# Regarding the change of names mentioned in the document, such as Mitsubishi Electric and Mitsubishi XX, to Renesas Technology Corp.

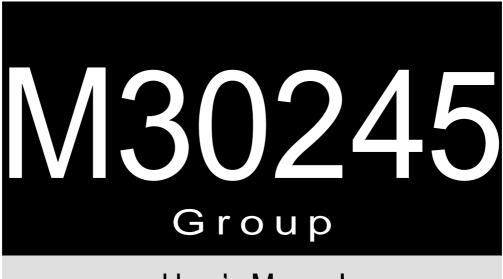
The semiconductor operations of Hitachi and Mitsubishi Electric were transferred to Renesas Technology Corporation on April 1st 2003. These operations include microcomputer, logic, analog and discrete devices, and memory chips other than DRAMs (flash memory, SRAMs etc.) Accordingly, although Mitsubishi Electric, Mitsubishi Electric Corporation, Mitsubishi Semiconductors, and other Mitsubishi brand names are mentioned in the document, these names have in fact all been changed to Renesas Technology Corp. Thank you for your understanding. Except for our corporate trademark, logo and corporate statement, no changes whatsoever have been made to the contents of the document, and these changes do not constitute any alteration to the contents of the document itself.

Note : Mitsubishi Electric will continue the business operations of high frequency & optical devices and power devices.

Renesas Technology Corp. Customer Support Dept. April 1, 2003



### MITSUBISHI 16-BIT SINGLE-CHIP MICROCOMPUTER M16C FAMILY / M16C/20 SERIES



User's Manual

# Preliminary Specifications Chapter 3

## http://www.infomicom.maec.co.jp/indexe.htm

Before using this material, please visit the above website to confirm that this is the most current document available.



Preliminary Specifications Rev. A Revision date: Jan. 24, 2003

#### Keep safety first in your circuit designs!

 Mitsubishi Electric 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 Mitsubishi semiconductor 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 Mitsubishi Electric Corporation or a third party.
- Mitsubishi Electric 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 Mitsubishi Electric Corporation without notice due to product improvements or other reasons. It is therefore recommended that customers contact Mitsubishi Electric Corporation or an authorized Mitsubishi Semiconductor product distributor for the latest product information before purchasing a product listed herein.

The information described here may contain technical inaccuracies or typographical errors. Mitsubishi Electric 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 Mitsubishi Electric Corporation by various means, including the Mitsubishi Semiconductor home page (http://www.mitsubishichips.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. Mitsubishi Electric Corporation assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
- Mitsubishi Electric 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 Mitsubishi Electric Corporation or an authorized Mitsubishi Semiconductor 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 Mitsubishi Electric 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 Mitsubishi Electric Corporation or an authorized Mitsubishi Semiconductor product distributor for further details on these materials or the products contained therein.

#### **REVISION HISTORY**

#### M30245 GROUP USER'S MANUAL

Rev.	Date		Description		
		Page	Summary		
A	1/24/2003	0	First edition		

### Chapter 3

### **Examples of Peripheral Functions Applications**



This chapter presents applications in which peripheral functions built in the M30245 are used. They are shown here as examples. In practical use, make suitable changes and perform sufficient evaluation. For basic use, see Chapter 2 Peripheral Functions Usage.





### [ MEMO ]

Under

3-3



#### 3.1 Long-Period Timers

- Overview In this process, Timer A0 and Timer A1 are connected to make a 16-bit timer with a 16-bit prescaler. Figure 3.1.1 shows the operation timing, Figure 3.1.2 shows the connection diagram, and Figures 3.1.3 and 3.1.4 show the set-up procedure. Use the following peripheral functions:
  - Timer mode of timer A
  - Event counter mode of timer A

#### Specifications

- (1) Set timer A0 to timer mode, and set timer A1 to event counter mode.
- (2) Perform a count on count source f1 using timer A0 to count for 1 ms, and perform a count on timer A0 using timer A1 to count for 1 second.
- (3) Connect a 16-MHz oscillator to XIN.
- Operation (1) Setting the count start flag to "1" causes the counter to begin counting. The counter of timer A0 performs a down count on count source f1.
  - (2) If the counter of timer A0 underflows, the counter reloads the content of the reload register and continues counting. At this time, the timer A0 interrupt request bit goes to "1". The counter of timer A1 performs a down count on underflows in timer A0.
  - (3) If the counter of timer A1 underflows, the counter reloads the content of the reload register and continues counting. At this time, the timer A1 interrupt request bit goes to "1".

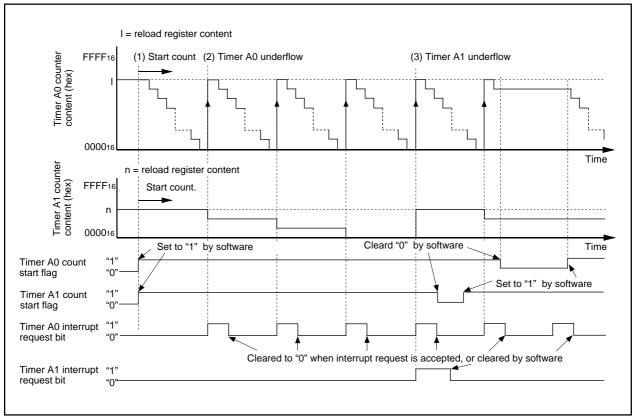


Figure 3.1.1. Operation timing of long-period timers





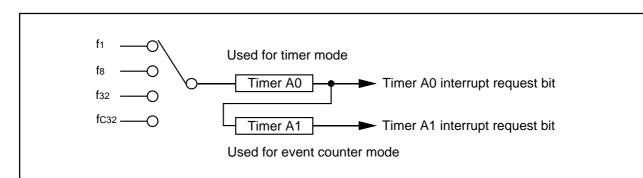


Figure 3.1.2. Connection diagram of long-period timers





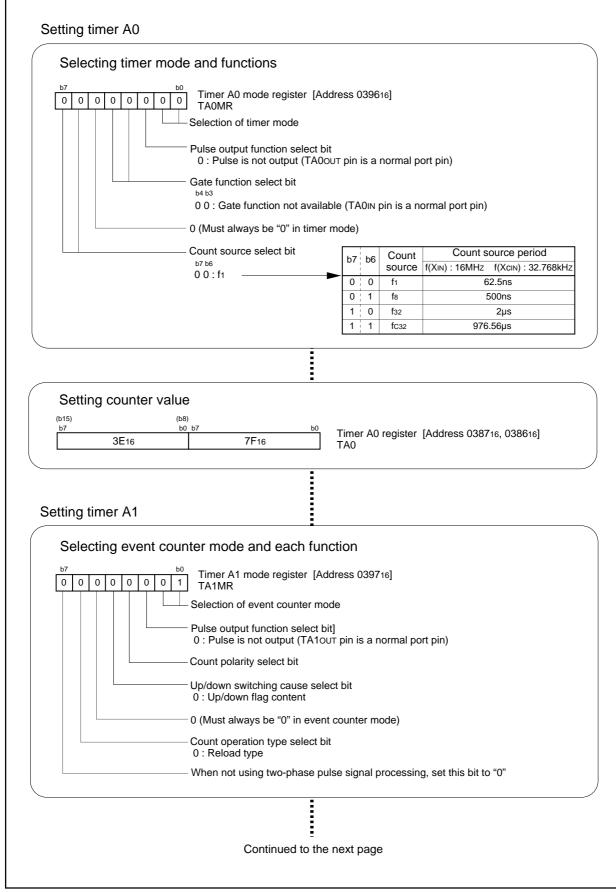


Figure 3.1.3. Set-up procedure of long-period timers (1)





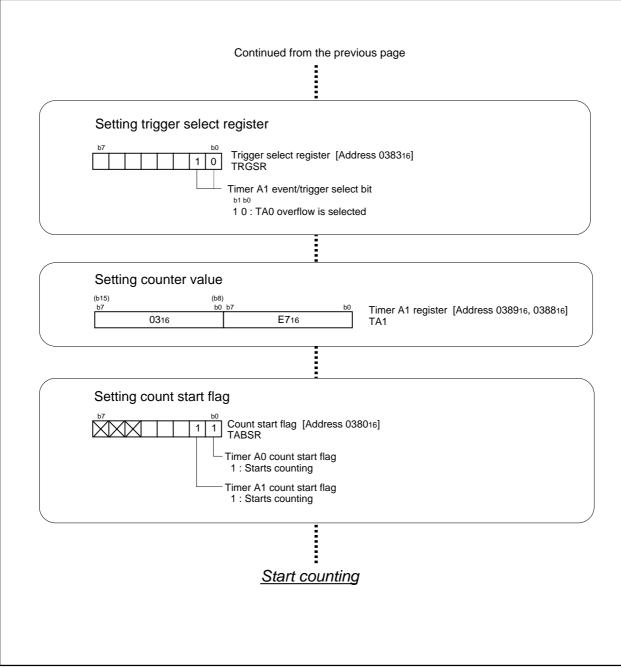


Figure 3.1.4. Set-up procedure of long-period timers (2)





#### 3.2 Variable-Period Variable-Duty PWM Output

Overview In this process, Timer A0 and A1 are used to generate variable-period, variable-duty PWM output. Figure 3.2.1 shows the operation timing, Figure 3.2.2 shows the connection diagram, and Figures 3.2.3 and 3.2.4 show the set-up procedure.

Use the following peripheral functions:

- Timer mode of timer A
- One-shot timer mode of timer A

#### Specifications

- (1) Set timer A0 in timer mode, and set timer A1 in one-shot timer mode with pulse-output function.
- (2) Set 1 ms, the PWM period, to timer A0. Set 500 µs, the width of PWM "H" pulse, to timer A1. Both timer A0 and timer A1 use f1 for the count source.
- (3) Connect a 16-MHz oscillator to XIN.
- Operation (1) Setting the count start flag to "1" causes the counter of timer A0 to begin counting. The counter of timer A0 performs a down count on count source f1.
  - (2) If the counter of timer A0 underflows, the counter reloads the content of the reload register and continues counting. At this time, the timer A0 interrupt request bit goes to "1".
  - (3) An underflow in timer A0 triggers the counter of timer A1 and causes it to begin counting. When the counter of timer A1 begins counting, the output level of the TA10UT pin goes to "H".
  - (4) As soon as the count of the counter of timer A1 becomes "000016", the output level of TA10UT pin goes to "L", and the counter reloads the content of the reload register and stops counting. At the same time, the timer A1 interrupt request bit goes to "1".





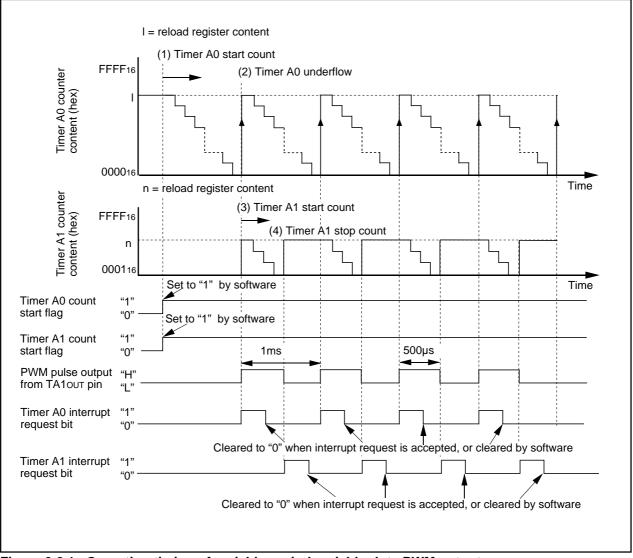


Figure 3.2.1. Operation timing of variable-period variable-duty PWM output

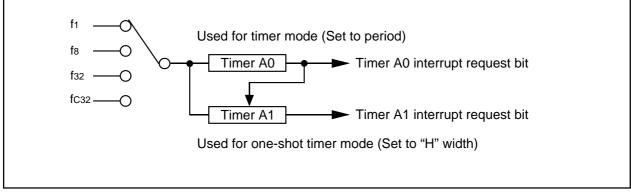


Figure 3.2.2. Connection diagram of variable-period variable-duty PWM output





Setting timer A0 Selecting timer mode and functions Timer A0 mode register [Address 039616] 0 0 0 0 0 0 0 **TA0MR** Selection of timer mode Pulse output function select bit 0 : Pulse is not output (TA0out pin is a normal port pin) Gate function select bit b4 b3 00: Gate function not available (TA0IN pin is a normal port pin) 0 (Must always be "0" in timer mode) Count source select bit Count source period Count b7 b6 <sup>b7 b6</sup> 00:f1 source f(XIN): 16MHz f(XCIN): 32.768kHz 0 0 f1 62.5ns 0 f8 1 500ns 0 f32 1 2µs 1 976.56µs 1 fC32 Setting counter value (b15) (b8 b0 b Timer A0 register [Address 038716, 038616] 3E16 **7F**16 TA0 Setting timer A1 Selecting one-shot timer mode and functions b7 b0 Timer A1 mode register [Address 039716] 0 0 0 1 0 1 1 0 TA1MR Selection of one-shot timer mode Pulse output function select bit 1 : Pulse is output External trigger select bit (Invalid when choosing timer's overflow as trigger) Trigger select bit 1 : Selected by event/trigger select register 0 (Must always be "0" in one-shot timer mode) Count source select bit Count Count source period b7 b6 b7 b6 source f(XIN): 16MHz f(XCIN): 32.768kHz 00:f1 0 0 f1 62.5ns 0 1 f8 500ns 1 0 f32 2µs 1 1 976.56µs fC32 Continued to the next page

Figure 3.2.3. Set-up procedure of variable-period variable-duty PWM output (1)





Preliminary Specifications REV.A Specifications in this manual are tentative and subject to change.

**Timer A Applications** 

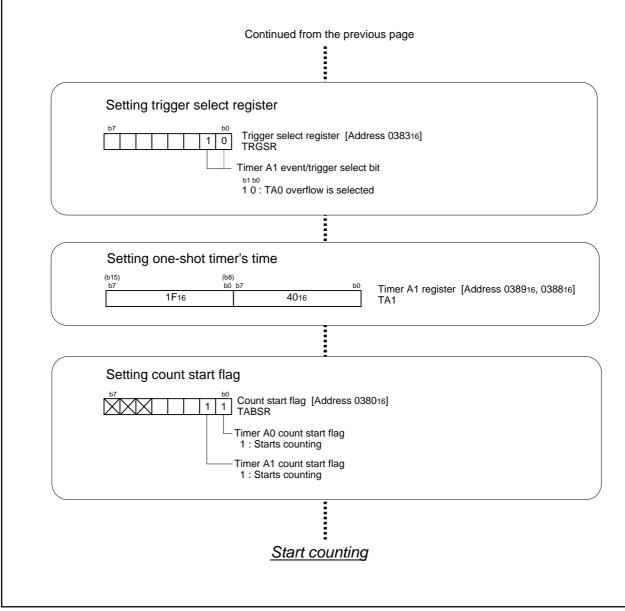


Figure 3.2.4. Set-up procedure of variable-period variable-duty PWM output (2)





#### 3.3 Delayed One-Shot Output

- Overview The following are steps of outputting a pulse only once after a specified elapse since an external trigger is input. Figure 3.3.1 shows the operation timing, Figure 3.3.2 shows the connection diagram, and Figures 3.3.3 and 3.3.4 show the set-up procedure.
  - Use the following peripheral function:
  - One-shot timer mode of timer A

#### Specifications

- (1) Set timer A0 in one-shot timer mode, and set timer A1 in one-shot timer mode with pulseoutput function.
- (2) Set 1 ms, an interval before a pulse is output, in timer A0; and set 50 μs, a pulse width, in timer A1. Both timer A0 and timer A1 use f1 for the count source.
- (3) Connect a 16-MHz oscillator to XIN.
- Operation (1) Setting the trigger select bit to "1" and setting the count start flag to "1" enables the counter of timer A0 to count.
  - (2) If an effective edge, selected by use of the external trigger select bit, is input to the TA0IN pin, the counter begins a down count. The counter of timer A0 performs a down count on count source f1.
  - (3) As soon as the counter of timer A0 becomes "000016", the counter reloads the content of the reload register and stops counting. At this time, the timer A0 interrupt request bit goes to "1".
  - (4) An underflow in timer A0 triggers the counter of timer A1 and causes it to begin counting. When timer A1 begins counting, the output level of the TA10∪T pin goes to "H".
  - (5) As soon as the counter of timer A1 becomes "000016", the output level of the TA10UT pin goes to "L", the counter reloads the content of the reload register, and stops counting. At this time, timer A1 interrupt request bit goes to "1".







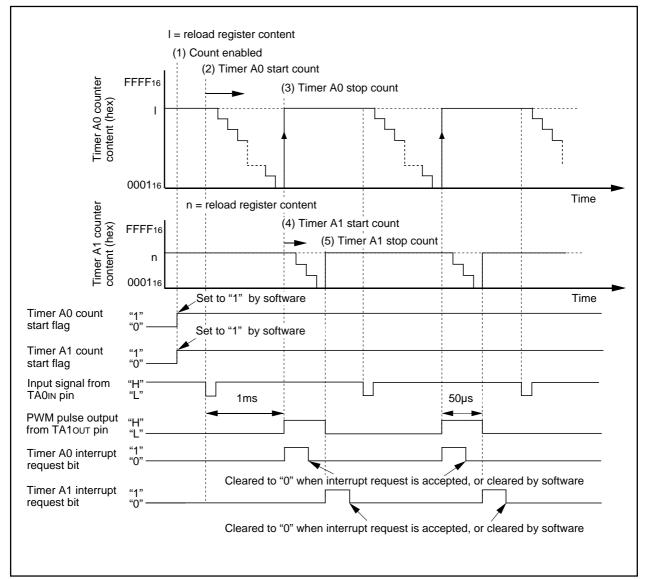


Figure 3.3.1. Operation timing of delayed one-shot output

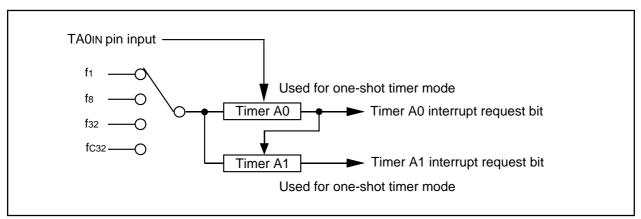


Figure 3.3.2. Connection diagram of delayed one-shot output





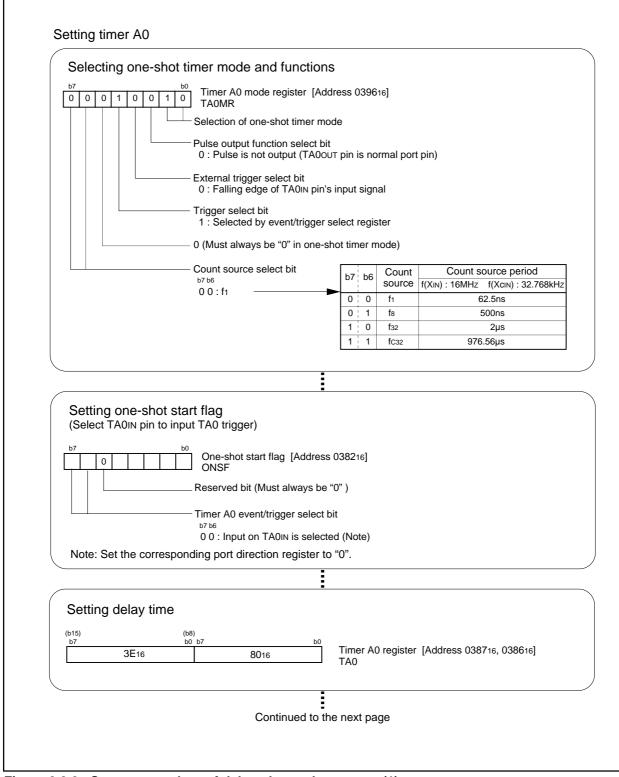
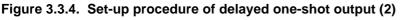


Figure 3.3.3. Set-up procedure of delayed one-shot output (1)





Continued from the previous page Setting timer A1 Selecting one-shot timer mode and functions Timer A1 mode register [Address 039716] TA1MR 0 0 0 1 0 1 0 1 Selection of one-shot timer mode Pulse output function select bit 1 : Pulse is output (TA1out pin is pulse output pin) External trigger select bit Invalid when choosing timer's overflow Trigger select bit 1 : Selected by event/trigger select register 0 (Must always be "0" in one-shot timer mode) Count source select bit Count source period Count b7<sup>1</sup> b6 b7 b6 source f(XIN): 16MHz f(XCIN): 32.768kHz 00:f1 0 0 f1 62.5ns 500ns 0 1 f8 1 0 f32 2µs 976.56µs 1 1 fC32 Ē Setting trigger select register (Set timer A0 to trigger timer A1) Trigger select register [Address 038316] TRGSR 1 0 Timer A1 event/trigger select bit b1 b0 1 0 : TA0 overflow is selected Setting one-shot timer's time (b15) b0 b7 Timer A1 register [Address 038916, 038816] 0316 2016 TA1 Setting count start flag Count start flag [Address 038016] TABSR 1 1 Timer A0 count start flag 1 : Starts counting Timer A1 count start flag 1 : Starts counting ÷ Start counting







#### 3.4 Buzzer Output

- Overview The timer mode is used to make the buzzer ring. Figure 3.4.1 shows the operation timing, and Figure 3.4.2 shows the set-up procedure. Use the following peripheral function:
  - The pulse-outputting function in timer mode of timer A.

Specifications

- (1) Sound a 2-kHz buzz beep by use of timer A0.
- (2) Effect pull-up in the relevant port by use of a pull-up resistor. When the buzzer is off, set the port high-impedance, and stabilize the potential resulting from pulling up.
- (3) Connect a 16-MHz oscillator to XIN.
- Operation (1) The microcomputer begins performing a count on timer A0. Timer A0 has disabled interrupts.
  - (2) The microcomputer begins pulse output by setting the pulse output function select bit to "Pulse output effected". P70 changes into TA00UT pin and outputs 2-kHz pulses.
  - (3) The microcomputer stops outputting pulses by setting the pulse output function select bit to "Pulse output not effected". P70 goes to an input pin, and the output from the pin becomes high-impedance.

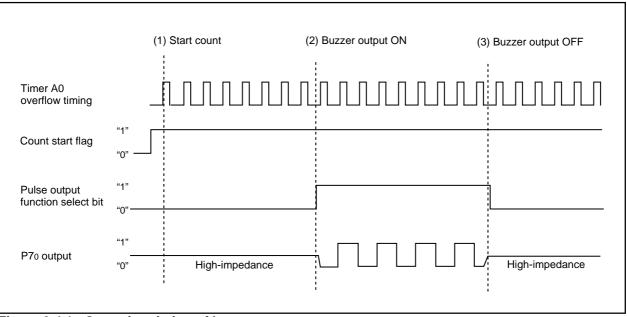


Figure 3.4.1. Operation timing of buzzer output





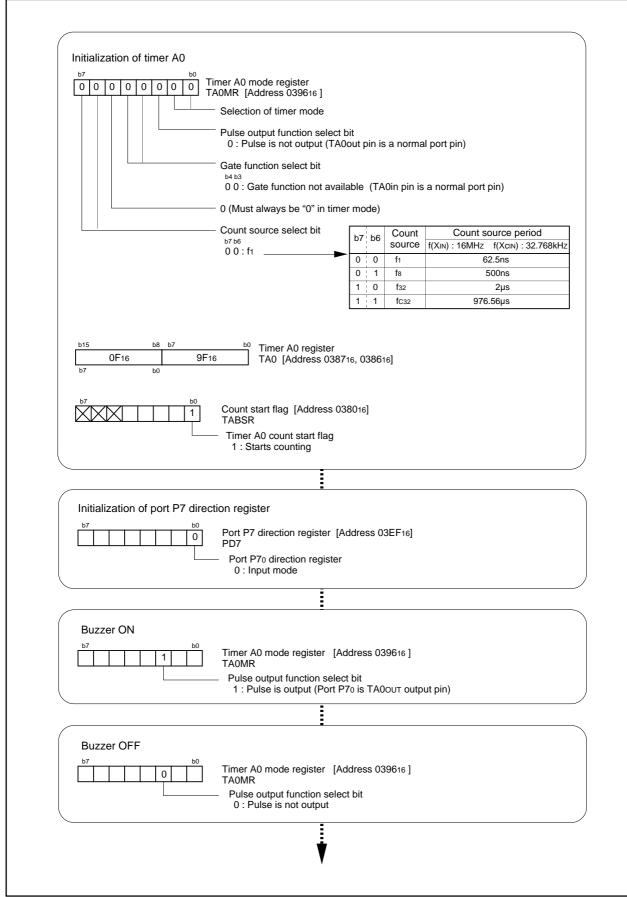


Figure 3.4.2. Set-up procedure of buzzer output





#### 3.5 Solution for External Interrupt Pins Shortage

- The following are solution for external interrupt pins shortage. Figure 3.5.1 shows the set-up Overview procedure.
  - Use the following peripheral function:
  - Event counter mode of timer A

#### Specifications

- (1) Inputting a falling edge to the TAOIN pin generates a timer A0 interrupt.
- Operation (1) Set timer A0 to event counter mode, set timer to "0", and set interrupt priority levels in timer A0. (2) Inputting a falling edge to the TA0IN pin generates a timer A0 interrupt.

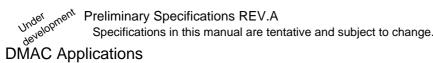




Initialization of timer A0	
	Timer A0 mode register TA0MR [Address 039616]
	Selection of event counter mode
	<ul> <li>Pulse output function select bit</li> <li>0 : Pulse is not output (TA0out pin is a normal port pin)</li> </ul>
	<ul> <li>Count polarity select bit</li> <li>Counts external signal's falling edge</li> </ul>
	<ul> <li>Up/down switching cause select bit</li> <li>0 : Up/down flag's content</li> </ul>
	0 (Must always be "0" in event counter mode)
	<ul> <li>Count operation type select bit</li> <li>0 : Reload type</li> </ul>
	When not using two-phase pulse signal processing, set this bit to "0"
b15 b8 b7 0016 b7 b0	b0         Timer A0 register           0016         TA0 [Address 038716, 038616]
b7	0 Up/down flag [Address 038416] UDF
b7	Timer A0 up/down flag 0 : Down count
	Count start flag [Address 038016] TABSR Timer A0 count start flag 1 : Starts counting
ь7 0000	Dimeter of the start flag [Address 038216]
	<ul> <li>Reserved bit (Must always be "0")</li> <li>Timer A0 event/trigger select flag</li> <li>b7 b6</li> </ul>
	0 0 : Input on TA0IN is selected (Note 1)
Note: Set the corresp	ponding port direction register to "0".
Setting interrupt priorit	-
	Timer A0 interrupt control register [Address 005416]
	Interrupt control level (set a value 1 to 7)
	I
Initialization of port P7	direction register
ь7	Port P7 direction register [Address 03EF16]
	Port P71 direction register     0 : Input mode
	Setting interrupt enable flag (I flag)

Figure 3.5.1. Set-up procedure of solution for a shortage of external interrupt pins





#### 3.6 Memory to Memory DMA Transfer

Overview The following are steps for changing both source address and destination address to transfer data from memory to another. The DMA transfer utilizes the workings that assign a higher priority to the DMA0 transfer if transfer requests simultaneously occur in two DMA channels. Figure 3.6.1 shows the operation timing, Figure 3.6.2 shows the block diagram, and Figures 3.6.3 and 3.6.4 show the set-up procedure.

Use the following peripheral functions:

- Timer mode of timer A
- Two DMAC channels
- One-byte temporary RAM (address 080016)

Specifications

- (1) Transfer the content of memory extending over 128 bytes from address F600016 to a 128byte area starting from address 0040016. Transfer the content every time a timer A0 interrupt request occurs.
- (2) Use DMA0 for a transfer from the source to built-in memory, and DMA1 for a transfer from built-in memory to the destination.
- Operation (1) A timer A interrupt request occurs. Though both a DMA0 transfer request and a DMA1 transfer request occur simultaneously, the former is executed first.
  - (2) DMA0 receives a transfer request and transfers data from the source to the built-in memory. At this time, the source address is incremented.
  - (3) Next, DMA1 receives a transfer request and transfers data involved from built-in memory to the destination. At this time, the destination address is incremented.

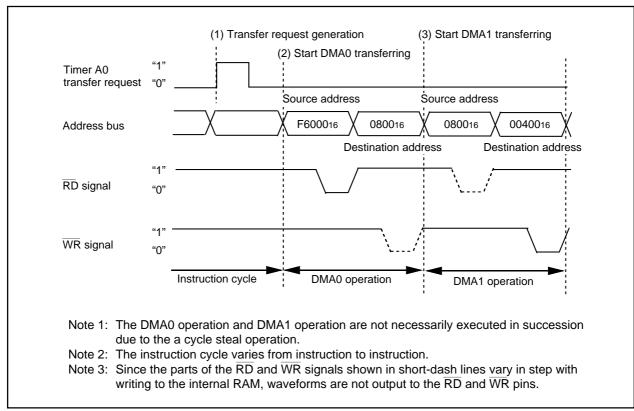


Figure 3.6.1. Operation timing of memory to memory DMA transfer





#### **DMAC** Applications

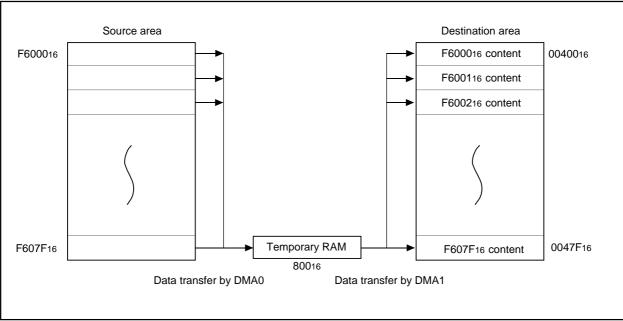


Figure 3.6.2. Block diagram of memory to memory DMA transfer





#### **DMAC** Applications

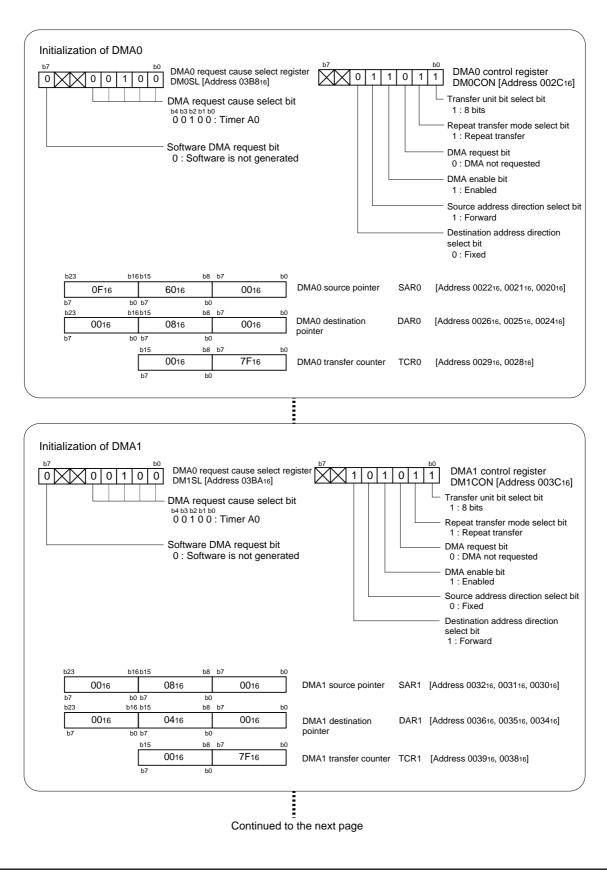


Figure 3.6.3. Set-up procedure of memory to memory DMA transfer (1)





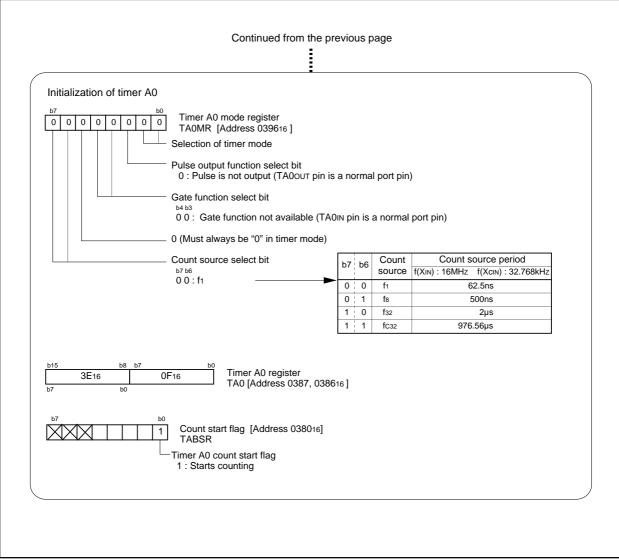


Figure 3.6.4. Set-up procedure of memory to memory DMA transfer (2)



Under

**CRC Snoop Function Applications** 

#### 3.7 CRC Calculation SFR Access Snoop Function in Clock Synchronous Serial Data Transmit

The M30245 group, by use of DMAC, transfers data from the internal RAM to the UART1 and the Overview result is transferred to the UART1 by use of SFR access snoop function. The block diagram is shown in Figure 3.7.1 and the setting routine is shown in Figure 3.7.2 to Figure 3.7.4. The peripheral functions to be used are as follows:

- DMAC 1 Channel
- Internal RAM (address 0040016) 512 bytes
- UART1 (Clock synchronous serial I/O mode)
- CRC calculation circuit
- SFR access snoop function

Specifications

- (1) Data transfer is performed starting at address 0040016 from the area with 512 bytes to the UART1. Data are transferred from area between the address 0040016 and the 512nd byte to the UART1. Transfer is executed every time 1 byte of serial transmit is completed.
- (2) Use the DMA0 to transfer data from the internal RAM to the UART1. Select the UART1 transmit to the DMA0 request factor. Select the single transfer mode and set the DMA0 transfer counter to 511 bytes (512-1).
- (3) Set the CRC calculation circuit to the CRC-CCITT and set CRC snoop address register to the address of UART1 transmit buffer register (write snoop).
- (4) On completing the DMA, 2-byte data of CRC data register (calculation result) are transferred to the UART1 and operation is completed.

#### Operation

- (1) Initialize the UART1 related registers.
- (2) Initialize the DMA0 related registers in DMA disable state.
- (3) Set the DMA0 transfer counter to the transfer data consisting of 511 bytes (in this case, 8-bit transfer).
- (4) Initialize the CRC calculation circuit and the SFR access snoop function.
- (5) Set the software DMA request bit of DMA0 to "1". At this time, 1st byte data are transferred from RAM to the transmit buffer of the UART1. Simultaneously, the transfer source address is incremented and the content of the transfer counter is down-counted. The transferred data are automatically written in CRC input register by the SFR access snoop function.
- (6) When the transmit buffer of the UART1 becomes writable state, the DMA transfer request is occurred by the UART1. At this time, the next data are transferred from RAM to the transmit buffer of the UART1. Simultaneously, the transfer source address is incremented and the content of the transfer counter is down-counted. The transferred data are automatically written in CRC input register by the SFR access snoop function.
- (7) As a result of repetition of the above (6), when the DMA0 transfer counter underflow, DMA enable bit is set to "0" to complete the DMA0 transfer. Simultaneously, the DMA0 interrupt request occurs. When the DMA0 interrupt request is detected, CRC data register (2 bytes) is read, it is transferred to the UART1 transmit buffer sequentially.





Under Preliminary Specifications REV.A Specifications in this manual Specifications in this manual are tentative and subject to change.

#### **CRC Snoop Function Applications**

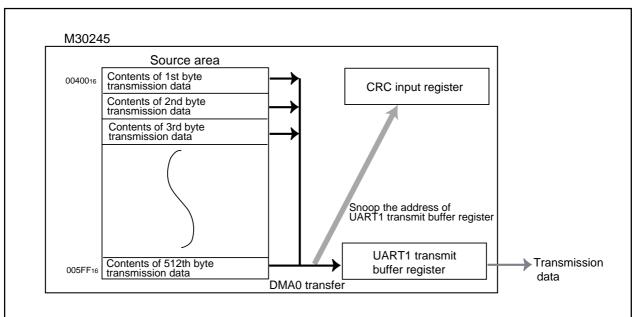


Figure 3.7.1. Block diagram of DMA transfer from RAM to UART and SFR snooping function





	e "2.3.2 Operation of Serial I/O (transmission in clock synchronous serial I/O mode" for detail.)
• En	able UART1 transmit
b7	UART1 transmit / receive control register 1 U1C1 [Address 036D16]
	Transmit enable bit 1 : Transmit enable
	able DMA0
b7	DMA0 control register DM0CON [Address 002C16] DMA enable bit 0 : Disabled
	tting DMA0 cause select register
b7	DMA0 cause select register DMOSL [Address 03B816] DMA request cause select bits 0 1 1 1 0 : UART1 transmit Nothing is assigned. Write "0" when writing to these bits. Software DMA request bit 0 : Not occurred
	0       1       0       0       1         0       1       0       0       1         MOCON       [Address 002C16]         Transfer unit select bit       1 : 8 bits         Repeat transfer mode select bit       0 : Single transfer         DMA request bit       0 : DMA not requested         DMA enable bit       0 : Disabled         Source address direction select bit       1 : Forward         Destination address direction select bit       0 : Fixed
<ul> <li>Set</li> </ul>	ting source pointer(internal RAM address) and destination pointer (UART1 transmit buffe
	b19 b16b15 b8 b7 b0 0 0 0 0 0416 0016 DMA0 source pointer SAR0 [Address 002216 to 002016]
	Stores the internal RAM address (040016) Nothing is assigned. Write "0" when writing to these bits.
	b19 b16b15 b8 b7 b0 DMA0 destination pointer 0 0 0 0 0316 6A16 DAR0 [Address 002616 to 002416]
	Stores the address of UART1 transmit buffer register (036A
	Nothing is assigned. Write "0" when writing to these bits.

Figure 3.7.2. Setting routine (1) of DMA transfer from RAM to UART using SFR snooping function





#### **CRC** Snoop Function Applications

• Enable DM	
b7	DMA0 control register DM0CON [Address 002C16] DMA request bit 0 : DMA not requested DMA enable bit
	1 : Enabled
Clear CRC	data register
0016	0016 CRC data register CRCD [Address 03BD16, 03BC16]
	0 : CRC-CCITT CRC mode selection bit 0 1 : MSB first mode
	C snoop address register CRC snoop address register CRCSRA [Address 03B516, 03B416] SFR snoop address bit Set address: 036A16 (UART1 transmit buffer register) CRCSAR Read 0 : Disabled CRCSAR Write 1 : Enabled
• Set Softwar	re DMA request bit to "1" in the status that DMA enable bit is "1" DMA0 cause select register DM0SL [Address 03B816]
	Software DMA request bit 1 : Occurred Transfer to CBC input register of the con-
	Transfer to CRC input register at the san time by SFR snoop function
	<b>:</b>





#### **CRC Snoop Function Applications**

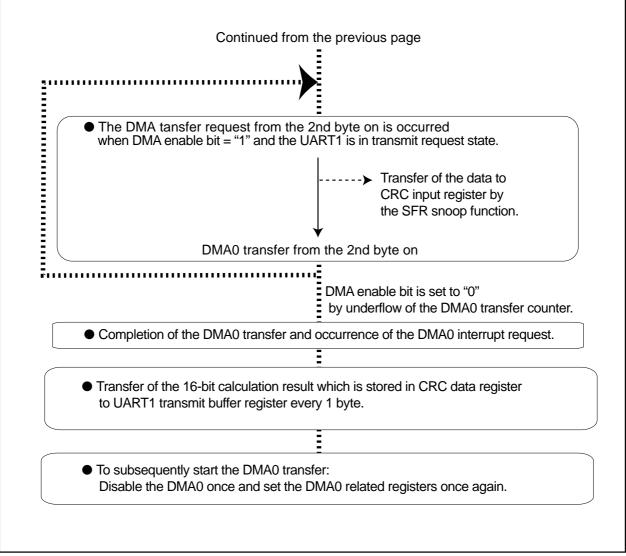
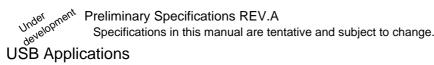


Figure 3.7.4. Setting routine (3) of DMA transfer from RAM to UART using SFR snooping function





#### 3.8 Transfer from USB FIFO to Serial Sound Interface

Overview The M30245 group, by use of DMAC, transfers data from the USB endpoint 1 OUT FIFO to SS interface 1 transmit buffer register and fetches one packet data.

The block diagram is shown in Figure 3.8.1 and the setting routine is shown in Figure 3.8.2 to Figure 3.8.4.

The peripheral functions to be used are as follows:

- DMAC 1 channel
- USB endpoint 1 OUT (Receive)
- Serial sound interface 1

Specifications

- (1) Receive packet data of the endpoint 1 OUT FIFO are transferred to SS interface 1 transmit buffer register. Transfer is executed every time the DMA transfer factor of the serial sound interface 1 occurs.
- (2) Use the DMA0 to transfer data from the endpoint 1 OUT FIFO to SS interface 1 transmit buffer register. Select the serial sound interface 1 transmit to the DMA0 request factor. Select the single transfer mode and set the DMA0 transfer counter to 1/2 X (the data count of one packet received with endpoint 1 OUT) −1.
- (3) Set the endpoint 1 OUT maximum packet size to 288 bytes (when sampling 48KHz/ 24-bit/ stereo) and disable the AUTO\_CLR function. The data count of receive packet of endpoint 1 (endpoint 1 OUT write count register) is set to 288 bytes. Endpoint 1 OUT is used in isochronous transfer.
- (4) On completing the DMA0 transfer, fetch of one packet data from the endpoint 1 OUT FIFO is completed by setting CLR\_OUT\_BUF\_RDY bit of endpoint 1 to "1".

Operation

- (1) Initialize the DMA0 related registers in the state which DMA is disabled and USB DMA0 request register is not selected (in this case, 16-bit transfer).
- (2) When the OUT\_BUF\_STS1 flag of endpoint 1 is set to "1" and packet data receive has been detected, set the DMA0 transfer counter to the 1/2 X (the data count of receive one packet) –1 (in this application example, 143 value is set).
- (3) Set DMA enable bit of DMA0CON to "1" (DMA0 is enabled). Then, the DMA0 transfer request from the serial sound interface occurs.
- (4) When the transfer request is received, the DMA0 transfers the 1st word (16-bit) data from the endpoint 1 OUT FIFO to the serial sound interface 1. Simultaneously, the content of the transfer counter is down-counted. Then, the DMA0 transfer request from the serial sound interface occurs.
- (5) As a result of repetition of the above (4), when the DMA0 transfer counter underflow, DMA enable bit is set to "0" to complete the DMA0 transfer. Simultaneously, the DMA0 interrupt request occurs. When the DMA0 interrupt request is detected, set CLR\_OUT\_BUF\_RDY bit of endpoint 1 OUT to "1".





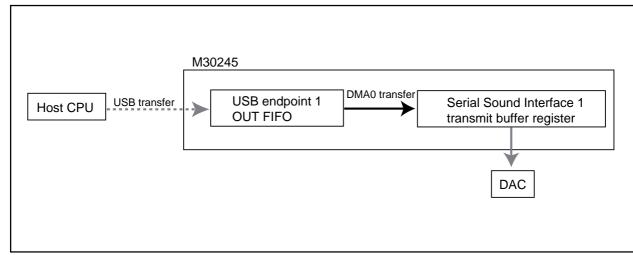


Figure 3.8.1. Block diagram of DMA transfer from USB FIFO to serial sound interface





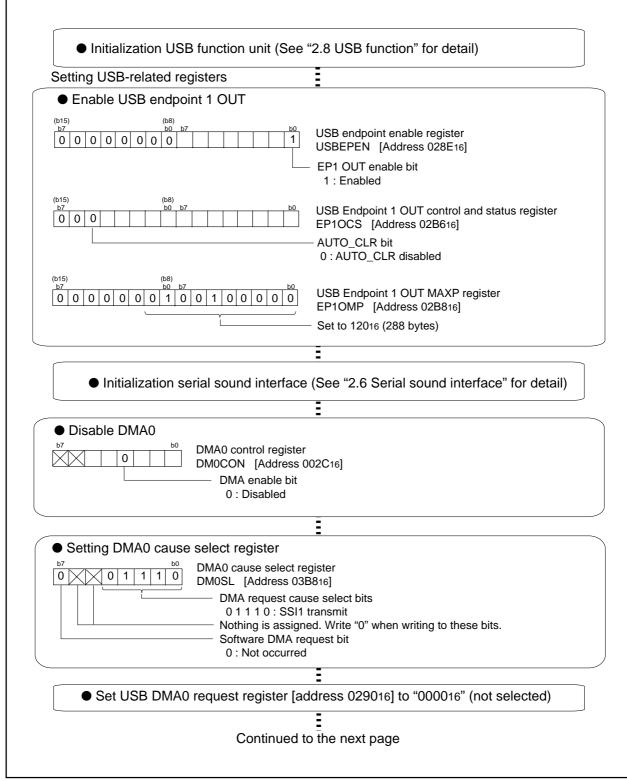


Figure 3.8.2. Setting routine (1/3) of DMA transfer from USB OUT FIFO to serial sound interface





#### Continued from the previous page Ξ Setting DMA0 control register b0 DMA0 control register $\times$ 0 0 0 0 0 0 DM0CON [Address 002C16] Transfer unit select bit 0:16 bits Repeat transfer mode select bit 0 : Single transfer DMA request bit 0 : DMA not requested DMA enable bit 0: Disabled Source address direction select bit 0: Fixed Destination address direction select bit 0: Fixed Setting source pointer(endpoint 1 OUT FIFO data register) and destination pointer (SS interface 1 transmit buffer register) DMA0 source pointer 0000 0216 E616 SAR0 [Address 002216 to 002016] Stores the endpoint 1 OUT FIFO (Address 02E616) Nothing is assigned. Write "0" when writing to these bits. b16b15 b8 DMA0 destination pointer 0000 0316 7416 DAR0 [Address 002616 to 002416] Stores the SS interface 1 transmit register (Address 037416) Nothing is assigned. Write "0" when writing to these bits. = • Checking that OUT\_BUF\_STS1 flag is "1" and setting the number of the transfer bytes (Note) (b15) (b8) b0 b7 h7 b0 DMA0 transfer counter 0016 8F16 TCR0 [Address 002916, 002816] Note: Set 1/2 X (the value of Endpoint 1 OUT write count register) -1. Enable serial sound interface 1 Serial Sound Interface 1 mode register 0 1 SSI1MR0 [Address 037016] Serial Sound Interface enable bit 1: Enabled Enable DMA0 DMA0 control register 1 0 DM0CON [Address 002C16] DMA request bit 0 : DMA not requested DMA enable bit 1 : Enabled Continued to the next page

Figure 3.8.3. Setting routine (2/3) of DMA transfer from USB OUT FIFO to serial sound interface





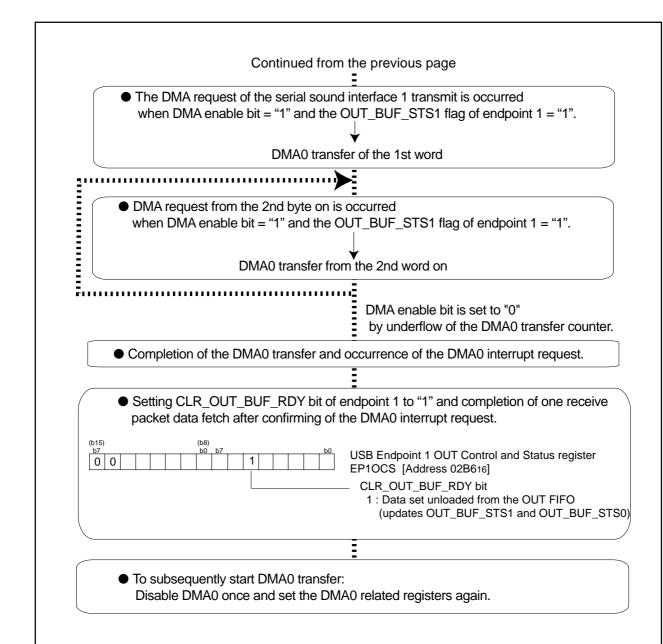


Figure 3.8.4. Setting routine (3/3) of DMA transfer from USB OUT FIFO to serial sound interface





**Controlling Power Applications** 

#### 3.9 Controlling Power Using Stop Mode

Overview The following are steps for controlling power using stop mode. Figure 3.9.1 shows the operation timing, Figure 3.9.2 shows an example of circuit, and Figures 3.9.3 and 3.9.4 show the set-up procedure.

Use the following peripheral functions:

- Key-input interrupts
- Stop mode
- Pull-up function

This example is not performed USB power control. Please refer section 2.7.4 for the power control of USB related.

#### Specifications

- (1) Use P00 through P03 for the scan output pins of a key matrix. Use the input pins ( $\overline{\text{KI0}}$  through  $\overline{\text{KI3}}$ ) of the key-input interrupt function for the key-input reading pins. The pull-up function is also used.
- (2) If a key-input interrupt request occurs, clear the stop mode and read a key.
- Operation (1) Enable a key-input interrupt and set the pull-up function to pins  $\overline{\text{KI0}}$  through  $\overline{\text{KI3}}$ . Change the output of P00 through P03 to "L" and enter stop mode.
  - (2) If a key is pressed, "L" is input to one of pins  $\overline{\text{KI}_0}$  through  $\overline{\text{KI}_3}$  to clear stop mode. A key-input interrupt occurs to execute the key-input interrupt handling routine.
  - (3) Sequentially set P00 through P03 to "L" to determine which key was pressed.
  - (4) When the process to determine the key pressed is completed, change the output from P00 through P03 to "L" again and enter stop mode.





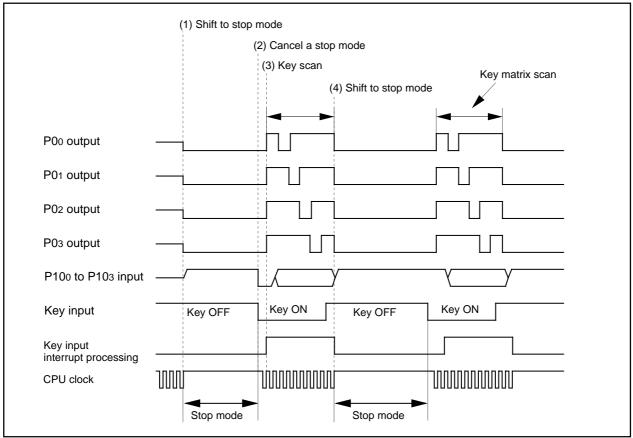
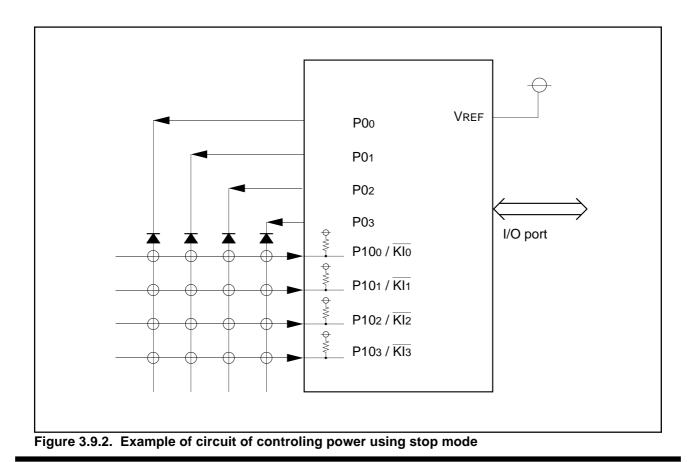


Figure 3.9.1. Operation timing of controlling power using stop mode





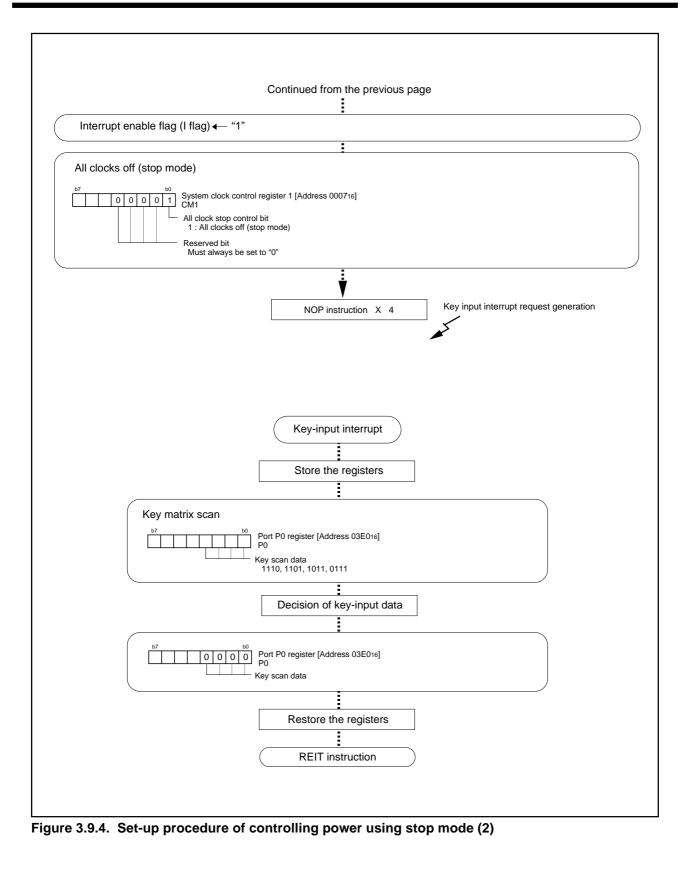


	Main					
Initial condition	•					
b7 b0 Pull-up control register 2 [Address 03FE16] PUR2 P100 to P103 pulled high	b7 b0 Port P0 direction register [Address 03E216] PD0 Key scan output port					
b7 b0 Address 03F616] PD0 Key scan input port	b7 Key input interrupt control register [Address 004116] KUPIC					
b7         b0         Port P0 register           [Address 03E016]         P0           P0         Key scan data	Interrupt priority level select bit Set higher value than the present IPL Processor interrupt priority level (IPL) = 0 Interrupt enable flag (I) =0					
	•					
Sotting interrupt execut step made cancel	Ē					
S13BCNIC [Address 004316	, 004516, 004716, 005716, 005916] ] 6] 6, 004E16, 005016, 005216] 5, 005116, 004F16, 004D16] 3] 6] 6] 6]					
b7 b0 Interrupt priority level select bit 000 : Interrupt disabled	b7       b0       INTIIC(i=0 to 2) [Address 005F16, 004416, 005E16]         S1RIC       [Address 004816]         S02BCNIC       [Address 004916]         Interrupt priority level select bit       000 : Interrupt disabled         Reserved bit       Must always be set to "0"					
	<sup>116]</sup> k control registers 0 and 1 (addresses 000616 and 000716) isters (addresses 03DB16 to 03DF16)					
Setting operation clock after returning from st						
(When operating with XIN after returning) b7 b0 System clock control register	(When operating with XCIN after returning) 0 b7 b0 System clock control register 0					
0 0 0 0 0 0 CM0 [Address Outfold register Reserved bit Must always be set to "0" Main clock (XIN-XOUT) stop bit On System clock select bit XIN, XOUT	0     0					
As this register becomes setting mentioned above when operating with XIN (count source of BCLK is XIN), the user does not need to set it again. When operating with XCIN, set main clock (XIN-XOUT) stop to "0" before setting system clock select bit to "0". The both bits cannot be set at the same time.						
Continued to the next page						

Figure 3.9.3. Set-up procedure of controlling power using stop mode (1)











**Controlling Power Applications** 

#### 3.10 Controlling Power Using Wait Mode

Overview The following are steps for controling power using wait mode. Figure 3.10.1 shows the operation timing, and Figures 3.10.2 to 3.10.4 show the set-up procedure.

Use the following peripheral functions:

- Timer mode of timer A
- Wait mode

A flag named "F-WIT" is used in the set-up procedure. The purpose of this flag is to decide whether or not to clear wait mode. If  $F_WIT = "1"$  in the main program, the wait mode is entered; if  $F_WIT = "0"$ , the wait mode is cleared.

Specifications

- (1) Connect a 32.768-kHz oscillator to XCIN to serve as the timer count source. As interrupts occur every one second, which is a count the timer reaches, the controller returns from wait mode and count the clock using a program.
- (2) Clear wait mode if a  $\overline{INT0}$  interrupt request occurs.

Operation (1) Switch the system clock from XIN to XCIN to get low-speed mode.

- (2) Stop XIN and enter wait mode. In this instance, enable the timer A2 interrupt and the INTO interrupt.
- (3) When a timer A2 interrupt request occurs (at 1-second intervals), start supplying the BCLK from XCIN. At this time, count the clock within the routine that handles the timer A2 interrupts and enter wait mode again.
- (4) If a INTO interrupt occurs, start supplying the BCLK from XCIN. Start the XIN oscillation within the INTO interrupt, and switch the system clock to XIN.

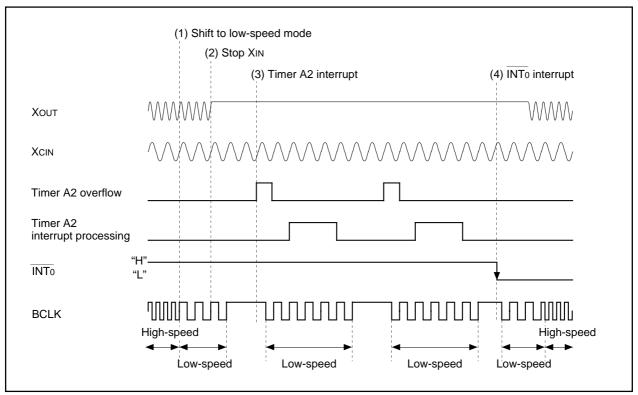


Figure 3.10.1. Operation timing of controling power using wait mode



Under

Operation mode select bit 0 0 : Timer mode Count source select bit

FF16

TA2 interrupt priority level

- INT0 interrupt priority level

-Interrupt priority level select bit 0 0 0 : Interrupt disabled

Interrupt priority level select bit 000 : Interrupt disabled

Must always be set to "0"

Rrescaler is reset

TABSR TA2 start counting

INTOIC

1 1 : fc32 (f(Xcin) divided by 32)

Count start flag [Address 038016]

Clock prescaler reset flag [Address 038116] CPSRF

Timer A2 interrupt control register [Address 004716] TA2IC

INT0 interrupt control register [Address 005F16]

b7 b6

b7

Х

0 0 1

0 0 1

Processor interrupt priority level (IPL) = 0

Interrupt enable flag (I) = 0

Interrupt control register KUPIC SiRIC(i=0,2,3)

b0

Setting interrupt except clearing wait mode

[Address 004316]

0 0 0

0 0 0

[Address 004116] [Address 004A16, 004216, 005516]

TAIIC(i=0,1,3,4) [Address 005416, 004516, 005716, 005916] RSMIC EPOIC [Address 005416] RSTIC

#### Controlling Power Appl

b15

1

 $XX^{0}$ 

S13BCNIC

0316

XXXD

cifications in this manual are tentative and subject to change. /er Applications SINGLE-CHIP 16-BIT CMO	M30245 group IOS MICROCOMPUTER	
Main		
Initial condition          b7       0       0       1       0       0       0       System clock control register 0 [Address 000616]         CM0       CM0       Reserved bit       Reserved bit       Reserved bit		
Must always be set to "0" WAIT peripheral function clock stop bit 0 : Do not stop peripheral function clock in wait mode XCIN-XCOUT drive capacity select bit		
Port Xc select bit 1 : Functions as XCIN-XCOUT oscillator Main clock (XIN-XOUT) stop bit 0 : Oscillating		
Main clock divide ratio select bit 0 System clock select bit 0 : XIN-XOUT		
b7     b0       1     1       0     0       TA2MR		

Timer A2 register [Address 038A16, 038B16] TA2

ADIC [Address 004B16] DMilC(i=0 to 3) [Address 004C16, 004E16, 005016, 005216] SiTIC(i=0 to 3) [Address 005316, 005116, 004F16, 004D16] SUSPIC [Address 005616]

[Address 005816] [Address 005A16]

[Address 005B16] [Address 005C16]

[Address 005D16]

Mitsubishi Microcomputers

Continued to the next page

SOFIC

VBDIC

[Address 004416, 005E16] [Address 004816] [Address 004916]

USBFIC

Figure 3.10.2. Set-up procedure of controlling power using wait mode (1)

INTiIC(i=1,2)

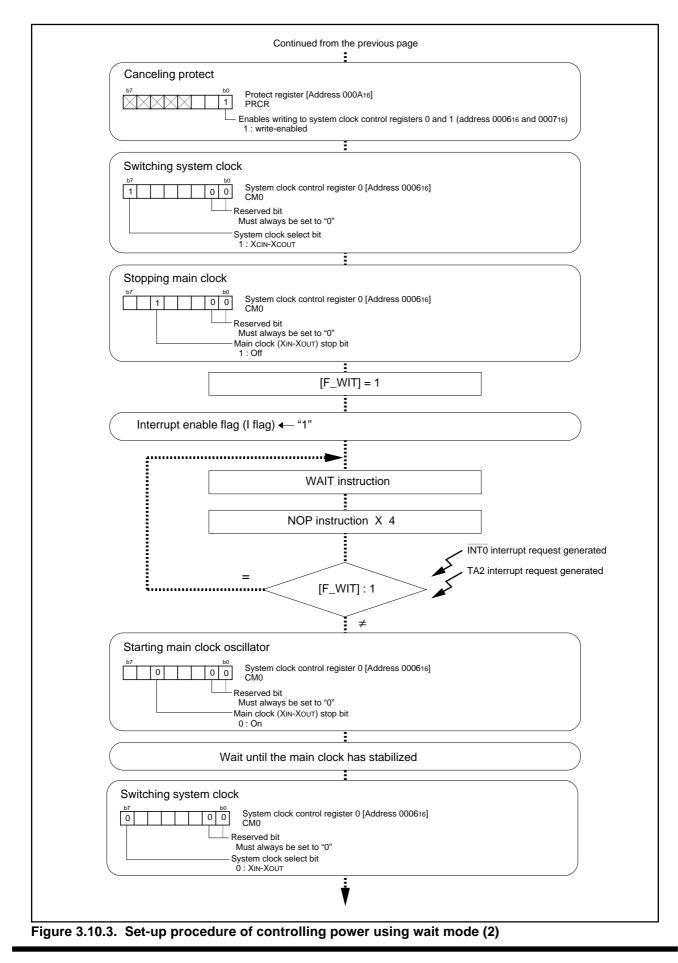
S02BCNIC

Reserved bit

S1RIC

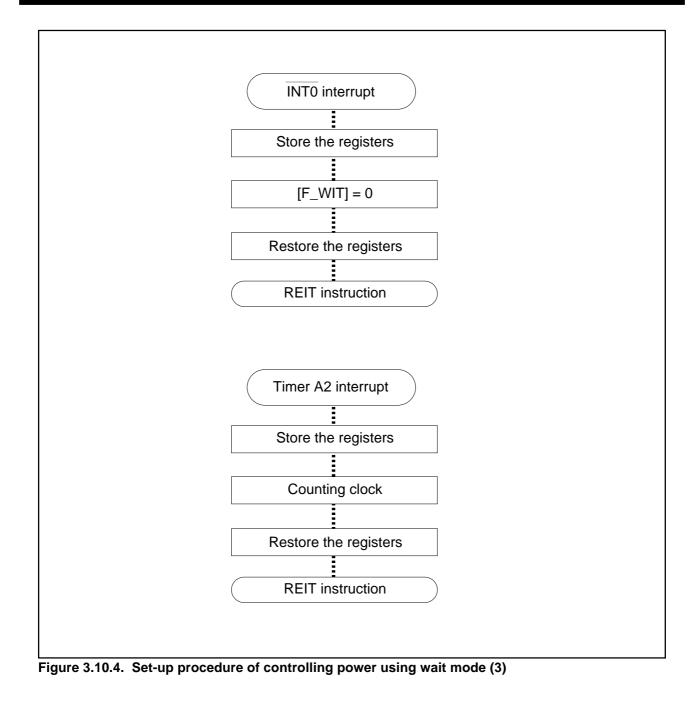












#### MITSUBISHI SEMICONDUCTORS USER'S MANUAL M30245 Group

Jan. First Edition 2003

Editioned by Committee of editing of Mitsubishi Semiconductor User's Manual

Published by Mitsubishi Electric Corp., Semiconductor Marketing Division

This book, or parts thereof, may not be reproduced in any form without permission of Mitsubishi Electric Corporation.

©2003 MITSUBISHI ELECTRIC CORPORATION

### User's Manual M30245 Group

# RenesasTechnologyCorp. Nippon Bldg.,6-2,Otemachi 2-chome,Chiyoda-ku,Tokyo,100-0004 Japan