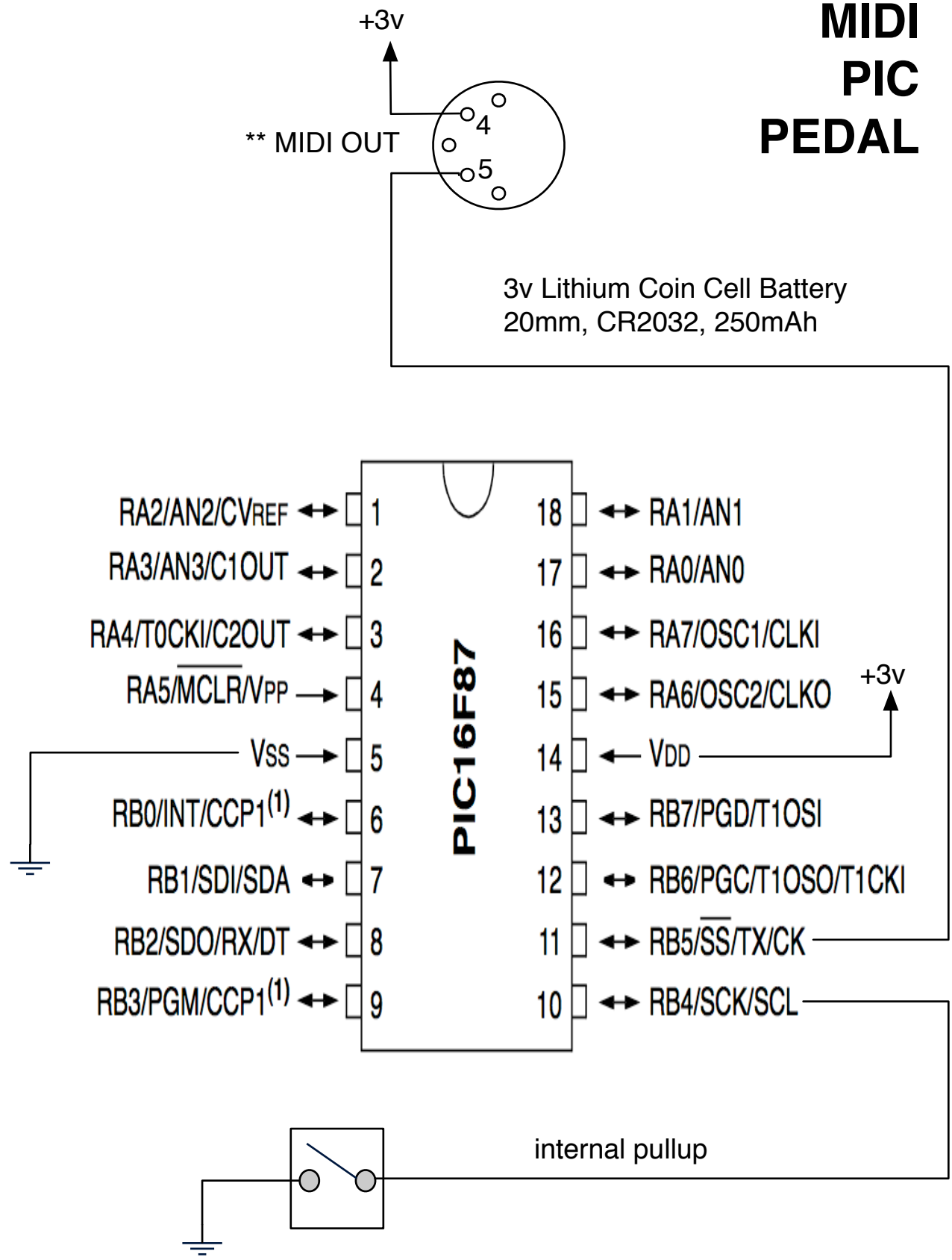
MIDI PIC PEDAL



** No 220Ω resistors on the MIDI lines because of 3v operation and 70Ω Thevenin equivalent resistance on low pic output lines.

```
;_____________________PIC16F87  MIDI FOOT PEDAL PROJECT_____________________
;MIDI Note-On (Off is Note-On with zero velocity) transmitted from one pedal switch.
;
;RB5 = TX used as MIDI Out. From the internal AUSART serial transmitter.
;
;RB4 = Switch Input, with the internal weak-pull-up enabled.  "PORTB Change Interrupts" enablec
;so that any activity on RB4 will wake up the processor and start the interrupt routine "main".  The
;processor sleeps with very low power consumption inbetween pedal activity.
;
;Use INTI01, internal 8MHz clock, check RA6 for 8MHz/4, can be tweeked with OSCTUNE Reg.
;Load SPBRG with 15 (decimal) for a baud rate of 31.25KHz (Midi Rate) = 8MHz/(16(x+1))
;
; ------------------------------------------------------------------------------------------------------
        list    p=16F87
;
;               CONFIGURATION BITS set in MPLab:
;
;               OSC = INTIO1, INTRC-OSC2 as Clock Out
;               Watchdog Timer = Off
;               Power Up Timer = Off
;               Brown Out Detect = Off
;               Low Voltage Program = Off
;               Data EE Read Protect = Off
;               Flash Program Write Protect = Off
;               Code Protect = Off
;               Fail-Safe Clock Mon. Enb = Disabled
;               Internal Ext. Switch Over Mode = Disabled

MOVLF   macro   literal, dest
        movlw   literal
        movwf   dest
        endm
; ------------------------------------------------------------------------------------------------------
; VARIABLES  --  STORED IN REGISTERS ABOVE THE DEDICATED REGISTERS
;------------------------------------------------------------------------------------------------------
        cblock          0x20

                vel                     ;holds 0 for note off or MIDI_VEL for note on
                x                       ;delay register
                y                       ;delay register

        endc
; ------------------------------------------------------------------------------------------------------
;               USEFUL EQUATES (Some not used here, mentioned for future programs)
;------------------------------------------------------------------------------------------------------

PORTA   equ     H'0005'
```

```
PORTB       equ         H'0006'
TRISA       equ         H'0085'     ;Data Direction for PORTA
TRISB       equ         H'0086'     ;Data Direction for PORTB
STATUS      equ         H'0003'
PCL         equ         H'0002'     ;Low Program Counter
OPTION_REG  equ         H'0081'     ;Option Control Reg.

; ----------------------AUSART Registers----------------------------------------------------------
TXSTA       equ         H'0098'     ;Transmit Status and Control Reg.
RCSTA       equ         H'0018'     ;Receive Status and Control Reg.
SPBRG       equ         H'0099'     ;Baud Rate Generator for AUSART
TXREG       equ         H'0019'     ;Transmit Data Reg.
RCREG       equ         H'001A'     ;Receive Data Reg.

; ----------------------Internal OSC Registers----------------------------------------------------
OSCCON      equ         H'008F'     ;OSC Control Reg.
OSCTUNE     equ         H'0090'     ;OSC Tuning of Internal clock

; ----------------------Interrupt Control Registers-----------------------------------------------
INTCON      equ         H'000B'     ;Interrupt Control Reg.
PIE1        equ         H'008C'     ;Peripheral Interrupt Enable Reg1
PIE2        equ         H'008D'     ;Peripheral Interrupt Enable Reg2
PIR1        equ         H'000C'     ;Peripheral Interrupt Request Flags

; ----------------------Individual Bits of Registers Above----------------------------------------
RP0         equ         H'05'       ;Bank Select bit in STATUS
C           equ         H'00'       ;Carry bit in STATUS
Z           equ         H'02'       ;Zero bit in STATUS
TXIF        equ         H'04'       ;PIR1 Reg.  Reads 1 when TXREG is empty i.e.
                                    ;TSR loaded. Cleared by loading TXREG.
TRMT        equ         H'01'       ;TXSTA Reg. 1=TSR empty, 0=TSR full
                                    ;TSR is the Transmit Shift Register.
RCIF        equ         H'05'       ;PIR1 Reg.  Reads 1 when RCREG is full ?
                                    ;byte received.  Cleared by reading RCREG.
ADIF        equ         H'06'       ;PIR1 Reg. Reads 1 when A/D convert is done.
                                            ;Cleared in software.
TMR1IF      equ         H'00'       ;PIR1 Reg. Reads 1 when TMR1 overflows
                                            ;Cleared in software.
GIE         equ         H'07'       ;INTCON reg. Global Interrupt Enable
RBIE        equ         H'03'       ;INTCON reg. RB port change interrupt enable
RBIF        equ         H'00'       ;INTCON reg. RB port change interrupt flag


; ----------------------Other values -------------------------------------------------------
F           equ         H'01'       ;Instruction results go to the specified register
W           equ         H'00'       ;Instruction results go to W
NOTE_ON     equ         H'90'       ;MIDI Status byte - NoteOn, Chnl 0 (low nibble)
```

```
95  PEDAL_IN equ        H'04'       ;Pedal Switch connected to RB4
96  MIDI_KEY equ        H'30'       ;MIDI key value 00 thru 7F (arbitrary)
97  MIDI_VEL equ        H'60'       ;MIDI Note Velocity (arbitrary)
98  ;-------------------------------------------------------------------------------------------------
99
100             org         0x0000      ;Reset/PowerUp vector
101             goto        initialize
102
103             org         0x0004
104             goto        main        ;Interrupt Routine vector, GIE disabled
105 ;
106 ;-------------------------------------------------------------------------------------------------
107 ;       INITIALIZE Control Registers
108 ;-------------------------------------------------------------------------------------------------
109  initialize MOVLF    B'00000000', PIR1       ;Clear Periph Interrupt Request Flags
110             MOVLF    B'10000000', RCSTA      ;Enable UART
111
112             bsf      STATUS, RP0             ;switch to bank 1
113 ;
114             MOVLF    B'11111111', TRISA      ;Port A is all inputs
115             MOVLF    B'00010000', TRISB      ;RB5=TX, RB4=pedal in
116             MOVLF    B'01111100', OSCCON     ;OSC Control Reg
117             MOVLF    B'00000000', OSCTUNE    ;OSC Tune the 8MHz internal clock
118             MOVLF    B'00000000', OPTION_REG         ;Enables PORTB pull-ups
119             MOVLF    B'00000000', PIE1       ;Disable peripheral interrupt
120             MOVLF    B'00000000', PIE2       ;Disable peripheral interupts
121             MOVLF    B'10100110', TXSTA      ;Set up Transmit 8-bit, asynch, hi-speed
122             MOVLF    B'00001111', SPBRG      ;Set Transmit Baud Rate to 31.25KHz
123 ;
124             bcf      STATUS, RP0             ;switch back to bank 0
125
126             MOVLF    B'10001000', INTCON     ;Enable "RB port change" interrupts
127                                              ;GIE and RBIE enables set
128 sleep_again sleep
129             goto     sleep_again             ;power down till pedal changes
130
131 ;-------------------------------------------------------------------------------------------------
132 ;           MAIN INTERRUPT ROUTINE (RBIF flag interrupt - the only kind enabled)
133 ;-------------------------------------------------------------------------------------------------
134 main        MOVLF    vel, MIDI_VEL                   ;Setting Key Velocity
135             btfss    PORTB, 4                ;If pedal switch is 0, Then key vel = 0
136                                              ;(to  switch polarity use btfsc instead)
137             clrf     vel                     ; Else key velocity = MIDI_VEL
138             call     midi_tx                 ;Call the Transmit Subroutine
139             bcf      INTCON, RBIF                    ;Clear RB port change interrupt f
140             retfie                           ;GIE re-enabled for interrupts
141
```

```
142
143 ;-------------------------------------------------------------------------------------------------------
144 ;                   SUBROUTINE FOR TRANSMITTING MIDI NOTE ON
145 ;-------------------------------------------------------------------------------------------------------
146 midi_tx     MOVLF     NOTE_ON, TXREG      ;Send the MIDI Note-On Status Byte
147             MOVLF     1, x                ;delay 0.39ms
148             call      delay
149             MOVLF     MIDI_KEY, TXREG     ;Send the MIDI Note-On Key value
150             MOVLF     1, x                ;delay 0.39ms
151             call      delay
152             MOVLF     vel, TXREG          ;Send the MIDI Note-On Key Velocity
153             MOVLF     0x33, x             ;Delay 20ms for switch debouncing
154             call      delay
155             return
156
157 ;
158 ;-------------------------------------------------------------------------------------------------------
159 ;               SUBROUTINE FOR DELAY
160 ; With y=255,  x=1 is 0.39ms, x=3 is 1.2ms, x=8 is 3ms, x=16 is 6ms, x=51 is 20ms
161 ;-------------------------------------------------------------------------------------------------------
162 delay       nop                         ;Delay = (3xy+5x+6)(1/2MHz) seconds
163 outer_loop  MOVLF     0xff, y
164 inner_loop  decfsz     y, F
165             goto      inner_loop
166             decfsz    x, F
167             goto      outer_loop
168             return
169 ;-------------------------------------------------------------------------------------------------------
170             end
171
172
173
174
175
176 ; John Talbert, Oberlin Conservatory, May, 2009
177
178
179
```