

Index

Index	1
LCD-SED1520 - Graphic display	2
Connecting the graphic LC-Display	4
User Function Codes LCD-SED1520.TD2	6
Control characters of the graphic LC-display	7
ESC commands LCD-SED1520 (Text)	8
LCD-SED1520 Position cursor: ESC A	10
LCD-SED1520 Define cursor: ESC c	11
LCD-SED1520 Define backlight: ESC L	12
LCD-SED1520 - New character set	13
LCD-SED1520 - Special character set	17
LCD-SED1520 graphics	19
Output on the graphic screen	20
Graphic LCD functions (overview)	21
Documentation History	23

LCD-SED1520 - Graphic display

This device driver allows output to graphic LCDs carrying the SED1520 controller. Graphic applications are supported by functions (see chapter 'Graphic').

Further information on LCD-SED1520:

- Installation of the LCD-SED1520 device driver
- Secondary addresses
- Connecting the graphic LC-Display
- User-Function-Codes LCD-SED1520
- Control characters of the graphic LC-display
- ESC commands LCD-SED1520 (Text)
- LCD-SED1520 Position cursor: ESC A
- LCD-SED1520 Define cursor: ESC c
- LCD-SED1520 Define backlight: ESC L
- LCD-SED1520: New character set
- LCD-SED1520: Special character set
- LCD-SED1520: Graphics
- Output on the graphic screen
- Graphic LCD functions (overview)

File name: LCD-SED1520.TDD, LCD-SED1520.TD2

INSTALL DEVICE #D, " LCD-SED1520.TD2" [, P1, ..., P6]

D is a constant, variable or expression of the data type BYTE, WORD, LONG in the range 0...63 and stands for the device number of the driver.

P1...P13 are further parameters which alter the standard configuration of the LC-display.

	Leave unchanged	Standard value	Description of the parameter
P1	0EEH	1	Pin No. E1 (0...7)
P2		2	Pin No. E2 (0...7)
P3		0	Pin No. A0 (0...7)
P4		3	Pin No. Reset (0...7)
P5		4	Pin No. Light (0...7)
P6		0B8H	X-Port address data bus
P7		0B0H	X-Port address control lines
P8	0EEH	6	Logical port address of data bus
P9		3	Logical port address of control lines
P10		3	Pin No. ACLK (address clock)
P11		4	Pin No. DCLK (data clock)
P12	0	8	transmission speed (bytes / 1ms)
P13	0	0	Cursor 0: ON 0FFH: OFF

Output to the graphic display device has different functions on the secondary addresses:

Secondary address	Function	Output instruction -
0	Text	PRINT, PUT
1	Graphic	PUT
2	Set new character set	PUT
3	Set new special character set	PUT

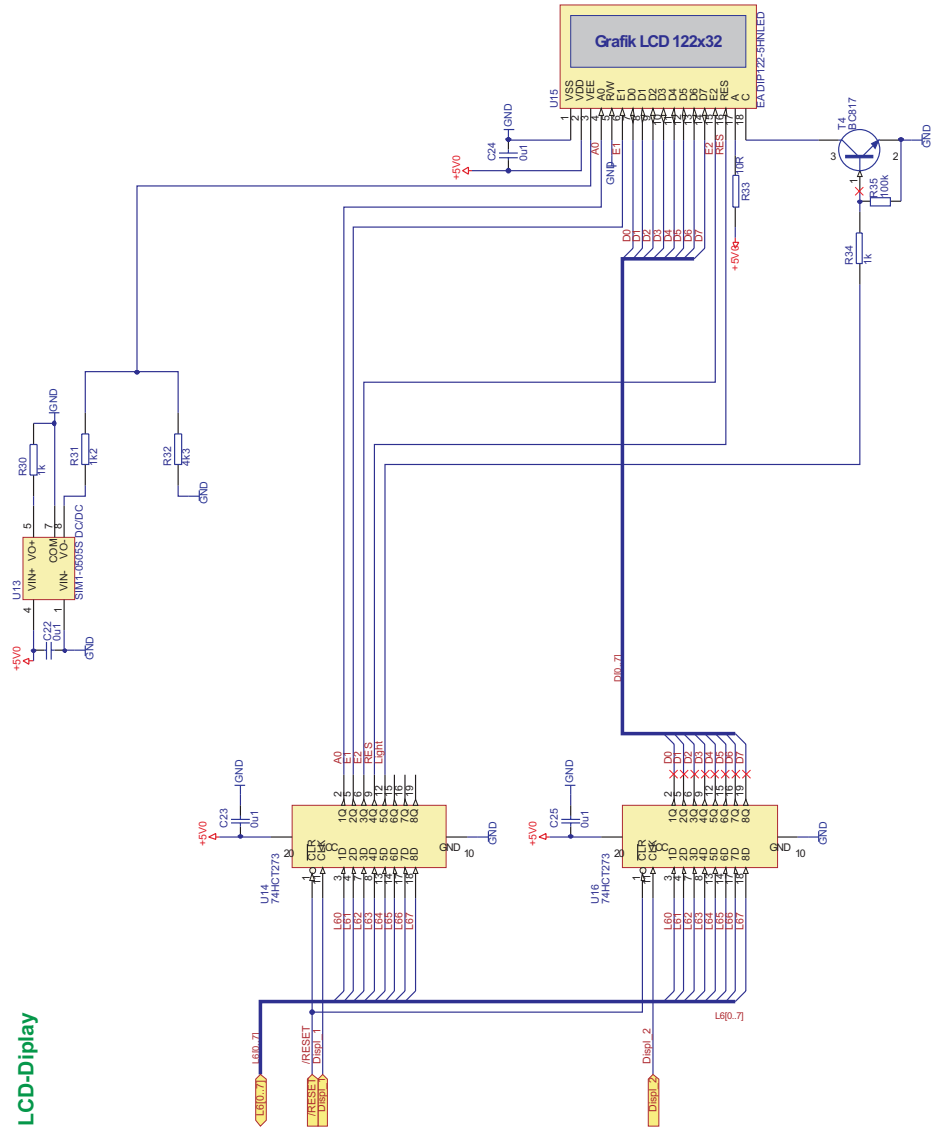
Before describing the output of data to the display some basic features of the driver LCD-SED1520 are explained, i.e. User-Function-Codes as well as ESC Sequences. Graphic on LC-display is largely supported, though by functions (see chapter 'Graphics' in the programming manual).

Connecting the graphic LC-Display

A graphic LC-display requires 2 X-Ports. The following table shows the standard pin assignment:

Function	Port	Pin
Data-bus	6	0...7
ACLK	3	3
DCLK	3	4
LCD I/O Data-bus	B8H (X-Port)	0...7
C/D	B0H (X-Port)	0
Enables left half of LCD	B0H (X-Port)	1
Enables right half of LCD	B0H (X-Port)	2
Reset	B0H (X-Port)	3
Backlight	B0H (X-Port)	4

LCD-Display



User Function Codes LCD-SED1520.TD2

User Function Codes of LCD-SED1520.TD2 to request parameters (GET):

No	Symbol Prefix UFCI_	Description
33	UFCI_OBU_FILL	Number of bytes in output buffer (Byte)
34	UFCI_OBU_FREE	Free space in output buffer (Byte)
35	UFCI_OBU_VOL	Size of output buffer (Byte)
99	UFCI_DEV_VERS	Driver version
128	UFCI_IS_ACT	Status of driver: 0: inactive >0: active (graphic data output)

User Function Codes of LCD-SED1520.TD2 to set parameters (PUT):

No	Symbol Prefix: UFCO_	Description
1	UFCO_IBU_ERASE	Delete input buffer
33	UFCO_OBU_ERASE	Delete output buffer
176	UFCO_LCD_RESET	resets LCD (hardware reset and new initialization)

Control characters of the graphic LC-display

Control characters are written directly to the LCD-device with no Esc and Eos.

CLR	<01>	Deletes the LCD text screen
HOME	<02>	Sets the cursor in the top left corner
FS	<05>	Cursor 1 position to the right
BS	<08>	Cursor 1 position to the left
LF, DO	<0Ah>	Cursor 1 position down
UP	<0Bh>	Cursor 1 position up
CR	<0Dh>	Carriage return

```
PRINT #LCD, "<1>";
```

moves the cursor to the 'home' position (X=0, Y=0). This can be directly followed by further text:

```
PRINT #LCD, "<2>Hello World"
```

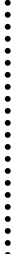
.....

.....

•

•

•



•
•
•
•
•
•

•
•
•
•
•
•

Summary :

ESC-SEQUENCE	Description
Esc, A, x, y, EoS	Absolute cursor address
Esc, c, n, EoS	Define cursor n=0: cursor off n=1: cursor on
Esc, L, n, EoS	Backlight n=0: backlight off n=1: backlight on

ESC sequences must always be output in a PRINT or PUT instruction. A number of ESC sequences can be output in an instruction if the line length allows.

LCD-SED1520 Position cursor: ESC A

```
PRINT #D, "<1Bh>A"; CHR$(x); CHR$(y); "<F0h>";
```

Absolute positioning of the cursor on the display.

D is a constant, variable or expression of the data type BYTE, WORD, LONG in the range from 0...63 and stands for the device number of the driver.

x x-co-ordinate (column) at which the cursor is to be positioned.

y y-co-ordinate (line), at which the cursor is to be positioned.

Lines and columns are counted from 0. The possible value range depends on the LC-display used. Values for **x** and **y** which are too large will be set to the remainder of the division by the maximum value.

LCD-SED1520 Define cursor: ESC c

PRINT #D, "<1Bh>c"; CHR\$(n); "<F0h>";

Defines the appearance of the cursor on the display.

D is a constant, variable or expression of the data type BYTE, WORD, LONG in the range from 0...63 and stands for the device number of the driver.

n Define cursor
n=0: cursor off
n=1: cursor on

LCD-SED1520 Define backlight: ESC L

```
PRINT #D, "<1Bh>L"; CHR$(n); "<F0h>";
```

Defines the backlight of the display.

D is a constant, variable or expression of the data type BYTE, WORD, LONG in the range from 0...63 and stands for the device number of the driver.

n Define backlight
n=0: backlight off
n=1: backlight on

LCD-SED1520 - New character set

The device driver contains a standard character set for text output on the LCD. The LCD device driver supports programming a new individual character set.

The new character set consists of 95 characters. They corresponds to the ASCII Codes 20H (32) to 7EH(126). Each TEXT character is made up of 8 bytes. The Font size of the characters is 7x5. Line no. 7 is reserved for the cursor and column 5 is the space between 2 characters. The following graphic shows, which pixel you can use for your font:

y/x	0	1	2	3	4	5	6	7
0	■	■	■	■	■	□	□	□
1	■	■	■	■	■	□	□	□
2	■	■	■	■	■	□	□	□
3	■	■	■	■	■	□	□	□
4	■	■	■	■	■	□	□	□
5	■	■	■	■	■	□	□	□
6	■	■	■	■	■	□	□	□
7	□	□	□	□	□	□	□	□



Use these pixels



Don't use these pixels

This shows the standard bitmap structure in Tiger Basic:

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 1								
Byte 2								
Byte 3								
Byte 4								
Byte 5								
Byte 6								
Byte 7								
Byte 8								

The character set of the LCD-SED1520 looks like this:

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Bit 0								
Bit 1								
Bit 2								
Bit 3								
Bit 4								
Bit 5								
Bit 6								
Bit 7								

The standard bitmap structure has to be converted to the structure of the LCD-SED1520. During run time this is possible with the function GRAPHIC_REFORMAT. The easiest way to include a new character set to the Basic program is to use the instruction DATA_FILTER. There is the possibility to include a complete character set bitmap converted to the structure of the LCD-SED1520. This is saved in the flash memory of the Tiger and can be used directly.

PUT #D, #2, data_string [, src_offset]

D is a constant, variable or expression of the data type BYTE, WORD, LONG in the range from 0...63 and stands for the device number of the driver.

#2 Secondary address = 2: new character set is to be defined.

data_string is a global or task-local variable of the type STRING and contains the source data for the new individually character set which is to be newly defined. It is also possible to specify a FLASH ADDRESS instead of a STRING. The new character set can be saved in the flash memory of the Tiger.

src_offset is a variable, constant or expression of the type BYTE, WORD or LONG and specifies the byte offset in the **pixel_string** for the first character. If nothing is specified, this value is 0.

The length of the STRING or FLASH has to be greater than or equal 760 Bytes (1 character consists of 8 Byte, 95 characters must be set, $95 \times 8 = 760$).

Example:

```
user_var_strict
#include define_a.inc
#include UFUNC4.inc

TASK MAIN
    datalabel CHSET_CONV
    install_device #LCD3, "LCD-SED1520.TD2"
    PUT #LCD3, #2, CHSET_CONV          ' new character set

    'PRINT #LCD3, #0, "    <-- 20 -->    "
    PRINT #LCD3, #0, "Hello world!"
    PRINT #LCD3, #0, "LCD-SED1520_TEXT.TIG" ;
    PRINT #LCD3, #0, "Wilke Technology"
    PRINT #LCD3, #0, "www.wilke.de"

CHSET_CONV::
DATA FILTER "LCDisplay_7x5px.bmp", "GRAPHFLT", 3

END
```


LCD-SED1520 - Special character set

Many applications require special character sets on the LC-display. The "Umlaute", which are not included in the standard character set, are used in Germany.

The LCD device driver supports programming of the special character set for the LC-display. The special character set can be defined via the secondary address 3. A set of up to 127 of your own text characters can be created. The special characters start with the code 80H. The first special character belongs to the character code 80H. Each TEXT character is made up of 8 bytes the same way like the standard character set (see above). Example:

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Bit 0								
Bit 1								
Bit 2								
Bit 3								
Bit 4								
Bit 5								
Bit 6								
Bit 7								

PUT #D, #3, data_string [, src_offset, src_len]

D is a constant, variable or expression of the data type BYTE, WORD, LONG in the range from 0...63 and stands for the device number of the driver.

#3 Secondary address = 3: new character set is to be defined.

data_string is a global or task-local variable of the type STRING and contains the source data for the new individually character set which is to be newly defined. It is also possible to specify a FLASH ADDRESS instead of a STRING. The new character set can be saved in the flash memory of the Tiger.

src_offset is a variable, constant or expression of the type BYTE, WORD or LONG and specifies the byte offset in the **pixel_string** for the first character. If nothing is specified, this value is 0.

scr_len is a variable, constant or expression of the type BYTE, WORD or LONG and specifies the number of bytes to be output after **src_offset**.

Example:

```
user_var_strict
#include define_a.inc
#include UFUNC4.inc

TASK MAIN
    datalabel CHSET_CONV
    install_device #LCD3, "LCD-SED1520.TD2"

    PUT    #LCD3, #3, CHSET_CONV, 0, 56          ' new character set
    PRINT  #LCD3, #0, "<80H><81H><82H><83H><84H><85H><86H><87H>";

CHSET_CONV::
DATA FILTER "LCDisplay_7x5px_sonderz.bmp", "GRAPHFLT", 3

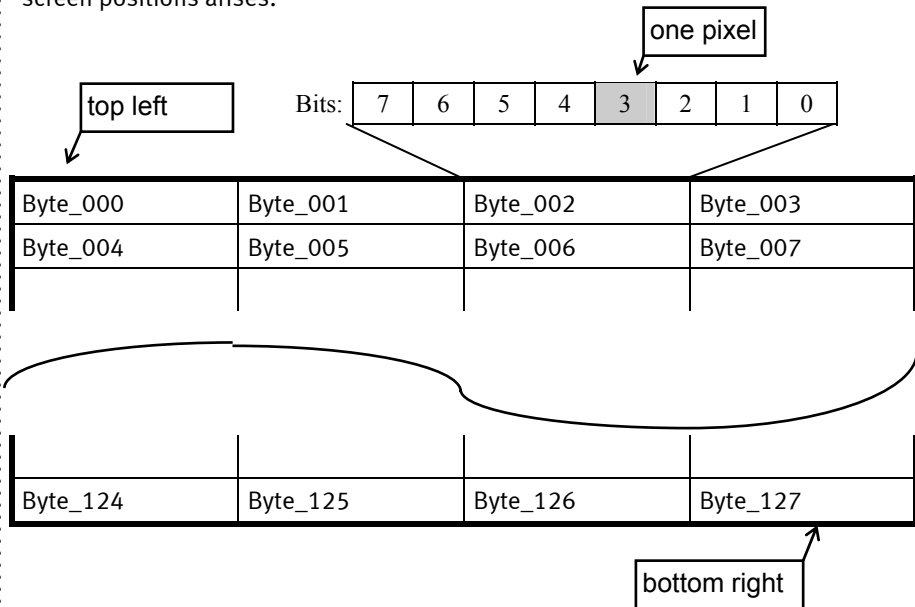
END
```

LCD-SED1520 graphics

The picture points (also called pixels or dots) are triggered bit-by-bit:

Bit = 0	light = background = unset point
Bit = 1	dark = set point

Each byte forms 8 horizontal dots on the LCD, whereby the highest-order bit is on the left and the bit-0 to the right. The graphic memory of the LCD starts in the top left corner so that with a 32x32 dot screen, the following arrangement of bytes and screen positions arises:



Output on the graphic screen

PUT #D, #1, *pixel_string*

D is a constant, variable or expression of the data type BYTE, WORD, LONG in the range from 0...63 and stands for the device number of the driver.

#1 Secondary address = 1: graphics are output

pixel_string is a global or task-local variable or constant of the type STRING and contains the source data for the graphic to be shown on the LCD. For a 122 x 32 LCD, the length of the **pixel_string** has to be 488 or greater.

The graphic is only output with the instruction PUT, never with PRINT. The pixel string must always exist! This means that variables with a limited life, e.g. local strings (in subroutines) or temporary strings (expressions) are not allowed. **Correct: global or task-local strings.**

Graphic LCD functions (overview)

The following table shows the numerous functions developed especially for graphics on LCDs. The Programmig Manual contains a chapter for the graphic functions.

Function name	Pixel-graphic	Vector graphic
OR2	•	2 graphics are overlayed with unconditioned OR
OR3	•	3 graphics are overlayed with unconditioned OR
OR4	•	4 graphics are overlayed with unconditioned OR
AND2	•	2 graphics are overlayed with unconditioned AND
AND3	•	3 graphics are overlayed with unconditioned AND
AND4	•	4 graphics are overlayed with unconditioned AND
XOR1	•	2 graphics are overlayed with unconditioned XOR
GRAPHIC_MASK_COPY	•	2 graphics are combined using a mask
GRAPHIC_MIRROR	•	mirrors agraphic: X-, Y- or X+Y axis
GRAPHIC_EXP_STRI	•	expands graphic bytes
GRAPHIC_COPY	•	copies graphic window
GRAPHIC_FILL_MASK	•	fills a square mask into a graphic
INVERT	•	inverts pixel values
DRAW_LINE		• begins a line inside graphic area
DRAW_NEXT_LINE		• continues a line inside graphic area
CLOSE_LINE		• closes a line
END_LINE		• ends a line
SET_ROTATION		• sets rotation angle 0,01-Grad
SET_SCALE		• sets scale (100% = 1000)

Function name	Pixel-graphic	Vector-graphic	
SET_BASE		•	sets Basis point X/Y
SET_GRAREA		•	sets graphic area: destin + B x H
SET_DOT		•	sets DOT to 0 or 1
FILL_AREA		•	fills graphic area
Supporting functions:			
DISTANCE			calculates distance between 2 co-ordinates
QUICK_WORD_SIN			calculates fast sine in WORD
QUICK_WORD_COS			calculates fast cosine in WORD

Documentation History

Version of Documentation	Version of driver	Description / Changes
003	1.00b	<ul style="list-style-type: none">- Look of document changed- Example programs for new charsets