EPSON
EXCEED YOUR VISION

S1C17801
FLASHC Application Note

When using the commands, follow the instructions of NOTICE_Application Notes Sample Programs.pdf being included in the downloaded compressed file.

SEIKO EPSON CORPORATION

Rev.1.0

NOTICE

No part of this material may be reproduced or duplicated in any form or by any means without the written permission of Seiko Epson. Seiko Epson reserves the right to make changes to this material without notice. Seiko Epson does not assume any liability of any kind arising out of any inaccuracies contained in this material or due to its application or use in any product or circuit and, further, there is no representation that this material is applicable to products requiring high level reliability, such as, medical products. Moreover, no license to any intellectual property rights is granted by implication or otherwise, and there is no representation or warranty that anything made in accordance with this material will be free from any patent or copyright infringement of a third party. This material or portions thereof may contain technology or the subject relating to strategic products under the control of the Foreign Exchange and Foreign Trade Law of Japan and may require an export license from the Ministry of Economy, Trade and Industry or other approval from another government agency.

All other product names mentioned herein are trademarks and/or registered trademarks of their respective companies.

©SEIKO EPSON CORPORATION 2008, All rights reserved.

OVERVIEW

This document is a reference to use the FLASHC function of S1C17801.

OPERATING ENVIRONMENT

- S5U1C17801T1100 (hereafter SVT17801:Software eValuation Tool for S1C17801)
  SVT17801 CPU board and SVT17 ICD board
- USB miniB cable
- PC
  Installed with GNU17 development tools
  Installed with the USB driver for the SVT17 ICD board

Table of Contents

1. SPECIFICATIONS

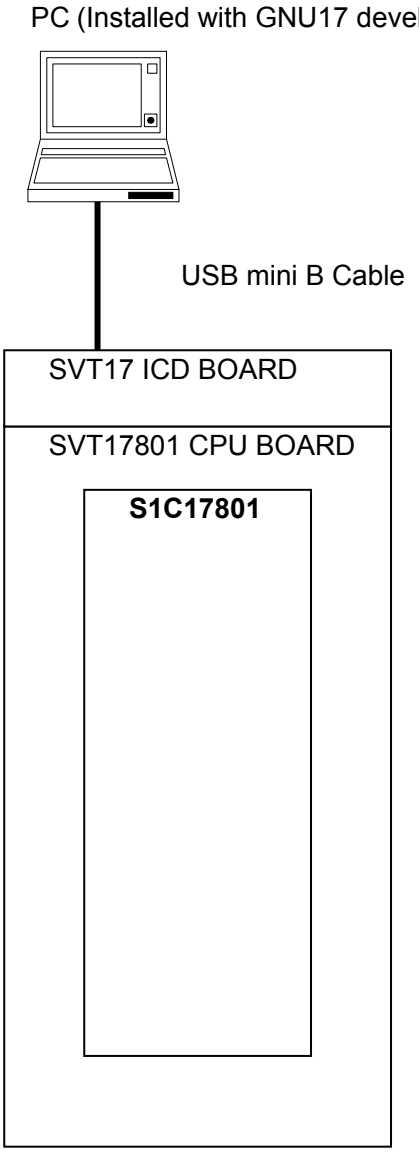
This sample software erases and writes data to the internal Flash built in S1C17801.

It inputs the following commands from “command >”prompt of [Simulated I/O] window.

Command list

Command name	Description
chip	Chip erase (erases all)
sec	Sector erase (Erases sectors in the specified address range)
write	Data write (Writes a sector to the specified address)
all	All write (Writes to the whole area)

The following describes the connection procedure.



2. DESCRIPTIONS OF FUNCTIONS USED

Input and output pins	The connection pins (A0 to A21, D0 to D15, CE, OE, WE, and RY/BY） to the external Flash is connected with the addresses A1 to A22, data D0 to D15, CE0, RD, WRL and P44. Addresses A1 to A22 are also used by general input and output port pins (P61 to P67, P70 to P77, and P80 to P86). Data D0 to D15 are also used by general input and output port pins (P90 to P97, and PB0 to PB7). CE0 is also used by general input and output port pins (PA0). RD is also used by general input and output port pins (PA4). WRL is also used by general input and output port pin (PA5). * The setting is defined within boot.c.
SRAMC	Defines the access Wait/DeviceSize/BSL method setting of CE0. * The setting is defined within boot.c.
Interrupt handler	Unused.

3. SOFTWARE DESCRIPTION

3.1 File Configuration

Filename	Function
boot.c	Startup module
main.c	Main function
inthdlr.c	Interrupt handler function
vector.c	Vector table settings
header¥reg_801.h	Register definitions
header¥vector.h	Vector table definitions
flashc_drv¥ flashc_drv.c	FLASHC driver API group
flashc_drv¥ flashc_api.h	FLASHC driver API definitions
cmu_drv/	CMU drivers group folder
gpio_drv/	GPIO drivers group folder
itc_drv/	ITC drivers group folder
sramc_drv/	SRAMC drivers group folder
flashc_gnu17IDE.lds	Linker script file
flashc_gnu17IDE.cmd	GDB command file
flashc_gnu17IDE.par	Parameter configuration file
flashc_gnu17IDE.mak	Make file
.cdtproject	Project file
.gnu17project	Project file
.project	Project file
GDB17 Launch for flashc.launch	Project startup file

3.2 Descriptions of the Modules

File name: main.c

Function name	Function
main	Displays messages in the [Simulated I/O] window, and controls the Flash by inputting commands.
PortInitialize	Sets general-purpose I/O ports
ItcInitialize	Configures ITC
FlashcInitialize	Configures FLASHC

For the GPIO driver functions, refer to the “Detailed description of GPIO driver” chapter.
For ITC driver functions, refer to the “Detailed description of ITC driver” chapter.

3.3 Global Variables

The following shows the global variables used in the sample program.

Variable name	Type	Function
inbuff[128]	char	For Simulated I/O input
outbuff[128]	char	For Simulated I/O output

3.4 Structure

The following describes the structure used in the sample program.

Definition name		
T_FLASHC_CFG	config	
Member		
flashWait	unsigned short	Access wait
TTBR	unsigned long	Trap Table base address
Remarks		
Structure for setting the initial value of FLASHC.		

3.5 Operating Procedures

Import the project

(1) Launch the IDE and import the "flashc" project.
    * For the import procedure, refer to S5U1C17001C Manual "3. Software Development Procedure."
    * Copy required drivers from the driver folder.

Build

(1) Build the "flashc" project using the IDE.

Connection and powering on procedures

(1) Connect SVT17801, USB miniB cable and PC.
(2) Reset SVT17 ICD board.

Run the program

(1) Build "flashc" project using the IDE.
(2) After executing "flashc", a "command >" prompt is displayed in the [Simulated I/O] window.
(3) Input one of the following commands.

Command list

Command name	Description
chip	Chip erase (erases all)
sec	Sector erase (Erases sectors in the specified address range)
write	Data write (Writes a sector to the specified address)
all	All write (Writes to the whole area)

Inputting the "chip" command starts to erase the whole area in the FlashROM. A message "Chip erase complete" indicates that the erasing process has been completed successfully.

Specify sectors to be erased without a break after inputting the "sec" command. Specify the start address when a prompt "start address (0xXXXX)>" is displayed, and the end address when a prompt "end address (0xXXXX)> " is displayed. Input the addresses by hexadecimal digits, then the specified addresses are displayed. A message "Sector erase complete" indicates that the erasing process has been completed successfully.

Specify sectors to be written without a break after inputting the "write" command. Specify the start address when a prompt "start address (0xXXXX)> " is displayed, and the end address when a prompt "end address (0xXXXX)>" is displayed. Input the addresses by hexadecimal digits, then the specified addresses are displayed. A message "Sector erase complete" indicates that the writing process has been completed successfully.

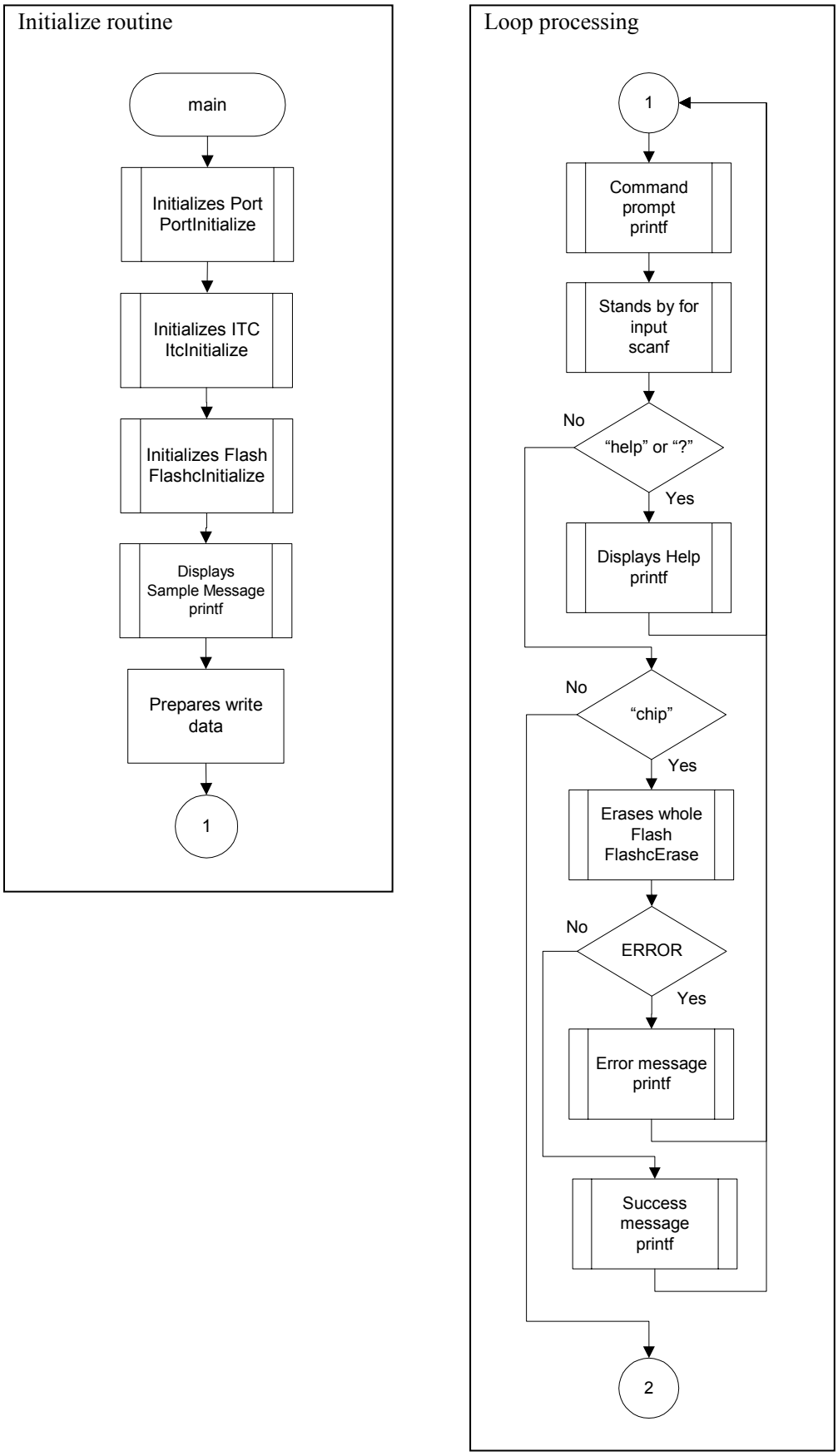
Inputting the "all" command starts to write the whole area in the FlashROM. A message "All write complete" indicates that the erasing process has been completed successfully.

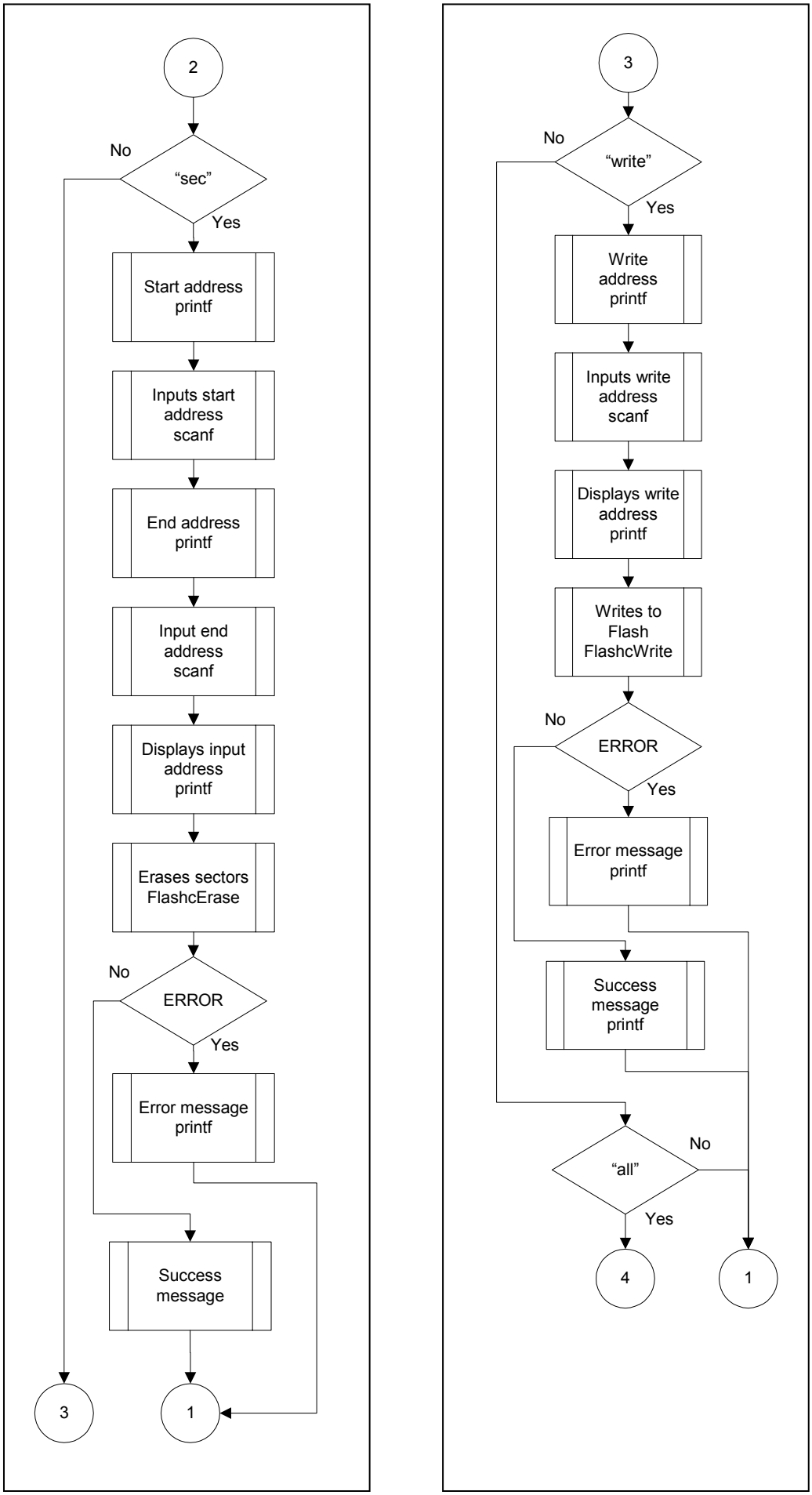
3.6 Outline of the sample program operations

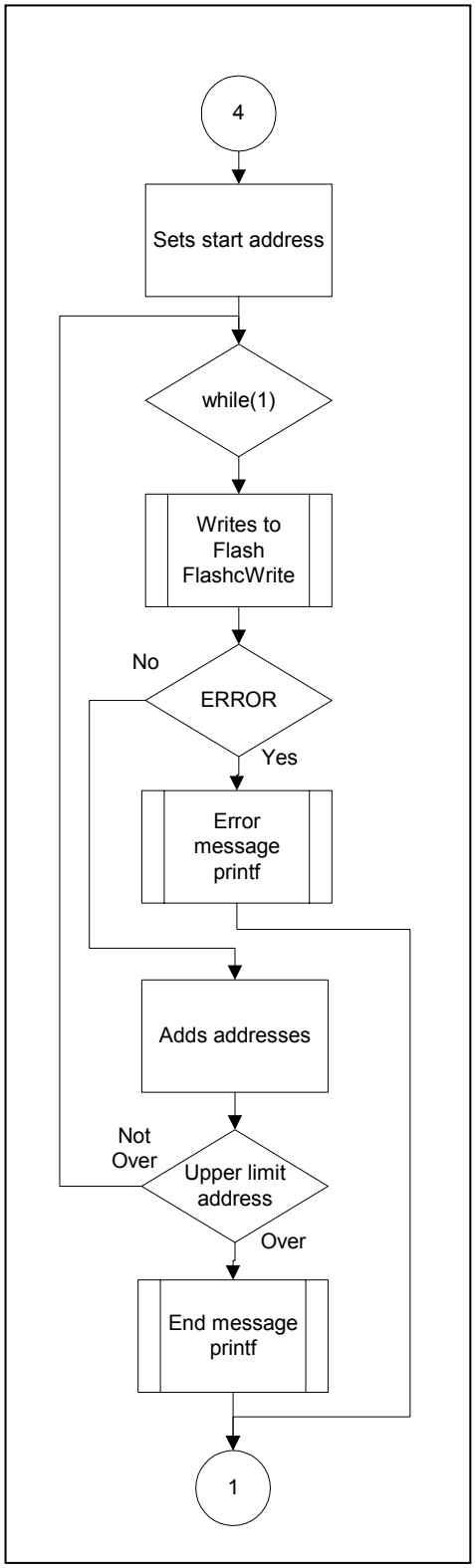
(1) Initialize the ITC.
(2) Initialize the FLASHC.
(3) Enable the CPU interrupt handler.
(4) Sets write data to data buffer.
(5) The program operates according to an input command from the Simulated I/O.

3.7 Flowchart

The following shows the flowchart of the main routine and interrupt handler functions.




2
No
"sec"
Yes
Start address printf
Inputs start address scanf
End address printf
Input end address scanf
Displays input address printf
Erases sectors FlashcErase
No
ERROR
Yes
Error message printf
Success message
3
1
3
No
"write"
Yes
Write address printf
Inputs write address scanf
Displays write address printf
Writes to Flash FlashcWrite
No
ERROR
Yes
Error message printf
Success message printf
No
"all"
Yes
4
1


4
Sets start address
while(1)
Writes to Flash FlashcWrite
No
ERROR
Yes
Error message printf
Adds addresses
Not Over
Upper limit address
Over
End message printf
1

3.8 Detailed Description of FLASHC Driver

The following explains the functions described in the files flashc_drv.c and flashc_api.h.

3.8.1 Initializing FLASHC

Format	**void FlashcInit(T_FLASHC_CFG* config)**
Function	Initializes the FLASHC functions.
Argument	config -in Initialization parameter for FLASHC
Return value	None

(Description)
Initializes the FLASHC.
 	
typedef struct {
unsigned short FlashWait; // Access wait value for the internal FlashROM
unsigned long TTBR; // Set value to TTBR
} T_FLASHC_CFG;
 	
Setting “0xFFFFFFFF” to TTBR disables writing TTBR.	

3.8.2 Setting FLASHC access wait

Format	**void FlashcSetFlashWait(unsigned char wait)**
Function	Sets FLASHC wait
Argument	wait -in Wait value FLASHC_WAIT_7 : 7 FLASHC_WAIT_6 : 6 FLASHC_WAIT_5 : 5 FLASHC_WAIT_4 : 4 FLASHC_WAIT_3 : 3 FLASHC_WAIT_2 : 2 FLASHC_WAIT_1 : 1 FLASHC_WAIT_0 : 0
Return value	None
(Description) Sets a WaitCycle value when reading the Flash.	

3.8.3 Obtaining FLASHC access wait

Format	**unsigned char FlashcGetFlashWait(void)**
Function	Obtains FLASHC wait value
Argument	None
Return value	FLASHC_WAIT_7 : 7 FLASHC_WAIT_6 : 6 FLASHC_WAIT_5 : 5 FLASHC_WAIT_4 : 4 FLASHC_WAIT_3 : 3 FLASHC_WAIT_2 : 2 FLASHC_WAIT_1 : 1 FLASHC_WAIT_0 : 0
(Description) Returns a WaitCycle value when reading the Flash.	

3.8.4 Sets FLASHC write protect

Format	**void FlashcProtect(void)**
Function	Sets FLASHC write protect
Argument	None
Return value	None
(Description) Sets the FLASHC register to the write disable state.	

3.8.5 Cancels FLASHC write protect

Format	**void FlashcUnprotect(void)**
Function	Cancels FLASHC write protect
Argument	None
Return value	None
(Description) Sets the FLASHC register to the write enable state.	

3.8.6 Sets FLASHC TTBR

Format	**void FlashcSetTTBR(unsigned long TTBR)**
Function	Sets FLASHC TTBR
Argument	TTBR -in TTBR address
Return value	None
(Description) Sets the address specified as TTBR.	

3.8.7 Obtaining FLASHC TTBR

Format	**unsigned long FlashcGetTTBR(void)**
Function	Sets FLASHC TTBR
Argument	None
Return value	TTBR set value
(Description) Returns the value set to TTBR.	

3.8.8 Erasing FLASHC Flash

Format	**int FlashcErase(unsigned char mode,** **unsigned long startAddress,** **unsigned long endAddress)**
Function	Erases the Flash
Argument	mode -in Erase mode FLASHC_ERASE_CHIP : Erases whole Flash FLASHC_ERASE_SECTOR : Erases flash sectors startAddress -in Erase start address endAddress -in Erase end address
Return value	SUCCESS: Terminated successfully ERROR: Failed in processing
(Description) Executes erasing process specified by the mode argument. StartAddress/EndAddress must be specified also for FLASHC_ERASE_CHIP. If an address that is not within the internal Flash area is specified, ERROR is returned. IF erasing process fails, ERROR is returned.	

3.8.9 Writing FLASHC Flash

Format	**int FlashcWrite(unsigned long address,** **unsigned long size,** **unsigned short *data)**
Function	Writes to Flash
Argument	address -in Write start address size -in Write size data -in Storage pointer to write data
Return value	SUCCESS: Terminated successfully ERROR: Failed in processing
(Description) Writes data with the size of area specified by the data argument, to the Flash starting from the address specified by the address argument. If writing process fails, ERROR is returned. If an odd number is specified for the address argument, ERROR is returned.	

3.9 Header Definitions

The tables below show the definitions used in the driver functions.

Definition name	Value	Description
CTL_STOP	0x0400	Stops erasing chips or sectors, or the Flash programming operations
CTL_STARTHOLD	0x0200	Starts the cycle holding operation mode
CTL_STARTERASE	0x0100	Starts erasing chips or sectors.
CTL_ERASE_CHIP	FLASHC_ERASE_CHIP	Sets the chip erase mode.
CTL_ERASE_SECTOR	FLASHC_ERASE_SECTOR	Sets the sector erase mode.
CTL_WRITE	0x0001	Sets the Flash programming mode
FLASHC_ERASE_CHIP	0x0004	Sets ERASE unit: CHIP unit
FLASHC_ERASE_SECTOR	0x0002	Sets ERASE unit: Sector unit
FLASHC_TOP_ADDRESS	0x20000	Start address
FLASHC_END_ADDRESS	0x3FFFF 0x37FFF	Depends on the end address or FlashROM size. （128KB: 0x3FFFF, 96KB: 0x37FFF）

3.10 Definition of a compile option

An unused function can be enabled or disabled by a compile option (Symbol definition).
The function is disabled by default value, which must be noted when compiling a program.

Symbol list

Definition name	Description
FOR_FLASHC_MINIMUM_API	The minimum functions required for controlling Flash are enabled.
FOR_FLASH_128	128K bytes of Flash size can be accessed.

* FOR_FLASH_128 is enabled in Make file (flashc_gnu17IDE.mak) in the sample program.

3.11 Operation speed

The “all” command is executed during CPU operation at 48MHz speed, enabling write to whole area in 2sec956ms.

REVISION HISTORY

Date	revision	modification	note
08/04/30	1.0	New issue	

EPSON

International Sales Operations

AMERICA

EPSON ELECTRONICS AMERICA, INC.
HEADQUARTERS
2580 Orchard Parkway
San Jose , CA 95131,USA
Phone: +1-800-228-3964 FAX: +1-408-922-0238

SALES OFFICES
Northeast
301 Edgewater Place, Suite 210
Wakefield, MA 01880, U.S.A.
Phone: +1-800-922-7667 FAX: +1-781-246-5443

EUROPE

EPSON EUROPE ELECTRONICS GmbH
HEADQUARTERS
Riesstrasse 15 Muenchen Bayern,
80992 GERMANY
Phone: +49-89-14005-0 FAX: +49-89-14005-110

ASIA

EPSON (CHINA) CO., LTD.
7F, Jinbao Bldg.,No.89 Jinbao St.,
Dongcheng District,
Beijing 100005, China
Phone: +86-10-6410-6655 FAX: +86-10-6410-7320

SHANGHAI BRANCH
7F, Block B, Hi-Tech Bldg., 900, Yishan Road,
Shanghai 200233, CHINA
Phone: +86-21-5423-5522 FAX: +86-21-5423-5512

EPSON HONG KONG LTD.
20/F., Harbour Centre, 25 Harbour Road
Wanchai, Hong Kong
Phone: +852-2585-4600 FAX: +852-2827-4346
Telex: 65542 EPSCO HX

EPSON (CHINA) CO., LTD.
SHENZHEN BRANCH
12/F, Dawning Mansion, Keji South 12th Road,
Hi- Tech Park, Shenzhen
Phone: +86-755-2699-3828 FAX: +86-755-2699-3838

EPSON TAIWAN TECHNOLOGY & TRADING LTD.
14F, No. 7, Song Ren Road,
Taipei 110
Phone: +886-2-8786-6688 FAX: +886-2-8786-6660

EPSON SINGAPORE PTE., LTD.
1 HarbourFront Place,
#03-02 HarbourFront Tower One, Singapore 098633
Phone: +65-6586-5500 FAX: +65-6271-3182

SEIKO EPSON CORPORATION
KOREA OFFICE
50F, KLI 63 Bldg., 60 Yoido-dong
Youngdeungpo-Ku, Seoul, 150-763, KOREA
Phone: +82-2-784-6027 FAX: +82-2-767-3677

GUMI OFFICE
2F, Grand B/D, 457-4 Songjeong-dong,
Gumi-City, KOREA
Phone: +82-54-454-6027 FAX: +82-54-454-6093

SEIKO EPSON CORPORATION
SEMICONDUCTOR OPERATIONS DIVISION

IC Sales Dept.
IC International Sales Group
421-8, Hino, Hino-shi, Tokyo 191-8501, JAPAN
Phone: +81-42-587-5814 FAX: +81-42-587-5117

Document Code: 411446100
First issue March 2008 in JAPAN Ⓗ