SH7268/SH7269 Group

E10A-USB Flash Memory Download Function (Download to the NOR Flash Memory)

R01AN1176EJ0100
Rev.1.00
Jul. 18, 2012

Abstract

E10A-USB emulator has the function to download a load module to the flash memory. This function requires a download program to access the flash memory (hereinafter called "FMTOOL").

This application note describes how to download a load module to the NOR flash memory (JEDEC standard compatible command method) to which the FMTOOL is applied.

Target Device

SH7268/SH7269 Group (hereinafter called "SH7269")

When using this application note with other Renesas MCUs, careful evaluation is recommended after making modifications to comply with the alternate MCU.

Contents

1. Specifications

Download the load module which is allocated in the external address space (CS0 space) to the flash memory using the FMTOOL. The FMTOOL allows the flash memory accessed when it is connected to the CS0 space (16-bit bus).

Table 1.1 lists the Peripheral Functions and Their Applications and Figure 1.1 shows the Procedure of Download Using FMTOOL.

Table 1.1 Peripheral Functions and Their Applications

Peripheral Function	Application
Bus state controller (CS0)	Downloads to the flash memory
On-chip high-speed RAM	Work memory
H-UDI	Connects the E10A-USB emulator

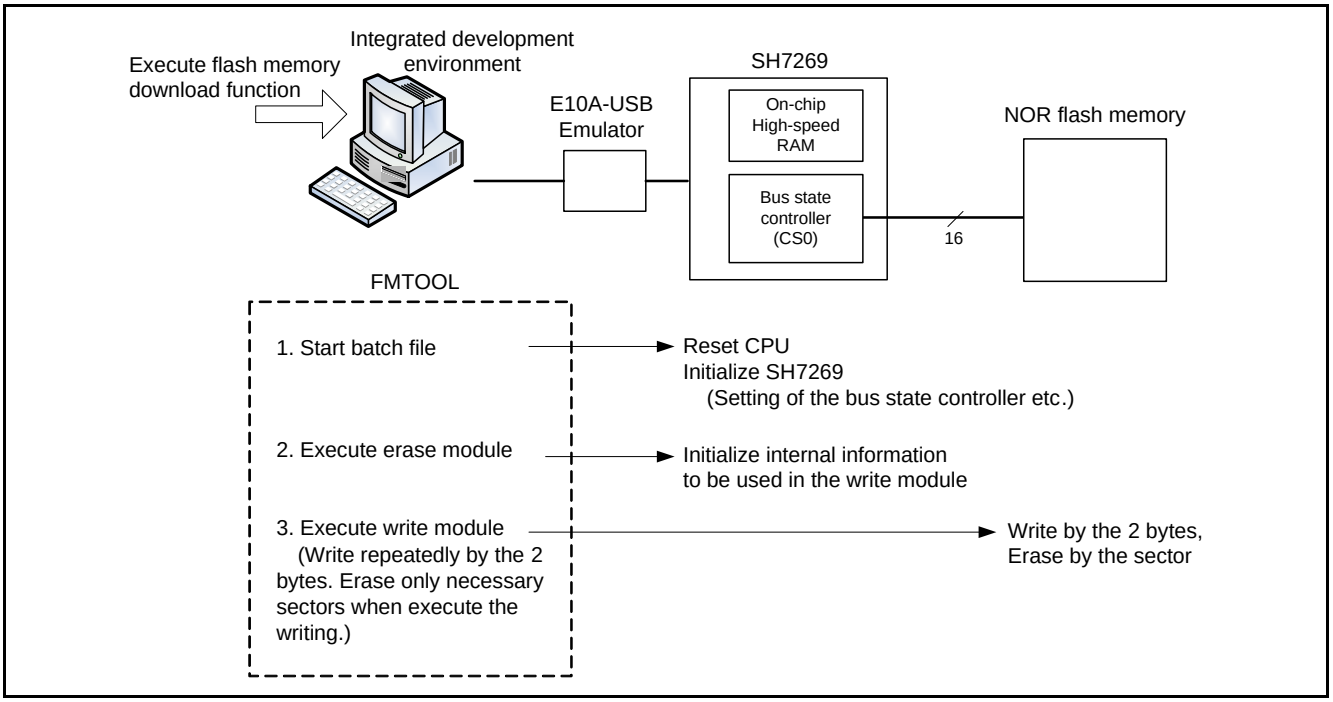


Figure 1.1 Procedure of Download Using FMTOOL

2. Operation Confirmation Conditions

The sample code accompanying this application note has been run and confirmed under the conditions below.

Table 2.1 Operation Confirmation Conditions

Item	Contents
MCU used	SH7269
Device used	Flash memory (connected by16-bit bus) Manufacturer: Spansion Model: S29GL256P90TFIR1
Operating frequency	CPU internal clock (Iϕ): 266.67MHz Internal clock (Bϕ): 133.33MHz Peripheral clock 1 (P1ϕ): 66.67MHz Peripheral clock 0 (P0ϕ): 33.33MHz
Operating voltage	Source power (I/O): 3.3V Source power (internal): 1.25V
Integrated development environment	Renesas Electronics High-performance Embedded Workshop Ver.4.07.00
C compiler	Renesas Electronics SuperH RISC engine Family C/C++Compiler Package Ver.9.03 Release02
	Complier option -cpu=sh2afpu -fpu=single -object="$(CONFIGDIR)\$(FILELEAF).obj" -debug -gbr=auto -chgincpath -errorpath -global_volatile=0 -opt_range=all -infinite_loop=0 -del_vacant_loop=0 -struct_alloc=1 -nologo (with default setting in the integrated development environment)
Board used	R0K572690C000BR

3. Reference Application Note

For additional information associated with this document, refer to the following application note.

- Flash Memory Download Program for the E10A-USB Emulator Application Note (document No. REJ10J1221)

4. Hardware

4.1 Pins used

Table 4.1 lists the Used Pins and Their Functions.

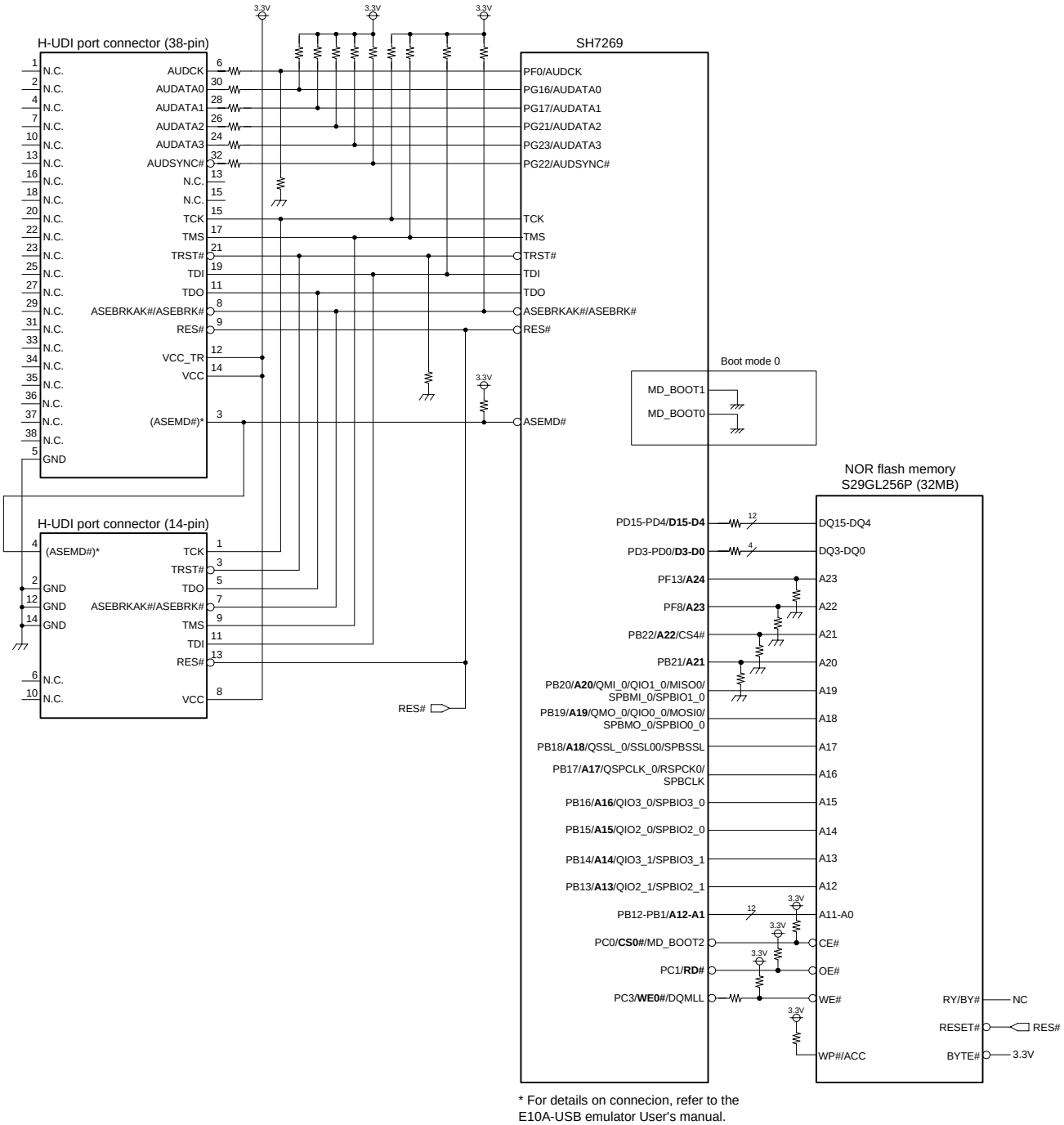
Table 4.1 Used Pins and Their Functions

Pin Name	Input/Output	Function
A24 to A1	Output	Address bus
D15 to D0	Input/output	Data bus
CS0#	Output	Output device selection signal to the flash memory
RD#	Output	Output read control signal to the flash memory
WE0#	Output	Output write enable control signal to the flash memory
MD_BOOT1, MD_BOOT0	Input	Selection of boot mode
AUDCK	Output	Clock output to the E10A-USB emulator (38-pin)
AUDATA3 to AUDATA0	Output	Address output to the E10A-USB emulator (38-pin)
AUDSYNC#	Output	Synchronous signal output to the E10A-USB emulator (38-pin)
TCK	Input	Clock input from the E10A-USB emulator
TMS	Input	Mode selection from the E10A-USB emulator
TRST#	Input	Reset input from the E10A-USB emulator
TDI	Input	Data input from the E10A-USB emulator
TDO	Output	Data output to the E10A-USB emulator
ASEBRKAK#/ASEBRK#	Input/output	Break request and response
RES#	Input	System reset signal
ASEMD#	Input	Selection of ASE mode

Note: The symbol "#" indicates a negative-true logic or an active low.

4.2 Reference Circuit

Figure 4.1 shows the Connection Example when downloading to the NOR flash memory.



* For details on connecion, refer to the E10A-USB emulator User's manual.

Note: The symbol "#" indicaates a negative-true logic or an active low.

Figure 4.1 Connection Example

5. Software

5.1 Operation Overview

The FMTOOL consists of two programs; the erase module and the write module. The E10A-USB emulator writes program data in the flash memory using these programs. For details on these modules, refer to the section "6.22 Download Function to the Flash Memory" in the "Super H™ Family E10A-USB Emulator User's Manual".

In the sample code, without using a chip erase, only necessary sectors are erased to speed up the download process. The erase processing for the sectors is executed in the write module.

5.1.1 Batch File

Executes a reset command to initialize the SH7269 using the batch file which is executed before downloading the load module. For details on the batch file and the reset command, refer to the manual listed in the integrated development environment on our web site.

5.1.2 Erase Module

When commencing download of the load module, the FMTOOL is transferred to the on-chip high-speed RAM on the SH7269. The erase module is executed only once after the transfer.

Although the primary purpose of the erase module operation is to implement a chip erase processing, initialization of the internal information to be used in the write module is executed in the sample code.

5.1.3 Write Module

The write module is executed repeatedly when downloading the load module. The program data divided into the access size is received as the argument.

Figure 5.1 shows the Write Module Diagram. The write module is executed on the on-chip high-speed RAM of the SH7269, receives the program data address and word data as the argument, and writes into the flash memory. When the destination address is in the sector that has not been erased, writes after erasing the sector.

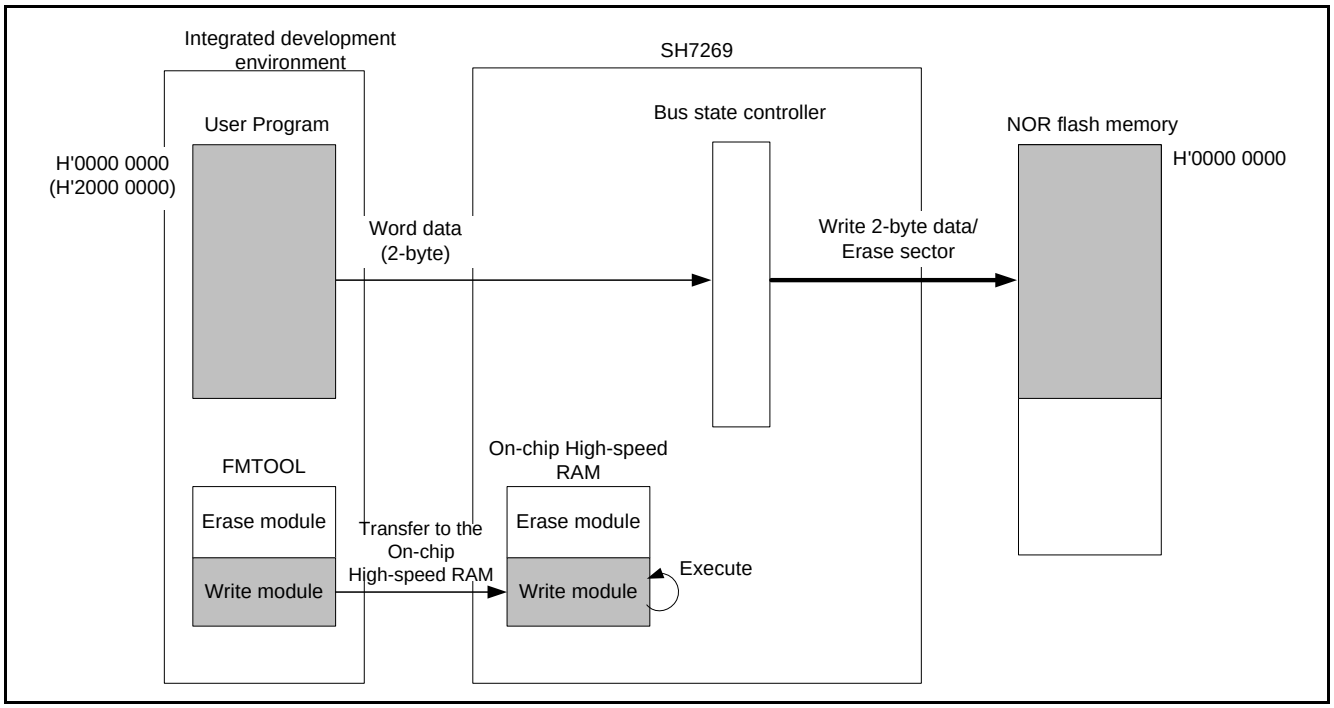


Figure 5.1 Write Module Diagram

5.2 File Composition

Table 5.1 lists the File Composition. Files generated by the integrated development environment should not be listed in this table.

Table 5.1 File Composition

File Name	Outline	Remarks
fmtool_entry.src	Entry module of the FMTOOL	Entry of erase module and write module
fmtool_main.c	Main module of the FMTOOL	
flash_cfg.h	Configuration of the flash memory	Configuration file of the flash memory (Spansion, S29GL256P) Customizing is required when the above flash memory does not comply with the device specification
sh7269_fmtool.hdc	Batch file	Registers in the integrated development environment

5.3 Constants

Table 5.2 lists the Constants Used in the Sample Code **(1/3)**.

Table 5.2 Constants Used in the Sample Code (1/3)

Constant Name	Setting Value	Contents	Need for change
FM_UNIFORM	0	Uniform type	
FM_TOP_BOOT	1	Top boot type	
FM_BOTTOM_BOOT	2	Bottom boot type	
FM_DUAL_BOOT	3	Dual boot type	
FM_BOOT_TYPE	FM_UNIFORM	Type of sector structure	x
FM_B_BOOT_SECTOR_START	0x00000000UL	Start address of the bottom boot sector (by the word)	x
FM_B_BOOT_SECTOR_SIZE	0x00000000UL	Size per sector for bottom boot sector (by the word)	x
FM_B_BOOT_SECTOR_NUM	0	Number of bottom boot sectors	x
FM_NORMAL_SECTOR_START	0x00000000UL	Start address of the area except for the boot sector (by the word)	x
FM_NORMAL_SECTOR_SIZE	0x00010000UL	Size per sector for the area except for the boot sector (by the word)	x
FM_NORMAL_SECTOR_NUM	256	Number of sectors in the area except for the boot sector	x
FM_T_BOOT_SECTOR_START	0x00000000UL	Start address of the top boot sector (by the word)	x
FM_T_BOOT_SECTOR_SIZE	0x00000000UL	Size per sector for top boot sector (by the word)	x
FM_T_BOOT_SECTOR_NUM	0	Number of top boot sectors	x
FM_END_ADDRESS	0x00FFFFFFUL	End address of the flash memory (by the word)	x
FM_CMD_RESET	0x00F0	Reset command	x
FM_CMD_S_ERASE_ADDR_1	0x0555	Address of 1st cycle sector erase command (address for word command)	x
FM_CMD_S_ERASE_ADDR_2	0x02AA	Address of 2nd cycle sector erase command (address for word command)	x
FM_CMD_S_ERASE_ADDR_3	0x0555	Address of 3rd cycle sector erase command (address for word command)	x
FM_CMD_S_ERASE_ADDR_4	0x0555	Address of 4th cycle sector erase command (address for word command)	x
FM_CMD_S_ERASE_ADDR_5	0x02AA	Address of 5th cycle sector erase command (address for word command)	x

Note: The constants marked with "x" in the "need for change" depend on the flash memory specification to be used. "6.2 Customizing FMTOOL" describes customizing method.

Table 5.3 Constants Used in the Sample Code (2/3)

Constant Name	Setting Value	Contents	Need for change
FM_CMD_S_ERASE_DATA_1	0x00AA	Data of 1st cycle sector erase command	x
FM_CMD_S_ERASE_DATA_2	0x0055	Data of 2nd cycle sector erase command	x
FM_CMD_S_ERASE_DATA_3	0x0080	Data of 3rd cycle sector erase command	x
FM_CMD_S_ERASE_DATA_4	0x00AA	Data of 4th cycle sector erase command	x
FM_CMD_S_ERASE_DATA_5	0x0055	Data of 5th cycle sector erase command	x
FM_CMD_SECTOR_ERASE	0x0030	Data of 6th cycle sector erase command	x
FM_CMD_PROGRAM_ADDR_1	0x0555	Address of 1st cycle single word program command (address for word command)	x
FM_CMD_PROGRAM_ADDR_2	0x02AA	Address of 2nd cycle single word program command (address for word command)	x
FM_CMD_PROGRAM_ADDR_3	0x0555	Address of 3rd cycle single word program command (address for word command)	x
FM_CMD_PROGRAM_DATA_1	0x00AA	Data of 1st cycle single word program command	x
FM_CMD_PROGRAM_DATA_2	0x0055	Data of 2nd cycle single word program command	x
FM_CMD_PROGRAM_DATA_3	0x00A0	Data of 3rd cycle single word program command	x
FM_CHK_DQ5	0x0020	Mask information of bit for excess of timing limit (DQ5)	
FM_CHK_DQ6	0x0040	Mask information of toggle bit (DQ6)	
FM_CHK_DQ7	0x0080	Mask information of data polling bit (DQ7)	
FM_CS0_NON_CACHE_START	0x20000000UL	Start address of the SH7269 CS0 space (cache-disabled space)	
FM_TYPE_BYTE	0x4220	R5 parameter of write module (access the flash memory by the byte-size)	
FM_TYPE_WORD	0x5720	R5 parameter of write module (access the flash memory by the word-size)	
FM_TYPE_LONG	0x4C20	R5 parameter of write module (access the flash memory by the long-size)	

Note: The constants marked with "x" in the "need for change" depend on the flash memory specification to be used. "6.2 Customizing FMTOOL" describes customizing method.

Table 5.4 Constants Used in the Sample Code (3/3)

Constant Name	Setting Value	Contents	Need for change
FM_TOOL_OK	0	Return value: no error	
FM_TOOL_E_ERASE	-1	Return value: erase error	
FM_TOOL_E_WRITE	-2	Return value: write error	
FM_TOOL_E_VERIFY	0x4254	Return value: verify error	
FM_TOOL_E_ARGUMENT	-16	Return value: argument error	
FM_WRITE_TOP	H'FFF80000	Start address of write module	
FM_ERASE_TOP	H'FFF81000	Start address of erase module	
FM_STACK_TOP	H'FFF90000	Stack pointer the FMTOOL uses	
FM_SECTOR_NUM	Total number of sectors	Number of all sectors (FM_T_BOOT_SECTOR_NUM+ FM_B_BOOT_SECTOR_NUM+ FM_NORMAL_SECTOR_NUM)	

Note: The constants marked with "x" in the "need for change" depend on the flash memory specification to be used. "6.2 Customizing FMTOOL" describes customizing method.

5.4 Variable

Table 5.5 lists the Static Variable

Table 5.5 Static Variable

Type	Variable Name	Contents	Function Used
uint32_t	fmtool_pre_erase_sctno[(FM_SECTOR_NUM/32)+1]	Management information of the erased sectors	FmtoolInit, FmtoolWrite, FmCheckPreErase

5.5 Functions

Table 5.6 lists the Functions.

Table 5.6 Functions

Function Name	Outline
_FMTOOL_ERASE	Entry processing for the erase module
_FMTOOL_WRITE	Entry processing for the write module
FmtoolInit	Main processing for the erase module (Initialization)
FmtoolWrite	Main processing for the write module (erasing/writing)
FmResetCmd	Reset processing for the flash memory
FmSectorEraseCmd	Sector erase processing
FmWordProgramCmd	Single word programming
FmCheckPreErase	Checking the erased sectors

5.6 Function Specifications

The following tables list the function specifications of the sample code.

_FMTOOL_ERASE

Outline	Entry processing for the erase module
Header	None
Declaration	_FMTOOL_ERASE:
Description	Allocates in the address H'FFF8 1000 in the entry section of the erase module which is commenced by the E10A-USB flash memory download function. This module saves the register to execute the function Fmtoollmit after setting the stack pointer. Then it restores the register and places the control back to the E10A-USB emulator according to the RTS instruction.
Argument	R4 register : Access size (Byte: H'4220, Word: H'5720, Long: H'4C20)
Return Value	None
Remarks	Described in the assembly language

_FMTOOL_WRITE

Outline	Entry processing for the write module
Header	None
Declaration	_FMTOOL_WRITE:
Description	Allocates in the address H'FFF8 0000 in the entry section of the write module which is commenced by the E10A-USB flash memory download function. This module saves the register to execute the function FmtoolWrite after setting the stack pointer. Then it restores the register and places the control back to the E10A-USB emulator according to the RTS instruction.
Argument	R4 register : Write address R5 register : Access size (Byte: H'4220, Word: H'5720, Long: H'4C20) R6 register : Write data
Setting Value	R0 register = 0: Normal end Except for R0 resister = 0: Error end
Remarks	Described in the assembly language

FmtoolInit

Outline	Main processing for the erase mod (initialization)
Header	"flash_cfg.h"
Declaration	void FmtoolInit(uint32_t access_size);
Description	Initializes the variable used in main processing for write module. This function is executed from the entry point of the FMTOOL (_FMTOOL_ERASE)
Argument	First argument: access_size : Access size (unused) (Byte: H'4220, Word: H'5720, Long:H'4C20)
Return Value	None
Remarks	

FmtoolWrite

Outline	Main processing for the write module (erasing/writing)
Header	"flash_cfg.h"
Declaration	int32_t FmtoolWrite(uint32_t addr, uint32_t access_size, uint32_t data);
Description	Executes erasing and writing of the flash memory. The flash memory is accessed by the sector for erasing and by the 16-bit for writing. When the destination address specified by the argument addr is not erased, writes after erasing the sector. When the sector has been erased, executes only writing. This function is executed from the entry point of the FMTOOL (_FMTOOL_WRITE)
Argument	First argument: addr : Write address Second argument: access_size : Access size (Byte: H'4220, Word: H'5720, Long: H'4C20) Third argument: data : Write data
Return Value	0: Normal end -1: Error end (erase error) -2: Error end (write error) H'4254: Error end (verify error) -16: Error end (argument error)
Remarks	Only word size is available for the access size. When any size except for the word size is specified, end this function with argument error.

FmResetCmd

Outline	Reset processing for the flash memory
Header	"flash_cfg.h"
Declaration	void FmResetCmd(volatile uint16_t *pResetAddr);
Description	Executes a reset command to the flash memory
Argument	First argument: *pResetAddr : Pointer to the address for reset
Return Value	None
Remarks	

FmSectorEraseCmd

Outline	Sector erase processing
Header	"flash_cfg.h"
Declaration	int32_t FmSectorEraseCmd(volatile uint16_t *pEraseBlock);
Description	Executes the sector erase to the specified sector. Checks the toggle bit and the polling bit after executing the erase command, and ends this function with error when the erase error comes up.
Argument	First argument: *pEraseBlock : Pointer to start address of the sector for erase
Return Value	0: Normal end -1: Error end (erase error)
Remarks	

FmWordProgramCmd

Outline	Single word programming
Header	"flash_cfg.h"
Declaration	int32_t FmWordProgramCmd(volatile uint16_t *pWriteAddr, uint16_t Data);
Description	Writes the specified data into the specified address. Checks the toggle bit and the polling bit after executing the write command and ends this function with error when the write error comes up. When the data can be written without error, executes verification, and ends this function with error if the verify error comes up.
Argument	First argument: *pWriteAddr : Pointer to the address for write Second argument: Data : Write data
Return Value	0: Normal end -2: Error end (write error) H'4254: error end (verify error)
Remarks	

FmCheckPreErase

Outline	Checking erased sectors
Header	"flash_cfg.h"
Declaration	int32_t FmCheckPreErase(uint32_t sect_no);
Description	Checks whether or not the address specified write address has been erased, and returns the result
Argument	First Argument: sect_no : Sector number
Return Value	0: The sector of write address is not erased 1: The sector of write address is erased
Remarks	

5.7 Flowcharts

This section describes the procedure of the batch file processing and the procedures of the major functions used in the sample code.

5.7.1 Batch File

Figure 5.2 shows the procedure of the Batch File Processing.

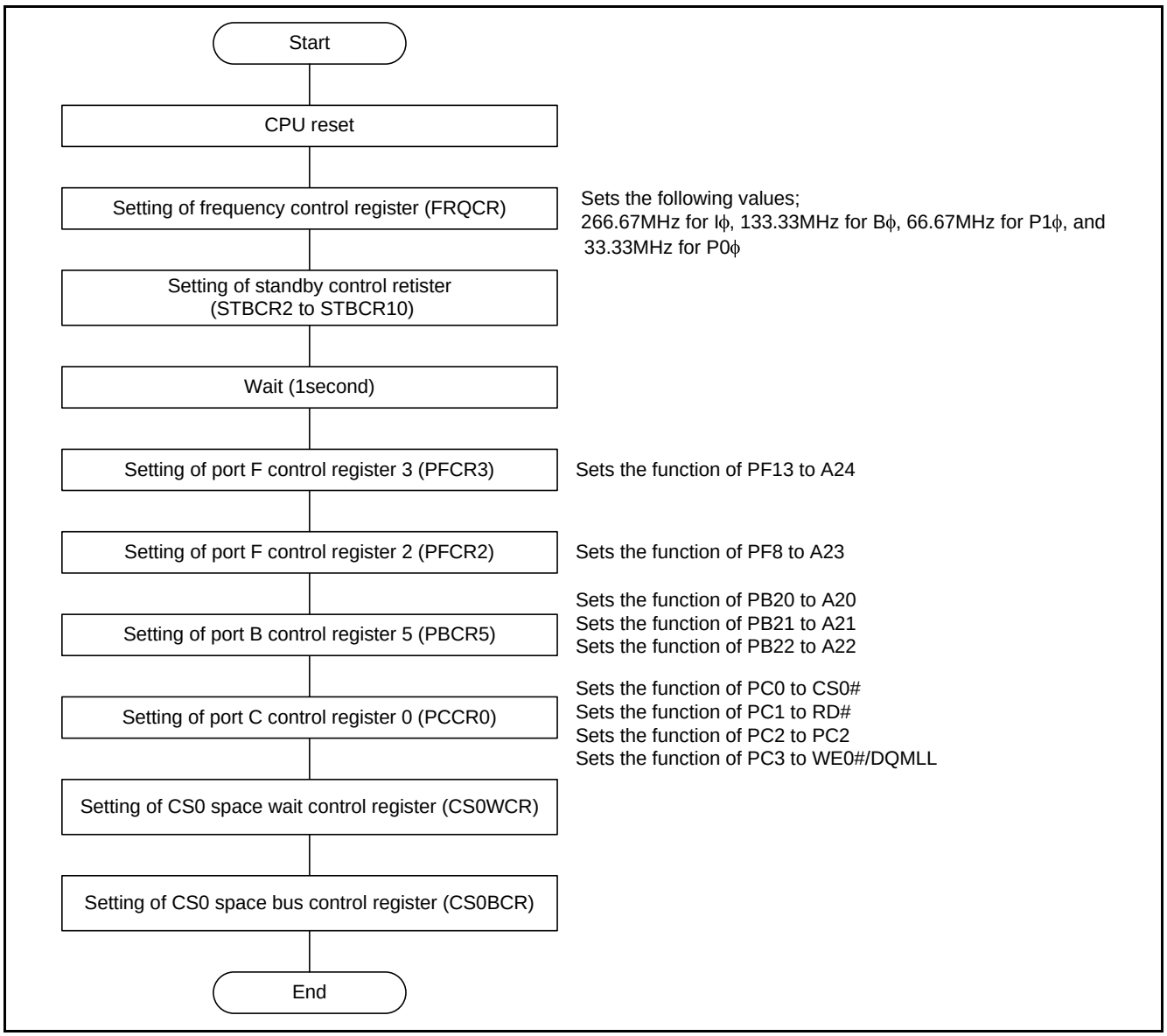


Figure 5.2 Batch File Processing

Note: On the SH7269, the pins of A25 to A21 have port functions in the initial state. When using these pins, it is required to set the address bus function in the batch file.

5.7.2 Erase Module

Figure 5.3 shows the Procedure of Erase Module.

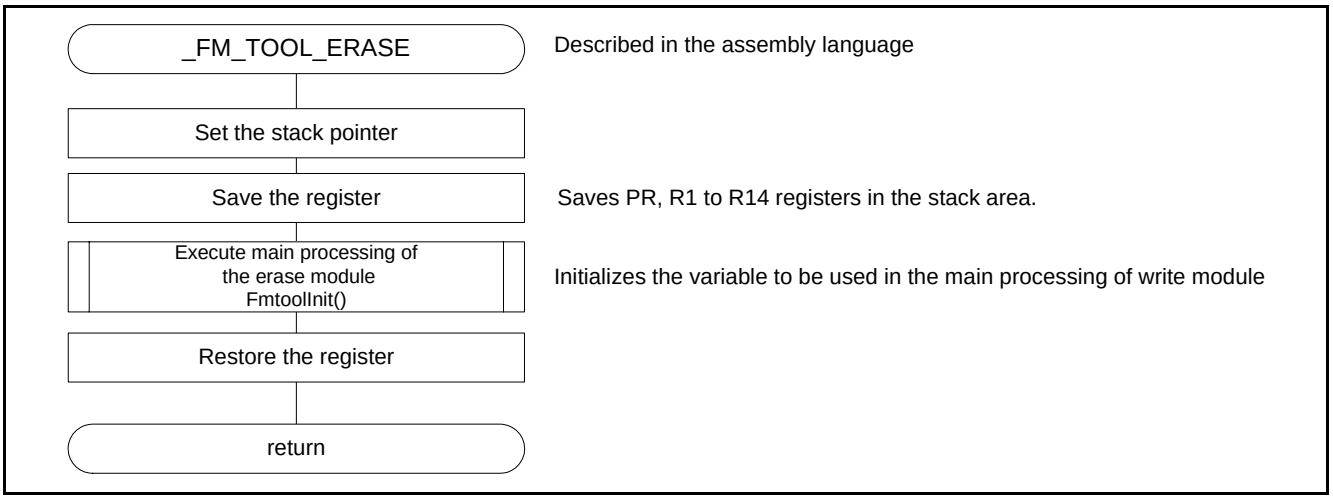


Figure 5.3 Procedure of Erase Module

5.7.3 Write module

Figure 5.4 shows the Procedure of Write Module.

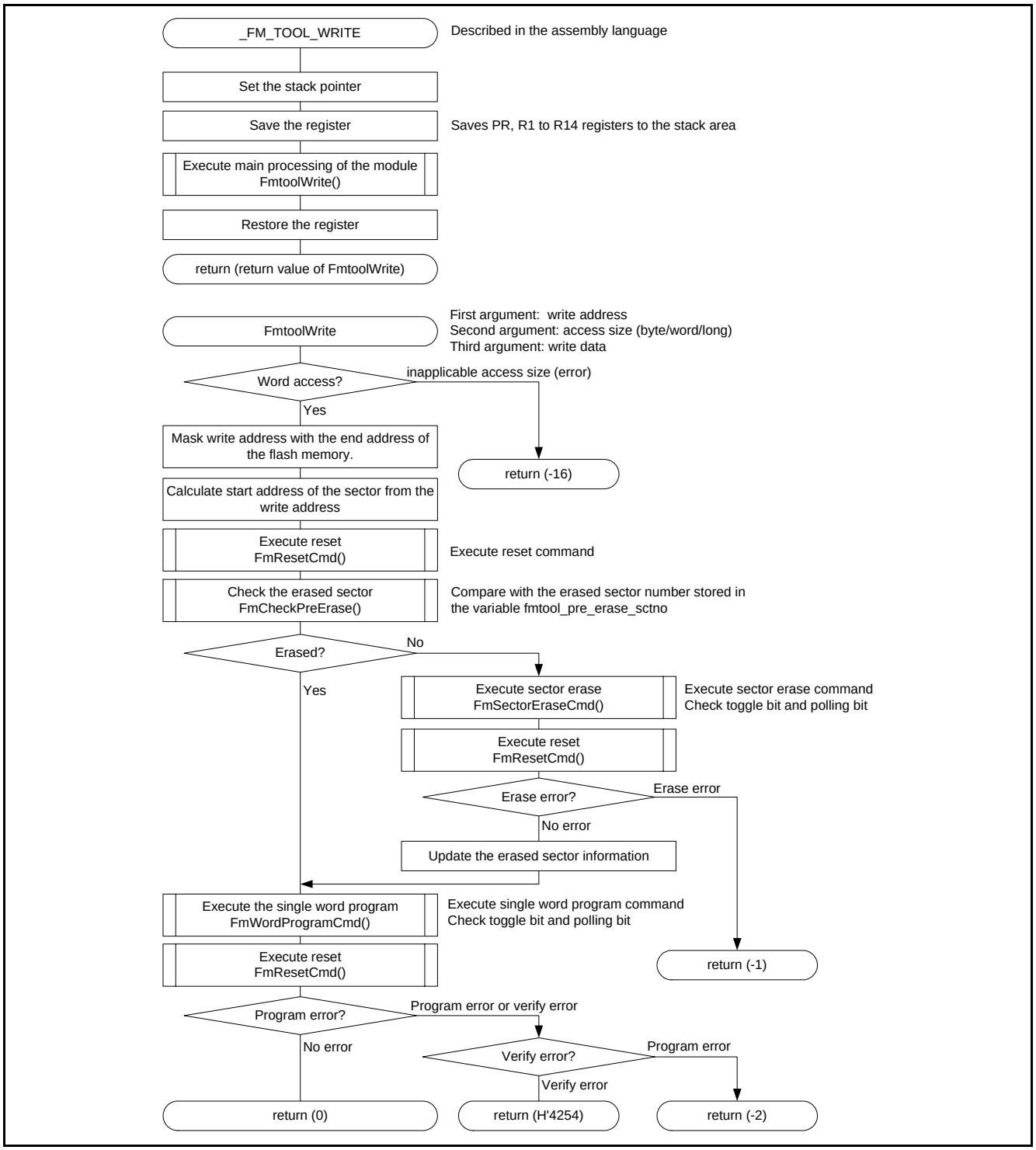


Figure 5.4 Procedure of Write Module

6. Application Example

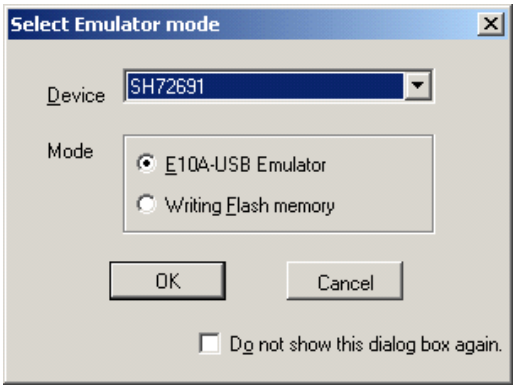
6.1 Procedure of User Program Download

This section describes the procedure of downloading user programs to the flash memory using the created FMTOOL (sh7269_fmtool.mot).

6.1.1 Prepare for Download Environment

1. Connect user's system with the E10A-USB emulator conned to PC.
2. Start the High-performance Embedded Workshop to open the work space for user programs.
3. The Device Select Dialog Box is open as shown in Figure 6.1.

   Select the CPU in use in the drop-down listbox for Device. Click the OK button.



Note: The shown window is an example agopting the SH72691

Figure 6.1 Device Select Dialog Box

4. The Connecting box is displayed and emulator connection starts.

   The RESET Signal Request Dialog Box shown in Figure 6.2 is displayed.

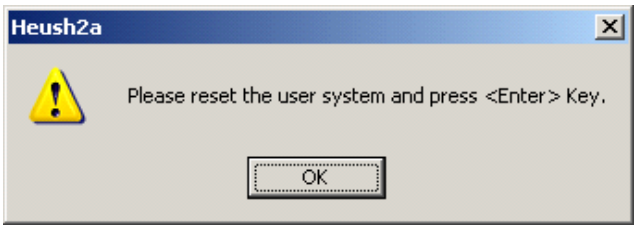


Figure 6.2 RESET Signal Request Dialog Box

5. Turn on the user's system.

   Having received the RESET signal from the user's system, click the OK button.

   When "connected" is displayed on the Output Window in the High-performance Embedded Workshop, the E10A-USB emulator successfully started.

6.1.2 Registering Batch File

1. Select in the menu; [Debug] → [Debug Settings]
2. The Window for Debug Setting shown in Figure 6.3 opens.
3. Select "Before download modules" in the pull-down menu for the "Command batch file load timing".
4. Click the "Add" at "Command line batch processing" to add a batch file.
5. Click the OK button, and registration is completed.

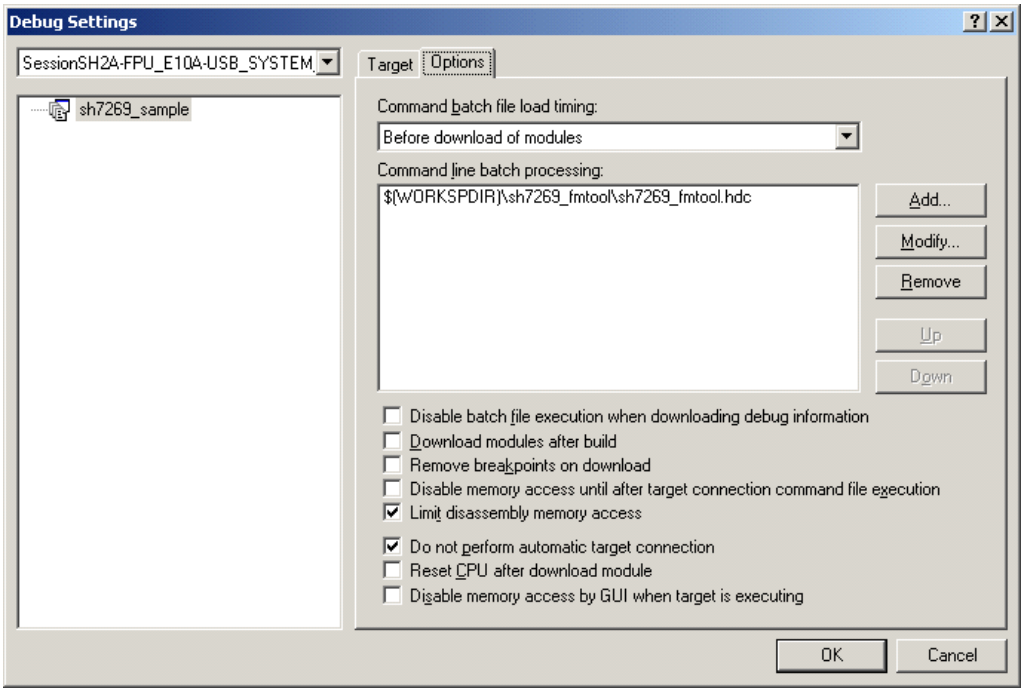


Figure 6.3 Window for Debug Setting

6.1.3 Setting Configuration Dialog Box

1. Select in the menu; [Setup] → [Emulator] → [System]
2. Figure 6.4 shows the Configuration Dialog Box (in the page of loading flash memory) for setting to download a user program to the external flash memory using the E10A-USB emulator.

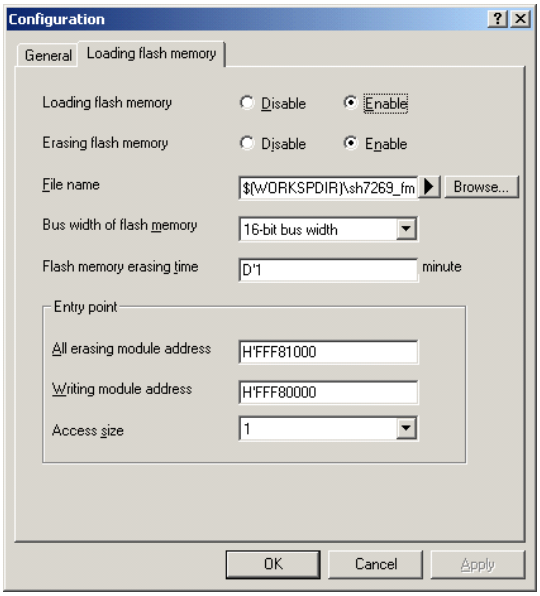


Figure 6.4 Configuration Dialog Box (in the page of loading flash memory)

Table 6.1 lists the setting in the items in the configuration dialog box. Finish setting and click the OK button, configuration is completed.

Table 6.1 Setting Value in the Configuration Dialog Box

Item	Setting Value
Loading flash memory	Enable
Erasing flash memory	Enable
File Name	(Directory stores the FMTOOL) \sh7269_fmtool.mot
Bus width of flash memory	16-bit bus width
All erasing module address	Assign start address of the erase module (H'FFF8 1000)
Writing module address	Assign start address of the write module (H'FFF8 0000)

6.1.4 Adding Download Module

Open the debug setting window from the debug menu. Click "Add". In the Download Module Window shown in Figure 6.5, add user programs to be loaded in the serial flash memory to the download module.

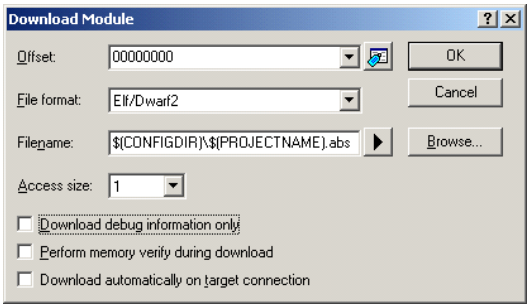


Figure 6.5 Download Module Window

6.1.5 Downloading User Programs

Using the download function shown in Figure 6.6, download the user programs.

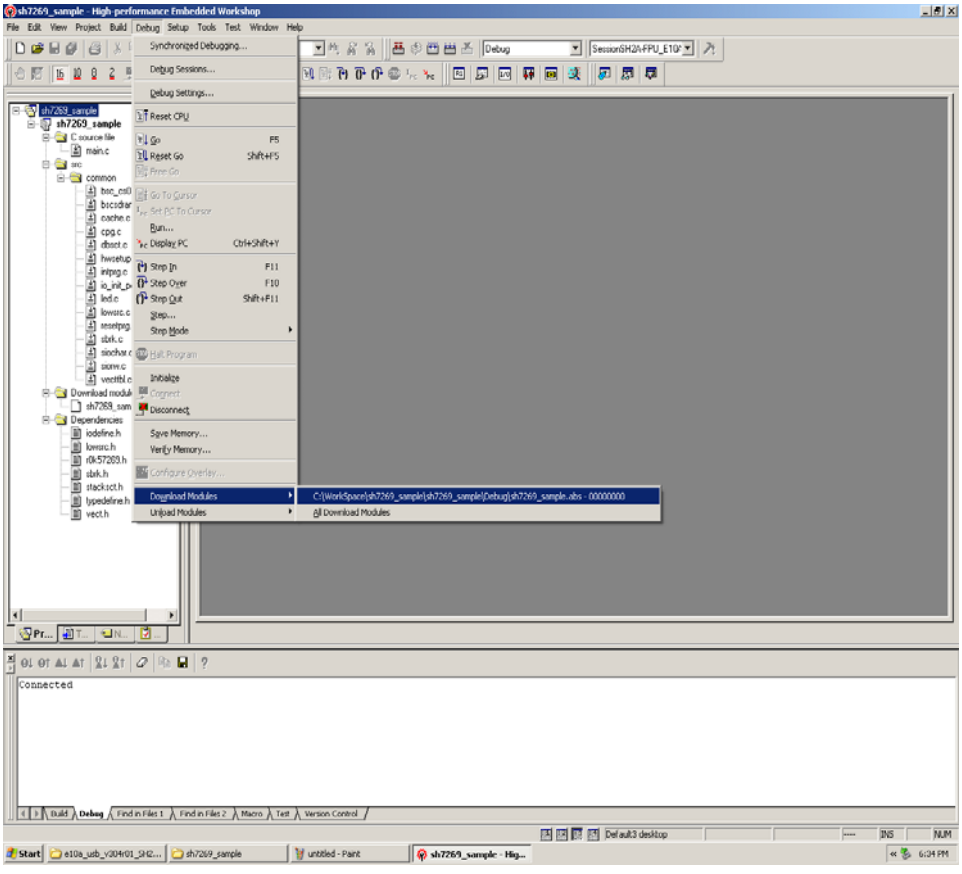


Figure 6.6 Downloading User Programs

6.2 Customizing FMTOOL

The sample code is dependent on the specification of the device in the serial flash memory. Customization of the program may be necessary when altering the device.

The sample code corresponds to the JEDEC standard compatible command method. When using the flash memory of the method, the sample code can be applied.

When using the flash memory of the CUI (Command User interface) command method, the sample code cannot be applied. In this case, the user needs to produce a new download program.

Notes: 1. JEDEC standard compatible command method is a writing method which requires to issue a command to the predetermined address (H'555, H'2AA, and etc.).
2. CUI command method is the method which requires issuing the write command (H'40) and the erase command (H'20) to the CUI.

6.2.1 Device Specification for Sample Code

Table 6.2 and Table 6.3 list the Specification of Used Device and the Commands Used in Sample Code respectively.

Table 6.2 Specification of Used Device

Item	Details
Manufacturer	Spansion Inc.
Model	S29GL256P90TFIR1
Capacity	32MB
Data bus width	16-bit
Access time	90ns
Sector structure	Uniform type
Sector size	64K words
Number of sector	256
Writing method	JEDEC standard compatible command method

Table 6.3 Commands Used in Sample Code

Item	Description		
	Cycle	Address	Data
Erase command (Sector erase)	1st cycle	H'555	H'AA
	2nd cycle	H'2AA	H'55
	3rd cycle	H'555	H'80
	4th cycle	H'555	H'AA
	5th cycle	H'2AA	H'55
	6th cycle	SA *1	H'30
Program command	1st cycle	H'555	H'AA
	2nd cycle	H'2AA	H'55
	3rd cycle	H'555	H'A0
	4th cycle	PA *2	PD *3
Reset command	1st cycle	- *4	H'F0

Notes: *1. SA indicates the sector address. It assigns sector address to be erased.
*2. PA indicates the program address. It assigns address to be written.
*3. PD indicates the program data. It assigns data to be written.
*4. The symbol "-" indicates the address of the space where the flash memory is allocated. Any address can be assigned in the space where the flash memory is allocated.

6.2.2 Applicable Flash Memory Boot Types

The following four flash memory boot types are applicable to the SH7269 by customizing the sample code.

1) Uniform type
2) Bottom boot type
3) Top boot type
4) Dual boot type

Figure 6.7 shows the Memory Map for Flash Memory Boot Types.

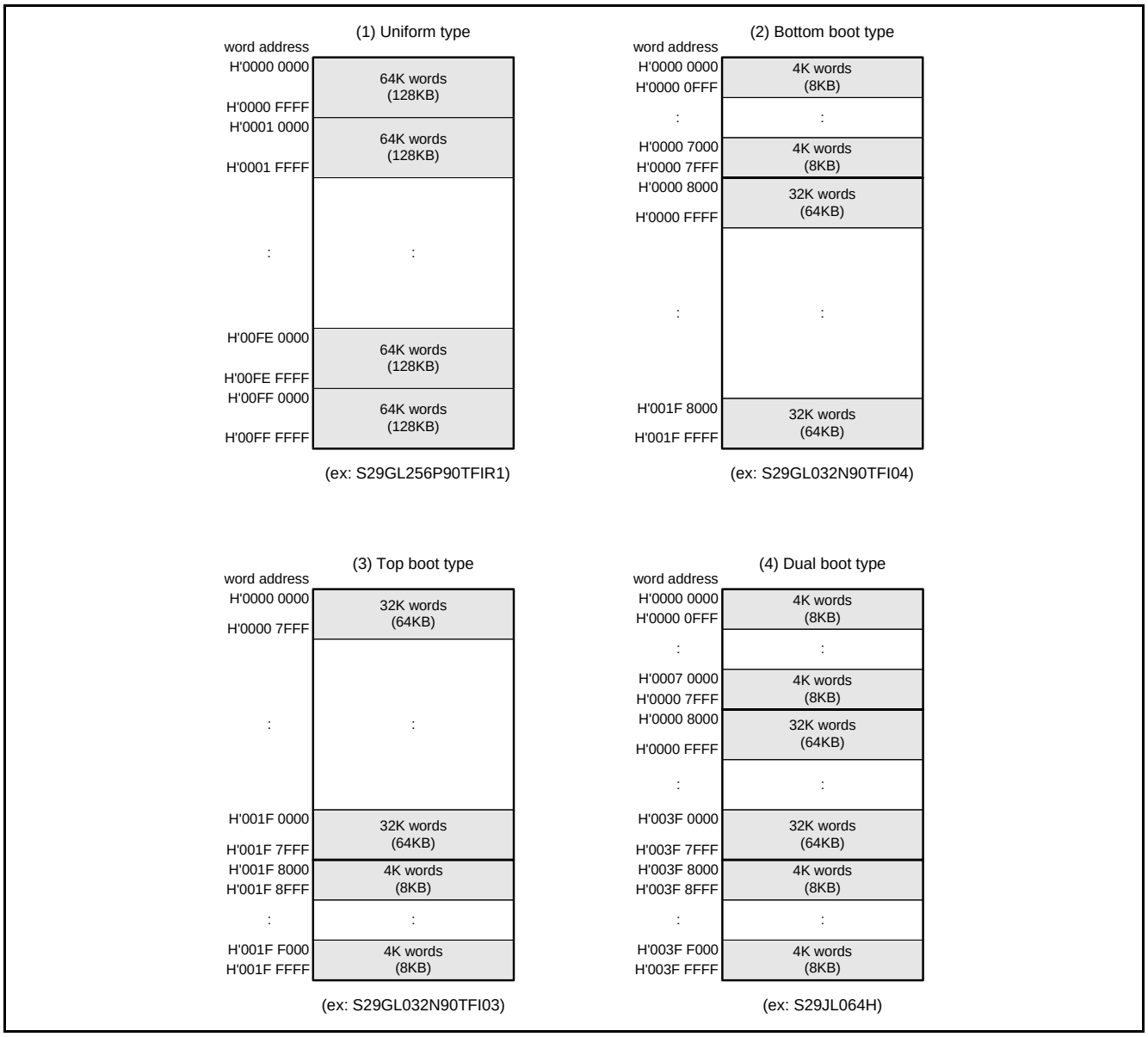


Figure 6.7 Memory Map for Flash Memory Boot Types

6.2.3 Contents of Customization

Table 6.4 lists the Cases for Customization and their Descriptions **(1/2)**.

Table 6.4 Cases for Customization and their Descriptions (1/2)

case	Description
When the boot type is different	The sample code corresponds to erasing and writing of the flash memory with Uniform type. When using the flash memory with Top boot, Bottom boot and Dual boot type, change the macro definition FM_BOOT_TYPE as follows: Top boot type: FM_TOP_BOOT Bottom boot type: FM_BOTTOM_BOOT Dual boot type: FM_DUAL_BOOT (The Uniform type FM_UNIFORM is defined as the initial value.)
Change the boot type to: (1) Bottom boot type	Change the sector definition -FM_B_BOOT_SECTOR_START -FM_B_BOOT_SECTOR_SIZE -FM_B_BOOT_SECTOR_NUM -FM_NORMAL_SECTOR_START -FM_NORMAL_SECTOR_SIZE -FM_NORMAL_SECTOR_NUM Refer to the section 6.2.5 to change the definition. The initial values of the following sector definitions are defined as 0. These are applicable without change. -FM_T_BOOT_SECTOR_START -FM_T_BOOT_SECTOR_SIZE -FM_T_BOOT_SECTOR_NUM
Change the boot type to: (2) Top boot type	Change the sector definition -FM_NORMAL_SECTOR_START -FM_NORMAL_SECTOR_SIZE -FM_NORMAL_SECTOR_NUM -FM_T_BOOT_SECTOR_START -FM_T_BOOT_SECTOR_SIZE -FM_T_BOOT_SECTOR_NUM Refer to the section 6.2.6 to change the definition. The initial values of the following sector definitions are defined as 0. These are applicable without change. -FM_B_BOOT_SECTOR_START -FM_B_BOOT_SECTOR_SIZE -FM_B_BOOT_SECTOR_NUM

Note: The FMTOOL depends on the flash memory specification. Note that the cases listed in Table 6.4 and Table 6.5 may not be applicable for every situation. Check the device data sheet and modify the FMTOOL according to the device specification.

Table 6.5 Necessary Customization and their Descriptions (2/2)

case	Description
Change the boot type to: (3) Dual boot type	Change the sector definition -FM_B_BOOT_SECTOR_START -FM_B_BOOT_SECTOR_SIZE -FM_B_BOOT_SECTOR_NUM -FM_NORMAL_SECTOR_START -FM_NORMAL_SECTOR_SIZE -FM_NORMAL_SECTOR_NUM -FM_T_BOOT_SECTOR_START -FM_T_BOOT_SECTOR_SIZE -FM_T_BOOT_SECTOR_NUM Refer to the section 6.2.7 to change the definition.
When the sector size or the number of sectors are different	Change the sector definition -FM_NORMAL_SECTOR_START -FM_NORMAL_SECTOR_SIZE -FM_NORMAL_SECTOR_NUM Refer to the section 6.2.4 to change the definition. The initial values of the following sector definitions are defined as 0. These are applicable without change. -FM_B_BOOT_SECTOR_START -FM_B_BOOT_SECTOR_SIZE -FM_B_BOOT_SECTOR_NUM -FM_T_BOOT_SECTOR_START -FM_T_BOOT_SECTOR_SIZE -FM_T_BOOT_SECTOR_NUM
When the memory capacity is different	Change the setting value for the macro FM_END_ADDRESS
When the boot type is different from the types listed in Figure 6.7	The operation function of the flash memory needs to be customized. Refer to the sample code for details.
When the command specifications for erasing, writing and reset are different	
When the write command requires the CUI command method	
When the SH7269 and the flash memory are connected by the bus except for 16-bit	

Note: The FMTOOL depends on the flash memory specification. Note that the cases listed in Table 6.4and Table 6.5 may not be applicable for every situation. Check the device data sheet and modify the FMTOOL according to the device specification.

6.2.4 Customizing Uniform Type

Figure 6.8 shows the Method to Customize Sector Size and Number of Sectors with Uniform Type. When using the flash memory with Uniform type which has different sector size and number of sectors from the S29GL256P90TFIR1, change the setting value for macro of the Uniform type sector information. When the different flash memory capacity is provided, change the setting value for macro of the end address.

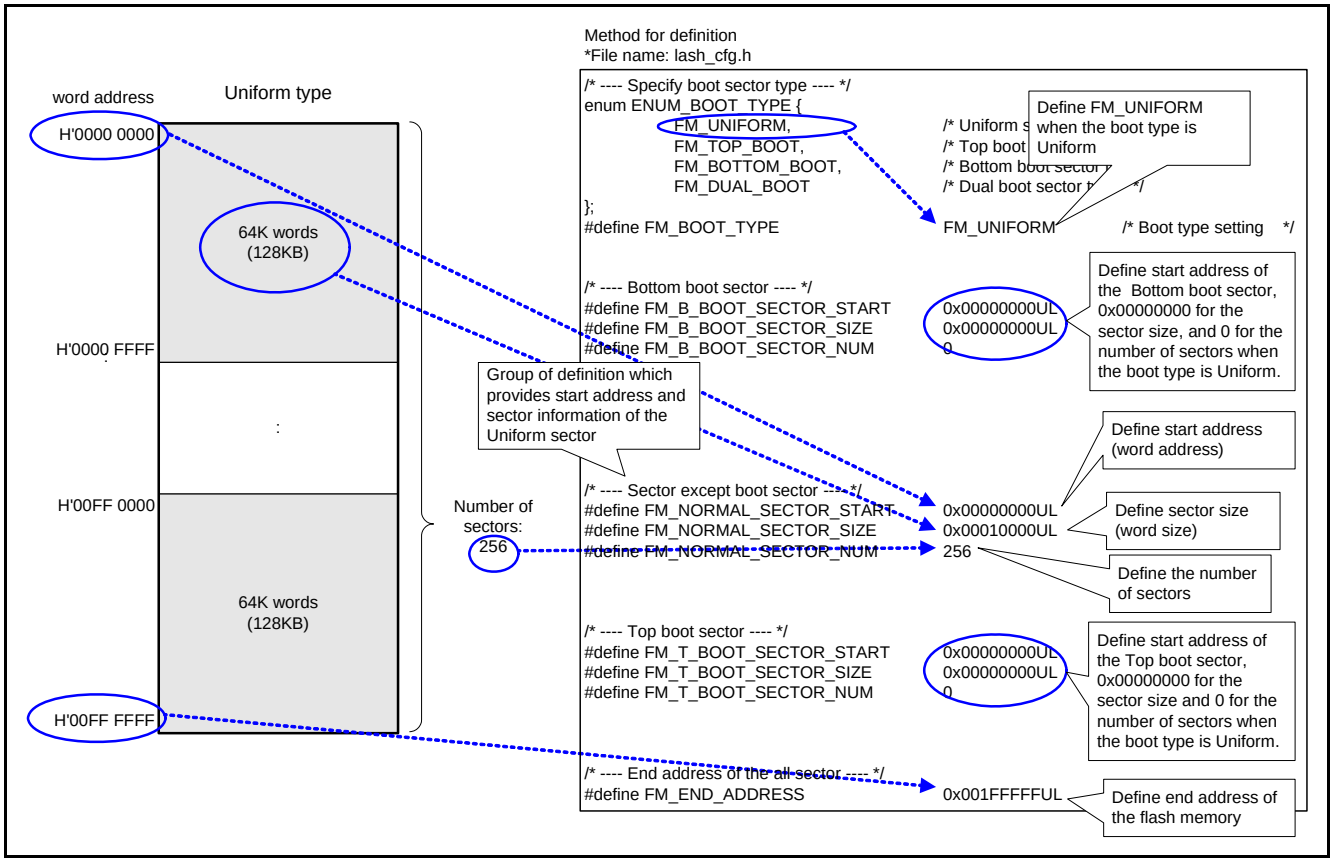


Figure 6.8 Method to Customize Sector Size and Number of Sectors with Uniform Type

6.2.5 Customizing to Bottom Boot Type

Figure 6.9 shows the Method to Customize to Bottom Boot Type. When using the flash memory with Bottom boot type, change the setting value for macro of the sector information which relates to the bottom boot sector area and other areas except for boot sectors. When the different flash memory capacity is provided, change the setting value for macro of the end address.

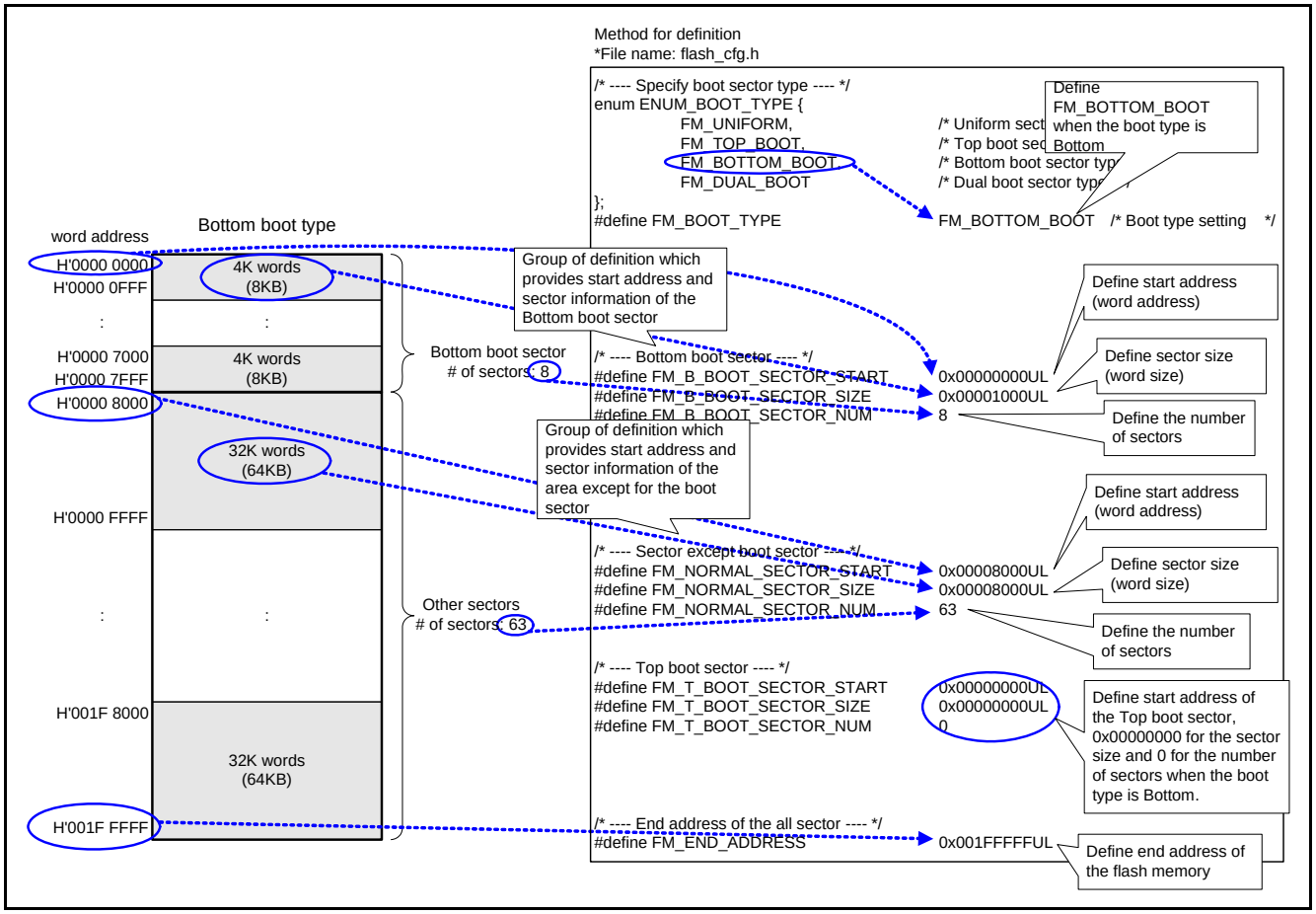


Figure 6.9 Method to Customize to Bottom Boot Type

6.2.6 Customizing to Top Boot Type

Figure 6.10 shows the Method to Customize to Top Boot Type. When using the flash memory with Top boot type, change the setting value for macro of sector information which relates to the top boot sector area and other areas except for the boot sector. When the different flash memory capacity is provided, change the setting value for macro of the end address.

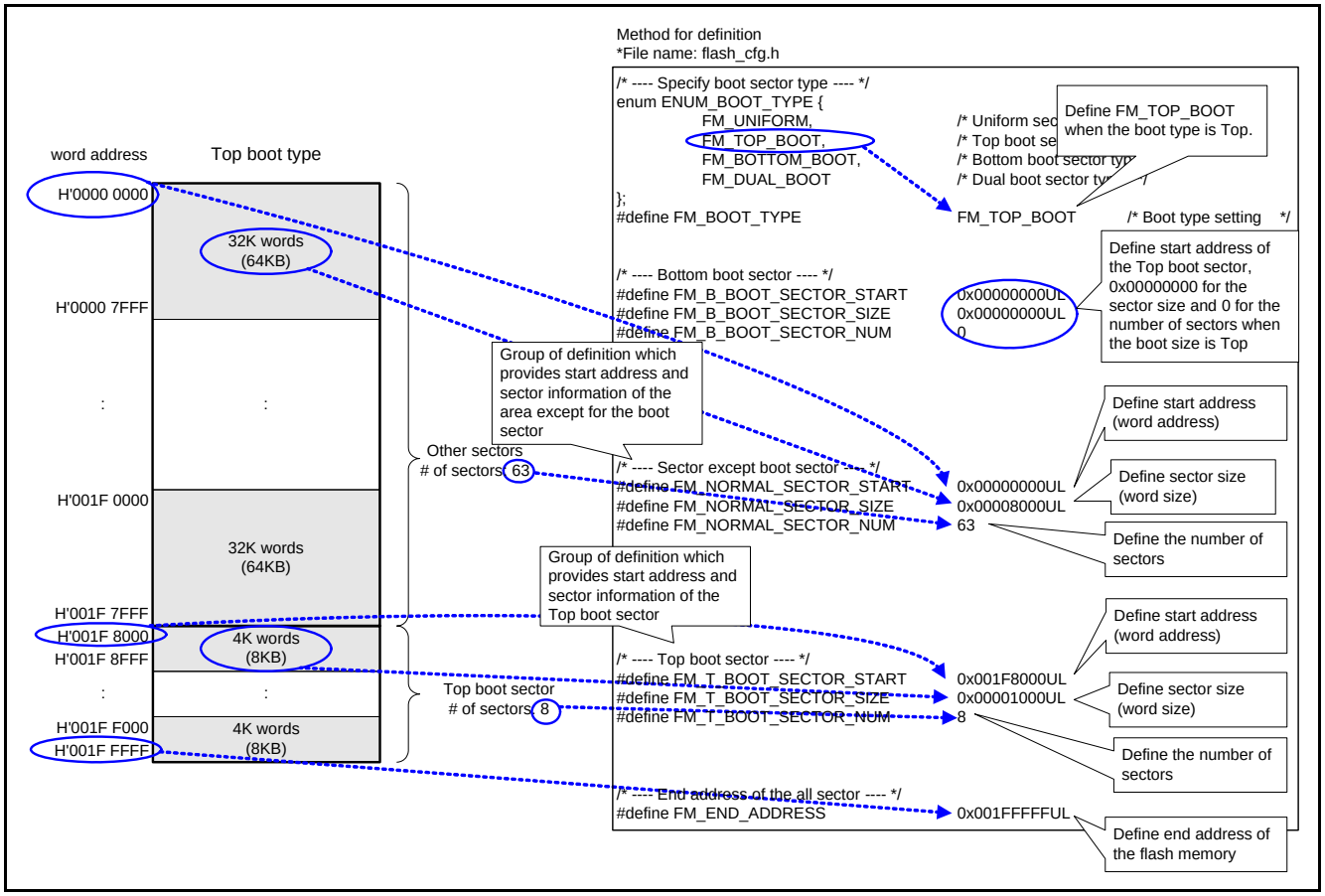


Figure 6.10 Method to Customize to Top Boot Type

6.2.7 Customizing to Dual Boot Type

Figure 6.11 shows the Method to Customize to Dual Boot Type. When using the flash memory with Dual boot type, change the setting value for macro of the sector information which relates to the top boot sector area, the bottom boot sector area and other areas except for the boot sectors. When the different flash memory capacity is provided, change the setting value for macro of the end address

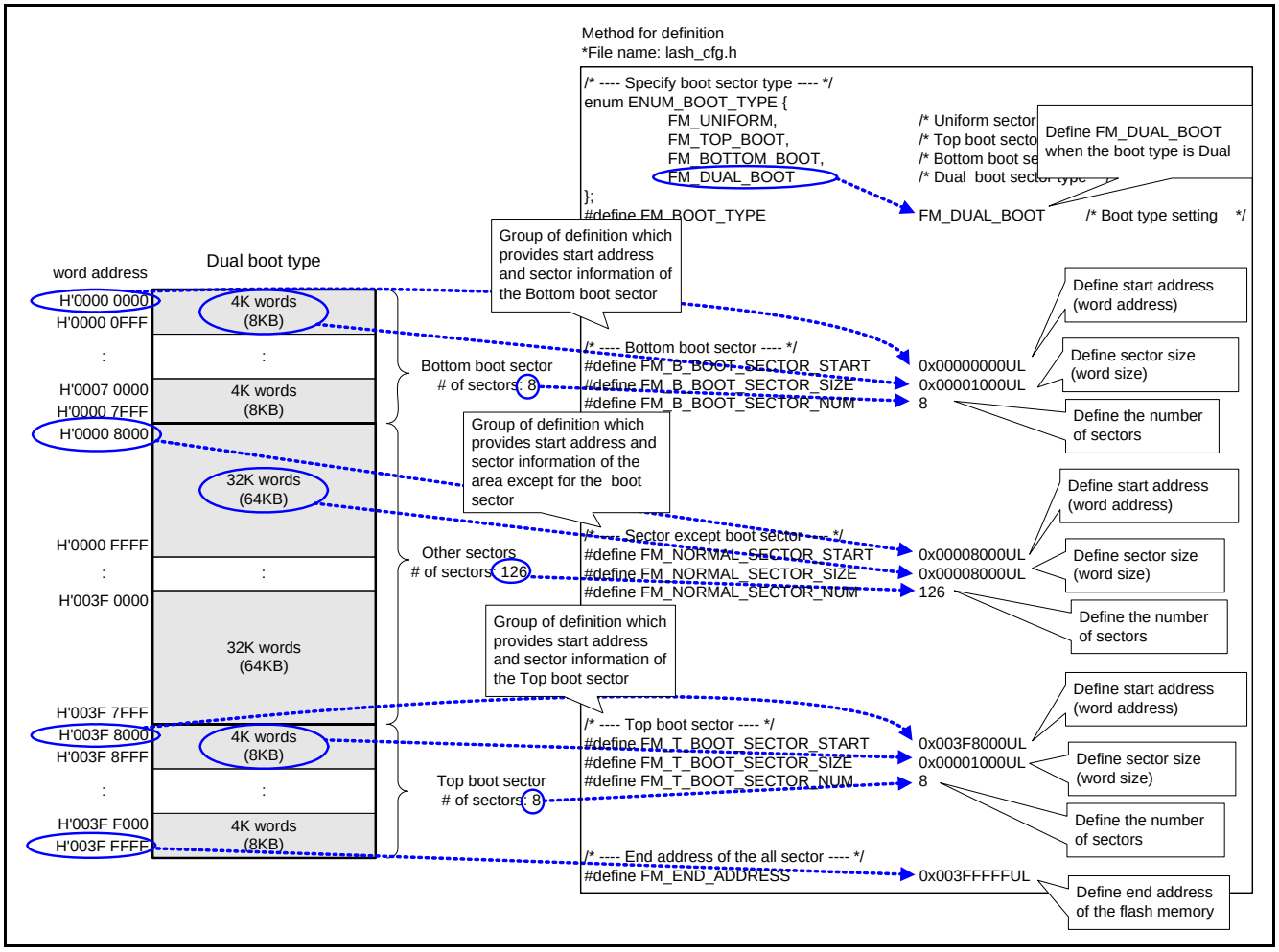


Figure 6.11 Method to Customize to Dual Boot Type

6.2.8 Definitions of Command

Figure 6.12 shows the Definitions of Command. The sample code corresponds to the JEDEC standard compatible command, and defines the command when connects the SH7269 and the flash memory by 16-bit. Refer to the word mode (x16-bit mode) command on the flash memory data sheet to verify that no difference exists in the command contents between the reset command, the sector erase command and program command (write command). Note that the sample code does not correspond to page program (page write).

* File name: flash_cfg.h

```
/* ==== Specify command ==== */
/* ---- Reset command ---- */
#define FM_CMD_RESET                    0x00F0
/* ---- Sector erase command ---- */
#define FM_CMD_S_ERASE_ADDR_1           0x0555      /* Address (First cycle)  */
#define FM_CMD_S_ERASE_ADDR_2           0x02AA
#define FM_CMD_S_ERASE_ADDR_3           0x0555
#define FM_CMD_S_ERASE_ADDR_4           0x0555
#define FM_CMD_S_ERASE_ADDR_5           0x02AA      /* Address (Fifth cycle)  */
#define FM_CMD_S_ERASE_DATA_1           0x00AA
#define FM_CMD_S_ERASE_DATA_2           0x0055
#define FM_CMD_S_ERASE_DATA_3           0x0080      /* Data (Third cycle)     */
#define FM_CMD_S_ERASE_DATA_4           0x00AA      /* Data (Fourth cycle)    */
#define FM_CMD_S_ERASE_DATA_5           0x0055
#define FM_CMD_SECTOR_ERASE             0x0030
/* ---- Single word program command ---- */
#define FM_CMD_PROGRAM_ADDR_1           0x0555
#define FM_CMD_PROGRAM_ADDR_2           0x02AA
#define FM_CMD_PROGRAM_ADDR_3           0x0555
#define FM_CMD_PROGRAM_DATA_1           0x00AA
#define FM_CMD_PROGRAM_DATA_2           0x0055
#define FM_CMD_PROGRAM_DATA_3           0x00A0
```

Define the rest command

Erase sequence
Define the write address

Erase sequence
Define the write data

Define the erase command

Program sequence
Define the write address

Program sequence
Define the write data

Figure 6.12 Definitions of Command

7. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

8. Reference Documents

User's Manual: Hardware
SH7268/SH7269 Group User's Manual: Hardware Rev.1.00
The latest version can be downloaded from the Renesas Electronics website.

Technical Update/Technical News
The latest information can be downloaded from the Renesas Electronics website.

Development Tool Manual
SuperH RISC Engine Family C/C++ Compiler Package V.9.04
C compiler User's Manual Rev.1.01
The latest version can be downloaded from the Renesas Electronics website.

SuperH Family E10A-USB Emulator User's Manual Rev.9.00
The latest version can be downloaded from the Renesas Electronics website.

SuperH Family E10A-USB Emulator Additional Document for User's Manual
Supplementary Information on Using the SH7264, SH7262, SH7266, SH7267, SH7268 and SH7269
The latest version can be downloaded from the Renesas Electronics website.

Website and Support

Renesas Electronics website
http://www.renesas.com

Inquiries
http://www.renesas.com/contact/

REVISION HISTORY	SH7268/SH7269 Group E10A-USB Flash Memory Download Function (Download to the NOR Flash Memory)

Rev.	Date	Description	
		Page	Summary
1.00	Jul. 18, 2012	-	First edition issued

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

1. Handling of Unused Pins
   Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.
   The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.
2. Processing at Power-on
   The state of the product is undefined at the moment when power is supplied.
   The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.
   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.
   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.
3. Prohibition of Access to Reserved Addresses
   Access to reserved addresses is prohibited.
   The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.
4. Clock Signals
   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.
   When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.
5. Differences between Products
   Before changing from one product to another, i.e. to one with a different type number, confirm that the change will not lead to problems.
   The characteristics of MPU/MCU in the same group but having different type numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different type numbers, implement a system-evaluation test for each of the products.

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
3. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.
   Renesas Electronics products are neither intended nor authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems, surgical implantations etc.), or may cause serious property damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
11. This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

RENESAS

SALES OFFICES

Renesas Electronics Corporation

http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 2.0