TIP #4 Drive Software

Pulse-Width Modulation (PWM) Algorithms

Pulse-Width Modulation is critical to modern digital motor controls. By adjusting the pulse width, the speed of a motor can be efficiently controlled without larger linear power stages. Some PICmicro devices have hardware PWM modules on them. These modules are built into the Capture/Compare/PWM (CCP) peripheral. CCP peripherals are intended for a single PWM output, while the Enhanced CCP (ECCP) is designed to produce the complete H-Bridge output for bidirectional Brushed DC motor control. If cost is a critical design point, a PICmicro device with a CCP module may not be available, so software generated PWM is a good alternative.

The following algorithms are designed to efficiently produce an 8-bit PWM output on the mid-range family of PICmicro microcontrollers. These algorithms are implemented as macros. If you want these macros to be a subroutine in your program, simply remove the macro statements and replace them with a label and a return statement.

EXAMPLE 4-1: 1 OUTPUT 8-BIT PWM

```
pwm_counter equ xxx ;variable
pwm         equ xxx ;variable

set_pwm macro A     ;sets the pwm
                    ;setpoint to the
                    ;value A
  MOVLW A
  MOVWF pwm
  endm

update_PWM macro    ;performs one update
                    ;of the PWM signal
                    ;place the PWM output
                    ;pin at bit 0 or 7 of
                    ;the port
  MOVF pwm_counter,w
  SUBWF pwm, w      ;if the output
                    ;is on bit 0
  RLF       PORTC,f ;replace PORTC with
                    ;the correct port if
                    ;the output is on bit
                    ;7 of the port
                    ;replace the rlf with
                    ;rrf incf
                    ;pwm_counter,f
```

EXAMPLE 4-2: 8 OUTPUT 8-BIT PWM

```
pwm_counter equ xxx     ;variable
pwm0        equ xxx     ;
pwm1        equ pwm0+1
pwm2        equ pwm1+1
pwm3        equ pwm2+1
pwm4        equ pwm3+1
pwm5        equ pwm4+1
pwm6        equ pwm5+1
pwm7        equ pwm6+1
output      equ pwm7+1
set_pwm macro A,b;sets pwm b with
             ;the value A
  MOVLW pwm0
  ADDLW b
  MOVWF fsr
  MOVLW a
  MOVWF indf
  endm

update_PWM macro        ;peforms one
                        ;update of all 8
                        ;PWM siganls
                        ;all PWM signals
                        ;must be on the
                        ;same port
  MOVF      pwm_counter,w
  SUBWF     pwm0,w
  RLF       output,f
  MOVF      pwm_counter,w
  SUBWF     pwm1,w
  RLF       output,f
  MOVF      pwm_counter,w
  SUBWF     pwm2,w
  RLF       output,f
```

EXAMPLE 4-3: 8 OUTPUT 8-BIT PWM (CONTINUED)

```
  MOVF      pwm_counter,w
  SUBWF     pwm3,w
  RLF       output,f
  MOVF      pwm_counter,w
  SUBWF     pwm4,w
  RLF       output,f
  MOVF      pwm_counter,w
  SUBWF     pwm5,w
  RLF       output,f
  MOVF      pwm_counter,w
  SUBWF     pwm6,w
  RLF       output,f
  MOVF      pwm_counter,w
  SUBWF     pwm7,w
  RLF       output,w
  MOVWF     PORTC
  INCF      pwm_counter,f
endm
```

TIP #5 Writing a PWM Value to the CCP Registers With a Mid-range PICmicro® Microcontroller

The two PWM LSb's are located in the CCPCON register of the CCP. This can make changing the PWM period frustrating for a developer.
Example 5-1 through Example 5-3 show three macros written for the mid-range product family that can be used to set the PWM period. The first macro takes a 16-bit value and uses the 10 MSb's to set the PWM period. The second macro takes a 16-bit value and uses the 10 LSb's to set the PWM period. The last macro takes 8 bits and sets the PWM period. This assumes that the CCP is configured for no more than 8 bits.

EXAMPLE 5-1: LEFT JUSTIFIED 16-BIT MACRO

```
pwm_tmp   equ xxx ;this variable must be
                  ;allocated someplace
setPeriod macro a ;a is 2 SFR's in
                  ;Low:High arrangement
                  ;the 10 MSb's are the
                  ;desired PWM value
 RRF      a,w     ;This macro will
                  ;change w
 MOVWF    pwm_tmp
 RRF      pwm_tmp,w
 ANDLW    0x30
 IORLW    0x0F
 MOVWF    CCP1CON
 MOVF     a+1,w
 MOVWF    CCPR1L
```

EXAMPLE 5-2: RIGHT JUSTIFIED 16-BIT MACRO

```
pwm_tmp   equ xxx ;this variable must be
                  ;allocated someplace
setPeriod macro a ;a is 2 bytes in
                  ;Low:High arrangement
                  ;the 10 LSb's are the
                  ;desired PWM value
 SWAPF    a,w     ;This macro will
                  ;change w
 ANDLW    0x30
 IORLW    0x0F
 MOVWF    CCP1CON
 RLF      a,w
 IORLW    0x0F
 MOVWF    pwm_tmp
 RRF      pwm_tmp,f
 RRF      pwm_tmp,w
 MOVWF    CCPR1L
```

EXAMPLE 5-3: 8-BIT MACRO

```
pwm_tmp   equ xxx ;this variable must be
                  ;allocated someplace
setPeriod macro a ;a is 1 SFR
 SWAPF    a,w     ;This macro will
                  ;change w
 ANDLW    0x30
 IORLW    0x0F
 MOVWF    CCP1CON
 RRF      a,w
 MOVWF    pwm_tmp
 RRF      pwm_tmp,w
 MOVWF    CCPR1L
```