MPLABS\April 98

```
;==========mditrg.asm======Midi Trigger Prog========3/18/98====================
;
; Program for converting 16 trigger inputs (active low) to Midi Note-On
; messages.  The 16 triggers are polled one at a time.  If a trigger input is
; high and its current state register shows a low, send Midi Note-On.  If the
; trigger input is low and its current state register shows a high, send Midi
; Note-On with velocity zero to turn the note off.
;
; RA0-RA3 outputs, 16 trigger select addresses
; RA4, input, trigger 32/10
; RB0-RB3 inputs, Midi Channel number
; RB7, output, Midi serial output
;
; Midi Note-On Bytes - 0x9(channel), 0(key#), 0(velocity)
;
;================================================================================
        list    p=16c84
        radix   hex
;--------------------------------------------------------------------------------
; CPU EQUATES
;--------------------------------------------------------------------------------
portb   equ     0x06
porta   equ     0x05
w       equ     0
f       equ     1
fselect equ     0x04
fpoint  equ     0x00
status  equ     0x03
count   equ     0x0c        ;trigger number
nstat   equ     0x0d        ;midi note-on status byte with channel#
bitcnt  equ     0x0e        ;counter used in midi send subroutine
send    equ     0x0f        ;folds midi byte in midi send subroutine

;--------------------------------------------------------------------------------
; PROGRAM EQUATES
;--------------------------------------------------------------------------------
noteon  equ     0x90        ;midi status for note-on
vel     equ     0x64        ;velocity value used in midi note-on
baseky  equ     0x20        ;lowest of 16 key numbers used in midi note-on
                            ;and associated with each of 16 trigger inputs
basef   equ     0x10        ;lowest of 16 registers holding the current
                            ;state (note-on/note-off) of 16 triggers
;--------------------------------------------------------------------------------
; PROGRAM INITIALIZATION
;--------------------------------------------------------------------------------
        org     0x00
init    movlw   0x00
        option              ;option register
        movlw   0x00
        movwf   0x0b        ;interrupt register, disable all
        movlw   0x10
        tris    porta       ;porta, bits0-3 outputs, bit4 input
        movlw   0x0f
        tris    portb       ;portb, bits0-3 inputs, bits4-7 outputs
        bsf     portb,7     ;initialize midi out line high
```

```
56  ;
57          movlw   0x0f
58          movwf   count        ;start the counter at 16
59  ;-------------------------------------------------------------------------------
60  ;       loop to clear trigger current state registers 0x10 to ox1f
61  ;-------------------------------------------------------------------------------
62  loop    movf    count,w      ;move current count to w
63          addlw   basef        ;calculate number of next register to clear
64          movwf   fselect      ;mov register number to file select reg
65          clrf    fpoint       ;clear the register pointed to
66          decf    count,f      ;decrement count register
67          movlw   0xff
68          subwf   count,w      ;count minus ff, to test for end of count=ff
69          btfss   status,2     ;test the Z bit in status register
70          goto    loop         ;continue loop if count not finished
71
72  ;
73  ;-------------------------------------------------------------------------------
74  ; MAIN PROGRAM LOOP
75  ;-------------------------------------------------------------------------------
76  ;       decrement the counter, modulo 16
77  ;-------------------------------------------------------------------------------
78  mloop   decf    count,f      ;decrement counter (if 00, then it goes to ff)
79          movfw   count
80          btfsc   count,7      ;test for counter high bit (ff)
81          andlw   0x0f         ;if high bit high, then clear high nibble (0f)
82          movwf   count        ;new count value
83  ;-------------------------------------------------------------------------------
84  ;       load trigger address outputs for polling next trigger input
85  ;-------------------------------------------------------------------------------
86  ;
87          movwf   porta        ;low nibble addresses one of 16 trigger inputs
88  ;
89  ;-------------------------------------------------------------------------------
90  ;       let trigger address settle while setting up Midi Note-On status byte
91  ;-------------------------------------------------------------------------------
92  ;
93          clrwdt               ;reset the watchdog timer
94          movfw   portb        ;get midi channel number from portb low nibble
95          andlw   0x0f         ;put midi channel numberin low nibble
96          iorlw   noteon       ;put midi note-on status in high nibble
97          movwf   nstat        ;store midi note-on status byte in register nstat
98  ;
99  ;-------------------------------------------------------------------------------
100 ;       Read trigger input.  If high, branch to trgon.  If low, branch to trgoff.
101 ;-------------------------------------------------------------------------------
102 ;
103         btfss   porta,4      ;read trigger input from porta bit4
104         goto    trgoff       ;if high continue to trgon, else goto trgoff
105 ;
106 ;-------------------------------------------------------------------------------
107 ; TRIGGER ON ROUTINE
108 ;-------------------------------------------------------------------------------
109 ;
110 trgon   movfw   count
111         addlw   basef
112         movwf   fselect      ;address trigger's current state register
```

```
113         btfss   fpoint,0     ;read current state of trigger in LSB
114         goto    mloop        ;if note is already on, continue to next trigger
115         movlw   0xff         ;else send Midi Note-On Message
116         movwf   fpoint
117         movfw   nstat
118         call    midi         ;send Midi Note-On status byte
119         movfw   count
120         addlw   baseky
121         call    midi         ;send Midi Note-On key data
122         movlw   vel
123         call    midi         ;send Midi Note-On velocity data
124 ;
125         goto mloop           ;next trigger
126 ;
127 ;-------------------------------------------------------------------------------
128  TRIGGER OFF ROUTINE
129 ;-------------------------------------------------------------------------------
130 ;
131 trgoff  movfw   count
132         addlw   basef
133         movwf   fselect      ;address trigger's current state register
134         btfss   fpoint,0     ;read current state of trigger in LSB
135         goto    mloop        ;if note is already off, continue to next trigger
136         clrw                 ;else send Midi Note-On Message
137         movwf   fpoint
138         movfw   nstat
139         call    midi         ;send Midi Note-On status byte
140         movfw   count
141         addlw   baseky
142         call    midi         ;send Midi Note-On key data
143         clrw
144         call    midi         ;send Note-On velecity=0 for a Note-Off
145 ;
146         goto    mloop        ;next trigger
147 ;
148 ;-------------------------------------------------------------------------------
149 ; MIDI SEND SUBROUTINE
150 ;
151 ; With a 2MHz clock, the internal clock is 500KHz, requiring 16 cycles between
152 ; sends to get the 31.25 KHz Midi Baud Rate.
153 ;-------------------------------------------------------------------------------
154 ;
155 midi    movwf   send         ;load byte into send register
156         bcf     portb,7      ;send low midi start bit out portb bit7
157         movlw   0x09
158         movwf   bitcnt       ;set up the bit counter to shift out 8 bits
159 contin  nop
160         nop
161         nop
162         nop
163         nop
164         decf    bitcnt,f     ;decrement counter
165         btfsc   status,2     ;test Z for bitcnt=0
166         goto    endbit       ;if zero go to send endbit
167         rrf     send,f       ;shift next bit into status carry
168         btfss   status,0     ;check carry
169         goto    send0
```

```
170         goto    send1
171 send0   nop
172         bcf     portb,7      ;if carry is low, send midi low
173         goto    contin
174 send1   bsf     portb,7      ;else send midi high
175         goto    contin
176 endbit  nop
177         nop
178         nop
179         nop
180         bsf     portb,7      ;send high midi stop bit
181         movlw   0x04         ;delay before sending next midi byte
182         movwf   bitcnt
183 delay   decfsz  bitcnt,f
184         goto    delay
185 ;
186         return               ;return from MIDI Send subroutine
187 ;
188 ;-------------------------------------------------------------------------------
189 ; THE END
190 ;===============================================================================
```