Abstract

This document describes using mirror registers with the three-phase motor control timers in the R32C/100 Series.

Products

R32C/120 Group
R32C/121 Group
R32C/145 Group
R32C/151 Group
R32C/152 Group
R32C/153 Group
R32C/156 Group
R32C/157 Group
R32C/160 Group
R32C/161 Group

When using this application note with other Renesas MCUs, careful evaluation is recommended after making modifications to comply with the alternate MCU.
Contents

1. Specifications .............................................................................................................. 3
2. Operation Confirmation Conditions ............................................................................. 4
3. Reference Application Notes ...................................................................................... 4
4. Peripheral Functions .................................................................................................... 5
   4.1 Timer Ai/Timer Ai-1 Mirror Registers ................................................................... 5
   4.2 Notes on Setting the TAi and TAi1 Registers (i = 1, 2, 4) ........................................ 7
5. Hardware ..................................................................................................................... 7
   5.1 Pins Used .............................................................................................................. 7
6. Software ..................................................................................................................... 8
   6.1 Operation Overview ............................................................................................... 9
   6.2 Constant ............................................................................................................... 10
   6.3 Structure/Union List ............................................................................................ 10
   6.4 Variables ............................................................................................................ 10
   6.5 Functions ........................................................................................................... 11
   6.6 Function Specifications ....................................................................................... 11
   6.7 Flowcharts .......................................................................................................... 12
      6.7.1 Main Processing ............................................................................................... 12
      6.7.2 DMAC II Initial Setting .................................................................................. 13
      6.7.3 Initial Setting of the Three-Phase Motor Control Timers .................................. 14
      6.7.4 DMAC II Transfer Complete Interrupt Handling ............................................ 16
7. Sample Code ............................................................................................................. 17
8. Reference Documents ............................................................................................... 17
1. Specifications

When using three-phase motor control timers to output a triangular wave, by combining the timer Ai/timer Ai-1 mirror registers with a DMAC II burst transfer, the timer value can be set by a single transfer request (i = 1, 2, 4).

Table 1.1 lists the Peripheral Functions and Their Applications. Figure 1.1 shows a Block Diagram.

### Table 1.1 Peripheral Functions and Their Applications

<table>
<thead>
<tr>
<th>Peripheral Function</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timer A (timers A1, A2, and A4)</td>
<td>Three-phase motor control timers</td>
</tr>
<tr>
<td>Timer B (timer B2)</td>
<td>Three-phase motor control timers</td>
</tr>
<tr>
<td>DMAC II</td>
<td>Transfers setting values to timer Ai/timer Ai-1 mirror registers</td>
</tr>
</tbody>
</table>

![Figure 1.1 Block Diagram](image)

**Note:**

1. Values written to the TAiM and TAi1M registers are reflected in the TAi and TAi1 registers (i = 1, 2, 4).
2. **Operation Confirmation Conditions**

The sample code accompanying this application note has been run and confirmed under the conditions below.

<table>
<thead>
<tr>
<th>Item</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCU used</td>
<td>R5F64219JFB (R32C/121 Group)</td>
</tr>
<tr>
<td>Operating frequencies</td>
<td>• Main clock: 8 MHz</td>
</tr>
<tr>
<td></td>
<td>• PLL clock: 128 MHz</td>
</tr>
<tr>
<td></td>
<td>• Base clock: 64 MHz</td>
</tr>
<tr>
<td></td>
<td>• CPU clock: 64 MHz</td>
</tr>
<tr>
<td></td>
<td>• Peripheral bus clock: 32 MHz</td>
</tr>
<tr>
<td></td>
<td>• Peripheral function clock source: 32 MHz</td>
</tr>
<tr>
<td>Operating voltage</td>
<td>5 V</td>
</tr>
<tr>
<td>Integrated development environment</td>
<td>Renesas Electronics</td>
</tr>
<tr>
<td></td>
<td>High-performance Embedded Workshop Version 4.07</td>
</tr>
<tr>
<td>C compiler</td>
<td>Renesas Electronics</td>
</tr>
<tr>
<td></td>
<td>R32C/100 Series C Compiler V.1.02 Release 01</td>
</tr>
<tr>
<td></td>
<td>Compile options</td>
</tr>
<tr>
<td></td>
<td>-D__STACKSIZE__=0X300 -D__ISTACKSIZE__=0X300</td>
</tr>
<tr>
<td></td>
<td>-DVECTOR_ADR=0x0FFFFFFBDC -c -finfo -dir &quot;$(CONFIGDIR)&quot;</td>
</tr>
<tr>
<td>Default setting is used in the integrated development environment.</td>
<td></td>
</tr>
<tr>
<td>Operating mode</td>
<td>Single-chip mode</td>
</tr>
<tr>
<td>Sample code version</td>
<td>Version 1.00</td>
</tr>
</tbody>
</table>

3. **Reference Application Notes**

Application notes associated with this application note are listed below. Refer to these application notes for additional information.

- R32C/100 Series Configuring PLL Mode (REJ05B1221-0100)
- R32C/100 Series Three-phase Motor Control Timers (Triangular Wave Modulation Mode, Three-phase Mode 1) (R01AN0029EJ0100)
- R32C/100 Series DMA II Setting Example (Memory-to-memory Burst Transfer) (REJ05B1228-0100)
4. Peripheral Functions

This chapter provides supplementary information on the timer Ai/timer Ai-1 mirror registers (i = 1, 2, 4). Refer to the User’s Manual (Hardware) for general information.

4.1 Timer Ai/Timer Ai-1 Mirror Registers

Values set to timer Ai and timer Ai-1 mirror registers can be reflected in timer Ai and timer Ai-1 registers (TAi and TAI1 registers). For example, if a value is set to the TA1M register, the same value is set to the TA1 register.
Figure 4.1 shows the timer Ai and timer Ai-1 registers, and Figure 4.2 shows the timer Ai and timer Ai-1 mirror registers.

**Timer Ai/Timer Ai-1 Registers (i = 1, 2, 4) (1 to 6)**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Address</th>
<th>Reset Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TA1, TA2, TA4</td>
<td>0349h to 0348h, 034Bh to 034Ah, 034Fh to 034Eh</td>
<td>Undefined</td>
</tr>
<tr>
<td>TA11, TA21, TA41</td>
<td>0303h to 0302h, 0305h to 0304h, 0307h to 0306h</td>
<td>Undefined</td>
</tr>
</tbody>
</table>

**Function**
The timer stops when the nth count source is counted after a start trigger is generated. Output signal for each phase is switched when timers A1, A2, and A4 stop (n = setting value)

<table>
<thead>
<tr>
<th>Setting Range</th>
<th>RW</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000h to FFFFh</td>
<td>WO</td>
</tr>
</tbody>
</table>

**Notes:**
1. A 16-bit write access to these registers should be performed.
2. When these registers are set to 0000h, the counter does not start, and no timer Ai interrupt request is generated.
3. The MOV instruction should be used to set these registers.
4. When the INV15 bit in the INVC1 register is set to 0 (dead time enabled), the turn-on output signal is switched to its active state with a delay. It is switched when the dead time timer stops.
5. When the INV11 bit in the INVC1 register is set to 0 (three-phase mode 0), the value of the TAI register is transferred to the reload register by a timer Ai start trigger. When the INV11 bit is set to 1 (three-phase mode 1), firstly the value of the TAi1 register is transferred to the reload register by a timer Ai start trigger. Then the value of the TAI register is transferred by the next timer Ai start trigger. After that the value of these registers TAi1 and TAI is alternately transferred to the reload register.
6. These registers should not be written when the timer B2 underflows.

**Figure 4.2 Timer Ai/Timer Ai-1 Mirror Registers**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Address</th>
<th>Reset Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TA1M, TA11M</td>
<td>0221h to 0220h, 0223h to 0222h</td>
<td>Undefined</td>
</tr>
<tr>
<td>TA2M, TA21M</td>
<td>0225h to 0224h, 0227h to 0226h</td>
<td>Undefined</td>
</tr>
<tr>
<td>TA4M, TA41M</td>
<td>0229h to 0228h, 022Bh to 022Ah</td>
<td>Undefined</td>
</tr>
</tbody>
</table>

**Function**
The timer stops when the nth count source is counted after a start trigger is generated. Output signal for each phase is switched when timers A1, A2, and A4 stop (n = setting value)

<table>
<thead>
<tr>
<th>Setting Range</th>
<th>RW</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000h to FFFFh</td>
<td>WO</td>
</tr>
</tbody>
</table>

**Notes:**
1. A 16-bit write access to these registers should be performed.
2. The values written in these registers are reflected to registers TAj and TAj1.
3. When these registers are set to 0000h, the counter does not start, nor the timer Ai interrupt request is generated.
4. The MOV instruction should be used to set these registers.
5. When the INV15 bit in the INVC1 register is set to 0 (dead time enabled), the turn-on output signal is switched to its active state with a delay; It is switched when the dead time timer stops.
6. When the INV11 bit in the INVC1 register is set to 0 (three-phase mode 0), the value of the TAI register is transferred to the reload register by a timer Ai start trigger. When the INV11 bit is set to 1 (three-phase mode 1), firstly the value of the TAi1 register is transferred to the reload register by a timer Ai start trigger. Then the value of the TAI register is transferred by the next timer Ai start trigger. After that the value of these registers TAi1 and TAI is alternately transferred to the reload register.
7. These registers should not be written at the timing when the timer B2 underflows.
4.2 Notes on Setting the TA\textsubscript{i} and TA\textsubscript{i1} Registers (i = 1, 2, 4)

Note the following when setting values to the TA\textsubscript{i} and TA\textsubscript{i1} registers.

(1) TA\textsubscript{i} setting value
When 0 is set to the TA\textsubscript{i} register (0 or 1 when the dead time timer count source is f1 divided by 2), as the TA\textsubscript{i} timer does not count and a falling edge does not occur, the current output waveform level is maintained.

If a value larger than the TB\textsubscript{2} setting value is set to TA\textsubscript{i} (when the dead time timer count source is f1 divided by 2, a value larger than the TB\textsubscript{2} setting value - 1), the TA\textsubscript{i} timer continues to be counted in the TB\textsubscript{2} period, a falling edge is not generated, and the output waveform maintains its current level. Therefore, only set a value larger than the TB\textsubscript{2} setting value to TA\textsubscript{i} when the output waveform level needs to be maintained.

(2) Restarting the dead time timer
Depending on the data set to the TA\textsubscript{i} register, if a dead time timer activation source is generated during a dead time timer count, the dead time timer will not restart. If the conditions below are met, note that the dead time timer will not restart.

In triangular wave modulation mode (three-phase mode 0): dead time timer count source f1
((TB\textsubscript{2} setting value + 1) - even numbered TA\textsubscript{i} setting value) + odd numbered TA\textsubscript{i} setting value is less than the dead time timer setting value
Even numbered TA\textsubscript{i} setting value + ((TB\textsubscript{2} setting value + 1) - odd numbered TA\textsubscript{i} setting value) is less than the dead time timer setting value

In triangular wave modulation mode (three-phase mode 1): dead time timer count source f1
((TB\textsubscript{2} setting value + 1) - TA\textsubscript{i1} setting value) + TA\textsubscript{i} setting value is less than the dead time timer setting value
TA\textsubscript{i1} setting value + ((TB\textsubscript{2} setting value + 1) - TA\textsubscript{i} setting value) is less than the dead time timer setting value

Sawtooth wave modulation mode: dead time timer count source f1
((TB\textsubscript{2} setting value + 1) - TA\textsubscript{i1} setting value) - 1 is less than the dead time timer setting value
TA\textsubscript{i} setting value - 1 is less than the dead time timer setting value

5. Hardware

5.1 Pins Used

Table 5.1 lists the Pins Used and Their Functions.

<table>
<thead>
<tr>
<th>Pin Name</th>
<th>I/O</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>P7_2/TA1OUT/V</td>
<td>Output</td>
<td>Three-phase PWM waveform output (V-phase)</td>
</tr>
<tr>
<td>P7_3/TA1IN/V</td>
<td>Output</td>
<td>Three-phase PWM waveform output (V-phase)</td>
</tr>
<tr>
<td>P7_4/TA2OUT/W</td>
<td>Output</td>
<td>Three-phase PWM waveform output (W-phase)</td>
</tr>
<tr>
<td>P7_5/TA2IN/W</td>
<td>Output</td>
<td>Three-phase PWM waveform output (W-phase)</td>
</tr>
<tr>
<td>P8_0/TA4OUT/U</td>
<td>Output</td>
<td>Three-phase PWM waveform output (U-phase)</td>
</tr>
<tr>
<td>P8_1/TA4IN/U</td>
<td>Output</td>
<td>Three-phase PWM waveform output (U-phase)</td>
</tr>
</tbody>
</table>
6. Software

This chapter describes using the three-phase motor control timers to alternately output two types of triangular waves. DMA II transfer occurs every two times timer B2 underflows, and the timer Ai/timer Ai-1 mirror register values are rewritten (i = 1, 2, 4). The timer Ai/timer Ai-1 mirror register values are changed in the DMAC II transfer complete interrupt handling. The settings are shown below.

DMAC II settings
- Interrupt request level 7 is available for DMA II transfer.
- Set the transfer source to memory-to-memory transfer.
- Set the transfer size to 16 bits.
- Set the source addressing to increment.
- Set the destination addressing to increment.
- Set the transfer mode to burst transfer.
- Set the number of DMA II transfer complete interrupts to 6.

Three-phase motor control timer settings
- Set to triangular wave modulation mode (three-phase mode 1).
- Set frequency of timer B2 interrupt occurrence to 2.
- Set timer B2 underflow as the ICTB2 count condition.
- Set timer B2 underflow as the timer A1, A2, and A4 start trigger.
- Enable dead time.
- Use the P7 and P8 U, Û, V, Û, W, and Û pins for three-phase output.
### 6.1 Operation Overview

1. **Initial setting**
   - Perform initial setting of three-phase motor control timers and DMAC II.

2. **Start timer count**
   - Set 96h to the TABSR register, and start the count of timers A1, A2, A4, and B2. At this point, only timer B2 starts counting.

3. **Timer B2 underflow**
   - When timer B2 underflows the first time, timers A1, A2, and A4 start counting.

4. **Timer B2 underflow and DMA II transfer**
   - When timer B2 underflows the second time, timers A1, A2, and A4 start counting, a DMA II transfer occurs, and the timer Ai/timer Ai-1 mirror register values are rewritten (i = 1, 2, 4).

Figure 6.1 shows the U-Phase and Ū-Phase Output Timing Diagram.

The above diagram assumes the following:
- The INV11 bit in the INVC1 register is 1 (three-phase mode 1)
- The ICTB2 register is 02h.

Notes:
- 1. Internal signal
- 2. Values for registers TA4M and TA41M are set using DMA II transfer.
6.2 Constant
Table 6.1 lists the Constant Used in the Sample Code.

<table>
<thead>
<tr>
<th>Constant Name</th>
<th>Setting Value</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>D ДМАС2_ DATA_MAX</td>
<td>6</td>
<td>Number of DMАС II transfers</td>
</tr>
</tbody>
</table>

6.3 Structure/Union List
Figure 6.2 shows the Structure/Union Used in the Sample Code.

```c
/* DMАС II index */
struct{
    union {
        struct{
            int16_t size : 1;  /* Transfer size select bit */
            int16_t imm  : 1;  /* Transfer source select bit */
            int16_t upds : 1;  /* Source addressing select bit */
            int16_t updd : 1;  /* Destination addressing select bit */
            int16_t oper : 1;  /* Calculation transfer select bit */
            int16_t brst : 1;  /* Burst transfer select bit */
            int16_t inte : 1;  /* DMАС II transfer complete interrupt select bit */
            int16_t chain : 1;  /* Chained transfer select bit */
            int16_t reserve : 7;  /* Reserved */
            int16_t mult : 1;  /* Multiple transfer select bit */
        } mod_bit;
        uint16_t mod_word;
    } mod;
    uint16_t count;  /* Transfer counter */
    uint16_t* sadr;  /* Source address */
    uint16_t* dadr;  /* Destination address */
    void (*iadr)(void);  /* DMАС II transfer complete interrupt vector address */
} dm_index;
```

Figure 6.2 Structure/Union Used in the Sample Code

The interrupt vector for the peripheral function interrupt becomes the DMАС II request source. Set the DMАС II index starting address to the interrupt vector. In this document, the timer B2 interrupt is used as the request source for DMАС II.

Figure 6.3 shows a setting example of the asm function programmed in C.

```c
asm" .rvector 23, _dm_index ";  /* Define DMАС II Index (Software Interrupt Number 23) */
```

Figure 6.3 Example of Setting the Peripheral Function Variable Vector Table When Using the Timer B2 Interrupt as the DMАС II Request Source

6.4 Variables
Table 6.2 lists the Global Variables.

<table>
<thead>
<tr>
<th>Type</th>
<th>Variable Name</th>
<th>Contents</th>
<th>Function Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>unsigned short</td>
<td>data1[ ][ ]</td>
<td>Timer A1, A2, and A4 setting values</td>
<td>timer_init</td>
</tr>
<tr>
<td>unsigned char</td>
<td>flg_timer_value</td>
<td>Flag for when timer A1, A2, and A4 setting values change</td>
<td>main, dmac2_int</td>
</tr>
</tbody>
</table>
6.5 Functions

Table 6.3 lists the Functions.

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Outline</th>
</tr>
</thead>
<tbody>
<tr>
<td>dmac2_init</td>
<td>DMAC II initial setting</td>
</tr>
<tr>
<td>timer_init</td>
<td>Initial setting for the three-phase motor control timer</td>
</tr>
<tr>
<td>dmac2_int</td>
<td>DMAC II transfer completed interrupt handling</td>
</tr>
</tbody>
</table>

6.6 Function Specifications

The following tables list the sample code function specifications.

### dmac2_init

<table>
<thead>
<tr>
<th>Outline</th>
<th>DMAC II initial setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>None</td>
</tr>
<tr>
<td>Declaration</td>
<td>void dmac2_init(void)</td>
</tr>
<tr>
<td>Description</td>
<td>Perform the initial setting of the DMAC II.</td>
</tr>
<tr>
<td>Argument</td>
<td>None</td>
</tr>
<tr>
<td>Returned value</td>
<td>None</td>
</tr>
<tr>
<td>Remark</td>
<td>The DMAC II request source is allocated to software interrupt number 23 (interrupt source is timer B2).</td>
</tr>
</tbody>
</table>

### timer_init

<table>
<thead>
<tr>
<th>Outline</th>
<th>Initial setting for three-phase motor control timers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>None</td>
</tr>
<tr>
<td>Declaration</td>
<td>void timer_init(void)</td>
</tr>
</tbody>
</table>
| Description | • Perform the initial setting of the three-phase motor control timers (timers A1, A2, A4, and B2)  
• Set the P7 and P8 U, U, V, V, W, and W pins to three-phase output. |
| Argument | None                    |
| Returned value | None          |
| Remark |                          |

### dmac2_int

<table>
<thead>
<tr>
<th>Outline</th>
<th>DMAC II transfer complete interrupt handling</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>None</td>
</tr>
<tr>
<td>Declaration</td>
<td>void dmac2_int(void)</td>
</tr>
<tr>
<td>Description</td>
<td>Change the timer A setting value and set the DMAC II index.</td>
</tr>
<tr>
<td>Argument</td>
<td>None</td>
</tr>
<tr>
<td>Returned value</td>
<td>None</td>
</tr>
<tr>
<td>Remark</td>
<td></td>
</tr>
</tbody>
</table>
6.7 Flowcharts

6.7.1 Main Processing

Figure 6.4 shows the Main Processing.

```
main

(1) Disable maskable interrupts
   I flag ← 0
   PLL clock setting
   SetPLLclock()
   Set each clock frequency in PLL mode.

(2) DMAC II initial setting
   dmac2_init()

(3) Initial setting of three-phase
   motor control timers
   timer_init()

(4) Enable maskable interrupts
   I flag ← 1

(5) Start timer count
    (A1, A2, A4, B2)
    TABSR register ← 96h
    TA1S bit = 1
    TA2S bit = 1
    TA4S bit = 1
    TB2S bit = 1
```

Figure 6.4 Main Processing
### 6.7.2 DMAC II Initial Setting

Figure 6.5 shows the initial setting for DMAC II.

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. (1)</td>
<td>Set registers to use interrupt request level 7 for DMA II transfer</td>
</tr>
<tr>
<td>2. (2)</td>
<td>Set the DMAC II index</td>
</tr>
</tbody>
</table>

#### dmac2_init()

- **RIPL1** register ← 20h
  - FSIT bit = 0
  - DMAII = 1: Use interrupt request level 7 for DMA II transfer

- **RIPL2** register ← 20h
  - FSIT bit = 0
  - DMAII bit = 1: Use interrupt request level 7 for DMA II transfer

- **MOD** ← 006Fh
  - **SIZE** bit = 1: Selected transfer size is 16 bits
  - **IMM** bit = 1: Selected transfer source is memory
  - **UPDS** bit = 1: Selected source addressing is incrementing addressing
  - **UPDD** bit = 1: Selected destination addressing is incrementing addressing
  - **OPER** bit = 0: Calculation transfer not used
  - **BRST** bit = 1: Selected transfer is burst transfer
  - **INTE** bit = 1: DMA II transfer complete interrupt used
  - **CHAIN** bit = 0: Chained transfer not used
  - **MULT** bit = 0: Multiple transfer not used

- **COUNT** ← 6: Set number of transfers to 6
- **SADR** ← Set start address in the transfer source data area
- **DADR** ← TA1M register address: Set transfer destination address
- **IADR** ← Jump address for the DMA II transfer complete interrupt handler: Set the DMA II transfer complete interrupt

---

**RIPL1** register

- **FSIT** bit = 0
- **DMAII** = 1: Use interrupt request level 7 for DMA II transfer

**RIPL2** register

- **FSIT** bit = 0
- **DMAII** bit = 1: Use interrupt request level 7 for DMA II transfer

**MOD** ← 006Fh

- **SIZE** bit = 1: Selected transfer size is 16 bits
- **IMM** bit = 1: Selected transfer source is memory
- **UPDS** bit = 1: Selected source addressing is incrementing addressing
- **UPDD** bit = 1: Selected destination addressing is incrementing addressing
- **OPER** bit = 0: Calculation transfer not used
- **BRST** bit = 1: Selected transfer is burst transfer
- **INTE** bit = 1: DMA II transfer complete interrupt used
- **CHAIN** bit = 0: Chained transfer not used
- **MULT** bit = 0: Multiple transfer not used

**COUNT** ← 6: Set number of transfers to 6

**SADR** ← Set start address in the transfer source data area

**DADR** ← TA1M register address: Set transfer destination address

**IADR** ← Jump address for the DMA II transfer complete interrupt handler: Set the DMA II transfer complete interrupt
### 6.7.3 Initial Setting of the Three-Phase Motor Control Timers

Figure 6.6 and Figure 6.7 show the initial setting of the three-phase motor control timers.

1. Set the timer B2 interrupt generating frequency counter
   - ICTB2 register ← 02h

2. Enable writing to registers
   - PRCR register ← 02h
   - PRC1 bit = 1: Registers INVC0, INVC1, and IOBC are write enabled

3. Set the three-phase PWM control registers
   - INVC0 register ← 1Ch
     - Bits INV01 and INV00 = 00b: The underflow of timer B2
     - INV02 bit = 1: Use the three-phase motor control timers
     - INV03 bit = 1: Enables the three-phase motor control timer output
     - INV04 bit = 1: Disables simultaneous turn-on signal output
     - INV06 bit = 0: Triangular wave modulation mode
   - INVC1 register ← 42h
     - INV10 bit = 0: The underflow of timer B2
     - INV11 bit = 1: Three-phase mode 1
     - INV12 bit = 0: Dead time timer count source is set to f1
     - INV14 bit = 0: Active low output
     - INV15 bit = 0: Enables dead time
     - INV16 bit = 1: Rising edge of the three-phase output shift register (phases U, V, and W)

4. Set the three-phase output buffer control register
   - IOBC register ← 00h
     - SDE bit = 0: Disable shutdown
     - TBSOUT bit = 0: Use pins U, V, V, W, and W of ports P7 and P8

5. Disable writing to registers
   - PRCR register ← 00h
   - PRC1 bit = 0: Registers INVC0, INVC1, and IOBC are write disabled

6. Set the timer B2 special mode register
   - TB2SC register ← 00h
     - PWCON bit = 0: Reload the timer B2 register if timer B2 underflows

7. Set the three-phase output buffer registers
   - IDB0 register ← 2Ah
     - DU0 bit = 0: Active (ON)
     - DU0 bit = 1: Inactive (OFF)
     - DV0 bit = 0: Active (ON)
     - DV0 bit = 1: Inactive (OFF)
     - DW0 bit = 0: Active (ON)
     - DW0 bit = 1: Inactive (OFF)
   - IDB1 register ← 15h
     - DU1 bit = 1: Inactive (OFF)
     - DUB1 bit = 0: Active (ON)
     - DV1 bit = 1: Inactive (OFF)
     - DVB1 bit = 0: Active (ON)
     - DW1 bit = 1: Inactive (OFF)
     - DBW1 bit = 0: Active (ON)

---

Figure 6.6 Initial Setting of the Three-Phase Motor Control Timers (1/2)
Figure 6.7 Initial Setting of the Three-Phase Motor Control Timers (2/2)
6.7.4 DMAC II Transfer Complete Interrupt Handling

Figure 6.8 shows the transfer complete interrupt handling.

```
<table>
<thead>
<tr>
<th>dmac2_int()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change the timer A setting value flag</td>
</tr>
<tr>
<td>COUNT ← 6: Set the number of transfers to 6</td>
</tr>
<tr>
<td>SADR ← Set start address in the transfer source data area</td>
</tr>
<tr>
<td>DADR ← TA1M register address: Set transfer destination address</td>
</tr>
</tbody>
</table>

(2) Set the DMAC II index
```

Figure 6.8 DMAC II Transfer Complete Interrupt Handling
7. Sample Code
Sample code can be downloaded from the Renesas Electronics website.

8. Reference Documents
R32C/120 Group User's Manual: Hardware Rev.1.10
R32C/121 Group User's Manual: Hardware Rev.1.10
R32C/145 Group User's Manual: Hardware Rev.1.00
R32C/151 Group User's Manual: Hardware Rev.1.10
R32C/152 Group User's Manual: Hardware Rev.1.10
R32C/156 Group User's Manual: Hardware Rev.1.10
R32C/160 Group User's Manual: Hardware Rev.1.02
R32C/161 Group User's Manual: Hardware Rev.1.02
The latest versions can be downloaded from the Renesas Electronics website.

Technical Update/Technical News
The latest information can be downloaded from the Renesas Electronics website.

C Compiler Manual
R32C/100 Series C Compiler Package V.1.02
C Compiler User’s Manual Rev.2.00
The latest version can be downloaded from the Renesas Electronics website.

Website and Support
Renesas Electronics website
http://www.renesas.com/

Inquiries
http://www.renesas.com/contact/
<table>
<thead>
<tr>
<th>Rev.</th>
<th>Date</th>
<th>Page</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>Feb. 24, 2012</td>
<td>—</td>
<td>First edition issued</td>
</tr>
</tbody>
</table>

All trademarks and registered trademarks are the property of their respective owners.
General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

1. Handling of Unused Pins
Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.
- The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on
The state of the product is undefined at the moment when power is supplied.
- The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.
  In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.
  In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses
Access to reserved addresses is prohibited.
- The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals
After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.
- When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

5. Differences between Products
Before changing from one product to another, i.e. to one with a different part number, confirm that the change will not lead to problems.
- The characteristics of MPU/MCU in the same group but having different part numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different part numbers, implement a system-evaluation test for each of the products.
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 manufacturer, 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 depend 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.

8. 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 depend 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.

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 design safety. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by 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 reconfigurable computer 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.