

# Microcontroladores

## Laboratorio

Semana 5

Profesor Kalun José Lau Gan

Semestre 2021-2

1

### Preguntas previas

- ¿?

2

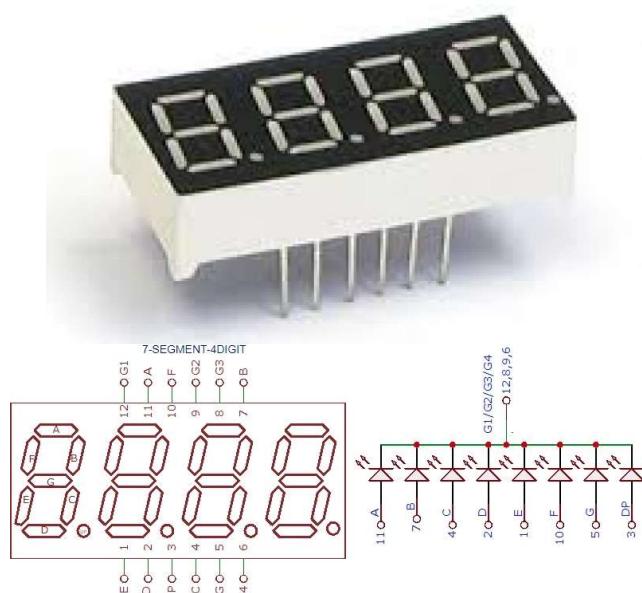
## Agenda

- 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.



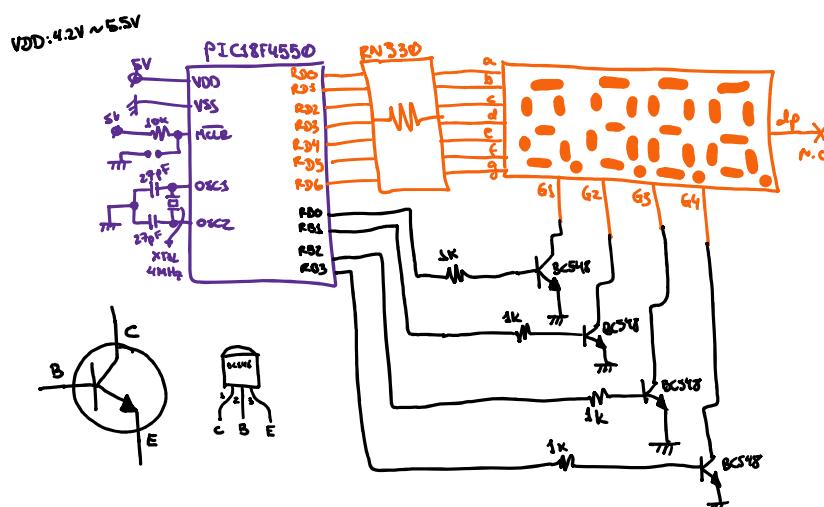
4

## 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

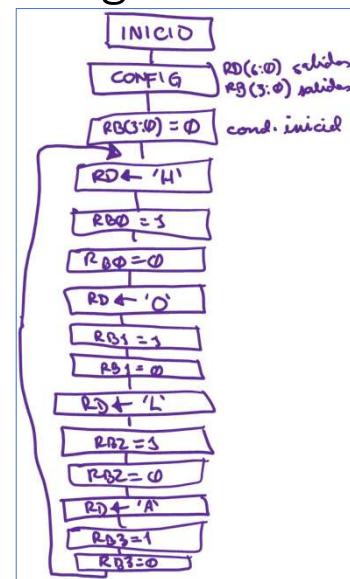
## Ejemplo: Visualizar “HOLA” en el display multiplexado



6

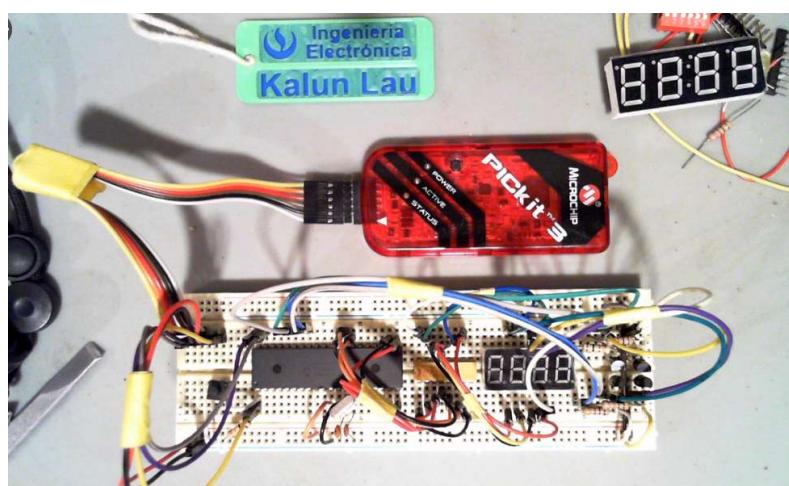
## Algoritmo para la multiplexación de los dígitos del display de siete segmentos

- Se debe de enviar el dato cuando se encuentren deshabilitados los dígitos
- Luego de enviar el dato se hará la habilitación del dígito respectivo, antes de cargar otro dato se deberá de deshabilitar el dígito activo.



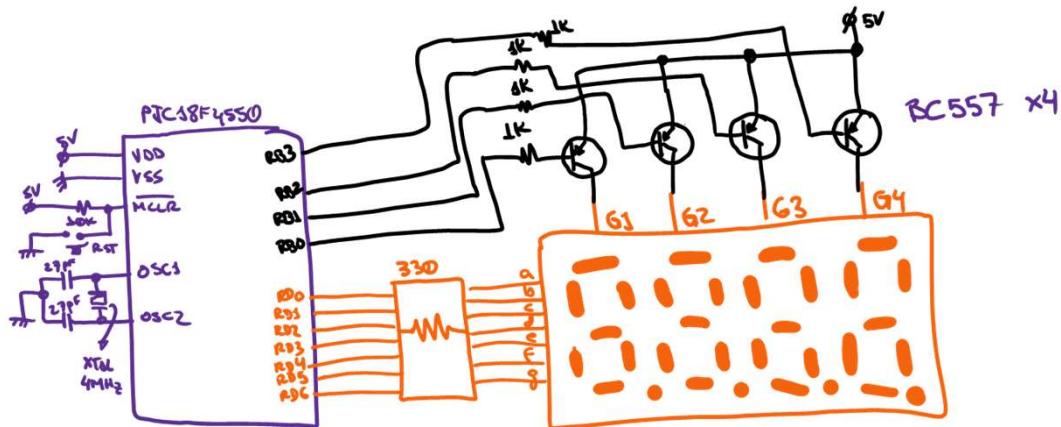
7

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



8

## Circuito para display multiplexado de ánodo común



## Código en XC8 PIC Assembler

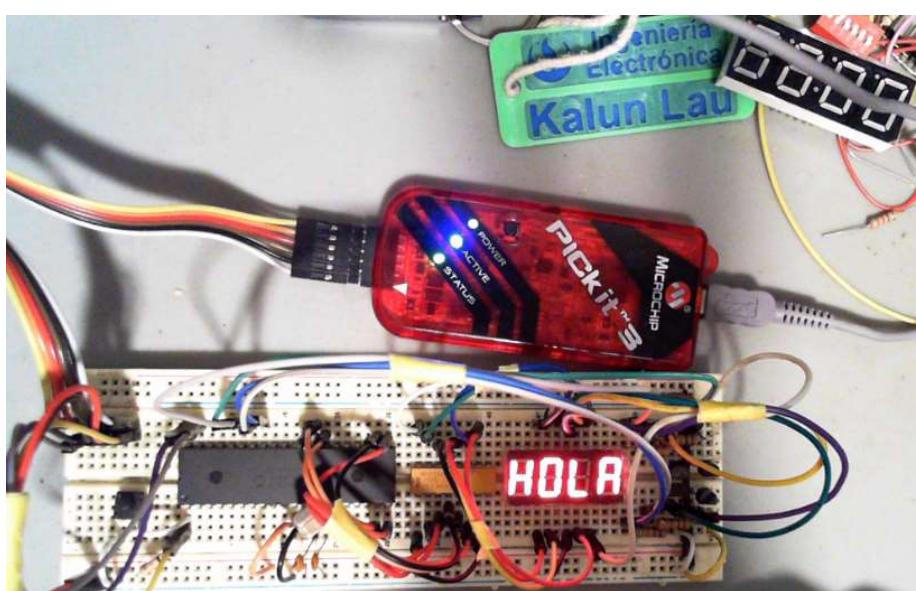
```

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 0x80
15     movwf TRISD      ;RD6-RD0 como salidas
16     movlw 0xF0
17     movwf TRISB      ;RB3-RB0 como salidas
18     clrf LATB        ;Condicion inicial de los habilitadores
19     inicio:
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 00111000B   ;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

```

11

## Pruebas en circuito:



12

## Código mejorado empleando TBLPTR (MPASM)

```

1 ;Este es un comentario
2      list p=18F4550
3      #include <p18f4550.inc>      ;libreria de nombre de los rs
4
5      CONFIG FOSC = INTOSCIO_EC      33
6      CONFIG CPUDIV = OSC1_PLL2      34
7      CONFIG FWRT = ON              35
8      CONFIG BOR = OFF              36
9      CONFIG WDT = OFF              37
10     CONFIG PBADEN = OFF           38
11     CONFIG LVP = OFF              39
12     CONFIG MCLRE = ON             40
13
14     org 0x0300                  41
15     ;mensaje db 0x76, 0x3F, 0x38, 0x77 ;HOLA      42
16     ;mensaje db 0x78, 0x79, 0x38, 0x77 ;tela      43
17     mensaje db b'01111100', b'00000110', b'00111001', 44
18
19     org 0x0000                  45
20     goto init_conf
21
22     org 0x0020                  46
23     init_conf: movlw 0x80          47
24         movwf TRISD            ;RD(6:0) como salidas 48
25         movlw 0xF0              49
26         movwf TRISB            ;RB(3:0) como salidas 50
27         clrf LATB              ;RB en cero        51
28         movlw HIGH mensaje      52
29         movwf TBLPTRH          53
30         movlw LOW mensaje       54
31         movwf TBLPTRL          55
32
33     loop: clrf TBLPTRL          56
34         TBLRD*+
35         movff TABLAT, LATD      57
36         bsf LATB, 0              58
37         call nop_group          59
38         bcf LATB, 0              60
39         TBLRD*+
40         movff TABLAT, LATD      61
41         bsf LATB, 1              62
42         call nop_group          63
43         bcf LATB, 1              64
44         TBLRD*+
45         movff TABLAT, LATD      65
46         bsf LATB, 2              66
47         call nop_group          67
48         bcf LATB, 2              68
49         TBLRD*+
50         movff TABLAT, LATD      69
51         bsf LATB, 3              70
52         call nop_group          71
53         bcf LATB, 3              72
54         goto loop
55
56     nop_group: nop
57         nop
58         nop
59         nop
60         return
61
62     end

```

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

13

## 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     movlw 0x80
19     movwf TRISD            ;RD6-RD0 como salidas
20     movlw 0xF0
21     movwf TRISB            ;RB3-RB0 como salidas
22     clrf LATB              ;Condicion inicial de los habilitadores
23     movlw HIGH mensaje
24     movwf TBLPTRH
25     movlw LOW mensaje
26     movwf TBLPTRL          ;TBLPTR apuntando a mensaje (300H)

```

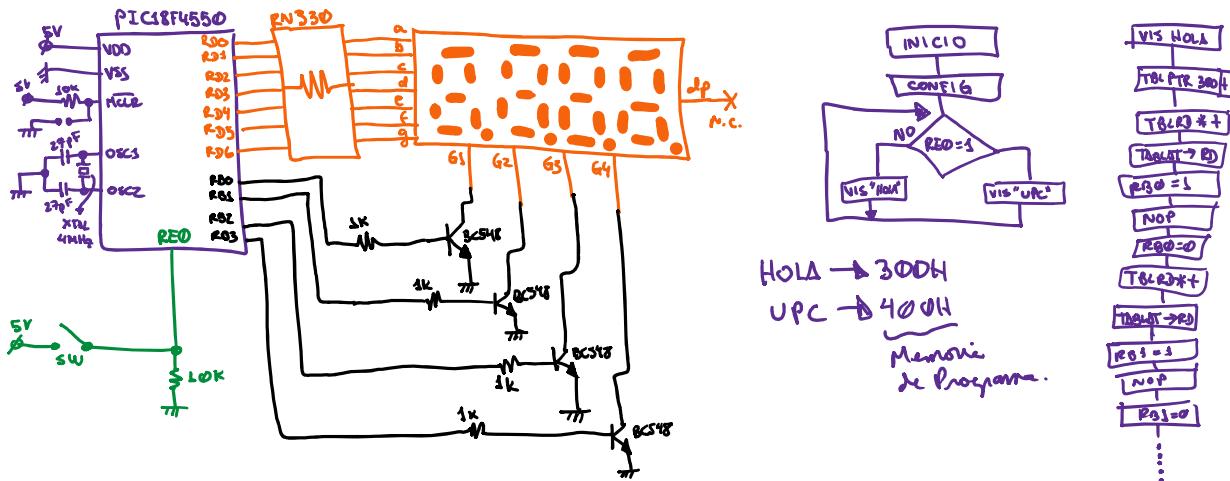
```

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

```

14

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



15

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

16

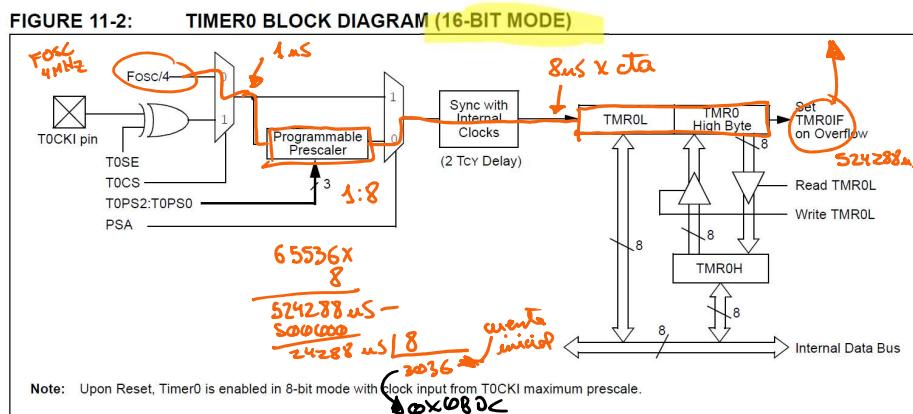
## Cambiar mensajes en el display multiplexado

- Para poder cambiar el mensaje de visualización de manera secuencial es necesario delegar la función de refresco o la función de cambio de mensaje a un módulo de temporización, en este caso el Timer0
- Como primera opción 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.

17

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

- El Timer0 debe de trabajar a 16 bits



18

(cont...)

REGISTER 11-1: T0CON: 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
TMR0N	T08BIT	T0CS	X	OP0	TOPS2	TOPS1	OP80
bit 7							bit 0
<b>Legend:</b>							
R = Readable bit -n = Value at POR	W = Writable bit '1' = Bit is set	U = Unimplemented bit, read as '0' '0' = Bit is cleared	x = Bit is unknown				
bit 7 <b>TMR0ON:</b> Timer0 On/Off Control bit 1 = Enables Timer0 ✓ 0 = Stops Timer0							<b>T0CON = 0x82</b>
bit 6 <b>T08BIT:</b> 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 <b>T0CS:</b> Timer0 Clock Source Select bit 1 = Transition on T0CKI pin ✓ 0 = Internal instruction cycle clock (CLKO)							
bit 4 <b>T0SE:</b> Timer0 Source Edge Select bit 1 = Increment on high-to-low transition on T0CKI pin X 0 = Increment on low-to-high transition on T0CKI pin							
bit 3 <b>PSA:</b> 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 <b>TOPS2:TOPS0:</b> 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							

19

(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-0
GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	REIE	TMR0IF	INT0IF	RFIF
bit 7							bit 0
<b>Legend:</b>							
R = Readable bit -n = Value at POR	W = Writable bit '1' = Bit is set	U = Unimplemented bit, read as '0' '0' = Bit is cleared	x = Bit is unknown				
bit 7 <b>GIE/GIEH:</b> 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							<b>INTCON = 0xA0</b>
bit 6 <b>PEIE/GIEL:</b> 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 <b>TMR0IE:</b> TMR0 Overflow Interrupt Enable bit 1 = Enables the TMR0 overflow interrupt ✓ 0 = Disables the TMR0 overflow interrupt							
bit 4 <b>INT0IE:</b> INT0 External Interrupt Enable bit 1 = Enables the INT0 external interrupt 0 = Disables the INT0 external interrupt							
bit 3 <b>RBIE:</b> RB Port Change Interrupt Enable bit 1 = Enables the RB port change interrupt 0 = Disables the RB port change interrupt							
bit 2 <b>TMR0IF:</b> TMR0 Overflow Interrupt Flag bit 1 = TMR0 register has overflowed (must be cleared in software) 0 = TMR0 register did not overflow							
bit 1 <b>INT0IF:</b> 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 <b>RBIF:</b> 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							

20

<pre> 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 init_conf: movlw 0x80 31      movwf TRISD      ;RD(6:0) como salidas 32      movlw 0xF0 33      movwf TRISB      ;RB(3:0) como salidas 34      clrf LATB        ;RB en cero 35      movlw HIGH mensaje 36      movwf TBLPTRH 37      movlw LOW mensaje 38      movwf TBLPTRL 39      movlw 0x81 40      movwf TOCON       ;Tmr0 ON, fosc/4, psc 1:8, modo 16bit 41      movlw 0xA0 42      movwf INTCON      ;GIE=1, TMROIE=1 interrupcion activo para 43 loop:   clrf TBLPTRL 44      TBLRD*+ 45      movff TABLAT, LATD 46      bsf LATB, 0 47      call nop_group 48      bcf LATB, 0 49      TBLRD*+ 50      movff TABLAT, LATD 51      bsf LATB, 1 52      call nop_group 53 </pre>	<pre> 54      bcf LATB, 1 55      TBLRD*+ 56      movff TABLAT, LATD 57      bsf LATB, 2 58      call nop_group 59      bcf LATB, 2 60      TBLRD*+ 61      movff TABLAT, LATD 62      bsf LATB, 3 63      call nop_group 64      bcf LATB, 3 65      goto loop 66 nop_group: nop 67      nop 68      nop 69      nop 70      return 71 Tmr0_ISR:  movlw 0x0B 72      movwf TMROH 73      moviw 0xDC 74      movwf TMROL      ;Carga de cuenta inicial a TMRO 75      movlw 0x05 76      cpfseq TBLPTRH  ;Pregunto si estoy en el ultimo mensaje 77      goto aunno 78      moviw 0x03      ;Paso al primer mensaje 79      movwf TBLPTRH 80      goto otro 81 aunno:   incf TBLPTRH, f ;Paso al siguiente mensaje 82 otro:    bcf INTCON, TMROIF 83      retfie 84      end 85 86 </pre>
--	---

21

## Estrategia1: Timer0 como elemento de refresco del display

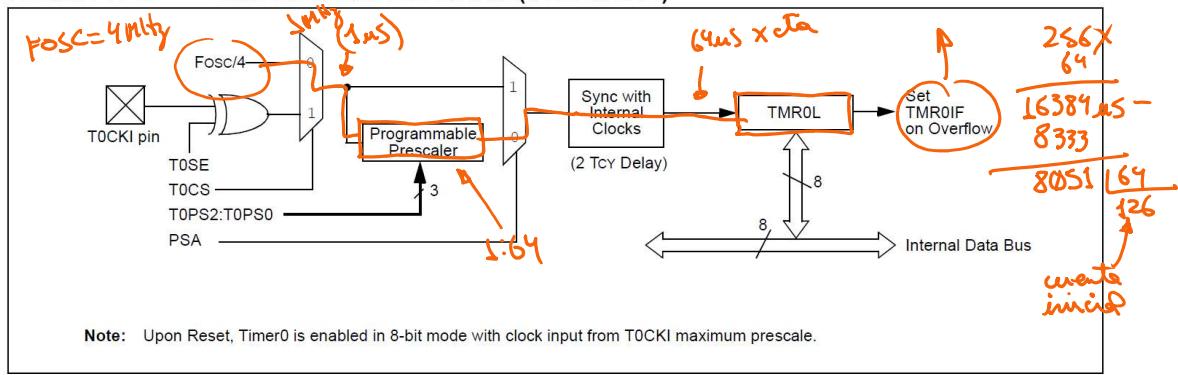
- Al optar que el Timer0 sea el elemento que haga el proceso de refresco en el display multiplexado, éste debe de temporizar lo suficientemente rápido para que aparezca el efecto de “persistencia visual”, a partir de una frecuencia de 50Hz el ojo humano ya le es difícil percibir un parpadeo o refresco en un display ó pantalla.
- Para el siguiente ejemplo se hará 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.

22

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

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

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



**Note:** Upon Reset, Timer0 is enabled in 8-bit mode with clock input from T0CKI maximum prescale.

23

(cont.)

#### REGISTER 11-1: T0CON: 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
TMROUN	T05BT	TOCS	TSE	PSA	T0PS2	T0PS1	TPS0
bit 7							bit 0

<b>Legend:</b>		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
n = Value at ROP	'1' = Bit is set	'0' = Bit is cleared

bit 7	<b>TMR0ON:</b> Timer0 On/Off Control bit 1 = Enables Timer0 0 = Stops Timer0
bit 6	<b>T08BIT:</b> 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	<b>T0CS:</b> Timer0 Clock Source Select bit 1 = Transition on T0CKI pin 0 = Internal instruction cycle clock (CLKO) ✓
bit 4	<b>T0SE:</b> 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	<b>PSA:</b> 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	<b>TOPS2:TOPS0:</b> 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

(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-0
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 <sup>(1)</sup> 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
<b>Legend:</b> 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							
<b>INTCON = 0XA0</b>							

25

## Código en MPASM

```

1 ;Este es un comentario, se le antecede un punto
2 list p18f4550 ;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 Reset
11 CONFIG WDT = OFF ; Watchdog Timer
12 CONFIG CCP2MX = ON ; CCP2 MUX
13 CONFIG PBADEN = OFF ; PORTB A/D
14 CONFIG MCLEN = 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 ;PERU
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 programas de usuario
37 init_conf: movlw 0x80
38     movwf TRISD ;RD(6:0) son salidas
39     movlw 0xF0
40     movwf TRISB ;RB(3:0) son salidas
41     clrf LATB ;Los cuatro habilitados
42     movlw 0xC5
43     movwf TCON ;TMR0 on, fosc/4, prescaler
44     movlw 0xA0
45     movwf INTCON ;GIE=1, TMROIE=1 (interna)
46     movlw .126
47     movwf THRL ;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 ;abajamos 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: call bucle2 ;Salto
102    decfsz var_j, f
103    goto otro2
104    return
105
106 bucle2:
107    movlw .50
108    movwf var_k
109
110    otro3: decfsz var_k, f
111    goto otro3
112    return
113
114    bucle3:
115    decfsz var_k, f
116    goto bucle3
117    return
118    end

```

26

## 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.

27

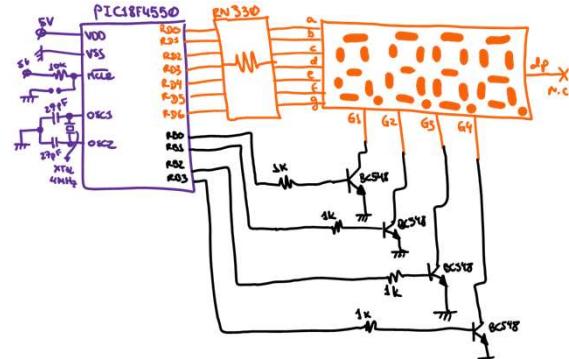
## Estrategia2: Timer0 como elemento de cambio (desplazamiento) del mensaje

- 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, se empleará entonces el modo de 16 bits.

28

## 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.



29

## 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, 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
21  ORG 00020H
22  configuracion:
23  movlw 0x80
24  movwf TRISD      ;RD6-RD0 como salidas
25  movlw 0xF0
26  movwf TRISE      ;RB3-RB0 como salidas
27  clrf LATB        ;Condicion inicial de los habilitadores
28  movlw HIGH mensaje
29  movwf TBLPTRH
30  movlw LOW mensaje
31  movwf TBLPTRL    ;TBLPTR apunta a 300H
32  movwf TOCON      ;Configuracion del TMRO: Fosc/4 16bit PSCL:4
33  movlw 0A0H
34  movwf INTCON      ;Configuracion de Ints: Desborde de TMRO
35  clrf cuenta

```

37	inicio:	38	movf cuenta, 0	39	addwf TBLPTRL	40	TBLRD++	41	movff TABLAT, LATD	42	bcf LATB, 0	43	nop	44	bcl LATB, 0	45	TBLRD++	46	movff TABLAT, LATD	47	bcl LATB, 1	48	nop	49	bcl LATB, 1	50	TBLRD++	51	movff TABLAT, LATD	52	bcl LATB, 2	53	nop	54	bcl LATB, 2	55	TBLRD++	56	movff TABLAT, LATD	57	bcl LATB, 3	58	nop	59	bcl 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 THROH	74	movlw 0DCH	75	movwf THROL	76	bcl INTCON, 2	77	retfie	78	end multiplexacion																		

30

## Adicional:

- Abecedario para display de siete segmentos:  
[https://en.wikichip.org/wiki/seven-segment\\_display/representing\\_letters](https://en.wikichip.org/wiki/seven-segment_display/representing_letters)

31

## Fin de la sesión

32