PROGRAMAR EN JAVA

Continuamos nuestro viaje al maravilloso mundo del lenguaje Java. Esta vez os proponemos un ejemplo de cómo se puede gestionar un puerto USB usando nuestra tarjeta LX.1734 para obtener 1.000 instrumentos diferentes que se pueden utilizar con los sistemas operativos más comunes: Windows, Linux, Mac.

Como hemos mencionado en el artículo "**Programar en Java el puerto serie**", el lenguaje de programación Java nace para la programación orientada a objetoscon la gran ventaja de una fácil portabilidad. Permite crear aplicaciones escribiendo el software una sola vez, sin tener en cuenta el tipo de equipo en el que se ejecutará.

Esta característica es posible gracias a la llamada Máquina Virtual de Java, también conocida como **JVM (Java Virtual Machine)**, que ejecuta el software en la mayoría de las plataformas: Windows, Linux, Mac OS X y Solaris.

Además, Java es un lenguaje de código abierto por lo que es mucho más fácil encontrar las aplicaciones y las librerías sin problemas de copyrght ni licencias de uso. De hecho, en este artículo se hará uso de la librería **RXTXcomm.jar** para la gestión de los puertos serie y paralelos, que se puede descargar gratuitamente desde Internet.

También vamos a ilustrar estos conceptos de forma práctica, con la ayuda de nuestro amigo experto desarrollador de sistemas Dr. Ing. Alessandro Pier

Aisa, que nos ha ayudado a desarrollar un par de aplicaciones Java en un entorno de desarrollo de software integrado IDE (Integrated Development Environment), que se llama **Net- Beans**.

Las dos aplicaciones Java son para usar con la "**Tarjeta USB de 1000 usos**" **LX.1734**, que ya publicamos y es un **monitor de los datos** que pasan a través del puerto USB y de un **DataLogger** para monitorizar los datos que dan los sensores conectados al puerto USB (como la temperatura ambiente, la medición del campo magnético o la medición de la temperatura de la piel usando una termopila).

Como hemos dicho, tratándose de software de código abierto los paquetes necesarios para la instalación del entorno NetBeans IDE y las librerías para gestionar el puerto USB se pueden descargar gratuitamente a través de Internet. Para vustra comodidad, hemos incluido en el CD-Rom que proporcionamos paquetes de instalación para **Windows, Linux y Mac**.

DOS PALABRAS SOBRE LA PROGRAMACIÓN DE OBJETOS

Antes de la llegada de la programación orientada a objetos, la planificación clásica se basaba en el denominado **paradigma de procedimiento**, que consistía en descomponer un problema complejo en problemas más simples, con el fin de identificar y crear las funciones necesarias para la aplicación del software final.

Este enfoque, aunque bien estructurado, tiene problemas de integridad y de consistencia del software, ya que obliga al programador a acordarse siempre de cómo se organizó el software, dónde declaró variables para evitar los incómodos **bugs**, difíciles de identificar.

EL PUERTO USB

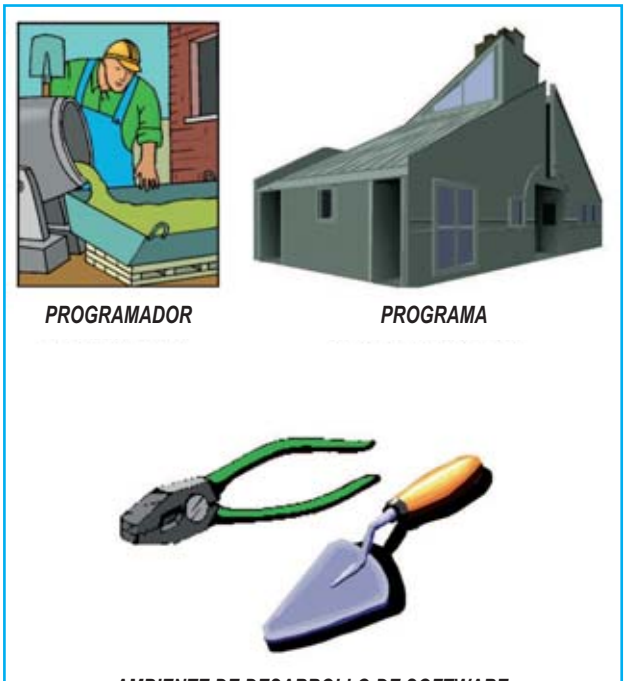


Fig.1 Programación orientada a objetos.

De hecho, si el proyecto es particularmente complejo es probable que genere gran cantidad de módulos de software y la pérdida de visibilidad de las relaciones entre ellos. La programación orientada a objetos, evita estos riesgos porque afronta el problema de manera diferente, adoptando un modo diferente de ver las cosas. Para entender la diferencia entre la programación tradicional y la programación orientada a objetos podemos utilizar el siguiente ejemplo.

Imagine tener que construir una casa. El programador sería el albañil, la casa sería el software y las herramientas del albañil el entorno de desarrollo de software (ver Figura 1).

El "albañil tradicional" para construir una pared de la casa se provee de elementos primarios (hormigón, agua, ladrillos, pintura, pincel), prepara

el cemento y apila los ladrillos uno tras otro para levantar un muro. Cuando termina, prepara la pintura y pinta la pared. Para cada nueva pared el enfoque es siempre el mismo, repitiendo los mismos pasos desde el principio.

El "albañil orientado a objetos", sin embargo, tiene objetos que ya están disponibles prefabricados (que se denominan clases) y de los que puede usar las características que le vengan mejor. En este ejemplo concreto, podemos pensar en paredes prefabricadas, para personalizar con paneles que decidan la forma, el espesor y el color. Cada panel nuevo que el albañil orientado a objetos construya será una instancia de la clase "paredes", es decir, una ocurrencia particular que hereda características de la clase necesaria para obtener el propósito del albañil.

A partir de este ejemplo, se puede concluir que al constructor orientado a objetos se le facilita la tarea porque no tienen que empezar desde lo básico cada vez, pero puede aprovechar los modelos existentes (objetos) que proporcionan funciones y tienen características ya predefinidas y puede establecer las funciones específicas que necesita.

JAVA Y EL ENTORNO NETBEANS IDE

Las aplicaciones Java pueden ser organizadas en muchos tipos diferentes de archivos: por ejemplo, las fuentes tienen la extensión .**java**, los archivos compilados extensión .class y los ejecutables, archivos de bibliotecas y otros archivos tienen la extensión .**jar**.

Una aplicación también puede tener muchas fuentes y usar muchas bibliotecas como soporte. Así, parece claro que para gestionar las aplicaciones de cierta complejidad, no es recomendable trabajar desde la línea de comandos usando un simple editor texto como el **Bloc de notas** de Windows (o **Nano** en Linux, o TextEdit en Mac), sino que es necesario un entorno de programación integrado, que de acceso de forma fácil y rápida a la información. **NetBeans IDE** es un entorno de desarrollo multi-lenguaje escrito enteramente en Java y nacido en torno al año 2000.

Sun **Microsystems** eligió **NetBeans** como IDE oficiale, a diferencia del más popular Eclipse desarrollado por un consorcio de compañías bien conocidas como Intel, HP e IBM, también escrito en Java.

También es posible equipar a Netbeans muchos plug-ins que hacen que sea más completo y multi-lenguaje, aunque requiere un mínimo de **512 megabytes** de RAM para poder usar las librerías gráficas estándar de Java, conocidas como **Swing**.

Citemos para completar algunos de lenguajes de programación más conocidos soportados por NetBeans: **Ajax C/C++, Groovy, Grails, JavaScript Mobile, PHP, Python Ruby, XML**.

PRINCIPALES CARACTERÍSTICAS DEL ENTORNO NETBEANS

NetBeans IDE ofrece todas las posibilidades de un entorno de desarrollo de software de nueva generación. Mencionemos sólo algunas de las características clave, que se utilizarán más adelante para realizar las aplicaciones.

Una de las principales características de NetBeans es el alto grado de **automatización**, que se expresa a través de los informes diseñados para minimizar el esfuerzo del programador en las operaciones de escritura del software. Por ejemplo, es muy útil la función de **code-templates** que permite escribir los comandos más usados mediante abreviaturas **keystrokes**. Escribiendo sout y presionando la tecla TAB, se crea el comando **System.out.println ("")**, auto que se usa para la salida del texto por pantalla.

En la misma familia de características avanzadas se puede citar el **autocompletado de código**, para los objetos.

Tecleando el carácter "." después de un objeto (recuerde que en Java el punto se usa utiliza para

Fig.2 La tarjeta del interfaz USB KM1734K ya publicada y las conexiones a los diferentes accesorios, es decir, el módulo a temperatura ambiente KM1734KT, el conector jack hembra, el diodo LED y su resistencia

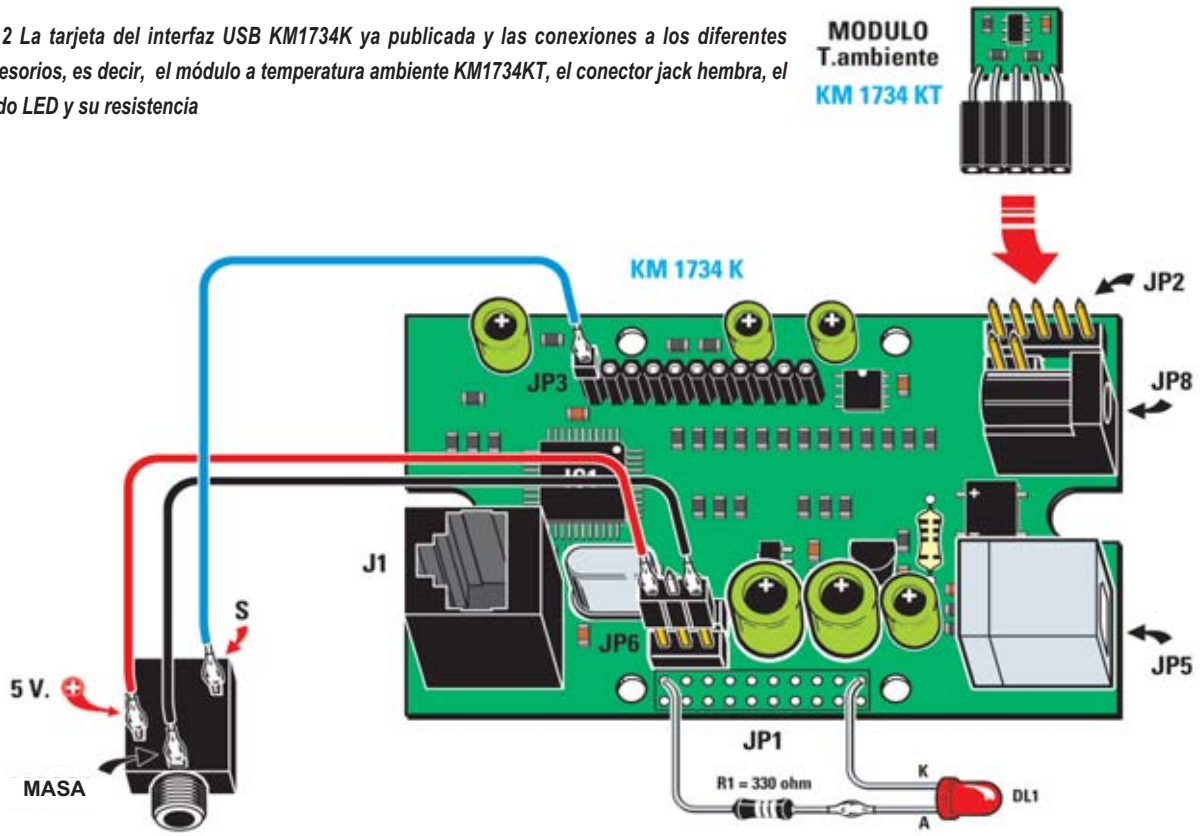
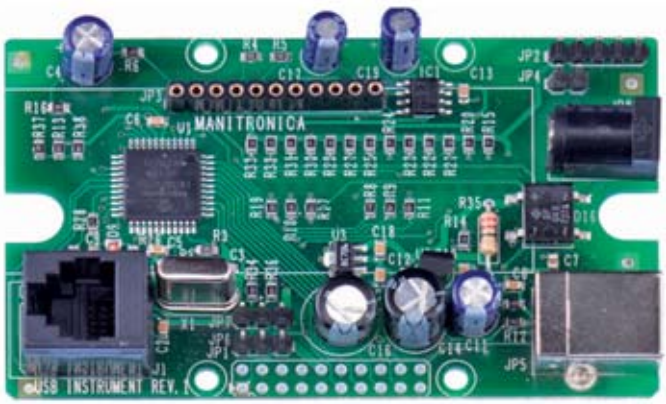


Fig. 3 Foto de la tarjeta del interfaz USB que proporcionamos ya probada y ensamblada con componentes SMD y probado.



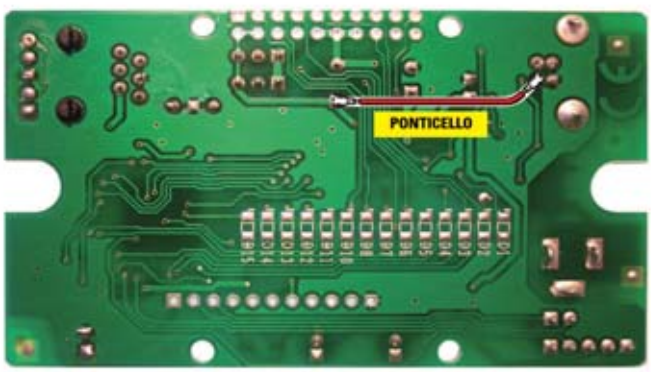


Fig.4 Foto de la tarjeta USB vista desde la parte posterior, es decir, desde el lado desde el que se tiene que soldar el puentecillo necesario para transferir a todos los componentes la alimentación de 5 voltios que proviene del puerto USB del ordenador.

acceder a los métodos o las propiedades del objeto) NetBeans muestra un **pop-up** con la lista de métodos y propiedades para el objeto a considerar y la ayuda en línea.

En lo que respecta a la interceptación de errores no hay que esperar a la fase de compilación del código ya que NetBeans, a medida que se escribe, **notifica** los errores e incluso **sugiere** posibles correcciones dependiendo del tipo de error, permitiendo al usuario comprender dónde hacer las modificaciones necesarias a través de una **bombilla de alarma** junto a la línea de código errónea.

Para la gestión de cambios en los archivos fuente, NetBeans ofrece un alto nivel de "**undo**" (deshacer): es posible eliminar los cambios hechos tecleando CTRL + Z muchas veces para restaurar la situación original.

Además, a través de la función de **Historial Local**, accesible con el botón derecho del ratón, se pueden obtener todas las versiones del archivo guardadas (cada vez que se ejecuta el código de NetBeans éste guarda las fuentes, propiedad conocida como "Guardar y compilar").

Una función fundamnetal es la que se conoce con el nombre de "**refactor**", que garantiza la posibilidad de reestructurar el software de forma automática; por ejemplo, para renombrar un objeto dentro del proyecto, NetBeans se encarga de realizar la operación de forma consistente, encontrando todas las ocurrencias del objeto y modificándolas.

NetBeans también tiene un potente **depurador**, que proporciona todas las funciones para ayudar al programador a encontrar errores (**bugs**) en el software.

La instalación de NetBeans IDE 6.9

En el momento de escribir este artículo la última versión disponible de NetBeans es la 6.9 que se puede instalar desde el CD o por Internet en el sitio **www.netbeans.org**

Instalación para Windows

En el CD hay que acceder a la carpeta de NetBeans y hacer doble clic en el archivo **netbeans-6.9-ml-javase-windows.exe**

Este programa instalará en su PC la versión de NetBeans IDE 6.9 y el Java SE Development Kit.

Instalación para Linux

En el CD, hay que acceder a la carpeta de Linux\jdk y copiar en la ubicación /usr/lib/jvm/ el archivo **jdk-6u17-linux-i586.bin**.

Luego desde la ventana de terminal situarse en /Usr/lib/jvm/ y ejecutar el comando ./**jdk-6u17-linux-i586.bin**.

Finalmente, de nuevo desde el CD, hay que acceder a la carpeta de Linux \NetBeans y hacer doble clic en el archivo **netbeans-6.9-ml-Javas-linux.sh**.

Estos comandos instalarán en el PC **Java Software Development Kit y NetBeans IDE versión 6.9**.

Instalación para Mac

Desde el CD, es necesario acceder a la carpeta Mac\NetBeans y hacer doble clic en el archivo **Netbeans-6.9-mljavasemacosx.dmg**.

El programa instalará NetBeans en la carpeta Aplicaciones.

TUTORIAL y DEMO

Una vez finalizada la instalación, se puede iniciar el entorno con doble clic en el icono "**NetBeans IDE 6.9**" que habrá en el escritorio. La página de inicio que aparece a la derecha del panel cuando se arranca IDE tiene dos vistas: **Welcome to NetBeans IDE y My NetBeans**.

Ambos ofrecen muchos enlaces de interés, como algunos tutoriales y demos, que explican paso a paso cómo crear aplicaciones. Todos los enlaces

```
/* Nuova Elettronica - 2010
 * LeggiUSB.java 2.0
 * Date: 04/08/2010
 * Author: P.A. Aisa
 * Copyright (c) 2010 Nuova Elettronica. All Rights Reserved.
 */

import java.io.InputStream;
import java.io.IOException;
import gnu.io.*;

public class Main {
    public static void main(String[] args) {
        // Dichiarazione variabili
        int                     idx = 0;
        CommPortIdentifier      portId ;
        SerialPort              port = null;
        InputStream             in;
        int                     charx ;
        int                     charread = 0 ;
        // Stringa di inizio programma
        System.out.println("START Program USB: Linea comando " + args[0]+ " "
                + args[1]);
        // Apertura Porta
        try
            {
                portId = CommPortIdentifier.getPortIdentifier(args[1]);
               System.out.println("Apertura porta " + portId.getName());
               System.out.println("Porta "
                + portId.getName() + " aperta con successo");
            }
        catch (NoSuchPortException e)
            {
               System.out.println("Porta " + args[1] + " non trovata !");
                portId = null ;
            }
        try
            {
                port = (SerialPort) portId.open("TempLogger", 2000);
                if (port == null)
                    {
                        System.out.println("Errore nell'apertura della porta"
                      + portId.getName());
                    }
            }
        catch (PortInUseException e)
            {
                System.out.println("Attesa della coda per la porta  "
                + portId.getName() + ": porta utilizzata da " + e.currentOwner);
            }
        // Lettura e Stampa dati da USB
        System.out.println("**********************\nLettura dati da USB" +
                "\n**********************");
        try
            {
                in = port.getInputStream();
                while (charread <300)
                    {
                        charx =  in.read();
                        System.out.print((char)charx) ;
                        charread++ ;
                    }
            }
        catch (IOException e)
                {
                 System.out.println("Non riesco ad aprire la input stream");
                }
    // Chiusura Porta
        System.out.println("**********************\nChiusura porta " + port) ;
        port.close() ;
    }}
```

1 2 3 4 5 6

Figura 5

de la página de inicio requieren tener conexión a Internet para su visualización. En concreto, os recomendamos ver el Quick Start Tutorial, que en pocos minutos os permitirá crear un programa sencillo que imprime en video el típico "**Hello World!**".

Dos aplicaciones para la gestión del puerto USB

Ahora crearemos un par de aplicaciones Java usando NetBeans y la tarjeta "**USB 1.000 usos**" **LX.1734** diseñada por nuestro estimado colaborador, el Dr. Ing. Alessandro Manigrassi.

Para simplificar la tarea y explicar bien la diferencia entre la programación tradicional y la programación orientada a objetos, procedemos a la creación de dos aplicaciones Java:

- LeerUSB: aplicación Java sin interfaz gráfico, realizada de acuerdo con el enfoque tradicional para monitorizar el tráfico de datos generado por la tarjeta **LX.1734** en el puerto USB.

- DataLogger: aplicación con interfaz gráfico diseñado de acuerdo a los cánones de la programación por objetos para el seguimiento de determinados entornos físicos como la temperatura y el campo magnético adquiridos por la tarjeta **LX.1734**.

APLICACIÓN LEERUSB.JAVA en entorno WINDOWS

Las siguientes instrucciones se refieren a la plataforma Windows. Para la instalación en Linux y Mac, en la siguiente sección.

La aplicación LeerUSB a través del puerto USB, lee los datos generados por la tarjeta **LX.1734**, relativos a los canales de captura analógica digital e imprime el contenido en video. El ejemplo está simplificado deliberadamente para entender cómo manejar el puerto USB que se usa como **puerto serie** "**virtual**" (el puerto serie se llama virtual porque es el controlador que soporta el puerto USB, que lo muestra al sistema como un puerto serie).

El programa consta de un solo archivo que contiene la única **main**, necesaria para la **Máquina Virtual de Java** para ejecutar el programa (ver las referencias a los bloques en la Fig. 5 donde se reporta el código), con la siguiente estructura:

1 – importación de las librerías necesarias
2 – declaración de variables
3 – apertura del puerto serie
4 – lectura desde el puerto de datos
5 – impresión de datos en video
6 – cierre del puerto serie

Antes de utilizar NetBeans debe instalar el driver **USB-serial**. Para ello, conecte la tarjeta y cuando se le pida la ubicación de los archivos para el driver hay que dar como ubicación dar el CD en la carpeta Windows\driverCCS, tal y como se muestra en las fig.6-7.

Figura 6

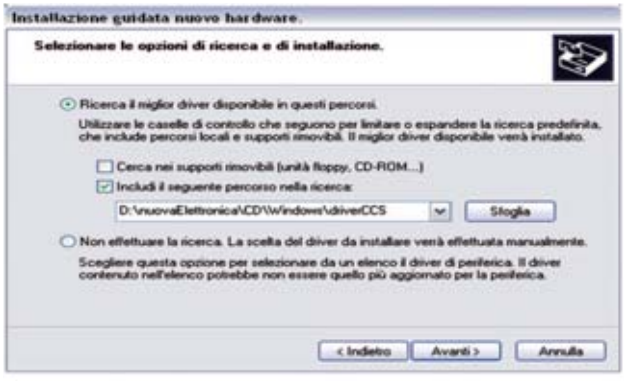


Figura 7

Luego hay que copiar las carpetas **LeerUSB** y **RXTX** desde el CD a la ubicación de trabajo (por ejemplo, en la carpeta nE en la unidad C, es decir, la ruta C:\nE).

Para crear el primer proyecto de NetBeans hay que seguir las siguientes instrucciones:

1 – Abrir NetBeans desde el menú Programas de Windows y crear un nuevo proyecto, haciendo clic en el segundo botón de la barra de menú, como se muestra en la fig.8: Seleccione **Java Project with Existing Sources** y haga clic en el botón **Siguiente**.

Figura 8

Ahora escriba el nombre **LeerUSB** en el primer campo y **C:\nE\LeerUSB** en el segundo campo (ver figura 9) y haga clic en "**Siguiente**".

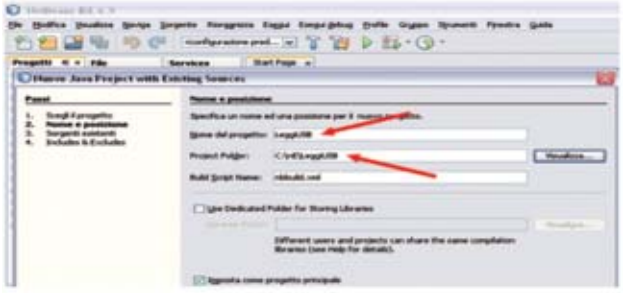


Figura 9

Ahora haga clic en el botón **Agregar carpeta...**, seleccione la carpeta **C:\nE\LeerUSB\sources** (como se muestra en la figura 10) y haga clic en **Siguiente** y en el botón **Finalizar**.

En este punto, en el entorno IDE aparece el proyecto LeerUSB compuesto por <**paquetes** (equivalentes a carpetas) predefinidos> y por librerías.

2 – Ahora es necesario introducir en los archivos del proyecto la librería para manejar el puerto serie: **RXTXcomm.jar**. Hasta que no se haga esta operación IDE informará de errores en todas las referencias a los objetos de esta librería.

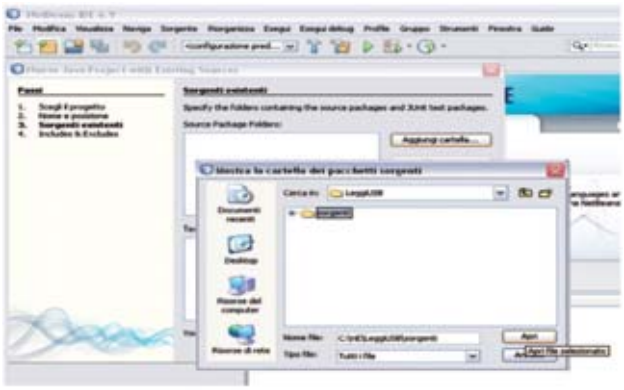


Figura 10

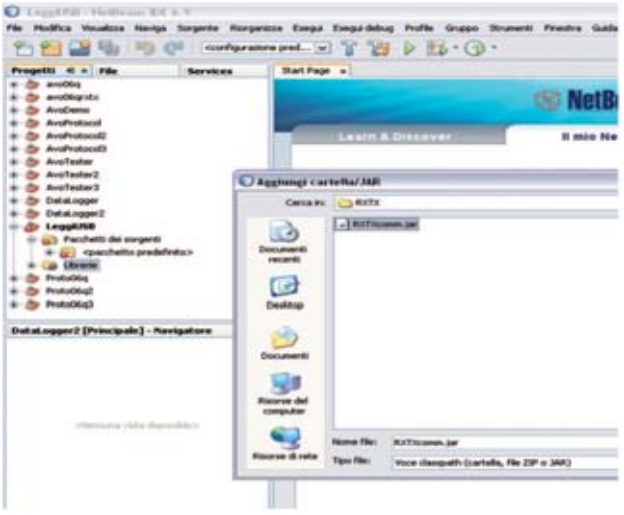


Figura 11

Por tanto, seleccione el paquete de **Librerías** pulsando el botón derecho del ratón y seleccione Agregar **directorio/JAR** y seleccionar en la ubicación **C:\nE\RXTX** el archivo **RXTXcomm.jar** y luego hacer clic en **Abrir**, como se muestra en la figura 11.

3 – Ahora hay que configurar la ejecución del programa mediante la introducción de la línea de comandos para especificar el puerto serie al que conectar la tarjeta USB.

Seleccionar en el menú de NetBeans **Ejecutar, Set Project Configuration...**, **Personalizar...** y especificar en la línea de comandos "-p COM4", sustituyendo **COM4** como puerto serie al que está conectada la tarjeta USB, como se muestra en la fig.12, y pulsar **Aceptar**. De hecho, en nuestro caso es **COM4**.

Para saber en qué puerto serie del PC se ve la tarjeta, se puede seleccionar con el botón derecho en **Recursos de Mi PC** y luego en **Propiedades**; desde el panel de **Gestión de Periféricos**, ir a **Puertos** (COM y LPT) y buscar el número de puerto COM correspondiente a "**USB to UART**", como se muestra en la Figura 13.

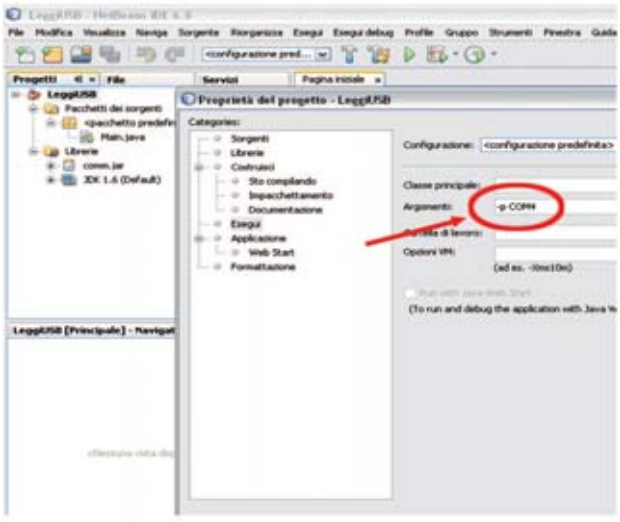


Figura 12

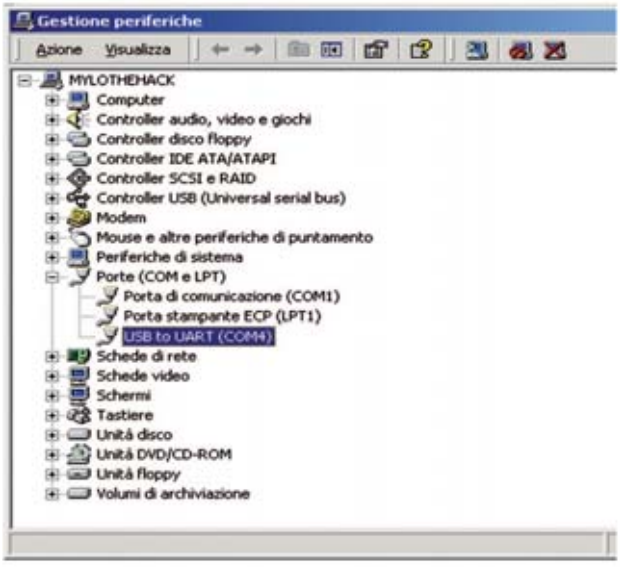


Figura 13

4 – En este punto, sólo tiene que conectar la tarjeta **LX.1734** y ejecutar el código, haciendo clic en el botón con la "flecha verde" o pulsando F6.El resultado debería aparecer en la ventana Output del entorno IDE, como se muestra en la fig. 14.

El programa se inicia con la línea "START PROGRAM USB", escribiendo la línea de comandos que se insertó en la fase de ejecución. A continuación, se abre el puerto serie especificado y si la operación ha tenido éxito empieza la impresión de los caracteres recibidos desde el puerto USB.

Por simplificar, el programa ha sido establecido con 300 caracteres de impresión. Para familiarizarse con NetBeans, se recomienda cambiar el código fuente aportando algunos cambios y volverlo a ejecutar para evaluar sus efectos. Por ejemplo, se puede variar el número de caracteres adquiridos.

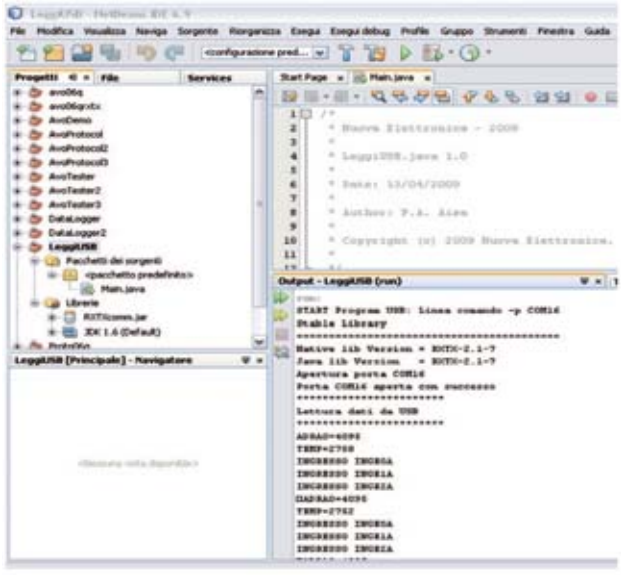


Figura 14

APLICACIÓN LEERUSB.JAVA en entorno LINUX

De lo indicado en el punto anterior se mantiene todo igual salvo las rutas que tienen que indicarse como archivos de sistema de tipo Linux (por ejemplo, la carpeta de trabajo de Windows C:\nE se enruta como linux/nE).

Para gestionar el puerto USB como un serial virtual bajo Linux, se debe instalar el módulo escribiendo la siguiente línea de comando (si no está prevista por la distribución Linux que se esté utilizando): **modprobe usbserial**

Para verificar la correcta carga del módulo de la línea de comandos, se puede escribir: **lsmod | grep**

usb. y verificar que aparezca como respuesta una línea de texto que contiene “usbserial”. Se deben copiar las carpetas LeerUSB y RXTX del CD a la ubicación del trabajo (por ejemplo, en la carpeta \nE) y las librerías “librxtxSerial.so“ y “**librxtxParallel.so**“, que están en el CD ebn la ubicación **Linux\RXTX** en la ruta “**\usr\lib**“.

Ahora repita los pasos 1 y 2 descritos en el párrafo año anterior.

Antes de proceder con el paso 3 es necesario para verificar cuál es el nombre del puerto serie asignado por Linux, cuando la tarjeta **LX.1734** está conectada al PC. Para comprobarlo una vez conectada la tarjeta, hay que escribir en una ventana de terminal la línea de comando: **ls-l/dev/tty** * y asegurarse de que aparezca el dispositivo tty (por ejemplo, nuestra distribución de Linux ha asignado **ttyACM0** a la tarjeta USB).

Continúe con el paso 3 descrito en el párrafo anterior poninedo como línea de configuración “**-P/dev/ttyACM0**” en lugar de “**p-COM4**” y lanzar la aplicación como se describe en el punto 4.

APLICACIÓN LEERUSB.JAVA en entorno Mac

De lo indicado en los párrafos anteriores se mantiene todo sin cambios a diferencia de las rutas que se gestionan como carpetas de Mac. Para administrar el puerto USB como una serial virtual debe instalar el driver USBserial haciendo doble clic en el archivo **PL2303_1.2.1r2.dmg** que se encuentra en el CD en la ruta \Mac\usb-serial\. A continuación, debe copiar las carpetas **LeerUSB** y **RXTX** desde el CD a una carpeta de trabajo (por ejemplo, en el escritorio). Copiar el archivo **librxtxserial.jnilib** del CD a la ruta \Mac\RXTX\libreria\Java\extensiones disponible en el icono **Macintosh HD** en el Finder.

Ahora repita los pasos 1 y 2 descritos en el párrafo anterior.

Antes de proceder con el paso 3 es necesario verificar cuál es el nombre del puerto serie asignado por Linux, cuando la tarjeta **LX.1734** está conectada al PC. Para ello, una vez conectada la tarjeta, hay que teclear la línea de comandos desde una ventana de terminal: **ls-l/dev/tty*** y asegurarse de que aparezca el dispositivo tty (por ejemplo, nuestro Mac ha asignado el nombre **tty.usbmodem431**).

Ahora realice el paso 3 descrito en el párrafo anterior, introduciendo cadena de configuración “**-p/dev/tty.usbmodem431**“ en lugar de “**-p COM4**“ y ejecutar la aplicación como se describe en el paso 4.

APLICACIÓN DATALOGGER

Esta aplicación utiliza siempre el puerto USB de la tarjeta **LX.1734** a fin de monitorizar en el tiempo las magnitudes alcanzadas por los sensores conectados a la misma. También ofrece la posibilidad de ver la magnitud de los gráficos de tipo XY en tiempo real y guardar los datos en archivos, incluso muy largos, para la grabación de los cambios en las magnitudes.

Para este artículo hemos seleccionado algunos módulos ya publicados anteriormente, el módulo de temperatura ambiente **KM1734KT**, el módulo de gausómetro **LX.1734/2** y el módulo de termopila **LX.1734/4**. Comparado con el ejemplo anterior, este programa se llevó a cabo siguiendo el enfoque de programación orientada a objetos, utilizando objetos de diseño gráfico ofrecidos por NetBeans.

En relación con el código fuente que se muestra en la figura 15, como ejemplo tomamos nota de la estructuración programa con una clase principal **Data Logger ()**, que proporciona propiedades y métodos. Los métodos principales se pueden resumir de la siguiente manera: **OpenCom, ReadCOM, CloseCOM** = como en la aplicación LeerUSB.java sirven para abrir, leer y cerrar el puerto serie virtual conectado al puerto USB.

WriteData, PlotCOM = sirven para actualizar gráficos de la ventana principal.

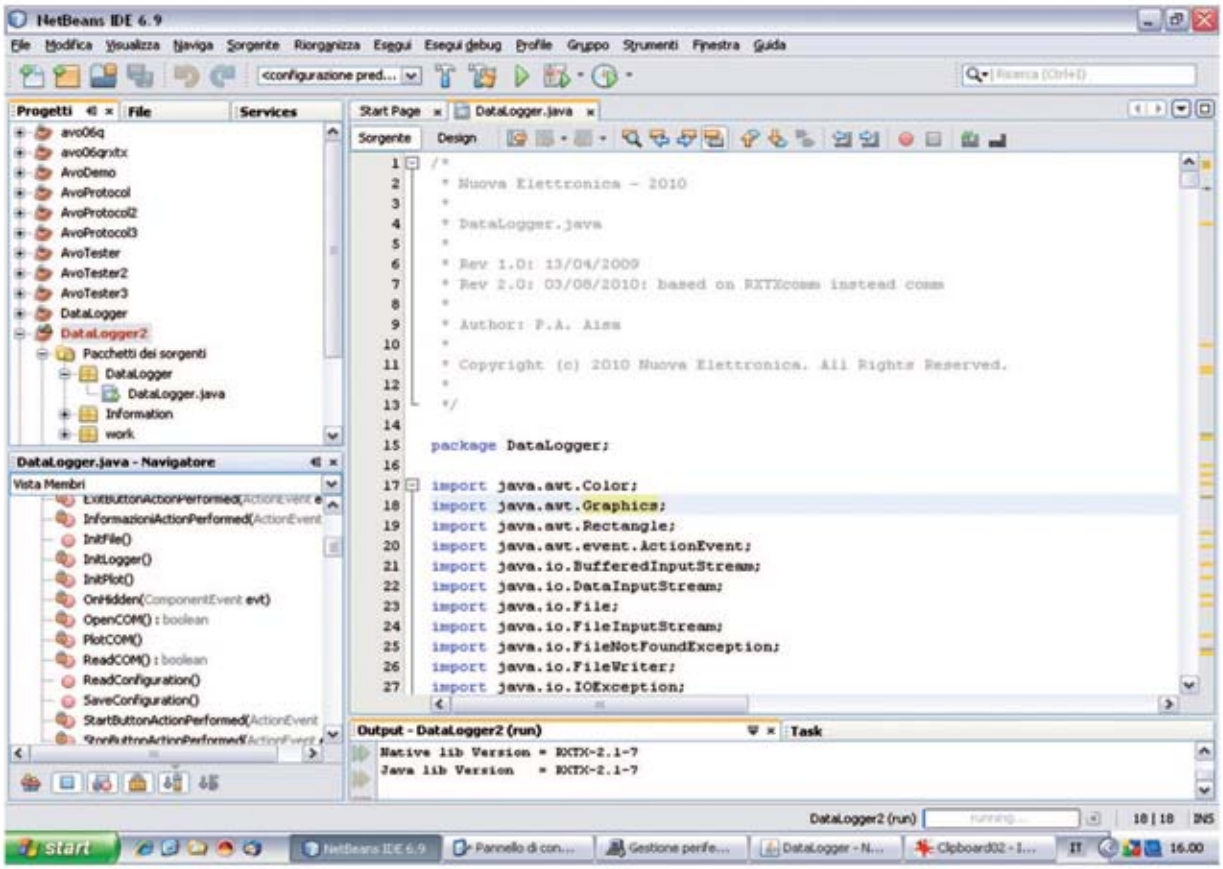


Figura 15

Figura 16

WriteFile = se utiliza para gestionar la escritura en los archivos.

Funcionamiento de la aplicación DATALOGGER

Para ejecutar el programa debe copiar la carpeta DataLogger del CD a una carpeta de trabajo (por ejemplo, C:\nE\Data Logger\) y luego hacer dible clic en el archivo **DataLogger.jar**.

Como primer paso es necesario rellenar los campos de la máscara representada en la fig.16.

Sensor = Seleccionar el sensor conectado a la tarjeta USB.

Puerto serie = es el nombre del puerto serie virtual asignado al puerto USB (ver fFigura 13, para identificar en qué puerto COM o TTY se ha "mappata" la tarjeta USB).

Reloj = representa el tiempo en segundos que transcurre entre dos adquisiciones sucesivas (debe ser un número entero diferente a 0).

TiempoMax = es el tiempo de ejecución de la adquisición en segundos desde el momento en que se pulsa el botón de **Start**, si no se activa la casilla "Ahora Inicio\Stop" (debe ser un número entero distinto de 0).

Intervalo X = es la resolución horizontal del gráfico. Introduzca un valor entero comprendido entre 4 y 200.

Intervalo Y = es la resolución vertical del gráfico. Introduzca un valor mayor que cero y menor que el fondo de la escala que se desea utilizar.

Emisividad = este menú desplegable se utiliza cuando se conecta una termopila y define el coeficiente de emisividad para dar la temperatura adquirida por la termopila.

Offset Start (horas) = número de horas con respecto a la actual a las que el programa iniciará la adquisición, si el cuadro Programación Start/Stop está seleccionado.

Offset Stop (horas) = número de horas con respecto a la actual a las que el programa terminará la adquisición, si el cuadro Programación Start/Stop está seleccionado.

Programación Start/Stop = si se selecciona, el programa se inicia a la hora resultante de la suma de la actual y la inndicada en el campo "offset start".

Trigger+ Trigger- = representa el nivel del eje Y en el que se colocan el trigger positivo y negativo. Si el "log out trigger" está seleccionado, sólo se adquirirán los valores que definidos por el intervalo de trigger+ y trigger-.

Logout Trigger = si se selecciona, el programa sólo captura los datos que están fuera del rango definido por los campos Trigger+ y Trigger-.

Beep on Trigger = si se selecciona, el equipo emite un sonido para los datos que están fuera del rango definido por los campos Trigger+ y Trigger-.

Log File = en este campo debe introducir el nombre del archivo que se creará y que contendrá todos los datos adquiridos.

Fecha y hora = si se selecciona, el formato de hora también incluye la fecha. La clave de la "información" proporciona información el programa.

En este punto, el registrador de datos se puede iniciar pulsando en el botón **Start** y detener pulsando **Stop**.

Para borrar la ventana de registro se puede utilizar el botón **Cancelar Log**.

Para salir del programa se puede utilizar el botón **Esc** o cerrar la ventana con el botón X, arriba a la derecha.

Cuando haya terminado se puede abrir el archivo de texto con el título definido por el el campo Log File, que contiene los datos adquiridos en el directorio C:\nE\Data Logger.

COSTE DE EJECUCIÓN

Interfaz USB **LX.1734** (ver fig. 2-3-4), incluyendo la tarjeta de **KM.1734K** premontada en SMD y el **CD-Rom CDR1734J** que contiene el código fuente, los drivers y las aplicaciones para instalar (excluido el módulo de temperatura **KM1734KT**): **135,00 euros**.

*Nota: bajo pedido también ofrecemos el módulo de temperatura ambiente **KM1734KT** (ver figura 2): **27,00 euros***

Los precios **no** incluyen el **IVA**, ni gastos de envíos a domicilio.