

Diseño de
Sistemas Informáticos Industriales

PRÁCTICAS

Ingeniería Técnica Industrial (Electrónico)
Departamento de Informática de Sistemas y Computadores
Escuela Técnica Superior de Ingeniería del Diseño

6

PROGRAMACIÓN DEL PUERTO SERIE EN C++ BUILDER

1. OBJETIVO

Aprender a usar la comunicación serie en el PC desde aplicaciones desarrolladas en C++ Builder y utilizando un componente.

2. MATERIAL NECESARIO

- 2 ordenadores PC Compatible
- Cable NULL-MODEM
- Borland C++ Builder
- Componente para comunicaciones serie
- Emulador de terminal para puerto serie
- Simulador de visor de báscula con salida serie

3. INTRODUCCIÓN

En esta práctica se hará uso del entorno C++ Builder y de un componente para comunicaciones serie que permite sacar gran partido al puerto de comunicaciones serie del PC.

Para usar el puerto serie desde Windows se puede recurrir a la programación directa del API del sistema operativo o usar un componente prefabricado de alguna compañía. Emplear un componente prefabricado permite obtener la máxima funcionalidad sin un gran esfuerzo de programación. En nuestro caso hemos recurrido al componente ComPort (freeware).

La práctica se desarrollará interconectando dos PCs a través del puerto serie COM? mediante un cable NULL-MODEM. En uno de los "lados" se empleará un programa que genere y lea información por el puerto serie (usando un emulador de terminal o un simulador) y en el otro se deberá desarrollar el software capaz de leer y escribir información de/en el puerto serie.

4. DESARROLLO

4.1 INSTALACIÓN DEL COMPONENTE

Instala el componente a partir de las instrucciones contenidas en el fichero de distribución proporcionado con la práctica.

4.2 PRIMER PROGRAMA

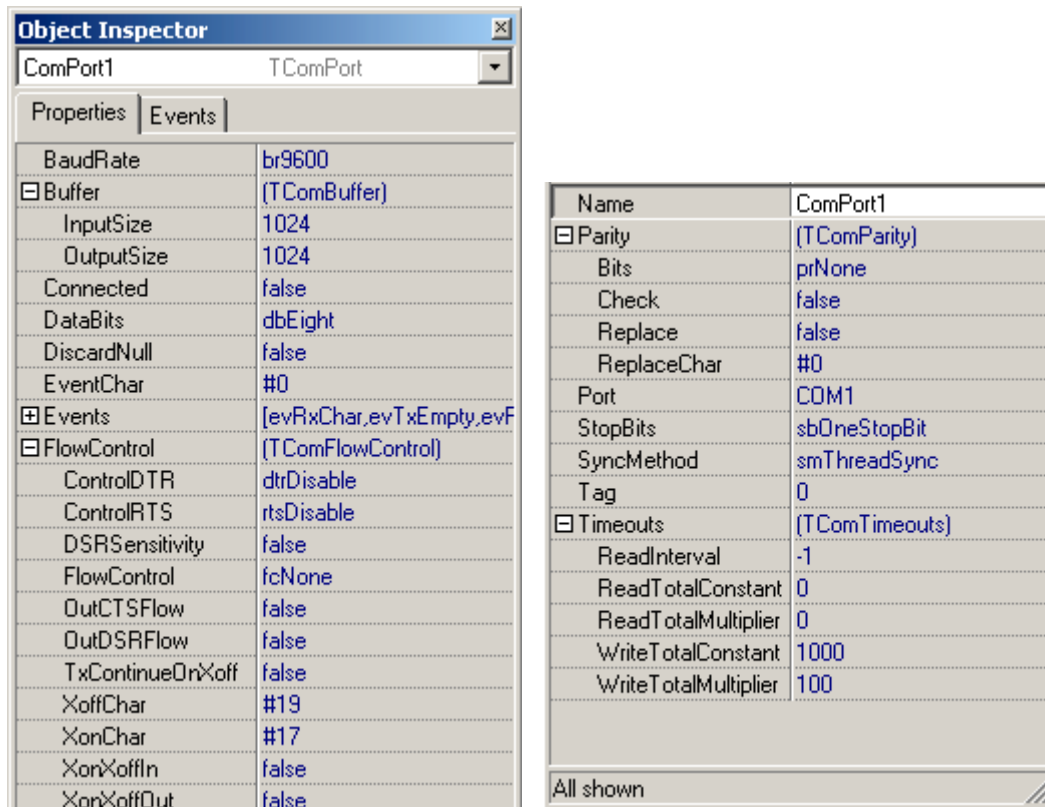
Abre Builder y guarda el proyecto en una nueva carpeta antes de empezar a trabajar. Prueba que el proyecto se ejecuta.

Añade el primer componente al formulario principal y ejecuta de nuevo la aplicación. Si todo se ha hecho bien en el apartado anterior, no debería haber ningún problema.



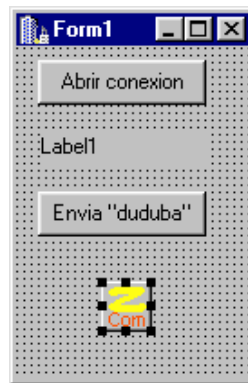
Para usar el componente se debe establecer primero la configuración del puerto. Después se "abre la conexión" y ya se puede enviar y recibir información. Cuando llegan caracteres por el serie se produce un evento que facilita la recogida asíncrona de información.

Una forma de configurar el componente es introducir los parámetros del puerto en el Inspector de Objetos. Mira el siguiente ejemplo, pero no lo hagas así.



La configuración del puerto se hará en el botón de "Abrir conexión" que se comenta a continuación.

Añade un par de botones y una etiqueta. Un botón se va a emplear para abrir la comunicación y otro para enviar una cadena al serie cada vez que se pulse en él. La etiqueta se va a emplear para mostrar lo que se recibe por el serie. La aplicación puede tener el siguiente aspecto:



Asocia el siguiente código:

Al botón de abrir conexión,

```
ComPort1->Port = "COM1"; //me falla en Builder 6
ComPort1->BaudRate = br9600;
ComPort1->Parity->Bits = prNone;
ComPort1->DataBits = dbEight;
ComPort1->StopBits = sbOneStopBit;
//ComPort1->ShowSetupDialog(); // probar con esto
ComPort1->Open();
if (ComPort1->Connected == false) {
    ShowMessage("Problemas abriendo el puerto serie");
    exit(-1);
}
```

Al botón de enviar algo,

```
ComPort1->WriteStr("duduba");
```

Al evento OnRxChar del componente,

```
AnsiString a;
ComPort1->ReadStr(a,Count);
Label1->Caption = a;
```

Al evento OnError del componente,

```
ShowMessage("Error del serie; ");
```

Prueba el programa empleando en el otro ordenador el emulador de terminal y/o el simulador del visor de peso. Fíjate en que la "label" no tiene por que representar un solo carácter, sino que puede aparecer más de una letra.

4.3 ACTIVIDADES

Amplía la aplicación anterior añadiendo mayor funcionalidad de manera que se interprete la información que se recibe por ele serie y se emplee para algo. Por ejemplo:

- Control remoto de la posición en pantalla de la aplicación que recibe caracteres.
- Que la aplicación envíe cada cierto tiempo las coordenadas donde está en pantalla.
- etc.