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/62

Using the M16C/62 DMAC in Forward Source Mode

1.0 Abstract

The following article introduces and shows an example of how to use the DMAC function of the M16C/62 with a forward counting source address and fixed destination address.

2.0 Introduction

The Renesas M16C/62 is a 16-bit MCU based on the M16C CPU core with 256 KB of user Flash. The MCU has two DMAC (Direct Memory Access Controller) channels that allow data to be transferred from a source memory location to a destination memory location without using the CPU. The DMAC utilizes the same internal address and data busses as the CPU yet is given a higher priority to the data bus than the CPU. This method of DMAC and CPU bus arbitration is termed a cycle stealing method.

Each DMAC controller is capable of transferring data to or from a fixed address to any other address in the 1Mbyte address space. The DMAC controllers can automatically transfer 128k bytes of data using word (16-bit) transfers, or 64k bytes of data using byte (8-bit) transfers. The source or destination address can also be auto-incremented. DMAC transfers can be initiated by an interrupt request signal or by manually writing to the software DMA request bit. When requests are initiated by an interrupt request signal, neither the interrupt enable flag (I flag) nor the interrupt priority level affects the DMA transfers.

3.0 DMAC with Forward Source, Fixed Destination

In the forward source counting address, fixed destination address mode, the DMAC controller will transfer bytes or words from an incrementing source address (that increments after each transfer) to a fixed destination address. The transfers can be either bytes or words. Loading a value into the transfer count register controls the number of automated transfers. Transfers will continue to occur each time the DMAC trigger event occurs until the transfer register underflows, therefore the number loaded into the register should be 1 less than the number of transfers desired. A control register bit determines whether each transfer is a byte or word of data. When the DMAC controller is configured to perform a single transfer cycle, the DMAC becomes disabled after the transfer register underflows. In the repeat mode the Source Pointer register and the Transfer Counter register are reloaded with their initial values after the Transfer Counter register underflows and the DMAC remains active. Therefore, in the repeat mode, transfers will occur each time a trigger event occurs until the DMA enable bit is set inactive ("0").

4.0 Configuring the DMAC for Forward Source, Fixed Destination

To configure a DMAC channel, the following choices must be configured (the configurations for this example are shown in parentheses):

1. Select the DMA request cause (UART0 receive interrupt request).
2. Select fixed or forward source (forward source).
3. Select fixed or forward destination (fixed destination).
4. Select 8 or 16-bit transfers (8-bit transfers).
5. Select a single transfer or multiple transfers (single transfer).
6. Select the source address for the transfer (Buffer address in RAM).
7. Select the destination address for the transfer UART0 transmit buffer address).
8. Select the number of bytes to be transferred (10).

The registers that are used to configure and control the DMAC channels are shown in Figure 1.

DMA0 request cause select register

b7	b6	b5	b4	b3	b2	b1	b0
		✕	✕				

Symbol: DM0SL — Address: 03B8₁₆ — When reset: 00₁₆

Bit symbol	Bit name	Function	R	W
DSEL0	DMA request cause select bit	b3 b2 b1 b0 0 0 0 0 : Falling edge of INT0 pin 0 0 0 1 : Software trigger 0 0 1 0 : Timer A0 0 0 1 1 : Timer A1 0 1 0 0 : Timer A2 0 1 0 1 : Timer A3 0 1 1 0 : Timer A4 (DMS=0) /two edges of INT0 pin (DMS=1) 0 1 1 1 : Timer B0 (DMS=0) Timer B3 (DMS=1) 1 0 0 0 : Timer B1 (DMS=0) Timer B4 (DMS=1) 1 0 0 1 : Timer B2 (DMS=0) Timer B5 (DMS=1) 1 0 1 0 : UART0 transmit 1 0 1 1 : UART0 receive 1 1 0 0 : UART2 transmit 1 1 0 1 : UART2 receive 1 1 1 0 : A-D conversion 1 1 1 1 : UART1 transmit	○	○
DSEL1			○	○
DSEL2			○	○
DSEL3			○	○
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be “0”.			–	–
DMS	DMA request cause expansion select bit	0 : Normal 1 : Expanded cause	○	○
DSR	Software DMA request bit	If software trigger is selected, a DMA request is generated by setting this bit to “1” (When read, the value of this bit is always “0”)	○	○

Figure 1 DMA0 Request Cause Select Register

DMA1 request cause select register

b7	b6	b5	b4	b3	b2	b1	b0
		×	×				

Symbol: DM1SL — Address: 03BA16 — When reset: 0016

Bit symbol	Bit name	Function	R	W
DSEL0	DMA request cause select bit	b3 b2 b1 b0 0 0 0 0 : Falling edge of INT1 pin 0 0 0 1 : Software trigger 0 0 1 0 : Timer A0 0 0 1 1 : Timer A1 0 1 0 0 : Timer A2 0 1 0 1 : Timer A3(DMS=0) /serial I/O3 (DMS=1) 0 1 1 0 : Timer A4 (DMS=0) /serial I/O4 (DMS=1) 0 1 1 1 : Timer B0 (DMS=0) /two edges of INT1 (DMS=1) 1 0 0 0 : Timer B1 1 0 0 1 : Timer B2 1 0 1 0 : UART0 transmit 1 0 1 1 : UART0 receive 1 1 0 0 : UART2 transmit 1 1 0 1 : UART2 receive 1 1 1 0 : A-D conversion 1 1 1 1 : UART1 receive	○	○
DSEL1			○	○
DSEL2			○	○
DSEL3			○	○
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be “0”.			–	–
DMS	DMA request cause expansion select bit	0 : Normal 1 : Expanded cause	○	○
DSR	Software DMA request bit	If software trigger is selected, a DMA request is generated by setting this bit to “1” (When read, the value of this bit is always “0”)	○	○

DMAi control register

b7	b6	b5	b4	b3	b2	b1	b0
×	×						

Symbol: DMiCON(i=0,1) — Address: 002C16, 003C16 — When reset: 00000X002

Bit symbol	Bit name	F unction	R	W
DMBIT	Transfer unit bit select bit	0 : 16 bits 1 : 8 bits	○	○
DMASL	Repeat transfer mode select bit	0 : Single transfer 1 : Repeat transfer	○	○
DMAS	DMA request bit (Note 1)	0 : DMA not requested 1 : DMA requested	○	○ (Note 2)
DMAE	DMA enable bit	0 : Disabled 1 : Enabled	○	○
DSD	Source address direction select bit (Note 3)	0 : Fixed 1 : Forward	○	○
DAD	Destination address direction select bit (Note 3)	0 : Fixed 1 : Forward	○	○
Nothing is assigned. In an attempt to write to these bits, write “0”. The value, if read, turns out to be “0”.			–	–

Note 1: DMA request can be cleared by resetting the bit.
Note 2: This bit can only be set to “0”.
Note 3: Source address direction select bit and destination address direction select bit cannot be set to “1” simultaneously.

Figure 2 DMA Control Registers

5.0 Reference

Renesas Technology Corporation Semiconductor Home Page

http://www.renesas.com

E-mail Support

support_apl@renesas.com

Data Sheets

- M16C/62 datasheets, 62aeds.pdf

6.0 Software Code

The example program was written to run on the MSV1632 Starter Kit but could be modified to implement in a user application. The program is written in C (the NC30 Compiler). The program demonstrates using the DMA0 channel to transfer data from a memory buffer to the UART0 transmit buffer. The program performs a single transfer of 10 bytes to UART0. At the completion of the transfer a DMA0 interrupt request is generated. UART0 on the starter kit board is connected to a 9-pin DB connector that can be used to connect to a PC running a terminal program, such as HyperTerminal. With the program running, the data contained in the DATA array will be sent to the terminal program and appear on the PC screen.

```
/************************************************************************
*
*       File Name: dma_fwd_src.c

*
*       Content:  DMAC from a memory buffer to UART transmit buffer
*=======================================================================
*
*       $Log:$
*=======================================================================*
/

#include "sfr62a.h"          /* SFR register definition */

#pragma Interrupt  dma0_isr


// prototypes

void uart_init (void);
void DMA_init (void);


unsigned const char *ptr_string;
unsigned const char data[]= "0123456789" ;
```

```
/***********************************************************************
Name:         main
Parameters:   None
Returns:      None
Description: Initializes the system and then loops forever.
***********************************************************************/
void main()
       {

       ptr_string= &data[0];

       uart_init ();               // initialize UART0 and pre-load first character
                                   // in transmit buffer
       DMA_init ();                // initialize DMA registers
       dmae_dm0con = 1;            // enable DMA transfers
       asm ("fset I");             // enable interrupts
       te_u0c1 = 1;                // enable UART0 transmit

       while (1);                  //loop forever

 }


/***********************************************************************
Name:         DMA_init
Parameters:   None
Returns:      None
Description: Initializes DMA for transfer from forward source to fixed
destinations. Sets DMAC0 for 10 byte transfers from memory to UART0
transmit buffer
***********************************************************************/
void DMA_init(void)
{
       dm0sl =  0x0a;   /* DMA0 trigger select UART0 transmit

                00001010;
                ||||||||------(DSEL0) the four bits (DSEL3-DSEL0) the DMA
                |||||||-------(DSEL1) request cause set for UART0 transmit
                ||||||--------(DESEL2)
                |||||---------(DSEL3)
                ||||----------not used set to 0
                |||-----------not used set to 0
                ||------------(DMS) DMA request cause expansion bit to normal
                |-------------(DSR) set to 1 to generate DMA request if software
                                   trigger selected */

    dm0con = 0X11;          /* DMA0 single transfer, 8 bit mode, forward source,
                                   fixed destination /*
               00010001;
               ||||||||------(DMBIT) transfer unit bit select bit  1 = 8 bits
               |||||||-------(DMASL) repeat transfer mode  0 = single transfer
               ||||||--------(DMAS) DMA request bit can only be set to 0
               |||||---------(DMAE)DMA enable bit  0= disabled
               ||||----------(DSD)source address direction  1 = forward
```

```
                |||-----------(DAD)destination address direction 0 = fixed
                ||------------not used set to 0
                |-------------not used set to 0 */

        dar0 = (unsigned long)&u0tb;        //  set destination register to address
                                            //  of uart0 transmit buffer

        sar0 = (unsigned long)ptr_string;   //  set source register to address of
                                            //  beginning of data buffer

        tcr0 = 0x9;                         //  set transfer counter for 10 transfers
                                            //   (number of transfers -1)

        dm0ic = 0x04;                       //  set interrupt priority for DMA0
                                            //  interrupt to 4
}

/***********************************************************************
Name:         dma0_isr
Parameters:   None
Returns:      None
Description:  This service routine is entered after the completion of the DMA
              transfer
***********************************************************************/
void dma0_isr(void)
{


}

/***********************************************************************
Name:         uart_init
Parameters:   None
Returns:      None
Description: Initializes  uart
***********************************************************************/
void uart_init(void)
{
   int  dummy;

   //  Configure Uart0 for 9600 baud, 8 data bits, 1 stop bit, no parity
        u0mr = 0x05;                // set mode register
        u0c0 = 0x10;                // set control register
        u0brg = 0x67;               // set bit rate generator
                                    //  (16Mhz/16/9600)-1

        u0tb = 0x20;                // Place data in the transmit buffer
                                    // so when the transmit enable bit is set
                                    // later on, this first byte will be sent
                                    // out and cause a UART transmit interrupt
                                    // to occurs triggering the DMAC
        s0tic = 0x00;               // Disable UART0 receive interrupt,

}
```

Keep safety first in your circuit designs!

- 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

- 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.
- 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.
- 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).
- 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.
- 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.
- The prior written approval of Renesas Technology Corporation is necessary to reprint or reproduce in whole or in part these materials.
- 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.
- Please contact Renesas Technology Corporation for further details on these materials or the products contained therein.