To our customers,

Old Company Name in Catalogs and Other Documents

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

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

April 1st, 2010
Renesas Electronics Corporation

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

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

RENESAS

M16C/60, M16C/Tiny

High Speed High Resolution PWM Using IAR C-Compiler

Introduction

The M32C and M16C family provides PWM (Pulse Width Modulation) functionality through timer A modes. Up to five timer A's are available for PWM mode with 8bit or 16bit resolution selectable. These timers use internal clock divider from Xin (or internal PLL output) signal to create the modulation signal. The maximum PWM frequency is achieved when using divide by 1 (also called f1 inside the device).The maximum PWM frequency is equal to f1 divided by 2 to the power of the PWM resolution in number of bits.

As example when 13bit resolution the maximum frequency would be 2.929KHZ with 24MHZ timer clock ($24 \times 10^6 / (2^{13}) = 2{,}929$)

For some applications it is required to have higher frequencies in order to avoid audible switching frequencies and/or to filter more easily the PWM signal. The purpose of this application note is to show how a high frequency PWM with a high resolution can be implemented. The example outputs a 20KHZ or higher frequency PWM with a 13 bits resolution.

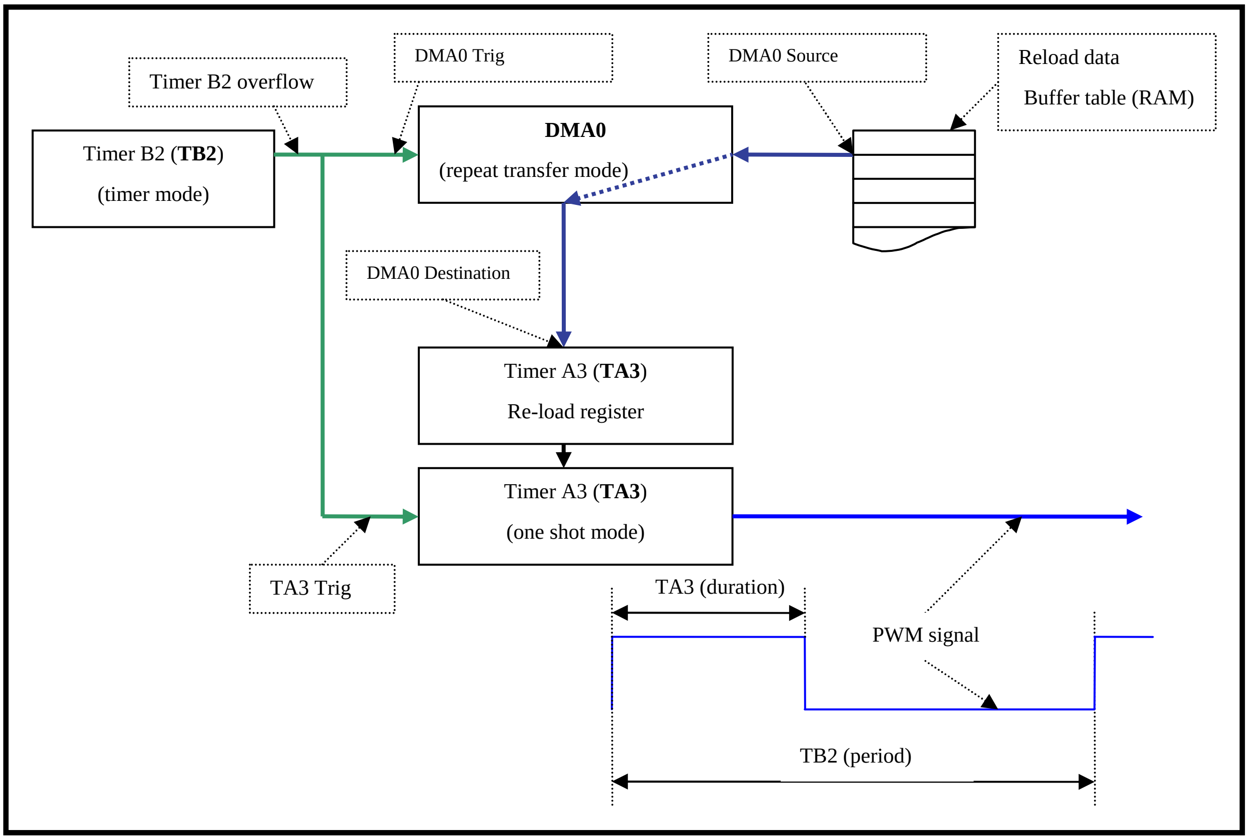
Contents

Peripheral usage and settings.

This application note uses the internal DMA channel in repeat mode to modulate the content of the PWM generation timer. The basic operations for variable PWM resolution are described in the application note *"How to generate variable resolution PWM with M16C family processors"*, it is recommended to read this application note before starting with current one.

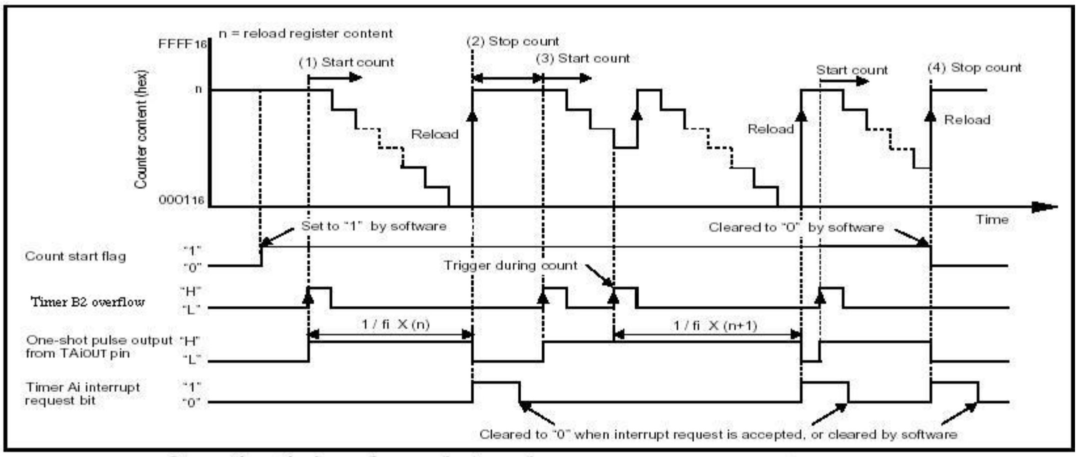
Timer B2 (TB2) is used as trigger source for DMA0 channel as well as for the Timer A3 (TA3). TA3 is set in one shot mode trigged by TB2. The PWM frequency is defined by TB2 settings and contents. TA3 contents define the output pulse high duration. DMA0 transfers data from a RAM table, which is updated by the MCU application software, to the TA3 register.

Such operation can also be used to generate specific signals (sinus, triangular, noise…) in an automatic manner.



1. Operation timing of Timer A one-shot mode

In this mode Timer A outputs a high level pulse with a duration time defined by Timer A reload register contents.



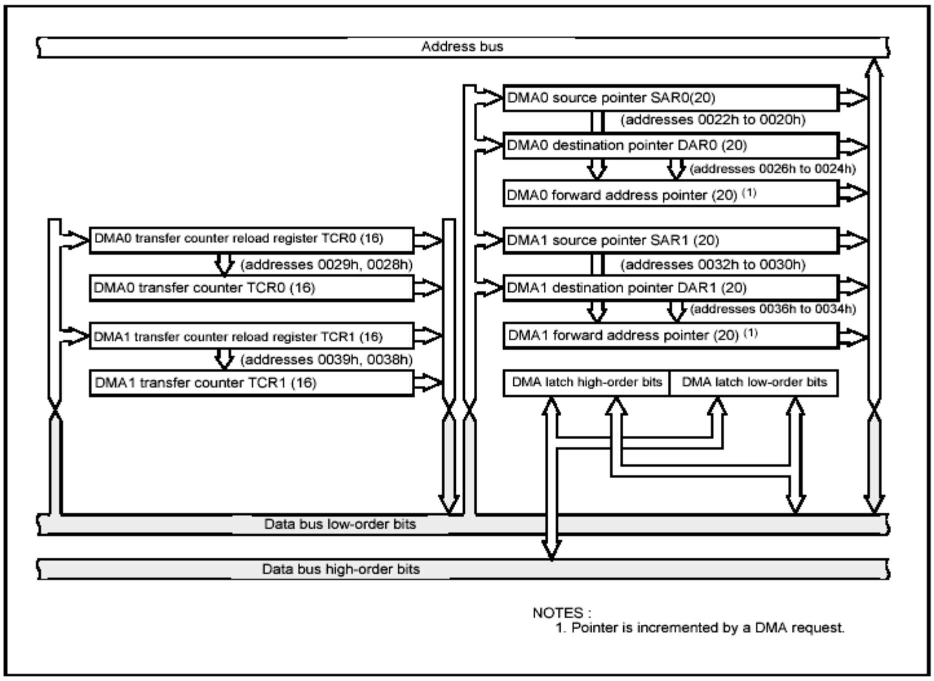
Operation timing of one-shot mode

2. M16C DMA operation.

All M16C devices have at least two DMA channels (DMA0 and DMA1). DMA's role is to transfer a certain amount of data (defined by software) from a source to a destination trigged by some event. To achieve this each DMA channel has its own source register (20bit address), destination register (also 20bits) and count register (16bits) along with configuration registers.

The DMA source and/or destination can be any location inside the M16C, it could be either RAM or Flash or SFR register. DMA block diagram is bellow.

The DMA channel only accepts source or destination increment (not both at same time).



3. DMA configuration.

Either DMA0 or DMA1 can be used to achieve this application; DMA0 has been selected for this application.

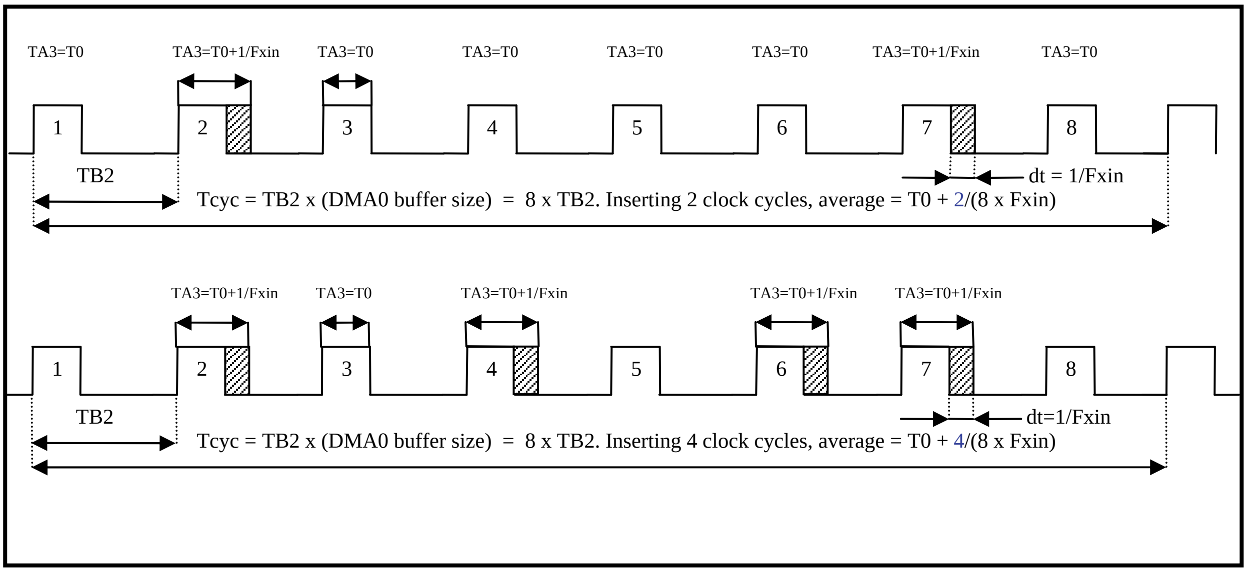
DMA0 is used in repeat mode with fixed destination and forward source which means that the source pointer is incremented after each DMA transfer. When the transfer counter (TCR0) reaches zero the source pointer and the transfer counter are automatically reloaded with their SFR reload register values and process continues.

The DMA0 channel source points to the buffer table that contains the data to be set for each PWM cycle, the destination is the TA3 reload register and the number of transfers is defined as source buffer table size (8 in this example).

Increasing PWM resolution keeping frequency.

As explained in the introduction chapter, the maximum PWM frequency is limited by the requested resolution and operating frequency of the timers. In this example the requested 20KHZ PWM frequency using 20MHZ clock frequency leads to have TB2=1000 (which can also achieve 24KHZ PWM for same TB2 value and Xin=24MHZ). In order to make things easier to calculate and understand it has been chosen to select TB2= 1024 which is the power of 2 value closest to 1000. This will lead to have a PWM frequency of 19.531KHZ with Xin=20MHZ or 23.437KHZ with Xin=24MHZ.

In normal operation for a fixed PWM value the duration at high level and low level are fixed from period to period. The principle implemented here is to change dynamically, using the DMA0, the PWM value in order to create an average result (over 8 PWM periods in this application note example) which increase the resolution compared to standard drive. Bellow figure explains the operation for 2 different PWM values.



If Th1 is the duration at high level for first period in the cycle and Th2 the same for second period and so on, the average value over 8 periods is equal to (Th1+Th2+…+Th8)/8.

Software

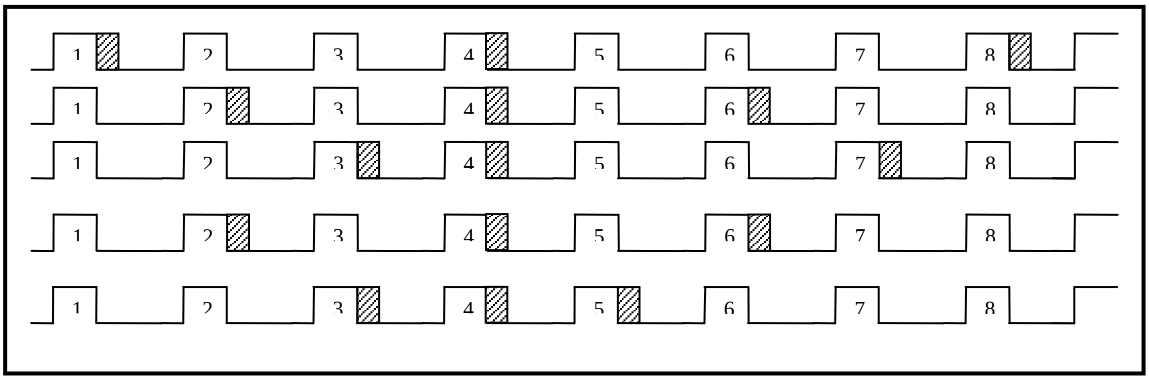
Software Algorithm

The main calculation function is *Filling_Dma*, it gets as parameter the PWM value (with 13 representative bits), splits the value in *MSB* (10bits) and *LSB* (3bits) and uses a constant look-up table to fill-in the *Pwm_Table* buffer. *MSB* is the default value for TA3, the *LSB* parameter indicates how many periods in a 8 period frame will have one additional unit in duration high of the PWM signal (*MSB*+1).

The *Pwm_Table* contents are sent automatically every TB2 overflow to TA3 by the DMA0.

The reason for the look-up table is to reduce the ripple noise due to the value averaging over 8 samples. The same average value can be reached with different *Pwm_Table* values as shown in bellow drawing example.

The drawing bellow shows some possible combinations when lsb = 3 (which means that 3 periods in a frame of 8 will have one unit more in pulse duration while the other 5 periods remain non modified).



Conclusion.

This piece of software allows expanding system performance without any extra cost.

The CPU usage is limited to the *Pwm_Table* calculation, when no change to PWM value is required the peripherals are set so that they run automatically which results on an almost zero cpu load.

As well the code size is very limited; the main routine with initialization and push button scan uses only 315 bytes of code.

Appendix

Code listing

```
#define uint unsigned int
#define uchar unsigned char
// defines definition
#define RESOLUTION_INCREASE 3 //number of bits increased in resolution
#define RECURRENCE_SIZE       8 // 2 to the power 3 (number of bits to increase the
resolution
#define SW_INT0             (P8&0x04)
#define SW_INT1             (P8&0x08)
#define SW_INT2             (P8&0x10)
#define LED4                P2_bit.P2_3
#define LED6                P2_bit.P2_5
```

```
#define LED8                P2_bit.P2_7
#define TEMPO               100
#define PWM_TIME              1024  // Value stored in TB2 and max value to be stored in
TA3
#define TMAX                (PWM_TIME*8-10)
#define TMIN                10
#define T0                  514*8
// Constant table (for bit timming insertion).
const uchar Stuffing[]={0,0x8,0x24,0x25,0xaa,0xab,0xbb,0xfb};
// Global variables definition
uint Pwm_Table[RECURRENCE_SIZE];  // Holds PWM values to be transfered to one shot timer
uchar tempo;  // Used to slow down push switches scanning code .
// -----------------------------------------------------------------------------
void Filling_Dma(value)
{
unsigned char lsb,i;
unsigned int  msb;

lsb = value & 0x07;         // store the 3 lsb bits for table update
msb = (value>>3) & 0x03ff;  // Limit the number to 10bits
lsb = Stuffing[lsb] ;        // Use the lsb bits to address the look-up table
for(i=0;i!= RECURRENCE_SIZE;i++)
  {
  if((lsb & 1)==1)
    Pwm_Table[i] = msb +1;
  else
    Pwm_Table[i] = msb;
  lsb = lsb>>1;
  }
}

void main (void)
{
unsigned int t; // Timer Period

  init();
  tempo = TEMPO;
  t = T0;
  Filling_Dma(T0);              // Initialise DMA table with default value
  timer_a3_init_one_shot_timer_mode ();
  timer_a3_set(Pwm_Table[0]); // Set first PWM value in One Shot timer (Timer A3)
  dma0_init_repeated_transfer_mode(Pwm_Table,(void *)&TA3,8);
  dma0_start();                 // Allow DMA
  timer_b2_init_timer_mode ();
  timer_b2_set (PWM_TIME-1);  // BCLK = 24MHZ => 24KHZ periodic pulse
  timer_b2_start ();          // Start Timer B2 (trig source for Timer A3)
  timer_a3_start ();          // Start Timer A3 (One shot timer)
  for(;;)         // Never ending loop
    {
// Increase duty cycle
```

```
  if(!SW_INT2)
    {
    LED4 = 1; // Light ON LED to show push button is pressed.
    tempo--;
    if(tempo == 0)      // Slow down loop
      {
      tempo = TEMPO;
      if(t<TMAX)  t++;  // Increase duty cycle by one unit
      }
    }
    else LED4 = 0;
// Decrease duty cycle
  if(!SW_INT0)
    {
    LED8 = 1;
    tempo--;
    if(tempo == 0)
      {
      tempo = TEMPO;
      if(t>TMIN) t--;
      }
    }
    else LED6 = 0;
// restore default duty cycle
  if(!SW_INT1)
    {
    LED6 = 1;
    t = T0 ;
    }
    else LED8 = 0;
  Filling_Dma(t);
  };
}
```

Website and Support

Renesas Technology Website
http://www.renesas.com/

Inquiries
http://www.renesas.com/inquiry
csc@renesas.com

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, 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 products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (http://www.renesas.com)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
   (1) artificial life support devices or systems
   (2) surgical implantations
   (3) healthcare intervention (e.g., excision, administration of medication, etc.)
   (4) any other purposes that pose a direct threat to human life

   Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas 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 applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.