AN244

Internal RC Oscillator Calibration

Author: *Chris Valenti*
Microchip Technology Inc.

INTRODUCTION

Recently, Microchip has introduced a series of new PIC16 and PIC18 PICmicro® microcontrollers that have an on-board Internal RC Oscillator, capable of eight frequencies from 31 kHz to 8 MHz. These microcontrollers include the following devices:

- PIC16F818/819
- PIC16F87/88
- PIC18F1X20
- PIC18F2220/2320/4220/4320

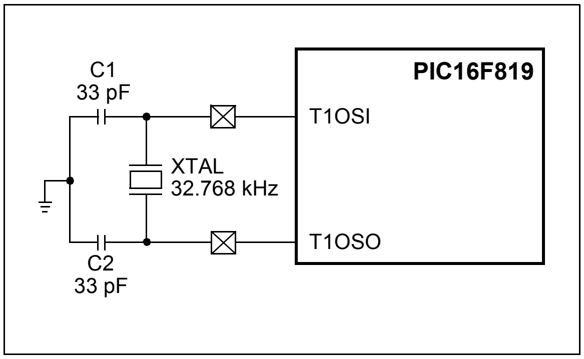
The internal RC oscillator is configured and tuned by the OSCCON and OSCTUNE registers, respectively. The OSCCON register sets the frequency and contains the IOFS bit, which indicates when the frequency has stabilized after a frequency change in the OSCCON register. OSCTUNE is responsible for tuning the frequency within a range of +/-12 percent on the selected frequency. The upper range of tuning is 01h - 1Fh and the lower range is 20h - 3Fh, with center frequency at 00h. It is recommended to consult the device data sheet for further details.

The assembly code included in this application note (see Appendix A) is only applicable for a PIC16F819 microcontroller; however, with minor changes, the code can be ported over to the devices listed above. Having an internal RC oscillator eliminates the need for capacitors and an external crystal, resulting in production cost savings.

You may be wondering, “Why would it be necessary to calibrate the internal oscillator if it is supposed to come from the factory already calibrated?” Even though the internal oscillator is factory calibrated at 25°C, temperature variations in the operating environment must be considered. Since a change in frequency, as little as two to three percent, can corrupt serial communication, the information presented in this application note can help you avoid this issue by allowing frequency calibration over temperature.

The ability to calibrate the internal oscillator means that a known time-base must be established as a reference point. This time-base will be derived from a 32.768 kHz clock crystal connected to Timer1 (see Figure 1). Other fixed time-bases that can be used are AC line frequency (50 or 60 Hz).

FIGURE 1: TIMER1 OSCILLATOR WITH 32.768 KHZ XTAL



In this application note, we will be calibrating the 4 MHz tap of the internal oscillator. The calibration method will compare the time it takes for a specific number of instructions to occur over a defined period of Timer1. It is not mandatory to calibrate the internal oscillator at 4 MHz as Table 1 indicates; however, if you choose 4 MHz for calibration and your code requires a frequency change after calibration is complete, the other frequency chosen will be calibrated.

TECHNICAL DETAILS

Before going further, let us first establish several system numbers that will be important for understanding the techniques used in this application note. The values in Table 1 were derived from the base numbers shown below in Figure 1.

Knowing that 30.5 instructions can be executed within one Timer1 count, we can now determine if the oscillator requires calibration and whether it is too fast or too slow.

There are several variables that are required for this calibration approach. The first is the number of Timer1 counts. This number is subtracted from FFFFh to determine what value TMR1H:TMR1L will be loaded with. This is not a code-defined variable, but you may choose to define it as such. A defined constant, called `calc_instr`, is set by the user for the expected number of instructions that will be executed for a defined Timer1 interval established by the TMR1H:TMR1L value.

The second defined constant is `f_delta_calc`, which is set by the user according to their accuracy requirements. This number is the allowable difference between the calculated number of instructions expected (`calc_instr`) and the measured (`mesrd_instr`) number of instructions executed.

As previously mentioned, Table 1 shows several of the available internal RC oscillator frequencies. For each of these frequencies, there are instruction execution times with +/-1% accuracy, and the calculated number of instructions for 25 Timer1 counts. The `mesrd_instr` variable increments until the Timer1 overflow flag is set. The `f_delta_count`, as mentioned previously, is the acceptable difference by `mesrd_inst` and `calc_instr`. By changing the value shown in the T1 CNTS column of Table 1, the following columns will yield different results. If the `f_delta_count` is increased, the calibration time is decreased, but accuracy will fall off.

TABLE 1: INTRC OSCILLATOR FACTORY CALIBRATION VALUES

INTRC Freq	FOSC/4	TCY (μs)	TCY/T1 Count	Drift (%)	Freq (-1%)	TCY (μs) (Slow)	TCY/T1 Count	Freq (+1%)	TCY (μs) (Fast)	TCY/T1 Count
8000000	2000000	0.500	61.036	1	1980000	0.505	60.426	2020000	0.495	61.646
4000000	1000000	1.000	30.518	1	990000	1.010	30.213	1010000	0.990	30.823
2000000	500000	2.000	15.259	1	495000	2.020	15.106	505000	1.980	15.412
1000000	250000	4.000	7.630	1	247500	4.040	7.553	252500	3.960	7.706

T1 CNTS	TCY Total (Slow)	TCY Total	TCY Total (Fast)	calc _instr	f_delta_count
25	1510.6	1525.90	1541.2	381.5	3.8
25	755.3	762.95	770.6	190.7	1.9
25	377.7	381.48	385.3	95.4	1.0
25	188.8	190.74	192.6	47.7	0.5

EXAMPLE 1: CALIBRATION BASE NUMBERS

- Timer1 Clock Crystal: 32.768 kHz → 1/32,768 = 30.5 μs period
- System Clock: 4 MHz → FOSC/4 = 1 MHz core clock = 1 μs Instruction time (TCY)
- 4 MHz @ +1% = 1.01 MHz core clock = .990 μs TCY
- 4 MHz @ -1% = .990 MHz core clock = 1.01 μs TCY
- @ 4 MHz (1 μs TCY), 30.5 core instructions can be executed for one count of Timer1.

CODE DESCRIPTION

Before the calibration code is executed as shown in the flow diagram in Figure 2, the user must ensure that the 32.768 kHz clock crystal connected to Timer1 has adequate time to become stable. If this stability is not ensured, then the calibration routine will be driven from an inaccurate time-base. The required initialization is minimal; the internal oscillator is configured for 4 MHz via the OSCCON register. Timer1 is configured via T1CON for:

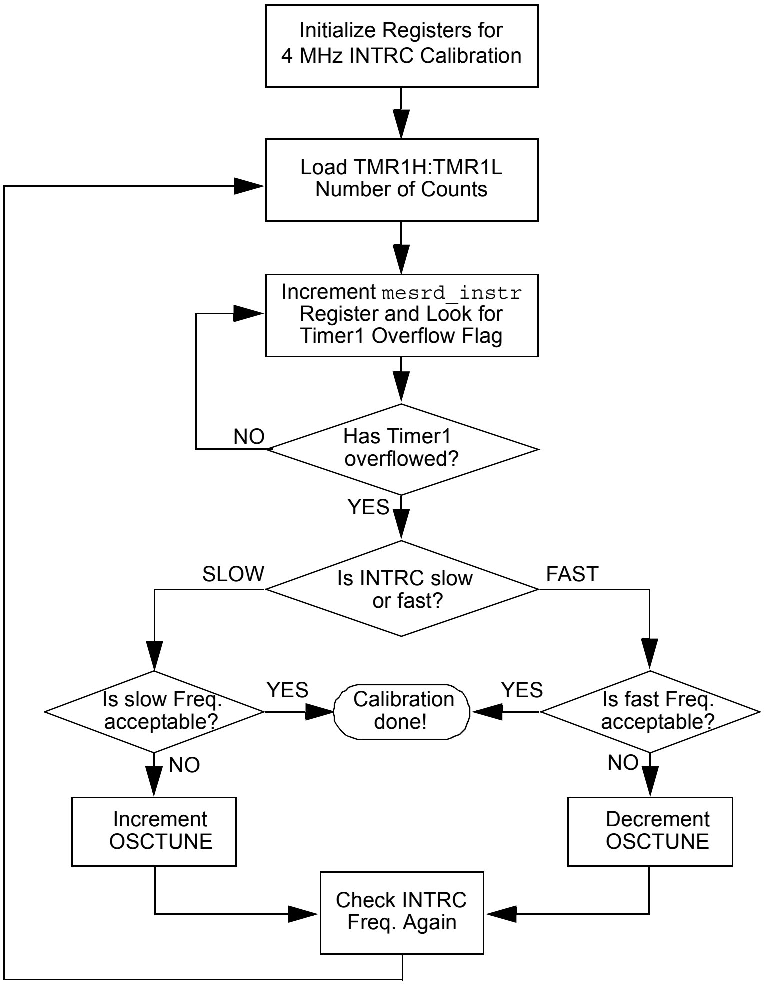
- 1:1 prescale
- Osc enabled
- Do not synchronize external clock input
- External clock from T1OSO
- Timer1 enabled

Upon entry into the calibration routine, the TMR1H:TMR1L registers are loaded with 0xFFE7h, which establishes 25 Timer1 counts until overflow. An incrementing loop is entered and exit only takes place when a Timer1 overflow occurs (TMR1IF = 1).

Next, the mesrd_instr variable is subtracted from the calc_instr constant to determine if the frequency is slower or faster than 4 MHz. Once this decision is made by a bit test on the carry flag, either the slower_freq or faster_freq subroutine will be entered. In both of these instruction sets, the current frequency is checked if it is acceptable for operation. This is determined by comparing the difference between mesrd_instr and calc_instr with the value of f_delta_count. If the result is acceptable, then the calibration is complete. If the result is unacceptable, the OSCTUNE register decrements or increments, depending on the speed of the measured frequency.

When the OSCTUNE register is modified, the new “tuned” values take approximately 256 μs to become stable. Therefore, a delay (tune_delay) must be called before the frequency can be checked again. When the delay_temp variable is loaded with 0x57, the tune_delay loop will take up to 266 μs @ 4 MHz.

FIGURE 2: CALIBRATION PROCESS FLOW DIAGRAM


Initialize Registers for 4 MHz INTRC Calibration
Load TMR1H:TMR1L Number of Counts
Increment mesrd_instr Register and Look for Timer1 Overflow Flag
NO
Has Timer1 overflowed?
YES
SLOW
Is INTRC slow or fast?
FAST
Is slow Freq. acceptable?
YES
Calibration done!
YES
Is fast Freq. acceptable?
NO
NO
Increment OSCTUNE
Decrement OSCTUNE
Check INTRC Freq. Again

CONCLUSION

The assembly code has been written as relocatable code so that it can easily be added to an existing application. The code usage is 56 words of program memory, 2 bytes of RAM and takes approximately 31 ms from start to finish. Although specific numbers have been used in this application, they may be different for other application requirements. TMR1H:TMR1L can be loaded with different values, which can speed up the calibration routine with less frequency sampling, or slow the routine with more sampling.

The `f_delta_count` value in this application is defined for +/-1 percent, but can be changed for 2 or 3 percent accuracy. These variables can be used in applications at the discretion of the user. Some applications may require that the OSCTUNE value after calibration be retained for RESET or POR conditions. This can be accomplished by storing the value in the PICmicro microcontroller’s on-board EEPROM.

So that an organized starting point in your code development can be established, it is recommended that you create a spreadsheet file similar to that of Table 1. Doing this will also help in fine tuning the calibration boundaries to the application.

APPENDIX A: SOURCE CODE

```
;*********************************************************************
;*     This code assumes the following conditions:
;*            1. Config bits FOSC2:FOSC0 set to INTRC.
;*
;*            2. INTOSC is configured for 4MHz system clock via OSCCON reg.
;*
;*            3. Timer1 has been configured for 32.768kHz external crystal
;*               and stable operation has been established.
;*
;*     Call from main: call    int_rc_cal
;*********************************************************************
;* FileName:          rc_cal.asm
;* Dependencies:      P16819.inc, 16f819.lkr
;* Processor:         16F819
;* MPLAB IDE          6.10.0.0
;* MPASM Tool Suite   1.10.0.0
;* Company:           Microchip Technology, Inc.
;*
;* Software License Agreement
;*
;* The software supplied herewith by Microchip Technology Incorporated
;* (the "Company") for its PICmicro® Microcontroller is intended and
;* supplied to you, the Company's customer, for use solely and
;* exclusively on Microchip PICmicro Microcontroller products. The
;* software is owned by the Company and/or its supplier, and is
;* protected under applicable copyright laws. All rights are reserved.
;* Any use in violation of the foregoing restrictions may subject the
;* user to criminal sanctions under applicable laws, as well as to
;* civil liability for the breach of the terms and conditions of this
;* license.
;*
;* THIS SOFTWARE IS PROVIDED IN AN "AS IS" CONDITION. NO WARRANTIES,
;* WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING, BUT NOT LIMITED
;* TO, IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
;* PARTICULAR PURPOSE APPLY TO THIS SOFTWARE. THE COMPANY SHALL NOT,
;* IN ANY CIRCUMSTANCES, BE LIABLE FOR SPECIAL, INCIDENTAL OR
;* CONSEQUENTIAL DAMAGES, FOR ANY REASON WHATSOEVER.
;*
;* Author             Date           Comment
;*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
;* Chris Valenti      Nov 19, 2002   Initial Release (V1.0)
;*
;********************************************************************/

       list           P=16F819
       #include       <P16F819.INC>

       errorlevel     -302

;********************** Defined constants ***************************
#define       calc_instr     .190    ;calculated # of instructions
```

```
#define        f_delta_calc   .1     ;acceptable difference
;*********************************************************************


;********************** Variables ***********************************
cal_var UDATA_SHR       0x20
mesrd_instr             RES       1   ;measured # of instructions executed
delay_temp              RES       1   ;delay variable
;*********************************************************************

cal     CODE       0x0005
;******** Internal RC Calibration Routine ***************************
int_rc_cal
        banksel TMR1H
        bcf     PIR1,TMR1IF           ;clear TIMER 1 interrupt flag
        clrf    mesrd_instr           ;clear loop count register
        movlw   0xFF                  ;use 25 counts of Timer 1
        movwf   TMR1H
        movlw   0xE7
        movwf   TMR1L                 ;T1 starts count here

        incf    mesrd_instr,f         ;increment
        btfss   PIR1,TMR1IF           ;has TMR1 overflowed ??
        goto    $-2                   ;NO, continue incrementing

count_done                            ;YES, start calibration
        banksel mesrd_instr
        movf    mesrd_instr,w
        sublw   calc_instr            ;calc_instr - mesrd_instr, find f_delta
        btfss   STATUS,C              ;Is the Frequency slow or fast ?
        goto    slower_freq           ;mesrd_count > calc_count, freq is fast
        goto    faster_freq           ;mesrd_count < calc_count, freq is slow

slower_freq                           ;DECREASE FREQUENCY
        sublw   0xFF                  ;FF - w_reg
        sublw   f_delta_calc          ;Check if freq is acceptable
        btfsc   STATUS,C              ;Is w_reg > f_delta_calc?
        goto    done                  ;NO, calibration done!
        banksel OSCTUNE
        decf    OSCTUNE,F             ;YES, decrease the frequency
        call    tune_delay            ;allow OSCTUNE change to take effect
        goto    int_rc_cal            ;calibrate again

faster_freq                           ;INCREASE FREQUENCY
        sublw   f_delta_calc          ;Check if freq is acceptable
        btfsc   STATUS,C              ;Is w_reg > f_delta_calc?
        goto    done                  ;NO, calibration done!
        banksel OSCTUNE
        incf    OSCTUNE,f             ;YES, increase the frequency
        call    tune_delay            ;allow OSCTUNE to take effect
        goto    int_rc_cal            ;calibrate again
done
        return                        ;continue with main code
        GLOBAL  int_rc_cal

;*************************************************************************
; Function: tune_delay
;
; Overview:
;              This routine is used as a delay to allow the tuned value of
;              OSCTUNE to take effect. >87 loaded into delay_temp will yield
;              a 266us delay @ 4MHz
;
; Input:       None
; Output:      None
```

```
;
; Stack Requirements: 1 level deep
;
; Side Effects: Databank changed
;***************************************************************************

tune_delay                    ;clock stabilization delay (8 clocks)
        banksel delay_temp
        movlw   .87
        movwf   delay_temp
        decfsz  delay_temp,f
        goto    $-1
        return

        end



                                  Section Info
                   Section    Type       Address    Location   Size(Bytes)
                   ---------  ---------  ---------  ---------  ---------
                   STARTUP    code       0x000000   program    0x00000a
                       cal    code       0x000005   program    0x000056
                    .cinit    romdata    0x000030   program    0x000004
                    main_c    code       0x000040   program    0x000012
                   .config    code       0x002007   program    0x000002
                   cal_var    udata      0x000020   data       0x000002



                          Program Memory Usage
                          Start      End
                          --------   --------
                          0x000000   0x000031
                          0x000040   0x000048
                          0x002007   0x002007
                   60 out of 2312 program addresses used, program memory utilization is 2%


              Symbols - Sorted by Address
Name           Address    Location   Storage    File
-----------    --------   --------   --------   --------
 int_rc_cal    0x000005   program    extern     C:\Program Files\MPLAB IDE\rc_cal.asm
  _cal_000F    0x00000f   program    static     C:\Program Files\MPLAB IDE\rc_cal.asm
 count_done    0x000010   program    static     C:\Program Files\MPLAB IDE\rc_cal.asm
slower_freq    0x000017   program    static     C:\Program Files\MPLAB IDE\rc_cal.asm
faster_freq    0x000020   program    static     C:\Program Files\MPLAB IDE\rc_cal.asm
       done    0x000028   program    static     C:\Program Files\MPLAB IDE\rc_cal.asm
 tune_delay    0x000029   program    static     C:\Program Files\MPLAB IDE\rc_cal.asm
  _cal_002E    0x00002e   program    static     C:\Program Files\MPLAB IDE\rc_cal.asm
      Start    0x000040   program    static     C:\Program Files\MPLAB IDE\main.asm
mesrd_instr    0x000020   data       static     C:\Program Files\MPLAB IDE\rc_cal.asm
 delay_temp    0x000021   data       static     C:\Program Files\MPLAB IDE\rc_cal.asm
```

NOTES:

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families 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 Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is 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 products.

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, KEELOQ, MPLAB, PIC, PICmicro, PICSTART, PRO MATE and PowerSmart are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

FilterLab, microID, MXDEV, MXLAB, PICMASTER, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Accuron, dsPIC, dsPICDEM.net, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, microPort, Migratable Memory, MPASM, MPLIB, MPLINK, MPSIM, PICC, PICkit, PICDEM, PICDEM.net, PowerCal, PowerInfo, PowerTool, rfPIC, Select Mode, SmartSensor, SmartShunt, SmartTel and Total Endurance are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

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.

Printed on recycled paper.

DNV Certification, Inc.
USA

ANSI ★ RAB
QMS
ACCREDITED

DNV

DNV MSC
The Netherlands
Accredited by the RvA

ISO 9001 / QS-9000
REGISTERED FIRM

Microchip received QS-9000 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona in July 1999 and Mountain View, California in March 2002. The Company’s quality system processes and procedures are QS-9000 compliant for its PICmicro® 8-bit MCUs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, non-volatile memory and analog 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: 480-792-4338

Atlanta
3780 Mansell Road, Suite 130
Alpharetta, GA 30022
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

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. 2401-2402, 24th Floor,
Ming Xing Financial Tower
No. 88 TIDU Street
Chengdu 610016, China
Tel: 86-28-86766200 Fax: 86-28-86766599

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 - Hong Kong SAR
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

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. 1812, 18/F, Building A, United Plaza
No. 5022 Binhe Road, Futian District
Shenzhen 518033, China
Tel: 86-755-82901380 Fax: 86-755-82966626

China - Qingdao
Rm. B503, Fullhope Plaza,
No. 12 Hong Kong Central Rd.
Qingdao 266071, China
Tel: 86-532-5027355 Fax: 86-532-5027205

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

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

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

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

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

EUROPE

Austria
Microchip Technology Austria GmbH
Durisolstrasse 2
A-4600 Wels
Austria
Tel: 43-7242-2244-399
Fax: 43-7242-2244-393

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

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

Germany
Microchip Technology GmbH
Steinheilstrasse 10
D-85737 Ismaning, 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
Microchip Ltd.
505 Eskdale Road
Winnersh Triangle
Wokingham
Berkshire, England RG41 5TU
Tel: 44 118 921 5869 Fax: 44-118 921-5820

12/05/02