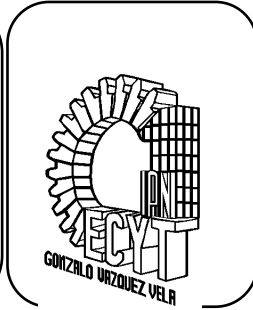


INSTITUTO POLITÉCNICO NACIONAL
Centro de Estudios Científicos y Tecnológicos N° 1
“Gonzalo Vázquez Vela”
Academia de Sistemas Digitales
Prácticas de Arquitectura de Microprocesadores y Microcontroladores



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 7

Puertos de salida del Microcontrolador

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. Desarrolla programas que habilite y deshabilite las salidas de microcontrolador por cierto tiempo.
3. Implementar programas en un circuito basado en microcontrolador.

<p>Equipo Necesario</p> <p>Computadora (con el Software MPLAB IDE, IC-PROG o similar)</p>	<p>Material Necesario</p> <p>Instrucciones del PIC 16F887 Microcontrolador PIC16F887 u otro de gama media</p>
---	---

Programador tipo JDM o similar.	Capacitores LED's Resistencias pushbutton Cristal de cuarzo de 4MZ (Para los valores de estos elementos ver figura 7.1)
---------------------------------	--

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
banksel ansel
clrf ansel
banksel anselh
clrf anselh
banksel trisa
movlw B'00001111'
movwf TRISA ; PortA como entrada
banksel porta
```

;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 guardian
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`

En seguida se muestra la forma de utilizarlo

```
__CONFIG _CONFIG1, _LVP_OFF & _FCMEN_OFF & _IESO_OFF & _BOR_OFF & _CPD_OFF & _CP_OFF & _MCLRE_OFF &
_PWRT_ON & _WDT_OFF & _XT_OSC
__CONFIG _CONFIG2, _WRT_OFF & _BOR21V
```

Definiendo las siguientes características:

Oscilador: XT
 Watchdog (WDT) Deshabilitado (Apagado)
 PWRT: (PUT) Habilitado (Apagado)
 CP: Deshabilitado (Apagado)
 CPD: Deshabilitado (Apagado)
 BODEN Habilitado (Encendido)
 LVP Deshabilitado (Apagado)

que equivalen a almacenar el valor E3C1H en dirección 2007H en el caso del PIC 16F887.

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.

ACTIVIDADES PRÁCTICAS

Parte 1

A. Armar el siguiente circuito

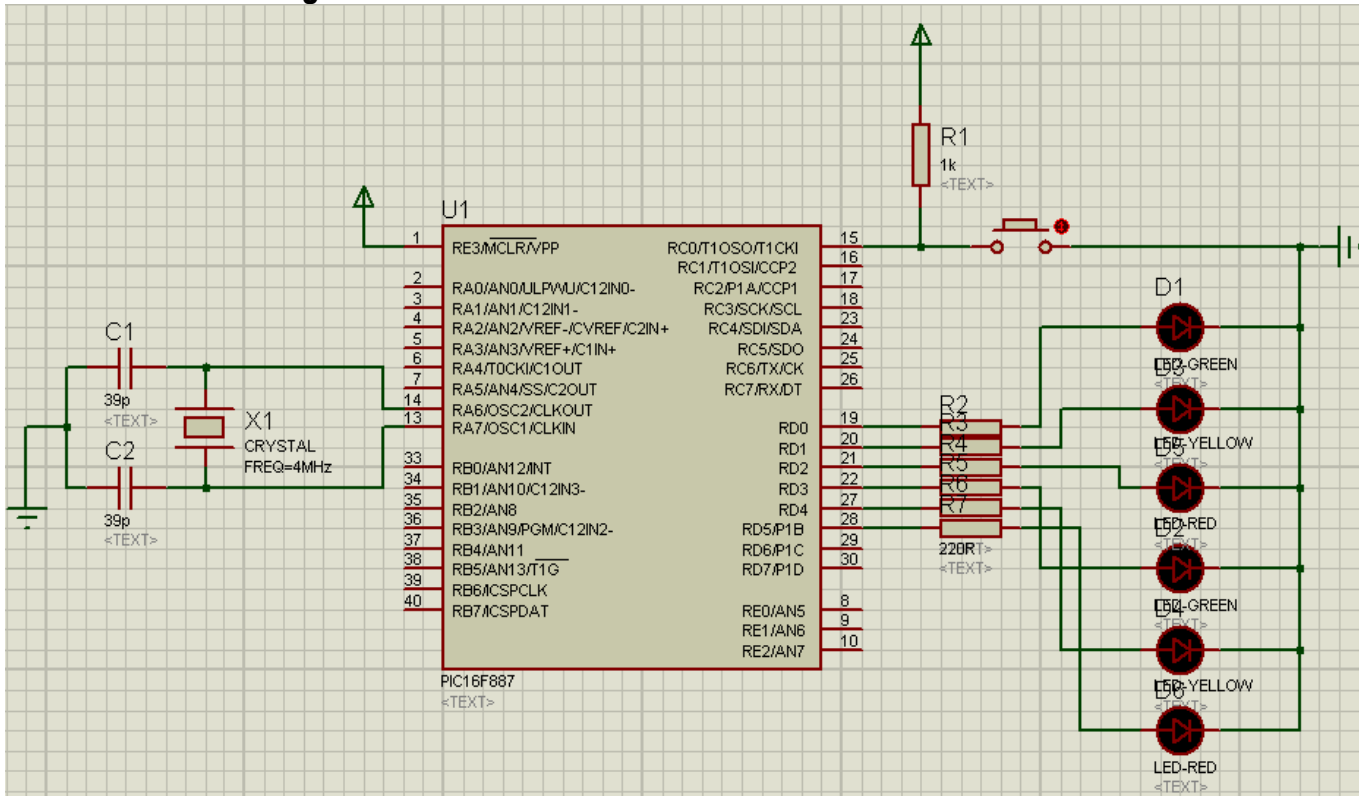


Figura 7.1

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

B. Crear un proyecto de nombre pra7 en la carpeta c:\PIC\ practica7. Los programas de cada ejercicio deben ser guardados con el nombre practica7X.asm con X= 1, 2, 3...,A.

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.

- Y seleccione los registros **PORTA, PORTB, PORTC, PORTD, PORTE, TRISA, TRISB, TRISC, TRISD, TRISE y W.**

F. Realizar el siguiente programa de salida de datos y observar su funcionamiento utilizando la opción watch y stopwatch del simulador y posteriormente grabarlo en el circuito de la figura 7.1

```

__CPD_OFF & __CP_OFF & __MCLRE_OFF & __CONFIG __CONFIG1, __LVP_OFF & __FCMEN_OFF & __IESO_OFF & __BOR_OFF &
__PWRTE_ON & __WDT_OFF & __XT_OSC
__CONFIG __CONFIG2, __WRT_OFF & __BOR21V
LIST P=16F887
#include <P16F887.INC>
CBLOCK 025

CONT1
CONT2
CONT3

                                ;CONFIGURAR COMO SALIDA
                                ;LIMPIA PUERTO D
MOVW B'00000000'
MOVWF PORTD
BSF STATUS,RP0 ;BANCO 1
MOVW B'00000000'
MOVWF TRISD ;PORTD COMO SALIDA
BCF STATUS,RP0 ;BANCO 0
                                ;PROGRAMA PRINCIPAL

CORRIMIENTO DE TRES BITS DEL PUERTO D
INICIO:
MOVW 00H ;LIMPIA PUERTO D
MOVWF PORTD
BSF PORTD,0 ;ENCIENDE BIT 0 DEL PUERTO D
CALL RETARDO1
NOP
BCF PORTD,0 ;APAGA BIT 0 DEL PUERTO D
BSF PORTD,1 ;ENCIENDE BIT 1 DEL PUERTO D
CALL RETARDO1
NOP
BCF PORTD,1 ;APAGA BIT 1 DEL PUERTO D
BSF PORTD,2 ;ENCIENDE BIT 2 DEL PUERTO D
CALL RETARDO1
NOP
BCF PORTD,2 ;APAGA BIT 2 DEL PUERTO D
GOTO INICIO

RETARDO1:
MOVW D'9'
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

```

- **Explica que sucede en programa.**
- **Determine el Tiempo de la subrutina Retardo, Retardo2 y Retardo3 utilizando Stopwatch y fundamentarlo con cálculos.**

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.

Parte 2

1. **Realice un programa que efectúe la función de un semáforo en un cruce con las siguientes condiciones, tome en cuenta la conexión de la figura 7.1**

Verde 4 segundos

Amarillo 2 segundos

Rojo 6 segundos

Sugerencia: realice un tabla muestre la transición entre cada color, recuerde que existe dos led's del mismo color.

2. **Realice un programa que enciendan los 6 primeros bits del PUERTO D de manera ascendente durante 1 seg cada bit y al terminar enciendan de manera descendente durante 2 seg cada bit . El programa debe ser cíclico.**
3. **Realizar un programa que enciendan los bit del PUERTO D impares de manera descendente durante 1 seg y los bits pares ascendentemente durante 2 seg, deben encender intercalados es decir un impar y luego un par. El programa debe ser cíclico.**

4. Conclusiones

A. Realizar conclusiones de manera individual.

5. Cuestionario

- a) Como se configuran los puertos como salida.
- b) Como se configuran los puertos como entrada.
- c) Con que instrucción se mando un dato de salida a un puerto
- d) Como se calcula de manera general una subrutina de retardo.

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.