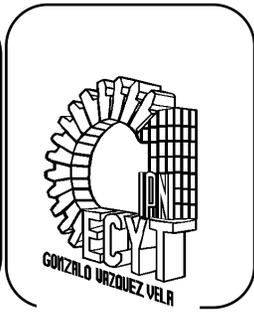




INSTITUTO POLITÉCNICO NACIONAL
Centro de Estudios Científicos y Tecnológicos N° 1
“Gonzalo Vázquez Vela”
Academia de Sistemas Digitales
Prácticas de Arquitectura de Microprocesadores y Microcontroladores



NOMBRE DEL ALUMNO: _____
Apellido Paterno

Apellido Paterno Nombre

N° DE BOLETA: _____ GRUPO: _____

ASIGNATURA: **Arquitectura de Microp. y Microcontr**

HOJA	DE	FECHA			EVALUACION
		DIA	MES	AÑO	

PROFESOR: _____

Práctica 8

Manejo de Puertos de Entrada y salida del Microcontrolador PIC

Competencias de La Unidad:

Realiza programas implementando diversos tipos de subrutinas básicas, avanzadas y servicio a interrupciones para optimizar la programación en lenguaje ensamblador.

Resultado de Aprendizaje Propuesto (RAP):

- Identifica la manipulación y configuración de puertos de entrada y salida, mediante programas ya escritos, e identifica las instrucciones básicas necesarias para los puertos de entrada y salida
- Identifica las subrutinas de tiempo, mediante programas escritos junto con las instrucciones que se tienen para la cuantificación y la obtención de la fórmula para encontrar el tiempo que ésta utiliza para ejecutarse.

Objetivos De La Práctica:

1. Realizar la simulación de un programa en ensamblador para comprobar su funcionamiento.
2. Realizar un control de elementos visuales tipo LED (Display 7 segmentos). Por medio de elementos de entrada
3. Comprender, la manera de grabar el Microcontrolador utilizado un programador.

<i>Equipo Necesario</i>	<i>Material Necesario</i>
<p>Computadora (con el Software MPLAB IDE, IC-PROG o similar)</p> <p>Programador tipo JDM o similar.</p>	<p>Instrucciones del PIC 16F887</p> <p>Microcontrolador PIC16F887 u otro de gama media</p> <p>Capacitores LED's Resistencias pushbutton Cristal de cuarzo de 4MZ</p> <p>(Para los valores de estos elementos ver figura 8.1)</p>

Introducción Teórica

Puertos de Entrada y salida

El Microcontrolador 16F887 cuenta con 5 puertos direccionales, denominados A, B, C, D Y E.

Sus principales características son:

- Programables como entradas o salidas individualmente.
- Capaces de trabajar con corrientes de 25 mA. en cada línea. No obstante la corriente total en los puertos A, B no puede superar los 200 mA. y en los puertos C otros 200 mA.
- Entradas tipo TTL o ST (Schmitt Trigger).
- Resistencias Pull-up (habilitadas por programa) en el puerto B

Las líneas de E/S están agrupadas en 5 puertos: A (6 bits), B (8 bits), C (8 bits), D (8 bits), E (3 bits). Cada puerto de E/S tiene asociados dos registros TRISX y PORTX. El primer registro dispone de un bit por cada línea del puerto, y controlará si funciona como entrada (Input, 1) o como salida (Output, 0). El segundo registro nos permite acceder al puerto. Con una escritura se modifican los bits configurados como salida, y con una lectura accedemos tanto a los de entrada como a los de salida. Si se realiza una escritura y de forma inmediata una lectura, puede que la salida no haya alcanzado el nivel adecuado generando incoherencias

Para inicializar los puertos de manera general en los PIC de gama media se realiza las siguientes instrucciones

Como entrada

- Colocarse en el banco 1,
- Cargar "1's" en el registro TRISX, si se desea que se comporte el puerto como entrada, si solo desea configurar como entrada bits individuales colocar 1 en los bits que desee.
- Regresar al banco 0.

Como Salida

- Colocarse en el banco 1.
- Cargar "0's" en el registro TRISX, si se desea que se comporte el puerto como salida, si solo desea configurar como salida bits individuales colocar 0 en los bits que desee.
- Regresar al banco 0

Para cambiar de banco se utiliza los bits 5 y 6 del registro de estado.

Un ejemplo se muestra en seguida

;Inicializa los puerto A como entrada

```
movlw B'00000000' ;Limpia Puerto A
movwf PORTA
movlw B'00001111'
banksel
clrf ansel
banksel anselh
clrf anselh
banksel trisa
movlw B'00001111'
movwf TRISA ; PortA como entrada
bcf STATUS, RP0 ; Banco 0
```

;En el caso del PIC16F887 es necesario utilizar el registro ansel y anselh para configurar las ;entradas digitales del puerto A, ya que están definidas como analógicas.

;Inicializa los puerto A como salida

```
movlw B'00000000' ;Limpia Puerto A
movwf PORTA
movlw B'00000000'
bsf STATUS, RP0 ; Banco 1
movwf TRISA ; PortA como salida
bcf STATUS, RP0 ; Banco 0
```

Bits de Configuración

Todos los PIC disponen de un cierto número de bits de configuración que están disponibles en la memoria EEPROM, y solo se accede a ellos cuando se programa el dispositivo, permitiendo determinar ciertas necesidades con el fin de adaptarlo a las aplicaciones que se realice, debido a que dependen del dispositivo.

Las características que se programan en los bits de configuración son las siguientes:

El tipo de oscilador.

La habilitación o no del perro guardián.

La protección de la memoria de programa.

La protección de la memoria EEPROM de datos, si existe en el dispositivo.

Las características del RESET y la alimentación del dispositivo.

La figura muestra un esquema de los bits de configuración de un microcontrolador de PIC16F88X.

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
-	-	\overline{DEBUG}	LVP	FCMEN	IESO	BOREN1	BOREN0	\overline{CPD}	\overline{CP}	MCLRE	$\overline{PWRTÉ}$	WDTE	FOSC2	FOSC1	FOSCO

\overline{DEBUG}

Modo de depuración del circuito

1 Deshabilitado (RB6 Y RB7 son de propósito general I/O)

0 Habilitado (RB6 Y RB7 se usan para la depuración)

LVP

Programación en bajo voltaje

1 Habilitada

0 Deshabilitada

FCMEN

Monitor del reloj a prueba de fallos

1 Habilitado

0 Deshabilitado

IESO

Bit interno de comunicación externa

1 Habilitado

0 Deshabilitado

BOREN 1,0

Reset por fallo de alimentación

11 Habilitado

10 Habilitado en operación y deshabilitado en modo sleep

01 Controlado por el bit SBOREN del registro PCON

00 Deshabilitado

\overline{CPD}

Código de protección de datos

1 Deshabilitado

0 Habilitado

\overline{CP}

Protección de memoria FLASH del programa

1 Deshabilitado

0 Habilitado

MCLRE

Función del pin RE3/ \overline{MCLRE}

1 Función en \overline{MCLRE}

0 Funciona como entrada digital, \overline{MCLRE} lo toma de VDD

$\overline{PWRTÉ}$

Temporizador de arranque de encendido

1 Deshabilitado

0 Habilitado

WDTE

Temporizador del perro guardián

1 Habilitado

0 Deshabilitado y puede ser habilitado por el bit SWDTEN del registro WDTCON

FOSC 2,1,0

Selección del tipo de oscilador

111 RC

110 RCIO

101 INTOSC

100 INTOSCIO

011 EC

010 HS

001 XT

000 LP

En el caso del PIC 16F887 los bits de configuración se encuentran en la dirección 2007H y podemos almacenar el valor E3C1H, En el caso de requerir utilizar una dispositivos diferente, se debe verificar en las hojas de especificaciones, cuales son los bits y posición en la cual están implementados.

El registro quedaría de la siguiente manera:

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
-	-	\overline{DEBUG}	LVP	FCMEN	IESO	BOREN1	BOREN0	\overline{CPD}	\overline{CP}	MCLRE	$\overline{PWRT\overline{E}}$	WDTE	FOSC2	FOSC1	FOSCO
1	1	1	0	0	0	1	1	1	1	0	0	0	0	0	1

Definiendo las siguientes características:

Oscilador: XT
 Watchdog (WDT) Deshabilitado (Apagado)
 PWRT: (PUT) Deshabilitado (Apagado)
 CP: Deshabilitado (Apagado)
 CPD: Deshabilitado (Apagado)
 BODEN Habilitado (Encendido)
 LVP Deshabilitado (Apagado)
 Los bits sin implementar se leen como '1'

Otra manera de hacerlo es especificando en el código fuente los bit de configuración , por lo cual el ensamblador de MPASM tiene la características de permitir definir los valores que tomaran al programar el dispositivo , mediante la directiva `__CONFIG`

Cabe comentar que los símbolos utilizados en la directiva `__CONFIG` no son los mismos para todos los dispositivos, pero pueden ser verificados el archivo **include** de Microchip.

Además, es importante tomar en cuenta que el dispositivo seleccionado en el programador sea el que se utiliza para generar el código fuente, debido a que los bits de configuración dependen del dispositivo.

ACTIVIDADES PRÁCTICAS

Parte 1

- A. Crear un proyecto de nombre pra8 en la carpeta c:\PIC\practica8. Los programas de cada ejercicio deben ser guardados con el nombre practica8X.asm con X= 1, 2, 3...,A..**
- B. Seleccionar dispositivo PIC16F887 y configurar bits correspondientes.**
- C. Habilitar Simulador MPLSB SIM, y modificar la frecuencia del simulador a 4 Mhz.**

- D. Utilizaremos la herramienta de stopwatch, para obtener la elija Debugger >> Stopwatch.
- E. Obtener la herramienta de watch, de la siguiente manera View>> watch.
- F. Y seleccione los registros PORTA, PORTB, PORTC, PORTD, PORTE, TRISA, TRISB, TRISC, TRISD, TRISE y W. y algún otro que necesite.
- G. Armar el circuito de la figura 8.1
- H. Simular y grabarlos los siguientes programa de ejemplo y observe su funcionamiento

Nota: En caso de no colocar la palabra de configuración con la directiva __CONFIG, al grabarlo deshabilitar en la palabra de configuración, WDT y LVP, además recuerde seleccionar el tipo de oscilador a XT.

Programa Ejemplo 1:

El siguiente código permite encender un display de 7 segmentos de cátodo común, para visualizar un conteo ascendente de 0 a 9 al oprimir el botón conectado al PINA0.

```

__CONFIG _CONFIG1, _LVP_OFF & _FCMEN_OFF & _IESO_OFF &
_BOR_OFF & _CPD_OFF & _CP_OFF & _MCLRE_OFF & _PWRTE_ON & _WDT_OFF & _XT_OSC
__CONFIG _CONFIG2, _WRT_OFF & _BOR21V
LIST P=16F887
#include <P16F887.INC>
CBLOCK 025

CONT1
CONT2
CONT3

                                ENDC

                                ;CONFIGURAR COMO
SALIDA PUERTO B Y COMO ENTRADA A
                                BANKSEL ANSEL
                                CLRF ANSEL
                                BCF STATUS,RP1
                                BSF STATUS,RP0                                ;BANCO 1
                                MOVLW B'00000000'
                                MOVWF TRISD                                ;PORTB COMO SALIDA
                                MOVLW B'11111111'                                ;PORTA COMO ENTRADA
                                MOVWF TRISA
                                MOVLW B'00000000'
                                MOVWF TRISC
                                BCF STATUS,RP0                                ;BANCO 0
                                CLRF PORTC                                ;LIMPIA PUERTO C
                                MOVLW 0X10
                                MOVWF PORTC

                                ;PROGRAMA
PRINCIPAL VISUALIZACION DE VALORES DEL 0 AL 9 PRESIONANDO EL BOTON EN PIN-A0
INICIO:
                                CLRF PORTD                                ;LIMPIA PUERTO B

                                BTFSC PORTA,0
                                GOTO INICIO
                                GOTO T1

T1:
                                MOVLW 3FH                                ;ENCIENDE EL CERO
                                MOVWF PORTD
                                CALL RETARDO1S
                                NOP
                                CLRF PORTD

```

```

MOVW 06H ;ENCIENDE EL UNO
MOVWF PORTD
CALL RETARDO1S
NOP
CLRF PORTD
MOVW 5BH ;ENCIENDE EL DOS
MOVWF PORTD
CALL RETARDO1S
NOP
CLRF PORTD
MOVW 4FH ;ENCIENDE EL TRES
MOVWF PORTD
CALL RETARDO1S
NOP
CLRF PORTD
MOVW 66H ;ENCIENDE EL CUATRO
MOVWF PORTD
CALL RETARDO1S
NOP
CLRF PORTD
MOVW 6DH ;ENCIENDE EL CINCO
MOVWF PORTD
CALL RETARDO1S
NOP
CLRF PORTD
MOVW 7DH ;ENCIENDE EL SEIS
MOVWF PORTD
CALL RETARDO1S
NOP
CLRF PORTD
MOVW 07H ;ENCIENDE EL SIETE
MOVWF PORTD
CALL RETARDO1S
NOP
CLRF PORTD
MOVW 7FH ;ENCIENDE EL OCHO
MOVWF PORTD
CALL RETARDO1S
NOP
CLRF PORTD
MOVW 67H ;ENCIENDE EL NUEVE
MOVWF PORTD
CALL RETARDO1S
NOP
CLRF PORTD
MOVW 40H ;ENCIENDE EL GUION
MOVWF PORTD
CALL RETARDO1S
NOP
CLRF PORTD
GOTO INICIO

RETARDO1S:
MOVW D'12'
MOVWF CONT1

CICLO1:
CALL RETARDO2
DECFSZ CONT1,1
GOTO CICLO1
RETURN

RETARDO2:
MOVW D'216'
MOVWF CONT2

CICLO2:
CALL RETARDO3
DECFSZ CONT2,1
GOTO CICLO2
RETURN

RETARDO3:
MOVW D'255'
MOVWF CONT3

```

CICLO3:

```
DECFSZ CONT3,1
GOTO CICLO3
RETURN
END
```

Programa Ejemplo 2:

El siguiente código permite encender un display de 7 segmentos de cátodo común, para visualizar un conteo ascendente de 0 a 9 al oprimir el botón conectado en PINA0 y un conteo descendente de 9 a 0 al oprimir el PINA1, al no oprimir o terminar cualquiera de las secuencias se visualiza un guion.

```

CONFIG CONFIG1, _LVP_OFF & _FCMEN_OFF & _IESO_OFF &
_BOR_OFF & _CPD_OFF & _CP_OFF & _MCLRE_OFF & _PWRTE_ON & _WDT_OFF & _XT_OSC
CONFIG CONFIG2, _WRT_OFF & _BOR21V
LIST P=16F887
#include <P16F887.INC>
CBLOCK 025

CONT1
CONT2
CONT3

ENDC
MOVLW B'01110101' ;CONFIGURA OSCILADOR
MOVWF OSCCON ;CONFIGURAR COMO

SALIDA PUERTO D Y COMO ENTRADA A
BANKSEL ANSEL
CLRF ANSEL
BCF STATUS,RP1
BSF STATUS,RP0 ;BANCO 1
MOVLW B'00000000'
MOVWF TRISD ;PORTD COMO SALIDA
MOVWF TRISC ;PORTA COMO ENTRADA
MOVLW B'11111111'
MOVWF TRISA ;BANCO 0
BCF STATUS,RP0
MOVLW 0X10
MOVWF PORTC ;PROGRAMA

PRINCIPAL VISUALIZACION DE VALORES DEL 0 AL 9 DE MANERA ASCENDENTE O DESCENDENTE PRESIONANDO
EL BOTON EN PIN-A0 O PIN-A1
INICIO:
MOVLW 40H ;ENCIENDE EL GUION
MOVWF PORTD

PIN0:
BTFSC PORTA,0
GOTO PIN1
GOTO ACT0

PIN1:
BTFSC PORTA,1
GOTO PIN0
GOTO ACT1

ACT0:
MOVLW 3FH ;ENCIENDE EL CERO
MOVWF PORTD
CALL RETARDO1S
NOP
CLRF PORTD ;LIMPIA EL PUERTO D
MOVLW 06H ;ENCIENDE EL UNO
MOVWF PORTD
CALL RETARDO1S
NOP
CLRF PORTD ;LIMPIA EL PUERTO D
MOVLW 5BH ;ENCIENDE EL DOS
MOVWF PORTD
CALL RETARDO1S
NOP
CLRF PORTD ;LIMPIA EL PUERTO D
```

```

MOVW 4FH ;ENCIENDE EL TRES
MOVWF PORTD
CALL RETARDO1S
NOP
CLRF PORTD ;LIMPIA EL PUERTO D
MOVW 66H ;ENCIENDE EL CUATRO
MOVWF PORTD
CALL RETARDO1S
NOP
CLRF PORTD ;LIMPIA EL PUERTO D
MOVW 6DH ;ENCIENDE EL CINCO
MOVWF PORTD
CALL RETARDO1S
NOP
CLRF PORTD ;LIMPIA EL PUERTO D
MOVW 7DH ;ENCIENDE EL SEIS
MOVWF PORTD
CALL RETARDO1S
NOP
CLRF PORTD ;LIMPIA EL PUERTO D
MOVW 07H ;ENCIENDE EL SIETE
MOVWF PORTD
CALL RETARDO1S
NOP
CLRF PORTD ;LIMPIA EL PUERTO D
MOVW 7FH ;ENCIENDE EL OCHO
MOVWF PORTD
CALL RETARDO1S
NOP
CLRF PORTD ;LIMPIA EL PUERTO D
MOVW 67H ;ENCIENDE EL NUEVE
MOVWF PORTD
CALL RETARDO1S
NOP
CLRF PORTD ;LIMPIA EL PUERTO D
MOVW 40H ;ENCIENDE EL GUION
MOVWF PORTD
CALL RETARDO1S
NOP
CLRF PORTD ;LIMPIA EL PUERTO D
GOTO INICIO

ACT1:
MOVW 67H ;ENCIENDE EL NUEVE
MOVWF PORTD
CALL RETARDO1S
NOP
CLRF PORTD ;LIMPIA EL PUERTO D
MOVW 7FH ;ENCIENDE EL OCHO
MOVWF PORTD
CALL RETARDO1S
NOP
CLRF PORTD ;LIMPIA EL PUERTO D
MOVW 07H ;ENCIENDE EL SIETE
MOVWF PORTD
CALL RETARDO1S
NOP
CLRF PORTD ;LIMPIA EL PUERTO D
MOVW 7DH ;ENCIENDE EL SEIS
MOVWF PORTD
CALL RETARDO1S
NOP
CLRF PORTD ;LIMPIA EL PUERTO D
MOVW 6DH ;ENCIENDE EL CINCO
MOVWF PORTD
CALL RETARDO1S
NOP
CLRF PORTD ;LIMPIA EL PUERTO D
MOVW 66H ;ENCIENDE EL CUATRO
MOVWF PORTD
CALL RETARDO1S

```

```

NOP
CLRF PORTD ;LIMPIA EL PUERTO D
MOVLW 4FH ;ENCIENDE EL TRES
MOVWF PORTD
CALL RETARDO1S
NOP
CLRF PORTD ;LIMPIA EL PUERTO D
MOVLW 5BH ;ENCIENDE EL DOS
MOVWF PORTD
CALL RETARDO1S
NOP
CLRF PORTD ;LIMPIA EL PUERTO D
MOVLW 06H ;ENCIENDE EL UNO
MOVWF PORTD
CALL RETARDO1S
NOP
CLRF PORTD ;LIMPIA EL PUERTO D
MOVLW 3FH ;ENCIENDE EL CERO
MOVWF PORTD
CALL RETARDO1S
NOP
CLRF PORTD ;LIMPIA EL PUERTO D

GOTO INICIO

RETARDO1S:
MOVWF CONT1
MOVWF CONT1

CICLO1:
CALL RETARDO2
DECFSZ CONT1,1
GOTO CICLO1
RETURN

RETARDO2:
MOVWF CONT2
MOVWF CONT2

CICLO2:
CALL RETARDO3
DECFSZ CONT2,1
GOTO CICLO2
RETURN

RETARDO3:
MOVWF CONT3
MOVWF CONT3

CICLO3:
DECFSZ CONT3,1
GOTO CICLO3
RETURN
END

```

Parte 2:

1. **Crear un programa que tenga cuatro interruptores conectados al puerto A y además, se tiene un display de cátodo común conectado en el puerto D. El programa realiza las siguientes acciones. (figura 8.1)**

- Si se activa PIN0, en el display deben de aparecer tu primer nombre y al terminar aparecer 0.
- Si se activa PIN1 en el display deben de aparecer C-E-C-y-T-1 y al terminar aparecer 0
- Si se activa PIN2 en el display deben de aparecer 0-1-2-3-3-2-1 y al terminar aparecer 0
- Si se activa PIN3 en el display deben de aparecer A-P-R-O-B-A-D-A y al terminar aparecer 0
- Si no se activa ningún PIN en el display debe aparecer 0

El programa debe ser cíclico.

Sugerencia: Debe crear una tabla que relacione los segmentos que se tienen que encender para cada una de las letras o números.

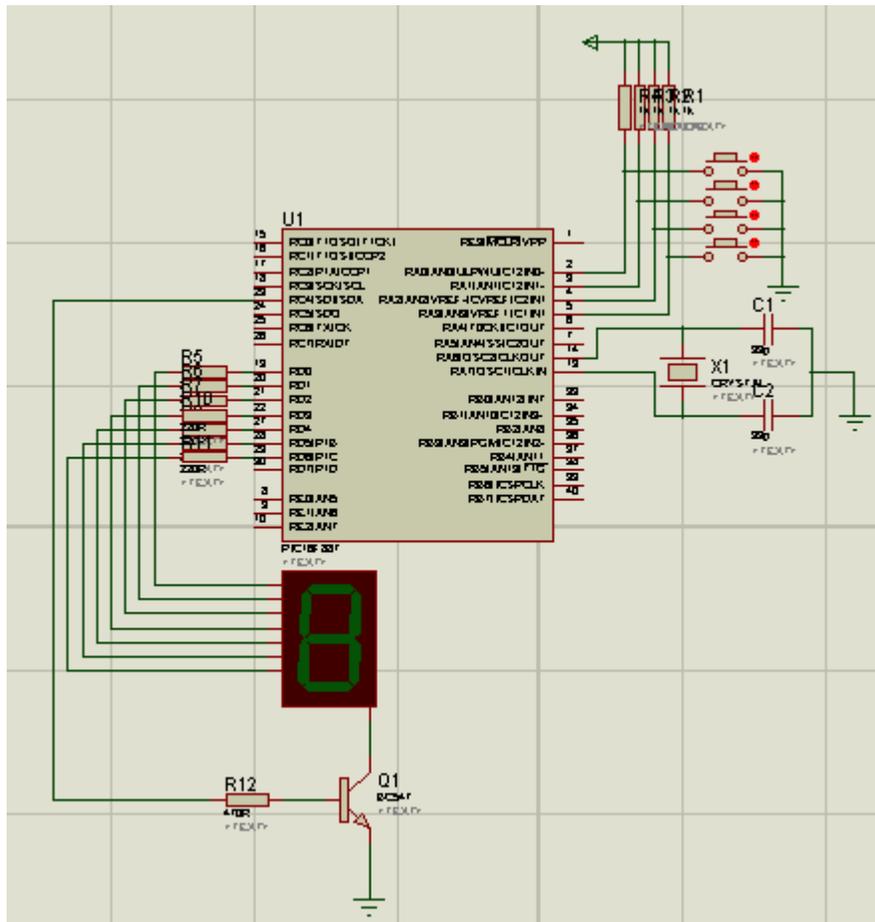


Figura 8.1 Control de un Display 7 segmentos

Nota: La terminal 11 o 32 del PIC16F887 se conectan a tierra.

2. Conclusiones

A. Realizar conclusiones de manera individual.

3. Cuestionario

- Como se configuran el puerto D como salida.
- Como se configuran el puerto A como entrada.
- Con que instrucción se mando un dato de salida a un puerto.
- Con que instrucción se lee un bit de entrada a un puerto.
- Realiza un programa para generar un contador ascendente del 0 al 9 en un display de ánodo común.
- Como conectaría un led a un pin de salida, para que se active con un cero.
- Como se calcula la resistencia de un led si se quiere activar con 5V.
- Como conectaría un interruptor para proporcionar un '1' de entrada.

i) A que se refiere el concepto de rebote en entradas digitales.

Comentarios Finales

- **El alumno entrega un reporte de la práctica, como el profesor lo indique.**
- **El reporte debe contener el diagrama de flujo o algoritmo (Seudo código) de cada uno de los programas.**
- **Además, en el reporte deben anexarse las conclusiones y cuestionario contestado.**