MICROCHIP

AN248

Interfacing MCP6S2X PGAs to PICmicro® Microcontroller

Author: *Ezana Haile*
Microchip Technology Inc.

INTRODUCTION

The MCP6S21/2/6/8 family of one, two, six or eight channel Programmable Gain Amplifiers (PGA) communicate using a standard 3-wire Serial Peripheral Interface (SPI™) protocol. This application note shows how to program the six channel MCP6S26 PGA gains, channels and shutdown registers using the PIC16C505 microcontroller.

The PIC16C505 microcontroller does not have a hardware SPI module, therefore, a firmware SPI (Bit Bang) method is used to program the PGA. The MCP6S2X Evaluation Board was used to develop this application note firmware. For additional information, refer to the MCP6S2X Evaluation Board User's Guide (DS51327), MCP6S21/2/6/8 PGA data sheet (DS21117) and PIC16C505 data sheet (DS40192).

COMMUNICATION

Figure 1 shows the SPI communication format with the clock idle low and the serial data latched at the rising edge of the clock. This format is sometimes referred to as "0,0" mode. Instructions for the MCP6S21/2/6/8 family of PGAs consists of sixteen serial clock cycles and two serial data bytes. The first byte is the instruction byte, which consists of register options, such as Write, Shutdown, Gain and Channel. The second byte is the data byte, whose three Least Significant Bits (LSB) are used to program gain or channel. There are eight gain settings and up to eight channel settings, depending on the specific device.

Command bits 7, 6 and 5 of the instruction byte determines whether to write to the registers or shutdown the device. A bit configuration of `010` directs the data byte to be written to the Gain or Channel registers. When writing to the registers, if command bit `0` is cleared, the data byte is shifted into the Gain register. If the command bit `0` is set, the data byte is shifted into the Channel register.

A `001` command bit configuration shuts down the device. If shutdown is selected, command bit `0` becomes a "don't care" bit.

Command bits 4, 3, 2 and 1, and data bits 7 through 3, are not used and are "don't cares". Refer to Register 1, Register 2 and Register 3 tables on pages 2 and 3.

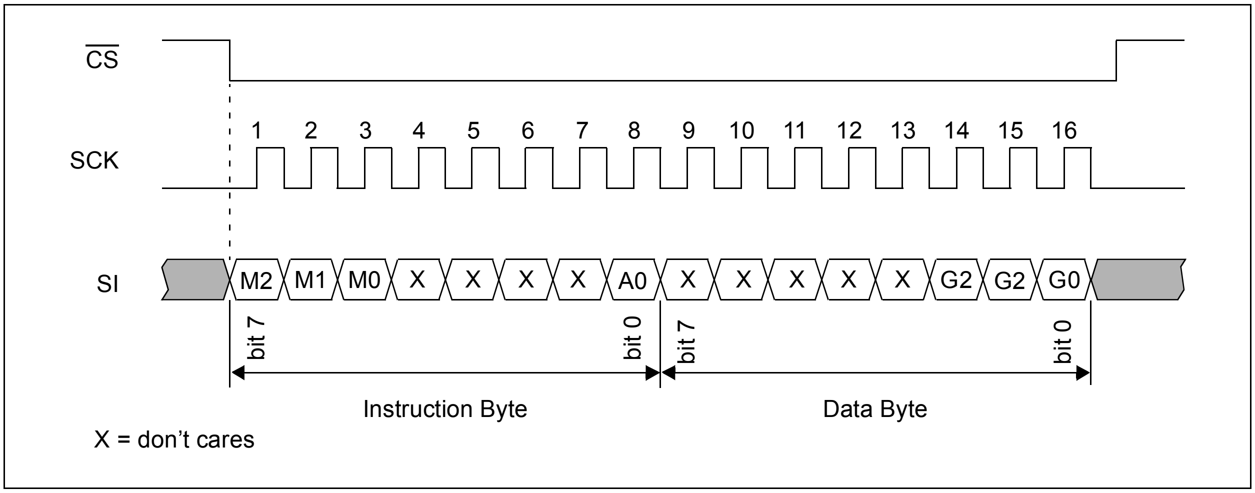

FIGURE 1: *PGA Instruction byte and data byte programming sequence.*

REGISTER 1: INSTRUCTION REGISTER

W-0	W-0	W-0	U-x	U-x	U-x	U-x	W-0
M2	M1	M0	—	—	—	—	A0
bit 7							bit 0

bit 7-5 **M2-M0: Command Bits**

`000` = `NOP` (Default) **(Note 1)**
`001` = PGA enters Shutdown Mode as soon as a full 16-bit word is sent and $\overline{CS}$ is raised. **(Notes 1 and 2)**
`010` = Write to register.
`011` = `NOP` (reserved for future use) **(Note 1)**
`1XX` = `NOP` (reserved for future use) **(Note 1)**

bit 4-1 **Unimplemented:** Read as ‘0’ (reserved for future use)

bit 0 **A0: Indirect Address Bit**

`1` = Addresses the Channel Register
`0` = Addresses the Gain Register (Default)

Note 1: All other bits in the 16-bit word (including A0) are don’t cares.

2: The device exits Shutdown mode when a valid command (other than `NOP` or Shutdown) is sent and $\overline{CS}$ is raised; that valid command will be executed. Shutdown does not toggle.

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	’1’ = Bit is set	’0’ = Bit is cleared	x = Bit is unknown

REGISTER 2: GAIN REGISTER

U-x	U-x	U-x	U-x	U-x	W-0	W-0	W-0
—	—	—	—	—	G2	G1	G0
bit 7							bit 0

bit 7-3 **Unimplemented:** Read as ‘0’ (reserved for future use)

bit 2-0 **G2-G0: Gain Select Bits**

`000` = Gain of +1 (Default)
`001` = Gain of +2
`010` = Gain of +4
`011` = Gain of +5
`100` = Gain of +8
`101` = Gain of +10
`110` = Gain of +16
`111` = Gain of +32

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	’1’ = Bit is set	’0’ = Bit is cleared	x = Bit is unknown

REGISTER 3: CHANNEL REGISTER

U-x	U-x	U-x	U-x	U-x	W-0	W-0	W-0
—	—	—	—	—	C2	C1	C0
bit 7							bit 0

bit 7-3 **Unimplemented:** Read as ‘0’ (reserved for future use)

bit 2-0 **C2-C0: Channel Select Bits**

	MCP6S21	**MCP6S22**	**MCP6S26**	**MCP6S28**
`000` =	CH0 (Default)	CH0 (Default)	CH0 (Default)	CH0 (Default)
`001` =	CH0	CH1	CH1	CH1
`001` =	CH0	CH0	CH2	CH2
`011` =	CH0	CH1	CH3	CH3
`100` =	CH0	CH0	CH4	CH4
`101` =	CH0	CH1	CH5	CH5
`110` =	CH0	CH0	CH0	CH6
`111` =	CH0	CH1	CH0	CH7

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	’1’ = Bit is set	’0’ = Bit is cleared	x = Bit is unknown

IMPLEMENTATION

Microchip Technology Inc.'s PIC16C505 microcontroller is used to program the six channel MCP6S26 PGA registers. Appendix A shows the assembly source code used for this application note. This source code is also availabe on the Microchip web site at www.microchip.com. This code shows the communication sequence necessary to program the PGA.

The code structure is as follows:

Initially, all the necessary constants, such as input/output lines, instruction bytes, gains, channels and shutdown are defined. Three Random Access Memory (RAM) bytes are then reserved. These bytes are used to temporarily store PGA register bits and SPI protocol counter bits. Once the RAM location is reserved, the code is originated at a program memory location and the input/output ports are defined.

This code has four short subroutines, labeled 'gain', 'channel', 'shutdown' and 'bitbang'. These subroutines are called using the 'call' instruction.

The 'gain' and 'channel' Subroutines

The 'gain' and 'channel' subroutines send the instruction and data bytes to program the PGA Gains and Channel registers. These bytes are defined by the user according to the constants defined within the source code. These constants are also listed in Table 1.

TABLE 1: SOURCE CODE CONSTANTS

Function	Constants	Bits
Gain 1	gain1	b'00000000'
Gain 2	gain2	b'00000001'
Gain 4	gain4	b'00000010'
Gain 5	gain5	b'00000011'
Gain 8	gain8	b'00000100'
Gain 10	gain10	b'00000101'
Gain 16	gain16	b'00000110'
Gain 32	gain32	b'00000111'
Channel 0	channel0	b'00000000'
Channel 1	channel1	b'00000001'
Channel 2	channel2	b'00000010'
Channel 3	channel3	b'00000011'
Channel 4	channel4	b'00000100'
Channel 5	channel5	b'00000101'
Gain register	PrgGain	b'01000000'
Channel register	PrgChannel	b'01000001'
Shutdown	PrgShdn	b'00100000'

The user can select the corresponding constant for the gain and channel before the 'call' instructions (as described within the source code) and compile the source code.

Before the 'gain' or the 'channel' subroutines are called using the 'call' instruction, the user-selected constant is transferred into the RAM location, labeled 'register'. Once these subroutines are called, chip-select 'cs' (defined as the 4th line of port C) is pulled low and the command byte ('PrgGain' for gain or 'PgrChannel' for channel) is loaded into the 'w' register. The 'call' instruction for the 'bitbang' subroutine is then executed to send the command bits to the PGA. Next, data stored in 'register' RAM location is transferred to the 'w' register. The 'bitbang' subroutine is called again to send the data to the PGA and, after this call, the chip-select line is pulled high.

The 'shutdown' Subroutine

The 'shutdown' subroutine sends the shutdown instruction to the PGA. The shutdown 'call' is commented ';', therefore, this call is not executed. In order to shutdown the PGA, the user must uncomment or delete the comment ';' and recompile the source code. This will execute the 'shutdown' subroutine.

When the 'shutdown' subroutine is called, 'cs' is pulled low and the command byte 'PrgShdn' is transferred into the 'w' register. The 'bitbang' subroutine is then called to send the data. This subroutine is called twice in order to complete the required 16 clock and data cycles, as shown in Figure 1. The second byte that is transferred to the PGA contains dummy bits.

The 'bitbang' Subroutine

The 'bitbang' subroutine transmits an eight bit serial word to the PGA using a firmware SPI protocol (or bit-bang). This subroutine sends out the data located in the 'w' register. This register is transferred into a RAM memory location, labeled 'buffer'. A counter byte is then loaded by hexadecimal '08' into RAM location (labeled 'SPIcounter') to count the transferring bits. The 7th bit of 'buffer' is transferred initially through the data-out 'do' (defined as the 3rd line of port C) line. Then the clock line 'sck' (defined as the 5th line of port C) is pulsed and the 'do' line is cleared. The 'buffer' is then shifted to the left using the 'rlf' instruction in order to send the next bit. This loop continues until the counter byte is decremented to hexadecimal '00', indicating that all bits are transferred.

CONCLUSION

This application note has illustrates a method of communication with the MCP6S21/2/6/8 family of PGAs using the PIC16C505 EPROM microcontroller. Because the firmware can be applied to other families of PICmicro microcontrollers, please refer to the PIC16C505 data sheet for source code compatibilities.

APPENDIX A: ASSEMBLY SOURCE CODE

```
;---------------------------------------------------------------------------
;
; This source code programs the PGA registers based on the gain and channel
; variabls that user selects. User must type the proper gain and channel
; according to the listed definitions and compile this source code.
;
;   File name:            pga2pic.asm
;   Date:                 02/19/03
;   File Version:         1.00
;
;   Programmer:           PRO MATE® II device programmer
;   File Required:        PIC16C505.inc
;
;   Author:               Ezana Haile
;   Company:              Microchip Technology Inc.
;
;---------------------------------------------------------------------------

    ERRORLEVEL -302
    ERRORLEVEL -305

#include <p16C505.inc>

    __CONFIG              _MCLRE_OFF & _WDT_OFF & _IntRC_OSC_RB4EN

; definitions for “data”
    #define gain1         b’00000000’    ; Gain of 1
    #define gain2         b’00000001’    ; Gain of 2
    #define gain4         b’00000010’    ; Gain of 4
    #define gain5         b’00000011’    ; Gain of 5
    #define gain8         b’00000100’    ; Gain of 8
    #define gain10        b’00000101’    ; Gain of 10
    #define gain16        b’00000110’    ; Gain of 16
    #define gain32        b’00000111’    ; Gain of 32
    #define channel0      b’00000000’    ; channel 0
    #define channel1      b’00000001’    ; channel 1
    #define channel2      b’00000010’    ; channel 2
    #define channel3      b’00000011’    ; channel 3
    #define channel4      b’00000100’    ; channel 4
    #define channel5      b’00000101’    ; channel 5

; definitions for "instructions"
    #define PrgGain       b’01000000’    ; instruction for gain register
    #define PrgChannel    b’01000001’    ; instruction for channel register
    #define PrgShdn       b’00100000’    ; instruction to shutdown PGA
```

```
; definitions for i/o ports for SPI communication
    #define cs              portc, rc4      ; chip select
    #define sck             portc, rc5      ; clock
    #define do              portc, rc3      ; buffer out

; reserve memory byte
      cblock     0X10
                 SPIcounter, buffer, register
      endc

;=============================================================
;==========    PROGRAM           ===========================
;=============================================================
PGA                                    ; code name
      org        0x00                  ; origination address
      movlw      b'00000111'           ; set the I/O for port c
      tris       portc
      movlw      b'00111111'           ; set the I/O for port b
      tris       portb
      bsf        cs                    ; unselect the device
      bcf        do                    ; keep the dataout (do) low
      bcf        sck                   ; clear clock
      clrf       buffer                ; clear buffer
      clrw                             ; clear w register

;*********************************************************************
; User can select the GAIN from the definition table provided above
; and type that as shown on the next line: "movlw gainx".
; where X is: 1, 2, 4, 5, 8, 10, 16, or 32
;*********************************************************************
      movlw      gain2                 ; <==== Change this gain constant
      movwf      register              ; move the data in RAM location
      call       Gain                  ; program gain

;*********************************************************************
; User can select the Channel from the definition table provided above
; and type that as shown on the next line: "movlw channelx".
; where X is: 0, 1, 2, 3, 4, or 5
;*********************************************************************
      movlw      channel0              ; <==== Change this channel constant
      movwf      register              ; move the data in RAM location
      call       Channel               ; program channel

;*********************************************************************
; If user needs to shutdown the PGA then the comment ";" needs
; to be removed from the following line: " ; call Shutdown "
;*********************************************************************
;     call       Shutdown              ; <==== uncomment this line for shutdown

      goto       finish                ; done

;-----------------------------------------------
;---- Gain programming subroutine
;-----------------------------------------------
Gain
      bcf        cs                    ; select PGA
      movlw      PrgGain               ; get the gain programing instruction
      call       bitbang               ; send the bits through SPI

      movf       register,w
      call       bitbang               ; send it through spi
      bsf        cs                    ; unselect the devices

      retlw      0                     ; return from call
```

```
;---------------------------------------------
;---- Channel programming subroutine
;---------------------------------------------
Channel
        bcf       cs                  ; select PGA
        movlw     PrgChannel          ; get the Channel programing instruction
        call      bitbang             ; send the bits through SPI

        movf      register,w
        call      bitbang             ; send it through spi
        bsf       cs                  ; unselect the devices

        retlw     0                   ; return from call

;---------------------------------------------
;---- Shutdown programming subroutine
;---------------------------------------------
Shutdown
        bcf       cs                  ; select PGA
        movlw     PrgShdn             ; get the Channel programing instruction
        call      bitbang             ; send the bits through SPI
        call      bitbang             ; send dummy bits
        bsf       cs                  ; unselect the devices

        retlw     0                   ; return from call

;---------------------------------------------
;---- Bit Bang SPI communication routine
;---------------------------------------------
bitbang
        movwf     buffer              ; move the data into RAM
        clrc
        movlw     h'08'               ; counter
        movwf     SPIcounter          ; set the bit bang counter
send    btfsc     buffer, 7           ; see the 7th bit of the buffer
        bsf       do                  ; if the 7th bit is set, then set the do high
        bsf       sck                 ; set clock (pulse clock)
        bcf       sck                 ; clear clock
        bcf       do                  ; clear the data
        rlf       buffer,f            ; roll the bits
        decfsz    SPIcounter, f       ; check end of counter
        goto      send                ; loop

        retlw     0                   ; return from call

;============================================================
;============================================================

finish  goto      finish
        end
```

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. Attempts to break microchip’s code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

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, Application Maestro, dsPIC, dsPICDEM, 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, PowerMate, PowerTool, rfLAB, 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
Marketing Support Division
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. B505A, 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
Marketing Support Division
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: 49-089-627-144-100
Fax: 49-089-627-144-44

Italy
Microchip Technology SRL
Via Quasimodo, 12
20025 Legnano (MI)
Milan, Italy
Tel: 39-0331-742611 Fax: 39-0331-466781

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

02/12/03