

Microcontroladores

Laboratorio

Semana 5

Profesor Kalun José Lau Gan

Semestre 2022-1

1

Preguntas previas

- Tengo un problema de error de sintaxis en la siguiente línea:

```
bcf INTCON, TMROIF ;Bajamos bandera de TMRO
```

- El XC8 PIC Assembler no contempla usar labels a los bits de los registros SFR como usualmente se hacía en MPASM, hay que reemplazarlo por la posición de bit:

```
bcf INTCON, 2 ;Bajada de bandera de TMRO
```

REGISTER 9-1: INTCON: INTERRUPT CONTROL REGISTER							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE/GIEH	PEIE/GIEL	TMROIE	INTOIE	RBIE	TMROIF	INTOIF	RBIF ⁽¹⁾
bit 7					bit 2		bit 0

- ¿?

2

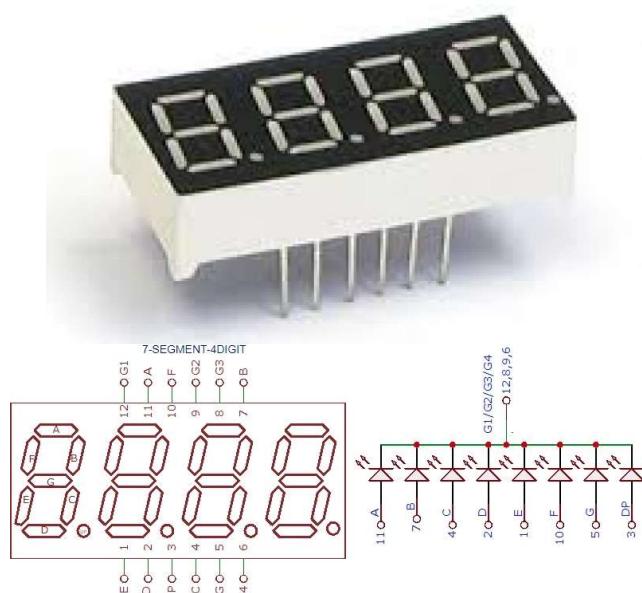
Agenda

- Multiplexación en displays de siete segmentos
- El Timer0 e interrupciones para el refresco de la visualización o el cambio de mensaje
- Multitarea

3

El display de cuatro dígitos de siete segmentos multiplexado:

- Verificar si el display es de cátodo común o ánodo común empleando el multímetro.
- Para la multiplexación se requerirá del uso de transistores para evitar que los pines del microcontrolador trabajen con corrientes excesivas.



4

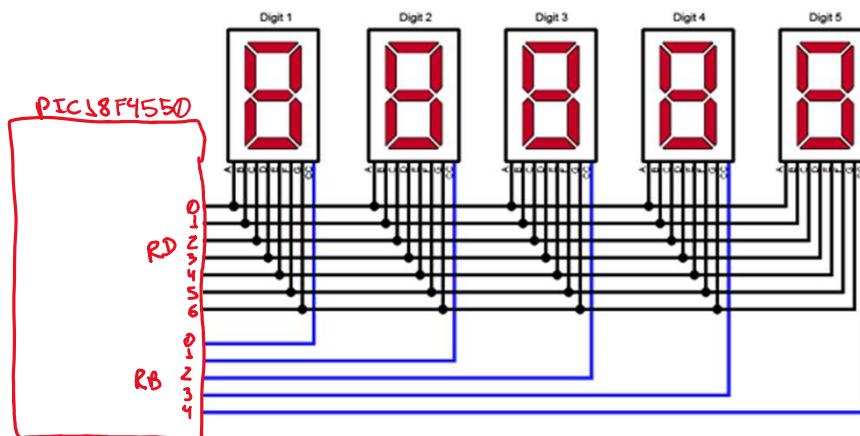
Multiplexación en displays de siete segmentos

- Los segmentos de cada dígito se encuentran conectados (Sa-Dig1 esta conectado con Sa-Dig2, con Sa-Dig3 y con Sa-Dig4, y demás segmentos)
- Se visualiza un dígito a la vez en el display multiplexado y en forma cíclica. Un ciclo de refresco involucra haber visualizado la información en todos los dígitos que componen el display.
- Reducimos la cantidad de E/S empleados en el microcontrolador (7 salidas para los segmentos excluyendo el punto decimal y 4 salidas para la selección del dígito)
- Por medio de un refresco a alta frecuencia (>50Hz) podremos ver los cuatro displays encendidos al mismo tiempo por el efecto de “persistencia visual”

5

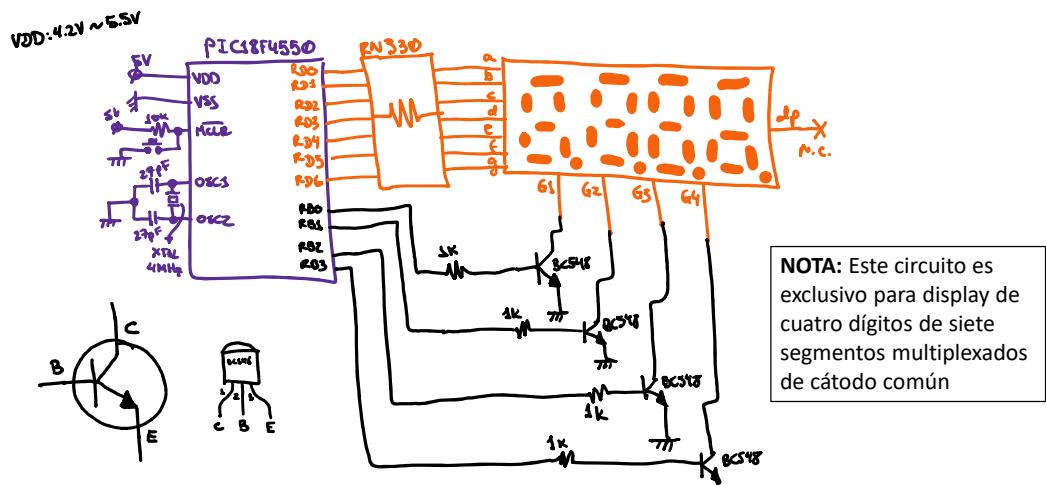
Caso: Malos diseños electrónicos en la multiplexación de displays de siete segmentos:

- Se recomienda no conectar los habilitadores de frente a los I/O del microcontrolador ya que dichos I/O no soportan entregar/recepicionar corrientes altas.



6

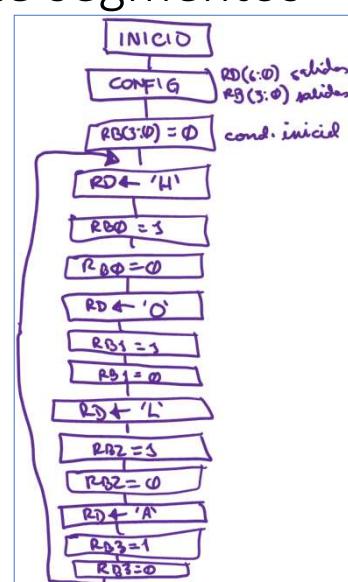
Ejemplo: Visualizar “HOLA” en el display multiplexado



7

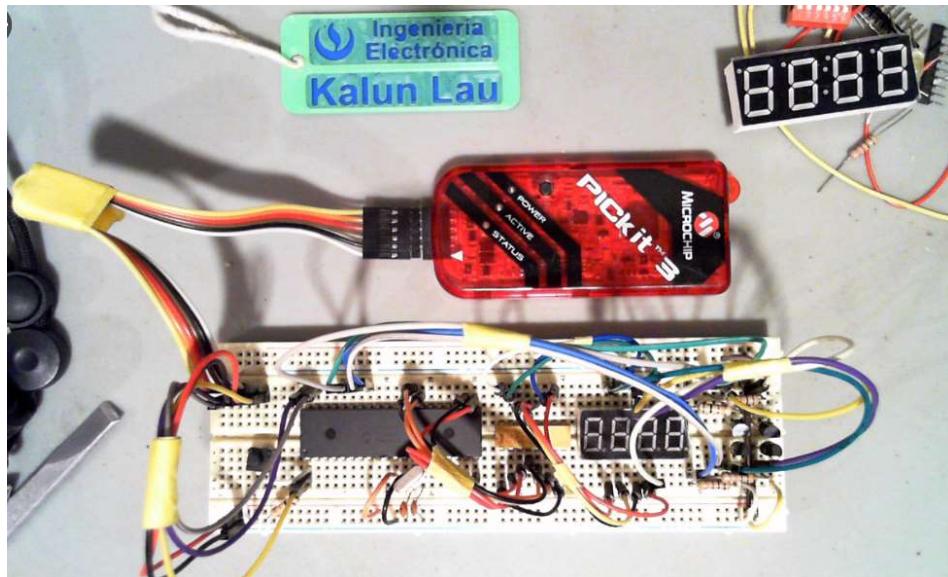
Algoritmo para la multiplexación de los dígitos del display de siete segmentos

- Se debe de enviar el dato cuando se encuentren deshabilitados los dígitos (para no tener “ghosting”)
 - Luego de enviar el dato se hará la habilitación del dígito respectivo, antes de cargar otro dato se deberá de deshabilitar el dígito activo.



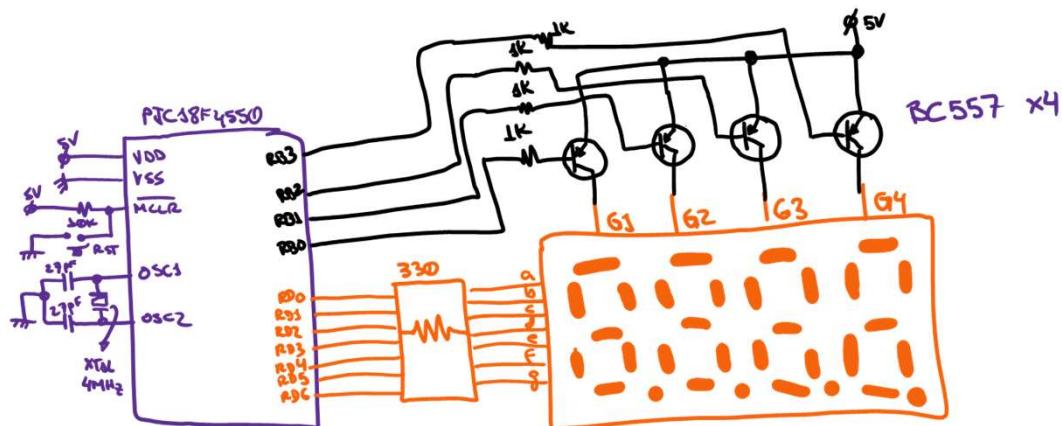
8

Circuito implementado con display de cuatro dígitos de siete segmentos cátodo común multiplexados



9

Circuito para display multiplexado de ánodo común



10

Código inicial en MPASM

```

14      org 0x0000          ;vector de reset
15      goto init_conf
16
17      org 0x0020          ;zona de programa de usuario
18      init_conf: movlw 0x80
19          movwf TRISD    ;RD(6:0) como salidas
20          movlw 0xF0
21          movwf TRISB    ;RB(3:0) como salidas
22          clrf LATB     ;RB en cero
23          movlw b'01110110' ;letra H
24          movwf LATD
25          bsf LATB, 0
26          call nop_group
27          bcf LATB, 0
28          movlw b'00111111' ;letra O
29          movwf LATD
30          bsf LATB, 1
31          call nop_group
32          bcf LATB, 1
33          movlw b'00111100' ;letra L
34          movwf LATD
35          bsf LATB, 2
36          call nop_group
37          bcf LATB, 2
38          movlw b'01110111' ;letra A
39          movwf LATD
40          bsf LATB, 3
41          call nop_group
42          bcf LATB, 3
43          goto loop
44
45      nop_group: nop
46          nop
47          nop
48          nop
49          return
50
51      end

```

11

Código en XC8 PIC Assembler

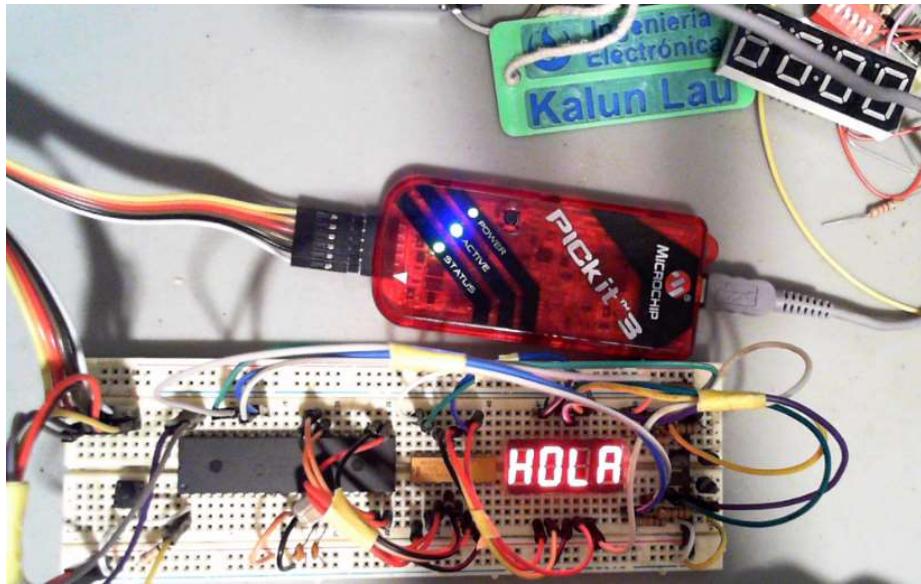
```

4      PROCESSOR 18f4550
5      #include "cabecera.inc"
6
7      PSECT multiplexacion,class=CODE,reloc=2,abs
8
9      ORG 00000H
10     multiplexacion: goto configuracion
11
12     ORG 00020H
13     configuracion:
14         movlw 80H
15         movwf TRISD    ;RD6-RD0 como salidas
16         movlw OFOH
17         movwf TRISB    ;RB3-RB0 como salidas
18         clrf LATB     ;Condicion inicial de los habilitadores
19
20         movlw 01110110B ;Letra H
21         movwf LATD
22         bsf LATB, 0
23         nop
24         bcf LATB, 0
25         movlw 00111111B ;Letra O
26         movwf LATD
27         bsf LATB, 1
28         nop
29         bcf LATB, 1
30         movlw 00111100B ;Letra L
31         movwf LATD
32         bsf LATB, 2
33         nop
34         bcf LATB, 2
35         movlw 01110111B ;Letra A
36         movwf LATD
37         bsf LATB, 3
38         nop
39         bcf LATB, 3
40         goto inicio
41     end multiplexacion

```

12

Pruebas en circuito:



13

Código mejorado empleando TBLPTR (MPASM)

```

1  ;Este es un comentario
2  list p=18f4550
3  #include <pl18f4550.inc>      ;libreria de nombre
4
5  CONFIG FOSC = INTOSCIO_EC
6  CONFIG CPUDIV = OSC1_PLL2
7  CONFIG PWRT = ON             ; Power-up Timer E
8  CONFIG BOR = OFF             ; Brown-out Reset
9  CONFIG WDT = OFF             ; Watchdog Timer E
10 CONFIG PBADEN = OFF           ; PORTB A/D Enable
11 CONFIG LVP = OFF              ; Single-Supply IC
12 CONFIG MC1RE = ON
13
14 org 0x0300
15 ;mensaje db 0x76, 0x3F, 0x38, 0x77 ;HOLA
16 ;mensaje db 0x78, 0x79, 0x38, 0x77 ;tela
17 mensaje db b'01111100', b'00000110', b'00111001', 0x
18
19 org 0x0000                  ;vector de reset
20 goto init_conf
21
22 org 0x0020                  ;zona de programa de
23 init_conf: movlw 0x80
24         movwf TRISD ;RD(6:0) como salidas
25         movlw 0xF0
26         movwf TRISB ;RB(3:0) como salidas
27         clrf LATB ;RB en cero
28         movlw HIGH mensaje
29         movwf TBLPTRH
30         movlw LOW mensaje
31         movwf TBLPTRL
32
33         loop:    clrf TBLPTRL
34             TBLRD+*
35             movff TABLAT, LATD
36             bsf LATB, 0
37             call nop_group
38             bcf LATB, 0
39             TBLRD+*
40             movff TABLAT, LATD
41             bsf LATB, 1
42             call nop_group
43             bcf LATB, 1
44             TBLRD+*
45             movff TABLAT, LATD
46             bsf LATB, 2
47             call nop_group
48             bcf LATB, 2
49             TBLRD+*
50             movff TABLAT, LATD
51             bsf LATB, 3
52             call nop_group
53             bcf LATB, 3
54             goto loop
55
56         nop_group:  nop
57             nop
58             nop
59             return
60
61         end
62

```

Se ha modificado para que el programa obtenga los datos a visualizar desde la memoria de programa empleando el TBLPTR

14

Código mejorado empleando TBLPTR (XC8 PIC Assembler)

```
4      PROCESSOR 18f4550
5      #include "cabecera.inc"
6
7      PSECT multiplexacion,class=CODE,reloc=2,abs
8
9      ORG 00300H
10     H      O      L      A
11     mensaje: db    76H, 3FH, 38H, 77H
12
13     ORG 00000H
14     multiplexacion: goto configuracion
15
16     ORG 00020H
17     configuracion:
18
19     movlw 80H
20
21     movwf TRISD ;RD6-RD0 como salidas
22     movlw 0FOH
23
24     movwf TRISB ;RB3-RB0 como salidas
25     clrf LATB ;Condicion inicial de los habilitadores
26
27     movlw HIGH mensaje
28
29     movwf TBLPTRH
30     movlw LOW mensaje
31
32     movwf TBLPTRL ;TBLPTR apuntando a mensaje (300H)
```

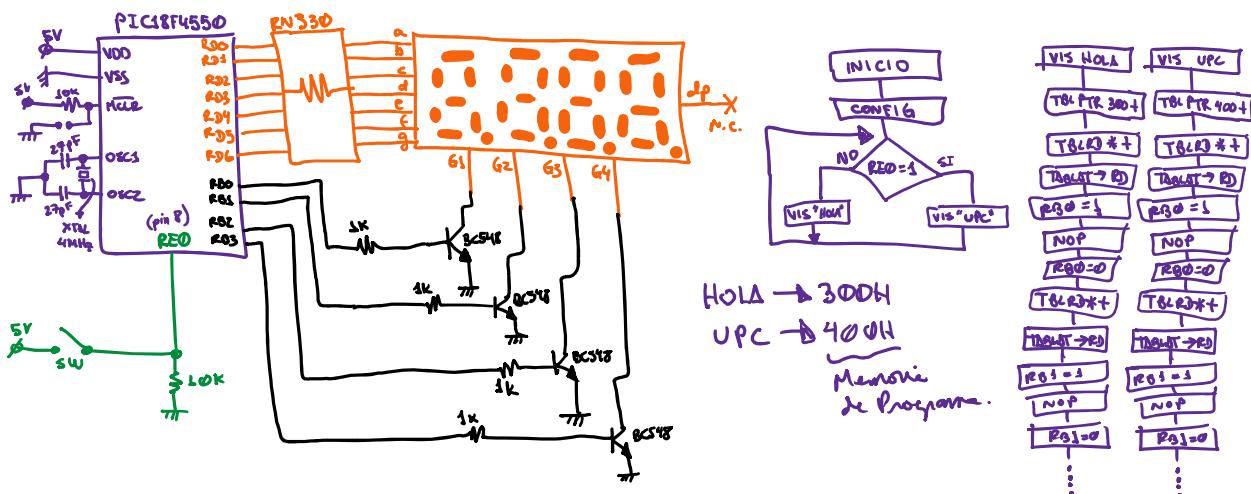
```

28 inicio:
29     TBLRD*+
30     movff TABLAT, LATD
31     bsf LATB, 0
32     nop
33     bcf LATB, 0
34     TBLRD*+
35     movff TABLAT, LATD
36     bsf LATB, 1
37     nop
38     bcf LATB, 1
39     TBLRD*+
40     movff TABLAT, LATD
41     bsf LATB, 2
42     nop
43     bcf LATB, 2
44     TBLRD*
45     movff TABLAT, LATD
46     bsf LATB, 3
47     nop
48     bcf LATB, 3
49     clrf TBLPTRL
50     goto inicio
51 end multiplexacion

```

15

Modificando el ejemplo para que pueda intercambiar el mensaje visualizado con la entrada REO: 0=HOLA, 1=UPC



16

Modificando el ejemplo para que pueda intercambiar el mensaje visualizado con la entrada RE0: 0=HOLA, 1=UPC

<pre> 4 PROCESSOR 18f4550 5 #include "cabecera.inc" 6 7 PSECT multiplexacion,class=CODE,reloc=2,abs 8 9 ORG 00300H 10 ; H O L A 11 mensaje1: db 76H, 3FH, 38H, 77H 12 ORG 00400H 13 ; U P C 14 mensaje2: db 00H, 3EH, 73H, 39H 15 16 ORG 00000H 17 multiplexacion: goto configuracion 18 19 ORG 00020H 20 configuracion: 21 movlw 0x80 22 movwf TRISD ;RD6-RD0 como salidas 23 movlw 0xF0 24 movwf TRISB ;RB3-RB0 como salidas 25 clrf LATB ;Condicion inicial de los habilitadores 26 movlw 0x0F 27 movwf ADCON1 ;Para que RE0 sea entrada digital 28 29 inicio: 30 btfss PORTE, 0 ;Pregunta si RE0=1 31 goto msg_hola 32 goto msg_upc 33 34 msg_hola: 35 movlw 03H 36 movwf TBLPTRH 37 clrf TBLPTRL 38 goto multiplex </pre>	<pre> 40 msg_upc: 41 movlw 04H 42 movwf TBLPTRH 43 clrf TBLPTRL 44 goto multiplex 45 46 multiplex: 47 TBLRD*+ 48 movwf TABLAT, LATD 49 bsf LATB, 0 50 nop 51 bcf LATB, 0 52 TBLRD*+ 53 movwf TABLAT, LATD 54 bsf LATB, 1 55 nop 56 bcf LATB, 1 57 TBLRD*+ 58 movwf TABLAT, LATD 59 bsf LATB, 2 60 nop 61 bcf LATB, 2 62 TBLRD*+ 63 movwf TABLAT, LATD 64 bsf LATB, 3 65 nop 66 bcf LATB, 3 67 goto inicio 68 end multiplexacion </pre>
--	--

17

CASO: Cambiar mensajes en el display multiplexado en forma automática

- Para poder cambiar el mensaje de visualización de manera secuencial es necesario delegar la **función de refresco** (la multiplexación) o la **función de cambio de mensaje** a un módulo de temporización, en este caso el Timer0. Por lo tanto tendremos dos estrategias que explicaremos a continuación.

18

Estrategia1: Timer0 como elemento de refresco del display

- Al optar que el Timer0 sea el elemento que haga el proceso de refresco en el display multiplexado, éste debe de temporizar lo suficientemente rápido para que aparezca el efecto de "persistencia visual", a partir de una frecuencia de 50Hz el ojo humano ya le es difícil percibir un parpadeo o refresco en un display ó pantalla.
- Para el siguiente ejemplo se mostrarán las palabras "Casa" y "Perú".
- Se considerará un refresco de 120Hz ó 8.333ms. Hay que tener en consideración dicho periodo para hacer la multiplexación de manera correcta a fin de que no perdamos intensidad luminosa en el display. Para ello se tiene que colocar un retardo de entre 1ms y 2ms luego de habilitar cada dígito.
- Si el Timer0 se encarga del refresco del display, la rutina principal del microcontrolador será el que haga el cambio de mensaje.
- Se está activando la interrupción por desborde de Timer0 para mejorar el desempeño de la solución.

19

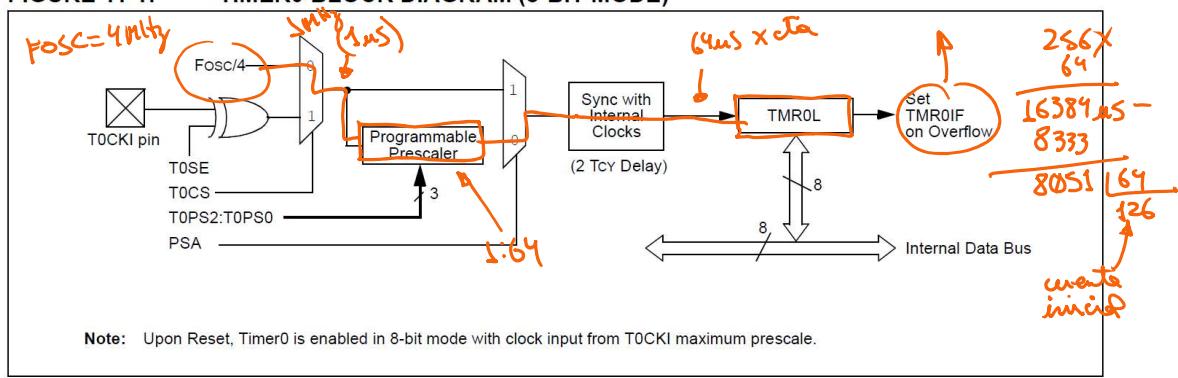
Análisis de Timer0 para refresco en 120Hz y su interrupción por desborde

$$f = \frac{1}{T}$$

$$T = \frac{1}{120\text{Hz}}$$

$$T = 8.33\text{ms}$$

FIGURE 11-1: TIMER0 BLOCK DIAGRAM (8-BIT MODE)



20

(cont.)

REGISTER 11-1: TOCON: TIMER0 CONTROL REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TMR0ON	TOB8T	TOCS	TOSE	PSA	TOPS2	TOPS0	TAPS0

bit 7 bit 0

Legend:

R = Readable bit
-n = Value at PORW = Writable bit
'1' = Bit is setU = Unimplemented bit, read as '0'
'0' = Bit is cleared
x = Bit is unknown

bit 7	TMR0ON: Timer0 On/Off Control bit 1 = Enables Timer0 0 = Stops Timer0
bit 6	TOB8T: Timer0 8-Bit/16-Bit Control bit 1 = Timer0 is configured as an 8-bit timer/counter 0 = Timer0 is configured as a 16-bit timer/counter
bit 5	TOCS: Timer0 Clock Source Select bit 1 = Transition on TOCKI pin 0 = Internal instruction cycle clock (CLKO)
bit 4	TOSE: Timer0 Source Edge Select bit 1 = Increment on high-to-low transition on TOCKI pin 0 = Increment on low-to-high transition on TOCKI pin
bit 3	PSA: Timer0 Prescaler Assignment bit 1 = Timer0 prescaler is NOT assigned. Timer0 clock input bypasses prescaler. 0 = Timer0 prescaler is assigned. Timer0 clock input comes from prescaler output.
bit 2-0	TOPS2:TOPS0: Timer0 Prescaler Select bits 111 = 1:256 Prescale value 110 = 1:128 Prescale value 101 = 1:64 Prescale value 100 = 1:32 Prescale value 011 = 1:16 Prescale value 010 = 1:8 Prescale value 001 = 1:4 Prescale value 000 = 1:2 Prescale value

TOCON: 0xC5

21

(cont...)

REGISTER 9-1: INTCON: INTERRUPT CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE/GIEH	PEIE	SIEL	TMROIE	INT0E	RBIE	TMROIF	INT0F

bit 7 bit 0

Legend:

R = Readable bit
-n = Value at PORW = Writable bit
'1' = Bit is setU = Unimplemented bit, read as '0'
'0' = Bit is cleared
x = Bit is unknown

bit 7	GIE/GIEH: Global Interrupt Enable bit When IPEN = 0: 1 = Enables all unmasked interrupts 0 = Disables all interrupts When IPEN = 1: 1 = Enables all high-priority interrupts 0 = Disables all interrupts
bit 6	PEIE/GIEL: Peripheral Interrupt Enable bit When IPEN = 0: 1 = Enables all unmasked peripheral interrupts 0 = Disables all peripheral interrupts When IPEN = 1: 1 = Enables all low-priority peripheral interrupts (if GIE/GIEH = 1) 0 = Disables all low-priority peripheral interrupts
bit 5	TMROIE: TMRO Overflow Interrupt Enable bit 1 = Enables the TMRO overflow interrupt 0 = Disables the TMRO overflow interrupt
bit 4	INT0IE: INT0 External Interrupt Enable bit 1 = Enables the INT0 external interrupt 0 = Disables the INT0 external interrupt
bit 3	RBIE: RB Port Change Interrupt Enable bit 1 = Enables the RB port change interrupt 0 = Disables the RB port change interrupt
bit 2	TMROIF: TMRO Overflow Interrupt Flag bit 1 = TMRO register has overflowed (must be cleared in software) 0 = TMRO register did not overflow
bit 1	INT0IF: INT0 External Interrupt Flag bit 1 = The INT0 external interrupt occurred (must be cleared in software) 0 = The INT0 external interrupt did not occur
bit 0	RBIF: RB Port Change Interrupt Flag bit ⁽¹⁾ 1 = At least one of the RB7:RB4 pins changed state (must be cleared in software) 0 = None of the RB7:RB4 pins have changed state

INTCON = 0xA0

22

Código en MPASM

```

1 ;Este es un comentario, se le antecede un punto
2 list p=18f4550 ;Modelo del microcontrolador
3 #include <p18f4550.inc> ;Llamada a la biblioteca
4
5 ;Directivas de preprocesador o bits de configuración
6 CONFIG PLDIV = 1 ; PLL Prescaler
7 CONFIG CPUDIV = OSC1_PLL2 ; System Clock
8 CONFIG FOSC = XT_XT ; Oscillator
9 CONFIG PWRT = ON ; Power-up Timer
10 CONFIG BOR = OFF ; Brown-out
11 CONFIG WDT = OFF ; Watchdog
12 CONFIG CCP2MX = ON ; CCP2 MUX
13 CONFIG PBADEN = OFF ; PORTB A/D
14 CONFIG MCLRE = ON ; MCLR Pin
15 CONFIG LVP = OFF ; Single-Supply
16
17 cblock 0x0000
18 var_lms
19 var_i
20 var_j
21 var_k
22 endc
23
24 org 0x0500
25 mensaje_1 db 0x39, 0x77, 0x6D, 0x77 ;CASA
26
27 org 0x0600
28 mensaje_2 db 0x73, 0x79, 0x50, 0x3E ;PERÚ
29
30 org 0x0000 ;vector de reset
31 goto init_conf
32
33 org 0x0008 ;vector de interrupciones
34 goto Timer0_ISR
35
36 org 0x0020 ;zona de programa de usuario
37 init_conf: movwf 0x80
38         movwf TRISD ;RD(6:0) son salidas
39         movlw 0xF0
40         movwf TRISE ;RB(3:0) son salidas
41         clrf LATB ;Los cuatro habilitados
42         movlw 0xC5
43         movwf TOCON ;TMRO on, fosc/4, prescaler
44         movlw 0xA0
45         movwf INTCON ;GIE=1, TMROIE=1 (interrupt)
46         movlw .126
47         movwf THROL ;carga de cuenta inicial
48
49 loop:   movlw 0x05
50         movwf TBLPTRH
51         call delay_long
52         movlw 0x06
53         movwf TBLPTRH
54         call delay_long
55         goto loop
56
57 Timer0_ISR: clrf TBLPTRL
58         TBLRD+
59         movff TABLAT, LATD
60         bcf LATD, 0 ;digito0 encendido
61         call delay_lms
62         bcf LATD, 0 ;digito0 apagado
63         TBLRD+
64         movff TABLAT, LATD
65         bcf LATD, 1 ;digito1 encendido
66         call delay_lms
67         bcf LATD, 1 ;digito1 apagado
68         TBLRD+
69         movff TABLAT, LATD
70         bcf LATD, 2 ;digito2 encendido
71         call delay_lms
72         bcf LATD, 2 ;digito2 apagado
73         TBLRD+
74         movff TABLAT, LATD
75         bcf LATD, 3 ;digito3 encendido
76         call delay_lms
77         bcf LATD, 3 ;digito3 apagado
78         bcf INTCON, TMROIF ;bajamos la bandera
79         retfie
80
81 delay_lms: movlw 0x90
82         movwf var_lms
83 otro:   decfsz var_lms, f
84         goto aun_no
85         return
86 aun_no:  nop
87         nop
88         goto otro
89
90 delay_long:
91         movlw .100
92         movwf var_i
93 otro1:  call bucle1 ;Salto
94         decfsz var_i, f
95         goto otro1
96         return
97
98 bucle1:
99         movlw .50
100        movwf var_j
101        otro2:
102         nop
103         nop
104         call bucle2 ;Salto
105         decfsz var_j, f
106         goto otro2
107         return
108
109 bucle2:
110        movlw .50
111        movwf var_k
112
113 otro3:
114         nop
115         decfsz var_k, f
116         goto otro3
117         return
118         end

```

23

Estrategia2: Timer0 como elemento de cambio (ó desplazamiento) del mensaje

- Si en la rutina principal se encuentra el proceso de la multiplexación, configuramos el Timer0 para que cuando se desborde haga el proceso de cambio del mensaje que se esta visualizando en el display multiplexado.
- El cambio de mensaje puede ser total o puede hacerse el efecto de desplazamiento del mensaje en cualquier dirección según requerimiento.
- El periodo de temporizado del Timer0 para esta estrategia por lo general es largo (ej. 500ms), se empleará entonces el modo de 16 bits.

Ejemplo: Visualizar mensaje “Cosa bonita”

- Para este ejemplo se tendrá que dividir la visualización en tres partes: “Cosa”, “boni” y “ta ”.
- El proceso de la multiplexación estará ubicado en la rutina principal.
- El cambio de mensaje lo realizará cada vez que ocurra un desborde del Timer0, dicho desborde ocurrirá en aproximadamente 500ms.
- No se utilizará interrupciones por lo que se tendrá que preguntar cuando se levante la bandera TMROIF.

25

(cont... código inicial)

```

4      PROCESSOR 18f4550
5      #include "cabecera.inc"
6
7      PSECT multiplexacion,class=CODE,reloc=2,abs
8
9      ORG 00300H
10     ;          C   O   S   A
11     mensaje1: db 39H, 3FH, 6DH, 77H
12     ORG 00400H
13     ;          b   O   n   i
14     mensaje2: db 7CH, 3FH, 54H, 04H
15     ORG 00500H
16     ;          t   a
17     mensaje3: db 78H, 77H, 00H, 00H
18
19     ORG 00000H
20     multiplexacion: goto configuracion
21
22     ORG 00020H
23     configuracion:
24     movlw 80H
25     movwf TRI5D      ;RD6-RD0 como salidas
26     movwf OF0H
27     movwf TRISB      ;RB3-RB0 como salidas
28     clrf LATB        ;Condición inicial de los hab.
29     movlw LOW mensaje1
30     movwf TBLPTRL    ;TBLPTR apuntando a mensaje1
31     movlw 82H
32     movwf TOCON      ;Timer0 ON, 16bit, 1:8PSC, F0
33
34     inicio:
35     movlw HIGH mensaje1
36     movwf TBLPTRH
37     TBLRD*+
38     movff TABLAT, LATD
39     bsf LATB, 0
40     nop
41     bcf LATB, 0
42     TBLRD*+
43     movff TABLAT, LATD
44
45     bcf LATB, 1
46     nop
47     bcf LATB, 1
48     TBLRD*+
49     movff TABLAT, LATD
50     bcf LATB, 2
51     nop
52     bcf LATB, 2
53     movff TABLAT, LATD
54     bcf LATB, 3
55     nop
56     bcf LATB, 3
57     clrf TBLPTRL
58     btfs INTCON, 2      ;Pregunto si se desbordó Timer0
59     goto inicio2
60     bcf INTCON, 2
61
62     inicio2:
63     movlw HIGH mensaje2
64     movwf TBLPTRH
65     TBLRD*+
66     movff TABLAT, LATD
67     bcf LATB, 0
68     nop
69     bcf LATB, 0
70     TBLRD*+
71     movff TABLAT, LATD
72     bcf LATB, 1
73     nop
74     bcf LATB, 1
75     TBLRD*+
76     movff TABLAT, LATD
77     bcf LATB, 2
78     nop
79     bcf LATB, 2
80     TBLRD*+
81     movff TABLAT, LATD
82     bcf LATB, 3
83     nop
84
85     bcf LATB, 3
86     clrf TBLPTRL
87     btfs INTCON, 2      ;Pregunto si se desbordó Timer0
88     goto inicio3
89     bcf INTCON, 2
90
91     inicio3:
92     movlw HIGH mensaje3
93     movwf TBLPTRH
94     TBLRD*+
95     movff TABLAT, LATD
96     bcf LATB, 0
97     nop
98     bcf LATB, 0
99     bcf LATB, 1
100    nop
101    bcf LATB, 1
102    TBLRD*+
103    movff TABLAT, LATD
104    bcf LATB, 2
105    bcf LATB, 2
106    nop
107    bcf LATB, 2
108    TBLRD*+
109    movff TABLAT, LATD
110    bcf LATB, 3
111    nop
112    bcf LATB, 3
113    clrf TBLPTRL
114    btfs INTCON, 2      ;Pregunto si se desbordó Timer0
115    goto inicio3
116    bcf INTCON, 2
117    goto inicio
118
119    end multiplexacion

```

26

(cont... optimizando el código y quitando redundancias)

<pre> 4 PROCESSOR 18f4550 5 #include "cabecera.inc" 6 7 PSECT multiplexacion,class=CODE,reloc=2,abs 8 9 ORG 00300H 10 ; C O S A 11 mensaje1: db 39H, 3FH, 6DH, 77H 12 ORG 00400H 13 ; b o n i 14 mensaje2: db 7CH, 3FH, 54H, 04H 15 ORG 00500H 16 ; t a 17 mensaje3: db 78H, 77H, 00H, 00H 18 19 ORG 00000H 20 multiplexacion: goto configuracion 21 22 ORG 00020H 23 configuracion: 24 movlw 80H 25 movwf TRISD ;RD6-RD0 como salidas 26 movlw 0FOH 27 movwf TRISB ;RB3-RB0 como salidas 28 clrf LATB ;Condicion inicial de los habilitadores 29 movlw LOW mensaje1 30 movwf TBLPTRL ;TBLPTR apuntando a mensaje (300H) 31 movlw HIGH mensaje1 32 movwf TBLPTRH 33 movlw 82H 34 movwf TOCON ;Timer0 ON, 16bit, 1:8PSC, FOSC/4 35 </pre>	<pre> 36 inicio: 37 TBLRD*+ 38 movff TABLAT, LATD 39 bsf LATB, 0 40 nop 41 bcf LATB, 0 42 TBLRD*+ 43 movff TABLAT, LATD 44 bsf LATB, 1 45 nop 46 bcf LATB, 1 47 TBLRD*+ 48 movff TABLAT, LATD 49 bsf LATB, 2 50 nop 51 bcf LATB, 2 52 TBLRD* 53 movff TABLAT, LATD 54 bsf LATB, 3 55 nop 56 bcf LATB, 3 57 clrf TBLPTRL 58 btfss INTCON, 2 ;Pregunto si se desbordó Timer0 59 goto inicio 60 bcf INTCON, 2 61 movlw 05H 62 cpfsq TBLPTRH 63 goto noes 64 movlw 03H 65 mowwf TBLPTRH 66 goto inicio 67 noes: 68 incf TBLPTRH, 1 69 goto inicio 70 71 end multiplexacion </pre>
---	--

27

Ejemplo: Visualizar la palabra “ingeniería” en el display de 4 dígitos multiplexados

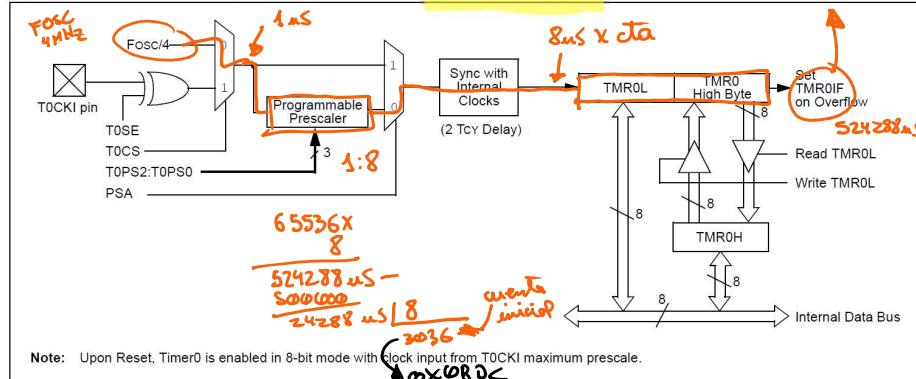
- Para visualizar la palabra “ingeniería” se debe de dividir en tres visualizaciones debido a que no se puede visualizar todas las letras a la vez. Se intentará visualizar “inge”, luego “nier” y finalmente “ia”.
- Se escogió que el Timer0 sea el mecanismo que cambie el mensaje visualizado a un periodo de 500ms mientras que la rutina principal seguirá haciendo la función del refresco.
- Se está considerando la activación de la interrupción por desborde del Timer0 para mejorar el desempeño.

28

Configuración del Timer0 para temporizar 500ms y activar su interrupción por desborde

- El Timer0 debe de trabajar a 16 bits

FIGURE 11-2: TIMER0 BLOCK DIAGRAM (16-BIT MODE)



29

(cont...)

Obtención de valor de configuración para el registro TOCON relacionado al Timer0

REGISTER 11-1: TOCON: TIMER0 CONTROL REGISTER							
R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TMRODN	T08BIT	TOCS	TOSE	PSA	TOPS2	TOPS1	TOPS0
bit 7							
bit 0							
Legend:							
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'					
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown				
bit 7	TMROON: Timer0 On/Off Control bit						
	1 = Enables Timer0 ✓	0 = Stops Timer0					
bit 6	T08BIT: Timer0 8-Bit/16-Bit Control bit						
	1 = Timer0 is configured as an 8-bit timer/counter ✓	0 = Timer0 is configured as a 16-bit timer/counter ✓					
bit 5	TOCS: Timer0 Clock Source Select bit						
	1 = Transition on T0CKI pin ✓	0 = Internal instruction cycle clock (CLKO) ✓					
bit 4	TOSE: Timer0 Source Edge Select bit						
	1 = Increment on high-to-low transition on T0CKI pin ✗	0 = Increment on low-to-high transition on T0CKI pin ✗					
bit 3	PSA: Timer0 Prescaler Assignment bit						
	1 = Timer0 prescaler is NOT assigned. Timer0 clock input bypasses prescaler. ✓	0 = Timer0 prescaler is assigned. Timer0 clock input comes from prescaler output. ✓					
bit 2-0	TOPS2:TOPSO: Timer0 Prescaler Select bits						
	111 = 1:256 Prescale value	110 = 1:128 Prescale value	101 = 1:64 Prescale value	100 = 1:32 Prescale value	011 = 1:16 Prescale value	010 = 1:8 Prescale value	001 = 1:4 Prescale value
	000 = 1:2 Prescale value						

30

(cont...)

Obtención de valor de configuración para el registro INTCON relacionado con las interrupciones.

REGISTER 9-1: INTCON: INTERRUPT CONTROL REGISTER							
	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
bit 7	GIE/GIEH	PEIE/SPIE	TMR0IE	NIE	RE	TR0IF	N0I
bit 0							
Legend:	R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'				
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown				
bit 7	GIE/GIEH: Global Interrupt Enable bit When IPEN = 0: 1 = Enables all unmasked interrupts ✓ 0 = Disables all interrupts When IPEN = 1: 1 = Enables all high-priority interrupts 0 = Disables all interrupts						
bit 6	PEIE/GIEL: Peripheral Interrupt Enable bit When IPEN = 0: 1 = Enables all unmasked peripheral interrupts 0 = Disables all peripheral interrupts When IPEN = 1: 1 = Enables all low-priority peripheral interrupts (if GIE/GIEH = 1) 0 = Disables all low-priority peripheral interrupts						
bit 5	TMR0IE: TMR0 Overflow Interrupt Enable bit 1 = Enables the TMR0 overflow interrupt ✓ 0 = Disables the TMR0 overflow interrupt						
bit 4	INT0IE: INT0 External Interrupt Enable bit 1 = Enables the INT0 external interrupt 0 = Disables the INT0 external interrupt						
bit 3	RBIE: RB Port Change Interrupt Enable bit 1 = Enables the RB port change interrupt 0 = Disables the RB port change interrupt						
bit 2	TMR0IF: TMR0 Overflow Interrupt Flag bit 1 = TMR0 register has overflowed (must be cleared in software) 0 = TMR0 register did not overflow						
bit 1	INT0IF: INT0 External Interrupt Flag bit 1 = The INT0 external interrupt occurred (must be cleared in software) 0 = The INT0 external interrupt did not occur						
bit 0	RBIF: RB Port Change Interrupt Flag bit! 1 = At least one of the RB7:RB4 pins changed state (must be cleared in software) 0 = None of the RB7:RB4 pins have changed state						

31

Código en MPASM

```

14      org 0x0300
15      mensaje db 0x06, 0x54, 0x6F, 0x79 ;INGE
16
17      org 0x0400
18      mensaje2 db 0x54, 0x06, 0x79, 0x50 ;NIER
19
20      org 0x0500
21      mensaje3 db 0x06, 0x77, 0x00, 0x00 ;IA
22
23      org 0x0000          ;vector de reset
24      goto init_conf
25
26      org 0x0008
27      goto Tmr0_ISR
28
29      org 0x0020          ;zona de programa de usuario
30
31 init_conf: movlw 0x80
32      movwf TRISD ;RD(6:0) como salidas
33      movlw 0xF0
34      movwf TRISB ;RB(3:0) como salidas
35      clrf LATB ;RB en cero
36      movlw HIGH mensaje
37      movwf TBLPTRH
38      movlw LOW mensaje
39      movwf TBLPTRL
40      movlw 0x81
41      movwf TOCON ;Tmr0 ON, fosc/4, psc 1:8, modo 16bit
42      movlw 0xA0
43      movwf INTCON ;GIE=1, TMROIE=1 interrupcion activo para
44
45 loop:    clrf TBLPTRL
46      TBLRD+*
47      movff TABLAT, LATD
48      bsf LATB, 0
49      call nop_group
50      bcf LATB, 0
51      TBLRD+*
52      movff TABLAT, LATD
53      bsf LATB, 1
54      call nop_group
55
56      bcf LATB, 1
57      TBLRD+*
58      movff TABLAT, LATD
59      bsf LATB, 2
60      call nop_group
61      bcf LATB, 2
62      TBLRD+*
63      movff TABLAT, LATD
64      bsf LATB, 3
65      call nop_group
66      bcf LATB, 3
67      goto loop
68
69      nop_group:    nop
70      nop
71      nop
72      nop
73      return
74
75      Tmr0_ISR:   movlw 0x0B
76      movwf TMROH
77      movlw 0xDC
78      movwf TMROL
79      movlw 0x05
80      cpfseq TBLPTRH ;Carga de cuenta inicial a TMRO
81
82      aunno:       movlw 0x03 ;Paso al primer mensaje
83      movwf TBLPTRH
84      goto otro
85      otro:        incf TBLPTRH, f ;Paso al siguiente mensaje
86      bcf INTCON, TMROIF
87      retfie
88      end

```

32

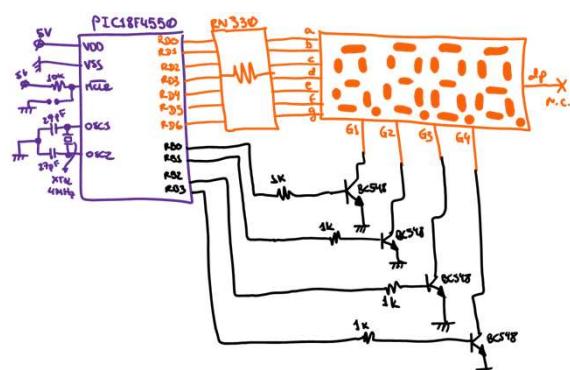
CASO: Efecto de desplazamiento del mensaje en el display multiplexado

- Se precisa de una sola tabla que contendrá las letras del mensaje a visualizar
- El efecto de desplazamiento de derecha a izquierda se logra al emplear una “posición guía” el cual indicará la primera de las cuatro letras de la tabla y que se visualizará en el display multiplexado de 4 dígitos.
- Al incrementar la “posición guía” cada cierto periodo obtendremos un efecto de desplazamiento de derecha a izquierda.

33

Ejemplo: Visualizar “HOLA UPC”

- Debido a que el mensaje a mostrar consta de mas de cuatro caracteres se debe de utilizar un efecto de desplazamiento a fin de que sea mostrado completamente.
- Se empleará el Timer0 para que dictamine el tiempo que demora en el desplazamiento de una letra a la vez de derecha a izquierda.



34

Código en XC8 PIC Assembler

```

4  PROCESSOR 18f4550
5  #include "cabecera.inc"
6
7  PSECT multiplexacion,class=CODE,reloc=2,abs
8
9  cuenta EQU 000H
10
11  ORG 00300H      ;Zona donde se alojaran las letras del mensaje
12  mensaje: db 00H, 00H, 00H, 00H, 76H, 3FH, 38H, 77H, 00H, 3EH, 73H, 39H, 00H, 00H, 00H
13
14  ORG 00000H      ;Vector de reset
15  multiplexacion: goto configuracion
16
17  ORG 00008H      ;Vector de interrupcion
18  vector_hp: goto TMRO_ISR
19
20  ORG 00020H
21  configuracion:
22  movlw 0x80
23  movwf TRISD      ;RD6-RD0 como salidas
24  movlw 0xF0
25  movwf TRISB      ;RB3-RB0 como salidas
26  clrf LATB        ;Condicion inicial de los habilitadores
27  movlw HIGH mensaje
28  movwf TBLPTRH
29  movlw LOW mensaje
30  movwf TBLPTRL     ;TBLPTR apunta a 300H
31  movlw 81H
32  movwf TOCON       ;Configuracion del TMRO: Fosc/4 16bit PSC1:4
33  movlw 0A0H
34  movwf INTCON      ;Configuracion de Ints: Desborde de TMRO
35  clrf cuenta

```

- El proceso de la multiplexación se encuentra en la rutina principal y el proceso de desplazamiento lo realiza la interrupción de desborde de Timer0

```

37  inicio:
38  movf cuenta, 0
39  addwf TBLPTRL
40  TBLRD+
41  movwf TABLAT, LATD
42  bsf LATB, 0
43  nop
44  bcf LATB, 0
45  TBLRD+
46  movwf TABLAT, LATD
47  bsf LATB, 1
48  nop
49  bcf LATB, 1
50  TBLRD+
51  movwf TABLAT, LATD
52  bsf LATB, 2
53  nop
54  bcf LATB, 2
55  TBLRD+
56  movwf TABLAT, LATD
57  bsf LATB, 3
58  nop
59  bcf LATB, 3
60  clrf TBLPTRL
61  goto inicio

```

63	TMRO_ISR:
64	movlw 11
65	cpfseq cuenta
66	goto incremento
67	clrf cuenta
68	goto otro
69	incremento:
70	incf cuenta, 1
71	otro:
72	movlw 0BH
73	movwf TMROH
74	movlw 0DCH
75	movwf TMROL
76	bcf INTCON, 2
77	retfie
78	end multiplexacion

35

Adicional:

- Abecedario para display de siete segmentos:
https://en.wikichip.org/wiki/seven-segment_display/representing_letters

36

Fin de la sesión