Facultad de ingeniería

Materia: Laboratorio de Microcomputadoras

REPORTE DE LA PRÁCTICA 4

Título: Puertos paralelos Entradas/Salidas Integrantes:

• Martínez Pérez Brian Erik - 319049792

• Nuñez Rodas Abraham Enrique - 114003546

• Vicenteño Maldonado Jennifer Michel - 317207251

Profesor: Moises Melendez Reyes

Grupo: 1

Fecha de Entrega: 23 de marzo de 2025

Semestre: 2025-2



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

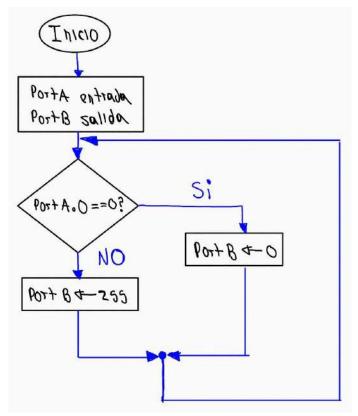
Ejercicios

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 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 | |

Tabla 4.1 Control de salidas controladas por un bit

Diagrama de flujo:



Funcionamiento de la solución:

imágenes:







Código:

PROCESSOR 16F877 INCLUDE <P16F877.INC>

ORG 0 GOTO INICIO

ORG 5

INICIO:

CLRF PORTA ;Limpia puerto A

BSF STATUS, RP0 BCF STATUS, RP1 MOVLW H'00'

MOVWF TRISB ;Convertimos a B en salida

CLRF PORTB ;Limpia puerto B

MOVLW H'06' MOVWF ADCON1 MOVLW H'3F'

MOVWF TRISA ;Convertimos A en entrada

BCF STATUS, RP0

LOOP:

BTFSC PORTA, 0

GOTO ON MOVLW H'00' MOVWF PORTB **GOTO LOOP**

ENCENDER_LED:

MOVLW H'FF' MOVWF PORTB GOTO ESPERA END

Análisis: Los objetivos se alcanzaron, ya que se logró utilizar los puertos de entrada y salida del microcontrolador para leer el estado del bit 0 del puerto A (RA0) y encender o apagar el puerto B (PORTB).

En el inicio, se limpian los puertos (CLRF PORTA, CLRF PORTB), se configura TRISA para definir A como entrada y TRISB para definir B como salida, se configura ADCON1 para desactivar las funciones analógicas en el puerto A.

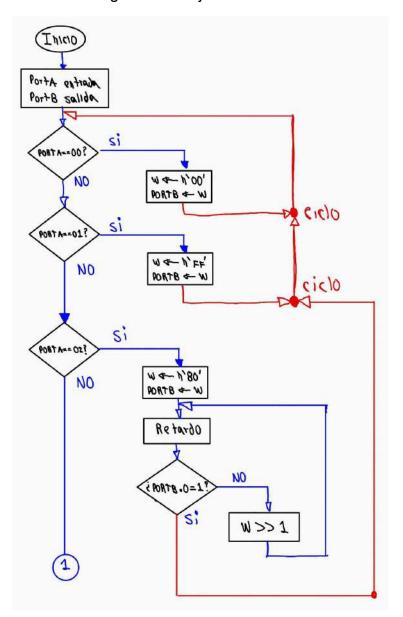
En el se loop se realiza lo siguiente, se lee el estado de RA0 (BTFSC PORTA, 0), si está en 1, se activa PORTB (MOVLW H'FF', MOVWF PORTB), de lo contrario si está en 0, se desactiva PORTB (MOVLW H'00', MOVWF PORTB), Se utilizaron modos de direccionamiento inmediato y directo (inmediato: MOVLW H'00') (directo: MOVWF TRISB)

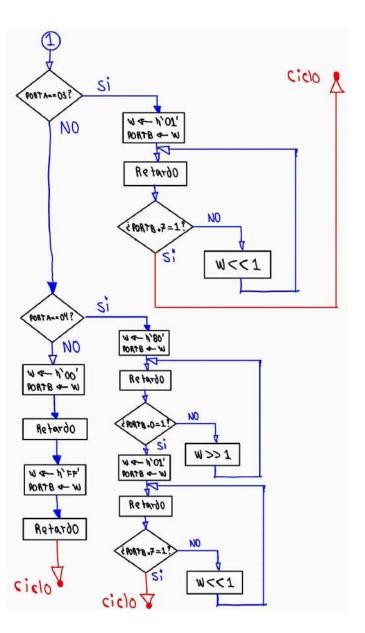
Ejercicio 2: Escribir un programa, el cuál realice las siguientes acciones de control indicadas, para lo cuál requiere trabajar un puerto de entrada y otro puerto de salida, usar los sugeridos en el ejercicio anterior; generar retardos de $\frac{1}{2}$ seg., en las secuencias que lo requieran.

| DATO PUERTO A | ACCION PUERTO B | Ejecución |
|------------------|--|--------------|
| 0x00 | Todos los leds apagados | 00000000 |
| 0x00 | Todos los leds apagados Todos los leds encendidos | 11111111 |
| 0x01 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 | 10000000 |
| | la derecha y a la izquierda | 01000000 |
| | The state of the s | *** *** **** |
| | | 00000001 |
| | | 00000010 |
| | | 10000000 |
| 0x05 | Apagar y encender todos los bits. | 00000000 |
| | 3.08.0 S | 11111111 |

Tabla 4.2 Control de salidas completo

Diagrama de flujo:



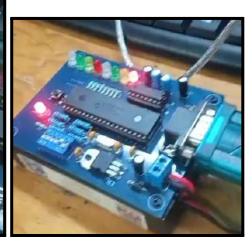


Funcionamiento de la solución:

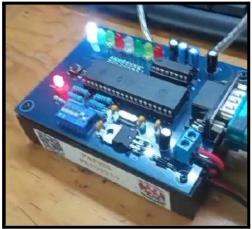
imágenes:











Código:

processor 16f877 include<p16f877.inc>

valor1 equ h'21' valor2 equ h'22' valor3 equ h'23' cte1 equ 20h cte2 equ 50h cte3 equ 90h

;Definicion de variables a utilizar para

v0 equ h'24'

v1 equ h'25'

v2 equ h'26'

v3 equ h'27'

```
v4 equ h'29'
ENTRADA 00 equ 0h
ENTRADA 01 equ 1h
ENTRADA 02 equ 2h
ENTRADA 03 equ 3h
ENTRADA 04 equ 4h
   org 0h
   goto INICIO
   org 05h
INICIO:
   clrf PORTA
   bsf STATUS,RP0 ;Cambio al Banco 1
   bcf STATUS,RP1
   movlw h'0'
   movwf TRISB ; Configura Puerto B como salida
                 ;Limpia los bits de Puerto 1
   clrf PORTB
   movlw 06h
                 ;Configura puertos A y E como digitales
   movwf ADCON1
   movlw 3fh
                ;Configura el Puerto A como entrada
   movwf TRISA
   bcf STATUS,RP0 ;Regresa al Banco 0
CICLO:
   movlw ENTRADA 00
   movwf v0
                   ;Mueve lo que hay en PORTA a W
   movfw PORTA
   xorwf v0,w
              ;Verifica si la entrada es $00
   btfsc STATUS,Z ;z=0?
   goto APAGAR LED
                         ;NO, entonces v0=W
            ;SI, entonves v0!=W
    movlw ENTRADA 01
   movwf v1
   movfw PORTA
                ;Verifica si la entrada es $01
   xorwf v1,w
   btfsc STATUS,Z
   goto ENCENDER_LED
   movlw ENTRADA 02
   movwf v2
   movfw PORTA
                ;Verifica si la entrada es $02
   xorwf v2,w
    btfsc STATUS,Z
```

goto DERECHA

```
movlw ENTRADA_03
   movwf v3
   movfw PORTA
   xorwf v3,w
               ;Verifica si la entrada es $03
   btfsc STATUS,Z
   goto IZQUIERDA
   movlw ENTRADA_04
   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
APAGAR LED:
                        ;Apaga los bits del puerto B
   movlw h'00'
   movwf PORTB
   goto CICLO
ENCENDER LED:
                           ;Enciende los bits del puerto B
   movlw h'FF'
   movwf PORTB
   goto CICLO
DERECHA:
                     ;Realiza corrimiento a la derecha
   movlw h'80'
   movwf PORTB
   call retardo
DER1:
   rrf PORTB,1
   call retardo
   btfss PORTB,0
   goto DER1
   goto CICLO
IZQUIERDA:
                      ;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
                 ;luego a la izquierda
    movlw h'80'
    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
retardo:
                  ;Rutina que genera un DELAY
    movlw cte1
    movwf valor1
LOOP 3:
    movwf cte2
    movwf valor2
LOOP 2:
    movlw cte3
    movwf valor3
LOOP 1:
    decfsz valor3
    goto LOOP_1
    decfsz valor2
```

goto LOOP 2

decfsz valor1 goto LOOP_3 return end

Análisis: El código configura el puerto A (PORTA) como entrada y el puerto B (PORTB) como salida. Esto se logra mediante la manipulación de los registros TRISA y TRISB en el banco de memoria correspondiente. Dependiendo del valor de PORTA, el programa realiza acciones como encender, apagar LEDs o realizar corrimientos de bits en PORTB, ademas se implementa una rutina de retardo que permite visualizar correctamente las acciones en PORTB. El mismo programa verifica las siguientes condiciones:

- Si PORTA = 00, se apagan los LEDs.
- Si PORTA = 01, se encienden los LEDs.
- Si PORTA = 02, se realiza un corrimiento a la derecha.
- Si PORTA = 03, se realiza un corrimiento a la izquierda.
- Si PORTA = 04, se alterna entre corrimientos a la derecha e izquierda.

Después de realizar una acción, el programa retorna al ciclo principal para leer nuevamente PORTA y repetir el proceso.

En el código se utilizan los modos de direccionamiento: Direccionamiento inmediato (movlw h'00'), Direccionamiento directo (movwf TRISB) y Direccionamiento indirecto (decfsz valor3)

Conclusiones:

Martínez Pérez Brian Erik

Para esta práctica cumplimos con el objetivo de configurar los puertos paralelos del PIC para poder utilizarlos como entrada y salida, en nuestro caso colocamos el puerto A como entrada y el puerto B como entrada. El dipwitch representaba nuestra entrada, ya que en ella colocaremos la opción de función de control que deseamos mostrar en los leds del puerto B.

Nuñez Rodas Abraham Enrique

Durante esta práctica, logramos profundizar en el uso eficiente de los puertos paralelos del microcontrolador PIC como interfaces de entrada y salida. A través de los ejercicios, reforzamos habilidades esenciales en la configuración de los registros TRIS para definir claramente la dirección de los puertos y el manejo correcto del banco de memoria para operaciones específicas. Además, aplicamos instrucciones como corrimientos de bits y retardos para realizar diversas secuencias, lo que nos permitió observar claramente cómo las entradas digitales controlan el comportamiento del sistema en tiempo real. Esta experiencia fue clave para entender conceptos prácticos de programación en lenguaje ensamblador y el funcionamiento interno del PIC.

Vicenteño Maldonado Jennifer Michel

Durante esta práctica, exploramos el uso de los puertos de entrada y salida del microcontrolador. Pudimos hacer que un puerto realizara acciones dependiendo de lo que detectara en otro, como encender o apagar LEDs usando interruptores. Además, desarrollamos un tipo de menú con opciones en binario para ejecutar distintas funciones, incluyendo secuencias con retardos. Esto nos ayudó a comprender mejor el funcionamiento de los puertos y cómo crear programas secuenciales en lenguaje ensamblador.

Referencias:

- del PIC, 2. 1-La Familia. (s/f). 2.- Descripción General del PIC16F877.
 Edu.ar., de https://exa.unne.edu.ar/ingenieria/sysistemas/public_html/Archi_pdf/HojaDato s/Microcontrolador es/PIC16F877.pdf
- (S/f). Newark.com. Recuperado de https://mexico.newark.com/microchip/pic16f877a-i-p/microcontroller-mcu-8-bit pic16/dp/69K7640?srsltid=AfmBOorWLTceQMTppGk0OMGmmjB6Upliiw55U 28F2qBZH2pUYET 3EInu
- Descripción General del PIC16F877 1 2.-Descripción General del PIC16F877
 2.1.-La Familia del PIC16F877. (n.d.).
 - https://exa.unne.edu.ar/ingenieria/sysistemas/public_html/Archi_pdf/HojaDatos/Microcontroladores/PIC16F877.pdf
- Manual de usuario Microchip PIC16F877A (280 páginas). (2025). Manual.cr.
 https://www.manual.cr/microchip/pic16f877a/manual?utm_
- PIC16F87XA Devices Included in this Data Sheet: High-Performance RISC
 CPU. (n.d.). https://ww1.microchip.com/downloads/en/DeviceDoc/39582C.pdf
- puertos-de-entradasalida MIKROE. (2024). MIKROE.
 https://www.mikroe.com/ebooks/microcontroladores-pic-programacion-en-c-con-ejemplos/puertos-de-entradasalida