MIDI Controlled AC Outlets

Retrofit the Gemmy Holiday Lightshow with a PIC16F84 Controller Circuit
PIC MIDI Power Outlet Project

Outputs connect to 200 ohm inputs of Gemmy Lightshow. Each output draws (5v-1.2v)/350 = 11ma when high. Maximum load on the PIC is 8 x 11ma, under the 100ma PortB max allowed current.
RA4 connects to the MIDI Input since it is a Schmitt Trigger Input. RB0-RB7 will be outputs.

The 8 PIC RBx outputs will connect to optoisolator/triac circuits to control up to 8 AC Power Outlets. For testing purposes, they can be connected to 8 LEDs with 330 ohm series resistors.

MIDI data arrives at RA4 in serial form, coming one bit at a time at a frequency of 31.25 KHz. The program collects the bits by carefully counting processor cycles. With a 4MHz Crystal oscillator feeding the PIC Clock input, each midi input bit will ‘hang around’ for exactly 32 processor cycles. The program is set up to read each bit right in the middle of its 32 cycle time frame. Eight bits are collected at a time (one byte), framed by one low start bit at the beginning and one or more high stop bits at the end of each midi byte.

The MIDI input stream is searched for Note_On command bytes with a targeted MIDI channel number as set by NOTE_ON. When found, the Key Number / Key Velocity data byte pairs that follow the status byte are searched for 8 specific Key# values as set by BASE_KEY. When one is found, if the Key Velocity of the targeted key is non-zero, a corresponding PIC output pin will be set high, and if the Key Velocity is zero, the corresponding PIC output pin will be set low. The program supports 'running status' midi where several Key/Velocity pairs can follow one Note_On status byte.

------------------------------------------------------------------------------------------------------------------
| VARIABLES -- STORED IN REGISTERS ABOVE THE DEDICATED REGISTERS |
------------------------------------------------------------------------------------------------------------------
| cblock 0x0C |
| midi_key ;stores current midi key value |
| midi_note? ;Is the current MIDI Status = NOTE_ON? |
| midi_key? ;Has a Midi Key value been loaded for the current pair? |
| delay_x ;timing delay set |
| bitcount ;holds the bit position in the byte for getbyte |
| recv ;holds results of getbyte |
------------------------------------------------------------------------------------------------------------------
| USEFUL EQUATES |
------------------------------------------------------------------------------------------------------------------
| PORTA equ H'0005' |
| PORTB equ H'0006' |
| STATUS equ H'0003' |
| PCL equ H'0002' |
| RP0 equ H'0005' ;Bank Select bit in STATUS |
| C equ H'0000' ;Carry bit in STATUS |
| Z equ H'0002' ;Zero bit in STATUS |
| TRISA equ H'0085' ;Data Direction for PORTA |
| TRISB equ H'0086' ;Data Direction for PORTB |
F equ H'0001' ;Instruction results go to the specified register
W equ H'0000' ;Instruction results go to W
NOTE_ON equ H'0090' ;MIDI Status byte for Note On, Chnl 0 (lower nibble)
MIDI_IN equ H'0004' ;Midi_In connected to RA4, a Schmitt Trig Input
BASE_KEY equ H'0030' ;valid midi keys are decimal 48-55, hex 30-37
;lower 3 bits of BASE_KEY need to be 0

;--------------------------------------------------------------------------------------------------------------------
org 0x00
;---------------------------------------------------------------------------------------------------------------------
; INITIALIZE Port A as inputs, Port B as outputs.
;---------------------------------------------------------------------------------------------------------------------

bsf STATUS, RP0 ;switch to bank 1
movlw 0xff
movwf TRISA ;Port A is all inputs
movlw 0x00
movwf TRISB ;Port B is outputs
bcf STATUS, RP0 ;switch back to bank 0

;---------------------------------------------------------------------------------------------------------------------
; ROUTINE FOR RECEIVING ONE MIDI INPUT BYTE
;---------------------------------------------------------------------------------------------------------------------

getbyte btfsc PORTA, MIDI_IN ;Wait for beginning of start bit (a low)
goto getbyte ;Detected start byte
movlw 8
movwf bitcount ;8 bits of data to get
clrf recv ;empty the recv register
nop
nop
movlw 0xC
movwf delay_x ;11cycles here + 37 cycles from delay
call delay ;go to center of first bit (16 + 32 cycles)

getbit bcf recv, 7
btfsc PORTA, MIDI_IN ;receive Midi bits, LSB first
bsf recv, 7 ;MSB of recv = Midi bit
decfsz bitcount, F ;Decrement the bit count
goto continue
goto done

continue rrf recv, F ;rotate bits down the recv register
movlw 0x6
movwf delay_x ;13 cycles here + 19 from delay
call delay ;go to middle of next Midi bit time (32 cycles)
goto getbit
done movlw 0x4 ;Finished collecting the MIDI byte
nop
nop
movwf   delay_x ; 13 cycles here + 13 from delay
call    delay   ; Go a little into the Stop Bit (16 + 10 cycles)

btfs   PORTA, MIDI_IN ; Test for framing error, stop bit should be high
goto   parse    ; Tested OK, go on to parsing routine

bcf     midi_note?, 1 ; Framing error - no Stop Bit
bcf     midi_key?, 1

stopbit btfs   PORTA, MIDI_IN ; find a high in the midi stream
goto   stopbit ; hopefully between midi byte sends
goto   getbyte

----------------------------------------------------------------------------------------------------------------------------

;  MIDI DATA PARSE -- COLLECT DATA FROM A VALID MIDI NOTE-ON COMMAND
;  (See logic flow chart)
----------------------------------------------------------------------------------------------------------------------------

parse  btfs   recv, 7 ; check MSB to determine data or status byte
goto   data_byte ; ------Midi Status byte------

status_byte bcf   midi_key?, 1 ; clear midi_key? flag
bsf     midi_note?, 1 ; For now, set midi_note? flag high
movlw   NOTE_ON ; Test Midi Status byte for NOTE_ON
subwf   recv, W
btfs   STATUS, Z ; If the Midi Status Byte is not = NOTE_ON
bcf     midi_note?, 1 ; Then set midi_note? flag low, else leave high

goto   getbyte ; Get the next midi byte

; ------Midi Data byte------
data_byte btfs   midi_note?, 1 ; If current Midi Status is not = NOTE_ON
goto   getbyte ; Then I'm not interested, get the next byte, else...

btfs   midi_key?, 1 ; If a Key value has already been collected

goto   check ; Then this must be Vel, go on to check for valid key#

movf   recv, W
movwf   midi_key

bsf     midi_key?, 1

goto   getbyte

; We have our key#/velocity pair, check it
; Key# is in midi_key, Velocity is in recv

check  bcf   midi_key?, 1 ; clear midi_key? flag to get ready for next pair

movf   midi_key, W ; Check for valid midi key value

andlw  B'11111000' ; Mask out the lower 3 bits

sublw  BASE_KEY ; Compare with BASE_KEY

btfs   STATUS, Z ; If not a valid midi key value

goto   getbyte ; then I'm not interested, get another byte

; else very interested, go to the output routine

----------------------------------------------------------------------------------------------------------------------------

;  OUTPUT ROUTINE -- MIDI NOTES ON/OFF TRANSLATE TO PORTB BITS ON/OFF
; We have a Midi Note-On command with a valid key value in midi_key and key velocity in recv.

    movlw   B'00000111'   ; mask for midi key value
    andwf   midi_key, F   ; lower three bits of midi key
    bcf     STATUS, C
    rlf     midi_key, F   ; key = key*2
    movf    recv, F
    btfsc   STATUS, Z
    goto    off
    call    bit_on
    goto    getbyte

bit_on   movf    midi_key, W   ; offset value
    addwf   PCL, F
    bcf     PORTB, 0
    return
    bcf     PORTB, 1
    return
    bcf     PORTB, 2
    return
    bcf     PORTB, 3
    return
    bcf     PORTB, 4
    return
    bcf     PORTB, 5
    return
    bcf     PORTB, 6
    return
    bcf     PORTB, 7
    return

off      call    bit_off
    goto    getbyte

bit_off  movf    midi_key, W   ; offset value
    addwf   PCL, F
    bcf     PORTB, 0
    return
    bcf     PORTB, 1
    return
    bcf     PORTB, 2
    return
    bcf     PORTB, 3
    return
    bcf     PORTB, 4
    return
    bcf     PORTB, 5
    return
    bcf     PORTB, 6
    return
    bcf     PORTB, 7
return

;---------------------------------------------------------------------------------------------------------------------
;        DELAY SUBROUTINE FOR TIMING THE MIDI SERIAL INPUT CAPTURE
;---------------------------------------------------------------------------------------------------------------------
delay decfsz delay_x, F ; (x-1)*3 + 4 cycles
  goto delay ;cycles = 4, 7, 10, 13, 16, 19, 22, 25, 28
return

;---------------------------------------------------------------------------------------------------------------------

end

; John Talbert, Oberlin Conservatory, March 2009
MIDI Power Outlet Project

**midi_note?** - Is the current Midi Status a Note On with Midi Channel = 00?

**midi_key?** - Is the midi_key data current? In which case, this second data byte must be the key velocity data.

* set up for "Running Status" MIDI
The controller outputs 0v to 5v control signals and 0 amps to 20ma.

One of 6 circuits

AC White are all connected together
AC Green are all connected together
For 115/240 Vac (rms) Application:
1. Solonoid/Valve Controls
2. Lighting Controls
3. Static Power Switches
4. Ac Motor Drives
5. Temperature Controls
6. E.M. Contactors
7. Ac Motor Starters
8. Solid State Relays

Outside Dimension: Unit (mm)

Schematic: Top View

Absolute Maximum Ratings

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Rating</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Forward current</td>
<td>IF</td>
<td>50</td>
<td>mA</td>
</tr>
<tr>
<td>Peak forward current</td>
<td>IFM</td>
<td>1</td>
<td>A</td>
</tr>
<tr>
<td>Reverse voltage</td>
<td>VR</td>
<td>6</td>
<td>V</td>
</tr>
<tr>
<td>Power dissipation</td>
<td>PD</td>
<td>70</td>
<td>mW</td>
</tr>
<tr>
<td>Output Off-State Output Terminal voltage</td>
<td>VORM</td>
<td>400</td>
<td>Vpeak</td>
</tr>
<tr>
<td>Peak Repetitive Surget Current</td>
<td>ITSM</td>
<td>1</td>
<td>A</td>
</tr>
<tr>
<td>Power dissipation</td>
<td>PD</td>
<td>300</td>
<td>mW</td>
</tr>
<tr>
<td>Total power dissipation</td>
<td>Plot</td>
<td>330</td>
<td>mW</td>
</tr>
<tr>
<td>Isolation voltage 1 minute</td>
<td>Viso</td>
<td>5000</td>
<td>Vms</td>
</tr>
<tr>
<td>Operating temperature</td>
<td>Topr</td>
<td>-40 to +80</td>
<td>°C</td>
</tr>
<tr>
<td>Storage temperature</td>
<td>Tstg</td>
<td>-40 to +125</td>
<td>°C</td>
</tr>
<tr>
<td>Snooking temperature 10 seconds</td>
<td>Tsnk</td>
<td>260</td>
<td>°C</td>
</tr>
</tbody>
</table>

Electro-optical Characteristics

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Conditions</th>
<th>MIN.</th>
<th>TYP.</th>
<th>MAX.</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Forward voltage</td>
<td>Vf</td>
<td>IF = 10mA</td>
<td>—</td>
<td>1.2</td>
<td>1.4</td>
<td>V</td>
</tr>
<tr>
<td>Peak forward voltage</td>
<td>VfM</td>
<td>IFM = 0.5A</td>
<td>—</td>
<td>—</td>
<td>3.5</td>
<td>V</td>
</tr>
<tr>
<td>Reverse Leakage Current</td>
<td>VR</td>
<td>VR = 4V</td>
<td>—</td>
<td>—</td>
<td>10</td>
<td>UA</td>
</tr>
<tr>
<td>Output Peak Blocking Current</td>
<td>IDRM</td>
<td>VORM = Rated</td>
<td>—</td>
<td>—</td>
<td>10²</td>
<td>A</td>
</tr>
<tr>
<td>On-State Voltage</td>
<td>V1M</td>
<td>IT1M = 100mA</td>
<td>—</td>
<td>1.5</td>
<td>3</td>
<td>V</td>
</tr>
<tr>
<td>Transfer characteristics</td>
<td>Holding Current</td>
<td>IH</td>
<td>—</td>
<td>100</td>
<td>—</td>
<td>uA</td>
</tr>
<tr>
<td>Critical rate of rise of OFF-state voltage</td>
<td>dV/dt</td>
<td>V(DRM) = (1/2) * Rated</td>
<td>600</td>
<td>—</td>
<td>—</td>
<td>V/µS</td>
</tr>
<tr>
<td>Isolation resistance</td>
<td>Riso</td>
<td>DC500V</td>
<td>5 x 10¹²</td>
<td>10¹⁰</td>
<td>—</td>
<td>ohm</td>
</tr>
<tr>
<td>Minimum trigger current</td>
<td>IFT</td>
<td>Main Terminal Voltage = 3V</td>
<td>—</td>
<td>—</td>
<td>5</td>
<td>mA</td>
</tr>
<tr>
<td>Turn-on time</td>
<td>Ton</td>
<td>VD = 6V, RL = 100 ohm, IF = 20mA</td>
<td>—</td>
<td>—</td>
<td>100</td>
<td>µS</td>
</tr>
</tbody>
</table>
SanRex Triac TMG8C60 is designed for full wave AC control applications. It can be used as an ON/OFF function or for phase control operation.

**Typical Applications**
- Home Appliances: Washing Machines, Vacuum Cleaners, Rice Cookers, Microwave Ovens, Hair Dryers, other control applications
- Industrial Use: SMPS, Copier Machines, Motor Controls, Dimmer, SSR, Heater Controls, Vending Machines, other control applications

**Features**
- \( I_{\text{TRMS}} = 8 \) A
- High Surge Current
- Low Voltage Drop
- Lead-Free Package

### Maximum Ratings

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Item</th>
<th>Reference</th>
<th>Ratings</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>( V_{\text{DRM}} )</td>
<td>Repetitive Peak Off-State Voltage</td>
<td></td>
<td>600</td>
<td>V</td>
</tr>
<tr>
<td>( I_{\text{T(RMS)}} )</td>
<td>R.M.S. On-State Current</td>
<td>( T_c = 105^\circ )C</td>
<td>8</td>
<td>A</td>
</tr>
<tr>
<td>( I_{\text{TSM}} )</td>
<td>Surge On-State Current</td>
<td>One cycle, 50Hz/60Hz, Peak value non-repetitive</td>
<td>80/88</td>
<td>A</td>
</tr>
<tr>
<td>( I_T )</td>
<td>( I_T ) (for fusing)</td>
<td></td>
<td>32</td>
<td>A*S</td>
</tr>
<tr>
<td>( P_{\text{GM}} )</td>
<td>Peak Gate Power Dissipation</td>
<td></td>
<td>5</td>
<td>W</td>
</tr>
<tr>
<td>( P_{\text{G(AV)}} )</td>
<td>Average Gate Power Dissipation</td>
<td></td>
<td>0.5</td>
<td>W</td>
</tr>
<tr>
<td>( I_{\text{GM}} )</td>
<td>Peak Gate Current</td>
<td></td>
<td>2</td>
<td>A</td>
</tr>
<tr>
<td>( V_{\text{GM}} )</td>
<td>Peak Gate Voltage</td>
<td></td>
<td>10</td>
<td>V</td>
</tr>
<tr>
<td>( T_j )</td>
<td>Operating Junction Temperature</td>
<td>(-40 \sim +125) °C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( T_{\text{stg}} )</td>
<td>Storage Temperature</td>
<td>(-40 \sim +150) °C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mass</td>
<td></td>
<td></td>
<td>2</td>
<td>g</td>
</tr>
</tbody>
</table>

### Electrical Characteristics

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Item</th>
<th>Reference</th>
<th>Ratings</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>( I_{\text{DRM}} )</td>
<td>Repetitive Peak Off-State Current</td>
<td>( V_0 = V_{\text{DRM}}, ) Single phase, half wave, ( T_j = 125^\circ )C</td>
<td>2</td>
<td>mA</td>
</tr>
<tr>
<td>( V_{\text{TM}} )</td>
<td>Peak On-State Voltage</td>
<td>( I_T = 12A, ) Inst. measurement</td>
<td>1.4</td>
<td>V</td>
</tr>
<tr>
<td>( I_{\text{GT1}} )</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( I_{\text{GT1}} )</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( I_{\text{GT3}} )</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( I_{\text{GT3}} )</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( V_{\text{GT1}} )</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( V_{\text{GT1}} )</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( V_{\text{GT3}} )</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( V_{\text{GT3}} )</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( V_{\text{GD}} )</td>
<td>Non-Trigger Gate Voltage</td>
<td>( T_j = 125^\circ )C, ( V_0 = \frac{1}{2} V_{\text{DRM}} )</td>
<td>0.2</td>
<td>V</td>
</tr>
<tr>
<td>( (dv/dt)_{\text{C}} )</td>
<td>Critical Rate of Rise of Off-State Voltage at Commutation</td>
<td>( T_j = 125^\circ )C, ( (dv/dt)<em>{\text{C}} = -4A/\mu s ), ( V_0 = \frac{1}{2} V</em>{\text{DRM}} )</td>
<td>10</td>
<td>V/μs</td>
</tr>
<tr>
<td>( I_H )</td>
<td>Holding Current</td>
<td></td>
<td>15</td>
<td>mA</td>
</tr>
<tr>
<td>( R_{\text{th}} )</td>
<td>Thermal Resistance</td>
<td>Junction to case</td>
<td>2.0</td>
<td>°C/W</td>
</tr>
</tbody>
</table>