Programmatic Motor Control Through PWM Registers

With 8 PWM channels, the MCU can control up to 8 motors (4 wheels, 1 drive and 1 steer motor per wheel). The motors are numbered 0 to 7 and will be referred to with variable \( x \). The motor operates on a time line that begins when power is on and ends when power is off. The time line is divided into periods of programmable length (described later). Each period is divided into a programmable number of sub-periods (\( \text{PWM PER}_x \)).

A motor runs for as long as it receives high electric signals from the MCU. A period is divided into two contiguous blocks of sub-periods: low and high. If the motor is running for the entirety of period (maximum speed), the low block covers zero sub-periods; the signal is high over all sub-periods of the period. If the motor is inactive (not moving), the high block covers zero sub-periods; the signal is low over all sub-periods of the period. In the context of the P08201 project, the low block is before the high block in any given period (\( \text{PPOL}_x \)). These conditions factor into the case when the motor is running at an intermediate speed. The signal will be low for some quantity of sub-periods (\( \text{PWM DTY}_x \)) and then high for remaining sub-periods until the end of the period.

A PWM clock ticks at a fraction of the bus clock (2 Mhz says Adam?). At each tick, a counter is incremented. At the beginning of a period, the counter starts at zero. When it reaches the duty value (\( \text{PWM DTY}_x \)), the output signal will be flipped from low to high. At the end of the period, the counter is reset for the next period. Thus, a sub-period represents a single clock tick on the time line. There are 4 clocks: A, B, SA, and SB. The A/SA clocks affect 4 specific channels; the B/SB clocks affect the other 4 channels (see section 4 in reference). A and B represent the fraction to apply to the bus clock (prescaled clock \( \text{PWM PRCLK} \)). As the registers are only a byte long, the tick can be lengthened (a fraction of the prescaled clock) by setting the clocks SA or SB (scaled A or B with \( \text{PWM SCLA} \) or \( \text{PWM SCLB} \)). The P08201 uses the SA/SB clocks (\( \text{PWM CLK} \)).

PWM Register Reference

Other registers used in the P08201 source such as DDRA and PORTA are described in the document linked below:

Multiplexed External Bus Interface

Loading New Code

Starting a new project in CodeWarrior

When starting a new project use the 'HC(S)12 New Project Wizard' to set up the base for your new project.

The first selection to make is which microprocessor we are using. Ours is the 'MC9S12DT256', so use the 'MC9S12DT256B' option.
Next are the languages you plan to program in, Assembly, C, and/or C++. Our microcontroller program will be written in C.

**Processor Expert** - The next option is for creating more initial code in the project, select 'No'.

**PC-lint** - This is for using a 3rd party software to monitor your code and find problem areas, select 'No'.

**Startup Code Level** - Selecting the code base for the project. Select 'ANSI startup code'.

**Floating Point Support** - This determines how to store floating point numbers on the microcontroller and which significant digit they go to. The three options are 'None', 'float is IEEE32, double is IEEE32', and 'float is IEEE32, double is IEEE64'. The first option gives no floating point support, in the second both floats and doubles use 32bits, and in the third floats use 32bits and doubles use 64bits. If your program is to use floating point numbers select either 2 or 3, depending on how much extra memory is available.

**Memory Model** - These are different memory usage models. Select 'Banked'

**Connections** - These are the build targets that will be available to your project. In order to put the new code onto the microcontroller you need to check the box next to 'P&E Multilink/Cyclone Pro'.

Click 'Finish' and your new project will be created.

## Transferring Code onto the Microcontroller

In order to flash the microcontroller with new code we need to connect the PC to the project board via the USB port.

The PC will attempt to install some drivers for the new USB port connection, select 'Install Automatically' and allow that to finish.

At this point CodeWarrior should be able to flash the microcontroller.

Open your project in the CodeWarrior IDE and select 'P&E Multilink CyclonePro' from the list of build targets.

Click 'Debug'. This should flash the chip with the code from your project.