

EET250 Lab PIC Light Shows

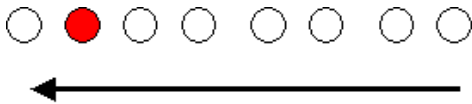
This lab demonstrates the use of Output subroutine and nested delay service to generate light show on Port D led set of the demo board.

Delay Service:

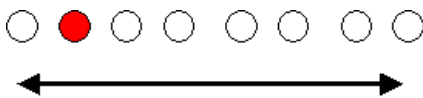
```
Delay:
    movwf    Delay2        ;
DelayLoop:
    decfsz  Delay1,f      ; Waste time.
    goto    DelayLoop     ; The Inner loop takes 3 instructions per loop * 256 loopss = 768 instructions
    decfsz  Delay2,f      ; The outer loop takes and additional 3 instructions per lap * 256 loops
    goto    DelayLoop     ; (768+3) * 256 = 197376 instructions / 1M instructions per second = 0.197 sec.
                                ; call it two-tenths of a second.
return
```

Figure 1 Delay Routine

Rotate left Leds Dynamic Pattern



Bounce Leds Dynamic Pattern



Spread Out Leds Dynamic Pattern



1.0 Program #1: Design for LED Rotate left Pattern

Use supplied program to demonstrate figure 1 pattern where single LED is continuously traveling right to left and then starting over. Enter and build program and program and debug using PICKIT2. Flowchart and demo to instructor.

Instructor Signoff _____

```

cblock 0x020
Delay1      ; Assign an address to label Delay1
Delay2
Display
endc

org 0x0000
RESET_V goto INIT      ;Reset Vector
INIT:
clrf PORTD      ;Clear PORTD output latches
bsf STATUS,RPO  ;Switch to bank 1
movlw b'00000000' ;Load value to make all bits output outputs
movwf TRISD     ;Move value to TRISD
bcf STATUS,RPO  ;Switch to bank 0
movlw 0x80
movwf Display

mainloop:
call Output
goto mainloop

Output:
movf Display,w      ; Copy the display to the LEDs
movwf PORTD
movlw .255
call Delay
bcf STATUS,C        ; ensure the carry bit is clear
rrf Display,f       ; Did the bit rotate into the carry?
btsc STATUS,C       ; yes, put it into bit 7.
bsf Display,7
return

Delay:
movwf Delay2        ;

DelayLoop:
decfsz Delay1,f    ; Waste time.
goto DelayLoop     ; The Inner loop takes 3 instructions per loop * 256 loops = 768 instructions
decfsz Delay2,f    ; The outer loop takes and additional 3 instructions per lap * 256 loops
goto DelayLoop     ; (768+3) * 256 = 197376 instructions / 1M instructions per second = 0.197 sec.
; call it two-tenths of a second.

return
END

```

Figure 2 program 1 code

2.0 Simulate and use Stop Watch/Breakpoints to Time Delay Subroutine

Select debugger for MPLAB simulator. Set break points as shown and use STOP WATCH to time Delay Subroutine. Does it match the calculations of 1 MHZ instruction T cycle count contained within the comments?

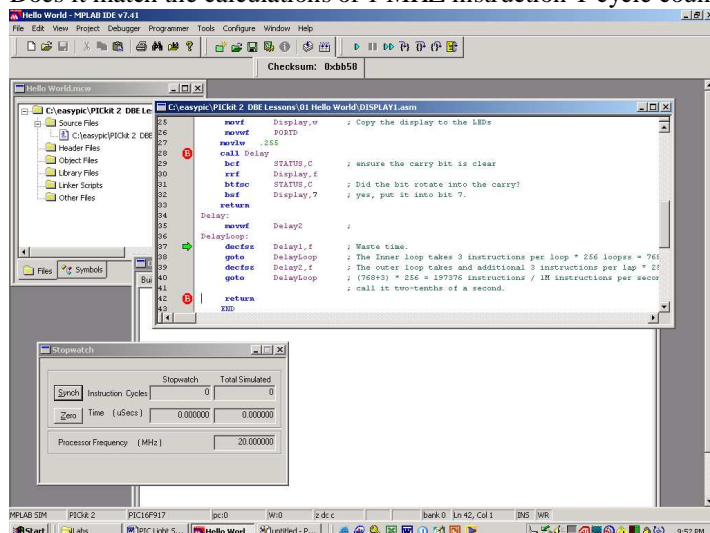


Figure 3 Using Simulated Stop Watch

3.0 Program #2: Design for Bounce LED Pattern

Follow indicated additions and use provided program template to enact figure 2 pattern where single LED is continuous traveling right to left and then left to right. Flowchart and demo to instructor.

Instructor Signoff _____

```

;add variables Delay1, Delay2, Display, Direction
org 0x0000
RESET_V goto INIT ;Reset Vector
INIT:
    movlw 0x01
    movwf Direction
    clrf PORTD ;Clear PORTD output latches
    bsf STATUS,RPO ;Switch to bank 1
    movlw b'00000000' ;Load value to make all bits output output
    movwf TRISD ;Move value to TRISD
    bcf STATUS,RPO ;Switch to bank 0
    movlw 0x80
    movwf Display
mainloop:
    call Output
    goto mainloop
Output:
    movf Display,w ; Copy the display to the LEDs
    movwf PORTD
    movlw .255
    call Delay
    bcf STATUS,C ; ensure the carry bit is clear
    btfsc Direction,0 ; if clear rotate right
    goto rotateLeft ;else rotate left
rotateright: rrf Display,f
    btfss STATUS,C ; Did the bit rotate into the carry?
    return
    bcf STATUS,C
    bsf Display,0
    bsf Direction,0
    return
rotateLeft:
    rlf Display,f
    btfss STATUS,C ; Did the bit rotate into the carry?
    return
    bcf STATUS,C
    bsf Display,7
    bcf Direction,0
    return
: add Delay Subroutine here

```

4.0 Program #3: Design fo rSpread out LED Pattern

Write a program that emulates spread pattern where dual LED is continuously traveling originating at center position and traveling out radial before starting over again. Flowchart and demo to instructor.

Instructor Signoff _____

```

Start:    Movlw 0x18
          CALL OUTPUT
          Movlw 0x24
          CALL OUTPUT
          Movlw 0x42
          CALL OUTPUT
          Movlw 0x81H
          CALL OUTPUT
          Goto Start

```