# MANUAL DE PRÁCTICAS DE SISTEMAS DIGITALES II

M.C. LIDIA HORTENCIA RASCON MADRIGAL

M.C. ERNESTO SIFUENTES DE LA HOYA

ASIGNATURA: SISTEMAS DIGITALES II

INSTITUTO DE INGENIERÍA Y TECNOLOGÍA

DEPARTAMENTO DE INGENIERÍA ELÉCTRICA Y COMPUTACIÓN


Avenida del Charro 450 Norte
Ciudad Juárez, Chihuahua

**Ficha Catalográfica**

**RASCON L., SIFUENTES E.**
**Manual de prácticas de Sistemas Digitales II**
Ciudad Juárez, Chih.
Universidad Autónoma de Ciudad Juárez
Año 2005.

Asignatura: Sistemas Digitales II
Páginas 74

Departamento de Ingeniería Eléctrica y Computación
Academia de Sistemas Digitales
Universidad Autónoma de Ciudad Juárez
Instituto de Ingeniería y Tecnología

# Índice

# Práctica 1
# Transistor como interruptor

## Objetivos

Al completar esta práctica el alumno:

- Será capaz de utilizar el transistor como interruptor para activar y desactivar una carga.
- Entenderá el concepto de corte y saturación de un transistor.
- Será capaz de hacer los cálculos necesarios para saturar un transistor

## Introducción

El transistor bipolar es un dispositivo semiconductor que permite el control y la regulación de una corriente grande mediante una señal muy pequeña. El uso del transistor en las zonas de corte y saturación permiten que éste funcione como un interruptor para activar o desactivar alguna carga.

### Zonas de Operación del transistor

**CORTE.-** No circula intensidad por la Base, por lo que, la intensidad de Colector y Emisor también es nula. La tensión entre Colector y Emisor es la de la batería. El transistor entre Colector y Emisor se comporta como un **interruptor abierto**.

$$I_B = I_C = I_E = 0;\ V_{CE} = V_{CC}$$

**SATURACION.-** Cuando por la Base circula una intensidad, se aprecia un incremento de la corriente de colector considerable. En este caso el transistor entre Colector y Emisor se comporta como un **interruptor cerrado**. De esta forma, se puede decir que la tensión de la batería se encuentra en la carga conectada en el Colector.

$$\uparrow I_B \Rightarrow \uparrow I_C;\ Vcc = R_C\ X\ I_C.$$

**ACTIVA.**- Actúa como **amplificador**. Puede dejar pasar más o menos corriente.

Cuando trabaja en la zona de corte y la de saturación se dice que trabaja en ***conmutación***. En definitiva, como si fuera un interruptor.

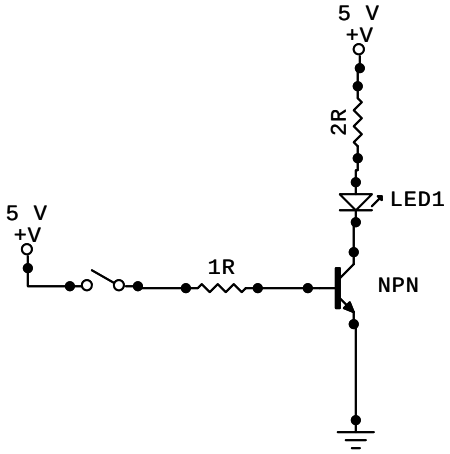
## Material y equipo utilizado

1 Transistor NPN que el alumno determine
1 Diodo LED
Resistencias de los valores calculados.
Multimetro con puntas

## Desarrollo

### 1. Encendiendo un LED.

a) Utilizando un transistor NPN que usted proponga, realice los cálculos necesarios para que cuando abra y cierre el interruptor el LED se apague y se encienda. Utilice el circuito de la figura.




Transistor propuesto: _____

Tipo: _____

Beta: _____

$IC_{SAT}$: _____

b) ¿Qué corriente necesita el LED para encender?

c) Calcule la Resistencia de base y la resistencia de colector.

RB _____
RC _____

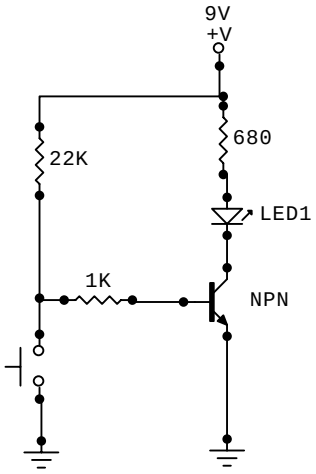
d) Proporcione la siguiente información.

| ***Parámetro*** | ***$V_B=0$*** | | ***$V_B=5$*** | |
|---|---|---|---|---|
| | ***Calculado*** | ***Medido*** | ***Calculado*** | ***Medido*** |
| $I_C$ | | | | |
| $I_B$ | | | | |
| $V_{BE}$ | | | | |
| $V_{CE}$ | | | | |

e) ¿Qué formulas utilizó?

**2. Arme el circuito siguiente**

a) Calcule el voltaje de carga y la corriente de carga.
b) Elija el transistor que pueda manejar la carga sin estresar al componente.




Icarga: _____

Vcarga: _____

Transistor propuesto: _____

Tipo: _____

Beta: _____

$IC_{SAT}$: _____

## Evaluación del aprendizaje

c) ¿En que zona trabaja el transistor cuando el interruptor está abierto?

d) ¿Que ocurre con el LED?

e) ¿Cuál es la corriente de la carga y el voltaje de la carga, mídalos?

f) ¿En que zona trabaja el transistor cuando el interruptor está cerrado?

g) ¿Que ocurre con el LED?

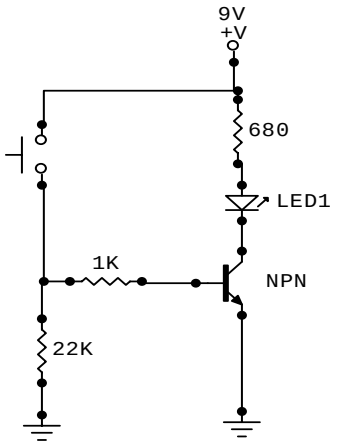
h) ¿Cuál es la corriente de la carga, mídala?

i) Llene la siguiente tabla con los valores correspondientes.

| ***Parámetro*** | ***Interruptor abierto*** | | ***Interruptor cerrado*** | |
|---|---|---|---|---|
| | ***Calculado*** | ***Medido*** | ***Calculado*** | ***Medido*** |
| $I_C$ | | | | |
| $I_B$ | | | | |
| $V_{BE}$ | | | | |
| $V_{CE}$ | | | | |

**3. Arme el circuito siguiente**

a) Calcule el voltaje de carga y la corriente de carga necesarios
b) Elija el transistor que pueda manejar la carga sin estresar al componente.




Icarga: _____

Vcarga: _____

Transistor propuesto: _____

Tipo: _____

Beta: _____

$Ic_{SAT}$: _____

## Evaluación del aprendizaje

c) ¿En que zona trabaja el transistor cuando el interruptor esta abierto?

d) ¿Que ocurre con el LED?

e) ¿Cuál es la corriente de la carga y el voltaje de la carga, mídalos?

f) ¿En que zona trabaja el transistor cuando el interruptor esta cerrado?

g) ¿Que ocurre con el LED?

h) ¿Cuál es la corriente de la carga y el voltaje de la carga, mídalos?

i) Llene la siguiente tabla con los valores correspondientes.

| ***Parámetro*** | ***Interruptor abierto*** | | ***Interruptor cerrado*** | |
|---|---|---|---|---|
| | ***Calculado*** | ***Medido*** | ***Calculado*** | ***Medido*** |
| $I_C$ | | | | |
| $I_B$ | | | | |
| $V_{BE}$ | | | | |
| $V_{CE}$ | | | | |

## Conclusiones individuales

# Práctica 2
# El transistor activando una carga de Corriente Alterna

## Objetivos

Al completar esta práctica el alumno:

- Estará capacitado para diseñar una interfaz de potencia utilizando un transistor y un relevador.
- Será capaz de activar y desactivar una carga de 120 VAC utilizando un transistor como interruptor.

## Introducción

El uso del transistor en las zonas de corte y saturación permiten que éste funcione como un interruptor para activar o desactivar cargas pequeñas o cargas que trabajen con la línea de alimentación comercial 120 VAC. Por ejemplo activar un foco o un ventilador.

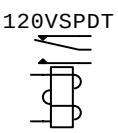
## Material y equipo utilizado

1 Transistor NPN BC137
1 Diodo de propósito general
Resistencias de los valores calculados.
1 Multimetro con puntas
1 Relevador de 12 o 24 V CC – 120 VCA
1 Foco de 120 V CA a 60 o 100W
1 Foco de 12 V DC

## Desarrollo

### 1. Encendiendo un foco de 12 V CC.

a) Identifique físicamente las terminales del relevador.



b) Conecte el voltaje a la bobina del relevador y verifique si éste conmuta.

c) Conecte un foco de 12 V CC y la fuente de 12 V CC a las terminales, de tal forma que cuando energice la bobina se prenda el foco. Dibuje el diagrama de conexiones.

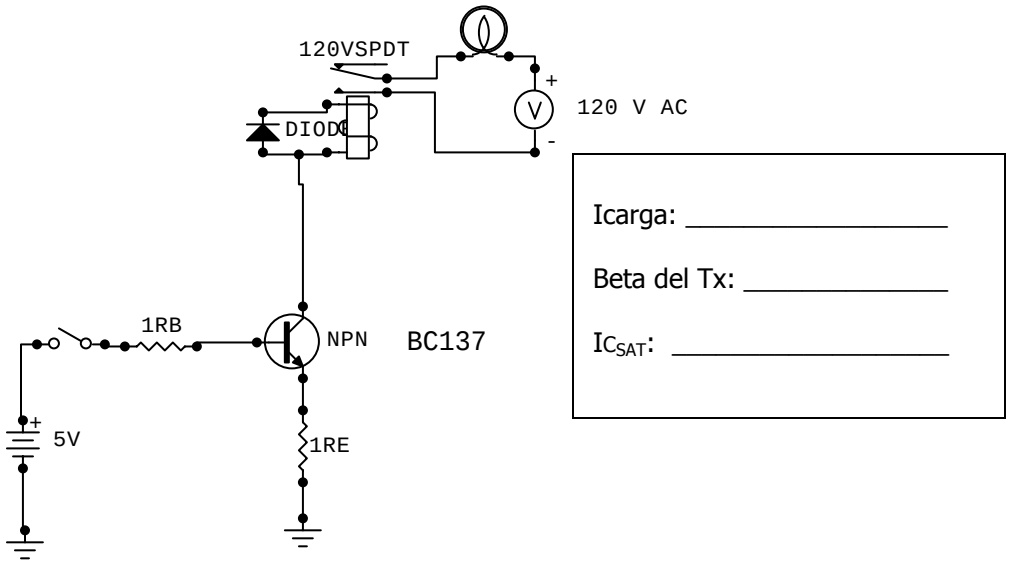
d) Conecte un foco de 12 V CC y la fuente de 12 V CC a las terminales de tal forma que cuando desconecte la bobina se prenda el foco. Dibuje el diagrama de conexiones.

e) ¿De que tipo es el relevador que está usando?

f) ¿Qué resistencia tiene el relevador?

g) ¿Cómo funciona un relevador?

**2. Para el circuito de la siguiente figura, Obtener los valores de RB y RE necesarios para activar y desactivar la carga (foco 120 VAC). Escriba sus cálculos.**




a) Arme el circuito sin conectar las fuente de 120 VAC y ni el foco de 120V. Verifique el relevador se active.
b) Si el relevador se activa. Conecte el foco de 120V AC y la línea de alimentación. Se cuidadoso. Evite accidentes. Energice el circuito.

## Evaluación del aprendizaje

c) ¿Cuál es la función del diodo que está en paralelo con la bobina?

d) ¿Qué pasa con el foco cuando el interruptor esta abierto?

e) ¿Qué pasa con el foco cuando el interruptor está cerrado?

## Conclusiones individuales

# Práctica 3
# Activación de una carga de Corriente Alterna desde la PC

## Objetivos

Al completar esta práctica el alumno:

- Estará capacitado para diseñar una interfaz de potencia utilizando un transistor y un relevador.
- Será capaz de utilizar la PC como elemento de control para activar y desactivar una carga de 120 VAC.
- Será capaz de implementar un programa en un lenguaje de programación de alto nivel que permita manejar el puerto paralelo de una PC.

## Introducción

Utilizando un lenguaje de programación como C o C++ es posible realizar un programa que permita enviar información a través del puerto paralelo de una PC. Es decir, se puede enviar un "uno" o un "cero" lógico por un pin del puerto. Este pin, puede a su vez activar la base de un transistor y hacer que éste funcione como un interruptor para activar o desactivar cargas a través de un relevador.

## Material y equipo utilizado

1 Transistor NPN 2N3904
1 Diodo de propósito general
2 Resistencias de 2.7 KΩ
1 Multimetro con puntas
1 Relevador de 12 o 24 V CC – 120 VCA
1 Foco de 120 V CA a 60 o 100W
Cables para conexión a la línea de 120 VAC

## Desarrollo

### 1. Manejo del puerto paralelo.

a) Realice un programa en C o C++ para enviar un número ya sea binario o hexadecimal al puerto paralelo. Utilice mascaras para manejar un solo pin usted elija el pin que quiere manejar.

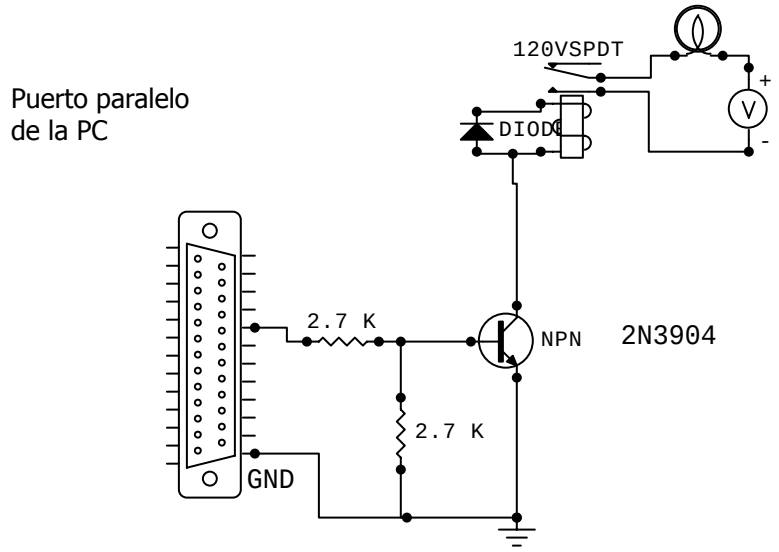
b) Pruebe el programa y verifique que el pin tenga los valores de bit enviado.

c) ¿Qué pin utilizo?

d) ¿Cuál es el pin de tierra del conector DB25?

e) Escriba el código del PROGRAMA

**2. Arme el circuito de la figura y verifique su funcionamiento conectando un voltaje de 0 o 5 V en lugar del pin de la computadora.**




**3. Si funciona bien el circuito conecte el pin de la computadora.**

a) Apague el foco, ¿Cuanto vale el bit?

b) Encienda el foco, ¿Cuanto vale el bit?

## Evaluación del aprendizaje

a) ¿Qué funciones del lenguaje de alto nivel utilizó para manipular el puerto paralelo de la PC?

b) ¿Qué dirección en hexadecimal tiene el puerto paralelo de la PC que está utilizando?

c) ¿Qué modificaciones tendría que hacer al programa para controlar el encendido y apagado de 8 cargas?

## Conclusiones individuales

# Práctica 4
# Activación una carga desde la PC utilizando un TRIAC

## Objetivos

Al completar esta práctica el alumno:

- Estará capacitado para diseñar una interfaz de potencia utilizando un TRIAC y un OPTOACOPLADOR.
- Será capaz de utilizar la PC como elemento de control para activar y desactivar una carga de 120 VAC.
- Será capaz de implementar un programa en un lenguaje de programación de alto nivel que permita manejar el puerto paralelo de una PC.

## Introducción

Hasta el momento se han activado y desactivado cargas con el relevador, sin embargo, éste puede ser reemplazado por un TRIAC para activar cargas de AC. También es posible realizar la activación de las cargas desde el puerto paralelo si se incluye un optotriac para aislar la PC de la corriente alterna y enviar el pulso de activación a la compuerta del TRIAC.

## Material y equipo utilizado

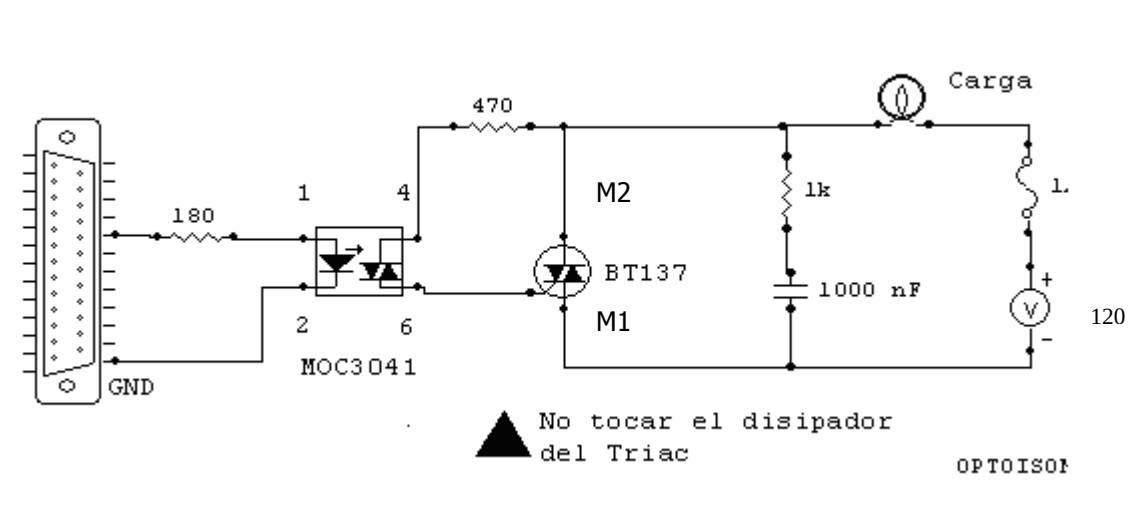
1 Triac BT137 8A 400V
1 MOC3041 0 MOC 3040
1 Resistencias de 180Ω, 1 de 470Ω, 1 de 1 KΩ
1 Capacitor de 1000 nF
1 Fusible
1 Foco de 120 V CA a 60 o 100W
Cables para conexión a la línea de 120 VAC

## Desarrollo

**1) Manejo del Triac.**

a) Verifique el funcionamiento del programa para enviar un bit por el puerto paralelo.
b) Armen el circuito de la figura. Sea muy cuidadoso con el manejo de la tierra digital.

NOTA. NO TOQUE EL DISIPADOR DEL TRIAC. EVITE ACCIDENTES.




c) Apague y encienda el foco mediante el programa realizado.

## Evaluación del aprendizaje

a) Explique con sus propias palabras el funcionamiento del circuito.

## Conclusiones individuales

# Práctica 5
# Convertidor Digital Análogo (DAC)

## Objetivos

Al completar esta práctica el alumno:

- Comprobará el funcionamiento de un DAC de 8 bits.
- Estará capacitado para elegir el DAC más adecuado para sus diseños posteriores.

## Introducción

Muchas variables físicas son de naturaleza analógica y pueden tomar cualquier valor dentro de un rango continuo. Ejemplos de variables de este tipo son: temperatura, presión, intensidad luminosa, posición, velocidad, impacto, fuerza, torsión, etc. Los sistemas digitales llevan a cabo todas sus operaciones internas mediante el uso de circuitería y operaciones digitales, por lo tanto, cualquier información que se tenga que introducir en un sistema digital, primero debe ponerse en forma digital. De manera similar las salidas de un sistema digital siempre son de naturaleza digital. Cuando un sistema digital, como una computadora o microcontrolador, va a ser utilizado para controlar algún proceso físico, el diseñador se encuentra con el problema de la diferencia entre la naturaleza digital y la analógica de las variables del proceso. Los convertidores análogo-digital (ADC) y digital-analógico (DAC) se utilizan para conectar los sistemas digitales con el mundo analógico de forma que éstos puedan controlar una variable física.

Básicamente, la conversión digital-analógica (DAC) es el proceso de tomar un valor digital (representado en código binario) y convertirlo en un voltaje o corriente que sea proporcional al valor digital.

## Materiales y equipo utilizado

1 DAC AD7524
1 Contador binario
Cables para conexión.
Componentes requeridos para el funcionamiento del DAC, según las especificaciones del fabricante

## Desarrollo

### 1. Análisis del DAC

a) Revise las hojas técnicas del DAC puede usar el AD7524, DAC0800 o algún otro.

b) Dibuje el diagrama para una operación binaria unipolar del DAC.

c) Armen el circuito de prueba, según el fabricante. Llene la siguiente tabla.

| ***Valor digital*** | ***Vreferencia (Volts)*** | ***Vo (en función de Vref)*** | ***Vo (Volts)*** |
|---|---|---|---|
| *1111 1111* | | | |
| *1001 0000* | | | |
| *1000 0000* | | | |
| *0001 0000* | | | |
| *0000 0001* | | | |
| *0000 0000* | | | |

d) Grafique el resultado obtenido.

**2. Contador binario de 8 bits.**

a) Seleccione un contador binario (el que usted conozca). Analice su funcionamiento.
b) Dibuje el diagrama de conexiones para un contador binario de 8 bits.

c) Arme el circuito
d) Pruebe que el contador funcione bien.

3. **Conecte el Contador al convertidor Digital Análogo**.

## Evaluación del aprendizaje

a) Ajuste el contador para que la cuenta cambie cada segundo
b) Mida el voltaje de salida del DAC con el voltímetro ¿Qué ocurre?

c) Incremente la velocidad de cuenta del contador y mida el voltaje de salida con el osciloscopio. ¿Qué observa?

d) Dibuje la grafica obtenida, ¿por qué resulto así?

e) ¿Por qué es negativo el voltaje de salida?

f) ¿Qué resolución tiene el DAC?

## Conclusiones individuales

# Práctica 6
# Monitoreo de temperatura a través de la PC y Un Convertidor Análogo Digital (ADC)

## Objetivos

Al completar esta práctica el alumno:

- Comprobará el funcionamiento de un ADC de 8 bits.
- Estará capacitado para elegir el ADC más adecuado para sus diseños posteriores.
- Será capaz de diseñar un sistema de monitoreo de cualquier variable física mediante una PC.

## Introducción

Un convertidor análogo-digital toma un voltaje de entrada analógico y después de cierto tiempo produce un código de salida digital que representa la entrada analógica.

El convertidor Análogo Digital es muy útil cuando se quiere alimentar una señal a un microcontrolador o a la computadora. Si la señal es análoga es necesario convertirla a digital para alimentarla al microcontrolador o computadora.

Como requisito se pide una investigación de características y manejo del puerto paralelo con lenguaje C.

## Materiales y equipo utilizado

1 ADC 0804
1 Sensor de temperatura LM335
1 Encendedor
1 termómetro (opcional)
Cables para conexión.
Componentes requeridos para el funcionamiento del DAC y el diseño del sistema

**Nota:** El ADC es un circuito muy sensible a la electricidad estática. Toque un cable aterrizado antes de remover el circuito de su esponja antiestática.

## Desarrollo

### 1. Análisis del sensor de temperatura

1. Revise las hojas técnicas del sensor de temparatura.

2. Dibuje y arme el diagrama de prueba (según el fabricante) y pruébelo.

3. Existe variación en el voltaje de salida al incrementar la temperatura cercana al sensor.

4. ¿De que magnitud es la variación observada?

5. Que ocurre el disminuir la temperatura cercana al sensor.

### 2. Análisis del ADC

a) Revise las hojas técnicas del ADC.
b) Dibuje y arme el circuito de prueba (según el fabricante) y pruébelo.

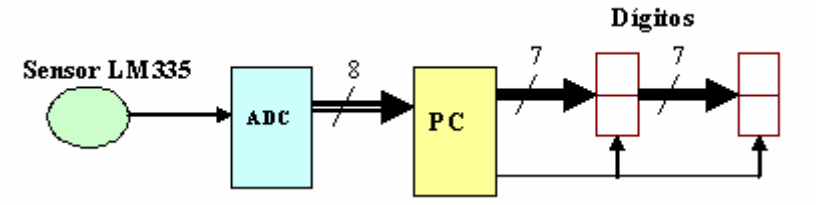
c) Pruebe el circuito con un voltaje de entrada en incrementos de 10 mV, ¿Qué resultados obtiene?. Utilice LEDs en la salida para ver los códigos generados por el ADC.

**3. Lectura de la temperatura digital a través del puerto paralelo de la computadora.**

a) Realice un programa para leer los datos de temperatura por el puerto paralelo (salidas del ADC).

b) Convierta los datos leídos a decimal y despliéguelos en la pantalla.

c) Conecte el sensor de temperatura al ADC y éste al puerto paralelo de la computadora, tomar el siguiente diagrama como referencia




d) Monitoree la temperatura del sensor en la computadora cada 5 segundos y despliegue su valor en 2 displays de 7 segmentos. Se quiere que la lectura de temperatura de desligue en un rango de 00 a 99 grados centígrados.

e) Dibuje el diagrama de conexiones completo.

## Evaluación del aprendizaje

a) ¿Cuántas líneas de entrada tiene el puerto paralelo?

b) ¿Qué lógica se utilizó para leer las 8 salidas del ADC por el puerto paralelo?

c) ¿Cuántas líneas de sañida tiene el puerto paralelo?

d) ¿Qué lógica se utilizó para conectar los displays al puerto paralelo?

## Conclusiones individuales

# Práctica 7
# Configuración de una comunicación serial RS-232

## Objetivos

Al completar esta práctica el alumno:

- Estará familiarizado con el protocolo RS-232-C, tanto en su parte física como en la parte lógica de control del hardware asociado.
- Será capaz de implementar un programa en lenguaje C para configurar el puerto serial de una PC.
- Establecerá una comunicación serial de PC a PC.

## Introducción

El puerto serial permite la comunicación de la PC con el mundo exterior, en este ejercicio se pretende que el alumno maneje la programación o configuración del puerto y realice las conexiones de hardware necesarias para comunicar dos computadoras y lograr una comunicación entre éstas en una especie de chat muy sencillo.

Como requisito previo se pide la lectura de la información sobre la interfaz serial RS-232 que se encuentra en el tutorial ***ptoserie.pdf*** (apuntes de clase unidad I.)

## Materiales y equipo utilizado

- Cable serie de 8 hilos
- 2 conectores DB9 hembra
- 2 Computadoras

## Desarrollo

**Construcción del cable serial.**

1. **Se realizará un cable RS-232 con un cable de 3 hilos y el conector DB-9, se hará la configuración de modem nulo como se muestra en la figura 1.**

El cable que vamos a fabricar es de tres hilos para conectar de forma simple dos DTEs. Se utilizarán conectores hembra DB-9 y cable.

La descripción de cada pin en el conector DB9 es la siguiente:

| Pin | **Nombre** | **Dirección** | Descripción |
|---|---|---|---|
| 1 | CD | ← | Detección de Portadora |
| 2 | RxD | ← | Recepción de Datos |
| 3 | TxD | → | Transmisión de Datos |
| 4 | DTR | → | Terminal de Datos Preparada |
| 5 | GND | — | Masa del Sistema |
| 6 | DSR | ← | Set de Datos Preparado |
| 7 | RTS | → | Petición para Enviar |
| 8 | CTS | ← | Listo para Enviar |
| 9 | RI | ← | Indicador de Llamada |

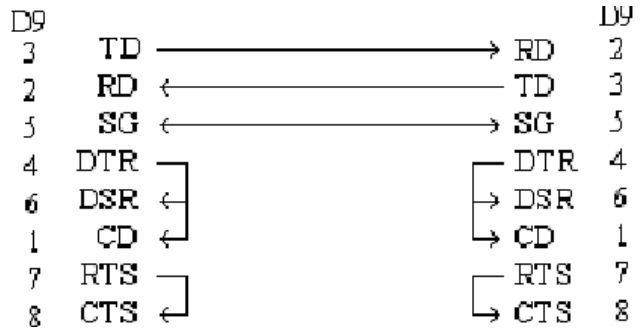



Figura 1. Configuración de modem nulo.

## 2. **Software de comunicación**.

a) Se utilizará el cable realizado en el punto 1 para conectar dos PC's por el puerto serie. A continuación se probará un programa en C que controla los registros del puerto serie RS-232. Este programa funciona como un Chat. Es decir, cuando en un computador se pulse un carácter. El carácter pulsado aparecerá en la pantalla del otro computador. Para ello el programa muestrea continuamente los registros apropiados del puerto y realiza una transmisión full-duplex.

Se utilizará para la explicación de este programa la información sobre el hardware de comunicaciones de datos de la UART 8250 incluido en el tutorial del ***ptoserie.pdf***.

Hay que utilizar un compilador básico de C para teclear, depurar y compilar el programa.

La función principal del programa es:

```
main( )
{
        unsigned char x, y;       / / Declaracion e inicializacion de variables
                                  / / Configuración del Puerto
        x=inport(0x3FB);          / / Leer el contenido del registro de control
        x=x|0x080;                / / Se le aplica una OR con 8 binario, para poner LAB=1
        outport(0x3FB,x);         / /  DLAB=1 sin modificar los demas bits del registro
        outport(0x3F9,1);         / /  se pone el divisor latch  (D=288) para una
        outport(0x3F8,32);        / /  velocidad de 400 bps a frec=1,8432 MHz * /
        outport(0x3FB,2);         / /  Se manda un 2 al registro de control en linea  para
                                  / /  configurar DLAB=0 y 7 bits de longitud, 1 bit de stop * /

        do
        {
                if( kbhit( )!=0)          / / Esta checando si se presiona una tecla
                {
                        y = getch();              / / Se lee la tecla pulsada
                        x = inport (0x3FD);       / / Espera que el buffer de transmisión
                                                  / / este vacío
                        while(!(x &0x30)) {
                             x = inport (0x3FD);    / / Bit 5 de 3FDh=1
                                    }
                           outport (0x3F8,y);     / /Se manda el  carácter
                };


                if((x = inport(0x3FD)) & 0x01)  / / Checa si hay datos en buffer de
                                                        / / recepcion
                {
                        x=inport(0x3F8);          / / Se lee el dato y
                        putch(x);                 / /  lo imprime en pantalla
                };

        } while(1);                               / / Se repite el ciclo de manera
undefinida
        }
```

## 3. Modifique el programa de tal forma que pueda ver usted en su pantalla lo que usted escribe.

## 4. Hacer las siguientes modificaciones al programa anterior

a) Agregar un menú al programa para que el usuario pueda elegir a que velocidad desea hacer la transmisión. Ejemplo:

*Elija una opción para modificar la velocidad de transmisión*

*a) 2,400*
*b) 4,800*
*c) 9,600*
*d) 19,200*
*e) 38,400*

b) Cambiar la configuración para una transmisión de: 8 bits de datos, 2 de stop.

### *Información adicional para desarrollar los incisos a) y b)*

1. Registros de la UART (dirección base=0x3F8 para COM1). En negrita se muestran los registros utilizados en el programa. Todos los registros son de 8 bits.

**Registro del buffer de receptor (0x3F8+0)**
**Registro del buffer de transmisor (0x3F8+0)**
Registro de activación de interrupciones (0x3F8+1)
Registro de identificación de interrupciones (0x3F8+2)
**Registro de formato de datos (0x3F8+3). Control de línea.**
Registro de control de salida RS-232 (0x3F8+4). Control de modem.
**Registro de estado de línea (0x3F8+5)**
Registro de estado de entrada RS-232 (0x3F8+6). Estado de modem.
Registro de "scratch pad" (0x3F8+7)
**Registro de "latch divisor" (0x3F8+0) (0x3F8+1) con DLAB=1**

Por ejemplo:

**Bits del registro de Control de línea (0x3F8+3):**

Bits 0-1. Número de bits a transmitir según la tabla:

| *Bit 1* | *Bit 0* | *Numero de bits de datos* |
|---|---|---|
| *0* | *0* | *5* |
| *0* | *1* | *6* |
| *1* | *0* | *7* |
| *1* | *1* | *8* |

Bit 2. Indica el número de bits de stop (0=> 1 bit, 1=> 2 bits)
Bit 3. Refleja si la paridad esta activada (1) o desactivada (0).
Bit 4. Selector de paridad, par (1) o impar (0).
Bit 5. Habilitación de paridad. (responde a otra tabla que no se muestra aquí )
Bit 6. BREAK.
Bit 7. DLAB. Acceso al "latch divisor".

**Por ejemplo** para 7 bits de datos, 1 bit de stop y sin utilizar paridad se debería cargar en este registro con el binario **00000010 (2 en decimal),** que además pone el DLAB a 0.

**Bits del registro de Estado de línea (0x3F8+5):**

Los que nos interesan (que se utilizan en el programa) son:

Bit 0. (RxRDY) Indica que se ha transferido un byte al buffer de recepción. Este bit se pondrá a '1' cuando se ha leído un carácter de entrada de forma satisfactoria. Pasa a '0' cuando se accede a dicho buffer y se lee.

Bit 5. (TBE) Indica una situación de buffer de transmisión vacío ("transmisor buffer empty") cuando se pone a '1'. Se activa para informar que ya se puede aceptar en parte de la CPU un nuevo carácter para transmisión. Si no se verifica este bit antes de enviar un nuevo carácter al 8250, puede producirse una sobre escritura en la transmisión (escribir sobre un dato que aún no ha sido transmitido), condición que no es detectada por el controlador.

2: Configuración de la velocidad (Baud Rate)

Para seleccionar la velocidad deseada se escribe en el 8250 el **Divisor (o latch divisor)** correspondiente mediante la siguiente secuencia de pasos:

1. Poner el bit DLAB a 1
2. Almacenar en 0x3F8+0 el octeto menos significativo
3. Almacenar en 0x3F8+1 el octeto más significativo
4. Poner el bit DLAB a 0 para proseguir con la operación normal

Donde **Divisor = (Frecuencia reloj referencia)/(16*Velocidad deseada)**

**Por ejemplo**, para una velocidad deseada de 400 bps, y sabiendo que la frecuencia de oscilación del cristal de la 8250 es de 1,8432 MHz, **Divisor** = $1{,}8432 \times 10^{6} / (16*400)$ = **288,** que ha de expresarse mediante dos octetos = 0000 0001 0010 0000. Por lo tanto el byte menos significativo, que hay que introducir en 0x3F8+0 será el valor **32**, y en 0x3F8+1 (MSB) se debería cargar un **1**. la cual es la velocidad a la que se encuentra configurado el programa.

## Evaluación del aprendizaje

a) ¿Cuál es el registro donde se configura la logitud del dato, paridad, bit de stop, ect.?

b) ¿Qué bit es el que se debe checar para saber que hay un dato para ser leído en el buffer?

c) Si se configura la velocidad de una PC a 2400 y la otra a 19200, ¿qué es lo que vemos en pantalla y porque?

d) ¿Cuál es la funcion del bit de paridad en la comunicación serial, de un ejemplo?

## Conclusiones individuales

# Práctica 8
# Construcción de un Grabador para el PIC16F84

## Objetivos

Al completar esta práctica el alumno:

- Será capaz de construir un programador serial para el PIC16F84 de bajo costo.
- Entenderá el concepto de programación in-circuit (ICSP)

## Introducción

El microprocesador que se usara en este curso es el 16F84 por lo cual se elaborará un programador serial de bajo costo, para que cada estudiante cuente con su propio programador, y de esa manera pueda elaborar las prácticas correspondientes.

## Material y equipo utilizado

1 Base de 18 pines
1Transistor BC547B
2 Resistencias 15K y 10 K
1 Capacitor de 100 µF
1 Diodo Zener 5.1 V
Cables para conexión.

## Desarrollo

1. **Mecanismo de programación.** El micro tiene 18 pines como se muestra en la figura 1. El mecanismo de programación se realiza en formato serie a través de cinco líneas: VCC, /MCLR(VPP), tierra, la señal de datos RB7 y la señal de reloj RB6. El micro permite la programación en circuito (ICSP: In Circuit Serial Programming).

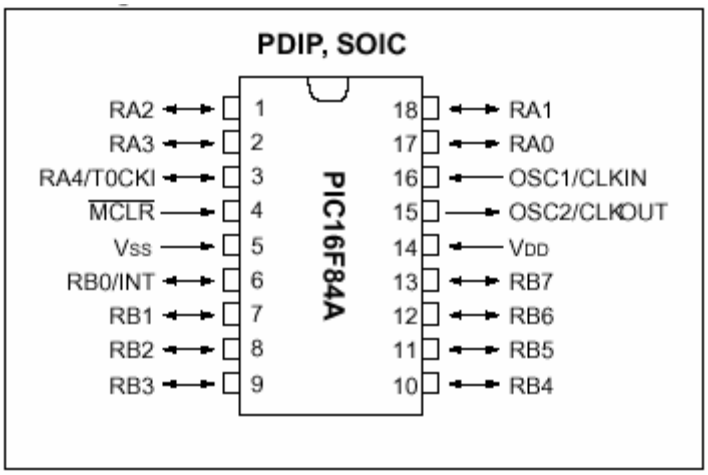



Figura 1. Terminales del Pic 16F84

## 2. Esquema básico

El esquema a realizar es de bajo costo y emplea tan solo unos pocos componentes. Este programador es conocido como JDM y su esquema de conexión se muestra en la figura 2.

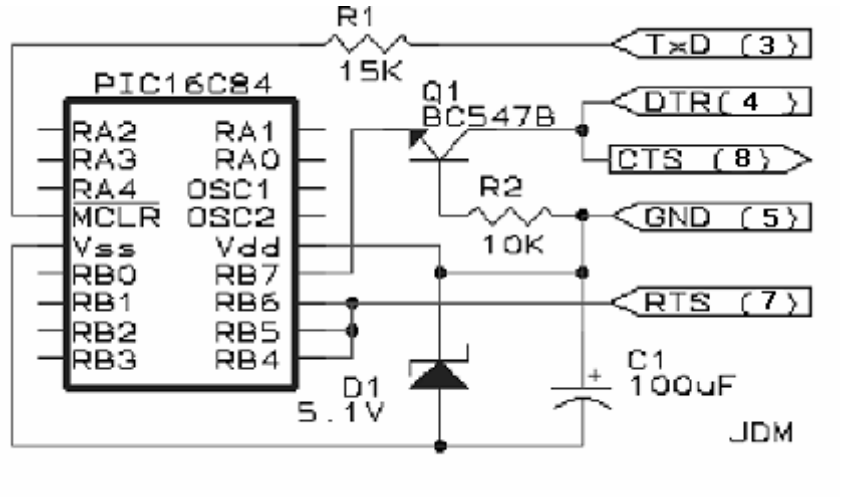



Figura 2.- Esquema del programador de bajo costo JDM

## 3. Arme el circuito y coloque el PIC a programar.

## 4. Verificar el funcionamiento del circuito

a) Ejecute en la computadora el programa ICprog
b) Conecte el cable serial del programador al puerto serial de la computadora.

c) Cargue el programa que lee un dato por el puerto A y lo envía al puerto B.
d) Borre el programa que pueda encontrarse en el PIC
e) Envíe el programa al programador del PIC

**5. Probar que el PIC se programó en forma correcta.**

a) En su protoboard coloque el PIC programado, arme el esquema típico para cualquier aplicación (cristal y circuito de reset).
b) Coloque los 5 bits en la entrada del puerto A (utilizando interruptores).
c) Utilizando LEDs Visualice los datos a la salida del puerto B.

## Conclusiones individuales

# Práctica 9
# Conociendo la herramienta de simulación MPLAB

## Objetivos

Al completar esta práctica el alumno:

- Conocerá la herramienta de edición y simulación MPLAB.
- Será capaz de implementar un programa en ensamblador para el PIC16F84 que realice operaciones aritméticas.
- Será capaz de simular cualquier programa en ensamblador para el PIC16F84 y relacionar cada instrucción del programa con las funciones internas del microcontrolador.

## Introducción

La unidad (ALU) del PIC PIC16F84 es capaz de sumar dos datos de 8 bits cada uno, pero debido a su configuración uno de los sumandos debe proceder del registro de trabajo W.

El problema a resolver en esta práctica consiste en diseñar y probar mediante la herramienta de simulación MPLAB, un programa en ensamblador para el microcontrolador PIC16F84 que realice la operación aritmética de sumar 2 números de 16 bits cada uno, las bases de diseño se muestran en el siguiente mapa de memoria.


## Material y equipo utilizado

Software MPLAB

## Desarrollo

1. **Diagrama de Flujo**. Elabore un diagrama de flujo que cumpla con el problema planteado.

**2. Programa**

a) ¿Cuáles son las instrucciones que necesita?

b) Escriba el programa.

**3. Simular**

c) Capture el programa en el editor.

d) Simule el programa.

## Evaluación de aprendizaje

a) ¿Cómo se muestra en pantalla la ventana del mapa de memoria RAM?

b) ¿Cómo se ejecuta el programa paso por paso?

c) ¿Qué ocurre después de aplicar un reset?

d) Introduzca los siguientes valores al programa y verifique el contenido de las localidades de memoria.

| *Operando 1* | *Operando 2* | *Resultado* |
|---|---|---|
| *001F* | *0023* | |
| *00FF* | *00FF* | |
| *FFFF* | *FFFF* | |
| *0607* | *3012* | |

## Conclusiones individuales

# Práctica 10
# Direccionamiento indirecto de la memoria RAM

## Objetivos

Al completar esta práctica el alumno:

- Conocerá la herramienta de edición y simulación MPLAB.
- Será capaz de implementar un programa en ensamblador para el PIC16F84 que maneje direccionamiento indirecto de la memoria RAM.

## Introducción

El problema a resolver en esta práctica consiste en editar, simular y documentar el siguiente programa que consiste en: almacenar el valor 11h en 15 posiciones contiguas de la memoria de datos mediante direccionamiento indirecto, empezando desde la dirección 0x10

```
            List    p=16F84         ;Tipo de procesador
            include "P16F84.INC"    ;Definición de registros internos

Contador    equ     0x0c            ;Contador interno
Primera     equ     0x10            ;Posición inicial

            org     0x00            ;Vector de Reset
            goto    Inicio

            org     0x05            ;Salva el vector de interrupción

Inicio      movlw   0x0f
            movwf   Contador
            movlw   Primera
            movwf   FSR
            movlw   0x11

Bucle       movwf   INDF
            incf    FSR,F
            decfsz  Contador,F
            goto    Bucle

Stop        nop

            end
```

## Material y equipo utilizado

Software MPLAB

## Desarrollo

1. **Diagrama de Flujo**. Elabore un diagrama de flujo que describa el comportamiento del programa anterior.

2. **Agregue un comentario** a cada una de las líneas del programa, indicando que hace cada instrucción.

**3. Simular**

e) Capture el programa en el editor.

f) Simule el programa.

## Evaluación de aprendizaje

a) Describa con sus palabras que entiende por direccionamiento indirecto

b) Explique con sus palabras como se maneja el direccionamiento indirecto en el PIC 16F84.

c) Mencione los registros que están involucrados en el direccionamiento indirecto.

d) ¿Qué posición de memoria ocupan?

## Conclusiones individuales

# Práctica 11
# Comparador de 2 números de 4 bits

## Objetivos

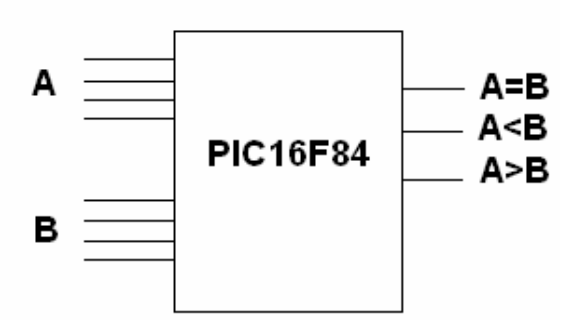
Al completar esta práctica el alumno:

- Utilizará herramienta de edición y simulación MPLAB para verificar el funcionamiento del programa realizado.
- Será capaz de implementar un programa en ensamblador para el PIC16F84 que configure los puertos del microcontrolador y realice operaciones de comparación.
- Será capaz de utilizar al microcontrolador para comunicarse con periféricos comunes como: diodo led e interruptores.

## Introducción

Dentro del repertorio de instrucciones del microcontrolador PIC16F84 no hay aquellas que nos permitan realizar operaciones de comparación. Sin embargo, habrá situaciones en las que será necesario comparar el contenido de 2 o más registros para poder decidir la acción siguiente. Hay que destacar que, al no haber instrucciones de comparación, éstas deben realizarse mediante restas.

El problema a resolver en esta práctica consiste en diseñar el hardware y confeccionar un programa para el PIC16F84 compare dos números (A y B ) de 4 bits c/u representados por 8 interruptores conectados al PUERTOB y genere el resultado en 3 leds conectados al PUERTOA, tomar la siguiente figura como base.




## Material y equipo

1 Microcontrolador PIC16F84
8 Interruptores, 8 Resistenc ias de 1K Ω
3 Diodos LED´s, 3 Resistencias de 330 Ω
Alambres para conexión

## Desarrollo

### 1. Diagrama esquemático.

a) Dibuje un diagrama donde muestre la conexión de los componentes que utilizará para la solución del problema.

### 2. Registros de configuración

a) ¿Cuáles son los registros del microcontrolador que se usarán?
b) ¿Cómo debe configurar cada registro utilizado?

| ***Nombre del Registro*** | ***Dirección del Registro*** | ***Código (binario)*** | ***Código (Hexadecimal)*** |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

### 3. Elaboración del Diagrama de flujo

a) Muestre en un diagrama de flujo la secuencia de pasos necesarios para elaborar el programa.

**4. Programa**

a) ¿Cuáles son las instrucciones que utilizará?

b) Escriba el programa

c) Capture el programa en el editor y compílelo
d) Cargue el programa con extensión .HEX en el ICprog
e) Borre el contenido del PIC y grabe el programa.

**5. Pruebe el prototipo**

a) Arme el circuito y pruébelo.

## Evaluación del aprendizaje

a) Que lógica utilizo para Comparar los números.

b) Introduzca los siguientes valores en los interruptores y verifique las salidas.

| *Número A* | *Número B* | *Salida* |
|---|---|---|
| *0011* | *0111* | |
| *0000* | *0000* | |
| *1111* | *1011* | |
| *1010* | *0101* | |
| *0000* | *1111* | |
| *1100* | *1100* | |

## Conclusiones individuales

# Práctica 12
# Exploración de una entrada mediante un ciclo infinito

## Objetivos

Al completar esta práctica el alumno:

- Utilizará herramienta de edición y simulación MPLAB para verificar el funcionamiento del programa realizado.
- Será capaz de implementar un programa en ensamblador para el PIC16F84 que explore una línea de entrada de manera continua.
- Entenderá el concepto polling.

## Introducción

Dentro del repertorio de instrucciones del microcontrolador, se encuentran las que manipula bits y las que comparan bits. Existen situaciones prácticas en las que será necesario estar monitoreando de manera continua el estado de algún sensor, para lo cual el microcontrolador deberá estar dedicado a realizar esa tarea. Al monitoreo continuo en espera de algún cambio para ejercer una acción se le conoce como "Polling".

El problema planteado para esta práctica consiste en diseñar el hardware y confeccionar un programa para el PIC16F84 que explore el estado de un interruptor conectado a la línea RA0 del puerto A, y lo muestre en un diodo led conectado a la línea RB0 del puerto B.

## Material y equipo

1 Microcontrolador PIC16F84
1 Interruptor
1 Diodo LED´s
1 Resistencia de 330 Ω
1 Resistencia de 1K Ω
Alambres para conexión
Protoboard

## Desarrollo

### 1. Diagrama esquemático.

a) Dibuje un diagrama donde muestre los componentes que utilizará para la solución del problema.

### 2. Elaboración del Diagrama de flujo

a) Muestre en un diagrama de flujo la secuencia de pasos necesarios para elaborar el programa.

**3. Programa**

a) ¿Cuáles son las instrucciones que utilizará?

b) Escriba el programa

**2) Pruebe el prototipo**

a) Arme el circuito y pruébelo.

## Evaluación del aprendizaje

a) Explique como funcionan las instrucciones para comparar bits.

b) Explique con sus palabras el concepto de polling.

c) Mencione una aplicación práctica donde se pueda utilizar el concepto de polling.

## Conclusiones individuales

# Práctica 13
# Contador módulo 15

## Objetivos

Al completar esta práctica el alumno:

- Utilizará herramienta de edición y simulación MPLAB para verificar el funcionamiento del programa realizado.
- Será capaz de utilizar instrucciones en ensamblador para el PIC16F84 que le permitan generar incrementos y decrementos.
- Será capaz de diseñar un contador ascendente o descendente módulo n utilizando un microcontrolador.

## Introducción

Dentro del repertorio de instrucciones del microcontrolador, no existe ninguna que nos permita comparar el contenido de los registros de memoria de una manera directa. Para solucionar este problema lo que se puede hacer es restar al contenido del registro el valor con el que queremos compararlo y analizar el resultado de dicha operación. Si el resultado de la resta es cero los dos valores son iguales; si por el contrario el resultado es distinto de cero los valores comparados son distintos. El bit (Z) en el registro STATUS indica cuando una operación aritmética o lógica realizada en la ALU da como resultado cero

El problema planteado para esta práctica consiste en diseñar el hardware y confeccionar un programa para el PIC 16F84 que comience poniendo a cero un contador, cheque el estado de un interruptor conectado a RA0; si el interruptor vale cero la cuenta será ascendente y si vale uno será descendente. El valor del contador debe ser mostrado en 4 leds conectados al puerto B.

## Material y equipo

1 Microcontrolador PIC16F84
1 Interruptor
4 Diodo LED´s
4 Resistencia de 330 Ω
1 Resistencia de 1K Ω
Alambres para conexión

## Desarrollo

### 1. Diagrama esquemático.

a) Dibuje un diagrama donde muestre los componentes que utilizará para la solución del problema.

### 2. Elaboración del Diagrama de flujo

a) Muestre en un diagrama de flujo la secuencia de pasos necesarios para elaborar el programa.

**3. Programa**

a) Escriba el programa

**3) Pruebe el prototipo**

a) Arme el circuito y pruébelo.

## Evaluación del aprendizaje

a) Explique cómo logró visualizar cada incremento de la cuenta en los LED´s.

b) Explique en que consiste el algoritmo para comparar con el límite de la cuenta en los casos ascendente y descendente.

## Conclusiones individuales

# Práctica 14
# Corrimiento de leds

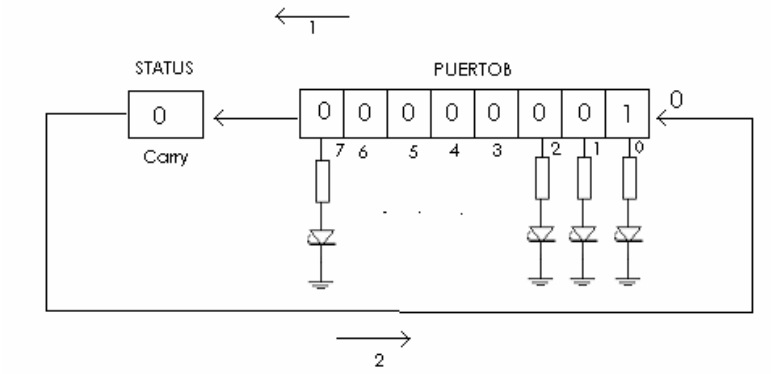
## Objetivos

Al completar esta práctica el alumno:

- Utilizará herramienta de edición y simulación MPLAB para verificar el funcionamiento del programa realizado.
- Será capaz de utilizar instrucciones en ensamblador para el PIC16F84 que le permitan generar corrimientos de izquierda a derecha y viceversa.

## Introducción

El problema planteado en esta práctica consiste en diseñar el hardware y confeccionar un programa para el PIC 16F84 que desplace el encendido de un led conectado al puerto B de izquierda a derecha y viceversa. La figura siguiente muestra lo que ocurre cuando se rota un bit a la izquierda. El bit más significativo del Puerto B se pasa al bit de Carry del registro de STATUS y luego se introduce al bit menos significativo de puerto B. Cada corrimiento debe ser ejecutado con espacio de aproximadamente 1 segundo.




## Material y equipo

1 Microcontrolador PIC16F84
7 Diodos LED´s
7 Resistencias de 330 Ω
Alambres para conexión

## Desarrollo

### 1. Diagrama esquemático.

a) Dibuje un diagrama donde muestre los componentes que utilizará para la solución del problema.

### 2. Elaboración del Diagrama de flujo

a) Muestre en un diagrama de flujo la secuencia de pasos necesarios para elaborar el programa.

**3. Programa**

b) Escriba el programa

**4. Pruebe el prototipo**

a) Arme el circuito y pruébelo.

## Evaluación del aprendizaje

a) Que instrucciones utilizo para hacer el corrimiento de izquierda a derecha. Explique como funcionan.

b) Que instrucciones utilizo para hacer el corrimiento de derecha a izquierda. Explique como funcionan.

c) Como desarrollo el algoritmo para generar el retardo de aproximadamente 1 segundo.

## Conclusiones individuales

# Práctica 15
# Generador de señales cuadradas

## Objetivos

Al completar esta práctica el alumno:

- Utilizará herramienta de edición y simulación MPLAB para verificar el funcionamiento del programa realizado.
- Será capaz de utilizar el TMR0 como temporizador.
- Será capaz de genera pulsos para diferentes periodos.

## Introducción

Una de las funciones más habituales en los programas de control suele ser determinar intervalos concretos de tiempo. También suele ser frecuente contar impulsos producidos en el exterior del sistema. En el microcontrolador PIC16F84 estas funciones la realiza un temporizador/contador de 8 bits, llamado TMR0 que actúa de dos maneras distintas:

- ***Contador de sucesos***, representados por los impulsos que se aplican al pin RA4/T0CKI. Al llegar al valor FFh se desbordará el contador y, con el siguiente impulso pasa a 00h, advirtiendo esta circunstancia activando un señalizador y/o provocando una interrupción.

- ***Temporizador*****,** se incrementa con cada ciclo de instrucción (4 / Freloj), o divisores del mismo, hasta que se desborda (pasa de 00h a FFh) y avisa poniendo a '1' un señalizador y/o provocando una interrupción.

Para que el TMR0 funcione como contador de impulsos aplicados en RA4/T0CKI hay que poner a '1' el bit T0CS, que es el que ocupa la posición 5 del registro OPTION. Para que el TMR0 funcione como temporizador el bit T0CS debe ponerse a '0'. TMR0 es un registro de propósito especial ubicado en la posición 01h del banco 0 de la memoria de datos RAM. En igual dirección pero en el banco 1 se encuentra el registro de configuración OPTION.

La temporización se calcula a partir del periodo de la señal de reloj (Treloj), el valor de un divisor de frecuencia definido en el registro OPTION y el valor del temporizador TMR0.

$$\textbf{Temporización} = 4 \cdot \textbf{Treloj} \cdot (255 - \textbf{TMR0}) \cdot \textbf{Divisor}$$
$$255 - \textbf{TMR0} = \textbf{Temporización} / (4 \cdot \textbf{Treloj} \cdot \textbf{Divisor})$$

El problema planteado para esta práctica consiste en diseñar el hardware y confeccionar un programa para el PIC 16F84 trabajando a 4MHz que produzca una señal cuadrada de periodo 100 ms (75 ms a nivel alto y 25 ms a nivel bajo), la señal generada debe ser mostrada por el pin RB0. Utilice la bandera que indica que el temporizador llego a su cuenta máxima. Para comprobar el periodo de la señal se debe utilizar el osciloscopio.

## Material y equipo

1 Microcontrolador PIC16F84
Alambres para conexión
Protoboard
Osciloscopio

## Desarrollo

**1. Diagrama esquemático.**

a) Dibuje un diagrama donde muestre los componentes que utilizará para la solución del problema.

**2. Elaboración del Diagrama de flujo**

a) Muestre en un diagrama de flujo la secuencia de pasos necesarios para elaborar el programa.

## 3. Programa

a) Escriba el programa

**4. Pruebe el prototipo**

a) Arme el circuito y pruébelo.

## Evaluación del aprendizaje

a) Escriba los cálculos desarrollados para obtener el valor del temporizador para 75ms y 25 ms.

b) Cuál es el error obtenido al comparar lo calculado con lo medido en el osciloscopio.

c) Qué considera usted que ocasionó el error.

## Conclusiones individuales

# Práctica 16
# Temporización de un segundo y activación de una carga

## Objetivos

Al completar esta práctica el alumno:

- Utilizará herramienta de edición y simulación MPLAB para verificar el funcionamiento del programa realizado.
- Será capaz de utilizar el TMR0 como temporizador.
- Será capaz de genera pulsos para periodos de un segundo o mayores.
- Será capaz de controlar cargas de 120 VAC

## Introducción

El problema planteado para esta práctica consiste en diseñar el hardware y confeccionar un programa para el PIC 16F84 trabajando a 4MHz que produzca una señal cuadrada de periodo 2 s (1 s a nivel alto y 1 s a nivel bajo), la señal generada debe ser mostrada por el pin RA4. Utilice la interrupción del TMR0 que indica que el temporizador llego a su cuenta máxima. Conecte el interruptor de para el control de la carga en PA0 y la carga en PB5. Utilice la práctica 4 como interfaz de potencia. Para comprobar el periodo de la señal se debe utilizar el osciloscopio.

**Nota.** La activación de la carga y la generación del pulso deben ser simultáneas.

## Material y equipo

1 Microcontrolador PIC16F84
Alambres para conexión
Protoboard
Osciloscopio
Componentes de acuerdo a su diseño.

## Desarrollo

**1. Diagrama esquemático.**

a) Dibuje un diagrama donde muestre los componentes que utilizará para la solución del problema.

## 2. Elaboración del Diagrama de flujo

a) Muestre en un diagrama de flujo la secuencia de pasos necesarios para elaborar el programa.

**3. Programa**

a) Escriba el programa

**4. Pruebe el prototipo**

a) Arme el circuito y pruébelo.

## Evaluación del aprendizaje

a) Con que valor cargo el timer (TMR0) para generar la señal de 1 segundo.

b) Qué problema tuvo para visualizar la señal de salida y como lo solucionó.

c) ¿Qué es una interrupción?

d) Describa el funcionamiento de su rutina de atención de servicio de interrupción del TMR0.

## Conclusiones individuales

# Práctica 17
# Decodificador de un display de 7 segmentos

## Objetivos

Al completar esta práctica el alumno:

- Utilizará herramienta de edición y simulación MPLAB para verificar el funcionamiento del programa realizado.
- Será capaz de generar tablas y acceder a ellas mediante instrucciones de ensamblador para el microcontrolador PIC16F84.
- Será capaz de utilizar visualizadores de 7 segmentos mediante un microcontrolador, sin decodificador externo.

## Introducción

Diseñar el hardware y Confeccionar un programa para el PIC 16F84 trabajando a 4MHz para controlar un display de 7 segmentos con 4 interruptores, es decir, el numero binario introducido por medio de tres interruptores visualizarlo en el display de 7 segmentos.

## Material y equipo

1 Microcontrolador PIC16F84
Alambres para conexión
Protoboard
Lo que usted considere en su diseño.

## Desarrollo

**1. Diagrama esquemático.**

a) Dibuje un diagrama donde muestre los componentes que utilizará para la solución del problema.

## 2. Tabla de codificación

a) Elabore una tabla de codificación para el display de 7 segmentos según lo haya escogido (ánodo común o cátodo común). Especifique el display que va utilizar.

| *Número* | *Código 7 segmentos* | *Código hexadecimal* |
|---|---|---|
| *0* | | |
| *1* | | |
| *2* | | |
| *3* | | |
| *4* | | |
| *5* | | |
| *6* | | |
| *7* | | |
| *8* | | |
| *9* | | |
| *A* | | |
| *b* | | |
| *C* | | |
| *d* | | |
| *E* | | |
| *F* | | |

## 3. Elaboración del Diagrama de flujo

a) Muestre en un diagrama de flujo la secuencia de pasos necesarios para elaborar el programa.

**4. Programa**

a) Escriba el programa

**5. Pruebe el prototipo**

a) Arme el circuito y pruébelo.

## Evaluación del aprendizaje

a) ¿Cómo se hace una consulta a una tabla?

b) Explique la instrucción para regresar de una tabla.

c) Mencione dos aplicaciones que le podría dar a una tabla.

## Conclusiones individuales

# Práctica 18
# El TMR0 como contador de eventos externos

## Objetivos

Al completar esta práctica el alumno:

- Utilizará herramienta de edición y simulación MPLAB para verificar el funcionamiento del programa realizado.
- Será capaz de utilizar el TMR0 del microcontrolador PIC16F84 como contador.
- Será capaz de utilizar visualizadores de 7 segmentos mediante un microcontrolador, sin decodificador externo.

## Introducción

El problema planteado para esta práctica consiste en diseñar el hardware y confeccionar un programa para el PIC 16F84 que haga lo siguiente:

Un sensor optoelectrónico (simulado por un interruptor) conectado a RA4 genera un pulso cada vez que un objeto se interpone entre el emisor y el receptor de luz. El TMR0 se encarga de contar cada pulso. El valor de la cuenta será desplegado en dos dígitos de 7 segmentos conectados al PUERTOB. Considere el siguiente segmento de código.

```
Loop    clrf    TMR0        ; Inicializa  contador
        btfsc   PORTA,3     ; Checa si RA4 está activo
        goto    Loop        ; Si, Cuenta detenida

        movf    TMR0,W      ;No, Lee el valor del contador
        Call    despliega   ;Despliega la cuenta en los dígitos
        goto    Loop        ; regresa
        end
```

**Nota:** Debe tenerse en cuenta el "efecto rebote mecánico " que se produce en el interruptor RA4.

## Material y equipo

1 Microcontrolador PIC16F84
Alambres para conexión
Protoboard
Lo que usted considere en su diseño.

## Desarrollo

**1. Diagrama esquemático.**

a) Dibuje un diagrama donde muestre los componentes que utilizará para la solución del problema.

**2. Diagrama de flujo.**

a) Elabore el diagrama de flujo para la solución del problema.

3. **Programa**
   a) Escriba el programa

**4. Pruebe el prototipo**
   a) Arme el circuito y pruébelo.

## Evaluación del aprendizaje

a) Mencione los registros involucrados para configurar el timer TMR0 como contador.

b) ¿Con qué valores los configuró?.

c) ¿Cómo elimino el rebote mecánico del interruptor?

## Conclusiones individuales

# Práctica 19
# La memoria EEPROM de datos

## Objetivos

Al completar esta práctica el alumno:

- Utilizará herramienta de edición y simulación MPLAB para verificar el funcionamiento del programa realizado.
- Será capaz de leer y escribir la memoria EEPROM del microcontrolador PIC16F84.
- Será capaz de eliminar por software el efecto del "rebote mecánico" generado por interruptores.

## Introducción

El problema a resolver en esta práctica consiste en editar, simular y documentar el siguiente programa; así como diseñar el hardware correspondiente para su funcionamiento.

Se trata de emular el funcionamiento de las máquinas tipo "SU TURNO" habituales en múltiples comercios. Sobre un display de 7 segmentos se visualizará el número del turno actual. Este se incrementa a cada pulso aplicado sobre un interruptor por RA0. En la memoria EEPROM del PIC16F84 se almacena el último número visualizado, de forma que, ante un fallo de alimentación, se reanude la cuenta en el último número.

```
;Si se parte de que el sistema se emplea por vez primera , se visualiza el 0

                List    p=16F84
                include "P16F84.INC"

Contador        equ     0x0c

                org     0x00
                goto    Inicio
                org     0x05

;*****************************************************************************
;EE_Write: Graba un byte en la EEPROM de datos. La dirección será la contenida en EEADR y el dato se
;supone está en EEDATA

EE_Write        bsf     STATUS,RP0
                bsf     EECON1,WREN
                movlw   b'01010101'
                movwf   EECON2
```

```
                movlw   b'10101010'
                movwf   EECON2              ;
                bsf     EECON1,WR
                bcf     EECON1,WREN
Wait            btfss   EECON1,EEIF
                goto    Wait
                bcf     EECON1,EEIF
                bcf     STATUS,RP0
                return

;*****************************************************************************************
;EE_Read: Leer un byte de la EEPROM. Se supone al registro EEADR cargado con la dirección a leer. En
:EEDATA aparecerá el dato leído.

EE_Read         bsf     STATUS,RP0
                bsf     EECON1,RD
                bcf     STATUS,RP0
                return

;*****************************************************************************************
;Tabla: Esta rutina convierte el código BCD presente en los 4 bits de menos peso del reg. W en su
;equivalente a 7 segmentos. El código 7 segmentos retorna también en el reg. W

Tabla:          addwf   PCL,F               ;Desplazamiento sobre la tabla
                retlw   b'00111111'         ;Dígito 0
                retlw   b'00000110'         ;Dígito 1
                retlw   b'01011011'         ;Dígito 2
                retlw   b'01001111'         ;Dígito 3
                retlw   b'01100110'         ;Dígito 4
                retlw   b'01101101'         ;Dígito 5
                retlw   b'01111101'         ;Dígito 6
                retlw   b'00000111'         ;Dígito 7
                retlw   b'01111111'         ;Dígito 8
                retlw   b'01100111'         ;Dígito 9

;*****************************************************************************************
;Delay_20_ms: Esta rutina de temporización tiene por objeto eliminar el "efecto rebote" de los periféricos
;electromecánicos. Realiza un delay de 20 mS. Si el PIC trabaja a una frecuencia de 4MHz, el TMR0
;evoluciona cada µS. Si queremos temporizar 20000 µS (20 mS) con un preescaler de 128, el TMR0 deberá
;contar 156 eventos (156 * 128). El valor 156 equivale a 9c hex. y como el TMR0 es ascendente habrá que
;cargar su complemento a 2 (63 hex.).

Delay_20_ms:    bcf     INTCON,T0IF
                movlw   0x63
                movwf   TMR0
Delay_20_ms_1   clrwdt
                btfss   INTCON,T0IF
                goto    Delay_20_ms_1
                bcf     INTCON,T0IF
                return

; *****************************************************************************************
; Rutina principal

Inicio          clrf    PORTB
                bsf     STATUS,RP0
```

```
                clrf    PORTB
                movlw   b'00011111'
                movwf   PORTA
                movlw   b'00000110'
                movwf   OPTION_REG
                bcf     STATUS,RP0


                clrf    EEADR
                call    EE_Read
                movlw   0x09
                subwf   EEDATA,W
                btfsc   STATUS,C
                goto    Ini_0
                goto    Ini_1
Ini_0           clrf    Contador
                goto    Loop
Ini_1           movf    EEDATA,W
                movwf   Contador

Loop            movf    Contador,W
                call    Tabla
                movwf   PORTB

Wait_0          clrwdt
                btfss   PORTA,0
                goto    Wait_0
                call    Delay_20_ms

Wait_1          clrwdt
                btfsc   PORTA,0
                goto    Wait_1
                call    Delay_20_ms

                incf    Contador,F
                movlw   .10
                subwf   Contador,W
                btfsc   STATUS,Z
                clrf    Contador
                movf    Contador,W
                movwf   EEDATA
                call    EE_Write
                goto    Loop

                end
```

## Material y equipo

1 Microcontrolador PIC16F84
Alambres para conexión
Protoboard
Lo que usted considere en su diseño.

## Desarrollo

**1. Diagrama esquemático.**

a) Dibuje un diagrama donde muestre los componentes que utilizará para la solución del problema.

**2. Simulación.**

a) Edite y simule el programa anterior.

**3. Diagrama de flujo.**

a) Elabore el diagrama de flujo en base a la simulación anterior.

4. **Documentación del programa**

a) Escriba un comentario en cada una de las líneas que indique que hace la instrucción.

**5. Pruebe el prototipo**

a) Programe el microcontrolador con el código anterior.
b) Arme el circuito y pruébelo.

## Evaluación del aprendizaje

a) Escriba los pasos para leer un dato de la memoria EEPROM.

b) Escriba los pasos para escribir un dato en la memoria EEPROM.

c) Relacione las rutinas del programa con los pasos que expresó en los incisos a y b.

## Conclusiones individuales

# Práctica 20
# Modo "sleep" y "wake-up" mediante el watchdog.

## Objetivos

Al completar esta práctica el alumno:

- Utilizará herramienta de edición y simulación MPLAB para verificar el funcionamiento del programa realizado.
- Comprenderá el funcionamiento de la instrucción sleep (bajo consumo) del microcontrolador PIC16F84.
- Comprenderá como se divide la frecuencia en el timer WDT (perro guardian).

## Introducción

El problema a resolver en esta práctica consiste en editar, simular y documentar el siguiente programa; así como diseñar el hardware correspondiente para su funcionamiento.

Se trata de mostrar el empleo de la instrucción SLEEP para poner al PIC16F84 en el modo de bajo consumo. El despertar se producirá cada vez que el WDT llegue a su cuenta máxima. En ese momento se producirá un incremento del valor del PUERTOB que actuará como contador binario reflejado en 8 leds, y nuevamente se volverá a la situación de bajo consumo. El preescaler se asociará al WDT y estará comprendido entre 1 y 128, dependiendo del estado lógico de los interruptores RA0-RA2. El valor nominal del WDT es de 18mS. Es decir, con un preescaler de 1, el PIC "despertará" cada 18mS, con un preescaler de 128, lo hará cada 2.3 segundos.

```
                List    p=16F84
                include "P16F84.INC"

                org     0x00
                goto    Inicio
                org     0x05

Inicio          clrf    PORTB
                bsf     STATUS,RP0
                clrf    PORTB
                movlw   b'00011111'
                movwf   PORTA
                movlw   b'00001000'
                movwf   OPTION_REG
                bcf     STATUS,RP0
Loop            sleep
```

```
incf    PORTB,F
movf    PORTA,W
andlw   b'00000111'
iorlw   b'00001000'
bsf     STATUS,RP0
movwf   OPTION_REG
bcf     STATUS,RP0
goto    Loop

end
```

## Material y equipo

1 Microcontrolador PIC16F84
Alambres para conexión
Protoboard
Lo que usted considere en su diseño.

## Desarrollo

**1. Diagrama esquemático.**

a) Dibuje un diagrama donde muestre los componentes que utilizará para la solución del problema.

**2. Diagrama de flujo.**

a) Elabore el diagrama de flujo en base al código anterior.

3. **Documentación del programa**

a) Escriba un comentario en cada una de las líneas que indique que hace la instrucción.

**4. Pruebe el prototipo**

a) Programe el microcontrolador con el código anterior.
b) Arme el circuito y pruébelo.

## Evaluación del aprendizaje

a) Qué hace el microcontrolador cuando entra al estado de bajo consumo de potencia.

b) ¿Cómo se incrementa la cuenta del temporizador WDT?

c) ¿Cuáles son las formas para sacar al microcontrolador del estado de bajo consumo de potencia?

## Conclusiones individuales