To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

RENESAS

H8SX Series

One-Shot Pulse Output

Introduction

As well as having an architecture that is upward-compatible with each CPU of the H8/300, H8/300H, and H8S series, so as to inherit a full complement of peripheral functions, the H8SX microcomputer series has a maximum operating frequency of 50 MHz and uses a 32-bit H8SX core CPU as well as an on-chip multiplier/divider to improve performance.

This H8SX series Application Note provides information you may be need during software and hardware design. This is a basic edition that provides operation examples that each use a single H8SX series on-chip peripheral function.

Although the operation of each program, circuit, and other aspects covered by this application note has been checked, make sure that you conduct your own operation checks before actually using the H8SX series.

Contents

1. Overview

One channel of the 16-bit timer pulse unit (TPU) of the H8SX series is used to output a one-shot pulse in synchronization with an external signal. The input capture detects the falling edge of the external trigger signal while the output compare match controls the output delay and pulse width of the one-shot pulse, as well as the buffer operation.

2. Configuration

The following example uses channel 0 of the 16-bit timer pulse unit (TPU). This sample uses timer general register A (TGRA_0) of channel 0 as the input capture of the external trigger signal to control the following two times with the output compare match of timer general register B (TGRB_0): Pulse output delay and pulse width. This sample also uses timer general register D (TGRD_0) as the TGRB_0 buffer register and timer general register C (TGRC_0) to control the disabling of one-shot pulse output (after one-shot pulse output). You can set any pulse output delay and pulse output width within the range of values that can be set in the timer general registers. When the peripheral module clock (Pϕ) is 25 MHz and the count clock is Pϕ/1, you can set up to 2.62 msec in the timer general registers. In the following explanation, channel 0 of the 16-bit timer pulse unit is called TPU0. Figure 1 is a block diagram.

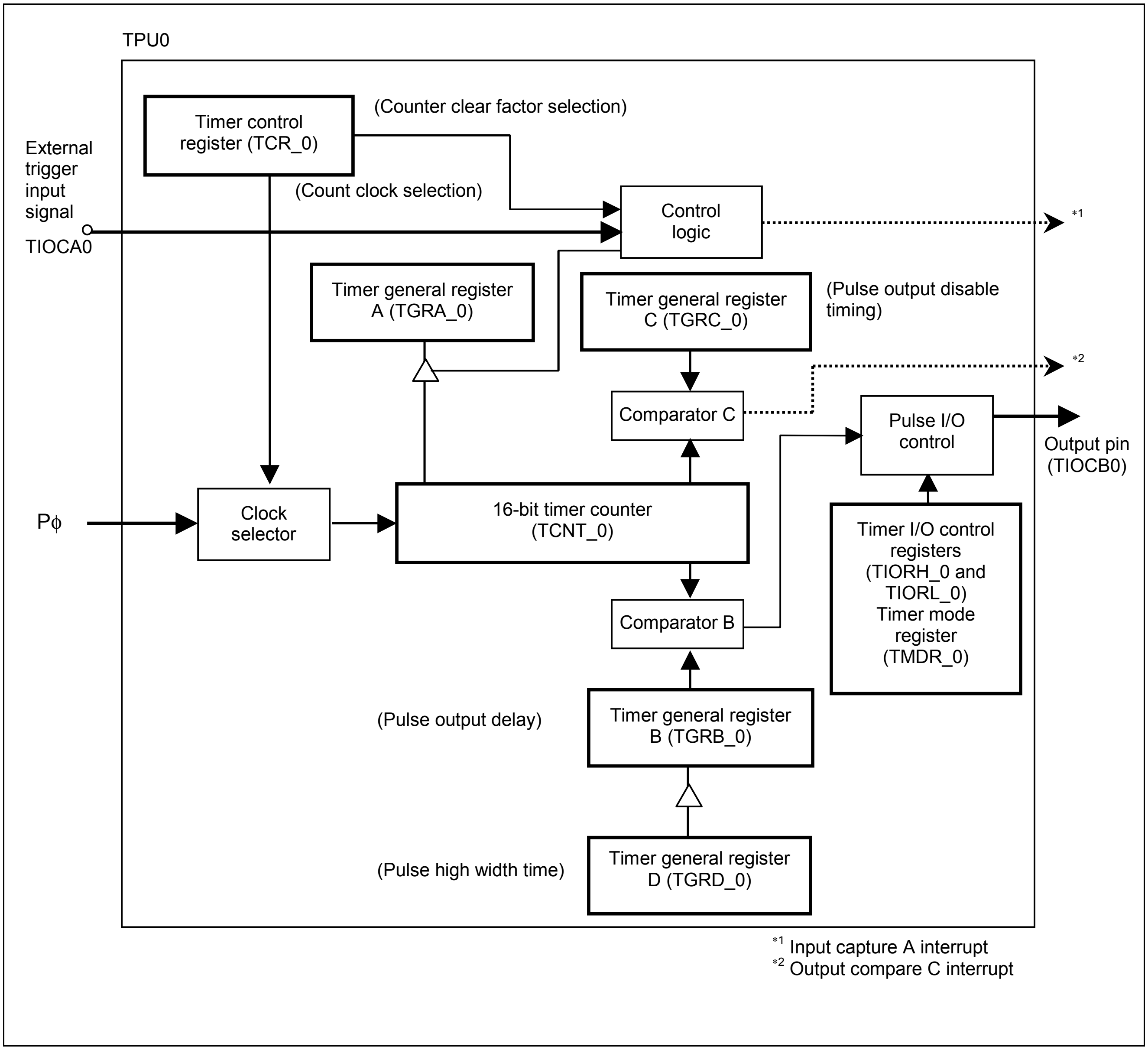


Figure 1 Block Diagram of One-Shot Pulse Output

Figure 2 illustrates one-shot pulse output.

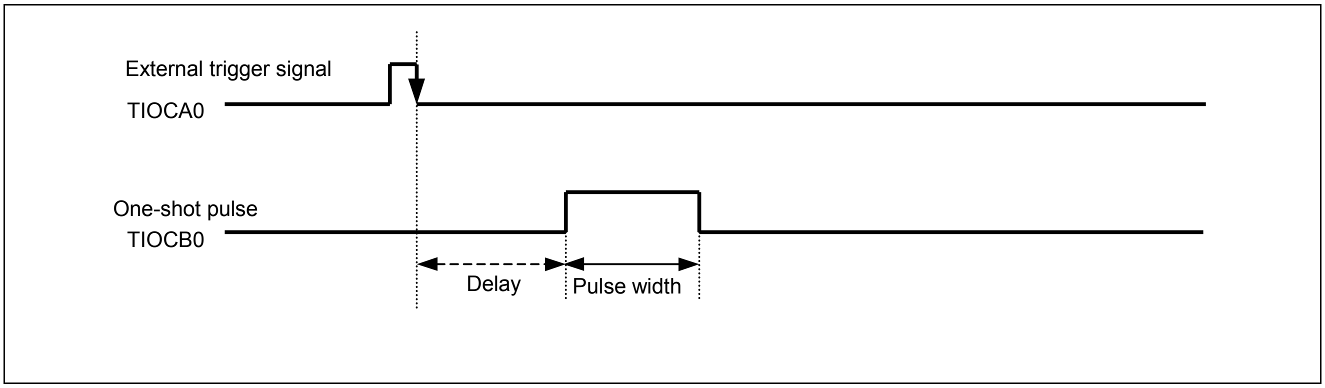


Figure 2   Example of One-Shot Pulse Output

3. Sample Program

3.1 Function

This sample program outputs pulses according to the timer counter values for the following two times for the one-shot pulse to be output:

- Delay between the falling edge of the external trigger signal and the start of high output of the pulse
- High width of the pulse to be output

You can calculate the timer values for the two times by using the following equations:

delay = timer-value × TPU0-count-clock
high-width = timer value × TPU0-count clock

Assume that the TPU0 count clock is peripheral module (Pϕ)/1. When Pϕ is 25 MHz, the TPU0 count clock is 40 nsec. Figure 3 shows an example of operation.

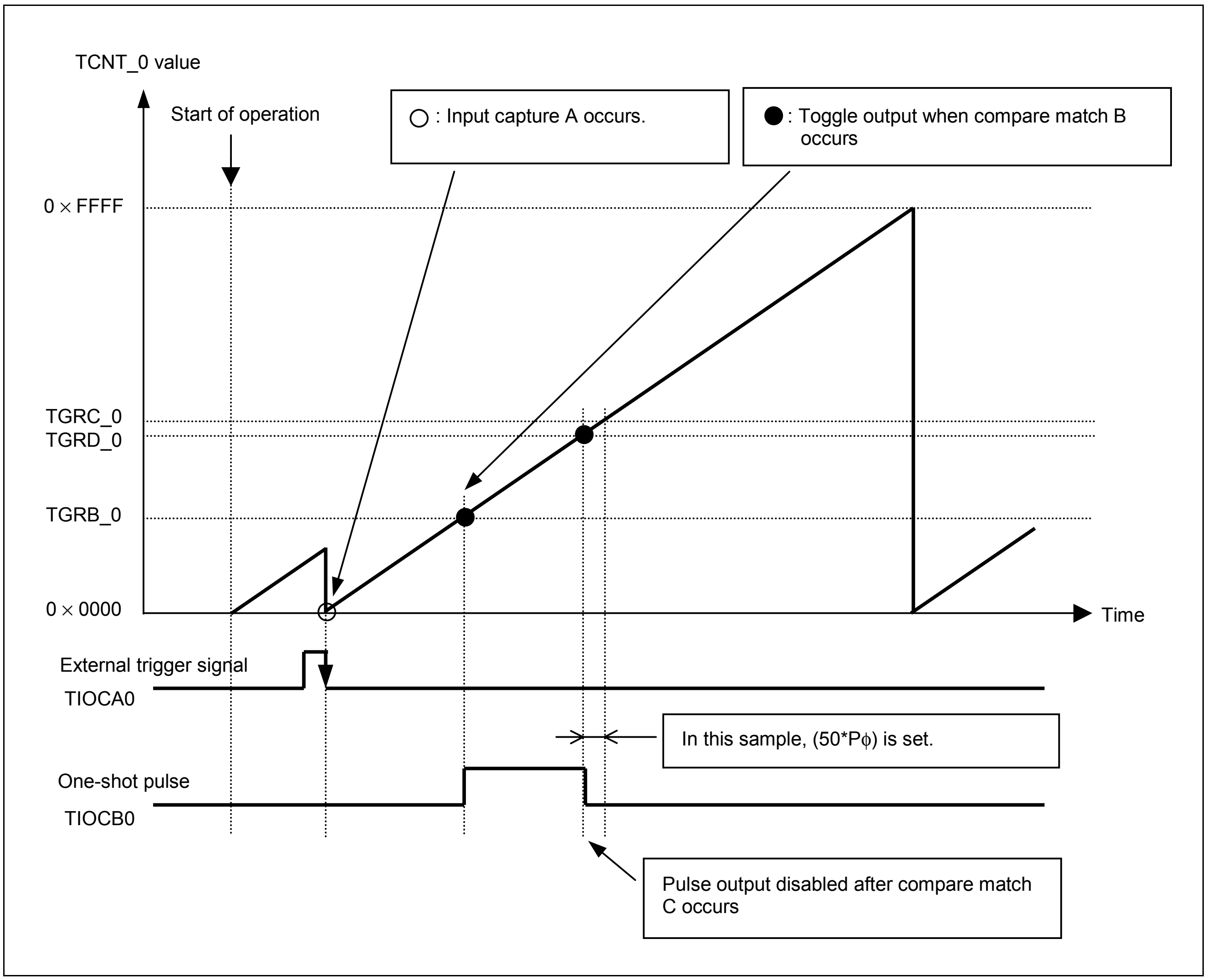


Figure 3 Example of One-Shot Pulse Output Operation

Table 1 lists the function allocations of channel 0 of the 16-bit timer pulse unit (TPU).

Table 1 Function Allocation of TPU0

Type	Name	Function
Register	MSTPCRA	Cancels TPU module stop mode.
	TSTR	Specifies whether to start or stop TPU0 timer count operation.
	TMDR_0	Sets TPU0 operating mode (buffer operation).
	TCR_0	Sets the TCNT_0 count clock and counter clear factor.
	TGRB_0	Compare match counter value for the pulse output delay time
	TGRD_0	Compare match counter value for the pulse output high width
	TGRC_0	Compare match counter value for the pulse output prohibit timing
	TIER_0	Enables interrupts by input capture A and compare match B.
	TIORH_0 TIORL_0	Sets TGRA_0 to the input capture and TGRB_0 and TGRC_0 to the compare matches. Sets the output level when a compare match occurs.
Input pin	TIOCA0	Input capture input pin
Output pin	TIOCB0	Compare match output pin

3.2 Function Specifications

The functions that set pulse output are shown as a sample program. The function specifications are listed below.

(1) Routine for setting one-shot pulse output

```
void  oneshot_set ( unsigned short  delay_ count, unsigned short  high_count )
```

Argument	Description
delay_count	Specifies the timer value for the delay between the falling edge of the external trigger signal and the start of one-shot pulse output (high output). The count clock is fixed to Pϕ/1.
high_count	Specifies the timer value for the high width of one-shot pulse output. The count clock is fixed to Pϕ/1.

Return value	Description
None	—

(2) Input capture A interrupt handler

```
void  inthdr_captureA ( void )
```

This function has neither an argument nor a return value because it is a TPU0 interrupt handler. You must register this interrupt handler in the interrupt vector table.

(3) Compare match C interrupt handler

```
void  inthdr_compareC ( void )
```

This function has neither an argument nor a return value because it is an interrupt handler. You must register this interrupt handler in the interrupt vector table.

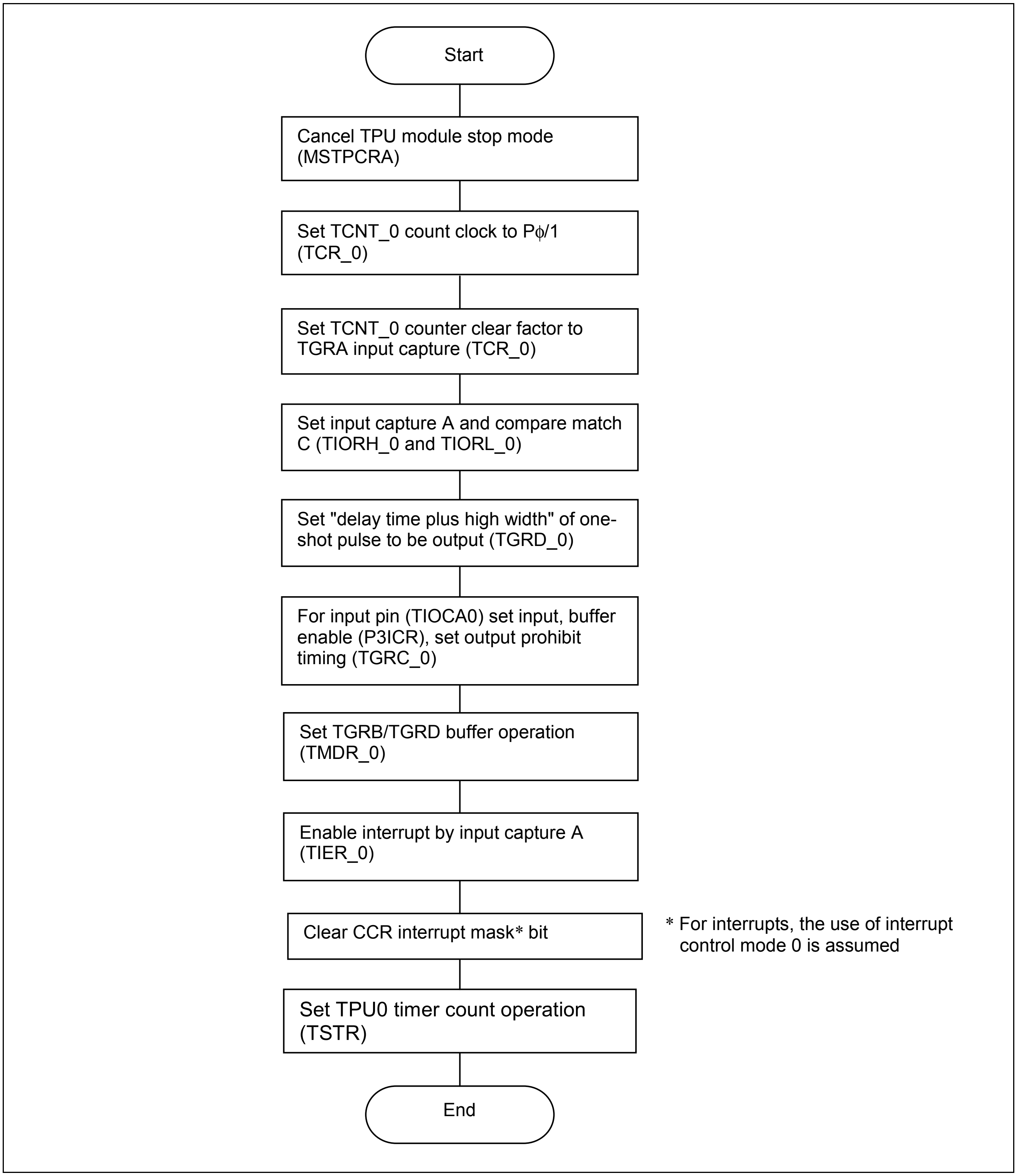
Example)

```
#define DELAY_TIME    1000                    // Delay time:  1000 µsec
#define ONSHOT_TIME    500                    // High width:  500 µsec
#define P_CLOCK         25                    // Pφ (MHz)
                                              // External function reference declaration
extern void  oneshot_set ( unsigned short, unsigned short );
void   main( void )                           // Main routine
{
   unsigned long  delay;
   unsigned long  width;

   delay = ((unsigned long)DELAY_TIME   *P_CLOCK);
   width = ((unsigned long)ONSHOT_TIME*P_CLOCK);
                                              // Sets one-shot pulse output.
   oneshot_set ( (unsigned short)delay,  (unsigned short)width );

   ..

}
```

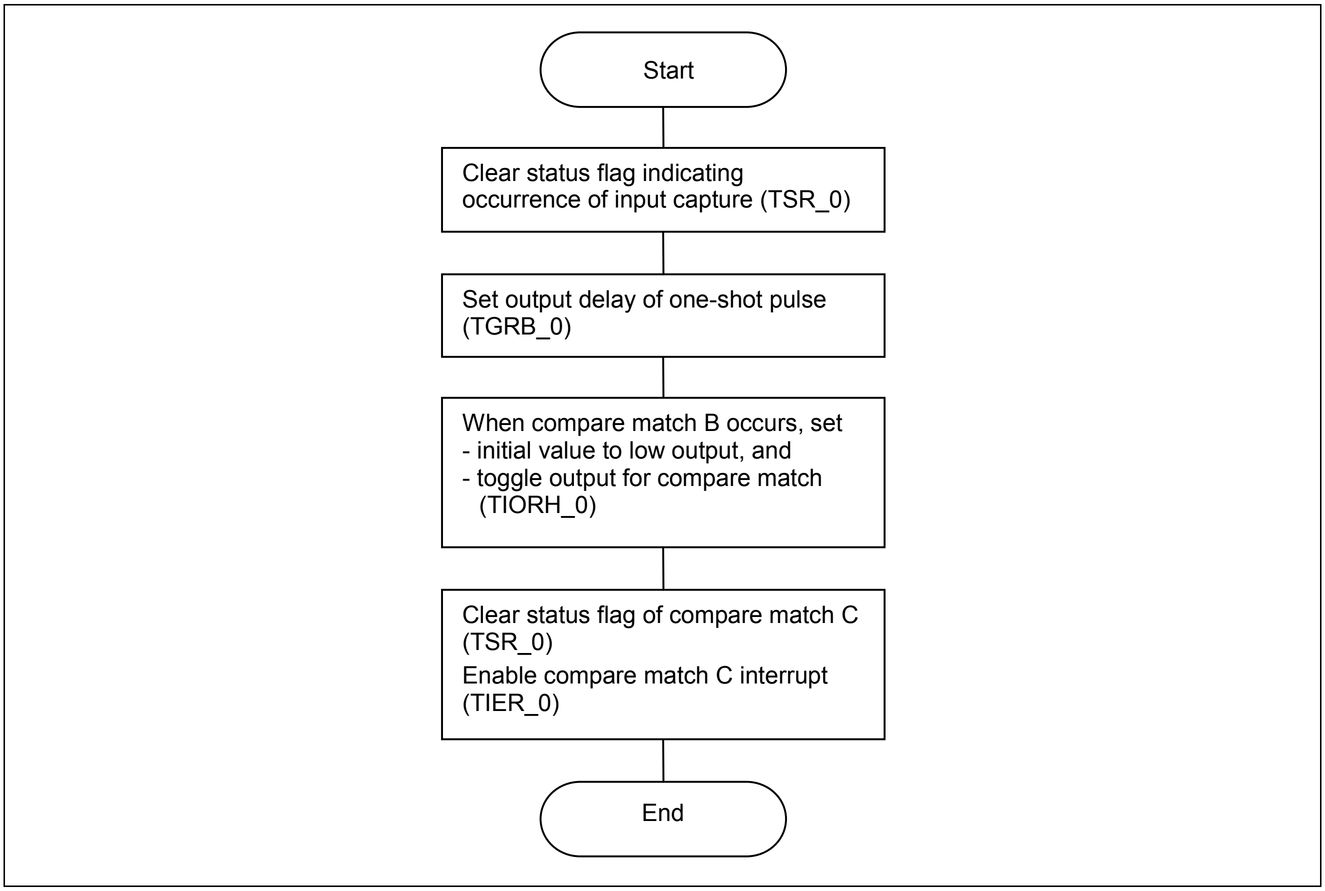
3.3 Flowchart

A processing flow is shown below.

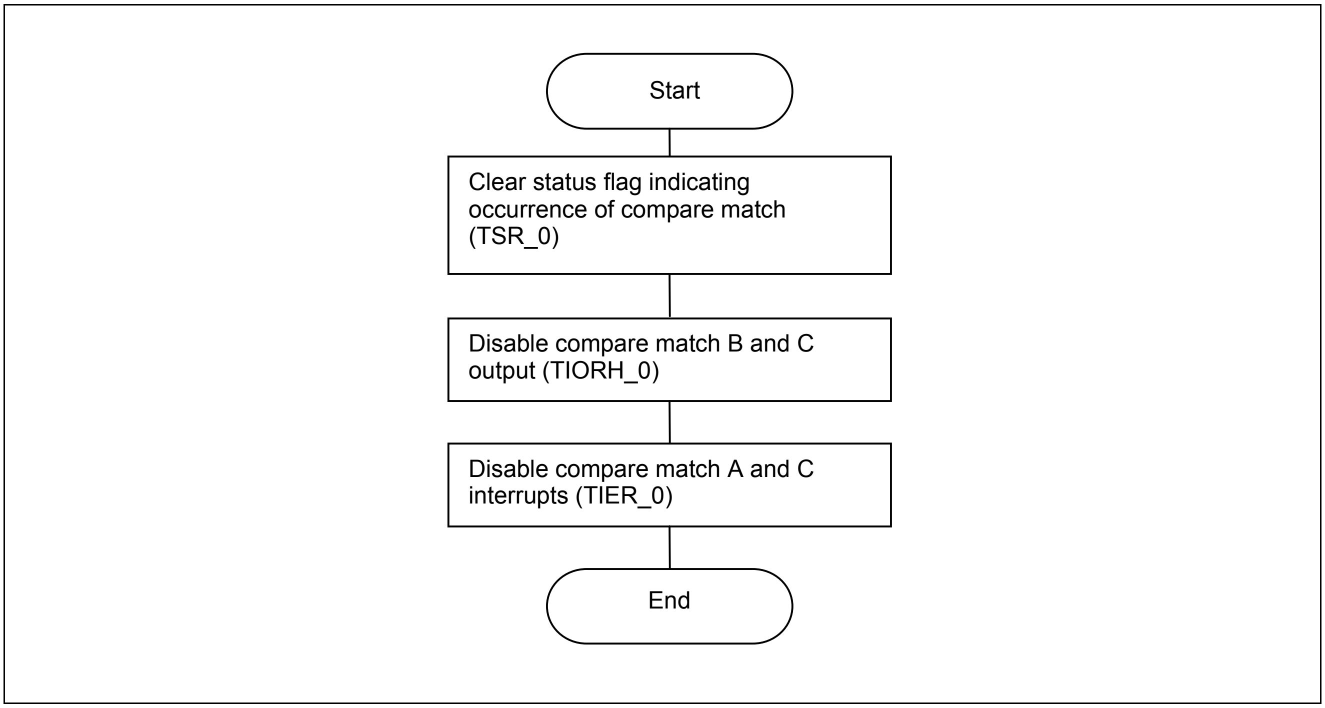
(1) void oneshot_set (unsigned short, unsigned short)



(2) void inthdr_captureA (void)



(3) void inthdr_compareC (void)



3.4 Program Listing

A source program listing is shown below. In this source program, Renesas's standard definition (file automatically generated by High-performance Embedded Workshop: iodefine.h) is used for I/O register structure definition. If you want to use a user-specified definition, change the I/O register structure in the sample program.

```
/****************************************************************/
/* include file                                                 */
/****************************************************************/
#include <machine.h>
#include "iodefine.h"

/****************************************************************/
/* function prototype                                           */
/****************************************************************/
void oneshot_set( unsigned short, unsigned short );

/****************************************************************/
/* static variable                                              */
/****************************************************************/
static unsigned short save_delay_count;

/****************************************************************/
/* function definition                                          */
/****************************************************************/
void  oneshot_set( unsigned short delay_count,
                unsigned short high_count )
{
```

```
    P_MSTPCRA.BIT.MSTPA0 = 0;    // reset module-standby for TPU
    P_TPU0.TCR.BIT.TPSC  = 0;    // set TPU0 countup clock source
    P_TPU0.TCR.BIT.CCLR  = 1;    // set TPU0 counter clear cause
    P_TPU0.TIOR.BIT.IOA  = 9;    // set TPU0 capture-edge
    P_TPU0.TIOR.BIT.IOC  = 0;    // set TPU0 compare-match-C
    P_TPU0.TGRD = (unsigned int)(delay_count+high_count);
    P_TPU0.TGRC = P_TPU0.TGRD+50;
    save_delay_count = delay_count;
    P_P3.ICR.BIT.Pn0ICR   = 1;   // set input buffer enable
    P_TPU0.TMDR.BIT.BFB   = 1;   // set TPU0 TGRB buffer-mode
    P_TPU0.TIER.BIT.TGIEA = 1;   // set TGI0A-interrupt enable
    set_imask_ccr(0);            // clear interrupt mask
    P_TPU.TSTR.BIT.CST0   = 1;   // start TPU0
}

/****************************************************************/
/* interrupt handler definition                                 */
/****************************************************************/
#pragma interrupt ( inthdr_captureA )
void inthdr_captureA( void )
{
    volatile unsigned char  dummy;
    dummy = P_TPU0.TSR.BYTE;      // read TPU0 interrupt status
    P_TPU0.TSR.BIT.TGFA = 0;      // clear TGI0A-interrupt status
    P_TPU0.TSR.BIT.TGFC = 0;      // clear TGI0C-interrupt status
    P_TPU0.TGRB =save_delay_count;// set TPU0 compare-match-B
    P_TPU0.TIOR.BIT.IOB   = 3;
    P_TPU0.TIER.BIT.TGIEC = 1;    // set TGI0C-interrupt enable
}

#pragma interrupt ( inthdr_compareC )
void inthdr_compareC( void )
{
    volatile unsigned char  dummy;
    dummy = P_TPU0.TSR.BYTE;     // read TPU0 interrupt status
    P_TPU0.TSR.BIT.TGFC   = 0;   // clear TGI0C-interrupt status
    P_TPU0.TIOR.BIT.IOB   = 0;   // reset TPU0 compare-match-B
    P_TPU0.TIOR.BIT.IOC   = 0;   // reset TPU0 compare-match-C
    P_TPU0.TIER.BIT.TGIEC = 0;   // set TGI0C-interrupt disable
    P_TPU0.TIER.BIT.TGIEA = 0;   // set TGI0A-interrupt disable
}
```

Revision Record

Rev.	Date	Description	
		Page	Summary
1.00	Sept.19.03	—	First edition issued

Keep safety first in your circuit designs!

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

Notes regarding these materials

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