



# USB-MINI-STICKS

Hardware-Beschreibung

2023 Januar

# INDEX

<b><u>1. Einleitung</u></b>	<b>9</b>
1.1. Vorwort	10
1.2. Kundenzufriedenheit	10
1.3. Kundenresonanz	10
1.4. Kurzbeschreibung	11
<b><u>2. Hardware Beschreibung</u></b>	<b>13</b>
2.1. Allgemeine technische Daten	14
2.1.1. Technische Daten	14
2.2. Übersichtsbilder	15
2.2.1. Übersichtsbild eines USB-MINI-REL2	15
2.2.2. Übersichtsbild eines USB-MINI-REL4	16
2.2.3. Übersichtsbild eines USB-MINI-OPT8	16
2.2.4. Übersichtsbild eines USB-MINI-TTL8	17
2.3. Pinbelegungen	17
2.3.1. Pinbelegung USB-MINI-REL2	17
2.3.2. Pinbelegung USB-MINI-REL4	18
2.3.3. Pinbelegung USB-MINI-OPT8	19
2.3.4. Pinbelegung USB-MINI-TTL8	20
<b><u>3. Software Beschreibung</u></b>	<b>21</b>
3.1. Benutzen unserer Produkte	22
3.1.1. Ansteuerung über unsere DELIB Treiberbibliothek	22
3.1.2. Ansteuerung über mitgelieferte Testprogramme	22
3.1.3. Ansteuerung auf Protokollebene	23
3.1.4. DELIB CLI (command-line interface) für Windows	24
3.1.4.1. Konfiguration des DELIB CLI	26
3.1.4.2. DELIB CLI Beispiele	27
3.1.5. Ansteuerung über grafische Anwendungen	31
3.1.5.1. LabVIEW	31
3.1.5.2. ProfilLab	31

# INDEX

3.1.5.3. Licht24 Pro	32
<b>3.1.6. Einbinden der DELIB in Programmiersprachen</b>	<b>33</b>
3.1.6.1. Einbinden der DELIB in Visual-C/C++	33
3.1.6.2. Einbinden der DELIB in Visual-C/C++ (Visual Studio 2015)	35
3.1.6.3. Einbinden der DELIB in Visual-C#	38
3.1.6.4. Einbinden der DELIB in Delphi	39
3.1.6.5. Einbinden der DELIB in Visual-Basic (VB)	40
3.1.6.6. Einbinden der DELIB in Visual-Basic.NET (VB.NET)	41
3.1.6.7. Einbinden der DELIB in MS-Office (VBA)	42
3.1.6.8. Einbinden der DELIB in LabVIEW	44
3.1.6.8.1. Einbinden der DELIB in LabVIEW	44
3.1.6.8.2. Verwendung der VIs in LabVIEW	53
3.1.6.8.3. Setzen der Modul-ID in LabVIEW	55
3.1.6.9. Einbinden der DELIB in Java	57
<b>3.2. DELIB Treiberbibliothek</b>	<b>58</b>
3.2.1. Übersicht	59
3.2.1.1. Unterstützte Programmiersprachen	60
3.2.1.2. Unterstützte Betriebssysteme	61
3.2.1.3. SDK-Kit für Programmierer	61
3.2.2. DELIB Setup	62
3.2.3. DELIB Configuration Utility	68
3.2.3.1. Einführung	68
3.2.3.2. Neue Konfiguration erstellen oder vorhandene Konfiguration bearbeiten	69
3.2.3.2.1. Modul Konfiguration USB	70
3.2.3.2.1.1. Beispiel zur Konfiguration identischer USB-Module	72
3.2.3.2.2. Modul Konfiguration Ethernet	77
3.2.3.2.2.1. Automatische Suche	82
3.2.3.2.2.2. Verschlüsselung einrichten	86
3.2.3.2.2.1. Manuelle Konfiguration	86
3.2.3.2.2.2. Automatische Konfiguration	91
3.2.3.2.2.1. Authentifizierung	92

# INDEX

3.2.3.2.3. Modul Konfiguration CAN	96
3.2.3.2.4. Modul Konfiguration Seriell	99
3.2.3.3. Modul testen	101
3.2.3.4. Debug Optionen einstellen	105
3.2.4. Benutzung des Modulselectors	106
3.2.4.1. via USB	106
3.2.4.2. Modul Info	107
3.2.5. DELIB Module Config	109
3.2.5.1. Modul Konfigurationen	109
3.2.5.1.1. Modul-Infoseite	110
3.2.5.1.2. Modul-Identifikation	111
3.2.5.1.3. LAN Netzwerkinformationen	112
3.2.5.1.4. LAN Netzwerkeinstellungen	113
3.2.5.1.5. WiFi Netzwerkinformationen	115
3.2.5.1.6. WiFi Netzwerkeinstellungen	117
3.2.5.1.7. WiFi WPS-Verbindung	119
3.2.5.1.8. NTP-Konfiguration	120
3.2.5.1.9. Serielle Konfiguration	122
3.2.5.1.10. I/O Kanal-Namen	124
3.2.5.1.11. CAN Konfiguration	125
3.2.5.1.11.1, CAN Status	125
3.2.5.1.11.1, CAN Status Interface	125
3.2.5.1.11.2, CAN Statistik TX/RX	126
3.2.5.1.11.2, CAN Main	127
3.2.5.1.11.1, CAN Main Interface	127
3.2.5.1.11.2, CAN Main I/O Init	129
3.2.5.1.11.3, CAN TX/RX Modi	131
3.2.5.1.11.1, CAN TX Mode	131
3.2.5.1.11.2, CAN RX Mode	133
3.2.5.2. I/O-Test	134
3.2.5.2.1. Timeout Test-Funktion	134
3.2.5.2.2. Digital In	135
3.2.5.2.3. Digital Out	136



# INDEX

3.2.5.2.4. Analog In	138
3.2.5.2.5. Analog Out	139
3.2.6. DELIB Module Demo	141
3.2.6.1. Auswahl des Moduls	142
3.2.6.2. Allgemein	143
3.2.6.2.1. Module Info	145
3.2.6.3. Digital Input	146
3.2.6.4. Digital Output	147
3.2.6.5. Analog Input	148
3.2.6.6. Analog Output	149
3.2.7. DT-Flasher	150
3.2.7.1. Über DEDITEC-Firmware	151
3.2.7.2. Auswahl des Moduls	151
3.2.7.3. Firmware Update durchführen	152
3.2.7.3.1. Flash-Files manuell aktualisieren	154
3.3. DELIB Sample Sources (Windows Programmbeispiele)	155
3.3.1. Installation DELIB Sample Sources	156
3.3.2. Benutzung der DELIB Sample Sources	160
3.3.2.1. Schritt 1 - Produktauswahl	161
3.3.2.2. Schritt 2 - Kategorieauswahl	162
3.3.2.3. Schritt 3 - Programmiersprachenauswahl	163
3.3.2.4. Schritt 4 - Quellcode	164
3.4. DELIB für Linux	167
3.4.1. Verwenden der DELIB Treiberbibliothek für Linux	170
3.4.1.1. Delib USB-Sample in Linux	170
3.4.1.2. Delib ETH-Sample in Linux	173
3.4.2. DELIB CLI (command-line interface) für Linux	177
3.4.2.1. Konfiguration des DELIB CLI	180
3.4.2.2. DELIB CLI Beispiele	183
<b>4. DELIB API Referenz</b>	<b>185</b>
4.1. Verfügbare DEDITEC Modul IDs	186

# INDEX

4.2. Verwaltungsfunktionen	189
4.2.1. DapiOpenModule	189
4.2.2. DapiCloseModule	190
4.2.3. DapiGetDELIBVersion	190
4.2.4. DapiSpecialCMDGetModuleConfig	191
4.2.5. DapiOpenModuleEx	194
4.2.6. DapiScanAllModulesAvailable	196
4.3. Fehlerbehandlung	197
4.3.1. DapiGetLastError	197
4.3.2. DapiGetLastErrorText	198
4.3.3. DapiClearLastError	199
4.3.4. DapiGetLastErrorByHandle	200
4.3.5. DapiClearLastErrorByHandle	201
4.4. Digitale Eingänge lesen	202
4.4.1. DapiDIGet1	202
4.4.2. DapiDIGet8	202
4.4.3. DapiDIGet16	203
4.4.4. DapiDIGet32	204
4.4.5. DapiDIGet64	205
4.4.6. DapiDIGetFF32	206
4.4.7. DapiDIGetCounter	207
4.4.8. DapiSpecialCounterLatchAll	208
4.4.9. DapiSpecialCounterLatchAllWithReset	209
4.4.10. DapiSpecialDIFilterValueSet	210
4.4.11. DapiSpecialDIFilterValueGet	211
4.4.12. Dapi_Special_DI_FF_Filter_Value_Set	212
4.4.13. Dapi_Special_DI_FF_Filter_Value_Get	213
4.5. Digitale Ausgänge verwalten	214
4.5.1. DapiDOSet1	214
4.5.2. DapiDOSet8	214
4.5.3. DapiDOSet16	215

# INDEX

4.5.4. DapiDOSet32	216
4.5.5. DapiDOSet64	217
4.5.6. DapiDOSet1_WithTimer	218
4.5.7. DapiDOReadback32	219
4.5.8. DapiDOReadback64	219
4.5.9. DapiDOSetBit32	220
4.5.10. DapiDOClrBit32	221
4.6. TTL-Ein-/Ausgangs Richtungen setzen mit DapiSpecialCommand	222
4.6.1. DAPI_SPECIAL_CMD_SET_DIR_DX_1	222
4.6.2. DAPI_SPECIAL_CMD_SET_DIR_DX_8	224
4.7. Ausgabe-Timeout verwalten	225
4.7.1. DapiSpecialCMDTimeout	225
4.7.1.1. DapiSpecialTimeoutSetValueSec	226
4.7.1.2. DapiSpecialTimeoutActivate	227
4.7.1.3. DapiSpecialTimeoutDeactivate	228
4.7.1.4. DapiSpecialTimeoutGetStatus	229
4.7.1.5. DapiSpecialTimeoutDoValueLoadDefault	230
4.8. Testfunktionen	231
4.8.1. DapiPing	231
4.9. Register Schreib-Befehle	232
4.9.1. DapiWriteByte	232
4.9.2. DapiWriteWord	233
4.9.3. DapiWriteLong	234
4.9.4. DapiWriteLongLong	235
4.10. Register Lese-Befehle	236
4.10.1. DapiReadByte	236
4.10.2. DapiReadWord	237
4.10.3. DapiReadLong	238
4.10.4. DapiReadLongLong	239
4.11. Programmier-Beispiel	240
4.12. Delib Übersichtstabelle	242

# INDEX

<b><u>5. Verzeichnisstruktur der DELIB</u></b>	<b>245</b>
5.1. Include Verzeichnis	248
5.2. Library-Verzeichnis	248
5.3. Library-Verzeichnis für Borland	248
5.4. Umgebungsvariablen	249
<b><u>6. Anhang</u></b>	<b>250</b>
6.1. Kontakt / Support	251
6.2. Umwelt und Entsorgung	251
6.3. Revisionen	252
6.4. Urheberrechte und Marken	253

# Einleitung

---



# **1. Einleitung**

## **1.1. Vorwort**

**Wir beglückwünschen Sie zum Kauf eines hochwertigen DEDITEC Produktes!**

Unsere Produkte werden von unseren Ingenieuren nach den heutigen geforderten Qualitätsanforderungen entwickelt. Wir achten bereits bei der Entwicklung auf flexible Erweiterbarkeit und lange Verfügbarkeit.

**Wir entwickeln modular!**

Durch eine modulare Entwicklung verkürzt sich bei uns die Entwicklungszeit und - was natürlich dem Kunden zu Gute kommt - wir verkaufen zu einem fairen Preis!

**Wir sorgen für eine lange Lieferverfügbarkeit!**

Sollten verwendete Halbleiter nicht mehr verfügbar sein, so können wir schneller reagieren. Bei uns müssen meistens nur Module redesigned werden und nicht das gesamte Produkt. Dies erhöht die Lieferverfügbarkeit.

## **1.2. Kundenzufriedenheit**

**Ein zufriedener Kunde steht bei uns an erster Stelle!**

Sollte mal etwas nicht zu Ihrer Zufriedenheit sein, wenden Sie sich einfach per Telefon oder Mail an uns.

Wir kümmern uns darum!

## **1.3. Kundenresonanz**

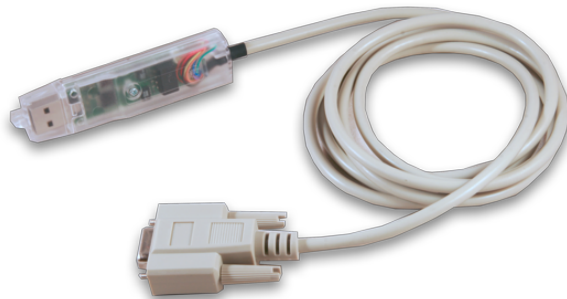
Die besten Produkte wachsen mit unseren Kunden. Für Anregungen oder Vorschläge sind wir jederzeit dankbar.

## 1.4. Kurzbeschreibung

Die USB-Mini-Sticks sind kleine praktische IO-Schaltmodule, die sich über den USB Bus ansprechen lassen.

Mit einfachen Programmierbefehlen können kleine Schalt- und Steueraufgaben einfach vom PC aus realisiert werden.

Die Installation und Inbetriebnahme erfolgt innerhalb weniger Minuten und sollte daher auch für Einsteiger kein Problem darstellen







# Hardware Beschreibung

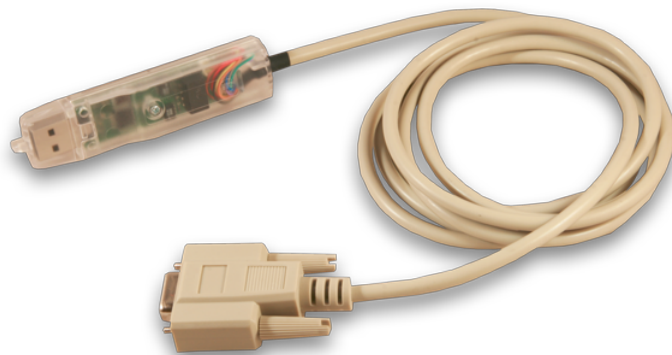
---



## 2. Hardware Beschreibung

### 2.1. Allgemeine technische Daten

#### 2.1.1. Technische Daten



##### **Elektrische Daten:**

Max. Schaltspannung:	30V AC/DC
Max. Schaltstrom:	0,3A bei 30V AC / 1A bei 24V DC

##### **Umgebung:**

Umgebungstemperatur:	+10..+50 °C
Luftfeuchtigkeit:	90 %
Betauung:	Nicht erlaubt

##### **Mechanik:**

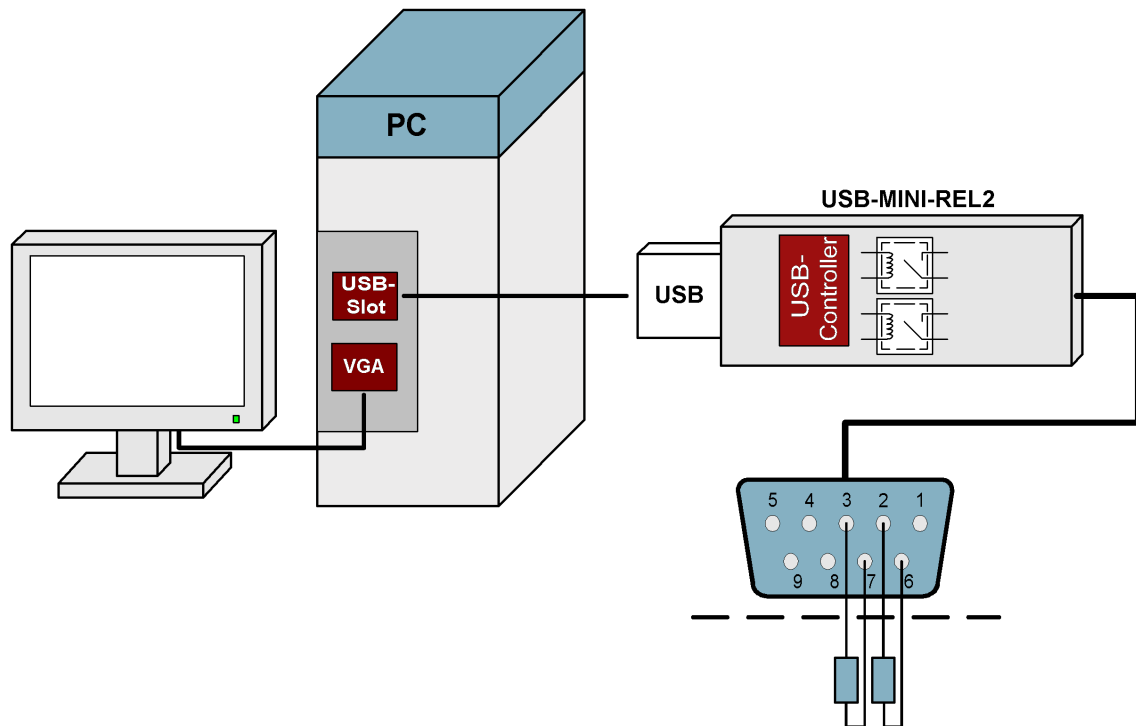
Abmessungen in mm (LxBxH):	84,5 x 21 x 12,5
Anschlusskabel:	ca. 1,8m langes Kabel mit D-SUB9 Buchse
LED:	1 LED blinkt bei USB Aktivität

##### **Mögliche Varianten:**

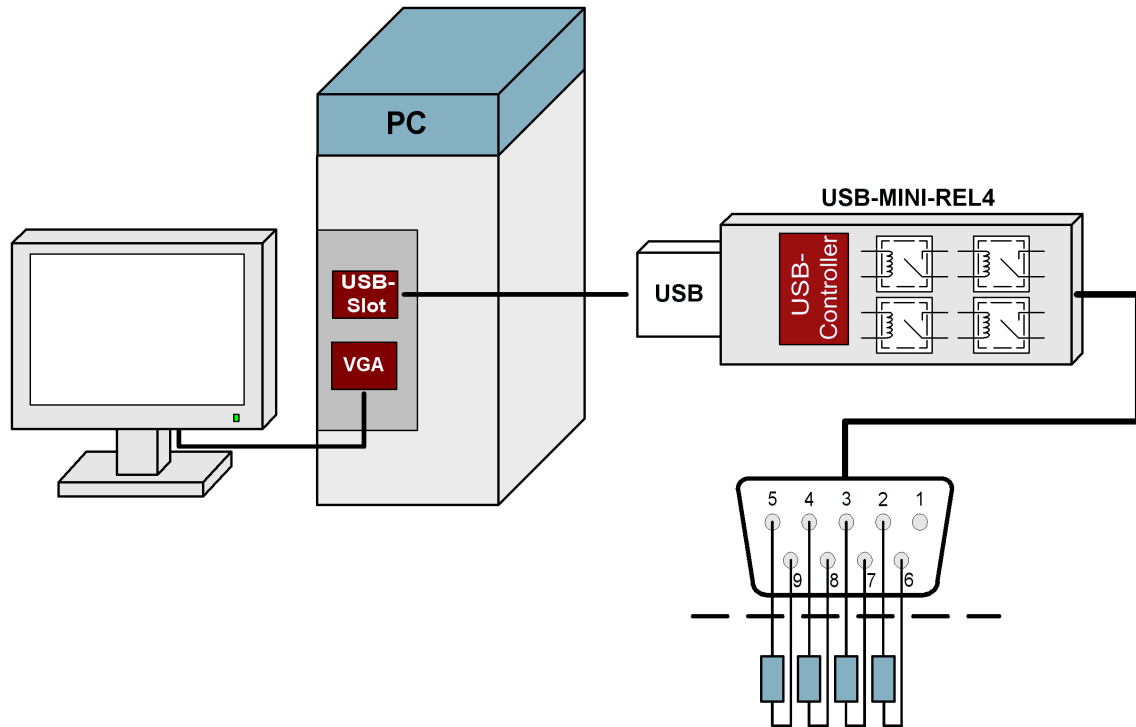
USB-MINI-REL2:	2 Relaisausgänge (Schließer)
USB-MINI-REL4:	4 Relaisausgänge (Schließer)
USB-MINI-OPT8	8 Optokopplereingänge
USB-MINI-TLL8	8 TTL Ein- /Ausgänge Ausgänge: bis zu 5mA je Kanal

## 2.2. Übersichtsbilder

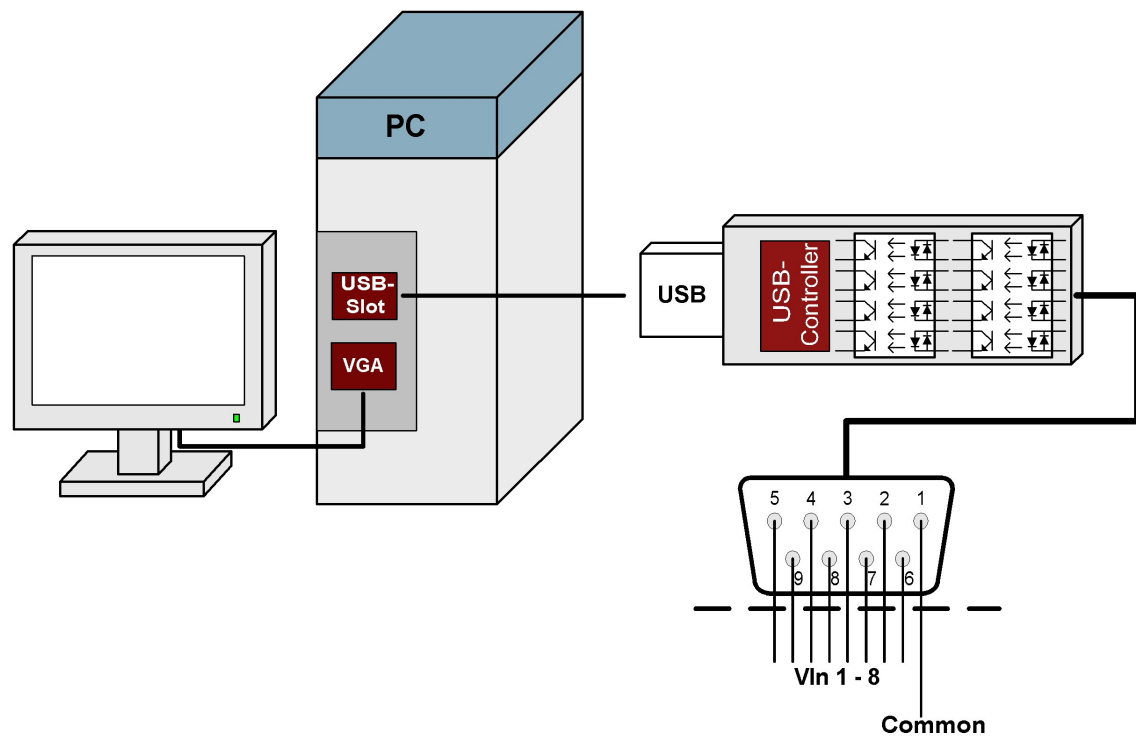
### 2.2.1. Übersichtsbild eines USB-MINI-REL2



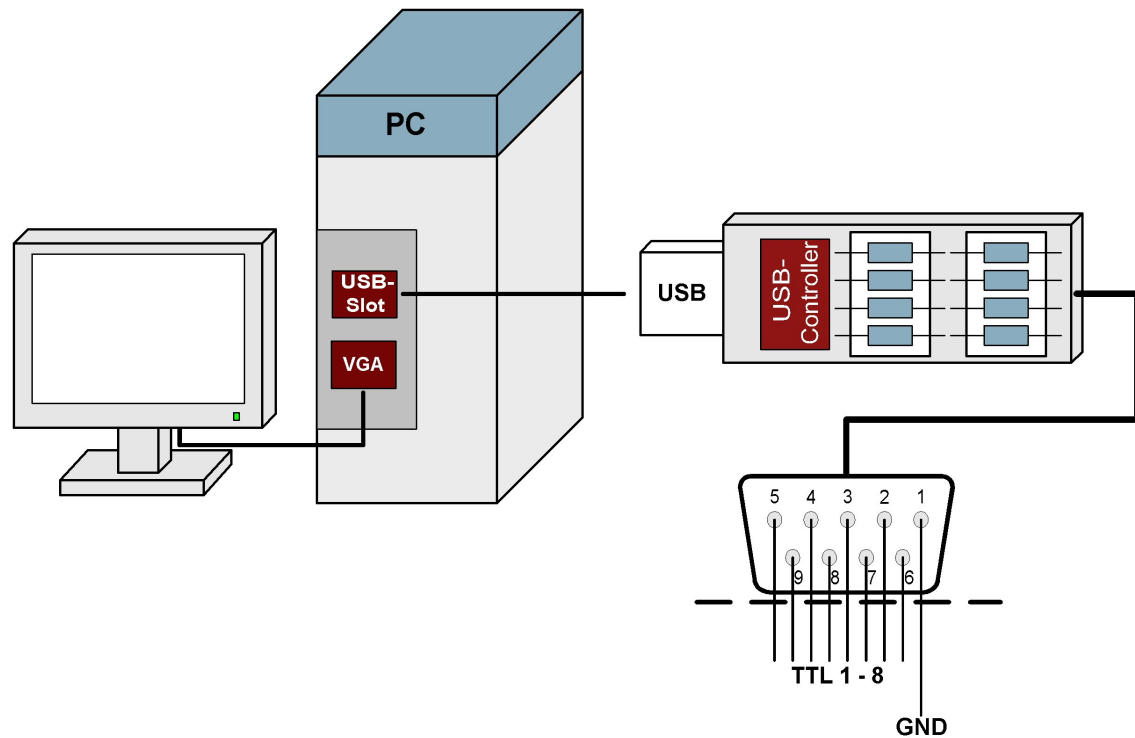
### 2.2.2. Übersichtsbild eines USB-MINI-REL4



### 2.2.3. Übersichtsbild eines USB-MINI-OPT8

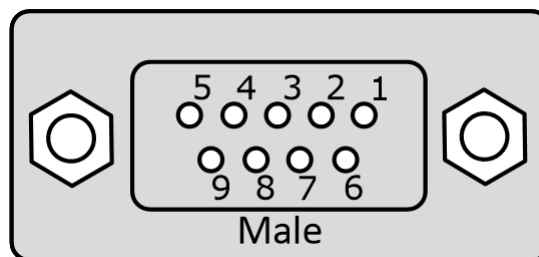


## 2.2.4. Übersichtsbild eines USB-MINI-TTL8



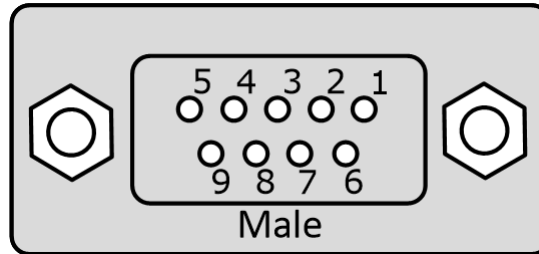
## 2.3. Pinbelegungen

### 2.3.1. Pinbelegung USB-MINI-REL2



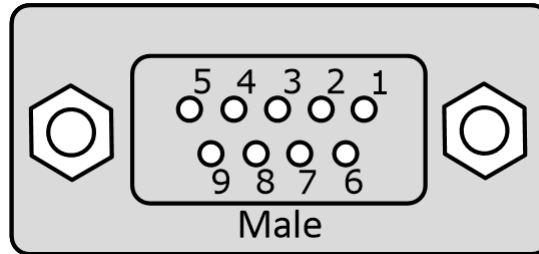
Pin	Beschreibung
3 & 7	Relais 1
2 & 6	Relais 2
1,4,5,8,9	NC

### 2.3.2. Pinbelegung USB-MINI-REL4



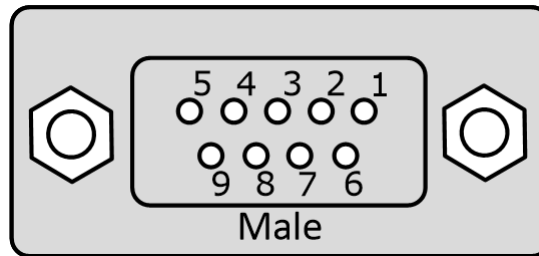
Pin	Beschreibung
5 & 9	Relais 1
4 & 8	Relais 2
3 & 7	Relais 3
2 & 6	Relais 4
1	NC

### 2.3.3. Pinbelegung USB-MINI-OPT8



Pin	Beschreibung
1	Common
5	Opto In 0
9	Opto In 1
4	Opto In 2
8	Opto In 3
3	Opto In 4
7	Opto In 5
2	Opto In 6
6	Opto In 7

#### 2.3.4. Pinbelegung USB-MINI-TTL8



Pin	Beschreibung
1	GND
5	TTL 0
9	TTL 1
4	TTL 2
8	TTL 3
3	TTL 4
7	TTL 5
2	TTL 6
6	TTL 7



# **Software Beschreibung**



## **3. Software Beschreibung**

### **3.1. Benutzen unserer Produkte**

#### **3.1.1. Ansteuerung über unsere DELIB Treiberbibliothek**

Im Lieferumfang unserer DELIB Treiberbibliothek ist die DELIB-API und diverse Programme zur Konfiguration Test unserer Produkte enthalten.

Die API bietet Ihnen Zugriff auf alle Funktionen die Sie zur Kommunikation mit unseren Produkten benötigen.

Im Kapitel **DELIB API Referenz** finden Sie alle Funktionen unserer Treiberbibliothek erklärt und mit Anwendungsbeispielen versehen.

#### **3.1.2. Ansteuerung über mitgelieferte Testprogramme**

Mit unserem DELIB Module Config können Sie unsere Steuer- & Regelungstechnik Produkte ohne großen Konfigurationsaufwand auf Funktionalität testen.

Für ausführliche Informationen siehe Kapitel **DELIB\_Module\_Config**.

### 3.1.3. Ansteuerung auf Protokollebene

Für Produkte mit Ethernet-, CAN- oder Serieller-Schnittstelle bieten wir Ihnen unsere offenen Protokolle an.

Diese Protokolle können ohne unsere DELIB Treiberbibliothek auf Geräten mit entsprechender Schnittstelle verwendet werden. Der Weg über unsere Protokolle sind Betriebssystem unabhängig.

**Unser Handbuch, Protokolle & Registerbelegung finden Sie hier:**

**Download PDF:**

[http://www.deditec.de/pdf/manual\\_d\\_deditec\\_communication\\_protocols.pdf](http://www.deditec.de/pdf/manual_d_deditec_communication_protocols.pdf)

**Online HTML-Manual:**

[http://manuals.deditec.de/de/manual\\_deditec\\_communication\\_protocols/index.html](http://manuals.deditec.de/de/manual_deditec_communication_protocols/index.html)

Dieses Handbuch bietet eine komplette Übersicht über die benötigten Registeradressen unserer Module sowie den Aufbau der verschiedenen Kommunikationsprotokolle.

### 3.1.4. DELIB CLI (command-line interface) für Windows

Da in manchen Programmiersprachen (wie zum Beispiel PHP) keine DLLs eingebunden werden können, gibt es hierzu ein extra Kommandozeilen-Befehl, der direkt aus dem Programm (mit den entsprechenden Parametern) heraus aufgerufen werden kann.

Der DELIB CLI Befehl für Windows befindet sich nach der Installation der DELIB Treiberbibliothek im Verzeichnis C:\Programme\DEDITEC\DELIB\programs\cli\.

#### Definition (Windows)

*delib\_cli command channel [value / unit ["nounit"]]*

**Hinweis:** Die einzelnen Parameter werden nur durch ein Leerzeichen getrennt. Groß und Kleinschreibung wird hierbei nicht beachtet.

#### Parameter

Befehl	Kanal	Wert		unit	nounit
di1	0, 1, 2, ...	-		hex	nounit
di8	0, 8, 16, ...				
di16					
di32					
ff	0, 32, ...	-		hex	nounit
do1	0, 1, 2, ...	0/1 (1-Bit Befehl)		-	-
do8	0, 8, 16, ...	8-Bit Wert	(Bit 0 für Kanal 1, Bit 1 für Kanal 2, ...)		
do16		16-Bit Wert			
do32		32-Bit Wert			

Befehl	Kanal	Wert	unit	nounit
ai	0, 1, 2, ...	-	hex, volt, mA	nounit
ao	0, 1, 2, ...	Ganz oder Hexadezimalzahl (beginnend mit 0x).	-	-

### Return-Wert

#### Zustand der gelesenen digitalen Eingänge

In Kombination mit Parameter unit "hex" wird der Zustand als hex gelesen

#### Zustand der FlipFlips der digitalen Eingänge

In Kombination mit Parameter unit "hex" wird der Zustand als hex gelesen

#### Zustand der gelesenen analogen Eingänge

In Kombination mit Parameter unit "hex" wird der Zustand als hex gelesen

In Kombination mit Parameter unit "volt" wird die Spannung gelesen

In Kombination mit Parameter unit "mA" wird der Strom gelesen

### 3.1.4.1. Konfiguration des DELIB CLI

Vor der ersten Verwendung des DELIB CLI muss die "delib\_cli.cfg" mit einem Texteditor bearbeitet werden.

#### Konfiguration unter Windows

Unter Windows befindet sich die "delib\_cli.cfg" nach der Installation der DELIB-Treiberbibliothek im Verzeichnis "C:\Programme\DEDITEC\DELIB\programs\cli".

#### Inhalt der "delib\_cli.cfg":

```
moduleID=14;  
moduleNR=0;  
RO-ETH_ipAddress=192.168.1.11;
```

#### moduleID

Als moduleID muss die entsprechende Nummer der eingesetzten Hardware eingetragen werden.

Diese Nummer kann der "delib.h" entnommen werden.

Unter Windows finden Sie diese im Verzeichnis C:\Programme\DEDITEC\DELIB\include\.

#### moduleNR

Die moduleNR wird im DELIB Configuration Utility vergeben.

Diese Nummer dient zur Identifizierung identischer Hardware.

Der Standardwert ist die 0.

#### RO-ETH\_ipAddress

Dieser Eintrag wird ausschließlich für die Verbindung zu unseren ETH Modulen benötigt.

Die IP-Adresse der ETH Module können über das DELIB Configuration Utility sowie über die Weboberfläche des Moduls eingestellt werden.

### 3.1.4.2. DELIB CLI Beispiele

#### Digitale Ausgänge

```
delib_cli DO1 17 1
```

→ schaltet das 18. digitale Relais an

```
delib_cli DO1 3 0
```

→ schaltet das 4. digitale Relais aus

```
delib_cli DO8 0 255
```

→ schaltet die digitalen Relais 1 bis 8 an

```
delib_cli DO16 0 0
```

→ schaltet die digitalen Relais 1 bis 16 aus

```
delib_cli DO16 16 65535
```

→ schaltet die digitalen Relais 17 bis 32 an

```
delib_cli DO32 0 4294967295
```

→ schaltet die digitalen Relais 1 bis 32 an

## Digitale Eingänge

```
delib_cli DI1 3
```

Beispiel eines Rückgabewertes: 1

→ lese den Zustand des 4. digitalen Eingangs und gebe ihn zurück

```
delib_cli DI8 0 hex
```

Beispiel eines Rückgabewertes: **0xC8**

(auf den Kanälen 4, 7 und 8 liegt ein Signal an)

→ lese den Wert von digitalen Eingang 1-8 als hexadezimalzahl

```
delib_cli DI16 0 hex
```

Beispiel eines Rückgabewertes: **0xE0C0**

(auf den Kanälen 7,8, 14 ,15 und 16 liegt ein Signal an)

→ lese den Wert von digitalen Eingang 1-16 als hexadezimalzahl

Alternativ kann das Argument "nunit" an alle zu formatierenden Ausgabeanfragen wie folgendermaßen angehängen werden:

```
delib_cli DI8 0 hex nunit
```

Beispiel eines Rückgabewertes: **FF**

(auf den Kanälen 1-8 liegt ein Signal an)

→ lese den Wert von digitalen Eingang 1-8 als hexadezimalzahl

```
delib_cli FF 0
```

Beispiel eines Rückgabewertes: **192**

(auf den Kanälen 7 und 8 wurde eine Zustandsänderung erkannt)

→ lese den Wert der FlipFlops der digitalen Eingänge 1-32

```
delib_cli FF 32
```

Beispiel eines Rückgabewertes: **65535**

(auf den Kanälen 33 bis 64 wurde eine Zustandsänderung erkannt)

→ lese den Wert der FlipFlops der digitalen Eingänge 33-64

```
delib_cli FF 0 hex
```

Beispiel eines Rückgabewertes: **0xD00**

(auf Kanälen 9, 11 und 12 wurde eine Zustandsänderung erkannt)

→ lese den Wert der FlipFlops der digitalen Eingänge 1-32 als hexadezimalzahl



### **Analoge Ausgänge**

```
delib_cli AO 7 4711
```

→ setzt den dezimalen Wert 4711 auf den 8. analogen Ausgang

```
delib_cli AO 6 0x4711
```

→ setzt den hexadezimalen Wert 0x4AF1 auf den 7. analogen Ausgang

```
delib_cli AO 7 3.7V
```

→ setzt die Spannung des 8. analogen Ausgangs auf 3,7 Volt

(sowohl Komma "," als auch Punkt "." können zur Kommatrennung verwendet werden)

```
delib_cli AO 7 13.3mA
```

→ setzt den Strom des 8. analogen Ausgangs auf 13,3 Milliampere

(sowohl Komma "," als auch Punkt "." können zur Kommatrennung verwendet werden)

### Analoge Eingänge

```
delib_cli AI 2
```

Beispiel eines Rückgabewertes: **1234**

→ liest den Wert des 3. analogen Eingangs als Dezimalzahl

```
delib_cli AI 2 hex
```

Beispiel eines Rückgabewertes: **0x1FA**

→ liest den Wert des 3. analogen Eingangs als Hexadezimalzahl

```
delib_cli AI 2 V
```

Beispiel eines Rückgabewertes: **12.500000V**

→ liest die Spannung des 3. analogen Eingangs als Kommazahl

```
delib_cli AI 2 mA
```

Beispiel eines Rückgabewertes: **20.551600mA**

→ liest den Strom des 3. analogen Eingangs als Kommazahl

Alternativ kann auch hier das Argument "nunit" an alle zu formatierenden Ausgabeanfragen wie folgendermaßen angehängt werden:

```
delib_cli AI 3 hex nunit
```

Beispiel eines Rückgabewertes: **1FA**

→ liest den Wert des 4. analogen Eingangs als Hexadezimalzahl

```
delib_cli AI 3 V nunit
```

Beispiel eines Rückgabewertes: **12.500000**

→ liest die Spannung des 4. analogen Eingangs als Kommazahl

### **3.1.5. Ansteuerung über grafische Anwendungen**

#### **3.1.5.1. LabVIEW**

Unsere DELIB API kann in LabVIEW importiert und verwendet werden. Alle Produkte die unsere DELIB API verwenden, sind somit mit LabVIEW kompatibel.

Folgendes Kapitel zeigt, wie Sie die DELIB API in LabVIEW einbinden können:  
Einbinden der DELIB in LabVIEW

#### **3.1.5.2. ProfiLab**

Die ProfiLab Software der Firma Abacom unterstützt eine große Anzahl unserer Steuer- & Regelungstechnik Produkte.

Link zum Hersteller: <http://www.abacom-online.de/html/profilab-expert.html>

**Die folgenden I/Os werden unterstützt:**

##### **Digitale Ein-/Ausgänge**

- Relais
- MOSFET
- Optokoppler
- Bistabile-Relais

##### **Analoge Ein-/Ausgänge**

- Analog zu digital Wandler
- Digital zu analog Wandler

##### **TTL-I/Os**

- 8/32/64 TTL-Kanäle

### **3.1.5.3. Licht24 Pro**

Die Licht24 Pro Software der Firma bksoft unterstützt ebenfalls eine hohe Anzahl unserer Produkte.

Mehr Informationen finden Sie unter: <http://www.bksoft.de/licht24pro.htm>

### 3.1.6. Einbinden der DELIB in Programmiersprachen

#### 3.1.6.1. Einbinden der DELIB in Visual-C/C++

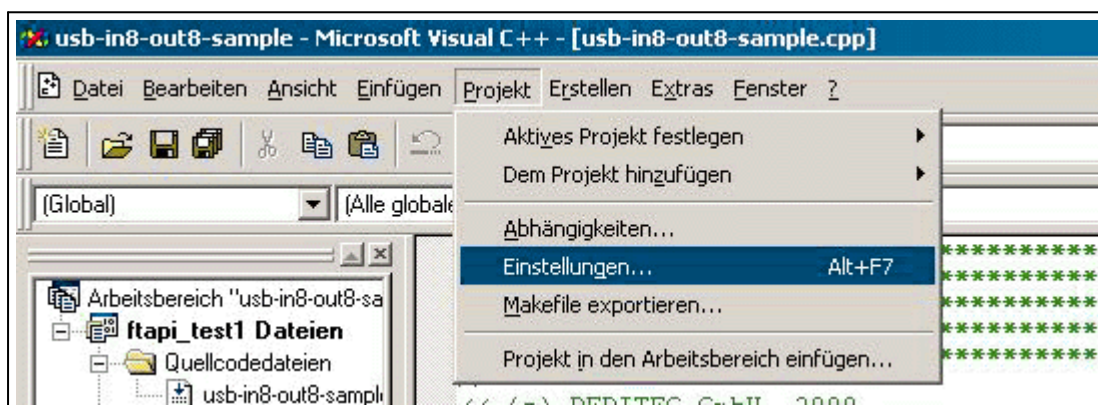
Zur Erleichterung für Verweise auf das DELIB-Include und das DELIB-Lib Verzeichnis werden bei installation der DELIB Umgebungsvariablen definiert.

DELIB\_LIB = C:\Programme\DEDITEC\DELIB\lib

DELIB\_INCLUDE = C:\Programme\DEDITEC\DELIB\include

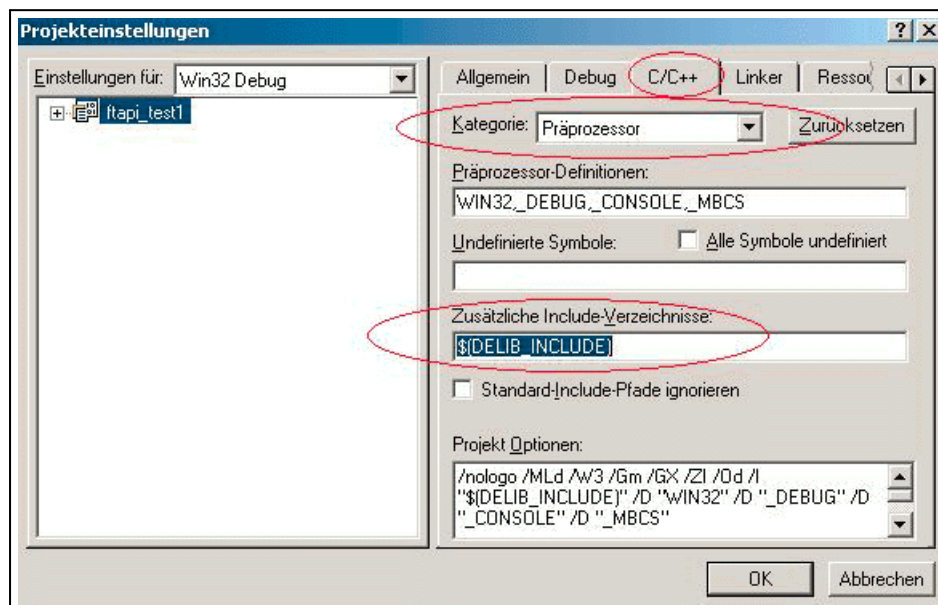
Diese werden im Folgenden in den Projekteinstellungen des Compilers eingetragen.

Visual-C/C++ Starten und im Menue "Projekt → Einstellungen" öffnen.



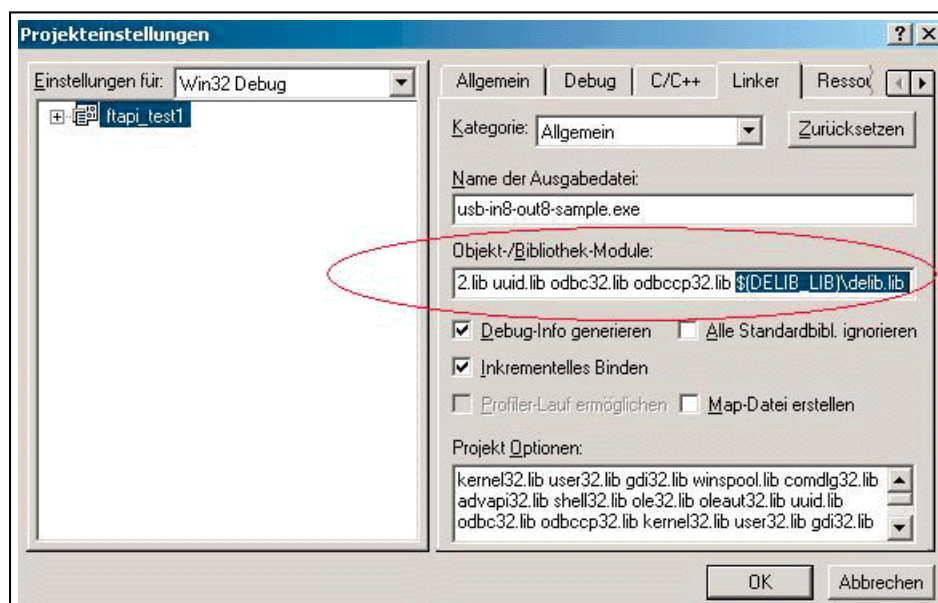
### DELIB.H Eintrag in den Visual-C/C++ Projekt Einstellungen

Unter dem Reiter "C/C++" die "Kategorie" Präprozessor auswählen und unter "Zusätzliche Include Verzeichnisse" "\$(DELIB\_INCLUDE)" eintragen.



### DELIB.LIB Eintrag in den Visual-C/C++ Projekt Einstellungen

Unter dem Reiter "Linker" bei "Objekt-/Bibliothek-Module" die vorhandene Zeile mit der Endung "\$(DELIB\_LIB)\delib.lib" erweitern.



### 3.1.6.2. Einbinden der DELIB in Visual-C/C++ (Visual Studio 2015)

Zur Erleichterung für Verweise auf das DELIB-Include und das DELIB-Lib Verzeichnis werden bei Installation der DELIB Umgebungsvariablen definiert.

#### 32 Bit DELIB Installation

DELIB\_LIB = C:\Programme\DEDITEC\DELIB\lib

DELIB\_INCLUDE = C:\Programme\DEDITEC\DELIB\include

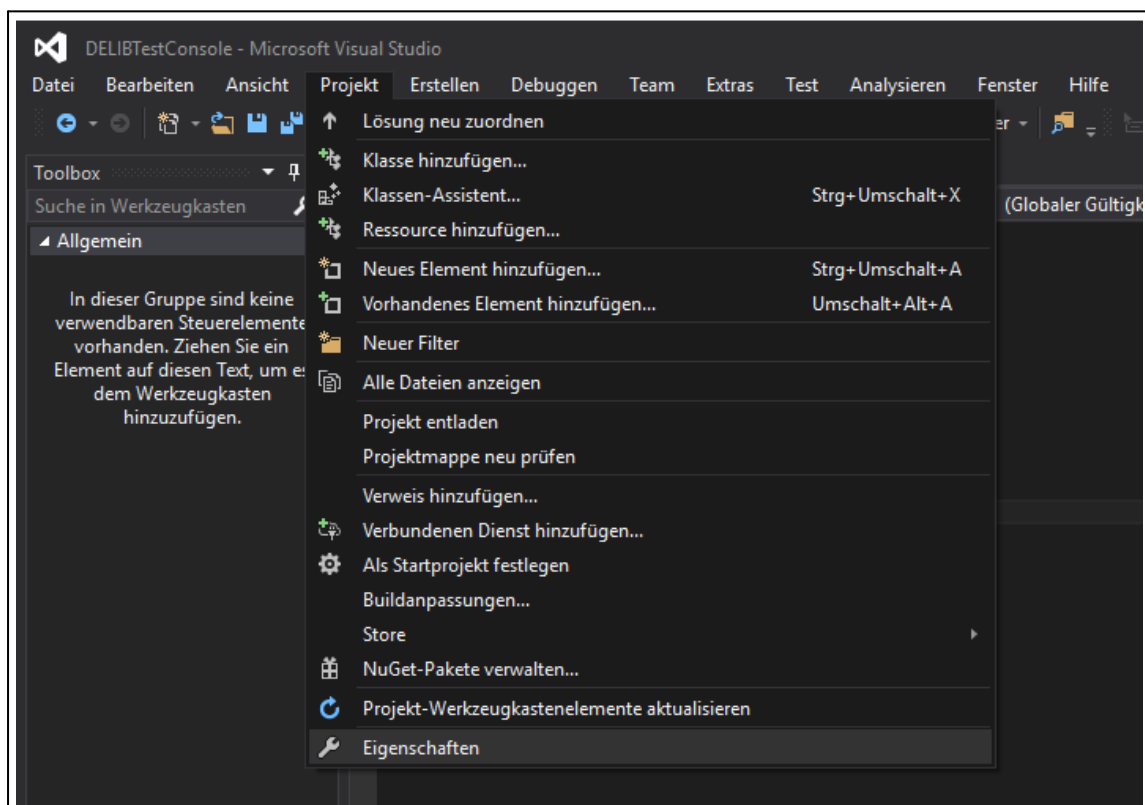
#### 64 Bit DELIB Installation

DELIB64\_LIB = C:\Programme\DEDITEC\DELIB64\lib

DELIB64\_INCLUDE = C:\Programme\DEDITEC\DELIB64\include

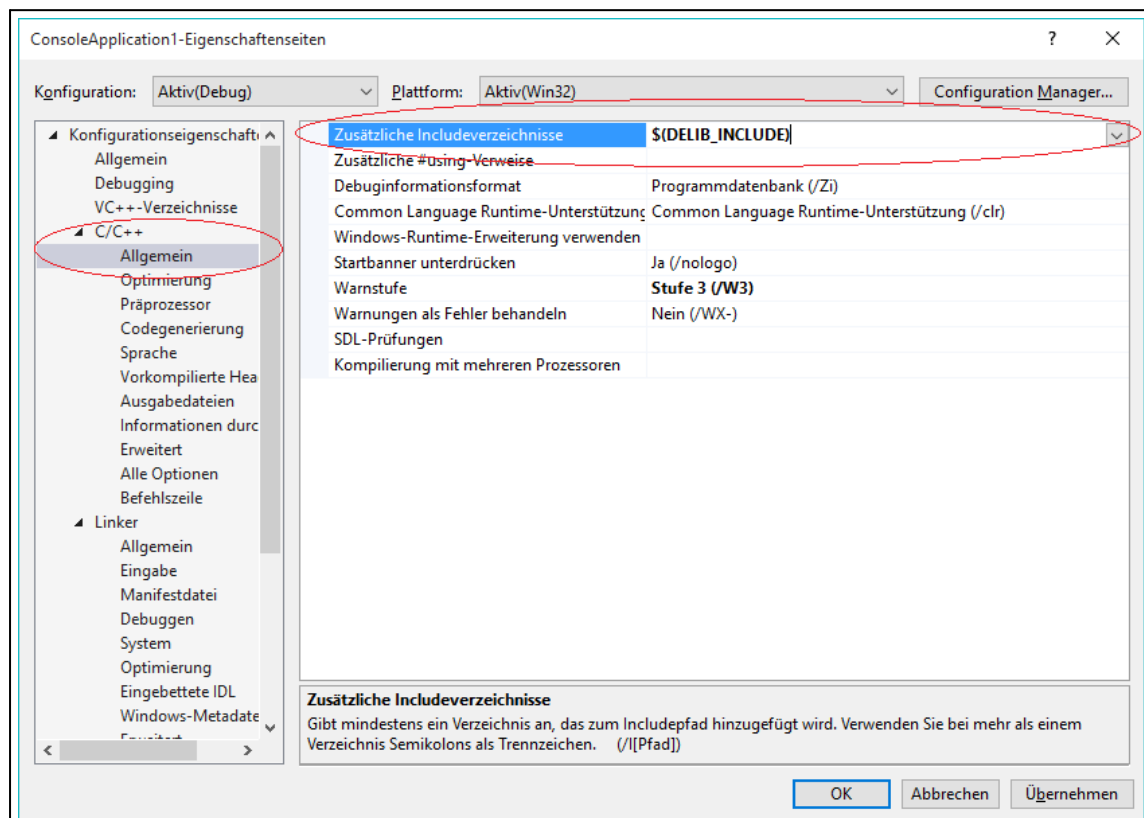
Diese werden im Folgenden in den Projekteinstellungen des Compilers eingetragen.

Visual-C/C++ Starten und im Menue "Projekt → Eigenschaften" öffnen.



DELIB.H Eintrag in den Visual-C/C++ Projekt Einstellungen

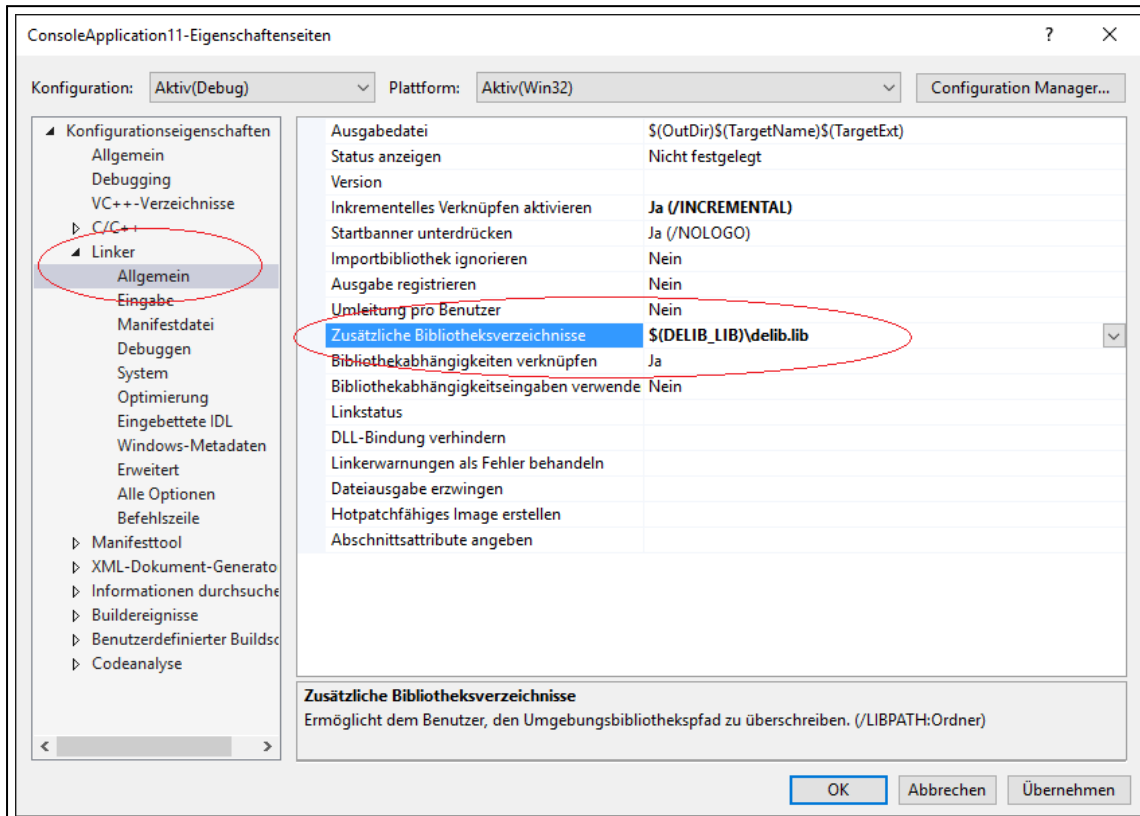
Unter dem Reiter "C/C++" die "Kategorie" Allgemein auswählen und unter "Zusätzliche Include Verzeichnisse" "\$(DELIB\_INCLUDE)" eintragen.





DELIB.LIB Eintrag in den Visual-C/C++ Projekt Einstellungen

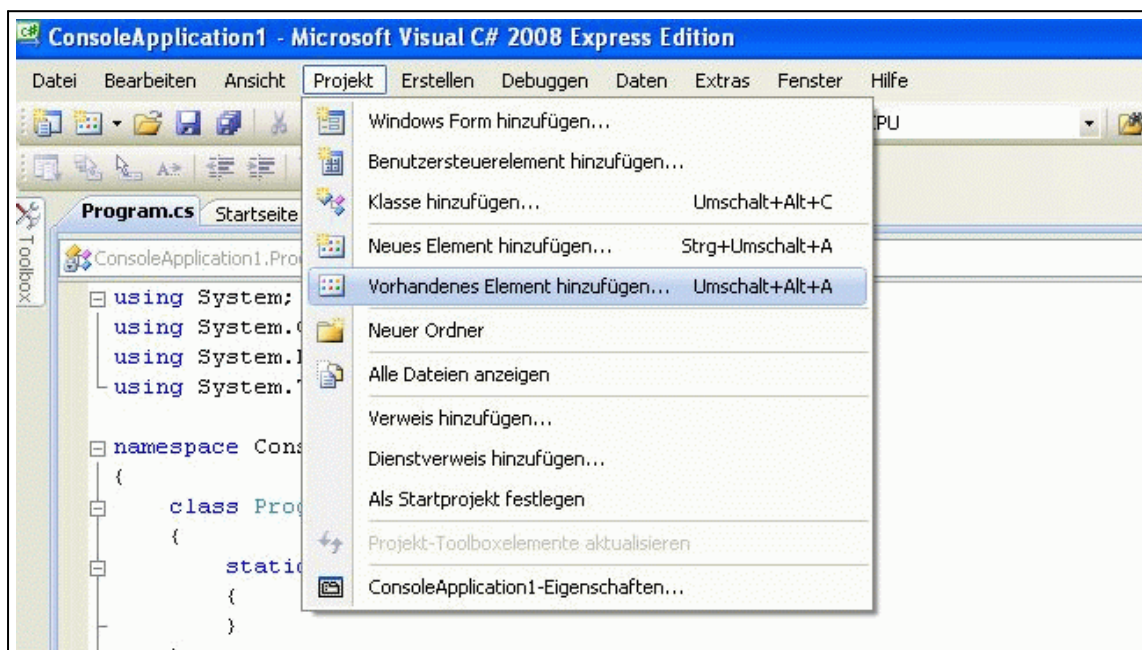
Unter dem Reiter "Linker" bei "Allgemein" "\$ (DELIB\_LIB)\delib.lib" eintragen.



### 3.1.6.3. Einbinden der DELIB in Visual-C#

Die benötigte Datei für Visual-C# befindet sich im Verzeichnis  
C:\Programme\DEDITEC\DELIB\include.

Visual-C# starten und über das Menue "Projekt → Vorhandenes Element hinzufügen" im Verzeichnis C:\Programme\DEDITEC\DELIB\include\ die Datei delib.cs zum Importieren öffnen.



Folgenden Verweis in Ihrem Programm hinzufügen:

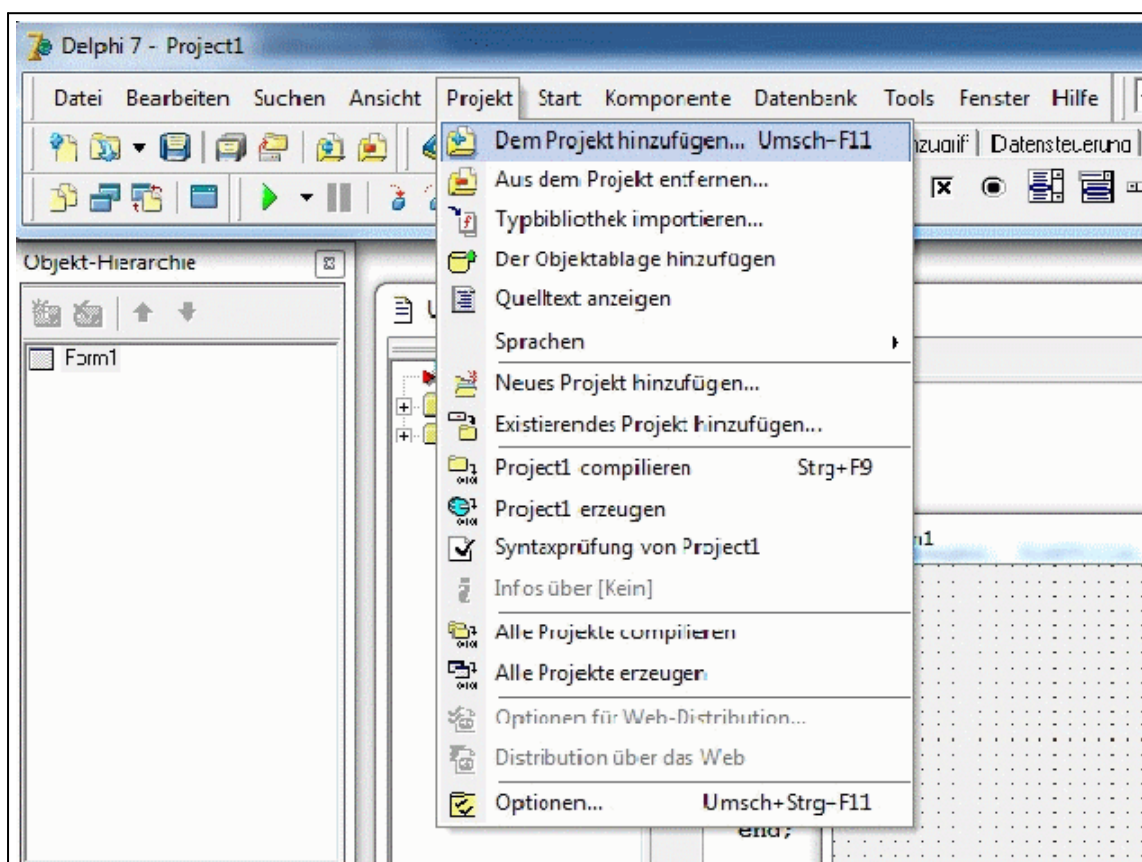
*using DeLib;*

#### 3.1.6.4. Einbinden der DELIB in Delphi

Die benötigte Datei für Delphi befindet sich im Verzeichnis

C:\Programme\DEDITEC\DELIB\include.

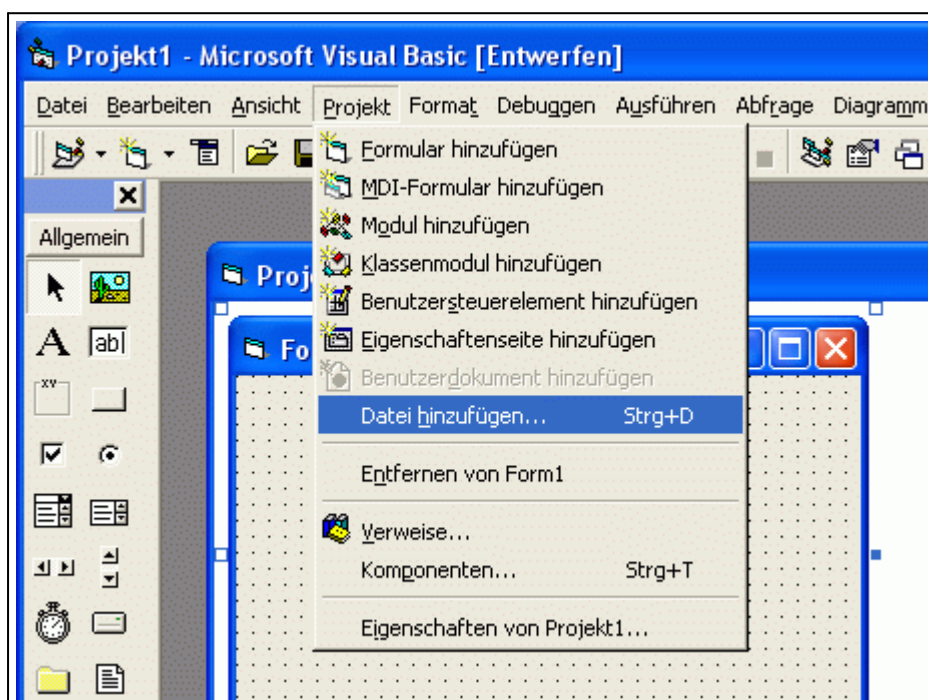
Delphi starten und über das Menue "Projekt → dem Projekt hinzufügen" im Verzeichnis C:\Programme\DEDITEC\DELIB\include\ die Datei delib.pas zum Importieren öffnen.



### 3.1.6.5. Einbinden der DELIB in Visual-Basic (VB)

Die benötigte Datei für Visual-Basic befindet sich im Verzeichnis  
C:\Programme\DEDITEC\DELIB\include.

Visual Basic starten und über das Menue "Projekt → Datei hinzufügen..." im Verzeichnis C:\Programme\DEDITEC\DELIB\include\ die Datei delib.bas zum Importieren öffnen.

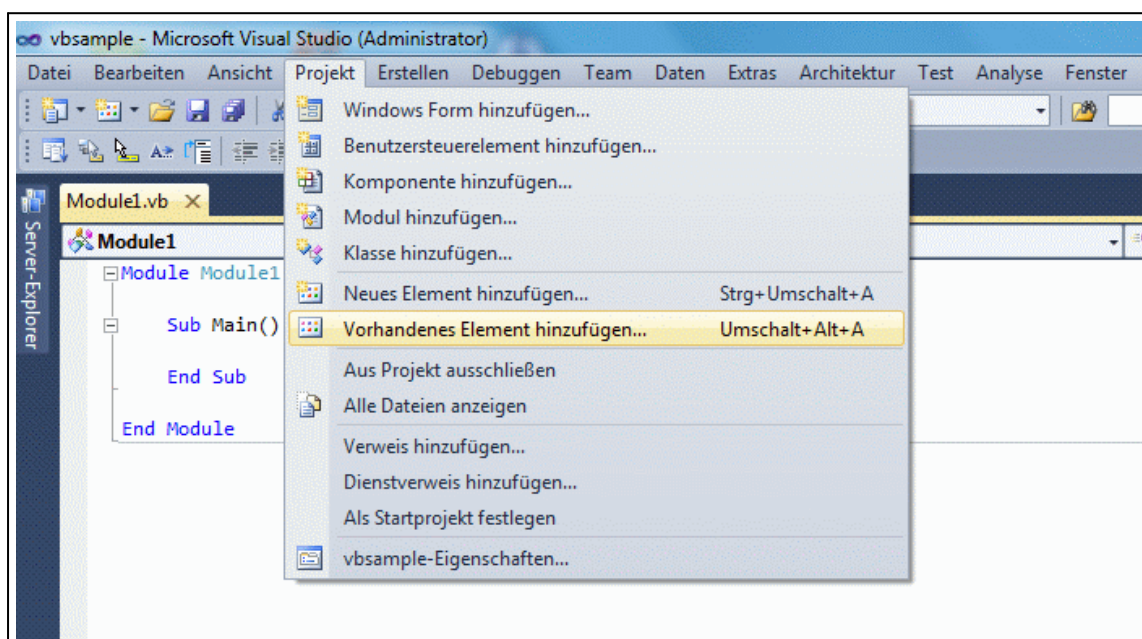


### 3.1.6.6. Einbinden der DELIB in Visual-Basic.NET (VB.NET)

Die benötigte Datei für VB.NET befindet sich im Verzeichnis

C:\Programme\DEDITEC\DELIB\include.

VB.NET starten und über das Menue "Projekt → Vorhandenes Element hinzufügen" im Verzeichnis C:\Programme\DEDITEC\DELIB\include\ die Datei delib.vb zum Importieren öffnen.

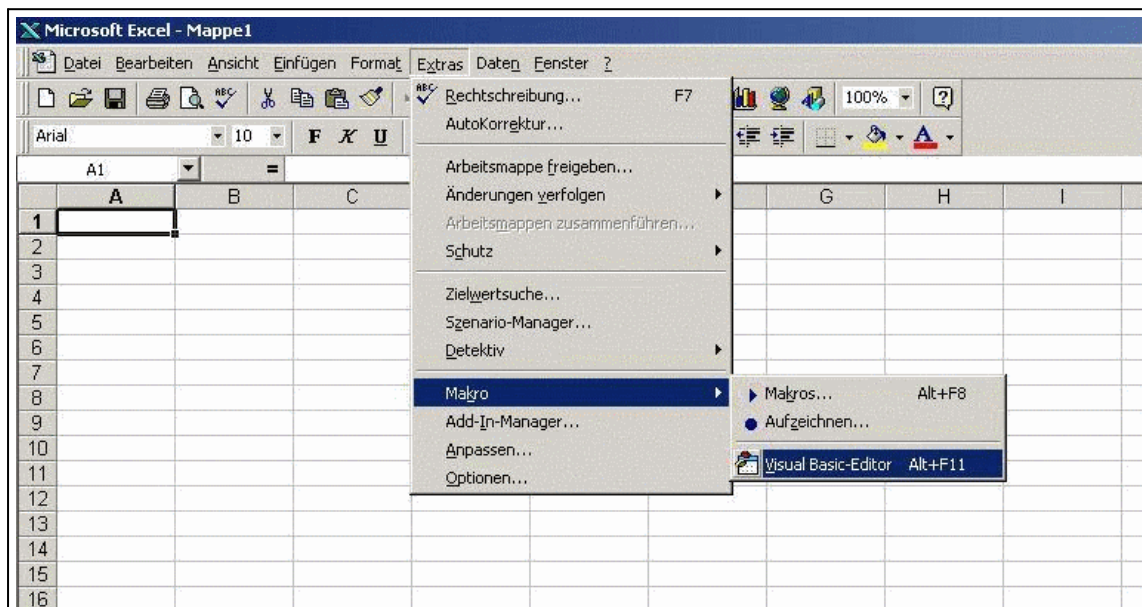


### 3.1.6.7. Einbinden der DELIB in MS-Office (VBA)

Die benötigte Datei für VBA befindet sich im Verzeichnis

C:\Programme\DEDITEC\DELIB\include.

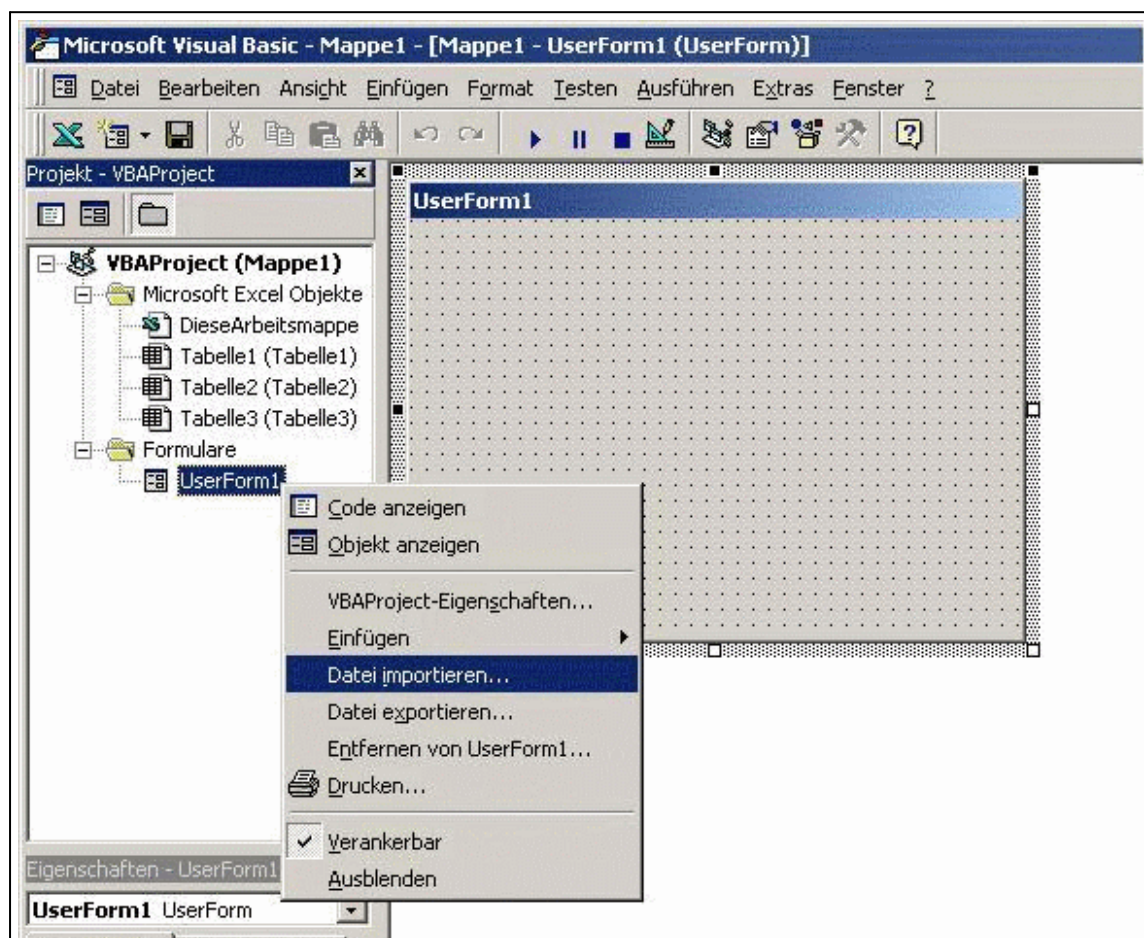
Microsoft Excel starten und über das Menue "Extras → Makro → Visual Basic Editor" öffnen.





## Erstellen der UserForm

Ein neues Arbeitsblatt (UserForm) über das Menue "Einfügen → UserForm" erstellen. Oben links im Projektmanager einen Rechtsklick auf "UserForm → Datei importieren". Im Verzeichnis C:\Programme\DEDITEC\DELIB\include die Datei delib.bas zum importieren öffnen.



### 3.1.6.8. Einbinden der DELIB in LabVIEW

#### 3.1.6.8.1. Einbinden der DELIB in LabVIEW

Das LabVIEW-Beispielprogramm "Deditec\_Modul\_Control.vi" ist keine EXE-Datei und benötigt deshalb zur Ausführung die LabVIEW Entwicklungsumgebung.

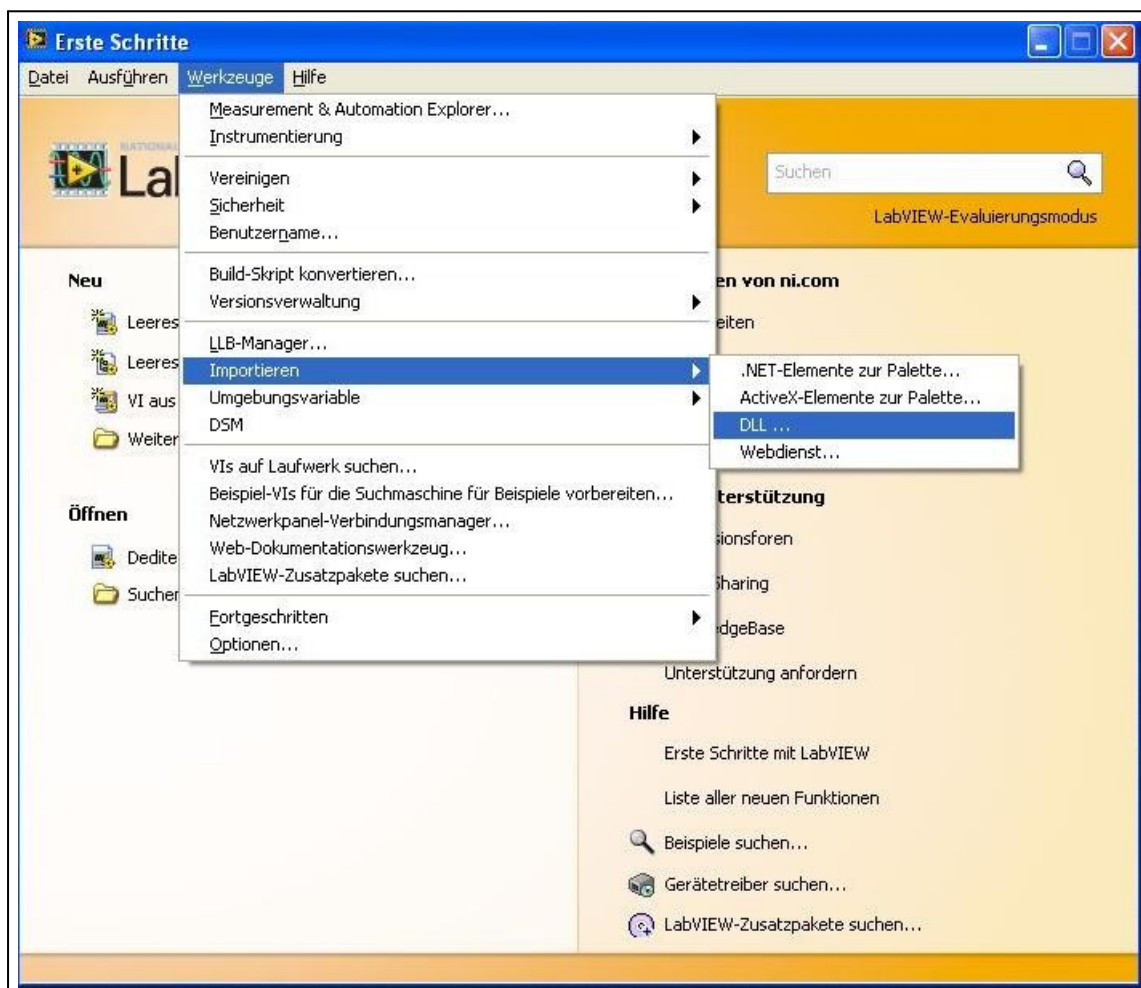
Beschreibung der Einbindung der "delib.dll" in LabVIEW Version 11

- Die benötigten Dateien für LabVIEW befinden sich im Verzeichnis

C:\Windows\System32\delib.dll und in

C:\Programme\DEDITEC\DELIB\include\delib.h

- LabVIEW starten und folgende Option auswählen "Werkzeuge → Importieren → DLL ..."





- Wählen Sie den Punkt "VIs für DLL erstellen" und drücken auf "Weiter"

**DLL importieren**

Erstellungs- oder Aktualisierungsmodus angeben

☒ VIs für DLL erstellen  
Erzeugt VIs basierend auf der vorliegenden DLL- und Header-Datei.

☐ VIs für DLL ändern  
Aktualisiert zuvor importierte VIs für die folgenden Projektbibliotheken

Projekt	DLL-Datei	Datum

Zurück Weiter Abbrechen Hilfe

- Im nächsten Fenster über die Browser-Buttons den Speicherort der delib.dll und der delib.h Datei angeben und mit "Weiter" fortfahren.

**DLL importieren**

DLL- und Header-Datei wählen

NATIONAL INSTRUMENTS

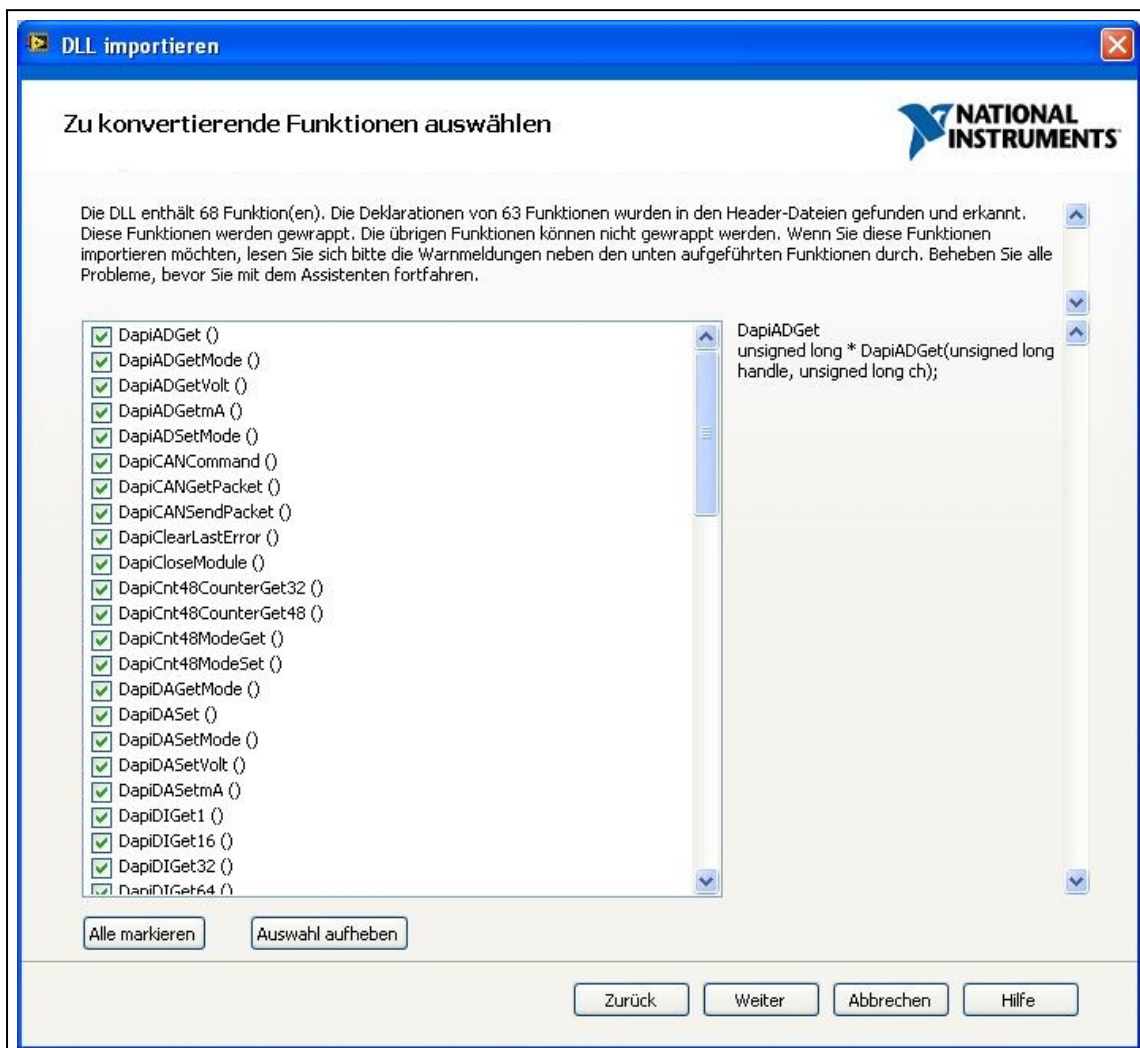
Bibliothek (\*.dll)  
C:\WINDOWS\system32\delib.dll

☐ DLL befindet sich nicht auf lokalem Computer

Header-Datei (.h)  
C:\Programme\DEDITEC\DELIB\include\delib.h

Zurück Weiter Abbrechen Hilfe

- Nochmals auf "Weiter" klicken um fortzufahren.
- Die Header-Datei wird nun analysiert. Anschließend fahren Sie im folgendem Fenster wieder mit "Weiter" fort.



- Den weiteren Anweisungen folgen, bzw. die Konfiguration und den Speicherort für die VIs anpassen.

**DLL importieren**

Einstellungen der Projektbibliothek konfigurieren

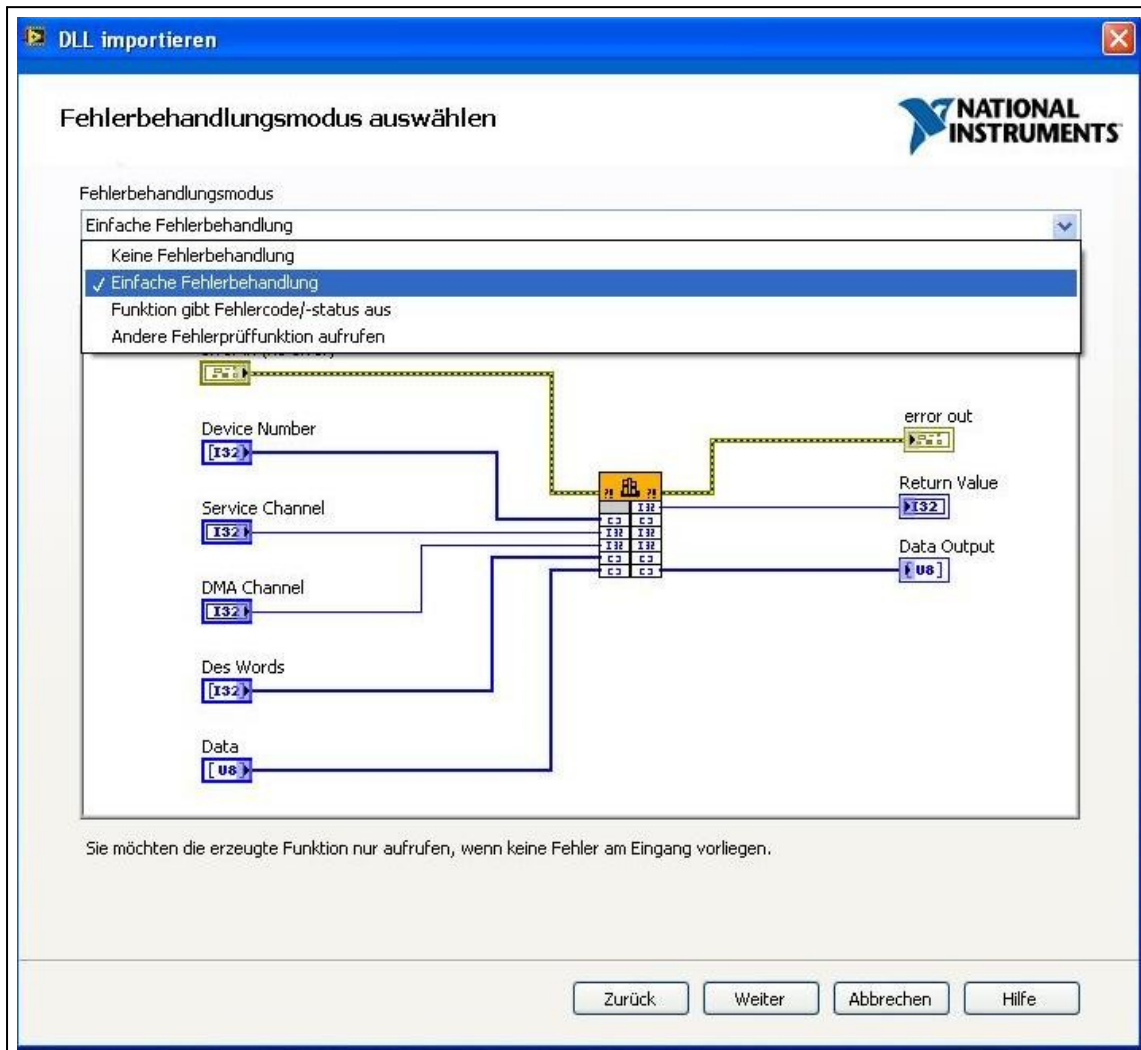
Projektbibliotheksname (.lvlib)  
delib

Pfad zur Projektbibliothek  
C:\Programme\National Instruments\LabVIEW 2011\user.lib\delib

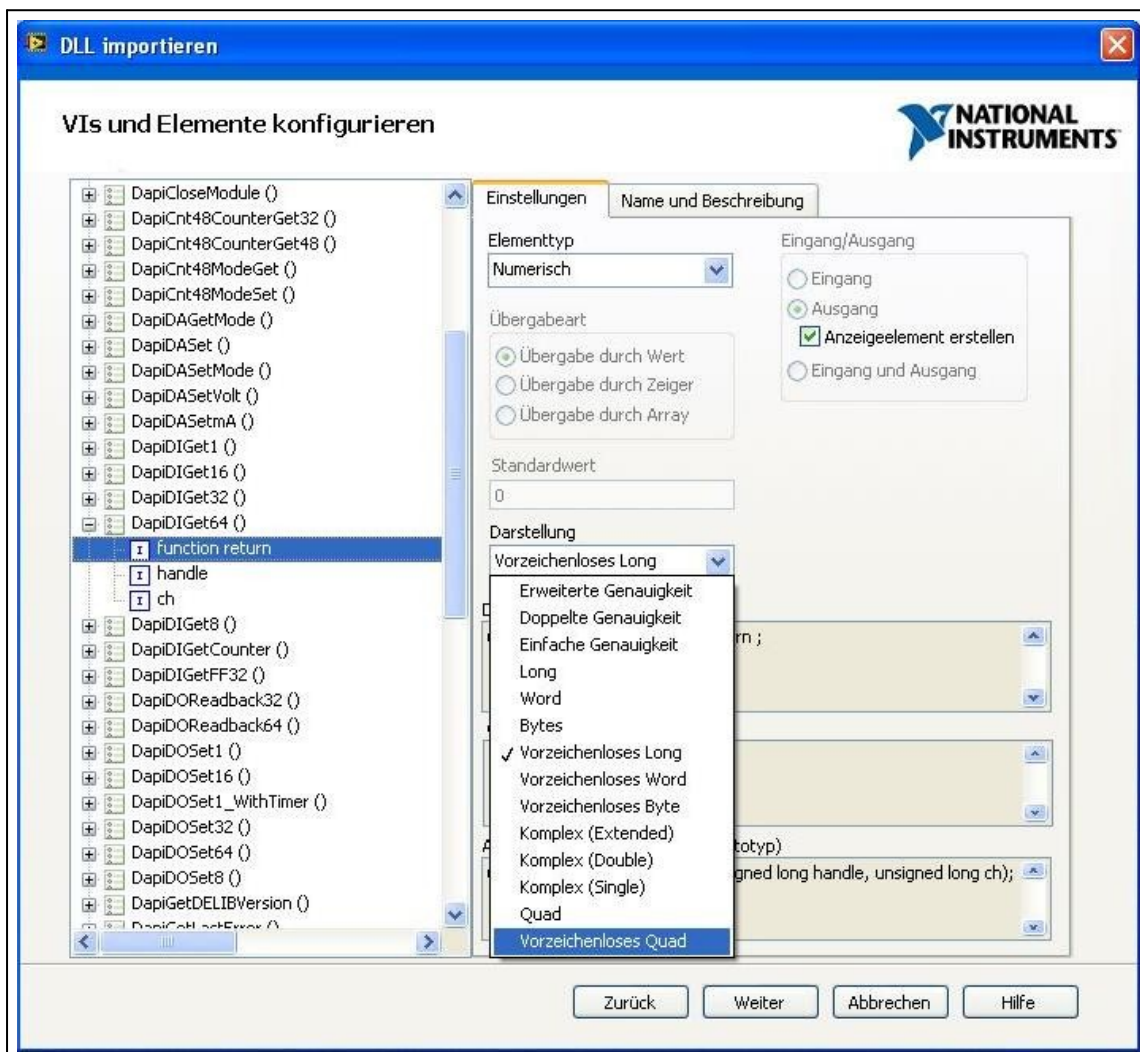
☐ DLL in Zielverzeichnis kopieren

Zurück Weiter Abbrechen Hilfe

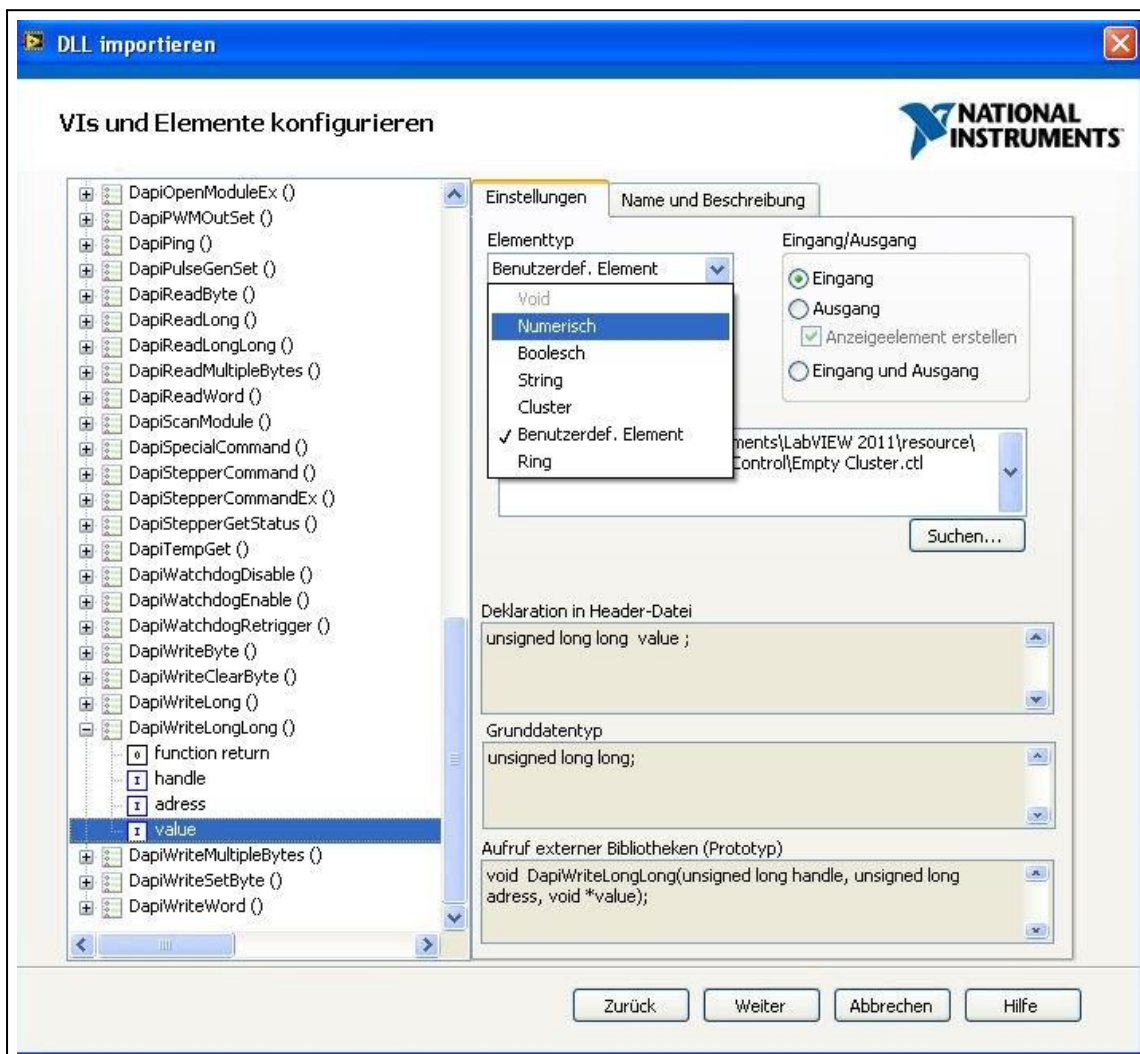
- Im folgendem Fenster wählen Sie im Drop-Down-Menü die Option "Einfache Fehlerbehandlung" aus und fahren mit "Weiter" fort.



- Bei VIs die mit 64-Bit Werten arbeiten, muss die Darstellung von "Vorzeichenloses Long" in "Vorzeichenloses Quad" geändert werden.
- Folgende VIs müssen bearbeitet werden:
  - DapiCNT48CounterGet48 (function return)
  - DapiDIGet64 (function return)
  - DapiDOSet64 (data)
  - DapiDOReadBack64 (function return)

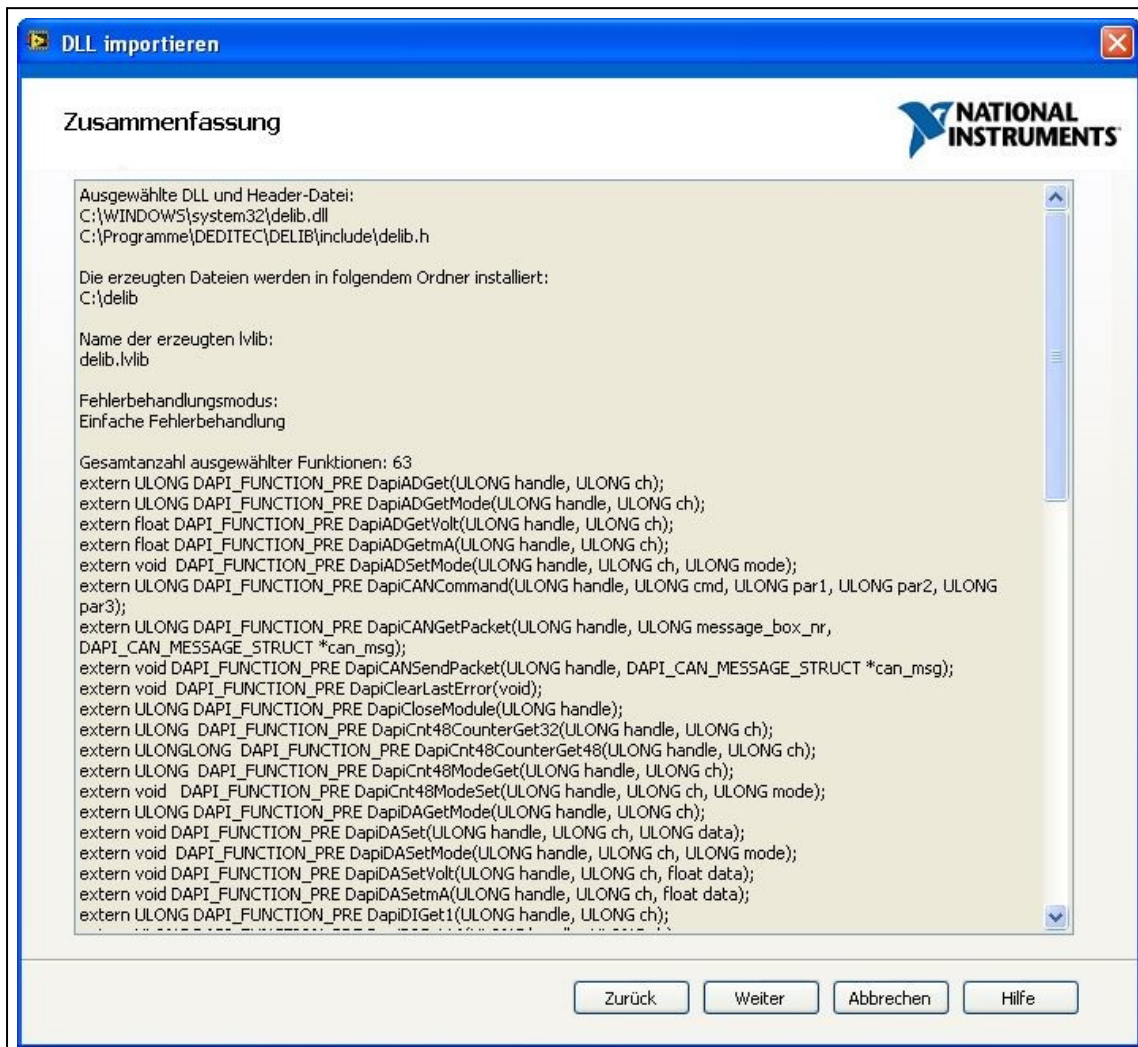


- Bei manchen VIs muss zusätzlich noch der Elementtyp auf "Numerisch" geändert werden und anschließend die Darstellung auf "Vorzeichenloses Quad"
- Folgende VIs müssen bearbeitet werden:
  - DapiWriteLongLong (value)
  - DapiReadLongLong (function return)





- Es erscheint eine Zusammenfassung der ausgeführten Schritte.
- Zum Fortfahren auf "Weiter" drücken.



- Die VIs werden nun erzeugt und können verwendet werden.



### 3.1.6.8.2. Verwendung der VIs in LabVIEW

In unseren Beispielprogrammen werden bei manchen Funktionen sogenannte Defines als Übergabeparameter verwendet.

Diese Defines werden in LabVIEW nicht unterstützt.

Dieses Beispiel soll zeigen, wie solche Funktionen in LabVIEW genutzt werden können.

Als Beispiel dient uns hierbei die Funktion zur Konfiguration des Spannungsbereiches eines A/D Wandlers.

**Die Definition für die Funktion lautet:**

*void DapiADSetMode(ULONG handle, ULONG ch, ULONG mode);*

Für die Funktion sind die Spannungsbereiche in der DELIB Treiberbibliothek bereits vordefiniert.

```
// -----  
// A/D and D/A Modes  
  
#define ADDA_MODE_UNIPOL_10V 0x00  
#define ADDA_MODE_UNIPOL_5V 0x01  
#define ADDA_MODE_UNIPOL_2V5 0x02
```

**Beispielcode in C/C++:**

*DapiADSetMode(handle, 0, ADDA\_MODE\_UNIPOL\_5V);*

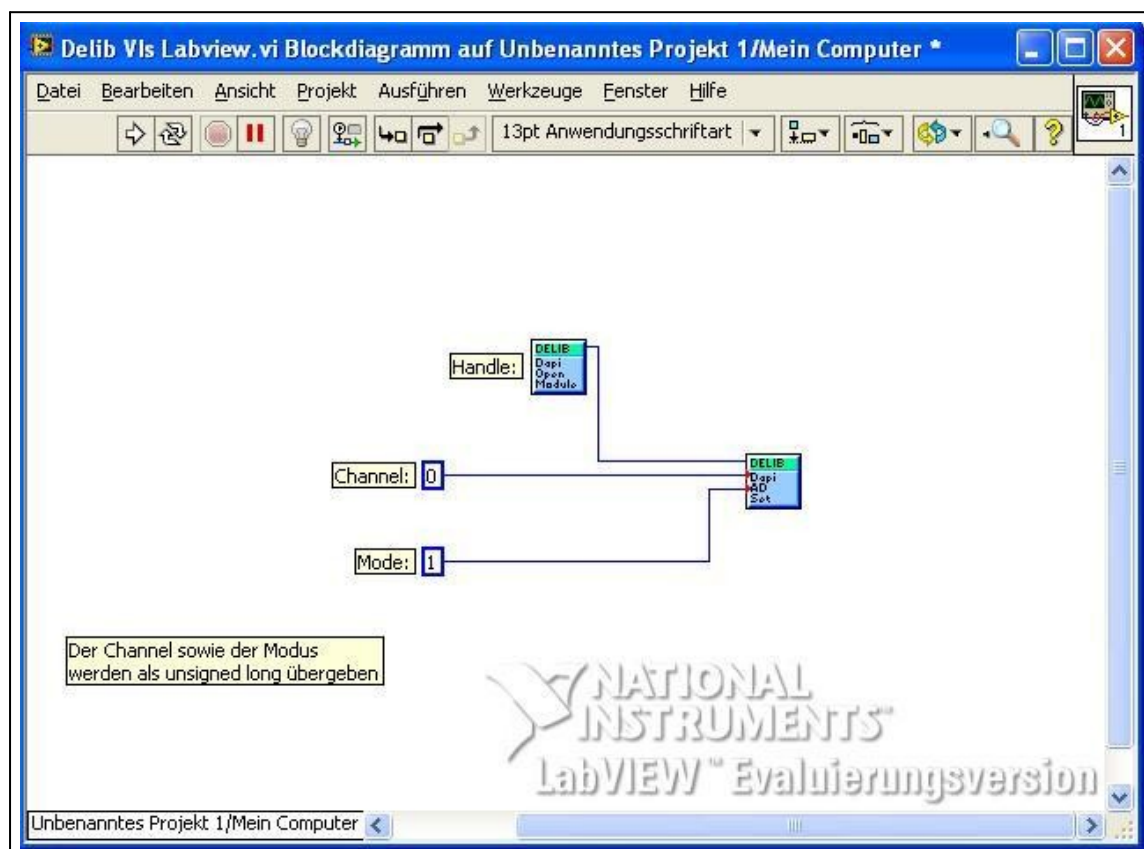
**Alternativ kann man auch folgende Schreibweise verwenden:**

*DapiADSetMode(handle, 0, 0x01);*

Hierbei wurde der Hexadezimalwert, den Sie aus der delib.h Datei entnehmen können, als Parameter für den Modus übergeben

Die delib.h Datei finden sie nach der Installation der DELIB Treiberbibliothek im Verzeichnis C:\Programme\Deditec\DELIB\Include

In LabVIEW könnte die Funktion dann so aussehen:



### 3.1.6.8.3. Setzen der Modul-ID in LabVIEW

Im folgendem Beispiel wird das Ansprechen eines RO-ETH-Moduls in LabVIEW gezeigt.

Die Verbindung zum Modul wird mittels der Funktion DapiOpenModule hergestellt.

**Die Definition für diese Funktion lautet:**

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

Als Parameter für moduleID wird üblicherweise die Modul-ID (z.B. "RO\_ETH") des verwendeten Moduls übergeben.

Eine Übersicht aller möglichen Modul-IDs kann der Datei "delib.h" entnommen werden.

Die delib.h finden Sie nach der Installation der DELIB Treiberbibliothek im Verzeichnis C:\Programme\Deditec\DELIB\Include

```
// *****
// *****
//
//
#define DELIB_VERSION                0x0141    // Actual DELIB-Version

// all Modul-ID's
#define USB_Interface8               1         // USB-Controller8/USB-TTL-IN8-OUT8
#define USB_CAN_STICK                2         // USB-CAN-Stick
#define USB_LOGI_500                 3         // USB-LOGI-500/USB-LOGI-250
#define RO_USB2                      4         // RO-CPU2 / 480 MBit/sec
#define RO_SER                       5         // RO-SER-Serie
#define USB_BITP_200                 6         // USB-BITP-200
#define RO_USB1                      7         // RO-USB-Serie
#define RO_USB                       7         // RO-USB-Serie
#define RO_ETH                       8         // RO-ETH-Serie
#define USB_MINI_STICK               9         // USB-MINI-Stick-Serie
#define USB_LOGI_18                  10        // USB-LOGI-100
#define RO_CAN                       11        // RO-CAN-Serie
#define USB_SPI_MON                  12        // USB_SPI_MON
#define USB_WATCHDOG                 13        // USB_Watchdog
#define USB_OPTOIN_8                 14        // USB-OPTOIN8 / USB-RELAIS-8
#define USB_RELAIS_8                 14        // USB-OPTOIN8 / USB-RELAIS-8
#define USB_OPTOIN_8_RELAIS_8        15        // USB-OPTOIN-8-RELAIS-8
#define USB_OPTOIN_16_RELAIS_16      16        // USB-OPTOIN-16-RELAIS-16
#define USB_OPTOIN_32                 16        // USB-OPTOIN-16-RELAIS-16
#define USB_RELAIS_32                 16        // USB-OPTOIN-16-RELAIS-16
#define USB_OPTOIN_32_RELAIS_32      17        // USB-OPTOIN-32-RELAIS-32
#define USB_OPTOIN_64                 17        // USB-OPTOIN-32-RELAIS-32
#define USB_RELAIS_64                 17        // USB-OPTOIN-32-RELAIS-32
#define USB_TTL_32                   18        // USB-TTL-32
#define USB_TTL_64                   18        // USB-TTL-64

#define MAX_NR_OF_MODULES 18
```

### Beispiel in C:

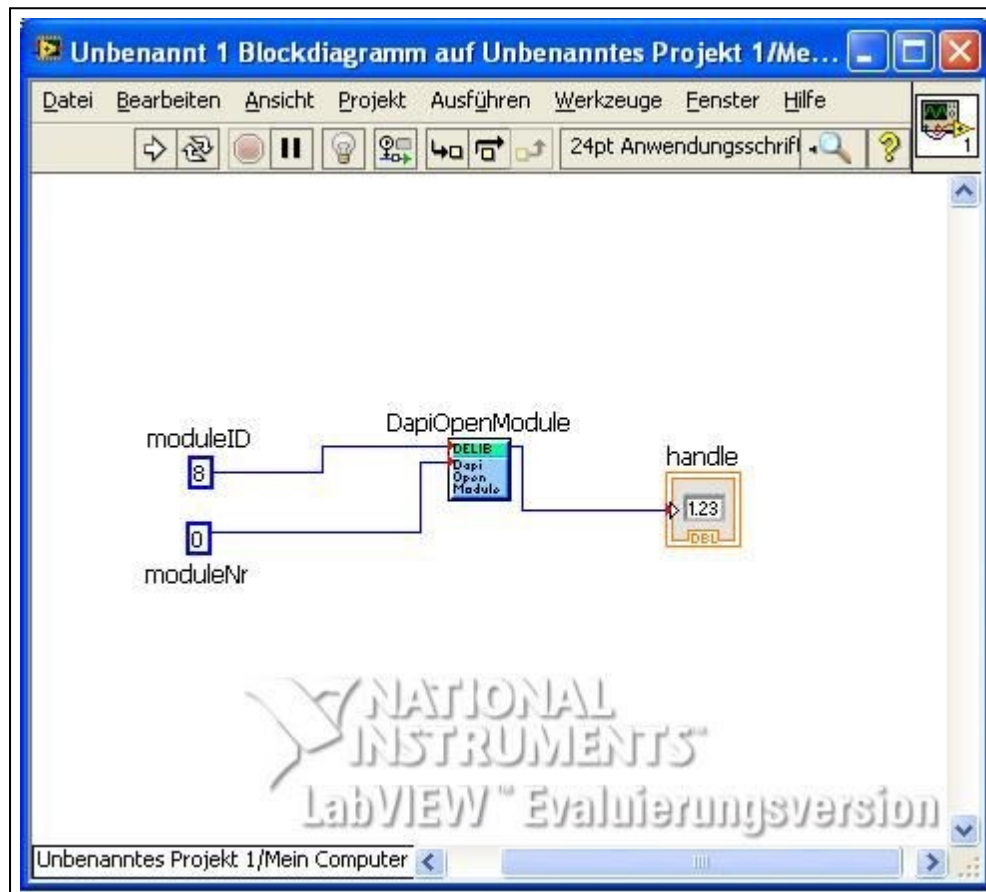
```
handle = DapiOpenModule(RO_ETH, 0);    // öffnet ein RO-ETH-Modul mit  
Modul-Nr 0.
```

### Alternativ kann man auch folgende Schreibweise verwenden:

```
handle = DapiOpenModule(8, 0);
```

Da es in LabVIEW nicht möglich ist, diese "C-Defines" als Parameter für die Funktion DapiOpenModule zu übergeben, muss hier die alternative Schreibweise verwendet werden.

### Beispiel in Labview:



### 3.1.6.9. Einbinden der DELIB in Java

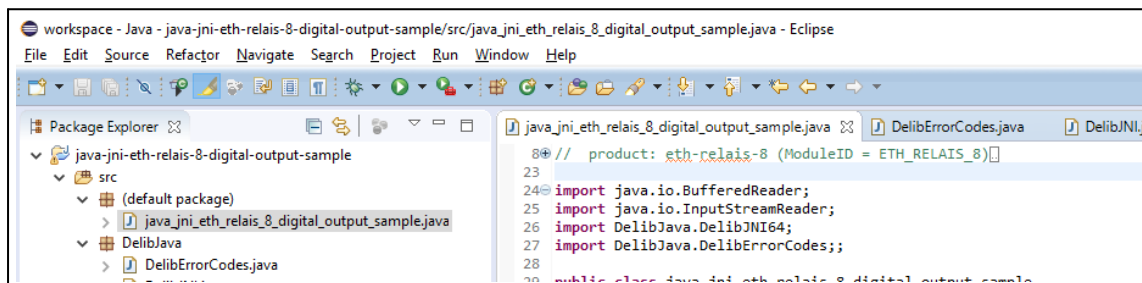
Die benötigten Dateien für Java befinden sich, je nach DELIB-Installation, in folgendem Verzeichnis

C:\Program Files (x86)\DEDITEC\DELIB\include\DelibJava (32 Bit Installation)

C:\Program Files\DEDITEC\DELIB64\include\DelibJava (64 Bit Installation)

Wird Eclipse verwendet, kann der DelibJava-Ordner einfach per Drag&Drop dem Projekt hinzugefügt werden.

Anschließend müssen die verwendeten Module noch importiert werden.



## 3.2. DELIB Treiberbibliothek

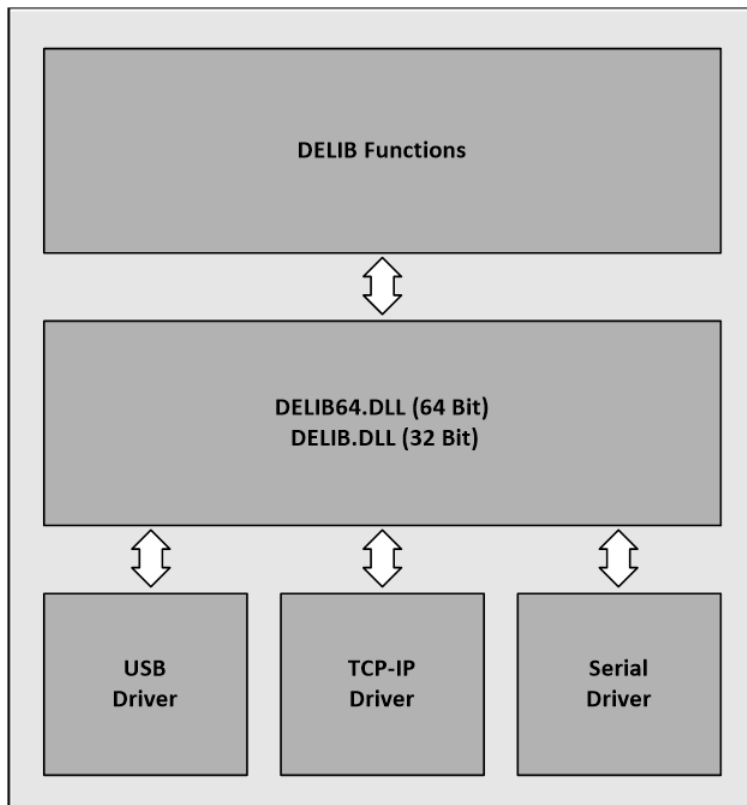
Die DELIB Treiberbibliothek enthält die DELIB-API und verschiedene Programme für den Konfigurationstest unserer Produkte.

Über die API haben Sie Zugriff auf alle Funktionen, die Sie zur Kommunikation mit unseren Produkten benötigen.

In dem Kapitel **DELIB API Referenz** finden Sie alle Funktionen unserer Treiberbibliothek erklärt und mit Anwendungsbeispielen versehen.

### 3.2.1. Übersicht

Die folgende Abbildung erläutert den Aufbau der DELIB Treiberbibliothek



Die DELIB Treiberbibliothek ermöglicht ein einheitliches Ansprechen von DEDITEC Hardware, mit der besonderen Berücksichtigung folgender Gesichtspunkte:

- Betriebssystem unabhängig
- Programmiersprachen unabhängig
- Produkt unabhängig

**Diese Versionen der Treiberbibliothek bieten wir an:**

- 32/64-Bit DELIB Treiberbibliothek für Windows
- 32/64-Bit DELIB Treiberbibliothek für Linux
- 32/64-Bit DELIB Treiberbibliothek ETH

### **3.2.1.1. Unterstützte Programmiersprachen**

Die folgenden Programmiersprachen werden von der DELIB Treiberbibliothek unterstützt:

- C
- C++
- C#
- Delphi
- VisualBasic
- VB.NET
- MS-Office (VBA)
- Java (Plattformunabhängig, nur für Ethernet-Produkte)
- Java JNI (nur für Windows, alle Produkte werden unterstützt)

Falls von der Programmiersprache/Entwicklungsumgebung vorgesehen, unterstützen wir sowohl 32-Bit als auch 64-Bit Projekte.



### **3.2.1.2. Unterstützte Betriebssysteme**

Die folgende Betriebssysteme sind mit unserer DELIB Treiberbibliothek kompatibel:

#### **32-Bit:**

- Windows 10
- Windows 7
- Windows 8
- Windows Server 2012
- Windows Server 2008
- Windows Vista
- Windows XP
- Windows Server 2003
- Windows 2000
- Linux

#### **64-Bit:**

- Windows 10 x64
- Windows 7 x64
- Windows 8 x64
- Windows Server 2012 x64
- Windows Server 2008 x64
- Windows Vista x64
- Windows XP x64
- Windows Server 2003 x64
- Linux x64

### **3.2.1.3. SDK-Kit für Programmierer**

Integrieren Sie die DELIB in Ihre Anwendung. Auf Anfrage erhalten Sie von uns kostenlos Installationsskripte, die es ermöglichen, die DELIB Installation in Ihre Anwendung mit einzubinden.

### 3.2.2. DELIB Setup

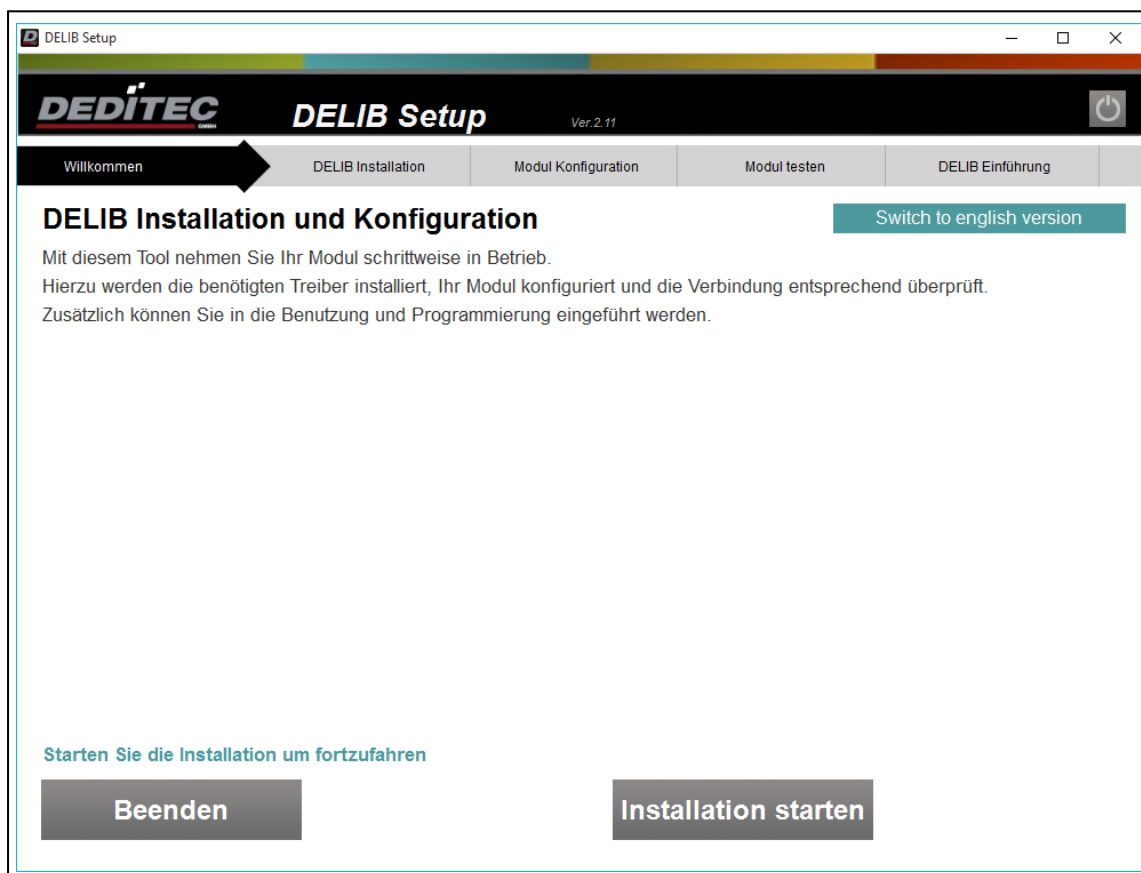
Das DELIB Setup führt Sie durch die Installation unserer DELIB Treiberbibliothek.

Anschließend werden Sie durch den Konfigurationsvorgang sowie Funktionstest für unsere verschiedenen Produkte geführt.

Die aktuelle Version des DELIB Setups finden Sie auf unserer Homepage zum Download.

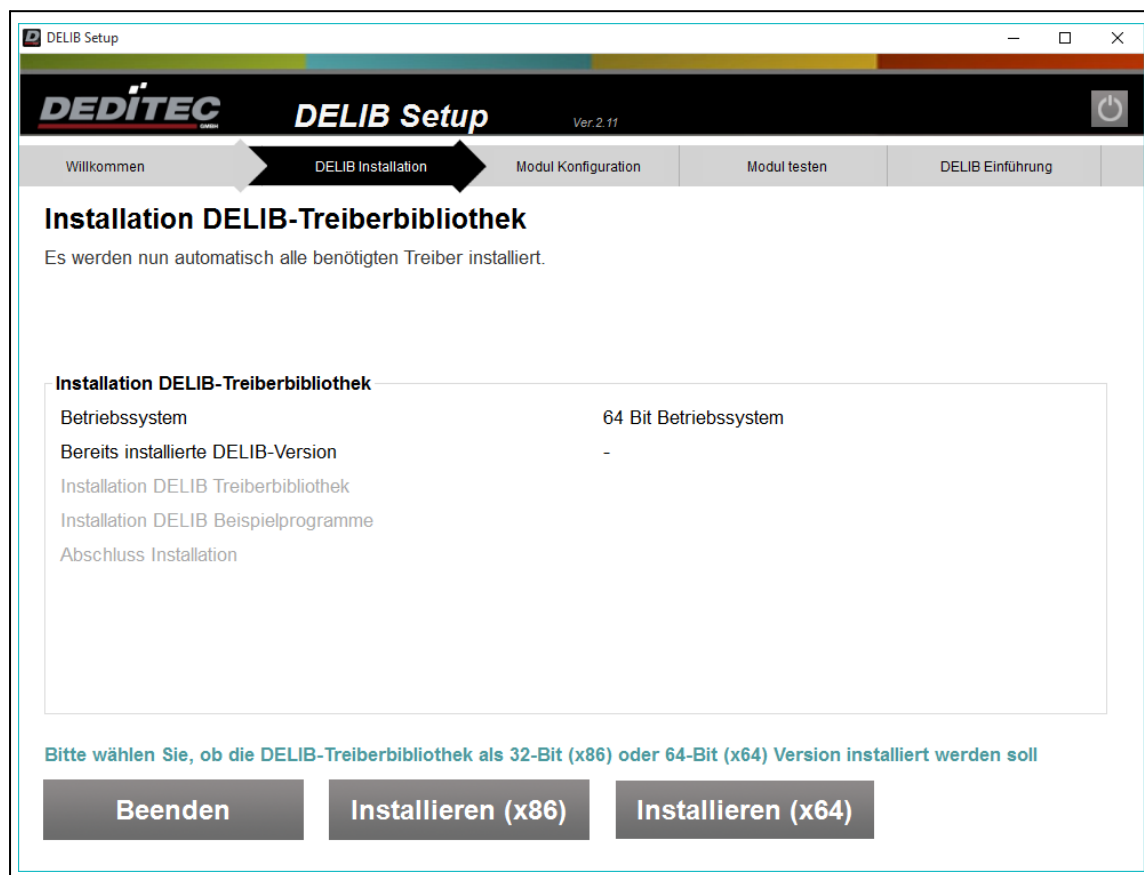
Link: <https://www.deditec.de/delib>

Das DELIB Setup führt Sie Schritt für Schritt durch die Installation der DELIB Treiberbibliothek einschließlich der Konfiguration und Inbetriebnahme der Produkte.

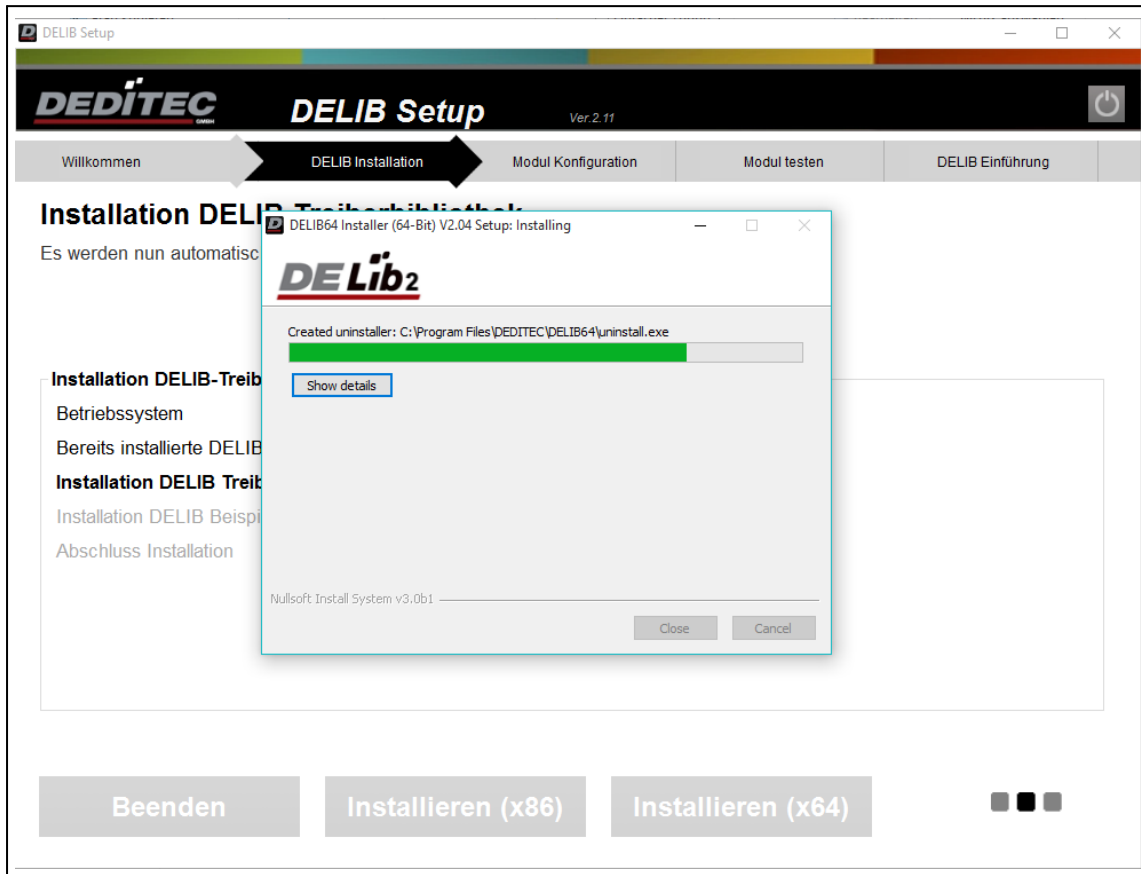


Das DELIB Setup prüft das Betriebssystem und ob bereits Versionen der DELIB Treiberbibliothek installiert sind.

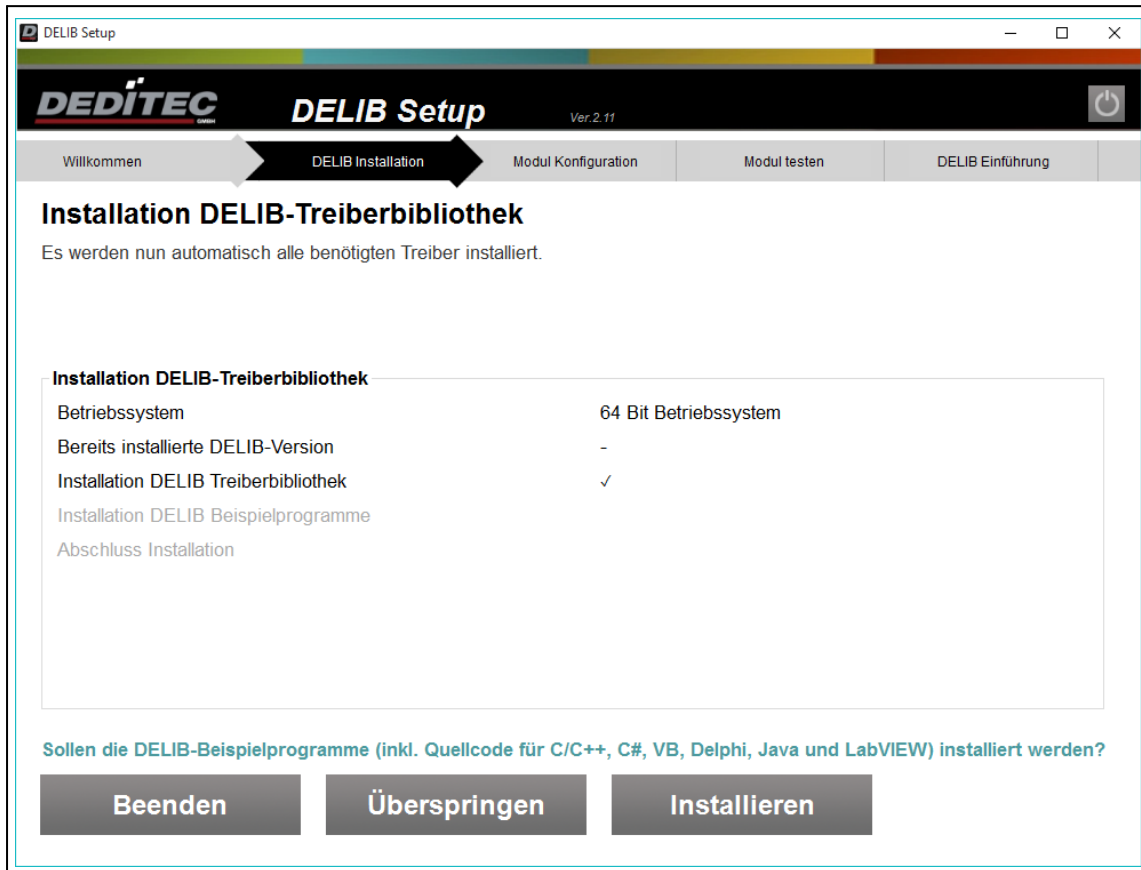
Anschließend können Sie wählen ob Sie die 32-Bit oder 64-Bit Version der DELIB Treiberbibliothek installieren möchten.



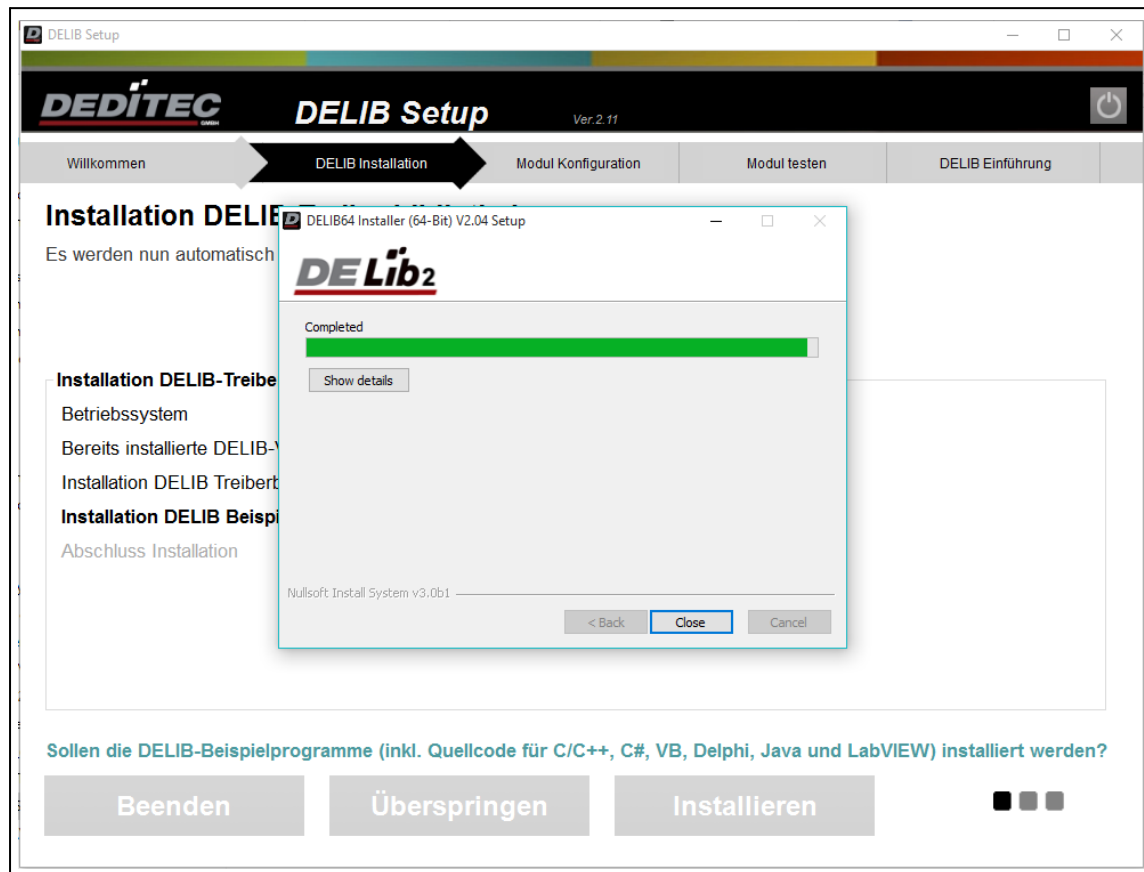
Installationsfortschritt der Treiberbibliothek.



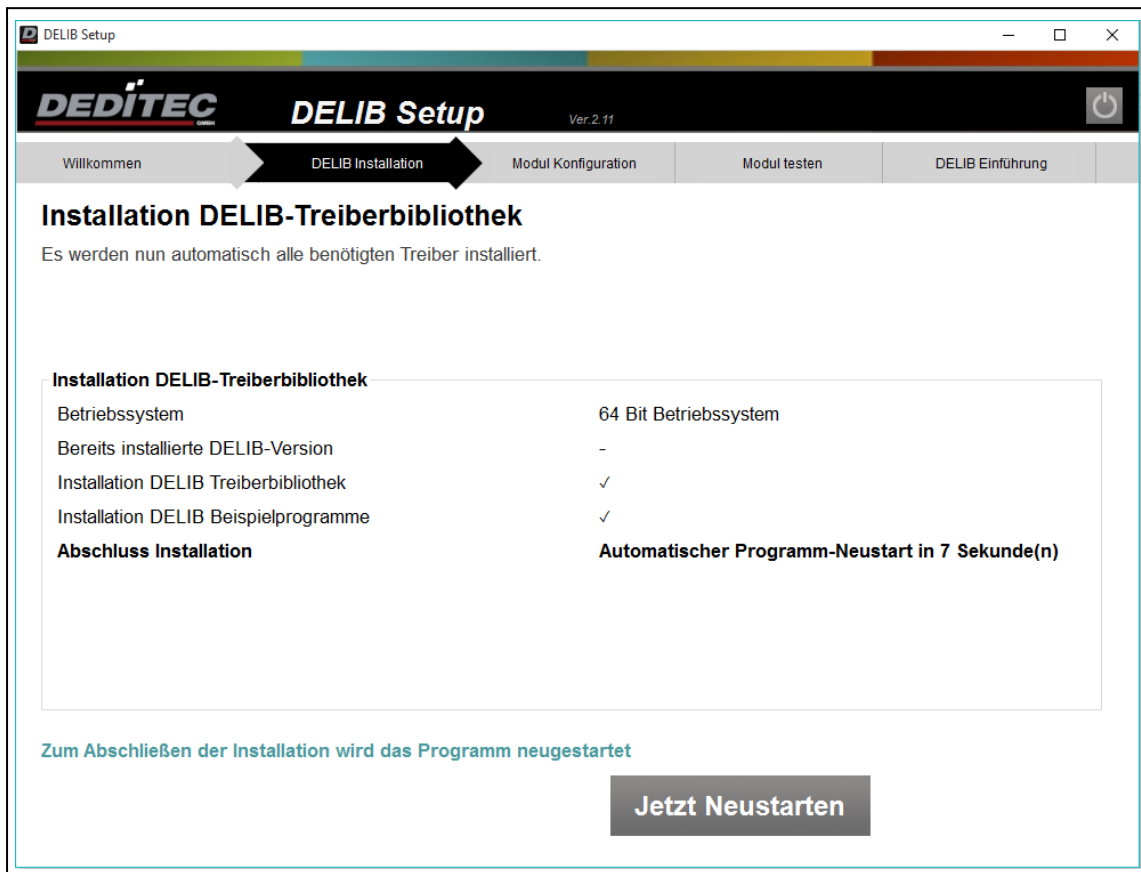
Zusätzlich können Sie wählen, ob Sie die DELIB Beispielprogramme installieren.



## Installationsfortschritt der DELIB Beispielprogramme.



Für den Abschluss der Installation wird das Programm neu gestartet. Im nächsten Schritt, wird mit dem DELIB Configuration Utility das Produkt konfiguriert und getestet.



### **3.2.3. DELIB Configuration Utility**

#### **3.2.3.1. Einführung**

Das DELIB Configuration Utility ermöglicht die Konfiguration der Ethernet-, CAN- oder seriellen Schnittstelle eines Produktes.

Die Konfiguration ist für die erste Inbetriebnahme erforderlich.

Ausgenommen sind Produkte mit USB-Schnittstelle. Diese müssen nur konfiguriert werden, falls Sie mehrere identische USB Produkte an einem PC betreiben möchten.

Das DELIB Configuration Utility ist in der Installation der DELIB Treiberbibliothek enthalten.

#### **Standardpfad:**

32-Bit: C:\Program Files (x86)\DEDITEC\DELIB\programs\delib-configuration-utility.exe

64-Bit: C:\Program Files\DEDITEC\DELIB64\programs\delib-configuration-utility\_x64.exe

Sie können das DELIB Configuration Utility auch über das Startmenü unter "Alle Programme" → "DEDITEC" → "DELIB Configuration Utility" öffnen.



### 3.2.3.2. Neue Konfiguration erstellen oder vorhandene Konfiguration bearbeiten

Für eine neue Konfiguration wählen Sie in der linken Auswahlbox unter "Neues Modul" die gewünschte Schnittstelle aus.

Möchten Sie eine bestehende Konfiguration bearbeiten, finden Sie rechts die Auswahlbox der vorhandenen Konfigurationen.

The screenshot shows the 'DELIB Configuration Utility' window. The title bar reads 'DELIB Configuration Utility'. The header area features the 'DEDITEC' logo, the product name 'DELIB Configuration Utility', the version 'Ver.2.11', and 'DELIB 2.101 (64 Bit)' with a power icon. A navigation bar contains five tabs: 'Modul Auswahl' (active), 'Modul Konfiguration', 'Konfiguration testen', 'Modul testen', and 'DELIB Einführung'. The main content area is titled 'Modul Konfiguration' and includes a 'Switch to english version' button. Below the title, there are two columns: 'Neues Modul' and 'Konfigurierte Module'. The 'Neues Modul' column has a 'Schnittstelle' label and a list of radio buttons for 'USB', 'ETH', 'CAN', and 'SER'. The 'Konfigurierte Module' column has a 'Module' label and a list showing 'RO-ETH (1)' and 'RO-SER (0)'. At the bottom, there is a prompt 'Wählen Sie die Schnittstelle oder ein bereits konfiguriertes Modul aus' and two buttons: 'Beenden' and 'Weiter'.

DELIB Configuration Utility

**DEDITEC** **DELIB Configuration Utility** Ver.2.11 **DELIB 2.101** (64 Bit)

Modul Auswahl | Modul Konfiguration | Konfiguration testen | Modul testen | DELIB Einführung

### Modul Konfiguration

Switch to english version

Mit dem DELIB Configuration Utility lassen sich DEDITEC Module in wenigen Schritten konfigurieren.  
Um ein neues Modul zu konfigurieren, wählen Sie bitte die Schnittstelle (links) aus.  
Um eine bereits erstellte Konfiguration für ein Modul zu editieren, wählen Sie das entsprechende Modul (rechts) aus.

Neues Modul	Konfigurierte Module
<p>Schnittstelle</p> <ul style="list-style-type: none"><li><input type="radio"/> USB</li><li><input type="radio"/> ETH</li><li><input type="radio"/> CAN</li><li><input type="radio"/> SER</li></ul>	<p>Module</p> <ul style="list-style-type: none"><li>RO-ETH (1)</li><li>RO-SER (0)</li></ul>

Wählen Sie die Schnittstelle oder ein bereits konfiguriertes Modul aus

**Beenden** **Weiter**

### 3.2.3.2.1. Modul Konfiguration USB

Die Konfiguration von USB-Modulen ist nur nötig, um mehrere Module einer USB-Produktfamilie (z.B. 2x USB-RELAIS-8) in einem System verwenden zu können.

Befindet sich nur ein USB-Modul, oder mehrere USB-Module aus unterschiedlichen Produktfamilien (z.B. RO-USB-016 und USB-RELAIS-8) im System, ist keine Konfiguration nötig, da die Produkte über die Modul-ID eindeutig identifizierbar sind.

Nach einem Klick auf das Produkt, welches Sie konfigurieren möchten, werden Ihnen rechts alle möglichen Einstellungen angezeigt.

The screenshot shows the 'DELIB Configuration Utility' window. The title bar includes the DEDITEC logo, the application name, version 'Ver. 2.19', and 'DELIB 2.19 (64 Bit)'. The main menu has five tabs: 'Modul Auswahl', 'Modul Konfiguration' (which is active), 'Konfiguration testen', 'Modul testen', and 'DELIB Einführung'. The active tab displays the title 'Konfiguration Module mit USB Schnittstelle' and instructions: 'Hier kann die Modul-Nr. des ausgewählten Moduls geändert werden. Die Modul-Nr. dient zur Unterscheidung mehrerer identischer USB-Module einer USB-Serie.' There is a 'Manuel' icon in the top right. Below the instructions, a table titled 'Gefundene USB-Module' shows one entry: 'Module: USB-OPT/REL-8, Nr.: 0'. To the right of the table are input fields: 'Aktuelle Modul-Nr.' with value '0', 'Neue Modul-Nr.' with a dropdown menu showing '1', a 'Neue Modul-Nr. setzen' button, and 'Übertragungsversuche (für alle USB-Module)' with a dropdown menu showing '5'. At the bottom, there are three large buttons: 'Hauptmenü', 'Test', and 'Fertig'.

Module	Nr
USB-OPT/REL-8	0

Aktuelle Modul-Nr. 0

Neue Modul-Nr. 1

Neue Modul-Nr. setzen

Übertragungsversuche (für alle USB-Module) 5

Hauptmenü Test Fertig

Die "Aktuelle Modul-Nr" bezieht sich auf die im Modul gespeicherte Modul Nummer. Diese Nummer dient zur Identifikation und muss für identische USB-Produkte unterschiedlich konfiguriert werden.

Mit dem Punkt "Neue Module-Nr" kann dem Produkt eine neue Nummer zwischen 0 und 255 zugewiesen werden. Im Auslieferungszustand haben alle Produkte die Modul-Nummer 0.

Mit "Neue Module-Nr. setzen" wird die aktuell ausgewählte Neue Modul-Nummer auf das Modul geschrieben

Über die Auswahlbox "Übertragungsversuche (für alle USB-Module)" können Sie festlegen, wie oft die DELIB im Falle eines Fehlers versucht mit dem Modul zu kommunizieren.

#### 3.2.3.2.1.1. Beispiel zur Konfiguration identischer USB-Module

Um mehrere identische USB-Module (USB-Module mit gleicher Modul-ID) in einem System verwenden zu können, muss jedem Modul mit dem DELIB Configuration Utility eine eindeutige Modul-Nr. zugeordnet werden.

Befindet sich nur ein USB-Modul, oder mehrere USB-Module mit unterschiedlicher Modul-ID (z.B. RO-USB-O16 und USB-RELAIS-8) im System, ist keine Konfiguration nötig, da die Produkte über die ID eindeutig identifizierbar sind.

Folgendes Beispiel zeigt die Konfiguration von zwei USB-OPTOIN-8 Modulen im gleichen System.

#### Schritt 1

Verbinden Sie zunächst nur ein USB-OPTOIN-8 mit dem PC und starten Sie das DELIB Configuration Utility.

Wählen Sie im linken Bereich die USB-Schnittstelle aus und klicken auf "Weiter".

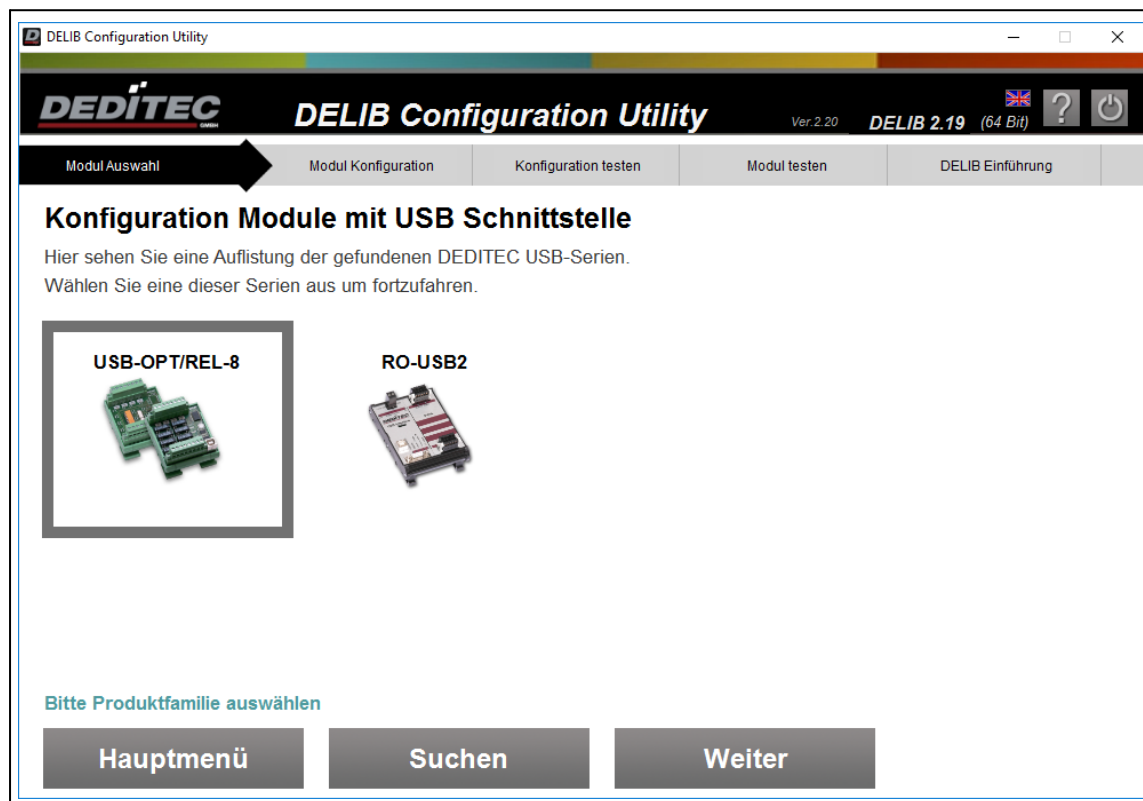
The screenshot shows the 'DELIB Configuration Utility' window. The title bar reads 'DELIB Configuration Utility'. The main header features the 'DEDITEC' logo, the product name 'DELIB Configuration Utility', the version 'Ver. 2.20', and 'DELIB 2.19 (64 Bit)'. Below the header is a navigation bar with five tabs: 'Modul Auswahl' (selected), 'Modul Konfiguration', 'Konfiguration testen', 'Modul testen', and 'DELIB Einführung'. The 'Modul Konfiguration' tab is active, displaying the title 'Modul Konfiguration' and a subtitle 'Mit dem DELIB Configuration Utility lassen sich DEDITEC Module in wenigen Schritten konfigurieren.' The interface is divided into two main sections. The left section, titled 'Neues Modul', contains a 'Schnittstelle' (Interface) dropdown menu with four radio button options: 'USB' (selected), 'ETH (Ethernet)', 'CAN', and 'SER (Seriell)'. The right section, titled 'Konfigurierte Module', contains a 'Module' list box showing a list of modules: 'USB-OPT/REL-8 (0)', 'RO-CAN2 (0)', 'RO-ETH (0)', 'RO-ETH (1)', 'RO-ETH/LC (0)', 'RO-ETH/LC (3)', 'RO-SER (0)', and 'BS-ETH (0)'. At the bottom, there is a blue instruction line: 'Wählen Sie die Schnittstelle oder ein bereits konfiguriertes Modul aus'. Below this are two large buttons: 'Beenden' (End) on the left and 'Weiter' (Next) on the right.

## Schritt 2

Sind mehrere USB-Module verschiedener DEDITEC-USB-Serien angeschlossen, muss in diesem Schritt die entsprechende Produktfamilie ausgewählt werden.

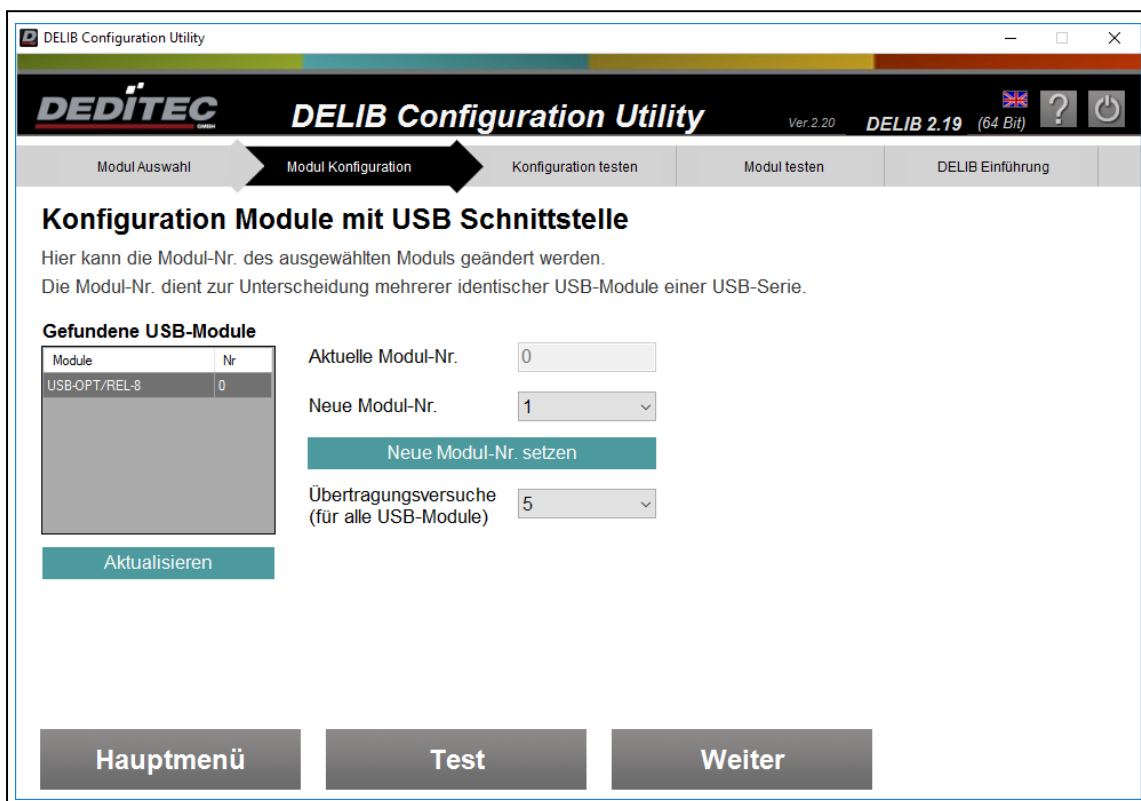
In diesem Beispiel sind Module der RO-USB2-Serie und USB-OPT/REL-8-Serie angeschlossen.

Dieser Schritt entfällt, wenn die angeschlossenen Module der gleichen Serie angehören.



### Schritt 3

1. Wählen Sie das entsprechende USB-Modul aus.
2. Ändern Sie die Neue Modul-Nr. auf "1". Im Auslieferungszustand ist diese Nummer bereits mit "0" vordefiniert.
3. Mit Neue Modul-Nr. setzen wird die neue Module-Nr. im Modul gespeichert.

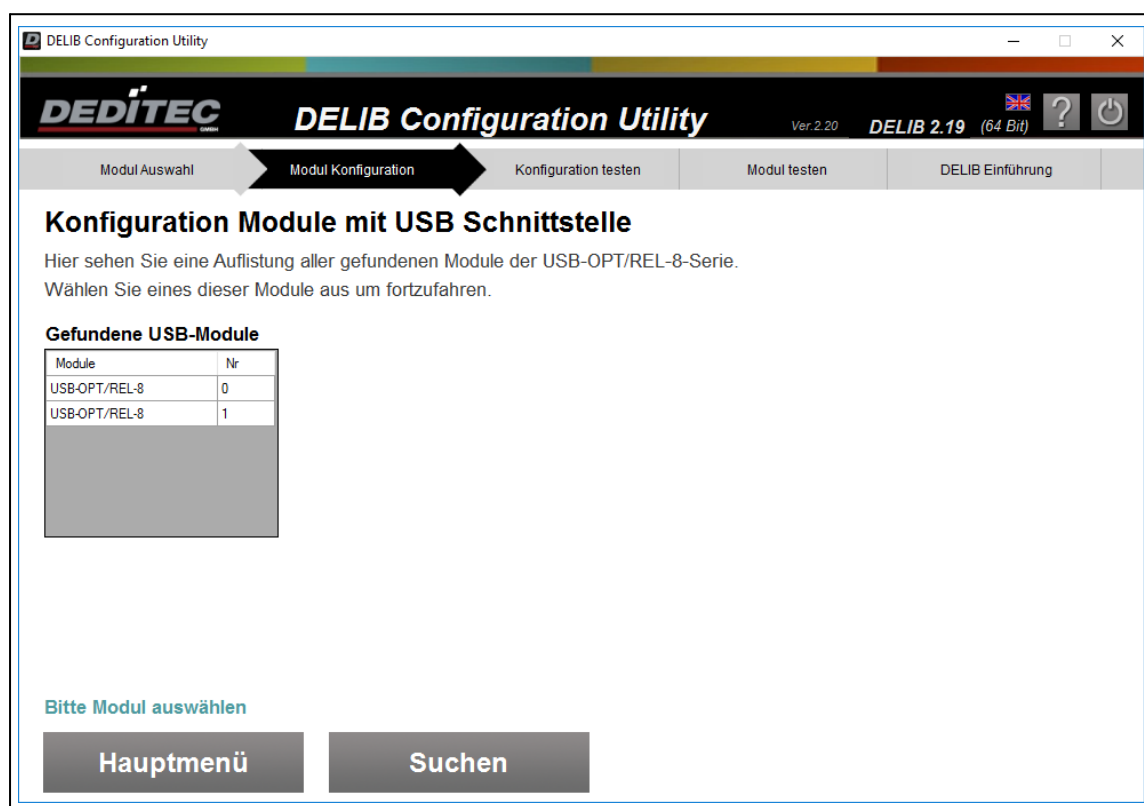


#### Schritt 4

Schließen Sie nun zusätzlich das zweite USB-OPTOIN-8 an den PC an.

Da im Auslieferungszustand die Modul-Nr bereits mit "0" vordefiniert und somit unterschiedlich zur Modul-Nr des ersten Moduls (Module-Nr = 1) ist, sind beide Module konfiguriert und betriebsbereit.

Mit Aktualisieren werden nun beide Module angezeigt.



## Schritt 5

Nachfolgend finden Sie Hinweise, was bei der Programmierung der beiden Module beachten werden muss.

Alle Module werden einheitlich mit dem Befehl `DapiOpenModule` geöffnet. Dieser Befehl ist wie folgt definiert:

*`ULONG DapiOpenModule(ULONG moduleID, ULONG nr);`*

### Ansprechen des USB-OPTOIN-8 mit der NR 0

```
ulong handle;  
handle = DapiOpenModule(USB_OPTOIN_8, 0);  
//öffnet das Modul USB-OPTOIN-8 mit der NR 0
```

### Ansprechen des USB-OPTOIN-8 mit der NR 1

```
ulong handle;  
handle = DapiOpenModule(USB_OPTOIN_8, 1);  
//öffnet das Modul USB-OPTOIN-8 mit der NR 1
```

### Hinweis:

Alle Module haben als Werkseinstellung die NR "0".

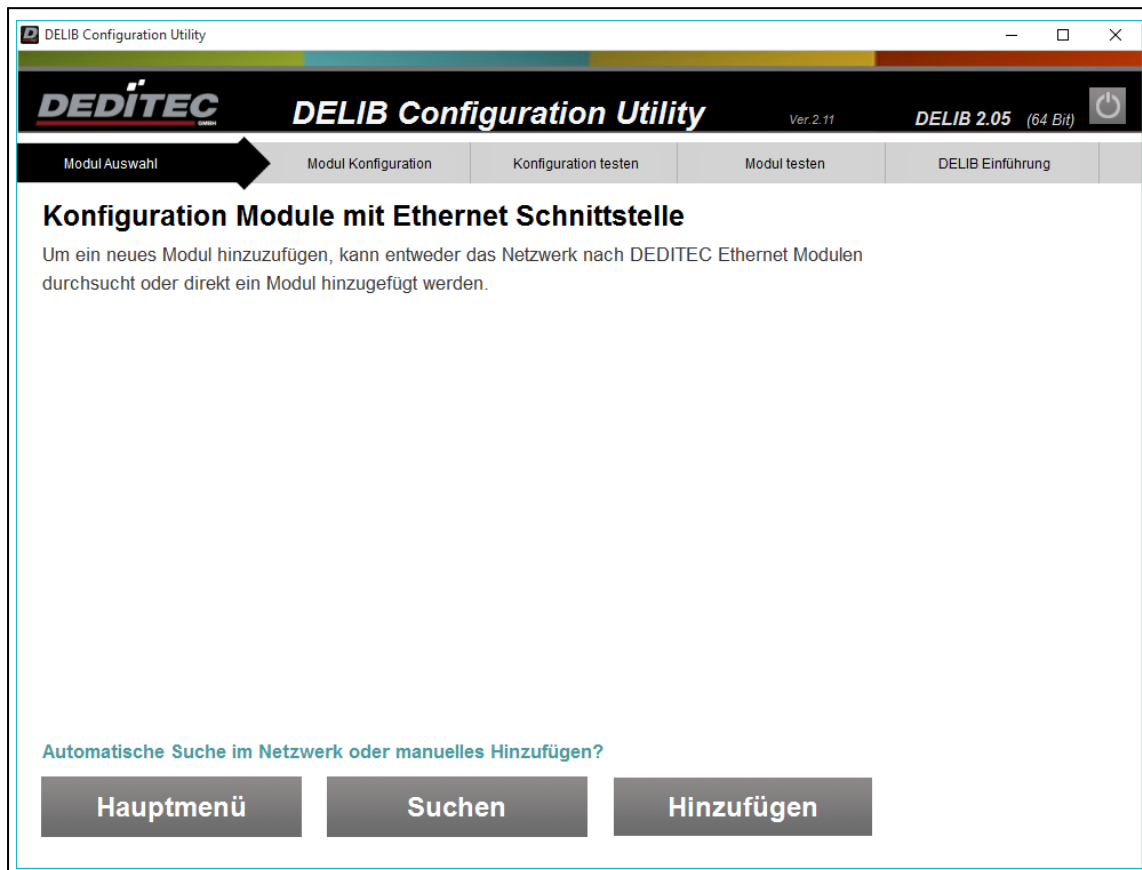
Wenn Sie mehrere identische USB-Module verwenden möchten, muss den Modulen nacheinander, eine eindeutige NR zugeordnet werden.



### 3.2.3.2.2. Modul Konfiguration Ethernet

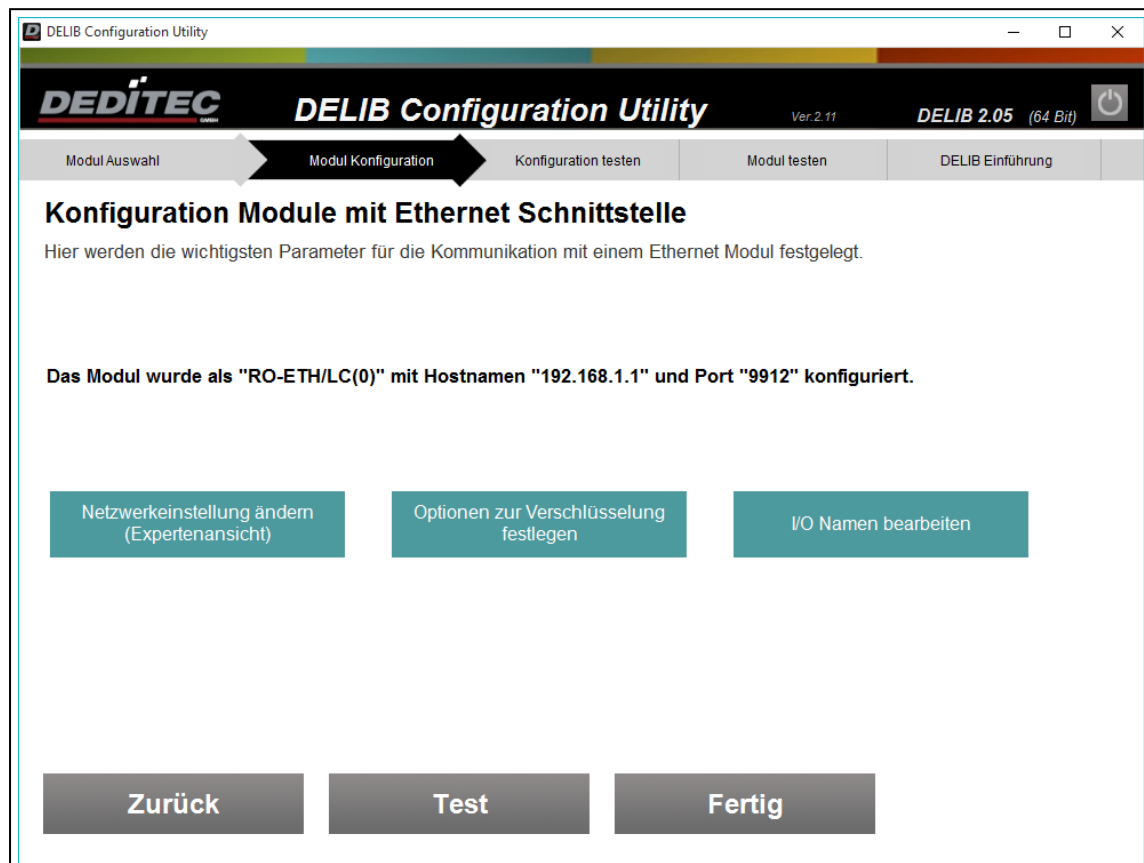
Bei der Konfiguration von Ethernet Produkten kann das Netzwerk automatisch nach DEDITEC Produkten durchsucht werden.

Auch eine manuelle Konfiguration kann durchgeführt werden.



Im letzten Schritt der Konfiguration können Sie weitere Optionen für die Kommunikationsverschlüsselung oder der I/O Namensvergabe durchführen.

Wenn Sie sich für eine manuelle Konfiguration entschließen, beginnen Sie an dieser Stelle.



Bei der Verschlüsselung kann zwischen Deaktiviert, User oder Admin ausgewählt werden.

### Deaktiviert

- unverschlüsselte Kommunikation
- kein Zugriff auf System-Einstellungen

### **User**

- verschlüsselte Kommunikation
- Lese-Zugriff auf System-Einstellungen

### **Admin**

- verschlüsselte Kommunikation
- Schreib-Lese-Zugriff auf System-Einstellungen

In das Feld Passwort muss das gewünschte Passwort für die Verschlüsselung eingetragen werden.

Durch einen Klick auf den Button "Einstellungen zur Verschlüsselung auf das Modul übertragen" werden Sie aufgefordert eine Hardware-Taste am Produkt zu betätigen. Erst nach dem Betätigen dieser Taste werden die Verschlüsselungsoptionen vom Produkt übernommen.

siehe Kapitel → **Authentifizierung**

The screenshot shows the 'DELIB Configuration Utility' window. The title bar includes the DEDITEC logo, the application name, version 'Ver. 2.11', and 'DELIB 2.05 (64 Bit)'. The main menu has five tabs: 'Modul Auswahl', 'Modul Konfiguration' (which is active), 'Konfiguration testen', 'Modul testen', and 'DELIB Einführung'. The active tab displays the title 'Konfiguration Module mit Ethernet Schnittstelle' and a subtitle 'Hier werden die wichtigsten Parameter für die Kommunikation mit einem Ethernet Modul festgelegt.' Below this, a status message reads: 'Das Modul wurde als "RO-ETH/LC(0)" mit Hostnamen "192.168.1.1" und Port "9912" konfiguriert.' The configuration area contains several elements: a button 'Netzwerkeinstellung ändern (Expertenansicht)' on the left; a 'Verschlüsselung' dropdown menu set to 'deaktiviert' with a 'Passwort' text input field below it; a button 'I/O Namen bearbeiten' on the right; and a button 'Einstellungen zur Verschlüsselung auf das Modul übertragen' centered below the password field. At the bottom of the window are three large buttons: 'Zurück', 'Test', and 'Fertig'.

Unsere Ethernet-Produkte, bieten Ihnen die Möglichkeit, für die digitalen und analogen I/Os, Namen zu vergeben. Diese werden beispielsweise auf der Weboberfläche oder in unserer App verwendet.

DELIB Configuration Utility

**DELIB Configuration Utility** Ver. 2.11 DELIB 2.101 (64 Bit)

Modul Auswahl → **Modul Konfiguration** → Konfiguration testen → Modul testen → DELIB Einführung

### Konfiguration Module mit Ethernet Schnittstelle

Hier werden die wichtigsten Parameter für die Kommunikation mit einem Ethernet Modul festgelegt.

**I/O Namen Konfiguration**

Auswahl I/O-Typ: Digital Output Kanal-Bereich: Channel (1..8)

Kanal	Name	Kanal	Name
CH 01	Example name	CH 05	Digital Output 5
CH 02	Digital Output 2	CH 06	Digital Output 6
CH 03	Digital Output 3	CH 07	Digital Output 7
CH 04	Digital Output 4	CH 08	Digital Output 8

Zurück Speichern

### Hinweis

Zum Speichern der Kanalnamen wird eine Admin-Kommunikation benötigt.  
Die maximale Zeichenlänge beträgt 16.

#### **3.2.3.2.2.1. Automatische Suche**

Verhalten des DELIB Configuration Utility bei der Verwendung mehrerer Netzwerkadapter in einem Windowssystem.

Bei der Suche nach DEDITEC Ethernet Modulen über den primären Netzwerkadapter (ETH0) des PCs werden die Netzwerkeinstellungen des DEDITEC Produktes ignoriert. Dies bedeutet, dass unsere Ethernet Produkte selbst dann gefunden werden, wenn diese bedingt durch die IP-Adresse und Subnetzmaske nicht im selben Netzwerk sind.

Ist das Ethernet Modul hingegen an einem anderen, bzw. nicht primären Netzwerkadapter (z.B. ETH1) angeschlossen, muss die Netzwerkkonfiguration des Ethernet-Moduls gültig sein, so dass das Modul gefunden wird.

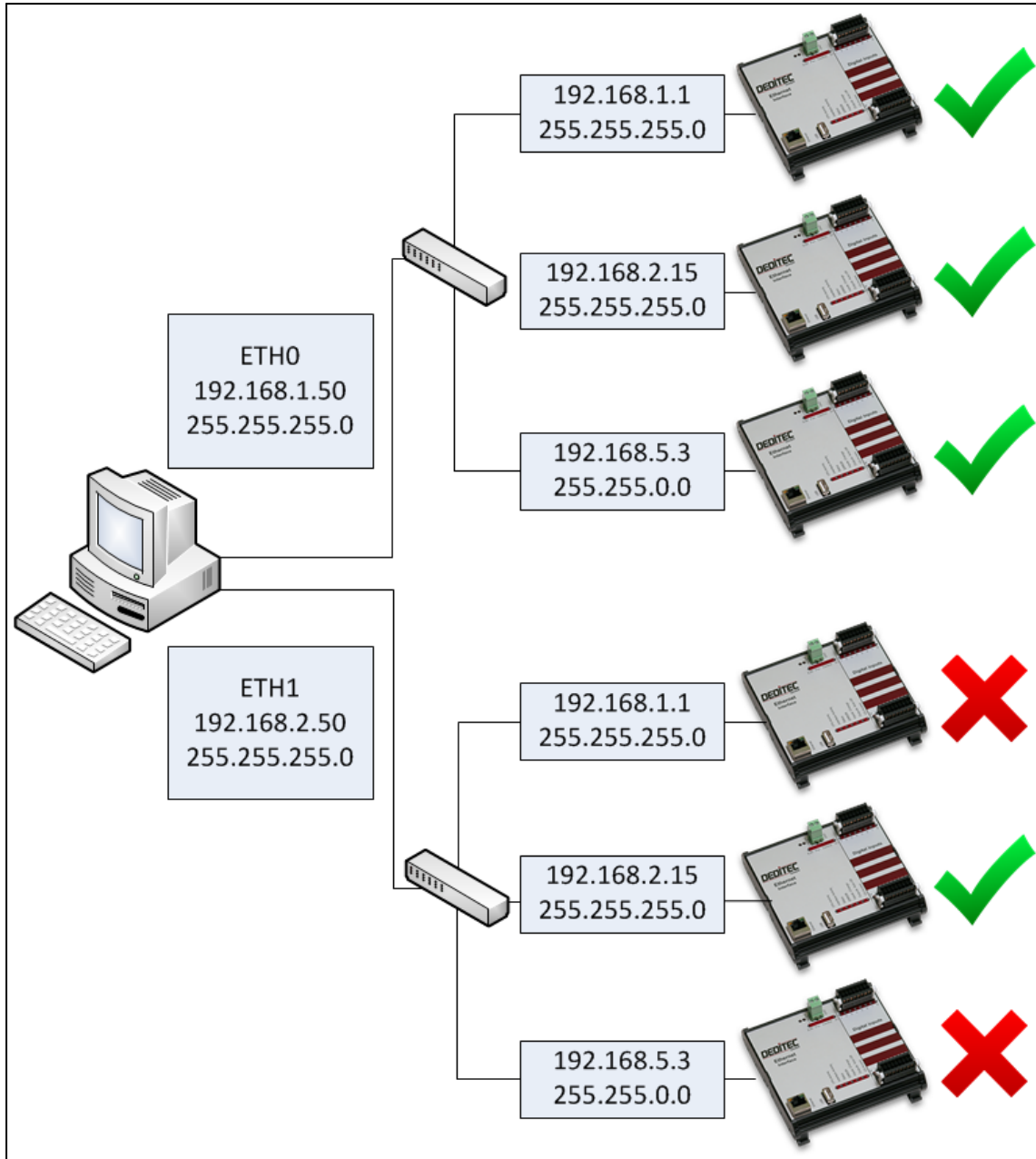
#### **Achtung:**

Wird ein Laptop verwendet, ist meistens der integrierte WLAN-Adapter als primärer Netzwerkadapter konfiguriert.

Hierdurch werden Module, die per LAN-Kabel mit dem Laptop verbunden sind, oft nicht erkannt, da die Netzwerkkonfiguration nicht gültig ist.

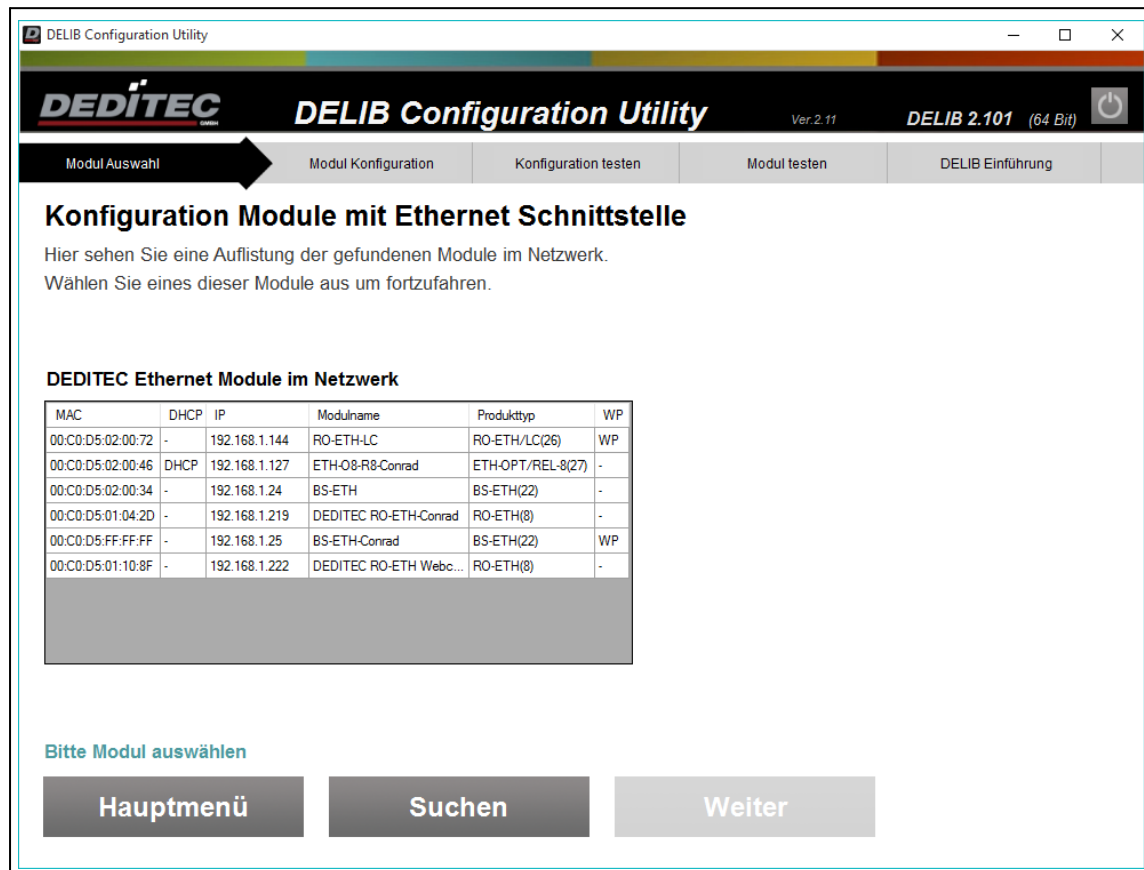
Zur Modul-Einrichtung empfiehlt es sich daher, den WLAN-Adapter kurzzeitig zu deaktivieren.

Folgende Grafik zeigt eine Übersicht der automatischen Suche.



Nach der automatischen Suche werden alle gefundenen DEDITEC Ethernet Produkte in einer Übersicht dargestellt.

Die Übersicht zeigt die MAC-Adresse, den DHCP Status, die IP, den Modulnamen, den Produkttypen sowie den Status der Write-Protection.





Mit einem Klick auf ein Modul kann die Netzwerkkonfiguration für dieses Modul geändert werden. Bitte beachten Sie, dass die Write-Protection ausgeschaltet sein muss, wenn Sie die Einstellungen speichern möchten. Die Write-Protection kann über die Authentifizierung temporär deaktiviert werden.

siehe Kapitel → **Authentifizierung**

**Konfiguration Module mit Ethernet Schnittstelle**

An dieser Stelle kann die aktuelle Netzwerk Konfiguration des Moduls (rechts) geändert werden.  
Mit "Übernehmen" wird diese Konfiguration an das Modul übertragen.  
Klicken Sie auf "Weiter" um die Kommunikation mit dem Modul zu testen.

**DEDITEC Ethernet Module im Netzwerk**

MAC	DHCP	IP	Modulname	Produkttyp	WP
00:C0:D5:02:00:72	-	192.168.1.1	RO-ETH-LC	RO-ETH/LC(26)	-
00:C0:D5:02:00:46	-	192.168.1.171	ETH-O8-R8-Conrad	ETH-OPT/REL-8(27)	-
00:C0:D5:FF:FF:FF	-	192.168.1.25	BS-ETH-Conrad	BS-ETH(22)	-
00:C0:D5:02:00:34	-	192.168.1.24	BS-ETH	BS-ETH(22)	-
00:C0:D5:01:10:8F	-	192.168.1.222	DEDITEC RO-ETH Webc...	RO-ETH(8)	-
00:C0:D5:01:04:2D	-	192.168.1.219	DEDITEC RO-ETH-Conrad	RO-ETH(8)	-

**Aktuelle Modul Konfiguration**

Board Name:

☐ IP-Adresse automatisch beziehen (DHCP)

IP Adresse:

Netzmaske:

Std.-Gateway:

## Hinweis

Bitte beachten Sie, dass in diesem Schritt nur die Modul-Konfiguration geändert wird.

Wurde das Modul bereits im DELIB-Configuration Utility konfiguriert, muss diese Konfiguration an die neue Modul-Konfiguration angepasst werden.

#### **3.2.3.2.2.2. Verschlüsselung einrichten**

Um wichtige Modul-Einstellungen (z.B. Netzwerkkonfiguration) über TCP bearbeiten zu können, muss die Kommunikation zwischen Modul und PC im sogenannten verschlüsselten Admin-Modus laufen.

Hierzu muss im Modul und auf der PC-Seite ein Passwort zur Verschlüsselung konfiguriert werden.

Diese Konfiguration kann wahlweise über den automatischen Assisten oder manuell erstellt werden.

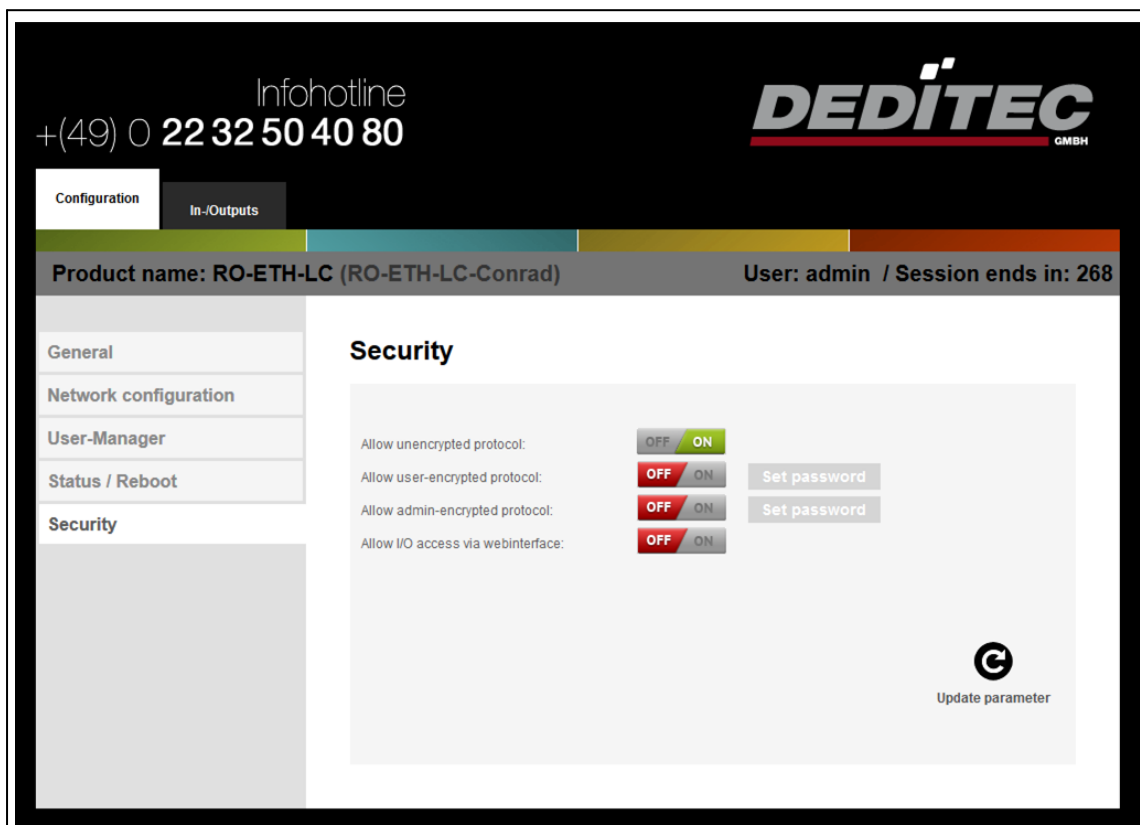
#### **Manuelle Konfiguration**

Bei der manuellen Konfiguration, muss jeweils auf der Modul- und PC-Seite ein Passwort zur Verschlüsselung festgelegt werden.

## Schritt 1 - Konfiguration Modul-Seite

Zuerst wird das Modul via Weboberfläche konfiguriert. Geben Sie die IP-Adresse des Moduls (Auslieferungszustand 192.168.1.1) in einem Internet-Browser ein. Sollte eine Authentifizierung notwendig sein, loggen Sie sich ein (Auslieferungszustand: user=admin + passwort=admin).

Öffnen Sie nun die Sicherheitseinstellungen (**Configuration → Security**)



Falls deaktiviert, aktivieren Sie die Einstellung Allow admin-encrypted protocol. Anschließend können Sie mit Set password ein neues Passwort zur Verschlüsselung einstellen.

Mit Update parameter wird, sofern ein neues Passwort korrekt eingegeben wurde, die Konfiguration abgeschlossen.

## Hinweis

Bei der Eingabe des Passwortes sind alle Zeichen erlaubt.

The screenshot displays the DEDITEC web interface. At the top left, there is an 'Infohotline' number: + (49) 0 22 32 50 40 80. The DEDITEC GMBH logo is in the top right. Below the header, a navigation bar shows 'Configuration' and 'In-/Outputs'. A status bar indicates 'Product name: RO-ETH-LC (RO-ETH-LC-Conrad)' and 'User: admin / Session ends in: 223'. On the left, a sidebar menu lists 'General', 'Network configuration', 'User-Manager', 'Status / Reboot', and 'Security'. The 'Security' section is active, showing a 'Security' title and several toggle switches: 'Allow unencrypted protocol' (OFF/ON), 'Allow user-encrypted protocol' (OFF/ON), 'Allow admin-encrypted protocol' (OFF/ON), and 'Allow I/O access via webinterface' (OFF/ON). There are two password input fields labeled 'New password:' and 'Confirm password:', each with a masked password (dots). A 'Set password' button is located to the right of the password fields. At the bottom right, there is a circular arrow icon and the text 'Update parameter'.

## Schritt 2 - Konfiguration PC-Seite

Starten Sie das DELIB Configuration Utility und wählen Sie das gewünschte Ethernet-Modul aus.

Hier bei Verschlüsselung "Admin" auswählen.

The screenshot shows the 'DELIB Configuration Utility' window. The title bar reads 'DELIB Configuration Utility'. The main header features the 'DEDITEC' logo, the product name 'DELIB Configuration Utility', the version 'Ver. 2.19', and 'DELIB 2.19 (64 Bit)' with a power icon. A navigation bar contains five tabs: 'Modul Auswahl', 'Modul Konfiguration' (which is active and highlighted with a black arrow), 'Konfiguration testen', 'Modul testen', and 'DELIB Einführung'. The main content area is titled 'Konfiguration Module mit Ethernet Schnittstelle' and includes a sub-header 'Hier werden die wichtigsten Parameter für die Kommunikation mit einem Ethernet Modul festgelegt.' and a 'Manual' icon. Under the 'Netzwerkkonfiguration' section, there are several input fields: 'Modul Auswahl' (a dropdown menu showing 'RO-ETH/LC(0)'), 'Aktiviert' (checked checkbox), 'Hostname verwenden' (unchecked checkbox), 'IP Adresse' (text field with '192.168.1.25'), 'Port' (text field with '9912'), 'Timeout [msec]' (text field with '5000'), and 'Verschlüsselung' (a dropdown menu showing 'deaktiviert'). A teal button labeled 'I/O Namen bearbeiten' is positioned to the right of the encryption dropdown. At the bottom, there are three large buttons: 'Hauptmenü', 'Test', and 'Fertig'.

Im neu erscheinenden Passwort Feld das Passwort aus Schritt 1 eintragen.  
Mit Test kann die neue verschlüsselten Admin-Modus Kommunikation getestet werden.

Das DELIB Configuration Utility kann nun geschlossen werden.

The screenshot shows the 'DELIB Configuration Utility' window. The title bar reads 'DELIB Configuration Utility'. The main header features the 'DEDITEC' logo, the title 'DELIB Configuration Utility', the version 'Ver. 2.19', and 'DELIB 2.19 (64 Bit)' with a power icon. A progress bar at the top is divided into five segments: 'Modul Auswahl', 'Modul Konfiguration', 'Konfiguration testen' (highlighted with a black arrow), 'Modul testen', and 'DELIB Einführung'. Below the header, the section 'Konfiguration Module mit Ethernet Schnittstelle' is displayed, followed by the text 'Hier werden die wichtigsten Parameter für die Kommunikation mit einem Ethernet Modul festgelegt.' and a 'Manual' icon. The 'Netzwerkkonfiguration' section contains the following fields: 'Modul Auswahl' (dropdown menu showing 'RO-ETH/LC(0)'), 'Aktiviert' (checked checkbox), 'Hostname verwenden' (unchecked checkbox), 'IP Adresse' (text box with '192.168.1.25'), 'Verschlüsselung' (dropdown menu showing 'Admin'), 'Port' (text box with '9912'), 'Passwort' (password box with '\*\*\*\*\*'), and 'Timeout [msec]' (text box with '5000'). There are two buttons: 'I/O Namen bearbeiten' (teal) and 'Einstellungen zur Verschlüsselung auf das Modul übertragen' (teal). A status message at the bottom reads 'Modul-Kommunikation ist OK! Aktuelle Modul-Firmware ist 2.20'. At the very bottom are three large buttons: 'Hauptmenü', 'Test', and 'Fertig'.

## Automatische Konfiguration

Starten Sie das DELIB Configuration Utility und wählen Sie das gewünschte Ethernet-Modul aus.

DELIB Configuration Utility

**DELIB Configuration Utility** Ver. 2.19 DELIB 2.19 (64 Bit)

Modul Auswahl Modul Konfiguration Konfiguration testen Modul testen DELIB Einführung

### Konfiguration Module mit Ethernet Schnittstelle

Hier werden die wichtigsten Parameter für die Kommunikation mit einem Ethernet Modul festgelegt.

**Netzwerkkonfiguration**

Modul Auswahl: RO-ETH/LC(0)

☒ Aktiviert ☐ Hostname verwenden

IP Adresse: 192.168.1.25 Verschlüsselung: Admin I/O Namen bearbeiten

Port: 9912 Passwort: \*\*\*\*\*

Timeout [msec]: 5000 Einstellungen zur Verschlüsselung auf das Modul übertragen

Hauptmenü Test Fertig

- 1) Bei Verschlüsselung Admin auswählen
- 2) Im neu erscheinenden Passwort Feld ein Passwort eintragen
- 3) Mit Einstellungen zur Verschlüsselung auf das Modul übertragen, wird die aktuelle Verschlüsselungs-Einstellung auf das Modul übertragen

## Hinweis

Bei der Eingabe des Passwortes sind alle Zeichen erlaubt.

Idealerweise sollte sich das Passwort aus einer Kombination von Groß- und Kleinbuchstaben, Zahlen, sowie Sonderzeichen zusammensetzen.

Um das Modul vor unrechtmäßigen Zugriffen zu schützen, muss beim Editieren von System-Einstellungen eine Authentifizierung durchgeführt werden.

Je nach Produkttyp werden Sie aufgefordert, den Firmware-Reset-Taster des Moduls für eine gewisse Zeit zu betätigen (RO-ETH und RO-CPU-800) oder DIP-Schalter-Einstellungen zu verändern (alle anderen Produkte, z.B. RO-ETH/LC, NET-ETH, ETH-RELAIS-8, ...).

Im nachfolgenden Beispiel wird gezeigt, wie die Authentifizierung bei einem RO-ETH/LC Modul durchgeführt wird.

Im ersten Schritt werden Sie aufgefordert, die Stellung von DIP-Schalter 2 zu invertieren.

DELIB Configuration Utility

**DELIB Configuration Utility** Ver. 2.19 DELIB 2.19 (64 Bit)

Modul Auswahl Modul Konfiguration Konfiguration testen Modul testen DELIB Einführung

### Konfiguration Module mit Ethernet Schnittstelle

Hier werden die wichtigsten Parameter für die Kommunikation mit einem Ethernet Modul festgelegt.

**Netzwerkkonfiguration**

Modul Auswahl: RO-ETH/LC(0)

☒ Aktiviert ☐ Hostname verwenden

IP Adresse: 192.168.1.25 Verschlüsselung: Admin I/O Namen bearbeiten

Port: 9912 Passwort: \*\*\*\*\*

Timeout [msec]: 5000 Einstellungen zur Verschlüsselung auf das Modul übertragen

Schritt 1: DIP-Schalter 2 (Schreibschutz) des ETH-Moduls auf Stellung 1 (0=OFF, 1=ON) setzen

Authentifizierung abbrechen

Hauptmenü Test Fertig



Im zweiten Schritt wird DIP-Schalter2 wieder in die Ausgangsstellung zurückgesetzt.

The screenshot shows the 'DELIB Configuration Utility' window. The title bar reads 'DELIB Configuration Utility'. The header features the 'DEDITEC' logo, the product name 'DELIB Configuration Utility', the version 'Ver. 2.19', and 'DELIB 2.19 (64 Bit)' with a power icon. A progress bar at the top indicates the current step is 'Modul Konfiguration'. Below the header, a navigation bar shows five steps: 'Modul Auswahl', 'Modul Konfiguration' (active), 'Konfiguration testen', 'Modul testen', and 'DELIB Einführung'. The main content area is titled 'Konfiguration Module mit Ethernet Schnittstelle' and includes a sub-header 'Netzwerkconfiguration'. It contains several input fields: 'Modul Auswahl' (dropdown menu showing 'RO-ETH/LC(0)'), 'Aktiviert' (checked checkbox), 'Hostname verwenden' (unchecked checkbox), 'IP Adresse' (text field with '192.168.1.25'), 'Verschlüsselung' (dropdown menu showing 'Admin'), 'Port' (text field with '9912'), 'Passwort' (password field with '\*\*\*\*\*'), and 'Timeout [msec]' (text field with '5000'). There are three buttons: 'I/O Namen bearbeiten', 'Einstellungen zur Verschlüsselung auf das Modul übertragen', and 'Authentifizierung abbrechen'. At the bottom, there are three large buttons: 'Hauptmenü', 'Test', and 'Fertig'. A status bar at the very bottom shows three colored squares (black, grey, grey).

DELIB Configuration Utility

DEDITEC DELIB Configuration Utility Ver. 2.19 DELIB 2.19 (64 Bit)

Modul Auswahl Modul Konfiguration Konfiguration testen Modul testen DELIB Einführung

### Konfiguration Module mit Ethernet Schnittstelle

Hier werden die wichtigsten Parameter für die Kommunikation mit einem Ethernet Modul festgelegt.

**Netzwerkconfiguration**

Modul Auswahl: RO-ETH/LC(0)

☒ Aktiviert ☐ Hostname verwenden

IP Adresse: 192.168.1.25 Verschlüsselung: Admin I/O Namen bearbeiten

Port: 9912 Passwort: \*\*\*\*\*

Timeout [msec]: 5000 Einstellungen zur Verschlüsselung auf das Modul übertragen

Schritt 2: DIP-Schalter 2 (Schreibschutz) des ETH-Moduls auf Stellung 0 (0=OFF, 1=ON) setzen Authentifizierung abbrechen

Hauptmenü Test Fertig

Die Authentifizierung wurde erfolgreich abgeschlossen. Sie sind nun temporär berechtigt, System-Einstellungen zu editieren.

The screenshot shows the 'DELIB Configuration Utility' window. The title bar reads 'DELIB Configuration Utility'. The header features the 'DEDITEC' logo, the product name 'DELIB Configuration Utility', the version 'Ver. 2.19', and 'DELIB 2.19 (64 Bit)' with a power icon. A navigation bar contains five tabs: 'Modul Auswahl', 'Modul Konfiguration' (which is active and highlighted with a black arrow), 'Konfiguration testen', 'Modul testen', and 'DELIB Einführung'. The main content area is titled 'Konfiguration Module mit Ethernet Schnittstelle'. Below this, it states: 'Hier werden die wichtigsten Parameter für die Kommunikation mit einem Ethernet Modul festgelegt.' A 'Manual' icon is visible on the right. Under the heading 'Netzwerkconfiguration', there are several input fields and checkboxes: 'Modul Auswahl' (dropdown menu showing 'RO-ETH/LC(0)'), 'Aktiviert' (checked checkbox), 'Hostname verwenden' (unchecked checkbox), 'IP Adresse' (text field with '192.168.1.25'), 'Verschlüsselung' (dropdown menu showing 'Admin'), 'Port' (text field with '9912'), 'Passwort' (password field with '\*\*\*\*\*'), and 'Timeout [msec]' (text field with '5000'). There are two buttons: 'I/O Namen bearbeiten' and 'Einstellungen zur Verschlüsselung auf das Modul übertragen'. A green status message 'Authentifizierung erfolgreich' is displayed. At the bottom, there are three buttons: 'Hauptmenü', 'Test', and 'Fertig', followed by three small square icons.

Nach erfolgreicher Authentifizierung werden abschließend die Verschlüsselungseinstellungen auf das Modul übertragen.

The screenshot shows the 'DELIB Configuration Utility' window. The title bar reads 'DELIB Configuration Utility'. The main header features the 'DEDITEC' logo, the product name 'DELIB Configuration Utility', the version 'Ver. 2.19', and 'DELIB 2.19 (64 Bit)' with a power icon. A progress bar at the top indicates the current step is 'Modul Konfiguration', with other steps being 'Modul Auswahl', 'Konfiguration testen', 'Modul testen', and 'DELIB Einführung'. The main content area is titled 'Konfiguration Module mit Ethernet Schnittstelle' and includes a sub-header 'Netzwerkconfiguration'. Below this, there are several input fields: 'Modul Auswahl' (a dropdown menu showing 'RO-ETH/LC(0)'), 'Aktiviert' (checked checkbox), 'Hostname verwenden' (unchecked checkbox), 'IP Adresse' (text field with '192.168.1.25'), 'Verschlüsselung' (dropdown menu showing 'Admin'), 'Port' (text field with '9912'), 'Passwort' (password field with '\*\*\*\*\*'), and 'Timeout [msec]' (text field with '5000'). To the right of these fields is a teal button labeled 'I/O Namen bearbeiten'. Below the input fields is a teal button labeled 'Einstellungen zur Verschlüsselung auf das Modul übertragen'. At the bottom of the window, there are three large buttons: 'Hauptmenü', 'Test', and 'Fertig'. A green status message at the bottom reads 'Einstellungen zur Verschlüsselung erfolgreich übertragen'.

DELIB Configuration Utility

DEDITEC DELIB Configuration Utility Ver. 2.19 DELIB 2.19 (64 Bit)

Modul Auswahl Modul Konfiguration Konfiguration testen Modul testen DELIB Einführung

### Konfiguration Module mit Ethernet Schnittstelle

Hier werden die wichtigsten Parameter für die Kommunikation mit einem Ethernet Modul festgelegt.

**Netzwerkconfiguration**

Modul Auswahl RO-ETH/LC(0)

☒ Aktiviert ☐ Hostname verwenden

IP Adresse 192.168.1.25 Verschlüsselung Admin

Port 9912 Passwort \*\*\*\*\*

Timeout [msec] 5000

I/O Namen bearbeiten

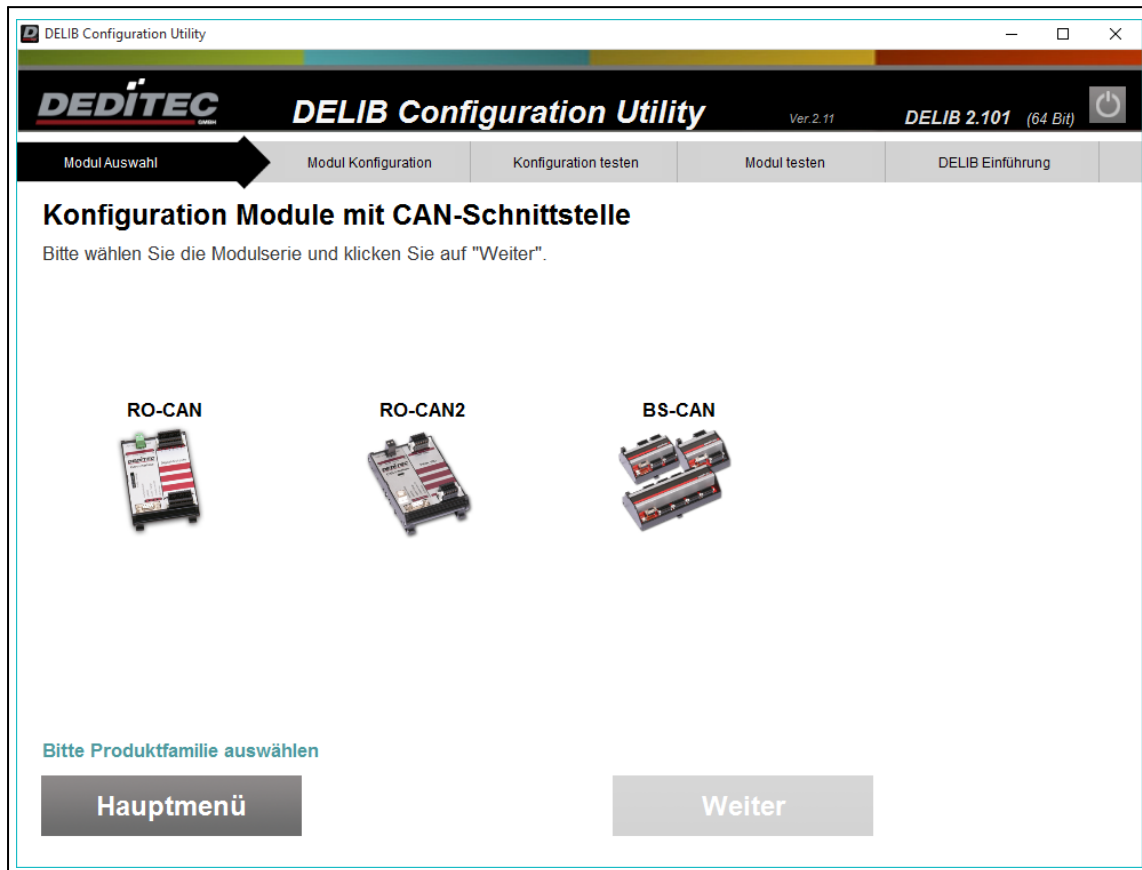
Einstellungen zur Verschlüsselung auf das Modul übertragen

Einstellungen zur Verschlüsselung erfolgreich übertragen

Hauptmenü Test Fertig

### 3.2.3.2.3. Modul Konfiguration CAN

Wenn Sie eine neue CAN-Konfiguration erstellen, müssen Sie zuerst die Produktfamilie auswählen.



Bei der Konfiguration von RO-CAN Modulen geschieht die Konfiguration des Produktes über eine serielle Verbindung. Hierzu muss der entsprechende COM Port festgelegt werden.

Zusätzlich muss die Anzahl der Übertragungsversuche im Falle eines Kommunikationsfehlers angegeben werden.

Mit dem Button "Test" wird geprüft, ob eine Kommunikation über den angegebenen COM Port möglich ist.

The screenshot shows the 'DELIB Configuration Utility' window. The title bar reads 'DELIB Configuration Utility'. The header area features the 'DEDITEC' logo, the title 'DELIB Configuration Utility', the version 'Ver. 2.11', and the model 'DELIB 2.101 (64 Bit)' with a power icon. A progress bar at the top is divided into five segments: 'Modul Auswahl' (grey), 'Modul Konfiguration' (black with white arrow), 'Konfiguration testen' (grey), 'Modul testen' (grey), and 'DELIB Einführung' (grey). Below the progress bar, the 'Modul Konfiguration' section is active. It contains the text: 'Konfiguration Module mit CAN-Schnittstelle' and 'Die Konfiguration der CAN-Schnittstelle erfolgt seriell. Hierzu muss ein COM-Port ausgewählt werden. Zusätzlich kann die Anzahl der maximalen Übertragungsversuche im Falle eines Kommunikationsfehlers festgelegt werden.' Below this text are three configuration fields: 'Modul Auswahl' with a dropdown menu showing 'RO-CAN(0)', 'COM Port' with a dropdown menu showing 'COM1', and 'Übertragungsversuche' with a dropdown menu showing '5'. At the bottom of the window are three buttons: 'Zurück', 'Test', and 'Fertig'.

Die Konfiguration eines RO-CAN2 oder BS-CAN Moduls findet über die USB-Schnittstelle statt. Wie bei einem USB-Interface muss bei der Verwendung identischer Produkte eine unterschiedliche Modul-Nr zur eindeutigen Identifikation vergeben werden.

siehe Kapitel **Beispiel zur Konfiguration identischer USB-Module**

Zusätzlich zur Modul-Nr kann auch die Anzahl an Übertragungsversuchen im Falle eines Fehlers festgelegt werden.

The screenshot shows the 'DELIB Configuration Utility' window. The title bar reads 'DELIB Configuration Utility'. The header bar features the 'DEDITEC' logo, the title 'DELIB Configuration Utility', the version 'Ver. 2.11', and the product name 'DELIB 2.101 (64 Bit)' with a power icon. The main menu has five tabs: 'Modul Auswahl', 'Modul Konfiguration' (which is active and highlighted with a black arrow), 'Konfiguration testen', 'Modul testen', and 'DELIB Einführung'. The active tab displays the 'Konfiguration Module mit USB Schnittstelle' section. Below the title, there is explanatory text: 'Hier kann die Modul-Nr. des ausgewählten Moduls geändert werden. Die Modul-Nr. dient zur Unterscheidung mehrerer identischer USB-Module einer USB-Serie. Zusätzlich kann die Anzahl der maximalen Übertragungsversuche im Falle eines Kommunikationsfehlers festgelegt werden.' On the left, under 'Gefundene USB-Module', there is a table with two columns: 'Module' and 'Nr'. The first row shows 'RO-CAN2' and '1'. To the right of the table are three input fields: 'Aktuelle Modul-Nr.' with a text box containing '1', 'Neue Module-Nr.' with a dropdown menu showing '0', and 'Übertragungsversuche (für alle USB-Module)' with a dropdown menu showing '5'. At the bottom, there is a blue text prompt: 'Klicken Sie "Weiter" zum Übernehmen der neuen Modul-Nr.' and three buttons: 'Hauptmenü', 'Suchen', and 'Weiter'.

Module	Nr
RO-CAN2	1

Aktuelle Modul-Nr.

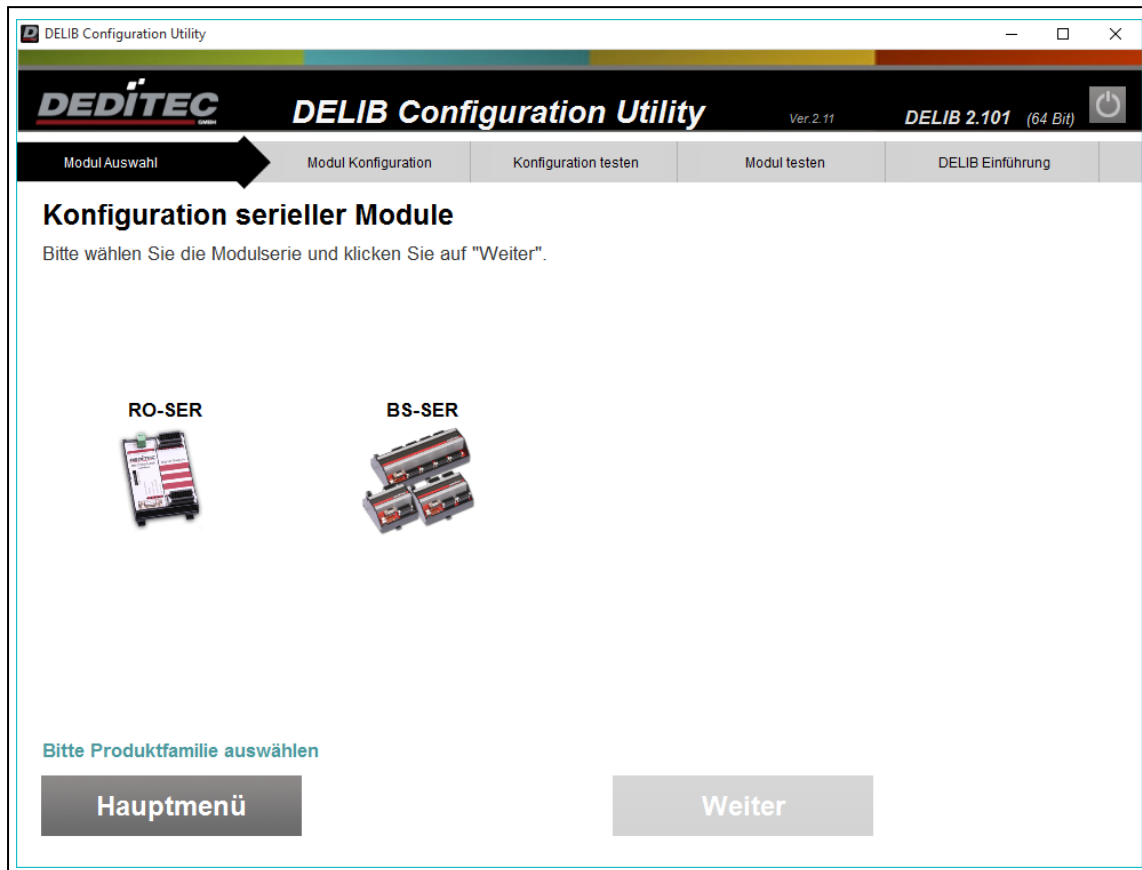
Neue Module-Nr.

Übertragungsversuche (für alle USB-Module)

Klicken Sie "Weiter" zum Übernehmen der neuen Modul-Nr.

#### 3.2.3.2.4. Modul Konfiguration Seriell

Wenn Sie eine neue serielle Konfiguration erstellen, müssen Sie zuerst die Produktfamilie auswählen.



Bei der Konfiguration der seriellen Produkte muss der entsprechende COM-Port festgelegt werden.

Zusätzlich muss die Anzahl der Übertragungsversuche im Falle eines Kommunikationsfehlers angegeben werden.

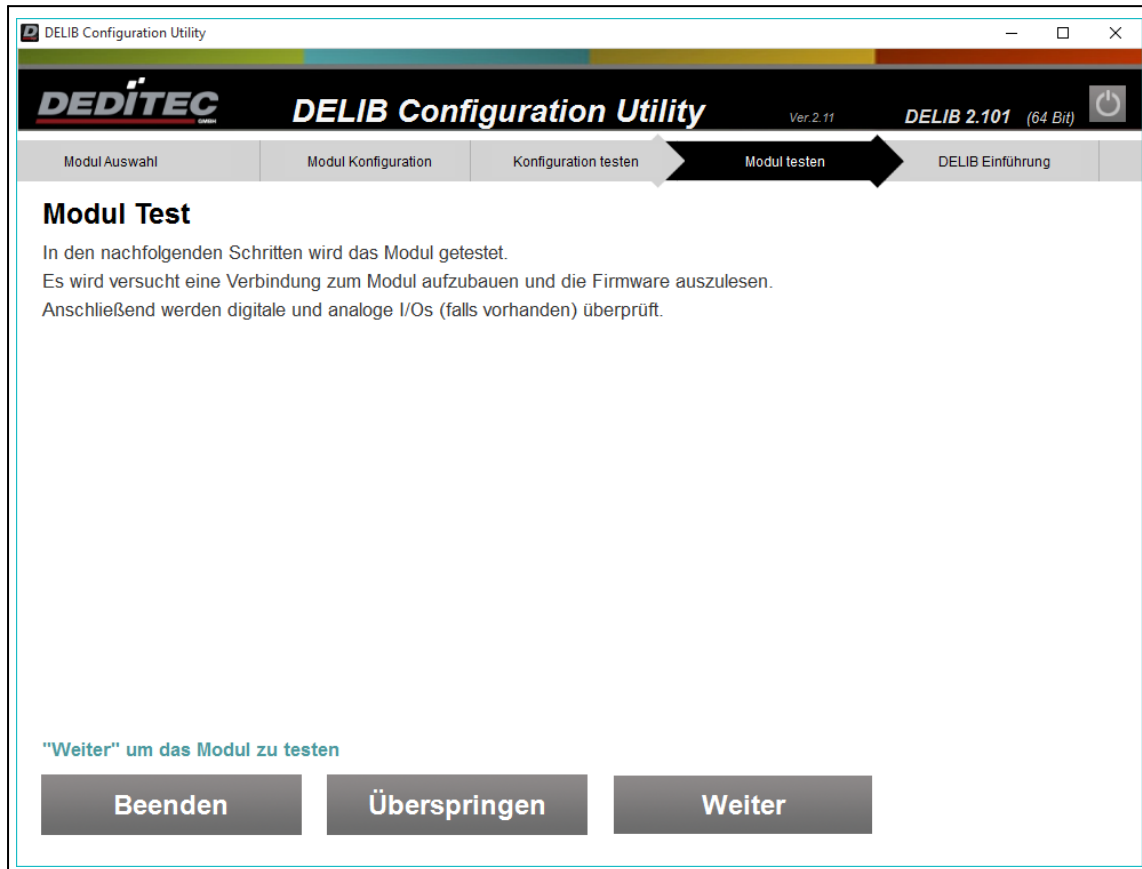
Mit dem Button "Test" wird geprüft, ob eine Kommunikation über den angegebenen COM-Port möglich ist.

The screenshot shows the 'DELIB Configuration Utility' window. The title bar includes the DEDITEC logo, the application name 'DELIB Configuration Utility', the version 'Ver.2.11', and the device model 'DELIB 2.101 (64 Bit)'. The main menu has five tabs: 'Modul Auswahl', 'Modul Konfiguration' (which is active and highlighted with a black arrow), 'Konfiguration testen', 'Modul testen', and 'DELIB Einführung'. Below the tabs, the section 'Konfiguration serieller Module' is displayed, with a subtitle 'Hier werden die wichtigsten Parameter für die Kommunikation mit einem seriellen Modul festgelegt.' Under the 'Modul Konfiguration' heading, there are three configuration items: 'Modul Auswahl' set to 'RO-SER(0)', 'COM Port' set to 'COM1', and 'Übertragungsversuche' set to '5'. At the bottom of the window, there are three buttons: 'Zurück', 'Test', and 'Fertig'.

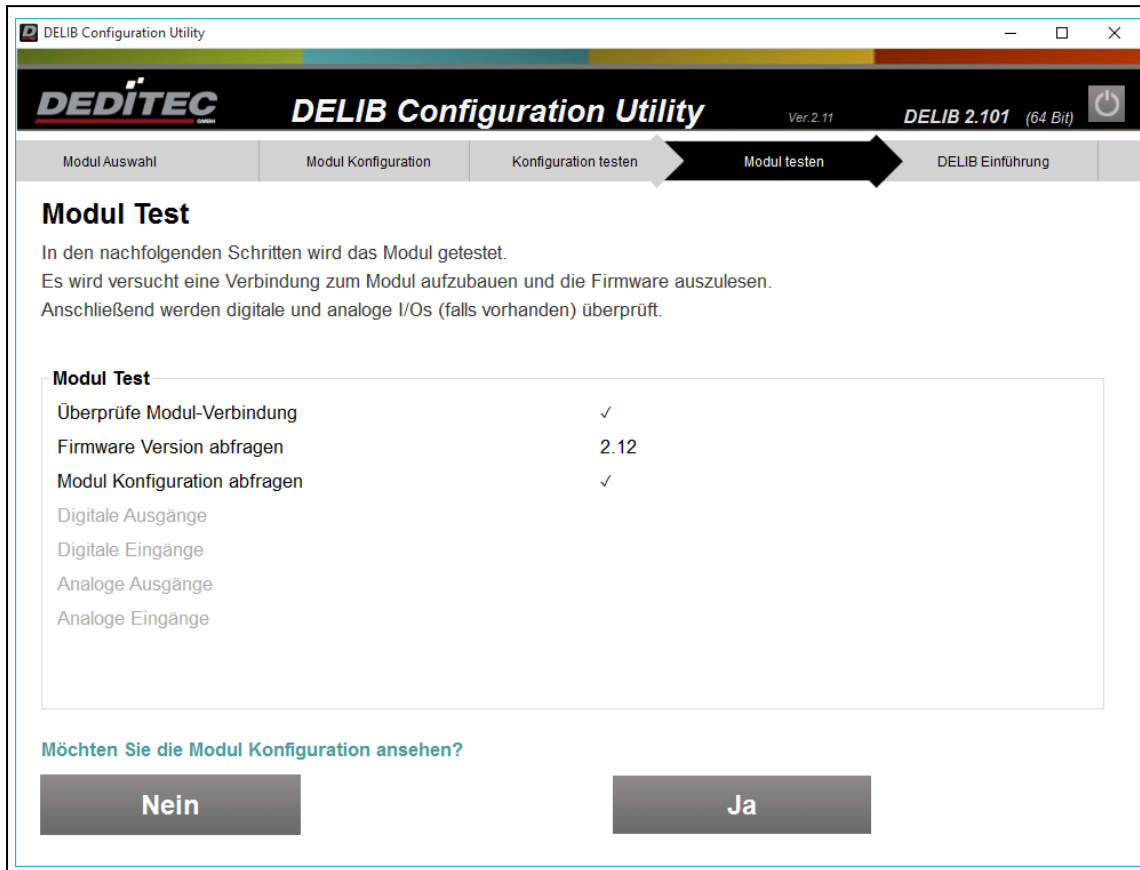


### 3.2.3.3. Modul testen

Nachdem die Konfiguration der Schnittstelle durchgeführt wurde, kann anschließend das Produkt getestet werden.



Test der Firmware und Anzeige der Modul-Info.



Die Modul-Info zeigt alle Eigenschaften des Produktes. Neben der Anzahl der vorhandenen I/Os werden auch die unterstützten Software-Features angezeigt.

DT\_ModuleInfo

×

General	Digital I/O	Analog I/O	Special
SW_FEATURE_1 e3373007	Digital Inputs 8	Analog Inputs 16	Stepper 2
HW_INTERFACE_1 01000003	Digital Outputs 8	Analog Outputs 4	
Firmware-Revision 2.11	Digital In-/Outputs 0	Temperature Inputs 4	
Main-Module: RO	Digital Input FlipFlops 8		
	Digital Input Counter 8		
	Pulse Gen Outputs 0		
	CNT8 0		
	Digital PWM Outputs 0		

Features-General	Features-Digital I/O	Features-Analog I/O	Features-Special
Supported by FW OK	DI Commands OK	DA Commands OK	Watchdog Commands -
Dev IO registry error OK	DI CNT Commands OK	AD Commands OK	Stepper Commands OK
AD FIFO OK	DI CNT Latch Feature -	Pt100 Commands OK	
Set-Clr Bit Commands OK	DI FF Commands OK		
EEPROM RN23 -	DO Commands OK		
EEPROM E2_2K -	DO Time Commands OK		
DX1 Mode -	PWM Commands -		
Support Channel Names OK	TTL Commands -		
HW-INT Supported by FV OK	PulseGen Commands -		
ETH OK	CNT8 Commands -		
CAN -	Auto-Off Timeout Commar OK		
RS232 -			
RS485 -			
USB1 -			
USB2 -			

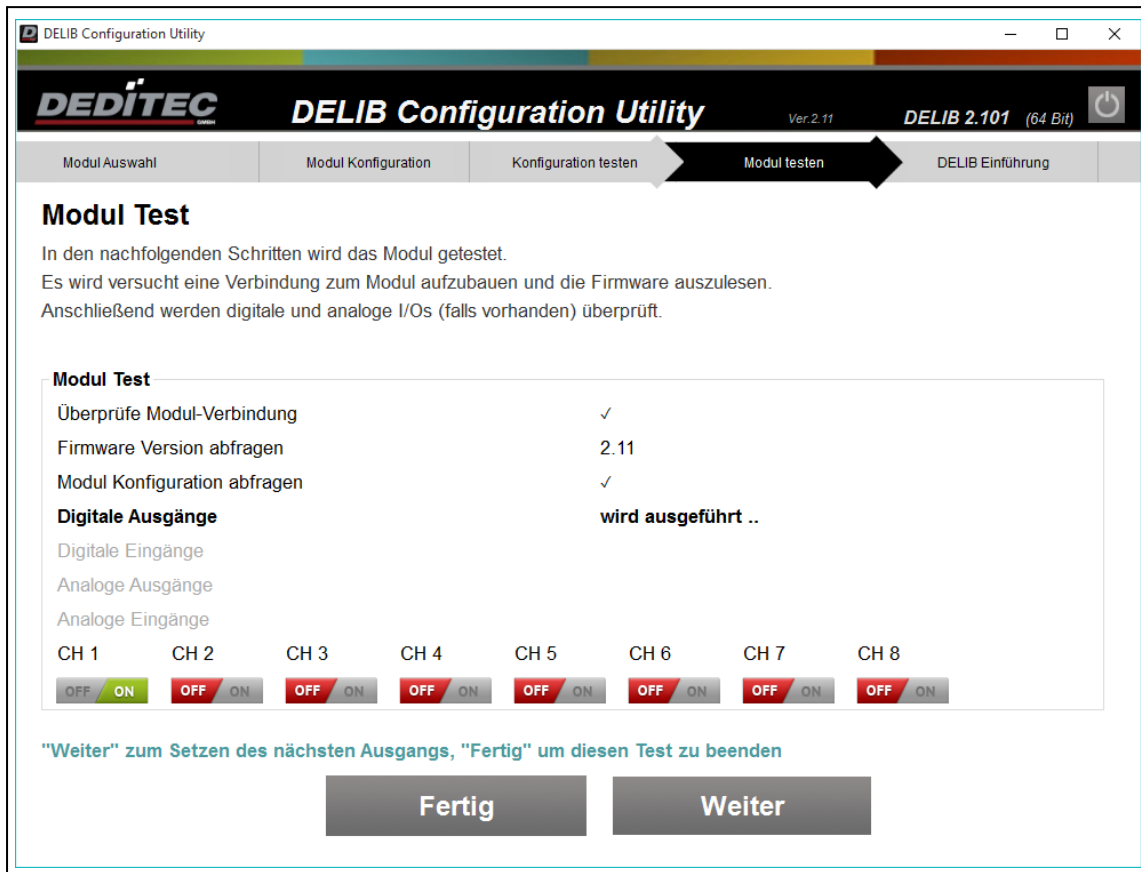
  

Submodule-Info's

Sub: 0 - RO-AD16DA4	FW:2.11
Sub: 1 - RO-08_R8	FW:2.10
Sub: 2 - RO-STEPPER2	FW:1.29
Sub: 3 - RO-PT100	FW:1.03

EXIT

Es folgt ein Test der I/Os. In diesem Beispiel werden die digitalen Ausgänge geschaltet.

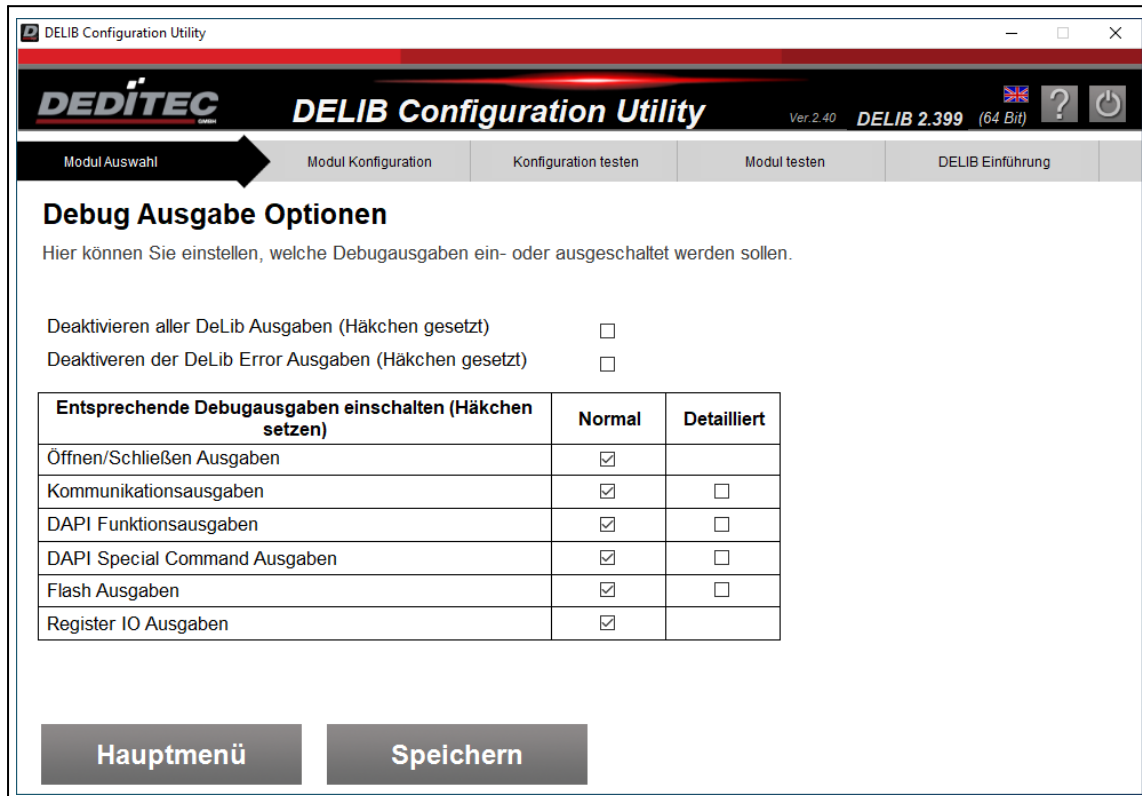


Wurden alle Tests erfolgreich durchlaufen, ist das Produkt einsatzbereit.

### 3.2.3.4. Debug Optionen einstellen

Über den Knopf "Debug Ausgabe Optionen" gelangen Sie in das folgende Optionsmenü.

Dort können Sie einstellen, welche Debugausgaben Sie ein- oder ausschalten möchten.



The screenshot shows the 'DELIB Configuration Utility' window. The title bar reads 'DELIB Configuration Utility'. The main header features the 'DEDITEC' logo, the title 'DELIB Configuration Utility', and version information 'Ver. 2.40 DELIB 2.399 (64 Bit)'. Below the header is a navigation bar with five tabs: 'Modul Auswahl', 'Modul Konfiguration', 'Konfiguration testen', 'Modul testen', and 'DELIB Einführung'. The 'Modul Auswahl' tab is currently selected, indicated by a black arrow pointing to it. The main content area is titled 'Debug Ausgabe Optionen' and contains the following text: 'Hier können Sie einstellen, welche Debugausgaben ein- oder ausgeschaltet werden sollen.' Below this text are two checkboxes: 'Deaktivieren aller DeLib Ausgaben (Häkchen gesetzt)' and 'Deaktivieren der DeLib Error Ausgaben (Häkchen gesetzt)', both of which are currently unchecked. A table follows, titled 'Entsprechende Debugausgaben einschalten (Häkchen setzen)'. The table has three columns: the first column lists the debug output types, the second column is labeled 'Normal', and the third column is labeled 'Detailliert'. The table contains six rows of data. At the bottom of the window are two buttons: 'Hauptmenü' and 'Speichern'.

Entsprechende Debugausgaben einschalten (Häkchen setzen)	Normal	Detailliert
Öffnen/Schließen Ausgaben	<input checked="" type="checkbox"/>	
Kommunikationsausgaben	<input checked="" type="checkbox"/>	<input type="checkbox"/>
DAPI Funktionsausgaben	<input checked="" type="checkbox"/>	<input type="checkbox"/>
DAPI Special Command Ausgaben	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Flash Ausgaben	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Register IO Ausgaben	<input checked="" type="checkbox"/>	

### 3.2.4. Benutzung des Modulselectors

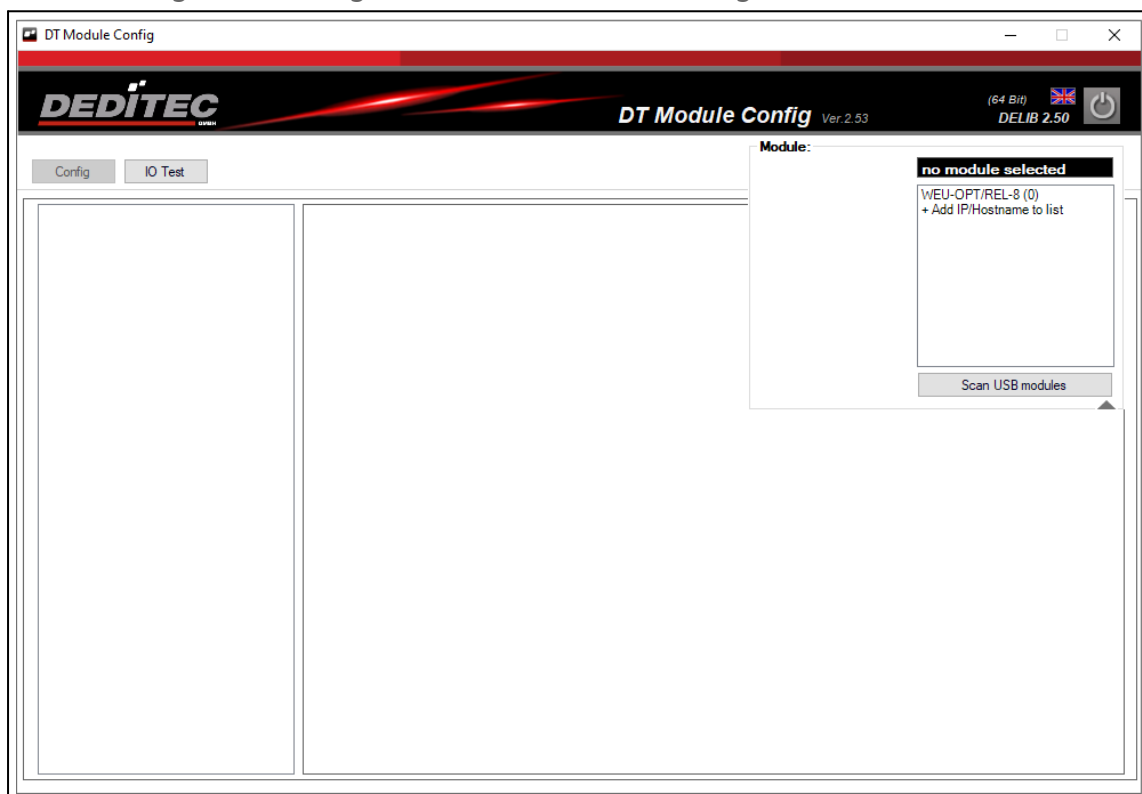
Um unsere Produkte mit der DEDITEC-Software benutzen zu können, müssen diese über den Modul Selector ausgewählt werden.

Je nach Modul, kann dies über verschiedene Schnittstellen bewerkstelligt werden.

#### 3.2.4.1. via USB

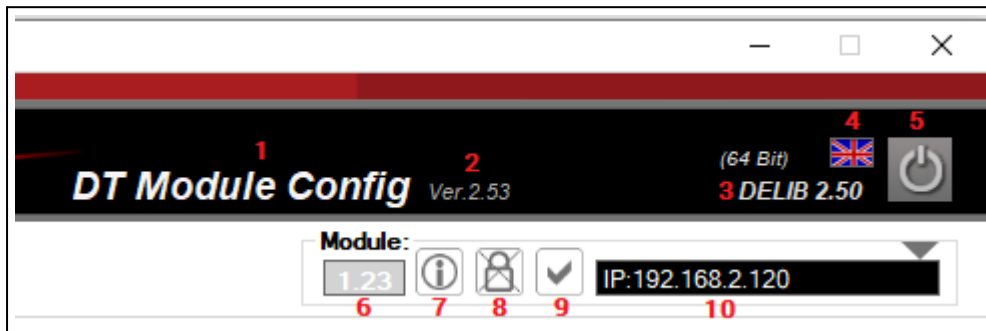
Haben Sie das Modul über die USB-Schnittstelle mit dem PC verbunden, kann das Modul direkt über einen Klick auf den Modul Selector in der rechten oberen Ecke ausgewählt werden.

Anschließend können Sie im Netzwerkbereich unter LAN - Konfiguration oder WiFi - Konfiguration die gewünschte Netzwerkkonfiguration vornehmen.



### 3.2.4.2. Modul Info

Bei einer erfolgreichen Verbindung mit dem Modul werden nun im Bereich des Modul Selector verschiedene Information, wie unten beschrieben, dargestellt.



#### Beschreibung:

1. Zeigt den Namen der verwendeten DEDITEC Software an
2. Zeigt die aktuell verwendete Versionsnummer der Software an
3. Zeigt die aktuell verwendete DELIB Version an
4. Durch einen Klick auf das Fahnnensymbol lässt sich die Sprache zwischen deutsch und englisch ändern
5. Schließt das Programm
6. Zeigt die aktuell verwendete Firmware Ihres Moduls an
7. Durch einen Klick auf die Informationsschaltfläche, öffnet sich das Informationsfenster des Moduls (s. Bild unten)
8. Zeigt an, ob eine ver- oder entschlüsselte Kommunikation mit dem Modul stattfindet
9. Zeigt den Kommunikationsstatus mit dem Modul an
10. Je nach Verbindungsart wird hier die IP oder der Boardname des aktuell verwendeten Moduls angezeigt

## Informationsfenster

Je nach angeschlossenem Modul werden hier Informationen zu dem verwendeten Interface und den Submodulen angezeigt.

Unter Anderem können Sie hier die Anzahl der angeschlossenen Ein- bzw. Ausgänge einsehen und welche DEDITEC Befehle unterstützt werden.

The screenshot shows a window titled "DT\_ModuleInfo" with a close button (X) in the top right corner. The window displays information for a module, organized into four main sections: General, Digital I/O, Analog I/O, and Special. Each section has a list of parameters and their values.

General		Digital I/O		Analog I/O		Special	
SW_FEATURE_1	0300a0c5	Digital Inputs	0	Analog Inputs	0	Stepper	0
HW_INTERFACE_1	00000103	Digital Outputs	8	Analog Outputs	0		
Firmware-Revision	1.23	Digital In-/Outputs	0	Temperature Inputs	0		
Main-Module:	-	Digital Input FlipFlops	0				
		Digital Input Counter	0				
		Pulse Gen Outputs	0				
		CNT8	0				
		Digital PWM Outputs	0				

Features-General		Features-Digital I/O		Features-Analog I/O		Features-Special	
Supported by FW	OK	DI Commands	-	DA Commands	-	Watchdog Commands	-
Dev IO registry error	OK	DI CNT Commands	-	AD Commands	-	Stepper Commands	-
RO-AD FIFO	-	DI CNT Latch Feature	-	Pt100 Commands	-		
NET-Software FIFO	-	DI FF Commands	-				
Set-Clr Bit Commands	OK	DO Commands	OK				
EEPROM RN23	-	DO Time Commands	-				
EEPROM E2_2K	-	PWM Commands	-				
DX1 Mode	-	TTL Commands	-				
Support Channel Names	-	PulseGen Commands	-				
HW-INT Supported by FW	OK	CNT8 Commands	-				
ETH	OK	Auto-Off Timeout	OK				
CAN	-	Auto-Off Timeout Mask	OK				
RS232/485	-	FTDI Userbyte 6	0x0				
USB1	-						
USB2	-						

An "EXIT" button is located in the bottom right corner of the window.

In diesem Beispiel wurde ein WEU-RELAIS-8 aus unserer Startet-Serie mit 8 digitalen Ausgängen über Ethernet angeschlossen.



### **3.2.5. DELIB Module Config**

Das Module Config ist eine neue Anwendung zur Konfiguration und zum Testen unserer Produkte. Dieses Programm ist im Installationspaket unserer DELIB Treiberbibliothek enthalten.

#### **3.2.5.1. Modul Konfigurationen**

Im Konfigurations-Bereich können Konfigurationseinstellungen des Moduls eingesehen oder geändert werden.

### 3.2.5.1.1. Modul-Infoseite

Mit dem Module Config lässt sich Ihr WEU-Modul nicht nur schnell und einfach konfigurieren, Sie können sich auch alle wichtigen Modulinformationen auf nur einen Blick anzeigen lassen.

**Info**  
Hier finden Sie Informationen zu Ihren Moduleigenschaften

Modul-Name	WEU-08-R8
Modul-ID	40 (dezimal) / 0x0028 (hex)
Firmware-Version	Ver. 1,23

Form für Module Overview (Ver. 1.01)

#### Modul-Name

Zeigt den Namen des aktuell verwendeten DEDITEC Modules an.

#### Modul-ID

Zeigt die ID Ihres verwendeten Moduls an. Diese wird für das Programmieren eigener Software mit DEDITEC Befehlen benötigt.

#### Firmware-Revision

Zeigt die aktuelle auf dem Modul installierte Firmware-Version an.

### 3.2.5.1.2. Modul-Identifikation

Identifizieren Sie das Modul, welches Sie gerade mit dem Modul Config ansprechen, um Verwechslungen vorzubeugen.

Dies ist besonders hilfreich, wenn mehrere Module gleichzeitig in Betrieb sind.

Durch das Betätigen von "Start" wird die Identifikation gestartet.

Es fängt nun die Status-LED wiederholt an zu blinken.

Dieser Vorgang wird durch drücken von "Stop" beendet.

### Identifikation

Hier können Sie Ihr Modul identifizieren

Identify module

**Modulidentifikation läuft!**

Mit der Identifikations-Funktion können Sie feststellen, welches Ihrer Module momentan durch das Modul Config angesteuert wird. Dies ist besonders hilfreich, bei der gleichzeitigen Verwendung mehrerer Module. Durch das Betätigen der 'Start' Taste blinken folgende LEDs zur Identifikation

- bei unseren ETH\_LC-Modulen blinkt die 'Int.Act' LED
- bei unseren WEU-Modulen blinkt die 'Status-LED'

Das Bedienen der 'Stop' Taste beendet diesen Vorgang.

FormConfModulIdent (Ver. 1.01)

### 3.2.5.1.3. LAN Netzwerkinformationen

Alle wichtigen LAN Netzwerkinformationen auf einen Blick.

Auf dieser Informationsseite finden Sie die aktuellen LAN-Einstellungen Ihres Modules.

**Info**  
Hier finden Sie die Netzwerkinformationen des ausgewählten Ethernet-Produktes

MAC address	40:F5:20:44:60:EB
Board name (Hostname in DHCP mode)	WEU-O8-R8
LAN status	Static-IP success
DHCP active	<input type="checkbox"/>
IP address	192.168.2.71
Subnet mask	255.255.255.0
Default gateway	192.168.2.254
TCP port	9912

☐ Auto Refresh

FormOfnNetwork LANInfo (Ver. 1.01)

#### MAC-Adresse

Die MAC-Adresse ist die physikalische Adresse des Produktes und ist fest mit der Hardware verbunden.

#### Board Name

Zeigt den aktuellen Board Name Ihres Modules an.

#### LAN-Status

Hier wird der Verbindungsstatus Ihres angeschlossenen Moduls angezeigt.

Sollte bei Ihnen der Status "Query not supported (FW-Update)" dargestellt werden, benötigt Ihre Modul eine aktuellere Firmware.

#### DHCP active

Zeigt an, ob das Modul über DHCP verbunden ist.

#### IP-Adresse, Netzmaske, Standard Gateway und TCP-Port


Zeigt die aktuelle Netzwerkkonfiguration, mit der das Modul verbunden ist, an.

#### 3.2.5.1.4. LAN Netzwerkeinstellungen

Hier können Sie Änderungen an den Netzwerkeinstellungen des ausgewählten WEU-Moduls vornehmen.

### Konfiguration

Hier können Sie die Netzwerkeinstellungen des ausgewählten Ethernet-Produktes ändern

Board name (Hostname in DHCP mode)	<input type="text" value="WEU-08-R8"/>
Network configuration protection active 	<input type="checkbox"/>
DHCP active	<input type="checkbox"/>
IP address	<input type="text" value="192.168.2.71"/>
Subnet mask	<input type="text" value="255.255.255.0"/>
Default gateway	<input type="text" value="192.168.2.254"/>
TCP port	<input type="text" value="9912"/>

Form("ToNetwork1\_AbIConfin\_Ver. 1.01")

#### Board Name

Der Board Name kann zur Geräteidentifizierung genutzt werden. Ist DHCP aktiv, wird der Board Name als Hostname verwendet.

Diese Option ist besonders bei der Verwendung mehrerer Module sehr hilfreich.

So können Sie zum Beispiel einem Modul einen speziellen Board Name wie "Garage" oder "Gartenlaube" vergeben. Im Modul Selector können Sie das Modul dann unter diesem Namen direkt ansteuern.

Mehr Infos zum Anbinden des Moduls per Board Name

siehe Kapitel: **Benutzung des Modulselectors**

### Network configuration protection active

Wenn diese Option aktiviert ist, können Netzwerkkonfiguration nur noch über die Weboberfläche geändert werden.

Dies verhindert unautorisierten Zugriff auf die Netzwerkkonfiguration (Beispielsweise über das Modul Config).

### DHCP active

Ist diese Option aktiviert, versucht das Gerät beim Start eine gültige IP-Adresse von einem DHCP Server im Netzwerk zu beziehen.

Der Board Name wird als Hostname verwendet.

### IP address, Subnet mask, Default gateway und TCP port

Diese Einstellungen werden verwendet, wenn DHCP deaktiviert ist. Fragen Sie gegebenenfalls bitte Ihren Systemadministrator.

### Werkseinstellungen laden

Hier wird die IP-Konfiguration auf die Werkseinstellungen zurückgesetzt. Diese sehen wie folgt aus:

Werkeinstellungen	
Board name	Modul abhängig
Network protection	off
DHCP	off
IP-Adresse	192.168.1.1
Subnet mask	255.255.255.0
Default gateway	192.168.1.254
TCP Port	9912

### 3.2.5.1.5. WiFi Netzwerkinformationen

Alle wichtigen WiFi Netzwerkinformationen auf einen Blick.

Auf dieser Informationsseite finden Sie die aktuellen WiFi-Einstellungen Ihres Moduls.

### Info

Hier finden Sie die Netzwerkinformationen des ausgewählten WiFi-Produktes

MAC address	40:F5:20:44:60:E8
Board name (Hostname in DHCP mode)	WEU-O8-R8_W
WLAN status	starting WIFI connection..
WLAN active	<input checked="" type="checkbox"/>
IP address	0.0.0.0
Subnet mask	0.0.0.0
Default gateway	0.0.0.0
TCP port	9912
Router name	TESTssid
Password	TESTpwd

☒ Auto Refresh

FormTitleNetworkWiFiInfo (Ver. 1.01)

### MAC-Adresse

Die MAC-Adresse ist die physikalische Adresse des Produkts und ist fest mit der Hardware verbunden.

### Board Name

Zeigt den aktuellen Board Name Ihres Moduls an.

### **WLAN-Status**

Hier wird der Verbindungsstatus Ihres angeschlossenen Moduls angezeigt.  
Sollte bei Ihnen der Status "Query not supported (FW-Update)" dargestellt werden, benötigt Ihr Modul eine aktuellere Firmware.

### **WLAN active**

Zeigt an, ob das Modul über WLAN verbunden ist.

### **IP-Adresse, Netzmaske, Standard Gateway und TCP-Port**

Zeigt die aktuelle Netzwerkkonfiguration mit der das Modul verbunden ist, an.

### **Router name**

Zeigt an, welcher Router Name zum Verbinden via WLAN verwendet wird.

### **Password**

Zeigt das verwendete Routerpasswort an.




### 3.2.5.1.6. WiFi Netzwerkeinstellungen

Hier können Sie die Änderungen an den WiFi-Einstellungen Ihres WEU-Moduls vornehmen.

## Konfiguration

Hier können Sie die Netzwerkeinstellungen des ausgewählten WiFi-Produktes ändern

Board name (Hostname in DHCP mode)	<input type="text" value="WEU-08-R8_W"/>
WLAN active	<input type="checkbox"/>
Router name	<input type="text" value="TESTssid"/>
Password	<input type="text" value="TESTpwd"/>
TCP port	<input type="text" value="9912"/> 
<input type="button" value="Restart network"/>	

Form für Netzwerk-WiFi Konfig (Ver. 1.01)

#### Board name

Der Board Name kann zur Geräteidentifizierung genutzt werden. Ist DHCP aktiv, wird der Board Name als Hostname verwendet.

Diese Option ist besonders bei der Verwendung mehrerer Module sehr hilfreich.

So können Sie zum Beispiel einem Modul einen speziellen Board Name wie "Garage" oder "Gartenlaube" vergeben. Im Modul Selector können Sie das Modul dann unter diesem Namen direkt ansteuern.

(Mehr Infos zum Anbinden des Modules per Board Name siehe Kapitel: Benutzung des Modulselectors)

### **WLAN active**

Mit dieser Option können Sie das WLAN Ihres Moduls aktivieren oder deaktivieren.

### **Router name**

Hier können Sie den Routername eintragen, welcher bei einer Verbindung via WLAN verwendet werden soll.

Fragen Sie hierfür gegebenenfalls Ihren Systemadministrator.

### **Password**

Hier können Sie das Routerpasswort des verwendeten Routers eingetragen werden.

Fragen Sie hierfür gegebenenfalls Ihren Systemadministrator.

### **TCP port**

Hier wird der verwendete TCP-Port dargestellt. Eine Änderung des Ports kann nur bei den LAN Netzwerkkonfigurationen vorgenommen werden.

### **Werkseinstellungen laden**

Hier wird die WiFi-Konfiguration auf die Werkseinstellungen zurückgesetzt. Diese sehen wie folgt aus:

<b>Werkseinstellungen</b>	
Board name	Modul abhängig
WLAN active	off
Routername	TESTssid
Password	TESTpwd

### 3.2.5.1.7. WiFi WPS-Verbindung

Hier können Sie Ihr WEU-Modul mit Hilfe der WPS-Funktion mit Ihrem PC-Netzwerk verbinden.

Klicken Sie hierfür den WPS-Knopf an Ihrem Router. Informationen dazu finden Sie im Handbuch Ihres Netzwerkgerätes.

Klicken Sie während des Suchlaufs Ihres Routers auf den "WPS-Start"-Knopf im Modul Config oder betätigen Sie den Taster direkt auf dem Board Ihres WEU-Moduls.

Mehr Informationen über den CFG-Taster siehe Kapitel **CFG-Taster**

Bei einer erfolgreichen Verbindung erscheint nun der verwendete Router Name und das Passwort.

## WPS

Hier können Sie eine Verbindung per WPS-Funktion mit Ihrem WiFi-Modul herstellen

WLAN status

starting WPS connection..

Router name

Password

WPS-Start

WPS-Stop

Mit Hilfe der WPS-Funktion, lässt sich Ihr Modul mit nur wenigen Schritten schnell und einfach automatisch mit Ihrem Router verbinden.

Für eine erfolgreiche Verbindung mit dem Netzwerk müssen folgende Punkte durchgeführt werden:

1. Betätigen Sie die 'WPS-Taste' an Ihrem Router (Hilfe dazu finden Sie im Handbuch des Routers)
2. Klicken Sie anschließend auf den obigen 'WPS-Start'-Knopf im Module Config
3. Das Modul verbindet sich nun automatisch mit Ihrem Router

Der aktuellen Status Ihrer Verbindung wird bei 'WLAN Status' angezeigt.  
Das Bedienen der 'Stop-Taste' beendet diesen Vorgang.

☒ Auto Refresh

FormCfgNetworkWiFiWPS (Ver. 1.01)

### 3.2.5.1.8. NTP-Konfiguration

Hier können Änderungen am NTP-Service vorgenommen werden.

#### NTP-Konfiguration

Hier können Sie Änderungen am NTP Service vornehmen

NTP service active	<input checked="" type="checkbox"/>
Server	<input type="text" value="0.de.pool.ntp.org"/>
Port	<input type="text" value="123"/>
Timezone	<input type="text" value="(GMT) Greenwich Mean Time: Dublin, Edinburgh"/>

Werkseinstellung laden

Einstellung speichern

FormfönNetwork NTP (Ver. 1.01)

#### NTP service active

Ist diese Option aktiviert, wird der NTP-Service aktiviert.

#### Server

Hier können Sie den NTP-Server, der verwendet werden soll, einstellen.

#### Port

Hier können Sie den NTP-Port, der verwendet werden soll, einstellen.

## Timezone

Hier können Sie die Zeitzone, die vom Modul verwendet werden soll, einstellen.

## Werkseinstellungen laden

Hier wird die TCP-Verschlüsselungseinstellungen auf die Werkseinstellungen zurückgesetzt. Diese sehen wie folgt aus:

Werkseinstellungen	
NTP service active	on
Server	0.de.pool.ntp.org
Port	123
Timezone	(GMT) Greenwich Mean Time: Dublin, Edinburgh, Lisbon, London

### 3.2.5.1.9. Serielle Konfiguration

Hier können Sie die gesamte Konfiguration unserer Produkte mit serieller Schnittstelle vornehmen.

## Seriell

Hier können Sie die gesamte Konfiguration unserer Produkte mit serieller Schnittstelle vornehmen

Special mode active	<input type="checkbox"/>
Baud rate	<input type="text" value="625000"/>
RS485 modul address	<input type="text" value="0"/>
Echo active	<input type="checkbox"/>
Register modus active	<input type="checkbox"/>

Werkseinstellung laden

Einstellung speichern

FormCtoSenell (Ver 1.01)

### Vorzugsmodus (Special mode)

Im Vorzugsmodus wird das Modul automatisch mit folgenden Einstellungen betrieben:

Baud rate: 115200

Modul-Nr. 0

Echo = Off

Register-Mode = On

### **Baud rate**

Ist der Vorzugsmodus deaktiviert, kann die Geschwindigkeit der Kommunikation festgelegt werden.

625000

250000

125000

115200

57600

50000

38400

19200

9600

4800

2400

1200

600

300

### **RS485 Modul-adresse**

Adresse für die Identifikation im RS485 Bus.

### **Echo**

Seriell empfangene Zeichen werden vom Modul zurückgesendet.

### **Registernodus**

Deaktivieren Sie den Registernodus um den Textmodus zu aktivieren.

### 3.2.5.1.10. I/O Kanal-Namen

Hier können Sie die Kanalnamen Ihres Haupt- bzw. Submoduls einstellen.

#### I/O Kanal-Namen

Hier können Sie die Namen der einzelnen I/O-Kanäle ändern und speichern

Ch. 0	<input type="text" value="Kanal 0"/>	Ch. 8	<input type="text" value="Kanal 8"/>
Ch. 1	<input type="text" value="Kanal 1"/>	Ch. 9	<input type="text" value="Kanal 9"/>
Ch. 2	<input type="text" value="Kanal 2"/>	Ch. 10	<input type="text" value="Kanal 10"/>
Ch. 3	<input type="text" value="Kanal 3"/>	Ch. 11	<input type="text" value="Kanal 11"/>
Ch. 4	<input type="text" value="Kanal 4"/>	Ch. 12	<input type="text" value="Kanal 12"/>
Ch. 5	<input type="text" value="Kanal 5"/>	Ch. 13	<input type="text" value="Kanal 13"/>
Ch. 6	<input type="text" value="Kanal 6"/>	Ch. 14	<input type="text" value="Kanal 14"/>
Ch. 7	<input type="text" value="Kanal 7"/>	Ch. 15	<input type="text" value="Kanal 15"/>

Set default channel name

Einstellung speichern

FormCfnSubmoduleIOnames (Ver. 1.01)

Sie können hier sämtliche Kanäle Ihres Haupt- oder Submoduls individuell benennen und speichern.

#### Hinweis:

Der Kanalname darf maximal 16 Zeichen lang sein.

#### Set default channel name

Schreibt den oben abgebildeten Text als Namensvorschlag in die Textfelder.  
Zum Übernehmen auf das Modul muss zusätzlich gespeichert werden.



### 3.2.5.1.11. CAN Konfiguration

#### 3.2.5.1.11.1. CAN Status

In diesem Bereich finden Sie alle Informationen rund um den Status des CAN-Interfaces und den TX- und RX-Paketen.

Hier werden Ihnen alle wichtigen Informationen zu Ihrem CAN-Interface angezeigt

**Interface**  
Hier finden Sie Informationen über den Status des CAN-Interface.

CAN-Baudrate	1000000 Bit/sec
DT-CAN-CMD-MODE - Modul-Address	100 [hex]
DT-CAN-CMD-MODE - Response-Address	200 [hex]
CAN is active	1
Use Ext ID	1
CAN config mode selection	1

☐ Auto Refresh

FormCfoCANStatus (Ver. 1.00)

#### CAN-Baudrate

Zeigt die aktuell eingestellte Baud rate Ihres CAN-Interfaces an

#### DT-CAN-CMD-MODE- Module-Address

#### DT-CAN-CMD-MODE - Response-Address

#### CAN is active

(Diese Funktion wird nur dargestellt, wenn es von Ihrem Modul unterstützt wird)

#### Use EXT ID

#### CAN config mode selection

Hier werden alle wichtigen Statistiken zu den TX- und RX-Paketen, wie zum Beispiel die Anzahl der gesendeten und empfangenen CAN-Pakete und deren Übertragungsgeschwindigkeit angezeigt.

### Statistik TX/RX

Hier finden Sie Informationen über die gesendeten und empfangenen CAN-Pakete.

**DT-CAN-CMD-MODE**

Frames total

TX0

RX0

TX-Modes	Frames total	Frames/Second	Ø Time/Frame	RX-Modes	Frames total
TX-1	0	0	-	RX-1	0
TX-2	0	0	-	RX-2	0
TX-3	0	0	-	RX-3	0
TX-4	0	0	-	RX-4	0
TX-5	0	0	-	RX-5	0
TX-6	0	0	-	RX-6	0
TX-7	0	0	-	RX-7	0
TX-8	0	0	-	RX-8	0

☒ Auto Refresh

FormCfoCANStatistic (Ver. 1.00)

### 3.2.5.1.11.2. CAN Main

In diesem Bereich können Sie Einstellungen direkt am CAN-Interface vornehmen.

Mit Hilfe dieser Einstellungen lässt sich das CAN Interface konfigurieren.

### Interface

Hier können Sie Einstellungen am CAN-Interface vornehmen.

Baudrate	<input type="text" value="1 MBit/s"/>
Address Mode	<input type="text" value="11 Bit Mode"/>
DT-CAN-CMD-MODE - Modul-Address [hex]	<input type="text" value="FFFFFF"/>
DT-CAN-CMD-MODE - Response-Address [hex]	<input type="text" value="FFFFFF"/>

Einstellung speichern

FormCfaCANConfiaStd (Ver. 1.00)

### Baudrate

Hier kann die Baudrate eingestellt werden, mit der das Modul kommunizieren soll.

### Address Mode

Der Address Mode gibt vor, wie viel Bit zur Adressierung verwendet werden.

#### **DT-CAN-CMD-MODE - Modul-Address[hex]**

Diese Modul-Address legt fest, unter welcher Adresse das Modul im CAN-Bus identifiziert wird.

#### **DT-CAN-CMD-MODE - Response-Address[hex]**

Response-Address gibt vor, an welche Modul-Adresse eine Bestätigung gesendet wird, sobald ein Paket empfangen wurde.

Diese Einstellungen dienen der Konfiguration der angeschlossenen Submodule. Es kann der jeweilige Filter / Modus eingestellt werden, in dem die angeschlossenen Submodule gestartet werden.

### I/O-Init

Hier können Sie die CAN-Einstellungen der angeschlossenen Submodule konfigurieren.

A/D Mode	16 Bit / $\pm 20V$
A/D Filter	None
D/A Mode	16 Bit / $\pm 10V$
Counter Mode	16 Bit Counter (default)
Timeout-Mode	Deactivate
Output Timeout	0.0 sec
CNT48 Mode	Read on rising edge (x1)
CNT48 Submode	Read
CNT48 Filter	20 ns

Einstellung speichern

FormCfaCANConfiaIO (Ver. 1.00)

### A/D Mode

Der Wertebereich gibt den Bereich an, in dem analoge Signale, digital (z.B. im Bereich 0-10V) umgesetzt werden.

### A/D Filter

Hier kann der A/D Filter eingestellt werden.

### D/A Mode

Der Wertebereich gibt den Bereich an, in dem digitale Signale, analog (z.B. im Bereich 0-10V) umgesetzt werden.

### Counter Mode

Ist zuständig für den Counter Modus. Wahlweise kann mit 16-Bit hochgezählt werden, oder mit je 8-Bit hoch- und herunter gezählt werden.

### **Timeout-Mode**

Hier kann der Timeout-Mode eingestellt werden. Mehr Informationen zu den einzelnen Modi finden Sie im Kapitel: DapiSpecialCMDTimeout. Wird kein Timeout gewünscht kann dieser mit "Deactivate", deaktiviert werden.

### **Output Timeout**

Gibt die Zeit vor, nach der die Ausgänge abschalten, wenn ein Modul nicht mehr erreicht werden kann.

### **CNT48 Mode**

Stellt ein, welcher Counter Modus benutzt werden soll. Es stehen hierbei 6 verschiedene Modi zur Auswahl.

### **CNT48 Submode**

In Abhängigkeit von dem unter "CNT48 Mode" gewählten Modus, stehen hier entsprechende Submodi zur Auswahl.

### **CNT48 Filter**

Stellt den Filter ein, wie lange ein Signal mindestens sein muss, damit dieses als High erkannt wird. Es kann aus 16 vorgegebenen Filtern zwischen 20ns und 5ms ausgewählt werden.

### 3.2.5.1.11.3. CAN TX/RX Modi

Hier können Sie Einstellungen an den TX- und RX-Paketen vornehmen.

Hier können Sie Einstellungen an den TX-Paketen vornehmen.

### TX-Mode[1]

Hier können Sie Einstellungen der TX-Paket-Konfiguration ändern.

Activate	<input checked="" type="checkbox"/>
Trigger Mode	Interval
Interval	1 * 1ms
Address Mode	11 Bit Mode
Send to CAN-ID [hex]	200
Data to send	OPTO-IN 1-64

Werkseinstellung laden

Einstellung speichern

FormCfCANConfiaTX (Ver. 1.00)

#### Activate

Aktiviert diesen TX-Mode

#### Trigger Mode

Gibt an mit welchem Modus das senden stattfinden soll. Zur Auswahl stehen die Modi "Interval" , "RX-Event" und "Fast as possible".

### **Intervall**

Wird der Interval-Modus eingestellt, können sie zusätzlich angeben, in welchem Zeitintervall gesendet werden soll.

### **Address Mode**

Gibt den Bit-Mode an welcher verwendet werden soll. Ausgewählt werden kann zwischen 11-Bit Mode und 29-Bit Mode

### **Send to CAN-ID[hex]**

Sendet die CAN-Pakete an diese Adresse. In dem obigen Beispiel wird an Adresse 0x200 gesendet.

### **Data to send**

Hier können Sie angeben, welche Daten an die zuvor eingestellte Adresse gesendet werden sollen.



Hier können Sie Einstellungen an den RX-Paketen vornehmen.

**RX-Mode[1]**  
Hier können Sie Einstellungen der RX-Paket-Konfiguration ändern.

Activate

☒

Address Mode

11 Bit Mode

Receive at CAN-ID [hex]

100

Data to send

Digital Out 1-64

Werkseinstellung laden

Einstellung speichern

FormCfoCANConfioRX (Ver. 1.00)

### Activate

Aktiviert diesen RX-Mode

### Address Mode

Hier kann der Address Mode 11-Bit oder 29-Bit eingestellt werden.

### Receive at CAN-ID[hex]

Gibt die Empfängeradresse an. In den obigen Beispiel wird ein CAN-Paket auf der Adresse 0x100 empfangen.

### Data to send

Wird ein Paket an der eingestellten Adresse empfangen, wird der Inhalt des Datenpaketes an die digitalen Ausgänge 1-64 weitergeleitet, woraufhin dort die Ausgänge ein- oder ausgeschaltet werden.

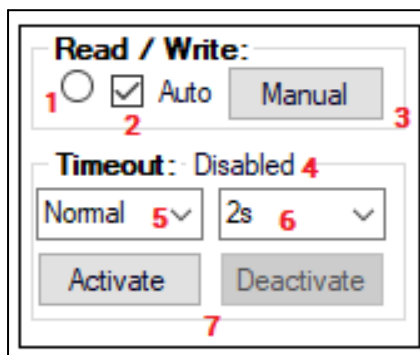
### 3.2.5.2. I/O-Test

Im I/O Bereich können Tests an den Modulen vorgenommen werden.

#### 3.2.5.2.1. Timeout Test-Funktion

Im "Read/Write" Bereich können Einstellungen am Timeout vorgenommen werden.

Mit diesen Funktionen lässt sich ein Timeout-Fall auslösen.



- 1 Das Feld zeigt durch wiederholtes Blinken an, ob eine Verbindung zum Modul besteht. Bleibt das Feld leer, ist die Kommunikation unterbrochen.
- 2 Durch das Entfernen des Häkchens, wird die Verbindung unterbrochen und somit ein Timeout-Fall ausgelöst. Ein erneutes Setzen des Hakens stellt die Verbindung wieder her.
- 3 Da die Benutzeroberfläche in einem Timeout-Fall nicht automatisch aktualisiert wird, kann dies mit einem Klick auf den "Manual" Knopf ausgelöst werden.
- 4 Hier wird der aktuelle Timeout-Status angezeigt.
- 5 Hier kann der gewünschte Timeout-Mode eingestellt werden. Mehr Informationen siehe Kapitel: **DapiSpecialCMDTimeout**.
- 6 Hier können Sie die Zeit einstellen, in welcher der Timeout ausgelöst werden soll.
- 7 "Activate" aktiviert den Timeout. Durch "Deactivate" wird er deaktiviert.

### 3.2.5.2.2. Digital In

Hier finden Sie Informationen zu den Digitalen Eingängen, sowie den Eingangszähler und den FlipFlop-Filter.

#### Digital In [0 .. 15]

Hier können Sie den Status der digitalen Eingangskanäle ablesen

	State on/off	Counter	FlipFlop
Kanal 0	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="checkbox"/>
Kanal 1	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="checkbox"/>
Kanal 2	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="checkbox"/>
Kanal 3	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="checkbox"/>
Kanal 4	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="checkbox"/>
Kanal 5	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="checkbox"/>
Kanal 6	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="checkbox"/>
Kanal 7	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="checkbox"/>

	State on/off	Counter	FlipFlop
Kanal 8	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="checkbox"/>
Kanal 9	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="checkbox"/>
Kanal 10	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="checkbox"/>
Kanal 11	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="checkbox"/>
Kanal 12	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="checkbox"/>
Kanal 13	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="checkbox"/>
Kanal 14	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="checkbox"/>
Kanal 15	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="checkbox"/>

☐ Read with reset

☐ Auto Refresh

FormIODigitalIn (Ver. 1.10)

#### State on/off

Zeigt den aktuellen Zustand der einzelnen Eingangskanäle.

#### Counter

Zeigt die Zählerstände der Eingangszähler an.

#### FlipFlop

Zeigt die Änderung der Eingangszustände seit dem letzten Auslesen an.

#### Read with reset

Mit dieser Option wird festgelegt, ob die Zähler beim nächsten Lesen zurückgesetzt werden sollen.

### 3.2.5.2.3. Digital Out

Hier können Sie die einzelnen digitalen Ausgänge Ihres Modules an- und ausschalten.

Eine LED an jedem Ausgangsrelais auf dem Board Ihres Modules, zeigt den aktuellen Status des Ausganges an (LED an = Relais an).

### Digital Out [0 .. 15]

Hier können Sie die digitalen Ausgänge des Modules ein- oder ausschalten

	On / Off	Readback	Timer		On / Off	Readback	Timer
Kanal 0	<input type="button" value="on"/> <input type="button" value="off"/>	<input type="checkbox"/>	0 s <input type="button" value="set"/>		Kanal 8	<input type="button" value="on"/> <input type="button" value="off"/>	0 s <input type="button" value="set"/>
Kanal 1	<input type="button" value="on"/> <input type="button" value="off"/>	<input type="checkbox"/>	0 s <input type="button" value="set"/>		Kanal 9	<input type="button" value="on"/> <input type="button" value="off"/>	0 s <input type="button" value="set"/>
Kanal 2	<input type="button" value="on"/> <input type="button" value="off"/>	<input type="checkbox"/>	0 s <input type="button" value="set"/>		Kanal 10	<input type="button" value="on"/> <input type="button" value="off"/>	0 s <input type="button" value="set"/>
Kanal 3	<input type="button" value="on"/> <input type="button" value="off"/>	<input type="checkbox"/>	0 s <input type="button" value="set"/>		Kanal 11	<input type="button" value="on"/> <input type="button" value="off"/>	0 s <input type="button" value="set"/>
Kanal 4	<input type="button" value="on"/> <input type="button" value="off"/>	<input type="checkbox"/>	0 s <input type="button" value="set"/>		Kanal 12	<input type="button" value="on"/> <input type="button" value="off"/>	0 s <input type="button" value="set"/>
Kanal 5	<input type="button" value="on"/> <input type="button" value="off"/>	<input type="checkbox"/>	0 s <input type="button" value="set"/>		Kanal 13	<input type="button" value="on"/> <input type="button" value="off"/>	0 s <input type="button" value="set"/>
Kanal 6	<input type="button" value="on"/> <input type="button" value="off"/>	<input type="checkbox"/>	0 s <input type="button" value="set"/>		Kanal 14	<input type="button" value="on"/> <input type="button" value="off"/>	0 s <input type="button" value="set"/>
Kanal 7	<input type="button" value="on"/> <input type="button" value="off"/>	<input type="checkbox"/>	0 s <input type="button" value="set"/>		Kanal 15	<input type="button" value="on"/> <input type="button" value="off"/>	0 s <input type="button" value="set"/>

☒ Invert DO-Timer  
on (data1) / off (data0)

☐ Auto Refresh

FormIODigitalOut (Ver. 1.10)

#### On/Off

Schaltet das jeweilige Ausgangsrelais an oder aus.

#### Readback

Zeigt den aktuellen Status des jeweiligen Relais an (on oder off).

#### Switch all states OFF / Switch all states ON

Mit diesen Knöpfen, lassen sich alle Ausgänge des Modules gleichzeitig an- oder ausschalten.

### **Kanäle schalten mit Timer-Funktion**

(Wird nur angezeigt, wenn es vom Modul unterstützt wird)

Geben Sie im Timer Bereich eine Zeit (in Sekunden) an, nach der die Relais ein oder ausgeschaltet werden sollen. Mit "set" starten Sie den Timer.

### **Invert DO-Timer**

Ist diese Option aktiviert, wird das Relais nach Ablauf des Timers deaktiviert. Ist diese Option deaktiviert, wird das Relais nach Ablauf des Timers aktiviert.

### 3.2.5.2.4. Analog In

Hier können Sie Einstellungen am A/D-Mode ändern und testen.

### Analog In [0 .. 15]

Hier können Sie den A/D - Mode der Kanäle ändern

	Value		Value
Kanal 0	<input type="text" value="0,027"/> V	Kanal 8	<input type="text" value="0,027"/> V
Kanal 1	<input type="text" value="0,026"/> V	Kanal 9	<input type="text" value="0,025"/> V
Kanal 2	<input type="text" value="0,025"/> V	Kanal 10	<input type="text" value="0,024"/> V
Kanal 3	<input type="text" value="0,025"/> V	Kanal 11	<input type="text" value="0,025"/> V
Kanal 4	<input type="text" value="0,025"/> V	Kanal 12	<input type="text" value="0,024"/> V
Kanal 5	<input type="text" value="0,025"/> V	Kanal 13	<input type="text" value="0,024"/> V
Kanal 6	<input type="text" value="0,025"/> V	Kanal 14	<input type="text" value="0,025"/> V
Kanal 7	<input type="text" value="0,025"/> V	Kanal 15	<input type="text" value="0,024"/> V

Set mode for all channels

Mode Readback

Set filter for all channels

Filter Readback

☒ Auto Refresh

FormIOAnalogIn (Ver. 1.10)

#### Value

Liest den wert an dem jeweiligen A/D-Kanals aus.

#### Set mode for all channels

Auswahl des Spannungs- /Strombereichs für alle Kanäle, in dem gemessen werden soll. Es werden nur Modi angezeigt, die von Ihrem Modul unterstützt werden.

#### Mode Readback

Liest den aktuell verwendeten A/D-Mode aus dem Modul.

#### Set filter for all channels

Auswahl des anzuwendenden A/D-Filters für alle Kanäle.

#### Filter Readback

Liest den aktuell verwendeten A/D-Filter aus dem Modul.

### 3.2.5.2.5. Analog Out

In diesem Bereich können Sie Einstellungen an den D/A-Kanälen Ihres Moduls vornehmen und testen.

### Analog Out [0 .. 13]

Hier können Sie den D/A - Mode der Kanäle ändern

	Value	Mode	Readback
Kanal 0	<input type="text" value="0"/>	16 Bit / 0-10V ▾	<input type="text" value="0"/> V
Kanal 1	<input type="text" value="1"/>	16 Bit / 0-10V ▾	<input type="text" value="0,999908"/> V
Kanal 2	<input type="text" value="2"/>	16 Bit / 0-10V ▾	<input type="text" value="1,999969"/> V
Kanal 3	<input type="text" value="3"/>	16 Bit / 0-10V ▾	<input type="text" value="2,999878"/> V
Kanal 4	<input type="text" value="4"/>	16 Bit / 0-10V ▾	<input type="text" value="3,999939"/> V
Kanal 5	<input type="text" value="5"/>	16 Bit / 0-10V ▾	<input type="text" value="5"/> V
Kanal 6	<input type="text" value="6"/>	16 Bit / 0-10V ▾	<input type="text" value="5,999908"/> V
Kanal 7	<input type="text" value="7"/>	16 Bit / 0-10V ▾	<input type="text" value="6,999969"/> V
Kanal 8	<input type="text" value="8"/>	16 Bit / 0-10V ▾	<input type="text" value="7,999878"/> V
Kanal 9	<input type="text" value="9"/>	16 Bit / 0-10V ▾	<input type="text" value="8,999939"/> V
Kanal 10	<input type="text" value="10"/>	16 Bit / 0-10V ▾	<input type="text" value="9,999847"/> V
Kanal 11	<input type="text" value="11"/>	16 Bit / 0-10V ▾	<input type="text" value="9,999847"/> V
Kanal 12	<input type="text" value="12"/>	16 Bit / 0-10V ▾	<input type="text" value="9,999847"/> V
Kanal 13	<input type="text" value="13"/>	16 Bit / 0-10V ▾	<input type="text" value="9,999847"/> V

Set mode for all channels

16 Bit / 0-10V ▾

Set

☒ Auto Refresh

FormIOAnalogOut (Ver. 1.10)

#### Value

Hier kann der Wert eingetragen werden, der an dem jeweiligen D/A-Kanal ausgegeben werden soll.

#### Mode

Auswahl des Spannungs-/ Strombereichs in dem der Wert des jeweiligen Kanals ausgegeben werden soll. Es können nur Modi ausgewählt werden, die vom Modul unterstützt werden.

#### Readback

Liest den Ist-Wert des jeweiligen D/A-Kanals zurück.

### **Set mode for all channels**

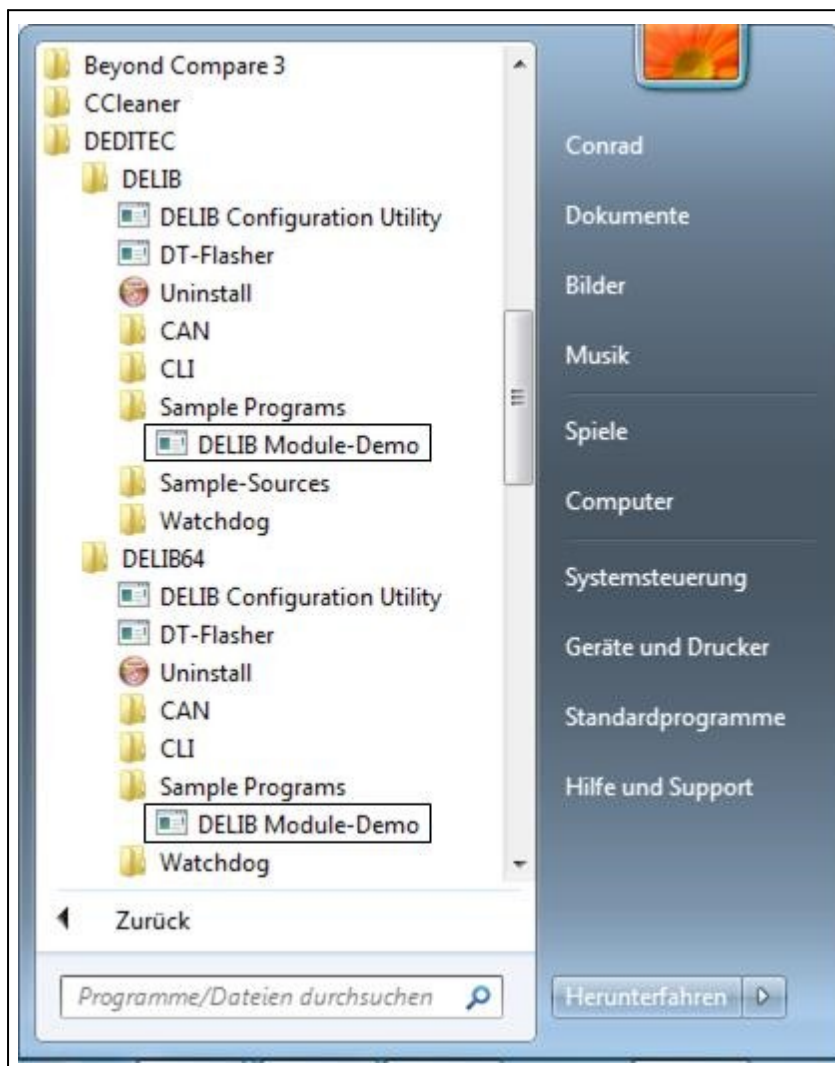
Mit dieser Funktion lässt sich der Spannungs-/ Strombereich für alle verfügbaren Kanäle auf einmal ändern.



### 3.2.6. DELIB Module Demo

Nach Installation der DELIB Treiberbibliothek kann das Programm DELIB Module Demo auf folgendem Weg gestartet werden:

Start → Programme → DEDITEC → DELIB oder DELIB64 → Sample Programs → DELIB Module Demo.

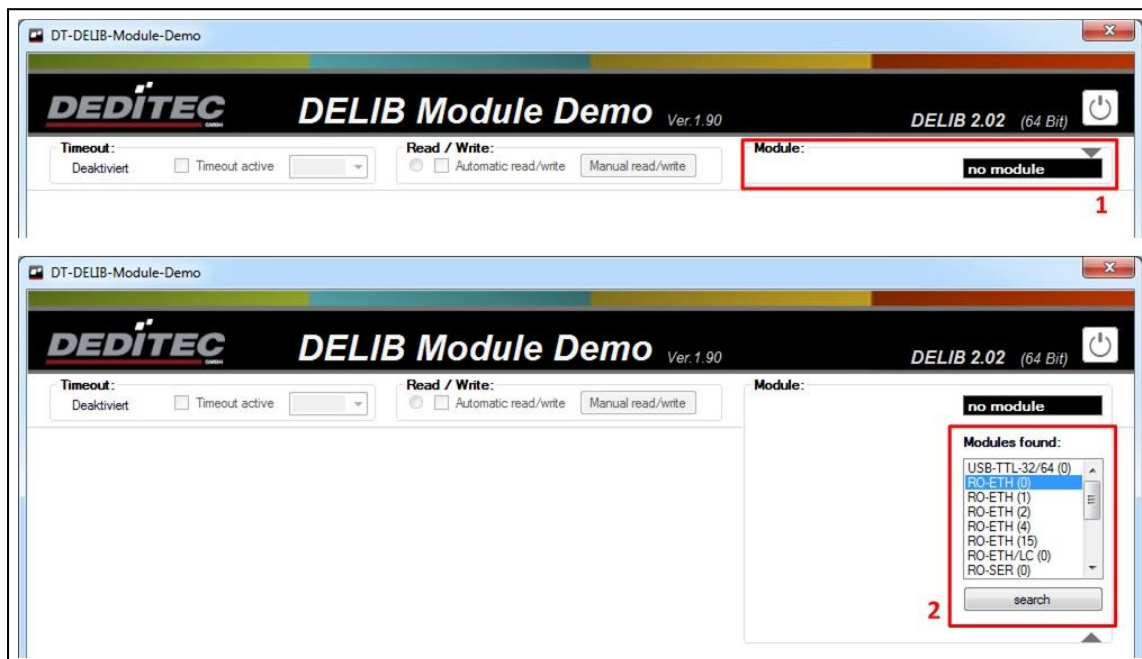


Das Programm DELIB Module Demo ist ein All-in-One Tool mit dem sämtliche I/Os aller Produkte aus unserem S&R Bereich gesteuert und getestet werden können.

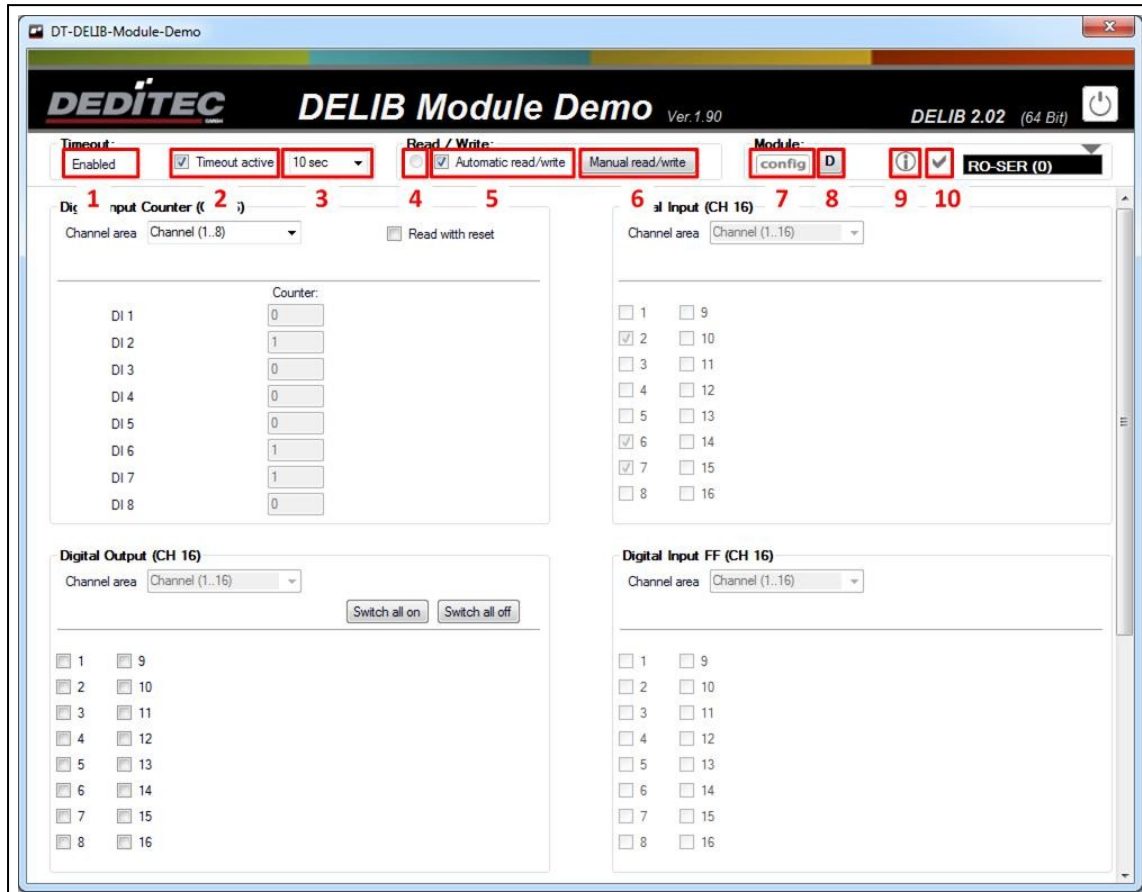
### 3.2.6.1. Auswahl des Moduls

Bei Programmstart muss ein Modul ausgewählt werden.

1. Klicken Sie den "Module-Selector" an. Sie erhalten eine Auflistung der verfügbaren/angeschlossenen Module.
2. Wählen Sie nun das gewünschte Modul aus.



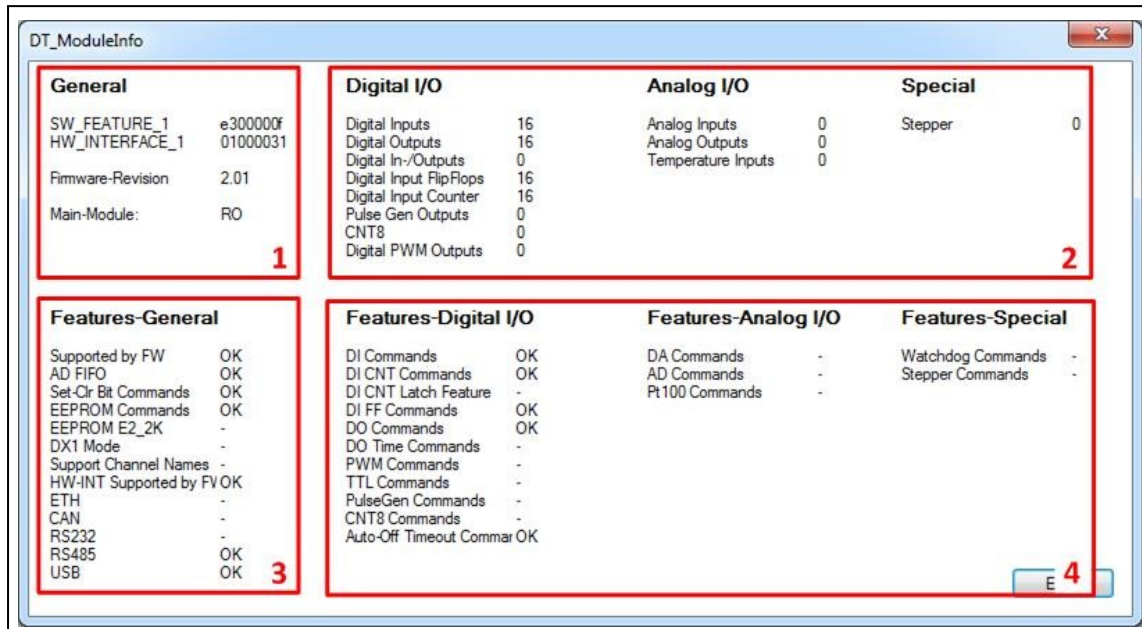
### 3.2.6.2. Allgemein



1. Timeout-Status ("Disabled", "Enabled" oder "Occured").
2. Aktiviert oder deaktiviert den Timeout-Schutz.
3. Timeout-Zeit für den Timeout-Schutz.
4. Status für "Automatic read/write" (Blinkt, wenn "Automatic read/write" aktiviert ist).
5. Mit dem Haken bei "Automatic read/write" wird festgelegt, ob die Messdaten automatisch gelesen/geschrieben werden sollen.

6. Manuelles Lesen/Schreiben. Nur aktiv, wenn "Automatic read/write" deaktiviert ist.
7. Hier kann das aktuell ausgewählte Modul via DELIB Configuration Utility konfiguriert werden.
8. Sofern vom Modul unterstützt, erhalten Sie hierüber detaillierte Debug-Informationen.

### 3.2.6.2.1. Module Info



Dieses Beispiel zeigt die erweiterten Informationen des Moduls RO-SER-016-M16.

1. Allgemeine Informationen des ausgewählten Moduls.
2. Anzahl der angeschlossenen I/O-Kanäle.
3. Übersicht der unterstützten Interface DELIB Kommandos.
4. Übersicht der unterstützten I/O DELIB Kommandos.

### 3.2.6.3. Digital Input

The screenshot shows the 'DT-DELIB-Module-Demo' window. The title bar includes 'DEDITEC', 'DELIB Module Demo', 'Ver. 1.90', 'DELIB 2.02 (64 Bit)', and a power icon. The main interface has several sections:

- Timeout:** Not Supported, Timeout active (10 sec).
- Read / Write:** Automatic read/write (checked), Manual read/write.
- Module:** config, D, RO-SER (0).

**Digital Input Counter (CH 16)**

Channel area: Channel (1..8) **1**

☐ Read with reset **2**

DI	Counter
DI 1	1
DI 2	2
DI 3	1
DI 4	0
DI 5	0
DI 6	2
DI 7	1
DI 8	0

**3**

**Digital Input (CH 16)**

Channel area: Channel (1..16)

<input checked="" type="checkbox"/> 1	<input type="checkbox"/> 9
<input checked="" type="checkbox"/> 2	<input type="checkbox"/> 10
<input checked="" type="checkbox"/> 3	<input type="checkbox"/> 11
<input type="checkbox"/> 4	<input type="checkbox"/> 12
<input type="checkbox"/> 5	<input type="checkbox"/> 13
<input checked="" type="checkbox"/> 6	<input type="checkbox"/> 14
<input type="checkbox"/> 7	<input type="checkbox"/> 15
<input type="checkbox"/> 8	<input type="checkbox"/> 16

**4**

**Digital Input FF (CH 16)**

Channel area: Channel (1..16)

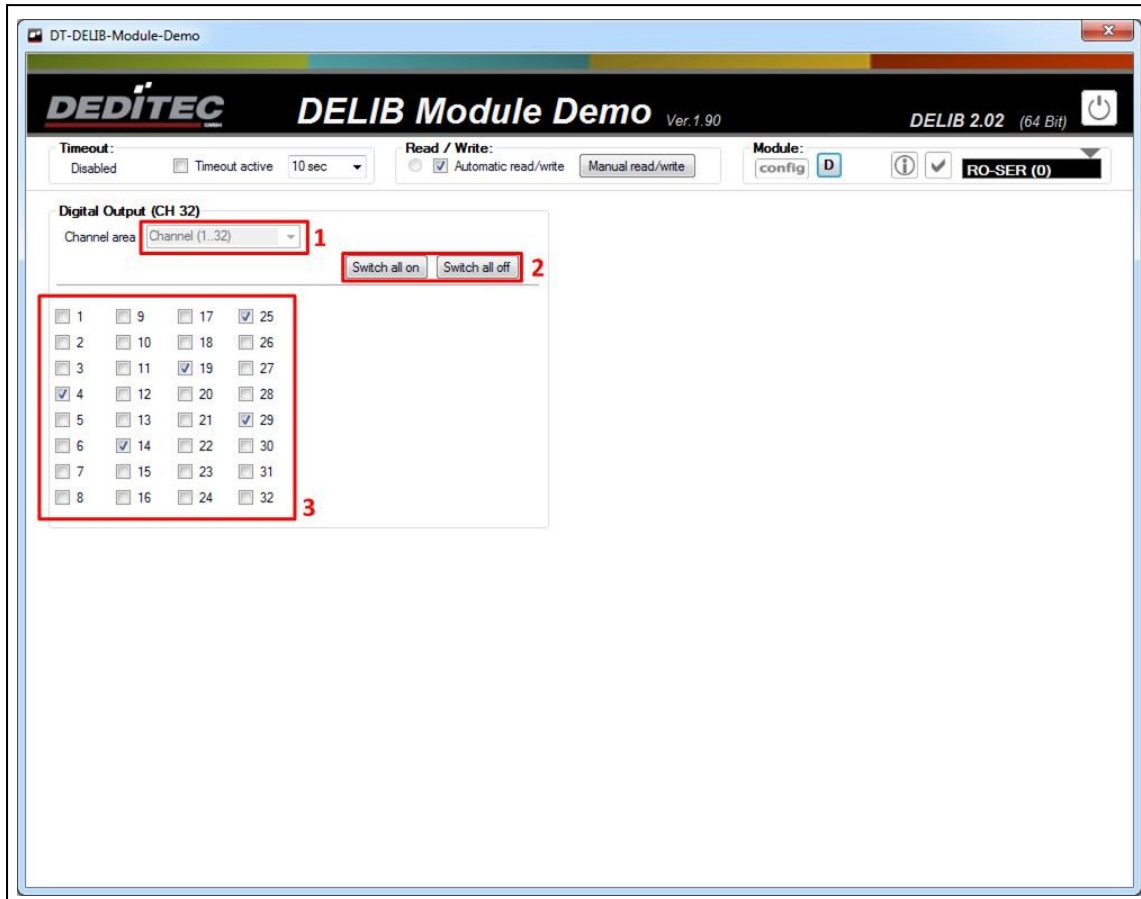
<input type="checkbox"/> 1	<input type="checkbox"/> 9
<input type="checkbox"/> 2	<input type="checkbox"/> 10
<input checked="" type="checkbox"/> 3	<input type="checkbox"/> 11
<input type="checkbox"/> 4	<input type="checkbox"/> 12
<input type="checkbox"/> 5	<input type="checkbox"/> 13
<input type="checkbox"/> 6	<input type="checkbox"/> 14
<input type="checkbox"/> 7	<input type="checkbox"/> 15
<input type="checkbox"/> 8	<input type="checkbox"/> 16

**5**

Dieses Beispiel zeigt die digitalen Eingänge eines RO-SER-016 Moduls.

1. Auswahl des Kanal-Bereichs, der angezeigt werden soll.
2. Mit dem Haken bei "Read with reset" wird festgelegt, ob die Zähler beim nächsten Lesen resettet werden.
3. Zählerstände der Eingangszähler.
4. Zustände der Eingänge.
5. Änderung der Eingangszustände (seit dem letzten Auslesen).

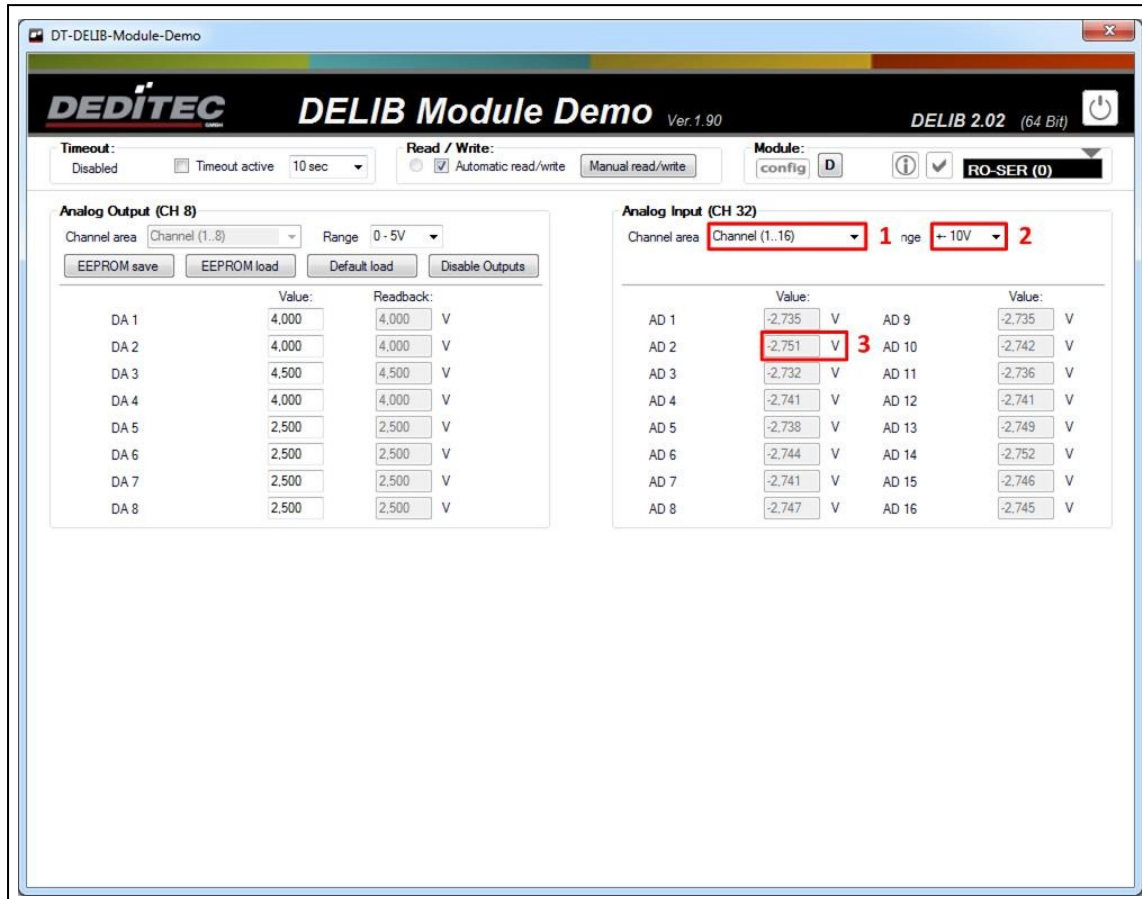
### 3.2.6.4. Digital Output



Dieses Beispiel zeigt die digitalen Ausgänge eines RO-SER-M32 Moduls.

1. Auswahl des Kanal-Bereichs, der angezeigt werden soll.
2. Hiermit werden alle Ausgänge des aktuellen Kanal-Bereichs ein- bzw. ausgeschaltet.
3. Hier können bestimmte Ausgänge gezielt ein- bzw. ausgeschaltet werden.

### 3.2.6.5. Analog Input

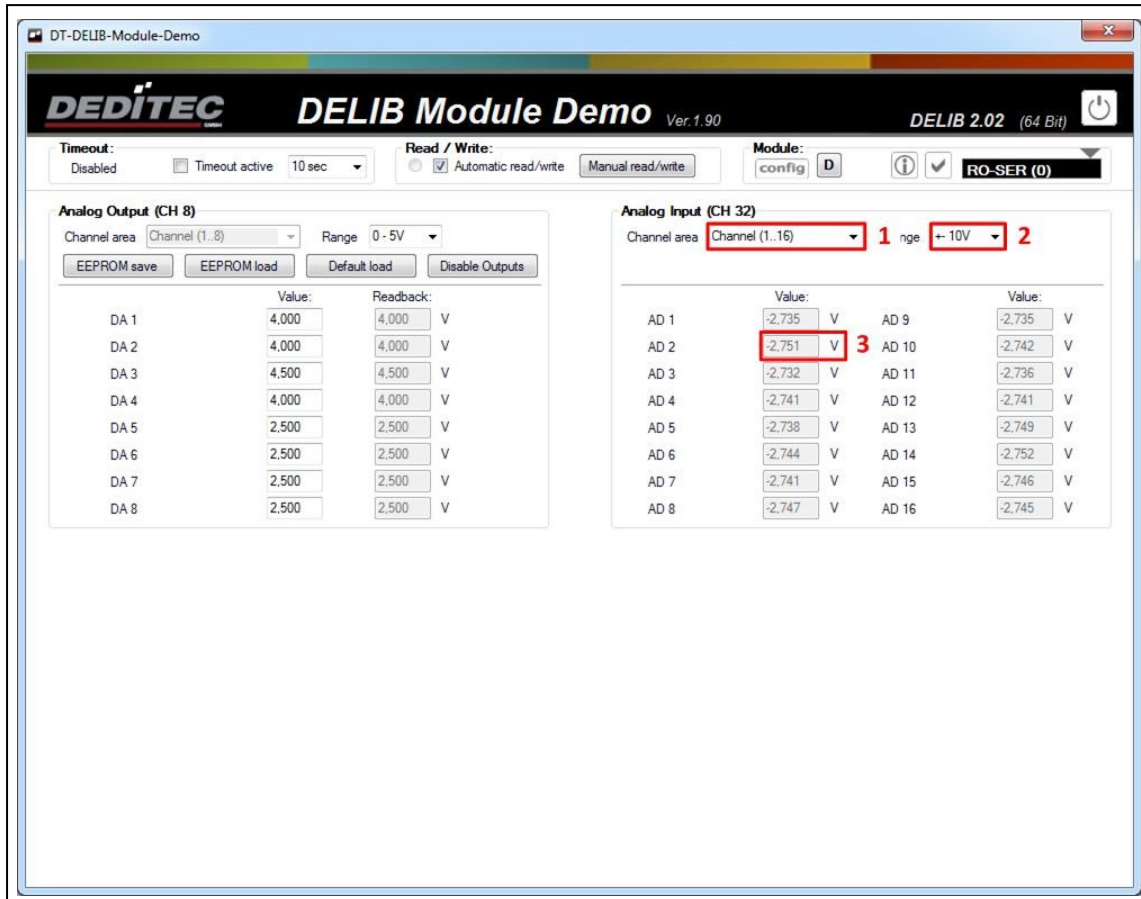


Dieses Beispiel zeigt die analogen Eingänge eines RO-SER-AD32-DA8 Moduls.

1. Auswahl des Kanal-Bereichs, der angezeigt werden soll.
2. Auswahl des Spannungs-/Strombereichs in dem gemessen werden sollen. Wird der Modus nicht unterstützt, erscheint die Meldung "illegal" rechts neben dem Drop-Down Menü.
3. Aktueller A/D-Wert an A/D-Kanal 2.



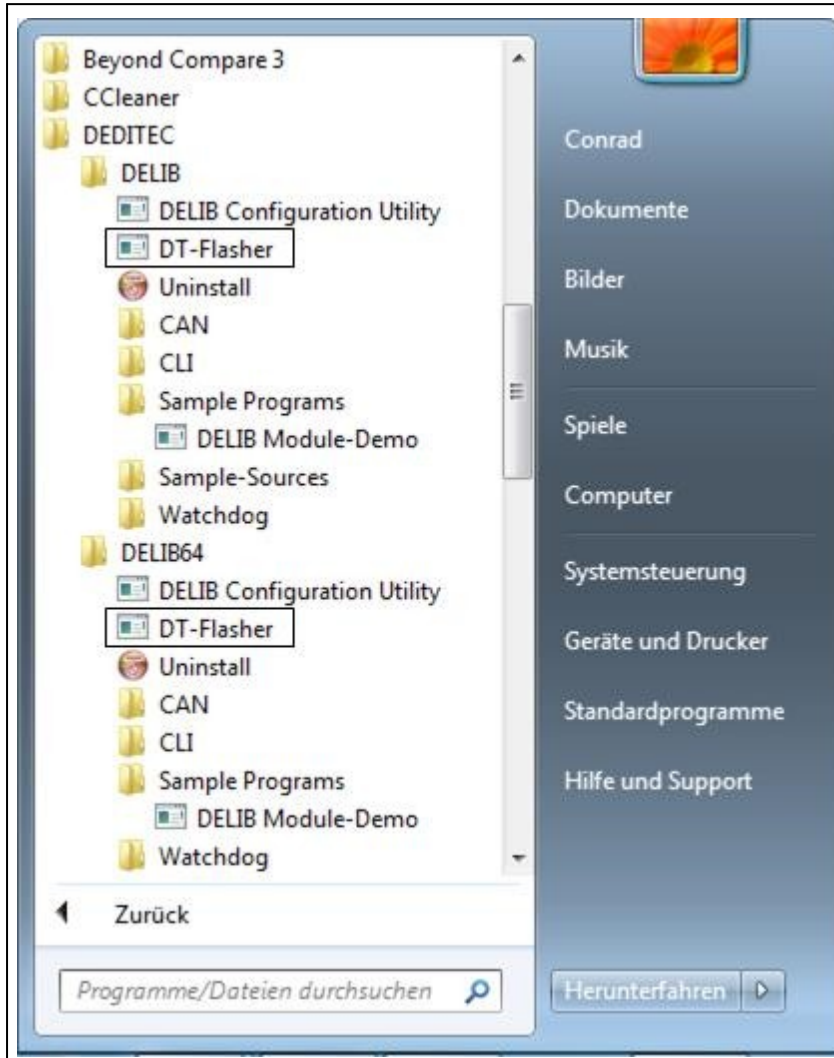
### 3.2.6.6. Analog Output



Dieses Beispiel zeigt die analogen Eingänge eines RO-SER-AD32-DA8 Moduls.

1. Auswahl des Kanal-Bereichs, der angezeigt werden soll.
2. Auswahl des Spannungs-/Strombereichs in dem gemessen werden sollen. Wird der Modus nicht unterstützt, erscheint die Meldung "illegal" rechts neben dem Drop-Down Menü.
3. Aktueller A/D-Wert an A/D-Kanal 2.

### 3.2.7. DT-Flasher



Nach Installation der DELIB Treiberbibliothek kann das Programm DT-Flasher auf folgendem Weg gestartet werden:

Start → Programme → DEDITEC → DELIB oder DELIB64 → DT-Flasher.

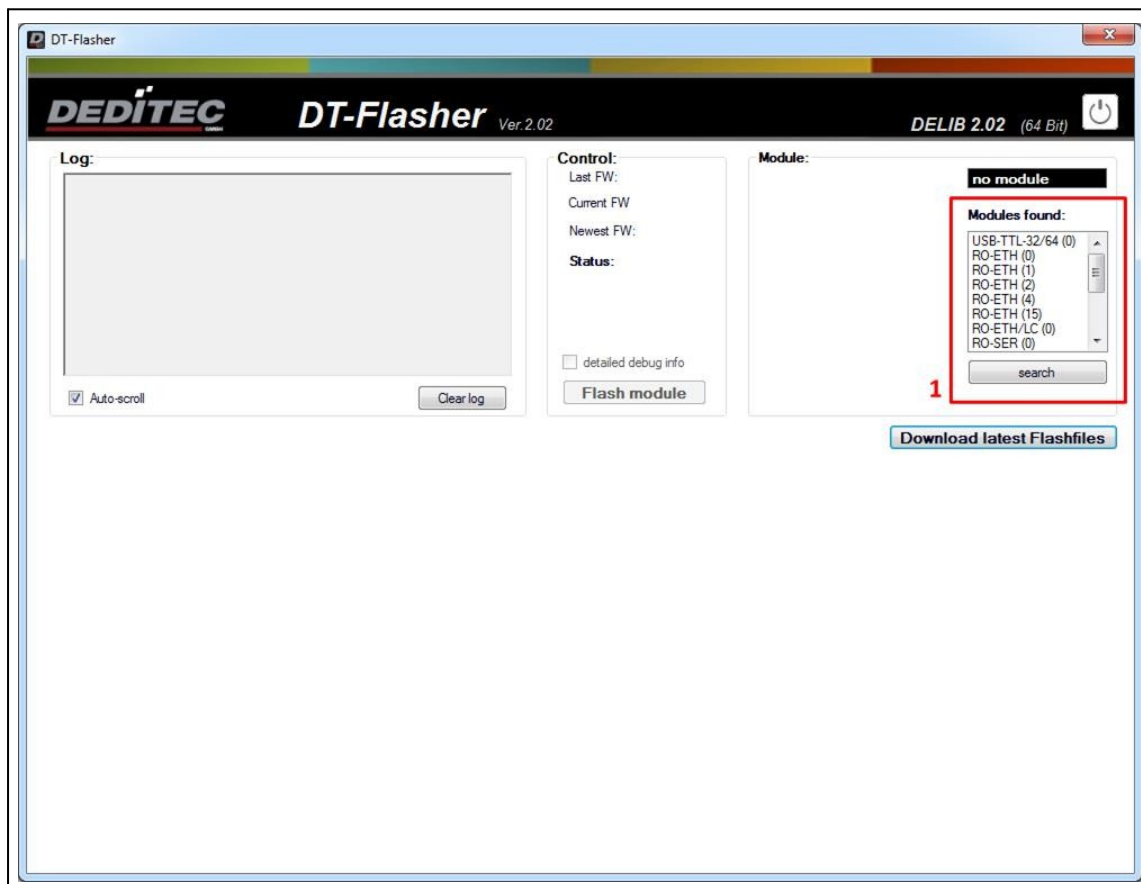
### 3.2.7.1. Über DEDITEC-Firmware

Die meisten DEDITEC Produkte verfügen über einen eigenen Microcontroller. Dieser Prozessor ist für die Steuerung aller Abläufe der Hardware verantwortlich. Um die für den Prozessor benötigte Firmware im Nachhinein zu ändern, stellen wir unser kostenloses Tool DT-Flasher zur Verfügung. Mit diesem Tool hat der Kunde die Möglichkeit neu veröffentlichte Firmware-Versionen, direkt bei sich vor Ort auf das Modul zu übertragen.

#### Hinweis:

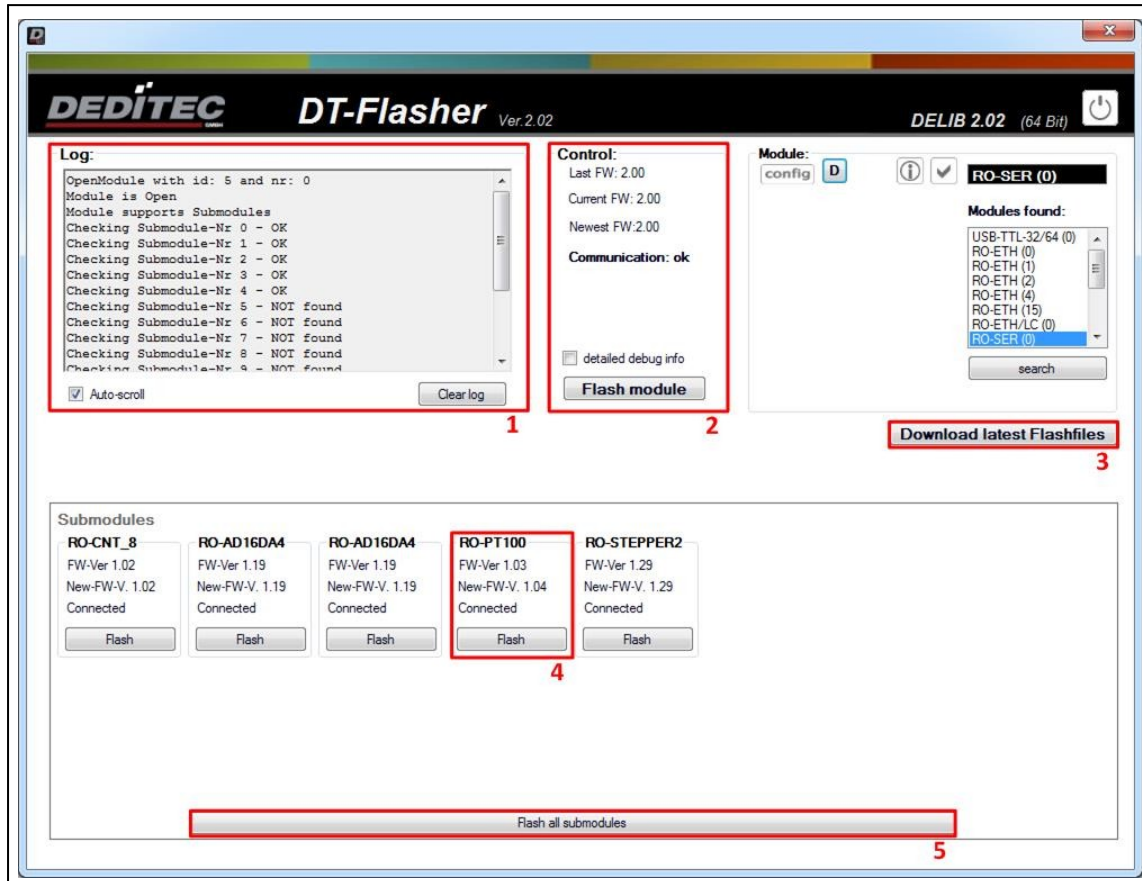
Da neue Firmware-Versionen in der Regel neue Funktionen für Ihr Produkt "freischalten", empfehlen wir daher ein regelmäßiges Firmware-Update Ihrer DEDITEC Produkte.

### 3.2.7.2. Auswahl des Moduls



1. Wählen Sie bei Programmstart das Modul aus, welches Sie mit einer neuen Firmware updaten möchten. Hierzu finden Sie eine Auflistung aller verfügbaren Module im "Module-Selector"

### 3.2.7.3. Firmware Update durchführen



Dieses Beispiel zeigt das Modul RO-SER-CNT8-AD32-DA8-PT100-4-STEPPER2 vor einem Firmware-Update.

1. Logbuch - Alle Meldungen während des Firmware-Updates werden hier angezeigt. Über Auto-scroll wird festgelegt, ob immer automatisch bis zum letzten Ereignis heruntergescrollt werden soll. Über Clear log wird das gesamte Logbuch gelöscht.

2. Hier erhalten Sie Informationen zum Interface-Modul (in diesem Beispiel das RO-SER-Interface). Newest FW zeigt die neuste Firmware-Version an, die für das Modul verfügbar ist. Current FW zeigt die Version an, die aktuell auf dem Modul vorhanden ist. Nachdem das Modul erfolgreich geflasht wurde, zeigt Last FW die Version an, die vor dem Firmware-Update aufgespielt war. Ist der Haken bei detailed debug info gesetzt, werden während des Firmware Updates detaillierte Meldungen ins Logbuch(1) geschrieben. Mit Flash module wird das Firmware Update für das Interface-Modul gestartet.

3. Hierüber können Sie direkt aus der Anwendung heraus die aktuellsten Firmware Versionen, sogenannte Flash-Files, herunterladen.

4. Firmware Version zeigt die aktuelle Firmware Version des Submoduls. New-FW-Ver zeigt die neuste Version an, die für dieses Submodul verfügbar ist. Über den Button Flash wird das Firmware Update für das jeweilige Submodul durchgeführt.

5. Über den Button Flash all submodules wird das Firmware Update für alle angeschlossenen Submodule durchgeführt.

#### **3.2.7.3.1. Flash-Files manuell aktualisieren**

In manchen Fällen ist es nötig, die Flash-Files manuell zu aktualisieren, z.B. wenn am PC keine Administratoren-Rechte verfügbar sind.

##### **Schritt 1**

Downloaden Sie die aktuellste Version der Flash-Files unter

[http://www.deditec.de/zip/deditec-flash\\_files.zip](http://www.deditec.de/zip/deditec-flash_files.zip)

##### **Schritt 2**

Entpacken Sie das heruntergeladene ZIP-Archiv, je nach DELIB Installation, in folgendes Verzeichnis:

x86

C:\Program Files(x86)\DEDITEC\DELIB\programs\

x64

C:\Program Files\DEDITEC\DELIB\programs

### 3.3. DELIB Sample Sources (Windows Programmbeispiele)

Die DELIB Sample Sources bieten Beispielprogramme inklusive Quellcode zu nahezu allen DEDITEC Produkten.

Um den Schnelleinstieg mit unseren Modulen zu vereinfachen, finden Sie Quellcodes zu folgenden Programmiersprachen:

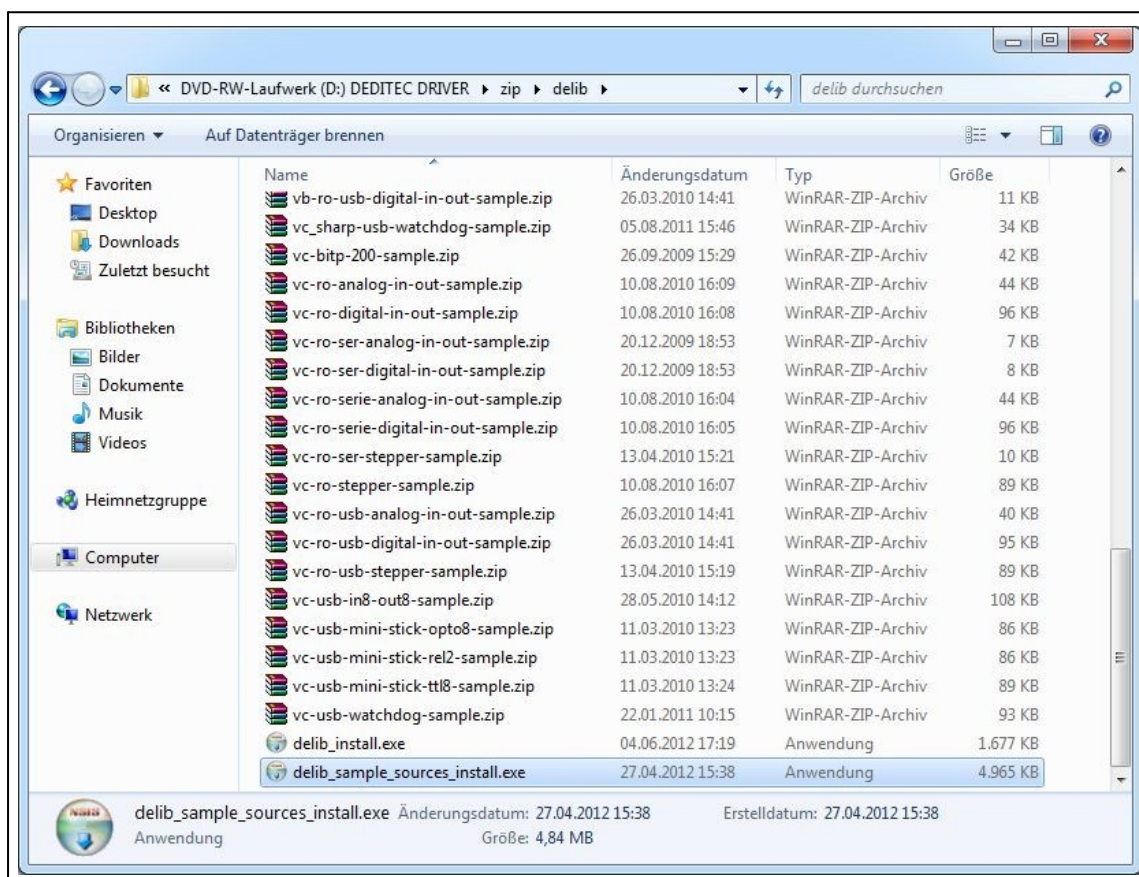
- C
- C++
- C#
- Delphi
- VisualBasic
- VB.NET
- MS-Office
- LabVIEW
- Java

### 3.3.1. Installation DELIB Sample Sources

Die DELIB Sample Sources können entweder während der Durchführung des DELIB Setups installiert werden oder als eigenständiges Setup.

Legen Sie die DEDITEC Driver-CD in das Laufwerk und starten Sie `delib_sample_sources_install.exe`.

Eine aktuelle Version der Sample Sources finden Sie auch im Internet unter <http://www.deditec.de/de/delib>

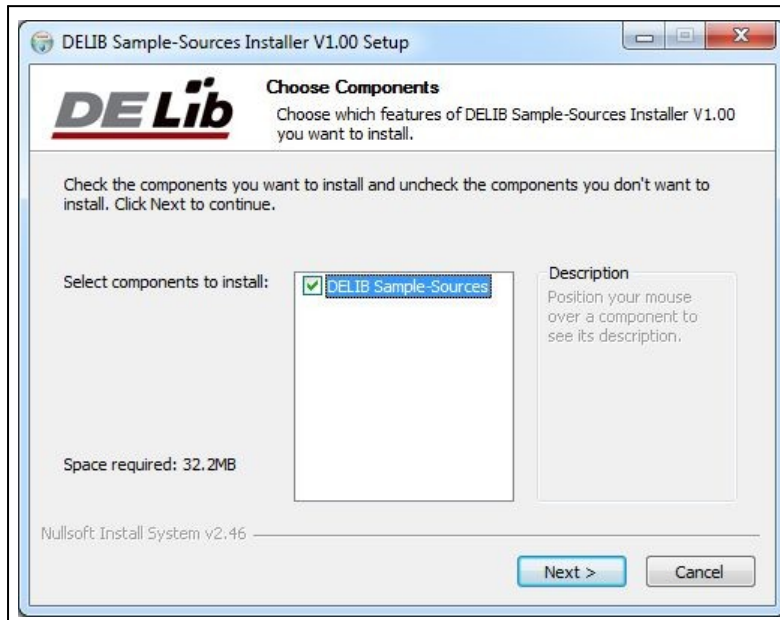




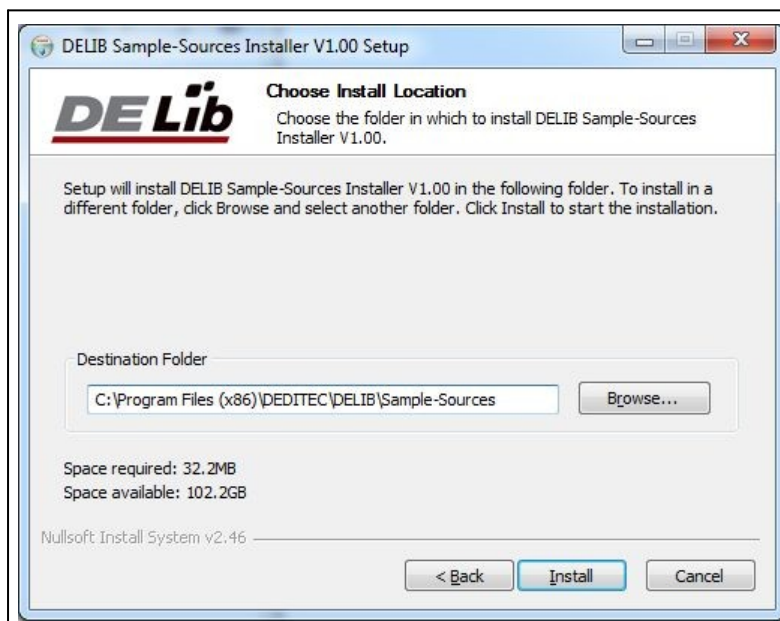
Startbild des DELIB Sample Sources Installer



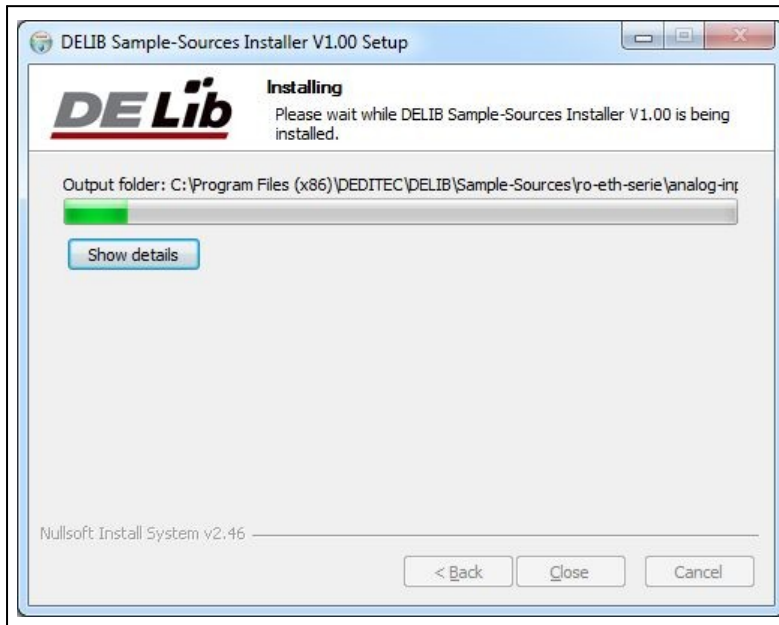
Drücken Sie Next.



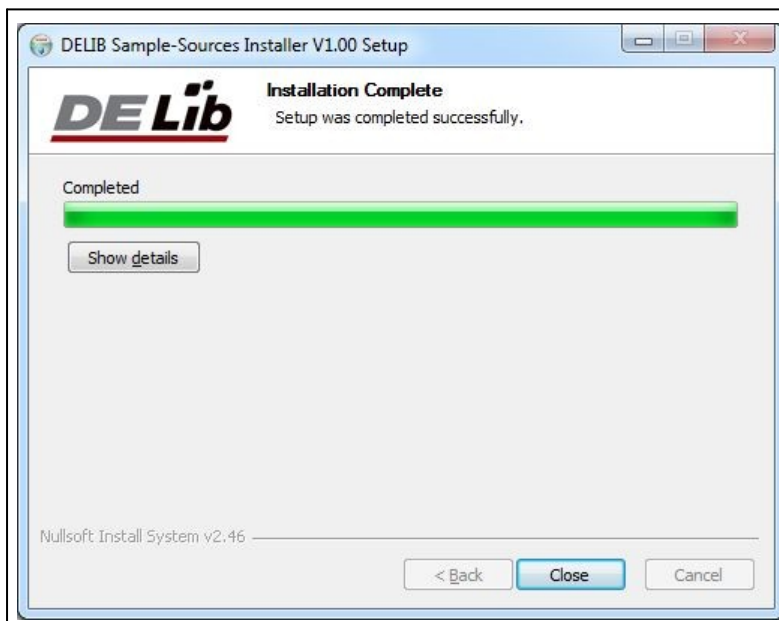
Wählen Sie den Installationsordner und drücken Sie Install.



Die DELIB Sample Sources werden nun installiert.



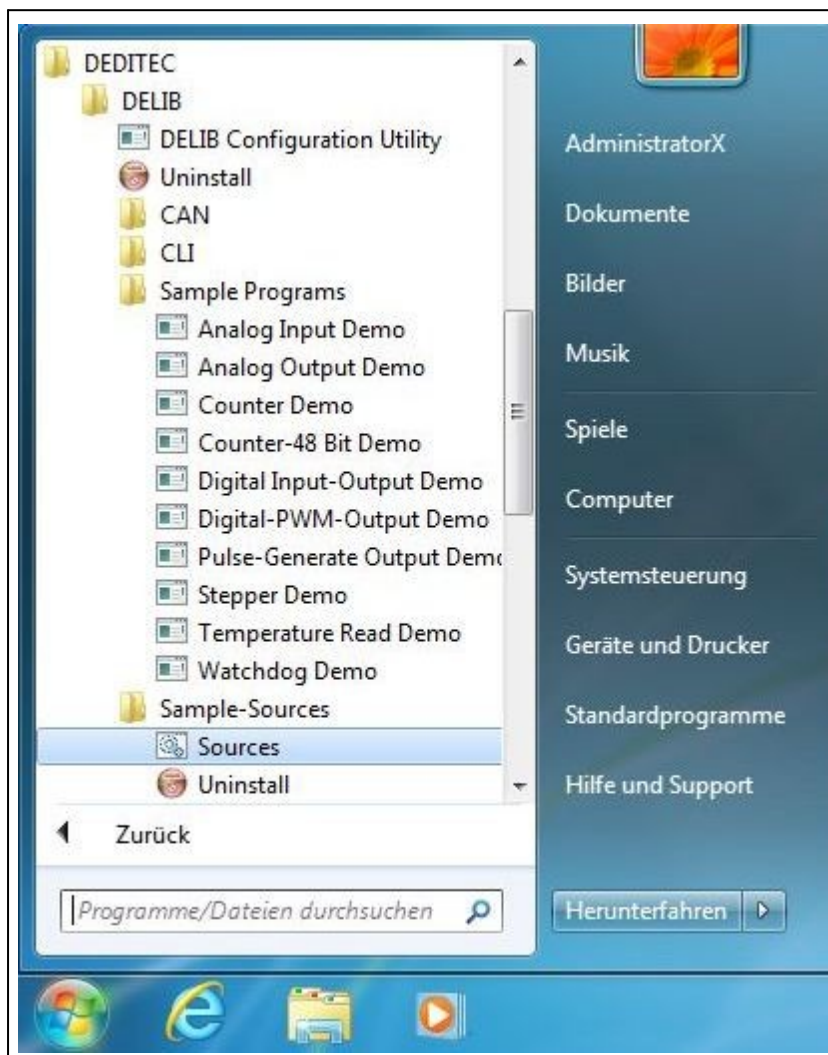
Die DELIB Sample Sources wurden erfolgreich installiert. Drücken Sie Close um die Installation zu beenden.



### 3.3.2. Benutzung der DELIB Sample Sources

Nach Installation der DELIB Sample Sources finden Sie diese unter

Start → Programme → DEDITEC → DELIB → Sample-Sources → Sources

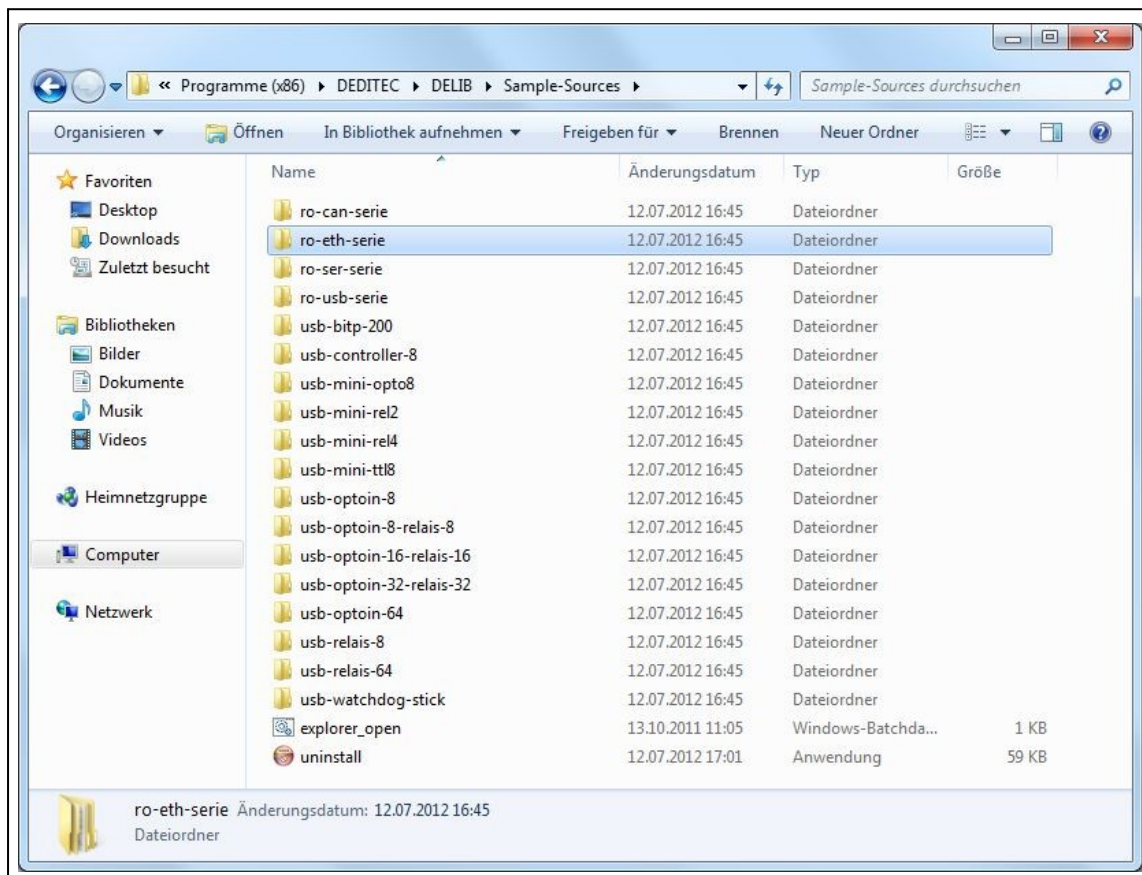


Nun öffnet sich der Windows-Explorer mit einer Übersicht aller Produkte für die ein Beispielprogramm verfügbar ist.

### 3.3.2.1. Schritt 1 - Produktauswahl

Sie benötigen beispielsweise eine Hilfestellung zur Programmierung der digitalen Eingänge eines RO-ETH-Moduls (z.B RO-ETH-016) in der Programmiersprache Visual-C.

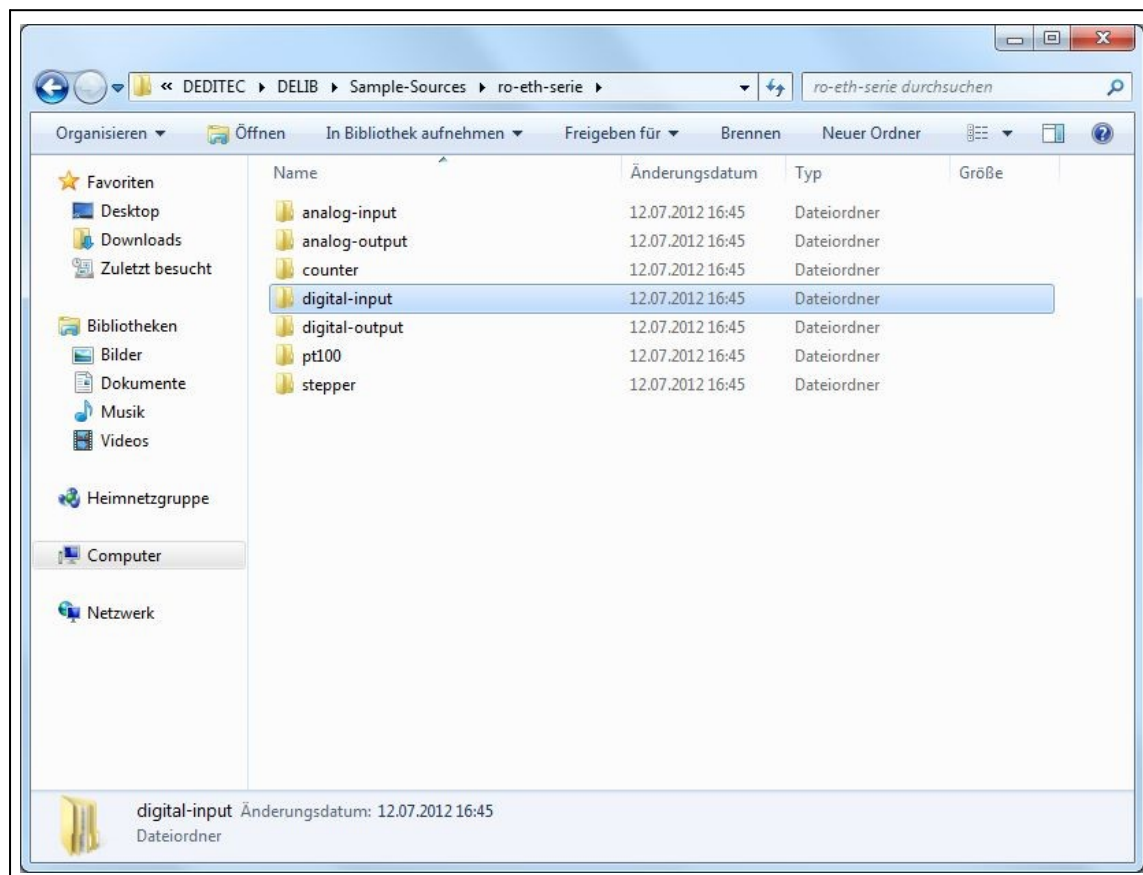
Da es sich um ein RO-ETH-Produkt handelt, wählen bzw. öffnen Sie den Ordner ro-eth-serie.



### 3.3.2.2. Schritt 2 - Kategorieauswahl

Im nächsten Schritt, finden Sie eine Übersicht der verfügbaren Kategorien für das ausgewählte Produkt.

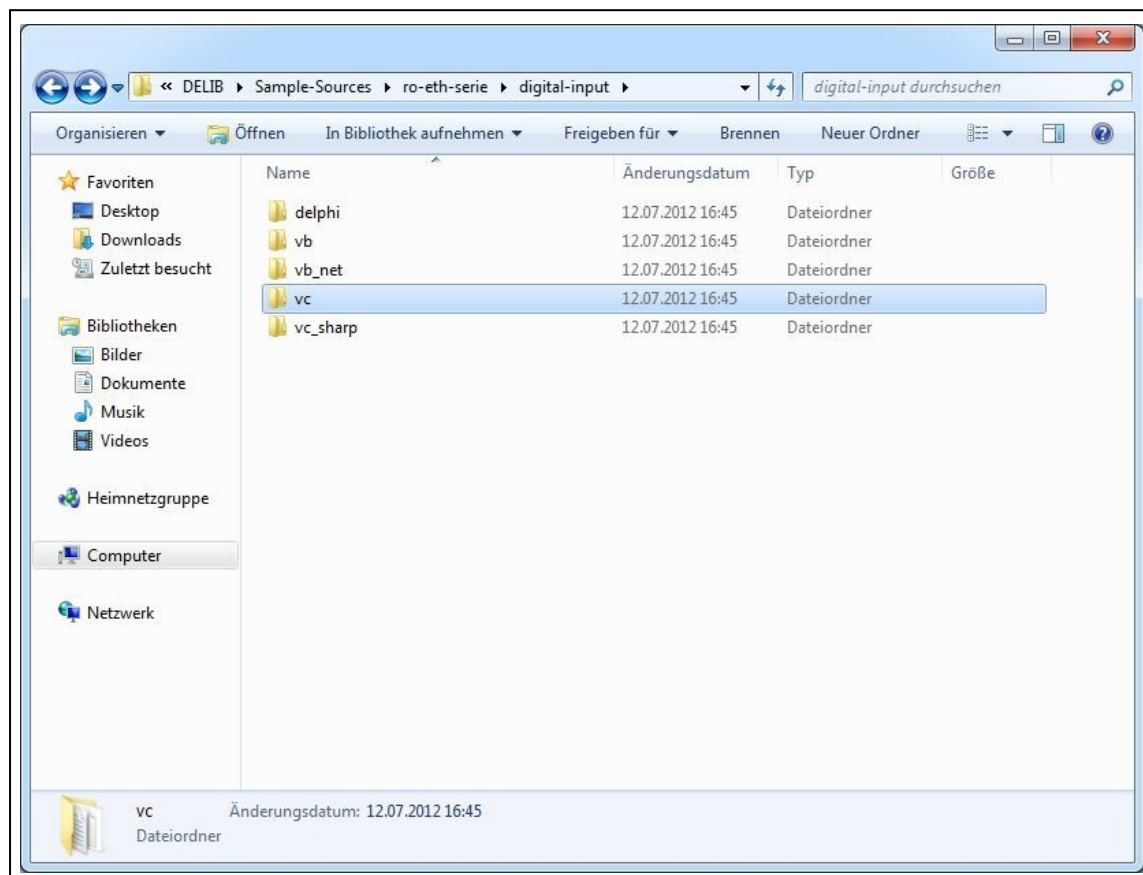
Da wir uns in diesem Beispiel auf die digitalen Eingänge konzentrieren, wählen Sie die Kategorie digital-input



### 3.3.2.3. Schritt 3 - Programmiersprachenauswahl

In diesem Schritt sehen Sie alle verfügbaren Programmierbeispiele der gewählten Kategorie, sortiert nach Programmiersprachen.

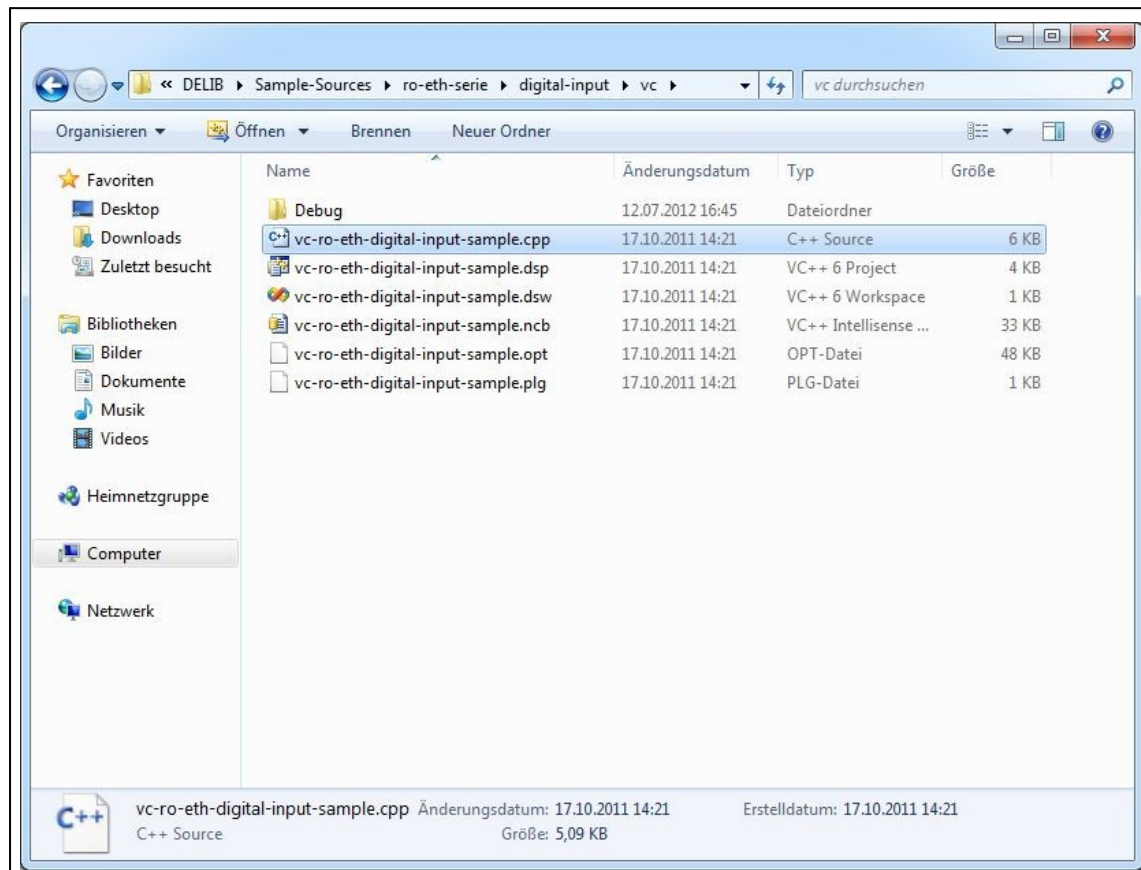
Da wir uns in diesem Beispiel auf die Programmiersprache Visual-C konzentrieren, öffnen Sie den Ordner vc.





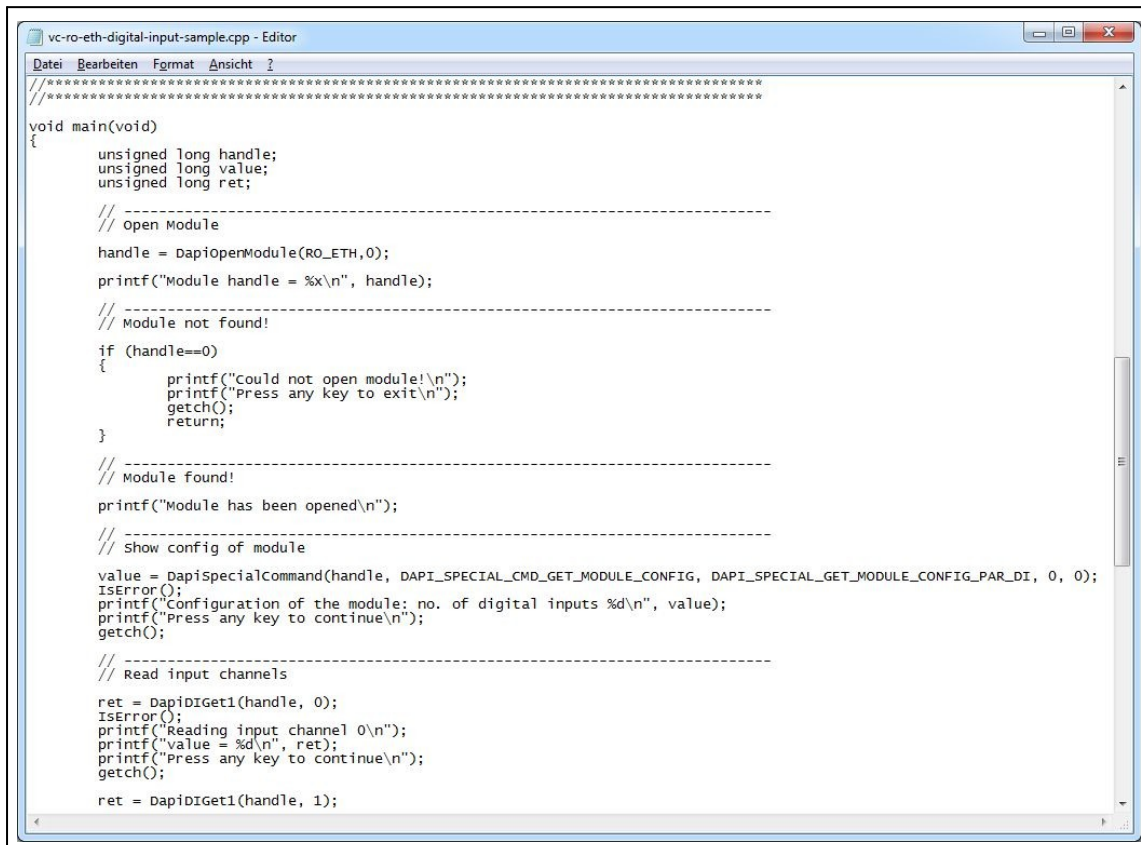
### 3.3.2.4. Schritt 4 - Quellcode

Nach Auswahl der Programmiersprache erhalten Sie folgende Übersicht:





Den Quellcode des Beispielprogramms (in diesem Fall .cpp-Datei) können Sie nun mit einem beliebigen Text-Editor öffnen.



```
vc-ro-eth-digital-input-sample.cpp - Editor
Datei Bearbeiten Format Ansicht ?
//*****
//*****
void main(void)
{
    unsigned long handle;
    unsigned long value;
    unsigned long ret;

    // -----
    // Open Module

    handle = DapiOpenModule(RO_ETH,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;
    }

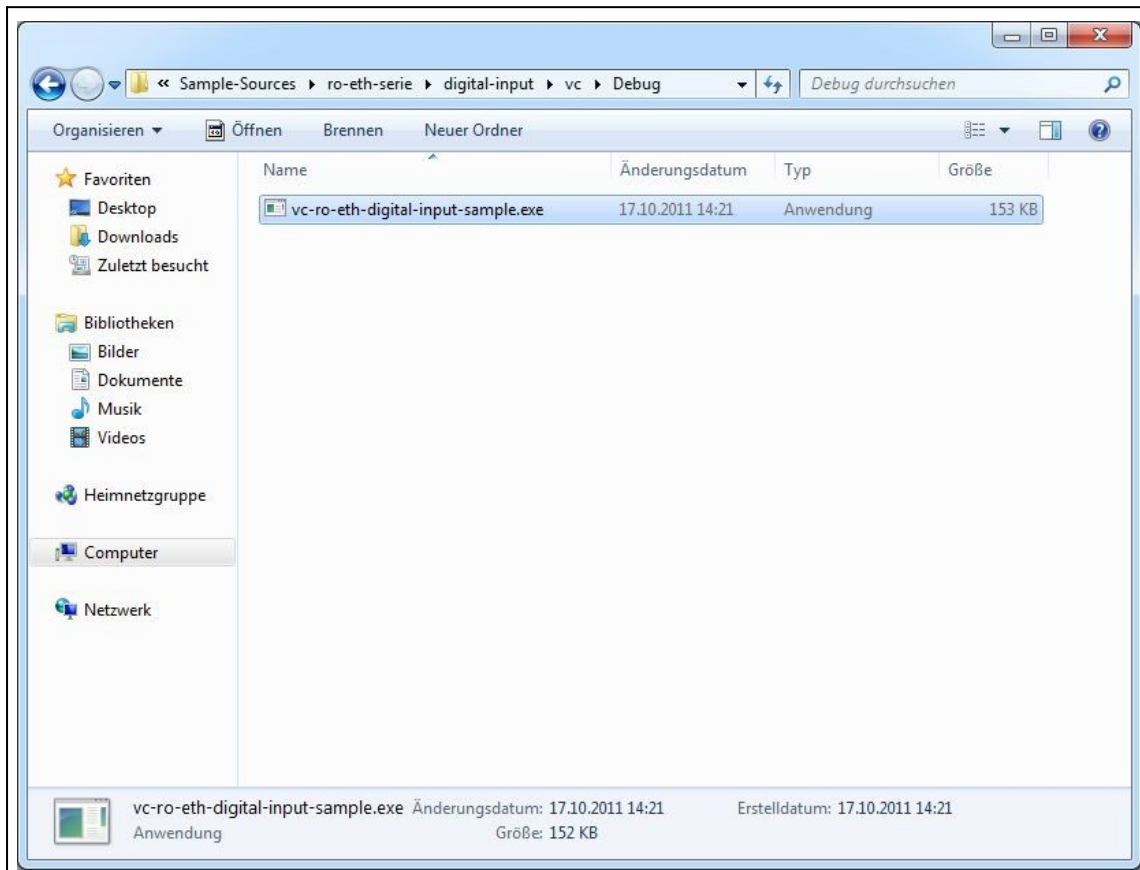
    // -----
    // Module found!
    printf("Module has been opened\n");

    // -----
    // Show config of module
    value = DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_GET_MODULE_CONFIG, DAPI_SPECIAL_GET_MODULE_CONFIG_PAR_DI, 0, 0);
    if (value < 0)
    {
        printf("Configuration of the module: no. of digital inputs %d\n", value);
        printf("Press any key to continue\n");
        getch();
    }

    // -----
    // Read input channels
    ret = DapiGetI(handle, 0);
    if (ret < 0)
    {
        printf("Reading input channel 0\n");
        printf("value = %d\n", ret);
        printf("Press any key to continue\n");
        getch();
    }

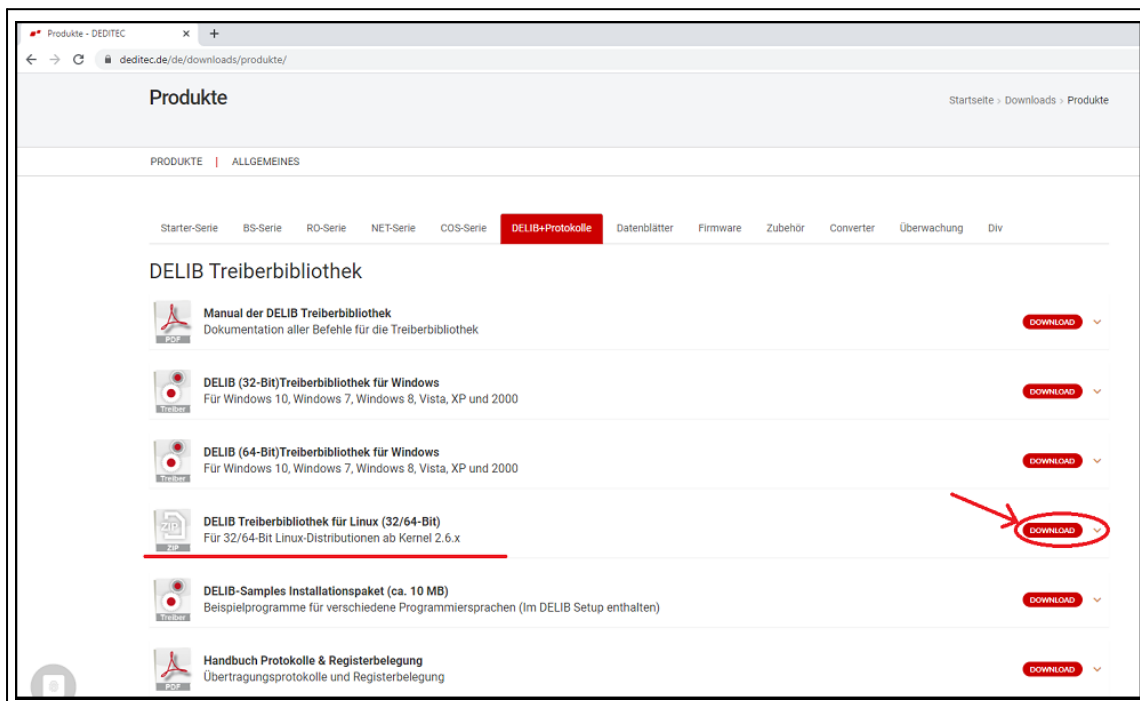
    ret = DapiGetI(handle, 1);
}
```

Zusätzlich finden Sie im Ordner debug ein bereits kompiliertes und ausführbares Programm zu diesem Projekt.

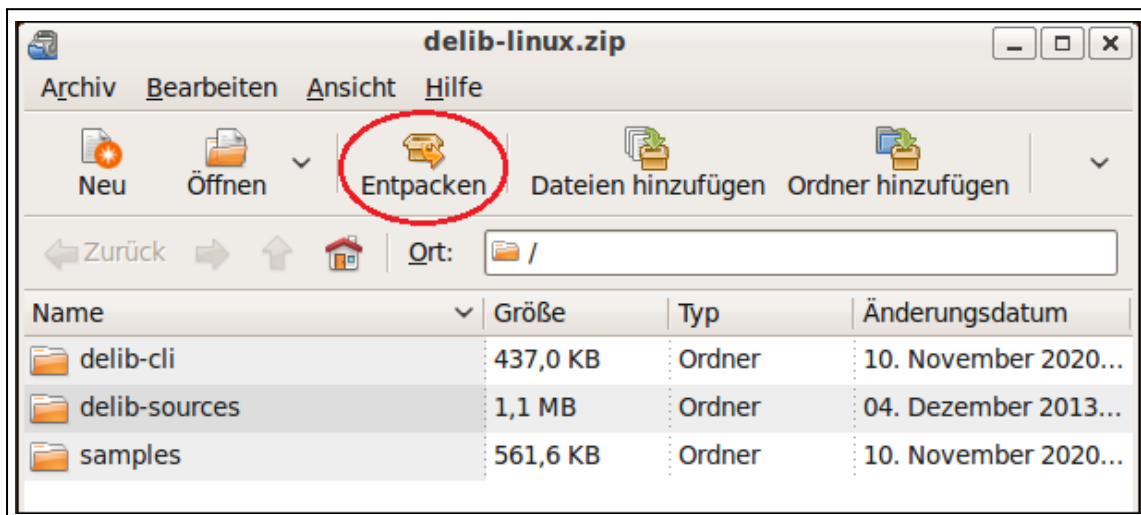


### 3.4. DELIB für Linux

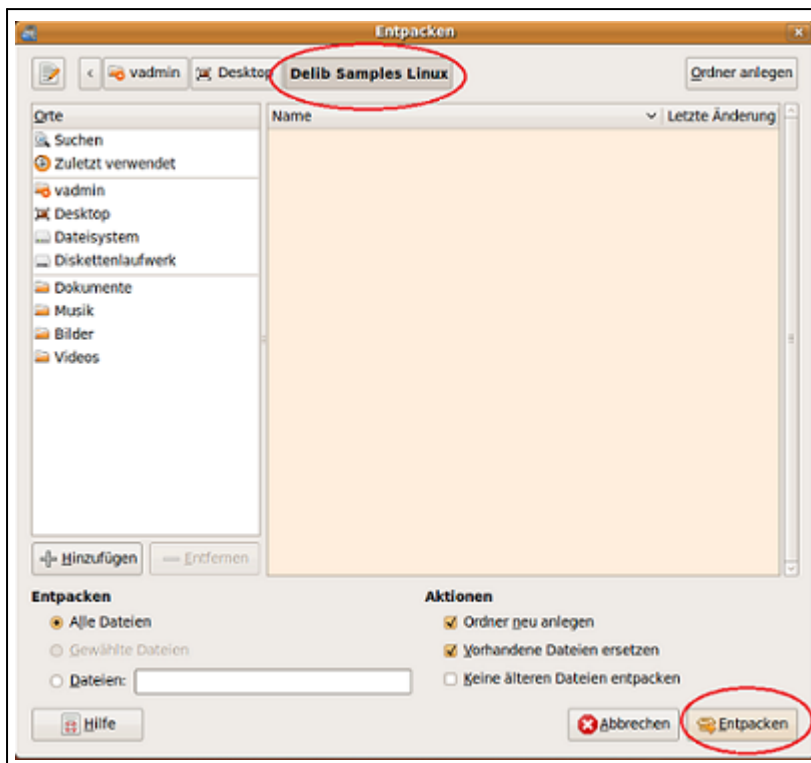
Laden Sie sich die Delib Linux Treiberbibliothek unter "[www.deditec.de/de/downloads/produkte/](http://www.deditec.de/de/downloads/produkte/)" im Reiter „DELIB+Protokolle“ oder unter "[www.deditec.de/media/zip/delib/delib-linux.zip](http://www.deditec.de/media/zip/delib/delib-linux.zip)" direkt auf ihr Linux-System.



Entpacken Sie die "delib-linux.zip" in einen beliebigen Zielordner. Doppelklicken Sie dafür auf die Zip-Datei und benutzen Sie dann den "Entpacken"-Knopf in der oberen Menüleiste.



Wählen Sie Ihren Zielordner aus und klicken Sie dann auf den "Entpacken"-Knopf.



### 3.4.1. Verwenden der DELIB Treiberbibliothek für Linux

#### 3.4.1.1. Delib USB-Sample in Linux

##### Voreinstellungen

In diesem Programmbeispiel wird ein USB\_REL AIS\_8 Modul angesprochen. Sollten Sie ein anderes Modul verwenden, müssen Sie in der Datei

„./samples/usb\_sample/source/usb\_sample.c“ bei dem Befehl „DapiOpenModule“ ihr Modul angeben. Die genaue Bezeichnung können Sie der „delib.h“ entnehmen. Diese finden sie im Verzeichnis „./delib-sources/delib/library/delib/delib.h“

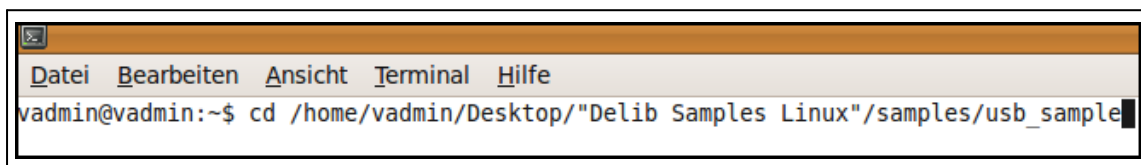
```
23
24 #include <stdio.h>
25 #include <stdlib.h>
26 #include <unistd.h>
27
28 #include "../../delib-sources/delib/library/delib/delib.h"
29
30 int main()
31 {
32     ULONG i;
33     ULONG handle=0;
34
35     printf("\n\n");
36     printf("-----\n");
37     printf("-----\n");
38     printf("-----\n");
39     printf("WICHTIG !!!\n");
40     printf("Dieses Programm bitte mit admin-Rechten ausfuehren\n");
41     printf("Also: sudo ./delib-test-digital-io <return>\n");
42     printf("-----\n");
43     printf("-----\n");
44     printf("-----\n");
45     printf("\n\n");
46
47     printf("-----\n");
48     printf("Try to open USB_REL AIS_8\n");
49     handle = DapiOpenModule(USB_REL AIS_8, 0);
50
51     if(handle == 0)
52     {
53         // Module not found
54         printf("Handle = 0x%x\n", (unsigned long) handle);
55         return 0;
56     }
57
58     printf("Handle = 0x%x\n", (unsigned long) handle);
59 }
```

## Kompilieren des USB-Samples

Für das Kompilieren des Testprogramms öffnen Sie ein Terminalfenster und navigieren mit dem Befehl

"cd /<Verzeichnispfad>" zunächst in das "/samples/usb\_sample" Verzeichnis.

Tipp: Sollten in Ihrem Ordernamen Leerzeichen enthalten sein, geben Sie diese wie im unteren Beispiel dargestellt in " " an.



```

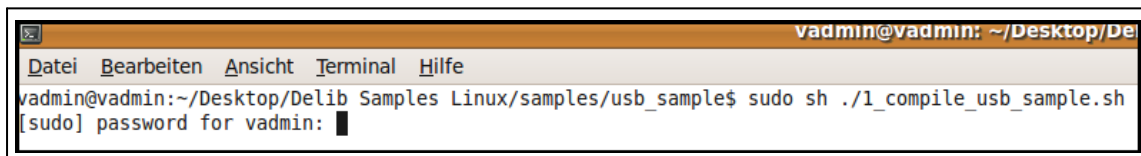
Datei Bearbeiten Ansicht Terminal Hilfe
vadmin@vadmin:~$ cd /home/vadmin/Desktop/"Delib Samples Linux"/samples/usb_sample

```

Zum Kompilieren öffnen Sie nun das darin enthaltene Shell-Skript mit dem Befehl

„sudo sh ./1\_compile\_usb\_sample.sh“.

Geben Sie, falls nötig, Ihr Benutzerkennwort ein.



```

vadmin@vadmin: ~/Desktop/Del
Datei Bearbeiten Ansicht Terminal Hilfe
vadmin@vadmin:~/Desktop/Delib Samples Linux/samples/usb_sample$ sudo sh ./1_compile_usb_sample.sh
[sudo] password for vadmin:

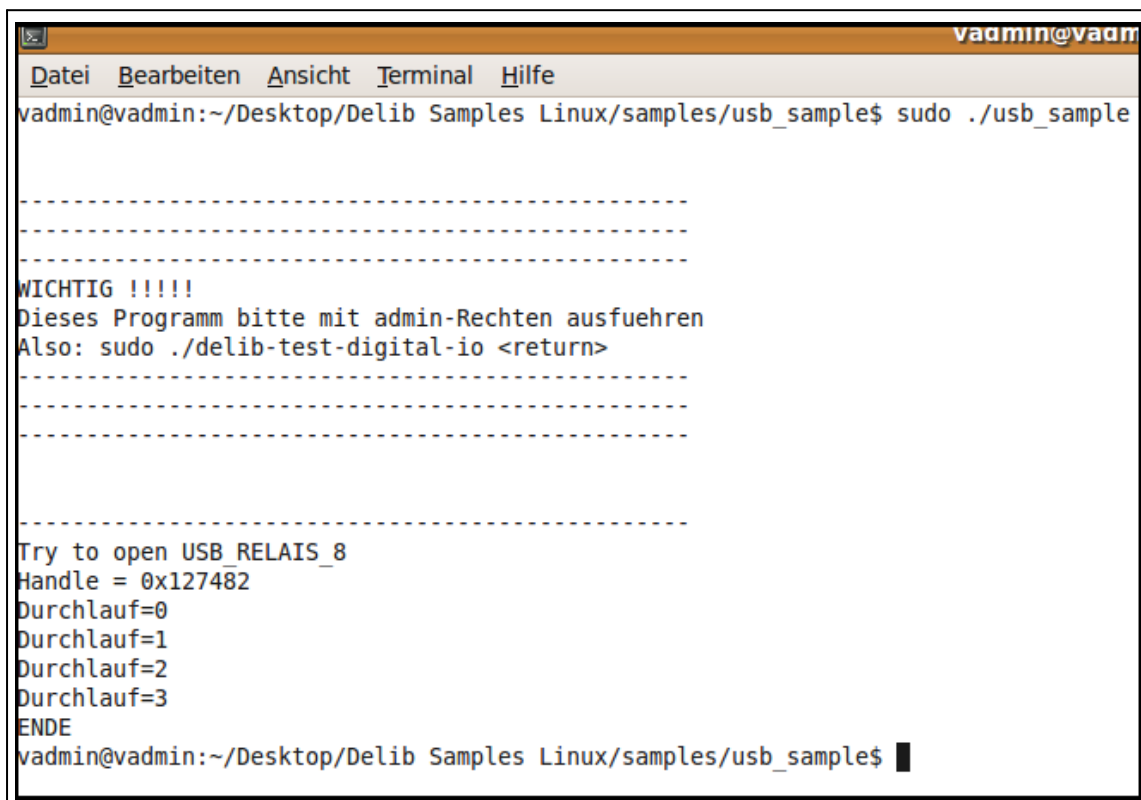
```

Bei erfolgreicher Kompilierung sollte nun "compiling successful" im Terminalfenster erscheinen.

Es wurde die Datei "usb\_sample" dem Verzeichnis hinzugefügt.

Jetzt können Sie das Beispielprogramm mit "sudo ./usb\_sample" ausführen.

**WICHTIG!!** Sie benötigen für das Ausführen Admin-Rechte. Benutzen Sie deshalb den Befehl mit "sudo"



```
vadmin@vadmin:~/Desktop/Delib Samples Linux/samples/usb_sample$ sudo ./usb_sample

-----
WICHTIG !!!!!
Dieses Programm bitte mit admin-Rechten ausfuehren
Also: sudo ./delib-test-digital-io <return>
-----

-----

Try to open USB_RELAIS_8
Handle = 0x127482
Durchlauf=0
Durchlauf=1
Durchlauf=2
Durchlauf=3
ENDE
vadmin@vadmin:~/Desktop/Delib Samples Linux/samples/usb_sample$
```

Das Programm wird nun ausgeführt.

In diesem Beispiel werden alle digitalen Ausgänge des USB\_RELAIS\_8 in einer Schleife an und wieder ausgeschaltet.



### 3.4.1.2. Delib ETH-Sample in Linux

#### Voreinstellungen

Bei diesem Programmbeispiel wird das Modul mit der IP "192.168.1.21" angesprochen. Diese können Sie in der Datei

„./samples/ethernet\_sample/source/eth\_sample.c“ ändern (siehe Bild unten).

Falls Sie ein Kennwort für eine verschlüsselte TCP Verbindung voreingestellt haben, können Sie dieses ebenfalls dort eintragen (siehe Bild unten). Haben Sie kein Passwort angegeben, können Sie diese Zeile unverändert lassen.

Die Konfiguration der ETH-Module können über das DELIB Configuration Utility, sowie über die Weboberfläche des Moduls eingestellt werden.

```
26 #include <string.h>
27 #include <unistd.h>
28
29 #include "../delib-sources/delib/library/delib/delib.h"
30
31 int main()
32 {
33     unsigned long i;
34     unsigned long handle;
35     unsigned long ret;
36     DAPI_OPENMODULEEX_STRUCT open_buffer;
37
38     strcpy((char*) open_buffer.address, "192.168.1.21"); // hostname
39     open_buffer.timeout = 5000; // 5000 msec
40     open_buffer.portno = 9912; // using default port
41
42     #ifdef ENABLE_TCP_ENCRYPTION
43         open_buffer.encryption_type = DAPI_OPEN_MODULE_ENCRYPTION_TYPE_ADMIN; // encrypted communication with admin priv
44         strcpy((char*) open_buffer.encryption_password, "myPassword"); // password for encrypted communication
45     #else
46         open_buffer.encryption_type = DAPI_OPEN_MODULE_ENCRYPTION_TYPE_NONE; // Falls vorher eingestellt, geben Sie hier das Passwort
47     #endif // Ihrer verschlüsselten TCP-Verbindung an.
48
49     handle = DapiOpenModuleEx(ETHERNET_MODULE, 0, (unsigned char*) &open_buffer, DAPI_OPEN_MODULE_OPTION_USE_EXBUFFER);
50
51     if(handle == 0)
52     {
```

Sollten Sie ein Modul ohne digitale Eingänge verwenden, müssen Sie die Zeilen wie unten dargestellt, in der gleichen Datei auskommentieren.

```
57     for(i=0; i!=4; ++i)
58     {
59         printf("Durchlauf = %ld\n", i);
60
61         DapiDOSet8(handle, 0, 0xff);
62
63         usleep(1000 * 500);          // 500 msec sleep
64
65         DapiDOSet8(handle, 0, 0);
66
67         usleep(1000 * 500);          // 500 msec sleep
68         //ret = DapiDIGet8(handle, 0);
69         //printf("DI0-7 = 0x%lx\n", ret);
70
71         usleep(1000 * 500);          // 500 msec sleep
72     }
73
74
```

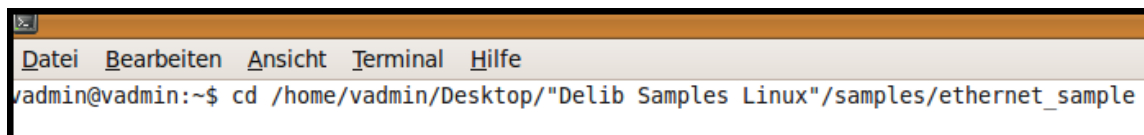
Auskommentieren,  
falls keine digitalen  
Eingänge vorhanden

## Kompilieren des ETH-Samples

Für das Kompilieren des Testprogramms, öffnen Sie ein Terminalfenster und navigieren mit dem Befehl

"cd /<Verzeichnispfad>" zunächst in das "/samples/ethernet\_sample" Verzeichnis.

Tipp: Sollten in Ihrem Ordnernamen Leerzeichen enthalten sein, geben Sie diese wie im unteren Beispiel dargestellt in " " an.

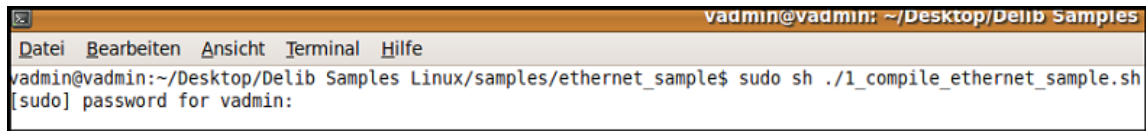
A screenshot of a terminal window with a menu bar containing 'Datei', 'Bearbeiten', 'Ansicht', 'Terminal', and 'Hilfe'. The terminal text shows a user 'vadmin' at host 'vadmin' in the directory '~' executing the command 'cd /home/vadmin/Desktop/"Delib Samples Linux"/samples/ethernet\_sample'.

```
vadmin@vadmin:~$ cd /home/vadmin/Desktop/"Delib Samples Linux"/samples/ethernet_sample
```

Zum Kompilieren öffnen Sie nun das gewünschte Shell-Skript mit dem Befehl „sudo sh ./<DATEINAME>“

- Möchten Sie das Modul über eine unverschlüsselte TCP Verbindung ansteuern, verwenden Sie die Datei „1\_compile\_ethernet\_sample.sh“
- Möchten Sie das Modul über eine verschlüsselte TCP Verbindung ansteuern, verwenden Sie die Datei „2\_compile\_ethernet\_sample\_with\_encryption.sh“

Geben Sie, falls nötig, Ihr Benutzerkennwort ein.



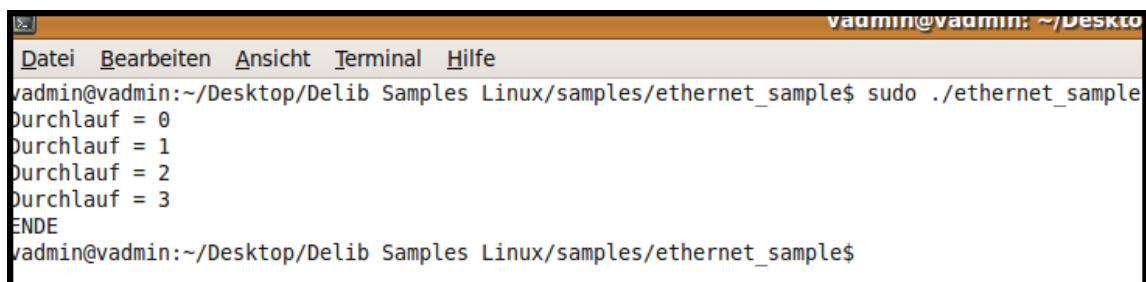
```
vadmin@vadmin: ~/Desktop/Delib Samples
Datei Bearbeiten Ansicht Terminal Hilfe
vadmin@vadmin:~/Desktop/Delib Samples Linux/samples/ethernet_sample$ sudo sh ./1_compile_ethernet_sample.sh
[sudo] password for vadmin:
```

Bei erfolgreicher Kompilierung sollte nun "compiling successful" im Terminalfenster erscheinen.

Es wurde die Datei "ethernet\_sample" dem Verzeichnis hinzugefügt.

Jetzt können Sie das Beispielprogramm mit "sudo ./ethernet\_sample" ausführen.

**WICHTIG!!** Sie benötigen für das Ausführen Admin-Rechte. Benutzen Sie deshalb den Befehl mit "sudo".



```
vadmin@vadmin: ~/Desktop
Datei Bearbeiten Ansicht Terminal Hilfe
vadmin@vadmin:~/Desktop/Delib Samples Linux/samples/ethernet_sample$ sudo ./ethernet_sample
Durchlauf = 0
Durchlauf = 1
Durchlauf = 2
Durchlauf = 3
ENDE
vadmin@vadmin:~/Desktop/Delib Samples Linux/samples/ethernet_sample$
```

Das Programm wird nun ausgeführt.

In diesem Beispiel werden alle Ausgänge des Moduls in einer Schleife an und wieder ausgeschaltet.

### 3.4.2. DELIB CLI (command-line interface) für Linux

Der DELIB CLI Befehl für Linux befindet sich nach Entpacken des Zip-Archivs "delib-linux-cli" im Ordner /deditec-cli/ .

Definition für USB-Module (Linux)

```
sudo delib_cli [command] [channel] [value / unit ["nount"]]
```

Definition für ETH-Module (Linux)

```
delib_cli [command] [channel] [value / unit ["nount"]]
```

#### **Hinweis:**

Die einzelnen Parameter werden nur durch ein Leerzeichen getrennt.

Groß und Kleinschreibung wird hierbei nicht beachtet.

## Parameter

Befehl	Kabal	Wert		unit	nounit
di1	0, 1, 2, ...	-		hex	nounit
di8	0, 8, 16, ...				
di16					
di32					
ff	0, 32, ...	-		hex	nounit
do1	0, 1, 2, ...	0/1 (1-Bit Befehl)		-	-
do8	0, 8, 16, ...	8-Bit Wert	(Bit 0 für Kanal 1, Bit 1 für Kanal 2, ...)		
do16		16-Bit Wert			
do32		32-Bit Wert			
ai	0, 1, 2, ...	-		hex, volt, mA	nounit
ao	0, 1, 2, ...	Ganz oder Hexadezimalzahl (beginnend mit 0x).		-	-

### **Return-Wert**

#### **Zustand der gelesenen digitalen Eingänge**

In Kombination mit Parameter unit "hex" wird der Zustand als hex gelesen

#### **Zustand der FlipFlips der digitalen Eingänge**

In Kombination mit Parameter unit "hex" wird der Zustand als hex gelesen

#### **Zustand der gelesenen analogen Eingänge**

In Kombination mit Parameter unit "hex" wird der Zustand als hex gelesen

In Kombination mit Parameter unit "volt" wird die Spannung gelesen

In Kombination mit Parameter unit "mA" wird der Strom gelesen

### 3.4.2.1. Konfiguration des DELIB CLI

#### Voreinstellungen

Vor der ersten Verwendung des DELIB CLI muss die "delib\_cli.cfg" mit einem Texteditor bearbeitet werden.

Sie finden die "delib\_cli.cfg" im Verzeichnis "/delib\_cli/".

#### Inhalt der "delib\_cli.cfg":

```
moduleID=14;  
moduleNR=0;  
RO-ETH_ipAddress=192.168.1.11;
```

#### moduleID

Als moduleID muss die entsprechende Nummer der eingesetzten Hardware eingetragen werden.

Diese Nummer kann der "delib.h" entnommen werden.

Unter Linux finden Sie diese im Zip-Archiv des "delib-linux" unter dem Pfad "delib-sources\delib\library\delib".

#### moduleNR

Die moduleNR wird im DELIB Configuration Utility vergeben.

Diese Nummer dient zur Identifizierung identischer Hardware.

Der Standardwert ist 0.

#### RO-ETH\_ipAddress

Dieser Eintrag wird ausschließlich für die Verbindung zu unseren ETH-Modulen benötigt.

Die IP-Adresse der ETH-Module können über das DELIB Configuration Utility sowie über die Weboberfläche des Moduls eingestellt werden.

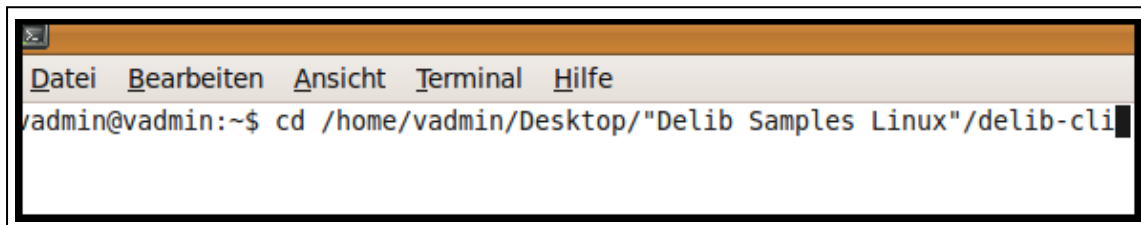


## Kompilieren des Delib-CLI-Samples

Für das Kompilieren des Testprogramms, öffnen Sie ein Terminalfenster und navigieren mit dem Befehl

"cd /<Verzeichnispfad>" zunächst in das "../delib\_cli/" Verzeichnis.

Tip: Sollten in Ihrem Ordernamen Leerzeichen enthalten sein, geben Sie diese wie im unteren Beispiel dargestellt in " " an.

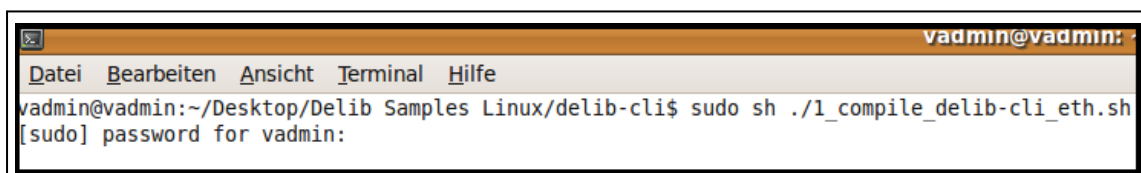


```
admin@vadmin:~$ cd /home/vadmin/Desktop/"Delib Samples Linux"/delib-cli
```

Zum Kompilieren öffnen Sie nun das gewünschte Shell-Skript mit dem Befehl „sudo sh ./<DATEINAME>“

- ETH - "1\_compile\_delib-cli\_eth.sh"
- USB - "2\_compile\_delib-cli\_usb.sh"

Geben Sie, falls nötig, Ihr Benutzerkennwort ein.

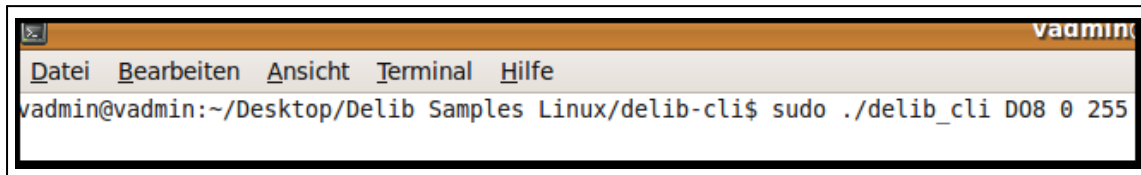


```
admin@vadmin:~/Desktop/Delib Samples Linux/delib-cli$ sudo sh ./1_compile_delib-cli_eth.sh
[sudo] password for vadmin:
```

Bei erfolgreicher Kompilierung sollte nun "compiling successfull" im Terminalfenster erscheinen. Es wurde die Datei "delib\_cli" im Verzeichnis erstellt. Jetzt können Sie das Beispielprogramm mit

"sudo ./delib\_cli [command] [channel] [value | unit ["nounit"]] " ausführen.

**WICHTIG!!** Sie benötigen für das Ausführen Admin-Rechte. Benutzen Sie deshalb den Befehl mit "sudo".



A terminal window titled 'vadmin' with a menu bar containing 'Datei', 'Bearbeiten', 'Ansicht', 'Terminal', and 'Hilfe'. The command prompt shows the user 'vadmin' at host 'vadmin' in the directory '~/Desktop/Delib Samples Linux/delib-cli'. The command being executed is 'sudo ./delib\_cli D08 0 255'.

```
vadmin@vadmin:~/Desktop/Delib Samples Linux/delib-cli$ sudo ./delib_cli D08 0 255
```

### 3.4.2.2. DELIB CLI Beispiele

#### Digitale Ausgänge

```
sudo delib_cli DO1 17 1
```

→ schaltet das 18. digitale Relais eines USB-Moduls an

```
sudo delib_cli DO1 3 0
```

→ schaltet das 4. digitale Relais eines RO-ETH-Moduls aus

#### Digitale Eingänge

```
sudo delib_cli DI1 3
```

Beispiel eines Rückgabewertes: **1**

→ lese den Zustand des 4. digitalen Eingangs eines USB-Moduls und gebe ihn zurück

```
sudo delib_cli DI8 0 hex
```

Beispiel eines Rückgabewertes: **0xFF**

(auf den Kanälen 1 bis 8 liegt ein Signal an)

→ lese den Wert von digitalen Eingang 1-8 eines RO-ETH-Moduls als hexadezimalzahl

```
sudo delib_cli FF 0
```

Beispiel eines Rückgabewertes: **192**

(auf den Kanälen 7 und 8 wurde eine Zustandsänderung erkannt)

→ lese den Wert der FlipFlops der digitalen Eingänge 1-32

```
sudo delib_cli FF 32
```

Beispiel eines Rückgabewertes: **65535**

(auf den Kanälen 33 bis 64 wurde eine Zustandsänderung erkannt)

→ lese den Wert der FlipFlops der digitalen Eingänge 33-64

```
sudo delib_cli FF 0 hex
```

Beispiel eines Rückgabewertes: **0xD00**

(auf Kanälen 9, 11 und 12 wurde eine Zustandsänderung erkannt)

→ lese den Wert der FlipFlops der digitalen Eingänge 1-32 als hexadezimalzahl

### Analoge Ausgänge

```
sudo delib_cli AO 7 4711
```

→ setzt den dezimalen Wert 4711 auf den 8. analogen Ausgang eines USB-Moduls

```
sudo delib_cli AO 6 0x4711
```

→ setzt den hexadezimalen Wert 0x4AF1 auf den 7. analogen Ausgang eines RO-ETH-Moduls

### Analoge Eingänge

```
sudo delib_cli AI 2
```

Beispiel eines Rückgabewertes: **1234**

→ liest den Wert des 3. analogen Eingangs als dezimalzahl eines USB-Moduls

```
sudo delib_cli AI 2 hex
```

Beispiel eines Rückgabewertes: **0x1FA**

→ liest den Wert des 3. analogen Eingangs als hexadezimalzahl eines RO-ETH-Moduls

## **DELIB API Referenz**

---



## 4. DELIB API Referenz

### 4.1. Verfügbare DEDITEC Modul IDs

Hier finden Sie eine Auflistung mit allen verfügbaren Modul IDs.

Diese ID wird benötigt, um beispielsweise das Modul zu öffnen und einen "handle" zu erhalten.

Mehr Informationen dazu finden Sie im Kapitel **DapiOpenModule**.

Modul Name	ID
USB_Interface8	1
USB_CAN_STICK	2
USB_LOGI_500	3
USB_SER_DEBUG	4
RO_SER	5
USB_BITP_200	6
RO_USB1	7
RO_USB	7
RO_ETH	8
USB_MINI_STICK	9
USB_LOGI_18	10
RO_CAN	11
USB_SPI_MON	12
USB_WATCHDOG	13
USB_OPTOIN_8	14

Modul Name	ID
USB_REL AIS_8	14
USB_OPTOIN_8_REL AIS_8	15
USB_OPTOIN_16_REL AIS_16	16
USB_OPTOIN_32	16
USB_REL AIS_32	16
USB_OPTOIN_32_REL AIS_32	17
USB_OPTOIN_64	17
USB_REL AIS_64	17
BS_USB_8	15
BS_USB_16	16
BS_USB_32	17
USB_TTL_32	18
USB_TTL_64	18
RO_ETH_INTERN	19
BS_SER	20
BS_CAN	21
BS_ETH	22
NET_ETH	23
RO_CAN2	24
RO_USB2	25
RO_ETH_LC	26

Modul Name	ID
ETH_RELAIS_8	27
ETH_OPTOIN_8	27
ETH_O4_R4_ADDA	28
ETHERNET_MODULE	29
ETH_TTL_64	30
NET_USB2	31
NET_ETH_LC	32
NET_USB1	33
NET_SER	34
NET_CAN_OPEN	35
NET_RAS_PI	36
USB_CANOPEN_STICK	37
ETH_CUST_0	38
WEU_RELAIS_8	39
WEU_OPTO_8	39
WEU_E_RELAIS_8	40
BS_WEU	41
BS_WEU_E	42



## 4.2. Verwaltungsfunktionen

### 4.2.1. DapiOpenModule

#### Beschreibung

Diese Funktion öffnet ein bestimmtes Modul.

#### Definition

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

#### Parameter

moduleID=Gibt das Modul an, welches geöffnet werden soll (siehe delib.h)

nr=Gibt an, welches (bei mehreren Modulen) geöffnet werden soll.

nr=0 → 1. Modul

nr=1 → 2. Modul

#### Return-Wert

handle=Entsprechender Handle für das Modul

handle=0 → Modul wurde nicht gefunden

#### Bemerkung

Der von dieser Funktion zurückgegebene Handle wird zur Identifikation des Moduls für alle anderen Funktionen benötigt.

#### Programmierbeispiel

```
// 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;
}
```

### 4.2.2. DapiCloseModule

#### Beschreibung

Dieser Befehl schließt ein geöffnetes Modul.

#### Definition

*ULONG DapiCloseModule(ULONG handle);*

#### Parameter

handle=Dies ist das Handle eines geöffneten Moduls.

#### Return-Wert

Keiner

#### Programmierbeispiel

```
// Modul schliessen  
DapiCloseModule(handle);
```

### 4.2.3. DapiGetDELIBVersion

#### Beschreibung

Diese Funktion gibt die installierte DELIB-Version zurück.

#### Definition

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

#### Parameters

mode=Modus, mit dem die Version ausgelesen wird (muss 0 sein).

par=Dieser Parameter ist nicht definiert (muss 0 sein).

#### Return-Wert

version=Versionsnummer der installierten DELIB-Version [hex].

#### Programmierbeispiel

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

#### 4.2.4. DapiSpecialCMDGetModuleConfig

##### Beschreibung

Diese Funktion gibt die Hardwareausstattung (Anzahl der Ein- und Ausgangskanäle) des Moduls zurück.

##### Definition

```
ULONG DapiSpecialCommand(ULONG handle,  
DAPI_SPECIAL_CMD_GET_MODULE_CONFIG, par, 0, 0);
```

##### Parameter

handle=Dies ist der handle eines offenen Moduls

##### Querying the number of digital input channels

par=DAPI\_SPECIAL\_GET\_MODULE\_CONFIG\_PAR\_DI

##### Query number of digital input flip-flops

par=DAPI\_SPECIAL\_GET\_MODULE\_CONFIG\_PAR\_DI\_FF

##### Query number of digital input counters (16-bit counter)

par=DAPI\_SPECIAL\_GET\_MODULE\_CONFIG\_PAR\_DI\_COUNTER

##### Query number of digital input counters (48-bit counter)

par=DAPI\_SPECIAL\_GET\_MODULE\_CONFIG\_PAR\_CNT48

##### Querying the number of digital output channels

par=DAPI\_SPECIAL\_GET\_MODULE\_CONFIG\_PAR\_DO

##### Querying the number of digital pulse generator outputs

par=DAPI\_SPECIAL\_GET\_MODULE\_CONFIG\_PAR\_PULSE\_GEN

##### Querying the number of digital PWM outputs

par=DAPI\_SPECIAL\_GET\_MODULE\_CONFIG\_PAR\_PWM\_OUT

##### Querying the number of digital input/output channels

par=DAPI\_SPECIAL\_GET\_MODULE\_CONFIG\_PAR\_DX

#### **Querying the number of analog input channels**

par=DAPI\_SPECIAL\_GET\_MODULE\_CONFIG\_PAR\_AD

#### **Querying the number of analog output channels**

par=DAPI\_SPECIAL\_GET\_MODULE\_CONFIG\_PAR\_DA

#### **Query number of temperature channels**

par=DAPI\_SPECIAL\_GET\_MODULE\_CONFIG\_PAR\_TEMP

#### **Query number of stepper channels**

par=DAPI\_SPECIAL\_GET\_MODULE\_CONFIG\_PAR\_STEPPER

#### **Return value**

##### **Querying the number of digital input channels**

return=number of digital input channels

##### **Query number of digital input flip-flops**

return=number of digital input flip-flops

##### **Query number of digital input counters (16-bit counter)**

return=number of digital input counters (16-bit counter)

##### **Query number of digital input counters (48-bit counter)**

return=number of digital input counters (48-bit counter)

##### **Querying the number of digital output channels**

return=number of digital output channels

##### **Querying the number of digital pulse generator outputs**

return=number of digital pulse generator outputs

##### **Querying the number of digital PWM outputs**

return=number of digital PWM outputs

### Querying the number of digital input/output channels

return=number of digital input/output channels

### Querying the number of analog input channels

return=number of analog input channels

### Querying the number of analog output channels

return=number of analog output channels

### Query number of temperature channels

return=number of temperature channels

### Query number of stepper channels

return=number of stepper channels

### Programmierbeispiele

```
ret=DapiSpecialCommand(handle,
DAPI_SPECIAL_CMD_GET_MODULE_CONFIG,
DAPI_SPECIAL_GET_MODULE_CONFIG_PAR_DI, 0, 0);
//Returns the number of digital input channels

ret=DapiSpecialCommand(handle,
DAPI_SPECIAL_CMD_GET_MODULE_CONFIG,
DAPI_SPECIAL_GET_MODULE_CONFIG_PAR_DO, 0, 0);
//Returns the number of digital output channels

ret=DapiSpecialCommand(handle,
DAPI_SPECIAL_CMD_GET_MODULE_CONFIG,
DAPI_SPECIAL_GET_MODULE_CONFIG_PAR_DX, 0, 0);
//Returns the number of digital input/output channels

ret=DapiSpecialCommand(handle,
DAPI_SPECIAL_CMD_GET_MODULE_CONFIG,
DAPI_SPECIAL_GET_MODULE_CONFIG_PAR_AD, 0, 0);
//Returns the number of analog input channels

ret=DapiSpecialCommand(handle,
DAPI_SPECIAL_CMD_GET_MODULE_CONFIG,
DAPI_SPECIAL_GET_MODULE_CONFIG_PAR_DA, 0, 0);
//Returns the number of analog output channels

ret=DapiSpecialCommand(handle,
DAPI_SPECIAL_CMD_GET_MODULE_CONFIG,
DAPI_SPECIAL_GET_MODULE_CONFIG_PAR_STEPPER, 0, 0);
//Returns the number of stepper channels
```

#### 4.2.5. DapiOpenModuleEx

##### Beschreibung

Diese Funktion öffnet gezielt ein Modul mit Ethernet-Schnittstelle. Dabei können die Parameter IP-Adresse, Portnummer, die Dauer des Timeouts und der Encryption Type bestimmt werden.

Das Öffnen des Moduls geschieht dabei unabhängig von den im DELIB Configuration Utility getroffenen Einstellungen.

##### Definition

*ULONG DapiOpenModuleEx(ULONG moduleID, ULONG nr, unsigned char\* exbuffer, 0);*

##### Parameter

moduleID = Gibt das Modul an, welches geöffnet werden soll (siehe delib.h)

nr = Gibt an, welches Modul (bei mehreren Modulen) geöffnet werden soll.

nr = 0 → 1. Modul

nr = 1 → 2. Modul

exbuffer = Buffer für IP-Adresse, Portnummer, Dauer des Timeouts und der Encryption Type

##### Return-Wert

handle = Entsprechender Handle für das Modul

handle = 0 → Modul wurde nicht gefunden

### Bemerkung

Der von dieser Funktion zurückgegebene Handle wird zur Identifikation des Moduls für alle anderen Funktionen benötigt.

Dieser Befehl wird von allen Modulen mit Ethernet-Schnittstelle unterstützt.

Universelle Ethernet moduleID

### Die moduleID:

ETHERNET\_MODULE = 29

ist eine universelle Ethernet moduleID und kann benutzt werden, um jedes Ethernet Produkt anzusprechen.

### Encryption Type

Folgende Encryption Types stehen zur Verfügung:

DAPI\_OPEN\_MODULE\_ENCRYPTION\_TYPE\_NONE = 0

DAPI\_OPEN\_MODULE\_ENCRYPTION\_TYPE\_NORMAL = 1

DAPI\_OPEN\_MODULE\_ENCRYPTION\_TYPE\_ADMIN = 2

### Programmierbeispiel

```
// Open ETH-Module with parameter
DAPI_OPENMODULEEX_STRUCT open_buffer;

strcpy((char*) open_buffer.address, "192.168.1.10");
open_buffer.portno = 0;
open_buffer.timeout = 5000;
open_buffer.encryption_type = 0;

handle = DapiOpenModuleEx(RO_ETH, 0, (unsigned char*)
&open_buffer, 0);
printf("Module handle = %x\n", handle);
```

#### 4.2.6. DapiScanAllModulesAvailable

##### Beschreibung

Mit dieser Funktion lassen sich alle am USB-Bus angeschlossenen Module scannen.

Hierbei wird die Modul-ID und die Modul-Nr. jedes gefundenen Modules ermittelt.

##### Definition

*ULONG DapiScanAllModulesAvailable(uint nr)*

##### Parameter

nr = 0: Es wird nach allen am USB-Bus angeschlossenen Module gesucht  
nr = i: Auslesen der einzelnen angeschlossenen Module

##### Return-Wert

Gibt die Anzahl der gefunden Module wieder.

##### Programmierbeispiel

```
no_of_modules =  
DT.Delib.DapiScanAllModulesAvailable(0);  
for (i = 1; i <= no_of_modules; i++)  
{  
    ret = DapiScanAllModulesAvailable(i);  
    moduleID = ret & 0x0000ffff;  
    moduleNr = (ret >> 16) & 0xff;  
}
```



## 4.3. Fehlerbehandlung

### 4.3.1. DapiGetLastError

#### Beschreibung

Diese Funktion liefert den letzten erfassten Fehler. Sofern ein Fehler aufgetreten ist, muss dieser mit **DapiClearLastError()** gelöscht werden, da sonst jeder Aufruf von DapiGetLastError() den "alten" Fehler zurückgibt.

Sollen mehrere Module verwendet werden, empfiehlt sich die Verwendung von **DapiGetLastErrorByHandle()**.

#### Definition

*ULONG DapiGetLastError();*

#### Parameter

Keine

#### Return-Wert

Fehler Code

0=kein Fehler. (siehe delib\_error\_codes.h)

#### Programmierbeispiel

```
BOOL IsError()
{
    unsigned char msg[500];
    unsigned long error_code = DapiGetLastError();
    if (error_code != DAPI_ERR_NONE)
    {
        DapiGetLastErrorText((unsigned char*) msg,
        sizeof(msg));
        printf("Error Code = 0x%x * Message = %s\n",
        error_code, msg);
        DapiClearLastError();
        return TRUE;
    }
    return FALSE;
}
```

### 4.3.2. DapiGetLastErrorText

#### Beschreibung

Diese Funktion liest den Text des letzten erfassten Fehlers. Sofern ein Fehler aufgetreten ist, muss dieser mit **DapiClearLastError()** gelöscht werden, da sonst jeder Aufruf von DapiGetLastErrorText() den "alten" Fehler zurückgibt.

#### Definition

*ULONG DapiGetLastErrorText(unsigned char \*msg, unsigned long msg\_length);*

#### Parameter

msg = Buffer für den zu empfangenden Text

msg\_length = Länge des Text Buffers

#### Programmierbeispiel

```
BOOL IsError()
{
    unsigned char msg[500];
    unsigned long error_code = DapiGetLastError();

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

        DapiClearLastError();

        return TRUE;
    }

    return FALSE;
}
```

### 4.3.3. DapiClearLastError

#### Beschreibung

Diese Funktion löscht den letzten Fehler, der mit **DapiGetLastError()** erfasst wurde.

#### Definition

*void DapiClearLastError();*

#### Parameter

Keine

#### Return-Wert

Keine

#### Programmierbeispiel

```
BOOL IsError()
{
    unsigned char msg[500];
    unsigned long error_code = DapiGetLastError();

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

        DapiClearLastError();

        return TRUE;
    }

    return FALSE;
}
```

#### 4.3.4. DapiGetLastErrorByHandle

##### Beschreibung

Diese Funktion liefert den letzten erfassten Fehler eines bestimmten Moduls (handle). Sofern ein Fehler aufgetreten ist, muss dieser mit **DapiClearLastErrorByHandle()** gelöscht werden, da sonst jeder Aufruf von DapiGetLastErrorByHandle() den "alten" Fehler zurückgibt.

##### Definition

*ULONG DapiGetLastErrorByHandle(ULONG handle);*

##### Parameter

handle=Dies ist das Handle eines geöffneten Moduls.

##### Return-Wert

Fehler Code

0=kein Fehler. (siehe delib\_error\_codes.h)

##### Programmierbeispiel

```
BOOL IsError(ULONG handle)
{
    unsigned long error_code =
    DapiGetLastErrorByHandle(handle);

    if (error_code != DAPI_ERR_NONE)
    {
        printf("Error detected on handle 0x%x - Error
Code = 0x%x\n", handle, error_code);

        DapiClearLastErrorByHandle(handle);

        return TRUE;
    }

    return FALSE;
}
```

### 4.3.5. DapiClearLastErrorByHandle

#### Beschreibung

Diese Funktion löscht den letzten Fehler eines bestimmten Moduls (handle), der mit **DapiGetLastErrorByHandle()** erfasst wurde.

#### Definition

*void DapiClearLastErrorByHandle();*

#### Parameter

handle=Dies ist das Handle eines geöffneten Moduls.

#### Return-Wert

Keine

#### Programmierbeispiel

```
BOOL IsError(ULONG handle)
{
    unsigned long error_code =
    DapiGetLastErrorByHandle(handle);

    if (error_code != DAPI_ERR_NONE)
    {
        printf("Error detected on handle 0x%x - Error
        Code = 0x%x\n", handle, error_code);

        DapiClearLastErrorByHandle(handle);

        return TRUE;
    }

    return FALSE;
}
```

## 4.4. Digitale Eingänge lesen

### 4.4.1. DapiDIGet1

#### Beschreibung

Dieser Befehl liest einen einzelnen digitalen Eingang.

#### Definition

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

#### Parameter

handle=Dies ist das Handle eines geöffneten Moduls

ch=Gibt die Nummer des Eingangs an, der gelesen werden soll (0, 1, 2, 3, .. )

#### Return-Wert

Zustand des Eingangs (0/1)

### 4.4.2. DapiDIGet8

#### Beschreibung

Dieser Befehl liest gleichzeitig 8 digitale Eingänge.

#### Definition

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

#### Parameter

handle=Dies ist das Handle eines geöffneten Moduls

ch=Gibt die Nummer des Eingangs an, ab dem gelesen werden soll (0, 8, 16, 24, .. )

#### Return-Wert

Zustand der gelesen Eingänge

#### 4.4.3. DapiDIGet16

##### Beschreibung

Dieser Befehl liest gleichzeitig 16 digitale Eingänge.

##### Definition

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

##### Parameter

handle=Dies ist das Handle eines geöffneten Moduls

ch=Gibt die Nummer des Eingangs an, ab dem gelesen werden soll (0, 16, 32, ...)

##### Return-Wert

Zustand der gelesenen Eingänge

#### 4.4.4. DapiDIGet32

##### Beschreibung

Dieser Befehl liest gleichzeitig 32 digitale Eingänge.

##### Definition

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

##### Parameter

handle=Dies ist das Handle eines geöffneten Moduls

ch=Gibt die Nummer des Eingangs an, ab dem gelesen werden soll (0, 32, 64, ..)

##### Return-Wert

Zustand der gelesenen Eingänge

##### Programmierbeispiel

```
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();
```



#### 4.4.5. DapiDIGet64

##### Beschreibung

Dieser Befehl liest gleichzeitig 64 digitale Eingänge.

##### Definition

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

##### Parameter

handle=Dies ist das Handle eines geöffneten Moduls

ch=Gibt die Nummer des Eingangs an, ab dem gelesen werden soll (0, 64, ..)

##### Return-Wert

Zustand der gelesenen Eingänge

#### 4.4.6. DapiDIGetFF32

##### Beschreibung

Dieser Befehl liest die Flip-Flops der Eingänge aus und setzt diese zurück (Eingangszustands-Änderung).

##### Definition

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

##### Parameter

handle=Dies ist das Handle eines geöffneten Moduls

ch=Gibt die Nummer des Eingangs an, ab dem gelesen werden soll (0, 32, 64, ..)

##### Return-Wert

Zustand von 32 Eingangszustandsänderungen

#### 4.4.7. DapiDIGetCounter

##### Beschreibung

Dieser Befehl liest den Eingangszähler eines digitalen Eingangs.

##### Definition

*ULONG DapiDIGetCounter(ULONG handle, ULONG ch, ULONG mode);*

##### Parameter

handle=Dies ist das Handle eines geöffneten Moduls

ch=Gibt die Nummer des Eingangs an, ab dem gelesen werden soll

mode=0 (Normale Zählfunktion)

mode=DAPI\_CNT\_MODE\_READ\_WITH\_RESET (Zähler auslesen und direktes Counter resettet)

mode=DAPI\_CNT\_MODE\_READ\_LATCHED (Auslesen des gespeicherten Zählerstandes)

##### Return-Wert

Angabe des Zählerwertes

##### Programmierbeispiel

```
value = DapiDIGetCounter(handle, 0 , 0);  
// Zähler von DI Chan 0 wird gelesen  
  
value = DapiDIGetCounter(handle, 1 , 0);  
// Zähler von DI Chan 1 wird gelesen  
  
value = DapiDIGetCounter(handle, 8 ,0);  
// Zähler von DI Chan 8 wird gelesen  
  
value = DapiDIGetCounter(handle, 0 ,  
DAPI_CNT_MODE_READ_WITH_RESET);  
// Zähler von DI Chan 0 wird gelesen UND resettet  
  
value = DapiDIGetCounter(handle, 1 ,  
DAPI_CNT_MODE_READ_LATCHED);  
// Auslesen des gespeicherten Zählerstandes von DI Chan 1
```

#### 4.4.8. DapiSpecialCounterLatchAll

##### Beschreibung

Dieser Befehl speichert die Zählerstände aller Eingangszähler gleichzeitig in ein Zwischenspeicher (Latch).

So können anschließend alle Zählerstände des Latches nacheinander ausgelesen werden.

Besonderheit hierbei ist, dass ein gleichzeitiges "Einfrieren" der Zählerstände möglich ist und die eingefrorenen Stände (Latch) dann einzeln nacheinander ausgelesen werden können.

##### Definition

```
void DapiSpecialCommand(ULONG handle, DAPI_SPECIAL_CMD_COUNTER,  
DAPI_SPECIAL_COUNTER_LATCH_ALL, 0, 0);
```

##### Parameter

Keine

##### Return-Wert

Keiner

##### Bemerkung

Module, die von diesen Befehlen unterstützt werden, können Sie unserer **DELIB Übersichtstabelle** entnehmen.

##### Programmierbeispiel

```
DapiSpecialCommand(ULONG handle,  
DAPI_SPECIAL_CMD_COUNTER,  
DAPI_SPECIAL_COUNTER_LATCH_ALL, 0, 0);
```

#### 4.4.9. DapiSpecialCounterLatchAllWithReset

##### Beschreibung

Dieser Befehl speichert die Zählerstände aller Eingangszähler gleichzeitig in einen Zwischenspeicher (Latch).

Zusätzlich werden die Zählerstände der Eingangszähler im Anschluss zurückgesetzt.

##### Definition

```
void DapiSpecialCommand(ULONG handle, DAPI_SPECIAL_CMD_COUNTER,  
DAPI_SPECIAL_COUNTER_LATCH_ALL_WITH_RESET, 0, 0);
```

##### Parameter

Keine

##### Return-Wert

Keiner

##### Bemerkung

Module, die von diesen Befehlen unterstützt werden, können Sie unserer

**DELIB Übersichtstabelle** entnehmen.

##### Programmierbeispiel

```
DapiSpecialCommand(ULONG handle,  
DAPI_SPECIAL_CMD_COUNTER,  
DAPI_SPECIAL_COUNTER_LATCH_ALL_WITH_RESET, 0, 0);
```

#### 4.4.10. DapiSpecialDIFilterValueSet

##### Beschreibung

Dieser Befehl setzt einen Eingansfilter in [ms], in welchem Zeitintervall Störimpulse bei digitalen Eingangskanälen, gefiltert werden.

##### Definition

*DapiSpecialCommand(handle, DAPI\_SPECIAL\_CMD\_DI,  
DAPI\_SPECIAL\_DI\_FILTER\_VALUE\_SET, ULONG time\_ms, 0);*

##### Parameter

handle=Dies ist das Handle eines geöffneten Moduls

time\_ms=Zeitintervall [ms], indem digitale Eingangskanäle gelesen werden.

##### Bemerkung

Standardwert: 0ms

Wertebereich: 0(=off) , 1(ms) - 254(ms)

Module, die von diesen Befehlen unterstützt werden, können Sie unserer **DELIB Übersichtstabelle** entnehmen.

##### Programmierbeispiel

```
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_DI,  
DAPI_SPECIAL_DI_FILTER_VALUE_SET, 5, 0);  
// Setzt das Zeitintervall auf 5ms
```

```
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_DI,  
DAPI_SPECIAL_DI_FILTER_VALUE_SET, 150, 0);  
// Setzt das Zeitintervall auf 150ms
```

#### 4.4.11. DapiSpecialDIFilterValueGet

##### Beschreibung

Dieser Befehl gibt den vorher festgelegten Wert des Zeitintervalls zur Filterung von Störimpulsen bei digitalen Eingangskanäle in [ms] zurück.

##### Definition

*ULONG DapiSpecialCommand(handle, DAPI\_SPECIAL\_CMD\_DI,  
DAPI\_SPECIAL\_DI\_FILTER\_VALUE\_GET, 0, 0);*

##### Parameter

handle=Dies ist das Handle eines geöffneten Moduls

##### Return-Wert

Zeit [ms]

##### Bemerkung

Module, die von diesen Befehlen unterstützt werden, können Sie unserer

**DELIB Übersichtstabelle** entnehmen.

##### Programmierbeispiel

```
value = DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_DI,  
DAPI_SPECIAL_DI_FILTER_VALUE_GET, 0, 0);  
//Gibt das Zeitintervall zum Auslesen der digitalen Eingangskanäle zurück.
```

#### 4.4.12. Dapi\_Special\_DI\_FF\_Filter\_Value\_Set

##### Beschreibung

Dieser Befehl setzt einen Filter [ms], in welchem Zeitintervall die Eingangs-Flip-Flops und die Eingangs-Zähler, abgefragt werden.

##### Definition

*DapiSpecialCommand(handle, DAPI\_SPECIAL\_CMD\_DI,  
DAPI\_SPECIAL\_DI\_FF\_FILTER\_VALUE\_SET, ULONG time\_ms, 0);*

##### Parameter

handle=Dies ist das Handle eines geöffneten Moduls

time\_ms=Zeitintervall [ms], indem digitale Eingangskanäle abgetastet werden.

##### Bemerkung

Dieser Befehl unterstützt nur Impulszeiten zwischen 5ms und 255ms.

Wird keine Zeit gesetzt, ist der Default-Wert 100ms.

Module, die von diesen Befehlen unterstützt werden, können Sie unserer **DELIB Übersichtstabelle** entnehmen.

##### Programmierbeispiel

```
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_DI,  
DAPI_SPECIAL_DI_FF_FILTER_VALUE_SET, 5, 0);  
// Setzt das Zeitintervall auf 5ms  
  
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_DI,  
DAPI_SPECIAL_DI_FF_FILTER_VALUE_SET, 150, 0);  
// Setzt das Zeitintervall auf 150ms
```



#### 4.4.13. Dapi\_Special\_DI\_FF\_Filter\_Value\_Get

##### Beschreibung

Dieser Befehl gibt den vorher festgelegten Wert des Zeitintervalls zur Abtastung der Eingangs-Flip-Flops und der Eingangs-Zähler in [ms] zurück.

##### Definition

*ULONG DapiSpecialCommand(handle, DAPI\_SPECIAL\_CMD\_DI,  
DAPI\_SPECIAL\_DI\_FF\_FILTER\_VALUE\_GET, 0, 0);*

##### Parameter

handle=Dies ist das Handle eines geöffneten Moduls

##### Return-Wert

Zeit [ms]

##### Bemerkung

Module, die von diesen Befehlen unterstützt werden, können Sie unserer

**DELIB Übersichtstabelle** entnehmen.

##### Programmierbeispiel

```
value = DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_DI,  
DAPI_SPECIAL_DI_FF_FILTER_VALUE_GET, 0, 0);  
//Gibt das Zeitintervall zum Abtasten der digitalen Eingangskanäle zurück.
```

## 4.5. Digitale Ausgänge verwalten

### 4.5.1. DapiDOSet1

#### Beschreibung

Dieser Befehl setzt einen einzelnen Ausgang.

#### Definition

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

#### Parameter

handle=Dies ist das Handle eines geöffneten Moduls

ch=Gibt die Nummer des zu setzenden Ausgangs an (0 .. )

data=Gibt den Datenwert an, der geschrieben wird (0 / 1)

#### Return-Wert

Keiner

### 4.5.2. DapiDOSet8

#### Beschreibung

Dieser Befehl setzt gleichzeitig 8 digitale Ausgänge.

#### Definition

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

#### Parameter

handle=Dies ist das Handle eines geöffneten Moduls

ch=Gibt die Nummer des Ausgangs an, ab dem geschrieben werden soll (0, 8, 16, 24, 32, ..)

data=Gibt die Datenwerte an, die geschrieben werden

#### Return-Wert

Keiner

### 4.5.3. DapiDOSet16

#### Beschreibung

Dieser Befehl setzt gleichzeitig 16 digitale Ausgänge.

#### Definition

```
void DapiDOSet16(ULONG handle, ULONG ch, ULONG data);
```

#### Parameter

handle=Dies ist das Handle eines geöffneten Moduls

ch=Gibt die Nummer des Ausgangs an, ab dem geschrieben werden soll (0, 16, 32, ..)

data=Gibt die Datenwerte an, die geschrieben werden

#### Return-Wert

Keiner

#### 4.5.4. DapiDOSet32

##### Beschreibung

Dieser Befehl setzt gleichzeitig 32 digitale Ausgänge.

##### Definition

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

##### Parameter

handle=Dies ist das Handle eines geöffneten Moduls

ch=Gibt die Nummer des Ausgangs an, ab dem geschrieben werden soll (0, 32, 64, ..)

data=Gibt die Datenwerte an, die geschrieben werden

##### Return-Wert

Keiner

##### Programmierbeispiel

```
// 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();
```

#### 4.5.5. DapiDOSet64

##### Beschreibung

Dieser Befehl setzt gleichzeitig 64 digitale Ausgänge.

##### Definition

```
void DapiDOSet64(ULONG handle, ULONG ch, ULONG data);
```

##### Parameter

handle=Dies ist das Handle eines geöffneten Moduls

ch=Gibt die Nummer des Ausgangs an, ab dem geschrieben werden soll (0, 64, ..)

data=Gibt die Datenwerte an, die geschrieben werden

##### Return-Wert

Keiner

#### 4.5.6. DapiDOSet1\_WithTimer

##### Beschreibung

Diese Funktion setzt einen Digitalausgang (ch) auf einen Wert (data - 0 oder 1) für eine bestimmte Zeit in ms.

##### Definition

```
void DapiDOSet1_WithTimer(ULONG handle, ULONG ch, ULONG data, ULONG time_ms);
```

##### Parameter

handle=Dies ist das Handle eines geöffneten Moduls

ch=Gibt die Nummer des Ausgangs an, ab dem geschrieben werden soll (0, 32, 64, ..)

data=Gibt die Datenwerte an, die geschrieben werden

time\_ms=Gibt die Zeit an, in der der Ausgang gesetzt wird [ms]

##### Return-Wert

Keiner

##### Bemerkung:

Dieser Befehl wird nur von unserem RO-O8-R8 Modul unterstützt.

Dieser Befehl verliert seine Gültigkeit, sofern er mit anderen Werten überschrieben wird.

Möchte man den Befehl deaktivieren, dann muss er mit time\_ms=0 überschrieben werden.

Module, die von diesen Befehlen unterstützt werden, können Sie unserer

**DELIB Übersichtstabelle** entnehmen.

##### Programmierbeispiel

```
DapiDOSet1_WithTimer(handle, 2, 1, 1000);  
//Setting channel 2 for 1000msec to 1
```

#### 4.5.7. DapiDOReadback32

##### Beschreibung

Dieser Befehl liest die 32 digitalen Ausgänge zurück.

##### Definition

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

##### Parameter

handle=Dies ist das Handle eines geöffneten Moduls

ch=Gibt die Nummer des Ausgangs an, ab dem zurückgelesen werden soll (0, 32, 64, ..)

##### Return-Wert

Zustand von 32 Ausgängen.

#### 4.5.8. DapiDOReadback64

##### Beschreibung

Dieser Befehl liest die 64 digitalen Ausgänge zurück.

##### Definition

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

##### Parameter

handle=Dies ist das Handle eines geöffneten Moduls

ch=Gibt die Nummer des Ausgangs an, ab dem zurückgelesen werden soll (0, 64, ..)

##### Return-Wert

Zustand von 64 Ausgängen.

#### 4.5.9. DapiDOSetBit32

##### Beschreibung

Mit diesem Befehl können Ausgänge gezielt auf 1 geschaltet werden, ohne die Zustände der benachbarten Ausgänge zu ändern.

##### Definition

*void DapiDOSetBit32(uint handle, uint ch, uint data);*

##### Parameter

handle = Dies ist das Handle eines geöffneten Moduls

ch = Gibt die Nummer des Ausgangs an, ab dem geschrieben werden soll

data = Gibt den Datenwert an, der geschrieben werden soll (bis zu 32 Bit)

##### Return-Wert

Keiner

##### Bemerkung:

Nur die Bits mit einer Wertigkeit von 1 im data Parameter werden vom Befehl berücksichtigt.

##### Programmierbeispiel

```
data = 0x1; // Ausgang 0 wird auf 1 gesetzt, der Zustand von Ausgang
1-31 bleibt unberührt
DapiDOSetBit32(handle, 0, data);
data = 0xf; // Ausgang 0-3 wird auf 1 gesetzt, der Zustand von Ausgang
4-31 bleibt unberührt
DapiDOSetBit32(handle, 0, data);
data = 0xff; // Ausgang 0-7 wird auf 1 gesetzt, der Zustand von
Ausgang 8-31 bleibt unberührt
DapiDOSetBit32(handle, 0, data);
data = 0xff000000; // Ausgang 23-31 wird auf 1 gesetzt, der Zustand
von Ausgang 0-22 bleibt unberührt
DapiDOSetBit32(handle, 0, data);
```



#### 4.5.10. DapiDOClrBit32

##### Beschreibung

Mit diesem Befehl können Ausgänge gezielt auf 0 geschaltet werden, ohne die Zustände der benachbarten Ausgänge zu ändern.

##### Definition

*void DapiDOClrBit32(uint handle, uint ch, uint data);*

##### Parameter

handle = Dies ist das Handle eines geöffneten Moduls

ch = Gibt die Nummer des Ausgangs an, ab dem geschrieben werden soll

data = Gibt den Datenwert an, der geschrieben werden soll (bis zu 32 Bit)

##### Return-Wert

Keiner

##### Bemerkung:

Nur die Bits mit einer Wertigkeit von 1 im data Parameter werden vom Befehl berücksichtigt.

##### Programmierbeispiel

```
data = 0x1; // Ausgang 0 wird auf 0 gesetzt, der Zustand von Ausgang
1-31 bleibt unberührt
DapiDOSetBit32(handle, 0, data);
data = 0xf; // Ausgang 0-3 wird auf 0 gesetzt, der Zustand von Ausgang
4-31 bleibt unberührt
DapiDOSetBit32(handle, 0, data);
data = 0xff; // Ausgang 0-7 wird auf 0 gesetzt, der Zustand von
Ausgang 8-31 bleibt unberührt
DapiDOSetBit32(handle, 0, data);
data = 0xff000000; // Ausgang 23-31 wird auf 0 gesetzt, der Zustand
von Ausgang 0-22 bleibt unberührt
DapiDOSetBit32(handle, 0, data);
```

## 4.6. TTL-Ein-/Ausgangs Richtungen setzen mit DapiSpecialCommand

### 4.6.1. DAPI\_SPECIAL\_CMD\_SET\_DIR\_DX\_1

#### Beschreibung

Dieser Befehl setzt die Richtung von 8 hintereinanderliegenden TTL-Ein/Ausgängen (1-Bit weise).

#### Definition

```
void DapiSpecialCommand(ULONG handle,  
DAPI_SPECIAL_CMD_SET_DIR_DX_1, ULONG ch, ULONG dir, 0);
```

#### Parameter

handle = Dies ist das Handle eines geöffneten Moduls

ch = Muss immer 0 sein!

dir = Gibt die Richtung für 8 Kanäle an (1=output / 0=input) / Bit 0 steht für Kanal 0, Bit 1 für Kanal 1 ...

#### Bemerkung

Nicht kompatibel mit USB-TTL-32/64.

Verwenden Sie für diese Module den DAPI\_SPECIAL\_CMD\_SET\_DIR\_DX\_8 Befehl.

#### Programmierbeispiel

```
DapiSpecialCommand(handle,  
DAPI_SPECIAL_CMD_SET_DIR_DX_1, 0, 0x01 , 0);  
// Set Dir of TTL-I/O CH0 to output, others to input  
DapiSpecialCommand(handle,  
DAPI_SPECIAL_CMD_SET_DIR_DX_1, 0, 0x02 , 0);  
// Set Dir of TTL-I/O CH1 to output, others to input  
DapiSpecialCommand(handle,  
DAPI_SPECIAL_CMD_SET_DIR_DX_1, 0, 0x04 , 0);  
// Set Dir of TTL-I/O CH2 to output, others to input  
DapiSpecialCommand(handle,  
DAPI_SPECIAL_CMD_SET_DIR_DX_1, 0, 0x08 , 0);  
// Set Dir of TTL-I/O CH3 to output, others to input  
DapiSpecialCommand(handle,  
DAPI_SPECIAL_CMD_SET_DIR_DX_1, 0, 0x10 , 0);  
// Set Dir of TTL-I/O CH4 to output, others to input
```

```
DapiSpecialCommand(handle,  
DAPI_SPECIAL_CMD_SET_DIR_DX_1, 0, 0x20 , 0);  
// Set Dir of TTL-I/O CH5 to output, others to input  
DapiSpecialCommand(handle,  
DAPI_SPECIAL_CMD_SET_DIR_DX_1, 0, 0x40 , 0);  
// Set Dir of TTL-I/O CH6 to output, others to input  
DapiSpecialCommand(handle,  
DAPI_SPECIAL_CMD_SET_DIR_DX_1, 0, 0x80 , 0);  
// Set Dir of TTL-I/O CH7 to output, others to input  
  
DapiSpecialCommand(handle,  
DAPI_SPECIAL_CMD_SET_DIR_DX_1, 0, 0x0f , 0);  
// Set Dir of TTL-I/O CH0-3 to output, others to input  
DapiSpecialCommand(handle,  
DAPI_SPECIAL_CMD_SET_DIR_DX_1, 0, 0xff , 0);  
// Set Dir of TTL-I/O CH0-7 to output, others to input
```

#### 4.6.2. DAPI\_SPECIAL\_CMD\_SET\_DIR\_DX\_8

##### Beschreibung

Dieser Befehl setzt die Richtung von bis zu 64 hintereinanderliegenden TTL-Ein/Ausgängen (8-Bit weise).

1-Bit repräsentiert dabei 8 TTL-Ein/Ausgänge.

##### Definition

```
void DapiSpecialCommand(ULONG handle,  
DAPI_SPECIAL_CMD_SET_DIR_DX_8, ULONG ch, ULONG dir, 0);
```

##### Parameter

handle = Dies ist das Handle eines geöffneten Moduls

ch = Muss immer 0 sein!

dir = (8-Bit) gibt die Richtung für bis zu 64 hintereinanderliegende TTL-Ein/Ausgänge an. (1=output / 0=input)

##### Bemerkung

Nur kompatibel mit USB-TTL-32/64.

Verwenden Sie für andere TTL-Produkte den DAPI\_SPECIAL\_CMD\_SET\_DIR\_DX\_1 Befehl.

##### Programmierbeispiel

```
DapiSpecialCommand(handle,  
DAPI_SPECIAL_CMD_SET_DIR_DX_8, 0, 0x1 , 0);  
// Set Dir of TTL-I/O CH0-7 to output, others to input  
DapiSpecialCommand(handle,  
DAPI_SPECIAL_CMD_SET_DIR_DX_8, 0, 0x3 , 0);  
// Set Dir of TTL-I/O CH0-15 to output, others to input  
DapiSpecialCommand(handle,  
DAPI_SPECIAL_CMD_SET_DIR_DX_8, 0, 0xc , 0);  
// Set Dir of TTL-I/O CH16-31 to output, others to input  
DapiSpecialCommand(handle,  
DAPI_SPECIAL_CMD_SET_DIR_DX_8, 0, 0x33 , 0);  
// Set Dir of TTL-I/O CH0-15 and CH32-47 to output, others to input  
DapiSpecialCommand(handle,  
DAPI_SPECIAL_CMD_SET_DIR_DX_8, 0, 0xff , 0);  
// Set Dir of TTL-I/O CH0-63 to output, others to input
```

## 4.7. Ausgabe-Timeout verwalten

### 4.7.1. DapiSpecialCMDTimeout

#### Beschreibung

Dieser Befehl dient zum Einstellen der Timeout-Schutz-Funktion.

#### Definition

```
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_TIMEOUT, cmd, par1, par2);
```

#### Parameter

handle=Dies ist das Handle eines geöffneten Moduls

cmd = auszuführende Funktion

par1 = Wert, der an die Funktion übergeben wird

par2 = Wert, der an die Funktion übergeben wird

#### 4.7.1.1. DapiSpecialTimeoutSetValueSec

##### Beschreibung

Dieser Befehl dient zum Setzen der Timeout-Zeit.

##### Definition

*DapiSpecialCommand(handle, DAPI\_SPECIAL\_CMD\_TIMEOUT, cmd, par1, par2);*

##### Parameter

cmd = DAPI\_SPECIAL\_TIMEOUT\_SET\_VALUE\_SEC

par1 = Sekunden [s]

par2 = Millisekunden [100ms] (Wert 6 = 600ms)

##### Bemerkung

Der zulässige Wertebereich der Zeitangabe liegt zwischen 0,1 Sekunden und 6553 Sekunden

##### Programmierbeispiel

```
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_TIMEOUT,  
DAPI_SPECIAL_TIMEOUT_SET_VALUE_SEC, 3, 7);  
//Die Zeit des Timeouts wird auf 3,7sek gesetzt.
```

#### 4.7.1.2. DapiSpecialTimeoutActivate

##### Beschreibung

Dieser Befehl aktiviert den "normalen" Timeout.

##### Definition

DapiSpecialCommand(handle, DAPI\_SPECIAL\_CMD\_TIMEOUT, cmd, 0, 0);

##### Parameter

cmd = DAPI\_SPECIAL\_TIMEOUT\_ACTIVATE

##### Programmierbeispiel

```
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_TIMEOUT,  
DAPI_SPECIAL_TIMEOUT_ACTIVATE, 0, 0);  
//Der "normale" Timeout wird aktiviert.
```

#### 4.7.1.3. DapiSpecialTimeoutDeactivate

##### Beschreibung

Dieser Befehl deaktiviert den Timeout.

##### Definition

*DapiSpecialCommand(handle, DAPI\_SPECIAL\_CMD\_TIMEOUT, cmd, 0, 0);*

##### Parameter

cmd = DAPI\_SPECIAL\_TIMEOUT\_DEACTIVATE

##### Programmierbeispiel

```
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_TIMEOUT,  
DAPI_SPECIAL_TIMEOUT_DEACTIVATE, 0, 0);  
//Der Timeout wird deaktiviert.
```



#### 4.7.1.4. DapiSpecialTimeoutGetStatus

##### Beschreibung

Dieser Befehl dient zum Auslesen des Timeout-Status.

##### Definition

*ULONG DapiSpecialCommand(handle, DAPI\_SPECIAL\_CMD\_TIMEOUT,  
DAPI\_SPECIAL\_TIMEOUT\_GET\_STATUS, 0, 0);*

##### Parameter

cmd = DAPI\_SPECIAL\_TIMEOUT\_GET\_STATUS

##### Return-Wert

Return = 0 (Timeout ist deaktiviert)

##### Werte für den "normalen" Timeout

Return = 1 (Timeout "normal" ist aktiviert)

Return = 2 (Timeout "normal" hat stattgefunden)

##### Programmierbeispiel

```
unsigned    long    status    =    DapiSpecialCommand(handle,  
DAPI_SPECIAL_CMD_TIMEOUT,  
DAPI_SPECIAL_TIMEOUT_GET_STATUS, 0, 0);  
printf("Status = %lu\n", status);  
//Abfrage des Timeout-Status mit Ausgabe.
```

#### 4.7.1.5. DapiSpecialTimeoutDoValueLoadDefault

##### Beschreibung

Setzt die SET- und CLR-Werte auf den Default-Wert zurück.

(SET-Wert = 0, CLR-Wert = FFFFFFFF)

##### Definition

*DapiSpecialCommand(handle, DAPI\_SPECIAL\_CMD\_TIMEOUT, cmd, 0, 0);*

##### Parameter

cmd = DAPI\_SPECIAL\_TIMEOUT\_DO\_VALUE\_LOAD\_DEFAULT

##### Programmierbeispiel

```
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_TIMEOUT,  
DAPI_SPECIAL_TIMEOUT_DO_VALUE_LOAD_DEFAULT, 0, 0);  
//SET- und CLR-Werte werden auf den Default-Wert gesetzt.
```

## 4.8. Testfunktionen

### 4.8.1. DapiPing

#### Beschreibung

Dieser Befehl prüft die Verbindung zu einem geöffneten Modul.

#### Definition

*ULONG DapiPing(ULONG handle, ULONG value);*

#### Parameter

handle=Dies ist das Handle eines geöffneten Moduls

value=Übergebener Testwert, im Wertebereich von 0-255 (8-Bit), an das Modul

#### Return-Wert

Hier muß der mit "value" übergebene Testwert zurückkommen

## 4.9. Register Schreib-Befehle

### 4.9.1. DapiWriteByte

#### Beschreibung

Dieser Befehl führt einen direkten Register Schreibbefehl auf das Modul aus.

#### Definition

*void DapiWriteByte(ULONG handle, ULONG adress, ULONG value);*

#### Parameter

handle=Dies ist das Handle eines geöffneten Moduls

adress=Adresse, auf die zugegriffen werden soll

value=Gibt den Datenwert an, der geschrieben wird (8 Bit)

#### Return-Wert

Keiner

#### Bemerkung

Dies sollte nur von erfahrenen Programmieren benutzt werden. So kann auf alle zur Verfügung stehenden Register direkt zugegriffen werden.

### 4.9.2. DapiWriteWord

#### Beschreibung

Dieser Befehl führt einen direkten Register Schreibbefehl auf das Modul aus.

#### Definition

*void DapiWriteWord(ULONG handle, ULONG adress, ULONG value);*

#### Parameter

handle=Dies ist das Handle eines geöffneten Moduls

adress=Adresse, auf die zugegriffen werden soll

value=Gibt den Datenwert an, der geschrieben wird (16 Bit)

#### Return-Wert

Keiner

#### Bemerkung

Dies sollte nur von erfahrenen Programmieren benutzt werden. So kann auf alle zur Verfügung stehenden Register direkt zugegriffen werden.

### 4.9.3. DapiWriteLong

#### Beschreibung

Dieser Befehl führt einen direkten Register Schreibbefehl auf das Modul aus.

#### Definition

*void DapiWriteLong(ULONG handle, ULONG adress, ULONG value);*

#### Parameter

handle=Dies ist das Handle eines geöffneten Moduls

adress=Adresse, auf die zugegriffen werden soll

value=Gibt den Datenwert an, der geschrieben wird (32 Bit)

#### Return-Wert

Keiner

#### Bemerkung

Dies sollte nur von erfahrenen Programmieren benutzt werden. So kann auf alle zur Verfügung stehenden Register direkt zugegriffen werden.

#### 4.9.4. DapiWriteLongLong

##### Beschreibung

Dieser Befehl führt einen direkten Register Schreibbefehl auf das Modul aus.

##### Definition

*void DapiWriteLongLong(ULONG handle, ULONG adress, ULONGLONG value);*

##### Parameter

handle=Dies ist das Handle eines geöffneten Moduls

adress=Adresse, auf die zugegriffen werden soll

value=Gibt den Datenwert an, der geschrieben wird (64 Bit)

##### Return-Wert

Keiner

##### Bemerkung

Dies sollte nur von erfahrenen Programmieren benutzt werden. So kann auf alle zur Verfügung stehenden Register direkt zugegriffen werden.

## 4.10. Register Lese-Befehle

### 4.10.1. DapiReadByte

#### Beschreibung

Dieser Befehl führt einen direkten Register Lese-Befehl auf das Modul aus.

#### Definition

*ULONG DapiReadByte(ULONG handle, ULONG adress);*

#### Parameter

handle=Dies ist das Handle eines geöffneten Moduls

adress=Adresse, auf die zugegriffen werden soll

#### Return-Wert

Inhalt des zu lesenden Registers (8 Bit)

#### Bemerkung

Dies sollte nur von erfahrenen Programmieren benutzt werden. So kann auf alle zur Verfügung stehenden Register direkt zugegriffen werden.



### 4.10.2. DapiReadWord

#### Beschreibung

Dieser Befehl führt einen direkten Register Lese-Befehl auf das Modul aus.

#### Definition

*ULONG DapiReadWord(ULONG handle, ULONG adress);*

#### Parameter

handle=Dies ist das Handle eines geöffneten Moduls

adress=Adresse, auf die zugegriffen werden soll

#### Return-Wert

Inhalt des zu lesenden Registers (16 Bit)

#### Bemerkung

Dies sollte nur von erfahrenen Programmieren benutzt werden. So kann auf alle zur Verfügung stehenden Register direkt zugegriffen werden.

### 4.10.3. DapiReadLong

#### Beschreibung

Dieser Befehl führt einen direkten Register Lese-Befehl auf das Modul aus.

#### Definition

*ULONG DapiReadLong(ULONG handle, ULONG adress);*

#### Parameter

handle=Dies ist das Handle eines geöffneten Moduls

adress=Adresse, auf die zugegriffen werden soll

#### Return-Wert

Inhalt des zu lesenden Registers (32 Bit)

#### Bemerkung

Dies sollte nur von erfahrenen Programmieren benutzt werden. So kann auf alle zur Verfügung stehenden Register direkt zugegriffen werden.

#### 4.10.4. DapiReadLongLong

##### Beschreibung

Dieser Befehl führt einen direkten Register Lese-Befehl auf das Modul aus.

##### Definition

*ULONGLONG DapiReadLongLong(ULONG handle, ULONG adress);*

##### Parameter

handle=Dies ist das Handle eines geöffneten Moduls

adress=Adresse, auf die zugegriffen werden soll

##### Return-Wert

Inhalt des zu lesenden Registers (64 Bit)

##### Bemerkung

Dies sollte nur von erfahrenen Programmieren benutzt werden. So kann auf alle zur Verfügung stehenden Register direkt zugegriffen werden.

## 4.11. Programmier-Beispiel

```
// *****
// *****
//
// (c) DEDITEC GmbH, 2009
//
// web: http://www.deditec.de
//
// mail: vertrieb@deditec.de
//
// dtapi_prog_beispiel_input_output.cpp
//
// *****
// *****
//
// Folgende Bibliotheken beim Linken mit einbinden: delib.lib
// Dies bitte in den Projekteinstellungen
// (Projekt/Einstellungen/Linker(Objekt-
// Bibliothek-Module) .. letzter Eintrag konfigurieren
#include <windows.h>
#include <stdio.h>
#include "conio.h"
#include "delib.h"
// *****
// *****

void main(void)
{
    unsigned long handle;
    unsigned long data;
    unsigned long anz;
    unsigned long i;
    unsigned long chan;
    // -----
    // USB-Modul öffnen
    handle = DapiOpenModule(USB_Interface8,0);
    printf("USB_Interface8 handle = %x\n", handle);
    if (handle==0)
    {
        // USB Modul wurde nicht gefunden
        printf("Modul konnte nicht geöffnet werden\n");
        printf("TASTE für weiter\n");
        getch();
        return;
    }
    // Zum Testen - ein Ping senden
    // -----
    printf("PING\n");
    anz=10;
    for(i=0;i!=anz;++i)
    {
        data=DapiPing(handle, i);
        if(i==data)
        {
            // OK
            printf(".");
        }
        else
        {

```

```

// No answer
printf("E");
}
}
printf("\n");

// -----
// Einen Wert auf die Ausgänge schreiben
data = 255;
DapiWriteByte(handle, 0, data);
printf("Schreibe auf Adresse=0 daten=0x%x\n", data);
// -----
// Einen Wert auf die Ausgänge schreiben
data = 255;
DapiWriteByte(handle, 1, data);
printf("Schreibe auf Adresse=0 daten=0x%x\n", data);
// -----
// Einen Wert auf die Ausgänge schreiben
data = 255;
DapiWriteByte(handle, 2, data);
printf("Schreibe auf Adresse=2 daten=0x%x\n", data);
// -----
// Einen Wert von den Eingängen lesen
data = (unsigned long) DapiReadByte(handle, 0);
printf("Gelesene Daten = 0x%x\n", data);
// -----
// Einen A/D Wert lesen
chan=11; // read chan. 11
data = DapiReadWord(handle, 0xff010000 + chan*2);
printf("Adress=%x, ret=%x volt=%f\n", chan, data, ((float) data) / 1024*5); //
Bei 5 Volt Ref
// -----
// Modul wieder schliessen
DapiCloseModule(handle);
printf("TASTE für weiter\n");
getch();
return ;
}

```

## 4.12. Delib Übersichtstabelle

Befehle	Verfügbar für
DAPI_SPECIAL_CMD_SET_DIR_DX_1	USB-MINI-TTL8
DAPI_SPECIAL_CMD_SET_DIR_DX_8	USB-MINI-TTL8 USB-TTL32 USB-TTL64 ETH-TTL64
DAPI_SPECIAL_CMD_GET_DIR_DX_1	wird nicht unterstützt
DAPI_SPECIAL_CMD_GET_DIR_DX_8	wird nicht unterstützt

Befehle	Verfügbar für	Geht nicht bei
DAPI_SPECIAL_CMD_TIMEOUT DAPI_SPECIAL_TIMEOUT_DO_VALUE_MASK_WR_SET32 DAPI_SPECIAL_TIMEOUT_DO_VALUE_MASK_RD_SET32 DAPI_SPECIAL_TIMEOUT_DO_VALUE_MASK_WR_CLR32 DAPI_SPECIAL_TIMEOUT_DO_VALUE_MASK_RD_CLR32 DAPI_SPECIAL_TIMEOUT_DO_VALUE_LOAD_DEFAULT	ETH-TTL64 ETH-RELAIS8 USB-RELAIS8 RO-SERIE BS-SERIE NET-SERIE USB-TTL-64	USB-Mini-Stick
DAPI_SPECIAL_TIMEOUT_SET_VALUE_SE DAPI_SPECIAL_TIMEOUT_ACTIVATE DAPI_SPECIAL_TIMEOUT_DEACTIVATE DAPI_SPECIAL_TIMEOUT_GET_STATUS	alle Module	

Befehl	Starter USB*	Starter ETH**	RO Serie	BS Serie	NET Serie	Sonstiges
DAPI_SPECIAL_COUNTER_ LATCH_ALL			x			
DAPI_SPECIAL_COUNTER_ LATCH_ALL_WITH_RESET			x			
DapiDOSet1_WithTimer			x			
DAPI_SPECIAL_CMD_SW_FIFO DAPI_SPECIAL_SW_FIFO_INIT_ AND_CLEAR ... DAPI_SPECIAL_SW_FIFO_ IO_DEACTIVATE					x	
DAPI_SPECIAL_CMD_AD DAPI_SPECIAL_RO_AD_ FIFO_ACTIVATE ... DAPI_SPECIAL_RO_AD_ FIFO_INIT			x			

\*: USB-OPTOIN8, USB-Mini-Stick, USB-TTL-64

\*\* : ETH-TTL64, ETH-OPTOIN8, ETH-RELAIS8

Befehl	Starter USB*	Starter ETH**	RO Serie	BS Serie	NET Serie	Sonstiges
DAPI_SPECIAL_DI_FF_FILTER DAPI_SPECIAL_DI_FF_FILTER_ VALUE_SET DAPI_SPECIAL_DI_FF_FILTER_ VALUE_GET	5-255	1-255	1-255	1-255	1-255	
DAPI_SPECIAL_DI_FILTER DAPI_SPECIAL_DI_FILTER_ VALUE_SET DAPI_SPECIAL_DI_FILTER_ VALUE_GET	x	0, 1-254	0, 1-254	0, 1-254	0, 1-254	
DAPI_SPECIAL_CMD_GET_ INTERNAL_STATISTIC	x	x	x	x		

\*: USB-OPTOIN8, USB-Mini-Stick, USB-TTL-64

\*\* : ETH-TTL64, ETH-OPTOIN8, ETH-RELAIS8

Befehle	Verfügbar für
DAPI_SPECIAL_CMDEXT_CAN_WR_RUNTIME_ _VALUE DAPI_SPECIAL_CMDEXT_CAN_RD_RUNTIME_ VALUE	NET-CPU-PRO, BS-WEU, RO-ETH-LC



## Verzeichnisstruktur der DELIB



## 5. Verzeichnisstruktur der DELIB

Nach erfolgreicher Installation liegt folgender Verzeichnisbaum vor:

\$DELIB\_DIR

- |
- | -> include Includes für Programmiersprachen
- | -> lib Library
- | -> lib\bc Borland Compiler Library
- | -> prod\_pics Produktbilder
- | -> programs Modul-Testprogramme
- + -> USB-Driver Treiber für USB-Module

Bitte beachten Sie, dass der "\$DELIB\_DIR" Ordner, je nach Betriebssystem und DELIB-Version, variieren kann.

DELIB Installation	Windows Installation	Pfad
32 Bit	32 Bit	C:\Programme\DEDITEC\DELIB\
32 Bit	64 Bit	C:\Programme (x86)\DEDITEC\DELIB\
64 Bit	64 Bit	C:\Programme\DEDITEC\DELIB64\

**Zudem werden im Windows System Ordner folgende Dateien installiert:**

\$SYSDIR\delib.dll, bzw. \$SYSDIR\delib64.dll (32 Bit, bzw. 64 Bit DELIB Version)

\$SYSDIR\delibJNI.dll, bzw. \$SYSDIR\delibJNI64.dll (32 Bit, bzw. 64 Bit DELIB Version)

\$SYSDIR\ftbusui.dll

\$SYSDIR\ftd2xx.dll

\$SYSDIR\FTLang.dll

\$SYSDIR\drivers\ftdibus.sys

**Bitte beachten Sie, dass der "\$SYSDIR" Ordner, je nach Betriebssystem und DELIB-Version, variieren kann.**

<b>DELIB Installation</b>	<b>Windows Installation</b>	<b>Pfad</b>
32 Bit	32 Bit	C:\Windows\System32
32 Bit	64 Bit	C:\Windows\SysWOW64
64 Bit	64 Bit	C:\Windows\System32

## 5.1. Include Verzeichnis

Das für die DELIB angelegte Include-Verzeichnis enthält die Dateien, welche die entsprechenden Library-Funktionen beschreiben. Diese sind für die Programmiersprachen C (.h), Delphi (.pas) und Visual Basic (.bas) gegeben.

## 5.2. Library-Verzeichnis

Hierin befindet sich die Datei "DELIB.lib". Sie dient als Bindeglied für das Compilieren von eigenen Programmen, die die "DELIB.dll" benutzen.

## 5.3. Library-Verzeichnis für Borland

Für Borland Compiler gibt es eine separate DELIB.lib, die sich im Unterverzeichnis "bc" befindet. Diese dient ebenfalls als Bindeglied für das Compilieren von eigenen Programmen, die die "DELIB.dll" benutzen.

## 5.4. Umgebungsvariablen

Zwei Umgebungsvariablen weisen auf wichtige Verzeichnisse hin, die Dateien für die Programmiersprachen C, Delphi und Visual Basic enthalten.

“DELIB\_INCLUDE” zeigt auf das Include-Verzeichnis.

%DELIB\_INCLUDE% à c:\Programme\DEDITEC\DELIB\include”

“DELIB\_LIB” zeigt auf das Library-Verzeichnis.

%DELIB\_LIB% à c:\ Programme\DEDITEC\DELIB\lib

# Anhang

---



## **6. Anhang**

### **6.1. Kontakt / Support**

Wenn Sie Fragen zum Produkt haben oder Unterstützung bei der Inbetriebnahme brauchen, erreichen Sie uns unter folgenden Rufnummern:

#### **Support Software**

Tel. +49 (0) 22 32 / 50 40 8 – 20

#### **Support Hardware**

Tel. +49 (0) 22 32 / 50 40 8 – 30

#### **Support via E-mail**

[support@deditec.de](mailto:support@deditec.de)

### **6.2. Umwelt und Entsorgung**

Sie können das defekte oder veraltete Produkt am Ende seiner Lebensdauer wieder an uns zurück senden. Als Hersteller und Vertreiber von Elektronikbaugruppen übernehmen wir für Sie die fachgerechte Entsorgung nach den geltenden gesetzlichen Bestimmungen. Nutzen Sie hierfür am besten unser Rücksendeformular auf der Homepage:

[Rücksendeformular](#)

## 6.3. Revisionen

Rev 3.00	DEDITEC Design Update
Rev 2.05	Kapitel "Software" und "DELIB API Referenz" überarbeitet
Rev 2.04	Index hinzugefügt
Rev 2.03	Kapitel "Firmware Update" hinzugefügt, Kapitel "Software" überarbeitet
Rev 2.02	Kapitel "USB-MINI-REL4" hinzugefügt
Rev 2.01	Neues Programmierbeispiel bei Befehl "DAPI_SPECIAL_CMD_SET_DIR_DX_1"
Rev 2.00	Designänderung
Rev 1.10	DELIB Configuration Utility / Digital Input/Output Sample
Rev 1.01	Überarbeitung der Blockschaltbilder
Rev 1.00	Erste DEDITEC Anleitung



## 6.4. Urheberrechte und Marken

Linux ist eine registrierte Marke von Linus Torvalds.

USB ist eine registrierte Marke von USB Implementers Forum Inc.

LabVIEW ist eine registrierte Marke von National Instruments.

Intel ist eine registrierte Marke von Intel Corporation.

AMD ist eine registrierte Marke von Advanced Micro Devices, Inc.

ProfiLab ist eine registrierte Marke von ABACOM Ingenieurbüro GbR.

ispVM System ist eine registrierte Marke von Lattice Semiconductor Corporation

Windows, Visual-C/C++, -C#, -Basic, -Basic.NET und Visual-Studio sind registrierte Marken von Microsoft Corporation.

Delphi ist eine registrierte Marke von Borland Software Corporation.

Java ist eine registrierte Marke von Oracle Corporation.