

Techniques to Disable Global Interrupts

*Author: Mark Palmer
Microchip Technology Inc.
Contributions by Martin Burghardt
Manager Applications (Central Europe)*

INTRODUCTION

This application note discusses four methods for disabling global interrupts. The method best suited for the application may then be used. All discussion will be specific to the PIC16CXXX family of products, but these concepts are also applicable to the PIC17C42, and are shown in the even numbered examples. Note that the PIC17C42's global interrupt bit is called GLINTD and has an inverse sense compared to the GIE bit of the PIC16CXXX family.

To disable all interrupts, either the Global Interrupt Enable (GIE) bit must be cleared or all the individual interrupt enable bits must be cleared. An issue arises when an instruction clears the GIE bit and an interrupt occurs "simultaneously". For example, when a program executes the instruction `BCF INTCON, GIE` (at address PC), there is a possibility that an interrupt will occur during this instruction. If an interrupt occurs during this instruction, the program would complete execu-

tion of this instruction, and then immediately branch to the user's interrupt service routine. This occurs because the GIE bit was not clear (disabled) when the interrupt occurred. Normally at the end of the interrupt service routine is the `RETFIE` instruction. This instruction causes the program to return to the instruction at PC + 1, but also sets the GIE bit (enabled). Therefore the GIE bit is not cleared as expected, and unintended program execution may occur.

One method to ensure that the GIE bit is cleared is shown in Example 1 and Example 2, as well as in the PIC16CXXX data sheets. This method tests the state of the GIE bit, after clearing, to ensure that it was not accidentally set in the user's interrupt service routine by the `RETFIE` instruction. If the GIE bit was accidentally set, the program branches back to the instruction that clears the GIE bit.

In this method, the time to ensure that the GIE bit is cleared is indeterminate. Depending on the frequency of the enabled interrupts during this code segment, unexpected delays into the following code segment may occur. For some applications, this may be undesirable. The following three methods address this issue.

EXAMPLE 1: CLEARING THE GIE BIT (DISABLING INTERRUPTS, METHOD 1, PIC16CXXX)

```

LOOP   BCF   INTCON, GIE   ; Disable Global Interrupt
       BTFSC INTCON, GIE   ; Global Interrupt Disabled?
       GOTO  LOOP         ; NO, try again
       :                 ; YES, continue with program flow
       :
       BSF   INTCON, GIE   ; Re-enable Global Interrupt
  
```

EXAMPLE 2: SETTING THE GLINTD BIT (DISABLING INTERRUPTS, METHOD 1, PIC17C42)

```

:
LOOP   BSF   CPUSTA, GLINTD ; Disable Global Interrupt
       BTFSS CPUSTA, GLINTD ; Global Interrupt Disabled?
       GOTO  LOOP         ; NO, try again
       :                 ; YES, continue with program flow
       :
       BCF   CPUSTA, GLINTD ; Re-enable Global Interrupt
  
```

AN576

The second method is to disable the individual interrupt enable bits. If it is known which bits are enabled at this point, it can easily be done. Example 3 and Example 4 show the disabling of interrupts, where it is known which sources are enabled (some peripheral interrupts and the T0CKI pin interrupt).

This method also requires the same number of instructions for the disabling/enabling of interrupts, as method 1, but requires a knowledge of which individual interrupt enable bits need to be disabled and (more importantly) re-enabled. The major advantage of this method is that it can minimize the time delay entering the code segment which follows the point where interrupts are disabled.

EXAMPLE 3: CLEARING KNOWN INDIVIDUAL INTERRUPT ENABLE BITS (METHOD 2, PIC16CXXX)

```
:
MOVLW  b'10011111'  ; Disable Peripheral and T0CKI pin interrupts,
ANDWF  INTCON, F    ; All other bits unchanged
:
:
:
MOVLW  b'01100000'  ; Re-enable Peripheral and T0CKI pin interrupts,
IORWF  INTCON, F    ; All other bits unchanged
:
```

EXAMPLE 4: CLEARING KNOWN INDIVIDUAL INTERRUPT ENABLE BITS (METHOD 2, PIC17C42)

```
:
MOVLW  b'11110011'  ; Disable Peripheral and T0CKI pin interrupts,
ANDWF  INTSTA, F    ; All other bits unchanged
:
:
:
MOVLW  b'00001100'  ; Re-enable Peripheral and T0CKI pin interrupts,
IORWF  INTSTA, F    ; All other bits unchanged
:
```

Method 3 can be used if the states of the individual interrupt enable bits are unknown. A temporary byte of data RAM is required to store the value of the INTCON register. This method is shown in Example 5 and Example 6.

This method also requires more instructions for the disabling/enabling of interrupts than in method 1 or method 2, and also a byte of data RAM to temporarily store the value of the INTCON register. The major advantage of this method is that it minimizes the time delay into the code segment which follows the point where interrupts are disabled.

EXAMPLE 5: CLEARING THE INDIVIDUAL INTERRUPT ENABLE BITS (METHOD 3, PIC16CXXX)

```
:
MOVWF  INTCON, W           ; Move the value in INTCON to
MOVWF  S_INTCON           ; a shadow register
MOVLW  b'10000111'       ; Disable all individual interrupts,
ANDWF  INTCON, F         ; All other bits unchanged
:
:
:
MOVWF  S_INTCON, W       ; Restore the INTCON register
IORWF  INTCON, F         ;
:
```

EXAMPLE 6: CLEARING THE INDIVIDUAL INTERRUPT ENABLE BITS (METHOD 3, PIC17C42)

```
:
MOVFPF INTSTA, S_INTSTA  ; Move the value in INTSTA to a shadow register
MOVLW  b'11110000'       ; Disable all individual interrupts,
ANDWF  INTSTA, F         ; All other bits unchanged
:
:
:
MOVFPF S_INTSTA, W       ; Restore the INTSTA register
IORWF  INTSTA, F         ;
:
```

AN576

The final method is to use a RAM location to “shadow” the value of the GIE bit. This shadow bit can then be used in the interrupt service routine to determine which return instruction to use. That is, either the RETURN or the RETFIE (which enables the GIE bit) instruction. Example 7 and Example 8 show this implementation, which require that a general purpose bit be available to hold the “shadow” GIE value. In these examples, the shadow GIE (S_GIE) bit is contained in the register FLAG_REG. If an interrupt occurs during the clearing of the shadow GIE, the interrupt is responded to. At the end of the interrupt service routine, the shadow GIE bit is cleared so the RETURN instruction is executed. The GIE bit remains disabled and program execution returns to the instruction which tries to clear the GIE bit

(disable). No interrupts can occur during this instruction since the GIE bit was not re-enabled after the interrupt service routine.

This method also requires more instructions for the disabling/enabling of interrupts than in method 1 or method 2, and a single bit of data RAM to temporarily store the value of the desired GIE value, and increases the interrupt service routine execution time by one instruction cycle, for most occurrences of interrupts (two cycles worst case). The major advantage of this method is that it minimizes the time delay into the code segment which follows the point where interrupts are disabled. Also, the individual interrupt enable bits need not be modified.

EXAMPLE 7: THE “SHADOW” GIE BIT (METHOD 4, PIC16CXXX)

```
:
      org      0x004
INT_SERVICE_ROUTINE
:
:
      BTFSC   FLAG_REG, S_GIE      ; Is the S_GIE bit enabled?
      RETFIE                      ; YES, the GIE should be enabled
      RETURN                      ; NO, the GIE should be disabled
END_INT_SERVICE_ROUTINE
;
MAIN:
:
:
      BCF     FLAG_REG, S_GIE      ; Clear the shadow GIE bit
      BCF     INTCON, GIE          ; Disable interrupts by clearing the GIE bit
:
:
:
      BSF     FLAG_REG, S_GIE      ; Set the shadow GIE bit
      BSF     INTCON, GIE          ; Enable interrupts by setting the GIE bit
:
:
:
      END
```

EXAMPLE 8: THE “SHADOW” GLINTD BIT (METHOD 4, PIC17C42)

```
:
      org      0x004
INT_SERVICE_ROUTINE
:
:
      BTFSS   FLAG_REG, S_GLINTD   ; Is the S_GLINTD bit enabled?
      RETFIE                      ; YES, the GLINTD should be enabled
      RETURN                      ; NO, the GLINTD should be disabled
END_INT_SERVICE_ROUTINE
;
MAIN:
:
:
      BSF     FLAG_REG, S_GLINTD   ; Set the shadow GLINTD bit
      BSF     CPUSTA, GLINTD       ; Disable interrupts by setting the GLINTD bit
:
:
:
      BCF     FLAG_REG, S_GLINTD   ; Clear the shadow GLINTD bit
      BCF     CPUSTA, GLINTD       ; Enable interrupts by clearing the GLINTD bit
:
:
:
      END
```

CONCLUSION

In conclusion, different methods exist to ensure that all interrupts are disabled. The requirement(s) of the application determines which of the methods is the best fit. A comparison of the different methods is shown in Table 1.

TABLE 1: COMPARISON OF DIFFERENT METHODS

	Program Memory	Data Memory	Cycle Delay (Tcy)	
			Best Case	Worst Case
Method 1	2 words * N	—	2	Indeterminate
Method 2	2 words * N	—	1	1 + TISR
Method 3 - PIC16CXXX - PIC17C42	4 words * N	1 byte	3	3 + TISR
	3 words * N	1 byte	2	2 + TISR
Method 4	2 words * N + 2 words	1 bit	1†	1 + (TISR + 2)

Legend: N - Number of occurrences to disable / re-enable interrupts.
 TISR - Time to execute the interrupt service routine.
 † This method increases the interrupt service routine time (TISR) by 1 cycle for most occurrences (2 cycles worst case).

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.

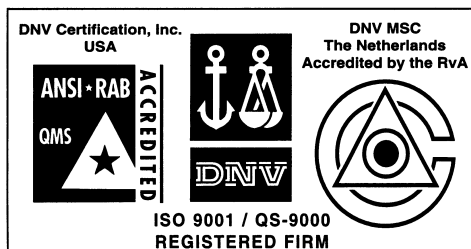
dsPIC, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, microPort, Migratable Memory, MPASM, MPLIB, MPLINK, MPSIM, MXDEV, PICC, PICDEM, PICDEM.net, rPIC, Select Mode and Total Endurance are trademarks of Microchip Technology Incorporated in the U.S.A.

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.



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.



MICROCHIP

WORLDWIDE SALES AND SERVICE

AMERICAS

Corporate Office

2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200 Fax: 480-792-7277
Technical Support: 480-792-7627
Web Address: <http://www.microchip.com>

Rocky Mountain

2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7966 Fax: 480-792-7456

Atlanta

500 Sugar Mill Road, Suite 200B
Atlanta, GA 30350
Tel: 770-640-0034 Fax: 770-640-0307

Boston

2 Lan Drive, Suite 120
Westford, MA 01886
Tel: 978-692-3848 Fax: 978-692-3821

Chicago

333 Pierce Road, Suite 180
Itasca, IL 60143
Tel: 630-285-0071 Fax: 630-285-0075

Dallas

4570 Westgrove Drive, Suite 160
Addison, TX 75001
Tel: 972-818-7423 Fax: 972-818-2924

Detroit

Tri-Atria Office Building
32255 Northwestern Highway, Suite 190
Farmington Hills, MI 48334
Tel: 248-538-2250 Fax: 248-538-2260

Kokomo

2767 S. Albright Road
Kokomo, Indiana 46902
Tel: 765-864-8360 Fax: 765-864-8387

Los Angeles

18201 Von Karman, Suite 1090
Irvine, CA 92612
Tel: 949-263-1888 Fax: 949-263-1338

New York

150 Motor Parkway, Suite 202
Hauppauge, NY 11788
Tel: 631-273-5305 Fax: 631-273-5335

San Jose

Microchip Technology Inc.
2107 North First Street, Suite 590
San Jose, CA 95131
Tel: 408-436-7950 Fax: 408-436-7955

Toronto

6285 Northam Drive, Suite 108
Mississauga, Ontario L4V 1X5, Canada
Tel: 905-673-0699 Fax: 905-673-6509

ASIA/PACIFIC

Australia

Microchip Technology Australia Pty Ltd
Suite 22, 41 Rawson Street
Epping 2121, NSW
Australia
Tel: 61-2-9868-6733 Fax: 61-2-9868-6755

China - Beijing

Microchip Technology Consulting (Shanghai)
Co., Ltd., Beijing Liaison Office
Unit 915
Bei Hai Wan Tai Bldg.
No. 6 Chaoyangmen Beidajie
Beijing, 100027, No. China
Tel: 86-10-85282100 Fax: 86-10-85282104

China - Chengdu

Microchip Technology Consulting (Shanghai)
Co., Ltd., Chengdu Liaison Office
Rm. 2401, 24th Floor,
Ming Xing Financial Tower
No. 88 TIDU Street
Chengdu 610016, China
Tel: 86-28-6766200 Fax: 86-28-6766599

China - Fuzhou

Microchip Technology Consulting (Shanghai)
Co., Ltd., Fuzhou Liaison Office
Unit 28F, World Trade Plaza
No. 71 Wusi Road
Fuzhou 350001, China
Tel: 86-591-7503506 Fax: 86-591-7503521

China - Shanghai

Microchip Technology Consulting (Shanghai)
Co., Ltd.
Room 701, Bldg. B
Far East International Plaza
No. 317 Xian Xia Road
Shanghai, 200051
Tel: 86-21-6275-5700 Fax: 86-21-6275-5060

China - Shenzhen

Microchip Technology Consulting (Shanghai)
Co., Ltd., Shenzhen Liaison Office
Rm. 1315, 13/F, Shenzhen Kerry Centre,
Renminnan Lu
Shenzhen 518001, China
Tel: 86-755-2350361 Fax: 86-755-2366086

Hong Kong

Microchip Technology Hongkong Ltd.
Unit 901-6, Tower 2, Metroplaza
223 Hing Fong Road
Kwai Fong, N.T., Hong Kong
Tel: 852-2401-1200 Fax: 852-2401-3431

India

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

Japan

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

Korea

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

Singapore

Microchip Technology Singapore Pte Ltd.
200 Middle Road
#07-02 Prime Centre
Singapore, 188980
Tel: 65-334-8870 Fax: 65-334-8850

Taiwan

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

EUROPE

Denmark

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

France

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

Germany

Microchip Technology GmbH
Gustav-Heinemann Ring 125
D-81739 Munich, Germany
Tel: 49-89-627-144 0 Fax: 49-89-627-144-44

Italy

Microchip Technology SRL
Centro Direzionale Colleoni
Palazzo Taurus 1 V. Le Colleoni 1
20041 Agrate Brianza
Milan, Italy
Tel: 39-039-65791-1 Fax: 39-039-6899883

United Kingdom

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

01/18/02