INTRODUCTION

PIC16C64/74 microcontrollers from Microchip Technology Inc. can be interfaced with ease into a multi-microprocessor environment using its built-in Parallel Slave Port (PSP). With their very high operating speeds (cycle times as low as 200 ns with a clock rate of 20 MHz), and an array of on-chip peripherals, these microcontrollers make ideal smart interfaces to the real world.

IMPLEMENTATION

PORTD operates as an 8-bit wide Parallel Slave Port, with PORTE providing the control signals. In parallel slave mode, PORTD is asynchronously readable and writable by the external world through the chip select (RE2/CS), Read (RE0/RD), and Write (RE1/WR) control inputs.

In order to use the Parallel Slave Port, the data direction bits in the TRISE register corresponding to RD, WR, and CS (TRISE<2:0>) must be configured as inputs (set = 1) and control bit PSPMODE (TRISE<4>) must be set.

The port pins are connected to two 8-bit latches, one for data output (from the PIC16CXXX) and one for data input. The PIC16CXXX sends data by writing to the output latch, and receives data by reading the input latch (note that the input and output latches are at the same address). In PSP mode the TRISD register is ignored, since the external device connected to the slave port controls the direction of data flow.

When the external device performs either a read or a write operation to the PIC16CXXX, interrupt flag, PSPIF (PIR1<7>), will be set and the processor interrupted if bit PSPIE (PIE1<7>) is set and interrupts are enabled (enable bits GIE and PEIE, (INTCON<7:6>) set). When the interrupt is serviced, bit PSPIF must be cleared by software.

The read-only status flag bit IBF, Input Buffer Full (TRISE<7>), is set if a received word is waiting to be read. Bit IBF is cleared upon read of the input buffer latch. If another word is received prior to the first being read, status flag bit IBOV (TRISE<5>) is set. Bit IBOV can be cleared by software.

The Output Buffer Full status bit, OBF (TRISE<6>), is set if a word written to PORTD latch is waiting to be read by the external bus.

When not in Parallel Slave Port mode the IBF and OBF bits are cleared. If flag bit IBOV was previously set, however, it must be cleared by software.

Note that the following registers are for a PIC16C74 and not all peripherals are available on the PIC16C64.

<table>
<thead>
<tr>
<th>Register Name</th>
<th>Function</th>
<th>Address</th>
<th>Power-on Reset Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>PORTD</td>
<td>Parallel slave port Read/Write Data</td>
<td>08h</td>
<td>xxxxx xxxxx</td>
</tr>
<tr>
<td>TRISD</td>
<td>PORTD data direction register</td>
<td>88h</td>
<td>1111 1111</td>
</tr>
<tr>
<td>PORTE</td>
<td>Read/Write/Chip Select signals</td>
<td>09h</td>
<td>----- -xxx</td>
</tr>
<tr>
<td>TRISE</td>
<td>Control bits for PORTD slave port</td>
<td>89h</td>
<td>0000 -111</td>
</tr>
<tr>
<td>INTCON</td>
<td>peripheral and global interrupt enable bits</td>
<td>0Bh</td>
<td>0000 000x</td>
</tr>
<tr>
<td>PIR1</td>
<td>Interrupt register (PSPIF bit)</td>
<td>0Ch</td>
<td>0000 0000</td>
</tr>
<tr>
<td>PIE1</td>
<td>Interrupt Enable register (PSPIE bit)</td>
<td>8Ch</td>
<td>0000 0000</td>
</tr>
</tbody>
</table>
**TABLE 2: PORTE FUNCTIONS**

<table>
<thead>
<tr>
<th>Name</th>
<th>Bit#</th>
<th>Buffer Type</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>RE0/RD/AN5</td>
<td>bit0</td>
<td>ST/TTL(1)</td>
<td>Input/output port pin or read control input in parallel slave port mode or analog input:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>RD</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 = Not a read operation</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 = Read operation. Reads PORTD register (if chip selected)</td>
</tr>
<tr>
<td>RE1/WR/AN6</td>
<td>bit1</td>
<td>ST/TTL(1)</td>
<td>Input/output port pin or write control input in parallel slave port mode or analog input:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>WR</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 = Not a write operation</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 = Write operation. Writes PORTD register (if chip selected)</td>
</tr>
<tr>
<td>RE2/CS/AN7</td>
<td>bit2</td>
<td>ST/TTL(1)</td>
<td>Input/output port pin or chip select control input in parallel slave port mode or analog input:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>CS</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 = Device is not selected</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 = Device is selected</td>
</tr>
</tbody>
</table>

Legend:  ST = Schmitt Trigger input  TTL = TTL input  
Note 1: Input buffers are Schmitt Triggers when in I/O mode and TTL buffers when in Parallel Slave Port Mode.

**FIGURE 1: TRISE REGISTER**

```
<table>
<thead>
<tr>
<th>IBF</th>
<th>OBF</th>
<th>IBOV</th>
<th>PSPMODE</th>
<th>Bit2</th>
<th>Bit1</th>
<th>Bit0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bit7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bit6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bit5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bit4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bit3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bit2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bit1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bit0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

bit 7: **IBF:** Input Buffer Full Status bit  
1 = A word has been received and waiting to be read by the CPU  
0 = No word has been received  

bit 6: **OBF:** Output Buffer Full Status bit  
1 = The output buffer still holds a previously written word  
0 = The output buffer has been read  

bit 5: **IBOV:** Input Buffer Overflow Detect bit (in microprocessor mode)  
1 = A write occurred when a previously input word has not been read (must be cleared in software)  
0 = No overflow occurred  

bit 4: **PSPMODE:** Parallel Slave Port Mode Select bit  
1 = Parallel slave port mode  
0 = General purpose I/O mode  

bit 3: **Unimplemented:** Read as ‘0’  

bit 2: **Bit2:** Direction control bit for pin RE2/CS/AN7  
1 = Input  
0 = Output  

bit 1: **Bit1:** Direction control bit for pin RE1/WR/AN6  
1 = Input  
0 = Output  

bit 0: **Bit0:** Direction control bit for pin RE0/RD/AN5  
1 = Input  
0 = Output
FIGURE 2: PIE1 REGISTER

<table>
<thead>
<tr>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSPIE</td>
<td>ADIE</td>
<td>RCIE</td>
<td>TXIE</td>
<td>SSPIE</td>
<td>CCP1IE</td>
<td>TMR2IE</td>
<td>TMR1IE</td>
</tr>
<tr>
<td>bit7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>bit0</td>
</tr>
</tbody>
</table>

- **PSPIE**: Parallel Slave Port Read/Write Interrupt Enable bit
  - 1 = Enables the PSP read/write interrupt
  - 0 = Disables the PSP read/write interrupt

- **ADIE**: A/D Converter Interrupt Enable bit
  - 1 = Enables the A/D interrupt
  - 0 = Disables the A/D interrupt

- **RCIE**: USART Receive Interrupt Enable bit
  - 1 = Enables the USART receive interrupt
  - 0 = Disables the USART receive interrupt

- **TXIE**: USART Transmit Interrupt Enable bit
  - 1 = Enables the USART transmit interrupt
  - 0 = Disables the USART transmit interrupt

- **SSPIE**: Synchronous Serial Port Interrupt Enable bit
  - 1 = Enables the SSP interrupt
  - 0 = Disables the SSP interrupt

- **CCP1IE**: CCP1 Interrupt Enable bit
  - 1 = Enables the CCP1 interrupt
  - 0 = Disables the CCP1 interrupt

- **TMR2IE**: TMR2 to PR2 Match Interrupt Enable bit
  - 1 = Enables the TMR2 to PR2 match interrupt
  - 0 = Disables the TMR2 to PR2 match interrupt

- **TMR1IE**: TMR1 Overflow Interrupt Enable bit
  - 1 = Enables the TMR1 overflow interrupt
  - 0 = Disables the TMR1 overflow interrupt

---

R = Readable bit
W = Writable bit
U = Unimplemented bit, read as '0'
- n = Value at POR reset
FIGURE 3: PIR1 REGISTER

<table>
<thead>
<tr>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R-0</th>
<th>R-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSPIF(1)</td>
<td>ADIF</td>
<td>RCIF</td>
<td>TXIF</td>
<td>SSPIF</td>
<td>CCP1IF</td>
<td>TMR2IF</td>
<td>TMR1IF</td>
</tr>
</tbody>
</table>

bit7

bit 7: **PSPIF(1)**: Parallel Slave Port Read/Write Interrupt Flag bit
- 1 = A read or a write operation has taken place (must be cleared in software)
- 0 = No read or write has occurred

bit 6: **ADIF**: A/D Converter Interrupt Flag bit
- 1 = An A/D conversion completed (must be cleared in software)
- 0 = The A/D conversion is not complete

bit 5: **RCIF**: USART Receive Interrupt Flag bit
- 1 = The USART receive buffer is full (cleared by reading RCREG)
- 0 = The USART receive buffer is empty

bit 4: **TXIF**: USART Transmit Interrupt Flag bit
- 1 = The USART transmit buffer is empty (cleared by writing to TXREG)
- 0 = The USART transmit buffer is full

bit 3: **SSPIF**: Synchronous Serial Port Interrupt Flag bit
- 1 = The transmission/reception is complete (must be cleared in software)
- 0 = Waiting to transmit/receive

bit 2: **CCP1IF**: CCP1 Interrupt Flag bit
  - Capture Mode
    - 1 = A TMR1 register capture occurred (must be cleared in software)
    - 0 = No TMR1 register capture occurred
  - Compare Mode
    - 1 = A TMR1 register compare match occurred (must be cleared in software)
    - 0 = No TMR1 register compare match occurred
  - PWM Mode
    - Unused in this mode

bit 1: **TMR2IF**: TMR2 to PR2 Match Interrupt Flag bit
- 1 = TMR2 to PR2 match occurred (must be cleared in software)
- 0 = No TMR2 to PR2 match occurred

bit 0: **TMR1IF**: TMR1 Overflow Interrupt Flag bit
- 1 = TMR1 register overflowed (must be cleared in software)
- 0 = TMR1 register did not overflow

Note 1: PIC16C73/73A/76 devices do not have a Parallel Slave Port implemented, this bit location is reserved on these devices, always maintain this bit clear.

Interrupt flag bits get set when an interrupt condition occurs regardless of the state of its corresponding enable bit or the global enable bit, GIE (INTCON<7>). User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.
## TABLE 3: INTCON REGISTER

<table>
<thead>
<tr>
<th>bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td><strong>GIE</strong>: Global Interrupt Enable bit</td>
</tr>
<tr>
<td></td>
<td>1 = Enables all un-masked interrupts</td>
</tr>
<tr>
<td></td>
<td>0 = Disables all interrupts</td>
</tr>
<tr>
<td>6</td>
<td><strong>PEIE</strong>: Peripheral Interrupt Enable bit</td>
</tr>
<tr>
<td></td>
<td>1 = Enables all un-masked peripheral interrupts</td>
</tr>
<tr>
<td></td>
<td>0 = Disables all peripheral interrupts</td>
</tr>
<tr>
<td>5</td>
<td><strong>T0IE</strong>: TMR0 Overflow Interrupt Enable bit</td>
</tr>
<tr>
<td></td>
<td>1 = Enables the TMR0 interrupt</td>
</tr>
<tr>
<td></td>
<td>0 = Disables the TMR0 interrupt</td>
</tr>
<tr>
<td>4</td>
<td><strong>INTE</strong>: RB0/INT External Interrupt Enable bit</td>
</tr>
<tr>
<td></td>
<td>1 = Enables the RB0/INT external interrupt</td>
</tr>
<tr>
<td></td>
<td>0 = Disables the RB0/INT external interrupt</td>
</tr>
<tr>
<td>3</td>
<td><strong>RBIE</strong>: RB Port Change Interrupt Enable bit</td>
</tr>
<tr>
<td></td>
<td>1 = Enables the RB port change interrupt</td>
</tr>
<tr>
<td></td>
<td>0 = Disables the RB port change interrupt</td>
</tr>
<tr>
<td>2</td>
<td><strong>T0IF</strong>: TMR0 Overflow Interrupt Flag bit</td>
</tr>
<tr>
<td></td>
<td>1 = TMR0 register has overflowed (must be cleared in software)</td>
</tr>
<tr>
<td></td>
<td>0 = TMR0 register did not overflow</td>
</tr>
<tr>
<td>1</td>
<td><strong>INTF</strong>: RB0/INT External Interrupt Flag bit</td>
</tr>
<tr>
<td></td>
<td>1 = The RB0/INT external interrupt occurred (must be cleared in software)</td>
</tr>
<tr>
<td></td>
<td>0 = The RB0/INT external interrupt did not occur</td>
</tr>
<tr>
<td>0</td>
<td><strong>RBIF</strong>: RB Port Change Interrupt Flag bit</td>
</tr>
<tr>
<td></td>
<td>1 = At least one of the RB7:RB4 pins changed state (must be cleared in software)</td>
</tr>
<tr>
<td></td>
<td>0 = None of the RB7:RB4 pins have changed state</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-0</th>
<th>R/W-x</th>
</tr>
</thead>
<tbody>
<tr>
<td>GIE</td>
<td>PEIE</td>
<td>T0IE</td>
<td>INTE</td>
<td>RBIE</td>
<td>T0IF</td>
<td>INTF</td>
<td>RBIF</td>
</tr>
</tbody>
</table>

- **R** = Readable bit
- **W** = Writable bit
- **U** = Unimplemented bit, read as '0'
- **n** = Value at POR reset
APPENDIX A: PIC16C64/74 PARALLEL SLAVE PORT

MPASM 01.40 Released PSP64.ASM 1-16-1997 17:03:44 PAGE 1

LOC OBJECT CODE LINE SOURCE TEXT

00001 ;*********************************************************************
00002 ;* 16C64/74 Parallel Slave port
00003 ;* This program demonstrates the Parallel Slave Port function of
00004 ;* the PIC16C64/74. The program is interrupt driven, when the PIC
00005 ;* is either read from or written to, an interrupt is generated. If
00006 ;* the interrupt was caused by a read, a register is incremented, and
00007 ;* the new count is placed in an output queue. If the interrupt was
00008 ;* caused by a write, the data is put on the Port B pins
00009 ;
00010 ; Program: PSP64.ASM
00011 ; Revision Date: 1-15-97 Compatibility with MPASMWIN 1.40
00012 ;
00013 ;********************************************************************
00014 ;;********************************************************************
00015         list p=16c64
00016         ERRORLEVEL -302
00017 ;
00018         include "p16c64.inc"
00019
00020 ; Register definitions
00021     FLAGREG equ     20h             ; Flag bit register
00022     OUTDATA equ     21h             ; Output data
00023     INDATA  equ     22h             ; Input data
00024     COUNT   equ     23h             ; Count of times output register read
00025
00026 ; Bit definitions for flag register
00027     err     equ     00h             ; Error flag bit
00028     OUTRDY  equ     01h             ; Output data ready flag
00029     INFULL  equ     02h             ; Input data received flag
00030
00031     org 0000h ; Reset Vector
00032     goto Start
00033
00034     org 0005h ; Interrupt Vector
00035     goto Service_Int
00036
00037 Start
00038     clrf OUTDATA          ; Clear data registers
00039     clrf INDATA
00040     bsf STATUS,RP0       ; Select register Bank1
00041     movlw b'00010111'     ; Set RD, WR, and CS as
00042     movwf TRISE           ; inputs, Enable Parallel Slave port
00043     movlw 0FFh
00044     movwf TRISB           ; Set Port_B to all outputs
00045     movlw b'10000000'     ;
00046     movwf PIE1            ; Enable Parallel Slave Port interrupt
00047     bcf STATUS,RP0       ; Select register Bank0
00048
00049     movf OUTDATA,W      ; Set output Data in PORTD
00050     movwf PORTD
00051     movlw b'11000000'     ; Set GIE, PEIE (enable interrupts)

Please check the Microchip BBS for the latest version of the source code. Microchip's Worldwide Web Address: www.microchip.com; Bulletin Board Support: MCHIPBBS using CompuServe® (CompuServe membership not required).
0013 008B 00052 movwf INTCON
00053
0014 00054 Loop
0014 1920 00055 btfsc FLAGREG,INFULL ;Check if input data received
0015 2819 00056 goto Checkout ;No data ready, check output
0016 1120 00057 bcf FLAGREG,INFULL ;Clear input data ready flag
0017 0822 00058 movf INDATA,W ;Get Input data
0018 0086 00059 movwf PORTB ;Output input data to Port_B
0019 00060 Checkout
0019 18A0 00061 btfsc FLAGREG,OUTRDY ;Check if data output already
001A 0062 goto Loop ;Not output yet, loop
001B 0063 incf COUNT,F ;Increment output data
001C 0064 movf COUNT,W ;Get data in output queue
001D 0065 movwf OUTDATA ;Put data in output queue
001E 0066 bcf FLAGREG,OUTRDY ;Set flag for interrupt routine
001F 0067 goto Loop
0068
0069 ;*********************************************************************

0070 ;*Interrupt Service Routine
0071 ;* Inputs: FLAGREG - Flag register to/from the main routine:
0072 ;*          Bit 1: OUTRDY - To Service_Int, indicates
0073 ;*              data ready in output queue
0074 ;*          Bit 2: INFULL - From Service_Int, indicates
0075 ;*              data received and in INDATA
0076 ;*          OUTDATA - Output data queue
0077 ;*          PIR1 - Interrupt flag register
0078 ;*          TRISE - Parallel slave port flag register
0079 ;*          PORTD - Input data from slave port
0080 ;* Outputs:
0081 ;*          PORTD - Output data to slave port
0082 ;*          INDATA - Input data queue
0083 ;*          FLAGREG - Flag register to/from the main routine:
0084 ;*          Bit 0: ERROR - From Service_Int, indicates
0085 ;*              input buffer overflow
0086 ;*          Bit 2: INFULL - From Service_Int, indicates
0087 ;*              data received and in INDATA
0088
0089 Service_Int
0090 1F8C 0090 btfs PIR1,PSPIF ;Test for Peripheral interrupt
0091 2832 0091 goto Intout ;Not a Peripheral interrupt, exit
0092 138C 0092 bcf PIR1,PSPIF ;Clear Peripheral interrupt
0093 1683 0093 bcf STATUS,RP0 ;Select Bank1
0094 1F89 0094 btfs TRISE,IBF ;Check if input data ready
0095 282A 0095 goto Notinput ;No input, check output
0096 1283 0096 bcf STATUS,RP0 ;Input ready, select Bank0
0097 1520 0097 bcf FLAGREG,INFULL ;Set flag for main routine
0098 0808 0098 movf PORTD,W ;Get input data
0099 00A2 0099 movwf INDATA ;Put byte in input queue
00A0 0100 009A Notinput
00A2 1B09 0101 btfs TRISE,OBF ;Check if output data read
00B2 2832 0102 goto Intout ;Not read, exit
00C2 1283 0103 bcf STATUS,RP0 ;Select Bank0
00D2 1CA0 0104 btfs FLAGREG,OUTRDY ;Check if data in output queue
00E2 2832 0105 goto Intout ;Output not read, exit
00F2 0821 0106 movf OUTDATA,W ;Get data from queue
0030 0088 0107 movf PORTD,F ;Put data in output buffer
0031 10A0 0108 bcf FLAGREG,OUTRDY ;Clear flag for main routine
0032 0109 Intout
0033 1683 0110 bcf STATUS,RP0 ;Select Bank1
0033 1A89 0111 btfs TRISE,IBOV ;Check input buffer overflow flag
0034 2837 0112 goto Intererr ;If not clear, error
0035 1283 0113 bcf STATUS,RP0 ;Select Bank0
0036 0009 0114 retfie ;Re-enable GIE and return
0037 0115 Intererr
0037 1283 0116 bcf STATUS,RP0 ;Select Bank0
0038 1420 0117 bcf FLAGREG,err ;Set error flag for main routine

© 1997 Microchip Technology Inc.
0039 0009 0019 retfie
00118 ;Re-enable GIE and return
00119
00120 end

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

0000 : X----XXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXX------
All other memory blocks unused.

Program Memory Words Used: 54
Program Memory Words Free: 1994

Errors : 0
Warnings : 0 reported, 0 suppressed
Messages : 0 reported, 6 suppressed
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.

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.