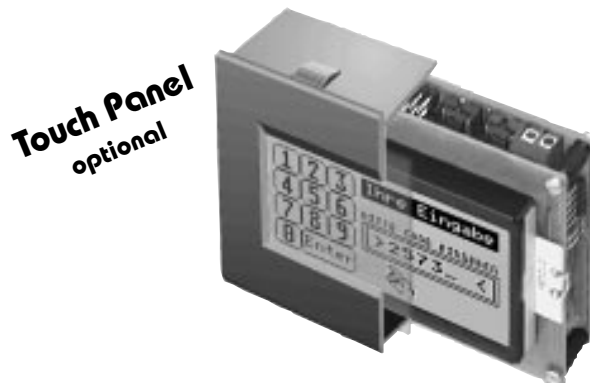


# GRAPHIC UNIT 128x64 DOTS

## WITH CHARACTER SETS AND GRAPH COMMANDS



*Housing size:  
102 x 80 x 36mm*

### FEATURES

- \* 128x64 DOTS SUPERTWIST-LCD WITH LED- ILLUMINATION (GREEN)
- \* BLACK SNAP-IN HOUSING WITH ANTIREFLEX GLASS
- \* TEXT MODE WITH SCROLL FUNCTION AND CR/LF EVALUATION
- \* 10 FONTS FROM 3,1mm (8x32 CHAR.) TO 4,2mm (8x21) UP TO 16,6mm (2x8)
- \* DRAWING LINES; ERASE AND INVERT AREAS
- \* CONNECTOR FOR 8x8 MATRIX PAD (EA KIT128-V24 AND -422 ONLY)
- \* SUPPLY VOLTAGE: +5V / typ. 45mA (EA KIT128-422 TYP. 160mA)
- \* BAUDRATES 1200,2400,4800,9600,19200BD (EA KIT128-V24,-422,-GXT ONLY)
- \* ADRESSABLE: SEVERAL DISPLAYS ONTO A SINGLE INTERFACE LINE
- \* GXT- VERSION: DISPLAY OF UP TO 256 TEXT BLOCKS OR GRAPHICS STORED IN EPROM OR OPTIONAL FLASH EPROM (OPT-FLASH128)
- \* GXT- VERSION: SELECTION THROUGH 8 LINES (SCREW TERMINAL 8 PIN)

### OPTIONS

- \* 8x4 TOUCH PANEL FOR VERSIONS RS-232C AND RS-422
- \* FOR 24V SUPPLY: OPT-18/36V, I.E. FOR SPS APPLICATIONS (+18..+36V)

### ORDERING INFORMATION

GRAPHIC UNIT FOR V.24/RS-232 WITH KEYBOARD CONN.  
 FOR RS-232C WITH TOUCH PANEL 8x4  
 FOR RS-422 WITH KEYBOARD CONNECTOR  
 FOR RS-422 WITH TOUCH PANEL 8x4  
 WITH SERIAL SYNCHRONOUS INTERFACE  
 FOR 8-BIT DATABUS OR CENTRONICS  
 TEXT- /GRAPHICS DISPLAY FOR 256 TEXTS / GRAPHICS  
 TOUCH PANEL 8X4 FOR VERSION -V24 AND -422  
 SUB-D PLUG CABLE FOR PC (RS-232) CONNECTION  
 UTILITY DISC FOR MACROS AND -GXT

**EA KIT128-V24**  
**EA KIT128-V24TP**  
**EA KIT128-422**  
**EA KIT128-422TP**  
**EA KIT128-SYNC**  
**EA KIT128-BUS**  
**EA KIT128-GXT**  
**EA TOUCH8X4-A**  
**EA KV24-9B**  
**EA DISK9719**

**ELECTRONIC**  
**ASSEMBLY** GMBH

LOCHHAMER SCHLAG 17 · D- 82 166 GRÄFELFING  
 TELEFON 089/854 1991 · TELEFAX 089/854 1721

| GUIDE   | FOR VERSION          | PAGE |
|---|----------------------|------|
| ORDER DESCRIPTION .....                           | ALL .....            | 1    |
| VERSIONS / DIFFERENCE .....                       | ALL .....            | 3    |
| TERMINAL, COMMANDS .....                          | ALL .....            | 4,5  |
| MACROS .....                                      | ALL .....            | 5    |
| CHARACTER SETS, USER DEFINED CHARACTERS .....     | ALL .....            | 6    |
| CONNECTION TO 8-BIT BUS .....                     | -BUS .....           | 7    |
| CONNECTION TO SERIAL, SYNCHRONOUS INTERFACE ..... | -SYNC .....          | 7    |
| CONNECTION TO RS-232C / V.24 .....                | -V24, -V24TP .....   | 8    |
| CONNECTION TO RS-422 .....                        | -422, -422TP .....   | 8    |
| CONNECTION TO KEY PAD 8x8 .....                   | -V24, -422 .....     | 9    |
| TOUCH PANEL 8x4 .....                             | -V24TP, -422TP ..... | 9    |
| VERSION FOR TEXT BLOCKS .....                     | -GXT .....           | 10   |
| COMPILER FOR MACROS + -GXT .....                  | ALL .....            | 11   |
| ADRESSING .....                                   | ALL .....            | 11   |
| ADDITIONAL OUTPUTS .....                          | ALL .....            | 11   |
| DIMENSIONS .....                                  | ALL .....            | 12   |
| SOLDER BRIDGES / RESET .....                      | ALL .....            | 12   |

## INTRODUCTION

The display kit EA KIT128 features a compact and an extremely simple programmable display unit. It shows simultaneously various fonts in different sizes. Graphic elements like company logos, icons and symbols are quickly created and displayed with comfortable instructions by graphic programs, i.e. Windows Paint. Also measurement graphs, diagrams and trend curves are thanks to the manyfold available graph commands quickly created.

Remarkable is the Cyrillic character set in 2 sizes, also the feature to connect either a matrix pad or a touch panel (only versions -V24 and -422 have an input terminal for key board connection).

## PUTTING INTO OPERATION, QUICK START UP

Before connecting the module please read first the instructions for operating and handling the module on last page of this data sheet.

Modul supply is +5 Volts/ max. 410mA which are connected to a 2 pin screw terminal block. After power on supply voltage the cursor will flash in the upper left hand corner. Adjustment of display contrast is possible by screw driver on a built in potentiometer.

The module can be connected now with the main processor unit according to the type of module version (see pages 7-10). For connecting the EA KIT128-V24- display, ELECTRONIC ASSEMBLY can deliver as an accessoir a ready to use cable type EA KV24-9B, which directly plugs into the 9-pol. SUB-D socket (i.e COM1) of the PC.

If data connection is correct, text's can be shown on display immediate (i.e. with assistance of the PC-program TERMINAL.EXE), available under purchase description EA DISK9719. Before using DOS command COPY xxxxx COM1, the interface must be initialised with MODE COM1:12,N,8,2. Various cursor controls like i.e. CR-LF are also possible. Switching to different character sizes or applying graph commands are described on page 4 and 5.

## ELECTRONIC ASSEMBLY

### SURVEY

#### EA KIT128-V24 with RS-232C interface and keyboard query

This graphic module can be connected directly by cable type EA KV24-9B to a PC- serial port. Baudrate is settable to 1200, 2400, 4800, 9600 or 19200 baud. Also a key pad (with max. 8x8 matrix) can be connected.

#### EA KIT128-422 for big distances with keyboard query

This graphic module can be connected to a RS-422-port and allows distances of up to 1200 meters. Baudrate is settable to 1200, 2400, 4800, 9600 or 19200 baud. A key pad with max. 8x8 matrix can be connected.

#### EA KIT128-BUS for a 8 bit processor bus or for centronics link

This graphic module can be connected to a fast processor bus. Available is a LOW- and a HIGH-active CS. Additional eight output lines for actuating purposes are accessible by commands.

#### EA KIT128-SYNC serial clocked for direct connection to $\mu$ C

This graphic module is designed especially for micro controller with a few outputs only. A clock- and data line transfers commands to the controller. There is a LOW- and a HIGH-active CS available. Additional eight output lines for actuating purposes are accessible by commands.

#### EA KIT128-GXT text- and graphic controller with 8 inputs and RS-232C

From non volatile memory may be called off up to 256 texts/graphics through 8 data lines. Texts/graphics are stored in an EPROM/ FLASH-EPROM. The lines are high-active and are on internal Pull-Down (100 k $\Omega$ ) resistors and Z-Diodes. This inputs can be operated by potentialfree switches. Direct connection to +5V or to +24V (SPS) level is allowed as well. Eight additional outputs for various control purposes can be used by commands. For module type EA KIT128-GXT an utility program disk type EA DISK9719 is required.

| Summary EA KIT128 |                              |                   |              |                    |                 |                  |                         |                     |
|-------------------|------------------------------|-------------------|--------------|--------------------|-----------------|------------------|-------------------------|---------------------|
| Part No.          | Data Input                   | Keyboard connect. | integr Fonts | Additional Outputs | Current w/o LED | Current with LED | Connectors              | Features            |
| EA KIT128-V24     | RS-232C / V.24               | yes               | 10           | upon request       | 45 mA           | 300 mA           | 2 Pinsockets 2x5/2x8    | Wide area standard  |
| EA KIT128-422     | RS-422                       | yes               | 10           | upon request       | 160 mA          | 410 mA           | 2 Pinsockets 2x5/2x8    | For big distances   |
| EA KIT128-SYNC    | 1-Bit serial: Data, Clock    | no                | 10           | yes                | 20 mA           | 270 mA           | 1x10 Pinsocket          | For microcontroller |
| EA KIT128-BUS     | 8-Bit parallel               | no                | 10           | yes                | 20 mA           | 270 mA           | 1x16 Pinsocket          | For microcontroller |
| EA KIT128-GXT     | Texts/Graphs through 8 lines | no                | 10           | yes                | 45 mA           | 300 mA           | 8 Terminal screw blocks | Stand-Alone / SPS   |

### EA DISK9719, UTILITY PROGRAMS

With order description EA DISK9719 a disc for PC's will be delivered, which contents several utility programs for own developments. On disc are a terminal program for DOS, a font editor including sample character sets and a font conversion program for converting into character set format of EA KIT128. Also available on disk is a conversion program for converting Windows Bitmap-Graphics (\*.BMP) into image format of EA KIT128. A compiler for creating macros and scripts resp. text- and graphic- blocks are on disc as well.

### TOUCH PANEL 8x4 (EA KIT128-V24TP AND -422TP ONLY)

The versions EA KIT128-V24TP and -422TP features a transparent touch pad with 32 fields. With touching the display datas can be transferred or actions are triggered. An inside buzzer acknowledges key strokes.

### EA OPT-18/36V

All versions with option EA OPT-18/36V can be operated on unstabilized supply voltage between 18V to 36V. Please make sure that the voltage on all display inputs does not exceed 5Volts (except EA KIT128-GXT with 4V to 50V inputs and all RS-232C interfaces with  $\pm$ 12V levels).

## TEXT / GRAPH / INSTRUCTIONS POOL

After power on the cursor flashes in first line and the module is ready for receiving. All incoming characters are shown in ASCII format (exception: CR, LF, FF, ESC). Carriage return is done automatically or with CR/LF- command. When display is filled, page scrolls to the top line by line. Command 'FF' (form feed) clears the display.

All additional commands (position cursor, select font, graph instructions, ...) must be started by an ESCAPE (hex 1B) followed by a command letter (capital- and small letters are equal) and some parameters more. Parameter handover is made in 8-bit binary form (not in ASCII).

| Table of commands (all versions) |       |   |                  |       |       |   |  |    |    |    |  |
|----------------------------------|-------|---|------------------|-------|-------|---|--|----|----|----|--|
| Command                          | Codes |   |                  |       |       | Description   |  |    |    |    |  |
| <b>Commands for text mode</b>    |       |   |                  |       |       |   |  |    |    |    |  |
| Form feed (FF)                   | ^L    |   |                  |       |       | Clears display and moves cursor to position (1,1)   |  |    |    |    |  |
| Carriage Return (CR)             | ^M    |   |                  |       |       | Cursor to left hand side  |  |    |    |    |  |
| Line feed (LF)                   | ^J    |   |                  |       |       | Cursor to 1 line below, with evtl. scrolling  |  |    |    |    |  |
| Cursor On / Off                  | ESC   | C | n1               |       |       | n1=0: hidden cursor; n1=1: cursor flashes   |  |    |    |    |  |
| Position cursor                  | ESC   | O | n1               | n2    |       | n1=column; n2=line; origin top left is (1,1)  |  |    |    |    |  |
| Inverse characters               | ESC   | J | n1               |       |       | n1=0: normal characters; n1=1: inverse characters   |  |    |    |    |  |
| Autom. line break up             | ESC   | W | n1               |       |       | n1=0: Autom. line break up off; n1=1: on  |  |    |    |    |  |
| <b>Graph commands</b>            |       |   |                  |       |       |   |  |    |    |    |  |
| Character-mode                   | ESC   | V | n1               |       |       | n1: 1=set; 2=clear; 3=invers; 4=replace; 5=invers replace;  |  |    |    |    |  |
| Set ASCII-characters             | ESC   | A | x1               | y1    | n1    | Character n1 will be placed to coordinate   |  |    |    |    |  |
| Display character chain          | ESC   | Z | x1               | y1    | ...   | NUL   | Display character chain(...); character 'NUL' (Binary 0) = End |    |    |    |  |
| Set point                        | ESC   | P | x1               | y1    |       | Coordinates (0,0) to (127,63)   |  |    |    |    |  |
| Draw line                        | ESC   | G | x1               | y1    | x2    | y2  | Beginning- and end coordinates (0,0) to (127,63)               |    |    |    |  |
| Continue line draw               | ESC   | T | x1               | y1    |       | Draw line from last end point to x1, y1   |  |    |    |    |  |
| Draw box                         | ESC   | R | x1               | y1    | x2    | y2  | Opposite corner points (0,0) to (127,63)                       |    |    |    |  |
| Clear Area                       | ESC   | L | x1               | y1    | x2    | y2  | Opposite corner points (0,0) to (127,63)                       |    |    |    |  |
| Invert Area                      | ESC   | I | x1               | y1    | x2    | y2  | Opposite corner points (0,0) to (127,63)                       |    |    |    |  |
| Fill Area                        | ESC   | S | x1               | y1    | x2    | y2  | Opposite corner points (0,0) to (127,63)                       |    |    |    |  |
| Define Bargraph                  | ESC   | - | L<br>R<br>O<br>U | Nr    | x1    | y1  | x2   | y2 | AW | EW | Define Bar to L(left),R(right),O(top),U(bottom) with 'No' (1..8). x1,y1,x2,y2 surrounds the rectangular of bargraph. AW, EW represents the values for 0% and 100%. |
| Draw Bargraph                    | ESC   | - | Nr               | Wert  |       | Set Bargraph with 'No' (1..8) to new user-'value'   |  |    |    |    |  |
| Load Bitmap                      | ESC   | B | x1               | y1    | Daten | Load display range to x1,y1; see display build up for datas   |  |    |    |    |  |
| <b>Other commands</b>            |       |   |                  |       |       |   |  |    |    |    |  |
| Set Font                         | ESC   | F | n1               |       |       | n1=1: Font No. n1 (1..8)  |  |    |    |    |  |
| Wide characters                  | ESC   | E | n1               |       |       | n1=0: normal character size; n1=1: wide characters  |  |    |    |    |  |
| Define characters                | ESC   | U | n1               | Daten |       | n1=Character No.; Datas=No.of Bytes according to Font   |  |    |    |    |  |
| Display commands                 | ESC   | D | n1               |       |       | n1=0:Off;n1=1:On;n1=2:Clear.;n1=3:Invert.;n1=10..20:LED-brightness  |  |    |    |    |  |
| Illumination On / Off            | ESC   | H | n1               |       |       | n1=0: Illum. Off; n1=1: Illum. On; n1=2..255: for n1 seconds On   |  |    |    |    |  |
| Select Controller                | ESC   | K | S                | n1    |       | Activate controller with adress n1 (n1=0..254; n1=255: all)   |  |    |    |    |  |
| Deselect Controller              | ESC   | K | D                | n1    |       | Deactivate controller with adress n1 (n1=0..254; n1=255: all)   |  |    |    |    |  |
| Select/Deselect Adress           | ESC   | K | A                | n1    |       | Define new adress n1 for controller (i.e for Power-On Makro)  |  |    |    |    |  |
| Set output port                  | ESC   | Y | n1               | n2    |       | n1=0: Set all 8 outputs according to n2 (=8-Bit binary value)<br>n1=1..8: Reset output n1 (n2=0); set (n2=1); invert (n2=2) |  |    |    |    |  |
| Wait (Pause)                     | ESC   | ! | n1               |       |       | Wait n1 tenth-seconds until next output shows up  |  |    |    |    |  |
| Status Check                     | ESC   | ? | C                | n1    |       | n1=0: Signal on RS232/422: 'O'=OK; 'E' or no reaction=ERROR<br>n1=1..8: OUT1..8 will be inverted=OK; no reaction=ERROR      |  |    |    |    |  |
| <b>Macro commands</b>            |       |   |                  |       |       |   |  |    |    |    |  |
| Run Script                       | ESC   | Q | n1               |       |       | Process Script n1 (max. 2 levels)   |  |    |    |    |  |
| Run Macro                        | ESC   | N | n1               |       |       | Process Macro n1 (max. 2 levels)  |  |    |    |    |  |
| Autom. Run of Makros             | ESC   | X | n1               | n2    | n3    | Cyclic Processing of Macros from n1 to n2 (n3=pause in 1/10 sec)  |  |    |    |    |  |

| Additional commands for -V24 and -422               |     |   |                  |           |           |      |   |
|---|-----|---|------------------|-----------|-----------|------|---|
| Query matrix key pad                                | ESC | M | n1               |           |           |      | n1=0: Automatic query OFF<br>n1=1: Automatic query total status ON<br>n1=2: Automatic query single key mode ON<br>n1=3: Aktuellen Zustand der Matrixastatur senden                            |
| Matrix key pad blank out                            | ESC | M | 4                | n1        |           |      | In order to use the output ports with command 'ESC Y', the matrix key pad query cannot be used. A '0' within the 8-Bits of parameter n1 deactivates the specified output line.                |
| Transmit graph                                      | ESC | > |                  |           |           |      | The complete content of graph will be transmitted by a leading 'B' mark, followed from height and width in number of dots and the graph datas.  |
| Transmit bytes                                      | ESC | = | n1               | ...       |           |      | After n1 following bytes are transmitted on V.24 (i.e. driving an external serial printer on RS232-C interface)   |
| Read output port                                    | ESC | ? | Y                | n1        |           |      | n1=0: read all 8 outputs OUT1..8 in 8-bit binary form<br>n1=1..8: read outputs OUT<n1> (1=high=5V, 0=low=0V)  |
| Commands for operating Touch-Panels (EA TOUCH8x4-A) |     |   |                  |           |           |      |   |
| Define touch key                                    | ESC | * | f1               | f2        | Ret. Code | Form | Displ. text   |
| Touch key reaction                                  | ESC | * | A<br>B<br>C<br>D |           |           |      | A:= no buzz when touching the touch key<br>B:= short buzz when touching the touch key<br>C:= non inverting when touching the touch key<br>D:= automatic inverting when touching the touch key |
| Invert touch key                                    | ESC | * | I                | Ret. Code |           |      | Touch key will be inverted with the related return code   |
| Reset touch key                                     | ESC | * | R                |           |           |      | Resets all touch keys (= 32 fields with codes 1..32)  |
| Buzz on / off                                       | ESC | . | n1               |           |           |      | n1=0: buzz off; n1=1: buzz on; n1=2..255: for n1 1/10sec. on. l   |

| Additional commands for -GXT |     |   |    |     |  |  |  |
|------------------------------|-----|---|----|-----|--|--|--|
| Get Display Data             | ESC | > |    |     |  |  | Complete content of graph will be transmitted. Transmission starts with a 'B' followed by height and width in pixel and the graph datas. |
| Transmit bytes               | ESC | = | n1 | ... |  |  | n1- Bytes will be sent in sequence on V.24 (i.e. driving an external serial RS232-C printer interface)                                   |
| Read output port             | ESC | ? | Y  | n1  |  |  | n1=0: All 8 output ports OUT1..8 read as 8-Bit Binary value<br>n1=1..8: output ports OUT<n1> read in (1=high=5V, 0=low=0V)               |

### EXAMPLE FOR APPLYING COMMANDS

Below examples in various programming languages will draw a rectangle on display.

| Example          | Codes output   |
|------------------|--|
| for compiler     | #R 0, 0, 127, 63   |
| in hex           | \$1B \$52 \$00 \$00 \$7F \$3F  |
| in decimal       | 27 82 0 0 127 63   |
| for Turbo-Pascal | write(aux, chr(27), 'R', chr(0), chr(0), chr(127), chr(63));   |
| for 'C'          | fprintf(stdaux, "%c%c%c%c%c%c", 27, 'R', 0, 0, 127, 63);   |
| for Q-Basic      | OPEN "COM1:1200,N,8,2,BIN" FOR RANDOM AS #1<br>PRINT #1,CHR\$(27)+"R"+CHR\$(0)+CHR\$(0)+CHR\$(127)+CHR\$(63) |



### MACROS

Several commands can be combined to a macro and can be executed by command *Run Macro*. Macros will be created with the compiler on disk EA DISK9719 and will be burned with an EPROMer into EPROM of EA KIT128.

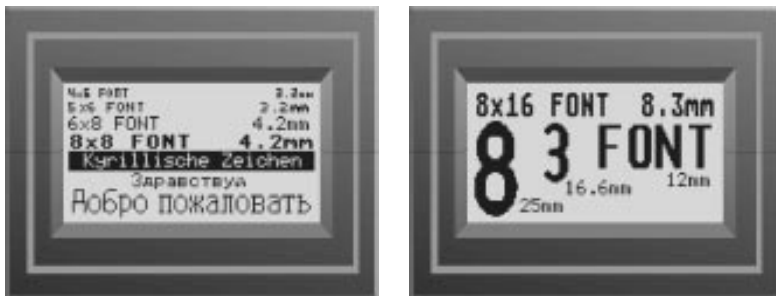
Power-On-Macro will be executed automatically immediate after supply voltage is available. Also commands like *Cursor off* (ESC C 0), set *Font 3* (ESC F 3), *Baudrate 9600 Baud* (ESC + 3<sup>\*)</sup>. A short description can be found on page 11.

<sup>\*)</sup> Setting baudrate with software is possible only when all 4 solder jumpers LB1..4 are open.

## CHARACTER SETS

Display module EA KIT128 contents 10 character sets. All of them are available in text mode as well as in graph mode. Graph mode allows character positioning pixel by pixel (commands 'ESC A' and 'ESC Z') where coordinate- values refer to top left hand corner.

| No. | Character height | Lines x Character | Size in Dots | ASCII-Range | Free def. ASCII-Code(s) | Remarks                          |
|-----|------------------|-------------------|--------------|-------------|-------------------------|----------------------------------|
| 1   | 4,2 mm           | 8 x 21            | 6 x 8        | 32 - 255    | 0..9                    | extended ASCII-Code              |
| 2   | 4,2 mm           | 8 x 16            | 8 x 8        | 32 - 255    | 0..7                    | extended ASCII-Code              |
| 3   | 8,3 mm           | 4 x 16            | 8 x 16       | 32 - 255    | 0..3                    | extended ASCII-Code              |
| 4   | 3,1 mm           | 8 x 32            | 4 x 6        | 32 - 158    | 0..15                   | Micro Characters                 |
| 5   | 3,1 mm           | 8 x 25            | 5 x 6        | 32 - 158    | 0..11                   | Mini Characters                  |
| 6   | 16,6 mm          | 2 x 8             | 16 x 32      | 42 - 58     | 0                       | Big Numbers '0'-'9', '*+', '-./' |
| 7   | 16,6 mm          | 2 x 8             | 16 x 32      | 32 - 127    | 0                       | ASCII-Code                       |
| 8   | 25 mm            | 1 x 5             | 24 x 48      | 45 - 58     | -                       | Super Big Numbers '0'-'9', '-./' |
| 9   | 4,2 mm           | 8 x 21            | 6 x 8        | 64 - 128    | 0..9                    | Cyrillic Characters              |
| 10  | 8,3 mm           | 4 x 16            | 8 x 16       | 64 - 128    | 0..3                    | Cyrillic Characters              |



## CUSTOMIZED CHARACTERS

Up to 16 characters per character set can be defined by your own, depending on size of character. These characters are in form of ASCII codes 0..15 and are stored in a 64 Byte sized internal RAM- area until supply voltage is switched off.

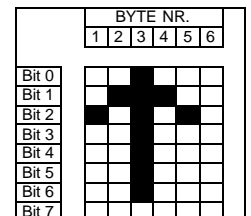
**Caution:** When own characters are created, do not execute the bargraph command.

| + Lower         | \$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) |
|-----------------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|----------|----------|----------|----------|----------|----------|
| Upper           |         |         |         |         |         |         |         |         |         |         |          |          |          |          |          |          |
| \$20 (dez: 32)  |         | !       | "       | #       | \$      | %       | &       | '       | (       | )       | *        | +        | ,        | -        | .        | /        |
| \$30 (dez: 48)  | 0       | 1       | 2       | 3       | 4       | 5       | 6       | 7       | 8       | 9       | :        | ;        | <        | =        | >        | ?        |
| \$40 (dez: 64)  | a       | b       | c       | d       | e       | f       | g       | h       | i       | j       | k        | l        | m        | n        | o        |          |
| \$50 (dez: 80)  | p       | q       | r       | s       | t       | u       | v       | w       | x       | y       | z        | [        | \        | ]        | ^        | _        |
| \$60 (dez: 96)  | `       | a       | b       | c       | d       | e       | f       | g       | h       | i       | j        | k        | l        | m        | n        | o        |
| \$70 (dez: 112) | p       | q       | r       | s       | t       | u       | v       | w       | x       | y       | z        | [        | \        | ]        | ^        | _        |
| \$80 (dez: 128) | 0       | 1       | 2       | 3       | 4       | 5       | 6       | 7       | 8       | 9       | :        | ;        | <        | =        | >        | ?        |
| \$90 (dez: 144) | €       | £       | ¥       | ¢       | ¤       | ₹       | ₠       | ₡       | ₢       | ₣       | ₤        | ₥        | ₦        | ₧        | ₨        | ₪        |
| \$A0 (dez: 160) | á       | â       | ã       | ä       | å       | æ       | ç       | è       | é       | ê       | ë        | ì        | í        | î        | ï        | ñ        |
| \$B0 (dez: 176) | ::      | ;       | <       | =       | >       | ?       | !       | "       | #       | \$      | %        | &        | '        | (        | )        | *        |
| \$C0 (dez: 192) | L       | T       | T       | T       | T       | T       | T       | T       | T       | T       | T        | T        | T        | T        | T        | T        |
| \$D0 (dez: 208) | À       | Á       | Â       | Ã       | Ä       | Å       | Æ       | Ç       | È       | É       | Ê        | Ë        | Ì        | Í        | Î        | Ï        |
| \$E0 (dez: 224) | à       | á       | â       | ã       | ä       | å       | æ       | ç       | è       | é       | ê        | ë        | ì        | í        | î        | ï        |
| \$F0 (dez: 240) | ë       | +       | z       | €       | £       | ¥       | ¢       | ¤       | ₹       | ₠       | ₡        | ₢        | ₣        | ₤        | ₥        | ₦        |

*Example 1:*

instruction for ASCII-no.3, with 6x8 character set will create an upward pointing arrow.

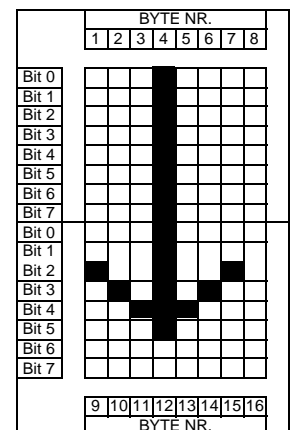
**ESC U \$03**  
**\$04 \$02 \$7F \$02 \$04 \$00**



*Exampel 2:*

instructions below creates a downward pointing arrow with ASCII-no.2 and 8x16 character set.

**ESC U \$02**  
**\$00 \$00 \$00 \$FF \$00 \$00 \$00 \$00**  
**\$04 \$08 \$10 \$3F \$10 \$08 \$04 \$00**



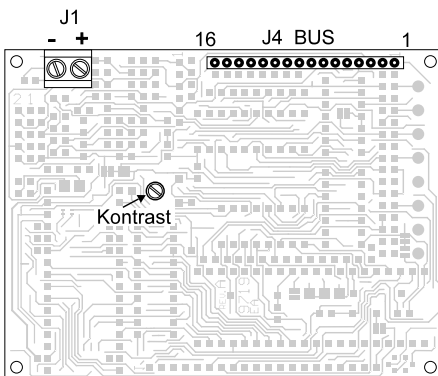
Font 3

| + Lower        | \$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) |
|----------------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|----------|----------|----------|----------|----------|----------|
| Upper          |         |         |         |         |         |         |         |         |         |         |          |          |          |          |          |          |
| \$20 (dez: 32) |         |         |         |         |         |         |         |         |         |         | x        | +        | ,        | -        | .        | /        |
| \$30 (dez: 48) | 0       | 1       | 2       | 3       | 4       | 5       | 6       | 7       | 8       | 9       | :        | ;        | <        | =        | >        | ?        |

Font 6

## ELECTRONIC ASSEMBLY

### CONNECTING EA KIT128-BUS TO AN 8 BIT DATA BUS



Bottom view

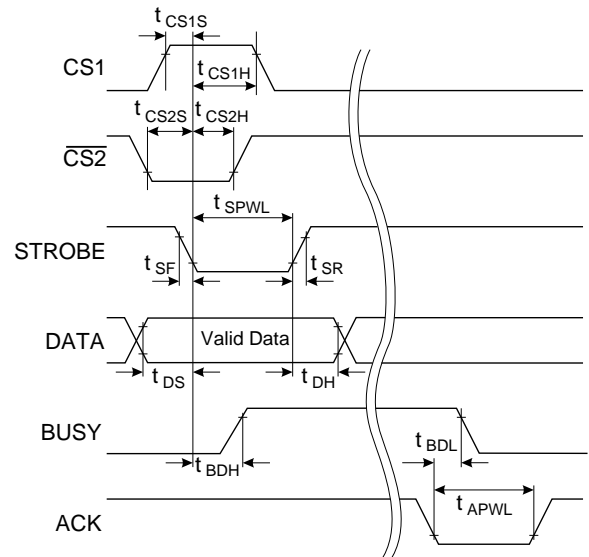
The display module EA KIT128-BUS is suitable for direct connection to an 8-bit data bus of a microprocessor system. Data handover happens at the L-H rise on pin "STROBE". Output "BUSY" marks with H-level that no instructions will be accepted anymore. When display EA KIT128-BUS has finished processing, pin "ACK" shows low level for short time, which allows i.e. trigger of an interrupt. Both inputs

CS1 and CS2 can be fed by the address decoder or are connected to VDD resp. to GND. Current consumption is 20mA typically, and typ. 270mA with LED illumination on.

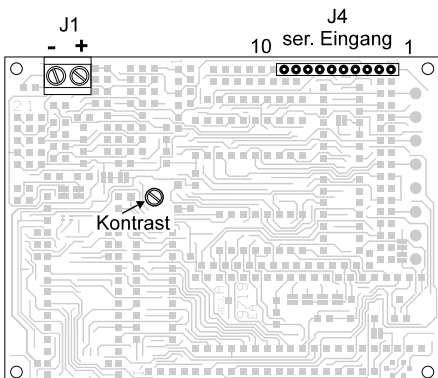
| BUS Connector J4 |        |       |                  |
|------------------|--------|-------|------------------|
| Pin              | Symbol | Level | Function         |
| 1                | CS1    | H     | Chip select high |
| 2                | /CS2   | L     | Chip select low  |
| 3                | ACK    | L     | Acknowledge      |
| 4                | NC     | -     | No Connect       |
| 5                | BUSY   | H     | In Process       |
| 6                | STROBE | L     | Takeover Datas   |
| 7                | VDD    | H     | + 5V             |
| 8                | GND    | L     | 0V Ground        |
| 9                | DATA 7 | H / L | Bit 7 (MSB)      |
| 10               | DATA 6 | H / L | Bit 6            |
| 11               | DATA 5 | H / L | Bit 5            |
| 12               | DATA 4 | H / L | Bit 4            |
| 13               | DATA 3 | H / L | Bit 3            |
| 14               | DATA 2 | H / L | Bit 2            |
| 15               | DATA 1 | H / L | Bit 1            |
| 16               | DATA 0 | H / L | Bit 0 (LSB)      |

| Timing -BUS and -SYNC       |            |                |      |      |      |
|-----------------------------|------------|----------------|------|------|------|
| Item                        | Symbol     | Standard Value |      |      | Unit |
|                             |            | min.           | typ. | max. |      |
| Chip Select 1 Setup Time    | $t_{CS1S}$ | 25             |      |      | ns   |
| Chip Select 1 Hold Time     | $t_{CS1H}$ | 0              |      |      | ns   |
| Chip Select 2 Setup Time    | $t_{CS2S}$ | 34             |      |      | ns   |
| Chip Select 2 Hold Time     | $t_{CS2H}$ | 0              |      |      | ns   |
| Strobe Pulse Width Low      | $t_{SPWL}$ | 20             |      |      | ns   |
| Strobe Fall Time            | $t_{SF}$   |                | 5    |      | ns   |
| Strobe Rise Time            | $t_{SR}$   |                | 5    |      | ns   |
| Data Setup Time             | $t_{DS}$   |                | 6    | 12   | ns   |
| Data Hold Time              | $t_{DH}$   |                | 0    | 4    | ns   |
| Busy Delay Time High        | $t_{BDH}$  |                | 20   |      | ns   |
| Busy Delay Time Low         | $t_{BDL}$  |                | 20   |      | ns   |
| Acknowledge Pulse Width Low | $t_{APWL}$ |                | 1100 |      | ns   |

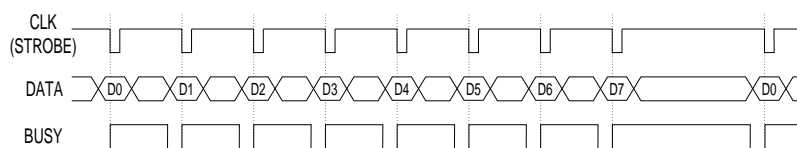
$V_{DD} = 5,0V \pm 5\%$   
 $T_a = 25^\circ C$



### SYNCHRONOUS VERSION EA KIT128-SYNC



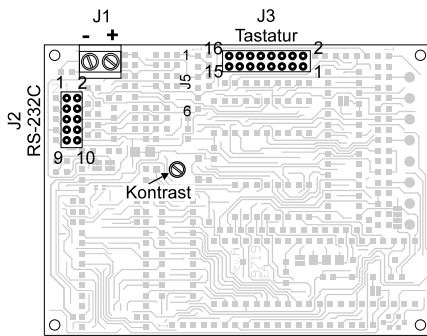
Bottom view



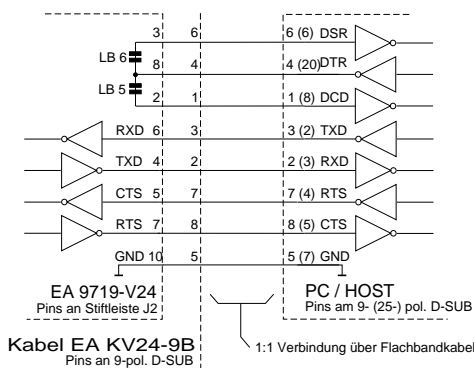
Module EA KIT128-SYNC is designed for connecting to a microcontroller. There is one clock- and one data line available. Communication is performed through 8 serial data bits, which can be transmitted with clock rates from 0...20 kcps. Output "BUSY" marks with H-level that further instructions won't be accepted. When display EA KIT128-SYNC has finished processing, pin "ACK" shows a short Low-level, which allows to trigger i.e. an interrupt. Timing is the same as on version EA KIT128-BUS. To synchronize the data stream, a Low Puls (>1µs) at input SYNC will reset the internal bit counter. Both inputs CS1 and CS2 can be fed by the address decoder or are connected to VDD resp. GND. Current consumption is 20mA typ., and typ. 270 mA with LED-illumination on.

| Synchron. serial connector J4 |        |       |                   |
|-------------------------------|--------|-------|-------------------|
| Pin                           | Symbol | Level | Function          |
| 1                             | CS1    | H     | Chip select high  |
| 2                             | /CS2   | L     | Chip select low   |
| 3                             | ACK    | L     | Acknowledge       |
| 4                             | SYNC   | L     | Bit counter reset |
| 5                             | BUSY   | H     | In Process        |
| 6                             | CLK    | L     | Clock input       |
| 7                             | VDD    | H     | + 5V Supply       |
| 8                             | GND    | L     | 0V Ground         |
| 9                             | DATA   | H / L | Data bit          |
| 10                            | NC     | -     | No Connect        |

## V.24/RS-232C VERSION EA KIT128-V24(TP)



Bottom view



Display module EA KIT128-V24(TP) is designed for connection to a bi-directional serial interface with true RS-232C levels ( $\pm 10V$ ). Option-cable EA KV24-9B can be plugged directly onto pin block J2. This cable has on other end a 9-pin, D-SUB plug, which fits COM1 of a PC. The module offers handshake lines RTS and CTS. A minus 10V level on pin RTS signals that no more data can be accepted until RTS shows a +10V level. On the other hand the EA KIT128-V24 blocks the data output if a -10V level is applied to pin CTS. In this case there is a possibility that keyboard inputs may get lost.

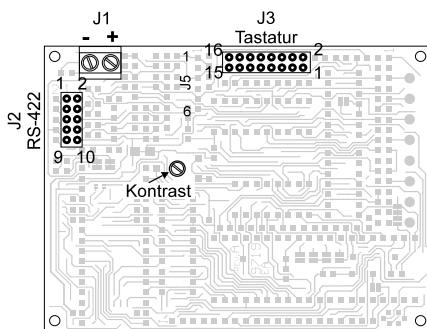
Transfer parameters for data are factory set to 8 data bits, 1 stop bit and 1200 Baud. Bridging solder jumpers 1 to 4 allows increased baud rates of 2400, 4800, 9600 oder 19200 Baud. Solder jumpers are described on last page.

| RS-232C connector J2 |        |        |                  |
|----------------------|--------|--------|------------------|
| Pin                  | Symbol | In/Out | Function         |
| 1                    | VDD    | -      | + 5V Supply      |
| 2                    | DCD    | -      | Via LB5 to DTR   |
| 3                    | DSR    | -      | Via LB6 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     | -      | No Connect       |
| 10                   | GND    | -      | 0V Ground        |

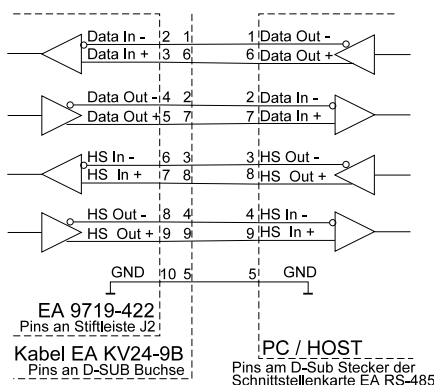
| CMOS-level connector J5 |        |                 |
|-------------------------|--------|-----------------|
| Pin                     | Symbol | Function        |
| 1                       | VDD    | + 5V Supply     |
| 2                       | GND    | 0V Ground       |
| 3                       | TxD    | Transmit Data   |
| 4                       | RxD    | Receive Data    |
| 5                       | RTS    | Request To Send |
| 6                       | CTS    | Clear To Send   |

**Note:** Only one solder jumper at a time may be closed. Never close two solder jumpers at a time! On 6-pin solder socket J5 the CMOS level of serial interface can be used for direct connection to micro processor. **Note:** The RS-232 level-converter SP232 (or equivalent) must be removed in this case! Current consumption is 45 mA typically, and typ.300 mA with LED illumination on.

## CONNECTING EA KIT128-422(TP)



Back view



On board RS-422 drivers generate differential voltages with approximate  $\pm 5V$  amplitudes. This guarantees an extremely safe transmission up to 19200 Baud even on very long lines (up to 1200 meters). Communication parameters are factory set on 8 data bits, 1 stop bit and 1200 baud. Shortening solder bridges 1 to 4 increases baudrate to 2400, 4800, 9600 or 19200 baud. Solder bridges are described on last page.

**Caution:** One solder bridge only may be shortened at a time. Never close 2 solder bridges at the same time! Data input takes connector J2 of EA KIT128-422(TP). Pinning is shown in table nearby. Typical current consumption is 160mA, with LED-illumination on typically 400mA.

**Note:** Solder bridges 5 + 6 must be open.

| RS-422 Connector J2 |            |               |
|---------------------|------------|---------------|
| 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    |

## ELECTRONIC ASSEMBLY

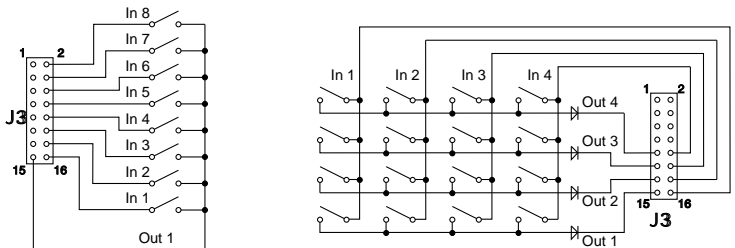
### KEYPAD OPERATION (EA KIT128-V24 AND EA KIT128-422 ONLY)

A keypad (single push buttons up to 8x8 matrix) can be connected to J3. Contact bounce is eliminated software. Please note, that keypad functions are supported by versions V.24 and RS-422 only.

Keys are connected to input- and output ports where each input port is shunted by a 100kΩ pullup resistor. Up to 8 keys may be connected to each output port. In order to recognize double key strokes, the output ports have to be decoupled. Best way for elimination is use of Schottky diodes (i.e.BAT 43). On multiple keystrokes (>2) each key must be decoupled by a separate diode.

| Matrix - keypad connector J3 |        |                 |     |        |                |
|------------------------------|--------|-----------------|-----|--------|----------------|
| Pin                          | Symbol | Function        | Pin | Symbol | Function       |
| 1                            | OUT 8  | Output column 8 | 2   | IN 8   | Input column 8 |
| 3                            | OUT 7  | Output column 7 | 4   | IN 7   | Input column 7 |
| 5                            | OUT 6  | Output column 6 | 6   | IN 6   | Input column 6 |
| 7                            | OUT 5  | Output column 5 | 8   | IN 5   | Input column 5 |
| 9                            | OUT 4  | Output column 4 | 10  | IN 4   | Input column 4 |
| 11                           | OUT 3  | Output column 3 | 12  | IN 3   | Input column 3 |
| 13                           | OUT 2  | Output column 2 | 14  | IN 2   | Input column 2 |
| 15                           | OUT 1  | Output column 1 | 16  | IN 1   | Input column 1 |

Mode 0: Command "ESC M 0" switches off the automatic keyboard request. Keystrokes won't be sent now automatically. Command "ESC M 3" queries the keys. After that command the actual keyboard status will be transmitted like in Mode 2.



Mode 1: Command "ESC M 1" activates the automatic query of keys. Each keypress. sends the corresponding key number (1..64). Key releasing is not sent (presetted).

Mode 2: Command "ESC M 2" activates the automatic query for the total status of all keys. On each key -press and - release all pressed keys are transmitted. Transmitting sequence is as follows: code letter "T" (\$54), followed by numbers of all pressed keys (1..64), ending code is "NUL" (\$00).

Number of key can be calculated as follows: **number of key = (output -1) \* 8 + input** (output/ input must be a number between 1 and 8) **Note !** If handshake line (i.e. CTS) does not allow transmitting, some key strokes may be lost.

### TOUCH-PANEL 8x4 (EA KIT128-V24TP AND -422TP ONLY)

The versions EA KIT128-V24TP and -422TP have got instead of an external 8x8 keypad a transparent touch key pad with 32 fields. The controller supports this touch panel with comfortable commands. It also allows to combine several keys to one big master key, or display a key image with centered text('s). Its also possible to assign a return code (1..255) to the key. If return-code 0 is assigned, the key will be deactivated and won't be recognized when stroked.

Touching the keyfield also can be automatically inverted or a buzz will signal the touch. The defined return code will be sent then simultaneously to serial interface or an internal SCRIPT with the number of the return code can be executed (dummy command 'ESC : ' must lead the script).

#### Example:

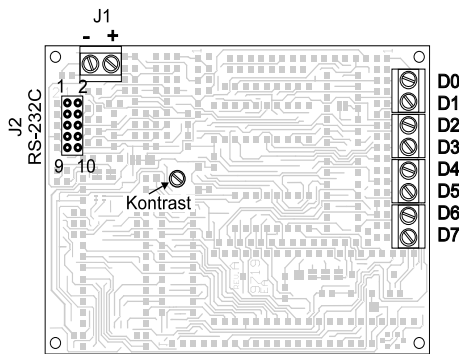
Define a key from field 11 to 21, with return-code 65='A' and the text "STOP"

|    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|
| 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  |
| 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 |
| 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 |

| Exampel      | Transmitted codes         |             |                             |                                 |             |            |             |      |      |      |      | Remarks                                  |   |
|--------------|---------------------------|-------------|-----------------------------|---------------------------------|-------------|------------|-------------|------|------|------|------|--|---|
| for Compiler | #* 11, 21, 'A', 2, "STOP" |             |                             |                                 |             |            |             |      |      |      |      | Number of texts will be not noted here ! |   |
| ASCII        | ESC                       | *           | .                           | .                               | A           | .          | .           | S    | T    | O    | P    | .  | Point '.' stands for not displayable ASCII-characters |
| Hex          | \$1B                      | \$2A        | \$0B                        | \$15                            | \$41        | \$02       | \$01        | \$53 | \$54 | \$4F | \$50 | \$00                                     |   |
| Decimal      | 27                        | 42          | 11                          | 21                              | 65          | 2          | 1           | 83   | 84   | 79   | 80   | 0  |   |
|              | Command for Touch         | Touch field | Touch field top left corner | Touch field bottom right corner | Return Code | Draw Frame | 1-line-text |      |      |      |      |  | Text terminator                                       |



## PERMANENT TEXT-/GRAPH VERSION EA KIT128-GXT



Bottom view

Display module EA KIT128-GXT is driven by 8 digital inputs. Each change on one of the inputs starts a script which is programmed into an EPROM. Inputs can be accessed by an 8 screw terminal block. Normal switches (closer) can be here connected as well as voltages between 0V and +50V. For creating a script you need a PC, the disc EA DISK9719 and an EPROMer or the option FLASH-128 for on board programming.

| Auswahl J6 |        |          |
|------------|--------|----------|
| Pin        | Symbol | Funktion |
| D0         | LSB    | Bit 0    |
| D1         |        | Bit 1    |
| D2         |        | Bit 2    |
| D3         |        | Bit 3    |
| D4         |        | Bit 4    |
| D5         |        | Bit 5    |
| D6         |        | Bit 6    |
| D7         | MSB    | Bit 7    |

### 1.) Connect to 8 switches

Each switch is connected to positive supply voltage and the input terminal block. An open input resp. an open switch represents a 0, this means, if no switches are connected or all switches are open, the script No. 0 will be executed. When all 8 switches are closed ("1"), script No. 255 will be executed.

### 2.) Connect to PLC/SPS or to CMOS-level

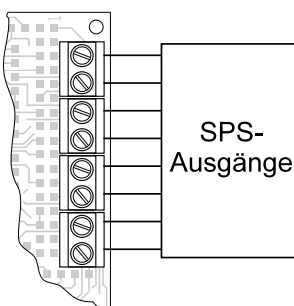
All 8 inputs will be connected to the corresponding voltage source. A common ground (minus) is necessary. When no voltage is applied, script No. 0 will be executed, when each gate is on min. +4 V script No. 255 will be executed.

Additional commands can be given by serial RS-232C interface connection see "V.24/RS-232C VERSION" on page 8. Typical current consumption is 45mA, with LED-illumination on typ. 300mA

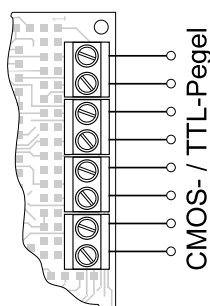
| RS-232C connector J2 |        |        |                  |
|----------------------|--------|--------|------------------|
| Pin                  | Symbol | In/Out | Function         |
| 1                    | VDD    | -      | + 5V Supply      |
| 2                    | DCD    | -      | Via LB5 to DTR   |
| 3                    | DSR    | -      | Via LB6 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     | -      | No Connect       |
| 10                   | GND    | -      | 0V Ground        |

## APPLICATION EXAMPLES

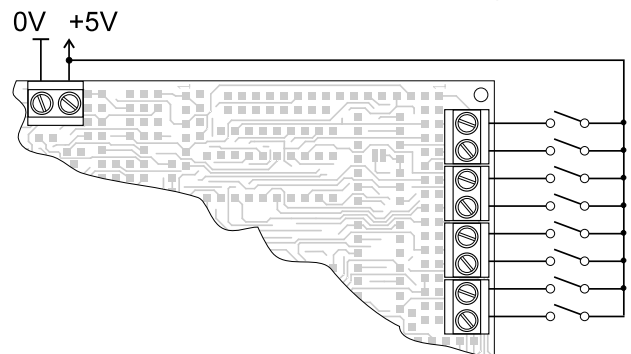
3 most frequent used connection variations, for PLC/SPS- controls, for direct connection to logic levels and connection of simple switches is shown below.



Connecting to PLC/SPS



Driving with CMOS-levels



Connecting to 8 switches

## CREATE MACROS (ALL VERSIONS)

Single or several command sequences can be combined in form of macros or scripts and stored permanently in EPROM/FLASH. These can be started by the commands *Run Macro / Run Script*. Version EA KIT128-GXT allows starting of scripts by the 8 input gates and sequence will be shown on the LCD-display. Following tools are needed for creating your own special macros:

- Disc EA DISK9719; containing the compiler, examples and fonts
- a PC with serial COM1 port, preferable 386/486 with approx. 1MB free memory on hard disk
- an EPROMer for EPROM type 27C512 (DIL, 28-pins,  $V_{PP}$  12,5V)

In case that no EPROMer is available, you can perform the test's with an EA KIT128-GXT / OPT-FLASH128 modul and send to us with your next order your final compiled EPROM-file on disc.

**ELECTRONIC ASSEMBLY****CREATING AN EPROM-FILE WITH INDIVIDUAL MACROS**

In order to get a script from a sequence of commands, all commands will be written into a file on a PC. The script- number is defined by the file extension (\*.000 up to \*.255).

When all script files are created, program COMPILE.EXE must be started. It makes an EPROM-file \*.EPR, which must be burned with an EPROMer into the EPROM or loaded with BIN2FLASH.EXE into FLASH-EPROM on display. When writing macro files it's possible to test individual scripts by connecting the display EA KIT128-GXT (or -V24 or -422) to serial port of PCs without burning repeatedly the EPROM new.

**DEFINITIONS**Macro

A macro contents a sequence of commands which are stored in an EPROM/FLASH permanent. The macro can be activated by command *Run Macro*. The macro allows automatization of repeating command sequences, i.e. clear display, draw a frame or display the company logo. Within a macro may be activated another macro, however this makro cannot start a further one. Up to 100 macros may be created (extension \*.N00 up to \*.N99). The sequence of numbers may have gaps of unused numbers. Macro \*.N00 has a special function: it's carried out automatically after power on, but can be activated also by command (power-on macro).

Script

A script is structured like a macro. Version EA KIT128-GXT allows in addition start of a script by an external digital level. According to status of input ports E0..7 the related script will be executed.

EA TOUCH8X4- equipped versions allows script activation by touch panel. Up to 256 scripts may be created (extension \*.000 up to \*.255). The sequence of numbers may have gaps of unused numbers.

Project name

A project name may have up to 8 characters (DOS-standard). If a project is named i.e. "DEMO", the related script files are named DEMO.000 to DEMO.255 and all macros must be named DEMO.N00 to DEMO.N99. Each project must have an own related initialising file which is named according to the project, i.e. DEMO.INI. Compiled EPROM-file will be named i.e. DEMO.EPR.

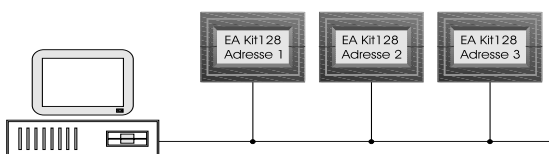
**ADDITIONAL OUTPUT PORTS**

The unused output ports for the keyboard query (J3, OUT1..OUT8) from versions EA KIT128-BUS, -SYNC and -GXT may be used as additional output ports (C-MOS level). Versions EA KIT128-422 and EA KIT128-V24 allows additional output ports only if individual outputs are relieved from keyboard query (command "ESC M 4 n1").

Command "ESC Y" activates 8 output lines (see command table on page 4). Each output port supplies a current of 0.25mA on H-level (typ.load 20k $\Omega$ ) and 20mA on L-level. Note: The solder bridges LB1..LB4 to set baud rates for versions EA KIT128-GXT, -422 and -V24 are located at output ports OUT1..OUT4. If one solder bridge is closed (at 2400, 4800, 9600 or 19200 baud) the related output cannot be used.

**ADDRESSING SEVERAL EA KIT128 ON A SINGLE INTERFACE**

Commands "select / deselect" allow to adress and operate several EA KIT128 on a single interface. The



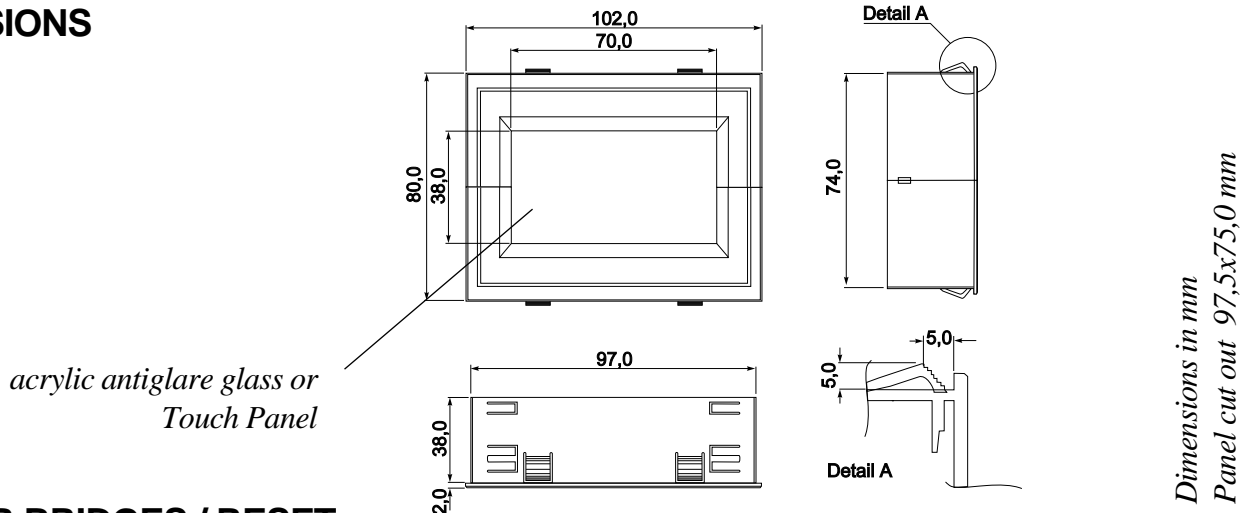
individual controller address is stored in EPROM (27C512) at address \$0046. EPROM value \$FF (factory setting) can be reprogrammed to other values any time. Another way for setting the controller address is to use the Power-On Macro with command "ESC K A n1".

NOTE: Output signal BUSY of versions EA KIT128-BUS and -SYNC must be connected with an gate and activated to prevent loss of datas in case of simultaneous writing.

NOTE: Output signals (TXD, RTS of EA KIT128-V24(TP) and EA KIT128-GXT resp. DataOut+, DataOut-, HsOut+ and HsOut- of EA KIT128-422(TP)) are not to be switched in parallel. When still needed (handshake, keyboard query), the signals must be connected to an external logic circuit.

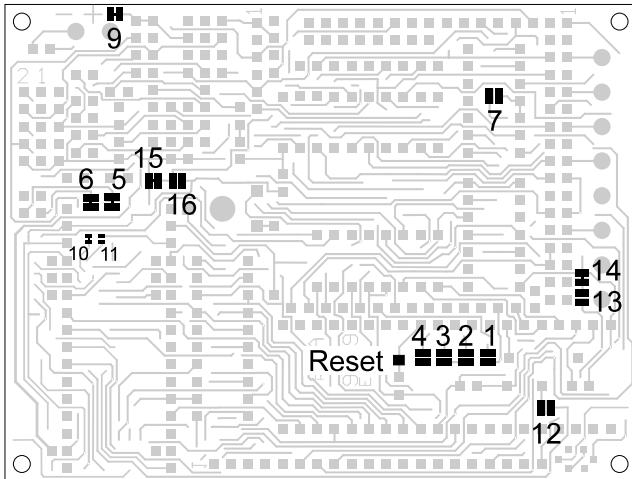
# EA KIT128

## DIMENSIONS



## SOLDER BRIDGES / RESET

Several bridging strips are placed on the backside for adapting EA KIT128 to special requirements. This solder bridges are presetted from factory and need not to be changed, only Baud rate of versions -V24(TP), -422(TP) and -GXT may be adapted to need (LB1-4 is presetted to 1200 Baud).



| Bridge | Denomin.     | Description   |     |     |     |     |  |
|--------|--------------|---|-----|-----|-----|-----|--|
|        |              | BAUD  | LB1 | LB2 | LB3 | LB4 |  |
| 1-4    | Baud rate    | Baud rate for EA KIT128-V24, -422, and -GXT   |     |     |     |     |  |
|        |              | One solder bridge only may be set!  |     |     |     |     |  |
|        |              | On all other versions the solder bridges must be always open!   |     |     |     |     |  |
|        |              | 1200  | off | off | off | off |  |
| 5      | DTR - DCD    | For versions EA KIT128-V24 and -GXT only:   |     |     |     |     |  |
|        |              | Often required connections on RS-232C interface are made here with solder bridges                                 |     |     |     |     |  |
|        |              | 2400  | on  | on  | on  | on  |  |
| 6      | DTR - DSR    | This solder bridge must be closed at EA KIT128-V24, -422 and -GXT, but must be open with EA KIT128-BUS and -SYNC! |     |     |     |     |  |
| 7      | STROBE       | This solder bridge has to be closed with 5V version and must be open with 24V version (option 18/36V) !           |     |     |     |     |  |
| 9      | DC / DC      | Possibility to adapt various EPROM's  |     |     |     |     |  |
| 10,11  | EPROM        | Opening of LB 12 of flash version EA KIT128-GXT prevents a reprogramming by mistake.                              |     |     |     |     |  |
| 12     | Prg. disable | EA KIT128-GXT only: Z-diodes relate to 0V (LB 14 closed) or to +5V (LB 13 closed, change polarity of Z-diodes !)  |     |     |     |     |  |
| 13,14  | Z-diodes     | Input resistor network relates to 0V (LB 15 closed) or to +5V (LB 16 closed)                                      |     |     |     |     |  |
| 15,16  | Pullup/down  | This pin allows a hardware reset on EA KIT128 (high-active)   |     |     |     |     |  |
| -      | Reset        |   |     |     |     |     |  |

Display checks should be made regularly with command 'ESC ? C n1' which inverts an output port n1. When the appropriate output port n1 inverts the display a proper function can be assumed. If output n1 does not react than a high level (>2µs) on terminal "Reset" will initiate a restart and the display must be written new. A software reset with commands 'ESC & F 0 0 0 0 J' will do it in some cases as well.

## HINTS FOR HANDLING AND OPERATING

- The module can be destroyed electrically by mispoled or overvoltage power supply, wrong polarity, overvoltage or static discharge on inputs or shortened outputs.
- Before disassembling the module, the power supply must be switched off. Also all inputs must not carry any current.
- Display, touch screen and cover glass are scratch-sensitive plastic materials and should not be touched with hard objects.
- Surfaces should be cleaned with soft fabric without using of chemical solvents.
- The module is designed for indoor use only. For operating in outside environment adequate precautions must be undertaken. Maximum operating temperature range from 0..+50°C should be not exceeded. In humid atmosphere or in condensing situations the module functions may drop out. Direct sun exposure to the display should be avoided. Storage temperature range is -20..+70°C.

