M16C/62

Using the M16C/62 Timer in Event Counter Mode

1.0 Abstract

Event counters are useful in automated packaging lines, tachometers, and mechanical equipment monitoring. Also, the event counters on the M16C/62 can be configured to interrupt on a single event, adding to the interrupt input pins. The following article describes how to configure the M16C/62 timers as event counters, referred to as "Event Counter Mode."

2.0 Introduction

The M16C/62 is a 16-bit MCU, based on the M16C CPU core, with features including 10-bit A/D, D/A, UARTS, timers, DMA, etc., and up to 256k bytes of user flash. The MCU has 5 'A' timers and 6 'B' timers. All 11 timers can operate in "Event Counter Mode."

Timer A 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 Period/Pulse Width Measurement Mode

Figure 1 illustrates the operation of timer A, and Figure 2, timer B. Note that there are some differences between the two timers but both operate similarly in Event Counter Mode. The remainder of this article focuses on setting up timer A2 in Event Counter Mode.

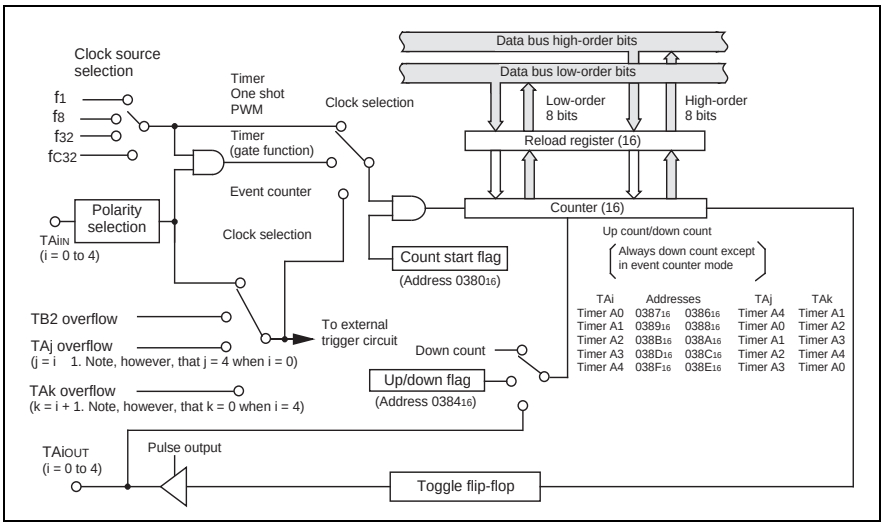


Figure 1 Bock Diagram of Timer A

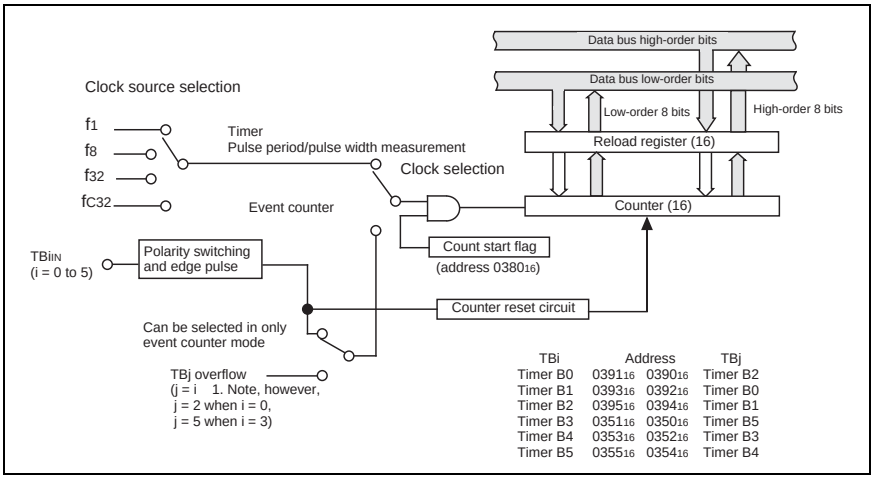


Figure 2 Bock Diagram of Timer B

3.0 Event Counter Mode Description

In general, the Timer TAi or TBi register counts an input signal and, at any time, the count value can be read. When the timer overflows (for up count) or underflows (down count), the timer interrupt request bit is set. An interrupt will be accepted when all of the following conditions are met:

- interrupt enable flag (I flag) = "1"
- interrupt request bit = "1"
- interrupt priority level > IPL (Processor Interrupt Priority Level)

If at any time during counting the count start flag is cleared, counting is suspended until set. This is illustrated in Figure 3.

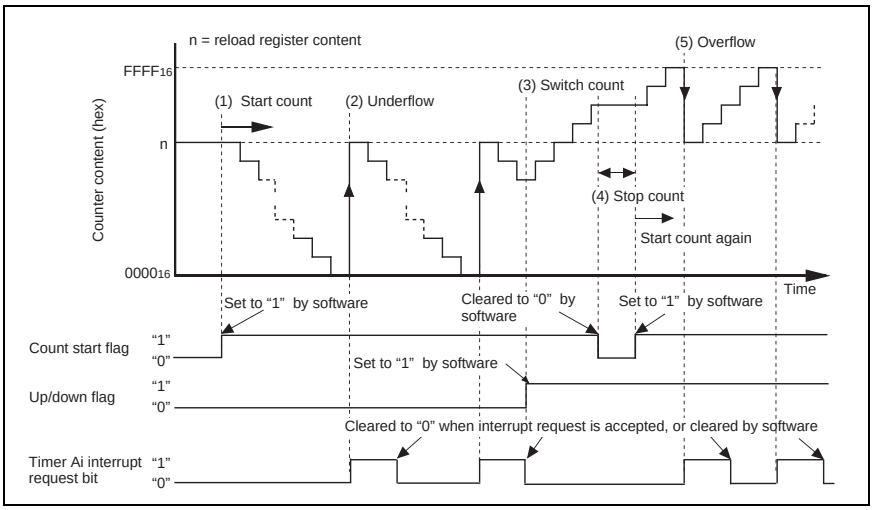


Figure 3 Operation Timing of Event Counter Mode, Reload Type Selected

Besides having the option of counting up or down, 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. and 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 Specifications in Event Counter Mode

(Single Phase Mode Only)

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 contents 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

Table 2 Timer Specifications in Event Counter Mode

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

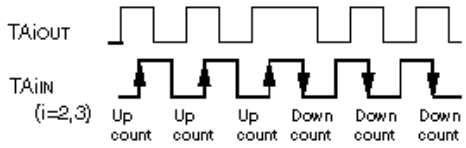
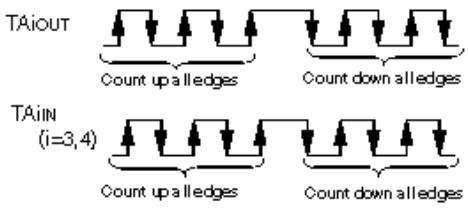
Item	Specification
Count source	• Two-phase pulse signals input to TAiin or TAiout pin
Count operation	• Up count or down count can be selected by two-phase pulse signal • When the timer overflows or underflows, the reload register content is reloaded and the timer starts over again (Note)
Divide ratio	1/ ($FFFF_{16}$ – 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	Two-phase pulse input
TAiOUT pin function	Two-phase pulse input
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	• Normal processing operation 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 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. 

Table 3 Timer B Specifications 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

To configure a timer for Event Counter Mode:

1. Load the Timer Ai mode register, TAiMR.
   - Select Event Counter Mode: bits TMOD0 = 1, TMOD1 = 0.
   - Set the remaining bits (MR0, MR1, MR2, TCK0, TCK1) depending on required functions (see mode register diagrams below).
2. Load the Timer Ai register, TAi (or TBi register) with the count source.
3. Select the trigger via the Trigger Select register, TRGSR or One-Shot Start Flag register, ONSF register (N/A for Timer B).
4. Select up or down count via the Up/down Flag register, UDF (N/A for Timer B, Timer B counts down only).
5. Load the Timer Interrupt Control register, TAiIC (or TBiIC) with an interrupt priority level, ILVL, to at least 1 if interrupts are desired.
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).

It is not necessary to perform these steps in the order listed, but 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 possibility of an interrupt occurring.

The required registers are shown in Figure 4 through Figure 13.

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 4 Timer Ai Mode Register (When Not Using Two-Phase Pulse Signal Processing)

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 5 Timer Ai Mode Register (When Using Two-Phase Pulse Signal Processing)

Timer Bi register (Note)

Symbol	Address	When reset
TB0	0391_{16}, 0390_{16}	Indeterminate
TB1	0393_{16}, 0392_{16}	Indeterminate
TB2	0395_{16}, 0394_{16}	Indeterminate
TB3	0351_{16}, 0350_{16}	Indeterminate
TB4	0353_{16}, 0352_{16}	Indeterminate
TB5	0355_{16}, 0354_{16}	Indeterminate

(b15 b7 b8) b0 b7 b0

Function	Values that can be set	R	W
• Timer mode Counts the timer©s period	0000_{16} to $FFFF_{16}$	O	O
• Event counter mode Counts external pulses input or a timer overflow	0000_{16} to $FFFF_{16}$	O	O
• Pulse period/pulse width measurement mode Measures a pulse period or width	———	O	X

Note: Read and write data in 16-bit units

Figure 6 Timer Bi Register

One-shot start flag

b7 b6 b5 b4 b3 b2 b1 b0

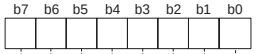
Symbol	Address	When reset
ONSF	0382_{16}	$00X00000_2$

Bit Symbol	Bit Name	Function	R	W
TM0OS	Timer A0 one-shot start flag	1 : Timer start When read, the value is indeterminate	O	O
TM1OS	Timer A1 one-shot start flag		O	O
TA2OS	Timer A2 one-shot start flag		O	O
TA3OS	Timer A3 one-shot start flag		O	O
TA4OS	Timer A4 one-shot start flag		O	O
Nothing is assigned. Write "0" when writing to this bit. If read, the value is indeterminate.			—	—
TA0TGL	Timer A0 event/trigger select bit	b1 b0 0 0 : Input on $TA0_{IN}$ is selected (Note) 0 1 : TB2 overflow is selected 1 0 : TA4 overflow is selected 1 1 : TA1 overflow is selected	O	O
TA0TGH			O	O

Note: Set the corresponding port direction register to "0"

Figure 7 One-Shot Start Flag Register

Trigger select register



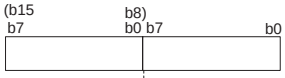
Symbol	Address	When reset
TRGSR	0383_{16}	00_{16}

Bit Symbol	Bit Name	Function	R	W
TA1TGL	Timer A1 event/trigger select bit	b1 b0 0 0 : Input on $TA1_{IN}$ is selected (Note) 0 1 : TB2 overflow is selected 1 0 : TA0 overflow is selected 1 1 : TA2 overflow is selected	O	O
TA1TGH			O	O
TA2TGL	Timer A2 event/trigger select bit	b3 b2 0 0 : Input on $TA2_{IN}$ is selected (Note) 0 1 : TB2 overflow is selected 1 0 : TA1 overflow is selected 1 1 : TA3 overflow is selected	O	O
TA2TGH			O	O
TA3TGL	Timer A3 event/trigger select bit	b5 b4 0 0 : Input on $TA3_{IN}$ is selected (Note) 0 1 : TB2 overflow is selected 1 0 : TA2 overflow is selected 1 1 : TA4 overflow is selected	O	O
TA3TGH			O	O
TA4TGL	Timer A4 event/trigger select bit	b7 b6 0 0 : Input on $TA4_{IN}$ is selected (Note) 0 1 : TB2 overflow is selected 1 0 : TA3 overflow is selected 1 1 : TA0 overflow is selected	O	O
TA4TGH			O	O

Note: Set the corresponding port direction register to "0"

Figure 8 Trigger Select Register

Timer Ai register (Note)



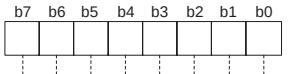
Symbol	Address	When reset
TA0	0387_{16}, 0386_{16}	Indeterminate
TA1	0389_{16}, 0388_{16}	Indeterminate
TA2	$038B_{16}$, $038A_{16}$	Indeterminate
TA3	$038D_{16}$, $038C_{16}$	Indeterminate
TA4	$038F_{16}$, $038E_{16}$	Indeterminate

Function	Values that can be set	R	W
• Timer mode Counts an internal count source	0000_{16} to $FFFF_{16}$	O	O
• Event counter mode Counts pulses from an external source or timer overflow	0000_{16} to $FFFF_{16}$	O	O
• One-shot timer mode Counts a one-shot width	0000_{16} to $FFFF_{16}$	X	O
• Pulse width modulation mode (16-bit PWM) Functions as a 16-bit pulse width modulator	0000_{16} to $FFFF_{16}$	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	00_{16} to FE_{16} (Both high-order and low-order addresses)	X	O

Note: Read and write data in 16-bit units

Figure 9 Timer Ai Register

Count start flag

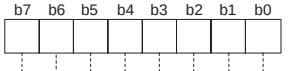


Symbol: TABSR | Address: 0380_{16} | When reset: 00_{16}

Bit Symbol	Bit Name	Function	R	W
TA0S	Timer A0 count start flag	0 : Counting stops 1 : Counting starts	O	O
TA1S	Timer A1 count start flag		O	O
TA2S	Timer A2 count start flag		O	O
TA3S	Timer A3 count start flag		O	O
TA4S	Timer A4 count start flag		O	O
TB0S	Timer B0 count start flag		O	O
TB1S	Timer B1 count start flag		O	O
TB2S	Timer B2 count start flag		O	O

Figure 10 Count Start Flag Register

Up/down flag



Symbol: UDF | Address: 0384_{16} | When reset: 00_{16}

Bit Symbol	Bit Name	Function	R	W
TA0UD	Timer A0 up/down flag	0 : Down count 1 : Up count This specification becomes valid when the up/down flag content is selected for up/down switching cause	O	O
TA1UD	Timer A1 up/down flag		O	O
TA2UD	Timer A2 up/down flag		O	O
TA3UD	Timer A3 up/down flag		O	O
TA4UD	Timer A4 up/down flag		O	O
TA2P	Timer A2 two-phase pulse signal processing select bit	0 : Disabled 1 : Enabled When not using the two-phase pulse signal processing function, set the select bit to "0"	O	O
TA3P	Timer A3 two-phase pulse signal processing select bit		O	O
TA4P	Timer A4 two-phase pulse signal processing select bit		O	O

Figure 11 Up/Down Flag Register

Interrupt control register (Note 2)

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

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	O	O
ILVL1			O	O
ILVL2			O	O
IR	Interrupt request bit	0 : Interrupt not requested 1 : Interrupt requested	O	O Note 2
Nothing is assigned. Write "0" when writing to these bits. The value is "0" if read.			–	–

Note 1: Rewrite the interrupt control register only if it will not generate an interrupt request for that register. See precautions listed at end of the Interrupts chapter.
Note 2: This bit can only be accessed for reset (=0) Set (=1) cannot be accessed.

Figure 12 Interrupt Control Register

Timer B3, 4, 5 count start flag

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
TBSR	0340_{16}	$000XXXXX_2$

Bit Symbol	Bit Name	Function	R	W
Nothing is assigned. Write "0" when writing to these bits. The value, if read, turns out to be indeterminate.			–	–
TB3S	Timer B3 count start flag	0 : Stops counting 1 : Starts counting	O	O
TB4S	Timer B4 count start flag		O	O
TB5S	Timer B5 count start flag		O	O

Figure 13 Timer B3, 4, 5 Count Start Flag Register

5.0 Reference

Renesas Technology Corporation Semiconductor Home Page

http://www.renesas.com

E-mail Support

support_apl@renesas.com

Data Sheets

- M16C/62 datasheets, 62aeds.pdf

User's Manual

- M16C/62 User's Manual, 62eum.pdf
- M16C/60 and M16C/20 C Language Programming Manual, 6020EC.pdf
- Application Note: Writing Interrupt Handlers in C for the M16C
- NC30 Ver. 4.0 User's Manual, NC30UE.pdf

6.0 Software Code

Below is a program written for the NC30 compiler to illustrate how to configure Event Counter Mode. The program counts 100 falling edges on the P7.5 (TA2IN) pin then flashes LED4 on the MSV1632/62 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. This program
*           works with the MSV1632/62 starter kit board.
*      Compiled with NC30 ver. 3.20.00.
*      All timing based on 16 Mhz Xtal
*
*     Copyright, 2003 Renesas Technology Corporation, Inc
*=======================================================================
*      $Log:$
*=======================================================================*/
#include "sfr62.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                // TA0  priority interrupt level
#define  LED  p7_2                    // LED port on MSV1632 board
#define  LED_PORT_DIRECTION  pd7_2    //LED port direction on MSV1632 board
#define  OUTPUT 1
```

```
//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:  initial()
Parameters: none
Returns:  nothing
Description: Timer TA2 setup for 5msec interrupts.
************************************************************************ */
 void init()
  {
   ta2 = 100;    //e.g for an automated packaging line, 100 items per cases

/* 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
   ta0ic |= CNTR_IPL;          // use read-modify-write instruction to write IPL
   ta0mr = TIME_CONFIG;
   _asm ("   fset i");

   ta0s = 1;  //start counting
  }
```

In order for this program to run properly, timer A2's interrupt vector needs to point to the function. 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/62
;
;   Copyright,2003 Renesas Technology Corporation, Inc
;       All Rights Reserved.
;
;      Written by T.Aoyama
;      Modified  for use on MSV1632/62 Starter Kit.
;       sect30.inc     : section definition
;       This program is applicable when using KD30 and the ROM Monitor.
;*************************************************************************
;-----------------------------------------------------------
```

```
                :
                :
                :
                :
                :

.lword          dummy_int           ; A-D(for user)(vector 14)
.lword          dummy_int           ; uart2 transmit(for user)(vector 15)
.lword          dummy_int           ; uart2 receive(for user)(vector 16)
.lword          dummy_int           ; uart0 transmit(for user)(vector 17)
.lword          dummy_int           ; uart0 receive(for user)(vector 18)
.lword          0ff900h             ; uart1 transmit(for user)(vector 19)
.lword          0ff900h             ; uart1 receive(for user)(vector 20)
.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)
.lword          dummy_int           ; timer B1(for user)(vector 27)
.lword          dummy_int           ; timer B2(for user)(vector 28)
.lword          dummy_int           ; int0 (for user)(vector 29)
.lword          dummy_int           ; int1 (for user)(vector 30)
.lword          dummy_int           ; int2 (for user)(vector 31)
                :
                :
                :
                :
                :
```

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.