Light Switch with Delay Turn-Off

Author: Brian Iehl
brian@dls.net

OVERVIEW

For cost effectiveness a PIC12C508 microcontroller is used. This light switch allows the user to turn off a light switch then have the circuit wait until the user has left the room to actually turn the light off. This increases safety in the home or work place by allowing people to not have to walk through the dark. There is no delay when the light is turned on. This device is intended to be used in stand alone lamps or in wall light switches. Therefore, this device needs to be low cost, small in size, and powered off of the AC power line.

APPLICATION OPERATION

This circuit monitors the state of a light switch. If the switch turns on, the microcontroller turns on a triac which in turn switches on the light. When the switch is turned off, the microcontroller waits 15 seconds (or other predetermined time), then turns the triac off which in turn switches the light off.

Since the PIC12C508 draws so little current, a very simple, low cost power supply circuit is used which does not need a transformer. This allows the entire circuit to be very small. The PIC12C508 is perfectly suited to this application because it doesn't need any supporting circuitry. Also, this device is in an amazingly small package which will allow the circuit to mounted in a standard wall light switch. Even though the program memory in the PIC12C508 is limited to 512 bytes, that is still about 10 times more than is required for this simple application. This allows the microcontroller to be inexpensive enough to be used in cost sensitive applications like this one where it replaces standard discrete devices.

MICROCHIP TOOLS USED

Assembler/Compiler version:

MPLAB™ 3.22.00

APPENDIX A: SOURCE CODE

```
;                Title  "Light Switch"
;                Subtitle"Version 1.0"
;                Written by Brian Iehl 6/28/97

        list p=12C508

                 INCLUDE c:\apps\mplab\p12c508.inc
SetIO            equ    B'00001000'            ; 0 for output, 1 for input

GPIO0            equ    0
GPIO1            equ    1
GPIO2            equ    2
GPIO3            equ    3
GPIO4            equ    4
GPIO5            equ    5

SwState          equ    GPIO3                  ; Input
SwOn             equ    0                      ; Switch closed GPIO0 is Low
SwOff            equ    1                      ; Switch open GPIO0 is high
SwValue          equ    B'00001000'            ; Used to test GPIO3 bit

LitCntl          equ    GPIO0                  ; Output
LitOn            equ    1                      ; Hi to turn light on
LitOff           equ    0                      ; Lo to turn light off

Debounce equ     D'100'                        ; mS to wait for switch to settle
LitDelay equ     D'15'                         ; S to wait for light to turn off

ScratchPadRam    equ     0x07
OldSwState       equ    ScratchPadRam+0        ; Save last state
NewSwState       equ    ScratchPadRam+1        ; Save current state
DelayValue       equ    ScratchPadRam+2        ; For DelayRoutine
SDelayValue      equ    ScratchPadRam+3        ; For SDelayRoutine

;********************  Macros ***********************************************

MOVLF            MACRO  LL,         FF         ; Move Literal to register file
                 MOVLW  LL                     ; Load literal
                 MOVWF  FF                     ; Store in register file
                 ENDM                          ; end MOVLF

                                               ; Assumes 4 MHz clock; Needs global DelayValue
mSDelay          MACRO  mS                     ; Number of mS to delay up to 255 mS
                                               ; each clock cycle is 1 uS = .001 mS
                 LOCAL  Loop, SetTmr
                 MOVLF  mS,         DelayValue ; store number of mS delay
                 CLRWDT                        ; avoid unitentional reset

SetTmr           MOVLW  B'00000111'            ; Set prescaler to 256, clear PSA, Clear T0CS
                 OPTION                        ; store prescaler value

                 MOVLF  -4,         TMR0       ; 4 * 256 = 1024 uS ~ 1 mS
Loop             MOVF   TMR0,       w          ; force check zero
                 BTFSS  STATUS,     Z          ; w = 0 if same, so Z is set
                 goto   Loop                   ; not 0 so loop again
                                               ; one more mS passed
                 DECFSZ DelayValue, f          ; count down number of mS
                 goto   SetTmr                 ; not done reset timer
                                               ; if DelayValue = 0 then done
                 ENDM                          ; end mSDelay

                                               ; Macro SDelay; Needs global SDelayValue
SDelay           MACRO  S                      ; Number of Seconds delay up to 63
                 LOCAL Loop
                 MOVLF   S*4,       SDelayValue ; store number of S delay
Loop             mSDelay D'250'                 ; Delay 0.25 sec
                 DECFSZ  SDelayValue,f          ; count down number of S
                 goto    Loop                   ; not done reset timer
                                                ; if DelayValue = 0 then done

                 ENDM                           ; end SDelay


;*******************************************************************

        org     0x0A       ;start address 0x0A
        goto    Start
        org     0x10

;
Start


Setup            MOVLW   SetIO                  ; Load IO configuration byte
                 TRIS    GPIO                   ; Set GPIO with contents of w
                 CLRF    OldSwState             ; Init RAM
                 CLRF    NewSwState
                 CLRF    DelayValue
                 CLRF    SDelayValue

                                                ; get state at power up
                 MOVF    GPIO,      w           ; read GPIO register
                 ANDLW   SwValue                ; Clear all bits except SwState
                 BTFSS   STATUS,    Z           ; if SwState = Lo
                 goto    TurnOff

TurnOn           BSF     GPIO,      LitCntl     ; Turn Light on
                 MOVLF   SwOn,      OldSwState  ; Save lastest switch state
                 goto    ReadIt

TurnOff          BCF     GPIO,      LitCntl     ; Turn Light Off
                 MOVLF   SwOff,     OldSwState  ; Save lastest switch state

ReadIt           mSDelay Debounce               ; Switch Debounce
                 MOVF    GPIO,                  w; read GPIO register
                 ANDWF   SwState,   w           ; Clear all bits except SwState
                 MOVWF   NewSwState             ; save for later reference
                 XORWF   OldSwState,w           ; Are they the same
                 BTFSC   STATUS,                Z; w = 0 if same, so Z is set
                 goto    ReadIt                 ; same to loop again
                                                ; not same so state was changed
                 MOVF    NewSwState,w           ; Load new switch state
                 BTFSC   STATUS,                Z; w = 0 if SwOn, so Z is set
                 goto    TurnOn                 ; Turn light on

SetTim           SDelay  LitDelay               ; Wait before turning light off
                 goto    TurnOff                ; Turn light off


                 END
```