AN618

Modifying PIC16C54A Code for the PIC16C58A

Author: *Stan D'Souza*
Microchip Technology Inc.

INTRODUCTION

Code written for a PIC16C54A device can be easily modified and used in a PIC16C58A. Therefore, a PIC16C58A may be used in place of a PIC16C54A. The PIC16C58A and the PIC16C54A are pin-compatible devices. Also, the PIC16C58A has more than enough memory (program and data) to support PIC16C54A code.

Device	Pins	EPROM/ROM	RAM
PIC16C54A	18	512	25
PIC16C58A	18	2048	73

PROGRAM MEMORY

The PIC16C54A has 512 words of EPROM program memory, while the PIC16C58A has 2048 words. Figure 1 and Figure 2 depict the memory organization for each of these devices. For this application, Page 0 of the PIC16C58A On-chip Program Memory will be used in place of the PIC16C54A On-chip Program Memory. However, the Reset vector location will change from 1FFh for the PIC16C54A to 7FFh for the PIC16C58A. (See the discussion on Reset in the next section.)

The PC, or program counter, is used on both devices to access program memory locations. Eight of the nine bits required to access the 512 words of program memory in the PIC16C54A are provided by the PCL file register (Figure 3). The most significant 9th bit is provided from the instruction word during a `GOTO` instruction. A `CALL` instruction, on the other hand, forces a '0' into the most significant 9th bit of the PC. Hence, all subroutines must reside in the top half (00h to FFh) of the program memory.

On the PIC16C58A, bits 0 through 8 of the PC operate the same as bits 0 through 8 of the PIC16C54A PC for `GOTO` and `CALL` instructions (Figure 4). The PIC16C58A PC differs from the PIC16C54A PC in the addition of two most significant bits. These 10th and 11th bits are provided by the PA0 and PA1 values of the STATUS register. The bits signify the program memory page for the PIC16C58A (i.e., 00 = Page 0, 01 = Page 1, etc.).

After a power-up reset, the PA1:PA0 bits are reset to '0'. So, if these two bits are not modified by the PIC16C54A program, the same program will correctly access the Page 0 program memory space on a PIC16C58A.

FIGURE 1: PIC16C54A PROGRAM MEMORY MAP AND STACK

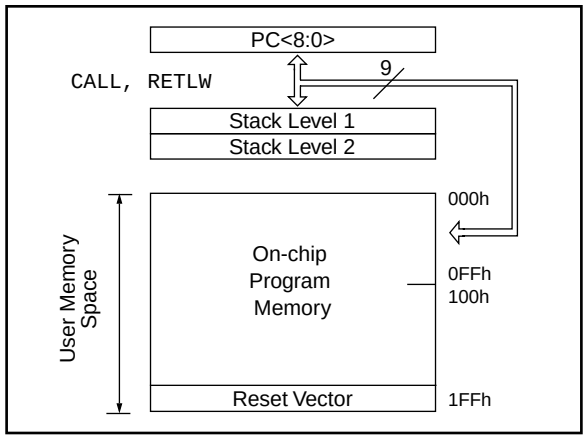
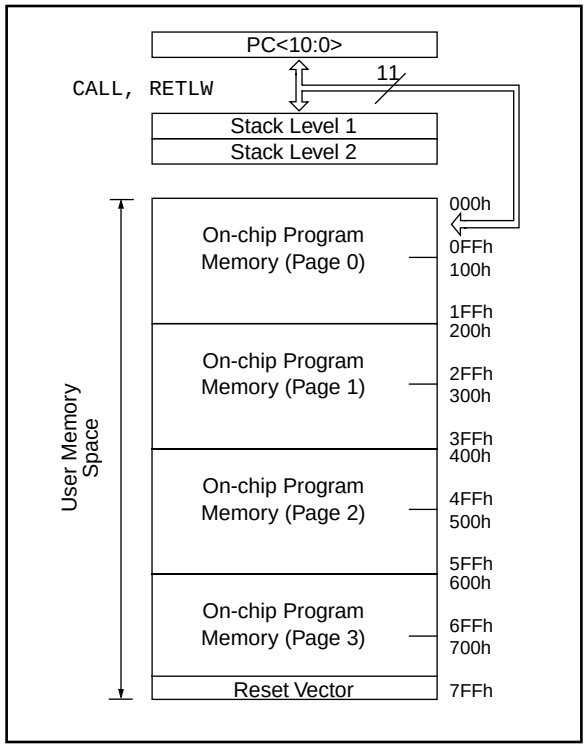


FIGURE 2: PIC16C58A PROGRAM MEMORY MAP AND STACK



© 1997 Microchip Technology Inc.

FIGURE 3: LOADING OF PIC16C54A PC FOR BRANCH INSTRUCTIONS

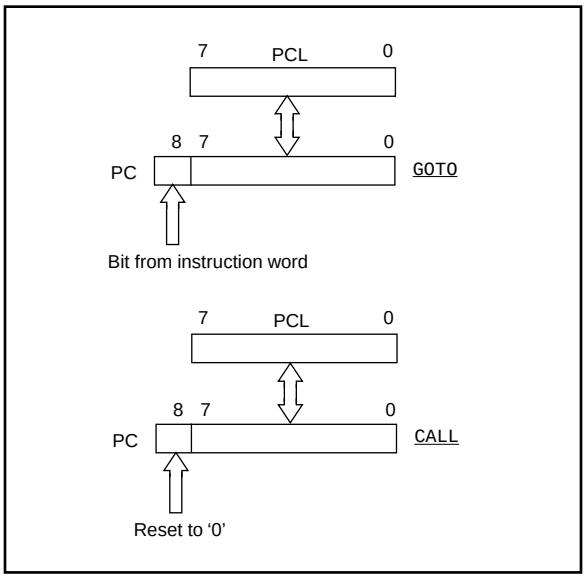
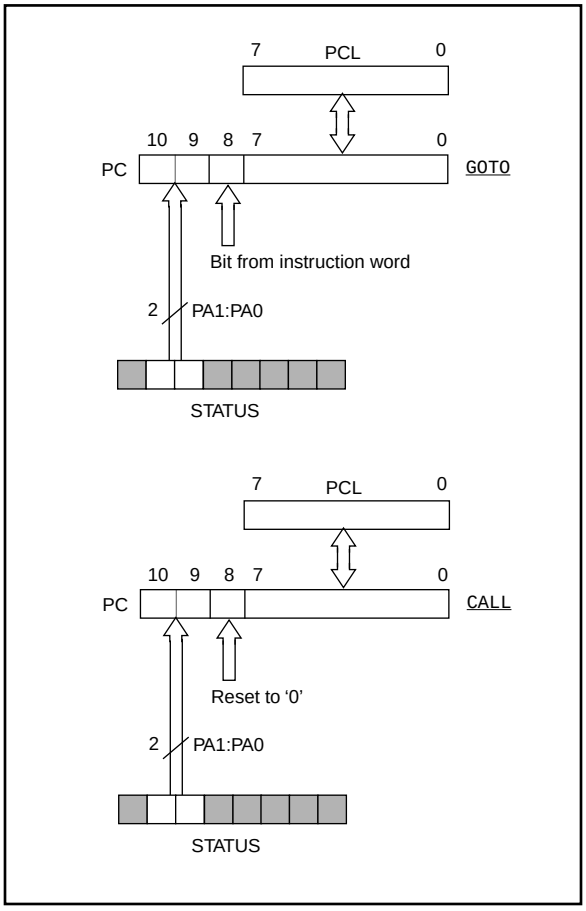


FIGURE 4: LOADING OF PIC16C58A PC FOR BRANCH INSTRUCTIONS



Reset Address

The reset address on the PIC16C54A is at location 1FFh. Normally, a "GOTO Start" instruction is placed at this location, where Start is the beginning of the program. If the user has used such an instruction, then in order for the program to operate correctly on a PIC16C58A the "GOTO Start" instruction should be moved to location 7FFh, the reset address for the PIC16C58A.

Operation in a Noisy Environment

What happens if the PA1 or PA0 bit value gets corrupted due to noise or some other external event? The PA1:PA0 bits are not directly mapped into PC<11:9>, but the PA1:PA0 bit values are loaded into PC<11:9> during a CALL or GOTO instruction. So, if the PA1:PA0 bit values get corrupted (ex: PA1:PA0 = 11 instead of 00), the program will go to a program memory page other than 0 when a CALL or GOTO instruction occurs. The program will then start executing "unimplemented" code space.

Normally the user would program "unimplemented" code space with the default FFFh value (XORLW FFh instruction). The PC would increment sequentially until it hit the last location (7FFh) or the Reset address, which is normally recommended to be a "GOTO Start" instruction. However, since the value in PA1:PA0 is still corrupted (PA1:PA0 = 11 in our example), the program execution will still be in Page 3. In other words, the program would get locked in an endless loop.

In order to avoid the above situation, each "unimplemented" page should be filled with the instruction "GOTO Recover". The Recover subroutines (Three will be required for the three (3) unimplemented pages of the PIC16C58A) should clear the PA1:PA0 bits in the STATUS register and execute a "GOTO Start" instruction.

EXAMPLE 1: RECOVER CODE

```
RECOVER  BCF STATUS, PA1
         BCF STATUS, PA0
         GOTO START
```

Please note that operation in a noisy environment is very application-dependent and may not affect the majority of the users. However, due to the presence of extra code space on the PIC16C58A, additional steps may have to be taken as mentioned above. Please check Appendix A for an example of the code implementation.

© 1997 Microchip Technology Inc.

DATA MEMORY

Figure 5 and Figure 6 depict the data memory registers for the PIC16C54A and the PIC16C58A. Banks 1, 2 and 3, which are not present in the PIC16C54A, are R/W memory RAM locations on the PIC16C58A. These locations are accessed using the FSR<6:5> bits. On the PIC16C54A, these bit values are ignored. However, they are not cleared during a power-up reset on any PIC16C5XA device. Therefore, the FSR<6:5> bits on the PIC16C58A should be cleared at the very start of the program and always maintained in that manner during the course of the program.

Operating in a Noisy Environment

The FSR<6:5> bits can get corrupted due to noise or some external conditions just like the PA1:PA0 bits of program memory. As part of that recovery, the user should also clear the FSR<6:5> bits. Please see Appendix A for a code example.

SUMMARY

Code transfer from a PIC16C54A to a PIC16C58A may be accomplished easily by remembering the following points:

- The reset address has to be re-located from 1FFh to 7FFh.
- The PA1:PA0 bits of the STATUS register must be kept cleared.
- The FSR<6:5> bits must be kept cleared.

FIGURE 5: PIC16C54A REGISTER FILE MAP

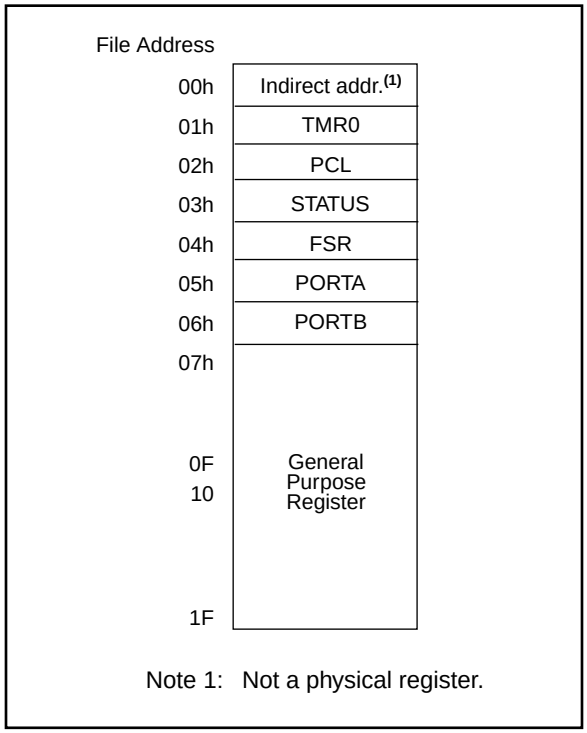
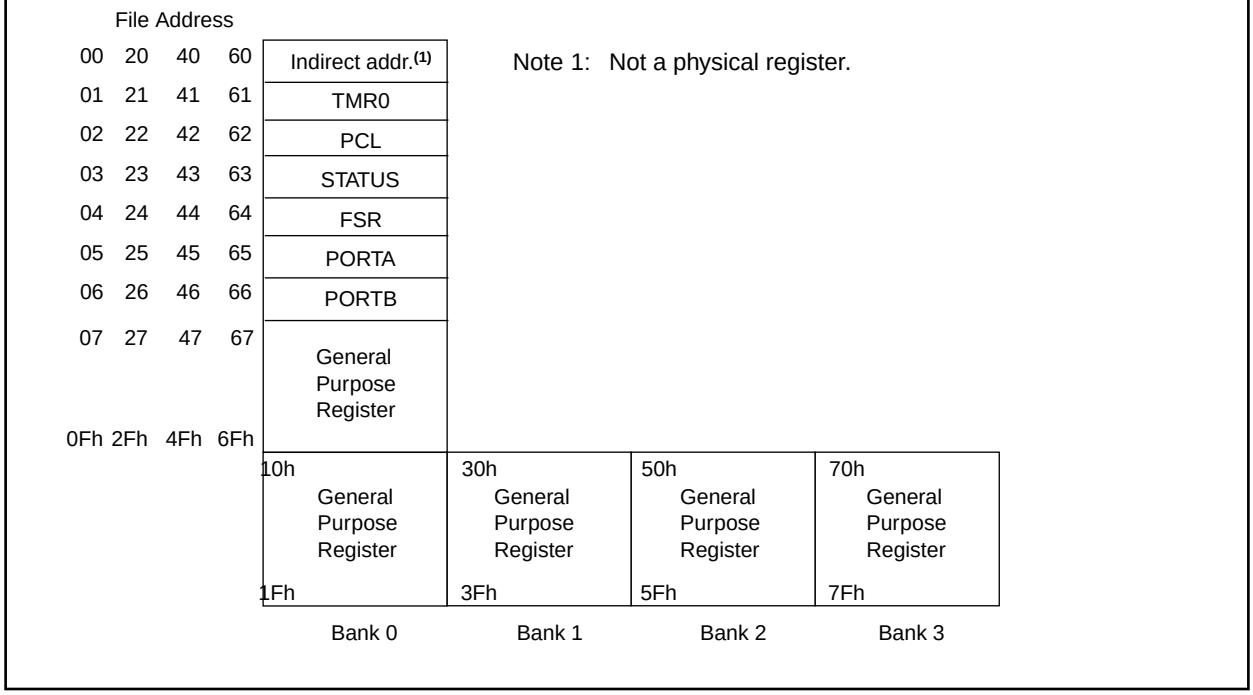


FIGURE 6: PIC16C58A REGISTER FILE MAP



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).

APPENDIX A: P54TO58.ASM

```
MPASM 01.21 Released          P54TO58.ASM   8-25-1995  12:33:54         PAGE  1

LOC  OBJECT CODE     LINE SOURCE TEXT
  VALUE

                     00001 ;This program is a code example of a program written for a PIC16C54A
                     00002 ;and modified for use in a PIC16C58A.
                     00003 ;
                     00004 ;In this program the following modifications have been done:
                     00005 ;1. The reset address is relocated from 0x1ff to 0x7ff
                     00006 ;2. The FSR<6,5> bits are initialized to 0 at the start of the program
                     00007 ;3. The Recover subroutine implemented in each “un-implemented”
                     00008 ;   page of the PIC16C58A.
                     00009 ;
                     00010 ;                                              by Stan D’Souza 8/29/95.
                     00011 ;
                     00012 ;
                     00013         list p=16C58A, f=inhx8m
                     00014 ;
                     00015 ;This program written for the PIC16C54A and operates on the PICDEM1
                     00016 ;demo board. Program has been modified to work on a PIC16C58A.
                     00017 ;The program blinks each of the 8 LEDs on PortB twice, then moves to the
                     00018 ;next LED and so on. If anytime the RA1 key is pressed, the direction
                     00019 ;of rotation is changed.
                     00020 ;
                     00021 ;
  0006               00022 PORTB   equ             6
  0005               00023 PORTA   equ             5
  0003               00024 STATUS  equ             3
  0004               00025 FSR     equ             4
  0005               00026 PA0     equ             5
  0006               00027 PA1     equ             6
  0000               00028 W       equ             0
  0001               00029 F       equ             1
  0000               00030 C       equ             0
  001B               00031 pb_buf  equ             0x1b
  001C               00032 count   equ             0x1c
  001D               00033 templ   equ             0x1d
  001E               00034 temph   equ             0x1e
  001F               00035 flag    equ             0x1f
                     00036 #define KeyPressed      flag,7
                     00037 #define KeyReleased     flag,6
                     00038 #define LtoR            flag,5
                     00039 ;
                     00040 ;*****************************************************************
                     00041 ;       org     0x1ff           ;reset vector for PIC16C54A, comment out
                     00042                                 ;when moving to a PIC16C58A
                     00043 ;*****************************************************************
07FF                 00044         org     0x7ff           ;added as PIC16C58A reset vector
07FF 0A10            00045         goto    Start
                     00046 ;*****************************************************************
                     00047 ;The fill command is used to load the unused 1.5K EPROM space on the
                     00048 ;PIC16C58A with a “goto Recover” subroutine. If the program ever
                     00049 ;enters the unused 1.5K space, this command will immediately and
                     00050 ;automatically cause the program execution to go back to the first page
                     00051 ;at address 0x00 to 0x1ff.
0200                 00052         org     0x200
0200                 00053 Recover1
0200 04C3            00054         bcf     STATUS,PA1
0201 04A3            00055         bcf     STATUS,PA0
```


© 1997 Microchip Technology Inc.

```
0202 0064        00056         clrf    FSR
0203 0A10        00057         goto    Start
0204 0A00        00058         FILL    (goto Recover1), (0x3ff - $ + 1)
                 00059         ;Fill all unused locations in page 1 with goto Recover1
                 00060 ;
                 00061 ;
0400             00062         org     0x400
0400             00063 Recover2
0400 04C3        00064         bcf     STATUS,PA1
0401 04A3        00065         bcf     STATUS,PA0
0402 0064        00066         clrf    FSR
0403 0A10        00067         goto    Start
0404 0A00        00068         FILL    (goto Recover2), (0x5ff - $ + 1)
                 00069         ;Fill all unused locations in page 2 with goto Recover2
                 00070 ;
                 00071 ;
                 00072 ;
0600             00073         org     0x600
0600             00074 Recover3
0600 04C3        00075         bcf     STATUS,PA1
0601 04A3        00076         bcf     STATUS,PA0
0602 0064        00077         clrf    FSR
0603 0A10        00078         goto    Start
0604 0A00        00079         FILL    (goto Recover3), (0x7fe - $ + 1)
                 00080         ;Fill all unused locations in page 3 with goto Recover3
                 00081 ;
                 00082 ;
0010             00083         org     0x10
0010             00084 Start
                 00085 ;*****************************************************************
0010 0064        00086         clrf    FSR             ;initialize the FSR register
                 00087                                 ;added for the PIC16C58A compatibility
                 00088 ;*****************************************************************
0011 0040        00089         clrw
0012 0026        00090         movwf   PORTB           ;set Port B as output and low
0013 0006        00091         tris    PORTB           ;       /
0014 007B        00092         clrf    pb_buf          ;clear buffer
0015 051B        00093         bsf     pb_buf,0        ;set up the first blink
0016 007F        00094         clrf    flag            ;clr flags
0017 05DF        00095         bsf     KeyReleased     ;       /
0018             00096 Repeat
0018 021B        00097         movf    pb_buf,W
0019 0026        00098         movwf   PORTB
001A 092F        00099         call    delay500        ;delay for 500mS
001B 0066        00100         clrf    PORTB
001C 092F        00101         call    delay500        ;
001D 06FF        00102         btfsc   KeyPressed      ;if no key then skip
001E 0A46        00103         goto    ChangeDirection ;else change direction
001F 021B        00104         movf    pb_buf,W
0020 0026        00105         movwf   PORTB
0021 092F        00106         call    delay500        ;delay for 500mS
0022 0066        00107         clrf    PORTB
0023 092F        00108         call    delay500        ;
0024 06FF        00109         btfsc   KeyPressed      ;if no key then skip
0025 0A46        00110         goto    ChangeDirection
0026             00111 Rotate
0026 0403        00112         bcf     STATUS,C        ;clr carry
0027             00113 RotAgain
0027 07BF        00114         btfss   LtoR            ;left to right?
0028 0A2D        00115         goto    RotateLeft      ;rotate Port B
0029 033B        00116         rrf     pb_buf,F        ;rotate Port B
002A             00117 Done
002A 0703        00118         btfss   STATUS,C        ;carry over?
002B 0A18        00119         goto    Repeat          ;no then do again
002C 0A27        00120         goto    RotAgain        ;rotate again
002D             00121 RotateLeft
```

© 1997 Microchip Technology Inc.

```
002D 037B        00122         rlf     pb_buf,F        ;rotate left
002E 0A2A        00123         goto    Done            ;check
                 00124 ;
002F             00125 delay500
002F 0C19        00126         movlw   .25             ;reload count
0030 003C        00127         movwf   count           ;       /
0031             00128 dly500lp
0031 06DF        00129         btfsc   KeyReleased     ;key released?
0032 0938        00130         call    CheckReleased   ;no then check?
0033 093E        00131         call    CheckPressed    ;else check if pressed?
0034 094E        00132         call    delay20         ;delay for 20mS
0035 02FC        00133         decfsz  count           ;see if delay over
0036 0A31        00134         goto    dly500lp        ;no then loop
0037 0800        00135         retlw   0               ;return
                 00136 ;
0038             00137 CheckReleased
0038 0625        00138         btfsc   PORTA,1         ;RA1 low?
0039 0A3B        00139         goto    RelAgain        ;no then check again
003A 0800        00140         retlw   0               ;go back
003B             00141 RelAgain
003B 0625        00142         btfsc   PORTA,1         ;RA1 low?
003C 05DF        00143         bsf     KeyReleased     ;no then key released
003D 0800        00144         retlw   0
                 00145 ;
003E             00146 CheckPressed
003E 06FF        00147         btfsc   KeyPressed      ;flag already set?
003F 0800        00148         retlw   0               ;yes then return
0040 0725        00149         btfss   PORTA,1         ;see if key low?
0041 0A43        00150         goto    PressAgain      ;check again
0042 0800        00151         retlw   0               ;else go back
0043             00152 PressAgain
0043 0725        00153         btfss   PORTA,1         ;see if low
0044 05FF        00154         bsf     KeyPressed      ;yes then set flag
0045 0800        00155         retlw   0
                 00156 ;
0046             00157 ChangeDirection
0046 04FF        00158         bcf     KeyPressed      ;key serviced
0047 04DF        00159         bcf     KeyReleased     ;see if key released
0048 07BF        00160         btfss   LtoR            ;check if Left to Rt.
0049 0A4C        00161         goto    TurnLeft        ;make it go left
004A 04BF        00162         bcf     LtoR
004B 0A26        00163         goto    Rotate
004C             00164 TurnLeft
004C 05BF        00165         bsf     LtoR
004D 0A26        00166         goto    Rotate
                 00167
                 00168 ;
                 00169 ;Delay loop for 20 mS
004E             00170 delay20
004E 0C14        00171         movlw   .20             ;load temph
004F 003E        00172         movwf   temph
0050             00173 dly4
0050 0CC8        00174         movlw   .200            ;load templ
0051 003D        00175         movwf   templ
0052             00176 dly
0052 0000        00177         nop
0053 0000        00178         nop
0054 02FD        00179         decfsz  templ
0055 0A52        00180         goto    dly
0056 02FE        00181         decfsz  temph
0057 0A50        00182         goto    dly4
0058 0800        00183         retlw   0
                 00184 ;
                 00185 ;
                 00186 ;
                 00187         end
```


© 1997 Microchip Technology Inc.

```
All other memory blocks unused.


Errors   :    0
Warnings :    0
Messages :    0
```

© 1997 Microchip Technology Inc.

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.

Printed on recycled paper.

DNV Certification, Inc. USA — ANSI-RAB QMS ACCREDITED — DNV MSC The Netherlands Accredited by the RvA — DNV — 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. 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 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
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

01/18/02

© 2002 Microchip Technology Inc.