



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE INGENIERÍA

Laboratorio de Microcomputadoras

Práctica 4.
Puertos Paralelos E/S

Grupo de Laboratorio: 4

Sanjuan Aldape Diana Paola
Grupo de Teoría: 5

Esparza Fuentes Jorge Luis
Grupo de Teoría: 4

Muñoz Tenorio Ricardo
Grupo de Teoría: 3

Fecha de realización: 8 de marzo del 2022



DESARROLLO

Para cada uno de los siguientes apartados, realizar los programas solicitados y comprobar el funcionamiento de ellos.

1.- Empleando dos puertos paralelos del microcontrolador PIC, uno de ellos configurado como entrada y el otro como salida; realizar un programa que de acuerdo al valor del bit menos significativo del puerto A, se genere la acción indicada en el puerto B.

Valor PA0	Acción puerto B
0	00000000
1	11111111

```
PROCESSOR 16F877
INCLUDE <P16F877.INC>
AUX EQU 0X20
ORG 0
GOTO INICIO
ORG 5
INICIO:
    CLRF PORTA      ;Limpia PORTA
    BSF STATUS, RP0 ;Cambia a Banco 1
    BCF STATUS, RP1
    CLRF TRISB
    MOVLW 0X07      ;Define puertos A y E como digitales
    MOVWF ADCON1
    MOVLW 0X3F      ;Configura puerto A como entrada
    MOVWF TRISA
    BCF STATUS, RP0 ;Cambia al banco 0
    MOVLW 0X00
    MOVWF PORTB     ;Pone en 0 PORTB
;CASO 0, CUANDO EL PUERTO A VALE 000000
LOOP:
    MOVF PORTA, W    ; Se guarda lo que se tiene del Puerto A
    MOVWF AUX
    MOVF AUX, F
    BTFSS STATUS, Z ;if(z==1) {Salta}
    GOTO LOOP2       ;else {se va a LOOP2}
    CALL CASO0
;CASO 1, CUANDO EL PUERTO A VALE 000001
LOOP2:
    MOVF AUX, W
    SUBLW 0X01       ;W = 0X01 - W
    BTFSS STATUS, Z ;if(z==1) {Salta}
    GOTO LOOP3       ;else {se va a LOOP3}
    CALL CASO1
LOOP3:
    GOTO LOOP
```



CASO0:

```
MOVLW 0X00  
MOVWF PORTB ;Se apagan los Leds poniendo 0 en PORTB  
RETURN
```

CASO1:

```
MOVLW 0XFF ;Se prenden todos los Leds poniendo FF en PORTB  
MOVWF PORTB  
RETURN  
END
```

2.- Escribir un programa, el cuál realice las siguientes acciones de control indicadas, para lo cuál requiere trabajar un puerto de entrada y otro puerto de salida, usar los sugeridos en el ejercicio anterior; generar retardos de ½ seg., en las secuencias que lo requieran.

DATO PUERTO A	ACCION PUERTO B	Ejecución
0x00	Todos los leds apagados	00000000
0x01	Todos los leds encendidos	11111111
0x02	Corrimiento del bit más significativo hacia la derecha	10000000 01000000 00100000 00000001
0x03	Corrimiento del bit menos significativo hacia la izquierda	00000001 00000010 00000100 10000000
0x04	Corrimiento del bit más significativo hacia la derecha y a la izquierda	10000000 01000000 00000001 00000010 10000000
0x05	Apagar y encender todos los bits.	00000000 11111111



```
processor 16f877
include<pl6f877.inc>
;Variables para el retardo
valor1 equ H'21'
valor2 equ H'22'
valor3 equ H'23'
cte1 equ H'10'
cte2 equ H'50'
cte3 equ H'60'
;Constantes para comparar la combinación entrante
c0 equ H'00'
c1 equ H'01'
c2 equ H'02'
c3 equ H'03'
c4 equ H'04'
c5 equ H'05'
org 0
goto inicio
org 5
inicio:
    clrf    PORTA        ;Limpia el puerto A
    bsf     STATUS,RP0   ;Cambio al Banco 1
    bcf     STATUS,RP1
    movlw   H'00'        ;Mueve 0h a w
    movwf   TRISE        ;Configura Puerto B como salida
    movlw   H'06'
    movwf   ADCON1       ;Configura puertos A y E como digitales
    movlw   H'3F'
    movwf   TRISA        ;Configura el Puerto A como entrada
    bcf     STATUS,RP0   ;Regresa al Banco 0
    clrf    PORTB        ;Limpia los bits de Puerto B (los apaga)
ciclo:
    ;COMBINACION PARA APAGAR LEDS
    movlw   c0           ;Mueve 0 a w
    xorwf   PORTA,w      ;Verifica si la entrada es 00
    btfsc   STATUS,Z     ;Verifica el resultado de la xor
    goto    apg          ;La combinacion es 00 y procede a la rutina
    ;COMBINACION PARA ENCENDER LEDS
    movlw   c1           ;Mueve 0 a w
    xorwf   PORTA,w      ;Verifica si la entrada es 01
    btfsc   STATUS,Z     ;Verifica el resultado de la xor
    goto    enciende     ;La combinacion es 01 y procede a la rutina
    ;COMBINACION PARA CORRIMIENTO A LA DERECHA
    movlw   c2           ;Mueve 0 a w
    xorwf   PORTA,w      ;Verifica si la entrada es 02
    btfsc   STATUS,Z     ;Verifica el resultado de la xor
    goto    derecha     ;La combinacion es 02 y procede a la rutina
    ;COMBINACION PARA CORRIMIENTO A LA IZQUIERDA
    movlw   c3           ;Mueve 0 a w
    xorwf   PORTA,w      ;Verifica si la entrada es 03
    btfsc   STATUS,Z     ;Verifica el resultado de la xor
    goto    izquierda    ;La combinacion es 03 y procede a la rutina
```



```
;COMBINACION PARA CORRIMIENTO DE DERECHA A IZQUIERDA
movlw    c4            ;Mueve 0 a w
xorwf    PORTA,w       ;Verifica si la entrada es 04
btfsc    STATUS,Z      ;Verifica el resultado de la xor
goto     der_izq       ;La combinacion es 04 y procede a la rutina
;COMBINACION PARA ENCENDIDO Y APAGADO DE LEDS
movlw    c5            ;Mueve 0 a w
xorwf    PORTA,w       ;Verifica si la entrada es 05
btfsc    STATUS,Z      ;Verifica el resultado de la xor
goto     enc_apg       ;La combinacion es 05 y procede a la rutina
goto     ciclo         ;Vuelve a preguntar por una combinacion valida
enc_apg:
movlw    h'00'         ;bits del puerto B
movwf    PORTB
call     retardo
movlw    h'FF'
movwf    PORTB
call     retardo
goto     ciclo
apg:
movlw    h'00'
movwf    PORTB
goto     ciclo
enciende:
movlw    h'FF'
movwf    PORTB
goto     ciclo
derecha:
movlw    h'80'
movwf    PORTB
call     retardo
derechal:
rrf      PORTB,1
call     retardo
btfss    PORTB,0
goto     derechal
goto     ciclo
izquierdal:
rlf      PORTB,1
call     retardo
btfss    PORTB,7
goto     izquierdal
goto     ciclo
der_izq:
movlw    h'80'
movwf    PORTB
call     retardo
derecha2:
rrf      PORTB,1
call     retardo
btfss    PORTB,0
goto     derecha2
movlw    h'01'
movwf    PORTB
call     retardo
```



```
izquierda2:
    rlf    PORTE,1
    call   retardo
    btfss  PORTE,7
    goto   izquierda2
    goto   ciclo

retardo:
    movlw  cte1          ;Rutina que genera un retardo
    movwf  valor1

tres:
    movwf  cte2
    movwf  valor2

dos:
    movlw  cte3
    movwf  valor3

uno:
    decfsz valor3
    goto   uno
    decfsz valor2
    goto   dos
    decfsz valor1
    goto   tres
    return
end
```

CONCLUSIONES

- Sanjuan Aldape Diana Paola

Con esta práctica aprendí a configurar los puertos paralelos como entrada y como salida. Para ambos casos es imprescindible limpiar el puerto usando, ya sea el mnemónico CLRFX, o bien, asignando un 0 al puerto.

- Esparza Fuentes Jorge Luis

En esta práctica se logró desarrollar soluciones a los problemas propuestos con base en el control de los puertos paralelos en la modalidad de salida y ahora también en la modalidad de entrada, a partir del datasheet del controlador se diseñaron las soluciones, de esta forma se obtuvieron los resultados esperados y de forma óptima.

- Muñoz Tenorio Ricardo

Esta práctica nos ayudó a entender mejor las configuraciones de los puertos paralelos para poder lograr llegar a la solución deseada en cada ejercicio de esta practica, aprendimos a que debemos de resetear el puerto con diferentes comandos utilizando estos en la salida y entrada a través del controlador.