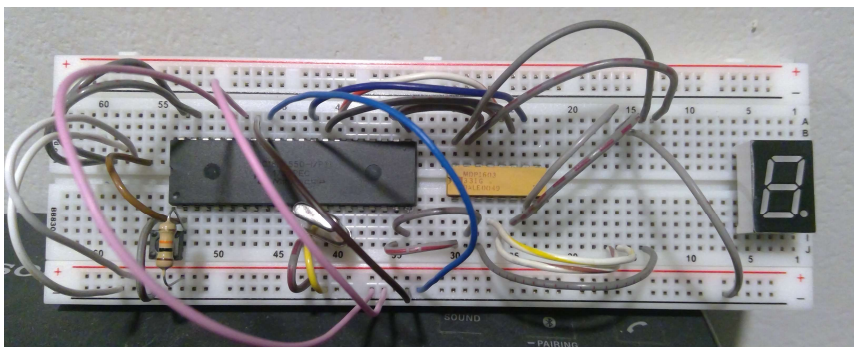


Microcontroladores

Laboratorio
Semana 2

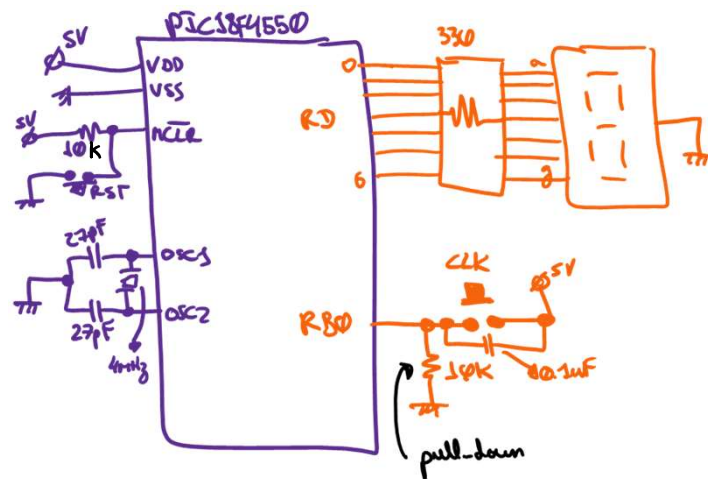
1

Implementación del decodificador BCD-7segmentos



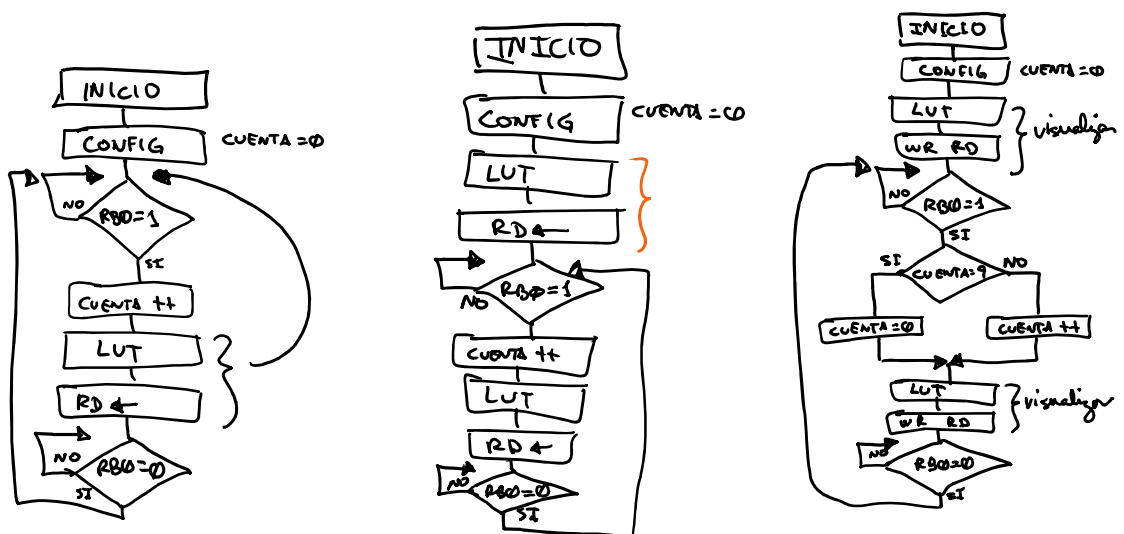
2

Ejemplo: Contador 0-9 visualizado en display de siete segmentos



3

Desarrollo del diagrama de flujo



4

Código en MPASM

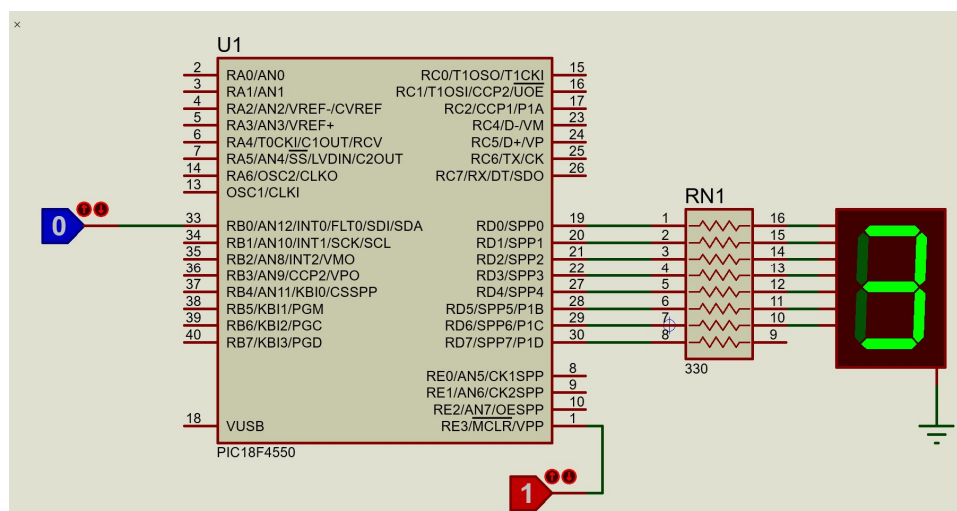
```

2 list p=18f4550 ;Modelo del microcontrolador
3 #include <pi18f4550.inc> ;Llamada a la libreria de nombre de
4
5 ;Directivas de preprocesador o bits de configuración
6 CONFIG PLLDIV = 1 ; PLL Prescaler Selection bits (
7 CONFIG CPUDIV = OSC1_PLL2 ; System Clock Postscaler Select
8 CONFIG FOSC = XT_XT ; Oscillator Selection bits (XT
9 CONFIG FWRT = ON ; Power-up Timer Enable bit (PWR
10 CONFIG BOR = OFF ; Brown-out Reset Enable bits (B
11 CONFIG WDT = OFF ; Watchdog Timer Enable bit (WDT
12 CONFIG CCP2MX = ON ; CCP2 MUX bit (CCP2 input/output
13 CONFIG PBADEN = OFF ; PORTB A/D Enable bit (PORTB<4
14 CONFIG MCLR = ON ; MCLR Pin Enable bit (MCLR pin
15 CONFIG LVP = OFF ; Single-Supply ICSP Enable bit
16
17 cblock 0x000
18 cuenta
19 miliret_a
20 miliret_b
21 endc
22
23 org 0x0500
24 tabla_7s db 0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x07, 0x7F, 0x67, 0x78, 0x79, 0x79, 0x79, 0x79, 0x79
25
26 org 0x0000
27 goto init_conf
28
29 ;Aquí se pueden declarar las constantes en la memoria de programa
30
31 org 0x0020
32 init_conf:
33 cllrf TRISD ;Todo RD como salida
34 movlw high tabla_7s
35 movwf TBLPTRH
36 movlw low tabla_7s
37 movwf TBLPTRL ;TBLPTR apuntando a 0x0500
38 cllrf cuenta
39 call visualizar ;subrutina para decodificar y visualizar
40
41 nopress:
42 btfsc PORTB, 0 ;pregunto si presione boton
43 goto nopress ;no se presiono
44 movlw .9 ;pregunta si cuenta=9
45 cpfseq cuenta
46 goto nollega ;cuenta no es 9
47 clrf cuenta ;cuenta es 9
48 goto termino
49
50 nollega:
51 inof cuenta, f
52 termino:
53 call visualizar ;retardo del rebote del boton
54 call miliretardo ;retardo del rebote del boton
55
56 nosuelta:
57 btfsc PORTB, 0 ;pregunta si soltaste boton
58 goto nosuelta ;boton no soltado
59 goto nopress ;boton soltado
60
61 visualizar:
62 movwf cuenta, TBLPTRL ;apunto
63 TBLRD*
64 movff TABLAT, LAID
65 return
66
67 miliretardo:
68 movlw .150
69 movwf miliret_a
70 otro1:
71 call anidl
72 decfsz miliret_a, f
73 goto otro1
74 anidl:
75 movlw .100
76 movwf miliret_b
77 otro2:
78 nop
79 decfsz miliret_b, f
80 goto otro2
81 return
82 end

```

5

Simulación en Proteus



6

Modificación del ejemplo anterior para que el contador sea autoincremental

```

2  list p=18f4550      ;Modelo del microcontrolador
3  #include <pi18f4550.inc> ;Llamada a la libreria
4
5  ;Directivas de preprocesador o bits de configuraci
6  CONFIG FLLDIV = 1      ; PLL Prescaler Sele
7  CONFIG CPUDIV = OSC1_PLL2 ; System Clock Postsc
8  CONFIG FOSC = XT_XT1   ; Oscillator Selectio
9  CONFIG PWRT = ON       ; Power-up Timer End
10 CONFIG BOR = OFF       ; Brown-out Reset End
11 CONFIG WDT = OFF       ; Watchdog Timer End
12 CONFIG CCF2MX = ON     ; CCF2 MUX bit (CCF2
13 CONFIG PSADEN = OFF    ; PORTB A/D Enable B
14 CONFIG MCILRE = ON     ; MCLR Pin Enable bi
15 CONFIG LVP = OFF       ; Single-Supply ICSP
16
17 cblock 0x000
18     cuenta
19     var_i
20     var_j
21     var_k
22 endc
23
24 org 0x0500
25 tabla_7s db 0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x07, 0x7F, 0x67, 0x79, 0x79, 0x79, 0x79, 0x79, 0x79
26
27 org 0x0000
28 goto init_conf
29
30 ;Aquí se pueden declarar las constantes en la memoria de programa
31
32 org 0x0020
33 init_conf:
34     clrf TRISD      ;Todo RD como salida
35     movlw high tabla_7s
36     movwf TBLPTRH
37     movlw low tabla_7s
38     movwf TBLPTRL   ;TBLPTR apuntando a 0x0500
39     clrf cuenta
40     call visualizar  ;subrutina para decodificar y visualizar
41     call delaymon
42
43 loop:
44     movlw .9        ;pregunta si cuenta=9
45     cpfseq cuenta
46     goto nollega    ;cuenta no es 9
47     clrf cuenta      ;cuenta es 9
48     goto termino
49
50 nollega:
51     incf cuenta, f
52
53 termino:
54     call visualizar
55     call delaymon
56     goto loop        ;boton soltado
57
58 visualizar:
59     movff cuenta, TBLPTRL ;apunto
60     TBLRD*
61     movff TABLAT, LATD
62     return
63
64 ;subrutina de retardo
65 delaymon:
66     movlw .50
67     movwf var_i
68
69 otro1:
70     call bucle1      ;Salto a subrutina
71     decfsz var_i, f
72     goto otro1
73     return
74
75 bucle1:
76     movlw .55
77     movwf var_j
78
79 otro2:
80     nop
81     nop
82     call bucle2      ;Salto a subrutina
83     decfsz var_j, f
84     goto otro2
85     return
86
87 bucle2:
88     movlw .20
89     movwf var_k
90
91 otro3:
92     nop
93     decfsz var_k, f
94     goto otro3
95     return
96
97 end

```

Memoria Datos

Memoria Program

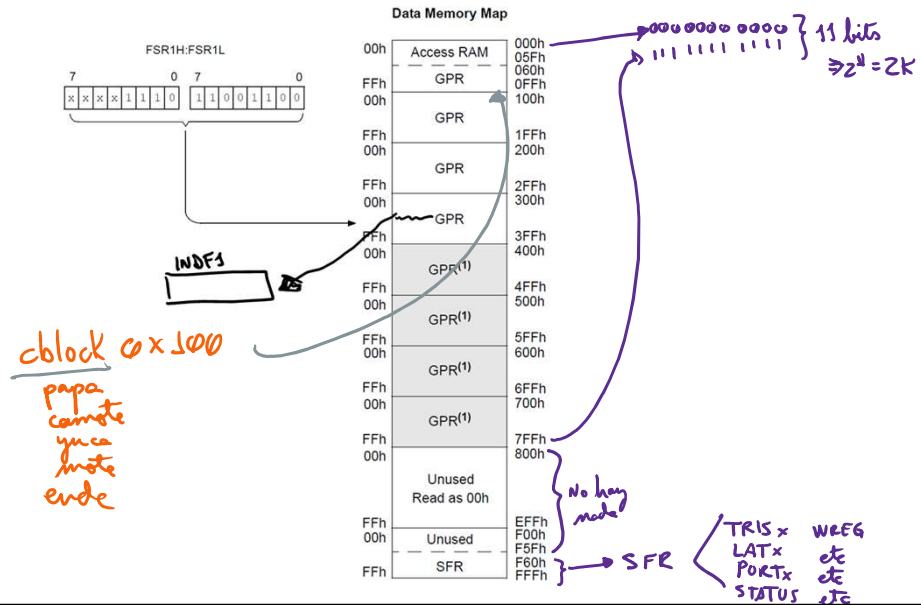
7

Manejo de la memoria de datos con punteros

- En la memoria de datos se encuentra mapeado los 2Kbyte de RAM (0x000 – 0x7FF) y los S.F.R. (0xF60 – 0xFFFF)
- El acceso a memoria mediante punteros se le denomina “direccionamiento indirecto”
- Se cuenta con tres punteros
 - FSR0 / INDF0
 - FSR1 / INDF1
 - FSR2 / INDF2
- FSRx es el registro donde se coloca la dirección de apunte
- INDx es el registro donde se aloja el contenido de la dirección apuntado por FSRx

8

Manejo de la memoria de datos con punteros



9

Ejemplos:

```

micro_delay:
    movlw .10
    movwf 0x000
bude:
    decfsz 0x000, f
    goto bude
    return
    
```

decrementa y pregunta si llega a cero

```

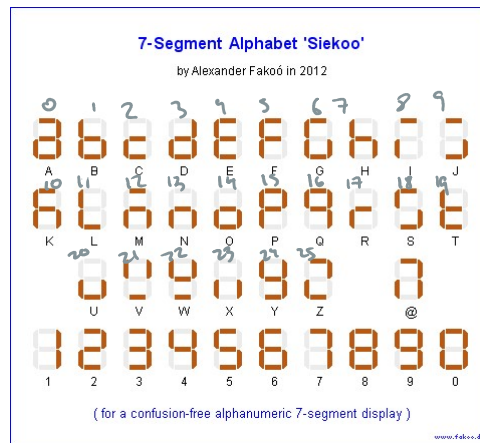
microdelay:
    movlw .10
    movwf 0x000
bude:
    decf 0x000, f
    btfss STATUS, Z
    goto bude
    return
    
```

solo decrementa

10

Ejercicio

- Obtener los números hexadecimales de todas las letras del alfabeto para ser visualizadas en un formato de display de siete segmentos cátodo común



11, 0, 20

10 minutos para
completar este tarea
(11:10)

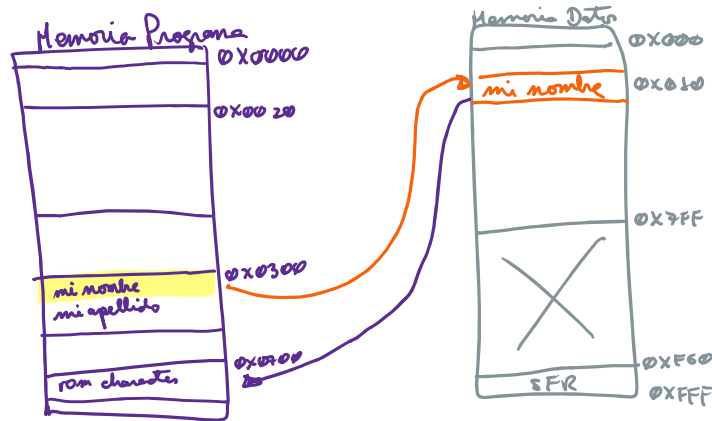
11

Ejercicio

- Los números hexadecimales obtenidos anteriormente deberán ser colocados en la posición de memoria de programa 0x0700, donde la posición 0x0700 estará la letra A, 0x0701 la letra B y así en sucesivo hasta la letra Z.
- Obtener las posiciones de las letras de tu nombre y colocarlas en la posición de memoria de programa 0x0300
- Visualizar tu apellido en el display de siete segmentos a razón de una letra a la vez con un periodo de transición de 500ms aproximadamente.
- Se debe de leer el contenido de la 0x0300 y almacenarlo temporalmente en la memoria RAM, cada dato almacenado deberá de pasar por la decodificación (datos en 0x0700) y obtener los datos a ser enviados al display de siete segmentos a través del puerto D.

12

Manipulación de las memorias



13

Código en MPASM

```

26      org 0x0700
27      ;
28      ;direccion hex  700  701  702  703  704  705  706  707  708  709  70A  70B  70C  70D  70E
29      romcharacter db 0x5F, 0x7C, 0x58, 0x5E, 0x79, 0x71, 0x3D, 0x74, 0x11, 0x0D, 0x75, 0x38, 0x55, 0x54, 0x5C,
30                      P  Q  R  S  T  U  V  W  X  Y  Z
31                      70F  710  711  712  713  714  715  716  717  718  719
32                      0x73, 0x67, 0x50, 0x2D, 0x78, 0x1C, 0x2A, 0x6A, 0x14, 0x6E, 0x1E
33
34      org 0x0300
35      minombre db 0x0A, 0x00, 0x0B, 0x14, 0x0D
36
37      org 0x0000
38      goto init_conf
39
40      org 0x0020
41      init_conf:
42      clrf TRISD
43      movlw 0x03
44      movwf TBLPTRH
45      movlw 0x00
46      movwf TBLPTRL
47      lfsr 0, 0x010
48      ;TBLPTR a
49      ;FSR0 apu
50      TBLRD*
51      movff TABLAT, INDF0
52      incf TBLPTRL, f
53      incf FSROL, f
54      TBLRD*
55      movff TABLAT, INDF0
56      incf TBLPTRL, f
57      incf FSROL, f
58      TBLRD*
59      movff TABLAT, INDF0
60      incf TBLPTRL, f
61      incf FSROL, f
62      TBLRD*
63      movff TABLAT, INDF0
64
65      loop:
66      movlw 0x07
67      movwf TBLPTRH
68      movlw 0x00
69      movwf TBLPTRL
70      lfsr 0, 0x010
71      movff INDF0, TBLPTRL
72      TBLRD*
73      movff TABLAT, LAID
74      call delaymon
75      incf FSROL, f
76      movff INDF0, TBLPTRL
77      TBLRD*
78      movff TABLAT, LAID
79      call delaymon
80      incf FSROL, f
81      movff INDF0, TBLPTRL
82      TBLRD*
83      movff TABLAT, LAID
84      call delaymon
85      incf FSROL, f
86      movff INDF0, TBLPTRL
87      TBLRD*
88      movff TABLAT, LAID
89      call delaymon
90      incf FSROL, f
91      movff INDF0, TBLPTRL
92      TBLRD*
93      movff TABLAT, LAID
94      call delaymon
95      goto loop
96
97      ;subrutina de retard
98      delaymon:
99      movlw .50
100     movwf var_i
101     otro1:
102     call bucle1
103     decfsz var_i, f
104     goto otro1
105     return
106     bucle1:
107     movlw .55
108     movwf var_j
109     otro2:
110     nop
111     nop
112     call bucle2
113     decfsz var_j, f
114     goto otro2
115     return
116     bucle2:
117     movlw .20
118     movwf var_k
119     otro3:
120     nop
121     decfsz var_k, f
122     goto otro3
123     return
124     end
125

```

1

```

;Este es un comentario, se le
list p=18f4550 ;Model
#include <18f4550.inc>

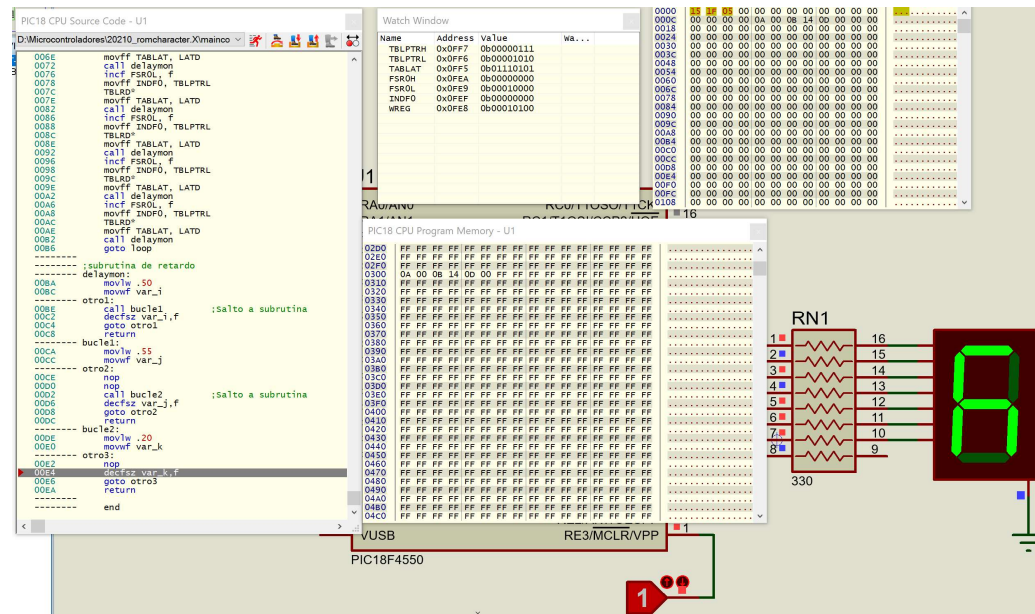
;Directivas de preprocesad
CONFIG PLLDIV = 1
CONFIG CPUDIV = OSC1_PLL2
CONFIG FOSC = XT_XT
CONFIG FWRT = ON
CONFIG BOR = OFF
CONFIG WDT = OFF
CONFIG CCP2MX = ON
CONFIG PBADEN = OFF
CONFIG MCLRE = ON
CONFIG LVP = OFF

;Aquí va el cblock o declaraac
cblock 0x000
var_i
var_j
var_k
endc

```

14

Simulación



15

Ejercicios:

- Usando el ejemplo anterior, imprimir en el display de siete segmentos tu nombre completo.
- Modificar el circuito para que se tenga cuatro displays de siete segmentos y así ampliar la visualización del mensaje con desplazamiento de derecha a izquierda
- Desarrollar un algoritmo que permita visualizar el mensaje pero de cabeza

16

Fin de la sesión