

### **Prácticas 5 y 6 - Microcontroladores PIC: Editor con teclado matricial, displays de 7 segmentos y matriz de leds**

Se trata de realizar el diseño del programa encargado de gestionar dos displays de siete segmentos y una matriz de led por barrido como dispositivos de salida y un teclado matricial 4x4 como elemento de entrada.

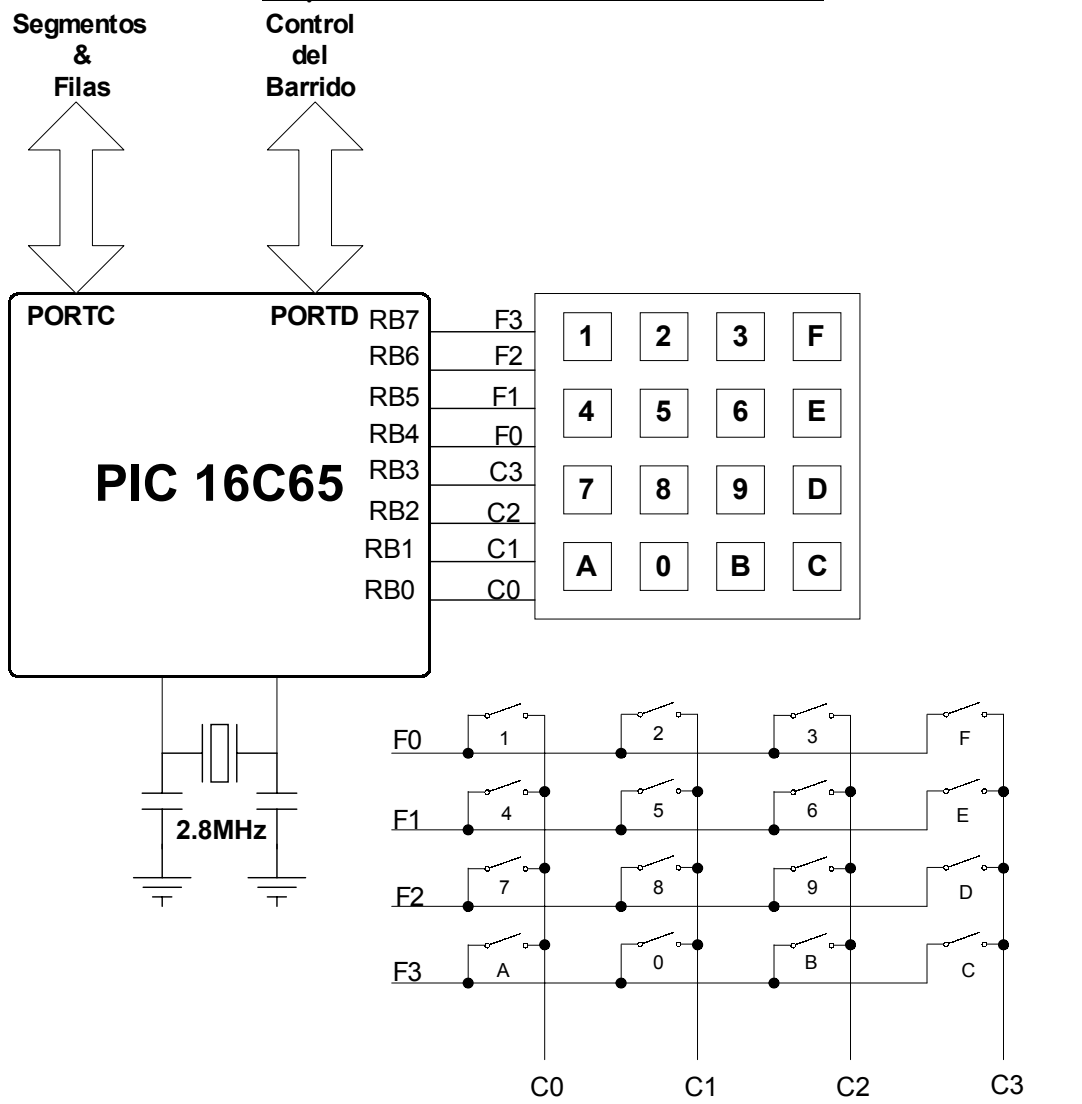
Se pueden representar los números del 0 al 9 y las letras de la A a la F que se seleccionan mediante el teclado. Inicialmente aparecerá 000 y a medida que se van pulsando las teclas, van apareciendo por la derecha los caracteres pulsados y desplazándose una posición hacia la izquierda el resto.

Se emplea una placa auxiliar que se conecta a la PICDEM2 y que tiene displays y matriz conectados en paralelo al puerto C y con control de la habilitación por medio del CI ULN2003 y el puerto D. El teclado irá conectado al PORTB según el esquema adjunto.

Se sugiere verificar el manejo del teclado por los 2 métodos conocidos: exploración secuencial y por inversión de línea.

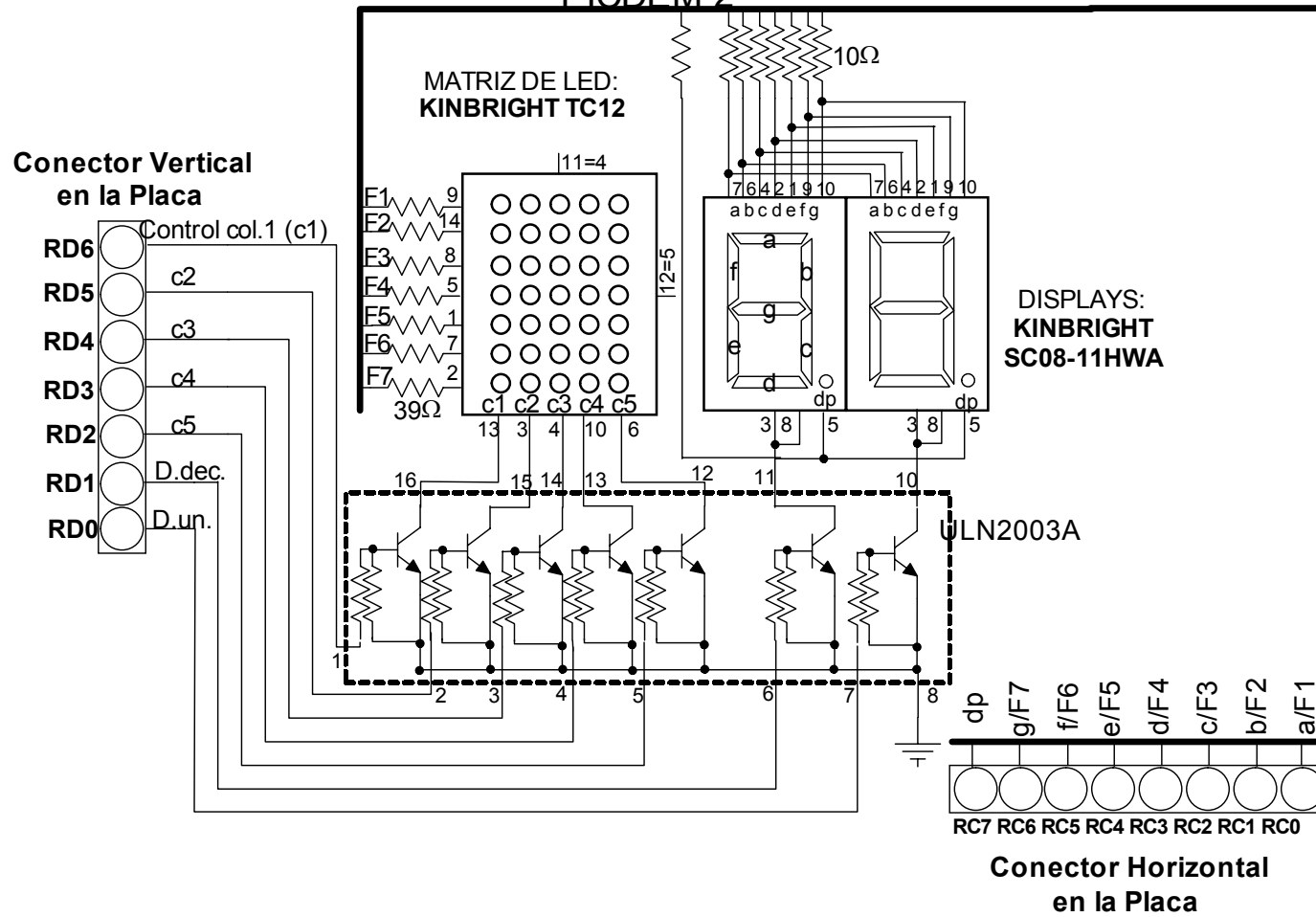
En la PICDEM2 se emplea un oscilador RC que proporciona una frecuencia de 2,8 MHz (aprox.) y el diseño se basa en el empleo de un microcontrolador PIC16C65B.

### Esquema de conexionado del teclado



Esquema de conexionado de los displays de 7 segmentos y de la matriz de leds.

PLACA DE MATRIZ DE LED Y DISPLAYS DE 7 SEGMENTOS PARA CONEXIÓN A PICDEM-2



## Programa para exploración de teclado mediante la técnica de inversión de línea (teclado.asm)

```
*****
;
; Ejemplo de manejo de dos displays de siete segmentos por barrido,
; una matriz de led que representa los números del 0 al 9 y las letras A a la F
; y de un teclado matricial de 4x4
;
; (Identificación de tecla por Inversión de Línea)
;
;
; Inicialmente aparece 000 y a medida que van pulsando las teclas, van entrando
; dígitos y letras por la derecha y desplazándose hacia la izquierda
;
; Inicialmente en el fichero teclado.asm
; Autor: Fernando Nuño García
;
;
;
; Se emplea una placa auxiliar que se conecta a la PICDEM2 y que tiene
; displays y matriz conectados en paralelo al puerto C y con control
; de la habilitación por medio del CI ULN2003 y el puerto D
; El teclado al PORTB
;
*****

list p=16c65           ;Procesador PIC16c65
include <p16c65b.inc>

DISPLAYS EQU 0x20      ;Posición de almacenamiento de displays
NUM_MTZ   EQU 0x21     ;Número a representar en la matriz de leds
AUX       EQU 0x22     ;Posición auxiliar
LEIDO     EQU 0x23     ;Posición auxiliar

COL       EQU 0x24     ;Columna a representar
ROTAR     EQU 0x25     ;Posición a rotar
W_temp    EQU 0x26     ;Salvaguarda de W en interrupción
ST_temp   EQU 0x27     ;Salvaguarda de STATUS en interrupción
AUX2      EQU 0x28

org 0           ;Vector de RESET
goto INICIO

org 4           ;Vector de interrupcion para las generadas
;por cambio en RB4 a RB7
goto TECLA

org 5

INICIO bsf STATUS,5 ;Pasamos al banco 1 de datos
movlw 0xF0    ;PORTB como entradas las altas y salidas las bajas (teclado)
movwf TRISB
clrf TRISC   ;Puerto C se programa como salida
clrf TRISD   ;Puerto D como salidas

movlw b'01000111' ;TMR0 como temporizador y prescaler de 256
movwf OPTION_REG ;asignado a TMR0 para esperas de 5ms
;el oscilador es RC y de 2.8MHz

bcf STATUS,5 ;Volvemos al banco 0 de datos

clrf PORTB   ;Sacamos ceros por las salidas del PORTB

clrf PORTC   ;Puesta a cero inicial de la salida

clrf NUM_MTZ ;Puesta a cero del contador de tiempo

clrf DISPLAYS ;los dos displays a cero también

;Interrupciones desactivadas tras el RESET, debemos programar interrupciones
;del PORTB

movlw b'10001000' ;Máscara global de interrupciones y la de
movwf INTCON      ;PORTB habilitadas
```

```

        movf    PORTB,W           ;Leemos por 1ª vez el PORTB para detectar cambios

BARR_DSP
        movlw  0x01              ;Activamos el display de unidades con
        movwf  PORTD             ;el bit RD0

        movlw  0x0F              ;Extraemos las unidades a representar
        andwf  DISPLAYS,W        ;se cargan en W y se llama al subprograma
        call  SACALED            ;que controla los 7 diodos led

        movlw  0xF2              ;Precarga en W para 5 mseg
        call  ESPERA             ;Esperaremos durante 5 mseg

        movlw  0x02              ;Activamos el display de decenas
        movwf  PORTD             ;con RD1

        movlw  0xF0              ;Extraemos las decenas a representar
        andwf  DISPLAYS,W        ;y se cargan en W

        movwf  AUX               ;se lo pasamos a traves de la
        swapf  AUX,W             ;variable AUX

        call  SACALED            ;llamada al programa de control de los led

        movlw  0xF2              ;Precarga en W para 5 mseg
        call  ESPERA             ;Retenemos durante otros 5 mseg

;Barrido de la matriz de diodos emisores de luz (LED)

        CALL  MATRIZ             ;Se llama al subprograma encargado del barrido
                                ;de la matriz de diodos led
        goto  BARR_DSP           ;y volvemos al barrido del display de 7 segmentos

;*****
;Subprograma para el control de los diodos led
;Recibe en los cuatro ultimos bits de W el digito a representar: del 0 al 9
;tambien puede ser un "blanco" (código A en hexadecimal)
;*****

SACALED  call  TABLA             ;Vamos a la tabla para volver en W con
        movwf  PORTC            ;lo que hay que sacar por el Puerto C
        return

TABLA    addwf  PCL,f            ;Suma del PC con el digito a representar (en W)
        retlw 0x3F              ;Para el cero
        retlw 0x06              ;el uno
        retlw 0x5B              ;el dos
        retlw 0x4F              ;el tres
        retlw 0x66              ;el cuatro
        retlw 0x6D              ;el cinco
        retlw 0x7D              ;el seis
        retlw 0x07              ;el siete
        retlw 0x7F              ;el ocho
        retlw 0x6F              ;el nueve
        retlw b'01110111'       ;la A
        retlw b'01111100'       ;la b
        retlw b'00111001'       ;la C
        retlw b'01011110'       ;la d
        retlw b'01111001'       ;la E
        retlw b'01110001'       ;la F

;Final del subprograma de sacar los led
    
```

```

;*****
;Subprograma de Espera que genera una pausa que depende del valor cargado
;en W, se emplea para retener la información en los displays (5mseg)
;y para hacer el barrido de las columnas de la matriz de led (1mseg)

;Emplea el TMRO que tiene asignado el prescaler de 256
;se emplea un oscilador de 2.8MHz, luego para que haya overflow al cabo de
;5mseg necesitamos contar  $5ms * 2.8MHz / (4 * 256) = 14$  pulsos (aprox), luego la
;precarga será de  $256 - 14 = 242$  que corresponde a F2 en hexadecimal

;Para que la espera sea de 1mseg, se deben contar aprox.3 pulsos, luego la
;precarga será de  $256 - 3 = 253$  que corresponde a FD
ESPERA  bcf  INTCON,T0IF      ;Ponemos a cero el flag de overflow
        movwf TMRO          ;Realizamos la precarga
NOLLEGO btfss INTCON,T0IF    ;Si se activó el flag salimos del subprograma
        goto NOLLEGO       ;si no, seguimos esperando
        return

;Final del subprograma de espera

;*****
;SUBPROGRAMA DEDICADO AL BARRIDO DE LA MATRIZ DE LED
MATRIZ  movlw 0x00           ;Inicialización de la columna
        movwf COL          ;de barrido

        movlw 0x05         ;Inicialización del contador
        movwf ROTAR        ;de rotaciones

        movlw 0x40         ;La primera columna, empezando por la izquierda corresponde a RD6
        movwf PORTD        ;y la sacamos al puerto

BARRMTZ call LEDMTZ        ;Llamamos a la tabla de sacar los led, se
                            ;retorna con el contenido en W y luego
                            ;lo sacamos por el PORTC
        movwf PORTC        ;Cargamos W para llamar a ESPERA de 1mseg
        call ESPERA        ;Aumentamos el número de la columna a barrer
        incf COL            ;Activamos la siguiente columna
        rrf PORTD           ;Decrementamos el contador de rotaciones
        decfsz ROTAR        ;Si no lo hemos hecho 5 veces repetimos
        goto BARRMTZ       ;Si no lo hemos hecho 5 veces repetimos
        return

;Salimos del subprograma de barrido de la matriz de led

;*****
;Subprograma que, a partir del dato (en posición TIEMPO) y de la columna que
; toca representar, saca los led a iluminar
LEDMTZ  movf NUM_MTZ,W      ;Se toma el número a representar
        movwf AUX          ;y se le pasa a posición AUX
        bcf STATUS,C        ;Puesto a cero del carry
        rlf AUX             ;Nos desplazamos en la tabla multiplicando
        rlf AUX             ;por 5 el dato y sumamos el número de la
        addwf AUX,W         ;columna a representar
        addwf COL,W         ;a continuación
        addwf PCL,f         ;Sumamos a PC y volvemos con W conteniendo
                            ;la información sobre los displays a iluminar

;Tabla para que la fila superior se controle con RC0 y así sucesivamente
;hasta que la fila inferior se controle con RC6
ELCERO  retlw b'00111110'
        retlw b'01000001'
        retlw b'01000001'
        retlw b'01000001'
        retlw b'00111110'
ELUNO   retlw b'01000100'
        retlw b'01000010'
        retlw b'01111111'
        retlw b'01000000'
        retlw b'01000000'

```

```
ELDOS  retlw b'01100010'  
        retlw b'01010001'  
        retlw b'01001001'  
        retlw b'01000101'  
        retlw b'01000010'  
ELTRES retlw b'00100010'  
        retlw b'01000001'  
        retlw b'01001001'  
        retlw b'01001001'  
        retlw b'00110110'  
ELCUATRO  
        retlw b'00011000'  
        retlw b'00010100'  
        retlw b'00010010'  
        retlw b'01111111'  
        retlw b'00010000'  
ELCINCO  
        retlw b'00100111'  
        retlw b'01000101'  
        retlw b'01000101'  
        retlw b'01000101'  
        retlw b'00111001'  
ELSEIS  retlw b'00111100  
        retlw b'01001010  
        retlw b'01001001  
        retlw b'01001001  
        retlw b'00110010  
ELSIETE retlw b'00000001'  
        retlw b'00000001'  
        retlw b'01111001'  
        retlw b'00000101'  
        retlw b'00000011'  
ELOCHO  retlw b'00110110'  
        retlw b'01001001'  
        retlw b'01001001'  
        retlw b'01001001'  
        retlw b'00110110'  
ELNUEVE  
        retlw b'00000110'  
        retlw b'01001001'  
        retlw b'01001001'  
        retlw b'00101001'  
        retlw b'00011110'  
LA_A    retlw b'01111100'  
        retlw b'00010010'  
        retlw b'00010001'  
        retlw b'00010010'  
        retlw b'01111100'  
LA_B    retlw b'01111111'  
        retlw b'01001001'  
        retlw b'01001001'  
        retlw b'01001001'  
        retlw b'00110110'  
LA_C    retlw b'00111110'  
        retlw b'01000001'  
        retlw b'01000001'  
        retlw b'01000001'  
        retlw b'00100010'  
LA_D    retlw b'01111111'  
        retlw b'01000001'  
        retlw b'01000001'  
        retlw b'00100010'  
        retlw b'00011100'  
LA_E    retlw b'01111111'  
        retlw b'01001001'  
        retlw b'01001001'  
        retlw b'01001001'  
        retlw b'01000001'  
LA_F    retlw b'01111111'  
        retlw b'00001001'  
        retlw b'00001001'  
        retlw b'00001001'  
        retlw b'00000001'
```

;Final de la tabla de numeros y letras por columnas

```
*****
;Programa de tratamiento de la interrupción generada por PORTB,
;se producira en cada cambio de RB4 a RB7
*****
TECLA   btfss INTCON,RBIF           ;Si entramos aquí por otro motivo
        retfie                       ;distinto a RBIF=1 salimos de inmediato

        movwf W_temp                 ;Salvamos el registro W
        swapf STATUS,W              ;y el registro STATUS por
        movwf ST_temp                ;si acaso se manipulan para el principal

;Espera para evitar rebotes (20ms)
        movlw d'200'                 ;Se debe precargar 200 en TMR0
        call ESPERA                  ;llamamos al subprograma de ESPERA

        movlw 0xF0                   ;Comprobamos los 4 bits más altos
        xorwf PORTB,W                ;si están a uno todos, nos salimos
        btfsc STATUS,Z
        goto SALIDA

;Almacenamos el PORTB

        movf PORTB,W
        movwf LEIDO

;Cambiamos entradas y salidas

        bsf STATUS,RP0
        movlw b'00001111'
        movwf TRISB
        bcf STATUS,RP0

;Sacamos al PORTB la lectura anterior

        movf LEIDO,W
        movwf PORTB
        nop
        nop

;Exploramos qué línea está a cero para identificar la columna
vercolumna
        btfss PORTB,0
        goto COL0

        btfss PORTB,1
        goto COL1

        btfss PORTB,2
        goto COL2

        btfss PORTB,3
        goto COL3

;Identificación de columna y codificación

COL0    clrw
        goto verfila
COL1    movlw 0x01
        goto verfila
COL2    movlw 0x02
        goto verfila
COL3    movlw 0x03

verfila btfss PORTB,4
        goto FILA0

        btfss PORTB,5
        goto FILA1

        btfss PORTB,6
        goto FILA2

        btfss PORTB,7
        goto FILA3
```

```

FILA0   goto codigo
FILA1   iorlw b'00000100'
        goto codigo
FILA2   iorlw b'00001000'
        goto codigo
FILA3   iorlw b'00001100'

codigo  call busca                ;Llamamos a busca llevando en W fila y columna
                                           ;volvemos trayendo en W el código de la tecla

;Desplazamos todo introduciendo nuevo codigo

        movwf AUX2                ;Guardamos código nuevo de tecla
        swapf DISPLAYS            ;Intercambiamos dígitos de DISPLAY
        movf DISPLAYS,W          ;Extraemos el dígito que pasa a la matriz
        andlw 0x0F                ;lo tenemos en W
        movwf NUM_MTZ            ;Se lo pasamos a la posición de la matriz
        movlw 0xF0                ;Limpiamos el dígito que desaparece
        andwf DISPLAYS,f         ;de los displays
        movf AUX2,W              ;Introducimos el código que habíamos almacenado
        addwf DISPLAYS,F

        bsf STATUS,RP0
        movlw b'11110000'        ;Dejamos las entradas y salidas como estaban
        movwf TRISB
        bcf STATUS,RP0

        clrf PORTB                ;Sacamos "ceros" por las salidas
        movf PORTB,w            ;Lo leemos

SALIDA  bcf INTCON,RBIF          ;Ponemos a cero el flag RBIF
        swapf ST_temp,W          ;Recuperamos el contexto
        movwf STATUS             ;STATUS antiguo y
        swapf W_temp,f           ;W antiguo
        swapf W_temp,W

        retfie                    ;Retorno del programa de tratamiento de
                                           ;la interrupción

;Tabla de teclas asignadas al teclado

busca   addwf PCL,f
        retlw 0x01                ;Fila 0 columna 0
        retlw 0x02                ;Fila 0 columna 1
        retlw 0x03                ;Fila 0 columna 2
        retlw 0x0F                ;Fila 0 columna 3
        retlw 0x04                ;Fila 1 columna 0
        retlw 0x05                ;Fila 1 columna 1
        retlw 0x06                ;Fila 1 columna 2
        retlw 0x0E                ;Fila 1 columna 3
        retlw 0x07                ;Fila 2 columna 0
        retlw 0x08                ;Fila 2 columna 1
        retlw 0x09                ;Fila 2 columna 2
        retlw 0x0D                ;Fila 2 columna 3
        retlw 0x0A                ;Fila 3 columna 0
        retlw 0x00                ;Fila 3 columna 1
        retlw 0x0B                ;Fila 3 columna 2
        retlw 0x0C                ;Fila 3 columna 3

*****
,
        END                        ;del fichero
    
```

## Programa para exploración de teclado mediante la técnica de exploración secuencial (teclad2.asm)

```

*****
;
;   Ejemplo de manejo de dos displays de siete segmentos por barrido,
;   una matriz de led que representa los números del 0 al 9 y las letras
;   de la A a la F como dispositivos de salida
;   y de un teclado matricial de 4x4 como elemento de entrada
;
;
;       Identificación de tecla por medio de exploración secuencial
;
;
;       Inicialmente aparece 000 y a medida que van pulsando las teclas, van entrando
;       los dígitos y las letras por la derecha y desplazándose hacia la izquierda
;
;
;       Inicialmente en el fichero teclado.asm
;       Autor: Fernando Nuño García
;
*****
;
;
;       Se emplea una placa auxiliar que se conecta a la PICDEM2 y que tiene
;       displays y matriz conectados en paralelo al puerto C y con control
;       de la habilitación por medio del CI ULN2003 y el puerto D
;       El teclado se conecta al PORTB
;
*****

                list p=16c65                ;Procesador PIC16c65
                include <p16c65b.inc>

DISPLAYS       EQU 0x20                    ;Posición de almacenamiento de displays
NUM_MTZ        EQU 0x21                    ;Número a representar en la matriz de leds
AUX            EQU 0x22                    ;Posición auxiliar
LEIDO          EQU 0x23                    ;Posición auxiliar

COL            EQU 0x24                    ;Columna a representar
ROTAR          EQU 0x25                    ;Posición a rotar
W_temp        EQU 0x26                    ;Salvaguarda de W en interrupción
ST_temp       EQU 0x27                    ;Salvaguarda de STATUS en interrupción
AUX2          EQU 0x28

                org 0                      ;Vector de RESET
                goto INICIO

                org 4                      ;Vector de interrupción para las generadas
                goto TECLA                ;por cambio en RB4 a RB7

                org 5

INICIO         bsf STATUS,5                ;Pasamos al banco 1 de datos
                movlw 0xF0                 ;PORTB como entradas las altas y salidas las bajas (teclado)
                movwf TRISB
                clrf TRISC                 ;Puerto C se programa como salida
                clrf TRISD                 ;Puerto D como salidas

                movlw b'01000111'         ;TMR0 como temporizador y prescaler de 256
                movwf OPTION_REG          ;asignado a TMR0 para esperas de 5ms
                ;el oscilador es RC y de 2.8MHz

                bcf STATUS,5                ;Volvemos al banco 0 de datos

                clrf PORTB                 ;Sacamos ceros por las salidas del PORTB

                clrf PORTC                 ;Puesta a cero inicial de la salida

                clrf NUM_MTZ                ;Puesta a cero del contador de tiempo

                clrf DISPLAYS              ;los dos displays

;Interrupciones desactivadas tras el RESET, debemos programar interrupciones
;del PORTB

                movlw b'10001000'         ;Máscara global de interrupciones y la de
                movwf INTCON              ;PORTB habilitadas
    
```

```

        movf PORTB,W           ;Leemos por 1ª vez el PORTB

BARR_DSP
    movlw 0x01                ;Activamos el display de unidades con
    movwf PORTD                ;el bit RDO

    movlw 0x0F                ;Extraemos las unidades a representar
    andwf DISPLAYS,W          ;se cargan en W y se llama al subprograma
    call SACALED              ;que controla los 7 diodos led

    movlw 0xF2                ;Precarga en W para 5 mseg
    call ESPERA               ;Esperaremos durante 5 mseg

    movlw 0x02                ;Activamos el display de decenas
    movwf PORTD                ;con RD1

    movlw 0xF0                ;Extraemos las decenas a representar
    andwf DISPLAYS,W          ;y se cargan en W

    movwf AUX                 ;se lo pasamos a través de la
    swapf AUX,W               ;variable AUX

    call SACALED              ;llamada al programa de control de los led

    movlw 0xF2                ;Precarga en W para 5 mseg
    call ESPERA               ;Retenemos durante otros 5 mseg

;Barrido de la matriz de diodos emisores de luz (LED)

    CALL MATRIZ                ;Se llama al subprograma encargado del barrido
                                ;de la matriz de diodos led
    goto BARR_DSP              ;y volvemos al barrido de los displays de 7 segmentos

;*****
;
;Subprograma para el control de los diodos led
;Recibe en los cuatro ultimos bits de W el digito a representar: del 0 al 9
;tambien puede ser un "blanco" (código A en hexadecimal)
;*****
;
SACALED
    call TABLA                 ;Vamos a la tabla para volver en W con
    movwf PORTC                ;lo que hay que sacar por el Puerto C
    return

TABLA  addwf PCL,f              ;Suma del PC con el dígito a representar (en W)
        retlw 0x3F              ;Para el cero
        retlw 0x06              ;el uno
        retlw 0x5B              ;el dos
        retlw 0x4F              ;el tres
        retlw 0x66              ;el cuatro
        retlw 0x6D              ;el cinco
        retlw 0x7D              ;el seis
        retlw 0x07              ;el siete
        retlw 0x7F              ;el ocho
        retlw 0x6F              ;el nueve
        retlw b'01110111'       ;la A
        retlw b'01111100'       ;la b
        retlw b'00111001'       ;la C
        retlw b'01011110'       ;la d
        retlw b'01111001'       ;la E
        retlw b'01110001'       ;la F

;Final del subprograma de sacar los led por los displays de 7 segmentos
    
```

```

;*****
;Subprograma de Espera que genera una pausa que depende del valor cargado
;en W, se emplea para retener la información en los displays (5mseg)
;y para hacer el barrido de las columnas de la matriz de led (1mseg)

;Emplea el TMRO que tiene asignado el prescaler de 256
;se emplea un oscilador de 2.8MHz, luego para que haya overflow al cabo de
;5mseg necesitamos contar  $5ms * 2.8MHz / (4 * 256) = 14$  pulsos (aprox), luego la
;precarga será de  $256 - 14 = 242$  que corresponde a F2 en hexadecimal

;Para que la espera sea de 1mseg, se deben contar aprox.3 pulsos, luego la
;precarga será de  $256 - 3 = 253$  que corresponde a 0xFD

ESPERA bcf INTCON,T0IF      ;Ponemos a cero el flag de overflow
        movwf TMR0          ;Realizamos la precarga
NOLLEGO
        btfss INTCON,T0IF   ;Si se activó el flag volvemos del subprograma
        goto NOLLEGO       ;si no, seguimos esperando
        return
;Final del subprograma de espera

;*****
;SUBPROGRAMA DEDICADO AL BARRIDO DE LA MATRIZ DE LED

MATRIZ movlw 0x00           ;Inicialización de la columna
        movwf COL           ;de barrido

        movlw 0x05          ;Inicialización del contador
        movwf ROTAR         ;de rotaciones

        movlw 0x40          ;La primera columna, empezando por la izquierda corresponde a RD6
        movwf PORTD         ;y la sacamos al puerto

BARRMTZ
        call LEDMTZ         ;Llamamos a la tabla de sacar los led, se
                            ;retorna con el contenido en W y luego
                            ;lo sacamos por el PORTC
        movwf PORTC         ;lo sacamos por el PORTC
        movlw 0xFD          ;Cargamos W para llamar a ESPERA de 1mseg
        call ESPERA
        incf COL             ;Aumentamos el número de la columna a barrer
        rrf PORTD           ;Activamos la siguiente columna
        decfsz ROTAR        ;Decrementamos el contador de rotaciones
        goto BARRMTZ       ;Si no lo hemos hecho 5 veces repetimos
        return

;Salimos del subprograma de barrido de la matriz de led

;*****
;Subprograma que, a partir del dato (en posición TIEMPO) y de la columna que
; toca representar, saca los led a iluminar

LEDMTZ movf NUM_MTZ,W       ;Se toma el número a representar
        movwf AUX           ;y se le pasa a posición AUX
        bcf STATUS,C        ;Puesta a cero del carry
        rlf AUX             ;Nos desplazamos en la tabla multiplicando
        rlf AUX             ;por 5 el dato y sumamos el número de la
        addwf AUX,W         ;columna a representar
        addwf COL,W         ;a continuación
        addwf PCL,f         ;Sumamos a PC y volvemos con W conteniendo
                            ;la información sobre los displays a iluminar

;Tabla para que la fila superior se controle con RC0 y así sucesivamente
;hasta que la fila inferior se controle con RC6

ELCERO retlw b'00111110'
        retlw b'01000001'
        retlw b'01000001'
        retlw b'01000001'
        retlw b'00111110'
ELUNO  retlw b'01000100'
        retlw b'01000010'
        retlw b'01111111'
        retlw b'01000000'
        retlw b'01000000'

```

```
ELDOS  retlw b'01100010'  
        retlw b'01010001'  
        retlw b'01001001'  
        retlw b'01000101'  
        retlw b'01000010'  
ELTRES retlw b'00100010'  
        retlw b'01000001'  
        retlw b'01001001'  
        retlw b'01001001'  
        retlw b'00110110'  
ELCUATRO  
        retlw b'00011000'  
        retlw b'00010100'  
        retlw b'00010010'  
        retlw b'01111111'  
        retlw b'00010000'  
ELCINCO  
        retlw b'00100111'  
        retlw b'01000101'  
        retlw b'01000101'  
        retlw b'01000101'  
        retlw b'00111001'  
ELSEIS  retlw b'00111100  
        retlw b'01001010  
        retlw b'01001001  
        retlw b'01001001  
        retlw b'00110010  
ELSIETE retlw b'00000001'  
        retlw b'00000001'  
        retlw b'01111001'  
        retlw b'00000101'  
        retlw b'00000011'  
ELOCHO  retlw b'00110110'  
        retlw b'01001001'  
        retlw b'01001001'  
        retlw b'01001001'  
        retlw b'00110110'  
ELNUEVE  
        retlw b'00000110'  
        retlw b'01001001'  
        retlw b'01001001'  
        retlw b'00101001'  
        retlw b'00011110'  
LA_A    retlw b'01111100'  
        retlw b'00010010'  
        retlw b'00010001'  
        retlw b'00010010'  
        retlw b'01111100'  
LA_B    retlw b'01111111'  
        retlw b'01001001'  
        retlw b'01001001'  
        retlw b'01001001'  
        retlw b'00110110'  
LA_C    retlw b'00111110'  
        retlw b'01000001'  
        retlw b'01000001'  
        retlw b'01000001'  
        retlw b'00100010'  
LA_D    retlw b'01111111'  
        retlw b'01000001'  
        retlw b'01000001'  
        retlw b'00100010'  
        retlw b'00011100'  
LA_E    retlw b'01111111'  
        retlw b'01001001'  
        retlw b'01001001'  
        retlw b'01001001'  
        retlw b'01000001'  
LA_F    retlw b'01111111'  
        retlw b'00001001'  
        retlw b'00001001'  
        retlw b'00001001'  
        retlw b'00000001'
```

;Final de la tabla de números y letras por columnas

;Tabla de teclas

```
busca  addwf PCL,f
      retlw 0x01      ;Fila 0 columna 0
      retlw 0x02      ;Fila 0 columna 1
      retlw 0x03      ;Fila 0 columna 2
      retlw 0x0F      ;Fila 0 columna 3
      retlw 0x04      ;Fila 1 columna 0
      retlw 0x05      ;Fila 1 columna 1
      retlw 0x06      ;Fila 1 columna 2
      retlw 0x0E      ;Fila 1 columna 3
      retlw 0x07      ;Fila 2 columna 0
      retlw 0x08      ;Fila 2 columna 1
      retlw 0x09      ;Fila 2 columna 2
      retlw 0x0D      ;Fila 2 columna 3
      retlw 0x0A      ;Fila 3 columna 0
      retlw 0x00      ;Fila 3 columna 1
      retlw 0x0B      ;Fila 3 columna 2
      retlw 0x0C      ;Fila 3 columna 3
```

```
,*****
;Programa de tratamiento de la interrupción generada por PORTB,
;se producirá en cada cambio de las entradas RB4 a RB7
,*****
```

```
TECLA  btfss INTCON,RBIF      ;Si entramos aquí por otro motivo
      retfie                  ;distinto a RBIF=1 salimos de inmediato

      movwf W_temp            ;Salvamos el registro W
      swapf STATUS,W         ;y el registro STATUS por
      movwf ST_temp          ;si acaso se manipulan para el principal
```

;Espera para evitar rebotes (20ms)

```
      movlw d'200'           ;Se debe precargar 200 en TMR0
      call ESPERA            ;llamamos al subprograma de ESPERA

      movlw 0xF0             ;Comprobamos los 4 bits más altos
      xorwf PORTB,W          ;si están a uno todos, nos salimos
      btfsc STATUS,Z
      goto SALIDA
```

;Almacenamos el PORTB

```
CI0    movlw b'11111110'     ;Sacamos un cero por la columna 0 (RB0)
      movwf PORTB
      nop                    ;Esperamos 2 ciclos para estabilización
      nop
      movf PORTB,W           ;Leemos el PORTB
      andlw 0xF0             ;Si en los 4 bits altos hay "unos" no es esa la columna
      xorlw 0xF0
      btfsc STATUS,Z
      goto CI1               ;Y nos vamos a la siguiente
      clrw                   ;Codificamos en los 4 bits más bajos de W: FILA COLUMNA
      btfss PORTB,4
      iorlw 0x00              ;Fila 0 columna 0
      btfss PORTB,5
      iorlw b'00000100'      ;Fila 1 columna 0
      btfss PORTB,6
      iorlw b'00001000'      ;Fila 2 columna 0
      btfss PORTB,7
      iorlw b'00001100'      ;Fila 3 columna 0
      goto codigo

CI1    movlw b'11111101'     ;Sacamos un cero por la columna 1 (RB1)
      movwf PORTB
      nop                    ;Esperamos 2 ciclos para estabilización
      nop
      movf PORTB,W           ;Leemos el PORTB
      andlw 0xF0             ;Si en los 4 bits altos hay "unos" no es esa la columna
      xorlw 0xF0
      btfsc STATUS,Z
      goto CI2               ;Y nos vamos a la siguiente
      movlw b'00000001'      ;Codificamos en los 4 bits más bajos de W: FILA COLUMNA
      btfss PORTB,4
      iorlw 0x00              ;Fila 0 columna 1
```

```

        btfss PORTB,5
        iorlw b'00000100'      ;Fila 1 columna 1
        btfss PORTB,6
        iorlw b'00001000'     ;Fila 2 columna 1
        btfss PORTB,7
        iorlw b'00001100'     ;Fila 3 columna 1
        goto codigo

CI2    movlw b'11111011'      ;Sacamos un cero por la columna 2 (RB2)
        movwf PORTB
        nop                    ;Esperamos 2 ciclos para estabilización
        nop
        movf PORTB,W          ;Leemos el PORTB
        andlw 0xF0            ;Extraemos los 4 bits altos
        xorlw 0xF0           ;Si en los 4 bits altos hay "unos" no es esa la columna
        btfsc STATUS,Z
        goto CI3              ;Y nos vamos a la siguiente
        movlw b'00000010'     ;Codificamos en los 4 bits más bajos de W: FILA COLUMNA
        btfss PORTB,4
        iorlw 0x00            ;Fila 0 columna 2
        btfss PORTB,5
        iorlw b'00000100'     ;Fila 1 columna 2
        btfss PORTB,6
        iorlw b'00001000'     ;Fila 2 columna 2
        btfss PORTB,7
        iorlw b'00001100'     ;Fila 3 columna 2
        goto codigo

CI3    movlw b'11110111'      ;Si llegamos aquí es que era la columna 3
        movwf PORTB          ;Sacamos un cero por la columna 3 (RB3)
        nop                    ;Esperamos estabilización
        nop
        movlw b'00000011'     ;Codificamos en los 4 bits más bajos de W: FILA COLUMNA=xx11
        btfss PORTB,4
        iorlw 0x00            ;Fila 0 columna 3
        btfss PORTB,5
        iorlw b'00000100'     ;Fila 1 columna 3
        btfss PORTB,6
        iorlw b'00001000'     ;Fila 2 columna 3
        btfss PORTB,7
        iorlw b'00001100'     ;Fila 3 columna 3

codigo  call busca           ;Llamamos a busca llevando en W fila y columna
        ;volvemos trayendo en W el código de la tecla

;Desplazamos todo introduciendo nuevo codigo
        movwf AUX2            ;Guardamos código nuevo de tecla
        swapf DISPLAYS        ;Intercambiamos dígitos de DISPLAY
        movf DISPLAYS,W      ;Extraemos el dígito que pasa a la matriz
        andlw 0x0F           ;lo tenemos en W
        movwf NUM_MTZ        ;Se lo pasamos a la posición de la matriz
        movlw 0xF0           ;Limpiamos el dígito que desaparece
        andwf DISPLAYS,f     ;de los displays
        movf AUX2,W          ;Introducimos el código que habíamos almacenado
        addwf DISPLAYS,F

SALIDA  clrf PORTB           ;Sacamos "ceros" por las salidas
        movf PORTB,w         ;Lo leemos

        bcf INTCON,RBIF      ;Ponemos a cero el flag RBIF
        swapf ST_temp,W      ;Recuperamos el contexto
        movwf STATUS         ;STATUS antiguo y
        swapf W_temp,f       ;W antiguo
        swapf W_temp,W

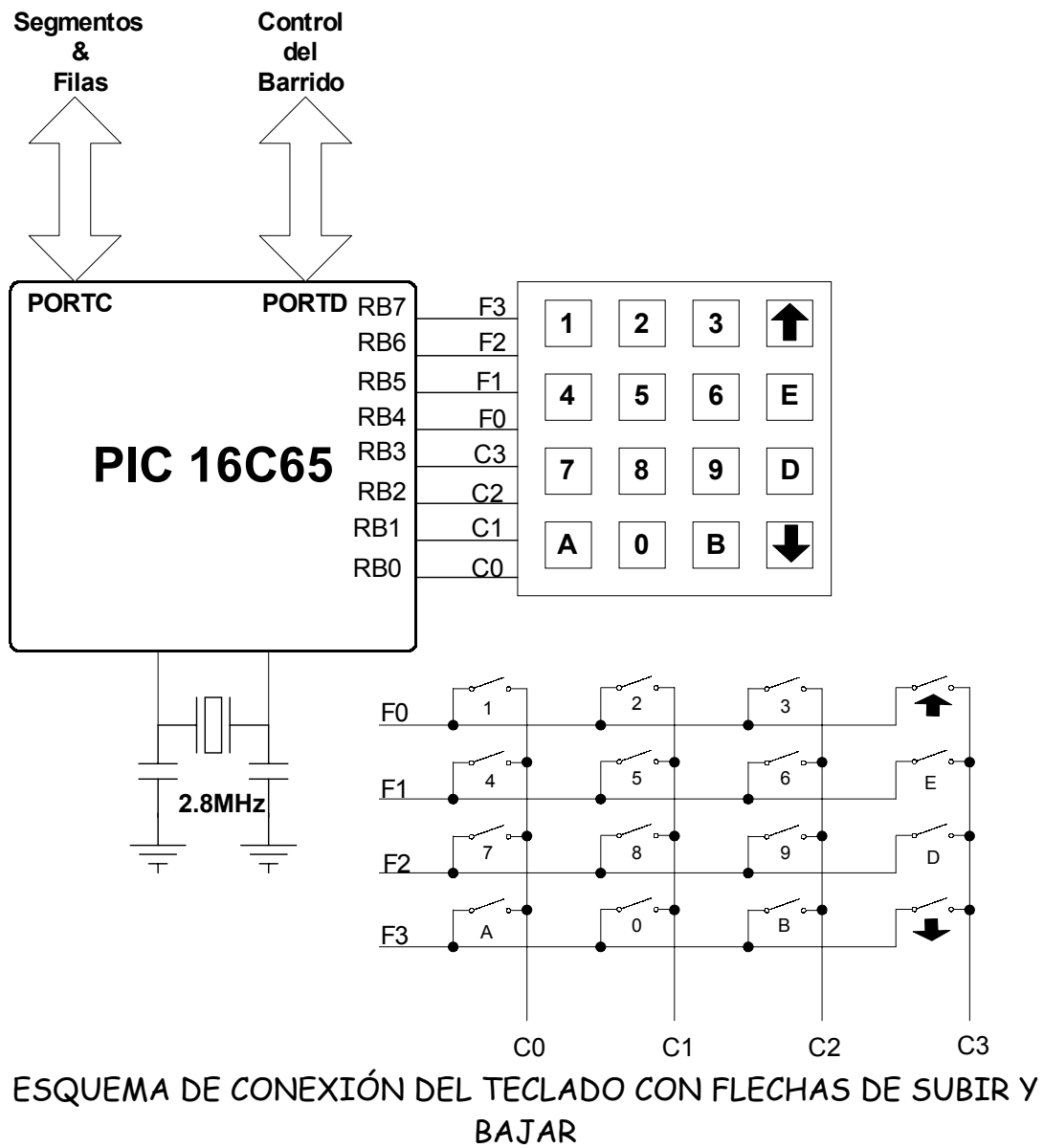
        retfie                ;Retorno del programa de tratamiento de
        ;la interrupción

;*****
;
        END                  ;del fichero
    
```

### **Trabajo de prácticas - Microcontroladores PIC: Editor con teclado matricial, displays de 7 segmentos y matriz de leds**

Como complemento a la práctica desarrollada en clase, el alumno deberá realizar un trabajo práctico que debe cumplir las siguientes especificaciones:

1. El trabajo consistirá en sacar por el conjunto de display y matriz de 7 segmentos dos indicaciones distintas: El DNI desplazándose de izquierda a derecha o una Z + dos últimos dígitos de vuestro número de matrícula parpadeando.
2. Tras el RESET del microcontrolador, la indicación activa será la del DNI desplazándose. El cambio de una indicación a otra se producirá cada vez que se pulse una tecla del teclado que no sea la flecha de subir ó la flecha de bajar.
3. El estado correspondiente a vuestro DNI desplazándose corresponde a mostrar los dígitos del DNI en paquetes de 3, tal y como se muestra en el ejemplo que se adjunta. Se empezará por los 3 dígitos menos significativos y se irá rotando a derechas, introduciendo el siguiente dígito más significativo en la matriz de leds. Al llegar el dígito más significativo, se volverán a coger los dígitos menos significativos tal y como se indica en el ejemplo adjunto. La cadencia del desplazamiento (paso de una combinación a la siguiente) será por defecto (tras un reset) de 500 mseg. No obstante está cadencia se podrá modificar entre unos límites de 200 mseg y 1000 mseg en pasos de 100 mseg según se vayan detectando pulsaciones de las teclas correspondientes a flecha subir o flecha bajar. Cada vez que se pulse flecha subir, la cadencia se incrementará en 100 mseg hasta alcanzar un límite de 1000 mseg. Si se pulsa flecha bajar, la cadencia se decrementará en 100 mseg hasta alcanzar un límite de 200 mseg. Si en cualquier momento se pulsa una tecla que no sea flecha subir o flecha bajar, la indicación cambiará a mostrar vuestro número de matrícula.
4. El estado correspondiente al número de matrícula, mostrará en la matriz de leds la letra Z y en los displays de 7 segmentos los dígitos de decenas y unidades de vuestro número de matrícula (ver figura adjunta). La indicación estará parpadeando con cadencia de 500mseg, es decir, 500 mseg los displays apagados y 500 mseg los displays mostrando parte del número de matrícula. Si en cualquier momento se pulsa una tecla que no sea flecha subir o flecha bajar, la indicación cambiará a mostrar el DNI desplazándose.





Ejemplo: N° MATRICULA ->00T040

