AN566

Using the PORTB Interrupt on Change as an External Interrupt

Author: *Mark Palmer*
Microchip Technology Inc.

INTRODUCTION

The PICmicro™ families of RISC microcontrollers are designed to provide advanced performance and a cost-effective solution for a variety of applications. To address these applications, there is the PIC16CXXX microcontroller family of products. This family has numerous peripheral and special features to better address user applications.

The feature this application note will focus on is the Interrupt on Change of the PORTB pins. This “interrupt on change” is triggered when any of the RB7:RB4 pins, configured as an input, changes level. When this interrupt is used in conjunction with the software programmable weak internal pull-ups, a direct interface to a keypad is possible. This is shown in application note AN552, Implementing Wake-up on Key Stroke. Another way to use the “interrupt on change” feature would be as additional external interrupt sources. This allows PIC16CXXX devices to support multiple external interrupts, in addition to the built-in external interrupt on the INT pin.

This application note will discuss some of the issues in using PORTB as additional external interrupt pins, and will show some examples. These examples can be easily modified to suit your particular needs.

USING A PORTB INPUT FOR AN EXTERNAL INTERRUPT

The interrupt source(s) cannot simply be directly connected to the PORTB pins, and expect an interrupt to occur the same as on the interrupt (INT) pin. To develop the microcontrollers hardware/software to act as an interrupt by an external signal, we must know the characteristics of the external signal. After we know this, we can determine the best way to structure the program to handle this signal. The characteristics that we need to consider when developing the interrupt include:

1. The rising edge and falling edges.
2. The pulse width of the interrupt trigger (high time / low time).

It is easy to understand the need of knowing about which edge triggers the interrupt service routine for the external interrupt. This allows one to ensure that the interrupt service routine is only entered for the desired edge, with all other edges ignored. Not so clear is the pulse width of the interrupt’s trigger. This characteristic helps determine the amount of additional overhead that the software routine may need.

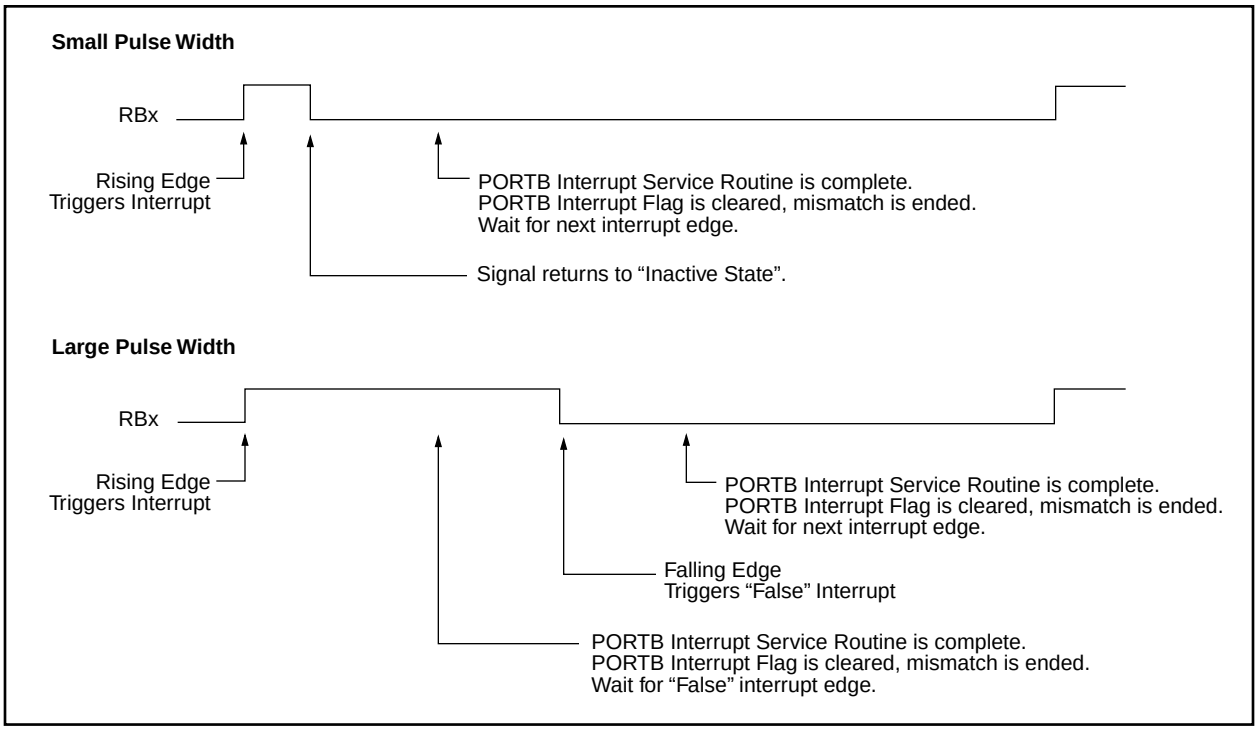
Figure 1 shows the two cases for the interrupt signal verses the time to complete the interrupt service routine. The first waveform is when the signal makes the low-to-high-to-low transitions before the interrupt service routine has completed (interrupt flag cleared). When the interrupt flag has been cleared, the interrupt signal has already returned to the inactive level. The next transition of the signal is due to another interrupt request. An interrupt signal with this characteristic will be called a small pulse width signal.

The second waveform is when the signal only makes the low-to-high transitions before the interrupt service routine has completed (interrupt flag cleared). The next transition (high-to-low) will return the interrupt signal to the inactive level. This will generate a “false” interrupt, that will need to be cleared. Then the following transition (low-to-high) will be a “true” interrupt. An interrupt signal with this characteristic will be called a wide pulse width signal.

An interrupt pulse with a small pulse width requires less overhead than a wide pulse width. A small pulse width signal must be less than the minimum execution time of the interrupt service routine, while a wide pulse width must be greater then the maximum time through the interrupt service routine.

Example 1 shows a single interrupt source on PORTB (RB7), which executes the interrupt service routine on a rising edge. The interrupt source has a small pulse width. In this case, since the interrupt pulse width is small, the pulse has gone high and then low again before PORTB is read to end the mismatch condition. So when PORTB is read it will read a low signal and will again be waiting for the rising edge transition.

FIGURE 1: INTERRUPT STEPS FOR SMALL AND WIDE PULSE WIDTHS



EXAMPLE 1: SINGLE INTERRUPT WITH A SMALL PULSE WIDTH

```
PER_INT       BTFSS   INTCON, RBIF          ; PortB interrupt?
              GOTO    OTHER_INT             ; Other interrupt
                      :                     ; Do task for INT on RB7
                      :                     ;
CLR_RBINTF    MOVF    PORTB, 1              ; Read PortB (to itself) to end
                                            ; mismatch condition
              BCF     INTCON, RBIF          ; Clear the RB interrupt flag.
              RETFIE                        ; Return from interrupt
OTHER_INT             :                     ; Do what you need to here
                      :
              RETFIE                        ; Return from interrupt
```

Example 2 shows a single interrupt source on PORTB (RB7), which executes the interrupt service routine on a rising edge. The interrupt source has a wide pulse width. In this case since the interrupt pulse width is large, the pulse is still high before PORTB is read to end the mismatch condition. So when PORTB is read it will read a high signal and will generate an interrupt on the next falling edge transition (which should be ignored).

EXAMPLE 2: SINGLE INTERRUPT WITH A WIDE PULSE WIDTH

```
PER_INT       BTFSS  INTCON, RBIF         ; PortB interrupt?
              GOTO   OTHER_INT            ; Other interrupt
              BTFSC  PORTB, RB7           ; Check for rising edge
              GOTO   CLR_RBINTF           ; Falling edge, clear PortB int
              :                           ;   flag
                     :                    ; Do task for INT on RB7
                     :
CLR_RBINTF    MOVF   PORTB, 1             ; Read PortB (to itself) to end
                                          ;    mismatch condition
              BCF    INTCON, RBIF         ; Clear the RB interrupt flag.
              RETFIE                      ; Return from interrupt
OTHER_INT             :                   ; Do what you need to here
                      :
              RETFIE                      ; Return from interrupt
```

Example 3 shows an interrupt on change with the interrupt source on PORTB (RB7). This executes the interrupt service routine on a both edges. The interrupt source must have a minimum pulse width to ensure that both edges can be “seen”. The minimum pulse width is the maximum time from the interrupt edge to the reading of PORTB and clearing the interrupt flag.

EXAMPLE 3: INTERRUPT ON CHANGE

```
PER_INT       BTFSS  INTCON, RBIF         ; PortB interrupt?
              GOTO   OTHER_INT            ; Other interrupt
CLR_RBINTF    MOVF   PORTB, 1             ; Read PortB (to itself) to end
                                          ;    mismatch condition
              BCF    INTCON, RBIF         ; Clear the RB interrupt flag.
                     :                    ; Do task for INT on RB7
                     :                    ;
              RETFIE                      ; Return from interrupt
OTHER_INT                                 ; Do what you need to here
                     :
              RETFIE                      ; Return from interrupt
```

USING PORTB INPUTS FOR MULTIPLE INTERRUPTS

The previous examples have been for a single external interrupt on PORTB. This can be extended to support up to four external interrupts. To do this requires additional software overhead, to determine which of the PORTB pins (RB7:RB4) caused the interrupt. Care should be taken in the software to ensure that no interrupts are lost.

In this example, the interrupt sources on RB7, RB5, and RB4 have a small pulse width, while the interrupt source on pin RB6 is wide and should cause a trigger on the rising edge.

SUMMARY

The PORTB interrupt on change feature is both a very convenient method for direct interfacing to an external keypad, with no additional components, but is also versatile in its uses the ability to add up to four additional external interrupts. Of course hybrid solutions are also possible. That is, for example, using PORTB<6:1> as a 3x3 keypad, with PORTB<7> as an external interrupt and PORTB<0> as a general purpose I/O. The flexibility of this feature allows the user to implement a best fit design for the application.

EXAMPLE 4: MULTIPLE INTERRUPTS WITH DIFFERENT PULSE WIDTHS

```
PER_INT         BTFSS   INTCON, RBIF          ; PortB interrupt?
                GOTO    OTHER_INT             ; Other interrupt
;
; PortB change interrupt has occurred. Must determine which pin caused
; interrupt and do appropriate action. That is service the interrupt,
; or clear flags due to other edge.
;
                MOVF    PORTB, 0              ; Move PortB value to the W register
                                              ;   This ends mismatch conditions
                MOVWF   TEMP                  ; Need to save the PortB reading.
                XORWF   LASTPB, 1             ; XOR last PortB value with the new
                                              ;   PortB value.
CK_RB7          BTFSC   LASTPB, RB7           ; Did pin RB7 change
                CALL    RB7_CHG               ; RB7 changed and caused the interrupt
CK_RB6          BTFSC   LASTPB, RB6           ; Did pin RB6 change
                CALL    RB6_CHG               ; RB6 changed and caused the interrupt
CK_RB5          BTFSC   LASTPB, RB5           ; Did pin RB5 change
                CALL    RB5_CHG               ; RB5 changed and caused the interrupt
CK_RB4          BTFSC   LASTPB, RB4           ; Did pin RB4 change
                GOTO    RB4_CHG               ; RB4 changed and caused the interrupt
;
RB7_CHG              :                        ; Do task for INT on RB7
                     :                        ;
                RETURN
RB6_CHG         BTFSC   PORTB, RB6            ; Check for rising edge
                RETURN                        ; Falling edge, Ignore
                     :                        ; Do task for INT on RB6
                     :
                RETURN
RB5_CHG              :                        ; Do task for INT on RB5
                     :                        ;
                RETURN
RB4_CHG              :                        ; Do task for INT on RB4
                     :                        ;
CLR_RBINTF      MOVF    TEMP, 0               ; Move the PortB read value to the
                MOVWF   LASTPB                ;   register LASTPB
                BCF     INTCON, RBIF          ; Clear the RB interrupt flag.
                RETFIE                        ; Return from interrupt
;
OTHER_INT            :                        ; Do what you need to here
                     :
                RETFIE                        ; Return from interrupt
```

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.