

```

title "Pager Reset for MOOS Mooring"
;*****
; Program: Pager Reset for MOOS Mooring
; Date: 11/05/2002
; Revision: A
; Programmer: Lance McBride
; Target: PIC16F627 PDIP
;
; This will act as a state machine for incoming pages for resetting
; the controller
; Internal RC Oscillator: 4.00MHz (EACH INSTRUCTION=1.0us)
;*****
LIST P=16F627, F=INH32, R=HEX
errorlevel 0,-305
INCLUDE "c:\apps\mplab\P16f627.inc"

; Register addresses
j equ 0x20;
k equ 0x21;
i equ 0x22;
test equ 0x23;
test2 equ 0x24;

__CONFIG _CP_OFF & _WDT_ON & _INTRC_OSC_NOCLKOUT & _BODEN_ON & _MCLRE_ON & _PWRTE_ON

PAGE
; Mainline of FIRST

org 0
goto Initialize ; Initialize the system

org 4 ; interrupt handler
CLRWDT ; clear watchdog timer
bcf INTCON, INTF ; clear port RB0/INT interrupt flag
bcf INTCON, RBIF ; clear RB Port Change interrupt flag
RETFIE ; on wake up, goto RCVD state reenabling all interrupts

Initialize:
CLRWDT ; clear watchdog timer
; initialize registers
movlw 0x00
movwf i
movlw 0x00
movwf j
movlw 0x00
movwf k
movlw 0x00
movwf test

CLRWDT ; clear watchdog timer

; starts in bank 0

movlw 0x00 ; set port A output latch to
movwf PORTA ; all pins to logic low
bcf STATUS, RP1 ; goto bank 1 to set port direction
bsf STATUS, RP0
movlw 0x1C ; set up direction on PORTA<1:0> as
movwf TRISA ; all out, PORTA<2:4> as input
bcf STATUS, RP1 ; return to bank 0
bcf STATUS, RP0

CLRWDT ; clear watchdog timer

movlw 0x00 ; set port B output latch to
movwf PORTB ; all pins to logic low
bcf STATUS, RP1 ; goto bank 1 to set port direction
bsf STATUS, RP0
movlw 0xFF ; set up direction on PORTB as
movwf TRISB ; all inputs
bcf OPTION_REG, NOT_RBPU ; enable weak pullups on port B
bcf STATUS, RP1 ; return to bank 0
bcf STATUS, RP0

CLRWDT ; clear watchdog timer

; Initialize Ports
movlw 0x00 ; set all PORTA pins to logic 0
movwf PORTA

CLRWDT ; clear watchdog timer

; Initialize watchdog timer
bcf STATUS, RP1 ; goto bank 1 to set port direction
bsf STATUS, RP0
bsf OPTION_REG, PS0 ; set watchdog division ratio to 1:128
bsf OPTION_REG, PS1 ; for a 2.3 second time-out period
bsf OPTION_REG, PS2 ;

```

```

bsf    OPTION_REG, PSA          ; assign prescaler to watchdog timer
bcf    STATUS, RP1              ; return to bank 0
bcf    STATUS, RP0

; Initialize Interrupts
bcf    STATUS, RP1              ; goto bank 1 to set port direction
bsf    STATUS, RP0
bcf    OPTION_REG, INTEDG       ; set RB0/INT interrupt to falling edge triggered
bcf    STATUS, RP1              ; return to bank 0
bcf    STATUS, RP0
bcf    INTCON, INTF             ; clear port RB0/INT interrupt flag
bcf    INTCON, RBIF             ; clear RB Port Change interrupt flag
bsf    INTCON, INTE             ; enable RB0/INT Interrupt
bsf    INTCON, RBIE             ; enable port B change interrupt

INIT_PAGER:
; Initialize Pager by resetting pager
bsf    PORTA, 0                 ; - turn off pager
call   Delay                   ; - wait ~5 seconds
bcf    PORTA, 0                 ; - turn on pager

;*****
StartStateMachineLoop:
;*****

RST_Start:
    btfsc PORTB,0               ; wait for reset signal to start
    goto RST_Start

RST_End:
    btfss PORTB,0               ; wait for reset signal to end
    goto RST_End

ClearInterrupts:
    bcf    INTCON, INTF         ; clear port RB0/INT interrupt flag
    bcf    INTCON, RBIF         ; clear RB Port Change interrupt flag
    bsf    INTCON, GIE          ; enable global interrupt

WAIT:
    sleep                       ; wait for page
    nop
    btfss STATUS, NOT_TO        ; check for watchdog first
    goto Reset_Watchdog
    goto TEST                   ; if not, goto the next state

Reset_Watchdog:
    CLRWDT                      ; otherwise, reset the watchdog and
    goto WAIT                   ; sleep again

TEST:
    CLRWDT                      ; clear watchdog timer
    bcf    INTCON, GIE          ; disable Global Interrupt
    btfsc INTCON, GIE           ; Global Interrupts disabled?
    goto TEST

BUOY_RESET:
    ; reset buoy
    bsf    PORTA, 1
    call   Delay
    bcf    PORTA, 1

RCVD:
    btfss PORTB,0               ; wait for page to end
    goto RCVD

OFF:
    ; reset pager
    bsf    PORTA, 0
    call   Delay                 ; - turn off pager
    call   Delay                 ; - wait ~5 seconds
    bcf    PORTA, 0              ; - turn on pager

    goto StartStateMachineLoop

;*****
; PROCEDURE: Simple multiple instruction delay
; Total procedure = 2,109,488 cycles
; Time/cycle @ 4MHz = 1us
; Total delay time = 2.109488 seconds
;*****
Delay:
    movlw 0x13
    movwf i
DelayLoop1:    ; 8 * (6 + 263680)
    CLRWDT
    movlw 0xFF
    movwf j
DelayLoop2:    ; 256 * (6 + 1024)
    CLRWDT

```

```

    movlw 0xFF
    movwf k
DelayLoop3:      ; 256 * 4
    CLRWDT      ; clear watchdog timer
    decfsz k      ; reduce loop counter
    goto DelayLoop3
    decfsz j
    goto DelayLoop2
    decfsz i
    goto DelayLoop1

return

,*****
;
; PROCEDURE: Short Debounce Delay
; Total procedure = 199824 cycles
; Time/cycle @ 4MHz = 1us
; Total delay time = 0.199824 seconds
,*****
Short_Delay:      ; 4 + 199820

    movlw 0xC2
    movwf i
DelayLoop4:      ; 194 * (6 + 1024)
    CLRWDT

    movlw 0xFF
    movwf j
DelayLoop5:      ; 256 * 4
    CLRWDT
    decfsz j
    goto DelayLoop5

    decfsz i
    goto DelayLoop4

return

,*****
Finished:        ; now, just spin forever
goto $

end

```