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.

M16C/80 Group

DMA setup procedures (when using ASM function)

1.0 Abstract

The following article introduces DMA (DMA0 to DMA4) setup procedures using and its application example.

2.0 Introduction

The explanation of this issue is applied of the following condition.

Applicable MCU: M16C/80 Group

3.0 Description of the application example

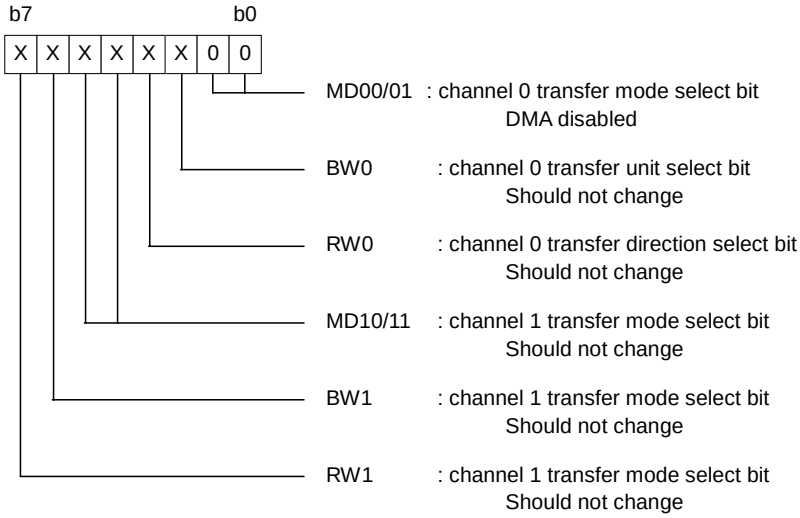
This chapter describes DMAC setup procedures using ASM function (in-line assemble description) for C compiler (M3T-NC308WA) for M32C/80, M16C/80 series.

3.1 Setup procedures

The setup procedures and the setting value will be shown to use DMA0. Refer to M16C/80 Group datasheet and technical news on DMA for the details of each register.

(1) Set DMA mode register 0 (DMD0)

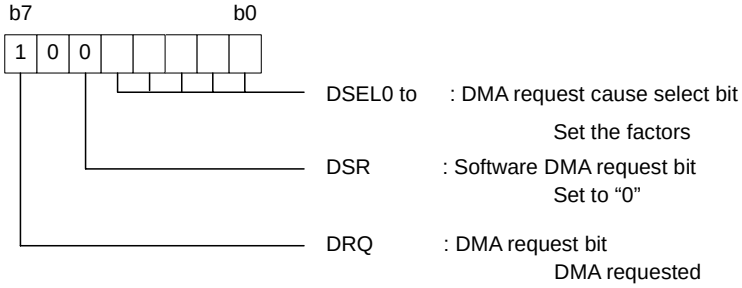
Set DMA0 transfer mode select bit “00” (DMA disabled)



(2) Set DMA0 request cause select register (DM0SL)

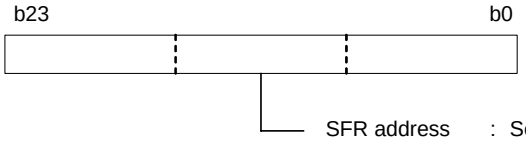
Set the request factors that trigger DMA transfer using DMA request cause select bit. (Note)

Set DMA request bit to “1” (DMA requested).



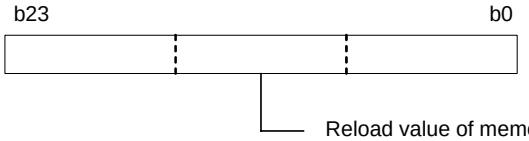
(Note) Peripheral functions selected by DMA request cause select bit should be set to disabled for this case.

(3) Set DMA0SFR address register (DSA0)



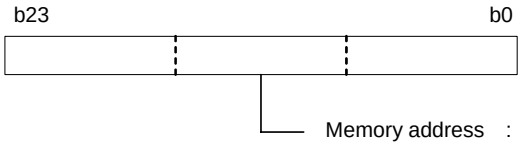
SFR address : Set the source address or the destination fixed address.

(4) Set DMA0 memory address reload register (DRA0)



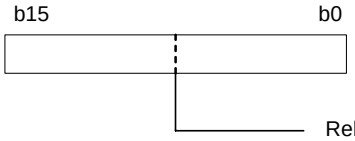
Reload value of memory address register : Set the source address or the destination address.

(5) Set DMA0 memory address register (DMA0)



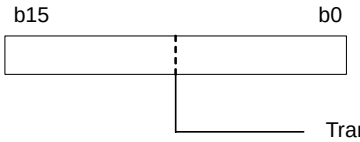
Memory address : Set the source address or the destination address.

(6) Set DMA0 transfer count reload register (DRC0)



Reload value of transfer count register : Set the number of transfer.

(7) Set DMA0 transfer count register (DCT0)



Transfer counter : Set the number of transfer

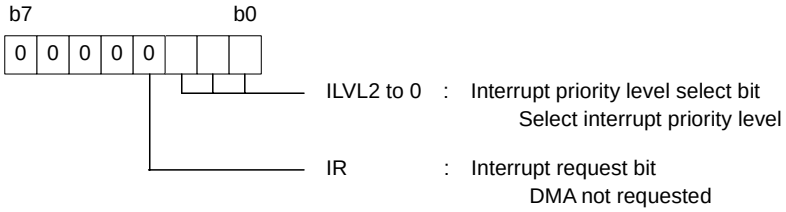
(8) Insert dummy cycle

Insert the same number of “NOP instruction” obtained by the following formula as the number of used DMA channel.

The number of dummy cycle = 8+6n (n=the number of every potential channel that DMA request factors generate except for corresponding channels)

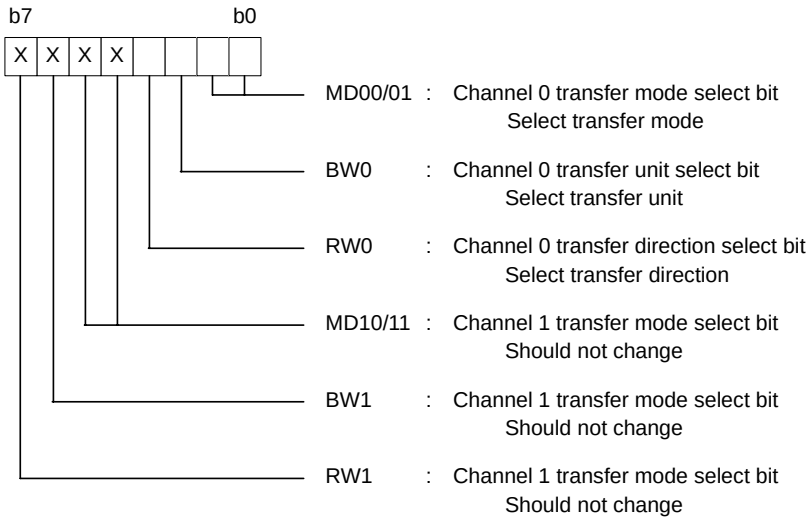
Refer to “6. Program Sample” for inserting dummy cycle.

(9) Set DMA0 interrupt control register (DM0IC)



(10)Reset DMA mode register 0 (DMD0)

Set transfer mode select bit, transfer unit select bit, and transfer direction set bit.



(11)Peripheral functions are initiated as DMA0 request factors

Cautions

When using DMA2 and DMA3,

Register bank 1 cannot be used.

High-speed interrupt cannot be used.

4.0 Reference

Datasheet

Refer to M16C/80 group datasheet

(Acquire the most current version from Renesas Technology website)

Technical News

Refer to technical news on M16C/80 group DMA.

- M16C-87-0207
- M16C-44-0001

(Acquire the most current version from Renesas Technology website)

5.0 Web-site and contact for support

Renesas Web-site

http://www.renesas.com

Contact for Renesas technical support

Mail to : support_apl@renesas.com

4.0 Program Sample

A sample program will be shown below when DMA request cause = “Timer A1, transfer memory space = from any address in the 16Mbyte space to a fixed address, transfer unit =8 bit, transfer mode = repeat transfer for DMA0 to DMA4 setting using ASM function.

```
/***********************************************************************/
/*      FILE NAME : rjj05b0111_src.c                                   */
/*      Ver       : 1.00                                               */
/*      CPU       : M16C/80                                            */
/*      FUNCTION  : The DMA setting procedure in the C language.       */
/*---------------------------------------------------------------------*/
/*   Copyright(C)2003, Renesas Technology Corp.                        */
/*   Copyright(C)2003, Renesas Solutions Corp.                         */
/*   All rights reserved.                                              */
/***********************************************************************/
/****************************************/
/*      include file                    */
/****************************************/
#include "sfr80144.h"       // Special Function Register Header File

/****************************************/
/*      Function declaration            */
/****************************************/
void ta1_init(void);
void dma0_int(void);
void dma1_int(void);
void dma2_int(void);
void dma3_int(void);


/****************************************/
/*      Global variable declaration     */
/****************************************/
//DMA0 output data.
static char     data0[] = {0x00,0x01,0x02,0x03,0x04,0x05,0x06,0x07,
                           0x08,0x09,0x0a,0x0b,0x0c,0x0d,0x0e,0x0f};
//DMA1 output data.
static char     data1[] = {0x10,0x11,0x12,0x13,0x14,0x15,0x16,0x17,
                           0x18,0x19,0x1a,0x1b,0x1c,0x1d,0x1e,0x1f};
//DMA2 output data.
static char     data2[] = {0x20,0x21,0x22,0x23,0x24,0x25,0x26,0x27,
                           0x28,0x29,0x2a,0x2b,0x2c,0x2d,0x2e,0x2f};
//DMA3 output data.
static char     data3[] = {0x30,0x31,0x32,0x33,0x34,0x35,0x36,0x37,
                           0x38,0x39,0x3a,0x3b,0x3c,0x3d,0x3e,0x3f};
//DMDi register temporally.
short           dmd_tmp;

/****************************************/
/*      main function                   */
/****************************************/
void main(void)
{

    pd0   = 0xff;       // P0 is an output port.(for dma0 output)
    pd1   = 0xff;       // P1 is an output port.(for dma1 output)
    pd2   = 0xff;       // P2 is an output port.(for dma2 output)
    pd3   = 0xff;       // P3 is an output port.(for dma3 output)
    pd6   = 0xff;       // P6 is an output port.
    pd8   = 0x1f;       // P8_0-P8_4 are an output port.
    p0    = 0;          // P0 initialize output
    p1    = 0;          // P1 initialize output
    p2    = 0;          // P2 initialize output
```

```
p3    = 0;            // P3 initialize output
p6    = 0;            // P6 initialize output
p8    = 0;            // P8 initialize output


ta1_init();           // Timer-A1 initialize(for DMA transmission trigger)

// (1) A setup of DMA0.
asm("ldc     #000h,       dmd0");          // (1-1) DMA0,1 disable(DMD0)
asm("mov.b   #084h,       _dm0sl_addr"); // (1-2) A setup of DM0SL(DRQ=1, TA1)
asm("ldc     #_p0_addr, dsa0");          // (1-3) A setup of DSA0(DMA0 SFR address register)
asm("ldc     #_data0,     dra0");        // (1-4) A setup of DRA0(DMA0 memory address reload register)
asm("ldc     #_data0,     dma0");        // (1-5) A setup of DMA0(DMA0 memory address register)
asm("ldc     #16,         drc0");        // (1-6) A setup of DRC0(DMA0 transfer count reload register)
asm("ldc     #16,         dct0");        // (1-7) A setup of DCT0(DMA0 transfer count register)
                                         // (1-8) Dummy cycle insertion
asm("NOP     ");                         //     8 * 6n (n = other used DMA number)
asm("NOP     ");
asm("NOP     ");
asm("NOP     ");
asm("NOP     ");
asm("NOP     ");
asm("NOP     ");
asm("NOP     ");
asm("NOP     ");
asm("NOP     ");
asm("NOP     ");
asm("NOP     ");
asm("NOP     ");
asm("NOP     ");
asm("NOP     ");
asm("NOP     ");
asm("NOP     ");
asm("NOP     ");
asm("NOP     ");
asm("NOP     ");
asm("NOP     ");
asm("NOP     ");
asm("NOP     ");
asm("NOP     ");
asm("NOP     ");
asm("NOP     ");

dm0ic = 0x5;                             // (1-9) A set of DM0IC(DMA0 interrupt control register)

asm("stc     dmd0,        _dmd_tmp");     // dmd0 is stored to _dmd_tmp
                                         //     (_dmd_tmp is automatic data for dmdi register temporally).
asm("or.b    #00bh,       _dmd_tmp");
asm("ldc     _dmd_tmp,    dmd0");         // (1-10)DMA0 permission(DMD0)
                                         //    Repeat transfer(memory -> fixed address)

// (2) A setup of DMA1.
asm("stc     dmd0,        _dmd_tmp");     // dmd0 is stored to _dmd_tmp
                                         //     (_dmd_tmp is automatic data for dmdi register temporally).
asm("and.b   #00fh,       _dmd_tmp");
asm("ldc     _dmd_tmp,    dmd0");         // (2-1) DMA1 disable(DMD0)
asm("mov.b   #084h,       _dm1sl_addr"); // (2-2) A setup of DM1SL(DRQ=1, TA1)
asm("ldc     #_p1_addr, dsa1");          // (2-3) A setup of DSA1(DMA0 SFR address register)
asm("ldc     #_data1,     dra1");        // (2-4) A setup of DRA1(DMA0 memory address reload register)
asm("ldc     #_data1,     dma1");        // (2-5) A setup of DMA1(DMA0 memory address register)
asm("ldc     #16,         drc1");        // (2-6) A setup of DRC1(DMA0 transfer count reload register)
asm("ldc     #16,         dct1");        // (2-7) A setup of DCT1(DMA0 transfer count register)
                                         // (2-8) Dummy cycle insertion
asm("NOP     ");                         //     8 * 6n (n = other used DMA number)
asm("NOP     ");
asm("NOP     ");
asm("NOP     ");
```

```
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");

dm1ic = 0x5;                                    // (2-9) A set of DM1IC(DMA1 interrupt control register)

asm("stc       dmd0,          _dmd_tmp");       // dmd0 is stored to _dmd_tmp
                                                //      (_dmd_tmp is automatic data for dmdi register temporally).
asm("or.b      #0b0h,         _dmd_tmp");
asm("ldc       _dmd_tmp,    dmd0");             // (2-10)DMA1 permission(DMD0)
                                                //    Repeat transfer(memory -> fixed address)

// (3) A setup of DMA2.
asm("ldc       #000h,         dmd1");           // (3-1) DMA2,3 disable(DMD1)
asm("mov.b    #084h,          _dm2sl_addr");    // (3-2) A setup of DM2SL(DRQ=1, TA1)
asm("fclr      I");                             // (3-3) Interrupt disabled.
asm("fset      B");                             // (3-4) Register-bank1 enable
asm("ldc       #_p2_addr, sb");                 // (3-5) A setup of DSA2(DMA2 SFR address register)
asm("ldc       #_data2,     svp");              // (3-6) A setup of DRA2(DMA2 memory address reload register)
asm("mov.l    #_data2,     a0");               // (3-7) A setup of DMA2(DMA2 memory address register)
asm("mov.w    #16,           r2");             // (3-8) A setup of DRC2(DMA2 transfer count reload register)
asm("mov.w    #16,           r0");             // (3-9) A setup of DCT2(DMA2 transfer count register)
asm("fclr      B");                             // (3-10)Register-bank1 disable
                                                // (3-11)Dummy cycle insertion
asm("NOP       ");                              //      8 * 6n (n = other used DMA number)
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
```

```
asm("NOP       ");
asm("NOP       ");

dm2ic = 0x5;                                    // (3-11)A set of DM2IC(DMA2 interrupt control register)

asm("stc       dmd1,          _dmd_tmp");       // dmd1 is stored to _dmd_tmp
                                                //      (_dmd_tmp is automatic data for dmdi register temporally).
asm("or.b      #00bh,         _dmd_tmp");
asm("ldc       _dmd_tmp,      dmd1");           // (3-12)DMA2 permission(DMD1)
                                                //    Repeat transfer(memory -> fixed address)

// (4) A setup of DMA3.
asm("stc       dmd1,          _dmd_tmp");       // dmd1 is stored to _dmd_tmp
                                                //      (_dmd_tmp is automatic data for dmdi register temporally).
asm("and.b     #00fh,         _dmd_tmp");
asm("ldc       _dmd_tmp,      dmd1");           // (4-1) DMA3 disable(DMD1)
asm("mov.b     #084h,         _dm3sl_addr"); // (4-2) A setup of DM3SL(DRQ=1, TA1)
asm("fclr      I");                           // (4-3) Interrupt disabled.
asm("fset      B");                            // (4-4) Register-bank1 enable
asm("ldc       #_p3_addr, fb");               // (4-5) A setup of DSA3(DMA3 SFR address register)
asm("ldc       #_data3,       vct");          // (4-6) A setup of DRA3(DMA3 memory address reload register)
asm("mov.l     #_data3,       a1");           // (4-7) A setup of DMA3(DMA3 memory address register)
asm("mov.w     #16,           r3");           // (4-8) A setup of DRC3(DMA3 transfer count reload register)
asm("mov.w     #16,           r1");           // (4-9) A setup of DCT3(DMA3 transfer count register)
asm("fclr      B");                           // (4-10)Register-bank1 disable
                                                // (4-10)Dummy cycle insertion
asm("NOP       ");                              //      8 * 6n (n = other used DMA number)
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");
asm("NOP       ");

dm3ic = 0x5;                                    // (4-11)A set of DM3IC(DMA3 interrupt control register)

asm("stc       dmd1,          _dmd_tmp");       // dmd1 is stored to _dmd_tmp
                                                //      (_dmd_tmp is automatic data for dmdi register temporally).
asm("or.b      #0b0h,         _dmd_tmp");
asm("ldc       _dmd_tmp,      dmd1");           // (4-12)DMA3 permission(DMD1)
                                                //    Repeat transfer(memory -> fixed address)



// (5) Interruption permission.
asm("fset I    ");

// (6) Operation of the circumference function of a DMA demand factor is permitted.
```

```
    ta1s = 1;

    while(1)
    {
        p8_1   = 1;             // test end...
    }


}

/****************************************/
/*      Timer-A1 initialization         */
/****************************************/
void ta1_init(void)
{
    ta1mr = 0x80;               // Timer-mode(f32)
    ta1    = 0x8fff;            // Timer value setup

}

/****************************************/
/*      DMA0 interrupt routine          */
/****************************************/
//#pragma     INTERRUPT/B   dma0_int
#pragma     INTERRUPT   dma0_int     //Please do not use a "/B" option, when you use DMA2 and DMA3.
void dma0_int(void)
{
    p6_0 = !p6_0;
}

/****************************************/
/*      DMA1 interrupt routine          */
/****************************************/
//#pragma     INTERRUPT/B   dma1_int
#pragma     INTERRUPT   dma1_int     //Please do not use a "/B" option, when you use DMA2 and DMA3.
void dma1_int(void)
{
    p6_1 = !p6_1;
}

/****************************************/
/*      DMA2 interrupt routine          */
/****************************************/
//#pragma     INTERRUPT/B   dma2_int
#pragma     INTERRUPT   dma2_int     //Please do not use a "/B" option, when you use DMA2 and DMA3.
void dma2_int(void)
{
    p6_2 = !p6_2;
}

/****************************************/
/*      DMA3 interrupt routine          */
/****************************************/
//#pragma     INTERRUPT/B   dma3_int
#pragma     INTERRUPT   dma3_int     //Please do not use a "/B" option, when you use DMA2 and DMA3.
void dma3_int(void)
{
    p6_3 = !p6_3;
}
```

REVISION HISTORY	M16C/80 Group Application Note DMAsetup procedures (when using ASM function)

Rev.	Date	Description	
		Page	Summary
1.00	Oct 9, 2003	-	First edition issued

Keep safety first in your circuit designs!

1. Renesas Technology Corporation 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 Corporation 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 Corporation or a third party.
2. Renesas Technology Corporation 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 Corporation without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corporation or an authorized Renesas Technology Corporation 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 Corporation 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 Corporation by various means, including the Renesas Technology Corporation 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 Corporation assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corporation 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 Corporation or an authorized Renesas Technology Corporation 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 Corporation 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 Corporation for further details on these materials or the products contained therein.