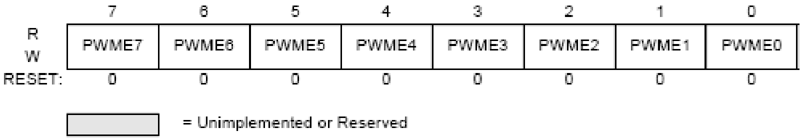
- **The 9S12 Pulse Width Modulation System**
- Huang Sections 8.10 and 8.11
- PWM_8B8C Block User Guide
  - What is Pulse Width Modulation
  - The 9S12 Pulse Width Modulation system
  - Registers used by the PWM system
  - How to set the period for PWM Channel 0
  - How to set the clock for PWM Channel 0
  - Interdependence of clocks for Channels 0 and 1
  - PWM Channels 2 and 3
  - Using the 9S12 PWM
  - A program to use the 9S12 PWM



	7	6	5	4	3	2	1	0
R W	PWME7	PWME6	PWME5	PWME4	PWME3	PWME2	PWME1	PWME0
RESET:	0	0	0	0	0	0	0	0

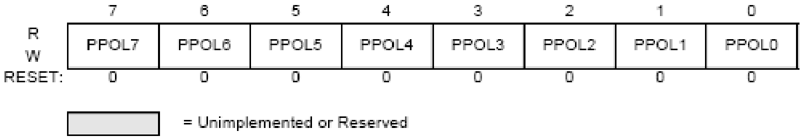
▭ = Unimplemented or Reserved

Figure 3-1 PWM Enable Register (PWME)

Read: anytime

Write: anytime

Set **PWMEn** = **1 to enable** PWM on Channel n.
If **PWMEn** = **0**, Port P bit n can be used for **general purpose I/O.**



	7	6	5	4	3	2	1	0
R W	PPOL7	PPOL6	PPOL5	PPOL4	PPOL3	PPOL2	PPOL1	PPOL0
RESET:	0	0	0	0	0	0	0	0

▭ = Unimplemented or Reserved

Figure 3-2 PWM Polarity Register (PWMPOL)

Read: anytime

Write: anytime

PPOLn − **Choose polarity 1** => high polarity 0 => low polarity
We will use high polarity only. PWMPOL = 0xFF;
With high polarity, duty cycle is amount of time output is high

	7	6	5	4	3	2	1	0
R W	PCLK7	PCLKL6	PCLK5	PCLK4	PCLK3	PCLK2	PCLK1	PCLK0
RESET:	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

Figure 3-3 PWM Clock Select Register (PWMCLK)

Read: anytime

Write: anytime

PCLKn – Choose clock source for Channel n
CH5, CH4, CH1, CH0 can use either **A** (0) or **SA** (1)
CH7, CH6, CH3, CH2 can use either **B** (0) or **SB** (1)

$$\mathbf{SA}=\mathbf{A} / (2 \times \mathrm{PWMSCLA}) \qquad \mathbf{SB}=\mathbf{B} / (2 \times \mathrm{PWMSCLB})$$

	7	6	5	4	3	2	1	0
R	0	PCKB2	PCKB1	PCKB0	0	PCKA2	PCKA1	PCKA0
W	▨				▨			
RESET:	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

Figure 3-4 PWM Prescale Clock Select Register (PWMPRCLK)

Read: anytime

Write: anytime

This register selects the **prescale clock source** for clocks **A** and **B** independently

PCKA[2–0] – Prescaler for Clock **A** $\mathbf{A} = 24\ \mathrm{MHz} / 2^{(\mathbf{PCKA[2-0]})}$
PCKB[2–0] – Prescaler for Clock **B** $\mathbf{B} = 24\ \mathrm{MHz} / 2^{(\mathbf{PCKB[2-0]})}$

	7	6	5	4	3	2	1	0
R W	CAE7	CAE6	CAE5	CAE4	CAE3	CAE2	CAE1	CAE0
RESET:	0	0	0	0	0	0	0	0

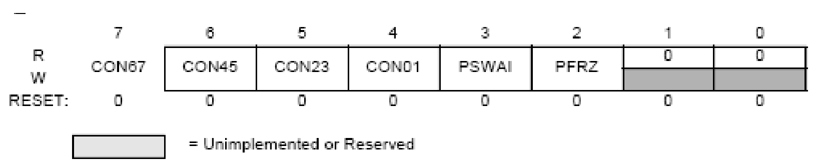
▨ = Unimplemented or Reserved

Figure 3-5 PWM Center Align Enable Register (PWMCAE)

Read: anytime

Write: anytime

Select **center aligned** outputs (1) or **left aligned** outputs (0)
Choose PWMCAE = 0x00 to choose left aligned mode



	7	6	5	4	3	2	1	0
R	CON67	CON45	CON23	CON01	PSWAI	PFRZ	0	0
W							▨	▨
RESET:	0	0	0	0	0	0	0	0

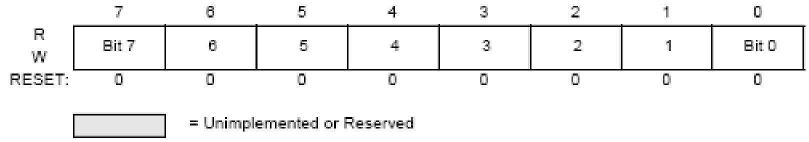
▨ = Unimplemented or Reserved

Figure 3-6 PWM Control Register (PWMCTL)

Read: anytime

Write: anytime

CONxy − Concatenate **PWMx** and **PWMy** into one **16 bit PWM.**
Choose PWMCTL = 0x00 to choose 8−bit mode



	7	6	5	4	3	2	1	0
R	Bit 7	6	5	4	3	2	1	Bit 0
W								
RESET:	0	0	0	0	0	0	0	0

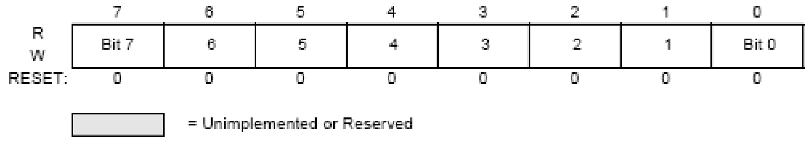
▨ = Unimplemented or Reserved

Figure 3-9 PWM Scale A Register (PWMSCLA)

Read: anytime

Write: anytime (causes the scale counter to load the PWMSCLA value)

PWMSCLA adjusts frequency of **Clock SA**



	7	6	5	4	3	2	1	0
R	Bit 7	6	5	4	3	2	1	Bit 0
W								
RESET:	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

Figure 3-10 PWM Scale B Register (PWMSCLB)

Read: anytime

Write: anytime (causes the scale counter to load the PWMSCLB value).

PWMSCLB adjusts frequency of **Clock SB**

PWMPERx sets the period of **Channel n**

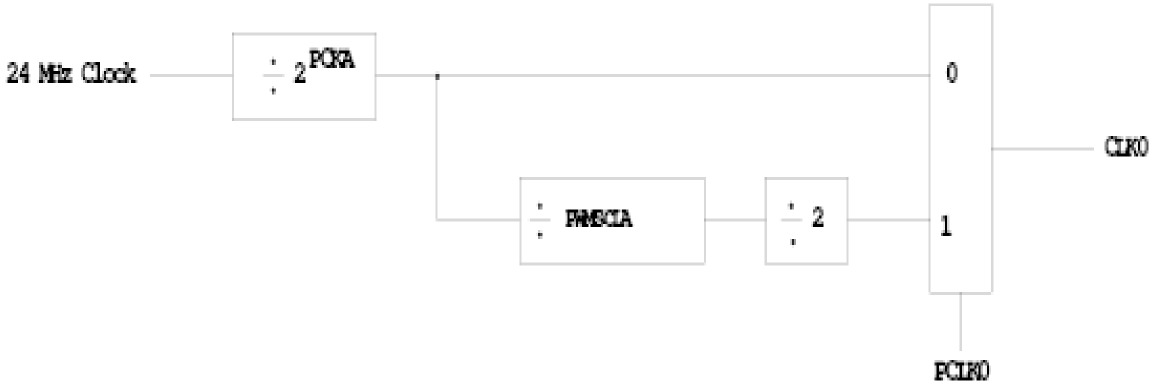
PWM Period = PWMPERn x Period of PWM Clock n

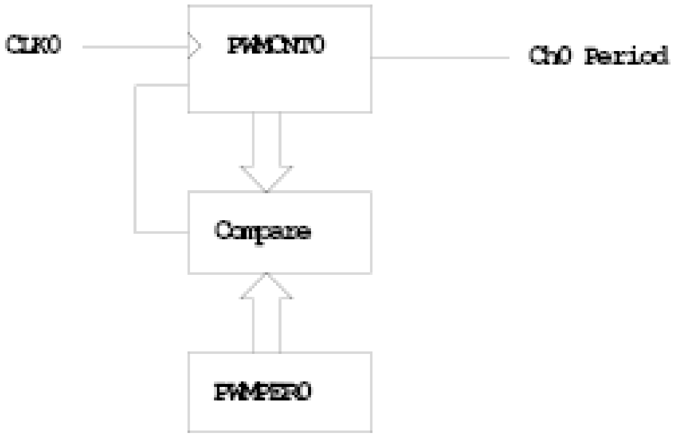
PWMDTYx sets the duty cycle of **Channel n**

PWM Duty Cycle = PWMDTYn / Period x 100%

How to set the clock for PWM Channel 0

You need to set PCKA, PWSCLA, PCLK0, and PWPER0





PWMCNT0 counts from **0** to **PWMPER0 – 1**
It takes PWMPER0 periods of CLK0 to make one Ch0 period

Ch0 Period = PWMPER0 x CLK0 Period

$$= \begin{cases} \text{PWMPER0 x } (2^{\text{PCKA}}) & \textbf{(PCLK0 = 0)} \\ \text{PWMPER0 x } (2^{\text{PCKA+1}}) \text{ x PWMSCLA } & \textbf{(PCLK0 = 1)} \end{cases}$$

How to set the Period for PWM Channel 0

- To set the period for PWM Channel 0:

– Set the PWM Period register for **Channel 0**, **PWMPER0**
– CLK0, the clock for Channel 0, drives a counter (PWCNT0)
– **PWCNT0** counts from **0** to **PWMPER0 - 1**
– The period for PWM Channel 0 is PWMPER0 × Period of CLK0

- There are two modes for the clock for PWM Channel 0
  - You select the mode by the PCLK0 bit
  - If PCLK0 == 0, CLK0 is generated by dividing the 24 MHz clock by 2^{PCKA}, where PCKA is between 0 and 7
  - If PCLK0 == 1, CLK0 is generated by dividing the 24 MHz clock by $2^{PCKA+1} \times$ PWSCLA, where PCKA is between 0 and 7 and PWSCALA is between 0 and 255 (a value of 0 gives a divider of 256)

- The Period for PWM Channel 0 (in number of 41.67 ns cycles) is calculated by

$$\text{Period} \quad = \begin{cases} \text{PWMPER0} \times 2^{\text{PCKA}} & \textbf{if PCLK0 == 0} \\ \text{PWMPER0} \times 2^{\text{PCKA+1}} \times \text{PWMSCLA} & \textbf{if PCLK0 == 1} \end{cases}$$

- With PCLK0 == 0, the maximum possible PWM period is 1.36 ms

- With PCLK0 == 1, the maximum possible PWM period is 0.695 s

- To get a 0.5 ms PWM period, you need 12,000 cycles of the 24 MHz clock.

$$12,000 \quad = \begin{cases} \text{PWMPER0} \times 2^{\text{PCKA}} & \textbf{if PCLK0 == 0} \\ \text{PWMPER0} \times 2^{\text{PCKA+1}} \times \text{PWMSCLA} & \textbf{if PCLK0 == 1} \end{cases}$$

- You can do this in many ways

  - With PCLK0 = 0, can have

PCKA	**PWMPER0**	**Precision**
6	187	Approx.
7	94	Approx.

  - With PCLK0 = 1, can have

PCKA	**PWMSCLA**	**PWMPER0**	**Precision**
0	24	250	Exact
0	25	240	Exact
0	30	200	Exact
1	12	250	Exact
1	15	200	Exact
2	6	250	Exact
2	10	150	Exact

and many other combinations

• You want PWMPER0 to be large (say, 100 or larger)

– If PWMPER0 is small, you don't have much control over the duty cycle
– For example, if PWMPER0 = 4, you can only have 0%, 25%, 50%, 75% or 100% duty cycles

• Once you choose a way to set the PWM period, you can program the PWM registers

• For example, to get a 0.5 ms period, let's use PCLK0 = 1, PCKA = 0, PWMSCLA = 30, and PWMPER0 = 200

• We need to do the following:
– Write 0x00 to PWMCTL (to set up 8-bit mode)
– Write 0xFF to PWMPOL (to select high polarity mode)
– Write 0x00 to PWMCAE (to select left aligned mode)
– Write 0 to Bits 2,1,0 of PWMPRCLK (to set PCKA to 0)
– Write 1 to Bit 0 of PWMCLK (to set PCLK0 = 1)
– Write 30 to PWMSCLA
– Write 200 to PWMPER0
– Write 1 to Bit 0 of PWME (to enable PWM on Channel 0)
– Write the appropriate value to PWDTY0 to get the desired duty cycle (e.g., PWDTY0 = 120 will give 60% duty cycle)

C code to set up PWM Channel 0 for 0.5 ms period (2 kHz frequency) PWM with 60% duty cycle

```
PWMCTL = 0x00;                           /* 8-bit Mode */
PWMPOL = 0xFF;                           /* High polarity mode */
PWMCAE = 0x00;                           /* Left-Aligned */
PWMPRCLK = PWMPRCLK & ~0x07;             /* PCKA = 0 */
PWMCLK = PWMCLK | 0x01;                  /* PCLK0 = 1 */
PWMSCLA = 30;
PWMPER0 = 200;
PWME = PWME | 0x01;                      /* Enable PWM Channel 0 */
PWDTY0 = 120;                            /* 60% duty cycle on Channel 0 */
```

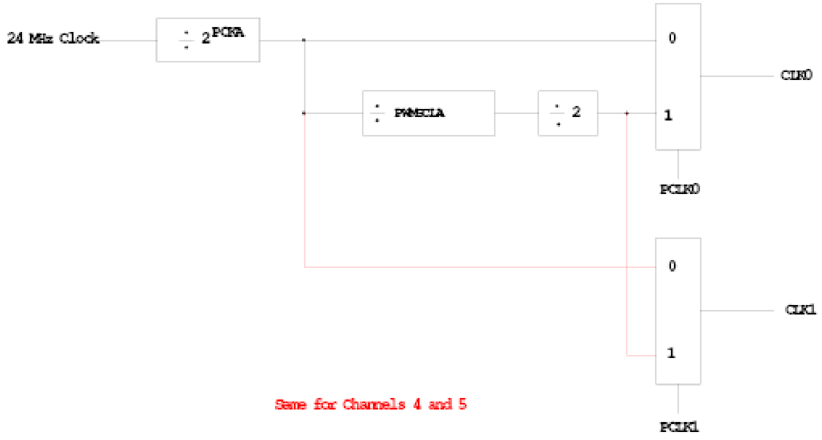
Interdependence of clocks for Channels 0, 1, 4 and 5

- The clocks for Channels 0, 1, 4 and 5 are interdependent

- They all use PCKA and PWMSCLA

- To set the clock for Channel n, you need to set PCKA, PCLKn, PWMSCLA (if PCLKn == 1) and PWMPERn where n = 0, 1, 4 or 5

Clock Select for PWM Channels 0 and 1



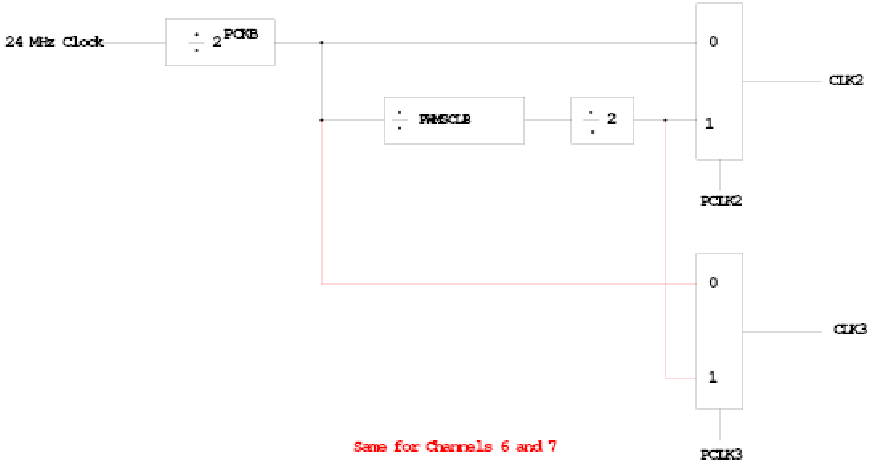
Same for Channels 4 and 5

PWM Channels 2, 3, 6 and 7

- PWM channels 2, 3, 6 and 7 are similar to PWM channels 0, 1, 4 and 5

- To set the clock for Channel n, you need to set PCKB, PCLKn, PWMSCLB (if PCLKn == 1) and PWMPERn where n = 2, 3, 6 or 7

Clock Select for PWM Channels 2 and 3



Using the HCS12 PWM

1. Choose 8-bit mode (PWMCTL = 0x00)

2. Choose high polarity (PWMPOL = 0xFF)

3. Choose left-aligned (PWMCAE = 0x00)

4. Select clock mode in PWMCLK:
   - **PCLKn = 0 for 2^N**,
   - **PCLKn = 1 for $2^{(N+1)}$ ×M**,

5. Select N in PWMPRCLK register:
   - **PCKA** for channels **5, 4, 1, 0**;
   - **PCKB** for channels **7, 6, 3, 2**.

6. If PCLKn = 1, select M
   - PWMSCLA = M for channels 5, 4, 1, 0
   - PWMSCLB = M for channels 7, 6, 3, 2.

7. Select PWMPERn, normally between 100 and 255.

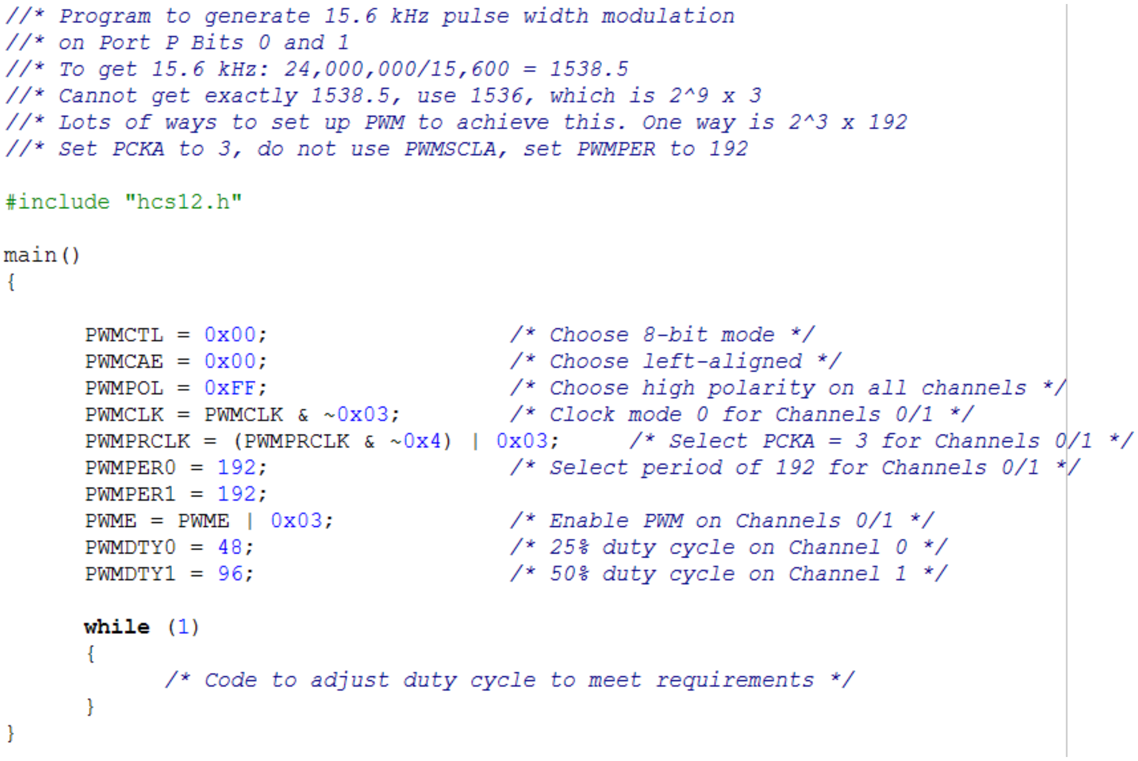
8. Enable desired PWM channels: PWME.

9. Select PWMDTYn, normally between 0 and PWMPERn. Then

   Duty Cycle n = (PWMDTYn / PWMPERn) × 100%
   Change duty cycle to control speed of motor or intensity of light, etc.

10. For 0% duty cycle, choose PWMDTYn = 0x00.

Program to use the HCS12 PWM System



```
//* Program to generate 15.6 kHz pulse width modulation
//* on Port P Bits 0 and 1
//* To get 15.6 kHz: 24,000,000/15,600 = 1538.5
//* Cannot get exactly 1538.5, use 1536, which is 2^9 x 3
//* Lots of ways to set up PWM to achieve this. One way is 2^3 x 192
//* Set PCKA to 3, do not use PWMSCLA, set PWMPER to 192

#include "hcs12.h"

main()
{

      PWMCTL = 0x00;                  /* Choose 8-bit mode */
      PWMCAE = 0x00;                  /* Choose left-aligned */
      PWMPOL = 0xFF;                  /* Choose high polarity on all channels */
      PWMCLK = PWMCLK & ~0x03;        /* Clock mode 0 for Channels 0/1 */
      PWMPRCLK = (PWMPRCLK & ~0x4) | 0x03;     /* Select PCKA = 3 for Channels 0/1 */
      PWMPER0 = 192;                  /* Select period of 192 for Channels 0/1 */
      PWMPER1 = 192;
      PWME = PWME | 0x03;             /* Enable PWM on Channels 0/1 */
      PWMDTY0 = 48;                   /* 25% duty cycle on Channel 0 */
      PWMDTY1 = 96;                   /* 50% duty cycle on Channel 1 */

      while (1)
      {
            /* Code to adjust duty cycle to meet requirements */
      }
}
```