Order this document
by EB192/D

Motorola Semiconductor Engineering Bulletin

EB192

A Quick PWM Tutorial for MC68HC11 K, KA, KW, P, and PH Series Microcontrollers

By John Bodnar
Austin, Texas

Introduction

New additions to the family of peripherals available on MC68HC11 microcontrollers are pulse width modulation (PWM) timer channels that may be used to accurately generate square wave signals of varying periods and duty cycles.

The MC68HC11 microcontrollers listed here are equipped with a common set of PWM timer channels that are described in this bulletin.

MC68HC11 K Series	MC68HC11 KA Series	MC68HC11 KW Series	MC68HC11 P Series	MC68HC11 PH Series
MC68HC11K0	MC68HC11KA0	MC68HC11KW1	MC68HC11P2	MC68HC11PH8
MC68HC11K1	MC68HC11KA1	—	MC68HC711P2	MC68HC711PH8
MC68HC11K3	MC68HC11KA2	—	—	—
MC68HC11K4	MC68HC711KA2	—	—	—
MC68HC711K4	MC68HC11KA3	—	—	—
—	MC68HC11KA4	—	—	—
—	MC68HC11KA4	—	—	—

Each of these devices has four 8-bit pulse width modulation (PWM) timer channels. Each pair of PWM timer channels may use one of two different frequency references derived from the E clock. Channels 1 and 2 may use the clock A reference which divides the E clock by 1, 2, 4, or 8. Channels 3 and 4 may use the clock B reference which divides the E clock by 1, 2, 4, 8, 16, 32, 64, or 128. All four channels may use the clock S reference which divides the clock A reference by twice a value from 1 to 256.

To produce PWM waveforms with longer periods, channels 1 and 2 may be concatenated to form a single 16-bit channel using clock A or clock S as the frequency reference. Channels 3 and 4 may be concatenated in the same fashion to form a single 16- bit channel using clock B or clock S as the frequency reference.

In this bulletin, the 8-bit PWM generation is covered first, and then the concepts presented are used to introduce 16-bit PWM generation. The section describing 16-bit operation also includes an example of simultaneous 8- and 16-bit operation which demonstrates the flexibility of the PWM timer channels.

8-Bit PWM Generation

Practical Limits and Frequency Calculations

As mentioned above, each 8-bit PWM channel can use one of a pair of frequency references. The frequency ranges of each 8-bit PWM channel now can be derived.

NOTE: *Lower frequencies can be produced with greater accuracy than higher frequencies.*

E Clock Divisor	Clock A — Reference channels 1 & 2 minimum frequency	Clock A — Reference channels 1 & 2 maximum frequency	Clock B — Reference channels 3 & 4, minimum frequency	Clock B — Reference channels 3 & 4, maximum frequency	Clock S — Reference all channels minimum frequency	Clock S — Reference all channels maximum frequency
1	E/(1*255)	E/(1*2)	E/(1*255)	E/(1*2)	E/(1*2*256*255)	E/(1*2*2)
2	E/(2*255)	E/(2*2)	E/(2*255)	E/(2*2)	E/(2*2*256*255)	E/(2*2*2)
4	E/(4*255)	E/(4*2)	E/(4*255)	E/(4*2)	E/(4*2*256*255)	E/(4*2*2)
8	E/(8*255)	E/(8*2)	E/(8*255)	E/(8*2)	E/(8*2*256*255)	E/(8*2*2)
16	Not available	Not available	E/(16*255)	E/(16*2)	Not available	Not available
32	Not available	Not available	E/(32*255)	E/(32*2)	Not available	Not available
64	Not available	Not available	E/(64*255	E/(64*2)	Not available	Not available
128	Not available	Not available	E/(128*255)	E/(128*2)	Not available	Not available

From this table, we can see that with a 4-MHz E clock, the maximum 8-bit PWM frequency would be 2 MHz and the minimum 8-bit PWM frequency would be 3.83 Hz.

Setting up the PWM timer channels for 8-bit operation is very simple. First, determine the frequency of the signal(s) you wish to generate. Higher frequency signals can be generated with duty cycles of limited accuracy. For instance, a 2-MHz signal may only be generated with a duty cycle of 50%, whereas a 10-kHz signal can be generated with a maximum useful duty cycle of 99.5% (100% duty cycles are possible).

The following equation is used to determine the value placed in one of the four PWM period registers (PWPER1–PWPER4).

$$PWPER = E\ clock \div (desired\ frequency \times prescaler)$$

This equation can also be used to determine the degree of accuracy with which your frequency can be generated. Choose a prescaler from the above table and solve the equation for PWPER. If you choose to use clock S as your frequency reference, the prescaler will take the form of the E clock divisor (for example, 1, 2, 4, 8...128) * 2 * the S prescaler (a value from 1 to 256). If PWPER is greater than 255, choose a larger prescaler. If PWPER is less than 2, choose a smaller prescaler. If PWPER is an integer, your frequency can be produced with no error. If

PWPER is not an integer, there will be some degree of error in the frequency generated. Larger values of PWPER will allow your frequency to be produced with a more accurate duty cycle.

The following examples illustrate the use of this equation.

Example 1

Desired frequency = 10 kHz

We will first try to solve the PWPER equation using an E clock prescaler of 1.

As noted above, a value of PWPER greater than 255 means a larger prescaler should be chosen. Next we will try an E clock prescaler of 2.

This is a legal value for PWPER. From inspection, it should be obvious that we could also choose E clock prescalers of 4, 8, and 16 without introducing any error into the generated signal. Choosing an E clock prescaler of 32 would give PWPER a value of 12.5. If we round this to 12 or 13 and solve the PWPER equation for desired frequency, we get values of 10.417 kHz and 9.615 kHz. Using an E clock prescaler of 32 would not permit exact reproduction of a 10-kHz signal.

Example 2

Desired frequency = 60 Hz

To produce such a low frequency (compared to 4 MHz) with an 8-bit PWM channel, the clock S frequency reference must be used. Initially, we will choose an E clock prescaler of 8 and an S prescaler of 60.

Inspection of this equation reveals that a value of PWPER between 69 and 70 allows a 60-Hz signal to be produced. Solving the equation for desired frequency with PWPER values of 69 and 70 results in frequencies of 60.386 Hz and 59.524 Hz. Further inspection reveals that the product of PWPER and the S prescaler should be 4166.6667, so there is no way to exactly produce a 60-Hz signal by varying PWPER and the S prescaler. But the accuracy can be improved by using an E clock prescaler of 4. This will require a value of the S prescaler between 138 and 140 to achieve a duty cycle of exactly 50%. The equation can now be solved for desired frequency using 139 for the S prescaler and 60 for PWPER.

Programming Examples

Now that you are familiar with the use of the PWPER equation, the appropriate registers can be programmed to generate PWM signals. To make the code examples which follow as clear as possible, we use Motorola's register mnemonics rather than hexadecimal addresses when specifying the PWM control registers.

The specific address references which follow assume the INIT register, which is used to relocate the internal memory resources of the MC68HC11, has not been modified. For example, if you relocated the internal register block to memory page $4000 by writing $04 to INIT, then PWCLK in the next paragraph would be located at $4060 instead of $0060.

The first register that needs to be programmed is PWCLK at $0060 which contains these fields:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
E/(128*2)	E/(128*2)	PCKA2	PCKA1	0	PCKB3	PCKB2	PCKB1

CON34 and CON12 are used to concatenate PWM channels 3 and 4 and PWM channels 1 and 2 to produce two independent 16-bit PWM channels. We will set these bits to 0 when using 8-bit PWM channels. PCKA[2:1] set the E clock prescaler used to generate the clock A frequency reference.

PCKA[2:1]	Value of Clock A
00	E
01	E/2
10	E/4
11	E/8

Just like PCKA[2:1], PCKB[3:1] are used to set the E clock prescaler used to generate the clock B frequency reference.

PCKB[3:1]	Value of Clock B
000	E
001	E/2
010	E/4
011	E/8
100	E/16
101	E/32
110	E/64
111	E/128

The PWPOL register at $0061 allows us to select a clock source and a signal polarity for each PWM channel. PWPOL has these fields.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PCLK4	PCLK3	PCLK2	PCLK1	PPOL4	PPOL3	PPOL2	PPOL1

PCLK[4:3] select the clock source for PWM channels 3 and 4, where a 0 selects the clock B reference (E clock divided by 1, 2, 4, 8...128) and a 1 selects the clock S reference (clock A divided by twice a value from 1 to 256).

PCLK[2:1] select the clock source for PWM channels 1 and 2 where a 0 selects the clock A reference (E clock divided by 1, 2, 4, or 8) and a 1 selects the clock S reference (clock A divided by twice a value from 1 to 256).

PPOL[4:1] select the polarity of each PWM channel, where a 0 indicates that the PWM signal will initially be low, changing to high when the duty cycle count is reached. Likewise, placing a 1 in the appropriate PPOL[4:1] bit indicates that the PWM signal will initially be high, changing to low when the duty cycle count is reached.

If you are using the clock S reference to generate lower frequencies, you will have to set the PWSCAL register at $0062 with the S prescaler. The value in PWSCAL is multiplied by two when scaling the clock A reference. A PWSCAL value of $00 is equivalent to an S prescaler of 256, thus allowing the clock A reference to be divided by a maximum of 512 (2 * 256).

The PWCNT1–PWCNT4 ($0064 to $0067) registers hold the running values of the counters for each PWM channel. A write to these registers sets them to 0. Before enabling a PWM channel, you should clear its associated PWCNT register.

The PWPER1–PWPER4 ($0068 to $006B) registers are used to set the period values of each PWM signal generated. When the value in a particular PWCNT register matches the value in the associated PWPER register, a new wave form cycle is started. Values of $00 or $01 placed in these registers will not generate a periodic signal at the associated PWM pin. The pin will take on the polarity value specified by the appropriate PPOL bit in the PWPOL register.

The PWDTY1–PWDTY4 ($006C to $006F) registers are used to specify the duty cycle count for each PWM channel. When the running counter in the PWCNT register for a particular channel matches the value in the proper PWDTY register, the polarity of that channel is reversed. Thus a value of $00 placed in a PWDTY register will not generate a periodic signal at the associated PWM pin. The pin will take on the polarity value specified by the appropriate PPOL bit in the PWPOL register. If the value in a particular PWDTY register is greater than or equal to the value in its PWPER register, the associated PWM pin will take on the polarity value specified by the appropriate PPOL bit in the PWPOL register.

The PWPER1–PWPER4 and PWDTY1–PWDTY4 registers are double-buffered so that writes to these registers will not take effect until completion of the current PWM wave form. Writes to the PWCNT1–PWCNT4 registers take effect immediately and can be used to prematurely truncate wave forms.

The PWEN register at $0063 allows each PWM channel to be enabled. When the MC68HC11 is operating in special test mode, PWEN allows certain PWM operating conditions to be modified.

The PWEN register contains these fields:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TPWSL	DISCP	0	0	PWEN4	PWEN3	PWEN2	PWEN1

In special test mode, setting TPWSL to 1 will output the value of the clock S counter to the PWSCAL register. Writes to modify the PWSCAL register will still function. When TPWSL is set to 0, reads of PWSCAL will always return the most recent value written to it. In special test mode, setting the DISCP bit to 1 effectively disables the PWSCAL register, forcing the S prescaler to be 256. Specifically, the counter that implements the clock S reference still runs, but it is not reset when its value matches that of the PWSCAL register. Writes to the PWSCAL register thus have no effect. When DISCP is set to 0, the PWSCAL register will function as expected. Finally, to enable a particular PWM channel, you must set its associated PWEN bit to 1.

Each PWM channel is associated with one of the port H output pins. The signals generated by 8-bit PWM channels 1, 2, 3, and 4 will appear on port H pins PH0, PH1, PH2, and PH3 respectively. The following code examples demonstrate the set up of the PWM registers for the frequencies and values specified in the calculations above.

Example 1

Desired frequency = 10 kHz

In the calculations above, we solved the PWPER equation for a value of 200. We will also set up this 10-kHz signal for a duty cycle of 50% with the PWM output starting out low and changing to high when the duty cycle count is reached.

```
LDAA    #%00010000      ;set the Clock A reference to E/2
STAA    PWCLK
CLR     PWPOL           ;let PWM channel 1 use the Clock A
                        reference and make the channel 1 output
                        low during the first
                        ;part of the duty cycle
CLR     PWCNT1          ;reset PWCNT1 before using channel 1
LDAA    #200            ;we need a period of 200 for a 10 kHz
                        signal
STAA    PWPER1
LDAA    #100            ;this gives us a duty cycle of 50%
STAA    PWDTY1
LDAA    #%00000001      ;enable PWM channel 1 on PH0
STAA    PWEN
```

Example 2 Desired frequency = 60 Hz

In the calculations above, we could not solve the PWPER equation for an integer value, but we could generate a signal very close to 60 Hz (actually 59.95 Hz) by using a smaller clock prescaler. We will set up PWM channel 2 to output a 60-Hz signal with a duty cycle of 10% and with the initial polarity high.

```
LDAA    #%00100000      ;the Clock A reference to E/4
STAA    PWCLK
LDAA    #%00100010      ;let PWM channel 2 use the Clock S
                        reference and
                        ;make the channel 2 output high during
                        the first
                        ;part of the duty cycle
STAA    PWPOL
LDAA    #139            ;use an S prescaler of 2 * 139
STAA    PWSCAL
CLR     PWCNT2          ;reset PWCNT2 before using channel 2
LDAA    #60             ;we need a period of 60 for a 60 Hz
                        signal
STAA    PWPER2
LDAA    #06             ;this gives us a duty cycle of 10%
STAA    PWDTY2
LDAA    #%00000010      ;enable PWM channel 2 on PH1
STAA    PWEN
```

Example 3 Desired frequencies of 10 kHz and 60 Hz

Both of these wave forms can be generated simultaneously with a somewhat more complex set up. Use the clock A reference in order to use clock S, so the 10-kHz signal has to be generated on channel 3 using the clock B reference. The 60-Hz signal can be generated on channel 2 as in the above example. We will use the same duty cycles and polarities as specified above.

```
LDAA    #%00100001      ;set the Clock A reference to E/4
                         and the
                        ;Clock B reference to E/2
STAA   PWCLK
LDAA    #%00100010      ;PWM channel 3 will use the Clock B
                         reference
                        ;and initial low polarity
                        ;PWM channel 2 will use the Clock S
                         reference
                        ;and initial high polarity
STAA    PWPOL
LDAA    #139            ;use an S prescaler of 2 * 139
```

```
STAA    PWSCAL
CLR     PWCNT2]          ;reset PWCNT2 before using channel 2
CLR     PWCNT3           ;reset PWCNT3 before using channel 3
LDAA    #60              ;we need a period of 60 for a 60 Hz
                          signal
STAA    PWPER2
LDAA    #06              ;this gives us a duty cycle of 10%
STAA    PWDTY2           ;channel 2 is now ready for the 60 Hz
                          signal
LDAA    #200             ;we need a period of 200 for a 10 kHz
                          signal
STAA    PWPER3
LDAA    #100             ;this gives us a duty cycle of 50%
STAA    PWDTY3           ;channel 3 is now ready for the 10 kHz
                          signal
LDAA    #%00000110       ;enable PWM channels 3 and 2 on PH2 and
                          PH1
STAA    PWEN
```

The previous examples should allow you to set up the PWM timer channels to generate signals with a variety of frequencies. The only real limitation will be your choice of E clock prescalers for the clock A and clock B references. All four PWM channels may be used simultaneously to generate a wide range of frequencies, but some accuracy may have to be sacrificed if the chosen E clock prescalers do not allow you to select exact PWPER values.

16-Bit PWM Generation

Practical Limits and Frequency Calculations

Now that you are familiar with 8-bit PWM signal generation, 16-bit PWM operation will be fairly simple to explain. With 16-bit periods and duty cycles, signals can be produced with longer periods and more accurate duty cycles. A table for the minimum and maximum frequencies that can be produced with 16-bit PWM channels can be derived, just as with 8-bit PWM channels.

E Clock Divisor	Clock A — Reference 16-bit channel 1 minimum frequency	Clock A — Reference 16-bit channel 1 maximum frequency	Clock B — Reference 16-bit channel 2 minimum frequency	Clock B — Reference 16-bit channel 2 maximum frequency	Clock S — Reference both channels minimum frequency	Clock S — Reference both channels maximum frequency
1	E/(1*65535)	E/2	E/(1*65535)	E/2	E/(1*2*256*65535)	E/(1*2*2)
2	E/(2*65535)	E/4	E/(2*65535)	E/4	E/(2*2*256*65535)	E/(2*2*2)
4	E/(4*65535)	E/8	E/(4*65535)	E/8	E/(4*2*256*65535)	E/(4*2*2)
8	E/(8*65535)	E/16	E/(8*65535)	E/16	E/(8*2*256*65535)	E/(8*2*2)
16	Not available	Not available	E/(16*65535)	E/32	Not available	Not available
32	Not available	Not available	E/(32*65535)	E/64	Not available	Not available
64	Not available	Not available	E/(64*65535)	E/128	Not available	Not available
128	Not available	Not available	E/(128*65535)	E/256	Not available	Not available

From this table, we can see that with a 4-MHz E clock, the maximum 16-bit PWM frequency would still be 2 MHz, but the minimum 16-bit PWM frequency would be 0.0149 Hz or a period of 67.1 seconds. With a 16-bit channel, a 10-kHz signal can be produced with a maximum useful duty cycle of 99.75%. At lower frequencies, the additional precision allows a 60-Hz square wave with a maximum useful duty cycle of 99.997% to be produced. More specifically, at 60 Hz the duty cycle may be adjusted with a tolerance of 0.3%.

We will again use the PWPER equation to calculate the contents of the PWPER1–PWPER4 registers and to determine the accuracy of the signal produced. As above, if PWPER is less than 2, choose a smaller prescaler; however, if PWPER is greater than 65535, choose a larger prescaler. If PWPER is an integer, your frequency can be produced exactly. If PWPER is not an integer, there will be some degree of error in the signal generated, but it will be much smaller than that present when an 8-bit PWM channel is used. The following examples will demonstrate this.

Example 1

Desired frequency = 60 Hz

With an 8-bit PWM channel, the closest we could come to 60 Hz was 59.952 Hz. With 16 bits, accuracy improves substantially. We will first try to solve the PWPER equation using an E clock prescaler of 1.

As we noted above for a 16-bit PWM channel, a value of PWPER greater than 65535 means we need to choose a larger prescaler. Next we will try an E clock prescaler of 2.

This is a legal value of PWPER for a 16-bit PWM channel, but as noted above, if PWPER is not an integer, we will not be able to exactly produce our frequency; however, we should solve the equation using a PWPER value of 33333 to see how much error will be present in the PWM signal produced.

For all practical purposes, this is 60 Hz. If a 50% duty cycle were selected for this signal, it, too, would be inexact by a fraction of a percent, but these errors are small enough to be ignored in all but the most sensitive applications.

Example 2

Desired frequency = 1/60 Hz, a period of 60 seconds

Sixteen bits of precision allow us to produce signals with long periods using the E clock prescalers alone, but for this case the clock S reference must be used. Let us first try to solve the PWPER equation with an E clock prescaler of 8 and an S prescaler of 256.

This value of PWPER is not an integer so the 1/60-Hz signal would not be produced exactly. Upon further inspection, we see that the product of PWPER * S prescaler should be 15000000, an integer. We can thus find values for PWPER and the S prescaler that will allow 1/60 Hz to be produced exactly. For convenience, we will use a PWPER of 60000 and an S prescaler of 250. Solving the equation for desired frequency, we find that 1/60 Hz can be produced exactly.

Programming Examples

By setting the CON34 and CON12 bits in the PWCLK register, we can concatenate the PWPER1–PWPER2, PWPER3–PWPER4, PWCNT1–PWCNT2, PWCNT3–PWCNT4, PWDTY1–PWDTY2, and PWDTY3–PWDTY4 registers to form two 16-bit PWM channels. If only one 16-bit channel is used, the remaining two 8-bit channels may also be used independent of the operation of the 16-bit channel.

The only possible problem here may arise if you must use the clock S reference which, as mentioned previously, depends upon the clock A reference. This could affect your choice of a prescaler for the clock A reference and the accuracy with which you can produce certain frequencies. Example 3 that follows is affected by this particular constraint.

Programming the PWM registers for 16-bit operation is not much different than it is for 8-bit operation. This table summarizes the programming changes for 16-bit PWM operation.

16-Bit PWM Channel 1	16-Bit PWM Channel 2	Concatenation Bit
CON12	CON34	Available frequency references
Clock A or clock S	Clock B or clock S	Frequency reference select bit
PCLK2	PCLK4	Polarity select bit
PPOL2	PPOL4	16-bit counter high byte
PWCNT1	PWCNT3	16-bit counter low byte
PWCNT2	PWCNT4	16-bit period high byte
PWPER1	PWPER3	16-bit period low byte
PWPER2	PWPER4	16-bit duty cycle high byte
PWDTY1	PWDTY3	16-bit duty cycle low byte
PWDTY2	PWDTY4	Enable bit
PWEN2	PWEN4	Port H output pin
PH1	PH3	—

As you can see from this table, the only real changes involve the period, counter, and duty cycle registers which are now 16 bits wide. The frequency reference, polarity, and enable bits for 16-bit PWM channel 1 are the same as those for 8-bit PWM channel 2. Likewise, the frequency reference, polarity, and enable bits for 16-bit PWM channel 2 are the same as those for 8-bit PWM channel 4. The following examples implement the frequencies we determined above with the PWPER equation using the 16-bit PWM channels.

Example 1

Desired frequency = 60 Hz

In the calculations above, we solved the PWPER equation for a value of 33333 when using an E clock prescaler of 2. We will also set up this 60-Hz signal to have a duty cycle of 50% with the PWM output starting out high and changing to low when the duty cycle count is reached. The first 16-bit PWM channel will be used.

```
LDAA    #%01010000        ;concatenate channels 1 and 2
                          ;set the Clock A reference to E/2
STAA    PWCLK
LDAA    #%00000010        ;we use the PCLK2 and PPOL2 bits to set
the
                          ;first 16 bit PWM channel to use the
Clock A
                          ;reference and high polarity during the
first
                          ;part of the duty cycle
STAA    PWPOL
CLR     PWCNT1            ;we have to reset both PWCNT1 and
                          PWCNT2 before
CLR     PWCNT2            ;using the first 16 bit channel
LDD     #3333             ;we need a period of 33333 for a 60 Hz
                          signal
STD     PWPER1            ;this stores the 16 bit value in
                          PWPER1:PWPER2
LDD     #16666            ;this gives us a duty cycle of almost
                          exactly 50%
STD     PWDTY1            ;this stores the 16 bit value in
                          PWDTY1:PWDTY2
LDAA    #%00000010        ;enable the first 16 bit PWM channel on
                          PH1
STAA    PWEN
```

Example 2 Desired frequency = 1/60 Hz

In the calculations above, we found that we can exactly produce this signal. We will use 16-bit PWM channel 2 with the clock S reference. The S prescaler will be 250, PWPER will be 60000, and the duty cycle will be 25% with low initial polarity.

```
LDAA    #%10110000      ;concatenate channels 3 and 4
                        ;set the Clock A reference to E/2
STAA    PWCLK
LDAA    #%10000000      ;we use the PCLK4 and PPOL4 bits to set
                        the
                        ;second 16 bit PWM channel to use the
                        Clock S
                        ;reference and low polarity during the
                        first
                        ;part of the duty cycle
STAA    PWPOL
LDAA    #250            ;use an S prescaler of 2 * 250
STAA    PWSCAL
CLR     PWCNT3          ;we have to reset both PWCNT3 and PWCNT4
                        before
CLR     PWCNT4          ;using the second 16 bit channel
LDD     #60000          ;we need a period of 60000 for a 1/60 Hz
                        signal
STD     PWPER3          ;this stores the 16 bit value in
                        PWPER3:PWPER4
LDD     #15000          ;this gives us a duty cycle of exactly 25%
STD     PWDTY3          ;this stores the 16 bit value in
                        PWDTY3:PWDTY4
LDAA    #%00001000      ;enable the second 16 bit PWM channel on
                        PH3
STAA    PWEN
```

Example 3 Desired frequencies of 2 MHz, 60 Hz, and 1/60 Hz

This example demonstrates the wide range of frequencies the MC68HC11 microcontrollers with PWM timers are capable of generating when using a 4-MHz E clock. The PWPER calculations for 2 MHz are simple and left as an exercise for the reader. The PWPER calculations for 60 Hz are somewhat more involved but can be readily solved by examining those calculations used above for 60 Hz. We will produce 1/60 Hz on the first 16-bit channel, 60 Hz on 8-bit channel 3, and 2 MHz on 8-bit channel 4. All polarities will initially be low and all duty cycles will be 50%.

```
LDAA    #%01110000;concatenate channels 1 and 2
                  ;set the Clock A reference to E/8
                  ;set the Clock B reference to E/1
STAA    PWCLK
LDAA    #%01100000;8 bit channel 4 will use Clock B
                  ;8 bit channel 3 will use Clock S
                  ;the first 16 bit channel will use Clock S
                  ;all polarities will be low during the
                  first
                  ;part of the duty cycle
STAA    PWPOL
LDAA    #250      ;use an S prescaler of 2 * 250
STAA    PWSCAL
CLR     PWCNT1    ;reset all of the PWCNT registers
CLR     PWCNT2
CLR     PWCNT3
CLR     PWCNT4
LDD     #60000    ;we need a period of 60000 for a 1/60 Hz
                  signal
STD     PWPER1    ;this stores the 16 bit value in
                  PWPER1:PWPER2
LDD     #30000    ;this gives us a duty cycle of exactly 50%
STD     PWDTY1    ;this stores the 16 bit value in
                  PWDTY1:PWDTY2
LDAA    #17       ;this gives a channel 3 frequency of
                  58.82 Hz
STAA    PWPER3
LDAA    #08       ;set channel 3 duty cycle to almost 50%
STAA    PWDTY3
LDAA    #02       ;this gives channel 4 a frequency of 2 MHz
STAA    PWPER4
LDAA    #01       ;set channel 4 duty cycle to exactly 50%
STAA    PWDTY4
LDAA    #%00001110;enable 8 bit channels 3 and 4 and the
                  first
                  ;16 bit channel on PH2, PH3, and PH1
                  respectively
STAA    PWEN
```

This example demonstrates mixed 8- and 16-bit operation. The 1/60-Hz and 2-MHz frequencies will be produced exactly, but because of the limited resolution of the 8-bit PWM with such a large E clock prescaler, the closest we can come to 60 Hz is 58.82 Hz.

How to reach us:

USA/EUROPE/Locations Not Listed: Motorola Literature Distribution, P.O. Box 5405, Denver, Colorado 80217. 1-800-441-2447 or 1-303-675-2140. Customer Focus Center, 1-800-521-6274

JAPAN: Motorola Japan Ltd.: SPD, Strategic Planning Office, 141, 4-32-1 Nishi-Gotanda, Shinagawa-Ku, Tokyo, Japan, 03-5487-8488

ASIA/PACIFIC: Motorola Semiconductors H.K. Ltd., Silicon Harbour Centre, 2 Dai King Street, Tai Po Industrial Estate, Tai Po, New Territories, Hong Kong, 852-26629298

Mfax™, Motorola Fax Back System: RMFAX0@email.sps.mot.com; http://sps.motorola.com/mfax/; TOUCHTONE, 1-602-244-6609; US and Canada ONLY, 1-800-774-1848

HOME PAGE: http://motorola.com/sps/