# AN514

# Software Interrupt Techniques

## INTRODUCTION

This application note describes a unique method for implementing interrupts in software on the PIC16C5X series of microcontrollers. This method takes advantage of the PIC16C5X's architecture which allows changing the program counter under software control. Up to eight interrupt lines are possible, but the practical limit for simple code generation is six interrupts, or 64 possible input conditions. The interrupt detection time is under software control and standard I/O pins are used as the interrupt lines.

## THEORY OF OPERATION

### SOFTWARE POLLING OF I/O LINES TO REPLACE HARDWARE INTERRUPTS

The interrupt conditions are determined by detecting changes on the I/O lines that have been selected to be the interrupt lines. These changes are used to create a jump table that allows a different program response to each interrupt condition. The interrupt response time is under software control and can be as short as ten to twenty microseconds, depending on main program and interrupt subroutine program length.

### CREATING THE INTERRUPT SUBROUTINE JUMP TABLE

Each I/O condition may have its own unique subroutine to respond to changes on the interrupt lines. Direct access to these routines is achieved by using the PIC16C5X's ability to change the program counter under software control. Here is an example of how two I/O lines may be polled:

```
MOVF   CONDTN,W ;LOAD I/O CONDITION INTO W
                ;REGISTER
ANDLW  3        ;MASK OFF TOP 6 BITS
ADDWF  2,1      ;ADD INPUT TO PROGRAM COUNTER
                ;TO CREATE JUMP TABLE
GOTO   MAIN     ;FOR NO CHANGE GOTO MAIN
                ;PROGRAM
GOTO   INT1     ;FOR CHANGE IN BIT 0 GOTO INT1
GOTO   INT2     ;FOR CHANGE IN BIT 1 GOTO INT2
GOTO   INT3     ;FOR BOTH CHANGES GOTO INT3
```

The changes to the I/O lines have been used to create a two bit number that is added to the program counter. The GOTO that is executed depends on the new program counter address.

## CREATING CONSTANT TIME POLLING

In most applications requiring interrupts, it is important to poll the interrupt lines at fixed time intervals, usually only a few microseconds in length. Two techniques may be used on the PIC16C5X to achieve this timing. One divides the main program into multiple sections and the other implements an elapsed time counter (Figure 1). Both of these techniques use the same program jump table concept that was described earlier. In the first technique, the main program is divided into several sections based on the desired I/O polling time. When MAIN is called a branch register is added to the program counter. This determines which section of MAIN code should be executed next. At the end of execution the branch register is decremented so the next section of code will be executed after the next polling. If the branch register is zero, then the number of sections of main code is added into it to start the main program over again.

In the second technique, an elapsed time counter can be implemented using the Timer0 counter. At the beginning of I/O polling the TMR0 register is cleared. It then starts counting the instruction cycles. Then after the main program subsection has been executed, the TMR0 register value is subtracted from the desired polling time. This determines how many instructions need to be executed before the next polling. A jump table is then created to execute these instructions before the next polling. An example is shown below. This example assumes from zero to 15 additional instruction cycles are needed. Actual numbers need to be computed for each individual application.

```
MOVLW  POLL  ;POLL=DESIRED POLL CYCLES-15
SUBWF  TMR0,W;DETERMINE HOW MUCH TIME TO WAIT
ADDWF  2,1   ;ADD WAIT TIME TO PROGRAM COUNTER
NOP          ;15 ADDITIONAL INSTRUCTION CYCLES
:
:            ;TOTAL OF 15 NOP'S
NOP          ;1 ADDITIONAL INSTRUCTION CYCLE
GOTO   START ;0 ADDITIONAL INSTRUCTION CYCLES
```

© 1997 Microchip Technology Inc.

For example, if the desired instruction time is 50 cycles and the subsection we just executed has a consumed a total of 40 instruction cycles (including all overhead cycles) the value of:

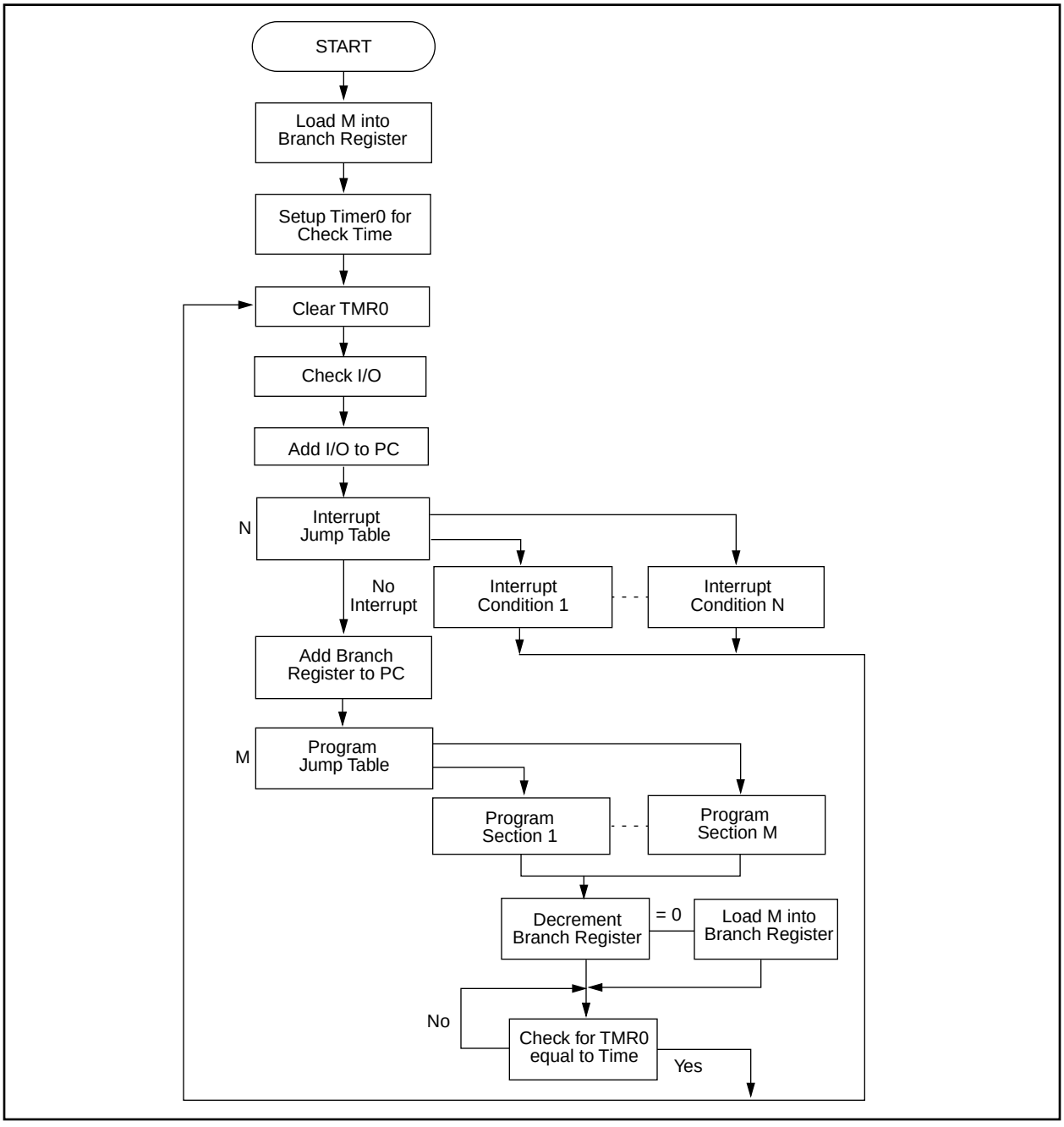
$$TMR0(40) - POLL(50 - 15(= 35)) = 5$$

will be added to the program counter. The program will then jump to the sixth NOP. That NOP, plus the nine following it, will be executed for a total of ten more instruction cycles. Note that the final `GOTO` has two instruction cycles and these must be included in the program overhead.

### Example

The following example (Figure 1 and Appendix A) is the core program for the software interrupt technique described previously. This program assumes four interrupt conditions, four main program sections, and eight additional elapsed time instructions.

**FIGURE 1: SOFTWARE INTERRUPT FLOWCHART**





© 1997 Microchip Technology Inc.

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: INTRUPT.ASM

```
MPASM 01.40 Released          INTRUPT.ASM   1-16-1997  12:37:20         PAGE  1


LOC  OBJECT CODE     LINE SOURCE TEXT
  VALUE

                     00001        TITLE 'SOFTWARE INTERRUPT PROGRAM REV 3-29-90'
                     00002        LIST    P=16C54
                     00003
                     00004 ;
                     00005 ;*****************************************************************
                     00006 ;
                     00007 ;    SOFTWARE INTERRUPT APPLICATIONS
                     00008 ;    BRANCH IS MAIN PROGRAM REGISTER
                     00009 ;
                     00010 ;       Program:          OHMETER.ASM
                     00011 ;       Revision Date:
                     00012 ;                         1-13-97      Compatibility with MPASMWIN 1.40
                     00013 ;
                     00014 ;*****************************************************************
                     00015 ;
  00000008           00016 BRANCH EQU     8
  00000009           00017 CNDTN  EQU     9
  0000000A           00018 IO     EQU     0A
  0000000B           00019 TEMP   EQU     0B
                     00020
0000 0069            00021 SETUP  CLRF    CNDTN
0001 0C04            00022        MOVLW   4
0002 0028            00023        MOVWF   BRANCH   ;FOUR MAIN PROGRAM SECTIONS
0003 0C08            00024        MOVLW   8
0004 0002            00025        OPTION           ;SET TMR0 TO ONE COUNT PER INSTRUCTION CYCLE
                     00026
0005 0061            00027 START  CLRF    1        ;CLEAR TMR0 REGISTER
0006 0206            00028        MOVF    6,W      ;READ I/O
0007 002A            00029        MOVWF   IO
0008 0109            00030        IORWF   CNDTN,W  ;THIS SECTION OF CODE CALCULATES THE
0009 002B            00031        MOVWF   TEMP     ;JUMP TABLE. ANY INPUT THAT CHANGES FROM
000A 0209            00032        MOVF    CNDTN,W  ;A ZERO TO A ONE IS CONSIDERED AN INTERRUPT.
000B 00AB            00033        SUBWF   TEMP,1   ;THE EQUATION IS:
000C 020A            00034        MOVF    IO,W     ;  (IO + CNDTN) - CNDTN = INTERRUPT
000D 0029            00035        MOVWF   CNDTN    ;WHERE IO IS CURRENT INPUT AND
000E 020B            00036        MOVF    TEMP,W   ;CNDTN IS PREVIOUS INPUT.
000F 0E03            00037        ANDLW   3        ;MASK OFF TOP 6 BITS
0010 01E2            00038        ADDWF   2,1      ;ADD INPUT TO PC TO CREATE JUMP TABLE
0011 0A1B            00039        GOTO    MAIN     ;FOR INPUT=00
0012 0A15            00040        GOTO    INT1     ;FOR INPUT=01
0013 0A17            00041        GOTO    INT2     ;FOR INPUT=10
0014 0A19            00042        GOTO    INT3     ;FOR INPUT=11
                     00043
0015 0000            00044 INT1   NOP              ;INTERRUPT LINE 1 CODE
0016 0A05            00045        GOTO    START
0017 0000            00046 INT2   NOP              ;INTERRUPT LINE 2 CODE
0018 0A05            00047        GOTO    START
0019 0000            00048 INT3   NOP              ;INTERRUPT LINES 1 AND 2 CODE
001A 0A05            00049        GOTO    START
                     00050
001B 0208            00051 MAIN   MOVF    BRANCH,W
001C 01E2            00052        ADDWF   2,1      ;ADD BRANCH TO PC TO CREATE JUMP TABLE
001D 0000            00053        NOP
001E 0A28            00054        GOTO    MAIN4    ;JUMP TABLE, LAST FIRST ON DECREMENT TABLE
```

```
001F 0A26           00055        GOTO    MAIN3
0020 0A24           00056        GOTO    MAIN2
0021 0A22           00057        GOTO    MAIN1
                    00058
0022 0000           00059 MAIN1  NOP             ;MAIN PROGRAM CODE BANK ONE
0023 0A2A           00060        GOTO    BRNCHK
0024 0000           00061 MAIN2  NOP             ;MAIN PROGRAM CODE SECTION TWO
0025 0A2A           00062        GOTO    BRNCHK
0026 0000           00063 MAIN3  NOP             ;MAIN PROGRAM CODE SECTION THREE
0027 0A2A           00064        GOTO    BRNCHK
0028 0000           00065 MAIN4  NOP             ;MAIN PROGRAM CODE SECTION FOUR
0029 0A2A           00066        GOTO    BRNCHK
                    00067
002A 02E8           00068 BRNCHK DECFSZ  BRANCH,1 ;DECREMENT BRANCH REGISTER AND CHECK FOR ZERO
002B 0A2E           00069        GOTO    TIMCHK
002C 0C04           00070        MOVLW   4
002D 0028           00071        MOVWF   BRANCH   ;RELOAD BRANCH WITH 4 AT END OF MAIN
                    00072
002E 0C29           00073 TIMCHK MOVLW   D'41'    ;CHECK TO SEE IF TMR0 HAS REACHED 50(50-7)
002F 0081           00074        SUBWF   1,W      ;DETERMINE WAIT TIME
0030 01E2           00075        ADDWF   2,1      ;ADD WAIT TIME TO PC
0031 0000           00076        NOP
0032 0000           00077        NOP
0033 0000           00078        NOP
0034 0000           00079        NOP
0035 0000           00080        NOP
0036 0000           00081        NOP
0037 0000           00082        NOP
0038 0A05           00083        GOTO    START
                    00084        END

0000 : XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXX XXXXXXXXX-------

All other memory blocks unused.

Program Memory Words Used:    57
Program Memory Words Free:   455


Errors   :     0
Warnings :     0 reported,     0 suppressed
Messages :     0 reported,     0 suppressed
```


© 1997 Microchip Technology Inc.

# WORLDWIDE SALES AND SERVICE

## AMERICAS

**Corporate Office**
Microchip Technology Inc.
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-786-7200 Fax: 480-786-7277
Technical Support: 480-786-7627
Web Address: http://www.microchip.com

**Atlanta**
Microchip Technology Inc.
500 Sugar Mill Road, Suite 200B
Atlanta, GA 30350
Tel: 770-640-0034 Fax: 770-640-0307

**Boston**
Microchip Technology Inc.
5 Mount Royal Avenue
Marlborough, MA 01752
Tel: 508-480-9990 Fax: 508-480-8575

**Chicago**
Microchip Technology Inc.
333 Pierce Road, Suite 180
Itasca, IL 60143
Tel: 630-285-0071 Fax: 630-285-0075

**Dallas**
Microchip Technology Inc.
4570 Westgrove Drive, Suite 160
Addison, TX 75248
Tel: 972-818-7423 Fax: 972-818-2924

**Dayton**
Microchip Technology Inc.
Two Prestige Place, Suite 150
Miamisburg, OH 45342
Tel: 937-291-1654 Fax: 937-291-9175

**Detroit**
Microchip Technology Inc.
Tri-Atria Office Building
32255 Northwestern Highway, Suite 190
Farmington Hills, MI 48334
Tel: 248-538-2250 Fax: 248-538-2260

**Los Angeles**
Microchip Technology Inc.
18201 Von Karman, Suite 1090
Irvine, CA 92612
Tel: 949-263-1888 Fax: 949-263-1338

**New York**
Microchip Technology Inc.
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

## AMERICAS (continued)

**Toronto**
Microchip Technology Inc.
5925 Airport Road, Suite 200
Mississauga, Ontario L4V 1W1, Canada
Tel: 905-405-6279 Fax: 905-405-6253

## ASIA/PACIFIC

**Hong Kong**
Microchip Asia Pacific
Unit 2101, Tower 2
Metroplaza
223 Hing Fong Road
Kwai Fong, N.T., Hong Kong
Tel: 852-2-401-1200 Fax: 852-2-401-3431

**Beijing**
Microchip Technology, Beijing
Unit 915, 6 Chaoyangmen Bei Dajie
Dong Erhuan Road, Dongcheng District
New China Hong Kong Manhattan Building
Beijing 100027 PRC
Tel: 86-10-85282100 Fax: 86-10-85282104

**India**
Microchip Technology Inc.
India Liaison Office
No. 6, Legacy, Convent Road
Bangalore 560 025, India
Tel: 91-80-229-0061 Fax: 91-80-229-0062

**Japan**
Microchip Technology Intl. Inc.
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
Tel: 82-2-554-7200 Fax: 82-2-558-5934

**Shanghai**
Microchip Technology
RM 406 Shanghai Golden Bridge Bldg.
2077 Yan'an Road West, Hong Qiao District
Shanghai, PRC 200335
Tel: 86-21-6275-5700 Fax: 86 21-6275-5060

## ASIA/PACIFIC (continued)

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

**Taiwan, R.O.C**
Microchip Technology Taiwan
10F-1C 207
Tung Hua North Road
Taipei, Taiwan, ROC
Tel: 886-2-2717-7175 Fax: 886-2-2545-0139

## EUROPE

**United Kingdom**
Arizona Microchip Technology Ltd.
505 Eskdale Road
Winnersh Triangle
Wokingham
Berkshire, England RG41 5TU
Tel: 44 118 921 5858 Fax: 44-118 921-5835

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

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

**Italy**
Arizona 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

11/15/99

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. 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.*

All rights reserved. © 1999 Microchip Technology Incorporated. Printed in the USA. 11/99 Printed on recycled paper.

Information contained in this publication regarding device applications and the like is intended for suggestion only and may be superseded by updates. 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. The Microchip logo and name are registered trademarks of Microchip Technology Inc. in the U.S.A. and other countries. All rights reserved. All other trademarks mentioned herein are the property of their respective companies.