



NAVODILA ZA UPORABO

Kompaktni digitalni USB modul z 8 optičnimi vhodi Deditec USB-OPTOIN-8_B

Kataloška št.: 19 67 04

KAZALO

1. UVOD	3
1.1 SPLOŠNE OPOMBE	3
1.2 ZADOVOLJSTVO KUPCA.....	3
1.3 ODZIV KUPCEV	3
2. OPIS STROJNE OPREME	3
2.1 HITRA NAMESTITEV	3
2.2 USB-OPTOIN-8	4
3. PROGRAMSKA OPREMA.....	6
3.1 UPORABA NAŠIH IZDELKOV.....	6
3.2 "DELIB DRIVER LIBRARY".....	6
3.3 TESTNI PROGRAMI	11
4. REFERENCA "DELIB API"	11
4.1 FUNKCIJE UPRAVLJANJA.....	11
4.2 ODPRAVA NAPAK.....	12
4.3 BRANJE DIGITALNIH VHODOV	13
4.4 NASTAVITEV DIGITALNIH IZHODOV.....	15
4.5 PRIMER PROGRAMA	17
5. DODATEK	20
5.1 POPRAVKI	20
GARANCIJSKI LIST	21

1. UVOD

1.1 SPLOŠNE OPOMBE

Najprej vam želimo čestitati, ker ste se odločili za nakup visoko kvalitetnega izdelka Deditec.

Naši inženirji izdelke razvijajo v skladu z zahtevami po kvaliteti visokih standardov. Že v času zasnove in razvoja poskrbijo, da imajo izdelki poleg kvalitete dolgotrajno uporabo in optimalno fleksibilnost.

Modularna zasnova

Modularna zasnova naših izdelkov skrajša čas in strošek razvoja. Zato vam lahko visoko kvalitetne izdelke ponudimo po konkurenčnih cenah.

Razpoložljivost

Če določena komponenta na trgu ni več na voljo, lahko zaradi modularne zasnove naših izdelkov, namesto celega izdelka ponovno izdelamo le modul.

1.2 ZADOVOLJSTVO KUPCA

Naša filozofija: zadovoljen kupec se vrača. Zato je zadovoljstvo kupca za nas na prvem mestu.

Če po naključju z našim izdelkom niste zadovoljni, nas takoj pokličite ali nam pošljite elektronsko sporočilo. Težavo bomo rešili.

1.3 ODZIV KUPCEV

Naši najboljši izdelki so posledica skupnega razvoja z našimi kupci. Zato smo vam hvaležni za komentarje in predloge.

2. OPIS STROJNE OPREME

2.1 HITRA NAMESTITEV

1. Korak 1 – Namestitev programske opreme in gonilnikov

Sedaj z datoteko "delib_install.exe" na priloženem Deditec CD-ju namestite gonilnik "DELIB library".

Najdete jo na Deditec CD-ju pod "\zip\DELIB\delib_install.exe".

Opomba: na spletni strani www.deditec.de lahko najdete zadnjo različico gonilnikov DELIB.

2. Korak 2 – Priklop modula

Z USB kablom povežite vaš računalnik in USB vhod na modulu.

Po približno 20 sekundah modul zazna gonilnike in sedaj ga lahko testirate in uporabljate.

3. Korak 3 – Testiranje povezave in modula

V "Start" meniju: "Start -> All Programs -> DEDITEC -> DELIB -> Sample Programs" boste našli nekatere primere programov za testiranje vašega modula.

2.2 USB-OPTOIN-8

1. TEHNIČNI PODATKI

- Vmesnik USB 1.1 / USB 2.0
- Oskrba z energijo +5V (napajanje poteka preko USB bus)
- 8 vhodov opto-spojnikov (opcijski 15V, 12V in 5V so na voljo)
 - 24V AC preklopna napetost (opcijsko 15V, 12V in 5V so na voljo)
 - 16-bitni števec za vsak vhod
 - zaznavanje pulzov med dvema izbirnima cikloma
- Galvansko izolirani uporabljeni opto-spojniki
- Območje variabilne vhodne napetosti min 5V, max 30V AC (standard: 15-30V)
- Beleženje impulzov med 2 odčitanimi cikli
- Kontrolni LED indikator: LED za 5V napajanje
- Dimenzije: 77mm x 67,5mm x 55mm (D x Š x V)
- Obratovalna temperatura: 10°C ... 50°C

Specifični podatki izdelkov v seriji:

Izdelek	Priklp	LED aktivnosti	Ohišje
USB-OPTOIN-8	Vijačne vrstne sponke	1 za vsak izhod	Da
USB-OPTOIN-8_A	Vijačni	-	-
USB-OPTOIN-8_B	Vijačne vrstne sponke	1 za vsak izhod	-

2. SLIKE IZDELKOV SERIJE

USB-OPTOIN-8



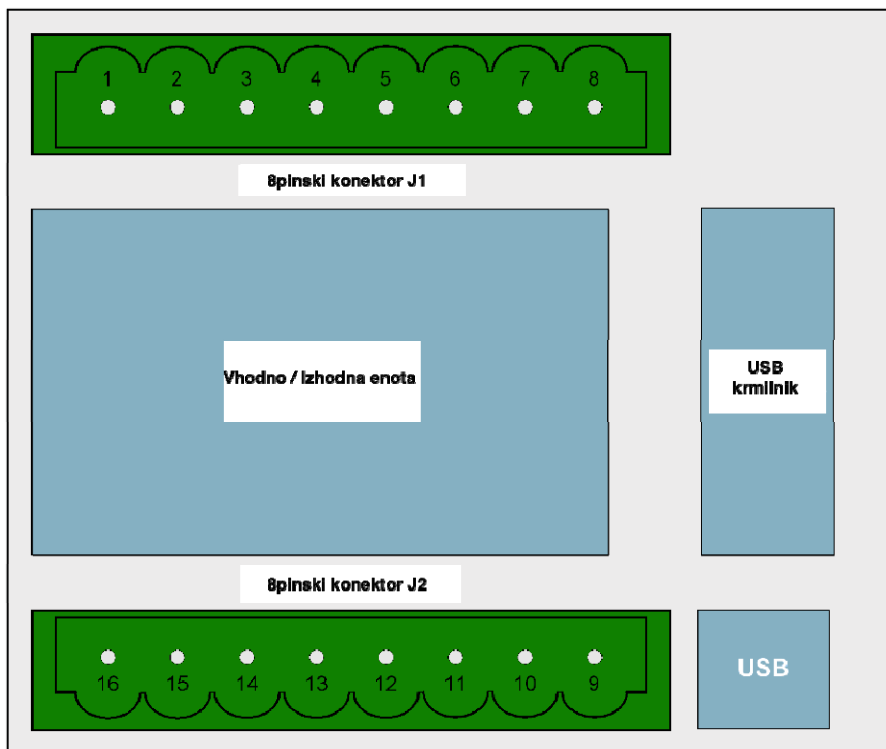
USB-OPTOIN-8_A



USB-OPTOIN-8_B

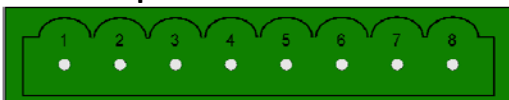


3. SHEMA NAPRAVE



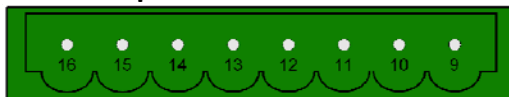
4. DOLOČITEV PINOV

Določitev pinov J1



PIN	Opis
1	Vhodni kanal 1 +
2	Vhodni kanal 1 –
3	Vhodni kanal 2 +
4	Vhodni kanal 2 –
5	Vhodni kanal 3 +
6	Vhodni kanal 3 –
7	Vhodni kanal 4 +
8	Vhodni kanal 4 –

Določitev pinov J2



PIN	Opis
9	Vhodni kanal 5 +
10	Vhodni kanal 5 –
11	Vhodni kanal 6 +
12	Vhodni kanal 6 –
13	Vhodni kanal 7 +
14	Vhodni kanal 7 –
15	Vhodni kanal 8 +
16	Vhodni kanal 8 –

5. VHODI

- **Registracija kratkih vhodnih impulzov**
Kratki vhodni impulzi med odčitanimi cikli se registrirajo skozi dodatno enoto in jih lahko odčitate ločeno.
- **Galvansko odspojeni skozi opto-spojnik**
AC vhodni opto-spojnik zagotavlja galvansko izolacijo modula glede na priključeno opremo. Zagotavlja tudi varno povezavo modula za obrnjene tokove in temena visoke napetosti.

3. PROGRAMSKA OPREMA

3.1 UPORABA NAŠIH IZDELKOV

1. DOSTOP Z GRAFIČNIMI APLIKACIJAMI

Nudimo pogonske vmesnike na primer za "LabVIEW" in "ProfiLab". "DELIB driver library" je osnova, ki jo lahko "ProfiLAB" direktno aktivira.

Za "LabVIEW" nudimo enostavno povezavo gonilnika s primeri!

2. DOSTOP Z "DELIB DRIVER LIBRARY"

V Dodatku lahko najdete popolno funkcijsko referenco za integracijo naših API-funkcij v vaši programski opremi. Dodatno nudimo primere za naslednje programske jezike:

- C
- C++
- C#
- Delphi
- VisualBasic
- VB.NET
- MS-Office

3. DOSTOP S PROTOKOLOM

Protokol za aktiviranje naših izdelkov je odprta koda. Tako lahko naše izdelke uporabljate na sistemih brez Windows ali Linux.

4. DOSTOP S PRILOŽENIMI TESTNIMI PROGRAMI

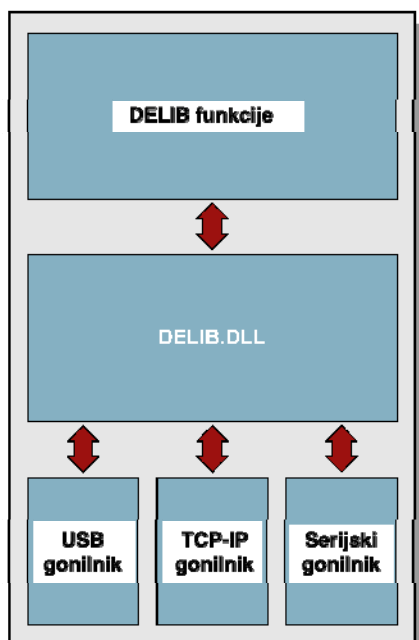
Za najpomembnejše funkcije naših izdelkov nudimo testne programe, ki so enostavni za uporabo. Ti se samodejno namestijo skupaj z namestitvijo "DELIB driver library"

Tako lahko na primer releje direktno testirate ali preverite napetost A/D pretvornika.

3.2 "DELIB DRIVER LIBRARY"

1. PREGLED

Spodnja slika opisuje strukturo "DELIB driver library":



“DELIB driver library“ omogoča enoten odziv DEDITEC strojne opreme, s posebnim upoštevanjem naslednjih vidikov:

- Neodvisno od operacijskega sistema
- Neodvisno od programskega jezika
- Neodvisno od izdelka

A. Program v različnih operacijskih sistemih

“DELIB driver library“ omogoča enoten odziv naših izdelkov na različne operacijske sisteme. Poskrbeli smo, da lahko vsi naši izdelki odzovejo z nekaj ukazi. Ne glede na to kateri operacijski sistem uporabljate.

B. Program z različnimi programskimi jeziki

Nudimo enotne ukaze za kreiranje lastnih aplikacij. Za to poskrbi “DELIB driver library“

Vi izberete programski jezik!

Aplikacije lahko enostavno razvijate z jeziki C++, C, Visual Basic, Delphi ali LabVIEW.

C. Program neodvisen od vmesnika

Zapišite svojo aplikacijo neodvisno od strežnika!

Programirajte aplikacijo za naš USB izdelek. – Deluje tudi z Ethernetom ali našim RS-232 izdelkom!

D.SDK-Kit za programerja

“DELIB“ integrirajte v vašo aplikacijo. Na zahtevo boste brezplačno prejeli skripto namestitve, ki vam omogoča da “DELIB installation“ integrirate v vašo aplikacijo.

2. PODPRTI OPERACIJSKI SISTEMI

Naši izdelki podpirajo naslednje operacijske sisteme:

- Windows 7
- Windows Vista
- Windows XP

- Windows 2000
- Linux

3. PODPRTI PROGRAMSKI JEZIKI

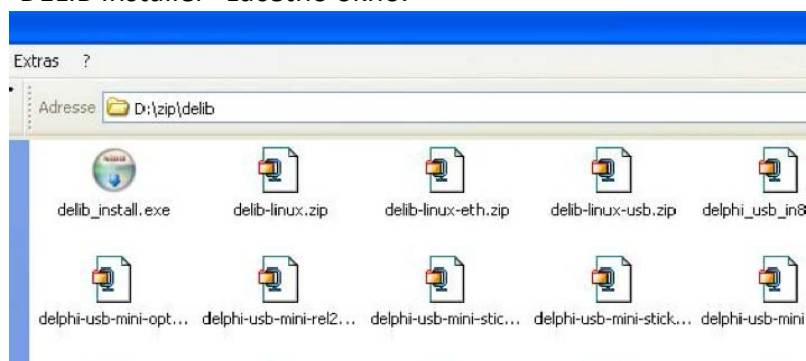
Naši izdelki so odzivni preko naslednjih programskih jezikov:

- C
- C++
- C#
- Delphi
- VisualBasic
- VB.NET
- MS-Office

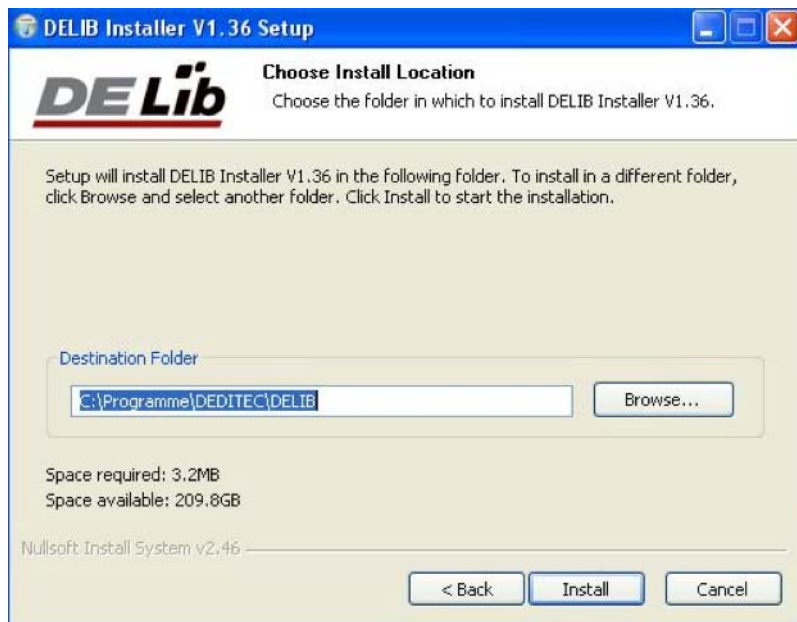
4. NAMESTITEV "DELIB DRIVER LIBRARY"



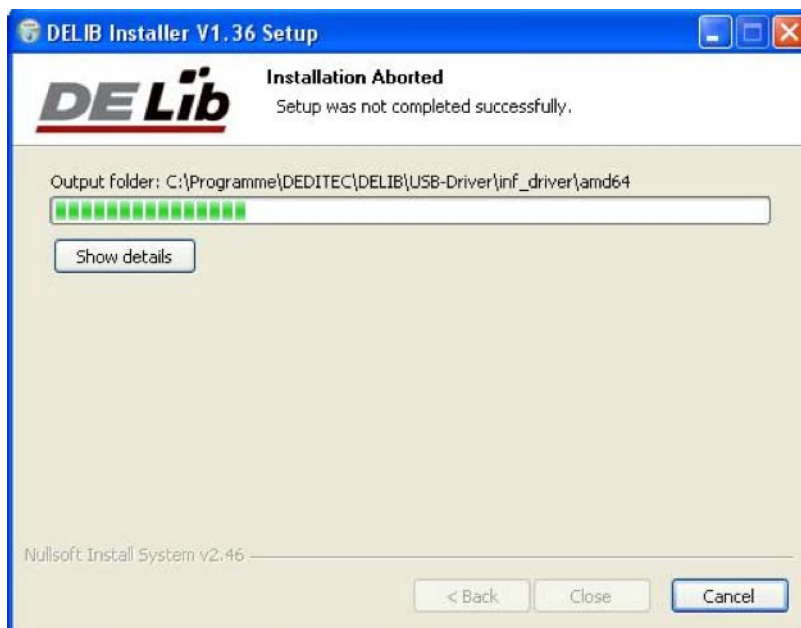
"DELIB Installer" začetno okno.



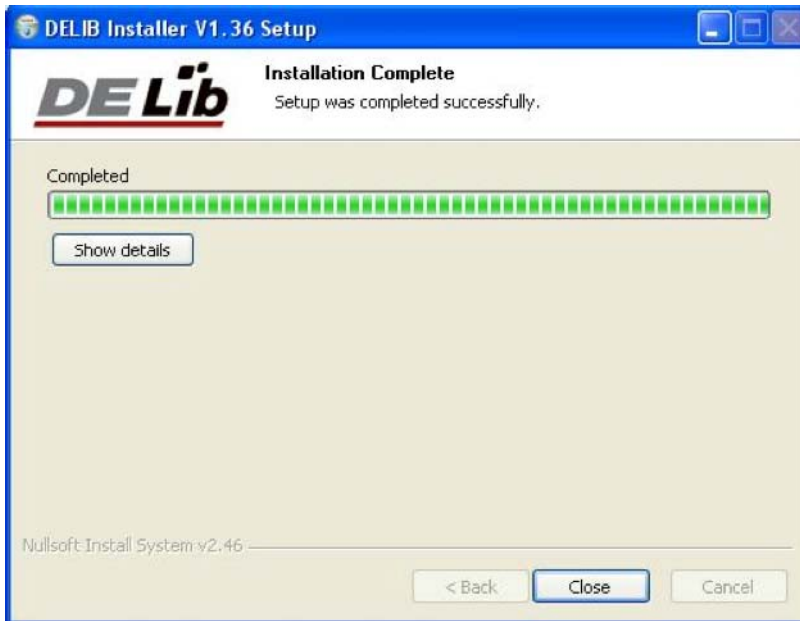
DEDITEC CD z gonilniki vstavite v pogon in zaženite "delib_install.exe" "DELIB driver library je na voljo tudi na <http://www.deditec.en/delib>.



Kliknite na **“Install”** (namesti).



Namestijo se gonilniki.



“DELIB driver library” je tako nameščena. Za dokončanje namestitve kliknite na “Close” (zapri).

Vaš modul lahko konfigurirate z “DELIB Configuration Utility” (glejte naslednje poglavje). To je potrebno samo v primeru, če je prisoten več kot en modul.

5. DELIB KONFIGURACIJSKI PRIPOMOČEK (“DELIB Configuration Utility”)



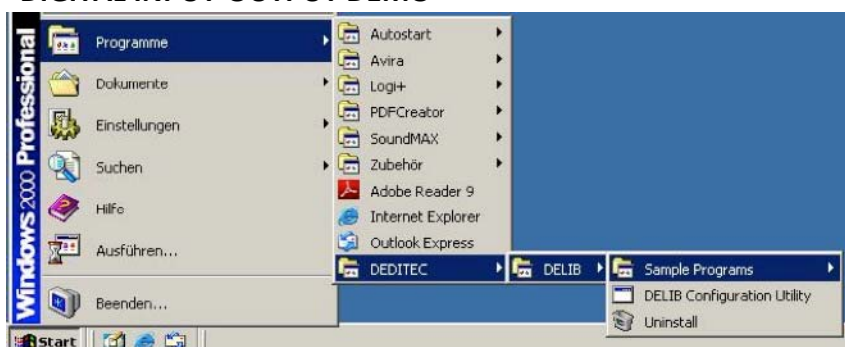
Konfiguracijski pripomoček zaženite na naslednji način:

“Start → Programs → DEDITEC → DELIB → DELIB Configuration Utility”

“DELIB Configuration Utility” je program za konfiguracijo in razdelitev identičnih USB modulov v sistemu. To je potrebno samo če je prisoten več kot en modul.

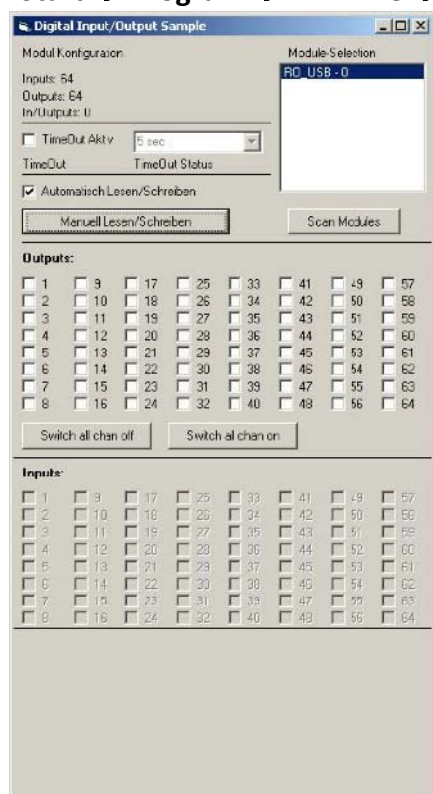
3.3 TESTNI PROGRAMI

1. "DIGITAL INPUT-OUTPUT DEMO"



"Digital Input-Output Demo" zaženite kot je prikazano:

"Start → Program → DEDITEC → DELIB → Digital Input-Output Demo".



Posnetek zaslona prikazuje test RO-USB-O64-R64. Konfiguracija modula (64 vhodov in 64 izhodov) je prikazana na zgornji levi strani.

4. REFERENCA "DELIB API"

4.1 FUNKCIJE UPRAVLJANJA

1. "DapiOpenModule"

Opis: s to funkcijo odprete določen modul.

Definicija: *ULONG DapiOpenModule(ULONG moduleID, ULONG nr);*

Parametri:

modulID = določa modul, ki naj se odpre (glejte delib.h)

nr = 0 -> 1. modul

nr = 1 -> 2. modul

Povratna vrednost:

handle = handle na ustrezen modul

handle = 0 -> modul ni bil najden

Opombe: "Handle" vrnjen s to funkcijo je potreben za identifikacijo modula za vse ostale funkcije.

Primer programa:

```
// USB-Modul öffnen
handle = DapiOpenModule(RO_USB1, 0);
printf("handle = %x\n", handle);
if (handle==0)
{
// USB Modul wurde nicht gefunden
printf("Modul konnte nicht geöffnet werden\n");
return;
}
```

2. "DapiCloseModule"

Opis: s to funkcijo zaprete določen modul.

Definicija: *ULONG DapiCloseModule(ULONG handle);*

Parametri:

handle = to je handle odprtega modula

Povratna vrednost:

je ni

Primer programa:

```
// Close the module
DapiCloseModule(handle);
```

3. "DapiGetDELIBVersion"

Opis: s to funkcijo vrnete nameščeno DELIB različico.

Definicija: *ULONG DapiGetDELIBVersion(ULONG mode, ULONG par);*

Parametri:

mode = Mode s katerim se bere različica (mora biti 0).

par = Ta parameter ni definiran (mora biti 0).

Povratna vrednost:

version = številka različice nameščene DELIB različice [hex].

Primer programa:

```
version = DapiGetDELIBVersion(0, 0);
//Bei installierter Version 1.32 ist version = 132(hex)
```

4.2 ODPRAVA NAPAK**1. "DapiGetLastError"**

Opis: s to funkcijo vrnete zadnjo registrirano napako.

Definicija: *ULONG DapiGetLastError();*

Parametri:

jih ni

Povratna vrednost:

Error code (koda napake)

0 = ni napake. (glejte delib.h)

Primer programa:

```
ULONG error;
error=DapiGetLastError();
if(error==0) return FALSE;
printf("ERROR = %d", error);
```

2. "DapiGetLastErrorText"

Opis: s to funkcijo preberete besedilo zadnje registrirane napake.

Definicija: *extern ULONG_stdcall DapiGetLastErrorText(unsigned char * msg, unsigned long msg_length);*

Parametri:

msg = besedilo medpomnilnika

msg_length = dolžina medpomnilnika

Primer programa:

```
BOOL IsError ()
{
    if (DapiGetLastError () != DAPI_ERR_NONE)
    {
        unsigned char msg[500];

        DapiGetLastErrorText((unsigned char*) msg, sizeof(msg));
        printf ("Error Code = %x * Message = %s\n", 0, msg);
        return TRUE;
    }
    return FALSE;
}
```

4.3 BRANJE DIGITALNIH VHODOV

1. "DapiDIGet1"

Opis: s tem ukazom preberete vhod enega znaka.

Definicija: *ULONG DapiDIGet1(ULONG handle, ULONG ch);*

Parametri:

handle = to je handle odprtega modula.

ch = določa številko vhoda ki bo prebran (0 ..).

Povratna vrednost:

Status vhoda (0 / 1).

2. "DapiDIGet8"

Opis: s tem ukazom preberete 8 digitalnih vhodov hkrati.

Definicija: *ULONG DapiDIGet8(ULONG handle, ULONG ch);*

Parametri:

handle = to je handle odprtega modula.

ch = določa številko vhoda od katerega se začne branje (0, 8, 16, 24, 32, ..).

Povratna vrednost:

Status prebranih vhodov.

3. "DapiDIGet16"

Opis: s tem ukazom preberete 16 digitalnih vhodov hkrati.

Definicija: *ULONG DapiDIGet16(ULONG handle, ULONG ch);*

Parametri:

handle = to je handle odprtega modula.

ch = določa številko vhoda od katerega se začne branje (0, 16, 32, ..).

Povratna vrednost:

Status prebranih vhodov.

4. "DapiDIGet32"

Opis: s tem ukazom preberete 32 digitalnih vhodov hkrati.

Definicija: *ULONG DapiDIGet32(ULONG handle, ULONG ch);*

Parametri:

handle = to je handle odprtega modula.

ch = določa številko vhoda od katerega se začne branje (0, 32, 64 ..).

Povratna vrednost:

Status prebranih vhodov.

Primer programa:

```
unsigned long data;
// -----
// Einen Wert von den Eingängen lesen (Eingang 1-31)
data = (unsigned long) DapiDIGet32(handle, 0);
// Chan Start = 0
printf("Eingang 0-31 : 0x%x\n", data);
printf("Taste für weiter\n");
getch();
// -----
// Einen Wert von den Eingängen lesen (Eingang 32-64)
data = (unsigned long) DapiDIGet32(handle, 32);
// Chan Start = 32
printf("Eingang 32-64 : 0x%x\n", data);
printf("Taste für weiter\n");
getch();
```

5. "DapiDIGet64"

Opis: s tem ukazom preberete 64 digitalnih vhodov hkrati.

Definicija: *ULONG DapiDIGet64(ULONG handle, ULONG ch);*

Parametri:

handle = to je handle odprtega modula.

ch = določa številko vhoda od katerega se začne branje (0, 64 ..).

Povratna vrednost:

Status prebranih vhodov.

6. "DapiDIGetFF32"

Opis: s tem ukazom preberete "flip-flops" z vhodov in jih ponastavite (sprememba statusa vhoda).

Definicija: *ULONG DapiDIGetFF32(ULONG handle, ULONG ch);*

Parametri:

handle = to je handle odprtega modula.

ch = določa številko vhoda od katerega se začne branje (0, 32, ..).

Povratna vrednost:

Status 32 vhoda spremembe statusa.

4.4 NASTAVITEV DIGITALNIH IZHODOV

1. "DapiDOSet1"

Opis: s tem ukazom nastavite enojni izhod.

Definicija: *void DapiDOSet1(ULONG handle, ULONG ch, ULONG data);*

Parametri:

handle = to je handle odprtega modula.

ch = določa številko izhoda za nastavitvev (0 ..).

data = določa vrednost podatkov ki bodo zapisani (0 / 1)

Povratna vrednost:

Je ni.

2. "DapiDOSet8"

Opis: s tem ukazom nastavite 8 digitalnih izhodov hkrati.

Definicija: *void DapiDOSet8(ULONG handle, ULONG ch, ULONG data);*

Parametri:

handle = to je handle odprtega modula.

ch = določa številko izhoda, od katerega se začne pisanje (0, 8, 16, 24, 32, ..).

data = določa vrednost podatkov za zapis na izhode.

Povratna vrednost:

Je ni.

3. "DapiDOSet16"

Opis: s tem ukazom nastavite 16 digitalnih izhodov hkrati.

Definicija: *void DapiDOSet16(ULONG handle, ULONG ch, ULONG data);*

Parametri:

handle = to je handle odprtega modula.

ch = določa številko izhoda, od katerega se začne pisanje (0, 16, 32, ..).

data = določa vrednost podatkov za zapis na izhode.

Povratna vrednost:

Je ni.

4. "DapiDOSet32"

Opis: s tem ukazom nastavite 32 digitalnih izhodov hkrati.

Definicija: *void DapiDOSet32(ULONG handle, ULONG ch, ULONG data);*

Parametri:

handle = to je handle odprtega modula.

ch = določa številko izhoda, od katerega se začne pisanje (0, 32, 64, ..).

data = določa vrednost podatkov za zapis na izhode.

Povratna vrednost:

Je ni.

Primer programa:

```
// Einen Wert auf die Ausgänge schreiben
data = 0x0000ff00; // Ausgänge 9-16 werden auf 1 gesetzt
DapiDOSet32(handle, 0, data); // Chan Start = 0
printf("Schreibe auf Ausgänge Daten=0x%x\n", data);
printf("Taste für weiter\n");
getch();
// -----
// Einen Wert auf die Ausgänge schreiben
data = 0x80000000; // Ausgang 32 wird auf 1 gesetzt
DapiDOSet32(handle, 0, data); // Chan Start = 0
printf("Schreibe auf Ausgänge Daten=0x%x\n", data);
printf("Taste für weiter\n");
getch();
// -----
// Einen Wert auf die Ausgänge schreiben
data = 0x80000000; // Ausgang 64 wird auf 1 gesetzt
DapiDOSet32(handle, 32, data); // Chan Start = 32
printf("Schreibe auf Ausgänge Daten=0x%x\n", data);
printf("Taste für weiter\n");
getch();
```

5. "DapiDOSet64"

Opis: s tem ukazom nastavite 64 digitalnih izhodov.

Definicija: *void DapiDOSet64(ULONG handle, ULONG ch, ULONG data);*

Parametri:

handle = to je handle odprtega modula.

ch = določa številko izhoda, od katerega se začne pisanje (0, 64, ..).

data = določa vrednost podatkov za zapis na izhode.

Povratna vrednost:

Je ni.

6. "DapiDOReadback32"

Opis: s tem ukazom nazaj preberete 32 digitalnih izhodov.

Definicija: *ULONG DapiDOReadback32(ULONG handle, ULONG ch);*

Parametri:

handle = to je handle odprtega modula.

ch = določa številko vhoda, od katerega se začne branje (0, 32, ..).

Povratna vrednost:

Status 32 izhodov.

7. "DapiDOReadback64"

Opis: s tem ukazom nazaj preberete 64 digitalnih izhodov.

Definicija: *ULONG DapiDOReadback64(ULONG handle, ULONG ch);*

Parametri:

handle = to je handle odprtega modula.

ch = določa številko vhoda, od katerega se začne branje (0, 64, ..).

Povratna vrednost:

Status 64 izhodov

4.5 PRIMER PROGRAMA

```

//*****
//*****
//*****
//*****
//*****
//
//
// product: usb-optoin-8-relais-8 (ModuleID = USB_OPTOIN_8_RELAIS_8)
// configuration: digital-outputs
// programming language: vc
//
//
// (c) DEDITEC GmbH, 2011
// web: http://www.deditec.de/
// mail: vertrieb@deditec.de
//
//
//*****
//*****
//*****
//*****
//*****
//
//
// Please include the following library on linking: delib.lib
//
// This can be done at the project settings (Project/Settings/Link ->
// Object/library modules) .. extend the existing line with the ending
// "$ (DELIB_LIB)\delib.lib" (with quotation marks)
//
// Including the header file delib.h (Project/Settings/C/C++ -> select
// category
// "Preprocessor" -> Additional include directories) .. enter the line
// "$ (DELIB_INCLUDE)" (with quotation marks)

#include <windows.h>
#include <stdio.h>
#include "conio.h"

#include "delib.h"

// -----
// GetLastError function

BOOL IsError()
{
    unsigned char msg[500];

    if (DapiGetLastError() != DAPI_ERR_NONE)
    {
        DapiGetLastErrorText((unsigned char*) msg, sizeof(msg));
        printf("Error Code = %x * Message = %s\n", 0, msg);
    }
}

```

```

        DapiClearLastError();

        return TRUE;
    }

    return FALSE;
}

//*****
//*****
//*****
//*****
//*****

void main(void)
{
    unsigned long handle;
    unsigned long value;

    // -----
    // Open Module

    handle = DapiOpenModule(USB_OPTOIN_8_RELAIS_8, 0);

    printf("Module handle = %x\n", handle);

    // -----
    // Module not found!

    if (handle==0)
    {
        printf("Could not open module!\n");
        printf("Press any key to exit\n");
        getch();
        return;
    }

    // -----
    // Module found!

    printf("Module has been opened\n");

    // -----
    // Show config of module

    value = DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_GET_MODULE_CONFIG,
        DAPI_SPECIAL_GET_MODULE_CONFIG_PAR_DO, 0, 0);
    IsError();
    printf("Configuration of the module: no. of digital outputs %d\n",
value);
    printf("Press any key to continue\n");
    getch();

    // -----
    // Write output channels

```

```

DapiDOSet1(handle, 0, 1);
IsError();
printf("Output channel 0 has been switched on\n");
printf("Press any key to continue\n");
getch();

DapiDOSet1(handle, 0, 0);
IsError();
printf("Output channel 0 has been switched off\n");
printf("Press any key to continue\n");
getch();

DapiDOSet1(handle, 1, 1);
IsError();
printf("Output channel 1 has been switched on\n");
printf("Press any key to continue\n");
getch();

DapiDOSet1(handle, 1, 0);
IsError();
printf("Output channel 1 has been switched off\n");
printf("Press any key to continue\n");
getch();

DapiDOSet8(handle, 0, 0xff); //hexadecimal
IsError();
printf("Output channel 0-7 have been switched on\n");
printf("Press any key to continue\n");
getch();

DapiDOSet8(handle, 0, 0);
IsError();
printf("Output channel 0-7 have been switched off\n");
printf("Press any key to continue\n");
getch();

// -----
// Write and readback output channels

DapiDOSet8(handle, 0, 31);
IsError();
printf("Output channel 0-7 have been switched on\n");
printf("Press any key to continue\n");
getch();

value = DapiDOReadback32(handle, 0);
IsError();
printf("Readback output channel 0-3\n");
printf("value = %d\n", value);
printf("Press any key to continue\n");
getch();

DapiDOSet8(handle, 0, 0);
IsError();
printf("Output channel 0-7 have been switched off\n");
printf("Press any key to continue\n");

```

```

    getch();

    value = DapiDOReadback32(handle, 0);
    IsError();
    printf("Readback output channel 0-3\n");
    printf("value = %d\n", value);
    printf("Press any key to continue\n");
    getch();

    // -----
    // Set timeout of output channels to 5 seconds

    DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_TIMEOUT,
DAPI_SPECIAL_TIMEOUT_SET_VALUE_SEC, 5, 0);
    IsError();
    printf("Timeout has been set to 5 seconds\n");
    printf("Press any key to continue\n");
    getch();

    // -----
    // Activate timeout and switch on output channels 0-3

    DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_TIMEOUT,
DAPI_SPECIAL_TIMEOUT_ACTIVATE, 0, 0);
    IsError();
    DapiDOSet8(handle, 0, 15);
    IsError();
    printf("Timeout has been activated\n");
    printf("Output channels 0-3 have been switched on and will be switched
off automatically after 5 seconds\n");
    printf("Press any key to continue\n");
    getch();

    // -----
    // Deactivate timeout

    DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_TIMEOUT,
DAPI_SPECIAL_TIMEOUT_DEACTIVATE, 0, 0);
    IsError();
    printf("Timeout has been deactivated\n");
    printf("Press any key to continue\n");
    getch();

    // -----
    // Close Module

    DapiCloseModule(handle);
    printf("Module closed\n");
    printf("End of program!\n");
    printf("Press any key to exit\n");
    getch();

    return ;
}

```

5. DODATEK

5.1 POPRAVKI

Rev 2.00 "First DEDITEC issue"



GARANCIJSKI LIST

Conrad Electronic d.o.o. k.d.
Ljubljanska c. 66, 1290 Grosuplje
Fax: 01/78 11 250, Tel: 01/78 11
248
www.conrad.si, info@conrad.si

Izdelek: **Kompaktni digitalni USB modul z 8 optičnimi vhodi Deditec USB-OPTOIN-8_B**

Kat. št.: **19 67 04**

Garancijska izjava:

Proizvajalec jamči za kakovost oziroma brezhibno delovanje v garancijskem roku, ki začne teči z izročitvijo blaga potrošniku. **Garancija velja na območju Republike Slovenije. Garancija za izdelek je 1 leto.**

Izdelek, ki bo poslan v reklamacijo, vam bomo najkasneje v skupnem roku 45 dni vrnili popravljenega ali ga zamenjali z enakim novim in brezhibnim izdelkom. Okvare zaradi neupoštevanja priloženih navodil, nepravilne uporabe, malomarnega ravnanja z izdelkom in mehanske poškodbe so izvzete iz garancijskih pogojev. **Garancija ne izključuje pravic potrošnika, ki izhajajo iz odgovornosti prodajalca za napake na blagu.**

Vzdrževanje, nadomestne dele in priklopne aparate proizvajalec zagotavlja še 3 leta po preteku garancije.

Servisiranje izvaja proizvajalec sam na sedežu firme CONRAD ELECTRONIC SE, Klaus-Conrad-Strasse 1, Nemčija.

Pokvarjen izdelek pošljete na naslov: Conrad Electronic d.o.o. k.d., Ljubljanska cesta 66, 1290 Grosuplje, skupaj z izpolnjenim garancijskim listom.

Prodajalec: _____

Datum izročitve blaga in žig prodajalca:

Garancija velja od dneva izročitve izdelka, kar kupec dokaže s priloženim, pravilno izpolnjenim garancijskim listom.