

Inledning till DLL för USB Experiment-Interfacekortet K8055

O K8055, Interface USB, tem 5 canais digitais de entrada e 8 canais digitais de saída. Adicionalmente, existem duas entradas analógicas, duas saídas de tensão analógicas e duas saídas PWM (Pulse Width Modulation) com uma resolução de 8 bit. É possível aumentar o número de entradas e saídas através da ligação de mais placas (até quatro, no máximo) aos conectores USB do seu PC. Cada placa tem o seu próprio número de identificação que se determina através dos dois jumpers, SK5 e SK6 (ver Tabela 1 para a numeração das placas).

Todas as rotinas de comunicação estão agrupadas numa DLL (Dynamic Link Library) K8055D.DLL.

Este documento descreve todas as funções e procedimentos da DLL disponível para a sua aplicação. Invocando as funções e os procedimentos da DLL, pode escrever as suas próprias aplicações em Windows (98SE, 2000, Me, XP), Delphi, Visual Basic, C++ Builder ou qualquer outra ferramenta com 32 bits que suporte chamadas a DLL.

Uma visão completa dos procedimentos e funções que são exportados pelo K8055D.DLL vêm mencionados a seguir. No final deste documento encontrará listagens de exemplos de programa (Exemplo) para que possa perceber e ter uma ideia do modo como fazer os programas para a sua própria aplicação. Este Exemplo foi escrito em Delphi, Visual Basic e C++ Builder. Nas listagens poderá ver explicações para as funções e procedimentos DLL.

Note que todas as funções e procedimentos de Exemplos descritos na secção 'Descrição' foram escritos para Delphi.

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

TABELA 1: Colocação dos Jumpers: SK5 e SK6

Nota: Estas ligações devem ser feitas antes da ligação do cabo USB à placa K8055 ou antes de ligar o seu PC.

Resumo dos Procedimentos e Funções da K8055D.DLL

Procedimentos Gerais

<code>OpenDevice(CardAddress)</code>	<i>Abra o link de comunicação com o dispositivo K8055</i>
<code>CloseDevice</code>	<i>Feche o link</i>

Procedimentos de conversão Analógico / Digital

<code>ReadAnalogChannel(ChannelNo)</code>	<i>Leia o estado de um canal de entrada analógico</i>
<code>ReadAllAnalog(Data1, Data2)</code>	<i>Leia o estado de dois canais de entrada analógicos</i>

Procedimentos de conversão Digital / Analógica

<code>OutputAnalogChannel(Channel, Data)</code>	<i>Ajuste o canal de saída analógico de acordo com os dados</i>
<code>OutputAllAnalog(Data1, Data2)</code>	<i>Ajuste os dois canais de saída analógicos de acordo com os dados</i>
<code>ClearAnalogChannel(Channel)</code>	<i>Coloque o canal de saída analógico no mínimo</i>
<code>ClearAllAnalog</code>	<i>Coloque todos os canais de saída analógicos no mínimo</i>
<code>SetAnalogChannel(Channel)</code>	<i>Coloque o canal de saída analógico no máximo</i>
<code>SetAllAnalog</code>	<i>Coloque todos os canais de saída analógicos no máximo</i>

Procedimentos das Saídas Digitais

<code>WriteAllDigital(Data)</code>	<i>Ajuste as saídas digitais de acordo com os dados</i>
<code>ClearDigitalChannel(Channel)</code>	<i>Limpe o canal de saída digital</i>
<code>ClearAllDigital</code>	<i>Limpe todos os canais de saída digital</i>
<code>SetDigitalChannel(Channel)</code>	<i>Ajuste o canal de saída</i>
<code>SetAllDigital</code>	<i>Ajuste todos os canais de saída</i>

Funções e Procedimento das Entradas Digitais

<code>ReadDigitalChannel(Channel)</code>	<i>Leia o estado do canal de entrada</i>
<code>ReadAllDigital(Buffer)</code>	<i>Leia o estado de todos os canais de entrada</i>

Procedimentos e Funções dos Contadores

<code>resetCounter(CounterNr)</code>	<i>Reinicie o contador de impulsos 16 bit número 1 ou 2</i>
<code>ReadCounter(CounterNr)</code>	<i>Leia o conteúdo do contador de impulsos número 1 ou 2</i>
<code>SetCounterDebounceTime(CounterNr, DebounceTime)</code>	<i>Ajuste o tempo de 'debounce' do contador de impulsos</i>

Procedimentos e Funções da K8055D.DLL

OpenDevice

Síntaxe

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

Parâmetro

CardAddress: Valor entre 0 e 3 que corresponde ao jumper (SK5, SK6) colocado na placa K8055. Ver tabela 1.

Resultado

Longint: Se bem sucedido, o valor obtido será o endereço da placa K8055.
Se o valor obtido for (-1), a placa K8055 não foi encontrada.

Descrição

Abra o link de comunicação com a placa K8055. Carregue as drivers necessárias para poder comunicar através da porta USB. Este procedimento deve ser feito antes de tentar comunicar com a placa K8055.

Esta função também pode ser utilizada para escrever e ler na placa K8055. Todas as rotinas de comunicação que chegam depois desta função, são endereçadas a esta placa até que se seleccione outra placa com esta função.

Exemplo

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

CloseDevice

Síntaxe

```
PROCEDURE CloseDevice;
```

Descrição

Carregue as rotinas de comunicação para a placa K8055 e o driver necessário para comunicar através da porta USB. É a última acção do programa de aplicação antes de o fechar.

Exemplo

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

ReadAnalogChannel

Síntaxe

FUNCTION ReadAnalogChannel (Channel: Longint): Longint;

Parâmetro

Channel: Valor entre 1 e 2 que corresponde ao canal AD cujo estado deve ser lido.

Resultado

Longint: Leitura dos dados correspondentes ao conversor Analógico / Digital.

Descrição

A tensão de entrada do canal de conversão Analógico / Digital de 8 bit é convertido para um valor entre 0 e 255.

Exemplo

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

ReadAllAnalog

Síntaxe

PROCEDURE ReadAllAnalog(var Data1, Data2: Longint);

Parâmetro

Data1, Data2: Ponteiros para os inteiros onde se irá ler os dados.

Descrição

O estado dos Conversores Analógico / Digital é lido num array de inteiros.

Exemplo

```
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

Síntaxe

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

Parâmetros

Channel: Valor entre 1 e 2 que corresponde ao número do canal DA de 8-bit cujos dados devem ser determinados.

Data: Valor entre 0 e 255 que deve ser enviado ao conversor Digital / Analógico de 8-bit.

Descrição

O canal do conversor Digital / Analógico de 8-bit é alterado de acordo com o novo dado. Isto significa que os dados correspondem a uma tensão específica. O valor 0 corresponde à tensão de saída mínima (0 Volt) e o valor 255 corresponde ao valor da tensão de saída máxima (+5V). É possível traduzir um valor 'Data' entre estes extremos através da seguinte fórmula: $\text{Data} / 255 \times 5V$.

Exemplo

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

OutputAllAnalog

Síntaxe

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

Parâmetros

Data1, Data2: Valor entre 0 e 255 deve ser enviado ao conversor Digital / Analógico de 8-bit.

Descrição

Modificaram-se os dois canais do conversor Digital / Analógico de 8-bit de acordo com os novos dados. Isto significa que os dados correspondem a uma tensão específica. O valor 0 corresponde à tensão de saída mínima (0 Volt) e o valor 255 corresponde ao valor da tensão de saída máxima (+5V). É possível traduzir um valor 'Data1' ou 'Data2' entre estes extremos através da seguinte fórmula: $\text{Data} / 255 \times 5V$.

Exemplo

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

ClearAnalogChannel

Síntaxe

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

Parâmetro

Channel: Valor entre 1 e 2 que corresponde ao número do canal DA de 8-bit cujos dados devem ser apagados.

Descrição

Coloca-se o canal DA seleccionado na tensão de saída mínima (0 Volt).

Exemplo

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

ClearAllAnalog

Síntaxe

```
PROCEDURE ClearAllAnalog;
```

Descrição

Colocam-se os dois canais DA na tensão de saída mínima (0 Volt).

Exemplo

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

SetAnalogChannel

Síntaxe

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

Parâmetro

Channel: Valor entre 1 e 2 que corresponde ao número do canal DA de 8-bit cujos dados devem ser ajustados para o máximo.

Descrição

Coloca-se o canal do conversor Digital / Analógico de 8-bit na tensão de saída máxima.

Exemplo 15

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

SetAllAnalog

Síntaxe

```
PROCEDURE SetAllAnalog;
```

Descrição

Colocam-se todos os canais do conversor Digital / Analógico de 8-bit na tensão de saída máxima.

Exemplo

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

WriteAllDigital

Síntaxe

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

Parâmetro

Data: Valor entre 0 e 255 que é enviado para a porta de saída (8 canais).

Descrição

Os canais da porta de saída digital são actualizados com o estado dos bits correspondentes aos parâmetros dos dados. Um nível elevado (1) significa que se ajustou a saída do microcontrolador IC1, e um nível baixo (0) significa que a saída foi apagada.

Exemplo

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

ClearDigitalChannel

Síntaxe

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

Parâmetro

Channel: Valor entre 1 e 8 que corresponde ao canal de saída que deve ser apagado.

Descrição

O canal seleccionado foi apagado.

Exemplo

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

ClearAllDigital

Síntaxe

```
PROCEDURE ClearAllDigital;
```

Resultado

Todas as saídas digitais foram apagadas.

Exemplo

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

SetDigitalChannel

Síntaxe

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

Parâmetro

Channel: Valor entre 1 e 8 que corresponde ao canal de saída que se deve ajustar.

Descrição

O canal de saída digital seleccionado foi ajustado.

Exemplo

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

SetAllDigital

Síntaxe

```
PROCEDURE SetAllDigital;
```

Descrição

Todos os canais de saída foram ajustados.

Exemplo

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

ReadDigitalChannel

Síntaxe

FUNCTION ReadDigitalChannel(Channel: Longint): Boolean;

Parâmetro

Channel: Valor entre 1 e 5 que corresponde ao canal de entrada cujo estado deve ser lido.

Resultado

Boolean: TRUE significa que o canal foi ajustado e FALSE que foi apagado.

Descrição

O estado do canal de entrada seleccionado foi lido.

Exemplo

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

ReadAllDigital

Síntaxe

FUNCTION ReadAllDigital: Longint;

Resultado

Longint: O 5 LSB corresponde ao estado dos canais de entrada. Um sinal elevado (1) significa que o canal está HIGH. Um sinal baixo (0) significa que o canal está LOW.

Descrição

A função devolve o estado das entradas digitais.

Exemplo

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

ResetCounter

Síntaxe

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

Parâmetro

CounterNumber: Valor 1 ou 2, correspondente ao contador que se deve reiniciar.

Descrição

Reiniciou-se o contador de impulsos seleccionado.

Exemplo

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

ReadCounter

Síntaxe

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

Parâmetro

CounterNumber: Valor 1 ou 2, correspondente ao contador que se deve ler.

Resultado

Longint: O conteúdo do contador de impulsos de 16 bit.

Descrição

A função devolve o estado do contador de impulsos de 16 bit seleccionado.

O contador número 1 conta os impulsos que entram pela entrada I1 e o contador número 2 os impulsos que entram pela entrada I2.

Exemplo

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

SetCounterDebounceTime

Síntaxe

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

Parâmetro

CounterNumber: Valor 1 ou 2, correspondente ao contador que se deve ajustar.

DebounceTime: Debounce time para o contador de impulsos.

O valor DebounceTime corresponde ao tempo de debounce em milissegundos (ms) que se deve ajustar para o contador de impulsos. O valor deste Debounce pode variar entre 0 e 5000.

Descrição

Os sinais de entrada do contador são submetidos a debounce no software a fim de evitar activações incorrectas quando se utilizam entradas mecânicas ou com relés. O tempo de debounce é o mesmo, tanto para flancos descendentes como para flancos ascendentes. O tempo standard de debounce é de 2ms. Isto significa que a entrada do contador deve manter-se estável durante, pelo menos, 2ms para ser reconhecido, o que limita o número de impulsos em cerca de 200 por segundo.

Se colocar o tempo de debounce a 0, então o número máximo de impulsos será cerca de 2000 por segundo.

Exemplo

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

Utilização da K8055D.DLL em Delphi

Neste Exemplo de aplicação encontra explicações dos procedimentos e funções K8055D.DLL e um Exemplo de como utilizar as funções mais importantes: **OpenDevice** e **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.
```

Utilização da K8055D.DLL em Visual Basic

Neste Exemplo de aplicação encontra explicações dos procedimentos e funções K8055D.DLL e um Exemplo de como utilizar as funções mais importantes: **OpenDevice** e **CloseDevice**.

Nota: Verifique se o ficheiro K8055D.DLL foi copiado para a pasta do Windows 'SYSTEM32':

```
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
```

Utilização da K8055D.DLL em Borland C++ Builder

Em baixo encontra uma listagem da K8055D.h incluindo explicações dos procedimentos e funções K8055D.DLL. Nesta encontra uma listagem de um Exemplo de como utilizar as funções mais importantes: **OpenDevice** e **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();
}
//-----
```