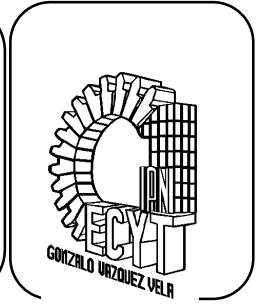


**INSTITUTO POLITÉCNICO NACIONAL**  
**Centro de Estudios Científicos y Tecnológicos N° 1**  
**“Gonzalo Vázquez Vela”**  
**Academia de Sistemas Digitales**  
**Prácticas de Micro Electrónica Programable**



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

---

Apellido Paterno Nombre

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

ASIGNATURA: **Micro Electrónica Programable**

HOJA	DE	FECHA	EVALUACION
		DÍA    MES    AÑO	

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

## **Práctica 1**

### **Subrutinas de tiempo**

#### **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. Implementar subrutinas de tiempo de diferentes valores.
2. Determinar con el simulador el tiempo total de subrutina de tiempo.
3. Realizar la simulación de un programa en ensamblador para comprobar su funcionamiento
4. Desarrolla programas que habilite y deshabilite las salidas de microcontrolador por cierto tiempo.
5. 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 )</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</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 1.1)</p>

## **Introducción Teórica**

### **Subrutinas**

Existen secuencias de instrucciones que son usadas por el programa principal varias veces. Entonces, **una subrutina es un conjunto de instrucciones que se agrupan para realizar un función específica, y se escriben una sola vez dentro del programa principal. Pero pueden ser ejecutadas varias veces.**

Las subrutinas son subprogramas dentro un programa principal, indicados con un nombre específico (etiqueta), Para llamar una subrutina en los microcontroladores PIC se utiliza la instrucción **CALL (llamar)** y para indicar que la subrutina ha finalizar se utiliza la instrucción **RETURN (regresar)**.

### **Subrutinas de Retardo**

Por lo general, cuando se requiere la salida o entrada de datos, es conveniente dentro del programa provocar tiempos de retardo; para permitir que los dispositivos respondan en un tiempo determinado Por lo tanto, un retardo, es una forma de control de tiempo en la programación del PIC,

Las instrucciones 'simples' utilizan un ciclo de máquina para ejecutarse, un ciclo máquina es la unidad básica de tiempo de ejecución de un programa en un PIC y depende de la velocidad del oscilador.

Hay instrucciones llamadas de salto como goto, return, call, btfss etc. que necesitan 2 ciclos máquina para ejecutarse. Si contamos los ciclos máquina de un determinado número de instrucciones del programa, podremos controlar los tiempos de retardo.

Como sabemos  $F=1/T$ , siendo F=frecuencia y T= tiempo.

Por consecuencia, podemos determinar cuánto tiempo consumirá una instrucción en el microcontrolador, sabiendo que para ejecutar una instrucción se utiliza un ciclo de maquina (CM) que equivale a 4 pulsos de reloj.

Para nuestro caso: Si el microcontrolador funciona a 4MHz, entonces

$F=1/T$  por lo tanto  $T=1/F$

Si  $F=4\text{MHz}$

$T=1/F = 1 / 4 \text{ Mhz}=0.25\mu\text{seg}$

Es decir que para un reloj de 4 MHz, cada instrucción simple ( $1 \text{ CM}= 4*T$ ) tardará 1  $\mu\text{seg}$ , ( $T_{\text{CM}}=1 \mu\text{seg}$ ) y para las instrucciones de salto (2 CM) tardará 2 $\mu\text{seg}$ .

El algoritmo general para obtener la subrutina de retardo consiste en los siguientes pasos:

1. Cargar un dato k en un registro.
2. Decrementar el registro y verificar si no es cero
3. Si es cero ir a paso 4, si no regresar a paso 2
4. Terminar subrutina

El código en ensamblador es el siguiente

```
Retardo          ;LA LLAMADA CALL APORTA 2 CICLOS
                 MOVLW  d'249'      ;APORTA 1 CICLO DE MAQUINA, "K= 249"
                 MOVWF  CONT1       ;APORTA 1 CICLO DE MAQUINA
CICLO            DECFSZ  CONT1,1     ;(K-1)x 1 CM (CUANDO NO SALTA),
                 ;2 AL SALTAR
                 GOTO   CICLO       ;APORTA (K-1)x2 CM
                 NOP                ;1 CICLO DE MAQUINA
                 RETURN             ;EL RETORNO APORTA 2
```

El tiempo total de la subrutina es:

$2 + 1 + 1 + (K-1) \times 1 + 2 + (K-1) \times 2 + 1 + 2 = 6 + 3K = N$  ciclos de maquina

Tiempo total es:  $N$  ciclos de maquina \*  $(4*T) = N$  ciclos de maquina \*  $(T_{\text{CM}})$

Para poder obtener retardos de mayor tiempo únicamente se tiene que realizar estructura básica de manera anidada.

### 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
-	-	$\overline{DEBUG}$	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

## 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 a 07FFh 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
-	-	DEBUG	LVP	FCMEN	IESO	BOREN1	BOREN0	CPD	CP	MCLRE	PWRTÉ	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'

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`

En seguida 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
  
```

La directiva anterior define los siguientes de bits de configuración:

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

## ACTIVIDADES PREVIAS

- **Crear un proyecto de nombre pra1 en la carpeta c:\MEPIC\ practica1. Los programas de cada ejercicio deben ser guardados con el nombre practica1X.asm con X= 1, 2, 3...,A.**
- **Habilitar Simulador MPLAB SIM y modificar la frecuencia del simulador a 4 Mhz.**
- **Utilizaremos la herramienta de stopwatch, para obtener la elija Debugger >> Stopwatch.**
- **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**

## ACTIVIDADES PRÁCTICAS

### Parte 1

1. Implementar la subrutina de retardo básica y con ayuda del simulador obtener el tiempo mínimo y máximo.

Sugerencia: Utilice la herramienta stopwatch y el modo paso a paso

2. Con ayuda del simulador crear subrutinas de retardo de 600ms, 1 seg y 60 seg.

A. Escribir el código en ensamblador y utilizando stopwatch mostrar su resultado

### Parte 2

#### A. Armar el siguiente circuito

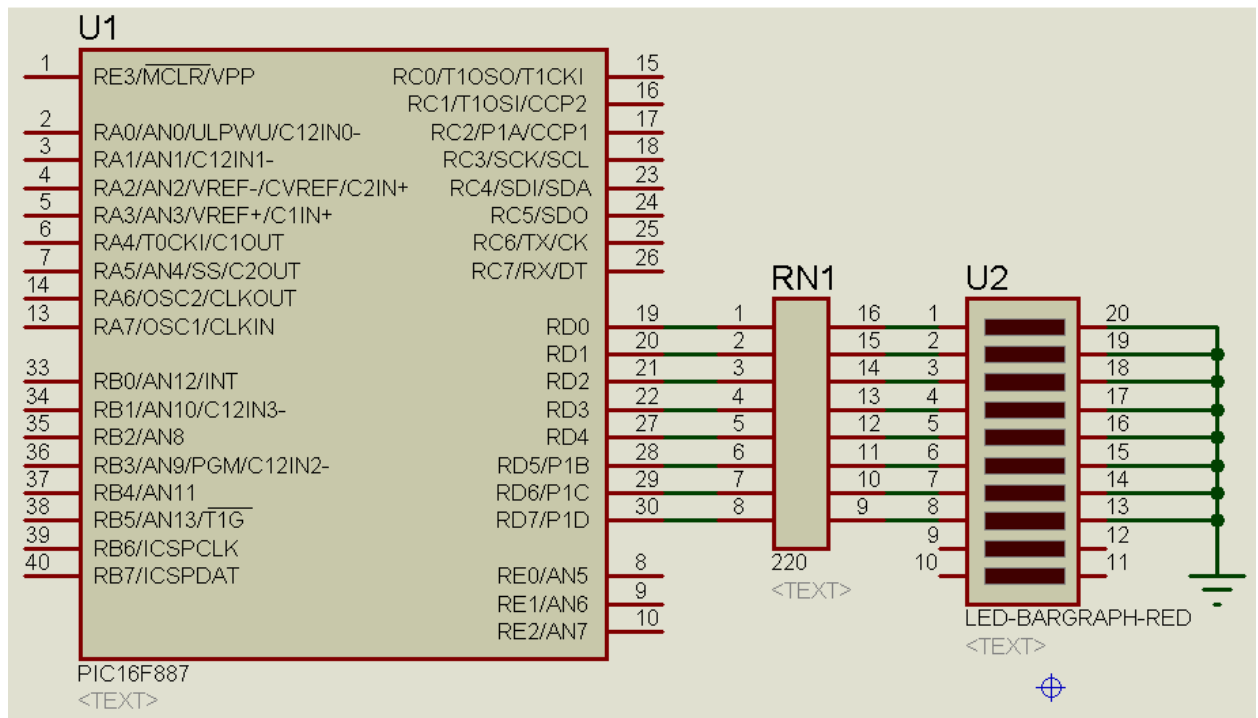


Figura 1.1

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

## B. Compile el siguiente programa y grábelo en circuito de la figura 1.1

```
__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'00000000'
MOVWF PORTD

BSF STATUS,RP0
MOVLW B'00000000'
MOVWF TRISD
BCF STATUS,RP0

INICIO
MOVLW B'00000000'
MOVWF PORTD
BSF PORTD,0
CALL RETARDO1S
BCF PORTD,0
BSF PORTD,1
CALL RETARDO1S
BCF PORTD,1

BSF PORTD,2
CALL RETARDO1S
BCF PORTD,2
GOTO INICIO

RETARDO1S
MOVLW D'6'
MOVWF CONT3

CICLO3
CALL RETARDO196MS
DECFSZ CONT3,1
GOTO CICLO3
NOP
RETURN

RETARDO196MS
MOVLW D'255'
MOVWF CONT2

CICLO2
CALL RETARDO771US
DECFSZ CONT2,1
GOTO CICLO2
NOP
RETURN
```



```

RETARDO771US
    MOVLW D'255'
    MOVWF CONT1
CICLO
    DECFSZ CONT1,1
    GOTO CICLO
    NOP
    RETURN

    END

```

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

**C. Modifique el programa anterior para los led's que prendan y apaguen cada 2 seg.**

**D. Modifique el programa del inciso B para que el led conectado a la terminal RD0 encienda y apague durante 500ms cuatro veces, enseguida el led conectado a la terminal RD1 encienda y apague durante 2 segundos 5 veces, y finalmente el led RD2 encienda y apague durante un segundo 3 veces. Todo el programa debe repetirse 4 veces y esperar 2 segundos para volver a ejecutarse.**

### 3. Conclusiones

A. Realizar conclusiones de manera individual.

### 4. Cuestionario

- Menciona la estructura básica de una subrutina de tiempo
- Diseñe la estructura de una subrutina que contenga tres estructuras básicas de retardo anidadas
- Diseña subrutinas de 10, 30 y 90 segundos.
- ¿Cuál es la función de la instrucción BSF?
- ¿Cuál es la función de la instrucción BCF?
- ¿Cuál es la función de la instrucción DECFSZ?
- ¿Cuántos puertos cuenta el PIC16F887?
- Menciona las características de los puertos del PIC16F887

### 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.