Electromechanical Switch Replacement

A Triple Input, Inverting Debounce Circuit

Author: *Jim Nagy*
London, Ontario, Canada
email: nagy@wwdc.com

APPLICATION OPERATION

This program configures the PIC12C508 as an interface between mechanical contacts and electronic circuits. This program creates a 'triple input, inverting debounce circuit', that requires no support circuitry other than the input contacts and a 2.5 to 5.5V supply.

Often several R-C networks and schmitt trigger gates are combined discretely to provide this function. But by using the internal RC oscillator, internal pullup resistors and software time delay circuits, I was able to create the interface entirely within the PIC12C508. One other advantage to using this chip is the ability to use the sleep mode to reduce power consumption when inputs are not changing.

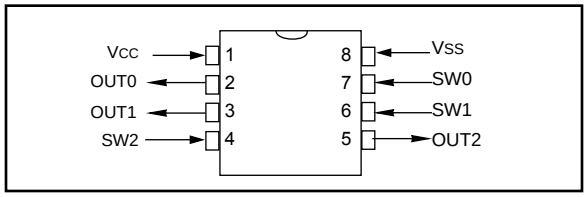
After the mechanical contacts have been processed by my circuit, the logic levels are clean and free of the bounce or jitter normally produced by a mechanical interface. High speed electronic circuits can then process the signals, without fear that the input is just random noise, or that bounces may cause multiple triggers.

The program was developed and assembled on a Macintosh computer, so I have followed IEEE syntax rather than MPASM.

GRAPHICAL HARDWARE REPRESENTATION

The PIC12C508 is connected as follows. Input contacts are connected between circuit common (Vss) and the inputs. Although the sample program only inverts the inputs, any combination could actually be used to control the outputs, which control electrical circuits/ loads.

BLOCK DIAGRAM



DEBOUNCE.LST

```
  1             ;
  2             ;                           Debounce
  3             ;                           ========
  4             ;
  5             ;                           by Jim Nagy, June 1997
  6             ;
  7             ; A program to use the 12C508 as an interface circuit between
  8             ; mechanical contacts and electronic circuits. Electronic time delays
  9             ; are used to ensure that the positions of up to three switches/contacts
 10             ; are stable ('debounced') before passing their status on. No
 11             ;       external components (other than the switches) are required.
 12             ;
 13             ; Circuit inputs are  GP0(pin7), GP1(pin6), and GP3(pin4),
 14             ; while respective outputs are GP5(pin2), GP4(pin3), and GP2(pin5).
 15             ; The internal pullups are used for the three inputs, simplifying
 16             ; the connection of switches between circuit common and the inputs.
 17             ;
 18             ; After approximately 0.5sec of inactivity, the '508 puts itself into
 19             ; the sleep mode, consuming little power if no switch is being pressed.
 20             ; The circuit will wake up (almost instantly) for any change in input.
 21             ;
 22             ; This circuit inverts the logic level of each input, so a closed switch
 23             ; between pin 7 and ground provides a logic high at pin 2, etc. There is
 24             ; currently no interaction between the three inputs and the outputs,
 25             ; but the logic could easily be altered to provide other functions.
 26             ;
 27             ; Currently, a 'debounce' time of about 24mS is used, but this can be
 28             ; varied (in 2mS increments) by changing the value of DBTime below.
 29             ;
 30             ; ************************************************************************
 31
 32             ;Equates
 33     = 0000      W       EQU     0
 34     = 0001      F       EQU     1
 35     = 0002      Z       EQU     2
 36     = 0002      Zero    EQU     2
 37     = 0000      C       EQU     0
 38     = 0000      Carry   EQU     0
 39     = 0007      GPWUF   EQU     7
 40
 41     = 0000      Sw0     EQU     0       ; switch names
 42     = 0001      Sw1     EQU     1
 43     = 0003      Sw2     EQU     3
 44
 45     = 0007      TFlag   EQU     7       ; bit numbers for flags
 46
 47     = 000C      DBTime  EQU     12      ; debounce for 12 * 2.048mS
 48
 49
 50                 ; '508 Register Assignments
 51     = 0000      INDF    EQU     H'00'
 52     = 0001      TMR0    EQU     H'01'
 53     = 0002      PCL             EQU     H'02'
 54     = 0003      STATUS  EQU     H'03'
 55     = 0004      FSR             EQU     H'04'
 56     = 0005      OSCCAL  EQU     H'05'
 57     = 0006      GPIO    EQU     H'06'
 58
 59                 ; program variables
 60     = 0007      SleepT  EQU     H'07' ; inactivity counter to allow sleep
 61     = 0008      Flags   EQU     H'08' ; storage space for messages
 62     = 0009      ThisSw  EQU     H'09' ; current switch status
 63
 64                 ; switch0
```

```
 65      = 000A      LastSw0  EQU H'0A'    ; last value - for detecting change
 66      = 000B      SwState0 EQU H'0B'    ; the official (debounced) status
 67      = 000C      DBTimer0 EQU H'0C'    ; counter used during debouncing of the switch
 68
 69                  ; switch1
 70      = 000D      LastSw1  EQU H'0D'    ;       "      "
 71      = 000E      SwState1 EQU H'0E'    ;       "      "
 72      = 000F      DBTimer1 EQU H'0F'    ;       "      "
 73
 74                  ; switch2
 75      = 0010      LastSw2  EQU H'10'    ;       "      "
 76      = 0011      SwState2 EQU H'11'    ;       "      "
 77      = 0012      DBTimer2 EQU H'12'    ;       "      "
 78
 79
 80                  ;       ******************************************
 81                  ;       Setting the ID words...
 82
 83                                  ORG H'0200'
 84 0200   0000      ID0             Data.W  H'0000'
 85 0201   0000      ID1             Data.W  H'0000'
 86 0202   0000      ID2             Data.W  H'0000'
 87 0203   0003      ID3             Data.W  H'0003'
 88
 89                  ;       ******************************************
 90                  ;       and the Fuses...
 91                  ;       MCLRDisabled, NoCodeProtect, NoWatchDog, IntRCOsc
 92
 93                                      ORG H'0FFF'
 94 0FFF   000A      CONFIG  Data.W  H'000A'
 95
 96
 97                  ;       ******************************************
 98
 99
100                  ORG H'00'
101
102 0000   0025      MOVWF   OSCCAL  ; store the factory osc. calibration value
103
104 0001   0004      Start   CLRWDT              ; setting up options
105 0002   0C02      MOVLW   B'00000010'         ; use int clock input, /8 prescaler
106 0003   0002      OPTION                      ; pullups on, and wakeup on pin change
107 0004   07E3      BTFSS   STATUS,GPWUF        ; don't change outputs on a wakeup
108 0005   0066      CLRF    GPIO                ; but output 0s on a powerup
109 0006   0C0B      MOVLW   B'00001011'         ; GP0, GP1 and GP3 are inputs,
110 0007   0006      TRIS    GPIO                ; GP2, GP4 and GP5 are outputs
111
112 0008   0067     CLRF    SleepT              ; always start with a clean slate
113 0009   0068     CLRF    Flags
114 000A   006C     CLRF    DBTimer0
115 000B   006F     CLRF    DBTimer1
116 000C   0072     CLRF    DBTimer2
117 000D   06E3     BTFSC   STATUS,GPWUF        ; on wakeup, don't change the switch states
118 000E   0A15     GOTO    Main
119
120 000F   006A     CLRF    LastSw0             ; but if it's a powerup,
121 0010   006B     CLRF    SwState0            ; assume they're all off (open)
122 0011   006D     CLRF    LastSw1
123 0012   006E     CLRF    SwState1
124 0013   0070     CLRF    LastSw2
125 0014   0071     CLRF    SwState2
126
127
128                ; The Main loop has to be executed every 1mS in order to provide
129                ; proper timekeeping, and to monitor the switches often enough.
130                ; In this instance, this is not a problem, but conceivably the loop
```

```
131                ; could be expanded to do other things...
132
133 0015   0925        Main    CALL    DoTime      ; need clock 'running'for debounce
134 0016   0934        CALL    Switch0             ; check switch0 position,
135 0017   094D        CALL    Switch1             ; then switch1,
136 0018   0966        CALL    Switch2             ; and switch2
137
138 0019   0040        CLRW                        ; assume no switch is pressed
139
140 001A   022B        MOVF    SwState0,F          ; check the state of sw0
141 001B   0743        BTFSS   STATUS,Zero         ; do nothing if it's not pressed
142 001C   0D20        IORLW   B'00100000'         ; else set GP5 high
143
144 001D   022E        MOVF    SwState1,F          ; same for sw1
145 001E   0743        BTFSS   STATUS,Zero
146 001F   0D10        IORLW   B'00010000'         ; output on GP4
147
148 0020   0231        MOVF    SwState2,F          ; and sw2
149 0021   0743        BTFSS   STATUS,Zero
150 0022   0D04        IORLW   B'00000100'         ; with output on GP2
151
152 0023   0026        MOVWF   GPIO
153
154 0024   0A15        GOTO    Main
155
156
157
158            ;       ******************************************
159            ;               DoTime
160            ;       Continually checks TMR0 and updates the program timers
161            ;       Returns with W = 0 if there was no time change, and W = 1 if there was.
162
163 0025   07E1        DoTime  BTFSS   TMR0,7           ; high bit of timer set?
164 0026   0A29        GOTO    dt1                     ; no - we've overflowed
165 0027   05E8        BSF             Flags,TFlag     ; yes, set the flag
166 0028   0800        RETLW   0
167
168 0029   07E8        dt1     BTFSS   Flags,TFlag     ; have the timers been serviced?
169 002A   0800        RETLW   0                       ; yes
170
171 002B   02AC        INCF    DBTimer0,F              ; no, increment all the timers
172 002C   02AF        INCF    DBTimer1,F
173 002D   02B2        INCF    DBTimer2,F
174 002E   02A7        INCF    SleepT,F
175 002F   04E8        BCF             Flags,TFlag     ; reset the service flag, and
176 0030   0743        BTFSS   STATUS,Zero             ; see if the sleep timer has overflowed
177 0031   0801        RETLW   1
178
179                           ; There's been no activity for 256*2mS, it's time to sleep
180 0032   0206        MOVF    GPIO,W                  ; read the current pin status
181 0033   0003        SLEEP                           ; and good night
182
183
184            ;       ******************************************
185            ;               Switch0
186            ; Gets the current (debounced) status of the switch connected to GP0 (pin 7)
187            ; Will return with W = 0 if the switch is not pressed and W = 1 if it is.
188
189 0034   0069        Switch0 CLRF    ThisSw          ; assume switch is not pressed
190 0035   0706        BTFSS   GPIO,Sw0                ; check sw status
191 0036   02A9        INCF    ThisSw,F                ; the sw was pressed
192
193            ;       Compare to last state
194 0037   020A        MOVF    LastSw0,W
195 0038   0189        XORWF   ThisSw,W
196 0039   0743        BTFSS   STATUS,Zero             ; Zero is set if there's been no change
```

```
197 003A   0A46        GOTO    sw01
198
199           ;       No change since last scan, but are we in a debounce phase?
200 003B   020B         MOVF    SwState0,W
201 003C   0189         XORWF   ThisSw,W    ; compare present state of sw to the official state
202 003D   0643         BTFSC   STATUS,Zero ; Zero is Clr if they differ (we're in debounce)
203 003E   0A4A         GOTO    sw02
204
205        ;       Switch is changing, have we gone past the debounce time?
206 003F   0C0C          MOVLW   DBTime                  ; get the debounce time
207 0040   008C          SUBWF   DBTimer0,W              ; and compare to elapsed
208 0041   0703          BTFSS   STATUS,Carry            ; Carry is set if DBTimer >= DBTicks
209 0042   0A4A          GOTO    sw02
210
211        ;       we've exceeded the debounce time - change the official state of the switch
212 0043   0209          MOVF    ThisSw,W
213 0044   002B          MOVWF   SwState0                ; store current state in SwState
214 0045   0A4A          GOTO    sw02
215
216        ;       the switch input is changing state - prepare to debounce
217 0046   0209          sw01    MOVF    ThisSw,W
218 0047   002A          MOVWF   LastSw0       ; remember this passes' state
219 0048   0067          CLRF    SleepT        ; we've had activity - reset the sleep timer
220 0049   006C          CLRF    DBTimer0      ; and reset the debounce timer
221
222         ;       we're done for this pass...
223 004A   070B         sw02    BTFSS   SwState0,0      ; check the switch state, and
224 004B   0800         RETLW   0                       ; return with 0 if not pressed,
225 004C   0801         RETLW   1                       ; 1 if pressed
226
227
228         ;       ****************************************
229         ;               Switch1
230         ;       Gets the current (debounced) status of the switch connected to GP1 (pin 6)
231         ;       Will return with W = 0 if the switch is not pressed and W = 1 if it is.
232
233 004D   0069          Switch1 CLRF    ThisSw        ; assume switch is not pressed
234 004E   0726          BTFSS   GPIO,Sw1              ; check sw status
235 004F   02A9          INCF    ThisSw,F              ; the sw was pressed
236
237         ;       Compare to last state
238 0050   020D          MOVF    LastSw1,W
239 0051   0189          XORWF   ThisSw,W
240 0052   0743          BTFSS   STATUS,Zero           ; Zero is set if there's been no change
241 0053   0A5F          GOTO    sw11
242
243        ;       No change since last scan, but are we in a debounce phase?
244 0054   020E          MOVF    SwState1,W
245 0055   0189          XORWF   ThisSw,W    ; compare present state of sw to the official state
246 0056   0643          BTFSC   STATUS,Zero ; Zero is Clr if they differ (we're in debounce)
247 0057   0A63          GOTO    sw12
248
249        ;       Switch is changing, have we gone past the debounce time?
250 0058   0C0C          MOVLW   DBTime                  ; get the debounce time
251 0059   008F          SUBWF   DBTimer1,W              ; and compare to elapsed
252 005A   0703          BTFSS   STATUS,Carry            ; Carry is set if DBTimer >= DBTicks
253 005B   0A63           GOTO    sw12
254
255        ;       we've exceeded the debounce time - change the official state of the switch
256 005C   0209          MOVF    ThisSw,W
257 005D   002E          MOVWF   SwState1                ; store current state in SwState
258 005E   0A63          GOTO    sw12
259
260        ;       the switch input is changing state - prepare to debounce
261 005F   0209          sw11    MOVF    ThisSw,W
262 0060   002D          MOVWF   LastSw1                 ; remember this passes' state
```

```
263 0061   0067          CLRF    SleepT      ; we've had activity - reset the sleep timer
264 0062   006F          CLRF    DBTimer1    ; and reset the debounce timer
265
266        ;       we're done for this pass...
267 0063   070E          sw12    BTFSS   SwState1,0   ; check the switch state, and
268 0064   0800          RETLW   0                   ; return with 0 if not pressed,
269 0065   0801          RETLW   1                      ; 1 if pressed
270
271
272       ;       ****************************************
273       ;              Switch2
274       ;       Gets the current (debounced) status of the switch connected to GP3 (pin 4)
275       ;       Will return with W = 0 if the switch is not pressed and W = 1 if it is.
276
277 0066   0069          Switch2 CLRF    ThisSw     ; assume switch is not pressed
278 0067   0766          BTFSS   GPIO,Sw2           ; check sw status
279 0068   02A9          INCF    ThisSw,F           ; the sw was pressed
280
281        ;       Compare to last state
282 0069   0210          MOVF    LastSw2,W
283 006A   0189          XORWF   ThisSw,W
284 006B   0743          BTFSS   STATUS,Zero        ; Zero is set if there's been no change
285 006C   0A78          GOTO    sw21
286
287        ;       No change since last scan, but are we in a debounce phase?
288 006D   0211          MOVF    SwState2,W
289 006E   0189          XORWF   ThisSw,W  ; compare present state of sw to the official state
290 006F   0643          BTFSC   STATUS,Zero  ; Zero is Clr if they differ (we're in debounce)
291 0070   0A7C          GOTO    sw22
292
293        ;       Switch is changing, have we gone past the debounce time?
294 0071   0C0C           MOVLW   DBTime                  ; get the debounce time
295 0072   0092           SUBWF   DBTimer2,W              ; and compare to elapsed
296 0073   0703           BTFSS   STATUS,Carry            ; Carry is set if DBTimer >= DBTicks
297 0074   0A7C           GOTO    sw22
298
299        ;       we've exceeded the debounce time - change the official state of the switch
300 0075   0209           MOVF    ThisSw,W
301 0076   0031           MOVWF   SwState2                ; store current state in SwState
302 0077   0A7C           GOTO    sw22
303
304        ;       the switch input is changing state - prepare to debounce
305 0078   0209           sw21    MOVF    ThisSw,W
306 0079   0030           MOVWF   LastSw2  ; remember this passes' state
307 007A   0067           CLRF    SleepT   ; we've had activity - reset the sleep timer
308 007B   0072           CLRF    DBTimer2 ; and reset the debounce timer
309
310        ;       we're done for this pass...
311 007C   0711           sw22    BTFSS   SwState2,0     ; check the switch state, and
312 007D   0800           RETLW   0                      ; return with 0 if not pressed,
313 007E   0801           RETLW   1                      ; 1 if pressed
314
315                       END
```

NOTES:

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.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.
14651 Dallas Parkway, Suite 816
Dallas, TX 75240-8809
Tel: 972-991-7177 Fax: 972-991-8588

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

Los Angeles
Microchip Technology Inc.
18201 Von Karman, Suite 1090
Irvine, CA 92612
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

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

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
RM 3801B, Tower Two
Metroplaza
223 Hing Fong Road
Kwai Fong, N.T., Hong Kong
Tel: 852-2-401-1200 Fax: 852-2-401-3431

India
Microchip Technology Inc.
India Liaison Office
No. 6, Legacy, Convent Road
Bangalore 560 025, India
Tel: 91-80-229-4036 Fax: 91-80-559-9840

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

Singapore
Microchip Technology Taiwan
Singapore Branch
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 2-717-7175 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: 44-1628-851077 Fax: 44-1628-850259

France
Arizona Microchip Technology SARL
Zone Industrielle de la Bonde
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 Müchen, 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: 39-39-6899939 Fax: 39-39-6899883

JAPAN

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

8/29/97