

	Cliente SQL en Windows CE 6.0 / Digi ConnectCore Wi9C	Nota de Aplicación
		CoAN-004
	Lector de códigos con GUI para comercio.	Publicado: 00/00/0000
		Página 1 de 1

Revisión	Fecha	Comentario	Autor
0	19/10/2007		Ulises Bigliati

1. Objetivos:

Presentar una aproximación al desarrollo de software embebido para Windows CE 6.0 sobre la base del .NET compact framework y el IDE de Visual Studio 2005.

Generar una aplicación embebida en el módulo Digi mediante Visual Basic .NET y metodología orientada a objetos y obtener desde allí acceso a un servidor de base de datos.

Que fuera del alcance de este trabajo toda descripción o instrucción relativas a la configuración y puesta en marcha del servidor de base de datos y a la creación de la base de datos provista.

Se asume la existencia de una instancia de SQL server 2005 express con la base de datos provista adjuntada y conectado a la red de trabajo del módulo Digi.

Se asume que la instalación de las herramientas de desarrollo del Jump Start Kit CCWi9C para WinCE6.0 de Digi ha sido realizada satisfactoriamente.

2. Introducción

El presente trabajo pretende construir una aplicación comercial concreta utilizando las herramientas presentes en el kit de desarrollo Jump Start Kit de la firma Digi, para uno de sus módulos basado en ARM 9: el Connect Core Wi9C.

Entre estas herramientas se encuentran:

- El Visual Studio 2005 Professional edition (versión de evaluación de 180 días)
- Windows Embedded CE 6.0 + Platform Builder (versión de evaluación de 180 días)
- El BSP (board support package) desarrollado por Digi para Windows Embedded CE 6.0.

El SQL server 2005 express es gratuito y se instala junto con el Visual Studio 2005.

Recomendamos la descarga e instalación del software de administración del servidor de base de datos "SQL server Management Studio express" que puede obtenerse en forma gratuita desde el sitio web de Microsoft.

3. Desarrollo del proyecto

3.1. Requisitos:

El caso práctico:

Nos hemos planteado la necesidad hipotética de tener que generar una solución orientada a un comercio -mas concretamente una librería- que nos permita leer los códigos de artículo, mostrando con posterioridad en una pantalla, su precio, descripción, una imagen del producto y algunos otros datos específicos del negocio. El listado de artículos se encuentra almacenado en una base de datos.

3.2. Análisis:

Comenzamos a analizar el caso, desde una perspectiva general para ir avanzando hacia lo más específico, es decir, utilizamos un enfoque top-down:

Realmente esta aplicación es muy sencilla y encaja en un esquema de entrada – proceso – salida donde tendremos:

 CONTINEA <small>Microprocesamiento modular + Conectividad</small>	Cliente SQL en Windows CE 6.0 / Digi ConnectCore Wi9C	Nota de Aplicación CoAN-004
	Lector de códigos con GUI para comercio.	Publicado: 00/00/0000 Página 2 de 2

Entrada -> Captura del código de identificación de los artículos.

Es evidente que vamos a necesitar un medio de identificación y un lector de códigos. Vamos a seleccionar para esta función un conjunto de tags de RF y su correspondiente lector de RFID. El motivo de esta elección no es ni as ni menos que la comodidad de tenerlo disponible en el laboratorio.

Proceso -> Ejecución de una consulta a una base de datos y manipulación de los resultados.

Es este punto notamos la necesidad de incorporar a nuestra aplicación un cliente de base de datos. El resto consistirá en trabajar sobre los datos obtenidos para preparar la salida.

Por supuesto que en este punto está implícita la condición de contar en nuestro sistema con conectividad de red.

Salida-> Despliegue de imágenes, y otros datos en una pantalla gráfica.

El componente que resta añadir a nuestro sistema en esta etapa del análisis es un display gráfico, con esto queda completo el esquema para esta etapa de análisis.

3.3. Diseño:

3.3.1. Hardware

3.3.1.1. Entrada

Nuestro lector presenta una interfaz RS232 con salida de datos ASCII y nos entrega directamente un string de caracteres correspondiente al ID del código recibido mas un caracter de fin de línea.

De esta manera, vamos a conectarnos a nuestro procesador mediante dicho protocolo, por lo tanto necesitaremos un puerto serie disponible en nuestro sistema host.

3.3.1.2. Procesamiento

Tal como lo adelantamos, la plataforma de procesamiento consistirá en un módulo Connectcore Wi9C, ya que en este dispositivo residen los elementos necesarios para satisfacer nuestros requisitos y mucho mas, a saber:

- **Controlador para display** color de 640x480.
- Interfaz **ethernet** cableada y/o inalámbrica
- Cuatro puertos seriales (en la placa de prototipos dos son TTL y dos son RS232)
- Sistema operativo Windows CE embebido, con lo cual disponemos del modelo de acceso a datos del .NET compact framework.
- Adicionalmente disponemos de interfaz **USB host** en el módulo, con lo cual, de ser necesario podremos correr nuestro software desde allí o incluso instalar componentes de software adicionales como veremos mas adelante.

3.3.1.3. Salida

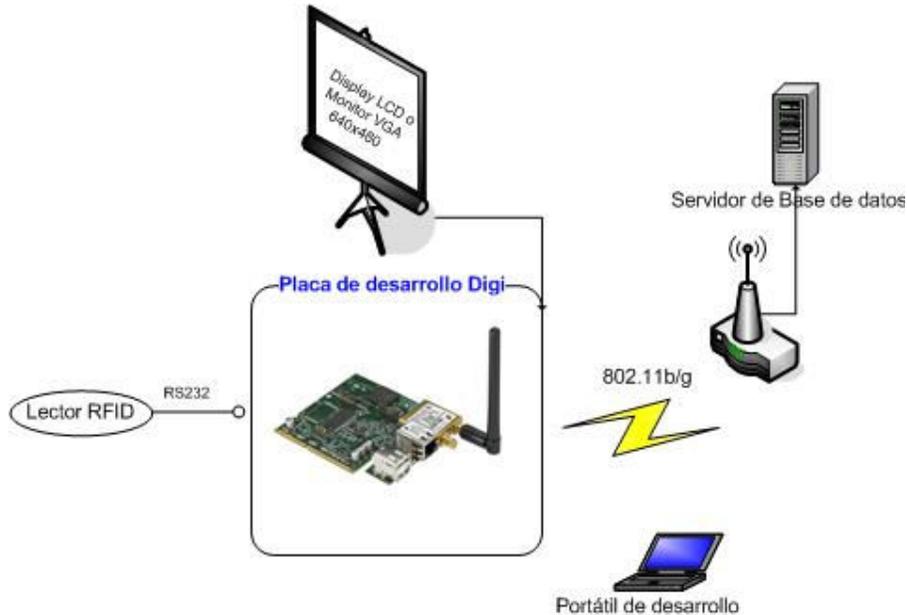
Para el bloque de salida, utilizaremos un display gráfico color de 640x480, pero además, durante el tiempo de desarrollo contamos con salida VGA en la placa de prototipos, lo cual nos permite ensayar el desarrollo directamente en un monitor estándar, sin la necesidad de disponer del display gráfico.

Por otra parte, tenemos la posibilidad de desarrollar en Visual Studio .NET, con lo cual tendremos un rápido desarrollo de la interfaz gráfica como así también del acceso a datos.

	Cliente SQL en Windows CE 6.0 / Digi ConnectCore Wi9C	Nota de Aplicación
		CoAN-004
	Lector de códigos con GUI para comercio.	Publicado: 00/00/0000
		Página 3 de 3

3.3.1.4. Integración

Hasta acá tenemos reunidos todos los elementos de hardware necesarios para generar la aplicación. A continuación nos ocuparemos del diseño del sistema desde el punto de vista del software.



3.3.2. Software

Al utilizar este entorno de desarrollo, contamos con la posibilidad de utilizar la programación visual y orientada a objetos.

Tenemos la comodidad de estar desarrollando sobre un sistema operativo embebido en el módulo, y disponemos del entorno de desarrollo Visual studio .NET con lo cual tenemos acceso al modelo de objetos y componentes presentes en el .NET compact framework como así también podremos acceder a las API de Windows CE.

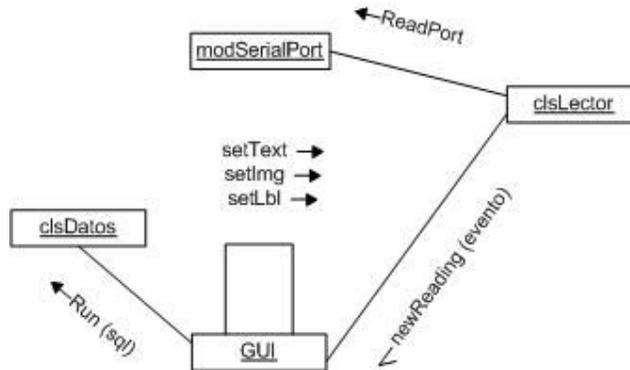
Particularmente nos interesan para esta aplicación las librerías de acceso a datos del .NET compact framework para conectarnos al servidor SQL y además las API's que nos proporcionan el control de los puertos serie para recibir los datos del lector de RFID.

En nuestra aplicación vamos a necesitar definir clases a partir de las cuales se instanciarán los objetos que estarán representando en software a nuestros componentes concretos del hardware de nuestra aplicación. Así surgen en correspondencia con el análisis realizado mas arriba, las siguientes clases que luego implementaremos en el entorno de desarrollo:

- Debemos acceder a un servidor de base de datos, entonces necesitamos una clase de acceso a datos. Podría llamarse clsDatos.
- Es necesaria una clase para manipular al lector y obtener los ID's que este nos entregue. Podría llamarse clsLector.
- También es indispensable acceder y controlar un puerto serie, de tal forma que crearemos un módulo con todas las funciones necesarias. Podría llamarse modSerialPort.
- Por último debemos desplegar la información que manejamos, existe una interfaz de usuario que se muestra en un display o un monitor, y esto lo materializamos mediante un formulario de windows estándar. Podría llamarse frmADOTest.

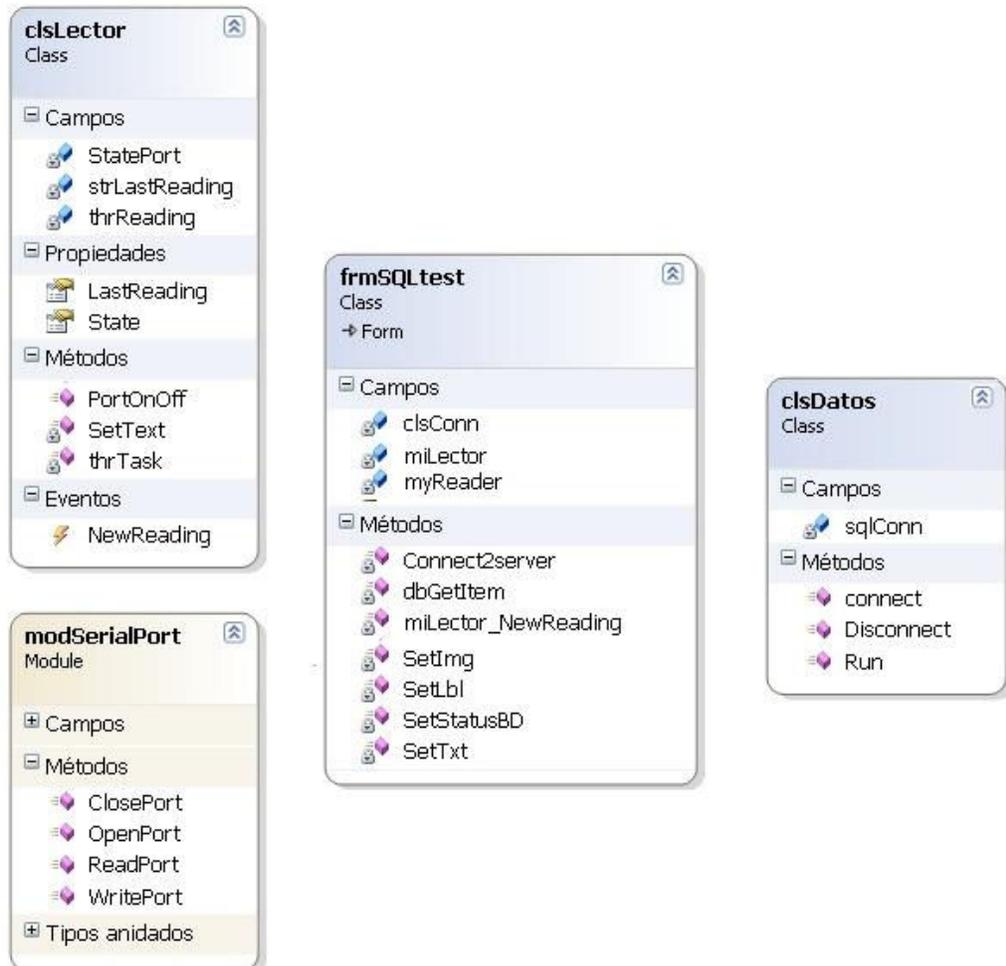
3.3.2.1. Diagrama de objetos

Un sencillo diagrama que representa el funcionamiento básico de nuestro programa es el siguiente:



3.3.2.2. Diagrama de clases

El diagrama de clases que ya comienza a delinear mayores detalles puede apreciarse a continuación. Este diagrama se traducirá luego en código fuente, cada una de las clases definidas se materializará en un módulo de software independiente.



 CONTINEA Microprocesamiento modular + Conectividad	Cliente SQL en Windows CE 6.0 / Digi ConnectCore Wi9C	Nota de Aplicación
		CoAN-004
	Lector de códigos con GUI para comercio.	Publicado: 00/00/0000
		Página 5 de 5

3.3.2.3 Especificación de clases

Los métodos, propiedades y eventos que pueden verse en el diagrama constituirán las funciones que implementará cada clase, a través de las cuales se logrará el comportamiento necesario para satisfacer los requerimientos de la aplicación.

A continuación se describen brevemente las funcionalidades implementadas por cada clase, no se ahondará en los detalles del código fuente, ya que por un lado en esta instancia del diseño no es necesario ahondar en detalles de implementación, por otra parte, el código fuente se encuentra disponible junto al presente trabajo y en gran parte se auto-explica.

modSerialPort	
Tipo	módulo estándar de visual basic.
Función	proveer acceso y control del puerto serie.
Metodos	
ClosePort	Abre el puerto serie especificado
OpenPort	Cierra el puerto serie especificado
ReadPort	Lee la cantidad de bytes especificada y los devuelve en un buffer.
WritePort	Escribe la cantidad de bytes especificada desde un buffer

clsLector	
Tipo	clase de visual basic.
Función	obtener y entregar los ID's de los artículos en forma asíncrona.
Metodos	
PortOnOff:	establece o termina la comunicación serial con el lector RFID.
Propiedades	
LastReading	Mantiene disponible la última lectura obtenida.
Eventos	
NewReading	evento disparado para alertar al obejo contenedor de la existencia de un nuevo ID para procesar.
Objetos privados de la clase	
thrReading	se trata de un thread que lee permanentemente el puerto serie hasta completar una lectura. El motivo de utilizar un thread es que la función del lectura del puerto es bloqueante y de esta manera se establece un esquema cooperativo entre tareas, manejado en forma transparente al programador.

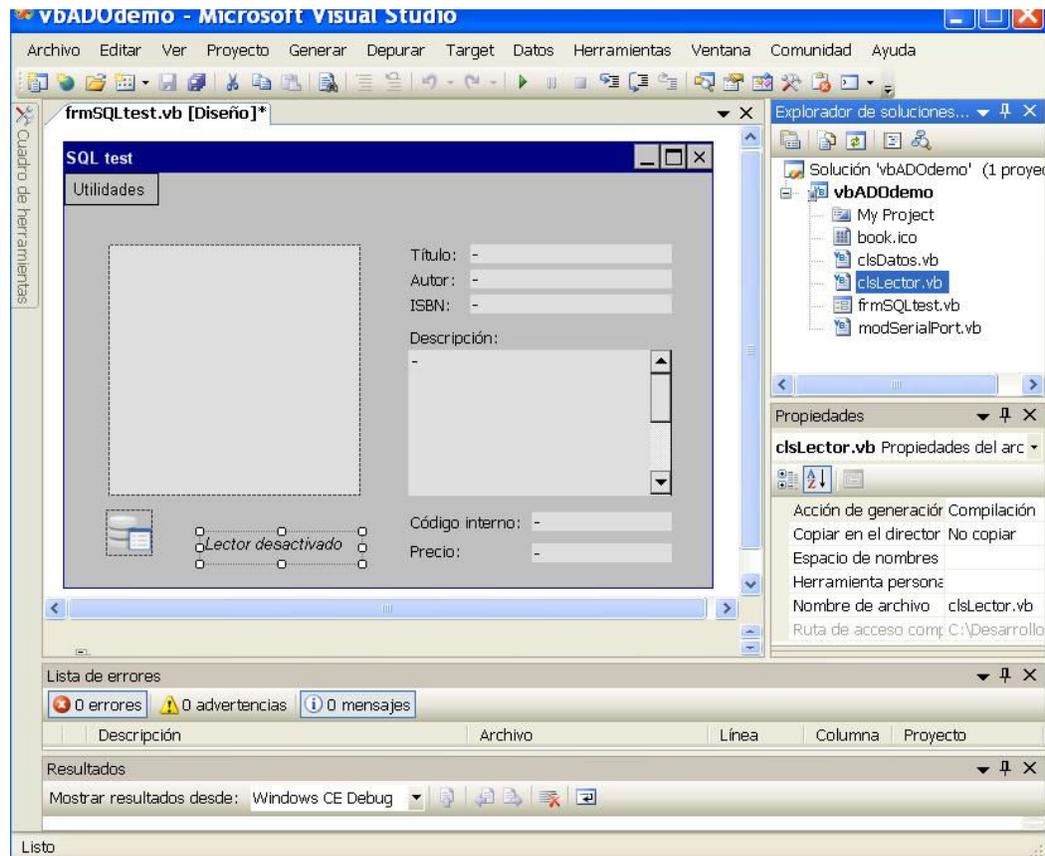
clsDatos	
Tipo	clase de visual basic.
Función	proveer acceso al servidor de base de datos SQL server. La comunicación se establece vía TCP/IP en forma transparente al programador gracias al modelo de objetos de acceso a datos del .NET compact framework.
Metodos	
Connect	Establece una conexión con el servidor de base de datos.
Disconnect	Cierra la conexión.
Run	Ejecuta una consulta especificada a la base de datos y devuelve un set de registros para ser procesados.
Objetos privados de la clase	
sqlConn	Objeto que proveer la interfaz con servidores y archivos de bases de datos.

	Cliente SQL en Windows CE 6.0 / Digi ConnectCore Wi9C	Nota de Aplicación
		CoAN-004
	Lector de códigos con GUI para comercio.	Publicado: 00/00/0000
		Página 6 de 6

frmSQLtest	
Tipo	formulario estándar de visual basic.
Función	es la GUI y al mismo tiempo oficia de contenedor del resto de los objetos.
Metodos	
Connect2Server	establece la conexión con el servidor de base de datos utilizando clsDatos.
dbGetItem	ejecuta una consulta a la base de datos mediante clsDatos.
miLector_NewReading	punto de recepción del evento generado por clsLector, similar a una función callback.
setImg, setLbl, setTxt	muestran los datos recibidos según su tipo en el formulario.
Objetos privados de la clase	
miLector	la instancia de la clase clsLector.
clsConn	la instancia de la clase clsDatos
myReader	objeto manipulador de registros y campos resultado de las consultas a la BD. Pertenece al modelo de objetos de acceso a datos del .NET CF.

3.3.2.4. Interfaz de usuario

Para finalizar esta etapa del diseño podremos ver a continuación una imagen preliminar de la interfaz de usuario que a estas alturas ya es necesario delinear. Por otra parte puede verse a la derecha en el explorador de soluciones, la lista de los módulos de software correspondientes a los objetos que acabamos de describir.



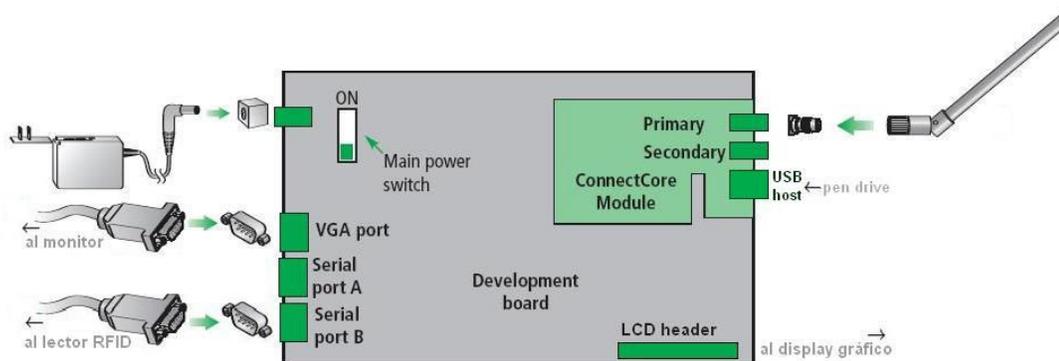
	Cliente SQL en Windows CE 6.0 / Digi ConnectCore Wi9C	Nota de Aplicación CoAN-004
	Lector de códigos con GUI para comercio.	Publicado: 00/00/0000
		Página 7 de 7

3.4. Implementación

Finalmente estamos en condiciones de iniciar la última etapa, en la cual realizaremos la implementación integral de la aplicación. Dividiremos la implementación en dos bloques, por un lado el hardware y por otro el software.

3.4.1. Hardware:

Todo el desarrollo de la aplicación se ha ensayado sobre la **placa de prototipos** que está incluida en los kits de desarrollo de Digi International. En este caso, esta placa de prototipos dispone, como puede apreciarse en el esquema, de todas las interfaces de hardware para interconectar las diferentes partes que componen el proyecto.



3.4.1.1. Interfaz gráfica

Durante nuestros ensayos utilizamos un display gráfico de Prime View International, modelo PD064VT4 que puede adquirirse en Cika Electrónica. Este display es del tipo TFT y maneja una resolución de hasta 640x480 y 262.144 colores.

Por supuesto, si no disponemos inmediatamente de un display gráfico, o no tenemos pensado utilizarlo podemos emplear directamente la **salida VGA** de la placa de desarrollo y conectarnos a un monitor estándar.

Nota: excede el alcance de este trabajo la explicación de cómo interfacear el display color con el header presente en la placa de desarrollo. Para dicha tarea será necesario recurrir simplemente a las hojas de datos del development board de Digi y a las especificaciones del display.

3.4.1.2. Lector de códigos

Tal como ya hemos mencionado nuestro lector de códigos es un lector de proximidad de tags de identificación de RF. Presenta una interfaz serial con niveles RS232 y por lo tanto solo es necesario alimentarlo con 12 volts y conectar, además de GND, la salida TX al pin RX del conector DB9 de la placa de desarrollo, utilizaremos el port B del módulo Digi para conectarnos con el lector. El lector de RF también puede adquirirse en Cika Electrónica, su código es el EM9917AA.

3.4.1.3. Red

El módulo ConnectCore Wi9C integra una interfaz ethernet 802.11b/g, por lo tanto nos conectaremos a un access point en modo infraestructura. De esta forma tendremos acceso a nuestro servidor de base de datos que puede hallarse en cualquier parte de nuestra LAN.

	Cliente SQL en Windows CE 6.0 / Digi ConnectCore Wi9C	Nota de Aplicación
		CoAN-004
	Lector de códigos con GUI para comercio.	Publicado: 00/00/0000
		Página 8 de 8

3.4.1.4. USB host

Adicionalmente contamos con dos conectores USB host, donde por un lado podremos conectar un mouse y por otro nos será útil para utilizar a modo de unidad de disco. Ya que, hasta tanto no incorporemos nuestra aplicación a la imagen del sistema y grabemos con ella la flash del módulo debemos almacenar la aplicación en memoria RAM, o tal como lo proponemos en una memoria externa como es el caso de un pen-drive.

3.4.2. Software

Ya hemos mencionado que como entorno de desarrollo para trabajar con módulos Digi ConnectCore Wi9C/9C que ejecutan Windos CE 6.0 debemos utilizar el Visual Studio .NET. Por una cuestión de ganar rapidez en el desarrollo del software elejimos utilizar el lenguaje interpretado Visual Basic, pero a la hora de trabajar en este entorno el lector podrá seleccionar el lenguaje de su agrado, es decir, que podrá optar por Visual Basic, C# o C++.

En esta sección asumimos que todo el sistema se encuentra instalado, conectado y funcionando, ya que toda la documentación al respecto se encuentra bien detallada acompañando a cada Jump Star Kit de Digi International.

3.4.2.1. La base de datos

Entre el material que acompaña a esta nota se pueden hallar los archivos correspondientes a la base de datos para SQL2005 que se ha utilizado junto con el programa.

Solo se deben ubicar los archivos en el directorio de datos del servidor, y luego adjuntar la base de datos utilizando la herramienta de administración del servidor.

Desde el programa se asume que el servicio de base de datos está escuchando en el puerto 1433 y se accede directamente mediante TCP/IP (no se utilizan canalizaciones con nombre).

Otro punto importante es que el servidor debe admitir autenticación mixta, es decir que debe permitir el login de usuarios definidos internamente, según la seguridad propia de SQL server y no solo a los usuarios de Windows.

3.4.2.2. Componentes de software para Windows CE.

Para que sea posible utilizar el cliente de acceso a datos desde nuestro módulo Digi, debemos instalar un componente de software adicional, que podemos encontrar dentro de nuestra instalación del Visual Studio .NET y el Platform Builder.

Se trata del Cabinet File *sql.wce5.armv4i.CAB* que normalmente puede encontrarse en la siguiente ubicación de nuestra instalación del Visual Studio y que de todas formas se provee junto con esta nota.
C:\Archivos de programa\Microsoft Visual Studio 8\SmartDevices\SDK

SQL Server\Client\v2.0\wce500\armv4i

Para instalar este componente en Windows CE, simplemente se puede copiar el archivo .CAB dentro de un pen-drive y luego transferirlo al módulo, instalandolo de la forma habitual, doble click mediante.

	Cliente SQL en Windows CE 6.0 / Digi ConnectCore Wi9C	Nota de Aplicación
		CoAN-004
	Lector de códigos con GUI para comercio.	Publicado: 00/00/0000
		Página 9 de 9

3.4.2.3 El proyecto en .NET

En este punto estamos en condiciones de abrir el proyecto software que acompaña a este trabajo. El archivo que mantiene la información del proyecto es el *vbADODemo.sln*, al abrirlo se puede acceder mediante el IDE de Visual Studio .NET a todo el código fuente antes descrito desde el punto de vista de diseño.

Junto al archivo *vbADODemo.sln* se encuentra la carpeta del proyecto que lleva el mismo nombre, en su interior están todos los archivos fuente de este trabajo.

Partiendo de la ubicación de esta carpeta, en la ruta *\vbADODemo\bin\Release* encontramos el archivo ejecutable de nuestra aplicación y una DLL que encapsula los objetos de acceso a datos:

- *VbADODemo.exe*
- *System.Data.SqlClient.dll*

El programa puede probarse en la misma PC de desarrollo, ya que estando instalado el .NET CF, se cuenta con la misma capa de abstracción de software presente en el módulo Digi y corriendo sobre el WinCE 6.0 embebido.

Si lo desean pueden modificar el código, y recompilar el proyecto, y tomar luego el archivo binario, de la carpeta correspondiente al modo de compilación (Release o Debug).

Para ejecutar la aplicación en el módulo Digi, puede hacerse de dos maneras diferentes:

- Directamente desde el IDE de Visual Studio y la aplicación podrá ser debuggeada pasando a través de ethernet, hacia el módulo y grabándose en RAM.
- O pueden grabarse los archivos antes mencionado en un pen-drive para luego conectarlo al módulo Digi y ejecutar la aplicación desde allí, tal como si de una unidad de disco se tratara.

Aclaración importante:

Para realizar la conexión con el servidor SQL server se utiliza desde el programa, un string de conexión que especifica a los objetos de acceso a datos los parámetros que deben utilizar para conectarse a la base de datos. Para lograr dicha conexión se debe adaptar este string al servidor de base de datos que se esté utilizando. El string de conexión utilizado en la demo puede encontrarse dentro de la función `Connect2server()` en el código del formulario *frmADOTest* y es el siguiente:

```
"Data Source=" & strServer & ",1433;Initial _
Catalog=bdContinea; User Id=sa;Password=MagicCookie;"
```

Donde:

- `strServer` es el nombre del host o la dirección IP del servidor SQL server, aquí este dato es enviado como parámetro y puede cambiarse en tiempo de ejecución y reintentarse la conexión.
- `1433` es el puerto donde escucha el servidor.
- `BdContinea` es el nombre de la base de datos.
- `sa` es el nombre de usuario para autenticación contra el SQL server.
- `MagicCookie` es el password.

Adicionalmente en la función inicial de la aplicación: `frmSQLtest_Load()` se realiza el siguiente llamado a función: `SetStatusBD(Connect2server("Contimovil", clsConn))` conectarse al servidor cuando la aplicación comienza.

El usuario debería reemplazar el string `"Contimovil"` por el nombre o dirección IP del host correspondiente al servidor SQL server de su red.

 CONTINEA <small>Microprocesamiento modular + Conectividad</small>	Cliente SQL en Windows CE 6.0 / Digi ConnectCore Wi9C	Nota de Aplicación CoAN-004
	Lector de códigos con GUI para comercio.	Publicado: 00/00/0000 Página 10 de 10

4. Prueba

Finalmente, en estas instancias, estaríamos en condiciones de probar el sistema completo:

- El servidor SQL server express 2005 está ejecutándose en algún lugar de nuestra red con la base de datos *bdContinea* adjuntada.
- Tenemos nuestro módulo Digi funcionando, ejecutando Windows CE 6.0, y conectado a nuestra red local mediante ethernet o WiFi.
- El lector de RFID está conectado al puerto serie de la placa de desarrollo y listo para leer y transmitir los ID's.
- El display color o el monitor se encuentra también conectado y allí podemos ver en pantalla la clásica interfaz gráfica de Windows.
- Los componentes para el cliente de acceso a datos del CF están instalados en Windows CE.
- Nuestra aplicación espera que alguien la ejecute, almacenada, por ejemplo, en el pen-drive.

Solo queda iniciar el programa de la forma habitual en que se hace esto en Windows y la aplicación se mostrará en la pantalla.

Si la conexión con el servidor de base de datos fue exitosa, se mostrará el icono de la BD limpio, de lo contrario se mostrará atravesado por una cruz roja.

En segundo lugar, se debe activar el lector desde el menu del formulario.

Ahora pueden leerse los códigos de los artículos, y se verán los resultados obtenidos en pantalla.