

## Introductie tot de DLL voor de USB Experiment Interface Board K8055/VM110

Voor K8055.DLL of K8055D\_C.DLL, versie 2.0x en hoger.

De K8055 interface board heeft 5 digitale inputkanalen en 8 digitale outputkanalen. Daarnaast zijn er 2 analoge ingangen, 2 analoge spanningsuitgangen en 2 PWM (Pulse Width Modulation) uitgangen met 8 bit resolutie. Het aantal in- en uitgangen kan uitgebreid worden door meer (max. 4) kaarten aan te sluiten op de USB-connectoren van de PC. Iedere kaart heeft een eigen identificatienummer dat bepaald wordt met 2 jumpers, SK5 en SK6 (zie tabel 1 hieronder voor de kaartnummering).

Alle communicatieroutines zijn verzameld in een Dynamic Link Library (DLL) K8055D.DLL.

Dit document omschrijft alle functies en procedures van de DLL die toegankelijke zijn via uw applicatieprogramma. Door de functies en procedures van de DLL op te roepen, kunt u uw eigen Windowsapplicaties (98SE, 2000, Me, XP) schrijven in Delphi, Visual Basic, C++ Builder of gelijk welke andere 32-bit Windows applicatie-ontwikkelingstool die oproepen naar een DLL ondersteunt.

Wenst u Visual C++ te gebruiken, dan heeft u de speciale K8055D\_C.DLL en K8055D\_C.LIB files nodig, inclusief in het VC++ project.

Alle ondersteunde functies en procedures van de K8055D.DLL en K8055D\_C.DLL zijn identiek. Het enige verschil is de 'calling convention' voor C++ gebruikers.

Nu volgt een volledig overzicht van de procedures en functies die door de K8055D.DLL geëxporteerd worden. Achteraan dit document staan listings van voorbeeldprogramma's om u een idee te geven hoe u uw eigen applicatieprogramma's kunt aanmaken. De voorbeelden zijn geschreven in Delphi, Visual Basic en C++ Builder. In de listings staan volledige verklaringen voor de DLL functies en procedures.

Merk op dat alle voorbeelden in het stuk over functie- en procedureomschrijvingen geschreven zijn voor Delphi.

SK5	SK6	CARD ADDRESS
ON	ON	0
OFF	ON	1
ON	OFF	2
OFF	OFF	3

**TABLE 1: Jumper SK5, SK6 Settings**

**Opmerking:** De instelling van de jumpers moet gebeuren voor de USB-kabel verbonden wordt met de K8055 kaart of voor de PC aangezet wordt.

\* VM110 is de gebouwde versie van de K8055.

## Overzicht van de procedures en functies van de K8055D.DLL

### Algemene procedures

OpenDevice(CardAddress)  
CloseDevice

*Opent de communicatielink met de K8055  
Sluit de link met de K8055*

### Analoog - digitaal omzettingsprocedures

ReadAnalogChannel(Channelno)  
ReadAllAnalog(Data1, Data2)

*Leest de status van 1 analoog ingangskanaal  
Leest de status van beide analoge ingangskanalen*

### Digitaal - analoog omzettingsprocedures

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

*Stelt het analoge uitgangskanaal in volgens de data  
Stelt beide analoge uitgangskanalen in volgens de data  
Zet het analoge uitgangskanaal op minimum  
Zet alle analoge uitgangskanalen op minimum  
Zet het analoge uitgangskanaal op maximum  
Zet alle analoge uitgangskanalen op maximum*

### Digitale uitgangprocedures

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

*Stelt de digitale uitgangen in volgens de data  
Maakt het uitgangskanaal vrij  
Maakt alle uitgangskanalen vrij  
Stelt het uitgangskanaal in  
Stelt alle uitgangskanalen in*

### Digitale ingangprocedures en -functies

ReadDigitalChannel(Channel)  
ReadAllDigital(Buffer)

*Leest de status van het ingangskanaal  
Leest de status van alle ingangskanalen*

### Functies en procedures voor de tellers

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

*Stelt de 16-bit pulse-teller nummer 1 of teller nummer 2 terug  
Leest de inhoud van pulse-teller nummer 1 of teller nummer 2  
Stelt de ontdenderingstijd in volgens de pulse-teller*

---

## Procedures en functies van de K8055D.DLL

---

### OpenDevice

#### *Syntax*

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

#### *Parameter*

CardAddress: Waarde tussen 0 en 3 die overeenkomt met de instelling van de jumper (SK5, SK6) op de K8055. Zie tabel 1.

#### *Resultaat*

Longint: Indien geslaagd zal de returnwaarde het kaartadres van de K8055 hardware zijn. Een returnwaarde van -1 geeft aan dat de K8055-kaart niet gevonden is.

#### *Omschrijving*

Opent de communicatielink met de K8055-kaart. Laadt de nodige drivers om via de USB-poort te communiceren. Deze procedure moet uitgevoerd worden voor u kunt proberen te communiceren met de K8055-kaart.

Deze functie kan ook gebruikt worden om de actieve K8055-kaart te lezen en erop te schrijven. Alle communicatie-routines die na deze functie komen, gaan naar die kaart tot via deze functie een andere kaart geselecteerd wordt.

#### *Voorbeeld*

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

---

### CloseDevice

#### *Syntax*

```
PROCEDURE CloseDevice;
```

#### *Omschrijving*

Haalt de communicatieroutines voor de K8055-kaart en de driver voor communicatie via de USB-poort af. Dit is de laatste actie van het applicatieprogramma voor het afsluit.

#### *Voorbeeld*

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

---

## ReadAnalogChannel

### *Syntax*

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

### *Parameter*

Channel: Waarde tussen 1 en 2, komt overeen met het AD-kanaal waarvan de status moet worden gelezen.

### *Resultaat*

Longint: De corresponderende data van de digitaal-analoog converter wordt gelezen.

### *Omschrijving*

De ingangsspanning van het geselecteerde 8-bits analoog-digitaal converterkanaal wordt omgezet naar een waarde tussen 0 en 255.

### *Voorbeeld*

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

---

## ReadAllAnalog

### *Syntax*

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

### *Parameter*

Data1, Data2: Verwijst naar de lange integers waar de data zal worden gelezen.

### *Omschrijving*

De status van beide analoog-digitaal converters worden gelezen in een reeks lange integers.

### *Voorbeeld*

```
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);
```

### *Parameters*

Channel: Waarde tussen 1 en 2 die overeenkomt met het 8-bits DA kanaalnummer waarvan de data moet worden bepaald.

Data: Waarde tussen 0 en 255 die naar de 8-bit digitaal-analoog converter moet worden gestuurd.

### *Omschrijving*

Het aangegeven 8-bits digitaal-analoog converterkanaal wordt aangepast volgens de nieuwe data. Dit betekent dat de data overeenkomt met een specifieke spanning. De waarde 0 komt overeen met een

minimum uitgangsspanning (0 Volt) en de waarde 255 komt overeen met de maximum uitgangsspanning (+5V). Een datawaarde tussen deze uitersten kan worden vertaald met deze formule:  $\text{Data} / 255 \times 5V$ .

#### *Voorbeeld*

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

---

## OutputAllAnalog

#### *Syntax*

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

#### *Parameters*

Data1, Data2: Waarde tussen 0 en 255 die naar de 8-bits digitaal-analoog converter moet worden gestuurd.

#### *Omschrijving*

Beide 8-bits digitaal-analoog converterkanalen wordt aangepast volgens de nieuwe data. Dit betekent dat de data overeenkomt met een specifieke spanning. De waarde 0 komt overeen met een minimum uitgangsspanning (0 Volt) en de waarde 255 komt overeen met de maximum uitgangsspanning (+5V). Een datawaarde tussen deze uitersten kan worden vertaald met deze formule:  $\text{Data} / 255 \times 5V$ .

#### *Voorbeeld*

```
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);
```

#### *Parameter*

Channel: Waarde tussen 1 en 2 die overeenkomt met het 8-bits DA kanaalnummer waarop de data moet worden gewist.

#### *Omschrijving*

Het geselecteerde DA-kanaal wordt op de minimum uitgangsspanning gezet (0 Volt).

#### *Voorbeeld*

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

---

## ClearAllAnalog

#### *Syntax*

```
PROCEDURE ClearAllAnalog;
```

#### *Omschrijving*

Beide DA-kanalen worden op de minimum uitgangsspanning gezet (0 Volt) .

*Voorbeeld*

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

---

## SetAnalogChannel

*Syntax*

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

*Parameter*

**Channel** : Waarde tussen 1 en 2 die overeenkomt met het 8-bits DA kanaalnummer waarop de data op het maximum moet worden gezet.

*Omschrijving*

Het geselecteerde 8-bits digitaal-analoog converterkanaal wordt op de maximum uitgangsspanning gezet.

*Voorbeeld*

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

---

## SetAllAnalog

*Syntax*

```
PROCEDURE SetAllAnalog;
```

*Omschrijving*

Alle kanalen van de 8-bits digitaal-analoog converters worden op de maximum uitgangsspanning gezet.

*Voorbeeld*

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

---

## WriteAllDigital

### Syntax

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

### Parameter

Data: Waarde tussen 0 en 255 die naar de uitgangspoort gestuurd wordt (8 kanalen).

### Omschrijving

De kanalen van de digitale uitgangspoort worden upgedate met de status van de corresponderende bits in de data parameter. Een hoog (1) niveau betekent dat de uitgang van de microcontroller IC1 ingesteld is, en een laag (0) niveau betekent dat de uitgang gewist is.

### Voorbeeld

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

---

## ClearDigitalChannel

### Syntax

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

### Parameter

Channel: Waarde tussen 1 en 8 die overeenkomt met het uitgangskanaal dat moet worden gewist.

### Omschrijving

Het geselecteerde kanaal wordt gewist.

### Voorbeeld

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

---

## ClearAllDigital

### Syntax

```
PROCEDURE ClearAllDigital;
```

### Resultaat

Alle digitale uitgangen worden gewist.

### Voorbeeld

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

---

## SetDigitalChannel

### *Syntax*

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

### *Parameter*

Channel: Waarde tussen 1 en 8 die overeenkomt met het uitgangskanaal dat moet worden ingesteld.

### *Omschrijving*

Het geselecteerde digitale uitgangskanaal wordt ingesteld.

### *Voorbeeld*

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

---

## SetAllDigital

### *Syntax*

```
PROCEDURE SetAllDigital;
```

### *Omschrijving*

Alle digitale uitgangskanalen worden ingesteld.

### *Voorbeeld*

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

---

## ReadDigitalChannel

### *Syntax*

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

### *Parameter*

Channel: Waarde tussen 1 en 5 die overeenkomt met het ingangskanaal dat moet worden gelezen.

### *Resultaat*

Boolean: TRUE betekent dat het kanaal ingesteld is en FALSE betekent dat het gewist is.

### *Omschrijving*

De status van het geselecteerde ingangskanaal wordt gelezen.

### *Voorbeeld*

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

---



## ReadAllDigital

### *Syntax*

FUNCTION ReadAllDigital: Longint;

### *Resultaat*

Longint: De 5 LSB komen overeen met de status van de ingangskanalen. Een hoog signaal (1) betekent dat het kanaal hoog staat (HIGH), een laag signaal (0) betekent dat het kanaal laag staat (LOW).

### *Omschrijving*

De functie stuurt de status van de digitale ingangen terug.

### *Voorbeeld*

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

---

## ResetCounter

### *Syntax*

PROCEDURE ResetCounter(CounterNumber: Longint);

### *Parameter*

CounterNumber: Waarde 1 of 2, die overeenkomt met de teller die moet worden teruggesteld.

### *Omschrijving*

De geselecteerde pulsteller wordt teruggesteld.

### *Voorbeeld*

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

---

## ReadCounter

### *Syntax*

FUNCTION ReadCounter(CounterNumber: Longint): Longint;

### *Parameter*

CounterNumber: Waarde 1 of 2, die overeenkomt met de teller die moet worden gelezen.

### *Resultaat*

Longint: De inhoud van de 16-bit pulsteller.

### *Omschrijving*

De functie stuurt de status van de geselecteerde 16-bit pulsteller terug.

Teller nummer 1 telt de impulsen die binnenkomen via ingang I1 en teller nummer 2 telt de impulsen die binnenkomen via ingang I2.

### *Voorbeeld*

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

## SetCounterDebounceTime

### *Syntax*

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

### *Parameter*

CounterNumber: Waarde 1 of 2, die overeenkomt met de in te stellen teller.

DebounceTime: Ontdenderingstijd voor de pulsteller.

De DebounceTime-waarde komt overeen met de ontdenderingstijd in milliseconden (ms) die moet ingesteld worden voor de pulsteller. De ontdenderingstijd kan variëren van 0 tot 5000.

### *Omschrijving*

De ingangssignalen van de teller worden in de software ontdenderd om valse activering te voorkomen als er mechanische of relaisingen worden gebruikt. De ontdenderingstijd is dezelfde voor zowel dalende als stijgende zijden. De standaard ontdenderingstijd is 2ms. Dit betekent dat de telleringang tenminste 2ms stabiel moet blijven om herkend te worden, wat resulteert in een maximum van ongeveer 200 pulsen per seconde.

Als de ontdenderingstijd op 0 gezet wordt, kan er een maximum van 2000 impulsen per seconde gehaald worden.

### *Voorbeeld*

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

---

## Nieuwe multiscard functies en procedures

---

### SearchDevices

#### *Syntaxis*

FUNCTION SearchDevices(): Longint;

#### *Omschrijving*

De functie stuurt alle op de computer aangesloten toestellen terug. De terug te sturen waarde is een bitveld.

Terug te sturen waarde

- Bin 0000, Dec 0: Geen enkel toestel gevonden.
- Bin 0001, Dec 1: Kaartadres 0 gevonden.
- Bin 0010, Dec 2: Kaartadres 1 gevonden.
- Bin 0100, Dec 4: Kaartadres 2 gevonden.
- Bin 1000, Dec 8: Kaartadres 3 gevonden.

Voorbeeld: teruggestuurde waarde 9 = toestellen met adres 0 en 3 zijn aangesloten.

#### *Opmerking*

**Als er éénmaal een specifiek toesteladres aan een programma is aangesloten, krijgt een ander programma geen toegang.**

#### *Voorbeeld*

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

---

### SetCurrentDevice

#### *Syntaxis*

FUNCTION SetCurrentDevice(Address: Longint): Longint;

#### *Omschrijving*

De functie stelt het huidige aanstuurtoestel in. De teruggestuurde waarde is het toesteladres. Wanneer deze waarde -1 is, is er geen toestel met adresparameter gevonden.

#### *Parameter*

Adres: Waarde 0 tot 3, stemt overeen met het toesteladres.

#### *Voorbeeld*

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

---

### Versie

#### *Syntaxis*

PROCEDURE Version;

#### *Omschrijving*

De procedure toont een venster met de DLL softwareversie, kan gevraagd worden in geval van problemen met de support.

*Voorbeeld*

END;

## De K8055D.DLL gebruiken in Delphi

In dit applicatievoorbeeld staan de verklaringen van de K8055D.DLL procedures en functies, en een voorbeeld van hoe de twee belangrijkste functies, **OpenDevice** en **CloseDevice**, worden gebruikt.

```
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.

```

## De K8055D.DLL gebruiken in Visual Basic

In dit applicatievoorbeeld staan de verklaringen van de K8055D.DLL procedures en functies, en een voorbeeld van hoe de twee belangrijkste functies, **OpenDevice** en **CloseDevice**, worden gebruikt.

**Opmerking:** Verzeker u ervan dat het bestand K8055D.DLL naar Window's 'SYSTEM32'-folder wordt gekopieerd:

```

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

```

## De K8055D.DLL gebruiken in Borland C++ Builder

Hieronder staat een listing van de K8055D.h die de verklaringen van de K8055D.DLL procedures en functions bevat. Een listing van een applicatievoorbeeld toont hoe de twee belangrijkste DLL-functies, **OpenDevice** en **CloseDevice**, worden gebruikt.

```

//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();
}
//-----

```