

Microcontroladores

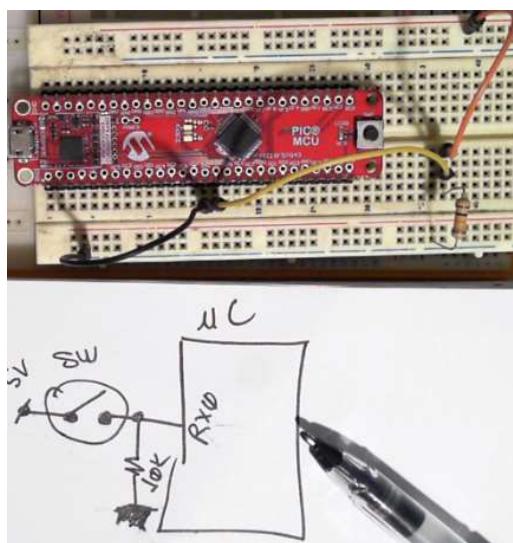
Semana 5

Semestre 2023-2
Por Kalun José Lau Gan

1

Preguntas previas

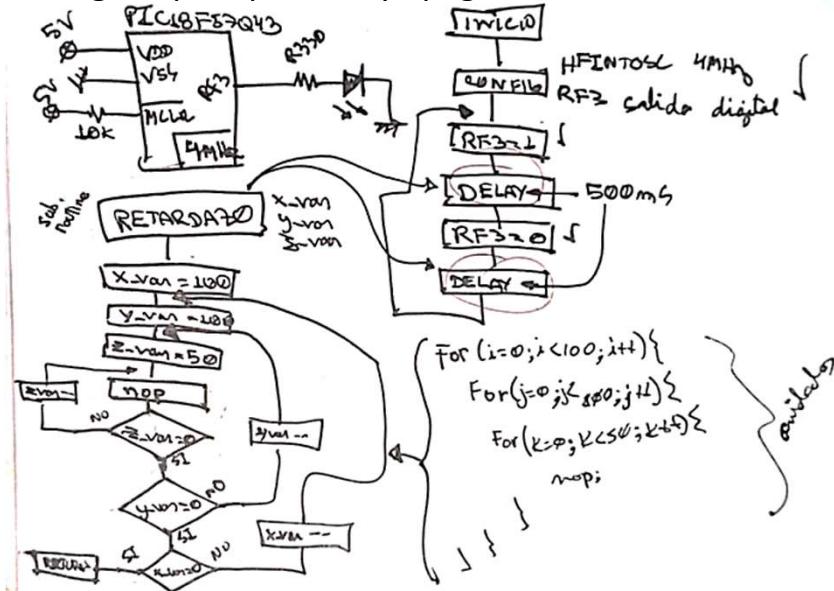
- ¿Reemplazar un switch con solamente un cable jumper?



2

Preguntas previas

- ¿Retardo prolongado para prender y apagar un LED?



3

Preguntas previas

- ¿Cómo se trabaja con el bit de overflow?

STATUS Register								
Bit	7	6	5	4	3	2	1	0
Access		TO	R	R	OV R/W 0	Z R/W 0	DC R/W 0	C R/W 0
Reset		1	1	0				

cuenta EQU 500H

```

inicio:
    movlb 5H
    movlw 250
    movwf cuenta, 1
    incf cuenta, 1, 1
    movlb 4H
    btfss STATUS, 3, 1
    bra aunno
    return
aunno:
    movlb 5H
    bra bucle
  
```

4

Preguntas previas

- ¿Qué significa BRA \$-2 ó GOTO \$-2?
 - Indica salto a una posición de 2 bytes anteriores (una instrucción simple)

```

bucle:    movlw 55H          ;01010101B
             movwf LATB, 1
             comf LATB, 1, 1
             bra $-6

retardon: ---           ;subrutina
             ---
             ---
             ---
             ---
             return
  
```

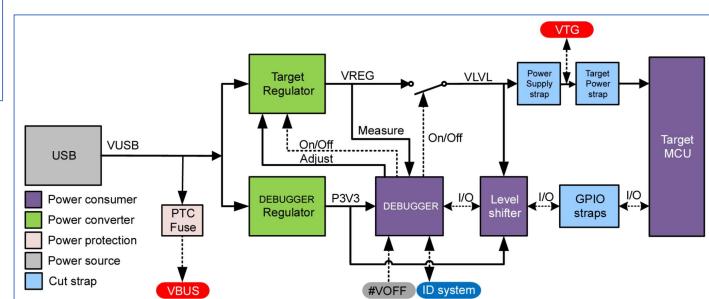
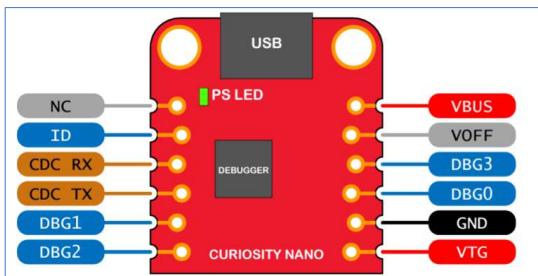
```

inicio:   btfss PORTB, 4, 1
             bra $-2
             btg LATF, 3, 1
             btfsc PORTB, 4, 1
             bra $-2
             bra inicio
  
```

5

Preguntas previas

- Para alimentar el protoboard desde el Curiosity Nano. ¿Se conectan VBUS y VTG a la línea roja?
 - No, ni lo intentes!



6

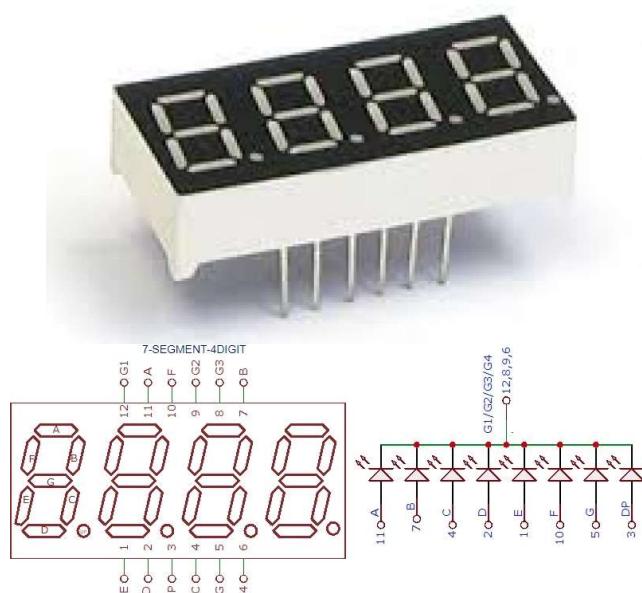
Agenda

- Multiplexación en displays de siete segmentos
- El display multiplexado de 4 dígitos de siete segmentos
- Ejemplos de aplicaciones con multiplexación de displays de siete segmentos

7

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.



8

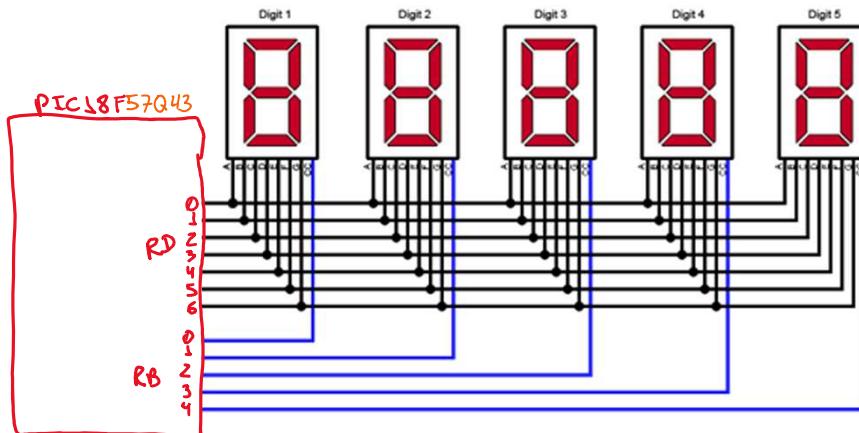
Multiplexación en displays de siete segmentos

- 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”

9

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

- Al igual que en displays de siete segmentos individuales, se deben de colocar resistencias en cada uno de los segmentos para limitar la corriente que pasa por ellos.
- Se recomienda no conectar los habilitadores de frente a los I/O del microcontrolador ya que dichos I/O no soportan entregar/recepcionar corrientes altas.

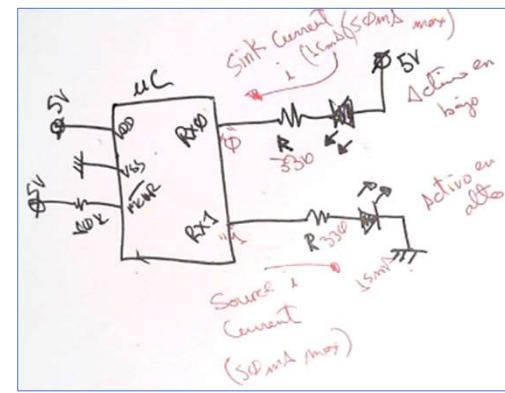


10

¿Corriente máxima por I/O en el PIC18F57Q43?

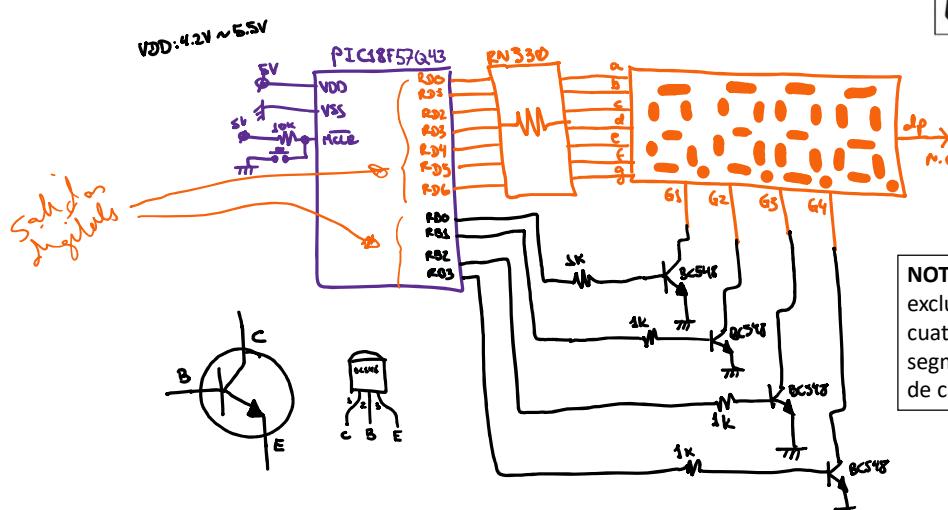
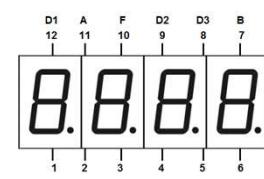
Absolute Maximum Ratings⁽¹⁾

Parameter	Rating
Ambient temperature under bias	-40°C to +125°C
Storage temperature	-65°C to +150°C
Voltage on pins with respect to V _{SS}	
• on V _{DD} pin:	-0.3V to +6.5V
• on MCLR pin:	-0.3V to +9.0V
• on all other pins:	-0.3V to (V _{DD} + 0.3V)
Maximum current ⁽¹⁾	
• on V _{SS} pin	-40°C ≤ T _A ≤ +85°C 85°C < T _A ≤ +125°C 120 mA
• on V _{DD} pin (28-pin devices)	-40°C ≤ T _A ≤ +85°C 85°C < T _A ≤ +125°C 250 mA
• on V _{DD} pin (40-pin devices)	-40°C ≤ T _A ≤ +85°C 85°C < T _A ≤ +125°C 350 mA
• on any standard I/O pin	120 mA ±50 mA
Clamp current, I _K (V _{PIN} < 0 or V _{PIN} > V _{DD})	±20 mA
Total power dissipation ⁽²⁾	800 mW



11

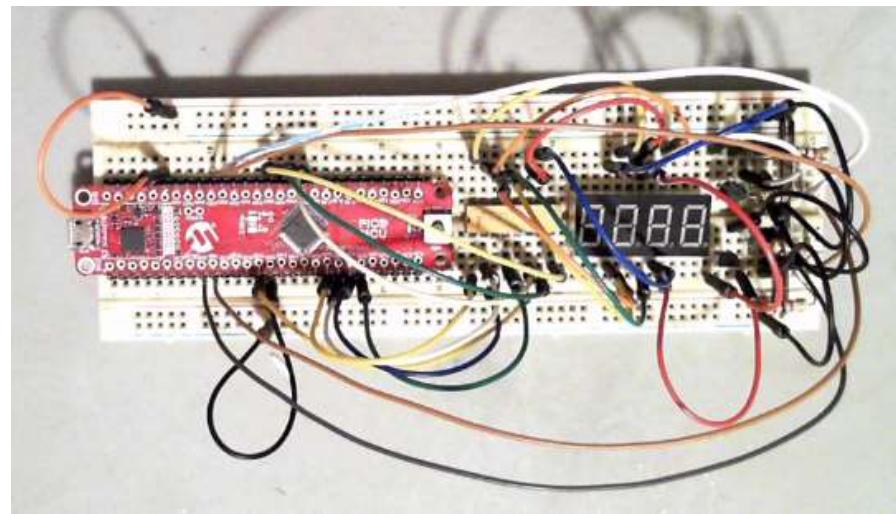
Ejemplo: Visualizar "HOLA" en el display multiplexado



12

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

Circuito implementado



13

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

7-Segment LED Display Animator - 4-Digit

Digit 1 0x78 0b01110110	Digit 2 0x3f 0b00111111	Digit 3 0x38 0b00111000	Digit 4 0x77 0b01110111
<input type="button" value="Save Frame"/> <input type="button" value="Playback Animation"/> <input type="button" value="Clear"/> <input type="button" value="Frame = 0"/> <input type="button" value="Delay = 100"/> <input type="button" value="HGFEDECBA"/>			

Instructions: Click segments above to toggle LED state and click "Save" to record frame.
Data for the animation will be generated below. Click the "Copy Code" button and paste the data definition into your sketch. See: [TM1637TinyDisplay](#) (Animator: 4-Digit | 6-Digit)

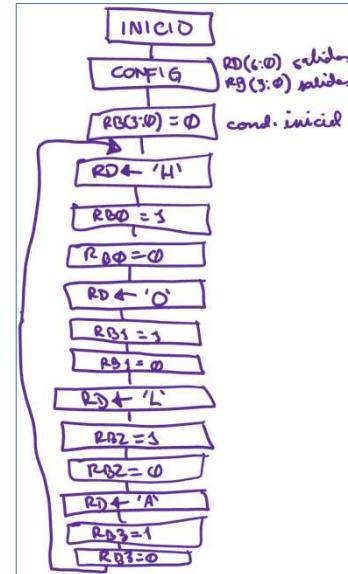
Animation Data	Copy Code	Clear Data	Format Data	Example
[0x00, 0x00, 0x00, 0x00], // Frame 0				

<https://jasonacox.github.io/TM1637TinyDisplay/examples/7-segment-animator.html>

14

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

- “visualización dinámica”
- 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.



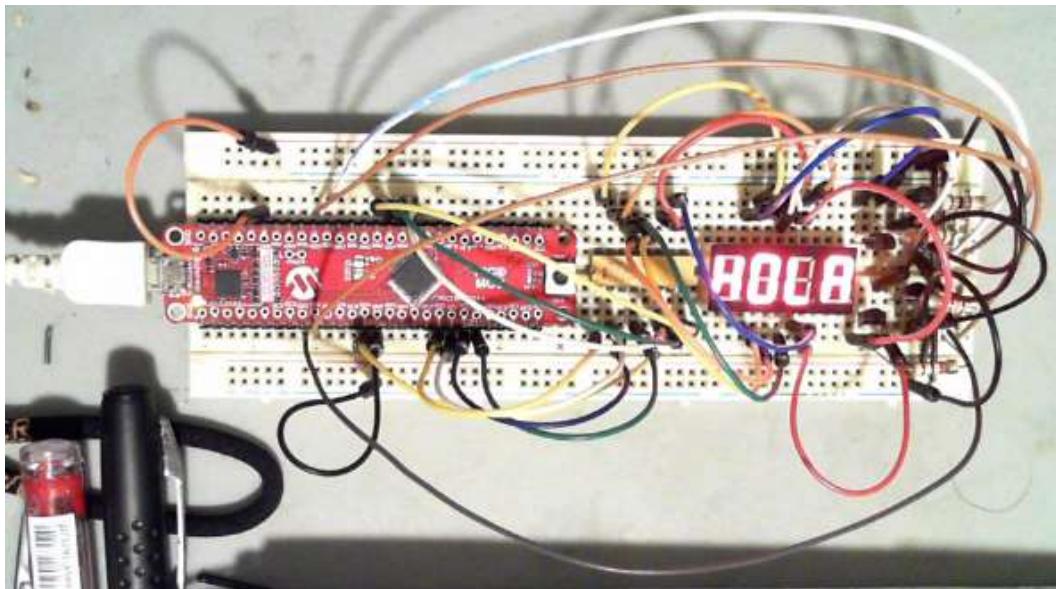
15

Código en XC8 PIC Assembler

<pre> 1 PROCESSOR 18F57Q43 2 #include "cabecera.inc" 3 4 PSECT upcino, class=CODE, reloc=2, abs 5 upcino: 6 ORG 000000H 7 bra configuro 8 9 ORG 000100H 10 configuro: 11 movlb 0H 12 movlw 60H 13 movwf OSCCON1, 1 14 movlw 02H 15 movwf OSCFRC, 1 16 movlw 40H 17 movwf OSCEN, 1 18 movlb 4H 19 movlw 80H 20 movwf TRISD, 1 ;RD6-RD0 como salidas 21 movwf ANSEL0, 1 ;RD6-RD0 como digitales 22 movlw OFOH 23 movwf TRISB, 1 ;RB3-RB0 como salidas 24 movwf ANSEL0, 1 ;RB3-RB0 como digitales 25 clrf LATB, 1 ;Condicion inicial: habilitadores todos en cero </pre>	<pre> 26 inicio: 27 movlw 76H 28 movwf LATD, 1 ;cargando H en RD 29 bcf LATB, 0, 1 ;habilito digito 0 30 call nopes 31 bcf LATB, 0, 1 ;deshabilito digito 0 32 movlw 3FH 33 movwf LATD, 1 ;cargando O en RD 34 bcf LATB, 1, 1 ;habilito digito 1 35 call nopes 36 bcf LATB, 1, 1 ;deshabilito digito 1 37 movlw 38H 38 movwf LATD, 1 ;cargando L en RD 39 bcf LATB, 2, 1 ;habilito digito 2 40 call nopes 41 bcf LATB, 2, 1 ;deshabilito digito 2 42 movlw 77H 43 movwf LATD, 1 ;cargando A en RD 44 bcf LATB, 3, 1 ;habilito digito 3 45 call nopes 46 bcf LATB, 3, 1 ;deshabilito digito 3 47 bra inicio 48 49 nopes: 50 nop 51 nop 52 nop 53 nop 54 nop 55 nop 56 return 57 58 end upcino </pre>
---	---

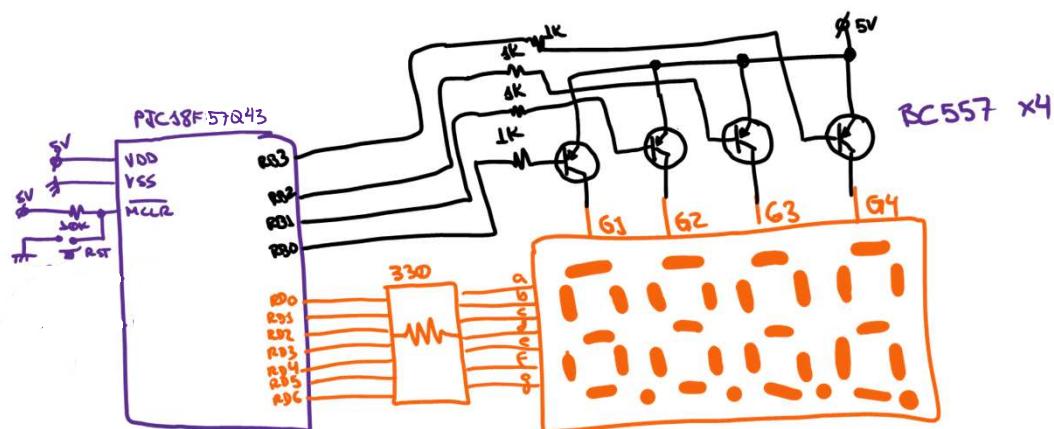
16

Pruebas en circuito:



17

Observación: Circuito para display multiplexado de ánodo común



18

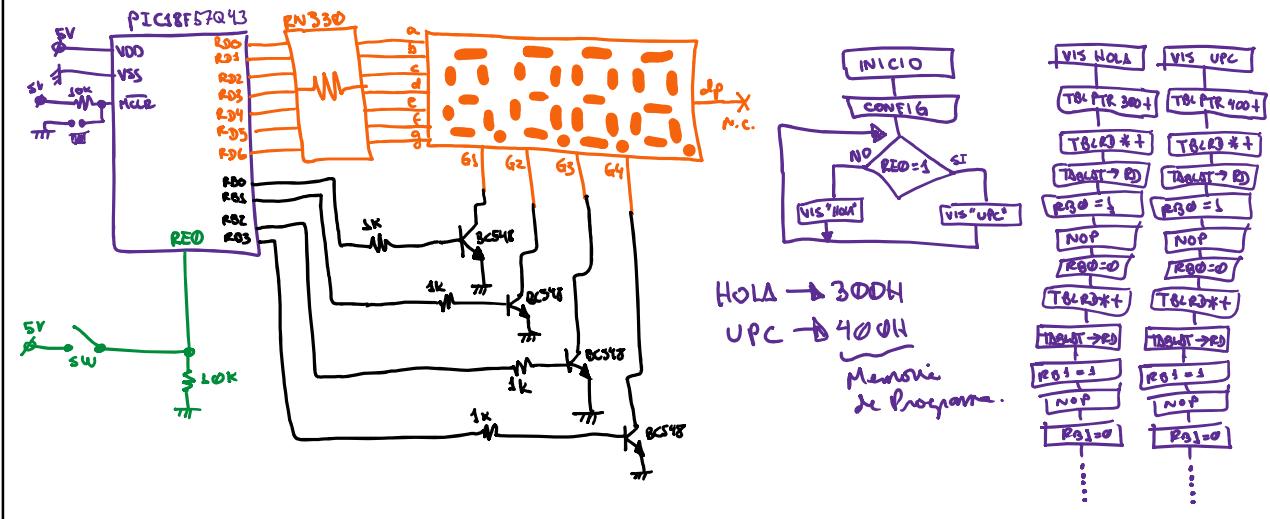
Código mejorado empleando TBLPTR (XC8 PIC Assembler)

1	PROCESSOR 18F57Q43	29	inicio:
2	#include "cabecera.inc"	30	clrf TBLPTR, 1
3		31	movlw 07H
4	PSECT upcino, class=CODE, reloc=2, abs	32	movwf TBLPTRH,1
5	upcino:	33	clrf TBLPTRL, 1 ;TBLPTR esta apuntando a la direccion 700H de mem prog
6	ORG 000000H	34	TBLRD*+ ;Lee el contenido de TBLPTR (700H) y luego lo incrementa
7	bra configuro	35	movff TABLAT, LATD ;Mueve el contenido de TABLAT hacia LATD
8		36	bsf LATB, 0, 1 ;habilito digito 0
9	ORG 000700H	37	call nopes
10	mensaje: DB 76H, 3FH, 38H, 77H	38	bcf LATB, 0, 1 ;deshabilito digito 0
11		39	TBLRD*+ ;Lee el contenido de TBLPTR (701H) y luego lo incrementa
12	ORG 000100H	40	movff TABLAT, LATD ;Mueve el contenido de TABLAT hacia LATD
13	configuro:	41	bsf LATB, 1, 1 ;habilito digito 1
14	movlb 0H	42	call nopes
15	movlw 60H	43	bcf LATB, 1, 1 ;deshabilito digito 1
16	movwf OSCCON1, 1	44	TBLRD*+ ;Lee el contenido de TBLPTR (702H) y luego lo incrementa
17	movlw 02H	45	movff TABLAT, LATD ;Mueve el contenido de TABLAT hacia LATD
18	movwf OSCFRC, 1	46	bsf LATB, 2, 1 ;habilito digito 2
19	movlw 40H	47	call nopes
20	movwf OSCEN, 1	48	bcf LATB, 2, 1 ;deshabilito digito 2
21	movlb 4H	49	TBLRD*+ ;Lee el contenido de TBLPTR (703H) y luego lo incrementa
22	movlw 80H	50	movff TABLAT, LATD ;Mueve el contenido de TABLAT hacia LATD
23	movwf TRISD, 1 ;RD6-RD0 como salidas	51	bsf LATB, 3, 1 ;habilito digito 3
24	movwf ANSELD, 1 ;RD6-RD0 como digitales	52	call nopes
25	movlw 0FOH	53	bcf LATB, 3, 1 ;deshabilito digito 3
26	movwf TRISB, 1 ;RB3-RB0 como salidas	54	bra inicio
27	movwf ANSELB, 1 ;RB3-RB0 como digitales		
28	clrf LATB, 1 ;Condicion inicial: habilitadores todos en cero		
		56	nopes:
		57	nop
		58	nop
		59	nop
		60	nop
		61	nop
		62	nop
		63	return
		64	
		65	end upcino

19

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

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



20

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

```

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
39
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 multiplexacion

```

21

Modificando el ejemplo para que pueda intercambiar el mensaje con la entrada RB4: 0=HOLA, 1=PATO

```

1      PROCESSOR 18F57Q43
2      #include "cabecera.inc"
3
4      PSECT upcino, class=CODE, reloc=2, abs
5      upcino:
6      ORG 000000H
7      bra configuro
8
9      ORG 000700H
10     ;H   O   L   A
11     mensaje: DB 76H, 3FH, 38H, 77H
12     ORG 000750H
13     ;P   A   T   O
14     mensaje2: DB 73H, 77H, 78H, 3FH
15
16     ORG 000100H
17     configuro:
18     movlb OH
19     movlw 60H
20     movwf OSCCON1, 1
21     movlw 02H
22     movwf OSCFRQ, 1
23     movlw 40H
24     movwf OSCEN, 1
25     movlb 4H
26     movlw 80H
27     movwf TRISD, 1    ;RD6:RD0 como salidas
28     movwf ANSELB, 1   ;RD6:RD0 como digitales
29     movlw 0F0H
30     movwf TRISB, 1    ;RB3:RB0 como salidas
31     movwf ANSELB, 1   ;RB3:RB0 como digitales
32     bsr TRISE, 4, 1   ;RB4 como entrada
33     bcf ANSELB, 4, 1   ;RB4 como digital
34     bcf WPUB, 4, 1    ;RB4 con pullup
35     clrf LATB, 1       ;condicion inicial
36
37     inicio:
38     btfsc PORTB, 4, 1  ;pregunto si RB4=1
39     bra noperacion
40     bra siperacione
41
42     noperacione:
43     clrf TBLPTRU, 1
44     movlw 07H
45     movwf TBLPTRH, 1
46     clrf TBLPTRL, 1
47     TBLRD*+
48     movff TABLAT, LATD
49     bsf LATB, 0
50     call nopes
51     bcf LATB, 0
52     TBLRD*+
53     movff TABLAT, LATD
54     bsf LATB, 1
55     call nopes
56     bcf LATB, 1
57     TBLRD*+
58     movff TABLAT, LATD
59     bsf LATB, 2
60     call nopes
61     bcf LATB, 2
62     TBLRD*+
63     movff TABLAT, LATD
64     bsf LATB, 3
65     call nopes
66     bcf LATB, 3
67     bra inicio
68
69     siperacione:
70     clrf TBLPTRU, 1
71     movlw 07H
72     movwf TBLPTRH, 1
73     movlw 50H
74     movwf TBLPTRL, 1
75     TBLRD*+
76     movff TABLAT, LATD
77     bsf LATB, 0, 1
78     call nopes
79     bcf LATB, 0, 1
80     TBLRD*+
81     movff TABLAT, LATD
82     bsf LATB, 1, 1
83     call nopes
84     bcf LATB, 1, 1
85     TBLRD*+
86     movff TABLAT, LATD
87     bsf LATB, 1, 1
88     call nopes
89     bcf LATB, 2, 1
90     TBLRD*+
91     movff TABLAT, LATD
92     bsf LATB, 2, 1
93     call nopes
94     bcf LATB, 2, 1
95     TBLRD*+
96     movff TABLAT, LATD
97     call nopes
98     bcf LATB, 3, 1
99     call nopes
100    bcf LATB, 3, 1
101    call nopes
102    bcf LATB, 3, 1
103    bcf LATB, 3, 1
104    bcf LATB, 3, 1
105    bcf LATB, 3, 1
106    bcf LATB, 3, 1
107    bcf LATB, 3, 1
108    bra inicio
109
110    nopes:
111    nop
112    nop
113    nop
114    nop
115    nop
116    nop
117    nop
118    return
119
120    end upcino

```

22

Ejemplo para visualizar HOLA y CERO con control en RB4

```

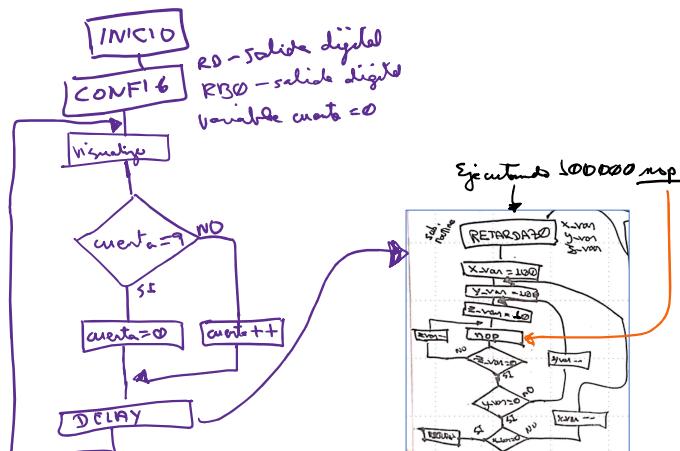
1  PROCESSOR 18F57Q43
2  #include "cabecera.inc"
3
4  PSECT upcino, class=CODE,reloc=2, abs
5  upcino:
6      ORG 000000H ;Vector de RESET
7      bra configuro
8
9      ORG 000200H ;Direccion de memoria de progr
10     ;H O L A
11     mensaje1: DB 76H, 3FH, 38H, 77H
12     ORG 000220H ;Direccion de memoria de progr
13     ;C E R O
14     mensaje2: DB 39H, 79H, 50H, 3FH
15
16     ORG 000100H
17 configuro:
18     movlb 0H
19     movlw 60H
20     movwf OSCCON1, 1
21     movlw 02H
22     movwf OSCFRC, 1
23     movlw 40H
24     movwf OSCEN, 1
25     movlb 4H
26     movlw 80H
27     movwf TRISD, 1 ;RD6-RD0 como salidas
28     movwf ANSEL0, 1 ;RD6-RD0 como digitales
29     movlw OFOH
30     movwf TRISB, 1 ;RB3-RB0 como salidas
31     movwf ANSELB, 1 ;RB3-RB0 como digitales
32     bsf TRISB, 4, 1 ;RB4 como entrada
33     bcf ANSELB, 4, 1 ;RB4 como digital
34     bsf WPUB, 4, 1 ;RB4 con pullup activado
35     clrf LATB, 1 ;Habilitadores en cero (cond i
36
37     inicio:
38         btfs PORTB, 4, 1 ;Pregunto si presione el boton
39         bra nopesione ;viene aqui cuando es falso
40         clrf TBLPTRU, 1 ;viene aqui cuando es verdadero
41         movlw 02H
42         movwf TBLPTRH, 1
43         movlw 20H
44         movwf TBLPTRL, 1 ;El TBLPTR esta mirando a dirección 22
45         bra siguiente
46         nopresione:
47             clrf TBLPTRU, 1
48             movlw 02H
49             movwf TBLPTRH, 1
50             clrf TBLPTRL, 1 ;El TBLPTR esta mirando a dirección 20
51         siguiente:
52             TBLRD*+ ;Lectura del contenido apuntado por TB
53             movff TABLAT, LATD
54             bsf LATB, 0, 1
55             call nopes
56             bcf LATB, 0, 1
57             TBLRD*+ ;Lectura del contenido apuntado por TB
58             movff TABLAT, LATD
59             bsf LATB, 1, 1
60             call nopes
61             bcf LATB, 1, 1
62             TBLRD*+ ;Lectura del contenido apuntado por TBLPTR y post increm
63             movff TABLAT, LATD
64             bsf LATB, 2, 1
65             call nopes
66             bcf LATB, 2, 1
67             TBLRD*+ ;Lectura del contenido apuntado por TBLPTR y post increm
68             movff TABLAT, LATD
69             bsf LATB, 3, 1
70             call nopes
71             bcf LATB, 3, 1
72             bra inicio
73
74     nopes:
75         nop
76         nop
77         nop
78         nop
79         nop
80         nop
81         nop
82         nop
83         nop
84         nop
85         return
86
87     end upcino

```

23

Ejemplo: Contador 0-9 (BCD) autoincremental con periodo de incremento de aprox 100ms

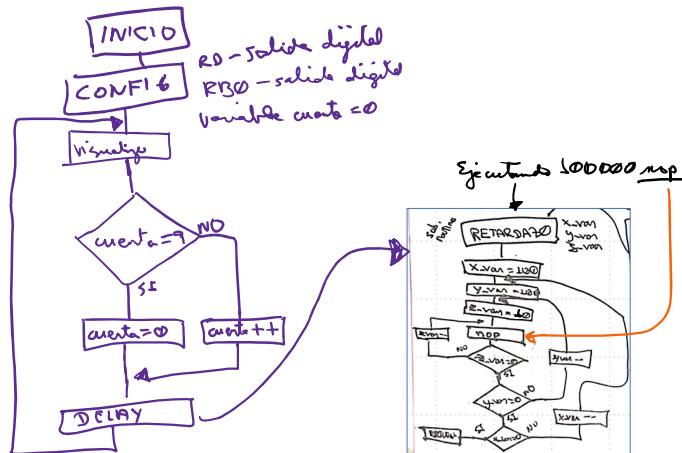
- El mismo hardware que el ejemplo anterior
- No habrá multiplexación, se usara solamente un dígito del display



24

Ejemplo: Contador 0-9 (BCD) autoincremental con periodo de incremento de aprox 100ms

- El mismo hardware que el ejemplo anterior
- No habrá multiplexación, se usara solamente un dígito del display



25

Ejemplo: Contador 0-9 (BCD) autoincremental con periodo de incremento de aprox 100ms

```

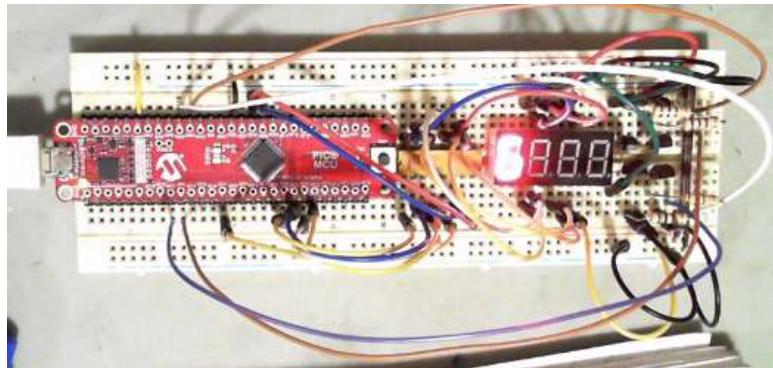
1  PROCESSOR 18F57Q43
2  #include "cabecera.inc"
3
4  PSECT upcino, class=CODE, reloc=2, abs
5
6  REG CUENTA EQU 500H ;etiqueta para la cuenta del contador
7  X_var EQU 501H
8  Y_var EQU 502H
9  Z_var EQU 503H
10
11 upcino:
12   ORG 000000H
13   bra configuro
14
15   ORG 000300H
16   ; 0 1 2 3 4 5 6 7 8 9
17 tabla_deco: DB 3FH, 06H, 5BH, 4FH, 66H, 6DH, 7DH, 07H, 7EH, 67H
18
19   ORG 000100H
20 configuro:
21   movlb 0H
22   movlw 60H
23   movwf OSCCON1, 1
24   movlw 02H
25   movwf OSCFRQ, 1 ;HFINTOSC = 4MHz
26   movlw 40H
27   movwf OSCEN, 1
28   movlw 5H
29   clrf REG CUENTA, 1
30   movlb 4H
31   movlw 80H
32   movwf TRISD, 1 ;RD6 al RDO como salidas
33   movwf ANSEL0, 1 ;RD6 al RDO como digitales
34   movlw 0FH
35   movwf TRISB, 1 ;RB3 al RB0 como salidas
36   movwf ANSEL1, 1 ;RB3 al RB0 como digitales
37   movlw 08H
38   movwf LATB, 1 ;Esta habilitado el cuarto dígito del display
40   inicio:
41   movlb 4H
42   clrf TBLPTRU, 1
43   movlw 03H
44   movwf TBLPTRH, 1
45   clrf TBLPTRL, 1 ;puntero de tabla apuntando a 300H
46   movlb 5H
47   movf REG CUENTA, 0, 1 ;recabo cuenta
48   movlb 4H
49   movwf TBLPTRL, 1 ;muevo contenido de Wreg a TBLPTR
50   TBLRD+ ;lectura de lo que esta apuntando
51   movwf TABLAT, LATD ;mueve lo leido hacia RD
52   movlb 5H
53   movlw 9
54   cpfseq REG CUENTA, 1 ;pregunto si la cuenta = 9
55   bra noescierto ;cuenta esta entre 0 y 8
56   bra siescierto ;cuenta = 9
57 noescierto:
58   incf REG CUENTA, 1, 1 ;incremento la cuenta
59   bra siguiente
60 siescierto:
61   clrf REG CUENTA, 1 ;limpiando la cuenta
62   bra siguiente
63 siguiente:
64   call retardazo
65   bra inicio
66
67   retardazo:
68   movlb 5H
69   movlw 100
70   movwf X_var, 1
71 bucle3:
72   movlw 100
73   movwf Y_var, 1
74 bucle2:
75   movlw 10
76   movwf Z_var, 1
77 bucle1:
78   nop
79   decfsz Z_var, 1, 1
80   bra bucle1
81   decfsz Y_var, 1, 1
82   bra bucle2
83   decfsz X_var, 1, 1
84   bra bucle3
85   return
86
87 end upcino

```

26

Ejemplo: Contador 0-9 (BCD) autoincremental con periodo de incremento de aprox 100ms

- Implementación



27

Ejercicios:

- Ampliar la aplicación de visualización de dos mensajes para cuatro mensajes (empleando dos pulsadores para las cuatro combinaciones)
- Modificar la aplicación de visualización de dos mensajes con cambio manual en base a una entrada, hacia una visualización automática de ambos mensajes con periodo de cambio de 2 segundos.
- Realizar un contador 0000-9999 ya sea autoincremental con periodo de cuenta de un segundo aproximadamente como de manera manual a través de una entrada de reloj.
- Elaborar una estrategia para que los mensajes a visualizar tengan un efecto de desplazamiento de derecha a izquierda.

28

Fin de la sesión