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.

SH7206 Group

Example of Program Transfer to On-Chip RAM Using the DMAC

Introduction

This application note describes an example of transferring a program stored in an external ROM to the on-chip RAM of the SH7206.

Target Device

SH7206

Contents

1. Overview

1.1 Specifications

- The direct memory access controller (DMAC) is activated to transfer a program in an external ROM to the on-chip RAM, and the program in the on-chip RAM is executed.

1.2 MCU Functions Used

- Direct memory access controller (DMAC channel 0)

1.3 Conditions for Application

- MCU: SH7206 (R5S72060)
- Operating frequency: Internal clock: 200 MHz
  Bus clock: 66.67 MHz
  Peripheral clock: 33.33 MHz
- C compiler: SuperH RISC Engine Family C/C++ Compiler Package: version 9.00
  (from Renesas Technology Corp.)
- Compiler options: Default setting of HEW (-cpu = sh2a -debug –gbr = auto -global_volatile = 0
  -opt_range = all -infinite_loop = 0 -del_vacant_loop = 0
  -struct_alloc = 1)

1.4 Related Application Notes

- The operation of the sample program in this application note was confirmed with the configuration specified in the application note "Example of SH7206 Initial Configuration". Please refer to that note in combination with this one.
- Details on the DMAC are described in the SH7206 application note "Example-of Memory Transfer by the DMAC".

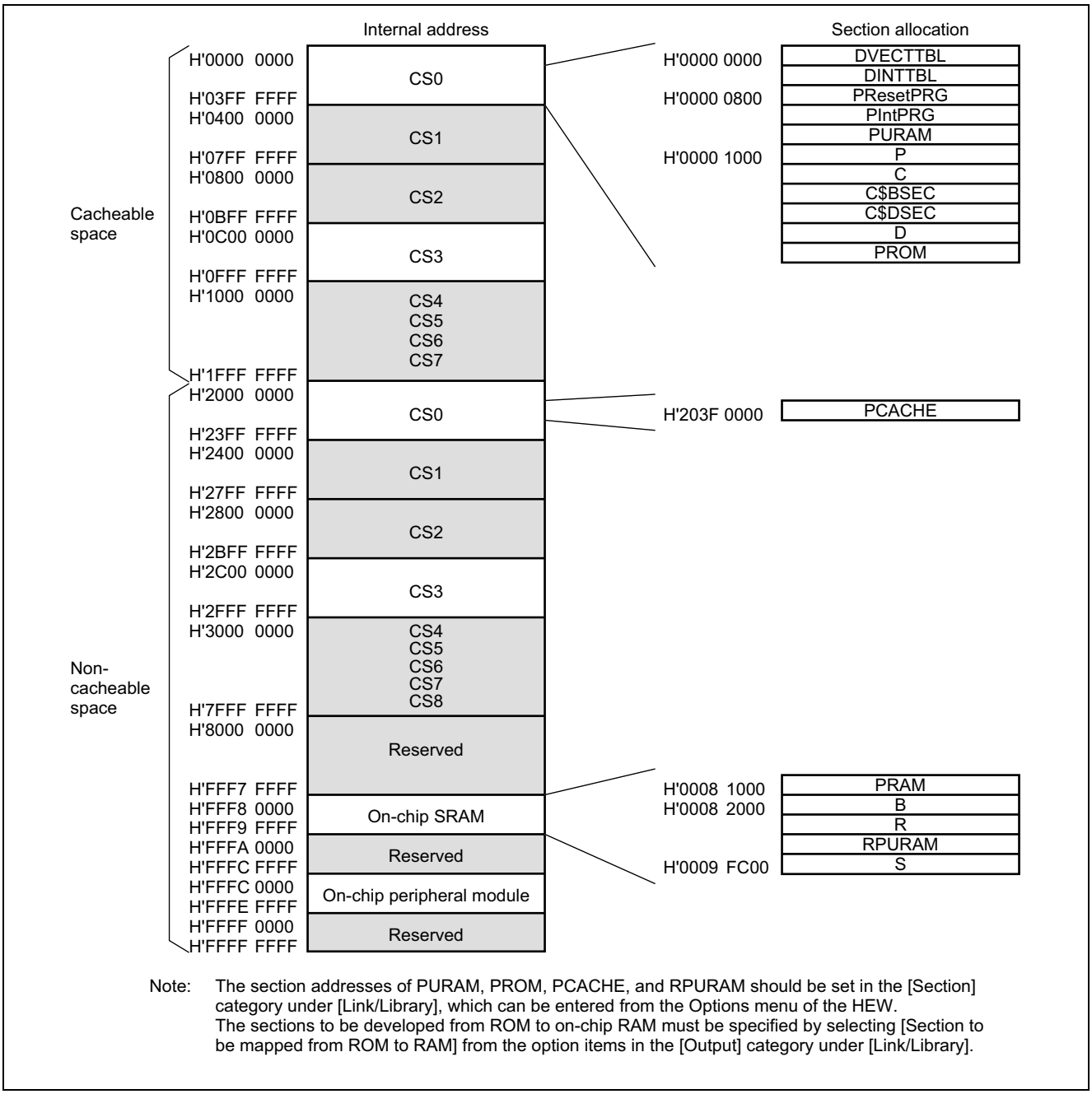
2. Description of Sample Application

This sample application applies the direct memory access controller (DMAC) to transfer of programs in an external ROM to the on-chip RAM, and executes the program in the on-chip RAM.

2.1 Section Allocation of Sample Program

The section name of the program to be transferred can be changed using "#pragma section", compiler's expanded function. In the sample program, the program section of the program to be transferred is changed to section PROM, and the program section in the on-chip RAM area to which the program is transferred will be section PRAM.

Figure 1 shows a memory map for the sample program.
Since the cache setting program must be executed in a non-cacheable space, section PCACHE is allocated in a non-cacheable space. Likewise, the program for setting AC characteristics switching must be executed in on-chip RAM, so section PURAM is allocated in the transfer source ROM area and section RPURAM is allocated in the transfer destination RAM area.



Note: The section addresses of PURAM, PROM, PCACHE, and RPURAM should be set in the [Section] category under [Link/Library], which can be entered from the Options menu of the HEW.
The sections to be developed from ROM to on-chip RAM must be specified by selecting [Section to be mapped from ROM to RAM] from the option items in the [Output] category under [Link/Library].

Figure 1 Memory Map

2.2 Setting the Linkage Editor

The section addresses are specified by options of the linkage editor. Table 1 lists the section names to be specified in the sample program. Table 2 shows linkage editor options to be used.

Table 1 Section Names to be Specified in Sample Program

Section Name	Description
PROM	Transfer source in which the program to be transferred is stored
PRAM	Transfer destination to which the program is transferred
PCACHE	A cache setting program is stored
PURAM	Transfer source of the program for setting AC characteristics switching
RPURAM	Transfer destination of the program for setting AC characteristics switching

Note: Section addresses are specified by selecting “SuperH RISC engine Standard Toolchain” from the pull-down menu in the HEW window. For details, refer to the HEW Manual.

Table 2 Linkage Editor Options

Option	Description
-rom=D=R,**PROM=PRAM**,PURAM=RPURAM	Specifies the ROM sections to be mapped to RAM.
start=DVECTTBL,DINTTBL/00,PResetPRG, PIntPRG,PURAM/0800,P,C,C$BSEC,C$DSEC,D, **PROM/1000**,**PCACHE/203F0000**, **PRAM/FFF81000**, B,R,RPURAM/FFF80000,S/FFF9FC00	Specifies the section start addresses.

2.3 Methods for Acquiring Section Addresses

Section address operators of the compiler are used to acquire section addresses by the program. Table 3 lists the section address operators.

Table 3 Address Operators

Syntax	Function
_sectop(“<section name>”)	Refers to the start address of the specified <section name>.
_secend(“<section name>”)	Refers to the end address of the specified <section name> + 1.
_secsize(“<section name>”)	Generates the size of the specified <section name>.

2.4 Operation of Sample Program

In the sample program, the direct memory access controller (DMAC) is placed in auto request mode, and data equal to the size of section PROM are transferred from the start address of section PROM in the CS0 space to section PRAM allocated in the on-chip RAM. For confirmation of the operation, function cmt0, which uses the compare-match timer, is placed in section PROM and is transferred to the on-chip RAM. After the transfer has been completed, function cmt0 is executed.

2.5 Notes when Changing to Transfer by the CPU

If a program is transferred to cacheable space by the CPU (the transfer is by software) while the operand cache (write-back mode) is enabled, the transferred program is taken into the operand cache and instruction fetches may not succeed. When a program is transferred by the CPU, the program should be transferred to a non-cacheable space, or write-back of the operand cache should be performed.

Since the sample program uses the DMAC to transfer a program, transfer is not affected by the state of the operand cache.

2.6 Register Settings and Processing Sequence of Sample Program

The register settings of the sample program are shown in table 4, and processing flow of the sample program is shown in figure 2.

Table 4 Register Settings in the Sample Program

Register Name	Address	Setting	Description
Standby control register 2 (STBCR2)	H'FFFE0018	H'00	MSTP8 = 0: The DMAC runs.
DMA channel control register_0 (CHCR_0)	H'FFFE100C	H'00000000	Before initial setting for DMA is made, DE = 0: DMA transfer is disabled.
		H'00005428	DMA initial setting TC = 1: Transfer by the number of times set in DMATCR in response to a DMA request. DM = B'01: Destination address is incremented. SM = B'01: Source address is incremented. RS = B'0100: Auto request TB = 1: Burst mode TS = B'01: Word transfer DE = 0: DMA transfer is disabled.
		H'80005429	When enabling DMA transfer, DE = 1: DMA transfer is enabled.
DMA source address register_0 (SAR_0)	H'FFFE1000	—	Transfer source address: Start address of section PROM
DMA destination address register_0 (DAR_0)	H'FFFE1004	—	Transfer destination address: Start address of section PRAM
DMA transfer count register_0 (DMATCR_0)	H'FFFE1008	—	Number of DMA transfers: Half the size of section PROM
DMA operation register (DMAOR)	H'FFFE1200	H'00000001	DME = 1: DMA transfer is enabled on all channels.

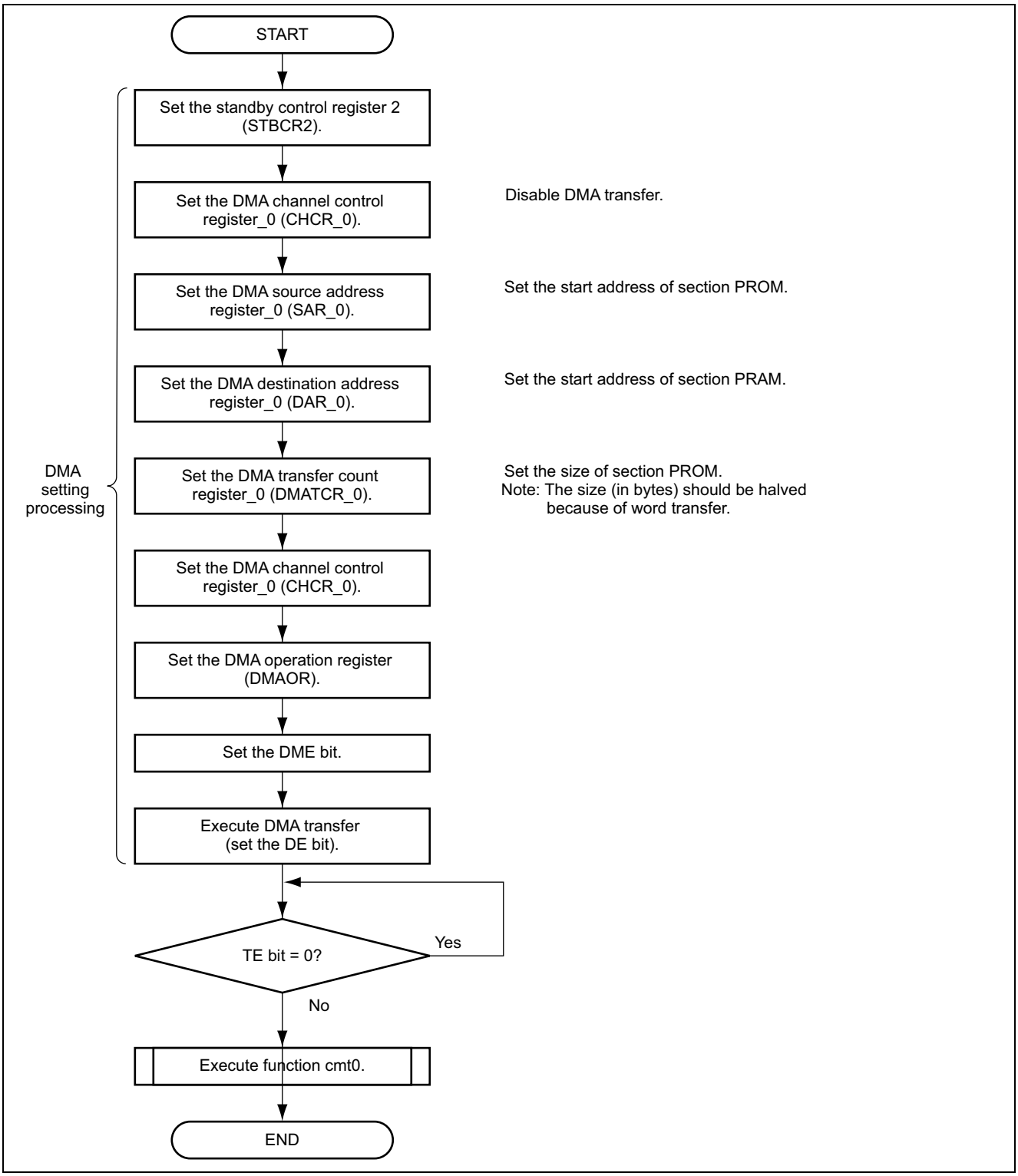


Figure 2 Processing Flow of the Sample Program

3. Sample Program Listing

1. Sample Program Listing: main.c (1)

```
/*""FILE COMMENT""*******************************************************************************
*
*   System Name : SH7206 Sample Program
*   File Name   : main.c
*   Contents    : Transferring a program section from ROM to on-chip RAM
*   Version     : 1.00.00
*   Model       : M3A-HS60
*   CPU         : SH7206
*   Compiler    : SHC9.0.00
*
*   Note        : Sample program for transferring the specified program section
*               : from external memory (ROM) to on-chip SRAM by the DMAC and
*               " executing the transferred program
*
*                <Caution>
*                This sample program is for reference
*                and its operation is not guaranteed.
*                Customers should use this sample program for technical reference
*                in software development.
*
*    Copyright (C) 2004 Renesas Technology Corp. All Rights Reserved
*    and Renesas Solutions Corp. All Rights Reserved
*
*    history     : 2004.10.28 ver.1.00.00
*""FILE COMMENT END""*************************************************************************/
#include <machine.h>
#include "iodefine.h"                   /* iodefine.h is automatically created by HEW    */

/* ==== Prototype declarations ==== */
void main(void);
void cmt0(void);
void io_init_cmt0(void);

```

2. Sample Program Listing: main.c (2)

```
/*""FUNC COMMENT""*****************************************************************************
* ID            :
* Module summary: Main function of the sample program (program transfer from ROM to RAM)
*-----------------------------------------------------------------------------------------------
* Include       : #include "iodefine.h"
*-----------------------------------------------------------------------------------------------
* Declaration   : void main(void)
*-----------------------------------------------------------------------------------------------
* Functional description: Section PROM is transferred to section PRAM
*               : allocated in on-chip SRAM by the DMAC, and the transferred
*               : program (function cmt0) is executed.
*-----------------------------------------------------------------------------------------------
* Argument      : None
*-----------------------------------------------------------------------------------------------
* Return value  : None
*-----------------------------------------------------------------------------------------------
* Notes         : Section PROM and section PRAM as the transfer destination must be
*               : added to the section definitions of the optimizing linkage editor,
*               : and [Section to be mapped from ROM to RAM] must be specified.
*""FUNC COMMENT END""*************************************************************************/
void main(void)
{
    /* ==== Setting standby control register 2 (STBCR2) ==== */
    CPG.STBCR2.BIT.MSTP8 = 0x0;             /* Cancel DMAC module stop mode               */

    /* ---- Setting DMA channel control register (CHCR_0) ---- */
    DMAC.CHCR0.BIT.DE = 0x0ul;              /* Disable DMA transfer                       */

    /* ==== Setting DMA source address register_0 (SAR_0) ==== */
    DMAC.SAR0.LONG = (unsigned long)__sectop("PROM");
                                            /* Refer to the start address of section PROM */
    /* ==== Setting DMA destination address register_0 (DAR_0) ==== */
    DMAC.DAR0.LONG = (unsigned long)__sectop("PRAM");
                                            /* Refer to the start address of section PRAM */
    /* ==== Setting DMA transfer count register_0 (DMATCR_0) ==== */
    DMAC.DMATCR0.LONG = __secsize("PROM")/2;

    /* ==== Setting DMA channel control register_0 (CHCR_0) ==== */
    DMAC.CHCR0.LONG |= 0x80005428ul;
        /*
            bit31   : TC: 1-------------- Transfer by the number of times specified in DMATCR
            bit30-29: reserved 0
            bit28   : RLD OFF : 0-------------------- Disable reload function
            bit27-24: reserved 0
            bit23   : DO over run0 : 0--------------- Unused
            bit22   : TL TEND low active : 0--------- Unused
            bit21-20: reserved 0
            bit19   : HE :0-------------------------- Unused
            bit18   : HIE :0------------------------- Unused
            bit17   : AM :0-------------------------- Unused
            bit16   : AL :0-------------------------- Unused
            bit15-14: DM1:0 DM0:1-------------------- Increment destination address
            bit13-12: SM1:0 SM0:1-------------------- Increment source address
            bit11-8 : RS : auto request : B'0100----- Auto request
            bit7    : DL : DREQ level : 0 ----------- Unused
            bit6    : DS : DREQ select :0 Low level-- Unused
            bit5    : TB : cycle :1------------------ Burst mode
            bit4-3  : TS : transfer size: B'01------- Word transfer
            bit2    : IE : interrupt enable: 0------- Disable interrupt
            bit1    : TE : transfer end : 0---------- Clear the TE flag
            bit0    : DE : DMA enable bit: 0--------- Disable DMA transfer
        */

```

3. Sample Program Listing: main.c (3)

```
 97      /* ==== Setting DMA operation register (DMAOR) ==== */
 98      DMAC.DMAOR.WORD &= 0x0000u;
 99          /*
100             bit15-14 : reserved 0
101             bit13-12 : CMS1:0 CMS0:0------------------ Normal mode
102             bit11-10 : reserved 0
103             bit9-8   : PR1:0 PR0:0 ------------------- Fixed mode 1
104             bit7-3   : reserved 0
105             bit2     : AE: 0 ------------------------- Clear the address error flag
106             bit1     : NMIF: 0 ----------------------- Clear the NMI flag
107             bit0     : DME:0 ------------------------- Disable DMA transfer on all channels
108          */
109
110     /* ==== Setting DME bit ==== */
111     DMAC.DMAOR.BIT.DME = 1ul;
112
113     /* ==== Executing DMA transfer (setting DE bit) ==== */
114     DMAC.CHCR0.BIT.DE = 1ul;
115
116     /* ==== TE bit = "0"?==== */
117     while(DMAC.CHCR0.BIT.TE == 0ul){
118         /* ==== Wait for completion of DMA transfer ==== */
119     }
120
121     /* ==== Executing function CMT0 ==== */
122     cmt0();                                  /* Port E1 inversion processing          */
123 }
124
125
```

4. Sample Program Listing: main.c (4)

```
126  #pragma section ROM             /* P section from here is defined as section PROM */
127
128  /*""FUNC COMMENT""*****************************************************************************
129  * ID            :
130  * Module summary: Counting at a constant cycle
131  *-----------------------------------------------------------------------------------------------
132  * Include       : #include "iodefine.h"
133  *-----------------------------------------------------------------------------------------------
134  * Declaration   : void cmt0(void)
135  *-----------------------------------------------------------------------------------------------
136  * Functional description: Initializes (1 ms) IO port PE1 (connected to an LED) and
137  *               : compare-match timer CMT0, and turns on or off the LED connected
138  *               : to PE1 every thousandth setting of 1-ms flag (interrupt request bit).
139  *-----------------------------------------------------------------------------------------------
140  * Argument      : None
141  *-----------------------------------------------------------------------------------------------
142  * Return value  : None
143  *-----------------------------------------------------------------------------------------------
144  * Notes         : This module is placed in section PROM so that it is transferred.
145  *               : Section PROM and section PRAM as the transfer destination must
146  *               : be added to the section definitions of the optimizing linkage editor,
147  *               : and [Section to be mapped from ROM to RAM] must be specified.
148  *""FUNC COMMENT END""*************************************************************************/
149  void cmt0(void)
150  {
151      volatile unsigned int CountCMT0;   /* For one-second software counter */
152
153      /* ==== Port E initialization ==== */
154      PORT.PECRL1.WORD = 0x0000;         /* Set the pin as PE1 */
155      PORT.PEIORL.WORD = 0x0002;         /* Set PE1 as an output pin */
156      PORT.PEDRL.WORD  = 0x0002;         /* Write output value of 1 to port E data register */
157
158      /* ==== One-second software counter (CountCMT0) initialization ==== */
159      CountCMT0 = 1000u;                 /* Count 1000 times */
160
161      /* ==== CMT0 (1-ms constant cycle timer) initialization processing ==== */
162      io_init_cmt0();
163
164      while(1){
165          /* ---- Checking compare match (1ms) flag ---- */
166          while (CMT.CMCSR0.BIT.CMF == 0){
167              /* Wait until 1 ms elapses */
168          }
169
170          CMT.CMCSR0.BIT.CMF = 0;        /* Clear the compare match flag (CMF) to 0         */
171          CountCMT0--;                   /* Update one-second software counter (CountCMT0)  */
172
173          /* ---- Checking one-second software counter ---- */
174          if(CountCMT0 == 0u){
175              CountCMT0 = 1000u;         /* Re-initialize the one-second software counter   */
176              PORT.PEDRL.BIT.PE1DR ^= 1u ; /* Port E1 output inversion processing           */
177          }
178      }
179  }
180
181
```

5. Sample Program Listing: main.c (5)

```
/*""FUNC COMMENT""*****************************************************************************
* ID            :
* Module summary: Configuring CMT0 as a periodic timer
*-----------------------------------------------------------------------------------------------
* Include       : #include "iodefine.h"
*-----------------------------------------------------------------------------------------------
* Declaration   : void io_init_cmt0(void)
*-----------------------------------------------------------------------------------------------
* Functional description: Sets CMT0 so that the CMF flag is set every 1 ms.
*-----------------------------------------------------------------------------------------------
* Argument      : None
*-----------------------------------------------------------------------------------------------
* Return value  : None
*-----------------------------------------------------------------------------------------------
* Notes         : This module is placed in section PROM so that it is transferred.
*               : In the section definitions of the optimizing linkage editor, section
*               : PROM and transfer destination section PRAM must be added along with
*               : specification as "Section to be mapped from ROM to RAM".
*""FUNC COMMENT END""*************************************************************************/
void io_init_cmt0(void)
{
    /* ==== Initial settings for periodic (1 ms) timer ==== */
    /* ---- Setting standby control register 4 (STBCR4) ---- */
    CPG.STBCR4.BIT.MSTP42 = 0x0;            /* Cancel CMT module stop mode              */

    /* ---- Setting compare match timer start register (CMSTR) ---- */
    CMT.CMSTR.WORD = 0x0000;                /* Stop channel 0 counter                   */

    /* ---- Setting compare match timer control/status register (CMCSR0) ---- */
    CMT.CMCSR0.WORD = 0x0002;               /* Disable compare match interrupt,         */
                                            /*          and set 1/128 peripheral clock  */
    /* ---- Setting compare match timer counter register (CMCNT0) ---- */
    CMT.CMCNT0.WORD = 0x0000;               /* Clear timer counter                      */

    /* ---- Setting compare match timer constant register (CMCOR0) ---- */
    CMT.CMCOR0.WORD = 0x0104;               /* Set the period until compare match (1 ms) */

    /* ---- Setting compare match timer start register (CMSTR) ---- */
    CMT.CMSTR.BIT.STR = 0x1;                /* Start counting                           */
}
/* End of File */
```

4. Reference Documents

- SH-2A SH2A-FPU Software Manual (Rev.3.00)
  (Download the latest edition from the website of Renesas Technology Corp.)
- SH7206 Group Hardware Manual (Rev. 1.00)
  (Download the latest edition from the website of Renesas Technology Corp.)

5. Website

- Website of Renesas Technology Corp.
  http://www.renesas.com/

Revision Record

		Description	
Rev.	Date	Page	Summary
1.00	Sep.05.05	—	First edition issued

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.