Software Interrupt Techniques

INTRODUCTION

This application note describes a unique method for implementing interrupts in software on the PIC16C5X series of microcontrollers. The 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 REPLACES HARDWARE INTERRUPT

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/0 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 GO TO MAIN
                 ;PROGRAM
GOTO   INT1      ;FOR CHANGE IN BIT 0 GOTO INT1
GOTO   INT2      ;FOR CHANGE IN BIT 1 GOTO INT2
GOTO   INT3      ;FOR BOTH CHANGE 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. They are dividing the main program into multiple sections and implementing an elapsed time counter (see flow chart). Both of these techniques use the same program jump table concept that was described above. First, 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.

An elapsed time counter can be implemented using the RTCC counter. At the beginning of I/O polling the RTCC register is cleared. It then starts counting the instruction cycles. Then after the main program subsection has been executed, the RTCC register 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  RTCC,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 CYCLES
GOTO   START   ;0 ADDITIONAL INSTRUCTION CYCLES
```

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

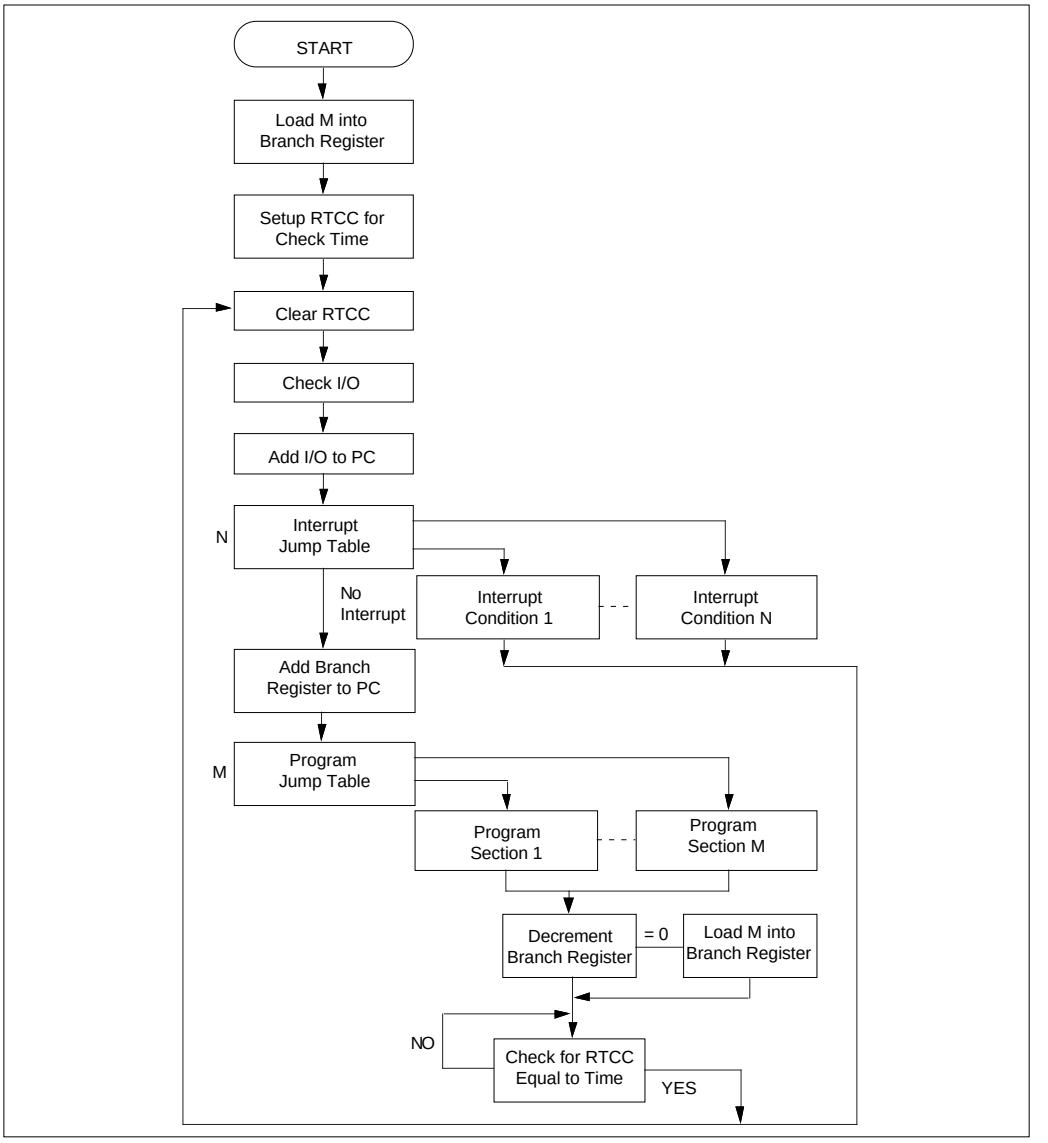
RTCC(40) - POLL(50-15(35)) =5

will be added to the program counter. The program will then jump to the sixth NOP. That NOP plus the 9 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 (see flow chart and code) is the core program for the software interrupt technique described above. This program assumes four interrupt conditions, four main program sections and an eight additional elapsed time instructions.

FIGURE 1 - SOFTWARE INTERRUPT FLOW CHART



APPENDIX A:

```
MPASM B0.54                                                    PAGE  1



                        LIST    P=16C54

                                        ;SOFTWARE INTERRUPT APPLICATIONS
                                        ;BRANCH IS MAIN PROGRAM REGISTER
0008                BRANCH EQU     8
0009                CNDTN  EQU     9
000A                IO     EQU     0A
000B                TEMP   EQU     0B

0000 0069           SETUP  CLRF    CNDTN
0001 0C04                  MOVLW   4
0002 0028                  MOVWF   BRANCH  ;FOUR MAIN PROGRAM SECTIONS
0003 0C08                  MOVLW   8
0004 0002                  OPTION          ;SET RTCC TO ONE COUNT PER INSTRUCTION CYCLE

0005 0061           START  CLRF    1       ;CLEAR RTCC REGISTER
0006 0206                  MOVF    6,W     ;READ I/O
0007 002A                  MOVWF   IO
0008 0109                  IORWF   CNDTN,W ;THIS SECTION OF CODE CALCULATES THE
0009 002B                  MOVWF   TEMP    ;JUMP TABLE. ANY INPUT THAT CHANGES FROM
000A 0209                  MOVF    CNDTN,W ;A ZERO TO A ONE IS CONSIDERED AN INTERRUPT.
000B 00AB                  SUBWF   TEMP,1  ;THE EQUATION IS:
000C 020A                  MOVF    IO,W    ;  (IO + CNDTN) - CNDTN = INTERRUPT
000D 0029                  MOVWF   CNDTN   ;WHERE IO IS CURRENT INPUT AND
000E 020B                  MOVF    TEMP,W  ;CNDTN IS PREVIOUS INPUT.
000F 0E03                  ANDLW   3       ;MASK OFF TOP 6 BITS
0010 01E2                  ADDWF   2,1     ;ADD INPUT TO PC TO CREATE JUMP TABLE
0011 0A1B                  GOTO    MAIN    ;FOR INPUT=00
0012 0A15                  GOTO    INT1    ;FOR INPUT=01
0013 0A17                  GOTO    INT2    ;FOR INPUT=10
0014 0A19                  GOTO    INT3    ;FOR INPUT=11

0015 0000           INT1   NOP             ;INTERRUPT LINE 1 CODE
0016 0A05                  GOTO    START
0017 0000           INT2   NOP             ;INTERRUPT LINE 2 CODE
0018 0A05                  GOTO    START
0019 0000           INT3   NOP             ;INTERRUPT LINES 1 AND 2 CODE
001A 0A05                  GOTO    START

001B 0208           MAIN   MOVF    BRANCH,W
001C 01E2                  ADDWF   2,1     ;ADD BRANCH TO PC TO CREATE JUMP TABLE
001D 0000                  NOP
001E 0A28                  GOTO    MAIN4   ;JUMP TABLE, LAST FIRST ON DECREMENT TABLE
001F 0A26                  GOTO    MAIN3
0020 0A24                  GOTO    MAIN2
0021 0A22                  GOTO    MAIN1

0022 0000           MAIN1  NOP             ;MAIN PROGRAM CODE BANK ONE
0023 0A2A                  GOTO    BRNCHK
0024 0000           MAIN2  NOP             ;MAIN PROGRAM CODE SECTION TWO
0025 0A2A                  GOTO    BRNCHK
0026 0000           MAIN3  NOP             ;MAIN PROGRAM CODE SECTION THREE
0027 0A2A                  GOTO    BRNCHK
0028 0000           MAIN4  NOP             ;MAIN PROGRAM CODE SECTION FOUR
0029 0A2A                  GOTO    BRNCHK

002A 02E8           BRNCHK DECFSZ  BRANCH,1 ;DECREMENT BRANCH REGISTER AND CHECK FOR ZERO
002B 0A2E                  GOTO    TIMCHK
002C 0C04                  MOVLW   4
002D 0028                  MOVWF   BRANCH  ;RELOAD BRANCH WITH 4 AT END OF MAIN

002E 0C29           TIMCHK MOVLW   D'41'   ;CHECK TO SEE IF RTCC HAS REACHED 50(50-7)
```

```
002F 0081                  SUBWF   1,W     ;DETERMINE WAIT TIME
0030 01E2                  ADDWF   2,1     ;ADD WAIT TIME TO PC
0031 0000                  NOP
0032 0000                  NOP
0033 0000                  NOP
0034 0000                  NOP
0035 0000                  NOP
0036 0000                  NOP
0037 0000                  NOP
0038 0A05                  GOTO    START
                           END


Errors   :    0
Warnings :    0
```

WORLDWIDE SALES & SERVICE

AMERICAS

Corporate Office
Microchip Technology Inc.
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 602 786-7200 Fax: 602 786-7277
Technical Support: 602 786-7627
Web: http://www.mchip.com/microhip

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: 708 285-0071 Fax: 708 285-0075

Dallas
Microchip Technology Inc.
14651 Dallas Parkway, Suite 816
Dallas, TX 75240-8809
Tel: 214 991-7177 Fax: 214 991-8588

Dayton
Microchip Technology Inc.
35 Rockridge Road
Englewood, OH 45322
Tel: 513 832-2543 Fax: 513 832-2841

Los Angeles
Microchip Technology Inc.
18201 Von Karman, Suite 455
Irvine, CA 92715
Tel: 714 263-1888 Fax: 714 263-1338

New York
Microchip Technology Inc.
150 Motor Parkway, Suite 416
Hauppauge, NY 11788
Tel: 516 273-5305 Fax: 516 273-5335

AMERICAS (continued)

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

ASIA/PACIFIC

Hong Kong
Microchip Technology
Unit No. 3002-3004, Tower 1
Metroplaza
223 Hing Fong Road
Kwai Fong, N.T. Hong Kong
Tel: 852 2 401 1200 Fax: 852 2 401 3431

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

Singapore
Microchip Technology
200 Middle Road
#10-03 Prime Centre
Singapore 188980
Tel: 65 334 8870 Fax: 65 334 8850

Taiwan
Microchip Technology
10F-1C 207
Tung Hua North Road
Taipei, Taiwan, ROC
Tel: 886 2 717 7175 Fax: 886 2 545 0139

EUROPE

United Kingdom
Arizona Microchip Technology Ltd.
Unit 6, The Courtyard
Meadow Bank, Furlong Road
Bourne End, Buckinghamshire SL8 5AJ
Tel: 44 0 1628 851077 Fax: 44 0 1628 850259

France
Arizona Microchip Technology SARL
2 Rue du Buisson aux Fraises
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 Muenchen, Germany
Tel: 49 89 627 144 0 Fax: 49 89 627 144 44

Italy
Arizona Microchip Technology SRL
Centro Direzionale Colleoni
Palazzo Pegaso Ingresso No. 2
Via Paracelso 23, 20041
Agrate Brianza (MI) Italy
Tel: 39 039 689 9939 Fax: 39 039 689 9883

JAPAN

Microchip Technology Intl. Inc.
Benex S-1 6F
3-18-20, Shin Yokohama
Kohoku-Ku, Yokohama
Kanagawa 222 Japan
Tel: 81 45 471 6166 Fax: 81 45 471 6122

9/22/95