- **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

### Registers

<table>
<thead>
<tr>
<th>PWME7</th>
<th>PWME6</th>
<th>PWME5</th>
<th>PWME4</th>
<th>PWME3</th>
<th>PWME2</th>
<th>PWME1</th>
<th>PWME0</th>
<th>0x00A0 PWME</th>
</tr>
</thead>
</table>

Set PWME\(n\) = 1 to enable PWM on Channel \(n\)
If PWME\(n\) = 0, Port P bit \(n\) can be used for general purpose I/O

<table>
<thead>
<tr>
<th>PPOL7</th>
<th>PPOL6</th>
<th>PPOL5</th>
<th>PPOL4</th>
<th>PPOL3</th>
<th>PPOL2</th>
<th>PPOL1</th>
<th>PPOL0</th>
<th>0x00A1 PWMPOL</th>
</tr>
</thead>
</table>

PPOL\(n\) – 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

<table>
<thead>
<tr>
<th>PCLK7</th>
<th>PCLK6</th>
<th>PCLK5</th>
<th>PCLK4</th>
<th>PCLK3</th>
<th>PCLK2</th>
<th>PCLK1</th>
<th>PCLK0</th>
<th>0x00A2 PWMCLK</th>
</tr>
</thead>
</table>

PCLK\(n\) – 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)

\[
SA = A/(2 \times PWMSCLA) \quad SB = B/(2 \times PWMSCLB)
\]

<table>
<thead>
<tr>
<th>0</th>
<th>PCKB2</th>
<th>PCKB1</th>
<th>PCKB0</th>
<th>0</th>
<th>PCKA2</th>
<th>PCKA1</th>
<th>PCKA0</th>
<th>0x00A3 PWMPRCLK</th>
</tr>
</thead>
</table>

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

PCKA\(2–0\) – Prescaler for Clock A \(A = 24 \text{ MHz} / 2^{(\text{PCKA}[2–0])}\)
PCKB\(2–0\) – Prescaler for Clock B \(B = 24 \text{ MHz} / 2^{(\text{PCKB}[2–0])}\)
CAE7  CAE6  CAE5  CAE4  CAE3  CAE2  CAE1  CAE0  0x00A4 PWMCAE

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

CON67  CON45  CON23  CON01  PSWAI  PFRZ  0  0  0x00A5 PWMCTL

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

BIT 7  BIT 6  BIT 5  BIT 4  BIT 3  BIT 2  BIT 1  BIT 0  0x00A8 PWMSCLA

PWMSCLA adjusts frequency of Clock SA

BIT 7  BIT 6  BIT 5  BIT 4  BIT 3  BIT 2  BIT 1  BIT 0  0x0098 PWMSCLB

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, PWSCALA, PCLK0, and PWPER0

[Diagram of clock settings]
PWMCNT0 counts from 0 to PWMPER0 − 1
It takes PWMPER0 periods of CLK0 to make one Ch0 period

Ch0 Period = PWMPER0 x CLK0 Period

\[
\begin{align*}
\text{if } PCLK0 = 0 & : PWMPER0 \times (2^{pcka}) \\
\text{if } PCLK0 = 1 & : PWMPER0 \times (2^{pcka+1}) \times PWMSCLA
\end{align*}
\]

**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 (PWMCNT0)
  - PWMCNT0 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
    \(2^{pcka+1} \times PWSCALA\), 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
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.

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 cycle

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

```c
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

Clock Select for PWM Channels 2 and 3

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

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:
• $PCLK_n = 0$ for $2^N$,
• $PCLK_n = 1$ for $2^{(N+1)} \times M$,

5. Select $N$ in PWMPRCLK register:
   • PCKA for channels 5, 4, 1, 0;
   • PCKB for channels 7, 6, 3, 2.
6. If $PCLK_n = 1$, 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
   \[
   \text{Duty Cycle } n = \left( \frac{\text{PWMDTY}n}{\text{PWMPER}n} \right) \times 100\%
   \]
   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 $2^9 \times 3$
* Lots of ways to set up PWM to achieve this. One way is $2^3 \times 192$
* Set PCKA to 3, do not use PWMSCLA, set PWMPER to 192
*
*/

#include "hcs12.h"

main()
{
    /* Choose 8-bit mode */
    PWMCTL = 0x00;

    /* Choose left-aligned */
    PWMCAE = 0x00;

    /* Choose high polarity on all channels */
    PWMPOL = 0xFF;

    /* Select clock mode 0 for Channels 1 and 0 (no PWMSCLA) */
    PWMCLK = PWMCLK & ~0x03;
/* Select PCKA = 3 for Channels 1 and 0 */
PWMPRCLK = (PWMPRCLK & ~0x4) | 0x03;

/* Select period of 192 for Channels 1 and 0 */
PWMPER0 = 192;
PWMPER1 = 192;

/* Enable PWM on Channels 1 and 0 */
PWME = PWME | 0x03;

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 */
}
}