5,7" LCD CONTROL PANEL WITH TOUCH PANEL

EA KIT320-8CTP
Dim. 153x120mm

TECHNICAL DATA

* 5.7" LCD GRAPHICS DISPLAY WITH DIVERSE GRAPHICS FUNCTIONS AND FONTS
* 320x240 PIXELS WITH CFL ILLUMINATION, BLUE NEGATIVE (RECOMMENDED)
* 320x240 PIXELS WITH LED ILLUMINATION, WHITE, BLACK CHAR. FSTN
* FONT ZOOM FROM approx. 2mm TO approx. 80mm, ROTATABLE in 90° STEPS
* SUPPLY +5V±2% @ 500mA (CFL)/400mA (LED) OR OPTIONALLY +9..35V
* RS-232 OR OPTIONALLY RS-422 WITH BAUD RATES OF 2,400 TO 115,200
* POSITIONING **ACCURATE TO THE PIXEL** WITH ALL FUNCTIONS
* STRAIGHT LINE, POINT, AREA, AND/OR/EXOR, BAR GRAPH...
* CLIPBOARD FUNCTIONS, PULL-DOWN MENUS
* UP TO 256 IMAGES STORABLE INTERNALLY
* UP TO 1024 MACROS PROGRAMMABLE (FLASH WITH 480 KB)
* ILLUMINATION SWITCHABLE BY MEANS OF SOFTWARE
* COMBINATIONS OF TEXT AND GRAPHICS, FLASHING ATTRIBUTES, INVERTED
* ANALOG TOUCH PANEL: VARIABLE GRID WITH 10x8 FIELDS, FOR EXAMPLE
* FREELY DEFINABLE KEYS AND SWITCHES
* MENUS AND BAR GRAPH CAN BE SET BY TOUCH
* DOT-MATRIX DISPLAY CAN BE CONNECTED DIRECTLY AS SECONDARY DISPLAY

ORDER DESIGNATION

320x240 DOTS 5.7" WITH CFL ILLUMINATION, BLUE NEGATIVE	**EA KIT320-8CTP**
AS ABOVE, BUT WITHOUT TOUCH PANEL	**EA KIT320-8C**
320x240 DOTS, WHITE LED ILLUMINATION, POSITIVE MODE, FSTN	**EA KIT320-8LWTP**
AS ABOVE FSTN, BUT WITHOUT TOUCH PANEL	**EA KIT320-8LW**

OPTIONS / ACCESSORIES

SUPPLY +9..35V= INSTEAD OF +5V=	**EA OPT-9/35V**
RS-422 INTERFACE INSTEAD OF RS-232	**EA OPT-RS4224**
OPTOCOUPLER ONBOARD FOR 8 INPUTS AND OUTPUTS	**EA OPT-OPTO16**
ALUMINUM MOUNTING BEZEL, ANODIZED MATT BLACK	**EA 0FP320-8SW**
ALUMINUM MOUNTING BEZEL, ANODIZED BLUE	**EA 0FP320-8BL**
CABLE (1.5m) FOR CONNECTING TO 9-PIN SUB-D (RS-232 FEMALE)	**EA KV24-9B**
FLOPPY DISK FOR MACRO PROGRAMMING (PC WIN95/98/2K)	**EA DISK320**

ELECTRONIC ASSEMBLY GMBH

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

GENERAL

The EA KIT320 is a fully assembled control and operating unit with a variety of integrated functions. The display has very compact dimensions and offers excellent super-twist contrast, which means the unit can be put into operation immediately. It is controlled via the standard RS-232 or RS-422 interface. In addition to complete graphics routines for display output, the operating unit also contains a wide variety of fonts.
Graphics commands similar to high-level language are used for programming. There is no longer any need for the time-consuming programming of character sets and graphics routines. The ease of use offered by macros and input via touch panel make it a real power display.

HARDWARE

The operating unit is designed to work with an operating voltage of +5V. A supply voltage of 9..35V is also possible. Serial asynchronous data transfer is carried out in RS-232 or RS-422 format. The transmission format is set permanently to 8 data bits, 1 stop bit, and no parity. Rates between 2,400 baud and 115,200 baud can be selected by means of DIP switches. RTS and CTS handshake lines are available.

Data format:

TOUCH PANEL

The EA KIT320-8CTP and -8LWTP versions are equipped with an integrated touch panel. You can make entries and menu or bar graph settings by touching the display. The labeling of the „keys“ is flexible and can also be changed during runtime (different languages, icons). The drawing of the individual „keys“ and the labeling is handled by the integrated software.

SOFTWARE

The operating unit is programmed by means of commands, such as *Draw a rectangle from (0,0) to (64,15)*. No additional software or drivers are required. Strings can be placed with pixel accuracy. Flashing attributes can be assigned as often as you like – for graphics as well. Text and graphics can be combined at any time. Up to 16 different character sets can be used. Each one can be zoomed from 2 to 8 times. With the largest character set, the words and numbers displayed will fill the screen.

ACCESSORIES

Floppy disk for macro creation

A floppy disk (EA DISK320) is required for macro programming*). This converts the commands entered in a text file into a code that can be read by the operating unit, and programs them into the FLASH PROM.

Cable for PC

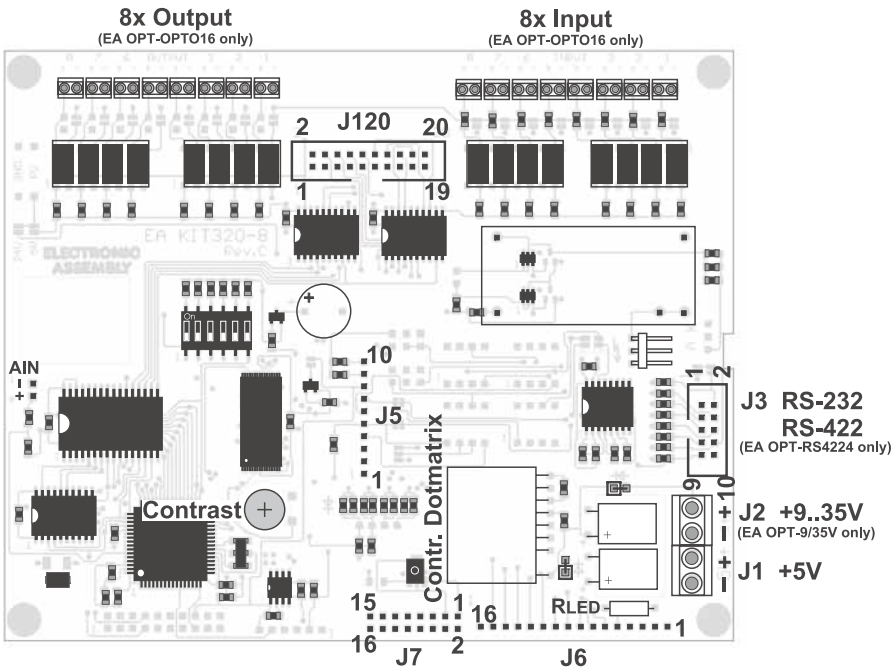
To enable simple connection to PCs (specifically for macro programming), we provide an optional 1.5m cable and a 9-pin SUB-D female connector (EA KV24-9B). Simply insert it into COM 1 or COM 2 and get started. Note: The cable is not suitable for the RS-422 version EA OPT-RS4224.

*) *also available at http://www.lcd-module.de/deu/disk/disk320.zip*

SUPPLY VOLTAGE

In the standard model, the supply voltage of +5V is fed in via the screw-type terminal J1. In the case of the version for 9V to 35V (EA OPT-9/35V), the power is supplied via J2.

Important: It is imperative that the polarity is correct. Even very brief polarity reversal can damage the entire operating unit immediately and irreparably.



View from rear side

BAUD RATES

The baud rate can be set by means of the three DIP switches on the left. When the equipment is delivered, the setting is 9,600 baud (DIP 3 ON). Please note that the internal data buffer is only 128 bytes. The RTS handshake line must therefore be queried (+10V level: data can be accepted; -10V level: display is busy). The data format is set permanently to 8 data bits, 1 stop bit, no parity.

Baudrates

DIP Switch 1	DIP Switch 2	DIP Switch 3	data format 8,N,1
OFF	ON	ON	2,400
ON	OFF	ON	4,800
OFF	OFF	ON	9,600
ON	ON	OFF	19,200
OFF	ON	OFF	38,400
ON	OFF	OFF	57,600
OFF	OFF	OFF	115,200

RS-232 Connector J3

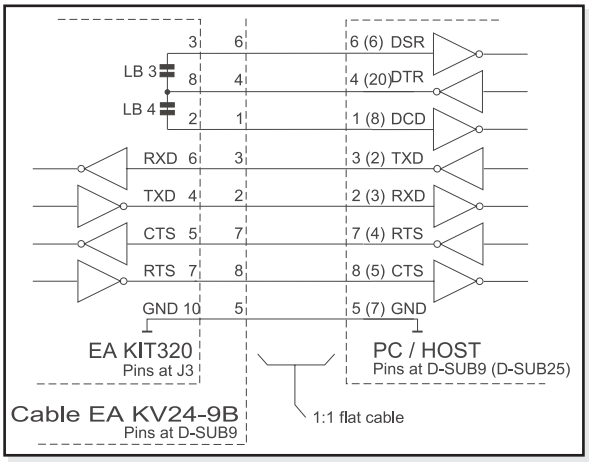
Pin	Symbo	In/Out	Function
1	VDD	-	+ 5V Supply
2	DCD	-	Connected to DTR
3	DSR	-	Connected to DTR
4	TxD	Out	Transmit Data
5	CTS	In	Clear To Send
6	RxD	In	Receive Data
7	RTS	Out	Request To Send
8	DTR	-	see Pin 2, Pin 3
9	-	-	NC
10	GND	-	0V Ground

RS-232/RS-422

The operating unit is shipped with an RS-232 interface as standard. The pin assignment of connector J3 is then as shown in the table on the left. J3 has a grid of 2.54mm. If the operating unit is ordered together with the EA OPT-RS4224 option, special RS-422 drivers are fitted. The pin assignment in the table on the right then applies.

RS-422 Connector J3

Pin	Symbol	Function
1	VDD	+ 5V Supply
2	Data In-	Receive Data
3	Data In+	Receive Data
4	Data Out-	Transmit Data
5	Data Out+	Transmit Data
6	HS In-	Handshake
7	HS In+	Handshake
8	HS Out-	Handshake
9	HS Out+	Handshake
10	GND	0V Ground



Incidentally, the same serial data with 5V levels and TTL logic is available at the J5 eyelet strip. These levels are suitable for direct connection to a µC. If these signals are used, 4 solder straps LB1, LB2, LB 5 und LB 6 has to be opened !

Extension J5

Pin	Symbol	In/Out	Function
1	VU	-	9..35V Supply
2	VDD	-	+ 5V Supply
3	GND	-	0V, Ground
4	TxD5	Out	Transmit Data
5	RxD5	In	Receive Data
6	RTS5	Out	Request To Send
7	CTS5	In	Clear To Send
8	RESET	In	L: Reset
9	SCL	Out	I2C Bus, Clock
10	SDA	In/Out	I2C Bus, Data

INPUTS AND OUTPUTS

All EA KIT320 operating units are supplied with 8 digital inputs and 8 outputs (5V CMOS level, non-isolated).

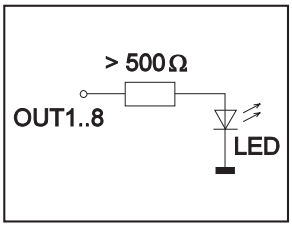
In- and Output J120						
Pin	Symbol	Function		Pin	Symbol	Function
1	VDD	+5V Supply		2	GND	0V, Ground
3	OUT1 / MO8	Port Output 1 Matrix Output 8		4	IN1 / MI8	Port Input 1 Matrix Input 8
5	OUT2 / MO7	Port Output 2 Matrix Output 7		6	IN2 / MI7	Port Input 2 Matrix Input 7
7	OUT3 / MO6	Port Output 3 Matrix Output 6		8	IN3 / MI6	Port Input 3 Matrix Input 6
9	OUT4 / MO5	Port Output 4 Matrix Output 5		10	IN4 / MI5	Port Input 4 Matrix Input 5
11	OUT5 / MO4	Port Output 5 Matrix Output 4		12	IN5 / MI4	Port Input 5 Matrix Input 4
13	OUT6 / MO3	Port Output 6 Matrix Output 3		14	IN6 / MI3	Port Input 6 Matrix Input 3
15	OUT7 / MO2	Port Output 7 Matrix Output 2		16	IN7 / MI2	Port Input 7 Matrix Input 2
17	OUT8 / MO1	Port Output 8 Matrix Output 1		18	IN8 / MI1	Port Input 8 Matrix Input 1
19	GND	0V, Ground		20	VDD	+5V Supply

8 outputs

Each line can be controlled by means of the „ESC Y W“ command. The maximum current per line is 6mA. It is therefore possible to connect an LED (low current) directly to an output. Higher currents can be amplified by means of external transistors.

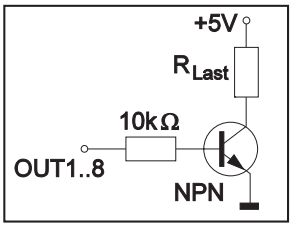
8 inputs

The inputs can be queried and evaluated („ESC Y R“) directly via the serial interface. It is also possible to call a bit/port macro automatically in the event of changes at the inputs. Automatic port querying can be deactivated by means of the „ESC Y A 0“ command.



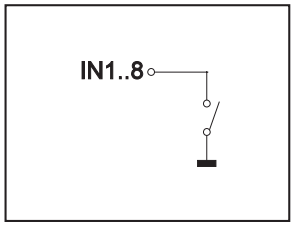
Port macros: Up to 256 port macros can be addressed by means of the binary combination of 8 inputs.

Bit macros each only affect one input. Bit macro 1..8 is called at one of inputs 1...8 in the event of a change to HIGH level. Bit macros 9..16 are called in the event of a change to LOW level.



At each change of the input port, the bit macros are executed first, followed by the port macro. If there is no macro defined, the new port status is sent via RS232/RS422. Each of the macros can change the contents of the screen or switch outputs. This allows a wide range of control tasks to be carried out. To create the macros, you need a PC and the floppy disk EA DISK320.

Note: The logic circuitry is designed for slow operations; in other words, more than 3 changes per second cannot be easily executed. If an input is open, this is evaluated as high (internal 100 kOhm pullup).

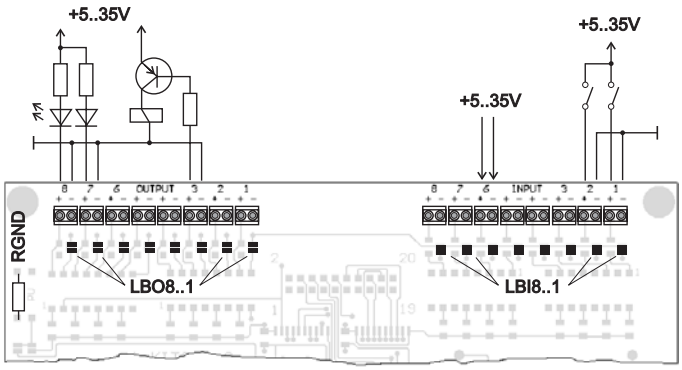


INPUTS AND OUTPUTS VIA OPTOCOUPLERS (EA OPT-OPTO16)

The inputs and outputs can be equipped optionally with optocouplers (EA OPT-OPTO16). The inputs and outputs are then isolated from the rest of the electronic components as well as each other. The connection is made via 16 different screw-type terminals.

Voltages of 5..35V can be applied directly at all 8 inputs. Voltages of over 4V are identified as high (H) level; voltages of under 2V are identified as low (L) level. Voltages between 2 and 4V are undefined.

The collector and emitter of a transistor are each implemented as outputs on the screw-on terminals. Each output can switch a maximum of 10mA.



Note: The negative pole of each screw-on terminal can be interconnected by closing solder straps LBI1..8 or LBO1..8. In addition, these solder straps can be connected to system GND (solder 0Ω bridge RGND).

Note: The optocouplers invert the input logic (all inputs open: port macro n°255). It is advisable here (in the power-on macro, for example) to use the „ESC Y I 1“ command to evaluate the inputs inversely (i.e. all inputs open: port macro n°0).

EXTERNAL MATRIX KEYBOARD

A matrix keyboard (anything from individual keys to an 8x8 matrix keyboard) can be connected at the plug-in connection J120. The number of inputs and outputs of the ports (n1,n2=1..8) used is defined and the key debouncing is specified (n3=0..7 in 50ms increments) by means of the 'ESC Y M n1 n2 n3' command. Please note when an external keyboard is connected that the digital inputs are reduced by the number n1 and the outputs are reduced by the number n2.
Each key is generally switched between an output and an input. Each input has a 100kΩ (approx.) pullup. In order to identify double keystrokes, the outputs must be decoupled from each other. This is best done with Schottky diodes (e.g. BAT 43).

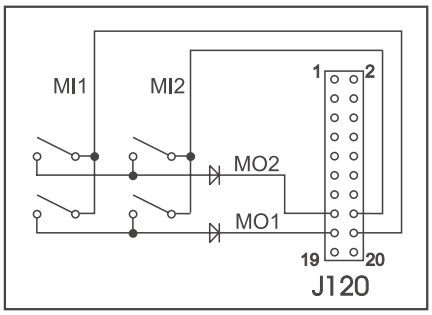
Transmitting the keystrokes
At each keystroke (key number 1..64), the associated matrix macro is executed or, if no macro is defined, the key number is transmitted with code letters. The release of the key is not transmitted. If the release of the key is to be transmitted as well, this can be done by defining matrix macro no. 0.

Note If the CTS handshake line does not permit transmission, up to 8 keystrokes are stored in the key buffer. When the buffer is full, older keystrokes may be lost.

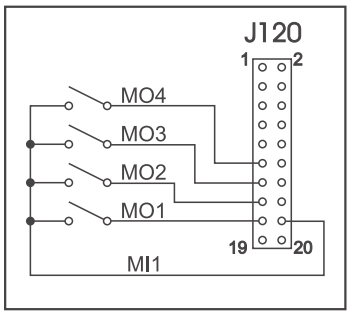
Determining the key number:
Key no. = (output no. - 1) * no. of inputs + no. of outputs
(output = MOx, input = MIx).

Example: Connection of 4 keys in 3 variations

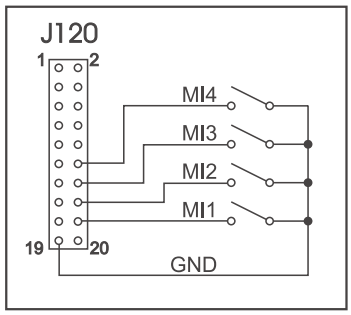
- Variant 1: The 4 keys are defined as a 2x2 matrix by means of the 'ESC Y M 2 2 ..' command. The keys are connected to 2 inputs (MI1, MI2) and 2 outputs (MO1, MO2). The outputs are decoupled from each other by means of diodes so that double keystrokes can be identified. There are 6 inputs and 6 outputs available as port connections.
- Variant 2: The 4 keys are defined as a 1x4 matrix by means of the 'ESC Y M 1 4 ..' command. The keys are connected to 4 outputs (MO1, MO2) and read in via input MI1. There are 7 inputs and 4 outputs available as port connections.
- Variant 3: If only one output is used (4x1 matrix), the keys can also be connected to ground and read in directly at the inputs (= 4x0 matrix). The 4 keys are defined at the 4 inputs (MI1..MI4) by means of the 'ESC Y M 4 0 ..' command. There are 4 inputs and 8 outputs available as port connections.



Type 1: 2x2 Matrix



Type 2: 1x4 Matrix



Type 3: 4x0 Matrix

CONNECTION FOR DOT-MATRIX DISPLAY

You can connect a single external dot-matrix module (with HD44780 or compatible) with 1x8 up to 4x20 or 2x40 characters at eyelets J6 and J7. This dot-matrix display can be addressed very conveniently by means of 'ESC T xx' terminal commands. Alternatively, 'ESC L xx' commands are available for controlling the HD44780 directly. A potentiometer for contrast adjustment is already fitted. A suitable series resistor R_{LED} can be fitted for LED backlighting.

Dotmatrix Connector J6 + J7

Pin	Symbol	Pegel	Description
1	VSS	L	0V, Ground
2	VDD	H	+5V Supply
3	VEE	-	Display voltage 0V-5V
4	RS	H / L	Register Select
5	R/W	H / L	H: Read / L: Write
6	E	H	Enable
7	D0	H / L	Data 0 (LSB)
8	D1	H / L	Data 1
9	D2	H / L	Data 2
10	D3	H / L	Data 3
11	D4	H / L	Data 4
12	D5	H / L	Data 5
13	D6	H / L	Data 6
14	D7	H / L	Data 7 (MSB)
15	A	-	Anode (RLED)
16	K	L	Kathode (=VSS)

MACRO PROGRAMMING

Single or multiple command sequences can be grouped together in macros and stored in the data flash. You can then start them by using the *Run macro* commands. There are different types of macro:

Normal macros (0..255)
These are started by means of an 'ESC MN xx' command via the serial interface or from another macro. A series of macros occurring one after the other can be called cyclically (movie, hourglass, multi-page help text). These automatic macros continue to be processed until a command is received via RS-232 or another macro is activated (e.g. touch, port or matrix macro).

Touch macro (1..255)
Started when you touch/release a touch field (only in versions with a touch panel - TP) or issue an 'ESC MT xx' command.

Menu macro (1..255)
Started when you choose a menu item or issue an 'ESC MM xx' command.

Bit macro (1..8) or (9..16)
Started when a voltage is applied/changed at individual inputs IN 1..8 (bitwise) or by means of an 'ESC MB xx' command. Bit macros 1..8 respond to a rising edge, whereas bit macros 9..16 respond to a falling edge of inputs 1..8.

Port macro (0..255)
Started when a voltage is applied/changed at the 8 inputs IN 1..8 (binary combined) or by means of an 'ESC MP xx' command.

Matrix macro (0..64)
Matrix macro 1..64: Started when you press a key or issue an 'ESC MX xx' command. Matrix macro 0: Started on release when a key is no longer depressed or by means of a command. The matrix keyboard is connected at the inputs and outputs; a single 8x8 matrix keyboard can be connected at most.

Power-on macro
Started after power-on. You can switch off the cursor and define an opening screen, for example.

Reset macro
Started after an external reset or after a voltage drop under 4.7V (VDD-VSS).

Watchdog macro
Started after a fault/error (e.g. failure).

Important: If a continuous loop is programmed in the power-on, reset or watchdog macro, the display can no longer be addressed. In this case, the only thing you can do is: DIP switch 5 to ON, power-off, power-on and then DIP 5 to OFF. The macros then have to be read in again.

256 IMAGES STORED IN THE INTERNAL DATA FLASH

To reduce the transmission times of the serial interface or to save storage space in the processor system, up to 256 images can be stored in the internal data flash. They can be called using the „ESC U I“ command or from within a macro. All images in the Windows BMP format (monochrome images only) can be used. They can be created and edited using widely available software such as Windows Paint or Photoshop (only black and white = 1 bit).

CREATING INDIVIDUAL MACROS

To create your own macros, you need the following:
- The EA DISK320*) floppy disk, which contains a compiler, examples and fonts
- A PC with a COM1 or COM2 serial interface and approximately 500KB of hard disk space
- A text editor such as WordPad or Textpad

To define a sequence of commands as a macro, all the commands are written to a file on the PC (e.g. DEMO.KMC). You specify which character sets are to be integrated and which command sequences are to be in which macros.
Once the macros are defined, you start the program C:>KITCOMP DEMO.KMC. This creates a data flash file called DEMO.DF, which is then automatically stored in the data flash with the baud rate entered. This only takes a few seconds, and you can then use your user-defined macros immediately.
You will find a detailed description of how to program macros, together with examples, in the files DOKU.DOC (for WORD) and DOKU.TXT (DOS) on the EA DISK320*) floppy disk.

WRITE PROTECTION FOR MACRO PROGRAMMING

PCB Rev.C and newer: by setting DIP switch 6 (write enable) to OFF, you can prevent the programmed macros, images and fonts from being overwritten inadvertently. After successful programming, this DIP switch 6 should be set to OFF.
(Caution: On older PCB boards Rev.A and Rev.B (before 2002) DIP switch 6 had reverse meaning: ON=write protect; OFF=programming possible)

*) *also available at http://www.lcd-module.de/deu/disk/disk320.zip*

internal Font 1: 4x6 monospaced

+ 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)		ü			ä										Ä	
$90 (dez: 144)					ä					Ö	ü				ß	

internal Font 2: 5x6 monospaced

+ 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	ƒ
$A0 (dez: 160)	á	í	ó	ú	ñ	Ñ	ª	º	¿	⌐	¬	½	¼	¡	«	»
$B0 (dez: 176)	░	▒	▓	│	┤	╡	╢	╖	╕	╣	║	╗	╝	╜	╛	┐
$C0 (dez: 192)	└	┴	┬	├	─	┼	╞	╟	╚	╔	╩	╦	╠	═	╬	╧
$D0 (dez: 208)	╨	╤	╥	╙	╘	╒	╓	╫	╪	┘	┌	█	▄	▌	▐	▀
$E0 (dez: 224)	α	ß	Γ	π	Σ	σ	µ	τ	Φ	Θ	Ω	δ	∞	φ	ε	∩
$F0 (dez: 240)	≡	±	≥	≤	⌠	⌡	÷	≈	°	∙	·	√	ⁿ	²	■	

internal Font 3: 6x8 monospaced

+ 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	ƒ
$A0 (dez: 160)	á	í	ó	ú	ñ	Ñ	ª	º	¿	⌐	¬	½	¼	¡	«	»
$B0 (dez: 176)	░	▒	▓	│	┤	╡	╢	╖	╕	╣	║	╗	╝	╜	╛	┐
$C0 (dez: 192)	└	┴	┬	├	─	┼	╞	╟	╚	╔	╩	╦	╠	═	╬	╧
$D0 (dez: 208)	╨	╤	╥	╙	╘	╒	╓	╫	╪	┘	┌	█	▄	▌	▐	▀
$E0 (dez: 224)	α	ß	Γ	π	Σ	σ	µ	τ	Φ	Θ	Ω	δ	∞	φ	ε	∩
$F0 (dez: 240)	≡	±	≥	≤	⌠	⌡	÷	≈	°	∙	·	√	ⁿ	²	■	

internal Font 4: 8x8 monospaced

+ 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	ƒ
$A0 (dez: 160)	á	í	ó	ú	ñ	Ñ	ª	º	¿	⌐	¬	½	¼	¡	«	»
$B0 (dez: 176)	░	▒	▓	│	┤	╡	╢	╖	╕	╣	║	╗	╝	╜	╛	┐
$C0 (dez: 192)	└	┴	┬	├	─	┼	╞	╟	╚	╔	╩	╦	╠	═	╬	╧
$D0 (dez: 208)	╨	╤	╥	╙	╘	╒	╓	╫	╪	┘	┌	█	▄	▌	▐	▀
$E0 (dez: 224)	α	ß	Γ	π	Σ	σ	µ	τ	Φ	Θ	Ω	δ	∞	φ	ε	∩
$F0 (dez: 240)	≡	±	≥	≤	⌠	⌡	÷	≈	°	∙	·	√	ⁿ	²	³	¯

internal Font 5: 7x12 monospaced

+ 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	ƒ
$A0 (dez: 160)	á	í	ó	ú	ñ	Ñ	ª	º	¿	⌐	¬	½	¼	¡	«	»
$B0 (dez: 176)	░	▒	▓	│	┤	╡	╢	╖	╕	╣	║	╗	╝	╜	╛	┐
$C0 (dez: 192)	└	┴	┬	├	─	┼	╞	╟	╚	╔	╩	╦	╠	═	╬	╧
$D0 (dez: 208)	╨	╤	╥	╙	╘	╒	╓	╫	╪	┘	┌	█	▄	▌	▐	▀
$E0 (dez: 224)	α	ß	Γ	π	Σ	σ	µ	τ	Φ	Θ	Ω	δ	∞	φ	ε	∩
$F0 (dez: 240)	≡	±	≥	≤	⌠	⌡	÷	≈	°	∙	·	√	ⁿ	²	■	¯

internal Font 6: 8x16 monospaced

+ 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	ƒ
$A0 (dez: 160)	á	í	ó	ú	ñ	Ñ	ª	º	¿	⌐	¬	½	¼	¡	«	»
$B0 (dez: 176)	░	▒	▓	│	┤	╡	╢	╖	╕	╣	║	╗	╝	╜	╛	┐
$C0 (dez: 192)	└	┴	┬	├	─	┼	╞	╟	╚	╔	╩	╦	╠	═	╬	╧
$D0 (dez: 208)	╨	╤	╥	╙	╘	╒	╓	╫	╪	┘	┌	█	▄	▌	▐	▀
$E0 (dez: 224)	α	ß	Γ	π	Σ	σ	µ	τ	Φ	Θ	Ω	δ	∞	φ	ε	∩
$F0 (dez: 240)	≡	±	≥	≤	⌠	⌡	÷	≈	°	∙	·	√	ⁿ	²	³	¯

INTEGRATED AND EXTERNAL FONTS

There are 6 monospaced character sets integrated as standard that can be used in terminal and graphics mode. Each character set can be increased in height from 1 to 8 times in graphics output. Independently of this, the width can also be increased two to eight times.
Each character can be positioned with **pixel accuracy**. Text and graphics can be combined as required. Several different font sizes can also be displayed together.
Each text can be output left justified, right justified or centered. Rotation in 90° steps is also possible (for vertical installation of the display, for example).
Macro programming permits a further 10 fonts to be integrated. Proportional character sets are also possible (in graphics mode only); these look better and take up less space on the screen. All conceivable fonts up to 255x240 pixels in size can be created using a text editor and programmed in using the kit compiler.

e.g. external Font 10: GENEVA15.FXT proportional

+ 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)	Ä	Å	Ç	É	Ñ	Ö	Ü	á	à	â	ä	ã	å	ç	é	è
$90 (dez: 144)	ê	ë	í	ì	î	ï	ñ	ó	ò	ô	ö	õ	ú	ù	û	ü
$A0 (dez: 160)	†	°	¢	£	§	•	¶	ß	®	©	™	´	¨	≠	Æ	Ø
$B0 (dez: 176)	∞	±	≤	≥	¥	µ	∂	Σ	Π	π	∫	ª	º	Ω	æ	ø
$C0 (dez: 192)	¿	¡	¬	√	ƒ	≈	Δ	«	»	…		À	Ã	Õ	Œ	œ
$D0 (dez: 208)	–	—	“	”	‘	’	÷	◊	ÿ							
$E0 (dez: 224)																
$F0 (dez: 240)																

e.g. external Font 7: CHICAGO.FXT proportional

+ 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	ƒ
$A0 (dez: 160)	á	í	ó	ú	ñ	Ñ	ª	º	¿	⌐	¬	½	¼	¡	«	»
$B0 (dez: 176)	ã	õ	Ø	ø	œ	Œ	À	Ã	Õ	¨	´	†	¶	©	®	™
$C0 (dez: 192)	ij	IJ	ʞ	ɔ	ɟ	ʅ	ɲ	ı	ɪ	ᴎ	ᴙ	˄	ɔ	ʅ	ᴓ	ɹ
$D0 (dez: 208)	ᴏ	ᴚ	ɟ	ʎ	ʃ	ʅ	ʍ	ᴎ	ʃ	ʅ	ᴏ	ᴓ	ᴓ	§	^	∞
$E0 (dez: 224)	α	β	Γ	π	Σ	σ	µ	τ	Φ	θ	Ω	δ	φ	φ	ε	∩
$F0 (dez: 240)	≡	±	≥	≤	⌠	⌡	÷	≈	°	•	•	√	n	2	3	¯

TIP: FONT EFFECTS

With large fonts, you can use the command 'ESC ZM' mode (link, pattern) to produce interesting effects through overlaying (writing and offsetting a word several times).

*Original font 8x16 with ZOOM 3 at **position 0,0** without pattern*

*"Outline font" produced by overlaying **(EXOR)** at **pos. 1,1***

*Overlaying **(EXOR)** of the "outline font" at **pos. 2,2**. results in an "outline font with fill"*

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

FONT EXAMPLES

This hardcopy shows 6 internal and 8 external fonts.

ALL FUNCTIONS AT A GLANCE

EA KIT320: Command table 1										After reset
Command	**Codes**								**Remarks**	
Comands for outputting strings										
Output string L: left justified, C: centered R: right justified	ESC	Z	L C R	xx1	yy1	text ...	NUL		A string (...) is output to xx1,yy1. ´NUL´ ($00), 'LF' ($0A) or 'CR' ($0D) = end of string; several lines are separated by the character '|' ($7C); text between two '~' ($7E) characters flashes on/off; text between two '@' ($40) characters flashes inversely;	
Set font	ESC	Z	F	n1					Set font with the number n1 (1..16)	5
Font zoom factor	ESC	Z	Z	n1	n2				n1 = X zoom factor (1x..8x); n2 = Y zoom factor (1x..8x)	1,1
Add. line spacing	ESC	Z	Y	n1					Insert n1 pixels between two lines of text as additional line spacing	
Text angle	ESC	Z	W	n1					Text output angle: n1=0: 0°; n1=1: 90°; n1=2: 180°; n1=3: 270°;	0
Text mode	ESC	Z	V	n1					Mode n1: 1=set; 2=delete; 3=inverse; 4=replace; 5=inverse replace;	4
Text pattern	ESC	Z	M	n1					Link text with pattern no. n1; 0 = do not link text with pattern	0
String for terminal	ESC	Z	T	text ...					Command for outputting a string in a macro to the terminal	
Draw straight lines and points										
Draw rectangle	ESC	G	R	xx1	yy1	xx2	yy2		Draw four straight lines as a rectangle from xx1,yy1 to xx2,yy2	
Draw straight line	ESC	G	D	xx1	yy1	xx2	yy2		Draw straight line from xx1,yy1 to xx2,yy2	
Continue straight line	ESC	G	W	xx1	yy1				Draw a straight line from last end point to xx1, yy1	0
Draw point	ESC	G	P	xx1	yy1				Set one point at coordinates xx1, yy1	
Point size/line thickness	ESC	G	Z	n1	n2				n1 = X point size (1..15); n2 = Y point size (1..15);	1,1
Graphic mode	ESC	G	V	n1					Set drawing mode n1: 1=set; 2=delete; 3=inverse;	1
Pattern	ESC	G	M	n1					Set straight line/point pattern no. n1; 0 = do not use pattern	0
Change/draw rectangular areas										
Delete area	ESC	R	L	xx1	yy1	xx2	yy2		Delete an area from xx1,yy1 to xx2,yy2 (all pixels out)	
Invert area	ESC	R	I	xx1	yy1	xx2	yy2		Invert an area from xx1,yy1 to xx2,yy2 (invert all pixels)	
Fill area	ESC	R	S	xx1	yy1	xx2	yy2		Fill an area from xx1,yy1 to xx2,yy2 (all pixels on)	
Area with fill pattern	ESC	R	M	xx1	yy1	xx2	yy2	n1	Draw an area from xx1,yy1 to xx2,yy2 with pattern n1 (always set)	
Draw box	ESC	R	O	xx1	yy1	xx2	yy2	n1	Draw a rectangle xx1,yy1 to xx2,yy2 with fill pattern n1 (always replace)	
Draw frame	ESC	R	R	xx1	yy1	xx2	yy2	n1	Draw a frame of the type n1 from xx1,yy1 to xx2,yy2 (always set)	
Draw frame box	ESC	R	T	xx1	yy1	xx2	yy2	n1	Draw a frame box of the type n1 from xx1,yy1 to xx2,yy2 (always replace)	
Bitmap image commands										
Image from clipboard	ESC	U	C	xx1	yy1				The current contents of the clipboard are loaded to xx1,yy1 with all the image attributes	
Load internal image	ESC	U	I	xx1	yy1	nr			Load internal image with the no. (0..255) from EEPROM to xx1,yy1	
Load image	ESC	U	L	xx1	yy1	data ...			Load an image to xx1,yy1; see image structure for image data	
Image zoom factor	ESC	U	Z	n1	n2				n1 = X zoom factor (1x..8x); n2 = Y zoom factor (1x..8x)	1,1
Image angle	ESC	U	W	n1					Output angle: n1=0: 0°; n1=1: 90°; n1=2: 180°; n1=3: 270°;	0
Image link mode	ESC	U	V	n1					Mode n1: 1=set; 2=delete; 3=inverse; 4=replace; 5=inverse replace;	4
Image pattern	ESC	U	M	n1					Link image with pattern no. n1; 0 = do not link image with pattern	0
Image flashing attribute	ESC	U	B	n1					n1=0 image attribute flashing off; n1=1 image flashes on/off; n1=2 image flashes inversely;	0
Send hard copy	ESC	H							A full image is requested in Windows BMP format. The image header is sent first via RS232, followed by the actual image data (9662 bytes).	
Display commands (effect on the entire display)										
Delete display	ESC	D	L						Delete display contents (all pixels off)	
Invert display	ESC	D	I						Invert display contents (invert all pixels)	
Fill display	ESC	D	S						Fill display contents (all pixels on)	
Switch display off	ESC	D	A						Display contents become invisible but are retained, commands continue to be possible	
Switch display on	ESC	D	E						Display contents become visible again	On
Display update	ESC	D	U	n1					n1=0: Display outputs are no longer visible (but continue to be executed) n1=1: Display outputs are visible immediately n1=2: Refresh display contents (previous outputs become visible)	1
Macro commands										
Run macro	ESC	M	N	n1					Call the (normal) macro with the number n1 (0..255) (max. 7 levels)	
Run touch macros	ESC	M	T	n1					Call the touch macro with the number n1 (0..255) (max. 7 levels)	
Run port macro	ESC	M	P	n1					Call the port macro with the number n1 (0..255) (max. 7 levels)	
Run bit macro	ESC	M	B	n1					Call the bit macro with the number n1 (1..16) (max. 7 levels)	
Run menu macro	ESC	M	M	n1					Call the menu macro with the number n1 (0..255) (max. 7 levels)	
Run matrix macro	ESC	M	X	n1					Call the matrix macro with the number n1 (0..64) (max. 7 levels)	
Autom. macro cyclical	ESC	M	A	n1	n2	n3			Automatically process macros n1..n2 cyclically; n3=pause in 1/10s	
Autom. macro pingpong	ESC	M	J	n1	n2	n3			Automatically process macros n1..n2..n1 (pingpong); n3=pause in 1/10s	
Flashing area commands										
Define flashing area	ESC	Q	B	xx1	yy1	xx2	yy2		Define a flashing area (on/off) from xx1,yy1 to xx2,yy2	
Inverted flashing area	ESC	Q	I	xx1	yy1	xx2	yy2		Define an inverted flashing area from xx1,yy1 to xx2,yy2	
Delete flashing attribute	ESC	Q	L	xx1	yy1	xx2	yy2		Delete the flashing attribute from xx1,yy1 to xx2,yy2	
Set flashing time	ESC	Q	Z	n1					Set the flashing time n1= 1..15 in 1/10s; 0=deactivate flashing function	6

DEFAULT SETTINGS

After power on or a reset, some functions are set to a particular value (see last column entitled 'After reset' in the table). Please note that all the settings can be overwritten by creating a power-on macro.

EA KIT320: Command table 2													After reset
Command	**Codes**								**Remarks**				
Bar graph commands													
Define bar graph	ESC	B	R L O U	n1	xx1	yy1	xx2	yy2	sv	ev	Typ.	pat	Define bar graph to L(eft), R(ight), O(ben) (up), U(nten) (down) with the "nr" n1. xx1,yy1,xx2,yy2 form the rectangle enclosing the bar graph. sv, ev are the values for 0% and 100%. Type=0: bar; type=1: bar in rectangle; pat=bar pattern type=2: line; type=3: line in rectangle; pat= line width	No bar define
Update bar graph			A	n1	valu				Set and draw the bar graph with the number n1 to the new user "value."				
Draw new bar graph			Z	n1					Draw the bar graph with the number n1 completely				
Send bar graph value			S	n1					Send the current value of bar graph no. n1 on the serial interface				
Clipboard commands (buffer for image areas)													
Select clipboard no.	ESC	C	N	n1					2 clipboards are available, the current clipboard is selected with n1= (1,2).				1,blan
Save display contents			B						The entire contents of the display are copied to the clipboard as an image area				
Save area			S	xx1	yy1	xx2	yy2		The image area from xx1,yy1 to xx2,yy2 is copied to the clipboard				
Restore area			R						The image area on the clipboard is copied back its original position in the display				
Copy area			K	xx1	yy1				The image area on the clipboard is copied to xx1,yy1 in the display				
Menu/pop-up commands													
Define menu and display	ESC	N	D	xx1	yy1	nr	text ...	NUL	A menu is drawn as of the corner xx1,yy1 with the current menu font. nr:= currently inverted entry (e.g.: 1 = 1st entry) Text:= string with menu items. The different items are separated by the character '|' ($7C,dec:124) (e.g. "item1|item2|item3"). The background of the menu is saved automatically. If a menu is already defined, it is automatically canceled+deleted.				
Next item			N						The next item is inverted or remains at the end				
Previous item			P						The previous item is inverted or remains at the beginning				
End of menu/send			S						The menu is removed from the display and replaced with the original background. The current item is sent as a number (1..n) (0=no menu displayed)				
End of menu/macro			M	n1					The menu is removed from the display and replaced with the original background. Menu macro n1 is called for item 1, menu macro nr+1 for entry 2, and so on				
End of menu/cancel			A						The menu is removed from the display and replaced with the original background				
Set menu font			F	n1					Set font with the number n1 (1..16) for menu display				5
Menu font zoom factor			Z	n1	n2				n1 = X zoom factor (1x..8x); n2 = Y zoom factor (1x..8x)				1,1
Add. line spacing			Y	n1					Insert n1 pixels between two menu items as additional line spacing				
Menu angle			W	n1					Menu display angle: n1=0: 0°; n1=1: 90°; n1=2: 180°; n1=3: 270°;				0
Port commands													
Write output port	ESC	Y	W	n1	n2				n1=0: Set all 8 output ports in accordance with n2 (=8-bit binary value) n1=1..8: Reset output port n1 (n2=0); set (n2=1); invert (n2=2)				Ports 1-8=0
Read input port			R	n1					n1=0: Read all 8 input ports as 8-bit binary value n1=1..8: Read input port <n1> (1=H level=5V, 0=L level=0V)				
Port scan on/off			A	n1					The automatic scan of the input port is n1=0: deactivated; n1=1: activated				1
Invert input port			I	n1					The input port is n1=0: normal; n1=1: evaluated inverted				0
Matrix keyboard			M	n1	n2	n3			Specifies an external matrix keyboard at the inputs and outputs. n1=number of inputs (1..8); n2=number of outputs (0..8); n3= debouncing (0..7)				0
Illumination on/off/half			L	n1					CFL/LED illumination n1=0: OFF; n1=1: ON; n1=2: half brightness;				1
Other commands													
Wait (pause)	ESC	X	n1						Wait n1 tenths of a second before the next command is executed.				
Beep on/off	ESC	J	n1						n1=1..255: Tone on for n1 1/10s				OFF
Send bytes	ESC	S	B	num	data ...				num (=1..255) bytes are sent on the RS-232/RS-422 data ... = num bytes (e.g. control of an external serial printer)				
Send analog value	ESC	S	D						The current value of the analog input AIN is sent on the RS-232/RS-422				
Send version	ESC	S	V						The software version no. + date is sent as a string on the RS-232/RS-422				
Commands to HD44780 *)	ESC	L	B	num	data ...				num (=1..255) commands are sent to the ext. dot-matrix module with HD44780.				
Data to HD44780 *)			D	num	data ...				num (=1..255) data is sent to the ext. dot-matrix module with HD44780.				
Read EEPROM	ESC	E	R	addr	num				num (=1..255) bytes are requested from the internal user EEPROM as of the address addr and sent via the RS-232/RS-422.				
Write EEPROM			W	addr	num	data ...			num (=1..255) bytes are written to the internal user EEPROM as of the address addr. data ... = num bytes				
Read I2C bus	ESC	I	R	addr	num				num (=1..255) bytes are requested from the block on the I2C bus with the device address addr and sent via the RS-232/RS-422.				
Write I2C bus			W	addr	num	data ...			num (=1..255) bytes are sent on the 12C bus for the block with the device address addr. data ... = num bytes				

*) *Only for Version 1.1 and higher*

EA KIT320: Commands for the touch panel												After reset
Command	**Codes**							**Remarks**				
Touch: Define areas												
Define touch key (key remains depressed as long as there is contact)	ESC A	C	f1 f2	down code	up code	text ...	NUL				'C': The touch fields f1 to f2 are defined for a key. 'T': The area from xx1,yy1 to xx2,yy2 is defined as a key. 'U': Image no. n1 is loaded to xx1,yy2 and defined as a key. 'down code':(1-255) Return/touch macro when key pressed. 'up code': (1-255) Return/touch macro when key released. (down/up code = 0 press/release not reported). ´text´: A string that is centered with the current touch font in the touch key follows; multiline text is separated with the character '|' ($7C, dec: 124); 'NUL': ($00) = end of string	
		T	xx1 yy1	xx2	yy2	down code	up code	text ...	NUL			
		U	xx1 yy1	n1	down code	up code	text ...	NUL				
Define touch switch (status of the switch toggles after each contact on/off)	ESC A	G	f1 f2	down code	up code	text ...	NUL				'G': The touch fields f1 to f2 are defined for a switch. 'K': The area from xx1,yy1 to xx2,yy2 is defined as a switch. 'J': Image no. n1 is loaded to xx1,yy2 and defined as a switch. 'down code': (1-255) Return/touch macro when switched on. 'up code': (1-255) Return/touch macro when switched off. (down/up code = 0 on/off not reported). ´text´: A string that is centered with the current touch font in the touch key follows; multiline text is separated with the character '|' ($7C, dec: 124); 'NUL': ($00) = end of string	
		K	xx1 yy1	xx2	yy2	down code	up code	text ...	NUL			
		J	xx1 yy1	n1	down code	up code	text ...	NUL				
Define touch key with menu function	ESC A	M	xx1 yy1	xx2	yy2	down code	up code	mnu code	text ...	NUL	The area from xx1,yy1 to xx2,yy2 is defined as a menu key. 'down code':(1-255) Return/touch macro when pressed. 'up Code':(1-255) Return/touch macro when menu canceled 'mnu Code':(1-255) Return/menu macro+(item no. 1) after selection of a menu item. (down/up code = 0 activation/cancellation of the menu not reported). 'text':= string with the menu key text and the menu items. The different items are separated by the character '|' ($7C,dec:124) (e.g. "key|item1|item2|item3". The key text is drawn with the current touch font and the menu items are drawn with the current menu font. The background of the menu is saved automatically.	
Define drawing area	ESC A	D	xx1 yy1	xx2	yy2	n1					A drawing area is defined. You can then draw with a line width of n1 within the corner coordinates xx1,yy1 and xx2,yy2.	
Define free touch area*)	ESC A	H	xx1 yy1	xx2	yy2						A freely usable touch area is defined. Touch actions (down, up and drag) within the corner coordinates xx1,yy1 and xx2,yy2 are sent via RS232.	
Set bar by touch	ESC A	B	nr								The bar graph with the no. n1 is defined for input by touch panel.	
Touch: settings												
Touch frame	ESC A	E	n1								The frame type for the display of touch keys/switches is set with n1	1
Touch key response		I	n1								Automatic inversion when touch key touched: n1=0=OFF; n1=1=ON;	1
		S	n1								Tone sounds briefly when a touch key is touched: n1=0=OFF; n1=1=ON	1
Invert touch key		N	Code								The touch key with the assigned return code is inverted manually	
Query touch switch		X	Code								The status of the switch (off=0; on=1) is sent via the serial interface.	
Set touch switch		P	Code n1								The status of the switch is changed by means of a command n1=0=off; n1=1=on.	
Delete touch area		L	Code n1								The touch area with the return code (code=0: all touch areas) is removed from the touch query. When n1=0, the area remains visible on the display; when n1=1, the area is deleted from the display.	
Send bar value on/off		Q	n1								Automatic transmission of a new bar graph value by touch input is deactivated (n1=0) or activated (n1=1)	1
Touch query on/off		A	n1								Touch query is deactivated (n1=0) or activated (n1=1)	1
Touch: Label font												
Label font	ESC A	F	n1								Set font with the number n1 (1..16) for touch key label	5
Label zoom factor		Z	n1 n2								n1 = X zoom factor (1x..8x); n2 = Y zoom factor (1x..8x)	1,1
Add. line spacing		Y	n1								Insert n1 pixels between two lines of text as additional line spacing	
Label angle		W	n1								Text output angle: n1=0: 0°; n1=1: 90°; n1=2: 180°; n1=3: 270°;	0

*) *Only for Version 1.1 and higher*

TOUCH PANEL ADJUSTMENT

The EA KIT320-8xxTP has an analog, resistive touch panel. This touch panel is perfectly adjusted and immediately ready for operation on delivery. As a result of aging and wear, it may become necessary to readjust the touch panel.

Adjustment procedure:

1. Touch the touch panel at power-on and keep it depressed. After the message *„touch adjustment?“* appears, release the touch panel. Alternative to that issue the ‘ESC @’ command.
2. Touch the touch panel (again) within a second for at least one second.
3. Follow the instructions for adjustment (press 2 the points *upper left* and *lower right*).

EA KIT320: Command table for terminal mode									After reset
Command	**Codes**							**Remarks**	
Commands for terminal mode									
FF: Form feed (dec:12)	^L							The contents of the terminal area are deleted and the cursor is placed at pos. (1,1)	
CR: carriage return (d:13)	^M							Cursor to the beginning of the line on the extreme left	
LF: line feed (dec:10)	^J							Cursor is set to the next line	
Position cursor	ESC	T	P	n1	n2			n1=column; n2=line; origin upper-left corner (1,1)	1,1
Cursor on/off	ESC	T	C	n1				n1=0: Cursor is invisible; n1=1: Cursor flashes;	1
Terminal mode	ESC	T	M	n1				n1=0: Clear mode; n1=1: Overwrite mode; n1=2: Scroll mode	2
Autom. line feed	ESC	T	Z	n1				The automatic line feed is switched on (n1=1) or off (n1=0)	1
Terminal invisible	ESC	T	A					Terminal display not visible; outputs continue to be executed	
Terminal visible	ESC	T	E					Terminal display is visible again;	On
Redirect terminal outputs									
Suppress terminal	ESC	T	N					ASCII characters,FF,CR,LF are suppressed. Commands (ESC T) are executed	
Terminal output internal	ESC	T	I					All terminal outputs/commands affect the internal terminal of the EA KIT320	Intern
Terminal output external	ESC	T	X					All terminal outputs/commands affect the external dot-matrix module	
Settings for the internal terminal									
Set font	ESC	T	F	n1				Set font no. n1 (1..16) for terminal mode (monospaced fonts only)	5
Add. line spacing	ESC	T	Y	n1				n1 pixels are defined additionally for the current font as the line spacing	
Define window	ESC	T	W	xx1	yy1	xx2	yy2	w	The terminal output is executed only within the window from xx1,yy1 (=upper-left corner) to xx2,yy2 (=lower-right corner); xx=0..319; yy=0..239; w=angle (0=0°; 1=90°; 2=180°; 3=270°) of the terminal display	0..319 0..239
Settings for the external dot-matrix module (optionally to J6 or J7)									
Initialize dot-matrix module	ESC	T	D	n1	n2			Initialize an external dot-matrix display (HD44780 compatible) - n1 = number of characters; n2 = number of lines	

Responses of the EA KIT320 via the serial interface							
Id		**num**	**data**				**Remarks**
Automatic response from the KIT320							
ESC	A	1	code				Response from the analog touch panel when a key/switch is pressed. code = down or up code of the key/switch. Only transmitted if no touch macro is defined with the "down code" !
ESC	N	1	code				After a menu item is selected by touch, the selected menu item code is transmitted. Only transmitted if no touch macro is defined with the no. code !
ESC	P	1	value				After the input port is changed, the new 8-bit value is transmitted. The automatic port scan must be activated. See the 'ESC Y A n1' command. It is only transmitted when there is no port macro defined with the no. value!
ESC	M	1	nr				When a keystroke of the external matrix keyboard is detected, the newly pressed key number nr is transmitted. Only transmitted if no touch macro is defined with the no. nr!
ESC	B	2	nr	value			When a bar graph is set by touch, the current value of the bar is transmitted with nr. Transmission of the bar balue must be activated (see the 'ESC A Q r command).
ESC	H	5	Typ.	xLO	xHI	yLO	yHI	*) The following is transmitted in the case of a free touch area event: type=0 release; type=1 is touch; type=2 is drag within the free touch area at the x,y coordinates (16-bit values)
Response only when requested by command							
ESC	N	1	nr				After the 'ESC N S' command, the currently selected menu item is transmitted nr=0: no menu item is selected.
ESC	B	2	nr	value			After the 'ESC B S n1' command, the current value of the bar is transmitted w nr.
ESC	X	2	code	value			After the 'ESC A X' command, the current status of the touch switch is transmitted with code (the return code). value = 0 or 1
ESC	Y	2	nr	value			After the 'ESC Y R' command, the requested input port is transmitted. nr=0: value is an 8-bit binary value of all 8 inputs. nr=1..8: value is 0 or 1 dependin on the status of the input nr
ESC	D	2	LO-byt value	HI-byt value			After the 'ESC S D' command, the current analog value (max. 1/2 VDD) from the AIN input. (value = 0..1023 corresponds approximately to 0..2.5V)
ESC	E	num	data ...				After the 'ESC E R addr num' command, the requested bytes are transmitted from the user EEPROM.
ESC	I	num	data ...				After the 'ESC I R addr num' command, the requested bytes are transmitted from the 12C bus.
Response without ESC and length specification (num)							
B	M	+ 9660 bytes of image data					After the 'ESC H' command, 9662 bytes bytes are transmitted (=320x240 BM image). The first two bytes of the BMP image always begin with 'BM'
E	A	String ..				NUL	After the 'ESC S V' command, the version of the KIT firmware is transmitted a a string (end code is the character NUL = $00). The first two bytes of the stri always begin with 'EA'

TERMINAL MODE

When you switch the unit on, the cursor flashes in the first line, indicating that the display is ready for operation. All the incoming characters are displayed in ASCII format on the terminal (exception: CR,LF,FF,ESC,'#'). Line breaks are automatic or can be executed by means of the 'LF' character. If the last line is full, the contents of the terminal scroll upward. The 'FF' character (page feed) deletes the contents of the terminal display and positions cursor in the upper-left corner.

The terminal has its own layer for display and is thus completely independent of the graphical output; moreover, the size of the terminal window can be freely defined. If the graphics screen is deleted with 'ESC DL', for example, that does not affect the contents of the terminal window (the terminal level is deleted with 'FF').

The character '#' is used as an escape character (see below) and thus cannot be displayed directly on the terminal. If the character '#' is to be output on the terminal, it must be transmitted twice: '##'.

COMMAND TRANSFER/PARAMETERS

The operating unit unit can be programmed by means of various integrated commands. Each command begins with ESCAPE followed by one or two command letters and then parameters. There are two ways to transmit commands:

1. ASCII mode
- The ESC character corresponds to the character '#' (hex: $23, dec: 35).
- The command letters come directly after the '#' character.
- The parameters are transmitted as plain text (several ASCII characters) followed by a separating character (such as a comma ',').
- Strings (text) are written directly without quotation marks and concluded with CR (hex: $0D) or LF (hex: $0A).

2. Binary mode
- The escape character corresponds to the character ESC (hex: $1B, dec: 27).
- The command letters are transmitted directly.
- The coordinates xx and yy are transmitted as 16-bit binary values (fiirst the LOW byte and then the HIGH byte).
- All the other parameters are transmitted as 8-bit binary values (1 byte).
- Strings (text) are concluded with CR (hex: $0D) or LF (hex: $0A) or NUL (hex: $00).

No separating characters, such as spaces or commas, may be used in binary mode. The commands require **no final byte**, such as a carriage return (apart from the string $00).

PROGRAMMING EXAMPLE

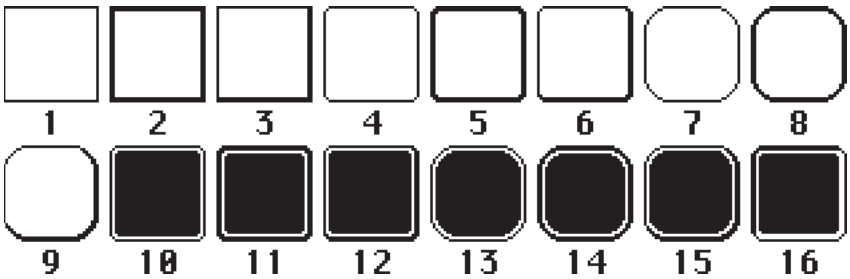
In the following example you can see how the string „Test“ can be output left justified at the coordinates 117,32.

Example	Codes can be output in ASCII mode	
for terminal.exe	#ZL117, 32,Test	<Return>
for Turbo-Pascal	write(aux, '#ZL117, 32,Test', chr(13));	
for ´C´	fprintf(stdaux, "#ZL%d,%d,%s\x0D", 117, 32, "Test");	
for Q-Basic	OPEN "COM1:9600,N,8,1,BIN" FOR RANDOM AS #1 PRINT #1,"#ZL117,32,Test"+CHR$(13)	

Example	Codes can be output in binary mode											
in ASCII	ESC	Z	L	u	NUL	space	NUL	T	e	s	t	NUL
in Hex	$1B	$5A	$4C	$75	$00	$20	$00	$54	$65	$73	$74	$00
in Decimal	27	90	76	117	0	32	0	84	101	115	116	0
for Turbo-Pascal	write(aux, chr(27), 'Z', 'L', chr(117), chr(0), chr(32), chr(0), 'Test', chr(0));											
for ´C´	fprintf(stdaux, "\x1BZL%c%c%c%c%s\x00", 117, 0, 32, 0, "Test");											
for Q-Basic	OPEN "COM1:9600,N,8,1,BIN" FOR RANDOM AS #1 PRINT #1,CHR$(27)+"ZL"+CHR$(117)+CHR$(0)+CHR$(32)+CHR$(0)+"Test"+CHR$(0)											

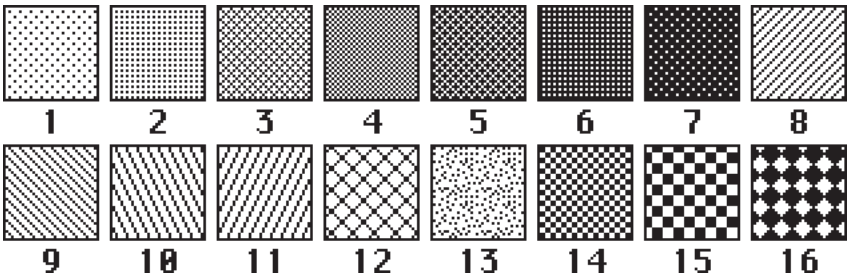
FRAMES AND KEY FORMS

A frame type can be set by using the *Draw frame* or *Draw frame box* command or by drawing touch keys. There are 16 internal frame types available; in addition, some frame types can be integrated by means of the kit compiler.



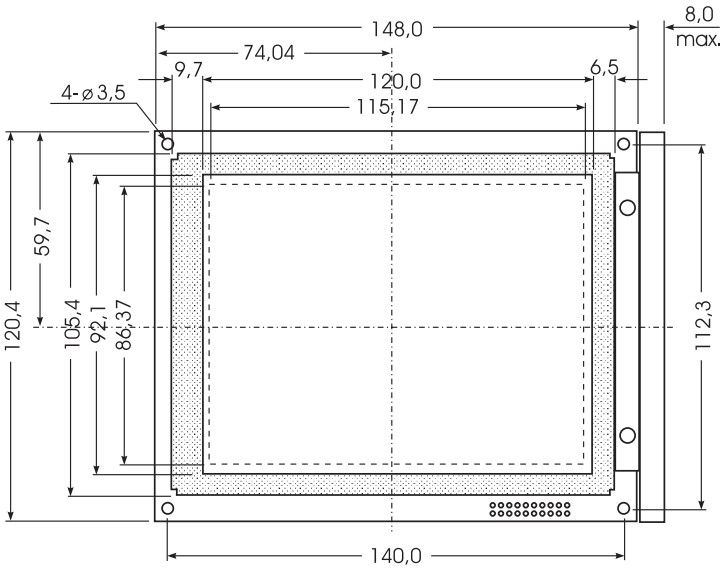
PATTERN

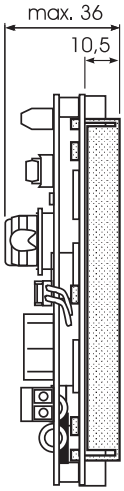
A pattern type can be set as a parameter with some commands. In this way, rectangular areas, bar graphs and even text can be filled with different patterns. There are 16 internal fill patterns available; in addition, some fill patterns can be integrated by means of the kit compiler.

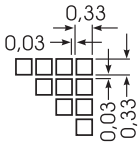


EA KIT320-8

DIMENSIONS W/O TOUCH PANEL

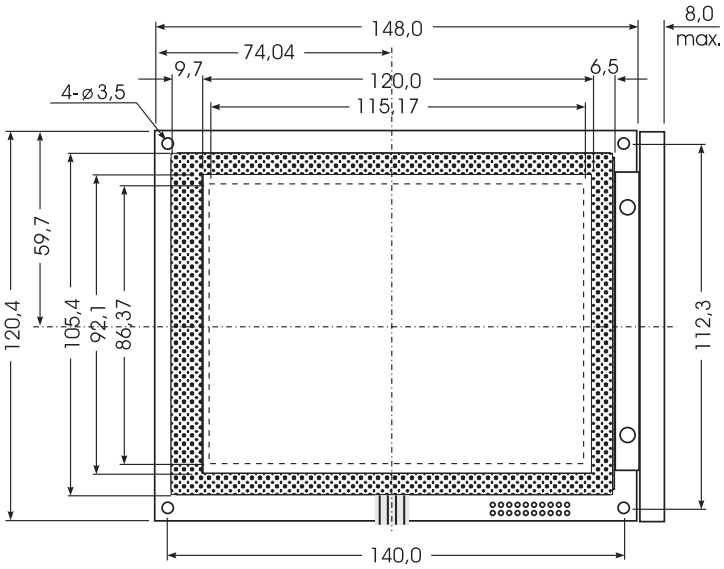


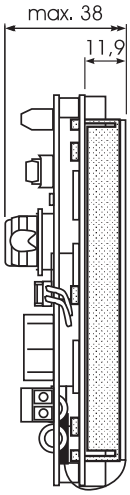


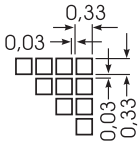


all dimensions are in mm

DIMENSIONS WITH TOUCH PANEL

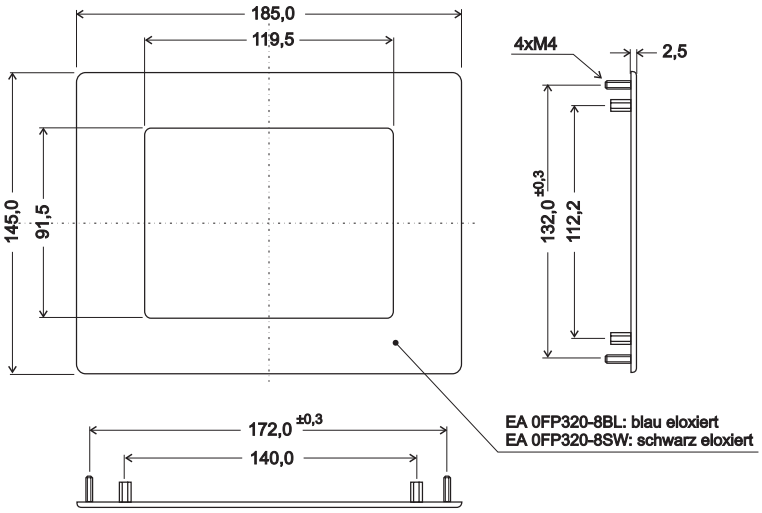




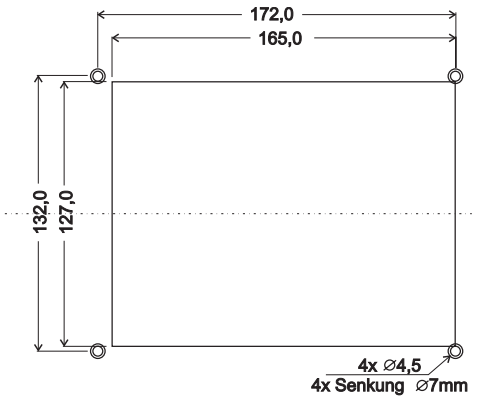


all dimensions are in mm

FRONTPANEL EA 0FP320-8



PANEL CUT OUT



all dimensions are in mm

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

ELECTRONIC ASSEMBLY GMBH