

PIC MICRO ESTUDIO

Display de 2 y4 Dígitos RS-232 (Clave 701 y 702)

www.electronicaestudio.com

Guía de Uso

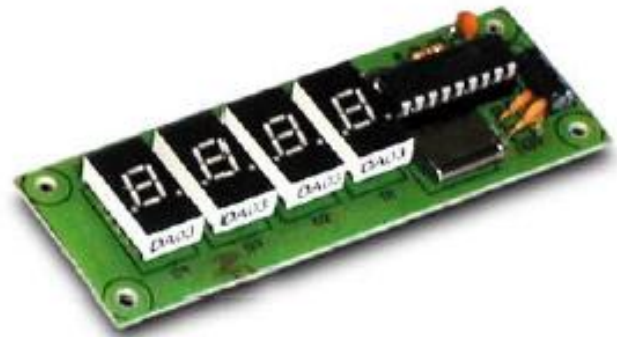
Ing. Wilfrido González Bonilla

Display de 2 y4 Dígitos RS-232 con el PIC16F84

Son innumerables las aplicaciones de los contadores, los podemos encontrar en máquinas de herramientas para contar las piezas que se producen, o para indicar el tiempo que se invierte en la fabricación de un producto. Se usan en los aparatos de medición como voltímetros y amperímetros y en todo tipo de relojes o de timers.

También existen múltiples maneras de implementar un contador. Se pueden diseñar con compuertas TTL con los CI 74LS247, 248, 249 que son decodificadores / drivers BCD a 7 segmentos. Es por esta razón que

[Picmicro Estudio](#) ofrece este módulo con el nombre de **Disp4 (Clave 702)**. El cual consta de las siguientes características:



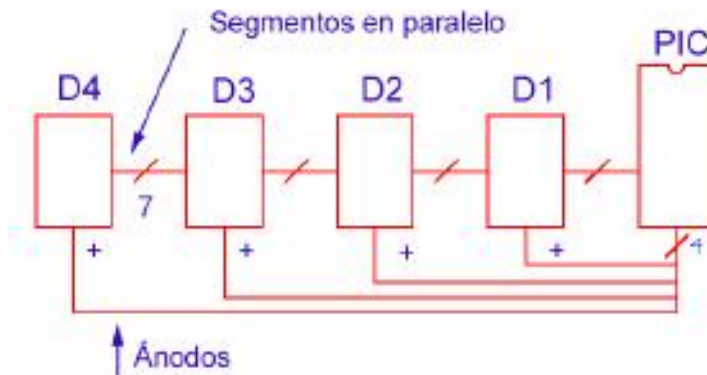
- **4 Displays de 7 segmentos**
- **1 Microcontrolador PIC16F84**
- **1 cristal de 4 Mhz**
- **Capacidad de comunicarse por el puerto serial de una computadora sin necesidad del CI MAX-232**

Además este modulo puede funcionar como:

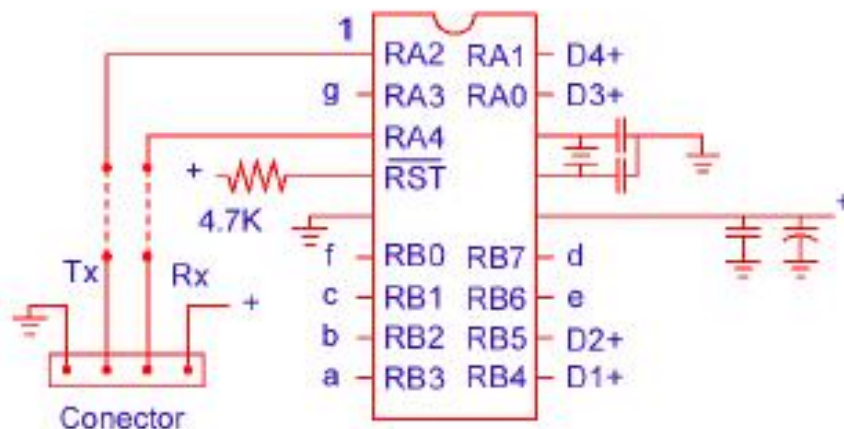
- **Contador de Pulsos**
- **Tacometro**

Descripción del Circuito

Los 7 segmentos de cada uno de los 4 Displays están conectados en paralelo. Las 7 líneas llegan a los bornes del PIC. El común de cada dígito D1, D2, D3, D4 también se lleva al PIC.



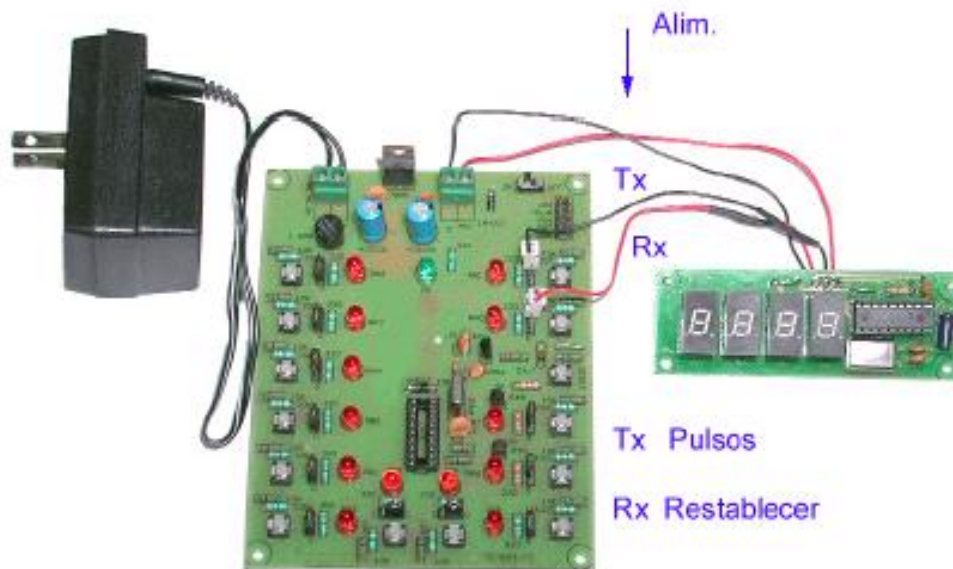
Mediante un conector se proporciona alimentación a la tarjeta y se tiene acceso a los pines del PIC, Tx (RA2) y Rx (RA4). Con esta estructura son muchas las aplicaciones que se le pueden dar a este módulo. Tx y Rx pueden conectarse a una PC o a cualquier otro microcontrolador para enviar y recibir datos en forma serial RS-232, la siguiente figura describe lo anteriormente dicho.



En este caso usaremos el módulo para realizar un contador de pulsos, con botón de restablecer. **La terminal Tx** estará destinada a contar los pulsos de entrada (frentes negativos), y **la terminal Rx** estará destinada a conectarse a un botón para restablecer el contador.

Con la tarjeta “**Entrenadora del PIC16F84**” (Clave 502), que ya hemos utilizado en ejercicios anteriores, podemos alimentar y simular mediante botones las señales **Tx** y **Rx** que requiere el contador.

La tarjeta entrenadora cuenta con dos bornes que proporcionan 5 volts de DC, justo lo que necesitamos en este caso. Esta conexión no tiene problemas. Los botones RA1 y RA2 los podemos utilizar para simular las entradas del contador.

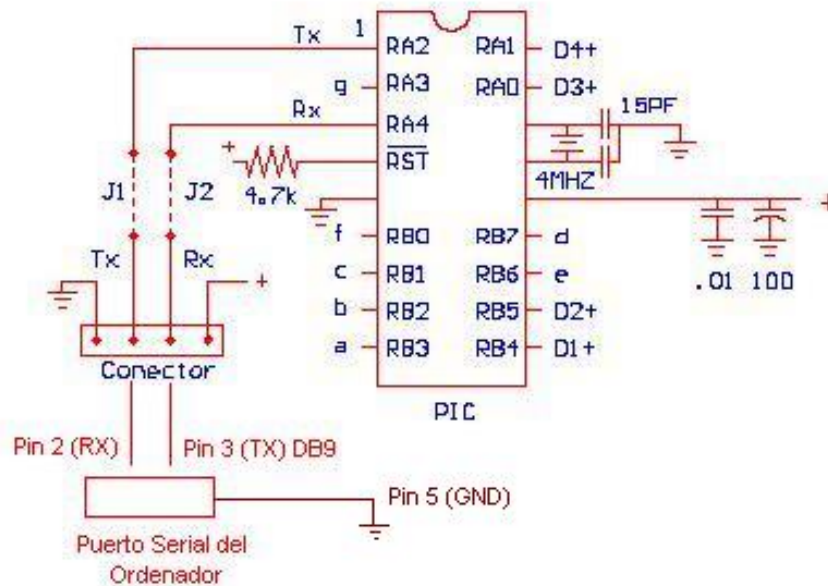


En esta figura se presenta una fotografía del proyecto armado en nuestro laboratorio. Para su comodidad el software, que en esta ocasión es bastante más grande lo puede encontrar en: www.prodigyweb.net.mx/wgb/articulos bajo el nombre de **contador.zip**

Visual Basic para el Display de 4 Dígitos RS-232

Siguiendo con el artículo anterior, en esta ocasión proponemos 2 programas en VBasic para mostrar el funcionamiento de este modulo.

Cabe mencionar que el microcontrolador debe estar programado con el programa **disp4-7**, el cual puede encontrar en el disco adjunto a este producto. Por otra parte el hardware para estas practicas es el siguiente.



Para conectar directamente al ordenador

- **J1** sustituir por un resistencia de **1k**
- **J2** sustituir por un resistencia de **22k**

Del puerto serial de la computadora se puede utilizar un cable serial 1 a 1, con conectores **DB9** macho y hembra en cada lado, el cual se puede conseguir en cualquier tienda donde se venda equipo para computadoras.

El Nombre del programa para este primer ejemplo es : enviar.vbp

En la siguiente figura se muestra este programa



Básicamente, como se puede observar el operador dispone de 3 botones, 1 de ellos envía el numero deseado al microcontrolador, este numero es enviado a través del puerto serial con los siguientes parámetros

- **Bits por segundo: 2400 bps**
- **Numero de bits: 8**
- **Paridad: ninguna**
- **Bits de paro: 1**

Cuando el usuario aprieta el botón con el rotulo **Enviar Dato**, este se deberá visualizar en el display de 4 dígitos.

A continuación se muestra el código que hace posible esta aplicación

DESCRIPCIÓN DEL SOFTWARE



Para poder acceder al puerto serial y así poder enviar datos utilizando una aplicación creada en Visual Basic, se hace uso del control **MS COMM**, el cual trae incorporadas todas las funciones para configurar el puerto.

Las propiedades más importantes de este control son las siguientes:

- **ComPort:** Activa y regresa el número del puerto serial (Comm1, 2)
- **PortOpen:** Activa y regresa el acceso al puerto.
- **Input:** Regresa los caracteres del buffer receptor.
- **Output:** Escribe una cadena sobre el buffer Transmisor.
- **Settings:** Activa y regresa la razón de Baudios, paridad, número de bits, bits de paro.

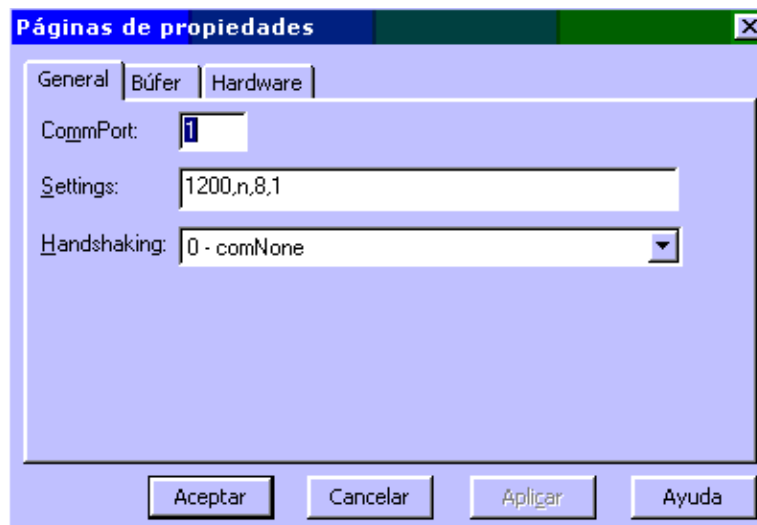
Para poder tener acceso a cualquier propiedad del puerto serial se utiliza la siguiente sintaxis:

Nombre del Control . Propiedad = Valor

En este caso el objeto es MS Comm1, por lo tanto si quisiera abrir el puerto, la instrucción sería:

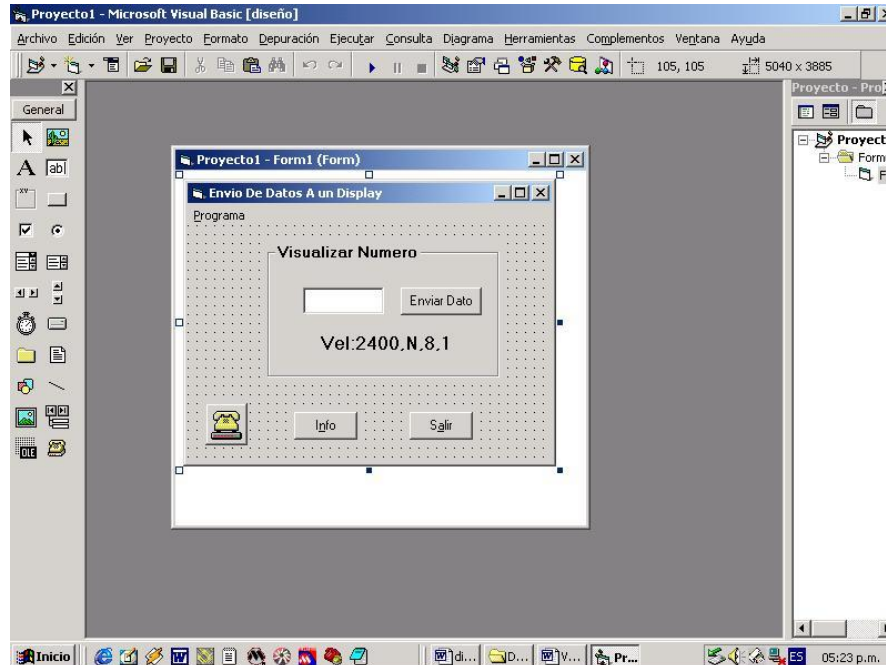
MS Comm1.PortOpen = True

Sin embargo, para poder utilizar el puerto serial, primeramente, se debe colocar el control MS Comm1 en la forma y hacer clic con el botón derecho del mouse, para que puedan aparecer sus propiedades, tal y como lo muestra la siguiente figura.



Enviar.exe

En la siguiente Figura se muestra la Forma y los diferentes componentes del programa en Visual Basic 6. dicho programa tiene el nombre de **enviar.vbp**, el cual puede encontrar en el disco que acompaña este módulo



Una vez entendido esto, se procede a declarar las variables a utilizar, en este caso se declara la variable **comando** de tipo cadena, en la cual se almacenara los números tecleados por el usuario en el objeto Textbox1.

Option Explicit
Dim comando As String

Se debe recordar que en cualquier programa de **Vbasic**, el código que primero se ejecuta, es aquel que esta contenido en la forma. Por lo tanto se ejecuta las siguientes instrucciones:

Private Sub Form_Load()
serie.InputLen = 1024
serie.PortOpen = True


```

comando = "DN"
serie.Output = comando + "0000"
End Sub

```

Como se observa el nombre del control **MSCOMM** se a cambiado por **serie**, el cual hace referencia al puerto serial.

Como se menciono anteriormente, al iniciar un aplicación en Vbasic en donde se utilice el puerto serial, se debe primeramente, determinar el ancho del buffer, en este caso 1024. Acto seguido se debe abrir el puerto, esto se hace con la instrucción **serie.portopen=true**. Una vez hecho esto se procede a enviar a través del puerto la cadena **"DN0000"**, donde **"DN"** esta almacenado en la variable **Comando**.

Cuando el usuario apriete el botón con el rotulo **"Enviar dato"**, se ejecuta las siguiente instrucción:

```

Private Sub Command1_Click()
serie.Output = comando + Text1.Text
End Sub

```

Al hacer un click sobre este botón, el texto que este en el control **Text1**, es agregado a la constante comando, y se envía a través del puerto.

Sin embargo cuando el usuario se dirige al control **Text1**, este puede escribir cualquier carácter y presionar sobre el botón **Enviar dato**, **produciendo un error en el envió de datos ya que el microcontrolador no entendería este comando**. Para evitar esto se creo una función en dicho control en el cual, el usuario solo podrá escribir caracteres numéricos, evitando así este problema. Esto se logra con el siguiente código:

```

Private Sub Text1_KeyPress(KeyAscii As Integer)
Dim car As String * 1, tem
car = Chr(KeyAscii)
If car < "0" Or car > "9" Then
KeyAscii = 0
Beep
Else
tem = Asc(car) - &H30
End If
End Sub

```

Se declaran dos variables: **Car de tipo cadena** y **tem de tipo variant**. Cuando el usuario desea insertar un carácter en el control **Text**, este es almacenado en la variable **KeyAscii**, inmediatamente la función **Chr()** convierte un carácter en numero, y este es almacenado en la variable **Car**.

Después se hace una comparación en donde se pregunta si la variable **Car** esta dentro del rango de los números “0 y 9”, y cuando se teclaa cualquier letra que no sea un numero, esta condición se cumple, regresando el la variable **KeyAscii** un valor de 0 seguido un pitido de error.

Por otra parte cuando se teclaa un numero, este cae en la segunda condición y se procede a convertirlo a través de la función **Asc()**, la cual convierte un numero en **ASCII**.

Por ultimo cuando el usuario desea terminar la ejecución del programa, este tiene dos opciones, una es a través del botón con el rotulo **Salir**, y al otra con el botón de cerrar, en la esquina superior derecha. Independientemente se cual sea la opción escogida, las dos opciones cierran el puerto serial antes de terminar la aplicación.

```
Private Sub Form_Unload(Cancel As Integer)
    serie.Output = comando + "----"
    serie.PortOpen = False
End
End Sub
```

```
Private Sub mnsalir_Click()
    serie.Output = comando + "----"
    serie.PortOpen = False
End
End Sub
```

Programa contador. vbp

Este programa tiene como objetivo la elaboración de un contador , en donde el usuario a través de la computadora envía el dato a desplegar en los displays. Cada vez que se apriete el botón con el rotulo “**iniciar cuenta**”, el numero desplegado en la pantalla del monitor se incrementara en una unidad, y este deberá ser visualizado en la tarjeta.



Para la elaboración de este programa se declaran tres variables:

Comando del tipo **String**

xx del tipo **Variant**

mat(9999) la cual es una matriz del tipo **String**,

la cual cuenta con **10000 elementos**

Option Explicit

Dim comando As String

Dim mat(9999) As String

Dim xx

Recordando que en cualquier programa de **Vbasic**, el código que primero se ejecuta, es aquel que esta contenido en la forma, se procede a explicar las siguientes líneas.

Private Sub Form_Load()

Dim ss

serie.InputLen = 1024

serie.PortOpen = True

serie.Output = "DN----"

comando = "DN"

For ss = 0 To 9

```

    mat(ss) = "DN000" + CStr(ss)
Next ss

For ss = 10 To 99
    mat(ss) = "DN00" + CStr(ss)
Next ss

For ss = 100 To 999
    mat(ss) = "DN0" + CStr(ss)
Next ss

For ss = 1000 To 9999
    mat(ss) = "DN" + CStr(ss)
Next ss

End Sub

```

Como se observa, se declara una variable con el nombre **ss**, la cual servirá como índice para dirigirse a cualquier elemento de la matriz:

Cada elemento de la matriz contendrá la cadena “**DN**”, existiendo 4 ciclos **FOR** para rellenar cada uno de los **10000 elementos** de la matriz, cada FOR, representa las unidades, decenas, centenas y unidades de millar, respectivamente.

La función **Cstr()**, convierte un numero en cadena, esto se hace por que a través del puerto serial solo se pueden enviar cadenas de caracteres.

De tal forma que por ejemplo:

El elemento mat(998), contendrá la cadena “DN0998”

Por otra parte cada vez que el usuario se dirige al botón con el rotulo “**iniciar cuenta**”, el puerto serial envía el contenido de la variable **mat**, y la variable **xx**, actúa como índice incrementándose en uno cada vez que se aprieta dicho botón.

Cuando la variable **xx** llega a 1000, se cumple cierta condición, la cual pone a 0 dicha variable. A continuación se muestra lo antes mencionado.

```
Private Sub Command2_Click()  
    serie.Output = mat(xx)  
    Label1.Caption = xx  
    If xx = 1000 Then  
        xx = 0  
        serie.Output = comando + "0000"  
    End If  
    xx = xx + 1  
End Sub
```

Por ultimo cuando se desea terminar la ejecución del programa, simplemente el usuario deberá apretar el botón con rotulo **“Resetear”**. Lo siguiente se muestra a continuación.

```
Private Sub Command1_Click()  
    serie.Output = comando + "0000"  
    Label1.Caption = "0"  
    xx = 0  
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)  
    serie.Output = "DN----"  
    serie.PortOpen = False  
    End  
End Sub
```