

## Introduction to the DLL for the USB Experiment Interface Board K8055/VM110\*

Zahrnuje K8055D.DLL nebo K8055D\_C.DLL, Verze 2.0.x a vyšší

Interfejsová karta K8055 obsahuje 5 digitálních vstupních kanálů a 8 digitálních výstupních kanálů. Navíc jsou zde dva analogové vstupy, dva analogové napájecí výstupy a dva PWM (pulzní šířková modulace) výstupy s 8 bitovým rozlišením. Počet vstupů/výstupů může být dále zvětšen připojením více (až maximálně čtyři) karet k USB konektoru na PC. Každá karta má přiděleno vlastní identifikační číslo pomocí dvou jumperů, SK5 a SK6 (viz tabulka 1 níže s číslováním karet).

Všechny komunikační rutiny jsou obsaženy v knihovně (DLL) K8055.DLL

Tento dokument popisuje všechny funkce a procedury knihovny DLL, které jsou k dispozici pro vaše aplikační programy. Pro volání funkcí a procedur exportovaných pomocí DLL si můžete napsat pro Windows (98SE, 2000, Me, XP) vlastní aplikace v Delphi, Visual Basicu, C++ Builderu nebo jakémkoliv jiném 32-bitovém vývojovém nástroji pro Windows aplikace, který podporuje volání DLL

Uživatelé Visual C++ budou muset zahrnout do svého VC++ projektu speciální soubory K8055D\_C.DLL a K8055D\_C.LIB.

Všechny podporované funkce a procedury K8055D.DLL a K8055D\_C.DLL jsou stejné. Jediný rozdíl je ve volací konvenci pro uživatele C++.

Následuje kompletní přehled procedur a funkcí, které jsou exportovány pomocí K8055D.DLL. Na konci tohoto dokumentu naleznete výpisy vzorových programů pro získání náhledu jak konstruovat vaše vlastní aplikační programy. Příklady jsou napsány v Delphi, Visual Basicu a C++ Builderu. Ve výpisech jsou plné deklarace DLL funkcí a procedur.

Všechny příklady v oddílu popisu funkcí a procedur jsou psány pro Delphi.

SK5	SK6	ADRESA KARTY
ON	ON	0
OFF	ON	1
ON	OFF	2
OFF	OFF	3

**TABULKA 1: Nastavení jumperů SK5, SK6**

**Pozn.:** Tato nastavení je třeba provést před připojením USB kabelu ke kartě K8055 nebo před zapnutím PC.

\* VM110 je smontovaná verze K8055.

## Přehled procedur a funkcí K8055D.DLL

### Obecné procedury

OpenDevice(CardAddress)  
CloseDevice

Otevře komunikační kanál k zařízení K8055  
Uzavře kanál k zařízení K8055

### Procedury analogového a digitálního převodníku

ReadAnalogChannel(ChannelNo)  
ReadAllAnalog(Data1, Data2)

Načte stav jednoho analogového vstupního kanálu  
Načte stav obou analogových vstupních kanálů

### Procedury digitálního analogového převodu

OutputAnalogChannel(Channel, Data)  
OutputAllAnalog(Data1, Data2)  
ClearAnalogChannel(Channel)  
ClearAllAnalog  
  
SetAnalogChannel(Channel)  
SetAllAnalog

Nastaví analogový výstupní kanál podle dat  
Nastaví oba analogové výstupní kanály podle dat  
Nastaví analogový výstupní kanál na minimum  
Nastaví všechny analogové výstupní kanály na minimum  
Nastaví analogový výstupní kanál na maximum  
Nastaví všechny analogové výstupní kanály na maximum

### Procedury digitálního výstupu

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

Nastaví digitální výstupy podle dat  
Vynuluje výstupní kanál  
Vynuluje všechny výstupní kanály  
Nastaví výstupní kanál  
Nastaví všechny výstupní kanály

### Digitální vstupní procedury a funkce

ReadDigitalChannel(Channel)  
ReadAllDigital(Buffer)

Přečte stav vstupního kanálu  
Přečte stav všech vstupních kanálů

### Procedury a funkce čítače

ResetCounter(CounterNr)  
  
ReadCounter(CounterNr)  
  
SetCounterDebounceTime(CounterNr, DebounceTime)

Resetuje 16 bitový čítač pulzů číslo 1 nebo čítač číslo 2  
Načte obsah čítače pulzů číslo 1 nebo čítače číslo 2  
Nastaví protizákmitový čas čítače pulzů

---

## Procedury a funkce K8055D.DLL

---

### OpenDevice

*Syntax*

FUNCTION OpenDevice(CardAddress: Longint): Longint;

*Parametr*

CardAddress: Hodnota mezi 0 a 3 která odpovídá nastavení jumper (SK5, SK6) na kartě K8055. Viz tabulka 1.

*Výsledek*

Longint: V případě úspěchu udává návratová hodnota adresu karty na kterou z hardwaru K8055. Návratová hodnota -1 indikuje, že karta K8055 nebyla nalezena.

*Popis*

Otevírá komunikační kanál karty K8055. Natáhne nezbytné ovladače pro komunikaci přes USB port. Tato procedura musí být provedena před jakýmkoliv pokusem o komunikaci s kartou K8055.

Tato funkce může být také použita k výběru aktivní karty K8055 pro čtení a zápis dat. Veškeré komunikační rutiny volané po této funkci jsou adresovány této kartě dokud není touto funkcí zvolena jiná karta.

*Příklad*

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

---

### CloseDevice

*Syntax*

PROCEDURE CloseDevice;

*Popis*

Uvolní z paměti komunikační rutiny pro kartu K8055 a vymaže ovladač potřebný pro komunikaci přes USB port. Toto je poslední akce aplikace programu před jeho ukončením.

*Příklad*

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

## ReadAnalogChannel

### *Syntax*

FUNCTION ReadAnalogChannel (Channel: Longint): Longint;

### *Parametr*

Kanál: Hodnota mezi 1 a 2 která odpovídá AD kanálu jehož stav je nařízen.

### *Výsledek*

Longint: Jsou na této adrese data odpovídajícího analogového digitálního převodníku.

### *Popis*

Vstupní napětí zvoleného kanálu 8-bitového analogového digitálního převodníku je převedeno na hodnotu která leží mezi 0 a 255.

### *Příklad*

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

---

## ReadAllAnalog

### *Syntax*

PROCEDURE ReadAllAnalog(var Data1, Data2: Longint);

### *Parametr*

Data1, Data2: Pointery na long integer kde budou data načtena.

### *Popis*

Do pole long integer jsou načteny stavy obou analogových digitálních převodníků.

### *Příklad*

```
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

### Syntax

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

### Parametry

Channel: Hodnota mezi 1 a 2 která odpovídá kanálu 8-bitového DA převodníku, jehož data se mají nastavit.

Data: Hodnota mezi 0 až 255 která má být poslána do 8-bitového digitálního analogového převodníku.

### Popis

Označený 8-bitový digitální analogový kanál je nastaven podle nových dat. To znamená, že data odpovídají určitému napětí. Hodnota 0 odpovídá minimálnímu výstupnímu napětí (0 Volt) a hodnota 255 odpovídá maximálnímu výstupnímu napětí (+5V). Hodnota 'Data' ležící mezi těmito extrémy může být položena následujícím vzorcem :  $\text{Data} / 255 \times 5V$ .

### Příklad

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

## OutputAllAnalog

### Syntax

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

### Parameters

Data1, Data2: Hodnota mezi 0 a 255 která má být poslána do 8-bitového digitálního analogového převodníku.

### Popis

Oba 8-bitové kanály digitálního analogového převodníku jsou nastaveny podle nových dat. To znamená, že data odpovídají určitému napětí. Hodnota 0 odpovídá minimálnímu výstupnímu napětí (0 Volt) a hodnota 255 odpovídá maximálnímu výstupnímu napětí (+5V). Hodnota 'Data' ležící mezi těmito extrémy může být položena následujícím vzorcem :  $\text{Data} / 255 \times 5V$ .

### Příklad

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

## ClearAnalogChannel

### Syntax

PROCEDURE ClearAnalogChannel(Channel: Longint);

### Parametr

Kanál: Hodnota mezi 1 a 2 která odpovídá číslu kanálu 8-bitového DA převodníku jehož data se mají vymazat.

### Popis

Zvolený DA-kanál je nastaven na minimální výstupní napětí (0 Volt).

### Příklad

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

## ClearAllAnalog

*Syntax*

PROCEDURE ClearAllAnalog;

*Popis*

Oba DA-kanály jsou nastaveny na minimální výstupní napětí (0 Volt) .

*Příklad*

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

---

## SetAnalogChannel

*Syntax*

PROCEDURE SetAnalogChannel(Channel: Longint);

*Parametr*

Kanál: Hodnota mezi 1 a 2, která odpovídá číslu 8-bitového DA kanálu v němž se mají data nastavit na maximum.

*Popis*

Zvolený kanál 8-bitového digitálního analogového převodníku je nastaven na maximální výstupní napětí.

*Příklad 15*

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

---

## SetAllAnalog

*Syntax*

PROCEDURE SetAllAnalog;

*Popis*

Všechny kanály 8-bitového digitálního analogového převodníku jsou nastaveny na maximální výstupní napětí.

*Příklad*

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

---

## WriteAllDigital

### *Syntax*

PROCEDURE WriteAllDigital(Data: Longint);

### *Parametr*

Data: Hodnota mezi 0 a 255, která je poslána na výstupní port (8 kanál ).

### *Popis*

Kanály digitálního výstupního portu jsou nastaveny podle stavu odpovídajících bitů v datovém parametru. Vysoká (1) úroveň znamená, že bude výstup mikrokontroleru IC1 nastaven, a nízká (0) úroveň znamená, že bude výstup vynulován.

### *Příklad*

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

---

## ClearDigitalChannel

### *Syntax*

PROCEDURE ClearDigitalChannel(Channel: Longint);

### *Parametr*

Channel: Hodnota mezi 1 a 8, která odpovídá výstupnímu kanálu, který má být vynulován.

### *Popis*

Zvolený kanál je vynulován.

### *Příklad*

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

---

## ClearAllDigital

### *Syntax*

PROCEDURE ClearAllDigital;

### *Výsledek*

Všechny digitální výstupy jsou vynulovány.

### *Příklad*

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

## SetDigitalChannel

*Syntax*

PROCEDURE SetDigitalChannel(Channel: Longint);

*Parametr*

Channel: Hodnota mezi 1 a 8, která odpovídá výstupnímu kanálu, který má být nastaven.

*Popis*

Zvolený digitální výstupní kanál je nastaven.

*Příklad*

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

---

## SetAllDigital

*Syntax*

PROCEDURE SetAllDigital;

*Popis*

Všechny digitální výstupní kanály jsou nastaveny.

*Příklad*

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

---

## ReadDigitalChannel

*Syntax*

FUNCTION ReadDigitalChannel(Channel: Longint): Boolean;

*Parametr*

Channel: Hodnota mezi 1 a 5, která odpovídá vstupnímu kanálu jehož stav se má na íst.

*Výsledek*

Boolean: TRUE znamená, že byl kanál nastaven a FALSE znamená, že byl vynulován.

*Popis*

Stav zvoleného vstupního kanálu je na íst.

*Příklad*

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

---



## ReadAllDigital

### Syntax

FUNCTION ReadAllDigital: Longint;

### Výsledek

Longint: 5 LSB odpovídá stavu vstupních kanálů. Vysoká úroveň (1) znamená, že je kanál ve stavu HIGH, nízká úroveň (0) znamená, že je kanál ve stavu LOW.

### Popis

Funkce vrací stav digitálních vstupů.

### Příklad

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

---

## ResetCounter

### Syntax

PROCEDURE ResetCounter(CounterNumber: Longint);

### Parametr

CounterNumber: Hodnota 1 nebo 2, která odpovídá číslu, který má být vynulován.

### Popis

Zvolené číslo pulzu je vynulováno.

### Příklad

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

---

## ReadCounter

### Syntax

FUNCTION ReadCounter(CounterNumber: Longint): Longint;

### Parametr

CounterNumber: Hodnota 1 nebo 2, která odpovídá číslu, který má být načten.

### Výsledek

Longint: Obsah 16 bitového čísla a pulzu.

### Popis

Funkce vrací stav zvoleného 16 bitového čísla a pulzu.  
Číslo 1 čísla pulzu přivedené na vstup I1 a číslo 2 čísla pulzu přivedené na vstup I2.

### Příklad

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

## SetCounterDebounceTime

### *Syntax*

PROCEDURE SetCounterDebounceTime(CounterNr, DebounceTime: Longint);

### *Parametr*

CounterNumber: Hodnota 1 nebo 2, jež odpovídá řádku, který má být nastaven.

DebounceTime: Protizákmitový čas pro řádkový pulz.

Hodnota protizákmitového času odpovídá protizákmitovému času v milisekundách (ms) jež se má nastavit pro řádkový pulz. Hodnota protizákmitového času může být v rozsahu 0 až 5000.

### *Popis*

Vstupy řádku jsou softwarově ošetřeny proti zákmitům aby se zabránilo chybnému spouštění při použití mechanických přepínačů nebo relé na vstupu. Protizákmitový čas je stejný pro vzestupnou i sestupnou hranu. Základní hodnota protizákmitového času je 2ms. To znamená, že vstupní hodnota řádku musí být stabilní po dobu min. 2ms aby byla rozeznána, z čehož plyne maximální rychlost ítání asi 200 na tení za sekundu.

Pokud je protizákmitový čas nastaven na 0, je maximální rychlost ítání asi 2000 na tení za sekundu.

### *Příklad*

BEGIN

SetCounterDebounceTime(1,100);

// The debounce time for counter number 1 is set to 100ms

END;

---

## Nové funkce a procedury multikarty

---

### SearchDevices

#### Syntaxe

FUNCTION SearchDevices(): Longint;

#### Popis

Tato funkce vrací seznam všech p ípojených za ízení k po íta í. Návratová hodnota je bitové pole.

#### Návratová hodnota

- 1 Bin 0000, Dec 0 : Nenalezeno žádné za ízení
- 2 Bin 0001, Dec 1 : Nalezena karta s adresou 0.
- 3 Bin 0010, Dec 2 : Nalezena karta s adresou 1.
- 4 Bin 0100, Dec 4 : Nalezena karta s adresou 2.
- 5 Bin 1000, Dec 8 : Nalezena karta s adresou 3.

P íklad : návratová hodnota 9 = jsou p ípojena za ízení s adresami 0 a 3.

#### Poznámka

**Jakmile je ur ítá adresa za ízení propojena s programem, nem ůže k ní získat p ístup jiný program.**

#### P íklad

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

---

### SetCurrentDevice

#### Syntaxe

FUNCTION SetCurrentDevice(Address: Longint): Longint;

#### Popis

Funkce nastavuje práv ůvládané za ízení. Návratová hodnota je adresa za ízení, je-li tato hodnota -1, nebylo nalezeno žádné za ízení s tímto adresovým parametrem.

#### Parametr

Adresa: Hodnota 0 až 3, která odpovídá adresám za ízení.

#### P íklad

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

---

### Version

#### Syntaxe

PROCEDURE Version;

#### Popis

Tato procedura zobrazí okno s íslem verze DLL softwaru, m ůže být volána podp ůrnými programy.

#### P íklad

```
BEGIN  
    Version; // Popup window with version number  
END;
```

## Použití K8055D.DLL v Delphi

V tomto příkladu aplikace jsou deklarace procedur a funkcí K8055D.DLL a příklad jak použít dvě nejčastější funkce DLL : **OpenDevice** a **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.
```

## Použití K8055D.DLL ve Visual Basicu

Ve výpisu aplikace jsou deklarace procedur a funkcí K8055D.DLL a příklad jak použít dvě nejčastější funkce DLL: **OpenDevice** a **CloseDevice**.

**Pozn.:** Ujistěte se, že je soubor K8055D.DLL nakopírován do složky SYSTEM32 ve 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
```

## Použití K8055D.DLL v Borland C++ Builderu

Níže naleznete výpis souboru K8055D.h zahrnující deklarace procedur a funkcí K8055D.DLL. Výpis aplikace ního příkladu ukazuje jak použít dvě nejzákladnější funkce DLL: **OpenDevice** a **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();
}
//-----
```