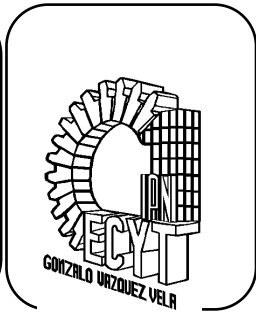


INSTITUTO POLITÉCNICO NACIONAL
 Centro de Estudios Científicos y Tecnológicos N° 1
 “Gonzalo Vázquez Vela”
 Academia de Sistemas Digitales
 Prácticas de Arquitectura de Microprocesadores y Microcontroladores



NOMBRE DEL ALUMNO: _____
 Apellido Paterno

_____ Apellido Paterno Nombre

N° DE BOLETA: _____ GRUPO: _____

UNIDAD APRE: **Arquitectura de Microp. y Microcontr**

HOJA	DE	FECHA			EVALUACION
		DIA	MES	AÑO	

PROFESOR: _____

Práctica 10

Manejo de Tabla de datos con 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 un convertidor de Binario a BCD mediante de elementos de entrada y de salida.
2. Realizar programa que obtenga datos de entrada, visualice datos en un display de 7 segmentos.
3. Comprender, la manera de grabar el Microcontrolador utilizado un programador.

Equipo Necesario	Material Necesario
<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</p> <p>Capacitores</p> <p>LED's</p> <p>Resistencias</p> <p>Display Cátodo Común</p> <p>Mnidip</p> <p>(Para los valores ver figura 10.1 y 10.2)</p>

Introducción Teórica

Tablas en los Microcontroladores PIC

Una tabla de datos de la memoria ROM es un conjunto de datos (Constantes) que se almacenan en la memoria EEPROM de datos de microcontrolador por lo cual la única manera de modificarlo es grabando de nuevo el dispositivo.

Es importante comentar, que las tablas de datos de ROM son diferentes a las tablas de datos de la memoria RAM que contienen variables almacenadas en los registro de la memoria de datos.

Por lo tanto, una tabla de datos es una lista de constantes que el programa puede obtener mediante la instrucción `retlw` o la directiva de ensamblador `DT`. A continuación se describe brevemente el uso de cada una de las opciones anteriores.

- **Instrucción `retlw`**

Esta instrucción produce un retorno de una subrutina (como la haría `RETURN`) pero con la diferencia de que regresa un valor constante al registro `W`

La forma de la instrucción es:

RETLW k

- **Directiva `DT`**

Esta directiva es la facilitada por el ensamblador para simplificar el uso de la instrucción `retlw`, esto significa que la sustituye el uso repetitivo de la instrucción.

Su sintaxis es:

`DT <Expr1>,<Expr2>,<Expr-3>,,,,,<Expr-n>`

Durante el proceso de ensamblado cada expresión se <expr> genera una instrucción retlw.

Ejemplo de uso de la instrucción retlw y DT,

1. RETLW

```
INTRUCCION 0
MOVLW 03H
CALL TABLA
INTRUCCION 3
```

```
TABLA ADDWF PCL,F
      RETLW 05H ; Configuración 0
      RETLW 07H ; Configuración 1
```

END

2. Directiva DT

```
INTRUCCION 0
MOVLW 03H
CALL Tabla
INTRUCCION 3
```

```
TABLA ADDWF PCL,F
      DT 05H,03H
```

END

Los ejemplo anteriores representa la misma función, que consiste en obtener un dato de la tabla para lo cual se utiliza al registro W como un índice de la tabla, el cual será sumado al contador de programa (PCL) mediante el salto indexado o relativo **ADDWF PCL,F**, posicionándolo en el valor requerido y extrayendo el valor mediante la instrucción retlw.

Nota: Por facilidad utilizaremos la directiva DT para generar tablas.

ACTIVIDADES PRÁCTICAS

Parte 1

- A. Crear un proyecto de nombre pra10 en la carpeta c:\PIC\practica10. Los programas de cada ejercicio deben ser guardados con el nombre practica10X.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, W y algún otro que necesite.
- G. Simular y grabarlos los siguientes programa de ejemplo y observe su funcionamiento en el circuito de la figura 10.1 y figura10.2

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, convertir el valor BCD obtenido mediante un minidip. Es decir si en el minidip todo están en cero debe aparecer un cero, si se tiene 0001 se tiene uno y así sucesivamente (Utilizando Directiva DT)

Implemente un archivo ASM , compílelo con ayuda de MPLAB y posteriormente grabarlo en el circuito de la figura 10.1, para observar su funcionamiento.

```

        __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
CONT4
CONT5
CONT6
DATO

                                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          ;PORTD COMO SALIDA
    MOVWF TRISC
    MOVLW B'11111111'    ;PORTA COMO ENTRADA
    MOVWF TRISA
    BCF STATUS,RP0      ;BANCO 0
    MOVLW 0X10
    MOVWF PORTC

INICIO:
    CLRF PORTD          ;LIMPIA PUERTO D Y APAGA EL

DISPLAY
LECTURA:
    MOVF PORTA,0
    COMF PORTA,0
    ANDLW 0X0F
    MOVWF DATO
    SUBLW D'9'
    BTFSS STATUS,C
    GOTO INICIO
    MOVF DATO,0
    CALL TABLA
    MOVWF PORTD
    GOTO LECTURA
    GOTO INICIO

;TABLA DE DATOS
TABLA:
    ADDWF PCL,F
    DT 3FH,06H,5BH,4FH,66H,6DH,7DH,07H,7FH,67H
    END

```

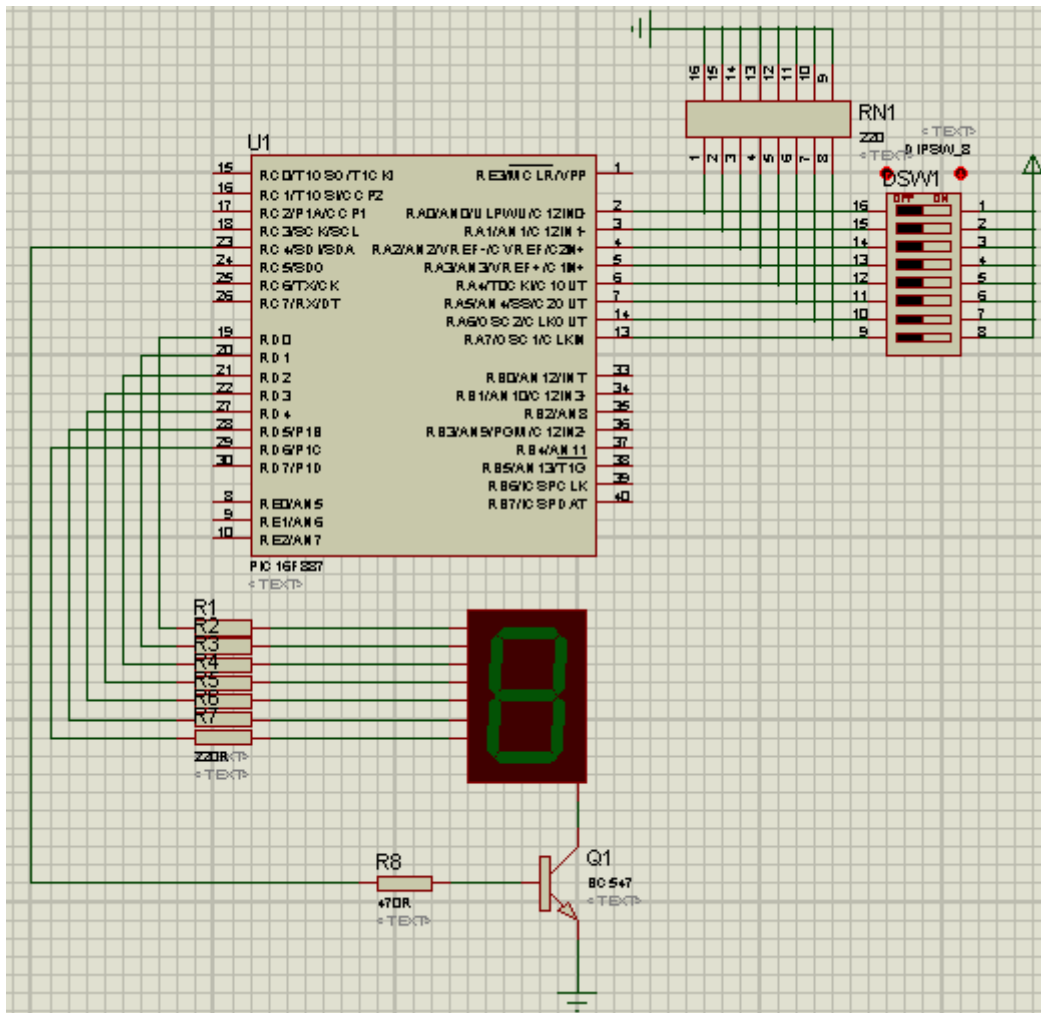


Figura 10.1

Programa Ejemplo 2:

El siguiente código permite realizar un programa que cada valor binario le asigne una letra del mensaje PRACTICADT y que al colocar en el minidip en algún valor se visualice la letra en el display durante 2 segundos y al terminar se visualice un guion durante un segundo y visualizar de nuevo el valor correspondiente a la lectura del minidip.

Implemente un archivo ASM, compílelo con ayuda de MPLAB y posteriormente grabarlo en el circuito de la figura 10.1, para observar su funcionamiento.

```

__CONFIG __CONFIG1, _LVP_OFF & _FCMEN_OFF & _IESO_OFF & _BOR_OFF &
_CPDP_OFF & _CP_OFF & _MCLR_OFF & _PWRTE_ON & _WDT_OFF & _XT_OSC
__CONFIG __CONFIG2, _WRT_OFF & _BOR21V
LIST P=16F887
#include <P16F887.INC>
CBLOCK 025

```

CONT1
CONT2
CONT3

ENDC

SALIDA PUERTO B 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
MOVLW B'11111111' ;PORTA COMO ENTRADA
MOVWF TRISA
BCF STATUS,RP0 ;BANCO 0
MOVLW 0X10
MOVWF PORTC

```

;PROGRAMA

PRINCIPAL VISUALIZACION DE VALORES DEL 0 AL 9 PRESIONANDO EL BOTON EN PIN-A0

INICIO:

```

CLRF PORTD ;LIMPIA PUERTOD Y APAGA

```

DISPLAY

GUION:

```

MOVLW 0X40
MOVWF PORTD
CALL RETARDO1S

```

LECTURA:

```

MOVF PORTA,0
COMF PORTA,0
ANDLW 0X0F
CALL TABLA
MOVWF PORTD
CALL RETARDO1S
CALL RETARDO1S
GOTO GUION
GOTO INICIO

```

;TABLA DE DATOS

TABLA:

```

ADDWF PCL,F
DT 40H,73H,50H,77H,39H,78H,06H,39H,77H,5EH,78H

```

RETARDO1S:

```

MOVLW D'12'
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

```

Programa Ejemplo 3:

El siguiente código permite realizar un conteo de forma ascendente, incrementando de uno en uno al presionar un botón conectado a RA0 y mostrándolo en un display de 7-segmentos conectado al puerto D (Utilizando Directiva DT)

Implemente un archivo ASM , compílelo con ayuda de MPLAB y posteriormente grabarlo en el circuito de la figura 10.2, para observar su funcionamiento.

```

        __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
CUENTA
CUENTA1

        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                    ;PORTD COMO SALIDA
        MOVWF TRISC
        MOVLW B'11111111'             ;PORTA COMO ENTRADA
        MOVWF TRISA
        BCF STATUS,RP0                ;BANCO 0
        MOVLW 0X10
        MOVWF PORTC

        ;PROGRAMA PRINCIPAL
        VISUALIZACION DE VALORES DEL 0 AL 9 PRESIONANDO EL BOTON EN PIN-A0
        INICIO:
                CLRF PORTD                ;LIMPIA PUERTO D Y APAGA DISPLAY
        REINICIO:
                MOVLW 40H                ;LIMPIA PUERTO D Y APAGA DISPLAY
                MOVWF PORTD
                CALL RETARDO197MS
        REST:
                CLRW
                MOVWF CUENTA1
                MOVLW D'7'
                MOVWF CUENTA
        LECTURA:
                CALL RETARDO197MS
                BTFSC PORTA,0
                GOTO LECTURA
                GOTO SIGUE
        SIGUE:
                MOVF CUENTA1,W
                CALL TABLA
                MOVWF PORTD
                INCF CUENTA1,1
                CALL RETARDO197MS

```



```

DEC:
    DECFSZ CUENTA,1
    GOTO LECTURA
    GOTO REST

TABLA:
    ADDWF PCL,F
    DT 3FH,06H,5BH,4FH,66H,6DH,7DH

RETARDO197MS:
    MOVLW D'2'
    MOVWF CONT1

CICLO1:
    CALL RETARDO2
    DECFSZ CONT1,1
    GOTO CICLO1
    RETURN

RETARDO2:
    MOVLW D'255'
    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

```

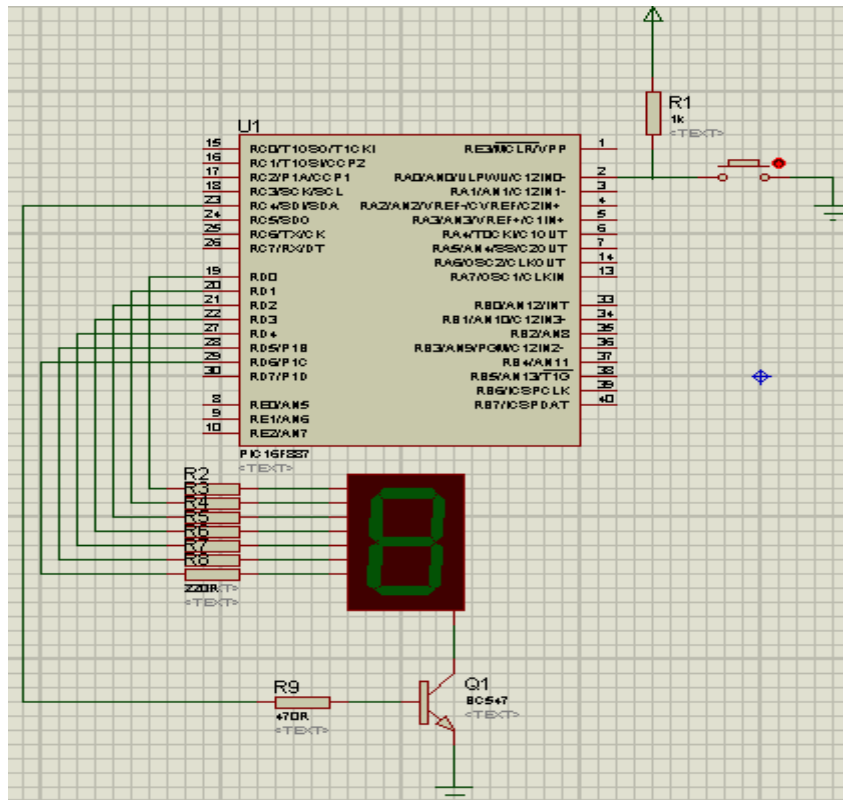


Figura 10.2

Parte 2:

1. **Crear un programa que realice un dado electrónico que:**
 - **Cuenta del 0 al 6 y al presionar el interruptor conectado a RA0 se mantenga el valor durante 2 segundos y continúe la cuenta.**

Implemente un archivo ASM, compílelo con ayuda de MPLAB y posteriormente grábalo en el circuito de la figura 10.2, para observar su funcionamiento

2. Conclusiones

A. Realizar conclusiones de manera individual.

3. Cuestionario

- a) Como se define una tabla de datos de memoria ROM.
- b) Donde se almacena los valores de las tablas de datos.
- c) Para qué sirve la instrucción retlw.
- d) Para qué sirve la directiva DT.
- e) Qué diferencia existe entre la directiva DT y retlw
- f) Con que instrucción se realiza un salto relativo o indexado

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.**