



Experimentální USB rozhraní



VM110N (modul)



Obj. č. 19 11 37

Vážený zákazníku,

děkujeme Vám za Vaši důvěru a za nákup experimentální USB rozhraní.

Tento návod k obsluze je součástí výrobku. Obsahuje důležité pokyny k uvedení výrobku do provozu a k jeho obsluze. Jestliže výrobek předáte jiným osobám, dbejte na to, abyste jim odevzdali i tento návod k obsluze.

Ponechejte si tento návod, abyste si jej mohli znovu kdykoliv přečíst!

Tato experimentální karta je rozhraní s 16. digitálními IN/OUT kanály. Obsahuje navíc 8 analogových výstupů s 6-bitovým rozlišením a 4 analogové vstupy s 8-bitovým rozlišením. V případě potřeby většího množství digitálních výstupů, je možné využít analogové výstupy s omezením pro minimální a maximální výstupní napětí. Proto je velmi jednoduché sledovat stav otočného regulátoru a průběhy různého napětí. Vezměte však na vědomí, že tyto zvláštní vstupní a výstupní kanály nejsou vzájemně opticky odděleny. Počet vstupů a výstupů je možné dále rozšířit po připojení dalších karet (maximálně 4). Každou kartu představuje její vlastní identifikace prostřednictvím dvoupolového přepínače DIP-SW1 (více v tabulce číslování kanálů). Kartu připojte do počítače prostřednictvím tiskového portu (bez potřeby instalace zvláštního tiskového portu). Celkem 3 trasy z tohoto portu jsou: „**Select**“ (pin 13), „**Autofeed**“ (pin 14) a „**Select in**“ (pin 17). Komunikace mezi počítačem a kartou probíhá prostřednictvím sériového linku. Pomocí jednoho pinu (Select in) je generován hodinový signál, druhým linkem (Autofeed) datový výstup a třetím (Select) vstup dat. Veškerá komunikace je založena na Dynamic Link Library (DDL) K8D.DLL. V návodu naleznete veškeré DLL funkce a procesy, které jsou dostupné v programové aplikaci. Použití DLL umožňuje tvorbu vlastních aplikací (v systému Windows 9x, NT nebo XP) v jazycích Visual C++, Delphi, Visual Basic nebo jině 32-bitové Windows aplikaci podporující DLL. Navíc je při tom automaticky zajišťován komunikační protokol. V návodu jsou popsány veškeré procesy, funkce a proměnné použité „K8D.DLL“. Uvedený příklad programování můžete využít pro vlastní zdokonalení a získání zkušeností s konstrukcí aplikací programů. Příklady jsou napsané v jazyce Delphi. V závěru návodu naleznete všechny potřebné údaje a operace pro jazyky Delphi a Visual Basic.

Tabulka pro nastavení „SW1“

Číslo karty	Číslo čipu	Číslo kanálu
0 (OFF-OFF)	IO-čip č.: 0	IO-kanály: 1...8
	IO-čip č.: 1	IO-kanály: 9...16
	DAC-čip č.: 0	DAC-kanály: 1...8
	AC-čip č.: 0	AD-kanály: 1...4 DA-kanály: 1
1 (OFF-ON)	IO-čip č.: 2	IO-kanály: 17...24
	IO-čip č.: 3	IO-kanály: 25...32
	DAC-čip č.: 1	DAC-kanály: 9...16
	AD-čip č.: 1	AD-kanály: 5...8 DA-kanály: 2
2 (ON-OFF)	IO-čip č.: 4	IO-kanály: 33...40
	IO-čip č.: 5	IO-kanály: 41...48
	DAC-čip č.: 2	AD-kanály: 9...12 DA-kanály: 3
3 (ON-ON)	IO-čip č.: 6	IO-kanály: 49...56
	IO-čip č.: 7	IO-kanály: 57...64
	DAC-čip č.: 3	DAC-kanály: 25...32
	AD-čip č.: 3	AD-kanály: 13...16 DA-kanály: 4

Přehled „K8D.DLL“ proměnných

Proměnné	Typ	Vstupní hodnota	Popis
DA	Pole (1..4) celého čísla	0	Obsahuje data (hodnota mezi 0 a 255) čtvrtého, 8-bitového digitálně – analogového konvertoru
DAC	Pole (1..32) celého čísla	0	Obsahuje data (hodnota mezi 0 – 63) 32. 6-bitového digitálně-analog. převodového kanálu
IOconfig	Pole (0..7) celého čísla	\$OFF	Každý bit obsahuje stav příslušného kanálu 8. IO portů. Bit high (1) = kanále je zapnutý, bit low (0) = kanál je vypnutý

Konstanta	Hodnota	Popis
MaxIOcard	3	Nejvyšší možná adresa karty rozhraní
MaxIOchip	7	Nejvyšší možné Input/Output číslo čipu
MaxIOchannel	64	Nejvyšší možný Input/Output kanál
MaxDACchannel	32	Nejvyšší možný kanál 6-bitového DA převodníku
MaxADchannel	16	Nejvyšší možný AD kanál
MaxDAchannel	4	Nejvyšší možný DA 8-bitový analogový kanál

Přehled „K8D.DLL“ postupů a funkcí

8-bit AD konverze	
ReadADchannel (Channelno)	Čtení stavu analogového vstupního kanálu
8-bit DA konverze	
OutputDAchannel (Channelno, Data)	Nastavení analogového výstupního kanálu podle příslušných dat
ClearDAchannel (Channelno)	Nastavení analogového výstupního kanálu na minimum
ClearAllDA	Nastavení všech výstupních analogových kanálů na minimum
SetDAchannel (Channelno)	Nastavení analogového výstupního kanálu na maximum
SetAllDA	Nastavení všech výstupních analogových kanálů na maximum
6-bit konverze D-A	
OutputDACchannel (Channelno, Data)	Nastavení výstupního analogového kanálu podle konkrétních dat

ClearDACchannel(Channelno)	Nastavení výstupního analogového kanálu na minimum
ClearDACchip(Chipno)	Nastavení 8. výstupních analogových kanálů DAC čipu na minimum
ClearAllDAC	Nastavení všech výstupních analogových kanálů na minimum
SetDACchannel(Channelno)	Nastavení výstupního analogového kanálu na maximum
SetDACchip(Chipno)	Nastavení 8. výstupních analogových kanálů DAC čipu na maximum
SetAllDAC	Nastavení všech výstupních analogových kanálů na maximum
IO konfigurace	
ConfigAllIOasInput	Konfigurace všech IO-kanálů jako vstupních
ConfigIOchipAsInput(Chipno)	Konfigurace všech IO-kanálů IO-čipu jako vstupních
ConfigIOchannelAsInput(Channelno)	Konfigurace IO-kanálu jako vstupního
ConfigAllIOasOutput	Konfigurace všech IO-kanálů jako výstupních
ConfigIOchipAsOutput(Chipno)	Konfigurace všech IO-kanálů IO-čipu jako výstupních
ConfigIOchannelAsOutput(Channel)	Konfigurace IO-kanálu jako výstupního
Nastavení IO dat a IO proměnných (skutečný stav IO-kanálů se nemění)	
UpdateIOdataArray(Chipno,Data)	Nastavení výstupního stavu podle konkrétních dat (vstupy se nemění)
ClearIOchArray(Channelno)	Vynulování výstupního stavu vybraných kanálů (low)
ClearIOdataArray(Chipno)	Vynulování výstupního stavu kanálů IO-čipu (low)
SetIOchArray(Channelno)	Nastavení výstupního stavu vybraného kanálu (high)
SetIOdataArray(Chipno)	Nastavení výstupního stavu kanálů IO-čipu (high)
Výstupní procesy	
IOoutput(Chipno,Data)	Nastavení výstupů IO-čipu podle konkrétních dat (vstupy se nijak nemění)
UpdateIOchip(Chipno)	Nastavení výstupů IO-čipu podle stavu proměnné „IOdata“
UpdateAllIO	Nastavení všech výstupů v závislosti na stavu v proměnné „IOdata“
ClearIOchannel(Channelno)	Vynulování výstupního kanálu
ClearIOchip(Chipno)	Vynulování výstupních kanálů IO-čipu
ClearAllIO	Vynulování všech výstupních kanálů
SetIOchannel(Channelno)	Nastavení výstupního kanálu
SetIOchip(Chipno)	Nastavení výstupních kanálů IO-čipu
SetAllIO	Nastavení všech výstupních kanálů
Vstupní procesy a funkce	
ReadIOchannel(Channelno)	Čtení stavu vstupního kanálu
ReadIOchip(Chipno)	Čtení stavu vstupních kanálů IO-čipu
ReadIOconfigArray(Buffer)	Načtení dat konfigurace IO z DLL do aplikačního programu
ReadIOdataArray(Buffer)	Načtení dat IO stavu z DLL do aplikačního programu
ReadDACarray(Buffer)	Načtení dat DAC z DLL do aplikačního programu
ReadDAarray(Buffer)	Načtení dat DA z DLL do aplikačního programu
Hlavní procesy	
SelectI2CprinterPort(Printer_no)	Výběr komunikačního portu
Start_K8000	Otevření linku k K8000 zařízení
Stop_K8000	Uzavření linku do K8000 zařízení

Start_K8000

Syntaxe

```
PROCEDURE Start_K8000;
```

Popis

Spuštění inicializačních procesů karty K8000. Načtení ovladačů potřebných pro komunikaci přes LPT port. Tento proces musí být spuštěn na začátku aplikačního programu.

Příklad

```
BEGIN
    Start_K8000;
END;
```

Stop_K8000

Syntaxe

```
PROCEDURE Stop_K8000;
```

Popis

Deaktivace komunikačního procesu K8000 a ovladačů pro komunikaci LPT portu. Jedná se o poslední proces před ukončením programu.

Příklad

```
BEGIN
    Stop_K8000;
END;
```

SelectI2CprinterPort

Syntaxe

```
PROCEDURE SelectI2CprinterPort(Printer_no: Longint);
```

Parametr

Printer_no: Hodnota mezi 0 a 2 poskytována tiskovým portem, ke kterému je karta rozhraní připojena.

0: adresa tiskového portu je 0BC (hex)

1: adresa tiskového portu je 378 (hex), zpravidla adresa LPT1

2: adresa tiskového portu je 278 (hex), zpravidla adresa LPT2

Výsledek

Komunikace mezi PC a K8000 proběhne prostřednictvím nastavené adresy LPT portu.

Popis

Tiskový port musí být specifikován na začátku programu. V opačném případě nebude správně probíhat vzájemná komunikace. Výchozí výběr je LPT1 s možností změny v nastavení.

Příklad

```
BEGIN
    SelectI2CprinterPort(1);
    // LPT1 address on mainboard is set to 378
END;
```

ReadADchannel

Syntaxe

```
FUNCTION ReadADchannel(Channel_no: Longint):Longint;
```

Parametr

Channel_no: Hodnota mezi 1 a 16 korespondující s AD kanálem, jehož stav má být přečten.

Výsledek

AD: Korespondující „AD“ data jsou přečtena v závislosti na stavu AD vstupu.

Popis

Vstupní napětí vybraného 8-bitového analogovo-digitálního konverzního kanálu je převedeno na hodnotu mezi 0 a 255 a zaneseno do příslušné „AD“ proměnné.

Příklad

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

ReadADchannel

Syntaxe

```
FUNCTION ReadADchannel(Channel_no: Longint):Longint;
```

Parametr

Channel_no: Hodnota mezi 1 a 16 korespondující s AD kanálem, jehož stav má být přečten.

Výsledek

AD: Přečtena jsou korespondující „AD“ data v závislosti na stavu AD výstupu.

Popis

Vstupní napětí zvoleného 8-bitového analog.-digitálního konvertoru jsou převedena na hodnotu mezi 0 a 255 a zanesena do příslušné „AD“ proměnné.

Příklad

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

OutputDchannel

Syntaxe

```
PROCEDURE OutputDchannel(Channel_no: Longint; Data: Longint);
```

Parametr

Channel_no: Hodnota mezi 1 a 4 korespondující s 8-bitovým číslem DA kanálu, jehož data mají být upravena.

Data: Hodnota mezi 0 a 255, která má být odeslána do 8-bitového DA konvertoru.

Výsledek

DA:

Proměnná data „DA“ vybraného kanálu jsou nastavena v závislosti na datech, která mají být odeslána.

Popis

8-bitový DA konverzní trasa je změněna podle nových dat. Data tak představují specifickou hodnotu napětí. Hodnota 0 představuje minimální výstupní napětí (0 V) a hodnota 255 maximální výstupní napětí (Vmax) nastavené pro kartu rozhraní. Hodnotu „Data“, která se nachází mezi těmito dvěma hodnotami, je možné formulovat: Data x Vmax/255.

Příklad

```
BEGIN
    OutputDchannel(1,127);
    // DA channel 1 is at 1/2 Vmax
END;
```

ClearDchannel

Syntaxe

```
PROCEDURE ClearDchannel(Channel_no: Longint);
```

Parametr

Channel_no: Hodnota mezi 1 a 4 korespondující s 8-bitovým DA kanálem, ve kterém mají být data vynulována.

Výsledek

DA: Proměnná data „DA“ zvoleného DA kanálu jsou nastavena na minimum (0). Vybrané DA kanály jsou nastaveny na minimální výstupní napětí (0 V).

Popis

Zvolený 8-bitový DA konverzní kanál je nastaven na minimální výstupní napětí (0 V).

Příklad

```
BEGIN
    ClearDchannel(1);
    // DA channel 1 is at Vmin
END;
```

ClearAllDA

Syntaxe

```
PROCEDURE ClearAllDA;
```

Výsledek

Všechny DA kanály jsou nastavené na minimální výstupní napětí (0 V).

Popis

Všechny DA kanály 8-bitového DA konvertoru jsou nastaveny na minimální výstupní napětí (0 V).

Příklad

```
BEGIN
    ClearAllDA;
    // All DA channels 1..4 are at Vmin
END;
```

SetDchannel

Syntaxe

```
PROCEDURE SetDchannel(Channel_no: TDchannel);
```

Parametr

Channel_no: Hodnota mezi 1 a 4, která koresponduje s 8-bitovým DA kanálem, ve kterém mají být data nastavena na maximum.

Výsledek

Vybraný DA kanál je nastaven na maximální výstupní napětí.

Popis

Zvolený 8-bitový DA konverzní kanál je nastaven na maximální výstupní napětí.

Příklad

```
PROGRAM Set_DA_channel;
USES I2C, WinCrt;
BEGIN
    SetDAchannel(1);
    Writeln('Set DA channel 1 at Vmax');
END;
```

SetAllDA

Syntaxe

```
PROCEDURE SetAllDA;
```

Výsledek

Všechny DA kanály jsou nastaveny na maximální výstupní napětí.

Popis

Všechny DA kanály 8-bitového DA konvertoru jsou nastaveny na maximální výstupní napětí.

Příklad

```
BEGIN
    SetAllDA;
    // All DA channels 1...4 are at Vmax
END;
```

OutputDACchannel

Syntaxe

```
PROCEDURE OutputDACchannel(Channel_no: Longint; Data: Longint);
```

Parameters

Channel_no: Hodnota mezi 1 a 32 korespondující s 6-bitovým DAC kanálem ve kterém mají být data upravena.

Data: Hodnota mezi 0 a 63 k odeslání do 6-bitového DA konvertoru.

Výsledek

Aktualizace dat DAC kanálu.

Popis

6-bitový DA konverzní kanál je změněn v závislosti na nových datech. Data představují konkrétní napětí. Hodnota 0 představuje pro kartu minimální výstupní napětí (Vmin) a 63 maximální výstupní napětí (Vmax). Hodnotu „Data“ mezi těmito dvěma hodnotami je možné formulovat jako: $V_{min} + Data \times (V_{max} - V_{min}) / 63$

Příklad

```
BEGIN
    OutputDACchannel(1,21);
    // DAC channel 1 is at Vmin + 1/3(Vmax-Vmin)
END;
```

ClearDACchannel

Syntaxe

```
PROCEDURE ClearDACchannel(Channel_no: Longint);
```

Parametr

Channel_no: Hodnota mezi 1 a 32 korespondující s 6-bitovým DAC kanálem ve kterém mají být data vynulována.

Výsledek

Zvolený DAC kanál je nastavený na minimální výstupní napětí (Vmin).

Popis

Vybraný 6-bitový DA konverzní kanál je nastavený na minimální výstupní napětí (Vmin).

Example

```
BEGIN
    ClearDACchannel(2);
    // DAC channel 2 is at Vmin
END;
```

ClearDACchip

Syntaxe

```
PROCEDURE ClearDACchip(Chip_no: Longint);
```

Parametr

Chip_no: Hodnota mezi 0 a 3 korespondující s adresou 6-bitového DAC čipu, ve kterém má být všech 8 kanálů nastaveno na minimální výstupní napětí.

Výsledek

Celkem 8 DAC kanálů určeného DAC čipu bude nastaveno na minimální výstupní napětí.

Popis

Celkem 8 DAC kanálů vybraného 6-bitového DAC čipu je nastaveno na minimum (Vmin).

Příklad

```
BEGIN
    ClearDACchip(0);
    // DAC channels 1...8 are at Vmin
END;
```

ClearAllDAC

Syntaxe

```
PROCEDURE ClearAllDAC;
```

Výsledek

Všechny DAC kanály jsou nastaveny na minimální výstupní napětí.

Popis

Všechny DAC kanály 6-bitových DA konvertorů jsou nastaveny na minimální výstupní napětí (Vmin).

Příklad

```
BEGIN
    ClearAllDAC;
    // All DAC channels 1...32 are at Vmin
END;
```

SetDACchannel

Syntaxe

```
PROCEDURE SetDACchannel(Channel_no: Longint);
```

Parametr

Channel_no: Hodnota mezi 1 a 32 korespondující s 6-bitovým DAC kanálem ve kterém mají být data nastaveny na maximum.

Výsledek

Zvolený DAC kanál je nastavený na maximální výstupní napětí.

Popis

Vybraný 6-bitový DAC kanál je nastavený na maximální výstupní napětí.

Příklad

```
BEGIN
    SetDACchannel(3);
    // Set DAC channel 3 at Vmax
END;
```

SetDACchip

Syntaxe

```
PROCEDURE SetDACchip(Chip_no: Longint);
```

Parametr

Chip_no: Hodnota mezi 0 a 3 korespondující s adresou 6-bitového DAC čipu ve kterém má být celkem 8 kanálů nastaveno na maximální výstupní napětí.

Výsledek

Celkem 8 DAC kanálů příslušného DAC čipu je nastaveno na maximální výstupní napětí.

Popis

Celkem 8 DAC kanálů zvoleného 6-bitového DAC čipu je nastaveno na maximální výstupní napětí (Vmax). Proměnná data „DAC“ příslušných DAC kanálů jsou obdobně upravena.

Příklad

```
BEGIN
    SetDACchip(0);
    // DAC channels 1...8 are at Vmax
END;
```

SetAllDAC

Syntaxe

```
PROCEDURE SetAllDAC;
```

Výsledek

Všechny DAC kanály jsou nastaveny na maximální výstupní napětí.

Popis

Všechny DAC kanály 6-bitového DAC jsou nastaveny na maximální výstupní napětí (Vmax).

Příklad

```
BEGIN
    SetAllDAC;
    // All DAC channels 1...32 are at Vmax
END;
```

ConfigAllIOasInput

Syntaxe

```
PROCEDURE ConfigAllIOasInput;
```

Výsledek

IOconfig: Proměnná „IOconfig“ pro všechny I/O porty jsou nastaveny na hodnotu 255.

Popis

Všechny digitální I/O kanály (1...64) jsou konfigurovány jako vstupy. Každý IO čip (0...7) zahrnuje proměnnou, která zajišťuje návrat IO pinu ve formě dat (bitů), která nemají vysokou úroveň (high). Pokud je takový bit na úrovni high (1), představuje vstup. Zápis do IO kanálu pak nezpůsobí žádné změny. Stav IO kanálů může být určující pouze externím signálem.

Příklad

```
BEGIN
    ConfigAllIOasInput;
    // All IO channels are now configured as inputs
END;
```

ConfigIOchipAsInput

Syntaxe

```
PROCEDURE ConfigIOchipAsInput(Chip_no: Longint);
```

Parametr

Chip_no: Hodnota mezi 0 a 7 korespondující s nastavením adresy Input/Output čipu ve kterém mají být kanály konfigurovány jako vstupy.

Výsledek

IOconfig: Proměnná „IOconfig“ příslušného Input/Output čipu je nastavena na hodnotu 255.

Popis

Celkem 8 digitálních vstupních/výstupních kanálů vybraného IO čipu je konfigurováno jako vstupy. Každý IO čip (0...7) obsahuje proměnnou, kterou je konfigurace každého IO pinu vrácena ve formě dat (bit) jež nejsou v úrovni high. Pokud je takový bit high (1) jedná se o vstup. Stav IO kanálů může být určen pouze externím signálem.

Příklad

```
BEGIN
    ConfigIOchipAsInput(0);
    // The 8 channels from IO Chip 0 are now configured as inputs
END;
```

ConfigIOchannelAsInput

Syntaxe

```
PROCEDURE ConfigIOchannelAsInput(Channel_no: Longint);
```

Parametr

Channel_no: Hodnota mezi 1 a 64 korespondující s IO kanálem, který má být konfigurován jako vstup.

Výsledek

IOconfig: Počet kanálů, IO čipů a bitů je určen v případě nastavení proměnné „IOconfig“ do úrovně high (1).

Popis

Zvolený IO kanál je konfigurován jako vstup, zatímco konfigurace ostatních kanálů zůstává nezměněna. To je zajištěno správným bitem (1) v konfiguraci proměnné příslušného IO čipu. Stav IO kanálu může být zjištěn pouze externím signálem.

Příklad

```
BEGIN
    ConfigIOchannelAsInput(1);
    // IO channel 1 is now configured as input';
END;
```

ConfigAllIOasOutput

Syntaxe

```
PROCEDURE ConfigAllIOasOutput;
```

Výsledek

IOconfig: Proměnná „IOconfig“ všech IO portů je nastavena na low (0).

Popis

Všechny digitální IO kanály (1..64) jsou konfigurovány jako výstupní. Každý IO čip (0..7) obsahuje proměnnou, kterou je konfigurace každého IO pinu vrácena ve formě dat (bit), které nejsou v úrovni high. Pokud je tento bit low (0) představuje výstup. Stav IO kanálu je určován hodnotou v těchto kanálech.

Příklad

```
BEGIN
    ConfigAllIOasOutput;
    // All IO channels are now configured as outputs
END;
```

ConfigIOchipAsOutput

Syntaxe

```
PROCEDURE ConfigIOchipAsOutput(Chip_no: Longint);
```

Parametr

Chip_no: Hodnota mezi 0 a 7 korespondující s nastavením adresy IO čipu, jehož všechny kanály mají být konfigurovány jako výstupy.

Výsledek

IOconfig: Proměnná „IOconfig“ příslušného Input/Output čipu je nastavena do úrovně low (0).

Popis

Celkem 8 digitální IO kanálů vybraného IO čipu je konfigurováno jako výstupy. Každý IO čip (0..7) zahrnuje proměnnou, kterou je konfigurace každého IO pinu vrácena ve formě dat (bit), které nejsou v úrovni high. Pokud je tento bit low (0) představuje výstup. Stav těchto IO kanálů je určován hodnotami zapsanými v těchto kanálech.

Příklad

```
BEGIN
    ConfigIOchipAsOutput(1);
    // The 8 channels from IO Chip 1 are now configured as outputs
END;
```

ConfigIOchannelAsOutput

Syntaxe

```
PROCEDURE ConfigIOchannelAsOutput(Channel_no: Longint);
```

Parametr

Channel_no: Hodnota mezi 1 a 64 korespondující s IO kanálem, který má být konfigurován jako výstup.

Výsledek

IOconfig: Počet kanálů, IO čipů a bitů je určován v případě nastavení proměnné „IOconfig“ do úrovně low (0).

Popis

Vybraný IO kanál je konfigurován jako výstup, zatímco konfigurace ostatních kanálů zůstává nezměněna. To je způsobeno bitem v úrovni low (0) konfigurace proměnné příslušného IO čipu. Stav tohoto IO kanálu je určován hodnotou zapsanou v tomto kanálu.

Příklad

```
BEGIN
    ConfigIOchannelAsOutput(2);
    // IO channel 2 is now configured as output
END;
```

UpdateIOdataArray

Syntaxe

```
PROCEDURE UpdateIOdataArray(Chip_no: Longint; Data: Longint);
```

Parametry

Chip_no: Hodnota mezi 0 a 7 korespondující s nastavením adresy IO čipu, ve kterém mají být data změněna.

Data: Hodnota mezi 0 a 255, která mají být odeslána do IO portu (8 kanálů).

Výsledek

IOdata: Proměnná „Odata“ vybraného čipu jsou aktualizována novými daty, stav vstupních kanálů přitom zůstává nezměněn.

Popis

Stav IO portů je udržován v proměnné „IO data“. Každý bit této proměnné koresponduje se stavem IO kanálu. Pokud stav 8 kanálů IO čipu má být změněn, není zapotřebí nastavení dat v proměnné „IOdata“, protože stav vstupních kanálů by nekorespondoval se skutečným stavem. Pro předcházení možných konfliktů je proto nezbytné vždy použít tento proces. Proměnné „IOdata“ jsou uložena pouze ve vyrovnávací (buffer) paměti. Pokud je tato hodnota upravena, dochází k okamžité změně stavu. Stav se mění pouze v případě, že jeho hodnota je odeslána do IO čipu.

Příklad

```
BEGIN
    ConfigIOchipAsOutput(0);
    UpdateIOdataArray(0,204);
    // 204 is in binary format 11001100
    // this sets channels 3,4,7,8 and clears channels 1,2,5,6
END;
```

ClearIOchArray

Syntaxe

```
PROCEDURE ClearIOchArray(Channel_no: Longint);
```

Parametr

Channel_no: Hodnota mezi 1 a 64 korespondující s IO kanálem, který má být vynulován.

Výsledek

IOdata: Pokud je vybraný kanál výstupem, pak správný bit v proměnné „IOdata“ korespondujícího čipu je nastaven do úrovně low (0).

Popis

Stav IO portů je udržován v proměnné „IO data“. Každý bit této proměnné koresponduje se stavem IO kanálu. V případě, že výstup IO čipu má být vynulován, není zapotřebí ihned nastavovat korespondující bit v proměnné „IOdata“ na low (0). Pokud byl vybraný kanál konfigurován jako vstup, potom jeho stav nebude korespondovat se skutečným stavem. Jako prevence před takovými konflikty je nezbytné vždy použít tento proces. Proměnná „IOdata“ jsou pouze v buffer paměti. Po úpravě hodnot nedochází k okamžité změně stavu. K tomu dochází až po odeslání hodnoty do IO čipu.

Example

```
BEGIN
  ConfigIOchipAsOutput(0);
  ConfigIOchannelAsInput(1); {channel 1 configured as input}
  ConfigIOchannelAsInput(2); {channel 2 configured as input}
  ConfigIOchannelAsInput(3); {channel 3 configured as input}
  ConfigIOchannelAsInput(4); {channel 4 configured as input}
  ClearIOchArray(2); // Clear channel 2 from IOdata array
  // The status of the inputs stays unchanged
  ClearIOchArray(8); // Clear channel 8 from IOdata array
END;
```

ClearIOdataArray

Syntaxe

```
PROCEDURE ClearIOdataArray(Chip_no: Longint);
```

Parametr

Chip_no: Hodnota mezi 0 a 7 korespondující s nastavením adresy IO čipu, ve kterém mají být výstupy vynulovány.

Výsledek

IOdata: Bity proměnné „IOdata“ vybraného čipu jsou nastaveny do úrovně low (0), ale bity, které korespondují se vstupy zůstávají nezměněny.

Popis

Stav IO portů je udržován v proměnné „IOdata“. Každý bit této proměnné koresponduje se stavem IO kanálu. Pokud výstupy IO čipu mají být vynulovány, nedoporučuje se ihned nastavovat proměnnou „IOdata“ na 0, protože stav vstupních kanálů nemusí korespondovat se skutečným stavem. Pro zabránění konfliktům je nutné použít tento proces. Proměnná „IOdata“ je pouze v buffer memory. Pokud dojde ke změně hodnot na výstupech, neměňte tento stav okamžitě. K tomu může dojít pouze, pokud jsou hodnoty odesílány do IO čipu.

Example

```
BEGIN
  ConfigIOchipAsOutput(0);
  ConfigIOchannelAsInput(1); {channel 1 configured as input}
  ConfigIOchannelAsInput(2); {channel 2 configured as input}
  ConfigIOchannelAsInput(3); {channel 3 configured as input}
  ConfigIOchannelAsInput(4); {channel 4 configured as input}
  ClearIOdataArray(0); // Clear channels 1 to 8 from IOdata array
  // The status of the inputs stays unchanged
END;
```

SetIOchArray

Syntaxe

```
PROCEDURE SetIOchArray(Channel_no: Longint);
```

Parameter

Channel_no: Hodnota mezi 1 a 64 korespondující s IO kanálem, který má být nastaven.

Výsledek

IOdata: Pokud je zvolený kanál výstupním, potom správný výstupní bit v proměnné „IOdata“ korespondujícího čipu, je nastaven do úrovně high (1).

Popis

Stav IO portů je udržován v proměnné „IOdata“. Každý bit této proměnné koresponduje se stavem IO kanálu. Jestliže výstup IO čipu má být upravován, není vhodné okamžitě nastavovat korespondující bit proměnné „IOdata“ do úrovně high. Pokud zvolený kanál má být konfigurován jako vstupní, potom stav nemusí odpovídat jeho skutečnému stavu. Tento proces je proto vždy nezbytné použít. Proměnná „IOdata“ jsou ukládány pouze v buffer paměti. Pokud dojde k vlastní změně hodnoty na výstupu, neměňte okamžitě tento stav. K tomu může dojít pouze, je-li hodnota odeslána do IO čipu.

Příklad

```
BEGIN
  ConfigIOchipAsOutput(0);
  ConfigIOchannelAsInput(1); {channel 1 configured as input}
  ConfigIOchannelAsInput(2); {channel 2 configured as input}
  ConfigIOchannelAsInput(3); {channel 3 configured as input}
  ConfigIOchannelAsInput(4); {channel 4 configured as input}
  SetIOchArray(1); // Set channels 1 from IOdata array
  SetIOchArray(5); // Set channels 5 from IOdata array
  // The status of the input channel 1 stays unchanged
END;
```

SetIOdataArray

Syntaxe

```
PROCEDURE SetIOdataArray(Chip_no: Longint);
```

Parametr

Chip_no: Hodnota mezi 0 a 7 korespondující s nastavením adresy IO čipu, ve kterém mají být výstupy upravovány.

Výsledek

IOdata: Bity proměnné „IOdata“ zvoleného čipu jsou nastaveny do úrovně high (1), ale bity korespondující se vstupy zůstanou nezměněny.

Popis

Stav IO portů je udržován v proměnné „IOdata“. Každý bit této proměnné koresponduje se stavem IO kanálu. V případě, že jsou výstupy IO čipu konfigurovány, není nezbytné okamžité načtení 255 v proměnné „IOdata“, vzhledem k tomu, že stav vstupních kanálů pak nemusí korespondovat se skutečným stavem. Tento proces je nutné ponechat proběhnout tak, aby nedocházelo ke konfliktům. Proměnné „IOdata“ jsou ukládány pouze v buffer paměti. Při probíhajících změnách výstupních hodnot nemějte tento stav. K tomu však dochází pouze, pokud jsou hodnoty odesílány do IO čipu. Proměnné „IO“ příslušného IO-čipu jsou rovněž upravovány.

Example

```
BEGIN
  ConfigIOchipAsOutput(0);
  ConfigIOchannelAsInput(1); {channel 1 configured as input}
  ConfigIOchannelAsInput(2); {channel 2 configured as input}
  ConfigIOchannelAsInput(3); {channel 3 configured as input}
  ConfigIOchannelAsInput(4); {channel 4 configured as input}
  SetIOdataArray(0); // Set channels 1 to 8 from IOdata array
// The status of the inputs stays unchanged
END;
```

IOoutput

Syntaxe

```
PROCEDURE IOoutput(Chip_no: Longint; Data: Longint);
```

Parametry

Chip_no: Hodnota mezi 0 a 7 korespondující s nastavením adresy IO čipu, ve kterém mají být data upravována.

Data: Hodnota mezi 0 a 255 odesílaná do IO portu (8 kanálů).

Výsledek

IOdata: Proměnná „IOdata“ vybraného čipu jsou aktualizována novými daty a tím zůstává stav vstupních kanálů beze změn. Výstupy vybraného IO čipu jsou aktualizovány.

Popis

Kanály zvoleného IO čipu, které byly konfigurovány jako výstupy, jsou aktualizovány stavem korespondujících bitů v parametru data. Úroveň high (1) představuje nastavený výstup a nízká úroveň low (0) znamená, že výstup je vynulován. Stav na vstupech zůstává beze změny.

Příklad

```
BEGIN
  ConfigIOchipAsOutput(1);
  IOoutput(1,128);
  // Output channels 9..15 are off, Output channel 16 is on
END;
```

UpdateIOchip

Syntaxe

```
PROCEDURE UpdateIOchip(Chip_no: Longint);
```

Parametr

Chip_no: Hodnota mezi 0 a 7, která koresponduje s nastavením adresy IO čipu, v němž mají být výstupy upraveny.

Výsledek

Všechny výstupy vybraného IO čipu jsou upraveny v závislosti na stavu korespondujících bitů v proměnné „IOdata“.

Popis

Všechny kanály zvoleného IO čipu, které byly konfigurovány jako výstupy, jsou upraveny podle stavu korespondujících bitů v proměnné „IOdata“. Vysoká úroveň high (1) představuje nastavený výstup, nízká úroveň low (0) znamená, že výstup byl vynulován. Stav vstupů zůstává beze změny.

Příklad

```
BEGIN
  ConfigIOchipAsOutput(1);
  UpdateIOdataArray(1,64); // channels 9..14 & 16 off, 15 on
  UpdateIOchip(1);
  // Output channels 9..14 & 16 are off, Output channel 15 is on
END;
```

UpdateAllIO

Syntaxe

```
PROCEDURE UpdateAllIO;
```

Výsledek

Všechny výstupy jsou upraveny podle stavu korespondujících bitů v proměnné „IOdata“.

Popis

Všechny kanály IO čipů, které byly konfigurovány jako výstupy, jsou upraveny vzhledem ke stavu korespondujících bitů v proměnné „IOdata“. Vysoká úroveň high (1) představuje nastavený výstup, nízká úroveň low (0) znamená, že výstup je vynulován. Stav vstupů zůstává beze změny.

Příklad

```
BEGIN
  ConfigAllIOasOutput;
  UpdateIOdataArray(0,1); // channel 1 on
  UpdateIOdataArray(1,128); // channel 16 on
  UpdateAllIO;
  // Output channel 1 & 16 are on, Output channels 2..15 are off
END;
```

ClearIOchannel

Syntaxe

```
PROCEDURE ClearIOchannel(Channel_no: Longint);
```

Parametr

Channel_no: Hodnota mezi 1 a 64, která koresponduje IO kanálu, jež má být vynulována.

Výsledek

IOdata: Pakliže je zvolený kanál výstupní, potom správný bit v proměnné „IOdata“ odpovídajícího čipu je nastaven na nízkou úroveň low (0).

Popis

Pakliže byl vybraný kanál konfigurován jako výstup, dojde k jeho vynulování. Stav vstupů zůstává beze změn. Proměnné „IOdata“ a „IO“ jsou přizpůsobeny vzhledem k tomuto novému stavu.

Příklad

```
BEGIN
  ConfigIOchannelAsOutput(9);
  ClearIOchannel(9); // Output channel 9 is off
END;
```


ClearIOchip

Syntaxe

```
PROCEDURE ClearIOchip(Chip_no: Longint);
```

Parametr

Chip_no: Hodnota mezi 0 a 7 korespondující s nastavením adresy IO čipu, ve kterém mají být výstupy vynulovány.

Výsledek

IOdata: Bity v proměnné „IOdata“ zvoleného IO čipu, která korespondují s výstupy, jsou nastaveny na nízkou úroveň low (0). Vstupy zůstávají nezměněny.

Popis

Všechny kanály zvoleného IO čipu, které byly konfigurovány jako výstupy, jsou vynulovány. Stav vstupů zůstává beze změny. Proměnné „IOdata“ a „IO“ jsou upraveny podle nového stavu.

Příklad

```
BEGIN
    ConfigIOchipAsOutput(1);
    ClearIOChip(1); // Output channels 9...16 are off
END;
```

ClearAllIO

Syntaxe

```
PROCEDURE ClearAllIO;
```

Výsledek

IOdata: Bity v proměnné „IOdata“, které korespondují s výstupy, jsou nastaveny na nízkou úroveň low (0). Vstupy zůstávají beze změny. Všechny výstupy jsou vynulovány.

Popis

Všechny kanály IO čipu, které byly konfigurovány jako výstupy, jsou vynulovány. Stav vstupů je nezměněn. „IOdata“ jsou upravena podle nového stavu.

Příklad

```
BEGIN
    ConfigAllIOasOutput;
    ConfigIOchannelAsInput(2); // channel 2 configured as input
    ClearAllIO;
    // All Output channels are off
    // Input channel 2 unchanged
END;
```

SetIOchannel

Syntaxe

```
PROCEDURE SetIOchannel(Channel_no: Longint);
```

Parametr

Channel_no: Hodnota mezi 1 a 64, která koresponduje s IO kanálem, který má být upravován.

Výsledek

IOdata: Pokud je zvolený kanál výstupní, potom správný bit proměnné „IOdata“ příslušného čipu je nastaven na vysokou úroveň high (1).

Popis

V případě, že vybraný kanál byl konfigurován jako výstup, dojde k jeho úpravě. Stav vstupů zůstává beze změn. Proměnné v „IOdata“ jsou upraveny podle nového stavu.

Příklad

```
BEGIN
    ConfigIOchannelAsOutput(9);
    SetIOchannel(9); // Output channel 9 is on
END;
```

SetIOchip

Syntaxe

```
PROCEDURE SetIOchip(Chip_no: Longint);
```

Parametr

Chip_no: Hodnota mezi 0 a 7 korespondující s nastavením adresy IO čipu, ve kterém mají být výstupy konfigurovány.

Výsledek

IOdata: Bity v proměnné „IOdata“ konkrétního IO čipu, která korespondují s výstupy, jsou nastaveny na vysokou úroveň high (1). Vstupy přitom zůstávají nezměněny.

Popis

Všechny kanály zvoleného IO čipu, které byly konfigurovány jako výstupy, jsou upraveny. Stav vstupů je beze změn.

Příklad

```
BEGIN
    ConfigIOchipAsOutput(1);
    SetIOchip(1); // Output channels 9...16 are on
END;
```

SetAllIO

Syntaxe

```
PROCEDURE SetAllIO;
```

Výsledek

IOdata: Bity proměnné „IOdata“, které korespondují s výstupy, jsou nastaveny na vysokou úroveň high (1), vstupy přitom zůstávají nezměněny.

IO: Proměnné „IO“ jež korespondují s výstupy, jsou nastaveny na vysokou úroveň high (True), přitom vstupní kanály zůstávají nezměněny. U všech výstupů dochází k úpravám.

Popis

Všechny kanály IO čipů, které byly konfigurovány jako výstupy, jsou upraveny. Stav vstupů je beze změn. Proměnné „IOdata“ jsou upraveny vzhledem k novému stavu.

Příklad

```
BEGIN
    ConfigAllIOasOutput;
    ConfigIOchannelAsInput(2); // channel 2 configured as input
    SetAllIO;
    // All Output channels are on
    // Input channel 2 unchanged
END;
```

ReadIOchannel

Syntaxe

```
FUNCTION ReadIOchannel(Channel_no: Longint): Boolean;
```

Parametr

Channel_no: Hodnota mezi 1 a 64, jež koresponduje s IO kanálem, jehož stav má být přečten.

Výsledek

IOdata: Bit v proměnné „IOdata“ zvoleného kanálu je upraven vzhledem ke stavu kanálu. Úroveň high (1) představuje kanál, který byl upraven, nízká úroveň low (0) znamená, že došlo k jeho vynulování.

Popis

Stav vybraného IO kanálu je přečten a zanesen do proměnné „IOdata“. Touto funkcí dochází k vrácení stavu tohoto kanálu.

Příklad

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

ReadIOchip

Syntaxe

```
FUNCTION ReadIOchip(Chip_no: Longint): Longint;
```

Parametr

Chip_no: Hodnota mezi 0 a 7 korespondující s nastavením adresy IO čipu, ve kterém má být stav vstupů přečten.

Výsledek

IOdata: Bity proměnné „IOdata“ vybraného IO čipu jsou upraveny v závislosti na stavu IO kanálů. Úroveň high (1) znamená, že došlo k vynulování kanálu, low (0) představuje vynulování kanálů.

Popis

Stav všech kanálů IO čipu je přečten a zanesen do proměnné „IOdata“. Tato funkce provádí navrácení stavu u vstupů.

Příklad

```
var status: longint;
BEGIN
    ConfigIOchipAsInput(0);
    status := ReadIOchip(0); // Read Input channels from chip 0
END;
```

ReadIOconfigArray

Syntaxe

```
PROCEDURE ReadIOconfigArray(Buffer:Pointer);
```

Parametr

Buffer: Ukazatel na pole „dlouhých čísel“ (long integers), kde dojde k přečtení dat IO konfigurace.

Popis

Data z pole IO konfigurace jsou načtena z K8D.DLL do aplikačního programu.

Příklad

```
var // global variables
    IOconfig: ARRAY[0..MaxIOchip] OF Integer;
procedure TForm1.Button1Click(Sender: TObject);
var p:pointer;
i:integer;
s:string;
begin
    p:=@IOconfig; // Address of the data buffer for config array
    ReadIOconfigArray(p); // Read the data from K8D.DLL
    mem01.clear;
    s:='';
    for i:=0 to MaxIOchip do s:=s +inttostr(IOconfig[i])+chr(9);
    mem01.lines.add(s); // Display the IO cofig data
end;
```

ReadIOconfigArray

Syntaxe

```
PROCEDURE ReadIOdataArray(Buffer:Pointer);
```

Parametr

Buffer: Ukazatel na pole dlouhých čísel, kde mají být IO data přečtena.

Popis

Data z pole IO stavu jsou načtena z K8D.DLL do aplikačního programu.

Příklad

```
var // global variables
    IOdata: ARRAY[0..MaxIOchip] OF Integer;
procedure TForm1.Button1Click(Sender: TObject);
var p:pointer;
i:integer;
s:string;
begin
    p:=@IOdata; // Address of the data buffer for the data array
    ReadIOdataArray(p); // Read the data from K8D.DLL
    mem01.clear;
    s:='';
    for i:=0 to MaxIOchip do s:=s +inttostr(IOdata[i] and $ff)+chr(9);
    mem01.lines.add(s); // Display the IO data
end;
```

ReadDACarray

Syntaxe

```
PROCEDURE ReadDACarray(Buffer:Pointer);
```

Parametr

Buffer: Ukazatel na pole s daty dlouhých čísel, kde mají být DAC data přečtena.

Popis

Data z pole DAC jsou načtena z K8D.DLL do aplikačního programu.

Příklad

```
var // global variables
    DAC: ARRAY[1..MaxDACchannel] OF Integer;
procedure TForm1.Button1Click(Sender: TObject);
var p:pointer;
i:integer;
s:string;
begin
    p:=@DAC; // Address of the data buffer for the DAC array
    ReadDACarray(p); // Read the data from K8D.DLL
    mem01.clear;
    s:='';
    for i:=1 to MaxDACchannel do s:=s +inttostr(DAC[i] and $ff)+chr(9);
    mem01.lines.add(s); // Display the DAC dataend;
```

ReadIOconfigArray

Syntaxe

```
PROCEDURE ReadDAarray(Buffer:Pointer);
```

Parametr

Buffer: Ukazatel do pole s daty dlouhých čísel, kde mají být data přečtena.

Popis

Pole s DA daty jsou načteny z K8D.DLL do aplikačního programu.

Příklad

```
var // global variables
    DA: ARRAY[1..MaxDAchannel] OF Integer;
procedure TForm1.Button1Click(Sender: TObject);
var p:pointer;
i:integer;
s:string;
begin
    p:=@DA; // Address of the data buffer for the DA array
    ReadDAarray(p); // Read the data from K8D.DLL
    mem01.clear;
    s:='';
    for i:=1 to MaxDAchannel do s:=s +inttostr(DA[i] and $ff)+chr(9);
    mem01.lines.add(s); // Display the DA data
end;
```

Použití DLL v Delphi

Seznam některých polí v proměnných, K8D.DLL procesů a funkcí a dvou nejdůležitějších procesů aplikačního programu (FormCreate a FormClose).

```
var
    IOconfig: ARRAY[0..MaxIOchip] OF Integer;
    IOdata: ARRAY[0..MaxIOchip] OF Integer;
    DAC: ARRAY[1..MaxDACchannel] OF Integer;
    DA: ARRAY[1..MaxDAchannel] OF Integer;
implementation
{$R *.DFM}
{IO CONFIGURATION PROCEDURES}
PROCEDURE ConfigAllIOasInput; stdcall; external 'K8D.dll';
PROCEDURE ConfigAllIOasOutput; stdcall; external 'K8D.dll';
PROCEDURE ConfigIOchipAsInput(Chip_no: TIOchip); stdcall; external
'K8D.dll';
PROCEDURE ConfigIOchipAsOutput(Chip_no: TIOchip); stdcall; external
'K8D.dll';
```

```
PROCEDURE ConfigIOchannelAsInput(Channel_no: TIOchannel); stdcall; external
'K8D.dll';
PROCEDURE ConfigIOchannelAsOutput(Channel_no: TIOchannel); stdcall;
external 'K8D.dll';
{UPDATE IOdata & IO ARRAY PROCEDURES}
PROCEDURE UpdateIOdataArray(Chip_no: TIOchip; Data:Longint); stdcall;
external 'K8D.dll';
PROCEDURE ClearIOdataArray(Chip_no: TIOchip); stdcall; external 'K8D.dll';
PROCEDURE SetIOdataArray(Chip_no: TIOchip); stdcall; external 'K8D.dll';
PROCEDURE SetIOchArray(Channel_no: TIOchannel); stdcall; external
'K8D.dll';
PROCEDURE ClearIOchArray(Channel_no: TIOchannel); stdcall; external
'K8D.dll';
{OUTPUT PROCEDURES}
PROCEDURE IOoutput(Chip_no: TIOchip ; Data: Longint); stdcall; external
'K8D.dll';
PROCEDURE UpdateAllIO; stdcall; external 'K8D.dll';
PROCEDURE ClearAllIO; stdcall; external 'K8D.dll';
PROCEDURE SetAllIO; stdcall; external 'K8D.dll';
PROCEDURE UpdateIOchip(Chip_no: TIOchip); stdcall; external 'K8D.dll';
PROCEDURE ClearIOchip(Chip_no: TIOchip); stdcall; external 'K8D.dll';
PROCEDURE SetIOchip(Chip_no: TIOchip); stdcall; external 'K8D.dll';
PROCEDURE SetIOchannel(Channel_no: TIOchannel); stdcall; external
'K8D.dll';
PROCEDURE ClearIOchannel(Channel_no: TIOchannel); stdcall; external
'K8D.dll';
{6 BIT DAC CONVERTER PROCEDURES}
PROCEDURE OutputDACchannel(Channel_no: TDACchannel ; Data: Longint);
stdcall; external
'K8D.dll';
PROCEDURE ClearDACchannel(Channel_no: TDACchannel); stdcall; external
'K8D.dll';
PROCEDURE SetDACchannel(Channel_no: TDACchannel); stdcall; external
'K8D.dll';
PROCEDURE ClearDACchip(Chip_no: TIOcard); stdcall; external 'K8D.dll';
PROCEDURE SetDACchip(Chip_no: TIOcard); stdcall; external 'K8D.dll';
PROCEDURE ClearAllDAC; stdcall; external 'K8D.dll';
PROCEDURE SetAllDAC; stdcall; external 'K8D.dll';
{8 BIT DA CONVERTER PROCEDURES}
PROCEDURE OutputDACHannel(Channel_no: TDAchannel ; Data: Longint); stdcall;
external
'K8D.dll';
PROCEDURE ClearDACHannel(Channel_no: TDAchannel); stdcall; external
'K8D.dll';
PROCEDURE SetDACHannel(Channel_no: TDAchannel); stdcall; external
'K8D.dll';
PROCEDURE ClearAllDA; stdcall; external 'K8D.dll';
PROCEDURE SetAllDA; stdcall; external 'K8D.dll';
{GENERAL PROCEDURES}
PROCEDURE SelectI2CprinterPort(Printer_no: Longint); stdcall; external
'K8D.dll';
PROCEDURE Start_K8000; stdcall; external 'K8D.dll';
PROCEDURE Stop_K8000; stdcall; external 'K8D.dll';
{INPUT FUNCTIONS}
function ReadIOchip(Chip_no: TIOchip):longint; stdcall; external 'K8D.dll';
function ReadIOchannel(Channel_no: TIOchannel):boolean; stdcall; external
'K8D.dll';
function ReadADchannel(Channel_no:TADchannel):longint; stdcall; external
'K8D.dll'
PROCEDURE ReadIOconfigArray(Buffer:Pointer); stdcall; external 'K8D.dll';
PROCEDURE ReadIOdataArray(Buffer:Pointer); stdcall; external 'K8D.dll';
```

```

PROCEDURE ReadDACarray(Buffer:Pointer); stdcall; external 'K8D.dll';
PROCEDURE ReadDAarray(Buffer:Pointer); stdcall; external 'K8D.dll';
procedure TForm1.FormCreate(Sender: TObject);
begin
    Start_K8000;
end;
procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
begin
    Stop_K8000;
end;

```

Použití DLL ve Visual Basic

Seznam K8D.DLL procedur a funkcí, některých polí proměnných a 2 nejdůležitější postupy aplikačního programu (Form_Load a Form_Terminate).

```

Option Explicit
'Declare use of the DLL
'K8D.DLL interfaces
'IO CONFIGURATION PROCEDURES
Private Declare Sub ConfigAllIOasInput Lib "k8d.dll" ()
Private Declare Sub ConfigAllIOasOutput Lib "k8d.dll" ()
Private Declare Sub ConfigIOchipAsInput Lib "k8d.dll" (ByVal Chip_no As Long)
Private Declare Sub ConfigIOchipAsOutput Lib "k8d.dll" (ByVal Chip_no As Long)
Private Declare Sub ConfigIOchannelAsInput Lib "k8d.dll" (ByVal Channel_no As Long)
Private Declare Sub ConfigIOchannelAsOutput Lib "k8d.dll" (ByVal Channel_no As Long)
'OUTPUT PROCEDURES
Private Declare Sub IOoutput Lib "k8d.dll" (ByVal Chip_no As Long, ByVal Data As Long)
Private Declare Sub UpdateAllIO Lib "k8d.dll" ()
Private Declare Sub ClearAllIO Lib "k8d.dll" ()
Private Declare Sub SetAllIO Lib "k8d.dll" ()
Private Declare Sub UpdateIOchip Lib "k8d.dll" (ByVal Chip_no As Long)
Private Declare Sub ClearIOchip Lib "k8d.dll" (ByVal Chip_no As Long)
Private Declare Sub SetIOchip Lib "k8d.dll" (ByVal Chip_no As Long)
Private Declare Sub SetIOchannel Lib "k8d.dll" (ByVal Channel_no As Long)
Private Declare Sub ClearIOchannel Lib "k8d.dll" (ByVal Channel_no As Long)
'INPUT FUNCTIONS AND PROCEDURES
Private Declare Function ReadIOchip Lib "k8d.dll" (ByVal Chip_no As Long) As Long
Private Declare Function ReadIOchannel Lib "k8d.dll" (ByVal Channel_no As Long) As Boolean
Private Declare Sub ReadIOconfigArray Lib "k8d.dll" (Array_Pointer As Long)
Private Declare Sub ReadIODataArray Lib "k8d.dll" (Array_Pointer As Long)
Private Declare Sub ReadDACarray Lib "k8d.dll" (Array_Pointer As Long)
Private Declare Sub ReadDAarray Lib "k8d.dll" (Array_Pointer As Long)
'How to use these calls:
' ReadIOconfigArray IOconfig(0)
' ReadIODataArray Iodata(0)
' ReadDACarray DAC(1)
' ReadDAarray DA(1)
'6 BIT DAC CONVERTER PROCEDURES
Private Declare Sub OutputDACchannel Lib "k8d.dll" (ByVal Channel_no As Long, ByVal Data As Long)
Private Declare Sub ClearDACchannel Lib "k8d.dll" (ByVal Channel_no As Long)

```

```

Private Declare Sub SetDACchannel Lib "k8d.dll" (ByVal Channel_no As Long)
Private Declare Sub ClearDACchip Lib "k8d.dll" (ByVal Chip_no As Long)
Private Declare Sub SetDACchip Lib "k8d.dll" (ByVal Chip_no As Long)
Private Declare Sub ClearAllDAC Lib "k8d.dll" ()
Private Declare Sub SetAllDAC Lib "k8d.dll" ()
'8 BIT DA CONVERTER PROCEDURES
Private Declare Sub OutputDachannel Lib "k8d.dll" (ByVal Channel_no As Long, ByVal Data As Long)
Private Declare Sub ClearDachannel Lib "k8d.dll" (ByVal Channel_no As Long)
Private Declare Sub SetDachannel Lib "k8d.dll" (ByVal Channel_no As Long)
Private Declare Sub ClearAllDA Lib "k8d.dll" ()
Private Declare Sub SetAllDA Lib "k8d.dll" ()
'8 BIT AD CONVERTER FUNCTION
Private Declare Function ReadADchannel Lib "k8d.dll" (ByVal Channel_no As Long) As Boolean
'GENERAL PROCEDURES
Private Declare Sub SelectI2CprinterPort Lib "k8d.dll" (ByVal port As Long)
Private Declare Sub Start_K8000 Lib "k8d.dll" ()
Private Declare Sub Stop_K8000 Lib "k8d.dll" ()
'COMMON USED GLOBALS
Const MaxIOcard As Long = 3
Const MaxIOchip As Long = 7
Const MaxDACchannel As Long = 32
Const MaxDachannel As Long = 4
'Declare variables
Dim IOconfig(0 To MaxIOchip) As Long
Dim Iodata(0 To MaxIOchip) As Long
Dim DAC(1 To MaxDACchannel) As Long
Dim DA(1 To MaxDachannel) As Long
Private Sub Form_Load()
Start_K8000
End Sub
Private Sub Form_Terminate()
Stop_K8000
End Sub

```



Recyklace



Elektronické a elektrické produkty nesmějí být vřazovány do domovních odpadů. Likviduje odpad na konci doby životnosti výrobku přiměřeně podle platných zákonných ustanovení.

Šetřete životní prostředí! Přispějte k jeho ochraně!

Překlad tohoto návodu zajistila společnost Conrad Electronic Česká republika, s. r. o.

Všechna práva vyřazena. Jakékoliv druhy kopii tohoto návodu, jako např. fotokopie, jsou předmětem souhlasu společnosti Conrad Electronic Česká republika, s. r. o. Návod k použití odpovídá technickému stavu při tisku! **Změny vyřazeny!**

© Copyright Conrad Electronic Česká republika, s. r. o.

REI/10/2015