LCD-S1D13700.TDD

This driver functions in many ways analog to the LCD-6963.TDD, only that instead of the Toshiba T6963C, it supports the Epson S1D13700-controller. There are some differences, however:

Installing the driver

INSTALL DEVICE #*D*, "LCD-S1D13700.TDD" [, *P1, ..., P7*]

D	is a variable, a constant, or an expression of the data type BYTE, WORD, LONG in the range between 0...63 and stands for the device number of the driver.
P1...P7	are further parameters that change the standard pin assignment of the LC display.

All parameters P1...P6 are bytes and can remain unchanged as values when inserting 0 or 0EEH (=238). When you want to change the bit assignment of the control lines, all parameters from P3a to P3d have to be given, while one single '0EEH' as a value for the 3rd parameter means that all control lines remain unchanged.

	Keep default	Description of the parameter
P1	0	Logical data bus address (6 or 8) (Default: 6)
P2	0	Logical port-address of the control lines (4, 6, 7, 8 or 9) (Default: 8)
P3a P3b P3c P3d	0EEH	Bit positions (0...7) for the control lines of the LC display (when unchanged, only one parameter OEEH, otherwise all 4 parameters have to be given): -WR (Default: 1) -RD (Default: 0) -CE (Default: 4) C/D (Default: 3)
P4	0EEH	LCD type (see table) (Default: 1)
P5	-	Transmission rate in kbyte/sec. (4...254), concerns desired CPU load (only for graphics-output). (Default: 15)
P6	-	Delay when writing (1...254). The higher the value, the larger the delay. (Default: 011H)
P7	-	Transmission method (Default: 0)

Type list

LC display types with Epson controller: S1D13700

No.	LCD type	Pixel columns x lines	Font	Text lines x columns	RAM
1	LCD_EPS_BW1	320 x 240	8 x 8	30 x 40	32k

Transmission methods

Method 0

This method can e.g. be used on the Tiny Tiger 2 prototyping board. For writing a byte to the LCD the WR line is simply changed in its state (XOR). This saves time when sending the data to the LCD. For this a special hardware is of course required, that generates a pulse (LOW-HIGH) from a changing edge. If this hardware is not available, please use method 2.

Method 1

Here, when sending the data the WR line is first drawn to LOW and then again set to HIGH. There is no need for a special hardware; however, the process is more time-consuming.

Secondary addresses

The output to the LCD device is distributed on several secondary addresses according to the function:

Secondary address	Function	Output instruction
0	Text output	PRINT, PUT
1	Graphical output	PUT
2	Graphical output layer 2	PUT
7	Send command to controller	PUT

Example for text output (like with LCD-6963):

```
PRINT #LCD3, "Hello world"
PUT #LCD3, #0, Value1
```

Example for graphical output (like with LCD-6963):

```
PUT #LCD3, #1, Screen$, 0, 0, 9600
```

Example for command output

```
command$ = "02 46 B0 04 05 42"% + "pixel"
PUT #LCD3, #7, command$
```

The structure of a command to the controller is as follows:

Byte 1:	Number of parameters that are to be sent
Byte 2:	Controller command
Byte 3 ... Byte n:	Number of parameters according to byte 1

Several commands can be sent with one single instruction this way. The commands and their parameters can be found in the manual for the Epson controller.

User function codes

User-Function-Codes of the LCD-S1D13700.TDD for requesting parameters (instruction GET, secondary address 0, if not marked in any particular way):

No.	Symbol prefix UFCI_	Description
33	UFCI_OBU_FILL	Filling level of the output buffer (byte)
34	UFCI_OBU_FREE	Free space in the output buffer (byte)
35	UFCI_OBU_VOL	Size of the output buffer (byte)
94	UFCI_TEXT_HOME	Home address for text (LCD RAM)
95	UFCI_GRA_HOME	Home address for graphics (LCD RAM)
99	UFCI_DEV_VERS	Version of the driver
128	UFCI_IS_ACT	Status of the driver: 0: driver active (output from DACC string) ›0: driver inactive at present
129	UFCI_STAT_TEXT	Status of the text display: 0: text display OFF ›0: text display ON
130	UFCI_STAT_GRAPHIC	Status of the graphic display: 0: graphic display OFF ›0: graphic display ON

User-Function-Codes of the LCD-S1D13700.TDD for setting of parameters (PUT):

Nr.	Symbol prefix: UFCO_	Description
136	UFCO_SET_INV	Set output mode: 0 = normal output 1 = inverted output
144	UFCO_LCD_TYPE	Set new LCD type

ESC commands (text output)

The ESC command “A” for the cursor positioning for the text output is available in the driver LCD-S1D13700.TDD. The way of functioning corresponds exactly to the driver LCD-6963.TDD.

ESC commands “G” and “T” for switching the graphics and the text mode are also available.

It is advised to turn off the text mode when it is not required. Otherwise a flickering can occur during the graphics output, due to the LCD. The text mode is turned off as follows:

```
PRINT #D, "<1Bh>T"; CHR$(0); "<0F0h>";
```

Positioning the cursor: ESC A

PRINT #D, "<1Bh>A"; CHR$(x); CHR$(y); "<0F0h>";

Positions the cursor absolutely on the display

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

x x-coordinate (column), onto which the cursor is to be positioned.

y y-coordinate (line), onto which the cursor is to be positioned.

The counting of the columns and lines starts at 0. The possible range depends on the kind of LC display that is used. Too large values for **x** and **y** are set to the maximum value.

Turning the graphics display on/off: ESC G

PRINT #D, "<1BH>G"; CHR$(n); "<0F0H>";

Turns the graphical layer of the display on or off

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

n 0 = off, 1 = on.

With this ESC sequence, the graphics on the display is turned on or off. The text content is not influenced by this.

You can read out the status of the text display with the User Function Code *UFCI_STAT_GRAPHIC*.

Turning the text display on/off: ESC T

PRINT #D, "<1BH>T"; CHR$(n); "<0F0H>";

Turns the text layer of the displays on or off

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

n 0 = off, 1 = on.

With this ESC-sequence, the text on the display is turned on or off. The graphics content is not influenced by this.

You can read out the status of the text display with the User Function Code *UFCI_STAT_TEXT*.

Reading from the controller

It is possible to read out the internal RAM of the LCD controller. This is done simply with *GET.* The secondary address of the *GET* instruction corresponds to the starting address in the LCD-RAM. This makes it now very easily possible to read out the current LCD content, which might be needed when several graphics layers are used.

Example for reading out the current LCD content:

```
GET #1, #0, #UFCI_GRA_HOME, 0, adrGraphic       ' Start address of graphics
GET #1, #adrGraphic, 0, lcd$                    ' Read out graphical content
```

Output onto the graphics screen

PUT #D, #L, *pixel_string* [, *lcd_offset*, *src_offset*, *src_len*]

D	is a constant, variable or an expression of data type WORD, LONG, BYTE in the range 0...63 and stands for the device number of the driver.
#L	Secondary address = 1: graphics is output (Layer 1) Secondary address = 2: graphics is output (Layer 2)
pixel_string	is a global or task-local variable or a constant of type STRING and contains the source data of the graphics, that is to be shown on the LCD.
lcd_offset	is a constant, variable or an expression of data type WORD, LONG, BYTE and determines the byte offset in the LCD graphics-RAM. The top left corner of the screen has the offset 0.
src_offset	is a constant, variable or an expression of data type WORD, LONG, BYTE and determines the byte offset in the **pixel_string**. It is possible to output only a part of the graphics data with this.
scr_len	is a constant, variable or an expression of data type WORD, LONG, BYTE and determines the number of bytes, that are to be output beginning from **src_offset**.

Graphics layer 2

Through the secondary address 1 one can output graphics on the LCD as usual. But now it is also possible to output a second layer in addition to the first graphics layer. This functions exactly like secondary address 1. A string is output to secondary address 2 which is logical add (OR) to the first layer. In this way, 2 pictures can be mixed very easily. It would be convenient to use one layer as background that does not change and one layer for the changing elements.

Example fort he output of 2 layers:

```
PUT #1, #1, Layer1$              ' output layer 1
PUT #1, #2, Layer2$              ' output layer 2 (layers are logical add)
```

Documentation History

Version of Documentation	Version of SER5	Description / Changes
004	1.00i	- Install_device default values - Sending methods number
005	1.00k	- Status text and graphic mode
006	1.00k	- Telephone number changed