

Introduzione alla DLL per la scheda d'interfaccia K8055 sperimentale USB

La scheda d'interfaccia K8055 ha 5 canali di ingresso digitali e 8 canali di uscita digitali. Vi sono inoltre 2 ingressi analogici, 2 uscite di tensione analogiche e 2 uscite PWM (Pulse Width Modulation) con una risoluzione di 8 bit. Il numero di ingressi/uscite può essere ulteriormente aumentato collegando più schede (fino a un massimo di 4) ai connettori USB del PC. Ad ogni scheda viene attribuito un numero d'identificazione per mezzo di due ponticelli, SK5 e SK6 (vedere tabella 1 qui sotto per la numerazione della scheda).

Tutte le routine di comunicazione sono contenute in una Dynamic Link Library (DLL) K8055D.DLL.

Questo documento descrive tutte le funzioni e procedure della DLL che sono disponibili per il programma applicativo. Richiamando le funzioni e procedure esportate dalla DLL, si potranno scrivere applicazioni Windows personalizzate (98SE, 2000, Me, XP) in Delphi, Visual Basic, C++ Builder o qualsiasi altro strumento di sviluppo di applicazioni Windows a 32 bit che supporta chiamate ad una DLL.

Segue una panoramica completa delle procedure e funzioni che sono esportate dalla K8055D.DLL. Alla fine di questo documento vi sono elenchi di esempi che permettono di acquisire familiarità su come costruire i propri programmi applicativi. Gli esempi sono scritti in Delphi, Visual Basic e C++ Builder. Negli elenchi vi sono dichiarazioni complete sulle funzioni e procedure DLL.

Notare che tutti gli esempi nella sezione descrittiva relativa alle funzioni e procedure sono scritti per Delphi.

SK5	SK6	INDIRIZZO SCHEDA
ON	ON	0
OFF	ON	1
ON	OFF	2
OFF	OFF	3

TABELLA 1: Parametri ponticelli SK5 e SK6

Nota: Questi parametri devono essere impostati prima che il cavo USB sia collegato alla scheda K8055 o prima di accendere il PC.

Panoramica delle procedure e funzioni della K8055D.DLL

Procedure generali

OpenDevice(CardAddress)
CloseDevice

Apre il collegamento al dispositivo K8055
Chiude il collegamento al dispositivo K8055

Procedure convertitore analogico/digitale

ReadAnalogChannel(Channelno)
ReadAllAnalog(Data1, Data2)

Legge lo stato di un canale di ingresso analogico
Legge lo stato di entrambi i canali di ingresso analogici

Procedure conversione digitale/analogica

OutputAnalogChannel(Channel, Data)

OutputAllAnalog(Data1, Data2)

ClearAnalogChannel(Channel)
ClearAllAnalog
SetAnalogChannel(Channel)
SetAllAnalog

Imposta il canale di uscita analogico in funzione dei dati
Imposta entrambi i canali di uscita analogici in funzione dei dati
Imposta il canale di uscita analogico al minimo
Imposta i canali di uscita analogici al minimo
Imposta il canale di uscita analogico al massimo
Imposta i canali di uscita analogici al massimo

Procedure uscita digitale

WriteAllDigital(Data)
ClearDigitalChannel(Channel)
ClearAllDigital
SetDigitalChannel(Channel)
SetAllDigital

Imposta le uscite digitali in funzione dei dati
Azzerata il canale di uscita
Azzerata tutti i canali di uscita
Imposta il canale digitale
Imposta tutti i canali digitali

Procedure e funzioni ingresso digitale

ReadDigitalChannel(Channel)
ReadAllDigital(Buffer)

Legge lo stato del canale di ingresso
Legge lo stato di tutti i canali di ingresso

Procedure e funzioni counter

ResetCounter(CounterNr)

ReadCounter(CounterNr)

SetCounterDebounceTime(CounterNr, DebounceTime)

Resetta il contatore di impulsi a 16 bit numero 1 o numero 2
Legge il contenuto del contatore di impulsi numero 1 o numero 2
Imposta il tempo di antirimbato del contatore di impulsi

Procedure e funzioni della K8055D.DLL

OpenDevice

Sintassi

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

Parametro

CardAddress: Valore compreso fra 0 e 3 che corrisponde all'impostazione dei ponticelli (SK5, SK6) sulla scheda K8055. Vedere tabella 1.

Risultato

Longint: In caso positivo, il valore di ritorno sarà l'indirizzo della scheda letto dalla macchina K8055.

Valore di ritorno -1 indica che la scheda K8055 non è stata trovata.

Descrizione

Apri il collegamento con la scheda K8055. Carica i driver necessari a comunicare tramite la porta USB. Questa procedura deve essere eseguita prima di ogni tentativo di comunicare con la scheda K8055.

Questa funzione si può anche utilizzare per selezionare la scheda K8055 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.

Esempio

```
var h: longint;  
BEGIN  
    h:=OpenDevice(0); // Opens the link to card number 0  
END;
```

CloseDevice

Sintassi

```
PROCEDURE CloseDevice;
```

Descrizione

Scarica le routine di comunicazione per la scheda K8055 e scarica il driver necessario a comunicare tramite la porta USB. Questa è l'ultima azione del programma applicativo prima dell'interruzione.

Esempio

```
BEGIN  
    CloseDevice; // The communication to the K8055 device is closed  
END;
```

ReadAnalogChannel

Sintassi

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

Parametro

Channel: Valore compreso fra 1 e 2 che corrisponde al canale AD il cui stato deve essere letto.

Risultato

Longint: Vengono letti i corrispondenti dati del convertitore analogico/digitale.

Descrizione

La tensione di ingresso del canale del convertitore analogico/digitale a 8 bit selezionato viene convertita in un valore compreso tra 0 e 255.

Esempio

```
var data: longint;  
BEGIN  
    data := ReadAnalogChannel(1);  
    // AD channel 1 is read to variable 'data'  
END;
```

ReadAllAnalog

Sintassi

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

Parametro

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

Descrizione

Lo stato di entrambi i convertitori analogici/digitali viene letto a un insieme di interi lunghi.

Esempio

```
procedure TForm1.Button1Click(Sender: TObject);  
var Data1, Data2: Longint;  
begin  
    ReadAllAnalog(Data1, Data2); // Read the data from the K8055  
    Label1.caption:=inttostr(Data1); // Display CH1 data  
    Label2.caption:=inttostr(Data2); // Display CH2 data  
end;
```

OutputAnalogChannel

Sintassi

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

Parametri

Channel: Valore compreso fra 1 e 2 che corrisponde al canale DA a 8 bit i cui dati devono essere impostati.

Data: Valore compreso fra 0 e 255 che deve essere inviato al convertitore digitale/analogico a 8 bit.

Descrizione

Il canale del convertitore digitale/analogico a 8 bit indicato viene modificato in funzione dei nuovi dati. Ciò significa che i dati corrispondono ad una tensione specifica. Il valore 0 corrisponde ad una tensione di uscita minima (0 Volt) e il valore 255 corrisponde ad una tensione di uscita massima (+5V). Un valore di 'Dati' compreso fra questi estremi può essere ottenuto con la seguente formula: Dati / 255

x 5V.

Esempio

```
BEGIN
  OutputAnalogChannel (1,127);
  // DA channel 1 is set to 2.5V
END;
```

OutputAllAnalog

Sintassi

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

Parametri

Data1, Data2: Valore compreso fra 0 e 255 che deve essere inviato al convertitore digitale/analogico a 8 bit.

Descrizione

Entrambi i canali del convertitore digitale/analogico a 8 bit vengono modificati in funzione dei nuovi dati. Ciò significa che i dati corrispondono ad una tensione specifica. Il valore 0 corrisponde ad una tensione di uscita minima (0 Volt) e il valore 255 corrisponde ad una tensione di uscita massima (+5V). Un valore di 'Dati1' o 'Dati2' compreso fra questi estremi può essere ottenuto con la seguente formula:
 $\text{Dati} / 255 \times 5V$.

Esempio

```
BEGIN
  OutputAllAnalog(127, 255);
  // DA channel 1 is set to 2.5V and channel 2 is set to 5V
END;
```

ClearAnalogChannel

Sintassi

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

Parametro

Channel: Valore compreso fra 1 e 2 che corrisponde al numero di canale DA a 8 bit dove i dati devono essere cancellati.

Descrizione

Il canale DA selezionato è impostato sulla tensione di uscita minima (0 Volt).

Esempio

```
BEGIN
  ClearAnalogChannel (1); // DA channel 1 is set to 0V
END;
```

ClearAllAnalog

Sintassi

```
PROCEDURE ClearAllAnalog;
```

Descrizione

Entrambi i canali DA sono impostati sulla tensione di uscita minima (0 Volt).

Esempio

```
BEGIN  
  ClearAllAnalog; // All DA channels 1 and 2 are set to 0V  
END;
```

SetAnalogChannel

Sintassi

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

Parametro

Channel: Valore compreso fra 1 e 2 che corrisponde al numero di canale DA a 8 bit dove i dati devono essere impostati sul valore massimo.

Descrizione

Il canale del convertitore digitale/analogico a 8 bit è impostato sulla tensione di uscita massima.

Esempio 15

```
BEGIN  
  SetAnalogChannel(1); // DA channel 1 is set to +5V  
END;
```

SetAllAnalog

Sintassi

```
PROCEDURE SetAllAnalog;
```

Descrizione

Tutti i canali dei convertitori digitali/analogici a 8 bit sono impostati sulla tensione di uscita massima.

Esempio

```
BEGIN  
  SetAllAnalog; // DA channels 1 and 2 are set to +5V  
END;
```

WriteAllDigital

Sintassi

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

Parametro

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

Descrizione

I canali della porta di uscita digitale sono aggiornati con lo stato dei corrispondenti bit nel parametro dei dati. Un livello alto (1) significa che è impostata l'uscita IC1 del microcontroller e un livello basso (0) significa che l'uscita è azzerata.

Esempio

```
BEGIN
  WriteAllDigital(7);
  // Output channels 1...3 are on, output channels 4...8 are off
END;
```

ClearDigitalChannel

Sintassi

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

Parametro

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

Descrizione

Il canale selezionato è azzerato.

Esempio

```
BEGIN
  ClearIOchannel(4); // Digital output channel 4 is OFF
END;
```

ClearAllDigital

Sintassi

```
PROCEDURE ClearAllDigital;
```

Risultato

Tutte le uscite digitali sono azzerate.

Esempio

```
BEGIN
  ClearAllDigital; // All Output channels 1 to 8 are OFF
END;
```

SetDigitalChannel

Sintassi

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

Parametro

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

Descrizione

Il canale di uscita digitale selezionato è impostato.

Esempio

```
BEGIN
  SetDigitalChannel(1); // Digital output channel 3 is ON
END;
```

SetAllDigital

Sintassi

```
PROCEDURE SetAllDigital;
```

Descrizione

Tutti i canali di uscita digitali sono impostati.

Esempio

```
BEGIN
  SetAllDigital; // All Output channels are ON
END;
```

ReadDigitalChannel

Sintassi

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

Parametro

Channel: Valore compreso fra 1 e 5 che corrisponde al canale di uscita il cui stato deve essere letto.

Risultato

Boolean: VERO significa che il canale è stato impostato e FALSO significa che è stato azzerato.

Descrizione

Lo stato del canale di ingresso selezionato viene letto.

Esempio

```
var status: boolean;
BEGIN
  status := ReadIOchannel(2); // Read Input channel 2
END;
```

ReadAllDigital

Sintassi

```
FUNCTION ReadAllDigital: Longint;
```

Risultato

Longint: I 5 LSB corrispondono allo stato dei canali di uscita. Un valore elevato (1) significa che il canale è ALTO, un valore basso (0) significa che il canale è BASSO.

Descrizione

La funzione rinvia allo stato degli ingressi digitali.

Esempio

```
var status: longint;  
BEGIN  
    status := ReadAllDigital; // Read the Input channels  
END;
```

ResetCounter

Sintassi

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

Parametro

CounterNumber: Valore 1 o 2, che corrisponde al contatore da resettare.

Descrizione

Il contatore di impulsi selezionato è resettato.

Esempio

```
BEGIN  
    ResetCounter(2); // Reset the counter number 2  
END;
```

ReadCounter

Sintassi

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

Parametro

CounterNumber: Valore 1 o 2, che corrisponde al contatore che deve essere letto.

Risultato

Longint: Il contenuto del contatore di impulsi a 16 bit.

Descrizione

La funzione rinvia allo stato del contatore di impulsi a 16 bit selezionato.

Il contatore numero 1 conta gli impulsi inviati all'ingresso I1 e il contatore numero 2 conta gli impulsi inviati all'ingresso I2.

Esempio

```
var pulses: longint;  
BEGIN  
  pulses := ReadCounter(2); // Read the counter number 2  
END;
```

SetCounterDebounceTime

Sintassi

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

Parametro

CounterNumber: Valore 1 o 2, che corrisponde al contatore da impostare.

DebounceTime: Tempo di antirimbato del contatore di impulsi.

Il tempo di antirimbato corrisponde al tempo in millisecondi (ms) da impostare per il contatore di impulsi. Il valore può essere compreso fra 0 e 5000.

Descrizione

Nel software, gli ingressi del contatore eliminano i rimbalzi, allo scopo di evitare scatti intempestivi quando si utilizzano interruttori meccanici o ingressi a relè. Il tempo di antirimbato è uguale per entrambi i fronti discendente e ascendente. Il tempo di antirimbato predefinito è di 2ms. Ciò significa che l'ingresso del contatore deve essere stabile per almeno 2ms prima che venga riconosciuto, fornendo la velocità di conteggio massima di 200 conteggi al secondo.

Se il tempo di antirimbato è impostato su 0, allora la velocità di conteggio massima è di circa 2000 conteggi al secondo.

Esempio

```
BEGIN  
  SetCounterDebounceTime(1,100);  
  // The debounce time for counter number 1 is set to 100ms  
END;
```

Utilizzare la K8055D.DLL in Delphi

In questo esempio, vi sono le dichiarazioni delle procedure e funzioni K8055D.DLL e un esempio su come utilizzare le due chiamate di funzione DLL più importanti: `OpenDevice` e `CloseDevice`.

```

unit K8055;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, ExtCtrls, ComCtrls;

type
  TForm1 = class(TForm)
    GroupBox1: TGroupBox;
    SK6: TCheckBox;
    SK5: TCheckBox;
    Button1: TButton;
    Label1: TLabel;
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure Button1Click(Sender: TObject);

  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;
  timed:boolean;

implementation

{$R *.DFM}
function OpenDevice(CardAddress: Longint): Longint; stdcall; external 'K8055d.dll';
procedure CloseDevice; stdcall; external 'K8055d.dll';
function ReadAnalogChannel(Channel: Longint):Longint; stdcall; external 'K8055d.dll';
procedure ReadAllAnalog(var Data1, Data2: Longint); stdcall; external 'K8055d.dll';
procedure OutputAnalogChannel(Channel: Longint; Data: Longint); stdcall; external
'K8055d.dll';
procedure OutputAllAnalog(Data1: Longint; Data2: Longint); stdcall; external 'K8055d.dll';
procedure ClearAnalogChannel(Channel: Longint); stdcall; external 'K8055d.dll';
procedure ClearAllAnalog; stdcall; external 'K8055d.dll';
procedure SetAnalogChannel(Channel: Longint); stdcall; external 'K8055d.dll';
procedure SetAllAnalog; stdcall; external 'K8055d.dll';
procedure WriteAllDigital(Data: Longint);stdcall; external 'K8055d.dll';
procedure ClearDigitalChannel(Channel: Longint); stdcall; external 'K8055d.dll';
procedure ClearAllDigital; stdcall; external 'K8055d.dll';
procedure SetDigitalChannel(Channel: Longint); stdcall; external 'K8055d.dll';
procedure SetAllDigital; stdcall; external 'K8055d.dll';
function ReadDigitalChannel(Channel: Longint): Boolean; stdcall; external 'K8055d.dll';
function ReadAllDigital: Longint; stdcall; external 'K8055d.dll';
function ReadCounter(CounterNr: Longint): Longint; stdcall; external 'K8055d.dll';
procedure ResetCounter(CounterNr: Longint); stdcall; external 'K8055d.dll';
procedure SetCounterDebounceTime(CounterNr, DebounceTime:Longint); stdcall; external
'K8055d.dll';

procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  CloseDevice;
end;

procedure TForm1.Button1Click(Sender: TObject);
var h,CardAddr:longint;
begin
  CardAddr:= 3-(integer(SK5.Checked) + integer(SK6.Checked) * 2);
  h:= OpenDevice(CardAddr);
  case h of
    0..3: label12.caption:='Card '+ inttostr(h)+' connected';
  end;
end;

```

```
-1: labell12.caption:='Card '+ inttostr(CardAddr)+' not found';
end;
end;

end.
```

Utilizzare la K8055D.DLL in Visual Basic

In questo esempio, vi sono le dichiarazioni delle procedure e funzioni K8055D.DLL e un esempio su come utilizzare le due chiamate di funzione DLL più importanti: **OpenDevice** e **CloseDevice**.

Nota: Verificare che il file K8055D.DLL venga copiato nella cartella Windows' SYSTEM32:

```
Option Explicit
Private Declare Function OpenDevice Lib "k8055d.dll" (ByVal CardAddress As Long) As Long
Private Declare Sub CloseDevice Lib "k8055d.dll" ()
Private Declare Function ReadAnalogChannel Lib "k8055d.dll" (ByVal Channel As Long) As Long
Private Declare Sub ReadAllAnalog Lib "k8055d.dll" (Data1 As Long, Data2 As Long)
Private Declare Sub OutputAnalogChannel Lib "k8055d.dll" (ByVal Channel As Long, ByVal Data As Long)
Private Declare Sub OutputAllAnalog Lib "k8055d.dll" (ByVal Data1 As Long, ByVal Data2 As Long)
Private Declare Sub ClearAnalogChannel Lib "k8055d.dll" (ByVal Channel As Long)
Private Declare Sub SetAllAnalog Lib "k8055d.dll" ()
Private Declare Sub ClearAllAnalog Lib "k8055d.dll" ()
Private Declare Sub SetAnalogChannel Lib "k8055d.dll" (ByVal Channel As Long)
Private Declare Sub WriteAllDigital Lib "k8055d.dll" (ByVal Data As Long)
Private Declare Sub ClearDigitalChannel Lib "k8055d.dll" (ByVal Channel As Long)
Private Declare Sub ClearAllDigital Lib "k8055d.dll" ()
Private Declare Sub SetDigitalChannel Lib "k8055d.dll" (ByVal Channel As Long)
Private Declare Sub SetAllDigital Lib "k8055d.dll" ()
Private Declare Function ReadDigitalChannel Lib "k8055d.dll" (ByVal Channel As Long) As Boolean
Private Declare Function ReadAllDigital Lib "k8055d.dll" () As Long
Private Declare Function ReadCounter Lib "k8055d.dll" (ByVal CounterNr As Long) As Long
Private Declare Sub ResetCounter Lib "k8055d.dll" (ByVal CounterNr As Long)
Private Declare Sub SetCounterDebounceTime Lib "k8055d.dll" (ByVal CounterNr As Long, ByVal DebounceTime As Long)

Private Sub Connect_Click()
    Dim CardAddress As Long
    Dim h As Long
    CardAddress = 0
    CardAddress = 3 - (Check1(0).Value + Check1(1).Value * 2)
    h = OpenDevice(CardAddress)
    Select Case h
        Case 0, 1, 2, 3
            Labell1.Caption = "Card " + Str(h) + " connected"
        Case -1
            Labell1.Caption = "Card " + Str(CardAddress) + " not found"
    End Select
End Sub

Private Sub Form_Terminate()
    CloseDevice
End Sub
```

Utilizzare la K8055D.DLL in Borland C++ Builder

Qui sotto è riportato un elenco della K8055D.h che include le dichiarazioni delle procedure e funzioni K8055D.DLL. L'esempio mostra come utilizzare le due chiamate di funzione DLL più importanti: **OpenDevice** e **CloseDevice**.

```

//Listing K8055D.h
#ifdef __cplusplus
extern "C" {
#endif

#define FUNCTION __declspec(dllimport)

FUNCTION long __stdcall OpenDevice(long CardAddress);
FUNCTION __stdcall CloseDevice();
FUNCTION long __stdcall ReadAnalogChannel(long Channel);
FUNCTION __stdcall ReadAllAnalog(long *Data1, long *Data2);
FUNCTION __stdcall OutputAnalogChannel(long Channel, long Data);
FUNCTION __stdcall OutputAllAnalog(long Data1, long Data2);
FUNCTION __stdcall ClearAnalogChannel(long Channel);
FUNCTION __stdcall ClearAllAnalog();
FUNCTION __stdcall SetAnalogChannel(long Channel);
FUNCTION __stdcall SetAllAnalog();
FUNCTION __stdcall WriteAllDigital(long Data);
FUNCTION __stdcall ClearDigitalChannel(long Channel);
FUNCTION __stdcall ClearAllDigital();
FUNCTION __stdcall SetDigitalChannel(long Channel);
FUNCTION __stdcall SetAllDigital();
FUNCTION bool __stdcall ReadDigitalChannel(long Channel);
FUNCTION long __stdcall ReadAllDigital();
FUNCTION long __stdcall ReadCounter(long CounterNr);
FUNCTION __stdcall ResetCounter(long CounterNr);
FUNCTION __stdcall SetCounterDebounceTime(long CounterNr, long DebounceTime);

#ifdef __cplusplus
}
#endif

```

```

//Listing Unit1.cpp

```

```

//-----
#include <vcl.h>
#pragma hdrstop

#include "Unit1.h"
#include "K8055D.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//-----

__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
}
//-----

void __fastcall TForm1::Connect1Click(TObject *Sender)
{
    int CardAddr = 3 - (int(CheckBox1->Checked) + int(CheckBox2->Checked) * 2);
    int h = OpenDevice(CardAddr);
    switch (h) {
        case 0 :
        case 1 :
        case 2 :
        case 3 :
            Label1->Caption = "Card " + IntToStr(h) + " connected";
            break;
        case -1 :
            Label1->Caption = "Card " + IntToStr(CardAddr) + " not found";
    }
}
//-----

void __fastcall TForm1::FormClose(TObject *Sender, TCloseAction &Action)
{
}

```

```
CloseDevice();
}
//-----

#include <vcl.h>
#pragma hdrstop

#include "Unit1.h"
#include "K8055D.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//-----

__fastcall TForm1::TForm1(TComponent* Owner)
: TForm(Owner)
{
}
//-----

void __fastcall TForm1::Connect1Click(TObject *Sender)
{
    int CardAddr = 3 - (int)(CheckBox1->Checked) + int(CheckBox2->Checked) * 2;
    int h = OpenDevice(CardAddr);
    switch (h) {
        case 0 :
        case 1 :
        case 2 :
        case 3 :
            Label1->Caption = "Card " + IntToStr(h) + " connected";
            break;
        case -1 :
            Label1->Caption = "Card " + IntToStr(CardAddr) + " not found";
    }
}
//-----

void __fastcall TForm1::FormClose(TObject *Sender, TCloseAction &Action)
{
    CloseDevice;
}
//-----
```