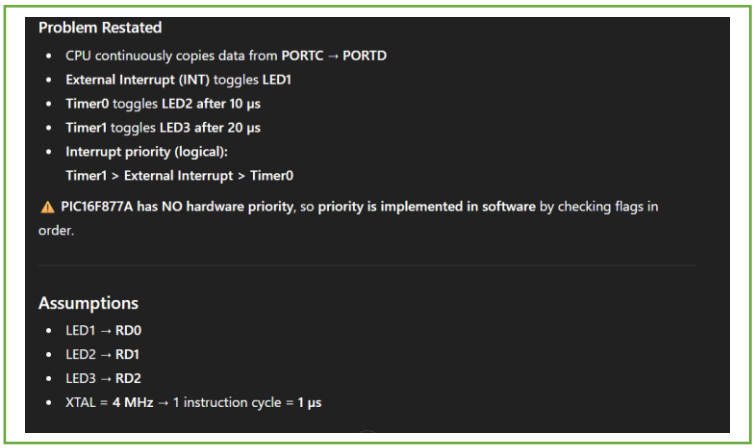


```
        LIST    P=16F877A
        #include <P16F877A.INC>

        ORG     0x00
        GOTO    MAIN

        ORG     0x04            ; Interrupt Vector
        GOTO    ISR

;---------------------------------
MAIN
        BSF     STATUS, RP0
        CLRF    TRISC           ; PORTC input
        CLRF    TRISD           ; PORTD output
        BCF     STATUS, RP0

        CLRF    PORTD

; Enable Interrupts
        BSF     INTCON, GIE
        BSF     INTCON, INTE     ; External INT
        BSF     INTCON, T0IE     ; Timer0 INT
        BSF     PIE1, TMR1IE     ; Timer1 INT

; Timer0 setup
        MOVLW   .246
        MOVWF   TMR0

; Timer1 setup
        BSF     STATUS, RP0
        MOVLW   0xFF
        MOVWF   TMR1H
        MOVLW   0xEC
        MOVWF   TMR1L
        BCF     STATUS, RP0
        BSF     T1CON, TMR1ON

;---------------------------------
LOOP
        MOVF    PORTC, W
        MOVWF   PORTD
        GOTO    LOOP

;---------------------------------
ISR
; Priority order implemented by checking flags
```

```
; ---- Timer1 (Highest) ----
        BTFSS   PIR1, TMR1IF
        GOTO    EXT_INT
        BCF     PIR1, TMR1IF
        BTG     PORTD, 2        ; Toggle LED3

        MOVLW   0xFF
        MOVWF   TMR1H
        MOVLW   0xEC
        MOVWF   TMR1L
        RETFIE

; ---- External Interrupt ----
EXT_INT
        BTFSS   INTCON, INTF
        GOTO    TIMER0_INT
        BCF     INTCON, INTF
        BTG     PORTD, 0        ; Toggle LED1
        RETFIE

; ---- Timer0 (Lowest) ----
TIMER0_INT
        BTFSS   INTCON, T0IF
        RETFIE
        BCF     INTCON, T0IF
        BTG     PORTD, 1        ; Toggle LED2
        MOVLW   .246
        MOVWF   TMR0
        RETFIE

    END
```

(A) Toggle PORTD 700 times

Logic

- Each toggle flips PORTD
- Loop counter = **700 = 7 × 100**
- Use nested loops

```
        LIST    P=16F877A
        #include <P16F877A.INC>

        ORG     0x00

        BSF     STATUS, RP0
        CLRF    TRISD
        BCF     STATUS, RP0

        MOVLW   D'7'
        MOVWF   COUNT1
```

```
LOOP1
        MOVLW   D'100'
        MOVWF   COUNT2

LOOP2
        COMF    PORTD, F
        CALL    DELAY
        DECFSZ  COUNT2, F
        GOTO    LOOP2

        DECFSZ  COUNT1, F
        GOTO    LOOP1

        END

DELAY
        MOVLW   D'250'
        MOVWF   TEMP
D1      DECFSZ  TEMP, F
        GOTO    D1
        RETURN
```

Requirements

Light	Time
Red	5 sec
Yellow	2 sec
Green	5 sec

Pedestrian Button Freeze RED for 10 sec

Pin Assignment

- Red → RD0
- Yellow → RD1
- Green → RD2
- Pedestrian Button → RB0

```
        LIST    P=16F877A
        #include <P16F877A.INC>

        ORG     0x00

        BSF     STATUS, RP0
        CLRF    TRISD
```

```
        BSF     TRISB, 0
        BCF     STATUS, RP0

START
; RED
        BSF     PORTD,0
        BCF     PORTD,1
        BCF     PORTD,2
        CALL    DELAY5

        BTFSC   PORTB,0
        CALL    PED_WAIT

; GREEN

        BCF     PORTD,0
        BCF     PORTD,1
        BSF     PORTD,2
        CALL    DELAY5

; YELLOW

        BCF     PORTD,2
        BSF     PORTD,1
        CALL    DELAY2

        GOTO    START

PED_WAIT

        CALL    DELAY10
        RETURN

;-------------------------
DELAY1S
        MOVLW   D'250'
        MOVWF   T1

D1      MOVLW   D'250'
        MOVWF   T2

D2      DECFSZ  T2,F
        GOTO    D2
        DECFSZ  T1,F
        GOTO    D1
        RETURN

DELAY5
        MOVLW   D'5'
        MOVWF   C1

L5      CALL    DELAY1S
        DECFSZ  C1,F
        GOTO    L5
        RETURN

DELAY2
        MOVLW   D'2'
        MOVWF   C1

L2      CALL    DELAY1S
        DECFSZ  C1,F
        GOTO    L2
        RETURN
```

```
DELAY10
        MOVLW   D'10'
        MOVWF   C1

L10     CALL    DELAY1S

        DECFSZ  C1,F
        GOTO    L10
        RETURN

        END
```