AN545

Using the Capture Module

Author: *Mark Palmer*
Microchip Technology Inc.

INTRODUCTION

The PICmicro™ family of RISC microcontrollers has been designed to provide advanced performance and a cost-effective solution for a variety of applications. This application note provides examples which illustrate the uses of input capture using the PIC17C42 Timer3 module. These examples may be modified to suit the specific needs of an application.

This application note has 4 examples that use the Timer3 input capture. They are:

- Frequency Counter (Period Measurement)
- Frequency Counter (Period Measurement) using a Free Running Timer
- Pulse Width Measurement
- Frequency Counter (Period Measurement) with Input Prescaler

TIMER3 DESCRIPTION

Timer3 is a 16-bit timer/counter that has two modes of operation which are software selected. The CA1/$\overline{\text{PR3}}$ bit (TCON2<3>) selects the mode of operation. The two modes are:

- Timer3 with Period Register and Single Capture Register (Figure 1).
- Timer3 and Dual Capture Registers (Figure 2).

Timer3 is the time-base for capture operations.

FIGURE 1: TIMER3 WITH PERIOD REGISTER AND SINGLE CAPTURE REGISTER

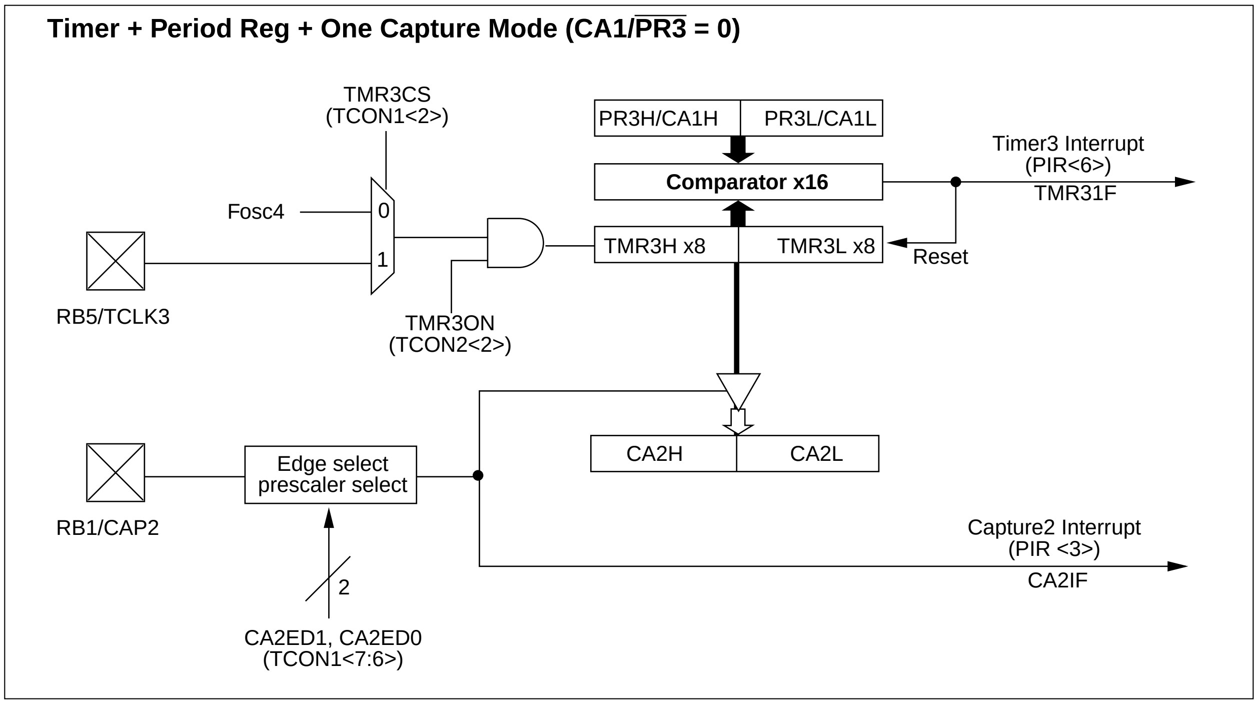
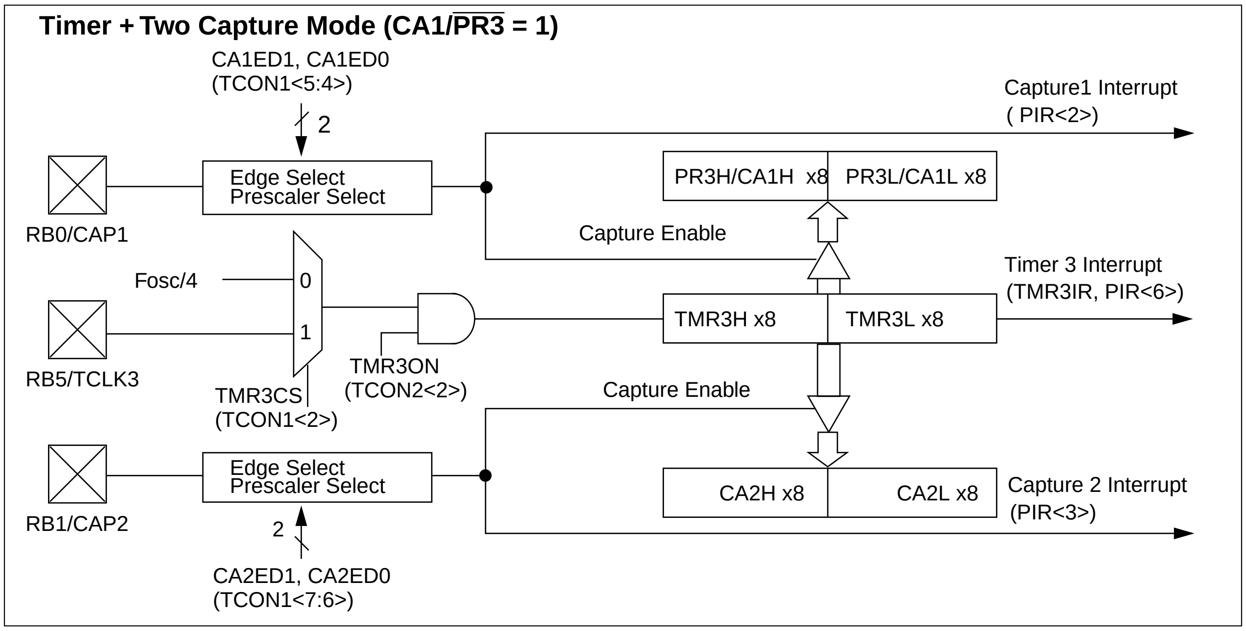


FIGURE 2: TIMER3 AND DUAL CAPTURE REGISTERS



The period register allows the time base of Timer3 to be something other than the 2^{16} counter overflow value, which corresponds to FFFFh (65536) cycles. This is accomplished by loading the desired period value into the PR3H/CA1H:PR3L/CA1L register pair. The overflow time can be calculated by this equation:

$T_{OFL} = T_{CLK} \bullet$ (value in PR3H/CA1H:PR3L/CA1L register pair + 1).

Where T_{CLK} is either the internal system clock (T_{CY}) or the external clock cycle time. Table 1 the shows time-out periods for different period values at different frequencies. The values in the register are the closest approximation for the period value. All examples in this application note uses a Timer3 overflow value of FFFFh.

TABLE 1: TIMER3 OVERFLOW TIMES

	Period Register					
Overflow Time	**@ 16 MHz (250 ns)**	**@ 10 MHz (400 ns)**	**@ 8 MHz (500 ns)**	**@5 MHz (500 ns)**	**@2 MHz (2.0 µs)**	**@32 kHz (125 µs)**
8.192 s	N.A.	N.A.	N.A.	N.A.	N.A.	0xFFFF
131.072 ms	N.A.	N.A.	N.A.	N.A.	0xFFFF	0x0418
52.428 ms	N.A.	N.A.	N.A.	0xFFFF	0x6666	0x01A3
32.7675 ms	N.A.	N.A.	0xFFFF	0x9FFF	0x3FFF	0x0106
26.214 ms	N.A.	0xFFFF	0xCE20	0x80D4	0x3388	0x00D3
16.384 ms	0xFFFF	0xA000	0x8000	0x5000	0x2000	0x0083
10.0 ms	0x9C40	0x61A8	0x4E20	0x30D4	0x1388	0x0050
4.0 ms	0x3E80	0x2710	0x1F40	0x1388	0x07D0	0x0020
1.0 ms	0x0FA0	0x09C4	0x07D0	0x04E2	0x01F4	0x0008
600 µs	0x0960	0x05DC	0x04B0	0x02EE	0x012C	0x0005
100 µs	0x0190	0x00FA	0x00C8	0x007D	0x0032	N.A.

The uses of an input capture are all for time based measurements. These include:

- Frequency measurement
- Duty cycle and pulse width measurements

The PIC17C42 has two pins (RB0/CAP1 and RB1/CAP2) which can be used for capturing the Timer3 value, when a specified edge occurs. The input capture can be specified to occur on one of the following four events:

- Falling Edge
- Rising Edge
- 4th Rising Edge
- 16th Rising Edge

These are specified bits 7:6 for CAP2 and 5:4 for CAP2 by the register TCON1<7:4>.

This flexibility allows an interface without the need of additional hardware to change polarity or specify an input prescaler.

The control registers that are used for by Timer3 are shown in Table 2. Shaded Boxes are control bits that are not used by the Timer3 module, the Peripheral Interrupt enable and flag bits, and the Global Interrupt enable bit.

TABLE 2: REGISTERS ASSOCIATED WITH TIMER3 AND CAPTURE

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-on Reset	Value on all other resets (Note1)
16h, Bank 3	TCON1	CA2ED1	CA2ED0	CA1ED1	CA1ED0	T16	TMR3CS	TMR2CS	TMR1CS	0000 0000	0000 0000
17h, Bank 3	TCON2	CA2OVF	CA1OVF	PWM2ON	PWM1ON	CA1/$\overline{\text{PR3}}$	TMR3ON	TMR2ON	TMR1ON	0000 0000	0000 0000
10h, Bank 2	TMR1	Timer1 register								xxxx xxxx	uuuu uuuu
11h, Bank 2	TMR2	Timer2 register								xxxx xxxx	uuuu uuuu
16h, Bank 1	PIR	RBIF	TMR3IF	TMR2IF	TMR1IF	CA2IF	CA1IF	TXIF	RCIF	0000 0010	0000 0010
17h, Bank 1	PIE	RBIE	TMR3IE	TMR2IE	TMR1IE	CA2IE	CA1IE	TXIE	RCIE	0000 0000	0000 0000
07h, Unbanked	INTSTA	PEIF	T0CKIF	T0IF	INTF	PEIE	T0CKIE	T0IE	INTE	0000 0000	0000 0000
06h, Unbanked	CPUSTA	—	—	STKAV	GLINTD	$\overline{\text{TO}}$	$\overline{\text{PD}}$	—	—	--11 11--	--11 qq--
14h, Bank 2	PR1	Timer1 period register								xxxx xxxx	uuuu uuuu
15h, Bank 2	PR2	Timer2 period register								xxxx xxxx	uuuu uuuu
10h, Bank 3	PW1DCL	DC1	DC0	—	—	—	—	—	—	xx-- ----	uu-- ----
11h, Bank 3	PW2DCL	DC1	DC0	TM2PW2	—	—	—	—	—	xx0- ----	uu0- ----
12h, Bank 3	PW1DCH	DC9	DC8	DC7	DC6	DC5	DC4	DC3	DC2	xxxx xxxx	uuuu uuuu
13h, Bank 3	PW2DCH	DC9	DC8	DC7	DC6	DC5	DC4	DC3	DC2	xxxx xxxx	uuuu uuuu

Legend: x = unknown, u = unchanged, - = unimplemented read as a '0', q - value depends on condition, shaded cells are not used by Timer1 or Timer2.

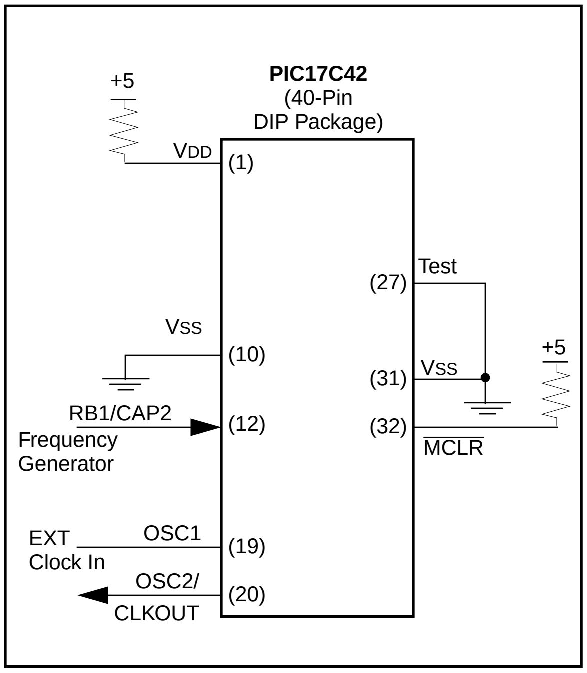
Note 1: Other (non power-up) resets include: external reset through $\overline{\text{MCLR}}$ and WDT Timer Reset.

This Application Note has 4 examples that USE the Timer3 input capture. They are:

- Frequency Counter (Period Measurement)
- Frequency Counter (Period Measurement) using a Free Running Timer
- Pulse Width Measurement
- Frequency Counter (Period Measurement) with Input Prescaler

All these examples can be run from a simple setup, this is shown in Figure 3.

FIGURE 3: APPLICATION HARDWARE SETUP



A discussion of each application with the operation of the software and application issues. The source listings for these are in appendices A-D.

PERIOD MEASUREMENT (FREQUENCY COUNTER)

Period measurement is simply done by clearing the counter to 0000h, then starting the counter on the 1st rising edge. On the following rising edge, the capture2 register is loaded with the Timer3 value and the TMR3 register CA2H:CA2L is cleared. If the period is greater than the overflow rate of TMR3, the register overflows, causing an interrupt. With a TMR3 overflow, an interrupt occurs and the overflow counter may need to be incremented. The overflow counter should be incremented if:

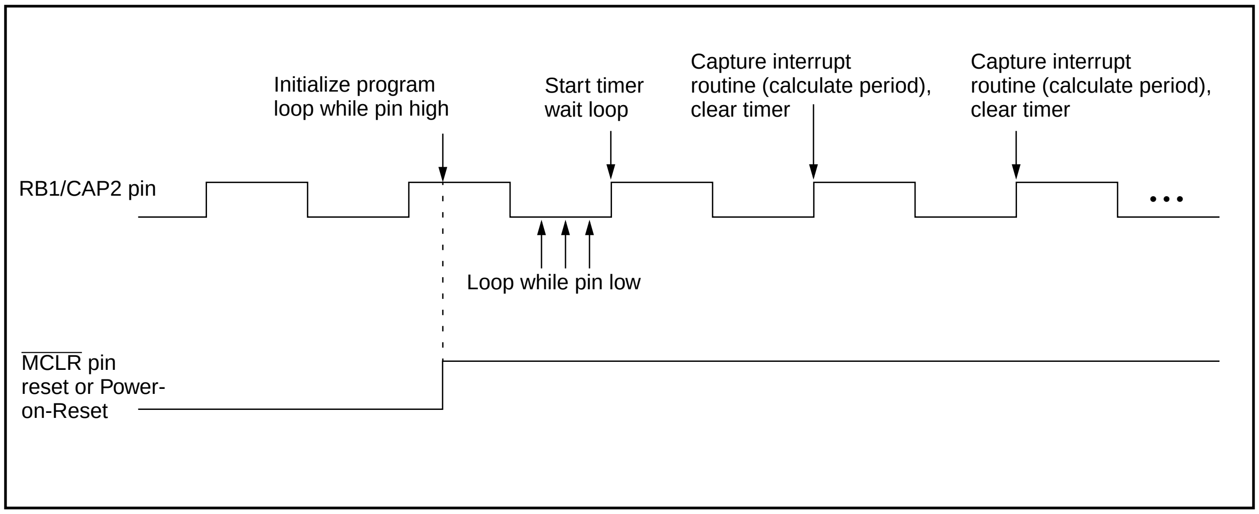
- The TMR3 overflow is the only interrupt source.
- Both the TMR3 overflow and capture2 interrupts occurred at near the same time, but the TMR3 overflow occurred first, Most Significant Byte of the Capture2 register is cleared (CA24-00h)).

Once a capture has occurred, the capture registers are moved to data RAM, the capture2 interrupt flag is cleared and the TMR3 register is loaded with an offset value. This offset value is the number of cycles from the time the interrupt routine is entered to when the TMR3 register is reloaded. In this example a data RAM location is used as an overflow counter. This gives in effect a 24-bit timer. The software flow for this routine is shown in Figure 4.

The program listing in Appendix A implements this, assuming only TMR3 overflow and capture2 interrupt sources. This example may be modified to suit the particular needs of your application. The following is a performance summary for this program (@ 16 MHz):

Code size:	30 Words
RAM used:	4 Bytes
Maximum frequency that can be measured:	130 kHz
Minimum frequency that can be measured:	0.25 Hz
Measurement Accuracy:	± TCY (± 250 ns)

FIGURE 4: SOFTWARE TIMING FLOW RELATIVE TO INPUT SIGNAL ON RB1/CAP2 PIN


Initialize program loop while pin high
Start timer wait loop
Capture interrupt routine (calculate period), clear timer
Capture interrupt routine (calculate period), clear timer
RB1/CAP2 pin
Loop while pin low
MCLR pin reset or Power-on-Reset

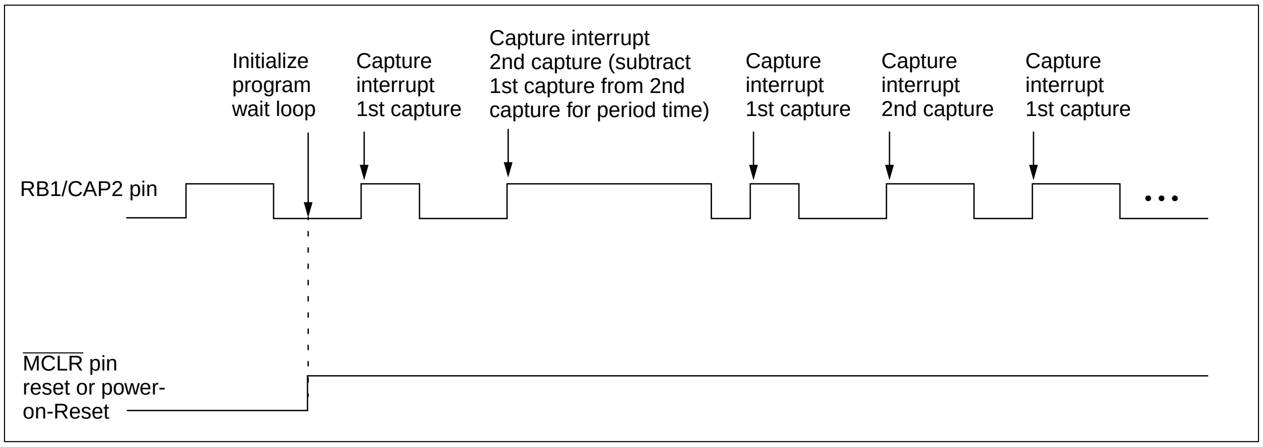
PERIOD MEASUREMENT (FREQUENCY COUNTER) USING A FREE RUNNING TIMER

In many applications the timer would need to be used for multiple tasks, and is required not to be reset (modified) by any one of these tasks. This is called a free running timer. To do period measurement in an application with a free-running timer, the program needs to store each capture in a data RAM location pair (word). The 1st capture in data RAM locations input capture2A (IC2AH:IC2AL) and the 2nd capture in data RAM locations input capture2B (IC2BH:IC2BL). Once the two captures have occurred, the values in these two words are subtracted. Since this is a free running timer, the value in input capture2B may be less than the value in input capture2A. This is if the 1st capture occurs, then the TMR3 overflows, and then the 2nd capture occurs. So an overflow counter should only be incremented if the TMR3 overflow occurs after a capture1 but before the capture2 occurs. With the use of an overflow counter this becomes an effective 24-bit period counter. The software flow for this routine is shown in Figure 5.

The program listing in Appendix B implements this, assuming only TMR3 overflow and capture2 interrupt sources. This example may be modified to suit the particular needs of your application. The following is a performance summary for this program (@ 16 MHz):

Code size:	41 Words
RAM used:	7 Bytes
Maximum frequency that can be measured:	71 kHz
Minimum frequency that can be measured:	0.25 Hz
Measurement Accuracy:	± TCY (± 250 ns)

FIGURE 5: SOFTWARE TIMING FLOW RELATIVE TO INPUT SIGNAL ON RB1/CAP2 PIN


Initialize program wait loop
Capture interrupt 1st capture
Capture interrupt 2nd capture (subtract 1st capture from 2nd capture for period time)
Capture interrupt 1st capture
Capture interrupt 2nd capture
Capture interrupt 1st capture
RB1/CAP2 pin
MCLR pin reset or power-on-Reset

PULSE WIDTH MEASUREMENT USING A FREE RUNNING TIMER

Applications that require the measurement of a pulse width can also be easily handled. The PIC17C42 can be programmed to measure either the low or the high pulse time. The software example in Appendix C measures the high pulse time. The program is initialized to capture on the rising edge of the RB1/CAP2 pin. After this event occurs, the capture mode is switched to the falling edge of the RB1/CAP2 pin. When the capture edge is modified (rising to falling, or falling to rising) a capture interrupt is generated. This "false" interrupt request must be cleared before leaving the interrupt service routine, or the program will immediately re-enter the interrupt service routine due to this "false" request. When the falling edge of the RB1/CAP2 pin occurs, the difference of the two capture values is calculated. The flow for this is shown in Figure 6.

Due to the software overhead of the peripheral interrupt routine the following are the limitations on the input signal on the RB1/CAP2 pin. This does not include any software overhead that may be required in the main routine, or if additional peripheral interrupt features need to be included. This is shown in Table 3. If you assume that the input is a square wave (high time = low time), one needs to take the worst case time of the two minimum pulse times (11 µs) times two, to determine the period. The maximum continuous input frequency would then be approximately 45.5 kHz. For a single pulse measurement, minimum pulse width is 4.5 µs.

The program listing in Appendix C implements this, assuming only TMR3 overflow and capture2 interrupt sources. This example may be modified to suit the particular needs of your application. The following is a performance summary for this program (@ 16 MHz):

Code size:	51 Words
RAM used:	7 Bytes
Maximum frequency that can be measured:	71 kHz
Minimum frequency that can be measured:	0.25 Hz
Measurement Accuracy:	± TCY (± 250 ns)

FIGURE 6: SOFTWARE TIMING FLOW RELATIVE TO INPUT SIGNAL ON RB1/CAP2

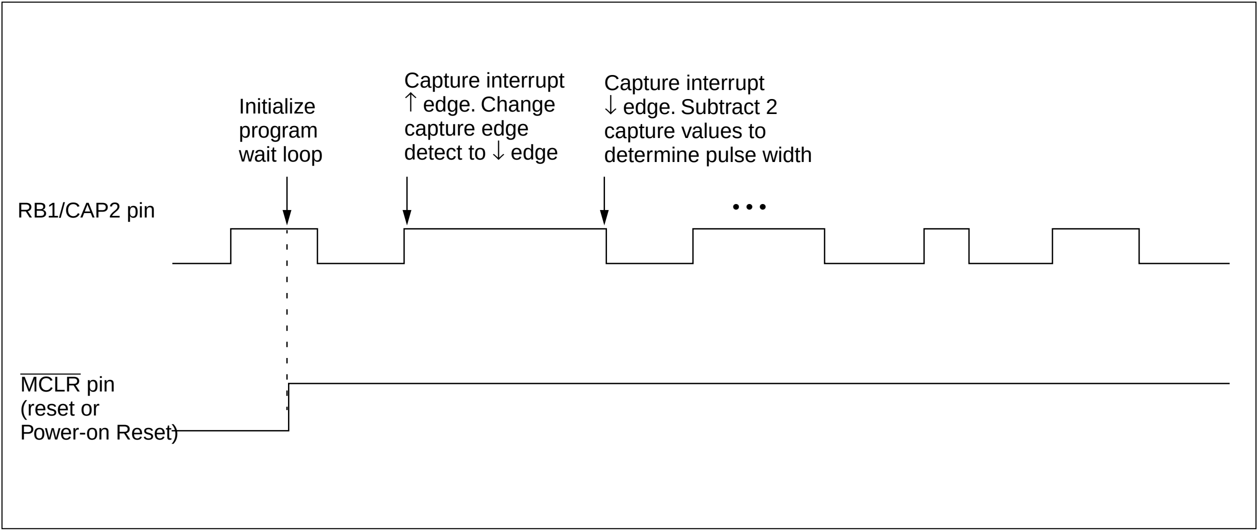

Initialize program wait loop
Capture interrupt ↑ edge. Change capture edge detect to ↓ edge
Capture interrupt ↓ edge. Subtract 2 capture values to determine pulse width
RB1/CAP2 pin
• • •
$\overline{\text{MCLR}}$ pin (reset or Power-on Reset)

TABLE 3: PERIPHERAL INTERRUPT ROUTINE

EVENT		# of Cycles	Time @ 16 MHz
1st CAPTURE	Capture1 only Capture1 and Timer Overflow	18 30	4.5 µs 7.5 µs
2nd CAPTURE	Capture only Capture and Timer Overflow	35 41	8.75 µs 10.25 µs
Minimum Pulse High	Capture1 and Timer Overflow + INT Latency	33	8.25 µs
Minimum Pulse Low	Capture2 and Timer Overflow + INT Latency	44	11 µs
Minimum Period (square wave)	2 • (Minimum Pulse Low)	88	22 µs

PERIOD MEASUREMENT (FREE RUNNING TIMER) WITH A PRESCALER

Occasionally the application may require a prescaler on the input signal. This may be due to application requirements, such as:

- Require higher resolution measurement of the input signal
- Reduce interrupt service overhead
- The input frequency is higher than interrupt service routine

The software selectable prescaler of the PIC17C42 allows the designer to easily implement this in their system without the cost of additional hardware. Care must be taken in determining if this option is appropriate. For example, if the input frequency is not stable (excessive frequency change per period) then the prescaler will give a less accurate capture value than the individual measurements.

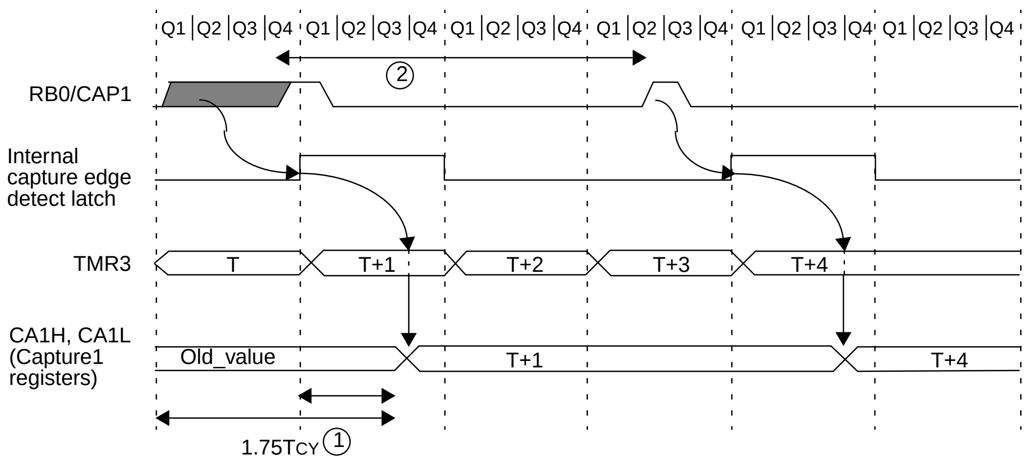
In cases where the resolution of the input frequency is important, the prescaler can be used to reduce the input capture error. There are two components to input capture:

- Resolution Error
- Input Synchronization Error

These two errors combine to form the total capture error. Resolution error is dependent on the rate at which the timer is incremented. Remember the timer may be based on an external clock (must be slower than TCY). The input synchronization error is dependent on the system clock speed (TCY), and will be less than TCY.

It is easy to see that when a capture occurs the synchronization error (TESYSC) can be up to 1 TCY (Figure 7). This error is constant regardless of the number of edges that occur before the capture is taken. So a capture on the 1st edge gives a synchronization error per sample up to TCY. While a capture taken on the 16th edge gives a synchronization error per sample only up to TCY / 16, by achieving a smaller percentage of error, the captured value becomes more accurate.

FIGURE 7: SYNCHRONIZATION ERROR WITH NO CAPTURE PRESCALER


Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4
2
RB0/CAP1
Internal capture edge detect latch
TMR3
T
T+1
T+2
T+3
T+4
CA1H, CA1L (Capture1 registers)
Old_value
T+1
T+4
1.75TCY 1

Note 1: Capture edge to actual register update latency is 1.75 TCY maximum, 0.75 TCY minimum. This implies that when measuring a pulse or a period, the measurement error is ± TCY.

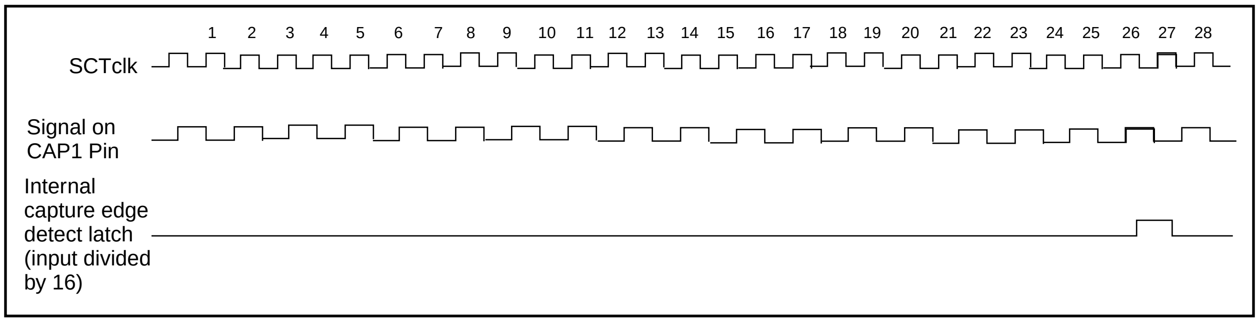
Note 2: With no prescaler on the input capture, two consecutive capturing edges must be apart by at least TCY. This allows the internal “capture edge detect latch” to reset.

Another scenario is when the signal on the input capture pin is different than the system cycle time (TCY), for example 1.6TCY. If you tried to capture this you would have a capture value of 1. If you set the prescaler to actually capture on the 16th edge you would have 16 * 1.6 TCY = 25.6TCY, which would be latched on the 26th TCY (Figure 8). This 0.4 TCY error is over 16 samples, which therefore gives an effective error/sample of 0.025TCY.

The program listing in Appendix D implements this, assuming only TMR3 overflow and Capture2 interrupt sources. This example may be modified to suit the particular needs of your application. The following is a performance summary for this program (@ 16 MHz):

Code size:	41 Words
RAM used:	7 Bytes
Maximum frequency that can be measured:	80 kHz
Minimum frequency that can be measured:	0.25 Hz
Measurement Accuracy:	± TCY (± 16.625 ns)

FIGURE 8: INPUT CAPTURE DIVIDED BY 16 PRESCALE EXAMPLE



Please check the Microchip BBS for the latest version of the source code. Microchip's Worldwide Web Address: www.microchip.com; Bulletin Board Support: MCHIPBBS using CompuServe® (CompuServe membership not required).

APPENDIX A: PERIOD MEASUREMENT EXAMPLE CODE

```
MPASM 01.40 Released         IC_D16_2.ASM   1-16-1997  15:15:17         PAGE  1


LOC  OBJECT CODE     LINE SOURCE TEXT
  VALUE

                    00001         LIST    P = 17C42, n = 66
                    00002 ;
                    00003 ;****************************************************************
                    00004 ;
                    00005 ; This is the basic outline for a program that can determine the
                    00006 ; frequency of an input, via input capture. The input capture has
                    00007 ; been selected to capture on the 16th rising edge. This is useful
                    00008 ; for high frequency inputs, where an interrupt on each rising edge
                    00009 ; would not be able to be serviced (at that rate). This particular
                    00010 ; example can support an input signal with a period of approximatly
                    00011 ; 625 nS. Without the divide by 16 selected, this is approximatly
                    00012 ; 10 us. This period time increases (frequency decreases) as the
                    00013 ; overhead in the main routine increases.
                    00014 ;
                    00015 ; This routine uses an 8-bit register to count the times that timer3
                    00016 ; overflowed. At the Max crystal frequency of 16 MHz, this gives an
                    00017 ; overflow time of (16)(2**8 + 1)(2**16)(250 nS) > 67.37 sec. If
                    00018 ; measurement of longer time intervals is required, the overflow
                    00019 ; counter could be extended to 16 (or more) bits.
                    00020 ;
                    00021 ; Timer 3 in this example is a free running timer. The input
                    00022 ; capture is generated on the RB1/CAP2 pin. There is a flag
                    00023 ; that specifies if this is the 1st or 2nd capture.
                    00024 ; The first capture is the start of the period measurement. The
                    00025 ; second capture value gives the end of the period. In this type
                    00026 ; of measurement If the 2nd capture value < the 1st captue value
                    00027 ; then the overflow counter should be decremented.
                    00028 ;
                    00029 ;       Program:          IC_D16_2.ASM
                    00030 ;       Revision Date:
                    00031 ;                         1-14-97    Compatibility with MPASMWIN 1.40
                    00032 ;
                    00033 ;*********************************************************************
                    00034 ;
                    00035 ;
                    00036 ; Do the EQUate table
                    00037 ;
  00000020          00038 IC2OF           EQU     0x20        ; T3 overflow register
  00000021          00039 IC2BH           EQU     0x21        ; T3 ICA2 MSB register (2nd Cap)
  00000022          00040 IC2BL           EQU     0x22        ; T3 ICA2 LSB register
  00000023          00041 IC2AH           EQU     0x23        ; T3 ICB2 MSB register (1st Cap)
  00000024          00042 IC2AL           EQU     0x24        ; T3 ICB2 LSB register
  00000025          00043 T3OFLCNTR       EQU     0x25        ; Temperay T3 overflow register
                    00044 ;
  00000026          00045 FLAG_REG        EQU     0x26        ; Register that has the Flag bits
                    00046 ;
                    00047 ;    FLAG_REG  bit   7   6   5   4   3   2   1   0
                    00048 ;                    -   -   -   -   -   -  UFL CAP1
                    00049 ;    CAP1 = 0,  1st Capture
                    00050 ;         = 1,  2nd Capture
                    00051 ;
                    00052 ;    UFL = 0,  No Underflow
                    00053 ;        = 1,  Underflow during subtract
                    00054 ;
  000007FF          00055 END_OF_PROG_MEM         EQU 0x07FF
```

```
                    ;
                    ;
  00000004          ALUSTA          EQU     0x04
  00000006          CPUSTA          EQU     0x06
  00000007          INTSTA          EQU     0x07
  0000000A          W               EQU     0x0A
                    ;
  00000012          PORTB           EQU     0x12        ; Bank 0
                    ;
  00000016          PIR             EQU     0x16        ; Bank 1
  00000017          PIE             EQU     0x17
                    ;
  00000012          TMR3L           EQU     0x12        ; Bank 2
  00000013          TMR3H           EQU     0x13
  00000016          T3PRL           EQU 0x16
  00000017          T3PRH           EQU 0x17
                    ;
  00000014          CA2L            EQU     0x14        ; Bank 3
  00000015          CA2H            EQU     0x15
  00000016          TCON1           EQU     0x16
  00000017          TCON2           EQU     0x17
                     PAGE
0000                                ORG     0x0000      ; Origin for the RESET vector
0000 C028                           GOTO    START       ; On reset, go to the start of
                                                        ;   the program
0008                                ORG     0x0008      ; Origin for the external RA0/INT
                                                        ;   interrupt vector
0008 C068                           GOTO    EXT_INT     ; Goto the ext. interrupt
                                                        ;   on RA0/INT routine
0010                                ORG    0x0010       ; Origin for the TMR0
                                                        ;   overflow interrupt vector
0010 C069                           GOTO    TMR0INT     ; Goto the TMR0 overflow interrupt
                                                        ;   routine
0018                                ORG     0x0018      ; Origin for the external
                                                        ;   RB1/T0CKI interrupt vector
0018 C06A                           GOTO    T0INT      ; Goto the ext. interrupt on
                                                        ;   RB1/T0CKI routine
0020                                ORG     0x0020      ; Origin for the interrupt vector
                                                        ;   of any enabled peripheral
0020 C03E                           GOTO    PER_INT     ; Goto the interrupt from a
                                                        ;   peripheral routine
                     PAGE
0028                                ORG     0x0028      ; Origin for the top of
                                                        ;   program memory
0028 8406           START           BSF     CPUSTA,4    ; Disable ALL interrupts via the
                                                        ;   Global Interrupt Disable
                                                        ;   (GLINTD) bit.
                                                        ;
0029                MAIN                                ; Place Main program here
0029 B803                           MOVLB   3           ; Select register Bank 3
002A 2817                           CLRF    TCON2,0     ; Stop the timers, Single Capture
002B B0F0                           MOVLW   0x0F0       ; Initalize TCON1 so that
002C 0116                           MOVWF   TCON1       ; T1 (8-bit), T2 (8-bit),
                                                        ; and T3 run off the internal
                                                        ; system clock. Capture2 captures
                                                        ; on the 16th rising edge.
                    ;
                    ; Initialize Timer 3, load the timer with the number of cycles that
                    ; the device executes (from RESET) before the timer is turned on
                    ; Therefore the offset is required due to software overhead.
                    ;
002D B802                           MOVLB   2           ; Select register Bank 2
002E 280A                           CLRF    W,0         ; Clear the W register
002F 0126                           MOVWF   FLAG_REG    ; Initalize to 0
0030 0113                           MOVWF   TMR3H       ; Timer3 MSB = 0
0031 B000                           MOVLW   0x00        ; Timer3 LSB = Offset
```

```
0032 0112                   MOVWF   TMR3L       ;
            ;
            ; Load the Timer 3 period register with 0xFFFF, which will give an
            ; interrupt on the overflow of Timer3
            ;
0033 B0FF                   MOVLW   0xFF        ;
0034 0117                   MOVWF   T3PRH       ;
0035 0116                   MOVWF   T3PRL       ;
            ;
            ; the timer should be started and interrupts enabled.
            ;
0036 B803                   MOVLB   3           ; Select register Bank 3
0037 8217                   BSF     TCON2,2     ; Turn on timer 3.
0038 8307                   BSF     INTSTA,3    ; Turn on Peripheral Interrupts
0039 B801                   MOVLB   1           ; Select register Bank 1
003A B048                   MOVLW   0x48        ; Enable Caputure 2 and Timer3
003B 0117                   MOVWF   PIE         ;    Interrupts (when GLINTD = 0)
            ;
            ; This is where you would do the things you wanted to do.
            ; this example will only loop waiting for the interrupts.
            ;
003C 8C06   WAIT            BCF     CPUSTA,4    ; Enable ALL interrupts
003D C03C                   GOTO    WAIT        ; Loop here waiting for a timer
                                                ;   Interrupt
             PAGE
            ;
            ; The interrupt routine for any peripheral interrupt, This routine
            ; only deals with Timer3 (T3) interrupts.
            ;
            ; Time required to execute interrupt routine. Not including
            ; interrupt latency (time to enter into the interrupt routine)
            ;
            ;       case1 - only T3 overflow             = 12 cycles
            ;       case2 - 1st capture                  = 14 cycles
            ;       case3 - 2nd capture                  = 30 cycles
            ;       case4 - T3 overflow and 1st capture  = 34 cycles
            ;       case5 - T3 overflow and 2nd capture  = 50 cycles
            ;
            ;
003E B801   PER_INT         MOVLB   1           ; Select register Bank 1
003F 9E16                   BTFSC   PIR,6       ; Did T3 overflow?
                                                ; If not skip next Instruction
0040 C055                   GOTO    T3OVFL      ; Inc overflow cntr and clear flag
0041 9316   CK_CAP          BTFSS   PIR,3       ; Did the RB1/CAP2 pin cause an
                                                ;   interrupt?
0042 0005                   RETFIE              ; No RB1/CAP2 interrupt,
                                                ;   Return from Interrupt
            ;
            ; This potion of the code takes the 1st capture and stores its
            ; value in register pair IC2AH:IC2AL. When the 2nd capture
            ; is take, its value is stored in register pair IC2BH:IC2BL.
            ; A 16-bit subtract is performed, with the final 24-bit result
            ; being stored in IC2OF:IC2BH:IC2BL. This value will no longer
            ; be correct after the next capture occurs (IC2BH:IC2BL will
            ; change), so the main routine must utilize this value before
            ; it changes.
            ;
0043 8B16   CAPTURE         BCF     PIR,3       ; Clear Capture2 interrupt flag
0044 B803                   MOVLB   3           ; Select register Bank 3
0045 9826                   BTFSC   FLAG_REG,0  ; 1st or 2nd capture2?
0046 C04B                   GOTO    CAP2        ; It was the 2nd Capture
0047 5424   CAP1            MOVPF   CA2L,IC2AL  ; Move the captured value to
0048 5523                   MOVPF   CA2H,IC2AH  ;   temporary registers
0049 8026                   BSF     FLAG_REG,0  ; Have 1st capture2
004A 0005                   RETFIE              ; Return from Interrupt
                                                ;
             PAGE
004B 5422   CAP2            MOVPF   CA2L,IC2BL  ; Move the captured value to
004C 5521                   MOVPF   CA2H,IC2BH  ;   temporary registers
                                                ;   (to prevent being overwritten)
                                                ;
004D E061                   CALL    SUB16       ; Call routine which subtracts
                                                ;   2 16-bit numbers.
004E 9926                   BTFSC   FLAG_REG,1  ; Underflow during SUB16?
004F 0725                   DECF    T3OFLCNTR,1 ; Since underflow, decrement the
                                                ;   overflow counter.
0050 2926                   CLRF    FLAG_REG,1  ; Clear the flag bits for
                                                ;   underflow and 2nd capture2
0051 6A25                   MOVFP   T3OFLCNTR,W ; Store the T3 input capture
0052 4A20                   MOVPF   W,IC2OF     ;   overflow value in IC2OF
0053 2825                   CLRF    T3OFLCNTR,0 ; Clear the Data register which
                                                ;   counts how many times Timer 3
                                                ;   overflows.
0054 0005                   RETFIE              ; Return from interrupt
            ;
            ; When Timer 3 has overflowed, the overflow counter only should
            ; be incremented when the overflow occurs after a capture 1
            ; but before the capture 2. The 4 possible cases when entering
            ; the T3OVFL section of the PER_INT routine are as follows:
            ;   Case 1: T3 overflow (only) and FLAG_REG.0 = 0 (waiting
            ;           for Capture 1 to occur). Do Not increment counter
            ;   Case 2: T3 overflow (only) and FLAG_REG.0 = 1 (waiting
            ;           for Capture 2 to occur). Increment counter
            ;   Case 3: T3 Overflow happened after Capture. Do Not
            ;           increment overflow counter
            ;   Case 4: T3 Overflow occured before Capture 2 and FLAG_REG.0 = 1
            ;           (waiting for Capture 2 to occur). Increment counter

            ;
0055 8E16   T3OVFL          BCF     PIR,6       ; Clear Overflow interrupt flag
0056 9316                   BTFSS   PIR,3       ; Did the RB1/CAP2 pin also
                                                ;   cause an interrupt?
0057 C05E                   GOTO    FR0         ; No, Check if between 1st
                                                ;   and 2nd capture
0058 B803                   MOVLB   3           ; Bank 3
0059 280A                   CLRF    W,0         ; W = 0
005A 3115                   CPFSEQ  CA2H        ; if CA2H = 0, overflow happened
005B C05E                   GOTO    FR0         ;   first, must check FLAG_REG
                                                ;   bit 0
005C B801                   MOVLB   1           ; Back to bank 1
005D C043                   GOTO    CAPTURE     ; Capture happened first, do NOT
                                                ; Increment overflow counter
                                                ;   and do capture routine
005E 9826   FR0             BTFSC   FLAG_REG,0  ; Between Capture 1 and Capture 2?
005F 1525                   INCF    T3OFLCNTR,1 ; Yes, Inc. the overflow counter
0060 0005                   RETFIE              ; Return from overflow interrupt
            ;
0061 6A24   SUB16           MOVFP   IC2AL,W     ; Do the 16-bit subtraction
0062 0522                   SUBWF   IC2BL,1     ;
0063 6A23                   MOVFP   IC2AH,W     ;
0064 0321                   SUBWFB  IC2BH,1     ;
0065 9004                   BTFSS   ALUSTA,0    ; Is the result pos. or neg. ?
0066 8126                   BSF     FLAG_REG,1  ; neg., Set the underflow flag
0067 0002                   RETURN              ; Return from the subroutine
             PAGE
            ;
            ; Other Interrupt routines. (Not utilized in this example)
            ;
0068 0005   EXT_INT RETFIE                      ; RA0/INT interrupt routine
                                                ;   (NOT used in this program)
0069 0005   TMR0INT RETFIE                      ; TMR0 overflow interrupt routine
                                                ;   (NOT used in this program)
```

```
006A 0005          00254 T0INT  RETFIE                      ; RA1/T0CKI interrupt routine
                   00255                                    ;   (NOT used in this program)
                   00256                                    ;
006B C028          00257 SRESET  GOTO    START              ; If program became lost, goto
                   00258                                    ;   START and reinitialize.
                   00259 ;
                   00260 ;
                   00261 ; When the executed address is NOT in the program range, the
                   00262 ; 16-bit address should contain all 1's (a CALL 0x1FFF). At
                   00263 ; this location you could branch to a routine to recover or
                   00264 ; shut down from the invalid program execution.
                   00265 ;
07FF               00266                 ORG             END_OF_PROG_MEM ;
07FF C06B          00267                 GOTO    SRESET     ; The program has lost it's mind,
                   00268                                    ;   do a system reset
                   00269                 END
MEMORY USAGE MAP ('X' = Used,  '-' = Unused)

0000 : X-------X------- X-------X------- X-------XXXXXXXX XXXXXXXXXXXXXXXX
0040 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXX---- ----------------
07C0 : ---------------- ---------------- ---------------- ---------------X

All other memory blocks unused.

Program Memory Words Used:    74


Errors   :     0
Warnings :     0 reported,     0 suppressed
Messages :     0 reported,     0 suppressed
```

Please check the Microchip BBS for the latest version of the source code. Microchip's Worldwide Web Address: www.microchip.com; Bulletin Board Support: MCHIPBBS using CompuServe® (CompuServe membership not required).

APPENDIX B: PERIOD MEASUREMENT, FREE RUNNING TIMER EXAMPLE CODE

```
MPASM 01.40 Released          IC_FRT2.ASM   1-16-1997  15:15:48         PAGE  1


LOC  OBJECT CODE     LINE SOURCE TEXT
  VALUE

                    00001        LIST    P = 17C42, n = 66
                    00002 ;
                    00003 ;**********************************************************
                    00004 ;
                    00005 ; This is the basic outline for a program that can determine the
                    00006 ; frequency of an input, via input capture. This routine uses an
                    00007 ; 8-bit register to count the times that timer3 overflowed. At the
                    00008 ; Max crystal frequency of 16 MHz, this gives an overflow time of
                    00009 ; (2**16)(256 + 1)(250 nS) > 4.21 sec or a frequncy < 0.25 Hz. If
                    00010 ; measurement of longer time intervals is required, the overflow
                    00011 ; counter could be extended to 16 (or more) bits.
                    00012 ;
                    00013 ; Timer 3 in this example is a free running timer. The input
                    00014 ; capture is generated on the RB1/CAP2 pin. There is a flag
                    00015 ; that specifies if this is the 1st or 2nd capture.
                    00016 ; The first capture is the start of the period measurement. The
                    00017 ; second capture value gives the end of the period. In this type
                    00018 ; of measurement If the 2nd capture value < the 1st captue value
                    00019 ; then the overflow counter should be decremented.
                    00020 ;
                    00021 ; Do the EQUate table
                    00022 ;
  00000020          00023 IC2OF           EQU     0x20      ; T3 overflow register
  00000021          00024 IC2BH           EQU     0x21      ; T3 ICA2 MSB register (2nd Cap)
  00000022          00025 IC2BL           EQU     0x22      ; T3 ICA2 LSB register
  00000023          00026 IC2AH           EQU     0x23      ; T3 ICB2 MSB register (1st Cap)
  00000024          00027 IC2AL           EQU     0x24      ; T3 ICB2 LSB register
  00000025          00028 T3OFLCNTR       EQU     0x25      ; Temperay T3 overflow register
                    00029 ;
  00000026          00030 FLAG_REG        EQU     0x26      ; Register that has the Flag bits
                    00031 ;
                    00032 ;    FLAG_REG  bit   7   6   5   4   3   2   1   0
                    00033 ;                    -   -   -   -   -   -  UFL CAP1
                    00034 ;    CAP1 = 0,  1st Capture
                    00035 ;         = 1,  2nd Capture
                    00036 ;
                    00037 ;    UFL = 0,  No Underflow
                    00038 ;        = 1,  Underflow during subtract
                    00039 ;
                    00040 ;       Program:          IC_FRT2.ASM
                    00041 ;       Revision Date:
                    00042 ;                         1-14-97   Compatibility with MPASMWIN 1.40
                    00043 ;
                    00044 ;********************************************************************
                    00045 ;
                    00046 ;
  000007FF          00047 END_OF_PROG_MEM EQU 0x07FF
                    00048 ;
                    00049 ;
  00000004          00050 ALUSTA          EQU     0x04
  00000006          00051 CPUSTA          EQU     0x06
  00000007          00052 INTSTA          EQU     0x07
  0000000A          00053 W               EQU     0x0A
```

```
                  ;
  00000012        PORTB          EQU     0x12        ; Bank 0
                  ;
  00000016        PIR            EQU     0x16        ; Bank 1
  00000017        PIE            EQU     0x17
                  ;
  00000012        TMR3L          EQU     0x12        ; Bank 2
  00000013        TMR3H          EQU     0x13
  00000016        T3PRL          EQU     0x16
  00000017        T3PRH          EQU     0x17
                  ;
  00000014        CA2L           EQU     0x14        ; Bank 3
  00000015        CA2H           EQU     0x15
  00000016        TCON1          EQU     0x16
  00000017        TCON2          EQU     0x17
                   PAGE
0000                             ORG     0x0000      ; Origin for the RESET vector
0000 C028                        GOTO    START       ; On reset, go to the start of
                                                     ;   the program
0008                             ORG     0x0008      ; Origin for the external RA0/INT
                                                     ;   interrupt vector
0008 C068                        GOTO    EXT_INT     ; Goto the ext. interrupt
                                                     ;   on RA0/INT routine
0010                             ORG     0x0010      ; Origin for the TMR0
                                                     ;   overflow interrupt vector
0010 C069                        GOTO    TMR0INT     ; Goto the TMR0 overflow interrupt
                                                     ;   routine
0018                             ORG     0x0018      ; Origin for the external
                                                     ;   RB1/RT interrupt vector
0018 C06A                        GOTO    T0INT       ; Goto the ext. interrupt on
                                                     ;   RB1/RT routine
0020                             ORG     0x0020      ; Origin for the interrupt vector
                                                     ;   of any enabled peripheral
0020 C03E                        GOTO    PER_INT     ; Goto the interrupt from a
                                                     ;   peripheral routine
                   PAGE
0028                             ORG     0x0028      ; Origin for the top of
                                                     ;   program memory
0028 8406         START          BSF     CPUSTA,4    ; Disable ALL interrupts via the
                                                     ;   Global Interrupt Disable
                                                     ;   (GLINTD) bit.
                                                     ;
0029              MAIN                               ; Place Main program here
0029 B803                        MOVLB   3           ; Select register Bank 3
002A 2817                        CLRF    TCON2,0     ; Stop the timers, Single Capture
002B B070                        MOVLW   0x070       ; Initialize TCON1 so that
002C 0116                        MOVWF   TCON1       ;   T1 (8-bit), T2 (8-bit),
                                                     ;   and T3 runs off the internal
                                                     ;   system clock. Capture2
                                                     ;   captures on the rising edge.
                  ;
                  ; Initialize Timer 3, load the timer with the number of cycles that
                  ; the device executes (from RESET) before the timer is turned on
                  ; Therefore the offset is required due to software overhead.
                  ;
002D B802                        MOVLB   2           ; Select register Bank 2
002E 280A                        CLRF    W,0         ; Clear the W register
002F 0126                        MOVWF   FLAG_REG    ; Initalize to 0
0030 0113                        MOVWF   TMR3H       ; Timer3 MSB = 0
0031 B013                        MOVLW   0x13        ; Timer3 LSB = Offset
0032 0112                        MOVWF   TMR3L       ;
                  ;
                  ; Load the Timer 3 period register with 0xFFFF, which will give an
                  ; interrupt on the overflow of Timer3
                  ;
0033 B0FF                        MOVLW   0xFF        ;
```

```
0034 0117                            MOVWF   T3PRH        ;
0035 0116                            MOVWF   T3PRL        ;
                    ;
                    ; the timer should be started and interrupts enabled.
                    ;
0036 B803                            MOVLB   3            ; Select register Bank 3
0037 8217                            BSF     TCON2,2      ; Turn on timer 3.
0038 8307                            BSF     INTSTA,3     ; Turn on Peripheral Interrupts
0039 B801                            MOVLB   1            ; Select register Bank 1
003A B048                            MOVLW   0x48         ; Enable Capture 2 and Timer3
003B 0117                            MOVWF   PIE          ; Interrupts (when GLINTD = 0)
                    ;
                    ; This is where you would do the things you wanted to do.
                    ; this example will only loop waiting for the interrupts.
                    ;
003C 8C06           WAIT             BCF     CPUSTA,4     ; Enable ALL interrupts
003D C03C                            GOTO    WAIT         ; Loop here waiting for a timer
                                                          ;   Interrupt
                     PAGE
                    ;
                    ; The interrupt routine for any peripheral interrupt, This routine
                    ; only deals with Timer3 (T3) interrupts.
                    ;
                    ; Time required to execute interrupt routine. Not including
                    ; interrupt latency (time to enter into the interrupt routine)
                    ;
                    ;       case1 - only T3 overflow             = 12 cycles
                    ;       case2 - 1st capture                  = 14 cycles
                    ;       case3 - 2nd capture                  = 30 cycles
                    ;       case4 - T3 overflow and 1st capture  = 34 cycles
                    ;       case5 - T3 overflow and 2nd capture  = 50 cycles
                    ;
                    ;
003E B801           PER_INT          MOVLB   1            ; Select register Bank 1
003F 9E16                            BTFSC   PIR,6        ; Did T3 overflow?
                                                          ;   If not skip next Instruction
0040 C055                            GOTO    T3OVFL       ; Inc overflow cntr and clear flag
0041 9316           CK_CAP           BTFSS   PIR,3        ; Did the RB1/CAP2 pin cause an
                                                          ;   interrupt?
0042 0005                            RETFIE               ; No RB1/CAP2 interrupt,
                                                          ;   Return from Interrupt
                    ;
                    ; This portion of the code takes the 1st capture and stores its
                    ; value in register pair IC2AH:IC2AL. When the 2nd capture
                    ; is taken, its value is stored in register pair IC2BH:IC2BL.
                    ; A 16-bit subtract is performed, with the final 24-bit result
                    ; being stored in IC2OF:IC2BH:IC2BL. This value will no longer
                    ; be correct after the next capture occurs (IC2BH:IC2BL will
                    ; change), so the main routine must utilize this value before
                    ; it changes.
                    ;
0043 8B16           CAPTURE          BCF     PIR,3        ; Clear Capture2 interrupt flag
0044 B803                            MOVLB   3            ; Select register Bank 3
0045 9826                            BTFSC   FLAG_REG,0   ; 1st or 2nd capture2?
0046 C04B                            GOTO    CAP2         ; It was the 2nd Capture
0047 5424           CAP1             MOVPF   CA2L,IC2AL   ; Move the captured value to
0048 5523                            MOVPF   CA2H,IC2AH   ;   temporary registers
0049 8026                            BSF     FLAG_REG,0   ; Have 1st capture2
004A 0005                            RETFIE               ; Return from Interrupt
                                                          ;
                     PAGE
004B 5422           CAP2             MOVPF   CA2L,IC2BL   ; Move the captured value to
004C 5521                            MOVPF   CA2H,IC2BH   ;   temporary registers
                                                          ;   (to prevent being overwritten)
                                                          ;
004D E061                            CALL    SUB16        ; Call routine which subtracts
```

```
                                                        ;   2 16-bit numbers.
004E 9926                           BTFSC   FLAG_REG,1  ; Underflow during SUB16?
004F 0725                           DECF    T3OFLCNTR,1 ; Since underflow, decrement the
                                                        ;   overflow counter.
0050 2926                           CLRF    FLAG_REG,1  ; Clear the flag bits for
                                                        ;   underflow and 2nd capture2
0051 6A25                           MOVFP   T3OFLCNTR,W ; Store the T3 input capture
0052 4A20                           MOVPF   W,IC2OF     ;   overflow value in IC2OF
0053 2825                           CLRF    T3OFLCNTR,0 ; Clear the Data register which
                                                        ;   counts how many times Timer 3
                                                        ;   overflows.
0054 0005                           RETFIE              ; Return from interrupt
                    ;
                    ; When Timer 3 has overflowed, the overflow counter only should
                    ; be incremented when the overflow occurs after a capture 1
                    ; but before the capture 2. The 4 possible cases when entering
                    ; the T3OVFL section of the PER_INT routine are as follows:
                    ;   Case 1: T3 overflow (only) and FLAG_REG.0 = 0 (waiting
                    ;           for Capture 1 to occur). Do Not increment counter
                    ;   Case 2: T3 overflow (only) and FLAG_REG.0 = 1 (waiting
                    ;           for Capture 2 to occur). Increment counter
                    ;   Case 3: T3 Overflow happened after Capture. Do Not
                    ;           increment overflow counter
                    ;   Case 4: T3 Overflow occurred before Capture 2 and FLAG_REG.0 = 1
                    ;           (waiting for Capture 2 to occur). Increment counter

                    ;
0055 8E16           T3OVFL          BCF     PIR,6       ; Clear Overflow interrupt flag
0056 9316                           BTFSS   PIR,3       ; Did the RB1/CAP2 pin also
                                                        ;   cause an interrupt?
0057 C05E                           GOTO    FR0         ; No, Check if between 1st
                                                        ;   and 2nd capture
0058 B803                           MOVLB   3           ; Bank 3
0059 280A                           CLRF    W,0         ; W = 0
005A 3115                           CPFSEQ  CA2H        ; if CA2H = 0, overflow happened
005B C05E                           GOTO    FR0         ;   first, must check FLAG_REG
                                                        ;   bit 0
005C B801                           MOVLB   1           ; Back to bank 1
005D C043                           GOTO    CAPTURE     ; Capture happened first, do NOT
                                                        ;   Increment overflow counter
                                                        ;   and do capture routine
005E 9826           FR0             BTFSC   FLAG_REG,0  ; Between Capture 1 and Capture 2?
005F 1525                           INCF    T3OFLCNTR,1 ; Yes, Inc. the overflow counter
0060 0005                           RETFIE              ; Return from overflow interrupt
                    ;
0061 6A24           SUB16           MOVFP   IC2AL,W     ; Do the 16-bit subtraction
0062 0522                           SUBWF   IC2BL,1     ;
0063 6A23                           MOVFP   IC2AH,W     ;
0064 0321                           SUBWFB  IC2BH,1     ;
0065 9004                           BTFSS   ALUSTA,0    ; Is the result pos. or neg. ?
0066 8126                           BSF     FLAG_REG,1  ; neg., Set the underflow flag
0067 0002                           RETURN              ; Return from the subroutine
                     PAGE
                    ;
                    ; Other Interrupt routines. (Not utilized in this example)
                    ;
0068 0005           EXT_INT RETFIE                      ; RA0/INT interrupt routine
                                                        ;   (NOT used in this program)
0069 0005           TMR0INT RETFIE                      ; TMR0 overflow interrupt routine
                                                        ;   (NOT used in this program)
006A 0005           T0INT   RETFIE                      ; RB1/RT interrupt routine
                                                        ;   (NOT used in this program)
                                                        ;
006B C028           SRESET  GOTO    START               ; If program became lost, goto
                                                        ;   START and reinitalize.
                    ;
                    00252 ;
                    00253 ; When the executed address is NOT in the program range, the
                    00254 ; 16-bit address should contain all 1’s (a CALL 0x1FFF). At
                    00255 ; this location you could branch to a routine to recover or
                    00256 ; shut down from the invalid program execution.
                    00257 ;
07FF                00258                ORG            END_OF_PROG_MEM ;
07FF C06B           00259                GOTO    SRESET     ; The program has lost it’s mind,
                    00260                                   ;   do a system reset
                    00261                END
MEMORY USAGE MAP (‘X’ = Used,  ‘-’ = Unused)

0000 : X-------X------- X-------X------- X-------XXXXXXXX XXXXXXXXXXXXXXXX
0040 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXX---- ----------------
07C0 : ---------------- ---------------- ---------------- ---------------X

All other memory blocks unused.

Program Memory Words Used:    74


Errors   :     0
Warnings :     0 reported,     0 suppressed
Messages :     0 reported,     0 suppressed
```

APPENDIX C: PULSE WIDTH MEASUREMENT EXAMPLE CODE

```
MPASM 01.40 Released             PW02.ASM   1-16-1997  15:16:15         PAGE  1


LOC  OBJECT CODE     LINE SOURCE TEXT
  VALUE

                    00001         LIST    P = 17C42, n = 66
                    00002 ;
                    00003 ;**********************************************************
                    00004 ;
                    00005 ; This is the basic outline for a program that can determine the
                    00006 ; Pulse Width of an input, via input capture. This routine uses an
                    00007 ; 8-bit register to count the times that timer3 overflowed. At the
                    00008 ; Max crystal frequency of 16 MHz, this gives an overflow time of
                    00009 ; (2**16)(256 + 1)(250 nS) > 4.21 sec or a frequency < 0.25 Hz. If
                    00010 ; measurement of longer time intervals is required, the overflow
                    00011 ; counter could be extended to 16 (or more) bits.
                    00012 ;
                    00013 ; Do the EQUate table
                    00014 ;
  00000020          00015 IC2OF        EQU     0x20       ; T3 overflow register
  00000021          00016 IC2BH        EQU     0x21       ; T3 ICA2 MSB register (2nd Cap)
  00000022          00017 IC2BL        EQU     0x22       ; T3 ICA2 LSB register
  00000023          00018 IC2AH        EQU     0x23       ; T3 ICB2 MSB register (1st Cap)
  00000024          00019 IC2AL        EQU     0x24       ; T3 ICB2 LSB register
  00000025          00020 T3OFLCNTR    EQU     0x25       ; Temperature T3 overflow register
                    00021 ;
  00000026          00022 FLAG_REG        EQU 0x26         ; Register that has the Flag bits
                    00023 ;
                    00024 ;    FLAG_REG  bit   7   6   5   4   3   2   1   0
                    00025 ;                    -   -   -   -   -   -  UFL CAP1
                    00026 ;    CAP1 = 0,  1st Capture
                    00027 ;         = 1,  2nd Capture
                    00028 ;
                    00029 ;    UFL = 0,  No Underflow
                    00030 ;        = 1,  Underflow during subtract
                    00031 ;
                    00032 ;       Program:          PW02.ASM
                    00033 ;       Revision Date:
                    00034 ;                         1-14-97   Compatibility with MPASMWIN 1.40
                    00035 ;
                    00036 ;*********************************************************************
                    00037 ;
                    00038 ;
  000007FF          00039 END_OF_PROG_MEM EQU     0x07FF
                    00040 ;
  00000004          00041 ALUSTA          EQU     0x04
  00000006          00042 CPUSTA          EQU     0x06
  00000007          00043 INTSTA          EQU     0x07
  0000000A          00044 W               EQU     0x0A
                    00045                                    ;
  00000012          00046 PORTB           EQU     0x12       ; Bank 0
                    00047                                    ;
  00000016          00048 PIR             EQU     0x16       ; Bank 1
  00000017          00049 PIE             EQU     0x17
                    00050                                    ;
  00000012          00051 TMR3L           EQU     0x12       ; Bank 2
  00000013          00052 TMR3H           EQU     0x13
  00000016          00053 T3PRL           EQU     0x16
  00000017          00054 T3PRH           EQU     0x17
```

```
                                                    ;
  00000014          CA2L         EQU     0x14       ; Bank 3
  00000015          CA2H         EQU     0x15
  00000016          TCON1        EQU     0x16
  00000017          TCON2        EQU     0x17
                     PAGE
                    ;
0000                             ORG     0x0000     ; Origin for the RESET vector
0000 C028                        GOTO    START      ; On reset, go to the start of
                                                    ;   the program
0008                             ORG     0x0008     ; Origin for the external RA0/INT
                                                    ;   interrupt vector
0008 C072                        GOTO    EXT_INT    ; Goto the ext. interrupt
                                                    ;   on RA0/INT routine
0010                             ORG     0x0010     ; Origin for the TMR0
                                                    ;   overflow interrupt vector
0010 C073                        GOTO    TMR0INT    ; Goto the TMR0 overflow interrupt
                                                    ;   routine
0018                             ORG    0x0018      ; Origin for the external
                                                    ;   RA1/T0CKI interrupt vector
0018 C074                        GOTO    T0CKI_INT  ; Goto the ext. interrupt on
                                                    ;   RA1/T0CKI routine
0020                             ORG      0x0020    ; Origin for the interrupt vector
                                                    ;   of any enabled peripheral
0020 C03E                        GOTO    PER_INT    ; Goto the interrupt from a
                                                    ;   peripheral routine
0028                             ORG     0x0028     ; Origin for the top of
                                                    ;   program memory
                     PAGE
0028 8406           START        BSF    CPUSTA,4    ; Disable ALL interrupts via the
                                                    ;   Global Interrupt Disable
                                                    ;   (GLINTD) bit.
                                                    ;
0029                MAIN                            ; Place Main program here
0029 B803                        MOVLB   3          ; Select register Bank 3
002A 2817                        CLRF    TCON2,0    ; Stop the timers, Single Capture
002B B070                        MOVLW   0x070      ; Initialize TCON1 so that
002C 0116                        MOVWF   TCON1      ;   T1 (8-bit), T2 (8-bit),
                                                    ;   and T3 run off the internal
                                                    ;   system clock. Capture2
                                                    ;   captures on the rising edge.
                    ;
                    ; Initialize Timer 3, load the timer with the number of cycles that
                    ; the device executes (from RESET) before the timer is turned on
                    ; Therefore the offset is required due to software overhead.
                    ;
002D B802                        MOVLB   2          ; Select register Bank 2
002E 280A                        CLRF    W,0        ; Clear the W register
002F 0126                        MOVWF   FLAG_REG   ; Initialize to 0
0030 0113                        MOVWF   TMR3H      ; Timer3 MSB = 0
0031 B013                        MOVLW   0x13       ; Timer3 LSB = Offset
0032 0112                        MOVWF   TMR3L      ;
                    ;
                    ; Load the Timer 3 period register with 0xFFFF, which will give an
                    ; interrupt on the overflow of Timer3
                    ;
0033 B0FF                        MOVLW   0xFF      ;
0034 0117                        MOVWF   T3PRH     ;
0035 0116                        MOVWF   T3PRL     ;
                    ;
                    ; the timer should be started and interrupts enabled.
                    ;
0036 B803                        MOVLB   3          ; Select register Bank 3
0037 8217                        BSF     TCON2,2    ; Turn on timer 3.
0038 8307                        BSF     INTSTA,3   ; Turn on Peripheral Interrupts
0039 B801                        MOVLB   1          ; Select register Bank 1
```

```
003A B048                        MOVLW   0x48       ; Enable Capture 2 and Timer3
003B 0117                        MOVWF   PIE        ; Interrupts (when GLINTD = 0)
                    ;
                    ; This is where you would do the things you wanted to do.
                    ; this example will only loop waiting for the interrupts.
                    ;
003C 8C06           WAIT         BCF     CPUSTA,4   ; Enable ALL interrupts
003D C03C                        GOTO    WAIT       ; Loop here waiting for a timer
                                                    ;   Interrupt
                     PAGE
                    ;
                    ; The interrupt routine for any peripheral interrupt, This routine
                    ; only deals with Timer3 (T3) interrupts.
                    ;
                    ; Time required to execute interrupt routine. Not including
                    ; interrupt latency (time to enter into the interrupt routine)
                    ;
                    ;       case1 - only T3 overflow             = 12 cycles
                    ;       case2 - 1st capture                  = 20 cycles
                    ;       case3 - 2nd capture                  = 34 cycles
                    ;       case4 - T3 overflow and 1st capture  = 32 cycles
                    ;       case5 - T3 overflow and 2nd capture  = 44 cycles
                    ;
                    ;
003E B801           PER_INT      MOVLB   1          ; Select register Bank 1
003F 9E16                        BTFSC   PIR,6      ; Did T3 overflow?
                                                    ;   If not skip next Instruction
0040 C05A                        GOTO    T3OVFL     ; Inc overflow cntr and clear flag
0041 9316           CK_CAP       BTFSS   PIR,3      ; Did the RB1/CAP2 pin cause an
                                                    ;   interrupt?
0042 0005                        RETFIE             ; No RB1/CAP2 interrupt,
                                                    ;   Return from Interrupt

                    ;
                    ; This portion of the code takes the 1st capture and stores its
                    ; value in register pair IC2AH:IC2AL. When the 2nd capture
                    ; is take, its value is stored in register pair IC2BH:IC2BL.
                    ; A 16-bit subtract is performed, with the final 24-bit result
                    ; being stored in IC2OF:IC2BH:IC2BL. This value will no longer
                    ; be correct after the next capture occurs (IC2BH:IC2BL will
                    ; change), so the main routine must utilize this value before
                    ; it changes.
                    ;
0043 B803           CAPTURE      MOVLB   3          ; Select register Bank 3
0044 9826                        BTFSC   FLAG_REG,0 ; Capture on rising or falling edge?
0045 C04B                        GOTO    FALLING    ; It was the 2nd Capture
0046 5424           RISING       MOVPF   CA2L,IC2AL ; Move the captured value to
0047 5523                        MOVPF   CA2H,IC2AH ;   temporary registers
0048 8026                        BSF     FLAG_REG,0 ; Set flag for 1st capture
0049 8E16                        BCF     TCON1,6    ; Change edge from rising
                                                    ;   to falling
004A C055                        GOTO    FALSE_C    ;** With the changing of the capture
                                                    ;** edge, we have a false capture
                                                    ;
                     PAGE
004B 5422           FALLING      MOVPF   CA2L,IC2BL ; Move the captured value to
004C 5521                        MOVPF   CA2H,IC2BH ;   temporary registers
                                                    ;   (to prevent being overwritten)
                                                    ;
004D E06B                        CALL    SUB16      ; Call routine which subtracts
                                                    ;   2 16-bit numbers.
004E 9926                        BTFSC   FLAG_REG,1 ; Underflow during SUB16?
004F 0725                        DECF    T3OFLCNTR,1; Since underflow, decrement the
                                                    ;   overflow counter.
0050 2926                        CLRF    FLAG_REG,1 ; Clear the flag bits for
                                                    ;   underflow and 2nd capture2
```

```
0051 6A25                        MOVFP   T3OFLCNTR,W; Store the T3 input capture
0052 4A20                        MOVPF   W,IC2OF    ;   overflow value in IC2OF
0053 2825                        CLRF    T3OFLCNTR,0; Clear the Data register which
                                                    ;   counts how many times Timer 3
                                                    ;   overflows.
0054 8616                        BSF     TCON1,6    ; Change edge from falling
                                                    ;   to rising
                    ;
                    ; Note when you change the edge of the capture, an additional capture
                    ; is generated. The capture register must be read before the capture
                    ; flag is cleared.
                    ;
0055 550A           FALSE_C      MOVPF   CA2H,W     ; Dummy read of Capture 2
0056 540A                        MOVPF   CA2L,W     ;
                    ;
0057 B801                        MOVLB   1          ; Select register Bank 1
0058 8B16                        BCF     PIR,3      ; Clear Capture2 Interrupt flag
0059 0005                        RETFIE             ; Return from interrupt, wait for
                                                    ;   T3 overflow or falling edge
                                                    ;   capture
                                                    ;
                     PAGE
                    ;
                    ; When Timer 3 has overflowed, the overflow counter only should
                    ; be incremented when the overflow occurs after a capture 1
                    ; but before the capture 2. The 6 possible cases when entering
                    ; the T3OVFL section of the PER_INT routine are as follows:
                    ;
                    ;   Case 1: T3 overflow (only) and FLAG_REG.0 = 0 (waiting
                    ;           for Capture 1 to occur). Do Not increment counter
                    ;   Case 2: T3 overflow (only) and FLAG_REG.0 = 1 (waiting
                    ;           for Capture 2 to occur). Increment counter
                    ;   Case 3: T3 Overflow, Then Capture1 happened. Do Not
                    ;           increment overflow counter
                    ;   Case 4: T3 Overflow, Then Capture 2 happened
                    ;           Increment counter
                    ;   Case 5: Capture1, Then T3 Overflow happened
                    ;           Increment counter
                    ;   Case 6: Capture2, Then T3 Overflow happened. Do Not
                    ;           Increment counter
                    ;
005A 8E16           T3OVFL       BCF     PIR,6      ; Clear Overflow interrupt flag
005B 9316                        BTFSS   PIR,3      ; Did the RB1/CAP2 pin also
                                                    ;   cause an interrupt?
005C C068                        GOTO    FR0        ; No, only overflow interrupt
005D B803                        MOVLB   3          ; Bank 3
005E 280A                        CLRF    W,0        ; W = 0
005F 9826                        BTFSC   FLAG_REG,0 ; T3 overflow with Capture 1
                                                    ;   or Capture 2?
0060 C064                        GOTO    OF_C1      ; Overflow with Capture 1
0061 3115           OF_C2        CPFSEQ  CA2H       ; if CA2H = 0, overflow happened
                                                    ;   first
0062 1525                        INCF    T3OFLCNTR,1; Increment counter
0063 C043                        GOTO    CAPTURE    ; Do capture routine
0064 3115           OF_C1        CPFSEQ  CA2H       ; if CA2H = 0, overflow happened
                                                    ;   first
0065 C043                        GOTO    CAPTURE    ; Capture happened first, do NOT
                                                    ;   Increment overflow counter
                                                    ;   and do capture routine
0066 1525                        INCF    T3OFLCNTR,1; Yes, Inc. the overflow counter
0067 C043                        GOTO    CAPTURE    ; Do capture routine
                    ;
                    ; Only increment overflow counter if between 1st and 2nd capture
                    ;
0068 9826           FR0          BTFSC   FLAG_REG,0 ; Between Capture 1 and Capture 2?
0069 1525                        INCF    T3OFLCNTR,1; Yes, Inc. the overflow counter
```

```
006A 0005                        RETFIE             ; Return from overflow interrupt
                    ;
006B 6A24           SUB16        MOVFP   IC2AL,W    ; Do the 16-bit subtraction
006C 0522                        SUBWF   IC2BL,1    ;
006D 6A23                        MOVFP   IC2AH,W    ;
006E 0321                        SUBWFB  IC2BH,1    ;
006F 9004                        BTFSS   ALUSTA,0   ; Is the result pos. or neg. ?
0070 8126                        BSF     FLAG_REG,1 ; neg., Set the underflow flag
0071 0002                           RETURN          ; Return from the subroutine
                     PAGE
                    ;
                    ; Other Interrupt routines. (Not utilized in this example)
                    ;
0072 0005           EXT_INT RETFIE                  ; RA0/INT interrupt routine
                                                    ;   (NOT used in this program)
0073 0005           TMR0INT RETFIE                  ; TMR0 overflow interrupt routine
                                                    ;   (NOT used in this program)
0074 0005           T0INT  RETFIE                   ; RA1/T0CKI interrupt routine
                                                    ;   (NOT used in this program)
                                                    ;
0075 C028           SRESET  GOTO    START           ; If program became lost, goto
                                                    ;   START and reinitalize.
                    ;
                    ;
                    ; When the executed address is NOT in the program range, the
                    ; 16-bit address should contain all 1’s (CALL 0x1FFF). At
                    ; this location you could branch to a routine to recover or
                    ; shut down from the invalid program execution.
                    ;
07FF                         ORG     END_OF_PROG_MEM;
07FF C075                    GOTO    SRESET         ; The program has lost it’s mind,
                                                    ;   do a system reset
                                    END
MEMORY USAGE MAP (‘X’ = Used,  ‘-’ = Unused)

0000 : X-------X------- X-------X------- X-------XXXXXXXX XXXXXXXXXXXXXXXX
0040 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXX----------
07C0 : ---------------- ---------------- ---------------- ---------------X

All other memory blocks unused.

Program Memory Words Used:    84


Errors   :     0
Warnings :     0 reported,     0 suppressed
Messages :     0 reported,     0 suppressed
```

MICROCHIP

WORLDWIDE SALES AND SERVICE

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200 Fax: 480-792-7277
Technical Support: 480-792-7627
Web Address: http://www.microchip.com

Rocky Mountain
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7966 Fax: 480-792-7456

Atlanta
500 Sugar Mill Road, Suite 200B
Atlanta, GA 30350
Tel: 770-640-0034 Fax: 770-640-0307

Boston
2 Lan Drive, Suite 120
Westford, MA 01886
Tel: 978-692-3848 Fax: 978-692-3821

Chicago
333 Pierce Road, Suite 180
Itasca, IL 60143
Tel: 630-285-0071 Fax: 630-285-0075

Dallas
4570 Westgrove Drive, Suite 160
Addison, TX 75001
Tel: 972-818-7423 Fax: 972-818-2924

Detroit
Tri-Atria Office Building
32255 Northwestern Highway, Suite 190
Farmington Hills, MI 48334
Tel: 248-538-2250 Fax: 248-538-2260

Kokomo
2767 S. Albright Road
Kokomo, Indiana 46902
Tel: 765-864-8360 Fax: 765-864-8387

Los Angeles
18201 Von Karman, Suite 1090
Irvine, CA 92612
Tel: 949-263-1888 Fax: 949-263-1338

New York
150 Motor Parkway, Suite 202
Hauppauge, NY 11788
Tel: 631-273-5305 Fax: 631-273-5335

San Jose
Microchip Technology Inc.
2107 North First Street, Suite 590
San Jose, CA 95131
Tel: 408-436-7950 Fax: 408-436-7955

Toronto
6285 Northam Drive, Suite 108
Mississauga, Ontario L4V 1X5, Canada
Tel: 905-673-0699 Fax: 905-673-6509

ASIA/PACIFIC

Australia
Microchip Technology Australia Pty Ltd
Suite 22, 41 Rawson Street
Epping 2121, NSW
Australia
Tel: 61-2-9868-6733 Fax: 61-2-9868-6755

China - Beijing
Microchip Technology Consulting (Shanghai) Co., Ltd., Beijing Liaison Office
Unit 915
Bei Hai Wan Tai Bldg.
No. 6 Chaoyangmen Beidajie
Beijing, 100027, No. China
Tel: 86-10-85282100 Fax: 86-10-85282104

China - Chengdu
Microchip Technology Consulting (Shanghai) Co., Ltd., Chengdu Liaison Office
Rm. 2401, 24th Floor,
Ming Xing Financial Tower
No. 88 TIDU Street
Chengdu 610016, China
Tel: 86-28-6766200 Fax: 86-28-6766599

China - Fuzhou
Microchip Technology Consulting (Shanghai) Co., Ltd., Fuzhou Liaison Office
Unit 28F, World Trade Plaza
No. 71 Wusi Road
Fuzhou 350001, China
Tel: 86-591-7503506 Fax: 86-591-7503521

China - Shanghai
Microchip Technology Consulting (Shanghai) Co., Ltd.
Room 701, Bldg. B
Far East International Plaza
No. 317 Xian Xia Road
Shanghai, 200051
Tel: 86-21-6275-5700 Fax: 86-21-6275-5060

China - Shenzhen
Microchip Technology Consulting (Shanghai) Co., Ltd., Shenzhen Liaison Office
Rm. 1315, 13/F, Shenzhen Kerry Centre,
Renminnan Lu
Shenzhen 518001, China
Tel: 86-755-2350361 Fax: 86-755-2366086

Hong Kong
Microchip Technology Hongkong Ltd.
Unit 901-6, Tower 2, Metroplaza
223 Hing Fong Road
Kwai Fong, N.T., Hong Kong
Tel: 852-2401-1200 Fax: 852-2401-3431

India
Microchip Technology Inc.
India Liaison Office
Divyasree Chambers
1 Floor, Wing A (A3/A4)
No. 11, O'Shaugnessey Road
Bangalore, 560 025, India
Tel: 91-80-2290061 Fax: 91-80-2290062

Japan
Microchip Technology Japan K.K.
Benex S-1 6F
3-18-20, Shinyokohama
Kohoku-Ku, Yokohama-shi
Kanagawa, 222-0033, Japan
Tel: 81-45-471- 6166 Fax: 81-45-471-6122

Korea
Microchip Technology Korea
168-1, Youngbo Bldg. 3 Floor
Samsung-Dong, Kangnam-Ku
Seoul, Korea 135-882
Tel: 82-2-554-7200 Fax: 82-2-558-5934

Singapore
Microchip Technology Singapore Pte Ltd.
200 Middle Road
#07-02 Prime Centre
Singapore, 188980
Tel: 65-6334-8870 Fax: 65-6334-8850

Taiwan
Microchip Technology Taiwan
11F-3, No. 207
Tung Hua North Road
Taipei, 105, Taiwan
Tel: 886-2-2717-7175 Fax: 886-2-2545-0139

EUROPE

Denmark
Microchip Technology Nordic ApS
Regus Business Centre
Lautrup hoj 1-3
Ballerup DK-2750 Denmark
Tel: 45 4420 9895 Fax: 45 4420 9910

France
Microchip Technology SARL
Parc d'Activite du Moulin de Massy
43 Rue du Saule Trapu
Batiment A - ler Etage
91300 Massy, France
Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79

Germany
Microchip Technology GmbH
Gustav-Heinemann Ring 125
D-81739 Munich, Germany
Tel: 49-89-627-144 0 Fax: 49-89-627-144-44

Italy
Microchip Technology SRL
Centro Direzionale Colleoni
Palazzo Taurus 1 V. Le Colleoni 1
20041 Agrate Brianza
Milan, Italy
Tel: 39-039-65791-1 Fax: 39-039-6899883

United Kingdom
Arizona Microchip Technology Ltd.
505 Eskdale Road
Winnersh Triangle
Wokingham
Berkshire, England RG41 5TU
Tel: 44 118 921 5869 Fax: 44-118 921-5820

03/01/02