

K8055 USB testi- ja liityntäkortin DLL-kirjastofunktiot

Tässä ohjeessa on yksityiskohtainen kuvaus K8055D.DLL-kirjaston funktioista ja proseduureista, joita voidaan kutsua sovellusohjelmasta.

Näitä funktio- ja proseduurikutsuja käyttäen voidaan kortille kirjoittaa Windows (98SE, 2000, Me, XP) sovelluksia Delphi, Visual Basic, C++ Builder tai muilla 32-bittisillä ohjelmointikielillä, jotka tukevat DLL-funktiokutsuja.

K8055-kortilla on viisi digitaalituloa ja kahdeksan digitaalilähtöä. Lisäksi kortilla on kaksi analogiatuloa, kaksi analogista jännitelähtöä ja kaksi pulssinleveysmodulaattorilähtöä (PWM). Kaikki analogiatulot ja -lähdöt ovat 8-bittisiä. Liityntöjen määrää voidaan kasvattaa kytkemällä useampia kortteja PC:n USB liittimiin. Maksimissaan voidaan samaan koneeseen liittää neljä korttia, joilla kullakin on oma osite. Osoite asetetaan oikosulkupalloilla SK5 ja SK6 (ks. Taulukko 1).

Kaikki kommunikointi kortin ja PC:n välillä tapahtuu kirjaston K8055D.DLL välityksellä.

Tämän ohjeen lopussa on listauksia esimerkkiohjelmista, joita voit käyttää lähtökohtana omien sovellusten kirjoittamiseen.

Esimerkkiohjelmat on kirjoitettu Delphi, Visual Basic and C++ Builder -kielillä.

Ohjelmalistauksissa on kaikkien kirjastofunktioiden ja -proseduurien prototyypin esittelyt.

Huom. Tämän ohjeen kaikki muut esimerkkilistaukset on kirjoitettu Delphillä.

SK5	SK6	KORTIN OSOITE
ON	ON	0
POIS	ON	1
ON	POID	2
POIS	POIS	3

Taulukko 1: Oikosulkupalojen SK5, SK6 asetus

Huom: Oikosulkupalojen asetukset pitää tehdä ennen kortin kytkemistä PC:n USB-liittimeen tai ennen PC:n virran kytkemistä.

Yhteenveto K8055D.DLL-kirjaston funktioista ja proseduureista

Yleiset proseduurit

OpenDevice(CardAddress)	<i>Avaa kommunikointiyhteyden K8055 kortille</i>
CloseDevice	<i>Sulkee kommunikointiyhteyden K8055 kortille</i>

A/D-muunnostulosten luenta

ReadAnalogChannel(Channelno)	<i>Lukee yhden analogiatulon tilan</i>
ReadAllAnalog(Data1, Data2)	<i>Lukee molempien analogiatulojen tilat</i>

D/A-muunnoksen ohjaus

OutputAnalogChannel(Channel, Data)	<i>Asettaa yhden analogialähdön tilan</i>
OutputAllAnalog(Data1, Data2)	<i>Asettaa molempien analogialähtöjen tilat</i>
ClearAnalogChannel(Channel)	<i>Asettaa yhden analogialähdön tilan pienimpään arvoon</i>
ClearAllAnalog	<i>Asettaa molemmat analogialähdöt pienimpään arvoon</i>
SetAnalogChannel(Channel)	<i>Asettaa yhden analogialähdön tilan suurimpaan arvoon</i>
SetAllAnalog	<i>Asettaa molemmat analogialähdöt suurimpaan arvoon</i>

Digitaalilähtöjen ohjaus

WriteAllDigital(Data)	<i>Asettaa digitaalilähdöt Data-arvon mukaiseen tilaan</i>
ClearDigitalChannel(Channel)	<i>Asettaa halutun digitaalilähdön alas</i>
ClearAllDigital	<i>Asettaa kaikki digitaalilähdöt alas</i>
SetDigitalChannel(Channel)	<i>Asettaa yhden digitaalilähdön ylös</i>
SetAllDigital	<i>Asettaa kaikki digitaalilähdöt ylös</i>

Digitaalitulojen luenta

ReadDigitalChannel(Channel)	<i>Lukee yhden digitaalitulon tilan</i>
ReadAllDigital(Buffer)	<i>Lukee kaikkien digitaalitulojen tilat</i>

Laskurin luenta ja ohjaus

ResetCounter(CounterNr)	<i>Nollaa 16-bittisen pulssilaskurin 1 tai 2</i>
ReadCounter(CounterNr)	<i>Lukee pulssilaskurin 1 tai 2 tilan</i>
SetCounterDebounceTime(CounterNr, DebounceTime)	<i>Asettaa pulssilaskurin kosketinvärähtelyjen estoajan pituuden</i>

DLL-proseduurien ja -funktioiden käyttö

OpenDevice

Syntaksi

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

Parametri

CardAddress: Kortin osoite 0 ... 3. Valitaan oikosulkupaloilla SK5 ja SK6 (ks. Taulukko 1).

Tulos

Longint: Jos funktion toteutus onnistui, niin funktio palauttaa K8055 kortilta luetun osoitteen.

Funktio palauttaa -1 jos osoitteella haettua korttia ei löytynyt.

Kuvaus

Funktio avaa kommunikaatioyhteyden K8055 kortin ja PC:n välille ja lataa tarvittavat USB-ajurit. Tämä funktio pitää suorittaa sovellusohjelmassa alussa, ennen muita K8055-kortin ohjaus- ja luentayrityksiä.

Tällä funktiolla voidaan valita aktiivinen K8055 kortti. Kaikki kommunikaatio tämän funktiokutsun jälkeen tapahtuu valitun kortin kanssa.

Esimerkki

```
var h: longint;  
BEGIN  
    h:=OpenDevice(0); // Avaa yhteyden kortille 0  
END;
```

CloseDevice

Syntaksi

```
PROCEDURE CloseDevice;
```

Kuvaus

Sulkee kommunikointiyhteyden K8055-kortille ja vapauttaa USB-ajurin. Tämä proseduri tulee suorittaa ennen sovellusohjelman lopetusta.

Esimerkki

```
BEGIN  
    CloseDevice; // Sulkee kommunikatioyhteyden K8055-kortille  
END;
```

ReadAnalogChannel

Syntaksi

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

Parametri

Channel: Sen A/D-tulokanavan numero (1 tai 2), jonka tila halutaan lukea.

Tulos

Longint: A/D-muuntimen tila.

Kuvaus

Analogiatulon 8-bittinen muunnostulos (0 ... 255).

Esimerkki

```
var data: longint;  
BEGIN  
    data := ReadAnalogChannel(1);  
    // AD-kanava 1 tila luetaan muuttuun 'data'  
END;
```

ReadAllAnalog

Syntaksi

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

Parametri

Data1, Data2: Osoittimet 32 bittisiin muuttujiin, joihin A/D-muuntimien tilat luetaan.

Kuvaus

Kummankin A/D-muuntimen tilat luetaan 32-bittisiin muuttujiin.

Esimerkki

```
procedure TForm1.Button1Click(Sender: TObject);  
var Data1, Data2: Longint;  
begin  
    ReadAllAnalog(Data1, Data2); // Luetaan A/D-muuntimien tilat  
    Label1.caption:=inttostr(Data1); // Näytetään kanavan 1 tila  
    Label2.caption:=inttostr(Data2); // Näytetään kanavan 2 tila  
end;
```

OutputAnalogChannel

Syntaksi

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

Parametri

Channel: Sen D/A-lähtökanavan numero (1 tai 2), jonka tila halutaan asettaa.

Data: Luku (0 ... 255), joka asetetaan D/A-muuntimelle.

Kuvaus

D/A-muuntimen data päivitetään. Analogialähdöstä saadaan lukua vastaava jännite. Lukua 0 vastaa pienin lähtöjännite (0 V) ja lukua 255 vastaa suurin lähtöjännite (n. +5 V).

Lähtöjännitteen kaava: $V_{out} = 5V * Data/255$

Esimerkki

```
BEGIN  
    OutputAnalogChannel (1,127);  
    // DA-kanavan 1 jännitteeksi asetetaan 2.5V  
END;
```

OutputAllAnalog

Syntaksi

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

Parametrit

Data1, Data2: Luku (0 ... 255), joka lähetetään 8-bittiselle D/A-muuntimelle.

Kuvaus

Molempien D/A-muuntimien datat päivitetään. Analogialähdöistä saadaan lukuja vastaavat jännitteet. Lukua 0 vastaa pienin lähtöjännite (0 V) ja lukua 255 vastaa suurin lähtöjännite (n. +5 V).
Lähtöjännitteen kaava: $V_{out} = 5V * Data_n / 255$

Esimerkki

```
BEGIN
  OutputAllAnalog(127, 255);
  // DA-kanavaan 1 asetetaan +2.5V ja DA-kanavaan 2 asetetaan +5V
END;
```

ClearAnalogChannel

Syntaksi

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

Parametri

Channel: Sen D/A-lähtökanavan numero (1 tai 2), jonka tila halutaan asettaa.

Kuvaus

Valitun analogialähdön tila asetetaan pienimpään arvoon (0 V).

Esimerkki

```
BEGIN
  ClearAnalogChannel (1); // DA-kanavaan 1 asetetaan 0V
END;
```

ClearAllAnalog

Syntaksi

```
PROCEDURE ClearAllAnalog;
```

Kuvaus

Molempien analogialähtöjen tilat asetetaan pienimpään arvoon (0 V).

Esimerkki

```
BEGIN
  ClearAllAnalog; // DA-kanavat 1 ja 2 asetattaan 0V:iin
```

END;

SetAnalogChannel

Syntaksi

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

Parametri

Channel: Sen D/A-lähtökanavan numero (1 tai 2), jonka tila halutaan asettaa.

Kuvaus

Valitun analogialähdön tila asetetaan suurimpaan arvoon (n. +5 V).

Esimerkki

```
BEGIN
  SetAnalogChannel (1); // DA-kanavaan 1 asetetaan +5V
END;
```

SetAllAnalog

Syntaksi

```
PROCEDURE SetAllAnalog;
```

Kuvaus

Molempien analogialähtöjen tilat asetetaan suurimpaan arvoon (n. +5 V).

Esimerkki

```
BEGIN
  SetAllAnalog; // DA-kanavat 1 ja 2 asetetaan +5V:iin
END;
```

WriteAllDigital

Syntaksi

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

Parametri

Data: Luku alueella 0 ... 255, joka lähetetään digitaalilähtöjen (8 kanavaa) tilaksi.

Kuvaus

Asettaa digitaalilähdöt (8 kanavaa) Data-arvon mukaiseen tilaan (0 ... 255). Jos bitin tila on 1, mikrokontrollerin vastaava lähtö menee ylös, jos tila on 0, niin mikrokontrollerin lähtö menee alas.

Esimerkki

```
BEGIN
  WriteAllDigital(7);
  // Digitaalilähdöt 1...3 ovat ylhäällä ja lähdöt 4...8 ovat alhaalla
END;
```

ClearDigitalChannel

Syntaksi

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

Parametri

Channel: Luku (1 ... 8), joka vastaa asetettavan digitaalilähdön numeroa.

Kuvaus

Valittu mikrokontrollerin lähtö asetetaan alas.

Esimerkki

```
BEGIN
    ClearIOchannel(4); // Digitaalilähtö 4 asetetaan alas
END;
```

ClearAllDigital

Syntaksi

```
PROCEDURE ClearAllDigital;
```

Tulos

Kaikki mikrokontrollerin digitaalilähdöt (1 ... 8) asetetaan alas.

Esimerkki

```
BEGIN
    ClearAllDigital; // Kaikki lähdöt alas
END;
```

SetDigitalChannel

Syntaksi

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

Parametri

Channel: Luku (1 ... 8), joka vastaa asetettavan digitaalilähdön numeroa

Kuvaus

Valittu mikrokontrollerin lähtö asetetaan ylös.

Esimerkki

```
BEGIN
    SetDigitalChannel(1); // Digitaalilähtö 1 asetetaan ylös
END;
```

SetAllDigital

Syntaksi

```
PROCEDURE SetAllDigital;
```

Kuvaus

Kaikki mikrokontrollerin digitaalilähdöt (1 ... 8) asetetaan ylös.

Esimerkki

```
BEGIN
  SetAllDigital; // Kaikki lähdöt ylös
END;
```

ReadDigitalChannel

Syntaksi

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

Parametri

Channel: Luku (1 ... 5), joka vastaa luettavan digitaalitulon numeroa.

Tulos

Boolean: TRUE = mikrokontrollerin tulo on ylhäällä, FALSE = mikrokontrollerin tulo on alhaalla.

Kuvaus

Luetaan valitun kanavan tila.

Esimerkki

```
var status: boolean;
BEGIN
  status := ReadIOchannel(2); // Luetaan digitaalisen tulokanavan 2 tila
END;
```

ReadAllDigital

Syntaksi

```
FUNCTION ReadAllDigital: Longint;
```

Tulos

Longint: Luvun viisi alinta bittiä vastaavat digitaalitulojen tiloja. 1 = mikrokontrollerin tulo on ylhäällä, 0 = mikrokontrollerin tulo on alhaalla.

Kuvaus

Luetaan kaikkien digitaalitulojen tilat.

Esimerkki

```
var status: longint;
BEGIN
  status := ReadAllDigital; // Luetaan digitaalitulojen tilat muuttujaan
```

```
'status'  
  END;
```

ResetCounter

Syntaksi

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

Parametri

CounterNumber: Pulssilaskurin numero (1 tai 2), joka halutaan nollata.

Kuvaus

Nollataan valittu pulssilaskuri.

Esimerkki

```
  BEGIN  
    ResetCounter(2); // Pulssilaskuri 2 nollataan  
  END;
```

ReadCounter

Syntaksi

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

Parametri

CounterNumber: Luettavan pulssilaskurin numero (1 tai 2).

Tulos

Longint: Pulssilaskurin 16-bittinen sisältö.

Kuvaus

Funktio palauttaa 16-bittisen pulssilaskurin sisällön.

Laskuri 1 laskee digitaalitulon I1 tuotuja pulsseja ja laskuri 2 laskee tulon I2 tuotuja pulsseja.

Esimerkki

```
  var pulses: longint;  
  BEGIN  
    pulses := ReadCounter(1); // Luetaan pulssilaskurin 1 tila  
  END;
```

SetCounterDebounceTime

Syntaksi

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

Parametri

CounterNumber: Pulssilaskurin numero (1 tai 2), jonka kosketinvärähtelyjen estoajan pituus asetetaan. DebounceTime: Kosketinvärähtelyjen estoajan pituus millisekunneina (ms). Asettava aika

voi olla välillä 0 ... 5000.

Kuvaus

Pulssilaskurit on varustettu mekaanisten kytkinten ja releiden kosketinvärähtelyjen estolla. Estoaika on samansuuruinen pulssin nousevalle ja laskevalle reunalle. Oletusarvo on 2 ms. Tämä tarkoittaa sitä, että tulonastan tilan on oltava vakaana 2 ms ajan ennen kuin tilanvaihto tunnistetaan tapahtuneeksi.

Oletusasetuksella suurin laskentataajuus on n. 200 Hz.

Kun estojaksi asetetaan 0 ms, niin suurin laskentataajuus on n. 2000 Hz.

Esimerkki

```
BEGIN
  SetCounterDebounceTime(1,100);
  // Laskurin 1 kosketinvärähtelyjen estojaksi asetetaan 100ms
END;
```

K8055D.DLL kirjastofunktioiden käyttö Delphi-ohjelmoinnissa.

Tässä esimerkissä on kaikkien K8055D.DLL-kirjastofunktioiden esittely ja esimerkki siitä, miten tärkeimpiä funktiokutsuja `OpenDevice` ja `CloseDevice` käytetään.

```
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: labell2.caption:='Card '+ inttostr(h)+' connected';
-1: labell2.caption:='Card '+ inttostr(CardAddr)+' not found';
end;
end;

end.

```

K8055D.DLL kirjastofunktioiden käyttö Visual Basic -ohjelmoinnissa.

Tässä esimerkissä on kaikkien K8055D.DLL-kirjastofunktioiden esittely ja esimerkki siitä, miten tärkeimpiä funktiokutsuja `OpenDevice` ja `CloseDevice` käytetään.

Huom. Varmista, että K8055D.DLL-tiedosto on kopioitu Windowsin SYSTEM32-hakemistoon.

```

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

```

K8055D.DLL kirjastofunktioiden käyttö Borland C++ Builder -ohjelmoinnissa.

Tässä esimerkissä on kaikkien K8055D.DLL-kirjastofunktioiden esittely ja esimerkki siitä, miten tärkeimpiä funktiokutsuja `OpenDevice` ja `CloseDevice` käytetään.

```
//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
#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();  
}  
//-----
```