To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

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

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

H8/300L Super Low Power Series

Using Timer F to Implement Clock Operation

Introduction

A 32.768-kHz subclock (ϕ_W) is used for clock operation using timer F. Timer F interrupts are generated every 1 s to count up the counter for clock use in RAM.

When the counter is counted until 59 min. 59 s, the next count is initialized.

Target Device

H8/38024

Contents

1. Specifications

1. A 32.768-kHz subclock (ϕ_W) is used for clock operation using timer F.
2. Timer F interrupts are generated every 1 s, and a counter provided for clock use in RAM is incremented.
3. The clock counter provided in RAM has eight bits for counting seconds and eight bits for counting minutes. It starts counting from 00 min, 00 s, and after counting up to 59 min, 59 s, it is initialized to 00 min, 00 s in the next cycle and restarts counting.
4. After completion of initialization, a transition is made from active (high-speed) mode to watch mode. A timer F interrupt request causes a transition to subactive mode, the counter provided in RAM is incremented, and a transition is made again to watch mode.
5. The mode transition diagram for this sample task is shown in figure 1.

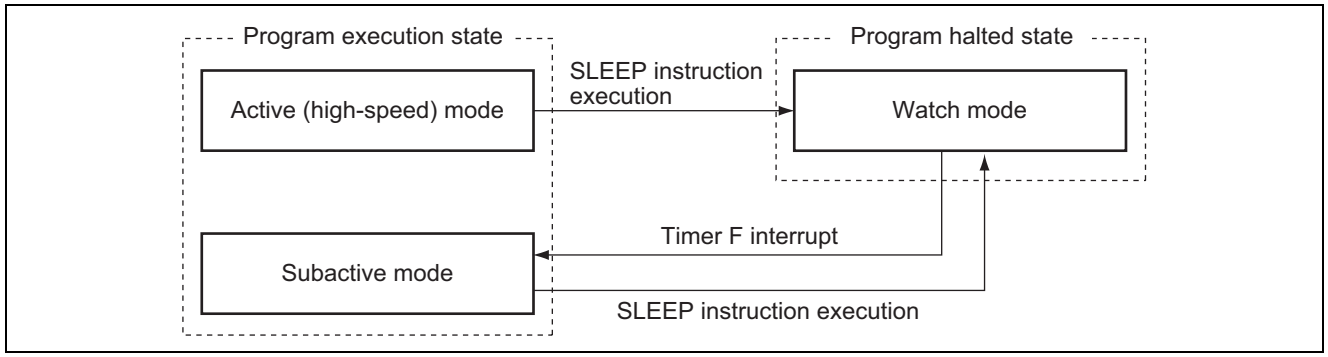


Figure 1 Diagram of Mode Transitions for this Sample Task

2. Description of Functions Used

1. In this sample task, clock operation is performed in which timer F is used to increment a counter provided in RAM every 1s. The features of timer F are as follows.
   - The system clock (ϕ) is a 5-MHz clock and a reference clock to operate the CPU and its peripheral functions.
   - The prescaler S (PSS) is a 13-bit counter using ϕ as its input clock and counted up every cycle.
   - ϕ_w is the output clock (32.768 kHz) of the subclock oscillator.
   - The prescaler W (PSW) is a 5-bit counter using a clock input obtained by dividing ϕ_w by 4 ($\phi_w/4$).
   - Four internal clocks ($\phi/32$, $\phi/16$, $\phi/4$, $\phi_w/4$) or an external clock can be selected (external event counting is possible).
   - A single compare-match signal can be used to output a toggle signal to the TMOFH pin (the initial value of the toggle output can be set).
   - The counter can be reset by a compare-match signal.
   - There are a total of two interrupt factors: one is compare-match, and the other is overflow.
   - Operation as two independent 8-bit timers (timer FH and timer FL) is also possible (in 8-bit mode).
   - When $\phi_w/4$ is selected as the internal clock, operation in watch mode, subactive mode, or sleep mode is possible.
   - Using the module standby mode, it is possible to set the module in standby mode independently when not in use.

2. Figure 2 shows a block diagram of the 16-bit output compare function of timer F used in this sample task.

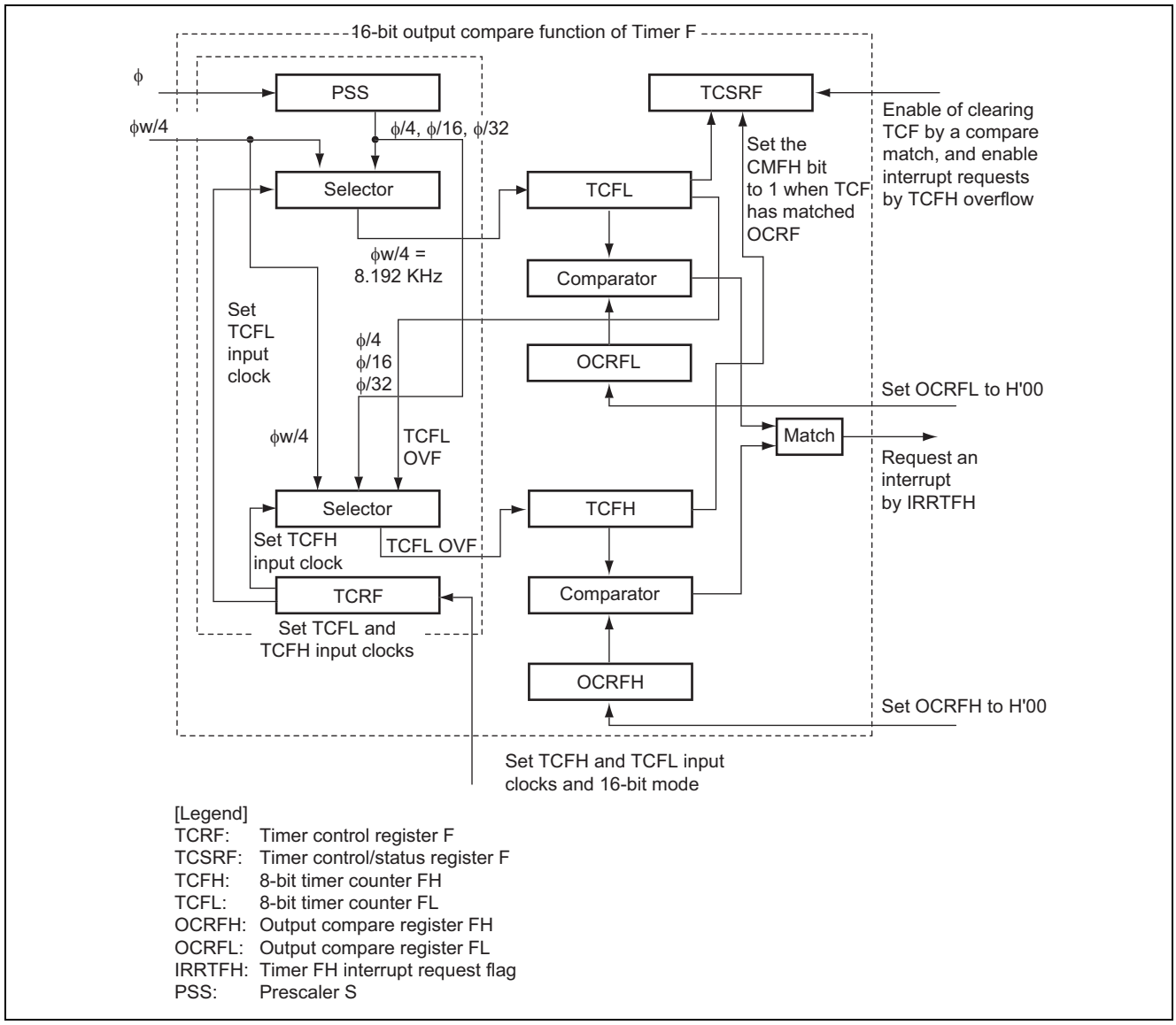


Figure 2 Block Diagram of 16-Bit Output Compare Function of Timer F

3. Table 1 shows timer F functions.

Table 1 Timer F Functions

Register	Functions
Timer control register F (TCRF)	TCRF is an 8-bit write-only register. It switches between 16-bit mode and 8-bit mode, selects the clock source from among four types of internal clocks and an external event, and sets the output levels of the TMOFH and TMOFL pins. On reset, TCRF is initialized to H'00.
Timer control status register F (TCSRF)	TCSRF is an 8-bit readable/writable register. It selects counter clear, sets the overflow flag, sets the compare match flag, and controls enabling interrupt requests due to overflows.
16-bit timer counter F (TCF)	TCF is a 16-bit readable/writable up-counter. It consists of cascade connections of the 8-bit timer counters (TCFH and TCFL). In addition to use as a 16-bit counter employing TCFH for the upper eight bits and TCFL for the lower eight bits, TCFH and TCFL can also be used as independent 8-bit counters. TCFH and TCFL can be read and written by the CPU, but when used in 16-bit mode, data transfer with the CPU is via a temporary register (TEMP). On reset, TCFH and TCFL are both initialized to H'00. When the CKSH2 bit in TCRF is cleared to 0, then TCF functions as a 16-bit counter. The TCF input clock can be selected using the CKSL2 to CKSL0 bits in TCRF. The CCLRH bit in TCSRF can be used to clear TCF on compare match. When TCF overflows, the OVFH bit in TCSRF is set to 1, and when the IENTFH bit in IENR2 is 1, an interrupt request is sent to the CPU.
8-bit timer counter FH (TCFH)	TCFH can be made to operate as an independent 8-bit counter by setting the CKSH2 bit in TCRF to 1. The TCFH input clock is selected using the CKSH2 to CKSH0 bits in TCRF. The CCLRH bit in TCSRF can be used to clear TCFH on a compare match. When there is an overflow of TCFH, the OVFH bit in TCSRF is set to 1. At this time, when the OVIEH bit in TCSRF is 1, the IRRTFH bit in IRR2 is set to 1, and when the IENTFH bit in IENR2 is 1, an interrupt request is sent to the CPU.
8-bit timer counter FL (TCFL)	TCFL can be made to operate as an independent 8-bit counter by setting the CKSH2 bit in TCRF to 1. The TCFL input clock is selected using the CKSL2 to CKSL0 bits in TCRF. The CCLRL bit in TCSRF can be used to clear TCFL on a compare match. When there is an overflow in TCFL, the OVFL bit in TCSRF is set to 1. At this time, if the OVIEL bit in TCSRF is 1, the IRRTFL bit in IRR2 is set to 1. When the IENTFL bit in IENR2 is 1, an interrupt request is sent to the CPU.

Register	Functions
16-bit output compare register F (OCRF)	OCRF is a 16-bit readable/writable register which consists of two registers (OCRFH and OCRFL). In addition to being used as a 16-bit register of which OCRFH is the upper eight bits and OCRFL is the lower eight bits, OCRFH and OCRFL can also be used as independent 8-bit registers. OCRFH and OCRFL can be read and written by the CPU, but when used in 16-bit mode, data transfer with the CPU is via TEMP. On reset, OCRFH and OCRFL are both initialized to H'FF. On clearing the CKSH2 bit in TCRF to 0, OCRF functions as a 16-bit register. The contents of OCRF are constantly compared with TCF, and when the values of the two match, the CMFH bit in TCSRF is set to 1. At the same time, the IRRTFH bit in IRR2 is also set to 1. At this time when the IENTFH bit in IENR2 is 1, an interrupt request is sent to the CPU. The toggle signal of a compare match can be output from the TMOFH pin. In addition, the TOLH bit in TCRF can be used to select the output level (high/low).
8-bit output compare register FH (OCRFH)	When the CKSH2 bit in TCRF is set to 1, OCRF functions as two 8-bit registers. The contents of OCRFH are compared with TCFH, and the contents of OCRFL are compared with TCFL. When the values of OCRFH and TCFH match, the CMFH bit in TCSRF is set to 1. At the same time, the IRRTFH bit in IRR2 is also set to 1. At this time when the IENTFH bit in IENR2 is 1, an interrupt request is sent to the CPU. The toggle signal of a compare match can be output from the TMOFH pin. In addition, the TOLH bit of TCRF can be used to select the output level (high/low).
8-bit output compare register FL (OCRFL)	When the CKSH2 bit in TCRF is set to 1, OCRF functions as two 8-bit registers. The contents of OCRFH are compared with TCFH, and the contents of OCRFL are compared with TCFL. When the values of OCRFL and TCFL match, the CMFL bit in TCSRF is set to 1. At the same time, the IRRTFL bit in IRR2 is also set to 1. At this time when the IENTFL bit in IENR2 is 1, an interrupt request is sent to the CPU. The toggle signal of a compare match can be output from the TMOFL pin. In addition, the TOLL bit of TCRF can be used to select the output level (high/low).
Timer FH interrupt request flag (IRRTFH)	IRRTFH is set to 1, when TCF matches OCRF in 16-bit mode, when TCFH matches OCRFH in 8-bit mode, or when TCF and TCFH overflow in the state that IENTFH is set to 1. IRRTFH is cleared to 0, when IRRTFH is written to 1 in the state that IRRTFH is set to 1.
Timer FL interrupt request flag (IRRTFL)	IRRTFL is set to 1, when TCFL matches OCRFL in 8-bit mode or when TCFL overflows in the state that IENTFL is set to 1. IRRTFL is cleared to 0, when IRRTFL is written to 1 in the state that IRRTFL is set to 1.
Timer FH interrupt enable (IENTFH)	IENTFH enables or disables interrupt requests caused by timer FH compare match or overflow.
Timer FL interrupt enable (IENTFL)	IENTFL enables or disables interrupt requests caused by timer FL compare match or overflow.
Timer F event input (TMIF)	TIMF is used as an event input pin to TCFL.
Timer FH output (TMOFH)	TMOFH is a timer FH toggle output pin.
Timer FL output (TMOFL)	TMOFL is a timer FL toggle output pin.

4. The method for setting the timer FH interrupt cycle is described below.
   In this sample task, 32.768 kHz is used for ϕ_w, and the timer F functions as a clock time base.
   By setting CKSL2 to 1, CKSL1 to 1 and CKSL0 to 1 in TCRF, the TCF input clock is set to $\phi_w/4$. Here $\phi_w/4$ is given by the equation below
   $\phi_w/4$ = 32.768 kHz/4 = 8.192 kHz
   Hence the TCF input clock cycle is
   1/8.912 kHz ≅ 122.07 µs
   Here when OCRF is set to H'2000, then the time until the values of TCF and OCRF match is calculated as
   H'2000 × (1/8.192 kHz) = 8192 × 122.07 µs = 1 s
   Hence the settings for OCRF used to set the timer FH interrupt cycle T_{FH} are calculated by the following equation.
   OCRF setting = T_{FH} / (1/8.192 kHz) = T_{FH} × 8.192 kHz
   Table 2 shows timer FH interrupt cycle T_{FH} values and OCRF setting examples.

Table 2 Examples of Timer FH Interrupt Cycles and OCRF Settings

T_{FH} (s)	Calculation	OCRF Setting
0.125	0.125 s × 8.192 kHz = 1024	H'0400
0.250	0.250 s × 8.192 kHz = 2048	H'0800
0.500	0.500 s × 8.192 kHz = 4096	H'1000
1.000	1.000 s × 8.192 kHz = 8192	H'2000
2.000	2.000 s × 8.192 kHz = 16384	H'4000

5. The interface with the CPU is described below.
   TCF and OCRF are 16-bit readable/writable registers. On the other hand, the data bus between the CPU and internal peripheral modules has an 8-bit data width. Hence when the CPU accesses TCF or OCRF, it must do so via the 8-bit temporary register (TEMP).
   When reading or writing TCF or writing OCRF in 16-bit mode, operations must always be performed in 16-bit units (with two consecutive byte-size MOV instructions), in the order of the upper byte firstly and lower byte secondly. When only the upper byte or only the lower byte is accessed, the data is not transferred correctly.
   In 8-bit mode, there are no restrictions on access order.
   a. Write operation
      By writing the upper byte, the upper byte data is transferred to TEMP. Next the lower byte is written; the data in TEMP is written to the upper byte register, and the lower-byte data is written directly to the lower byte register.
      The TCF write operation when H'AA55 is written to TCF is shown in Figure 3.

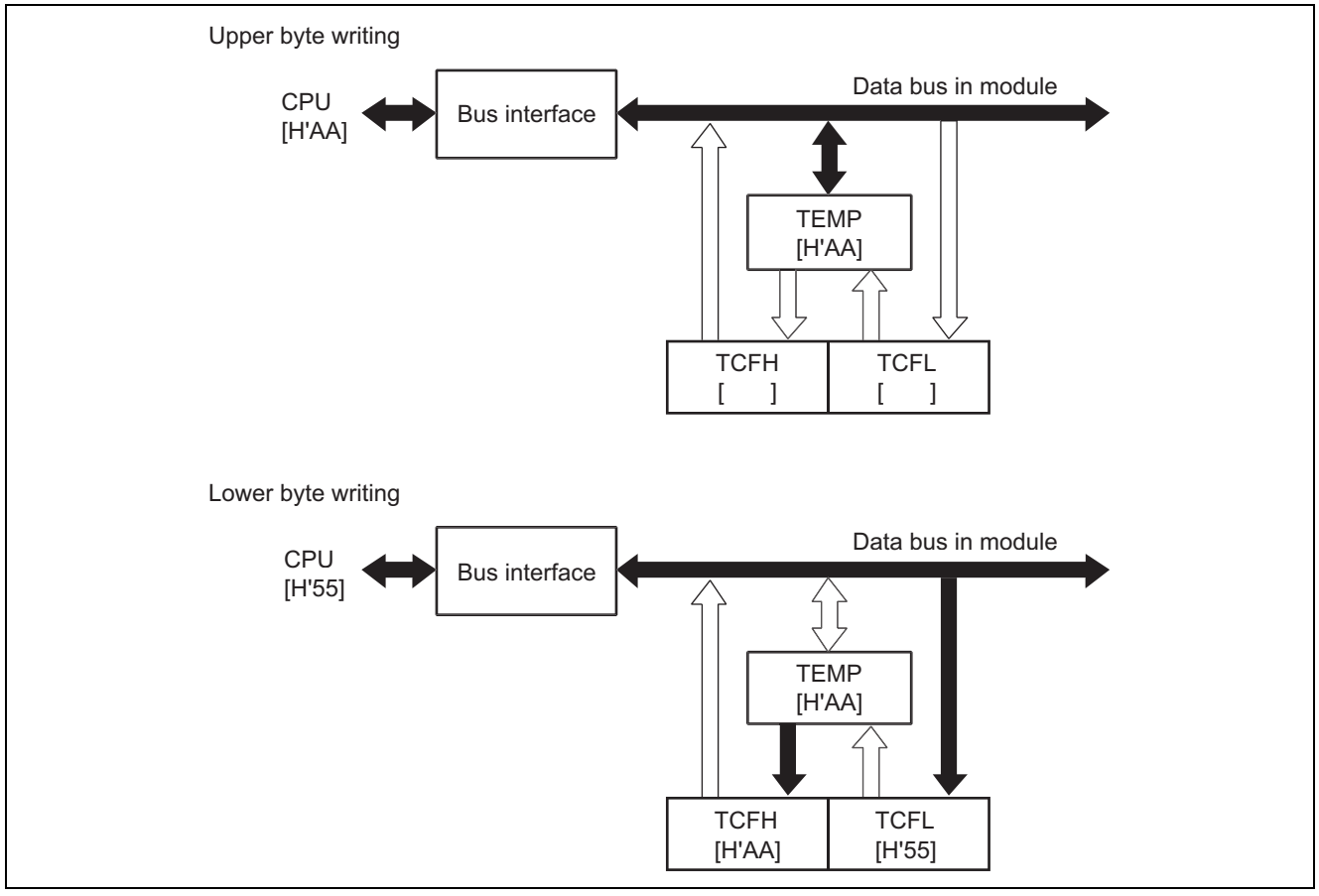


Figure 3 TCF Write Operation

b. Read operation

In the case of TCF, when the upper byte is read, the upper byte data is transferred directly to the CPU, and the lower byte data is transferred to TEMP. Next the lower byte data is read; the lower byte data in TEMP is transferred to the CPU.

In the case of OCRF, in upper byte reading the upper byte data is transferred directly to the CPU. In lower byte reading the lower byte data is also transferred directly to the CPU.

Figure 4 shows a TCF read operation when TCF contains H'AAFF

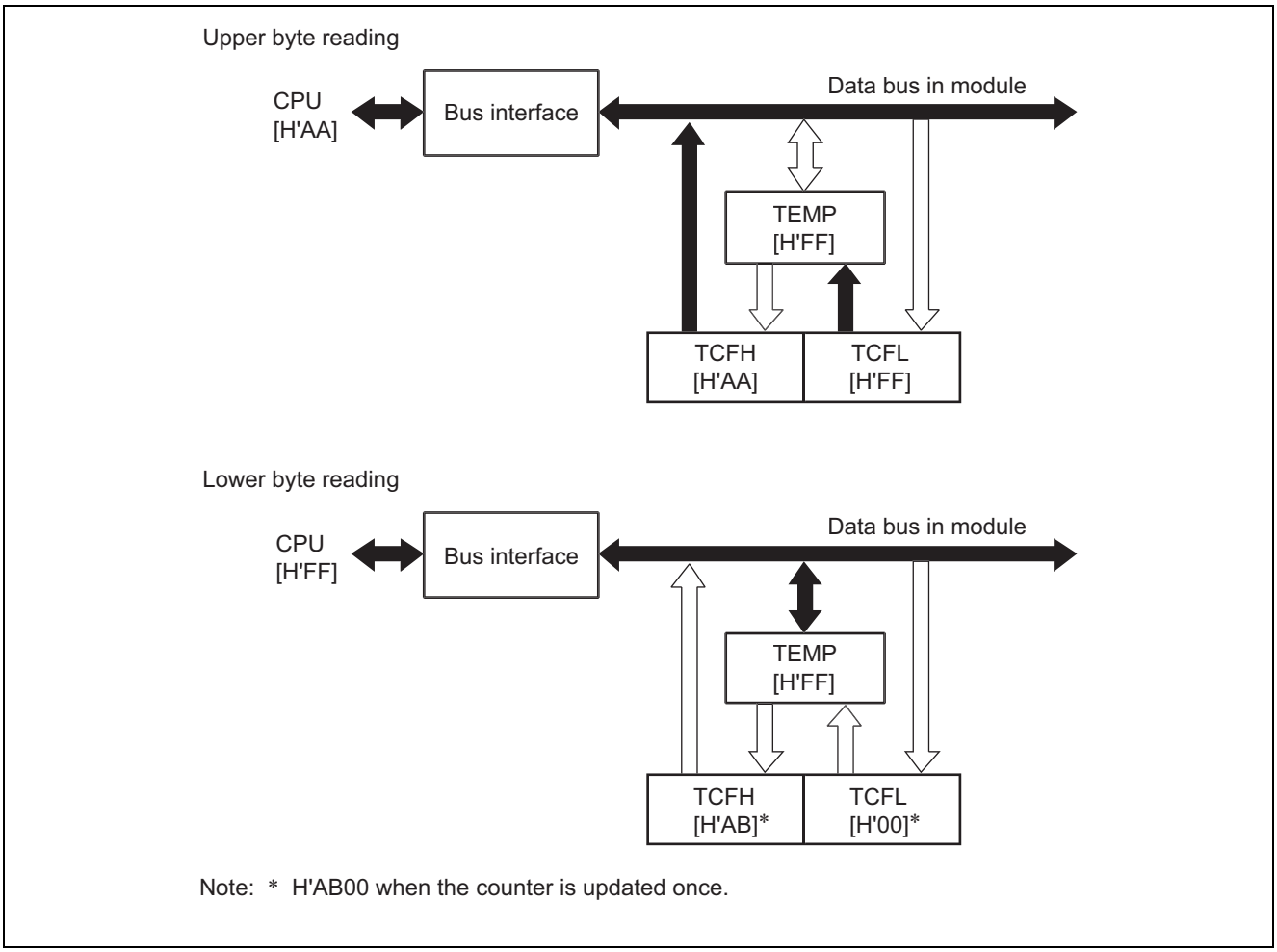


Figure 4 TCF Read Operation

6. Notes on use of the timer F
   While the timer F is operating in 16-bit timer mode, the following conflicts and operations may occur.
   a. When all 16 bits match and a compare match signal is generated, the toggle signal is output from the TMOFH pin. When TCRF writing by a MOV instruction and a compare match signal occur simultaneously, TOLH data resulting from TCRF writing is output to TMOFH pin. In 16-bit mode, the TMOFL pin output is unstable, and should not be used. Use it as a port instead.
   b. When OCRFL writing and compare match signal generation occur simultaneously, the compare match signal is invalid. However, when the written data compare matches the counter value, at that point a compare match signal is generated. The compare match signal is output in synchronization with the TCFL clock, so that when the clock is stopped, no compare match signal is generated even when a compare match occurs.
   c. When all 16 bits match and a compare match signal is generated, the compare match flag CMFH is set. Similarly, when setting conditions for the lower 8 bits are satisfied, CMFL is set.
   d. When there is a TCF overflow, OVFH is set; when the lower 8 bits of OVFL overflow, when setting conditions are satisfied, OVFL is set. When TCFL writing and overflow signal output occur simultaneously, the overflow signal is not output.
   e. In active mode and sleep mode, when $\phi w/4$ is selected as the TCF internal clock, synchronization is established by the synchronization circuit since the system clock and internal clock are mutually asynchronous, and so synchronization is established by asynchronization. This results in a maximum count cycle error of $1/\phi$ (sec). In order to prevent this error from occurring, the system must be operated in subactive mode, subsleep mode, or watch mode.

7. Table 3 shows the assignment of functions in this sample task.

Table 3 Function Assignment

Function	Assignment
TCRF	Sets 16-bit mode and selects TCFL input clock.
TCSRF	Selects counter clear, sets the overflow flag, sets the compare match flag, and enables/disables interrupt requests due to overflows.
TCF	Functions as a 16-bit up-counter configured by TCFH and TCFL. TCF counts with internal clock ($\phi_W/4$), and sets the IRTFH and CMFH bits to 1 when a compare match occurs.
OCRF	16-bit register configured by OCRFH and OCRFL. When the value set in OCRF matches the one in TCF, a compare match signal is generated.
SYSCR1	Controls power down modes.
SYSCR2	Controls power down modes.
IRRTFH	Indicates whether or not a timer FH interrupt is requested.
IENTFH	Enables or disables timer FH interrupt requests.
ϕ_w	Subclock frequency; this is set to 32.768 kHz in this sample task.
TMA3	Sets transition to watch mode.

3. Principle of Operation

1. Operation of the timer F in 16-bit timer mode is described below.
   Timer F is a 16-bit counter which is incremented each time a clock pulse is input. The value of the timer counter F is constantly compared with the value set in the output compare register F; when they match, the operations that the counter is cleared, an interrupt request is issued, and port toggle is output are possible. It can also function as two independent 8-bit timers.
   When CKSH2 bit in timer control register F (TCRF)
   Immediately after a reset, the timer counter F (TCF) is initialized to H'0000, the output compare register F (OCRF) to H'FFFF, and the timer control register F (TCRF) and timer control status register F (TCSRF) are both initialized to H'00. The counter starts incrementing by input from an external event (TMIF). The external event edge selection is set by the IEG3 bit in the IRQ edge select register (IEGR).
   Using the CKSL2 through CKSL0 bits in TCRF, the operating clock for timer F is selected from three kinds of internal clocks output by PSS, an internal clock of $\phi_W/4$, or an external clock.
   The contents of TCF and OCRF are constantly compared; when the both values match, the CMFH bit in TCSRF is set to 1. At this time when the IENTFH bit in IENR2 is 1 an interrupt request is sent to the CPU, and at the same time the toggle signal is output from the TMOFH pin. Also, when the CCLRH bit in TCSRF is 1, TCF is cleared. The output from pin TMOFH can be set by the TOLH bit in TCRF.
   When TCF overflows (H'FFFF → H'0000), the OVFH bit in TCSRF is set to 1. At the time, when both the OVIEH bit in TCSRF and the IENTFH bit in IENR2 are 1, an interrupt request is sent to the CPU.

2. Timer F operating modes are shown in Table 4.

Table 4 Timer F Operating Modes

Operating Mode	Reset	Active	Sleep	Watch	Sub-active	Sub-sleep	Standby	Module standby
TCF	Reset	Functions	Functions	Functions/ Halted	Functions/ Halted*	Functions/ Halted*	Halted	Halted
OCRF	Reset	Functions	Held	Held	Functions	Held	Held	Held
TCRF	Reset	Functions	Held	Held	Functions	Held	Held	Held
TCSRF	Reset	Functions	Held	Held	Functions	Held	Held	Held

Note: * When $\phi_W/4$ is selected as the TCF's internal clock in active mode or sleep mode, the system clock and internal clock are mutually asynchronous, and so synchronization is established by a synchronization circuit. This results in a maximum error of 1/ϕ(s) in the count period.
When the counter is operated in subactive mode, watch mode, or subsleep mode, $\phi_W/4$ must always be selected as the internal clock. The counter will not operate when any other internal clock is selected.

3. Figure 5 shows the operating theory in this sample task.

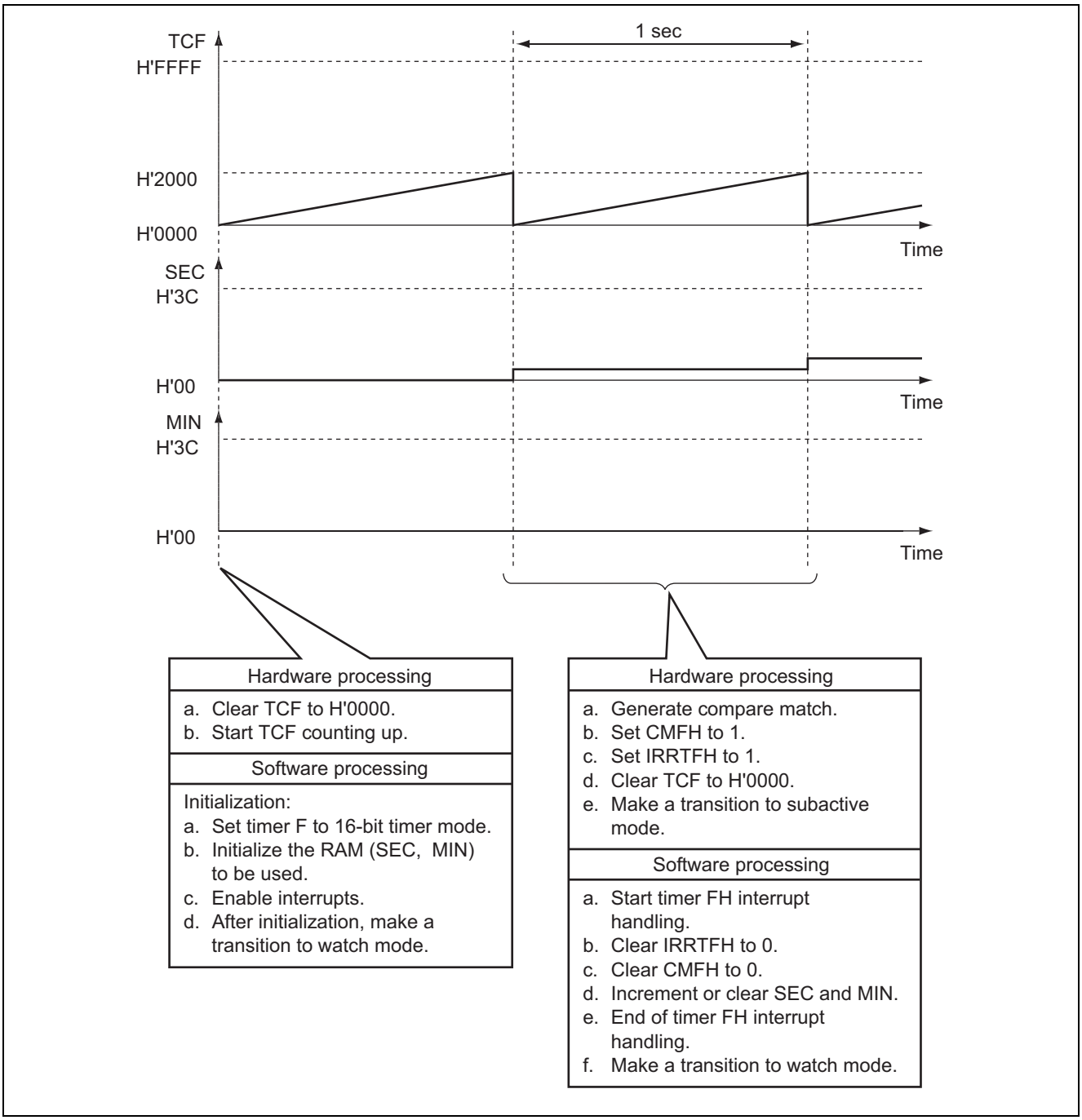


Figure 5 Timer F Operating Theory

4. Description of Software

4.1 Modules

The modules used in this sample task are shown in table 5.

Table 5 Description of Modules

Module	Label	Function
Main routine	main	Initializes RAM and timer F, enables interrupts, and makes a transition to watch mode.
Timer F interrupt handling routine	tfint	Clears IRRTFH and CMFH to 0, save register data, increments or clears SEC and MIN defined in RAM, and restores register data.

4.2 Arguments

Table 6 describes the arguments used in this sample task.

Table 6 Description of Arguments

Arguments	Function	Module Used	Data Length	Input/ Output
SEC	A clock counter for counting seconds	main, tfint	1 byte	Output
MIN	A clock counter for counting minutes	main, tfint	1 byte	Output

4.3 Internal Registers

Table 7 shows the internal registers used in this sample task.

Table 7 Description of Internal Registers

Register		Function	Address	Setting
TCRF	CKSH2 CKSH1 CKSH0	Timer Control Register F (Clock Select H2 to 0) Select the clock input to TCFH from among four internal clock sources or a TCFL overflow. When CKSH2 = 0, CKSH1 = 0 and CKSH0 = 0, TCFL overflow is selected When CKSH2 = 0, CKSH1 = 0 and CKSH0 = 1, TCFL overflow is selected When CKSH2 = 0, CKSH1 = 1 and CKSH0 = 0, TCFL overflow is selected When CKSH2 = 0, CKSH1 = 1 and CKSH0 = 1, this setting is prohibited When CKSH2 = 1, CKSH1 = 0 and CKSH0 = 0, internal clock φ/32 is selected When CKSH2 = 1, CKSH1 = 0 and CKSH0 = 1, internal clock φ/16 is selected When CKSH2 = 1, CKSH1 = 1 and CKSH0 = 0, internal clock φ/4 is selected When CKSH2 = 1, CKSH1 = 1 and CKSH0 = 1, internal clock φw/4 is selected	H'FFB6 Bit 6 Bit 5 Bit 4	CKSH2 = 0 CKSH1 = 0 CKSH0 = 0
	CKSL2 CKSL1 CKSL0	Timer Control Register F (Clock Select L2 to 0) Select the clock input to TCFL from among four internal clock sources or an external event. When CKSL2 = 0, CKSL1 = 0 and CKSL0 = 0, an external event is selected When CKSL2 = 0, CKSL1 = 0 and CKSL0 = 1, an external event is selected When CKSL2 = 0, CKSL1 = 1 and CKSL0 = 0, an external event is selected When CKSL2 = 0, CKSL1 = 1 and CKSL0 = 1, this setting is prohibited When CKSL2 = 1, CKSL1 = 0 and CKSL0 = 0, internal clock φ/32 is selected When CKSL2 = 1, CKSL1 = 0 and CKSL0 = 1, internal clock φ/16 is selected When CKSL2 = 1, CKSL1 = 1 and CKSL0 = 0, internal clock φ/4 is selected When CKSL2 = 1, CKSL1 = 1 and CKSL0 = 1, internal clock φw/4 is selected	H'FFB6 Bit 2 Bit 1 Bit 0	CKSL2 = 1 CKSL1 = 1 CKSL0 = 1

Register		Function	Address	Setting
TCSRF	OVFH	Timer Control/Status Register F (Timer Overflow Flag H) A status flag indicating whether or not TCF has overflowed. When OVFH = 0, indicates no overflow of TCF When OVFH = 1, indicates TCF overflow	H'FFB7 Bit 7	0
	CMFH	Timer Control/Status Register F (Compare Match Flag H) A status flag indicating whether or not TCF has compare matched OCRF. When CMFH = 0, indicates no compare match between TCF and OCRF When CMFH = 1, indicates TCF has compare matched OCRF	H'FFB7 Bit 6	0
	OVIEH	Timer Control/Status Register F (Timer Overflow Interrupt Enable H) Enables or disables interrupt requests when TCF overflows. When OVIEH = 0, disables TCF overflow interrupt requests. When OVIEH = 1, enables TCF overflow interrupt requests.	H'FFB7 Bit 5	1
	CCLRH	Timer Control/Status Register F (Counter Clear H) Selects whether or not TCF is cleared when TCF has compare matched OCRF. When CCLRH = 0, disables TCF clear by compare match When CCLRH = 1, enables TCF clear by compare match	H'FFB7 Bit 4	1
TCFH		Timer Counter FH Upper 8 bits of 16-bit timer counter F (TCF); functions as an 8-bit up-counter using a TCFL overflow signal as an input clock.	H'FFB8	H'00
TCFL		Timer Counter FL Lower 8 bits of 16-bit timer counter F (TCF); functions as an 8-bit up-counter using ϕw/4 of internal clock as an input clock.	H'FFB9	H'00
OCRFH		Output Compare Register FH Upper 8 bits of 16-bit output compare register (OCRF); generates a compare match signal when OCRF has matched TCF.	H'FFBA	H'20
OCRFL		Output Compare Register FL Lower 8 bits of 16-bit output compare register (OCRF); generates a compare match signal when OCRF has matched TCF.	H'FFBB	H'00
IENR2	IENTFH	Interrupt Enable Register 2 (Timer FH Interrupt Enable) Enables or disables timer FH interrupt requests. When IENTFH = 0, disables timer FH interrupt requests When IENTFH = 1, enables timer FH interrupt requests	H'FFF4 Bit 3	1
IRR2	IRRTFH	Interrupt Request Register 2 (Timer FH Interrupt Request Flag) Indicates whether or not there has been a timer FH interrupt request. When IRRTFH = 0, indicates that no timer FH interrupt has been requested When IRRTFH = 1, indicates that a timer FH interrupt has been requested	H'FFF7 Bit 3	0

Register		Function	Address	Setting
SYSCR1	SSBY	System Control Register 1 (Software Standby) Carries out transitions to standby mode or watch mode. When SSBY = 0, after executing a SLEEP instruction in active mode, a transition is made to sleep mode, or after executing a SLEEP instruction in subactive mode, a transition is made to subsleep mode. When SSBY = 1, after executing a SLEEP instruction in active mode, a transition is made to standby mode or to watch mode, or after executing a SLEEP instruction in subactive mode, a transition is made to watch mode.	H”FFF0 Bit 7	1
	STS2 STS1 STS0	System control register 1 (Standby Timer Select 2 to 0) Specify the time for the CPU and peripheral functions to wait until the clock stabilizes when standby mode or watch mode is cleared and a transition is made to active mode due to a specific interrupt. Note that the standby time must be specified to be equal to or longer than the oscillation stabilization time according to the operating frequency. When STS2 to STS1 = 000, standby time is 8,192 states When STS2 to STS1 = 001, standby time is 16,384 states When STS2 to STS1 = 010, standby time is 1,024 states When STS2 to STS1 = 011, standby time is 2,048 states When STS2 to STS1 = 100, standby time is 4,096 states When STS2 to STS1 = 101, standby time is 2 states When STS2 to STS1 = 110, standby time is 8 states When STS2 to STS1 = 111, standby time is 16 states	H'FFF0 Bit 6 Bit 5 Bit 4	STS2 = 0 STS1 = 0 STS0 = 0
	LSON	System Control Register 1 (Low Speed On Flag) When watch mode is cleared, selects either the system clock or the subclock as the CPU operating clock. When LSON = 0, selects the system clock as the CPU operating clock When LSON = 1, selects the subclock as the CPU operating clock	H'FFF0 Bit 3	1
SYSCR2	DTON	System Control Register 2 (Direct Transfer On Flag) Specifies whether or not to make direct transitions among active (high-speed) mode, active (medium-speed) mode, and subactive mode when a SLEEP instruction is executed. When DTON = 0 and a SLEEP instruction is executed in active mode, a transition to standby mode, watch mode or sleep mode occurs When DTON = 1 and a SLEEP instruction is executed in active (high-speed) mode, a direct transition occurs to active (medium-speed) mode (when SSBY = 0, MSON = 1, LSON = 0) or to subactive mode (when SSBY = 1, TMA3 = 1, LSON = 1)	H'FFF1 Bit 3	0
	MSON	System Control Register 2 (Medium Speed On Flag) Selects whether to operate in active (high-speed) mode or in active (medium-speed) mode after clearing standby mode, watch mode, or sleep mode. When MSON = 0, operates in active (high-speed) mode When MSON = 1, operates in active (medium- speed) mode	H'FFF1 Bit 2	0

Register		Function	Address	Setting
TMA	TMA3	Timer Mode Register A (Timer Mode Register A3) Selects the clock source input to TCA. When TMA3 = 0, a change to standby mode can be made. When TMA3 = 1, a change to active (high-speed) mode, active (medium-speed) mode, subactive mode, watch mode, or subsleep mode can be made.	H'FFB0 Bit 3	1

5. RAM

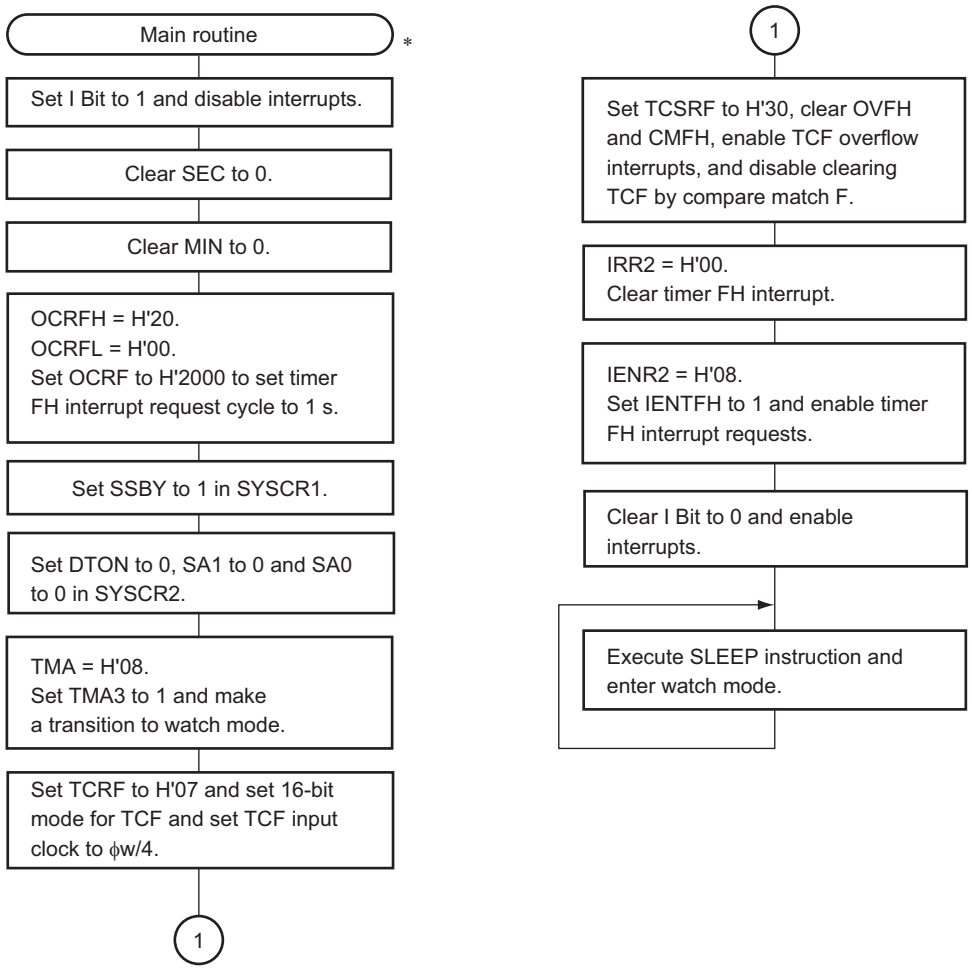
Table 8 describes the RAMs used in this sample task.

Table 8 Description of RAM Used

Label	Function	Address	Used in
SEC	Counter used for a clock, counting the seconds	H'FB80	Main routine Timer F interrupt handling routine
MIN	Counter used for a clock, counting the minutes	H'FB81	Main routine Timer F interrupt handling routine

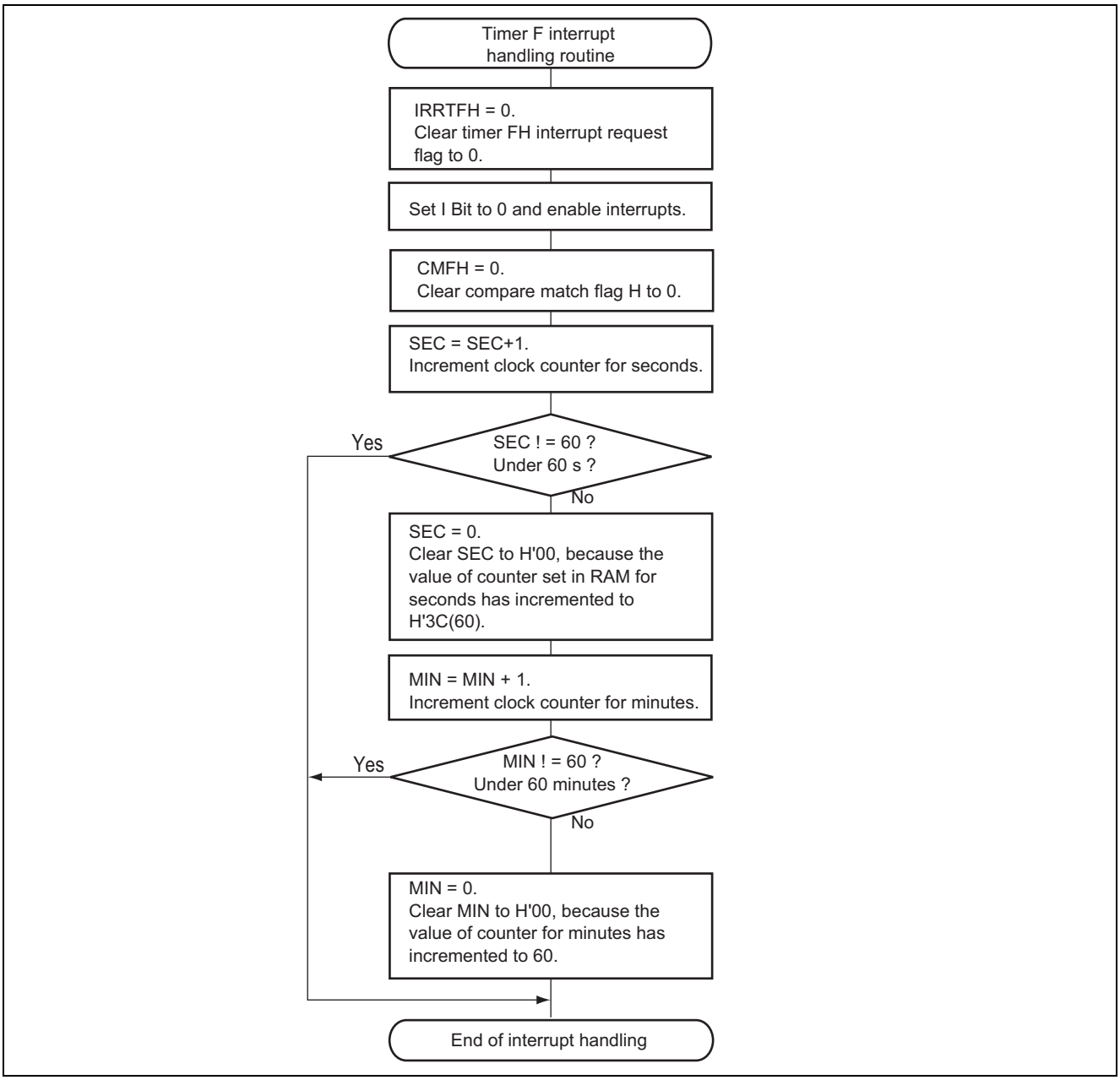
6. Flowchart

1. Main routine



Note: * In this task example, the stack pointer is set in INIT.SRC (assembly).

2. Timer F interrupt handling routine



7. Program Listing

INIT.SRC (Program listing)

```
    .EXPORT    _INIT
    .IMPORT    _main
;
    .SECTION   P,CODE
_INIT:
    MOV.W      #H'FF80,R7
    LDC.B      #B'10000000,CCR
    JMP        @_main
;
    .END
```

```
/************************************************************************************************/
/*                                                                                              */
/*  H8/300L Super Low Power Series                                                              */
/*      -H8/38024 Series-                                                                       */
/*  Application Note                                                                            */
/*                                                                                              */
/*  'Timer F -Clock Time Base-'                                                                 */
/*                                                                                              */
/*  Function                                                                                    */
/*  : Timer F                                                                                   */
/*                                                                                              */
/*  External Clock :  10MHz                                                                     */
/*  Internal Clock :  5MHz                                                                      */
/*  Sub Clock :       32.768kHz                                                                 */
/*                                                                                              */
/************************************************************************************************/

#include     <machine.h>

/************************************************************************************************/
/*  Symbol Definition                                                                           */
/************************************************************************************************/
struct BIT {
    unsigned char     b7:1;       /* bit7 */
    unsigned char     b6:1;       /* bit6 */
    unsigned char     b5:1;       /* bit5 */
    unsigned char     b4:1;       /* bit4 */
    unsigned char     b3:1;       /* bit3 */
    unsigned char     b2:1;       /* bit2 */
    unsigned char     b1:1;       /* bit1 */
    unsigned char     b0:1;       /* bit0 */
};
```

```
#define   TMA         *(volatile unsigned char *)0xFFB0  /* Timer Mode Register A              */
#define   TCA         *(volatile unsigned char *)0xFFB1  /* Timer Counter A                    */
#define   TCRF        *(volatile unsigned char *)0xFFB6  /* Timer Control Register F           */
#define   TCRF_BIT    (*(struct BIT *)0xFFB6)            /* Timer Control Register F           */
#define   TOLH        TCRF_BIT.b7                        /* Toggle Output Level F              */
#define   CKSH2       TCRF_BIT.b6                        /* Clock Select H2                    */
#define   CKSH1       TCRF_BIT.b5                        /* Clock Select H1                    */
#define   CKSH0       TCRF_BIT.b4                        /* Clock Select H0                    */
#define   TCSRF       *(volatile unsigned char *)0xFFB7  /* Timer Control Status Register F    */
#define   TCSRF_BIT   (*(struct BIT *)0xFFB7)            /* Timer Control Status Register F    */
#define   OVFH        TCSRF_BIT.b7                       /* Timer Overflow Flag H              */
#define   CMFH        TCSRF_BIT.b6                       /* Compare Match Flag H               */
#define   OVIEH       TCSRF_BIT.b5                       /* Timer Overflow Interrupt Enable    */
#define   CCLRH       TCSRF_BIT.b4                       /* Output Select 3                    */
#define   OCRFH       *(volatile unsigned char *)0xFFBA  /* Output Compare Register FH         */
#define   OCRFL       *(volatile unsigned char *)0xFFBB  /* Output Compare Register FH         */
#define   TMG         *(volatile unsigned char *)0xFFBC  /* Output Compare Register FH         */
#define   SYSCR1      *(volatile unsigned char *)0xFFF0  /* System Control Register 1          */
#define   SYSCR1_BIT  (*(struct BIT *)0xFFF0)            /* System Control Register 1          */
#define   SSBY        SYSCR1_BIT.b7                      /* Software Standby                   */
#define   STS2        SYSCR1_BIT.b6                      /* Standby Timer Select 2             */
#define   STS1        SYSCR1_BIT.b5                      /* Standby Timer Select 1             */
#define   STS0        SYSCR1_BIT.b4                      /* Standby Timer Select 0             */
#define   LSON        SYSCR1_BIT.b3                      /* Low Speed On Flag                  */
#define   MA1         SYSCR1_BIT.b1                      /* Active Mode Clock Select 1         */
#define   MA0         SYSCR1_BIT.b0                      /* Active Mode Clock Select 0         */
#define   SYSCR2      *(volatile unsigned char *)0xFFF1  /* System Control Register 2          */
#define   SYSCR2_BIT  (*(struct BIT *)0xFFF1)            /* System Control Register 2          */
#define   NESEL       SYSCR2_BIT.b4                      /* Noise Elimination Sampling         */
                                                         /* Frequency Select                   */
#define   DTON        SYSCR2_BIT.b3                      /* Direct Transfer On Flag            */
#define   MSON        SYSCR2_BIT.b2                      /* Middle Speed On Flag               */
#define   SA1         SYSCR2_BIT.b1                      /* Subactive Mode Clock Select 1      */
#define   SA0         SYSCR2_BIT.b0                      /* Subactive Mode Clock Select 0      */
#define   IENR1_BIT   (*(struct BIT *)0xFFF3)            /* Interrupt Enable Register 1        */
#define   IENTA       IENR1_BIT.b7                       /* Timer A Interrupt Enable           */
#define   IENR2       *(volatile unsigned char *)0xFFF4  /* Interrupt Enable Register 2        */
#define   IENR2_BIT   (*(struct BIT *)0xFFF4)            /* Interrupt Enable Register 2        */
#define   IENDT       IENR2_BIT.b7                       /* Direct Transfer Interrupt Enable   */
#define   IENTFH      IENR2_BIT.b3                       /* Timer FH Interrupt Enable          */
#define   IENTG       IENR2_BIT.b4                       /* Timer FH Interrupt Enable          */
#define   IRR1_BIT    (*(struct BIT *)0xFFF6)            /* Interrupt Request Register 1       */
#define   IRRTA       IRR1_BIT.b7                        /* Timer A Interrupt Request Flag     */
#define   IRR2        *(volatile unsigned char *)0xFFF7  /* Interrupt Request Register 2       */
#define   IRR2_BIT    (*(struct BIT *)0xFFF7)            /* Interrupt Request Register 2       */
#define   IRRDT       IRR2_BIT.b7                        /* Direct Transfer Interrupt          */
                                                         /* Request Flag                       */
#define   IRRTFH      IRR2_BIT.b3                        /* Timer FH Interrupt Request Flag    */
#define   IRRTG       IRR2_BIT.b4                        /* Timer FH Interrupt Request Flag    */
#define   CKSTPR1     *(volatile unsigned char *)0xFFFA  /*                                    */
#define   CKSTPR2     *(volatile unsigned char *)0xFFFB  /*                                    */

#pragma interrupt     (tfint)
```

```
/*****************************************************************************************/
/*  Function define                                                                      */
/*****************************************************************************************/
extern void  INIT ( void );                         /* SP Set                            */
void    main ( void );
void    tfint ( void );

/*****************************************************************************************/
/*  RAM define                                                                           */
/*****************************************************************************************/
unsigned char         SEC;
unsigned char         MIN;

/*****************************************************************************************/
/*  Vector Address                                                                       */
/*****************************************************************************************/
#pragma section       V1                            /* Vector Section Set                */
void (*const VEC_TBL1[])(void) = {
    INIT                                            /* 0x0000 Reset Vector               */
};
#pragma section       V2                            /* Vector Section Set                */
void (*const VEC_TBL2[])(void) = {
    tfint                                           /* 0x001E Timer F Interrupt Vector   */
};

#pragma section                                     /* P                                 */
/*****************************************************************************************/
/*  Main Program                                                                         */
/*****************************************************************************************/
void    main ( void )
{
    set_imask_ccr(1);                               /* Interrupt Disable                 */

    SEC = 0;
    MIN = 0;

    OCRFH = 0x20;                                   /* Initialize Compare Match FH Value */
    OCRFL = 0x00;                                   /* Initialize Compare Match FL Value */

    SYSCR1 = 0x8F;                                  /* Set SYSCR1                        */
    SYSCR2 = 0xE0;                                  /* Set SYSCR2                        */

    TMA = 0x08;                                     /* Initialize TCA Overflow Period    */

    TCRF = 0x07;                                    /* TMOFH High level Output           */
    TCSRF = 0x30;                                   /* Initialize Overflow Interrupt     */

    IRR2  = 0x00;
    IENR2 = 0x08;                                   /* Timer A Interrupt Enable          */
    set_imask_ccr(0);                               /* Interrupt Enable                  */

    while(1){
        sleep();                                    /* Transition to Sleep Mode          */
    }
}
```

```
/*****************************************************************************************/
/*  Timer F Interrupt                                                                    */
/*****************************************************************************************/
void tfint ( void )
{
    IRRTFH = 0;                                     /* Clear Timer F Interrupt Request     */
    set_imask_ccr(0);                               /* Interrupt Enable                    */

    CMFH = 0;                                       /* Clear Compare Match Flag H          */
    SEC++;                                          /* Increment Second Counter            */

    if ( SEC = =  60 ){
        SEC = 0;                                    /* Initialize Second Counter           */
        MIN++;                                      /* Increment Minute Counter            */
        if ( MIN = =  60 ){
             MIN = 0;                               /* Initialize Minute Counter           */
        }
    }
}
```

Link address specifications

Section Name	Address
CV1	H'0000
CV2	H'001E
P	H'0100
B	H'FB80

Website and Support

Renesas Technology Website
http://www.renesas.com/

Inquiries
http://www.renesas.com/inquiry
csc@renesas.com

Revision Record

		Description	
Rev.	**Date**	**Page**	**Summary**
1.00	Dec.19.03	—	First edition issued
2.00	Nov.30.06	All pages	Content correction

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, 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 products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (http://www.renesas.com)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
   (1) artificial life support devices or systems
   (2) surgical implantations
   (3) healthcare intervention (e.g., excision, administration of medication, etc.)
   (4) any other purposes that pose a direct threat to human life

   Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. 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 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 applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.