Using the CCP Modules

This application note discusses the operation of a Capture Compare and PWM (CCP) module, and the interaction of multiple CCP modules with the timer resources.

The Capture Compare and PWM (CCP) module is software programmable to operate in one of three modes:

1. A Capture input
2. A Compare output
3. A Pulse Width Modulation (PWM) output

For the CCP module to function, Timer resources must be used in conjunction with the CCP module. The desired CCP mode of operation determines which timer resources are required. Table 1 shows the CCP mode with the corresponding timer resource required. Both the Capture and Compare modes require that Timer 1 be operating in timer mode or synchronized counter mode.

Note: Capture and Compare modes may not operate if Timer1 is operated in asynchronous counter mode.

TABLE 1: CCP MODE - TIMER RESOURCE

CCP Mode	Timer Resource
Capture	Timer 1
Compare	Timer 1
PWM	Timer 2

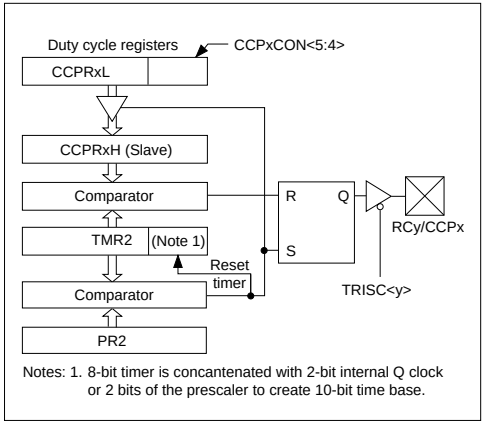
CCP OPERATION

The following three sections discuss the operation of the CCP module in each of its modes of operation. There is a simple example program for each mode of operation. The software example for the capture mode, also uses a second CCP module in compare mode to generate the signal to capture.

PWM Mode

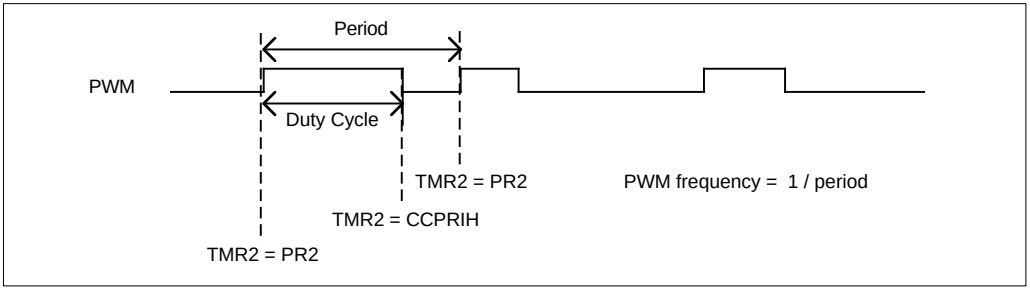
A Pulse Width Modulation output (shown in Figure 1) is a signal that has a timebase (period) and a time that the output stays high (duty cycle). The period is the duration after which the PWM rising edge repeats itself. The resolution of the PWM output is the granularity with which the duty cycle can be varied. The frequency of a PWM is simply the inverse of the period (1 / period).

FIGURE 2: PWM MODE BLOCK DIAGRAM



Notes: 1. 8-bit timer is concantenated with 2-bit internal Q clock or 2 bits of the prescaler to create 10-bit time base.

FIGURE 1: PWM OUTPUT



Using the CCP Modules

Each CCP module can support one Pulse Width Modulation (PWM) output signal, with minimal software overhead. This PWM signal can attain a resolution of up to 10-bits, from the 8-bit Timer 2 module. This gives 1024 steps of variance from an 8-bit overflow counter. This gives a maximum accuracy of Tosc (50 ns, when the device is operated at 20 MHz). Figure 2 shows a block diagram of the CCP module in PWM mode. When the Timer 2 overflows (timer = Period Register), the value in the duty cycle registers (CCPRxL:CCPRxCON<5:4>) is latched into the 10-bit slave latch. A new duty cycle value can be loaded into the duty cycle register(s) at any time, but is only latched into the slave latch when Timer 2 = Timer 2 Period Register (PR2).

The period of Timer 2 (and PWM) is determined by the frequency of the device, the Timer 2 prescaler value (1, 4 or 16), and the Timer 2 Period Register. Equation 1 shows the calculation of the PWM period, duty cycle, and the minimum and maximum frequncies.

EQUATION 1: PWM PERIOD, DUTY CYCLE, AND FREQUNCIES

PWM Period = $[(PR2) + 1) \bullet 4\ T_{OSC} \bullet$ (Timer 2 prescale value)

PWM Duty Cycle = $[CCPRxL:CCPRxCON<5:4>] \bullet 4\ T_{OSC} \bullet$ (Timer 2 prescale value)

PWM maximum frequency

(High Resolution mode) = $4 / (PR2 \bullet T_{CY})$

(Low Resolution mode) = $1 / (PR2 \bullet T_{CY})$

PWM minimum frequency

(High Resolution mode) = $4 / (PR2 \bullet 16 \bullet T_{CY})$

(Low Resolution mode) = $1 / (PR2 \bullet 16 \bullet T_{CY})$

Table 2 shows the minimum and maximum PWM frequency for different device frequencies. The Timer2 prescaler will be selected to give either the minimum or maximum frequencies as shown.

Appendix A is a program which generates up to a 10-bit PWM output. The PWM period and duty cycle are updated after the overflow of Timer1. Upon the overflow of Timer1, ports A, B and D are read. The 10-bit duty cycle is specified by the value on PORTB:PORTA<1:0>, while the period is specified by the value on PORTD. By setting the conditional assemble flag `PICMaster` to `TRUE`, these values are read from internal registers which are dummy registers for the ports (DUMMY_Px). This allows the software to be verified without the use of hardware and external stimulus.

Since the PWM duty cycle is double buffered, the duty cycle registers are only loaded when there is sufficient time to complete the update the 10-bit value before the Timer2 = PR2 match occurs. After the duty cycle has been updated and the Timer2 = PR2 match has occurred, the period (stored in the PR2 register) is updated. The operation of the CCP module in PWM mode is similar to the PIC17C42's PWM. Additional concepts of PWM operation can be found in Application Notes AN564 and AN539.

TABLE 2: PWM FREQUENCY FOR DIFFERENT DEVICE FREQUENCIES

	20 MHz		10 MHz		2 MHz		
PWM Resolution	Min	Max	Min	Max	Min	Max	Units
10-bit	1.22	19.53	0.613	9.77	0.123	1.96	KHz
9-Bit	1.22	39.06	0.613	9.77	0.123	3.92	KHz
10-bit	1.22	19.53	0.613	9.77	0.123	1.96	KHz
9-Bit	1.22	39.06	0.613	9.77	0.123	3.92	KHz
10-bit	1.22	19.53	0.613	9.77	0.123	1.96	KHz
9-Bit	1.22	39.06	0.613	9.77	0.123	3.92	KHz
10-bit	1.22	19.53	0.613	9.77	0.123	1.96	KHz
9-Bit	1.22	39.06	0.613	9.77	0.123	3.92	KHz

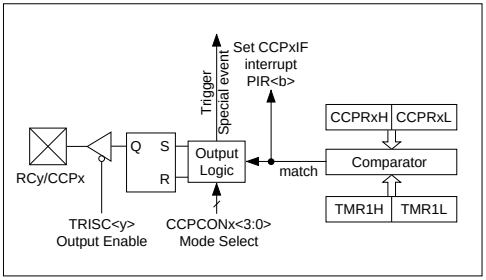
Compare Mode

In compare mode, the 16-bit value of Timer1 is compared to the CCPRxH:CCPRxL registers. When these registers match, the S/W configured event occurs on the CCPx pin. The events that can be S/W selected are:

- Clear CCPx pin on match
- Set CCPx pin on match
- Generate S/W interrupt (CCPx pin unchanged)
- Trigger special event (CCPx pin unchanged)
  - CCP1 clears Timer1
  - CCP2 clears Timer1 and sets the A/D's GO bit

The CCPxM<3:0> control bits, in register CCPxCON, configures the operation of the CCP module. The compare function must have the data direction of the CCPx pin configured as an output, if the compare event is to control the state of the CCPx pin.

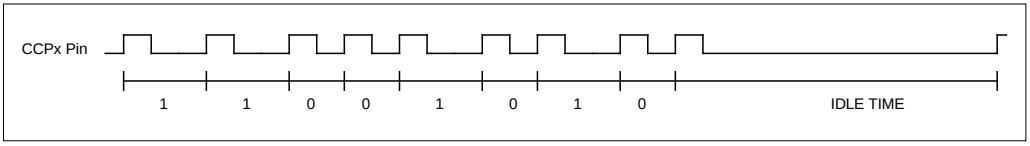
FIGURE 3: COMPARE MODE BLOCK DIAGRAM



When the CCP module is in the OFF state (CCPxM<3:0> = 0h), the CCPx output latch is forced to a low level, though the level on the CCPx pin will be determined by the value in the data latch of the port. Figure 3 shows the block diagram of the CCP module in Compare mode.

Appendix B is a program which uses the CCP module to transmit a pulse train dependent on the data byte. Timer1 is used as a free running timer, with each "new" compare value being an offset added to the present CCP compare latch value. The data is transmitted every 600 µs. Each data bit has a sync pulse (High level) of 8.8 µs. Then the data is transmitted as a low pulse. The time duration of the low pulse determines the value of the data bit. A '0' bit is low for 18.8 µs while a '1' bit is low for 37.6 µs. After the last data bit has been transmitted, another sync pulse is transmitted and the output remains low (idle time) until the 600 µs data period has completed. An example of the pulse train for the a data byte of CAh is shown in Figure 4, and has an idle time of 224 µs. These pulse times are based off the device operational frequency. The program header file, COMP.H, calculates the values to loaded into the compare registers from the specified Device_freq. The data to be transmitted is read from PORTB, during the idle time. By setting the conditional assemble flag `PICMaster` to TRUE, these values are read from internal registers which are dummy registers for the ports (DUMMY_Px). This allows the software to be verified without the use of hardware and external stimulus.

FIGURE 4: TRANSMIT PULSE TRAIN (DATA = 0X0CA)



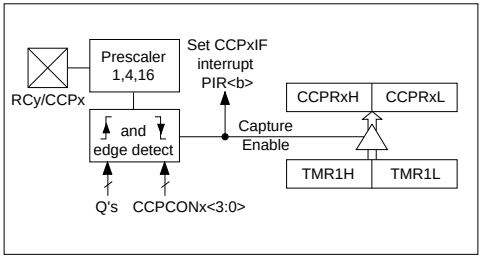
Capture Mode

In capture mode, the 16-bit value of Timer1 is latched into the CCPRxH:CCPRxL registers, when the S/W configured event occurs on the CCPx pin. The events that can cause a capture are:

- Every falling edge
- Every rising edge
- Every 4th rising edge
- Every 16th rising edge

The CCPxM<3:0> control bits, in register CCPxCON, configures the operation of the CCP module. The capture function works regardless of the data direction of the CCPx pin (input or output). With the CCPx pin is configured as an output, a write to the CCPx pin (in PORTC) will cause a capture when the capture requirement is met.

FIGURE 5: CAPTURE MODE BLOCK DIAGRAM

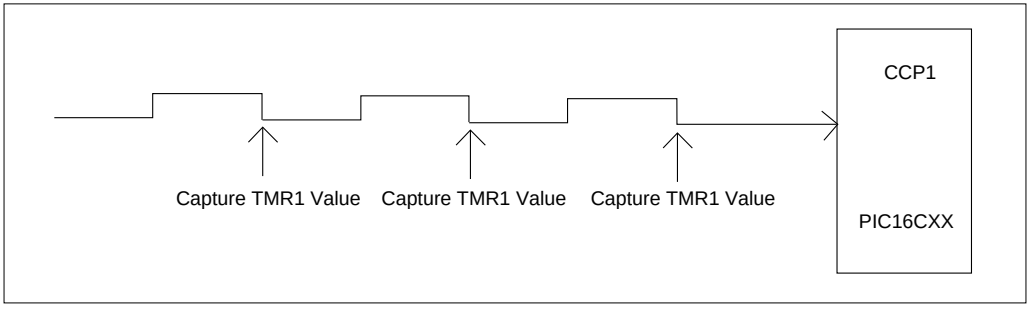


The changing of the capture mode, via the CCPxM<3:0> bits, may cause the CCPxIF bit to be set. This "false" interrupt should be cleared (ignored) after changing between capture modes. The CCP prescaler is only cleared by configuring the CCP module into the OFF state (CCPxM<3:0> = 0h). Figure 5 shows the block diagram of the CCP module in Capture mode. The utilization of the CCP module in capture mode is similar to the PIC17C42's capture. Additional concepts of capture operation can be found in Application Note AN545.

Appendix C is a program which implements a 16-bit capture from a free running timer (TMR1). The capture event is configured as each rising edge. The 16-bit capture value is the "new" 16-bit capture value minus the "old" 16-bit capture value. If the time between captures is greater than 2^{16} Timer1 increments, an invalid result will occur. This invalid result is not indicated by the software. After the capture period result is calculated, the "new" capture value is loaded into the "old" register.

The waveform that is captured is generated from a second CCP module in compare mode. The value that is loaded in to the CCPR2H:CCPR2L is read from the PORTB and PORTD registers. By setting the conditional assemble flag `PICMaster` to `TRUE`, these values are read from internal registers which are dummy registers for the ports (DUMMY_Px). This allows the software to be verified without the use of hardware and external stimulus. Figure 6 shows an input into the CCPx pin, and the capture measurement points.

FIGURE 6: EXAMPLE CAPTURE WAVE FORM



INTERACTION OF CCP MODULES

Due to the modularity of the PIC16CXX peripherals, future devices with two or more CCP modules on a device are possible. Each CCP module operates independently from the others, though their interaction with the timer resources must be taken into account.

When two or more CCP modules exist on a device, there can be an interaction between the CCP modules. This interaction is shown in Table 3. These interactions do NOT include any interaction (S/W) caused by the main program nor the interrupt service routines of the CCP sources.

Interaction of Two Capture Modes

When two CCP modules are in a Capture mode, Timer1 is the timebase for both captures. This means that they will have the same capture resolution, as determined by the TMR1 prescaler and frequency of the timer/counter clock. This clock can come from an external source (on the RC0/T1OSO/T1CKI pin), but must be synchronized to the device.

Interaction of One Capture Mode and One Compare Mode

When one CCP module is in a Capture mode and a second CCP module is in Compare mode, Timer1 is the timebase for both the captures and the compare. This means that the capture and the compare will have the same resolution, as determined by the TMR1 prescaler and frequency of the timer/counter clock. This clock can come from an external source (on the RC0/T1OSO/T1CKI pin), but must be synchronized to the processor clock. Also, care must be taken in that the compare can be configured to clear TMR1 (when in special Trigger mode). Care must be taken in system design to ensure that this clearing of the TMR1 does not have any negative impact on the capture function.

Interaction of Two Compare Modes

When two CCP modules are in a Compare mode, Timer1 is the timebase for both compares. This means that they will have the same compare resolution, as determined by the TMR1 prescaler and frequency of the timer/counter clock. This clock can come from an external source (on the RC0/T1OSO/T1CKI pin), but must be synchronized to the processor clock. Since the compare modules can be configured to clear TMR1 (when in special Trigger mode), care must be taken in system design to ensure that this clearing of the TMR1 does not have any negative impact on the compare function. If both compares are configured with a special trigger, which clears the TMR1, then the compare register that is closest to (but greater than) the TMR1 value is the compare value that will reset TMR1. Example 1 shows a possible case.

TABLE 3: INTERACTION OF TWO CCP MODULES

CCPx Mode	CCPy Mode	Interaction
Capture	Capture	Same TMR1 timebase.
Capture	Compare	The compare could be configured for trigger special event, which clears TMR1.
Compare	Compare	The compare(s) could be configured for trigger special event, which clears TMR1.
PWM	PWM	The PWMs will have the same frequency, and update rate (TMR2 interrupt).
PWM	Capture	None
PWM	Compare	None

EXAMPLE 1:

ACTION	TIMER 1 STATE	COMMENT
CCPR1H:CCPR1L = 0x0465	0x????	
CCP1CON = 0x?B	0x????	CCP1 in Compare - Special
:		Trigger Mode
:	0x0232	
:		
CCPR2H:CCPR2L = 0x0165	0x0333	
CCP2CON = 0x?B	0x0334	CCP2 in Compare - Special
:		Trigger Mode
	0x0465	CCP1 resets TMR1 and CCP1 -
	0x0000	Special Trigger function occurs
:		
	0x0165	CCP2 resets TMR1 and CCP2 -
	0x0000	Special Trigger function occurs
:		
	0x0165	CCP2 resets TMR1 and CCP2 -
	0x0000	Special Trigger function occurs
:		

Interaction of Two PWM Modes

When two CCP modules are in a PWM mode, Timer2 is the timebase for both PWM outputs. This means that they will have the same PWM frequency and update rates, as determined by the TMR2 prescaler and frequency of the device. The resolution of the two PWMs may be different, since each CCP module has its own CCPxX:CCPxY bits for high resolution mode. These bits are found in the CCPxCON<5:4> register.

CONCLUSION

The Capture / Compare / PWM modules offer enormous flexibility in the use of the device timer resources. As with all resources, care must be taken to ensure that no adverse system complications can occur with the interaction between multiple CCP modules. The programs for simple operation of the various CCP modes should be a good foundation for modifications to suite your particular needs.

Written by: Mark Palmer - Sr. Application Engineer
Logic Products Division

```
MPASM 01.01 Released    PWM_1.ASM   7-13-1994  14:26:3          PAGE  1


LOC  OBJECT CODE     LINE SOURCE TEXT

                     0001         LIST    P = 16C74,  F = INHX8M, n = 66
                     0002 ;
                     0003 ;*****************************************************************************
                     0004 ;
                     0005 ; This program outputs a PWM signal on the CCP1 pin. The duty cycle and period
                     0006 ; of the PWM is read every time TMR1 overflows.
                     0007 ;    PERIOD     = PORTB
                     0008 ;    DUTY CYCLE = PORTD and PORTE<1:0>
                     0009 ;
                     0010 ; The prescaler of TMR2 is selected by the state of PORTA<1:0> after reset
                     0011 ;     RA1:RA0    Prescaler multiplies Tcyc by
                     0012 ;       0 0       1
                     0013 ;       0 1       4
                     0014 ;       1 x      16
                     0015 ;
                     0016 ;
                     0017 ;       Program = PWM_1.ASM
                     0018 ;       Revision Date:   7-13-94
                     0019 ;
                     0020 ;*****************************************************************************
                     0021 ;
                     0022 ;
                     0023 ; HARDWARE SETUP
                     0024 ;    PORTA<1:0>   -  Prescaler to TMR2, read only after reset
                     0025 ;    PORTB        -  Period of PWM
                     0026 ;    PORTD        -  Duty Cycle high of PWM (8-bits)
                     0027 ;    PORTE<1:0>   -  Duty Cycle low of PWM (2-bits)
                     0028 ;
                     0029 ;
                     0030             INCLUDE <C74_reg.h>
                     0247
                     0030
                     0031             INCLUDE <PWM.h>
                     0047
                     0031
                     0032 ;
0001                 0033 PICMaster       EQU     TRUE            ; A Debugging Flag
0001                 0034 Debug           EQU     TRUE            ; A Debugging Flag
0001                 0035 Debug_PU        EQU     TRUE            ; A Debugging Flag
                     0036 ;
                     0037 ;
                     0038 ; Reset address. Determine type of RESET
```

```
                        0039 ;
                        0040         org     RESET_V                 ; RESET vector location
0000 1683               0041 RESET       BSF     STATUS, RP0         ; Bank 1
0001 188E               0042             BTFSC   PCON, POR           ; Power-up reset?
0002 2832               0043             GOTO    START               ; YES
0003 285C               0044             GOTO    OTHER_RESET         ; NO, a WDT or MCLR reset
                        0045 ;
                        0046 ; This is the Periperal Interrupt routine. Need to determine the type
                        0047 ; of interrupt that occurred. The following interrupts are enabled:
                        0048 ;   1.  CCP Capture Occured
                        0049 ;
                        0051         org     ISR_V                   ; Interrupt vector location
                        0052 PER_INT_V
0004 1283               0053             BCF     STATUS, RP0         ; Bank 0
0005 180C               0054             BTFSC   PIR1, TMR1IF        ; TMR1 Overflow Interrupt occured?
0006 280D               0055             GOTO    T1OVFL              ; YES, Service the TMR1 Interrupt
                        0056 ERROR1                                  ; NO, Error Condition - Unknown Interrupt
0007 1505               0057             BSF     PORTA, 2            ; Toggle a PORT pin
0008 1105               0058             BCF     PORTA, 2
0009 2807               0059             GOTO    ERROR1
                        0060 ;
                        0061 ERROR2                                  ; NO, Error Condition - Unknown Interrupt
000A 1585               0062             BSF     PORTA, 3            ; Toggle a PORT pin
000B 1185               0063             BCF     PORTA, 3
000C 280A               0064             GOTO    ERROR2
                        0065 ;
                        0066 T1OVFL
000D 100C               0067             BCF     PIR1, TMR1IF        ; Clear T1 Overflow Interrupt Flag
                        0068         if (PICMaster )
000E 0853               0069             MOVF    DUMMY_PD, W         ;
                        0070         else
                        0071             MOVF    PORTD, W            ;
                        0072         endif
000F 00D5               0073             MOVWF   DC_HI               ;
                        0074         if (PICMaster )
0010 0854               0075             MOVF    DUMMY_PE, W         ;
                        0076         else
                        0077             MOVF    PORTE, W            ;
                        0078         endif
0011 00D6               0079             MOVWF   DC_LO               ;
                        0080         if (PICMaster )
0012 0851               0081             MOVF    DUMMY_PB, W         ;
                        0082         else
                        0083             MOVF    PORTB, W            ;
                        0084         endif
0013 1683               0085             BSF     STATUS, RP0         ; Bank 1
0014 00A0               0086             MOVWF   T2_PERIOD           ;
0015 1283               0087             BCF     STATUS, RP0         ; Bank 0
```

```
                        0088 ;
                        0089 WAIT_DC
0016 0811               0090            MOVF    TMR2, W         ; Read present TMR2 register value
0017 0212               0091            SUBWF   PR2, W          ; How close is the timer to rolling over
0018 390F               0092            ANDLW   0x0F            ; Does this make it zero?
0019 1903               0093            BTFSC   STATUS, Z       ; If Z is set, near rollover
001A 2816               0094            GOTO    WAIT_DC         ; loop until rolled over
001B 0855               0095            MOVF    DC_HI, W        ; else losd the duty cycle values
001C 0095               0096            MOVWF   CCPR1L          ; Load DC high
001D 300F               0097            MOVLW   0x0F            ;
001E 0597               0098            ANDWF   CCP1CON, F      ; Set the DC low bits
001F 18D6               0099            BTFSC   DC_LO, 1        ;
0020 1697               0100            BSF     CCP1CON, CCP1X  ;
0021 1856               0101            BTFSC   DC_LO, 0        ;
0022 1617               0102            BSF     CCP1CON, CCP1Y  ;
0023 108C               0103            BCF     PIR1, TMR2IF    ; Clear the TRM2 = PR2 flag
                        0104 ;
                        0105 WAIT_PR
0024 1C8C               0106            BTFSS   PIR1, TMR2IF    ; LOOP waiting for TRM2 = PR2
0025 2824               0107            GOTO    WAIT_PR         ; Need to wait until TMR2 = PR2 so that
                        0108                                    ;    Duty Cycle is latched
0026 1683               0109            BSF     STATUS, RP0     ; Bank 1
0027 300F               0110            MOVLW   0x0F            ; Load TMR2 period with minimum value Fh
0028 0092               0111            MOVWF   PR2             ;
0029 30F0               0112            MOVLW   0xF0            ;
002A 0520               0113            ANDWF   T2_PERIOD, W    ; Determine if period needs to be greater
002B 1903               0114            BTFSC   STATUS, Z       ;
002C 2830               0115            GOTO    NO_OFFSET       ; NO, Period is the minimum
                        0116 PR_OFFSET
002D 300F               0117            MOVLW   0x0F            ; Yes, calculate additional offset
002E 0220               0118            SUBWF   T2_PERIOD, W    ;
002F 0792               0119            ADDWF   PR2, F          ; ADD Period offset
                        0120 ;
                        0121 NO_OFFSET
0030 1283               0122            BCF     STATUS, RP0     ; Bank 0
0031 0009               0123            RETFIE                  ; Return / Enable Global Interrupts
                        0124 ;
                        0126 ;
                        0127 ;*****************************************************************************
                        0128 ;*****      Start program here, Power-On Reset occurred.
                        0129 ;*****************************************************************************
                        0130 ;
                        0131 START                              ; POWER_ON Reset (Beginning of program)
0032 1283               0132            BCF     STATUS, RP0     ; Bank 0
0033 018F               0133            CLRF    TMR1H           ;
0034 018E               0134            CLRF    TMR1L           ;
                        0135 ;
                        0136 MCLR_RESET                         ; A Master Clear Reset
```

```
0035 0183           0137            CLRF    STATUS          ; Do initialization (Bank 0)
0036 018B           0138            CLRF    INTCON
0037 018C           0139            CLRF    PIR1
0038 1683           0140            BSF     STATUS, RP0     ; Bank 1
0039 3080           0141            MOVLW   0x80            ;
003A 0081           0142            MOVWF   OPTION_R        ;
003B 018C           0143            CLRF    PIE1            ; Disable all peripheral interrupts
003C 30FF           0144            MOVLW   0xFF            ;
003D 009F           0145            MOVWF   ADCON1          ; Port A is Digital.
                    0146 ;
                    0147 ;
003E 1283           0148            BCF     STATUS, RP0     ; Bank 0
003F 0185           0149            CLRF    PORTA           ; ALL PORT output should output Low.
0040 0186           0150            CLRF    PORTB
0041 0187           0151            CLRF    PORTC
0042 0188           0152            CLRF    PORTD
0043 0189           0153            CLRF    PORTE
                    0154 ;
0044 1683           0155            BSF     STATUS, RP0     ; Select Bank 1
0045 30FF           0156            MOVLW   0xFF            ;
0046 0085           0157            MOVWF   TRISA           ; RA5 -  0 inputs
0047 0086           0158            MOVWF   TRISB           ; RB7 - 0 inputs
0048 0187           0159            CLRF    TRISC           ; RC Port are outputs
0049 0088           0160            MOVWF   TRISD           ; RD Port are inputs
004A 0089           0161            MOVWF   TRISE           ; RE Port are inputs
004B 0092           0162            MOVWF   PR2             ; Default PWM period
004C 140C           0163            BSF     PIE1, TMR1IE    ; Enable TMR1 Interrupt
004D 1283           0164            BCF     STATUS, RP0     ; Select Bank 0
                    0165 ;
004E 300C           0166            MOVLW   0X0C            ; CCP module is in
004F 0097           0167            MOVWF   CCP1CON         ; PWM output mode
                    0168 ;
                    0169 ; Initialize the Special Function Registers (SFR) interrupts
                    0170 ;
0050 018C           0171            CLRF    PIR1            ;
0051 0190           0172            CLRF    T1CON           ;
0052 0192           0173            CLRF    T2CON           ;
                    0174        if (PICMaster )
0053 1850           0175            BTFSC   DUMMY_PA, 0     ;
                    0176        else
                    0177            BTFSC   PORTA, 0        ;
                    0178        endif
0054 1412           0179            BSF     T2CON, 0        ;
                    0180 ;
                    0181        if (PICMaster )
0055 18D0           0182            BTFSC   DUMMY_PA, 1     ;
                    0183        else
                    0184            BTFSC   PORTA, 1        ;
```

```
                    0185         endif
0056 1492           0186             BSF     T2CON, 1        ;
                    0187 ;
0057 170B           0188             BSF     INTCON, PEIE    ; Enable Peripheral Interrupts
0058 178B           0189             BSF     INTCON, GIE     ; Enable all Interrupts
0059 1410           0190             BSF     T1CON, TMR1ON   ; Turn Timer 1 ON
005A 1512           0191             BSF     T2CON, TMR2ON   ; Turn Timer 2 ON
                    0192 ;
005B 285B           0193 lzz         goto    lzz             ; Loop waiting for TMR1 interrupt
                    0194 ;
                    0195 ; Here is where you do things depending on the type of RESET (Not a Power-On Reset).
                    0196 ;
005C 1E03           0197 OTHER_RESET   BTFSS  STATUS, TO     ; WDT Time-out?
005D 2807           0198 WDT_TIMEOUT   GOTO   ERROR1        ; YES, This is error condition
                    0199         if ( Debug_PU )
005E 2832           0200             goto   START           ; MCLR reset, Goto START
                    0201         else
                    0202             GOTO   MCLR_RESET      ; MCLR reset, Goto MCLR_RESET
                    0203         endif
                    0204 ;
                    0205         if (Debug )
005F 0000           0206 END_START     NOP                  ; END lable for debug
                    0207         endif
                    0208 ;
                    0209 ;
                    0210     org     PMEM_END               ; End of Program Memory
07FF 2807           0211             GOTO    ERROR1         ; If you get here your program was lost
                    0212
                    0213     end
                    0214
                    0215
```

```
MEMORY USAGE MAP ('X' = Used,  '-' = Unused)

0000 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX
0040 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX ________ ________

0780 : ________ ________ ________ ________
07C0 : ________ ________ ________ _______X

All other memory blocks unused.


Errors   :    0
Warnings :   17
```

APPENDIX A2: PWM.H

```
        nolist
;*************************************************************************
;
;  This is the custom Header File for the real time clock application note
;       PROGRAM:       CLOCK.H
;       Revision:      7-13-94
;
;*************************************************************************
; This is used for the ASSEMBLER to recalculate certain frequency
; dependant variables. The value of Dev_Freq must be changed to
; reflect the frequency that the device actually operates at.
;
Dev_Freq               EQU            D'10000000'                  ; Device Frequency is 4 MHz
PULSE_TIME      EQU     (( Dev_Freq / D'4000' ) * D'188' / D'10000' )
;

DB_HI_BYTE             EQU            (HIGH ((( Dev_Freq / 4 ) * 1 / D'1000' ) / 3 ) ) + 1
LCD_INIT_DELAY EQU            (HIGH ((( Dev_Freq / 4 ) * D'46' / D'10000' ) / 3 ) ) + 1
INNER_CNTR             EQU            40                                    ; RAM Location
OUTER_CNTR             EQU            41                                    ; RAM Location
;
T1OSO                  EQU            0                            ; The RC0 / T1OSO / T1CKI
;
RESET_V                EQU            0x0000                ; Address of RESET Vector
ISR_V                  EQU            0x0004                ; Address of Interrupt Vector
PMEM_END               EQU            0x07FF                ; Last address in Program Memory
TABLE_ADDR             EQU            0x0400                ; Address where to start Tables
;
COUNTER         EQU     0x021           ;
;
XMIT_DATA       EQU     0x30
DATA_CNT        EQU     0x31
ONES_CNT        EQU     0x32
CCP1_INT_CNT    EQU     0x33
CCPREG_HI       EQU     0x40
CCPREG_LO       EQU     0x41
DUMMY_PA        EQU     0x50
DUMMY_PB        EQU     0x51
DUMMY_PC        EQU     0x52
DUMMY_PD        EQU     0x53
DUMMY_PE        EQU     0x54
DC_HI           EQU     0x55
DC_LO           EQU     0x56
T2_PERIOD       EQU     0xA0
;
        list
```

```
MPASM 01.01 Released    COMP_1.ASM   7-13-1994  9:7:22                PAGE  1


LOC  OBJECT CODE     LINE SOURCE TEXT

                     0001         LIST    P = 16C74,  F = INHX8M, n = 66
                     0002 ;
                     0003 ;*****************************************************************************
                     0004 ;
                     0005 ; This program outputs a pulse train on the CCP1 pin, as specified by the
                     0006 ; values in the CCPR1H:CCPR1L.
                     0007 ;
                     0008 ;                          ____           ____      __
                     0009 ;  Pulse Train    ___|    |________|     |____|
                     0010 ;  Data Value          <-- 1 --> <- 0 ->
                     0011 ;                        ^      ^      ^    ^
                     0012 ;                        |      |      |    |
                     0013 ;                        |  T_ONE_BIT  |  T_ZERO_BIT
                     0014 ;                        |             |
                     0015 ;                    PULSE_TIME    PULSE_TIME
                     0016 ;
                     0017 ;
                     0018 ;       Program = COMP_1.ASM
                     0019 ;       Revision Date:   7-13-94
                     0020 ;
                     0021 ;*****************************************************************************
                     0022 ;
                     0023 ;
                     0024 ; HARDWARE SETUP
                     0025 ;    PORTB        -  Data to serial transmit on CCP pin
                     0026 ;
                     0027 ;
                     0028             INCLUDE <C74_reg.h>
                     0247
                     0028
                     0029             INCLUDE <COMP.h>
                     0053
                     0029
                     0030 ;
0001                 0031 PICMaster       EQU     TRUE        ; A Debugging Flag
0001                 0032 Debug           EQU     TRUE        ; A Debugging Flag
0001                 0033 Debug_PU        EQU     TRUE        ; A Debugging Flag
                     0034 ;
                     0035 ;
                     0036 ; Reset address. Determine type of RESET
                     0037 ;
                     0038         org     RESET_V             ; RESET vector location
```

```
0000 1683          0039 RESET       BSF     STATUS, RP0    ; Bank 1
0001 188E          0040             BTFSC   PCON, POR     ; Power-up reset?
0002 287C          0041             GOTO    START          ; YES
0003 28BA          0042             GOTO    OTHER_RESET    ; NO, a WDT or MCLR reset
                   0043 ;
                   0044 ; This is the Peripheral Interrupt routine. Need to determine the type
                   0045 ; of interrupt that occurred. The following interrupts are enabled:
                   0046 ;   1.  CCP Capture Occured
                   0047 ;
                   0049         org     ISR_V              ; Interrupt vector location
                   0050 PER_INT_V
                   0051         if ( Debug )
0004 1405          0052             bsf     PORTA, 0       ; Turn on strobe
                   0053         endif                      ;
0005 1283          0054             BCF     STATUS, RP0    ; Bank 0
0006 190C          0055             BTFSC   PIR1, CCP1IF   ; Compare Interrupt occured?
0007 280E          0056             GOTO    CCP1_INT       ; YES, Service the TMR1 Interrupt
                   0057 ERROR1                             ; NO, Error Condition - Unknown Interrupt
0008 1505          0058             BSF     PORTA, 2       ; Toggle a PORT pin
0009 1105          0059             BCF     PORTA, 2
000A 2808          0060             GOTO    ERROR1
                   0061 ;
                   0062 ERROR2                             ; NO, Error Condition - Unknown Interrupt
000B 1585          0063             BSF     PORTA, 3       ; Toggle a PORT pin
000C 1185          0064             BCF     PORTA, 3
000D 280B          0065             GOTO    ERROR2
                   0066 ;
                   0067 ;
                   0068 ;*************************************************************************
                   0069 ; In the CCP interrupt.
                   0070 ; Since timer1 is not cleared on a CCP match, the value in the
                   0071 ; CCPR1H:CCPR1L register pair must be updated. This is done with
                   0072 ; a 16-bit add. Also after the 1st CCP1 match (CCP1 pin goes high)
                   0073 ; the next match will force it low. Depending on the value of the data bit
                   0074 ; determines the value add to the CCPR1H:CCPR1L register pair.
                   0075 ;
                   0076 ; After the data has been transmitted, the pin will have a sync pulse and
                   0077 ; then remain low for 300 us.
                   0078 ;*************************************************************************
                   0079 ;
                   0080
                   0081 CCP1_INT
000E 110C          0082             BCF     PIR1, CCP1IF     ; Clear CCP1 Interrupt Flag
000F 0AB3          0083             INCF    CCP1_INT_CNT     ;
0010 1C33          0084             BTFSS   CCP1_INT_CNT, 0  ;
0011 2833          0085             GOTO    SYNC_PULSE
                   0086 DATA_PULSE
0012 03B1          0087             DECF    DATA_CNT         ; Decrement the Count of data bits
```

```
0013 1903          0088             BTFSC   STATUS, Z           ; Have we transmitted all the Data Bits?
0014 2827          0089             GOTO    PERIOD_DELTA        ; YES, Delay to 300 us
0015 0DB0          0090             RLF     XMIT_DATA, F        ; NO, get next bit to transmit
0016 1803          0091             BTFSC   STATUS, C           ; Is the bit to transmit a '1'?
0017 281F          0092             GOTO    ONE_DATA            ; YES, Stay low for 17.6 us
                   0093 ZERO_DATA
0018 302F          0094             MOVLW   LOW ( T_ZERO_BIT )  ; NO, Stay low for 8.8 us
0019 0795          0095             ADDWF   CCPR1L, F           ; Update Compare register pair latch
001A 1803          0096             BTFSC   STATUS, C           ;
001B 0A96          0097             INCF    CCPR1H, F           ;
001C 3000          0098             MOVLW   HIGH (T_ZERO_BIT )  ;
001D 0796          0099             ADDWF   CCPR1H, F           ;
001E 287A          0100             GOTO    RET_FIE
                   0101 ;
                   0102 ONE_DATA
001F 305E          0103             MOVLW   LOW ( T_ONE_BIT )   ; Stay low for 17.6 us
0020 0795          0104             ADDWF   CCPR1L, F           ; Update Compare register pair latch
0021 1803          0105             BTFSC   STATUS, C           ;
0022 0A96          0106             INCF    CCPR1H, F           ;
0023 3000          0107             MOVLW   HIGH ( T_ONE_BIT )  ;
0024 0796          0108             ADDWF   CCPR1H, F           ;
0025 0AB2          0109             INCF    ONES_CNT            ; Increment the number of '1's in the byte
0026 287A          0110             GOTO    RET_FIE
                   0111 ;
                   0112 PERIOD_DELTA
0027 0832          0113             MOVF    ONES_CNT, W         ;
0028 390F          0114             ANDLW   0x0F                ; Only want 9 states (0 1s to 8 1s)
0029 0782          0115             ADDWF   PCL, F              ;
002A 283B          0116             GOTO    ZERO_1              ; There was 0 ones in the data byte
002B 2842          0117             GOTO    ONE_1               ; There was 1 one in the data byte
002C 2849          0118             GOTO    TWO_1               ; There was 2 ones in the data byte
002D 2850          0119             GOTO    THREE_1             ; There was 3 ones in the data byte
002E 2857          0120             GOTO    FOUR_1              ; There was 4 ones in the data byte
002F 285E          0121             GOTO    FIVE_1              ; There was 5 ones in the data byte
0030 2865          0122             GOTO    SIX_1               ; There was 6 ones in the data byte
0031 286C          0123             GOTO    SEVEN_1             ; There was 7 ones in the data byte
0032 2873          0124             GOTO    EIGHT_1             ; There was 8 ones in the data byte
                   0125 ;
                   0126 SYNC_PULSE
0033 302F          0127             MOVLW   LOW ( PULSE_TIME )  ; Update Compare register pair latch
0034 0795          0128             ADDWF   CCPR1L, F           ;
0035 1803          0129             BTFSC   STATUS, C           ;
0036 0A96          0130             INCF    CCPR1H, F           ;
0037 3000          0131             MOVLW   HIGH ( PULSE_TIME ) ;
0038 0796          0132             ADDWF   CCPR1H, F           ;
0039 1417          0133             BSF     CCP1CON, 0          ; On Compare match, CCP1 pin = L
003A 0009          0134             RETFIE
                   0135 ;
```

```
0136 ZERO_1
0137                    MOVLW   LOW ( ZERO_1S )     ; Update Compare register pair latch
0138                    ADDWF   CCPR1L, F           ;
0139                    BTFSC   STATUS, C           ;
003E 0A96               INCF    CCPR1H, F           ;
0140                    MOVLW   HIGH ( ZERO_1S )    ;
003F 3002               ADDWF   CCPR1H, F           ;
0040 0796               GOTO    RET_FIE             ;
0041 287A
0144 ;
0145 ONE_1
0146                    MOVLW   LOW ( ONE_1S )      ; Update Compare register pair latch
0147                    ADDWF   CCPR1L, F           ;
0148                    BTFSC   STATUS, C           ;
0149                    INCF    CCPR1H, F           ;
0150                    MOVLW   HIGH ( ONE_1S )     ;
0151                    ADDWF   CCPR1H, F           ;
0152                    GOTO    RET_FIE             ;
0153 ;
0154 TWO_1
0155                    MOVLW   LOW ( TWO_1S )      ; Update Compare register pair latch
0156                    ADDWF   CCPR1L, F           ;
0157                    BTFSC   STATUS, C           ;
0158                    INCF    CCPR1H, F           ;
0159                    MOVLW   HIGH ( TWO_1S )     ;
004D 3002               ADDWF   CCPR1H, F           ;
004E 0796               GOTO    RET_FIE             ;
004F 287A
0162 ;
0163 THREE_1
0164                    MOVLW   LOW ( THREE_1S )    ; Update Compare register pair latch
0165                    ADDWF   CCPR1L, F           ;
0166                    BTFSC   STATUS, C           ;
0167                    INCF    CCPR1H, F           ;
0168                    MOVLW   HIGH ( THREE_1S )   ;
0169                    ADDWF   CCPR1H, F           ;
0170                    GOTO    RET_FIE             ;
0171 ;
0172 FOUR_1
0173                    MOVLW   LOW ( FOUR_1S )     ; Update Compare register pair latch
0174                    ADDWF   CCPR1L, F           ;
0175                    BTFSC   STATUS, C           ;
0176                    INCF    CCPR1H, F           ;
0177                    MOVLW   HIGH ( FOUR_1S )    ;
0178                    ADDWF   CCPR1H, F           ;
0179                    GOTO    RET_FIE             ;
0180 ;
0181 FIVE_1
0182                    MOVLW   LOW ( FIVE_1S )     ; Update Compare register pair latch
0183                    ADDWF   CCPR1L, F           ;

003B 30EC
003C 0795
003D 1803
003E 0A96
003F 3002
0040 0796
0041 287A

0042 30BD
0043 0795
0044 1803
0045 0A96
0046 3002
0047 0796
0048 287A

0049 308E
004A 0795
004B 1803
004C 0A96
004D 3002
004E 0796
004F 287A

0050 305F
0051 0795
0052 1803
0053 0A96
0054 3002
0055 0796
0056 287A

0057 3030
0058 0795
0059 1803
005A 0A96
005B 3002
005C 0796
005D 287A

005E 3001
005F 0795
```

```
0060 1803               BTFSC   STATUS, C               ;
0061 0A96               INCF    CCPR1H, F               ;
0062 3002               MOVLW   HIGH ( FIVE_1S )        ;
0063 0796               ADDWF   CCPR1H, F               ;
0064 287A               GOTO    RET_FIE
                        ;
0190 SIX_1
0191 30D2               MOVLW   LOW ( SIX_1S )          ; Update Compare register pair latch
0192 0795               ADDWF   CCPR1L, F               ;
0193 1803               BTFSC   STATUS, C               ;
0194 0A96               INCF    CCPR1H, F               ;
0195 3001               MOVLW   HIGH ( SIX_1S )         ;
0196 0796               ADDWF   CCPR1H, F               ;
0197 287A               GOTO    RET_FIE
0198 ;
0199 SEVEN_1
0200 30A3               MOVLW   LOW ( SEVEN_1S )        ; Update Compare register pair latch
0201 0795               ADDWF   CCPR1L, F               ;
0202 1803               BTFSC   STATUS, C               ;
0203 0A96               INCF    CCPR1H, F               ;
0204 3001               MOVLW   HIGH ( SEVEN_1S )       ;
0205 0796               ADDWF   CCPR1H, F               ;
0206 287A               GOTO    RET_FIE
0207 ;
0208 EIGHT_1
0209 3074               MOVLW   LOW ( EIGHT_1S )        ; Update Compare register pair latch
0210 0795               ADDWF   CCPR1L, F               ;
0211 1803               BTFSC   STATUS, C               ;
0212 0A96               INCF    CCPR1H, F               ;
0213 3001               MOVLW   HIGH ( EIGHT_1S )       ;
0214 0796               ADDWF   CCPR1H, F               ;
0215 287A               GOTO    RET_FIE
0216
0217 RET_FIE
0218                    BCF     CCP1CON, 0              ; On Compare match, CCP1 pin = H
0219 0009               RETFIE                          ; Return / Enable Global Interrupts
0220 ;
0222 ;
0223 ;*************************************************************************
0224 ;*****    Start program here, Power-On Reset occurred.
0225 ;*************************************************************************
0226 ;
0227 START                                               ; POWER_ON Reset (Beginning of program)
0228 007C 1283          BCF     STATUS, RP0             ; Bank 0
0229 007D 018F          CLRF    TMR1H                   ;
0230 007E 018E          CLRF    TMR1L                   ;
0231 ;
0232 MCLR_RESET                                         ; A Master Clear Reset
```

```
007F 1283           0233            BCF     STATUS, RP0     ; Bank 0
0080 0183           0234            CLRF    STATUS          ; Do initialization (Bank 0)
0081 018B           0235            CLRF    INTCON
0082 018C           0236            CLRF    PIR1
0083 1683           0237            BSF     STATUS, RP0     ; Bank 1
0084 3080           0238            MOVLW   0x80            ; Disable PORTB weak pull-ups
0085 0081           0239            MOVWF   OPTION_R        ;
0086 018C           0240            CLRF    PIE1            ; Disable all peripheral interrupts
0087 30FF           0241            MOVLW   0xFF            ;
0088 009F           0242            MOVWF   ADCON1          ; Port A is Digital.
                    0243 ;
                    0244 ;
0089 1283           0245            BCF     STATUS, RP0     ; Bank 0
008A 0185           0246            CLRF    PORTA           ; ALL PORT output should output Low.
008B 0186           0247            CLRF    PORTB
008C 0187           0248            CLRF    PORTC
008D 0188           0249            CLRF    PORTD
008E 0189           0250            CLRF    PORTE
008F 1010           0251            BCF     T1CON, TMR1ON   ; Timer 1 is NOT incrementing
                    0252 ;
0090 1683           0253            BSF     STATUS, RP0     ; Select Bank 1
0091 0185           0254            CLRF    TRISA           ; RA5 -  0 outputs
0092 30FF           0255            MOVLW   0xFF            ;
0093 0086           0256            MOVWF   TRISB           ; RB Port are inputs
0094 0187           0257            CLRF    TRISC           ; RC Port are outputs
0095 0188           0258            CLRF    TRISD           ; RD Port are outputs
0096 0189           0259            CLRF    TRISE           ; RE Port are outputs
0097 150C           0260            BSF     PIE1, CCP1IE    ; Enable CCP1 Interrupt
0098 1283           0261            BCF     STATUS, RP0     ; Select Bank 0
                    0262 ;
                    0264 ;
                    0265 ;
                    0266 ; Initialize the Special Function Registers (SFR) interrupts
                    0267 ;
0099 018C           0268            CLRF    PIR1            ;
009A 0190           0269            CLRF    T1CON           ; Timer mode
009B 170B           0270            BSF     INTCON, PEIE    ; Enable Peripheral Interrupts
009C 178B           0271            BSF     INTCON, GIE     ; Enable all Interrupts
                    0272 ;
                    0273 ; Set-up timer and compare latches and then turn timer1 on.
                    0274 ;
009D 1010           0275            BCF     T1CON, TMR1ON   ; Turn OFF timer1
009E 3041           0276            MOVLW   CCPREG_HI       ; TMR1 = CCPR1H:CCPR1L - 1
009F 008F           0277            MOVWF   TMR1H           ;
00A0 3042           0278            MOVLW   CCPREG_LO       ;
00A1 008E           0279            MOVWF   TMR1L           ;
00A2 038E           0280            DECF    TMR1L           ;
00A3 1803           0281            BTFSC   STATUS, C       ;
```

```
00A4 038F           0282            DECF    TMR1H           ;
00A5 3008           0283            MOVLW   0x08            ; On match CCP1 = H level
00A6 0097           0284            MOVWF   CCP1CON         ;
00A7 3009           0285            MOVLW   0x09            ;
00A8 00B1           0286            MOVWF   DATA_CNT        ; 8-bits to transfer
00A9 01B2           0287            CLRF    ONES_CNT        ; Result after xmit holds the number of 1's in a byte
00AA 30FF           0288            MOVLW   0xFF            ;
00AB 00B3           0289            MOVWF   CCP1_INT_CNT    ; No CCP1 transmit interrups yet
00AC 1410           0290            BSF     T1CON, TMR1ON   ; Turn ON timer1
                    0291 ;
                    0292 ;
                    0293 ; This code segment is an infinite loop that will always transmit the data
                    0294 ; contained in the XMIT_DATA register. After each byte is transmitted a new
                    0295 ; byte is read. If using PICMASTER (in stand alone mode), this is read from
                    0296 ; a register that is updated after a break (at NOP). If in a system, PORTB
                    0297 ; is read. All other variables are reinitalized after each byte.
                    0298 ;
                    0299 NEXT_BYTE
                    0300 ;
00AD 0831           0301 WAIT       MOVF    DATA_CNT, w     ;
00AE 1D03           0302            BTFSS   STATUS, Z       ; Is DATA_CNT = 0 ?
00AF 28AD           0303            GOTO    WAIT            ; NO, must wait until YES
00B0 0000           0304            NOP                     ;
                    0305        if ( Debug )
00B1 1005           0306            bcf     PORTA, 0        ; Turn off strobe
                    0307        endif                       ;
                    0308
                    0309        if ( PICMaster )
00B2 0840           0310            MOVF    DUMMY_PB, W     ;
                    0311        else
                    0312            MOVF    PORTB, W        ;
                    0313        endif
00B3 00B0           0314            MOVWF   XMIT_DATA       ; New data to transmit
00B4 30FF           0315            MOVLW   0xFF            ;
00B5 00B3           0316            MOVWF   CCP1_INT_CNT    ;
00B6 3009           0317            MOVLW   0x09            ;
00B7 00B1           0318            MOVWF   DATA_CNT        ;
00B8 01B2           0319            CLRF    ONES_CNT        ;
00B9 28AD           0320            GOTO    NEXT_BYTE       ;
                    0321 ;
                    0322 ;
                    0323 ; Here is where you do things depending on the type of RESET (Not a Power-On Reset).
                    0324 ;
00BA 1E03           0325 OTHER_RESET   BTFSS  STATUS, TO     ; WDT Time-out?
00BB 2808           0326 WDT_TIMEOUT   GOTO   ERROR1        ; YES, This is error condition
                    0327        if ( Debug_PU )
00BC 287C           0328            goto    START           ; MCLR reset, Goto START
                    0329        else
```

```
                     0330             GOTO   MCLR_RESET       ; MCLR reset, Goto MCLR_RESET
                     0331         endif
                     0332 ;
                     0333         if (Debug )
00BD 0000            0334 END_START     NOP                   ; END lable for debug
                     0335         endif
                     0336 ;
                     0337 ;
                     0338     org     PMEM_END                 ; End of Program Memory
07FF 2808            0339             GOTO    ERROR1          ; If you get here your program was lost
                     0340
                     0341     end
                     0342
                     0343
                     0344



MEMORY USAGE MAP ('X' = Used,  '-' = Unused)

0000 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX
0040 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX

0080 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXX-
00C0 : ________ ________ ________ ________

0780 : ________ ________ ________ ________
07C0 : ________ ________ ________ _______X

All other memory blocks unused.


Errors   :    0
Warnings :   10
```

APPENDIX B2: COMP.H

```
    nolist
;*****************************************************************************
;
;  This is the custom Header File for the real time clock application note
;       PROGRAM:        CLOCK.H
;       Revision:       7-13-94
;
;*****************************************************************************
; This is used for the ASSEMBLER to recalculate certain frequency
; dependant variables. The value of Dev_Freq must be changed to
; reflect the frequency that the device actually operates at.
;
Dev_Freq        EQU     D'10000000'             ; Device Frequency is 4 MHz
PULSE_TIME      EQU     (( Dev_Freq / D'4000' ) * D'188' / D'10000' )
T_ZERO_BIT      EQU     (( Dev_Freq / D'4000' ) * D'188' / D'10000' )
T_ONE_BIT       EQU     ( Dev_Freq / D'4000' ) * D'376' / D'10000' )
;
ZERO_1S         EQU     ((( Dev_Freq / D'4000' ) * (D'6000' - (D'16' * D'188'))) / D'10000' )
ONE_1S          EQU     ( ( Dev_Freq / D'4000' ) * (D'6000' - (3 * D'188' + D'14' * D'188')) / D'10000')
TWO_1S          EQU     ( ( Dev_Freq / D'4000' ) * (D'6000' - (6 * D'188' + D'12' * D'188')) / D'10000')
THREE_1S        EQU     ( ( Dev_Freq / D'4000' ) * (D'6000' - (D'9' * D'188' + D'10' * D'188')) / D'10000')
FOUR_1S         EQU     ( ( Dev_Freq / D'4000' ) * (D'6000' - (D'12' * D'188' + 8 * D'188')) / D'10000')
FIVE_1S         EQU     ( ( Dev_Freq / D'4000' ) * (D'6000' - (D'15' * D'188' + 6 * D'188')) / D'10000')
SIX_1S          EQU     ( ( Dev_Freq / D'4000' ) * (D'6000' - (D'18' * D'188' + 4 * D'188')) / D'10000')
SEVEN_1S        EQU     ( ( Dev_Freq / D'4000' ) * (D'6000' - (D'21' * D'188' + 2 * D'188')) / D'10000')
EIGHT_1S        EQU     ( ( Dev_Freq / D'4000' ) * (D'6000' - (D'24' * D'188') ) / D'10000' )

DB_HI_BYTE      EQU     (HIGH ((( Dev_Freq / 4 ) * 1 / D'1000' ) / 3 ) ) + 1
LCD_INIT_DELAY  EQU     (HIGH ((( Dev_Freq / 4 ) * D'46' / D'10000' ) / 3 ) ) + 1
INNER_CNTR      EQU     40                      ; RAM Location
OUTER_CNTR      EQU     41                      ; RAM Location
;
T1OSO           EQU     0                       ; The RC0 / T1OSO / T1CKI
;
RESET_V         EQU     0x0000                  ; Address of RESET Vector
ISR_V           EQU     0x0004                  ; Address of Interrupt Vector
PMEM_END        EQU     0x07FF                  ; Last address in Program Memory
TABLE_ADDR      EQU     0x0400                  ; Address where to start Tables
;
COUNTER         EQU     0x021                   ;
;
XMIT_DATA       EQU     0x30
DATA_CNT        EQU     0x31
ONES_CNT        EQU     0x32
CCP1_INT_CNT    EQU     0x33
DUMMY_PB        EQU     0x40
CCPREG_HI       EQU     0x41
CCPREG_LO       EQU     0x42
;
        list
```

```
MPASM 01.01 Released    CAPT_2.ASM   7-19-1994  10:54:15                PAGE  1


LOC  OBJECT CODE     LINE SOURCE TEXT

                     0001         LIST    P = 16C74,  F = INHX8M, n = 66
                     0002 ;
                     0003 ;*******************************************************************************
                     0004 ;
                     0005 ; This program implements a real time clock using the TMR1 module of the
                     0006 ; PIC16Cxx family.
                     0007 ;
                     0008 ;       Program = CAPT_2.ASM
                     0009 ;       Revision Date:   7-19-94
                     0010 ;
                     0011 ;*******************************************************************************
                     0012 ;
                     0013 ;
                     0014 ; HARDWARE SETUP
                     0015 ;
                     0016 ;           CCP2 Compare Output
                     0017 ;           CCP1 Capture Input
                     0018 ;           CCP2  –>  CCP1
                     0019 ;
                     0020 ;
                     0021             INCLUDE <C74_reg.h>
                     0249
                     0021
                     0022             INCLUDE <CAPT.h>
                     0022
                     0023 ;
                     0024 ;
0001                 0025 PICMaster       EQU     TRUE           ; A Debugging Flag
0001                 0026 Debug           EQU     TRUE           ; A Debugging Flag
0001                 0027 Debug_PU        EQU     TRUE           ; A Debugging Flag
                     0028 ;
                     0029 ;
                     0030 ; Reset address. Determine type of RESET
                     0031 ;
                     0032         org     RESET_V                ; RESET vector location
0000 1683            0033 RESET       BSF     STATUS, RP0        ; Bank 1
0001 188E            0034             BTFSC   PCON, POR          ; Power-up reset?
0002 282F            0035             GOTO    START              ; YES
0003 2861            0036             GOTO    OTHER_RESET        ; NO, a WDT or MCLR reset
                     0037 ;
                     0038 ; This is the Peripheral Interrupt routine. Need to determine the type
                     0039 ; of interrupt that occurred. The following interrupts are enabled:
```

```
                    ;   1.  CCP1 Capture Occured
                    ;   2.  CCP2 Compare Occured
                    ;
                            org     ISR_V                   ; Interrupt vector location
                    PER_INT_V
0004 1283           
0005 190C           
0006 281D           
0007 180D           
0008 2811           
0009 180C           
000A 282D           
                    ERROR1                                  ; NO, Error Condition - Unknown Interrupt
000B 1488           
000C 1088           
000D 280B           
                    ;
                    ERROR2                                  ; NO, Error Condition - Unknown Interrupt
000E 1508           
000F 1108           
0010 280E           
                    ;
                    ; The Compare generates a Square wave based on the value on PORTB (in DUMMY_PB)
                    ; and on PORTD (in DUMMY_PD). PORTB is loaded into low compare latch and PORTD
                    ; is loaded into the high compare latch. If the value of the ports is not changed,
                    ; a capture overflow condition will occur when PORTD:PORTB > 7Fh. This overflow
                    ; is only indicated by the time between captures being much less then expected.
                    ;
                    COMPARE
0011 100D           
                            if ( PICMaster )
0012 0851           
                            else
                            endif
0013 079B           
0014 1803           
0015 0A9C           
                            if ( PICMaster )
0016 0853           
                            else
                            endif
0017 079C           
0018 0AB3           
0019 141D           
001A 1C33           
001B 101D           
```

```
                            BCF     STATUS, RP0             ; Bank 0
                            BTFSC   PIR1, CCP1IF            ; CCP1 Interrupt occured? (Capture)
                            GOTO    CAPTURE                 ; YES, Service the CCP1 Interrupt
                            BTFSC   PIR2, CCP2IF            ; CCP2 Interrupt occured? (Compare)
                            GOTO    COMPARE                 ; YES, Service the CCP2 Interrupt
                            BTFSC   PIR1, TMR1IF            ; NO, Timer 1 Overflow?
                            GOTO    T1OVFL                  ; YES,
                            BSF     PORTD, 1                ; Toggle a PORT pin
                            BCF     PORTD, 1
                            GOTO    ERROR1
                            BSF     PORTD, 2                ; Toggle a PORT pin
                            BCF     PORTD, 2
                            GOTO    ERROR2
                            BCF     PIR2, CCP2IF            ; Clear CCP2 Interrupt Flag
                                MOVF    DUMMY_PB, W         ;
                                MOVF    PORTB, W            ;
                            ADDWF   CCPR2L, F               ; Update Compare register pair latch
                            BTFSC   STATUS, C               ;
                            INCF    CCPR2H, F               ;
                                MOVF    DUMMY_PD, W         ;
                                MOVF    PORTD, W            ;
                            ADDWF   CCPR2H, F               ;
                            INCF    CCP2_INT_CNT            ;
                            BSF     CCP2CON, 0              ; On Compare match, CCP2 pin = L
                            BTFSS   CCP2_INT_CNT, 0         ;
                            BCF     CCP2CON, 0              ; On Compare match, CCP2 pin = H
```

```
                     0089 END_COMPARE
001C 0009            0090             RETFIE                  ; Return / Enable Global Interrupts
                     0091
                     0092 ;
                     0093 ; The result of the new capture minus the old capture is stored in the new capture
                     0094 ; registers (CAPT_NEW_H:CAPT_NEW_L)
                     0095 ;
                     0096 CAPTURE
001D 110C            0097             BCF     PIR1, CCP1IF    ; Clear CCP1 Interrupt Flag
001E 0815            0098             MOVF    CCPR1L, W       ; New capture value (low byte)
001F 00C1            0099             MOVWF   CAPT_NEW_L      ;
0020 0816            0100             MOVF    CCPR1H, W       ; New capture value (high byte)
0021 00C0            0101             MOVWF   CAPT_NEW_H      ;
                     0102 ;
0022 0843            0103             MOVF    CAPT_OLD_L, W   ;
0023 02C1            0104             SUBWF   CAPT_NEW_L, F   ; Subtract the low bytes of the 2 captures
0024 1C03            0105             BTFSS   STATUS, C       ; Did a borrow occur?
0025 03C0            0106             DECF    CAPT_NEW_H, F   ; YES, Decrement old capture (high byte)
0026 0842            0107             MOVF    CAPT_OLD_H, W   ; New capture value (low byte)
0027 02C0            0108             SUBWF   CAPT_NEW_H, F   ; Subtract the low bytes of the 2 captures
0028 0815            0109 LOAD_OLD    MOVF    CCPR1L, W       ; New capture value (low byte)
0029 00C3            0110             MOVWF   CAPT_OLD_L      ;
002A 0816            0111             MOVF    CCPR1H, W       ; New capture value (high byte)
002B 00C2            0112             MOVWF   CAPT_OLD_H      ;
                     0113 END_CAPTURE
002C 0009            0114             RETFIE
                     0115 ;
                     0116 ;
                     0117 T1OVFL
002D 100C            0118             BCF     PIR1, TMR1IF    ; Clear T1 Overflow Interrupt Flag
002E 0009            0119             RETFIE                  ; Return / Enable Global Interrupts
                     0120 ;
                     0121 ;
                     0122 ;*******************************************************************************
                     0123 ;*****      Start program here, Power-On Reset occurred.
                     0124 ;*******************************************************************************
                     0125 ;
                     0126 START                               ; POWER_ON Reset (Beginning of program)
002F 1283            0127             BCF     STATUS, RP0     ; Bank 0
0030 018F            0128             CLRF    TMR1H           ;
0031 018E            0129             CLRF    TMR1L           ;
                     0130 ;
                     0131 MCLR_RESET                          ; A Master Clear Reset
0032 1283            0132             BCF     STATUS, RP0     ; Bank 0
0033 0183            0133             CLRF    STATUS          ; Do initialization (Bank 0)
0034 018B            0134             CLRF    INTCON
0035 018C            0135             CLRF    PIR1
0036 1683            0136             BSF     STATUS, RP0     ; Bank 1
```

```
0037 3000           0137            MOVLW   0x00            ; The LCD module does not like to work w/ weak pull-ups
0038 0081           0138            MOVWF   OPTION_R        ;
0039 018C           0139            CLRF    PIE1            ; Disable all peripheral interrupts
003A 018D           0140            CLRF    PIE2            ; Disable all peripheral interrupts
003B 30FF           0141            MOVLW   0xFF            ;
003C 009F           0142            MOVWF   ADCON1          ; Port A is Digital.
                    0143 ;
                    0144 ;
003D 1283           0145            BCF     STATUS, RP0     ; Bank 0
003E 0185           0146            CLRF    PORTA           ; ALL PORT output should output Low.
003F 0186           0147            CLRF    PORTB
0040 0187           0148            CLRF    PORTC
0041 0188           0149            CLRF    PORTD
0042 0189           0150            CLRF    PORTE
0043 1010           0151            BCF     T1CON, TMR1ON   ; Timer 1 is NOT incrementing
                    0152 ;
0044 1683           0153            BSF     STATUS, RP0     ; Select Bank 1
0045 0185           0154            CLRF    TRISA           ; RA5 -  0 outputs
0046 30FF           0155            MOVLW   0xFF            ;
0047 0086           0156            MOVWF   TRISB           ; RB7 - 0 inputs
0048 0187           0157            CLRF    TRISC           ; RC Port are outputs
0049 1507           0158            BSF     TRISC, 2        ; CCP1 is an INPUT
004A 0088           0159            MOVWF   TRISD           ; RD Port are inputs
004B 0189           0160            CLRF    TRISE           ; RE Port are outputs
004C 150C           0161            BSF     PIE1, CCP1IE    ; Enable CCP1 Interrupt
004D 140D           0162            BSF     PIE2, CCP2IE    ; Enable CCP2 Interrupt
004E 1283           0163            BCF     STATUS, RP0     ; Select Bank 0
                    0164 ;
                    0165 ;
                    0166 ; Initialize the Special Function Registers (SFR) interrupts
                    0167 ;
004F 018C           0168            CLRF    PIR1            ;
0050 018D           0169            CLRF    PIR2            ;
0051 0190           0170            CLRF    T1CON           ; Timer mode
0052 170B           0171            BSF     INTCON, PEIE    ; Enable Peripheral Interrupts
0053 178B           0172            BSF     INTCON, GIE     ; Enable all Interrupts
                    0173 ;
                    0174 ; Set-up timer and compare latches and then turn timer1 on.
                    0175 ;
0054 1010           0176            BCF     T1CON, TMR1ON    ; Turn OFF timer1
                    0177        if ( PICMaster )
0055 0851           0178            MOVF    DUMMY_PB, W      ;
                    0179        else
                    0180            MOVF    PORTB, W         ;
                    0181        endif
0056 079B           0182            ADDWF   CCPR2L, F        ; Update Compare register pair latch
0057 1803           0183            BTFSC   STATUS, C        ;
0058 0A9C           0184            INCF    CCPR2H, F        ;
```

```
                        0185         if ( PICMaster )
0059 08D3               0186             MOVF    DUMMY_PD           ;
                        0187         else
                        0188             MOVF    PORTD, W           ;
                        0189         endif
005A 079C               0190             ADDWF   CCPR2H, F          ;
005B 3008               0191             MOVLW   0x08               ; On match CCP2 = H level
005C 009D               0192             MOVWF   CCP2CON            ;
005D 3005               0193             MOVLW   0x05               ; Capture on every rising edge
005E 0097               0194             MOVWF   CCP1CON            ;
005F 1410               0195             BSF     T1CON, TMR1ON      ; Turn ON timer1
                        0196 ;
                        0197 ;
                        0198 ;
0060 2860               0199 lzz         goto    lzz                ; Loop waiting for interrupts (for use with PICMASTER)
                        0200 ;
                        0201 ;
                        0202 ; Here is where you do things depending on the type of RESET (Not a Power-On Reset).
                        0203 ;
0061 1E03               0204 OTHER_RESET   BTFSS  STATUS, TO         ; WDT Time-out?
0062 280B               0205 WDT_TIMEOUT   GOTO   ERROR1            ; YES, This is error condition
                        0206         if ( Debug_PU )
0063 282F               0207             goto   START               ; MCLR reset, Goto START
                        0208         else
                        0209             GOTO   MCLR_RESET          ; MCLR reset, Goto MCLR_RESET
                        0210         endif
                        0211 ;
                        0212         if (Debug )
0064 0000               0213 END_START     NOP                      ; END lable for debug
                        0214         endif
                        0215 ;
                        0216 ;
                        0217     org     PMEM_END                   ; End of Program Memory
07FF 280B               0218             GOTO    ERROR1             ; If you get here your program was lost
                        0219
                        0220     end
                        0221
                        0222
```

```
MEMORY USAGE MAP ('X' = Used,  '-' = Unused)

0000 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX
0040 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXX___________ ________________

0780 : ________ ________ ________ ________
07C0 : ________ ________ ________ _______X

All other memory blocks unused.

Errors   :   0
Warnings :  13
```

APPENDIX C2: CAPT.H

```
       nolist
;*************************************************************************
;
;  This is the custom Header File for the real time clock application note
;       PROGRAM:       CLOCK.H
;       Revision:      7-19-94
;
;*************************************************************************
; This is used for the ASSEMBLER to recalculate certain frequency
; dependant variables. The value of Dev_Freq must be changed to
; reflect the frequency that the device actually operates at.
;
Dev_Freq               EQU            D'4000000'            ; Device Frequency is 4 MHz
DB_HI_BYTE             EQU            (HIGH ((( Dev_Freq / 4 ) * 1 / D'1000' ) / 3 ) ) + 1
LCD_INIT_DELAY         EQU            (HIGH ((( Dev_Freq / 4 ) * D'46' / D'10000' ) / 3 ) ) + 1
INNER_CNTR             EQU            40                    ; RAM Location
OUTER_CNTR             EQU            41                    ; RAM Location
;
T1OSO                  EQU            0                     ; The RC0 / T1OSO / T1CKI
;
RESET_V                EQU            0x0000                ; Address of RESET Vector
ISR_V                  EQU            0x0004                ; Address of Interrupt Vector
PMEM_END               EQU            0x07FF                ; Last address in Program Memory
TABLE_ADDR             EQU            0x0400                ; Address where to start Tables
;
COUNTER                EQU            0x021                 ;
CCP2_INT_CNT           EQU            0x33
;
;
; DUMMY_PD:DUMMY_PB contain the value to be loaded into the CCP2 compare registers
; (CCPR2H:CCPR2L)
;
DUMMY_PA               EQU            0x50
DUMMY_PB               EQU            0x51
DUMMY_PC               EQU            0x52
DUMMY_PD               EQU            0x53
DUMMY_PE               EQU            0x54
;
;
; CAPT_NEW_H:CAPT_NEW_L stores the NEW captured value and the result of the
; subtraction between this capture and the previous.
;    CAPT_NEW_H:CAPT_NEW_L  =  CAPT_NEW_H:CAPT_NEW_L - CAPT_OLD_H:CAPT_OLD_L
;
; After all computations the new capture value is moved to the CAPT_OLD_H:CAPT_OLD_L
; in preperation for the next capture value.
;
CAPT_NEW_H             EQU            0x040                 ;
CAPT_NEW_L             EQU            0x041                 ;
CAPT_OLD_H             EQU            0x042                 ;
CAPT_OLD_L             EQU            0x043                 ;
;
       list
```

WORLDWIDE SALES & SERVICE

AMERICAS

Corporate Office
Microchip Technology Inc.
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 602 786-7200 Fax: 602 786-7277
Technical Support: 602 786-7627
Web: http://www.mchip.com/microhip

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

Boston
Microchip Technology Inc.
5 Mount Royal Avenue
Marlborough, MA 01752
Tel: 508 480-9990 Fax: 508 480-8575

Chicago
Microchip Technology Inc.
333 Pierce Road, Suite 180
Itasca, IL 60143
Tel: 708 285-0071 Fax: 708 285-0075

Dallas
Microchip Technology Inc.
14651 Dallas Parkway, Suite 816
Dallas, TX 75240-8809
Tel: 214 991-7177 Fax: 214 991-8588

Dayton
Microchip Technology Inc.
35 Rockridge Road
Englewood, OH 45322
Tel: 513 832-2543 Fax: 513 832-2841

Los Angeles
Microchip Technology Inc.
18201 Von Karman, Suite 455
Irvine, CA 92715
Tel: 714 263-1888 Fax: 714 263-1338

New York
Microchip Technology Inc.
150 Motor Parkway, Suite 416
Hauppauge, NY 11788
Tel: 516 273-5305 Fax: 516 273-5335

AMERICAS (continued)

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

ASIA/PACIFIC

Hong Kong
Microchip Technology
Unit No. 3002-3004, Tower 1
Metroplaza
223 Hing Fong Road
Kwai Fong, N.T. Hong Kong
Tel: 852 2 401 1200 Fax: 852 2 401 3431

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

Singapore
Microchip Technology
200 Middle Road
#10-03 Prime Centre
Singapore 188980
Tel: 65 334 8870 Fax: 65 334 8850

Taiwan
Microchip Technology
10F-1C 207
Tung Hua North Road
Taipei, Taiwan, ROC
Tel: 886 2 717 7175 Fax: 886 2 545 0139

EUROPE

United Kingdom
Arizona Microchip Technology Ltd.
Unit 6, The Courtyard
Meadow Bank, Furlong Road
Bourne End, Buckinghamshire SL8 5AJ
Tel: 44 0 1628 851077 Fax: 44 0 1628 850259

France
Arizona Microchip Technology SARL
2 Rue du Buisson aux Fraises
91300 Massy - France
Tel: 33 1 69 53 63 20 Fax: 33 1 69 30 90 79

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

Italy
Arizona Microchip Technology SRL
Centro Direzionale Colleoni
Palazzo Pegaso Ingresso No. 2
Via Paracelso 23, 20041
Agrate Brianza (MI) Italy
Tel: 39 039 689 9939 Fax: 39 039 689 9883

JAPAN

Microchip Technology Intl. Inc.
Benex S-1 6F
3-18-20, Shin Yokohama
Kohoku-Ku, Yokohama
Kanagawa 222 Japan
Tel: 81 45 471 6166 Fax: 81 45 471 6122

9/22/95