To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

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

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. 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.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. 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.
5. When exporting the 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. You should not use Renesas Electronics products or the 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. 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.
6. 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.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. 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 categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. 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 an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   "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.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. 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.
9. 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 system manufactured by you.
10. 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.
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.

H8, H8SX Family
Flash Memory Download Program for the E10A-USB Emulator

Introduction

The following function is supported to the E10A-USB emulator for Renesas 32bits CISC microcomputer H8, H8SX microcomputer family.

- Program download function to the external flash memory
- Setting of PC breakpoint to the external flash memory
- Direct rewriting of data of the external flash memory in memory window

The data of the external flash memory can be rewritten by using these functions as well as on-chip flash memory. However, the user needs to prepare a download program to use the download function to the external flash memory. In this Application Note, the sample program of the download program is given, together with customization and downloading procedure using the E10A-USB emulator.

Contents

1. Important Information

Before creating this document, the operation has been confirmed by Renesas. However, this does not mean that the Renesas is responsible for guaranteed operation.

Before reading this Application Note, please refer to the section of Download Function to the Flash Memory Area, in the Debugger Part of the E10A-USB emulator user's manual for H8S, H8SX microcomputer family.

In this Application Note, an example of downloading the user program to the external flash memory connected to the HSB8SX1651 CPU Board (manufactured by Hokuto Denshi, Co., Ltd.) is shown. If the specifications of the user system is different from those of the HSB8SX1651 CPU Board, the initialization , write and erase modules supplied as the same programs require customization. When using the sample program, read this Application Note thoroughly and understand its contents before customization, and confirm its operation.

2. Description of Sample Program

By the sample program of this Application Note, the user program can be downloaded to the external flash memory area using the flash memory download function of the E10A-USB emulator.

This Application Note describes the contents to be modified for customization of the initialization , write and erase modules, which are used for downloading the user program to the external flash memory connected to the microcomputer, to suit the specifications of the user system (connection from with the target microcomputer).

2.1 Operation Confirmation Environment of Sample Program

The operation was confirmed by this sample program under the following environment. If the specifications of the user system is different from following environment, the initialization , write and erase modules supplied as the sample programs require customization.

Table 1 Software Environment

Software name	Specification	Note
Integrated Development Environment High-performance Embedded Workshop(HEW)	Ver.4.00	—
Sample Program(Down load Program) • Source file : fmtool.src • Object file : fmtool.mot	Program for writing to the external flash memory	Called **the write module** hereafter
	Program for Erasing to the external flash memory	Called **the erase module** hereafter
	Program for Initializing device for accessing the external flash memory	Called **the initialization module** hereafter
User Program	Program for downloading to the external flash memory	

Note: The sample program is for word-mode connection to the TOSHIBA 16Mbit flash memory. The source file name: fmtool.src is not cared about by an arbitrary name.

Table 2 Hardware Environment

Hardware name		Specification	Note
Host PC		DOS/V PC	
E10A-USB Emulator		HS0005KCU01H	
User System	CPU Board “HSB8SX1651” Hokuto Densi Co., Ltd.	Renesas 32bit CISC microcomputer “H8SX1651” -> On-chip RAM 40K -> Advanced CPU mode -> Connect CS0 area to Flash memory	**The RAM area in the user system is needed. The on-chip RAM area is used in this example of this Application Note.**
	External Flash memory	TOSHIBA 16Mbit Flash Memory “TC58FVT160AFT-70” -> Word mode(16-bit bus width)	

3. Contents of the Sample Program(Download Program)

The sample program (fmtool.src) described in section 2, Description of Sample Program, needs to be changed in accordance with the specifications of the user system.

3.1 Structure of Sample Program

The contents of sample program that can be downloaded from the web are shown below. (Unzip the sample program after downloading as it is compressed.)

```
FM_TOSHIBA folder                    <------- Folder name
    | - - FMTOOL.HWS                 <------- HEW workspace
    |
    | - - FMTOOL folder
            | - - FMTOOL.hwp
            | - - FMTOOL.src.        <------- Source file
            | - - FMTOOL.tps
            |
            | - - Release folder
                    | - - FMTOOL.abs
                    | - - FMTOOL.hlk
                    | - - FMTOOL.mot <------- Execution file
                    | - - FMTOOL.h8a
                    | - - Release.hdp
```

Figure 1 Structure of Sample Program

- [Folder name]:
  — Folder name is the same as the zip file that is available on the web page.
- [HEW workspace]:
  — This is a set of workspace for HEW. This workspace is opened on the HEW when a sample program is edited.
- [Source file]:
  — This is am assembly-language source file of the initialization , write and erase modules to be used for the download function to the flash memory of E10A-USB emulator.
- [Execution file]:
  — This file is specified by “the download function to the flash memory area” which is a function of the E10A-USB emulator.

3.2 Interface with Initialization , Write and Erase Modules and Emulator Firmware

The initialization , write and erase modules must be branched from the emulator firmware. To branch from the emulator firmware to be the initialization , write and erase modules, or to return from the initialization , write and erase modules, or to return from the initialization , write and erase modules to emulator firmware, the following conditions must be observed.

(1) Describe all the initialization , write and erase modules with the assembly language.
(2) The initialization , write and erase modules size must be within 4K Bytes. (Include work area and stack area).
(3) Save and return all the general register values and control register values before and after calling the initialization , write or erase module.
(4) Return the initialization , write or erase module to the calling source after processing.
(5) The initialization , write and erase module must be an Motorola-type file. The module interface must be as follows to pass correctly the information that is required for flash memory accessing.

Table 3 Write Module Interface

Module name	Parameter	Return Address	Specification
Write Module	• ER0(L) : Write Address • ER1(L) : Write Data	• ER0(L) : End Code Normal End=0 Abnormal End=1	The write data ER1 is written at write address ER0 and it returns to the calling function. The size of the write data is a size set with [Bus width] in Table 5.

Note: The (L) means the long-word size.

Table 4 Erase Module Interface

Module name	Parameter	Return Address	Specification
Erase Module	• ER0(L) : Erase Block Address	• ER0(L) : End Code Normal End=0 Abnormal End=1	The erase block address ER0 is erased, the result is written in ER0, and it returns to the calling function. This function is a block erasing. This function doesn't operate normally at the chip erasing.

Note: The (L) means the long-word size.

Table 5 Initialization Module Interface

Module name	Parameter	Return Address	Specification
Initialization Module	-	• ER0(L) : End Code Normal End=0 Abnormal End=1	The initialization module is used to set the device that is required for accessing the external flash memory. Store the result in ER0(L) and it returns to the calling function. If the device can access to the external flash memory in initial setting ,the initialization module is unnecessary.

Note: The (L) means the long-word size.

4. Creating the Execution File

After modifying the fmtool.src file as required by referring to section 3, Contents of the Sample Program (Download Program), create the execution file by the following procedure.

Press the [All Build] button of the HEW to execute a build. This performs assembly and linkage and creates an execution file.

When the build is successfully completed, the execution file fmtool.mot is created under the folder with the same name as the build configuration (usually in folder “release”)

5. Procedure for the Downloading the User Program

This section describes the procedure for the downloading the user program to the flash memory using file fmtool.mot which is created in section 4, Creating the Execution File. An Example using the HSB8SX1651 CPU Board (manufactured by Hokuto Denshi, Co., Ltd.) is shown below.

5.1 Preparation for Download Environment

(1) Connect the E10A-USB emulator connected to the host computer.
(2) Connect the user system interface cable to the connector of the E10A-USB emulator.
(3) Connect the user system interface cable to the connector in the user system.
(4) Select [Renesas] -> [High-performance Embedded Workshop] -> [High-performance Embedded Workshop] from [Program] in the [Start] menu.
(5) The [Welcome!] dialog box is displayed.

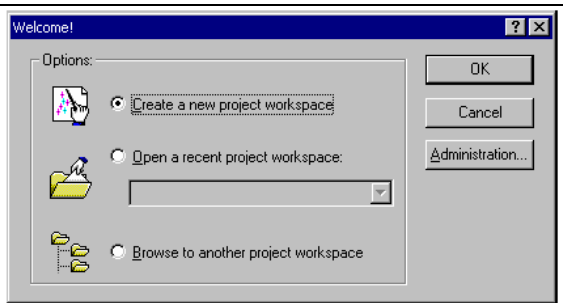

Figure 2 [Welcome!] Dialog Box

(6) The [Select Emulator mode] dialog is shown in Figure 3 displayed. Select “H8SX/1651” CPU used from the drop-down list box, and press the [OK] button.

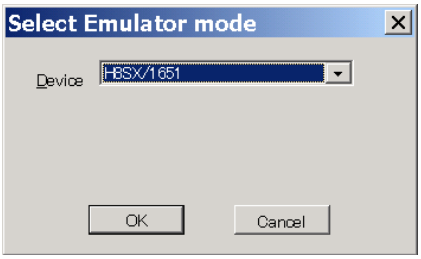

Figure 3 [Select Emulator mode] Dialog Box

(7) The [Flash memory setting] dialog box shown in Figure 7 is displayed. Set the Execution File name and the external flash memory information. For details, please refer to section 5.2, Setting the [Flash memory setting] Dialog Box.
(8) The dialog box is displayed as shown in Figure 4. Power on the user system. Input the reset signal from the user system, and click the [OK] button.

Figure 4 Dialog Box of the Power –on Request Message

(9) The [Connecting] dialog box is displayed and emulator connection is started. When “Connected” is displayed in the [Output] window of the High-performance Embedded Workshop, the emulator initiation is completed.

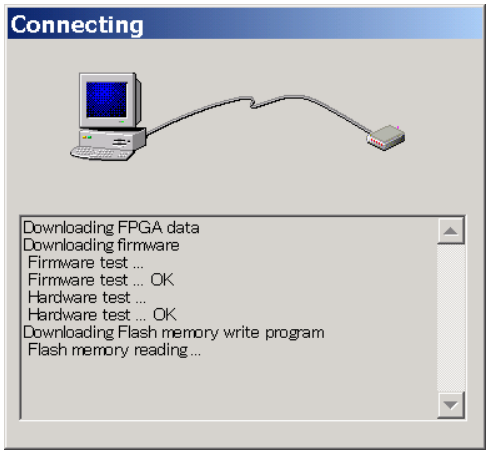

Figure 5 [Connecting] Dialog Box

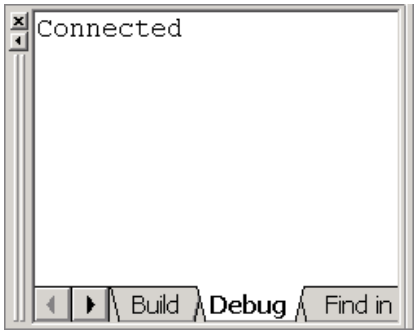

Figure6 [Output] Window

5.2 Setting the [Flash memory setting] Dialog Box

For downloading to the flash memory, set the items on the [Flash memory setting] dialog box. Table 4 shows the items for the [Flash memory setting] dialog box.

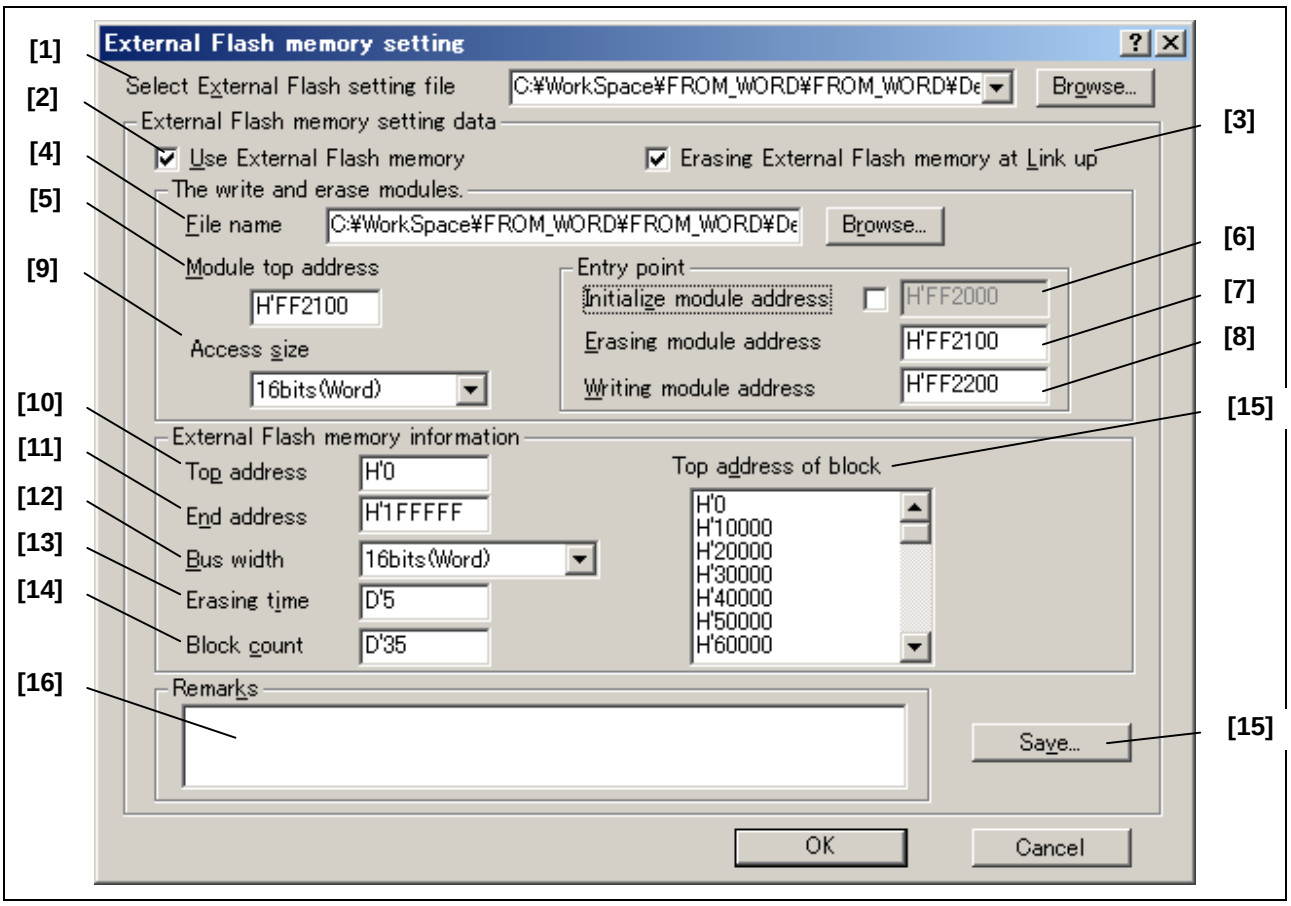

Figure 7 [Flash memory setting] Dialog Box

The sample setting for HSB8SX1651 CPU Board is given below.

[1] [Select External Flash setting file]	: Select the file that was saved in [17]
[2] [Use External Flash Memory]	: This box is checked to downloading to the flash memory.
[3] [Erasing flash memory at Link up]	: When the data of the flash memory is erased when the emulator starts, this box is checked. It doesn't check it in this example.
[4] [File name]	: Set the execution file(download program) name.
[5] [Module top address]	: ”H’FF2100”
[6] [Initialize top address]	: The initialization module is used to set the device that is required for accessing the external flash memory. The initialization module is not checked in this example because of unnecessary.
[7] [Erasing module address]	: ”H’FF2100”
[8] [Writing module address]	: ”H’FF2200”
[9] [Access size]	: ”16bits(words)”
[10] [Top address]	: ”H’0”
[11] [End address]	: ”H’1FFFFF”
[12] [Bus width]	: ”16bits(words)”
[13] [Erasing time]	: ”D’5”
[14] [Block count]	: ”D’35”
[15][Top address of block]	: ”H’0”、”H’10000”、”H’20000”…….”H’1FC000”
[16][Remarks]	: Use for writing a text.
[17][Save]	: It is used to save setting.

Note1: The on-chip RAM area of microcomputer H8SX1651 is used. The on-chip RAM area of microcomputer H8SX1651 is "H' FF2000- H'FFBFFF”.

Note2: Please refer to the data sheet of TOSHIBA 16M(4M x8/2M x16) flash memory "TC58FVT160AFT-70" for the number of blocks and the block address (There are 35-block number of blocks). The block counts and block address to be set differs depending on the flash memory. For details, please refer to the data sheet for each flash memory.

Table 4 Setting the [Flash memory setting] Dialog Box

No	Item	Contents
[1]	[Select External Flash setting file]	Specify the data file (*.EFF) for setting the external flash memory. If not specified, select the setting of [recent setting data: data previously changed] in the combo box. To make a new setting, select [initialize] and input data.
[2]	[Use Flash Memory] Check Box	Sets Enable for flash memory downloading. At Enable, when the flash memory is downloaded. • Check : Enable(Download to the flash memory) • No Check : Disable(Not Downloaded to the flash memory)
[3]	[Erasing flash memory at Link up] Check Box	Sets Enable for erasing before the flash memory is programmed. At Enable, the erase module is called before calling the write module. • Check : Enable(Erase the flash memory) • No Check : Disable(Not erase the flash memory)
[4]	[File name] Edit Box	Sets the initialization/write /erase module name. The file that has been set is loaded to the RAM area before loading to the flash memory.
[5]	[Module top address] Edit Box	Sets the start address of the stack area for initialization , write and erase module. E10A-emulator firmware preserves data from the address set here for 4K Bytes.
[6]	[Initialize module address] Edit Box	Entry address of the initialization module. The initialization module is used to set the device that is required for accessing the external flash memory. If not specified, disable entry. Checked: Enables entry. Not checked: Disables entry (default).
[7]	[Erasing module address] Edit Box	Inputs the calling destination address of the erase module.
[8]	[Writing module address] Edit Box	Inputs the calling destination address of the write module.
[9]	[Access size] List Box	Select the access size of the RAM area where the initialization/write /erase module is loaded. 8bits(Byte) : Byte, 16bits(Word) : Word, 32bits(Long) : Longword
[10]	[Top address] Edit Box	Inputs the start address of the flash memory. The capacity that can be set is 4M Byte or less.
[11]	[End address] Edit Box	Inputs the End address of the flash memory. The capacity that can be set is 4M Byte or less.
[12]	[Bus width] List Box	Sets the bus width of the flash memory. Please give the specified access size to the width of the bus of the flash memory of the system additionally. Please make the write module for the access size set here additionally. 8bits(Byte) : Byte, 16bits(Word) : Word, 32bits(Long) : Longword
[13]	[Erasing time] Edit Box	Sets the timeout value at flash memory erasing. The wait time period may be extended according to the set value. Therefore, it is recommended to input the minimum value by considering the erasing time of the flash memory. The radix for the input value is decimal. It becomes hexadecimal by adding H'.
[14]	[Block count] Edit Box	Sets the Block count at the flash memory erasing. A maximum of 99 Blocks can be input the Block count. The radix for the input value is decimal. It becomes hexadecimal by adding H'.

[15]	[Top address of block] Edit Box	Sets the Start address (physical memory address) of each block in the flash memory. Please input the number of all blocks when there is an unused block Please insert changing line between block first addresses.
[16]	[Remarks] Edit Box	Use for writing a text. Contents of data that has been set can be entered. If not specified, setting is not needed.
[17]	[Save] button	Save the data file (*.EFF) for setting the external flash memory.

Note1: The content input to the [Flash memory setting] dialog box is preserved, and when starting, the previous setting will be displayed next time.

Note2: Time hangs in the connection to load the data of the memory area when [Erasing flash memory at Link up] check box is made no check. The data of the flash memory is erased when [Erasing flash memory at Link up] check box is checked, However connect time shortens.

5.3 Downloading the User Program

When the [Configuration] dialog box setting has been completed. Download the user program which you wants to download to the flash memory using the HEW program loading function shown in Figure 8 and 9. To download a program, select the load module from [Debug] -> [Download] or select [Download] from the popup menu opened by clicking the right-hand mouse button on the load module in [Download modules] of the [Workspace] window.

Note: Before downloading a program, it must be registered to the High-performance Embedded Workshop as a load module. For registration, refer to the section of setting at emulator activation of the E10A-USB emulator user's manual.

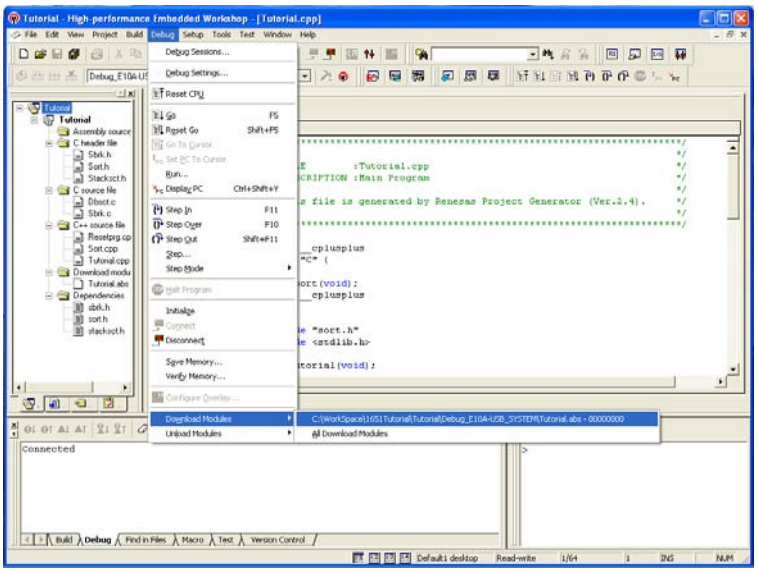

Figure 8 Downloading the User Program (menu [Debug] -> [Download])

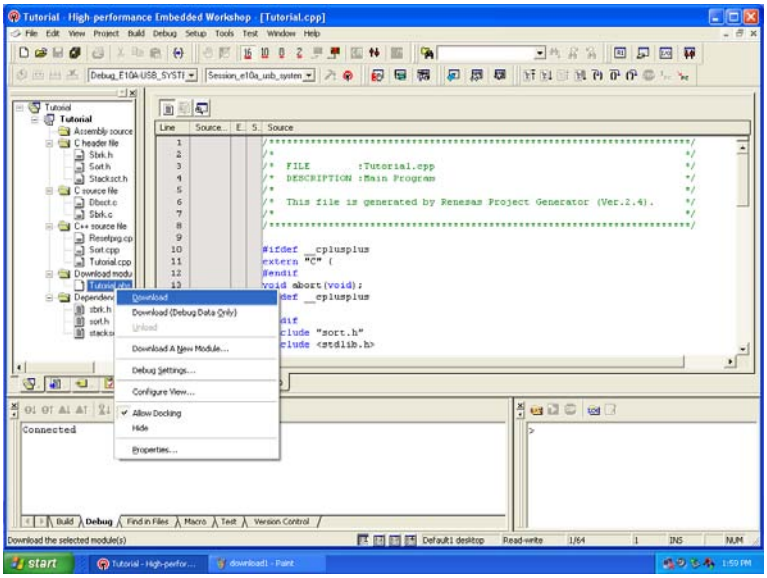

Figure 9 Downloading the User Program (popup menu item)

6. Modifying the Sample Program

The sample program given below is for word-mode connection to the flash memory from TOSHIBA. This section explains the modification of the initialization , write and erase modules when downloading the user program to the flash memory that is not verified using the E10A-USB emulator.

6.1 Verification of Command Method of Flash Memory

In flash memory command, basically, there are two types of command method.

Table 5 Flash Memory Command

Command	Method	Ex. Program	Manufacture
JEDEC Standard Command Compatibility Method	Issue a command to the predefined address such as 0x555, 0x2AA	1. 555H/AAH 2. 2AAH/55H 3. 555H/A0H 4. PA/PD	Spansion Toshiba NEC Etc.
CUI Command Method CUI : Command User Interface	Issue the write command (0x40) or erase command (0x20) via the CUI.	1. XXXH/40H 2. PA/PD	Intel Sharp Etc.

Please check the writing method by referring to definition of commands described the datasheet of the flash memory to be used. When the JEDEC standard command compatibility method is used, the flash memory sample program from this Application Note can be reused. The JEDEC standard commands of the flash memory in this Application Note are shown in Table 6.

When the CUI command method is used, the flash memory sample program can not be used. Accordingly, the user needs to write a new program for downloading.

Note the flash memory downloading function of the E10A-USB does not support the page-write mode.

Table 6 JEDEC Standard Command (Sample Program: TOSHIBA TC58FV160AFT-70 [Word mode])

Command (mode)	1st Bus Cycle		2nd Bus Cycle		3rd Bus Cycle		4th Bus Cycle		5th Bus Cycle		6th Bus Cycle	
	Addr.	Data	Addr.	Data	Addr.	Data	Addr.	Data	Addr.	Data	Addr.	Data
Reset	XXXh	F0h										
Write = Program (Word)	555h *1	AAh	2AAh *1	55h	555h *1	A0h	PA	PD				
Block Erase (Word)	555h *1	AAh	2AAh *1	55h	555h *1	80h	555h *1	AAh	2AAh *1	55h	BA	30H

*1 : Please add the top address of the flash memory to this address, except when the first address of the flash memory is H'0.

Example:
The top address of the flash memory: H'200000
Addr: H'555
Please change Addr from H'555 to H'200555(H'200000+H'555=H'200555).

6.2 Initialization module

The initialization module is used to set the device that is required for accessing the external flash memory.
For this case, the setting such as the bus controllers and the pin function controller is necessary, and make the initialization module referring to the hardware manual of the device.
If the device can access to the external flash memory in initial setting ,the initialization module is unnecessary.

6.3 Customizing the Source Program

This section describes how to customize the individual program according to JEDEC standard command method. Please check the datasheet the flash memory to be used, and change the parts of the program, which are enclosed in circle. The sample program given below is for word-mode connection (when one flash memory unit is used) to the flash memory from TOSHIBA. This sample program is not supported Byte mode (the bus width of 8bits) and Long-word mode (the bus width of 32bits). The user needs to prepare a download program to support Byte mode and Long-word mode.

JEDEC Standard Command Compatibility Method (File name: fmtool.src)

```
;+=============================================================================+
;|                                                                             |
;|       Flash memory tool program sample                                      |
;|               H8SX Family Flash memory load is supported                    |
;|               Copyright (C) 2006 Renesas Technology Corp. All rights reserved.|
;|               Licensed Material of Renesas Technology Corp.                 |
;|                                                                             |
;|                                                   Erasing flas              |
;|                                                   Writing flas              |
;|                                                                             |
;|               Target flash memory : TOSHIBA TC58FV16(                       |
;|                                                                             |
;+=============================================================================+
;
;+=============================================
;|    EQU
;+=============================================
;
O_FMInit            .equ    H'00FF2000
O_FMErase           .equ    H'00FF2100
O_FMWrite           .equ    H'00FF2200
;
FM_CMD_RESET        .equ    H'00F0
FM_CMD_ERASE        .equ    H'0030
FM_CHK_DQ7  .equ    H'0080
FM_CHK_DQ5  .equ    H'0020
;
FM_OK               .equ    H'00000000
FM_NG               .equ    H'FFFFFFFF
;
        .align  4
;
;=============================================
```

Start address of the initialization module
The RAM area in the user system is used.
This start address is also used in 5.2, Setting the [Flash memory setting] Dialog Box.
When initialization module is unnecessary, this description is not needed.

Start address of the erase module
The RAM area in the user system is used.
This start address is also used in 5.2, Setting the [Flash memory setting] Dialog Box.

Start address of the write module
The RAM area in the user system is used
This start address is also used in 5.2, Setting the [Flash memory setting] Dialog Box.

Reset command

Block Erase command

Status Check flag
Used for checking whether or not the command processing is correctly completed
DQ7 (data polling flag)
DQ5 (timing limit elapsed flag)

```
;
; NAME = FmEraseWord;
; FUNC = The routine of erasing flash memory(Bus width is 16 bits).
; NOTE = NEW;
; HIST = 2006.01.10;
; INPU = ER0.L  = Block address;
; OUTP = ER0.L  = Status;
;
;=====================================================================
;
        .section  fm_erase,CODE,LOCATE=O_FMErase
;
FmEraseWord:
        mov.l   er1,@-er7                       ;
        mov.l   er2,@-er7                       ;
        mov.l   er3,@-er7                       ;
        mov.l   er4,@-er7                       ;
;
;>>> Clear all status
        mov.w   #FM_CMD_RESET,r2                ; C
        mov.w   r2,@er0                         ;
        nop                                     ;
;
;>>> Erase command sequence
        mov.l   #H'00000AAA,er1                 ; ER1 <- command address
        mov.w   #H'00AA,r2                      ; R2  <- command data
        mov.w   r2,@er1                         ; 1st bus cycle
;
        mov.l   #H'00000554,er1                 ; ER1 <- command address
        mov.w   #H'0055,r2                      ; R2  <- command data
        mov.w   r2,@er1                         ; 2nd bus cycle
;
        mov.l   #H'00000AAA,er1                 ; ER1 <- command address
        mov.w   #H'0080,r2                      ; R2  <- command data
        mov.w   r2,@er1                         ; 3rd bus cycle
;
        mov.l   #H'00000AAA,er1                 ; ER1 <- command address
        mov.w   #H'00AA,r2                      ; R2  <- command data
        mov.w   r2,@er1                         ; 4th bus cycle
;
        mov.l   #H'00000554,er1                 ; ER1 <- command address
        mov.w   #H'0055,r2                      ; R2  <- command data
        mov.w   r2,@er1                         ; 5th bus cycle
;
        mov.w   #FM_CMD_ERASE,r2                ; Write the erase command to flash memory
        mov.w   r2,@er0                         ; 6th bus cycle
;
FmEraseWord_Loop:
;
;>>> Data polling sequence
        mov.w   @er0,r2                         ; Read status
        mov.w   r2,r3                           ;
        mov.w   #FM_CHK_DQ7,r4                  ;
        and.w   r4,r2                           ;
```

Please add the top address of the flash memory to this address, except when the first address of the flash memory is H'0.

```
        cmp.w   r4,r2                           ; if status.DQ7 == 1 (0 = Busy / 1 = End)
        beq     FmEraseWord_Loop_Next           ; then exit
        nop                                     ;
;
        mov.w   r3,r2                           ;
        mov.w   #FM_CHK_DQ5,r4                  ;
        and.w   r4,r2                           ;
        cmp.w   r4,r2                           ; if status.DQ5 == 0 (0 = OK / 1 = Fail)
        bne     FmEraseWord_Loop                ; then loop
;
;>>> Clear all status
        mov.w   #FM_CMD_RESET,r2                ; Clear the status of flash memory
        mov.w   r2,@er0                         ;
        nop                                     ;
;
;>>> Return value
        mov.l   #FM_NG,er0                      ;
        bra     FmEraseWord_End                 ; Error end
        nop                                     ;
;
FmEraseWord_Loop_Next:
;
;>>> Clear all status
        mov.w   #FM_CMD_RESET,r2                ; Clear the status of flash memory
        mov.w   r2,@er0                         ;
        nop                                     ;
;
;>>> Return value
        mov.l   #FM_OK,er0                      ; Set OK
                                                ;
FmEraseWord_End:
        mov.l   @er7+,er4                       ;
        mov.l   @er7+,er3                       ;
        mov.l   @er7+,er2                       ;
        mov.l   @er7+,er1                       ;
        rts                                     ;
        nop                                     ;
;
;
;
;=====================================================================
;
; NAME = FmWriteWord
; FUNC = The routine of writing data to flash memory
; NOTE = NEW;
; HIST = 2006.01.10;
; INPU = ER0.L  = Write address;
;             ER1.L = Write data;
; OUTP = ER0.L  = Status;
;
;=====================================================================
;
        .section  fm_Write,CODE,LOCATE=O_FMWrite
;
```

```
FmWriteWord:
        mov.l   er1,@-er7                       ;
        mov.l   er2,@-er7                       ;
        mov.l   er3,@-er7                       ;
        mov.l   er4,@-er7                       ;
        mov.l   er5,@-er7                       ;
        mov.l   er6,@-er7                       ;
        mov.l   er0,@-er7                       ;
;
;>>> Clear all status
        mov.w   #FM_CMD_RESET,r2                ; CPlease add the top address of the flash memory to this address,
        mov.w   r2,@er0                         ; except when the first address of the flash memory is H'0.
        nop                                     ;
;
;>>> Write command sequence
        mov.l   #H'00000AAA,er3                 ; ER3 <- command address
        mov.w   #H'00AA,r2                      ; R2  <- command data
        mov.w   r2,@er3                         ; 1st bus cycle
;
        mov.l   #H'00000554,er3                 ; ER3 <- command address
        mov.w   #H'0055,r2                      ; R2  <- command data
        mov.w   r2,@er3                         ; 2nd bus cycle
;
        mov.l   #H'00000AAA,er3                 ; ER3 <- command address
        mov.w   #H'00A0,r2                      ; R2  <- command data
        mov.w   r2,@er3                         ; 3rd bus cycle
;
        mov.w   r1,@er0                         ; Write the data to flash memory
;
FmWriteWord_Loop:
;
;>>> Data polling sequence
        mov.w   @er0,r3                         ; Read status
        mov.w   r1,r5                           ;
        mov.w   r3,r4                           ;
        mov.w   #FM_CHK_DQ7,r6                  ;
        and.w   r6,r3                           ;
        and.w   r6,r5                           ;
        cmp.w   r3,r5                           ; if status.DQ7 == write data (No = Busy / Yes = End)
        beq     FmWriteWord_Loop_Next           ; then exit
        nop                                     ;
;
        mov.w   r4,r3                           ;
        mov.w   #FM_CHK_DQ5,r6                  ;
        and.w   r6,r3                           ;
        cmp.w   r6,r3                           ; if status.DQ5 == 0 (0 = OK / 1 = Fail)
        bne     FmWriteWord_Loop                ; then loop
;
;>>> Clear all status
        mov.w   #FM_CMD_RESET,r2                ; Clear the status of flash memory
        mov.w   r2,@er0                         ;
        nop                                     ;
;
;>>> Return value
```

```
        mov.l   @er7+,er0                       ;
        mov.l   #FM_NG,er0                      ; Set the NG code
        bra     FmWriteWord_End                 ; Error end
        nop                                     ;
;
FmWriteWord_Loop_Next:
;
;>>> Verify check
        extu.w  r1                              ; Verify check
        mov.w   @er0,r3                         ;
        extu.w  r3                              ;
        cmp.w   r1,r3                           ; if read data == write data
        beq     CheckVeriW_OK                   ; then OK
;
        bra     CheckVeriW_NG                   ; else NG
        nop                                     ;
;
CheckVeriW_NG:
;
;>>> Clear all status
        mov.w   #FM_CMD_RESET,r2                ; Clear the status of flash memory
        mov.w   r2,@er0                         ;
        nop                                     ;
;
;>>> Return value
        mov.l   @er7+,er0                       ;
        mov.l   #FM_NG,er0                      ; Set the NG code
        bra     FmWriteWord_End                 ; Error end
;
CheckVeriW_OK:
;
;>>> Clear all status
        mov.w   #FM_CMD_RESET,r2                ; Clear the status of flash memory
        mov.w   r2,@er0                         ;
        nop                                     ;
;
;>>> Return value
        mov.l   @er7+,er0                       ;
        mov.l   #FM_OK,er0                      ; Set the OK code
;
FmWriteWord_End:
        mov.l   @er7+,er6                       ;
        mov.l   @er7+,er5                       ;
        mov.l   @er7+,er4                       ;
        mov.l   @er7+,er3                       ;
        mov.l   @er7+,er2                       ;
        mov.l   @er7+,er1                       ;
        rts                                     ;
        nop                                     ;
;
        .end
```

Revision Record

Rev.	Date	Description	
		Page	Summary
1.00	Mar.16.06	—	First edition issued
1.01	Apr.20.09	—	The initialization module has been added.

Keep safety first in your circuit designs!

1. Renesas Technology Corp. puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage.
   Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of nonflammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corp. product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corp. or a third party.
2. Renesas Technology Corp. assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corp. without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor for the latest product information before purchasing a product listed herein.
   The information described here may contain technical inaccuracies or typographical errors.
   Renesas Technology Corp. assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.
   Please also pay attention to information published by Renesas Technology Corp. by various means, including the Renesas Technology Corp. Semiconductor home page (http://www.renesas.com).
4. When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corp. assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corp. semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
6. The prior written approval of Renesas Technology Corp. is necessary to reprint or reproduce in whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.
   Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
8. Please contact Renesas Technology Corp. for further details on these materials or the products contained therein.