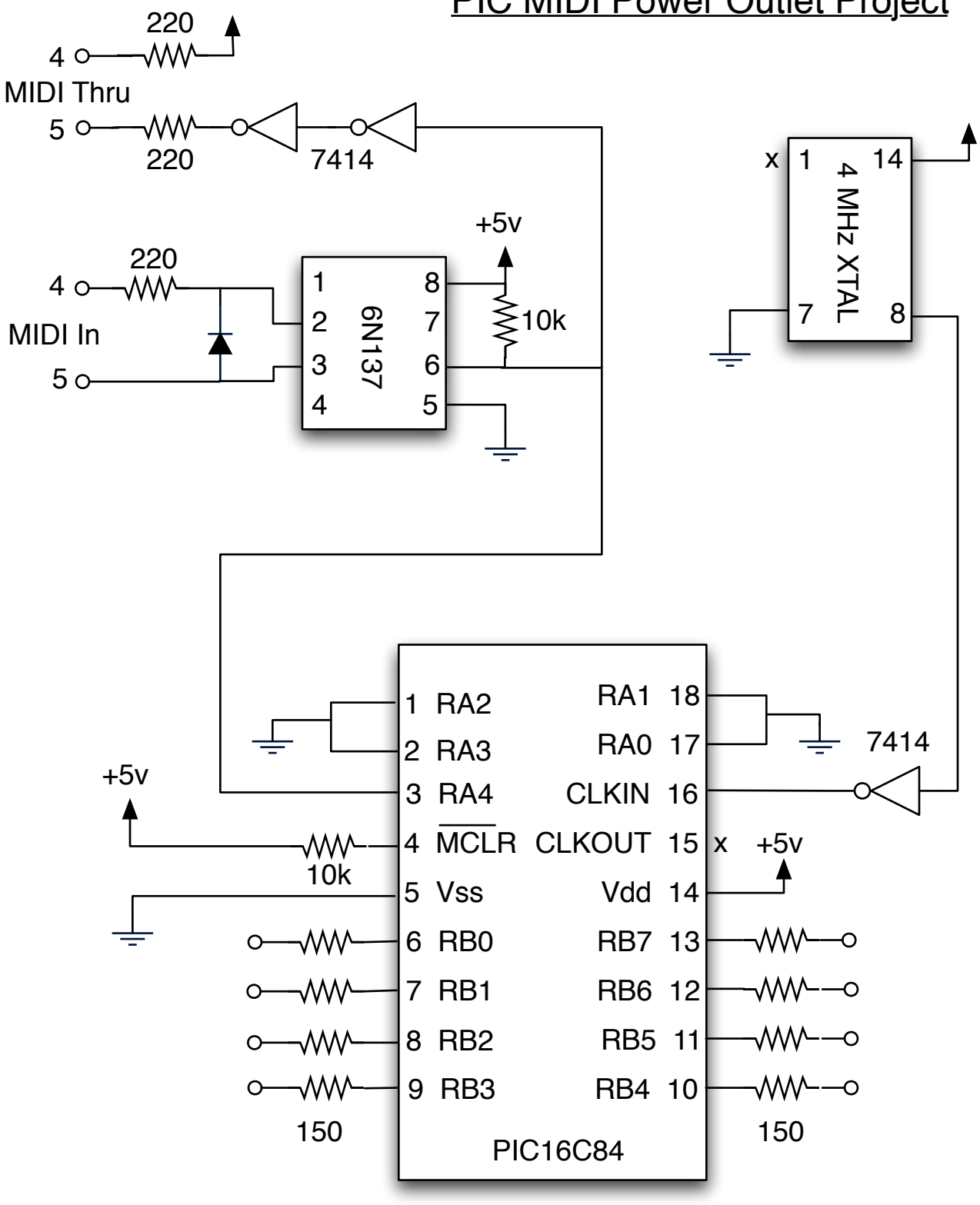
MIDI Controlled AC Outlets

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.

```
;_____PIC MIDI POWER OUTLET PROJECT_____
;
;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 staus byte.
;

; ------------------------------------------------------------------------------------------------------------
                list            p=16c84
;               __FUSES  _CP_OFF  &  _WDT_OFF  &  _XT_OSC  &  _PWRTE_ON
; ------------------------------------------------------------------------------------------------------------
; 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

                endc
; ------------------------------------------------------------------------------------------------------------
;                   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                 ;13 cycles here + 19 from delay
            movwf   delay_x             ;go to middle of next Midi bit time (32 cycles)
            call    delay
            goto    getbit

                                        ;Finished collecting the MIDI byte
done        movlw   0x4
            nop
            nop
```

```
                movwf   delay_x                 ;13 cycles here + 13 from delay
                call    delay                   ;Go a little into the Stop Bit (16 + 10 cycles)

                btfsc   PORTA, MIDI_IN          ;Test for framing error, stop bit should be high
                goto    parse                   ;Tested OK, go on to parsing routine
                                                ;Framing error - no Stop Bit
                bcf     midi_note?, 1           ;reset flags to start over after framing error
                bcf     midi_key?, 1
stopbit         btfss   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           btfss   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
                btfss   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       btfss   midi_note?, 1           ;If current Midi Status is not = NOTE_ON
                goto    getbyte                 ;Then I'm not interested, get the next byte, else…
                btfsc   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                 ;Else store this new key value, and set flag
                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
                btfss   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                 ;Test for zero velocity (= note off)
                btfsc   STATUS, Z
                goto    off
                call    bit_on
                goto    getbyte

bit_on          movf    midi_key, W             ;offset value
                addwf   PCL, F                  ;offset to one of the opcodes below
                bsf     PORTB, 0
                return
                bsf     PORTB, 1
                return
                bsf     PORTB, 2
                return
                bsf     PORTB, 3
                return
                bsf     PORTB, 4
                return
                bsf     PORTB, 5
                return
                bsf     PORTB, 6
                return
                bsf     PORTB, 7
                return

off             call    bit_off
                goto    getbyte

bit_off         movf    midi_key, W             ;offset value
                addwf   PCL, F                  ;offset to one of the opcodes below
                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
```

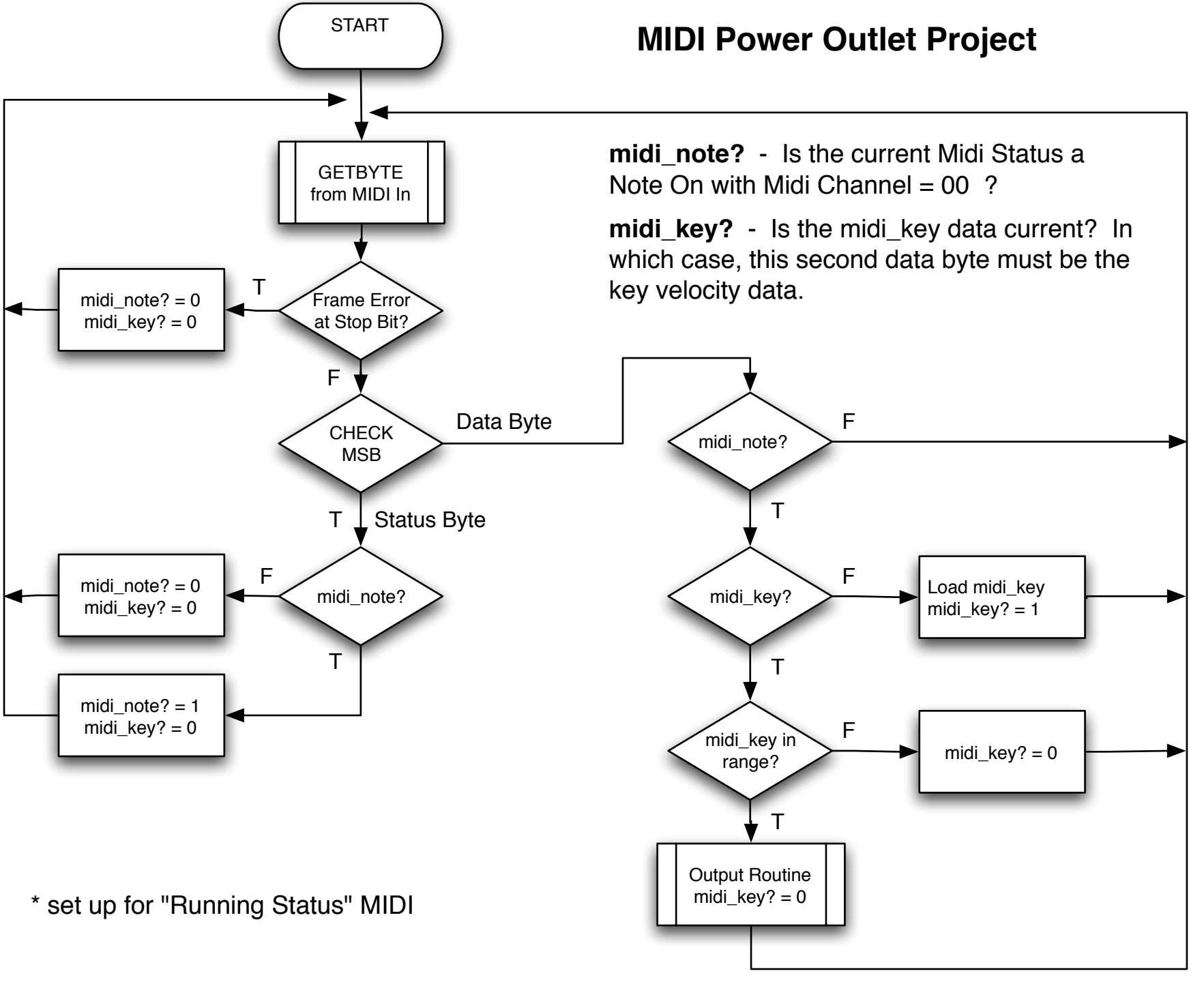
```
                return

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

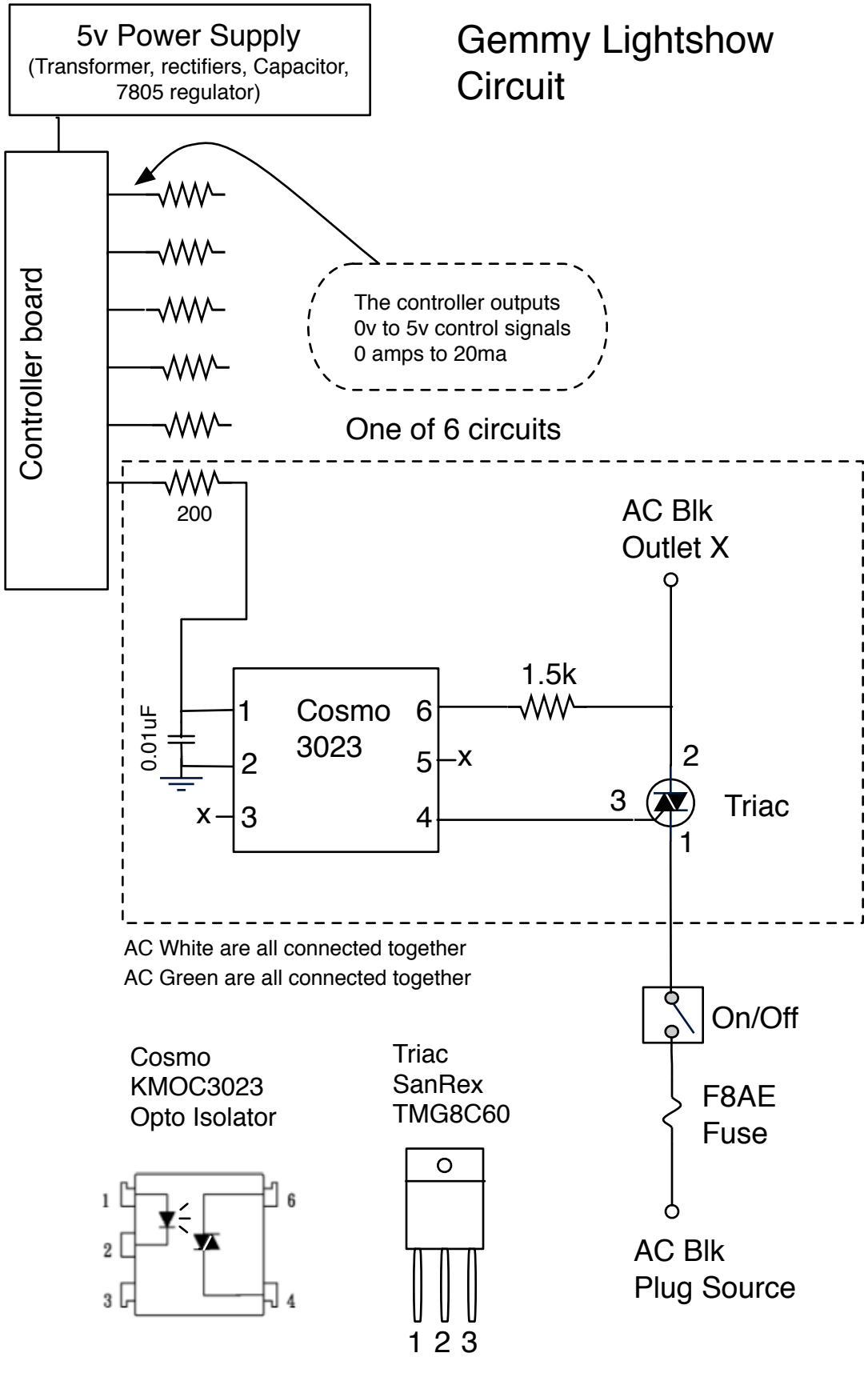


midi_note? - Is the current Midi Status a Note On with Midi Channel = 00 ?

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

Gemmy Lightshow Circuit



cosmo

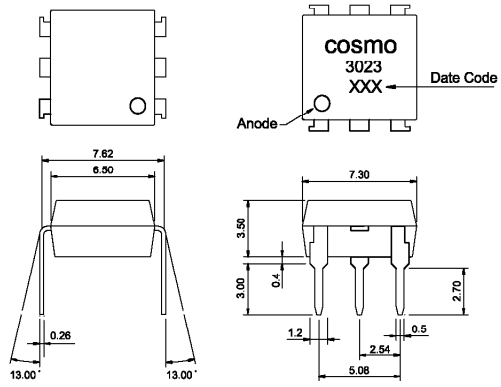
Optoisolators Triac Driver Output (400V Volts Peak) KMOC3023

UL 1577 (File No.E169586) **VDE** 0884 / 0860 / 0805 (File No.101347)

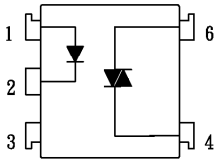
For 115/240 Vac (rms) Application:

1. Solenoid/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
9. Available package : DIP/ SMD/ H.

Outside Dimension : Unit (mm)



Schematic : Top View



1. Anode
2. Cathode
3. NC
4. MAIN TERMINAL
6. MAIN TERMINAL

Absolute Maximum Ratings

(Ta=25°C)

	Parameter	Symbol	Rating	Unit
Input	Forward current	I_F	50	mA
	Peak forward current	I_{FM}	1	A
	Reverse voltage	V_R	6	V
	Power dissipation	P_D	70	mW
Output	Off-State Output Terminal voltage	V_{DRM}	400	Vpeak
	Peak Repetitive Surget Current	I_{TSM}	1	A
	Power dissipation	P_D	300	mW
	Total power dissipation	Ptot	330	mW
	Isolation voltage 1 minute	Viso	5000	Vrms
	Operating temperature	Topr	-40 to +80	°C
	Storage temperature	Tstg	-40 to +125	°C
	Soldering temperature 10 seconds	Tsol	260	°C

Electro-optical Characteristics

(Ta=25°C)

	Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input	Forward voltage	V_F	I_F =10mA	—	1.2	1.4	V
	Peak forward voltage	V_{FM}	I_{FM} =0.5A	—	—	3.5	V
	Reverse Leakage Current	I_R	V_R =4V	—	—	10	UA
Output	Peak Blocking Current	I_{DRM}	V_{DRM} =Rated	—	—	10^{-7}	A
	ON-State Voltage	V_{TM}	I_{TM} =100mA	—	1.6	3	V
Transfer charac-teristics	Holding Current	I_H		—	100	—	uA
	Critical rate of rise of OFF-state voltage	dV/dt	V_{DRM}= (1/√2) *Rated	600	—	—	V/uS
	Isolation resistance	Riso	DC500V	$5x10^{10}$	10^{11}	—	ohm
	Minimum trigger current	I_{FT}	Main Terminal Voltage=3V	—	—	5	mA
	Turn-on time	Ton	V_D =6v,R_L =100 ohm,I_F =20mA	—	—	100	uS

Fig.1 Forward Current vs. Ambient Temperature

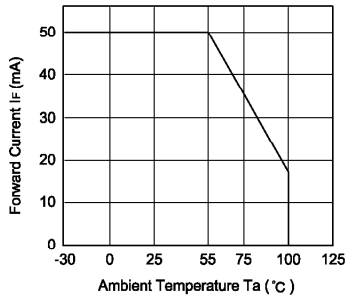


Fig.2 Diode Power Dissipation vs. Ambient Temperature

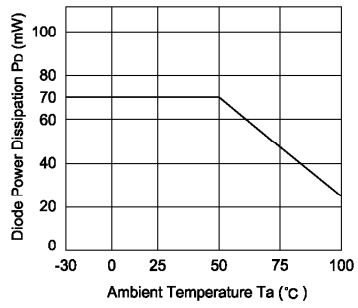


Fig.3 On-State R.M.S. Current vs. Ambient Temperature

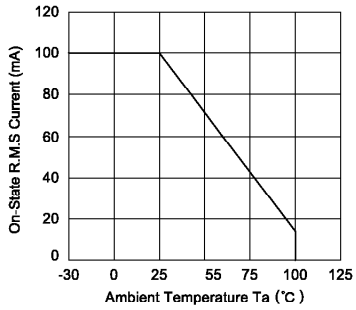


Fig.4 Total Power Dissipation vs. Ambient Temperature

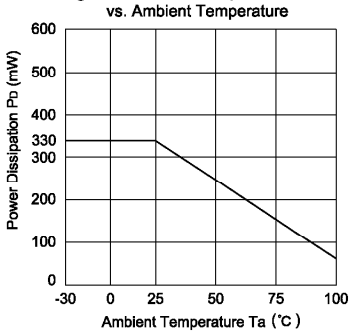


Fig.5 Peak Forward Current vs. Duty Ratio

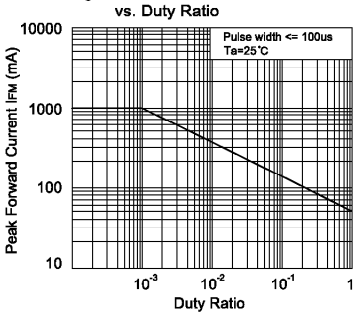


Fig.6 Forward Current vs. Forward Voltage

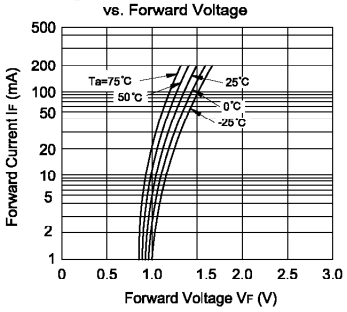


Fig.7 On-State Characteristics

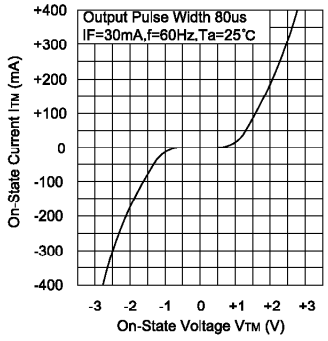


Fig.8 Leakage with LED off vs. Ambient Temperature

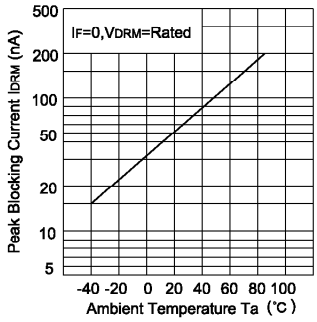
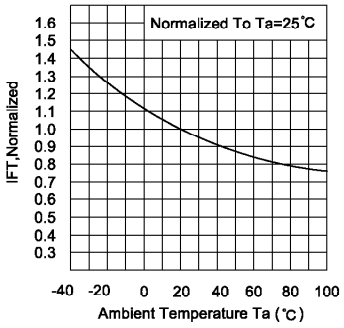


Fig.9 Tigger Current vs. Ambient Temperature



TRIAC(Through Hole / Non-isolated)

TMG8C60

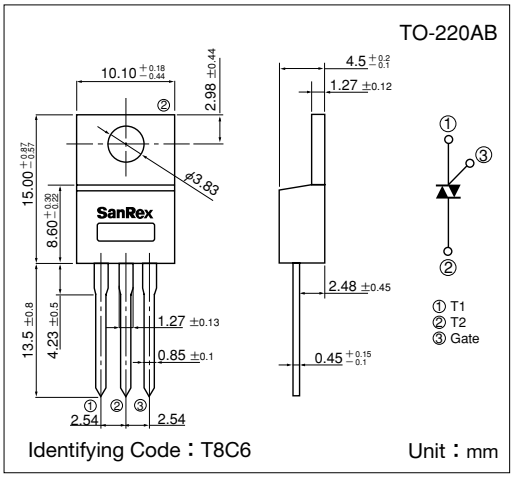
SanRex Triac **TMG8C60** is designed for full wave AC control applications.
It can be used as an ON/OFF function or for phase control operation.

Typical Applications

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

Features

- $I_{T(RMS)}$=8A
- High Surge Current
- Low Voltage Drop
- Lead-Free Package



■Maximum Ratings

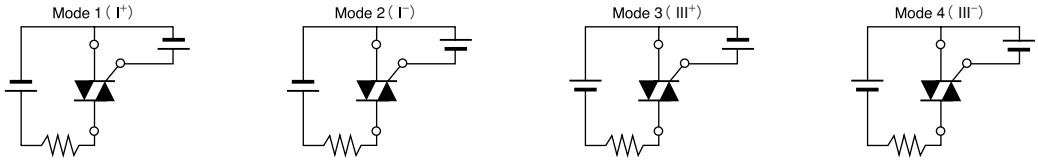
(Tj=25℃ unless otherwise specified)

Symbol	Item	Reference	Ratings	Unit
V_{DRM}	Repetitive Peak Off-State Voltage		600	V
$I_{T(RMS)}$	R.M.S. On-State Current	Tc=105℃	8	A
I_{TSM}	Surge On-State Current	One cycle, 50Hz/60Hz, Peak value non-repetitive	80/88	A
I^2t	I^2t (for fusing)		32	A^2S
P_{GM}	Peak Gate Power Dissipation		5	W
$P_{G(AV)}$	Average Gate Power Dissipation		0.5	W
I_{GM}	Peak Gate Current		2	A
V_{GM}	Peak Gate Voltage		10	V
Tj	Operating Junction Temperature		−40～+125	℃
Tstg	Storage Temperature		−40～+150	℃
	Mass		2	g

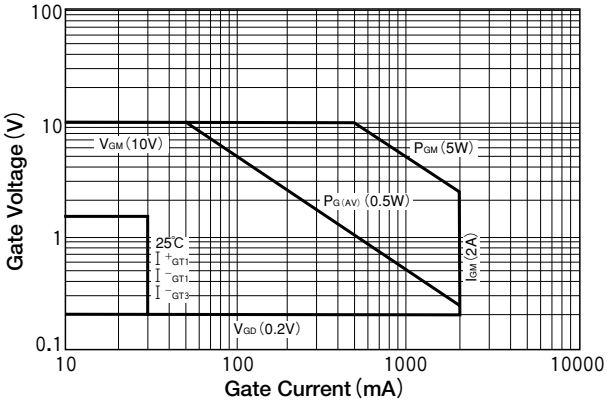
■Electrical Characteristics

Symbol		Item	Reference	Ratings			Unit
				Min.	Typ.	Max.	
I_{DRM}		Repetitive Peak Off-State Current	$V_D=V_{DRM}$, Single phase, half wave, Tj=125℃			2	mA
V_{TM}		Peak On-State Voltage	I_T=12A, Inst. measurement			1.4	V
I_{GT1}^{+}	1	Gate Trigger Current	V_D=6V, R_L=10Ω			30	mA
I_{GT1}^{-}	2					30	
I_{GT3}^{+}	3					—	
I_{GT3}^{-}	4					30	
V_{GT1}^{+}	1	Gate Trigger Voltage				1.5	V
V_{GT1}^{-}	2					1.5	
V_{GT3}^{+}	3					—	
V_{GT3}^{-}	4					1.5	
V_{GD}		Non-Trigger Gate Voltage	Tj=125℃, $V_D=\frac{1}{2}V_{DRM}$	0.2			V
(dv/dt)c		Critical Rate of Rise of Off-State Voltage at Commutation	Tj=125℃, (di/dt)c=−4A/ms, $V_D=\frac{2}{3}V_{DRM}$	10			V/μs
I_H		Holding Current			15		mA
Rth		Thermal Resistance	Junction to case			2.0	℃/W

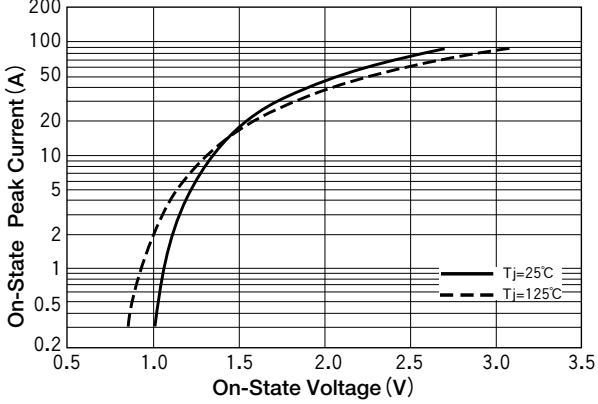
Trigger mode of the triac



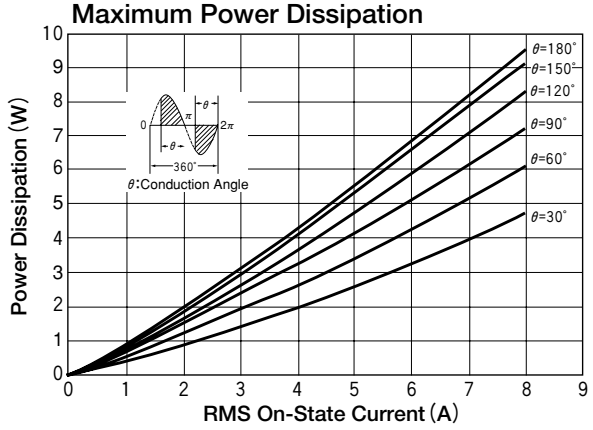
Gate Characteristics



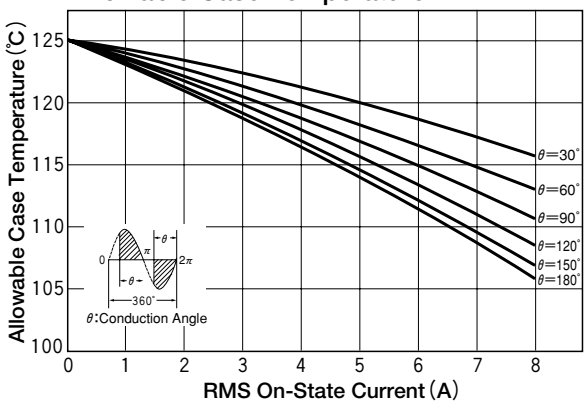
On-State Characteristics (MAX)



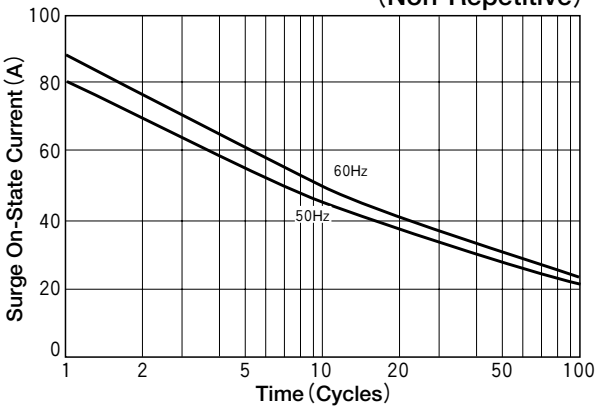
RMS On-State vs Maximum Power Dissipation



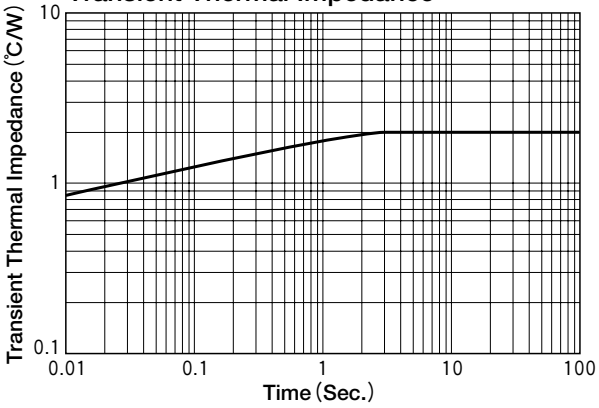
RMS On-State vs Allowable Case Temperature



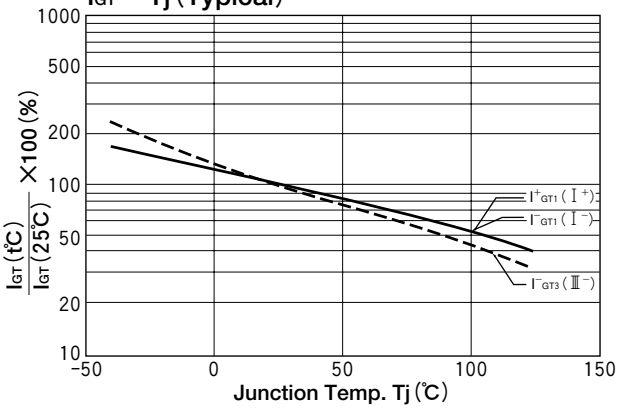
Surge On-State Current Rating (Non-Repetitive)



Transient Thermal Impedance



IGT — Tj (Typical)



VGT — Tj (Typical)