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.

M32C/85 Group

Application Example of Multiple-Channel PWM Using DMAC

1. Abstract

The following document describes procedures and examples of use to perform multiple-channel PWM output using DMAC.

2. Introduction

Application examples described in this document are applied to the following microcomputer and condition:
Applicable Microcomputer : M32C/85 Group

This program can be used for the other M16C Families which have the same SFR (Special Function Register) as the one in the M32C/85 Group. However, since some functions may be modified such as added functions, please ensure in a manual. Please evaluate sufficiently when using this application note.

3. Description of Use Example

Application examples which perform 64 PWM outputs from the ports P0 to P7 every timing of Timer B0 underflow using Timer B0 and 4 DMACs are described after preparing PWM output pattern on ROM.

Example of Use :

- System condition
  VCC1=VCC2=5.0V, XIN=8MHz, PLL=quadruple, f1=32MHz

- DMAC setting

	DMA Request Factor	Transfer Mode	Transfer Unit	Transfer Direction
DMA0	Timer B0 Interrupt Request	Repeat Transfer	16 Bit	Memory(forward)→Fixed address(P0) (Output to P0 to P1 in 16-bit unit)
DMA1				Memory(forward)→Fixed address(P2) (Output to P2 to P3 in 16-bit unit)
DMA2				Memory(forward)→Fixed address(P4) (Output to P4 to P5 in 16-bit unit)
DMA3				Memory(forward)→Fixed address(P6) (Output to P6 to P7 in 16-bit unit)

- TB0 setting
  Timer mode, count source=f1(32MHz), Period (Modify between 100μs to 900μs)

Operation :

PWM output is performed to 64 ports of P0 to P7 which function as output port by DMA transfer every underflow period of Timer B0. (connect (pull-up) to VCC1 via resistor since P7_0 and P7_1 are ports for the N-channel open drain output). Also, PWM pulse output width is modified by changing the timer value of Timer B0 during Timer B0 interrupt process. Figure 3.1 shows the PWM Output Timing Chart.

Precautions :

When timer pulse output is performed combining Timer B and DMAC, note the following points.

- DMA transfer priority
  When DMA request is generated simultaneously, transfer is performed by the priority of DMA0>DMA1>DMA2>DMA3.
- Delay cycle number of DMA transfer
  When transfer is performed under the conditions of this use example (program allocation area=internal ROM, transfer source=internal ROM, transfer distination=SFR area), delay is generated in 0 to 5 cycles transfer of the CPU clock. (The delayed cycle numbers are changed depending on wait numbers of memory when using the external memory).
- Delay time when Timer B starts
  The execution time of the instruction until Timer B starts after setting the port direction register to output is assumed as delay time for the first pulse output when Timer B starts.
- Delay time of pulse output
  The actual pulse output is output behind DMA transfer cycle numbers and DMA transfer delay cycle numbers after Timer B interrupt request is generated.

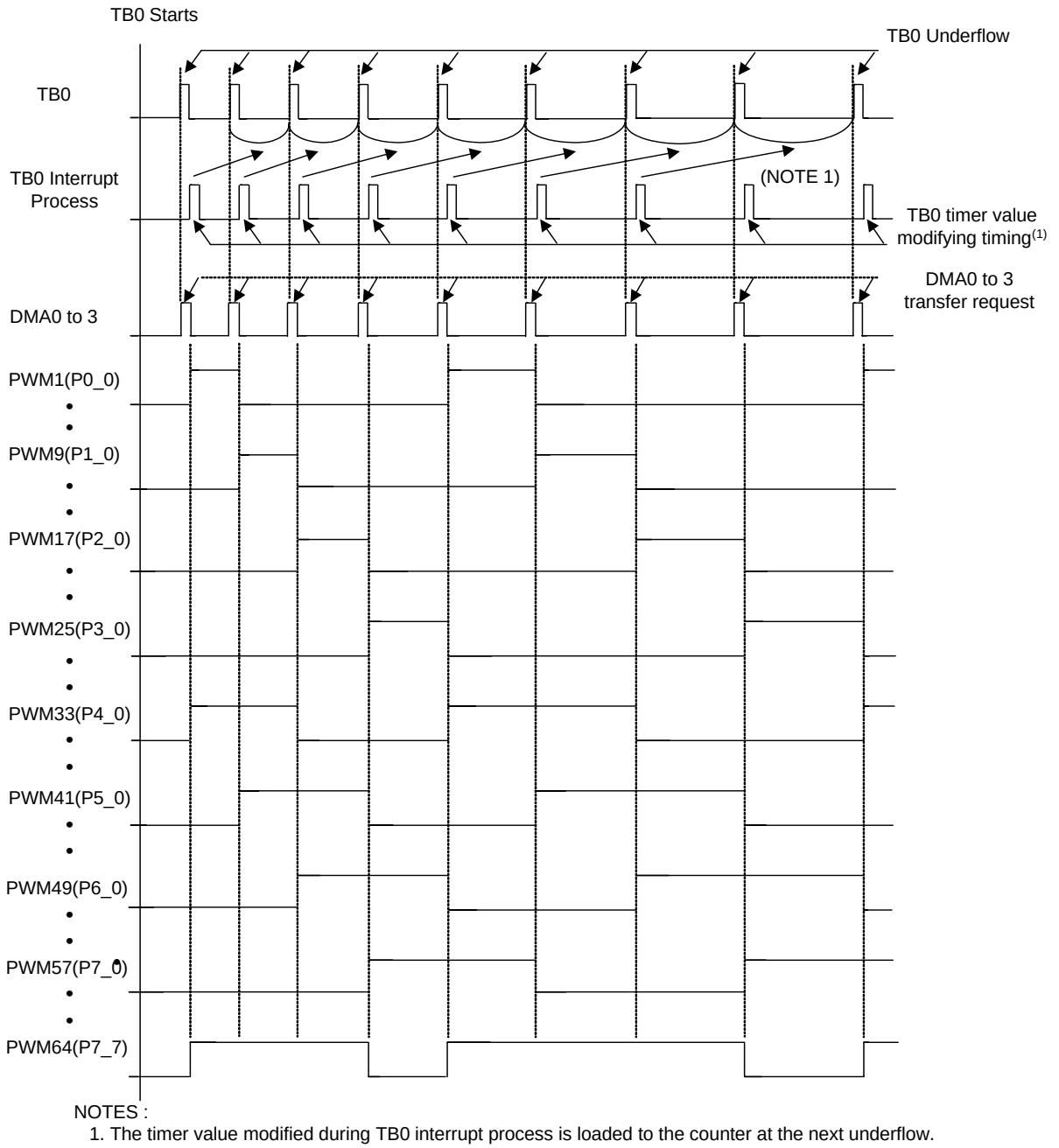
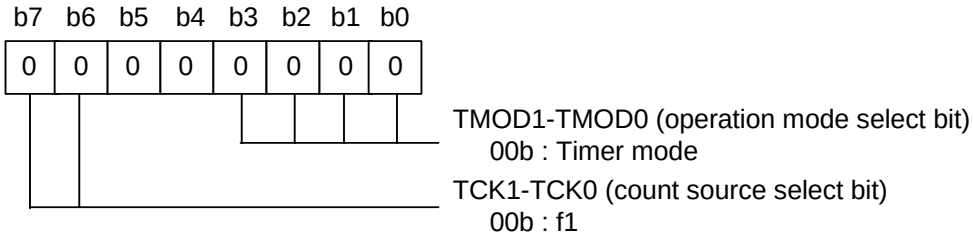


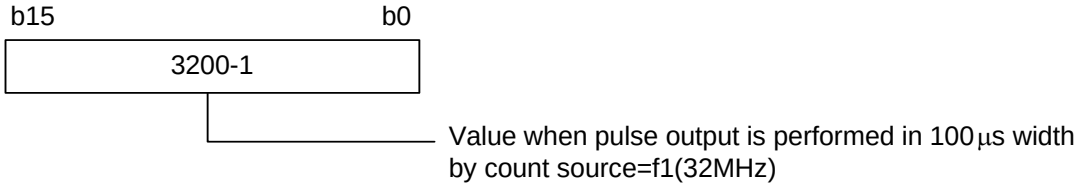
Figure 3.1 PWM Output Timing Chart

(1) Timer B0 (DMA0 to 3 request factors) setting

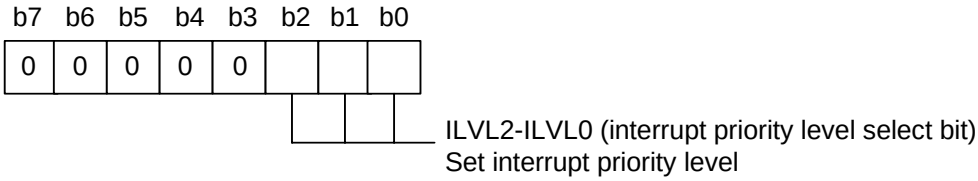
• Set TB0MR register (Timer B0 mode register)



• Set PWM pulse width default value to TB0 (Timer B0 register)

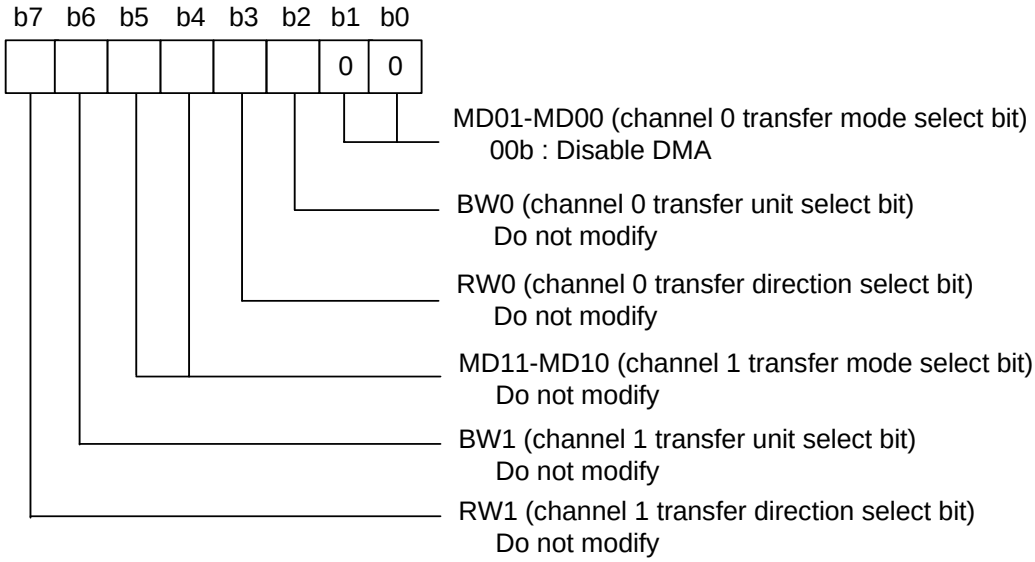


• Set TB0IC (Timer B0 interrupt control register)

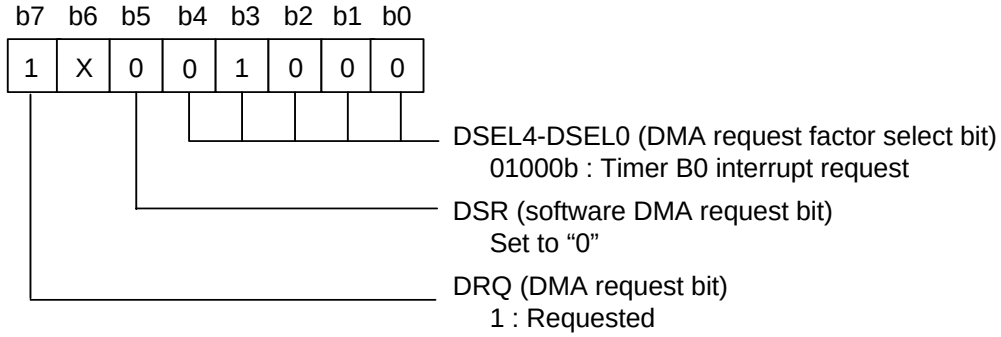


(2) DMA0 setting

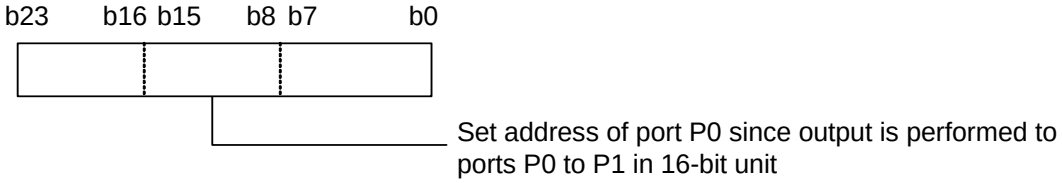
• Set (disable DMA0) DMD0 register (DMA mode register 0)



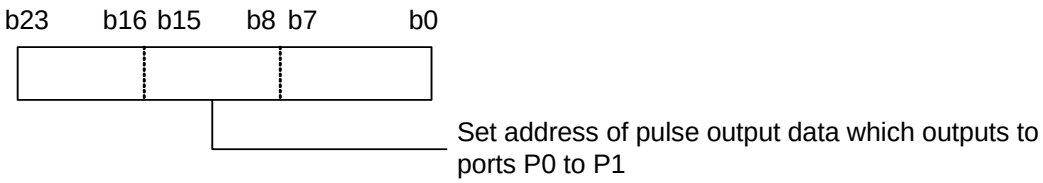
• Set DM0SL register (DMA0 factor select register)



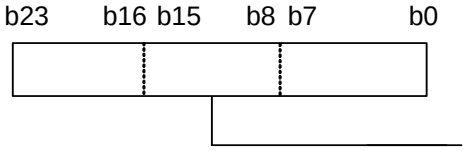
• Set DSA0 register (DMA0 SFR address register)



• Set DRA0 register (DMA0 memory address reload register)

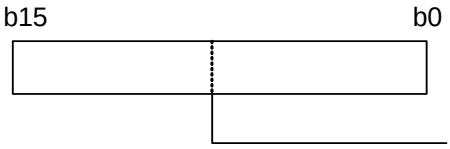


• Set DMA0 register (DMA0 memory address register)



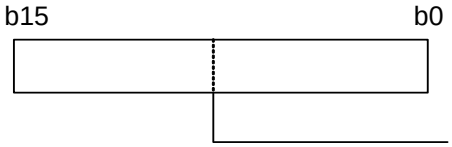
Set address of pulse output data which outputs to ports P0 to P1

• Set DRC0 register (DMA0 transfer count reload register)



Set numbers of pulse output data

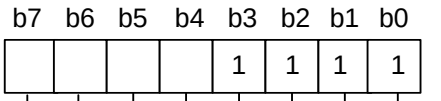
• Set DCT0 register (DMA0 transfer count register)



Set numbers of pulse output data

• Insert dummy cycle
Enable DMA after setting the DM0SL register and waiting for 6 cycles by BCLK.
This document shows inserting 6 NOPs waits for 6 cycles.

• Re-set (enable DMA0) DMD0 register (DMA mode register 0)



MD01-MD00 (channel 0 transfer mode select bit)
11b : Repeat transfer

BW0 (channel 0 transfer unit select bit)
1 : 16 bit

RW0 (channel 0 transfer direction select bit)
1 : Memory (forward) → Fixed address

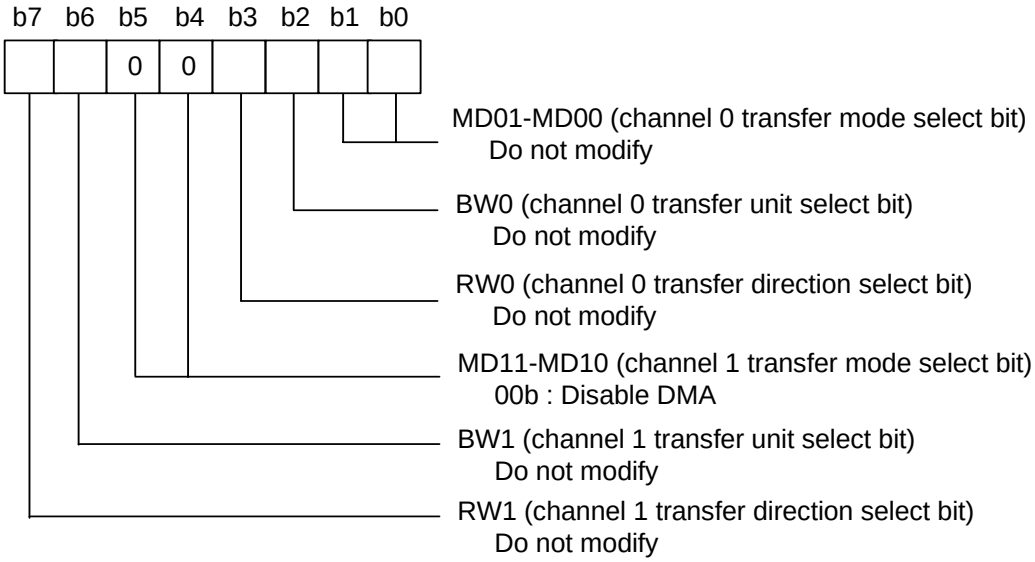
MD11-MD10 (channel 1 transfer mode select bit)
Do not modify

BW1 (channel 1 transfer unit select bit)
Do not modify

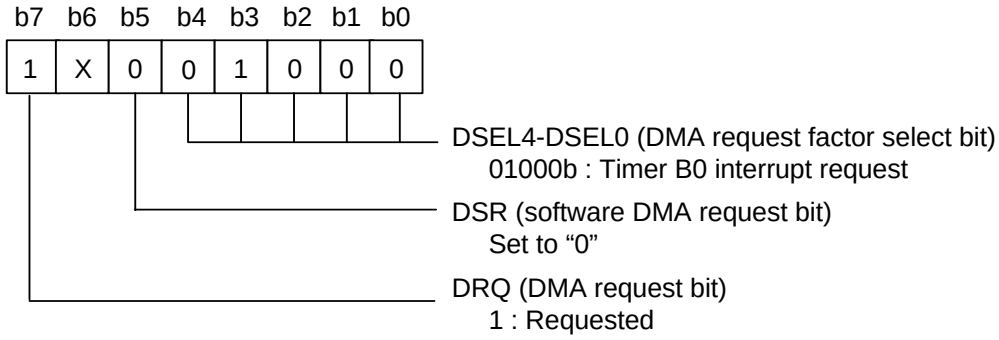
RW1 (channel 1 transfer direction select bit)
Do not modify

(3) DMA1 setting

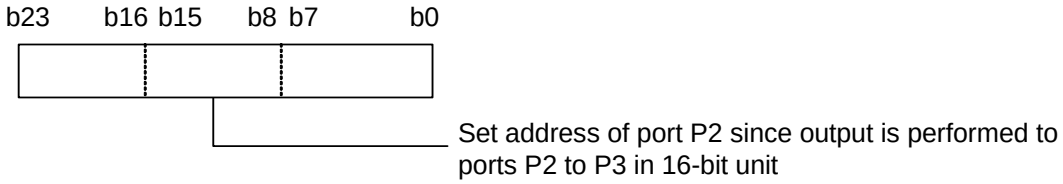
• Set (disable DMA1) DMD0 register (DMA mode register 0)



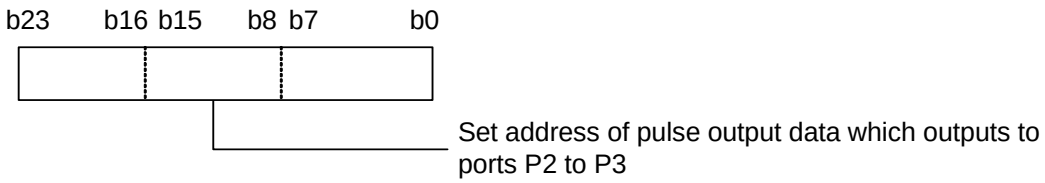
• Set DM1SL register (DMA1 factor select register)



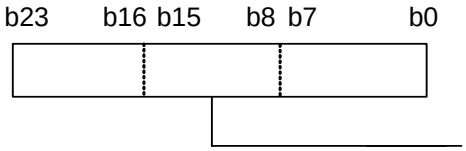
• Set DSA1 register (DMA1 SFR address register)



• Set DRA1 register (DMA1 memory address reload register)

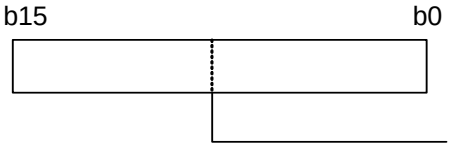


• Set DMA1 register (DMA1 memory address register)



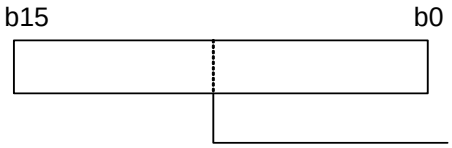
Set address of pulse output data which outputs to ports P2 to P3

• Set DRC1 register (DMA1 transfer count reload register)



Set numbers of pulse output data

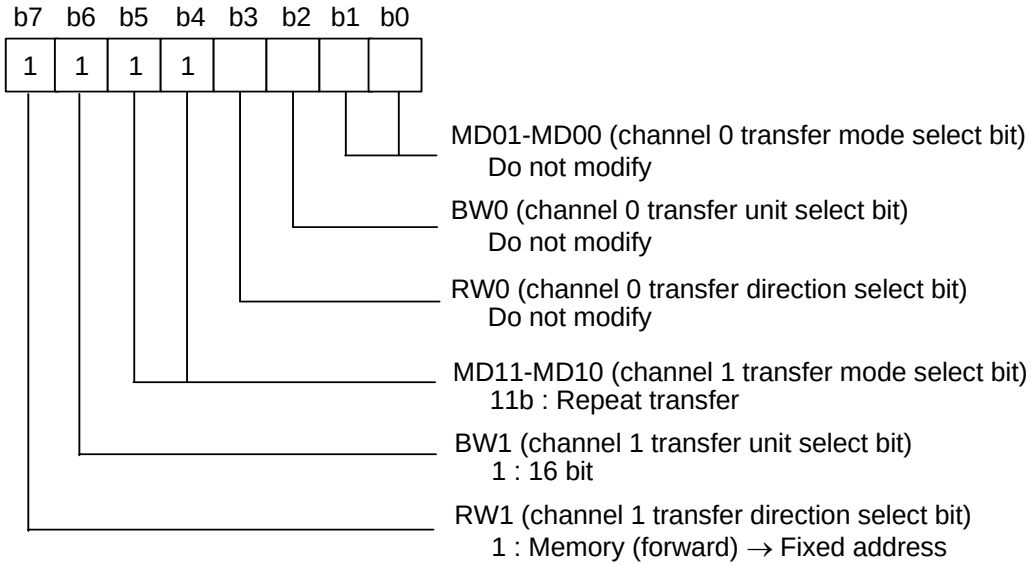
• Set DCT1 register (DMA1 transfer count register)



Set numbers of pulse output data

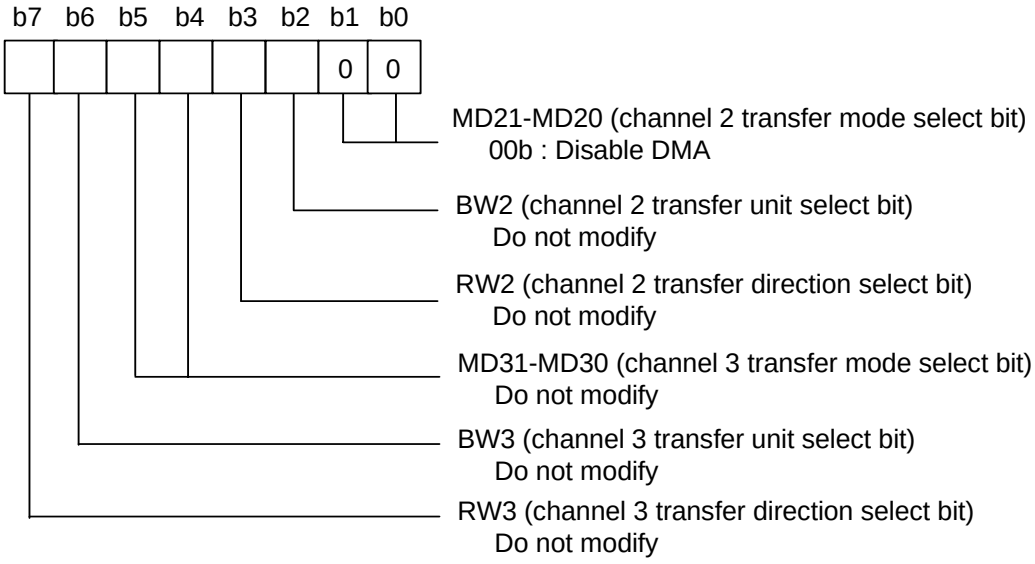
• Insert dummy cycle
Enable DMA after setting the DM1SL register and waiting for 6 cycles by BCLK.
This document shows inserting 6 NOPs waits for 6 cycles.

• Re-set (enable DMA1) DMD0 register (DMA mode register 0)

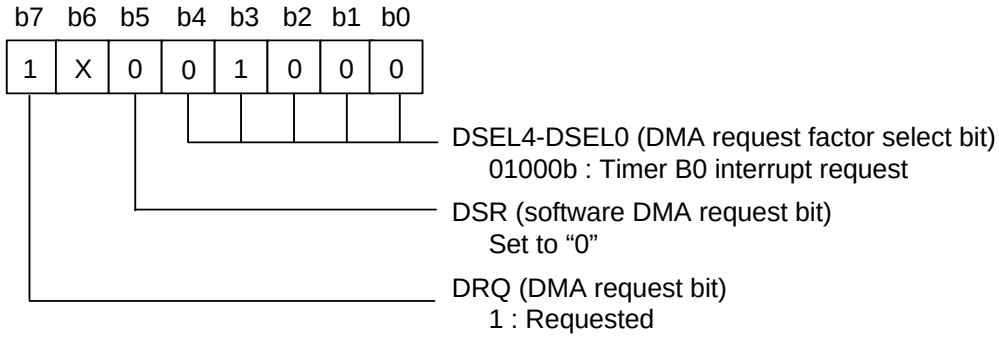


(4) DMA2 setting

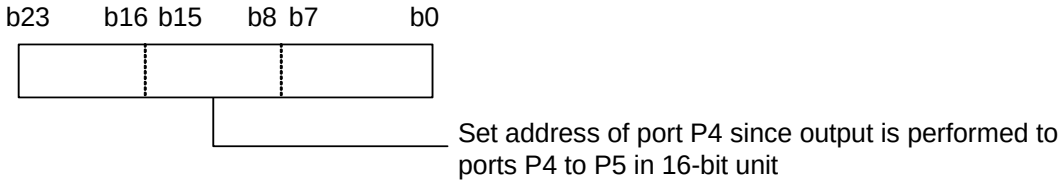
• Set (disable DMA2) DMD1 register (DMA mode register 1)



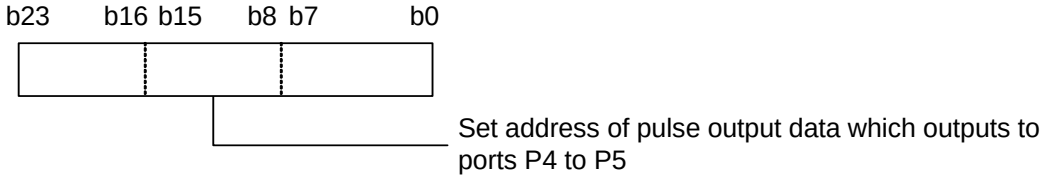
• Set DM2SL register (DMA2 factor select register)



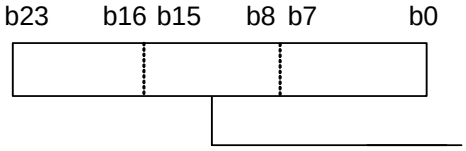
• Set DSA2 register (DMA2 SFR address register)



• Set DRA2 register (DMA2 memory address reload register)

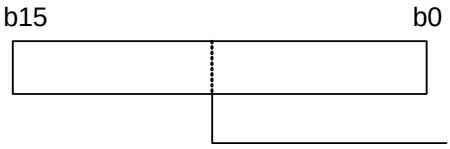


• Set DMA2 register (DMA2 memory address register)



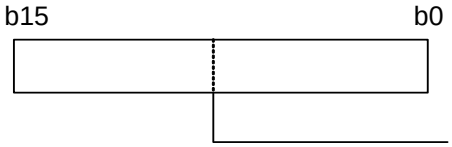
Set address of pulse output data which outputs to ports P4 to P5

• Set DRC2 register (DMA2 transfer count reload register)



Set numbers of pulse output data

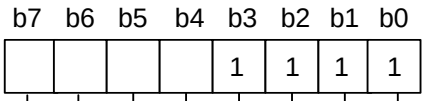
• Set DCT2 register (DMA2 transfer count register)



Set numbers of pulse output data

• Insert dummy cycle
Enable DMA after setting the DM2SL register and waiting for 6 cycles by BCLK.
This document shows inserting 6 NOPs waits for 6 cycles.

• Re-set (enable DMA2) DMD1 register (DMA mode register 1)



MD21-MD20 (channel 2 transfer mode select bit)
11b : Repeat transfer

BW2 (channel 2 transfer unit select bit)
1 : 16 bit

RW2 (channel 2 transfer direction select bit)
1 : Memory (forward) → Fixed address

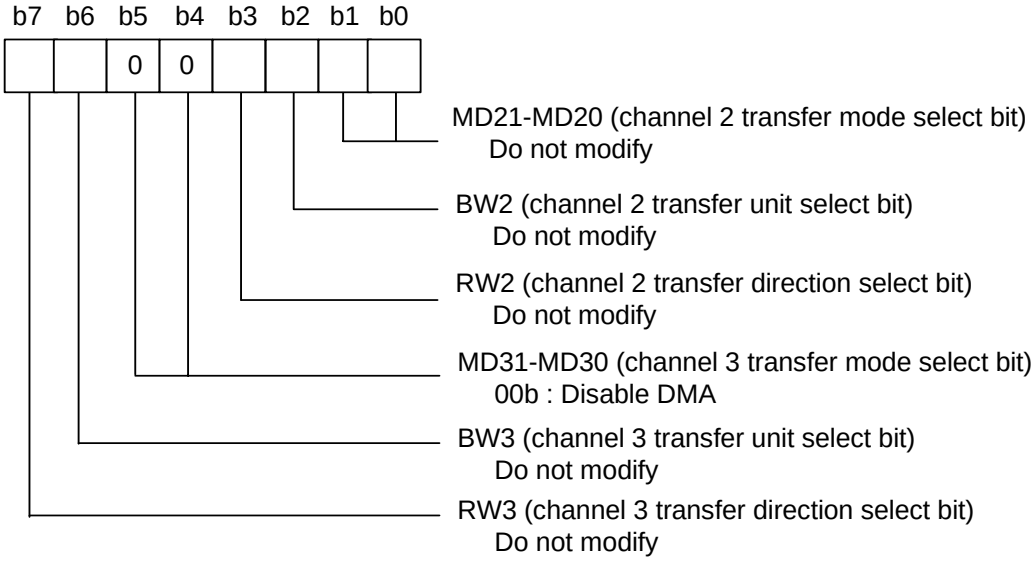
MD31-MD30 (channel 3 transfer mode select bit)
Do not modify

BW3 (channel 3 transfer unit select bit)
Do not modify

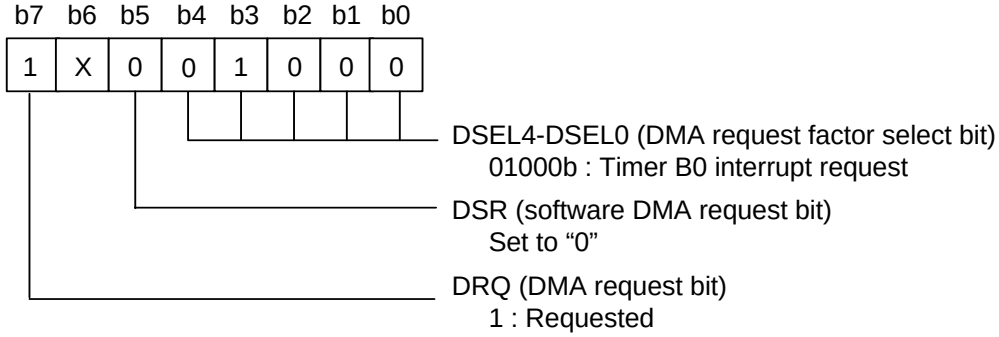
RW3 (channel 3 transfer direction select bit)
Do not modify

(5) DMA 3 setting

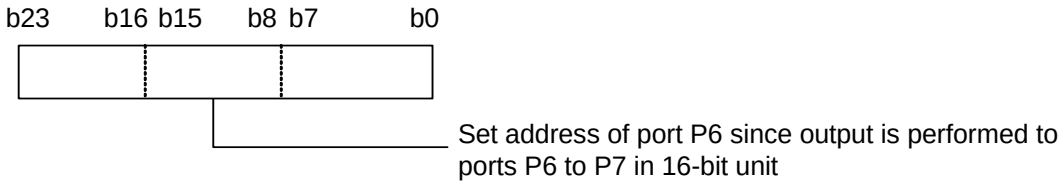
• Set (disable DMA3) DMD1 register (DMA mode register 1)



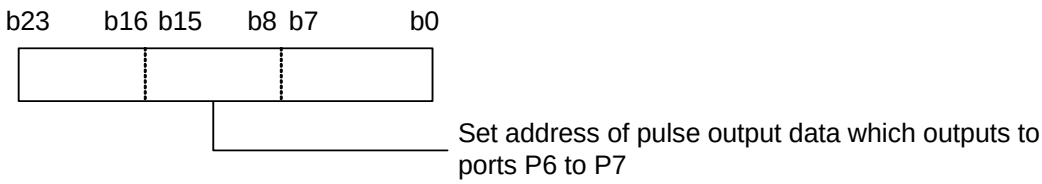
• Set DM3SL register (DMA3 factor select register)



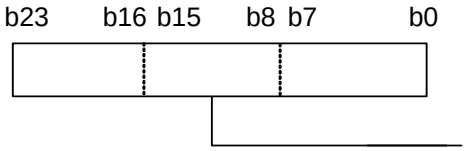
• Set DSA3 register (DMA3 SFR address register)



• Set DRA3 register (DMA3 memory address reload register)

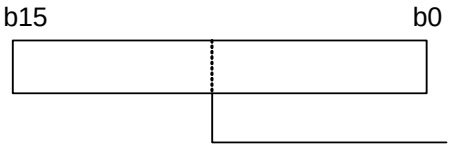


• Set DMA3 register (DMA3 memory address register)



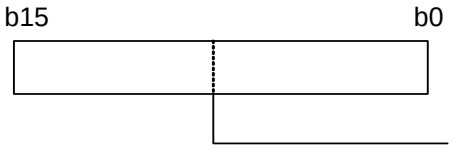
Set address of pulse output data which outputs to ports P6 to P7

• Set DRC3 register (DMA3 transfer count reload register)



Set numbers of pulse output data

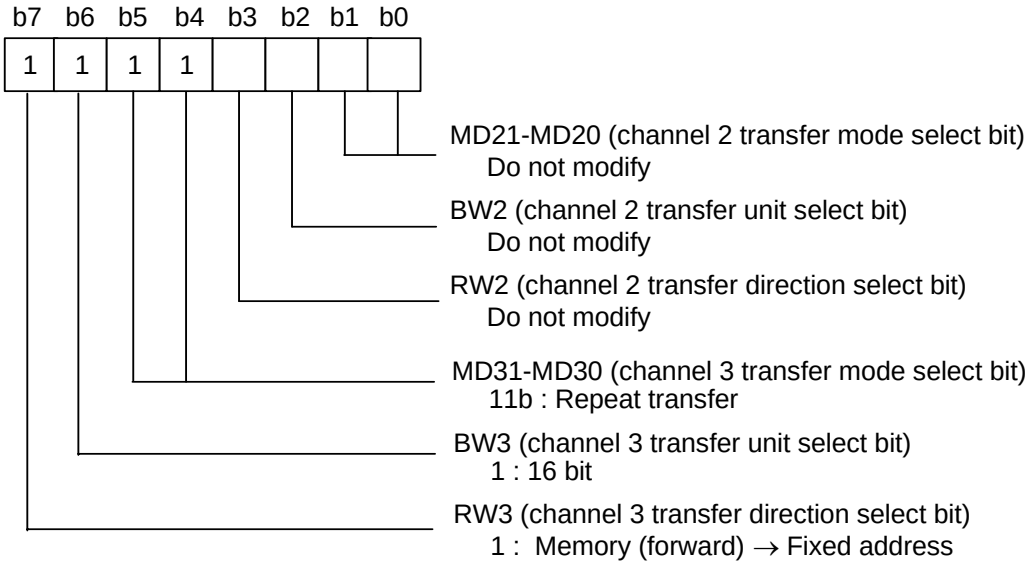
• Set DCT3 register (DMA3 transfer count register)



Set numbers of pulse output data

• Insert dummy cycle
Enable DMA after setting the DM3SL register and waiting for 6 cycles by BCLK.
This document shows inserting 6 NOPs waits for 6 cycles.

• Re-set (enable DMA3) DMD1 register (DMA mode register 1)



(6) Set a port for PWM output to an output port

• Set default value to P0 to P7 registers

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	0	0	0

Pi_j (i=0 to 7, j=0 to 7) (Port Pi_j bit)
0 : Output "L" as default value

• Set to output port in PD0 to PD7 registers (port PD0 to PD7 direction register)

b7	b6	b5	b4	b3	b2	b1	b0
1	1	1	1	1	1	1	1

PDi_j (i=0 to 7, j=0 to 7) (Port Pi_j direction bit)
1 : Output mode

(7) Set an interrupt to enable (I flag = "1")

(8) Timer B0 starts

• Set TABSR register (count start flag)

b7	b6	b5	b4	b3	b2	b1	b0
		1					

TB0S (Timer B0 count start flag)
1 : Timer B0 count starts

(9) Timer B0 interrupt process
Modify the value in the TB0 register (Timer B0 register) to modify PWM pulse output width.

4. Reference Program

A program example which performs 64 PWM outputs between 100µs to 900µs period using Timer B0 and DMA0 to 3 is shown as follows.

Operation condition : VCC1=VCC2=5.0V, XIN=8MHz, PLL=quadruple, f1=32MHz

```
/***********************************************************************/
/*                                                                     */
/*  M32C/85 Group Program Collection                                   */
/*                                                                     */
/*  FILE NAME : rjj05b0579_src.c                                       */
/*  CPU       : M32C/85 Group                                          */
/*  FUNCTION  : The sample program of the 64ch multiplex PWM output    */
/*            using DMAC.                                              */
/*  HISTORY   : 2004.09.15 Ver 1.00                                    */
/*                                                                     */
/*  Copyright (C) 2004. Renesas Technology Corp.                       */
/*  Copyright (C) 2004. Renesas Solutions Corp.                        */
/*  All right reserved.                                                */
/*                                                                     */
/***********************************************************************/

/*****************************************/
/*    include file                       */
/*****************************************/
#include "sfr32c8586.h"     // Special Function Register Header File

/*****************************************/
/*    Function declaration               */
/*****************************************/
void tb0_int(void);         // TB0 interrupt routine

/*****************************************/
/*    Global variable declaration        */
/*****************************************/
                    // PWM ch0-ch15 output data.
const unsigned short     pwm0_data[8] =
      {0x0101, 0x0202, 0x0404, 0x0808, 0x1010, 0x2020, 0x4040, 0x8080 };
const unsigned short     pwm1_data[8] =
      {0x0303, 0x0606, 0x0c0c, 0x1818, 0x3030, 0x6060, 0xc0c0, 0x8181 };
const unsigned short     pwm2_data[8] =
      {0x0707, 0x0e0e, 0x1c1c, 0x3838, 0x7070, 0xe0e0, 0xc1c1, 0x8383 };
const unsigned short     pwm3_data[8] =
      {0x0f0f, 0x1e1e, 0x3c3c, 0x7878, 0xf0f0, 0xe1e1, 0xc3c3, 0x8787 };

unsigned short           tb_value;      // TB0 current value

/*****************************************/
/*    #define declaration                */
/*****************************************/
#define TB_INI_VALUE    3200-1   // TB0 initial value
#define TB_MAX_VALUE    32000-1  // TB0 maximum value
#define TB_UP_VALUE     3200     // TB0 incremental value

/*****************************************/
/*    DMAC register declaration          */
/*****************************************/
```

```
                        // CPU internal registor
unsigned short  dmd0;
#pragma DMAC    dmd0    DMD0    // DMD0(DMA mode register0)
unsigned short  dct0;
#pragma DMAC    dct0    DCT0    // DCT0(DMA0 transfer count register)
unsigned short  drc0;
#pragma DMAC    drc0    DRC0    // DRC0(DMA0 transfer count reload register)
void _far       *dma0;
#pragma DMAC    dma0    DMA0    // DMA0(DMA0 memory address register)
void _far       *dsa0;
#pragma DMAC    dsa0    DSA0    // DSA0(DMA0 SFR address register)
void _far       *dra0;
#pragma DMAC    dra0    DRA0    // DRA0(DMA0 memory address reload register)

unsigned short  dmd1;
#pragma DMAC    dmd1    DMD1    // DMD1(DMA mode register1)
unsigned short  dct1;
#pragma DMAC    dct1    DCT1    // DCT1(DMA1 transfer count register)
unsigned short  drc1;
#pragma DMAC    drc1    DRC1    // DRC1(DMA1 transfer count reload register)
void _far       *dma1;
#pragma DMAC    dma1    DMA1    // DMA1(DMA1 memory address register)
void _far       *dsa1;
#pragma DMAC    dsa1    DSA1    // DSA1(DMA1 SFR address register)
void _far       *dra1;
#pragma DMAC    dra1    DRA1    // DRA1(DMA1 memory address reload register)

/*************************************/
/*    SFR declaration                */
/*************************************/
#pragma ADDRESS     plc0_w      0026H   // PLL control register 0 & 1
unsigned short      plc0_w;

/*****************************************/
/*    Main Program                       */
/*****************************************/
void main(void)
{
   short   dmd0_tmp;                // DMD0 register temp
   short   dmd1_tmp;                // DMD1 register temp
   short   pll_wait;                // PLL wait counter

   prcr = 3;
   plc0_w = 0x0254;                  // PLL clock = Main clock x4.
   plc0   = 0xd4;                   // Start PLL.
                            // It waits until a PLL clock is stabilized.
   for (pll_wait=0;pll_wait<4500;pll_wait++);
   cm17 = 1;                        // Main clock = PLL clock.
   mcd = 0x12;                       // Set main-clock no division mode.
   prcr = 0;

   // A setup a TB0(For DMA request cause).
   tb0mr = 0x00;                     // Set TB0MR register.
                            //  <TMOD1-0> : timer mode
                            //  <TCK1-0>  : f1

   tb_value = TB_INI_VALUE;
   tb0 = tb_value;                  // Set TB0 register.
```

```
                        //  Pulse output cycle=1ms(XIN=32MHz)

tb0ic = 6;                  // Set TB0 interrupt priority level = 6.

// A setup of DMA0.
dmd0_tmp =  dmd0;               // DMA0 inhibit(DMD0)
dmd0_tmp &= 0x00fc;               //  <MD01-00> : DMA0 inhibit
dmd0     =  dmd0_tmp;             //

dm0sl = 0x88;                   // Set DM0SL register.
                        //  <DSEL4-0> : TB0
                        //  <DRQ>     : DMA requested

dsa0  = &p0;                    // Set DSA0 register.
dma0  = &pwm0_data[0];            // Set DMA0 register.
dra0  = &pwm0_data[0];            // Set DRA0 register.
dct0  = 8;                  // Set DCT0 register.
drc0  = 8;                  // Set DRC0 register.
                        // Dummy cycle insertion
asm("NOP    ");                 //  It waits by 6 cycles by BCLK.
asm("NOP    ");
asm("NOP    ");
asm("NOP    ");
asm("NOP    ");
asm("NOP    ");

dmd0_tmp |= 0x0f;               // DMA0 permission(DMD0)
                        //  <MD01-00> : repeat transfer
                        //  <BW0>     : 16bit
                        //  <RW0>     : Memory to Fixed address
dmd0     = dmd0_tmp;

// A setup of DMA1.
dmd0_tmp =  dmd0;               // DMA1 inhibit(DMD0)
dmd0_tmp &= 0x00cf;               //  <MD11-10> : DMA1 inhibit
dmd0     =  dmd0_tmp;             //

dm1sl = 0x88;                   // Set DM1SL register.
                        //  <DSEL4-0> : TB0
                        //  <DRQ>     : DMA requested

dsa1  = &p2;                    // Set DSA1 register.
dma1  = &pwm1_data[0];            // Set DMA1 register.
dra1  = &pwm1_data[0];            // Set DRA1 register.
dct1  = 8;                  // Set DCT1 register.
drc1  = 8;                  // Set DRC1 register.
                        // Dummy cycle insertion
asm("NOP    ");                 //   It waits by 6 cycles by BCLK.
asm("NOP    ");
asm("NOP    ");
asm("NOP    ");
asm("NOP    ");
asm("NOP    ");

dmd0_tmp |= 0xf0;               // DMA1 permission(DMD0)
                        //  <MD11-10> : repeat transfer
                        //  <BW1>     : 16bit
                        //  <RW1>     : Memory to Fixed address
```

```
dmd0     = dmd0_tmp;

// (3) A setup of DMA2.
dmd1_tmp =  dmd1;               // DMA2 inhibit(DMD1)
dmd1_tmp &= 0x00fc;               //  <MD21-20> : DMA2 inhibit
dmd1     =  dmd1_tmp;           //


asm("mov.b  #088h, _dm2sl_addr");   // Set DM2SL register.
                        //  <DSEL4-0> : TB0
                        //  <DRQ>     : DMA requested
asm("fclr   I");                // Interrupt disabled.
asm("fset   B");                // Register-bank1 enable
asm("ldc    #_p4_addr, sb");        // Set DSA2(SB) regisster.
asm("ldc    #_pwm2_data, svp");     // Set DRA2(SVP) register.
asm("mov.l  #_pwm2_data, a0");      // Set DMA2(A0) register.
asm("mov.w  #8, r2");               // Set DRC2(R2) register.
asm("mov.w  #8, r0");               // Set DCT2(R0) register.
asm("fclr   B");                // Register-bank1 disable
                        // Dummy cycle insertion
asm("NOP    ");                 //   It waits by 6 cycles by BCLK.
asm("NOP    ");
asm("NOP    ");
asm("NOP    ");
asm("NOP    ");
asm("NOP    ");

dmd1_tmp |= 0x0f;               // DMA2 permission(DMD1)
                        //  <MD21-20> : repeat transfer
                        //  <BW0>     : 16bit
                        //  <RW0>     : Memory to Fixed address
dmd1     = dmd1_tmp;


// (4) A setup of DMA3.
dmd1_tmp =  dmd1;               // DMA3 inhibit(DMD1)
dmd1_tmp &= 0x00cf;               //  <MD31-30> : DMA3 inhibit
dmd1     =  dmd1_tmp;           //


asm("mov.b  #088h, _dm3sl_addr");   // A setup of DM3SL(DRQ=1, TA1)
                        //  <DSEL4-0> : TB0
                        //  <DRQ>     : DMA requested
asm("fclr   I");                // Interrupt disabled.
asm("fset   B");                // Register-bank1 enable
asm("ldc    #_p6_addr, fb");        // Set DSA3(FB) register.
asm("ldc    #_pwm3_data, vct");     // Set DRA3(VCT) register.
asm("mov.l  #_pwm3_data, a1");      // Set DMA3(A1) register.
asm("mov.w  #8, r3");               // Set DRC3(R3) register.
asm("mov.w  #8, r1");               // Set DCT3(R1) register.
asm("fclr   B");                // Register-bank1 disable
                        // Dummy cycle insertion
asm("NOP    ");                 //   It waits by 6 cycles by BCLK.
asm("NOP    ");
asm("NOP    ");
asm("NOP    ");
asm("NOP    ");
asm("NOP    ");
```

```
    dmd1_tmp |= 0xf0;                   // DMA3 permission(DMD1)
                                //  <MD11-10> : repeat transfer
                                //  <BW1>     : 16bit
                                //  <RW1>     : Memory to Fixed address
    dmd1      = dmd1_tmp;

    // A setup a Port(For PWM output).
    p0  = 0;
    p1  = 0;
    p2  = 0;
    p3  = 0;
    p4  = 0;
    p5  = 0;
    p6  = 0;
    p7  = 0;
    pd0 = 0xff;                     // P0 is output port.
    pd1 = 0xff;                     // P1 is output port.
    pd2 = 0xff;                     // P2 is output port.
    pd3 = 0xff;                     // P3 is output port.
    pd4 = 0xff;                     // P4 is output port.
    pd5 = 0xff;                     // P5 is output port.
    pd6 = 0xff;                     // P6 is output port.
    pd7 = 0xff;                     // P7 is output port.

    asm("fset i");                  // Interrupt enabled.

    tb0s = 1;                       // TB0 start.

    while(1);

}

/****************************************/
/*  TB0 interrupt routine               */
/****************************************/
#pragma INTERRUPT/B tb0_int
// "/B" = Instead of saving the registers to the stack,
//        you can switch to the alternate registers.

void tb0_int(void)
{
                                // A next timer value is calculated.
    tb_value += TB_UP_VALUE;
    if (tb_value >= TB_MAX_VALUE) tb_value = TB_INI_VALUE;

    tb0 = tb_value;                 // Next timer value set.

}
```

5. Reference Document

Hardware Manual
M32C/85 Group Hardware Manual
Use the latest version on the Renesas Technology Corporation Semiconductor Home Page

6. Home Page and E-mail Support

Renesas Technology Corporation Semiconductor Home Page
http://www.renesas.com

M16C Family MCU Technical E-mail Support
E-mail: support_apl@renesas.com

REVISION HISTORY	M32C/85 Group Application Example of Multiple-Channel PWM Using DMAC

Rev.	Date	Description	
		Page	Summary
1.00	Oct 15, 2004	–	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 non-flammable 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.