Pulse Width Modulation for the HC9S12

These instructions pertain to the MC9S12DP256B derivative of the family of HCS12 Microcontrollers

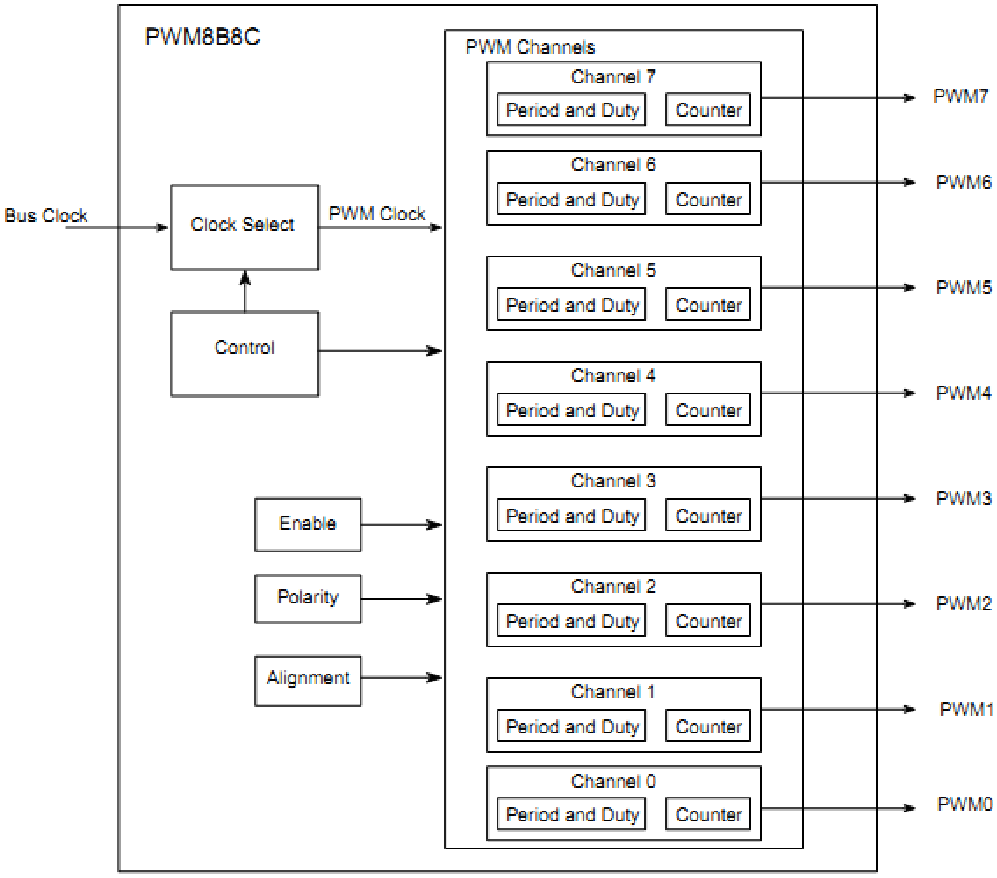
Use of PWM

Pulse Width Modulation has been previously seen as a way to control the speed of a DC motor by varying the Duty Cycle of a Square wave connected to the motor.

The PWM Features

- Eight independent PWM channels with programmable period and duty cycle
- Dedicated counter for each PWM channel and programmable PWM enable/disable for each channel
- Software selection of PWM duty pulse polarity for each channel
- Programmable center or left aligned outputs on individual channels
- Eight 8-bit channels or four 16-bit channels PWM resolution
- Four clock sources for a wide range of frequencies with programmable clock select logic

PWM Block Diagram



Programming the PWM Module

Programming the PWM module requires that the user configure a selection of registers to control such things as the Clock source selection, output waveform polarity, alignment, period and duty cycle. There is also a register to enable the channel(s) of the PWM module to be used.

PWME (PWM Enable) Register – address $00A0

This 8 bit register enables various PWM channels.

The location of the bits in this register is shown.

PWME7	PWME6	PWME5	PWME4	PWME3	PWME2	PWME1	PWME0

PWME7 – PWME0 PWM Channel Enable	
PWME7	1 = PWM channel 7 is enabled; 0 = PWM channel 7 is disabled
PWME6	1 = PWM channel 6 is enabled; 0 = PWM channel 6 is disabled
PWME5	1 = PWM channel 5 is enabled; 0 = PWM channel 5 is disabled
PWME4	1 = PWM channel 4 is enabled; 0 = PWM channel 4 is disabled
PWME3	1 = PWM channel 3 is enabled; 0 = PWM channel 3 is disabled
PWME2	1 = PWM channel 2 is enabled; 0 = PWM channel 2 is disabled
PWME1	1 = PWM channel 1 is enabled; 0 = PWM channel 1 is disabled
PWME0	1 = PWM channel 0 is enabled; 0 = PWM channel 0 is disabled

PWM Clock Sources – CLOCK A and CLOCKB

The PWM module has four clock inputs called ClockA, CLOCKB, CLOCKSA and ClockSB that are derived from the 24 MHz Bus clock. Collectively these are referred to as the PWM clock. ClockA and ClockSA are used for PWM channels 0, 1, 4, and 5. ClockB and ClockSB are used for PWM channels 2, 3, 6, and 7. ClockA (and ClockB) are generated by subdividing the Bus Clock by the use of a prescale factor in the PWMPRCLK register. The minimum pre-scale value is 1 and the maximum is 128. Additional subdividing of ClockA (and ClockB) will produce ClockSA and ClockSB frequencies .

ClockSA

To produce a lower frequency for the PWM channels, CLOCKA can be further subdivided to produce clock SA by using the PWMSCLA register. The highest number for dividing CLOCKA is 512 and the lowest number is 2.

ClockSB

To produce a lower frequency for the PWM channels, the CLOCKB can be further subdivided to produce clock SB by using the PWMSCLB register. The highest number for dividing CLOCKB is 512 and the lowest number is 2.

PWMPRCLK (PWM Prescaler Clock) Register – address $00A3

The 8 bits in this register can be used to generate the ClockA and ClockB pre-scale values according to the table.

0	PCKB2	PCKB1	PCKB0	0	PCKA2	PCKA1	PCKA0

Prescaling ClockA

PCKA2	**PCKA1**	**PCKA0**	**Prescale for ClockA**
0	0	0	ClockA = Bus clock
0	0	1	ClockA = Bus clock/2
0	1	0	ClockA = Bus clock/4
0	1	1	ClockA = Bus clock/8
1	0	0	ClockA = Bus clock/16
1	0	1	ClockA = Bus clock/32
1	1	0	ClockA = Bus clock/64
1	1	1	ClockA = Bus clock/128

Prescaling ClockB

PCKB2	**PCKB1**	**PCKB0**	**Prescale for ClockB**
0	0	0	ClockB = Bus clock
0	0	1	ClockB = Bus clock/2
0	1	0	ClockB = Bus clock/4
0	1	1	ClockB = Bus clock/8
1	0	0	ClockB = Bus clock/16
1	0	1	ClockB = Bus clock/32
1	1	0	ClockB = Bus clock/64
1	1	1	ClockB = Bus clock/128

PWMSCLA (PWM Scale A) Register – address $00A8

The location of the bits in the PWMSCLA register is shown.

D7	D6	D5	D4	D3	D2	D1	D0

The 8 bits in this register can be used to subdivide the ClockA according to the expression and the table.

ClockSA = ClockA / (2 x D7…D0 value)

D7	D6	D5	D4	D3	D2	D1	D0	Pre-scale for ClockSA
0	0	0	0	0	0	0	1	Divide by 2
0	0	0	0	0	0	1	0	Divide by 4
0	0	0	0	0	0	1	1	Divide by 6
0	0	0	0	0	1	0	0	Divide by 8
0	0	0	0	0	1	0	1	Divide by 10
0	0	0	0	0	1	1	0	Divide by 12
0	0	0	0	0	1	1	1	Divide by 14
.	.	.	.	.	.	.	.	
1	1	1	1	1	1	0	0	Divide by 504
1	1	1	1	1	1	0	1	Divide by 506
1	1	1	1	1	1	1	0	Divide by 508
1	1	1	1	1	1	1	1	Divide by 510
0	0	0	0	0	0	0	0	Divide by 512

PWMSCLB (PWM Scale B) Register – address $00A9

The location of the bits in this register is shown.

D7	D6	D5	D4	D3	D2	D1	D0

The 8 bits in this register can be used to subdivide the ClockB according to the expression and the table.

ClockSB = ClockB / (2 x D7…D0 value)

D7	D6	D5	D4	D3	D2	D1	D0	Pre-scale for ClockSB
0	0	0	0	0	0	0	1	Divide by 2
0	0	0	0	0	0	1	0	Divide by 4
0	0	0	0	0	0	1	1	Divide by 6
0	0	0	0	0	1	0	0	Divide by 8
0	0	0	0	0	1	0	1	Divide by 10
0	0	0	0	0	1	1	0	Divide by 12
0	0	0	0	0	1	1	1	Divide by 14
.	.	.	.	.	.	.	.	
1	1	1	1	1	1	0	0	Divide by 504
1	1	1	1	1	1	0	1	Divide by 506
1	1	1	1	1	1	1	0	Divide by 508
1	1	1	1	1	1	1	1	Divide by 510
0	0	0	0	0	0	0	0	Divide by 512

PWMCLK (PWM Clock Selection) Register – address $00A2

The location of the 8 bits in this register is shown.

PCLK7	PCLK6	PCLK5	PCLK4	PCLK3	PCLK2	PCLK1	PCLK0

The 8 bits in this register are used to select which Clock is to be used.

PCLK7 – PCLK0 PWM Clock Selection	
PCLK7	1 = ClockSB is the source for PWM channel 7; 0 = ClockB is the source
PCLK6	1 = ClockSB is the source for PWM channel 6; 0 = ClockB is the source
PCLK5	1 = ClockSA is the source for PWM channel 5; 0 = ClockA is the source
PCLK4	1 = ClockSA is the source for PWM channel 4; 0 = ClockA is the source
PCLK3	1 = ClockSB is the source for PWM channel 3; 0 = ClockB is the source
PCLK2	1 = ClockSB is the source for PWM channel 2; 0 = ClockB is the source
PCLK1	1 = ClockSA is the source for PWM channel 1; 0 = ClockA is the source
PCLK0	1 = ClockSA is the source for PWM channel 0; 0 = ClockA is the source

Examples

1. If the Bus frequency is 24 MHz what are the highest and lowest values for CLOCKA ? What are the required values in the PWMPRCLK register?

CLOCKA highest value = 24 MHz /1 = 24 MHz PWMPRCLK register value = $00

CLOCKA lowest value = 24 MHz /128 = 187,500 Hz PWMPRCLK register value = $07

2. If the Bus frequency is 24 MHz and CLOCK A is set to a value of 1.5 MHz, what is the CLOCKA pre-scale value. What are the highest and lowest values for CLOCKSA? What are the required values in the PWMSCLA register?

CLOCKA pre scale value = 24 MHz/1.5 MHz = 16

CLOCKSA highest value = 1.5 MHz /2 = 750 kHz PWMSCLA register value = $01

CLOCKSA lowest value = 1.5 MHz /512 = 2.93 kHz PWMSCLA register value = $00

3. Determine the values needed for registers PWMPRCLK and PWMSCLB to give CLOCKB = 3 MHz and CLOCKSB = 150 kHz?

To get CLOCKB = 3 MHz requires a pre scale factor of 8. PWMPRCLK register value = $30

To get CLOCKSB = 150 kHz requires a pre scale factor of 20. PWMSCLB register value = $0A

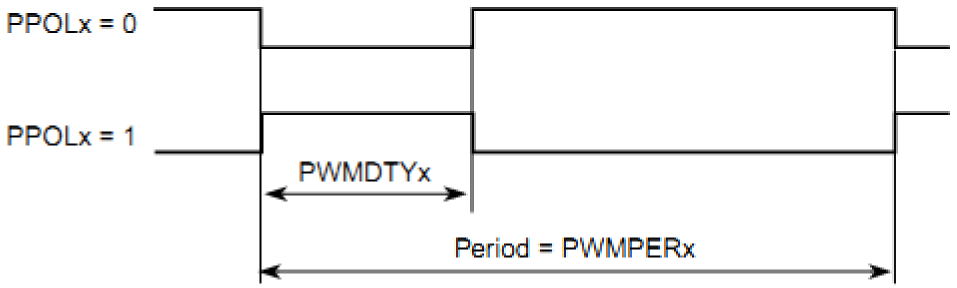
Channel Polarity Selection

PWMPOL (Polarity Selection) Register – address $00A1

The location of the bits in this register is shown.

PPOL7	PPOL6	PPOL5	PPOL4	PPOL3	PPOL2	PPOL1	PPOL0

The 8 bits in this register are used to select the polarity of the output waveform.



Left Aligned Output

PPOL7 – PPOL0 PWM Polarity Selection for Channels 7 - 0	
PPOL7: ch 7	1 = O/P is high at start of period then goes low at end of period; 0 = O/P is low at start …
PPOL6: ch 6	1 = O/P is high at start of period then goes low at end of period; 0 = O/P is low at start …
PPOL5: ch 5	1 = O/P is high at start of period then goes low at end of period; 0 = O/P is low at start …
PPOL4: ch 4	1 = O/P is high at start of period then goes low at end of period; 0 = O/P is low at start …
PPOL3: ch 3	1 = O/P is high at start of period then goes low at end of period; 0 = O/P is low at start …
PPOL2: ch 2	1 = O/P is high at start of period then goes low at end of period; 0 = O/P is low at start …
PPOL1: ch 1	1 = O/P is high at start of period then goes low at end of period; 0 = O/P is low at start …
PPOL0: ch 0	1 = O/P is high at start of period then goes low at end of period; 0 = O/P is low at start …

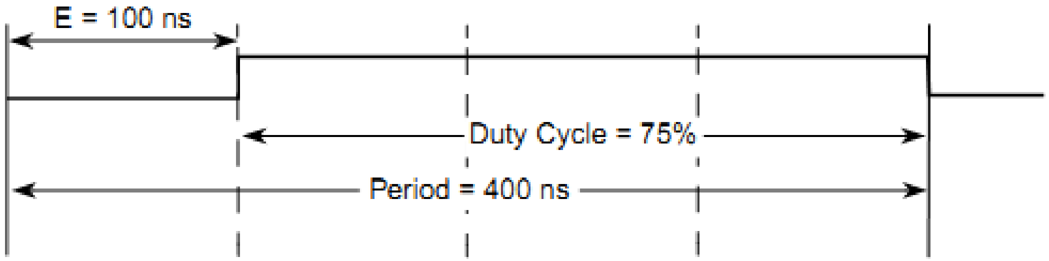
PWMCAE (PWM Center Aligned Enable) Register – address $00A4

This 8 bit register selects either Center or Left Aligned output.

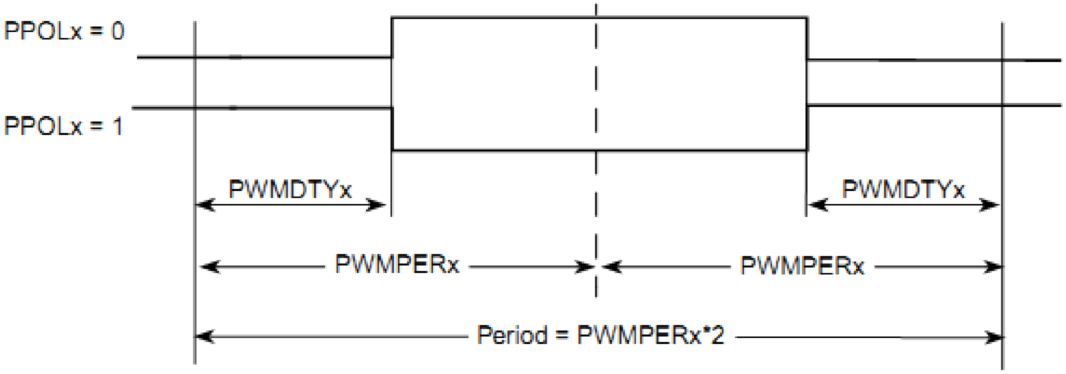
The location of the bits in this register is shown.

CAE7	CAE6	CAE5	CAE4	CAE3	CAE2	CAE1	CAE0

CAE7 – CAE0 PWM Alignment Selection for Channels 7 - 0	
CAE7	1 = Center aligned for channel 7; 0 = Left aligned for channel 7
CAE6	1 = Center aligned for channel 6; 0 = Left aligned for channel 7
CAE5	1 = Center aligned for channel 5; 0 = Left aligned for channel 7
CAE4	1 = Center aligned for channel 4; 0 = Left aligned for channel 7
CAE3	1 = Center aligned for channel 3; 0 = Left aligned for channel 7
CAE2	1 = Center aligned for channel 2; 0 = Left aligned for channel 7
CAE1	1 = Center aligned for channel 1; 0 = Left aligned for channel 7
CAE 0	1 = Center aligned for channel 0; 0 = Left aligned for channel 7



PWM Left Aligned Output Example Waveform



PWM Center Aligned Output Waveform

PWMPERx (PWM Period) Registers – address $00B4 to $00BB

These 8 bit registers set the value of the period for the PWM0 to PWM7 channels.

Example: **PWMPER2 is at address $00B6**

PWMDTYx (PWM Duty Cycle) Registers – address $00BC to $00C3

These 8 bit registers set the value of the duty cycle for the PWM0 to PWM7 channels.

Example: **PWMDTY3 is at address $00BF**

PWMCTL (PWM Control) Register – address $00A5

This register can be used to select 8 or 16 bit PWM channels. **Standard value = $0C (0000 1100)**

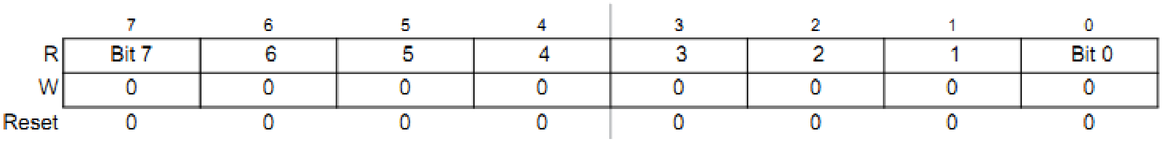
The location of the 8 bits in this register is shown.

<table>
<tr><td>CON67</td><td>CON45</td><td>CON23</td><td>CON01</td><td>PSWA1</td><td>PFRZ</td><td>0</td><td>0</td></tr>
<tr><td>CON67</td><td colspan="7">1 = Ch 6 & 7 joined to form 16 bit PWM channel ; 0 = Ch 6 & 7 are separate PWM channels</td></tr>
<tr><td>CON45</td><td colspan="7">1 = Ch 4 & 5 joined to form 16 bit PWM channel ; 0 = Ch 4 & 5 are separate PWM channels</td></tr>
<tr><td>CON23</td><td colspan="7">1 = Ch 2 & 3 joined to form 16 bit PWM channel ; 0 = Ch 2 & 3 are separate PWM channels</td></tr>
<tr><td>CON01</td><td colspan="7">1 = Ch 0 & 1 joined to form 16 bit PWM channel ; 0 = Ch 0 & 1 are separate PWM channels</td></tr>
<tr><td>PSWA1</td><td colspan="7">1 = Stops the I/P clock to prescaler in Wait mode; 0 = Allows I/P to prescaler to continue</td></tr>
<tr><td>PFRZ</td><td colspan="7">1 = Disable I/P clock to prescaler in Freeze mode; 0 = Allows PWM to continue in Freeze</td></tr>
<tr><td>Bit1</td><td colspan="7">Not used</td></tr>
<tr><td>Bit0</td><td colspan="7">Not used</td></tr>
</table>

PWM Channel Counter Register (PWMCNTx) – address $00AC to $00B3

Example: PWMCNT2 is at address $00AE

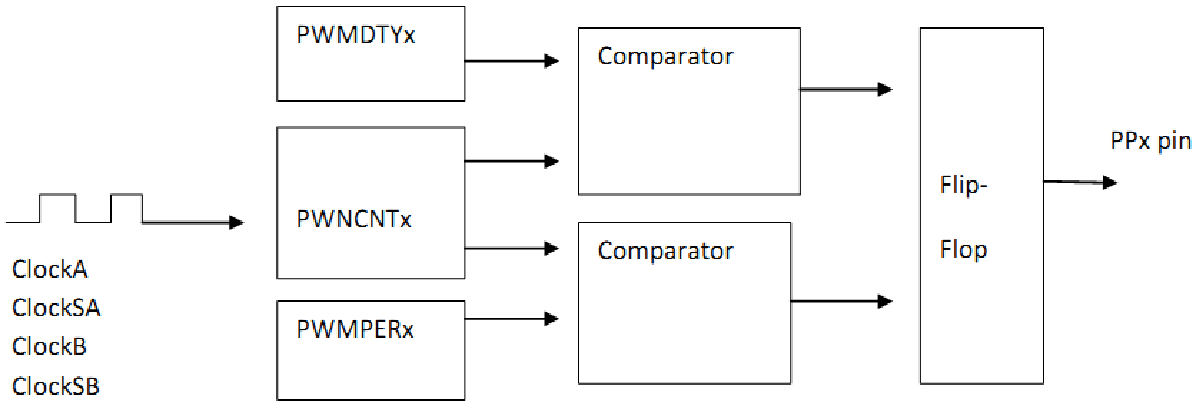
Each channel has a dedicated 8-bit up/down counter which runs at the rate of the selected clock source. The counter can be read at any time without affecting the count or the operation of the PWM channel. In left aligned output mode, the counter counts from 0 to the value in the period register - 1. In center aligned output mode, the counter counts from 0 up to the value in the period register and then back down to 0. Any value written to the counter causes the counter to reset to $00, the counter direction to be set to up, the immediate load of both duty and period registers with values from the buffers, and the output to change according to the polarity bit.



	7	6	5	4	3	2	1	0
R	Bit 7	6	5	4	3	2	1	Bit 0
W	0	0	0	0	0	0	0	0
Reset	0	0	0	0	0	0	0	0

Controlling Period and Duty Cycle

The diagram below shows how the Counter, Period and Duty Cycle registers and other circuitry work together. The value in the PWMPERx register controls the period of the output waveform at the PORTP pin. One of 4 possible clock sources in connected to the Counter PWMCNTx. After the PWM channel is enabled the PWMCNTx register starts to count up from zero. As it counts its value is compared to the period register and when they match the output at the PORTP pin changes state. The value in the duty cycle register controls the duty cycle of the waveform. When the counter reaches the value in the duty cycle register the output at the PORTP pin changes state.



Determining Period and Duty Cycle

Finding the values needed for the PWMPERx and PWMDTYx registers give the desired values for Period and Duty Cycle. These vary according to the Polarity and Alignment options chosen.

PWMPERx Values

Left Aligned Output – CAEx = 0

The value in the period register determines the period and frequency of the output waveform.

PWMPERx = PWM Clock (A, B, SA, or SB) frequency/PWMx Freq

PWMDTYx Values

The duty cycle is set by the value in PWMDTYx. The values will vary based on the setting of the Polarity bit.

Polarity = 0

Duty Cycle = [(PWMPERx-PWMDTYx)/PWMPERx] * 100%

PWMDTYx = (1 - Duty Cycle) x PWMPERx

Polarity = 1

Duty Cycle = [PWMDTYx / PWMPERx] * 100%

PWMDTYx = (Duty Cycle) x PWMPERx

Center Aligned Output – CAEx = 1

The value in the period register determines the period and frequency of the waveform. In this mode The 8-bit counter operates as an up/down counter and adds a factor of 2 to the expression for PWMPERx.

PWMPERx = PWM Clock (A, B, SA, or SB) frequency/2 x PWMx Freq

PWMDTYx Values

The duty cycle is set by the value in PWMDTYx. The values will vary based on the setting of the Polarity bit.

Polarity = 0

Duty Cycle = [(PWMPERx-PWMDTYx)/PWMPERx] * 100%

PWMDTYx = (1 - Duty Cycle) x PWMPERx

Polarity = 1

Duty Cycle = [PWMDTYx / PWMPERx] * 100%

PWMDTYx = (Duty Cycle) x PWMPERx

Examples

1. ClockA = 750 kHz, Polarity = 1, Left Alignment. Determine values of PWMPERx and PWMDTYx to give an output frequency of 5 kHz with an 80% duty cycle.

   PWMPERx = 750 kHz/5 kHz = 150   PWMDTYx = 0.80 x 150 = 120

2. ClockA = 750 kHz, Polarity = 0, Left Alignment. Determine values of PWMPERx and PWMDTYx to give an output frequency of 5 kHz with an 80% duty cycle.

   PWMPERx = 750 kHz/5 kHz = 150   PWMDTYx = (1 - 0.80) x 150 = 30

3. ClockSA = 150 kHz, Polarity = 1, Centre Alignment. Determine values of PWMPERx and PWMDTYx to give an output frequency of 1 kHz with an 20% duty cycle.

   PWMPERx = 150 kHz/2 x 2.5 kHz = 30   PWMDTYx = (0.20) x 30 = 6

4. ClockSB = 150 kHz, Polarity = 0, Centre Alignment. Determine values of PWMPERx and PWMDTYx to give a output frequency of with an 80% duty cycle.

   PWMPERx = 150 kHz/2 x 2.5 kHz = 30   PWMDTYx = (1 - 0.20) x 30 = 24

Sample programs

Examine the following Assembly language program and then answer the questions.

```
;*************************************
;*  PWM Example 1
;*************************************

PWMCLK:    equ    $A2      ;address of PWMCLK register
PWMPRCLK:  equ    $A3      ;address of PWMPRCLK register
PWMPOL:    equ    $A1      ;address of PWMPOL register
PWMCAE:    equ    $A4      ;address of PWMPOL register
PWMCTL:    equ    $A5      ;address of PWMCTL register
PWMPER0:   equ    $B4      ;address of PWMPER0 register
PWMDTY0:   equ    $BC      ;address of PWMDTY0 register
PWMCNT0:   equ    $AC      ;address of PWMCNT0 register
PWME:      equ    $A0      ;address of PWME register


      org $2000
      movb  #0,   PWMCLK      ;select Clock
      movb  #1,   PWMPRCLK    ;select Clock prescale
      movb  #1,   PWMPOL      ;select PWM Polarity
      movb  #0,   PWMCAE      ;select PWM alignment mode
      movb  #$0C, PWMCTL      ;8 bit mode, stop PWM in wait and freeze mode
      movb  #120, PWMPER0     ;set PWM channel period value to X usec
      movb  #30,  PWMDTY0     ;set PWM channel duty cycle value to Y%
      movb  #0,   PWMCNT0     ;reset PWM channel counter
      movb  #$01, PWME        ;enable PWM channel
      swi
      end
```

Questions

1. What PWM channel is being used?
2. What Clock is being used for PWM?
3. What pre scale factor is being used?
4. What is the PWM frequency?
5. What is the value of the Polarity bit?
6. What Alignment type is being used?
7. What are the frequency, period and duty cycle of the output?
8. What are the frequency, period and duty cycle if the Polarity bit is complemented?
9. On what pin on the Demo board header is the signal found?
10. What is the largest value that can be loaded into any PWMPERx register?

C Program Code

Examine the following C language program and then answer the questions.

```
//*************************************
//*
//*  PWM Example 2
//*
//***********************************************

#include <hidef.h>        /* common defines and macros */
#include "derivative.h"   /* derivative-specific definitions */


void main(void) {

  PWMCLK=0x02;     // select clock for PWM channel
  PWMPRCLK=2;      // set clock A pre scale
  PWMSCLA=1;       // set clockSA pre scale
  PWMPOL=0x02;     // PWM channel O/P start with a high
  PWMCAE=0x02;     // PWM channel center aligned
  PWMCTL=0x0C;     // select 8 bit PWM
  PWMPER1=150;     // set PWM channel period to X usec
  PWMDTY1=90;      // set PWM channel Duty Cycle to Y%
  PWMCNT1=0x02;    // reset PWM channel counter
  PWME |= 0x02;    // enable PWM channel

 asm("swi");
}
```

Questions

1. What PWM channel is being used?
2. What Clock is being used for PWM?
3. What prescale factors are being used?
4. What is the PWM frequency?
5. What is the value of the Polarity bit?
6. What Alignment type is being used?
7. What are the frequency period and duty cycle of the output?
8. What are the frequency period and duty cycle if the Polarity bit is complemented?
9. On what pin on the Demo board header is the signal found?

Application Note
Written by David Lloyd
Computer Engineering Program
Humber College