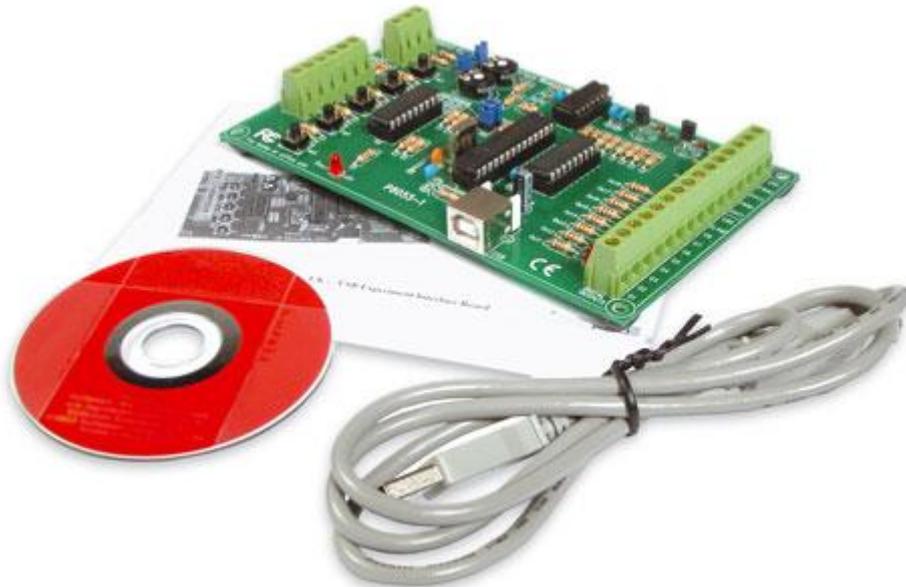




Disponibile

Prezzo € 38,00

### INTERFACCIA USB PER PC CON DEMOBOARD



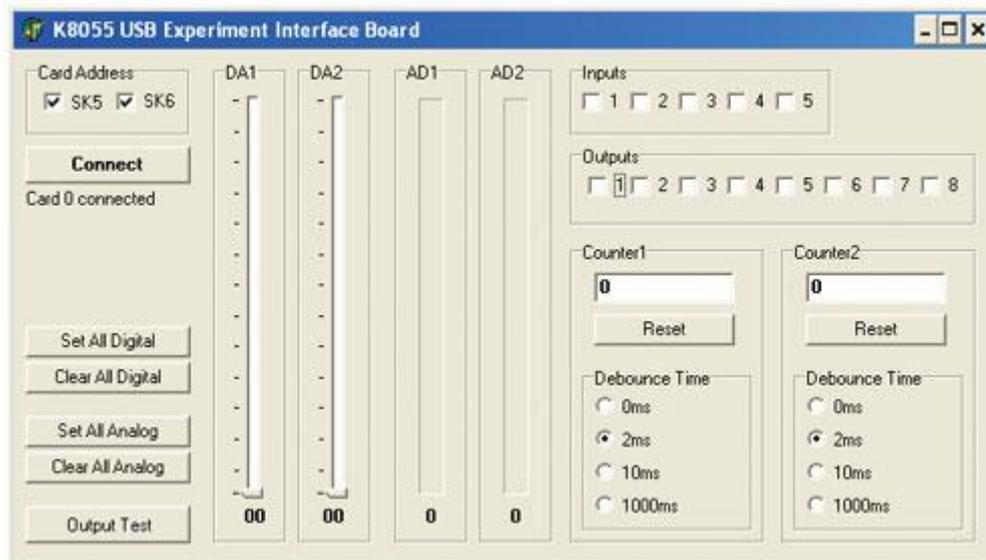
Questa scheda, collegata al PC tramite la porta USB, consente di **controllare delle uscite** analogiche e digitali nonché di "**leggere**" degli ingressi sia analogici che digitali. A tale scopo viene fornito un apposito software da caricare sul PC.

A questa prima interessante applicazione se ne aggiunge un'altra, forse più importante: vengono infatti forniti tutti gli strumenti software per consentire a chiunque di **realizzare un programma personalizzato**. Per rendere più agevole questa operazione, sulla scheda sono presenti alcune risorse per simulare segnali di ingresso di vario genere nonché led di segnalazione sulle varie uscite. L'interfaccia dispone di 5 canali di ingresso digitali e 8 canali di uscita digitali. In più, sono presenti due ingressi analogici e 2+2 uscite analogiche caratterizzate da una risoluzione di 8 bit. Il numero di ingressi/uscite può essere espanso in seguito collegando fino a un massimo di 4 schede al connettore USB del PC. Con la scheda viene fornito un completo programma (scritto in Delphi) in grado di gestire tutte le risorse ovvero tutti gli ingressi e tutte le uscite. Tutte le routine di comunicazione sono contenute all'interno della Dynamic Link Library (DLL) K8055D.DLL fornita insieme al kit unitamente ad alcuni esempi di utilizzo in Delphi, Visual Basic, C++ Builder. Chiunque, con questi strumenti, potrà realizzare un pannello di controllo personalizzato!

#### CARATTERISTICHE E PROGRAMMA

- 5 ingressi digitali (0=massa, 1=aperto) (tasto di test disponibile sulla scheda);
- 2 ingressi analogici con opzioni di attenuazione e amplificazione (test interno di +5V disponibile);
- 8 uscite digitali open collector (valori massimi: 50V/100mA, LED di indicazione sulla scheda);
- 2+2 uscite analogiche (da 0 a 5V, impedenza di uscita 1,5K) o onda PWM (da 0% a 100% uscite di open collector; livelli massimi: 100mA/40V, indicatori a LED presenti sulla scheda);
- tempo di conversione medio: 20ms per comando;
- alimentazione richiesta dalla porta USB: circa 70mA;

- include DLL software per diagnostica e comunicazione- test separati degli ingressi/uscite;
- funzione di "clear all" e "set all";
- funzione di conteggio sugli ingressi 1 e 2 con possibilità di impostare il periodo dell'antirimbalo;
- cursori indicanti il livello dell'uscita analogica;
- indicazione mediante grafo a barra dell'ingresso analogico;
- Requisiti minimi di sistema:
  - CPU di classe Pentium;
  - connessione USB1.0 o superiore;
  - sistema operativo Windows 98SE or superiore (Win NT escluso);
  - lettore CD ROM e mouse.

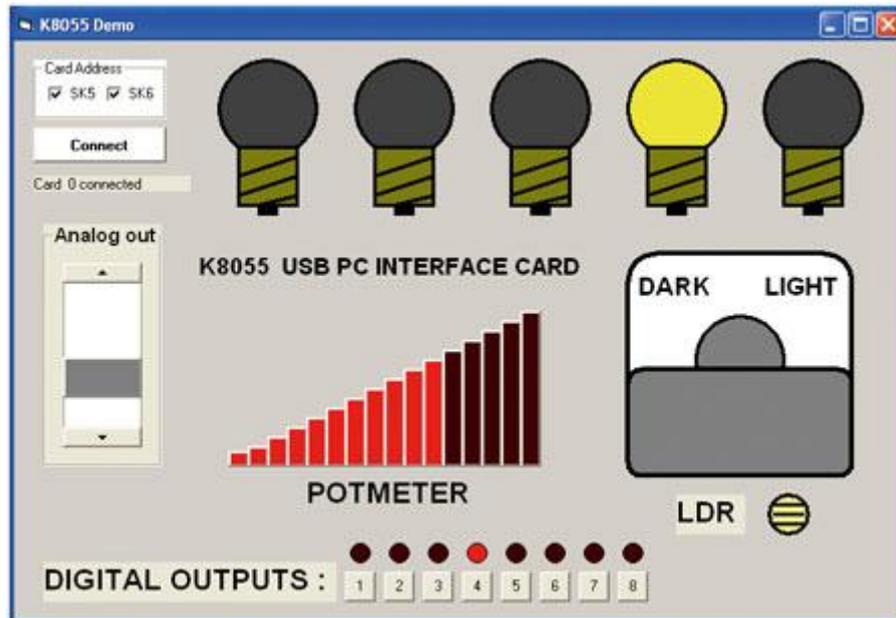


Il software (sviluppato in Delphi) fornito con questo dispositivo consente una gestione completa di tutte le risorse della scheda. Mediante il pannello di controllo è possibile gestire quattro differenti schede in quanto ciascuna interfaccia dispone di un proprio ID scelto tra quattro diverse combinazioni. Ogni scheda dispone infatti di due ponticelli contrassegnati dalle sigle SK5 e SK6: chiudendo o meno i ponticelli è possibile assegnare l'indirizzo voluto. Lo stato ON corrisponde ovviamente alla presenza del ponticello mentre lo stato OFF indica l'assenza del ponticello. Queste impostazioni vanno effettuate prima di collegare l'interfaccia al computer o prima di accendere il PC. Ricordiamo che le schede connesse al PC possono essere gestite una sola alla volta. Infatti, anche nel pannello di controllo, dobbiamo selezionare l'indirizzo della scheda da controllare. Sono presenti due caselle denominate SK5 ed SK6: il segno di spunta corrisponde allo stato "ON". Per stabilire la connessione con la scheda selezionata è necessario cliccare su Connect: se la procedura di indirizzamento è stata eseguita correttamente e la scheda selezionata è presente, apparirà la scritta "Card x connected". Sulla sinistra del pannello di controllo sono presenti una serie di comandi che permettono di porre le uscite digitali o analogiche al livello di alimentazione o a massa. Cliccando su Output Test, le 8 uscite digitali si attiveranno una alla volta ciclicamente. La sezione Inputs mostra lo stato degli ingressi: quando un ingresso viene portato a massa (normalmente è "alto") la casella corrispondente sull'interfaccia grafica appare spuntata. Analogamente selezionando una delle 8 caselle relative alle uscite digitali (Outputs) la corrispondente uscita si porta "bassa". Sulla scheda sono presenti anche due sezioni che consentono di contare il numero degli impulsi che giungono ai relativi ingressi. I contatori 1 e 2 sono convertitori a 16 bit integrati nell'hardware; vengono fatti scattare da I1 e I2. E' possibile testare il contatore con i pulsanti SW1 e SW2: il contatore aggiunge 1 ogni volta che si preme uno dei due pulsanti. Il controllo relativo all'eliminazione del rimbalzo consente di determinare il tempo di reazione del contatore (0ms - 2ms - 10 ms - 1000ms). Le uscite analogiche possono essere impostate agendo sui comandi DA1 e DA2 mentre la visualizzazione dei segnali analogici di ingresso è affidata alle barre AD1 e AD2. Anche in questo caso è possibile effettuare delle simulazioni utilizzando RV1 e RV2.

[Scarica il software di gestione!!](#)

## UN PANNELLO DI CONTROLLO IN VISUAL BASIC

Presentiamo di seguito un altro programma per il controllo dell'interfaccia USB realizzato in Visual Basic mediante il quale, come nel caso precedente, è possibile controllare tutte le risorse della scheda.



[Scarica l'applicativo in Visual Basic!!](#)

## K8055.DLL: PANORAMICA DELLE PROCEDURE

NB: Gli esempi riportati nella descrizione della DLL sono scritti in Delphi.

### Procedure generali

*OpenDevice (CardAddress)*: Apre il collegamento al dispositivo;

*CloseDevice*: Chiude il collegamento al dispositivo.

### Procedure convertitore analogico/digitale

*ReadAnalogChannel (Channel no)*: Legge lo stato di un canale di ingresso analogico;

*ReadAllAnalog (Data1, Data2)*: Legge lo stato di entrambi i canali di ingresso analogici.

### Procedure conversione digitale/analogica

*OutputAnalogChannel (Channel, Data)*: Imposta il canale di uscita analogico in funzione dei dati;

*OutputAllAnalog (Data1, Data2)*: Imposta entrambi i canali di uscita analogici in funzione dei dati;

*ClearAnalogChannel (Channel)*: Imposta il canale di uscita analogico al minimo;

*ClearAllAnalog*: Imposta i canali di uscita analogici al minimo;

*SetAnalogChannel (Channel)*: Imposta il canale di uscita analogico al massimo;

*SetAllAnalog*: Imposta i canali di uscita analogici al massimo.

### Procedure uscita digitale

*WriteAllDigital (Data)*: Imposta le uscite digitali in funzione dei dati;

*ClearDigitalChannel (Channel)*: Azzera il canale di uscita;

*ClearAllDigital*: Azzera tutti i canali di uscita;

*SetDigitalChannel (Channel)*: Imposta il canale digitale;

*SetAllDigital*: Imposta tutti i canali digitali.

### Procedure e funzioni ingresso digitale

*ReadDigitalChannelIn (Channel)*: Legge lo stato del canale di ingresso;

*ReadAllDigital (Buffer)*: Legge lo stato di tutti i canali di ingresso.

### Procedure e funzioni counter

*ResetCounter (CounterNr)*: Resetta il contatore di impulsi a 16 bit numero 1 o numero 2;

*ReadCounter (CounterNr)*: Legge il contenuto del contatore di impulsi numero 1 o numero 2;

*SetCounterDebounceTime (CounterNr, Debounce Time)*: Imposta il tempo di antirimbato del contatore di

impulsi.

### **Procedure e funzioni della K8055D.DLL**

#### **OpenDevice**

Apri il collegamento con la scheda e carica i driver necessari a comunicare tramite la porta USB. Questa procedura deve essere eseguita prima di ogni tentativo di comunicare con la scheda. Questa funzione si può anche utilizzare per selezionare la scheda attiva per leggere e scrivere i dati. Tutte le routine di comunicazione dopo questa chiamata di funzione sono indirizzate a questa scheda fino a quando non viene selezionata l'altra scheda da parte di questa chiamata di funzione. La sintassi è la seguente:

```
FUNCTION OpenDevice (CardAddress: Longint): Longint;
```

I parametri da utilizzare sono:

*CardAddress*: Valore compreso fra 0 e 3 corrispondente all'impostazione dei ponticelli (SK5, SK6);

Il risultato è il seguente:

*Longint*: In caso positivo, il valore di ritorno sarà l'indirizzo della scheda letto; se il valore di ritorno è -1 significa che la scheda non è stata trovata.

Esempio:

```
var h: longint;
```

```
BEGIN
```

```
h:=OpenDevice(0); //Apri il link con la scheda num. 0
```

```
END;
```

#### **CloseDevice**

Scarica le routine di comunicazione per la scheda e scarica il driver necessario a comunicare tramite la porta USB. Questa è l'ultima azione del programma applicativo prima dell'interruzione.

La sintassi è la seguente:

```
PROCEDURE CloseDevice;
```

Esempio:

```
BEGIN
```

```
CloseDevice; // La comunicazione con il dispositivo è chiusa
```

```
END;
```

#### **ReadAnalogChannel**

Il convertitore AD selezionato legge la tensione di ingresso relativa e la converte in un valore compreso tra 0 e 255. La sintassi è la seguente:

```
FUNCTION ReadAnalogChannel (Channel: Longint): Longint;
```

I parametri da utilizzare sono:

*Channel*: Canale AD (1 o 2) che deve essere letto.

Il risultato ha il seguente significato:

*Longint*: Valore della conversione analogico/digitale.

Esempio:

```
var data: longint;
```

```
BEGIN
```

```
data := ReadAnalogChannel(1);
```

```
// Nella variabile data viene inserito il valore del convertitore AD numero 1
```

```
END;
```

#### **ReadAllAnalog**

Lo stato di entrambi i convertitori analogici/digitali viene letto a un insieme di interi lunghi. La sintassi è la seguente:

```
PROCEDURE ReadAllAnalog (var Data1, Data2: Longint);
```

I parametri da utilizzare sono:

*Data1, Data2*: Indicatori degli interi lunghi dove i dati verranno letti.

Esempio:

```
procedure TForm1.Button1Click(Sender: TObject);
```

```
var Data1, Data2: Longint;
```

```
begin
```

```
ReadAllAnalog(Data1, Data2); // Leggi i dati provenienti dalla scheda
```

```
Label1.caption:=inttostr(Data1); // Visualizza CH1
```

```
Label2.caption:=inttostr(Data2); // Visualizza CH2
```

```
end;
```

### **OutputAnalogChannel**

Consente di modificare il valore della tensione di uscita del DA specificato in funzione dei nuovi dati. Il valore 0 corrisponde alla minima tensione di uscita (0 Volt) mentre il valore 255 corrisponde alla massima tensione di uscita (+5V). Per ricavare il valore di 'Dati' necessario per ottenere una determinata tensione di uscita è possibile utilizzare la seguente formula:  $\text{Dati} / 255 \times 5V$ . Sintassi:

*PROCEDURE OutputAnalogChannel (Channel: Longint; Data: Longint);*

I parametri da utilizzare sono i seguenti:

*Channel*: Canale DA (1 e 2) da impostare.

*Data*: Valore compreso fra 0 e 255 che deve essere inviato al convertitore in questione.

Esempio:

*BEGIN*

*OutputAnalogChannel (1,127); // il canale DA n. 1 deve fornire una tensione di 2,5 V*

*END;*

### **OutputAllAnalog**

Entrambe le uscite dei due convertitori DA vengono modificate in funzione dei nuovi dati. Come nel caso precedente il valore 0 corrisponde alla minima tensione (0 Volt) mentre al valore 255 corrisponde la massima tensione di uscita (+5V). Per determinare i valori da utilizzare per ottenere specifiche tensioni di uscita può essere impiegata la seguente formula:  $\text{Dati} / 255 \times 5V$ .

La sintassi è la seguente:

*PROCEDURE OutputAllAnalog (Data1: Longint; Data2: Longint);*

I parametri sono i seguenti:

*Data1, Data2*: Valore compreso fra 0 e 255 da assegnare al relativo convertitore DA. Esempio:

*BEGIN*

*OutputAllAnalog (127, 255); // l'uscita del DA numero 1 è impostata per 2,5V, la numero 2 per 5 V*

*END;*

### **ClearAnalogChannel**

Consente di azzerare l'uscita del canale DA selezionato. Sintassi:

*PROCEDURE ClearAnalogChannel (Channel: Longint);*

Dove:

*Channel*: è il canale DA (1 o 2) da azzerare. Esempio:

*BEGIN*

*ClearAnalogChannel (1); // il canale DA numero 1 è posto a 0 V*

*END;*

### **ClearAllAnalog**

Con questa procedura entrambi i canali DA vengono impostati per la minima tensione di uscita (0 V). Sintassi:

*PROCEDURE ClearAllAnalog;*

Esempio:

*BEGIN*

*ClearAllAnalog; // Entrambi i canali DA vengono posti a 0 V*

*END;*

### **SetAnalogChannel**

L'uscita DA selezionata viene configurata per ottenere la massima tensione di uscita (+5V). Sintassi:

*PROCEDURE SetAnalogChannel (Channel: Longint);*

I parametri da impostare sono i seguenti:

*Channel*: Uscita DA 1 o 2.

Esempio:

*BEGIN*

*SetAnalogChannel(1); // L'uscita del canale DA n. 1 presenta il valore massimo (+ 5V)*

*END;*

### **SetAllAnalog**

Entrambe le uscite DA vengono settate per il massimo valore d'uscita (+5V). Sintassi:

*PROCEDURE SetAllAnalog;*

Esempio:

*BEGIN*

*SetAllAnalog; // Le uscite di entrambi i canali DA vengono poste a + 5V.*

*END;*

### **WriteAllDigital**

Lo stato delle uscite digitali viene aggiornato con lo stato dei corrispondenti bit nel parametro dei dati. Nel byte il valore più significativo (quello più a sinistra) corrisponde all'uscita 8, quello meno significativo all'uscita numero 1. Se il bit è posto a 1 l'uscita è attiva (alta), in caso contrario (0) l'uscita è a massa.

La sintassi è la seguente:

```
PROCEDURE WriteAllDigital (Data: Longint);
```

I parametri sono:

*Data*: Valore compreso fra 0 e 255 che viene inviato alla porta di uscita (8 canali).

Esempio:

```
BEGIN
```

```
WriteAllDigital(7);
```

```
// Le uscite 1,2,3 sono attive, le altre sono OFF (7 = 00000111)
```

```
END;
```

#### **ClearDigitalChannel**

Consente di mandare in OFF il canale selezionato. Sintassi:

```
PROCEDURE ClearDigitalChannel (Channel: Longint);
```

I parametri da utilizzare sono:

*Channel*: Valore compreso fra 1 e 8 che corrisponde al canale di uscita che deve essere azzerato. Esempio:

```
BEGIN
```

```
ClearIOchannel(4); // Poni in OFF l'uscita 4
```

```
END;
```

#### **ClearAllDigital**

Consente di porre in OFF tutte le uscite. Sintassi:

```
PROCEDURE ClearAllDigital;
```

Esempio:

```
BEGIN
```

```
ClearAllDigital; //Tutte le uscite vengono poste in OFF
```

```
END;
```

#### **SetDigitalChannel**

Consente di porre in ON il canale selezionato. Sintassi:

```
PROCEDURE SetDigitalChannel(Channel: Longint);
```

I parametri da utilizzare sono i seguenti:

*Channel*: Valore compreso fra 1 e 8 corrispondente al canale che deve essere posto in ON. Esempio:

```
BEGIN
```

```
SetDigitalChannel(1); // Poni in ON l'uscita 3
```

```
END;
```

#### **SetAllDigital**

Consente di porre in ON tutte le uscite digitali. Sintassi:

```
PROCEDURE SetAllDigital;
```

Esempio:

```
BEGIN
```

```
SetAllDigital; // Poni in ON tutte le uscite digitali
```

```
END;
```

#### **ReadDigitalChannel**

Legge lo stato dell'ingresso digitale selezionato.

Sintassi:

```
FUNCTION ReadDigitalChannel(Channel: Longint): Boolean;
```

I parametri da impostare sono i seguenti:

*Channel*: Valore compreso fra 1 e 5 corrispondente all'ingresso digitale da leggere.

Il risultato è il seguente:

VERO significa che l'ingresso è ON, FALSO significa che l'ingresso è OFF.

Esempio:

```
var status: boolean;
```

```
BEGIN
```

```
status := ReadIOchannel(2); // Leggi l'ingresso 2
```

```
END;
```

#### **ReadAllDigital**

Legge lo stato di tutti gli ingressi digitali. Sintassi:

*FUNCTION ReadAllDigital: Longint;*

Il risultato che si ottiene è il seguente:

*Longint*: i 5 bit meno significativi del dato rappresentano lo stato dei cinque ingressi (1 = ingresso alto, 0 = ingresso basso).

Esempio:

```
var status: longint;
```

```
BEGIN
```

```
status := ReadAllDigital; // Leggi tutti gli ingressi
```

```
END;
```

#### **ResetCounter**

Consente di resettare il contatore degli impulsi. Sintassi:

```
PROCEDURE ResetCounter (CounterNumber: Longint);
```

I parametri da utilizzare sono i seguenti:

*CounterNumber*: Valore 1 o 2, corrispondente al contatore da resettare. Esempio:

```
BEGIN
```

```
ResetCounter(2); // Resetta il contatore 2
```

```
END;
```

#### **ReadCounter**

La funzione legge lo stato del contatore di impulsi a 16 bit selezionato. Il contatore numero 1 conta gli impulsi che giungono all'ingresso I1 mentre il contatore numero 2 conta gli impulsi che giungono all'ingresso I2.

Sintassi:

```
FUNCTION ReadCounter(CounterNumber: Longint): Longint;
```

I parametri da utilizzare sono i seguenti:

*CounterNumber*: Valore (1 o 2) corrispondente al contatore che deve essere letto.

Il risultato è il seguente:

*Longint*: Rappresenta il contenuto del contatore di impulsi a 16 bit. Esempio:

```
var pulses: longint;
```

```
BEGIN
```

```
pulses := ReadCounter(2); // Leggi il contatore n.2
```

```
END;
```

#### **SetCounterDebounceTime**

Questa funzione riguarda la durata dell'antirimbato. Per consentire una precisa lettura degli impulsi, specie quando si utilizzano interruttori meccanici o relè, l'impulso deve rimanere stabile per almeno 2 ms (valore di default). Il tempo è uguale sia per il fronte di salita che per quello di discesa. Con il valore di default la velocità di conteggio massima è di 200 impulsi circa mentre con un valore di 0 ms la velocità massima di conteggio è di circa 2.000 impulsi. Con questa funzione è possibile impostare il tempo dell'antirimbato tra 0 e 5000 millisecondi. Sintassi:

```
PROCEDURE SetCounterDebounceTime(CounterNr, DebounceTime: Longint);
```

I parametri da inserire sono i seguenti:

*CounterNumber*: Valore (1 o 2) corrispondente al contatore da impostare.

*DebounceTime*: Tempo di antirimbato del contatore.

Esempio:

```
BEGIN
```

```
SetCounterDebounceTime(1,100);
```

```
// Il tempo dell'antirimbato relativo al contatore 1 è fissato in 100ms
```

```
END;
```

Di seguito presentiamo alcuni esempi su come realizzare i programmi applicativi e come utilizzare le due chiamate più importanti (OpenDevice e CloseDevice):

[In Delphi](#);

[In Visual Basic](#) (in questo caso bisogna verificare che il file K8055D.DLL venga copiato nella cartella Windows' SYSTEM32);.

[In Borland C++ Builder](#).

## SORGENTI

Presentiamo infine i sorgenti dei programmi proposti in modo da consentire una rapida modifica degli stessi, l'aggiunta di altre risorse, la modifica di alcune funzioni, eccetera.

[Sorgente in Delphi](#)

[Sorgente in Visual Basic](#)

[Sorgente in C++ Borland Builder](#)

---