EVM Application #2

Creating a Sine Modulated PWM Signal Using the TMS320F240 EVM

APPLICATION REPORT: SPRA411

David Figoli

Digital Signal Processing Solutions
January 1999
IMPORTANT NOTICE

Texas Instruments and its subsidiaries (TI) reserve the right to make changes to their products or to discontinue any product or service without notice, and advise customers to obtain the latest version of relevant information to verify, before placing orders, that information being relied on is current and complete. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgement, including those pertaining to warranty, patent infringement, and limitation of liability.

TI warrants performance of its semiconductor products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

CERTAIN APPLICATIONS USING SEMICONDUCTOR PRODUCTS MAY INVOLVE POTENTIAL RISKS OF DEATH, PERSONAL INJURY, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE ("CRITICAL APPLICATIONS"). TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS. INCLUSION OF TI PRODUCTS IN SUCH APPLICATIONS IS UNDERSTOOD TO BE FULLY AT THE CUSTOMER'S RISK.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards must be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance or customer product design. TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used. TI's publication of information regarding any third party's products or services does not constitute TI's approval, warranty, or endorsement thereof.

Copyright © 1999, Texas Instruments Incorporated
TRADEMARKS

TI is a trademark of Texas Instruments Incorporated.

Other brands and names are the property of their respective owners.
CONTACT INFORMATION

US TMS320 HOTLINE     (281) 274-2320
US TMS320 FAX          (281) 274-2324
US TMS320 BBS          (281) 274-2323
US TMS320 email        dsph@ti.com
Tables

Table 1. Look-Up Table Example 1 .......................................................... 11
Table 2. Look-Up Table Example 2 .......................................................... 11
Abstract

This document describes the EVM Application #2, which generates an asymmetrical pulse width modulated (PWM) signal with a varying duty cycle. This application is implemented using C2xx Assembly code. Included in this document are:

- Modules used
- Input and output information
- Background information on how the sine wave is implemented and generated
- Code used with the target system, the Texas Instruments (TI™) TMS320F240 Evaluation Module (EVM)
Product Support

Related Documentation

The following list specifies product names, part numbers, and literature numbers of corresponding TI documentation.

- Digital Signal Processing Applications with the TMS320 Family: Theory, Algorithms, and Implementations Volume 1, Literature number SPRA012A

World Wide Web

Our World Wide Web site at www.ti.com contains the most up to date product information, revisions, and additions. Users registering with TI&ME can build custom information pages and receive new product updates automatically via email.

Email

For technical issues or clarification on switching products, please send a detailed email to dsph@ti.com. Questions receive prompt attention and are usually answered within one business day.
Overview

This application generates an asymmetrical pulse width modulated (PWM) signal with a varying duty cycle. The period of the PWM signal is 0.05ms, which is equivalent to a 20kHz signal. The duty cycle is modulated with a sine function that can be varied in frequency. The implementation of the sine wave modulation is through a look-up table. This application is implemented using C2xx Assembly code. The algorithm described in this application report was implemented using the TI TMS320F240 EVM.

Module(s) Used

- Event Manager Module
- General Purpose Timer 1

Input

None

Output

T1PWM / T1CMP
Background and Methodology

The implementation of the sine wave modulated PWM signal is simply a modification of Application #1 (PWM0.ASM) except that the compare registers are modified periodically instead of being held constant.

The generation of the sine wave is performed using a look up table. To be able to control the frequency of the modulation with some accuracy, a method based on the modulo mathematical operation is used (i.e. any overflow is disregarded and only the remainder is kept)\(^1\).

In this application a 16-bit counter is used to determine the location of the next value. A step value is added to the counter every time a new value from the sine table is to be loaded. By changing the value of the step, one can accurately control the frequency of the sine wave.

Although a 16-bit counter is used, the upper byte determines the location of the next sine value to be used; thus, by changing how quickly values overflow from the lower byte (i.e., manipulating the step value), the frequency of the sine wave can be changed. The modulo mathematical operation is used when there is overflow in the accumulator from the lower word to the upper word. When an overflow occurs, only the remainder (lower word) is stored.

For example, the counter is set to 0000h and the step value is set to 40h. Every time a value is to be looked up in the table, the value 40h is added to the counter; however, since the upper byte is used as the pointer on the look up table, the first, second, and third values will point to the same location. In the fourth step, which results in an overflow into the upper byte, the value that is loaded will change. Since the upper byte is used as the pointer, the look-up table has 256 values, which is equivalent to the number of possibilities for an 8-bit number: 0 to 255. Additionally, since the upper word of the accumulator is disregarded, the pointer for the sine look up table does not need to be reset.

---

\(^{1}\) For more information reference the Texas Instruments application note, Digital Signal Processing Applications with the TMS320 Family: Theory, Algorithms, and Implementations Volume 1, literature number SPRA012A.
Table 1. Look-Up Table Example 1

<table>
<thead>
<tr>
<th>Step</th>
<th>Accumulator</th>
<th>Counter</th>
<th>Pointer</th>
<th>Step Value = 40h</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0000 0000h</td>
<td>0000h</td>
<td>00h</td>
<td>1st value of sine table</td>
</tr>
<tr>
<td>1</td>
<td>0000 0040h</td>
<td>0040h</td>
<td>00h</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0000 0080h</td>
<td>0080h</td>
<td>00h</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0000 00C0h</td>
<td>00C0h</td>
<td>00h</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0000 0100h</td>
<td>0100h</td>
<td>01h</td>
<td>2nd value of sine table</td>
</tr>
</tbody>
</table>

The step size controls the frequency that is output; as a result, the larger the step, the quicker the overflow into the upper byte, and the faster the pointer traverses through the sine look-up table.

Table 2. Look-Up Table Example 2

<table>
<thead>
<tr>
<th>Step</th>
<th>Counter</th>
<th>Pointer</th>
<th>Step Value = C0h</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0000h</td>
<td>00h</td>
<td>1st value of sine table</td>
</tr>
<tr>
<td>1</td>
<td>00C0h</td>
<td>00h</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0180h</td>
<td>01h</td>
<td>2nd value of sine table</td>
</tr>
<tr>
<td>3</td>
<td>0240h</td>
<td>02h</td>
<td>3rd value of sine table</td>
</tr>
<tr>
<td>4</td>
<td>0300h</td>
<td>03h</td>
<td>4th value of sine table</td>
</tr>
</tbody>
</table>

Although the step size indicates how quickly the pointer moves through the look up table, the step size does not provide much information about the approximate frequency that the sine wave will be modulating the PWM signal. To determine the frequency of the sine wave, determine how often the value in the compare register will be modified.

In this application, the routine to load a new value in the compare register is accessed every time that the timer value matches the value in the period register. Consequently, the routine will be accessed at the same frequency as the PWM signal (20kHz). Because the compare register will be updated each time that the period register and the timer values are equal, the routine that modifies the compare register will be implemented as an interrupt service routine. As a result, the proper registers, EVIMRA and the core IMR need to unmask the proper interrupt levels so that the compare register can be updated.
The frequency that the sine wave will be modulated at can be calculated from the following formula

\[ f(\text{step}) = \frac{\text{step}}{T_S \times 2^n} \]

where:

- \( f(\text{step}) \) = desired frequency
- \( T_S \) = the time period between each update (in this case, the PWM period)
- \( n \) = the number of bits in the counter register
- \( \text{step} \) = the step size used

The frequency that the PWM signal will be modulated is proportional to the step size and inversely proportional to the size of the counter register and the period at which the routine is accessed. Thus, to increase the resolution that one can increment or decrement the frequency of the PWM modulation, one needs to have a larger counting register or access the routine at a slower frequency by increasing the period.

Since this program is interrupt driven, once the registers have been set for the PWM signal, the program can be ended with an unconditional branch and the output will continue because of the interrupt structure. The output will stop when the user halts the program or the software masks the corresponding interrupt levels.

Adding the following watch in the debugger environment

\[ \text{wa } *\text{FREQSTEP},u \]

will allow one to modify the step size to change the frequency of modulation.
; File Name: pwm1.asm
; Originator: Digital Control systems Apps group - Houston
; Target System: 'C240 Evaluation Board
;
; Description: Pulse Width Modulator - Sets up the registers
; for an asymmetric PWM output. The output is a
; square wave with a sine wave modulated duty cycle.
; PWM Period is 0.05ms => 20kHz
;
; Entering the command
; wa *FREQSTEP,,u
; Allows one to change the step size to change
; the frequency in the debugger environment
;
; Last Updated: 20 June 1997
;
.include f240regs.h

; Vector address declarations

.sect ".vectors"

RSVECT   B START   ; Reset Vector
INT1     B PHANTOM; Interrupt Level 1
INT2     B SINE    ; Interrupt Level 2
INT3     B PHANTOM; Interrupt Level 3
INT4     B PHANTOM; Interrupt Level 4
INT5     B PHANTOM; Interrupt Level 5
INT6     B PHANTOM; Interrupt Level 6
RESERVED B PHANTOM; Reserved
SW_INT8  B PHANTOM; User S/W Interrupt
SW_INT9  B PHANTOM; User S/W Interrupt
SW_INT10 B PHANTOM; User S/W Interrupt
SW_INT11 B PHANTOM; User S/W Interrupt
SW_INT12 B PHANTOM; User S/W Interrupt
SW_INT13 B PHANTOM; User S/W Interrupt
SW_INT14 B PHANTOM; User S/W Interrupt
SW_INT15 B PHANTOM; User S/W Interrupt
SW_INT16 B PHANTOM; User S/W Interrupt
TRAP     B PHANTOM; Trap vector
NMINT    B PHANTOM; Non-maskable Interrupt
EMU_TRAP B PHANTOM; Emulator Trap
SW_INT20 B PHANTOM; User S/W Interrupt
SW_INT21 B PHANTOM; User S/W Interrupt
SW_INT22 B PHANTOM; User S/W Interrupt
SW_INT23 B PHANTOM; User S/W Interrupt
;==--------------------------------------------------------
;  M A I N   C O D E  - starts here
;==--------------------------------------------------------

.text
NOP

START:    SETC INTM ;Disable interrupts
SPLK #0002h,IMR ;Mask all core interrupts
; except INT2

LACC IFR ;Read Interrupt flags
SAACL IFR ;Clear all interrupt flags

CLRC SXM ;Clear Sign Extension Mode
CLRC OVM ;Reset Overflow Mode
CLRC CNF ;Config Block B0 to Data mem

;--------------------------------------------------------
; Set up PLL Module
;--------------------------------------------------------

LDP   #00E0h

;The following line is necessary if a previous program set the PLL
;to a different ;setting than the settings which the application
;uses. By disabling the PLL, the CKCR1 register can be modified so
;that the PLL can run at the new settings when it is re-enabled.

SPLK #0000000000100001b,CKCR0 ;CLKMD=PLL Disable
  ;SYSCLK=CPUCLK/2

;  5432109876543210
SPLK #0000000010111011b,CKCR1
  ;CLKIN(OSC)=10MHz,CPUCLK=20MHz

;CKCR1 - Clock Control Register 1
; Bits 7-4 (1011) CKINF(3)-CKINF(0) - Crystal or Clock-In
; Frequency
; Frequency = 10MHz
; Bit 3 (1) PLLDIV(2) - PLL divide by 2 bit
; Divide PLL input by 2
; Bits 2-0 (011) PLLFB(2)-PLLFB(0) - PLL multiplication ratio
; PLL Multiplication Ration = 4

;  5432109876543210
SPLK #0000000010111011b,CKCR1
  ;CLKIN(OSC)=10MHz,CPUCLK=20MHz

;CKCR0 - Clock Control Register 0
; Bits 7-6 (11) CLKMD(1),CLKMD(0) - Operational mode of
; Clock Module
; PLL Enabled; Run on CLKin on exiting low power mode
; Bits 5-4 (00) PLLOCK(1),PLLOCK(0) - PLL Status.
; READ ONLY
; Bits 3-2 (00) PLLPM(1),PLLPM(0) - Low Power Mode
; LPM0
; Bit 1 (0) ACLKENA - 1MHz ACLK Enable
; ACLK Disabled
; Bit 0 (1) PLLPS - System Clock Prescale Value
; \[ f(sysclk) = \frac{f(cpuclk)}{2} \]

; 5432109876543210
SPLK #0100000011000000b,SYSCR ;CLKOUT=CPUCLK

;SYSCR - System Control Register
; Bit 15-14 (01) RESET1,RESET0 - Software Reset Bits
; No Action
; Bits 13-8 (000000) Reserved
; Bit 7-6 (11) CLKSRC1,CLKSRC0 - CLKOUT-Pin Source
; Select
; CPUClk: CPU clock output mode
; Bit 5-0 (000000)Reserved

;SYSCR - System Control Register
; Bit 15-14 (01) RESET1,RESET0 - Software Reset Bits
; No Action
; Bits 13-8 (000000) Reserved
; Bit 7-6 (11) CLKSRC1,CLKSRC0 - CLKOUT-Pin Source
; Select
; CPUClk: CPU clock output mode
; Bit 5-0 (000000)Reserved

SPLK #006Fh, WDCR ;Disable WD if VCCP=5V
; (JP5 in pos. 2-3)
KICK_DOG ;Reset Watchdog

;-----------------------------------
; Set up Digital I/O Port
;-----------------------------------
LDP #225 ;DP=225, Data Page to Configure OCRA

SPLK #0011000000000000b,OCRA

;OCRA - Output Control Register A
; Bit 15 (0) CRA.15 - IOPB7
; Bit 14 (0) CRA.14 - IOPB6
; Bit 13 (1) CRA.13 - T3PWM/T3CMP
; Bit 12 (1) CRA.12 - T2PWM/T2CMP
; Bit 11 (1) CRA.11 - T1PWM/T1CMP
; Bit 10 (0) CRA.10 - IOPB2
; Bit  9 (0) CRA. 9 - IOPB1
; Bit  8 (0) CRA. 8 - IOPB0
; Bits 7-4 (0000) Reserved
; Bit  3 (0) CRA. 3 - IOPA3
; Bit  2 (0) CRA. 2 - IOPA2
; Bit  1 (0) CRA. 1 - IOPA1
; Bit  0 (0) CRA. 0 - IOPA0

;*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
; - Event Manager Module Reset
;*
This section resets all of the Event Manager Module Registers.
This is necessary for silicon revision 1.1; however, for
silicon revisions 2.0 and later, this is not necessary

LDP #232 ;DP=232, Data Page for the Event Manager
SPLK #0000h,GPTCON ;Clear General Purpose Timer Control
SPLK #0000h,T1CON ;Clear GP Timer 1 Control
SPLK #0000h,T2CON ;Clear GP Timer 2 Control
SPLK #0000h,T3CON ;Clear GP Timer 3 Control
SPLK #0000h,COMCON ;Clear Compare Control
SPLK #0000h,ACTR
;Clear Full Compare Action Control Register
SPLK #0000h,SACTR
;Clear Simple Compare Action Control Register
SPLK #0000h,DBTCON
;Clear Dead-Band Timer Control Register
SPLK #0000h,CAPCON ;Clear Capture Control
SPLK #FFFFh,EVIFRA ;Clear Interrupt Flag Register A
SPLK #FFFFh,EVIFRB ;Clear Interrupt Flag Register B
SPLK #FFFFh,EVIFRC ;Clear Interrupt Flag Register C
SPLK #0000h,EVIMRA ;Clear Event Manager Mask Register A
SPLK #0000h,EVIMRB ;Clear Event Manager Mask Register B
SPLK #0000h,EVIMRC ;Clear Event Manager Mask Register C

T1COMPARE .set 0 ; T1Compare Initialized to 0
T1PERIOD .set 1000 ; T1Period Initialized to
; 1000 = 20kHz value

.text
LDP #232 ;DP=232, Data Page for
; Event Manager Addresses
SPLK #T1COMPARE,T1CMPR;T1CMPR = 0

2109876543210
SPLK #0000010101010b,GPTCON

;GPTCON - GP Timer Control Register
; Bit 15 (0) T3STAT - GP Timer 3 Status. READ ONLY
; Bit 14 (0) T2STAT - GP Timer 2 Status. READ ONLY
; Bit 13 (0) T1STAT - GP Timer 1 Status. READ ONLY
; Bits 12-11 (00) T3TOADC - ADC start by event of GP Timer 3
; No event starts ADC
; Bits 10-9 (00) T2TOADC - ADC start by event of GP Timer 2
; No event starts ADC
; Bits 8-7 (00) T1TOADC - ADC start by event of GP Timer 1
; No event starts ADC
; Bit 6 (1) TCOMPOE - Compare output enable
; Enable all three GP timer compare outputs
; Bits 5-4 (01) T3PIN - Polarity of GP Timer 3 compare output
; Active Low
; Bits 3-2 (01) T2PIN - Polarity of GP Timer 2 compare output
; Active Low
; Bits 1-0 (01) T1PIN - Polarity of GP Timer 1 compare output
; Active Low

SPLK #T1PERIOD,T1PR ; T1PR = 1000
SPLK #0000h,T1CNT ; Initialize Timer 1
SPLK #0000h,T2CNT ; Initialize Timer 2
SPLK #0000h,T3CNT ; Initialize Timer 3

; T1CON - GP Timer 1 Control Register
; Bits 15-14 (00) FREE,SOFT - Emulation Control Bits
; Stop immediately on emulation suspend
; Bits 13-11 (010) TMODE2-TMODE0 - Count Mode Selection
; Continuous-Up Count Mode
; Bits 10-8 (000) TPS2-TPS0 - Input Clock Prescaler
; Divide by 1
; Bit 7 (0) Reserved
; Bit 6 (0) TENABLE - Timer Enable
; Disable timer operations
; Bits 5-4 (00) TCLKS1,TCLKS0 - Clock Source Select
; Internal Clock Source
; Bits 3-2 (01) TCLD1,TCLD0 - Timer Compare Register
; Reload Condition
; When counter is 0 or equals period register value
; Bit 1 (1) TECMPR - Timer compare enable
; Enable timer compare operation
; Bit 0 (0) Reserved

; SPLK #0001000000000000b,T1CON ;Not used
;T2CON - GP Timer 2 Control Register
; Bits 15-14 (00) FREE,SOFT - Emulation Control Bits
; Stop immediately on emulation suspend
; Bits 13-11 (000) TMODE2-TMODE0 - Count Mode Selection
; Stop/Hold
; Bits 10-8 (000) TPS2-TPS0 - Input Clock Prescaler
; Divide by 1
; Bit 7 (0) TSWT1 - GP Timer 1 timer enable bit
; Use own TENABLE bit
; Bit 6 (0) TENABLE - Timer Enable
; Disable timer operations
; Bits 5-4 (00) TCLKS1,TCLKS0 - Clock Source Select
; Internal Clock Source
; Bits 3-2 (00) TCLD1,TCLD0 - Timer Compare Register
; Reload Condition When counter is 0
; Bit 1 (0) TECMPR - Timer compare enable
; Disable timer compare operation
; Bit 0 (0) SELT1PR - Period Register select
; Use own period register

; 5432109876543210
SPLK #000000000000000b,T3CON ;Not Used

;T3CON - GP Timer 3 Control Register
; Bits 15-14 (00) FREE,SOFT - Emulation Control Bits
; Stop immediately on emulation suspend
; Bits 13-11 (000) TMODE2-TMODE0 - Count Mode Selection
; Stop/Hold
; Bits 10-8 (000) TPS2-TPS0 - Input Clock Prescaler
; Divide by 1
; Bit 7 (0) TSWT1 - GP Timer 1 timer enable bit
; Use own TENABLE bit
; Bit 6 (0) TENABLE - Timer Enable
; Disable timer operations
; Bits 5-4 (00) TCLKS1,TCLKS0 - Clock Source Select
; Internal Clock Source
; Bits 3-2 (00) TCLD1,TCLD0 - Timer Compare Register
; Reload Condition When counter is 0
; Bit 1 (0) TECMPR - Timer compare enable
; Disable timer compare operation
; Bit 0 (0) SELT1PR - Period Register select
; Use own period register

SBIT1 T1CON,B6_MSK ;Sets Bit 6 of T1CON

;T1CON - GP Timer 1 Control Register
; Bit 6 (1) TENABLE - Timer Enable
; Enable Timer Operations

SPLK #0FFFFh,EVFIRA ;Clear all pending interrupts
SPLK #0080h,EVIMRA ;Enable Timer 1 Period Interrupt
;-----------------------------------
; Generate Sine Wave Modulated PWM
;-----------------------------------

.bss TABLE,1 ;Keeps address of the pointer in
;the SINE Table
.bss TOPTABLE,1 ;Keeps the reset value for the
:pointer
.bss COMPARET1,1 ;A register to do calculations
;since the
;TelCMPR register is double
;buffered
.bss FREQSTEP,1 ;Frequency modulation of the sine
;wave
.bss MODREG,1 ;Rolling Modulo Register
.bss SINEVAL,1 ;Value from look up table

NORMAL .set 500

.text

LDP #0 ;DP = 0
SPLK #0000h,TABLE ;Initialize Pointer to Top
SPLK #STABLE,TOPTABLE ;Initialize TOPTABLE to
; address of sine table
SPLK #4,FREQSTEP ;Set the step size to 4
SPLK #0000h,MODREG ;Initialize the 16 bit
; counter register

CLRC INTM ;Enable interrupts

END B END ;End of Program

;-------------------------------------------------------------------
; Generate PWM Sine Wave ISR
;-------------------------------------------------------------------

SINE LDP #0 ; DP = 0
LACC MODREG ; ACC = 16 bit Counter Register
ADD FREQSTEP ; ACC = Counter + Step
SACL MODREG ; Counter assigned new value
LACC MODREG,8 ; ACC = Counter shifted to left
; by 8
SACH TABLE ; TABLE = upper byte of
; counter = pointer
LACC TABLE ; ACC = TABLE = Pointer
ADD TOPTABLE ; Offset Addr from top of table
TBLR SINEVAL ; Read sine value and store to
; SINEVAL
; Normalization of the Sine value to prevent the compare value from being negative

LT SINEVAL ; TREG = SINEVAL (Q15)
MPY #NORMAL ; PREG = TREG * NORMAL (Q30)
    ; NORMAL = T/2
PAC ; ACC = PREG (Q30)
SACH COMPARET1,1 ; COMPARET1 = PREG (Q15)
LACC COMPARET1 ; ACC = COMPARET1
ADD #NORMAL ; ACC = COMPARET1 + NORMAL

LDP #232 ; DP = 232
SACL T1CMPR ; T1CMPR = ACC = Normalize Sine Value

; Clear the interrupt flags of the Event Manager Module

LACC EVIFRA ; ACC = EVIFRA
SACL EVIFRA ; EVIFRA = ACC; resets the interrupt flag
CLRC INTM ; Enable core interrupts
RET ; Return to end of program
<table>
<thead>
<tr>
<th>Index</th>
<th>Angle</th>
<th>Sin(Angle)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0.0000</td>
</tr>
<tr>
<td>1</td>
<td>1.41</td>
<td>0.0245</td>
</tr>
<tr>
<td>2</td>
<td>2.81</td>
<td>0.0491</td>
</tr>
<tr>
<td>3</td>
<td>4.22</td>
<td>0.0736</td>
</tr>
<tr>
<td>4</td>
<td>5.63</td>
<td>0.0980</td>
</tr>
<tr>
<td>5</td>
<td>7.03</td>
<td>0.1224</td>
</tr>
<tr>
<td>6</td>
<td>8.44</td>
<td>0.1467</td>
</tr>
<tr>
<td>7</td>
<td>9.84</td>
<td>0.1710</td>
</tr>
<tr>
<td>8</td>
<td>11.25</td>
<td>0.1951</td>
</tr>
<tr>
<td>9</td>
<td>12.66</td>
<td>0.2191</td>
</tr>
<tr>
<td>10</td>
<td>14.06</td>
<td>0.2430</td>
</tr>
<tr>
<td>11</td>
<td>15.47</td>
<td>0.2667</td>
</tr>
<tr>
<td>12</td>
<td>16.88</td>
<td>0.2903</td>
</tr>
<tr>
<td>13</td>
<td>18.28</td>
<td>0.3137</td>
</tr>
<tr>
<td>14</td>
<td>19.69</td>
<td>0.3369</td>
</tr>
<tr>
<td>15</td>
<td>21.09</td>
<td>0.3599</td>
</tr>
<tr>
<td>16</td>
<td>22.50</td>
<td>0.3827</td>
</tr>
<tr>
<td>17</td>
<td>23.91</td>
<td>0.4052</td>
</tr>
<tr>
<td>18</td>
<td>25.31</td>
<td>0.4276</td>
</tr>
<tr>
<td>19</td>
<td>26.72</td>
<td>0.4496</td>
</tr>
<tr>
<td>20</td>
<td>28.13</td>
<td>0.4714</td>
</tr>
<tr>
<td>21</td>
<td>29.53</td>
<td>0.4929</td>
</tr>
<tr>
<td>22</td>
<td>30.94</td>
<td>0.5141</td>
</tr>
<tr>
<td>23</td>
<td>32.34</td>
<td>0.5350</td>
</tr>
<tr>
<td>24</td>
<td>33.75</td>
<td>0.5556</td>
</tr>
<tr>
<td>25</td>
<td>35.16</td>
<td>0.5758</td>
</tr>
<tr>
<td>26</td>
<td>36.56</td>
<td>0.5957</td>
</tr>
<tr>
<td>27</td>
<td>37.97</td>
<td>0.6152</td>
</tr>
<tr>
<td>28</td>
<td>39.38</td>
<td>0.6344</td>
</tr>
<tr>
<td>29</td>
<td>40.78</td>
<td>0.6532</td>
</tr>
<tr>
<td>30</td>
<td>42.19</td>
<td>0.6716</td>
</tr>
<tr>
<td>31</td>
<td>43.59</td>
<td>0.6895</td>
</tr>
<tr>
<td>32</td>
<td>45.00</td>
<td>0.7071</td>
</tr>
<tr>
<td>33</td>
<td>46.41</td>
<td>0.7242</td>
</tr>
<tr>
<td>34</td>
<td>47.81</td>
<td>0.7410</td>
</tr>
<tr>
<td>35</td>
<td>49.22</td>
<td>0.7572</td>
</tr>
<tr>
<td>36</td>
<td>50.63</td>
<td>0.7730</td>
</tr>
<tr>
<td>37</td>
<td>52.03</td>
<td>0.7883</td>
</tr>
<tr>
<td>38</td>
<td>53.44</td>
<td>0.8032</td>
</tr>
<tr>
<td>39</td>
<td>54.84</td>
<td>0.8176</td>
</tr>
<tr>
<td>40</td>
<td>56.25</td>
<td>0.8315</td>
</tr>
<tr>
<td>41</td>
<td>57.66</td>
<td>0.8449</td>
</tr>
<tr>
<td>42</td>
<td>59.06</td>
<td>0.8577</td>
</tr>
<tr>
<td>43</td>
<td>60.47</td>
<td>0.8701</td>
</tr>
<tr>
<td>44</td>
<td>61.88</td>
<td>0.8819</td>
</tr>
</tbody>
</table>
Creating a Sine Modulated PWM Signal Using the TMS320F240 EVM
<table>
<thead>
<tr>
<th>Address</th>
<th>Value</th>
<th>Frequency</th>
<th>Duty Cycle</th>
</tr>
</thead>
<tbody>
<tr>
<td>22594</td>
<td>97</td>
<td>136.41</td>
<td>0.6895</td>
</tr>
<tr>
<td>22005</td>
<td>98</td>
<td>137.81</td>
<td>0.6716</td>
</tr>
<tr>
<td>21403</td>
<td>99</td>
<td>139.22</td>
<td>0.6532</td>
</tr>
<tr>
<td>20787</td>
<td>100</td>
<td>140.63</td>
<td>0.6344</td>
</tr>
<tr>
<td>20159</td>
<td>101</td>
<td>142.03</td>
<td>0.6152</td>
</tr>
<tr>
<td>19519</td>
<td>102</td>
<td>143.44</td>
<td>0.5957</td>
</tr>
<tr>
<td>18868</td>
<td>103</td>
<td>144.84</td>
<td>0.5758</td>
</tr>
<tr>
<td>18204</td>
<td>104</td>
<td>146.25</td>
<td>0.5556</td>
</tr>
<tr>
<td>17530</td>
<td>105</td>
<td>147.66</td>
<td>0.5350</td>
</tr>
<tr>
<td>16846</td>
<td>106</td>
<td>149.06</td>
<td>0.5141</td>
</tr>
<tr>
<td>16151</td>
<td>107</td>
<td>150.47</td>
<td>0.4929</td>
</tr>
<tr>
<td>15446</td>
<td>108</td>
<td>151.88</td>
<td>0.4714</td>
</tr>
<tr>
<td>14732</td>
<td>109</td>
<td>153.28</td>
<td>0.4496</td>
</tr>
<tr>
<td>14010</td>
<td>110</td>
<td>154.69</td>
<td>0.4276</td>
</tr>
<tr>
<td>13279</td>
<td>111</td>
<td>156.09</td>
<td>0.4052</td>
</tr>
<tr>
<td>12539</td>
<td>112</td>
<td>157.50</td>
<td>0.3827</td>
</tr>
<tr>
<td>11793</td>
<td>113</td>
<td>158.91</td>
<td>0.3599</td>
</tr>
<tr>
<td>11039</td>
<td>114</td>
<td>160.31</td>
<td>0.3369</td>
</tr>
<tr>
<td>10278</td>
<td>115</td>
<td>161.72</td>
<td>0.3137</td>
</tr>
<tr>
<td>9512</td>
<td>116</td>
<td>163.13</td>
<td>0.2903</td>
</tr>
<tr>
<td>8739</td>
<td>117</td>
<td>164.53</td>
<td>0.2667</td>
</tr>
<tr>
<td>7962</td>
<td>118</td>
<td>165.94</td>
<td>0.2430</td>
</tr>
<tr>
<td>7179</td>
<td>119</td>
<td>167.34</td>
<td>0.2191</td>
</tr>
<tr>
<td>6393</td>
<td>120</td>
<td>168.75</td>
<td>0.1951</td>
</tr>
<tr>
<td>5602</td>
<td>121</td>
<td>170.16</td>
<td>0.1710</td>
</tr>
<tr>
<td>4808</td>
<td>122</td>
<td>171.56</td>
<td>0.1467</td>
</tr>
<tr>
<td>4011</td>
<td>123</td>
<td>172.97</td>
<td>0.1224</td>
</tr>
<tr>
<td>3212</td>
<td>124</td>
<td>174.38</td>
<td>0.0980</td>
</tr>
<tr>
<td>2410</td>
<td>125</td>
<td>175.78</td>
<td>0.0736</td>
</tr>
<tr>
<td>1608</td>
<td>126</td>
<td>177.19</td>
<td>0.0491</td>
</tr>
<tr>
<td>804</td>
<td>127</td>
<td>178.59</td>
<td>0.0245</td>
</tr>
<tr>
<td>0</td>
<td>128</td>
<td>180.00</td>
<td>0.0000</td>
</tr>
<tr>
<td>64731</td>
<td>129</td>
<td>181.41</td>
<td>-0.0245</td>
</tr>
<tr>
<td>63927</td>
<td>130</td>
<td>182.81</td>
<td>-0.0491</td>
</tr>
<tr>
<td>63125</td>
<td>131</td>
<td>184.22</td>
<td>-0.0736</td>
</tr>
<tr>
<td>62323</td>
<td>132</td>
<td>185.63</td>
<td>-0.0980</td>
</tr>
<tr>
<td>61524</td>
<td>133</td>
<td>187.03</td>
<td>-0.1224</td>
</tr>
<tr>
<td>60727</td>
<td>134</td>
<td>188.44</td>
<td>-0.1467</td>
</tr>
<tr>
<td>59933</td>
<td>135</td>
<td>189.84</td>
<td>-0.1710</td>
</tr>
<tr>
<td>59142</td>
<td>136</td>
<td>191.25</td>
<td>-0.1951</td>
</tr>
<tr>
<td>58356</td>
<td>137</td>
<td>192.66</td>
<td>-0.2191</td>
</tr>
<tr>
<td>57573</td>
<td>138</td>
<td>194.06</td>
<td>-0.2430</td>
</tr>
<tr>
<td>56796</td>
<td>139</td>
<td>195.47</td>
<td>-0.2667</td>
</tr>
<tr>
<td>56023</td>
<td>140</td>
<td>196.88</td>
<td>-0.2903</td>
</tr>
<tr>
<td>55257</td>
<td>141</td>
<td>198.28</td>
<td>-0.3137</td>
</tr>
<tr>
<td>54496</td>
<td>142</td>
<td>199.69</td>
<td>-0.3369</td>
</tr>
<tr>
<td>53742</td>
<td>143</td>
<td>201.09</td>
<td>-0.3599</td>
</tr>
<tr>
<td>52996</td>
<td>144</td>
<td>202.50</td>
<td>-0.3827</td>
</tr>
<tr>
<td>52256</td>
<td>145</td>
<td>203.91</td>
<td>-0.4052</td>
</tr>
<tr>
<td>51525</td>
<td>146</td>
<td>205.31</td>
<td>-0.4276</td>
</tr>
<tr>
<td>50803</td>
<td>147</td>
<td>206.72</td>
<td>-0.4496</td>
</tr>
<tr>
<td>50089</td>
<td>148</td>
<td>208.13</td>
<td>-0.4714</td>
</tr>
</tbody>
</table>
Creating a Sine Modulated PWM Signal Using the TMS320F240 EVM
<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>.word</td>
<td>33564</td>
<td>;</td>
<td>201</td>
<td>282.66</td>
</tr>
<tr>
<td>.word</td>
<td>33750</td>
<td>;</td>
<td>202</td>
<td>284.06</td>
</tr>
<tr>
<td>.word</td>
<td>33955</td>
<td>;</td>
<td>203</td>
<td>285.47</td>
</tr>
<tr>
<td>.word</td>
<td>34179</td>
<td>;</td>
<td>204</td>
<td>286.88</td>
</tr>
<tr>
<td>.word</td>
<td>34422</td>
<td>;</td>
<td>205</td>
<td>288.28</td>
</tr>
<tr>
<td>.word</td>
<td>34683</td>
<td>;</td>
<td>206</td>
<td>289.69</td>
</tr>
<tr>
<td>.word</td>
<td>34964</td>
<td>;</td>
<td>207</td>
<td>291.09</td>
</tr>
<tr>
<td>.word</td>
<td>35262</td>
<td>;</td>
<td>208</td>
<td>292.50</td>
</tr>
<tr>
<td>.word</td>
<td>35579</td>
<td>;</td>
<td>209</td>
<td>293.91</td>
</tr>
<tr>
<td>.word</td>
<td>35914</td>
<td>;</td>
<td>210</td>
<td>295.31</td>
</tr>
<tr>
<td>.word</td>
<td>36267</td>
<td>;</td>
<td>211</td>
<td>296.72</td>
</tr>
<tr>
<td>.word</td>
<td>36637</td>
<td>;</td>
<td>212</td>
<td>298.13</td>
</tr>
<tr>
<td>.word</td>
<td>37025</td>
<td>;</td>
<td>213</td>
<td>299.53</td>
</tr>
<tr>
<td>.word</td>
<td>37430</td>
<td>;</td>
<td>214</td>
<td>300.94</td>
</tr>
<tr>
<td>.word</td>
<td>37852</td>
<td>;</td>
<td>215</td>
<td>302.34</td>
</tr>
<tr>
<td>.word</td>
<td>38290</td>
<td>;</td>
<td>216</td>
<td>303.75</td>
</tr>
<tr>
<td>.word</td>
<td>38745</td>
<td>;</td>
<td>217</td>
<td>305.16</td>
</tr>
<tr>
<td>.word</td>
<td>39216</td>
<td>;</td>
<td>218</td>
<td>306.56</td>
</tr>
<tr>
<td>.word</td>
<td>39703</td>
<td>;</td>
<td>219</td>
<td>307.97</td>
</tr>
<tr>
<td>.word</td>
<td>40206</td>
<td>;</td>
<td>220</td>
<td>309.38</td>
</tr>
<tr>
<td>.word</td>
<td>40724</td>
<td>;</td>
<td>221</td>
<td>310.78</td>
</tr>
<tr>
<td>.word</td>
<td>41256</td>
<td>;</td>
<td>222</td>
<td>312.19</td>
</tr>
<tr>
<td>.word</td>
<td>41804</td>
<td>;</td>
<td>223</td>
<td>313.59</td>
</tr>
<tr>
<td>.word</td>
<td>42365</td>
<td>;</td>
<td>224</td>
<td>315.00</td>
</tr>
<tr>
<td>.word</td>
<td>42941</td>
<td>;</td>
<td>225</td>
<td>316.41</td>
</tr>
<tr>
<td>.word</td>
<td>43530</td>
<td>;</td>
<td>226</td>
<td>317.81</td>
</tr>
<tr>
<td>.word</td>
<td>44132</td>
<td>;</td>
<td>227</td>
<td>319.22</td>
</tr>
<tr>
<td>.word</td>
<td>44748</td>
<td>;</td>
<td>228</td>
<td>320.63</td>
</tr>
<tr>
<td>.word</td>
<td>45376</td>
<td>;</td>
<td>229</td>
<td>322.03</td>
</tr>
<tr>
<td>.word</td>
<td>46016</td>
<td>;</td>
<td>230</td>
<td>323.44</td>
</tr>
<tr>
<td>.word</td>
<td>46667</td>
<td>;</td>
<td>231</td>
<td>324.84</td>
</tr>
<tr>
<td>.word</td>
<td>47331</td>
<td>;</td>
<td>232</td>
<td>326.25</td>
</tr>
<tr>
<td>.word</td>
<td>48005</td>
<td>;</td>
<td>233</td>
<td>327.66</td>
</tr>
<tr>
<td>.word</td>
<td>48689</td>
<td>;</td>
<td>234</td>
<td>329.06</td>
</tr>
<tr>
<td>.word</td>
<td>49384</td>
<td>;</td>
<td>235</td>
<td>330.47</td>
</tr>
<tr>
<td>.word</td>
<td>50089</td>
<td>;</td>
<td>236</td>
<td>331.88</td>
</tr>
<tr>
<td>.word</td>
<td>50803</td>
<td>;</td>
<td>237</td>
<td>333.28</td>
</tr>
<tr>
<td>.word</td>
<td>51525</td>
<td>;</td>
<td>238</td>
<td>334.69</td>
</tr>
<tr>
<td>.word</td>
<td>52256</td>
<td>;</td>
<td>239</td>
<td>336.09</td>
</tr>
<tr>
<td>.word</td>
<td>52996</td>
<td>;</td>
<td>240</td>
<td>337.50</td>
</tr>
<tr>
<td>.word</td>
<td>53742</td>
<td>;</td>
<td>241</td>
<td>338.91</td>
</tr>
<tr>
<td>.word</td>
<td>54496</td>
<td>;</td>
<td>242</td>
<td>340.31</td>
</tr>
<tr>
<td>.word</td>
<td>55257</td>
<td>;</td>
<td>243</td>
<td>341.72</td>
</tr>
<tr>
<td>.word</td>
<td>56023</td>
<td>;</td>
<td>244</td>
<td>343.13</td>
</tr>
<tr>
<td>.word</td>
<td>56796</td>
<td>;</td>
<td>245</td>
<td>344.53</td>
</tr>
<tr>
<td>.word</td>
<td>57573</td>
<td>;</td>
<td>246</td>
<td>345.94</td>
</tr>
<tr>
<td>.word</td>
<td>58356</td>
<td>;</td>
<td>247</td>
<td>347.34</td>
</tr>
<tr>
<td>.word</td>
<td>59142</td>
<td>;</td>
<td>248</td>
<td>348.75</td>
</tr>
<tr>
<td>.word</td>
<td>59933</td>
<td>;</td>
<td>249</td>
<td>350.16</td>
</tr>
<tr>
<td>.word</td>
<td>60727</td>
<td>;</td>
<td>250</td>
<td>351.56</td>
</tr>
<tr>
<td>.word</td>
<td>61524</td>
<td>;</td>
<td>251</td>
<td>352.97</td>
</tr>
<tr>
<td>.word</td>
<td>62323</td>
<td>;</td>
<td>252</td>
<td>354.38</td>
</tr>
</tbody>
</table>
Creating a Sine Modulated PWM Signal Using the TMS320F240 EVM

.word   63125        ;         253      355.78   -0.0736
.word   63927        ;         254      357.19   -0.0491
.word   64731        ;         255      358.59   -0.0245
.word   65535        ;         256      360.00    0.0000

;===================================================================
; I S R   -   PHANTOM
;
; Description:  Dummy ISR, used to trap spurious interrupts.
;
; Modifies:  Nothing
;
; Last Update:  16 June 95
;===================================================================

PHANTOM     KICK_DOG           ;Resets WD counter
            B       PHANTOM