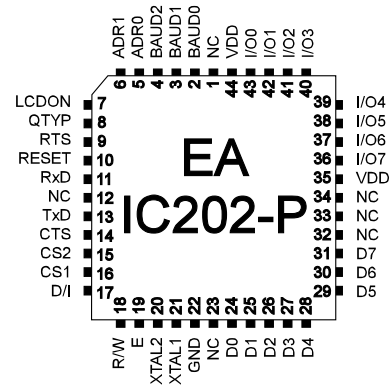


# HIGH-LEVEL GRAPHICS CONTROLLER FOR HD61202 (KS0108B) - DISPLAYS



PLCC44J

## FEATURES

- \* FOR LC GRAPHICS DISPLAYS 128x64 WITH HD 61202 OR COMPATIBLE
- \* NO TIMING PROBLEMS WITH FAST BUS SYSTEMS
- \* EASY PROGRAMMING OF VARIOUS BUILT IN GRAPH FUNCTIONS:
- \* STRAIGHT LINE, DOT, AREA, AND/OR/EXOR, BAR GRAPH, PATTERNS...
- \* 3 DIFFERENT FONTS BUILT IN
- \* ZOOM FUNCTION FOR ALL FONTS (2-, 3- AND 4-TIMES)
- \* 4-16 FREE DEFINEABLE CHARACTERS (DEPENDING ON SIZE)
- \* MIXING TEXT AND GRAPHICS
- \* INPUT ON RS-232 WITH  $\pm 10V$ / CMOS-LEVEL
- \* DIRECT ACCESS TO MAX232 OR EQUIVALENT
- \* PROGRAMMABLE BAUD RATES FROM 300 UP TO 115,200 BAUD
- \* ONLY MAX. 4 EXTERNAL COMPONENTS REQUIRED
- \* 8 DIGITAL I/O'S FREE AVAILABLE FOR CUSTOM DESIGNS
- \* 5V / typ. 12mA RATED INPUT (AT 11,059 MHz)
- \* POWER-SAVE FUNCTION: CONSUMPTION max. 100 $\mu$ A @ 5V
- \* HARDWARE CODES UP TO 4 ADDRESSES

## OPTIONS (IN PROGRESS)

- \* VERSION FOR VERTICAL DISPLAY MOUNTING (90° ROTATED)
- \* 40-PIN DIL PACKAGE VERSION

## ORDER DESIGNATION

HIGH-LEVEL GRAPH CONTROLLER 128x64 FOR HD61202	<b>EA IC202-PGH</b>
SUITABLE GRAPHICS DISPLAY INCL. HD61202, 128x64	<b>EA P128-6N3LED</b>
CERAMIC RESONATOR SMD 7,37MHz, 3 PINS INCL. CAPs	<b>EA KERS7M37-C</b>
GRAPHICS MODULE WITH IC202-PGH AND RS-232C	<b>EA GE128-6N3V24</b>

**ELECTRONIC  
ASSEMBLY** GMBH

LOCHHAMER SCHLAG 17 · D-82166 GRÄFELFING  
PHONE +49-89-8541991 · FAX +49-89-8541721

## GENERAL

The EA IC202 high-level Graphics Controller links your system processor to your graphic display. Inputs accepts a serial asynchronous RS-232 interface. The graphics controller includes complete graph routines and various character sizes.

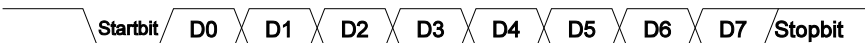
Programming is made by high level language graph commands; time consuming programming of character sets and graph routines is not necessary anymore. Expenditure for developing of your product is reduced significant and several features are gained on top of it:

- no timing problems with fast processor bus
- enough memory space (operating memory and characterset memory especially for  $\mu\text{C}$ )
- no time consuming graphic calculations which would slow down processor speed

Connecting to hardware is very simple. Display and main processor are connected directly. Drivers, decoders or port modules are not required. A simple display input need 1 wire (RxD) only. 2 up to maximal 4 external components will complete the circuit: a quartz with 2 capacitors and a reset-capacitor. **No external components** are needed when you operate with 8051-compatible systems. Clock and reset signal can be taken from main processor.

## HARDWARE

Supply voltage of system is +5 Volts. Data transfer is asynchronous serial in RS-232 format at CMOS level. Data format is set to 8 data bits, 1 stop bit and no parity. Baud rate can be selected on 3 pins from 150 Baud up to 115200 Baud. Handshake lines RTS and CTS on board too. There is no interpretation needed for small quantities of datas.

Data format:  Startbit / D0 / D1 / D2 / D3 / D4 / D5 / D6 / D7 / Stopbit

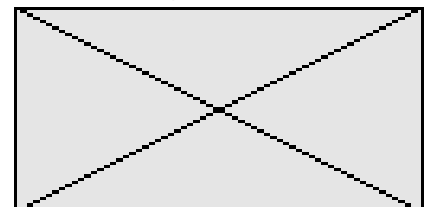
Additional 8 I/O-ports are available for free usage. This may be wired as inputs or outputs on individual desire. Possible application is switching of backpanel illumination or reading of several keys.

## SOFTWARE

Programming of this High-Level graphics controller is performed by commands like i.e. "draw a rectangular box from (0,0) to (64,15) which origins in top left hand corner of display". Therefore the serial interface has to transmit this sequence of bytes: \$52 \$00 \$00 \$40 \$0F. Characters string can be placed precisely to the pixel. Mixing of graphic images with text elements is possible anytime. Three different character sets are available where each of them can be zoomed by factors 2-, 3- or 4-times. The biggest character set 16x8 allows when using 4-x zoom (=64x32) a totally filled display with letters and numbers.

## TEST MODE

As long as pin 9 (RTS) is after Power-On or after Reset connected with GND, the graphics controller is in test mode. Display shows now a marked flashing box. When connection Pin 9 (RTS) to GND is removed, the graphics controller returns to normal operation mode but testbox still remains visible.



## ELECTRONIC ASSEMBLY

## TECHNICAL DESCRIPTION

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

Datas are valid for  $T_a = 0..+70^{\circ}\text{C}$  and  $VDD = 5,0\text{V} \pm 20\%$  if not noted otherwise.

## POWER-SAVE FUNCTION

Power-Save function allows use in battery powered devices. Command "Q \$01" puts the High-Level-Graphics-Controller from normal operation into power save mode. Current consumption (without display) is reduced to about 100µA. Because display is connected to supply voltage, still some mA current are drawn depending on type of display. Total content of display, i.e. all by your own created characters and bargraph-definitions remain preserved. In Power-Save-mode no commands will be accepted and executed (RTS at H-level).

Command "Q \$02" sets controller and a transistor driven display (see application example on page 5) to Power-Save Mode. Note that content of display gets lost but all by your own created characters and bargraph-definitions remain preserved. Also this mode will not accept and execute commands (RTS at H-level).

An at least 10ms wide reset puls (H-level) turns the graph controller back to normal operation. The internal registers are resetted as listed in above table.

Default settings			
Register	Comman	after Power-On	after Power-Save-Mo
Display Control	C	Display on	Display off
Text-Mode	T	right, set, black	right, set, black
Graphics Mode	V	set	set
Font	F	6x8	6x8
Font Faktor width/height	F	1/1	1/1
Last xy	W	(0;0)	(0;0)
Self-defined character	E	undefined	as defined
Bar Graph 1..8	B	undefined	as defined
High-Level Graphics controller	K	selected	selected
Inputs/Outputs I/O0..7	Y	H-Level	H-Level

## PINNING

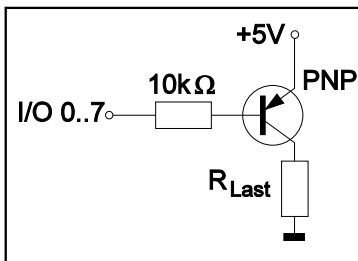
Pin Description				
Pin	Symbol	In / Out	Level	Description
1	NC			do not connect
2,3,4	BAUD0..2	In	lo	Baud Rates
5,6	ADR0, ADR1	In	lo	Adressing
7	LCDON	Out	hi	lo: Power-Save-Mode, hi: Controller/LCD on
8	QTYP	In	lo	Crystal type, see Baud Rates
9	RTS	Out	lo	lo: RS-232 Data receive enable; hi: RS-232 Data receive disable
10	RESET	In	hi	Default controller settings
11	RxD	In	lo	RS-232 Receive
12	NC			do not connect
13	TxD	Out	lo	RS-232 Transmit
14	CTS	In	lo	lo: RS-232 Data transmit enable; hi: RS-232 Data transmit disable
15,16	CS2, CS1	Out	hi	Display: left (CS1) or right (CS2) half of display
17	D/I	Out		Display: hi: Data; lo: Command
18	R/W	Out		Display: hi: Data/Command read; lo: Data/Command write
19	E	Out	hi	Display: Enable
20	XTAL2	Out		System clock
21	XTAL1	In		System clock (external input)
22	GND	GND	lo	Power supply 0V
23	NC			do not connect
24,25,26,27, 28,29,30,31	D0..7	I/O		Display: Data lines
32,33,34	NC			do not connect
35	VDD	VDD	hi	Power supply +5V
36,37,38,39, 40,41,42,43	IO7..0	I/O		8 Inputs/Outputs
44	VDD	VDD	hi	Power supply +5V

## DIGITAL IN-/ OUTPUTS IO 0..7

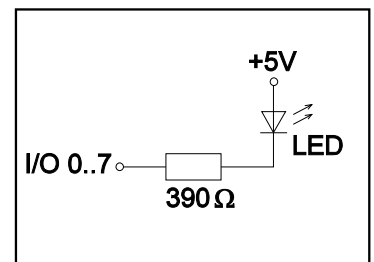
8 pins of this High-Level Graphics Controller may be used as free available programmable In- and Outputs. Also mixed operation of i.e. 3 Outputs and 5 Inputs is possible.

### Output circuit

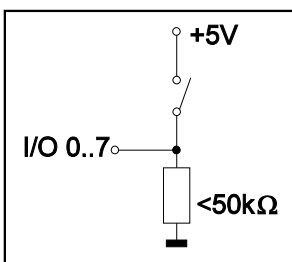
Command "Y n1 n2"<sup>1)</sup> sets any selected pin IO 0..7 to H- or L-level; it can be compared with an Open-Drain output without pull-up resistor. Current



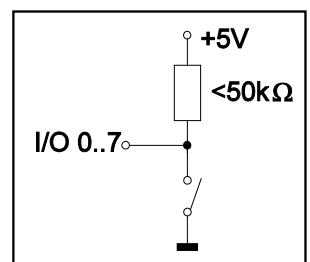
flows only when L-level is applied. A single pin may be loaded with max. 10mA, all pins together may be loaded with 26mA in all, i.e. 2 pins @ 10mA plus 1pin @ 6mA. Its possible i.e. to connect and switch a LED directly. Higher current can be provided by an external transistor circuit. Note that after Power-On resp. in Power-Save-mode all outputs will be at H-level.



### Input circuit



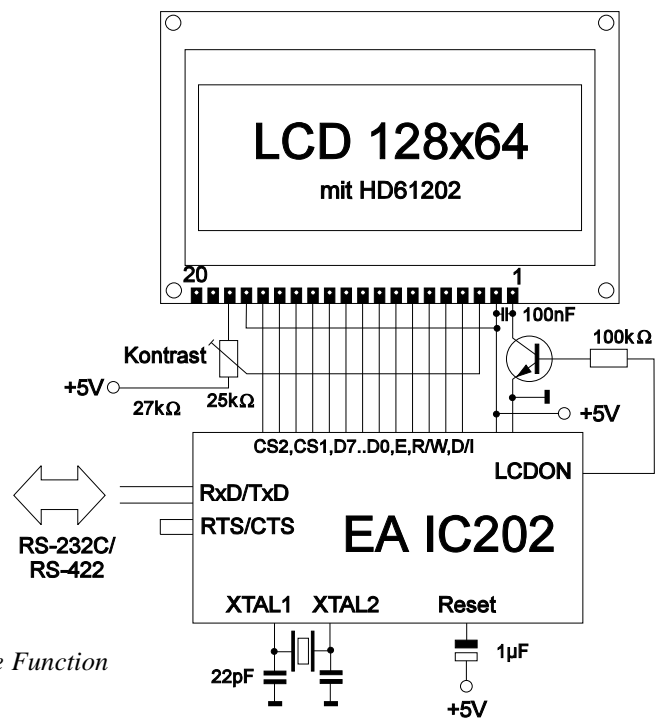
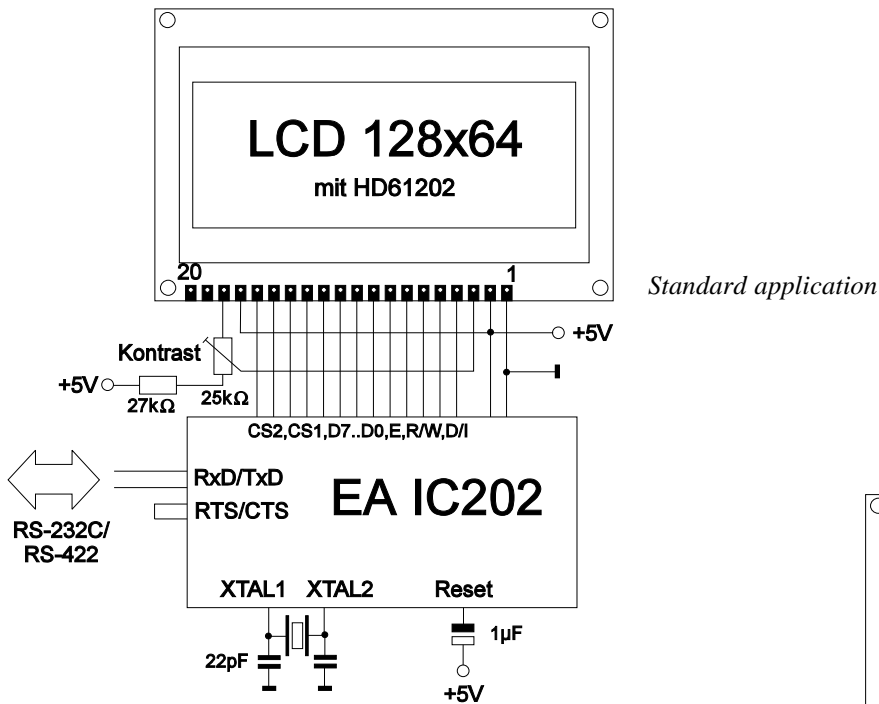
Input pins accept voltage levels between -0,5V and +0,2V\*VDD-0,1V. Leakage current is max. ±10µA. Trigger levels are listed in table of technical description on page 3. Command "X n1"<sup>1)</sup> allows read in of datas on each individual pin IO 0..7. Voltage level must be stable all over read in procedure. There is no built in debounce function.



<sup>1)</sup>Command descriptions you will find on page 13.

## ELECTRONIC ASSEMBLY

### APPLICATION EXAMPLES



### APPLICATION HINTS

In order to ensure trouble free operation, you should be aware of basic construction rules in digital electronic applications already during development of circuit routing:

- Watch for straight ground routing in your layout (no ground loops)
- Supply voltage distribution is best performed star shaped by wide conductor tracks, preferably by multilayer boards with separate power supply layers.
- Components resp. circuits with high or variable current consumption need separate supply cables. Remaining circuits must be decoupled and filtered on demand. Also LED-illumination of display should be supplied separate.
- Use blocking capacitors at all active components.
- Keep tracks carrying high frequency signals resp. high rising slopes as short as possible (XTAL1 and XTAL2 !)



## OVERVIEW OF ALL GRAPHIC FUNCTIONS

Our High-Level Graphics Controller can be programmed by a number of built in commands. Each command starts with a command letter followed by a number of parameters.

Command table EA IC202											
Command											Remarks
<b>Functions for outputting text</b>											
Text mode	T	R L O U	n1	ptn							R/L/O/U: Write character string (R)ight, (L)eft, (O)pen (up), (U)nten (down); n1: overlay combination mode for text output 1=set; 2=delete; 3=inverse; 4=replace; 5=inverse replace; ptn: use pattern no. 0..7;
Set font	F	n1	n2	n3							Set font no. n1; n1=1:4x6 font; n1=2:6x8 font; n2=3:8x16 font n2+n3=zoom factor (1..4); n2=X factor; n3=Y factor;
Set ASCII characters	A	x1	y1	n1							The character n1 will be set at coordinate x1,y1. (Reference top left)
Set character string	Z	x1	y1	...	NUL						Output character string (...) to x1,y1; character 'NUL' (\$00)=end
Define character	E	n1	data ...								n1=character no.; data =number of bytes dep. on current font
<b>Graphics commands with overlay mode</b>											
Graphics mode	V	n1									n1: 1=set; 2=delete; 3=inverse; 4=replace; 5=inverse replace;
Set point	P	x1	y1								Set one pixel at coordinates x1, y1
Draw straight line	G	x1	y1	x2	y2						Draw straight line from x1,y1 to x2,y2
Continue straight line	W	x1	y1								Draw a straight line from last end point to x1, y1
Draw rectangle	R	x1	y1	x2	y2						Draw a rectangle; x1,y1,x2,y2 = opposite corner points
Draw round corner	N	x1	y1	x2	y2						Draw a rectangle with round corners; x1,y1,x2,y2 = corner points
Area with fill pattern	M	x1	y1	x2	y2	ptn					Draw area with pattern ptn (0..7); x1,y1,x2,y2 = corner points
<b>Other graphics commands</b>											
Delete display	D	L									Delete entire contents of display (set to white);
Invert display	D	I									Invert entire contents of display;
Fill display	D	S									Fill entire contents of display; (set to black);
Delete area	L	x1	y1	x2	y2						Delete an area; x1,y1,x2,y2 = opposite corner points
Invert area	I	x1	y1	x2	y2						Invert an area; x1,y1,x2,y2 = opposite corner points
Fill area	S	x1	y1	x2	y2						Fill an area; x1,y1,x2,y2 = opposite corner points
Draw box	O	x1	y1	x2	y2	ptn					Draw a rectangle with fill pattern ptn (0..7); (always replace)
Draw round box	J	x1	y1	x2	y2	ptn					Draw a round corner with fill pattern ptn (0..7); (always replace)
Draw bar graph	B	nr	valu								Set the bar graph with the 'nr' (1..8) to the new user 'value'
Upload picture area	U	x1	y1	data ...							Load a picture area to x1,y1; see picture structure for picture data
<b>Control / definition commands</b>											
Define bar graph	B	R L O U	nr	x1	y1	x2	y2	aw	ew	ptn	Define bar graph to L(ef), R(ight), O(up), U(down) with the 'nr' (1..8). x1,y1,x2,y2 form the rectangle enclosing the bar graph. aw, ew are the values for 0% and 100%. ptn=pattern (0..7).
Display control	C	n1									n1=0: display off (entire contents unchanged) n1=1: display on
Select / Deselect graphics lcd	K	S	n1								Activate display with address n1 (n1=0..3; n1=255: all)
		D									Deactivate display with address n1 (n1=0..3; n1=255: all)
Power Save Mode	Q	n1									n1=1: Power Save for controller; RTS->high n1=2: Power Save for controller and display; RTS->high, LCDON->low
Write I/O port	Y	n1	n2								n1=0..7: reset I/O port n1 (n2=0); set (n2=1); invert (n2=2) n1=8: Set all 8 I/O ports in accordance with n2 (=8 bit binary value)
<b>Send commands</b>											
Hard copy	H	x1	y1	x2	y2						An area is requested as a picture. The width and height are sent in pixels first of all, followed by the actual picture data, via RS232.
Read I/O port	X	n1									n1=0..7: load I/O port <n1> (1=H level=5V, 0=L level=0V) n1=8: load all 8 I/O ports I/O0..I/O7 as 8-bit binary value
Query display type	?										This command is used to query the display type. 3 bytes are sent back: 128 64 V (128x64 dots, vertical picture)

## ELECTRONIC ASSEMBLY

### PARAMETERS

All commands with parameters, coordinates and other hand over datas are expected as of Bytes. No seperating characters are allowed, i.e. no space bars, no commas. End of command **does not need a final Byte** i.e. no Carrige Return.

**A..Z, L/R/O/U** ..... All commands are transmitted as ASCII code.  
Example: G = 71 (dec.) = \$47 initiates the straight line drawing command.

**x1, x2, y1, y2** ..... Coordinates are transmitted with one Byte; applicable values are 0..127 for x- resp. 0..63 for y- coordinates.  
Example: x1= 10 (dec.) = \$0A

**n1,n2,nr,aw,ew,value,ptn,data** ..... Parameters with numbers are transmitted with one Byte.  
Example: n1= 15 (dec.) = \$0F

### EXAMPLE OF PROGRAMMING

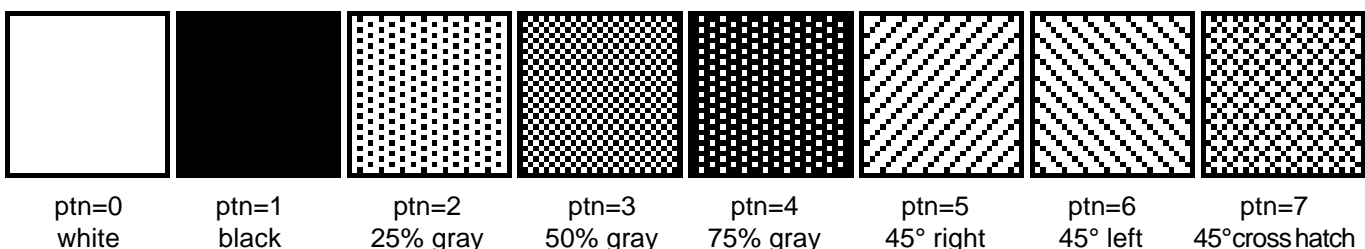
The following table shows an example in wich the character string "Test" is output at coordinates 7,3.

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

### PATTERN

Several commands allow pattern type parameters (ptn = 0..7). They are used e.g. for rectangular areas, bargraphs or text lines with various pattern.

The following fill patterns are available:



## DESCRIPTION OF VARIOUS GRAPHICS FUNCTIONS

The following pages show detailed descriptions in alphabetical order for each individual function. Examples are shown as hardcopy in an enlarged window of 50 x 32 pixel once the command has been executed. Examples show all transferred bytes in Hex code.

**A x1 y1 n1**

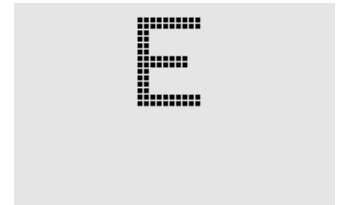
A character **n1** will be displayed on coordinates **x1,y1** with preset font 'F' and text mode 'T' (setzen / löschen / invertieren / replace / invers replace / pattern). Origin is (0,0) at top left hand corner of display. Datas for coordinates apply also to top left hand corner of a given character. Note: Font No.1 shows capital letters only.

Example: \$41 \$13 \$02 \$45

Character 'E' is displayed at coordinates 19,2

Preset font: 6x8, width 2x and height 2x

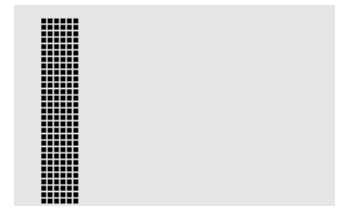
Text mode: Replace and black pattern

**Set ASCII-Characters****B L/R/O/U nr x1 y1 x2 y2 aw ew ptn**

Up to 8 bargraphs (**nr=1..8**) can be defined, which may oscillate to **L=left**, **R=right**, **O=up** or **U=down** direction. Bargraph full level range coordinates are described from **x1,y1** to **x2,y2**. Scaling of bargraph is performed by starting zero position **aw** (=0..254) and max. ending position (full size) **ew** (=0..254). Bargraph always is displayed in inverse-mode using the **ptn**-pattern type: the background remains preserved in any case. (Note: executing this command only the bargraph range is defined but nothing is visible on display).

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

Defines bargraph no. 1 which oscillates vertical up to top. At full level its coordinates ranges from 4,2 to 9,30. Displayed start- and end values represent current values of 4..20 mA. (Hardcopy shows bargraph at its full level operating at \$42 \$01 \$14)

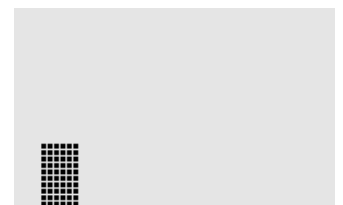
**Define Bargraph****B nr value**

Bargraph number **n1** (1..8) will be adjusted to a new value (**aw <= value <= ew**).

If **value > ew**, than final value will be displayed. Bargraph must be defined before, see above example.

Example: \$42 \$01 \$0A

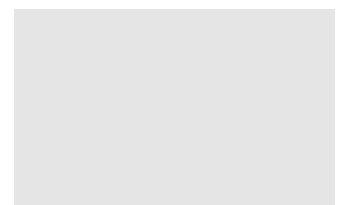
Above defined bargraph no. 1 is set now to value 10.

**Draw Bargraph****C n1**

Switches display On (**n1=1**) or Off (**n1=0**); all display datas remains preserved and commands can be executed furthermore.

Example: \$43 \$00

Content of display becomes invisible, but remains preserved.

**Display Control**



## ELECTRONIC ASSEMBLY

### D L/I/S

### Display Command

Total content of display will be: **L**=deleted (white), **I**=inverted or **S**=filled (black)

Example: \$44 \$49

inverts total content of display

### E n1 daten

### Define Character

Up to 16 characters can be defined by yourself (depending on font size). This characters have ASCII codes from 1 to max.16 and remain stored in an internal 64 Byte RAM-area until the supply voltage is switched off. Up to 16 characters can be defined with font no.1, up to 10 characters with font no.2 and up to 4 characters with the biggest font, no.3. Attention! If several characters out of different fonts should be defined, than you have to consider that i.e. one character from 8x16 font, code 1 needs the same RAM-space as a character from 4x6 font, code 1 to 4 (see table nearby) !

Example 1:

\$45 \$03

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

defines a up pointing arrow with ASCII-No. 3 and with a preset 6x8 character set.

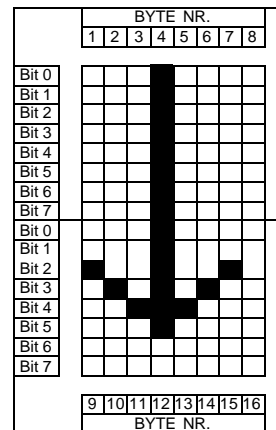
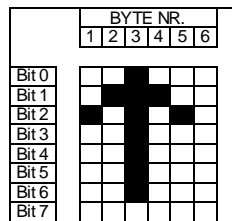
Example 2:

\$45 \$02

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

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

defines a down pointing arrow with ASCII-No. 2 and with a preset 8x16 character set.



Define characters (ASCII)		
4x6	6x8	8x16
1	1	1
2	2	
3		
4		
5	3	2
6	4	
7	5	
8	6	
9	7	3
10	8	
11	9	
12	10	
13	10	4
14	10	
15		
16		

### F n1 n2 n3

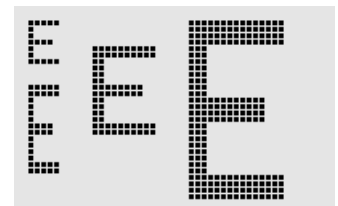
### Set Font

Font No. **n1** (1=4x6 capital letters only; 2=6x8; 3=8x16) is choosed. Different zoom factors (1..4x) for width **n2** and for height **n3** are selected.

Example: \$46 \$02 \$03 \$04

6x8- font is now set to 3-times width and to 4-times height.

Beside hardcopy shows letter 'E' from font 6x8 in different sizes.



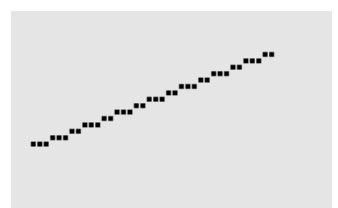
### G x1 y1 x2 y2

### Draw straight line

Straight line is drawn from coordinates **x1,y1** to **x2,y2** taking into account the preset graph mode 'V' (set / clear / invert).

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

Draws a straight line from 3,20 to 50,6.



## H x1 y1 x2 y2

### Get Hardcopy from Display contents

Range from top left hand corner **x1,y1** to right hand bottom corner **x2,y2** is requested. Graphic chip transmits first width and height of requested display-section and afterwards the datas of display-content. See command Uploud picture 'U' for building up display datas.

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

top left hand section of display-content (size 32 x 16 pixel) will be sent via RS-232 immediately.

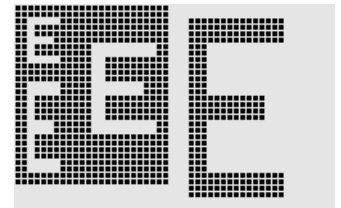
## I x1 y1 x2 y2

### Invert area

The area from top left hand corner **x1,y1** to right hand bottom corner **x2,y2** will be inverted (black pixel turns to white and reversed).

Example: \$49 \$00 \$00 \$17 \$1B

inverts display-content of example "Set Font" in area from 0,0 to 23,27.



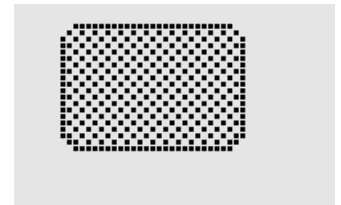
## J x1 y1 x2 y2 ptn

### Draw a round box

A rectangular box with rounded corners is drawn from top left hand corner **x1,y1** to right hand bottom corner **x2,y2** and filled with pattern **ptn** and delete background. See also Draw 'N' rounded corner boxes on next page.

Example: \$4A \$07 \$03 \$23 \$16 \$03

draws a rounded corner box from 7,3 to 35,22 filled with pattern 3=50% black.



## K S/D n1

### Select / Deselect Graphics Controller

**S**=select or **D**=deselect Graphics Controller with its hardware adress **n1** (0..3); adress 255=\$FF is the master-adress for access all Graphics Controller. Address setting is done by hardware (pin ADR0/1 see page 16).

Example: \$4B \$44 \$00

all commands for Graphics Controller with adress \$00 will be ignored from now on.

## L x1 y1 x2 y2

### Delete area

Area from top left hand corner **x1,y1** up to bottom right hand corner **x2,y2** will be cleared.

Example:

\$44 \$53

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

Display is filled first with 'D', 'S' followed by clearing the area from 6,4 to 40,25.



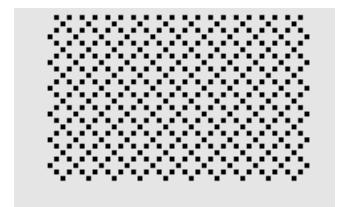
## M x1 y1 x2 y2 ptn

### Area with fill pattern

A rectangular area starting from top left hand corner **x1,y1** up to bottom right hand corner **x2,y2** is filled with pattern **ptn** honoring the preset graphics mode 'V' (set / delete / inverse / replace / inverse replace).

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

draws a cross hatch pattern 7=45° from 5,1 to 45,26.

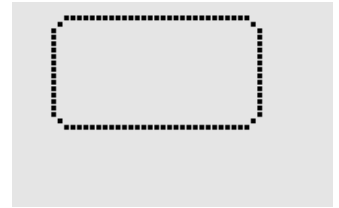


**ELECTRONIC ASSEMBLY****N x1 y1 x2 y2**

Draw a rectangle with rounded corners from top left hand corner **x1,y1** up to bottom right hand corner **x2,y2** honoring preset graphic mode 'V' (set / delete / inverse). Content of box remains unchanged. Compare drawing rectangle with rounded corners 'J'.

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

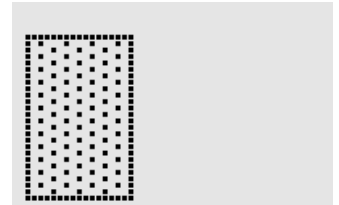
draws a box with rounded corners from coordinates 6,2 up to coordinates 38,19.

**Draw round corner****O x1 y1 x2 y2 ptn**

A rectangular box is drawn from top left hand corner **x1,y1** up to bottom right hand corner **x2,y2** filled with pattern **ptn**. Background of box will be cleared. Consult also 'R' = draw rectangular box.

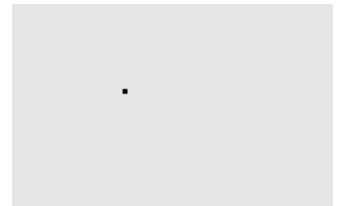
Example: \$4F \$02 \$05 \$12 \$1E \$02

draws a box from coordinates 2,5 up to coordinates 18,30 with pattern 2= 25% black.

**Draw Box****P x1 y1**

A single Pixel will be placed at coordinates **x1, y1** honoring preset graphic mode 'V' (set / delete / inverse).

Example: \$50 \$0D \$11 places a single Pixel at coordinates 17,13.

**Set Point****Q n1**

Example: \$51 \$01

switches our High-Level Graphic Controller into Power-Save-Mode. Handshake line RTS shows H-level which means that no more commands will be accepted. Supply voltage may be reduced now down to 2 Volts. Applying a wide pulse of >10 msec onto reset pin will awake the controller. Contents of registers remains unchanged, see table on page 3.

n1: 1, High-Level Graphic Controller in Power-Save-Mode (Pin LCDON remains on VDD-level: Display is ON)  
n1: 2, High-Level Graphic Controller with Display are in Power-Save-Mode (Pin LCDON at GND-level: Display is OFF).

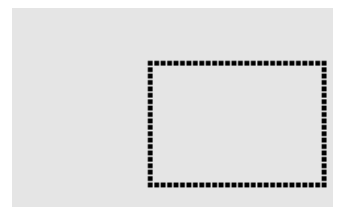
**Power Save Mode****R x1 y1 x2 y2**

A rectangle box is drawn from top left hand corner **x1,y1** up to right hand bottom corner **x2,y2** honoring the preset graphic mode 'V' (set / delete / inverse). Contents of rectangular box remains unchanged.

Compare 'O' = draw rectangular box with rounded corners.

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

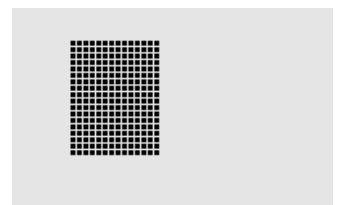
draws a rectangle from coordinates 21,8 up to coordinates 48,37.

**Draw rectangle****S x1 y1 x2 y2**

An area from top left hand corner **x1,y1** up to right hand bottom corner **x2,y2** will be filled with black pixels.

Example: \$53 \$09 \$05 \$16 \$16

fills an area from coordinates 9,5 to coordinates 22,22 with black pixels.

**Fill area**

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

Overlay combination mode **n1** and pattern **ptn** sets ASCII-characters 'A' in text mode and displays a character string when setting to 'Z'. For display a character string (command 'Z') the typing-direction must be set by: **L**=left, **R**=right, **O**=up and **U**=down.

Example: \$54 \$52 \$03 \$03

Initiates the overlay mode for text functions with gray characters (pattern 3 = 50% black), inverts the background and writes character string from left to right.

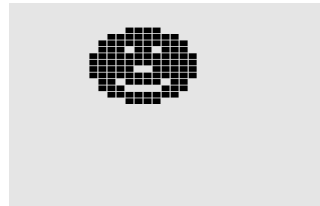
### Overlay combination mode n1:

- 1 = set: black Pixel without regarding previous value (OR)
- 2 = delete: white Pixel without regarding previous parameter
- 3 = invert: black Pixel turns to white Pixel and vice versa (EXOR)
- 4 = replace: clear background and set black colored Pixel
- 5 = invers replace: fill background and place white colored Pixel

## U x1 y1 data

An image will be loaded at coordinates **x1,y1**

- daten:**
- 1 Byte for width of image Pixels
  - 1 Byte for height of image Pixels
  - Image datas: Qty = ((height+7) / 8) \* width Bytes.
  - 1 Byte equals 8 vertical Pixel on screen;
  - 0=white, 1=black; LSB: top, MSB: bottom;
- Image builds up from left hand to right hand.



The program BMP2BLV.EXE on disk EA DISKIC-1, which is available as an accessory, generates from a monochrome Windows-Bitmap- Graphic the datas for our Image inclusive width- and height information.

Example: \$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

loads beside shown image at coordinates 9,4.

## V n1

Set overlay combination mode **n1** i.e. for following graph-functions: set dot 'P', draw straight line 'G', continue drawing straight line 'W', draw rectangular box 'R', draw box with rounded corners 'N', fill area with pattern 'M'.

Example: \$56 \$03

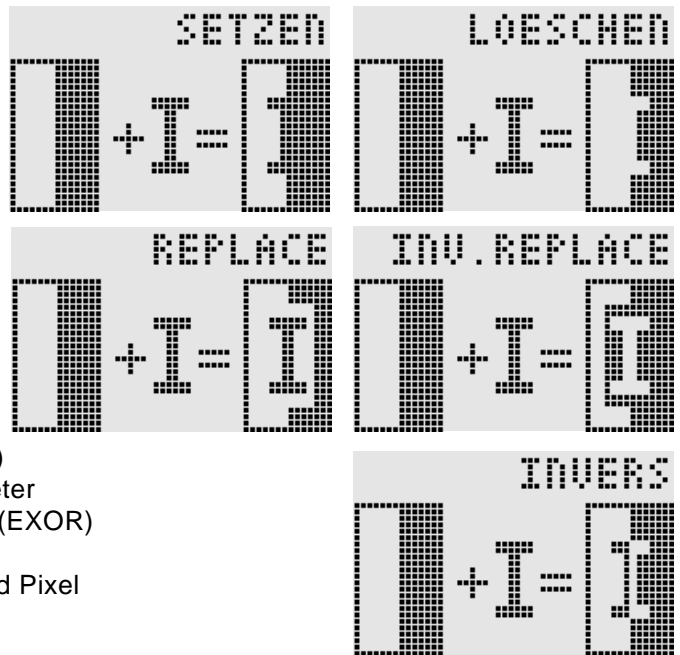
sets overlay mode to inverse.

Beside example illustrates drawing of a rectangular box in overlay modes set, clear and invert.

### Overlay combination mode n1:

- 1 = set: black Pixel without considering previous parameter (ODER)
- 2 = delete: white Pixel without regarding previous parameter
- 3 = invert: black Pixel turns to white Pixel and reversed (EXOR)
- 4 = replace: clear background and set Pixels; area with filling pattern ptn only
- 5 = invers replace: fill background, delete Pixels; area with fill pattern ptn only

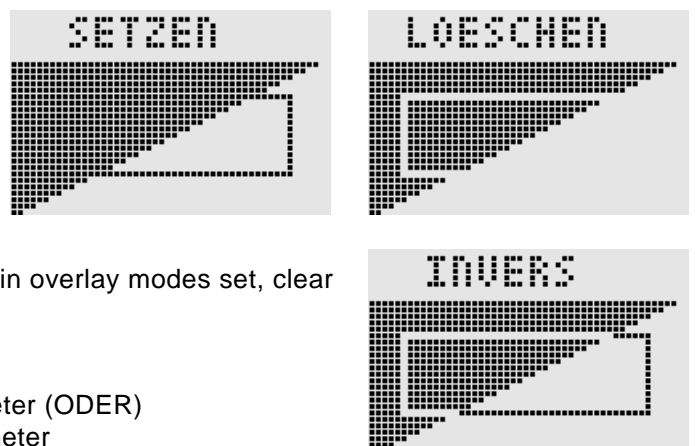
## Set Text Mode



## Upload picture

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

## Set Graphics Mode



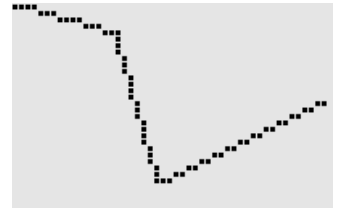
**ELECTRONIC ASSEMBLY****W x1 y1**

Draws a straight line from last drawn line end resp. (see page 3: Last xy) to **x1,y1** under considering the Graphic Mode 'V' (set / delete / inverse).

Example:

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

Draws first a straight line from 0,0 to 16,4; then continuing from 22,27 to 48,15.

**Continue straight line****X n1**

Reads in Port (**n1**: 0..7 = I/O: 0..7).

When **n1** = 8, all I/O 0..7 are read in binary; I/O 0: LSB, I/O 7: MSB

See application on page 4.

Example: \$58 \$02

reads in signal at I/O 2 and transmits L-level a \$00 resp. H-level a \$01 via RS-232

**I/O Read Port****Y n1 n2**

Changes port value (**n1**: 0..7 = I/O: 0..7) into value **n2** (0=L-level; 1=H-level; 2=invert port level).

When **n1** becomes 8, all I/O 0..7 will be transfered as binary value **n2**; I/O 0: LSB, I/O 7: MSB

See application on page 4.

Example: \$59 \$02 \$01

sets port I/O 2 to H-level

**I/O Set Port****Z x1 y1 ASCII... NUL**

Writes a character string **ASCII....** beginning from coordinate **x1,y1** honoring preset mode 'T' (set / delete / inverse / replace / inverse replace / pattern / direction).

Character strings must be finished with **NUL** (\$00). Origin (0,0) starts in top left hand corner of display. Datas of coordinates refer to top left hand corner of character. Note: Font Nr. 1 has capital letters only.

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

writes character string "Test" starting at coordinate 6,11.

Font: 8x16 with normal height and width.

Text mode: Writing direction is from right hand, Overlay mode is Replace with Pattern Black.

**Write character string****?****Query display type**

Resolution of display and way of building up an image will be displayed. High-Level-Graphics Controller IC202-PGH features a permanent resolution of 128 x 64 Pixel and organizes image building up in vertical direction. This command is available for communication of external programmes with our High-Level-Graphics Controller. IC6963-PGH is designed for displays with i.e. Toshiba controller managing variabel resolutions up to 240x128 Pixel with horizontal image building up organisation.

Example: \$3F

This command transmits first the resolution X- (128) and Y- (64) and then the way of image building up ('V') in vertical organization via RS-232 interface.

## INTEGRATED FONTS

High-Level Graphic Controller EA IC202 incorporates 3 integrated character sets (Font1: 4x6 Pixel; Font2: 6x8 Pixel and Font 3 8x16 Pixel). Each character set can be used in 1-, 2-, 3- or 4-times height. Independant of height the width may be doubled, tripled or quadrupled. 4-16 self defined characters will be stored as long as the supply voltage is switched on, (see command '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

+ 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)	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	
\$B0 (dez: 176)	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
\$C0 (dez: 192)	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!	!
\$D0 (dez: 208)	#	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
\$E0 (dez: 224)	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	
\$F0 (dez: 240)	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?

Each individual character can be placed exactly by pixel. You may mix text with graphics in any way at your desire. Also characters with different sizes can be used in common.

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

## ELECTRONIC ASSEMBLY

### EA GE120-5NV24 GRAPHIC UNIT 120x32, 2 FONTS, RS-232

<b>Dimensions</b>	68 x 39 x 11mm
<b>Display</b>	120x32, Supertwist, LED-Backlight
<b>Supply Voltage</b>	+5V / 100mA
<b>Terminals</b>	10 Pins, 2,54mm Grid
<b>Operating Temperature</b>	0 ... +70°C
<b>Storage Temperature</b>	-20 ... +70°C



- \* 120x32 Pixel Supertwist-LCD with green color LED-Backlight
- \* 2 FONTS from 3,2mm (5x30 Characters) to 4,6mm (4x20) up to 18,2 mm (1x5)
- \* DRAW STRAIGHT LINES, INVERT OR DELTE AREAS
- \* SUPPLY VOLTAGE: +5V / typ. 100mA
- \* BAUDRATES 1200,2400,4800,9600,19200BD

#### Ordering Information

Graphicunit 120x64 with LED-Backlight  
Cable with 9-pin D-SUB plug (female)

**EA GE120-5NV24**  
**EA KV24-9B**

### EA GE128-6N3V24 GRAPHIC UNIT 128x64, 3 FONTS, RS-232C

Graphic Device EA GE128-6N3V24 is based on High-Level-Graphic Controller EA IC202-PGH. This Graphic Device operates either on real RS-232C levels ( $\pm 10$ Volts) or with CMOS- levels.

<b>Dimensions</b>	84 x 60 x 25mm
<b>Display</b>	128x64, Supertwist, LED-Backlight
<b>Supply Voltage</b>	+5V / 150mA
<b>Terminals</b>	10 Pins, 2,54mm Grid
<b>Operating Temperature</b>	0 ... +70°C
<b>Storage Temperature</b>	-20 ... +70°C



- \* 128x64 Pixel Supertwist-LCD with green color LED-Backlight
- \* 3 FONTS: 2,6mm (10x32 Characters); 6,2mm (4x16) and 25 mm (1x5)
- \* DRAW STRAIGHT LINES, INVERT OR DELETE AREAS
- \* AUTOMATIC BARGRAPH SCALING FUNCTION
- \* SUPPLY VOLTAGE: +5V / typ. 150mA
- \* BAUDRATES 1200,2400,4800,9600,19200BD

#### Ordering Information

Graphicunit 128x64 with LED-Backlight  
Cable with 9-pin D-SUB plug (female)

**EA GE128-6N3V24**  
**EA KV24-9B**

### EA GE128-7KV24 GRAPHIC UNIT 128x128, 3 FONTS, RS-232

<b>Dimensions</b>	85 x 100 x 25mm
<b>Display</b>	128x128, Supertwist, LED-Backlight
<b>Supply Voltage</b>	+5V / 500mA
<b>Terminals</b>	10 Pins, 2,54mm Grid
<b>Operating Temperature</b>	0 ... +50°C
<b>Storage Temperature</b>	-20 ... +70°C



- \* 128x128 Pixel Supertwist-LCD with green color LED-Backlight
- \* 3 FONTS from 2,2mm (21x32 Characters) to 5,2mm (8x16) up to 21 mm (2x5)
- \* DRAW STRAIGHT LINES, INVERT OR DELETE AREAS
- \* AUTOMATIC BARGRAPH SCALING FUNCTION
- \* SUPPLY VOLTAGE: +5V / typ. 500mA
- \* BAUD RATES 1200,2400,4800,9600,19200BD

#### Ordering Information

Graphicunit 128x128 with LED-Backlight  
Cable with 9-pin D-SUB plug (female)

**EA GE128-7KV24**  
**EA KV24-9B**

More Display Sizes are available: i.e. 240x64 and 240x128 (as of Dec `97).