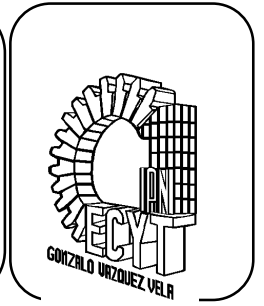


**INSTITUTO POLITÉCNICO NACIONAL**  
**Centro de Estudios Científicos y Tecnológicos N° 1**  
**“Gonzalo Vázquez Vela”**  
**Academia de Sistemas Digitales**  
**Prácticas de Micro Electrónica Programable**



NOMBRE DEL ALUMNO: \_\_\_\_\_  
 \_\_\_\_\_  
 Apellido Paterno

\_\_\_\_\_  
 Apellido Paterno                      Nombre

N° DE BOLETA: \_\_\_\_\_ GRUPO: \_\_\_\_\_

ASIGNATURA: **Micro Electrónica Programable**

HOJA	DE	FECHA			EVALUACION
		DIA	MES	AÑO	

PROFESOR: \_\_\_\_\_

## **Práctica 2**

### **Gestión de puertos de salida**

#### **Competencias de La Unidad:**

- Emplea el microcontrolador en la comunicación de datos y control de periféricos de forma multiplexada.

#### **Resultado de Aprendizaje Propuesto (RAP):**

- Usa diferentes elementos periféricos utilizando los puertos del microcontrolador de forma multiplexada.
- Utiliza los puertos del microcontrolador de forma bidireccional en la solución de un problema.

#### **Objetivos de la Práctica:**

1. Realizar la simulación de un programa para comprobar su funcionamiento utilizando herramientas computacionales
2. Desarrolla programas que habilite y deshabilite las salidas de microcontrolador por cierto tiempo.
3. Implementar un programa en un circuito basado en microcontrolador.

<b>Equipo Necesario</b>	<b>Material Necesario</b>
<p>Computadora (con el Software MPLAB IDE, IC-PROG o similar, compilador C, Simulador de circuitos electrónicos "Proteus")</p> <p>Programador tipo JDM o similar.</p>	<p>Instrucciones del PIC 16F887</p> <p>Microcontrolador PIC16F887 u otro de gama media o alta</p> <p>Capacitores</p> <p>LED's</p> <p>Resistencias</p> <p>pushbutton</p> <p>Cristal de cuarzo de 4MZ</p> <p>(Para los valores de estos elementos ver figura 2.1)</p>

## **Introducción Teórica**

### **Puertos de Entrada y salida**

El Microcontrolador 16F887/884 cuenta con 5 puertos direccionales, denominados A, B C,D y E y la versión 16F882/883/886 cuenta solo con los puertos denominados A,B y C.

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 y E no puede superar los 200 mA. y en los puertos C,D 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 en el PIC 16F887 están agrupadas en 5 puertos: A (8 bits), B (8 bits), C (8 bits). D (8 bits ) y E (4 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 para realizar una lectura de los bits, así como modificar los bits configurados como 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 se realiza las siguientes instrucciones en ensamblador

### **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. Al cambiar al banco 1, configure todos los puertos. En el caso del PIC16F887, debe configurarse los pines del Puerto A como entradas digitales colocan el valor 00H en el registro ANSEL

En el caso del compilador de C, los paso anteriores se realizan utilizando las directivas

```
#use standard_io(X)
set_tris_X(0x00);
```

## Bits de Configuración

Todos los PIC disponen de un cierto número de bits de configuración que están disponibles en la memoria programa, 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 generales 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
-	-	<i>DEBUG</i>	LVP	FCMEN	IESO	BOREN1	BOREN0	<i>CPD</i>	<i>CP</i>	MCLRE	<i>PWRT</i>	WDTE	FOSC2	FOSC1	FOSCO

<i>DEBUG</i>	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
$\overline{CPD}$		Codigo de protección de datos
	1	Deshabilitado
	0	Habilitado
$\overline{CP}$		Proteccion de memoria FLASH del programa
	1	Deshabilitado
	0	Habilitado
MCLRE		Funcion del pin RE3/ $\overline{MCLRE}$
	1	Funcion en $\overline{MCLRE}$
	0	Funciona como entrada digital, $\overline{MCLRE}$ lo toma de VDD
$\overline{PWRT\overline{E}}$		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

## Registro de configuración de la direccion2008H

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
-	-	-	-	-	WRT1	WRT0	BOR4V	-	-	-	-	-	-	-	-

WRT <1:0> Bits de habilitación para la seguridad de la memoria de programa

### **PIC16F883/PIC16F884**

00 = 0000h a 07FFh protegido contra escritura, 0800h a 0FFFh puede ser modificado por el registro de control EECON

01 = 0000h a 03FFh protegido contra escritura, 0400h a 0FFFh puede ser modificado por el registro de control EECON

10 = 0000h a 00FFh protegido contra escritura, 0100h a 0FFFh puede ser modificado por el registro de control EECON

11 = protegido contra escritura deshabilitado

### **PIC16F886/PIC16F887**

00 = 0000h a 0FFFh protegido contra escritura, 1000h a 0FFFh puede ser modificado por el registro de control EECON

01 = 0000h a 07FFh protegido contra escritura, 0800h a 1FFFh puede ser modificado por el registro de control EECON

10 = 0000h a 00FFh protegido contra escritura, 0100h a 1FFFh puede ser modificado por el registro de control EECON

11 = protegido contra escritura deshabilitado

### **PIC16F882**

00 = 0000h a 03FFh protegido contra escritura, 0400h a 07FFh puede ser modificado por el registro de control EECON

01 = 0000h a 00FFh protegido contra escritura, 0100h a07FFh puede ser modificado por el registro de control EECON

11 = protegido contra escritura deshabilitado

BOR4V Bits de selección de Reset por fallo de alimentación  
 0 = Reset por fallo de alimentación habilitado a 2.1V  
 1 = Reset por fallo de alimentación habilitado a 24.0V

En el caso del PIC 16F887 los bits de configuración se encuentran en la dirección 2007H y 2008H, en el caso de la dirección 2007H se puede almacenar el valor E3C1H y el 3EFFH en la 2008H. En el caso de requerir utilizar un dispositivo diferente, se debe verificar en las hojas de especificaciones, cuáles son los bits y posición en la cual están implementados.

El registro de la dirección 2007H quedaría de la siguiente manera:

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

El registro de la dirección 2008H se define como sigue:

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
-	-	-	-	-	WRT1	WRT0	BOR4V	-	-	-	-	-	-	-	-
1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1

Las características más relevantes definidas son las siguientes:

Oscilador: XT  
 Watchdog (WDT) Deshabilitado (Apagado)  
 PWRT: (PUT) Deshabilitado (Apagado)  
 CP: Deshabilitado (Apagado)  
 CPD: Deshabilitado (Apagado)  
 BODEN Habilitado (Encendido)  
 LVP Deshabilitado (Apagado)  
 MCLRE Terminal de reset deshabilitada,  $\overline{MCLRE}$  lo toma de VDD  
 Los bits sin implementar se leen como '1'

**Nota:** Para modificar los bits de configuración, por lo cual el ensamblador de MPASM tiene las características de permitir definir los valores que tomarán al programar el dispositivo, mediante la directiva `__CONFIG`

Enseguida se muestra la forma de utilizarlo

```
__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
```

que equivalen a almacenar el valor E3C1H en dirección 2007H y el 3EFFH en la dirección 2008H para 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 incluye de Microchip.

Pero para el caso del compilador de C existe una directiva similar a `__CONFIG` denominada `#fuses` expresada como sigue:

```
#fuses XT,NOWDT,NOPUT,NOMCLR,NOPROTECT,NOCPD,NOBROWNOUT,NOIESO,NOFCMEN,NOLVP
```

La cual permite almacenar el valor E3C1H en dirección 2007H y el 3EFFH en la dirección 2008H, para el caso del PIC 16F887.

## Instrucciones de salida

Las instrucción para enviar información hacia exterior es la siguiente:

`OUTPUT_X(Valor)` Envía el valor correspondiente (0-255) al puerto indicado en X

## ACTIVIDADES PREVIAS

- **Crear un proyecto de nombre pra2 en la carpeta c:\MEPIC\practica2 en MPLAB o PIC C Compiler. Los programas de cada ejercicio deben ser guardados con el nombre practica2X.c con X= 1, 2, 3...,A.**
- **En el caso de utilizar MPLAB, realizar los siguientes pasos:**
  - a. **Utilizar Project wizard y seleccionar el compilador de c**
  - b. **Agregar al proyecto los archivos adecuados con extensión c y h.**
  - c. **Habilitar Simulador MPLAB 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**
- **Si usa PIC C compiler crear el proyecto únicamente e incluir un archivo nuevo.**

# ACTIVIDADES PRÁCTICAS

## Parte 1

### 1. Armar el siguiente circuito

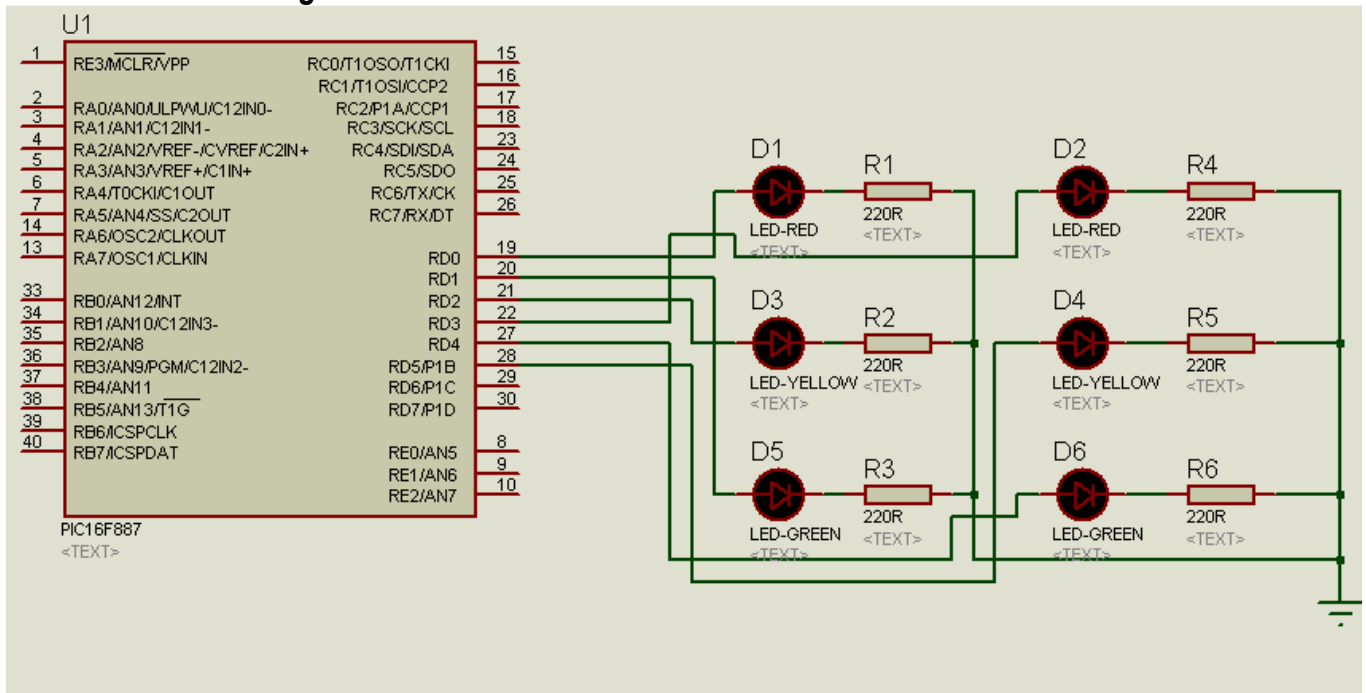


Figura 2.1

**Nota:** La terminal 12 o 31 del PIC16F887 se conectan a tierra.

**Realizar el siguiente programa de salida de datos y observar su funcionamiento mediante el simulador PROTEUS y posteriormente grabarlo en el circuito de la figura 2.1**

//Código de ejemplo

```
#include <16f887.h>
#use delay(clock=4000000)
#fuses XT,NOWDT,NOPUT,NOMCLR,NOPROTECT,NOCPD,NOBROWNOUT,NOIESO,NOFCMEN,NOLVP
```

```
#use standard_io(d)
```

```
void main()
```

```
{
```

```
set_tris_d(0x00);
```

```
while(true)
```

```
{
```

```
output_d(0x00);
```

```
output_d(0x01);
```

```
delay_ms(500);
```

```
output_d(0x00);
```

```
output_d(0x02);
```

```
delay_ms(500);
```

```
output_d(0x00);
```

```
output_d(0x04);
```

```
delay_ms(500);
```

```
output_d(0x00);
```

```
}
```

```
}
```

- **Nota:** A grabarlo **deshabilitar WDT y LVP y habilitar PWRTE y BODEN en la palabra de configuración, además recuerde seleccionar el tipo de oscilador a XT**

## Parte 2

- I. **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 2.1 y efectúe la simulación en PROTEUS**

<b>Verde</b>	<b>4 segundos</b>
<b>Amarillo</b>	<b>2 segundos</b>
<b>Rojo</b>	<b>6 segundos</b>

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

- II. **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. y efectúe la simulación en PROTEUS**

## 2. Conclusiones

A. *Realizar conclusiones de manera individual.*

## 3. Cuestionario

- a) Menciona la estructura función para realizar una subrutina de tiempo
- b) ¿Qué función tiene la sentencia while?
- c) ¿Cuál es la instrucción para enviar datos por el puerto?
- d) Menciona que función tiene la directiva #fuses y #use delay
- e) ¿Qué diferencia existirían entre el PIC16F887 y el PIC 18F4520 ?
- f) ¿Cuántos puertos cuenta el PIC18F4520?
- g) Menciona las características de los puertos del PIC18F4520

## 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.