RL78/G24

Timer RD2 Using Input Capture Function and Output Compare Function

Introduction

This application note explains how to utilize both the input capture function and output compare function of the timer RD2 in timer mode on the RL78/G24.

Target Device

RL78/G24

When applying the sample program covered in this application note to another microcomputer, modify the program according to the specifications for the target microcomputer and conduct an extensive evaluation of the modified program.

Contents

1. Specifications 3
1.1 Specification overview 3
1.2 Operation overview 4

2. Operation Confirmation Conditions 9

3. Hardware Description 10
3.1 Example of Hardware Configuration 10
3.2 List of used Pins 10

4. Software Description 11
4.1 Smart Configurator Settings 11
4.1.1 System Configuration 11
4.1.2 Component Configurations 13
4.2 Folder Structure 15
4.2.1 Changes in Config_TRD0_user.c 16
4.3 List of Option Byte Settings 18
4.4 List of Constants 18
4.5 List of Variables 18
4.6 List of Functions 19
4.7 Function Specifications 19
4.8 Flowchart 20
4.8.1 Main Process 20
4.8.2 R_Config_TRD0_Create_UserInit Function 21
4.8.3 r_Config_TRD0_trd0_interrupt function 22

5. Sample Code 23

6. Reference Documents 23

Revision History 24

1. Specifications

1.1 Specification overview

Use both the input capture function and output compare function of the timer RD2 counter 0 (Timer RD20).

With the input capture function, calculate the pulse width of the waveform input to the TRDIOA0 pin.

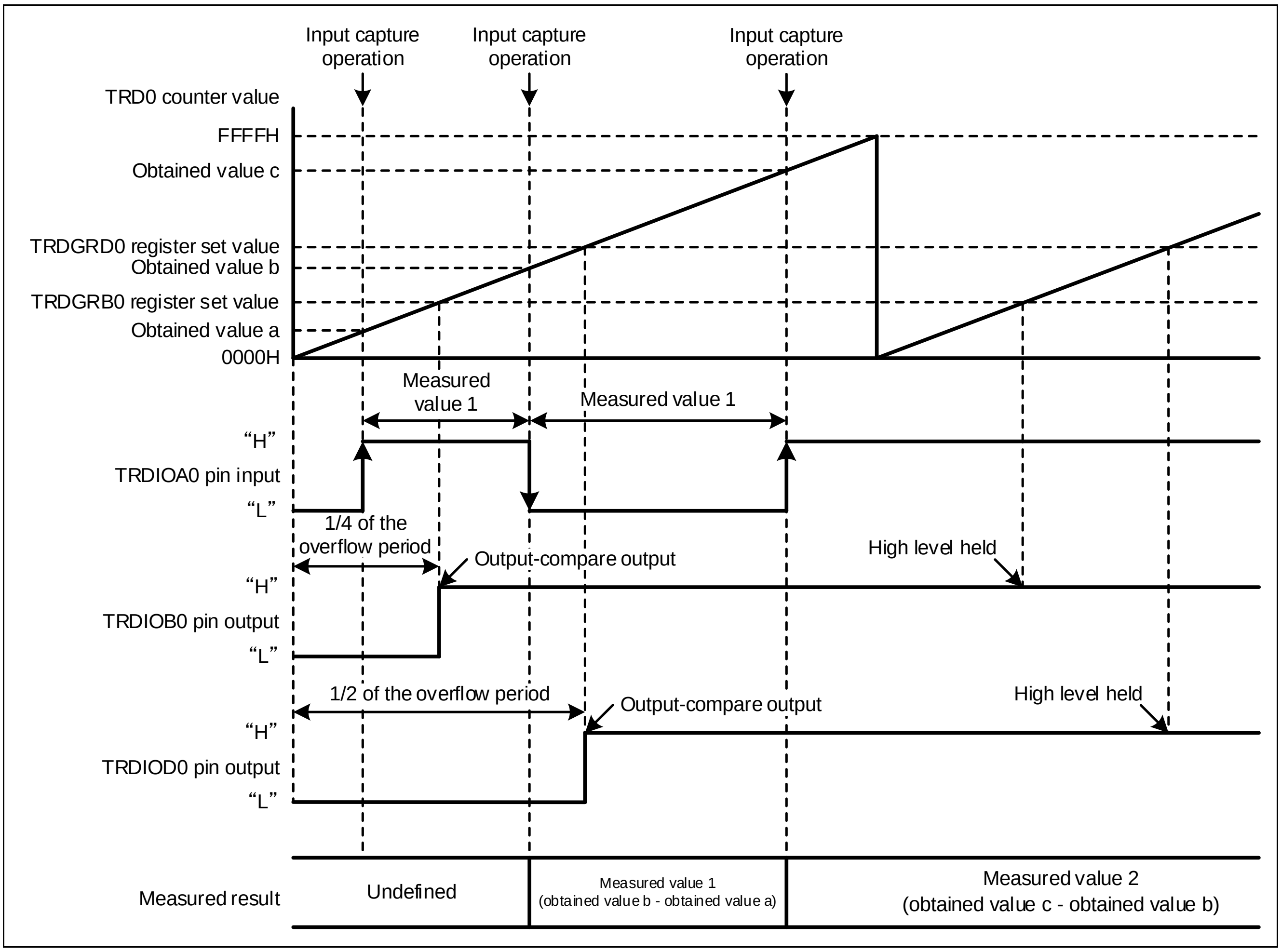
With the output compare function, change the output levels of the TRDIOB0 and TRDIOD0 pins from "L" to "H".

Table 1-1 lists the peripheral functions and their purposes, and Figure 1-1 shows the phase counting mode and the output waveform of the port.

Table 1-1 Peripheral Functions and Their Usage

Peripheral	Usage
Timer RD2 (Timer RD20)	Pulse width measurement and waveform output

Figure 1-1 Combined waveform output using input capture function and output compare function



1.2 Operation overview

Use both the input capture function and the output compare function of Timer RD20. The input capture function is assigned to the TRDIOA0 terminal, while the output compare function is designated for the TRDIOB0 and TRDIOD0 terminals. This application note employs code generation from the smart configurator. However, since it doesn't directly support the combined use of the two functions, the input capture function is managed through the smart configurator. For the output compare function, we use the UserInit function and incorporate the required operations.

The input capture function identifies both rising and falling edges inputted to the TRDIOA0 terminal. Based on the detection results, it computes the pulse width (high-level width and low-level width). Here are the settings for the input capture function:

<Settings>

Use the input capture function to configure TRD0.

Set the count source to fTRD (96MHz).

Disable counter reset.

The register function setting for both TRDGRC0 and TRDGRD0 is designated as general registers.

Input configuration of the TRDIOA0 terminal is set to both edges.

Employ the digital filter function on the TRDIOA0 terminal and select the count source (96MHz) as the sampling clock.

Enable the TRDGRA0 input capture interrupt and TRD0 overflow interrupt.

Figure 1-2 shows the operation of the input capture function

<Without TRD0 Register Overflow>

1. Detects a rising edge input to the TRDIOA0 terminal, leading to a TRDGRA0 input capture interrupt. Within the INTTRD0 interrupt process, compute the pulse width using the formula for no counter overflow and clear the IMFA flag.
2. A falling edge input to the TRDIOA0 terminal is detected, triggering the TRDGRA0 input capture interrupt. Within the INTTRD0 interrupt process, compute the pulse width without TRD0 counter overflow and clear the IMFA flag.

For the interval without overflow (as shown in Figures 1 and 2 section ①), the pulse width formula is:

Pulse Width = Time per count (1 / 96 [MHz] = 10.42 [ns]) × (current measurement - previous measurement)

= 10.42 [ns] × (44200 - 25000)

= 200 [µs]

<With TRD0 Register Overflow>

3. Overflow of the TRD0 register triggers a TRD0 overflow interrupt. Within the INTTRD0 interrupt process, increment the overflow counter value and clear the OVF flag.

4. Detects a rising edge input to the TRDIOA0 terminal, resulting in a TRDGRA0 input capture interrupt. During the INTTRD0 interrupt process, compute the pulse width using the formula accounting for TRD0 counter overflow and clear the IMFA flag.

For the interval with one overflow (as shown in Figures 1 and 2 section ②), the pulse width formula is:

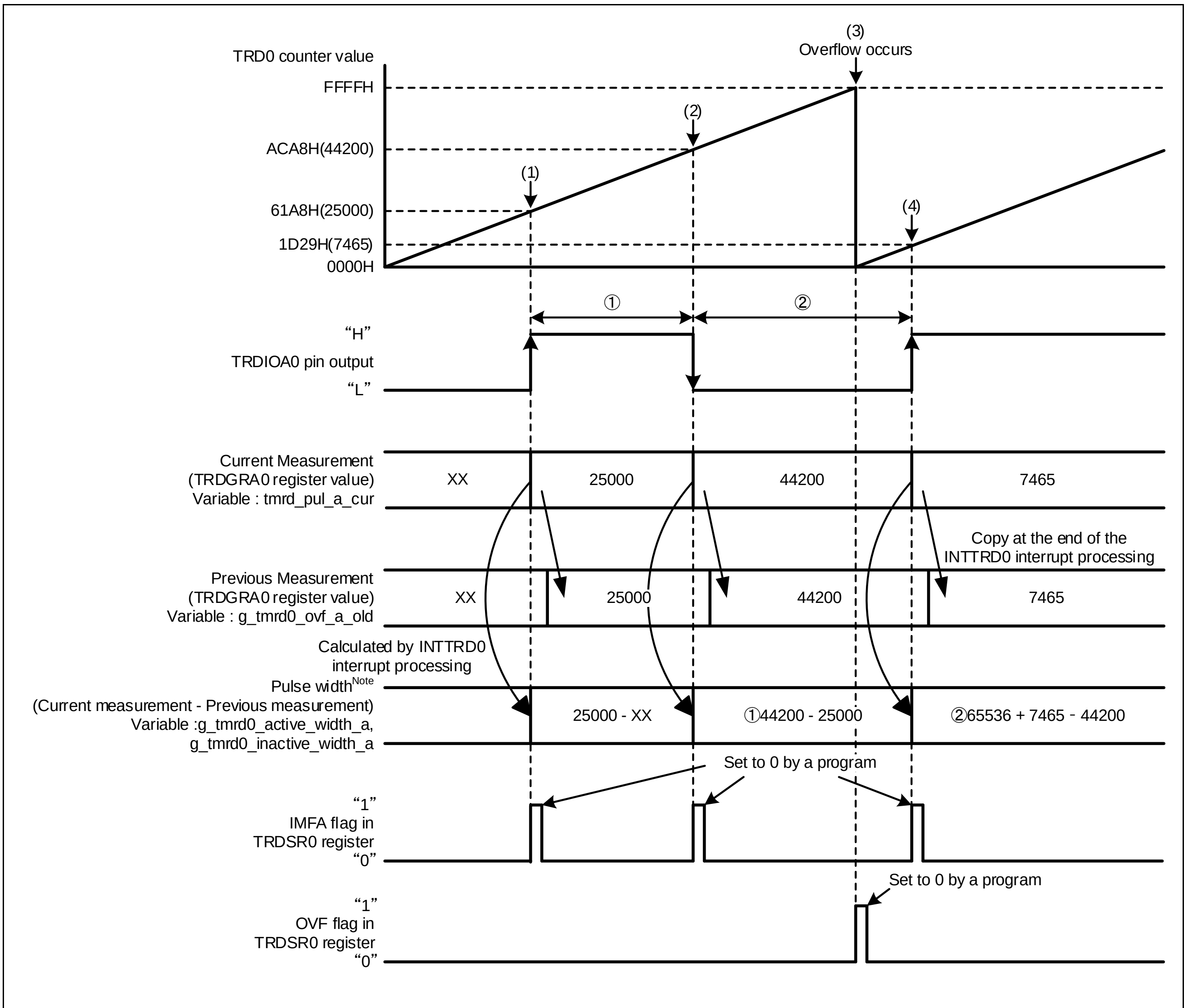
Pulse Width = Time per count (1 / 96 [MHz] = 10.42 [ns]) ×

(10000H(65536) × Overflow Count (1 time) + current measurement - previous measurement)

= 10.42 [ns] × (65536 + 7465 - 44200)

= 300 [µs]

Figure 1-2 The operation of input capture function



XX: Initial value

Note: In this application note, for the sake of simplification of calculations, the difference in measured values is treated as pulse width, and no conversion to time is performed.

The output compare function changes the output level of the TRDIOB0 terminal to "H" from "L" after 1/4 of the overflow cycle (683µs) from the start of timer RD0 count, and changes the output level of the TRDIOD0 terminal to "H" from "L" after 1/2 of the overflow cycle (683µs). Subsequently, both the TRDIOB0 and TRDIOD0 terminals maintain the "H" level.
Below are the settings for the output compare function.

<Settings>

· Set the count source to fTRD (96MHz).

· Set the counter to continue counting after matching TRDGRA0 compare, and prohibit clearing.

· The register function settings are set to general register for both TRDGRD0 and TRDGRC0.

· For the compare value setting, generate the code with the default as it will be set manually.

· The output settings are set to initial Low output for both TRDIOB0 and TRDIOD0 terminals, and to High level output upon compare match.

· Interrupts are not used.

Figure 1-3 illustrates the operation of the output compare function:

(1) Timer RD0 count starts.

(2) When the TRD0 register matches the TRDGRB0 register, the output level of the TRDIOB0 terminal changes from "L" to "H".

(3) When the TRD0 register matches the TRDGRD0 register, the output level of the TRDIOD0 terminal changes from "L" to "H".

(4) An overflow occurs in the TRD0 register, and the TRD0 register is cleared to “0000H”.

(5) The TRD0 register matches the TRDGRB0 register, but since the TRDGRB0 terminal output level is “H”, it maintains the output level.

(6) The TRD0 register matches the TRDGRD0 register, but since the TRDGRD0 terminal output level is “H”, it maintains the output level.

The formulas for the low-level output width of the TRDIOB0 and TRDIOD0 terminals are as follows:

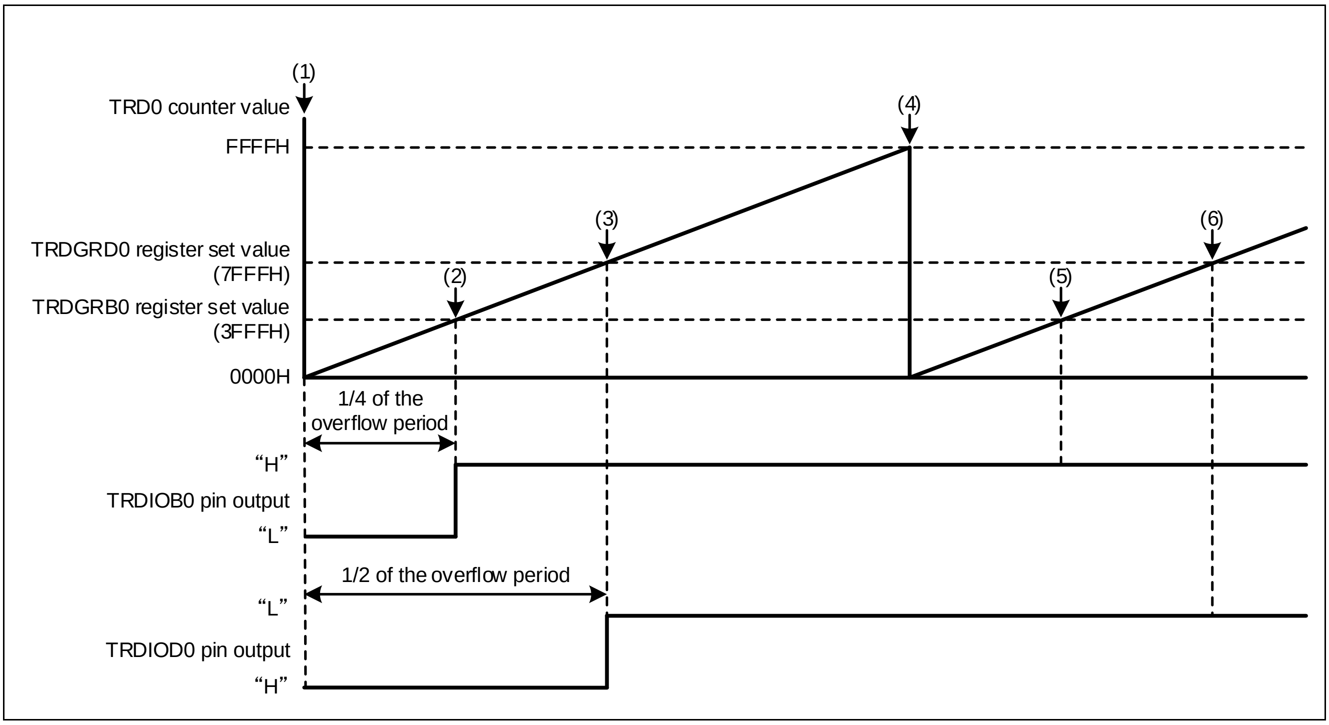
TRDIOB0 terminal: 1/4 of the overflow cycle (683µs) = 1 / 96 [MHz] × (TRDGRB0 register setting value + 1)

= 10.42 [ns] × (16383 + 1)

TRDIOD0 terminal: 1/2 of the overflow cycle (683µs) = 1 / 96 [MHz] × (TRDGRB0 register setting value + 1)

= 10.42 [ns] × (32767 + 1)

Figure 1-3 The operation of output compare function



By adding the following settings to the R_Config_TRD0_Create_UserInit function, simultaneous use of the input capture function and the output compare function can be achieved.

```
void R_Config_TRD0_Create_UserInit(void)
{
    /* Start user code for user init. Do not edit comment generated here */

    TRDOER1  = 0xF5U;        /* TRDIOD0 output disable : Output enabled
                              TRDIOC0 output disable : Output disabled
                             (TRDIOC0  pin functions as an I/O port.)
                              TRDIOB0 output disable : Output enable
                              TRDIOA0 output disable : Output disabled
                             (TRDIOA0 pin functions as an I/O port.) */

    TRDOCR   = 0x00U;        /* TRDIOD0 initial output level select :
                              Low initial output
                              TRDIOB0 initial output level select :
                              Low initial output */

    TRDIORA0 = 0x26U;        /* High output by compare match with TRDGRB0
                              TRDGRB mode select : Output compare function */

    TRDIORC0 = 0xACU;        /* High output by compare match with TRDGRD0
                              TRDGRD mode select : Output compare function */

    TRDGRB0  = 0x3FFFU;      /* Compare value : Set 1/4 of
                              the overflow period(683us) */

    TRDGRD0  = 0x7FFFU;      /* Compare value : Set 1/2 of
                              the overflow period(683us) */

    /* Set TRDIOB0 pin */
    POM1  &= 0xDFU;
    PMCA1 &= 0xDFU;
    P1    &= 0xDFU;
    PM1   &= 0xDFU;

    /* Set TRDIOD0 pin */
    POM1  &= 0xEFU;
    PMCA1 &= 0xEFU;
    P1    &= 0xEFU;
    PM1   &= 0xEFU;

    /* End user code. Do not edit comment generated here */
}
```

2. Operation Confirmation Conditions

The sample code described in this application note has been confirmed under the following conditions.

Table 2-1 Operation Confirmation Conditions

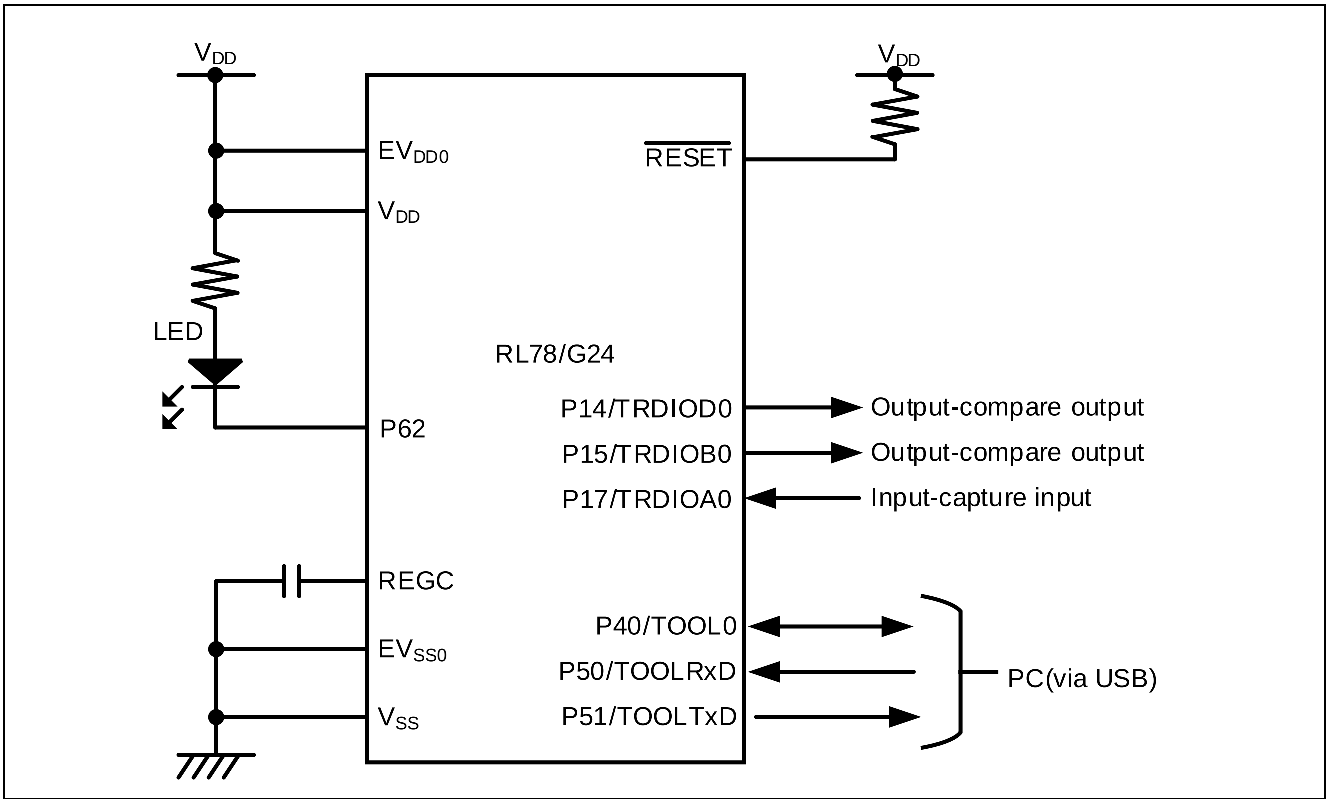
Item	Description
MCU used	RL78/G24 (R7F101GLG）
Operating frequency	· High-Speed On-Chip Oscillator Clock (fHOCO): 8MHz · PLL Oscillator Circuit Output (fPLL): 96MHz · CPU/Peripheral Hardware Clock (fCLK): 48MHz
Operating voltage	· 3.3V (Can operate between 2.7V to 5.5V) · LVD0 Operation (VLVD0): Reset Mode Rising edge = 2.97V Falling edge = 2.91V
Integrated development environment (CS+)	CS+ for CC V8.10.00 Manufactured by Renesas Electronics
C compiler (CS+)	CC-RL V1.12.01 Manufactured by Renesas Electronics
Integrated development environment (e^2 studio)	e^2 studio 2023-07 (23.7.0) Manufactured by Renesas Electronics
C compiler (e^2 studio)	CC-RL V1.12.00 Manufactured by Renesas Electronics
Integrated development Environment (IAR)	IAR Embedded Workbench for Renesas RL78 V4.21.1 Manufactured by IAR Systems
C compiler (IAR)	
Smart Configurator	V.1.7.0
Board Support Package (r_bsp)	V.1.60
Emulator	CS+, e^2 studio: COM port IAR: E2 Emulator Lite
Board used	RL78/G24 Fast Prototyping Board (RTK7RLG240C00000BJ)

3. Hardware Description

3.1 Example of Hardware Configuration

Figure 3-1 shows the hardware configuration example used in the sample code for this application.

Figure 3-1 Example of Hardware Configuration



Note 1. This simplified circuit diagram was created to show an overview of connections only. When actually designing your circuit, make sure the design includes appropriate pin handling and meets electrical characteristic requirements (connect each input-only port to VDD or VSS through a resistor).

Note 2. Connect any pins whose name begins with EVSS to VSS, and any pins whose name begins with EVDD to VDD, respectively.

Note 3. VDD must not be lower than the reset release voltage (VLVD0) that is specified for the LVD0.

3.2 List of used Pins

Table 3-1 shows the pins used and their functions.

Table 3-1 Pins Used and Their Functions

Pin name	I/O	Function
P14/TRDIOD0	Output	Output Compare Output
P15/TRDIOB0	Output	Output Compare Output
P17/TRDIOA0	Input	Input Capture Input
P62	Output	LED1 (Low Active)

Caution: In this application note, only the used pins are processed. When actually designing your circuit, make sure the design includes sufficient pin processing and meets electrical characteristic requirements.

4. Software Description

4.1 Smart Configurator Settings

This section presents the settings of the Smart Configurator used in this sample program. The items and settings in each table for the Smart Configurator are described as they appear in the configuration screen.

4.1.1 System Configuration

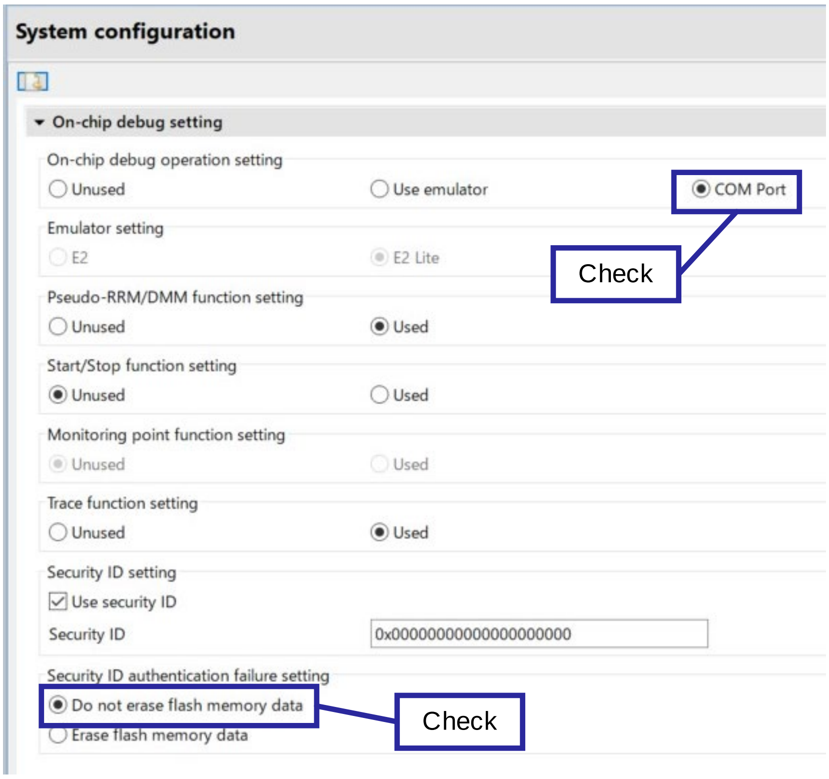
The system configuration used in this sample program are shown below.

Note that the system settings used in this sample program are the same for the integrated development environments e2 studio and CS+, but different for IAR. Please adjust the settings appropriately according to the environment you are using.

Firstly, Figure 4-1 shows the system configuration used in this sample program (for e2 studio and CS+).

If you are conducting a COM port debug on the RL78/G24 Fast Prototyping Board (RTK7RLG240C00000BJ), it is necessary to set the integrated development environments (e2 studio and CS+) appropriately. For details, please refer to the "RL78/G24 Fast Prototyping Board User’s Manual (R20UT5091)", specifically "7.1 Using COM Port Debugging with the e^2 studio" and "7.2 Using COM Port Debugging in CS+".

Figure 4-1 System Configuration (e^2 studio, CS+)



System configuration

On-chip debug setting

On-chip debug operation setting
○ Unused ○ Use emulator ● COM Port

Emulator setting
○ E2 ● E2 Lite

Pseudo-RRM/DMM function setting
○ Unused ● Used

Start/Stop function setting
● Unused ○ Used

Monitoring point function setting
● Unused ○ Used

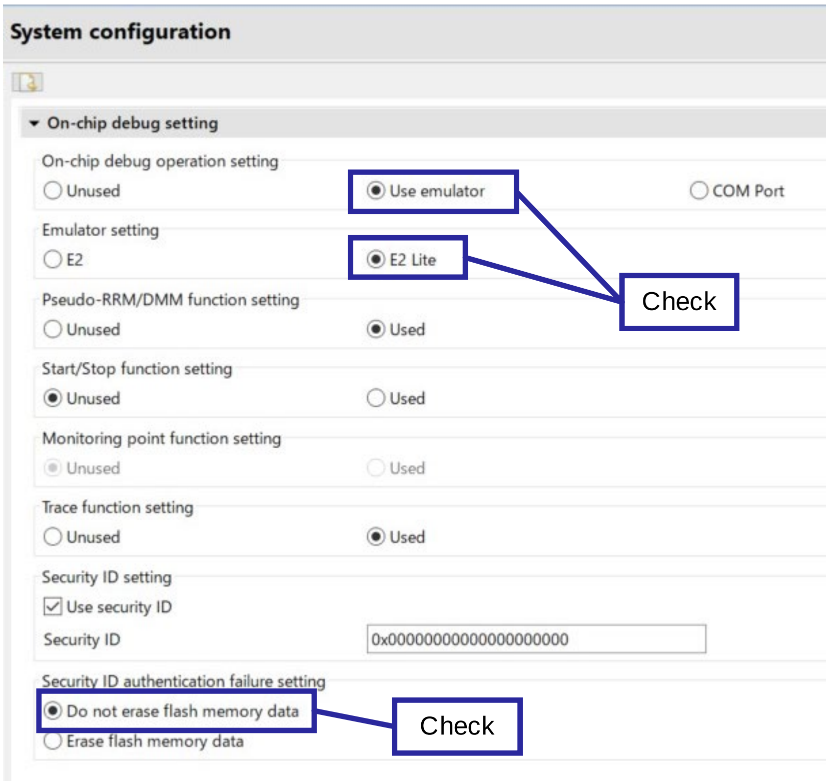
Trace function setting
○ Unused ● Used

Security ID setting
☑ Use security ID
Security ID 0x00000000000000000000

Security ID authentication failure setting
● Do not erase flash memory data
○ Erase flash memory data

Figure 4-2 shows the system configurations used in this sample program for IAR.

Figure 4-2 System Configurations (IAR)



System configuration

On-chip debug setting

On-chip debug operation setting
○ Unused ● Use emulator ○ COM Port

Emulator setting
○ E2 ● E2 Lite

Pseudo-RRM/DMM function setting
○ Unused ● Used

Start/Stop function setting
● Unused ○ Used

Monitoring point function setting
● Unused ○ Used

Trace function setting
○ Unused ● Used

Security ID setting
☑ Use security ID
Security ID 0x00000000000000000000

Security ID authentication failure setting
● Do not erase flash memory data
○ Erase flash memory data

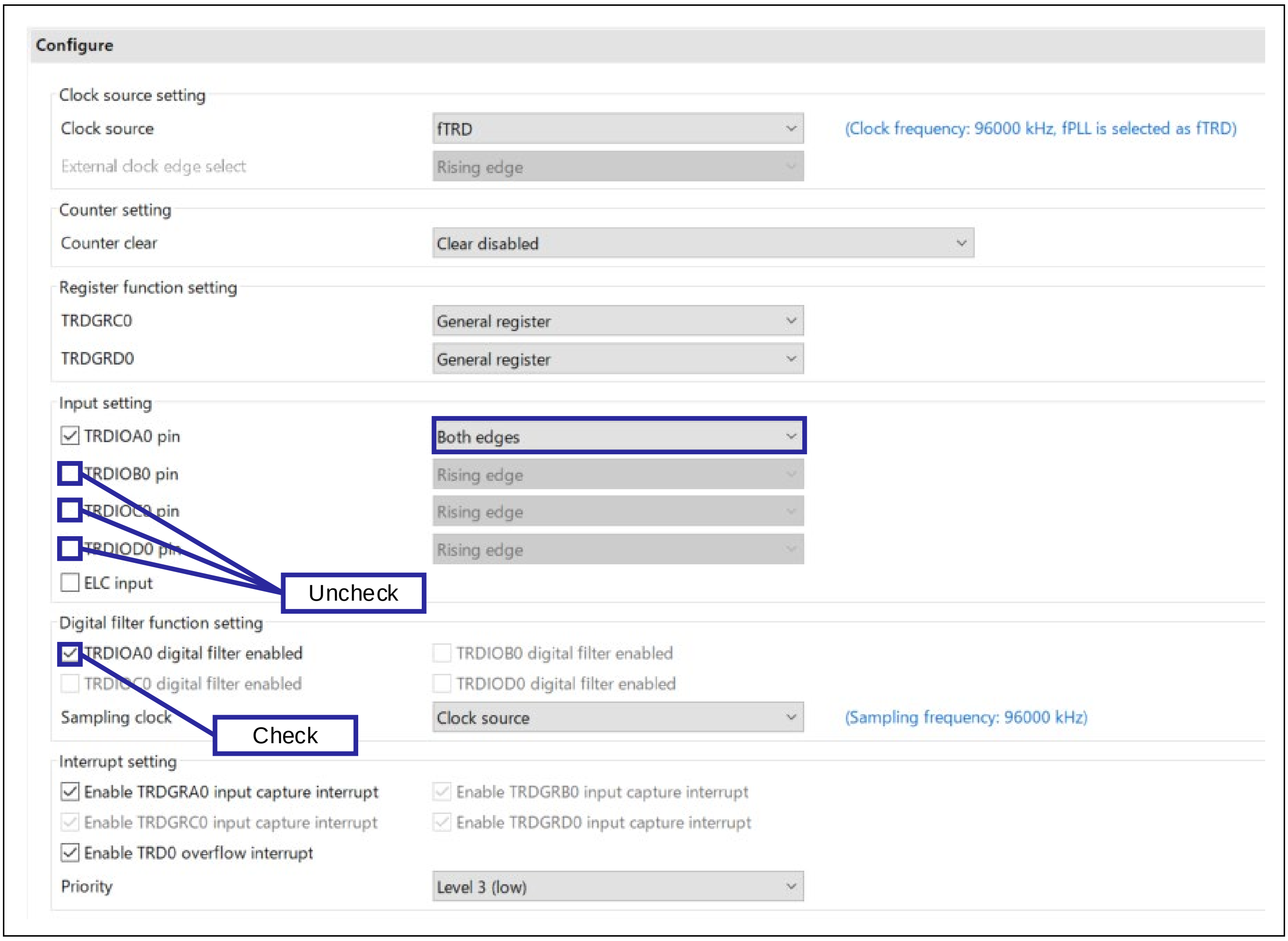
4.1.2 Component Configurations

This section presents the component configurations used in this sample program.

Table 4-1 Component Configurations (Timer RD2)

Item	Content
Component	Input capture mode
Configuration Name	Config_TRD0
Resource	TRD0

Figure 4-3 Configuration of Timer RD20



Configure

Clock source setting
- Clock source: fTRD (Clock frequency: 96000 kHz, fPLL is selected as fTRD)
- External clock edge select: Rising edge

Counter setting
- Counter clear: Clear disabled

Register function setting
- TRDGRC0: General register
- TRDGRD0: General register

Input setting
- ☑ TRDIOA0 pin: Both edges
- □ TRDIOB0 pin: Rising edge
- □ TRDIOC0 pin: Rising edge
- □ TRDIOD0 pin: Rising edge
- □ ELC input

Uncheck

Digital filter function setting
- ☑ TRDIOA0 digital filter enabled
- □ TRDIOB0 digital filter enabled
- □ TRDIOC0 digital filter enabled
- □ TRDIOD0 digital filter enabled
- Sampling clock: Clock source (Sampling frequency: 96000 kHz)

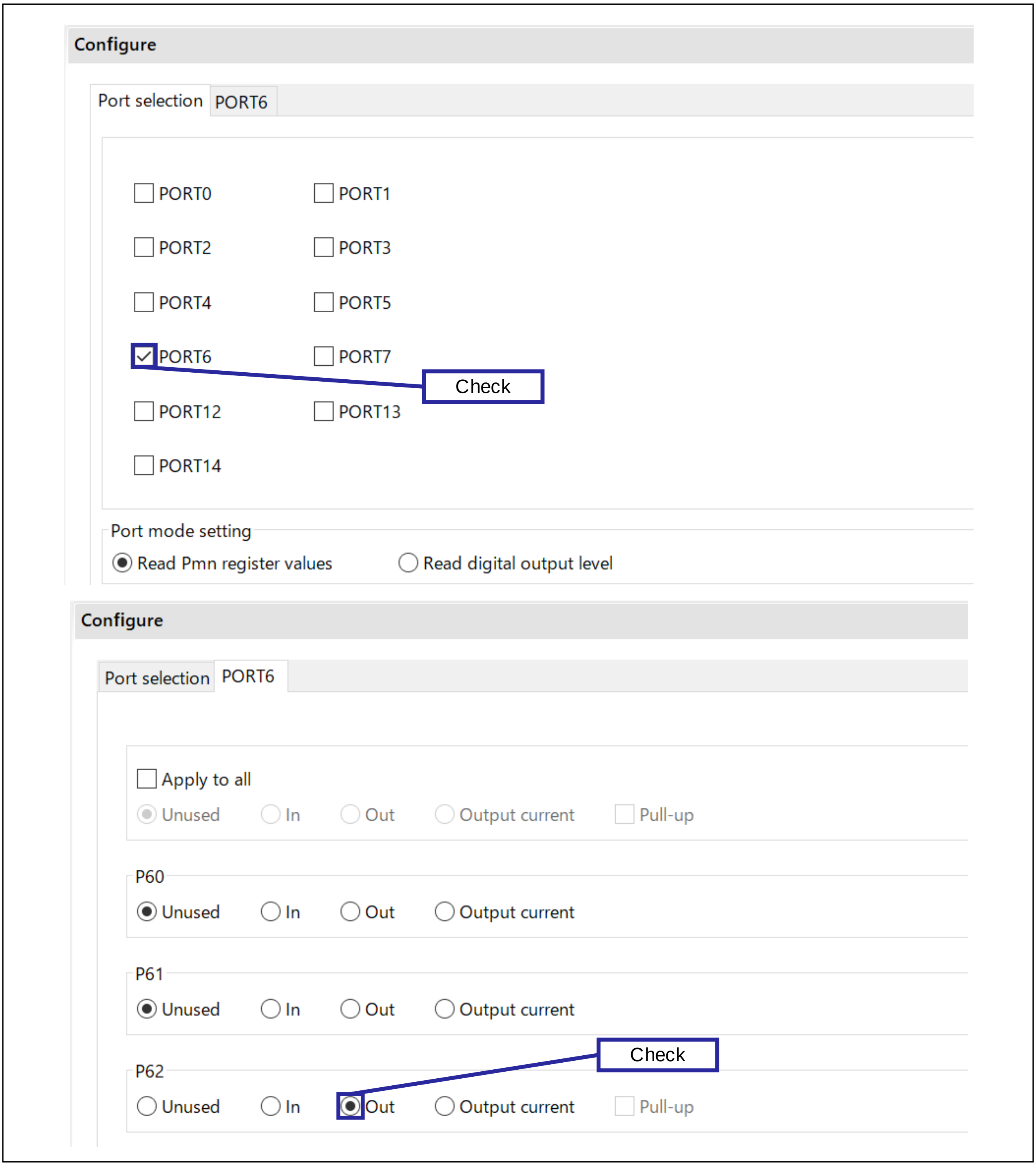
Check

Interrupt setting
- ☑ Enable TRDGRA0 input capture interrupt
- ☑ Enable TRDGRB0 input capture interrupt
- ☑ Enable TRDGRC0 input capture interrupt
- ☑ Enable TRDGRD0 input capture interrupt
- ☑ Enable TRD0 overflow interrupt
- Priority: Level 3 (low)

Table 4-2 Component Configurations (PORT)

Item	Content
Component	PORT
Configuration Name	Config_PORT
Resource	PORT

Figure 4-4 Configuration of PORT



Configure

Port selection | PORT6

☐ PORT0 ☐ PORT1
☐ PORT2 ☐ PORT3
☐ PORT4 ☐ PORT5
☑ PORT6 ☐ PORT7
☐ PORT12 ☐ PORT13
☐ PORT14

Check

Port mode setting
● Read Pmn register values ○ Read digital output level

Configure

Port selection | PORT6

☐ Apply to all
● Unused ○ In ○ Out ○ Output current ☐ Pull-up

P60
● Unused ○ In ○ Out ○ Output current

P61
● Unused ○ In ○ Out ○ Output current

P62
○ Unused ○ In ● Out ○ Output current ☐ Pull-up

Check

4.2 Folder Structure

Table 4-3 shows the structure of the source files/header files used in the sample code. Note that files automatically generated by the integrated development environment and files from the BSP environment are excluded.

Table 4-3 Folder Structure

Folder/File Name	Description	Generated by Smart Configurator
\r01an6892_trd2_timer<DIR>[NOTE 3]	Sample code folder	
\src<DIR>	Program storage folder	
main.c	Sample code source file	
\smc_gen<DIR>	Smart configurator generated folder	√
\Config_PORT<DIR>	PORT program storage folder	√
Config_PORT.c	PORT source file	√
Config_PORT.h	PORT header file	√
Config_PORT_user.c	PORT interrupt source file	√[NOTE 1]
\Config_TRD0<DIR>	TRD0 program storage folder	√
Config_TRD0.c	TRD0 source file	√
Config_TRD0.h	TRD0 header file	√
Config_TRD0_user.c	TRD0 interrupt source file	√[NOTE 2]
¥general<DIR>	Initialization and common program storage folder	√
¥r_bsp<DIR>	BSP program storage folder	√
¥r_config<DIR>	Program storage folder	√

Note: "<DIR>" indicates a directory.

Note 1: Not used in the sample code.

Note 2: "Modifications have been made to the output of the Smart Configurator. For details on the changes, please refer to section 4.2.1.

Note 3: The sample code for IAR contains the r01an6892_trd2_timer.ipcf file.
For details on the .ipcf file, please refer to "RL78 Smart Configurator User's Guide: IAR" (R20AN0581).

4.2.1 Changes in Config_TRD0_user.c

In this application note, changes have been made to the Config_TRD0_user.c generated by the Smart Configurator. In the R_Config_TRD0_Create_UserInit function, the settings described in '1.2 Operational Overview' have been added. In the r_Config_TRD0_trd0_interrupt function, the sections highlighted in yellow below have been modified.

<table>
<tr><th colspan="2">Function name: r_Config_TRD0_trd0_interrupt</th></tr>
<tr><th>Before change</th><th>After change</th></tr>
<tr><td>static void __near</td><td>static void __near</td></tr>
<tr><td>r_Config_TRD0_trd0_interrupt(void)</td><td>r_Config_TRD0_trd0_interrupt(void)</td></tr>
<tr><td>{</td><td>{</td></tr>
<tr><td>uint16_t tmrd_pul_a_cur = TRDGRA0;</td><td>uint16_t tmrd_pul_a_cur = TRDGRA0;</td></tr>
<tr><td>uint8_t trdier0_temp = TRDIER0;</td><td>uint8_t trdier0_temp = TRDIER0;</td></tr>
<tr><td>TRDIER0 = 0x00U;</td><td>TRDIER0 = 0x00U;</td></tr>
<tr><td>/* overflow process */</td><td>/* overflow process */</td></tr>
<tr><td>if ((TRDSR0 &</td><td>if ((TRDSR0 &</td></tr>
<tr><td>_10_TRD_INTOV_GENERATE_FLAG) ==</td><td>_10_TRD_INTOV_GENERATE_FLAG) ==</td></tr>
<tr><td>_10_TRD_INTOV_GENERATE_FLAG)</td><td>_10_TRD_INTOV_GENERATE_FLAG)</td></tr>
<tr><td>{</td><td>{</td></tr>
<tr><td>TRDSR0 &=</td><td>TRDSR0 &=</td></tr>
<tr><td>(uint8_t)~_10_TRD_INTOV_GENERATE_FLAG;</td><td>(uint8_t)~_10_TRD_INTOV_GENERATE_FLAG;</td></tr>
<tr><td>g_tmrd0_ovf_a += 1U;</td><td>g_tmrd0_ovf_a += 1U;</td></tr>
<tr><td></td><td>/* Start user code */</td></tr>
<tr><td></td><td>g_capture_flg |= 0x01U;</td></tr>
<tr><td></td><td>/* End user code */</td></tr>
<tr><td>}</td><td>}</td></tr>
<tr><td>/* TRDGRA0 input capture interrupt */</td><td>/* TRDGRA0 input capture interrupt */</td></tr>
<tr><td>if ((TRDSR0 &</td><td>if ((TRDSR0 &</td></tr>
<tr><td>_01_TRD_INTA_GENERATE_FLAG) ==</td><td>_01_TRD_INTA_GENERATE_FLAG) ==</td></tr>
<tr><td>_01_TRD_INTA_GENERATE_FLAG)</td><td>_01_TRD_INTA_GENERATE_FLAG)</td></tr>
<tr><td>{</td><td>{</td></tr>
<tr><td>TRDSR0 &=</td><td>TRDSR0 &=</td></tr>
<tr><td>(uint8_t)~_01_TRD_INTA_GENERATE_FLAG;</td><td>(uint8_t)~_01_TRD_INTA_GENERATE_FLAG;</td></tr>
<tr><td>if (0U == g_tmrd0_ovf_a)</td><td>if (0U == g_tmrd0_ovf_a)</td></tr>
<tr><td>{</td><td>{</td></tr>
<tr><td>if ((P1 & 0x80U) == 0x80U)</td><td>if ((P1 & 0x80U) == 0x80U)</td></tr>
<tr><td>{</td><td>{</td></tr>
<tr><td>g_tmrd0_inactive_width_a =</td><td>g_tmrd0_inactive_width_a =</td></tr>
<tr><td>(uint32_t)((uint32_t)tmrd_pul_a_cur -</td><td>(uint32_t)((uint32_t)tmrd_pul_a_cur -</td></tr>
<tr><td>(uint32_t)g_tmrd0_trdgra_old);</td><td>(uint32_t)g_tmrd0_trdgra_old);</td></tr>
<tr><td></td><td>/* Start user code */</td></tr>
<tr><td></td><td>g_capture_flg |= 0x02U;</td></tr>
<tr><td></td><td>/* End user code */</td></tr>
<tr><td>}</td><td>}</td></tr>
<tr><td>else</td><td>else</td></tr>
<tr><td>{</td><td>{</td></tr>
</table>

```
                g_tmrd0_active_width_a = 
(uint32_t)((uint32_t)tmrd_pul_a_cur - 
(uint32_t)g_tmrd0_trdgra_old);





            }
        }
        else
        {
            if ((P1 & 0x80U) == 0x80U)
            {
                g_tmrd0_inactive_width_a = 
(uint32_t)(((0x10000UL * (uint32_t)g_tmrd0_ovf_a) 
+ (uint32_t)tmrd_pul_a_cur) 
- (uint32_t)g_tmrd0_trdgra_old);





            }
            else
            {

                g_tmrd0_active_width_a = 
(uint32_t)(((0x10000UL * (uint32_t)g_tmrd0_ovf_a) 
+ (uint32_t)tmrd_pul_a_cur) 
- (uint32_t)g_tmrd0_trdgra_old);





            }
            g_tmrd0_ovf_a = 0U;
        }
        g_tmrd0_trdgra_old = tmrd_pul_a_cur;
    }

    TRDIER0 = trdier0_temp;
}
```

```
                g_tmrd0_active_width_a = 
(uint32_t)((uint32_t)tmrd_pul_a_cur - 
(uint32_t)g_tmrd0_trdgra_old);
                /* Start user code */

                g_capture_flg |= 0x04U;

                /* End user code */
            }
        }
        else
        {
            if ((P1 & 0x80U) == 0x80U)
            {
                g_tmrd0_inactive_width_a = 
(uint32_t)(((0x10000UL * (uint32_t)g_tmrd0_ovf_a) 
+ (uint32_t)tmrd_pul_a_cur) 
- (uint32_t)g_tmrd0_trdgra_old);
                /* Start user code */

                g_capture_flg |= 0x02U;

                /* End user code */
            }
            else
            {

                g_tmrd0_active_width_a = 
(uint32_t)(((0x10000UL * (uint32_t)g_tmrd0_ovf_a) 
+ (uint32_t)tmrd_pul_a_cur) 
- (uint32_t)g_tmrd0_trdgra_old);
                /* Start user code */

                g_capture_flg |= 0x04U;

                /* End user code */
            }
            g_tmrd0_ovf_a = 0U;
        }
        g_tmrd0_trdgra_old = tmrd_pul_a_cur;
    }

    TRDIER0 = trdier0_temp;
}
```

Note: The function r_Config_TRD0_trd0_interrupt utilizes designated tags (/* Start user code */ .../* End user code */) where user codes can be added at any position. As a result, even if the program is auto-generated using the Smart Configurator again, the output value (before modification) won't be overwritten, and the added user code will be merged (protected). For details on the designated tags, please refer to the "2.3.1 User code protection feature for Smart Configurator Code Generation component "RL78 Smart Configurator V1.5.0 Release Notes".

4.3 List of Option Byte Settings

Figure 4-3 shows the option byte settings.

Table 4-4 Option Byte Settings

Address	Setting Value	Description
000C0H/040C0H	1110 1111B (EFH)	Watchdog Timer stopped operation (Count stops after reset release)
000C1H/040C1H	1111 1011B (FBH)	LVD0 reset mode. Detection voltage: Rising 2.97V / Falling 2.91V
000C2H/040C2H	1110 1010B (EAH)	Flash operation mode: High-speed main mode. High-speed on-chip oscillator frequency: 8MHz
000C3H/040C3H	1000 0101B (85H)	On-chip debug operation allowed

4.4 List of Constants

Constant is not used in the sample code.

4.5 List of Variables

Table 4-5 shows the variables used in the sample code.

The following variables are generated by the Smart Configurator.

Table 4-5 Variables used in the sample code

Type	Variable Name	Contents	Function that uses the variable
g_tmrd0_active_width_a	uint32_t	Storage for the calculated high-level pulse width	r_Config_TRD0_trd0_interrupt
g_tmrd0_inactive_width_a	uint32_t	Storage for the calculated low-level pulse width	r_Config_TRD0_trd0_interrupt
g_tmrd0_trdgra_old	uint16_t	Previous measurement value (Value of TRDGRA0 register)	r_Config_TRD0_trd0_interrupt
g_tmrd0_ovf_a	uint8_t	Overflow counter	r_Config_TRD0_trd0_interrupt
tmrd_pul_a_cur	uint16_t	Current measurement value (Value of TRDGRA0 register)	r_Config_TRD0_trd0_interrupt
g_capture_flg	uint8_t	Pulse width calculation completion flag	r_Config_TRD0_trd0_interrupt

4.6 List of Functions

Table 4-6 lists the functions used in the sample code. However, functions generated by the Smart Configurator that have not been modified are excluded.

Table 4-6 List of Functions

Function Name	Description	Source File
main	main process	main.c
R_Config_TRD0_Create_UserInit	Initial setting for the output compare function	Config_TRD0_user.c
r_Config_TRD0_trd0_interrupt	Timer RD20 interrupt processing	Config_TRD0_user.c

4.7 Function Specifications

The function specifications of the sample code are presented.

[Function Name] main

Outline	Main process
Header	r_smc_entry.h
Declaration	void main (void);
Explanation	Start the operation of Timer RD20
Arguments	-
Return value	-
Remarks	-

[Function Name] R_Config_TRD0_Create_UserInit

Outline	Initial setting for the output compare function
Header	Config_TRD0.h
Declaration	void R_Config_TRD0_Create_UserInit(void);
Explanation	Initial settings for the output compare function are performed.
Arguments	-
Return value	-
Remarks	-

[Function Name] r_Config_TRD0_trd0_interrupt

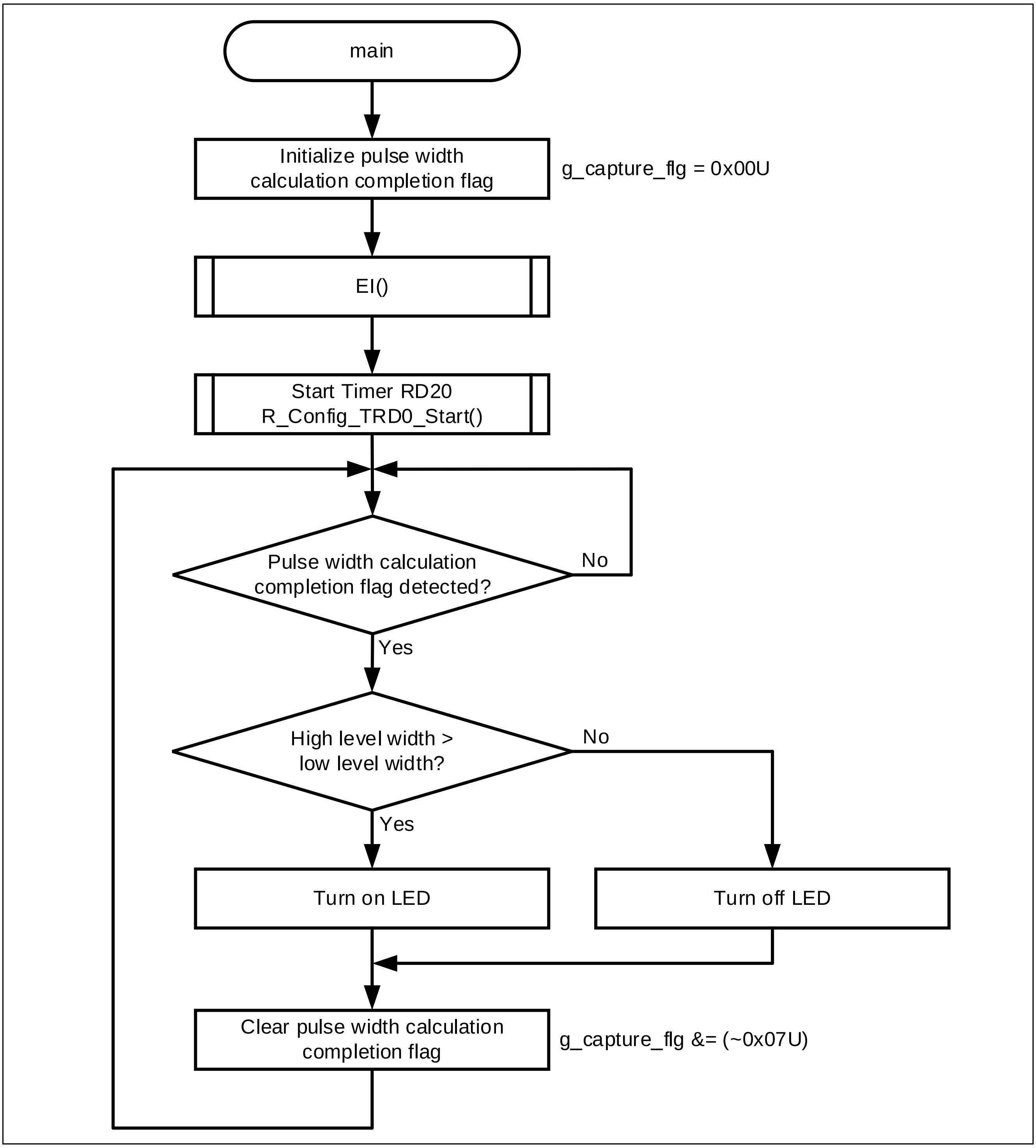
Outline	Timer TD20 Interrupt Processing
Header	Config_TRD0.h
Declaration	static void __near r_Config_TRD0_trd0_interrupt(void)
Explanation	When an edge is input to the TRDIOA0 terminal, pulse width calculation is performed. In addition, the pulse width calculation completion flag, g_capture_flg, is set under the following conditions: ・When an overflow is detected: set the 0th bit to 1 (0x01U). ・When the high-level width calculation result is stored: set the 1st bit to 1 (0x02U). ・When the low-level width calculation result is stored: set the 2nd bit to 1 (0x04U)。
Arguments	-
Return value	-
Remarks	-

4.8 Flowchart

4.8.1 Main Process

Figure 4-5 shows the flowchart for the main process.

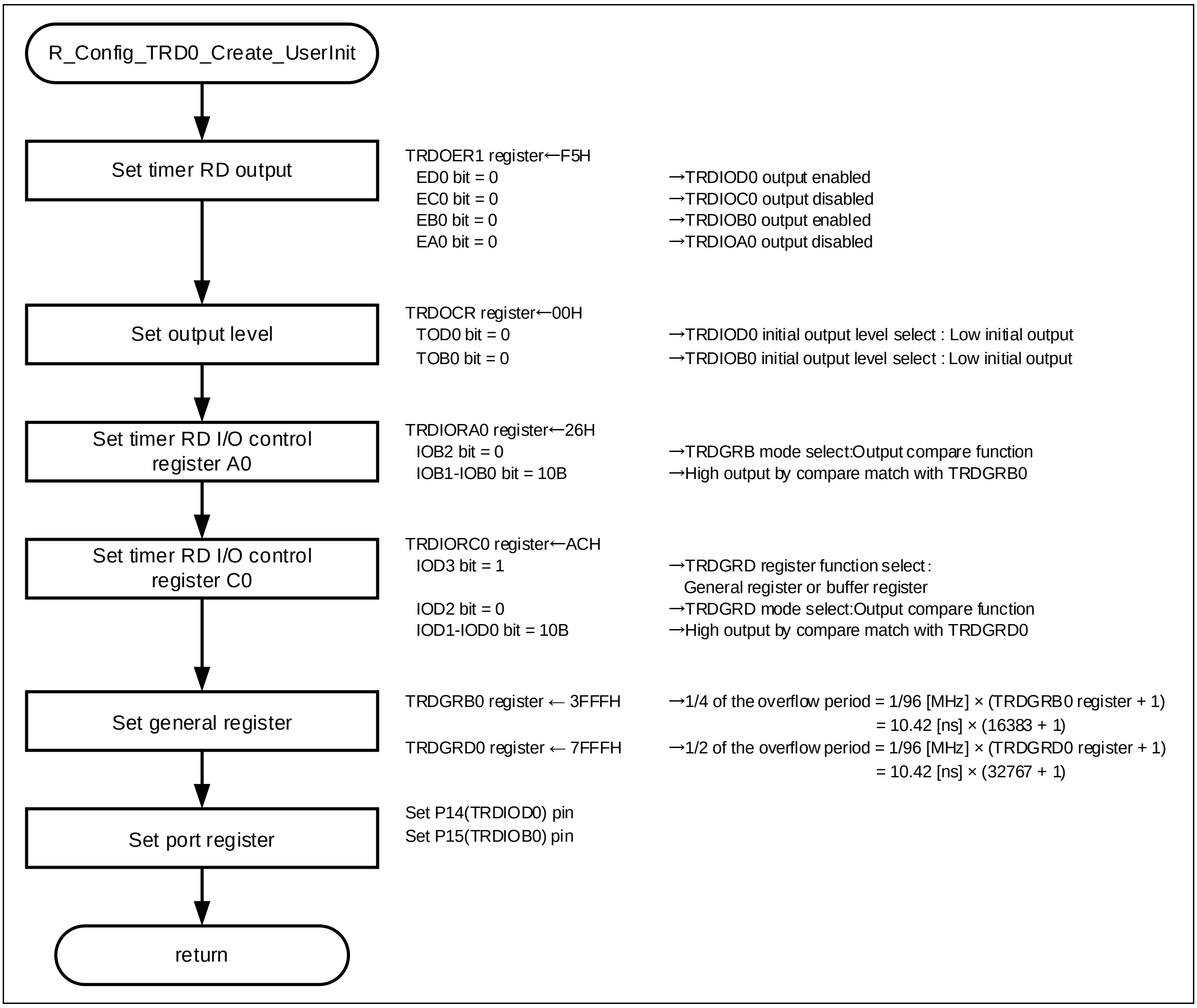
Figure 4-5 Main Process


main
Initialize pulse width calculation completion flag
g_capture_flg = 0x00U
EI()
Start Timer RD20
R_Config_TRD0_Start()
Pulse width calculation completion flag detected?
No
Yes
High level width > low level width?
No
Yes
Turn on LED
Turn off LED
Clear pulse width calculation completion flag
g_capture_flg &= (~0x07U)

4.8.2 R_Config_TRD0_Create_UserInit Function

Figure 4.6 shows the flowchart of the R_Config_TRD0_Create_UserInit function.

Figure 4-6 R_Config_TRD0_Create_UserInit Function


R_Config_TRD0_Create_UserInit
Set timer RD output
Set output level
Set timer RD I/O control register A0
Set timer RD I/O control register C0
Set general register
Set port register
return

TRDOER1 register←F5H
ED0 bit = 0 →TRDIOD0 output enabled
EC0 bit = 0 →TRDIOC0 output disabled
EB0 bit = 0 →TRDIOB0 output enabled
EA0 bit = 0 →TRDIOA0 output disabled

TRDOCR register←00H
TOD0 bit = 0 →TRDIOD0 initial output level select : Low initial output
TOB0 bit = 0 →TRDIOB0 initial output level select : Low initial output

TRDIORA0 register←26H
IOB2 bit = 0 →TRDGRB mode select:Output compare function
IOB1-IOB0 bit = 10B →High output by compare match with TRDGRB0

TRDIORC0 register←ACH
IOD3 bit = 1 →TRDGRD register function select :
General register or buffer register
IOD2 bit = 0 →TRDGRD mode select:Output compare function
IOD1-IOD0 bit = 10B →High output by compare match with TRDGRD0

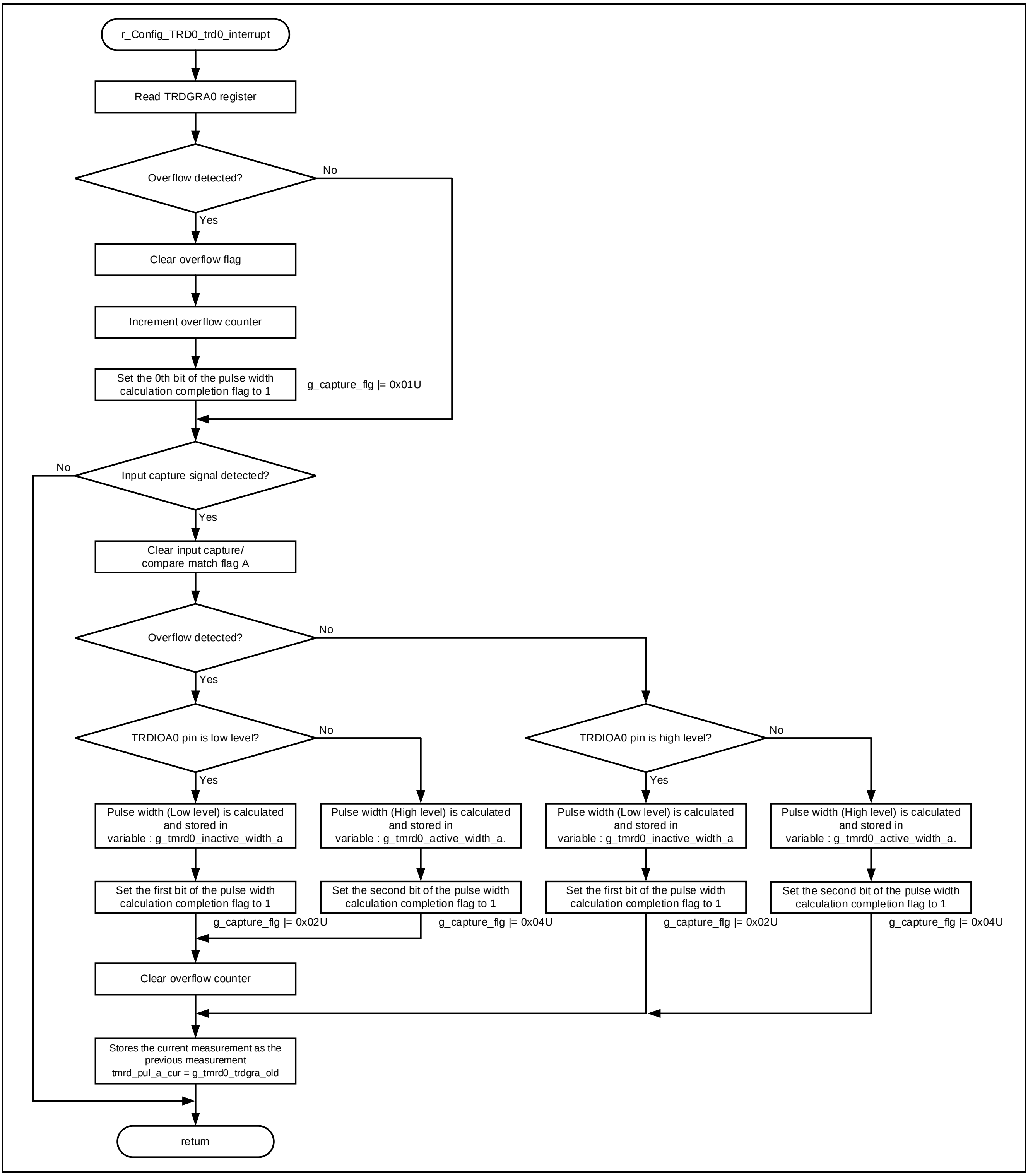
TRDGRB0 register ← 3FFFH →1/4 of the overflow period = 1/96 [MHz] × (TRDGRB0 register + 1)
= 10.42 [ns] × (16383 + 1)
TRDGRD0 register ← 7FFFH →1/2 of the overflow period = 1/96 [MHz] × (TRDGRD0 register + 1)
= 10.42 [ns] × (32767 + 1)

Set P14(TRDIOD0) pin
Set P15(TRDIOB0) pin

4.8.3 r_Config_TRD0_trd0_interrupt function

Figure 4-7 shows the flowchart of r_Config_TRD0_trd0_interrupt function

Figure 4-7 r_Config_TRD0_trd0_interrupt function


r_Config_TRD0_trd0_interrupt
Read TRDGRA0 register
Overflow detected?
No
Yes
Clear overflow flag
Increment overflow counter
Set the 0th bit of the pulse width calculation completion flag to 1
g_capture_flg |= 0x01U
No
Input capture signal detected?
Yes
Clear input capture/ compare match flag A
Overflow detected?
No
Yes
TRDIOA0 pin is low level?
No
Yes
TRDIOA0 pin is high level?
No
Yes
Pulse width (Low level) is calculated and stored in variable : g_tmrd0_inactive_width_a
Pulse width (High level) is calculated and stored in variable : g_tmrd0_active_width_a.
Pulse width (Low level) is calculated and stored in variable : g_tmrd0_inactive_width_a
Pulse width (High level) is calculated and stored in variable : g_tmrd0_active_width_a.
Set the first bit of the pulse width calculation completion flag to 1
Set the second bit of the pulse width calculation completion flag to 1
Set the first bit of the pulse width calculation completion flag to 1
Set the second bit of the pulse width calculation completion flag to 1
g_capture_flg |= 0x02U
g_capture_flg |= 0x04U
g_capture_flg |= 0x02U
g_capture_flg |= 0x04U
Clear overflow counter
Stores the current measurement as the previous measurement
tmrd_pul_a_cur = g_tmrd0_trdgra_old
return

5. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

6. Reference Documents

RL78/G24 User’s Manual: Hardware (R01UH0961)

RL78 family User's Manual: Software (R01US0015)

RL78/G24 Fast Prototyping Board User’s Manual (R20UT5091)

RL78 Smart Configurator User’s Gude: CS+ (R20AN0580)

RL78 Smart Configurator User’s Gude: e2 studio (R20AN0579)

RL78 Smart Configurator User’s Gude: IAR (R20AN0581)

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

Technical Update/Technical News

(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

Rev.	Date	Description	
		Page	Summary
1.00	Sep.07.23	-	First Edition

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/.