NOMBRE DEL ALUMNO: _____
Apellido Paterno

_____
Apellido Paterno Nombre

N° DE BOLETA: _____ GRUPO: _____

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

HOJA	DE

FECHA		
DIA	MES	AÑO

EVALUACION

PROFESOR: _____

Práctica 6

Subrutinas de tiempo e Inicialización de Puertos 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. Implementar subrutinas de tiempo de diferentes valores.
2. Determinar con el simulador el tiempo total de subrutina de tiempo.
3. Conocer la forma de configurar los puertos como entrada y como salida.
4. Realizar la simulación de un programa en ensamblador para comprobar su funcionamiento
5. Desarrolla programas que habilite y deshabilite las salidas de microcontrolador por cierto tiempo.
6. Implementar un programa en un circuito basado en microcontrolador.

Equipo Necesario	*Material Necesario*
Computadora (con el Software MPLAB IDE, IC-PROG o similar) Programador tipo JDM o similar.	Instrucciones del PIC 16F887 MIcrocontrolador PIC16F887 u otro de gama media Capacitores LED's Resistencias pushbutton Cristal de cuarzo de 4MZ (Para los valores de estos elementos ver figura 6.1)

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 subprogarmas dentro un programa principal, indicados con un nombre especifico (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=4MHz
T=1/F = 1 / 4 Mhz=0.25µseg

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

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 paso2
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)x1 + 2 + (K-1)x2 +1+ 2 = 6 + 3K = N ciclos de maquina

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

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

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 ansel
      clrf ansel
      banksel trisa
      movlw B'11111111'
      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.
```


Dr. Oscar Octavio Gutiérrez Frías
Academia de Sistemas Digitales T.V.

```
;Inicializa los puerto A como salida


      movlw B'00000000' ;Limpia Puerto A
      movwf PORTA
      movlw B'0000000'
      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	$\overline{PWRTE}$	WDTE	FOSC2	FOSC1	FOSC0

$\overline{DEBUG}$	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
$\overline{CPD}$	Codigo de protección de datos
	1 Deshabilitado
	0 Habilitado
$\overline{CP}$	Proteccion de memoria FLASH del programa
	1 Deshabilitado
	0 Habilitado

Dr. Oscar Octavio Gutiérrez Frías
Academia de Sistemas Digitales T.V.

MCLRE Funcion del pin RE3/$\overline{MCLRE}$

1 Funcion en $\overline{MCLRE}$

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

$\overline{PWRTE}$ 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	$\overline{PWRTE}$	WDTE	FOSC2	FOSC1	FOSC0
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'

ACTIVIDADES PREVIAS

- ***Crear un proyecto de nombre pra6 en la carpeta c:\PIC\practica6. Los programas de cada ejercicio deben ser guardados con el nombre practica6X.asm con X= 1, 2, 3…,A.***
- ***Habilitar Simulador MPLSB SIM y modificar la frecuencia del simulador a 4 Mhz.***
- ***Utilizaremos la herramienta de stopwatch, para obtener la elija Debugger >> Stopwatch.***

Dr. Oscar Octavio Gutiérrez Frías
Academia de Sistemas Digitales T.V.

- ***Obtener la herramienta de watch, de la siguiente manera View>> watch.***
- ***Y seleccione los registros PORTA, PORTB, PORTC, TRISA, TRISB, TRISC 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, 2seg, 10 seg y 1 minuto.***

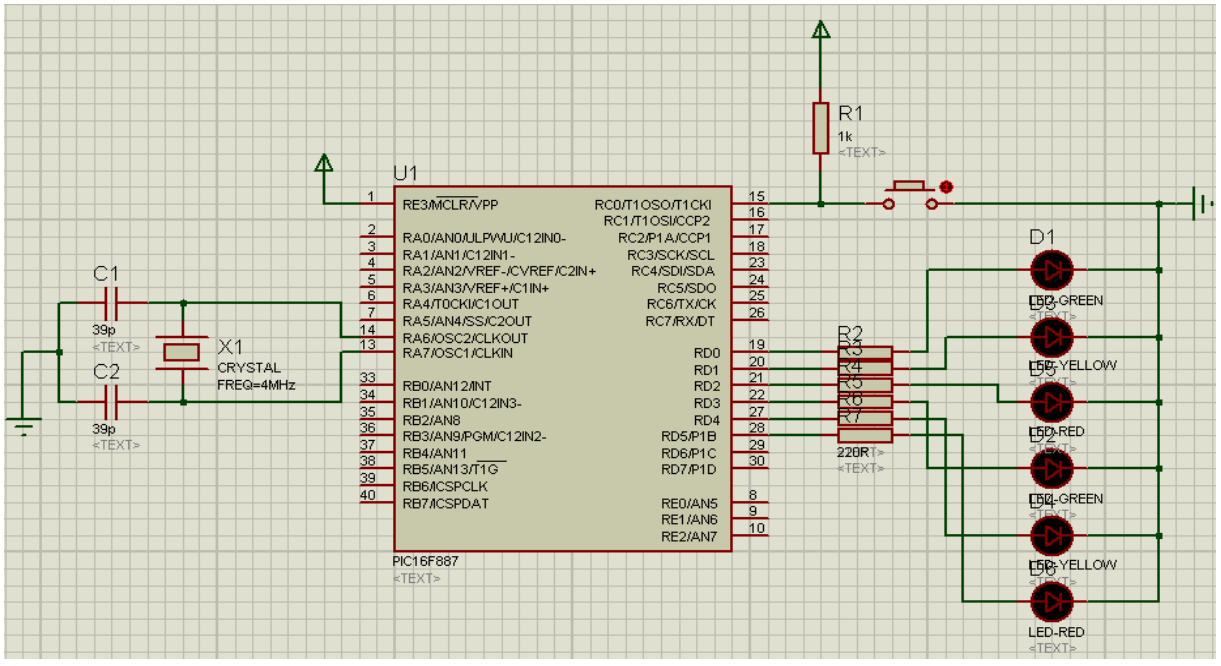
   A. Escribir el código en ensamblador y utilizando stopwatch mostrar su resultado-

3. ***Crear los códigos que configuren los puertos de la siguiente manera:***

   i. Puerto B como salida y los primeros 4 pines del puerto A como entrada
   ii. Puerto B como salida y Puerto C como entrada
   iii. Puerto B como salida y Puerto C como salida

Parte 2

A. Armar el siguiente circuito



Dr. Oscar Octavio Gutiérrez Frías
Academia de Sistemas Digitales T.V.

Figura 6.1

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

B. Compile el siguiente programa y grábelo en circuito de la figura 6.1

```
                    LIST P=16F887
                    #INCLUDE <P16F887.INC>
                    CBLOCK 025
CONT1
CONT2
CONT3
                    ENDC
                                                        ;CONFIGURAR COMO SALIDA
                    MOVLW B'00000000'           ;LIMPIA PUERTO D
                    MOVWF PORTD
                    BSF STATUS,RP0              ;BANCO 1
                    MOVLW B'00000000'
                    MOVWF TRISD                         ;PORTD COMO SALIDA
                    BCF STATUS,RP0              ;BANCO 0
                                                        ;PROGRAMA PRINCIPAL
INICIO:
                    MOVLW B'00000000'           ;LIMPIA PUERTO D
                    MOVWF PORTD
                    BSF PORTD,0
                    CALL RETARDO1S
                    BCF PORTD,0
                    BSF PORTD,1
                    CALL RETARDO1S
                    BCF PORTD,1
                    GOTO INICIO
RETARDO1S:
                    MOVLW D'6'
                    MOVWF CONT1
CICLO1:
                    CALL RETARDO2
                    DECFSZ CONT1,1
                    GOTO CICLO1
                    RETURN
RETARDO2:
                    MOVLW D'216'
                    MOVWF CONT2
CICLO2:
                    CALL RETARDO3
                    DECFSZ CONT2,1
                    GOTO CICLO2
                    RETURN
RETARDO3:
                    MOVLW D'255'
                    MOVWF CONT3
CICLO3:
                    DECFSZ CONT3,1
                    GOTO CICLO3
                    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 se ejecute 4 veces y espere 2 seg. para volver a repetirse.

4. Conclusiones

A. Realizar conclusiones de manera individual.

5. Cuestionario

a) Menciona la estructura básica de una subrutina de tiempo
b) Diseñe la estructura de una subrutina que contenga tres estructuras básicas de retardo anidadas
c) Diseña subrutinas de 10, 30 y 90 segundos.
d) Menciona los pasos para configurar los puertos en los PIC de gama media.
e) ¿Cuál es la función de la instrucción BSF?
f) ¿Cuál es la función de la instrucción BCF?
g) ¿Cuál es la función de la instrucción DECFSZ?
h) ¿Cuántos puertos cuenta el PIC16F876?
i) Menciona las características de los puertos del PIC16F877

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.**