To all our customers

Regarding the change of names mentioned in the document, such as Mitsubishi Electric and Mitsubishi XX, to Renesas Technology Corp.

The semiconductor operations of Hitachi and Mitsubishi Electric were transferred to Renesas Technology Corporation on April 1st 2003. These operations include microcomputer, logic, analog and discrete devices, and memory chips other than DRAMs (flash memory, SRAMs etc.) Accordingly, although Mitsubishi Electric, Mitsubishi Electric Corporation, Mitsubishi Semiconductors, and other Mitsubishi brand names are mentioned in the document, these names have in fact all been changed to Renesas Technology Corp. Thank you for your understanding. Except for our corporate trademark, logo and corporate statement, no changes whatsoever have been made to the contents of the document, and these changes do not constitute any alteration to the contents of the document itself.

Note : Mitsubishi Electric will continue the business operations of high frequency & optical devices and power devices.

Renesas Technology Corp.
Customer Support Dept.
April 1, 2003

RENESAS
Renesas Technology Corp.

APPLICATION NOTE

M16C

Pulse Period/Width Measurement on the M16C/62

1. Abstract

Measuring the frequency (1/period) or the pulse width of an input signal is useful in applications such as tachometers, DC motor control, power usage calculations, and so on. The following article describes how to use timer B to measure the period and pulse width of an input waveform, referred to as 'Pulse Period/Pulse Width Measurement Mode'.

2. Introduction

The Mitsubishi M16C/62 is a 16-bit MCU, based on the M16C CPU core, with an impressive list of features including 10-bit A/D, D/A, UARTS, timers, DMA, etc., and up to 256k bytes of user flash. The MCU has 6 B timers and all 6 timers can operate in Pulse Period/Pulse Width Measurement Mode.

Timer B has the following additional modes of operation:

- Timer Mode
- Event Counter Mode

Figure 1 illustrates the operation of timer B. The remainder of this article focuses on setting up timer B0 to measure pulse width, and timer B1 to measure pulse period.

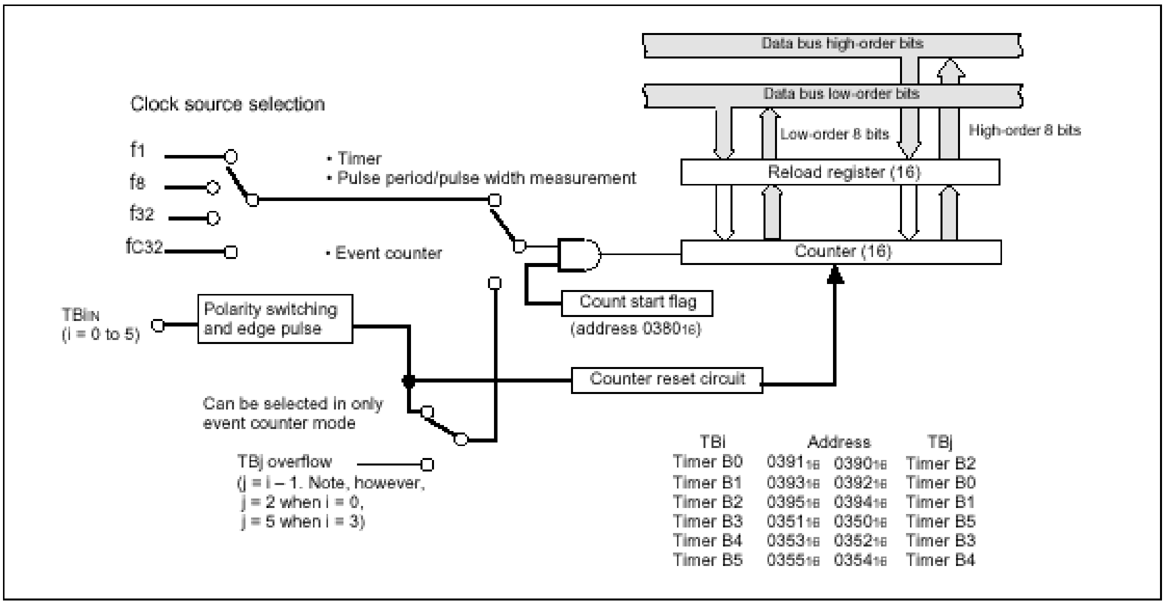

Figure 1. Block Diagram of Timer B

3. Pulse Period/Pulse Width Measurement Mode Description

As illustrated in Figure 1, the timer TBi register consists of two parts, a counter and a reload register. In Measurement Mode, when an effective edge appears on the TBiIN pin, the count value is transferred to the reload register and the CPU can read this value by performing a read on the TBi register. The measured time is the counter value (TBi) divided by the frequency of the clock source (Fi). Two period measurement options are available that measure from falling edge to falling edge or rising edge to rising edge. For width measurement, the measurement is taken at both edges and software determines if the measured value is for the high width or low.

3.1 Pulse Period Measurement

In Period Measurement Mode (e.g. falling edge to falling edge), after the ‘start count flag’ is set, the counter counts up using the selected clock source and every time a falling edge is detected on the TBiIN pin, the value in the counter is transferred to the reload register, the counter is reset to zero, and then continues counting. At the same time, 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 register is cleared, the interrupt will not be serviced until the flag is set). If the timer’s counter overflows within a period, it will also generate the interrupt and the MR3 bit in the TBiMR is set to distinguish between the interrupt causes. Note that the measurement is free running and the reload register contains the most recent measurement. The user has the option of polling the TBi register or reading it in an interrupt service routine. Also note that the value of the counter immediately after the ‘start count flag’ is set is indeterminate and an overflow could occur before the first falling edge. Figure 2 illustrates this.

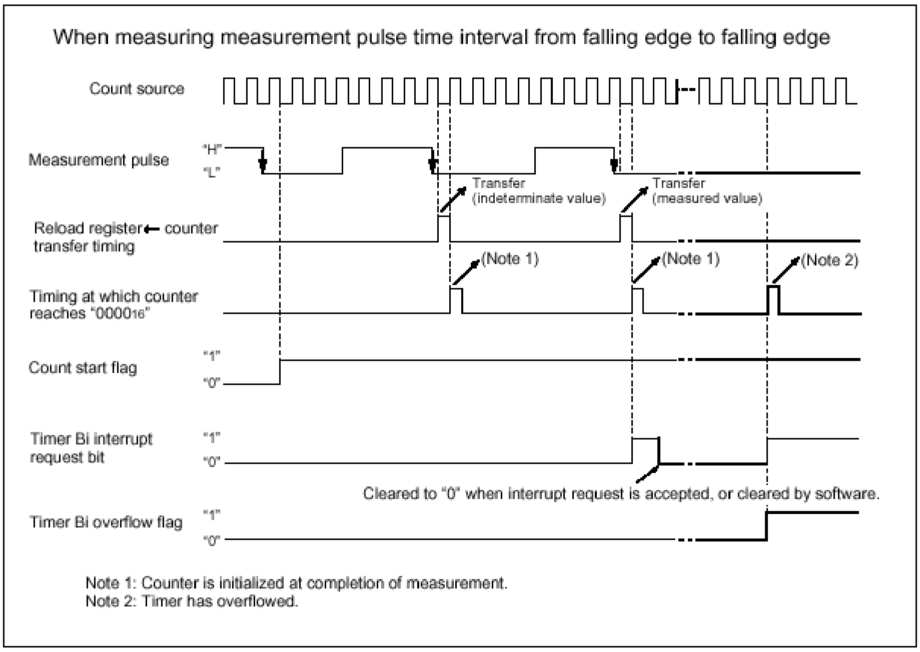

Figure 2. Operation Timing When Measuring a Pulse Period

3.2 Pulse Width Measurement

Pulse Width Measurement Mode operates in much the same way except the count register is transferred to the reload register for every edge detected on the TBiIN pin, and the counter resets and resumes counting, as shown in Figure 3. Again, note that the value of the counter immediately after the 'start count flag' is set is indeterminate and an overflow could occur before the first falling edge. This measurement is also free running but now the user must determine by software whether the measurement is for the high or low width.

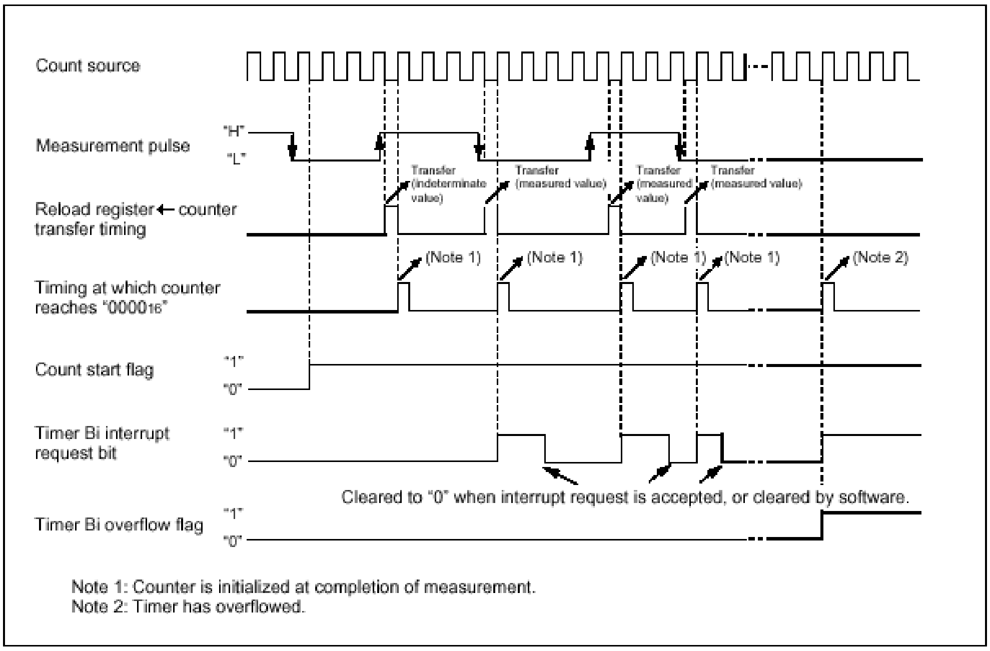

Figure 3. Operation Timing When Measuring a Pulse Width

4. Configuring Pulse Period/Pulse Width Measurement Mode

To configure a timer for Pulse Period/Pulse Width Measurement Mode:

1. Load the timer mode register, TAiMR.

- Select Measurement Mode: bits TMOD0 = 0, TMOD1 = 1.
- Set the MR0 and MR1 bits for period or width measurement.
- Clear the MR2 bit for period or width measurement.
- MR3 is the timer Bi overflow flag (can be cleared but not set).
- Select the clock source (f1, f/8, f/32, or fc/32): bits TCK0,TCK1 register.

2. Set the timer 'interrupt priority level', TBiIC, to at least 1 if required.

3. Enable interrupts (CPU I flag set).

4. Set the 'start count flag' bit, 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.

Timer Bi mode register

b7 b6 b5 b4 b3 b2 b1 b0: bits b1, b0 = 1 0

Symbol	Address	When reset
TBiMR(i=0 to 5)	$039B_{16}$ to $039D_{16}$	$00XX0000_2$
	$035B_{16}$ to $035D_{16}$	$00XX0000_2$

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0: 1 0 : Pulse period / pulse width measurement mode	O	O
TMOD1			O	O
MR0	Measurement mode select bit	b4 b3: 0 0 : Pulse period measurement (Interval between measurement pulse's falling edge to falling edge); 0 1 : Pulse period measurement (Interval between measurement pulse's rising edge to rising edge); 1 0 : Pulse width measurement (Interval between measurement pulse's falling edge to rising edge, and between rising edge to falling edge); 1 1 : Inhibited	O	O
MR1			O	O
MR2	0 (Fixed to "0" in pulse period/pulse width measurement mode; i = 0, 3)		O (Note 2)	O
	Nothing is assigned (i = 1, 2, 4, 5). In an attempt to write to this bit, write "0". The value, if read, turns out to be indeterminate.		X (Note 3)	X
MR3	Timer Bi overflow flag (Note 1)	0 : Timer did not overflow; 1 : Timer has overflowed	O	X
TCK0	Count source select bit	b7 b6: 0 0 : f_1; 0 1 : f_8; 1 0 : f_{32}; 1 1 : f_{C32}	O	O
TCK1			O	O

Note 1: The timer Bi overflow flag changes to "0" when the count start flag is "1" and a value is written to the timer Bi mode register. This flag cannot be set to "1" by software.
Note 2: Timer B0, timer B3.
Note 3: Timer B1, timer B2, timer B4, timer B5.

Figure 4. Timer Bi Mode Register in Pulse Period/Pulse Width Measurement Mode

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
TB3	0351₁₆, 0350₁₆	Indeterminate
TB4	0353₁₆, 0352₁₆	Indeterminate
TB5	0355₁₆, 0354₁₆	Indeterminate

Function	Values that can be set	R	W
• Timer mode Counts the timer's period	0000₁₆ to FFFF₁₆	○	○
• Event counter mode Counts external pulses input or a timer overflow	0000₁₆ to FFFF₁₆	○	○
• Pulse period / pulse width measurement mode Measures a pulse period or width	—	○	×

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

Figure 5. Timer Bi Register

Count start flag

Symbol	Address	When reset
TABSR	0380₁₆	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 6. Count Start Flag

Timer B3, 4, 5 count start flag

Symbol	Address	When reset
TBSR	0340₁₆	000XXXXX₂

Bit symbol	Bit name	Function	R	W
Nothing is assigned. In an attempt to write to these bits, write "0". The value, if read, turns out to be indeterminate.			—	—
TB3S	Timer B3 count start flag	0 : Stops counting 1 : Starts counting	○	○
TB4S	Timer B4 count start flag		○	○
TB5S	Timer B5 count start flag		○	○

Figure 7. Timer B3, 4, 5 Count Start Flag

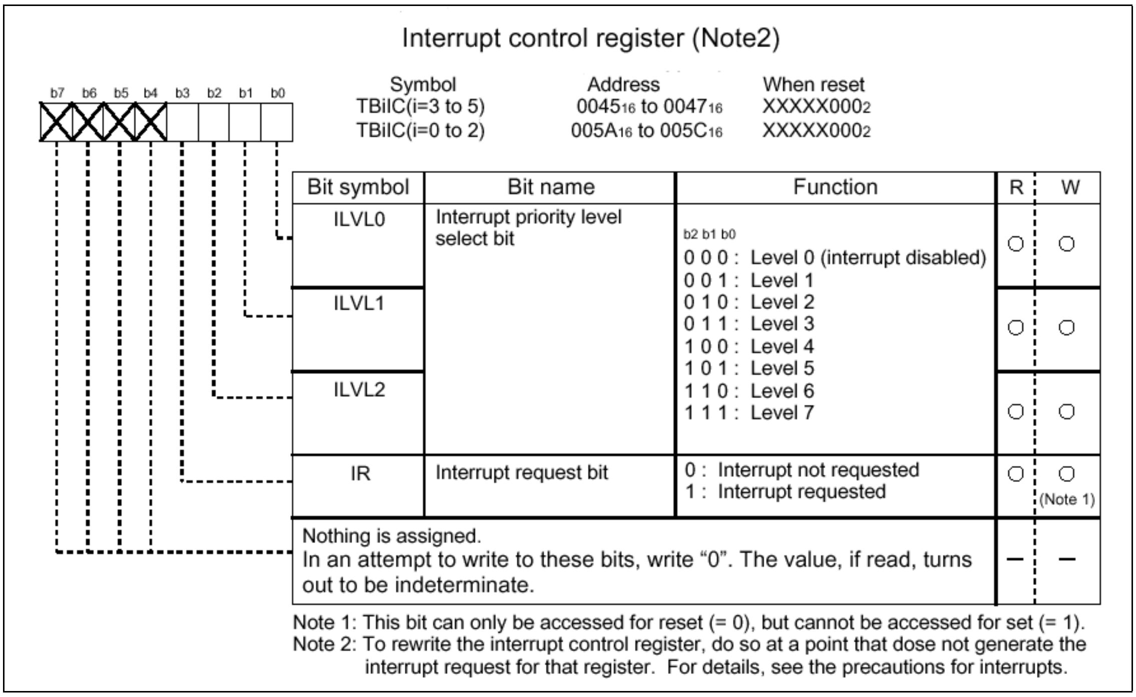

Figure 8. Interrupt Control Register

5. References

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

6. Software Code

Below is a program written for Mitsubishi’s NC30 compiler to illustrate how to configure Pulse Period/Pulse Width Measurement Mode. The program can measure up to about a 16msec period and runs on the MSV1632 Starter Kit Board. Using the KD30 debugger, the program can be ‘stopped’ and the global variables ‘widthlow’, ‘width_hi’, and ‘period’ viewed from the global watch window.

To become familiar with this mode, try changing the clock source or even switch to a different timer (e.g. TB2, TB3, etc.).

```
/****************************************************************************
*
*     File Name: period_width.c
*
*     Content: Example program using Timer B in 'pulse width / period measurement
*              mode'. This program is written for the "Pulse Period/Width
*              Measurement" application note. Timer B0 is configured to measure
*              pulse width(TB0in pin),and timer B1,measures the period(TB1in pin).
*              Tested using a 250Hz square wave with a 0.5msec pulse width(high).
*              This program works with the MSV1632 starter kit board.
*
*      Compiled with NC30 ver. 3.20.00.
*
*      All timing based on 16 Mhz Xtal
*
*      Copyright,2001 MITSUBISHI ELECTRIC CORPORATION
*      AND MITSUBISHI SEMICONDUCTOR SYSTEM CORPORATION
*      and Mitsubishi Electric and Electronics USA
*============================================================================
*      $Log:$
*===========================================================================*/
```

```
#include "sfr62.h"

#define B1TIME_CONFIG 0x42  /*  01000010 value to load into timer B1 mode register
                                ||||||||_  TMOD0,TMOD1: PULSE MEASUREMENT MODE
                                ||||||____ MR0,MR1:     PULSE PERIOD MODE
                                ||||_______MR2:         = 0 FOR PULSE MEASUREMENT
                                |||_______ MR3:         OVERFLOW FLAG
                                ||________ TCK0,TCK1:   F DIVIDED BY 8 SELECTED */

#define B0TIME_CONFIG 0x4a  /*  01001010 value to load into timer B0 mode register
                                ||||||||_  TMOD0,TMOD1: PULSE MEASUREMENT MODE
                                ||||||____ MR0,MR1:     PULSE WIDTH MODE
                                ||||_______MR2:         = 0 FOR PULSE MEASUREMENT
                                |||_______ MR3:         OVERFLOW FLAG
                                ||________ TCK0,TCK1:   F DIVIDED BY 8 SELECTED */

#define  CNTR_IPL 0x03   // TB0  priority interrupt level

int period,widthlow,width_hi;

//prototypes
 void init(void);

#pragma INTERRUPT /B TimerB0Int
 void TimerB0Int(void);

/*****************************************************************************
Name:      TimerB0Int()
Parameters: none
Returns:  nothing
Description: Timer B0 Interrupt Service Routine. The overflow flag is check
             to determine if the TB0 register contains valid data. If so, the
             input is tested to determine if the value in the TB0 register
             is the high pulse width or low width and stored in the appropriate
             variable.
***************************************************************************** */

void TimerB0Int(void)
{
  if (mr3_tb0mr ==1)       // check for timer overflow
  { tb0mr = B0TIME_CONFIG;     // if so clear flag and
    return;                       // data invaild, so leave
  }
 if (p9_0== 1)
  widthlow = tb0 ;             // if input now hi, just measured a low width
 else
  width_hi = tb0;

}
```

```
/*****************************************************************************
Name:    main()
Parameters: none
Returns:  nothing
Description: initializes variables. Then the variable 'period' is constantly
             updated with the period count in timer TB1. This is to illustrate
             that the period measurement is free running. Note that the first
             few times TB1 is read, the data may not be valid.
*****************************************************************************/

void main (void)
{

  init();
  while (1)
  {
      period = tb1 ;             // period measured in polled mode
  }
}
/*****************************************************************************
Name:  initial()
Parameters: none
Returns:  nothing
Description: Timer TB0 setup for pulse width interrupts and TB1 configured for
             pulse period measurement (no interrupts).
**************************************************************************** */
 void init()
  {

/* 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
   tb0ic |= CNTR_IPL;        // use read-modify-write instruction to write IPL
   tb0mr = B0TIME_CONFIG;
   _asm ("   fset i");

   tb0s = 1;  //start counting

   tb1mr = B1TIME_CONFIG;
   tb1s = 1;  //start counting

  }
```

In order for this program to run properly, timer B0'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 "TimerB0Int" into the interrupt vector table at vector 26 as shown below.

```
;*****************************************************************************
;
;       C Compiler for M16C/62
;
;       Copyright,2000 MITSUBISHI ELECTRIC CORPORATION
;       AND MITSUBISHI SEMICONDUCTOR SYSTEM CORPORATION
;       and Mitsubishi Electric and Electronics USA
;       All Rights Reserved.
;
;       Written by T.Aoyama
;       Modified  for use on MSV1632 Starter Kit.
;       sect30.inc      : section definition
;       This program is applicable when using KD30 and the ROM Monitor.
;
;       $Id:
;
;*****************************************************************************

                                        :
                                        :
                                        :
                                        :
                                        :
                                        :

        .lword          dummy_int           ; timer A0(for user)(vector 21)
        .lword          dummy_int           ; timer A1(for user)(vector 22)
        .lword          dummy_int           ; timer A2(for user)(vector 23)
        .lword          dummy_int           ; timer A3(for user)(vector 24)
        .lword          dummy_int           ; timer A4(for user)(vector 25)
        .glb            _TimerB0Int
        .lword          _TimerB0Int         ; 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)

        .lword          dummy_int           ; vector 32 (for user or MR30)
        .lword          dummy_int           ; vector 33 (for user or MR30)
        .lword          dummy_int           ; vector 34 (for user or MR30)
        .lword          dummy_int           ; vector 35 (for user or MR30)
        .lword          dummy_int           ; vector 36 (for user or MR30)
        .lword          dummy_int           ; vector 37 (for user or MR30)
                                        :
                                        :
                                        :
```

Keep safety first in your circuit designs!

- Mitsubishi Electric Corporation puts the maximum effort into making semiconductor products better and more reliable, but there is always a possibility of trouble occurring. 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 backup, auxiliary circuits, (ii) use of non-flammable material, or (iii) prevention against any malfunction or mishap.

Notes regarding this information

- This information is intended as a reference to assist our consumers in the selection of the Mitsubishi semiconductor product best suited to the customer's application; it does not convey any license under any intellectual property rights, or any other rights, belonging to Mitsubishi Electric Corporation or a third party.
- Mitsubishi Electric 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 is subject to change by Mitsubishi Electric Corporation without notice due to product improvements or other reasons. It is therefore recommended that customers contact Mitsubishi Electric Corporation or an authorized Mitsubishi Semiconductor product distributor for the latest product information before purchasing a product listed herein.
- The information described herein may contain technical inaccuracies or typographical errors. Mitsubishi Electric Corporation assumes no responsibility for any damage, liability, or other loss arising from these inaccuracies or errors.
- Please also pay attention to information published by Mitsubishi Electric Corporation by various means, including the Mitsubishi Semiconductor home page (http://www.mitsubishichips.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. Mitsubishi Electric Corporation assumes no responsibility for any damage, liability, or other loss resulting from the information contained herein.
- Mitsubishi Electric 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 Mitsubishi Electric Corporation or an authorized Mitsubishi Semiconductor 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 Mitsubishi Electric Corporation is necessary to reprint or reproduce these materials in whole or in part.
- If these products or technologies are subject to 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 Mitsubishi Electric Corporation or an authorized Mitsubishi Semiconductor product distributor for further details on these materials or the products contained therein.

RENESAS
Renesas Technology Corp.

Mitsubishi Electric & Electronics USA, Inc.
Electronic Device Group

USA Headquarters

Mitsubishi Electric & Electronics USA, Inc.
Electronic Device Group
1050 East Arques Avenue
Sunnyvale, CA 94085-4601
Phone: 408-730-5900
FAX: 408-732-9382

Direct Sales Offices – USA

Northwest

Mitsubishi Electric & Electronics USA, Inc.
Electronic Device Group
1050 East Arques Avenue
Sunnyvale, CA 94085-4601
Phone: 408-730-5900
FAX: 408-737-1129

Southwest

Mitsubishi Electric & Electronics USA, Inc.
Electronic Device Group
20 Fairbanks, Suite 181
Irvine, CA 92618
Phone: 949-859-9453
FAX: 949-859-9450

Rocky Mountain

Mitsubishi Electric & Electronics USA, Inc.
Electronic Device Group
PO Box 273377
Fort Collins, CO 80527
Phone: 970-669-4068
FAX: 970-669-4129

North Central

Mitsubishi Electric & Electronics USA, Inc.
Electronic Device Group
3300 Edinborough Way, Suite 511
Edina, MN 55435
Phone: 952-837-9053
FAX: 952-837-9059

Mitsubishi Electric & Electronics USA, Inc.
(Automotive Market Only)
50 West Big Beaver Rd., Suite 136
Troy, MI 48084
Phone: 248-526-9580
FAX: 248-526-9583

South Central

Mitsubishi Electric & Electronics USA, Inc.
Electronic Device Group
8310 Capital of Texas Hwy. N., Suite 260
Austin, TX 78731
Phone: 512-346-4200
FAX: 512-346-4434

Northeast

Mitsubishi Electric & Electronics USA, Inc.
Electronic Device Group
92 Montvale Avenue, Suite 2500
Stoneham, MA 02180
Phone: 781-245-6956
FAX: 781-245-4233

Mid-Atlantic

Mitsubishi Electric & Electronics USA, Inc.
Electronic Device Group
92 Montvale Avenue, Suite 2500
Stoneham, MA 02180
Phone: 781-245-6956
FAX: 781-245-4233

South Atlantic

Mitsubishi Electric & Electronics USA, Inc.
Electronic Device Group
2625 Meridian Parkway, Suite 100
Durham, NC 27713
Phone: 919-767-7900
FAX: 919-767-7902

Southeast

Mitsubishi Electric & Electronics USA, Inc.
Electronic Device Group
2810 Premiere Parkway, Suite 400
Duluth, GA 30097
Phone: 678-258-4518
FAX: 678-258-4519

Mexico Inquiries

Western Mexico

Mitsubishi Electric & Electronics USA, Inc.
Electronic Device Group
20 Fairbanks, Suite 181
Irvine, CA 92618
Phone: 949-859-9453
FAX: 949-859-9450

Eastern Mexico

Mitsubishi Electric & Electronics USA, Inc.
Electronic Device Group
8310 Capital of Texas Hwy. N., Suite 260
Austin, TX 78731
Phone: 512-346-4200
FAX: 512-346-4434

Puerto Rico Inquiries

Puerto Rico

Mitsubishi Electric & Electronics USA, Inc.
Electronic Device Group
2810 Premiere Parkway, Suite 400
Duluth, GA 30097
Phone: 678-258-4518
FAX: 678-258-4519

Mitsubishi Electric Sales Canada, Inc.

Direct Sales Offices – Canada

Central & Western Canada

Mitsubishi Electric Sales Canada, Inc.
4299 14th Avenue
Markham, Ontario,
Canada L3R 0J2
Phone: 905-475-7728
FAX: 905-475-1918

Eastern Canada

Mitsubishi Electric Sales Canada, Inc.
340 March Road, Suite 300
Kanata, Ontario,
Canada K2K 2E4
Phone: 613-591-3348
FAX: 613-591-3948

QS-9000

JQA-QS0161

Mitsubishi Electric MCU products sold into the Automotive marketplace are QS-9000 certified.

Printed in U.S.A.

www.mitsubishichips.com