

Introducción a la DLL para la tarjeta interface USB de experimentación K8055

Para K8055.DLL o K8055D_C.DLL, versión 2.0x y superior

La tarjeta interface K8055 está equipada de 5 canales de entrada digitales y de 8 canales de salida digitales. Además, está provista de 2 entradas analógicas y de 2 salidas de tensión analógicas y 2 salidas PWM (Pulse Width Modulation) con una resolución de 8 bits. Es posible aumentar el número de entradas y de salidas conectando más tarjetas (máx. 4) a los conectores USB del ordenador. Cada tarjeta tiene su propio número de identificación que se determina con 2 cables de puente, SK5 y SK6 (véase abajo, tabla 1 para la numeración de las tarjetas).ring)

Todas las rutinas de comunicación están agrupadas en una DLL (Dynamic Link Library - biblioteca de enlaces dinámicos) K8055D.DLL.

Este documento describe todas las funciones y procedimientos de DLL que están disponibles vía su programa de aplicación. Llamando las funciones y los procedimientos de la DLL, puede escribir su propias aplicaciones en Windows (98SE, 2000, Me, XP), Delphi, Visual Basic, C++ Builder o cualquier otra herramienta de desarrollo con 32 bits que soporta llamadas a una DLL

Si quiere utilizar Visual C++, necesitará ficheros especiales K8055D_C.DLL y K8055D_C.LIB, incluso en el proyecto VC++.

Todas las funciones y los procedimientos soportados del K8055D.DLL y K8055D_C.DLL son idénticos. La única diferencia se sitúa en el convenio de llamada para los usuarios de C++.

A continuación, Ud. encuentra un resumen completo de los procedimientos y funciones exportados por la K8055D.DLL. Al final de este documento encontrará unos listados de programas ejemplos para darse una idea cómo puede hacer sus propios programas de aplicación. Los ejemplos están escritos en Delphi, Visual Basic y C++ Builder.

Los listados contienen explicaciones completas para las funciones y procedimientos DLL.

Observe que todos los ejemplos en la parte de descripciones de funciones y procedimientos están escritos para Delphi.

| SK5 | SK6 | DIRECCIÓN DE LA TARJETA |
|-----|-----|-------------------------|
| ON | ON | 0 |
| OFF | ON | 1 |
| ON | OFF | 2 |
| OFF | OFF | 3 |

TABLA 1: Ajustes de los cables de puente SK5, SK6

Observación: Ajuste los cables de puente antes de conectar el cable USB a la tarjeta K8055 o antes de encender el ordenador.

* VM110 es la versión montada de la K8055.

Resumen de los procedimientos y de las funciones de la K8055D.DLL

Procedimientos generales

OpenDevice(CardAddress)
CloseDevice

Abre el enlace de comunicación con K8055
Cierra el enlace con K8055

Procedimientos de conversión analógicos-digitales

ReadAnalogChannel(Channelno)
ReadAllAnalog(Data1, Data2)

Lee el estado de un canal de entrada analógico
Lee el estado de dos canales de entrada analógicos

Procedimientos de conversión digitales-analógicos

OutputAnalogChannel(Channel, Data)
OutputAllAnalog(Data1, Data2)

Ajusta el canal de salida analógico según los datos
Ajusta los dos canales de salida analógicos según los datos

ClearAnalogChannel(Channel)
ClearAllAnalog

Coloca el canal de salida analógico en el mínimo
Coloca todos los canales de salida analógicos en el mínimo

SetAnalogChannel(Channel)
SetAllAnalog

Coloca el canal de salida analógico en el máximo
Coloca todos los canales de salida analógicos en el máximo

Procedimientos de las salidas digitales

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

Ajusta las salidas digitales según los datos
Borra el canal de salida digital
Borra todos los canales de salida digitales
Ajusta el canal de salida
Ajusta todos los canales de salida

Funciones y procedimientos de las entradas digitales

ReadDigitalChannel(Channel)
ReadAllDigital(Buffer)

Lee el estado del canal de entrada
Lee el estado de todos los canales de entrada

Procedimientos y funciones de los contadores

ResetCounter(CounterNr)

Reinicialice el contador de impulsos 16 bit número 1 o 2

ReadCounter(CounterNr)

Lee el contenido del contador de impulsos número 1 o 2

SetCounterDebounceTime(CounterNr, DebounceTime)

Ajusta el tiempo de eliminación de rebote (debounce time) según el contador de impulsos

Procedimientos y funciones de la K8055D.DLL

OpenDevice

Sintaxis

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

Parámetro

CardAddress: Valor entre 0 y 3, correspondiendo al ajuste del cable de puente (SK5, SK6) en la K8055. Véase tabla 1.

Resultado

Longint: Si ha logrado, el valor de recaída será la dirección de tarjeta del software de K8055. Un valor de recaída de -1 indica que no se ha encontrada la tarjeta K8055.

Descripción

Abre el enlace de comunicación con la tarjeta K8055. Cargue los driver necesarios para poder comunicar vía el puerto USB. Ejecute este procedimiento antes de intentar comunicar con la tarjeta K8055.

También, es posible usar esta función para escribir en la tarjeta K8055 activa y leerla. Todas las rutinas de comunicación que llegan después de esta función, se dirigen a esta tarjeta hasta que se seleccione otra tarjeta con esta función.

Ejemplo

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

CloseDevice

Sintaxis

```
PROCEDURE CloseDevice;
```

Descripción

Carga las rutinas de comunicación para la tarjeta K8055 y el driver que necesita para comunicar por el puerto USB. Es la última acción del programa de aplicación antes de que se cierre.

Ejemplo

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

ReadAnalogChannel

Sintaxis

FUNCTION ReadAnalogChannel (Channel: Longint): Longint;

Parámetro

Channel: Valor entre 1 y 2, correspondiendo al canal AD cuyo estado debe ser leído.

Résultat

Longint: Se leen los datos correspondientes del convertidor digital-analógico.

Descripción

La tensión de entrada del canal de convertidor analógico-digital 8 bit seleccionado se convierte en un valor entre 0 y 255.

Ejemplo

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

ReadAllAnalog

Sintaxis

PROCEDURE ReadAllAnalog(var Data1, Data2: Longint);

Parámetro

Data1, Data2: Se refiere a los enteros largos donde se leerá los datos.

Descripción

Se lee el estado de los dos convertidores analógicos-digitales en una serie de enteros largos.

Ejemplo

```
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

Sintaxis

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

Parámetro

Channel: Valor entre 1 y 2, correspondiendo al número de canal DA 8 bits cuyos datos se deben determinar.

Data: Valor entre 0 en 255 que se debe enviar al convertidor digital-analógico 8 bit.

Descripción

El canal del convertidor digital-analógico 8 bit indicado se modifica según los nuevos datos. Esto significa que los datos corresponden a una tensión específica. El valor 0 corresponde a la tensión de salida mín. (0 Volt) y el valor 255 corresponde a la tensión de salida máx. (+5V). Es posible traducir un valor de dato entre estos extremos con la siguiente fórmula: $\text{Dato} / 255 \times 5V$.

Ejemplo

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

OutputAllAnalog

Sintaxis

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

Parámetros

Data1, Data2: Valor entre 0 y 255 que debe ser enviado al convertidor digital-analógico 8 bit.

Descripción

Se han modificado los dos canales del convertidor digital-analógico 8 bit según los nuevos datos. Esto significa que los datos corresponden a una tensión específica. El valor 0 corresponde a la tensión de salida mín. (0 Volt) y el valor 255 corresponde a la tensión de salida máx. (+5V). Es posible traducir un valor de dato entre estos extremos con la siguiente fórmula: $\text{Dato} / 255 \times 5V$.

Ejemplo

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

ClearAnalogChannel

Sintaxis

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

Parámetro

Channel: Valor entre 1 y 2, correspondiendo al número del canal NA 8 bit cuyos datos se deben borrar.

Descripción

Se coloca el canal NA seleccionado en la tensión de salida mín. (0 Volt).

Ejemplo

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

ClearAllAnalog

Sintaxis

```
PROCEDURE ClearAllAnalog;
```

Descripción

Se colocan los dos canales NA en la tensión de salida mín. (0 Volt).

Ejemplo

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

SetAnalogChannel

Sintaxis

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

Parámetro

Channel: Valor entre 1 y 2, correspondiendo al número de canal NA 8 bit cuyos datos se deben ajustar al máximo.

Descripción

Se coloca el canal del convertidor NA 8 bit seleccionado en la tensión de salida máx.

Ejemplo 15

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

SetAllAnalog

Sintaxis

```
PROCEDURE SetAllAnalog;
```

Descripción

Se colocan todos los canales de convertidores NA 8 bit en la tensión de salida máx.

Ejemplo

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

WriteAllDigital

Sintaxis

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

Parámetro

Data: Valor entre 0 y 255 que se envía al puerto de salida (8 canales).

Descripción

Se actualizan los canales del puerto de salida digital con el estado de los bits correspondientes en el parámetro de los datos. Un nivel elevado (1) significa que se ha ajustado la salida del microcontrolador IC1, y un nivel bajo (0) significa que se ha borrado la salida.

Ejemplo

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

ClearDigitalChannel

Sintaxis

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

Parámetro

Channel: Valor entre 1 y 8, correspondiendo al canal de salida que se debe borrar.

Descripción

Se ha borrado el canal seleccionado.

Ejemplo

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

ClearAllDigital

Sintaxis

```
PROCEDURE ClearAllDigital;
```

Resultado

Se han borrado todas las salidas digitales.

Ejemplo

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

SetDigitalChannel

Sintaxis

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

Parámetro

Channel: Valor entre 1 y 8, correspondiendo al canal de salida que se debe ajustar.

Descripción

Se ha ajustado el canal de salida digital seleccionado.

Ejemplo

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

SetAllDigital

Sintaxis

```
PROCEDURE SetAllDigital;
```

Descripción

Se han ajustado todos los canales de salida digitales.

Ejemplo

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

ReadDigitalChannel

Sintaxis

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

Parámetro

Channel: Valor entre 1 y 5, correspondiendo al canal de entrada cuyo estado debe ser leído.

Resultado

Boolean: TRUE significa que se ha ajustado el canal y FALSE significa que se ha borrado el canal.

Descripción

Se ha leído el estado del canal de entrada seleccionado.

Ejemplo

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

ReadAllDigital

Sintaxis

FUNCTION ReadAllDigital: Longint;

Resultado

Longint: Los 5 LSB corresponden al estado de los canales de entrada. Una señal elevada (1) significa que el canal es elevada (HIGH), Una señal baja (0) significa que el canal es baja (LOW).

Descripción

La función devuelve los estados de las entradas digitales.

Ejemplo

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

ResetCounter

Sintaxis

PROCEDURE ResetCounter(CounterNumber: Longint);

Parámetro

CounterNumber: Valor 1 o 2, correspondiendo al contador que se debe reinicializar.

Descripción

Se ha reinicializado el contador de impulsos seleccionado.

Ejemplo

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

ReadCounter

Sintaxis

FUNCTION ReadCounter(CounterNumber: Longint): Longint;

Parámetro

CounterNumber: Valor 1 o 2, correspondiendo al contador que se debe leer.

Resultado

Longint: El contenido del contador de impulsos 16 bit.

Descripción

La función devuelve el estado del contador de impulsos 16 bit seleccionado.

El contador número 1 cuenta los impulsos que entran por la entrada I1 y el contador número 2 cuenta los impulsos que entran por la entrada I2.

Ejemplo

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

SetCounterDebounceTime

Sintaxis

PROCEDURE SetCounterDebounceTime(CounterNr, DebounceTime: Longint);

Parámetro

CounterNumber: Valor 1 o 2, correspondiendo al contador que se debe ajustar.

DebounceTime: tiempo de eliminación de rebote para el contador de impulsos.

El valor DebounceTime corresponde al tiempo de eliminación de rebote en milisegundos (ms) que se debe ajustar para el contador de impulsos. El tiempo de eliminación de rebote puede variar de 0 a 5000.

Descripción

Las señales de entrada del contador son sometidos a una eliminación de rebote en el software a fin de evitar activaciones incorrectas cuando se usan entradas mecánicas o con relé. El tiempo de eliminación de rebote es el mismo, tanto para flancos descendentes como para flancos ascendentes. El tiempo de eliminación de rebote estándar es de 2ms. Esto significa que la entrada del contador debe quedarse estable durante al mín. 2ms para ser reconocido, lo que limita el número de impulsos por segundo a 200.

Si se coloca el tiempo de eliminación de rebote en 0, es posible un máx. de 2000 impulsos por segundo.

Ejemplo

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

Nuevas funciones y procedimientos multicard

SearchDevices

Sintaxis

FUNCTION SearchDevices(): Longint;

Descripción

La función devuelve todos los aparatos conectados al ordenador. El valor devuelto es un champ bit.

Valor devuelto

- Bin 0000, Dec 0: no está identificado ningún aparato.
- Bin 0001, Dec 1: Dirección de la tarjeta 0 identificada.
- Bin 0010, Dec 2: Dirección de la tarjeta 1 identificada.
- Bin 0100, Dec 4: Dirección de la tarjeta 2 identificada.
- Bin 1000, Dec 8: Dirección de la tarjeta 3 identificada.

Ejemplo: valor devuelto 9 = los aparatos con dirección 0 y 3 están conectados.

Nota

En cuanto se haya conectado una dirección específica del aparato a un programa, ningún otro programa tendrá acceso.

Ejemplo

```
var devices: longint;  
BEGIN  
    devices := SearchDevices; // Returns all devices  
END;
```

SetCurrentDevice

Sintaxis

FUNCTION SetCurrentDevice(Address: Longint): Longint;

Descripción

La función establece el aparato de control. El valor devuelto constituye la dirección del aparato. Si este valor es idéntico a -1, no se ha identificado ningún aparato con parámetro de dirección.

Parámetro

Dirección: Valor de 0 a 3, coincide con la dirección del aparato.

Ejemplo

```
var device: longint;  
BEGIN  
    device := SetCurrentDevice(3); // Returns 3 if device is connected  
END;
```

Versión

Sintaxis

PROCEDURE Version;

Descripción

El procedimiento muestra una pantalla con la versión del software DLL., puede ser útil en caso de problemas con el soporte.

Ejemplo

Usar la K8055D.DLL en Delphi

En este ejemplo de aplicación se encuentran las explicaciones de los procedimientos y funciones K8055D.DLL y un ejemplo demostrando cómo usar las dos funciones más importantes, **OpenDevice** y **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';
    -1: label12.caption:='Card ' + inttostr(CardAddr)+' not found';
  end;
end;

end.
```

Usar la K8055D.DLL en Visual Basic

En este ejemplo de aplicación se encuentran las explicaciones de los procedimientos y funciones K8055D.DLL y un ejemplo demostrando cómo usar las dos funciones más importantes, **OpenDevice** y **CloseDevice**.

Observación : Asegúrese de que se copie K8055D.DLL al fichero 'SYSTEM32' en Windows:

```
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
            Label1.Caption = "Card " + Str(h) + " connected"
        Case -1
            Label1.Caption = "Card " + Str(CardAddress) + " not found"
    End Select
End Sub

Private Sub Form_Terminate()
    CloseDevice
End Sub
```

Usar la K8055D.DLL en Borland C++ Builder

A continuación, Ud. encuentra un listado de K8055D.h que contiene las explicaciones de los procedimientos y funcionamientos K8055D.DLL. Un listado de un ejemplo de aplicación demuestra cómo usar las dos funciones más importantes, **OpenDevice** y **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();
}
//-----
```