RX210, RX21A, and RX220 Groups

R01AN1706EJ0100
Rev. 1.00
Oct. 10, 2014

HOCO Calibration Using the CAC

Abstract

This document describes the method to adjust frequencies of the high-speed on-chip oscillator (HOCO) using the clock frequency accuracy measurement circuit (CAC) during the RX210, RX21A or RX220 MCU operation.

Products

- RX210, RX21A, and RX220 Groups

Note: • 48-pin packages of the RX210 and RX220 Groups are not included as the target products.

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

Contents

1. Specifications

HOCO oscillation frequency may differ from the factory-configured frequency due to external factors such as ambient temperature. Calibration needs to be performed to compensate the frequency error. The high-speed on-chip oscillator trimming register (HOCOTRRn (n = 0 to 3)) is used to adjust the HOCO oscillation frequency. Registers HOCOTRR0, HOCOTRR1, HOCOTRR2, and HOCOTRR3 correspond to oscillation frequencies 32 MHz, 36.864 MHz, 40 MHz, and 50 MHz, respectively. When the HOCO clock is 32 MHz, the HOCO frequency error can be compensated by adjusting the value of the HOCOTRR0 register in constant intervals.

In this application note, the HOCO oscillation frequency is measured in 1 second periods using the CAC. Then the value in the HOCOTRR0 register is adjusted with the obtained measurement result. Also a waveform which is equivalent to HOCO divided by 4 is output from the TMO3 pin.

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

Table 1.1 Peripheral Functions and Their Applications

Peripheral Function	Application
CAC	Measures the HOCO frequency based on the sub-clock.
RTC	Counter for 1 second period
TMR	Outputs a waveform which is equivalent to the HOCO clock divided by 4.

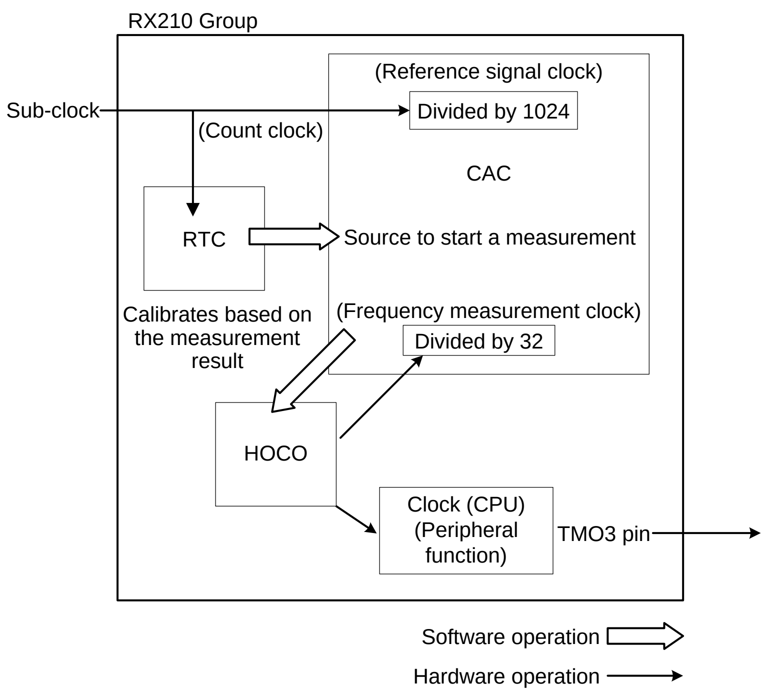


Figure 1.1 Block Diagram

2. Operation Confirmation Conditions

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

Table 2.1 Operation Confirmation Conditions

Item	Contents
MCU used	R5F5210BBDFP (RX210 Group)
Operating frequencies	• HOCO clock: 32 MHz • Sub-clock: 32.768 kHz • System clock (ICLK): 32 MHz (HOCO clock divided by 1)
Operating voltage	5.0 V
Integrated development environment	Renesas Electronics Corporation High-performance Embedded Workshop Version 4.09.01
C compiler	Renesas Electronics Corporation C/C++ Compiler Package for RX Family V.1.02 Release 01
	Compile options -cpu=rx200 -output=obj="$(CONFIGDIR)\$(FILELEAF).obj" -debug -nologo (The default setting is used in the integrated development environment.)
iodefine.h version	Version 1.50
Endian	Little endian
Operating mode	Single-chip mode
Processor mode	Supervisor mode
Sample code version	Version 1.00
Board used	Renesas Starter Kit for RX210 (product part no.: R0K505210S003BE)

3. Reference Application Notes

For additional information associated with this document, refer to the following application notes.

- RX210 Group Initial Setting Rev. 2.20 (R01AN1002EJ)
- RX21A Group Initial Setting Rev. 1.10 (R01AN1486EJ)
- RX220 Group Initial Setting Rev. 1.10 (R01AN1494EJ)

The initial setting functions in the reference application notes are used in the sample code accompanying this application note. The revision numbers of the reference application notes are as of when this application note was made. However the latest version is always recommended. Visit the Renesas Electronics Corporation website to check and download the latest version.

4. Hardware

4.1 Pin Used

Table 4.1 lists the Pin Used and Its Function.

Table 4.1 Pin Used and Its Function

Pin Name	I/O	Function
P32/TMO3	Output	Outputs a waveform which is equivalent to the HOCO clock divided by 4.

5. Software

5.1 Operation Overview

In this application note, the HOCO clock (32 MHz) divided by 32 is used as the frequency measurement clock for the CAC and the sub-clock (32.768 kHz) divided by 1024 is used as the reference signal generation clock. The CAC counts valid edges of the HOCO in the period from a rising edge to the next rising edge of the sub-clock.

The measurement is triggered by the realtime clock (RTC) 1-second periodic interrupt and performed four times continuously.

The average of these four measurement results is calculated (measurement result for adjustment). If the difference (A) between the measurement result and the theoretical value is "A ≤ -256 counts" or "+256 counts ≤ A", addition or subtraction is performed on the HOCOTRR0 register value. If the difference between the measurement result and the theoretical value is "-256 counts < A < +256 counts", the HOCOTRR0 register value remains as it is.

Table 5.1 lists the Specifications of Register Value Adjustment, Figure 5.1 shows the Specification of Register Value Adjustment, and Figure 5.2 shows the Timing Diagram of Calibration.

Table 5.1 Specifications of Register Value Adjustment

Pattern	Condition	Adjustment Value
1	X ≥ Z (Z + L)	- 1
2	X ≤ (Z - L)	+ 1
3	(Z - L) < X < (Z + L)	As is

X: Measurement result, Z: Theoretical value, L: Allowable error range

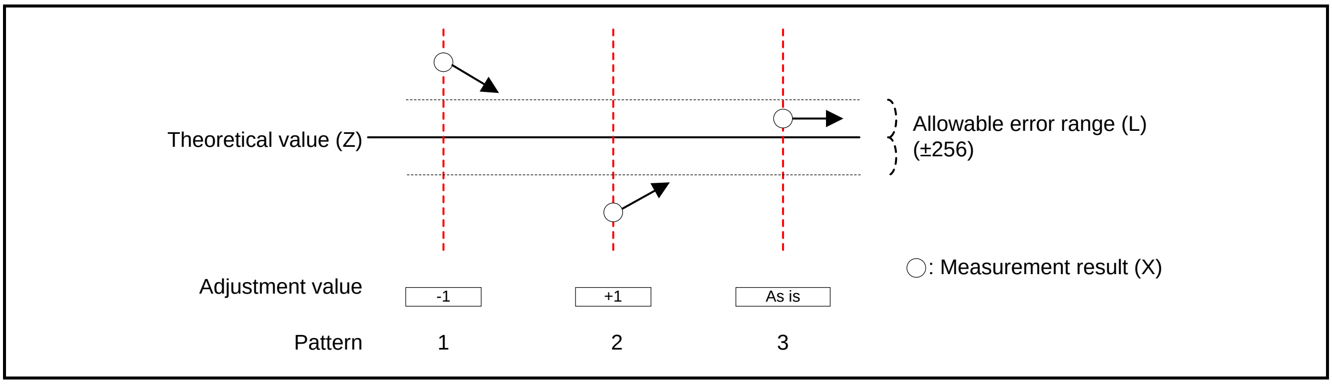


Figure 5.1 Specification of Register Value Adjustment

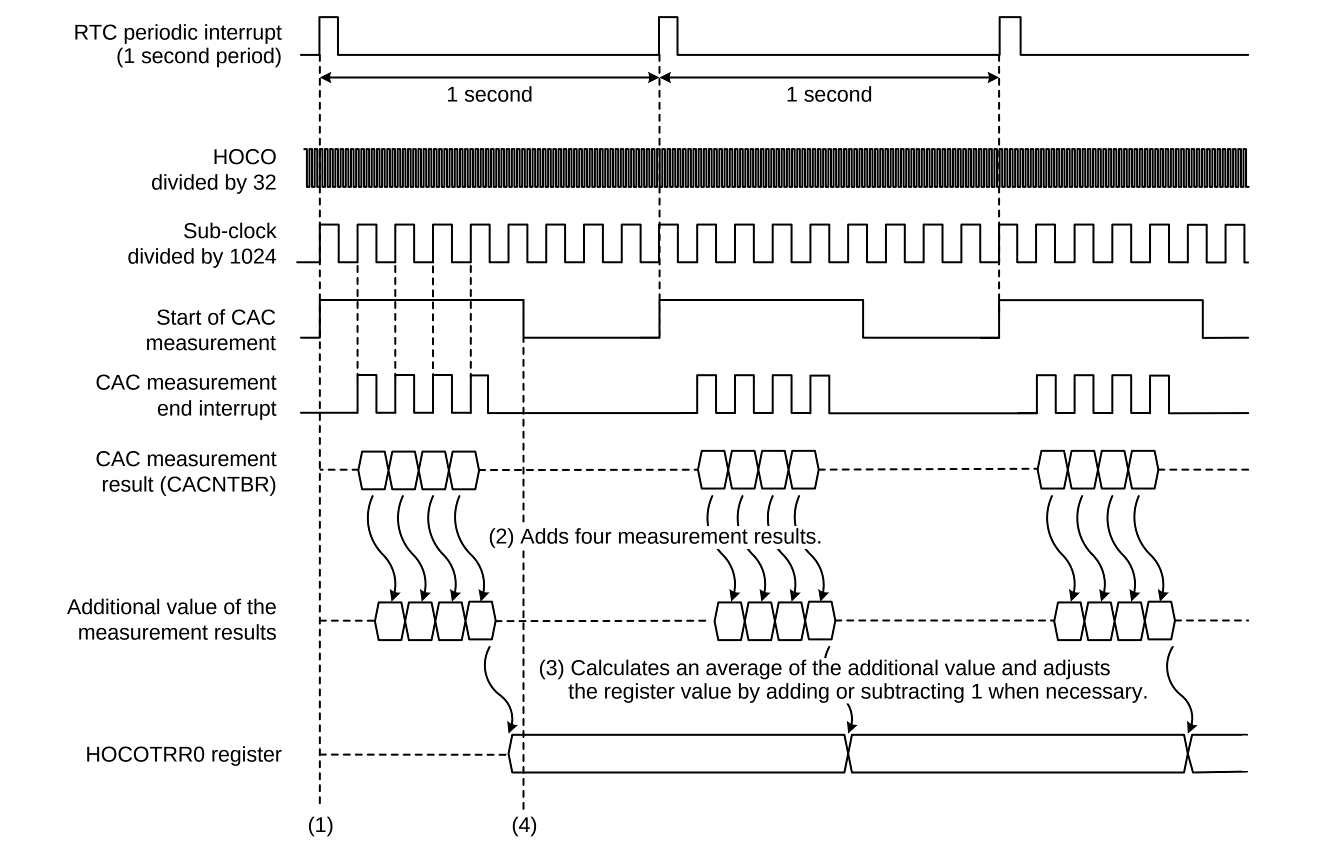


Figure 5.2 Timing Diagram of Calibration

(1) Using the RTC 1-second periodic interrupt as a trigger, the CAC start setting is configured and a measurement is started.

(2) Valid edges of the HOCO divided by 32 are counted in the period from a rising edge to the next rising edge of the sub-clock divided by 1024. The measurement result is obtained in CAC measurement end interrupt handling. Four measurements are handled as a set of the measurement result.

(3) A set of the measurement result is averaged and the value in the HOCOTRR0 register is adjusted using the averaged value [(1)].

(4) When the adjustment of the HOCO oscillation frequency is completed, the CAC measurement is stopped. A wait for HOCO oscillation stabilization is processed until the next RTC 1-second periodic interrupt occurs while HOCO is being provided to the CPU or the peripheral functions such as ICLK.

Note:

1. The following errors are caused by errors between the HOCO and a crystal used for the sub-clock, and the allowable error range (±256 counts) of the theoretical value.
   - When the HOCO is 32 MHz, the maximum error is ±0.9%.
   - When the HOCO is 36.864 MHz, the maximum error is ±0.8%.
   - When the HOCO is 40 MHz, the maximum error is ±0.7%.
   - When the HOCO is 50 MHz, the maximum error is ±0.6%.

5.2 File Composition

Table 5.2 lists the Files Used in the Sample Code. Files generated by the integrated development environment are not included in this table.

Table 5.2 Files Used in the Sample Code

File Name	Outline	Remarks
main.c	Main processing	
r_init_stop_module.c	Stop processing for active peripheral functions after a reset	
r_init_stop_module.h	Header file for r_init_stop_module.c	
r_init_non_existent_port.c	Nonexistent port initialization	
r_init_non_existent_port.h	Header file for r_init_non_existent_port.c	
r_init_clock_an1706.c	Clock initialization	
r_init_clock.h	Header file for r_init_clock.c	
rtc_func.c	RTC initialization and 1-second periodic interrupt handling	
rtc_func.h	Header file for rtc_func.c	
cac_func.c	CAC initialization and measurement end interrupt handling	
cac_func.h	Header file for cac_func.c	
tmr_func.c	TMR initialization for PCLK output	

5.3 Option-Setting Memory

Table 5.3 lists the Option-Setting Memory Configured in the Sample Code. When necessary, set a value suited to the user system.

Table 5.3 Option-Setting Memory Configured in the Sample Code

Symbol	Addresses	Setting Value	Contents
OFS0	FFFF FF8Fh to FFFF FF8Ch	FFFF FFFFh	The IWDT is stopped after a reset. The WDT is stopped after a reset.
OFS1	FFFF FF8Bh to FFFF FF88h	FFFF FFFFh	The voltage monitor 0 reset is disabled after a reset. HOCO oscillation is disabled after a reset.
MDES	FFFF FF83h to FFFF FF80h	FFFF FFFFh	Little endian

5.4 Constants

Table 5.4 lists the Constants Used in the Sample Code (Constants from the Initial Setting). Table 5.5 list the Constants Used in the Sample Code.

Table 5.4 Constants Used in the Sample Code (Constants from the Initial Setting)

Constants listed in the table are changed their values from those used in the Initial Setting application note.

Constant Name	Setting Value	Contents
SEL_MAIN	B_NOT_USE	Selection of the main clock operation: The main clock is not used (main clock stopped).
SEL_SUB	B_USE	Selection of the sub-clock usage for the system clock: The sub-clock is used (1).
SEL_RTC	B_USE	Selection of the sub-clock usage for the RTC count source: The sub-clock is used.
SEL_PLL	B_NOT_USE	Selection of the PLL clock operation: The PLL clock is not used (PLL clock stopped).
SEL_HOCO	B_USE	Selection of the HOCO clock operation: The HOCO clock is used (HOCO clock oscillating).
SEL_SYSCLK	CLK_HOCO	Clock source selection for the system clock HOCO is selected.

Note:

1. For the clock setting, No. 4 is used in Table 1.3 Examples of Clock Selections in the Initial Setting application note except the SEL_SUB constant. In this application note, “B_USE (used)” is set to the SEL_SUB constant since the sub-clock is used for the CAC.

Table 5.5 Constants Used in the Sample Code

Constant Name	Setting Value	Contents
CHECK_CNT	4	Number of measurements
MEND_FINISH	0	Frequency measurement counter: Measurement stopped
MEND_START	1	Frequency measurement counter: Measurement starts
MEND_CHECK_FINISH	CHECK_CNT	Frequency measurement counter: Measurement completed
ACCEPTABLE_RANGE	256	Allowable error range of the measurement result
MAIN_CLOCK_CYCLE	(1,000,000,000L /MAIN_CLOCK_Hz)	Main clock cycles (ns)
SUB_CLOCK_CYCLE	(1,000,000,000L / SUB_CLOCK_Hz)	Sub-clock cycles (ns)
FOR_CMT0_TIME	232727	Count cycles (ns) for the CMT0 timer to wait for the oscillation stabilization wait time: (1/LOCO) × 32, where LOCO = 137.5 kHz (max.), and 32 = PCLKB divided by 32
REG_HOCOTRR (when HOCO operates at 32 MHz)	SYSTEM.HOCOTRR0	HOCOTRR0 register
TYP_CAC_RESULT (when HOCO operates at 32 MHz)	31250	Measurement theoretical value when HOCO operates at 32 MHz.
REG_HOCOTRR (when HOCO operates at 36.864 MHz)	SYSTEM.HOCOTRR1	HOCOTRR1 register
TYP_CAC_RESULT (when HOCO operates at 36.864 MHz)	36000	Measurement theoretical value when HOCO operates at 36.864 MHz.
REG_HOCOTRR (when HOCO operates at 40 MHz)	SYSTEM.HOCOTRR2	HOCOTRR2 register
TYP_CAC_RESULT (when HOCO operates at 40 MHz)	39062	Measurement theoretical value when HOCO operates at 40 MHz.
REG_HOCOTRR (when HOCO operates at 50 MHz)	SYSTEM.HOCOTRR3	HOCOTRR3 register
TYP_CAC_RESULT (when HOCO operates at 50 MHz)	48828	Measurement theoretical value when HOCO operates at 50 MHz.

5.5 Variables

Table 5.6 lists the Global Variables.

Table 5.6 Global Variables

Type	Variable Name	Contents	Function Used
static uint8_t	buffer_counter	Frequency measurement counter	INIT_CAC Excep_CAC_MENDF
static uint32_t	result_buffer	Storage buffer for the frequency measurement result	INIT_CAC Excep_CAC_MENDF
static int32_t	result_diff	Storage buffer for the result	INIT_CAC Excep_CAC_MENDF

5.6 Functions

Table 5.7 lists the Functions.

Table 5.7 Functions

Function Name	Outline
main	Main processing
R_INIT_StopModule	Stop processing for active peripheral functions after a reset
R_INIT_NonExistentPort	Nonexistent port initialization
R_INIT_Clock	Clock initialization
tmr_init	TMR initialization
CGC_oscillation_sub_an1706	Sub-clock oscillation setting
enable_RTC_an1706	RTC initialization
Excep_RTC_PRD	RTC periodic interrupt handling
cmt0_wait_16bit	CMT0 wait processing
INIT_CAC	CAC initialization
Excep_CAC_MENDF	CAC measurement end interrupt handling

5.7 Function Specifications

The following tables list the sample code function specifications.

main	
Outline	Main processing
Header	None
Declaration	void main(void)
Description	Calls the following functions: Stop processing for active peripheral functions after a reset, nonexistent port initialization, clock initialization, and TMR initialization.
Arguments	None
Return Value	None

R_INIT_StopModule	
Outline	Stop processing for active peripheral functions after a reset
Header	r_init_stop_module.h
Declaration	void R_INIT_StopModule(void)
Description	Configures the setting to enter the module stop state.
Arguments	None
Return Value	None
Remarks	Transition to the module stop state is not performed in the sample code. For details on this function, refer to the Initial Setting application note for the product used.

R_INIT_NonExistentPort	
Outline	Nonexistent port initialization
Header	r_init_non_existent_port.h
Declaration	void R_INIT_NonExistentPort(void)
Description	Initializes port direction registers for ports that do not exist in products with less than 144 pins.
Arguments	None
Return Value	None
Remarks	The number of pins in the sample code is set for the 100-pin package (PIN_SIZE=100). After this function is called, when writing in byte units to the PDR registers or PODR registers which have nonexistent ports, set the corresponding bits for nonexistent ports as follows: set the I/O select bits in the PDR registers to 1 and set the output data store bits in the PODR registers to 0. For details on this function, refer to the Initial Setting application note for the product used.

R_INIT_Clock	
Outline	Clock initialization
Header	r_init_clock.h
Declaration	void R_INIT_Clock(void)
Description	Initializes the clock.
Arguments	None
Return Value	None
Remarks	The sample code selects processing which uses the HOCO as the system clock and the sub-clock for the RTC and CAC. For details on this function, refer to the Initial Setting application note for the product used.

tmr_init	
Outline	TMR initialization
Header	tmr_func.h
Declaration	void tmr_init(void)
Description	Configures channel 3 of the TMR to output PCLK divided by 2.
Arguments	None
Return Value	None

CGC_oscillation_sub_an1706	
Outline	Sub-clock oscillation setting
Header	r_init_clock.h
Declaration	void CGC_oscillation_sub_an1706 (void)
Description	Configures settings to use the sub-clock for either of the system clock or the RTC, or both.
Arguments	None
Return Value	None

enable_RTC_an1706	
Outline	RTC initialization
Header	rtc_func.h
Declaration	void enable_RTC_an1706 (void)
Description	Initializes the RTC (setting for clock provision and the RTC, and RTC software reset).
Arguments	None
Return Value	None

Excep_RTC_PRD	
Outline	RTC periodic interrupt handling
Header	rtc_func.h
Declaration	static void Excep_RTC_PRD(void)
Description	The CAC is initialized in this interrupt handling.
Arguments	None
Return Value	None

cmt0_wait_16bit	
Outline	CMT0 wait processing
Header	None
Declaration	static void cmt0_wait_16bit(uint16_t cnt)
Description	This function is used when waiting for RTC clock provision that is six clocks of the sub-clock.
Arguments	uint16_t cnt: Oscillation stabilization wait time cnt = oscillation stabilization wait time (ns) (1) ÷ FOR_CMT0_TIME (2)
Return Value	None
Remarks	1. The oscillation stabilization wait time varies depending on the crystal/ceramic resonator. Set the value based on the calculation method in the Initial Setting application note for the product used. 2. The value of FOR_CMT0_TIME is calculated where LOCO is 137.5 kHz (max.). The actual wait time may differ depending on the LOCO frequency.

INIT_CAC	
Outline	CAC initialization
Header	cac_func.h
Declaration	void INIT_CAC(void)
Description	Initializes the CAC.
Arguments	None
Return Value	None

Excep_CAC_MENDF	
Outline	CAC measurement end interrupt handling
Header	cac_func.h
Declaration	static void Excep_CAC_MENDF(void)
Description	Adjusts HOCO frequencies based on the measurement results.
Arguments	None
Return Value	None

5.8 Flowcharts

5.8.1 Main Processing

Figure 5.3 shows the Main Processing.

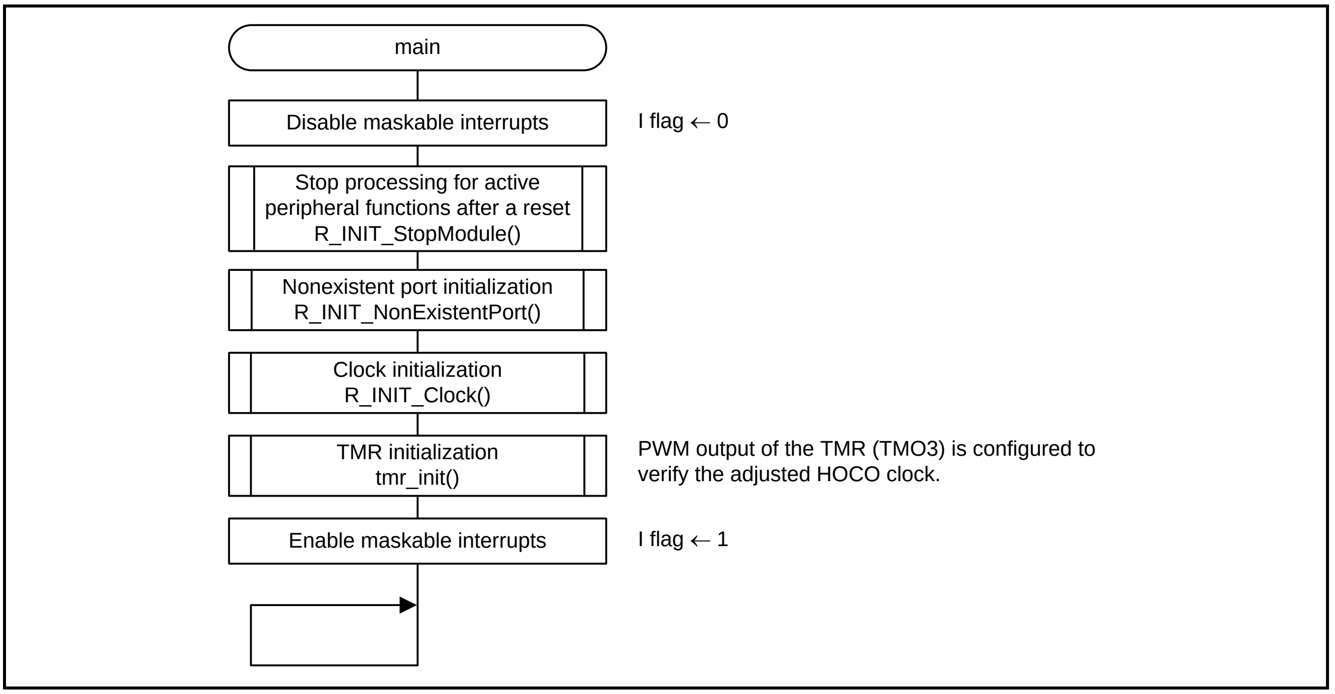


Figure 5.3 Main Processing

5.8.2 TMR Initialization

Figure 5.4 shows the TMR Initialization.

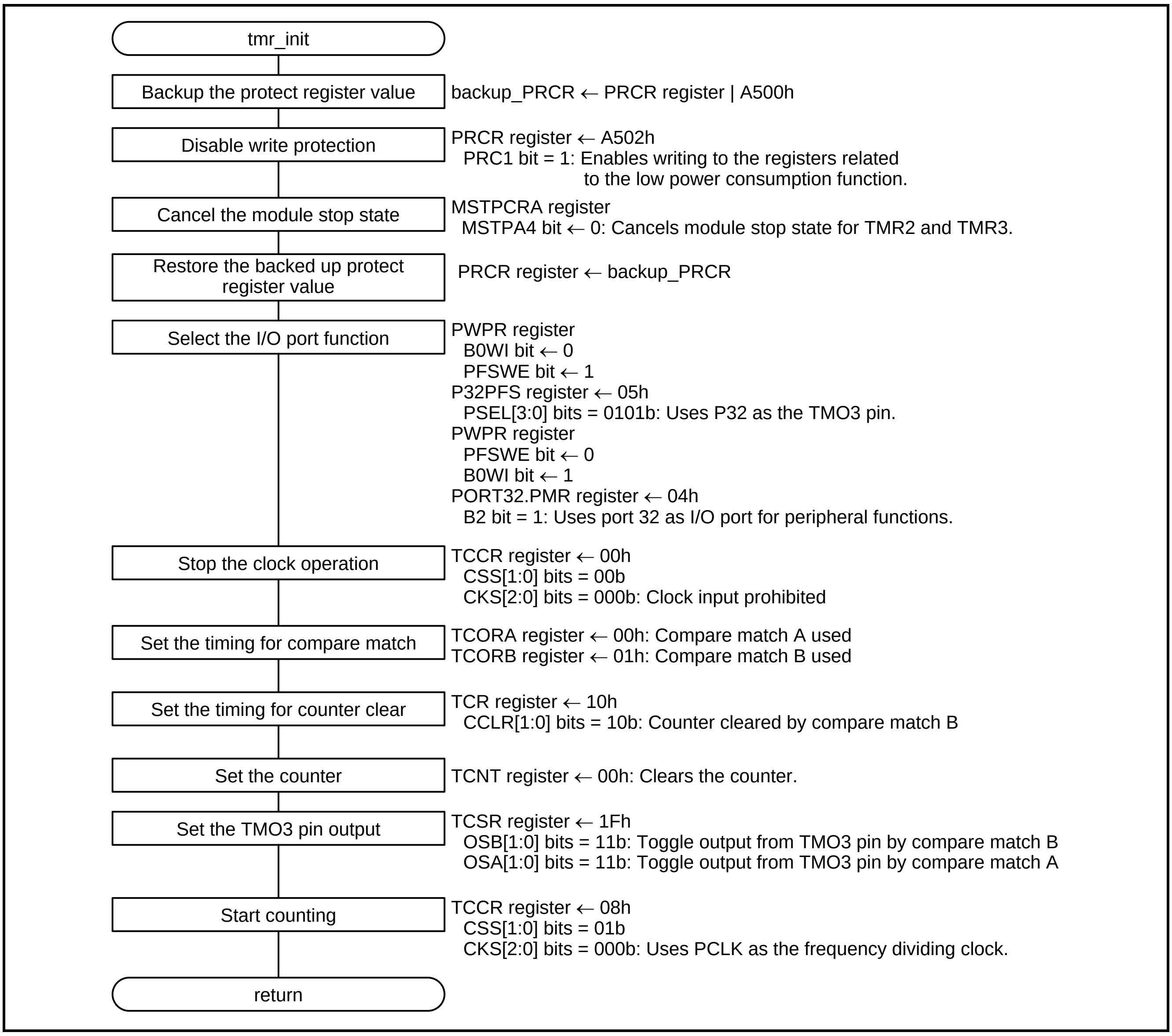


Figure 5.4 TMR Initialization

5.8.3 Sub-Clock Oscillation Setting

Figure 5.5 shows the Sub-Clock Oscillation Setting.

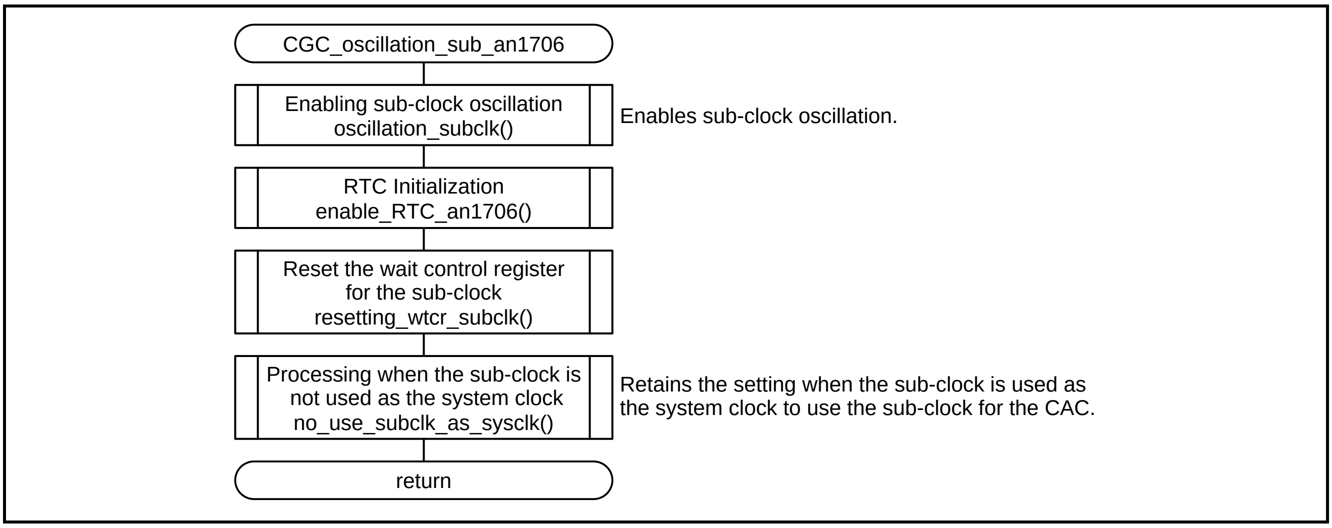


Figure 5.5 Sub-Clock Oscillation Setting

5.8.4 RTC Initialization

Figure 5.6 shows the RTC Initialization.

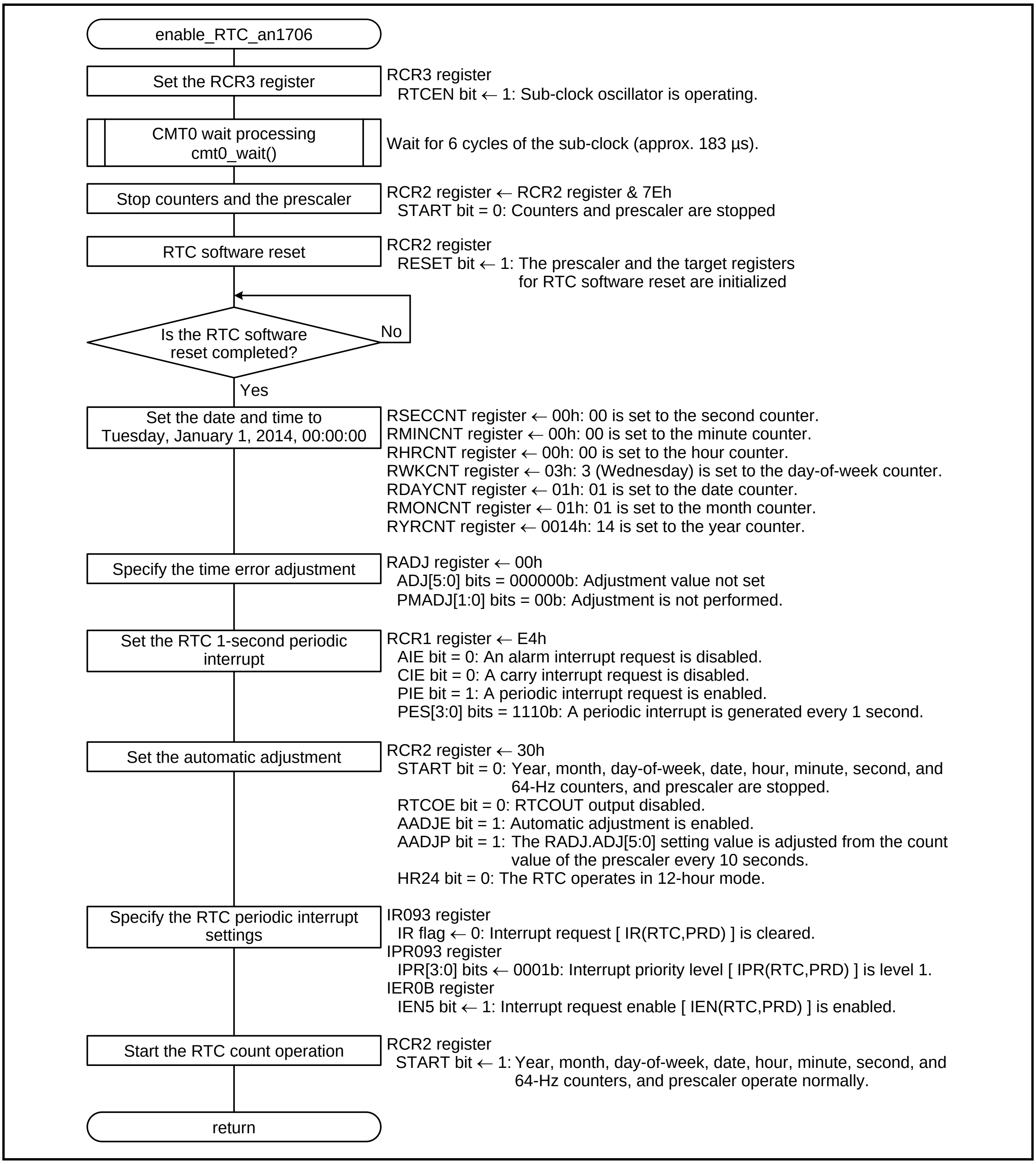


Figure 5.6 RTC Initialization

5.8.5 RTC Periodic Interrupt Handling

Figure 5.7 shows the RTC Periodic Interrupt Handling.

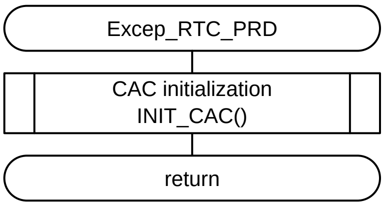
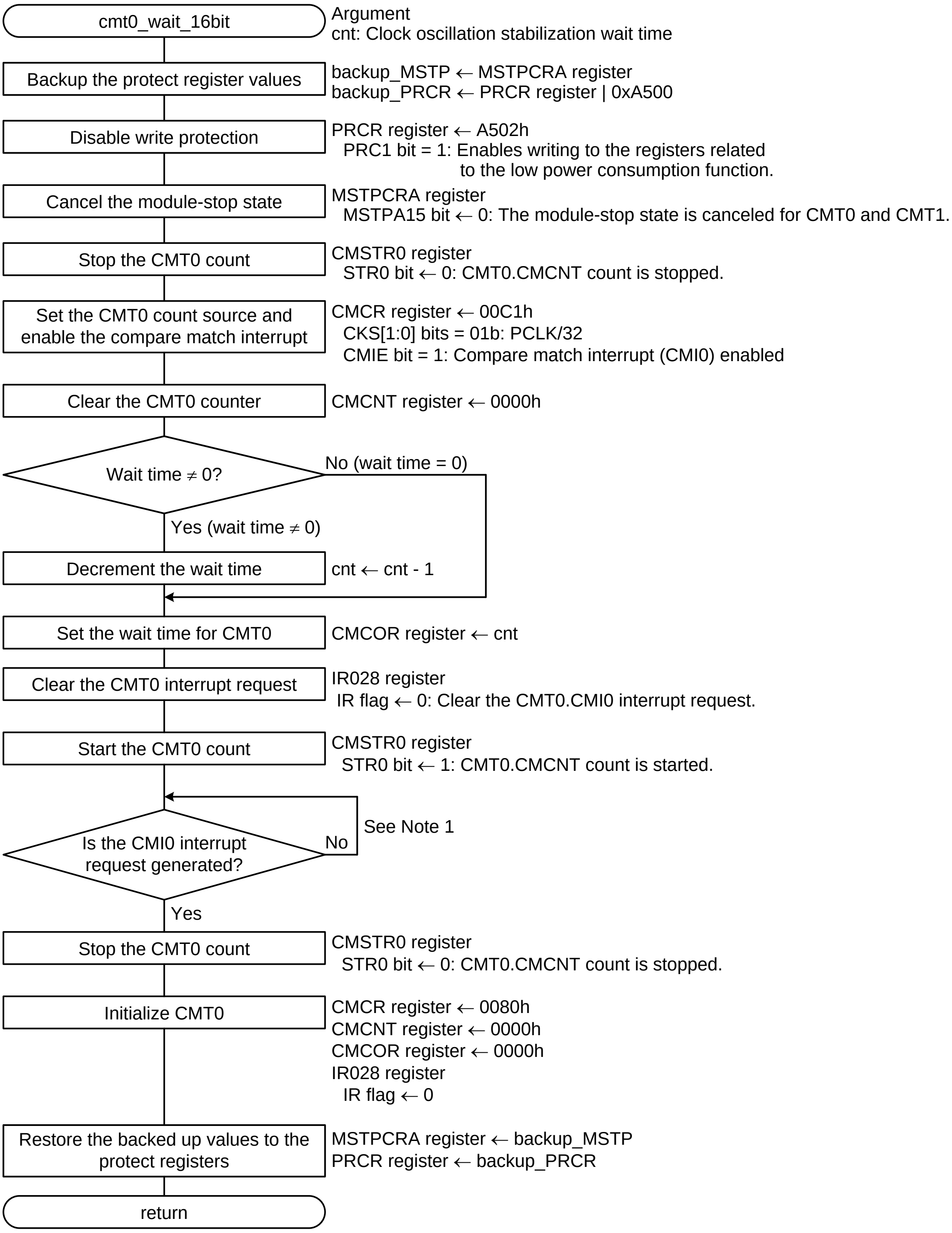


Figure 5.7 RTC Periodic Interrupt Handling

5.8.6 CMT0 Wait Processing

Figure 5.8 shows the CMT0 Wait Processing.



Note:
1. When the counter of the watchdog timer (WDT) or independent watchdog timer (IWDT) is operating, refresh the counter in this loop as required.

Figure 5.8 CMT0 Wait Processing

5.8.7 CAC Initialization

Figure 5.9 shows the CAC Initialization.

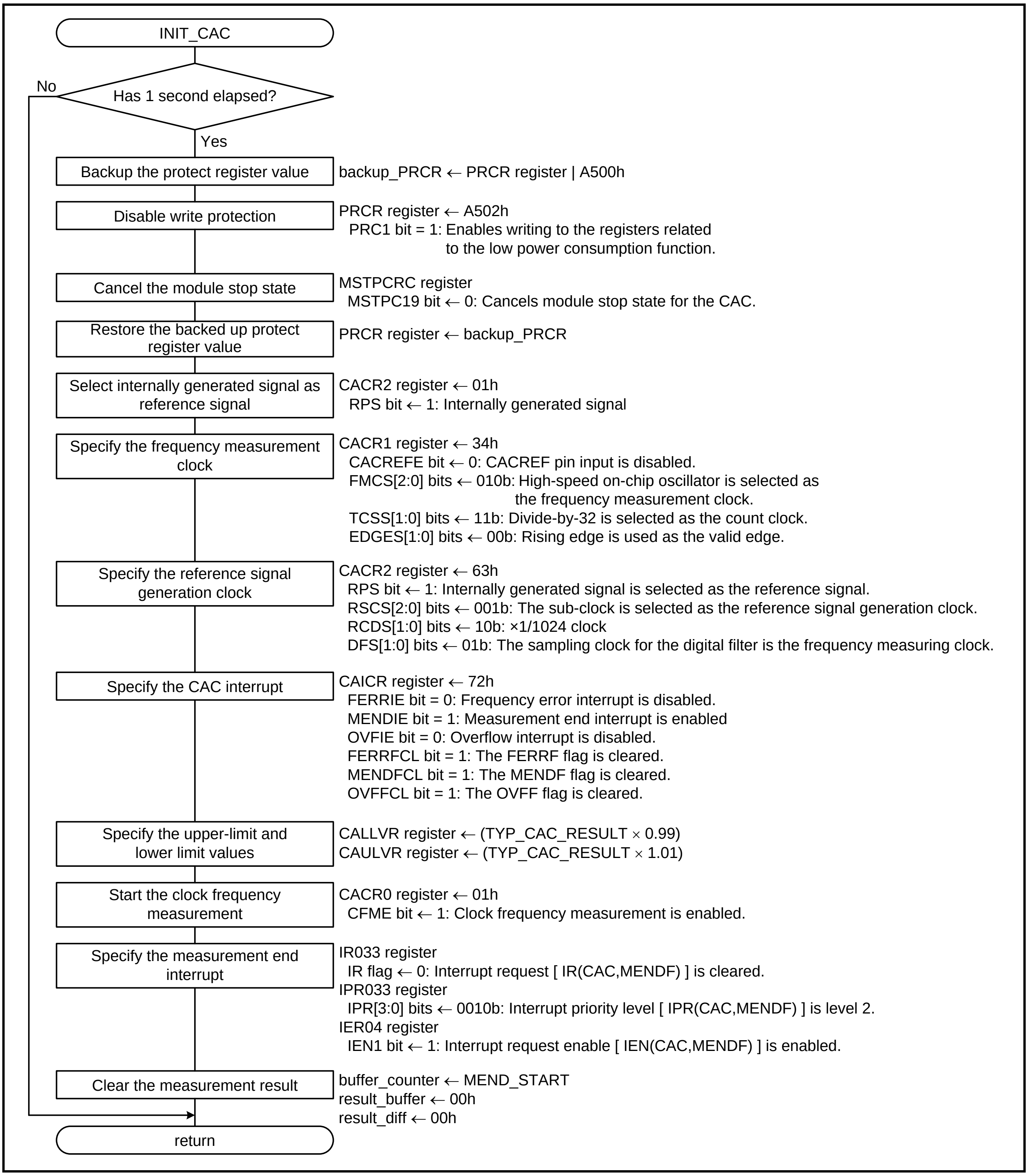


Figure 5.9 CAC Initialization

5.8.8 CAC Measurement End Interrupt Handling

Figure 5.10 shows the CAC Measurement End Interrupt Handling.

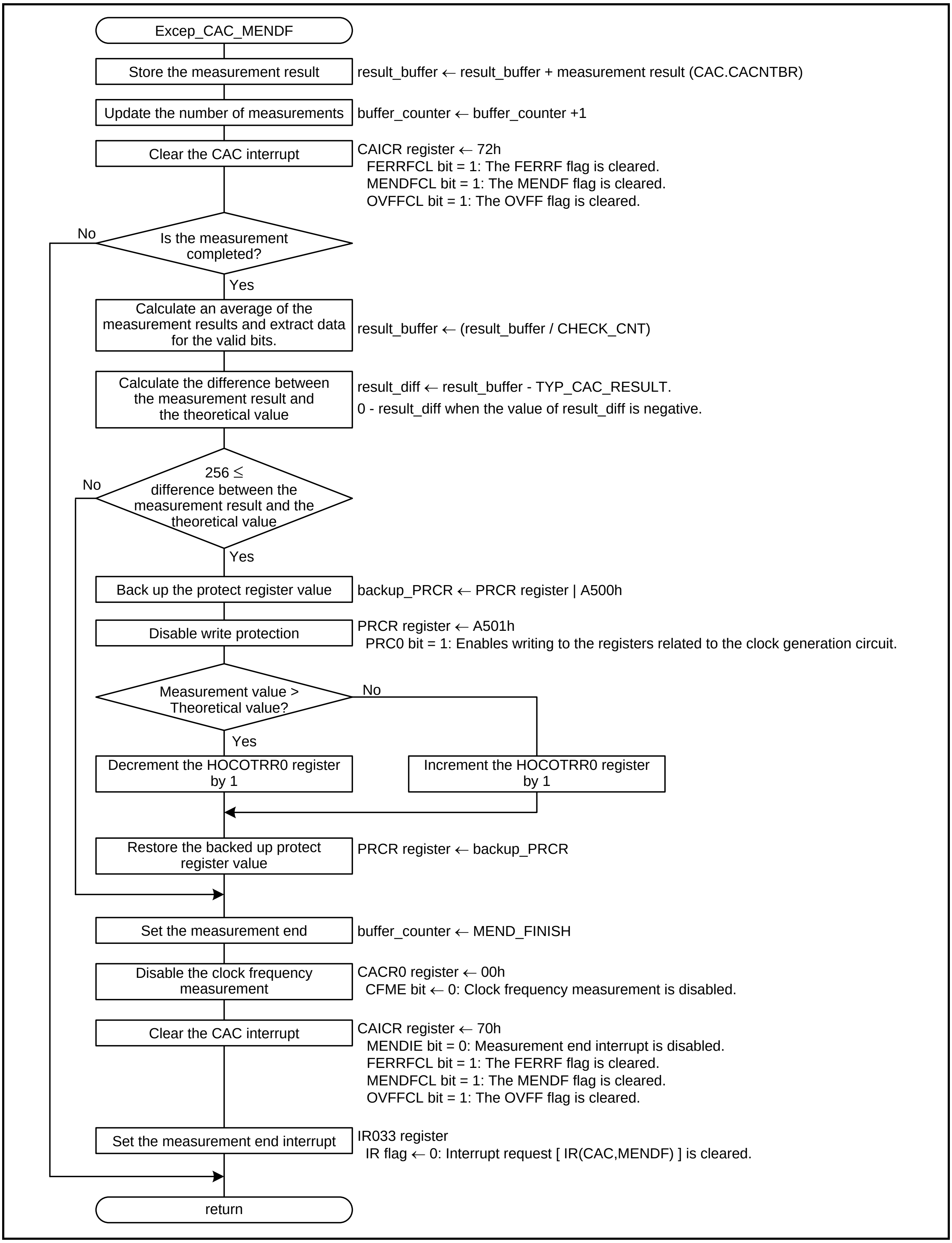


Figure 5.10 CAC Measurement End Interrupt Handling

6. Applying This Application Note to the RX21A or RX220 Group

The sample code accompanying this application note has been confirmed to operate with the RX210 Group. To make the sample code operate with the RX21A or RX220 Group, use this application note in conjunction with the Initial Setting application note for each group.

For details on using this application note with the RX21A and RX220 Groups, refer to “5. Applying the RX210 Group Application Note to the RX21A Group” in the RX21A Group Initial Setting application note, and “4. Applying the RX210 Group Application Note to the RX220 Group” in the RX220 Group Initial Setting application note.

Note: • Files r_init_clock.h and r_init_clock.c will be overwritten when applying the RX21A or RX220 Group Initial Setting. Make the settings in the overwritten files be same as the original settings in files r_init_clock.h and r_init_clock.c accompanying this application note.

7. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

8. Reference Documents

User’s Manual: Hardware

RX210 Group User’s Manual: Hardware Rev.1.50 (R01UH0037EJ)
RX21A Group User’s Manual: Hardware Rev.1.00 (R01UH0251EJ)
RX220A Group User’s Manual: Hardware Rev.1.10 (R01UH0292EJ)
The latest versions can be downloaded from the Renesas Electronics website.

Technical Update/Technical News

The latest information can be downloaded from the Renesas Electronics website.

User’s Manual: Development Tools

RX Family C/C++ Compiler Package V.1.01 User’s Manual Rev.1.00 (R20UT0570EJ)
The latest version can be downloaded from the Renesas Electronics website.

Website and Support

Renesas Electronics website

http://www.renesas.com

Inquiries

http://www.renesas.com/contact/

REVISION HISTORY	RX210, RX21A, and RX220 Groups Application Note HOCO Calibration Using the CAC

Rev.	**Date**	**Description**	
		Page	**Summary**
1.00	Oct. 10, 2014	—	First edition issued

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Handling of Unused Pins

   Handle unused pins in accordance with the directions given under Handling of Unused Pins in the manual.

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

2. Processing at Power-on

   The state of the product is undefined at the moment when power is supplied.

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

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.

   — When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

5. Differences between Products

   Before changing from one product to another, i.e. to a product with a different part number, confirm that the change will not lead to problems.

   — The characteristics of an MPU or MCU in the same group but having a different part number may differ in terms of the internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100191, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, P. R. China 200333
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Ku, Seoul, 135-920, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 4.0