

Inledning till DLL för USB Experiment-Interfacekortet K8055

Interfacekortet K8055 har 5 digitala inkanaler och 8 digitala utkanaler. Dessutom finns det två analoga spänningsingångar, två analoga spänningsutgångar och två PWM (Pulse Width Modulation)-utgångar med 8 bitars upplösning. Pulse Width Modulation ~ Pulsbredds-modulation. Antalet ingångar/utgångar kan expanderas ytterligare genom anslutning av flera (upp till ett maximum av 4) kort till PC:ns USB-kontakter. Varje kort har fått ett eget identifikationsnummer med hjälp av "jumprar", SK5 och SK6 (se tabellen nedan för kortnumrering).

Alla kommunikationsrutiner ingår i ett dynamiskt länkbibliotek, DLL (Dynamic Link Library), K8055D.DLL

Detta dokument beskriver alla funktioner och procedurer hos DLL, som är tillgängliga för ditt applikationsprogram.

Genom att anropa funktionerna och procedurerna exporterade av DLL, kan du skriva Windowsbaserade (98SE, 2000, Me XP)-kundapplikationer i Delphi, Visuell Basic, C++ Builder eller vilket annat 32-bitars Windows applikationsutvecklingsverktyg som stödjer anrop till ett DLL.

Nedan följer en fullständig översikt av procedurerna och funktionerna som exporteras av K8055D.DLL. I slutet av detta dokument finns listningar av programexempel för att vinna insikt i hur man skriver egna applikationsprogram. Exempelen är skrivna i Delphi, Visuell Basic och C++ Builder. I listningarna finns det fullständiga förklaringar av DLL-funktionerna och –procedurerna.

Notera att alla exempel i funktions- och procedurbeskrivningskapitlet är skrivna i Delphi.

SK5	SK6	KORT ADRESS
PÅ	PÅ	0
AV	PÅ	1
PÅ	AV	2
AV	AVF	3

TABLE 1: Jumper SK5, SK6 uppsättningar

Notera: Dessa uppsättningar måste göras innan USB-kabeln ansluts till K8055-kortet eller innan PC:n slås på.

Översikt av procedurerna och funktionerna hos K8055D.DLL

Allmänna procedurer

OpenDevice(CardAddress)
CloseDevice

Öppnar kommunikationslänken till K8055-enheten
Stänger kommunikationslänken till K8055-enheten

Procedurer för Analog/Digitalomvandling

ReadAnalogChannel(Channelno)
ReadAllAnalog(Data1, Data2)

Läser status hos en analog input-kanal
Läser status hos båda analoga input-kanalerna

Procedurer för Digital/Analogomvandling

OutputAnalogChannel(Channel, Data)

OutputAllAnalog(Data1, Data2)

ClearAnalogChannel(Channel)
ClearAllAnalog
SetAnalogChannel(Channel)
SetAllAnalog

Ställer in den analoga output-kanalen med avseende på data
Ställer in båda analoga output-kanalerna med avseende på data
Ställer in den analoga output-kanalen på minimum
Sets all analogue output channels to minimum
Ställer in den analoga output-kanalen på maximum
Ställer in alla analoga output-kanaler på maximum

Digitala outputprocedurer

WriteAllDigital(Data)

ClearDigitalChannel(Channel)
ClearAllDigital
SetDigitalChannel(Channel)
SetAllDigital

Ställer in de digitala output-kanalerna med avseende på data
Nollställer output-kanalen
Nollställer alla output-kanaler
Ställer in output-kanalen
Ställer in alla output-kanaler

Digitala inputprocedurer och funktioner

ReadDigitalChannel(Channel)
ReadAllDigital(Buffer)

Läser input-kanalens status
Läser alla input-kanalers status

Procedurer och funktioner för räknare

ResetCounter(CounterNr)

ReadCounter(CounterNr)

SetCounterDebounceTime(CounterNr, DebounceTime)

Ställer tillbaka 16-bitars pulsräknaren nummer 1 eller räknare nummer 2
Läser innehållet i pulsräknare nummer 1 eller räknare nummer 2
Ställer in "debounce"-tiden på pulsräknaren

Procedurer och funktioner hos K8055D.DLL

OpenDevice

Syntax

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

Parameter

Kortadress: Värde mellan 0 och 3 som motsvarar jumprarna:s (SK5, SK6) inställning på K8055-kortet. Se tabell 1.

Resultat

Longint: Om operationen lyckas kommer returnerande värde vara kortadressen avläst från K8055-hårdvaran.

Returvärde -1 indikerar att det EJ finns något K8055-kort.

Beskrivning

Öppnar kommunikationslänken till K8055-kortet. Laddar drivrutinerna nödvändiga för kommunikationen med USB-porten. Denna procedur måste utföras innan några försök görs för att kommunicera med K8055-kortet.

Denna funktion kan också användas till att få det aktiva K8055-kortet att läsa och skriva data. Alla kommunikationsrutiner efter detta funktionsanrop adresseras till detta kort ända till dess ett annat kort väljs av funktionsanropet.

Exempel

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

CloseDevice

Syntax

```
PROCEDURE CloseDevice;
```

Beskrivning

Avladdar kommunikationsrutinerna för K8055-kortet och avladdar drivrutinen som krävs för att kommunicera med USB-porten. Detta är det sista applikationsprogrammet gör innan det terminerar.

Exempel

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

ReadAnalogChannel

Syntax

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

Parameter

Channel: Värde mellan 1 och 2 som motsvarar numret hos den AD(AnalogDigital)-kanal vars status skall läsas.

Resultat

Longint: Motsvarande data för AD-omvandlaren läses.

Beskrivning

Inputspänningen hos den valda 8-bitars AD-omvandlarkanalerna omvandlas till ett värde mellan 0 och 255.

Exempel

```
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: Pekare till långa heltal varifrån data kommer att läsas

Beskrivning

Status hos båda AD-omvandlarna inläses till en vektor av långa heltal.

Exempel

```
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);
```

Parametrar

Channel: Värde mellan 1 och 2 som motsvarar numret hos den 8-bitars DA-kanal vars data skall inställas

Data: Värde mellan 0 och 255 som kommer att sändas till 8-bitars DA-omvandlaren

Beskrivning

Den indikerade 8-bitars DA-omvandlarkanalerna förändras enligt de nya data. Detta betyder att data motsvarar en specifik spänning. Värdet 0 motsvarar en min utspänning på 0V och värdet 255 motsvarar en max utspänning på +5V. Ett värde mellan dessa extremer kan översättas med följande formel: $\text{Data} / 255 \times 5V$.

Exempel

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

OutputAllAnalog

Syntax

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

Parametrar

Data1, Data2: Värde mellan 0 och 255, vilket kommer att sändas till 8-bitars DA-omvandlaren.

Beskrivning

Båda 8-bitars DA-omvandlarkanalerna förändras enligt de nya data. Detta betyder att data motsvarar en specifik spänning. Värdet 0 motsvarar en min utspänning på 0V och värdet 255 motsvarar en max utspänning på +5V. Ett värde mellan dessa extremer kan översättas med följande formel: $\text{Data} / 255 \times 5V$.

Exempel

```
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: Värde mellan 1 och 2 som motsvarar numret hos den 8-bitars DA-kanal vars data skall raderas.

Beskrivning

Den valda DA-kanalen sätts till minimal utspänning (0V).

Exempel

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

ClearAllAnalog

Syntax

```
PROCEDURE ClearAllAnalog;
```

Beskrivning

Båda DA-kanalerna sätts till minimal utspänning (0V).

Exempel

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

SetAnalogChannel

Syntax

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

Parameter

Channel: Värde mellan 1 och 2 som motsvarar kanalnumret hos den 8-bitars DA-omvandlare för vilken data sätts till maximum.

Beskrivning

Den valda 8-bitars DA-omvandlarkanalen sätts till maximal utspänning.

Exempel 15

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

SetAllAnalog

Syntax

```
PROCEDURE SetAllAnalog;
```

Beskrivning

Alla kanaler hos 8-bitars DA-omvandlarna sätts till maximal utspänning.

Exempel

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

WriteAllDigital

Syntax

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

Parameter

Data: Värde mellan 0 och 255 som kommer att sändas till utporten (8 kanaler).

Beskrivning

Kanalerna hos den digitala utporten uppdateras med status för motsvarande bitar i dataparametern. En hög (1) nivå betyder att mikroregulatorn IC1:s output sätts, och en låg (0) nivå betyder att outputen nollställs..

Exempel

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

ClearDigitalChannel

Syntax

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

Parameter

Channel: Värde mellan 1 och 8, vilket motsvarar outputkanalen som skall nollställas.

Beskrivning

Den valda kanalen nollställs.

Exempel

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

ClearAllDigital

Syntax

```
PROCEDURE ClearAllDigital;
```

Resultat

Alla digitala outputar nollställs.

Exempel

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

SetDigitalChannel

Syntax

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

Parameter

Channel: Värde mellan 1 och 8, vilket motsvarar den outputkanal som skall sättas.

Beskrivning

Den valda outputkanalen sätts.

Exempel

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

SetAllDigital

Syntax

```
PROCEDURE SetAllDigital;
```

Beskrivning

Alla digitala outputkanaler sätts.

Exempel

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

ReadDigitalChannel

Syntax

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

Parameter

Channel: Värde mellan 1 och 5, vilket motsvarar den inputkanal, vars status skall läsas.

Resultat

Boolean: TRUE betyder att kanalen har satts och FALSE betyder att den nollställts.

Beskrivning

Status hos den valda inputkanalen läses.

Exempel

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

ReadAllDigital

Syntax

```
FUNCTION ReadAllDigital: Longint;
```

Resultat

Longint: 5 LSB motsvarar statusen hos inputkanalerna. Hög (1) betyder att kanalen är HIGH, låg (0) betyder att kanalen är LOW.

Beskrivning

Funktionen returnerar status hos de digitala inputarna.

Exempel

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

ResetCounter

Syntax

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

Parameter

CounterNumber: Värde 1 eller 2, vilket motsvarar den räknare som skall återställas.

Beskrivning

Den valda pulsräknaren återställs.

Exempel

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

ReadCounter

Syntax

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

Parameter

CounterNumber: Värde 1 eller 2, vilket motsvarar den räknare som skall läsas.

Resultat

Longint: Innehållet hos 16-bitars pulsräknare.

Beskrivning

Funktionen returnerar status hos den valda 16-bitars pulsräknaren.

Räknare nummer 1 räknar pulser matade mot input I1 och räknare nummer 2 räknar pulser matade mot input I2.

Exempel

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

SetCounterDebounceTime

Syntax

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

Parameter

CounterNumber: Värde 1 eller 2, vilket motsvarar den räknare som skall sättas.

DebounceTime: "Debounce time" för pulsräknaren.

DebounceTime-värdet motsvarar den "debounce time" som skall sättas för pulsräknaren. "Debounce time"-värdet kan variera mellan 0 och 5000.

Beskrivning

Räknarinputs är "debounced" i programvaran för att förhindra falsk trigging när mekaniska switchar eller reläinputs används. "Debounce"-tiden är lika både för fallande som stigande flanker. Default "debounce"-tid är 2ms. Det betyder att räknarinputen måste vara stabil i åtminstone 2 ms innan den mäts, vilket innebär en maximal räknarhastighet på omkring 200 antal uppräknningar per sekund.

Om "debounce"-tiden är satt till 0, betyder det att maximala räkningshastigheten är omkring 2000 uppräknningar per sekund.

Exempel

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

Användning av K8055.DLL i Delphi

I följande applikationsexempel illustreras procedurerna och funktionerna i K8055D.DLL och ett exempel på hur man använder de två viktigaste DLL funktionsanropen: **OpenDevice** och **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.
```

Användning av K8055D.DLL i Visuell Basic.

I följande applikationsexempel illustreras procedurerna och funktionerna i K8055D.DLL och ett exempel på hur man använder de två viktigaste DLL funktionsanropen: **OpenDevice** och **CloseDevice**.

Notera: Var säker på att filen K8055D.DLL kopieras till Windows' SYSTEM32 mappen:

```
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
```

Användning av K8055D.DLL i Borland C++ Builder

Nedan följer en listning av K8055D.h inkluderande deklarationerna i K8055D.DLL:s procedurer och funktioner. En listning av ett applikationsexempel visar hur man använder de två viktigaste DLL funktionsanropen: **OpenDevice** och **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 :
        Labell->Caption = "Card " + IntToStr(CardAddr) + " not found";
    }
}
//-----

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