EPSON®
EXCEED YOUR VISION

S1D13706 Embedded Memory LCD Controller

13706PLAY Diagnostic Utility

Document Number: X31B-B-003-02

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. Microsoft and Windows are registered trademarks of Microsoft Corporation. All other trademarks are the property of their respective owners.

THIS PAGE LEFT BLANK

13706PLAY

13706PLAY is a diagnostic utility allowing a user to read/write to all the S1D13706 registers, Look-Up Tables and display buffer. 13706PLAY 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, terminal for embedded platforms). This utility requires the target platform to support standard IO (stdio).

13706PLAY commands can be entered interactively by a user, or be executed from a script file. Scripting is a powerful feature which allows command sequences to be used repeatedly without re-entry.

The 13706PLAY diagnostic utility must be configured and/or compiled to work with your hardware platform. The program 13706CFG.EXE can be used to configure 13706PLAY. For further information on 13706CFG, refer to the *13706CFG Users Manual*, document number X31B-B-001-xx.

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, where 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.

S1D13706 Supported Evaluation Platforms

13706PLAY supports the following S1D13706 evaluation platforms:

- PC with an Intel 80x86 processor running Windows® 9x/NT.
- M68EC000IDP (Integrated Development Platform) board, revision 3.0, with a Motorola M68EC000 processor.
- MC68030IDP (Integrated Development Platform) board, revision 3.0, with a Motorola MC68030 processor.
- SH3-LCEVB board, revision B, with an Hitachi SH-3 HD6417780 processor.
- DSP56654ADS (Applications Development System) board, with a Motorola REDCAP2 processor.

Note

The 13706PLAY source code can be modified or recompiled to allow 13706PLAY to run on other evaluation platforms not listed above.

Installation

PC platform

Copy the file **13706play.exe** to a directory in the path (e.g. PATH=C:\S1D13706).

Embedded platform

Download the program **13706play** to the system.

Usage

PC platform

At the prompt, type:

```
13706play [/?]
```

Where:

/? displays copyright and program version information.

Embedded platform

Execute **`13706play`** and at the prompt, type the command line argument **/?**.

Where:

/? displays copyright and program version information.

Commands

The following commands are designed to be used from within the 13706PLAY program. However, simple commands can also be executed from the command line. If a command with multiple arguments is executed from the command line, it must be enclosed in double quotes (e.g. **13706play “f 0 14000 AB” q**).

Note

If the endian mode of the host platform is big endian, reading/writing words and dwords to/from the registers and display buffer may be incorrect. It may be necessary for the user to manually swap the bytes in order to perform the IO correctly. For further information on little/big endian and the S1D13706 byte/word swapping capabilities, see the *S1D13706 Hardware Functional Specification*, document number X31B-A-001-xx.

CLKI [?] iFreq

Selects a preset clock frequency (MHz) for CLKI. If the “?” option is used, the list of available frequencies for CLKI is displayed.

Where:

?	Displays a list of available frequencies for CLKI (MHz).
iFreq	Sets CLKI to an index representing a preset frequency (MHz) specified by iFreq. iFreq is based on the table provided with the command **CLKI ?**.

Note

The CLKI command programs preset frequencies available on the S5U13706B00C evaluation board. This function is not designed for use on other evaluation platforms or prototype designs.

CLKI2 [?] iFreq

Selects a preset clock frequency (MHz) for CLKI2. If the “?” option is used, the list of available frequencies for CLKI2 is displayed.

Where:

?	Displays a list of available frequencies for CLKI2 (MHz).
iFreq	Sets CLKI2 to an index representing a preset frequency (MHz) specified by iFreq. iFreq is based on the table provided with the command **CLKI2 ?**.

Note

The CLKI2 command programs preset frequencies available on the S5U13706B00C evaluation board. This function is not designed for use on other evaluation platforms or prototype designs.

CW word

Sends a 24-bit hexadecimal value to the programmable clock. Note that the programmable clock documentation uses the term “word” to describe the 24-bit value. The use of “word” does not imply a 16-bit value in this case.

F addr1 addr2 data...

Fills a specified address range with 8-bit data (bytes).

Where:

addr1	Start address of the range to be filled (hex).
addr2	End address of the range to be filled (hex).
data	Data to be written (hex). Data can be a list of bytes to be repeated for the duration of the fill. To use decimal values, attach a “t” suffix to the value. (e.g. 100t is 100 decimal)

FD addr1 addr2 data...

Fills a specified address range with 32-bit data (dwords).

Where:

addr1	Start address of the range to be filled (hex).
addr2	End address of the range to be filled (hex).
data	Data to be written (hex). Data can be a list of dwords to be repeated for the duration of the fill. To use decimal values, attach a “t” suffix to the value. (e.g. 100t is 100 decimal)

FW addr1 addr2 data...

Fills a specified address range with 16-bit data (words).

Where:

addr1	Start address of the range to be filled (hex).
addr2	End address of the range to be filled (hex).
data	Data to be written (hex). Data can be a list of words to be repeated for the duration of the fill. To use decimal values, attach a “t” suffix to the value. (e.g. 100t is 100 decimal).

H [lines]

Sets the number of lines of data that are displayed at a time. The display is halted after the specified number of lines. Setting the number of lines to 0 disables the halt function and allows the data to continue displaying until all data has been shown.

Where:

lines	Number of lines that are shown before halting the displayed data (decimal value).

I

Initializes the S1D13706 registers with the default register settings as configured by the utility 13706CFG. To initialize the S1D13706 with different register values, reconfigure 13706PLAY using 13706CFG. For further information on 13706CFG, see the *13706PLAY User Manual*, document number X31B-B-001-xx.

L index [red green blue]
Writes red, green, and blue Look-Up Table (LUT) components for a given display type. If the red, green, and blue components are not specified, reads the components at the given index.
Where:

index	Index into the LUT (hex).
red	Red component of the LUT (hex).
green	Green component of the LUT (hex).
blue	Blue component of the LUT (hex).

Note
Only bits 7-2 of each color are used in the LUT. For example, 04h is the first color intensity after 00h. Valid LUT colors follow the pattern 00h, 04h, ..., FCh.

LA
Reads all LUT values.

Note
Only bits 7-2 of each color are used in the LUT. For example, 04h is the first color intensity after 00h. Valid LUT colors follow the pattern 00h, 04h, ..., FCh.

M [bpp]
Sets the color depth (bpp). If no color depth is provided, information about the current settings are listed.
Where:

bpp	Color depth to be set (1/2/4/8/16 bpp).

Q
Quits the program.

P [on | off]
Controls the power on/off state of the S1D13706.
Where:

on	Powers on the chip.
off	Powers off the chip.

R addr [count]
Reads a certain number of bytes from the specified address. If no value is provided for count, it defaults to 10h.
Where:

addr	Address from which byte(s) are read (hex).
count	Number of bytes to be read (hex).

RD addr [count]
Reads a certain number of dwords from the specified address. If no value is provided for count, it defaults to 10h.
Where:

addr	Address from which dword(s) are read (hex).
count	Number of dwords to be read (hex).

RW addr [count]
Reads a certain number of words from the specified address. If no value is provided for count, it defaults to 10h.
Where:

addr	Address from which word(s) are read (hex).
count	Number of words to be read (hex).

T xx
Tests VNDP read for xx seconds. **(This option was developed for testing purposes only and is not supported.)**
Where:

xx	The number of seconds VNDP is tested (decimal).

V
Calculates the current frame rate from the VNDP count.

W addr data ...
Writes byte(s) of data to specified memory address.
Where:

addr	Address data is written to.
data	Data to be written (hex). Data can be a list of bytes to be repeated for the duration of the write. To use decimal values, attach a “t” suffix to the value (e.g. 100t is 100 decimal). To use binary values attach a “‘b” suffix to the value (e.g. 0111‘b).

WD addr data ...
Writes dword(s) of data to specified memory address.
Where:

addr	Address data is written to
data	Data to be written (hex). Data can be a list of dwords to be repeated for the duration of the write. To use decimal values, attach a “t” suffix to the value (e.g. 100t is 100 decimal). To use binary values attach a “‘b” suffix to the value (e.g. 0111‘b).

WW addr data ...
Writes word(s) of data to specified memory address.
Where:

addr	Address data is written to
data	Data to be written (hex). Data can be a list of words to be repeated for the duration of the write. To use decimal values, attach a “t” suffix to the value (e.g. 100t is 100 decimal). To use binary values attach a “‘b” suffix to the value (e.g. 0111‘b).

X index [data]
Writes byte data to the register at index. If no data is specified, reads the 8-bit (byte) data from the register at index.
Where:

index	Index into the registers (hex).
data	Data to be written to/read from register (hex). Data can be a list of bytes to be repeated for the duration of the write. To use decimal values, attach a “t” suffix to the value (e.g. 100t is 100 decimal). To use binary values attach a “‘b” suffix to the value (e.g. 0111‘b).

XA
Reads all the S1D13706 registers.

XD index [data]
Writes dword data to the register at index. If no data is specified, reads the 32-bit (dword) data from the register at index.
Where:

index	Index into the registers (hex).
data	Data to be written to/read from register (hex). Data can be a list of dwords to be repeated for the duration of the write. To use decimal values, attach a “t” suffix to the value (e.g. 100t is 100 decimal). To use binary values attach a “‘b” suffix to the value (e.g. 0111‘b).

XW index [data]
Writes word data to the register at index. If no data is specified, reads the 16-bit (word) data from the register at index.
Where:

index	Index into the registers (hex).
data	Data to be written to/read from register (hex). Data can be a list of words to be repeated for the duration of the write. To use decimal values, attach a “t” suffix to the value (e.g. 100t is 100 decimal). To use binary values attach a “‘b” suffix to the value (e.g. 0111‘b).

?
Displays the help screen.

13706PLAY Example

1. Configure **13706PLAY** using the utility **13706CFG**. For further information on 13706CFG, see the *13706CFG User Manual*, document number X31B-B-001-xx.
2. Type **13706PLAY** to start the program.
3. Type **?** for help.
4. Type **i** to initialize the registers.
5. Type **xa** to display the contents of the registers.
6. Type **x 34** to read register 34h.
7. Type **x 34 10** to write 10h to register 34h.
8. Type **f 0 ffff aa** to fill the first FFFFh bytes of the display buffer with AAh.
9. Type **r 0 100** to read the first 100h bytes of the display buffer.
10. Type **q** to exit the program.

Scripting

13706PLAY can be driven by a script file. This is useful when:

- there is no display output and a current register status is required.
- various registers must be quickly changed to view results.

A script file is an ASCII text file with one 13706PLAY command per line. All scripts must end with a “q” (quit) command.

On a PC platform, a typical script command line might be:

13706PLAY < dumpregs.scr > results

This causes the file **dumpregs.scr** to be interpreted as commands by 13706PLAY and the results to be sent to the file **results.**

Example 1: Create a script file that reads all registers and then exits.

```
; This file initializes the S1D13706 and reads the registers.
; Note: after a semicolon (;), all characters on a line are ignored.
; Note: all script files must end with the “q” command.

; Initialize the S1D13706
i

; Read all registers
xa

; Exit the program
q
```

Comments

- All displayed numeric values are considered to be hexadecimal unless identified otherwise. For example:
  - 10 = 10h = 16 decimal.
  - 10t = 10 decimal.
  - 010‘b = 2 decimal.
- Redirecting commands from a script file (PC platform) allows those commands to be executed as if entered by a user.

THIS PAGE LEFT BLANK