerstand besitzt. Strom kann also nur bei L-Pegel fließen. Jeder Pin kann max. 10mA liefern, alle Pins zusammen dürfen mit nicht mehr als 26mA belastet werden (z.B. 2x10mA und 1x6mA). Es ist somit möglich, mit einem Ausgang direkt eine LED zu schalten. Größere Ströme können durch Verwendung eines externen Transistors geschaltet werden. Nach dem Power-On bzw. Power-Save-Mode liegen alle Ausgänge auf H-Pegel.



### Beschaltung als Eingang

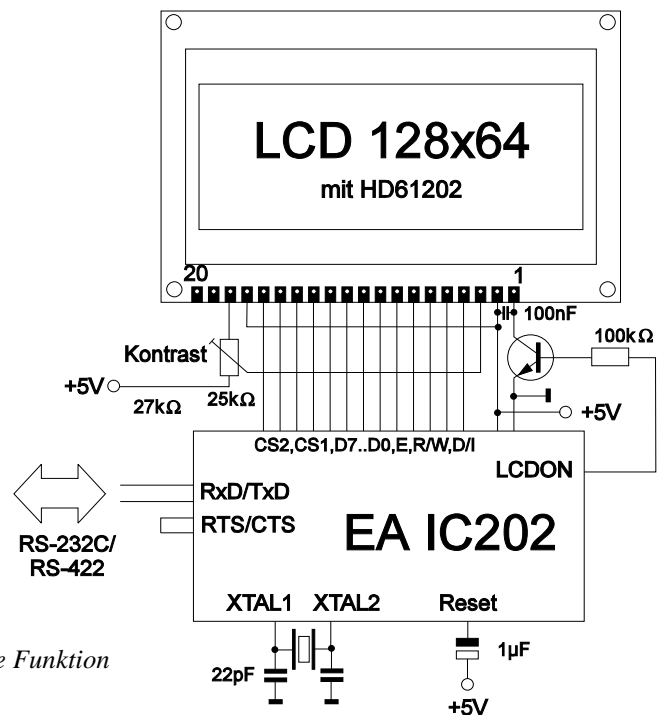
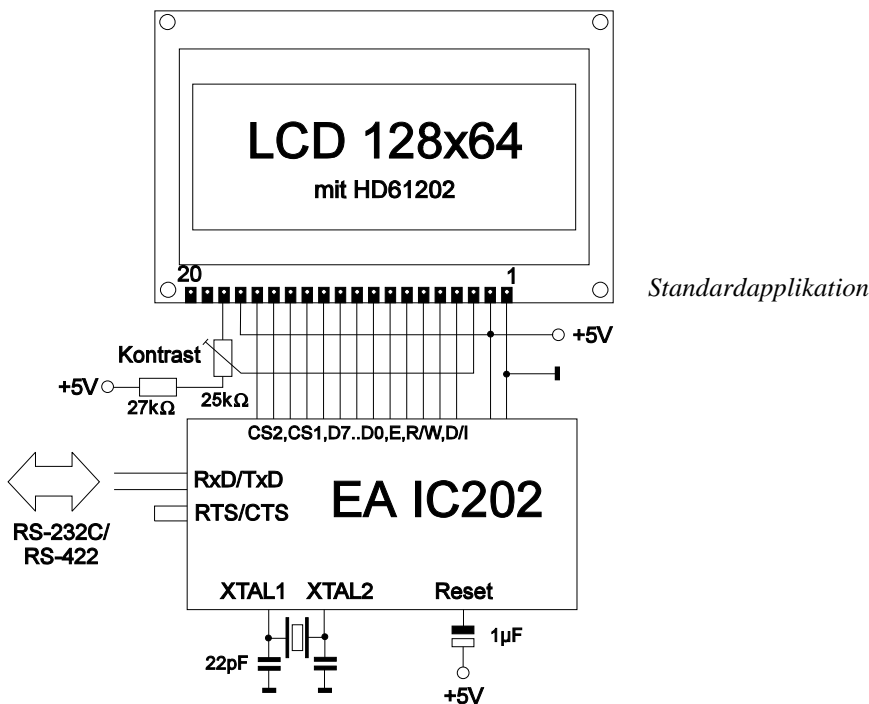


Am Eingang dürfen Spannungspegel zwischen -0,5V und +0,2V\*VDD-0,1V anliegen. Der Leckstrom beträgt max. ±10µA. Die Schaltschwellen entnehmen Sie bitte den technischen Daten auf der Seite 3. Mit dem Befehl "X n1"<sup>1)</sup> kann jeder Pin IO 0..7 eingelesen werden. Der Spannungspegel muß während des gesamten Einlesevorgangs stabil sein. Eine Entprellfunktion ist nicht eingebaut.



<sup>1)</sup>eine Befehlsbeschreibung finden Sie auf der Seite 13

### APPLIKATIONSBEISPIELE



### APPLIKATIONSHINWEISE

Um einen störungsfreien Betrieb zu gewährleisten, sollten bei der Leiterplattenentflechtung folgende in der Digitalelektronik übliche 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 !)



## ALLE GRAFIKFUNKTIONEN AUF EINEN BLICK

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

Befehlstabelle EA IC202											
Befehl						Anmerkung					
<b>Funktionen zur Textausgabe</b>											
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: Verknüpfungsmodus 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..4); n2=X-Faktor; n3=Y-Faktor;					
ASCII-Zeichen setzen	A		x1	y1	n1	Das Zeichen n1 wird an Koordinate x1,y1 gesetzt. (Bezug links oben)					
Zeichenkette ausgeben	Z		x1	y1	...	NUL	Eine Zeichenkette (...) an x1,y1 ausgeben; Zeichen 'NUL' (\$00)=Ende				
Zeichen definieren	E		n1	daten ...			n1=Zeichen Nr.; daten=Anzahl Bytes je nach akt. Font				
<b>Grafik-Befehle mit Verknüpfungsmodus</b>											
Grafik-Modus	V		n1				n1: 1=setzen; 2=löschen; 3=invers; 4=Replace; 5=Invers Replace;				
Punkt setzen	P		x1	y1			Ein Pixel an die Koordinaten x1, y1 setzen				
Gerade zeichnen	G		x1	y1	x2	y2	Eine Gerade von x1,y1 nach x2,y2 zeichnen				
Gerade weiter zeichnen	W		x1	y1			Eine Gerade vom letzten Endpunkt bis x1, y1 zeichnen				
Rechteck zeichnen	R		x1	y1	x2	y2	Ein Rechteck zeichnen; x1,y1,x2,y2 = Gegenüberliegende Eckpunkte				
Rundreck 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			
<b>sonstige Grafik-Befehle</b>											
Display löschen	D	L					Gesamten Displayinhalt löschen (auf weiß setzen);				
Display invertieren	D	I					Gesamten Displayinhalt invertieren;				
Display füllen	D	S					Gesamten Displayinhalt füllen; (auf schwarz setzen);				
Bereich löschen	L		x1	y1	x2	y2	Einen Bereich löschen; x1,y1,x2,y2 = Gegenüberliegende Eckpunkte				
Bereich invertieren	I		x1	y1	x2	y2	Einen Bereich invertieren; x1,y1,x2,y2 = Gegenüberliegende Eckpunkte				
Bereich füllen	S		x1	y1	x2	y2	Einen Bereich füllen; x1,y1,x2,y2 = Gegenüberliegende Eckpunkte				
Box zeichnen	O		x1	y1	x2	y2	mst	Ein Rechteck mit Füllmuster mst (0..7) zeichnen; (immer Replace)			
Rundbox zeichnen	J		x1	y1	x2	y2	mst	Ein Rundreck 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				
<b>Kontroll- / Definitions-Befehle</b>											
Bargraph definieren	B	R L O U	nr	x1	y1	x2	y2	aw	ew	mst	Einen 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 Nr. (0..7)
Display Control	C		n1							n1=0:Display Aus Inhalt bleibt erhalten; n1=1:Display Ein Inhalt sichtbar	
Selekt / Deselekt Grafikkontroller	K	S D	n1							Kontroller mit Adresse n1 (n1=0..3; n1=4: alle) aktivieren Kontroller mit Adresse n1 (n1=0..3; n1=4: alle) deaktivieren	
Power-Save-Modus	Q		n1							n1=1: Powersave für Grafikkontroller; RTS->HIGH; n1=2: Powersave für Grafikkontroller, RTS->HIGH; LCDON->LOW	
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		
<b>Sende-Befehle</b>											
Hardcopy	H		x1	y1	x2	y2			Es wird der angegebene Bildinhalt angefordert. Zuerst werden die Breite und Höhe in Pixel und dann die eigentlichen Bilddaten 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: 128, 64, 'V' (128x64 Pixel Auflösung, vertikal Organisiert)	

## ELECTRONIC ASSEMBLY

### PARAMETER

Alle Befehle und deren Parameter wie Koordinaten und sonstige Übergabewerte werden immer als Bytes erwartet. Dazwischen dürfen keine Trennzeichen z.B. Leerzeichen oder Kommas verwendet werden. Die Befehle benötigen auch **kein Abschlussbyte** wie z.B. Carriage Return.

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

**x1, x2, y1, y2** ..... Koordinatenangaben werden mit 1 Byte übertragen; gültig sind Werte von 0..127 für x- bzw. 0..63 für y-Koordinaten.  
Beispiel: x1= 10 (dez.) = \$0A

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

### PROGRAMMIERBEISPIEL

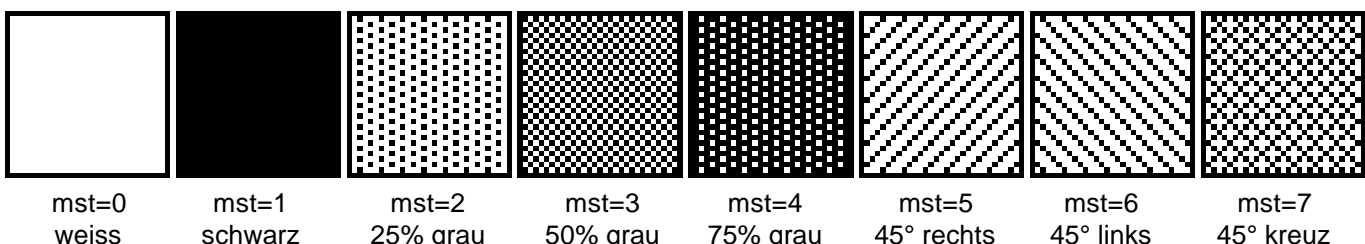
In der nachfolgenden Tabelle ist ein Beispiel zu sehen, welches die Zeichenkette "Test" an den Koordinaten 7,3 ausgibt.

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

### MUSTER

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

Folgende Füllmuster stehen dabei zur Verfügung:



### BESCHREIBUNG DER EINZELNEN GRAFIKFUNKTIONEN

Auf den nächsten Seiten befindet sich eine detaillierte alphabetisch sortierte Beschreibung zu jeder einzelnen Funktion. Als Beispiel wird jeweils ein vergrößerter Bildausschnitt von 50 x 32 Pixeln als Hardcopy gezeigt, der den Displayinhalt nach Ausführung des Befehls darstellt. In den Beispielen sind die zu übertragenden Bytes als Hex-Werte abgebildet.

#### A x1 y1 n1

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

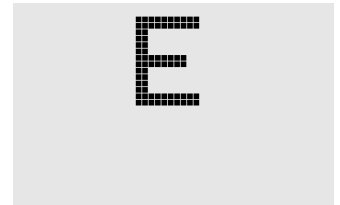
Beispiel: \$41 \$13 \$02 \$45

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

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

Textmodus: Replace und Muster Schwarz

#### ASCII-Zeichen setzen



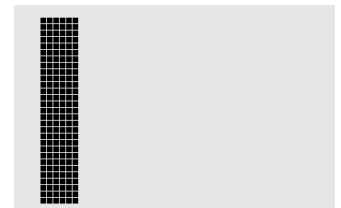
#### B L/R/O/U nr x1 y1 x2 y2 aw ew mst

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 Vollausschlag einen Bereich mit den Koordinaten **x1,y1** bis **x2,y2**. Mit dem Anfangswert (kein Ausschlag) **aw** (=0..254) und dem Endwert (Vollausschlag) **ew** (=0..254) wird der Bargraph skaliert. Der Bargraph wird immer im Inversmodus mit dem Muster **mst** gezeichnet: Der Hintergrund bleibt somit in jedem Fall erhalten. (Achtung! Nach 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 definiert. Bei Vollausschlag nimmt er einen Bereich von den Koordinaten 4,2 bis 9,30 ein. Anfangs- und Endwert entspricht einer 4..20 mA Anzeige. (Das Bild zeigt den Bargraph im Vollausschlag wie er mit \$42 \$01 \$14 dargestellt wird)

#### Bargraph definieren



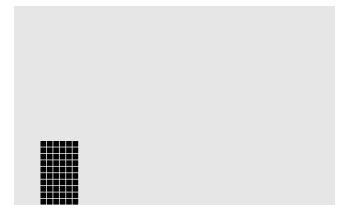
#### B nr wert

Der Bargraph mit der Nummer **n1** (1..8) wird auf den neuen Wert eingestellt (**aw** <= **wert** <= **ew**). Wenn **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.

#### Bargraph zeichnen



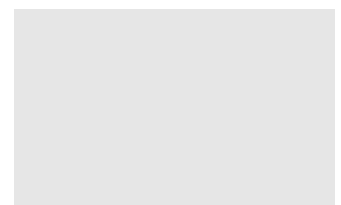
#### C n1

schaltet das Display Ein (**n1=1**) oder Aus (**n1=0**); alle Displaydaten bleiben erhalten und es können weiterhin Befehle ausgeführt werden.

Beispiel: \$43 \$00

Der Displayinhalt wird unsichtbar, der Inhalt bleibt jedoch erhalten.

#### Display Control





### 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

### E n1 daten

### Zeichen definieren

Es ist möglich, bis zu 16 Zeichen selbst zu definieren (je nach Fontgröße). Diese Zeichen haben dann die ASCII Codes 1 bis max.16 und bleiben bis zum Abschalten der Versorgungsspannung in einem 64 Byte großen internen RAM-Bereich erhalten. Bei Font 1 können bis zu 16 Zeichen definiert werden, bei Font 2 noch 10 Zeichen und beim größten Font 3 immer noch 4 Zeichen. Achtung! Sollen mehrere Zeichen aus unterschiedlichen Fonts definiert werden, so ist darauf zu achten daß z.B. ein Zeichen mit Code 1 vom 8x16 Font denselben Platz im RAM benötigt wie die Zeichen mit den Codes 1 bis 4 vom 4x6 Font (siehe Tabelle nebenan) !

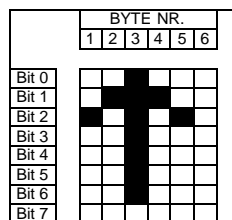
Beispiel 1:

Mit dem Befehl

\$45 \$03

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

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



Beispiel 2:

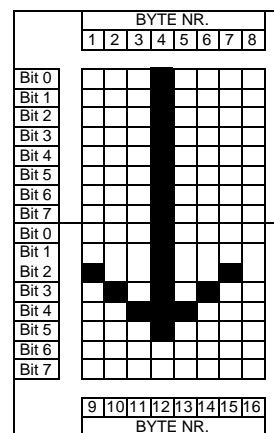
Mit dem Befehl

\$45 \$02

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

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

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



Selbstdefinierbare Zeichen (Code)		
4x6	6x8	8x16
1	1	1
2	2	
3		
4	3	
5	4	2
6	5	
7		
8	6	3
9	7	
10		
11	8	
12	9	4
13	10	
14		
15	16	

### F n1 n2 n3

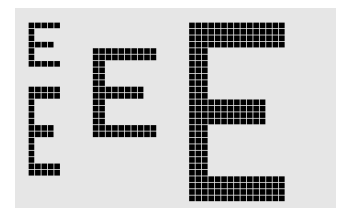
### Font einstellen

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

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

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

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



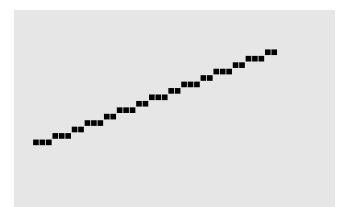
### G x1 y1 x2 y2

### Gerade zeichnen

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

Beispiel: \$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 die Höhe des Bildausschnittes und danach die Bilddaten. Zum Aufbau der Bilddaten siehe den Befehl Bild Upload 'U'.

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

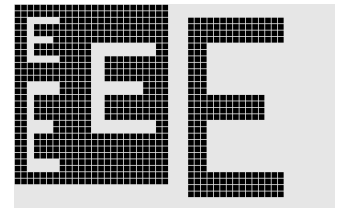
und sofort wird der linke obere Teil des Bildschirms mit der Grösse 32 x 16 Pixel über RS-232 gesendet.

**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

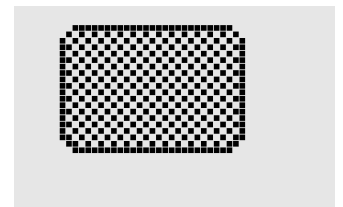
invertiert bei vorhandenem Displayinhalt aus dem Beispiel "Font einstellen" den Bereich von 0,0 nach 23,27.

**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' Runddeck 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 16).

Beispiel: \$4B \$44 \$00

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

**L x1 y1 x2 y2****Bereich löschen**

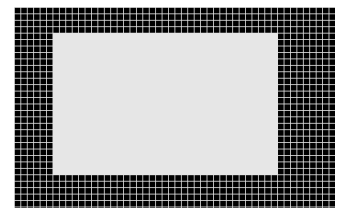
Der Bereich von der linken oberen Ecke **x1,y1** bis zur rechten unteren Ecke **x2,y2** wird gelöscht.

Beispiel:

\$44 \$53

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

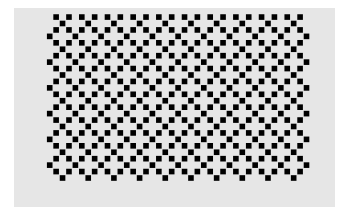
Zuerst wird das Display mit 'D', 'S' gefüllt und dann der Bereich von 6,4 nach 40,25 gelöscht .

**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.



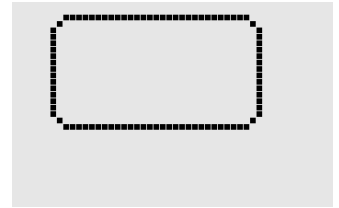
## ELECTRONIC ASSEMBLY

**N** x1 y1 x2 y2

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 Rundercks wird nicht verändert. Vergleiche 'J' Rundbox zeichnen.

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

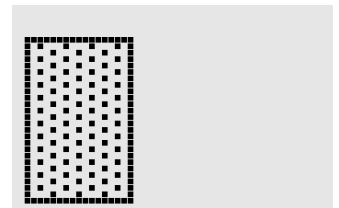
zeichnet ein Runderck von 6,2 nach 38,19.

**Runderck zeichnen****O** x1 y1 x2 y2 mst

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

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

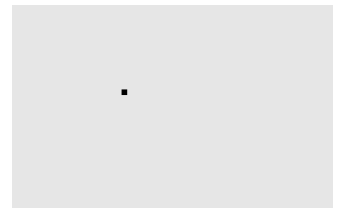
zeichnet eine Box von 2,5 nach 18,30 mit dem Muster 2=25%Grau.

**Box zeichnen****P** x1 y1

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.

**Punkt setzen****Q** n1

Beispiel: \$51 \$01

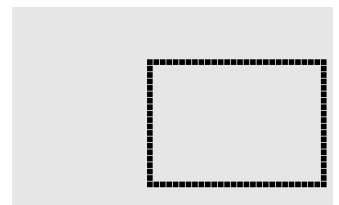
schaltet den High-Level Grafikkontroller in den Power-Save-Modus. Die Handshakeleitung RTS zeigt mit H-Pegel an, daß keine weiteren Befehle mehr angenommen werden. Die Versorgungsspannung kann nun auf bis zu 2V reduziert werden. Durch einen >10ms langen Impuls am Reset Pin startet der Kontroller wieder. Die Registerinhalte werden, wie in der Tabelle auf der Seite 3 angegeben, gesetzt bzw. bleiben unverändert.  
n1: 1, High-Level Grafikkontroller im Power-Save-Modus (Pin LCDON bleibt auf VDD-Pegel: Display ein)  
n1: 2, High-Level Grafikkontroller und Display im Power-Save-Modus (Pin LCDON auf GND-Pegel: Display aus)

**Power Save Modus****R** x1 y1 x2 y2

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' Runderck zeichnen.

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

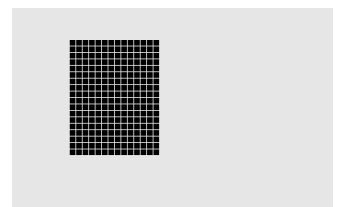
zeichnet ein Rechteck von 21,8 nach 48,37.

**Rechteck zeichnen****S** x1 y1 x2 y2

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

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

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

**Bereich füllen**

## T L/R/O/U n1 mst

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.

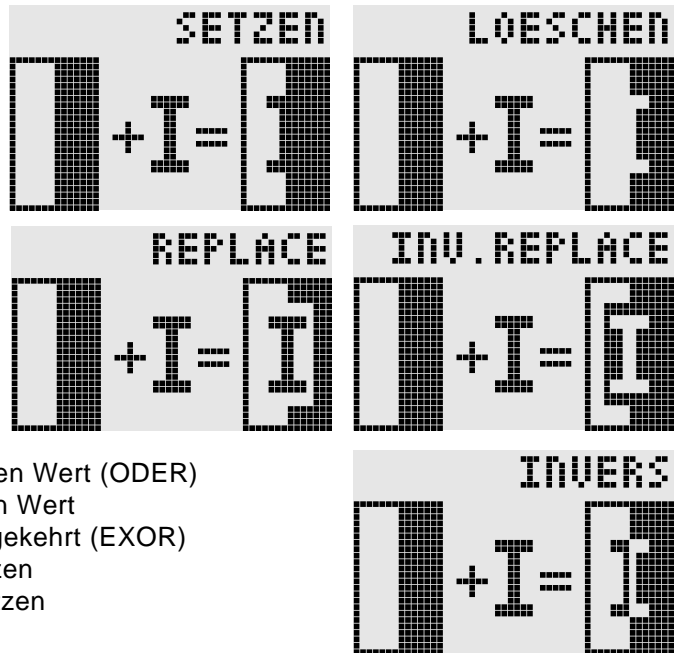
Beispiel: \$54 \$52 \$03 \$03

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

## Text-Modus einstellen

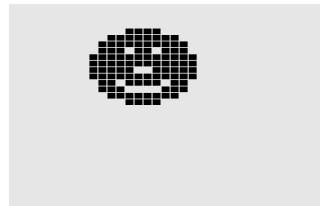


## U x1 y1 daten

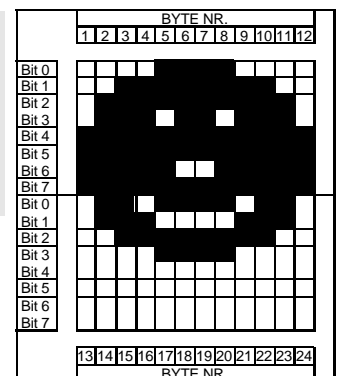
Ein Bild wird an die Koordinate **x1,y1** geladen.

**daten:** - 1 Byte für die Bildbreite in Pixeln  
 - 1 Byte für die Bildhöhe in Pixeln  
 - Bilddaten: Anzahl = ((Höhe+7) / 8) \* Breite Bytes.  
 1 Byte steht für 8 senkrechte Pixel am Bildschirm;  
 0=weiß, 1=schwarz; LSB: oben, MSB: unten;  
 Das Bild ist von links nach rechts abgelegt.  
 Das Programm BMP2BLV.EXE erzeugt aus monochromen Windows-Bitmap-Grafiken die Bilddaten inkl. der Angabe von Breite und Höhe.

Beispiel: \$55 \$09 \$04 \$0C \$0C  
 \$F0 \$FC \$FE \$FE \$F7 \$BF \$BF \$F7 \$FE \$FE \$FC \$F0  
 \$00 \$03 \$07 \$06 \$0D \$0D \$0D \$0D \$06 \$07 \$03 \$00



## Bild Upload



lädt das nebenstehende Bild an die Koordinate 9,4.

## V n1

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

Beispiel: \$56 \$03

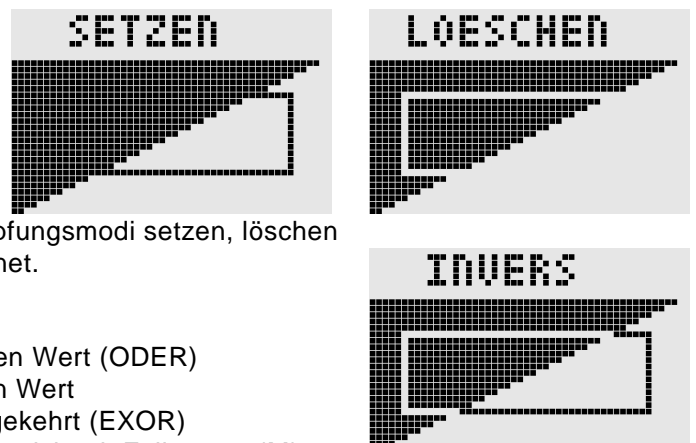
stellt den Verknüpfungsmodus auf invers.

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

Verknüpfungsmodus n1:

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

## Grafik-Modus einstellen

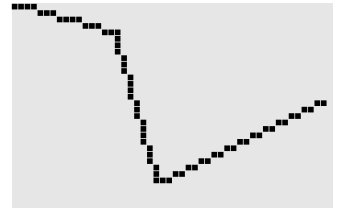


**W x1 y1****Gerade weiterzeichnen**

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

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 4.

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 4.

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 Koordinatenangaben beziehen sich auf das linke obere Eck des Zeichens. Achtung: Font Nr.1 zeigt nur Großbuchstaben.



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

schreibt an die Koordinate 6,11 die Zeichenkette "Test"

Eingestellter Font: 8x16 mit normaler Breite und Höhe

Textmodus: Schreibrichtung nach Rechts, Verknüpfung Replace mit Muster Schwarz

**?****Displaytyp abfragen**

Die Auflösung des Displays und die Art des Bildaufbaus wird abgefragt. Bei dem High-Level-Grafikkontroller IC202-PGH ist die Auflösung immer 128 x 64 Pixel und der Bildaufbau vertikal organisiert. Dieser Befehl ist für externe Programme die auf den High-Level-Grafikkontroller zugreifen gedacht. Das IC6963-PGH ist z.B. für Displays mit Toshibacontroller bei dem die Auflösung variabel bis 240x128 Pixel und die Organisation horizontal ist.

Beispiel: \$3F

Nach diesem Befehl wird zuerst die X- (128) und Y-Auflösung (64) und dann die Art des Bildaufbaus ('V') für die vertikale Organisation über die RS-232 Schnittstelle gesendet.

## INTEGRIERTE FONTS

Im High-Level Grafikkontroller EA IC202 sind 3 Zeichensätze integriert (Font1: 4x6 Pixel; Font2: 6x8 Pixel und Font 3 8x16 Pixel). Jeder Zeichensatz kann in 1-, 2-, 3- oder 4-facher Höhe verwendet werden. Unabhängig davon läßt sich auch die Breite verdoppeln, verdreifachen oder vervierfachen. Zusätzlich können 4-16 eigene Zeichen definiert werden, die solange erhalten bleiben, bis die Versorgungsspannung abgeschaltet wird. (Siehe Befehl 'E').

+ Lower Upper	\$0 (0)	\$1 (1)	\$2 (2)	\$3 (3)	\$4 (4)	\$5 (5)	\$6 (6)	\$7 (7)	\$8 (8)	\$9 (9)	\$A (10)	\$B (11)	\$C (12)	\$D (13)	\$E (14)	\$F (15)
\$20 (dez: 32)		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
\$30 (dez: 48)	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
\$40 (dez: 64)	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	
\$50 (dez: 80)	p	q	r	s	t	u	v	w	x	y	z	[	\	]	^	_

Font 1

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

+ Lower Upper	\$0 (0)	\$1 (1)	\$2 (2)	\$3 (3)	\$4 (4)	\$5 (5)	\$6 (6)	\$7 (7)	\$8 (8)	\$9 (9)	\$A (10)	\$B (11)	\$C (12)	\$D (13)	\$E (14)	\$F (15)
\$20 (dez: 32)		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
\$30 (dez: 48)	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
\$40 (dez: 64)	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)	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
\$90 (dez: 144)	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	
\$A0 (dez: 160)	p	q	r	s	t	u	v	w	x	y	z	[	\	]	^	_
\$B0 (dez: 176)	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
\$C0 (dez: 192)	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	
\$D0 (dez: 208)	p	q	r	s	t	u	v	w	x	y	z	[	\	]	^	_
\$E0 (dez: 224)	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
\$F0 (dez: 240)	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	

Font 2

+ Lower Upper	\$0 (0)	\$1 (1)	\$2 (2)	\$3 (3)	\$4 (4)	\$5 (5)	\$6 (6)	\$7 (7)	\$8 (8)	\$9 (9)	\$A (10)	\$B (11)	\$C (12)	\$D (13)	\$E (14)	\$F (15)
\$20 (dez: 32)		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
\$30 (dez: 48)	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
\$40 (dez: 64)	a	b	c	d	e	f	g	h	i	j	k	l	m	n	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)	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
\$90 (dez: 144)	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	

Font 3

### EA GE120-5NV24 LOWCOST GRAFIK EINHEIT 120x32, 2 FONTS, RS-232

<b>Abmessungen</b>	68 x 39 x 11mm
<b>Display</b>	120x32, Supertwist, LED- beleuchtet
<b>Betriebsspannung</b>	+5V / 100mA
<b>Anschluß</b>	10 Pins, Raster 2,54mm
<b>Arbeitstemperatur</b>	0 ... +70°C
<b>Lagertemperatur</b>	-20 ... +70°C



- \* 120x32 Pixel Supertwist-LCD mit LED-Beleuchtung grün
- \* 2 FONTS von 3,2mm (5x30 Zeichen) über 4,6mm (4x20) bis 18,2 mm (1x5)
- \* GERADEN ZEICHNEN, BEREICHE LÖSCHEN UND INVERTIEREN
- \* VERSORGUNGSSPANNUNG: +5V / typ. 100mA
- \* BAUDRATEN 1200,2400,4800,9600,19200BD

#### Bestellbezeichnung

Grafikeinheit 120x64 mit LED-Bel. **EA GE120-5NV24**  
 Kabel mit 9-pol. D-SUB Stecker (female) **EA KV24-9B**

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

Die Grafikeinheit EA GE128-6N3V24 baut auf dem High-Level-Grafikkontroller EA IC202-PGH auf. Diese Grafikeinheit kann wahlweise mit echten RS-232C Pegeln ( $\pm 10V$ ) oder mit CMOS-Pegeln betrieben werden.

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



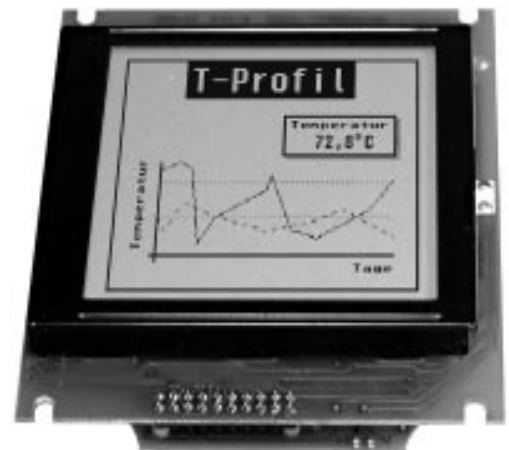
- \* 128x64 Pixel Supertwist-LCD mit LED-Beleuchtung grün
- \* 3 FONTS: 2,6mm (10x32 Zeichen) über 6,2mm (4x16) bis 25 mm (1x5)
- \* GERADEN ZEICHNEN, BEREICHE LÖSCHEN UND INVERTIEREN
- \* AUTOMATISCH SKALIERENDE BARGRAPHFUNKTION
- \* VERSORGUNGSSPANNUNG: +5V / typ. 150mA
- \* BAUDRATEN 1200,2400,4800,9600,19200BD

#### Bestellbezeichnung

Grafikeinheit 128x64 mit LED-Bel. **EA GE128-6N3V24**  
 Kabel mit 9-pol. D-SUB Stecker (female) **EA KV24-9B**

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

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



- \* 128x128 Pixel Supertwist-LCD mit LED-Beleuchtung grün
- \* 3 FONTS von 2,2mm (21x32 Zeichen) über 5,2mm (8x16) bis zu 21 mm (2x5)
- \* GERADEN ZEICHNEN, BEREICHE LÖSCHEN UND INVERTIEREN
- \* AUTOMATISCH SKALIERENDE BARGRAPHFUNKTION
- \* VERSORGUNGSSPANNUNG: +5V / typ. 500mA
- \* BAUDRATEN 1200,2400,4800,9600,19200BD

#### Bestellbezeichnung

Grafikeinheit 128x128 mit LED-Bel. **EA GE128-7KV24**  
 Kabel mit 9-pol. D-SUB Stecker (female) **EA KV24-9B**

Weitere Displaygrößen in Vorbereitung: z.B. 240x64 und 240x128 (Stand: Juni '97).

# EA IC202

## 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 = 1	QTYP = 0		
			11,0592 MHz	3,6864 MHz	7,3728 MHz	14,7456 MHz
0	0	0	1200	300	600	1200
0	0	1	2400	600	1200	2400
0	1	0	4800	1200	2400	4800
0	1	1	9600	2400	4800	9600
1	0	0	19200	4800	9600	19200
1	0	1	38400	9600	19200	38400
1	1	0	57600	14400	28800	57600
1	1	1	115200	28800	57600	115200

## ADRESSIERUNG

Bis zu 4 High-Level Grafikkontroller können an einer seriellen Schnittstelle adressiert betrieben werden. Die jeweilige Adresse wird über die Pins ADR0 und ADR1 eingestellt.

**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.

Adressen		
ADR 1	ADR 0	Adresse
0	0	0
0	1	1
1	0	2
1	1	3

## ABMESSUNGEN

Gehäuse: PLCC44J; alle Maße in Inch (mm)

