

GRAPHICS UNIT 120x32

INCL. 2 FONTS, ZOOM AND LED-BACKLIGHT

Mixing Text+Graphics

Pattern

Font Zoom

Image Download



Size:
68x39x16 mm

FEATURES

- * 2 DIFFERENT FONTS BUILT IN
- * ZOOM FUNCTION FOR ALL FONTS (2x, 3x AND 4x)
- * EASY PROGRAMMING OF VARIOUS BUILT IN GRAPH FUNCTIONS:
- * STRAIGHT LINE, DOT, AREA, AND/OR/EXOR, BAR GRAPH, PATTERNS...
- * MIXING TEXT AND GRAPHICS
- * 4-16 FREELY DEFINABLE CHARACTERS (DEPENDING ON SIZE)
- * INPUT ON RS-232 / CMOS-LEVEL
- * PROGRAMMABLE BAUDRATES FROM 300 UP TO 9,600 BAUD
- * NO TIMING PROBLEMS WITH FAST BUS SYSTEMS
- * 8 DIGITAL I/O'S FREELY AVAILABLE FOR CUSTOM DESIGNS
- * +5V / typ. 100mA
- * DOWNLOAD OF CONVERTED WINDOWS-BMP GRAPHICS

ACCESSORIES

- * PC DISK WITH SOFTWARE TO CONVERT WINDOWS-BMP PICTURES:
EA DISKIC1

ORDERING INFORMATION

GRAPHICS DEVICE 120x32, 3 FONTS, RS-232
3½" DISK WITH SOFTWARE TO CONVERT
WINDOWS BMP-FILES

EA GE120-5NV24

EA DISKIC1

**ELECTRONIC
ASSEMBLY** GMBH

ZEPPELINSTRASSE 19 · D-82205 GILCHING
PHONE +49-8105-778090 · FAX +49-8105-778099 · <http://www.lcd-module.de>

GENERAL

Our graphics unit LCD EA GE120-5NV24 is designed for small up to medium quantities. Nearly all known processor systems can be connected within a few hours work to an attractive and informative looking display because of it's simple way to program, it's small outside dimensions and it's excellent supertwist contrast. Input ports accept a serial asynchronous RS-232 interface. The display includes complete graphics routines and various character sizes to drive the display.

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

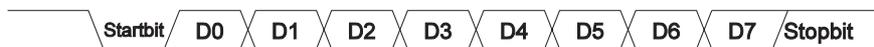
- no timing problems with fast processor bus
- enough memory space (operating memory and character set memory especially for μC)
- no time consuming graphic calculations which would slow down processor speed

Drivers, decoders or port modules are not required. Display control can be made in simple cases through a single RxD line only.

HARDWARE

Supply voltage of system is +5 Volts. Data transfer is asynchronous serial in RS-232 format at CMOS level with true RS-232 level ($\pm 10\text{V}$) or with 5V CMOS level. Data format is set firmly to 8 databits, 1 stop bit and no parity. Baud rate can be selected on 2 solder straps from 300 Baud up to 9,600 Baud. Handshake lines RTS and CTS are available also. On small amounts of datas there is no interpretation required.

Data format:



Additional 8 I/O-ports are available on J2 eyelet strip for freely usage. This may be wired on your individual desire. Each output port drives a load of max. -10mA.

SOFTWARE

Programming of this graphics unit is performed by commands like i.e. "plot a rectangular box from (0,0) to (20,15) which origins in top left hand corner of display. Therefore the serial interface has to transmit this sequence of bytes: \$52 \$00 \$00 \$14 \$0F. Character strings can be placed exactly to the pixel. Mixing of graphics images with text elements is possible anytime. Two different character sets are available where each of them can be zoomed 2x, 3x and 4x.



ELECTRONIC ASSEMBLY

BUILT IN FONTS

Graphics unit EA GE120-5NV24 contents two integrated character sets (font1: 4x6 pixel; font2: 6x8 pixel). Each character set can be used in 1-, 2-, 3- or 4-times height. Their width can be doubled, tripled or quadrupled, independent of height. In addition you can define 4-16 characters on your own which remain alive as long as power is 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

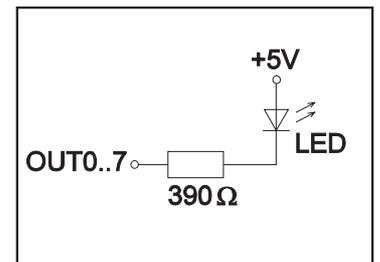
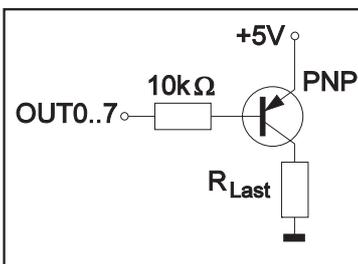
+ Lower Upper	\$0 (0)	\$1 (1)	\$2 (2)	\$3 (3)	\$4 (4)	\$5 (5)	\$6 (6)	\$7 (7)	\$8 (8)	\$9 (9)	\$A (10)	\$B (11)	\$C (12)	\$D (13)	\$E (14)	\$F (15)
\$20 (dez: 32)		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
\$30 (dez: 48)	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
\$40 (dez: 64)	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
\$50 (dez: 80)	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)									¡	¢	£	¤	¥	¦	§	¨
\$90 (dez: 144)	©	ª	«	¬	­	®	¯	°	±	²	³	´	µ	¶	·	¸

Each individual character can be placed precisely to the pixel. You may mix text with graphics in any way at your desire. Several different character sizes can also be displayed together.

Font 2

DIGITAL OUTPUT PORTS OUT 0..7 (CONNECTOR J2)

Command "Y n1 n2"¹⁾ sets any selected pin OUT 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 practicable i.e. to 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.



OVERVIEW OF ALL GRAPHICS FUNCTIONS

This graphics unit can be programmed by a number of built in commands. Each command starts with a command letter and will be extended by several parameters.

Command table EA GE120-5NV24							
Command						Remarks	
Functions for outputting text							
Text mode	T	R	n1	ptn		Write character string; n1: overlay combination mode for text output 1=set; 2=delete; 3=inverse; 4=replace; 5=inverse repl.; ptn: use pattern no. 0..7;	
Set font	F	n1	n2	n3		Set font no. n1; n1=1:4x6 font; n1=2:6x8 font; n1=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	
Graphics commands with overlay mode							
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
Area with fill pattern	M	x1	y1	x2	y2	ptn	Draw area with pattern ptn (0..7); x1,y1,x2,y2 = corner points
Other graphics commands							
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)
Upload picture area	U	x1	y1	data ...			Load a picture area to x1,y1; see picture structure for picture data
Control / definition commands							
Adjust display contrast	K	n1					n1=0..20 Contrast adustment; 0= low, 20 high
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)

ELECTRONIC ASSEMBLY

PARAMETER

Various built in commands can be used to program the high-level graphics controller. Each command starts with a command letter which is followed by a number of parameters. All commands with parameters, coordinates and other hand over datas are expected in form of Bytes. No space blanks are allowed, i.e. no space bars, no commas. End of command **does not need a closing byte** such as a Carriage 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, acceptable values are 0..119 for x- resp. 0..31 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

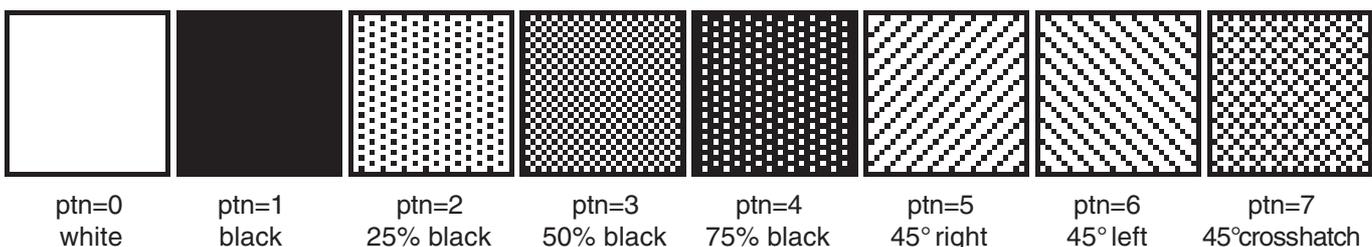
Below table shows the character string "Test" which is displayed at coordinates 7,3.

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%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 setting of pattern type parameters (ptn = 0..7). They are used e.g. for rectangular areas, bargraphs and even text lines with various pattern.

This pattern are available:



DESCRIPTION OF INDIVIDUAL GRAPH FUNCTIONS

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 transfered Bytes all in Hex codes.

A x1 y1 n1

A character **n1** will be displayed on coordinates **x1,y1** with preset font 'F' and text mode 'T' (set / delete / invert / 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.

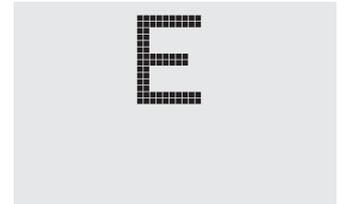
Example: \$41 \$13 \$02 \$45

Character 'E' will be displayed at coordinates 19,2

Preset font: 6x8, with double width and double height

Text mode: Replace and black pattern

Set ASCII-Characters



D L/I/S

The entire content of the display will be **L**=deleted (white), **I**=inverted, or **S**=filled (black).

Example: \$44 \$49

This will invert the entire content of the display.

Display command

E n1 data

You can define up to 8 characters yourself (depending on size of font). These characters will then have the ASCII codes 1 to 8, and will remain in an intenal RAM area.

Note: One ASCII code no. will define one font size only.

Define characters

Example 1:

Commands

\$45 \$03

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

define an up arrow for ASCII no. 3.

Example 2:

Commands

\$45 \$02

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

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

define a down arrow for ASCII no. 2, using the character set 8x16.

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

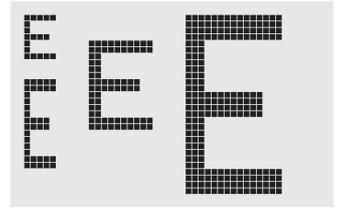
		BYTE NR.							
		1	2	3	4	5	6	7	8
Bit 0									
Bit 1									
Bit 2									
Bit 3									
Bit 4									
Bit 5									
Bit 6									
Bit 7									
Bit 0									
Bit 1									
Bit 2									
Bit 3									
Bit 4									
Bit 5									
Bit 6									
Bit 7									
		9	10	11	12	13	14	15	16
	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

ELECTRONIC ASSEMBLY**F n1 n2 n3**

Font no. **n1** (1=4x6 capital letters only; 2=6x8) will be set. In addition, a zoom factor (1..4 times) for the width **n2** and the height **n3** will be set separately.

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

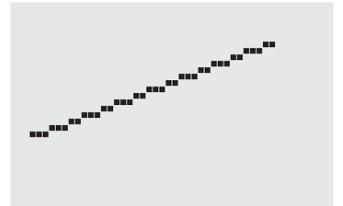
The 6x8 font with the width enlarged three times and the height enlarged four times. In the hardcopy alongside the character 'E' from the 6x8 font is shown with different zoom factors.

Set font**G x1 y1 x2 y2**

A straight line will be drawn from coordinates **x1,y1** to coordinates **x2,y2**, taking into account the graphics mode 'V' that has been set (set / delete / inverse).

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

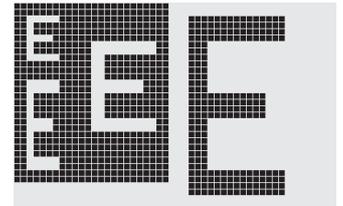
A straight line will be drawn from 3,20 to 50,6.

Draw straight line**I x1 y1 x2 y2**

The area extending from the top left-hand corner **x1,y1** to the bottom right-hand corner **x2,y2** will be inverted (black pixels will become white, and vice versa).

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

This will invert the area extending from 0,0 to 23,27 while the content of the display is shown.

Invert area**K n1****Set contrast of display**

Command K n1 changes contrast of display. Standard parameter value after power on is n1=8; the display gets darker with n1=9..20; contrast of display is reduced with n1=0..7.

Example: \$4B \$14

Sets contrast to maximum value 20, display is "black".

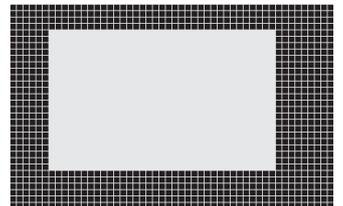
L x1 y1 x2 y2**Clear specified display area**

The area extending from the top left-hand corner **x1,y1** to the bottom right-hand corner **x2,y2** will be deleted.

Example: \$44 \$53

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

To begin with, the display will be filled with 'D', 'S', and then the area extending from 6,4 to 40,25 will be deleted.



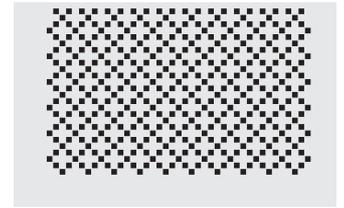
M x1 y1 x2 y2 ptn

A rectangular area will be drawn from the top left-hand corner **x1,y1** to the bottom right-hand corner **x2,y2**, with the pattern **ptn** and taking into account the graphics mode 'V' that has been set (set / delete / invert / replace / inverse replace).

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

This will draw the pattern 7=45°cross from 5,1 to 45,26.

Area with fill pattern



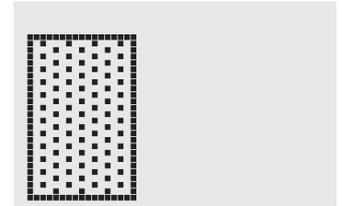
O x1 y1 x2 y2 ptn

A rectangle will be drawn from the top left-hand corner **x1,y1** to the bottom right-hand corner **x2,y2**, with the pattern **ptn**. The background of the box will be deleted. Compare 'R, Draw rectangular'. Note, that minimum box size must be 3 pixel wide and 3 pixel in height.

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

This will draw a box from coordinates 2,5 to 18,30, with the pattern 2=25% black.

Draw box



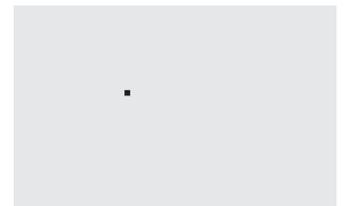
P x1 y1

A single pixel will be placed at coordinates **x1,y1**, taking into account the graphics mode 'V' that has been set (set / delete / invert).

Example: \$50 \$OD \$11

This will set the pixel at coordinates 17,13.

Place a dot



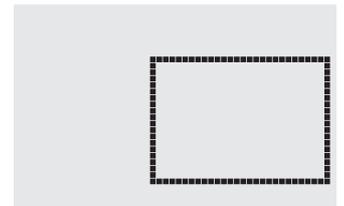
R x1 y1 x2 y2

This draws a rectangular frame from the top left-hand corner **x1,y1** to the bottom right-hand corner **x2,y2**, taking into account the graphics mode 'V' that has been set (set / delete / inverse). The contents of the rectangle will not be altered in this procedure. . Compare 'O, Draw box'.

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

This will draw a rectangle from position 21,8 to 48,37.

Draw rectangular



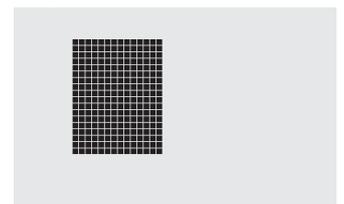
S x1 y1 x2 y2

The area extending from the top left-hand corner **x1,y1** to the bottom right-hand corner **x2,y2** will be filled with black pixels.

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

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

Fill area



ELECTRONIC ASSEMBLY

T R n1 ptn

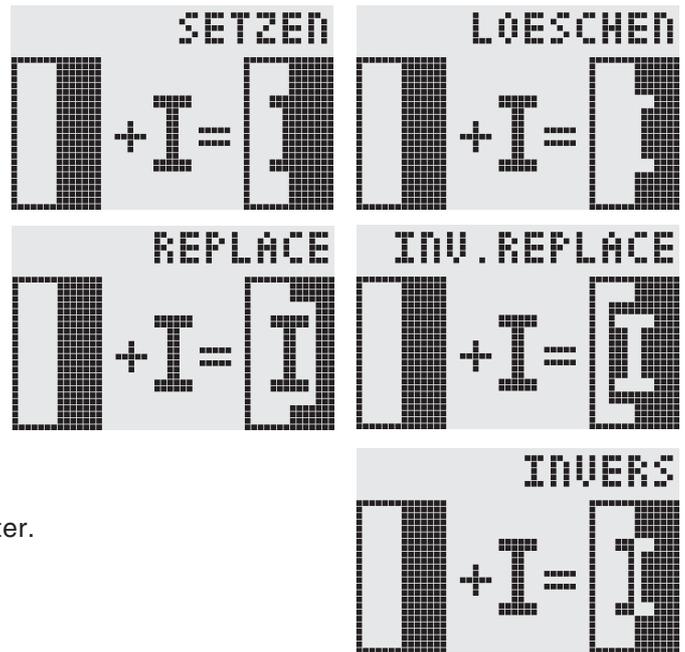
The link mode **n1**, and pattern mode **ptn** is set for both text-functions: set ASCII- characters (command 'A') and display character string (command 'Z').

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

This will set the overlay combination mode for all of the following text functions to gray characters (pattern 3 = 50% black) and inverts the background

Overlay combination mode n1:

- 1 = set: Blackpixels, irrespective of the previous value (OR).
- 2 = delete: White pixels, irrespective of the previous parameter.
- 3 = inverse: Black pixels become white, and vice versa (EXOR).
- 4 = replace: Delete background, and set black character.
- 5 = inverse replace: Fill background, and set white character.

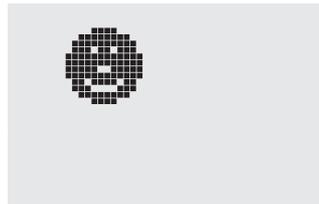


Set text mode

U x1 y1 data

An image will be loaded to coordinates **x1,y1**.

- Datas:**
- 1 byte for width of image in pixels.
 - 1 byte for height of image in pixels.
 - Image data: pixel qty = ((height+7) / 8) x width bytes.
- 1 byte stands for 8 vertical pixels on the screen;
0=white, 1=black; LSB: top, MSB:bottom.

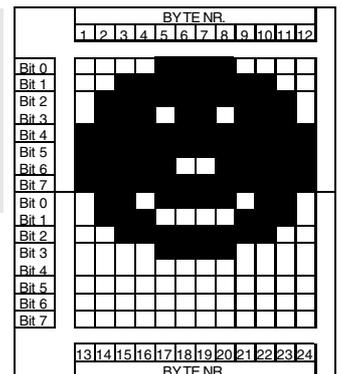


The image builds up from left hand side to right hand side.
Programme BMP2BLV.EXE generates the image data - including details of width and height - from monochrome Windows bitmap graphics.

Example: \$55 \$09 \$04 \$00 \$00
\$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 nearby shown image to coordinates 9.4.

Upload image



V n1

This sets the overlay combination mode **n1** for the following graphics functions: set point ('P'), draw straight line ('G'), continue drawing straight line ('W'), draw rectangle ('R'), fill area with pattern ('M').

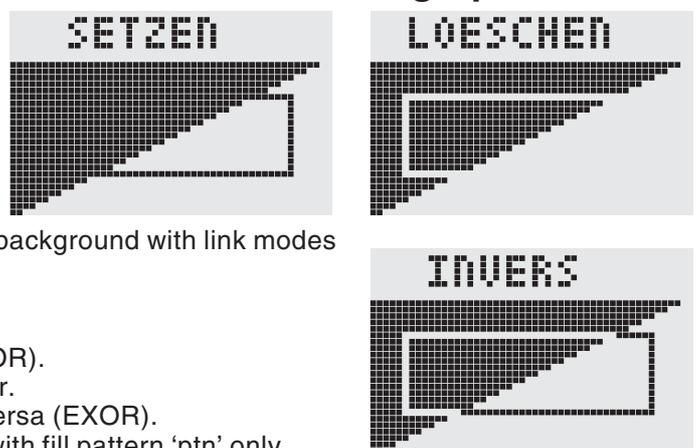
Example: \$56 \$03

This will set the overlay combination mode to inverse.

As an example, a rectangle is drawn here on an existing background with link modes set, delete, and inverse.

Overlay combination mode n1:

- 1=set: Black pixels, irrespective of previous parameter (OR).
- 2=delete: White pixels, irrespective of previous parameter.
- 3=inverse: Black pixels are changed to white, and vice versa (EXOR).
- 4=replace: Clear background and set pixels inside area with fill pattern 'ptn' only.
- 5=inverse replace: Fill background, delete pixels from area with fill pattern 'ptn' only.



Set graphics mode

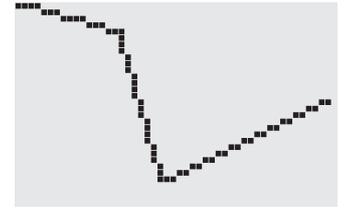
W x1 y1

Continue straight line

This continues a straight line, from the last placed dot or the end of the line last drawn (see page 3 Draw straight line) to **x1,y1**, taking into account the graphics mode 'V' that has been set.

Example:

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



Draws first a straight line from 0,0 to 16,4. It will then be continued to 22,27 and to 48,15.

Y n1 n2

Set output ports

This changes the port (**n1**: 0..7 = OUT: 0..7) to parameter **n2** (0=L level; 1=H level; 2=invert port).
If **n1**=8, all output ports OUT 0..7 will be output as a binary value **n2**; OUT 0: LSB, OUT 7: MSB.
See application on page 3.

Example: \$59 \$02 \$01

This will switch the port OUT 2 to H-level.

Z x1 y1 ASCII... NUL

Write character string

This writes the character string **ASCII...** to coordinates **x1,y1**, taking into account the text mode 'T' that has been set (set / delete / invert / replace / inverse replace / fill pattern / direction). The character string must be terminated with **NUL** (\$00). The origin (0,0) is at top left-hand corner of display. Datas of coordinates refer to top left-hand corner of character.

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



This will write the character string "Test" starting at coordinates 6,11.

Font set is: 8x16 with normal width and height.

Text mode: Write from left to right, overlay combination replace with black pattern.

EA GE120-5NV24

ELECTRONIC ASSEMBLY

EA GE128-6N3V24 GRAPHICS UNIT 128x64, 3 FONTS, RS-232

Size	84 x 60 x 25mm
Display	128x64, Supertwist, LED- illuminated
Supply voltage	+5V / 150mA
Pinning	10 pins, grid 2,54mm
Operating temperature	0 ... +70°C ambient
Storage temperature	-20 ... +70°C

new



- * 128x64 PIXEL SUPERTWIST-LCD WITH GREEN LED-IllumINATION
- * 3 FONTS FROM 2,6mm (10x32 CHARACTERS) TO 6,2mm (4x16) UP TO 25 mm (1x5)
- * DRAW STRAIGHT LINES, DELETE AREAS AND INVERT
- * AUTOMATIC SCALING FUNCTION FOR BAR GRAPH
- * SUPPLY VOLTAGE: +5V / typ. 150mA
- * BAUD RATES 1200,2400,4800,9600,19200BD

Ordering information:

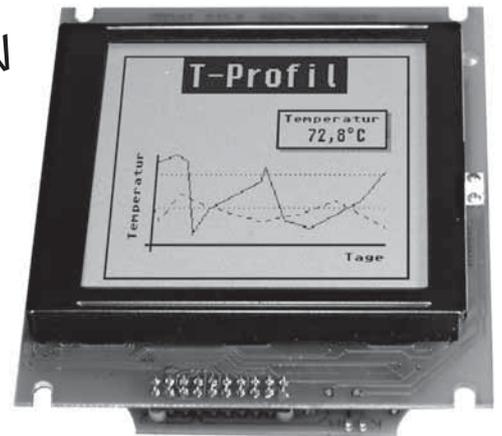
Graphics unit 128x64 with LED-Illumin.
Cable with 9-pin D-SUB plug (female)

EA GE128-6N3V24
EA KV24-9B

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

Size	85 x 100 x 25mm
Display	128x128, Supertwist, LED- illuminated
Supply voltage	+5V / 500mA
Pinning	10 pins, grid 2,54mm
Operating temperature	0 ... +50°C ambient
Storage temperature	-20 ... +70°C

new



- * 128x128 PIXEL SUPERTWIST-LCD WITH GREEN LED-IllumINATION
- * 3 FONTS FROM 2,2mm (21x32 CHARACTERS) TO 5,2mm (8x16) UP TO 21 mm (2x5)
- * DRAW STRAIGHT LINES, DELETE AREAS AND INVERT
- * AUTOMATIC SCALING FUNCTION FOR BAR GRAPH
- * SUPPLY VOLTAGE: +5V / typ. 500mA
- * BAUD RATES 1200,2400,4800,9600,19200BD

Ordering information:

Graphics unit 128x128 with LED-Illumin.
Cable with 9-pin D-SUB plug (female)

EA GE128-7KV24
EA KV24-9B

EA GE240-6K2V24 GRAPHICS UNIT 240x64, 3 FONTS, RS-232

Size	180 x 65 x 25mm
Display	240x64, STN, LED- illum.
Supply voltage	+5V / 500mA
Pinning	10 pins, grid 2,54mm
Operating temperature	0 ... +50°C ambient
Storage temperature	-20 ... +70°C

new



- * 240x64 PIXEL SUPERTWIST-LCD WITH GREEN LED-IllumINATION
- * 3 FONTS FROM 2,6mm (10x60 CHARACTERS) TO 6,4mm (4x21) UP TO 25 mm (1x10)
- * DRAW STRAIGHT LINES, DELETE AREAS AND INVERT
- * SUPPLY VOLTAGE: +5V / typ. 500mA
- * BAUD RATES 1200,2400,4800,9600,19200BD
- * AUTOMATIC SCALING FUNCTION FOR BAR GRAPH

Ordering information:

Graphics unit 240x64 with LED-Illumin.
Cable with 9-pin D-SUB plug (female)
Front bezel (window 131,0 x 38,0mm)

EA GE240-6K2V24
EA KV24-9B
EA 017-10UKE

EA GE120-5NV24

BAUD RATES

Baud rate is set on 2 solder bridges *Baud 0* and *Baud 1*. Factory setting is 9,600 Baud (Baud 0 shorten). Data format is fixed to 8 data bits, 1 stop bit, no parity. Please note that the internal data buffer stores 20 bytes only. Make sure that the handshake line RTS is queried when transmitting higher data volumes; +10V level: accept datas; -10V level: display is busy (hi). When level-solder bridges are closed in lo- direction than pin RTS goes to 0V for Ready and to +5V for Busy instead.

Baud rates		
Solder Stripes		Data format 8,N,1
Baud 0	Baud 1	
short	short	1200
cut	short	2400
short	cut	9600
cut	cut	19200

PINNING

Supply voltage (+5V) and RS-232 datas are fed into the 10 pin strip. The display expects "true" RS-232 level ($\pm 12V$) in case that both level solder bridges *hi* are closed to the top (factory setting, see drawing). When solder bridges are closed to bottom (*lo*), the display can be driven directly by an uC output port (5V- level).

RS-232 Connector J1			
Pin	Symbol	In/Out	Function
1	VDD	-	+ 5V Supply
2	DCD	-	Connection to DTR
3	DSR	-	Connection to DTR
4	TxD	Out	Transmit Data
5	CTS	In	Clear To Send
6	RxD	In	Receive Data
7	RTS	Out	Request To Send
8	DTR	-	Connection to Pin 2&3
9	NC	-	not connected
10	GND	-	0V Masse

Connector J2			
Pin	Symbo	In/Out	Function
1	VDD	-	+ 5V Supply
2..9	I/O0..7	In/Out	In-/Output
10	GND	-	0V Ground

Solderpad strip J2 carries digital output signals OUT 0 .. OUT 7 which can be used for special purposes by the customer. See page 3 for details.



DIMENSIONS

all sizes in mm

