



# UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO



## FACULTAD DE INGENIERÍA

### MATERIA

- Laboratorio de Microcomputadoras
  - Grupo:04

### PRÁCTICA 04

#### Puertos paralelos E/S

### PROFESOR

- M.I. Rubén Anaya García

### ALUMNOS

- Carreón Guzmán Mariana Ivette
  - Núm. Cta.: 312103914
    - Gpo. Teoría: 04
  - Rojas Méndez Gabriel
    - Gpo. Teoría:01
  - Núm. Cta.: 314141712

### SEMESTRE 2022-1

## Objetivo.

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

## Introducción

Cuando el microcontrolador PIC será configurado como entrada, se recomienda limpiar el contenido del registro de datos del puerto mediante la instrucción CLRF PORTX, esto con la finalidad de iniciar los latches de datos del puerto, con esta instrucción se configurará al puerto de manera correcta.

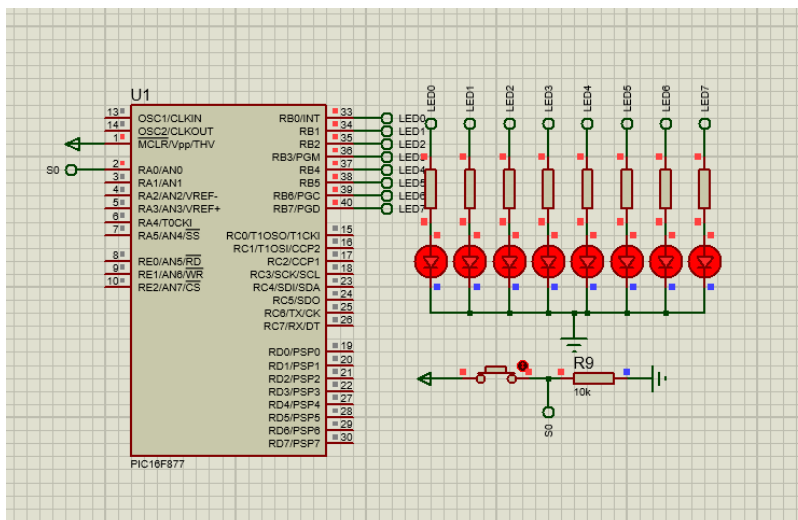
Además de lo anterior, para el caso del puerto A y E se requiere indicar en el registro ADCON1 ubicado en el banco 1 que se desea utilizar como E/S digitales, por lo que se escribirá un 06H o 07H en dicho registro, para posteriormente cargar el dato de configuración al registro TRISA o TRISE

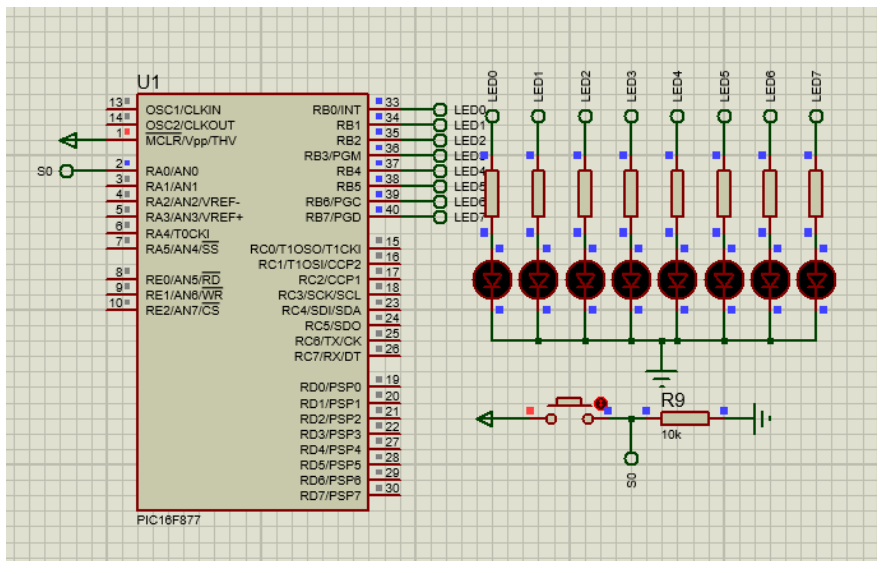
## Desarrollo

### Ejercicio 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 con el 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





```

C:\...Ejercicio1.asm

PROCESSOR 16F877
INCLUDE <P16F877.INC>

ORG 0
GOTO INICIO

ORG 5
INICIO: CLRWF PORTA
        BSF STATUS, RP0
        BCF STATUS, RP1
        MOVLW H'0'
        MOVWF TRISE
        CLRWF PORTE
        MOVLW 06H
        MOVWF ADCON1
        MOVLW H'3F'
        MOVWF TRISA
        BCF STATUS, RP0

LOOP:   BTFSF PORTA, 0
        GOTO ON
        MOVLW H'00'
        MOVWF PORTE
        GOTO LOOP

ON:     MOVLW H'FF'
        MOVWF PORTE
        GOTO LOOP
END

```

```

Output

Build Version Control Find in Files

Language tool versions: MPASMWIN.exe v5.51, mp
Sat Nov 27 01:08:54 2021

Clean: Deleting intermediary and output files.
Clean: Deleted file "C:\Users\hp\Documents\Facultad\Microcor
Clean: Done.
Executing: "C:\Program Files (x86)\Microchip\MPAS
Message[301] C:\PROGRAM FILES (X86)\MICROCH
Message[302] C:\USERS\HP\DOCUMENTS\FACU
Message[302] C:\USERS\HP\DOCUMENTS\FACU
Message[302] C:\USERS\HP\DOCUMENTS\FACU
Executing: "C:\Program Files (x86)\Microchip\MPAS
MPLINK 4.49, Linker
Device Database Version 1.14
Copyright (c) 1998-2011 Microchip Tech
Errors : 0

Loaded C:\Users\hp\Documents\Facultad\Microcor

Release build of project "C:\Users\hp\Documents\F
Language tool versions: MPASMWIN.exe v5.51, mp
Sat Nov 27 01:08:57 2021

BUILD SUCCEEDED

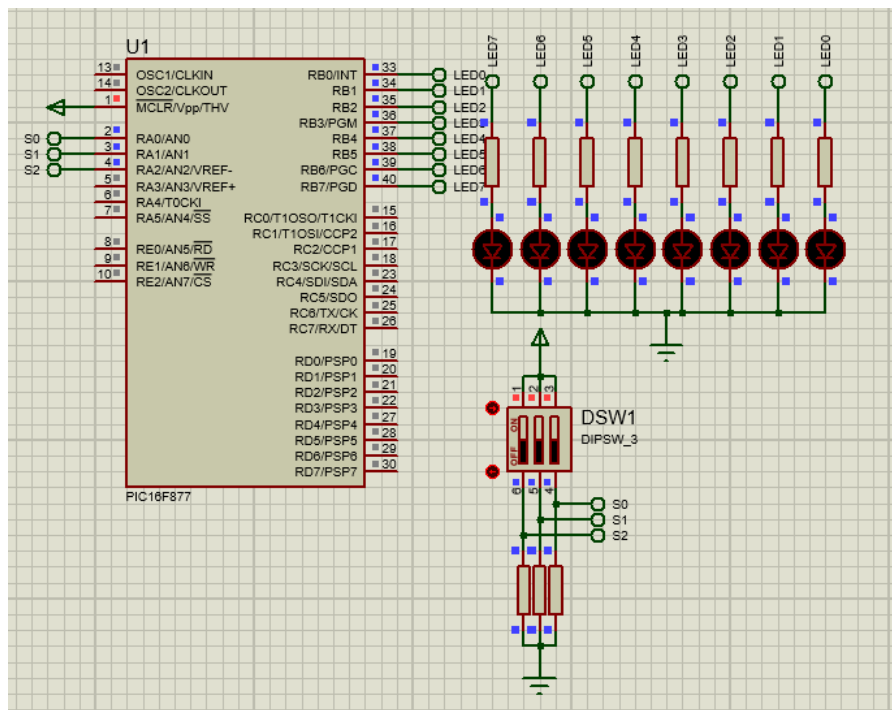
```

## Ejercicio 2

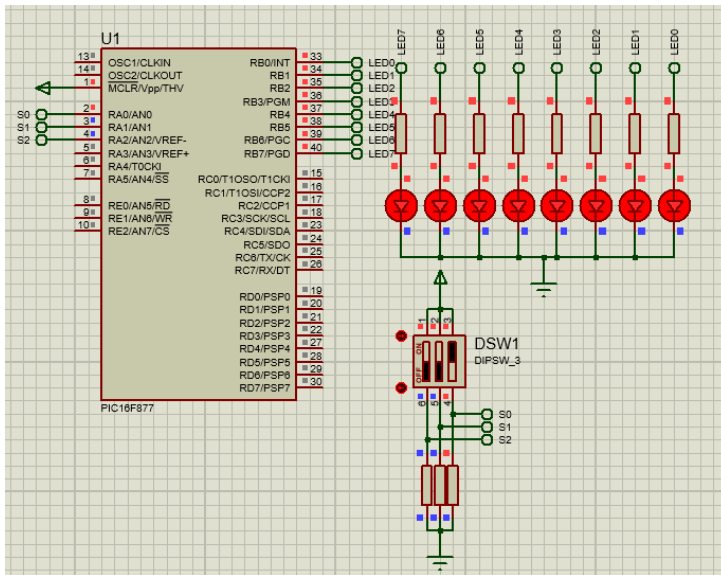
Realizar un programa, el cuál realice las siguientes acciones de control, para lo cual requiere trabajar un puerto de entrada y otro puerto de salida, usar los sugeridos en el ejercicio anterior; generar retardos de  $\frac{1}{2}$  seg., en las secuencias que lo requieran.

DATO	ACCION	Ejecución
\$00	Todos los leds apagados	00000000
\$01	Todos los leds encendidos	11111111
\$02	Corrimiento del bit más significativo hacia la derecha	10000000 01000000 00100000 ..... 00000001
\$03	Corrimiento del bit menos significativo hacia la izquierda	00000001 00000010 00000100 ..... 10000000
\$04	Corrimiento del bit más significativo hacia la derecha y a la izquierda	10000000 01000000 ..... 00000001 00000010 ..... 10000000
\$05	Apagar y encender todos los bits.	00000000 11111111

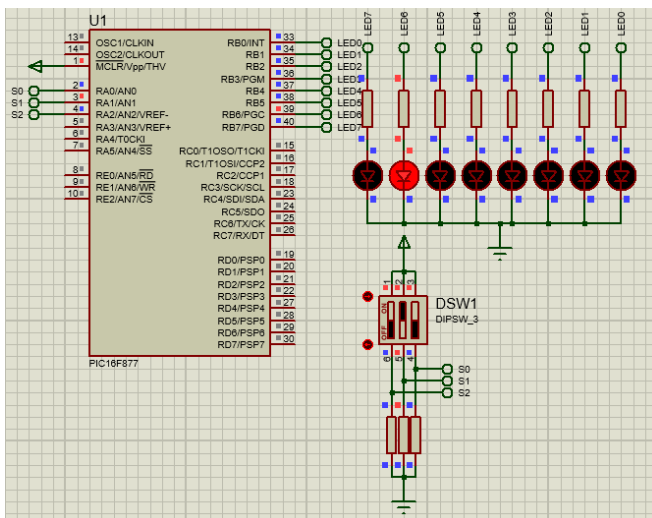
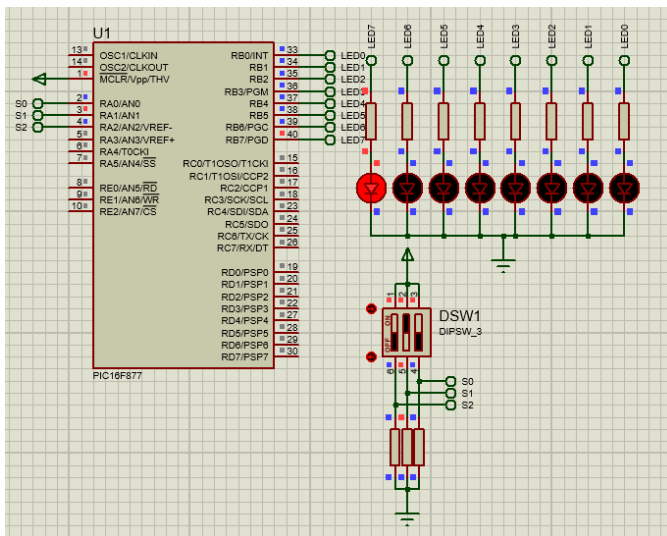
Dato: 00 Todos los leds están apagados



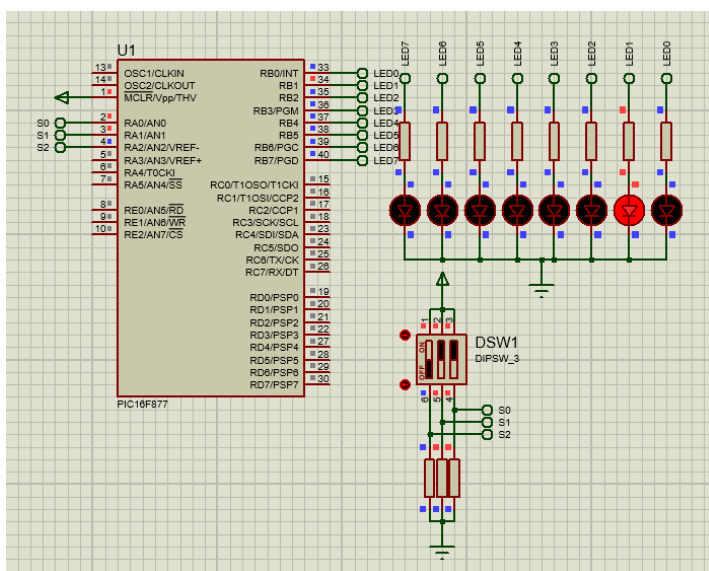
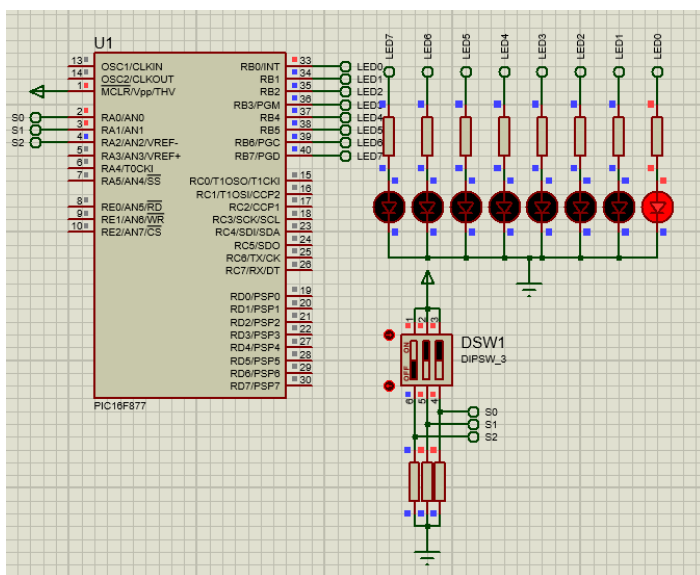
Dato: 01 Todos los leds están encendidos



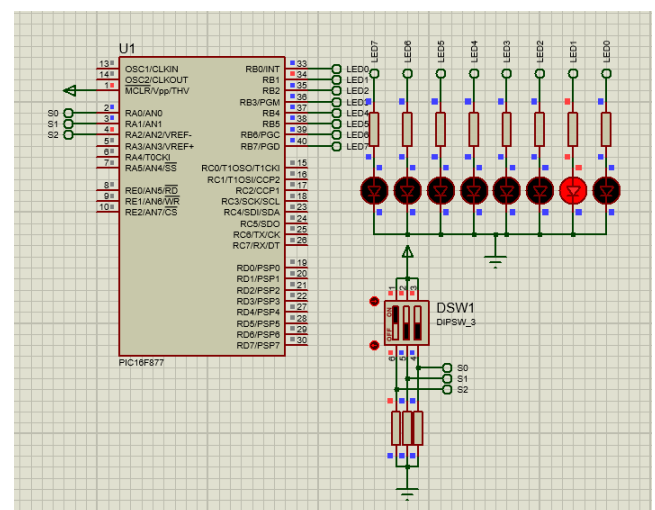
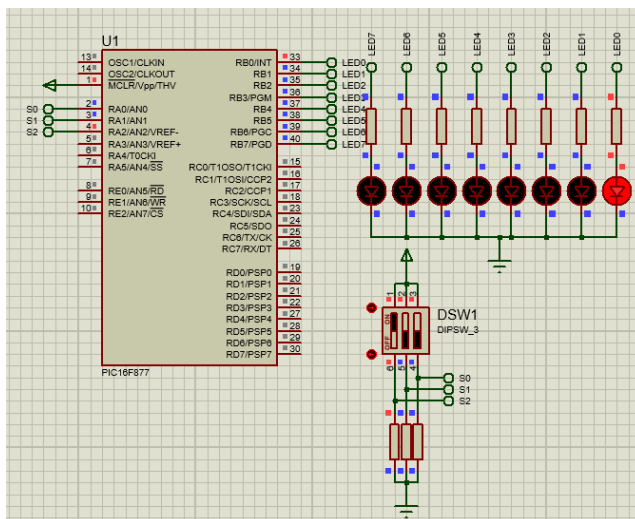
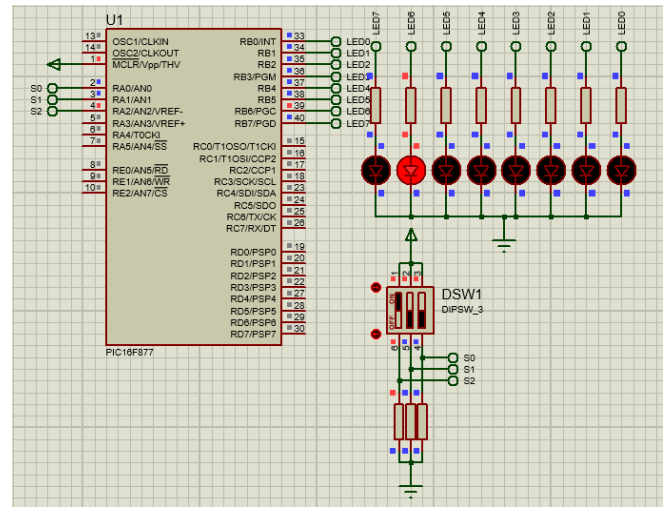
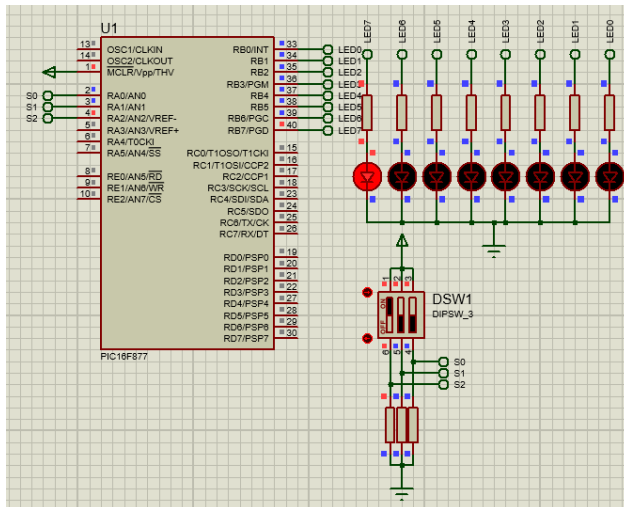
**Dato: 02 Corrimiento del más significativo a la derecha**



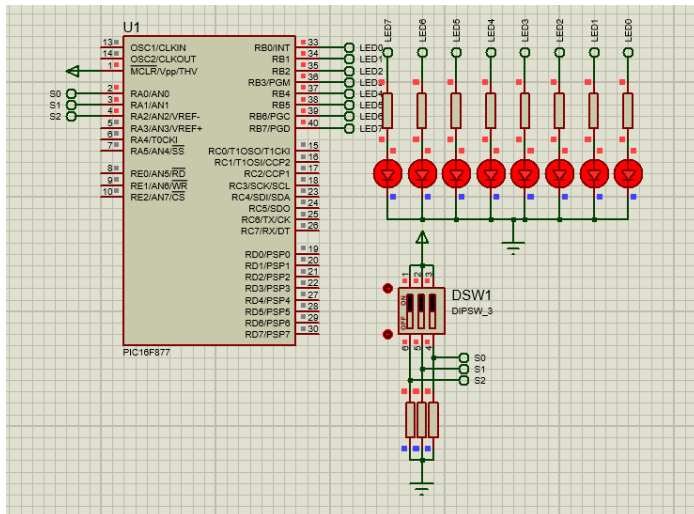
**Dato: 03 Corrimiento del menos significativo a la izquierda**



**Dato 04: Corrimiento del bit más significativo de derecha a izquierda**



**Dato 05: Apagar y encender todos los bits**



```

processor 16f877
include<pl6f877.inc>

;Variables para el DELAY
valor1 equ h'21'
valor2 equ h'22'
valor3 equ h'23'
cte1 equ 10h
cte2 equ 50h
cte3 equ 60h

;Definicion de variables a utilizar para
;comparar las entradas a traves del puerto A
v0 equ h'24'
v1 equ h'25'
v2 equ h'26'
v3 equ h'27'
v4 equ h'29'
v5 equ h'30'

c0 equ 0h
c1 equ 1h
c2 equ 2h
c3 equ 3h
c4 equ 4h
c5 equ 5h

org 0h
goto INICIO
org 05h

```

```

INICIO:
    clrf PORTA
    bsf STATUS,RPO ;Cambio al Banco 1
    bcf STATUS,RP1
    movlw h'0'
    movwf TRISE ;Configura Puerto B como salida
    clrf PORTE ;Limpia los bits de Puerto 1

    movlw 06h ;Configura puertos A y E como digitales
    movwf ADCON1
    movlw 3fh ;Configura el Puerto A como entrada
    movwf TRISA
    bcf STATUS,RPO ;Regresa al Banco 0

CICLO:
    movlw c0
    movwf v0
    movfw PORTA ;Mueve lo que hay en PORTA a W
    xorwf v0,w ;Verifica si la entrada es $00
    btfs STATUS,Z ;z=0?
    goto APG ;NO, entonces v0=W
    ;SI, entonces v0!=W

    movlw c1
    movwf v1
    movfw PORTA
    xorwf v1,w ;Verifica si la entrada es $01
    btfs STATUS,Z
    goto UNOS

    movlw c2
    movwf v2
    movfw PORTA
    xorwf v2,w ;Verifica si la entrada es $02
    btfs STATUS,Z
    goto DER

```



```

        movlw c3
        movwf v3
        movfw PORTA
        xorwf v3,w      ;Verifica si la entrada es $03
        btfsc STATUS,Z
        goto IZQ

        movlw c4
        movwf v4
        movfw PORTA
        xorwf v4,w      ;Verifica si la entrada es $04
        btfsc STATUS,Z
        goto DERIZQ

ENCAPG:      ;Loop que enciende y apaga los
        movlw h'00'      ;bits del puerto B
        movwf PORTB
        call retardo
        movlw h'FF'
        movwf PORTB
        call retardo
        goto CICLO

APG:         ;Apaga los bits del puerto B
        movlw h'00'
        movwf PORTB
        goto CICLO

UNOS:        ;Enciende los bits del puerto B
        movlw h'FF'
        movwf PORTB
        goto CICLO

DER:         ;Realiza corrimiento a la derecha
        movlw h'80'
        movwf PORTB
        call retardo

retardo      ;Rutina que genera un DELAY
        movlw ctel
        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

DER1:
        rrf PORTB,1
        call retardo
        btfss PORTB,0
        goto DER1
        goto CICLO

IZQ:         ;Realiza corrimiento a la izquierda
        movlw h'01'
        movwf PORTB
        call retardo

IZQ1:
        rlf PORTB,1
        call retardo
        btfss PORTB,7
        goto IZQ1
        goto CICLO

DERIZQ:      ;Realiza corrimiento a la derecha y
        movlw h'80'      ;luego a la izquierda
        movwf PORTB
        call retardo

DER2:
        rrf PORTB,1
        call retardo
        btfss PORTB,0
        goto DER2

        movlw h'01'
        movwf PORTB
        call retardo

IZQ2:
        rlf PORTB,1
        call retardo
        btfss PORTB,7
        goto IZQ2
        goto CICLO

```

## **Conclusiones**

### **Carreón Guzmán Mariana**

Esta práctica resultó muy interesante ya que pude aplicar algo que habíamos visto la clase pasada, en este caso vimos como usar el retardo para hacer los movimientos de los bits. Creo que en esta práctica se cumplieron con los objetivos ya que pudimos hacer distintas funciones de control, así como hacer uso de los puertos.

### **Rojas Méndez Gabriel**

En esta práctica pudimos observar como funcionan los puertos párelos y cómo es que se deben de configurar con las funciones de control. Esta práctica resultó muy didáctica ya que en el último ejercicio pudimos desarrollar distintos tipos de funciones de control, así como conjuntar información vista en prácticas pasadas.