



# USB-RELAIS-8 / USB-OPTOIN-8

Hardware-Beschreibung

2022 Juli

# INDEX

<b><u>1. Einleitung</u></b>	<b>8</b>
1.1. Vorwort	9
1.2. Kundenzufriedenheit	9
1.3. Kundenresonanz	9
1.4. Kurzbeschreibung	10
1.5. Lieferumfang	11
<b><u>2. Inbetriebnahme</u></b>	<b>12</b>
2.1. Sicherheitshinweise	13
2.2. Schritt 1 - Anschluss der Stromversorgung	14
2.3. Schritt 2 - Verbinden mit dem PC oder Netzwerk	14
2.4. Schritt 3 - Installation der Software und Treiber	15
2.5. Schritt 4 - Funktionstest	16
<b><u>3. Hardware Beschreibung</u></b>	<b>17</b>
3.1. Allgemeine technische Daten	18
3.1.1. USB-RELAIS-8	18
3.1.2. USB-OPTOIN-8	19
3.2. Übersichtsbild	20
3.2.1. USB-RELAIS/OPTOIN-8	20
3.3. Schnittstellen	21
3.3.1. USB	21
3.4. LEDs	22
3.4.1. Definition der LEDs	22
3.5. Digitale Ausgänge	23
3.5.1. Technische Daten	23
3.5.2. Pinbelegung	24
3.5.3. Anschlussbeispiel	25
3.5.4. Timeout Funktion	26

# INDEX

3.6. Digitale Eingänge	27
3.6.1. Technische Daten	27
3.6.2. Pinbelegung	28
3.6.3. Anschlussbeispiel	29
3.6.4. Eingangsfiler	30
3.6.5. Zustandsänderungen überwachen	31
<b><u>4. Software</u></b>	<b>32</b>
4.1. Benutzung unserer Produkte	33
4.1.1. Ansteuerung über unsere DELIB Treiberbibliothek	33
4.1.2. Ansteuerung über mitgelieferte Testprogramme	33
4.1.3. Ansteuerung auf Protokollebene	33
4.1.4. DELIB CLI (command-line interface) für Windows	35
4.1.4.1. Konfiguration des DELIB CLI	38
4.1.4.2. DELIB CLI Beispiele	39
4.1.5. Ansteuerung über grafische Anwendungen	42
4.1.5.1. LabVIEW	42
4.1.5.2. Profilab	43
4.1.5.3. Licht24 Pro	44
4.1.6. Einbinden der DELIB in Programmiersprachen	45
4.1.6.1. Einbinden der DELIB in Visual-C/C++	45
4.1.6.2. Einbinden der DELIB in Visual-C/C++ (Visual Studio 2015)	48
4.1.6.3. Einbinden der DELIB in Visual-C#	51
4.1.6.4. Einbinden der DELIB in Delphi	52
4.1.6.5. Einbinden der DELIB in Visual-Basic (VB)	53
4.1.6.6. Einbinden der DELIB in Visual-Basic.NET (VB.NET)	54
4.1.6.7. Einbinden der DELIB in MS-Office (VBA)	55
4.1.6.8. Einbinden der DELIB in LabVIEW	57
4.1.6.8.1. Einbinden der DELIB in LabVIEW	57
4.1.6.8.2. Verwendung der VIs in LabVIEW	66
4.1.6.8.3. Setzen der Modul-ID in LabVIEW	69
4.1.6.9. Einbinden der DELIB in Java	71

# INDEX

4.2. DELIB Treiberbibliothek	72
4.2.1. Übersicht	73
4.2.1.1. Unterstützte Programmiersprachen	76
4.2.1.2. Unterstützte Betriebssysteme	77
4.2.1.3. SDK-Kit für Programmierer	78
4.2.2. DELIB Setup	79
4.2.3. DELIB Configuration Utility	85
4.2.3.1. Einführung	85
4.2.3.2. Neue Konfiguration erstellen oder vorhandene Konfiguration bearbeiten	86
4.2.3.2.1. Modul Konfiguration USB	87
4.2.3.2.1.1 Beispiel zur Konfiguration identischer USB-Module	89
4.2.3.3. Modul testen	95
4.2.3.4. Debug Optionen einstellen	99
4.2.4. Benutzung des Modulselectors	100
4.2.5. DELIB Module Config	106
4.2.5.1. Modul Konfigurationen	106
4.2.5.1.1. Modul-Infoseite	106
4.2.5.1.2. Modul-Identifikation	108
4.2.5.1.3. LAN Netzwerkinformationen	109
4.2.5.1.4. LAN Netzwerkeinstellungen	111
4.2.5.1.5. NTP-Konfiguration	112
4.2.5.1.6. I/O Kanal-Namen	115
4.2.5.2. I/O-Test	116
4.2.5.2.1. Timeout Test-Funktion	116
4.2.5.2.2. Digital In	117
4.2.5.2.3. Digital Out	118
4.2.6. DELIB Module Demo	120
4.2.6.1. Auswahl des Moduls	121
4.2.6.2. Allgemein	122
4.2.6.2.1. Module Info	124
4.2.6.3. Digital Input	125
4.2.6.4. Digital Output	126

# INDEX

4.2.7. DT-Flasher	127
4.2.7.1. Über DEDITEC-Firmware	128
4.2.7.2. Auswahl des Moduls	128
4.2.7.3. Firmware Update durchführen	130
4.2.7.3.1. Flash-Files manuell aktualisieren	132
4.3. DELIB Sample Sources (Windows Programmbeispiele)	133
4.3.1. Installation DELIB Sample Sources	134
4.3.2. Benutzung der DELIB Sample Sources	138
4.3.2.1. Schritt 1 - Produktauswahl	139
4.3.2.2. Schritt 2 - Kategorieauswahl	140
4.3.2.3. Schritt 3 - Programmiersprachenauswahl	141
4.3.2.4. Schritt 4 - Quellcode	142
4.4. DELIB für Linux	145
4.4.1. Verwenden der DELIB Treiberbibliothek für Linux	147
4.4.1.1. Delib USB-Sample in Linux	147
4.4.2. DELIB CLI (command-line interface) für Linux	150
4.4.2.1. Konfiguration des DELIB CLI	153
4.4.2.2. DELIB CLI Beispiele	156
<b><u>5. DELIB API Referenz</u></b>	<b>158</b>
5.1. Verwaltungsfunktionen	159
5.1.1. DapiOpenModule	159
5.1.2. DapiCloseModule	161
5.1.3. DapiGetDELIBVersion	162
5.1.4. DapiSpecialCMDGetModuleConfig	163
5.1.5. DapiOpenModuleEx	167
5.2. Fehlerbehandlung	170
5.2.1. DapiGetLastError	170
5.2.2. DapiGetLastErrorText	172
5.2.3. DapiClearLastError	173
5.2.4. DapiGetLastErrorByHandle	174
5.2.5. DapiClearLastErrorByHandle	175

# INDEX

5.3. Digitale Eingänge lesen	176
5.3.1. DapiDIGet1	176
5.3.2. DapiDIGet8	177
5.3.3. DapiDIGet16	178
5.3.4. DapiDIGet32	179
5.3.5. DapiDIGet64	180
5.3.6. DapiDIGetFF32	181
5.3.7. DapiDIGetCounter	182
5.3.8. DapiSpecialCounterLatchAll	183
5.3.9. DapiSpecialCounterLatchAllWithReset	184
5.3.10. DapiSpecialDIFilterValueSet	185
5.3.11. DapiSpecialDIFilterValueGet	186
5.3.12. Dapi_Special_DI_FF_Filter_Value_Get	187
5.3.13. Dapi_Special_DI_FF_Filter_Value_Set	188
5.4. Digitale Ausgänge verwalten	189
5.4.1. DapiDOSet1	189
5.4.2. DapiDOSet8	190
5.4.3. DapiDOSet16	191
5.4.4. DapiDOSet32	192
5.4.5. DapiDOSet64	193
5.4.6. DapiDOSet1_WithTimer	194
5.4.7. DapiDOReadback32	196
5.4.8. DapiDOReadback64	197
5.4.9. DapiDOSetBit32	198
5.4.10. DapiDOClrBit32	200
5.5. Ausgabe-Timeout verwalten	202
5.5.1. DapiSpecialCMDTimeout	202
5.5.1.1. DapiSpecialTimeoutSetValueSec	206
5.5.1.2. DapiSpecialTimeoutActivate	207
5.5.1.3. DapiSpecialTimeoutActivateAutoReactivate	208
5.5.1.4. DapiSpecialTimeoutActivateSecureOutputs	209

# INDEX

5.5.1.5. DapiSpecialTimeoutDeactivate	210
5.5.1.6. DapiSpecialTimeoutGetStatus	211
5.5.1.7. DapiSpecialTimeoutDoValueMaskWRSet32	213
5.5.1.8. DapiSpecialTimeoutDoValueMaskRDSet32	214
5.5.1.9. DapiSpecialTimeoutDoValueMaskWRClr32	215
5.5.1.10. DapiSpecialTimeoutDoValueMaskRDClr32	216
5.5.1.11. DapiSpecialTimeoutDoValueLoadDefault	217
<b>5.6. Testfunktionen</b>	<b>218</b>
5.6.1. DapiPing	218
<b>5.7. Register Schreib-Befehle</b>	<b>219</b>
5.7.1. DapiWriteByte	219
5.7.2. DapiWriteWord	220
5.7.3. DapiWriteLong	221
5.7.4. DapiWriteLongLong	222
<b>5.8. Register Lese-Befehle</b>	<b>223</b>
5.8.1. DapiReadByte	223
5.8.2. DapiReadWord	224
5.8.3. DapiReadLong	225
5.8.4. DapiReadLongLong	226
5.9. Delib Übersichtstabelle	227
<b><u>6. Anhang</u></b>	<b>230</b>
6.1. Kontakt / Support	231
6.2. Umwelt und Entsorgung	231
6.3. Revisionen	231
6.4. Urheberrechte und Marken	233

# 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 - ein fairer 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

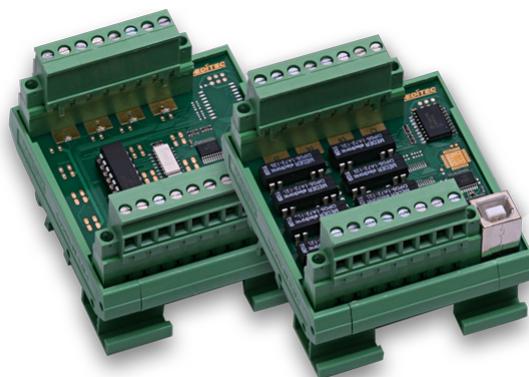
Unsere USB Module der Starter Serie mit reinen digitalen Aus- oder Eingängen sind eine kostengünstige Variante für den Einstieg in die mobile Welt der digitalen I/Os.

Dabei handelt es sich bei den Einsteigermodulen mit wahlweise 8 Optokoppler-Eingängen (USB-OPTOIN-8) oder 8-Relais-Ausgängen (USB-RELAIS-8) um ein kompaktes stand-alone USB Modul.

Digitale Datensignale werden hier über eine USB-Schnittstelle erfasst oder ausgegeben. Die Bereitstellung und Verarbeitung dieser Signale geschieht durch die Kundenapplikation auf dem Steuer PC. Als einfache Programmierschnittstelle bietet sich hier bspw. unsere DELIB API an, oder eine direkte Kommunikation über unser Ethernet-Protokoll.

Das im Lieferumfang enthaltene Konfigurationstool „DELIB-Module Config“ ermöglicht zusätzlich einen schnellen und unkomplizierten Einstieg bei der Inbetriebnahme.

Das Gehäuse besteht aus einem kompakten Kunststoff Profil und eignet sich zur Montage auf Hutschienen, wie sie typischerweise in Schaltschränken verwendet werden.



## 1.5. Lieferumfang

Folgende Artikel sind im Lieferumfang enthalten:

- USB-RELAIS-8 oder USB-OPTOIN-8 Modul
- 2x 8 pol. Steckverbinder in schwarz
- USB Kabel 1,5m
- Installations CD mit Handbüchern und Treibern

# Inbetriebnahme



## **2. Inbetriebnahme**

### **2.1. Sicherheitshinweise**

Bitte machen Sie sich vor der Inbetriebnahme Ihres DEDITEC Produktes mit diesem Handbuch vertraut und lesen Sie sich die nachfolgenden Punkte genau durch:

- Schäden, die durch Nichtbeachten dieser Bedienungsanleitung verursacht werden, führen zum Erlöschen der Gewährleistung bzw. Garantie dieses Produktes. Für Folgeschäden übernehmen wir keinerlei Haftung!
- Für Sach- oder Personenschäden, die durch unsachgemäße Handhabung oder Nichtbeachtung der Sicherheitshinweise entstehen könnten, übernehmen wir keinerlei Haftung!
- Vermeiden Sie ein direktes Berühren elektronischer Bauteile auf der Leiterplatte. Dies könnte zu elektrostatischen Entladungen führen und empfindliche Bauteile zerstören. Entladen Sie sich vorsichtshalber immer vor dem Berühren an einem elektrisch geerdeten Gegenstand.
- Eigenmächtige Umbauten oder technische Änderungen an diesem Produkt sind aus Sicherheits- und Zulassungsgründen (CE) nicht gestattet und führen zum Erlöschen der Gewährleistung bzw. Garantie.
- Betreiben Sie das Modul nicht außerhalb der maximal zulässigen technischen Daten.
- Das Produkt ist nicht für den Betrieb in feuchter oder nasser Umgebung geeignet.

## **2.2. Schritt 1 - Anschluss der Stromversorgung**

Bei unseren reinen USB-Modulen der Starter-Serie, wird das Modul über die USB-Schnittstelle mit der nötigen Versorgungsspannung von +5V versorgt.

Eine externe Versorgung wird nicht benötigt.

## **2.3. Schritt 2 - Verbinden mit dem PC oder Netzwerk**

Anschluss via USB:

Verbinden Sie das Modul, mit dem im Lieferumfang enthaltenen USB-Kabel, mit Ihrem PC oder USB-Hub.

Möchten Sie mehrere USB Module gleichzeitig an einem PC anschließen, muss zunächst jedem Modul eine eigene Modulnummer vergeben werden.

Siehe Kapitel → Beispiel zur Konfiguration identischer USB-Module

## 2.4. Schritt 3 - Installation der Software und Treiber

### Installation unter Windows:

Um dieses Produkt mit einem Windows basierten PC betreiben zu können, gehen Sie bitte wie folgt vor:

Installieren Sie zuerst die DELIB-Treiberbibliothek für Windows, indem Sie Datei "delib\_install.exe" von der DEDITEC-Treiber CD ausführen. Diese befindet sich Verzeichnis "\zip\delib\delib\_install.exe".

Alternativ können Sie die aktuellste DELIB Version auch von unserer Homepage herunterladen. → <http://www.deditec.de/delib>

### Installation unter Linux:

Um dieses Produkt mit einem Linux basierten PC betreiben zu können, gehen Sie bitte wie folgt vor:

Entpacken Sie das ZIP File "delib-linux.zip" von der DEDITEC Treiber CD und kopieren Sie sich die delib.dll in Ihr Projektverzeichnis.

Alternativ können Sie die aktuellste DELIB Version auch von unserer Homepage herunterladen. → <https://www.deditec.de/media/zip/delib/delib-linux.zip>

## 2.5. Schritt 4 - Funktionstest

Mit unserem Tool "DELIB-Module Config" können Sie das Modul relativ schnell und einfach und ohne Programmierkenntnisse in Betrieb nehmen und auf Funktion überprüfen.

Folgen Sie hierfür den Anweisungen im Kapitel "**DELIB-Module Config**".

# Hardware Beschreibung



# 3. Hardware Beschreibung

## 3.1. Allgemeine technische Daten

### 3.1.1. USB-RELAIS-8



#### Elektrische Daten:

Versorgungsspannung: +5V (über USB-Bus)  
Leistungsaufnahme: max. 5W

#### Umgebung:

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

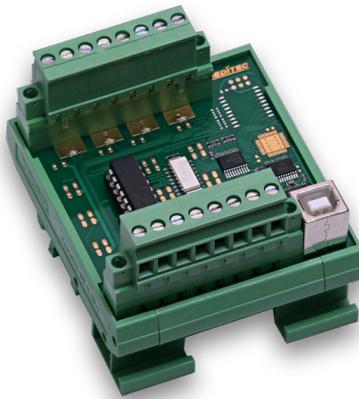
#### Mechanik:

Abmessungen in mm (LxBxH): 67,5 x 77 x 55  
Befestigung: Hutschiene TS 35 x 7,5 mm

#### Mögliche Varianten:

Produkt	Anschlussver- drahtung	Aktivitäts-LED	Gehäuse
USB-RELAIS-8	Schraubbare Klemmleisten	1 je Ausgang	Ja
USB-RELAIS-8_A	Schraubbar	-	-
USB-RELAIS-8_B	Schraubbare Klemmleisten	1 je Ausgang	-

### 3.1.2. USB-OPTOIN-8



#### Elektrische Daten:

Versorgungsspannung: +5V (über USB-Bus)  
Leistungsaufnahme: max. 5W

#### Umgebung:

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

#### Mechanik:

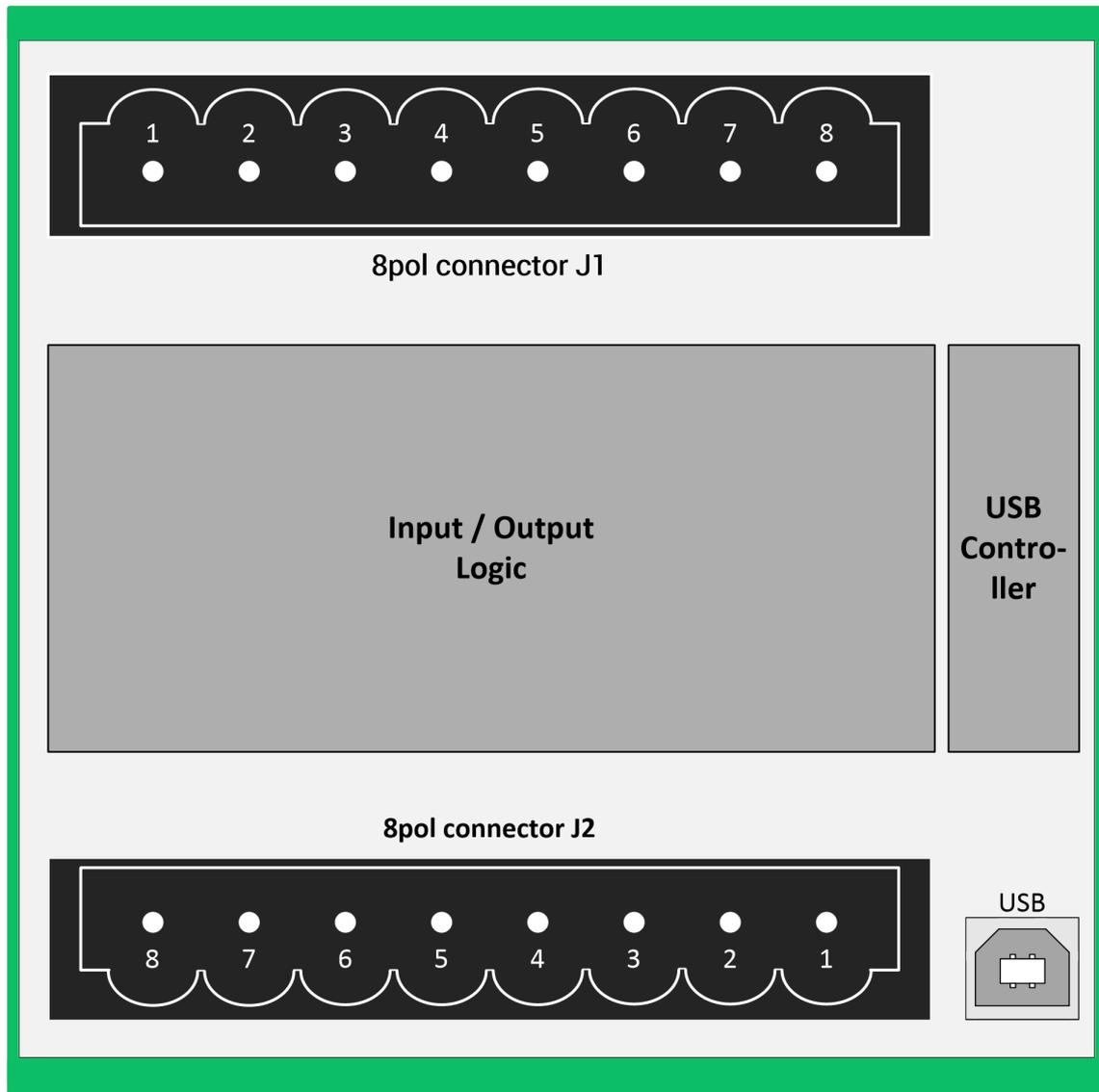
Abmessungen in mm (LxBxH): 67,5 x 77 x 55  
Befestigung: Hutschiene TS 35 x 7,5 mm

#### Mögliche Varianten:

Produkt	Anschlussver- drahtung	Aktivitäts-LED	Gehäuse
USB-OPTOIN-8	Schraubbare Klemmleisten	1 je Eingang	Ja
USB-OPTOIN-8_A	Schraubbar	-	-
USB-OPTOIN-8_B	Schraubbare Klemmleisten	1 je Eingang	-

## 3.2. Übersichtsbild

### 3.2.1. USB-RELAIS/OPTOIN-8



Übersichtsbild eines USB-RELAIS/OPTOIN-8

## 3.3. Schnittstellen

Die Standard Schnittstelle auf dem Modul ist USB.

### 3.3.1. USB

#### Technische Daten:

Standard:	USB 1.1 / USB 2.0
Verbindungsaufbau:	USB Kabel Typ A auf Typ B
Zugriffszeit PC auf Modul*:	4,06 ms**

\* Berechnet mit 1000 Zugriffen auf das Modul über die DELIB Treiberbibliothek mit dem Befehl DapiDoSet32

\*\* durchschnittliche Zeit für 32-Bit Zugriffe

## **3.4. LEDs**

Es gibt für jeden digitalen I/O Kanal jeweils eine LED.

### **3.4.1. Definition der LEDs**

**LED ACT:**

Leuchtet bei USB-Aktivität.

**LED I/O:**

Jeder digitale Ein- oder Ausgang verfügt über eine separate LED, die bei aktivem Signalzustand aufleuchtet.

## 3.5. Digitale Ausgänge

### 3.5.1. Technische Daten

#### Technische Daten Relais 1A:

Anzahl Kanäle pro Block: 8

Typ: Schließer (NO)

Max. Schaltspannung: 36V AC / DC

Max. Schaltstrom: 0.5A AC / DC

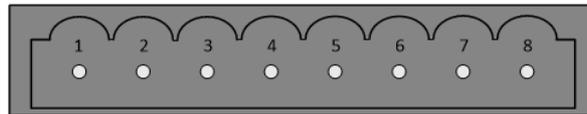
Max. Transportstrom: 1A AC / DC

Max. Schaltleistung: 10W

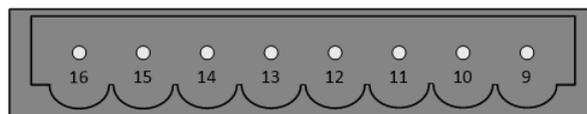
#### Zusätzliche Funktionen:

- Timeout Schutz

### 3.5.2. Pinbelegung



Pin	Belegung
1	Output Channel 1
2	Output Channel 1
3	Output Channel 2
4	Output Channel 2
5	Output Channel 3
6	Output Channel 3
7	Output Channel 4
8	Output Channel 4



Pin	Belegung
9	Output Channel 5
10	Output Channel 5
11	Output Channel 6
12	Output Channel 6
13	Output Channel 7
14	Output Channel 7
15	Output Channel 8
16	Output Channel 8

### 3.5.3. Anschlussbeispiel

Anschlussbeispiel für Relais:

```
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

### 3.5.4. Timeout Funktion

Die Timeout Funktion bietet die Möglichkeit, im Falle eines Verbindungsabbruches zwischen Steuer PC und dem DEDITEC Modul, automatisch die Ausgänge ein- oder auszuschalten. Dies kann für jeden einzelnen Kanal per Software definiert werden.

#### Funktionen:

- Zeitlich festlegbare automatische Aktivierung der Timeout-Schutz-Funktion im Timeout-Fall (zwischen 0,1 Sekunden und 6553 Sekunden)
- Im Timeout-Fall können digitale Ausgänge aktiviert, deaktiviert oder unverändert gelassen werden
- 3 verschiedene Timeout-Modi: "normal", "auto reactive" und "secure output" für verschiedene Vorgehensweisen bei einem Timeout-Fall

Näheres siehe Kapitel → Ausgabe-Timeout verwalten

## 3.6. Digitale Eingänge

### 3.6.1. Technische Daten

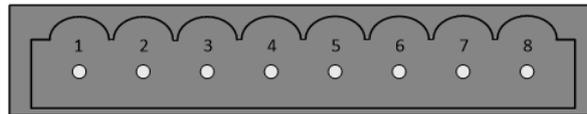
#### Technische Daten Optokoppler:

Anzahl Kanäle pro Block:	8
Spannungsbereiche:	15V – 30V DC/AC (optional 5V – 15V oder 30V – 50V DC/AC)
Eingangsstrom:	max. 14mA
Galvanische Trennung:	bis 2,5kV AC für 1 Minute

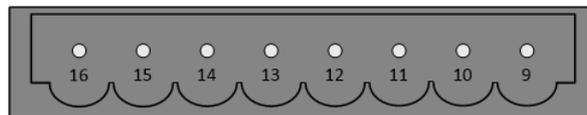
#### Zusätzliche Funktionen:

- 16 Bit Zähler je Kanal
- Maximal mögliche Zählvorgänge: 65535 / Kanal. Nach Speicherüberlauf Reset auf Null.
- Interne Zähllogik bis 10kHz mit Latch Funktion
- Programmierbarer Filter für Eingangskanäle (Flip-Flop und Counter): Mindest low bzw. high Impulsdauer: 5ms...255ms
- Erkennt einen Wechsel von Low auf High und High auf Low Pegel
- Erfassung von Eingangszustandsänderung zwischen zwei Auslesevorgängen

### 3.6.2. Pinbelegung



Pin	Belegung
1	Input Channel 1
2	Input Channel 1
3	Input Channel 2
4	Input Channel 2
5	Input Channel 3
6	Input Channel 3
7	Input Channel 4
8	Input Channel 4



Pin	Belegung
9	Input Channel 5
10	Input Channel 5
11	Input Channel 6
12	Input Channel 6
13	Input Channel 7
14	Input Channel 7
15	Input Channel 8
16	Input Channel 8

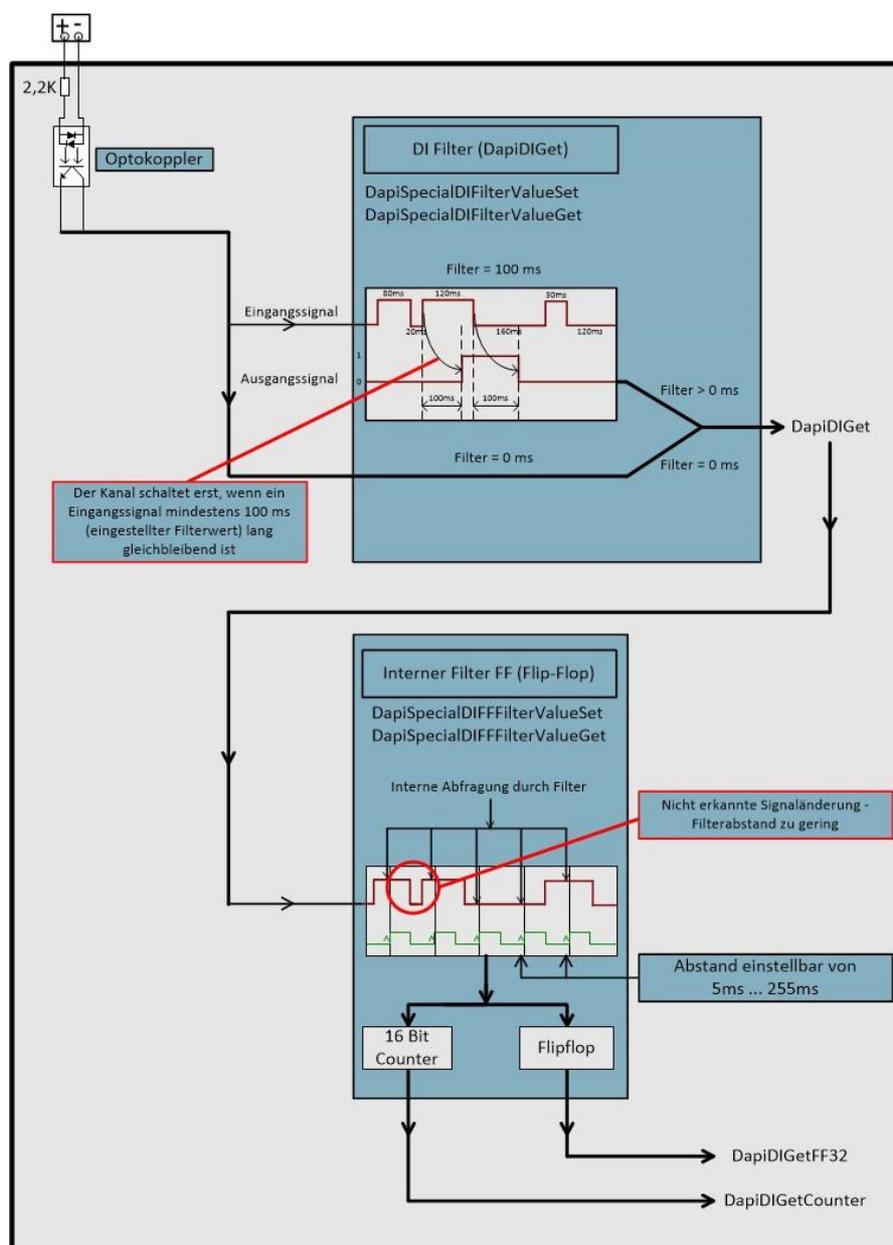


### 3.6.4. EingangsfILTER

Jeder Eingang kann mit einer digitalen Filterfunktion versehen werden. Unerwünschte Impulse werden somit vom System ignoriert. Die gültige Impulsdauer lässt sich softwareseitig zwischen 0..255ms einstellen. Ein Wert von 0 ms bedeutet, dass der Filter nicht aktiv ist.

Siehe auch Kapitel → Digitale Eingänge Lesen

Nachfolgendes Schema soll das Ganze noch einmal genauer verdeutlichen:



### 3.6.5. Zustandsänderungen überwachen

Diese Funktion ermöglicht es, Zustandsänderungen an den Eingängen zu überwachen. Eine interne Logik erkennt einen Zustandswechsel von High nach Low oder umgekehrt und schreibt diese Information in ein Register. Die LED I-Change leuchtet auf. Durch Auslesen der Software Register kann diese Information wieder zurück gesetzt und die LED deaktiviert werden.

Siehe Kapitel: → Digitale Eingänge lesen → DapiDiGetFF32

# Software



## **4. Software**

### **4.1. Benutzung unserer Produkte**

#### **4.1.1. 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.

#### **4.1.2. Ansteuerung über mitgelieferte Testprogramme**

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

Für ausführliche Informationen siehe Kapitel [DELIB Module Demo](#)

#### **4.1.3. Ansteuerung auf Protokollebene**

Für Produkte mit Ethernet, CAN oder serielle 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 unsere Module sowie den Aufbau der verschiedenen Kommunikationsprotokolle.

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

Da in manchen Programmiersprachen (wie z.B. 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	-	-
do16		16-Bit Wert		
do32		32-Bit Wert		
		(Bit 0 für Kanal 1, Bit 1 für Kanal 2, ...)		
ai	0, 1, 2, ...	-	hex, volt, mA	nounit
ao	0, 1, 2, ...	Ganz Hexadezimalzahl (beginnend mit 0x).	oder -	-

## **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

#### 4.1.4.1. 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.

### 4.1.4.2. 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

*delib\_cli DI32 0 hex*

Beispiel eines Rückgabewertes: **0xC0000003**

(auf den Kanälen 1, 2, 31 und 32 liegt ein Signal an)

-> lese den Wert von digitalen Eingang 1-32 als hexadezimalzahl

Alternativ kann das Argument "nountit" an alle zu formatierenden Ausgabeanfragen wie folgendermaßen angehängt werden:

*delib\_cli DI8 0 hex nountit*

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

*delib\_cli FF 32 hex*

Beispiel eines Rückgabewertes: **0x3**

(auf Kanälen 33 und 34 wurde eine Zustandsänderung erkannt)

-> lese den Wert der FlipFlops der digitalen Eingänge 33-64 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 milli Ampere  
(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

*delib\_cli AI 3 mA nunit*

Beispiel eines Rückgabewertes: **20.551600**

-> liest den Strom des 4. analogen Eingangs als Kommazahl

## **4.1.5. Ansteuerung über grafische Anwendungen**

### **4.1.5.1. 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](#)

#### **4.1.5.2. 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

### **4.1.5.3. 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>

## 4.1.6. Einbinden der DELIB in Programmiersprachen

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

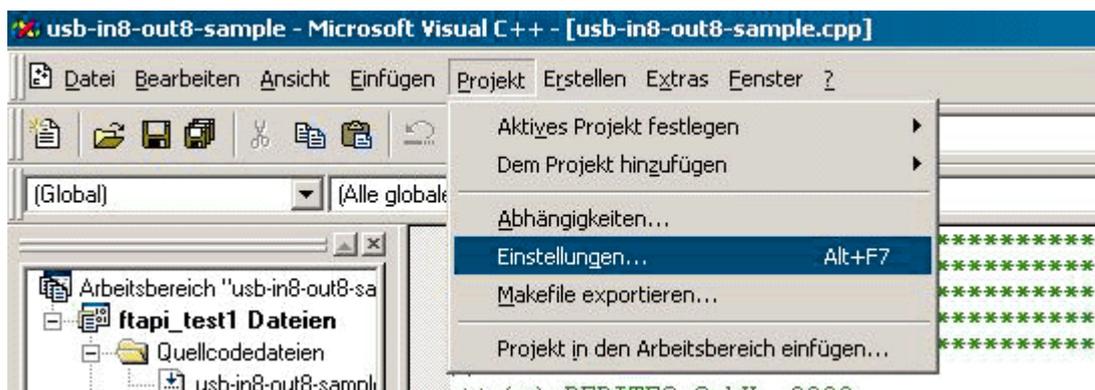
Zur Erleichterung für Verweise auf das DELIB-Include und das DELIB-Lib Verzeichnis werden bei installation der DELIB Umgebungsvariablen definiert.

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

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

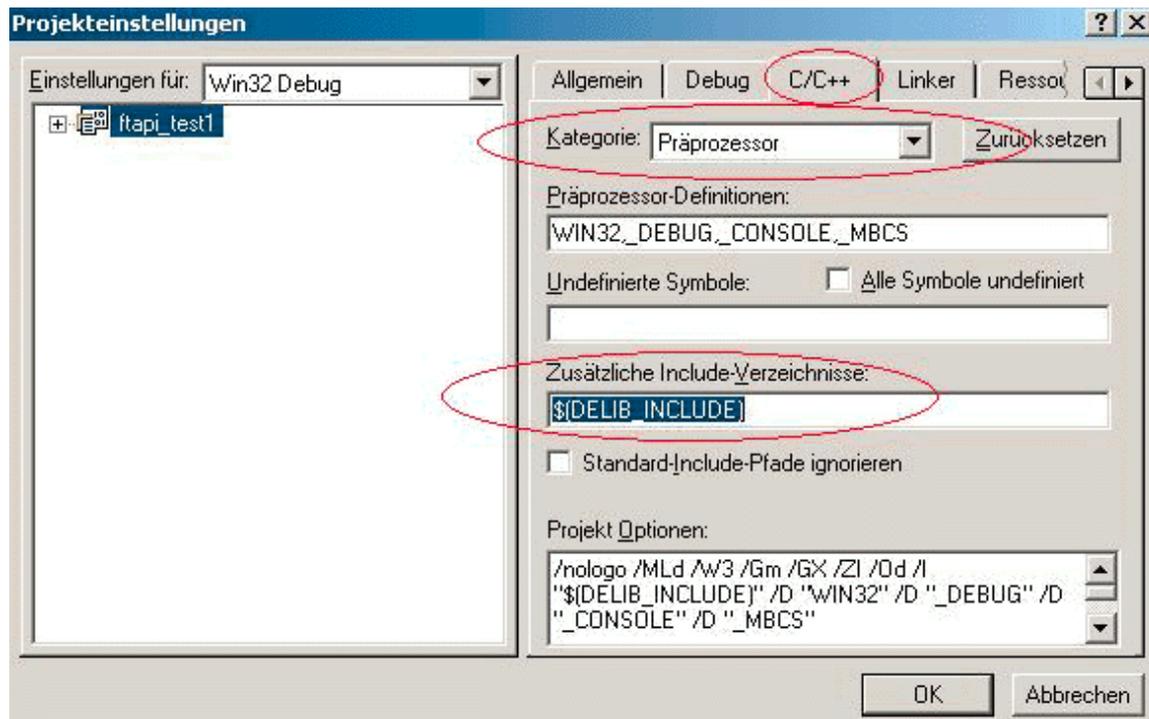
Diese werden im Folgenden in den Projekteinstellungen des Compilers eingetragen.

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



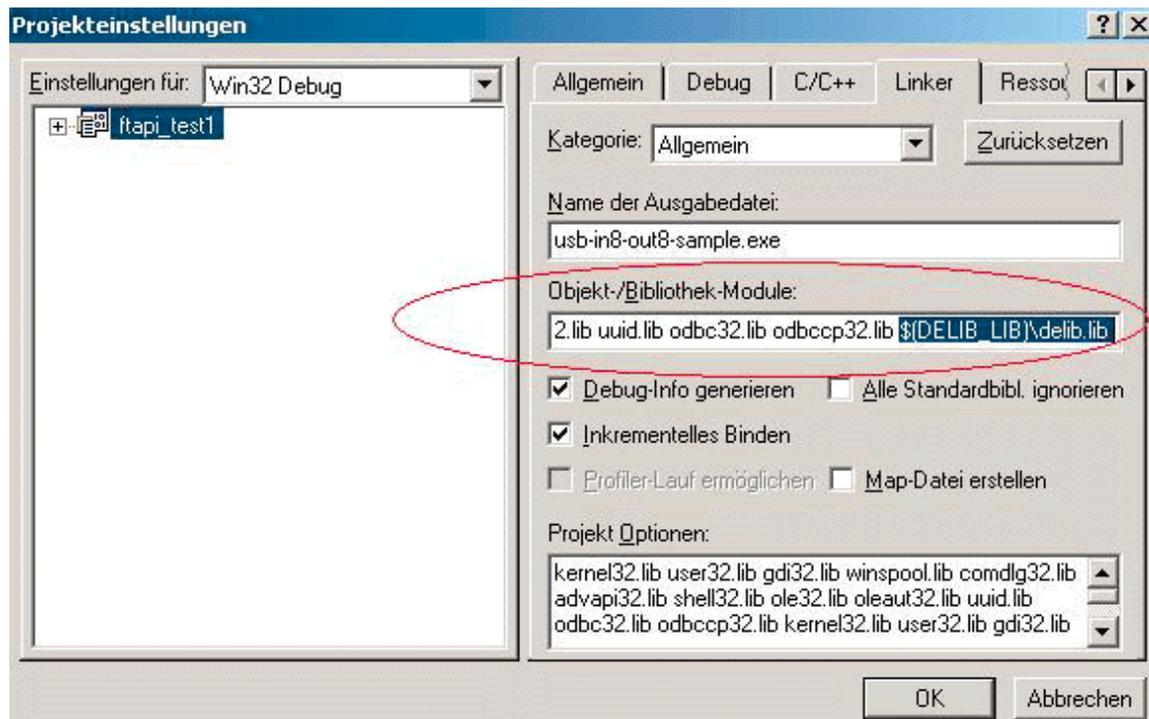
## DELIB.H Eintrag in den Visual-C/C++ Projekt Einstellungen

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.



#### 4.1.6.2. 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

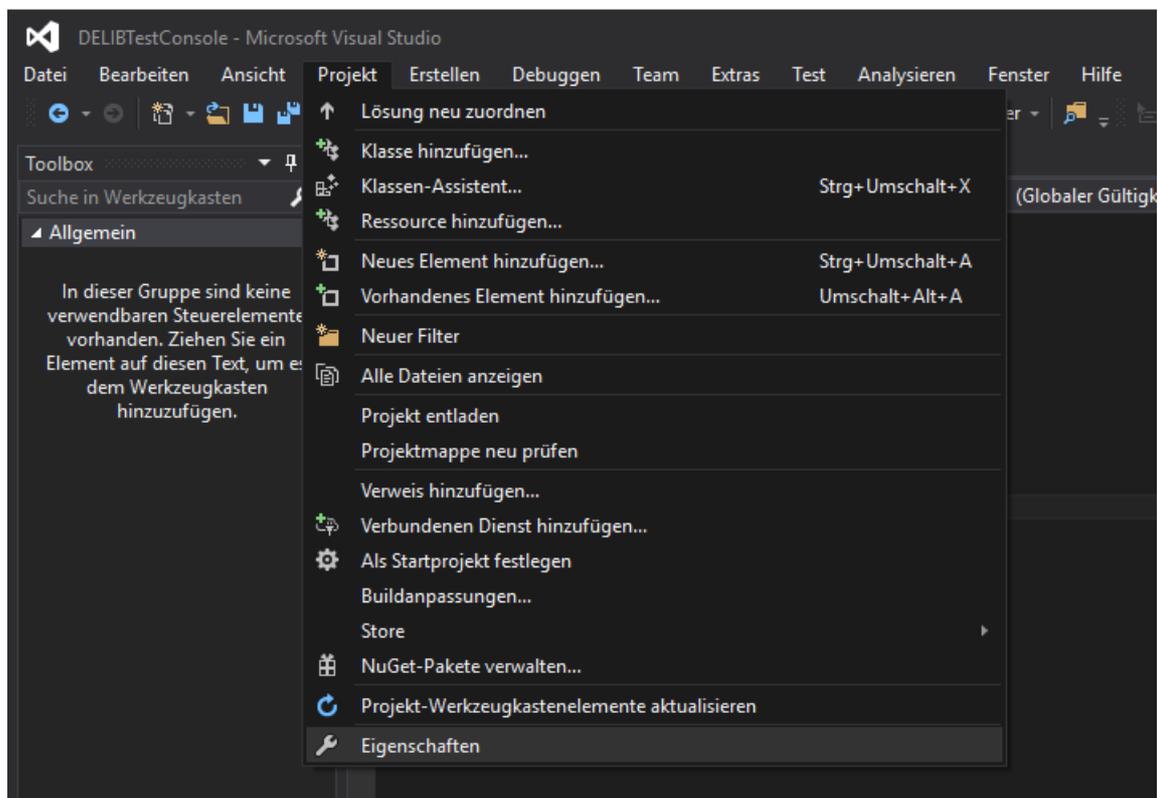
#### 64 Bit DELIB Installation

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

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

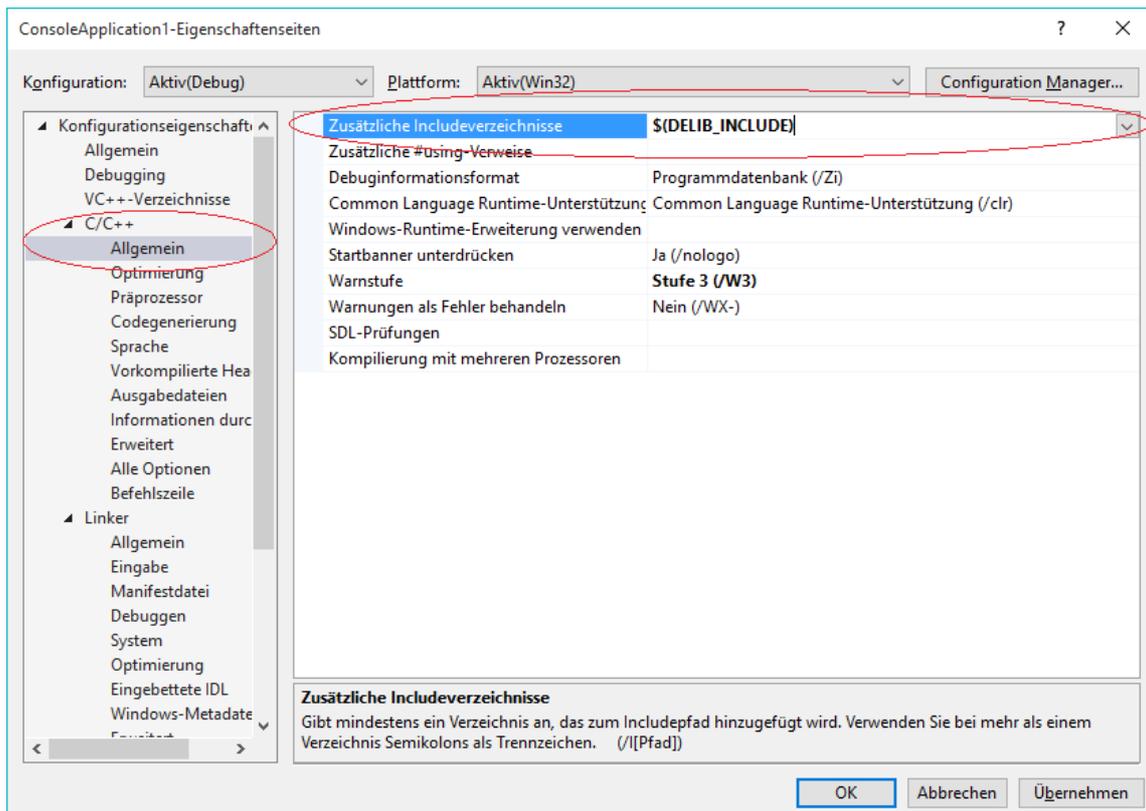
Diese werden im Folgenden in den Projekteinstellungen des Compilers eingetragen.

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



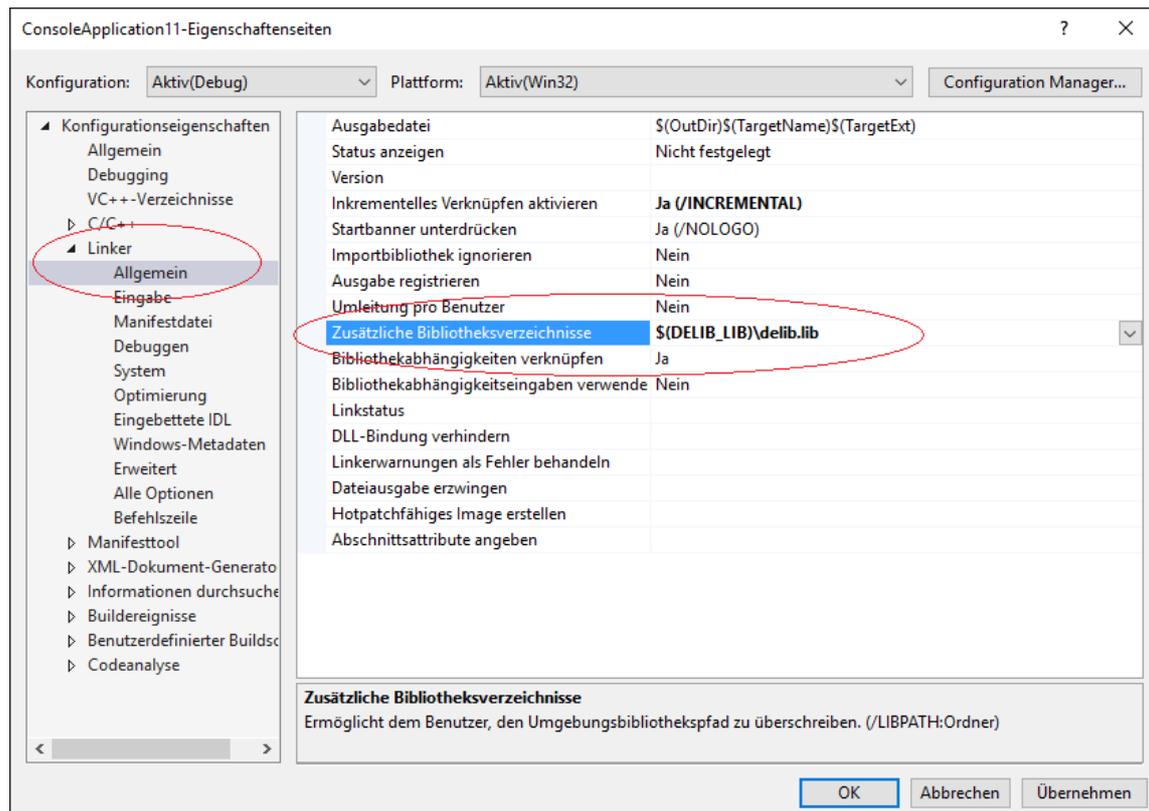
## DELIB.H Eintrag in den Visual-C/C++ Projekt Einstellungen

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" muss "\$(\DELIB\_LIB)\delib.lib" eingetragen werden.

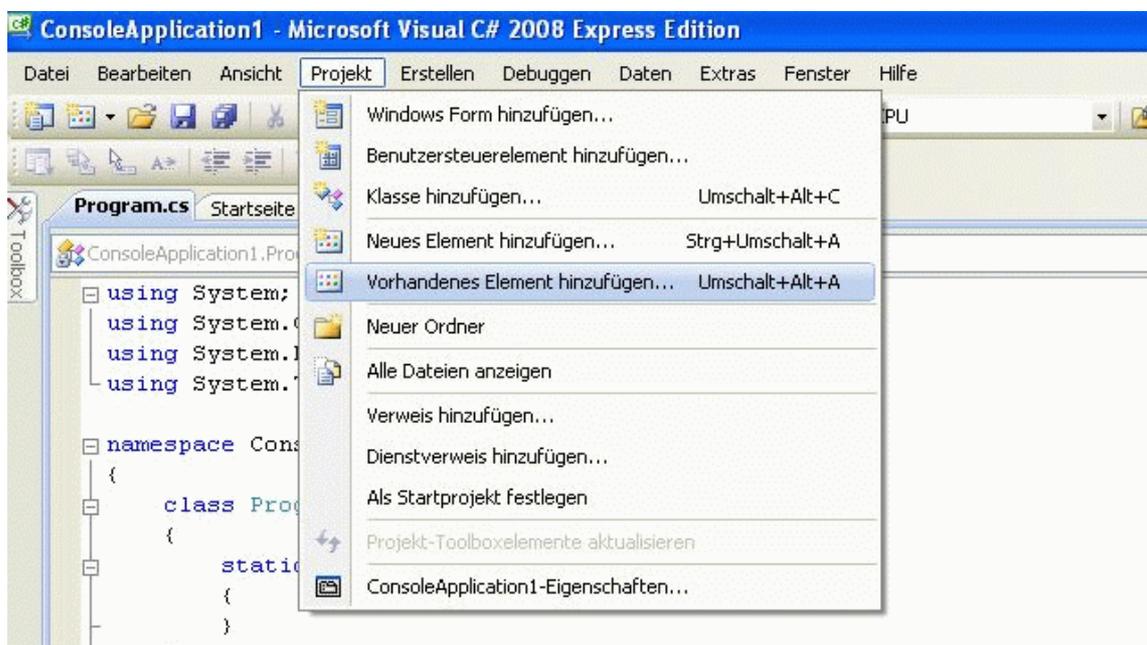


#### 4.1.6.3. Einbinden der DELIB in Visual-C#

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:

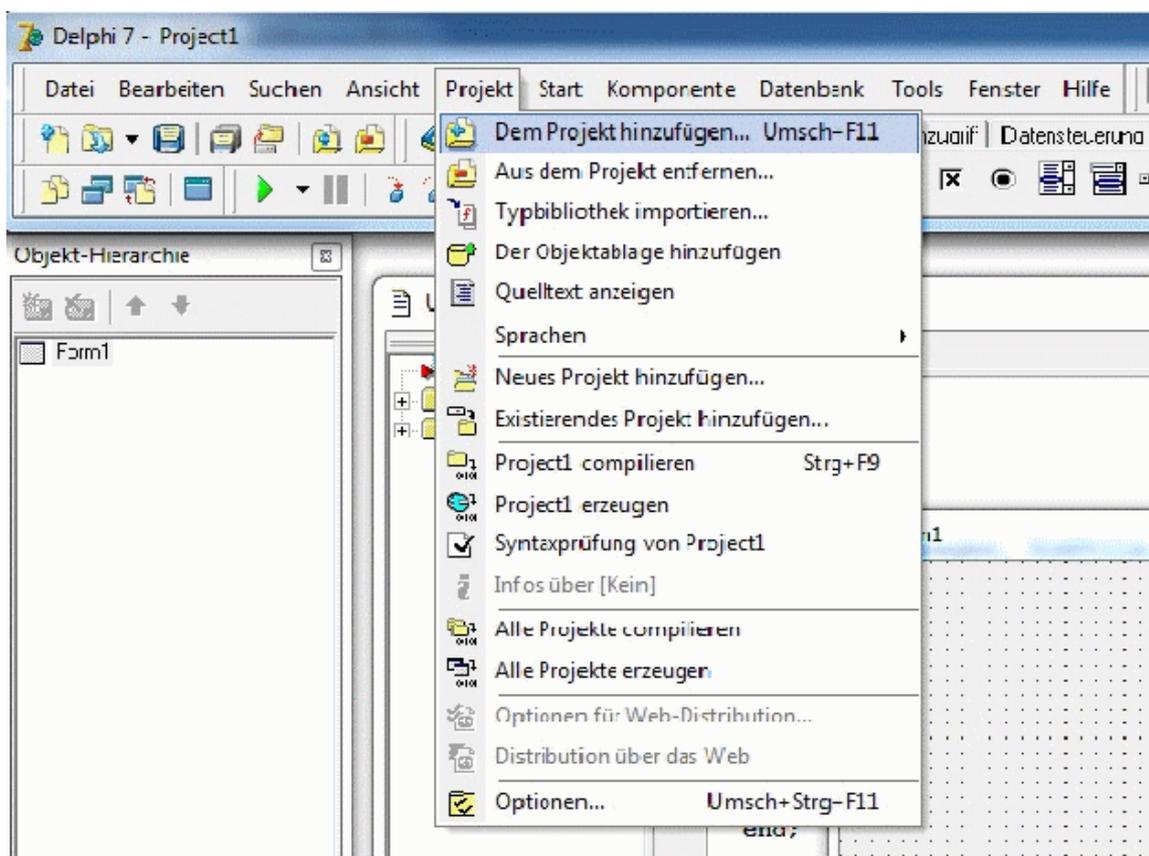
```
using DeLib;
```

#### 4.1.6.4. Einbinden der DELIB in Delphi

Die benötigte Datei für Delphi befindet sich im Verzeichnis

**C:\Programme\DEDITEC\DELIB\include.**

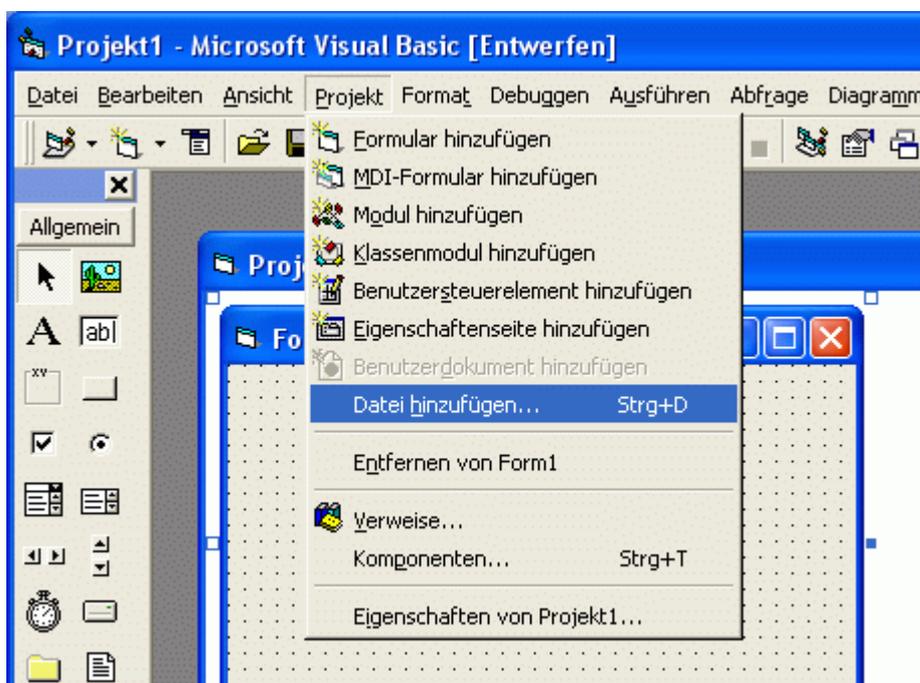
Delphi starten und über das Menue "Projekt -> dem Projekt hinzufügen" im Verzeichnis C:\Programme\DEDITEC\DELIB\include\ die Datei **delib.pas** zum Importieren öffnen.



#### 4.1.6.5. 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 Menu "Projekt -> Datei hinzufügen..." im Verzeichnis C:\Programme\DEDITEC\DELIB\include\ die Datei **delib.bas** zum Importieren öffnen.

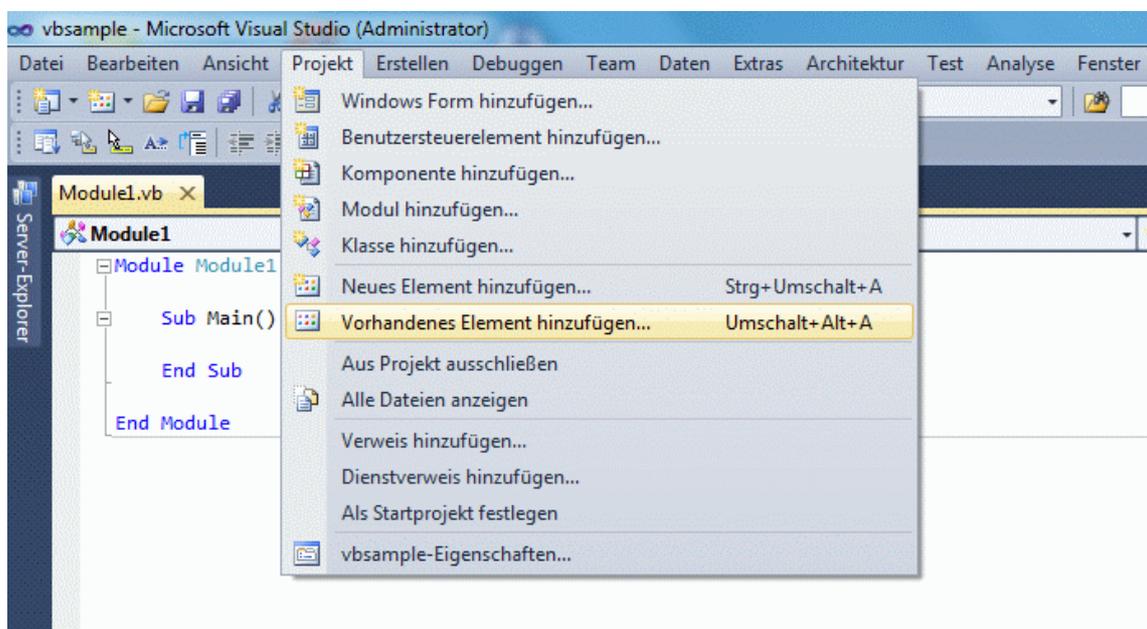


#### 4.1.6.6. Einbinden der DELIB in Visual-Basic.NET (VB.NET)

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.

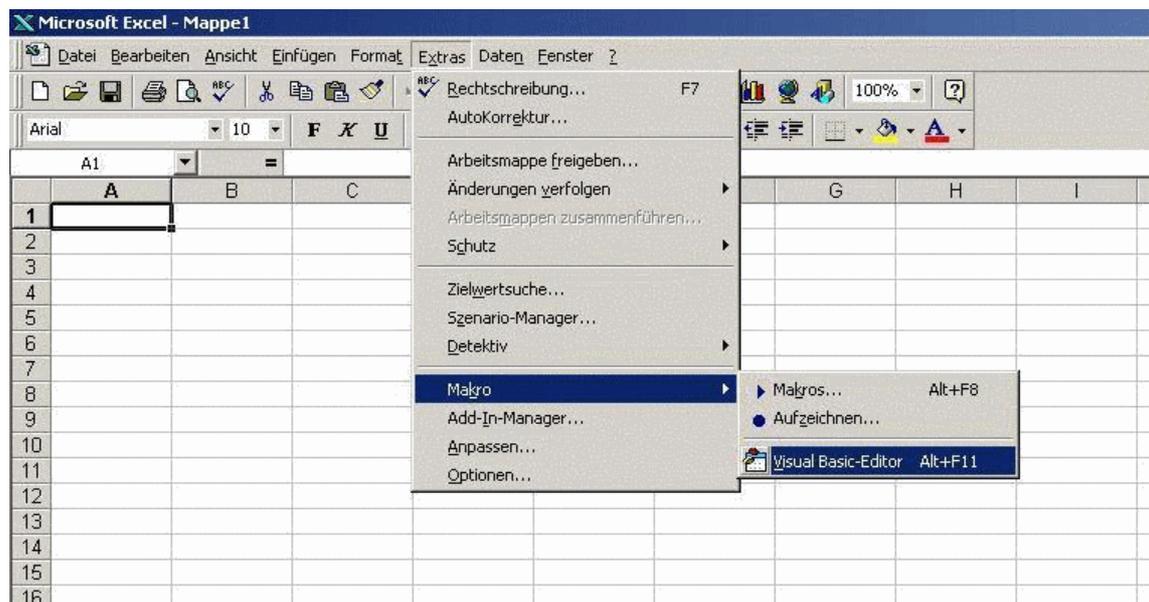


#### 4.1.6.7. Einbinden der DELIB in MS-Office (VBA)

Die benötigte Datei für VBA befindet sich im Verzeichnis

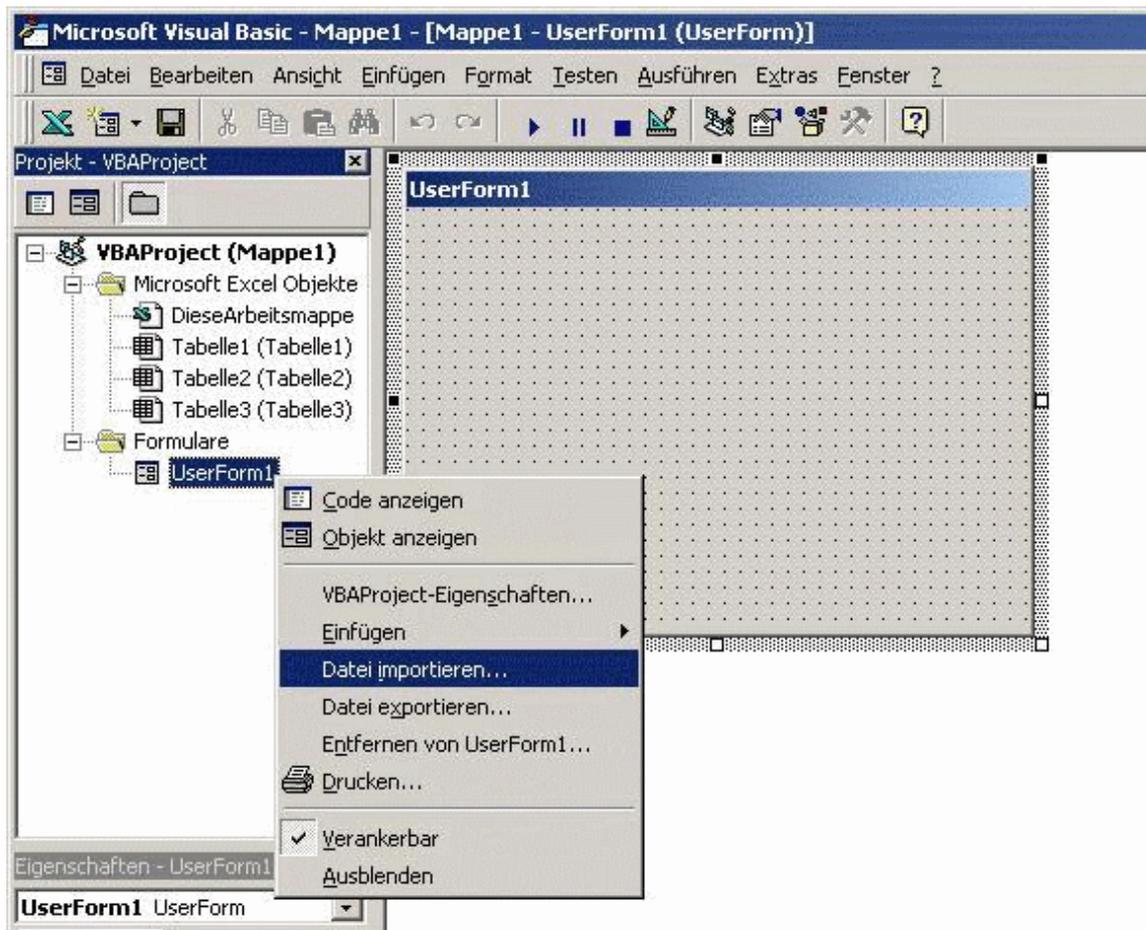
**C:\Programme\DEDITEC\DELIB\include.**

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.



## 4.1.6.8. Einbinden der DELIB in LabVIEW

### 4.1.6.8.1. Einbinden der DELIB in LabVIEW

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

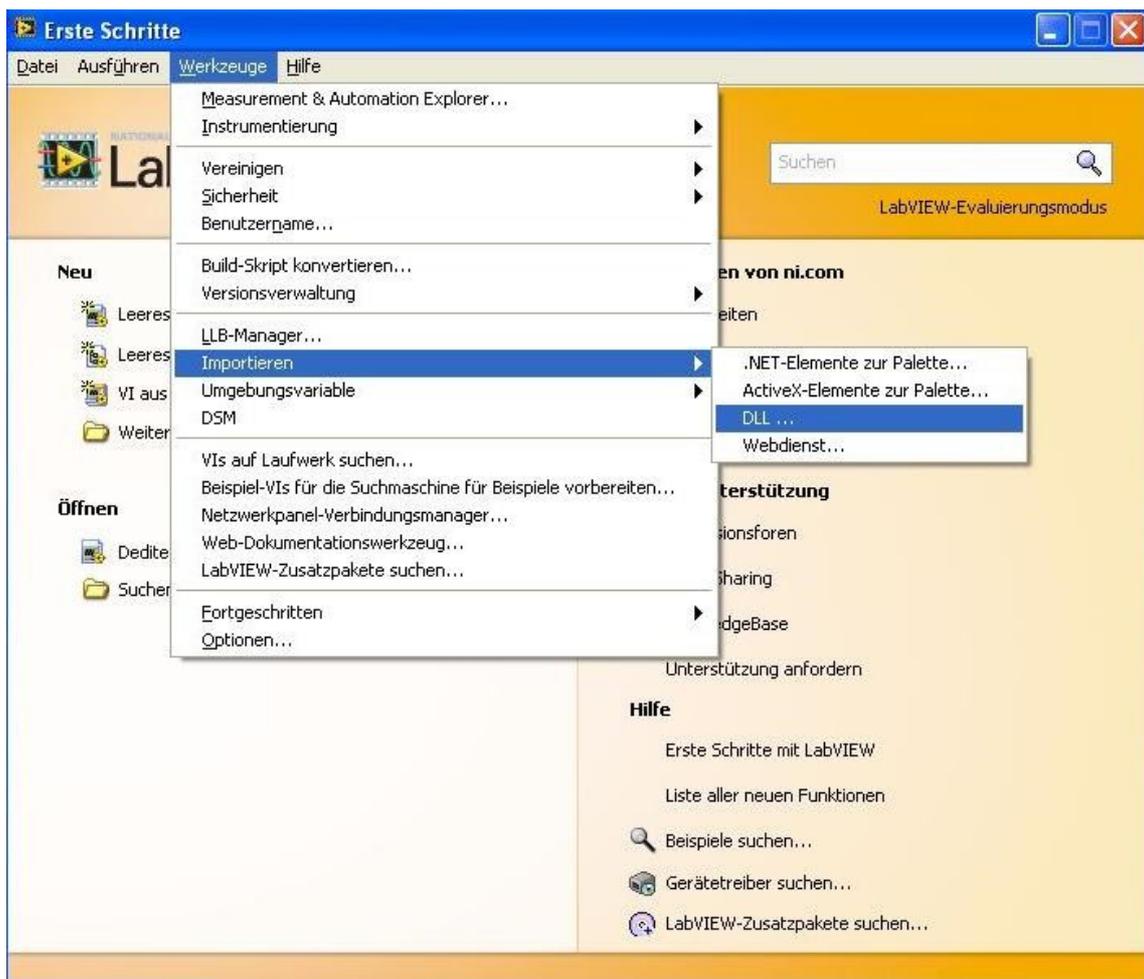
### Beschreibung der Einbindung der "delib.dll" in LabVIEW Version 11

- Die benötigten Dateien für LabVIEW befinden sich im Verzeichnis

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

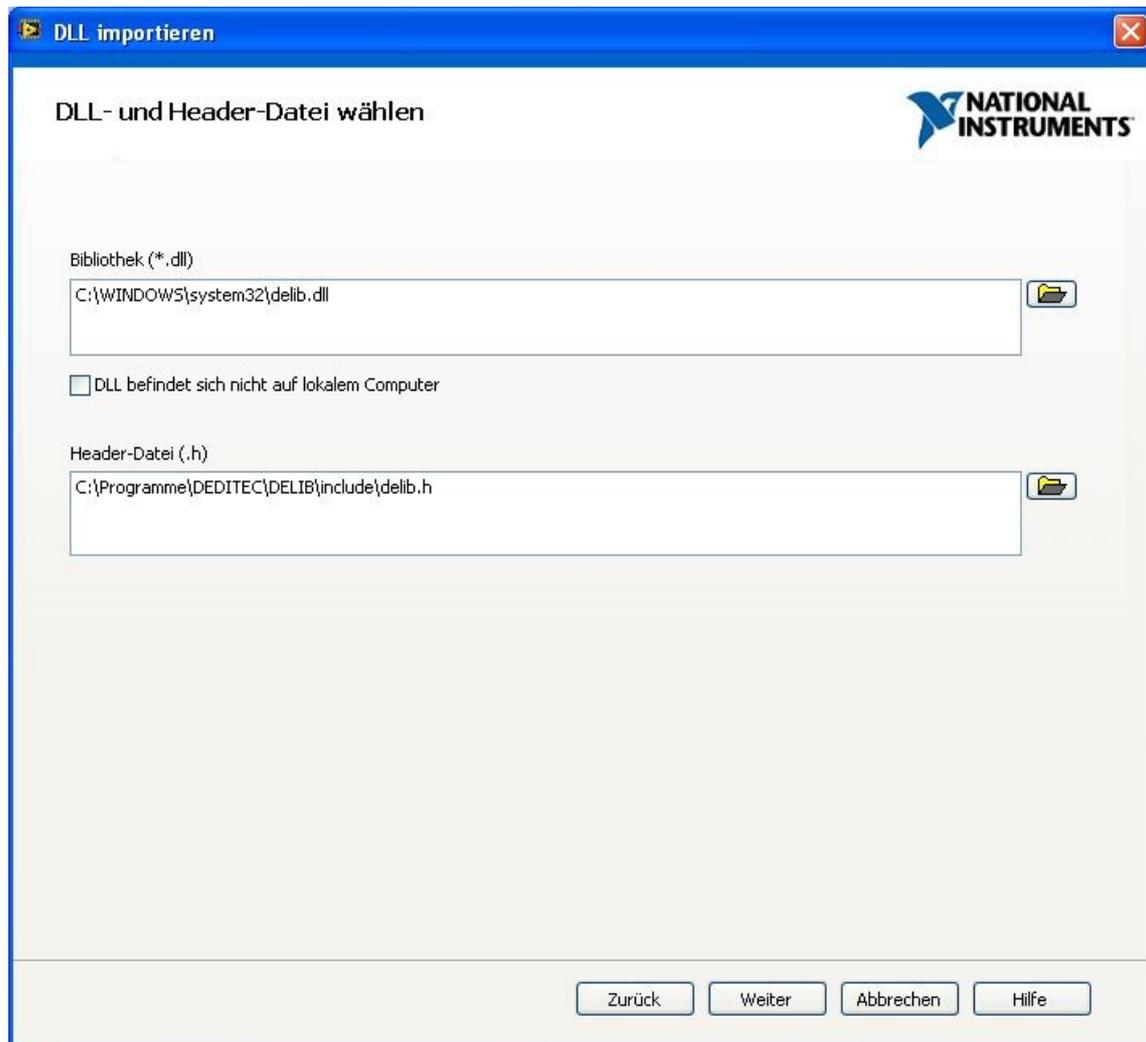
**C:\Programme\DEDITEC\DELIB\include\delib.h**

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

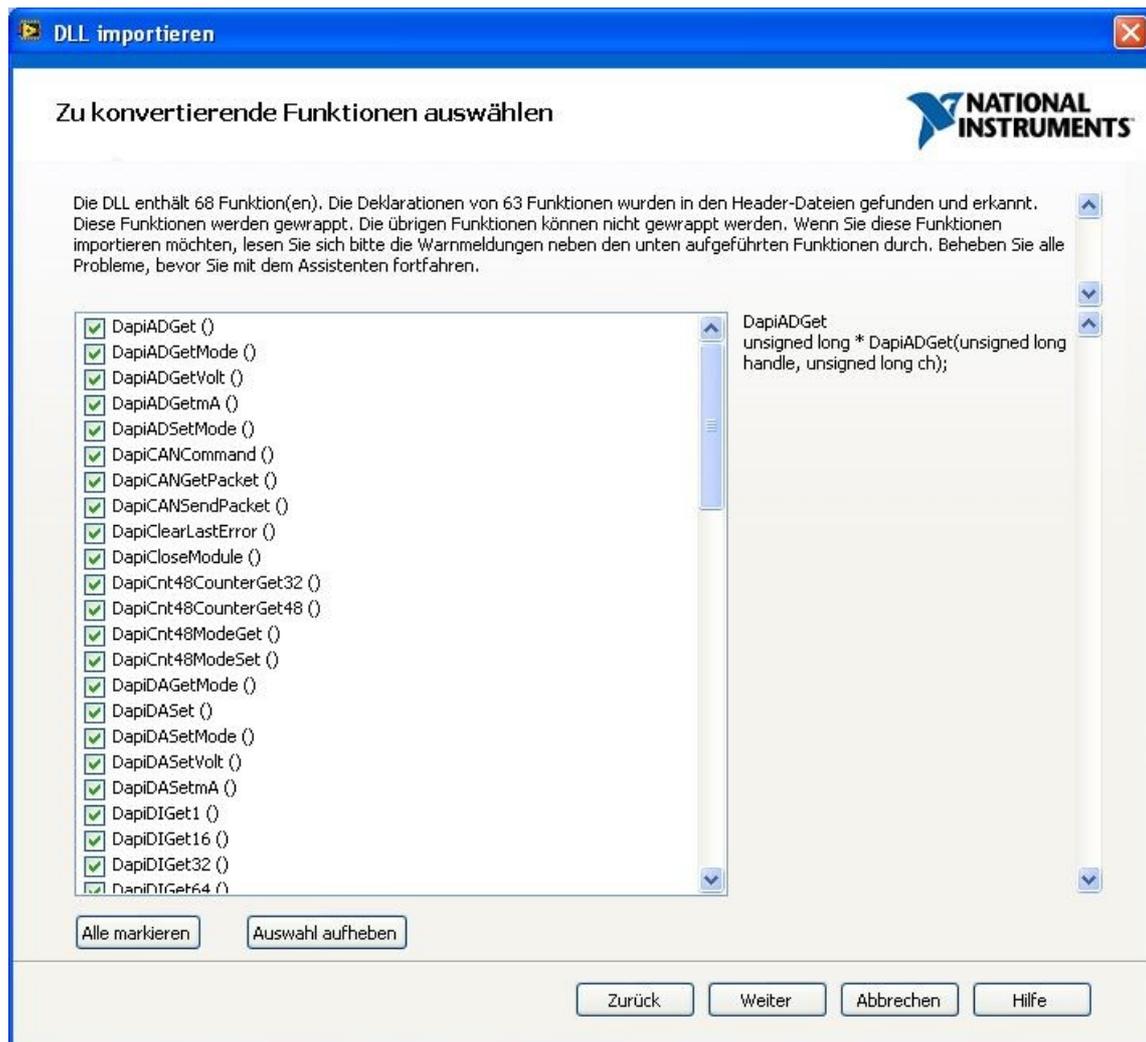




- Im nächsten Fenster über die Browse-Buttons den Speicherort der .DLL und der .H Datei angeben und mit "Weiter" fortfahren.



- 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

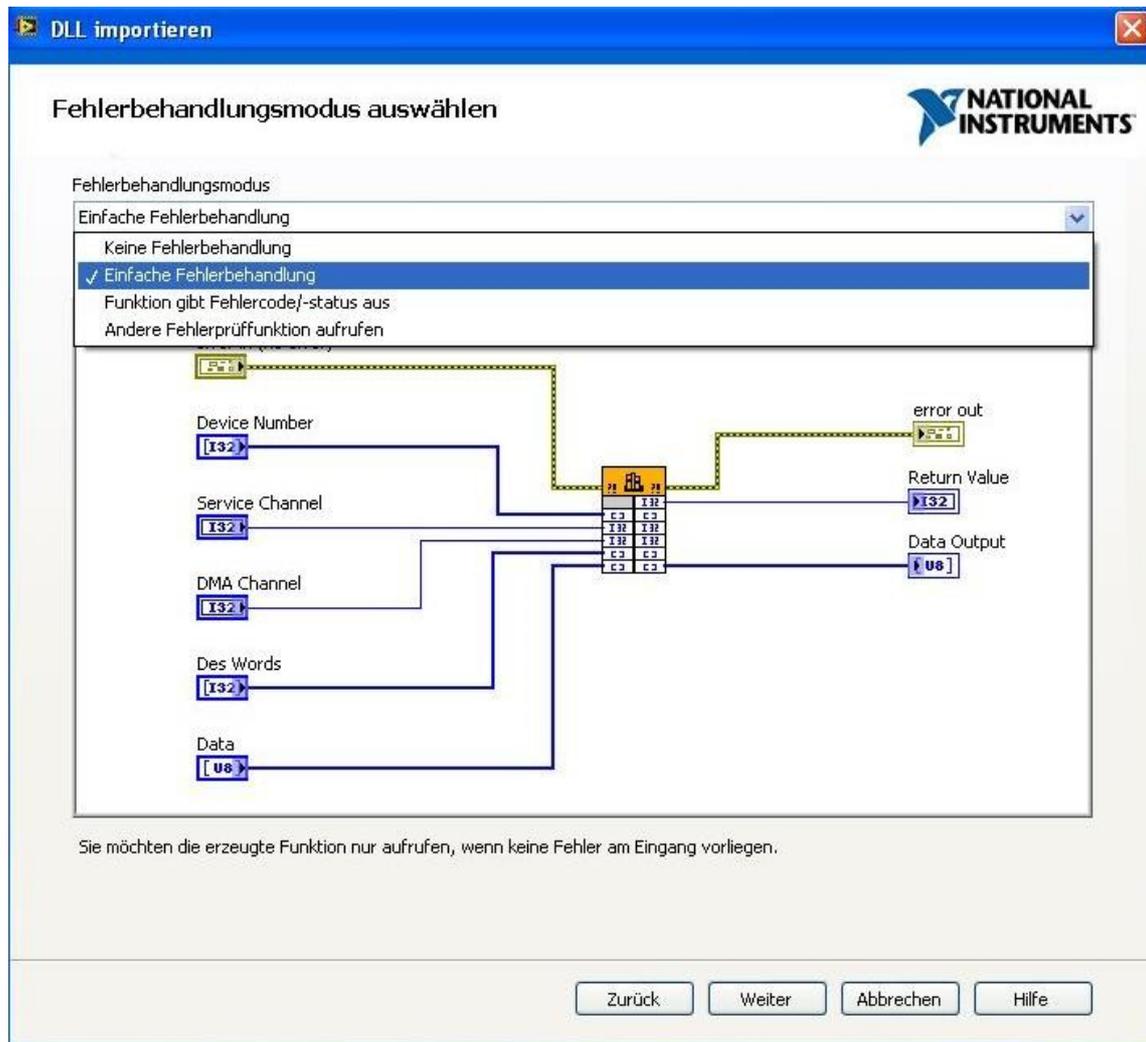
Projektbibliotheksname (.lvlib)  
delib

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.

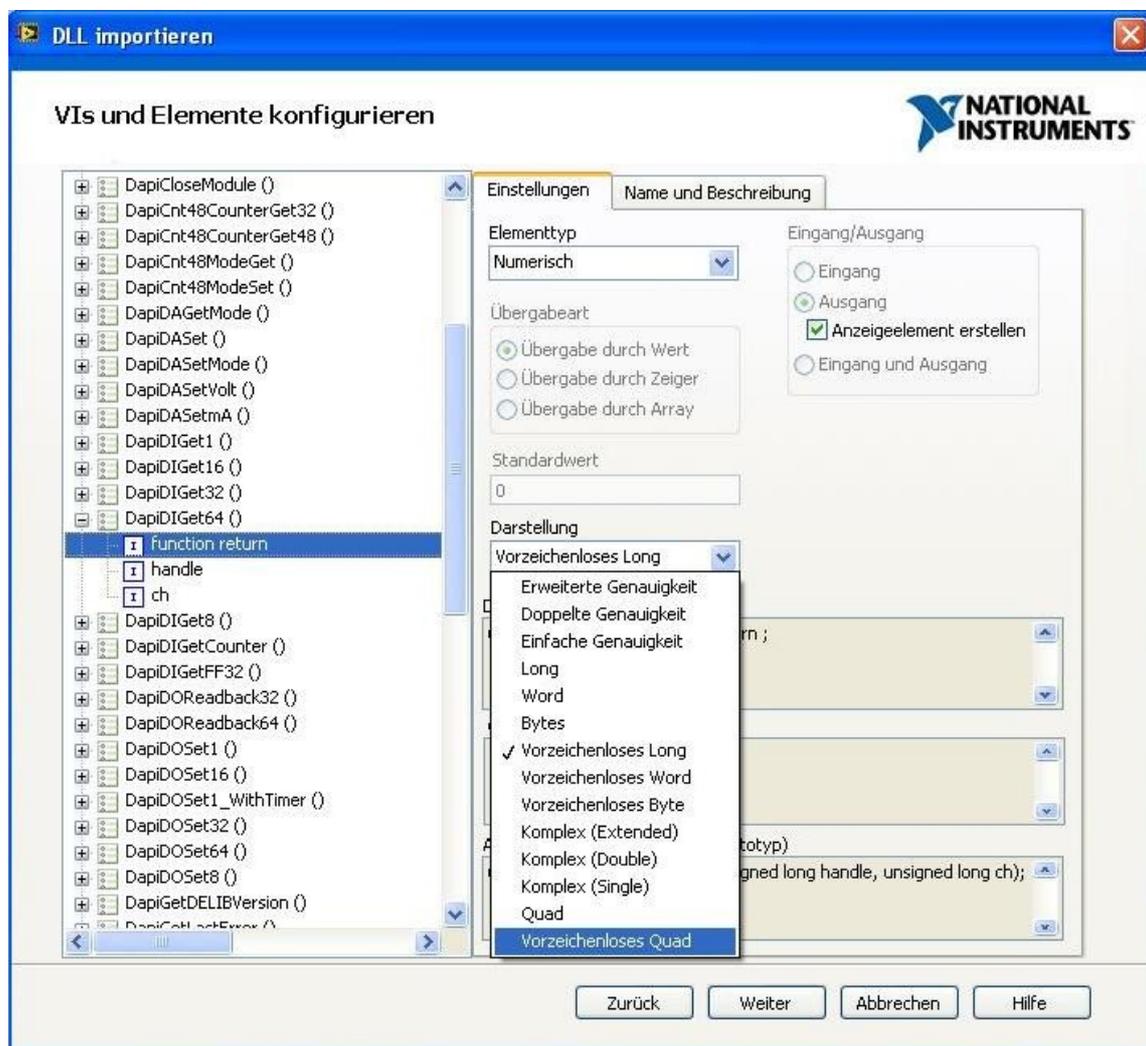
- Folgende VIs müssen bearbeitet werden:

-> DapiCNT48CounterGet48 (function return)

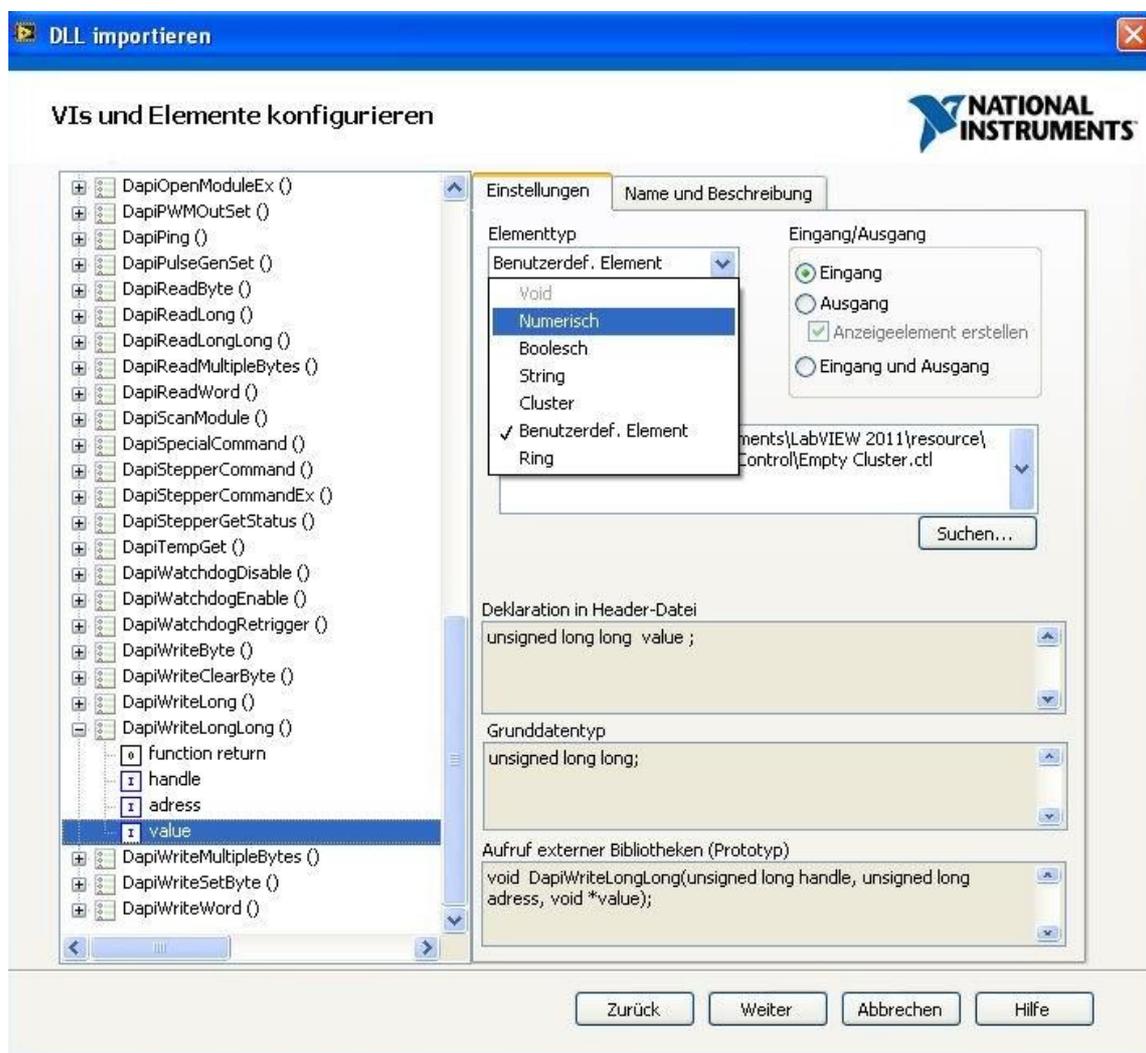
-> DapiDIGet64 (function return)

-> 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)
- Anschließend mit "Weiter" fortfahren.





#### 4.1.6.8.2. 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);
```

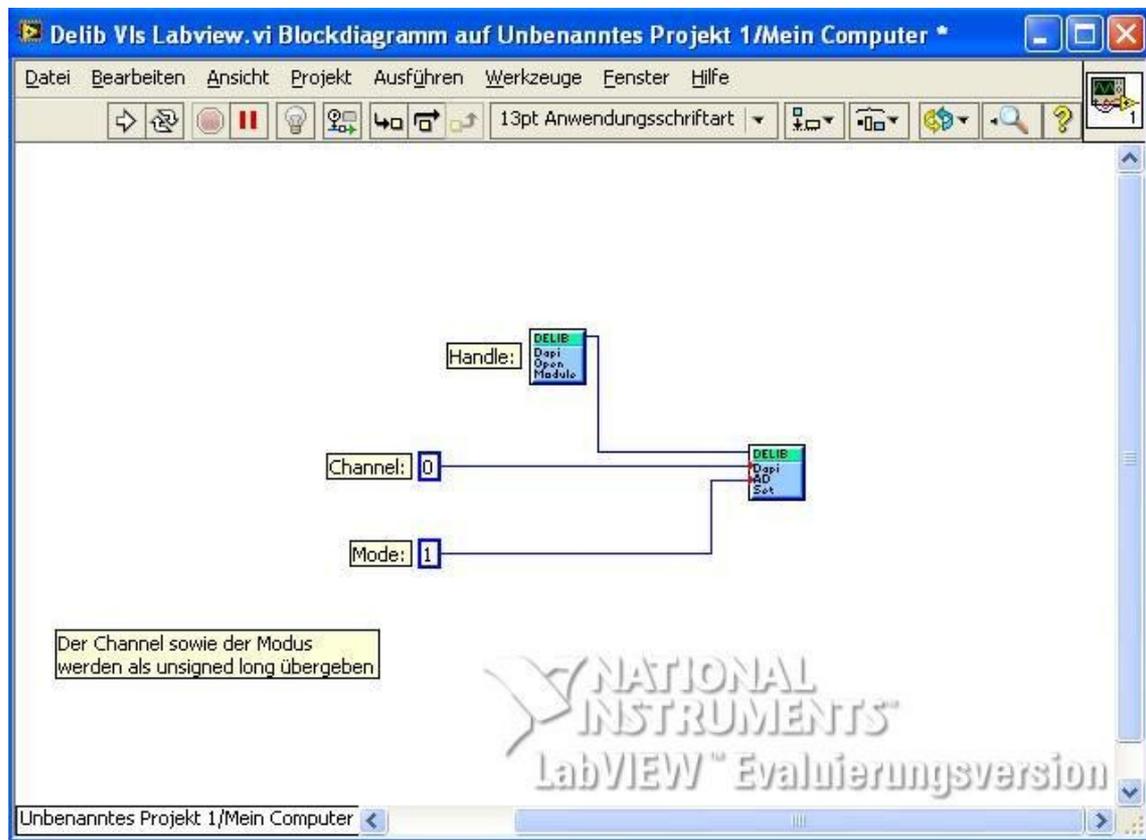
Alternativ kann man auch folgende Schreibweise verwenden:

```
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:



#### 4.1.6.8.3. 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_RELAIS_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:

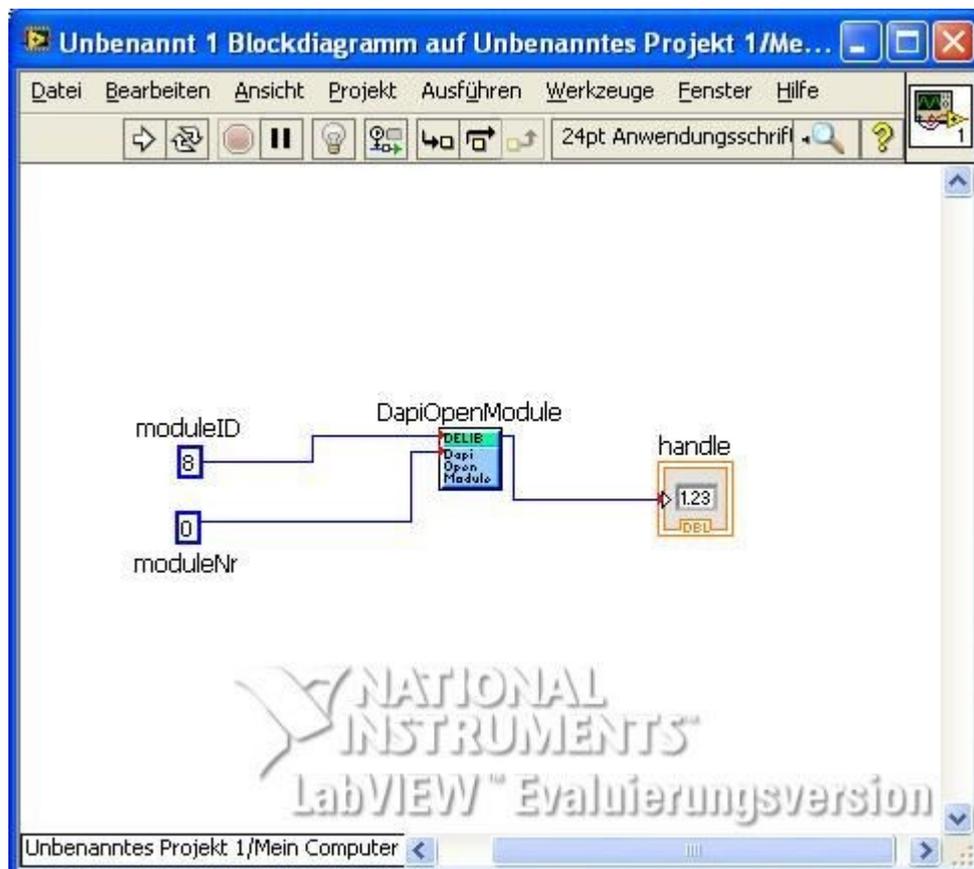
```
handle = DapiOpenModule(RO_ETH, 0); // öffnet ein RO-ETH-Modul mit Modul-Nr 0.
```

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:



#### 4.1.6.9. Einbinden der DELIB in Java

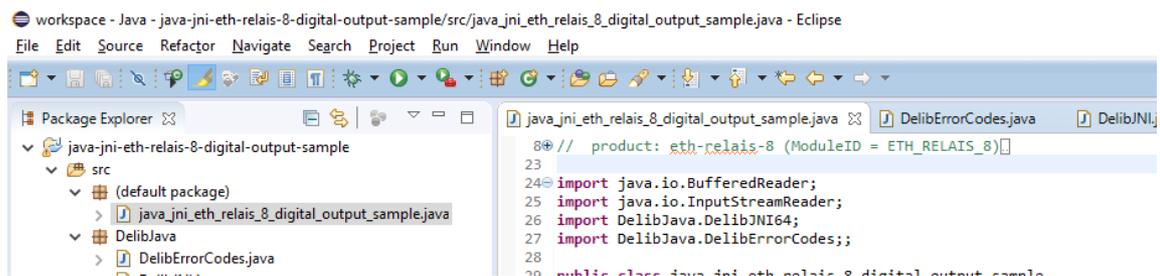
Die benötigten Dateien für Java befinden sich, je nach DELIB-Installation, in folgendem Verzeichnis

**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.



## 4.2. DELIB Treiberbibliothek

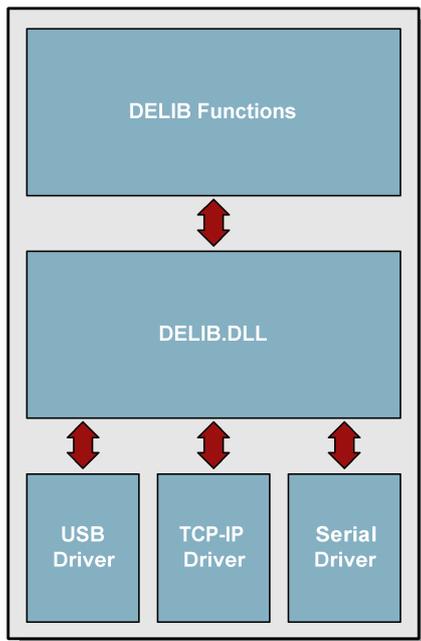
Die DELIB-Treiberbibliothek enthält die DELIB-API und verschiedene Programme für den Konfigurationstest unserer Produkte.

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.

### 4.2.1. 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

## **DELIB Treiberbibliothek ETH**

Während die DELIB für ALLE Produkte zur Verfügung steht, werden bei der DELIB-ETH auf keine weiteren Treiber zugegriffen (wie z. B. USB).

Dies bedeutet, dass die DELIB-ETH nicht installiert werden muss.

Kunden, die eigene Applikationen schreiben, müssen nicht mehr ein eigenes SETUP erstellen, welches auch z. B. USB Treiber installiert.

Die DELIB-ETH.dll Datei muss lediglich im Programmverzeichnis der Applikation liegen und dient hierbei als Schnittstelle zwischen Kunden-Applikation und Hardware.

Die DELIB-ETH bietet sämtliche DELIB Befehle und kann bei Ethernet Anwendungen einfach gegen die alte DELIB ausgetauscht werden.



#### 4.2.1.1. 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.

#### 4.2.1.2. 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

### **4.2.1.3. 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.

## 4.2.2. DELIB Setup

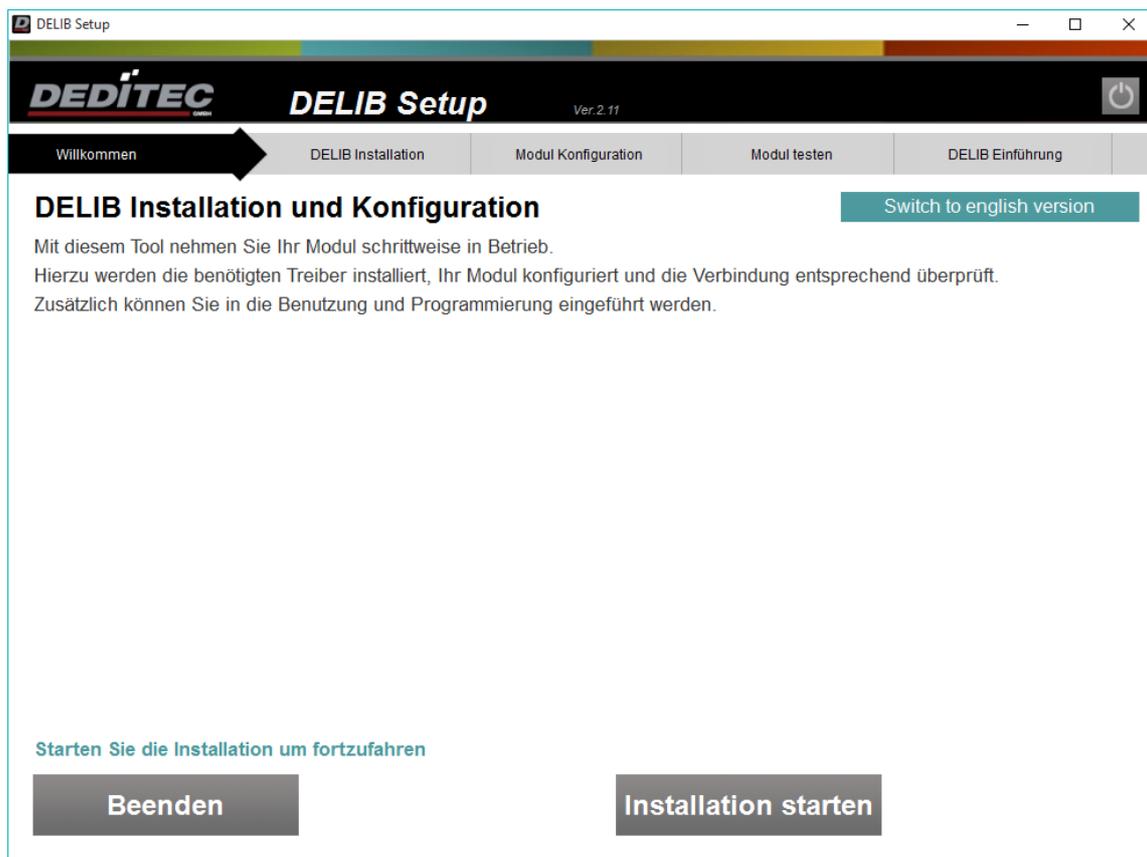
Das DELIB Setup führt Sie durch die Installation unserer DELIB Treiberbibliothek.

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.

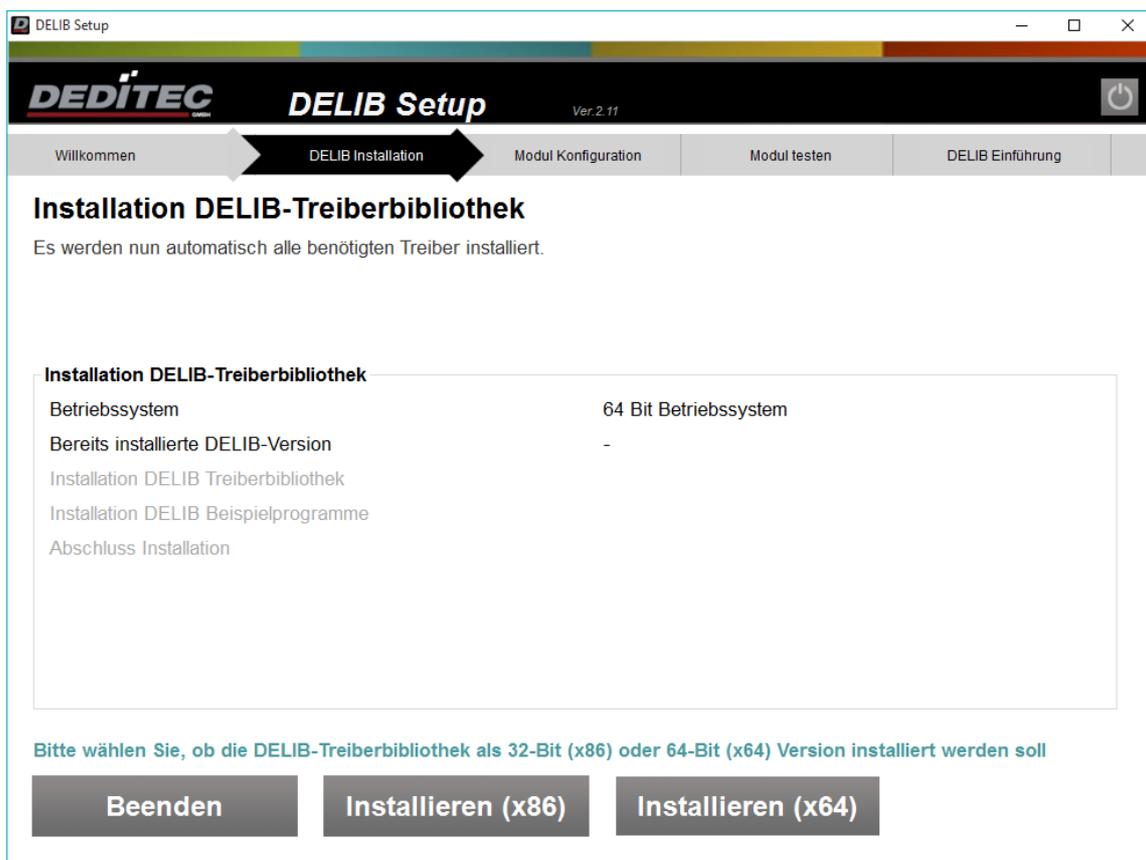
Link: <http://www.deditec.de/de/downloads/delib-treiber-bibliothek.html>

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

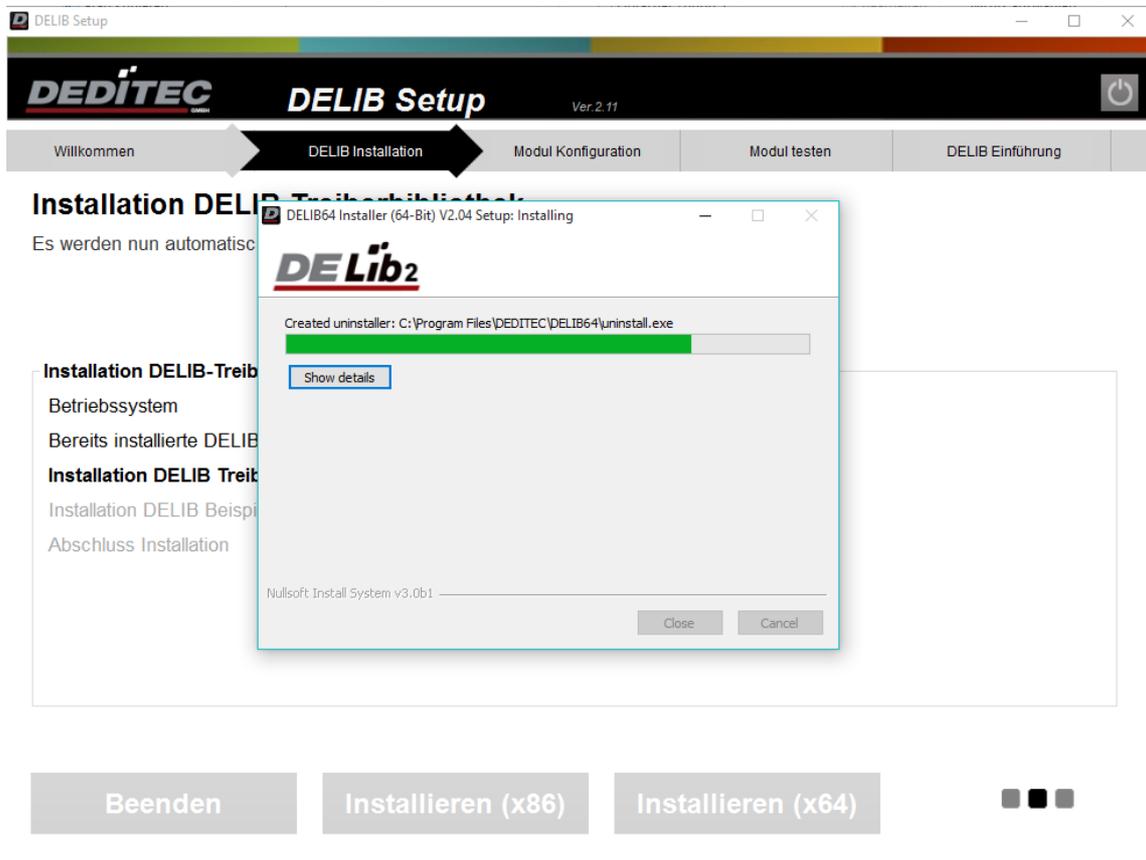


Das DELIB Setup prüft das Betriebssystem und ob bereits Versionen der DELIB Treiberbibliothek installiert sind.

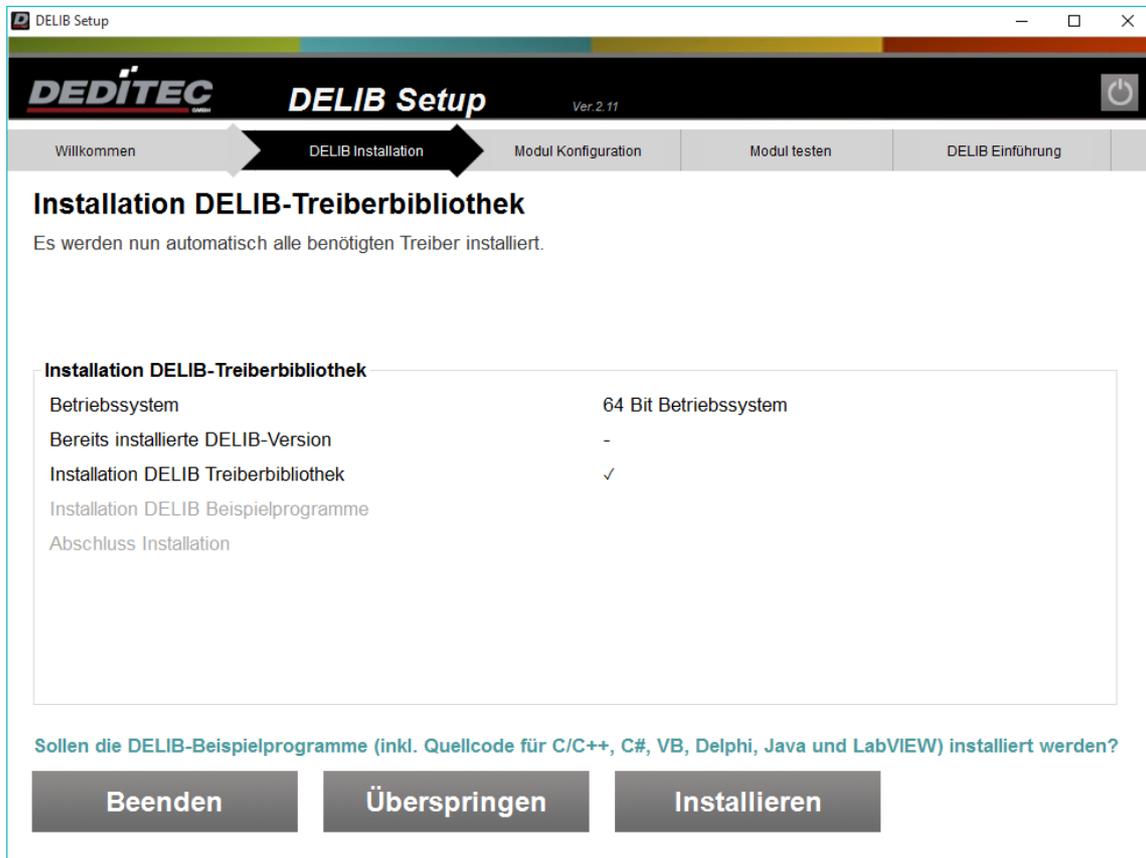
Anschliessend können Sie wählen ob Sie die 32-Bit oder 64-Bit Version der DELIB Treiberbibliothek installieren möchten.



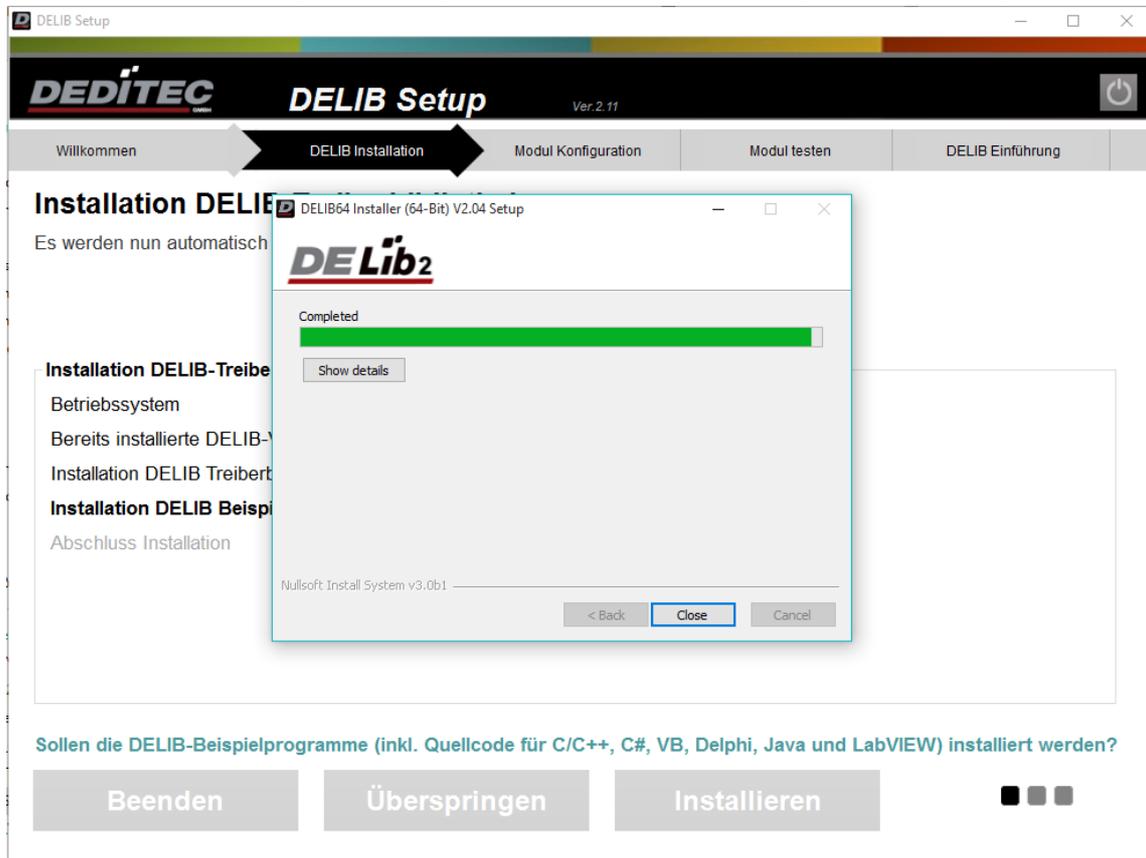
Installationsfortschritt der Treiberbibliothek.



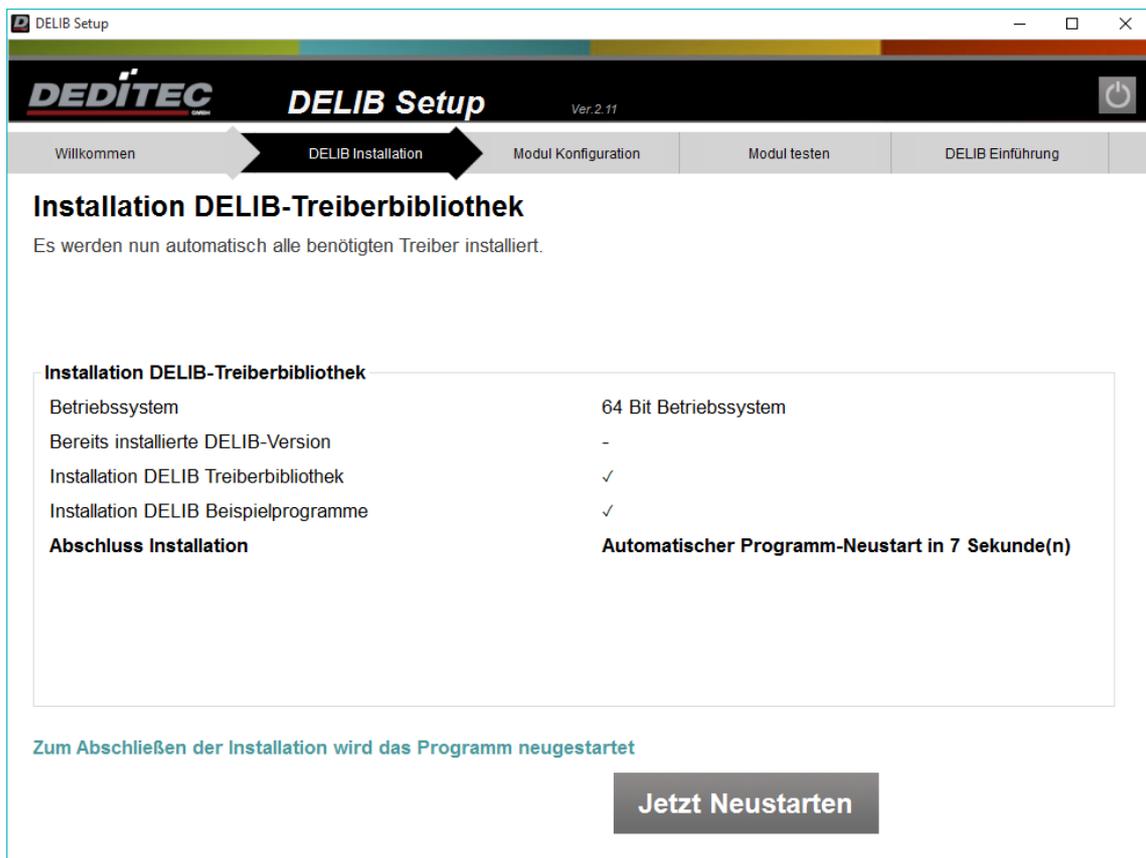
Zusätzlich können Sie wählen, ob Sie die DELIB-Beispielprogramme installieren.



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.



## 4.2.3. DELIB Configuration Utility

### 4.2.3.1. 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

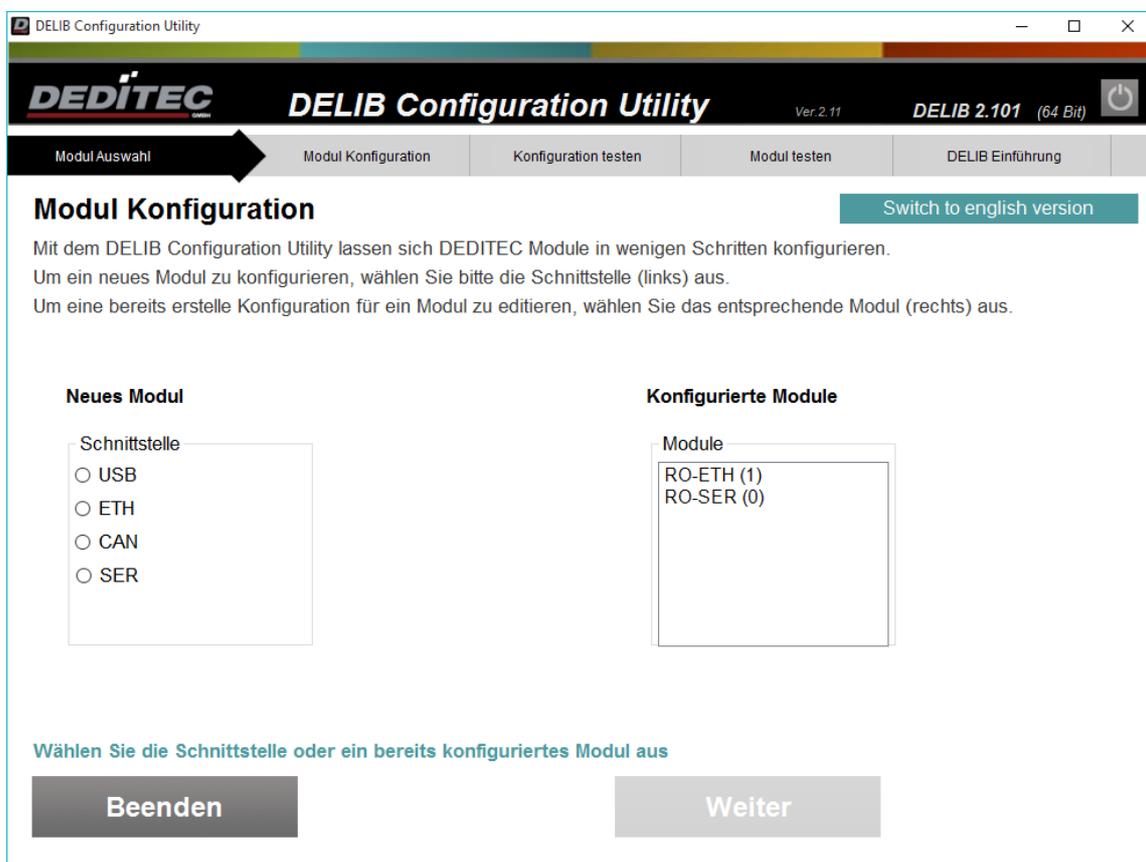
**64-Bit:** C:\Program Files\DEDITEC\DELIB64\programs\delib-configuration-utility\_x64.exe

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

### 4.2.3.2. 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.



#### 4.2.3.2.1. 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 product name 'DELIB Configuration Utility', version 'Ver. 2.19', and 'DELIB 2.19 (64 Bit)'. The main menu has five tabs: 'Modul Auswahl', 'Modul Konfiguration' (active), '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.' A 'Manual' icon is visible on the right. The 'Gefundene USB-Module' section contains a table with two columns: 'Module' and 'Nr.'. The table lists 'USB-OPT./REL-8' with '0'. To the right of the table are three input fields: 'Aktuelle Modul-Nr.' (0), 'Neue Modul-Nr.' (1), and 'Übertragungsversuche (für alle USB-Module)' (5). A 'Neue Modul-Nr. setzen' button is located between the second and third input fields. At the bottom of the window, there are three large buttons: 'Hauptmenü', 'Test', and 'Fertig'.

Module	Nr
USB-OPT./REL-8	0

Aktuelle Modul-Nr.

Neue Modul-Nr.

Übertragungsversuche (für alle USB-Module)

Neue Modul-Nr. setzen

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-Nr. 0.

Mit **"Neue Module-Nr. setzen"** wird die aktuell ausgewählte **Neue Modul-Nr.** 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.

#### 4.2.3.2.1.1. 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-016 und USB-RELAIS-8) im System, ist keine Konfiguration nötig, da die Produkte über die ID eindeutig identifizierbar sind.

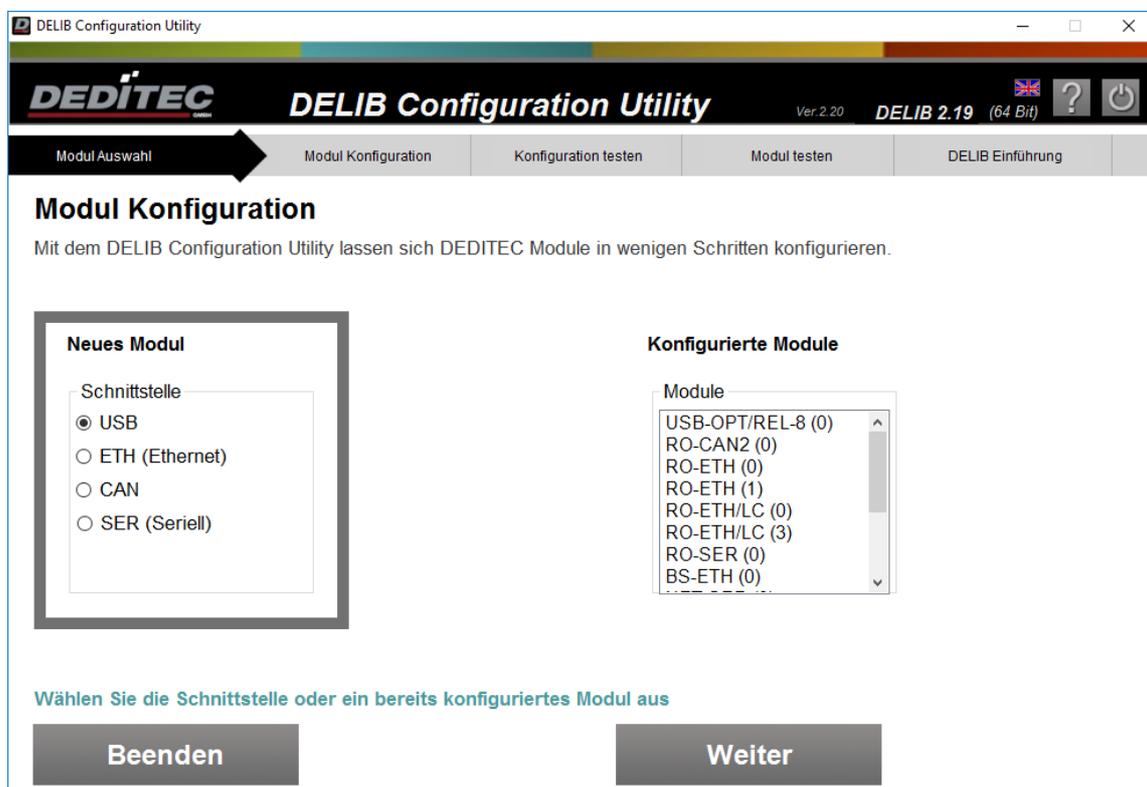
### Beispiel

Folgendes Beispiel zeigt die Konfiguration von zwei USB-OPTOIN-8 Modulen im gleichem 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".

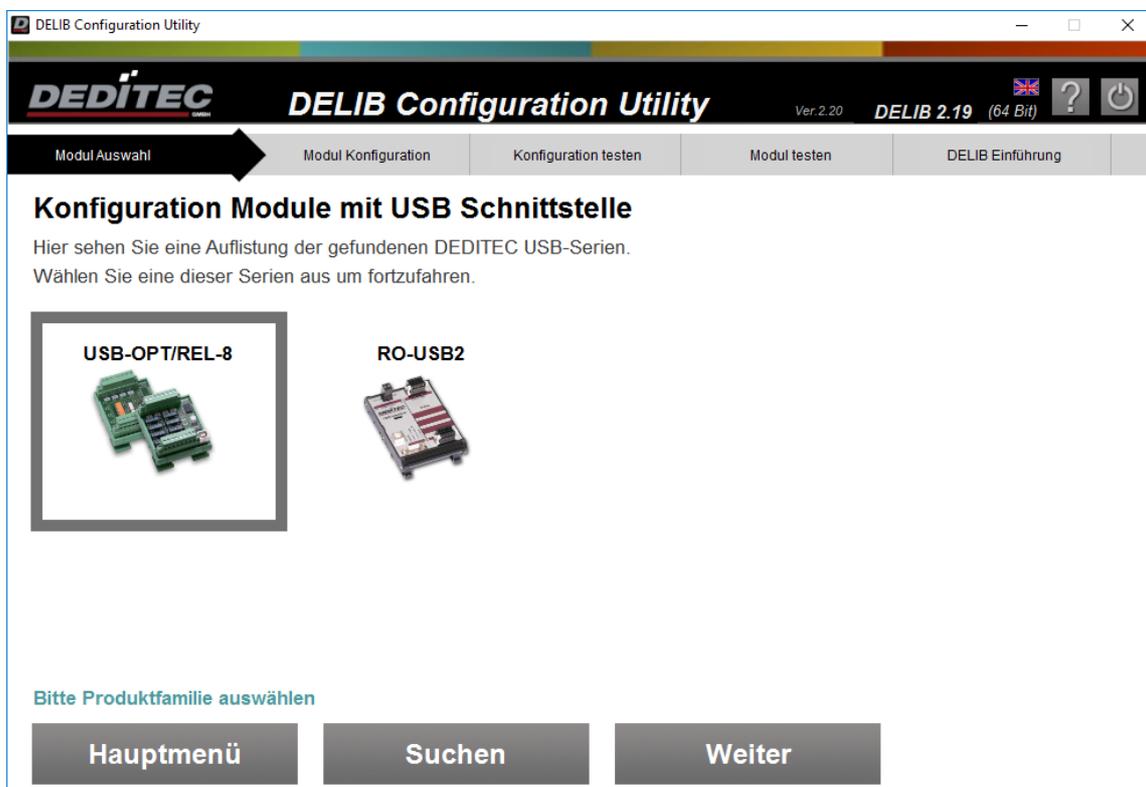


## Schritt 2

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.

The screenshot shows the 'DELIB Configuration Utility' window. The title bar includes the DEDITEC logo, the product name 'DELIB Configuration Utility', version 'Ver. 2.20', and 'DELIB 2.19 (64 Bit)'. The main menu has five items: 'Modul Auswahl', 'Modul Konfiguration' (highlighted with a black arrow), 'Konfiguration testen', 'Modul testen', and 'DELIB Einführung'. The main content area is titled 'Konfiguration Module mit USB Schnittstelle' and contains the following text: 'Hier kann die Modul-Nr. des ausgewählten Moduls geändert werden. Die Modul-Nr. dient zur Unterscheidung mehrerer identischer USB-Module einer USB-Serie.'

**Gefundene USB-Module**

Module	Nr
USB-OPT/REL-8	0

**Aktuelle Modul-Nr.**

**Neue Modul-Nr.**

**Neue Modul-Nr. setzen** (button)

**Übertragungsversuche (für alle USB-Module)**

**Aktualisieren** (button)

At the bottom, there are three large buttons: 'Hauptmenü', 'Test', and 'Weiter'.

DELIB Configuration Utility

**DEDITEC** **DELIB Configuration Utility** Ver. 2.20 DELIB 2.19 (64 Bit) ? [Power]

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

### Konfiguration Module mit USB Schnittstelle

Hier kann die Modul-Nr. des ausgewählten Moduls geändert werden.  
Die Modul-Nr. dient zur Unterscheidung mehrerer identischer USB-Module einer USB-Serie.

**Gefundene USB-Module**

Module	Nr
USB-OPT/REL-8	0

Aktuelle Modul-Nr.

Neue Modul-Nr.

Übertragungsversuche (für alle USB-Module)

**Modul-Kommunikation ist OK! Aktuelle Modul-Firmware ist 2.10**  
Automatische Suche in 1 Sekunde(n)

## 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

DELIB Configuration Utility

**DELIB Configuration Utility** Ver. 2.20 DELIB 2.19 (64 Bit)

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

### Konfiguration Module mit USB Schnittstelle

Hier sehen Sie eine Auflistung aller gefundenen Module der USB-OPT/REL-8-Serie.  
Wählen Sie eines dieser Module aus um fortzufahren.

**Gefundene USB-Module**

Module	Nr
USB-OPT/REL-8	0
USB-OPT/REL-8	1

[Bitte Modul auswählen](#)

Hauptmenü Suchen

## 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
```

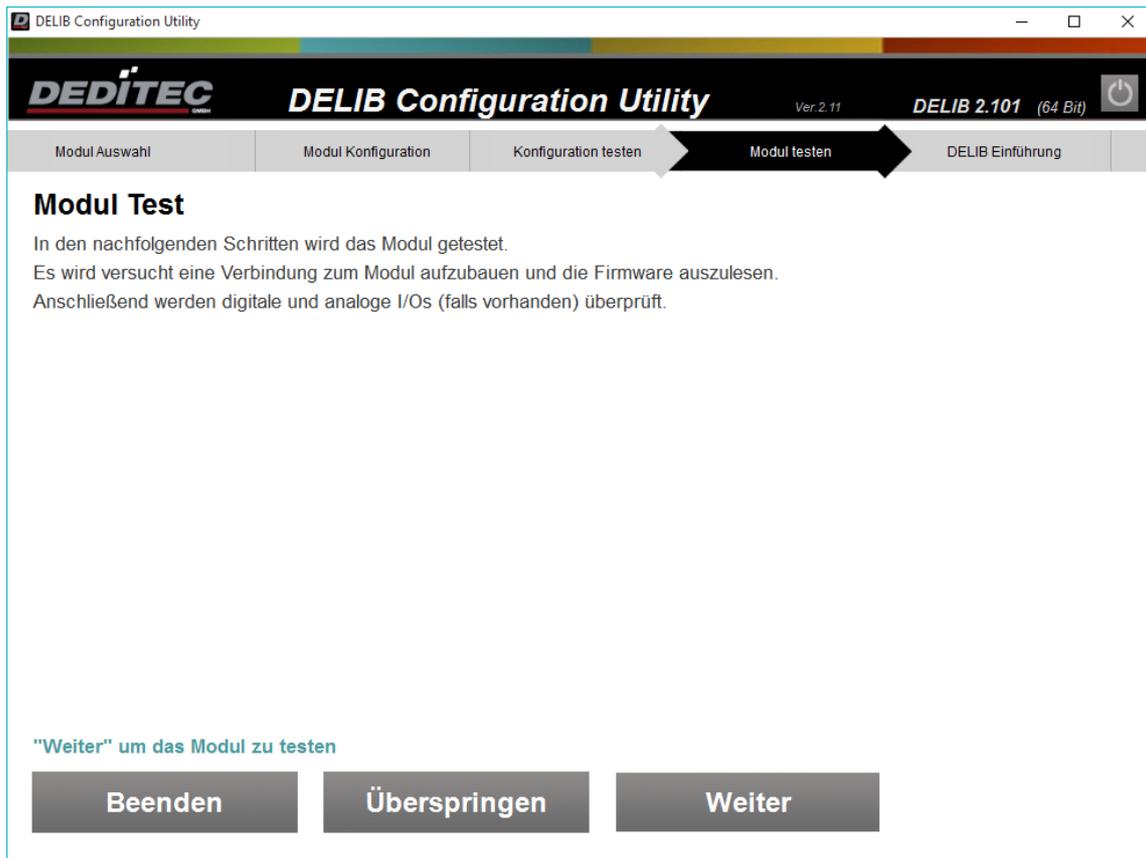
### Hinweis:

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.

### 4.2.3.3. Modul testen

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



Test der Firmware und Anzeige der Modul-Info.

The screenshot shows the 'DELIB Configuration Utility' window. The title bar includes the logo 'DEDITEC' and the text 'DELIB Configuration Utility Ver. 2.11 DELIB 2.101 (64 Bit)'. The main menu has five steps: 'Modul Auswahl', 'Modul Konfiguration', 'Konfiguration testen', 'Modul testen', and 'DELIB Einführung'. The 'Modul testen' step is currently active and highlighted with a black arrow. Below the menu, the 'Modul Test' section contains the following text: 'In den nachfolgenden Schritten wird das Modul getestet. Es wird versucht eine Verbindung zum Modul aufzubauen und die Firmware auszulesen. Anschließend werden digitale und analoge I/Os (falls vorhanden) überprüft.' Below this text is a table with the following data:

Modul Test	
Überprüfe Modul-Verbindung	✓
Firmware Version abfragen	2.12
Modul Konfiguration abfragen	✓
Digitale Ausgänge	
Digitale Eingänge	
Analoge Ausgänge	
Analoge Eingänge	

Below the table, there is a question: 'Möchten Sie die Modul Konfiguration ansehen?' with two buttons: 'Nein' and 'Ja'.

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

DT\_ModuleInfo ×

<b>General</b>	<b>Digital I/O</b>	<b>Analog I/O</b>	<b>Special</b>
SW_FEATURE_1 e3373007 HW_INTERFACE_1 01000003	Digital Inputs 8 Digital Outputs 8 Digital In-/Outputs 0 Digital Input FlipFlops 8 Digital Input Counter 8 Pulse Gen Outputs 0 CNT8 0 Digital PWM Outputs 0	Analog Inputs 16 Analog Outputs 4 Temperature Inputs 4	Stepper 2
Firmware-Revision 2.11 Main-Module: RO			
<b>Features-General</b>	<b>Features-Digital I/O</b>	<b>Features-Analog I/O</b>	<b>Features-Special</b>
Supported by FW OK Dev IO registry error OK AD FIFO OK Set-Clr Bit Commands OK EEPROM RN23 - EEPROM E2_2K - DX1 Mode - Support Channel Names OK HW-INT Supported by FV OK ETH OK CAN - RS232 - RS485 - USB1 - USB2 -	DI Commands OK DI CNT Commands OK DI CNT Latch Feature - DI FF Commands OK DO Commands OK DO Time Commands OK PWM Commands - TTL Commands - PulseGen Commands - CNT8 Commands - Auto-Off Timeout Commar OK	DA Commands OK AD Commands OK Pt100 Commands OK	Watchdog Commands - Stepper Commands OK
<b>Submodule-Info's</b>			
Sub: 0 - RO-AD16DA4 FW:2.11		Sub: 1 - RO-O8_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.

The screenshot shows the 'DELIB Configuration Utility' software window. The title bar indicates 'Ver. 2.11' and 'DELIB 2.101 (64 Bit)'. The main menu includes 'Modul Auswahl', 'Modul Konfiguration', 'Konfiguration testen', 'Modul testen', and 'DELIB Einführung'. The 'Modul testen' step is currently active and highlighted with a black arrow.

### Modul Test

In den nachfolgenden Schritten wird das Modul getestet.  
Es wird versucht eine Verbindung zum Modul aufzubauen und die Firmware auszulesen.  
Anschließend werden digitale und analoge I/Os (falls vorhanden) überprüft.

Modul Test	
Überprüfe Modul-Verbindung	✓
Firmware Version abfragen	2.11
Modul Konfiguration abfragen	✓
<b>Digitale Ausgänge</b>	<b>wird ausgeführt ..</b>
Digitale Eingänge	
Analoge Ausgänge	
Analoge Eingänge	

Below the table, there are eight channel status indicators labeled CH 1 through CH 8. Each indicator consists of a red 'OFF' button and a green 'ON' button. CH 1 has the 'ON' button highlighted in green. CH 2 through CH 8 have the 'OFF' button highlighted in red.

"Weiter" zum Setzen des nächsten Ausgangs, "Fertig" um diesen Test zu beenden

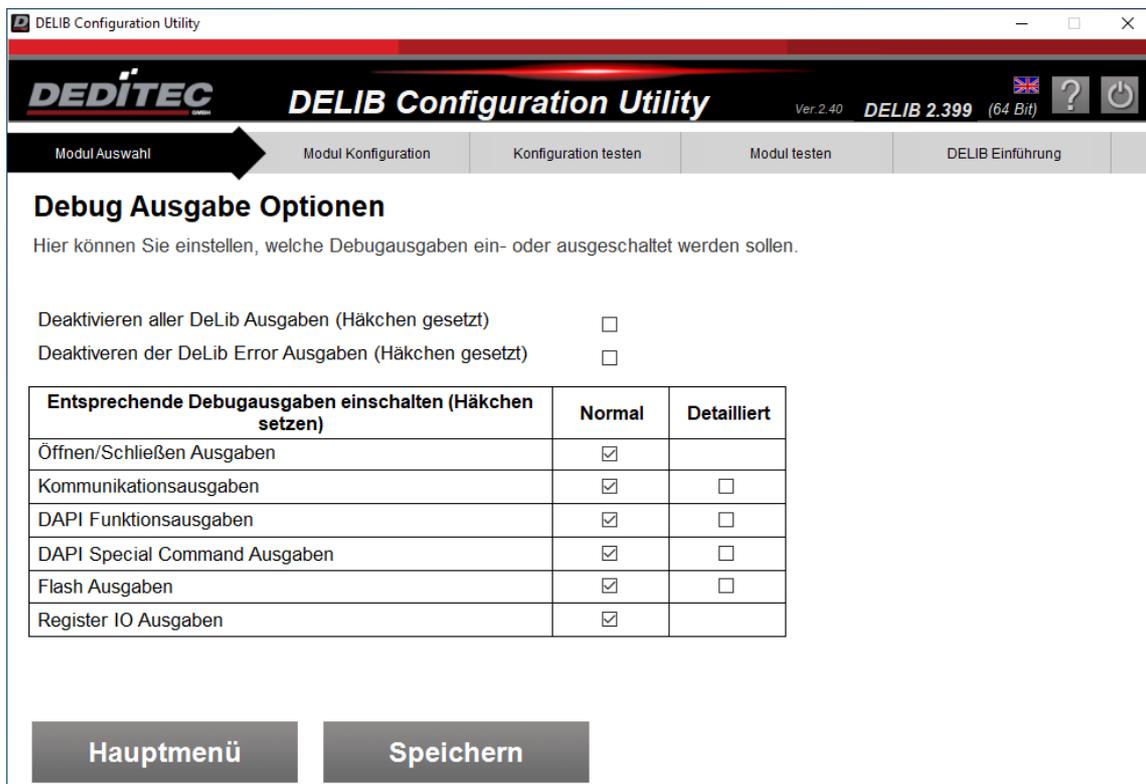
**Fertig**      **Weiter**

Wurden alle Tests erfolgreich durchlaufen, ist das Produkt einsatzbereit.

#### 4.2.3.4. 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 includes the DEDITEC logo, the application name 'DELIB Configuration Utility', the version 'Ver 2.40 DELIB 2.399 (64 Bit)', and icons for help and power. The main menu has five tabs: 'Modul Auswahl' (selected), 'Modul Konfiguration', 'Konfiguration testen', 'Modul testen', and 'DELIB Einführung'. The 'Debug Ausgabe Optionen' section contains the following text and controls:

Hier können Sie einstellen, welche Debugausgaben ein- oder ausgeschaltet werden sollen.

Deaktivieren aller DeLib Ausgaben (Häkchen gesetzt)

Deaktivieren der DeLib Error Ausgaben (Häkchen gesetzt)

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"/>	

At the bottom, there are two buttons: 'Hauptmenü' and 'Speichern'.

## 4.2.4. Benutzung des Modulselectors

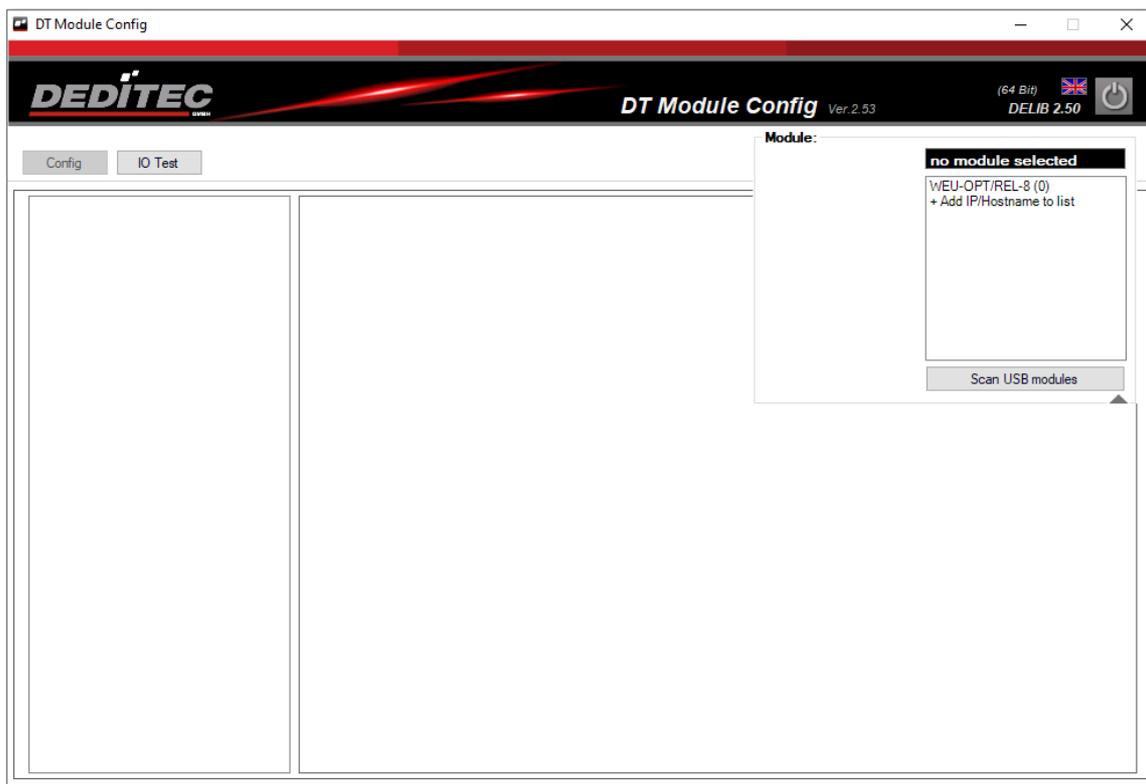
Um unsere Produkte mit der DEDITEC-Software benutzen zu können, müssen diese über den Modul Selector ausgewählt werden.

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

### 1. per 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 falls gewünscht vornehmen.

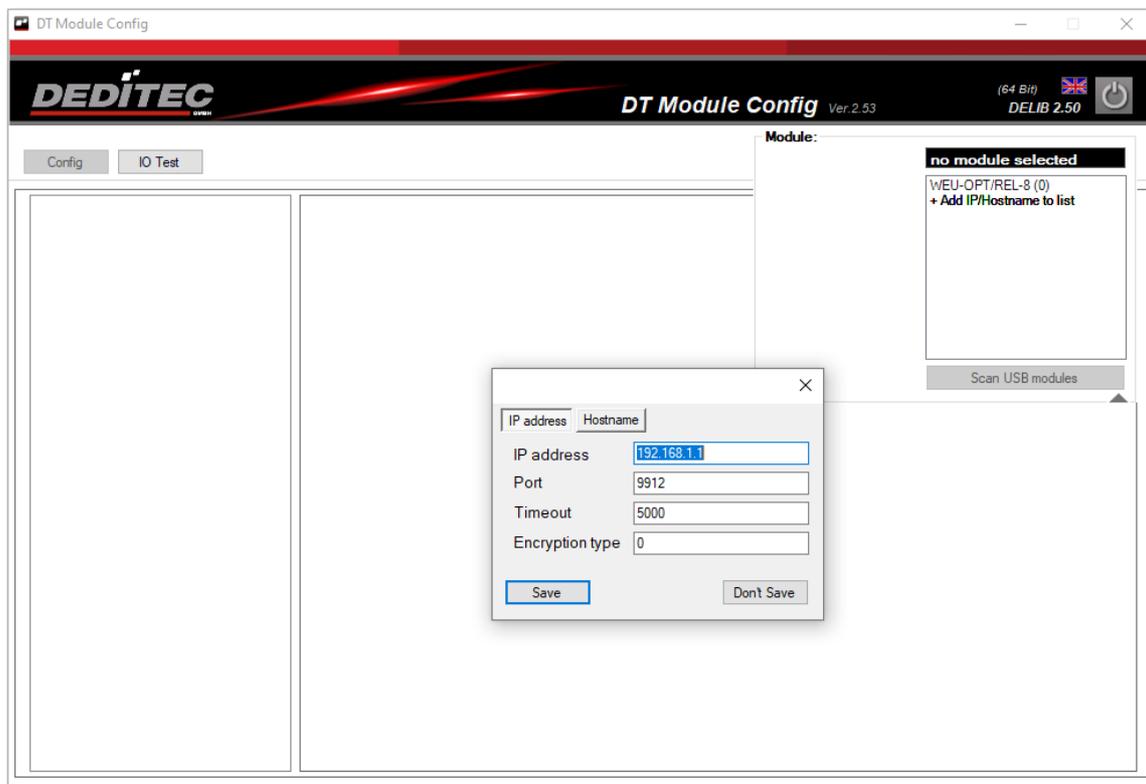


## 2. per Ethernet

Sollten Sie Ihr Modul über die Ethernet Schnittstelle angeschlossen haben, können Sie das Modul direkt über die im Netzwerk eingebundene IP-Adresse finden.

Fragen Sie hierfür gegebenenfalls Ihren Systemadministrator.

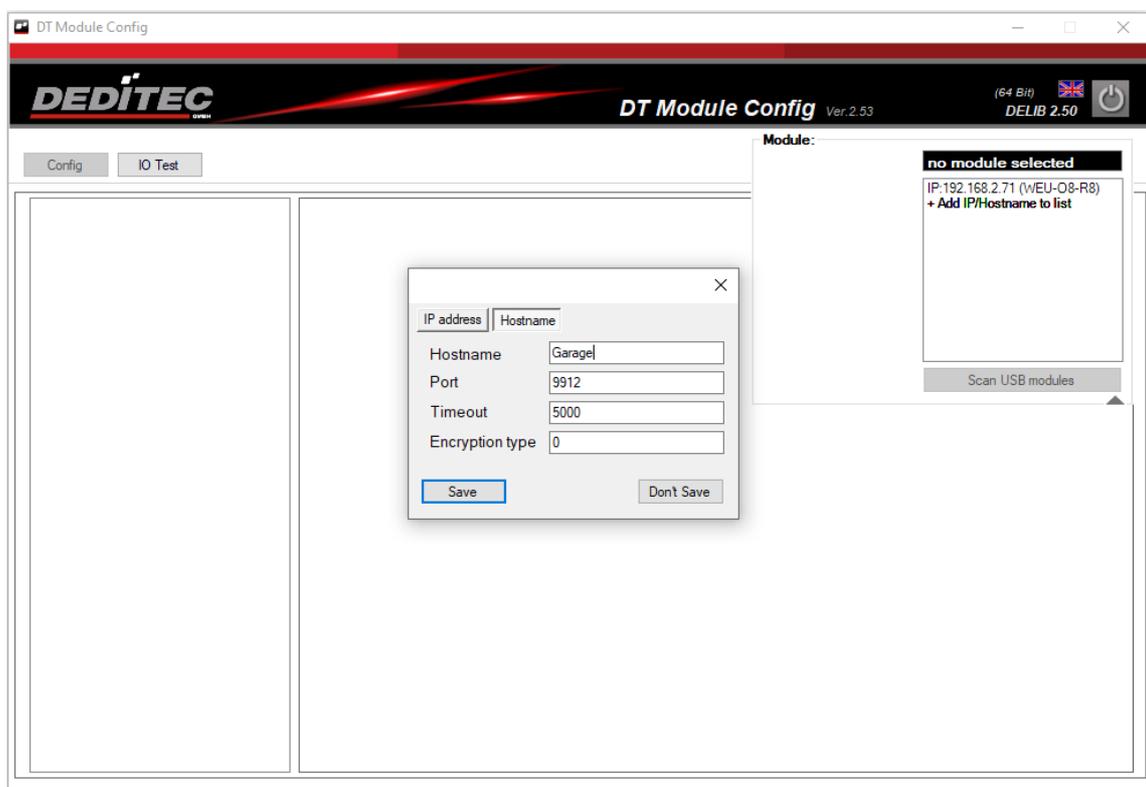
Klicken Sie im Modul Selector auf "Add IP/Hostname to list", tragen Sie dort die automatisch erhaltene IP-Adresse unter dem Reiter "IP Adresse" ein und bestätigen Sie die Eingabe mit "Save".



Ist ihr Modul im DHCP-Modus (siehe Kapitel: **LAN Netzwerkkonfiguration**) können Sie Ihr Modul auch mit Hilfe des Board Namens verbinden.

Diesen finden sie Modul Config im Bereich "LAN - Netzwerkinformation".

Für eine Verbindung per Board Name klicken Sie im Modul Selector auf "Add IP/Hostname to list", tragen Sie dort den Namen unter dem Reiter "Hostname" ein und bestätigen Sie die Eingabe mit "Save".



### 3. per WiFi (nur bei WEU-Modulen)

Um das Modul per WiFi zu verbinden, muss das Modul im Vorfeld mit USB oder Ethernet verbunden werden.

Nun kann unter dem Menüpunkt WiFi-Konfiguration WiFi aktiviert werden. Die an das Modul vergeben IP-Adresse finden Sie unter WiFi-Info

### 4. per WPS (nur bei WEU-Modulen)

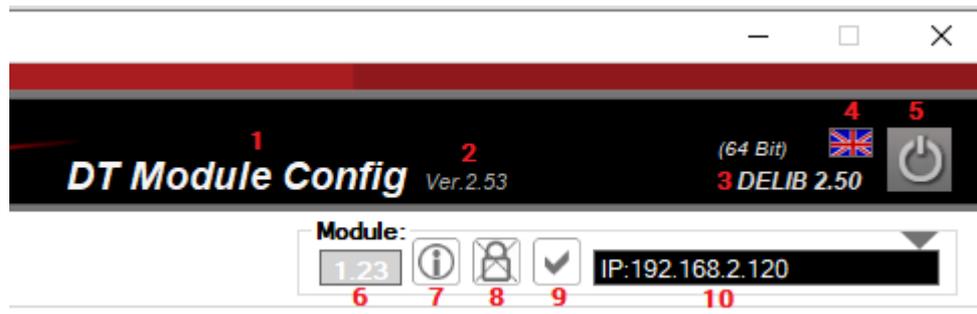
Ist das Modul noch nicht mit dem PC-Netzwerk verbunden, führen Sie den Verbindungsaufbau, wie im Kapitel **CFG-Taster** beschrieben, durch.

Klicken Sie danach im Modul Selector auf "Add IP/Hostname to list", tragen Sie dort die automatisch erhaltene IP-Adresse unter dem Reiter "IP Adresse" ein und bestätigen Sie die Eingabe mit "Save".

Fragen Sie hierfür gegebenenfalls Ihren Systemadministrator.

Sie können eine WPS-Verbindung auch mit Hilfe des Module Config starten. Führen Sie dafür die Schritte wie im Kapitel: **WiFi WPS-Verbindung** beschrieben aus.

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



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 Fahnsymbol 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 Modules (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.

DT_ModuleInfo									
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 FV	OK	CNT8 Commands	-						
ETH	OK	Auto-Off Timeout	OK						
CAN	-	Auto-Off Timeout Mask	OK						
RS232/485	-	FTDI Userbyte 6	0x0						
USB1	-								
USB2	-								

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

## 4.2.5. 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.

### 4.2.5.1. Modul Konfigurationen

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

#### 4.2.5.1.1. 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.

<b>Info</b>	
Hier finden Sie Informationen zu Ihren Moduleigenschaften	
Modul-Name	WEU-08-R8
Modul-ID	40 (dezimal) / 0x0028 (hex)
Firmware-Version	Ver. 1.23

FormCfmModuleOverview (Ver. 1.01)

### **Modul-Name**

Zeigt den Namen des aktuell verwendeten DEDITEC Modules an

### **Modul-ID**

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

### **Firmware-Revision**

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

#### 4.2.5.1.2. 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.

FormOffnModulIdent (Ver. 1.01)

#### 4.2.5.1.3. 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

FormCfoNetwork\ANInfo (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 Modules 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.

#### 4.2.5.1.4. LAN Netzwerkeinstellungen

Hier können Sie Änderungen an den Netzwerkeinstellungen des ausgewählten WEU-Modules 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"/>

FormCtnNetworkI ANConfIn (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 Moduleselectors**)

#### 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. Im Zweifel fragen Sie bitte Ihren zuständigen Systemadministrator.

### **Werkseinstellungen laden**

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

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

#### **4.2.5.1.5. 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"/>

FirmCFONetworkNTP (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

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

#### 4.2.5.1.6. I/O Kanal-Namen

Hier können Sie die Kanalnamen Ihres Haupt- bzw. Submodules 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"/>

FormCfnSubmoduleIOnames (Ver. 1.01)

Sie können hier sämtliche Kanäle Ihres Haupt- oder Submodules 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.

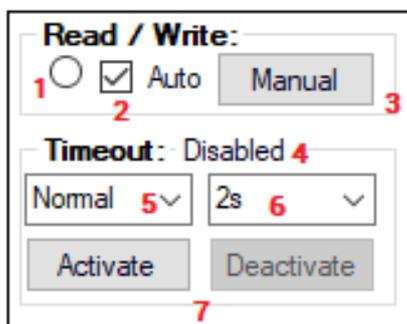
## 4.2.5.2. I/O-Test

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

### 4.2.5.2.1. 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.

#### 4.2.5.2.2. 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		State on/off	Counter	FlipFlop
Kanal 0	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="checkbox"/>	Kanal 8	<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 9	<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 10	<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 11	<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 12	<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 13	<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 14	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="checkbox"/>
Kanal 7	<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

FomIODigitalIn (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.

#### 4.2.5.2.3. 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

FomIODigitalOut (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

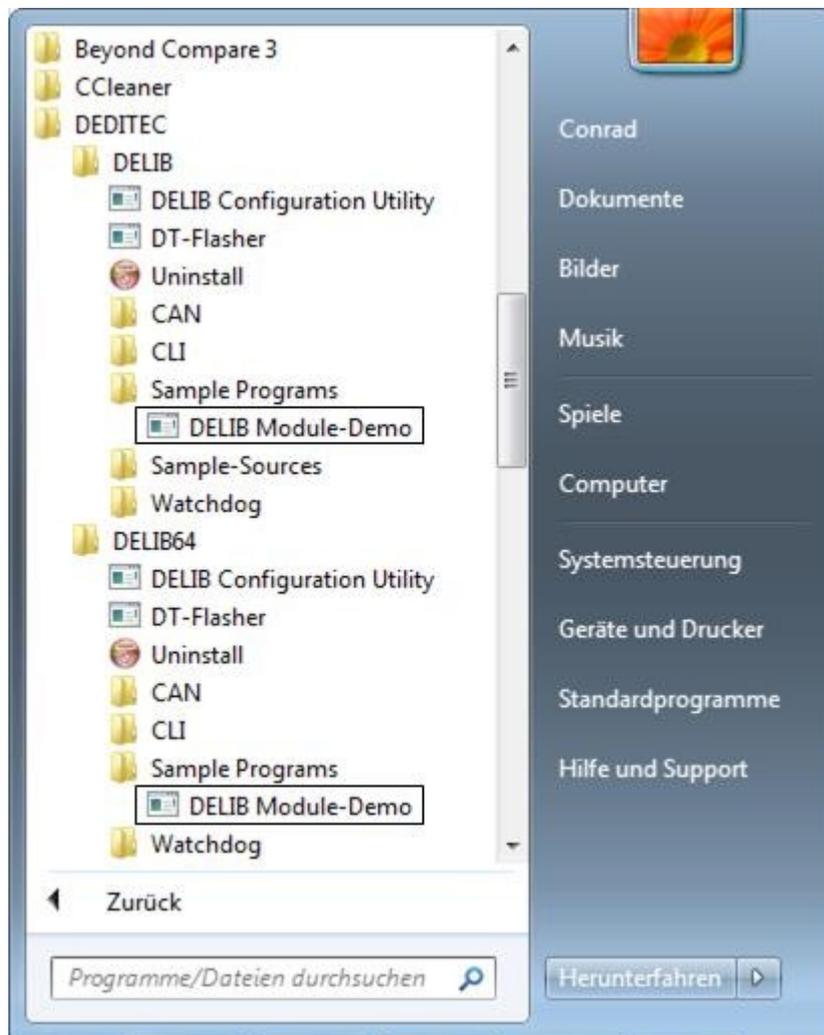
(Wird nur angezeigt, wenn es vom Modul unterstützt wird)

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.

#### 4.2.6. DELIB Module Demo

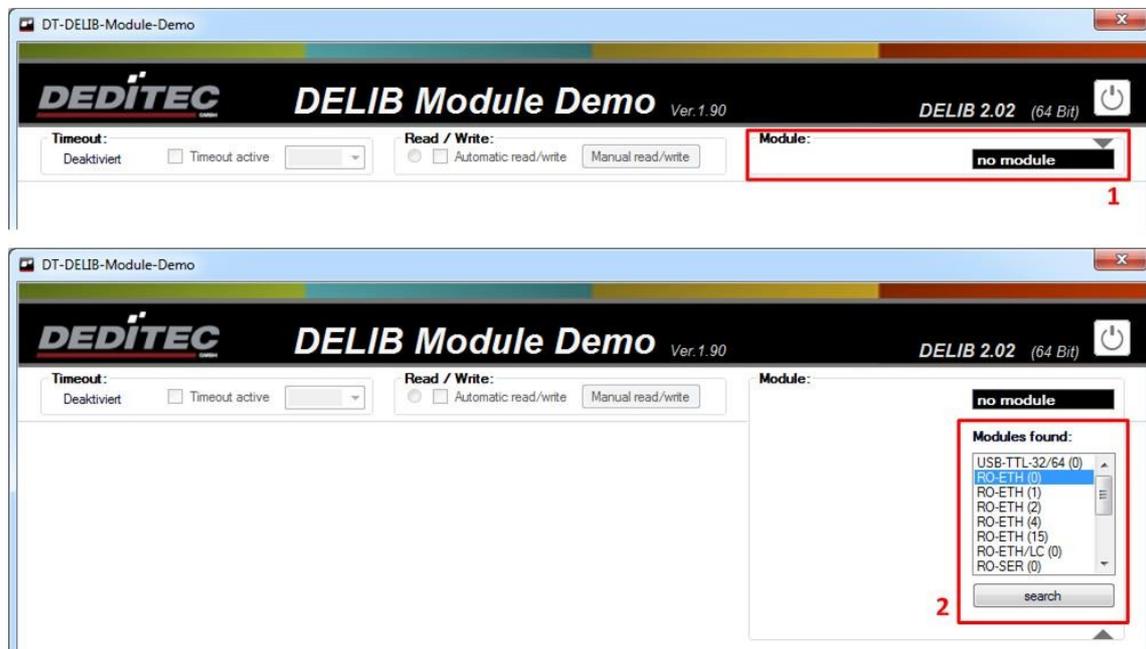


Nach Installation der **DELIB Treiberbibliothek** kann das Programm **DELIB Module Demo** auf folgendem Weg gestartet werden:

**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.

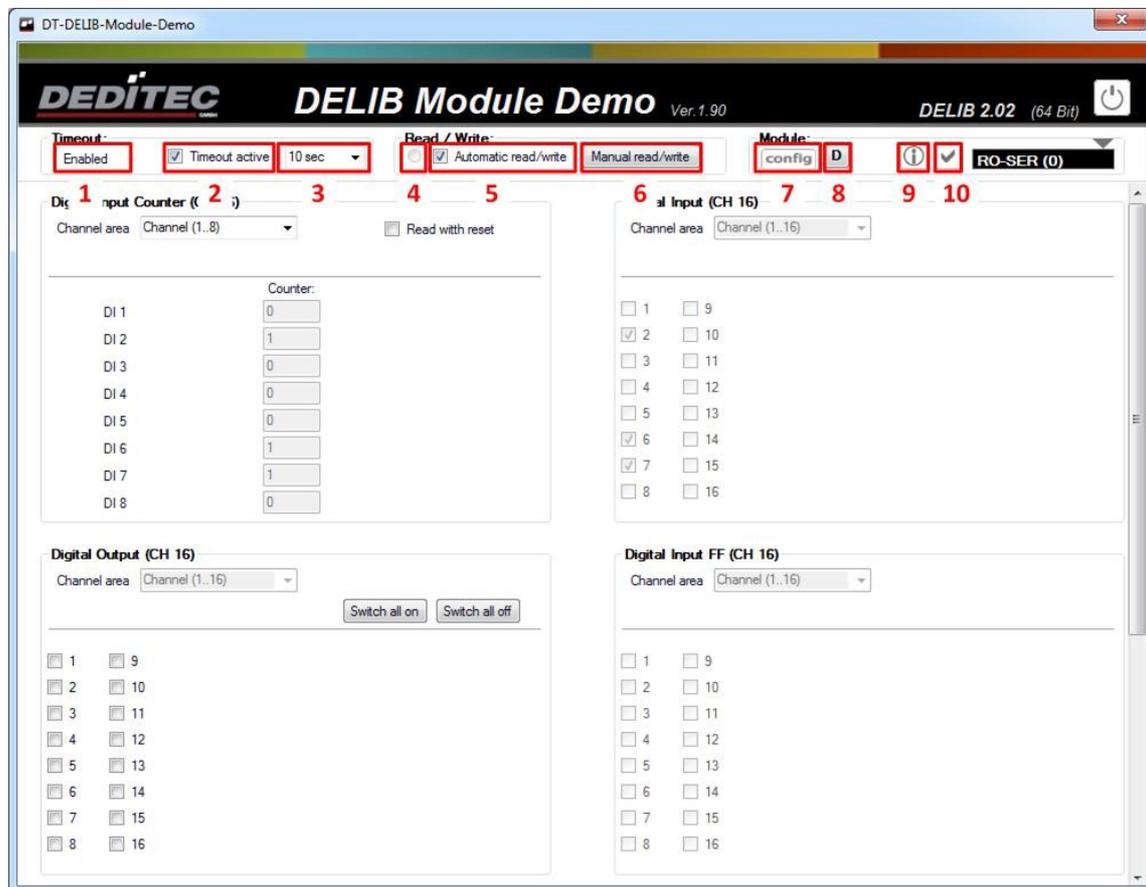
#### 4.2.6.1. Auswahl des Moduls



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.

## 4.2.6.2. 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.

#### 4.2.6.2.1. Module Info

The screenshot shows a window titled "DT\_ModuleInfo" with a close button in the top right corner. The window is divided into four main sections, each with a red border and a red number in the bottom right corner:

- 1. General:** SW\_FEATURE\_1: e300000f, HW\_INTERFACE\_1: 01000031, Firmware-Revision: 2.01, Main-Module: RO.
- 2. Digital I/O:** Digital Inputs: 16, Digital Outputs: 16, Digital In-/Outputs: 0, Digital Input FlipFlops: 16, Digital Input Counter: 16, Pulse Gen Outputs: 0, CNT8: 0, Digital PWM Outputs: 0.
- Analog I/O:** Analog Inputs: 0, Analog Outputs: 0, Temperature Inputs: 0.
- Special:** Stepper: 0.

Below these sections are four feature status sections:

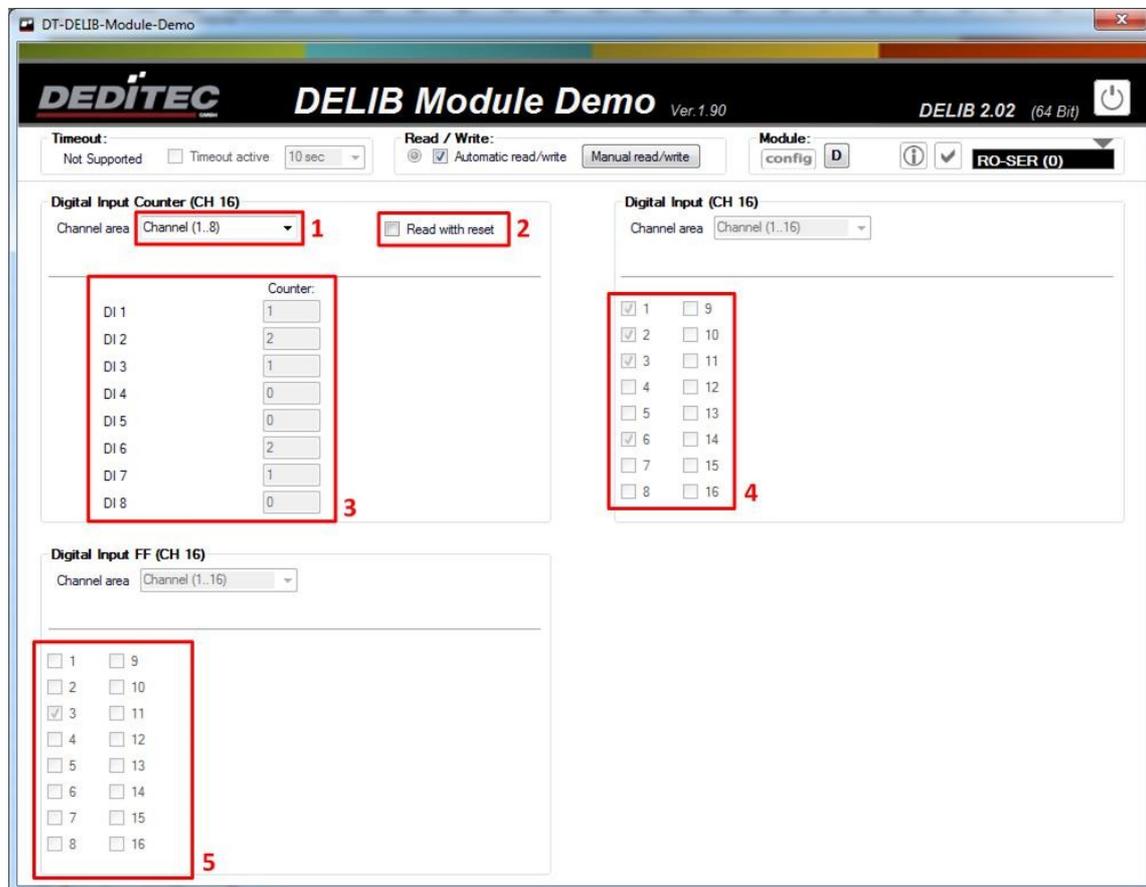
- 3. Features-General:** Supported by FW: OK, AD FIFO: OK, Set-Clr Bit Commands: OK, EEPROM Commands: OK, EEPROM E2\_2K: -, DX1 Mode: -, Support Channel Names: -, HW-INT Supported by FV: OK, ETH: -, CAN: -, RS232: -, RS485: OK, USB: OK.
- Features-Digital I/O:** DI Commands: OK, DI CNT Commands: OK, DI CNT Latch Feature: -, DI FF Commands: OK, DO Commands: OK, DO Time Commands: -, PWM Commands: -, TTL Commands: -, PulseGen Commands: -, CNT8 Commands: -, Auto-Off Timeout Commar: OK.
- Features-Analog I/O:** DA Commands: -, AD Commands: -, Pt100 Commands: -.
- Features-Special:** Watchdog Commands: -, Stepper Commands: -.

A small button labeled "E" with a red number "4" is located in the bottom right corner of the window.

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.

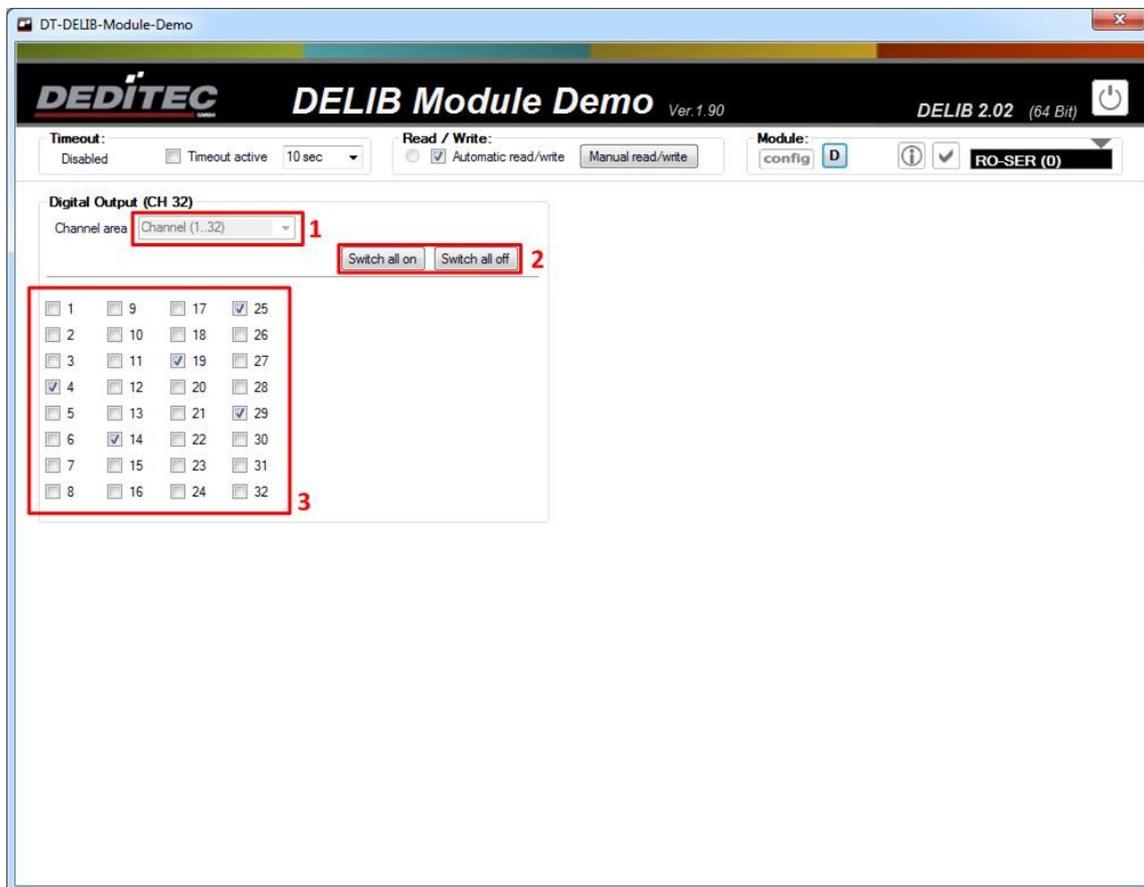
### 4.2.6.3. Digital Input



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).

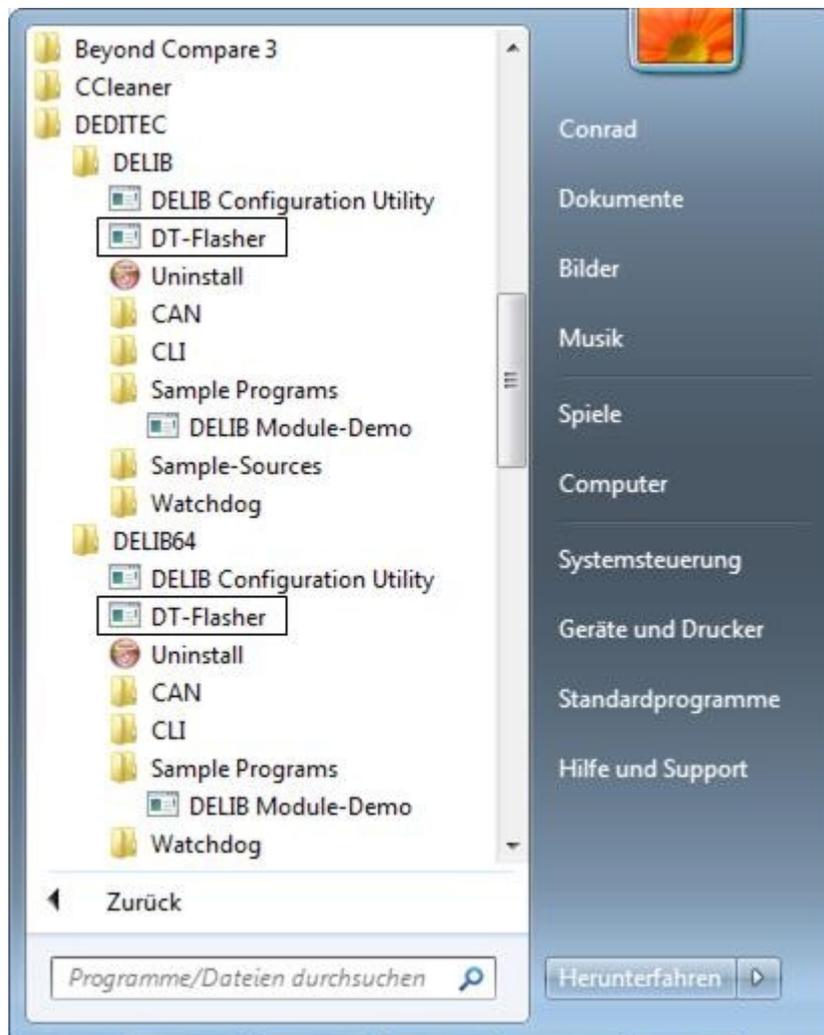
#### 4.2.6.4. Digital Output



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.

#### 4.2.7. 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.**

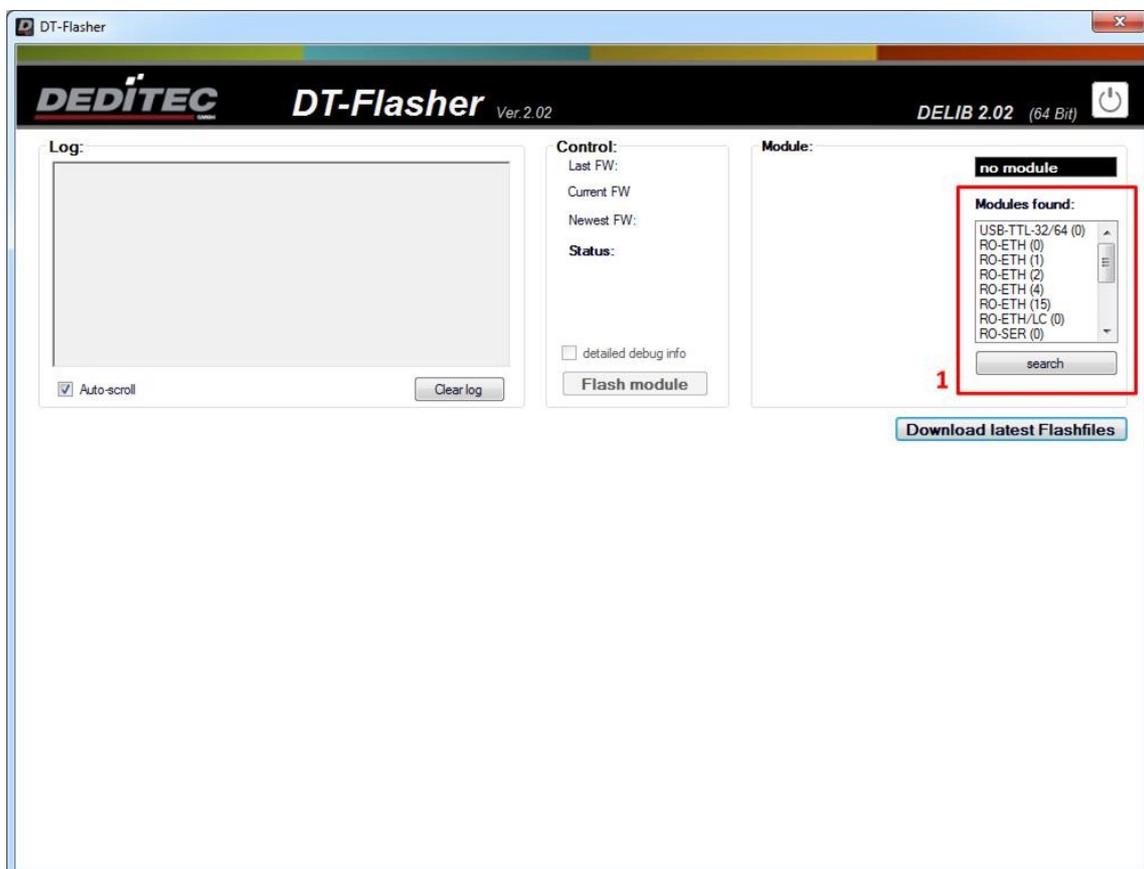
### 4.2.7.1. Über DEDITEC-Firmware

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.

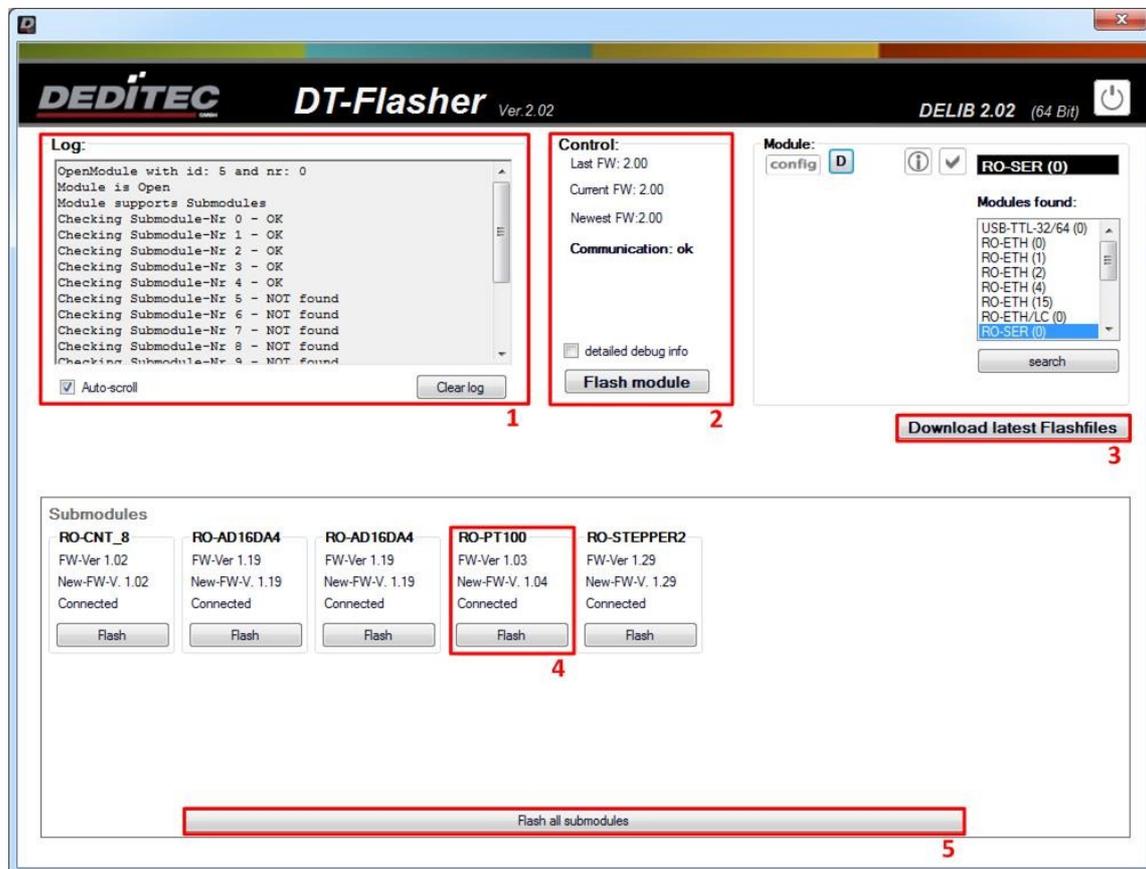
### 4.2.7.2. Auswahl des Moduls



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"

### 4.2.7.3. 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, sog. Flash-Files herunterladen.

4. **FW-Ver** 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.

#### 4.2.7.3.1. 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

### 4.3. 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:

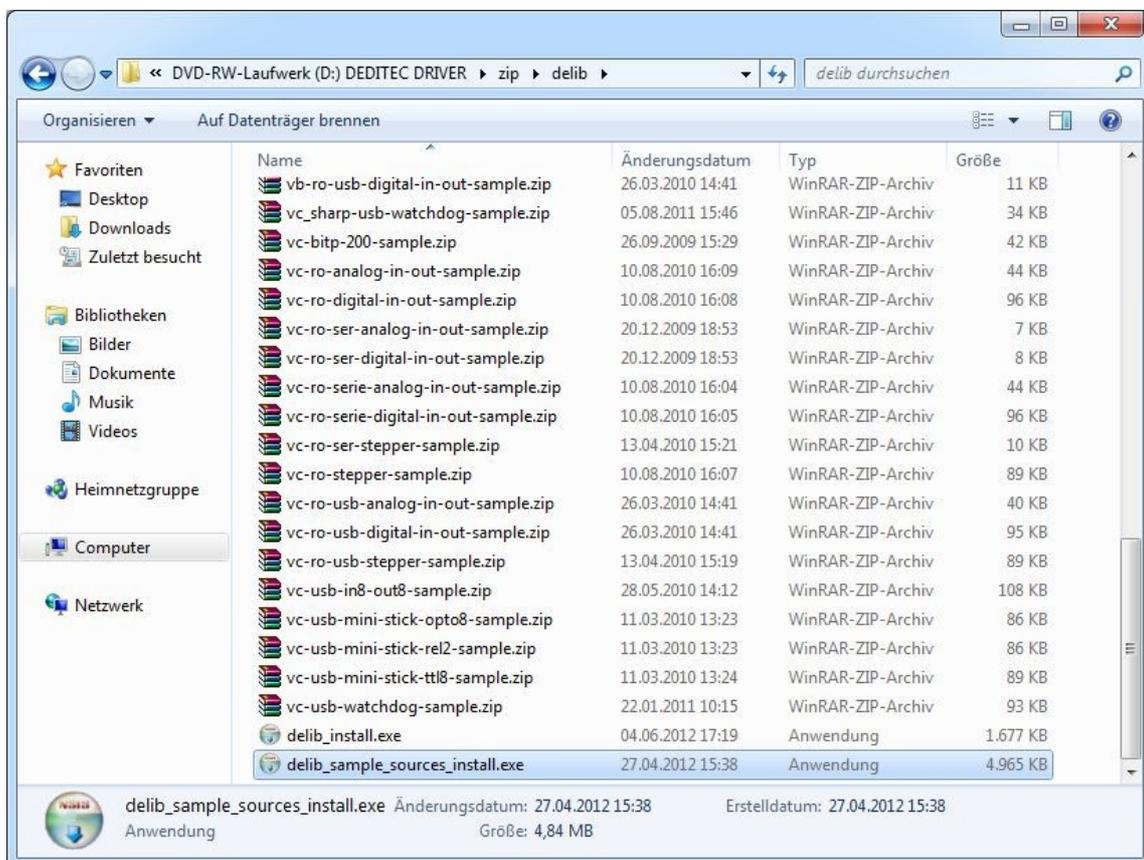
- C
- C++
- C#
- Delphi
- VisualBasic
- VB.NET
- MS-Office
- LabVIEW
- Java

### 4.3.1. 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**.

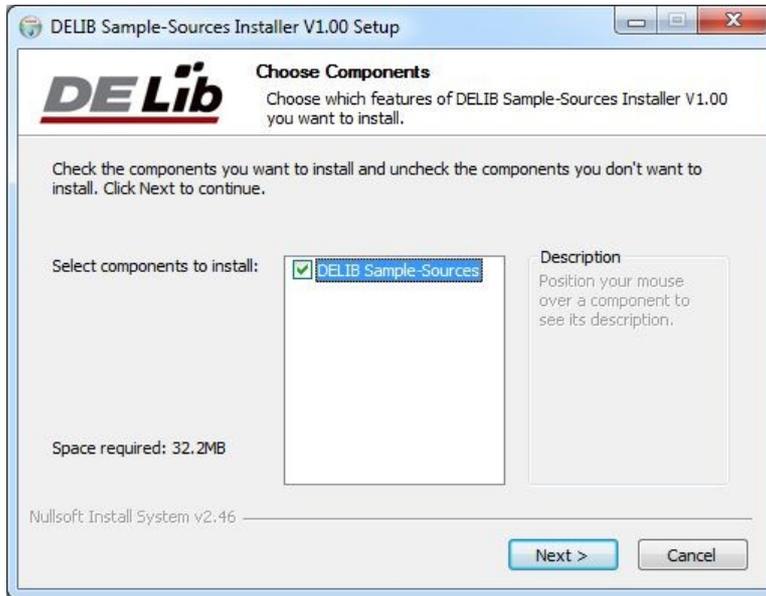
Eine aktuelle Version der Sample Sources finden Sie auch im Internet unter <http://www.deditec.de/de/delib>



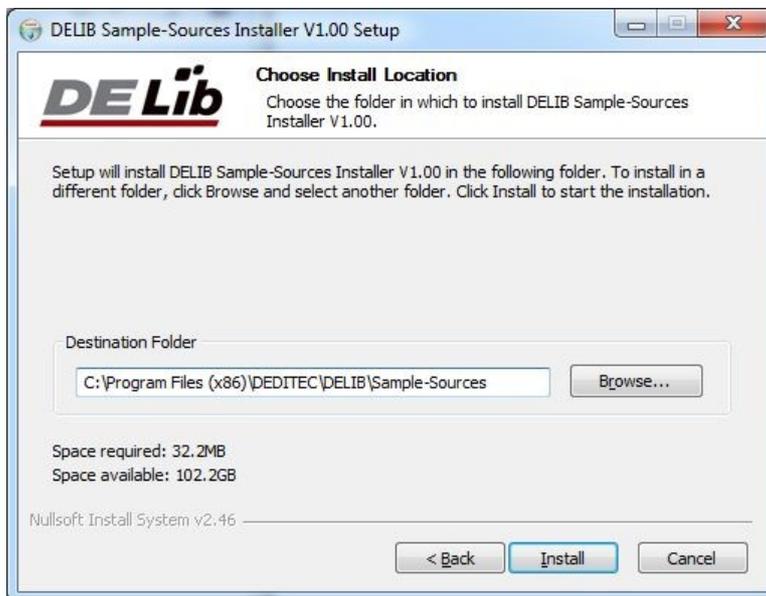
Startbild des DELIB Sample Sources Installer



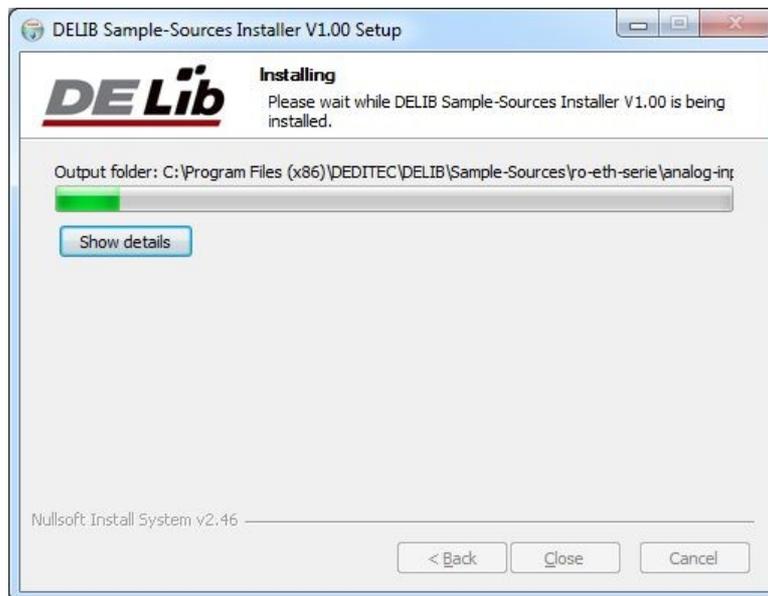
Drücken Sie **Next**.



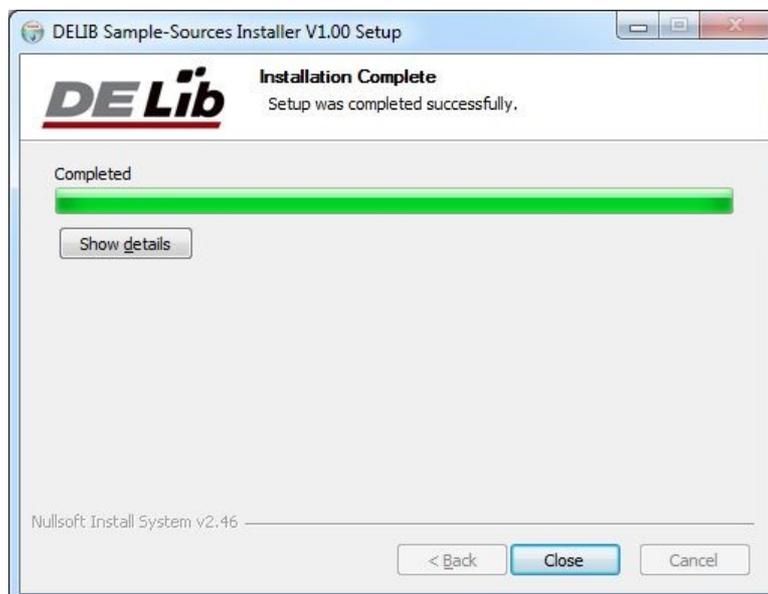
Wählen Sie den Installation Ordner und drücken Sie **Install**



Die DELIB Sample Sources werden nun installiert.



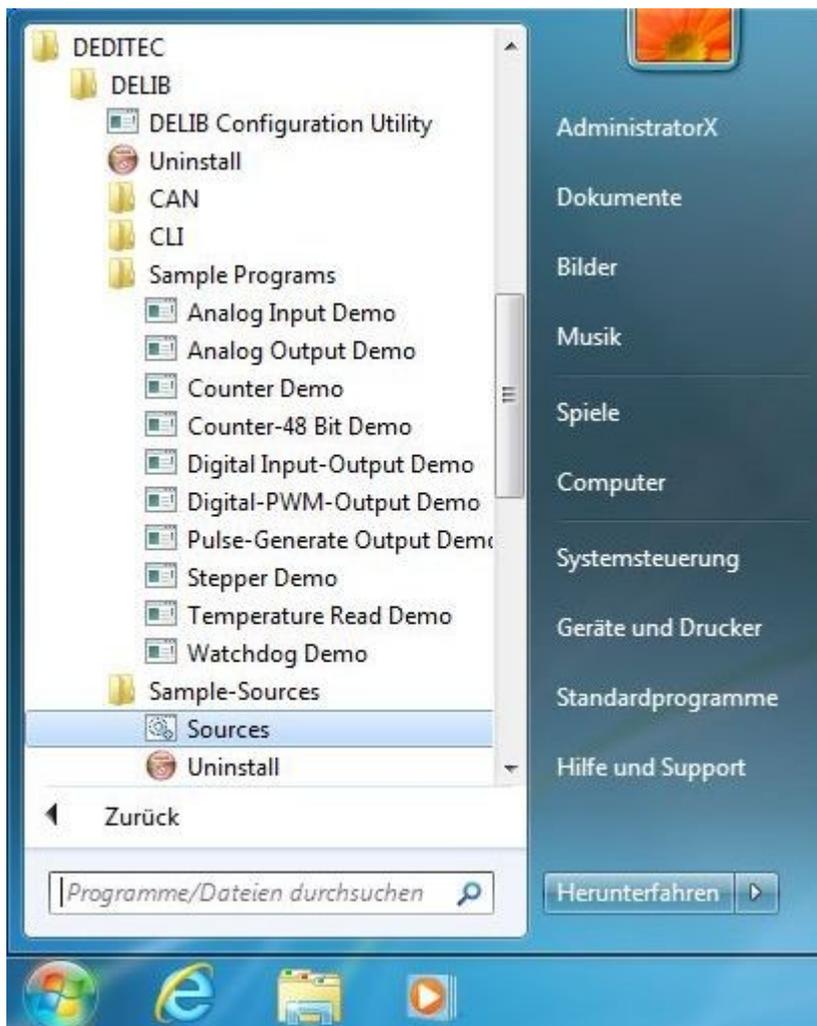
Die DELIB Sample Sources wurden erfolgreich installiert. Drücken Sie **Close** um die Installation zu beenden.



### 4.3.2. Benutzung der DELIB Sample Sources

Nach Installation der **DELIB Sample Sources** finden Sie diese unter

**Start -> Programme -> DEDITEC -> DELIB -> Sample-Sources -> Sources**

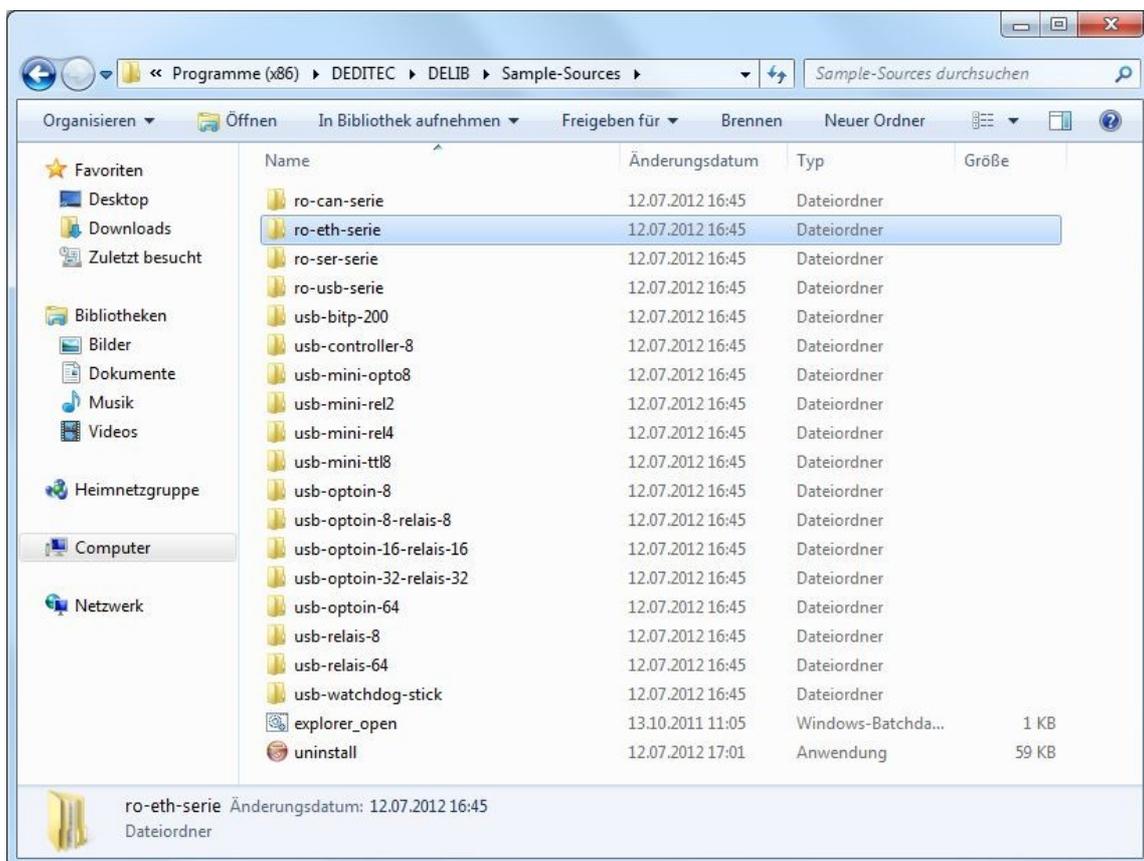


Nun öffnet sich der Windows-Explorer mit einer Übersicht aller Produkte für die ein Beispielprogramm verfügbar ist.

### 4.3.2.1. 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.

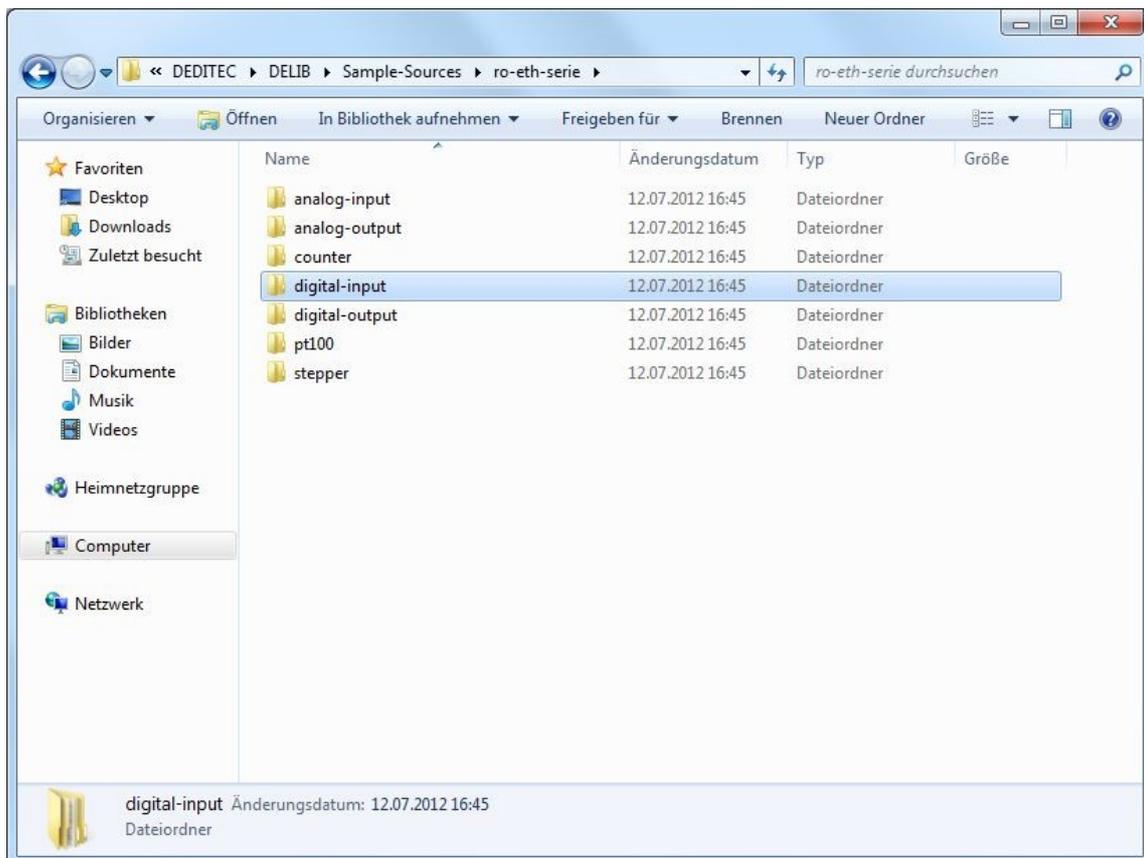
Da es sich um ein RO-ETH-Produkt handelt, wählen bzw. öffnen Sie den Ordner **ro-eth-serie**



#### 4.3.2.2. Schritt 2 - Kategorieauswahl

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

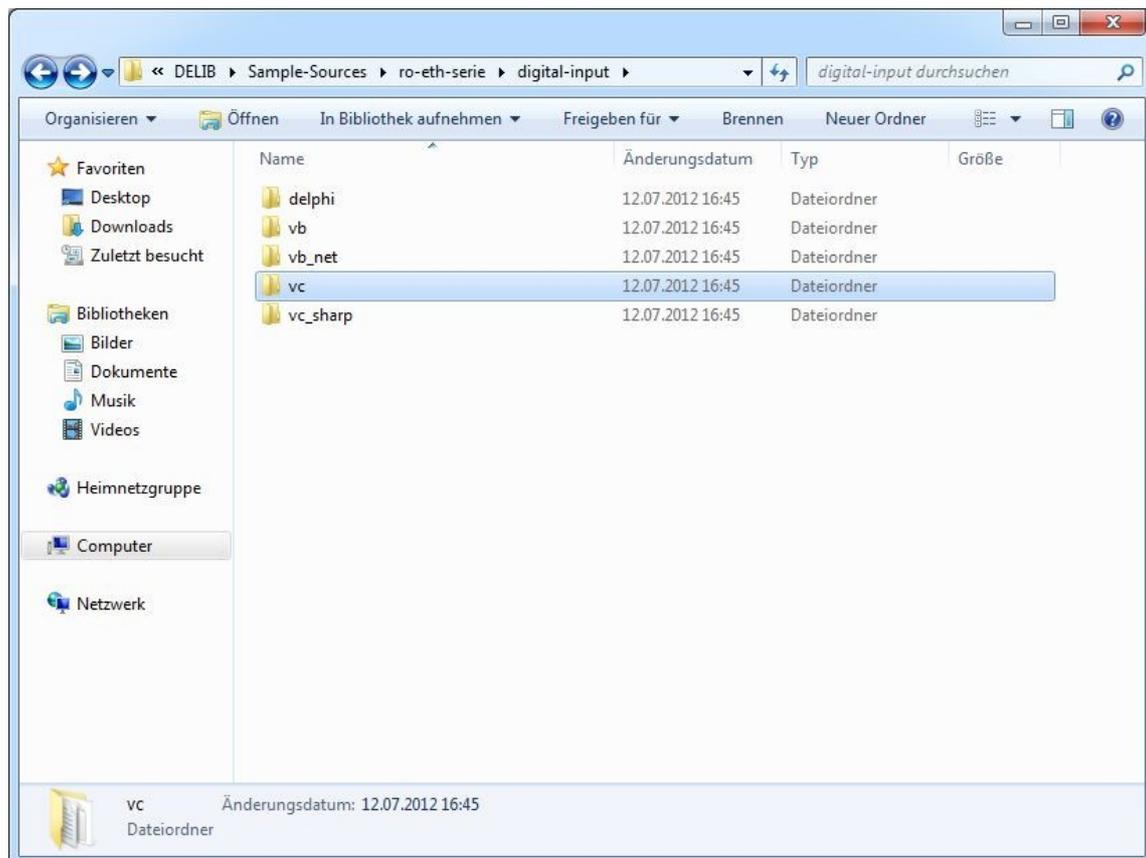
Da wir uns in diesem Beispiel auf die digitalen Eingänge konzentrieren, wählen Sie die Kategorie **digital-input**



### 4.3.2.3. Schritt 3 - Programmiersprachenauswahl

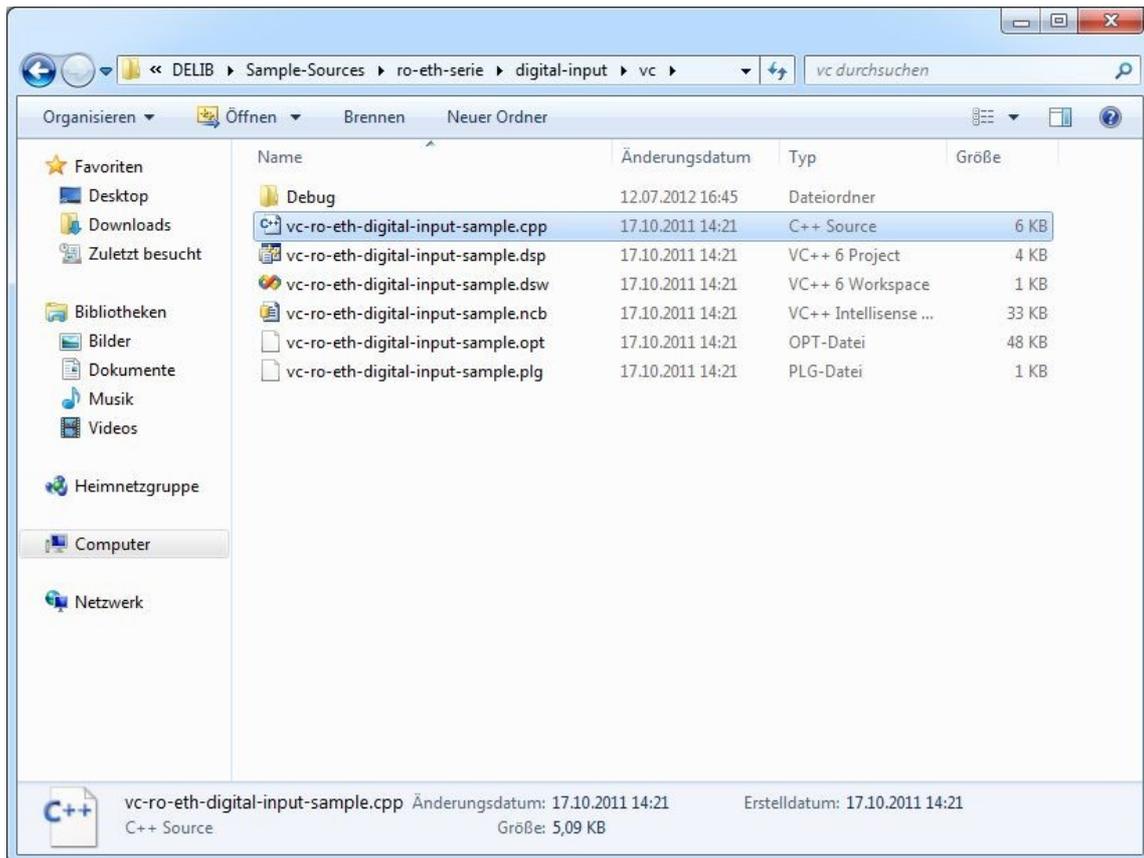
In diesem Schritt sehen Sie alle verfügbaren Programmierbeispiele der gewählten Kategorie, sortiert nach Programmiersprachen.

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

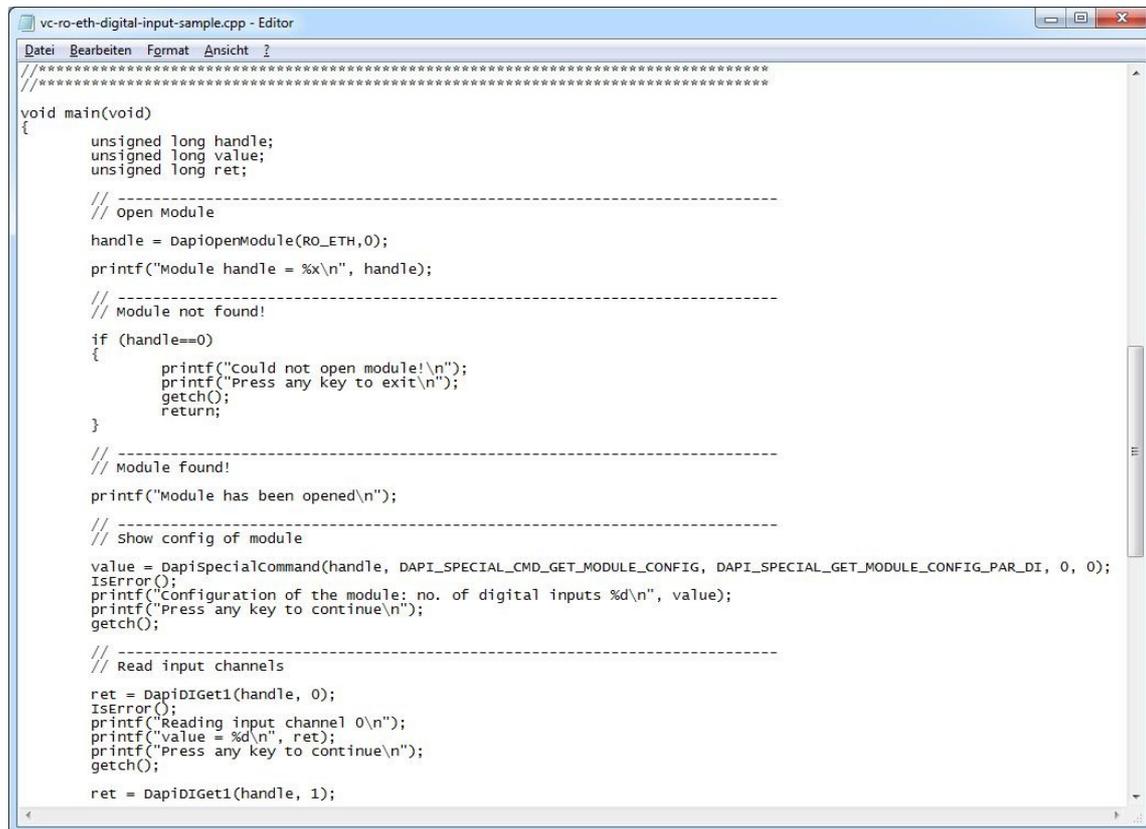


#### 4.3.2.4. Schritt 4 - Quellcode

Nach Auswahl der Programmiersprache erhalten Sie folgende Übersicht:



Den Quellcode des Beispielprogramm (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

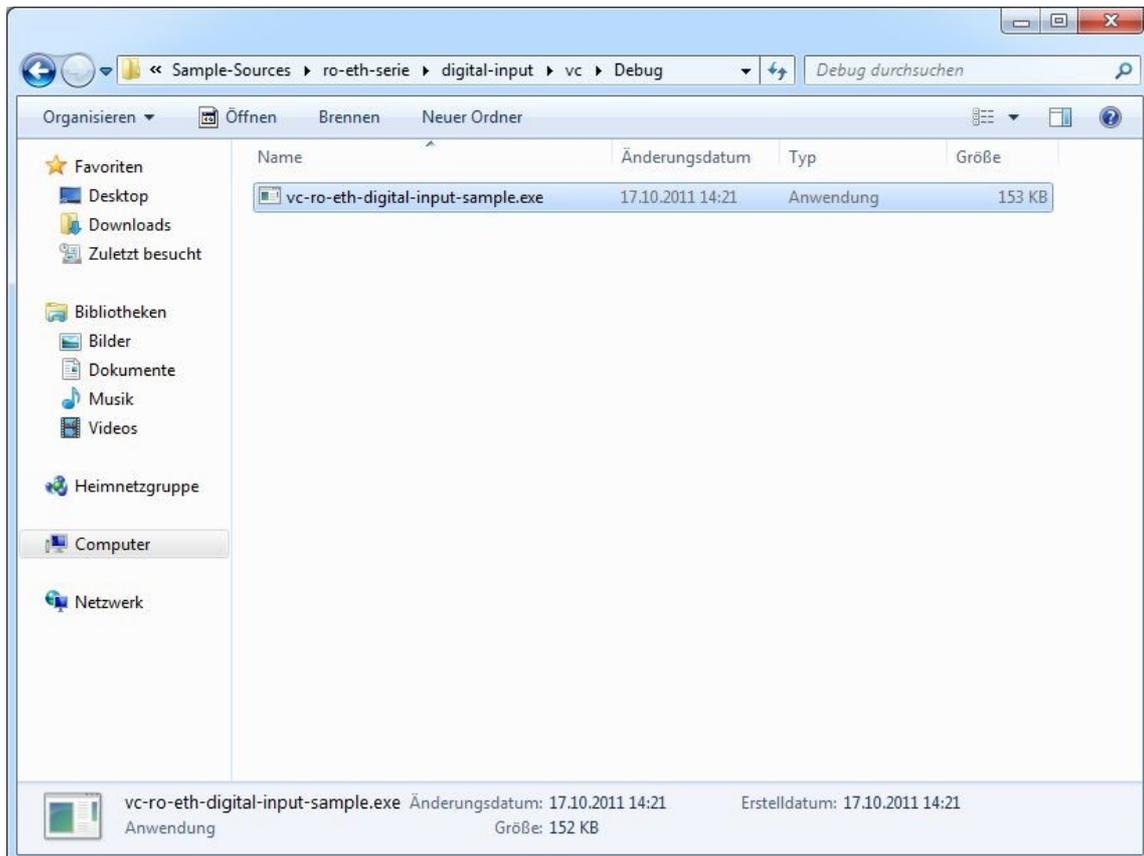
    value = DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_GET_MODULE_CONFIG, DAPI_SPECIAL_GET_MODULE_CONFIG_PAR_DI, 0, 0);
    isError();
    printf("configuration of the module; no. of digital inputs %d\n", value);
    printf("Press any key to continue\n");
    getch();

    // -----
    // Read input channels

    ret = DapiDGet1(handle, 0);
    isError();
    printf("Reading input channel 0\n");
    printf("value = %d\n", ret);
    printf("Press any key to continue\n");
    getch();

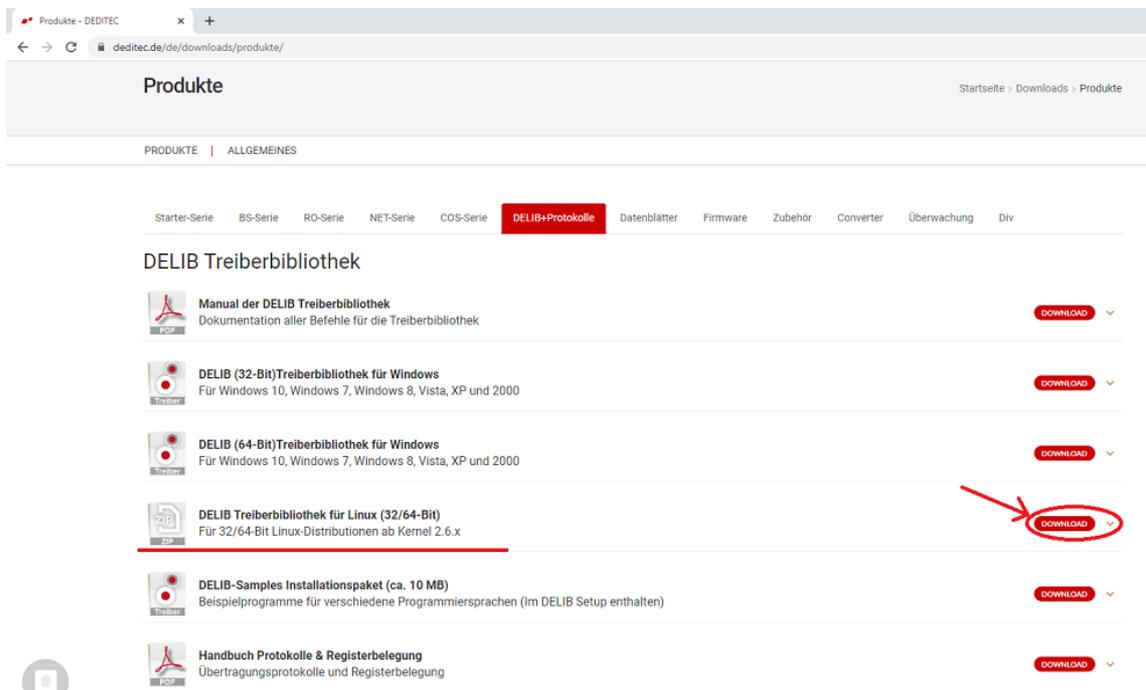
    ret = DapiDGet1(handle, 1);
}
```

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



## 4.4. 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.

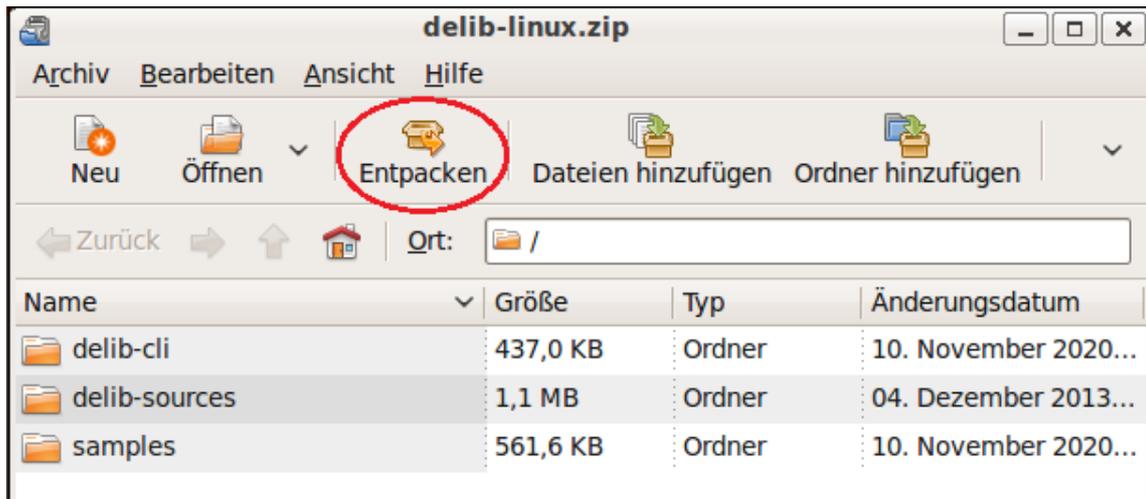


The screenshot shows a web browser window with the URL [www.deditec.de/de/downloads/produkte/](http://www.deditec.de/de/downloads/produkte/). The page title is "Produkte" and the breadcrumb is "Startseite > Downloads > Produkte". The navigation menu includes "PRODUKTE" and "ALLGEMEINES". The main content area is titled "DELIB Treiberbibliothek" and features a horizontal menu with "DELIB+Protokolle" selected. Below this, there is a list of download items:

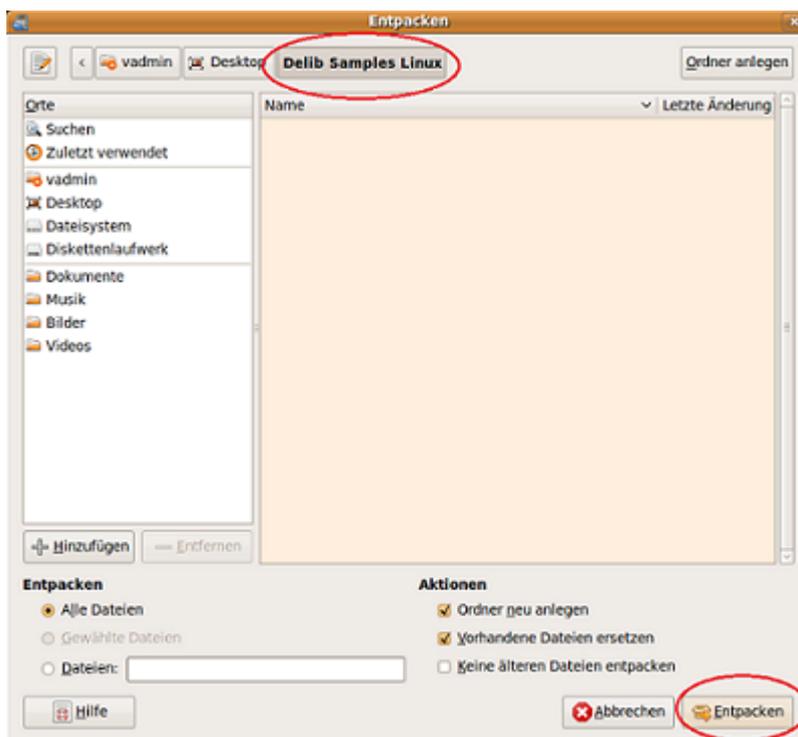
- Manual der DELIB Treiberbibliothek**  
Dokumentation aller Befehle für die Treiberbibliothek
- DELIB (32-Bit) Treiberbibliothek für Windows**  
Für Windows 10, Windows 7, Windows 8, Vista, XP und 2000
- DELIB (64-Bit) Treiberbibliothek für Windows**  
Für Windows 10, Windows 7, Windows 8, Vista, XP und 2000
- DELIB Treiberbibliothek für Linux (32/64-Bit)**  
Für 32/64-Bit Linux-Distributionen ab Kernel 2.6.x
- DELIB-Samples Installationspaket (ca. 10 MB)**  
Beispielprogramme für verschiedene Programmiersprachen (Im DELIB Setup enthalten)
- Handbuch Protokolle & Registerbelegung**  
Übertragungsprotokolle und Registerbelegung

A red arrow points to the "DOWNLOAD" button for the "DELIB Treiberbibliothek für Linux (32/64-Bit)" item, which is also circled in red.

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.



## 4.4.1. Verwenden der DELIB Treiberbibliothek für Linux

### 4.4.1.1. Delib USB-Sample in Linux

#### Voreinstellungen

In diesem Programmbeispiel wird ein **USB\_RELAIS\_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**“

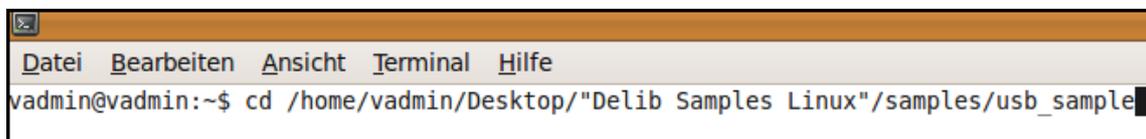
```
23
24 #include <stdio.h>
25 #include <stdlib.h>
26 #include <unistd.h>
27
28 #include ".././././delib-sources/delib/library/delib/delib.h"
29
30 int main()
31 {
32     ULONG i;
33     ULONG handle=0;
34
35     printf("\n\n");
36     printf("-----\n");
37     printf("-----\n");
38     printf("-----\n");
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_RELAIS_8\n");
49     handle = DapiOpenModule(USB_RELAIS_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 Ordnernamen Leerzeichen enthalten sein, geben Sie diesen wie im unteren Beispiel dargestellt in "" an.

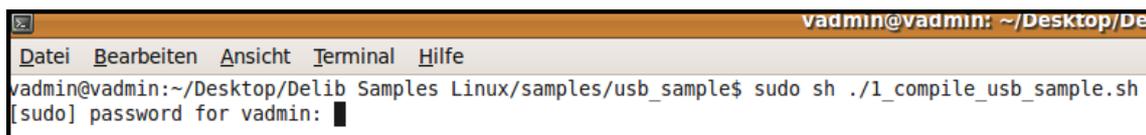


```
vadmin@vadmin:~$ cd /home/vadmin/Desktop/"Delib Samples Linux"/samples/usb_sample
```

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/De
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.



## 4.4.2. 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 ["nunit"] ]
```

### Definition für ETH-Module (Linux)

```
delib_cli [command] [channel] [value | unit ["nunit"] ]
```

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	-	-
do16		16-Bit Wert		
do32		32-Bit Wert		
		(Bit 0 für Kanal 1, Bit 1 für Kanal 2, ...)		
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

#### 4.4.2.1. 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 die 0.

##### RO-ETH\_ipAddress

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

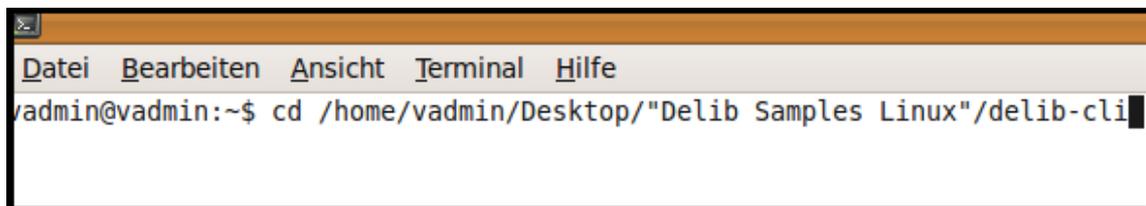
Die IP-Adresse der ETH Module können über das DELIB Configuration Utility sowie über die Weboberfläche des Moduls eingestellt werden.

## Kompilieren des Delib-CLI-Samples

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

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

**Tip:** Sollten in Ihrem Ordnernamen Leerzeichen enthalten sein, geben Sie diesen wie im unteren Beispiel dargestellt in **" "** an.



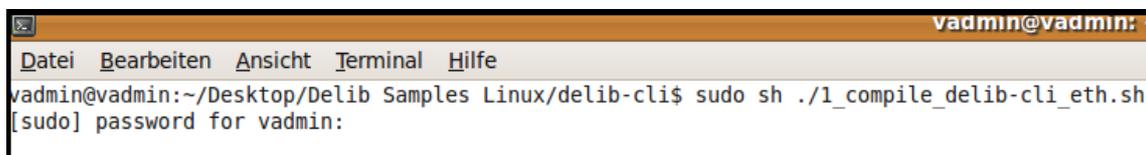
```
vadmin@vadmin:~$ cd /home/vadmin/Desktop/"Delib Samples Linux"/delib_cli
```

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.



```
vadmin@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"**

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

#### 4.4.2.2. DELIB CLI Beispiele

##### Digitale Ausgänge

`udo delib_cli DO1 17 1`

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

`udo delib_cli DO1 3 0`

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

##### Digitale Eingänge

`udo delib_cli DI1 3`

Beispiel eines Rückgabewertes: **1**

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

`udo 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

`udo 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

`udo 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

`udo 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

`udo delib_cli FF 32 hex`

Beispiel eines Rückgabewertes: **0x3**

(auf Kanälen 33 und 34 wurde eine Zustandsänderung erkannt)

-> lese den Wert der FlipFlops der digitalen Eingänge 0-64 als hexadezimalzahl

## Analoge Ausgänge

`udo delib_cli AO 7 4711`

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

`udo delib_cli AO 6 0x4711`

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

## Analoge Eingänge

`udo delib_cli AI 2`

Beispiel eines Rückgabewertes: **1234**

-> liest den Wert des 3. analogen Eingangs als Dezimalzahl eines USB-Moduls

`udo 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



## 5. DELIB API Referenz

### 5.1. Verwaltungsfunktionen

#### 5.1.1. 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;
}
```

## 5.1.2. 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);
```

### 5.1.3. 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)
```

## 5.1.4. 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
```

## 5.1.5. 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 (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);
```

## 5.2. Fehlerbehandlung

### 5.2.1. 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;
}
```

## 5.2.2. 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;
}
```

### 5.2.3. 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;
}
```

## 5.2.4. 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;
}
```

## 5.2.5. 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;
}
```

## 5.3. Digitale Eingänge lesen

### 5.3.1. 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)

### 5.3.2. 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 gelesenen Eingänge

### 5.3.3. 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

### 5.3.4. 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();
```

### 5.3.5. 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

### 5.3.6. 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

### 5.3.7. 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
```

### 5.3.8. 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);
```

### 5.3.9. DapiSpecialCounterLatchAllWithReset

#### Beschreibung

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

Zusätzlich werden die Zählerstände der Eingangszähler im Anschluß resettet.

#### 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);
```

### 5.3.10. 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
```

### 5.3.11. 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.
```

### 5.3.12. 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.
```

### 5.3.13. 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
```

## 5.4. Digitale Ausgänge verwalten

### 5.4.1. 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

## 5.4.2. 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

### 5.4.3. 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

## 5.4.4. 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();
```

### 5.4.5. 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 Ausganges an, ab dem geschrieben werden soll (0, 64, ..)

data=Gibt die Datenwerte an, die geschrieben werden

#### Return-Wert

Keiner

## 5.4.6. 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-08-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
```

### 5.4.7. 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.

## 5.4.8. 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 Ausganges an, ab dem zurückgelesen werden soll (0, 64, ..)

### Return-Wert

Zustand von 64 Ausgängen.

### 5.4.9. 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);
```

## 5.4.10. 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);
```

## 5.5. Ausgabe-Timeout verwalten

### 5.5.1. DapiSpecialCMDTimeout

#### Beschreibung

Dieser Befehl dient zum Einstellen der Timeout-Schutz-Funktion.

Es gibt seit 2021 drei unterschiedliche Timeout-Methoden.

#### "normalen" Timeout

Dies ist der Timeout, den unsere Module schon seit 2009 besitzen.

Vorgehensweise für den Timeout-Befehl:

Der Timeout wird per Befehl aktiviert.

Findet dann ein sogenanntes Timeout-Ereignis statt(Pause zwischen zwei Zugriffen auf das Modul ist grösser, als die erlaubte Timeout-Zeit) passiert folgendes:

- Alle Ausgänge werden ausgeschaltet.
- Der Timeout-Status geht auf "2".
- Die Timeout-LED geht an (bei Modulen, die solch einen Status haben)

Weitere Zugriffe auf die Ausgänge sind dann weiterhin möglich, aber der Timeout ist nicht weiter aktiv. Erst wieder, wenn er wieder aktiviert wurde.

### **"auto reactivate" Timeout**

Dies ist ein seit 2021 implementierter Timeout-Modus, der nach Auftreten des Timeout-Ereignisses den Timeout automatisch wieder aktiviert.

Vorgehensweise für den Timeout-Befehl:

Der Timeout wird per Befehl aktiviert.

Findet dann ein sogenanntes Timeout-Ereignis statt (Pause zwischen zwei Zugriffen auf das Modul ist grösser, als die erlaubte Timeout-Zeit) passiert folgendes:

- Alle Ausgänge werden ausgeschaltet.
- Der Timeout-Status geht auf "4".
- Die Timeout-LED geht an (bei Modulen, die solch einen Status haben)

Weitere Zugriffe auf die Ausgänge sind dann weiterhin möglich. UND der Timeout ist weiter aktiv. Bei erneuter Zeitüberschreitung der Timeout-Zeit werden die Ausgänge wieder ausgeschaltet.

## **"secure outputs" Timeout**

Dies ist ein seit 2021 implementierter Timeout-Modus, der nach Auftreten des Timeout-Ereignisses einen Schreibenden Zugriff auf die Ausgänge verhindert. Somit wird sichergestellt, dass die Software erst einmal einen "sicheren" Zustand der Ausgänge wiederherstellen muss, da der Timeout-Mechanismus des Moduls die Ausgänge auf vordefinierte Werte verändert hat.

Vorgehensweise für den Timeout-Befehl:

Der Timeout wird per Befehl aktiviert.

Findet dann ein sogenanntes Timeout-Ereignis statt (Pause zwischen zwei Zugriffen auf das Modul ist grösser, als die erlaubte Timeout-Zeit) passiert folgendes:

- Alle Ausgänge werden ausgeschaltet.
- Der Timeout-Status geht auf "6".
- Die Timeout-LED geht an (bei Modulen, die solch einen Status haben)

Weitere Zugriffe auf die Ausgänge sind NICHT möglich. Erst nach erneutem Aktivieren des Timeouts oder Deaktivieren des Timeouts können die Ausgänge beschrieben werden.

## **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

### 5.5.1.1. 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.
```

### 5.5.1.2. DapiSpecialTimeoutActivate

#### Beschreibung

Dieser Befehl aktiviert den "normalen" Timeout.

Nach dem Timeout-Ereignis werden..

- ..alle Ausgänge ausgeschaltet
- ..der Timeout-Status auf "2" gesetzt
- ..die Timeout-LED angeschaltet (bei Modulen, die solch einen Status haben)

Weitere Zugriffe auf die Ausgänge sind dann weiterhin möglich, aber der Timeout ist nicht weiter aktiv.

Erst wieder, wenn er wieder aktiviert wurde.

#### 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.
```

### 5.5.1.3. DapiSpecialTimeoutActivateAutoReactivate

#### Beschreibung

Dieser Befehl aktiviert den "auto reactivate" Timeout.

In diesem Modus wird der Timeout nach dem Timeout-Ereignis automatisch wieder aktiviert.

Nach dem Timeout-Ereignis werden..

- ..alle Ausgänge ausgeschaltet
- ..der Timeout-Status auf "4" gesetzt
- ..die Timeout-LED angeschaltet (bei Modulen, die solch einen Status haben)

Weitere Zugriffe auf die Ausgänge sind dann weiterhin möglich UND der Timeout ist weiter aktiv.

Bei erneuter Zeitüberschreitung der Timeout-Zeit werden die Ausgänge wieder ausgeschaltet.

#### Definition

*DapiSpecialCommand(handle, DAPI\_SPECIAL\_CMD\_TIMEOUT, cmd, 0, 0);*

#### Parameter

cmd = DAPI\_SPECIAL\_TIMEOUT\_ACTIVATE\_AUTO\_REACTIVATE

#### Programmierbeispiel

```
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_TIMEOUT,  
DAPI_SPECIAL_TIMEOUT_ACTIVATE_AUTO_REACTIVATE, 0, 0);  
//Der "auto reactivate" Timeout wird aktiviert.
```

#### 5.5.1.4. DapiSpecialTimeoutActivateSecureOutputs

##### Beschreibung

Dieser Befehl aktiviert den "secure" Timeout.

In diesem Modus wird ein schreibender Zugriff auf die Ausgänge nach einem Timeout-Ereignis verhindert.

Somit wird sichergestellt, dass die Software erst einmal einen "sicheren" Zustand der Ausgänge wiederherstellen muss,

da der Timeout-Mechanismus des Moduls die Ausgänge auf vordefinierte Werte verändert hat.

Nach dem Timeout-Ereignis werden..

- ..alle Ausgänge ausgeschaltet
- ..der Timeout-Status auf "6" gesetzt
- ..die Timeout-LED angeschaltet (bei Modulen, die solch einen Status haben)

Weitere Zugriffe auf die Ausgänge sind NICHT möglich. Erst nach erneutem Aktivieren des

Timeouts oder Deaktivieren des Timeouts können die Ausgänge beschrieben werden.

##### Definition

*DapiSpecialCommand(handle, DAPI\_SPECIAL\_CMD\_TIMEOUT, cmd, 0, 0);*

##### Parameter

cmd = DAPI\_SPECIAL\_TIMEOUT\_ACTIVATE\_SECURE\_OUTPUTS

##### Programmierbeispiel

```
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_TIMEOUT,  
DAPI_SPECIAL_TIMEOUT_ACTIVATE_SECURE_OUTPUTS, 0, 0);  
//Der "secure" Timeout wird aktiviert.
```

### 5.5.1.5. 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.
```

### 5.5.1.6. 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)

Werte für den "auto reactivate" Timeout

Return = 3 (Timeout "auto reactivate" ist aktiviert)

Return = 4 (Timeout "auto reactivate" hat ein oder mehrmals stattgefunden)

Werte für den "secure" Timeout

Return = 5 (Timeout "secure" ist aktiviert)

Return = 6 (Timeout "secure" hat stattgefunden. In diesem Status wird ein Schreiben auf die Outputs verhindert)

## 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.
```

### 5.5.1.7. DapiSpecialTimeoutDoValueMaskWRSet32

#### Beschreibung

Dieser Befehl bestimmt die Ausgänge, die bei einem Timeout gesetzt werden sollen.

#### Definition

*DapiSpecialCommand(handle, DAPI\_SPECIAL\_CMD\_TIMEOUT, cmd, ch, par2);*

#### Parameter

cmd = DAPI\_SPECIAL\_TIMEOUT\_DO\_VALUE\_MASK\_WR\_SET32

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

par2 = [32 Bit] Gibt die Ausgänge an, welche bei einem Timeout aktiviert werden sollen

#### Programmierbeispiel

```
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_TIMEOUT,  
DAPI_SPECIAL_TIMEOUT_DO_VALUE_MASK_WR_SET32, 0, 0xff);  
//Die ersten 8 Relais werden im Timeout Fall eingeschaltet.
```

### 5.5.1.8. DapiSpecialTimeoutDoValueMaskRDSet32

#### Beschreibung

Dieser Befehl dient zum Auslesen der übergebenen Werte.

#### Definition

*ULONG DapiSpecialCommand(handle, DAPI\_SPECIAL\_CMD\_TIMEOUT, cmd, 0, 0);*

#### Parameter

cmd = DAPI\_SPECIAL\_TIMEOUT\_DO\_VALUE\_MASK\_RD\_SET32

#### Return-Wert

[32 Bit] Wert der dem SET-Befehl übergeben wird

#### Programmierbeispiel

```
long value = DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_TIMEOUT,  
DAPI_SPECIAL_TIMEOUT_DO_VALUE_MASK_RD_SET32, 0, 0);  
printf("%0x\n", value);  
//Der Wert der dem SET-Befehl übergeben wurde, wird ausgelesen und dargestellt.
```

### 5.5.1.9. DapiSpecialTimeoutDoValueMaskWRClr32

#### Beschreibung

Dieser Befehl bestimmt die Ausgänge, die bei einem Timeout ausgeschaltet werden sollen.

#### Definition

*DapiSpecialCommand(handle, DAPI\_SPECIAL\_CMD\_TIMEOUT, cmd, ch, par2);*

#### Parameter

cmd = DAPI\_SPECIAL\_TIMEOUT\_DO\_VALUE\_MASK\_WR\_CLR32

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

par2 = [32 Bit] Gibt die Ausgänge an, welche bei einem Timeout deaktiviert werden sollen

#### Programmierbeispiel

```
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_TIMEOUT,  
DAPI_SPECIAL_TIMEOUT_DO_VALUE_MASK_WR_CLR32, 0, 0xff);  
//Die ersten 8 Relais werden im Timeout Fall ausgeschaltet.
```

### 5.5.1.10. DapiSpecialTimeoutDoValueMaskRDClr32

#### Beschreibung

Dieser Befehl dient zum Auslesen der übergebenen Werte.

#### Definition

*ULONG DapiSpecialCommand(handle, DAPI\_SPECIAL\_CMD\_TIMEOUT, cmd, 0, 0);*

#### Parameter

cmd = DAPI\_SPECIAL\_TIMEOUT\_DO\_VALUE\_MASK\_RD\_CLR32

#### Return-Wert

[32 Bit] Wert der dem CLR-Befehl übergeben wird

#### Programmierbeispiel

```
long value = DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_TIMEOUT,  
DAPI_SPECIAL_TIMEOUT_DO_VALUE_MASK_RD_CLR32, 0, 0);  
printf("%0x\n", value);  
//Der Wert der dem CLR-Befehl übergeben wurde, wird ausgelesen und dargestellt.
```

### 5.5.1.11. 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.
```

## 5.6. Testfunktionen

### 5.6.1. 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

## 5.7. Register Schreib-Befehle

### 5.7.1. 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.

## 5.7.2. 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.

### 5.7.3. 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.

## 5.7.4. 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.

## 5.8. Register Lese-Befehle

### 5.8.1. 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.

## 5.8.2. 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.

### 5.8.3. 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.

## 5.8.4. 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.

## 5.9. 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_S EC DAPI_SPECIAL_TIMEOUT_ACTIVATE DAPI_SPECIAL_TIMEOUT_DEACTIVATE DAPI_SPECIAL_TIMEOUT_GET_STATUS	alle Module	

Befehl	Starter USB*1	Starter ETH*2	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			

\*1: USB-OPTOIN8, USB-Mini-Stick, USB-TTL-64

\*2: ETH-TTL64, ETH-OPTOIN8, ETH-RELAIS8

Befehl	Starter USB*1	Starter ETH*2	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		

\*1: USB-OPTOIN8, USB-Mini-Stick, USB-TTL-64

\*2: ETH-TTL64, ETH-OPTOIN8, ETH-RELAIS8

# Anhang



## **6. Anhang**

### **6.1. Kontakt / Support**

Sollten Sie Fragen zum Produkt haben oder Unterstützung bei der Inbetriebnahme benötigen, so 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

### **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 3.01	DEDITEC Design Update 2022
Rev 3.00	DEDITEC Design Update 2021
Rev 2.05	Kapitel "Software" und "DELIB API Referenz" überarbeitet

Rev 2.04	Kapitel "DELIB Treiberbibliothek" überarbeitet.
Rev 2.03	Index hinzugefügt
Rev 2.02	Kapitel "Firmware Update" hinzugefügt, Kapitel "Software" überarbeitet
Rev 2.01	Manual um USB-RELAIS-8_A & USB-OPTOIN-8_A erweitert
Rev 2.00	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.