



MICROPROCESADORES 2003/2004

DADO ELECTRÓNICO

Jaime Pérez Crespo
Rubén Seijas Valverde

Introducción:

Con esta práctica se pretende familiarizarse con la programación en ensamblador de microcontroladores como el PIC utilizado. Se trata de la construcción de un sencillo dado electrónico mediante un PIC, un display y un pulsador, entre otros componentes.

El display, al conectarse el circuito, comenzará una cuenta desde el número 1 hasta el número 6 de forma progresiva, cambiando cada segundo. Cuando el jugador accione el pulsador, en el display se mostrarán todos los leds encendidos (salvo el punto decimal) para indicar que se está procediendo a una tirada. Al liberar el pulsador se mostrará durante un segundo el número aleatorio obtenido por el jugador. Transcurrido ese segundo, el dado volverá a retomar la cuenta exactamente en el número siguiente al último número mostrado previamente a ejecutar la tirada.

Listado:

A continuación se muestra el listado en ensamblador del programa construido para el dado electrónico.

```
;Practica del dado electronico de la asignatura
; Microprocesadores 2003-2004
;Autores:
; Jaime Perez Crespo
; Ruben Seijas Valverde
;Ultima modificacion: 3/11/2003

LIST      P=16F84
INCLUDE  "a:\P16F84.INC"  ;Libreria de registros del PIC
RADIX    HEX

SEG      EQU    0x0C
NUM      EQU    0x0D

ORG      0x00          ;Direccion de comienzo del programa
GOTO     INICIO
ORG      0x04          ;Captura de interrupciones
GOTO     INTERR        ;Salto a la rutina de tratamiento de
                        ;interrupciones

INICIO:
CLRF     SEG           ;Iniciamos la cuenta de fracciones de
                        ;segundo
CLRF     NUM           ;Iniciamos el Numero a mostrar
BSF      STATUS,RP0   ;Cambiamos al banco de memoria 1
MOVLW   0xff
MOVWF   TRISA         ;configuramos puerta A de entrada
MOVLW   0x00
MOVWF   TRISB         ;configuramos puerta B de salida
MOVLW   b'00000111'   ;cargamos los flags en OPTION
MOVWF   OPTION_REG
MOVLW   b'10100000'   ;Permitimos interrupciones en TMR0 y GIE
MOVWF   INTCON
BCF     STATUS,RP0    ;Cambiamos al banco de memoria 0
```

```

MOVLW b'00100110' ;Cargamos 38 en el TMR0 (255 - 217)
MOVFW TMR0
MOVLW b'11111111' ;Apagamos todos los leds del display
MOVWF PORTB

```

BUCLE:

```

BCF STATUS,RP0 ;Seleccionamos banco de memoria 0
BTFSS PORTA,0 ;Comprobamos si el pulsador esta
;accionado (RA0 = 0)
CALL GEN_NUM ;Pulsador accionado, generamos numero
;aleatorio
GOTO BUCLE ;No hay pulsacion, seguimos

```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Generacion del numero aleatorio ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

```

GEN_NUM:

```

BSF STATUS,RP0 ;Seleccionamos banco de memoria 1
MOVLW b'00000000' ;Deshabilitamos interrupciones
MOVWF INTCON
BCF STATUS,RP0 ;Seleccionamos banco de memoria 0
MOVLW b'10000000' ;Encendemos todos los leds del display
;salvo el punto
MOVWF PORTB

```

PULS:

```

INCF NUM,F ;Incrementamos el numero de tirada
MOVF NUM,W ;Comprobamos si la tirada es superior a 6
SUBLW 7 ;Restamos 7 a la tirada
BTFSS STATUS,Z ;Comprobamos si la resta es 0
;Si la resta anterior es 0, debemos poner
;a 1 la tirada, Z sera 1
GOTO COMP_PULS ;Tirada correcta, comprobamos el
;pulsador
MOVLW 0x01 ;Iniciamos a 1 la tirada
MOVWF NUM

```

COMP_PULS:

```

BCF STATUS,RP0 ;Seleccionamos banco de memoria 0
BTFSC PORTA,0 ;Comprobamos RA0 para ver el estado del
;pulsador
GOTO NO_PULS ;No hay pulsacion, mostramos el numero
GOTO PULS ;Pulsador accionado, seguimos generando
;el numero

```

NO_PULS:

```

CALL MOSTRAR_NUM ;Mostramos el numero en el display
;Numero mostrado, volvemos al bucle inicial
BCF STATUS,RP0 ;Cambiamos al banco de memoria 0
MOVLW b'00100110' ;Cargamos 38 en el TMR0
MOVFW TMR0
MOVLW 0x00 ;Empezamos a contar un segundo de nuevo
MOVWF SEG
MOVLW 0x00 ;Reiniciamos la cuenta desde 0
MOVWF NUM
MOVLW b'10100000' ;Permitimos interrupcion de TMR0 y la
;global GIE
MOVWF INTCON ;Habilitamos de nuevo las interrupciones
RETURN ;Retornamos de la rutina

```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Rutina para mostrar un numero en el display ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

```
MOSTRAR_NUM:
    BCF     STATUS,RP0           ;Seleccionamos el banco 0

UNO:      MOVF     NUM,W           ;Comprobamos si NUM es 1
          SUBLW   1
          BTFSS   STATUS,Z
          GOTO    DOS             ;Si no es uno
          MOVLW   b'11111001'     ;Guardamos en W las secuencia de leds
                                      ;necesaria para mostrar el numero
          GOTO    FIN

DOS:      MOVF     NUM,W           ;Comprobamos si NUM es 2
          SUBLW   2
          BTFSS   STATUS,Z
          GOTO    TRES           ;Si no es dos
          MOVLW   b'10100100'     ;Guardamos en W las secuencia de leds
                                      ;necesaria para mostrar el numero
          GOTO    FIN

TRES:     MOVF     NUM,W           ;Comprobamos si NUM es 3
          SUBLW   3
          BTFSS   STATUS,Z
          GOTO    CUATRO        ;Si no es tres
          MOVLW   b'10110000'     ;Guardamos en W las secuencia de leds
                                      ;necesaria para mostrar el numero
          GOTO    FIN

CUATRO:   MOVF     NUM,W           ;Comprobamos si NUM es 4
          SUBLW   4
          BTFSS   STATUS,Z
          GOTO    CINCO         ;Si no es cuatro
          MOVLW   b'10011001'     ;Guardamos en W las secuencia de leds
                                      ;necesaria para mostrar el numero
          GOTO    FIN

CINCO:    MOVF     NUM,W           ;Comprobamos si NUM es 5
          SUBLW   5
          BTFSS   STATUS,Z
          GOTO    SEIS          ;Si no es cinco
          MOVLW   b'10010010'     ;Guardamos en W las secuencia de leds
                                      ;necesaria para mostrar el numero
          GOTO    FIN

SEIS:     MOVLW   b'10000010'     ;Si no es ninguno de los anteriores es 6

FIN:      MOVWF   PORTB           ;Activamos las patillas del puerto B
                                      ;necesarias para mostras el NUM en el
                                      ;display
          RETURN                  ;Retornamos de la rutina
```

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Rutinas de tratamiento de interrupciones ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

INTERR:
    BCF     STATUS,RP0           ;Seleccionamos el banco 0
    BTFSS  INTCON,T0IF          ;Comprobamos el desbordamiento del TMR0
    RETFIE                                ;Sino retornamos de la interrupcion
    INCF   SEG,F                 ;Incrementamos el numero de
                                ;desbordamientos
    MOVF   SEG,W                 ;Comprobamos la fraccion de segundo
    SUBLW  d'18'                 ;Le restamos 18 (el numero de
                                ;desbordamientos por segundo)

    BTFSS  STATUS,Z             ;
    GOTO   CONT                  ;No ha transcurrido un segundo

;Ha transcurrido un segundo, actualizamos el display
UN_SEG: CLRf   SEG               ;Inicializamos SEG
        INCF  NUM,F              ;Incrementamos el numero a mostrar
        MOVF  NUM,W              ;Comprobamos si NUM es superior a 6
        SUBLW 7                  ;Restamos 7
        BTFSS STATUS,Z           ;Si la resta NO es 0:
        GOTO  OK                  ;Mostramos el numero
        MOVLW 0x01               ;En caso contrario:
        MOVWF NUM                ;Inicializamos NUM a 1

OK:     CALL  MOSTRAR_NUM        ;Lo mostramos

;Reseteamos el contador de tiempo para que vuelva a contar 1 segundo
CONT:   BCF   STATUS,Z           ;Seleccionamos el banco 0
        MOVLW b'00100110'       ;Introducimos 38 en el TMR0
        MOVWF TMR0
        MOVLW b'10100000'       ;Activamos de nuevo las interrupciones
        MOVWF INTCON
        RETFIE                    ;Retornamos de la rutina

        END

```

Rutinas utilizadas:

El funcionamiento del programa es bien sencillo. Tras realizar las tareas de inicialización pertinentes, nos introducimos en un bucle que en cada iteración comprueba si se ha accionado el pulsador. En tal caso, nos conduce a una rutina que calcula el número aleatorio.

Dicha rutina es `GEN_NUM`, durante la cual sencillamente deshabilitamos las interrupciones y esperamos en un bucle a que el jugador libere el pulsador. Mientras el pulsador esté accionado generaremos un número *aleatorio* al incrementar el número de tirada en cada iteración. Cuando el pulsador se libere, mostraremos el número de tirada que hemos obtenido durante un segundo, transcurrido el cual, restauraremos las interrupciones y volveremos al bucle principal.

Nótese que el número de tirada no se inicializa durante la rutina `GEN_NUM`, de forma que la aleatoriedad de dicho número depende no ya del tiempo que el jugador mantenga accionado el pulsador, sino también de cuál fué la última tirada.

En nuestro programa el único tratamiento de interrupciones que haremos será aquel que provenga de desbordamientos producidos en el reloj. Dicho tratamiento se realiza con la rutina `INTERR`. En la rutina mencionada, comprobaremos que la interrupción recibida sea causada por un desbordamiento en el reloj y en caso contrario descartaremos la interrupción, retornando de la rutina.

Si la interrupción ocurrida indica que el reloj se ha desbordado, incrementaremos nuestra cuenta de desbordamientos. En caso de que dicha cuenta alcance los 18 desbordamientos, habremos alcanzado un segundo de tiempo real. Incrementaremos el número de tirada que se muestra progresivamente en el display, o lo inicializaremos en caso de haber llegado a 6. Resetearemos todos los contadores para iniciar de nuevo una cuenta de un segundo.

Tanto en la rutina `GEN_NUM` como en la rutina de tratamiento de interrupciones `INTERR`, hacemos uso de otra rutina muy simple que se encarga de mostrar un número de 1 a 6 en el display. Dicha rutina es `MOSTRAR_NUM`. El funcionamiento de esta rutina es muy sencillo. Comprobaremos de forma secuencial, empezando por 1 y acabando por 5, si el número que nos indican que debemos dibujar es uno de dichos números. En tal caso, configuraremos de forma adecuada la salida a mostrar en el puerto `B` e interrumpiremos la comprobación, mostrando directamente dicha salida en el puerto y retornando de la rutina. En caso de que el número indicado no se corresponda con ninguno de los comprobados de 1 a 5, asumiremos que el número es 6 y lo mostraremos del mismo modo.

Problemas encontrados:

Estos son algunos de los problemas encontrados durante la realización de esta práctica:

Lentitud del simulador: quizás uno de los inconvenientes que más han dificultado el desarrollo ha sido el tener que lidiar con un simulador como `simupic`, que no trabaja en tiempo real. Esto retardaba las pruebas realizadas con el programa.

Dudas sobre montaje: una gran parte del tiempo con el montaje se perdió tratando de averiguar el porqué del malfuncionamiento del PIC. Dicho malfuncionamiento era debido a no haber incluido alimentación positiva conectada a la patilla `RESET` del PIC.

Cálculos del timer: una parte complicada fué obtener un número concreto para iniciar los registros `TMR0`, `prescaler`, etc, para realizar la cuenta de un segundo. Al final, tras múltiples cálculos y varias pruebas sobre el montaje, optamos por provocar que el `TMR0` se desbordase cada 38 pulsos de reloj, y contar un total de 18 desbordamientos para conseguir un segundo casi exacto.