EPSON®

S1D13705 Embedded Memory LCD Controller

13705PLAY Diagnostic Utility

Document No. X27A-B-005-04

Copyright © 2001 Epson Research and Development, Inc. All Rights Reserved.

Information in this document is subject to change without notice.You may download and use this document, but only for your own use in evaluating Seiko Epson/EPSON products. You may not modify the document. Epson Research and Development, Inc. disclaims any representation that the contents of this document are accurate or current. The Programs/Technologies described in this document may contain material protected under U.S. and/or International Patent laws.

EPSON is a registered trademark of Seiko Epson Corporation. All other Trademarks are the property of their respective owners

THIS PAGE LEFT BLANK

13705PLAY

13705PLAY is a utility which allows the user to easily read/write the S1D13705 registers, Look-Up Table and display memory.

The user interface for 13705PLAY is similar to the DOS DEBUG program; commands are received from the standard input device, and output is sent to the standard output device (console for Intel and terminal for embedded platforms). This utility requires the target platform to support standard IO.

13705PLAY commands can be entered interactively using a keyboard/monitor or they can be executed from a script file. Scripting is a powerful feature which allows command sequences played back from a file thus avoiding having to retype lengthy sequences.

The 13705PLAY display utility must be configured and/or compiled to work with your hardware platform. The program 13705CFG.EXE can be used to configure 13705PLAY. Consult the 13705CFG users guide, document number X27A-B-001-xx, for more information on configuring S1D13705 utilities.

This software is designed to work in both embedded and personal computer (PC) environments. For the embedded environment, it is assumed that the system has a means of downloading software from the PC to the target platform. Typically, this is done by serial communications. The PC uses a terminal program to send control commands and information to the target processor. Alternatively, the PC can program an EPROM, which is then placed in the target platform. Some target platforms can also communicate with the PC via a parallel port connection, or an Ethernet connection.

S1D13705 Supported Evaluation Platforms

13705PLAY has been tested with the following S1D13705 supported evaluation platforms:

- PC system with an x86 processor. Both 16-bit and 32-bit code is supported.
- M68EC000IDP (Integrated Development Platform) board, revision 3.0, with a Motorola M68EC000 processor.
- SH3-LCEVB board, revision B, with an Hitachi SH-3 HD6417780 processor.

If the platform you are using is different from the above, please see the S1D13705 Programming Notes and Examples manual, document number X26A-G-002-xx.

Installation

PC Intel Platform

For 16-Bit Program Version: copy the file 13705PLAY.EXE to a directory that is in the DOS path on your hard drive.

For 32-Bit Program Version: install the 32-bit Windows device driver S1D13XXX.VXD as described in the S1D13XXX 32-Bit Windows Device Driver Installation Guide, document number X00A-E-003-xx. Copy the file 13705PLAY.EXE to a directory that is in the DOS path on your hard drive.

Embedded Platform

Download the program 13705PLAY to the system.

Usage

PC platform: at the prompt, type **`13705play [/?]`**.

Embedded platform: execute **`13705play`** and at the prompt, type the command line argument.

Where: `/?` displays program revision information.

The following commands are valid within the 13705PLAY program.

`X index [data]`	Reads/writes the registers. Writes data to the register specified by the index when “data” is specified; otherwise the register is read.
`XA`	Reads all registers.
`L index [data1 data2 data3]`	Reads/writes Look-Up Table (LUT) values. Writes data to the LUT index when “data” is specified; otherwise the LUT index is read. Data must consist of 3 bytes: 1 red, 1 green, 1 blue. and range in value from 0x00 to 0x0F.
`LA`	Reads all LUT values.
`F[W] addr1 addr2 data . . .`	Fills bytes or words from address 1 to address 2 with data. Data can be multiple values (e.g. `F 0 20 1 2 3 4` fills address 0 to 0x20 with a repeating pattern of 1 2 3 4).

`R[W] addr [count]`	Reads “count” of bytes or words from the address specified by “addr”. If “count” is not specified, then 16 bytes/words are read.
`W[W] addr data . . .`	Writes bytes or words of data to address specified by “addr”. Data can be multiple values e.g. `W 0 1 2 3 4` writes the byte values 1 2 3 4 starting at address 0).
`I`	Initializes the chip with user specified configuration.
`M [bpp]`	Returns information about the current mode. If “bpp” is specified then set the requested color depth.
`P 0|1|2`	Sets software power save mode 0-2. Power save mode 0 is normal operation.
`H [lines]`	Halts after specified lines of display. This feature halts the display during long read operations to prevent data from scrolling off the display. Set 0 to disable.
`Q`	Quits this utility.
`?`	Displays Help information.

13705PLAY Example

1. Type “13705PLAY” to start the program.
2. Type "?" for help.
3. Type "i" to initialize the registers.
4. Type "xa" to display the contents of the registers.
5. Type "x 5" to read register 5.
6. Type "x 3 10" to write 10 hex to register 3.
7. Type "f 0 400 aa" to fill the first 400 hex bytes of display memory with AA hex.
8. Type "f 0 14000 aa" to fill 80k bytes of display memory with AA hex.
9. Type "r 0 ff" to read the first 100 hex bytes of display memory.
10. Type "q" to exit the program.

Scripting

13705PLAY can be driven by a script file. This is useful when:

- there is no standard display output to monitor command entry and results.
- various registers must be quickly changed faster than can achieved by typing.
- The same series of keystrokes is being entered time and again.

A script file is an ASCII text file with one 13705PLAY command per line. All scripts must end with a “q” (quit) command in order to return control to the operating system. The semi-colon is used as a comment delimitor. Everything on a line after the semi-colon will be ignored.

On a PC platform, a typical script command line is: “13705PLAY < dumpregs.scr > results”.

This causes the script file “dumpregs.scr” to be interpreted and the results to be sent to the file “results.”

Example 1: The script file “dumpregs.scr” can be created with and text editor and will look like the following:

```
; This file initializes the S1D13705 and reads the registers
i       ; Initialize the registers.
xa      ; Dump all the registers
la      ; And the LUT
q       ; Exit
```

Comments

- All numeric values are considered to be hexadecimal unless identified otherwise. For example, 10 = 10h = 16 decimal; 10t = 10 decimal; 010b = 2 decimal.
- Redirecting commands from a script file (PC platform) allows those commands to be executed as though they were typed.

Program Messages

>>> WARNING: DID NOT DETECT S1D13705 <<<

The HAL was unable to read the revision code register on the S1D13705. Ensure that the S1D13705 hardware is installed and that the hardware platform has been configured correctly. Also check that the display memory address has been configured correctly.

ERROR: Unable to locate/load S1D13XXX.VXD

13705PLAY was unable to load a required driver. The file S1D13XXX.VXD should be located in x:\WINDOWS\SYSTEM or in x:\WINNT\SYSTEM. If the file is not there, install it as described in the S1D13XXX 32-Bit Windows Device Driver Installation Guide, document number X00A-E-003-xx.

ERROR: An IOCTL error occurred

This message indicates an error at the IO control layer occurred. The usual cause for this is an incorrect hardware configuration.

ERROR: The HAL returned an unknown error

This message should never be displayed, it indicates that 13705SHOW is unable to determine the cause of an error returned from the HAL.

THIS PAGE LEFT BLANK