



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE INGENIERÍA



LABORATORIO DE MICROCOMPUTADORAS

PRÁCTICA 3:

PUERTOS PARALELOS II (CONTROL DE ACCIONES)

GRUPO: 12

PROFESORA: M.I LOURDES ANGÉLICA QUIÑONES JUAREZ

ALUMNO: CHÁVEZ DELGADO JORGE LUIS

Nº DE CUENTA: 312217493

FECHA DE ASIGNACIÓN:

24/02/17

FECHA DE ENTREGA:

28/02/17

PRÁCTICA 1: INTRODUCCIÓN GENERAL A UN MICROCONTROLADOR

OBJETIVO: Emplear los puertos paralelos que contiene un microcontrolador para realizar funciones de control, configurando estos como entrada y salida.

Ejercicio 1:

En este ejercicio aprendimos a configurar el puerto A como entrada es decir lo relacionamos con un dip switch, y el puerto B fue configurado como salida, en este caso un led que dado un 1 en el dip, nos encendía el led y dado un 0 se mantenía apagado el led. No fue necesario el uso de retardos puesto que se utilizó el dip switch.

Código:

```
processor 16f877 ;Indica la versión del procesador
include<pl16f877.inc>
org 0 ;Para el vector reset a la dirección de inicio
goto inicio ;Salta a inicio
org 5
inicio:
    clrf  PORTB ;Limpiamos lo que haya en puerto B
    clrf  PORTA ;Limpiamos lo que haya en le puerto A
    bsf   STATUS,RP0
    bcf   STATUS,RP1 ;CAMBIAMOS A BANCO 1
    movlw H'07'
    movwf ADCON1 ; Registro para Entradas/Salidas DIGITALES
    movlw H'FF'
    movwf TRISA
    movlw H'00' ;Movemos un 0
    movwf TRISB
    bcf STATUS,RP0 ;CAMBIAMOS A BANCO 0
```

ciclo:

```
btfsc PORTA,0 ; Si es cero entonces salta a apaga  
goto enciende ;Si es 1 salta a enciende  
goto apaga
```

apaga:

```
clrf PORTB ;Limpiamos lo que hay en el puerto B  
goto ciclo ; Se cicla lo que hay en puerto B para que  
permanezca en ese estado
```

enciende:

```
movlw H'FF' ;Para el ultimo bit del puerto B  
movwf PORTB ; Movemos lo que hay en w  
goto ciclo ;Saltamos al ciclo
```

end

Ejercicio 2:

En este ejercicio creamos un “menu” de opciones controlado con el dipswitch, las opciones estan relacionadas con los siguientes valores:

#de caso	Combinación	Descripción
Caso 1	000	El LED 0 se apaga.
Caso 2	001	El LED 1 se enciende.
Caso 3	010	Comienza el corrimiento de LED's a la derecha.
Caso 4	011	Comienza el corrimiento a la izquierda.
Caso 5	100	Comienza corrimiento de derecha a izquierda.
Caso 6	101	Se encienden y apagan LED's con retardo de ½ segundo

Código:

```
processor 16f877 ;Indica la versión del procesador
include<pl6f877.inc>
contador equ h'20' ;Nos ayudara en los corrimientos
valor1 equ h'21'
valor2 equ h'22'
valor3 equ h'23'
cte1 equ 20h ;Constantes para el retardo
cte2 equ 50h
cte3 equ 60h
```

```
org 0 ;Para el vector reset a la dirección de inicio
goto inicio ;salto incondicional a inicio
org 5
inicio:
```

```

clrf  PORTB ;Limpiamos el puerto B
clrf  PORTA ;Limpiamos el puerto A
bsf      STATUS,RP0
bcf      STATUS,RP1 ;CAMBIAMOS A BANCO 1
movlw    H'07' ;Movemos un 7 a w
movwf    ADCON1 ; Registro para Entradas/Salidas DIGITALES
movlw    H'FF' ;Movemos el valor FF de la literal a w
movwf    TRISA  ;Movemos al registro TRISA Lo de w
movlw    H'00'; Movemos lo de la literal a w
movwf    TRISB  ;Movemos lo que tiene w a TRISB
bcf STATUS,RP0 ;CAMBIAMOS A BANCO 0

```

ciclo:

```

    btfsc PORTA,2 ;Preguntamos por el bit 2 del Puerto A
    goto c56  ;Para el caso 5 o 6
    goto c1234 ; Para el caso del 1 al 4

```

c1234:

```

    btfsc PORTA,1 ;Preguntamos por el bit 1 del Puerto A
    goto c34 ;Para el caso 3 y 4
    goto c12 ; Para el caso 1 y 2

```

c12:

```

    btfsc PORTA,0 ;Preguntamos por el bit 0 del puerto A
    goto c2 ;Entramos al caso 2
    goto c1 ; Pasamos al caso 1

```

c34:

```

    btfsc PORTA,0 ;Preguntamos por el bit 0 del puerto A
    goto c4 ;Pasamos al caso4
    goto c3 ; Pasamos al caso3

```

c56:

```

    btfsc PORTA,0 ;Preguntamos por el bit 0 del puerto A
    goto c6 ;Para el caso 6

```

```

        goto c5 ;Para el caso 5
c1:    ;En este caso el LED se APAGA
        clrf PORTB ;Limpiamos lo que hay en el puerto B
        goto ciclo ;Regresamos al ciclo
c2:    ;En este caso el LED se ENCIENDE
        movlw H'FF'      ;Movemos a w el valor FF
        movwf PORTB ;Moviendo el valor de w al puerto B se
encendera el led
        goto ciclo

c3:    ;En este caso se hace corrimiento a la derecha con los LEDS
        bcf STATUS,0 ;Con esto limpiamos al carry
        movlw H'80'
        movwf PORTB
        movlw H'08'
        movwf contador
loop1: ;Ciclo para el corrimiento a la derecha
        call retardo      ;Llamamos al retardo
        rrf PORTB,1        ;recorremos los bits a la derecha
        decf contador      ;Decrementamos el contador
        btfsc STATUS,2 ;Preguntamos por el bit 2 del registro status
        goto ciclo
        goto loop1

c4:    ;En este caso se hace corrimiento a la izquierda con los LEDS
        bcf STATUS,0
        movlw H'80'
        movwf PORTB
        movlw H'08'
        movwf contador
loop2:

```

```

call retardo
rlf PORTB,1
decf contador
btfsc STATUS,2
goto ciclo
goto loop2

```

c5: ;En este caso hacemos ambos corrimientos, primero a la derecha, luego a la izquierda

```

;-----HACEMOS EL CORRIMIENTO DERECHA
bcf STATUS,0 ;Con esto limpiamos al carry
movlw H'80'
movwf PORTB
movlw H'08'
movwf contador
loop3:
    call retardo
    rrf PORTB,1
    decf contador
    btfss STATUS,2
    goto loop3
;----- HACEMOS EL CORRIMIENTO A LA IZQUIERDA
movlw H'08'
movwf contador
goto loop2 ;

```

c6: ;En este caso prendemos y apagamos todos los LEDS con retardo de 1/2 segundo

```

movlw b'11111111'
movwf PORTB
call retardo

```

```
movlw b'00000000'
```

```
movwf PORTB
```

```
call retardo
```

```
goto ciclo
```

```
goto c6
```

retardo ;Subrutina utilizada para controlar el tiempo de retardo

```
movlw ctel ;carga el valor de ctel en w
```

```
movwf valor1 ;almacena en valor1 lo que hay en w
```

tres movlw cte2

```
movwf valor2
```

dos movlw cte3

```
movwf valor3
```

uno

decfsz valor3 ;decrementa valor3 y compara con 0 si es distinto
repite el ciclo

```
goto uno
```

```
decfsz valor2
```

```
goto dos
```

```
decfsz valor1
```

```
goto tres
```

```
return
```

end

Conclusiones:

Principalmente puedo concluir que los objetivos se cumplieron, pues logramos configurar con éxito el puerto A como entrada que en este caso fue el dipswitch y el puerto B como salida que fueron los LED's.

También puedo concluir que podemos construir un selector de opciones con ayuda de las combinaciones de los bits, en este caso fue interesante como agrupamos los primeros cuatro datos del bit 2 para poder realizarlo más fácil e ir comparando los bit 0 y 1 para realizar la tarea que correspondía a la combinación.

Finalmente reafirmamos los conceptos de retardo y de corrimientos, quedando estos más claros y reforzados.