

Technická příručka

Úvod

Obecné

Experimentální USB deska K8055N má 5 digitálních vstupních kanálů a 8 digitálních výstupních kanálů. Kromě toho jsou na desce dva analogové vstupy, dva napěťové analogové výstupy a dva PWM (pulzní šířková modulace) výstupy s 8 bitovým rozlišením. Počet vstupů / výstupů lze ještě dále rozšířit připojením více (maximálně čtyř) karet k USB portům počítače. Každá karta je rozlišena svou vlastní adresou pomocí dvou propojek - SK5 a SK6 (viz Tab. 1 níže).

Všechny komunikační funkce a procedury jsou obsaženy v DLL knihovně K8055D.DLL.

V tomto návodu si detailně popíšeme každou z těchto funkcí a procedur obsažených této DLL knihovně. Voláním funkcí a procedur z DLL můžete napsat vlastní aplikaci pro Windows v Delphi, Visual Basic nebo v jakémkoliv jiném vývojovém prostředí pro 32bitové aplikace Windows, které podporuje volání knihovny DLL.

Kompletní přehled procedur a funkcí, které jsou obsaženy v dynamické knihovně K8055D.DLL. Všimněte si, že všechny příklady v popisu funkcí a procedur, jsou napsány v jazyce C++. Ve složce K8055 příkladů jsou příklady napsané v jazyce Visual Basic 2008 Express, Visual C# 2008 Express, Visual C++ 2008 Express, VB6.0, MS Excel VBA, Delphi 5, Borland C++ Builder 6 a Dev-C++.

Čtenáři by měli mít znalost základních datových typů, jakož i základní znalosti o operačním systému Microsoft Windows.

Uživatelé Microsoft Visual Studio vezměte, prosím, na vědomí: K8055D.DLL je standardní Windows DLL, nemůžete na ni odkazovat.

Konvence volání

Konvence volání je schéma, které udává jak lze přijímat parametry od jejich volajícího a jak se vrací výsledek. Různé programovací jazyky používají různé konvence volání, a proto je důležité vědět, jakou konvenci volání používá použitý programovací jazyk, a jaké konvence volání používá K8055D.DLL.

Nejběžnější konvence volání je **stdcall** konvence volání, a to je také ta, kterou jsme použili pro naši DLL.

Pokud používáte .NET (VB.NET nebo C#), nemusíte se o to starat, protože konvence volání v .NET je také **stdcall**. Nicméně, pokud používáte C pro import funkcí poskytovaných DLL, budete tomu muset věnovat zvláštní pozornost.

Nastavení adresy karty

SK5	SK6	Adresa karty
ZAP	ZAP	0
VYP	ZAP	1
ZAP	VYP	2
VYP	VYP	3

Tab. 1: Nastavení propojek SK5, SK6 pro určení adresy karty

Nastavení adresy karty musí být provedeno dříve, než je kabel USB připojen k K8055N kartě nebo před zapnutím do PC.

Přehled funkcí a procedur

Obecné funkce

<code>int</code> OpenDevice(<code>int</code> CardAddress);	Otevírá komunikační spojení na K8055N
<code>void</code> CloseDevice();	Zavře odkaz na zařízení K8055N
<code>int</code> SearchDevices();	Dává informaci o počtu připojených zařízení v počítači
<code>int</code> SetCurrentDevice(<code>int</code> lngCardAddress);	Nastaví aktuální ovládané zařízení
<code>int</code> Version();	Poskytuje informace číslu verze DLL knihovny

Funkce analogově-digitálního převodníku

<code>int</code> ReadAnalogChannel(<code>int</code> Channel);	Načte stav zadaného analogového vstupního kanálu
<code>void</code> ReadAllAnalog(<code>int</code> *Data1, <code>int</code> *Data2);	Načte stav obou analogových vstupních kanálů

Funkce digitálně-analogového převodníku

<code>void</code> OutputAnalogChannel(<code>int</code> Channel, <code>int</code> Data);	Nastaví analogový zadaný výstupní kanál podle Data
<code>void</code> OutputAllAnalog(<code>int</code> Data1, <code>int</code> Data2);	Nastaví oba analogové výstupní kanály podle Data1, 2
<code>void</code> ClearAnalogChannel(<code>int</code> Channel);	Nastaví zadaný analogový výstup na minimální hodnotu
<code>void</code> ClearAllAnalog();	Nastaví oba analogové výstupy na minimální hodnotu
<code>void</code> SetAnalogChannel(<code>int</code> Channel);	Nastaví zadaný analogový výstup na maximální hodnotu
<code>void</code> SetAllAnalog();	Nastaví oba analogové výstupy na maximální hodnotu

Funkce digitálních výstupů

<code>void</code> WriteAllDigital(<code>int</code> Data);	Nastaví digitální výstupy podle data
<code>void</code> ClearDigitalChannel(<code>int</code> Channel);	Nastaví zadaná bit na hodnotu 0
<code>void</code> ClearAllDigital();	Nastaví všechny bity na hodnotu 0
<code>void</code> SetDigitalChannel(<code>int</code> Channel);	Nastaví zadaná bit na hodnotu 1
<code>void</code> SetAllDigital();	Nastaví všechny bity na hodnotu 1

Funkce digitálních vstupů

<code>bool</code> ReadDigitalChannel(<code>int</code> Channel);	Čte stav vstupního kanálu (bitu)
<code>int</code> ReadAllDigital();	Čte stav všech vstupních kanálů (bitů)

Funkce čítačů

<code>void</code> ResetCounter(<code>int</code> CounterNr);	Vynuluje stav 16 bitového čítače pulsů 1 nebo 2
<code>int</code> ReadCounter(<code>int</code> CounterNr);	Načte aktuální hodnotu čítače pulsů 1 nebo 2
<code>void</code> SetCounterDebounceTime(<code>int</code> CounterNr, <code>int</code> DebounceTime);	Nastaví časovou konstantu pulsů, které budou počítány

Funkce zpětného čtení

<code>int</code> ReadBackDigitalOut();	Čte zpět výstupní hodnotu digitálních výstupů karty
<code>void</code> ReadBackAnalogOut(<code>int</code> *Buffer);	Čte zpět výstupní hodnotu D/A výstupů karty

Popis funkcí a procedur

OpenDevice

Syntaxe:

```
int OpenDevice(int CardAddress);
```

Parametry:

CardAddress: Hodnota mezi 0 a 3, které odpovídají propojky (SK5, SK6) nastavené na desce K8055N. Viz tabulka 1.

Výsledek:

Int: Návrátová hodnota pokud se podařilo najít K8055N dané adresy. Návrátová hodnota -1 znamená, že K8055N karta nebyla nalezena.

Popis:

Otevírá komunikační spojení a kartou K8055N. Načte ovladače nutné ke komunikaci přes USB port. Tento postup musí být proveden před jakýmkoli pokusy o komunikaci s K8055N kartou. Tuto funkci lze také použít k volbě aktivní K8055N karty pro čtení a zápis dat. Všechny komunikační rutiny po tomto volání funkce jsou určeny k této kartě, dokud nebude vybrána jiná karta zavoláním této funkce.

Příklad:

```
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->Text = "Karta c. " + h.ToString() + " připojena";
        Timer1->Enabled = true;
        break;
    case -1 :
        Label1->Text = "Karta c. " + CardAddr.ToString() + " nenalezena";
        break;
}
```

SearchDevices

Syntaxe:

```
int SearchDevices();
```

Popis:

Pomocí této funkce mohou být otevřeny všechny karty K8055N. Není potřeba používat OpenDevice. Tato funkce vrátí všechny připojené zařízení K8055N k počítači. Vracená hodnota je bitové pole.

Vracená hodnota

- BIN 0000, DEC 0 : nebylo nalezeno žádné zařízení
- BIN 0001, DEC 1 : byla nalezena karta adresy 0
- BIN 0010, DEC 2 : byla nalezena karta adresy 1
- BIN 0100, DEC 4 : byla nalezena karta adresy 2
- BIN 1000, DEC 8 : byla nalezena karta adresy 3

Příklad: návratová hodnota 9 = zařízení s adresou 0 a 3 jsou připojeny.

Příklad:

```
k = SearchDevices();  
if (k)  
    Timer1->Enabled = true;  
if (k & 1)  
    RadioButton9->Enabled = true;  
if (k & 2)  
    RadioButton10->Enabled = true;  
if (k & 4)  
    RadioButton11->Enabled = true;  
if (k & 8)  
    RadioButton12->Enabled = true;
```

SetCurrentDevice

Syntaxe:

```
int SetCurrentDevice(int CardAddress);
```

Popis:

Funkce nastavení aktuálního ovládané zařízení. Vracená hodnota je adresa zařízení, pokud je tato hodnota -1 nebyla nalezena žádná deska s adresou zadanou parametrem.

Parametry:

CardAddress: hodnoty 0 až 3, což odpovídá na adrese zařízení.

Příklad:

```
SetCurrentDevice(0);
```

CloseDevice

Syntaxe:

```
void CloseDevice();
```

Popis:

Uvolní komunikační rutiny pro K8055N karty a uvolní ovladače potřebné pro komunikaci přes USB port. Je to poslední akce aplikačního programu před ukončením.

Příklad:

```
private: System::Void Form1_FormClosed(System::Object^ sender,  
System::Windows::Forms::FormClosedEventArgs^ e)  
{  
    CloseDevice();  
}
```

ReadAnalogChannel

Syntaxe:

```
int ReadAnalogChannel(int Channel);
```

Parametry:

Channel: Hodnota 1 nebo 2, která odpovídá na AD kanálu, jehož stav je třeba číst.

Výsledek:

Int: Odpovídá načtené hodnotě zadaného analogově-digitálního převodníku.

Popis:

Vstupní napětí vybraného kanálu 8-bitový analogově-digitálního převodníku je převedena na hodnotu, která leží mezi hodnotami 0 a 255.

Příklad:

```
Label1->Text = ReadAnalogChannel(1).ToString();
```

ReadAllAnalog

Syntaxe:

```
void ReadAllAnalog(int *Data1, int *Data2);
```

Parametry:

Data1, Data2: Ukazatele na celá čísla (32-bit), kam budou data načtena.

Popis:

Hodnoty obou A-D převodníků jsou čteny do dvojice celých čísel.

Příklad:

```
int Data1;  
int Data2;  
ReadAllAnalog(&Data1, &Data2);
```

OutputAnalogChannel

Syntaxe:

```
void OutputAnalogChannel(int Channel, int Data);
```

Parametry:

Channel: Hodnota 1 nebo 2, která odpovídá číslu kanálu D-A převodníku, který má být nastaven.

Data: Hodnota mezi 0 a 255, která má být odeslána na 8-bitový D-A převodník.

Popis:

Zvolený kanál 8-bitového D-A převodníku se nastaví dle zadané hodnoty. To znamená, že odeslaná data odpovídají konkrétnímu výstupnímu napětí. Hodnota 0 odpovídá minimálnímu výstupnímu napětí (0 V) a hodnota 255 odpovídá maximální výstupnímu napětí (+5 V). Hodnoty proměnné "Data" ležící mezi těmito extrémy, mohou být přepočteny pomocí následujícího vzorce: $\text{Data} / 255 \times 5V$.

Příklad:

```
OutputAnalogChannel(1, 255 - VScrollBar1->Value);
```

OutputAllAnalog

Syntaxe:

```
void OutputAllAnalog(int Data1, int Data2);
```

Parametry:

Data1, Data2: Hodnoty mezi 0 a 255, které má být odeslány na 8-bitové D-A převodníky.

Popis:

Oba kanály 8-bitového D-A převodníku se nastaví dle zadaných hodnot. To znamená, že odeslaná data odpovídají konkrétním výstupním napětím. Hodnota 0 odpovídá minimální výstupní napětí (0 V) a hodnota 255 odpovídá maximální výstupní napětí (+5 V). Hodnoty proměnné "Data1" nebo "Data2" leží mezi těmito extrémy, mohou být přepočteny pomocí následujícího vzorce: $\text{Data} / 255 \times 5V$.

Příklad:

```
OutputAllAnalog(50, 255);
```

ClearAnalogChannel

Syntaxe:

```
void ClearAnalogChannel(int Channel);
```

Parametry:

Channel: Hodnota 1 nebo 2, která odpovídá číslu kanálu 8-bitového D-A převodníku, na kterém mají být data vymazána.

Popis:

Vybraný kanál D-A převodníku je nastaven na minimální výstupní napětí (0 V).

Příklad:

```
ClearAnalogChannel(1);
```

ClearAllAnalog

Syntaxe

```
void ClearAllAnalog();
```

Popis:

Oba kanály D-A převodníku jsou nastaveny na minimální výstupní napětí (0 V).

Příklad:

```
ClearAllAnalog();
```

SetAnalogChannel

Syntaxe:

```
void SetAnalogChannel(int Channel);
```

Parametry:

Channel: Hodnota 1 nebo 2, která odpovídá kanálu 8-bitového D-A převodníku, ve kterém mají být data nastavena na maximum.

Popis:

Zvolený kanál 8-bitového D-A převodníku je nastaven na maximální výstupní napětí.

Příklad:

```
SetAnalogChannel(1);
```

SetAllAnalog

Syntaxe:

```
void SetAllAnalog();
```

Popis:

Všechny kanály 8-bitového D-A převodníku jsou nastaveny na maximální výstupní napětí.

Příklad:

```
SetAllAnalog();
```

WriteAllDigital

Syntaxe:

```
void WriteAllDigital(int Data);
```

Parametry:

Data: Hodnota mezi 0 a 255, která je odeslána na výstupní digitální port (8 kanálů).

Popis:

Kanály digitálního výstupního portu jsou nastaveny dle odpovídajících bitů v parametru Data. Vysoká logická úroveň (1) znamená, že výstup mikrokontroléru IC3 je nastaven, a nízká logická úroveň (0) znamená, že výstup je uzemněn.

Příklad:

```
WriteAllDigital(0x55);
```

ClearDigitalChannel

Syntaxe:

```
void ClearDigitalChannel(int Channel);
```

Parametry:

Channel: Hodnota mezi 1 a 8, které odpovídá číslo výstupního kanálu, který má být vymazán (0).

Příklad:

```
CheckBox9->Checked ? SetDigitalChannel(1): ClearDigitalChannel(1);
```

ClearAllDigital

Syntaxe:

```
void ClearAllDigital();
```

Popis:

Všechny výstupní kanály budou vymazány (0).

Příklad:

```
ClearAllDigital();
```

SetDigitalChannel

Syntaxe:

```
void SetDigitalChannel(int Channel);
```

Parametry:

Channel: Hodnota mezi 1 a 8, která odpovídá číslu výstupního kanálu, který má být nastaven (1).

Popis:

Zadaný digitální výstupní kanál je nastaven (1).

Příklad:

```
CheckBox9->Checked ? SetDigitalChannel(1): ClearDigitalChannel(1);
```

SetAllDigital

Syntaxe:

```
void SetAllDigital();
```

Popis:

Všechny výstupní kanály budou nastaveny (1).

Příklad:

```
SetAllDigital();
```

ReadDigitalChannel

Syntaxe:

```
bool ReadDigitalChannel(int Channel);
```

Parametry:

Channel: Hodnota mezi 1 a 5, která odpovídá vstupnímu kanálu, jehož stav je třeba načíst.

Výsledek:

bool: TRUE znamená, že kanál byl nastaven a FALSE znamená, že je vymazán.

Popis:

Načtení stavu zadaného vstupního digitálního kanálu.

Příklad:

```
CheckBox4->Checked = ReadDigitalChannel(1);
```

ReadAllDigital

Syntaxe:

```
int ReadAllDigital();
```

Výsledek:

int: 5 LSB odpovídají stavu digitálních vstupních kanálů. Vysoká (1) znamená, že kanál je HIGH, nízká (0), znamená to, že kanál je LOW.

Description

Funkce vrací stav digitálních vstupů.

Příklad:

```
i = ReadAllDigital();
CheckBox4->Checked = (i & 1)>0;
CheckBox5->Checked = (i & 2)>0;
CheckBox6->Checked = (i & 4)>0;
CheckBox7->Checked = (i & 8)>0;
CheckBox8->Checked = (i & 16)>0;
```

ResetCounter

Syntaxe:

```
void ResetCounter(int CounterNr);
```

Parametry:

CounterNr: Hodnota 1 nebo 2, která odpovídá číslu čítače, který se resetuje.

Popis:

Resetuje zadaný čítač pulzů.

Příklad:

```
ResetCounter(1);
```

ReadCounter

Syntaxe:

```
int ReadCounter(int CounterNr);
```

Parametry:

CounterNr: Hodnota 1 nebo 2, která odpovídá číslu načteného čítače.

Výsledek:

int: Hodnota 16 bitového čítače pulzů.

Popis:

Funkce vrací stav vybraného 16 bitového čítače impulzů. Počítadlo číslo 1 počítá impulzy na vstupu I1 a číslo 2 počítá impulzy na vstupu I2.

Příklad:

```
TextBox1->Text = ReadCounter(1).ToString();
```

SetCounterDebounceTime

Syntaxe:

```
void SetCounterDebounceTime(int CounterNr, int DebounceTime);
```

Parametry:

CounterNr: Hodnota 1 nebo 2, která odpovídá číslu čítače, který bude nastaven.

DebounceTime: Časová konstanta pro čítač pulzů.

Hodnota DebounceTime odpovídá době trvání pulzu v milisekundách (ms), kterou je možné nastavit pro čítač pulzů. Časová konstanta hodnoty se může pohybovat mezi 0 a 5000.

Popis:

Vstupy čítačů lze softwarově ošetřit na délku trvání pulzu, aby se zabránilo falešnému spuštění, pokud jsou na vstupu použity mechanické spínače nebo relé. Čas vstupního zpoždění je mezi oběma hranami - sestupnou a vzestupnou hranou. Výchozí čas vstupního zpoždění je 2 ms. To znamená, že vstup čítače, musí být stabilní po dobu nejméně 2 ms předtím, než se uznává (přičemž maximální počet rychlost asi 200 impulzů za sekundu). Je-li doba vstupní zpoždění je nastavena na 0, pak je maximální rychlost počítání je asi 2000 impulzů za sekundu.

Příklad:

```
SetCounterDebounceTime(1, 10);
```

Version

Syntaxe:

```
int Version();
```

Výstup:

int: 32 bitové celé číslo, které reprezentuje verzi použité DLL (4 číslice). Každý byte je jedna číslice.

Popis:

Načte se číslo verze dané DLL knihovny.

Příklad:

```
int ver = Version();
Label9->Text = (ver >> 24).ToString()+"."+((ver >> 16) & 0xFF).ToString()+"."+
               +((ver >> 8) & 0xFF).ToString()+"."+(ver & 0xFF).ToString()
```

ReadBackDigitalOut

Syntaxe:

```
int ReadBackDigitalOut();
```

Výsledek:

int: Hodnota mezi 0 a 255, která je zpětně načtena z digitálního výstupního portu.

Popis:

Načte zpět byte, který je právě nastaven na digitálním výstupním portu.

Příklad:

```
int DigitalOut;
DigitalOut = ReadBackDigitalOut();
ReadBackAnalogOut(AnalogOut);

CheckBox9->Checked = (DigitalOut & 1)>0;
CheckBox10->Checked = (DigitalOut & 2)>0;
CheckBox11->Checked = (DigitalOut & 4)>0;
CheckBox12->Checked = (DigitalOut & 8)>0;
CheckBox13->Checked = (DigitalOut & 16)>0;
CheckBox14->Checked = (DigitalOut & 32)>0;
CheckBox15->Checked = (DigitalOut & 64)>0;
CheckBox16->Checked = (DigitalOut & 128)>0;
```

ReadBackAnalogOut

Syntaxe:

```
void ReadBackAnalogOut(int *Buffer);
```

Parametry:

Buffer: Ukazatel na pole 32 bitových celých čísel, kam se načtou data.

Popis:

Hodnoty obou D-A převodníků jsou načteny zpět do pole 32 bitových celých čísel.

Příklad:

```
ReadBackAnalogOut(AnalogOut);

OutputAnalogChannel(1, 255 - VScrollBar1->Value);
OutputAnalogChannel(2, 255 - VScrollBar2->Value);
Label4->Text = (255 - VScrollBar1->Value).ToString();
Label5->Text = (255 - VScrollBar2->Value).ToString();
```

Deklarace funkcí a procedur v jiných programovacích jazycích

Visual Basic 6.0

```
Private Declare Sub ReadAll Lib "k8055d.dll" (ByVal data As Long)
Private Declare Function Version Lib "k8055d.dll" () As Long
Private Declare Function SearchDevices Lib "k8055d.dll" () As Long
Private Declare Function SetCurrentDevice Lib "k8055d.dll" (ByVal CardAddress As Long) As Long
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" (ByVal Data1 As Long, ByVal 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 Declare Function ReadBackDigitalOut Lib "k8055d.dll" () As Long
Private Declare Sub ReadBackAnalogOut Lib "k8055d.dll" (ByRef Buffer As Long)
```

Visual Basic 2008 Express

```
Private Declare Function OpenDevice Lib "k8055d.dll" (ByVal CardAddress As Integer) As Integer
Private Declare Sub CloseDevice Lib "k8055d.dll"
Private Declare Function Version Lib "k8055d.dll" () As Integer
Private Declare Function SearchDevices Lib "k8055d.dll" () As Integer
Private Declare Function SetCurrentDevice Lib "k8055d.dll" (ByVal CardAddress As Integer) As Integer
Private Declare Function ReadAnalogChannel Lib "k8055d.dll" (ByVal Channel As Integer) As Integer
Private Declare Sub ReadAllAnalog Lib "k8055d.dll" (ByRef Data1 As Integer, ByRef Data2 As Integer)
Private Declare Sub OutputAnalogChannel Lib "k8055d.dll" (ByVal Channel As Integer, ByVal Data As Integer)
Private Declare Sub OutputAllAnalog Lib "k8055d.dll" (ByVal Data1 As Integer, ByVal Data2 As Integer)
Private Declare Sub ClearAnalogChannel Lib "k8055d.dll" (ByVal Channel As Integer)
Private Declare Sub SetAllAnalog Lib "k8055d.dll" ()
Private Declare Sub ClearAllAnalog Lib "k8055d.dll" ()
Private Declare Sub SetAnalogChannel Lib "k8055d.dll" (ByVal Channel As Integer)
Private Declare Sub WriteAllDigital Lib "k8055d.dll" (ByVal Data As Integer)
Private Declare Sub ClearDigitalChannel Lib "k8055d.dll" (ByVal Channel As Integer)
Private Declare Sub ClearAllDigital Lib "k8055d.dll" ()
Private Declare Sub SetDigitalChannel Lib "k8055d.dll" (ByVal Channel As Integer)
Private Declare Sub SetAllDigital Lib "k8055d.dll" ()
Private Declare Function ReadDigitalChannel Lib "k8055d.dll" (ByVal Channel As Integer) As Boolean
Private Declare Function ReadAllDigital Lib "k8055d.dll" () As Integer
Private Declare Function ReadCounter Lib "k8055d.dll" (ByVal CounterNr As Integer) As Integer
Private Declare Sub ResetCounter Lib "k8055d.dll" (ByVal CounterNr As Integer)
Private Declare Sub SetCounterDebounceTime Lib "k8055d.dll" (ByVal CounterNr As Integer, ByVal DebounceTime As Integer)
Private Declare Function ReadBackDigitalOut Lib "k8055d.dll" () As Integer
Private Declare Sub ReadBackAnalogOut Lib "k8055d.dll" (ByRef Buffer As Integer)
```

Visual C# 2008 Express

```
[DllImport("k8055d.dll")]
public static extern int OpenDevice(int CardAddress);
```

```

[DllImport("k8055d.dll")]
public static extern void CloseDevice();
[DllImport("k8055d.dll")]
public static extern int ReadAnalogChannel(int Channel);
[DllImport("k8055d.dll")]
public static extern void ReadAllAnalog(ref int Data1, ref int Data2);
[DllImport("k8055d.dll")]
public static extern void OutputAnalogChannel(int Channel, int Data);
[DllImport("k8055d.dll")]
public static extern void OutputAllAnalog(int Data1, int Data2);
[DllImport("k8055d.dll")]
public static extern void ClearAnalogChannel(int Channel);
[DllImport("k8055d.dll")]
public static extern void SetAllAnalog();
[DllImport("k8055d.dll")]
public static extern void ClearAllAnalog();
[DllImport("k8055d.dll")]
public static extern void SetAnalogChannel(int Channel);
[DllImport("k8055d.dll")]
public static extern void WriteAllDigital(int Data);
[DllImport("k8055d.dll")]
public static extern void ClearDigitalChannel(int Channel);
[DllImport("k8055d.dll")]
public static extern void ClearAllDigital();
[DllImport("k8055d.dll")]
public static extern void SetDigitalChannel(int Channel);
[DllImport("k8055d.dll")]
public static extern void SetAllDigital();
[DllImport("k8055d.dll")]
public static extern bool ReadDigitalChannel(int Channel);
[DllImport("k8055d.dll")]
public static extern int ReadAllDigital();
[DllImport("k8055d.dll")]
public static extern int ReadCounter(int CounterNr);
[DllImport("k8055d.dll")]
public static extern void ResetCounter(int CounterNr);
[DllImport("k8055d.dll")]
public static extern void SetCounterDebounceTime(int CounterNr, int DebounceTime);
[DllImport("k8055d.dll")]
public static extern int Version();
[DllImport("k8055d.dll")]
public static extern int SearchDevices();
[DllImport("k8055d.dll")]
public static extern int SetCurrentDevice(int lngCardAddress);
[DllImport("k8055d.dll")]
public static extern int ReadBackDigitalOut();
[DllImport("k8055d.dll")]
public static extern void ReadBackAnalogOut(int[] Buffer);

```

Delphi

```

function SetCurrentDevice(CardAddress: integer): integer; stdcall; external 'K8055d.dll';
function OpenDevice(CardAddress: integer): integer; stdcall; external 'K8055d.dll';
function SearchDevices: integer; stdcall; external 'K8055d.dll';
function Version: integer; stdcall; external 'K8055d.dll';
procedure CloseDevice; stdcall; external 'K8055d.dll';
function ReadAnalogChannel(Channel: integer): integer; stdcall; external 'K8055d.dll';
procedure ReadAllAnalog(var Data1, Data2: integer); stdcall; external 'K8055d.dll';
procedure OutputAnalogChannel(Channel: integer; Data: integer); stdcall; external 'K8055d.dll';
procedure OutputAllAnalog(Data1: integer; Data2: integer); stdcall; external 'K8055d.dll';
procedure ClearAnalogChannel(Channel: integer); stdcall; external 'K8055d.dll';
procedure ClearAllAnalog; stdcall; external 'K8055d.dll';
procedure SetAnalogChannel(Channel: integer); stdcall; external 'K8055d.dll';
procedure SetAllAnalog; stdcall; external 'K8055d.dll';
procedure WriteAllDigital(Data: integer); stdcall; external 'K8055d.dll';
procedure ClearDigitalChannel(Channel: integer); stdcall; external 'K8055d.dll';
procedure ClearAllDigital; stdcall; external 'K8055d.dll';
procedure SetDigitalChannel(Channel: integer); stdcall; external 'K8055d.dll';
procedure SetAllDigital; stdcall; external 'K8055d.dll';
function ReadDigitalChannel(Channel: integer): Boolean; stdcall; external 'K8055d.dll';
function ReadAllDigital: integer; stdcall; external 'K8055d.dll';

```

```

function ReadCounter(CounterNr: integer): integer; stdcall; external 'K8055d.dll';
procedure ResetCounter(CounterNr: integer); stdcall; external 'K8055d.dll';
procedure SetCounterDebounceTime(CounterNr, DebounceTime: integer); stdcall; external 'K8055d.dll';
procedure SetPWM(Channel: integer; Data: integer; Frequency: integer); stdcall; external 'K8055d.dll';
function ReadBackDigitalOut: Longint; stdcall; external 'K8055d.dll';
procedure ReadBackAnalogOut(Buffer: Pointer); stdcall; external 'K8055d.dll';

```

Borland C++Builder

```

#ifdef __cplusplus
extern "C" {
#endif
#define FUNCTION __declspec(dllimport)
FUNCTION int __stdcall OpenDevice(int CardAddress);
FUNCTION void __stdcall CloseDevice();
FUNCTION int __stdcall ReadAnalogChannel(int Channel);
FUNCTION void __stdcall ReadAllAnalog(int *Data1, int *Data2);
FUNCTION void __stdcall OutputAnalogChannel(int Channel, int Data);
FUNCTION void __stdcall OutputAllAnalog(int Data1, int Data2);
FUNCTION void __stdcall ClearAnalogChannel(int Channel);
FUNCTION void __stdcall ClearAllAnalog();
FUNCTION void __stdcall SetAnalogChannel(int Channel);
FUNCTION void __stdcall SetAllAnalog();
FUNCTION void __stdcall WriteAllDigital(int Data);
FUNCTION void __stdcall ClearDigitalChannel(int Channel);
FUNCTION void __stdcall ClearAllDigital();
FUNCTION void __stdcall SetDigitalChannel(int Channel);
FUNCTION void __stdcall SetAllDigital();
FUNCTION bool __stdcall ReadDigitalChannel(int Channel);
FUNCTION int __stdcall ReadAllDigital();
FUNCTION int __stdcall ReadCounter(int CounterNr);
FUNCTION void __stdcall ResetCounter(int CounterNr);
FUNCTION void __stdcall SetCounterDebounceTime(int CounterNr, int DebounceTime);
FUNCTION int __stdcall Version();
FUNCTION int __stdcall SearchDevices();
FUNCTION int __stdcall SetCurrentDevice(int CardAddress);
FUNCTION int __stdcall ReadBackDigitalOut();
FUNCTION void __stdcall ReadBackAnalogOut(int *Buffer);
#ifdef __cplusplus
}
#endif

```