

# Microcontroladores

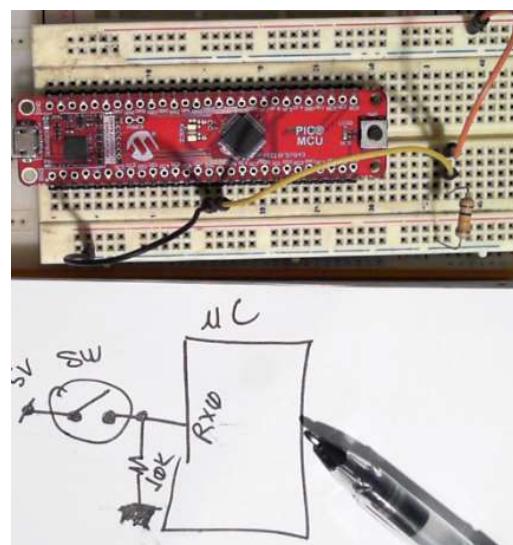
## Semana 5

Semestre 2024-1  
Por Kalun José Lau Gan

1

### Preguntas previas

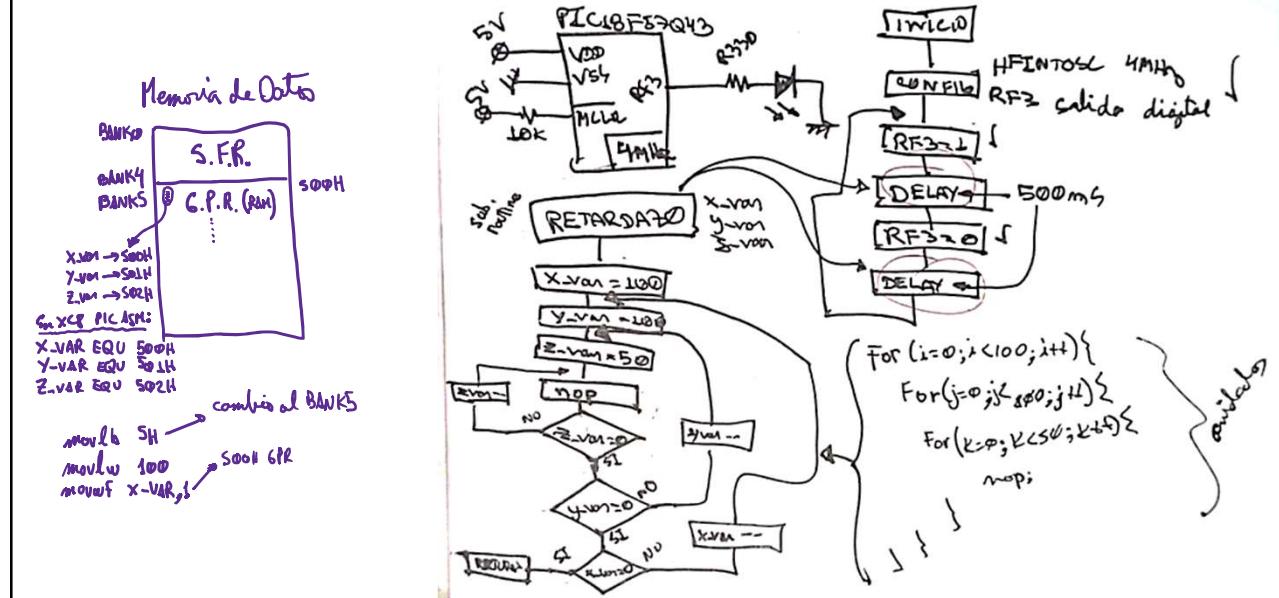
- ¿Reemplazar un switch con solamente un cable jumper?



2

## Preguntas previas

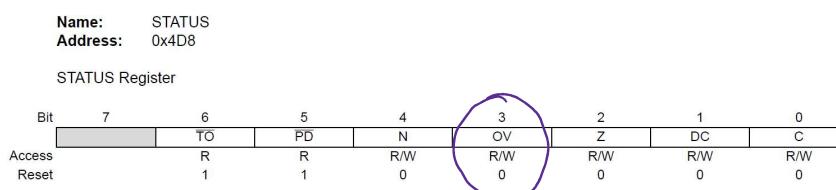
- ¿Retardo prolongado para prender y apagar un LED?



3

## Preguntas previas

- ¿Cómo se trabaja con el bit de overflow?



**cuenta EQU 500H**

```

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

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

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)

```

        movlw 55H           ;01010101B
        movwf LATB, 1
        comf LATB, 1, 1    ;10101010B
        bra $-6

bucle:   ---
retardon: ---          ;subrutina
--- 
--- 
--- 
--- 
--- 
return

```

```

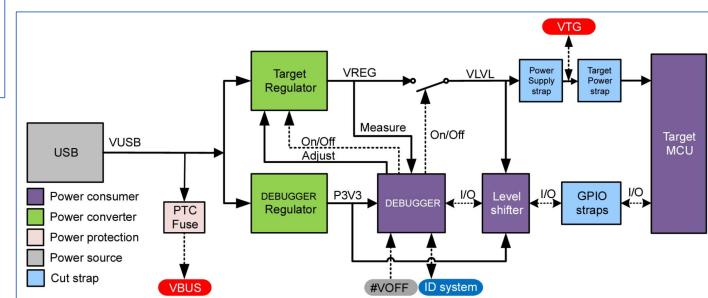
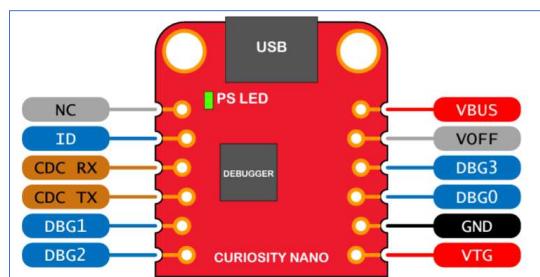
inicio:
btfs PORTB, 4, 1
bra $-2
btg LATF, 3, 1
btfc PORTB, 4, 1
bra $-2
bra inicio

```

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

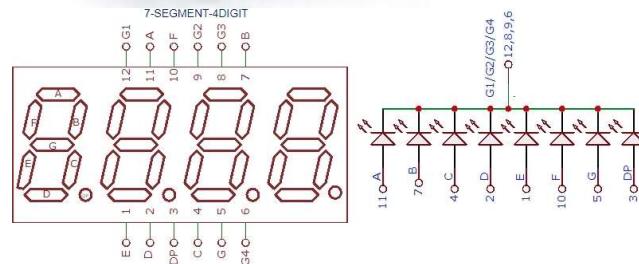
## Agenda

- 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

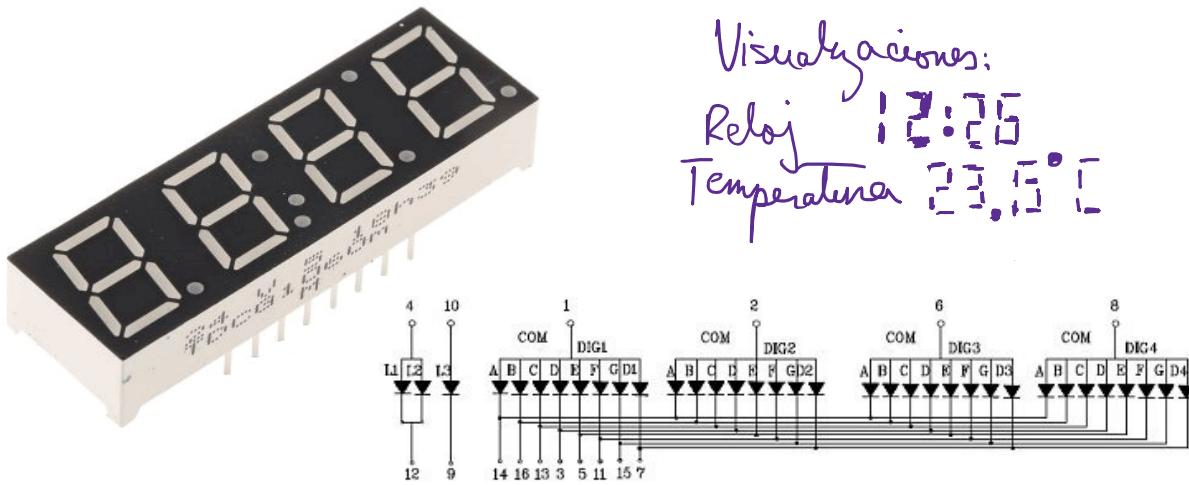
## El display de cuatro dígitos de siete segmentos multiplexado:

- 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

Ejemplo: display dedicado para aplicaciones de reloj y temperatura



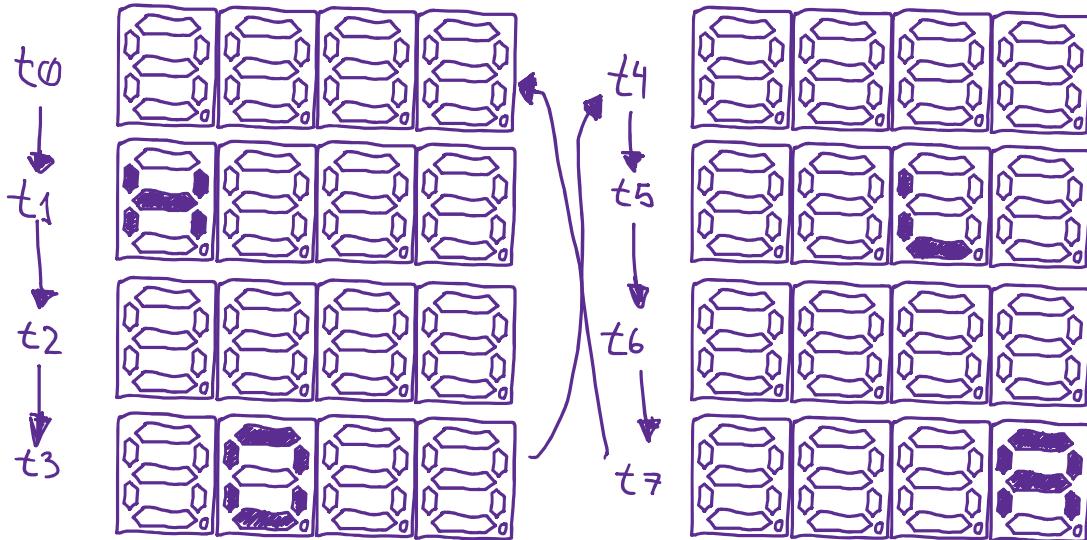
9

## 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 (refresh rate) involucra haber visualizado la información en todos los dígitos que componen el display.
- Por medio de un refresco a alta frecuencia (>50Hz) podremos ver los cuatro displays encendidos al mismo tiempo por el efecto de “persistencia visual”
- Si se detiene la multiplexación, se detiene la visualización en 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)

10

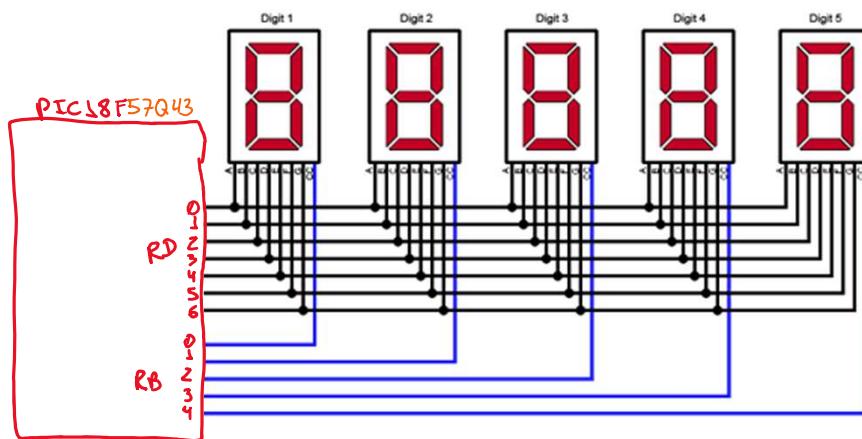
## Funcionamiento de la multiplexación: Visualización de la palabra “HOLA”



11

## 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.



12

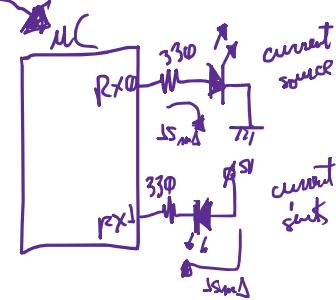
## Consideraciones eléctricas al usar el microcontrolador PIC18F57Q43

En la hoja técnica:  $\pm 50\text{ mA}$  máximo por I/O

Total el PIC18F57Q43 tiene 44 I/Os

$\cancel{\text{Total } \pm 2200\text{ mA?}}$  ¿Será cierto?

$V_{DD} \rightarrow$  máx permite  $350\text{ mA}$

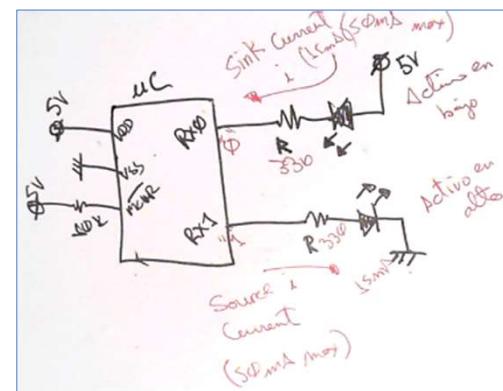


13

## ¿Corriente máxima por I/O en el PIC18F57Q43?

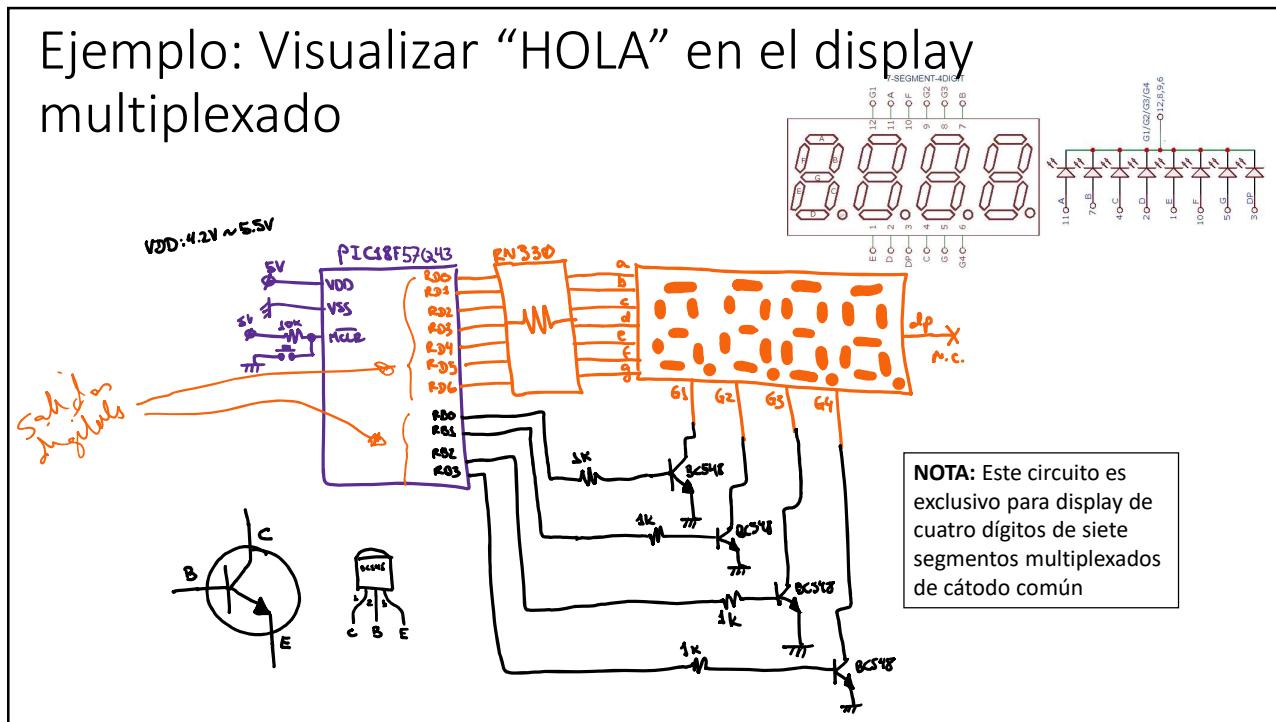
### Absolute Maximum Ratings<sup>(1)</sup>

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 $(V_{DD} + 0.3\text{ V})$
Maximum current <sup>(1)</sup>	
• on $V_{SS}$ pin	-40°C $\leq T_A \leq +85^\circ\text{C}$ 350 mA 85°C $< T_A \leq +125^\circ\text{C}$ 120 mA
• on $V_{DD}$ pin (28-pin devices)	-40°C $\leq T_A \leq +85^\circ\text{C}$ 250 mA 85°C $< T_A \leq +125^\circ\text{C}$ 85 mA
• on $V_{DD}$ pin (40-pin devices)	-40°C $\leq T_A \leq +85^\circ\text{C}$ 350 mA 85°C $< T_A \leq +125^\circ\text{C}$ 120 mA
• on any standard I/O pin	$\pm 50\text{ mA}$
Clamp current, $I_K$ ( $V_{PIN} < 0$ or $V_{PIN} > V_{DD}$ )	$\pm 20\text{ mA}$
Total power dissipation <sup>(2)</sup>	800 mW



14

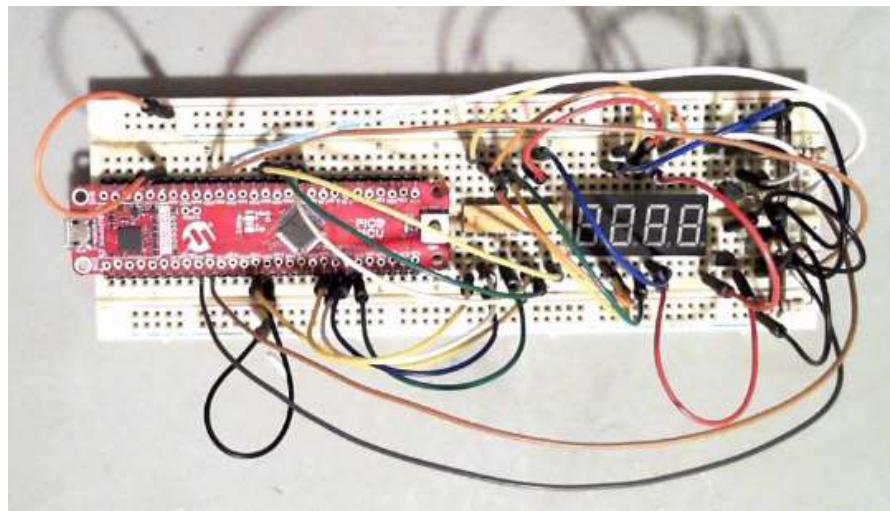
## Ejemplo: Visualizar “HOLA” en el display multiplexado



15

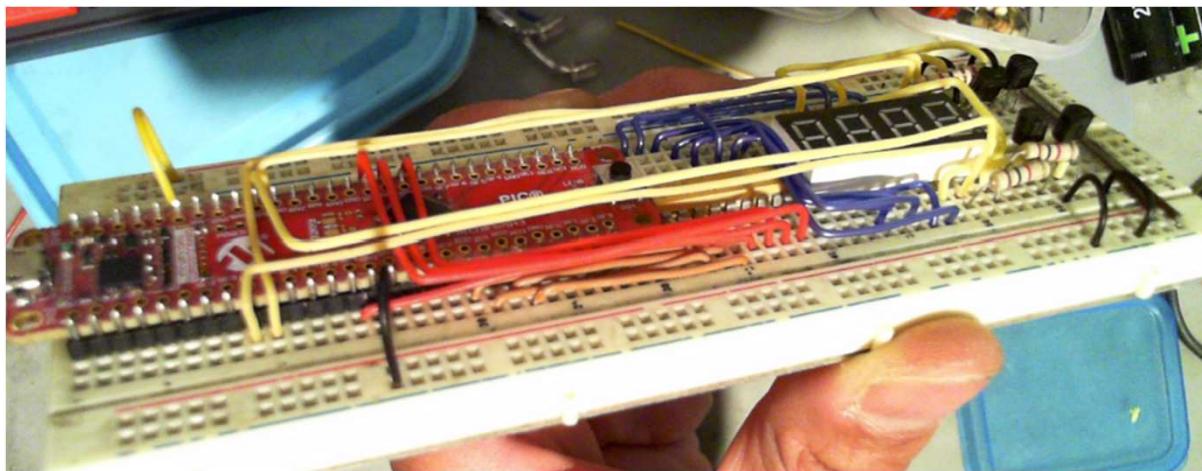
Circuito implementado con display de cuatro dígitos de siete segmentos cátodo común multiplexados

Circuito implementado



16

## Implementación 2024-1



17

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
-------------------------------	-------------------------------	-------------------------------	-------------------------------

Save Frame | Playback Animation | Clear | Frame = 0 | Delay = 100 | HGFEDCBA

**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**

{ 0x00, 0x00, 0x00, 0x00 }, // Frame 0	Copy Code	Clear Data	Format Data	Example
--	-----------	------------	-------------	---------

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

18

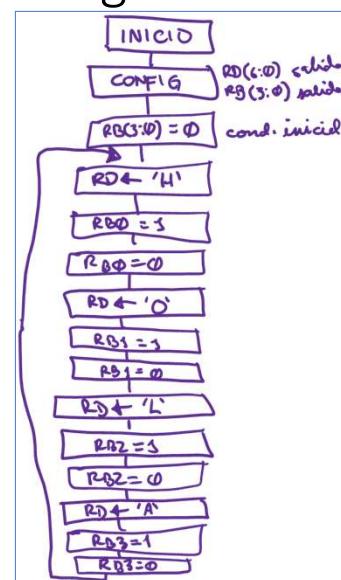
Possible visualización de las letras en el display de siete segmentos



19

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.



## Código en XC8 PIC Assembler

```

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 OSCFRQ, 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 OF0H
23   movwf TRISB, 1      ;RB3-RB0 como salidas
24   movwf ANSELB, 1      ;RB3-RB0 como digitales
25   clrf LATB, 1        ;Condicion inicial: habilitadores todos en cero
26
27 inicio:
28   movlw 76H
29   movwf LATD, 1        ;cargando H en RD
30   bsf LATB, 0, 1       ;habilito digito 0
31   call nopx16
32   bcf LATB, 0, 1       ;deshabilito digito 0
33   movlw 3FH
34   movwf LATD, 1        ;cargando O en RD
35   bsf LATB, 1, 1       ;habilito digito 1
36   call nopx16
37   bcf LATB, 1, 1       ;deshabilito digito 1
38   movlw 38H
39   movwf LATD, 1        ;cargando L en RD
40   bsf LATB, 2, 1       ;habilito digito 2
41   call nopx16
42   bcf LATB, 2, 1       ;deshabilito digito 2
43   movlw 77H
44   movwf LATD, 1        ;cargando A en RD
45   bsf LATB, 3, 1       ;habilito digito 3
46   call nopx16
47   bcf LATB, 3, 1       ;deshabilito digito 3
48   bra inicio
49 nopes:
50   nop
51   nop
52   nop
53   nop
54   nop
55   nop
56   return
57
58 end upcino

```

21

## Código 2024-1

- Visualización de “HOLA”
- Frecuencia 8MHz al CPU
- Se evidencia “ghosting” por lo que se requerirá bajar la frecuencia de trabajo del CPU

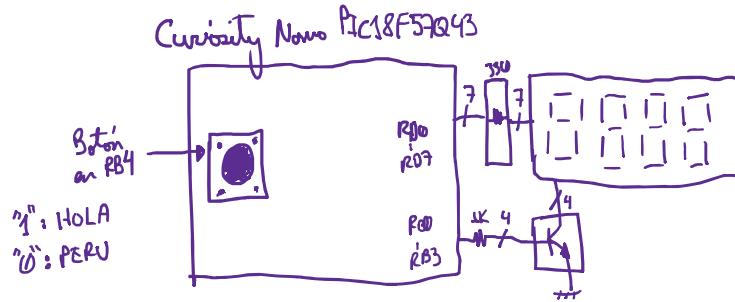
```

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 03H
15   movwf OSCFRQ, 1
16   movlw 40H
17   movwf OSCEN, 1
18   movlb 4H
19   clrf TRISD, 1
20   clrf ANSEL0, 1
21   movlw OF0H
22   movwf TRISB, 1
23   movwf ANSELB, 1
24   clrf LATB, 1
25
26 inicio:
27   movlw 76H
28   movwf LATD, 1
29   bsf LATB, 0, 1
30   call nopx16
31   bcf LATB, 0, 1
32   movlw 3FH
33   movwf LATD, 1
34   bsf LATB, 1, 1
35   call nopx16
36   bcf LATB, 1, 1
37   movlw 38H
38   movwf LATD, 1
39   bsf LATB, 2, 1
40   call nopx16
41   bcf LATB, 2, 1
42   movlw 77H
43   movwf LATD, 1
44   bsf LATB, 3, 1
45   call nopx16
46   bcf LATB, 3, 1
47   bra inicio
48 nopx16:
49   nop
50   nop
51   nop
52   nop
53   nop
54   nop
55   nop
56   nop
57   nop
58   nop
59   nop
60   nop
61   nop
62   nop
63   nop
64   nop
65   nop
66   return
67
68 end upcino

```

22

## ¿Cómo hacer para que con el pulsador integrado en RC4 se cambie de HOLA a PERU?



23

### Código 2024-1

- El botón integrado del Curiosity Nano (RB4) se está utilizando para cambiar la visualización entre la palabra HOLA y PERU.
- Tener en cuenta que se está empleando dos sectores de la memoria de programa para almacenar las palabras a visualizar, y se utiliza el puntero de tabla (TBLPTR) para poder acceder a los datos de decodificación

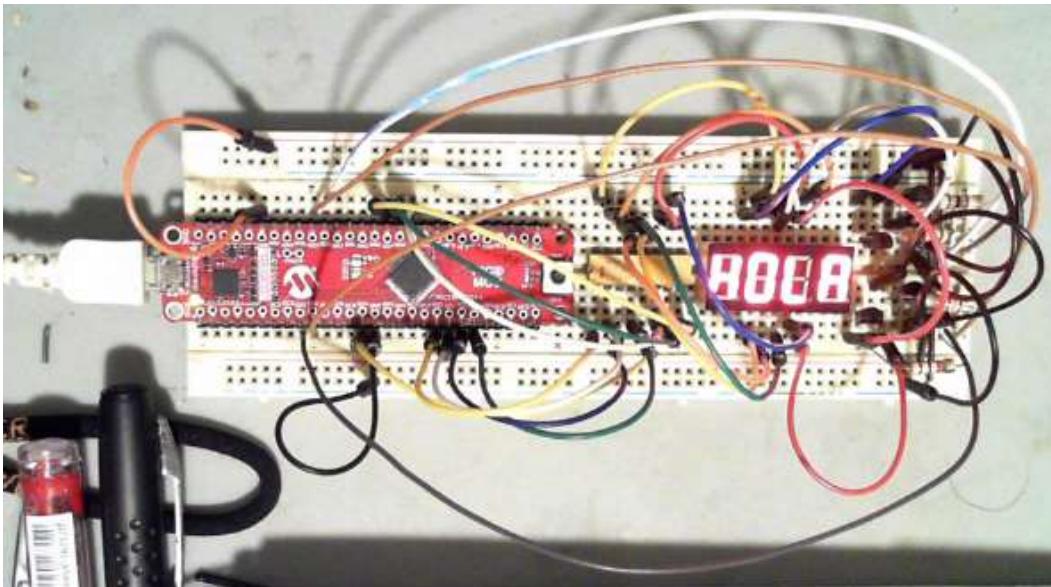
```

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 000500H
10         tabla_mensaje1: DB 73H, 79H, 50H, 3EH ;PERU
11
12         ORG 000600H
13         tabla_mensaje2: DB 76H, 3FH, 38H, 77H ;HOLA
14
15         ORG 000100H
16         configuro:
17             movlw 0H
18             movlw 60H
19             movwf OSCCON1, 1
20             movlw 00H
21             movwf OSCFRC, 1
22             movlw 40H
23             movwf OSCEN, 1
24             movlw 4H
25             clrf TRISD, 1
26             clrf ANSELD, 1
27             movlw 0FOH
28             movwf TRISB, 1
29             movlw 0EOH
30             movwf ANSELB, 1
31             bsf WPUB, 4, 1
32             clrf LATB, 1
33             clrf TBLPTR, 1
34
35         inicio:
36             call pregunta_btn
37             clrf TBLPTRL, 1
38             TBLRD+
39             movff TABLAT, LATD
40             bsf LATB, 0, 1
41             call nopx16
42             bcf LATB, 0, 1
43             TBLRD+
44             movff TABLAT, LATD
45             bsf LATB, 1, 1
46             call nopx16
47             bcf LATB, 1, 1
48             TBLRD+
49             movff TABLAT, LATD
50             bsf LATB, 2, 1
51             call nopx16
52             bcf LATB, 2, 1
53             TBLRD+
54             movff TABLAT, LATD
55             bsf LATB, 3, 1
56             call nopx16
57             bcf LATB, 3, 1
58             bra inicio
59
60         pregunta_btn:
61             btfsc PORTB, 4, 1
62             bra msg_hola
63             bra msg_peru
64
65         msg_hola:
66             movlw 06H
67             movwf TBLPTRH, 1
68
69         return
70
71         msg_peru:
72             movlw 05H
73             movwf TBLPTRH, 1
74             return
75
76         nopx16:
77             nop
78             nop
79             nop
80             nop
81             nop
82             nop
83             nop
84             nop
85             nop
86             nop
87             nop
88             nop
89             nop
90             nop
91             nop
92             return
93
94         end upcino

```

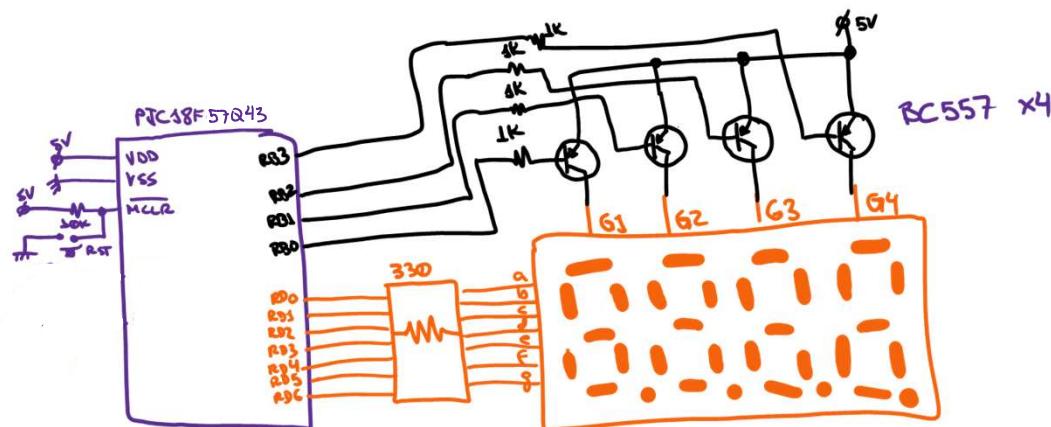
24

## Pruebas en circuito:



25

Observación: Circuito para display multiplexado de ánodo común



26

## Código mejorado empleando TBLPTR (XC8 PIC Assembler)

```

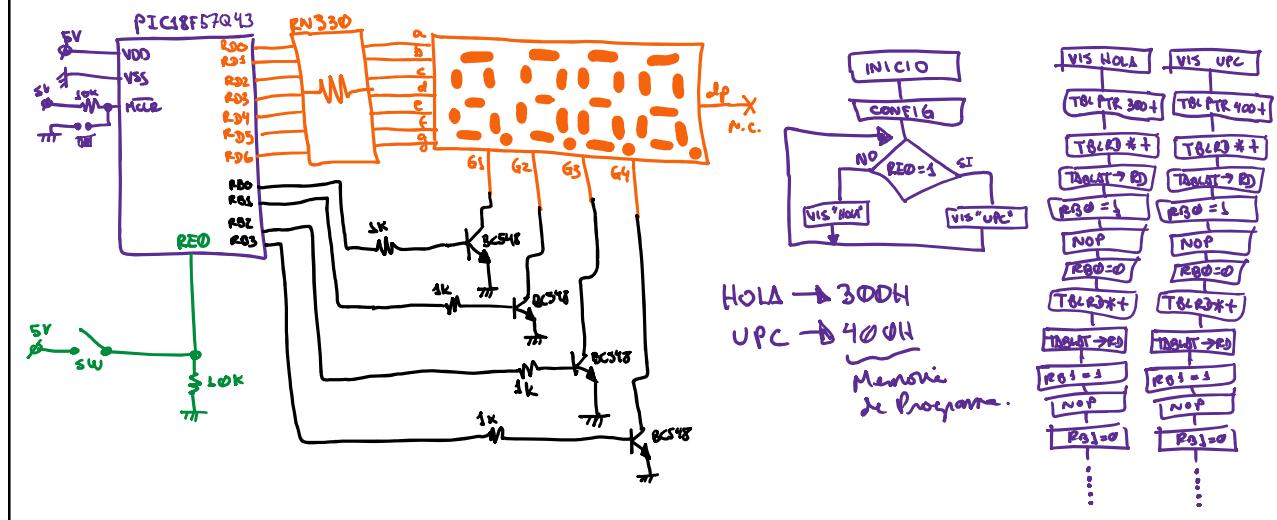
1      PROCESSOR 18F57Q43
2      #include "cabecera.inc"
3
4      PSECT upcino, class=CODE, reloc=2, abs
5      upcino:
6          ORG 000000H
7          bra configuro
8
9          ORG 000700H
10         mensaje: DB 76H, 3FH, 38H, 77H
11
12         ORG 000100H
13 configuro:
14         movlb 0H
15         movlw 60H
16         movwf OSCCON1, 1
17         movlw 02H
18         movwf OSCFRQ, 1
19         movlw 40H
20         movwf OSCEN, 1
21         movlb 4H
22         movlw 80H
23         movwf TRISD, 1 ;RD6-RD0 como salidas
24         movwf ANSELD, 1 ;RD6-RD0 como digitales
25         movlw OFOH
26         movwf TRISB, 1 ;RB3-RB0 como salidas
27         movwf ANSELB, 1 ;RB3-RB0 como digitales
28         clrf LATB, 1 ;Condicion inicial: habilitadores todos en cero
29
30         inicio:           clrf TBLPTRU, 1
31             movlw 07H
32             movwf TBLPTRH, 1
33             clrf TBLPTRL, 1 ;TBLPTR esta apuntando a la direccion 700H de mem prog
34             TBLRD*+
35             movff TABLAT, LATD ;Lee el contenido de TBLPTR (700H) y luego lo incrementa
36             bcf LATB, 0, 1 ;Mueve el contenido de TABLAT hacia LATD
37             call nopes
38             bcf LATB, 0, 1 ;habilito digito 0
39             TBLRD*+
40             movff TABLAT, LATD ;Lee el contenido de TBLPTR (701H) y luego lo incrementa
41             bcf LATB, 1, 1 ;Mueve el contenido de TABLAT hacia LATD
42             call nopes
43             bcf LATB, 1, 1 ;habilito digito 1
44             TBLRD*+
45             movff TABLAT, LATD ;Lee el contenido de TBLPTR (702H) y luego lo incrementa
46             bcf LATB, 2, 1 ;Mueve el contenido de TABLAT hacia LATD
47             call nopes
48             bcf LATB, 2, 1 ;habilito digito 2
49             TBLRD*+
50             movff TABLAT, LATD ;Lee el contenido de TBLPTR (703H) y luego lo incrementa
51             bcf LATB, 3, 1 ;Mueve el contenido de TABLAT hacia LATD
52             call nopes
53             bcf LATB, 3, 1 ;habilito digito 3
54             bra inicio
55
56         nopes:           nop
57             nop
58             nop
59             nop
60             nop
61             nop
62             nop
63             return
64
65         end upcino

```

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

27

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



28

## 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
69     end multiplexacion

```

29

## 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 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 TRISB, 1    ;RD6:RDO como salidas
28     movwf ANSELB, 1   ;RD6:RDO como digital
29     movlw 0FOH
30     ;11110000B
31     movwf TRISB, 1    ;RB3:RB0 como salidas
32     movwf ANSELB, 1   ;RB3:RB0 como digital
33     bcf ANSELB, 4, 1  ;RB4 como entrada
34     bcf WPUB, 4, 1   ;RB4 como digital
35     clrf LATB, 1      ;condicion inicial
36
37     inicio:
38     btfsc PORTB, 4, 1 ;pregunto si
39     bra noperacion
40     bra siperacione
41
42     noperacion:
43     clrf TBLPTRU, 1
44     movlw 07H
45     movwf TBLPTRH, 1
46     clrf TBLPTRL, 1
47     btfsc PORTB, 4, 1 ;El puntero de
48     TBLRD*+           ;lectura del
49     movff TABLAT, LATD ;mueve el cursor
50     bsf LATB, 0, 1
51     call nopes
52     bcf LATB, 0, 1
53     TBLRD*+           ;lectura del
54     movff TABLAT, LATD ;mueve el cursor
55     bsf LATB, 1, 1
56     call nopes
57     bcf LATB, 1, 1
58     TBLRD*+           ;lectura del
59     movff TABLAT, LATD ;mueve el cursor
60     bsf LATB, 2, 1
61     call nopes
62     bcf LATB, 2, 1
63     TBLRD*+           ;lectura del
64     movff TABLAT, LATD ;mueve el cursor
65     bsf LATB, 3, 1
66     call nopes
67     bcf LATB, 3, 1
68
69     siperacione:
70     clrf TBLPTRU, 1
71     movlw 07H
72     movwf TBLPTRH, 1
73     movlw 50H
74     movwf TBLPTRL, 1
75     clrf TBLPTRU, 1
76     movlw 07H
77     movwf TBLPTRH, 1
78     clrf TBLPTRL, 1
79     btfsc PORTB, 4, 1 ;El puntero de
80     TBLRD*+           ;lectura del
81     movff TABLAT, LATD ;mueve el cursor
82     bsf LATB, 0, 1
83     call nopes
84     bcf LATB, 0, 1
85     TBLRD*+           ;lectura del
86     movff TABLAT, LATD ;mueve el cursor
87     bsf LATB, 1, 1
88     call nopes
89     bcf LATB, 1, 1
90     TBLRD*+           ;lectura del
91     movff TABLAT, LATD ;mueve el cursor
92     bsf LATB, 2, 1
93     call nopes
94     bcf LATB, 2, 1
95     TBLRD*+           ;lectura del
96     movff TABLAT, LATD ;mueve el cursor
97     bsf LATB, 3, 1
98     call nopes
99     bcf LATB, 3, 1
100    TBLRD*+           ;lectura del
101    movff TABLAT, LATD ;mueve el cursor
102    bsf LATB, 2, 1
103    call nopes
104    bcf LATB, 2, 1
105    TBLRD*+           ;lectura del
106    movff TABLAT, LATD ;mueve el cursor
107    bsf LATB, 3, 1
108    call nopes
109    bcf LATB, 3, 1
110    bra inicio
111
112    nopes:
113    nop
114
115    end upcino

```

30

## Ejemplo para visualizar HOLA y CERO con control en RB4

```

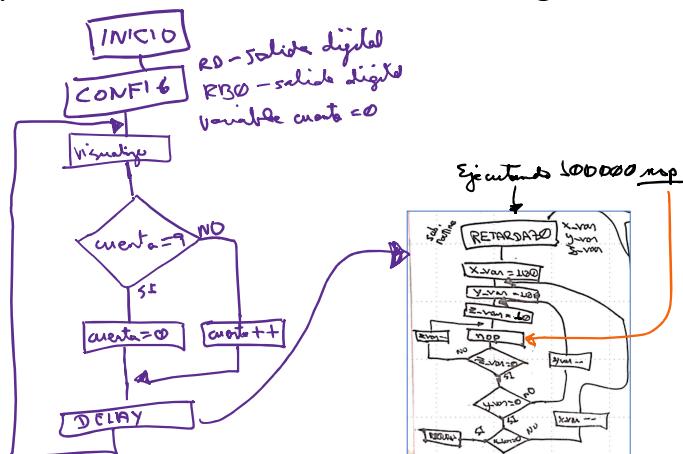
1 PROCESSOR 18F57Q43
2 #include "cabecera.inc"
3
4 PSECT upcino, class=CODE,reloc=2, abs
5 upcino:
6     ORG 000000H      ;Vector de RESET
7     bra configuro
8
9     ORG 000200H      ;Direccion de memoria de programacion
10    ;H O L A
11    mensaje1: DB 76H, 3FH, 38H, 77H
12    ORG 000220H      ;Direccion de memoria de programacion
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 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 ANSELD, 1   ;RD6-RD0 como digitales
29    movlw OF0H
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 WFUB, 4, 1    ;RB4 con pullup activado
35    clrf LATB, 1       ;Habilitadores en cero (cond 1)
36
37 inicio:
38     btfsc PORTB, 4, 1 ;Pregunto si presione el boton
39     bra noperacion
40     clrf TBLPTRU, 1    ;viene aqui cuando es falso
41     movlw 02H
42     movwf TBLPTRH, 1
43     movlw 20H
44     movwf TBLPTRL, 1   ;El TBLPTR esta mirando a direccion 22
45     bra siguiente
46
47 noperacion:
48     clrf TBLPTRU, 1
49     movlw 02H
50     movwf TBLPTRH, 1
51     clrf TBLPTRL, 1   ;El TBLPTR esta mirando a direccion 20
52 siguiente:
53     TBLRD++           ;Lectura del contenido apuntado por TBLPTR
54     movff TABLAT, LATD
55     bsf LATB, 0, 1
56     call nopes
57     bcf LATB, 0, 1
58     TBLRD++           ;Lectura del contenido apuntado por TBLPTR
59     movff TABLAT, LATD
60     bsf LATB, 1, 1
61     call nopes
62     bcf LATB, 1, 1
63     TBLRD++           ;Lectura del contenido apuntado por TBLPTR y post incremento
64     movff TABLAT, LATD
65     bsf LATB, 2, 1
66     call nopes
67     bcf LATB, 2, 1
68     TBLRD++           ;Lectura del contenido apuntado por TBLPTR y post incremento
69     movff TABLAT, LATD
70     bsf LATB, 3, 1
71     call nopes
72     bcf LATB, 3, 1
73     bra inicio
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

```

31

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

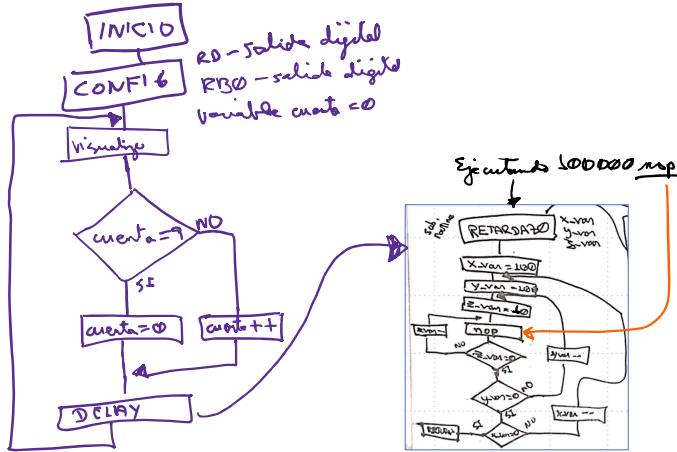
- El mismo hardware que el ejemplo anterior
  - No habrá multiplexación, se usara solamente un dígito del display



32

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

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



33

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

```

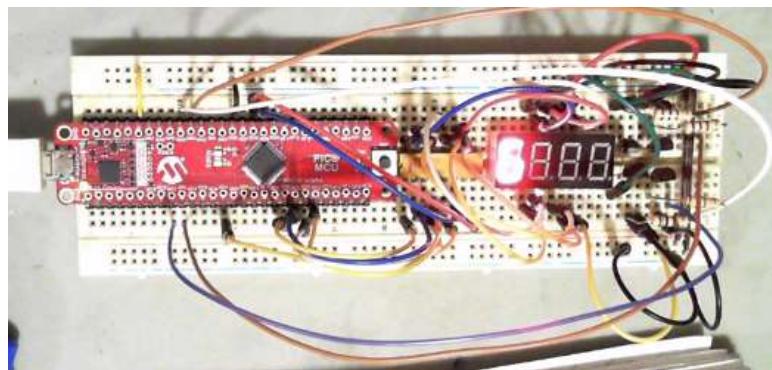
1  PROCESSOR 18F57Q43
2  #include "cabecera.inc"
3
4  PSECT upcino, class=CODE, reloc=2, abs
5
6  REG CUENTA EQU 500H ;etiqueta para la cuenta del contador
7  X_var EQU 501H
8  Y_var EQU 502H
9  Z_var EQU 503H
10
11 upcino:
12     ORG 00000H
13     bra configuro
14
15     ORG 000300H
16     ; 0 1 2 3 4 5 6 7 8 9
17 tabla_deco: DB 3FH, 06H, 5BH, 4FH, 66H, 6DH, 7DH, 07H, 7FH, 67H
18
19     ORG 000100H
20 configuro:
21     movlb 0H
22     movlw 60H
23     movwf OSCCON1, 1
24     movlw 02H
25     movwf OSCFRC, 1 ;HFINTOSC = 4MHz
26     moviw 40H
27     movwf OSCEN, 1
28     movib 5H
29     clrf REG CUENTA, 1
30     movlb 4H
31     movlw 80H
32     movwf TRISED, 1 ;RD6 al RDO como salidas
33     movwf ANSELBD, 1 ;RD6 al RDO como digitales
34     moviw 0FUH
35     movwf TRISB, 1 ;RB3 al RB0 como salidas
36     movwf ANSELB, 1 ;RB3 al RB0 como digitales
37     movlw 0BH
38     movwf LATB, 1 ;Esta habilitado el cuarto dígito del display
40     inicio:
41     movlb 4H
42     clrf TBLPTRL, 1
43     movlw 03H
44     movwf TBLPTRH, 1
45     clrf TBLPTRL, 1 ;puntero de tabla apuntando a 300H
46     movlb 5H
47     movwf REG CUENTA, 0, 1 ;recabo cuenta
48     movlw 4H
49     movwf TBLPTRL, 1 ;mueve 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

```

34

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

- Implementación



35

## 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.

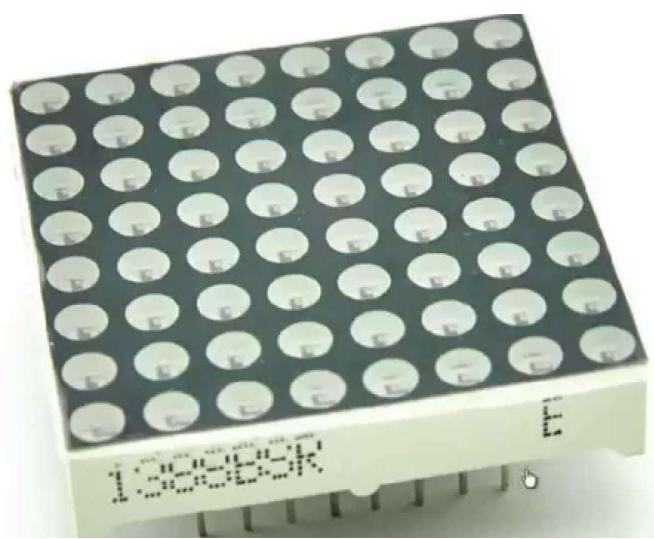
36

## Displays alfanuméricos



37

## Displays matriciales 8x8



38

Fin de la sesión