To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

Notice

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

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

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

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

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

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

M16C/62P Group

Application of Oscillation Stop, Re-Oscillation Detection Function

1. Abstract

This application note describes application examples of oscillation stop and re-oscillation detection function in the M16C/62P group. The sample program explains an example that the on-chip oscillator continues operating a program even when the main clock stops.

2. Introduction

Application examples described in this document are applied to the following MCU and conditions:

MCU	M16C/62P Group
Evaluation/Operation Environment	StarterKit for M16C/62P (M3A-0664)
Main Clock	6MHz
CPU Clock	PLL clock 24MHz (Quadruple of main clock)

This program can be used for the other M16C Families which have the same SFR (Special Function Register) as the one in the M16C/62P group. However, since some functions may be modified such as added functions, check it in a manual. Execute sufficient evaluation when using this application note.

3. Function Outline

This section describes the function outline used in this sample program.
Refer to the M16C/62P group hardware manual for details of each function.

Function	Reference for Details M16C/62P Group Hardware Manual
3.1 On-Chip Oscillator Clock	“Clock Generating Circuit”
3.2 Oscillation Stop, Re-Oscillation Detection Function	“Clock Generating Circuit”, “Oscillation Stop, Re-Oscillation Detection Function”

Associated registers set in this sample program are attached in “Appendix A”.
Refer to the updated hardware manual for details.

3.1 On-Chip Oscillator Clock

This clock, approximately 1MHz, is supplied by the on-chip oscillator which the CPU includes. This clock is used as the clock source for the CPU and peripheral function clocks. After reset, the on-chip oscillator stops. (Refer to "Electrical Characteristics" in the hardware manual for the on-chip oscillator specification.)

3.2 Oscillation Stop, Re-Oscillation Detection Function

The oscillation stop, re-oscillation detection function are functions for the main clock to stop by the external factors. The oscillation stop, re-oscillation detection function detects the main clock oscillation circuit stop and re-oscillation. The oscillation stop, re-oscillation detection function can be enabled and disabled by the CM20 bit in the CM2 register. Reset or oscillation stop, re-oscillation detection interrupt is generated when detecting the oscillation stop, re-oscillation. Which is to be generated can be selected by the CM27 bit in the CM2 register.

Table 3.1 lists the Specification of Oscillation Stop, Re-Oscillation Detection Function.

Table 3.1 Specification of Oscillation Stop, Re-Oscillation Detection Function

Item	Specification
Oscillation Stop Detectable Clock and Frequency Bandwidth	f(XIN) ≥ 2 MHz
Valid Condition for Oscillation Stop, Re-Oscillation Detection Function	Set CM20 bit to "1" (enable)
Operation at Oscillation Stop, Re-Oscillation Detection	• When CM27 bit = 0 : Reset occurs • When CM27 bit = 1 : Oscillation stop, re-oscillation detection interrupt generated

4. Sample Program Specification

- Use StarterKit for M16C/62P (M3A-0664)
- Main clock 6MHz
- Normally, operate a program with PLL quadruple (24MHz) of the CPU clock source.
- Enable the oscillation stop, re-oscillation detection interrupt function.
- When detecting the main clock oscillation stop, switch the CPU clock source to the on-chip oscillator.

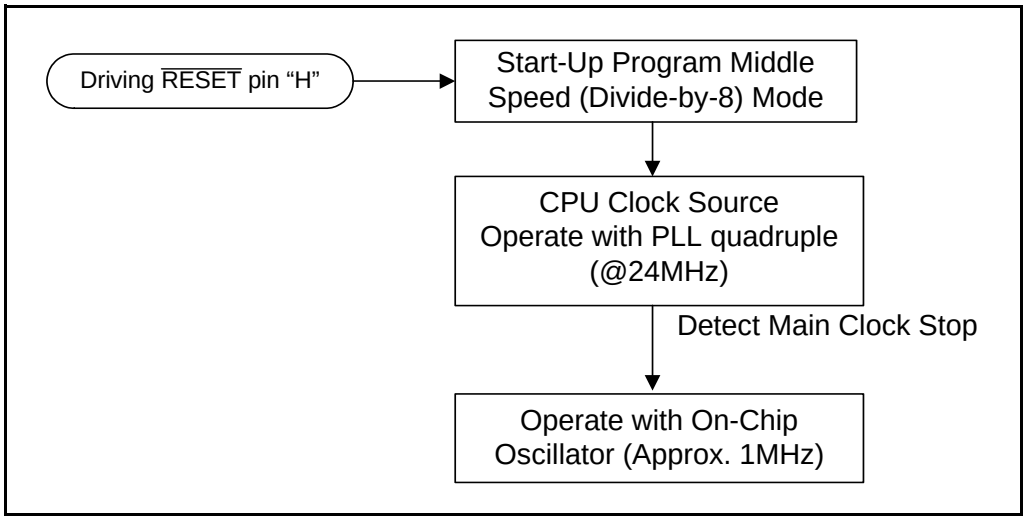


Figure 4.1 Sample Program Operation Outline

- Sample program specification
  LED repeats lighting at 125ms intervals when the CPU clock source operates with PLL quadruple (24MHz). Generate 1ms period by Timer A0 as 24MHz is considered as a CPU clock source and generate 125ms intervals by counting 125 times. The ports of P0_7 to P0_0, P1_0 and P1_1 are used for LED based on the StarterKit. A port connection of the StarterKit for M16C/62P (M3A-0664) is shown below.

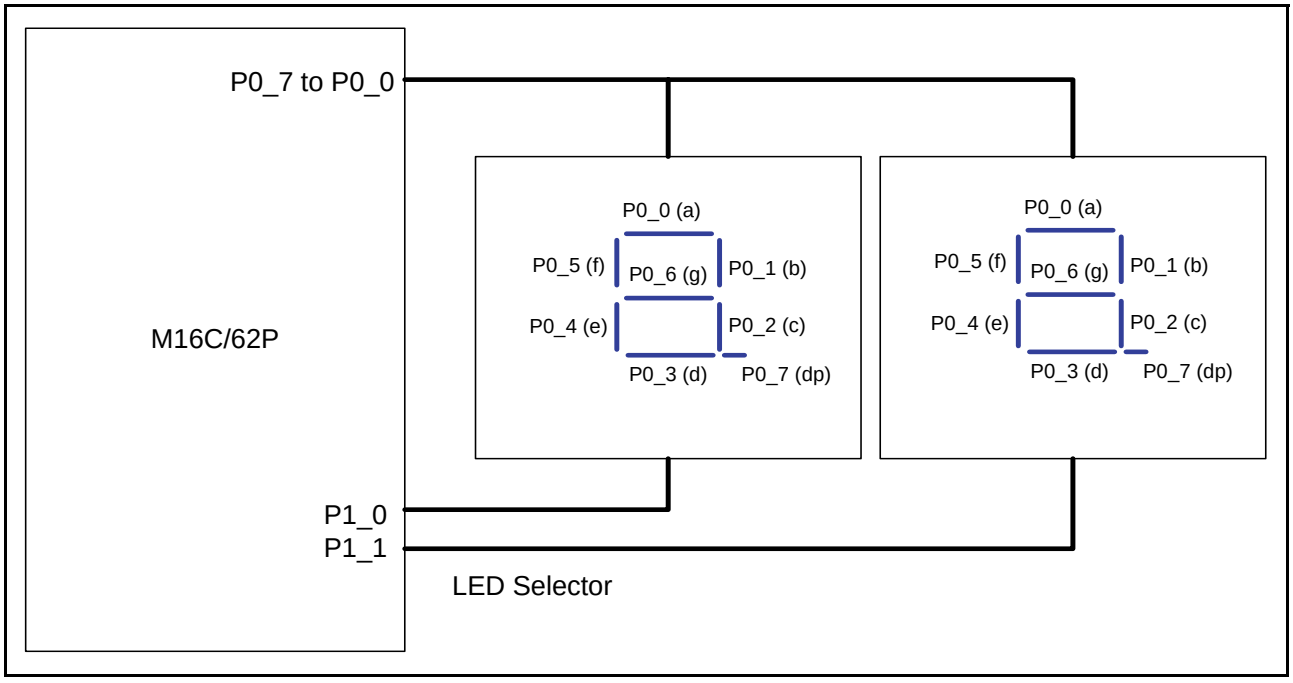
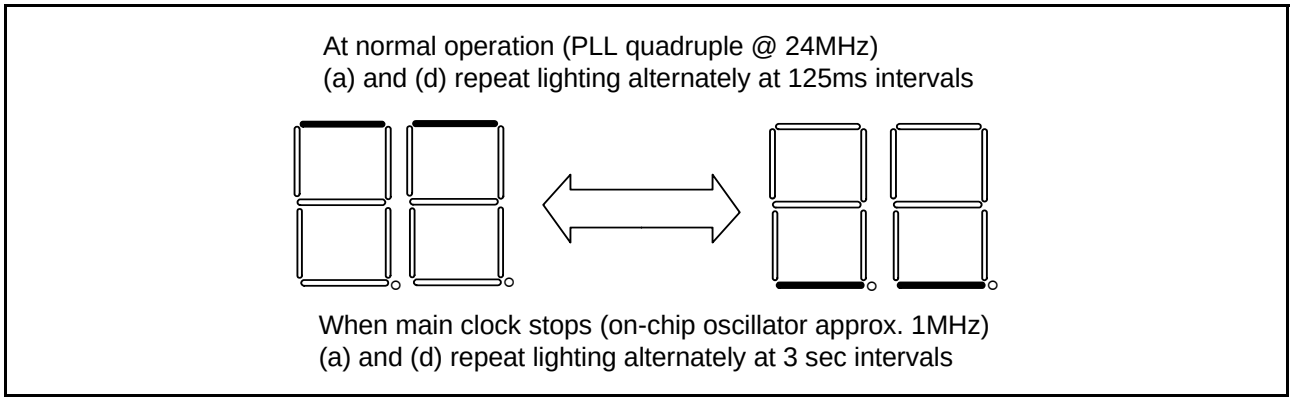


Figure 4.2 Port Connection of StarterKit for M16C/62P (M3A-0664)

- Operation of Sample Program

(1) (a) and (d) of LED repeat lighting alternately for a normal operation as the CPU clock source is considered as PLL24MHz. The lighting intervals of LED are 125ms.

(2) The CPU clock source switches to the on-chip oscillator (approximately 1MHz) when the main clock stops oscillating. The lighting intervals of LED will be 125ms*24=3sec and the program operation continues.



4.1 Register Setting

The following describe the register setting associated with functions used in the sample program.

(1) CM2 (oscillation stop detection register) setting

- Set the oscillation stop detection, re-oscillation detection function to “enabled” by setting the CM20 bit to “1”.
- Set the oscillation stop and operation at re-oscillation detection to “The re-oscillation detection interrupt is generated” by setting the CM27 bit to “1”.

Oscillation Stop Detection Register (1)

Symbol: CM2

Address: 000Ch

b7	b6	b5	b4	b3	b2	b1	b0
1	X	0	0				1

CM20 Oscillation Stop, Re-Oscillation Detection Enable Bit
1 : Oscillation stops, Re-oscillation detection function enabled

CM21 System Clock Select Bit 2
0 : Main clock or PLL clock
1 : On-Chip Oscillator Clock (On-chip oscillator on)

CM22 Oscillation Stop, Re-Oscillation Detection Flag
0 : Main clock stops, re-oscillation not detected
1 : Main clock stops, re-oscillation detected

CM23 XIN Monitor Flag
0 : Main clock oscillates
1 : Main clock stops

CM27 Operation Select Bit (When an oscillation stop, re-oscillation is detected)
1 : Oscillation stop, re-oscillation detection interrupt

NOTES:
1. Rewrite this register after setting the PRC0 bit in the PRCR register to “1” (write enable).

(2) Set fixed vector table

Set an interrupt process address of the oscillation stop, re-oscillation detection function to the vector address 0FFFF0h to 0FFFF3h of the fixed vector table. Refer to **Interrupt** in the M16C/62P group hardware manual for the fixed vector table and **6.1 sect30.inc (Section Definition File)** and the sample program for detail settings.

(3) Process when the oscillation stop, re-oscillation detection interrupts is generated

The oscillation stop, re-oscillation detection interrupts are generated by detecting the main clock stop or re-oscillation and the CM22 bit is set to “1”. At this time, the oscillation stop, re-oscillation detection interrupt will be disabled. Since the CM21 bit remains unchanged when the PLL clock is selected for the CPU clock source, set to “1” in the interrupt process and the on-chip oscillation is selected for the CPU clock source.

(4) Determine main clock state

Determine whether the main clock oscillates or stops by reading the CM23 bit several times in the oscillation stop and re-oscillation detection interrupt process.

5. Sample Program Flow Chart

5.1 Outline Flow

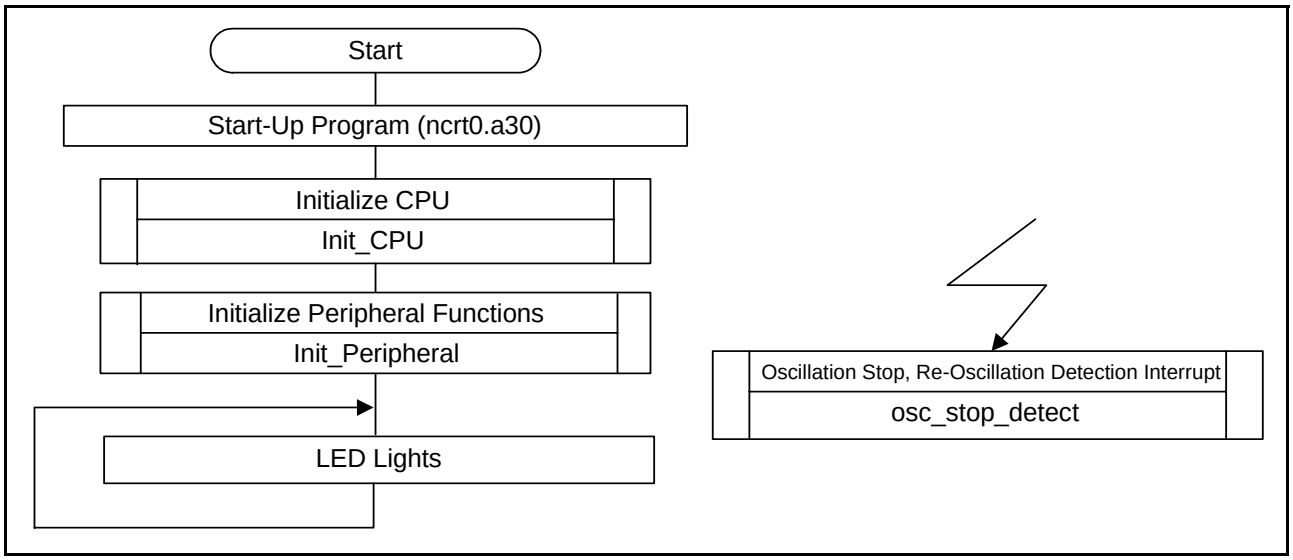


Figure 5.1 Outline Flow Chart of Sample Program

5.2 Initialization Flow of CPU (init_CPU)

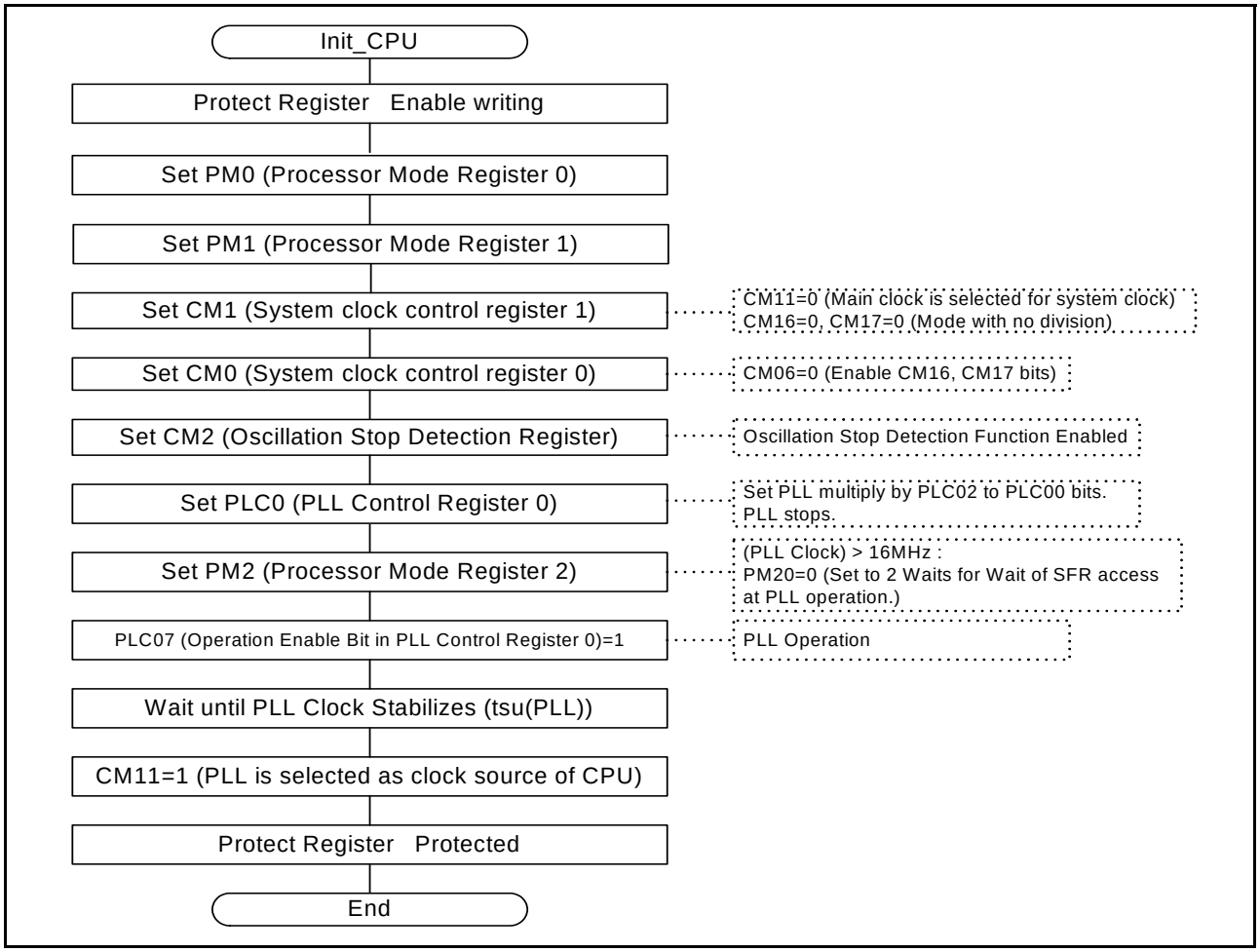


Figure 5.2 Initialization Flow Chart of CPU

5.3 Initialization Flow of Peripheral Functions (init_Peripheral)

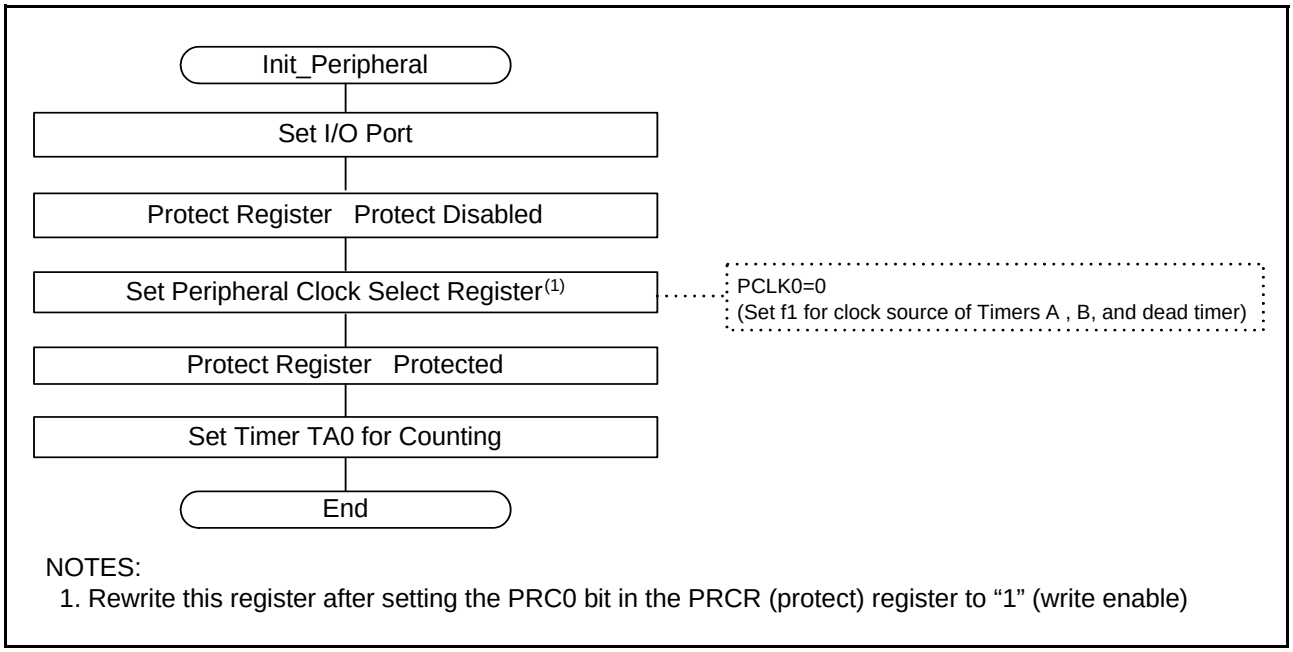


Figure 5.3 Initialization Flow Chart of Peripheral Functions

5.4 Oscillation Stop, Re-Oscillation Detection Interrupt Process Flow (osc_stop_detect)

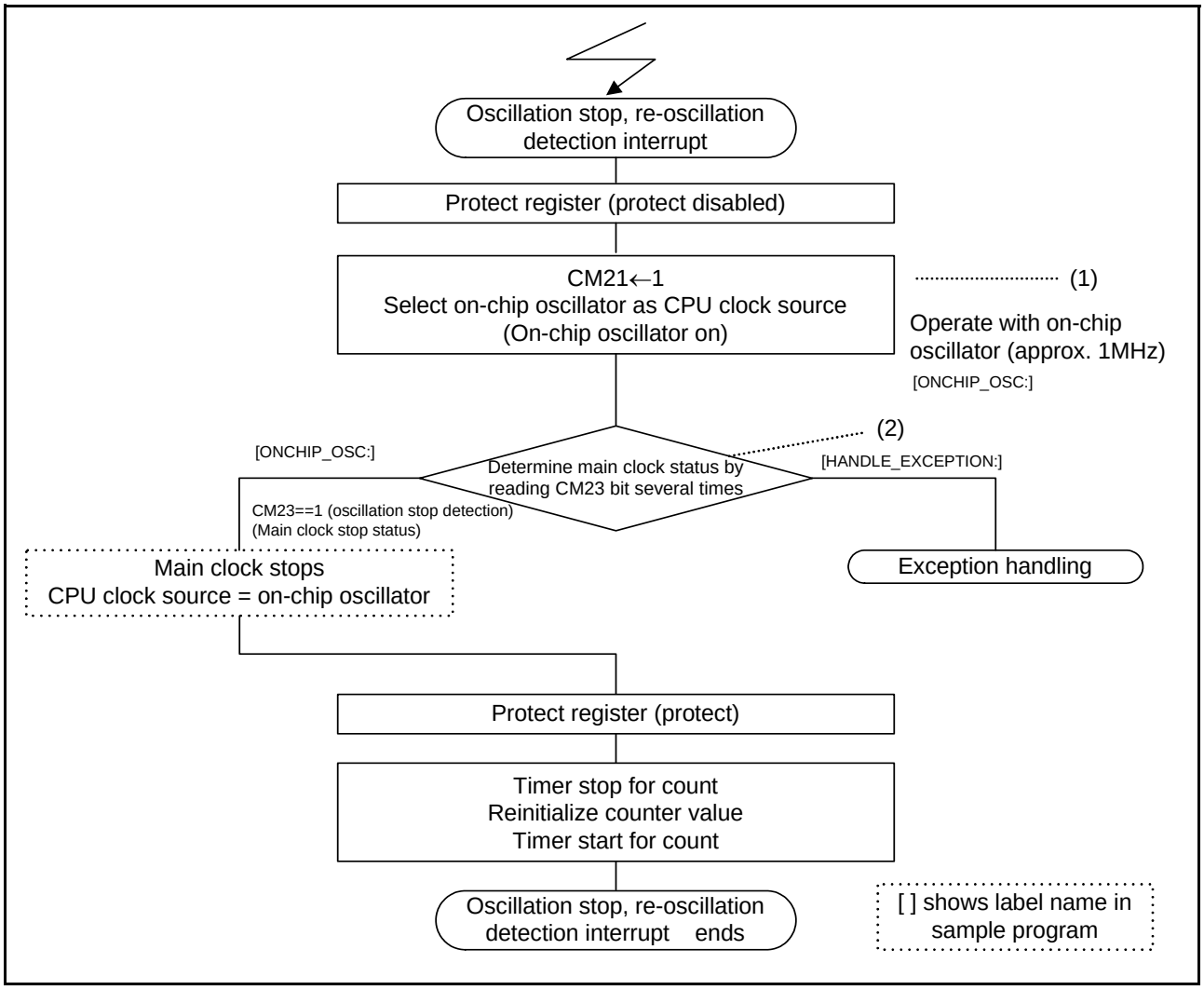


Figure 5.4 Flow Chart of Oscillation Stop, Re-Oscillation Detection Interrupt Process

[Description]

(1) Since the sample program selects the PLL clock as the CPU clock source, the CM21 bit remains unchanged. Therefore, the CM21 bit is set to “1” (on-chip oscillator clock) at the interrupt process beginning.

(2) The main clock status is determined by reading the CM23 bit (XIN monitor flag) several times in the interrupt process. When the CM23 bit value continuously remains stable as “1” several times in sample program, the main clock status is determined as the main clock oscillation stop. When the CM23 bit value continuously does not remain as “1”, the main clock status is determined as an exception. Refer to the next page for an example how to check the CM23 bit.

5.5 Example to Check CM23 Bit (XIN Monitor Flag)

The main clock status is determined by reading the CM23 bit (XIN monitor flag) several times in the sample program. A check flow of the sample program is shown below.

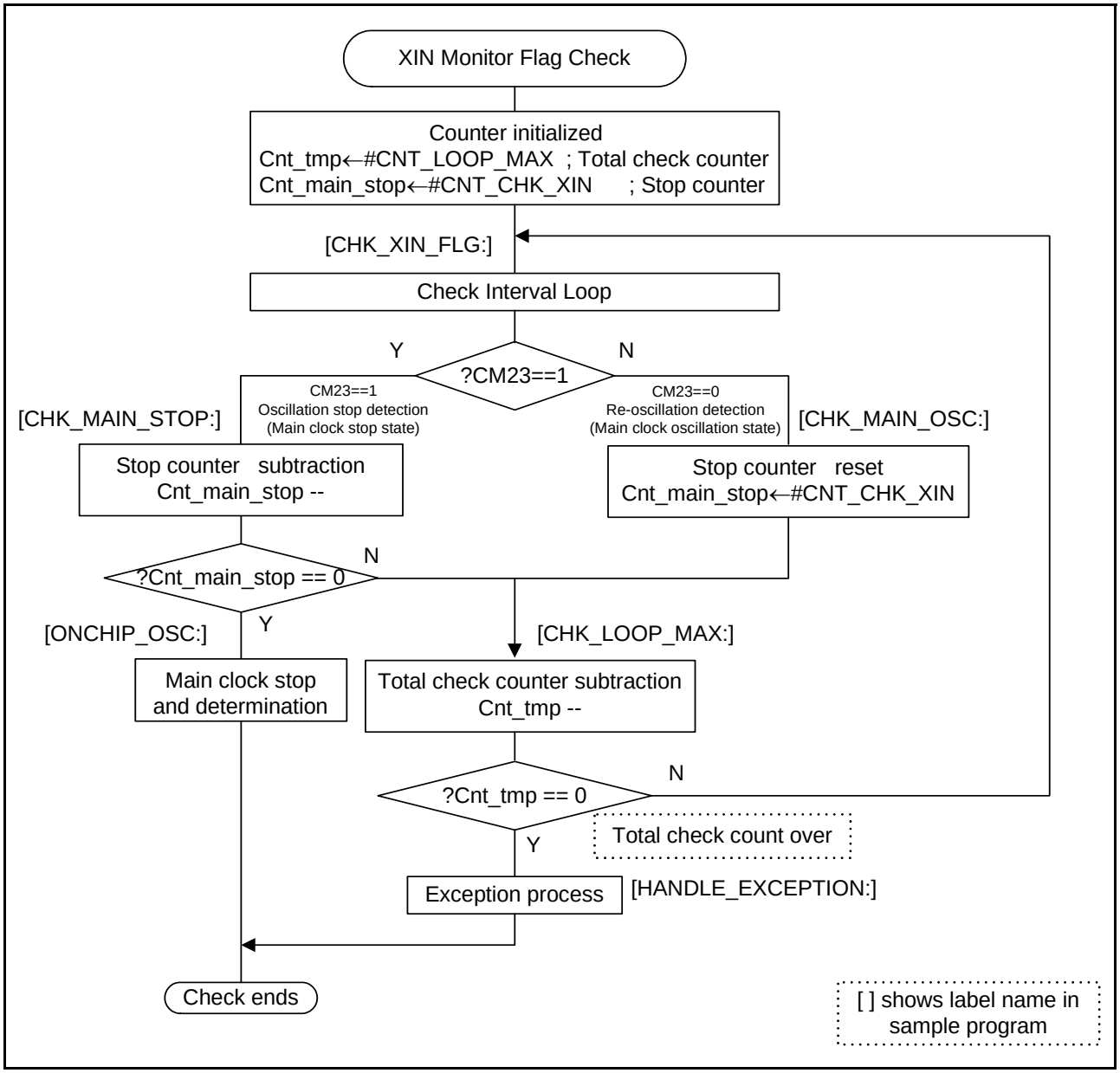


Figure 5.5 Check Flow of CM23 Bit (XIN Monitor Flag)

[Description]
When checking the CM23 bit with some intervals and it continuously remains stable as “1” CNT_CHK_XIN(times) in the sample program, the main clock status is determined as the main clock stop. An example to determine CM23 bit in the sample program is shown below.

5.5.1 Example to Determine CM23 Bit (XIN Monitor Flag)

(1) Check the CM23 bit with a check interval (approx. 5ms).
(Generate a check interval with a loop process. Refer to “6.3 A_interrupt.a30”.)
(2) The CM23 continuously remains “1” CNT_CHK_XIN=8(times), the main clock status is determined as the main clock stop.
(3) When check times of the CM23 bit exceed CNT_LOOP_MAX=16(times), it is determined that an exception occurs.

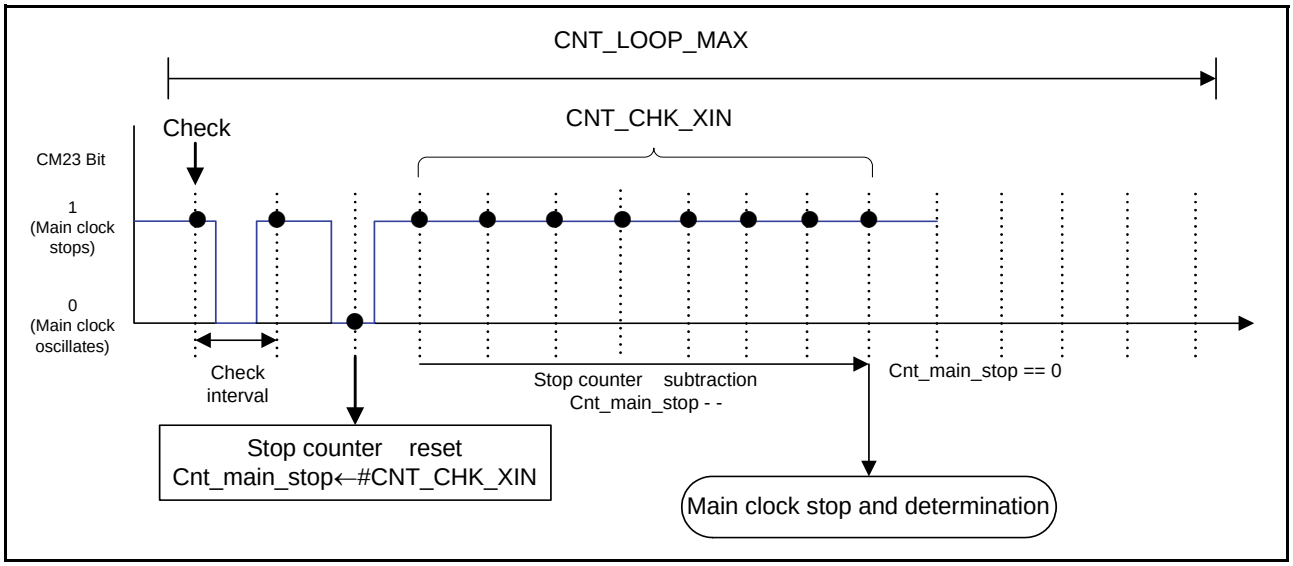


Figure 5.6 Example to Determine CM23 Bit

5.6 HANDLE_EXCEPTION (Exceptional process)

In the sample program, when the CM23 bit (XIN monitor flag) continuously remains stable as “1” (main clock stops) CNT_CHK_XIN(times) in the oscillation stop, re-oscillation detection interrupt process, this case is considered as an exception. As for the sample program exceptional process, the sample program can operate with the on-chip oscillator by jumping the CPU clock source to the process (ONCHIP_OSC) to switch to the on-chip oscillator even if an exception occurs.

6. Sample Program

This sample program is a sample program under the condition shown in 4.Sample Program Specification and it does not guarantee operations in any system/application.

This sample program describes by C language and assembly language. Also, the sample start-up program and section definition file attached to C compiler NC30WA Ver.5.30 Release 1 for the program ROM.

<Use of Compiler>
- NC30WA Ver.5.30 Release 1
- Compile option of sample program (Refer to NC30 users manual for details.)

-c	Make relocatable file (extension.r30), end process
-dSL	Generate an assembly language source file (extension“.a30”) output a C language source list as a comment. Generate an assembly language list file (extension“.lst”)
-fER	Make register storage class available
-g	Output the debug information to an assembly language source file (extension”.a30”). It allows C language level debug.
-finfo	Output necessary information for an inspector, “Stk Viewer”, “Map Viewer”, and “utl30”. This option cannot be specified in the entry version
-Wall	Indicate all detectable warnings
-WEF error output file name	Output an error message to the specified file

The following describe file configurations.

<File Configuration>

File Name	Content	
sfr62p.h, sfr62p.inc	SFR definition file for M16C/62P. Use the most updated file	
ncrt0.a30	Startup program (Same as the file attached to NC30WA Ver.5.30 Release 1)	
sec30.inc(section definition file)	Section definition file (use the file attached to NC30WA Ver.5.30 Release 1) • An interrupt process program is registered into the fixed interrupt vector • The program allocation address is changed because of small program size.	
board.h	Definition file for sample program (StarterKit for M16C/62P M3A-0664)	
C_main.c	Main program	
	Function	Process Outline
	main	Calls main function from startup program Blink process of LED with an endless loop
	init_CPU	Initial setting for MCU Initial setting for MCU operating mode, oscillation stop detection register
	init_Peripheral	Initial setting for peripheral functions Port setting connected with LED Timer A0 setting used for blink indication
A_interrupt.a30	Interrupt process program	
	Function	Process Outline
	osc_stop_detect	Oscillation stop, re-oscillation detection interrupt process When the oscillation stop, re-oscillation detection interrupt request is generated, this process is executed.

6.1 sect30.inc (Section Definition File)

In a section definition file, Modification of Flash ROM area starting address and oscillation stop, re-oscillation detection interrupt process (osc_stop_detect) are registered on the fixed vector table.

```
;***********************************************************************
***
;
; C Compiler for R8C/Tiny, M16C/60,30,20,10
; COPYRIGHT(C) 1999(2000-2002) RENESAS TECHNOLOGY CORPORATION
; AND RENESAS SOLUTIONS CORPORATION ALL RIGHTS RESERVED
;
;
; Written by T.Aoyama
;
; sect30.inc     : section definition
; This program is applicable when using the basic I/O library
;
; $Id: sect30.inc,v 1.22 2004/02/16 05:17:14 muranaka Exp $
;
;        2005.02.25 Ver.1.00
;                 Sample operation of Oscillation Stop and Re-ocsillation
Detect Function.
;                 This file was diverted from the sect30.inc attached to
;                 the NC30WA Ver.5.30 Release 1.
;                 The lines changed from original source are marked by @100
;***********************************************************************
**

.if      __R8C__ != 1
;
;        for M16C/60,30,20,10
;
;------------------------------------------------------------
;
;        Arrangement of section
;
;------------------------------------------------------------
; Near RAM data area
;------------------------------------------------------------
; SBDATA area
         .section data_SE,DATA
         .org     400H
data_SE_top:

         .section bss_SE,DATA,ALIGN
bss_SE_top:

         .section data_SO,DATA
data_SO_top:

         .section bss_SO,DATA
bss_SO_top:

; near RAM area
         .section data_NE,DATA,ALIGN
data_NE_top:

         .section bss_NE,DATA,ALIGN
bss_NE_top:

         .section data_NO,DATA
data_NO_top:

         .section bss_NO,DATA
bss_NO_top:

;------------------------------------------------------------
; Stack area
;------------------------------------------------------------
         .section stack,DATA
         .blkb    STACKSIZE
stack_top:

         .blkb    ISTACKSIZE
istack_top:
```

```
;------------------------------------------------------------
;       heap section
;------------------------------------------------------------
        .section heap,DATA
heap_top:
        .blkb    HEAPSIZE

;------------------------------------------------------------
; Near ROM data area
;------------------------------------------------------------
        .section rom_NE,ROMDATA,ALIGN
rom_NE_top:

        .section rom_NO,ROMDATA
rom_NO_top:

;------------------------------------------------------------
; Far RAM data area
;------------------------------------------------------------
        .section data_FE,DATA
        .org             10000H
data_FE_top:

        .section bss_FE,DATA,ALIGN
bss_FE_top:

        .section data_FO,DATA
data_FO_top:

        .section bss_FO,DATA
bss_FO_top:


;------------------------------------------------------------
; Far ROM data area
;------------------------------------------------------------
        .section rom_FE,ROMDATA
.IF 0
        .org             0F0000H
.ELSE
        .org             0FF000H      ; @100 Use Flash memory area from Block 0
.ENDIF
rom_FE_top:

        .section rom_FO,ROMDATA
rom_FO_top:

  (snip)
  :
  :
```

Since program size is small, modify to use Flash ROM area from Block 0

```
;===============================================================
; fixed vector section
;---------------------------------------------------------------
        .section fvector,ROMDATA
;       .org    0fffdcH
;UDI:
;       .lword  dummy_int
;OVER_FLOW:
;       .lword  dummy_int
;BRKI:
;       .lword  dummy_int
;ADDRESS_MATCH:
;       .lword  dummy_int
;SINGLE_STEP:
;       .lword  dummy_int
;-------------------------------------------------------------------------------
; @100
; Interrupt handling routine of Oscillation Stop and Re-oscillation Detect Function
.glb    osc_stop_detect
        .org    0ffff0H          ; Vector Table Address of
WDT:                             ; WDT, Oscillation Stop and, Re-oscillation, Voltage Down Detection
        .lword  osc_stop_detect       ;
;-------------------------------------------------------------------------------
;DBC:
;       .lword  dummy_int
;NMI:
;       .lword  dummy_int
        .org    0ffffcH
RESET:
        .lword  start

.else   ; __R8C__
   (The rest is omitted)
:
:
```

Oscillation stop, re-oscillation detection interrupt process (ocs_stop_detect) are registered on the fixed vector table

6.2 C_main.c

When operating the sample program, modify the setting value for the following values depending on the compile option and system.

Definition	Content
CNT_WAIT_PLL	Use for the function init_CPU. Define loop endurance to generate PLL oscillation stable waiting time tsu(PLL). Also, using the timers to make waiting time over tsu(PLL) accurately is recommended.

```
/************************************************************************
*
*   FILE     : C_main.c
*   CONTENTS : Sample operation of Oscillation Stop and Re-oscillation Detect Function
*                  Main routine
*
*   CPU      : M16C/62P (Starter Kit for M16C/62P :M3A-0644)
*   OS       : No used
*   COMMENT  :
*   HISTORY  : 2005.02.25 Ver.1.00
*
*   NOTE     :
*
*   Copyright(C)2004, Renesas Technology Corp.
*   Copyright(C)2004, Renesas Solutions Corp.
*   All rights reserved.
*
************************************************************************
*/

/************************************************************************
    Include files
************************************************************************
*/
#include "sfr62p.h"
#include "board.h"

/************************************************************************
    Defines
************************************************************************
*/
/* Set the value suitable for the system */
#define CNT_WAIT_PLL  (14000)       /* Count value of
                                       the PLL Frequency Synthesizer Stabilization Wait Time (@6MHz) */
#define MS_INTERVAL   (125)         /* Count up by 125ms @24MHz PLL */
#define TA0_INTERVAL  (24000-1)     /* 1ms @24MHz PLL, f1 */
#define INIT_LED_data ( ~( _a_ ) )
#define MASK_FLICKER  ( _a_ | _d_ )


/************************************************************************
    Global Variables
************************************************************************
*/
unsigned short Cnt_ms;


/************************************************************************
    Prototypes
************************************************************************
*/
void main( void );
void init_CPU( void );
void init_Peripheral( void );
```

```
/************************************************************************
    MODULE  : main
    FUNCTION: Main routine of sample program
    PARAMETERS: None
    RETURN    : None
************************************************************************
*/
void main( void )
{
    init_CPU();
    init_Peripheral();

    Cnt_ms = 0;
    LED_PORT = INIT_LED_data;

    ta0s = 1; // Start Period count Timer

    while(1){
        // ? TA0's interrupt request is generated
        if( ir_ta0ic == 1 ){
            ta0ic = 0; // Clear Interrupt Request Bit
            if ( MS_INTERVAL <= (Cnt_ms++) ){
                Cnt_ms = 0;
                LED_PORT ^= MASK_FLICKER;
            }
        }
    }
}
```

```
/************************************************************************
    MODULE : init_CPU
    FUNCTION: Initialize CPU
    Processor Mode: Single-Chip Mode
    CPU Clock: PLL Clock (Xin = 6MHz, Multiply by 4 --> 24MHz)
    Oscillation Stop, Re-Oscillation Detection function enabled
    PARAMETERS: None
    RETURN : None
************************************************************************
*/
    void init_CPU( void )
{
    volatile short cnt_word;
    prcr = 0x03; // Protect Register (write enabled)

    //=========================================================================
    // Setting Processor Mode
    //=========================================================================
    // Processor Mode Register 0
    pm0 = 0x00;
        // (b1-b0) Processor Mode Bit [00:Single-Chip]
        // (b2) R/W Mode Select Bit [0:(Note2)]
        // (b3) Software Reset Bit [0]
        // (b5-b4) Multiplexed Bus Space Select Bit [00:(Note2)]
        // (b6) Port P4_0 to P4_3 Function Select Bit [0:(Note2)]
        // (b7) BCLK Output Disable Bit [0:(Note2)]
        // (Note2) Effective when memory expansion mode or microprocessor mode

    // Processor Mode Register 1
    pm1 = 0x08;
        // (b0) CS2 Area Switch Bit [0:Block A disable]
        // (b1) Port P3_7 to P3_4 Function Select Bit [0:(Note3)]
        // (b2) Watchdog Timer Function Select Bit [0:Watchdog timer interrupt]
        // (b3) Internal Reserved Area Expansion Bit [1:The entire area is usable]
        // (b5-b4) Memory Area Expansion Bit [00:(Note3)]
        // (b6) Reserved bit (Set to "0") [0]
        // (b7) Wait Bit [0:No wait state]
        // (Note3) Effective when memory expansion mode or microprocessor mode

    //=========================================================================
    // Using the PLL clock as the clock source for the CPU
    //=========================================================================
    // System Clock Control Register 1
    cm1 = 0x20;
        // (b0) All Clock Stop Control Bit [0:Clock on]
        // (b1) System Clock Select Bit 1 [0:Main clock] (Note5)
        // (b4-b2) Reserved Bit (Set to "0") [000]
        // (b5) XIN-XOUT Drive Capacity Select Bit [1:HIGH]
        // (b7-b6) Main Clock Division Select Bit [00:No division mode]
        // (Note5) After setting the PLC07 bit in the PLC0 register to "1"(PLL operation),
        // wait until Tsu(PLL) elapses before setting the CM11 bit to "1"(PLL Clock)

    // System Clock Control Register 0
    cm0 = 0x00;
        // (b1-b0) Clock Output Function Select Bit [00: I/O port P5_7]
        // (b2) WAIT Mode Peripheral Function Clock Stop Bit
        // [0:Do not stop peripheral function clock in wait mode]
        // (b3) XCIN-XCOUT Drive Capacity Select Bit [0:Low]
        // (b4) Port XC Select Bit [0:I/O port P8_6, P8_7]
        // (b5) Main Clock Stop Bit [0:On]
        // (b6) Main Clock Division Select Bit [0:CM16 and CM17 valid]
        // (b7) System Clock Select Bit [0:Main clock, PLL clock, or on-chip oscillator clock]
```

```
//=========================================================================
//    Enable Detection Stop, Re-oscillation Detection Function
//=========================================================================
// Oscillation Stop Detection Register
cm2 = 0x81;
    // (b0) Oscillation Stop, Re-Oscillation Detection Bit
    // [1:Detection function enabled]
    // (b1) System Clock Select Bit 2 [0:Main clock or PLL clock]
    // (b2) Oscillation Stop, Re-Oscillation Detection Flag [0:(Note4)]
    // (b3) XIN Monitor Flag [0:(Read Only)]
    // (b5-b4) Reserved Bit (Set to "0") [00]
    // (b6) Nothing is assigned (When write, set to "0")[0]
    // (b7) Operation Select Bit [1:Oscillation stop, re-oscillation detection interrupt]
    // (Note4) This flag is set to "1" when the main clock is detected to have stopped and
    //         the main clock is detected to have restarted oscillating.

// PLL Control Register 0 (Set the PLC02 to PLC00 bits (multiplying factor))
plc0 = 0x12;
    // (b2-b0) PLL Muliplying Factor Select Bit [010:Multiply by 4]
    // (b3) Nothing is assigned (When write, set to "0")
    // [0]
    // (b4) Reserved Bit (Set to "1") [1]
    // (b6-b5) Reserved Bit (Set to "0") [00]
    // (b7) Operation Enable Bit [0:PLL Off]

// Processor Mode Register 2 (Set the PM20 bit to "0" (2 wait states))
pm2 = 0x00;
    // (b0) Specifying Wait when Accessing SFR at PLL Operation
    // [0:2 waits (Note2)]
    // (b1) System Clock Protective Bit [0:Clock is protected by PRCR register]
    // (b2) WDT Count Source Protective Bit [0:CPU clock is used for the WDT count source]
    // (b4-b3) Reserved Bit (Set to "0") [00]
    // (b7-b5) Nothing is assigned (When write, set to "0")
    // [000]
    // (Note2) The PM20 bit become effective when PLC07 bit in the PLC0 register is
    //         set to "1"(PLL On).
    //         Change the PM20 bit when the PLC07 bit is set to "0"(PLL Off).
    //         Set the PM20 bit to "0"(2 waits) when PLL clock > 16MHz.

// Set the PLC07 bit to "1" (PLL On)
plc07 = 1;
    // (b7) Operation Enable Bit [1:PLL On]

// Wait until the PLL clock becomes stable (tsu(PLL))
cnt_word = CNT_WAIT_PLL; while( cnt_word -- );

// Set the CM11 bit to "1" (PLL clock for the CPU clock soruce)
cm11 = 1;
    // (b1) System Clock Select Bit 1 [1:PLL clock]

prcr = 0x00;    // Protect Register (write protected)
}
```

```
/***********************************************************************
    MODULE : init_Peripheral
    FUNCTION: Initialize Peripheral
    PARAMETERS: None
    RETURN : None
***********************************************************************
*/
void init_Peripheral( void )
{
    //=========================================================================
    // Setting I/O port
    //=========================================================================
    LED_PORT = LED_data_blank; // Initialize LED Port
    ASSERT_LEDS; // Assert tens and first digit LED
    LED_PD = 0xff; // LED Port Direction
    LED_SELECTOR_PD |= 0x03; // LED Selector Port Direction

    //=========================================================================
    // Peripheral Clock Select Register
    //=========================================================================
    prcr = 0x01; // Protect Register (write enabled)
    // Peripheral Clock Select Register
    pclkr = 0x03;
        // (b0) Timers A, B Clock Select Bit [1:f1]
        // (b1) SI/O Clock Select Bit [1:f1SIO]
        // (b7-b2) Reserved bit (Set to "0") [000000]
    prcr = 0x00; // Protect Register (write protected)

    //=========================================================================
    // Timer A0 (Timer Mode, 1ms)
    //=========================================================================
    // Stop counting
    ta0s = 0;
    // Timer Mode Register
    ta0mr = 0x00;
        // (b1-b0) Operation Mode Select Bit [00:Timer mode]
        // (b2) Pulse Output Function Select Bit [0:Pulse is not output]
        // (b4-b3) Gate Function Select Bit [00:Gate function not available]
        // (b5) Set to "0" in timer mode [0]
        // (b7-b6) Count Source Select Bit [00:f1 or f2]
    // Interrupt Priority Level (Interrupt disable)
    ta0ic=0;
    // Timer Register
    ta0 = TA0_INTERVAL;
}
```

6.3 A_interrupt.a30

The oscillation stop, re-oscillation detection interrupt process of the sample program are shown below.
Set the following values when operating the sample program.
Change the oscillation stop detection logic depending on the system or oscillator.

Definition	Content
CNT_CHK_XIN	Count times of CM23 bit (XIN monitor flag) When the CM23 bit continuously remains stable as “1” CNT_CHK_XIN(times), it is determined as the main clock stops.
CNT_LOOP_MAX	Total check times Limits of CM23 bit check times
CNT_WAIT_CHK	Check interval (Check interval is generated by the loop process CNT_WAIT_CHK(times).)

```
;***********************************************************************
;
;   FILE     : A_interrupt.a30
;   CONTENTS : Sample operation of Oscillation Stop and Re-oscillation Detect Function
;              Interrupt handling routine
;
;   CPU      : M16C/62P (Starter Kit for M16C/62P :M3A-0644)
;   OS       : No used
;   COMMENT  :
;   HISTORY  : 2005.02.25 Ver.1.00
;
;   NOTE     :
;
;   Copyright(C)2004, Renesas Technology Corp.
;   Copyright(C)2004, Renesas Solutions Corp.
;   All rights reserved.
;
;***********************************************************************
;***********************************************************************
; Include files
;***********************************************************************
.list OFF
    .include sfr62p.inc
.list ON

;***********************************************************************
; Defines
;***********************************************************************
; Set the values suitable for the system
CNT_CHK_XIN  .equ (8)       ; Number of Oscillation stop counting
CNT_LOOP_MAX .equ (16)      ; Number of total check
CNT_WAIT_CHK .equ (500)     ; Number of loop that make check interval (Approx. 5ms @1MHz)

TA0_INTERVAL .equ (24000-1) ; 1ms @24MHz PLL, f1

;***********************************************************************
; Global symbols
;***********************************************************************
.glb osc_stop_detect
.glb _Cnt_ms

;***********************************************************************
; Variables
;***********************************************************************
  .section bss_NE, DATA
Cnt_main_stop: .blkw 1
Cnt_tmp: .blkw 1
Cnt_wait: .blkw 1
```

```
    .section program, align
;*****************************************************************************
;   MODULE  : osc_stop_detect
;   FUNCTION: Interrupt handling routine of
;             Oscillation Stop and Re-oscillation Detect Function.
;
;   NOTE    : This routine does not use any registers.
;             Except for automatically saved registers, if there are any registers
;             that are likely to be modified in the interrupt handling routine,
;             save them to the stack in software.
;
;   PARAMETERS: None
;   RETURN : None
;*****************************************************************************
osc_stop_detect:
    mov.b #01h, prcr     ; Protect Register (write enabled)
    bset cm21            ; On-chip oscillator oscillating
                         ; (Where the PLL clock corresponds to the CPU clock source,
                         ; the CM21 bit remains unchanged)
    ;=============================================================================
    ; Check XIN Monitor Flag
    ;=============================================================================
    mov.w #CNT_LOOP_MAX, Cnt_tmp
    mov.w #CNT_CHK_XIN, Cnt_main_stop

CHK_XIN_FLG:
    mov.w #CNT_WAIT_CHK, Cnt_wait
WAIT_CHK:
    adjnz.w #-1, Cnt_wait, WAIT_CHK
    ; Check CM23 (XIN Monitor Flag)
    btst cm23
    jz CHK_MAIN_OSC

CHK_MAIN_STOP:
    adjnz.w #-1, Cnt_main_stop, CHK_LOOP_MAX ; Counting main clock stop
    jmp ONCHIP_OSC

CHK_MAIN_OSC:
    mov.w #CNT_CHK_XIN, Cnt_main_stop        ; Reset main clock stop counter

CHK_LOOP_MAX:
    adjnz.w #-1, Cnt_tmp, CHK_XIN_FLG
    jmp HANDLE_EXCEPTION

    ;=============================================================================
    ; Switched to the On-chip Oscillator Clock
    ;=============================================================================
ONCHIP_OSC:
END_OSC_STOP_DETECT:
    mov.b #00h, prcr         ; Protect Register (write protected)

    mov.w #00h, _Cnt_ms      ; Clear Control Variables
    bclr ta0s                ; Stop counting
    mov.b #00h, ta0ic        ; Clear Timer's Interrupt Request Flag
    mov.w #TA0_INTERVAL, ta0 ; Reset Timer Register
    bset ta0s ; Start Count Timer
    reit

    ;=============================================================================
    ; Handle Exception
    ;=============================================================================
HANDLE_EXCEPTION:
    ; In this sample, If an exception occurred, processing jumps to ONCHIP_OSC.
    jmp ONCHIP_OSC

.END
```

7. Reference Document

Hardware Manual
M16C/62P Group Hardware Manual Rev.2.30
(Use the latest version on the Renesas Technology Corporation Semiconductor Home Page)

Technical Update/Technical News
(Use the latest version on the Renesas Technology Corporation Semiconductor Home Page)

8. Home Page and E-mail Support

Renesas Technology Corporation Semiconductor Home Page
http://www.renesas.com

M16C Family MCU Technical E-mail Support
E-mail: csc@renesas.com

Appendix 1.Registers Associated with Oscillation Stop, Re-Oscillation Detection Function

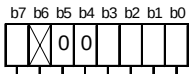
This sample program is referred to the following hardware manual:

- Referred hardware manual : M16C/62P Group hardware manual Rev.2.30

Refer to the latest hardware manual for specifications or functions of each register.

A.1 Oscillation Stop Detection Register

Oscillation Stop Detection Register (1)



Symbol	Address	After Reset
CM2	000Ch	0X000000b (11)

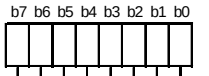
Bit Symbol	Bit Name	Function	RW
CM20	Oscillation Stop, Re-Oscillation Detection Enable Bit (7, 9, 10,11)	0: Oscillation stop, re-oscillation detection function disabled 1: Oscillation stop, re-oscillation detection function enabled	RW
CM21	System Clock Select Bit 2 (2, 3, 6, 8, 11, 12)	0: Main clock or PLL clock 1: On-chip oscillator clock (On-chip oscillator oscillates)	RW
CM22	Oscillation Stop, Re-Oscillation Detection Flag (4)	0: Main clock stops, re-oscillation not detected 1: Main clock stops, re-oscillation detected	RW
CM23	XIN Monitor Flag (5)	0: Main clock oscillates 1: Main clock stops	RO
— (b5-b4)	Reserved Bit	Set to “0”	RW
— (b6)	Nothing is assigned. When write, set to “0”. When read, its content is “0”.		—
CM27	Operation Select Bit (when an oscillation stop, re-oscillation is detected)(11)	0: Oscillation stop detection reset 1: Oscillation stop, re-oscillation detection interrupt	RW

NOTES :

1. Rewrite this register after setting the PRC0 bit in the PRCR register to “1” (write enable).
2. When the CM20 bit is set to “1” (oscillation stop, re-oscillation detection function enabled), the CM27 bit is set to “1” (oscillation stop, re-oscillation detection interrupt), and the CPU clock source is the main clock, the CM21 bit is set to “1” (on-chip oscillator clock) if the main clock stop is detected.
3. If the CM20 bit is set to “1” and the CM23 bit is set to “1” (main clock stops), do not set the CM21 bit to “0”.
4. This bit is set to “1” when the main clock stop is detected and the main clock re-oscillation is detected. When this flag changes state from “0” to “1”, an oscillation stop or a re-oscillation detection interrupt is generated. Use this bit in an interrupt routine to determine the factors of interrupts between the oscillation stop and re-oscillation detection interrupt and the watchdog timer interrupt. This bit is set to “0” by writing “0” in a program. (This bit remains unchanged even if writing “1”. Nor is it set to “0” when an oscillation stop or a re-oscillation detection interrupt request is acknowledged.)
   When the CM22 bit is set to “1” and an oscillation stop or a re-oscillation is detected, an oscillation stop or a re-oscillation detection interrupt is not generated.
5. Determine the main clock status by reading the CM23 bit several times in an oscillation stop or a re-oscillation detection interrupt routine
6. This bit is valid when the CM07 bit in the CM0 register is set to “0”.
7. When the PM21 bit in the PM2 register is set to “1” (disable clock modification), this bit remains unchanged even if writing to the CM20 bit.
8. Where the CM20 bit is set to “1” (oscillation stop, re-oscillation detection function enabled), the CM27 bit is “1” (oscillation stop, re-oscillation detection interrupt), and the CM11 bit is set to “1” (PLL clock is selected as the CPU clock source), the CM21 bit remains unchanged even if a main clock stop is detected. When the CM22 bit is set to “0” under these conditions, an oscillation stop, a re-oscillation detection interrupt request is generated at main clock stop detection. Set the CM21 bit to “1” (on-chip oscillator clock) in the interrupt routine.
9. Set the CM20 bit to “0” (disabled) before entering stop mode. Exit stop mode before setting the CM20 bit back to “1” (enabled).
10. Set the CM20 bit in the CM2 register to “0” (disabled) before setting the CM05 bit in the CM0 register to “1” (main clock stops).
11. The CM20, CM21 and CM27 bits remain unchanged at the oscillation stop detection reset.
12. When the CM21 bit is set to “0” (on-chip oscillator stops) and the CM05 bit is set to “1” (main clock stops), the CM06 bit is fixed to “1” (divide-by-8 mode) and the CM15 bit is fixed to “1” (drive capacity High).

A.2 System Clock Control Register 0

System Clock Control Register 0 (1)



Symbol	Address	After Reset
CM0	0006h	01001000b

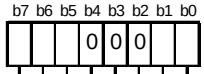
Bit Symbol	Bit Name	Function	RW
CM00	Clock Output Function Select Bit (Valid only in single-chip mode)	b1 b0 0 0 : I/O port P5_7 0 1 : Output fC 1 0 : Output f8 1 1 : Output f32	RW
CM01			RW
CM02	WAIT Mode Peripheral Function Clock Stop Bit (10)	0 : Peripheral function clock does not stop in wait mode 1 : Peripheral function clock stops in wait mode(8)	RW
CM03	XCIN-XCOUT Drive Capacity Select Bit (2)	0 : LOW 1 : HIGH	RW
CM04	Port XC Select Bit (2)	0 : I/O ports P8_6, P8_7 1 : XCIN-XCOUT oscillation function (9)	RW
CM05	Main Clock Stop Bit (3, 10, 12, 13)	0 : On 1 : Off (4, 5)	RW
CM06	Main Clock Division Select Bit 0 (7, 13, 14)	0 : CM16 and CM17 enabled 1 : Division-by-8 mode	RW
CM07	System Clock Select Bit (6, 10, 11, 12)	0 : Main clock, PLL clock, or on-chip oscillator clock 1 : Sub clock	RW

NOTES :

1. Rewrite this register after setting the PRC0 bit in the PRCR register to “1” (write enable).
2. The CM03 bit is set to “1” (high) while the CM04 bit is set to “0” (I/O port) or when entering stop mode.
3. This bit is provided to stop the main clock when the low power consumption mode or on-chip oscillator low power dissipation mode is selected. This bit cannot be used for detection as to whether the main clock stops or not. To stop the main clock, set bits as follows:
   (a) Set the CM07 bit to “1” (sub clock selected) or the CM21 bit in the CM2 register to “1” (On-chip oscillator selected)
   with the sub-clock stably oscillates.
   (b) Set the CM20 bit in the CM2 register to “0” (Oscillation stop, re-oscillation detection function disabled).
   (c) Set the CM05 bit to “1” (Stop).
4. During external clock input, Set the CM05 bit to “0” (oscillate).
5. When CM05 bit is set to “1”, the XOUT pin is held “H”. Because the internal feedback resistor remains connected, the XIN pin is pulled “H” to the same level as XOUT via the feedback resistor.
6. After setting the CM04 bit to “1” (XCIN-XCOUT oscillator function), wait until the sub-clock oscillates stably before switching the CM07 bit from “0” to “1” (sub-clock).
7. When entering stop mode from high-speed or middle-speed mode, on-chip oscillator mode or on-chip oscillator low power mode, the CM06 bit is set to “1” (divide-by-8 mode).
8. The fC32 clock does not stop. In low-speed mode or low power consumption mode, do not set this bit to “1” (peripheral clock stops in wait mode).
9. To use a sub-clock, set this bit to “1”. Also make sure ports P8_6 and P8_7 are directed for input, with no pull-ups.
10. When the PM21 bit in the PM2 register is set to “1” (disable clock modification), this bit remains unchanged even if writing to the CM02, CM05, and CM07 bits.
11. When setting the PM21 bit to “1”, set the CM07 bit to “0” (main clock) before setting the PM21 bit to “1”.
12. To use the main clock as the clock source for the CPU clock, set bits as follows.
    (a) Set the CM05 bit to “0” (oscillate).
    (b) Wait the main clock oscillation stabilizes.
    (c) Set the CM11, CM21 and CM07 bits to “0”.
13. When the CM21 bit is set to “0” (on-chip oscillator stops) and the CM05 bit is set to “1” (main clock stops), the CM06 bit is fixed to “1” (divide-by-8 mode) and the CM15 bit is fixed to “1” (drive capacity High).
14. To return from on-chip oscillator mode to high-speed or middle-speed mode, set the CM06 and CM15 bits to “1”.

A.3 System Clock Control Register 1

System Clock Control Register 1(1)



Symbol	Address	After Reset
CM1	0007h	00100000b

Bit Symbol	Bit Name	Function	RW
CM10	All Clock Stop Control Bit (4, 6)	0 : Clock on 1 : All clocks off (stop mode)	RW
CM11	System Clock Select Bit 1 (6, 7)	0 : Main clock 1 : PLL clock (5)	RW
— (b4-b2)	Reserved Bit	Set to “0”	RW
CM15	XIN-XOUT Drive Capacity Select Bit (2)	0 : LOW 1 : HIGH	RW
CM16	Main Clock Division Select Bit 1 (3)	b7 b6 0 0 : No division mode 0 1 : Divide-by-2 mode 1 0 : Divide-by-4 mode 1 1 : Divide-by-16 mode	RW
CM17			RW

NOTES :

1. Rewrite this register after setting the PRC0 bit in the PRCR register to “1” (write enable).
2. When entering stop mode from high-speed or middle-speed mode, or the CM05 bit is set to “1” (main clock stops) in low speed mode, the CM15 bit is set to “1” (drive capacity high).
3. This bit is valid when the CM06 bit is set to “0” (CM16 and CM17 bits enabled).
4. If the CM10 bit is set to “1” (stop mode), XOUT is held “H” and the internal feedback resistor is disconnected. The XCIN and XCOUT pins are in high-impedance state. When the CM11 bit is set to “1” (PLL clock), or the CM20 bit in the CM2 register is set to “1” (oscillation stop, re-oscillation detection function enabled), do not set the CM10 bit to “1”.
5. After setting the PLC07 bit in the PLC0 register to “1” (PLL operation), wait tsu (PLL) elapses before setting the CM11 bit to “1” (PLL clock).
6. When the PM21 bit in the PM2 register is set to “1” (disable clock modification), this bit remains unchanged even if writing to the CM10, CM11 bits.
   When the PM22 bit in the PM2 register is set to “1” (on-chip oscillator clock is selected as watchdog timer count source), this bit remains unchanged even if writing to the CM10 bit.
7. This bit is valid when the CM07 bit is set to “0” and the CM21 bit is set to “0”.

A.4 Oscillation Stop, Re-Oscillation Detection Interrupt Operation (CM27 bit=1)

In the sample program, the PLL clock is used as the CPU clock source. In this case, the oscillation stop, re-oscillation detection interrupt operations are shown below:

(1) When the main clock stops with the following states.

- The oscillation stop, re-oscillation detection interrupt request are generated.
- The CM22 bit is set to “1”. (The main clock stop is detected.)
- The CM23 bit is set to “1”. (The main clock stops.)
- The CM21 bit remains unchanged.

NOTES : When using the PLL clock as the CPU clock source, the CM21 bit remains unchanged even if the main clock stops. Set the CM21 bit to “1” (on-chip oscillator clock (on-chip oscillator on)) in the interrupt routine.

(2) When the main clock re-oscillates from the state that the main clock stops with the following states.

- The oscillation stop, re-oscillation detection interrupt request are generated.
- The CM22 bit is set to “1”. (The main clock stop is detected.)
- The CM23 bit is set to “0”. (The main clock oscillates.)
- The CM21 bit remains unchanged.

A.5 How to Use Oscillation Stop, Re-Oscillation Detection Function

- The oscillation stop, re-oscillation detection interrupt use both of the watchdog timer interrupt and vector. When using both the oscillation stop, re-oscillation detection interrupt and the watchdog timer interrupt, read the CM22 bit (oscillation stop, re-oscillation detection flag) in the interrupt routine and determine the interrupt request by which interrupt factor of the oscillation stop, re-oscillation detection interrupt or the watchdog timer interrupt.
- When the main clock re-oscillates after the oscillation stops, set the main clock back to the clock source of the CPU clock or peripheral functions by a program.
- The CM22 bit is set to “1” at the same time as the oscillation stop, re-oscillation detection interrupt generation. When the CM22 bit is se to “1”, the oscillation stop, re-oscillation detection interrupt is disabled. When setting the CM22 bit to “0” by a program, the oscillation stop, re-oscillation detection interrupt is enabled.
- When the CM20 bit is set to “1” and the main clock stops in low-speed mode, the oscillation stop, re-oscillation detection interrupt request is generated. At the same time, the on-chip oscillator starts oscillating. At this time, the sub clock remains as the clock source for the CPU clock. However, the on-chip oscillator clock is selected as the clock source for the peripheral function clock.
- When entering wait mode while the oscillation stop, re-oscillation detection function is used, set the CM02 bit to “0” (the peripheral function clock does not stop in wait mode).
- The oscillation stop, re-oscillation detection function is to provide for the main clock stop by the external factors. When stopping or oscillating the main clock by a program, in other words, when setting to stop mode or changing the CM05 bit, set the CM20 bit to “0” (oscillation stop, re-oscillation detection function disabled).
- When the main clock frequency is 2MHz or less, this function cannot be used. Set the CM20 bit to “0”.

REVISION HISTORY	M16C/62P Group Application of Oscillation Stop, Re-Oscillation Detection Function

Rev.	Date	Description	
		Page	Summary
1.01	June 23, 2005	–	First Edition issued

Keep safety first in your circuit designs!

1. Renesas Technology Corporation puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage. Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of non-flammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corporation product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corporation or a third party.
2. Renesas Technology Corporation assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corporation without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corporation or an authorized Renesas Technology Corporation product distributor for the latest product information before purchasing a product listed herein.
   The information described here may contain technical inaccuracies or typographical errors. Renesas Technology Corporation assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.
   Please also pay attention to information published by Renesas Technology Corporation by various means, including the Renesas Technology Corporation Semiconductor home page (http://www.renesas.com).
4. When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corporation assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corporation semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corporation or an authorized Renesas Technology Corporation product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
6. The prior written approval of Renesas Technology Corporation is necessary to reprint or reproduce in whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.
   Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
8. Please contact Renesas Technology Corporation for further details on these materials or the products contained therein.