

## Using the CCP Module(s)

*Author: Mark Palmer  
Microchip Technology Inc.*

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

The (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 Timer1 be operating in timer mode or synchronized counter mode.

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

**TABLE 1: CCP MODE - TIMER RESOURCE**

CCP MODE	Timer Resource
Capture	Timer1
Compare	Timer1
PWM	Timer2

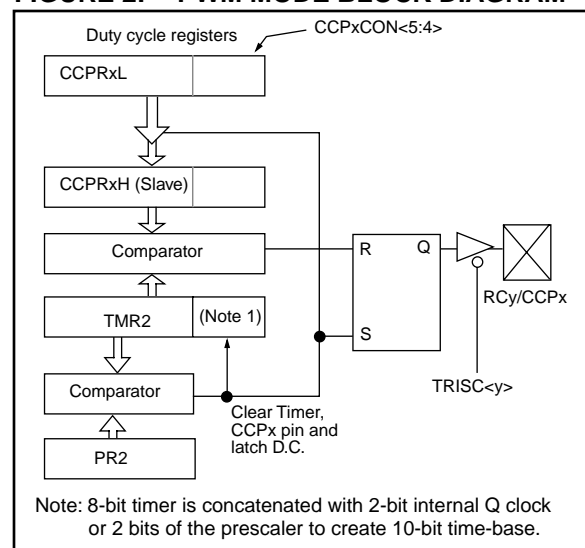
### CCP OPERATION

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.

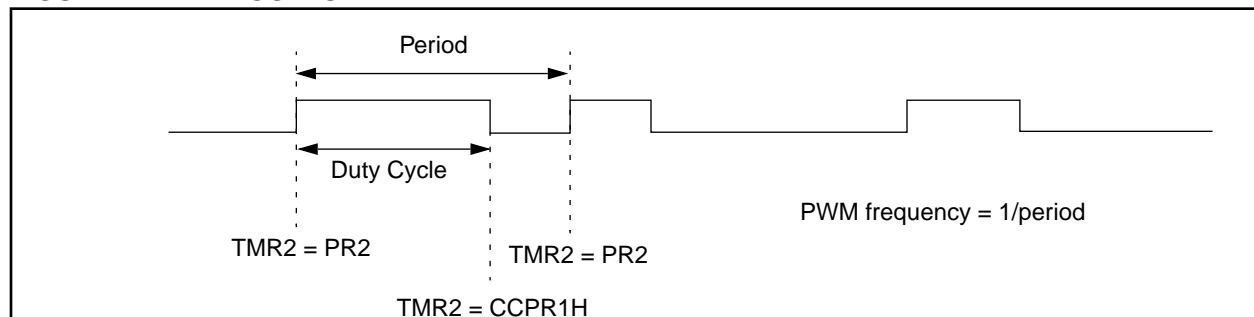
#### PWM Mode

A Pulse Width Modulation output (Figure 1) is a signal that has a time-base (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**



**FIGURE 1: PWM OUTPUT**



# AN594

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 Timer2 module. This gives 1024 steps of variance from an 8-bit overflow counter. This gives a maximum accuracy of  $T_{osc}$  (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 Timer2 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 Timer2 = Timer2 Period Register (PR2).

The period of Timer 2 (and PWM) is determined by the frequency of the device, the Timer2 prescaler value (1, 4 or 16), and the Timer2 Period Register. Equation 1 shows the calculation of the PWM period, duty cycle, and the minimum and maximum frequencies.

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 of 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.

## EQUATION 1: PWM PERIOD, DUTY CYCLE, AND FREQUENCIES

PWM Period =  $[(PR2) + 1] \cdot 4 T_{osc} \cdot (\text{Timer 2 prescale value})$

PWM Duty Cycle =  $[CCPRxL:CCPRxCON<5:4>] \cdot 4 T_{osc} \cdot (\text{Timer2 prescale value})$

PWM maximum frequency

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

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

PWM minimum frequency

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

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

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.

TABLE 2: PWM FREQUENCY FOR DIFFERENT DEVICE FREQUENCIES

PWM Resolution	20 MHz		10 MHz		2 MHz		Units
	Min	Max	Min	Max	Min	Max	
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

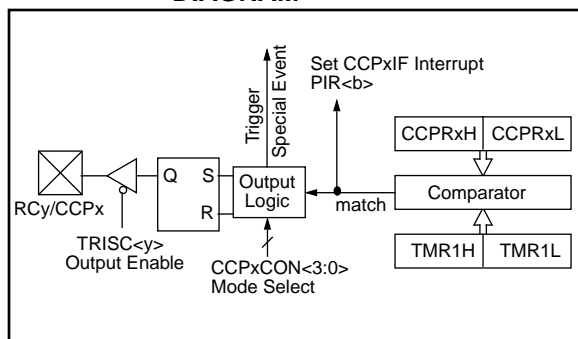
## Compare Mode

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/DONE bit

The CCPxM3:CCPxM0 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.

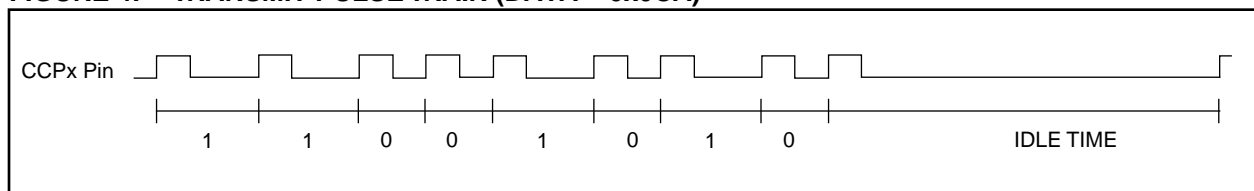
**FIGURE 3: COMPARE MODE BLOCK DIAGRAM**



When the CCP module is in the OFF state (CCPxM3:CCPxM0 = 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  $\mu$ s. Each data bit has a sync pulse (High level) of 8.8  $\mu$ s. 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  $\mu$ s while a '1' bit is low for 37.6  $\mu$ s. After the last data bit has been transmitted, another sync pulse is transmitted and the output remains low (idle time) until the 600  $\mu$ s data period has completed. An example of the pulse train for the a data byte of 0x0CA is shown in Figure 4, and has an idle time of 224  $\mu$ s. These pulse times are based off the device operational frequency. The program header file, COMP.H, calculates the values to be 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)**



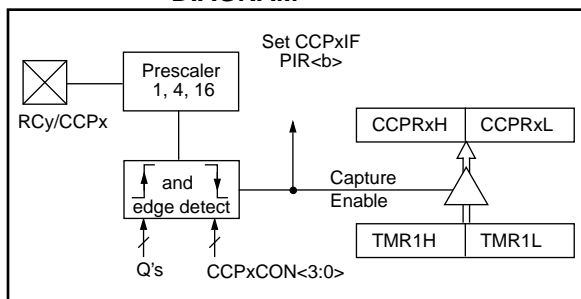
## Capture Mode

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 CCPxM3:CCPxM0 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**

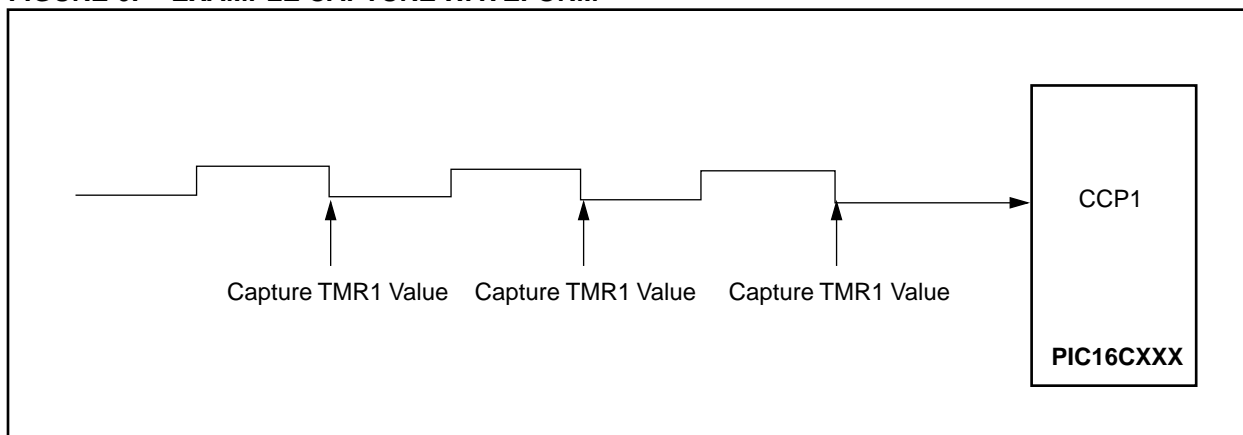


The changing of the Capture mode, via the CCPxM3:CCPxM0 bits, may cause the CCPxIF bit to be set. This “false” interrupt should be cleared (ignored) after changing between capture modes. The CCP prescaler is only cleared by configuring the CCP module into the OFF state (CCPxM3:CCPxM0 = 0h). Figure 5 shows the block diagram of the CCP module in Capture mode. The use 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 (Timer1). 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 WAVEFORM**



## INTERACTION OF CCP MODULES

Due to the modularity of the PIC16CXXX 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 time-base for both captures. This means that they will have the same capture resolution, as determined by the Timer1 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 time-base for both the captures and the compare. This means that the capture and the compare will have the same resolution, as determined by the Timer1 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 register (when in special Trigger mode). Care must be taken in system design to ensure that this clearing of the TMR1 register does not have any negative impact on the capture function.

**TABLE 3: INTERACTION OF TWO CCP MODULES**

CCPx Mode	CCPy Mode	Interaction
Capture	Capture	Same Timer1 time-base.
Capture	Capture	The compare could be configured for trigger special event, which clears the TMR1 register.
Capture	Capture	The compare(s) could be configured for trigger special event, which clears the TMR1 register.
PWM	PWM	The PWMs will have the same frequency, and update rate (Timer2 interrupt).
PWM	Capture	None
PWM	Capture	None

## EXAMPLE 1:

<u>ACTION</u>	<u>TIMER1 STATE</u>	<u>COMMENT</u>
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 Compare Modes

When two CCP modules are in a Compare mode, Timer1 is the time-base for both compares. This means that they will have the same compare resolution, as determined by the Timer1 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 register (when in special Trigger mode), care must be taken in system design to ensure that this clearing of the TMR1 register does not have any negative impact on the compare function. If both compares are configured with a special trigger, which clears the TMR1 register, then the compare register that is closest to (but greater than) the TMR1 register value is the compare value that will reset the TMR1 register. Example 1 shows a possible case.

## Interaction of Two PWM Modes

When two CCP modules are in a PWM mode, Timer2 is the time-base for both PWM outputs. This means that they will have the same PWM frequency and update rates, as determined by the Timer2 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.

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

## APPENDIX A: PWM\_1.ASM

MPASM 01.40 Released

PWM\_1.ASM 1-16-1997 17:35:52

PAGE 1

```

LOC  OBJECT CODE      LINE SOURCE TEXT
VALUE

00001          LIST    P = 16C74, n = 66
00002          ERRORLEVEL -302
00003 ;
00004 ;*****
00005 ;
00006 ; This program outputs a PWM signal on the CCP1 pin. The duty cycle and
00007 ; period of the PWM is read every time TMR1 overflows.
00008 ; PERIOD      = PORTB
00009 ; DUTY CYCLE = PORTD and PORTE<1:0>
00010 ;
00011 ; The prescaler of TMR2 is selected by the state of PORTA<1:0> after
00012 ; reset      RA1:RA0   Prescaler multiplies Tcyc by
00013 ;           0 0       1
00014 ;           0 1       4
00015 ;           1 x       16
00016 ;
00017 ;
00018 ; Program = PWM_1.ASM
00019 ; Revision Date: 7-13-94
00020 ;           1-16-97      Compatibility with MPASMWIN 1.40
00021 ;
00022 ;
00023 ;*****
00024 ;
00025 ;
00026 ; HARDWARE SETUP
00027 ; PORTA<1:0> - Prescaler to TMR2, read only after reset
00028 ; PORTB      - Period of PWM
00029 ; PORTD      - Duty Cycle high of PWM (8-bits)
00030 ; PORTE<1:0> - Duty Cycle low of PWM (2-bits)
00031 ;
00032 ;
00033          INCLUDE <p16c74.inc>
00001          LIST
00002 ; P16C74.INC Standard Header File, Ver. 1.00 Microchip Technology, Inc.
00318          LIST
00034
00000000     00035 FALSE          EQU      0
00000001     00036 TRUE          EQU      1
00037
00038          INCLUDE <PWM.h>
00046          list
00047
00039 ;
00000001     00040 PICMaster      EQU      TRUE      ; A Debugging Flag
00000001     00041 Debug          EQU      TRUE      ; A Debugging Flag
00000001     00042 Debug_PU       EQU      TRUE      ; A Debugging Flag
00043 ;
00044 ;
00045 ; Reset address. Determine type of RESET
00046 ;
0000          00047          org      RESET_V          ; RESET vector location
0000 1683     00048 RESET      BSF      STATUS, RP0      ; Bank 1
0001 188E     00049          BTFSC   PCON,NOT_POR      ; Power-up reset?

```

```

0002 2832      00050          GOTO    START          ; YES
0003 285C      00051          GOTO    OTHER_RESET    ; NO, a WDT or MCLR reset
00052 ;
00053 ; This is the Peripheral Interrupt routine. Need to determine the type
00054 ; of interrupt that occurred. The following interrupts are enabled:
00055 ; 1. CCP Capture Occured
00056 ;
00057          page
0004          00058          org      ISR_V          ; Interrupt vector location
0004          00059 PER_INT_V
0004 1283      00060          BCF      STATUS, RP0    ; Bank 0
0005 180C      00061          BTFSC    PIR1, TMR1IF  ; TMR1 Overflow Interrupt occurred?
0006 280D      00062          GOTO    T1OVFL        ; YES, Service the TMR1 Interrupt
0007          00063 ERROR1          ; NO, Error Condition-Unknown Interrupt
0007 1505      00064          BSF      PORTA, 2      ; Toggle a PORT pin
0008 1105      00065          BCF      PORTA, 2
0009 2807      00066          GOTO    ERROR1
00067 ;
000A          00068 ERROR2          ; NO, Error Condition-Unknown Interrupt
000A 1585      00069          BSF      PORTA, 3      ; Toggle a PORT pin
000B 1185      00070          BCF      PORTA, 3
000C 280A      00071          GOTO    ERROR2
00072 ;
000D          00073 T1OVFL
000D 100C      00074          BCF      PIR1, TMR1IF  ; Clear T1 Overflow Interrupt Flag
00075          if (PICMaster )
000E 0853      00076          MOVF     DUMMY_PD, W    ;
00077          else
00078          MOVF     PORTD, W      ;
00079          endif
000F 00D5      00080          MOVWF    DC_HI          ;
00081          if (PICMaster )
0010 0854      00082          MOVF     DUMMY_PE, W    ;
00083          else
00084          MOVF     PORTE, W      ;
00085          endif
0011 00D6      00086          MOVWF    DC_LO          ;
00087          if (PICMaster )
0012 0851      00088          MOVF     DUMMY_PB, W    ;
00089          else
00090          MOVF     PORTB, W      ;
00091          endif
0013 1683      00092          BSF      STATUS, RP0    ; Bank 1
0014 00A0      00093          MOVWF    T2_PERIOD    ;
0015 1283      00094          BCF      STATUS, RP0    ; Bank 0
00095 ;
0016          00096 WAIT_DC
0016 0811      00097          MOVF     TMR2, W        ; Read present TMR2 register value
0017 0212      00098          SUBWF    PR2, W        ; How close is the timer to rolling over
0018 390F      00099          ANDLW    0x0F          ; Does this make it zero?
0019 1903      00100          BTFSC    STATUS, Z      ; If Z is set, near rollover
001A 2816      00101          GOTO    WAIT_DC        ; loop until rolled over
001B 0855      00102          MOVF     DC_HI, W      ; else load the duty cycle values
001C 0095      00103          MOVWF    CCP1L         ; Load DC high
001D 300F      00104          MOVLW    0x0F          ;
001E 0597      00105          ANDWF    CCP1CON, F    ; Set the DC low bits
001F 18D6      00106          BTFSC    DC_LO, 1      ;
0020 1697      00107          BSF      CCP1CON, CCP1X;
0021 1856      00108          BTFSC    DC_LO, 0      ;
0022 1617      00109          BSF      CCP1CON, CCP1Y;
0023 108C      00110          BCF      PIR1, TMR2IF  ; Clear the TRM2 = PR2 flag
00111 ;
0024          00112 WAIT_PR
0024 1C8C      00113          BTFSS    PIR1, TMR2IF  ; LOOP waiting for TRM2 = PR2
0025 2824      00114          GOTO    WAIT_PR        ; Need to wait until TMR2 = PR2 so that
00115          ; Duty Cycle is latched

```



```

0026 1683      00116      BSF      STATUS, RP0      ; Bank 1
0027 300F      00117      MOVLW    0x0F              ; Load TMR2 period with minimum value Fh
0028 0092      00118      MOVWF    PR2              ;
0029 30F0      00119      MOVLW    0xF0              ;
002A 0520      00120      ANDWF    T2_PERIOD, W      ; Determine if period needs to be greater
002B 1903      00121      BTFSC    STATUS, Z          ;
002C 2830      00122      GOTO     NO_OFFSET          ; NO, Period is the minimum
002D           00123      PR_OFFSET
002D 300F      00124      MOVLW    0x0F              ; Yes, calculate additional offset
002E 0220      00125      SUBWF    T2_PERIOD, W      ;
002F 0792      00126      ADDWF    PR2, F            ; ADD Period offset
002F           00127      ;
0030           00128      NO_OFFSET
0030 1283      00129      BCF      STATUS, RP0      ; Bank 0
0031 0009      00130      RETFIE                     ; Return / Enable Global Interrupts
0031           00131      ;
0031           00132      page
0031           00133      ;
0031           00134      ;*****
0031           00135      ;*****      Start program here, Power-On Reset occurred.
0031           00136      ;*****
0031           00137      ;
0032           00138      START                      ; POWER_ON Reset (Beginning of program)
0032 1283      00139      BCF      STATUS, RP0      ; Bank 0
0033 018F      00140      CLRF     TMR1H              ;
0034 018E      00141      CLRF     TMR1L              ;
0034           00142      ;
0035           00143      MCLR_RESET                  ; A Master Clear Reset
0035 0183      00144      CLRF     STATUS            ; Do initialization (Bank 0)
0036 018B      00145      CLRF     INTCON
0037 018C      00146      CLRF     PIR1
0038 1683      00147      BSF      STATUS, RP0      ; Bank 1
0039 3080      00148      MOVLW    0x80              ;
003A 0081      00149      MOVWF    OPTION_REG        ;
003B 018C      00150      CLRF     PIE1              ; Disable all peripheral interrupts
003C 30FF      00151      MOVLW    0xFF              ;
003D 009F      00152      MOVWF    ADCON1            ; Port A is Digital.
003D           00153      ;
003D           00154      ;
003E 1283      00155      BCF      STATUS, RP0      ; Bank 0
003F 0185      00156      CLRF     PORTA              ; ALL PORT output should output Low.
0040 0186      00157      CLRF     PORTB
0041 0187      00158      CLRF     PORTC
0042 0188      00159      CLRF     PORTD
0043 0189      00160      CLRF     PORTE
0043           00161      ;
0044 1683      00162      BSF      STATUS, RP0      ; Select Bank 1
0045 30FF      00163      MOVLW    0xFF              ;
0046 0085      00164      MOVWF    TRISA              ; RA5 - 0 inputs
0047 0086      00165      MOVWF    TRISB              ; RB7 - 0 inputs
0048 0187      00166      CLRF     TRISC              ; RC Port are outputs
0049 0088      00167      MOVWF    TRISD              ; RD Port are inputs
004A 0089      00168      MOVWF    TRISE              ; RE Port are inputs
004B 0092      00169      MOVWF    PR2              ; Default PWM period
004C 140C      00170      BSF      PIE1, TMR1IE      ; Enable TMR1 Interrupt
004D 1283      00171      BCF      STATUS, RP0      ; Select Bank 0
004D           00172      ;
004E 300C      00173      MOVLW    0X0C              ; CCP module is in
004F 0097      00174      MOVWF    CCP1CON           ; PWM output mode
004F           00175      ;
004F           00176      ; Initialize the Special Function Registers (SFR) interrupts
004F           00177      ;
0050 018C      00178      CLRF     PIR1              ;
0051 0190      00179      CLRF     T1CON              ;
0052 0192      00180      CLRF     T2CON              ;
0052           00181      if (PICMaster )

```

# AN594

```
0053 1850      00182      BTFSC    DUMMY_PA, 0    ;
                00183      else
                00184      BTFSC    PORTA, 0        ;
                00185      endif
0054 1412      00186      BSF      T2CON, 0        ;
                00187      ;
                00188      if (PICMaster )
0055 18D0      00189      BTFSC    DUMMY_PA, 1    ;
                00190      else
                00191      BTFSC    PORTA, 1        ;
                00192      endif
0056 1492      00193      BSF      T2CON, 1        ;
                00194      ;
0057 170B      00195      BSF      INTCON, PEIE    ; Enable Peripheral Interrupts
0058 178B      00196      BSF      INTCON, GIE     ; Enable all Interrupts
0059 1410      00197      BSF      T1CON, TMR1ON   ; Turn Timer 1 ON
005A 1512      00198      BSF      T2CON, TMR2ON   ; Turn Timer 2 ON
                00199      ;
005B 285B      00200 lzz      goto    lzz          ; Loop waiting for TMR1 interrupt
                00201      ;
                00202      ; Here is where you do things depending on the type of RESET (Not a
                00203      ; Power-On Reset).
005C 1E03      00204 OTHER_RESET BTFSS STATUS,NOT_TO ; WDT Time-out?
005D 2807      00205 WDT_TIMEOUT GOTO    ERROR1    ; YES, This is error condition
                00206      if ( Debug_PU )
005E 2832      00207      goto    START          ; MCLR reset, Goto START
                00208      else
                00209      GOTO    MCLR_RESET        ; MCLR reset, Goto MCLR_RESET
                00210      endif
                00211      ;
                00212      if (Debug )
005F 0000      00213 END_START NOP                ; END label for debug
                00214      endif
                00215      ;
                00216      ;
07FF          00217      org      PMEM_END        ; End of Program Memory
07FF 2807      00218      GOTO    ERROR1          ; If you get here your program was lost
                00219      ;
                00220      end

__XT_OSC      00003FBD
__16C74      00000001
lzz          0000005B
```

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

```
0000 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX
0040 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX -----
07C0 : -----X
```

All other memory blocks unused.

Program Memory Words Used: 97  
Program Memory Words Free: 3999

Errors : 0  
Warnings : 0 reported, 0 suppressed  
Messages : 0 reported, 17 suppressed

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

## APPENDIX B: 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'1000000'; 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

```

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

## APPENDIX C: COMP\_1.LST

MPASM 01.40 Released

COMP\_1.ASM 1-16-1997 17:35:21

PAGE 1

```

LOC  OBJECT CODE      LINE SOURCE TEXT
VALUE

00001          LIST    P = 16C74, n = 66
00002          ERRORLEVEL  -302
00003 ;
00004 ;*****
00005 ;
00006 ; This program outputs a pulse train on the CCP1 pin, as specified by
00007 ; the values in the CCPR1H:CCPR1L.
00008 ;
00009 ;
00010 ; Pulse Train      _____
00011 ; Data Value      <---- 1 ----> <-- 0 -->
00012 ;                  ^           ^
00013 ;                  |           |
00014 ;                  T_ONE_BIT   T_ZERO_BIT
00015 ;                  |           |
00016 ;                  PULSE_TIME PULSE_TIME
00017 ;
00018 ;
00019 ;      Program = COMP_1.ASM
00020 ;      Revision Date:   7-13-94
00021 ;                      1-16-97      Compatibility with MPASMWIN 1.40
00022 ;
00023 ;
00024 ;*****
00025 ;
00026 ;
00027 ; HARDWARE SETUP
00028 ;      PORTB      - Data to serial transmit on CCP pin
00029 ;
00030 ;
00031          INCLUDE <p16c74.inc>
00001          LIST
00002 ;P16C74.INC Standard Header File, Version 1.00 Microchip Technology
00318          LIST
00032
00000000      00033 FALSE      EQU      0
00000001      00034 TRUE       EQU      1
00035
00036          INCLUDE <COMP.h>
00052          list
00053
00037 ;
00000001      00038 PICMaster EQU      TRUE      ; A Debugging Flag
00000001      00039 Debug     EQU      TRUE      ; A Debugging Flag
00000001      00040 Debug_PU  EQU      TRUE      ; A Debugging Flag
00041 ;
00042 ;
00043 ; Reset address. Determine type of RESET
00044 ;
0000      00045          org      RESET_V      ; RESET vector location
0000 1683      00046 RESET      BSF      STATUS, RP0      ; Bank 1
0001 188E      00047          BTFSC   PCON,NOT_POR      ; Power-up reset?
0002 287C      00048          GOTO    START      ; YES
0003 28BA      00049          GOTO    OTHER_RESET      ; NO, a WDT or MCLR reset

```

```

00050 ;
00051 ; This is the Periperal Interrupt routine. Need to determine the type
00052 ; of interrupt that occurred. The following interrupts are enabled:
00053 ; 1. CCP Capture Occured
00054 ;
00055     page
0004     00056     org     ISR_V             ; Interrupt vector location
0004     00057 PER_INT_V
00058         if ( Debug )
0004 1405     00059             bsf     PORTA, 0         ; Turn on strobe
00060             endif
0005 1283     00061             BCF     STATUS, RP0         ; Bank 0
0006 190C     00062             BTFSC    PIR1, CCP1IF         ; Compare Interrupt occurred?
0007 280E     00063             GOTO    CCP1_INT         ; YES, Service the TMR1 Interrupt
0008         00064 ERROR1
0008 1505     00065             BSF     PORTA, 2         ; Toggle a PORT pin
0009 1105     00066             BCF     PORTA, 2
000A 2808     00067             GOTO    ERROR1
00068 ;
000B         00069 ERROR2
000B 1585     00070             BSF     PORTA, 3         ; Toggle a PORT pin
000C 1185     00071             BCF     PORTA, 3
000D 280B     00072             GOTO    ERROR2
00073 ;
00074 ;
00075 ;*****
00076 ; In the CCP interrupt.
00077 ; Since timer1 is not cleared on a CCP match, the value in the
00078 ; CCP1H:CCP1L register pair must be updated. This is done with
00079 ; a 16-bit add. Also after the 1st CCP1 match (CCP1 pin goes high) the
00080 ; next match will force it low. Depending on the value of the data bit
00081 ; determines the value add to the CCP1H:CCP1L register pair.
00082 ;
00083 ; After the data has been transmitted, the pin will have a sync pulse
00084 ; and then remain low for 300 us.
00085 ;*****
00086 ;
00087
000E         00088 CCP1_INT
000E 110C     00089             BCF     PIR1, CCP1IF         ; Clear CCP1 Interrupt Flag
000F 0AB3     00090             INCF     CCP1_INT_CNT, F ;
0010 1C33     00091             BTFSS    CCP1_INT_CNT, 0 ;
0011 2833     00092             GOTO    SYNC_PULSE
0012         00093 DATA_PULSE
0012 03B1     00094             DECF     DATA_CNT, F         ; Decrement the Count of data bits
0013 1903     00095             BTFSC    STATUS, Z         ; Have we transmitted all the Data Bits?
0014 2827     00096             GOTO    PERIOD_DELTA         ; YES, Delay to 300 us
0015 0DB0     00097             RLF     XMIT_DATA, F         ; NO, get next bit to transmit
0016 1803     00098             BTFSC    STATUS, C         ; Is the bit to transmit a '1'?
0017 281F     00099             GOTO    ONE_DATA         ; YES, Stay low for 17.6 us
0018         00100 ZERO_DATA
0018 302F     00101             MOVLW    LOW ( T_ZERO_BIT ) ; NO, Stay low for 8.8 us
0019 0795     00102             ADDWF    CCP1L, F         ; Update Compare register pair latch
001A 1803     00103             BTFSC    STATUS, C         ;
001B 0A96     00104             INCF     CCP1H, F         ;
001C 3000     00105             MOVLW    HIGH ( T_ZERO_BIT ) ;
001D 0796     00106             ADDWF    CCP1H, F         ;
001E 287A     00107             GOTO    RET_FIE
00108 ;
001F         00109 ONE_DATA
001F 305E     00110             MOVLW    LOW ( T_ONE_BIT ) ; Stay low for 17.6 us
0020 0795     00111             ADDWF    CCP1L, F         ; Update Compare register pair latch
0021 1803     00112             BTFSC    STATUS, C         ;
0022 0A96     00113             INCF     CCP1H, F         ;
0023 3000     00114             MOVLW    HIGH ( T_ONE_BIT ) ;
0024 0796     00115             ADDWF    CCP1H, F         ;

```

# AN594

```
0025 0AB2      00116      INCF    ONES_CNT, F      ; Increment the number of 1's in the byte
0026 287A      00117      GOTO    RET_FIE
                                00118 ;
0027           00119 PERIOD_DELTA
0027 0832      00120      MOVF    ONES_CNT, W      ;
0028 390F      00121      ANDLW   0x0F            ; Only want 9 states (0 1s to 8 1s)
0029 0782      00122      ADDWF   PCL, F          ;
002A 283B      00123      GOTO    ZERO_1          ; There was 0 ones in the data byte
002B 2842      00124      GOTO    ONE_1           ; There was 1 one in the data byte
002C 2849      00125      GOTO    TWO_1          ; There was 2 ones in the data byte
002D 2850      00126      GOTO    THREE_1         ; There was 3 ones in the data byte
002E 2857      00127      GOTO    FOUR_1          ; There was 4 ones in the data byte
002F 285E      00128      GOTO    FIVE_1          ; There was 5 ones in the data byte
0030 2865      00129      GOTO    SIX_1           ; There was 6 ones in the data byte
0031 286C      00130      GOTO    SEVEN_1          ; There was 7 ones in the data byte
0032 2873      00131      GOTO    EIGHT_1         ; There was 8 ones in the data byte
                                00132 ;
0033           00133 SYNC_PULSE
0033 302F      00134      MOVLW   LOW ( PULSE_TIME ) ;Update Compare register pair latch
0034 0795      00135      ADDWF   CCP1L, F          ;
0035 1803      00136      BTFSC   STATUS, C          ;
0036 0A96      00137      INCF    CCP1H, F          ;
0037 3000      00138      MOVLW   HIGH ( PULSE_TIME );
0038 0796      00139      ADDWF   CCP1H, F          ;
0039 1417      00140      BSF     CCP1CON, 0         ; On Compare match, CCP1 pin = L
003A 0009      00141      RETFIE
                                00142 ;
003B           00143 ZERO_1
003B 30EC      00144      MOVLW   LOW ( ZERO_1S )    ; Update Compare register pair latch
003C 0795      00145      ADDWF   CCP1L, F          ;
003D 1803      00146      BTFSC   STATUS, C          ;
003E 0A96      00147      INCF    CCP1H, F          ;
003F 3002      00148      MOVLW   HIGH ( ZERO_1S )   ;
0040 0796      00149      ADDWF   CCP1H, F          ;
0041 287A      00150      GOTO    RET_FIE
                                00151 ;
0042           00152 ONE_1
0042 30BD      00153      MOVLW   LOW ( ONE_1S )     ; Update Compare register pair latch
0043 0795      00154      ADDWF   CCP1L, F          ;
0044 1803      00155      BTFSC   STATUS, C          ;
0045 0A96      00156      INCF    CCP1H, F          ;
0046 3002      00157      MOVLW   HIGH ( ONE_1S )    ;
0047 0796      00158      ADDWF   CCP1H, F          ;
0048 287A      00159      GOTO    RET_FIE
                                00160 ;
0049           00161 TWO_1
0049 308E      00162      MOVLW   LOW ( TWO_1S )     ; Update Compare register pair latch
004A 0795      00163      ADDWF   CCP1L, F          ;
004B 1803      00164      BTFSC   STATUS, C          ;
004C 0A96      00165      INCF    CCP1H, F          ;
004D 3002      00166      MOVLW   HIGH ( TWO_1S )    ;
004E 0796      00167      ADDWF   CCP1H, F          ;
004F 287A      00168      GOTO    RET_FIE
                                00169 ;
0050           00170 THREE_1
0050 305F      00171      MOVLW   LOW ( THREE_1S )   ; Update Compare register pair latch
0051 0795      00172      ADDWF   CCP1L, F          ;
0052 1803      00173      BTFSC   STATUS, C          ;
0053 0A96      00174      INCF    CCP1H, F          ;
0054 3002      00175      MOVLW   HIGH ( THREE_1S )  ;
0055 0796      00176      ADDWF   CCP1H, F          ;
0056 287A      00177      GOTO    RET_FIE
                                00178 ;
0057           00179 FOUR_1
0057 3030      00180      MOVLW   LOW ( FOUR_1S )    ; Update Compare register pair latch
0058 0795      00181      ADDWF   CCP1L, F          ;
```

```

0059 1803      00182      BTFSC    STATUS, C      ;
005A 0A96      00183      INCF     CCPR1H, F      ;
005B 3002      00184      MOVLW    HIGH ( FOUR_1S );
005C 0796      00185      ADDWF    CCPR1H, F      ;
005D 287A      00186      GOTO     RET_FIE
                00187 ;
005E          00188 FIVE_1
005E 3001      00189      MOVLW    LOW ( FIVE_1S ) ; Update Compare register pair latch
005F 0795      00190      ADDWF    CCPR1L, F      ;
0060 1803      00191      BTFSC    STATUS, C      ;
0061 0A96      00192      INCF     CCPR1H, F      ;
0062 3002      00193      MOVLW    HIGH ( FIVE_1S ;
0063 0796      00194      ADDWF    CCPR1H, F      ;
0064 287A      00195      GOTO     RET_FIE
                00196 ;
0065          00197 SIX_1
0065 30D2      00198      MOVLW    LOW ( SIX_1S ) ; Update Compare register pair latch
0066 0795      00199      ADDWF    CCPR1L, F      ;
0067 1803      00200      BTFSC    STATUS, C      ;
0068 0A96      00201      INCF     CCPR1H, F      ;
0069 3001      00202      MOVLW    HIGH ( SIX_1S ) ;
006A 0796      00203      ADDWF    CCPR1H, F      ;
006B 287A      00204      GOTO     RET_FIE
                00205 ;
006C          00206 SEVEN_1
006C 30A3      00207      MOVLW    LOW ( SEVEN_1S ) ; Update Compare register pair latch
006D 0795      00208      ADDWF    CCPR1L, F      ;
006E 1803      00209      BTFSC    STATUS, C      ;
006F 0A96      00210      INCF     CCPR1H, F      ;
0070 3001      00211      MOVLW    HIGH ( SEVEN_1S ) ;
0071 0796      00212      ADDWF    CCPR1H, F      ;
0072 287A      00213      GOTO     RET_FIE
                00214 ;
0073          00215 EIGHT_1
0073 3074      00216      MOVLW    LOW ( EIGHT_1S ) ; Update Compare register pair latch
0074 0795      00217      ADDWF    CCPR1L, F      ;
0075 1803      00218      BTFSC    STATUS, C      ;
0076 0A96      00219      INCF     CCPR1H, F      ;
0077 3001      00220      MOVLW    HIGH ( EIGHT_1S ) ;
0078 0796      00221      ADDWF    CCPR1H, F      ;
0079 287A      00222      GOTO     RET_FIE
                00223 ;
007A          00224 RET_FIE
007A 1017      00225      BCF      CCP1CON, 0      ; On Compare match, CCP1 pin = H
007B 0009      00226      RETFIE      ; Return / Enable Global Interrupts
                00227 ;
                00228      page
                00229 ;
                00230 ;*****
                00231 ;*****      Start program here, Power-On Reset occurred.
                00232 ;*****
                00233 ;
007C          00234 START      ; POWER_ON Reset (Beginning of program)
007C 1283      00235      BCF      STATUS, RP0      ; Bank 0
007D 018F      00236      CLRF     TMR1H      ;
007E 018E      00237      CLRF     TMR1L      ;
                00238 ;
007F          00239 MCLR_RESET      ; A Master Clear Reset
007F 1283      00240      BCF      STATUS, RP0      ; Bank 0
0080 0183      00241      CLRF     STATUS      ; Do initialization (Bank 0)
0081 018B      00242      CLRF     INTCON
0082 018C      00243      CLRF     PIR1
0083 1683      00244      BSF      STATUS, RP0      ; Bank 1
0084 3080      00245      MOVLW    0x80      ; Disable PORTB weak pull-ups
0085 0081      00246      MOVWF    OPTION_REG      ;
0086 018C      00247      CLRF     PIE1      ; Disable all peripheral interrupts

```

```

0087 30FF      00248      MOVLW    0xFF      ;
0088 009F      00249      MOVWF    ADCON1     ; Port A is Digital.
                00250 ;
                00251 ;
0089 1283      00252      BCF      STATUS, RP0    ; Bank 0
008A 0185      00253      CLRF     PORTA         ; ALL PORT output should output Low.
008B 0186      00254      CLRF     PORTB
008C 0187      00255      CLRF     PORTC
008D 0188      00256      CLRF     PORTD
008E 0189      00257      CLRF     PORTE
008F 1010      00258      BCF      T1CON, TMR1ON   ; Timer 1 is NOT incrementing
                00259 ;
0090 1683      00260      BSF      STATUS, RP0    ; Select Bank 1
0091 0185      00261      CLRF     TRISA         ; RA5 - 0 outputs
0092 30FF      00262      MOVLW    0xFF      ;
0093 0086      00263      MOVWF    TRISB         ; RB Port are inputs
0094 0187      00264      CLRF     TRISC         ; RC Port are outputs
0095 0188      00265      CLRF     TRISD         ; RD Port are outputs
0096 0189      00266      CLRF     TRISE         ; RE Port are outputs
0097 150C      00267      BSF      PIE1, CCP1IE    ; Enable CCP1 Interrupt
0098 1283      00268      BCF      STATUS, RP0    ; Select Bank 0
                00269 ;
                00270      page
                00271 ;
                00272 ;
                00273 ; Initialize the Special Function Registers (SFR) interrupts
                00274 ;
0099 018C      00275      CLRF     PIR1         ;
009A 0190      00276      CLRF     T1CON         ; Timer mode
009B 170B      00277      BSF      INTCON, PEIE    ; Enable Peripheral Interrupts
009C 178B      00278      BSF      INTCON, GIE     ; Enable all Interrupts
                00279 ;
                00280 ; Set-up timer and compare latches and then turn timer1 on.
                00281 ;
009D 1010      00282      BCF      T1CON, TMR1ON   ; Turn OFF timer1
009E 3041      00283      MOVLW    CCPREG_HI      ; TMR1 = CCPR1H:CCPR1L - 1
009F 008F      00284      MOVWF    TMR1H         ;
00A0 3042      00285      MOVLW    CCPREG_LO      ;
00A1 008E      00286      MOVWF    TMR1L         ;
00A2 038E      00287      DECF     TMR1L, F       ;
00A3 1803      00288      BTFSC    STATUS, C       ;
00A4 038F      00289      DECF     TMR1H, F       ;
00A5 3008      00290      MOVLW    0x08         ; On match CCP1 = H level
00A6 0097      00291      MOVWF    CCP1CON        ;
00A7 3009      00292      MOVLW    0x09         ;
00A8 00B1      00293      MOVWF    DATA_CNT      ; 8-bits to transfer
00A9 01B2      00294      CLRF     ONES_CNT       ; Result after xmit holds the number
00AA 30FF      00295      MOVLW    0xFF         ; of 1's in a byte
00AB 00B3      00296      MOVWF    CCP1_INT_CNT    ; No CCP1 transmit interrupts yet
00AC 1410      00297      BSF      T1CON, TMR1ON   ; Turn ON timer1
                00298 ;
                00299 ;
                00300 ; This code segment is an infinite loop that will always transmit the data
                00301 ; contained in the XMIT_DATA register. After each byte is transmitted a
                00302 ; new byte is read. If using PICMASTER (in stand alone mode), this is
                00303 ; read from a register that is updated after a break (at NOP). If in a
                00304 ; system, PORTB is read. All other variables are reinitialized after each
                00305 ; byte.
00AD           00306 NEXT_BYTE
                00307 ;
00AD 0831      00308 WAIT      MOVF     DATA_CNT, w    ;
00AE 1D03      00309          BTFSS    STATUS, Z       ; Is DATA_CNT = 0 ?
00AF 28AD      00310          GOTO     WAIT          ; NO, must wait until YES
00B0 0000      00311          NOP                    ;
                00312      if ( Debug )
00B1 1005      00313          bcf      PORTA, 0       ; Turn off strobe

```



```

00314         endif
00315
00316         if ( PICMaster )
00B2 0840     00317             MOVF     DUMMY_PB, W    ;
00318         else
00319             MOVF     PORTB, W        ;
00320         endif
00B3 00B0     00321             MOVWF    XMIT_DATA    ; New data to transmit
00B4 30FF     00322             MOVLW    0xFF          ;
00B5 00B3     00323             MOVWF    CCP1_INT_CNT    ;
00B6 3009     00324             MOVLW    0x09          ;
00B7 00B1     00325             MOVWF    DATA_CNT    ;
00B8 01B2     00326             CLRF     ONES_CNT      ;
00B9 28AD     00327             GOTO     NEXT_BYTE     ;
00328 ;
00329 ;
00330 ; Here is where you do things depending on the type of RESET (Not a
00331 ; Power-On Reset).
00BA 1E03     00332 OTHER_RESET  BTFSS    STATUS,NOT_TO ; WDT Time-out?
00BB 2808     00333 WDT_TIMEOUT  GOTO     ERROR1      ; YES, This is error condition
00334         if ( Debug_PU )
00BC 287C     00335             goto     START          ; MCLR reset, Goto START
00336         else
00337             GOTO     MCLR_RESET      ; MCLR reset, Goto MCLR_RESET
00338         endif
00339 ;
00340         if ( Debug )
00BD 0000     00341 END_START    NOP          ; END label for debug
00342         endif
00343 ;
00344 ;
07FF         00345         org      PMEM_END          ; End of Program Memory
07FF 2808     00346             GOTO     ERROR1          ; If you get here your program was lost
00347
00348         end
0000 : XXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXX
0040 : XXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXX
0080 : XXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXX--
07C0 : -----X

```

All other memory blocks unused.

```

Program Memory Words Used:   191
Program Memory Words Free:  3905
Errors      :      0
Warnings   :      0 reported,      0 suppressed
Messages   :      0 reported,     10 suppressed

```

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

## APPENDIX D:

```
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'1000000'; Device Frequency is 10 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
```

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

## APPENDIX E:

MPASM 01.40 Released

CAPT\_2.ASM 1-16-1997 17:34:47

PAGE 1

```

LOC  OBJECT CODE      LINE SOURCE TEXT
VALUE
                                00001      LIST      P = 16C74, n = 66
                                00002      ERRORLEVEL  -302
                                00003 ;
                                00004 ;*****
                                00005 ;
                                00006 ; This program implements a real time clock using the TMR1 module of the
                                00007 ; PIC16CXXX family.
                                00008 ;
                                00009 ;      Program = CAPT_2.ASM
                                00010 ;      Revision Date:   7-19-94
                                00011 ;                      1-16-97      Compatibility with MPASMWIN 1.40
                                00012 ;
                                00013 ;
                                00014 ;*****
                                00015 ;
                                00016 ;
                                00017 ; HARDWARE SETUP
                                00018 ;
                                00019 ;      CCP2 Compare Output
                                00020 ;      CCP1 Capture Input
                                00021 ;      CCP2 ---> CCP1
                                00022 ;
                                00023 ;
                                00024      INCLUDE <p16c74.inc>
                                00001      LIST
                                00002 ; P16C74.INC Standard Header File, Ver. 1.00 Microchip Technology, Inc.
                                00318      LIST
                                00025
                                00000000 00026 FALSE      EQU      0
                                00000001 00027 TRUE       EQU      1
                                00028
                                00029      INCLUDE <CAPT.h>
                                00052      list
                                00030 ;
                                00031 ;
                                00000001 00032 PICMaster    EQU      TRUE      ; A Debugging Flag
                                00000001 00033 Debug        EQU      TRUE      ; A Debugging Flag
                                00000001 00034 Debug_PU      EQU      TRUE      ; A Debugging Flag
                                00035 ;
                                00036 ;
                                00037 ; Reset address. Determine type of RESET
                                00038 ;
                                0000      00039      org      RESET_V      ; RESET vector location
                                0000 1683 00040 RESET      BSF      STATUS, RP0      ; Bank 1
                                0001 188E 00041      BTFSC     PCON,NOT_POR      ; Power-up reset?
                                0002 282F 00042      GOTO     START      ; YES
                                0003 2861 00043      GOTO     OTHER_RESET      ; NO, a WDT or MCLR reset
                                00044 ;
                                00045 ; This is the Periperal Interrupt routine. Need to determine the type
                                00046 ; of interrupt that occurred. The following interrupts are enabled:
                                00047 ; 1. CCP1 Capture Occurred
                                00048 ; 2. CCP2 Compare Occurred
                                00049 ;
                                00050      page

```

```

0004          00051      org      ISR_V          ; Interrupt vector location
0004          00052 PER_INT_V
0004 1283      00053          BCF      STATUS, RP0      ; Bank 0
0005 190C      00054          BTFSC   PIR1, CCP1IF     ; CCP1 Interrupt occurred? (Capture)
0006 281D      00055          GOTO    CAPTURE         ; YES, Service the CCP1 Interrupt
0007 180D      00056          BTFSC   PIR2, CCP2IF     ; CCP2 Interrupt occurred? (Compare)
0008 2811      00057          GOTO    COMPARE         ; YES, Service the CCP2 Interrupt
0009 180C      00058          BTFSC   PIR1, TMR1IF     ; NO, Timer 1 Overflow?
000A 282D      00059          GOTO    T1OVFL          ; YES,
000B          00060 ERROR1          ; NO, Error Condition-Unknown Interrupt
000B 1488      00061          BSF      PORTD, 1        ; Toggle a PORT pin
000C 1088      00062          BCF      PORTD, 1
000D 280B      00063          GOTO    ERROR1
000E          00064 ;
000E          00065 ERROR2          ; NO, Error Condition-Unknown Interrupt
000E 1508      00066          BSF      PORTD, 2        ; Toggle a PORT pin
000F 1108      00067          BCF      PORTD, 2
0010 280E      00068          GOTO    ERROR2
0010          00069 ;
0010          00070 ; The Compare generates a Square wave based on the value on PORTB (in
0010          00071 ; DUMMY_PB) and on PORTD (in DUMMY_PD). PORTB is loaded into low compare
0010          00072 ; latch and PORTD is loaded into the high compare latch. If the value of
0010          00073 ; the ports is not changed, a capture overflow condition will occur when
0010          00074 ; PORTD:PORTB > 7Fh. This overflow is only indicated by the time between
0010          00075 ; captures being much less then expected.
0011          00076 COMPARE
0011 100D      00077          BCF      PIR2, CCP2IF     ; Clear CCP2 Interrupt Flag
0011          00078          if ( PICMaster )
0012 0851      00079          MOVF     DUMMY_PB, W      ;
0012          00080          else
0012          00081          MOVF     PORTB, W          ;
0012          00082          endif
0013 079B      00083          ADDWF   CCPR2L, F        ; Update Compare register pair latch
0014 1803      00084          BTFSC   STATUS, C        ;
0015 0A9C      00085          INCF     CCPR2H, F        ;
0015          00086          if ( PICMaster )
0016 0853      00087          MOVF     DUMMY_PD, W      ;
0016          00088          else
0016          00089          MOVF     PORTD, W          ;
0016          00090          endif
0017 079C      00091          ADDWF   CCPR2H, F        ;
0018 0AB3      00092          INCF     CCP2_INT_CNT, F   ;
0019 141D      00093          BSF      CCP2CON, 0        ; On Compare match, CCP2 pin = L
001A 1C33      00094          BTFSS   CCP2_INT_CNT, 0   ;
001B 101D      00095          BCF      CCP2CON, 0        ; On Compare match, CCP2 pin = H
001C          00096 END_COMPARE
001C 0009      00097          RETFIE          ; Return / Enable Global Interrupts
001C          00098
001C          00099 ;
001C          00100 ; The result of the new capture minus the old capture is stored in the
001C          00101 ; new capture registers (CAPT_NEW_H:CAPT_NEW_L)
001C          00102 ;
001D          00103 CAPTURE
001D 110C      00104          BCF      PIR1, CCP1IF     ; Clear CCP1 Interrupt Flag
001E 0815      00105          MOVF     CCPR1L, W        ; New capture value (low byte)
001F 00C1      00106          MOVWF   CAPT_NEW_L        ;
0020 0816      00107          MOVF     CCPR1H, W        ; New capture value (high byte)
0021 00C0      00108          MOVWF   CAPT_NEW_H        ;
0021          00109 ;
0022 0843      00110          MOVF     CAPT_OLD_L, W     ;
0023 02C1      00111          SUBWF   CAPT_NEW_L, F     ; Subtract the low bytes of the 2 captures
0024 1C03      00112          BTFSS   STATUS, C        ; Did a borrow occur?
0025 03C0      00113          DECF     CAPT_NEW_H, F     ; YES, Decrement old capture (high byte)
0026 0842      00114          MOVF     CAPT_OLD_H, W     ; New capture value (low byte)
0027 02C0      00115          SUBWF   CAPT_NEW_H, F     ; Subtract the low bytes of the 2 captures
0028 0815      00116 LOAD_OLD      MOVF     CCPR1L, W     ; New capture value (low byte)

```

```

0029 00C3      00117      MOVWF    CAPT_OLD_L      ;
002A 0816      00118      MOVF     CCP1RH, W      ; New capture value (high byte)
002B 00C2      00119      MOVWF    CAPT_OLD_H      ;
002C           00120      END_CAPTURE
002C 0009      00121      RETFIE
                00122      ;
                00123      ;
002D           00124      T1OVFL
002D 100C      00125      BCF      PIR1, TMR1IF    ; Clear T1 Overflow Interrupt Flag
002E 0009      00126      RETFIE                  ; Return / Enable Global Interrupts
                00127      ;
                00128      ;
                00129      ;*****
00130 ;*****      Start program here, Power-On Reset occurred.
00131 ;*****
00132 ;
002F           00133      START                    ; POWER_ON Reset (Beginning of program)
002F 1283      00134      BCF      STATUS, RP0      ; Bank 0
0030 018F      00135      CLRF     TMR1H            ;
0031 018E      00136      CLRF     TMR1L            ;
                00137      ;
0032           00138      MCLR_RESET                ; A Master Clear Reset
0032 1283      00139      BCF      STATUS, RP0      ; Bank 0
0033 0183      00140      CLRF     STATUS            ; Do initialization (Bank 0)
0034 018B      00141      CLRF     INTCON
0035 018C      00142      CLRF     PIR1
0036 1683      00143      BSF      STATUS, RP0      ; Bank 1
0037 3000      00144      MOVLW    0x00            ; The LCD module does not like to work w/
0038 0081      00145      MOVWF    OPTION_REG        ; weak pull-ups
0039 018C      00146      CLRF     PIE1            ; Disable all peripheral interrupts
003A 018D      00147      CLRF     PIE2            ; Disable all peripheral interrupts
003B 30FF      00148      MOVLW    0xFF            ;
003C 009F      00149      MOVWF    ADCON1          ; Port A is Digital.
                00150      ;
                00151      ;
003D 1283      00152      BCF      STATUS, RP0      ; Bank 0
003E 0185      00153      CLRF     PORTA            ; ALL PORT output should output Low.
003F 0186      00154      CLRF     PORTB
0040 0187      00155      CLRF     PORTC
0041 0188      00156      CLRF     PORTD
0042 0189      00157      CLRF     PORTE
0043 1010      00158      BCF      T1CON, TMR1ON    ; Timer 1 is NOT incrementing
                00159      ;
0044 1683      00160      BSF      STATUS, RP0      ; Select Bank 1
0045 0185      00161      CLRF     TRISA            ; RA5 - 0 outputs
0046 30FF      00162      MOVLW    0xFF            ;
0047 0086      00163      MOVWF    TRISB            ; RB7 - 0 inputs
0048 0187      00164      CLRF     TRISC            ; RC Port are outputs
0049 1507      00165      BSF      TRISC, 2          ; CCP1 is an INPUT
004A 0088      00166      MOVWF    TRISD            ; RD Port are inputs
004B 0189      00167      CLRF     TRISE            ; RE Port are outputs
004C 150C      00168      BSF      PIE1, CCP1IE     ; Enable CCP1 Interrupt
004D 140D      00169      BSF      PIE2, CCP2IE     ; Enable CCP2 Interrupt
004E 1283      00170      BCF      STATUS, RP0      ; Select Bank 0
                00171      ;
                00172      ;
00173 ; Initialize the Special Function Registers (SFR) interrupts
00174 ;
004F 018C      00175      CLRF     PIR1            ;
0050 018D      00176      CLRF     PIR2            ;
0051 0190      00177      CLRF     T1CON            ; Timer mode
0052 170B      00178      BSF      INTCON, PEIE     ; Enable Peripheral Interrupts
0053 178B      00179      BSF      INTCON, GIE      ; Enable all Interrupts
                00180      ;
                00181      ; Set-up timer and compare latches and then turn timer1 on.
                00182      ;

```

```

0054 1010      00183      BCF      T1CON, TMR1ON      ; Turn OFF timer1
                00184      if ( PICMaster )
0055 0851      00185      MOVF     DUMMY_PB, W        ;
                00186      else
                00187      MOVF     PORTB, W          ;
                00188      endif
0056 079B      00189      ADDWF    CCPR2L, F          ; Update Compare register pair latch
0057 1803      00190      BTFSC    STATUS, C          ;
0058 0A9C      00191      INCF     CCPR2H, F          ;
                00192      if ( PICMaster )
0059 08D3      00193      MOVF     DUMMY_PD, F        ;
                00194      else
                00195      MOVF     PORTD, W          ;
                00196      endif
005A 079C      00197      ADDWF    CCPR2H, F          ;
005B 3008      00198      MOVLW    0x08              ; On match CCP2 = H level
005C 009D      00199      MOVWF    CCP2CON           ;
005D 3005      00200      MOVLW    0x05              ; Capture on every rising edge
005E 0097      00201      MOVWF    CCP1CON           ;
005F 1410      00202      BSF      T1CON, TMR1ON      ; Turn ON timer1
                00203      ;
                00204      ;
                00205      ;
0060 2860      00206      lzz      goto      lzz              ; Loop waiting for interrupts (for use
                00207                        ; with PICMASTER)
                00208      ;
                00209      ; Here is where you do things depending on the type of RESET (Not a
                00210      ; Power-On Reset).
0061 1E03      00211      OTHER_RESET  BTFSS    STATUS,NOT_TO ; WDT Time-out?
0062 280B      00212      WDT_TIMEOUT  GOTO     ERROR1        ; YES, This is error condition
                00213      if ( Debug_PU )
0063 282F      00214      goto      START              ; MCLR reset, Goto START
                00215      else
                00216      GOTO      MCLR_RESET          ; MCLR reset, Goto MCLR_RESET
                00217      endif
                00218      ;
                00219      if (Debug )
0064 0000      00220      END_START    NOP              ; END lable for debug
                00221      endif
                00222      ;
                00223      ;
07FF          00224      org      PMEM_END            ; End of Program Memory
07FF 280B      00225      GOTO      ERROR1            ; If you get here your program was lost
                00226
                00227      end
0000 : XXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXX
0040 : XXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXX XXXXX-----
07C0 : -----X

```

All other memory blocks unused.

```

Program Memory Words Used:   102
Program Memory Words Free:  3994
Errors      :      0
Warnings    :      0 reported,      0 suppressed
Messages    :      0 reported,     13 suppressed

```

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

## APPENDIX F:

```

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

```

---

**Note the following details of the code protection feature on PICmicro® MCUs.**

- The PICmicro family meets the specifications contained in the Microchip Data Sheet.
- Microchip believes that its family of PICmicro microcontrollers is one of the most secure products of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the PICmicro microcontroller in a manner outside the operating specifications contained in the data sheet. The person doing so may be engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as “unbreakable”.
- Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our product.

If you have any further questions about this matter, please contact the local sales office nearest to you.

---

Information contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. No representation or warranty is given and no liability is assumed by Microchip Technology Incorporated with respect to the accuracy or use of such information, or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Microchip's products as critical components in life support systems is not authorized except with express written approval by Microchip. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights.

#### **Trademarks**

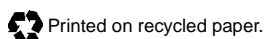
The Microchip name and logo, the Microchip logo, FilterLab, KEELOQ, microID, MPLAB, PIC, PICmicro, PICMASTER, PICSTART, PRO MATE, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

dsPIC, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, microPort, Migratable Memory, MPASM, MPLIB, MPLINK, MPSIM, MXDEV, PICC, PICDEM, PICDEM.net, rfPIC, Select Mode and Total Endurance are trademarks of Microchip Technology Incorporated in the U.S.A.

Serialized Quick Turn Programming (SQTP) is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

© 2002, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.



*Microchip received QS-9000 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona in July 1999. The Company's quality system processes and procedures are QS-9000 compliant for its PICmicro® 8-bit MCUs, KEELOQ® code hopping devices, Serial EEPROMs and microperipheral products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001 certified.*





## 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: 65-334-8870 Fax: 65-334-8850

### 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 høj 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 - 1er 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