To our customers,

# Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

# Notice

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

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

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

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

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

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

# H8/300H Tiny Series

## Buffer Operation of Timer Z Input Capture Function

### Introduction

The buffer operation of the input-capture function supported by timer Z is used to measure the high-level width and low-level width of a pulse.

### Target Device

H8/3687

### Contents

## 1. Specifications

1. As shown in figure 1.1, the buffer function of timer Z input-capture function is used to measure the high-level width and low-level width of a pulse input to the input-capture pin A0 (FTIOA0).
2. The timer counter 0 (TCNT0) measures the time from rising edge to falling edge of the pulse to measure the high-level width of the pulse.
3. The timer counter 0 (TCNT0) measures the time from falling edge to rising edge of the pulse to measure the low-level width of the pulse.
4. The maximum width of a pulse that can be measured is 32.768 ms with the accuracy of ±0.5 µs.

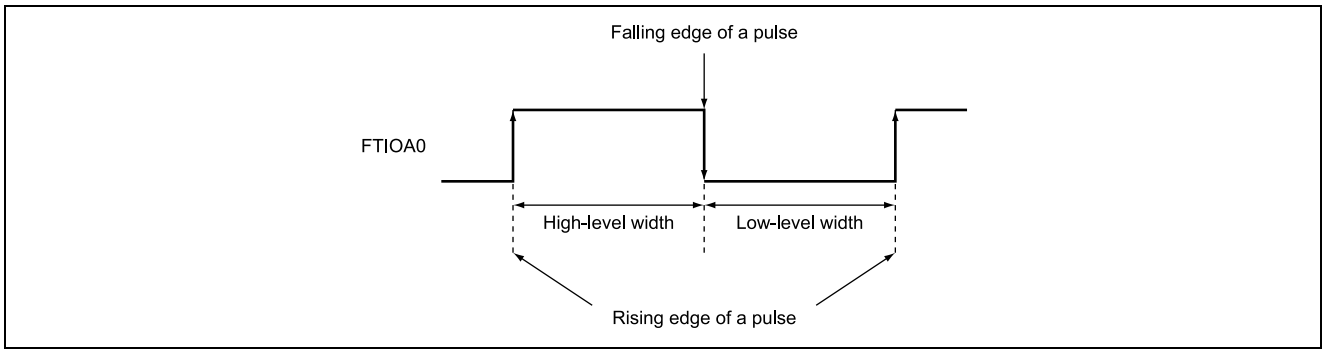



**Figure 1.1 Measurement of Input Pulse Width**

## 2. Description of Functions

1. In this sample task, the high-level width and low-level width of the pulse input to the input-capture input pin A0 (FTIOA0) is measured by using the buffer operation of timer Z input-capture function.
   Figure 2.1 is a block diagram of timer Z input-capture function. The elements of the block diagram are described below.

- The system clock (ϕ) is a 16-MHz clock that is used as a reference clock for operating the CPU and peripheral functions.
- Prescaler S (PSS) is a 13-bit counter with clock input of ϕ. PSS is incremented every cycle.
- Timer control register 0 (TCR0) selects input clock and TCNT0 clearing method. In this sample task, the input clock is specified as ϕ/8, the TCNT0 counts a the rising edge of ϕ/8, and the TCNT0 is specified to be cleared on compare-match/input-capture with GRA0.
- Timer I/O control register A0 (TIORA0) controls GRA0 and GRB0. In this sample task, GRA0 is used as an input-capture register, and the TCNT0 value is transferred to the GRA0 at the rising and falling edges of the FTIOA0 pin.
- Timer I/O control register C0 (TIORC0) controls the GRC0 and GRD0. In this sample task, the GRC0 is specified as an input-capture register.
- Timer status register 0 (TSR0) indicates the timer Z status. In this sample task, the overflow flag (OVF) is set to 1 when TCNT0 overflows and the input capture/compare-match flag A (IMFA) are set to one when a GRA0 input capture occurs.
- Timer interrupt enable register (TIER0) enables or disables various interrupt requests. In this sample task, interrupts by the TSR0 OVF and IMFA flag are enabled and other interrupts are disabled.
- Timer counter 0 (TCNT0) is a 16-bit readable/writable upward counter that is incremented by the internal clock or external clock input. In this sample task, TCNT0 is incremented at the rising edge of ϕ/8.
- General register A0 (GRA0) is a 16-bit readable/writable register. In this sample task, GRA0 functions as an input-capture register. The TCNT0 value is transferred to the GRA0 at the rising and falling edges of the FTIOA0 pin.
- General register C0 (GRC0) is a 16-bit readable/writable register. In this sample task, GRC0 functions as a buffer register for GRA0. The content of the GRA0 is transferred to the GRC0 at the rising and falling edges of the FTIOA0 pin.

- Timer start register (TSTR) starts or stops the TCNT0 and TCNT1 operation. In this sample task, TCNT0 is specified to start counting and TCNT1 to stop counting.
- Timer mode register (TMDR) selects synchronous or independent operation. In this sample task, GRA0 operates synchronous of GRC0. In this sample task, TCNT0 operates independently of TCNT1.
- Timer function control register (TFCR) specifies operation modes and selects the output level. In this sample task, channels 0 and 1 are specified for normal operation.
- Input-capture/output-compare pin A0 (FTIOA0) is specified as an input-capture input pin and the TCNT0 value is transferred to the GRA0 at the rising edge and falling edge of this pin.

Input pulse's width
= (TCNT0 value stored in plhigh or pllow) × (TCN0T input clock cycle)
= (TCNT0 value stored in plhigh or pllow) × (1/ (ϕ/PSS))
= (TCNT0 value stored in plhigh or pllow) × (1/ (16 MHz/8))
= (TCNT0 value stored in plhigh or pllow) × 0.5 μs

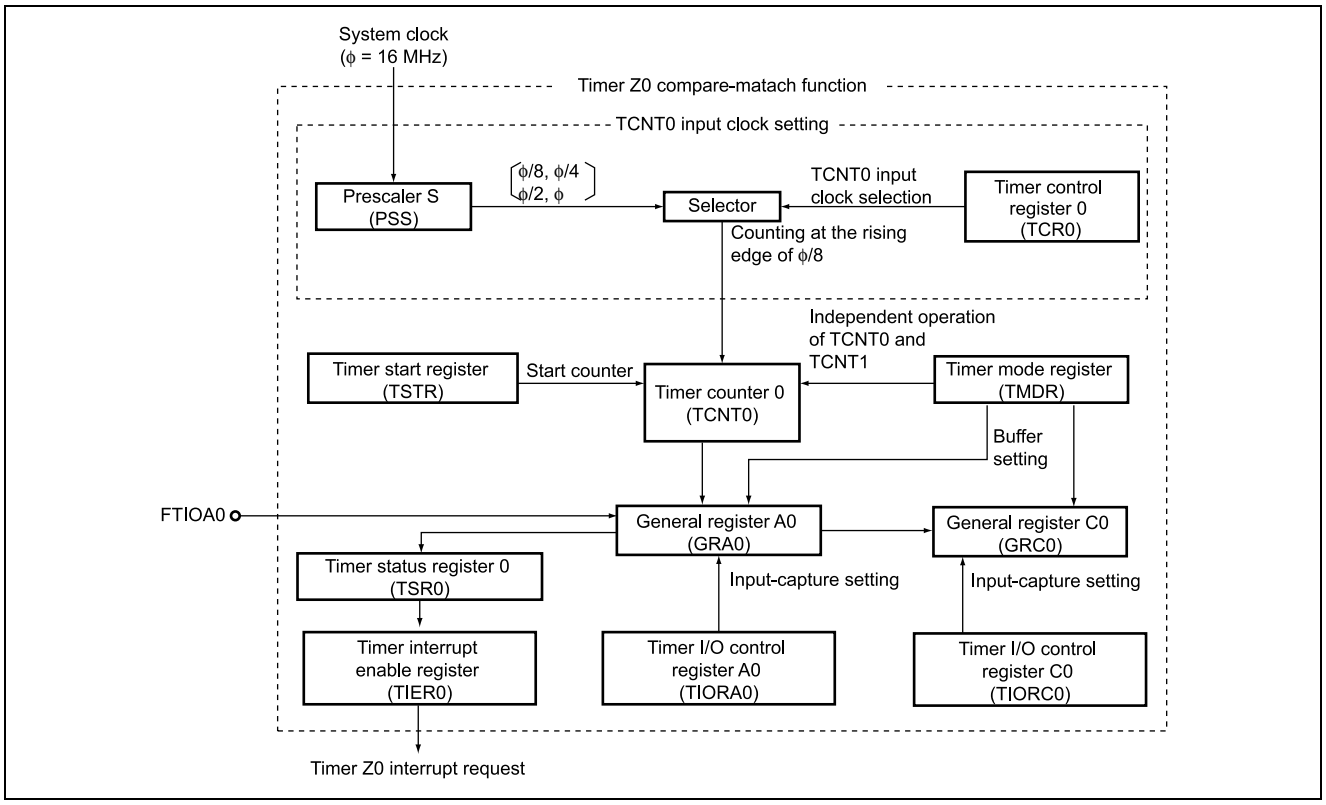



**Figure 2.1 Timer Z0 Block Diagram**

2. Table 2.1 lists the function allocation for this sample task. The functions listed in this table are allocated so that the high-level and low-level widths of the pulse can be measured.

**Table 2.1 Function Allocation**

| Function | Description |
|---|---|
| PSS | 13-bit counter with system clock input |
| TCR0 | Specifies the TCNT0 clearing method and input clock. |
| TIORA0 | Specifies the GRA0 as an input-capture register. |
| TIORC0 | Specifies the GRC0 as an input-capture register. |
| TCNT0 | 16-bit upward counter that is incremented at the rising edge of $\phi$/8. |
| TSR0 | Performs flag control for TNCT0 overflow and GRA0 input capture. |
| TIER0 | Enables TCNT0 overflow and GRA0 input-capture interrupt requests.. |
| GRA0 | The TCNT0 value is transferred to the GRA0 at the rising and falling edges of the FTIOA0 pin. |
| GRC0 | The GRA0 value is transferred to the GRC0 at the rising and falling edges of the FTIOA0 pin. |
| TSTR | Controls TCNT0 count start. |
| TMDR | Sets the GRA0 and GRC0 for buffer operation and selects independent operation of TCNT0 and TCNT1. |
| TFCR | Specifies channels 0 and 1 for normal operation. |
| FTIOA0 pin | Pulse input pin |

## 3. Description of Operation

Figure 3.1 illustrates the operation of this sample task. The hardware and software processing are applied as shown in figure 3.1 to measure the high-level and low-level widths of the input pulse.

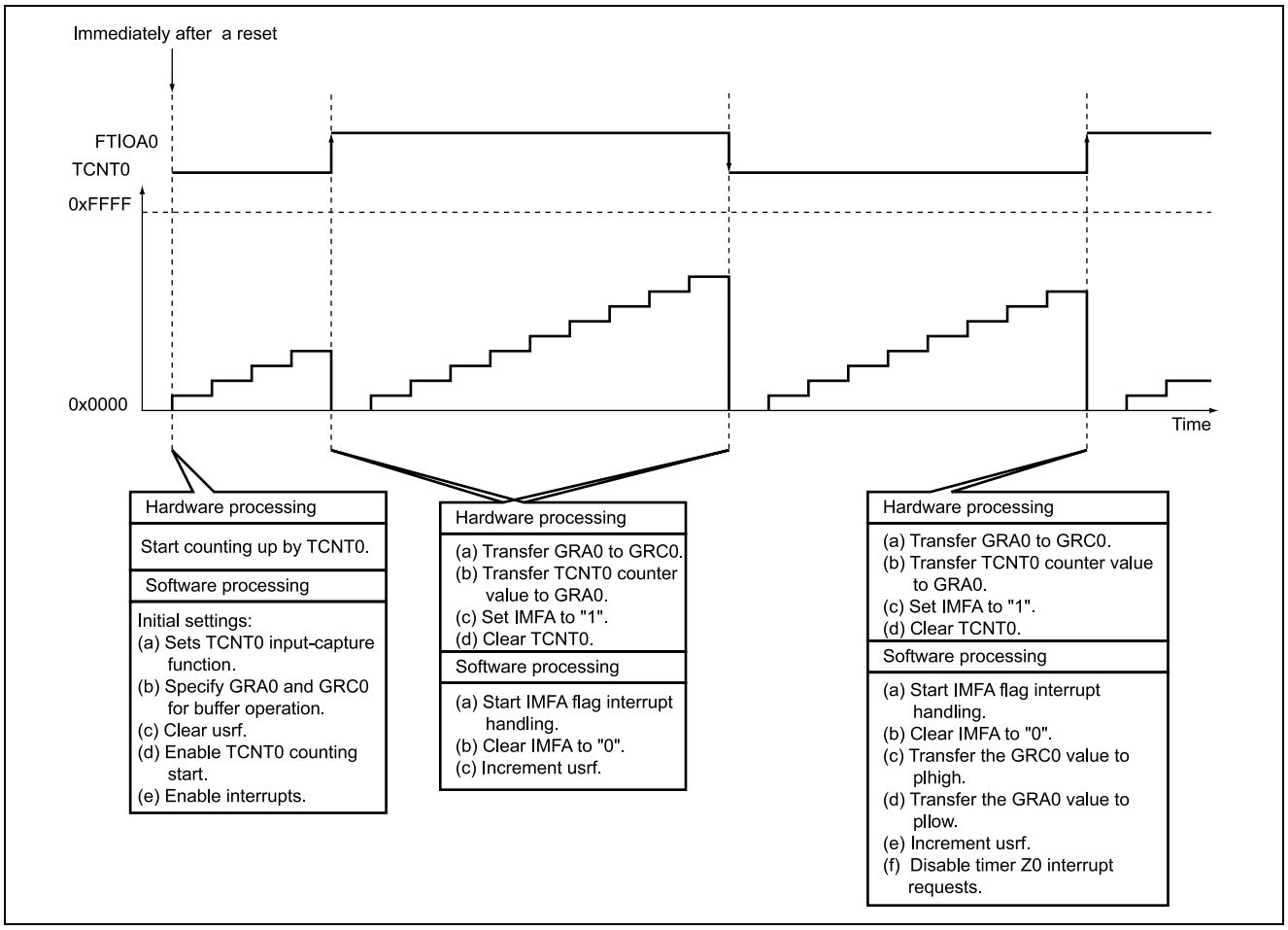



**Figure 3.1 Principle of Operation**

# 4. Description of Software

## 4.1 Modules

Table 4.1 describes the modules used in this sample task.

**Table 4.1 Description of Modules**

| Module Name | Label Name | Function |
|---|---|---|
| Main routine | main | Sets timer Z0 input-capture function and buffer operation, starts counter operation, and specifies interrupts. |
| Pulse measurement end | tz0int | Timer Z0 interrupt handling<br>Clears OVF and IMFA flags, and stores high-level and low-level widths of the pulse in RAM. |

## 4.2 Arguments

This sample task uses no arguments.

## 4.3 Internal Registers

The internal registers used in this sample task are described below.

- TCR0 Timer control register 0 Address: 0xF700

| Bit | Bit Name | Setting | Function |
|---|---|---|---|
| 7 | CCLR2 | CCLR2 = 0 | Counter clear 2 to 0 |
| 6 | CCLR1 | CCLR1 = 0 | CCLR2 = 0, CCLR1 = 0, CCLR0 = 1: |
| 5 | CCLR0 | CCLR0 = 1 | Clears TCNT0 on compare-match/input-capture with GRA0. |
| 4 | CKEG1 | CKEG1 = 0 | Clock edge 1 to 0 |
| 3 | CKEG0 | CKEG0 = 0 | CKEG1 = 0, CKEG0 = 0: Counts at the rising edge of the clock. |
| 2 | TPSC2 | TPSC2 = 0 | Timer prescaler 2 to 0 |
| 1 | TPSC1 | TPSC1 = 1 | TPSC2 = 0, TPSC1 = 1, TPSC0 = 1: Counts by $\phi$/8. |
| 0 | TPSC0 | TPSC0 = 1 | |

- TIORA0 Timer I/O control register A0 Address: 0xF701

| Bit | Bit Name | Setting | Function |
|---|---|---|---|
| 2 | IOA2 | IOA2 = 1 | I/O control A2 to A0 |
| 1 | IOA1 | IOA1 = 1 | IOA2 = 1, IOA1 = 1, IOA0 = X: |
| 0 | IOA0 | IOA0 = 0 | The GRA0 is used as an input-capture register and the TCNT0 value is transferred to the GRA0 at the rising and falling edges of the FTIOA0 pin.<br>(X: don't care) |

- TIORC0 Timer I/O control register C0 Address: 0xF702

| Bit | Bit Name | Setting | Function |
|---|---|---|---|
| 2 | IOC2 | IOC2 = 1 | I/O control C2 to C0 |
| 1 | IOC1 | IOC1 = 1 | IOC2 = 1, IOC1 = 1, IOC0 = X: |
| 0 | IOC0 | IOC0 = 0 | The GRC0 is used as an input-capture register and the GRA0 value is transferred to the GRC0 at the rising and falling edges of the FTIOA0 pin.<br>(X: don't care) |

- TSR0 Timer status register 0 Address: 0xF703

| Bit | Bit Name | Setting | Function |
|---|---|---|---|
| 4 | OVF | 0 | Timer overflow:<br>OVF = 0: Indicates that TCNT0 overflow has not occurred.<br>OVF = 1: Indicates that TCNT0 overflow has occurred. |
| 0 | IMFA | 1 | Input-capture/compare-match flag A:<br>When the GRA0 functions as an input-capture register, IMFA indicates that the TCNT0 value has been transferred to the GRA0 based on an input-capture signal.<br>IMFA = 0: Indicates that the TCNT0 value has not been transferred to GRA0.<br>IMFA = 1: Indicates that the TCNT0 value has been transferred to GRA0. |

- TIER0 Timer interrupt enable register 0 Address: 0xF704

| Bit | Bit Name | Setting | Function |
|---|---|---|---|
| 4 | OVIE | 1 | Timer overflow interrupt enable:<br>OVIE = 0: Disables interrupt requests by the OVF or UDF flag of TSR0.<br>OVIE = 1: Enables an interrupt requested by the OVF or UDF flag of TSR0. |
| 0 | IMIEA | 1 | Input-capture/compare-match interrupt A enable:<br>IMIEA = 0: Disables interrupt requests by the IMFA flag of TSR0.<br>IMIEA = 1: Enables interrupt requests by the IMFA flag of TSR0. |

- TCNT0 Timer counter 0 Address: 0xF706
  Function: A 16-bit upward counter that is incremented at the rising edge of $\phi/8$.
  Setting: 0x0000

- GRA0 General register A0 Address: 0xF708
  Function: During input capture operation, the TCNT0 value is transferred to this register at the rising and falling edges of the FTIOA0 pin.
  Setting: —

- GRC0 General register C0 Address: 0xF70C
  Function: When the GRC0 functions as the buffer register for the GRA0 in input capture operation, the GRA0 value is transferred to the GRC0 at the rising and falling edges of the FTIOA0 pin.
  Setting: —

- TSTR Timer start register Address: 0xF720

| Bit | Bit Name | Setting | Function |
|---|---|---|---|
| 0 | STR0 | 0 | Channel 0 counter start<br>STR0 = 0: Stops counting by TCNT0.<br>STR0 = 1: Starts counting by TCNT0. |

- TMDR Timer mode register Address: 0xF721

| Bit | Bit Name | Setting | Function |
|---|---|---|---|
| 4 | BFC0 | 1 | Buffer operation C<br>BFC0 = 0: Specifies GRC0 for normal operation.<br>BFC0 = 1: Specifies GRA0 and GRC0 for buffer operation. |
| 0 | SYNC | 0 | Timer synchronization<br>SYNC = 0: TCNT0 operates independently of TCNT1.<br>SYNC = 1: TCNT0 operates synchronously with TCNT1. |

- TFCR Timer function control register Address: 0xF723

| Bit | Bit Name | Setting | Function |
|---|---|---|---|
| 1 | CMD1 | CMD1 = 0 | Combination mode 1 to 0 |
| 0 | CMD0 | CMD0 = 0 | CMD1 = 0, CMD0 = 0: Specifies channels 0 and 1 for normal operation. |

## 4.4 Description of RAM

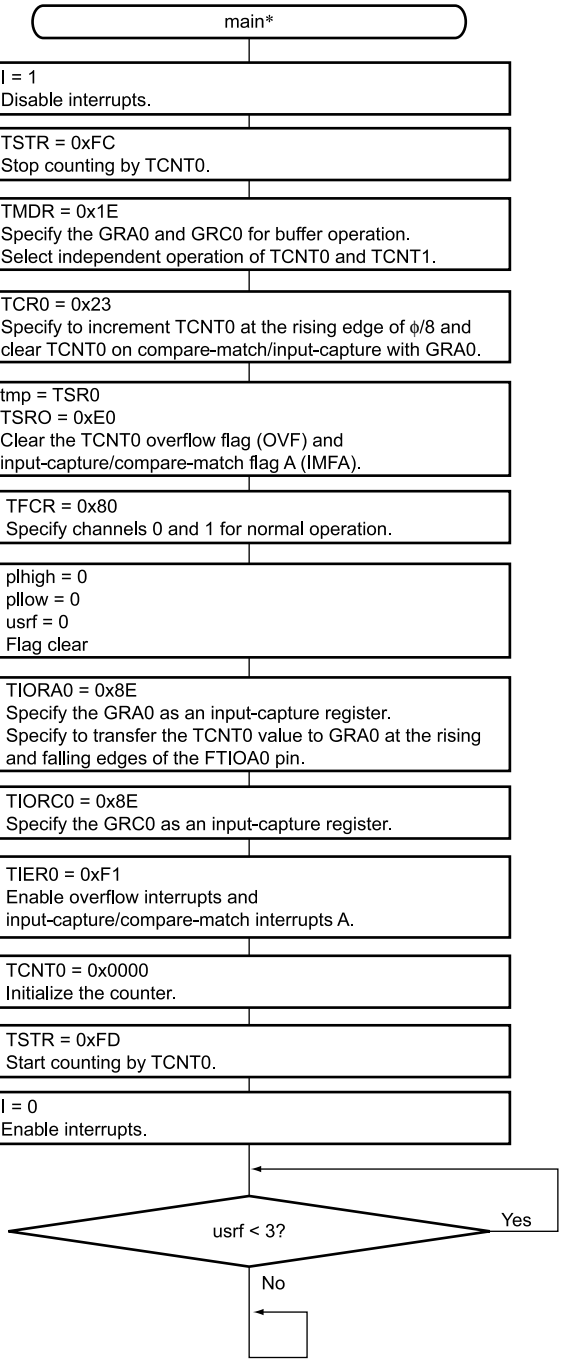
Table 4.2 describes the RAM usage in this sample task.

**Table 4.2 Description of RAM**

| Label Name | Function | Size | Used in |
|---|---|---|---|
| plhigh | Pulse high-level width measurement result | 2 bytes | Main routine<br>Pulse width measurement end |
| pllow | Pulse low-level width measurement result | 2 bytes | Main routine<br>Pulse width measurement end |
| usrf | Timer Z interrupt status indicator<br>usfr = 0: No interrupt has occurred.<br>usfr = 1: Pulse width measurement has started.<br>usfr = 2: Pulse high-level width measurement has completed.<br>usfr = 3: Pulse low-level width measurement has completed/ Pulse width measurement has completed normally. | 1 byte | Main routine<br>Pulse width measurement end |

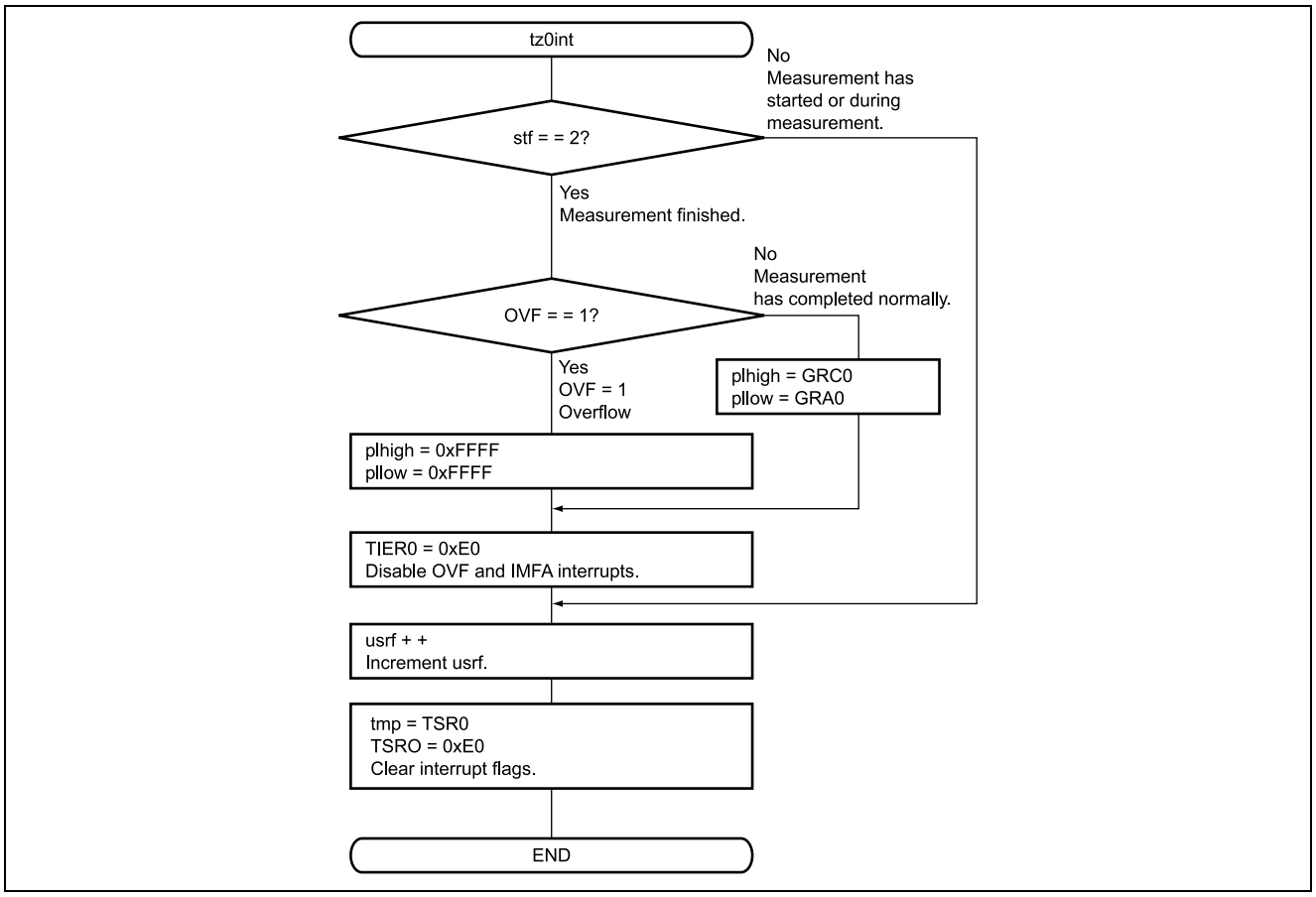
## 5. Flowcharts

1. Main routine




Note: * The stack pointer is specified by INIT.SRC (assembly language).

2. Pulse width measurement end



## 6. Program Listing

```
/************************************************************************************************************************/
/*                                                                                                                      */
/*   H8/300HN Series -H8/3687-                                                                                          */
/*   Application Note                                                                                                   */
/*                                                                                                                      */
/*   'Pulse Period Measurement by Input Caputure Function'                                                              */
/*                                                                                                                      */
/*   Function                                                                                                           */
/*   : Timer Z Input Caputure                                                                                           */
/*                                                                                                                      */
/*   External Clock :    16MHz                                                                                          */
/*   Internal Clock :    16MHz                                                                                          */
/*   Sub Clock :         32.768kHz                                                                                      */
/*                                                                                                                      */
/************************************************************************************************************************/

#include    <machine.h>

/************************************************************************************************************************/
/*   Symbol Definition                                                                                                  */
/************************************************************************************************************************/
struct BIT {
     unsigned char   b7:1;         /* bit7 */
     unsigned char   b6:1;         /* bit6 */
     unsigned char   b5:1;         /* bit5 */
     unsigned char   b4:1;         /* bit4 */
     unsigned char   b3:1;         /* bit3 */
     unsigned char   b2:1;         /* bit2 */
     unsigned char   b1:1;         /* bit1 */
     unsigned char   b0:1;         /* bit0 */
};

#define        TCR0           *(volatile unsigned char *)0xF700                /* Timer control register_0                */
#define        TIORA0         *(volatile unsigned char *)0xF701                /* Timer I/O Control Register A_0          */
#define        TIORC0         *(volatile unsigned char *)0xF702                /* Timer I/O Control Register C_0          */
#define        TSR0           *(volatile unsigned char *)0xF703                /* Timer status register_0                 */
#define        TSR0_BIT       (*(struct BIT *)0xF703)                          /* Timer status register_0                 */
#define        OVF            TSR0_BIT.b4                                      /* Over flow flag                          */
#define        IMFA           TSR0_BIT.b0                                      /* Input Capture/Compare Match FlagA       */
#define        TIER0          *(volatile unsigned char *)0xF704                /* Timer interrupt enable register0        */
#define        TIER0_BIT      (*(struct BIT *)0xF704)                          /* Timer interrupt enable register0        */
#define        IMIEA          TIER0_BIT.b0                                     /* Input Capture/Compare Match             */
                                                                               /* Interrupt Enable A                      */
#define        TCNT0          *(volatile unsigned short *)0xF706               /* Timer counter_0                         */
#define        GRA0           *(volatile unsigned short *)0xF708               /* General register A_0                    */
#define        GRC0           *(volatile unsigned short *)0xF70C               /* General register C_0                    */
#define        TSTR           *(volatile unsigned char *)0xF720                /* Timer start register                    */
#define        TMDR           *(volatile unsigned char *)0xF721                /* Timer mode register                     */
#define        TPMR           *(volatile unsigned char *)0xF722                /* Timer PWM mode register                 */
#define        TFCR           *(volatile unsigned char *)0xF723                /* Timer function control register         */
#define        TOER           *(volatile unsigned char *)0xF724                /* Timer output master enable register     */
#define        TOCR           *(volatile unsigned char *)0xF725                /* Timer output control register           */

#pragma interrupt   (tz0int)
```

```
/************************************************************************************************************************/
/*   Function define                                                                                                    */
/************************************************************************************************************************/
extern void INIT ( void );                                          /* SP Set                                           */
void main ( void );
void tz0int ( void );

/************************************************************************************************************************/
/*   RAM define                                                                                                         */
/************************************************************************************************************************/
volatile unsigned short plhigh;                                     /* Pulse time data                                  */
volatile unsigned short pllow;                                      /* Pulse time data                                  */
volatile unsigned char  usrf;                                       /* User flag                                        */

/************************************************************************************************************************/
/*   Vector Address                                                                                                     */
/************************************************************************************************************************/
#pragma section     V1                                              /* VECTOR SECTOIN SET                               */
void (*const VEC_TBL1[])(void) = {                                  /* 0x00 - 0x0f                                      */
     INIT                                                           /* 00 Reset                                         */
};
#pragma section     V2                                              /* VECTOR SECTOIN SET                               */
void (*const VEC_TBL2[])(void) = {
     tz0int                                                         /* 34 Timer Z0 Interrupt                            */
};

#pragma section                                                     /* P                                                */
/************************************************************************************************************************/
/*   Main Program                                                                                                       */
/************************************************************************************************************************/
void main ( void )
{
     unsigned char tmp;

     set_imask_ccr(1);                                              /* Interrupt Disable                                */

     TSTR = 0xFC;                                                   /* TCNT0 count stop                                 */
     TMDR = 0x1E;                                                   /* TCNT0,TCNT1 Single Mode                          */
                                                                    /* GRC0 is used buffer of GRA0                      */
     TCR0 = 0x23;                                                   /* Rising edge, phi/2 Clock count                   */
     tmp = TSR0;
     TSR0 = 0xE0;                                                   /* Interrupt Flag Clear                             */
     TFCR = 0x80;                                                   /* Channel 0,1 operate normally                     */
     plhigh = 0;                                                    /* Ram clear                                        */
     pllow = 0;                                                     /* Ram clear                                        */
     usrf = 0;                                                      /* Flag clear                                       */

     TIORA0 = 0x8E;                                                 /* Input capture to GRA0 at both                    */
                                                                    /* rising and falling edges                         */
     TIORC0 = 0x8E;                                                 /* Input capture to GRC0                            */
     TIER0 = 0xF1;                                                  /* IMFA Interrupt Enable                            */
     TCNT0 = 0x0000;                                                /* Clear TCNT0                                      */
     TSTR = 0xFD;                                                   /* TCNT0 count start                                */

     set_imask_ccr(0);                                              /* Interrupt Enable                                 */

     while(usrf < 3);

     while(1);
}
```

```
/************************************************************************************************************************/
/*   Timer Z0 Interrupt                                                                                                 */
/************************************************************************************************************************/
void tz0int ( void )
{
     unsigned char tmp;

     if(usrf == 2){
          if(OVF == 1){
               plhigh = 0xFFFF;                                   /* Overflow                          */
               pllow = 0xFFFF;
          }
          else{
               plhigh = GRC0;                                     /* Ram copy to GRC0                  */
               pllow = GRA0;                                      /* Ram copy to GRA0                  */
          }
          TIER0 = 0xE0;                                           /* OVF,IMFA Interrupt Disable        */
     }

     usrf++;                                                      /* User flag increment               */
     tmp = TSR0;
     TSR0 = 0xE0;                                                 /* Interrupt Flag Clear              */
}
```

### Link address specifications

| Section Name | Address |
|---|---|
| CV1 | 0x0000 |
| CV2 | 0x0034 |
| P | 0x0100 |
| B | 0xFB80 |

## Revision Record

| | | Description | |
|---|---|---|---|
| Rev. | Date | Page | Summary |
| 1.00 | Sep.29.03 | — | 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.