M16C/26

Using the M16C/26 Timer in Event Counter Mode

1.0 Abstract

Event counters are useful in automated packaging lines, tachometers, and mechanical equipment monitoring. The event counters on the M16C/26 can be configured to interrupt on a single event as well, creating additional interrupt inputs pins. The following article describes how configure the M16C/26 timers as event counters, referred to as 'Event Counter Mode'.

2.0 Introduction

The Renesas M30262 is a 16-bit MCU based on the M16C/60 series CPU core. The MCU features include up to 64K bytes of Flash ROM, 2K bytes of RAM, and 4K bytes of Virtual EEPROM. The peripheral set includes 10-bit A/D, UARTS, Timers, DMA, and GPIO. The MCU has eight timers that consists of five Timer A's and three Timer B's. All 8 timers can operate in 'Event Counter Mode'.

Timer A also has the following additional modes of operation:

- Timer Mode
- PWM Mode
- One-Shot Mode

Timer B has the following additional modes of operation:

- Timer Mode
- Pulse Width Measurement Mode

Figure 1 and Figure 2 shows the block diagrams for timers A and B. Note that there are some differences between the two timers but both operate similar in Event Counter Mode. The remainder of this document will focus on setting up timer A0 in Event Counter Mode.

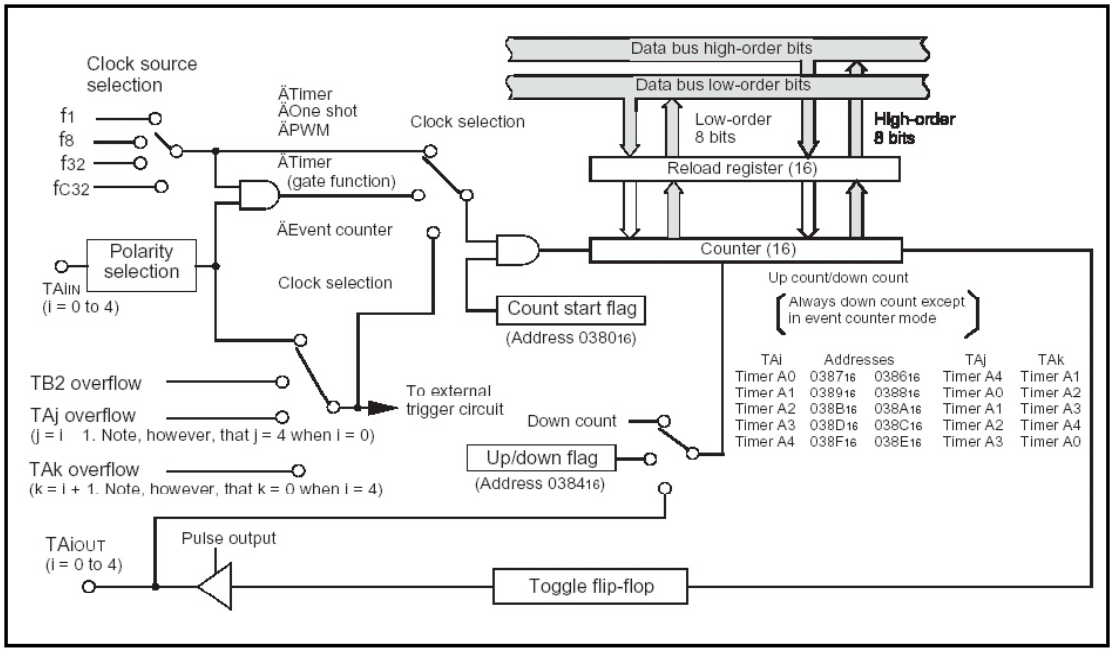


Figure 1 Bock Diagram of Timer A

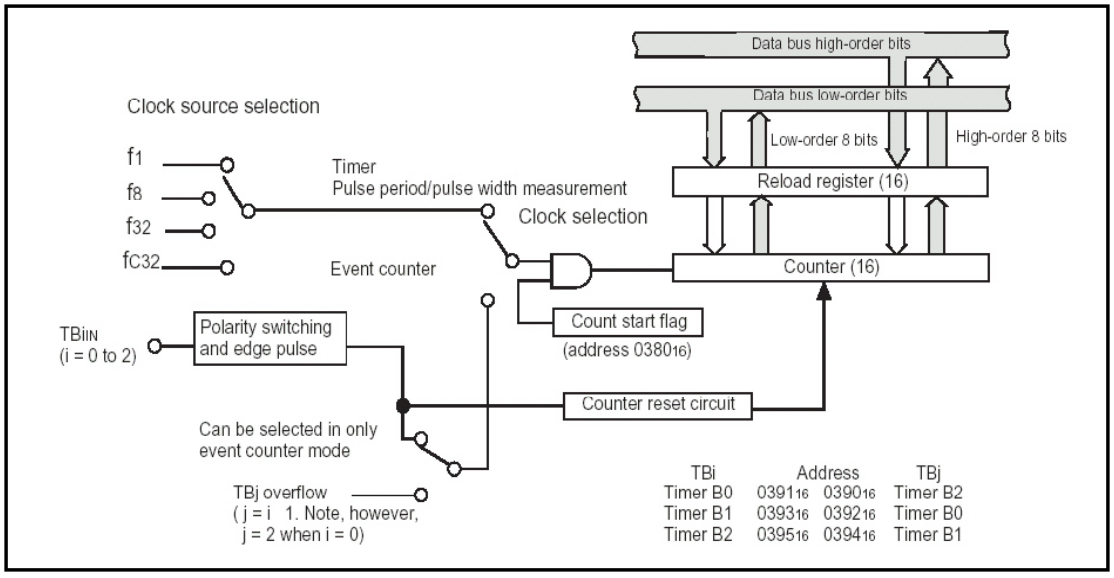


Figure 2 Block Diagram of Timer B

3.0 Event Counter Mode Description

In general, timers TAi or TBi register counts an input signal and at any time, the count value can be read. When the timer overflows (up count) or underflows (down count), the timer interrupt request bit is set and an interrupt is generated if the timer interrupt priority level is set above the current CPU priority level (if the I flag in the CPU flag registers is cleared, the interrupt will not be serviced until the flag is set). If at any time during counting the count start flag is cleared, counting is suspended until set. This is illustrated in Figure 3.

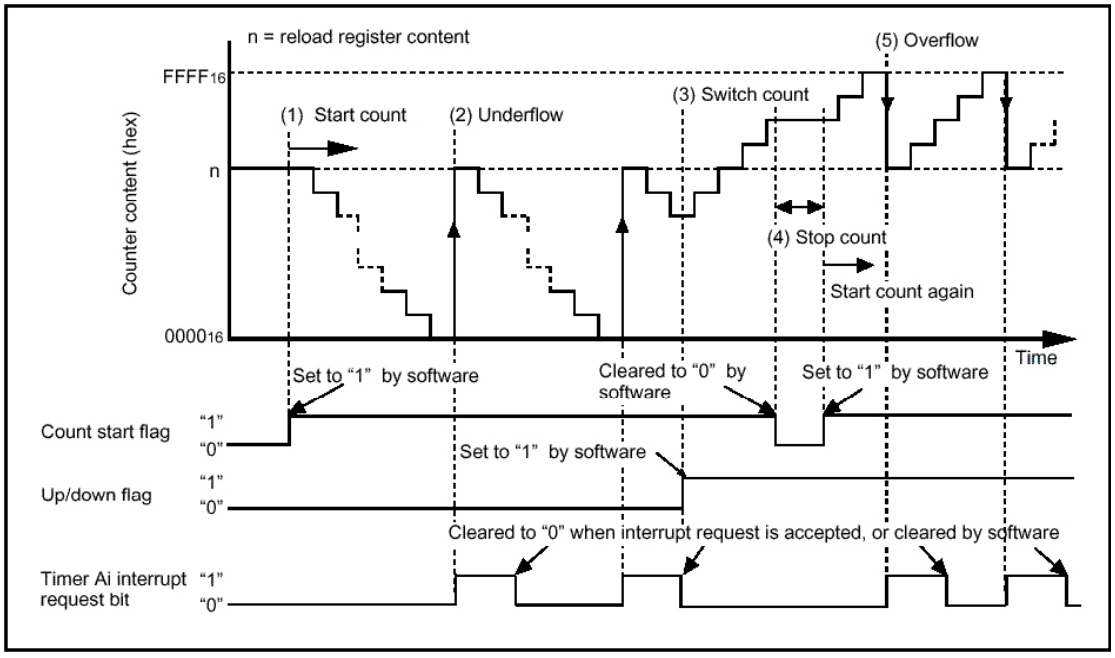


Figure 3 Timing of event counter mode, reload type selected

Besides having the option of counting up or down, the Event Counter Mode has many other options such as count source (TAiIN or TBiIN input pin or another timer), reload or free running type, etc. These options vary depending on which timer is used. The options and the timers they are associated with are summarized in Table 1, Table 2, and Table 3.

Table 1 Timer A specification in Event Counter Mode

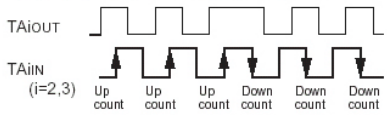
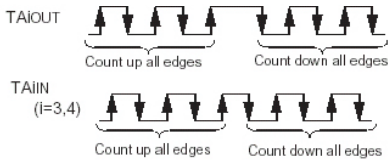
(When not processing two-phase pulse signal)

Item	Specification
Count source	• External signals input to TAiIN pin (effective edge can be selected by software) • TB2 overflow, TAj overflow
Count operation	• Up count or down count can be selected by external signal or software • When the timer overflows or underflows, it reloads the reload register con tents before continuing counting (Note)
Divide ratio	1/ (FFFF16 - n + 1) for up count 1/ (n + 1) for down count n : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	The timer overflows or underflows
TAiIN pin function	Programmable I/O port or count source input
TAiOUT pin function	Programmable I/O port, pulse output, or up/down count select input
Read from timer	Count value can be read out by reading timer Ai register
Write to timer	• When counting stopped When a value is written to timer Ai register, it is written to both reload register and counter • When counting in progress When a value is written to timer Ai register, it is written to only reload register (Transferred to counter at next reload time)
Select function	• Free-run count function Even when the timer overflows or underflows, the reload register content is not reloaded to it • Pulse output function Each time the timer overflows or underflows, the TAiOUT pin's polarity is reversed

Note: This does not apply when the free-run function is selected

Table 2 Timer A specifications in Event Counter Mode

(When processing two-phase pulse signal with timers A2, A3, and A4)

Item	Specification
Count source	• Two-phase pulse signals input to TAiIN or TAiOUT pin
Count operation	• Up count or down count can be selected by two-phase pulse signal • When the timer overflows or underflows, the reload register content is reloaded and the timer starts over again (Note 1)
Divide ratio	1/ (FFFF16 - n + 1) for up count 1/ (n + 1) for down count n : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	Timer overflows or underflows
TAiIN pin function	Two-phase pulse input (Set the TAiIN pin correspondent port direction register to "0".)
TAiOUT pin function	Two-phase pulse input (Set the TAiOUT pin correspondent port direction register to "0".)
Read from timer	Count value can be read out by reading timer A2, A3, or A4 register
Write to timer	• When counting stopped When a value is written to timer A2, A3, or A4 register, it is written to both reload register and counter • When counting in progress When a value is written to timer A2, A3, or A4 register, it is written to only reload register. (Transferred to counter at next reload time.)
Select function (Note 2)	• Normal processing operation (timer A2 and timer A3) The timer counts up rising edges or counts down falling edges on the TAiIN pin when input signal on the TAiOUT pin is "H".  • Multiply-by-4 processing operation (timer A3 and timer A4) If the phase relationship is such that the TAiIN pin goes "H" when the input signal on the TAiOUT pin is "H", the timer counts up rising and falling edges on the TAiOUT and TAiIN pins. If the phase relationship is such that the TAiIN pin goes "L" when the input signal on the TAiOUT pin is "H", the timer counts down rising and falling edges on the TAiOUT and TAiIN pins. 

Note: This does not apply when the free-run function is selected.

For Note 2 above, Timer A3 alone can be selected. Timer A2 is fixed to normal processing operation, and timer A4 is fixed to multiply-by-4 processing operation.

Table 3 Timer B specification in Event Counter Mode

Item	Specification
Count source	• External signals input to TBiIN pin • Effective edge of count source can be a rising edge, a falling edge, or falling and rising edges as selected by software
Count operation	• Counts down • When the timer underflows, it reloads the reload register contents before continuing counting
Divide ratio	1/(n+1) n : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	The timer underflows
TBiIN pin function	Count source input
Read from timer	Count value can be read out by reading timer Bi register
Write to timer	• When counting stopped When a value is written to timer Bi register, it is written to both reload register and counter • When counting in progress When a value is written to timer Bi register, it is written to only reload register (Transferred to counter at next reload time)

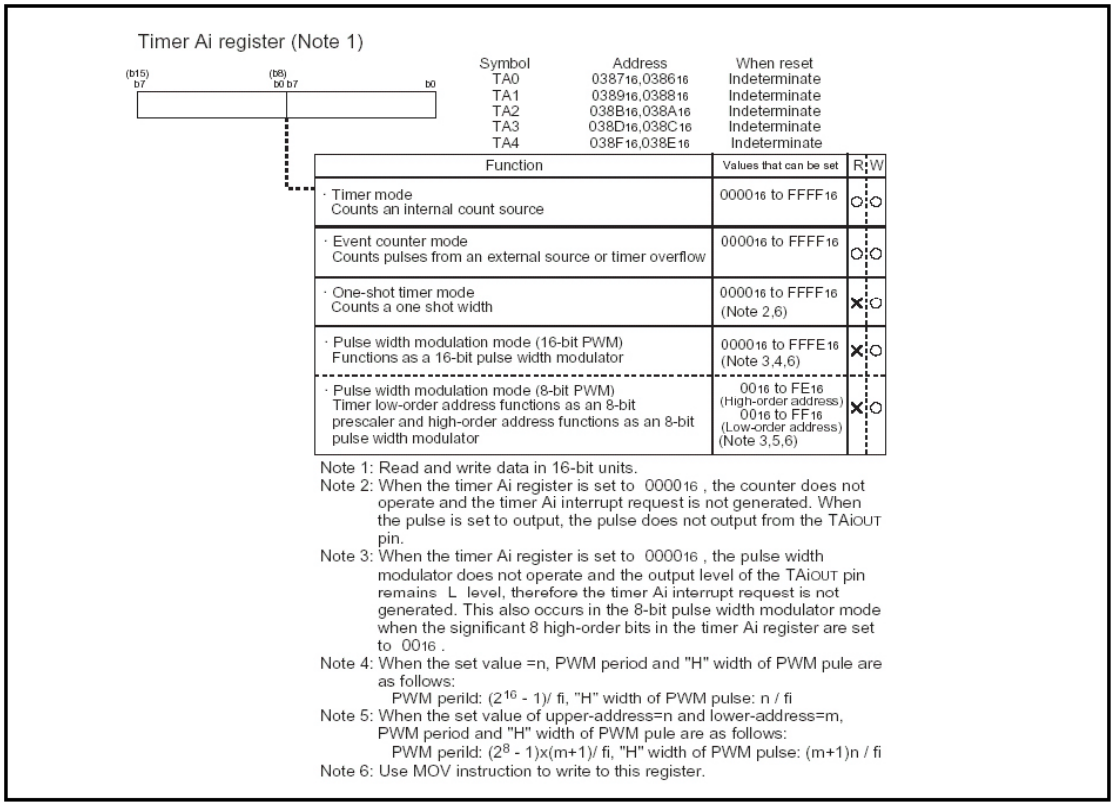
4.0 Configuring Event Counter Mode

The steps to configure a timer for Event Counter Mode are shown below.

1. Load the timer mode register, TAiMR
   a. Select Event Counter Mode: bits TMOD0 = 1, TMOD1 = 0.
   b. Set the remaining bits (MR0, MR1, MR2, TCK0, TCK1) depending on required functions (see mode register diagrams below)
2. Load the TAi or TBi register with the count source.
3. Select the trigger via the TRGSR or ONSF register (N/A for Timer B).
4. Select up or down count via the UDF register (N/A for Timer B, Timer B counts down only).
5. Set the timer ‘interrupt priority level’, TAiIC or TBiIC to at least 1 if required.
6. Enable interrupts (CPU I flag set).
7. Set the ‘start count’ flag bit, TAiS or TBiS in the ‘count start flag’ register, TABSR or TBSR.

For the most part, the order shown above is not important. However, the mode register should be loaded before the ‘start count’ flag is set. Also, the priority level should not be modified when there is a chance of an interrupt occurring.

The required registers are shown in Figure 4 to Figure 12.



Timer Ai register (Note 1)

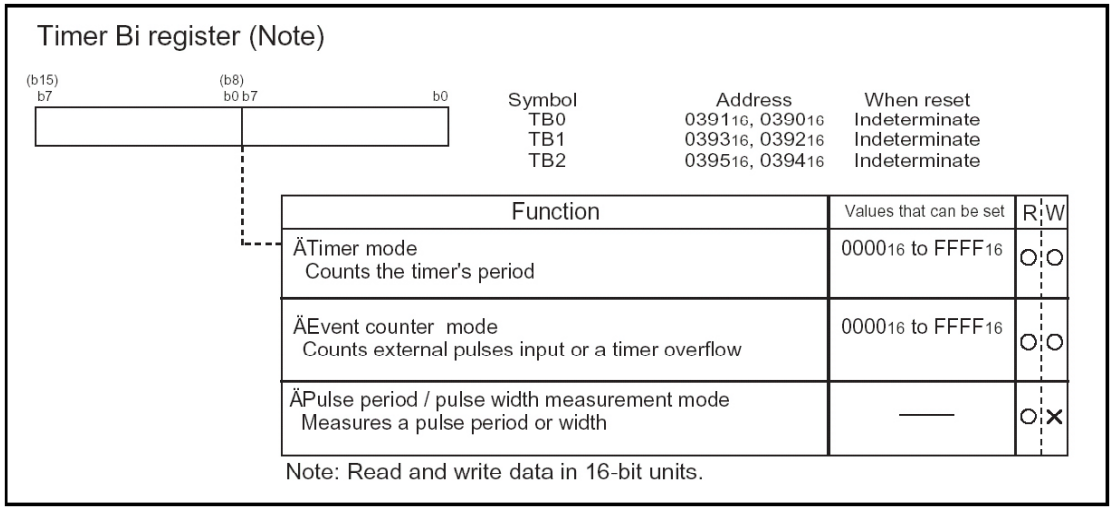
(b15) b7 (b8) b0 b7 b0

Symbol	Address	When reset
TA0	038716,038616	Indeterminate
TA1	038916,038816	Indeterminate
TA2	038B16,038A16	Indeterminate
TA3	038D16,038C16	Indeterminate
TA4	038F16,038E16	Indeterminate

Function	Values that can be set	R	W
· Timer mode Counts an internal count source	000016 to FFFF16	O	O
· Event counter mode Counts pulses from an external source or timer overflow	000016 to FFFF16	O	O
· One-shot timer mode Counts a one shot width	000016 to FFFF16 (Note 2,6)	X	O
· Pulse width modulation mode (16-bit PWM) Functions as a 16-bit pulse width modulator	000016 to FFFE16 (Note 3,4,6)	X	O
· Pulse width modulation mode (8-bit PWM) Timer low-order address functions as an 8-bit prescaler and high-order address functions as an 8-bit pulse width modulator	0016 to FE16 (High-order address) 0016 to FF16 (Low-order address) (Note 3,5,6)	X	O

Note 1: Read and write data in 16-bit units.
Note 2: When the timer Ai register is set to 000016 , the counter does not operate and the timer Ai interrupt request is not generated. When the pulse is set to output, the pulse does not output from the TAiOUT pin.
Note 3: When the timer Ai register is set to 000016 , the pulse width modulator does not operate and the output level of the TAiOUT pin remains L level, therefore the timer Ai interrupt request is not generated. This also occurs in the 8-bit pulse width modulator mode when the significant 8 high-order bits in the timer Ai register are set to 0016 .
Note 4: When the set value =n, PWM period and "H" width of PWM pule are as follows:
PWM perild: $(2^{16} - 1)$/ fi, "H" width of PWM pulse: n / fi
Note 5: When the set value of upper-address=n and lower-address=m, PWM period and "H" width of PWM pule are as follows:
PWM perild: $(2^{8} - 1)x(m+1)$/ fi, "H" width of PWM pulse: (m+1)n / fi
Note 6: Use MOV instruction to write to this register.

Figure 4 Timer Ai registers



Timer Bi register (Note)

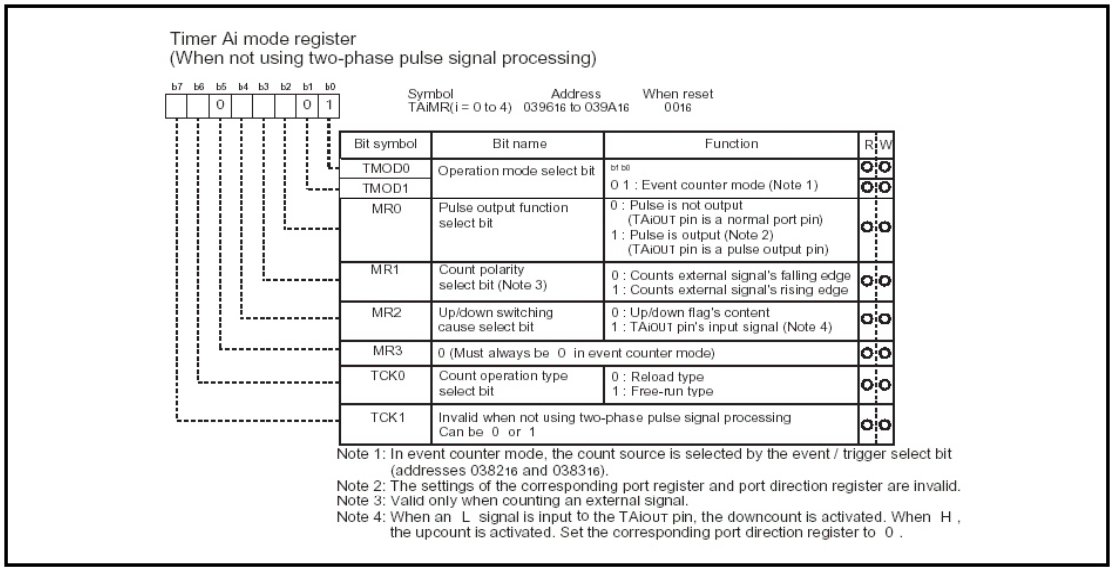
(b15) b7 (b8) b0 b7 b0

Symbol	Address	When reset
TB0	039116, 039016	Indeterminate
TB1	039316, 039216	Indeterminate
TB2	039516, 039416	Indeterminate

Function	Values that can be set	R	W
ÄTimer mode Counts the timer's period	000016 to FFFF16	O	O
ÄEvent counter mode Counts external pulses input or a timer overflow	000016 to FFFF16	O	O
ÄPulse period / pulse width measurement mode Measures a pulse period or width	——	O	X

Note: Read and write data in 16-bit units.

Figure 5 Timer Bi registers



Timer Ai mode register
(When not using two-phase pulse signal processing)

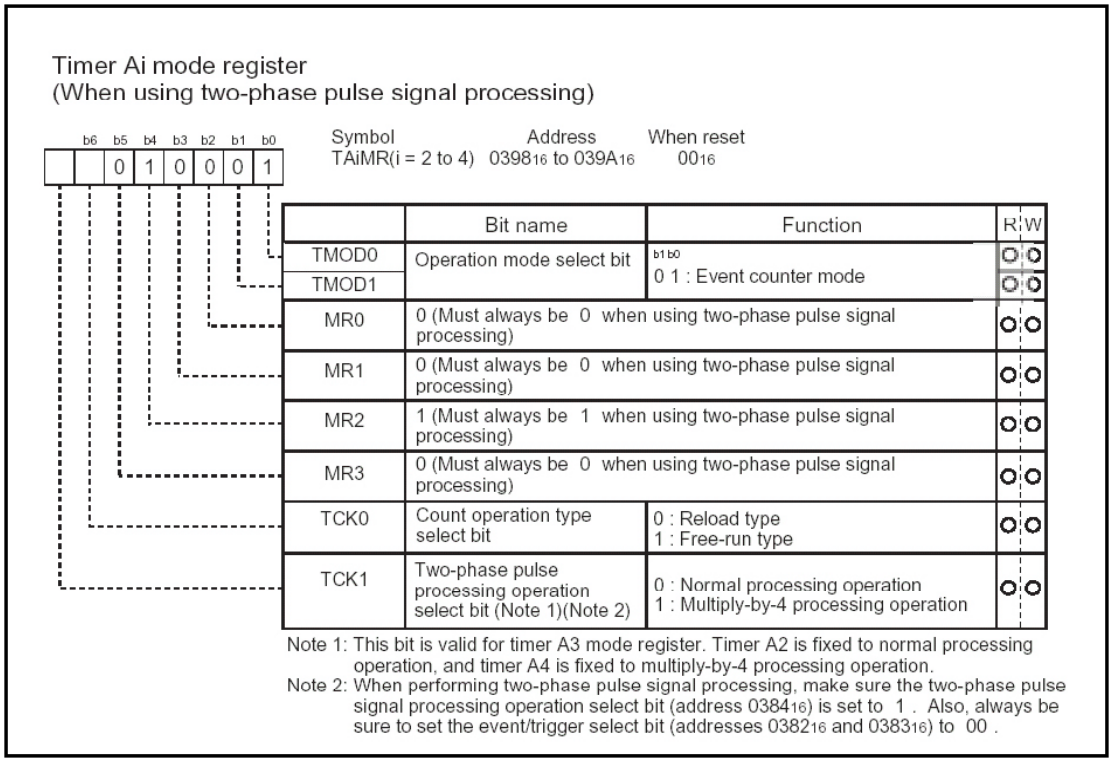
b7	b6	b5	b4	b3	b2	b1	b0
		0				0	1

Symbol: TAiMR(i = 0 to 4) — Address: 039616 to 039A16 — When reset: 0016

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 1 : Event counter mode (Note 1)	○	○
TMOD1			○	○
MR0	Pulse output function select bit	0 : Pulse is not output (TAiOUT pin is a normal port pin) 1 : Pulse is output (Note 2) (TAiOUT pin is a pulse output pin)	○	○
MR1	Count polarity select bit (Note 3)	0 : Counts external signal's falling edge 1 : Counts external signal's rising edge	○	○
MR2	Up/down switching cause select bit	0 : Up/down flag's content 1 : TAiOUT pin's input signal (Note 4)	○	○
MR3	0 (Must always be 0 in event counter mode)		○	○
TCK0	Count operation type select bit	0 : Reload type 1 : Free-run type	○	○
TCK1	Invalid when not using two-phase pulse signal processing Can be 0 or 1		○	○

Note 1: In event counter mode, the count source is selected by the event / trigger select bit (addresses 038216 and 038316).
Note 2: The settings of the corresponding port register and port direction register are invalid.
Note 3: Valid only when counting an external signal.
Note 4: When an L signal is input to the TAiOUT pin, the downcount is activated. When H, the upcount is activated. Set the corresponding port direction register to 0.

Figure 6 Timer Ai mode register in Event Counter Mode



Timer Ai mode register
(When using two-phase pulse signal processing)

b6	b5	b4	b3	b2	b1	b0
	0	1	0	0	0	1

Symbol: TAiMR(i = 2 to 4) — Address: 039816 to 039A16 — When reset: 0016

	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 1 : Event counter mode	○	○
TMOD1			○	○
MR0	0 (Must always be 0 when using two-phase pulse signal processing)		○	○
MR1	0 (Must always be 0 when using two-phase pulse signal processing)		○	○
MR2	1 (Must always be 1 when using two-phase pulse signal processing)		○	○
MR3	0 (Must always be 0 when using two-phase pulse signal processing)		○	○
TCK0	Count operation type select bit	0 : Reload type 1 : Free-run type	○	○
TCK1	Two-phase pulse processing operation select bit (Note 1)(Note 2)	0 : Normal processing operation 1 : Multiply-by-4 processing operation	○	○

Note 1: This bit is valid for timer A3 mode register. Timer A2 is fixed to normal processing operation, and timer A4 is fixed to multiply-by-4 processing operation.
Note 2: When performing two-phase pulse signal processing, make sure the two-phase pulse signal processing operation select bit (address 038416) is set to 1. Also, always be sure to set the event/trigger select bit (addresses 038216 and 038316) to 00.

Figure 7 Timer Ai mode register in Event Counter Mode

(when using 2-phase pulse signals)

Timer Bi mode register

b7	b6	b5	b4	b3	b2	b1	b0
						0	1

Symbol: TBiMR(i=0 to 2) — Address: 039B16 to 039D16 — When reset: 00XX00002

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 1 : Event counter mode	○	○
TMOD1			○	○
MR0	Count polarity select bit (Note 1)	b3 b2 0 0 : Counts external signal's falling edges 0 1 : Counts external signal's rising edges 1 0 : Counts external signal's falling and rising edges 1 1 : Must not be set.	○	○
MR1			○	○
MR2	0 (Must always be 0 in event counter mode; i = 0)		○ (Note 2)	○
	Nothing is assigned (i = 1, 2) In an attempt to write to this bit, write 0 . The value, if read, turns out to be indeterminate.		× (Note 3)	×
MR3	Invalid in event counter mode. In an attempt to write to this bit, write 0 . The value, if read in event counter mode, turns out to be indeterminate.		○	×
TCK0	Invalid in event counter mode. Can be 0 or 1 .		○	○
TCK1	Event clock select	0 : Input from TBiIN pin (Note 4) 1 : TBj overflow (j = i 1; however, j = 2 when i = 0)	○	○

Note 1: Valid only when input from the TBiIN pin is selected as the event clock.
If timer's overflow is selected, this bit can be 0 or 1 .
Note 2: Timer B0.
Note 3: Timer B1, timer B2.
Note 4: Set the corresponding port direction register to 0 .

Figure 8 Timer B mode register in Event Counter Mode

One-shot start flag

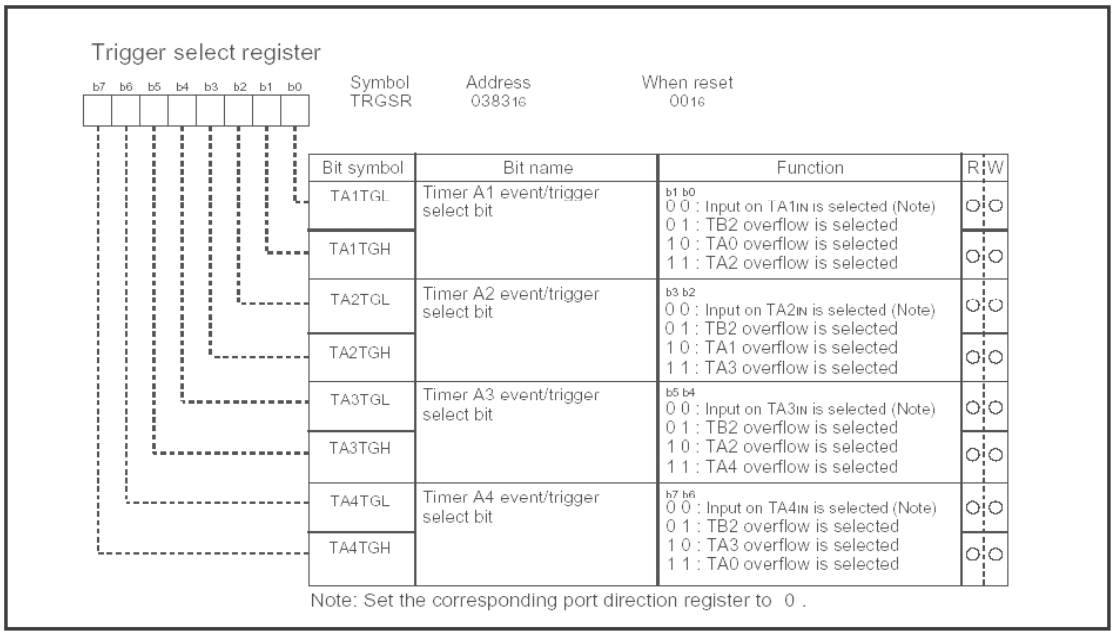
b7	b6	b5	b4	b3	b2	b1	b0
		0					

Symbol: ONSF — Address: 038216 — When reset: 0016

Bit symbol	Bit name	Function	R	W
TA0OS	Timer A0 one-shot start flag	1 : Timer start When read, the value is 0	○	○
TA1OS	Timer A1 one-shot start flag		○	○
TA2OS	Timer A2 one-shot start flag		○	○
TA3OS	Timer A3 one-shot start flag		○	○
TA4OS	Timer A4 one-shot start flag		○	○
Reserved bit		Must always be set to 0	○	○
TA0TGL	Timer A0 event/trigger select bit	b7 b6 0 0 : Input on TA0IN is selected (Note) 0 1 : TB2 overflow is selected 1 0 : TA4 overflow is selected 1 1 : TA1 overflow is selected	○	○
TA0TGH			○	○

Note: Set the corresponding port direction register to 0 .

Figure 9 One Shot Start Flag Register (contains Trigger Select for Timer A0)



Trigger select register

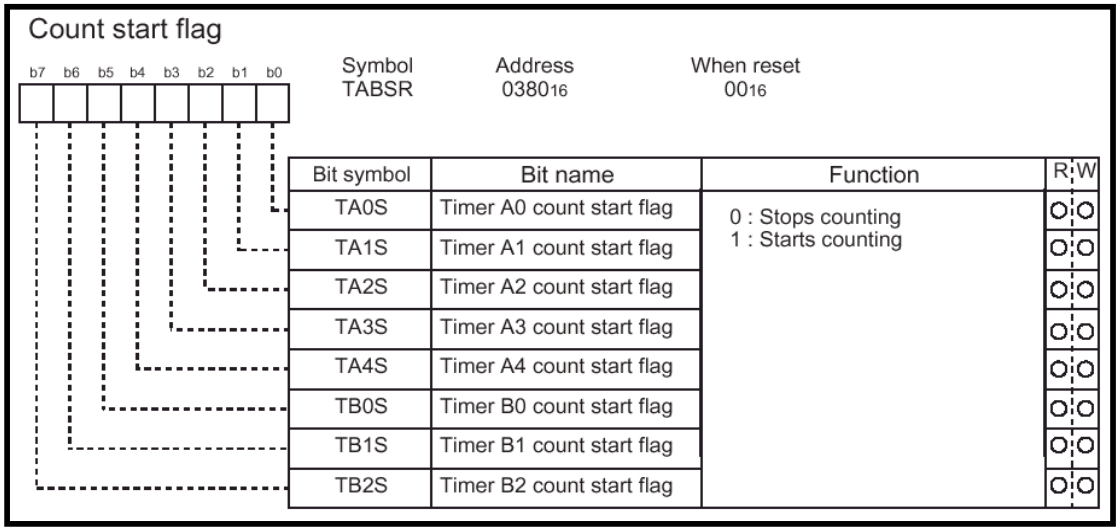
b7	b6	b5	b4	b3	b2	b1	b0

Symbol: TRGSR
Address: 038316
When reset: 0016

Bit symbol	Bit name	Function	R	W
TA1TGL	Timer A1 event/trigger select bit	b1 b0 0 0 : Input on TA1IN is selected (Note) 0 1 : TB2 overflow is selected 1 0 : TA0 overflow is selected 1 1 : TA2 overflow is selected	○	○
TA1TGH			○	○
TA2TGL	Timer A2 event/trigger select bit	b3 b2 0 0 : Input on TA2IN is selected (Note) 0 1 : TB2 overflow is selected 1 0 : TA1 overflow is selected 1 1 : TA3 overflow is selected	○	○
TA2TGH			○	○
TA3TGL	Timer A3 event/trigger select bit	b5 b4 0 0 : Input on TA3IN is selected (Note) 0 1 : TB2 overflow is selected 1 0 : TA2 overflow is selected 1 1 : TA4 overflow is selected	○	○
TA3TGH			○	○
TA4TGL	Timer A4 event/trigger select bit	b7 b6 0 0 : Input on TA4IN is selected (Note) 0 1 : TB2 overflow is selected 1 0 : TA3 overflow is selected 1 1 : TA0 overflow is selected	○	○
TA4TGH			○	○

Note: Set the corresponding port direction register to 0 .

Figure 10 Trigger Select Register



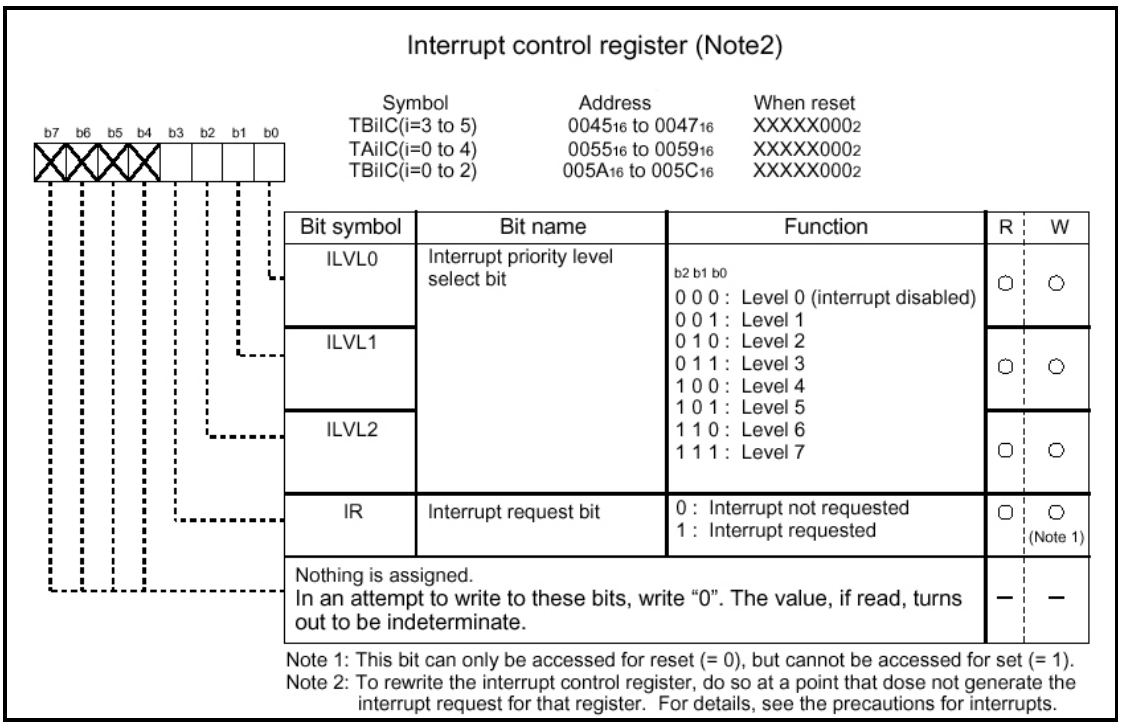
Count start flag

b7	b6	b5	b4	b3	b2	b1	b0

Symbol: TABSR
Address: 038016
When reset: 0016

Bit symbol	Bit name	Function	R	W
TA0S	Timer A0 count start flag	0 : Stops counting 1 : Starts counting	○	○
TA1S	Timer A1 count start flag		○	○
TA2S	Timer A2 count start flag		○	○
TA3S	Timer A3 count start flag		○	○
TA4S	Timer A4 count start flag		○	○
TB0S	Timer B0 count start flag		○	○
TB1S	Timer B1 count start flag		○	○
TB2S	Timer B2 count start flag		○	○

Figure 11 Count start flag register for Timers A and B



Interrupt control register (Note2)

Symbol	Address	When reset
TBiIC(i=3 to 5)	0045_{16} to 0047_{16}	$XXXXX000_2$
TAiIC(i=0 to 4)	0055_{16} to 0059_{16}	$XXXXX000_2$
TBiIC(i=0 to 2)	$005A_{16}$ to $005C_{16}$	$XXXXX000_2$

b7 b6 b5 b4 b3 b2 b1 b0

Bit symbol	Bit name	Function	R	W
ILVL0	Interrupt priority level select bit	b2 b1 b0 0 0 0 : Level 0 (interrupt disabled) 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	○	○
ILVL1			○	○
ILVL2			○	○
IR	Interrupt request bit	0 : Interrupt not requested 1 : Interrupt requested	○	○ (Note 1)
Nothing is assigned. In an attempt to write to these bits, write "0". The value, if read, turns out to be indeterminate.			–	–

Note 1: This bit can only be accessed for reset (= 0), but cannot be accessed for set (= 1).
Note 2: To rewrite the interrupt control register, do so at a point that dose not generate the interrupt request for that register. For details, see the precautions for interrupts.

Figure 12 Interrupt control register for Timers A and B

5.0 Reference

Renesas Technology Corporation Semiconductor Home Page

http://www.renesas.com

E-mail Support

support_apl@renesas.com

Data Sheets

- M16C/26 datasheet, M30262eds.pdf

User's Manual

- KNC30 Users Manual, KNC30UE.PDF
- M16C/60 and M16C/20 C Language Programming Manual, 6020EC.PDF
- Writing interrupt handlers in C for the M16C Application Note
- MSV30262-SKP or MSV-Mini26-SKP Quick start guide
- MSV30262-SKP or MSV-Mini26-SKP Users Manual
- MDECE30262 or MSV-Mini26-SKP Schematic

6.0 Software Code

A sample program written in C and compiled using the KNC30 compiler to illustrate how to configuring Event Counter Mode. The program counts 100 falling edges on the P7.5 (TA2IN) pin then flashes D5 on the MSV30262 Starter Kit Board.

To get familiar with this mode, try changing to up-count, the count value or even switch to a different timer (e.g. TA1, TB0, etc).

```
/***********************************************************************
*
*       File Name: event_mode.c
*
*       Content: Example program using Timer A2 in "Event Counter Mode".  This
*                program is written for the Event Counter Mode application note.
*                Counts falling edges on the TA2in pin. This program works with the
*                MSV30262 starter kit board.
*
*       All timing based on 20 Mhz Xtal
*
*      Copyright 2003 Renesas Technology America, Inc.
*      All Rights Reserved.
*=======================================================================
*       $Log:$
*=======================================================================*/
#include "sfr26.h"

#define  TIME_CONFIG 0x01  /*  00000001 value to load into timer mode register
                               ||||||||_  TMOD0,TMOD1: EVENT COUNTER MODE
                               ||||||____ MR0:         NO PULSE OUTPUT
                               |||||_____ MR1:         COUNT FALLING EDGES
                               ||||______ MR2:         USE UP/DOWN FLAG
                               |||_______ MR3:         = 0 IN EVENT COUNTER MODE
                               ||________ TCK0:        RELOAD TYPE
                               |_________ TCK1:        BIT NOT USED                  */
#define  CNTR_IPL 0x03                // TA2  priority interrupt level
#define  LED  p7_2                    // LED port on MSV30262 board
#define  LED_PORT_DIRECTION  pd7_2    // LED port direction on MSV30262 board
#define  OUTPUT 1

 int count;

//prototypes
 void init(void);

#pragma INTERRUPT /B TimerA2Int
 void TimerA2Int(void);
```

```
/***********************************************************************
Name:     TimerA2Int()
Parameters: none
Returns:  nothing
Description:Timer A2 Interrupt Service Routine. Interrupts every 100 falling
            edges on the TA2in pin. Flashes the LED and increments 'count'.
*********************************************************************** */
void TimerA2Int(void)
{
  int delaycntr;
  delaycntr = 0;
  count++;        // e.g for an automated packaging line, counts # of cases
  LED = 1;
  while( delaycntr <0xffff)  //software delay for flashing LED
        delaycntr++;
  LED = 0;
}

/***********************************************************************
Name:    main()
Parameters: none
Returns:  nothing
Description: initializes variables and LED port. Then does nothing but
             wait for TA2 interrupts.
*********************************************************************** */
void main (void)
{
  int temp;
  count = 0;
  LED_PORT_DIRECTION = OUTPUT;
  init();
  while (1);
}

/***********************************************************************
Name:  init()
Parameters: none
Returns:  nothing
Description: Timer TA2 setup for event counter mode, interrupts every 100 events.
*********************************************************************** */
void init()
{
   ta2 = 100;    //e.g for an automated packaging line, 100 items per case

/* the following procedure for writing an Interrupt Priority Level follows that as described
in the M16C data sheets under 'Interrupts' */

   _asm ("   fclr i") ;  // turn off interrupts before modifying IPL
   ta2ic |= CNTR_IPL;       // use read-modify-write instruction to write IPL
   ta2mr = TIME_CONFIG;
   _asm ("   fset i");

   ta2s = 1;  //start counting
}
```

In order for this program to run properly, timer A0's interrupt vector needs to point to the interrupt function, TimerA2Int. The interrupt vector table is near the end of the startup file "sect30.inc". Insert the function label "_TimerA2Int" into the interrupt vector table at vector 23 as shown below.

```
;******************************************************************************
;
;        C Compiler for M16C/26
;
;   Copyright,2003 Renesas Technology America, Inc.
;       All Rights Reserved.
;
;        Written by T.Aoyama
;        Modified  for use on MSV30262 Starter Kit.
;       sect30.inc     : section definition
;       This program is applicable when using KD30 and the ROM Monitor.
;******************************************************************************
;------------------------------------------------------------
                      :
                      :

         .lword   dummy_int                ; timer A0(for user)(vector 21)
         .lword   dummy_int                ; timer A1(for user)(vector 22)
         .glb   _TimerA2Int
         .lword _TimerA2Int                ; timer A2(for user)(vector 23)
         .lword   dummy_int                ; timer A3(for user)(vector 24)
         .lword   dummy_int                ; timer A4(for user)(vector 25)
         .lword   dummy_int                ; timer B0(for user)(vector 26)
                      :
                      :
                      :
```

Keep safety first in your circuit designs!

- 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

- 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.
- 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.
- 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).
- 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.
- 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.
- The prior written approval of Renesas Technology Corporation is necessary to reprint or reproduce in whole or in part these materials.
- 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.
- Please contact Renesas Technology Corporation for further details on these materials or the products contained therein.