

```

;
;           Leilani Electronics
;   program written & tested by William E. Nickles
;           Electronic Traffic Light (20mm)
;           rev 2.0 1/17/2025
;
;           list      p=12f609
;           __config  h'3c74'
;           radix     hex
;
status      equ      0x03
gpio        equ      0x05
tlcon       equ      0x10
intcon      equ      0x0b
option_reg  equ      0x81
trisio      equ      0x85
pcon        equ      0x8e
wpu         equ      0x95
ansel       equ      0x9f
;
temp0       equ      0x41
temp1       equ      0x42
temp2       equ      0x43
counter1    equ      0x44
counter2    equ      0x45
tcount      equ      0x46
count0      equ      0x47
;
c           equ      0           ; carrier bit
rp0         equ      5           ; bank select
;
sdi         equ      0           ; data output
clk         equ      1           ; clock output
sel         equ      2           ; cycles, yellow(0) or red(1)
;mclr       equ      3           ; system reset
le          equ      4           ; latch data
mode        equ      5           ; normal(1) or blink(0)
;
org         0x00
goto        start
;
start       bcf      status,rp0    ; stay in bank0
            clrf     intcon         ; no interrupts
            movlw    0xc0
            movwf    tlcon          ; int osc
            clrf     gpio           ; clear outputs
            bsf      status,rp0     ; goto bank1
            movlw    0x03
            movwf    pcon           ; all digital i/o's
            clrf     ansel          ; disable weak pull-ups
            movlw    0x2c
            movwf    trisio         ; set digital i/o's
            movlw    0x88
            movwf    option_reg     ; neg edge int, no timers
            bcf      status,rp0     ; goto bank0
;   clear i/o lines
            bcf      gpio,sdi
            nop
            bcf      gpio,clk
            nop
            bcf      gpio,le
;   normal pattern: G-Y-R... G-Y-R
drive       btfsc    gpio,sel       ; check cycles, 0=4, 1=8
            goto     lite_g
            movlw    0x04
            movwf    tcount
            goto     g_lite
lite_g      movlw    0x08
            movwf    tcount         ; 8 cycles back and forth
g_lite      call     green_r

```

```

    call    wait
    call    green_l
    call    wait                ; brief delay..
    btfss   gpio,mode          ; check mode, normal(1), flash(0)
    goto    blink              ; flash, go blink
    decfsz  tcount,f           ; normal
    goto    g_lite
    call    clear              ; clear display
;
    btfsc   gpio,sel
    goto    lite_y
    movlw   0x02                ; 2 cycles
    movwf   tcount
    goto    y_lite
lite_y     movlw   0x04
y_lite     movwf   tcount        ; 4 cycles back and forth
    call    yellow_r
    call    wait
    call    yellow_l
    call    wait
    btfss   gpio,mode          ; check mode, normal(1), flash(0)
    goto    blink              ; flash, go blink
    decfsz  tcount,f           ; normal
    goto    y_lite
    call    clear              ; clear display
;
    btfsc   gpio,sel
    goto    lite_r
    movlw   0x04
    movwf   tcount
    goto    r_lite
lite_r     movlw   0x08
r_lite     movwf   tcount        ; 8 cycles back and forth
    call    red_r
    call    wait
    call    red_l
    call    wait
    btfss   gpio,mode          ; check mode, normal(1), flash(0)
    goto    blink              ; flash, go blink
    decfsz  tcount,f           ; normal
    goto    r_lite
    call    clear
;
blink      btfsc   gpio,mode          ; normal or flash?
            goto    drive              ; normal, go back to drive
r_or_y     btfsc   gpio,sel          ; flash, yellow(0) or red(1)?
            goto    r_blink            ; red..
; blink yellow
y_blink    call    yellow_r
            call    wait
            btfss   gpio,sel          ; speed: slow(0) or fast(1)
            call    wait
            call    yellow_l
            call    wait
            btfss   gpio,sel
            call    wait
            call    clear
            call    wait
            goto    blink
; blink red
r_blink    call    red_r
            call    wait
            btfss   gpio,sel          ; speed: slow(0) or fast(1)
            call    wait
            call    red_l
            call    wait
            btfss   gpio,sel
            call    wait
            call    clear
            call    wait
            goto    blink
            ; check pattern again...

```

```

;   color scrolls
green_l   call    clock
          call    lvl1
          call    clear5
          call    clear5
          call    latch
          call    wait10
          call    clock
          call    lvl2
          call    clear5
          call    clear5
          call    latch
          call    wait10
          call    clock
          call    lvl3
          call    clear5
          call    clear5
          call    latch
          call    wait10
          call    clock
          call    lvl4
          call    clear5
          call    clear5
          call    latch
          call    wait10
          call    clock
          call    lvl5
          call    clear5
          call    clear5
          call    latch
          call    wait10
          call    clear
          call    wait
          return
green_r   call    clock
          call    lvl6
          call    clear5
          call    clear5
          call    latch
          call    wait10
          call    clock
          call    lvl7
          call    clear5
          call    clear5
          call    latch
          call    wait10
          call    clock
          call    lvl8
          call    clear5
          call    clear5
          call    latch
          call    wait10
          call    clock
          call    lvl9
          call    clear5
          call    clear5
          call    latch
          call    wait10
          call    clock
          call    lvl5
          call    clear5
          call    clear5
          call    latch
          call    wait10
          call    clear
          call    wait
          return
;
yellow_l  call    clock
          call    clear5
          call    lvl1

```

```

call    clear5
call    latch
call    wait10
call    clock
call    clear5
call    lvl2
call    clear5
call    latch
call    wait10
call    clock
call    clear5
call    lvl3
call    clear5
call    latch
call    wait10
call    clock
call    clear5
call    lvl4
call    clear5
call    latch
call    wait10
call    clock
call    clear5
call    lvl5
call    clear5
call    latch
call    wait10
call    clear
call    wait
return
yellow_r call    clock
call    clear5
call    lvl6
call    clear5
call    latch
call    wait10
call    clock
call    clear5
call    lvl7
call    clear5
call    latch
call    wait10
call    clock
call    clear5
call    lvl8
call    clear5
call    latch
call    wait10
call    clock
call    clear5
call    lvl9
call    clear5
call    latch
call    wait10
call    clock
call    clear5
call    lvl5
call    clear5
call    latch
call    wait10
call    clear
call    wait
return
;
red_l    call    clock
call    clear5
call    clear5
call    lvl1
call    latch
call    wait10
call    clock

```

```

    call    clear5
    call    clear5
    call    lv12
    call    latch
    call    wait10
    call    clock
    call    clear5
    call    clear5
    call    lv13
    call    latch
    call    wait10
    call    clock
    call    clear5
    call    clear5
    call    lv14
    call    latch
    call    wait10
    call    clock
    call    clear5
    call    clear5
    call    lv15
    call    latch
    call    wait10
    call    clear
    call    wait
    return
red_r    call    clock
        call    clear5
        call    clear5
        call    lv16
        call    latch
        call    wait10
        call    clock
        call    clear5
        call    clear5
        call    lv17
        call    latch
        call    wait10
        call    clock
        call    clear5
        call    clear5
        call    lv18
        call    latch
        call    wait10
        call    clock
        call    clear5
        call    clear5
        call    lv19
        call    latch
        call    wait10
        call    clock
        call    clear5
        call    clear5
        call    lv15
        call    latch
        call    wait10
        call    clear
        call    wait
        return
;   clear all led's
clear    call    clock
        movlw  0x00
        call    xmit
        movlw  0x00
        call    xmit
        movlw  0x00
        call    xmit
        call    latch
        return
;   clear single led
clear5    movlw  0x00

```

```

        call    xmit
        return
;   led levels, 1-5 (forward)
lv11    movlw   0x10
        call    xmit
        return
lv12    movlw   0x18
        call    xmit
        return
lv13    movlw   0x1c
        call    xmit
        return
lv14    movlw   0x1e
        call    xmit
        return
lv15    movlw   0x1f
        call    xmit
        return
;   1-5 (reverse)
lv16    movlw   0x01
        call    xmit
        return
lv17    movlw   0x03
        call    xmit
        return
lv18    movlw   0x07
        call    xmit
        return
lv19    movlw   0x0f
        call    xmit
        return
;   toggle clock
clock    bsf     gpio,clk
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        bcf     gpio,clk
        return
;   update driver
latch    bsf     gpio,le
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        bcf     gpio,le
        return
;   xmit 5 bits, msb first
xmit     movwf   temp0
        movlw   0x05           ; set counter to '8'
        movwf   count0
xmit1    bcf     status,c       ; clear carrier flag
        rrf     temp0,f         ; rotate out data bit
        btfss   status,c
        goto    xmit2
        bsf     gpio,sdi       ; c-bit is high, clock out a high bit
        goto    xmit3
xmit2    bcf     gpio,sdi       ; c-bit is low, clock out a low bit
xmit3    call    clock         ; toggle oled clock line

```

```

        decfsz count0,f      ; all bits sent out?
        goto xmit1          ; no, keep transmitting..
        return
; 0.11 second delay (11ms)
wait10  movlw 0xc0
        movwf counter2
repeat1 movlw 0xc0
        movwf counter1
        nop
loop1   decfsz counter1,f
        goto loop1
        decfsz counter2,f
        goto repeat1
        return
; 0.11 second delay or 20ms
wait    movlw 0xc0          ; 0xff, 20ms
        movwf counter2
repeat  movlw 0xc0          ; 0xff, 20ms
        movwf counter1
        nop
loop    decfsz counter1,f
        goto loop
        decfsz counter2,f
        goto repeat
        return
;
end

```