Abstract
This document describes real-time clock operation using the timer function in the R32C/100 Series MCU.

Products
R32C/116 Group
R32C/117 Group
R32C/118 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. Hardware ........................................................................... 5  
   4.1 Pin Used ........................................................................ 5  
5. Software ............................................................................. 5  
   5.1 Operation Overview ........................................................... 5  
   5.2 Constants ....................................................................... 6  
   5.3 Variables ........................................................................ 7  
   5.4 Functions ....................................................................... 7  
   5.5 Function Specifications ..................................................... 8  
   5.6 Flowcharts ..................................................................... 11  
   5.6.1 Main Processing .............................................................. 11  
   5.6.2 Timer A0 Initial Setting .................................................... 12  
   5.6.3 Time Setting ................................................................. 13  
   5.6.4 Date Setting ................................................................. 14  
   5.6.5 Leap Day Determination ............................................... 15  
   5.6.6 Sub Clock Oscillation Setting ........................................ 16  
   5.6.7 Wait Mode Preset Processing ........................................ 18  
   5.6.8 Power Control Processing ............................................. 19  
   5.6.9 Timer A0 Interrupt Handling ........................................ 20  
6. Sample Code ..................................................................... 21  
7. Reference Documents ....................................................... 21
1. Specifications

This document describes real-time clock operation using timer A0 in timer mode. Enter wait mode after setting the operating mode to low power mode (base clock source is the sub clock). Use the timer A0 interrupt to exit wait mode. The date, day, and time data are updated in the timer A0 interrupt handler. The date, day, and time data start counting from 00:00:00, Saturday, January 1, 2000. The count continues until 23:59:59, Saturday, December 31, 2099, and then resets to the initial date and time. The counter takes the leap day into account.

Table 1.1 lists the Peripheral Functions and Their Applications. Figure 1.1 shows the Relationship Between Date, Day, and Time Data. Figure 1.2 shows the Transition Between Operating Modes.

<table>
<thead>
<tr>
<th>Peripheral Function</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timer A0 in timer mode</td>
<td>1 second counter</td>
</tr>
<tr>
<td>Timer A1</td>
<td>Generates wait time for sub clock oscillation</td>
</tr>
</tbody>
</table>

![Figure 1.1 Relationship Between Date, Day, and Time Data](image1.png)

![Figure 1.2 Transition Between Operating Modes](image2.png)
2. Operation Confirmation Conditions

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

Table 2.1 Operation Confirmation Conditions

<table>
<thead>
<tr>
<th>Item</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCU used</td>
<td>R5F64189DFD (R32C/118 Group)</td>
</tr>
<tr>
<td>Operating frequencies</td>
<td>When in PLL mode</td>
</tr>
<tr>
<td></td>
<td>• Main clock: 16 MHz</td>
</tr>
<tr>
<td></td>
<td>• PLL clock: 100 MHz</td>
</tr>
<tr>
<td></td>
<td>• Base clock: 50 MHz</td>
</tr>
<tr>
<td></td>
<td>• CPU clock: 50 MHz</td>
</tr>
<tr>
<td></td>
<td>• Peripheral bus clock: 25 MHz</td>
</tr>
<tr>
<td></td>
<td>• Peripheral function clock source: 25 MHz</td>
</tr>
<tr>
<td></td>
<td>When in low power mode</td>
</tr>
<tr>
<td></td>
<td>• Main clock is stopped</td>
</tr>
<tr>
<td></td>
<td>• PLL clock is stopped</td>
</tr>
<tr>
<td></td>
<td>• Base clock: 32.768 kHz (sub clock: 32.768 kHz)</td>
</tr>
<tr>
<td></td>
<td>• CPU clock: 32.768 kHz</td>
</tr>
<tr>
<td></td>
<td>• Peripheral bus clock: 16.384 kHz</td>
</tr>
<tr>
<td></td>
<td>• Peripheral function clock source is stopped</td>
</tr>
<tr>
<td>Operating voltage</td>
<td>5 V</td>
</tr>
<tr>
<td>Integrated development</td>
<td>Renesas Electronics Corporation</td>
</tr>
<tr>
<td>environment</td>
<td>High-performance Embedded Workshop Version 4.08</td>
</tr>
<tr>
<td>C compiler</td>
<td>Renesas Electronics Corporation</td>
</tr>
<tr>
<td></td>
<td>R32C/100 Series C Compiler V.1.02 Release 01</td>
</tr>
<tr>
<td>Compile options</td>
<td>Compile options</td>
</tr>
<tr>
<td></td>
<td>-D__STACKSIZE__=0X300 -D__ISTACKSIZE__=0X300</td>
</tr>
<tr>
<td></td>
<td>-DVECTOR_ADR=0xFFFFFBDC -c -finfo -dir &quot;$(CONFIGDIR)&quot;</td>
</tr>
<tr>
<td>(Default setting is used</td>
<td>in the integrated development environment.)</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 Entering Low-speed Mode (REJ05B1222-0100)
- R32C/100 Series Configuring Wait Mode (REJ05B1223-0100)
4. Hardware

4.1 Pin Used
Table 4.1 lists the Pin Used and Its Function.

<table>
<thead>
<tr>
<th>Pin Name</th>
<th>I/O</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>P0_0</td>
<td>Output</td>
<td>Confirm period for transition to wait mode</td>
</tr>
</tbody>
</table>

5. Software

5.1 Operation Overview
Enter wait mode after setting the operating mode to low power mode. With timer A0 in timer mode, use a timer A0 interrupt with a 1 second period to exit wait mode. The date, day, and time data are updated in the timer A0 interrupt handler. Leap day determination is performed when the month changes. After the data is updated, a WAIT instruction is used to enter wait mode again.

Settings:
- Timer used: Timer A0
- Mode used: Timer mode
- Timer A0 count source: fC32
- Interrupt used: Timer A0 interrupt
- Gate function: Not used

Formula to calculate a 1 second counter:

\[
1000 \text{ ms} = \left(1 + \frac{fC32}{32}\right) \times (TA0 + 1)
\]

\[
= \left(1 + \frac{32.768 \text{ kHz} + 32}{32}\right) \times 1024
\]

\[
= 0.9765625 \text{ ms} \times 1024
\]

(1) Initial setting
Set timer A0 and low power mode as the initial setting.

(2) Timer A0 count start
Set the TA0S bit in the TABSR register to 1 to start the timer A0 count. After the timer A0 count starts, the real-time clock starts counting from 00:00:00, Saturday, January 1, 2000.

(3) WAIT instruction execution
Execute the WAIT instruction to enter wait mode.

(4) Timer A0 interrupt generation
When timer A0 underflows, the timer A0 interrupt is generated and the MCU exits wait mode.

(5) Timer A0 interrupt handling
Date, day, and time data are updated in the timer A0 interrupt handler. However, the values are stored in the variable in hexadecimal.

(6) Date, day, and time data initialization
The count continues until 23:59:59, Saturday, December 31, 2099, and then resets to the initial date and time.
Figure 5.1 shows an Example of Real-Time Clock Operation.

![Diagram of Real-Time Clock Operation]

5.2 Constants

Table 5.1 lists the Constants Used in the Sample Code.

<table>
<thead>
<tr>
<th>Constant Name</th>
<th>Setting Value</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>YEAR_MAX</td>
<td>2099</td>
<td>Data for the maximum number of years</td>
</tr>
<tr>
<td>MONTH_MAX</td>
<td>12</td>
<td>Data for the maximum number of months</td>
</tr>
<tr>
<td>WEEK_MAX</td>
<td>6</td>
<td>Data for the maximum number of weeks in a month</td>
</tr>
<tr>
<td>HOUR_MAX</td>
<td>23</td>
<td>Data for the maximum number of hours</td>
</tr>
<tr>
<td>MIN_MAX</td>
<td>59</td>
<td>Data for the maximum number of minutes</td>
</tr>
<tr>
<td>SEC_MAX</td>
<td>59</td>
<td>Data for the maximum number of seconds</td>
</tr>
</tbody>
</table>

Note:
1. This bit becomes 0 when an interrupt request is accepted.

Figure 5.1 Example of Real-Time Clock Operation
5.3 Variables

Table 5.2 lists the Global Variables, and Table 5.3 lists the const Variable.

<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>year_cnt</td>
<td>Store data for number of years (2000 to 2099)</td>
<td>date_set, leap_day_check</td>
</tr>
<tr>
<td>unsigned char</td>
<td>month_cnt</td>
<td>Store data for number of months (1 to 12)</td>
<td>date_set, leap_day_check</td>
</tr>
<tr>
<td>unsigned char</td>
<td>day_cnt</td>
<td>Store data for number of days (1 to 31)</td>
<td>time_set, date_set, leap_day_check</td>
</tr>
<tr>
<td></td>
<td>week_cnt</td>
<td>Store data for day of the week</td>
<td>time_set, date_set</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0: Sunday</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1: Monday</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2: Tuesday</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3: Wednesday</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4: Thursday</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5: Friday</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>6: Saturday</td>
<td></td>
</tr>
<tr>
<td>unsigned char</td>
<td>hour_cnt</td>
<td>Store data for number of hours (00 to 23)</td>
<td>time_set</td>
</tr>
<tr>
<td>unsigned char</td>
<td>min_cnt</td>
<td>Store data for number of minutes (00 to 59)</td>
<td>time_set</td>
</tr>
<tr>
<td>unsigned char</td>
<td>sec_cnt</td>
<td>Store data for number of seconds (00 to 59)</td>
<td>time_set</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type</th>
<th>Variable Name</th>
<th>Contents</th>
<th>Function Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>const unsigned char</td>
<td>day_max_tbl[12]</td>
<td>Data table for the maximum number of days per month</td>
<td>date_set</td>
</tr>
</tbody>
</table>

5.4 Functions

Table 5.4 lists the Functions.

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Outline</th>
</tr>
</thead>
<tbody>
<tr>
<td>timer_a0_int</td>
<td>Timer A0 initial setting</td>
</tr>
<tr>
<td>time_set</td>
<td>Time setting</td>
</tr>
<tr>
<td>date_set</td>
<td>Date setting</td>
</tr>
<tr>
<td>leap_day_check</td>
<td>Leap day determination</td>
</tr>
<tr>
<td>subclock_set</td>
<td>Sub clock oscillation setting</td>
</tr>
<tr>
<td>before_wait_mode_set</td>
<td>Wait mode preset processing</td>
</tr>
<tr>
<td>power_control</td>
<td>Power control processing</td>
</tr>
<tr>
<td>_timer_a0</td>
<td>Timer A0 interrupt handling</td>
</tr>
</tbody>
</table>
5.5 Function Specifications
The following tables list the sample code function specifications.

<table>
<thead>
<tr>
<th>Function</th>
<th>Outline</th>
<th>Header</th>
<th>Declaration</th>
<th>Description</th>
<th>Argument</th>
<th>Returned value</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>timer_a0_int</td>
<td>Timer A0 initial setting</td>
<td>None</td>
<td>void timer_a0_init(void)</td>
<td>Initial settings to use timer A0 in timer mode.</td>
<td>None</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>time_set</td>
<td>Time setting</td>
<td>None</td>
<td>void time_set(void)</td>
<td>After setting the data for the seconds, minutes, hours, date, and day of the week, the date setting function is called.</td>
<td>None</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>date_set</td>
<td>Date setting</td>
<td>None</td>
<td>void date_set(void)</td>
<td>The leap day determination function is called, and depending on the result, the date, month, and year data is set.</td>
<td>None</td>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>
### leap_day_check

<table>
<thead>
<tr>
<th>Outline</th>
<th>Leap day determination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>None</td>
</tr>
<tr>
<td>Declaration</td>
<td>unsigned char leap_day_check(void)</td>
</tr>
</tbody>
</table>
| Description      | • Leap day is determined.  
                     • Years that can be evenly divided by 4 are determined to have the leap day (February 29). However, years that can be divided by 4 or divided by 100, but cannot be divided by 400 are determined to not have a leap day. |
| Argument         | None                   |
| Returned value   | • Leap day: 1  
                     • Not a leap day: 0 |
| Remark           |                        |

### subclock_set

<table>
<thead>
<tr>
<th>Outline</th>
<th>Sub clock oscillation setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>None</td>
</tr>
<tr>
<td>Declaration</td>
<td>void subclock_set(void)</td>
</tr>
<tr>
<td>Description</td>
<td>After setting both bits PD8_6 and PD8_7 in the PD8 register to 0 (input mode) and the PU25 bit in the PUR2 register to 0 (pull-up resistor unused), set the CM04 bit in the CM0 register to 1 (XIN-XCIN oscillator).</td>
</tr>
<tr>
<td>Argument</td>
<td>None</td>
</tr>
<tr>
<td>Returned value</td>
<td>None</td>
</tr>
<tr>
<td>Remark</td>
<td>Set the oscillation stabilization time according to the manufacturer’s recommendation.</td>
</tr>
</tbody>
</table>

### before_wait_mode_set

<table>
<thead>
<tr>
<th>Outline</th>
<th>Wait mode preset processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>None</td>
</tr>
<tr>
<td>Declaration</td>
<td>void before_wait_mode_set(void)</td>
</tr>
</tbody>
</table>
| Description      | • Change the base clock source from the PLL clock to the sub clock and enter low speed mode.  
                     • Stop the main clock and PLL clock, and transition from low speed mode to low power mode. |
| Argument         | None                          |
| Returned value   | None                          |
| Remark           |                                |
### power_control

<table>
<thead>
<tr>
<th>Outline</th>
<th>Power control processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>None</td>
</tr>
<tr>
<td>Declaration</td>
<td>void power_control(void)</td>
</tr>
<tr>
<td>Description</td>
<td>Set the interrupt priority level for wake-up, execute the WAIT instruction, and enter wait mode.</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>

### _timer_a0

<table>
<thead>
<tr>
<th>Outline</th>
<th>Timer A0 interrupt handling</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header</td>
<td>None</td>
</tr>
<tr>
<td>Declaration</td>
<td>void _timer_a0(void)</td>
</tr>
<tr>
<td>Description</td>
<td>Call the timer_set function in the interrupt handler.</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>
5.6 Flowcharts

5.6.1 Main Processing

Figure 5.2 shows the Main Processing.

```
main

Disable maskable interrupts
I flag ← 0

PLL clock setting
SetPLLClock()

Initialize port P0_0

Timer A0 initial setting
timer_a0_init()

Sub clock oscillation setting
subclock_set()

Wait mode
preset processing
before_wait_mode_set()

Initialize the fc
divide-by-32 divider

Timer A0 count starts

Enable maskable interrupts
I flag ← 1

Power control processing
power_control()```

Clock frequencies used in PLL mode are set.

P0 register
P0_0 bit ← 1: Output high

PD0 register
PD0_0 bit ← 1: Output port

CPSRF register
CPSR bit ← 1

TA0S register
TA0S bit ← 1

Figure 5.2 Main Processing
5.6.2 Timer A0 Initial Setting

Figure 5.3 shows the Timer A0 Initial Setting.

- Set the interrupt priority level for wake-up
  - RIPL1 register ← 07h: Level 7
  - RIPL2 register ← 07h: Level 7
  - Bits RLVL2 to RLVL0 = 111b

- Stop timer A0 count
  - TABSR register
    - TA0S bit ← 0

- Disable timer A0 interrupt
  - TA0IC register ← 00h
  - Bits ILVL2 to ILVL0 = 000b: Level 0 (interrupt disabled)
  - IR bit = 0: No interrupt requested

- Set timer A0 mode register
  - TA0MR register ← C0h
    - Bits TMOD1 and TMOD0 = 00b: Timer mode
    - Bits MR2 and MR1 = 00b: No gate function
    - Bits TCK1 and TCK0 = 11b: fC32

- Set timer A0 register
  - TA0 register ← 1024 - 1: 1 second

- return
5.6.3 Time Setting

Figure 5.4 shows the Time Setting.

```
Figure 5.4 Time Setting
```

```
<table>
<thead>
<tr>
<th>time_set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Update data for number of seconds</td>
</tr>
<tr>
<td>Is number of seconds more than 59?</td>
</tr>
<tr>
<td>Yes</td>
</tr>
<tr>
<td>Update data for number of minutes</td>
</tr>
<tr>
<td>Set number of seconds to 00</td>
</tr>
<tr>
<td>No</td>
</tr>
<tr>
<td>Is number of minutes more than 59?</td>
</tr>
<tr>
<td>Yes</td>
</tr>
<tr>
<td>Update data for number of hours</td>
</tr>
<tr>
<td>Set number of minutes to 00</td>
</tr>
<tr>
<td>No</td>
</tr>
<tr>
<td>Is number of hours more than 23?</td>
</tr>
<tr>
<td>Yes</td>
</tr>
<tr>
<td>Update data for date</td>
</tr>
<tr>
<td>Update data for day of the week</td>
</tr>
<tr>
<td>Set number of hours to 00</td>
</tr>
<tr>
<td>Date setting date_set()</td>
</tr>
<tr>
<td>Has 1 week elapsed?</td>
</tr>
<tr>
<td>No</td>
</tr>
<tr>
<td>Yes</td>
</tr>
<tr>
<td>Set day of the week data to Sunday</td>
</tr>
<tr>
<td>return</td>
</tr>
</tbody>
</table>
```
### 5.6.4 Date Setting

Figure 5.5 shows the Date Setting.

![Flowchart for Date Setting](chart.png)

**Figure 5.5 Date Setting**
5.6.5 Leap Day Determination

Figure 5.6 shows the Leap Day Determination.

```
leap_day_check
  Set 0 to the return value (not a leap day)

  Is the date February 29?
    Yes
      Can the year be divided by 4?
        Yes
          Set 1 (leap day) to the results
        No
          Can the year be divided by 100?
            Yes
              Set 0 (not a leap day) to the results
            No
              Can the year be divided by 400?
                Yes
                  Set 1 (leap day) to the results
                No
                  return
```

Figure 5.6 Leap Day Determination
Fig. 5.7 and Fig. 5.8 show the sub clock oscillation setting.

```
subclock_set

Stop timer A1 count
TA1S bit ← 0

Enable writing to register

Enable writing to register

Set processor mode register 2
PM2 register
PM26 bit ← 1: f2n clock source is the main clock

Enable writing to register

Disable writing to register

Set processor mode register 2
PM2 register
PM26 bit ← 1: f2n clock source is the main clock

Set processor mode register 2
PM2 register
PM26 bit ← 1: f2n clock source is the main clock

Stop timer A1 count
TA1S bit ← 0

Disable timer A1 interrupt
TA1IC register ← 00h
Bits ILVL2 to ILVL0 = 000b

Enable writing to register

Enable writing to register

Set count source prescaler register

TCSPR register
CST bit ← 0: Stop divider operation
TCSPR register ← 0Ah
Bits CNT3 to CNT0 = 1010b: Main clock divided by 2^n where n=10
TCSPR register
CST bit ← 1: Start divider operation

TA1MR register ← 80h
Bits TMOD1 and TMOD0 = 00b: Timer mode
Bits MR2 and MR1 = 00b: No gate function
Bits TCK1 and TCK0 = 10b: f2n

Set timer A1 mode register

TA1 register ← 40000 - 1: 50 ms

Set XCIN-XCOUT pins

PD8 register
PD8_6 bit ← 0: Input port
PD8_7 bit ← 0: Input port
PUR2 register
PU25 bit ← 0: P8_6 and P8_7 pull-up resistor disabled

Enable writing to register

Enable writing to register

Set system clock control register 0
CM0 register
CM03 bit ← 1: XCIN-XCOUT drive strength high
CM04 bit ← 1: XCIN-XCOUT oscillator
CM05 bit ← 0: Main clock oscillator enabled

Enable writing to register

Enable writing to register

Start timer A1 count

Wait for sub clock oscillator to stabilize

Stop timer A1 count

Disable timer A1 interrupt
TA1IC register ← 00h
Bits ILVL2 to ILVL0 = 000b
```

Figure 5.7 Sub Clock Oscillation Setting (1/2)
Figure 5.8 Sub Clock Oscillation Setting (2/2)

1. Set count source prescaler register
2. Enable writing to register
3. Set processor mode register 2
4. Disable writing to register
5. return

TCSPR register
CST bit ← 0: Stop divider operation

PRCR register
PRC1 bit ← 1: Enable to write to PM2 register

PM2 register
PM26 bit ← 0: f2n clock source is peripheral clock source

PRCR register
PRC1 bit ← 0: Disable to write to PM2 register
5.6.7 Wait Mode Preset Processing
Figure 5.9 shows Wait Mode Preset Processing.

![Diagram of Wait Mode Preset Processing]

Note:
1. Perform this processing when using an embedded CPU rewrite program.

Note 1:
before_wait_mode_set

Enable writing to register

PRC2R register
PRC27 bit ← 1: Enable to write to the CM3 register

Set low speed mode base clock
CM3 register ← 00h
Bits CM31 and CM30 = 00b: fC

Disable writing to register

Enable writing to register

PRC2R register
PRC27 bit ← 0: Disable to write to the CM3 register

Enable writing to register

PRR register ← AAh: Enable to write to the CCR register

Set base clock source
CCR register
BCS bit ← 1: fC

Disable writing to register

Enable writing to registers

Set system clock
control register 0
CM0 register
CM03 bit ← 0: XCIN-XCOUT drive strength low
CM05 bit ← 1: Main clock oscillator disabled

Stop PLL oscillator

CM1 register
CM10 bit ← 1: PLL oscillator disabled

Disable writing to registers

Enable writing to registers

PRC0 bit ← 1: Enable to write to registers CM0 and CM1

Stop PLL oscillator CM1 register
CM10 bit ← 1: PLL oscillator disabled

Disable writing to registers

Enable writing to register

PRR register ← AAh: Enable to write to the FMCR register

Set CPU rewrite mode
FMCR register
FEW bit ← 0: Normal operating mode

Disable writing to register

Enable writing to register

PRR register ← 00h: Disable to write to the FMCR register

Enable writing to register

PRCR3 register
PRC31 bit ← 1: Enable to write to VRCR register

Main regulator shut-down
VRCR register
MRS bit ← 1: Main regulator stopped

Disable writing to register

PRCR3 register
PRC31 bit ← 0: Disable to write to VR CR register

return

(See Note 1)
5.6.8 Power Control Processing

Figure 5.10 shows the Power Control Processing.

![Flowchart showing the Power Control Processing](image-url)

- **power_control**
  - Disable maskable interrupts
    - I flag ← 0
  - Disable the timer A0 interrupt
    - TA0IC register ← 00h
    - Bits ILVL2 to ILVL0 = 000b
  - Perform a dummy read
  - Set the processor interrupt priority level to 0
  - Enable maskable interrupts
    - I flag ← 1
  - Insert two NOP instructions
  - Disable maskable interrupts
    - I flag ← 0
  - Enable timer A0 interrupt
    - TA0IC register ← 07h
    - Bits ILVL2 to ILVL0 = 111b: Level 7
  - Set the processor interrupt priority level to 3
  - Set the wake-up IPL setting registers
    - RIPL1 register ← 03h
    - Bits RLVL2 to RLVL0 = 011b: Level 3
    - RIPL2 register ← 03h
    - Bits RLVL2 to RLVL0 = 011b: Level 3
  - Enable writing to register
  - PRC0 bit ← 1: Enable to write to CM0 register
  - Set system clock control register 0
  - CM0 register
    - CM02 bit ← 1: Peripheral clock source stopped in wait mode
  - Disable writing to register
  - Enable maskable interrupts
    - I flag ← 1
  - Set port P0_0 to output low
    - P0 register
      - P0_0 bit ← 0
  - Execute WAIT instruction
  - Set port P0_0 to output high
    - P0 register
      - P0_0 bit ← 1

**Figure 5.10** Power Control Processing
5.6.9 Timer A0 Interrupt Handling

Figure 5.11 shows the Timer A0 Interrupt Handling.

RIPL1 register ← 07h
- Bits RLVL2 to RLVL0 = 111b: Level 7
RIPL2 register ← 07h
- Bits RLVL2 to RLVL0 = 111b: Level 7

Figure 5.11 Timer A0 Interrupt Handling
6. Sample Code
Sample code can be downloaded from the Renesas Electronics website.

7. Reference Documents
R32C/118 Group User's Manual: Hardware Rev.1.10
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>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>Aug. 24, 2012</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.

<table>
<thead>
<tr>
<th>1. Handling of Unused Pins</th>
</tr>
</thead>
<tbody>
<tr>
<td>Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.</td>
</tr>
<tr>
<td>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.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2. Processing at Power-on</th>
</tr>
</thead>
<tbody>
<tr>
<td>The state of the product is undefined at the moment when power is supplied.</td>
</tr>
<tr>
<td>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.</td>
</tr>
<tr>
<td>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.</td>
</tr>
<tr>
<td>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.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3. Prohibition of Access to Reserved Addresses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access to reserved addresses is prohibited.</td>
</tr>
<tr>
<td>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.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4. Clock Signals</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
</tr>
<tr>
<td>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.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5. Differences between Products</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
</tr>
<tr>
<td>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.</td>
</tr>
</tbody>
</table>
Notice

1. 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.

2. 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.

3. 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.

4. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.

5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.

"Standard": Computers, office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronics appliances; machine tools; personal electronic equipment; and industrial robots etc.

"High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.

Renesas Electronics products are neither intended nor authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems, surgical implants etc.) or may cause serious property damage (nuclear reactor control systems, military equipment etc.). 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 for which it is not intended. 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 which the product is not intended by Renesas Electronics.

6. 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.

7. 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 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 first products or systems manufactured by you.

8. 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.

9. Renesas Electronics products and technology may not be used for or incorporated into any products or systems where manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You should not use Renesas Electronics products or 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. When exporting the Renesas Electronics 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.

10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.

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.