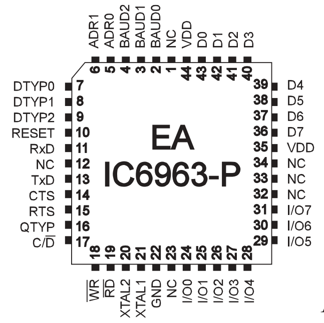
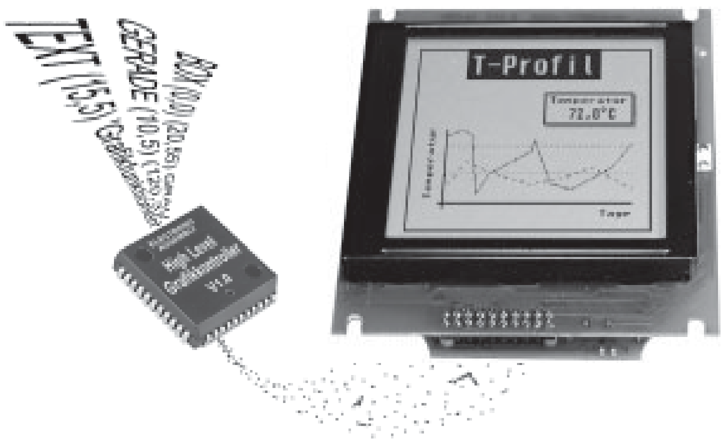
8.1.2002

EA IC6963

HIGH-LEVEL GRAPHICS CONTROLLER FOR DISPLAYS WITH T6963



PLCC44J



TEXT (15,5) Grafikkontroller
GERADE (10,5) (120,30)
High Level Grafikkontroller V1.0
T-Profil
Temperatur 72.0°C

FEATURES

* LCD GRAPHICS FOR DISPLAYS WITH T6963 I.E. 240x64, 240x128, 128x128
* NO TIMING PROBLEMS WITH FAST BUS SYSTEM
* POSITIONING ACCURATE TO THE PIXEL WITH ALL FUNCTIONS
* PROGRAMMING BY MEANS OF COMMANDS SIMILAR TO HIGH-LEVEL LANGUAGE
* STRAIGHT LINE, POINT, AREA, AND/OR/EXOR, BAR GRAPH...
* 3 DIFFERENT FONTS BUILT IN
* ZOOM FUNCTION FOR ALL FONTS (FROM 2x UP TO 8x)
* UP TO 21 FREE DEFINABLE CHARACTERS
* COMBINATIONS OF TEXT AND GRAPHICS
* 6 CLIPBOARD FUNCTIONS, CURSOR FUNCTIONS
* INPUT ON RS-232 WITH 5V-CMOS-LEVEL
* DIRECT ACCESS TO MAX232 OR EQUIVALENT
* BAUDRATES 1200, 2400, 9600 UP TO 115,200 BAUD
* DOES NOT SLOW DOWN PROCESSOR SPEED
* ONLY MAX. 4 EXTERNAL COMPONENTS REQUIRED
* 8 DIGITAL I/O-PORTS FREELY AVAILABLE FOR CUSTOM DESIGNS

ORDERING INFORMATION

HIGH-LEVEL GRAPHICS CONTROLLER FOR LCD WITH T6963	**EA IC6963-PGH**
GRAPHIC DISPLAY WITH 240x128 PIXEL, CFL-ILLUMINATION	**EA P240-7KC**
GRAPHIC DISPLAY WITH 240x128 PIXEL, LED- ILLUMINATION	**EA P240-7KLED**
GRAPHIC DISPLAY WITH 128x128 PIXEL, LED-ILLUMINATION	**EA P128-7KLED**
GRAPHIC DISPLAY WITH 240x 64 PIXEL, LED-ILLUMINATION	**EA P240-6K2LED**
CONVERTER PROGRAM FOR BMP-IMAGES (PC-DOS,GERMAN)	**EA DISKIC-1**

ELECTRONIC ASSEMBLY GMBH
LOCHHAMER SCHLAG 17 · D-82166 GRÄFELFING
PHONE +49-89-8 54 19 91 · FAX +49-89-8 54 17 21

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 graphic controller includes complete graph routines and various character sizes.

Programming is made by high level programming language with 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:

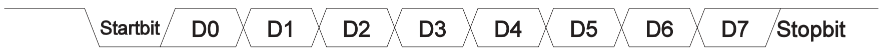
- no timing problems with fast processor bus
- enough memory space (operating memory and characterset memory especially for µC)
- no time consuming graphic calculations which would slow down processor speed

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 8 data bits firm, 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. On small amounts of datas there is no interpretation required (data buffer of 56 byte is built in).

Data format:



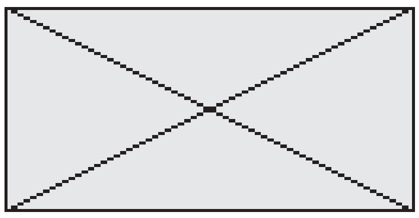
Additional 8 I/O-ports are available for freely usage. This may be wired as inputs or outputs on individual desire. Possible application is switching of backpanel illumination or reading in several keys.

SOFTWARE

Programming of this High-Level Graphics Controller is performed by commands like i.e. "plot 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. Character strings can be placed exactly 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 from 2x up to 8x. The biggest character set 8x16 allows when using 8-x zoom (=64x128) a totally filled display with letters and numbers.

TESTMODE

As long as pin 15 (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 15 (RTS) to GND is removed, the Graphcontroller returns to normal operation mode but testbox still remains visible.

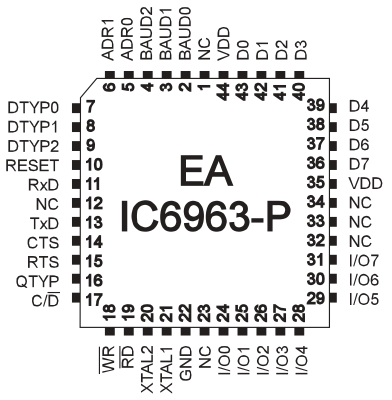


TECHNICAL DESCRIPTION

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

Datas are valid for T_a = 0..+70°C und VDD= 5,0V ±20% if not noted otherwise.

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
7,8,9	DTYP	In	lo	Display size
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	RTS	Out	lo	lo: RS-232 Data receive enable; hi: RS-232 Data receive disable
16	QTYP	In	hi/lo	Crystal type, lo=11.059 MHz, hi=18.432 MHz
17	C/D	Out		Display: hi: Data; lo: Command
18	WR	Out	hi/lo	Display: Data/Command write
19	RD	Out	hi/lo	Display: Data/Command read
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..31	IO0..7	I/O	hi/lo	8 Inputs/Outputs
32,33,34	NC			do not connect
35	VDD	VDD	hi	Power supply +5V
36..43	D7..0	I/O	hi/lo	Display: Data lines
44	VDD	VDD	hi	Power supply +5V

BAUDRATES

Various Baudrates can be set depending on System clock (Quartz, Ceramic resonator) for RS-232 data transfer. Setting is made by connecting pin BAUD0..2 and QTYP with VDD or GND-level. Programmable Baudrates are listed in beside table (0:GND, 1:VDD).

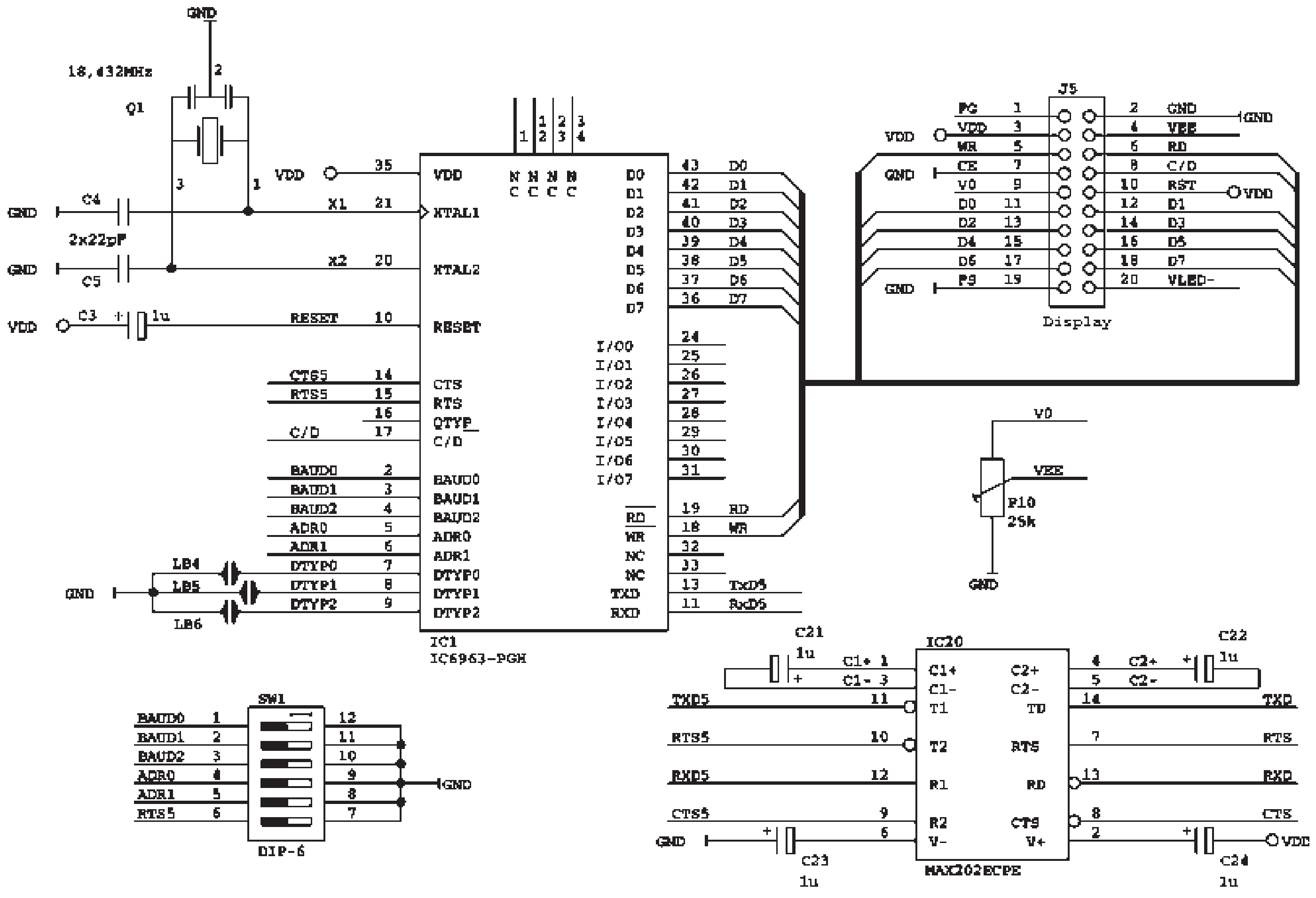
Baudrate				
Baud 2	Baud 1	Baud 0	QTYP = 0 11.0592 MHz	QTYP = 1 18.432 MHz
0	0	0	1200	1200
0	0	1	2400	2400
0	1	0	4800	4800
0	1	1	9600	9600
1	0	0	19200	19200
1	0	1	38400	38400
1	1	0	57600	57600
1	1	1	115200	115200

TYPE OF DISPLAYS

Seven standard display types can be selected by connecting pins DTYP0..2 with VDD- or GND-level (0: GND, 1: VDD). Please choose our ELECTRONIC ASSEMBLY displays from this table. Displays with other sizes can be set by command '!' (see page 16).

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

APPLICATION EXAMPLE



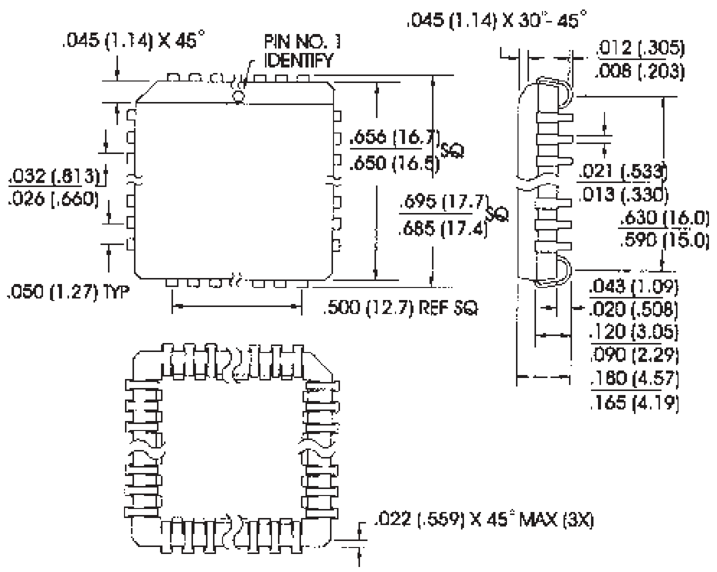
APPLICATION HINTS

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

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

DIMENSIONS EA IC6963-PGH

Housing: PLCC44J; all sizes in Inches (mm)



DEFAULT SETTINGS

Default settings		
Register	**comman**	**after power-on / reset**
Text mode	T	right, set, black
Graphics mode	V	set
Font	F	6x8
Font factor width/height	F	1/1
Last xy	W	(0;0)
Self-defined character	E	undefined
Bar graph 1..8	B	undefined
High-level graphics controller	K	selected
Flashing area	QD	(0;0)
Flashing mode	QC	inverse
Flashing time	QZ	0.6 sec.
Clipboard	C	empty
Inputs / outputs I/O0..7	Y	H level

ATTENTION

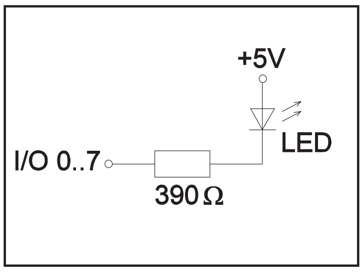
handling precautions!

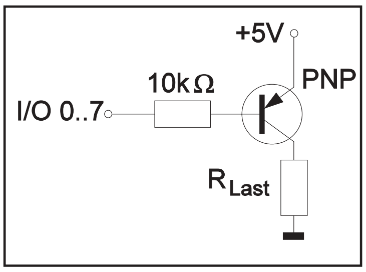
DIGITAL INPUTS AND OUTPUTS IO 0..7

8 pins of this High Level Graphics Controller may be used as freely available programmable inputs and outputs. Also mixed operation of i.e. 3 outputs and 5 inputs is possible.

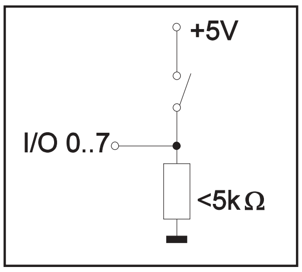
Output circuit

Command "Y n1 n2"[1)]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 practicable 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.

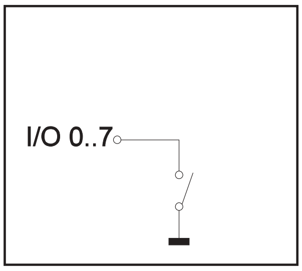




Input circuit



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"[1)] 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 filter circuit for contact jitter.



[1)] *Command descriptions you will find on page 15.*

EA IC6963

BUILT IN FONTS

EA IC 6963 high level graphic controller incorporates three integrated character sets. They can be used at their normal height or can be increased up to eight times, while their width can be increased from two to eight times, independent of the height.

+ 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: 4x6

+ 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)	Ç	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	ì	Ä	Å
$90 (dez: 144)	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	¢	£	¥	ß	

Font 2: 6x8

In addition, you can define up to twenty-one characters on your own, depending on the font being used. These characters will remain until the supply voltage is switched off (see command 'E').

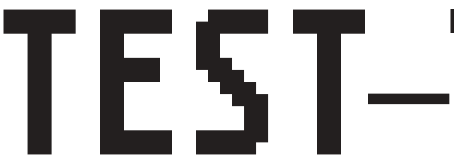
Each individual character can be placed precisely to the pixel. You may mix text with graphics in any way at your desire. Several different character sizes can also be displayed together.

+ 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)	Ç	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	ì	Ä	Å
$90 (dez: 144)	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	¢	£	¥	ß	

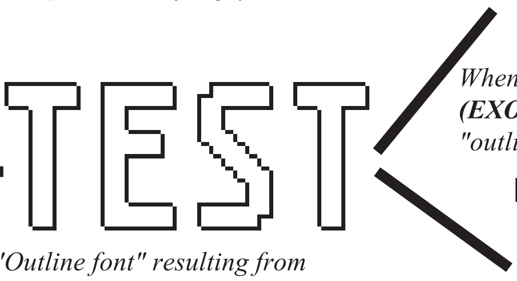
Font 3: 8x16

HINT: SPECIAL EFFECTS

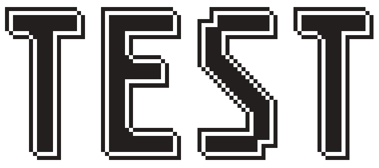
With large fonts, command 'T' TEXT-mode (link, pattern) allows you to create interesting effects through multiple overlaying (write and offset a word several times).



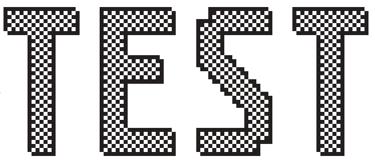
*Orginal font 8x16 with ZOOM 3x at **position 0,0** with black pattern.*



*"Outline font" resulting from overlaying **(EXOR)** at **pos.1,1**.*



*When "outline font" is overlayed again **(EXOR)** at **pos.2,2**. than the result is an "outline font with black filling".*



*Overlaying (OR) with 50% black pattern of the "outline font" at **pos.0,0**. results in a "font with pattern filling".*

Command table EA IC6963

Command											Remarks
Functions for outputting text											
Text mode	T	R L O U	n1	ptn							R/L/O/U: Write character string (R)ight, (L)eft, (O)ben (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..8); 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
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
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)
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
Control / definition commands											
Define bar graph	B	R L O U	nr	x1	y1	x2	y2	aw	ew	ptn	Define bar graph to L(eft), 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).
Clipboard commands *) (buffer for picture areas)	C	B									The entire contents of the display will be copied to the clipboard
		S	x1	y1	x2	y2					Picture area extending from x1, y1 to x2, y2 will be copied to the clipboard
		R									The picture area on the clipboard will be copied back to the display
		K	x1	y1							The picture area on the clipboard will be copied to x1, y1 in the display
		H									The picture area on the clipboard will be sent as hard copy via RS232
		L	data...								Load a picture area to the clipboard; see picture structure for picture data
Automatic flashing area (cursor function)	Q	D	x1	y1	x2	y2					Defines a flashing area x1,y1 to x2,y2; activate flashing function
		Z	n1								Set the flashing time n1= 1..15 in 1/10s; 0=deactivate flashing function
		M	I								Inverse mode (flashing area will be inverted); activate flashing function
		M	ptn								Clipboard mode*) ptn=pattern(0..7) of the block cursor; activate flashing
Select / deselect graphics lcd	K	S	n1								Activate display with address n1 (n1=0..3; n1=255: all)
		D	n1								Deactivate display with address n1 (n1=0..3; n1=255: all)
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)
Set display type	!	n1	n2	LO	HI						Another display can be set. n1=X resolution (64..240); n2=Y resolution (16..128); LO, HI 16-bit picture start address (normally $0000)
Send commands											
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: X resolution, Y resol., 'H' (e.g. 240, 64 (pixels), horizontal picture)

*) *All clipboard commands require a display RAM of at least 8 KB. The clipboard commands cannot be used with displays having a smaller RAM (e.g. 2 KB).*

PARAMETERS

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 characters are allowed, i.e. no space bars, no commas. End of command **does not need a closing byte** such as a Carrige Return (except character string: $00).

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. 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

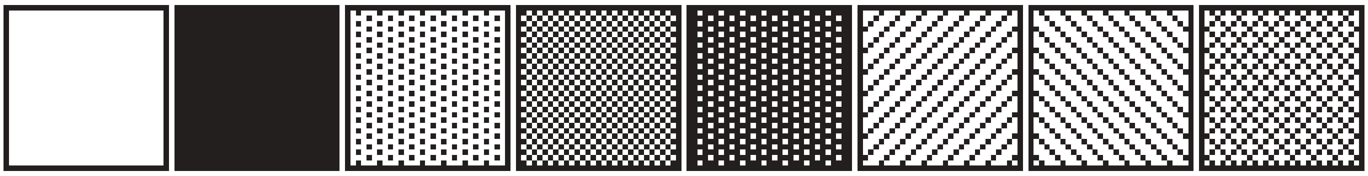
Below table shows the character string "Test" which is displayed at coordinates 7,3.

Example	**Codes**							
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 will link and display rectangular areas, bargraphs and even text lines with various pattern.

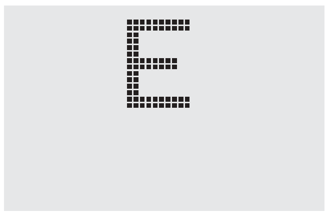
This pattern are available:



DESCRIPTION OF VARIOUS GRAPH FUNCTIONS

Coming 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 transfered Bytes as Hex codes.

A x1 y1 n1 — Set ASCII-Characters



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

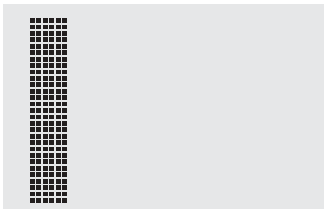
Character 'E' will be displayed at coordinates 19,2

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

Text mode: Replace and black pattern

B L/R/O/U nr x1 y1 x2 y2 aw ew ptn — Define Bargraph

Up to 8 bargraphs (**nr**=1..8) can be defined, which may oscillate to **L**=left, **R**=right, **O**=top or **U**=bottom 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).



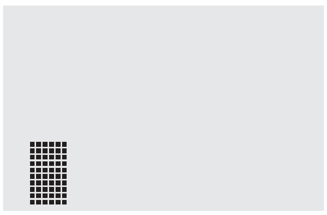
Example: $42 $4F $01 $04 $02 $09 $1E $04 $14 $01

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 a current value of 4..20 mA. (Hardcopy shows bargraph at its full level operating at $42 $01 $14)

B nr value — Draw Bargraph

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

If **value** > ew, than final value will be displayed. Shape of bargraph must be defined first, see above example.



Example: $42 $01 $0A

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

C B*) — Save content of display to clipboard

The entire contents of the display will be copied to the clipboard (buffer).

Example: $43 $42

This will save the entire contents of the display to the clipboard so that the screen can be restored later. The contents of the display will not be altered in the process.

**) All clipboard commands require a display RAM of at least 8 KB. The clipboard commands cannot be used with displays having a smaller RAM (e.g. 2 KB).*

C S x1 y1 x2 y2[*)] — Save area to clipboard

An area extending from the top left-hand corner **x1,y1** to the bottom right-hand corner **x2,y2** will be copied to the clipboard (buffer).

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

This will save the area extending from 0,0 to 23,27, so that the screen can be restored later. The contents of the display will not be altered.

C R[*)] — Restore area

The area that was saved last will be copied from the clipboard (buffer) back to the display. Target: original coordinates.

Example: $43 $52

This will restore the area last saved.

C K x1 y1[*)] — Copy area from clipboard

The area last saved to the clipboard (buffer) will be copied to a new position at **x1,y1** in the display.

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

This will take the area that was last saved and copy it to coordinate 10,32.

C L data[*)] — Load image onto the clipboard

This will take the data that now follows, and will load it onto the clipboard (buffer).

Example: $43 $4C as with the upload command ´U´.

This means that even with a low baud rate (slow), an image can be loaded into an invisible area, and can then be displayed “suddenly” at one or more places by means of the command ´C´, ´K´.

C H[*)] — Send image from the clipboard as hard copy

This requests the data from the clipboard (buffer). The function is similar to the ´H´, hard copy, command.

Example: $43 $48

and the image on the clipboard will be sent immediately via RS-232.

D L/I/S — Display command

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

Example: $44 $49

This will invert the entire contents of the display.

[*)] All clipboard commands require a display RAM of at least 8 KB. The clipboard commands cannot be used with displays having a smaller RAM (e.g. 2 KB).

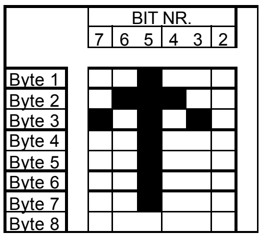
E n1 data — Define character

You can define up to twenty-one characters yourself (depending on the size of the font). These characters will then have the ASCII codes 1 to 21, and will remain in an invisible screen RAM of 128 bytes until the supply voltage is switched off. In the case of font 1, up to twenty-one characters can be defined; with font 2, the figure is sixteen; and with font 3, the largest, it is eight characters. Please note that if you specify several characters from different fonts, then you must bear in mind that a character with code 1 of the 8x16 font, for example, will need the same amount of RAM as characters with the codes 1 to 3 of the 4x6 font (see the table alongside).

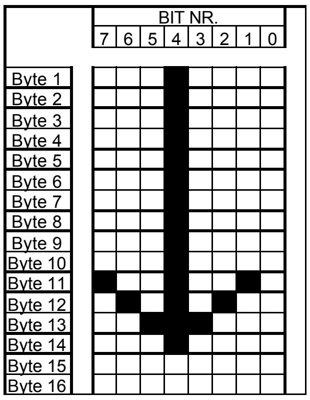
Define characters (ASCII)

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

Example 1:
$45 $01
$20 $70 $A8 $20 $20 $20 $20 $00
This defines an up arrow for ASCII no. 1, using the character set 6x8.



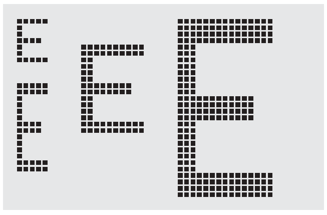
Example 2:
$45 $02
$10 $10 $10 $10 $10 $10 $10 $10 $10 $10 $92 $54 $38 $10 $00 $00
This defines a down arrow for ASCII no. 2, using the character set 8x16.



F n1 n2 n3 — Set font

The font with the no. **n1** (1=4x6 capital letters only; 2=6x8; 3=8x16) will be set. In addition, a zoom factor (1..8 times) for the width **n2** and the height **n3** will be set separately.

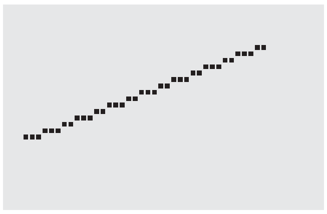
Example: $46 $02 $03 $04
The 6x8 font with the width enlarged three time and the height enlarged four times will be set with immediate effect. In the diagram alongside, the character ‘E’ from the 6x8 font is shown with different zoom factors.



G x1 y1 x2 y2 — Draw straight line

A straight line will be drawn from coordinate **x1,y1** to coordinate **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.



H x1 y1 x2 y2 — Get a hard copy of the display

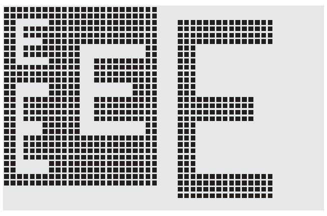
This requests the area extending from the top left-hand corner **x1,y1** to the bottom right-hand corner **x2,y2**. The graphics chip will immediately send the width and height of the section of the image, followed by the image data. See the upload image command, ‘U’, for the structure of the image data.

Example: $48 $00 $00 $1F $0F
The top left-hand part of the screen, measuring 32 x 16 pixels, will be sent immediately via RS-232.

I x1 y1 x2 y2 — Invert area

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 when the contents of the display are as shown in the example under “Set font”.

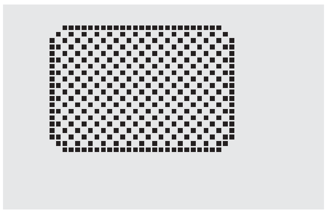


J x1 y1 x2 y2 ptn — Draw round box

A rectangle with rounded corners 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 will be deleted. Compare 'N, Draw round corner'.

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

This will draw a round box extending from 7,3 to 35,22, with the pattern 3=50% black.



K S/D n1 — Select / Deselect graphics controller

The graphics controller with the hardware address **n1** (0..3) will be **S**=selected or **D**=deselected; The address 255=$FF is a master address that is used to access all graphics controllers. The address is set by hardware (pins ADR0/1, see page 16).

Example: $4B $44 $00

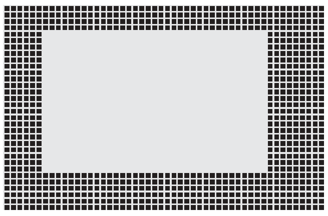
All commands for the graphics controller with the address $00 will be ignored with immediate effect.

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.

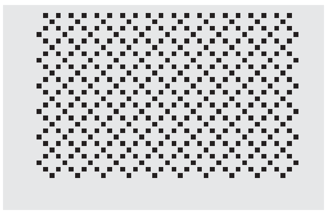


M x1 y1 x2 y2 ptn — Area with fill pattern

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.

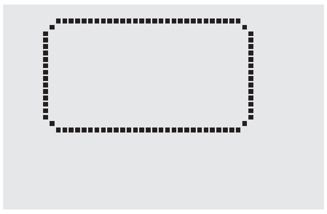


N x1 y1 x2 y2 — Draw a box with rounded corners

A rectangle with rounded corners will be drawn 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 round corner box will not be altered. Compare 'J, Draw round box'.

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

This will draw a round corner from 6,2 to 38,19.

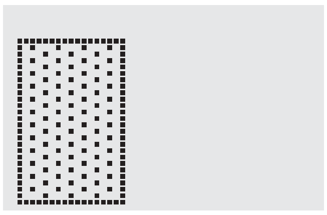


O x1 y1 x2 y2 ptn — Draw box

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 rectangle'.

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.



P x1 y1 — Place a dot

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

Example: $50 $11 $0D

This will set the pixel at coordinate 17,13.

Q D x1 y1 x2 y2 — Define flashing area

This specifies the area extending from the top left-hand corner **x1,y1** to the bottom right-hand corner **x2,y2** as an automatic flashing area. At the same time, it starts the flashing function. This makes it possible to present a "cursor" when entries are being made.

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

This defines the flashing area from 0,15 to 7,16. (Simulation of an underscore cursor for the 8x16 font, with a character at position 0,0.)

Q Z n1 — Set flashing time

This sets the flashing time to **n1** (=1..15) tenth seconds. At **n1**= 0, the flashing function will be deactivated, and the original screen will be restored.

Example: $51 $5A $05
This will set the flashing time to ½ second.

Q M I — Inverse flashing mode

This automatically inverts the specified flashing area, using the flashing time that has been set. At the same time, it starts the flashing function.

Example: $51 $49
This will set the inverse flashing mode.

Q M ptn[*)] — Block cursor flashing mode

This saves the defined flashing area to the clipboard. A cyclical changeover will be carried out between the original area and the pattern ptn (=0..7), using the flashing time that has been set. This means, for example, that a flashing cursor can be simulated (ptn=1 black), or a flashing word can be displayed (ptn=0 white). At the same time, the flashing function will be started. It will then no longer be possible to use the clipboard commands!

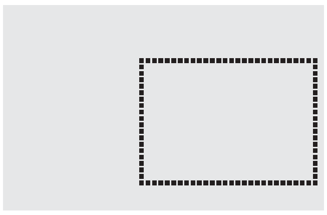
Example: $51 $43 $00

This will set the flashing mode block cursor with the white pattern. The area that has been set will therefore flash on a white background.

R x1 y1 x2 y2 — Draw rectangular box

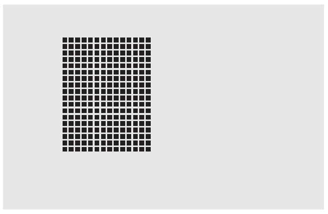
This draws a rectangular box from the top left-hand corner **x1,y1** to the bottom right-hand corner **x2,y2,** taking into account the graphics mode 'V' that has been set (set / delete / inverse). The contents of the rectangle will not be altered in the process. Compare 'N, Draw box with rounded corners' on page 12.

Example: $52 $15 $08 $30 $25
This will draw a rectangle from position 21,8 to 48,37.



[*)] The command Q M mst requires a display RAM of at least 8 KB. This command cannot be used with displays having a smaller RAM (e.g. 2 KB).

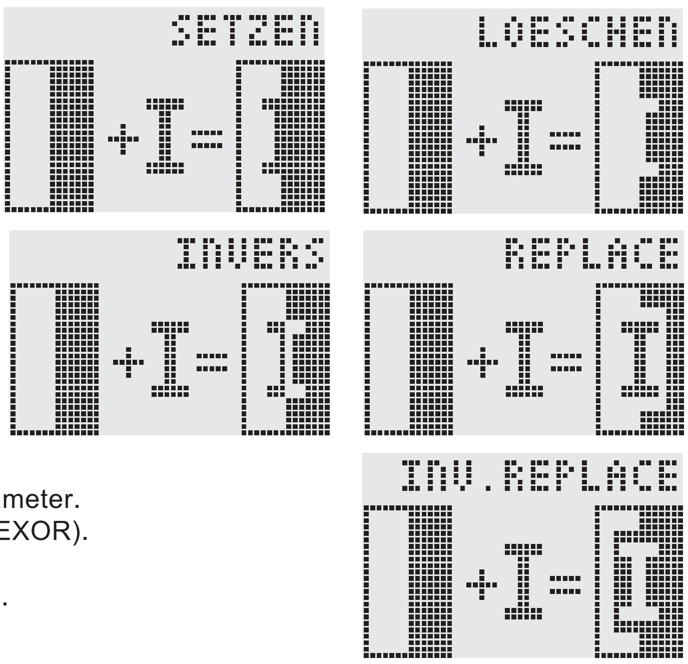
S x1 y1 x2 y2 — Fill area



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.

T L/R/O/U n1 ptn — Set text mode



The link mode **n1**, and pattern mode **ptn** sets ASCII-characters 'A' in text mode and displays a character string (command 'Z'). In addition, the write direction is stipulated for the command 'Z': **L**=left, **R**=right, **O**=oben (up), and **U**=unten (down).

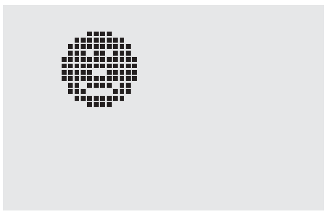
Example: $54 $52 $03 $03

This will set the link mode for all of the following text functions to gray characters (pattern 3 = 50% black), inverts the background and writes the character string from left to right.

Link mode n1:

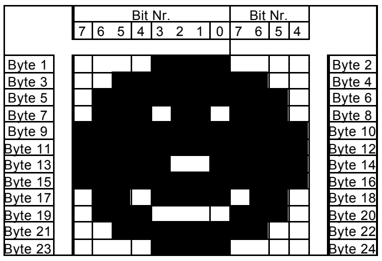
1 = set: Black pixels, irrespective of the previous value (OR).
2 = delete: White pixels, irrespective of the previous parameter.
3 = inverse: Black pixels become white, and vice versa (EXOR).
4 = replace: Delete background, and set black pixels.
5 = inverse replace: Fill background, and set white pixels.

U x1 y1 data — Upload image



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

data: - 1 byte for width of image in pixels.
- 1 byte for height of image in pixels.
- Image data: pixel qty = ((width+7) / 8) * height bytes.
1 byte stands for 8 horizontal pixels on the screen; 0=white, 1=black; MSB: left, LSB: right; the image builts up from top to bottom.
Programme BMP2BLH.EXE on our disk EA DISKIC1, which is available as an accessory, generates the image data - including details of width and height - from monochrome Windows bitmap graphics (*.BMP).

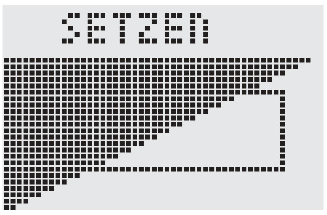


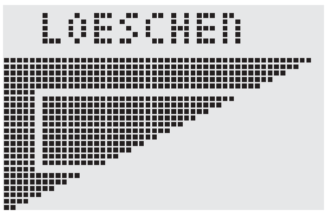
Example: $55 $09 $04 $0C $0C
$0F $00 $3F $C0 $7F $E0 $76 $E0 $FF $F0 $FF $F0
$F1 $F0 $FF $F0 $6F $60 $70 $E0 $3F $C0 $0F $00

Loads beside shown image to coordinates 9.4.

V n1 Set graphics mode

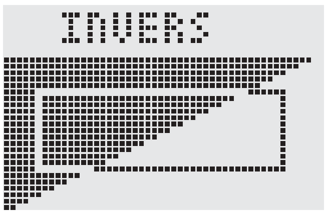
This sets the link mode **n1** for the following graphics functions: set point ('P'), draw straight line ('G'), continue drawing straight line ('W'), draw rectangle ('R'), draw box with rounded corners ('N'), fill area with pattern ('M').





Example: $56 $03
This will set the link mode to inverse.

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

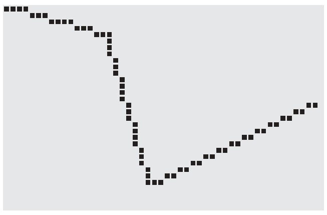


Link mode n1:

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.

W x1 y1 Continue straight line

This continues a straight line, from the point or the end of the line last drawn, 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.

X n1 Read I/O port

This reads in port (**n1**: 0..7 = I/O: 0..7). If **n1** = 8, all I/O 0..7 will be read in as a binary value; I/O 0: LSB, I/O 7: MSB. See application on page 13.

Example: $58 $02
This will read in signals at I/O 2, and will send $00 in the case of level L and $01 in the case of level H via RS-232.

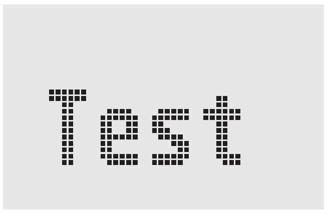
Y n1 n2 Set I/O port

This changes the port (**n1**: 0..7 = I/O: 0..7) to parameters **n2** (0=L level; 1=H level; 2=invert port). If **n1**= 8, all I/O 0..7 will be output as a binary value **n2**; I/O 0: LSB, I/O 7: MSB. See application on page 13.

Example: $59 $02 $01
This will switch the port I/O 2 to H-level.

Z x1 y1 ASCII... NUL Write character string

This writes the character string **ASCII...** to coordinate **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** (zero) ($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: 8x16 with normal width and height.
Text mode: Written from left to right in link mode replace and with black pattern.

EA IC6963

! n1 n1 adrLO adrHI — Set display

This command allows setting of displays which are not programmable by DTYPO..2 (see page 4). Width of display is set with **n1** and height of display is set with **n2**. Parameter range for **n1** is 64..256 (**n1**=0 for a width of 256 pixels). Parameter range for **n2** is 16..128. Also starting adress of an image **adrLO** and **adrHI** (16-bit) can be set (usually $0000).

Example: $21 $64 $32 $00 $00
sets a display size to 100 pixel width and to 50 pixel height.

? — Query display type

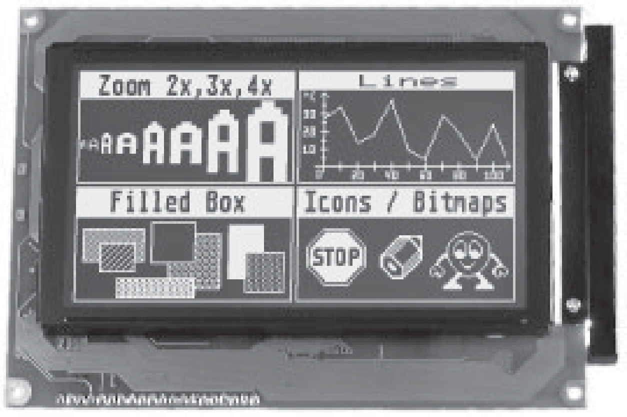
This queries the resolution of the display and the type of image structure.

Example: $3F

After this command, the X and Y resolution will be sent over the RS-232 interface first, followed by the type of image structure ('H') for horizontal build up.

GRAPHICS UNIT WITH IC6963
240x128 - 128x128 - 240x64

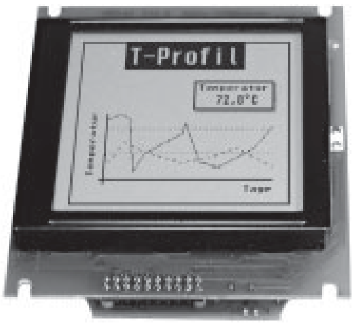
EA GE240-6KV24



EA GE240-7KCV24

FEATURES

* 240x128 PIXEL WITH CFL-ILLUM. BLUE NEGATIVE
* 240x128 PIXEL WITH LED-ILLUMINATION GN/YL
* 128x128 PIXEL WITH LED-ILLUMINATION GN/YL
* 240 x 64 PIXEL WITH LED-ILLUMINATION GN/YL
* 3 FONTS (ZOOM) FROM 2mm TO 5mm UP TO APPROX. 50mm
* SUPPLY VOLTAGE: +5V / 500..1000mA
* NEGATIVE VOLTAGE FOR CONTRAST IS GENERATED ON-BOARD
* CFL-CONVERTER ON-BOARD (EA GE240-7KCV24)
* "TRUE" RS-232 LEVEL ±10V
* RS-232 BAUDRATES 1200..115,200 BD



EA GE128-7KV24

ACCESSORIES

* CABLE FOR CONNECTING TO 9-POL. SUB-D (FEMALE): **EA KV24-9B**
* DISK WITH PROGRAMME EXAMPLES FOR PC-DOS (IN GERMAN): **EA DISKIC-1**

ORDERING INFORMATION

GRAPHICS UNIT 240x128 WITH CFL-ILLUM. BLUE NEGATIVE	**EA GE240-7KCV24**
GRAPHICS UNIT 240x128 WITH LED-ILLUMINATION GN/YL	**EA GE240-7KV24**
GRAPHICS UNIT 128x128 WITH LED-ILLUMINATION GN/YL	**EA GE128-7KV24**
GRAPHICS UNIT 240 x 64 WITH LED-ILLUMINATION GN/YL	**EA GE240-6KV24**