# RL78/G14

## Operable Long Timer in STOP Mode (ELC, RTC and Timer RJ are used)

## CC-RL

### Introduction

In this application note, creates an operable long timer in STOP mode by using the event link controller (ELC), Real-time Clock (RTC), and Timer RJ.

### Target Device

RL78/G14

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

Contents

1. Specifications ..... 3
2. Operation Confirmation Conditions ..... 4
3. Reference Application Note ..... 4
4. Hardware ..... 5
4.1 Hardware Configuration ..... 5
4.2 Pin Used ..... 5
5. Software ..... 6
5.1 Operation Overview ..... 7
5.2 Option Byte Settings ..... 11
5.3 Functions ..... 11
5.4 Function Specifications ..... 12
5.5 Flowcharts ..... 15
5.5.1 Overall Flowchart ..... 15
5.5.2 Initialization ..... 15
5.5.3 Peripheral Function Initialization ..... 16
5.5.4 CPU Clock Initialization ..... 17
5.5.5 Port Initialization ..... 18
5.5.6 RTC Initialization ..... 19
5.5.7 Timer RJ Initialization ..... 23
5.5.8 ELC Initialization ..... 28
5.5.9 Main Processing ..... 29
5.5.10 Main Initialization ..... 30
5.5.11 Timer RJ0 Operation Start ..... 31
5.5.12 RTC Wait Operation before Shifting to STOP Mode ..... 33
6. Sample Code ..... 35
7. Reference Documents ..... 35

# 1. Specifications

In this application note, creates an operable long timer in STOP mode by using the event link controller (ELC), Real-time Clock (RTC), and Timer RJ.

Sets the RTC fixed-cycle signal (fixed-cycle interrupt) as the event source, and the count source of timer RJ0 as the event destination. Generates fixed-cycle interrupt for every second in RTC and counts down the timer RJ for every second by ELC. By setting the counted value of the timer RJ as 5, generates the underflow interruption of the timer RJ every 5 seconds, starts CPU, and performs the toggle of the port output (P130) by software.

Table 1.1 lists the peripheral functions and their applications. Figure 1.1 shows the relationship among peripheral functions.

**Table 1.1 Peripheral Functions and Their Applications**

| Peripheral Function | Application |
|---|---|
| RTC | Event generation by the RTC fixed-cycle interrupt function (1 second) |
| ELC | Event source: RTC fixed-cycle signal<br>Event destination: Count source of timer RJ (Event input from ELC) |
| Timer RJ | Period counting of port output |
| P130 | Port output |

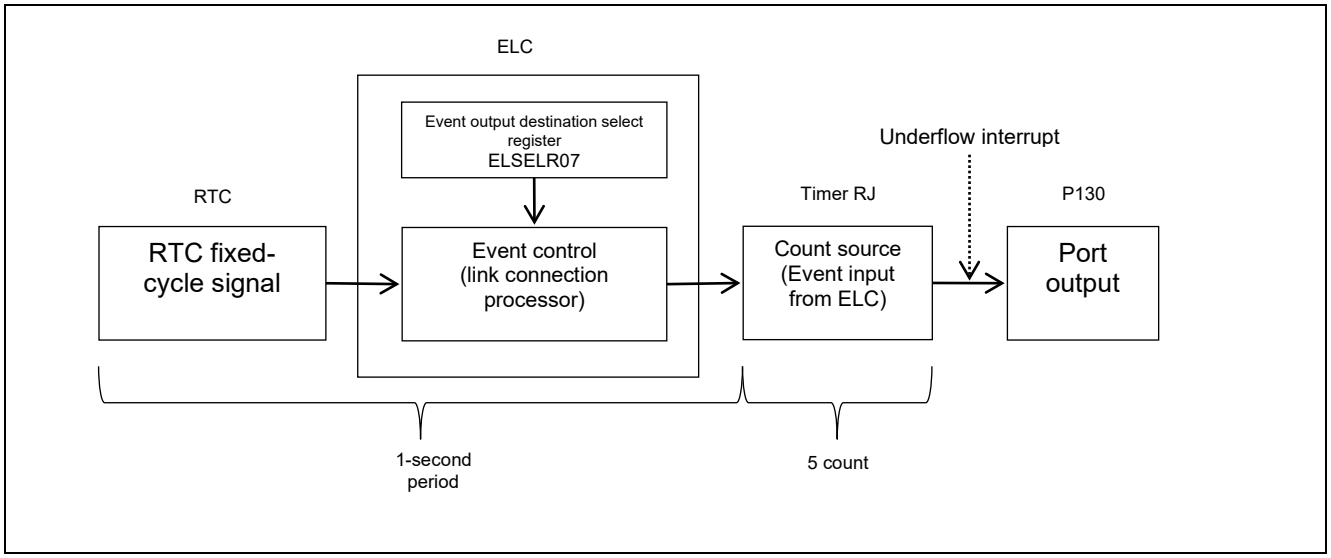


**Figure 1.1 Operation Overview**

## 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 | RL78/G14 (R5F104PJAFB) |
| Operating frequencies | High-speed on-chip oscillator clock: ($f_{HOCO}$): 8MHz<br>CPU/peripheral hardware clock: ($f_{CLK}$): 8MHz<br>RTC operation clock ($f_{sub}$): 32.768kHz (typical) |
| Operating voltage | 5.0V (Operation is possible over a voltage range of 2.9 to 5.5 V.)<br>LVD operation ($V_{LVI}$): Reset mode ( rising edge 2.81 V/falling edge 2.75 V) |
| Integrated development environment (CS+) | Renesas Electronics Corporation<br>CS+ V3.01.00 |
| C compiler (CS+) | Renesas Electronics Corporation<br>CC-RL V1.01.00 |
| Integrated development environment ($e^2$ studio) | Renesas Electronics Corporation<br>$e^2$ studio V4.0.0.26 |
| C compiler ($e^2$ studio) | Renesas Electronics Corporation<br>CC-RL V1.01.00 |
| Integrated development environment (IAR) | IAR Systems<br>IAR Embedded Workbench for Renesas RL78 V4.21.3 |
| C compiler (IAR) | IAR Systems<br>IAR C/C++ Compiler for Renesas RL78 V4.21.3.2447 |
| Board to be used | RL78/G14 CPU board (QB-R5F104PJ-TB) |

## 3. Reference Application Note

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

- RL78/G13 Initialization (R01AN2575E) Application note
- RL78/G14 How to Use the ELC for the RL78/G14 (R01AN0862E) Application note
- RL78/G14 Pulse Output Forced Cutoff Using the Clock Alarm Function and ELC (R01AN2782E) Application note

# 4. Hardware

## 4.1 Hardware Configuration

Figure 4.1 shows the Hardware Configuration used in this document.

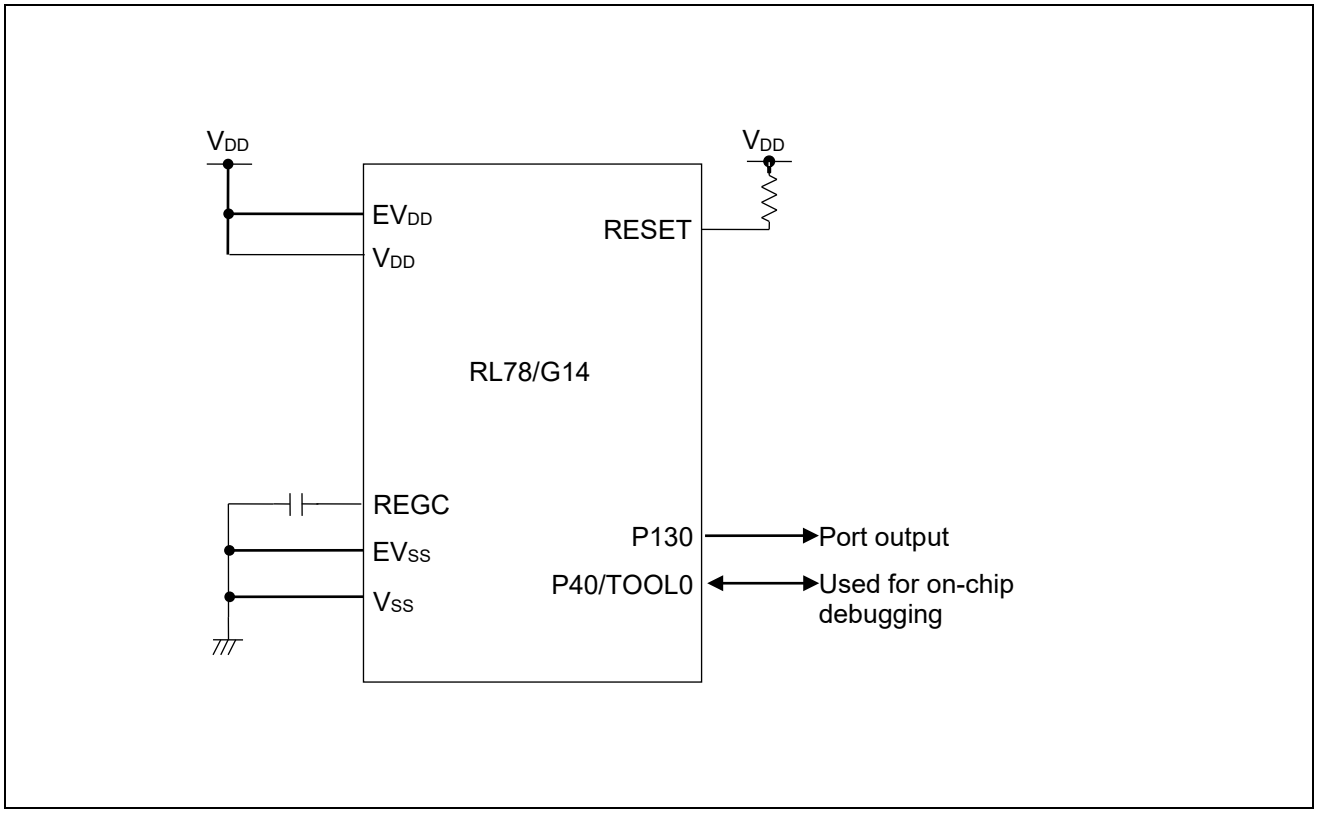


**Figure 4.1 Hardware Configuration**

Notes: 1. The above figure is simplified to show an overview of the hardware connection. When designing application circuits, make sure to handle unused pins appropriately to satisfy the electrical characteristics. (Connect input-only ports independently to $V_{DD}$ or $V_{SS}$ via resistors.)

2. Connect pins with names that begin with $EV_{SS}$ to $V_{SS}$, and pins with names that begin with $EV_{DD}$ to $V_{DD}$.

3. Make sure to set $V_{DD}$ greater than the detection voltage ($V_{LVI}$) specified by the LVD.

## 4.2 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 |
|---|---|---|
| P130 | Output | Port output |

## 5. Software

In this sample code, the code generation function of compiler is used. And CS+ version or e2studio version changes the property for code generation in order to edit generated function. Since the mode of code generation is set up for " Do nothing if a file exists " as follows, even if it performs code generation, the file which already exists in a project is not updated. When the mode is set as “merge a file” or “overwrites a file” and code generation is performed, please note that the file which exists in a project is updated but this sample code stops operating normally.

Figure 5.1 show the property setting screen for code generation.

- CS+

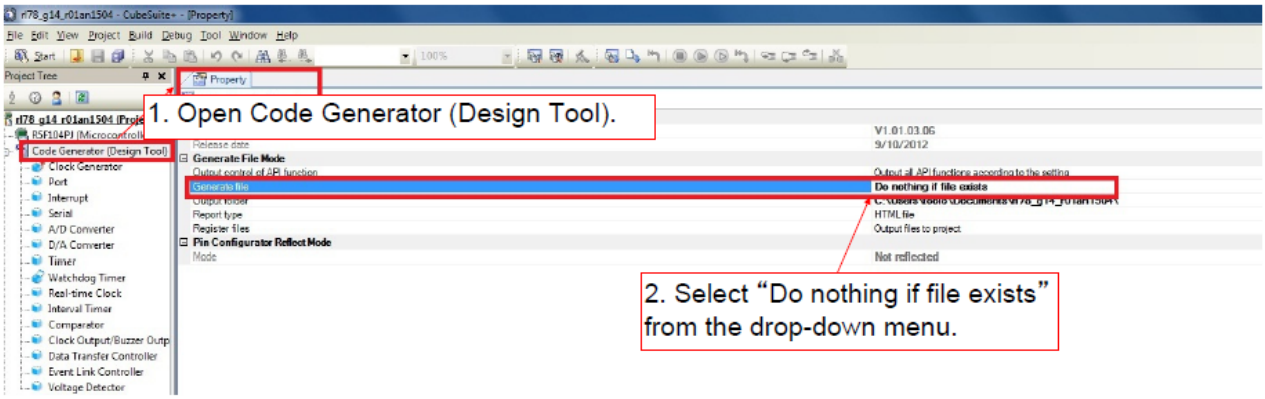


**Figure 5.1 Property Setting Screen for Code Generation (CS+)**

## 5.1 Operation Overview

In this application note, creates an operable long timer in STOP mode by using the event link controller (ELC), Real-time Clock (RTC), and Timer RJ.

Sets the RTC fixed-cycle signal (fixed-cycle interrupt) as the event source, and the count source of timer RJ0 as the event destination. Generates fixed-cycle interrupt for every second in RTC and counts down the timer RJ for every second by ELC. By setting the counted value of the timer RJ as 5, generates the underflow interruption of the timer RJ every 5 seconds, starts CPU, and performs the toggle of the port output (P130) by software.

Details are indicated to following (1) to (9).

**(1) Initializes ports.**

<Setting conditions >

- Sets P75 and P76 to High output, and turns off LED1 and LED2.
- Sets P130 to Low output.

**(2) Initializes RTC.**

< Setting conditions >

- Selects the subsystem clock ($f_{SUB}$) as the RTC operating clock.
- Sets the selection of fixed-cycle interruption (INTRTC) at a time (simultaneous with second count-up) every second.

**(3) Initializes ELC.**

<Setting conditions >

- Sets RTC fixed cycle signal as the event source.
- Sets the count source of timer RJ0 as the event destination.

When an ELC program is automatically generated using Applilet3, a build error occurs when declaring variables in the R_ELC_Stop function. Please add __no_bit_access to the variable declaration to prevent build errors.

```
void R_ELC_Stop(uint32_t  event)
{
        Volatile uint32_t w_count;
        Volatile uint8_t __no_bit_access * sfr_addr;

        sfr_addr = &ELSELR00;
```

addition

**(4) Initializes timer RJ.**

<Setting conditions >

- Sets the timer mode as the operating mode.
- Sets the event input from ELC as the count source setting.
- Counted value is set as five counts.

**(5) Initializes the main processing.**

<Setting conditions >

- Starts the count of the timer RJ.
  - Sets “0” (clear of interrupt request flag) to TRJIF0 bit of IF1H register.
  - Sets “0” (interrupt servicing enabled) to TRJMK0 bit of MK1H register.
  - Sets “1” (count starts) to TSTART bit of TRJCR0 register.

- Tarts the count of RTC.
  - Sets “1” (Starts counter operation) to RTCE bit of RTCC0 register.

- Carries out Wait before STOP mode shift.
  - Sets “1” (stops SEC to YEAR counters. Mode to read or write counter value) to RWAIT bit of RTCC1 register.
  - Sets “1” (mode to read or write counter value) to RWST bit of RTCC1 register.
  - Sets “0” (sets counter operation) to RWAIT bit of RTCC1 register.
  - Waits for that the RWST bit of RTCC1 register becomes “0” (in counter operation).

**(6) Shifts to the STOP mode.**

**(7) Returns from STOP mode after the underflow interrupt of the timer RJ occurring and reverses the output of P130.**

**(8) Initializes the timer RJ flag.**

- Sets “0” (no underflow) to TUNDF bit of TRJCR0 register.

**(9) After this, repeats from (6) to (8).**

Figure 5. shows the timing chart.

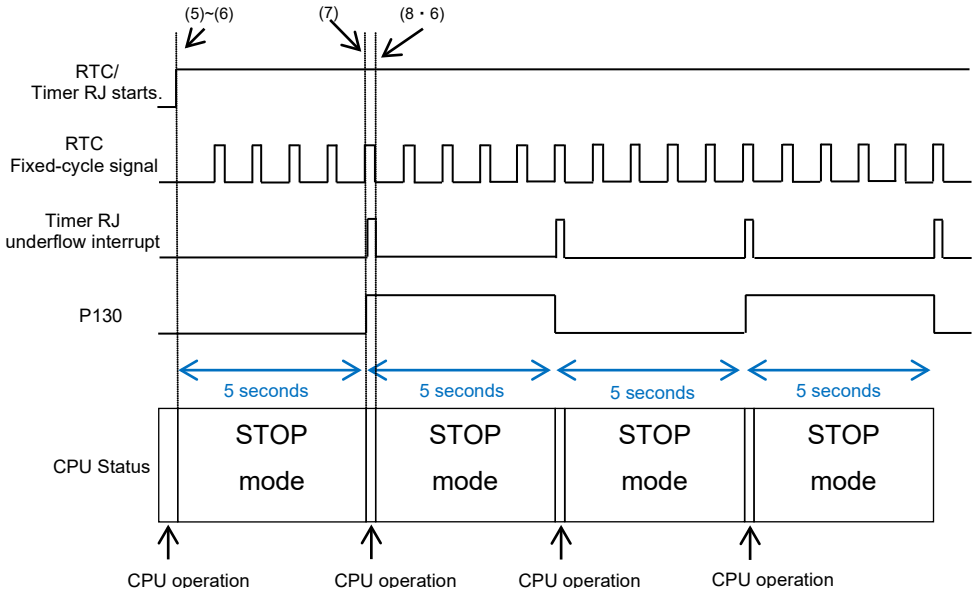


**Figure 5.2 Timing Chart**

## 5.2 Option Byte Settings

Table 5.1 lists the option byte settings.

**Table 5.1 Option Byte Settings**

| Address | Setting Value | Contents |
|---|---|---|
| 000C0H/010C0H | 11101111B | Stops the watchdog timer<br>(counting is stopped when a reset is canceled) |
| 000C1H/010C1H | 01111111B | Sets the LVD in reset mode<br>Detection voltage: 2.81 V at the rising edge, 2.75 V at the falling edge |
| 000C2H/010C2H | 10101010B | LS mode, HOCO clock: 8MHz |
| 000C3H/010C3H | 10000100B | Enables on-chip debugging<br>Data of flash memory is erased or not erased in case of failure in on-chip debug security ID authentication. |

## 5.3 Functions

Table 5.2 lists the functions.

**Table 5.2 Functions**

| Function Name | Outline |
|---|---|
| hdwinit | Initialization |
| R_Systeminit | Peripheral function initialization |
| R_CGC_Create | CPU clock initialization |
| R_PORT_Create | Port initialization |
| R_RTC_Create | RTC initialization |
| R_TMR_RJ0_Create | Timer RJ0 initialization |
| R_ELC_Create | ELC initialization |
| main | Main processing |
| R_MAIN_UserInit | Main initialization |
| R_TMR_RJ0_Start | Timer RJ0 operation start |
| R_RTC_Wait_For_Stop | Wait befog RTC STOP mode shift |

## 5.4 Function Specifications

The following tables list the sample code function specifications.

[Function Name] hdwinit

| | |
|---|---|
| Outline | Initialization |
| Header | None |
| Declaration | void hdwinit(void) |
| Description | Initializes peripheral functions. |
| Arguments | None |
| Return Value | None |
| Remarks | None |

[Function Name] R_Systeminit

| | |
|---|---|
| Outline | Peripheral function initialization |
| Header | None |
| Declaration | void R_Systeminit(void) |
| Description | Initializes peripheral functions to be used in this application note. |
| Arguments | None |
| Return Value | None |
| Remarks | None |

[Function Name] R_CGC_Create

| | |
|---|---|
| Outline | CPU clock initialization |
| Header | r_cg_cgc.h |
| Declaration | void R_CGC_Create(void) |
| Description | Initializes CPU clocks. |
| Arguments | None |
| Return Value | None |
| Remarks | None |

[Function Name] R_PORT_Create

| | |
|---|---|
| Outline | Port initialization |
| Header | r_cg_port.h |
| Declaration | void R_PORT_Create(void) |
| Description | Initializes ports. |
| Arguments | None |
| Return Value | None |
| Remarks | None |

[Function Name] R_RTC_Create

| | |
|---|---|
| Outline | RTC initialization |
| Header | r_cg_rtc.h |
| Declaration | void R_RTC_Create(void) |
| Description | Initializes RTC. |
| Arguments | None |
| Return Value | None |
| Remarks | None |

[Function Name] R_TMR_RJ0_Create

| | |
|---|---|
| Outline | Timer RJ initialization |
| Header | r_cg_timer.h |
| Declaration | void R_TMR_RJ0_Create(void) |
| Description | Initializes Timer RJ. |
| Arguments | None |
| Return Value | None |
| Remarks | None |

[Function Name] R_ELC_Create

| | |
|---|---|
| Outline | ELC initialization |
| Header | r_cg_elc.h |
| Declaration | void R_ELC_Create(void) |
| Description | Initializes ELC. |
| Arguments | None |
| Return Value | None |
| Remarks | None |

[Function Name] main

| | |
|---|---|
| Outline | Main processing |
| Header | None |
| Declaration | void main(void) |
| Description | Performs the main processing. |
| Arguments | None |
| Return Value | None |
| Remarks | None |

[Function Name] R_MAIN_UserInit

| | |
|---|---|
| Outline | Main initialization |
| Header | None |
| Declaration | void R_MAIN_UserInit(void) |
| Description | Performs required processing for the initialization of main processing. |
| Arguments | None |
| Return Value | None |
| Remarks | None |

[Function Name] R_TMR_RJ0_Start

| | |
|---|---|
| Outline | Timer RJ operation start |
| Header | r_cg_timer.h |
| Declaration | void R_TMR_RJ0_Start(void) |
| Description | Sets Timer RJ as “operation is enabled”. |
| Arguments | None |
| Return Value | None |
| Remarks | None |

[Function Name] R_RTC_Wait_For_Stop

| | |
|---|---|
| Outline | Wait befog RTC STOP mode shift |
| Header | r_cg_rtc.h |
| Declaration | void R_RTC_Wait_For_Stop(void) |
| Description | Performs wait processing before RTC STOP mode shift. |
| Arguments | None |
| Return Value | None |
| Remarks | None |

## 5.5 Flowcharts

### 5.5.1 Overall Flowchart

Figure 5. shows the overall flow.

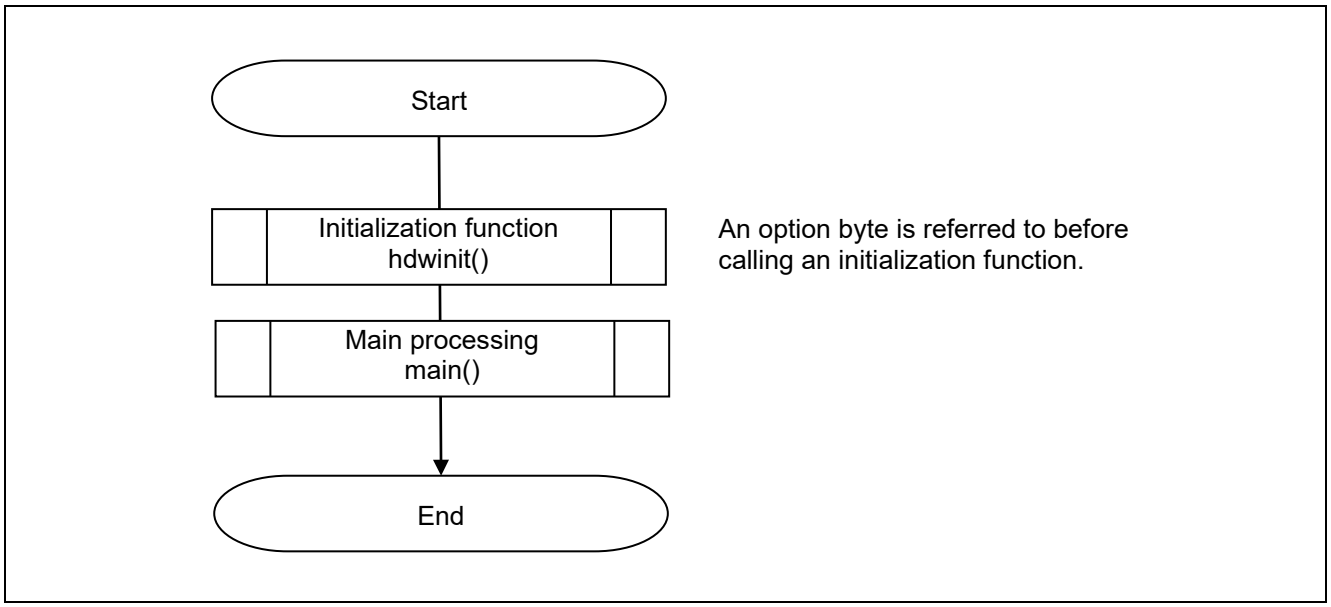


**Figure 5.3 Overall Flow**

### 5.5.2 Initialization

Figure 5. shows the initialization.

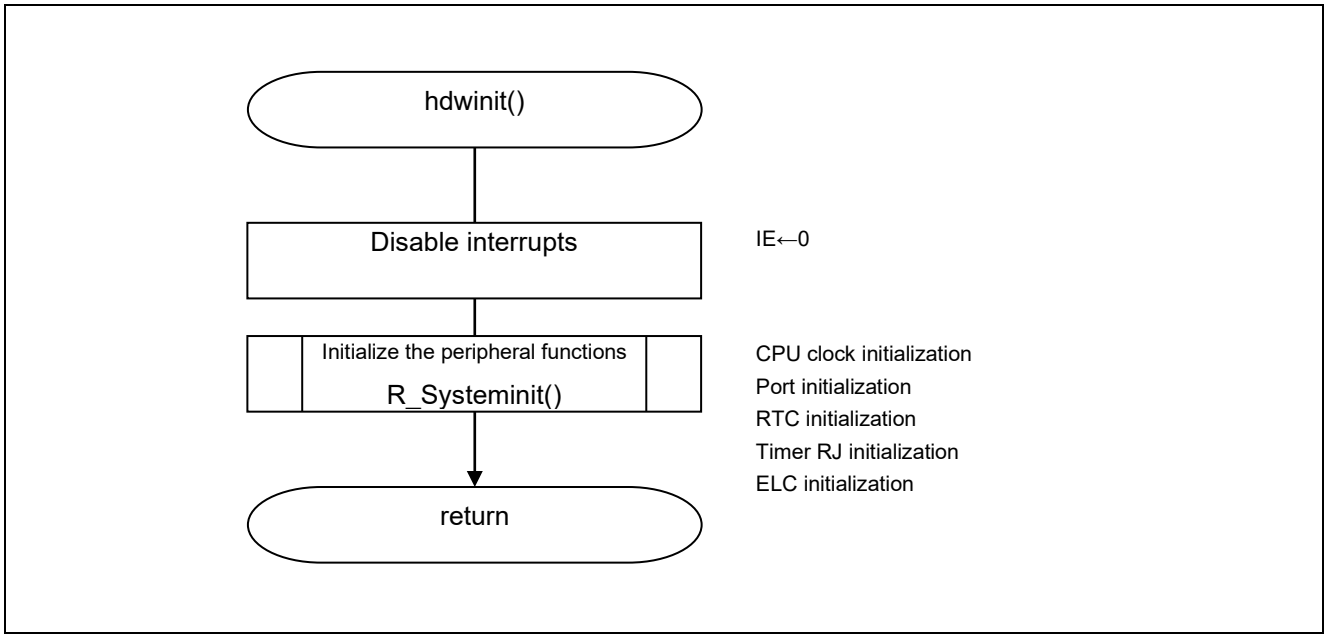


**Figure 5.4 Initialization**

### 5.5.3 Peripheral Function Initialization

Figure 5. shows the peripheral function initialization.

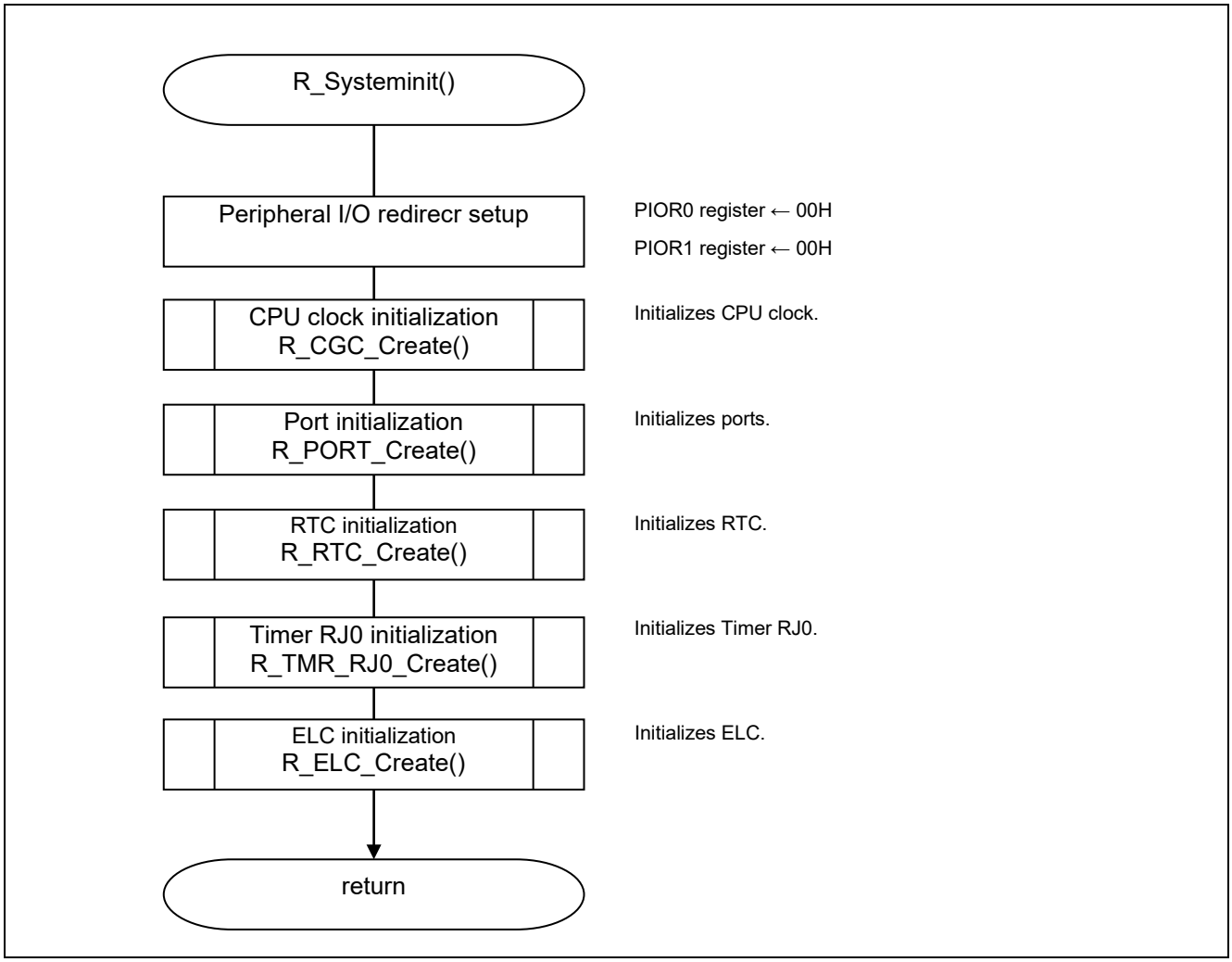


**Figure 5.5 Peripheral Function Initialization**

### 5.5.4 CPU Clock Initialization

Figure 5. shows the CPU clock initialization.

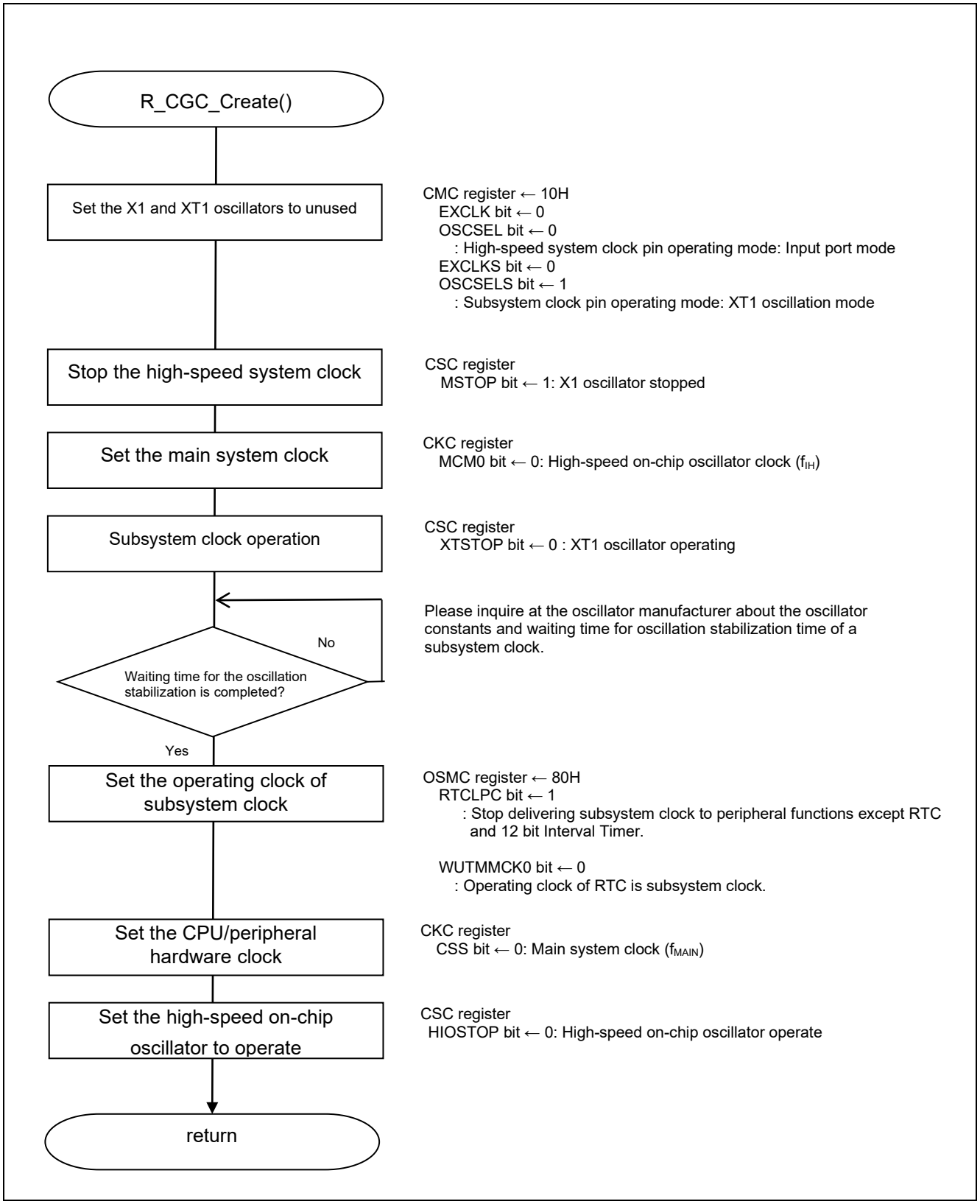


**Figure 5.6 CPU Clock Initialization**

### 5.5.5 Port Initialization

Figure 5. shows the port initialization.

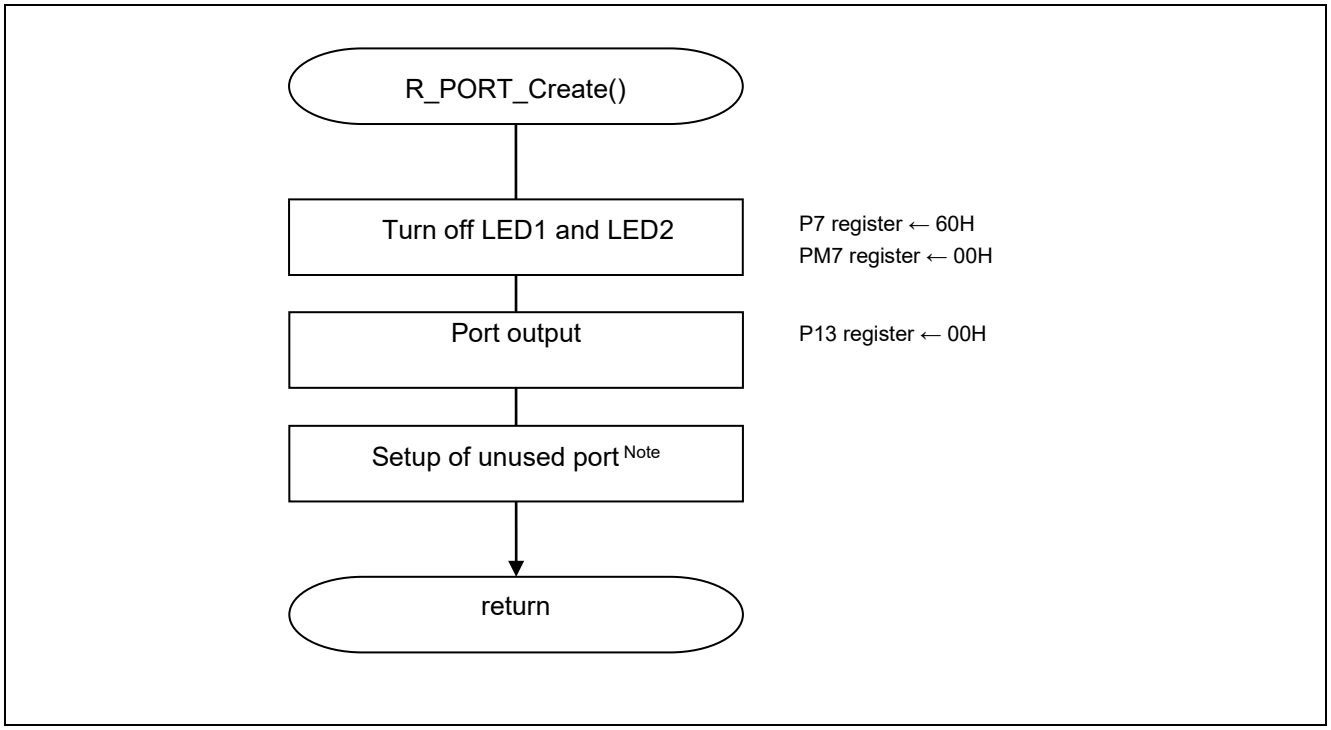


**Figure 5.7 Port Initialization**

Note Refer to RL78/G13 Initialization (R01AN2575E) Application Note “Flowchart” for unused port setup.

Cautions Provide proper treatment for unused pins so that their electrical specifications are observed.
Connect each of any unused input-only ports to $V_{DD}$ or $V_{SS}$ via a resistor.

### 5.5.6 RTC Initialization

Figure 5. shows the RTC initialization.

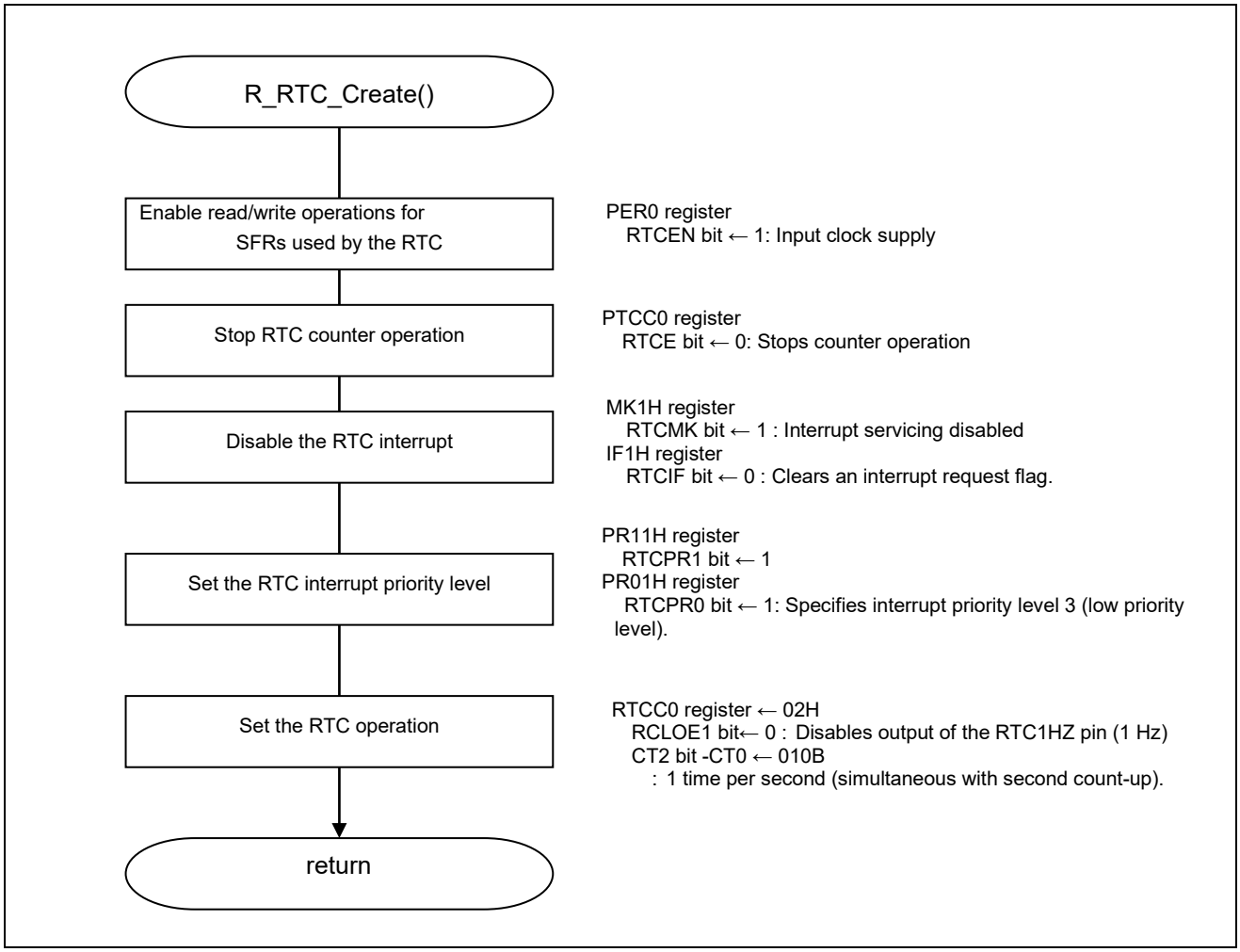


**Figure 5.8 RTC Initialization**

**Enabling read and write operations for SFRs used by the RTC**

・Peripheral enable register 0 (PER0)
Starts to supply clock to RTC.

Symbol：PER0

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RTCEN | IICA1EN | ADCEN | IICA0EN | SAU1EN | SAU0EN | TAU1EN | TAU0EN |
| **1** | x | x | x | x | x | x | x |

Bit 7

| RTCEN | Control of real-time clock (RTC) and 12-bit interval timer input clock supply |
|---|---|
| 0 | Stops input clock supply.<br>・SFR used by the real-time clock (RTC) cannot be written.<br>・The real-time clock (RTC) is in the reset status. |
| **1** | **Input clock supply.**<br>**・SFR used by the real-time clock (RTC) can be read/written.** |

### Stopping RTC counter operation

・Real-time clock control register 0 (RTCC0)
Stops counter operation.

Symbol：RTCC0

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RTCE | 0 | RCLOE1 | 0 | AMPM | CT2 | CT1 | CT0 |
| **0** | — | | — | | | | |

Bit 7

| RTCE | Real-time clock operation control |
|---|---|
| **0** | **Stops counter operation.** |
| 1 | Starts counter operation. |

### Disabling the RTC interrupt

・Interrupt Mask Flag Register (MK1H)
Interrupt servicing disabled.
・Interrupt Request Flag Register (IF1H)
Interrupt request flag clear.

Symbol：MK1H

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| TMMK10 | TRJMK0 | SRMK3<br>CSIMK31<br>IICMK31 | STMK3<br>CSIMK30<br>IICMK30 | KRMK | ITMK | RTCMK | ADMK |
| x | x | x | x | x | x | **1** | x |

Bit 1

| RTCMK | Interrupt servicing control |
|---|---|
| 0 | Interrupt servicing enabled |
| **1** | **Interrupt servicing disabled** |

Symbol：IF1H

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| TMIF10 | TRJIF0 | SRIF3<br>CSIIF31<br>IICIF31 | STIF3<br>CSIIF30<br>IICIF30 | KRIF | ITIF | RTCIF | ADIF |
| x | x | x | x | x | x | **0** | x |

Bit 1

| RTCIF | Interrupt request flag |
|---|---|
| **0** | **No interrupt request signal is generated** |
| 1 | Interrupt request is generated, interrupt request status |

### Setting the RTC interrupt priority level

・Priority Specification Flag Register (PR11H、PR01H)
Specifies level 3 (low priority level)

Symbol：PR11H

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| TMPR110 | TRJPR10 | SRPR13<br>CSIPR131<br>IICPR131 | STPR13<br>CSIPR130<br>IICPR130 | KRPR1 | ITPR1 | RTCPR1 | ADPR1 |
| x | x | x | x | x | x | **1** | x |

Symbol：PR01H

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| TMPR010 | TRJPR00 | SRPR03<br>CSIPR031<br>IICPR031 | STPR03<br>CSIPR030<br>IICPR030 | KRPR0 | ITPR0 | RTCPR0 | ADPR0 |
| x | x | x | x | x | x | **1** | x |

Bit 1

| RTCPR1 | RTCPR0 | Priority level selection |
|---|---|---|
| 0 | 0 | Specify level 0 (high priority level) |
| 0 | 1 | Specify level 1 |
| 1 | 0 | Specify level 2 |
| **1** | **1** | **Specify level 3 (low priority level)** |

### Setting the RTC operation

・Real-time clock control register 0 (RTCC0)
Output signals from the RTC1HZ pin: Stops disabling.
Fixed-cycle interrupt function: 1 time per second (simultaneous with second count-up).

Symbol：RTCC0

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RTCE | 0 | RCLOE1 | 0 | AMPM | CT2 | CT1 | CT0 |
| | — | **0** | — | 0 | **0** | **1** | **0** |

Bit 5

| RCLOE1 | RTC1HZ pin output control |
|---|---|
| **0** | **Disables output of the RTC1HZ pin (1 Hz).** |
| 1 | Enables output of the RTC1HZ pin (1 Hz). |

Bit 2 to 0

| CT2 | CT1 | CT0 | Constant-period interrupt (INTRTC) selection |
|---|---|---|---|
| 0 | 0 | 0 | Does not use fixed-cycle interrupt function. |
| 0 | 0 | 1 | Once per 0.5 s (synchronized with second count up) |
| **0** | **1** | **0** | **Once per 1 s (same time as second count up)** |
| 0 | 1 | 1 | Once per 1 m (second 00 of every minute) |
| 1 | 0 | 0 | Once per 1 hour (minute 00 and second 00 of every hour) |
| 1 | 0 | 1 | Once per 1 day (hour 00, minute 00, and second 00 of every day) |
| 1 | 1 | x | Once per 1 month (Day 1, hour 00 a.m., minute 00, and second 00 of every month) |

### 5.5.7 Timer RJ Initialization

Figure 5. shows Timer RJ initialization.

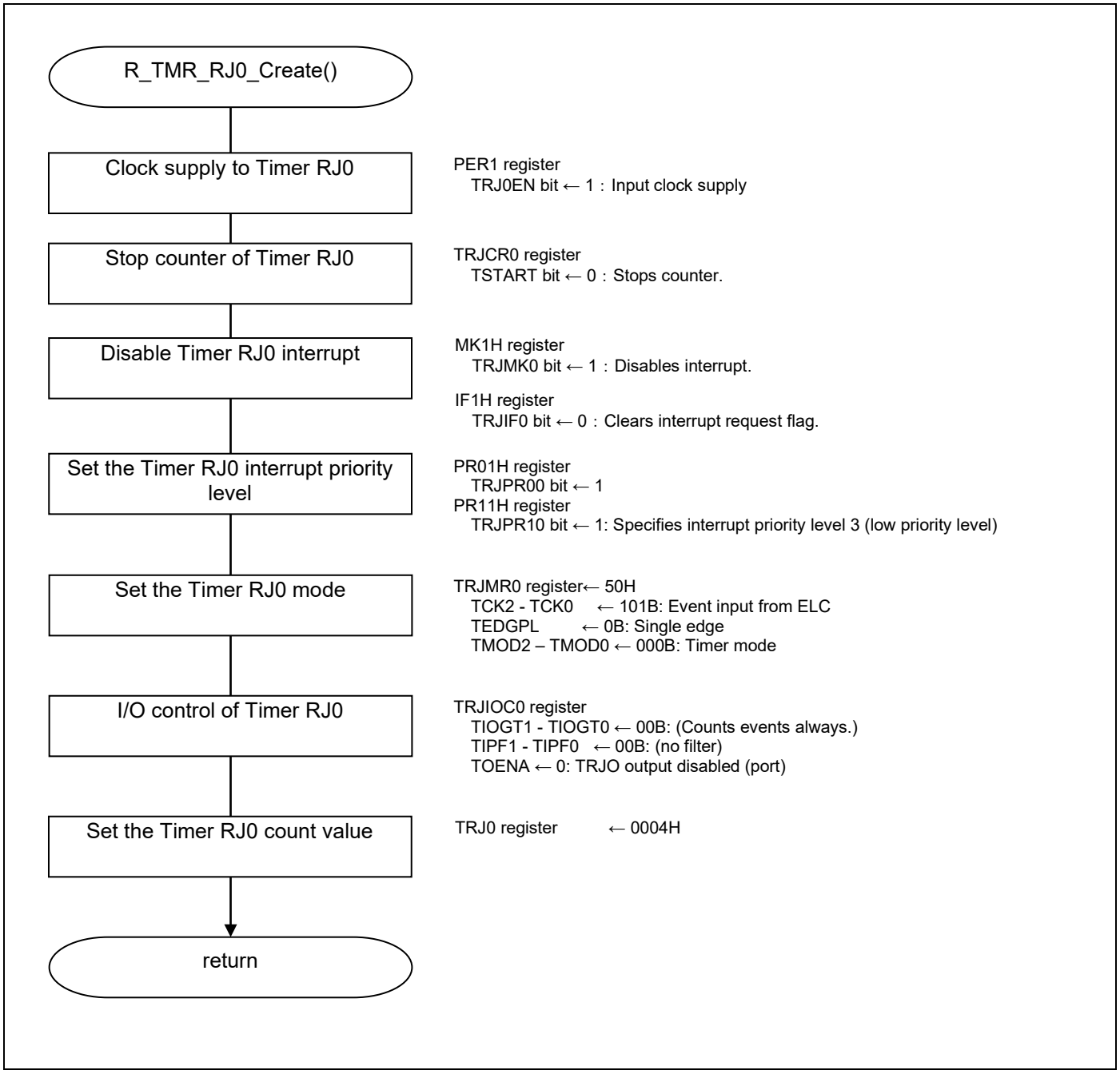


**Figure 5.9 Timer RJ Initialization**

### Timer RJ clock supply

・Peripheral enable register 1 (PER1)
Starts clock supply to Timer RJ0.

Symbol：PER1

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| DACEN | TRGEN | CMPEN | TRD0EN | DTCEN | 0 | 0 | TRJ0EN |
| x | x | x | x | x | x | x | **1** |

Bit 0

| TRJ0EN | Control of timer RJ0 input clock supply |
|---|---|
| 0 | Stops input clock supply. |
| **1** | **Enables input clock supply.** |

### Stop of Timer RJ0 counter

・Timer RJ control register 0 (TRJCR0)
Sets the stop of Timer RJ0 count.

Symbol：TRJCR0

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | TUNDF | TEDGF | 0 | TSTOP | TCSIF | TSTART |
| — | — | 0 | 0 | — | 0 | 0 | **0** |

Bit 0

| TSTART | Timer RJ count start |
|---|---|
| **0** | **Count stops** |
| 1 | Count starts |

### Disable Timer RJ0 interrupt

- Interrupt Mask Flag Register (MK1H)
  Interrupt servicing disabled.
- Interrupt Request Flag Register (IF1H)
  Clears interrupt request flag.

Symbol：MK1H

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| TMMK10 | TRJMK0 | SRMK3<br>CSIMK31<br>IICMK31 | STMK3<br>CSIMK30<br>IICMK30 | KRMK | ITMK | RTCMK | ADMK |
| x | **1** | x | x | x | x | x | x |

Bit 6

| TRJMK0 | Interrupt servicing control |
|---|---|
| 0 | Interrupt servicing enabled |
| **1** | **Interrupt servicing disabled** |

Symbol：IF1H

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| TMIF10 | TRJIF0 | SRIF3<br>CSIIF31<br>IICIF31 | STIF3<br>CSIIF30<br>IICIF30 | KRIF | ITIF | RTCIF | ADIF |
| x | **0** | x | x | x | x | x | x |

Bit 6

| TRJIF0 | Interrupt request flag |
|---|---|
| **0** | **No interrupt request signal is generated** |
| 1 | Interrupt request is generated, interrupt request status |

### Set the RTC interrupt priority level

・Priority Specification Flag Register (PR11H,PR01H)
Specifies interrupt priority level 3 (low priority level).

Symbol：PR11H

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| TMPR110 | TRJPR10 | SRPR13<br>CSIPR131<br>IICPR131 | STPR13<br>CSIPR130<br>IICPR130 | KRPR1 | ITPR1 | RTCPR1 | ADPR1 |
| x | **1** | x | x | x | x | x | x |

Symbol：PR01H

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| TMPR010 | TRJPR00 | SRPR03<br>CSIPR031<br>IICPR031 | STPR03<br>CSIPR030<br>IICPR030 | KRPR0 | ITPR0 | RTCPR0 | ADPR0 |
| x | **1** | x | x | x | x | x | x |

Bit 6

| TRJPR10 | TRJPR00 | Priority level selection |
|---|---|---|
| 0 | 0 | Specify level 0 (high priority level) |
| 0 | 1 | Specify level 1 |
| 1 | 0 | Specify level 2 |
| **1** | **1** | **Specify level 3 (low priority level)** |

### Set the Timer RJ0 mode

・Timer RJ I/O control register 0 (TRJMR0)
Sets a mode as event mode from ELC.
Sets the timer mode.

Symbol：TRJMR0

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0 | TCK2 | TCK1 | TCK0 | TEDGPL | TMOD2 | TMOD1 | TMOD0 |
| x | **1** | **0** | **1** | **0** | **0** | **0** | **0** |

Bit 6 to 4

| TCK2 | TCK1 | TCK0 | Timer RJ count source select |
|---|---|---|---|
| 0 | 0 | 0 | $f_{CLK}$ |
| 0 | 0 | 1 | $f_{CLK}/8$ |
| 0 | 1 | 1 | $f_{CLK}/2$ |
| 1 | 0 | 0 | $f_{IL}$ |
| **1** | **0** | **1** | **Event input from ELC** |
| 1 | 1 | 0 | $f_{SUB}$ |
| Other than above | | | Setting prohibited |

Bit 3

| TEDGPL | TRJIO edge polarity select |
|---|---|
| **0** | **One edge** |
| 1 | Both edges |

Bit 2 to 0

| TMOD2 | TMOD1 | TMOD0 | Timer RJ operating mode select |
|---|---|---|---|
| **0** | **0** | **0** | **Timer mode** |
| 0 | 0 | 1 | Pulse output mode |
| 0 | 1 | 1 | Event counter mode |
| 1 | 0 | 0 | Pulse width measurement mode |
| 1 | 0 | 1 | Pulse period measurement mode |
| Other than above | | | Setting prohibited |

### Timer RJ0 count value setup

・Timer RJ counter register 0 (TRJ0)
Sets up the count value.

Symbol：TRJ0

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **1** | **0** | **1** |

Bit 15 to 0

| Function |
|---|
| 16-bit counter |

### 5.5.8 ELC Initialization

Figure 5.2 shows the ELC initialization.

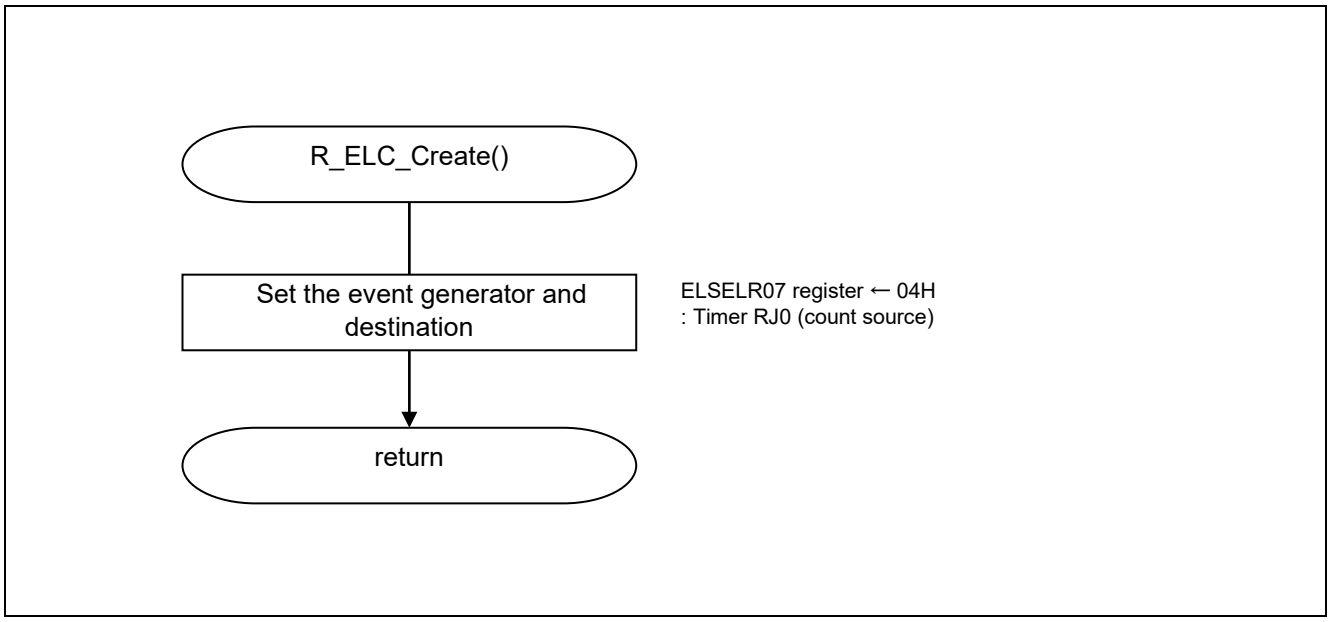


**Figure 5.2 ELC Initialization**

**Event generator and destination setup**

・Event output destination select register 07 (ELSELR07)
Event generator: RTC fixed-cycle signal/alarm match detection
Event destination: Timer RJ0 (count source)

| Register Name | Event Generator (Output Origin of Event Input 7) | Event Description |
|---|---|---|
| **ELSELR07** | **RTC fixed-cycle signal/Alarm match detection** | **INTRTC** |

Symbol：ELSELR07

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | ELSEL073 | ELSEL072 | ELSEL071 | ELSEL070 |
| — | — | | — | **0** | **1** | **0** | **0** |

Bit 3 to 0

| ELSEL073 | ELSEL070 | ELSEL071 | ELSEL070 | Event Link Selection |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | **Event link disabled** |
| 0 | 0 | 0 | 1 | Select operation of peripheral function 1 to link |
| 0 | 0 | 1 | 0 | Select operation of peripheral function 2 to link |
| 0 | 0 | 1 | 1 | Select operation of peripheral function 3 to link |
| **0** | **1** | **0** | **0** | **Link Destination Peripheral Function：Timer RJ0<br>Operation When Receiving Event：Count source** |
| 0 | 1 | 0 | 1 | Select operation of peripheral function 5 to link |
| 0 | 1 | 1 | 0 | Select operation of peripheral function 6 to link |
| 0 | 1 | 1 | 1 | Select operation of peripheral function 7 to link |
| 1 | 0 | 0 | 0 | Select operation of peripheral function 8 to link |
| 1 | 0 | 0 | 1 | Select operation of peripheral function 9 to link |
| Other than above | | | | Setting prohibited |

### 5.5.9 Main Processing

Figure 5.3 shows the main processing.

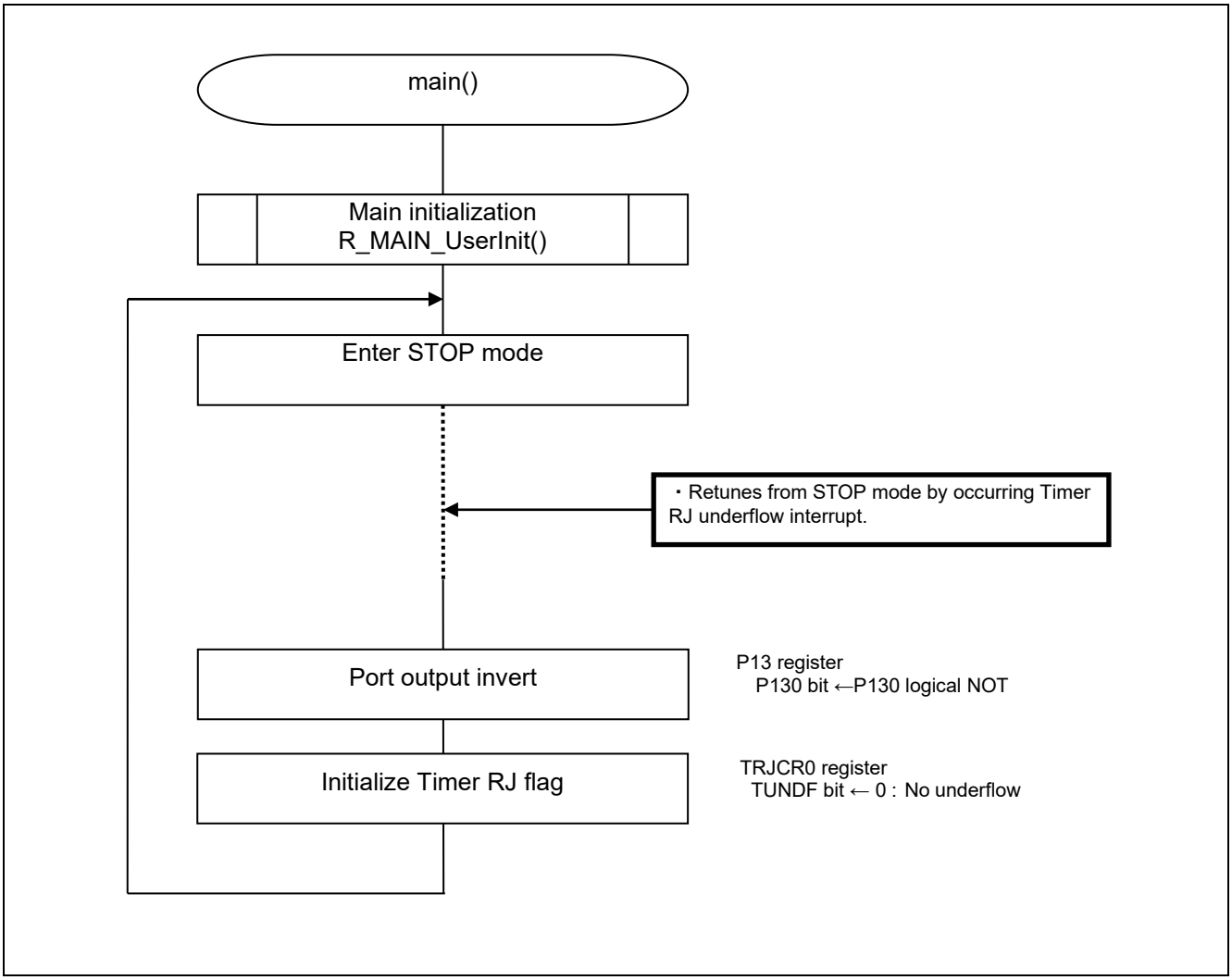


**Figure 5.3 Main Processing**

**Timer RJ flag Initialization**

・Timer RJ control register 0 (TRJCR0)
Initializes the underflow flag of Timer RJ.

Symbol：TRJCR0

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | TUNDF | TEDGF | 0 | TSTOP | TCSIF | TSTART |
| — | — | **0** | | — | | | |

Bit 5

| TUNDF | Timer RJ underflow flag |
|---|---|
| **0** | **No underflow** |
| 1 | Underflow |

### 5.5.10 Main Initialization

Figure 5.4 shows the main initialization.

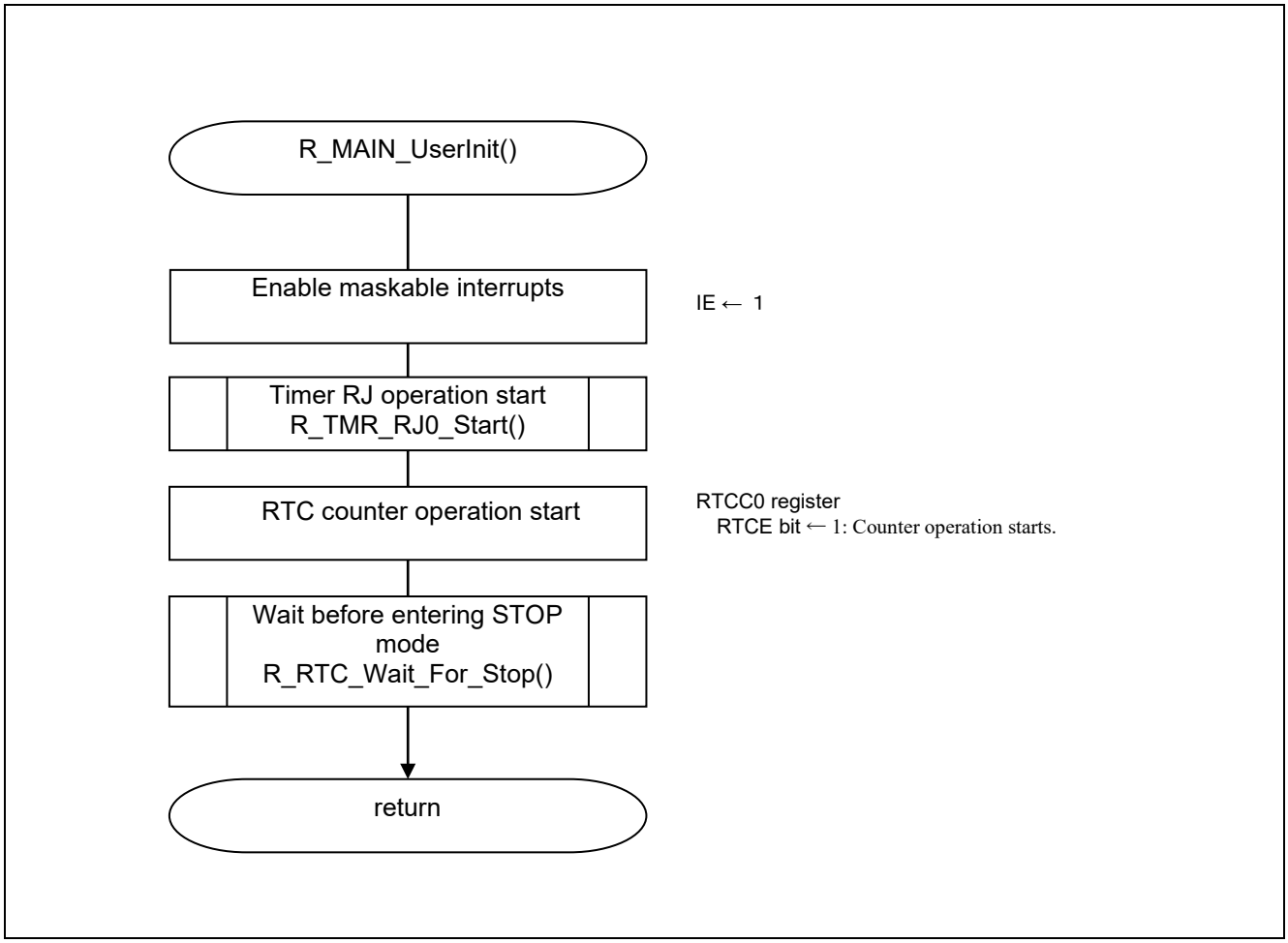


**Figure 5.4 Main Initialization**

**RTC counter operation start**

・Real-time clock control register 0 (RTCC0)
Starts counter operation.

Symbol：RTCC0

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RTCE | 0 | RCLOE1 | 0 | AMPM | CT2 | CT1 | CT0 |
| **1** | — | | — | | | | |

Bit 7

| RTCE | Real-time clock operation control |
|---|---|
| 0 | Stops counter operation. |
| **1** | **Starts counter operation.** |

### 5.5.11 Timer RJ0 Operation Start

Figure 5.5 shows Timer RJ0 operation start.

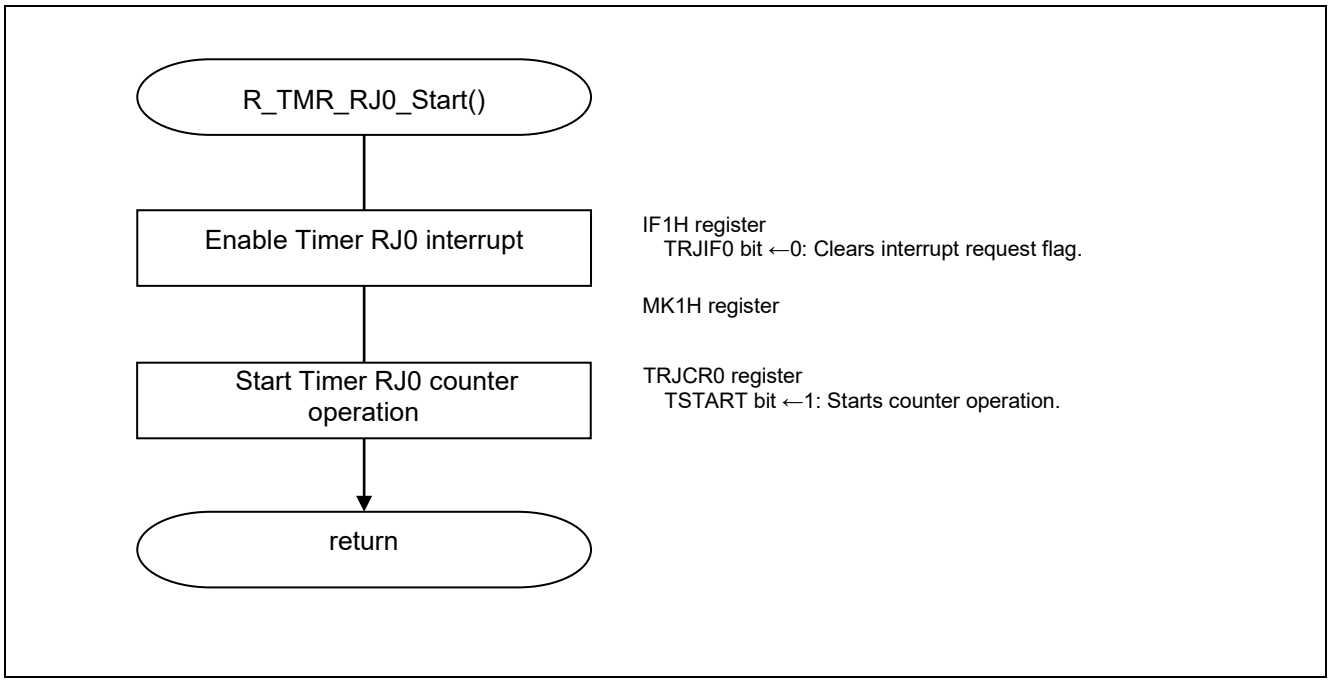


**Figure 5.5 Timer RJ0 Operation Start**

### Enable Timer RJ0 interrupt

- Interrupt Mask Flag Register (MK1H)
  Enables interrupt processing.
- Interrupt Request Flag Register (IF1H)
  Clears interrupt request flag.

Symbol：MK1H

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| TMMK10 | TRJMK0 | SRMK3<br>CSIMK31<br>IICMK31 | STMK3<br>CSIMK30<br>IICMK30 | KRMK | ITMK | RTCMK | ADMK |
| x | **0** | x | x | x | x | x | x |

Bit 6

| TRJMK0 | Interrupt servicing control |
|---|---|
| **0** | **Interrupt servicing enabled** |
| 1 | Interrupt servicing disabled |

Symbol：IF1H

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| TMIF10 | TRJIF0 | SRIF3<br>CSIIF31<br>IICIF31 | STIF3<br>CSIIF30<br>IICIF30 | KRIF | ITIF | RTCIF | ADIF |
| x | **0** | x | x | x | x | x | x |

Bit 6

| TRJIF0 | Interrupt request flag |
|---|---|
| **0** | **No interrupt request signal is generated** |
| 1 | Interrupt request is generated, interrupt request status |

### Start Timer RJ0 counter operation

- Timer RJ control register 0 (TRJCR0)
  Sets Timer RJ0 counter start.

Symbol：TRJCR0

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | TUNDF | TEDGF | 0 | TSTOP | TCSIF | TSTART |
| — | — | 0 | 0 | — | 0 | 0 | **1** |

Bit 0

| TSTART | Timer RJ count start |
|---|---|
| 0 | Count stops |
| **1** | **Count starts** |

### 5.5.12 RTC Wait Operation before Shifting to STOP Mode

Figure 5.6 shows the RTC wait operation before shifting to STOP mode.

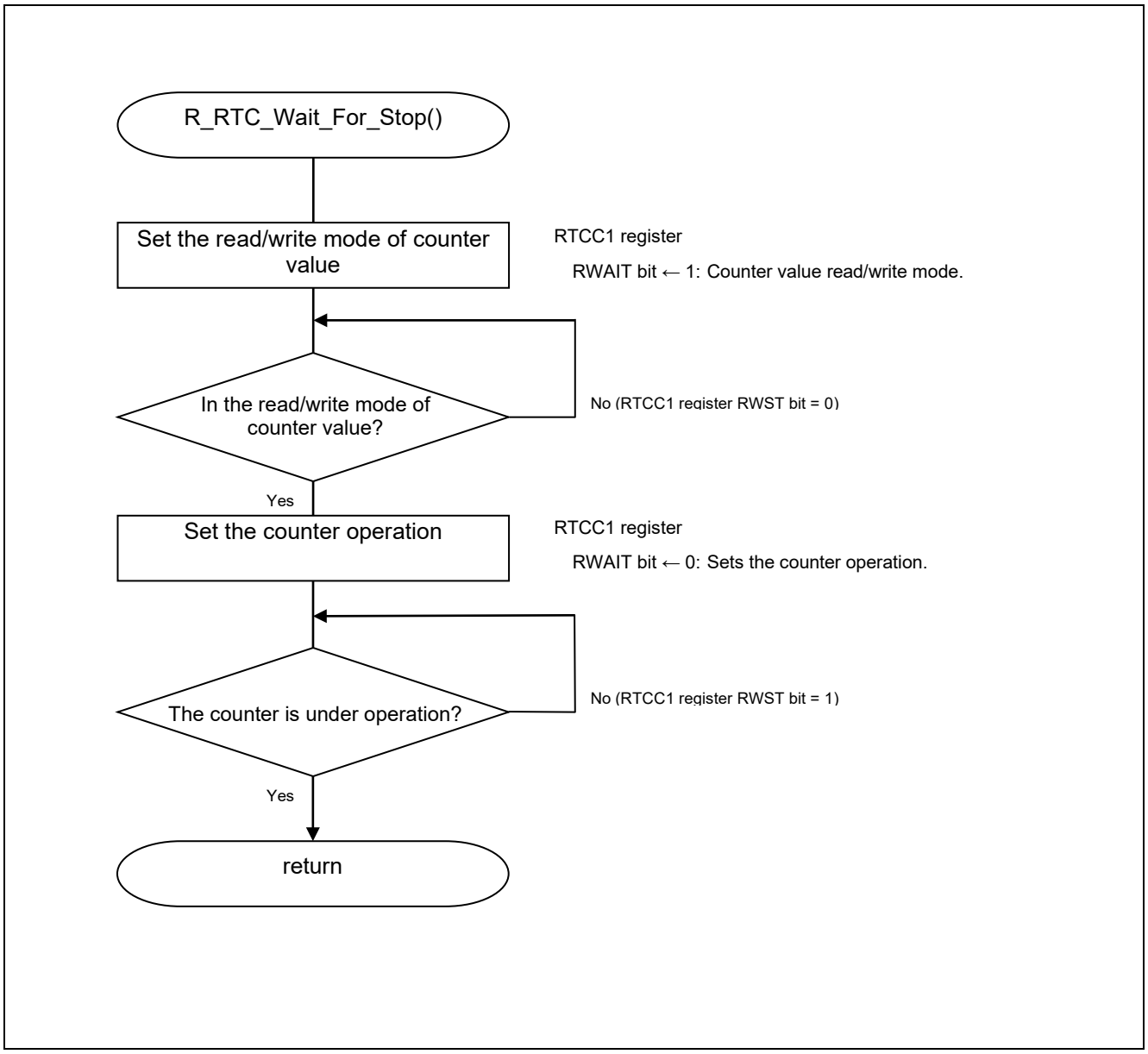


**Figure 5.6 RTC Wait Operation before Shifting to STOP Mode**

### RTC Wait Operation before Shifting to STOP Mode

・Real-time clock control register 1 (RTCC1)
Sets RTC before shifting to STOP mode.

Symbol：RTCC1

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| WALE | WALIE | 0 | WAFG | RIFG | 0 | RWST | RWAIT |
| | | — | | | — | **0** | **0** |

Bit 1

| RWST | Wait status flag of real-time clock |
|---|---|
| **0** | **Counter is operating.** |
| 1 | Mode to read or write counter value |

Bit 0

| RWAIT | Wait control of real-time clock |
|---|---|
| **0** | **Sets counter operation.** |
| 1 | Stops SEC to YEAR counters. Mode to read or write counter value |

## 6. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

## 7. Reference Documents

RL78/G14 User's Manual: Hardware (R01UH0146E)

RL78 Family User's Manual: Software (R01US0015E)

(The latest versions can be downloaded from the Renesas Electronics website.)

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

(The latest versions can be downloaded from the Renesas Electronics website.)

## Revision History

| | | Description | |
|---|---|---|---|
| **Rev.** | **Date** | **Page** | **Summary** |
| Rev. 1.00 | Oct. 01, 2015 | — | First edition issued |
| Rev. 1.10 | May. 11, 2022 | 4 | Updated operation check conditions |

# General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit 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. Precaution against Electrostatic Discharge (ESD)

   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.

2. Processing at power-on

   The state of the product is undefined at the time 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 time 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 time 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 time when power is supplied until the power reaches the level at which resetting is specified.

3. Input of signal during power-off state

   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.

4. 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 the 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.

5. Clock signals

   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is 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. Additionally, 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.

6. Voltage application waveform at input pin

   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between $V_{IL}$ (Max.) and $V_{IH}$ (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between $V_{IL}$ (Max.) and $V_{IH}$ (Min.).

7. Prohibition of access to reserved addresses

   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.

8. Differences between products

   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of 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.

# Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving 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, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, 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 and impractical, you are responsible for evaluating the safety of the final products or systems 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. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall 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. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.
(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

## Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

## Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

## Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/.