



# USB-WATCHDOG-STICK

Hardware-Beschreibung

2023 Februar

# 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        |
| <b><u>2. Hardware Beschreibung</u></b>                 | <b>11</b> |
| 2.1. Übersichtsbild                                    | 12        |
| 2.2. Technische Daten                                  | 13        |
| 2.3. Pinbelegung DSUB-9 Buchse                         | 14        |
| 2.4. Kontroll LED                                      | 14        |
| 2.5. Anwendungsbeispiele                               | 15        |
| 2.5.1. Resetten eines PCs                              | 15        |
| 2.5.1.1. Anschluss über den Reset-Taster               | 16        |
| 2.5.1.2. Anschluss über Netzteil                       | 18        |
| 2.5.1.2.1. Anschluss über ein ATX-/BTX-Adapterkabel    | 18        |
| 2.5.1.2.2. Anschluss direkt an das Netzteil            | 20        |
| <b><u>3. Software Beschreibung</u></b>                 | <b>21</b> |
| 3.1. Benutzen unserer Produkte                         | 22        |
| 3.1.1. Ansteuerung über unsere DELIB Treiberbibliothek | 22        |
| 3.1.2. Ansteuerung über mitgelieferte Testprogramme    | 22        |
| 3.1.3. Ansteuerung auf Protokollebene                  | 23        |
| 3.1.4. DELIB CLI (command-line interface) für Windows  | 24        |
| 3.1.4.1. Konfiguration des DELIB CLI                   | 26        |
| 3.1.4.2. DELIB CLI Beispiele                           | 27        |
| 3.1.5. Ansteuerung über grafische Anwendungen          | 31        |
| 3.1.5.1. LabVIEW                                       | 31        |
| 3.1.5.2. ProfiLab                                      | 31        |
| 3.1.5.3. Licht24 Pro                                   | 32        |
| 3.1.6. Einbinden der DELIB in Programmiersprachen      | 33        |

# INDEX

|   |           |
|---|-----------|
| 3.1.6.1. Einbinden der DELIB in Visual-C/C++                      | 33        |
| 3.1.6.2. Einbinden der DELIB in Visual-C/C++ (Visual Studio 2015) | 35        |
| 3.1.6.3. Einbinden der DELIB in Visual-C#                         | 38        |
| 3.1.6.4. Einbinden der DELIB in Delphi                            | 39        |
| 3.1.6.5. Einbinden der DELIB in Visual-Basic (VB)                 | 40        |
| 3.1.6.6. Einbinden der DELIB in Visual-Basic.NET (VB.NET)         | 41        |
| 3.1.6.7. Einbinden der DELIB in MS-Office (VBA)                   | 42        |
| 3.1.6.8. Einbinden der DELIB in LabVIEW                           | 44        |
| 3.1.6.8.1. Einbinden der DELIB in LabVIEW                         | 44        |
| 3.1.6.8.2. Verwendung der VIs in LabVIEW                          | 53        |
| 3.1.6.8.3. Setzen der Modul-ID in LabVIEW                         | 55        |
| 3.1.6.9. Einbinden der DELIB in Java                              | 57        |
| <b>3.2. DELIB Treiberbibliothek</b>                               | <b>58</b> |
| 3.2.1. Übersicht  | 59        |
| 3.2.1.1. Unterstützte Programmiersprachen                         | 62        |
| 3.2.1.2. Unterstützte Betriebssysteme                             | 63        |
| 3.2.1.3. SDK-Kit für Programmierer                                | 63        |
| 3.2.2. DELIB Setup  | 64        |
| 3.2.3. DELIB Configuration Utility                                | 70        |
| 3.2.3.1. Einführung   | 70        |
| 3.2.3.2. Neue Konfig erstellen oder vorhandene Konfig bearbeiten  | 71        |
| 3.2.3.2.1. Modul Konfiguration USB                                | 72        |
| 3.2.3.2.1.1, Beispiel zur Konfiguration identischer USB-Module    | 74        |
| 3.2.3.2.2. Modul Konfiguration Ethernet                           | 79        |
| 3.2.3.2.2.1, Automatische Suche                                   | 84        |
| 3.2.3.2.2.2, Verschlüsselung einrichten                           | 88        |
| 3.2.3.2.2.1, Manuelle Konfiguration                               | 88        |
| 3.2.3.2.2.2, Automatische Konfiguration                           | 93        |
| 3.2.3.2.2.1, Authentifizierung                                    | 94        |
| 3.2.3.2.3. Modul Konfiguration CAN                                | 98        |
| 3.2.3.2.4. Modul Konfiguration Seriell                            | 101       |

# INDEX

|   |     |
|---|-----|
| 3.2.3.3. Modul testen                                   | 103 |
| 3.2.3.4. Debug Optionen einstellen                      | 107 |
| 3.2.4. Benutzung des Modulselectors                     | 108 |
| 3.2.4.1. via USB  | 108 |
| 3.2.4.2. via Ethernet                                   | 109 |
| 3.2.4.3. via WiFi / WPS                                 | 111 |
| 3.2.4.4. Modul Info                                     | 112 |
| 3.2.5. DELIB Module Config                              | 114 |
| 3.2.5.1. Modul Konfigurationen                          | 114 |
| 3.2.5.1.1. Modul-Infoseite                              | 115 |
| 3.2.5.1.2. Modul-Identifikation                         | 116 |
| 3.2.5.1.3. LAN Netzwerkinformationen                    | 117 |
| 3.2.5.1.4. LAN Netzwerkeinstellungen                    | 118 |
| 3.2.5.1.5. WiFi Netzwerkinformationen                   | 120 |
| 3.2.5.1.6. WiFi Netzwerkeinstellungen                   | 122 |
| 3.2.5.1.7. WiFi WPS-Verbindung                          | 124 |
| 3.2.5.1.8. NTP-Konfiguration                            | 125 |
| 3.2.5.1.9. Serielle Konfiguration                       | 127 |
| 3.2.5.1.10. I/O Kanal-Namen                             | 129 |
| 3.2.6. DELIB Module Demo                                | 130 |
| 3.2.6.1. Auswahl des Moduls                             | 131 |
| 3.2.6.2. Allgemein                                      | 132 |
| 3.2.6.2.1. Module Info                                  | 134 |
| 3.2.6.3. Digital Input                                  | 135 |
| 3.2.6.4. Digital Output                                 | 136 |
| 3.2.6.5. Analog Input                                   | 137 |
| 3.2.6.6. Analog Output                                  | 138 |
| 3.2.7. CAN Configuration Utility                        | 139 |
| 3.2.7.1. Auswahl des Moduls                             | 141 |
| 3.2.7.2. Neue Konfiguration Erstellen, Laden, Speichern | 143 |
| 3.2.7.3. Konfiguration auf das Modul übertragen         | 145 |
| 3.2.7.4. Statistiken vom Modul abfragen                 | 146 |
| 3.2.7.5. Konfiguration                                  | 149 |



# INDEX

|  |     |
|--|-----|
| 3.2.7.5.1. Modul Konfiguration                                 | 150 |
| 3.2.7.5.2. I/O Konfiguration                                   | 151 |
| 3.2.7.5.3. TX-Konfiguration                                    | 154 |
| 3.2.7.5.3.1, Beispiel Interval                                 | 156 |
| 3.2.7.5.3.2, Beispiel Trigger                                  | 157 |
| 3.2.7.5.4. RX-Konfiguration                                    | 158 |
| 3.2.7.5.4.1, Beispiel RX-DA                                    | 160 |
| 3.2.7.5.4.2, Beispiel RX-DO                                    | 161 |
| 3.2.7.6. Aufbau der CAN-Pakete                                 | 162 |
| 3.2.7.6.1. Digitale Eingänge                                   | 162 |
| 3.2.7.6.2. Digitale Ausgänge                                   | 163 |
| 3.2.7.6.3. Digitale Eingangszähler (16-Bit)                    | 164 |
| 3.2.7.6.4. Digitale Eingangszähler (48-Bit) - 32-Bit Paket     | 165 |
| 3.2.7.6.5. Digitale Eingangszähler (48-Bit) - 64-Bit Paket     | 166 |
| 3.2.7.6.6. Analoge Ein- / Ausgänge                             | 167 |
| 3.2.7.6.6.1, Analoge Eingänge                                  | 167 |
| 3.2.7.6.6.2, Analoge Ausgänge                                  | 169 |
| 3.2.7.6.6.3, Beispiele   | 170 |
| 3.2.7.6.7. Temperatur Eingänge                                 | 173 |
| 3.2.7.6.8. Stepper   | 175 |
| 3.2.7.6.8.1, Command-Liste                                     | 175 |
| 3.2.7.6.8.2, Werte für par 1 zu Befehl SET_MOTORCHARACTERISTIC | 178 |
| 3.2.7.6.8.3, Werte für par 1 zu Befehl GO_REFSWITCH            | 181 |
| 3.2.7.6.8.4, Beispiel  | 181 |
| 3.2.8. DT-Flasher  | 182 |
| 3.2.8.1. Über DEDITEC-Firmware                                 | 183 |
| 3.2.8.2. Auswahl des Moduls                                    | 183 |
| 3.2.8.3. Firmware Update durchführen                           | 184 |
| 3.2.8.3.1. Flash-Files manuell aktualisieren                   | 186 |
| 3.3. DELIB Sample Sources (Windows Programmbeispiele)          | 187 |
| 3.3.1. Installation DELIB Sample Sources                       | 188 |
| 3.3.2. Benutzung der DELIB Sample Sources                      | 192 |
| 3.3.2.1. Schritt 1 - Produktauswahl                            | 193 |

# INDEX

|   |            |
|---|------------|
| 3.3.2.2. Schritt 2 - Kategorieauswahl                     | 194        |
| 3.3.2.3. Schritt 3 - Programmiersprachenauswahl           | 195        |
| 3.3.2.4. Schritt 4 - Quellcode                            | 196        |
| <b>3.4. DELIB für Linux</b>                               | <b>199</b> |
| 3.4.1. DELIB Sample Sources (Windows Programmbeispiele) 2 | 202        |
| 3.4.1.1. Installation DELIB Sample Sources 2              | 203        |
| 3.4.1.2. Benutzung der DELIB Sample Sources 2             | 207        |
| 3.4.1.2.1. Schritt 1 - Produktauswahl 2                   | 208        |
| 3.4.1.2.2. Schritt 2 - Kategorieauswahl 2                 | 209        |
| 3.4.1.2.3. Schritt 3 - Programmiersprachenauswahl 2       | 210        |
| 3.4.1.2.4. Schritt 4 - Quellcode 2                        | 211        |
| 3.4.2. Verwenden der DELIB Treiberbibliothek für Linux    | 214        |
| 3.4.2.1. Delib USB-Sample in Linux                        | 214        |
| 3.4.2.2. Delib ETH-Sample in Linux                        | 217        |
| 3.4.3. DELIB CLI (command-line interface) für Linux       | 221        |
| 3.4.3.1. Konfiguration des DELIB CLI                      | 224        |
| 3.4.3.2. DELIB CLI Beispiele                              | 227        |
| <b>4. DELIB API Referenz</b>                              | <b>229</b> |
| 4.1. Verfügbare DEDITEC Modul IDs                         | 230        |
| 4.2. Verwaltungsfunktionen                                | 233        |
| 4.2.1. DapiOpenModule                                     | 233        |
| 4.2.2. DapiCloseModule                                    | 234        |
| 4.2.3. DapiGetDELIBVersion                                | 234        |
| 4.2.4. DapiSpecialCMDGetModuleConfig                      | 235        |
| 4.2.5. DapiOpenModuleEx                                   | 238        |
| 4.2.6. DapiScanAllModulesAvailable                        | 240        |
| 4.3. Fehlerbehandlung                                     | 241        |
| 4.3.1. DapiGetLastError                                   | 241        |
| 4.3.2. DapiGetLastErrorText                               | 242        |
| 4.3.3. DapiClearLastError                                 | 243        |
| 4.3.4. DapiGetLastErrorByHandle                           | 244        |

# INDEX

|                                   |            |
|-----------------------------------|------------|
| 4.3.5. DapiClearLastErrorByHandle | 245        |
| 4.4. Watchdog Funktionen          | 246        |
| 4.4.1. DapiWatchdogEnable         | 246        |
| 4.4.2. DapiWatchdogDisable        | 247        |
| 4.4.3. DapiWatchdogRetrigger      | 248        |
| 4.5. Testfunktionen               | 249        |
| 4.5.1. DapiPing                   | 249        |
| 4.6. Register Schreib-Befehle     | 250        |
| 4.6.1. DapiWriteByte              | 250        |
| 4.6.2. DapiWriteWord              | 251        |
| 4.6.3. DapiWriteLong              | 252        |
| 4.6.4. DapiWriteLongLong          | 253        |
| 4.7. Register Lese-Befehle        | 254        |
| 4.7.1. DapiReadByte               | 254        |
| 4.7.2. DapiReadWord               | 255        |
| 4.7.3. DapiReadLong               | 256        |
| 4.7.4. DapiReadLongLong           | 257        |
| 4.8. Programmier-Beispiel         | 258        |
| 4.9. Delib Übersichtstabelle      | 260        |
| <b><u>5. Anhang</u></b>           | <b>263</b> |
| 5.1. Kontakt / Support            | 264        |
| 5.2. Umwelt und Entsorgung        | 264        |
| 5.3. Revisionen                   | 265        |
| 5.4. Urheberrechte und Marken     | 266        |

# Einleitung

---



# **1. Einleitung**

## **1.1. Vorwort**

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

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

**Wir entwickeln modular!**

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

**Wir sorgen für eine lange Lieferverfügbarkeit!**

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

## **1.2. Kundenzufriedenheit**

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

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

Wir kümmern uns darum!

## **1.3. Kundenresonanz**

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

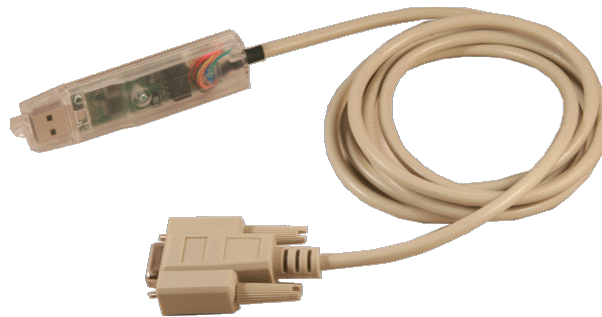
## 1.4. Kurzbeschreibung

Die in einem USB-Stick Gehäuse untergebrachte Elektronik sorgt für eine Überwachung mit Watchdogfunktion. Der eingesetzte Microcontroller wird in regelmäßigen Intervallen vom PC über den USB-BUS abgefragt.

Bei einer Zeitüberschreitung schaltet der Watchdog-Stick dann selbständig die beiden Relais mit einem vom Benutzer vorzugebenem Ablauf.

So können durch die Relais, mit entsprechender Verkabelung, bei einem Timeout z.B. der PC-Reset betätigt werden, ein externes SMS-Modem kann Warnungen versenden oder eine angeschlossene Hupe signalisiert einen Alarm.

Ein mitgeliefertes Testprogramm und ein Konfigurationstool erleichtern den schnellen Einstieg!

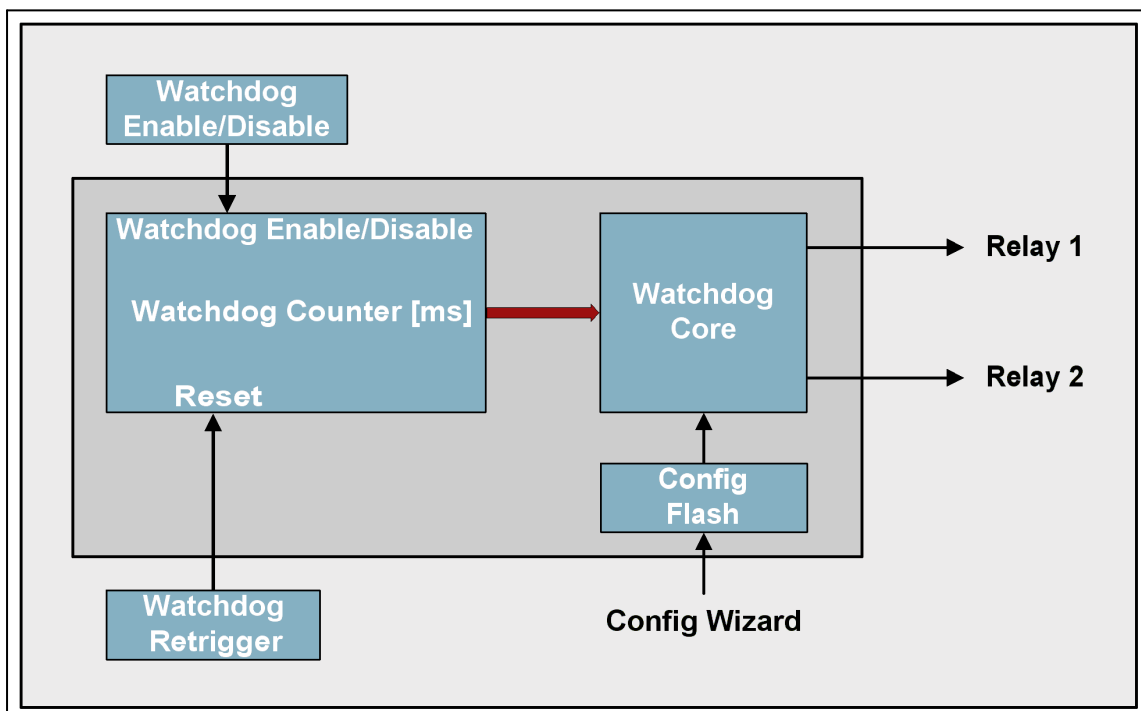
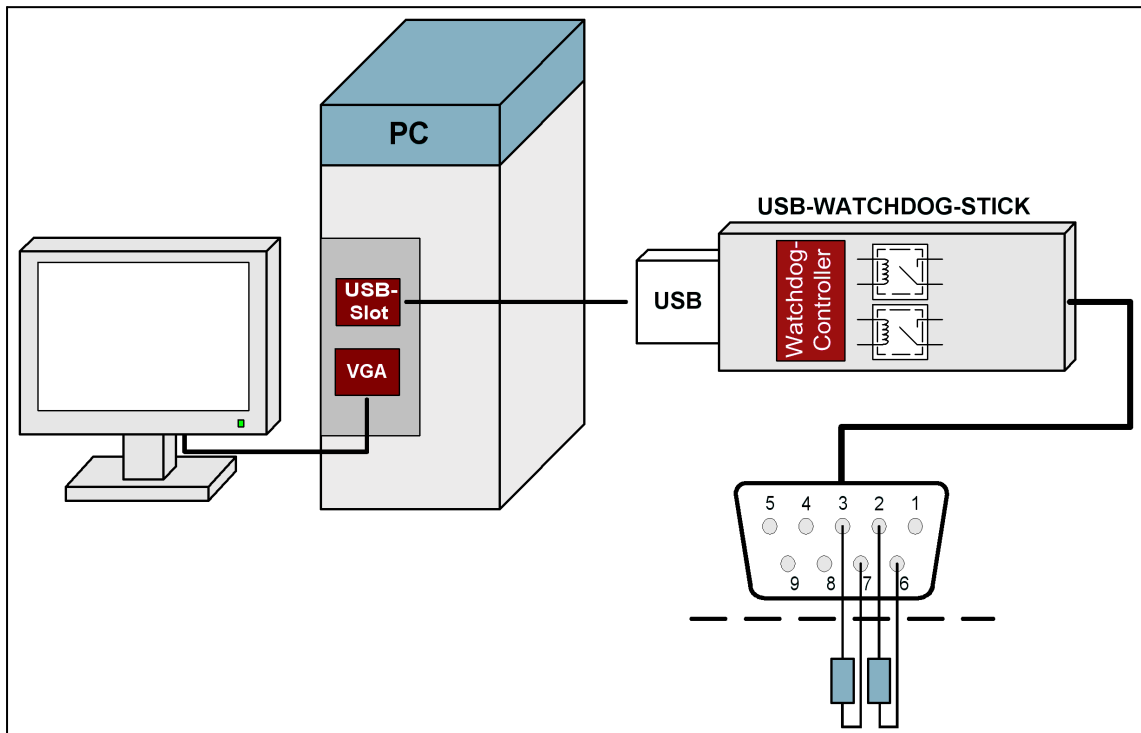


# **Hardware Beschreibung**



## 2. Hardware Beschreibung

### 2.1. Übersichtsbild

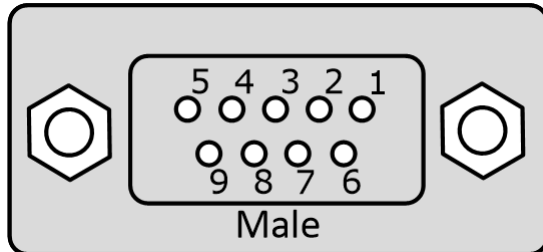




## 2.2. Technische Daten

- USB-Stick-Ausgabemodul mit USB 2.0 / USB 1.1 Interface
- Microcontroller-Überwachung
- 10ms bis 10h Timeoutzeiten einstellbar
- Windows Watchdog API
- 2 Relais für Schaltvorgänge
- Anschlußkabel (ca 1,8 m) mit DSUB-9 Buchse
- Abmessungen: 84,5 \* 21 \* 12,5 / 9,5 mm (ohne Kabel)

### 2.3. Pinbelegung DSUB-9 Buchse



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

### 2.4. Kontroll LED

Die LED auf dem Watchdog-Stick zeigt folgenden Status an:

| Beschreibung       | LED Blink Sequenz                                 |
|--------------------|---|
| Watchdog Disabled  | LED dauerhaft an (geht bei PC-Zugriffen kurz aus) |
| Watchdog Enabled   | 1*an + lange Pause                                |
| Watchdog Retrigger | 2*an + lange Pause                                |
| Watchdog Timeout   | 3*kurz an + lange Pause                           |

## 2.5. Anwendungsbeispiele

### 2.5.1. Resetten eines PCs

Um einen Reset des PCs mit dem Watchdog-Stick auszulösen, gibt es mehrere Anschlussmöglichkeiten:

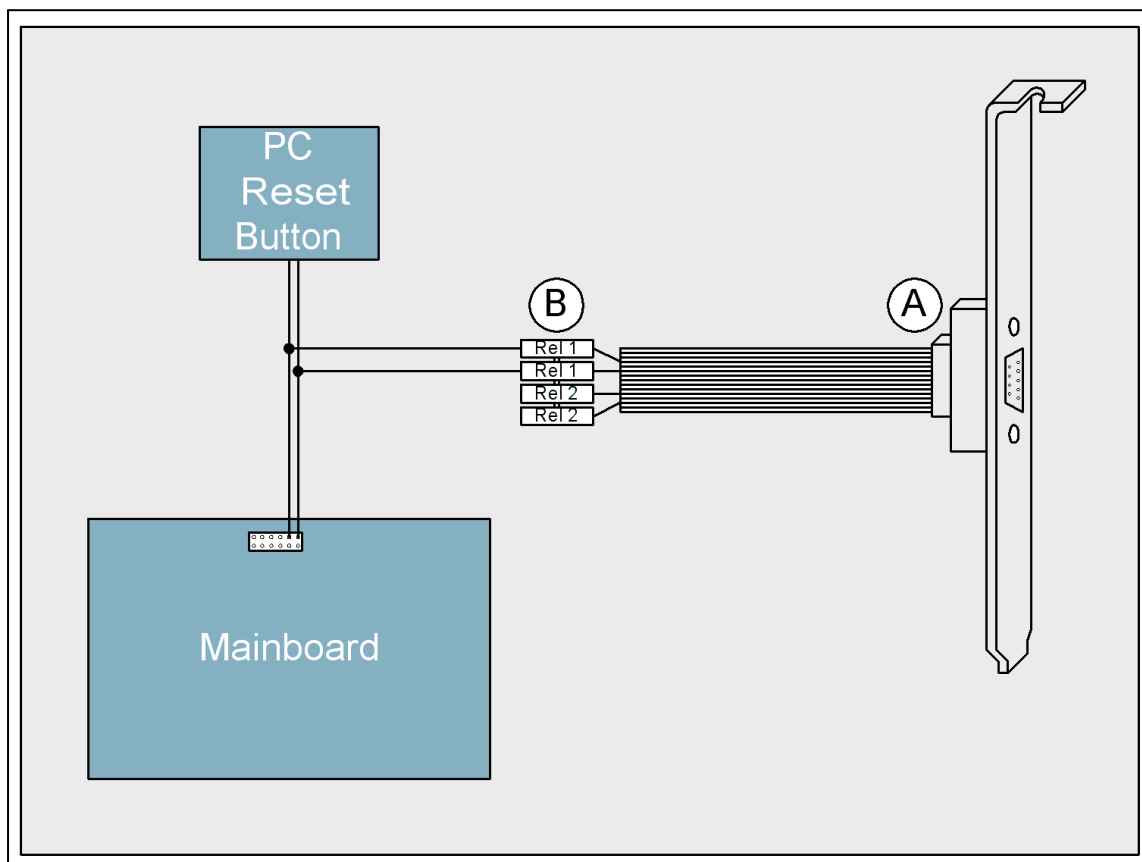
- Anschluss über den Reset-Taster

Falls der PC nicht über einen Reset-Taster verfügt:

- Anschluss über ein ATX-/BTX-Adapterkabel (optionales Zubehör)
- Anschluss direkt an das Netzteil

### 2.5.1.1. Anschluss über den Reset-Taster

Falls der PC über einen Reset-Taster verfügt, kann dieser direkt mit dem Watchdog-Stick verbunden werden. Im Falle einer Zeitüberschreitung kann der Reset-Taster vom Watchdog-Stick betätigt werden. Dazu muss das Reset-Taster-Kabel zwischen Reset-Taster und Mainboard mit dem 1. Relais (Schließer-Relais) des Watchdog-Sticks verbunden werden.



### **Vorgehensweise:**

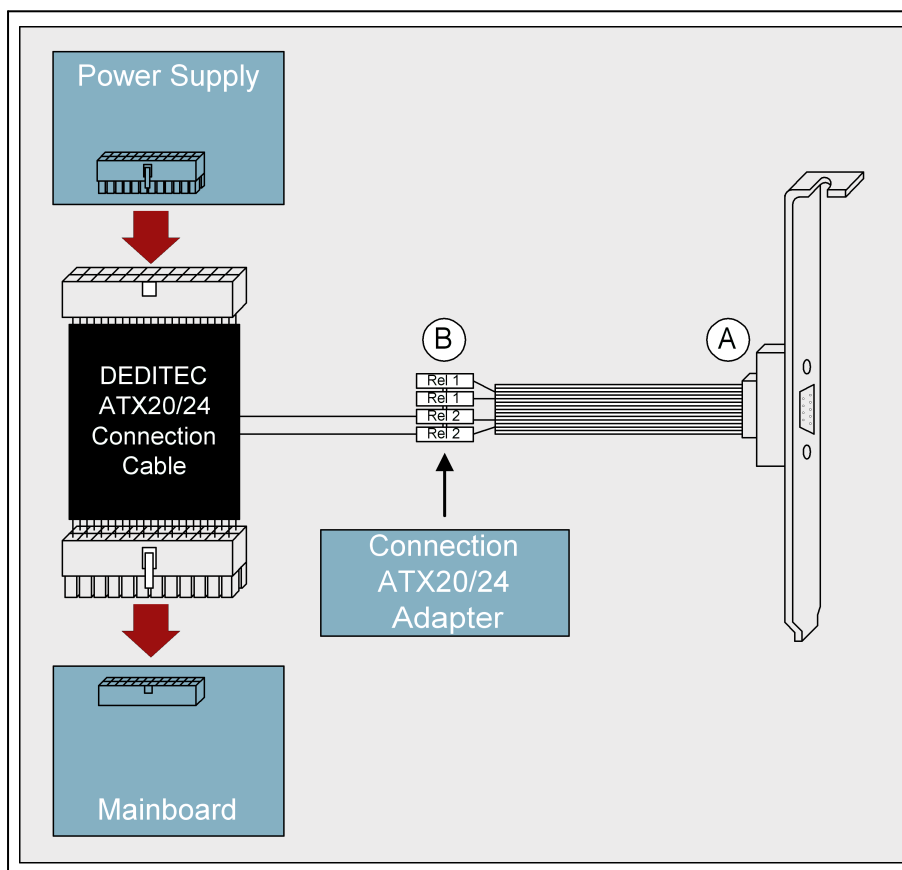
1. PC ausschalten und durch Ziehen des Netzkabels vom Stromnetz trennen.
2. D-SUB-Winkelblech (im Lieferumfang enthalten) in den PC einbauen.
3. Watchdog-Stick (USB-Seite) in eine freie USB-Buchse einstecken.
4. Watchdog-Stick (D-SUB-Seite) mit dem D-SUB-Winkelblech (A) verbinden.
5. Reset-Taster-Kabel zwischen Mainboard und Reset-Taster durchtrennen und abisolieren. Die entstehenden beiden Kabelenden mit der Lüsterklemme (1.Relais) des D-SUB-Winkelbleches (B) verbinden. Beachten Sie den korrekten Anschluss, bzw. die Nummerierung der Relais, da im Auslieferungszustand Relais 1 ein Schließer-Relais und Relais 2 ein Öffner-Relais ist.

### 2.5.1.2. Anschluss über Netzteil

Falls kein Reset-Taster vorhanden ist, muss der PC über das Netzteil resettet werden. Hier gilt es die Bauform des Rechners zu beachten.

#### 2.5.1.2.1. Anschluss über ein ATX-/BTX-Adapterkabel

Falls der PC über keinen Reset-Taster verfügt, muss der Reset über das Mainboard bzw. das Netzteil des PCs ausgelöst werden. Hierzu muss der Netzteil Stecker aus dem Mainboard gezogen und in die Buchsen-Seite des "DEDITEC ATX20/24 Connection Cable" (optionales Zubehör für den Watchdog-Stick) gesteckt werden. Danach, den Stecker des "DEDITEC ATX20/24 Connection Cable" in die entsprechende Buchse des Mainboards wieder reinstecken. Die beiden seitlich herausstehenden Kabel des "DEDITEC ATX20/24 Connection Cable" mit dem 2. Relais des D-SUB-Winkelbleches (B) verbinden.



### **Vorgehensweise:**

1. PC ausschalten und durch Ziehen des Netzkabels vom Stromnetz trennen.
2. D-SUB-Winkelblech (im Lieferumfang enthalten) in den PC einbauen.
3. Watchdog-Stick (USB-Seite) in eine freie USB-Buchse einstecken.
4. Watchdog-Stick (D-SUB-Seite) mit dem D-SUB-Winkelblech (A) verbinden.
5. Netzteil Stecker vom Mainboard entfernen (rechteckiger 20/24 pol. Anschluss). Netzteil Stecker in die Netzteil Buchse des "DEDITEC ATX20/24 Connection Cable" (optionales Zubehör) stecken. Stecker des "DEDITEC ATX20/24 Connection Cable" mit Mainboard verbinden. Die beiden nach aussen gehenden Kabel des "DEDITEC ATX20/24 Connection Cable" mit der Lüsterklemme (2.Relais) des D-SUB-Winkelbleches (B) verbinden. Beachten Sie den korrekten Anschluss, bzw. die Nummerierung der Relais, da im Auslieferungszustand Relais 1 ein Schließer-Relais und Relais 2 ein Öffner-Relais ist.

#### **2.5.1.2.2. Anschluss direkt an das Netzteil**

Bei manchen PC-Systemen, wie z.B. Dell, weicht die Pinbelegung, sowie die Bauform des Netzteil Steckers von der ATX-/BTX- Form ab. Bei diesen Systemen muss selbst das "PS\_ON#" Kabel durchtrennt und mit dem 2. Relais (Öffner-Relais) des Watchdog Sticks verbunden werden.

#### **Vorgehensweise:**

1. PC ausschalten und durch Ziehen des Netzkabels vom Stromnetz trennen.
2. D-SUB-Winkelblech (im Lieferumfang enthalten) in den PC einbauen.
3. Watchdog-Stick (USB-Seite) in eine freie USB-Buchse einstecken.
4. Watchdog-Stick (D-SUB-Seite) mit dem D-SUB-Winkelblech (A) verbinden.
5. Netzteil Stecker vom Mainboard entfernen (rechteckiger 20/24 pol. Anschluss). Das "PS\_ON#" Kabel des Netzteil Kabelstranges (bzgl. Pinbelegung - > siehe Netzteil Manual) durchtrennen und abisolieren. Die entstehenden beiden Kabelenden dann mit der Lüsterklemme (2.Relais) des D-SUB-Winkelbleches (B) verbinden. Beachten Sie den korrekten Anschluss, bzw. die Nummerierung der Relais, da im Auslieferungszustand Relais 1 ein Schließer-Relais und Relais 2 ein Öffner-Relais ist.



# **Software Beschreibung**



## **3. Software Beschreibung**

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

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

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

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

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

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

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

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

### 3.1.3. Ansteuerung auf Protokollebene

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

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

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

**Download PDF:**

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

**Online HTML-Manual:**

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

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

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

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

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

#### Definition (Windows)

*delib\_cli command channel [value / unit ["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         | (Bit 0 für Kanal 1, Bit 1 für Kanal 2, ...) |      |        |
| do16   |               | 16-Bit Wert        |   |      |        |
| do32   |               | 32-Bit Wert        |   |      |        |

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

### Return-Wert

#### Zustand der gelesenen digitalen Eingänge

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

#### Zustand der FlipFlips der digitalen Eingänge

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

#### Zustand der gelesenen analogen Eingänge

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

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

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

### 3.1.4.1. Konfiguration des DELIB CLI

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

#### Konfiguration unter Windows

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

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

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

#### moduleID

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

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

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

#### moduleNR

Die moduleNR wird im DELIB Configuration Utility vergeben.

Diese Nummer dient zur Identifizierung identischer Hardware.

Der Standardwert ist die 0.

#### RO-ETH\_ipAddress

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

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

### 3.1.4.2. DELIB CLI Beispiele

#### Digitale Ausgänge

```
delib_cli DO1 17 1
```

→ schaltet das 18. digitale Relais an

```
delib_cli DO1 3 0
```

→ schaltet das 4. digitale Relais aus

```
delib_cli DO8 0 255
```

→ schaltet die digitalen Relais 1 bis 8 an

```
delib_cli DO16 0 0
```

→ schaltet die digitalen Relais 1 bis 16 aus

```
delib_cli DO16 16 65535
```

→ schaltet die digitalen Relais 17 bis 32 an

```
delib_cli DO32 0 4294967295
```

→ schaltet die digitalen Relais 1 bis 32 an

## Digitale Eingänge

```
delib_cli DI1 3
```

Beispiel eines Rückgabewertes: 1

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

```
delib_cli DI8 0 hex
```

Beispiel eines Rückgabewertes: **0xC8**

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

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

```
delib_cli DI16 0 hex
```

Beispiel eines Rückgabewertes: **0xE0C0**

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

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

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

```
delib_cli DI8 0 hex nunit
```

Beispiel eines Rückgabewertes: **FF**

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

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

```
delib_cli FF 0
```

Beispiel eines Rückgabewertes: **192**

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

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

```
delib_cli FF 32
```

Beispiel eines Rückgabewertes: **65535**

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

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

```
delib_cli FF 0 hex
```

Beispiel eines Rückgabewertes: **0xD00**

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

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



### **Analoge Ausgänge**

`delib_cli AO 7 4711`

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

`delib_cli AO 6 0x4711`

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

`delib_cli AO 7 3.7V`

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

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

`delib_cli AO 7 13.3mA`

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

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

### Analoge Eingänge

```
delib_cli AI 2
```

Beispiel eines Rückgabewertes: **1234**

→ liest den Wert des 3. analogen Eingangs als dezimalzahl

```
delib_cli AI 2 hex
```

Beispiel eines Rückgabewertes: **0x1FA**

→ liest den Wert des 3. analogen Eingangs als hexadezimalzahl

```
delib_cli AI 2 V
```

Beispiel eines Rückgabewertes: **12.500000V**

→ liest die Spannung des 3. analogen Eingangs als kommazahl

```
delib_cli AI 2 mA
```

Beispiel eines Rückgabewertes: **20.551600mA**

→ liest den Strom des 3. analogen Eingangs als kommazahl

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

```
delib_cli AI 3 hex nunit
```

Beispiel eines Rückgabewertes: **1FA**

→ liest den Wert des 4. analogen Eingangs als hexadezimalzahl

```
delib_cli AI 3 V nunit
```

Beispiel eines Rückgabewertes: **12.500000**

→ liest die Spannung des 4. analogen Eingangs als kommazahl

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

#### **3.1.5.1. LabVIEW**

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

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

#### **3.1.5.2. ProfiLab**

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

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

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

##### **Digitale Ein-/Ausgänge**

- Relais
- MOSFET
- Optokoppler
- Bistabile-Relais

##### **Analoge Ein-/Ausgänge**

- Analog zu digital Wandler
- Digital zu analog Wandler

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

- 8/32/64 TTL-Kanäle

### **3.1.5.3. Licht24 Pro**

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

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

### 3.1.6. Einbinden der DELIB in Programmiersprachen

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

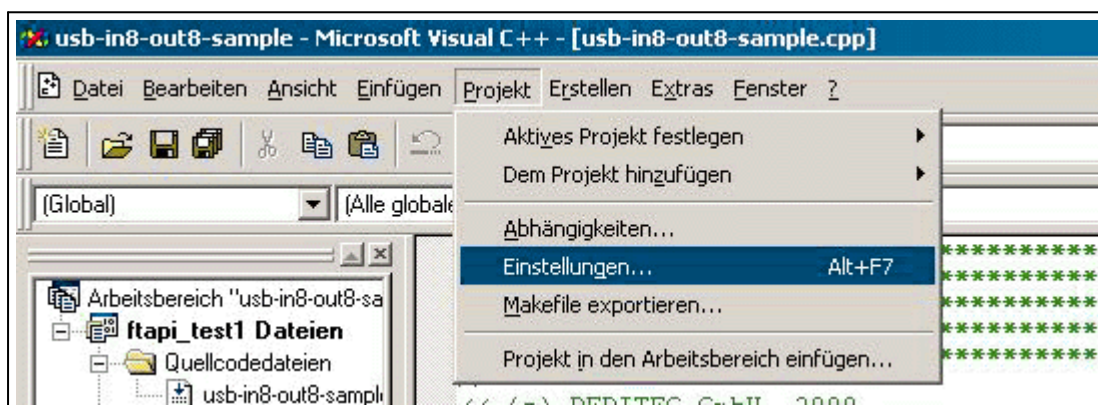
Zur Erleichterung für Verweise auf das DELIB-Include und das DELIB-Lib Verzeichnis werden bei installation der DELIB Umgebungsvariablen definiert.

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

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

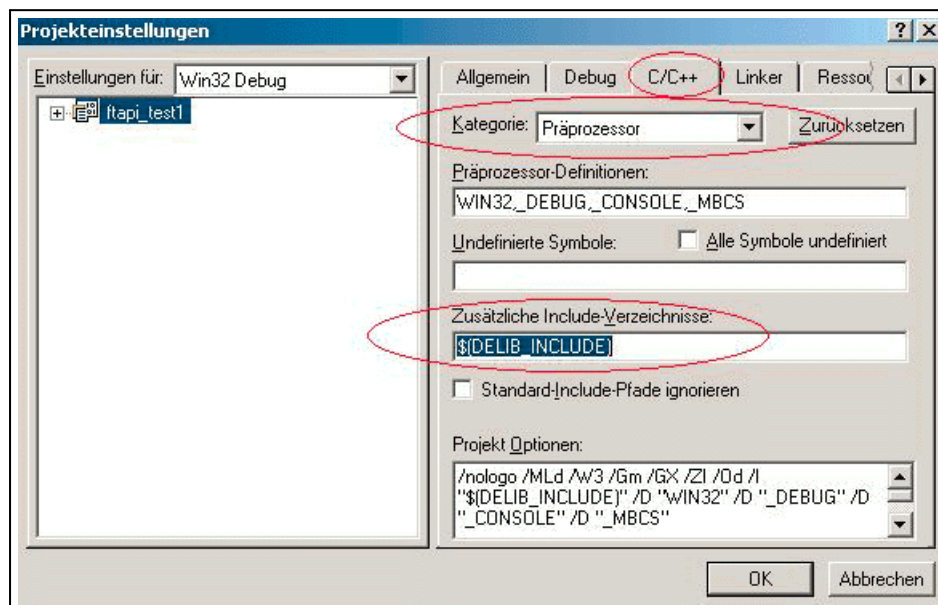
Diese werden im Folgenden in den Projekteinstellungen des Compilers eingetragen.

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



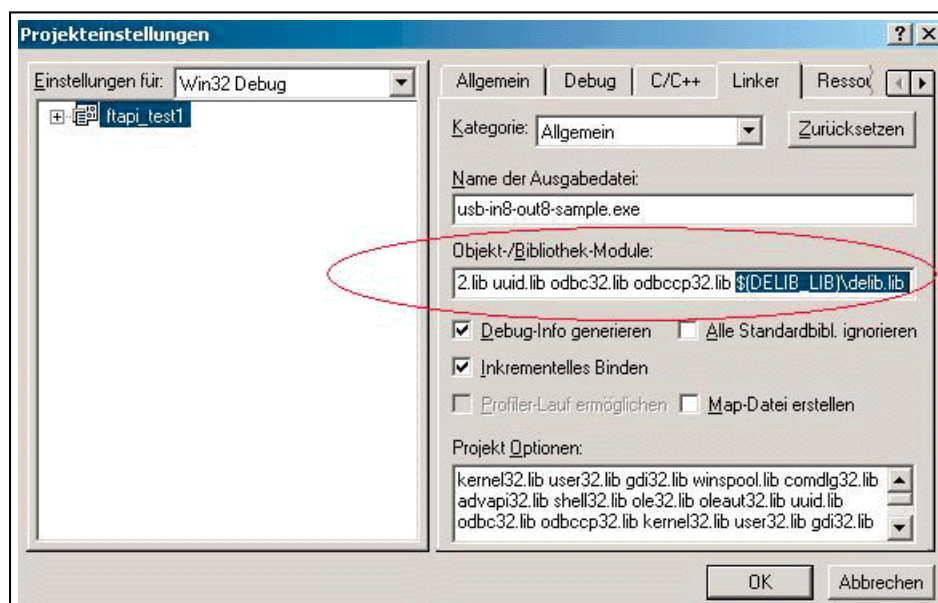
### DELIB.H Eintrag in den Visual-C/C++ Projekt Einstellungen

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



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

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



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

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

#### 32 Bit DELIB Installation

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

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

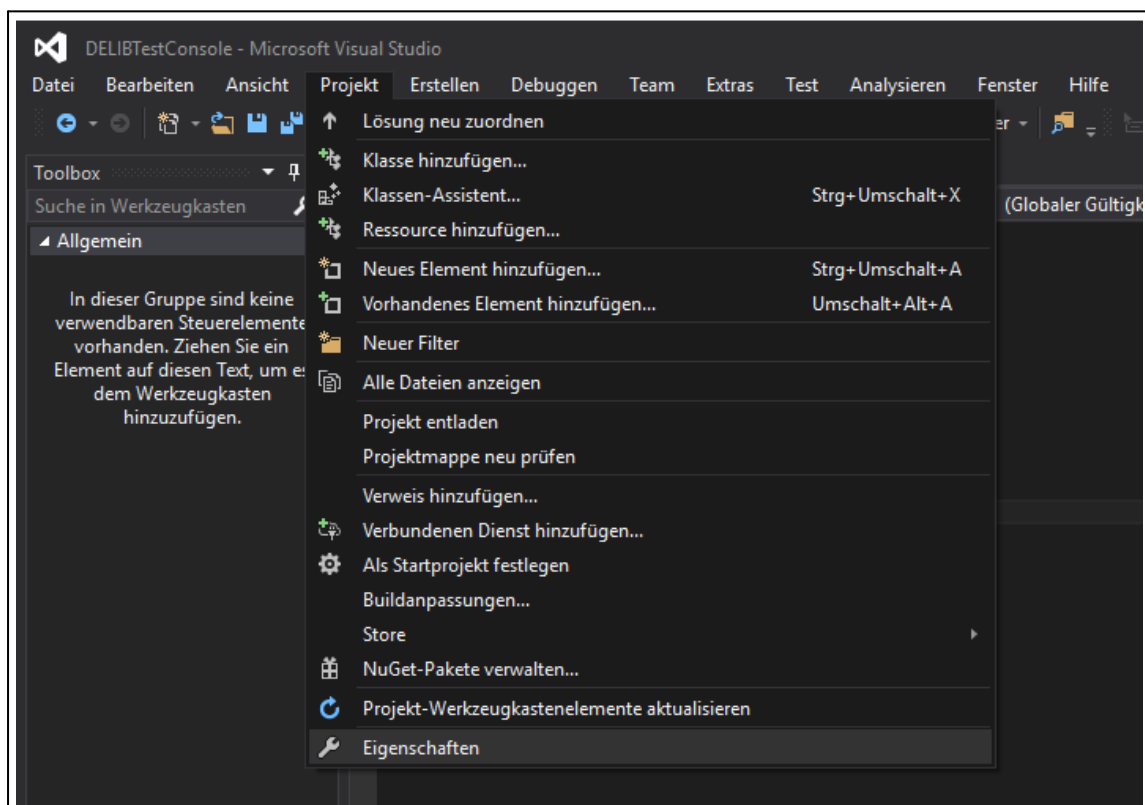
#### 64 Bit DELIB Installation

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

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

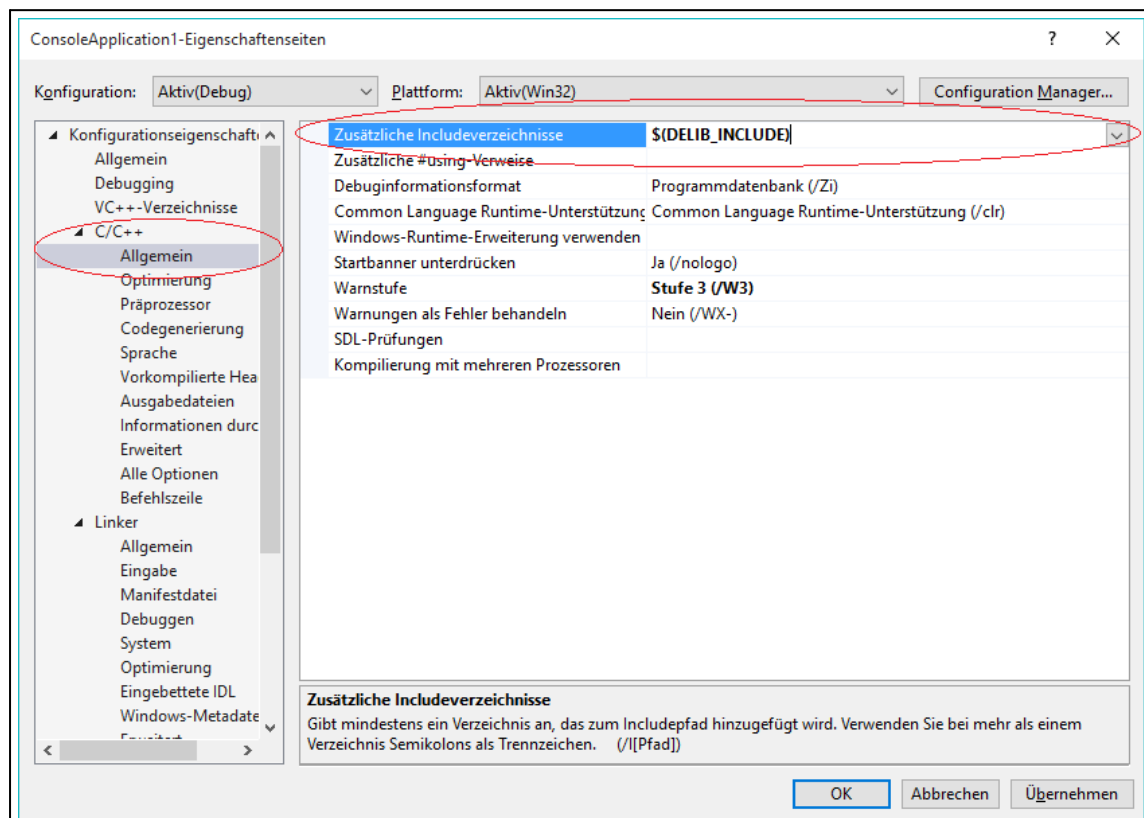
Diese werden im Folgenden in den Projekteinstellungen des Compilers eingetragen.

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



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

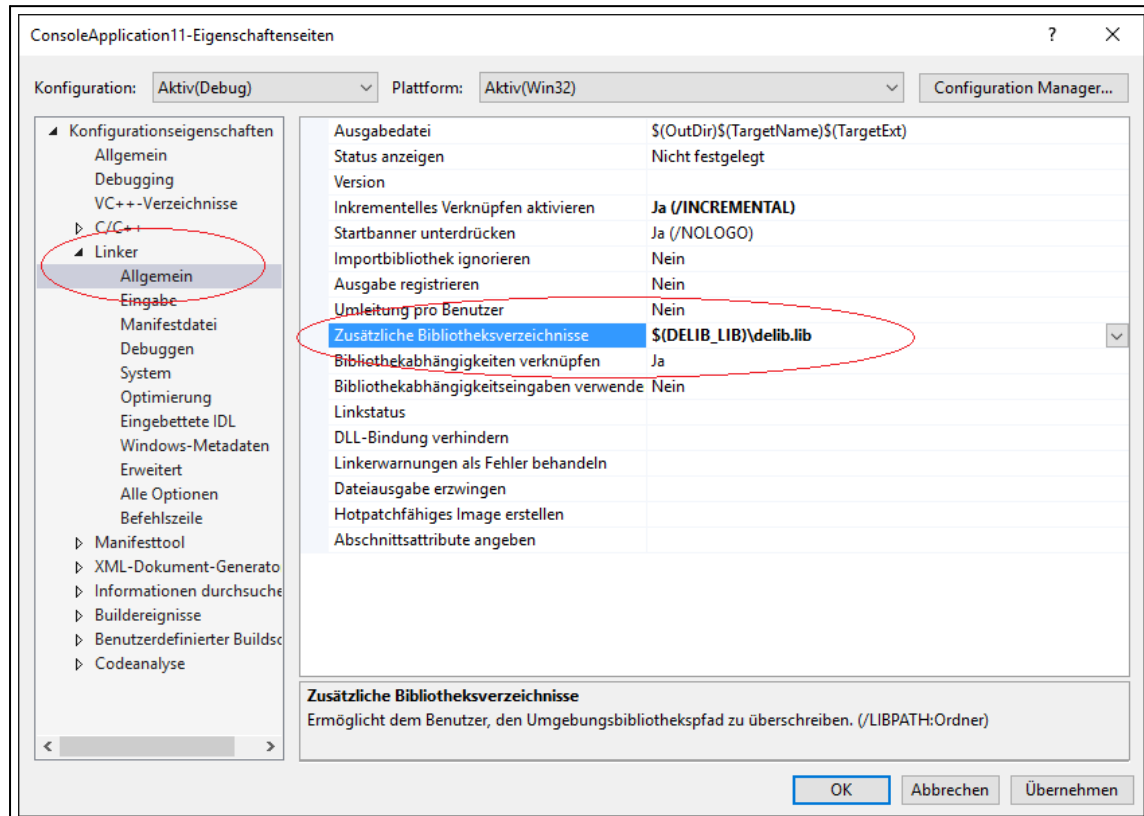
Unter dem Reiter "C/C++" die "Kategorie" Allgemein auswählen und unter "Zusätzliche Include Verzeichnisse" "\$(DELIB\_INCLUDE)" eintragen.





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

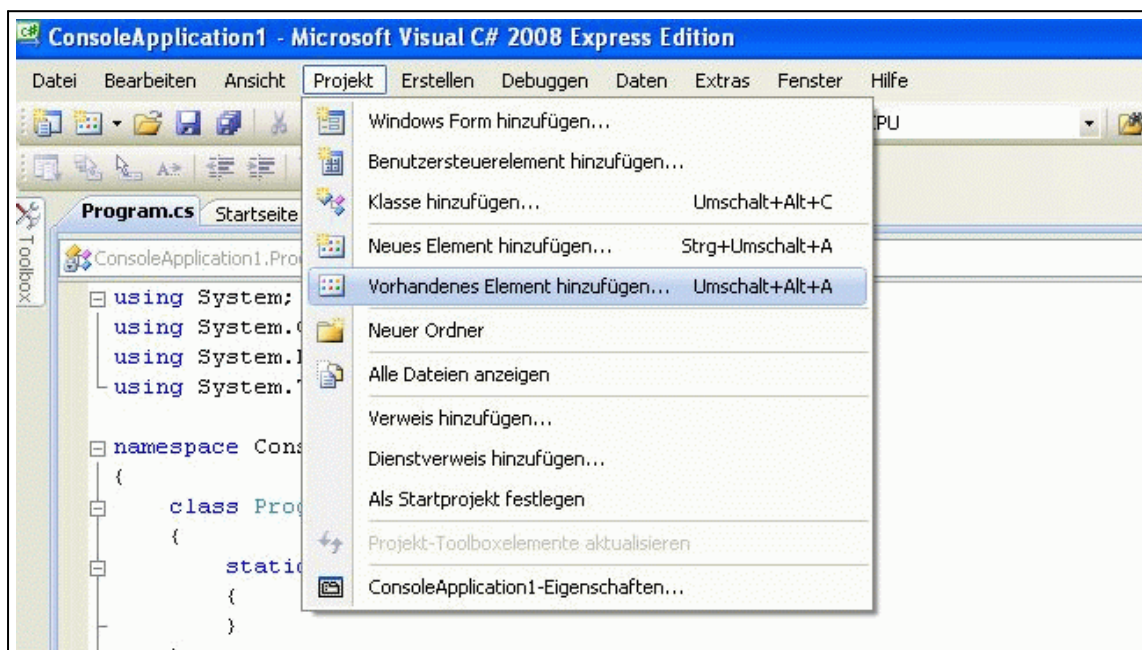
Unter dem Reiter "Linker" bei "Allgemein" "\$ (DELIB\_LIB)\delib.lib" eintragen.



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

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

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



Folgenden Verweis in Ihrem Programm hinzufügen:

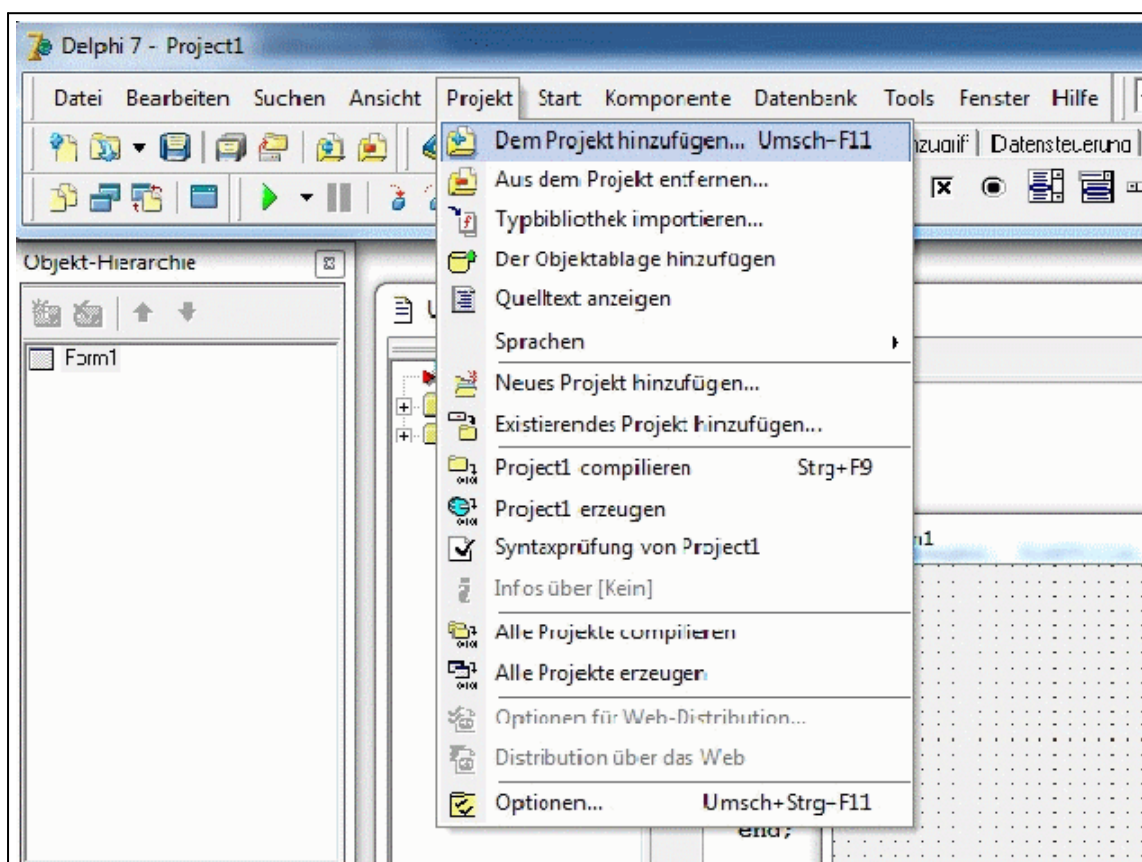
*using DeLib;*

#### 3.1.6.4. Einbinden der DELIB in Delphi

Die benötigte Datei für Delphi befindet sich im Verzeichnis

C:\Programme\DEDITEC\DELIB\include.

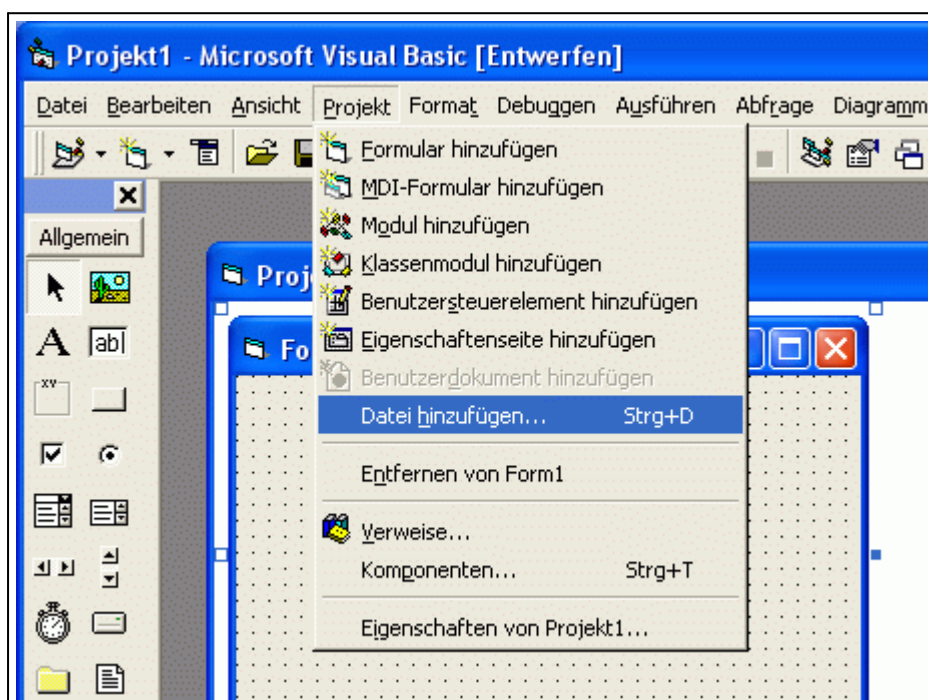
Delphi starten und über das Menue "Projekt → dem Projekt hinzufügen" im Verzeichnis C:\Programme\DEDITEC\DELIB\include\ die Datei delib.pas zum Importieren öffnen.



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

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

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

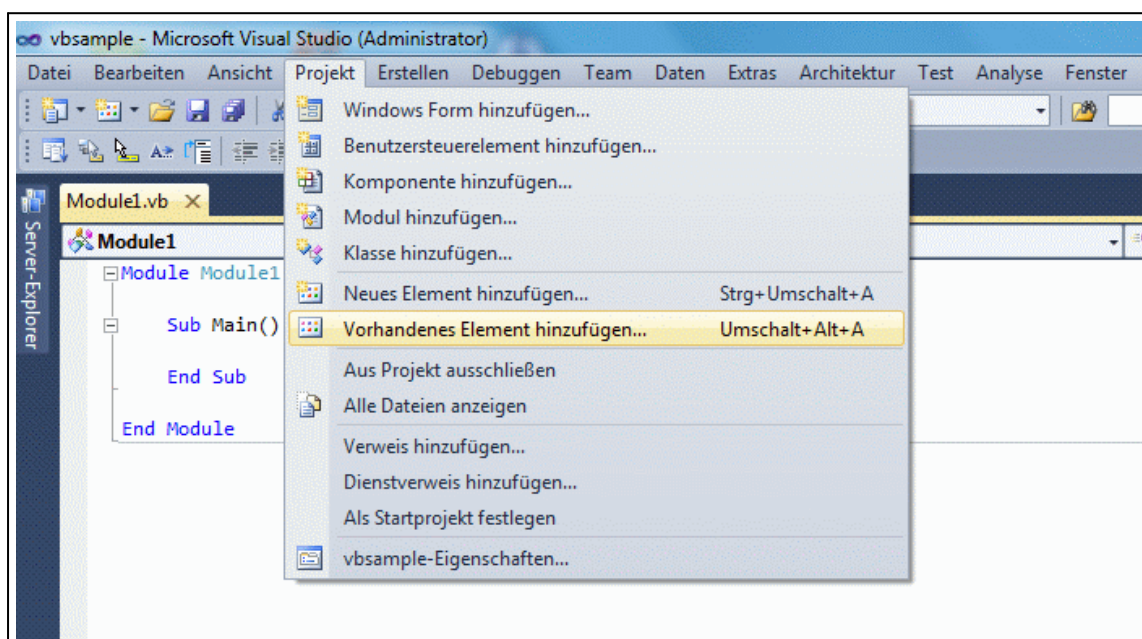


### 3.1.6.6. Einbinden der DELIB in Visual-Basic.NET (VB.NET)

Die benötigte Datei für VB.NET befindet sich im Verzeichnis

C:\Programme\DEDITEC\DELIB\include.

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

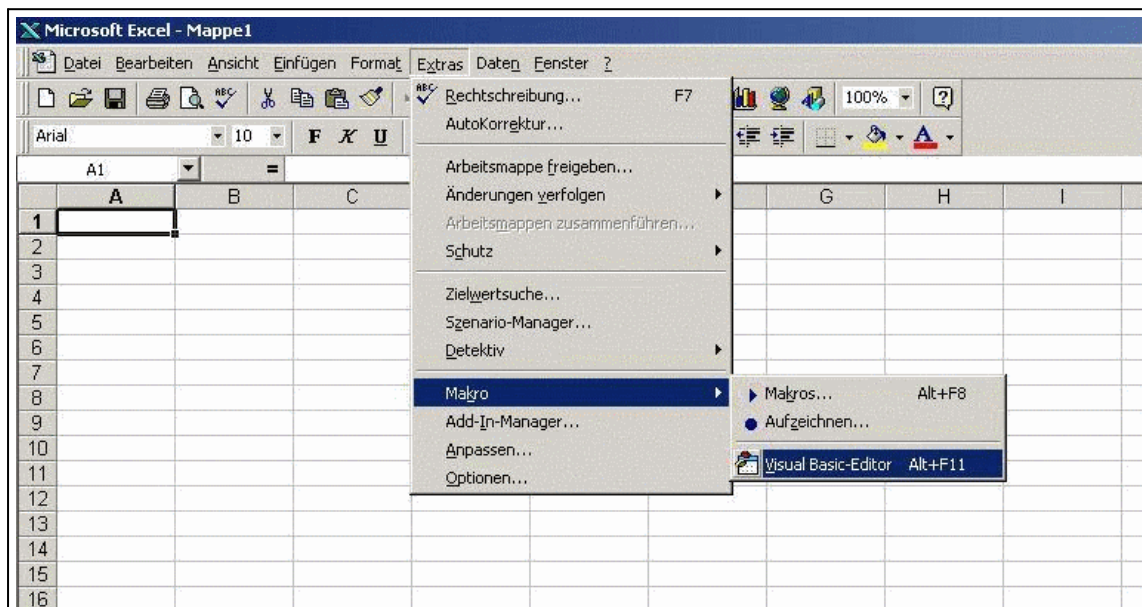


### 3.1.6.7. Einbinden der DELIB in MS-Office (VBA)

Die benötigte Datei für VBA befindet sich im Verzeichnis

C:\Programme\DEDITEC\DELIB\include.

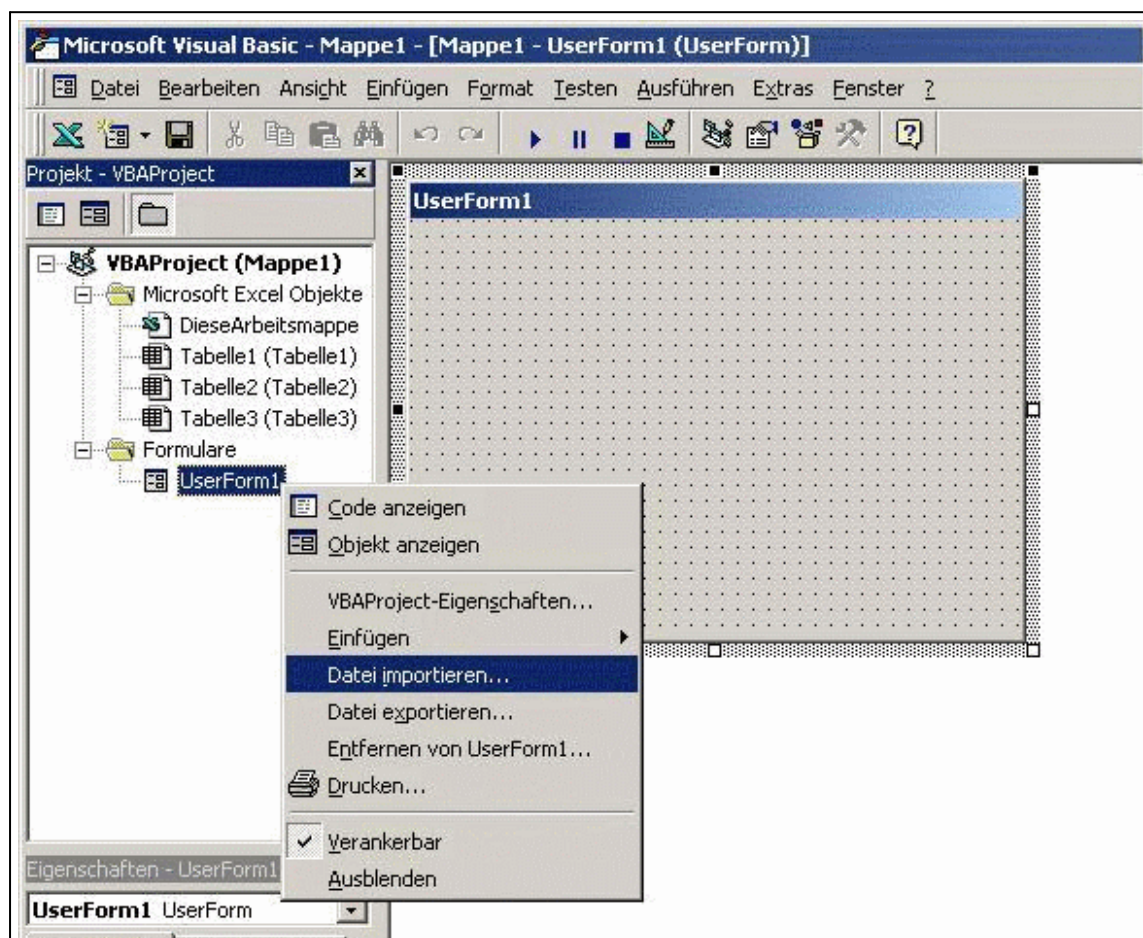
Microsoft Excel starten und über das Menue "Extras → Makro → Visual Basic Editor" öffnen.





## Erstellen der UserForm

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



### 3.1.6.8. Einbinden der DELIB in LabVIEW

#### 3.1.6.8.1. Einbinden der DELIB in LabVIEW

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

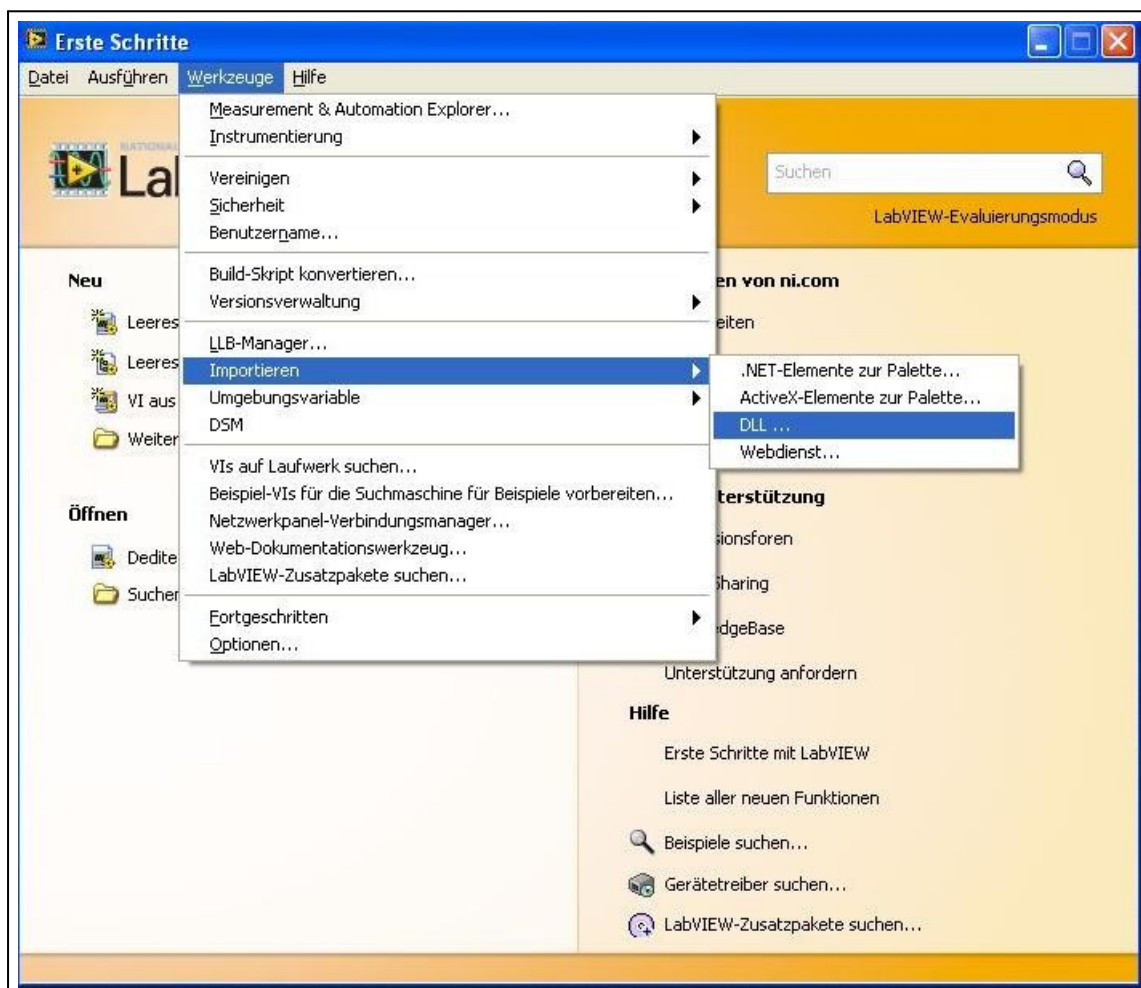
Beschreibung der Einbindung der "delib.dll" in LabVIEW Version 11

- Die benötigten Dateien für LabVIEW befinden sich im Verzeichnis

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

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

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





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

**DLL importieren**

Erstellungs- oder Aktualisierungsmodus angeben

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

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

| Projekt | DLL-Datei | Datum |
|---------|-----------|-------|
|         |           |       |
|         |           |       |
|         |           |       |
|         |           |       |
|         |           |       |
|         |           |       |
|         |           |       |
|         |           |       |
|         |           |       |
|         |           |       |
|         |           |       |
|         |           |       |
|         |           |       |
|         |           |       |
|         |           |       |

Zurück Weiter Abbrechen Hilfe

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

**DLL importieren**

**DLL- und Header-Datei wählen**

NATIONAL INSTRUMENTS

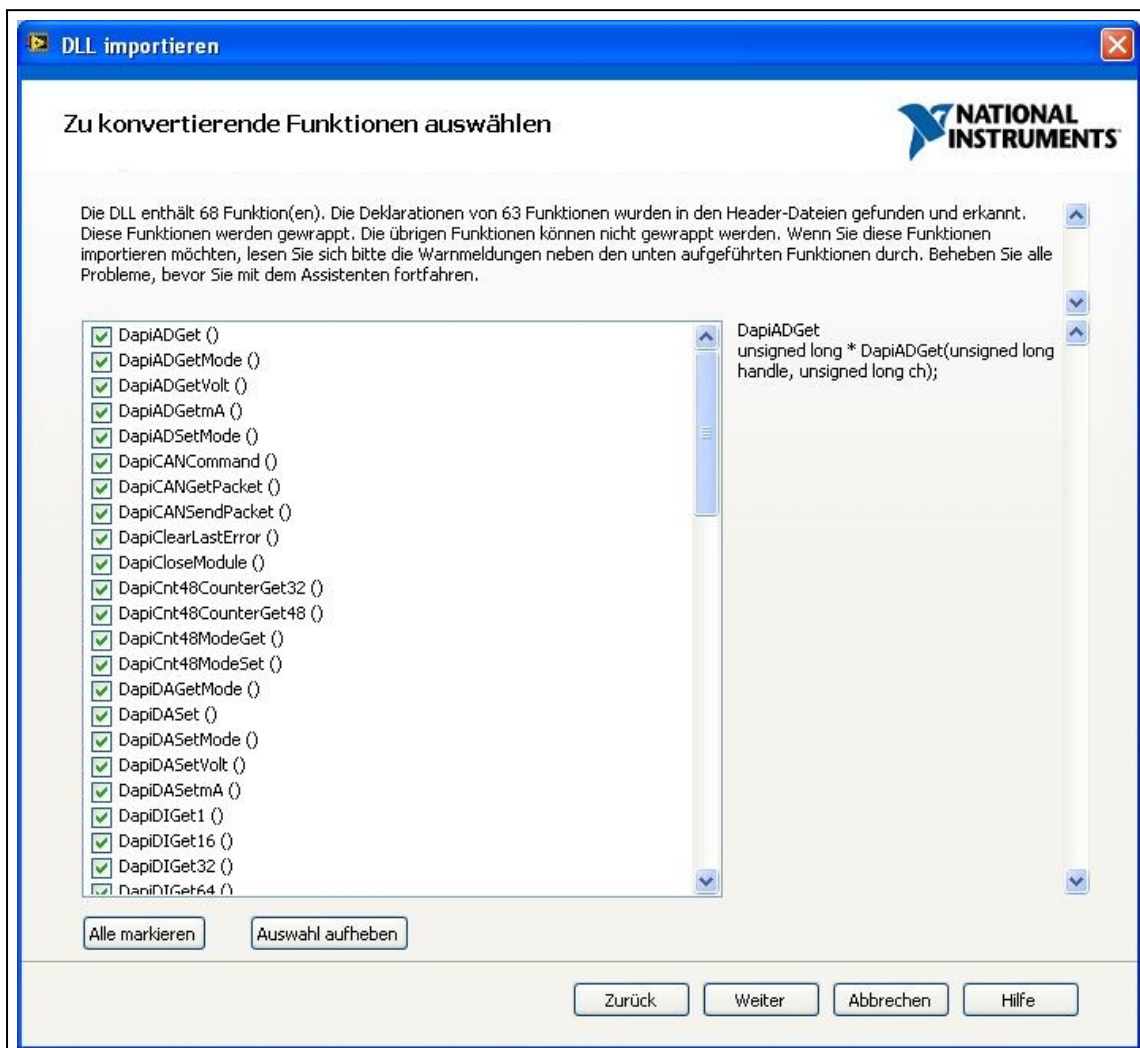
Bibliothek (\*.dll)  
C:\WINDOWS\system32\delib.dll

☐ DLL befindet sich nicht auf lokalem Computer

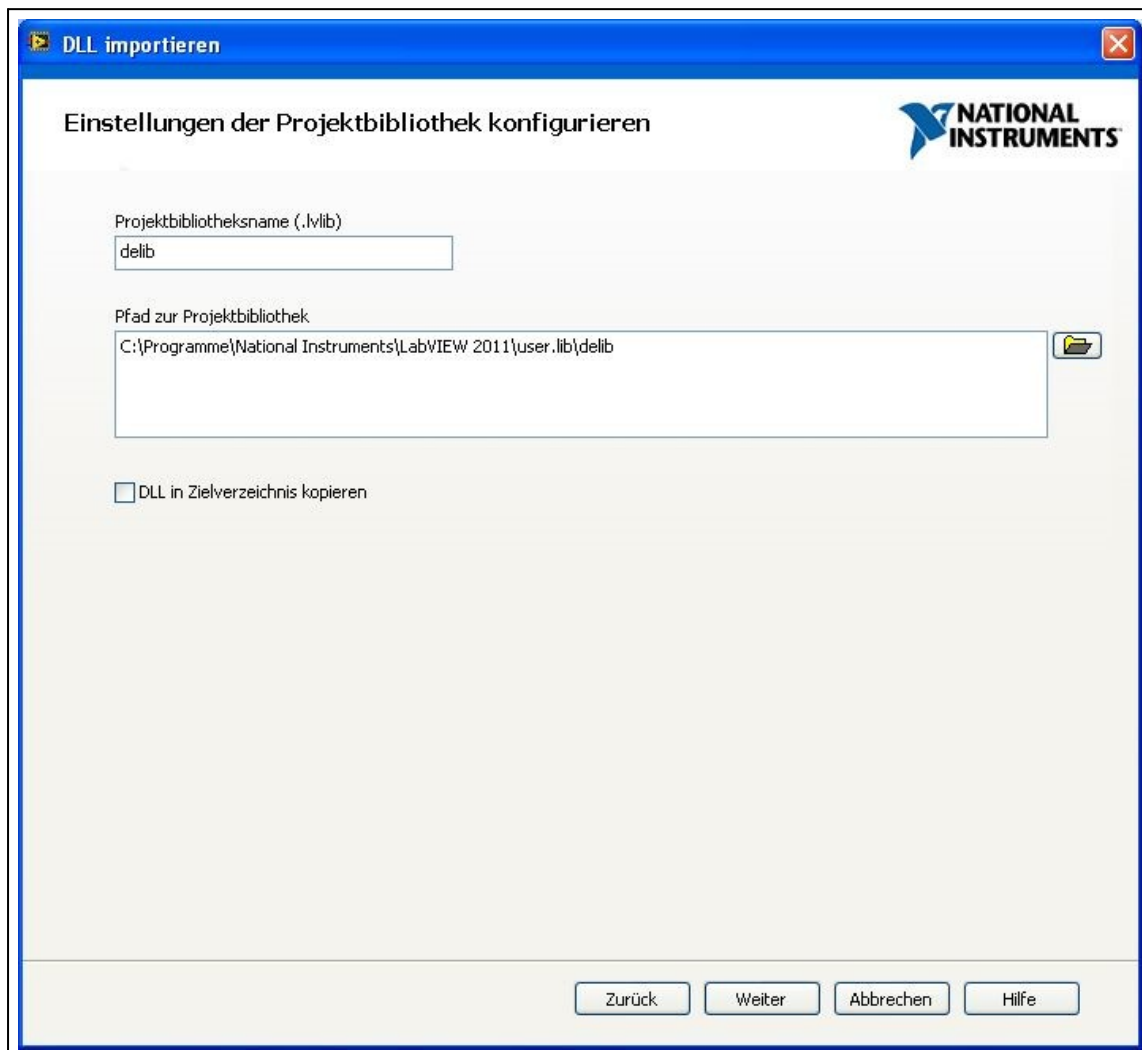
Header-Datei (.h)  
C:\Programme\DEDITEC\DELIB\include\delib.h

Zurück Weiter Abbrechen Hilfe

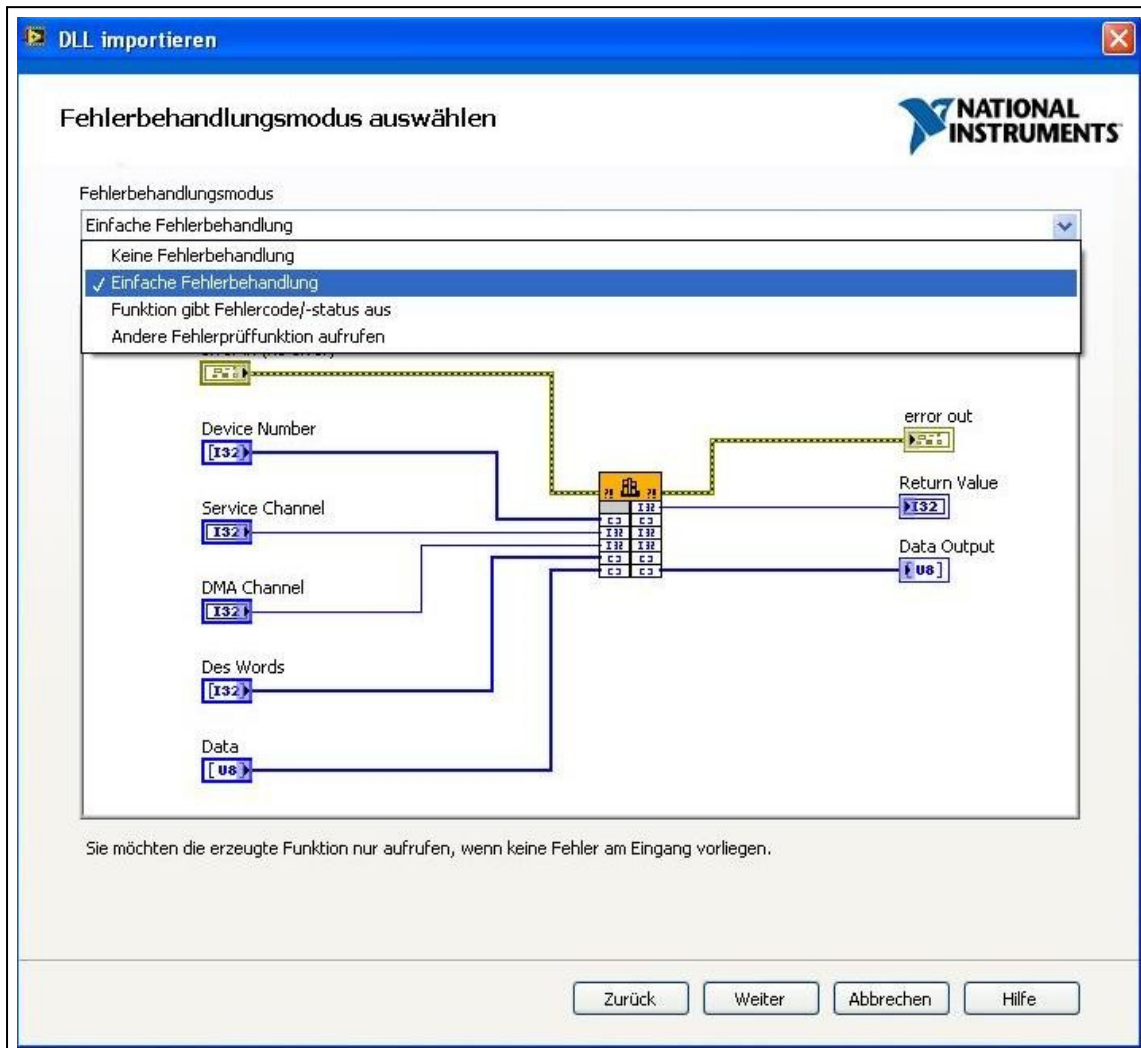
- Nochmals auf "Weiter" klicken um fortzufahren.
- Die Header-Datei wird nun analysiert. Anschließend fahren Sie im folgendem Fenster wieder mit "Weiter" fort.



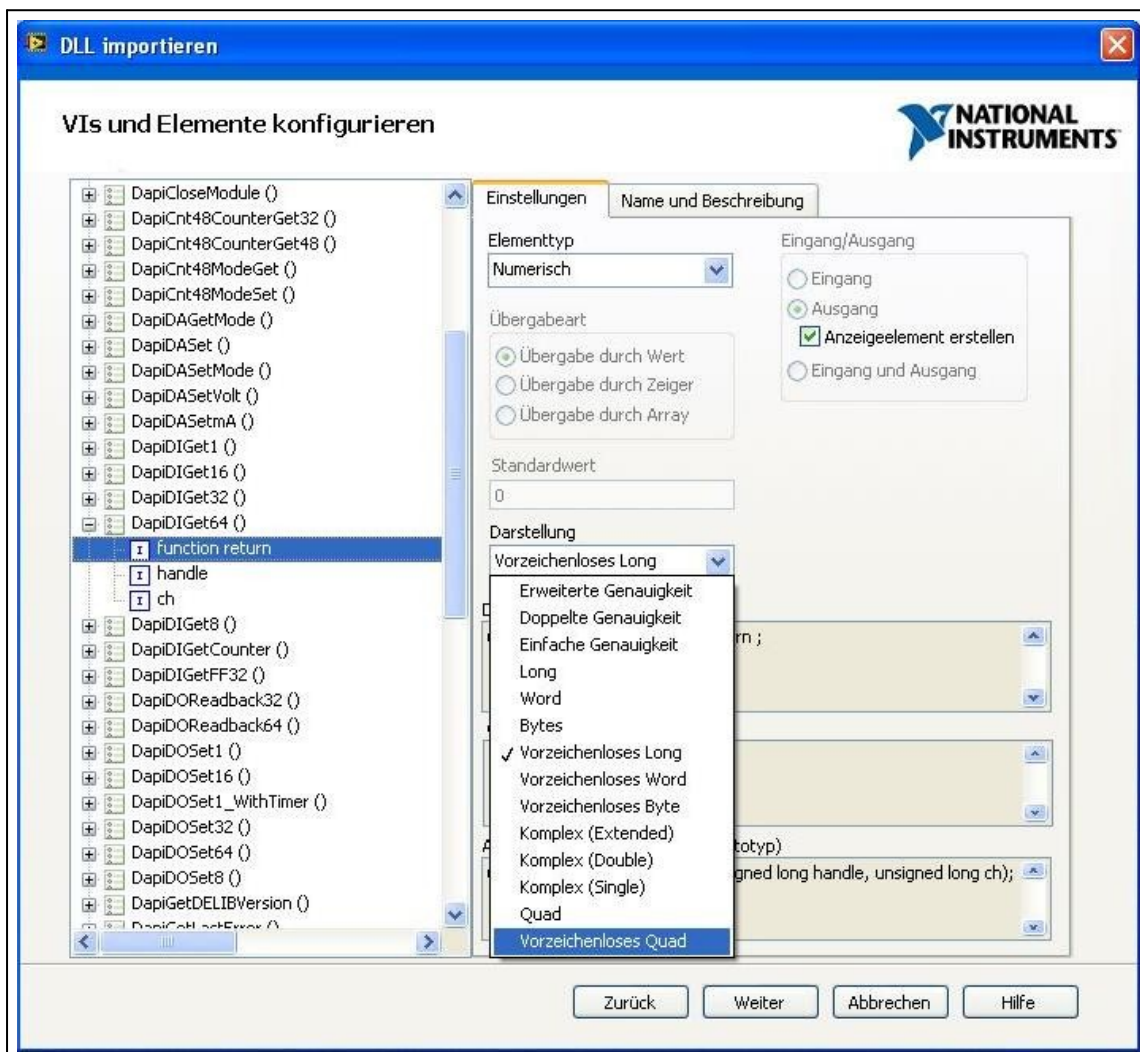
- Den weiteren Anweisungen folgen, bzw. die Konfiguration und den Speicherort für die VIs anpassen.



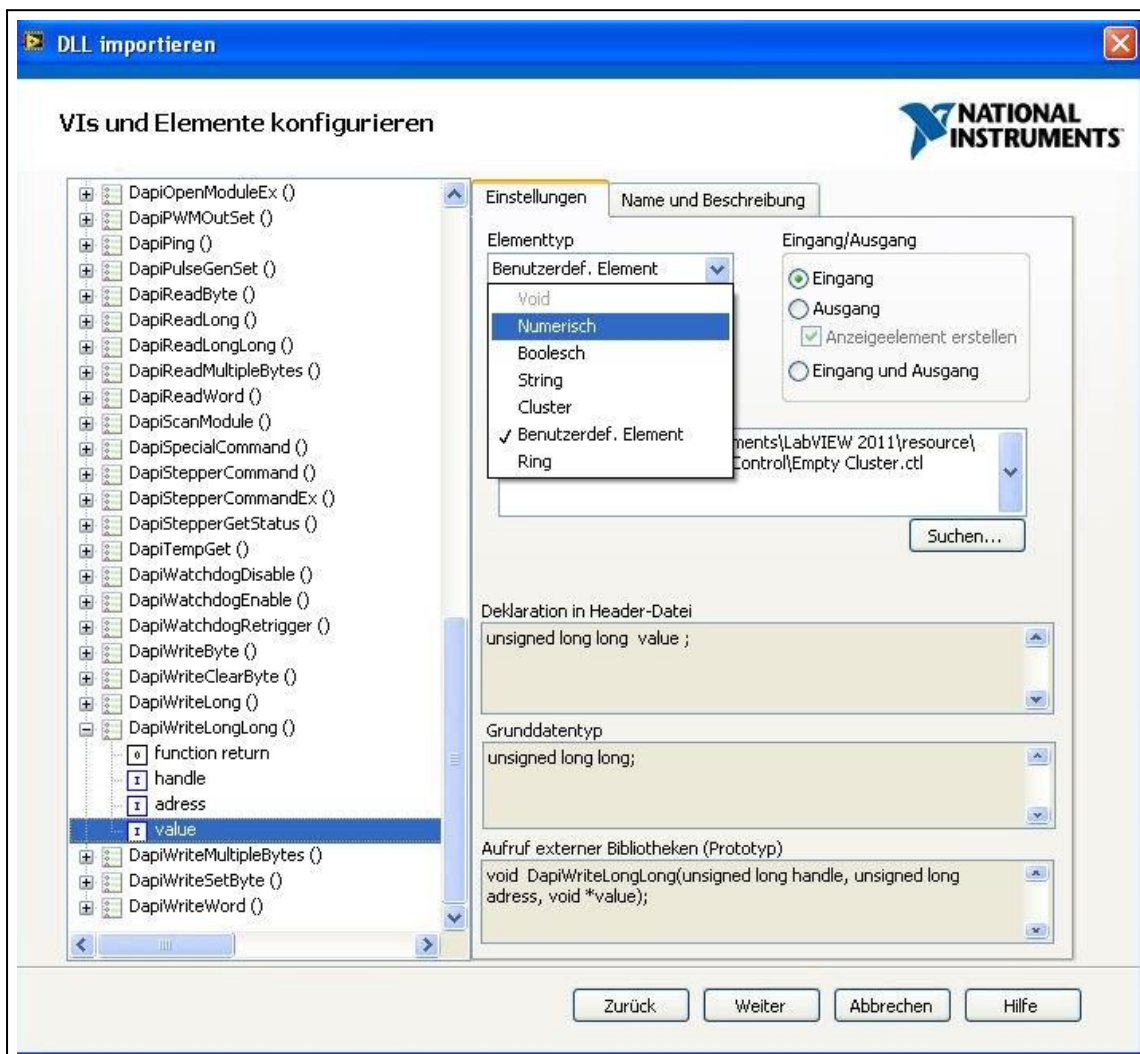
- Im folgendem Fenster wählen Sie im Drop-Down-Menü die Option "Einfache Fehlerbehandlung" aus und fahren mit "Weiter" fort.



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

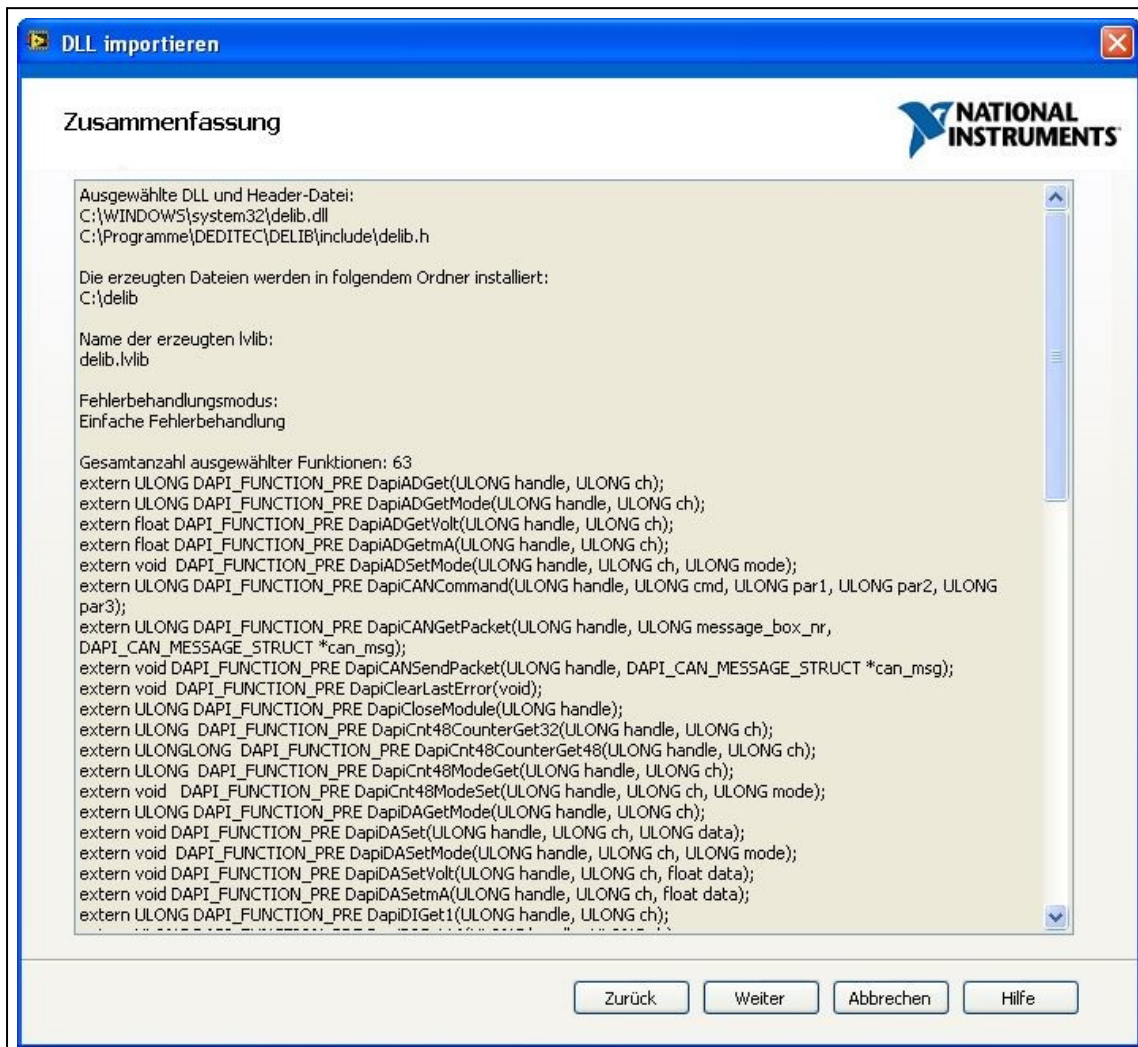


- Bei manchen VIs muss zusätzlich noch der Elementtyp auf "Numerisch" geändert werden und anschließend die Darstellung auf "Vorzeichenloses Quad"
- Folgende VIs müssen bearbeitet werden:
  - DapiWriteLongLong (value)
  - DapiReadLongLong (function return)





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



- Die VIs werden nun erzeugt und können verwendet werden.



### 3.1.6.8.2. Verwendung der VIs in LabVIEW

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

Diese Defines werden in LabVIEW nicht unterstützt.

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

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

**Die Definition für die Funktion lautet:**

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

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

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

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

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

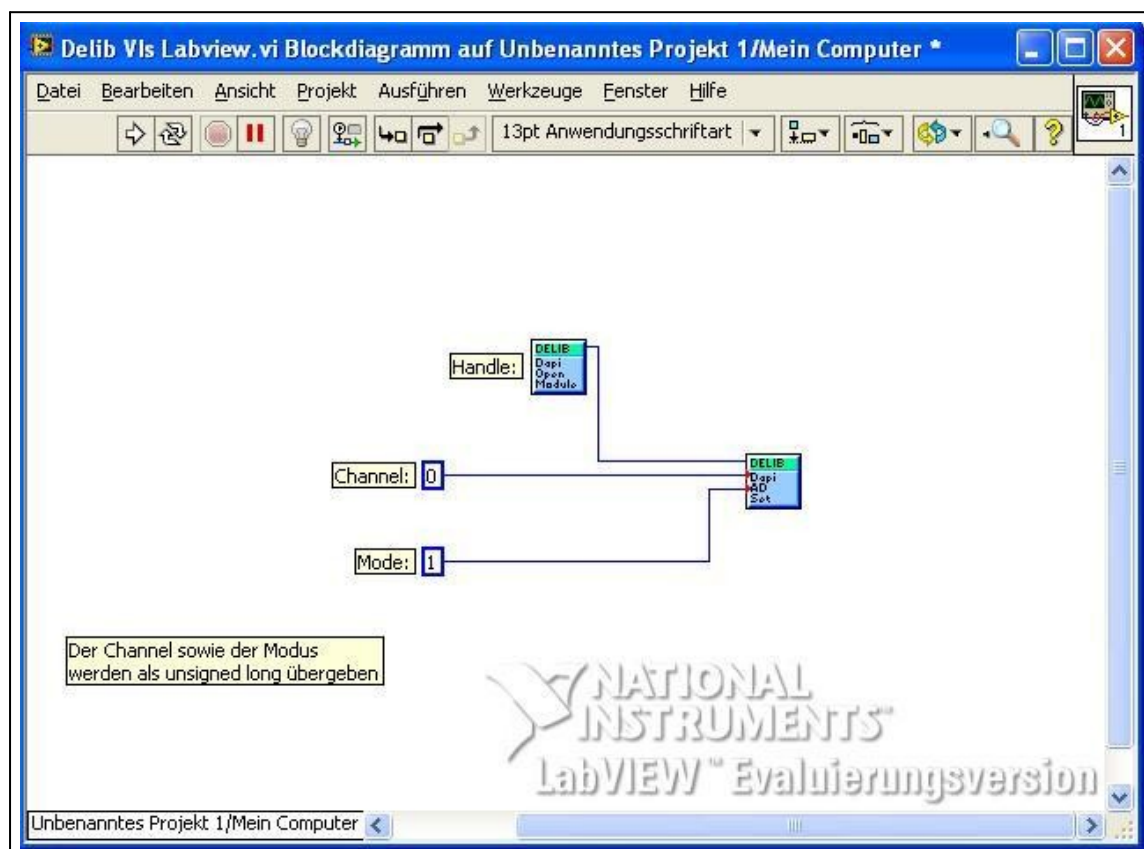
**Alternativ kann man auch folgende Schreibweise verwenden:**

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

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

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

In LabVIEW könnte die Funktion dann so aussehen:



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

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

Die Verbindung zum Modul wird mittels der Funktion DapiOpenModule hergestellt.

**Die Definition für diese Funktion lautet:**

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

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

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

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

```
// *****  
// *****  
//  
//  
#define DELIB_VERSION 0x0141 // Actual DELIB-Version  
  
// all Modul-ID's  
#define USB_Interface8 1 // USB-Controller8/USB-TTL-IN8-OUT8  
#define USB_CAN_STICK 2 // USB-CAN-Stick  
#define USB_LOGI_500 3 // USB-LOGI-500/USB-LOGI-250  
#define RO_USB2 4 // RO-CPU2 / 480 MBit/sec  
#define RO_SER 5 // RO-SER-Serie  
#define USB_BITP_200 6 // USB-BITP-200  
#define RO_USB1 7 // RO-USB-Serie  
#define RO_USB 7 // RO-USB-Serie  
#define RO_ETH 8 // RO-ETH-Serie  
#define USB_MINI_STICK 9 // USB-MINI-Stick-Serie  
#define USB_LOGI_18 10 // USB-LOGI-100  
#define RO_CAN 11 // RO-CAN-Serie  
#define USB_SPI_MON 12 // USB_SPI_MON  
#define USB_WATCHDOG 13 // USB_Watchdog  
#define USB_OPTOIN_8 14 // USB-OPTOIN8 / USB-RELAIS-8  
#define USB_RELAIS_8 14 // USB-OPTOIN8 / USB-RELAIS-8  
#define USB_OPTOIN_8_RELAIS_8 15 // USB-OPTOIN-8-RELAIS-8  
#define USB_OPTOIN_16_RELAIS_16 16 // USB-OPTOIN-16-RELAIS-16  
#define USB_OPTOIN_32 16 // USB-OPTOIN-16-RELAIS-16  
#define USB_RELAIS_32 16 // USB-OPTOIN-16-RELAIS-16  
#define USB_OPTOIN_32_RELAIS_32 17 // USB-OPTOIN-32-RELAIS-32  
#define USB_OPTOIN_64 17 // USB-OPTOIN-32-RELAIS-32  
#define USB_RELAIS_64 17 // USB-OPTOIN-32-RELAIS-32  
#define USB_TTL_32 18 // USB-TTL-32  
#define USB_TTL_64 18 // USB-TTL-64  
  
#define MAX_NR_OF_MODULES 18
```

### Beispiel in C:

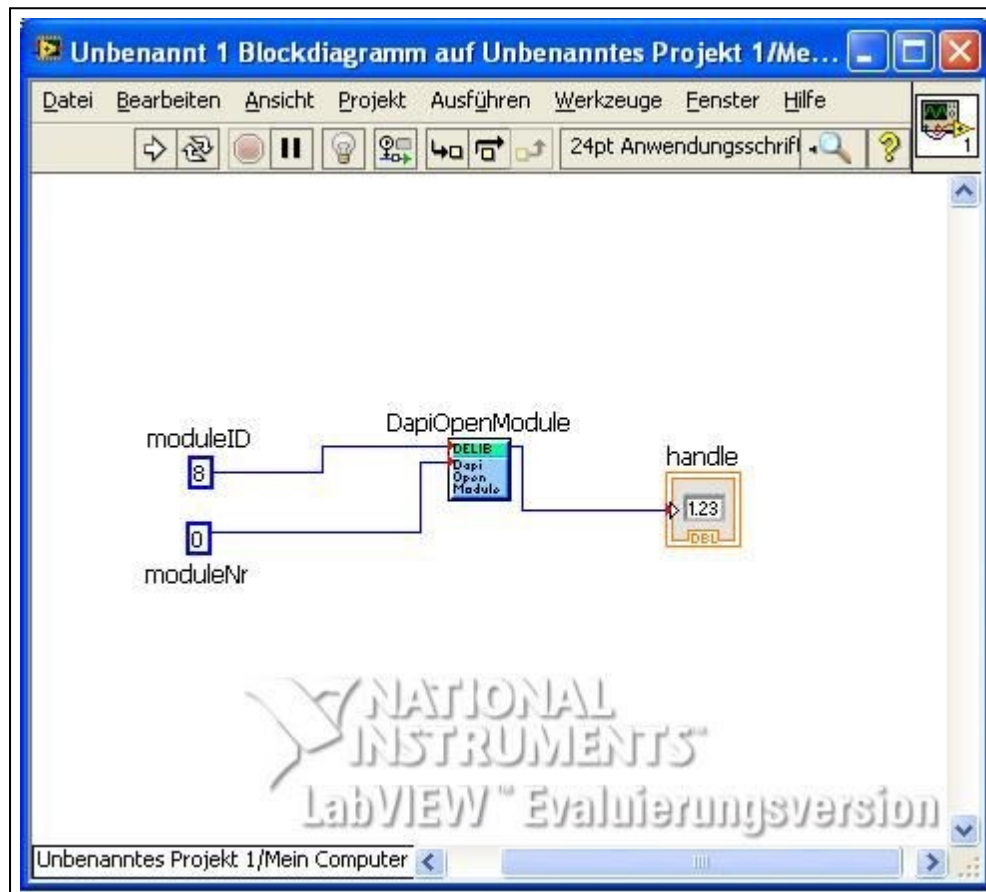
```
handle = DapiOpenModule(RO_ETH, 0);    // öffnet ein RO-ETH-Modul mit  
Modul-Nr 0.
```

### Alternativ kann man auch folgende Schreibweise verwenden:

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

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

### Beispiel in Labview:



### 3.1.6.9. Einbinden der DELIB in Java

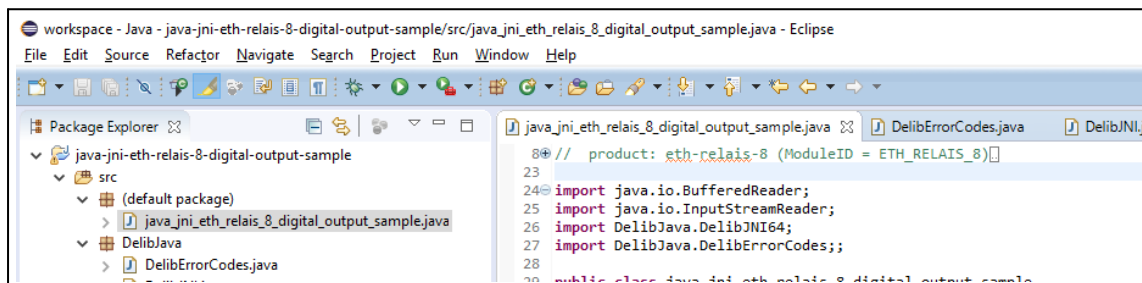
Die benötigten Dateien für Java befinden sich, je nach DELIB-Installation, in folgendem Verzeichnis

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

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

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

Anschließend müssen die verwendeten Module noch importiert werden.



## 3.2. DELIB Treiberbibliothek

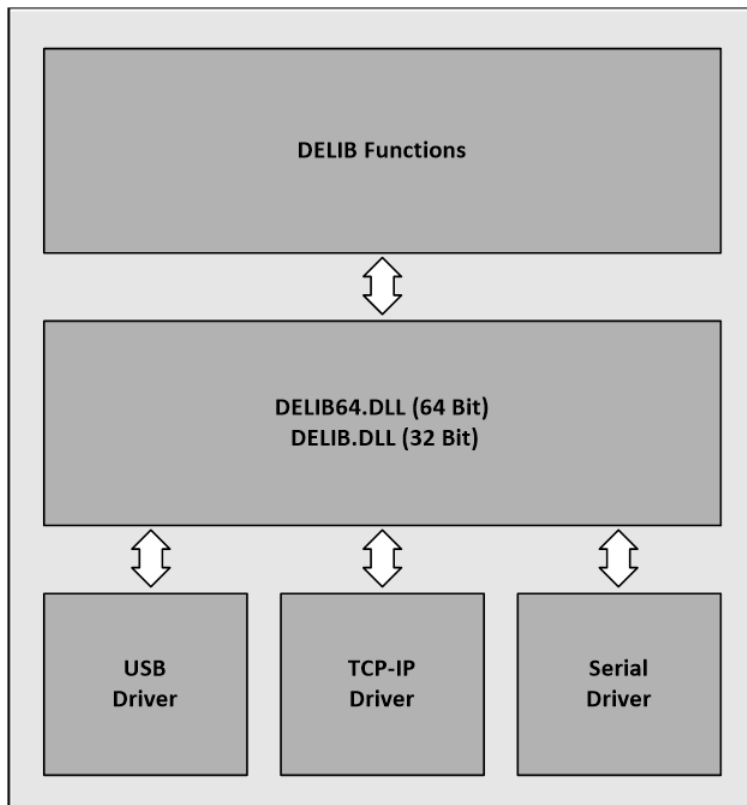
Die DELIB Treiberbibliothek enthält die DELIB-API und verschiedene Programme für den Konfigurationstest unserer Produkte.

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

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

### 3.2.1. Übersicht

Die folgende Abbildung erläutert den Aufbau der DELIB Treiberbibliothek



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

- Betriebssystem unabhängig
- Programmiersprachen unabhängig
- Produkt unabhängig

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

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

## DELIB Treiberbibliothek ETH

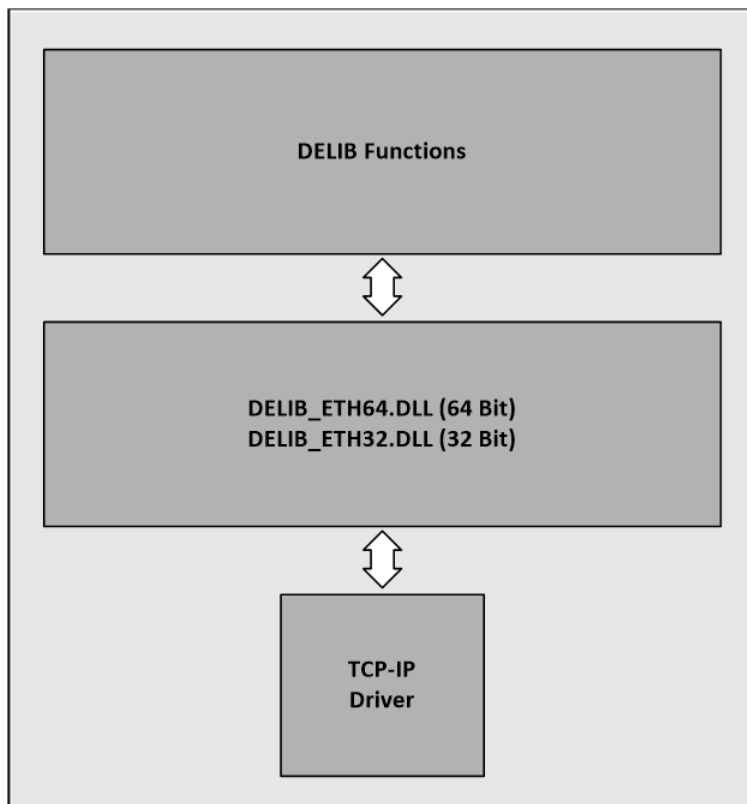
Während die DELIB für ALLE Produkte zur Verfügung steht, wird bei der DELIB-ETH auf keine weiteren Treiber zugegriffen (wie z. B. USB).

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.







### **3.2.1.1. Unterstützte Programmiersprachen**

Die folgenden Programmiersprachen werden von der DELIB Treiberbibliothek unterstützt:

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

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

### **3.2.1.2. Unterstützte Betriebssysteme**

Die folgende Betriebssysteme sind mit unserer DELIB Treiberbibliothek kompatibel:

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

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

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

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

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

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

### 3.2.2. DELIB Setup

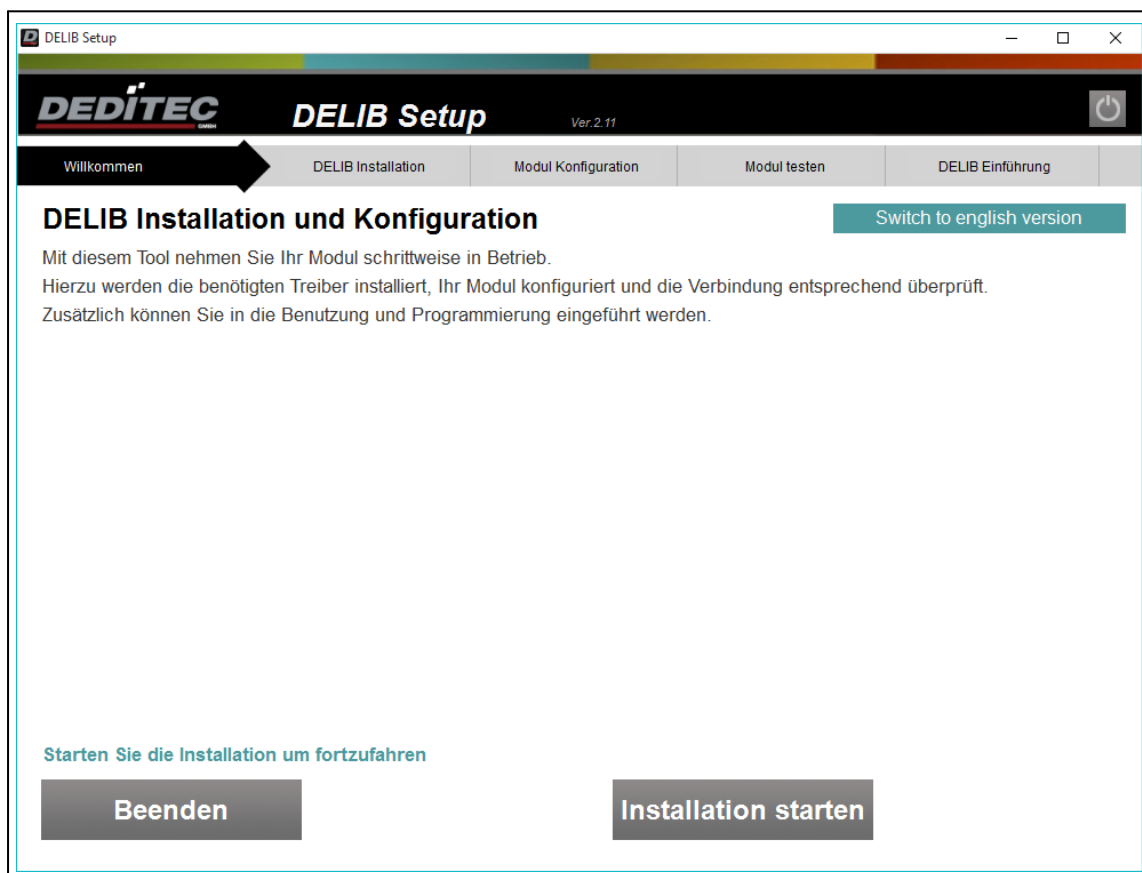
Das DELIB Setup führt Sie durch die Installation unserer DELIB Treiberbibliothek.

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

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

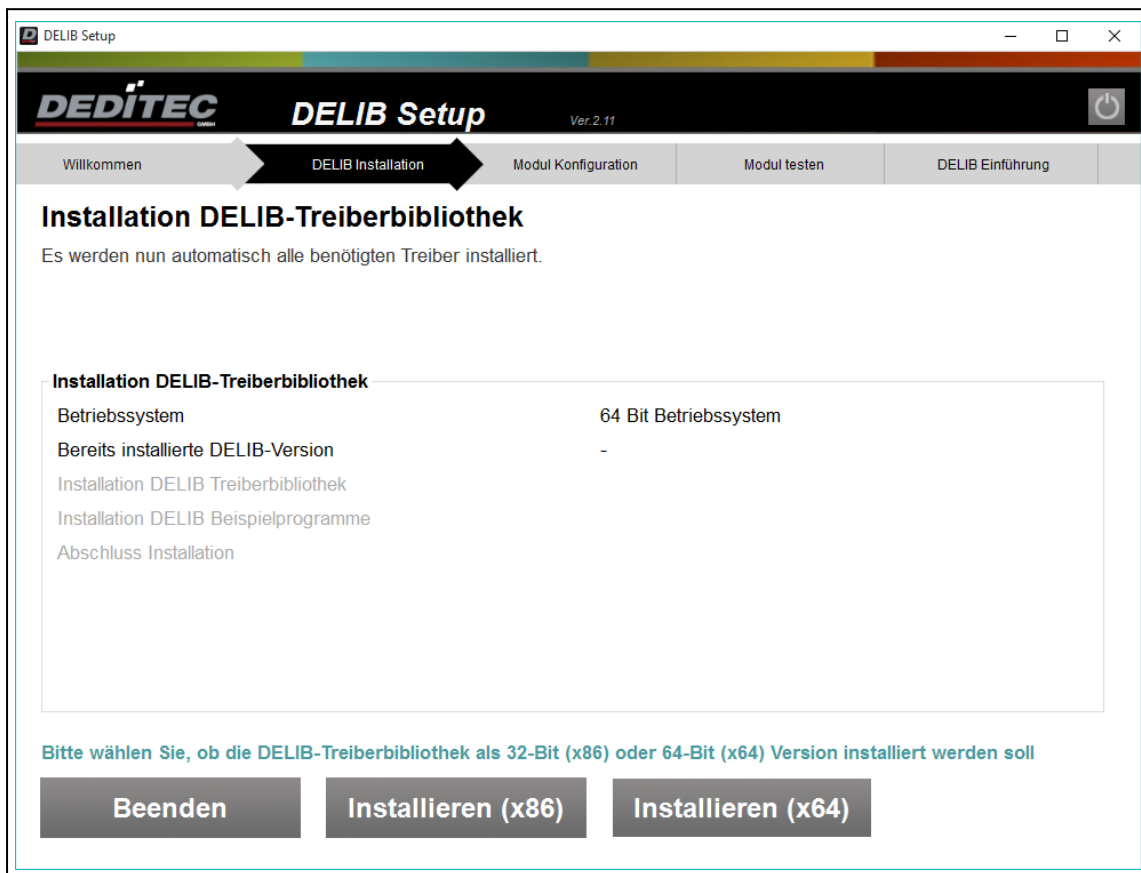
Link: <https://www.deditec.de/delib>

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

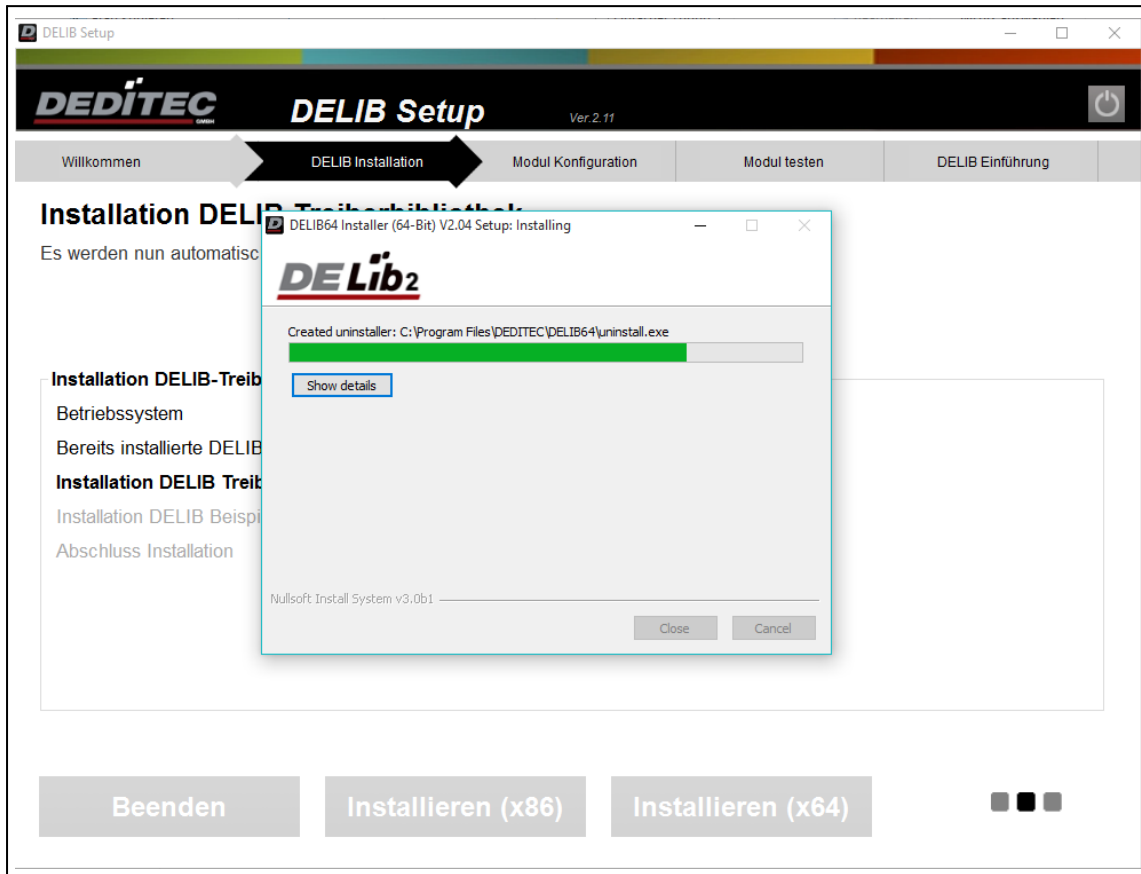


Das DELIB Setup prüft das Betriebssystem und ob bereits Versionen der DELIB Treiberbibliothek installiert sind.

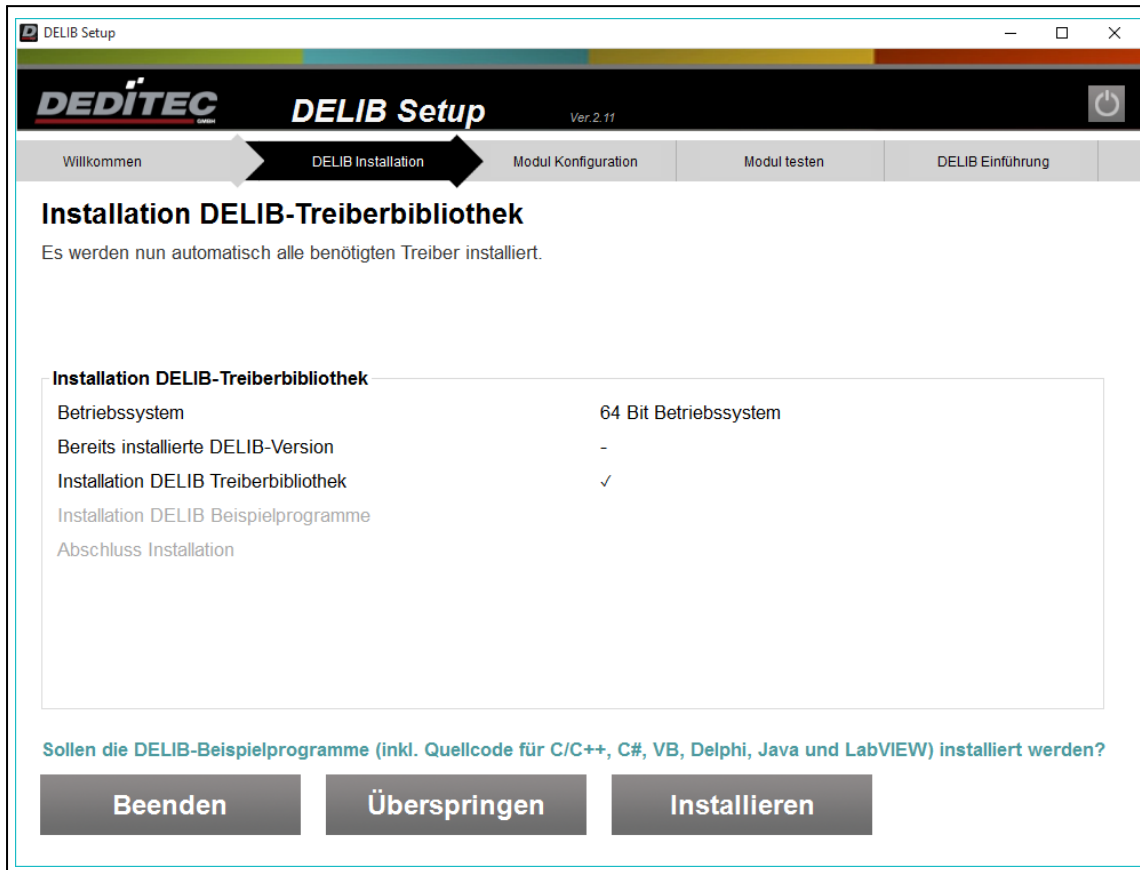
Anschließend können Sie wählen ob Sie die 32-Bit oder 64-Bit Version der DELIB Treiberbibliothek installieren möchten.



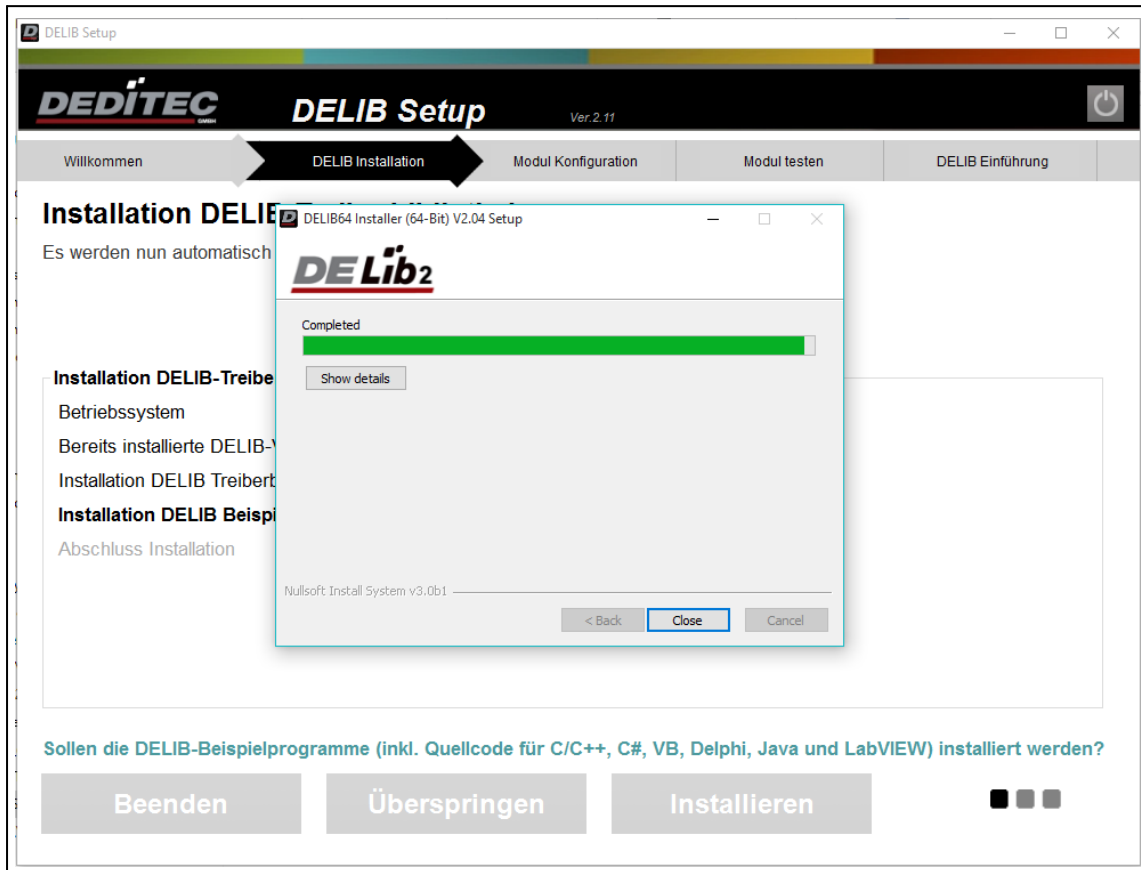
Installationsfortschritt der Treiberbibliothek.



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

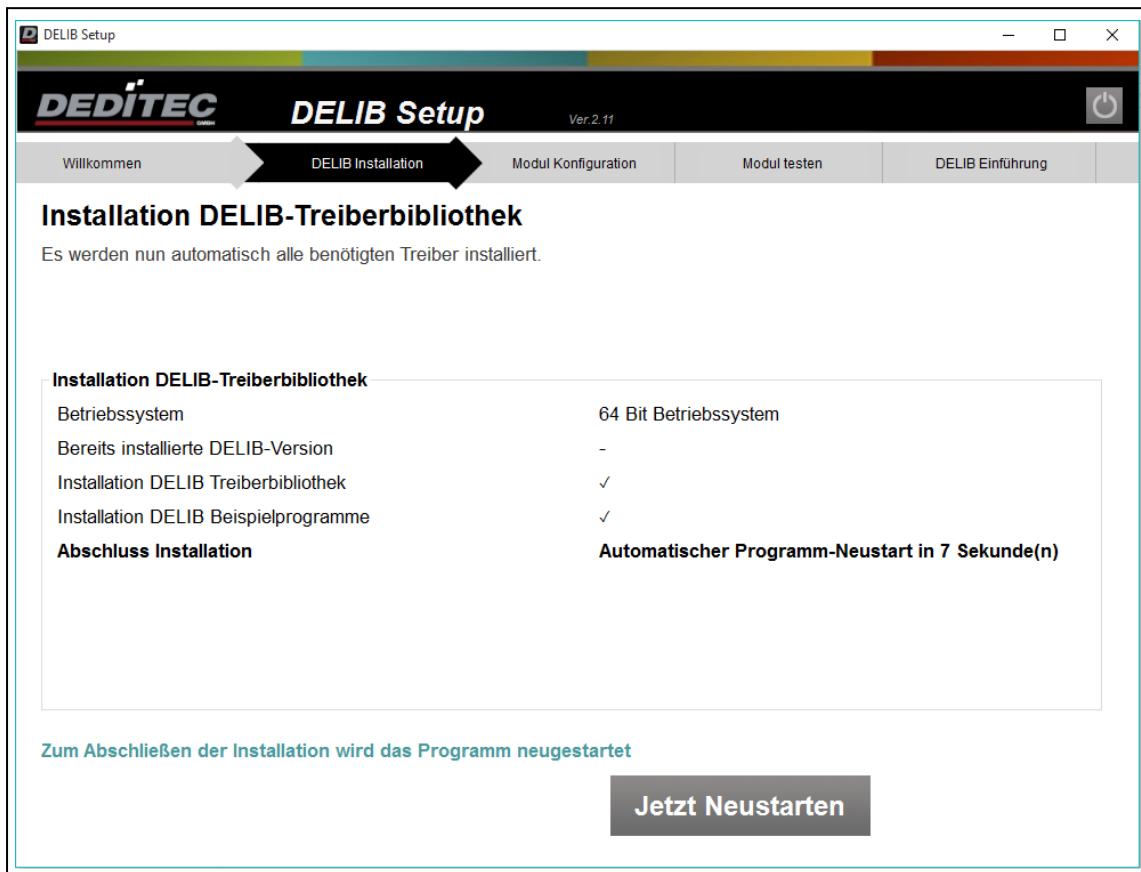


Installationsfortschritt der DELIB Beispielprogramme.





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



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

#### **3.2.3.1. Einführung**

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

Die Konfiguration ist für die erste Inbetriebnahme erforderlich.

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

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

#### **Standardpfad:**

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

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

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

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

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

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

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

### 3.2.3.2.1. Modul Konfiguration USB

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

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

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

The screenshot shows the 'DELIB Configuration Utility' window. The title bar includes the DEDITEC logo, the application name, version 'Ver. 2.19', and 'DELIB 2.19 (64 Bit)'. The main menu has five tabs: 'Modul Auswahl', 'Modul Konfiguration' (which is active), 'Konfiguration testen', 'Modul testen', and 'DELIB Einführung'. The active tab displays the title 'Konfiguration Module mit USB Schnittstelle' and a subtitle 'Hier kann die Modul-Nr. des ausgewählten Moduls geändert werden. Die Modul-Nr. dient zur Unterscheidung mehrerer identischer USB-Module einer USB-Serie.' Below this, there is a section 'Gefundene USB-Module' containing a table with two columns: 'Module' and 'Nr.'. The table lists 'USB-OPT/REL-8' with the number '0'. To the right of the table, there are input fields for 'Aktuelle Modul-Nr.' (set to 0), 'Neue Modul-Nr.' (a dropdown menu set to 1), and 'Übertragungsversuche (für alle USB-Module)' (a dropdown menu set to 5). A teal button labeled 'Neue Modul-Nr. setzen' is positioned between the 'Neue Modul-Nr.' and 'Übertragungsversuche' 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.

Neue Modul-Nr. setzen

Übertragungsversuche (für alle USB-Module)

Hauptmenü Test Fertig

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

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

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

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

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

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

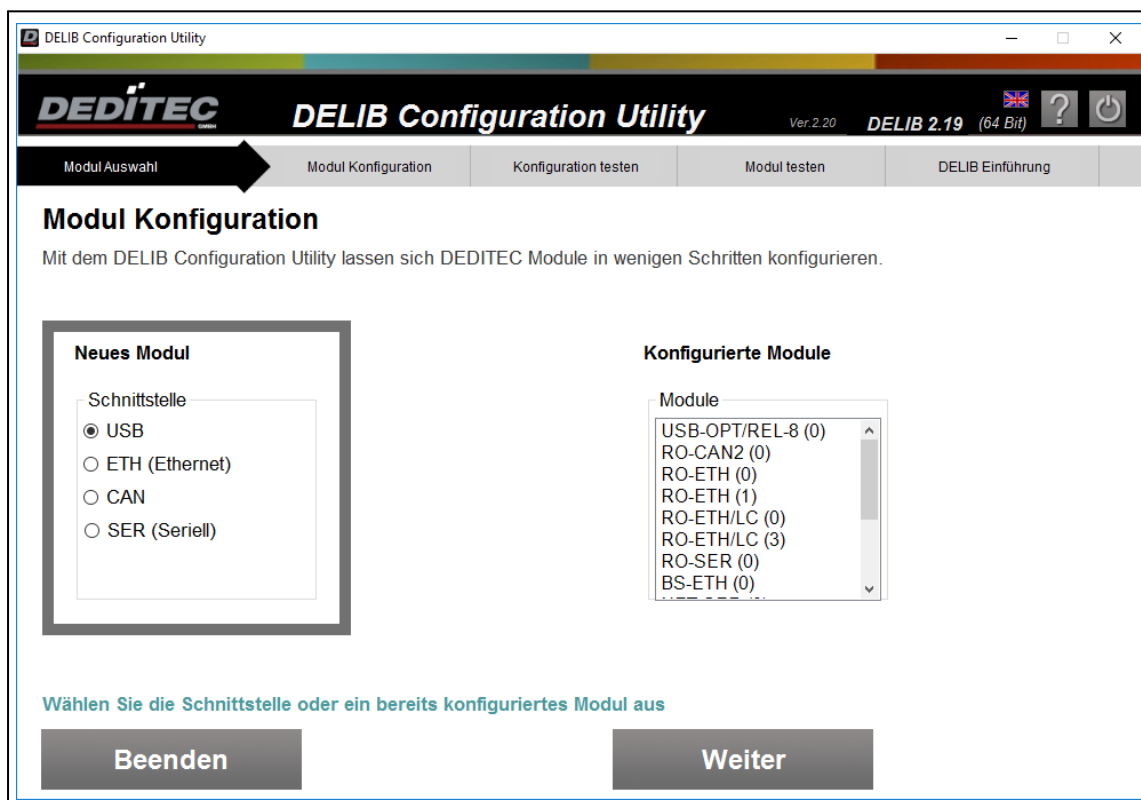
Befindet sich nur ein USB-Modul, oder mehrere USB-Module mit unterschiedlicher Modul-ID (z.B. RO-USB-016 und USB-RELAIS-8) im System, ist keine Konfiguration nötig, da die Produkte über die ID eindeutig identifizierbar sind.

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

#### Schritt 1

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

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

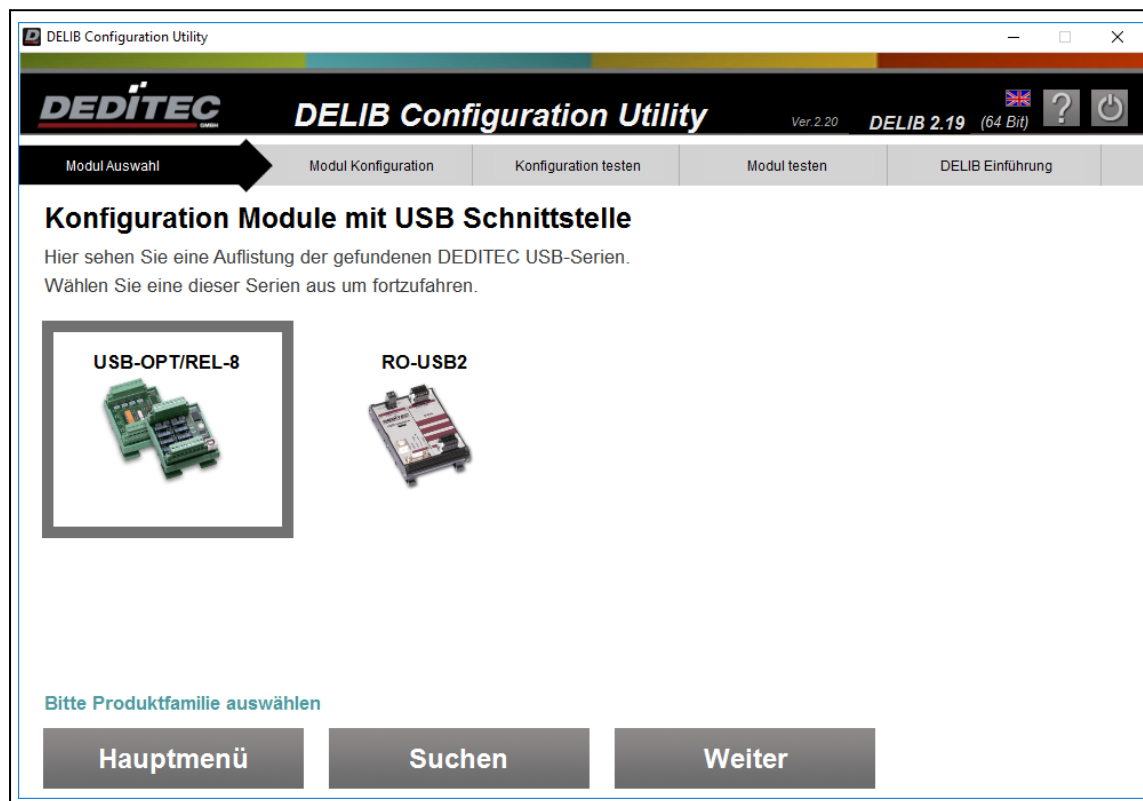


## Schritt 2

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

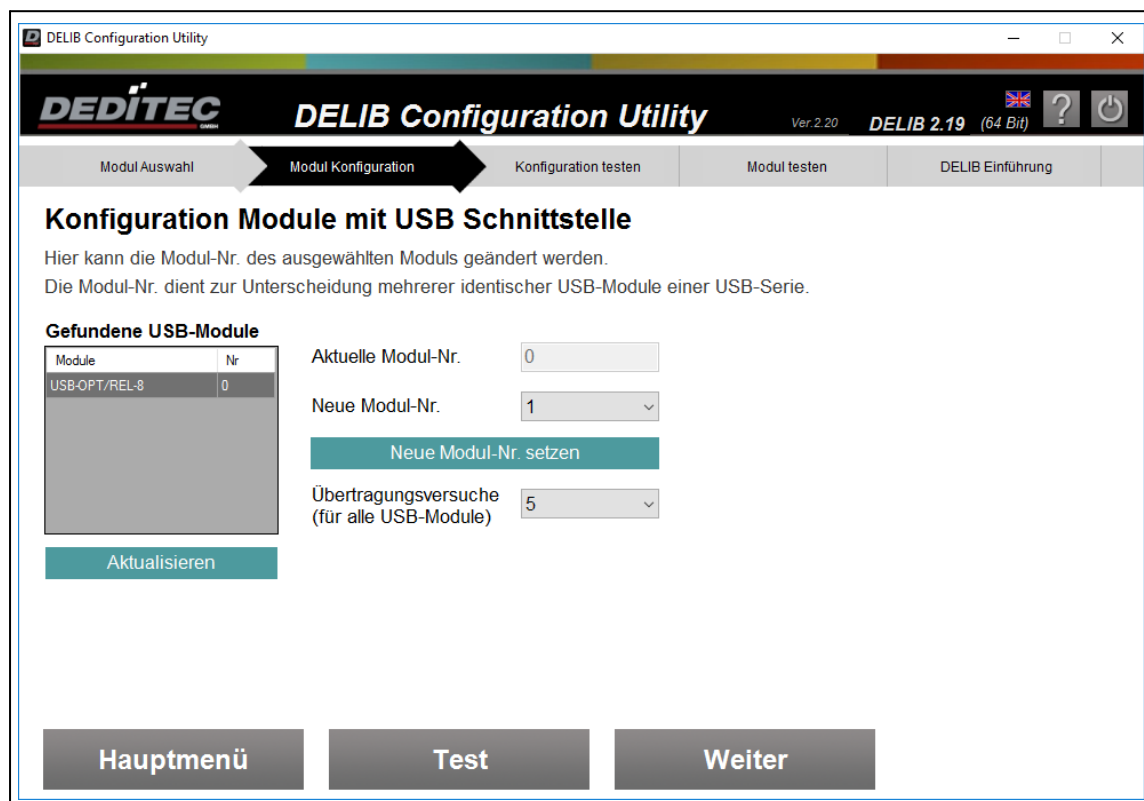
In diesem Beispiel sind Module der RO-USB2-Serie und USB-OPT/REL-8-Serie angeschlossen.

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



### Schritt 3

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



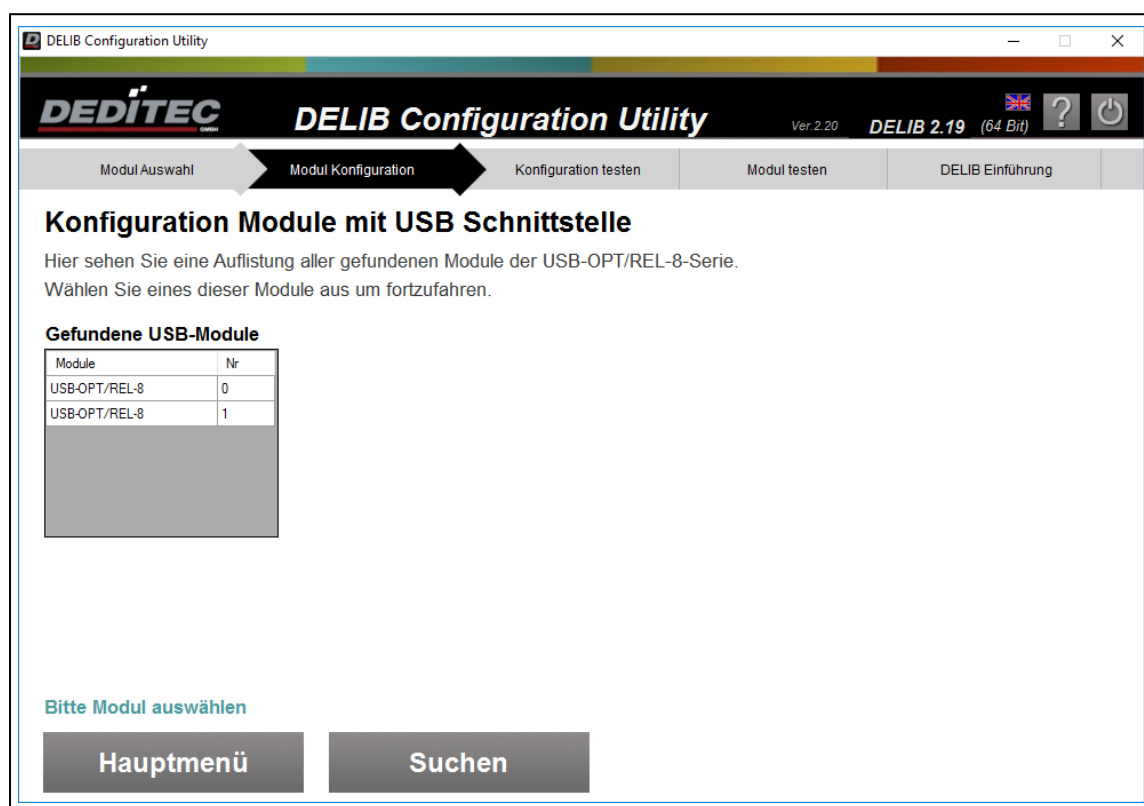


#### Schritt 4

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

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

Mit Aktualisieren werden nun beide Module angezeigt.



## Schritt 5

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

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

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

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

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

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

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

### Hinweis:

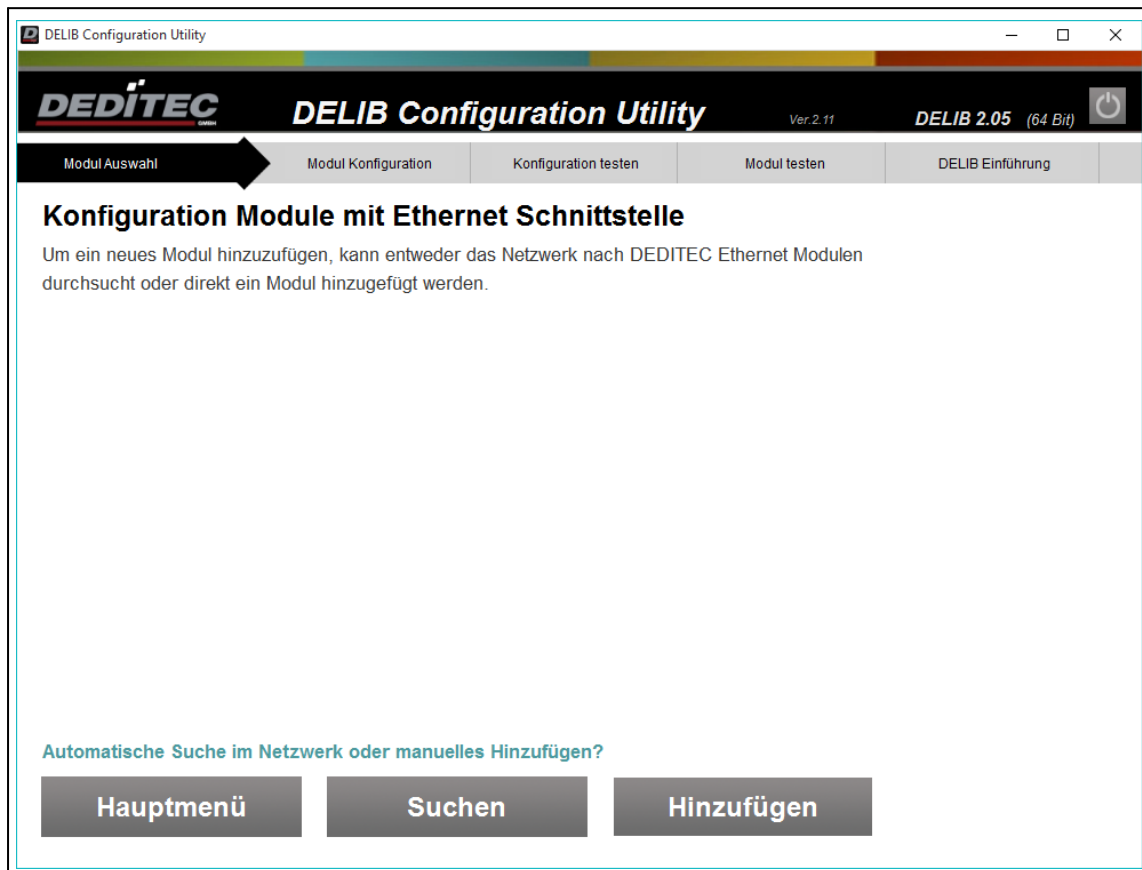
Alle Module haben als Werkseinstellung die NR "0".

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

### 3.2.3.2.2. Modul Konfiguration Ethernet

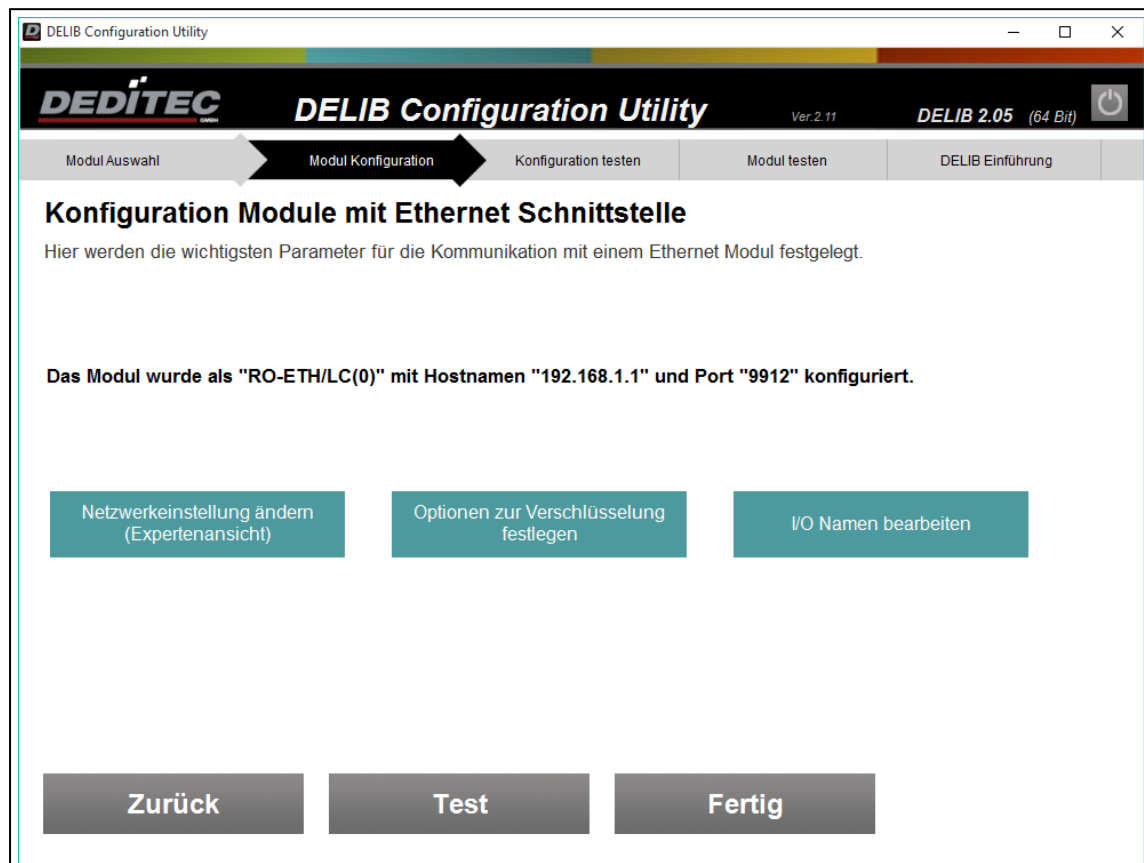
Bei der Konfiguration von Ethernet Produkten kann das Netzwerk automatisch nach DEDITEC Produkten durchsucht werden.

Auch eine manuelle Konfiguration kann durchgeführt werden.



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

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



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

### Deaktiviert

- unverschlüsselte Kommunikation
- kein Zugriff auf System-Einstellungen

### **User**

- verschlüsselte Kommunikation
- Lese-Zugriff auf System-Einstellungen

### **Admin**

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

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

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

siehe Kapitel → **Authentifizierung**

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

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

DELIB Configuration Utility

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

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

### Konfiguration Module mit Ethernet Schnittstelle

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

#### I/O Namen Konfiguration

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

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

Zurück Speichern

### Hinweis

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

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

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

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

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

#### **Achtung:**

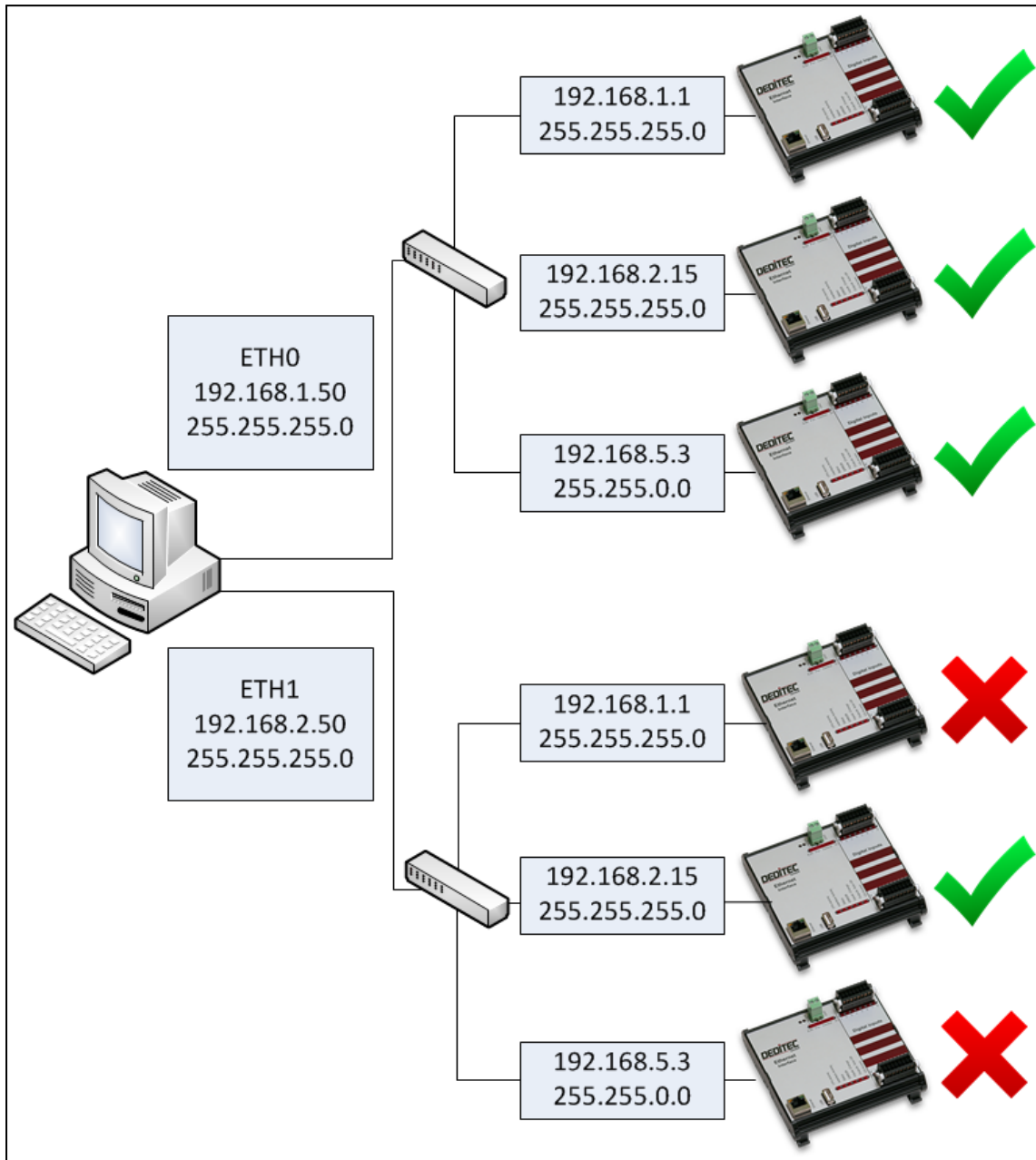
Wird ein Laptop verwendet, ist meistens der integrierte WLAN-Adapter als primärer Netzwerkadapter konfiguriert.

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

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

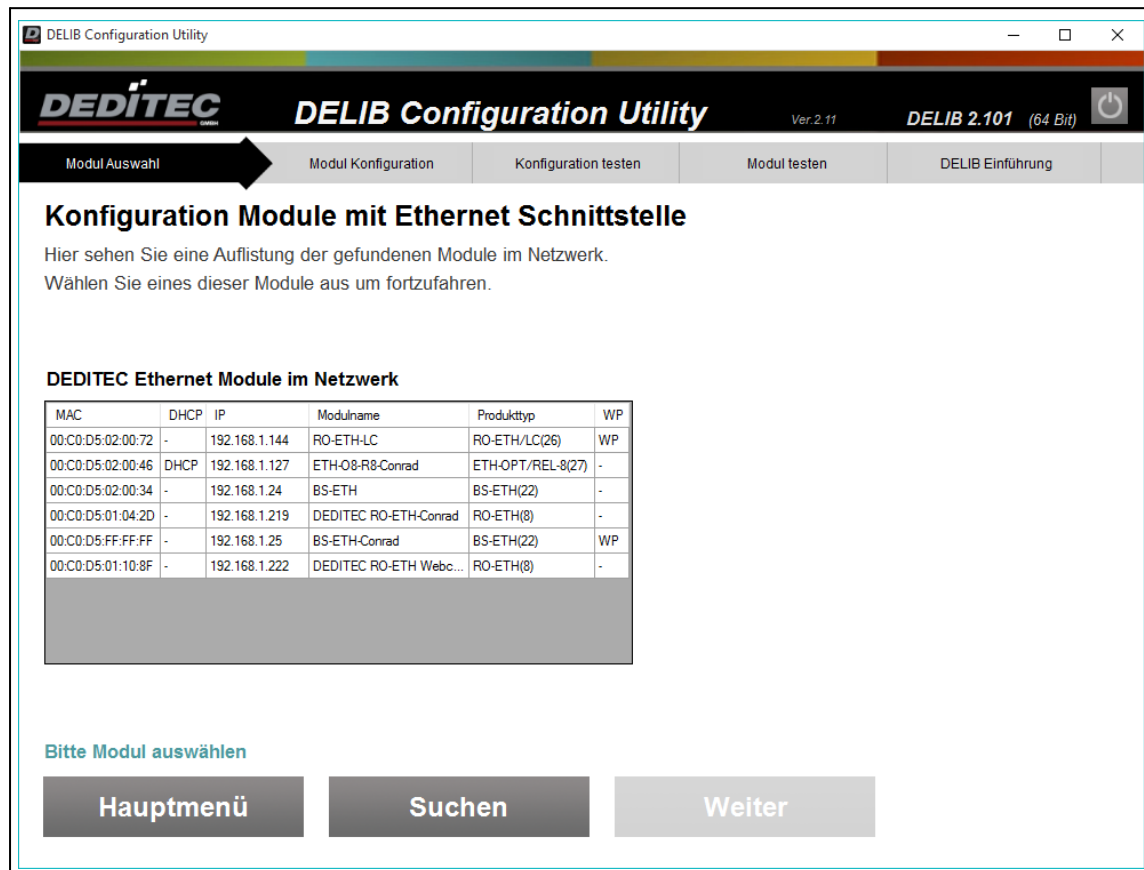


Folgende Grafik zeigt eine Übersicht der automatischen Suche.



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

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



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

siehe Kapitel → **Authentifizierung**

**Konfiguration Module mit Ethernet Schnittstelle**

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

**DEDITEC Ethernet Module im Netzwerk**

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

**Aktuelle Modul Konfiguration**

Board Name:

☐ IP-Adresse automatisch beziehen (DHCP)

IP Adresse:

Netzmaske:

Std.-Gateway:

## Hinweis

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

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

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

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

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

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

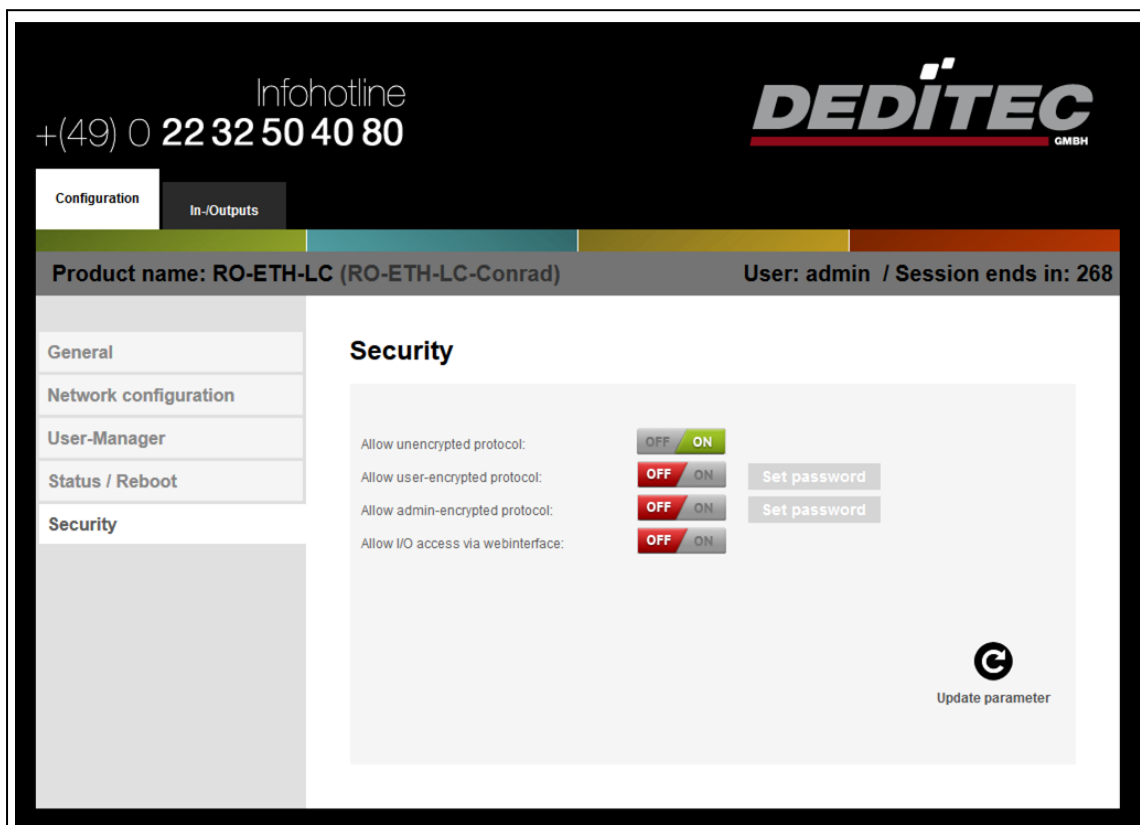
#### **Manuelle Konfiguration**

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

## Schritt 1 - Konfiguration Modul-Seite

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

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



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

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

## Hinweis

Bei der Eingabe des Passwortes sind alle Zeichen erlaubt.

The screenshot displays the DEDITEC web interface. At the top, there is an 'Infohotline' number: + (49) 0 22 32 50 40 80. The DEDITEC GMBH logo is in the top right corner. Below the header, there are two tabs: 'Configuration' (selected) and 'In-/Outputs'. A status bar shows 'Product name: RO-ETH-LC (RO-ETH-LC-Conrad)' and 'User: admin / Session ends in: 223'. On the left, a sidebar lists navigation options: 'General', 'Network configuration', 'User-Manager', 'Status / Reboot', and 'Security' (highlighted). The main content area is titled 'Security' and contains several settings:

- 'Allow unencrypted protocol:' with a toggle switch set to 'ON'.
- 'Allow user-encrypted protocol:' with a toggle switch set to 'OFF' and a 'Set password' button next to it.
- 'Allow admin-encrypted protocol:' with a toggle switch set to 'ON'.
- 'New password:' followed by a password input field with masked characters (dots).
- 'Confirm password:' followed by a password input field with masked characters (dots).
- 'Allow I/O access via webinterface:' with a toggle switch set to 'OFF'.

At the bottom right of the Security section, there is a circular icon with a 'C' and the text 'Update parameter'.

## Schritt 2 - Konfiguration PC-Seite

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

Hier bei Verschlüsselung "Admin" auswählen.

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

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

Das DELIB Configuration Utility kann nun geschlossen werden.

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



## Automatische Konfiguration

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

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

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

### Konfiguration Module mit Ethernet Schnittstelle

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

**Netzwerkkonfiguration**

Modul Auswahl: RO-ETH/LC(0)

☒ Aktiviert ☐ Hostname verwenden

IP Adresse: 192.168.1.25 Verschlüsselung: Admin

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

Timeout [msec]: 5000

I/O Namen bearbeiten

Einstellungen zur Verschlüsselung auf das Modul übertragen

Hauptmenü Test Fertig

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

## Hinweis

Bei der Eingabe des Passwortes sind alle Zeichen erlaubt.

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

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

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

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

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

**DELIB Configuration Utility**

Ver. 2.19 DELIB 2.19 (64 Bit)

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

### Konfiguration Module mit Ethernet Schnittstelle

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

**Netzwerkconfiguration**

Modul Auswahl: RO-ETH/LC(0)

☒ Aktiviert ☐ Hostname verwenden

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

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

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

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

Authentifizierung abbrechen

Hauptmenü Test Fertig

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

The screenshot shows the 'DELIB Configuration Utility' window. The title bar reads 'DELIB Configuration Utility'. The header features the 'DEDITEC' logo, the product name 'DELIB Configuration Utility', the version 'Ver. 2.19', and 'DELIB 2.19 (64 Bit)' with a power icon. A progress bar at the top has five steps: 'Modul Auswahl', 'Modul Konfiguration' (highlighted with a black arrow), 'Konfiguration testen', 'Modul testen', and 'DELIB Einführung'. Below the progress bar, the main heading is 'Konfiguration Module mit Ethernet Schnittstelle'. A sub-header reads: 'Hier werden die wichtigsten Parameter für die Kommunikation mit einem Ethernet Modul festgelegt.' To the right is a 'Manual' icon. The 'Netzwerkconfiguration' section contains the following fields and controls:

- 'Modul Auswahl' dropdown menu set to 'RO-ETH/LC(0)'.
- Checkboxes for 'Aktiviert' (checked) and 'Hostname verwenden' (unchecked).
- 'IP Adresse' text box with '192.168.1.25'.
- 'Verschlüsselung' dropdown menu set to 'Admin'.
- 'Port' text box with '9912'.
- 'Passwort' text box with masked characters '\*\*\*\*\*'.
- 'Timeout [msec]' text box with '5000'.
- 'I/O Namen bearbeiten' button.
- 'Einstellungen zur Verschlüsselung auf das Modul übertragen' button.

At the bottom, a teal instruction bar states: 'Schritt 2: DIP-Schalter 2 (Schreibschutz) des ETH-Moduls auf Stellung 0 (0=OFF, 1=ON) setzen'. To its right is a teal button labeled 'Authentifizierung abbrechen'. Below the instruction bar are three buttons: 'Hauptmenü', 'Test', and 'Fertig'. At the bottom right are three small square status indicators.

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

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

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

The screenshot shows the 'DELIB Configuration Utility' window. The title bar includes the DEDITEC logo, the application name, version 'Ver. 2.19', and 'DELIB 2.19 (64 Bit)'. The main menu has five tabs: 'Modul Auswahl', 'Modul Konfiguration' (active), 'Konfiguration testen', 'Modul testen', and 'DELIB Einführung'. The active tab displays the title 'Konfiguration Module mit Ethernet Schnittstelle' and a subtitle 'Hier werden die wichtigsten Parameter für die Kommunikation mit einem Ethernet Modul festgelegt.' Below this is the 'Netzwerkconfiguration' section with the following fields: 'Modul Auswahl' (dropdown menu showing 'RO-ETH/LC(0)'), 'Aktiviert' (checked checkbox), 'Hostname verwenden' (unchecked checkbox), 'IP Adresse' (text field with '192.168.1.25'), 'Verschlüsselung' (dropdown menu showing 'Admin'), 'Port' (text field with '9912'), 'Passwort' (password field with '\*\*\*\*\*'), and 'Timeout [msec]' (text field with '5000'). There are two teal buttons: 'I/O Namen bearbeiten' and 'Einstellungen zur Verschlüsselung auf das Modul übertragen'. A green status message at the bottom reads 'Einstellungen zur Verschlüsselung erfolgreich übertragen'. At the very bottom are three dark grey buttons: 'Hauptmenü', 'Test', and 'Fertig'.

DELIB Configuration Utility

Modul Auswahl Modul Konfiguration Konfiguration testen Modul testen DELIB Einführung

### Konfiguration Module mit Ethernet Schnittstelle

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

**Netzwerkconfiguration**

Modul Auswahl: RO-ETH/LC(0)

☒ Aktiviert ☐ Hostname verwenden

IP Adresse: 192.168.1.25 Verschlüsselung: Admin

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

Timeout [msec]: 5000

I/O Namen bearbeiten

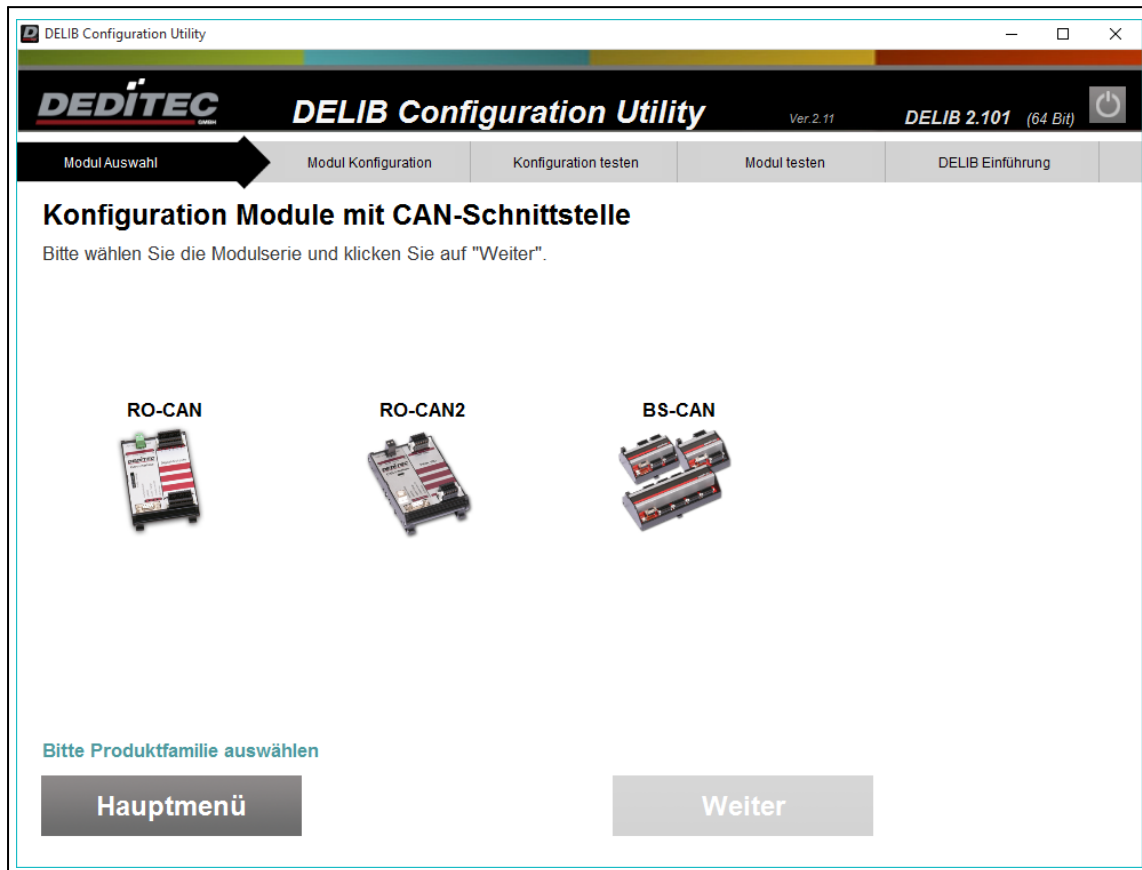
Einstellungen zur Verschlüsselung auf das Modul übertragen

Einstellungen zur Verschlüsselung erfolgreich übertragen

Hauptmenü Test Fertig

### 3.2.3.2.3. Modul Konfiguration CAN

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



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

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

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

The screenshot shows the 'DELIB Configuration Utility' window. The title bar reads 'DELIB Configuration Utility'. The interface has a top header with the 'DEDITEC' logo, the title 'DELIB Configuration Utility', version 'Ver. 2.11', and 'DELIB 2.101 (64 Bit)' with a power icon. Below the header is a navigation bar with five steps: '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 CAN-Schnittstelle' and contains the text: 'Die Konfiguration der CAN-Schnittstelle erfolgt seriell. Hierzu muss ein COM-Port ausgewählt werden. Zusätzlich kann die Anzahl der maximalen Übertragungsversuche im Falle eines Kommunikationsfehlers festgelegt werden.' Below this text is a section 'Modul Konfiguration' with three dropdown menus: 'Modul Auswahl' set to 'RO-CAN(0)', 'COM Port' set to 'COM1', and 'Übertragungsversuche' set to '5'. At the bottom are three buttons: 'Zurück', 'Test', and 'Fertig'.

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

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

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

The screenshot shows the 'DELIB Configuration Utility' window. The title bar includes the DEDITEC logo, the application name, version 'Ver. 2.11', and 'DELIB 2.101 (64 Bit)'. The main menu has five tabs: 'Modul Auswahl', 'Modul Konfiguration' (which is active), 'Konfiguration testen', 'Modul testen', and 'DELIB Einführung'. The active tab displays the title 'Konfiguration Module mit USB Schnittstelle' and instructions: 'Hier kann die Modul-Nr. des ausgewählten Moduls geändert werden. Die Modul-Nr. dient zur Unterscheidung mehrerer identischer USB-Module einer USB-Serie. Zusätzlich kann die Anzahl der maximalen Übertragungsversuche im Falle eines Kommunikationsfehlers festgelegt werden.'

Below the instructions is a table titled 'Gefundene USB-Module':

| Module  | Nr |
|---------|----|
| RO-CAN2 | 1  |

To the right of the table are three input fields:

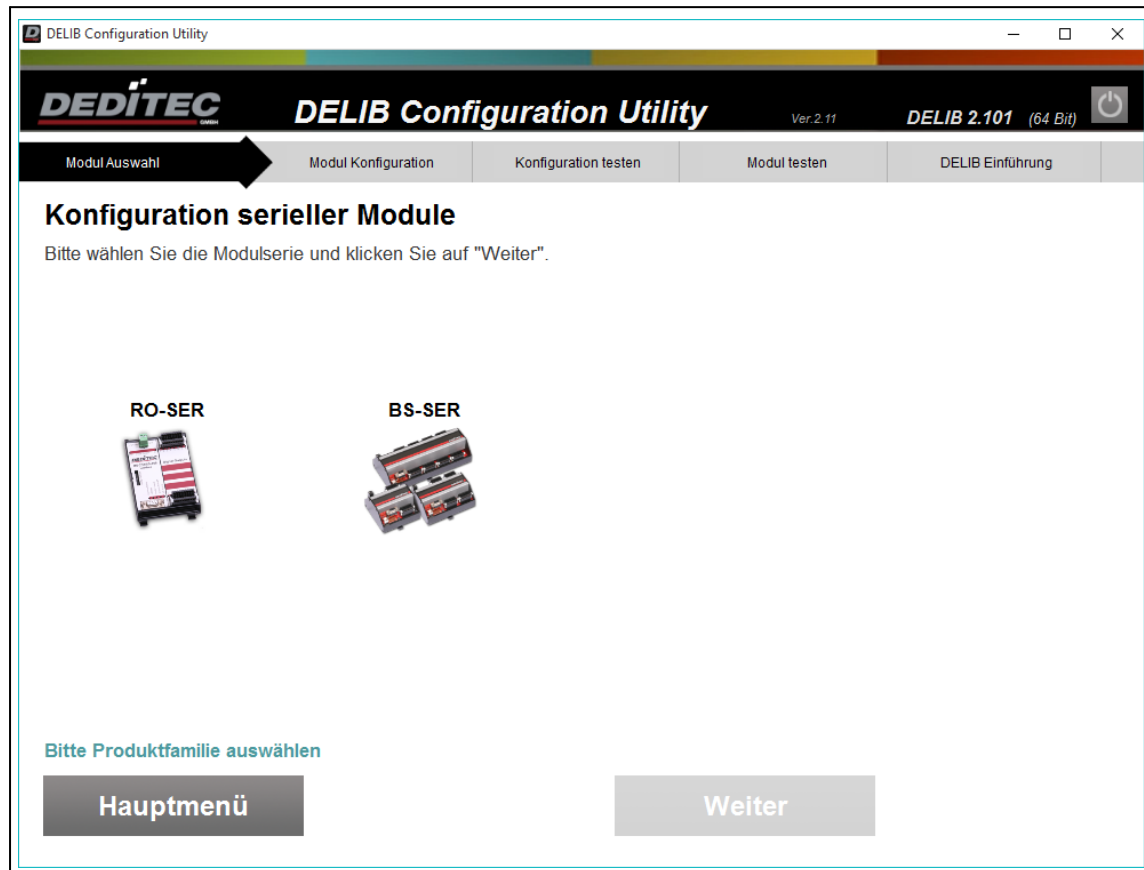
- 'Aktuelle Modul-Nr.' with a text input containing '1'.
- 'Neue Module-Nr.' with a dropdown menu showing '0'.
- 'Übertragungsversuche (für alle USB-Module)' with a dropdown menu showing '5'.

At the bottom, there is a blue link: 'Klicken Sie "Weiter" zum Übernehmen der neuen Modul-Nr.' and three buttons: 'Hauptmenü', 'Suchen', and 'Weiter'.



#### 3.2.3.2.4. Modul Konfiguration Seriell

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



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

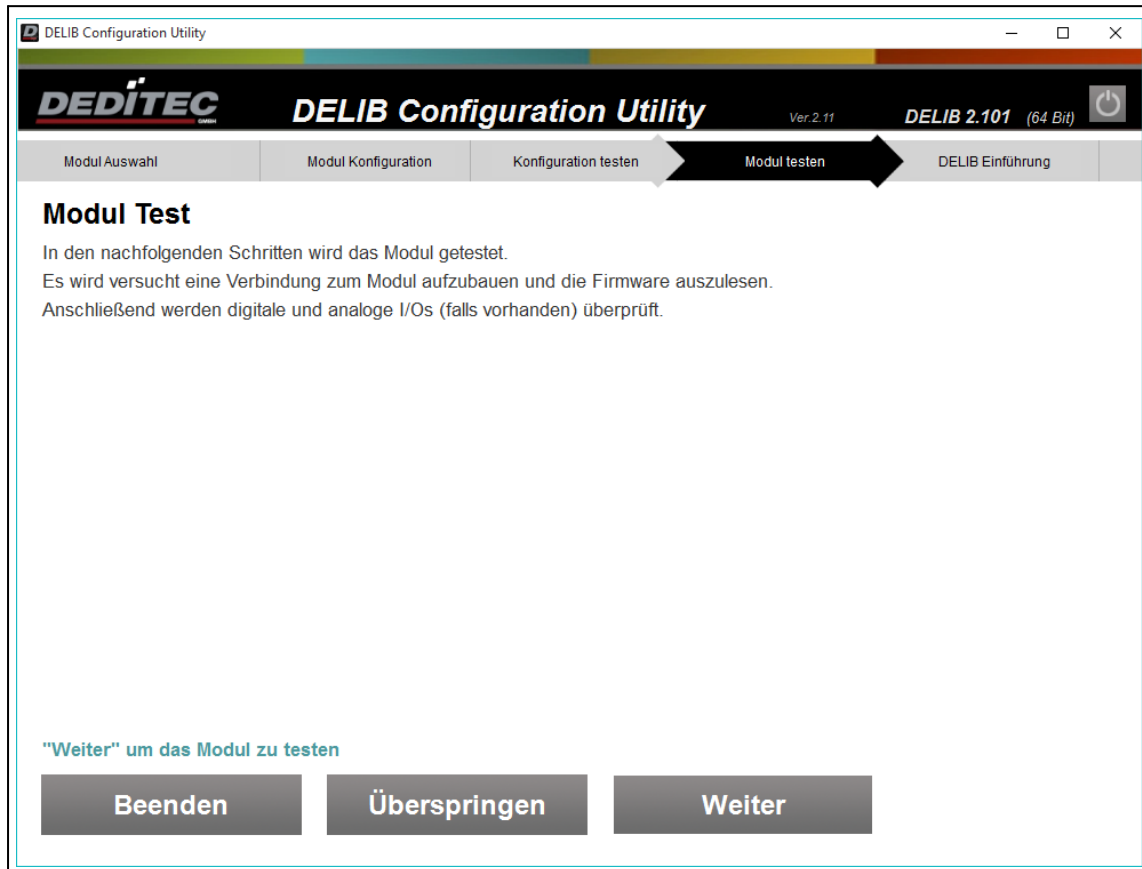
Zusätzlich muss die Anzahl der Übertragungsversuche im Falle eines Kommunikationsfehlers angegeben werden.

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

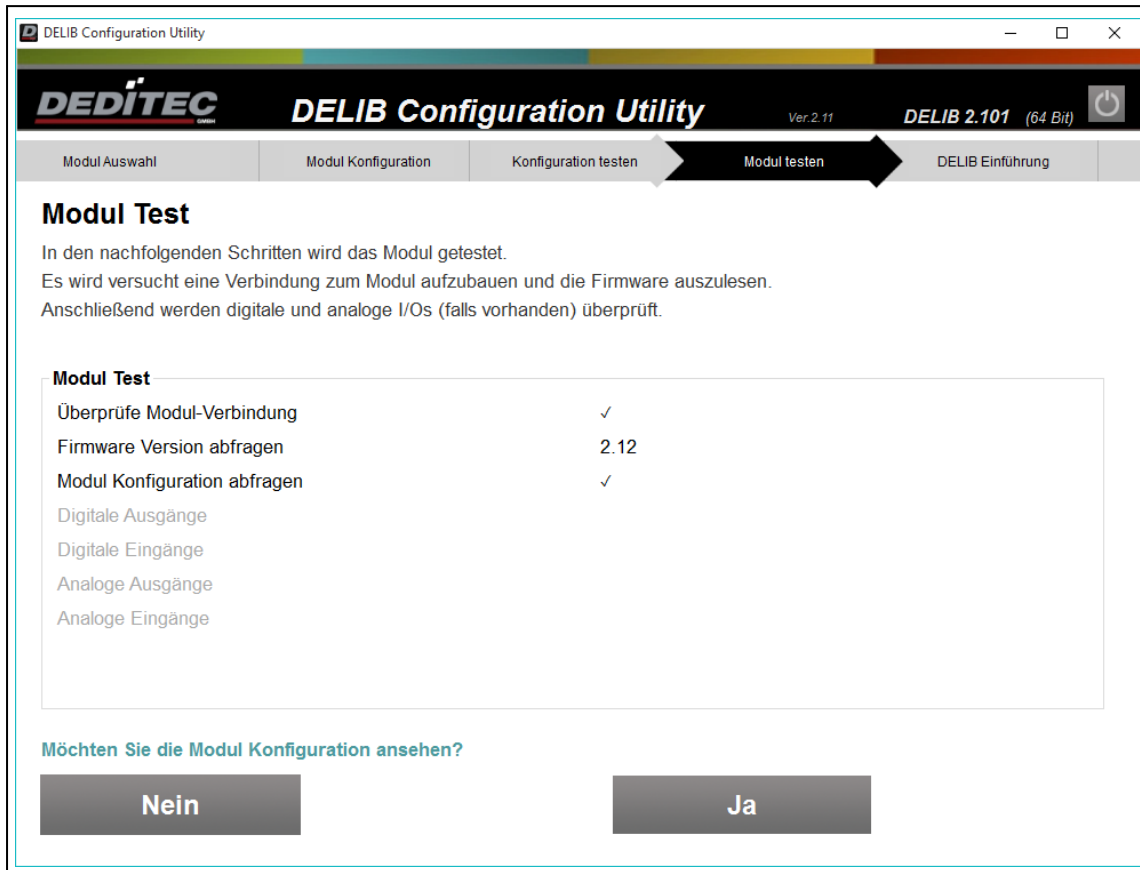
The screenshot shows the 'DELIB Configuration Utility' window. The title bar includes the DEDITEC logo, the application name 'DELIB Configuration Utility', the version 'Ver.2.11', and the device model 'DELIB 2.101 (64 Bit)'. The main menu has five tabs: 'Modul Auswahl', 'Modul Konfiguration' (which is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

### 3.2.3.3. Modul testen

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



Test der Firmware und Anzeige der Modul-Info.



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

DT\_ModuleInfo

×

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

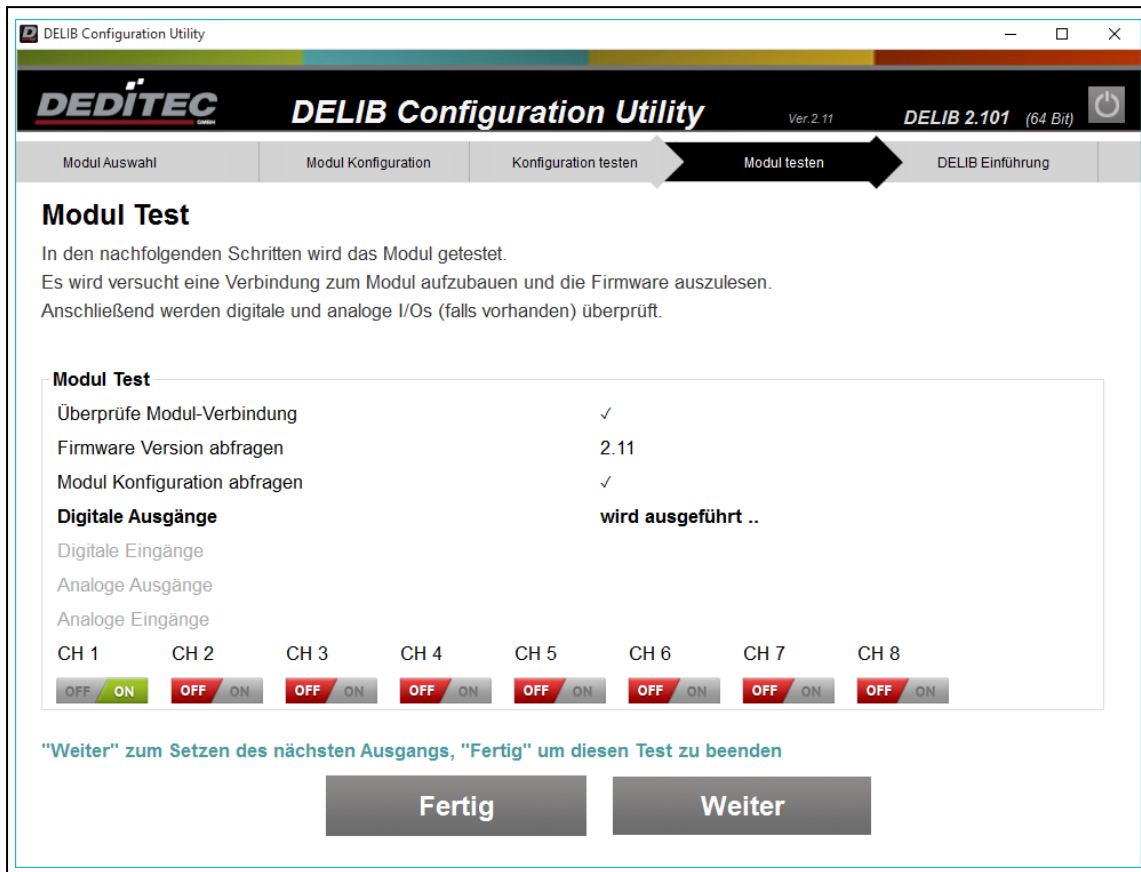
| Features-General          | Features-Digital I/O       | Features-Analog I/O | Features-Special    |
|---------------------------|----------------------------|---------------------|---------------------|
| Supported by FW OK        | DI Commands OK             | DA Commands OK      | Watchdog Commands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

Submodule-Info's

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

EXIT

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

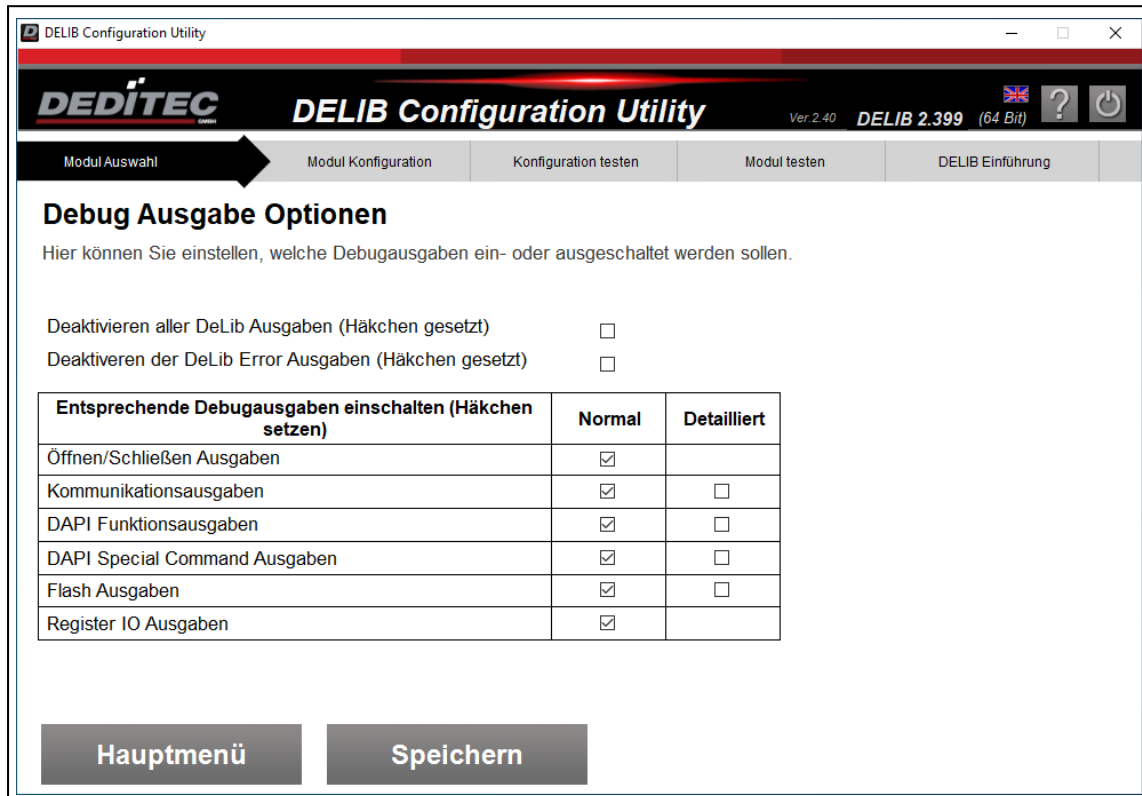


Wurden alle Tests erfolgreich durchlaufen, ist das Produkt einsatzbereit.

### 3.2.3.4. Debug Optionen einstellen

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

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



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

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

### 3.2.4. Benutzung des Modulselectors

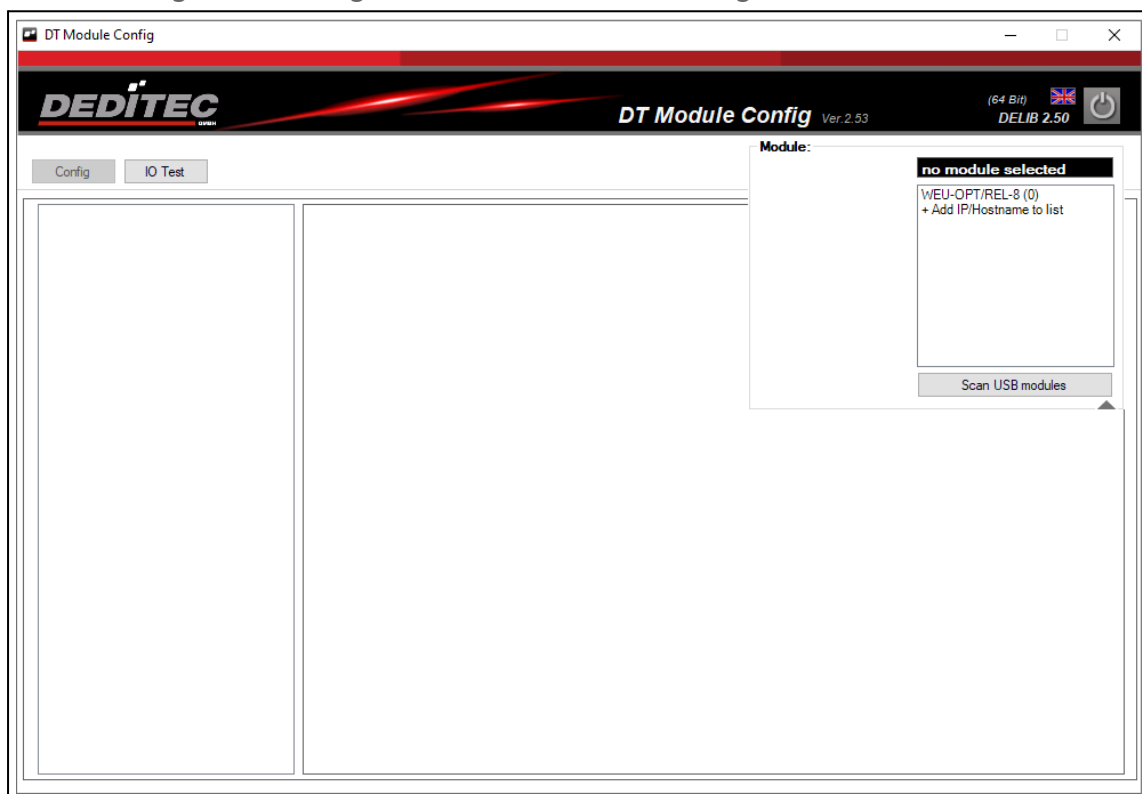
Um unsere Produkte mit der DEDITEC-Software benutzen zu können, müssen diese über den Modul Selector ausgewählt werden.

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

#### 3.2.4.1. via USB

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

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



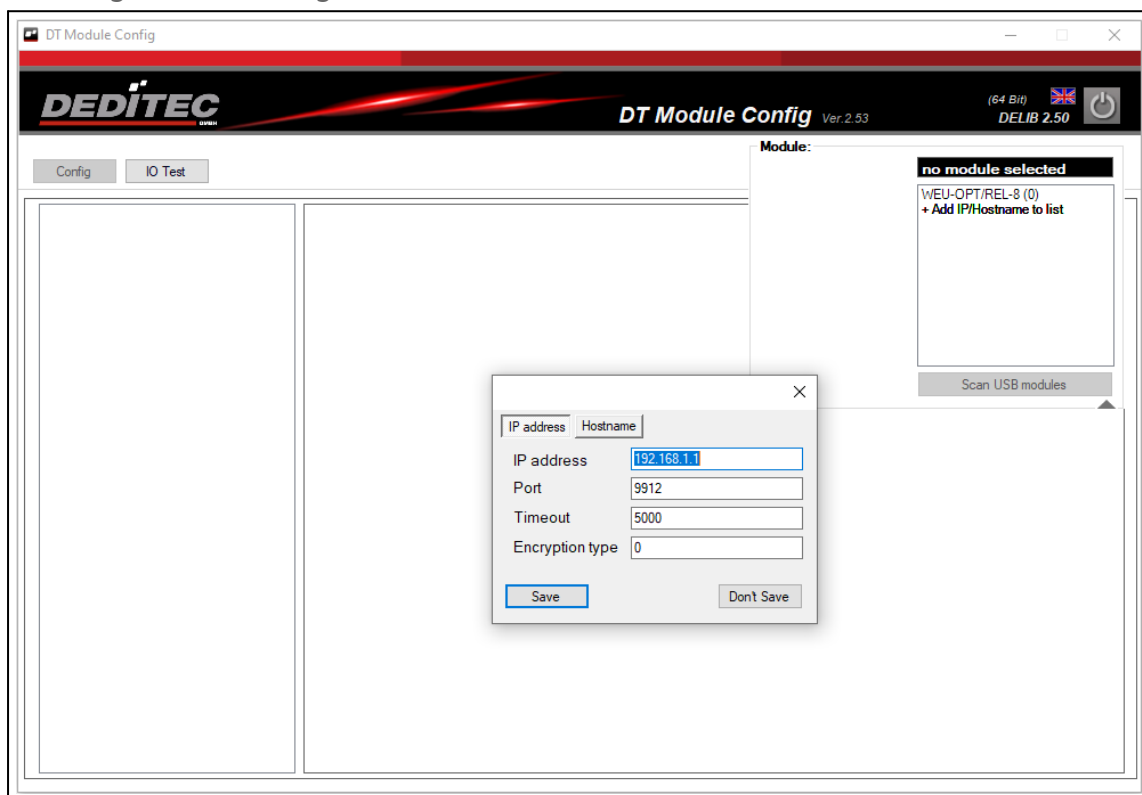


### 3.2.4.2. via Ethernet

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.

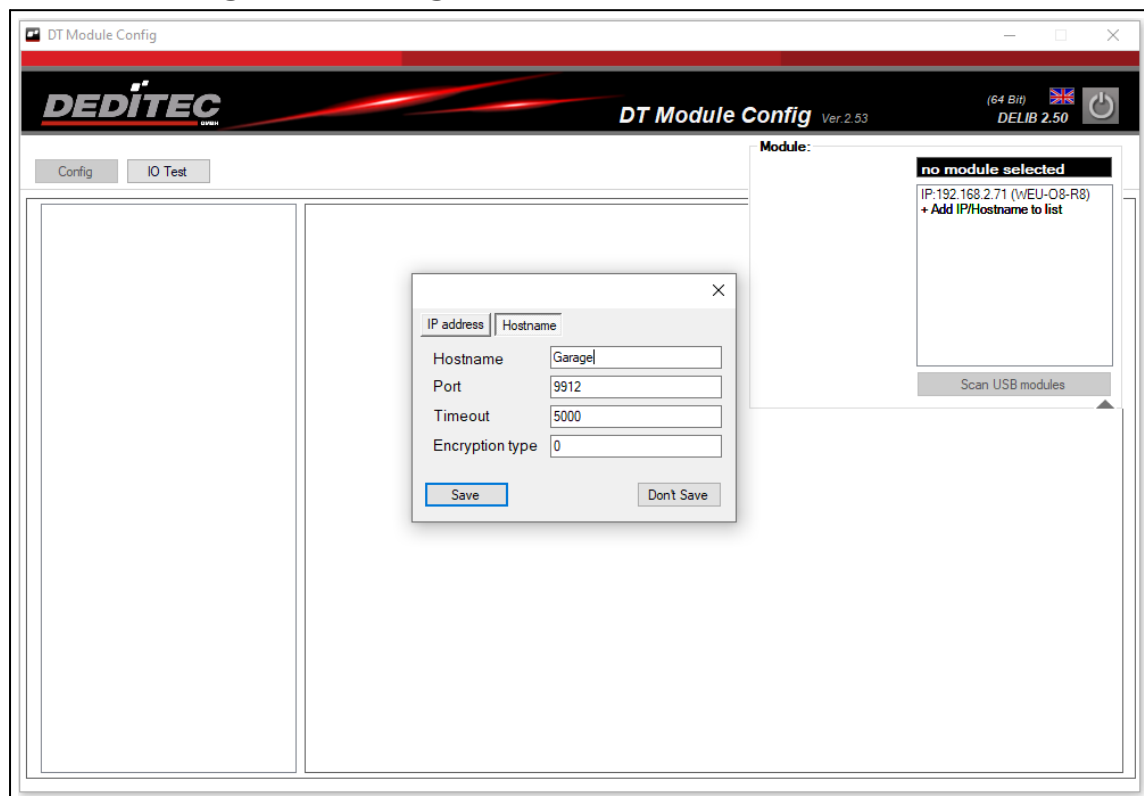
Klicken Sie im Modul Selector auf "Add IP/Hostname to list", tragen Sie dort die automatisch erhaltene IP-Adresse unter dem Reiter "IP Adresse" ein und bestätigen Sie die Eingabe mit "Save".



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

Diesen finden Sie im 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".



### **3.2.4.3. via WiFi / WPS**

#### **WiFi (nur bei WEU-Modulen)**

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

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

#### **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.

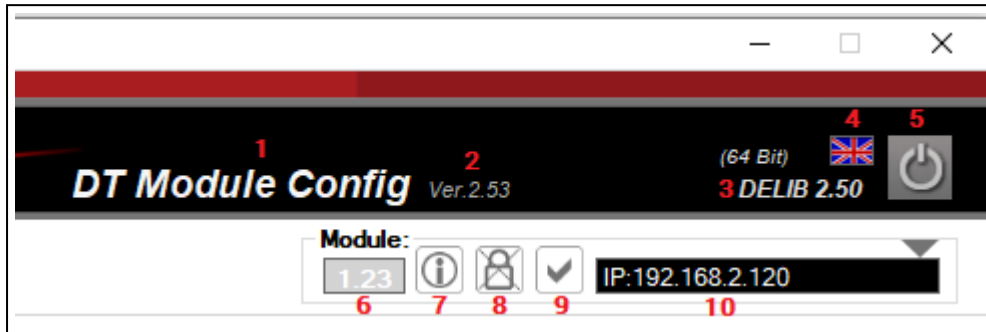
Klicken Sie danach im Modul Selector auf "Add IP/Hostname to list", tragen Sie dort die automatisch erhaltene IP-Adresse unter dem Reiter "IP Adresse" ein und bestätigen Sie die Eingabe mit "Save".

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.

#### 3.2.4.4. Modul Info

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



#### Beschreibung:

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

## Informationsfenster

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

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

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

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

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

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

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

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

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

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

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

### 3.2.5.1.1. Modul-Infoseite

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

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

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

Form für Module Overview (Ver. 1.01)

#### Modul-Name

Zeigt den Namen des aktuell verwendeten DEDITEC Modules an.

#### Modul-ID

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

#### Firmware-Revision

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

### 3.2.5.1.2. Modul-Identifikation

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

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

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

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

Dieser Vorgang wird durch drücken von "Stop" beendet.

### Identifikation

Hier können Sie Ihr Modul identifizieren

Identify module

**Modulidentifikation läuft!**

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

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

Das Bedienen der 'Stop' Taste beendet diesen Vorgang.

FormConfModulIdent (Ver. 1.01)



### 3.2.5.1.3. LAN Netzwerkinformationen

Alle wichtigen LAN Netzwerkinformationen auf einen Blick.

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

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

|                                    |                          |
|------------------------------------|--------------------------|
| MAC address                        | 40:F5:20:44:60:EB        |
| Board name (Hostname in DHCP mode) | WEU-08-R8                |
| LAN status                         | Static-IP success        |
| DHCP active                        | <input type="checkbox"/> |
| IP address                         | 192.168.2.71             |
| Subnet mask                        | 255.255.255.0            |
| Default gateway                    | 192.168.2.254            |
| TCP port                           | 9912                     |

☐ Auto Refresh

FormOfnNetwork LANInfo (Ver. 1.01)

#### MAC-Adresse

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

#### Board Name

Zeigt den aktuellen Board Name Ihres Modules an.

#### LAN-Status

Hier wird der Verbindungsstatus Ihres angeschlossenen Moduls angezeigt.

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

#### DHCP active

Zeigt an, ob das Modul über DHCP verbunden ist.

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


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

#### 3.2.5.1.4. LAN Netzwerkeinstellungen

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

### Konfiguration

Hier können Sie die Netzwerkeinstellungen des ausgewählten Ethernet-Produktes ändern

|   |  |
|---|--|
| Board name (Hostname in DHCP mode)  | <input type="text" value="WEU-O8-R8"/>     |
| Network configuration protection active  | <input type="checkbox"/>                   |
| DHCP active   | <input type="checkbox"/>                   |
| IP address  | <input type="text" value="192.168.2.71"/>  |
| Subnet mask   | <input type="text" value="255.255.255.0"/> |
| Default gateway   | <input type="text" value="192.168.2.254"/> |
| TCP port  | <input type="text" value="9912"/>          |

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

#### Board Name

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

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

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

Mehr Infos zum Anbinden des Moduls per Board Name

siehe Kapitel: **Benutzung des Modulselectors**

### Network configuration protection active

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

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

### DHCP active

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

Der Board Name wird als Hostname verwendet.

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

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

### Werkseinstellungen laden

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

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

### 3.2.5.1.5. WiFi Netzwerkinformationen

Alle wichtigen WiFi Netzwerkinformationen auf einen Blick.

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

### Info

Hier finden Sie die Netzwerkinformationen des ausgewählten WiFi-Produktes

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

☒ Auto Refresh

FormTitleNetworkWiFiInfo (Ver. 1.01)

### MAC-Adresse

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

### Board Name

Zeigt den aktuellen Board Name Ihres Moduls an.

### **WLAN-Status**

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

### **WLAN active**

Zeigt an, ob das Modul über WLAN verbunden ist.

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

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

### **Router name**

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

### **Password**


Zeigt das verwendete Routerpasswort an.

### 3.2.5.1.6. WiFi Netzwerkeinstellungen

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

## Konfiguration

Hier können Sie die Netzwerkeinstellungen des ausgewählten WiFi-Produktes ändern

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

Form für Netzwerk-WiFi Konfig (Ver. 1.01)

#### Board name

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

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

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

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

### **WLAN active**

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

### **Router name**

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

Fragen Sie hierfür gegebenenfalls Ihren Systemadministrator.

### **Password**

Hier können Sie das Routerpasswort des verwendeten Routers eingetragen werden.

Fragen Sie hierfür gegebenenfalls Ihren Systemadministrator.

### **TCP port**

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

### **Werkseinstellungen laden**

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

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

### 3.2.5.1.7. WiFi WPS-Verbindung

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

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

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

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

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

## WPS

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

WLAN status

starting WPS connection..

Router name

Password

WPS-Start

WPS-Stop

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

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

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

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

☒ Auto Refresh

FormCfgNetworkWiFiWPS (Ver. 1.01)



### 3.2.5.1.8. NTP-Konfiguration

Hier können Änderungen am NTP-Service vorgenommen werden.

#### NTP-Konfiguration

Hier können Sie Änderungen am NTP Service vornehmen

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

Werkseinstellung laden

Einstellung speichern

Formf'nNetwork NTP (Ver. 1.01)

#### NTP service active

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

#### Server

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

#### Port

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

## Timezone

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

## Werkseinstellungen laden

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

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

### 3.2.5.1.9. Serielle Konfiguration

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

#### Seriell

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

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

FormC3aSenell (Ver 1.01)

### Vorzugsmodus (Special mode)

Im Vorzugsmodus wird das Modul automatisch mit folgenden Einstellungen betrieben:

Baud rate: 115200

Modul-Nr. 0

Echo = Off

Register-Mode = On

### **Baud rate**

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

625000

250000

125000

115200

57600

50000

38400

19200

9600

4800

2400

1200

600

300

### **RS485 Modul-adresse**

Adresse für die Identifikation im RS485 Bus.

### **Echo**

Seriell empfangene Zeichen werden vom Modul zurückgesendet.

### **Registernodus**

Deaktivieren Sie den Registernodus um den Textmodus zu aktivieren.

### 3.2.5.1.10. I/O Kanal-Namen

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

#### I/O Kanal-Namen

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

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

Set default channel name

Einstellung speichern

FormCfnSubmoduleIOnames (Ver. 1.01)

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

#### Hinweis:

Der Kanalname darf maximal 16 Zeichen lang sein.

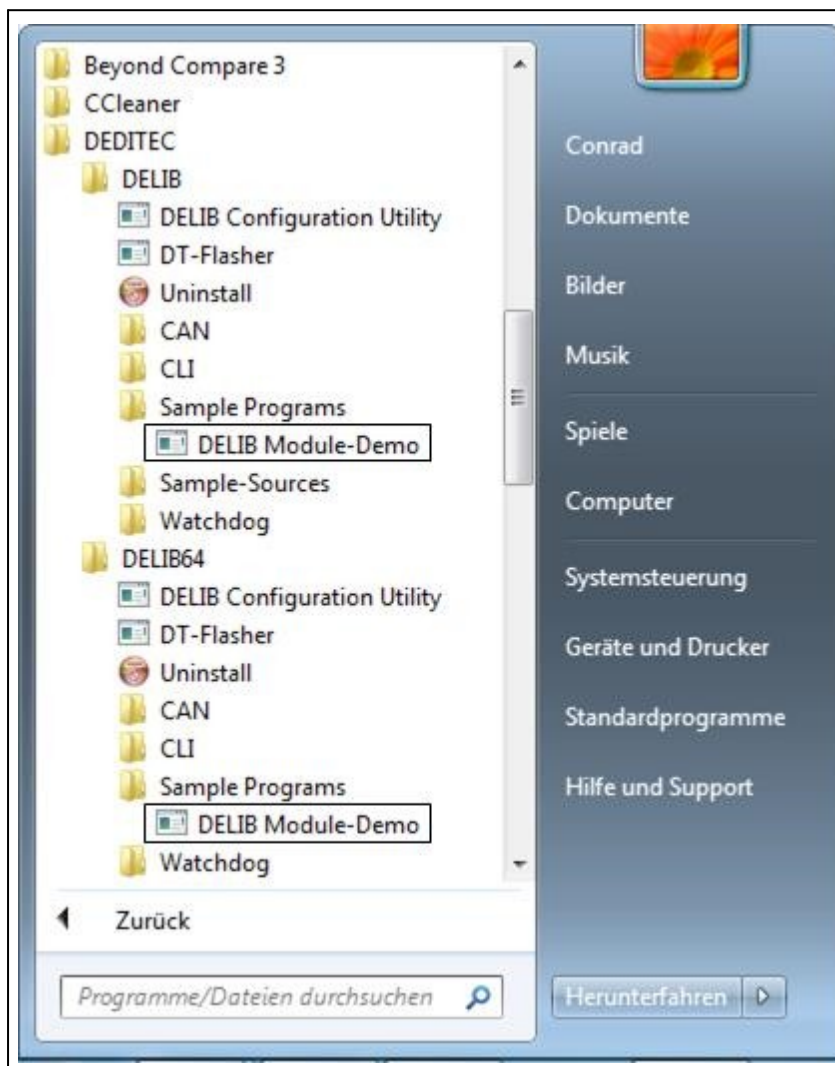
#### Set default channel name

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

### 3.2.6. DELIB Module Demo

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

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

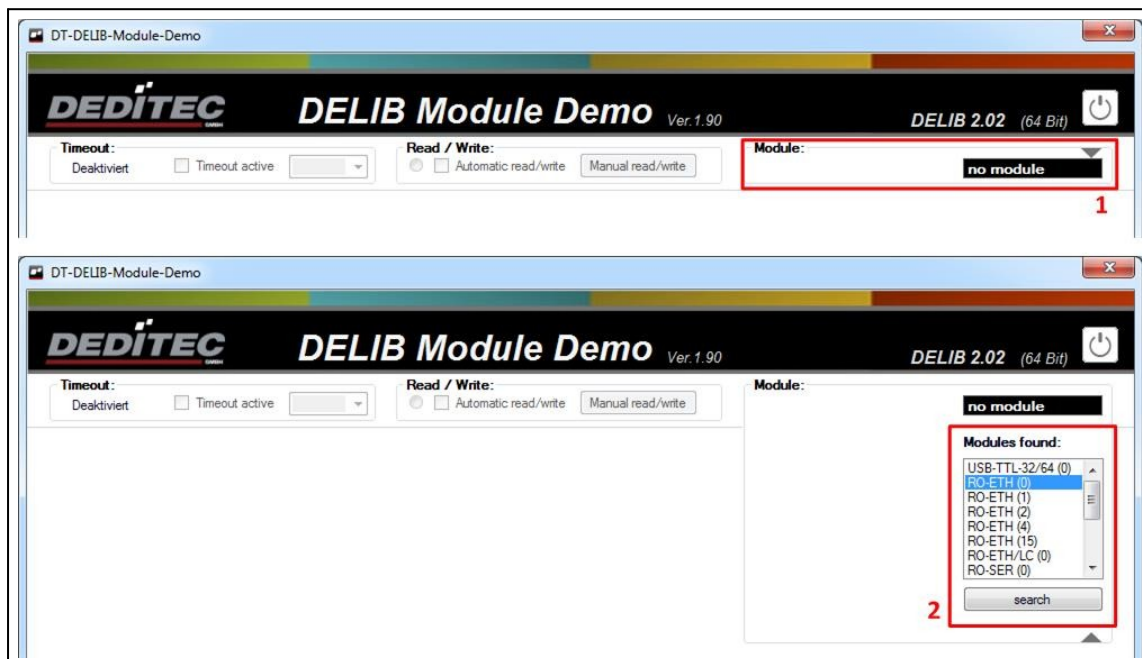


Das Programm DELIB Module Demo ist ein All-in-One Tool mit dem sämtliche I/Os aller Produkte aus unserem S&R Bereich gesteuert und getestet werden können.

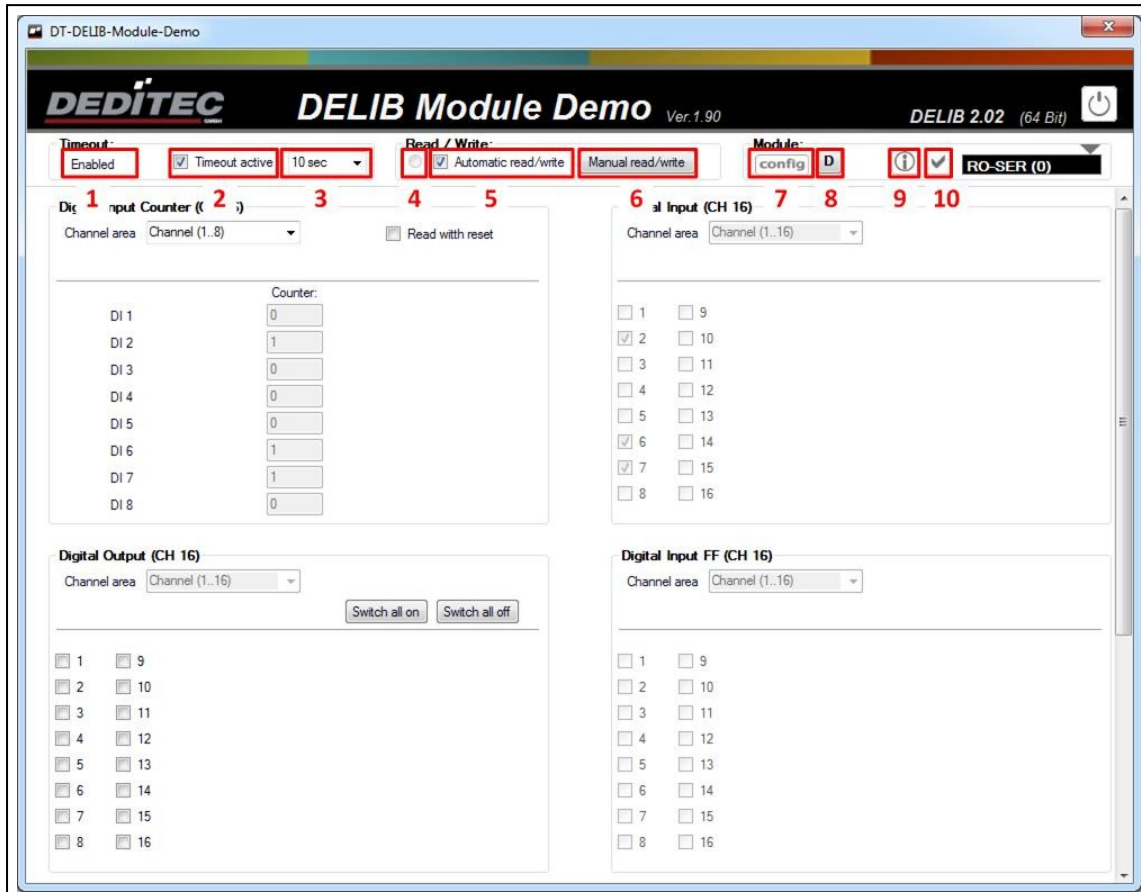
### 3.2.6.1. Auswahl des Moduls

Bei Programmstart muss ein Modul ausgewählt werden.

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



### 3.2.6.2. Allgemein

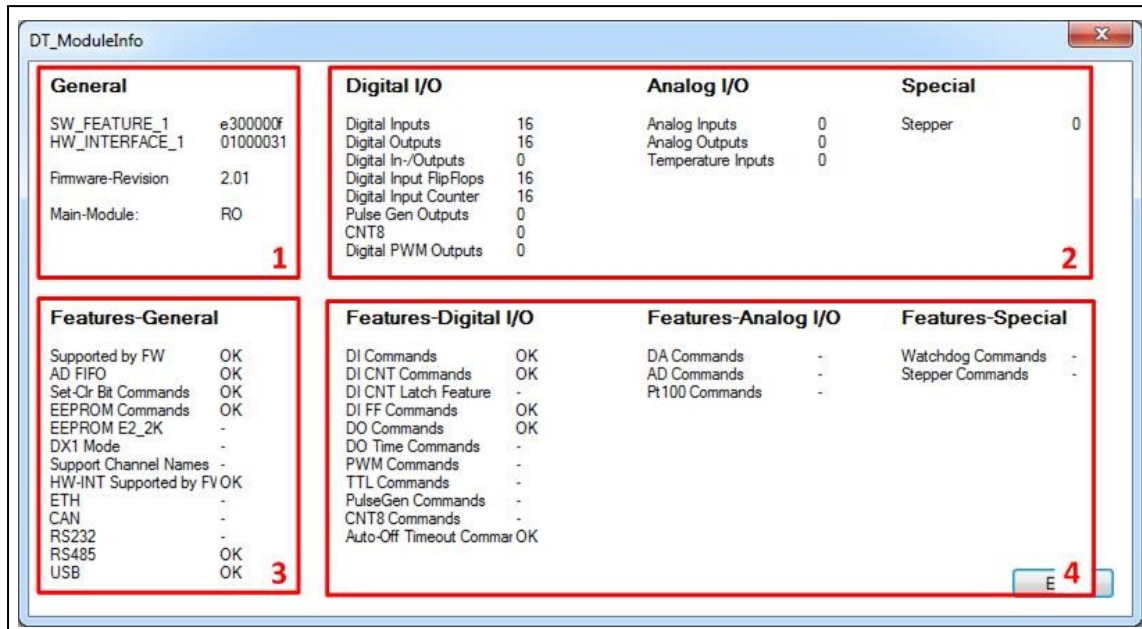


1. Timeout-Status ("Disabled", "Enabled" oder "Occured").
2. Aktiviert oder deaktiviert den Timeout-Schutz.
3. Timeout-Zeit für den Timeout-Schutz.
4. Status für "Automatic read/write" (Blinkt, wenn "Automatic read/write" aktiviert ist).
5. Mit dem Haken bei "Automatic read/write" wird festgelegt, ob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



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

### 3.2.6.2.1. Module Info



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

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

### 3.2.6.3. Digital Input

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

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

**Digital Input Counter (CH 16)**

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

☐ Read with reset **2**

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

**3**

**Digital Input (CH 16)**

Channel area: Channel (1..16)

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

**4**

**Digital Input FF (CH 16)**

Channel area: Channel (1..16)

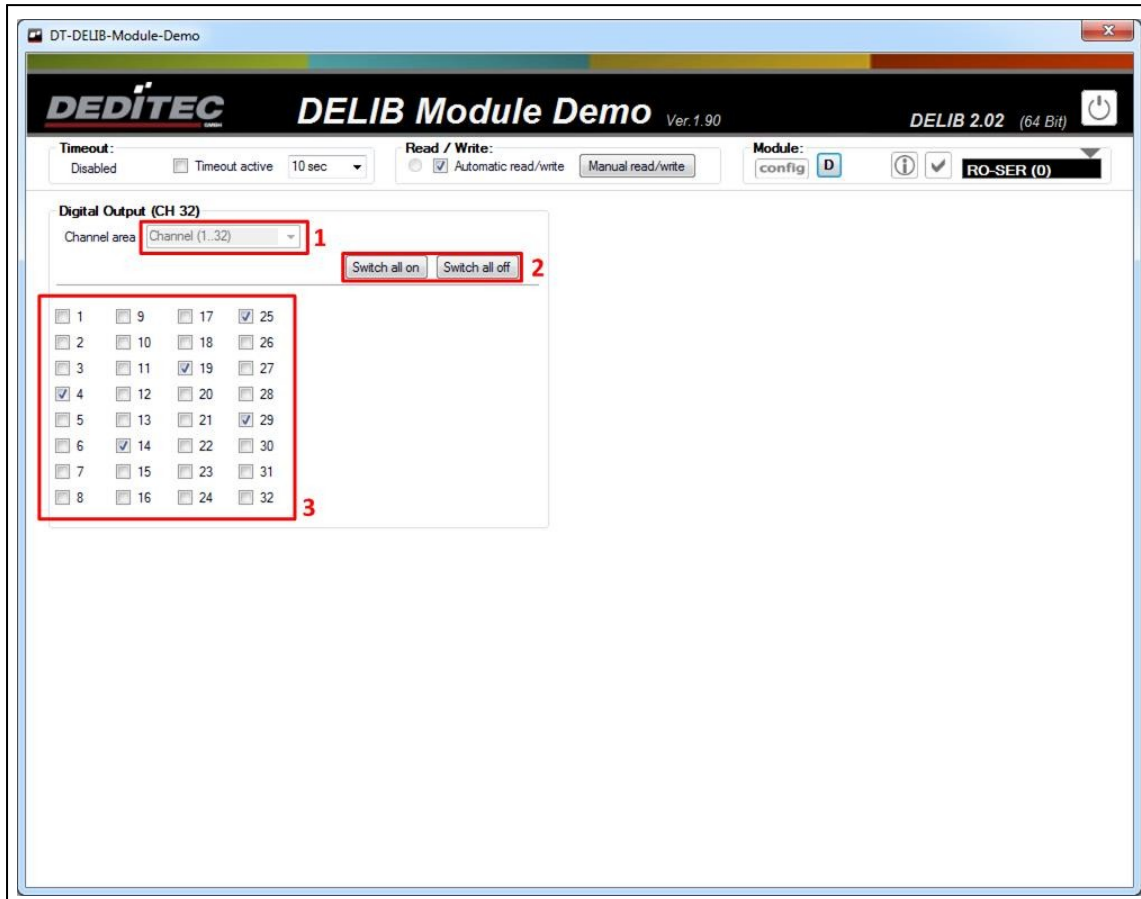
|                                       |                             |
|---------------------------------------|-----------------------------|
| <input type="checkbox"/> 1            | <input type="checkbox"/> 9  |
| <input type="checkbox"/> 2            | <input type="checkbox"/> 10 |
| <input checked=""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

**5**

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

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

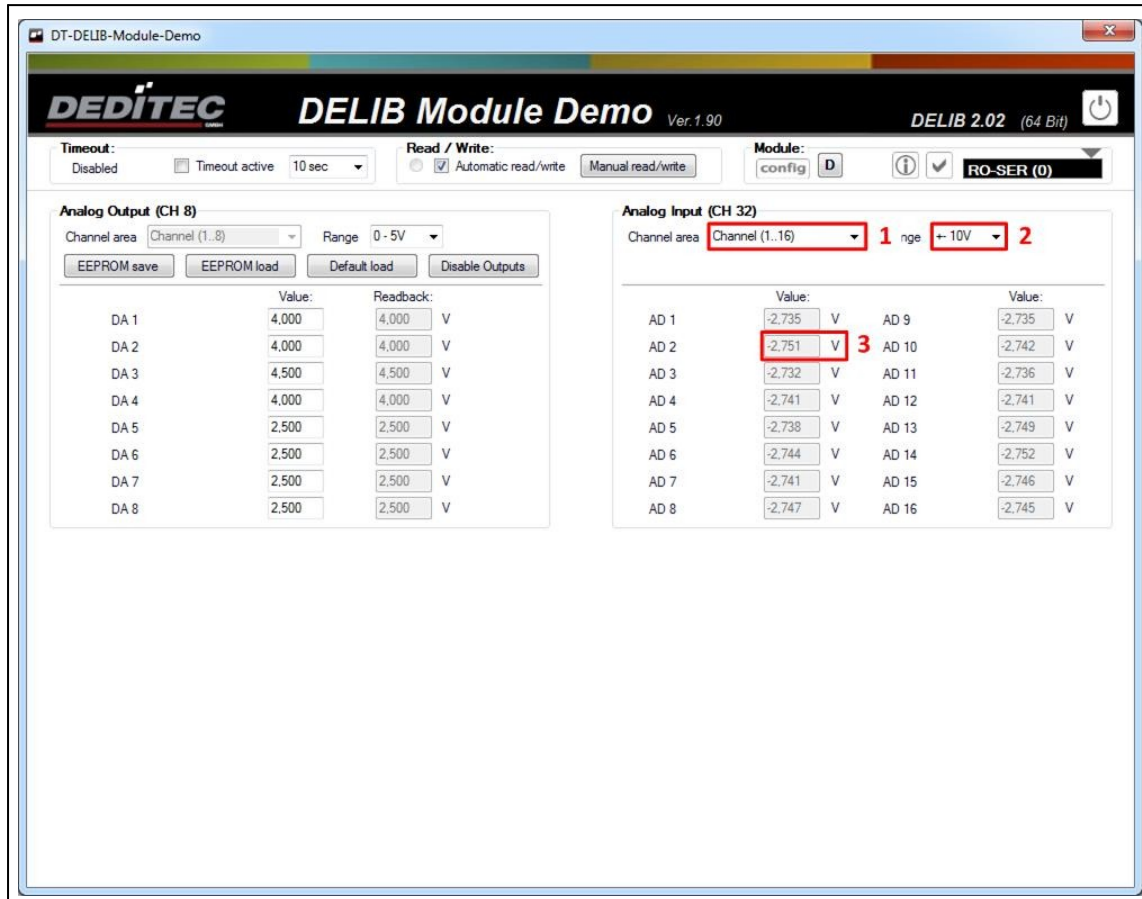
### 3.2.6.4. Digital Output



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

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

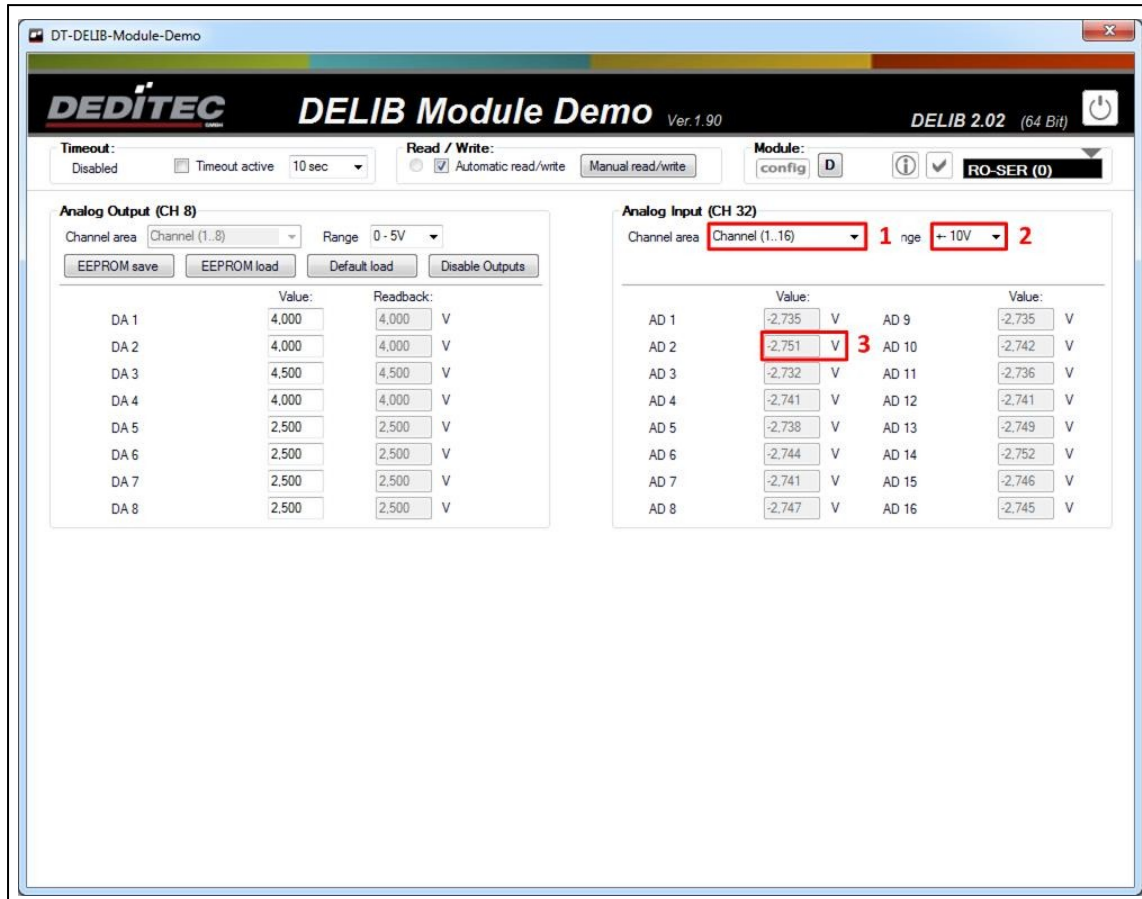
### 3.2.6.5. Analog Input



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

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

### 3.2.6.6. Analog Output



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

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

### 3.2.7. CAN Configuration Utility

#### Hinweis:

Um ein CAN-Modul konfigurieren zu können, muss dieses zunächst in den Software-Modus gebracht werden.

[Konfiguration Produkte der RO-Serie](#)

[Konfiguration Produkte der BS-Serie](#)

Das CAN-Configuration-Utility ermöglicht eine einfache Konfiguration von Produkten mit einer CAN-Schnittstelle. Es ist möglich Konfigurationen auf Module zu übertragen und auszulesen.

Zusätzlich kann der automatische Empfangsmodus (Auto-RX) und der automatische Sendemodus (Auto-TX) konfiguriert werden.

Der Auto-TX Modus erlaubt ein zyklisches Senden von Datenpaketen, wahlweise mit analogen oder digitalen Eingangszuständen an andere CAN-Adressen. Alternativ kann auch ein Trigger-Event definiert werden. Hierbei wird ein Datenpaket erst dann gesendet, wenn zuvor ein Datenpaket auf einer gewissen CAN-ID empfangen wurde (z.B. CAN-Sync auf ID 0x80).

Mit dem Auto-RX Modus hingegen werden empfangene Datenpakete direkt an analoge oder digitale Ausgänge weitergeleitet. So können beispielsweise Relais-Ausgänge über einen anderen CAN-Bus Teilnehmer gesetzt werden.

### **Folgende Einstellungen können verändert werden:**

- Baudrate
- Adress-Modus
- Modul-ID
- Response-ID
- Modi, mit denen Submodule gestartet werden
- Automatischer Sendemodus (TX-Modus)
- Automatischer Empfangsmodus (RX-Modus)

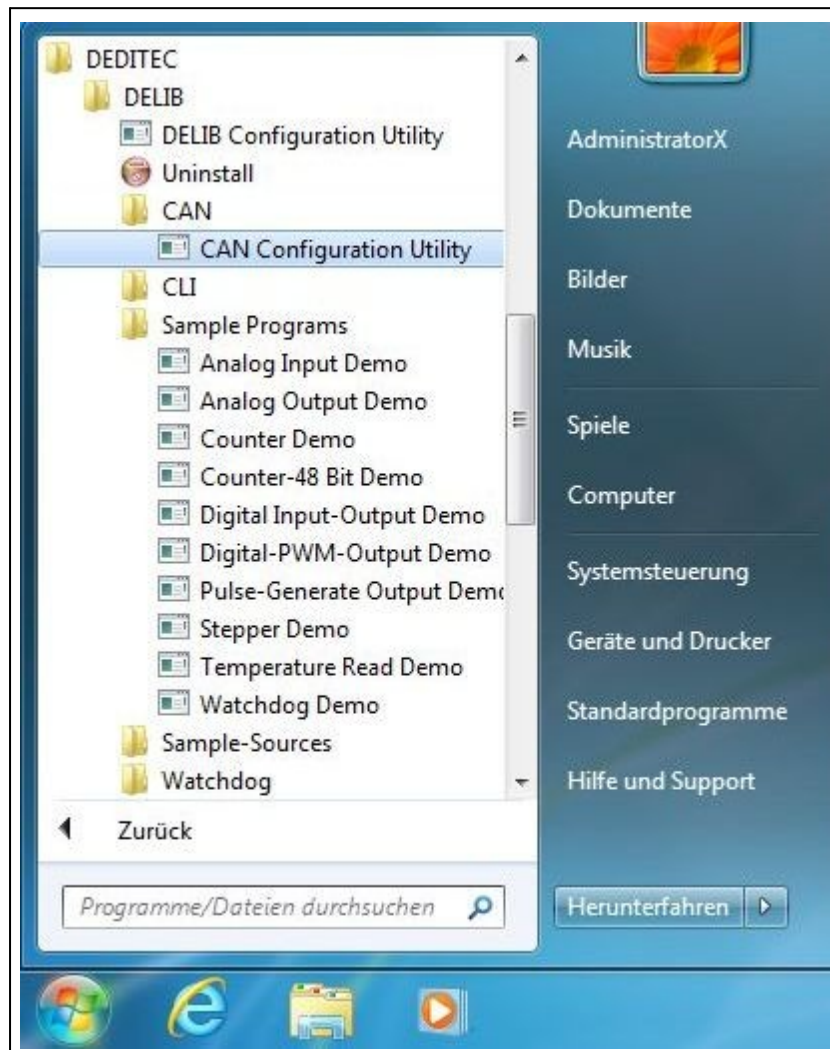
Auf den nachfolgenden Seiten wird Schritt für Schritt gezeigt, wie ein CAN-Modul konfiguriert werden kann.



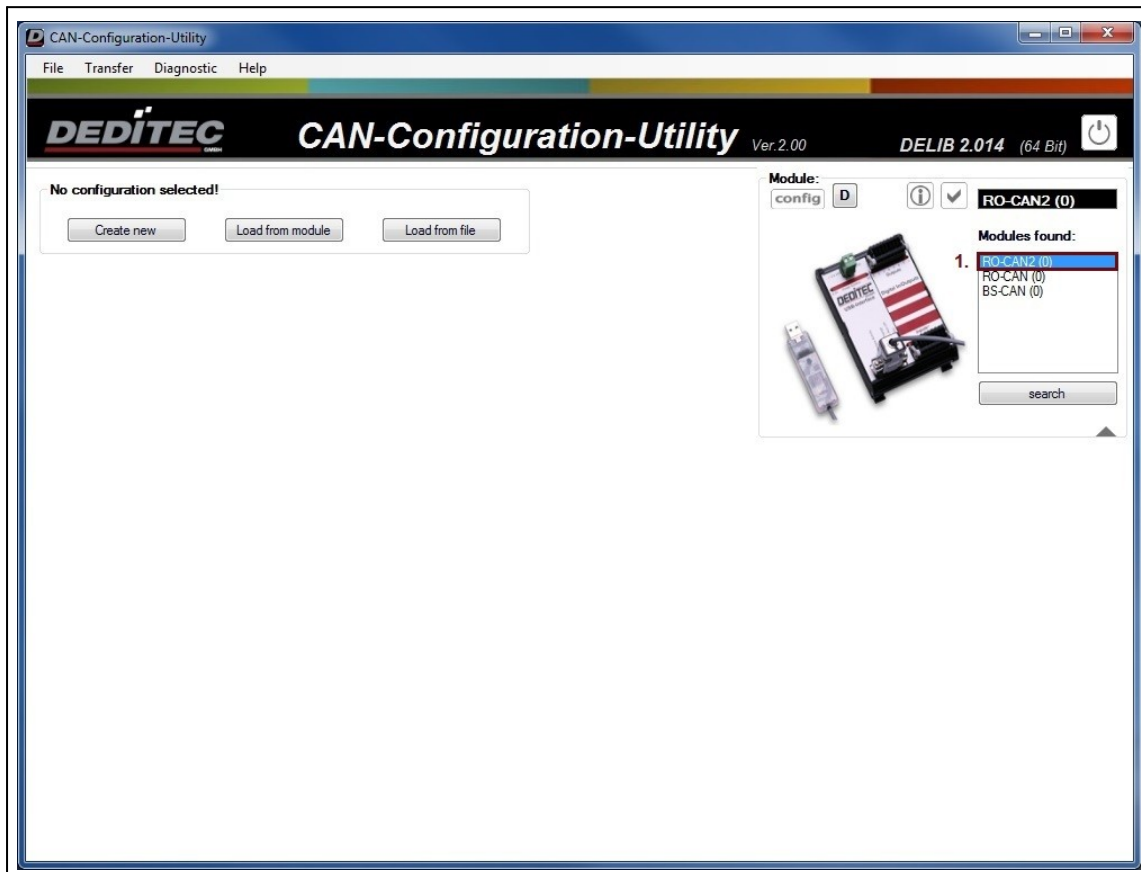
### 3.2.7.1. Auswahl des Moduls

Starten Sie das CAN-Configuration-Utility über:

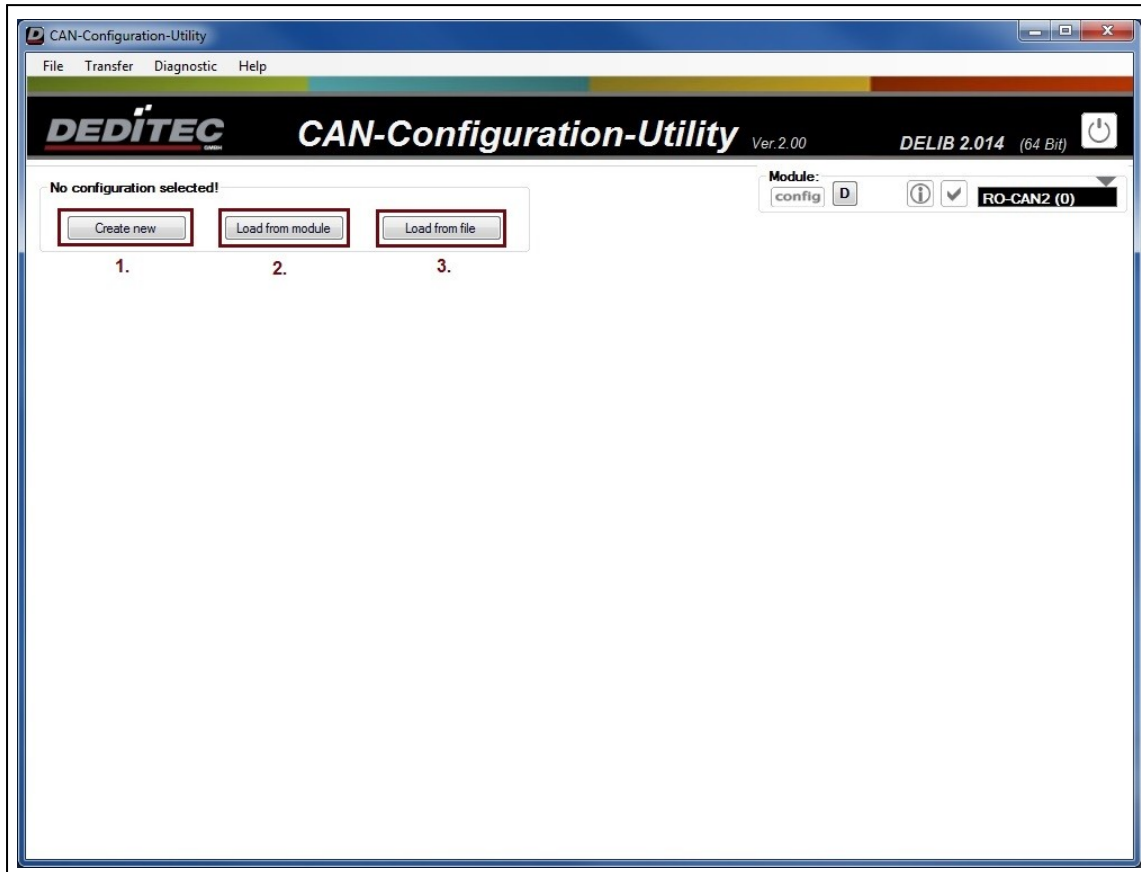
Start → Programme → DEDITEC → DELIB → CAN



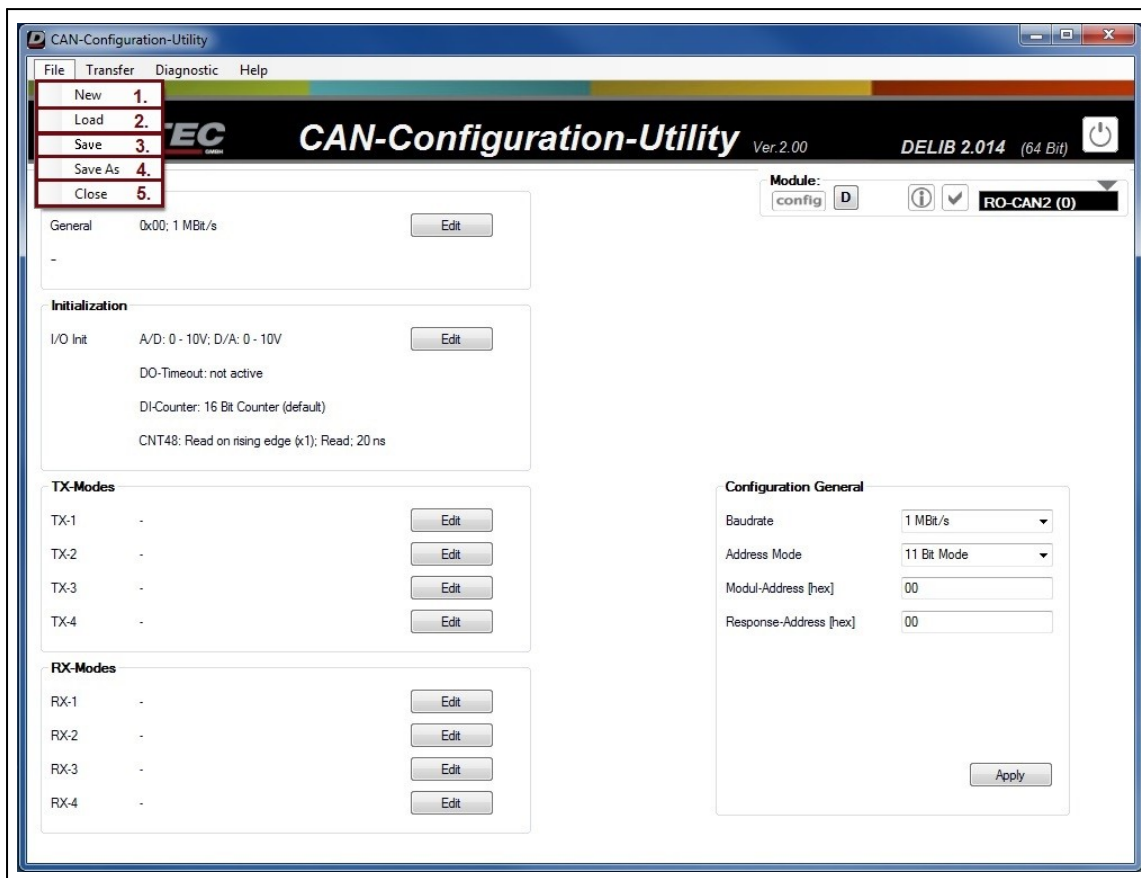
1. In der "Module-Selection" ein entsprechendes CAN-Modul (z.B. das RO-CAN2) auswählen.



### 3.2.7.2. Neue Konfiguration Erstellen, Laden, Speichern



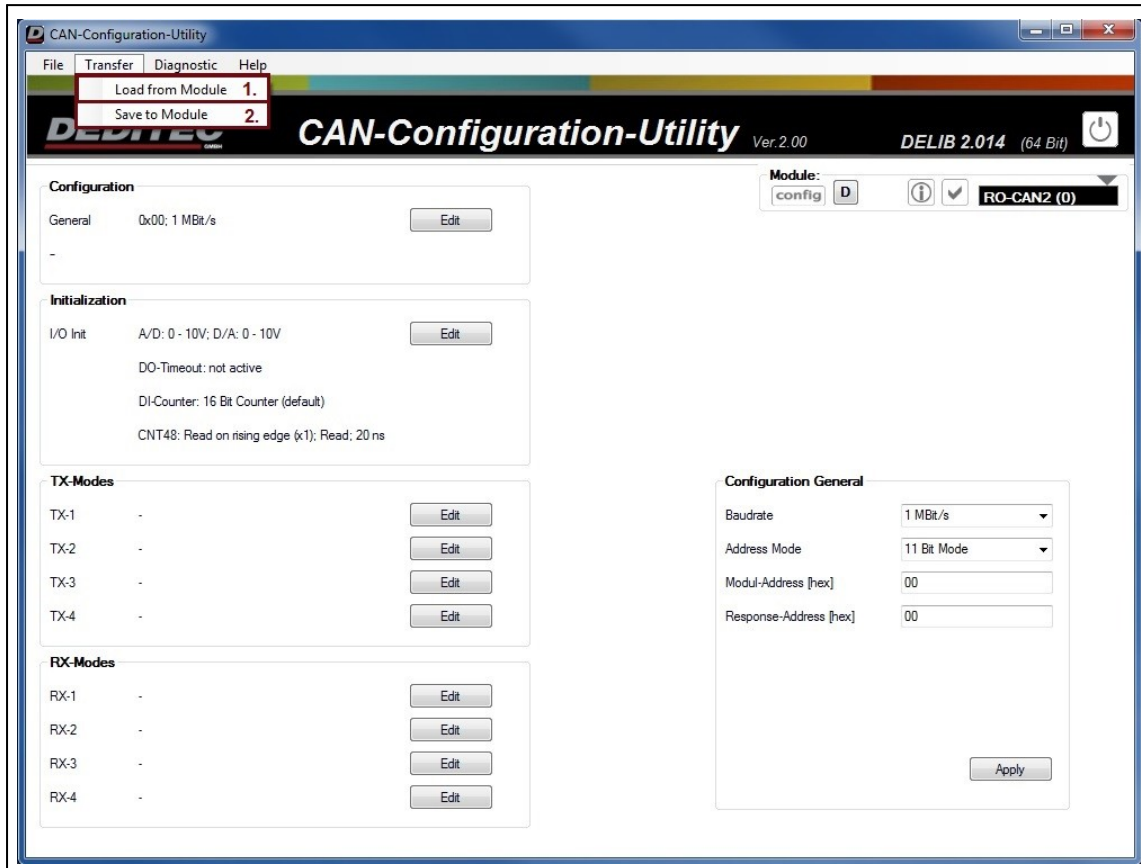
1. Über "Create new" kann eine neue CAN-Konfiguration erstellt werden.
2. Mit "Load from Module" kann die aktuell auf dem Modul vorhandene Konfiguration ausgelesen werden.
3. "Load from File" erlaubt es, zuvor auf einem Datenträger gespeicherte CAN-Konfigurationsdateien zu öffnen.



**Mit einem Klick auf "File" in der Menüleiste öffnet sich ein Untermenü:**

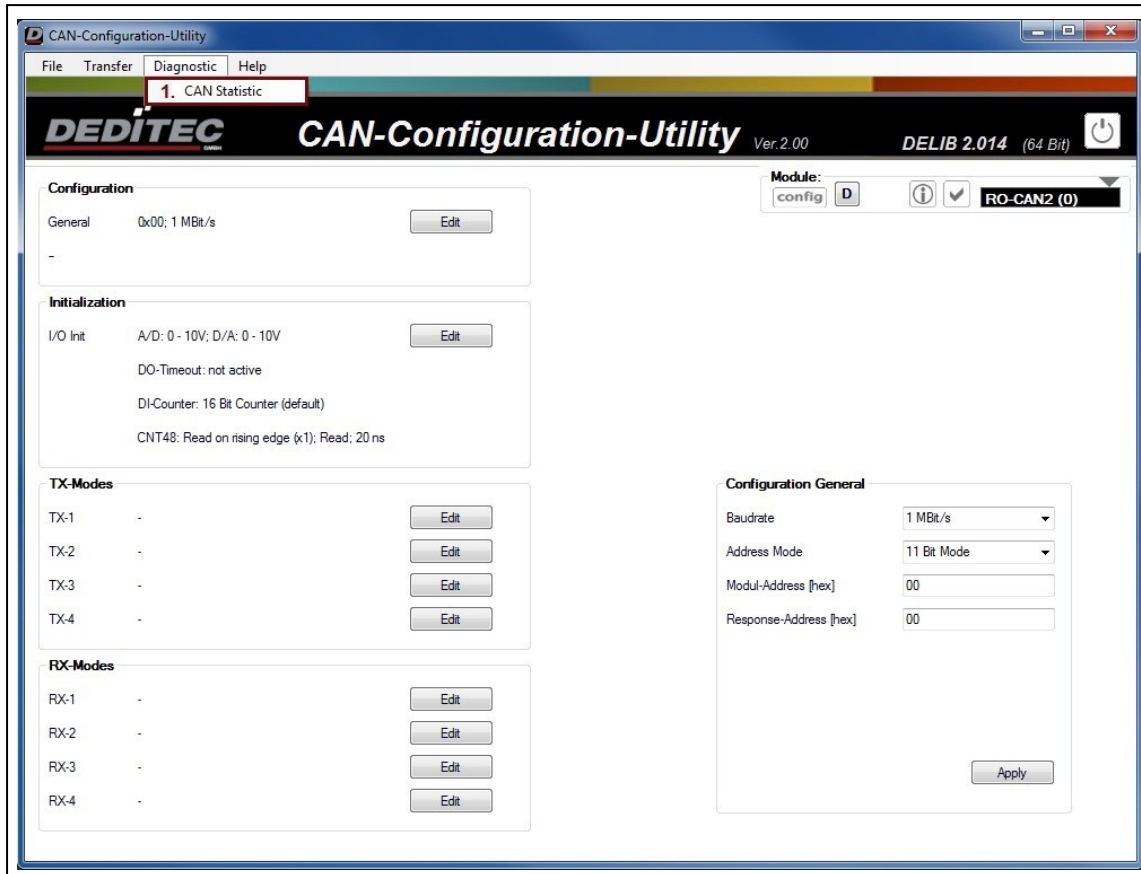
1. Über den Menüpunkt "New" kann eine neue CAN-Konfiguration erstellt werden.
2. "Load" bietet die Möglichkeit, zuvor gespeicherte CAN-Konfigurationsdateien zu öffnen.
3. Wurde eine CAN-Konfigurationsdatei geöffnet, so kann über "Save" die Datei überschrieben und aktuelle Änderungen gespeichert werden.
4. Durch einen Klick auf "Save as" kann die CAN-Konfiguration in einer neuen Datei abgespeichert werden.
5. Hier kann das CAN-Configuration-Utility beendet werden.

### 3.2.7.3. Konfiguration auf das Modul übertragen



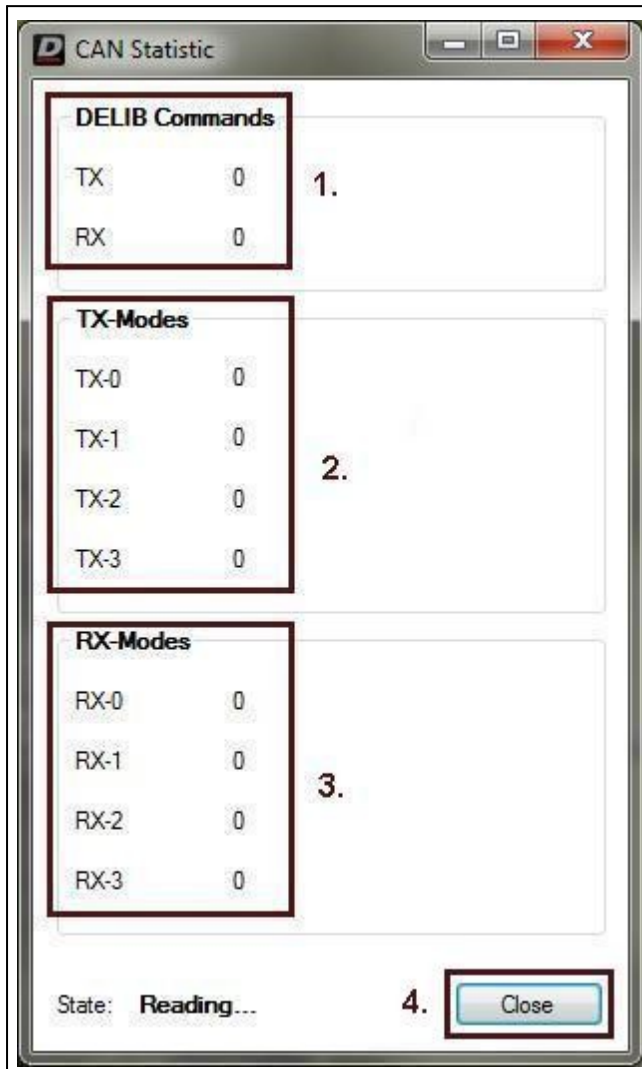
1. Über "Load from Module" kann die aktuelle Konfiguration des ausgewählten Moduls ausgelesen werden.
2. "Save to Module" überträgt die aktuelle CAN-Konfiguration auf das ausgewählte Modul.

### 3.2.7.4. Statistiken vom Modul abfragen



1. Über den Menüpunkt CAN-Statistic kann ein Informationsfenster angezeigt werden, welches die Anzahl der gesendeten und empfangenen TX- bzw. RX-Pakete darstellt. Ebenfalls wird die Anzahl der gesendeten und empfangenen DELIB Kommandos dargestellt.

Folgendes Fenster zeigt die Statistiken an:



1. Dieser Bereich zeigt die Anzahl der gesendeten und empfangenen DELIB Kommandos an.
2. Für den TX-Modus wird hier die Anzahl der gesendeten Pakete aufgelistet.
3. Die empfangenen RX-Pakete werden hier dargestellt.
4. Über "Close" kann das Fenster geschlossen werden.

### **Hinweis**

Die CAN-Statistik zählt im Hintergrund weiter, auch wenn das Fenster geschlossen ist. Bei einem Neustart des Moduls wird die Statistik auf null zurückgesetzt.



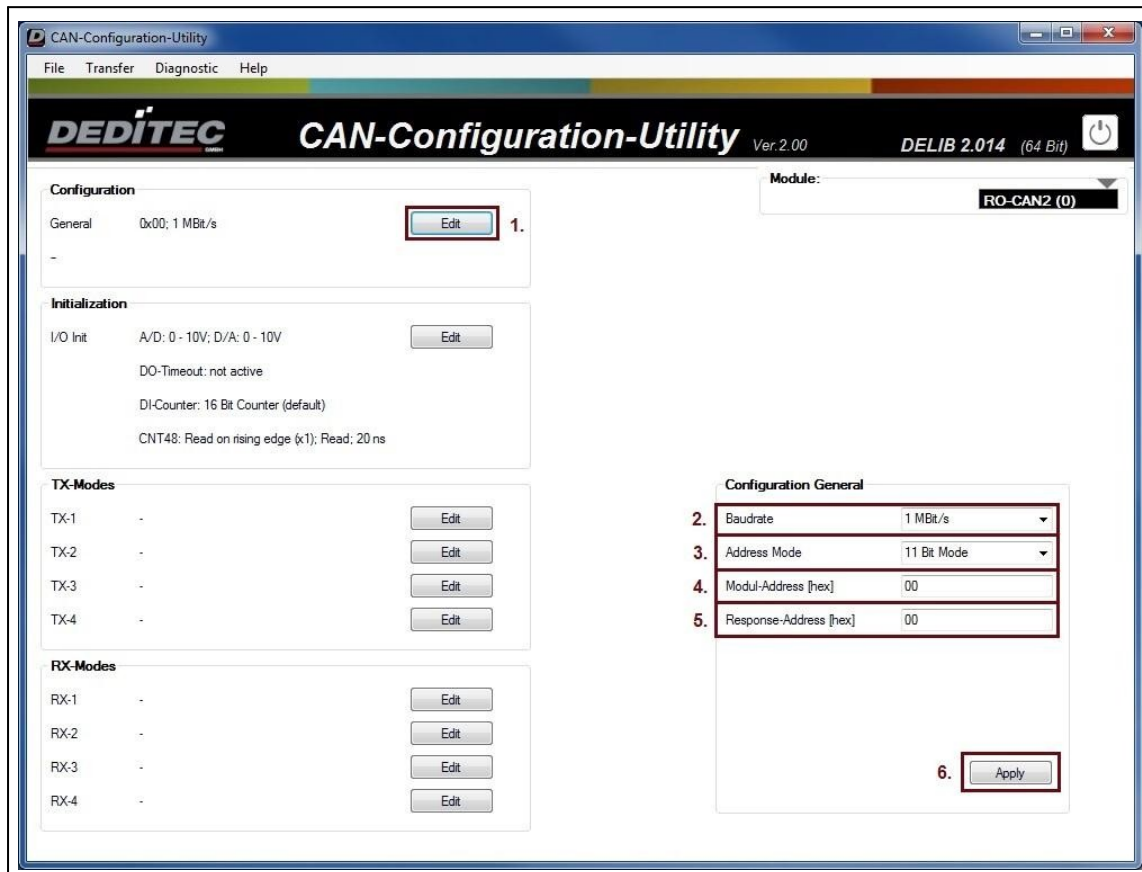
### **3.2.7.5. Konfiguration**

Für jedes CAN-Modul können 4 unterschiedliche Konfigurationen für TX- und RX-Pakete angelegt werden.

Ebenfalls kann definiert werden, in welchem Modus Submodule gestartet werden. So ist es z.B. möglich, dass ein AD-Modul in einem Messbereich von 0-5V und nicht im Standard Messbereich von  $\pm 10V$  gestartet wird.

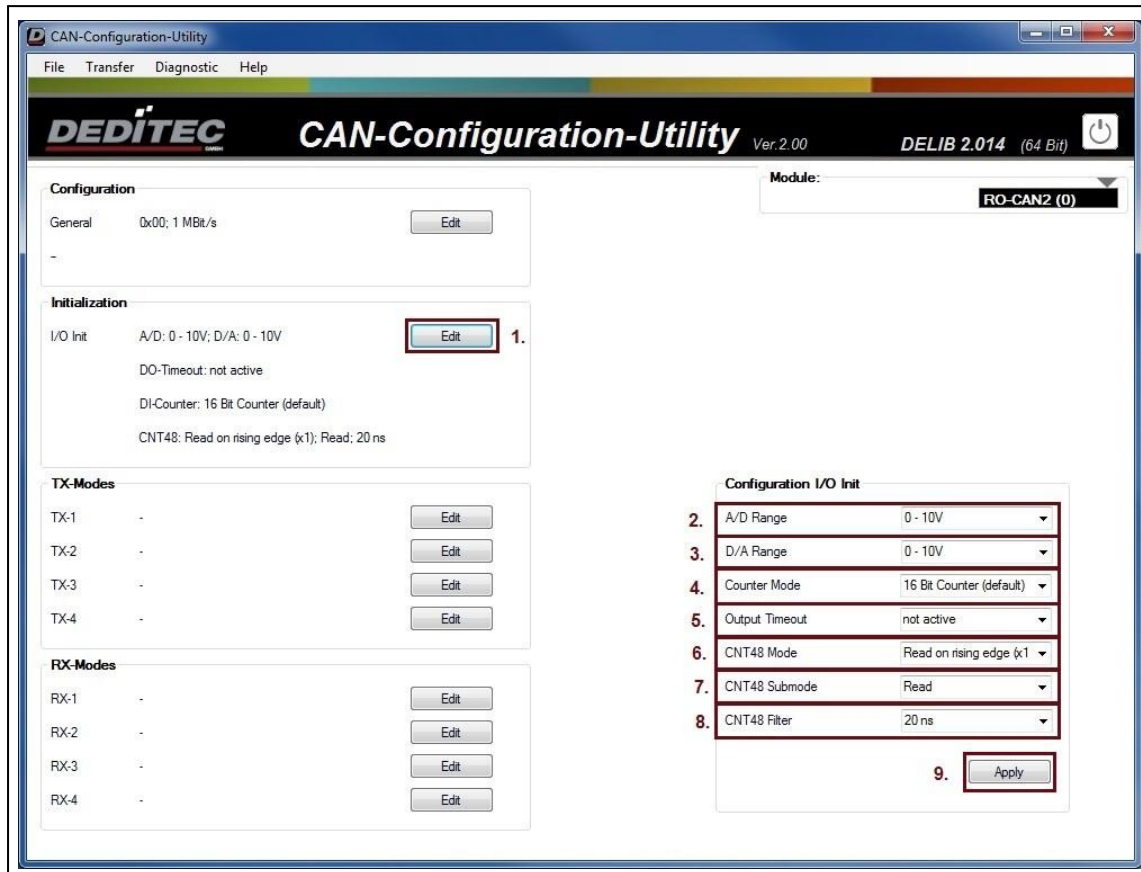
Auf den nachfolgenden Seiten wird gezeigt, wie die verschiedenen Modi konfiguriert werden.

### 3.2.7.5.1. Modul Konfiguration



1. Über den Menüpunkt "Edit" wird ein Zusatzfenster geöffnet, in dem die Konfiguration vorgenommen werden kann.
2. Hier kann die Baudrate eingestellt werden, mit der das Modul kommunizieren soll.
3. Der Address Mode gibt vor, wie viel Bit zur Adressierung verwendet werden.
4. Die Modul-Address legt fest, unter welcher Adresse das Modul im CAN-Bus identifiziert wird.
5. Die Response-Address gibt vor, an welche Module-Adresse eine Bestätigung gesendet wird, sobald ein Paket empfangen wurde.
6. Über "Apply" werden die Änderungen übernommen.

### 3.2.7.5.2. I/O Konfiguration



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

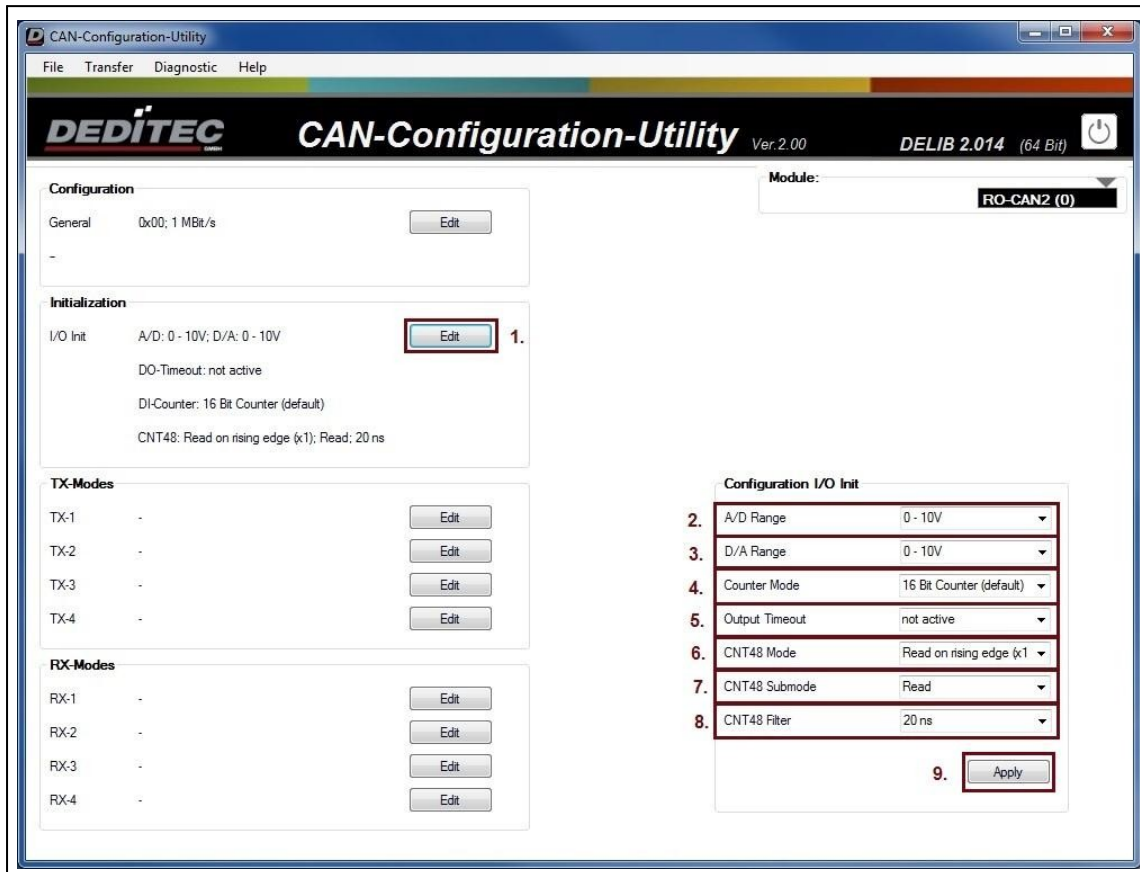
#### Hinweis:

Sind die entsprechenden Submodule nicht vorhanden, so haben die Einstellungen keine Auswirkung.

1. Über den Menüpunkt "Edit" wird ein Zusatzfenster geöffnet, in dem die Konfiguration vorgenommen werden kann.
2. Der Wertebereich gibt den Bereich an, in dem analoge Signale, digital (z.B. im Bereich 0-10V) umgesetzt werden.
3. Der Wertebereich gibt den Bereich an, in dem digitale Signale, analog (z.B. im Bereich 0-10V) umgesetzt werden.
4. Ist zuständig für den Counter Modus von O8-R8 Modulen. Wahlweise kann

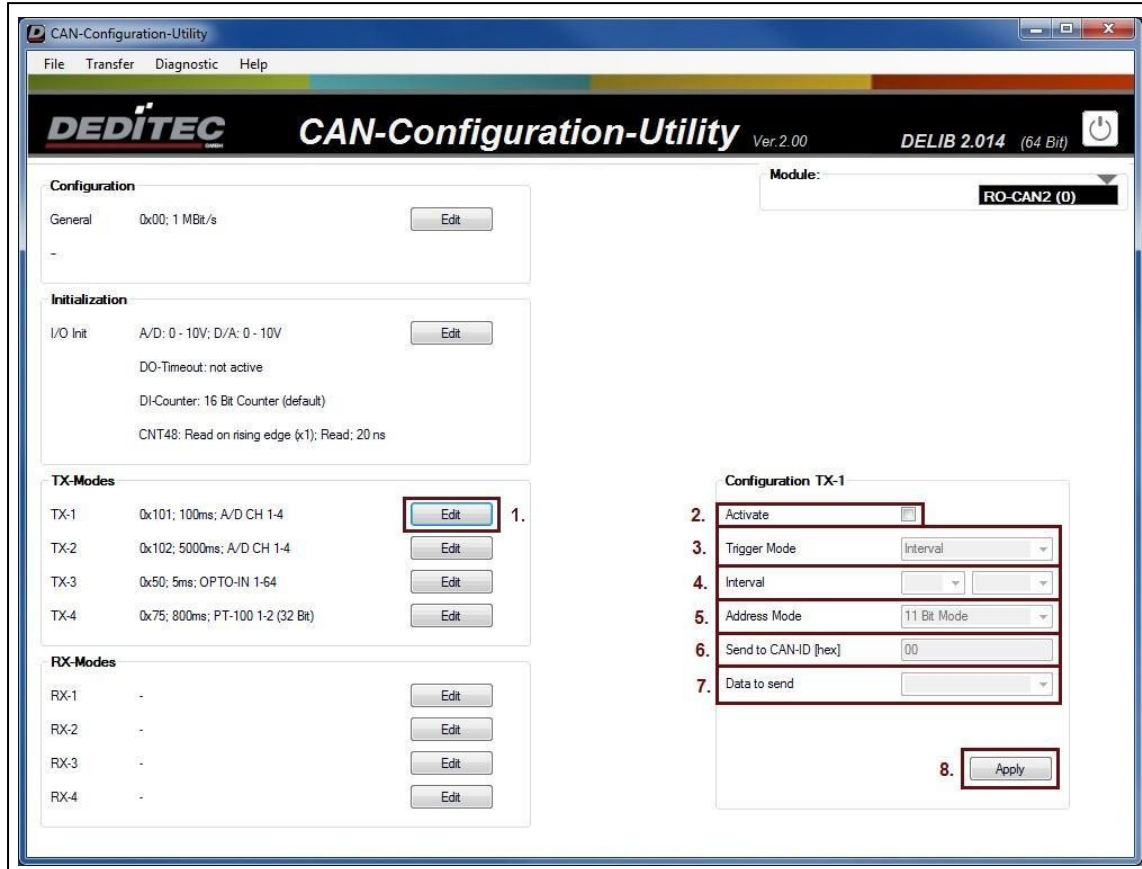
mit 16-Bit hochgezählt werden, oder mit je 8-Bit hoch- und heruntergezählt werden.

5. Gibt die Zeit vor, nach der die Ausgänge abschalten, wenn ein Modul nicht mehr erreicht werden kann. Wird kein Timeout erwünscht wählen Sie bitte die Einstellung "not active".



6. Stellt ein, welcher Counter Modus (nur für RO-CNT8 Module) benutzt werden soll. Es stehen hierbei 5 verschiedene Modi zur Auswahl. Eine genaue Beschreibung der Modi ist im Handbuch "RO-CNT8" zu finden.
7. In Abhängigkeit von dem unter 6. gewählten Modus, stehen hier entsprechende Submodi zur Auswahl. Eine genaue Beschreibung der Submodi ist im Handbuch "RO-CNT8" zu finden.
8. Stellt den Filter ein, wie lange ein Signal mindestens sein muss, damit dieses als High erkannt wird. Es kann aus 16 vorgegebenen Filtern zwischen 20ns und 5ms ausgewählt werden (nur für RO-CNT8 Module).
9. Über "Apply" werden die Änderungen übernommen.

### 3.2.7.5.3. TX-Konfiguration



Es können bis zu 4 unabhängige TX-Modi eingestellt werden. Die Konfiguration ist für alle Modi identisch. Das folgende Beispiel zeigt die Konfiguration für den ersten TX-Modus.

1. Über den Menüpunkt "Edit" wird ein Zusatzfenster geöffnet, in dem die Konfiguration vorgenommen werden kann.
2. Um die Konfiguration zu aktivieren, muss hier der Haken gesetzt werden.
3. Der Trigger Mode gibt vor, unter welcher Bedingung Daten gesendet werden. Entweder im Intervall oder in Abhängigkeit eines empfangenen RX-Paketes.
4. Hier wird das Intervall konfiguriert (wenn unter Punkt 3. "Intervall" ausgewählt), in dem Pakete versendet werden.
5. Der Address Mode gibt vor, wie viel Bit zur Adressierung verwendet werden.
6. Hier wird festgelegt, an welche Moduladresse CAN-Pakete gesendet werden.
7. Unter diesem Punkt wird definiert, welche Art von Daten an die unter Punkt 6. konfigurierte Adresse gesendet werden. Wie z.B. → der Status der digitalen Eingänge oder → die Spannung von A/D Kanälen.
8. Über "Apply" werden die Änderungen übernommen.

#### 3.2.7.5.3.1. Beispiel Interval

**Configuration TX-1**

|                      |                                     |
|----------------------|-------------------------------------|
| Activate             | <input checked="" type="checkbox"/> |
| Trigger Mode         | Interval                            |
| Interval             | 1 * 1sec                            |
| Address Mode         | 11 Bit Mode                         |
| Send to CAN-ID [hex] | 0x100                               |
| Data to send         | OPTO-IN 1-64                        |

Apply

Das Beispiel beinhaltet folgende Einstellungen:

Im Intervall von einer Sekunde werden die Daten der digitalen Eingänge 1-64 an die CAN-Adresse 0x100 gesendet.



### 3.2.7.5.3.2. Beispiel Trigger

**Configuration RX-1**

|                         |                                     |
|-------------------------|-------------------------------------|
| Activate                | <input checked="" type="checkbox"/> |
| Address Mode            | 11 Bit Mode                         |
| Receive at CAN-ID [hex] | 200                                 |
| Data to send            | Trigger Auto TX 1                   |

Apply

**Configuration TX-1**

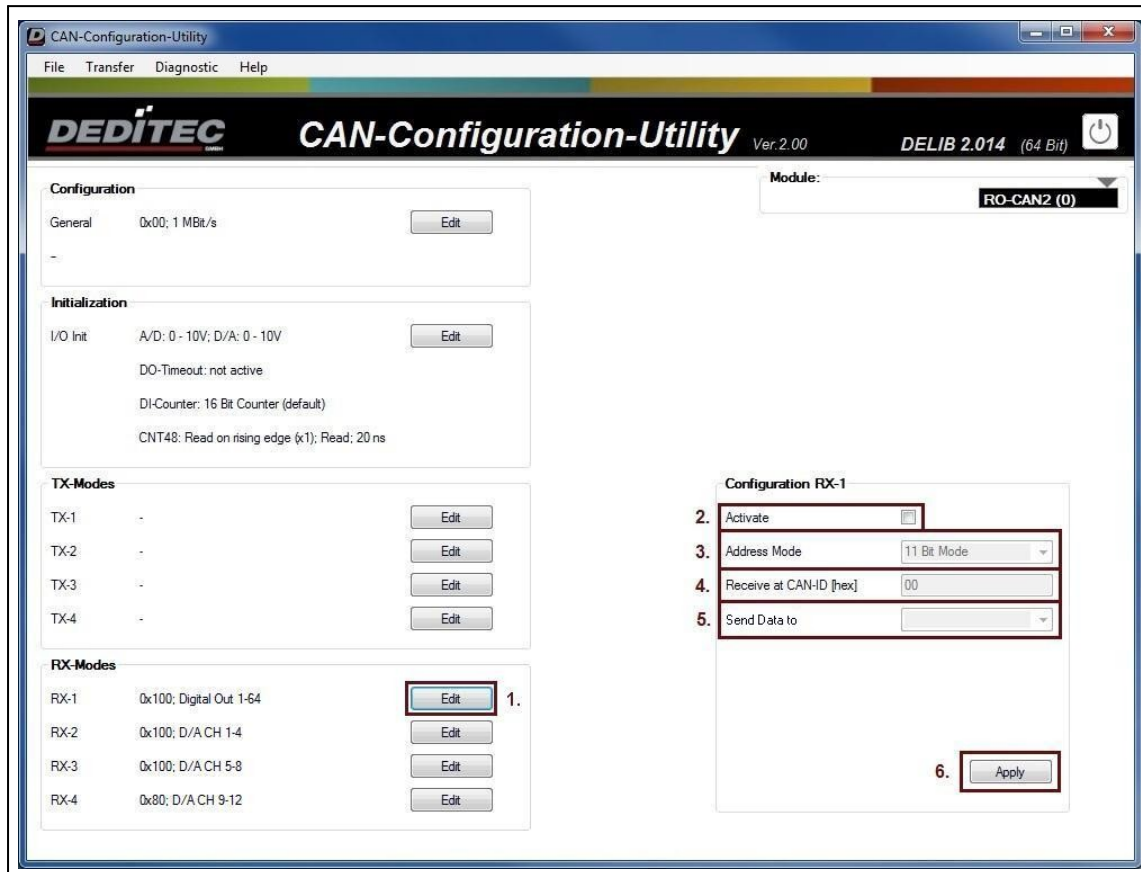
|                      |   |
|----------------------|---|
| Activate             | <input checked="" type="checkbox"/>       |
| Trigger Mode         | RX-Event                                  |
| Interval             | <input type="text"/> <input type="text"/> |
| Address Mode         | 11 Bit Mode                               |
| Send to CAN-ID [hex] | 100                                       |
| Data to send         | A/D CH 1-4                                |

Apply

#### Das Beispiel beinhaltet folgende Einstellungen:

Werden Daten auf der CAN-Adresse 0x200 empfangen, wird TX-1 ausgeführt (Bild oben), welches die Daten der A/D Kanäle 1-4 an die CAN-Adresse 0x100 versendet (Bild unten).

### 3.2.7.5.4. RX-Konfiguration



Es können bis zu 4 unabhängige RX-Modi eingestellt werden. Die Konfiguration ist für alle Modi identisch. Das folgende Beispiel zeigt die Konfiguration für den ersten RX-Modus.

1. Über den Menüpunkt "Edit" wird ein Zusatzfenster geöffnet, in dem die Konfiguration vorgenommen werden kann.
2. Um die Konfiguration zu aktivieren muss hier der Haken gesetzt werden.
3. Der Address Mode gibt vor, wie viel Bit zur Adressierung verwendet werden.
4. Hier wird festgelegt, auf welcher CAN-ID die Datenpakete erwartet werden.
5. Wurden auf der unter Punkt 4. konfigurierten ID Daten empfangen, so wird hier definiert, welche Aktion vom Modul ausgeführt werden soll. Wie z.B. → automatisch die Ausgangsspannung bei einem analogen Ausgangsmodul setzen oder → Relais-Ausgänge schalten.
6. Über "Apply" werden die Änderungen übernommen.

#### 3.2.7.5.4.1. Beispiel RX-DA

**Configuration RX-1**

|                         |                                     |
|-------------------------|-------------------------------------|
| Activate                | <input checked="" type="checkbox"/> |
| Address Mode            | 11 Bit Mode                         |
| Receive at CAN-ID [hex] | 201                                 |
| Data to send            | D/A CH 5-8                          |

Apply

#### Das Beispiel beinhaltet folgende Einstellungen:

Wurde ein CAN-Paket auf der Adresse 0x201 empfangen, wird der Inhalt des Datenpaketes an den analogen Ausgängen 5-8, unter Berücksichtigung des ausgewählten D/A-Modus gesetzt.

#### 3.2.7.5.4.2. Beispiel RX-DO

**Configuration RX-1**

Activate ☒

Address Mode 11 Bit Mode

Receive at CAN-ID [hex] 100

Data to send Digital Out 1-64

Apply

#### Das Beispiel beinhaltet folgende Einstellungen:

Wurde ein CAN-Paket auf der Adresse 0x100 empfangen, wird der Inhalt des Datenpaketes an die digitalen Ausgänge 1-64 weitergeleitet, woraufhin dort die Ausgänge ein- oder ausgeschaltet werden.

### 3.2.7.6. Aufbau der CAN-Pakete

Die folgenden Seiten zeigen, wie die CAN-Pakete der unterschiedlichen I/O-Module aufgebaut sind. Ebenfalls wird dargestellt, wie A/D- und D/A-Werte berechnet werden können.

Die Daten der CAN-Pakete sind jeweils 8 Byte groß. Welche Daten an welcher Stelle zu finden sind, entnehmen Sie bitte den folgenden Seiten.

#### 3.2.7.6.1. Digitale Eingänge

Aufbau eines 8 Byte langen CAN-Paketes:

| CAN-Data-Byte | Inhalt                     |
|---------------|----------------------------|
| 1             | DI channel 1-8 (Bit 0-7)   |
| 2             | DI channel 9-16 (Bit 0-7)  |
| 3             | DI channel 17-24 (Bit 0-7) |
| 4             | DI channel 25-32 (Bit 0-7) |
| 5             | DI channel 33-40 (Bit 0-7) |
| 6             | DI channel 41-48 (Bit 0-7) |
| 7             | DI channel 49-56 (Bit 0-7) |
| 8             | DI channel 57-64 (Bit 0-7) |

### 3.2.7.6.2. Digitale Ausgänge

Aufbau eines 8 Byte langen CAN-Paketes:

| CAN-Data-Byte | Inhalt                     |
|---------------|----------------------------|
| 1             | DO channel 1-8 (Bit 0-7)   |
| 2             | DO channel 9-16 (Bit 0-7)  |
| 3             | DO channel 17-24 (Bit 0-7) |
| 4             | DO channel 25-32 (Bit 0-7) |
| 5             | DO channel 33-40 (Bit 0-7) |
| 6             | DO channel 41-48 (Bit 0-7) |
| 7             | DO channel 49-56 (Bit 0-7) |
| 8             | DO channel 57-64 (Bit 0-7) |

### 3.2.7.6.3. Digitale Eingangszähler (16-Bit)

Aufbau eines 8 Byte langen CAN-Paketes:

| CAN-Data-Byte | Inhalt                  |
|---------------|-------------------------|
| 1             | DI counter 1 (Bit 0-7)  |
| 2             | DI counter 1 (Bit 8-15) |
| 3             | DI counter 2 (Bit 0-7)  |
| 4             | DI counter 2 (Bit 8-15) |
| 5             | DI counter 3 (Bit 0-7)  |
| 6             | DI counter 3 (Bit 8-15) |
| 7             | DI counter 4 (Bit 0-7)  |
| 8             | DI counter 4 (Bit 8-15) |



#### 3.2.7.6.4. Digitale Eingangszähler (48-Bit) - 32-Bit Paket

Aufbau eines 8 Byte langen CAN-Paketes:

| CAN-Data-Byte | Inhalt                     |
|---------------|----------------------------|
| 1             | CNT8 counter 1 (Bit 0-7)   |
| 2             | CNT8 counter 1 (Bit 8-15)  |
| 3             | CNT8 counter 1 (Bit 16-23) |
| 4             | CNT8 counter 1 (Bit 24-31) |
| 5             | CNT8 counter 2 (Bit 0-7)   |
| 6             | CNT8 counter 2 (Bit 8-15)  |
| 7             | CNT8 counter 2 (Bit 16-23) |
| 8             | CNT8 counter 2 (Bit 24-31) |

#### Hinweis:

Beachten Sie, dass jeweils nur die ersten 32-Bit der beiden 48-Bit Eingangszähler automatisch in einem CAN-Paket versendet werden können.

### 3.2.7.6.5. Digitale Eingangszähler (48-Bit) - 64-Bit Paket

Aufbau eines 8 Byte langen CAN-Paketes:

| CAN-Data-Byte | Bit | Inhalt                               |
|---------------|-----|--------------------------------------|
| 1             | 0-7 | CNT8 counter 1 (Bit 0-7)             |
| 2             | 0-7 | CNT8 counter 1 (Bit 8-15)            |
| 3             | 0-7 | CNT8 counter 1 (Bit 16-23)           |
| 4             | 0-7 | CNT8 counter 1 (Bit 24-31)           |
| 5             | 0-7 | CNT8 counter 1 (Bit 31-39)           |
| 6             | 0-7 | CNT8 counter 1 (Bit 40-47)           |
| 7             | 0-3 | CNT8 counter 1 Zähler-Modus          |
| 7             | 4-7 | CNT8 counter 1 Sub-Modus             |
| 8             | 0   | CNT8 counter 1 Eingangszustand (0/1) |
| 8             | 1-3 | Nicht benutzt                        |
| 8             | 4-7 | CNT8 counter 1 Eingangsfiler         |

### 3.2.7.6.6. Analoge Ein- / Ausgänge

#### 3.2.7.6.6.1. Analoge Eingänge

Aufbau eines 8 Byte langen CAN-Paketes:

| CAN-Data-Byte | Inhalt                   |
|---------------|--------------------------|
| 1             | A/D channel 5 (Bit 0-7)  |
| 2             | A/D channel 5 (Bit 8-15) |
| 3             | A/D channel 6 (Bit 0-7)  |
| 4             | A/D channel 6 (Bit 8-15) |
| 5             | A/D channel 7 (Bit 0-7)  |
| 6             | A/D channel 7 (Bit 8-15) |
| 7             | A/D channel 8 (Bit 0-7)  |
| 8             | A/D channel 8 (Bit 8-15) |

Der Wertebereich eines A/D Wandlers gibt an, in welchem Bereich analoge Signale (z.B. im Bereich 0-5V), digital umgesetzt werden. Die Einstellungen bezüglich des Wertebereiches können im CAN-Configuration-Utility vorgenommen werden.

**Anmerkung:**

Der hexadezimal Wert FFFF kennzeichnet immer die obere Grenze eines Wertebereiches, der Wert 0000 die untere.

Formel (A/D-Modus +/-10V, +/-5V oder +/-2,5V)

Spannung =  $(\text{Wert} * (\text{max. Spannungswert} * 2) / 0xFFFF) - \text{max. Spannungswert}$

Formel (A/D-Modus 0..10V, 0..5V oder 0..2,5V)

Spannung =  $\text{Wert} * \text{max. Spannungswert} / 0xFFFF$

Formel (A/D-Modus 0..20mA, 4..20mA oder 0..24mA)

Stromstärke =  $\text{Wert} * 25 / 0xFFFF$

#### 3.2.7.6.6.2. Analoge Ausgänge

##### Aufbau eines 8 Byte langen CAN-Paketes:

| CAN-Data-Byte | Inhalt                   |
|---------------|--------------------------|
| 1             | D/A channel 1 (Bit 0-7)  |
| 2             | D/A channel 1 (Bit 8-15) |
| 3             | D/A channel 2 (Bit 0-7)  |
| 4             | D/A channel 2 (Bit 8-15) |
| 5             | D/A channel 3 (Bit 0-7)  |
| 6             | D/A channel 3 (Bit 8-15) |
| 7             | D/A channel 4 (Bit 0-7)  |
| 8             | D/A channel 4 (Bit 8-15) |

Der Wertebereich eines D/A Wandlers gibt an, in welchem Bereich digitale Signale analog (z.B. im Bereich 0-5V) umgesetzt werden. Die Einstellungen bezüglich des Wertebereiches können im CAN-Configuration-Utility vorgenommen werden.

##### **Anmerkung:**

Der hexadezimal Wert FFFF kennzeichnet immer die obere Grenze eines Wertebereiches, der Wert 0000 die untere.

##### **Hinweis:**

Die Ausgabe in einen Strombereich ist nur bei Modulen möglich, die diesen Modus auch unterstützen.

### 3.2.7.6.6.3. Beispiele

Beispiel Spannungsbereich  $\pm 10\text{V}$

| Wert (hex) | Spannung |
|------------|----------|
| FFFF       | +10 V    |
| 8000       | 0 V      |
| 0000       | -10V     |

#### Beispiel zur Berechnung:

CAN-Data-Byte0, 1 = 0x4711[hex]

Wertebereich = +/-10 V

#### Berechnung:

Spannung =  $(0x4711 * (10 * 2) / 0xFFFF) - 10 = -4,45 \text{ V}$

Beispiel Spannungsbereich 0-5V

| Wert (hex) | Spannung |
|------------|----------|
| FFFF       | +5 V     |
| 8000       | +2,5 V   |
| 0000       | 0 V      |

**Beispiel zur Berechnung:**

CAN-Data-Byte0, 1 = 0x4711[hex]

Wertebereich = 0-5 V

**Berechnung:**

Spannung =  $0x4711 * 5 / 0xFFFF = 1,38 \text{ V}$

### Beispiel Strombereich

| Wert (hex) | Stromstärke |
|------------|-------------|
| FFFF       | 25 mA       |
| 8000       | 12,5 mA     |
| 0000       | 0 mA        |

### Beispiel zur Berechnung:

CAN-Data-Byte0, 1 = 0x4711[hex]

Wertebereich = 0..20 mA, 4..20 mA oder 0..24 mA

### Berechnung:

Stromstärke =  $0x4711 * 25 / 0xFFFF = 6,94 \text{ mA}$

### Hinweis:

Bitte beachten Sie, dass sich bei einem Auto-TX-Paket mit eingestelltem Strombereich, der Wert eines A/D-Kanals im CAN-Paket auf den Modus 0..25mA bezieht.



### 3.2.7.6.7. Temperatur Eingänge

Dieses Beispiel zeigt den Aufbau eines Auto-TX-Paketes mit den Einstellungen für PT-100 Kanal 1 und 2.

| CAN-Data-Byte | Bit | Inhalt  |
|---------------|-----|---|
| 1             | 0-7 | Wert Kanal 1 (Bit 0-7)  |
| 2             | 0-6 | Wert Kanal 1 (Bit 8-14)   |
|               | 7   | Vorzeichen Kanal 1 (0 = positiv, 1 = negativ)   |
| 3             | 0-1 | Faktor Kanal 1 (<br>0 [dez] = illegaler Wert / Sensor nicht verbunden,<br>1 [dez] = Faktor 10,<br>2 [dez] = Faktor 100) |
|               | 2-7 | nicht benutzt   |
| 4             | 0   | Status PT100 Sensor Kanal 1<br>(0 = nicht verbunden, 1 = verbunden)   |
|               | 1-7 | nicht benutzt   |
| 5             | 0-7 | Wert Kanal 2 (Bit 0-7)  |
| 6             | 0-6 | Wert Kanal 2 (Bit 8-14)   |
|               | 7   | Vorzeichen Kanal 2 (0 = positiv, 1 = negativ)   |
| 7             | 0-1 | Faktor Kanal 2 (<br>0 [dez] = illegaler Wert / Sensor nicht verbunden,<br>1 [dez] = Faktor 10,<br>2 [dez] = Faktor 100) |
|               | 2-7 | nicht benutzt   |
| 8             | 0   | Status PT100 Sensor Kanal 2   |

| CAN-Data-Byte | Bit | Inhalt                               |
|---------------|-----|--------------------------------------|
|               |     | (0 = nicht verbunden, 1 = verbunden) |
|               | 0-7 | nicht benutzt                        |

Berechnung der Temperatur

Temperatur = (Vorzeichen) Wert[dez] / Faktor

### 3.2.7.6.8. Stepper

Aufbau eines 8-Byte langen CAN-Paketes:

| CAN-Data-Byte | Inhalt           |
|---------------|------------------|
| 1             | COMMAND          |
| 2             | PAR1 (Bit 0-7)   |
| 3             | PAR1 (Bit 8-15)  |
| 4             | PAR1 (Bit 16-23) |
| 5             | PAR1 (Bit 24-31) |
| 6             | PAR2 (Bit 0-7)   |
| 7             | PAR2 (Bit 8-15)  |
| 8             | PAR3 (Bit 0-7)   |

#### 3.2.7.6.8.1. Command-Liste

| Kommando<br>DAPI_STEPPER_CMD_ | mit<br>Wert<br>(hex) | Bedeutung                          |
|-------------------------------|----------------------|------------------------------------|
| SET_MOTORCHARACTERISTIC       | 1<br>(hex)           | Setzen der Motor Konfiguration     |
| GET_MOTORCHARACTERISTIC       | 2<br>(hex)           | Abfrage der Motor Konfiguration    |
| SET_POSITION                  | 3<br>(hex)           | Setzen der Motorposition           |
| GO_POSITION                   | 4<br>(hex)           | Anfahren einer bestimmten Position |
| GET_POSITION                  | 5<br>(hex)           | Abfrage einer bestimmten Position  |

| Kommando<br>DAPI_STEPPER_CMD_        | mit<br>Wert<br>(hex) | Bedeutung  |
|--------------------------------------|----------------------|--|
| SET_FREQUENCY                        | 6<br>(hex)           | Einstellung der Motorsollfrequenz                    |
| SET_FREQUENCY_DIRECTLY               | 7<br>(hex)           | Einstellung der Motorfrequenz                        |
| GET_FREQUENCY                        | 8<br>(hex)           | Abfrage der Motorfrequenz                            |
| FULLSTOP                             | 9<br>(hex)           | Sofortiges Anhalten des Motors                       |
| STOP                                 | 10<br>(hex)          | Anhalten des Motors<br>(Bremsrampe wird eingehalten) |
| GO_REFSWITCH                         | 11<br>(hex)          | Anfahren einer Referenzposition                      |
| DISABLE                              | 14<br>(hex)          | Ein-/Ausschalten des Motors                          |
| MOTORCHARACTERISTIC_LOAD<br>_DEFAULT | 15<br>(hex)          | Setzen der Motorcharakteristik<br>auf Defaultwert    |
| MOTORCHARACTERISTIC_EEPR<br>OM_SAVE  | 16<br>(hex)          | Speichern der Motorcharakteristik<br>im EEPROM       |
| MOTORCHARACTERISTIC_EEPR<br>OM_LOAD  | 17<br>(hex)          | Laden der Motorcharakteristik<br>aus dem EEPROM      |

| Kommando<br>DAPI_STEPPER_CMD_ | mit<br>Wert<br>(hex) | Bedeutung                               |
|-------------------------------|----------------------|---|
| GET_CPU_TEMP                  | 18<br>(hex)          | Abfrage der Temperatur des CPU          |
| GET_MOTOR_SUPPLY_VOLTAGE      | 19<br>(hex)          | Abfrage der Versorgungsspannung des CPU |
| GO_POSITION_RELATIVE          | 20<br>(hex)          | Anfahren einer relativen Position       |

### 3.2.7.6.8.2. Werte für par 1 zu Befehl SET\_MOTORCHARACTERISTIC

| Parameter<br>(DAPI_STEPPER_MOTORCHAR_<br>PAR_ ...) | Wert<br>(dez) | Beschreibung  |
|--|---------------|---|
| STEPMODE   | 1             | Stepmode (Full-, 1/2-, 1/4-, 1/8-, 1/16-step)           |
| GOFREQUENCY  | 2             | Geschwindigkeit [Full-step / s] - bezogen auf full-step |
| STARTFREQUENCY                                     | 3             | Startfrequenz [Full-step / s]                           |
| STOPFREQUENCY                                      | 4             | Stopp-Frequenz [Full-step / s]                          |
| MAXFREQUENCY                                       | 5             | Maximale Frequenz [Full-step / s]                       |
| ACCELERATIONSLOPE                                  | 6             | Steigung der Beschleunigung [Full-step / ms]            |
| DECELERATIONSLOPE                                  | 7             | Steigung der Verzögerung [Full-step / 10ms]             |
| PHASECURRENT                                       | 8             | Phasenstrom [mA]  |
| HOLDPHASECURRENT                                   | 9             | Phasenstrom für Motorhaltung [mA]                       |
| HOLDTIME   | 10            | Zeit, in der der Halt zum Motorstopp geht [ms]          |
| STATUSLEDMODE                                      | 11            | Modus der Status-LED                                    |
| INVERT_ENDSW1                                      | 12            | umkehren des Endschalter1                               |
| INVERT_ENDSW2                                      | 13            | umkehren des Endschalter2                               |
| INVERT_REFSW1                                      | 14            | umkehren des Frequenzschalters1                         |

| Parameter<br>(DAPI_STEPPER_MOTORCHAR_<br>PAR_ ...) | Wert<br>(dez) | Beschreibung  |
|--|---------------|---|
| INVERT_REFSW2                                      | 15            | umkehren des<br>Frequenzschalters1                          |
| INVERT_DIRECTION                                   | 16            | Alle Richtungsangaben<br>umkehren                           |
| ENDSWITCH_STOPMODE                                 | 17            | Einstellung des<br>Stoppverhaltens (0=Fullstop /<br>1=Stop) |

| Parameter<br>(DAPI_STEPPER_MOTORCHAR_<br>PAR_ ...) | Wert<br>(dez) | Beschreibung   |
|--|---------------|--|
| GOREFERENCEFREQUENCY_TOEND<br>SWITCH               | 18            | Frequenz vor Endschalter [Full-<br>step / s]             |
| GOREFERENCEFREQUENCY_AFTER<br>ENDSWITCH            | 19            | Frequenz nach Endschalter<br>[Full-step / s]             |
| GOREFERENCEFREQUENCY_TOOFF<br>SET                  | 20            | Frequenz bis zum optionalen<br>Offset<br>[Full-step / s] |



### 3.2.7.6.8.3. Werte für par 1 zu Befehl GO\_REFSWITCH

### 3.2.7.6.8.4. Beispiel

#### Programmbeispiel

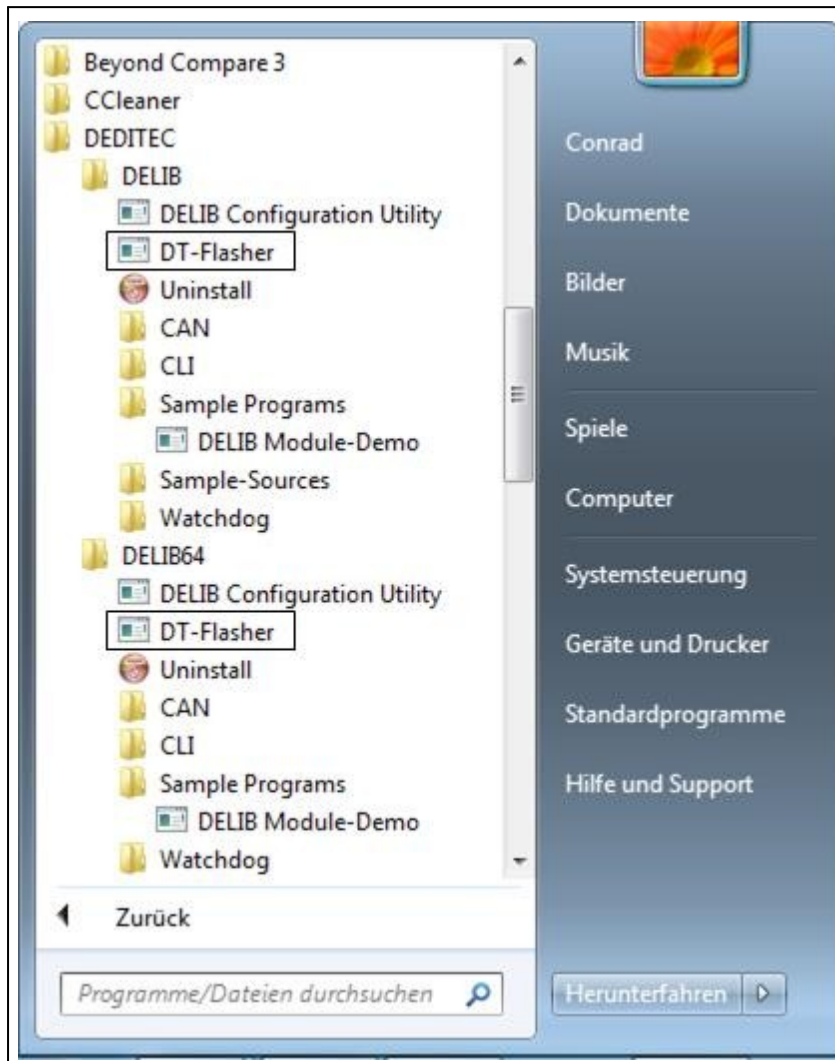
```
DapiStepperCommand(handle, 1,  
DAPI_STEPPER_CMD_GO_POSITION, 3200, 0, 0, 0);
```

wird in einem 8 Byte CAN Paket versendet.

Das Paket hat folgende Struktur:

| CAN-Byte | Type             | Wert | Byte |
|----------|------------------|------|------|
| 1        | COMMAND          | 4    | 04   |
| 2        | PAR1 (Bit 0-7)   | 3200 | 80   |
| 3        | PAR1 (Bit 8-15)  |      | 0C   |
| 4        | PAR1 (Bit 16-23) |      | 00   |
| 5        | PAR1 (Bit 24-31) |      | 00   |
| 6        | PAR2 (Bit 0-7)   | 0    | 00   |
| 7        | PAR2 (Bit 8-15)  |      | 00   |
| 8        | PAR3 (Bit 0-7)   | 0    | 00   |

### 3.2.8. DT-Flasher



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

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

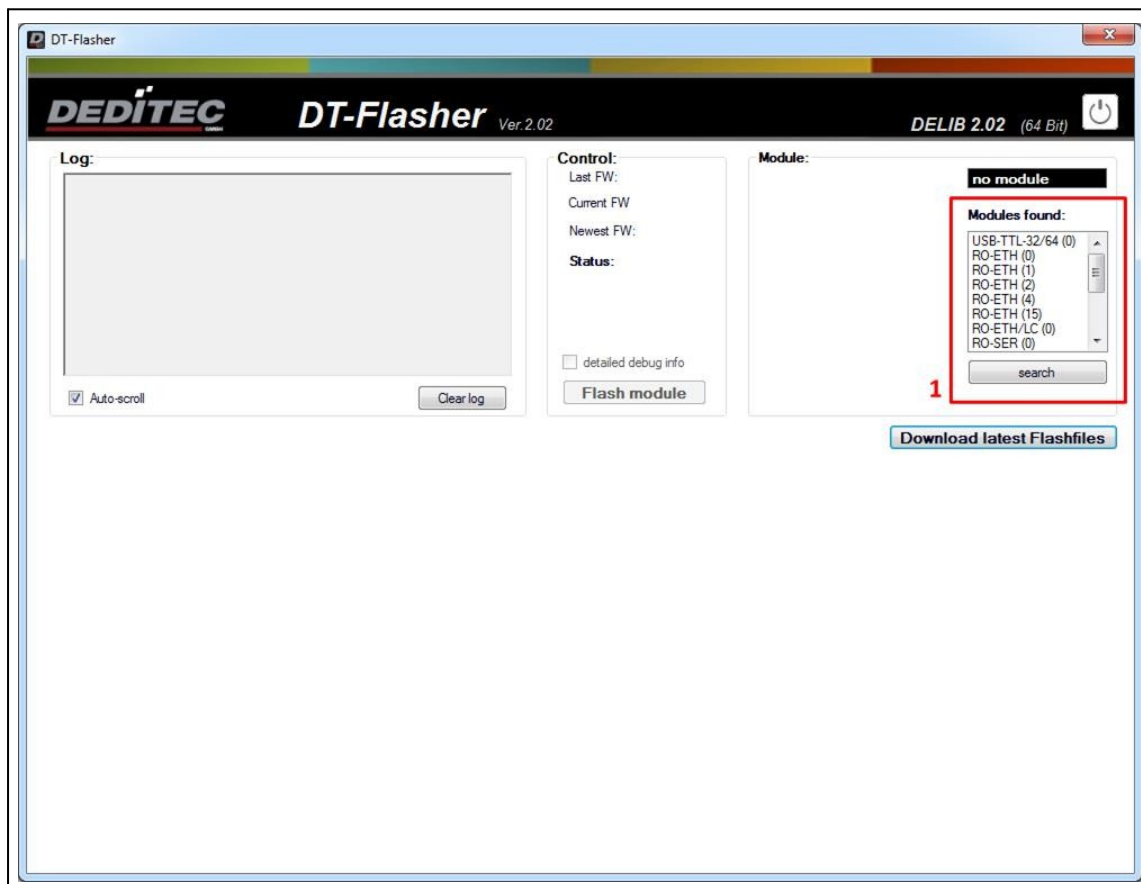
### 3.2.8.1. Über DEDITEC-Firmware

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

#### Hinweis:

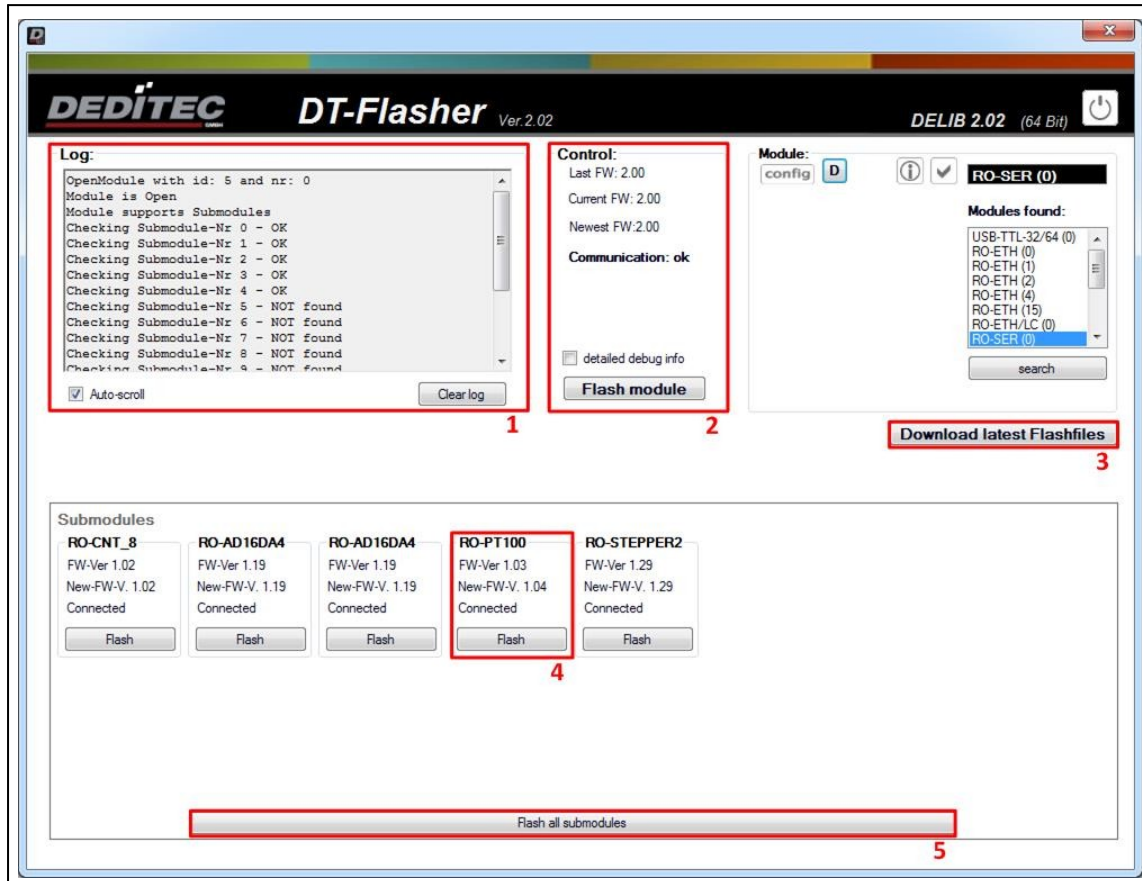
Da neue Firmware-Versionen in der Regel neue Funktionen für Ihr Produkt "freischalten", empfehlen wir daher ein regelmäßiges Firmware-Update Ihrer DEDITEC Produkte.

### 3.2.8.2. Auswahl des Moduls



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

### 3.2.8.3. Firmware Update durchführen



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

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

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

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

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

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

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

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

##### **Schritt 1**

Downloaden Sie die aktuellste Version der Flash-Files unter

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

##### **Schritt 2**

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

x86

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

x64

C:\Program Files\DEDITEC\DELIB\programs

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

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

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

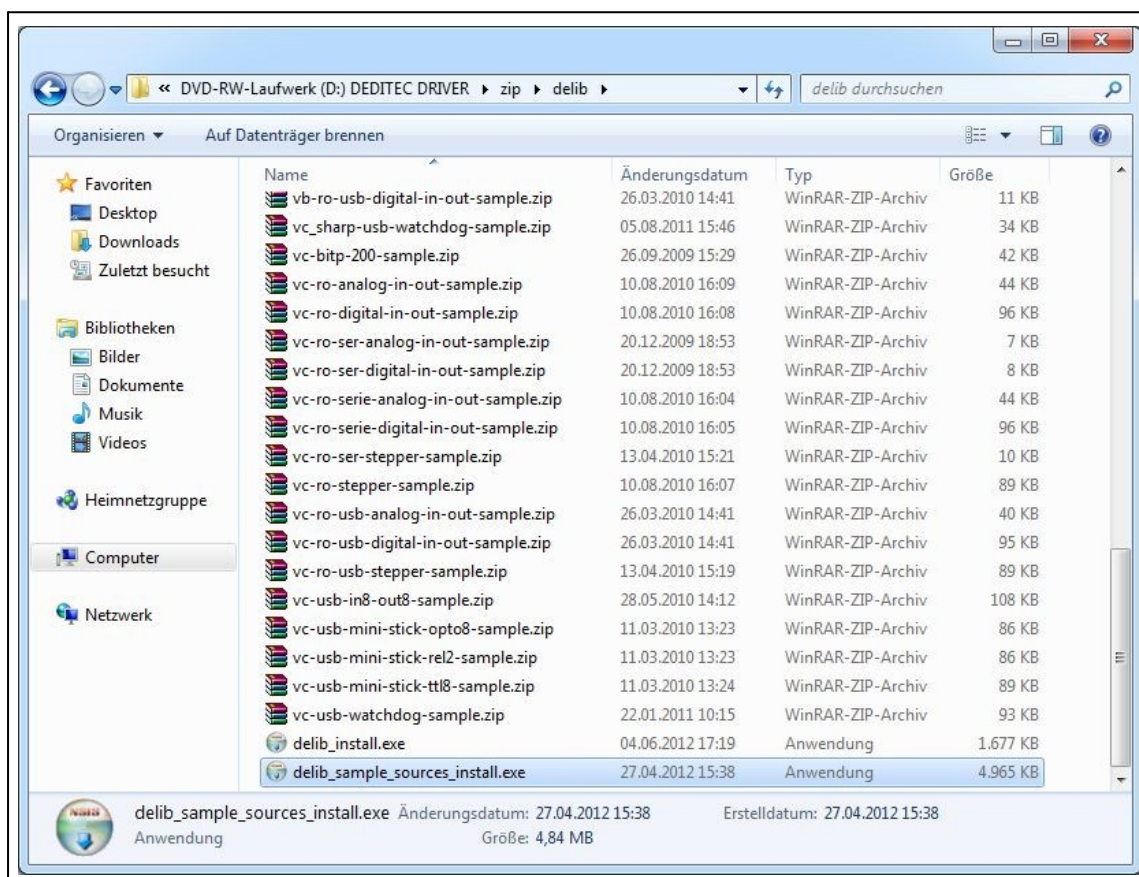
- C
- C++
- C#
- Delphi
- VisualBasic
- VB.NET
- MS-Office
- LabVIEW
- Java

### 3.3.1. Installation DELIB Sample Sources

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

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

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

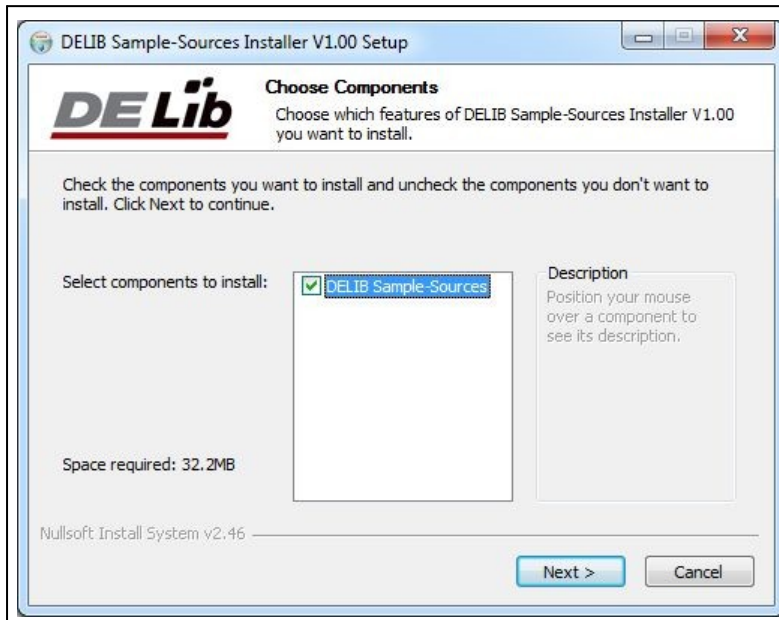




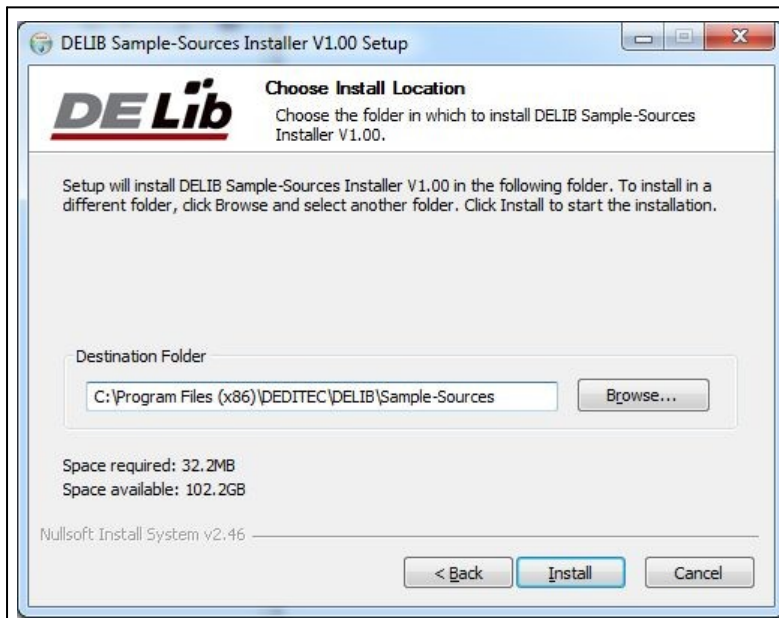
Startbild des DELIB Sample Sources Installer



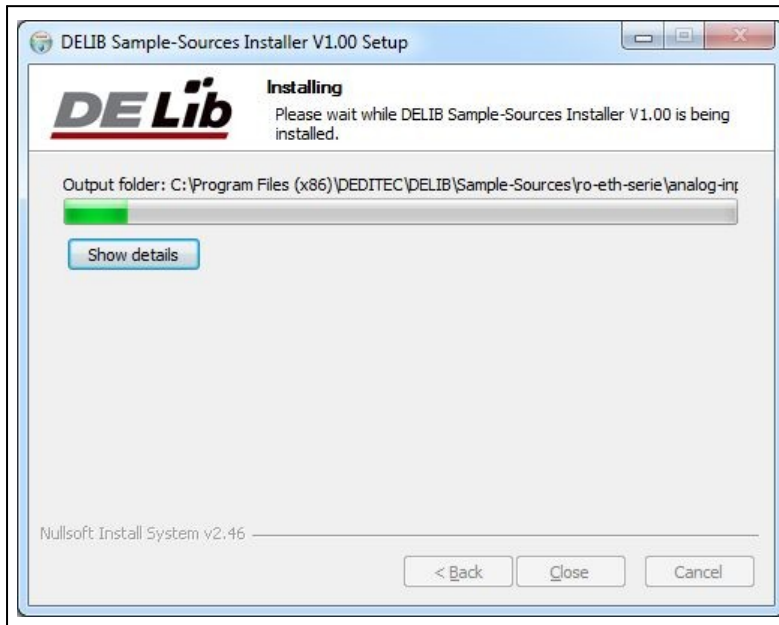
Drücken Sie Next.



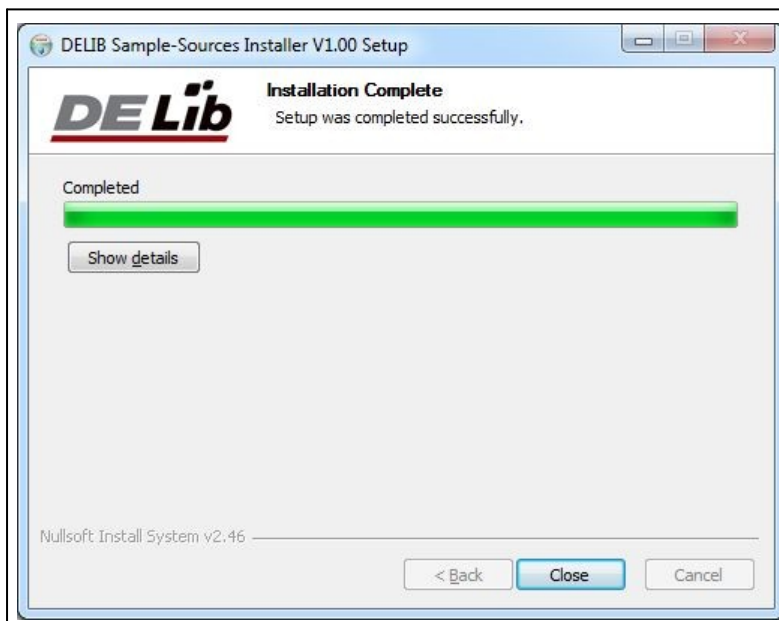
Wählen Sie den Installationsordner und drücken Sie Install.



Die DELIB Sample Sources werden nun installiert.



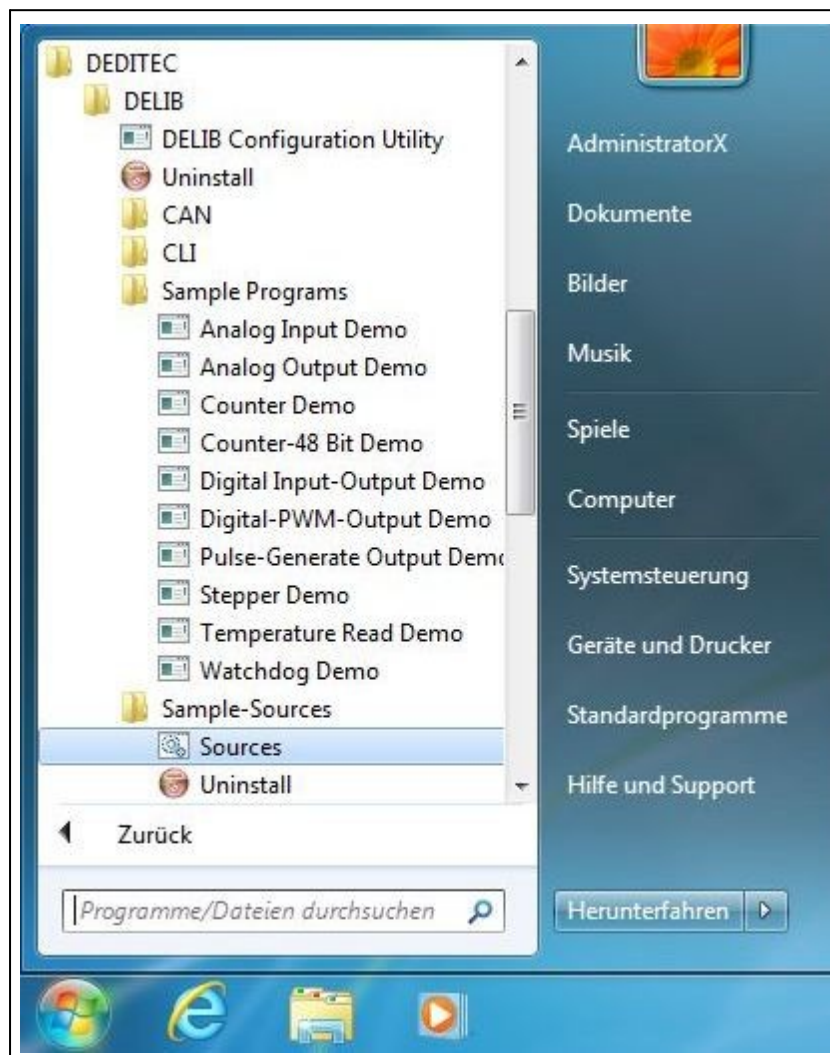
Die DELIB Sample Sources wurden erfolgreich installiert. Drücken Sie Close um die Installation zu beenden.



### 3.3.2. Benutzung der DELIB Sample Sources

Nach Installation der DELIB Sample Sources finden Sie diese unter

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

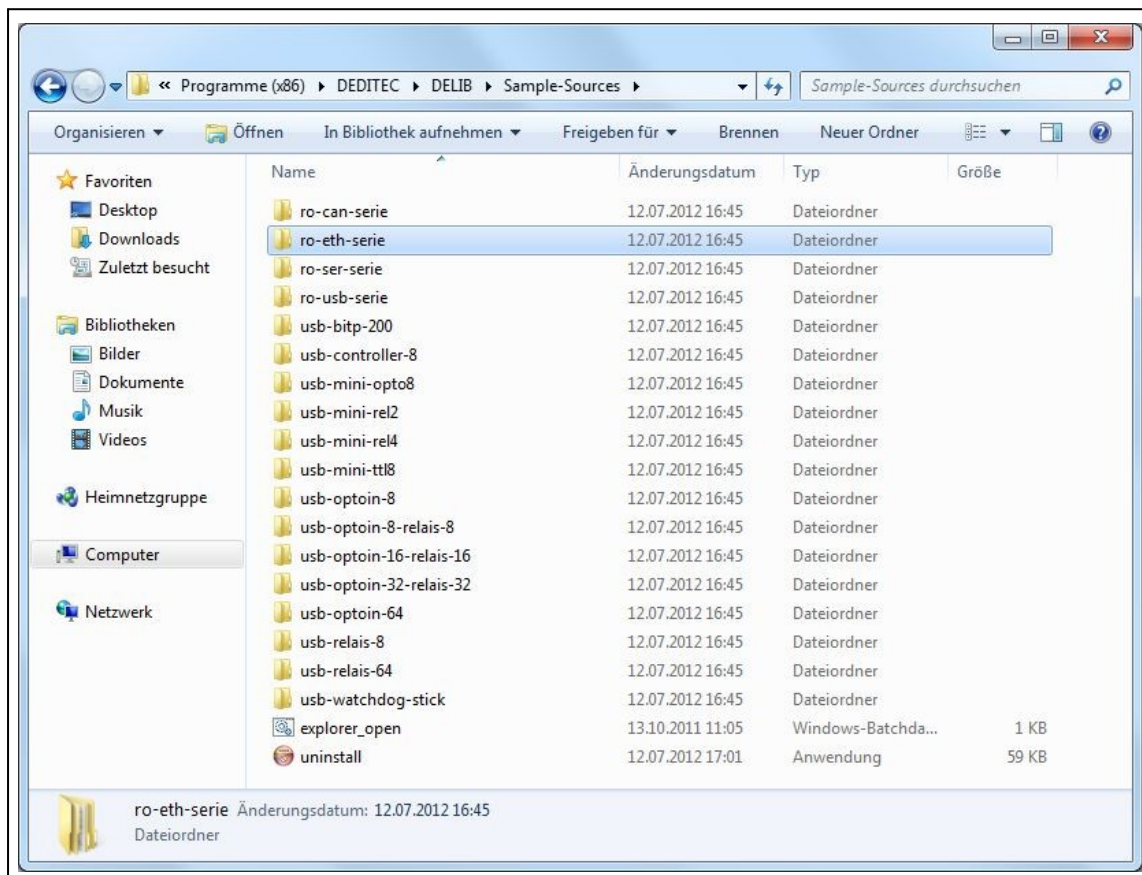


Nun öffnet sich der Windows-Explorer mit einer Übersicht aller Produkte für die ein Beispielprogramm verfügbar ist.

### 3.3.2.1. Schritt 1 - Produktauswahl

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

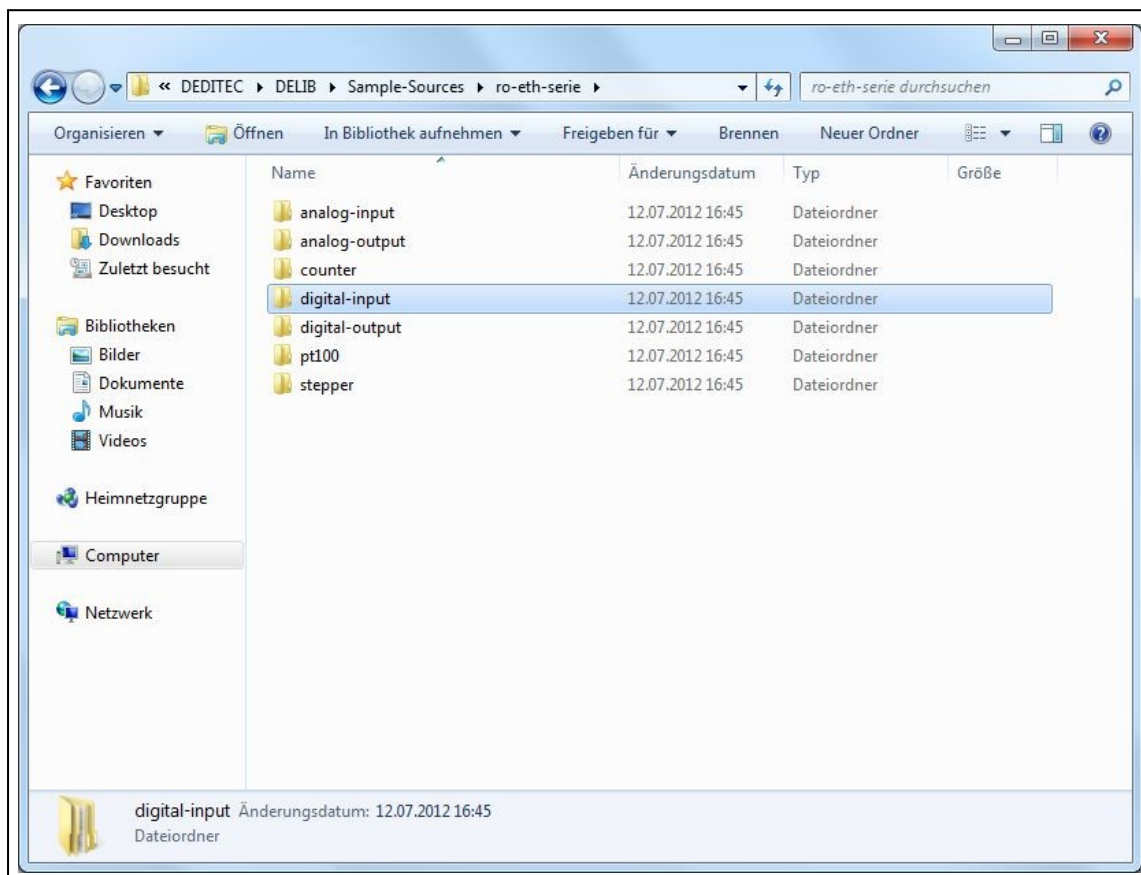
Da es sich um ein RO-ETH-Produkt handelt, wählen bzw. öffnen Sie den Ordner ro-eth-serie.



### 3.3.2.2. Schritt 2 - Kategorieauswahl

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

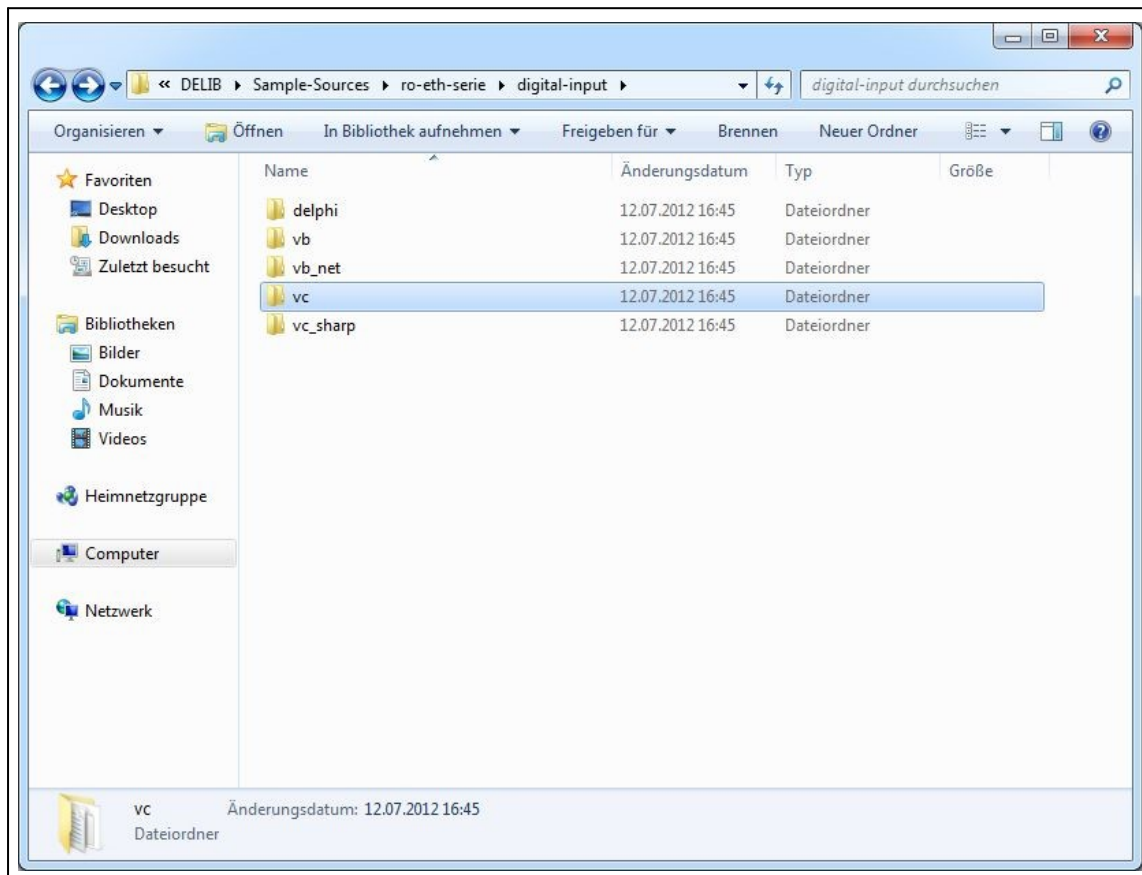
Da wir uns in diesem Beispiel auf die digitalen Eingänge konzentrieren, wählen Sie die Kategorie digital-input



### 3.3.2.3. Schritt 3 - Programmiersprachenauswahl

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

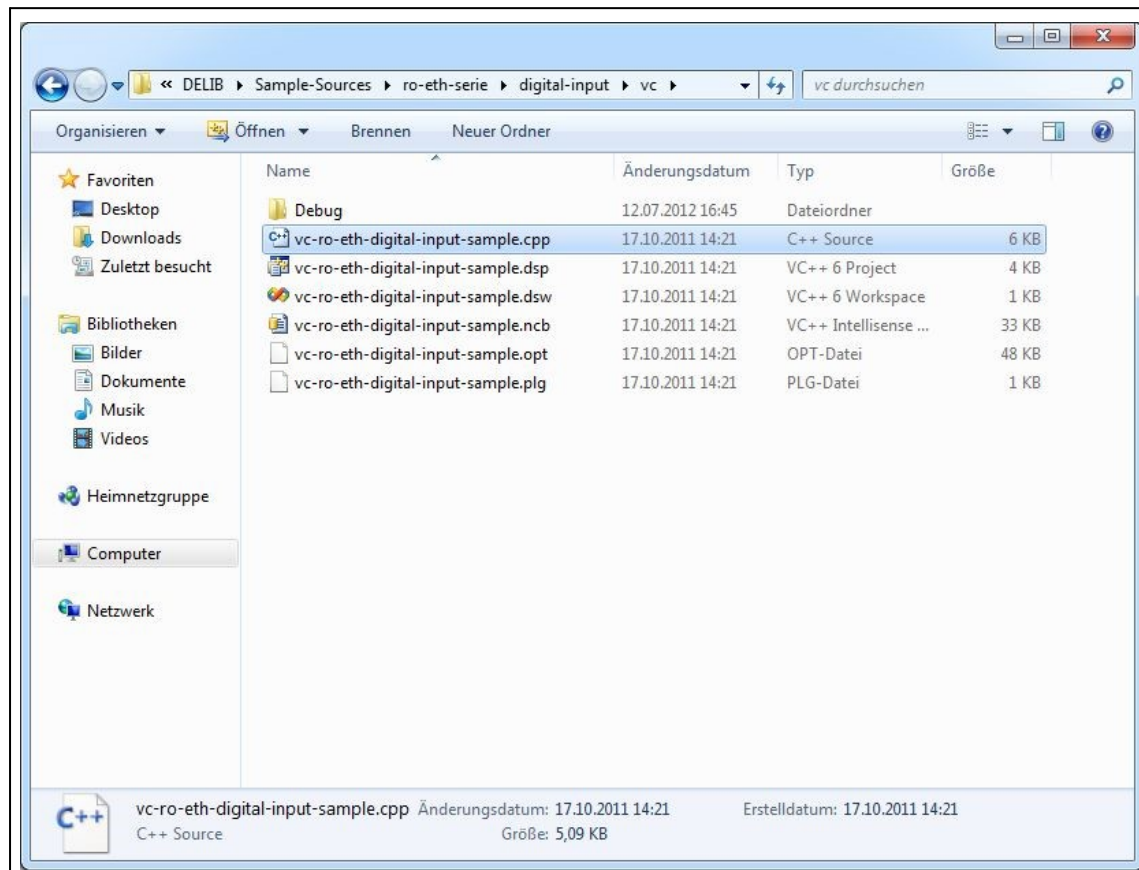
Da wir uns in diesem Beispiel auf die Programmiersprache Visual-C konzentrieren, öffnen Sie den Ordner vc.





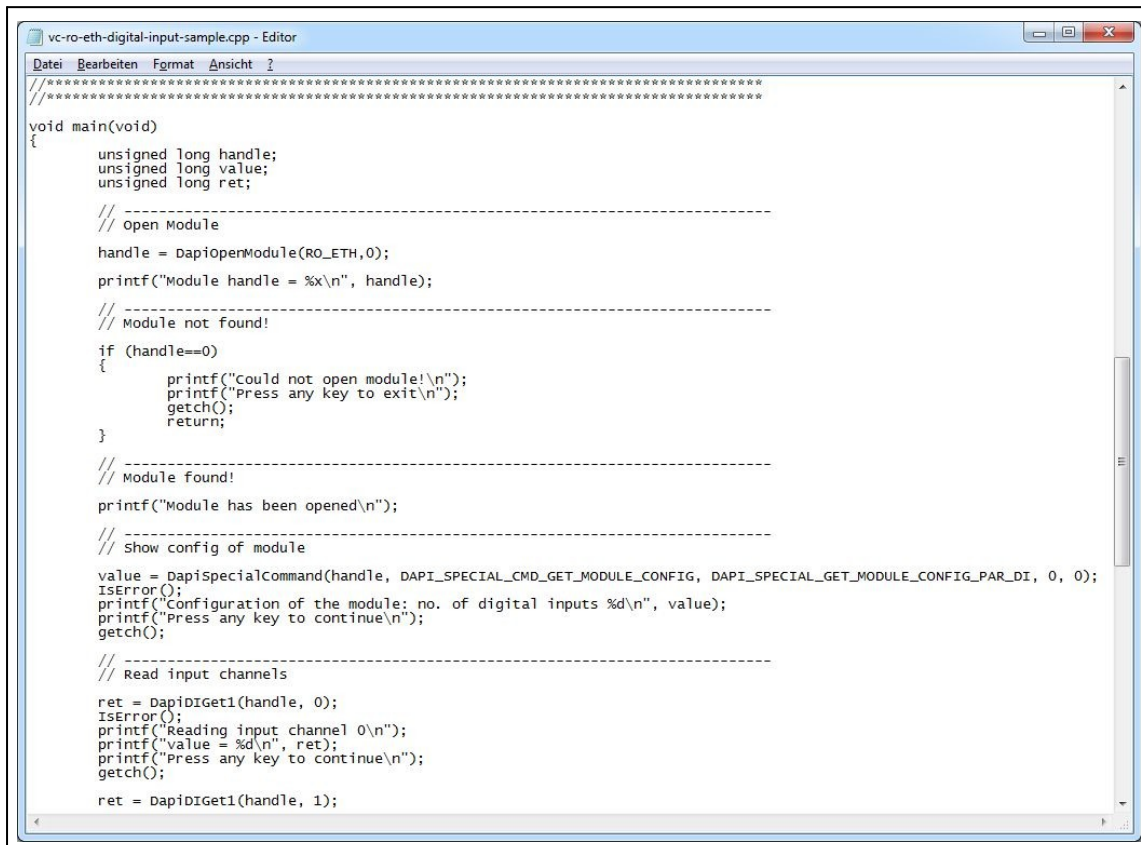
### 3.3.2.4. Schritt 4 - Quellcode

Nach Auswahl der Programmiersprache erhalten Sie folgende Übersicht:





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



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

    // -----
    // Open Module

    handle = DapiOpenModule(RO_ETH,0);
    printf("Module handle = %x\n", handle);

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

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

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

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

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

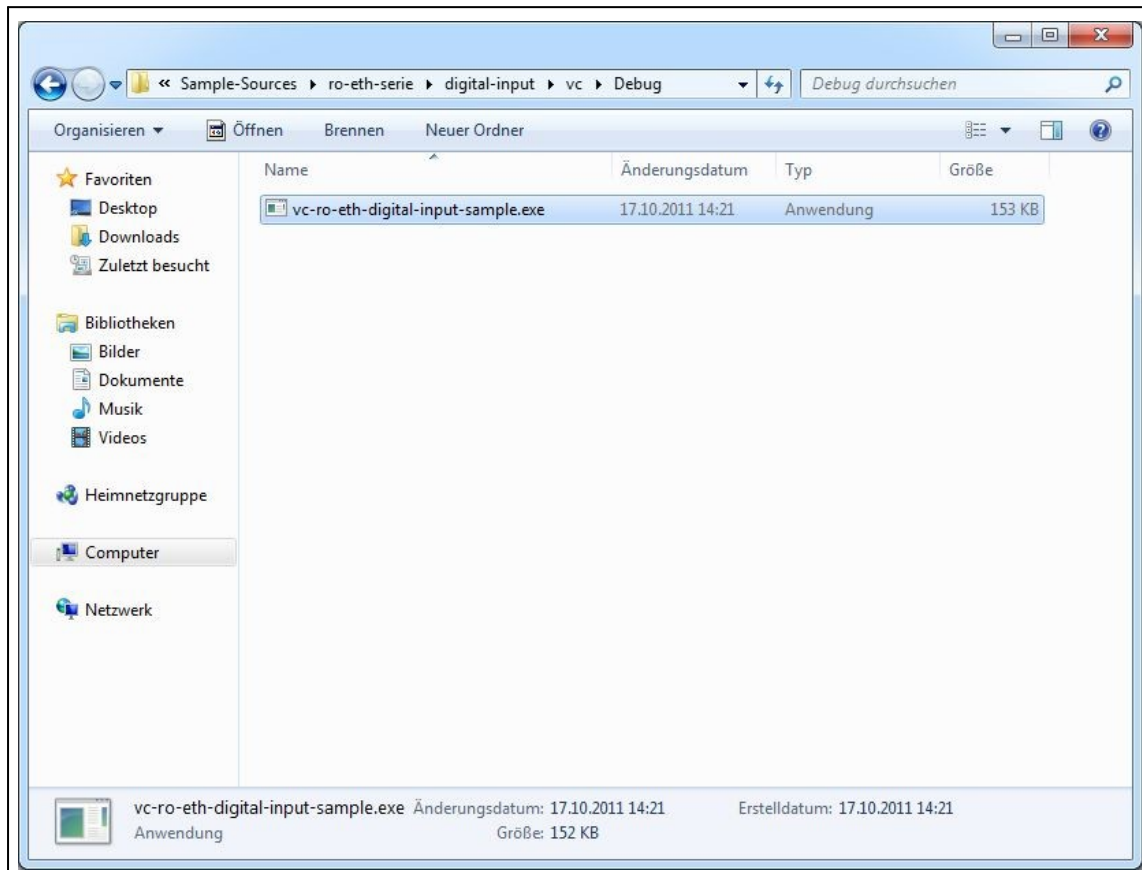
    value = DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_GET_MODULE_CONFIG, DAPI_SPECIAL_GET_MODULE_CONFIG_PAR_DI, 0, 0);
    ISERROR();
    printf("Configuration of the module: no. of digital inputs %d\n", value);
    printf("Press any key to continue\n");
    getch();

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

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

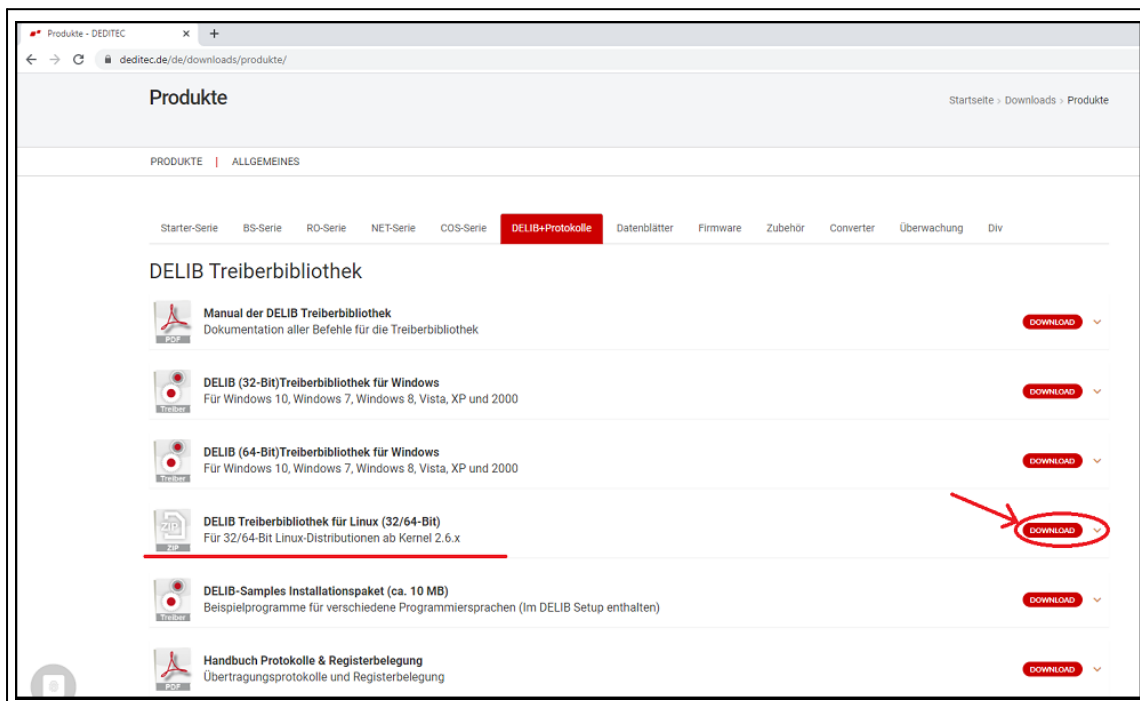
    ret = DapiDGetI(handle, 1);
}
```

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

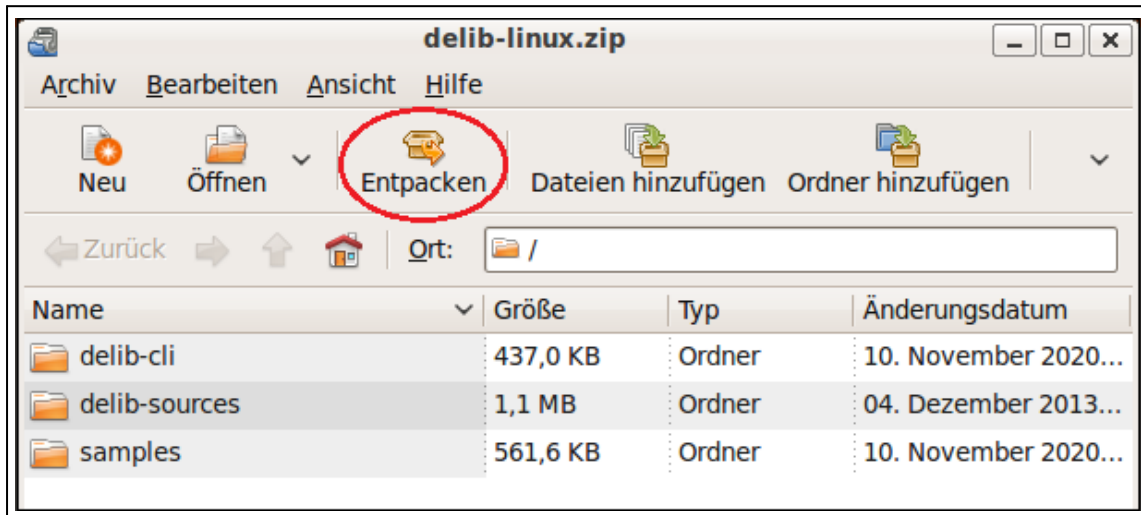


### 3.4. DELIB für Linux

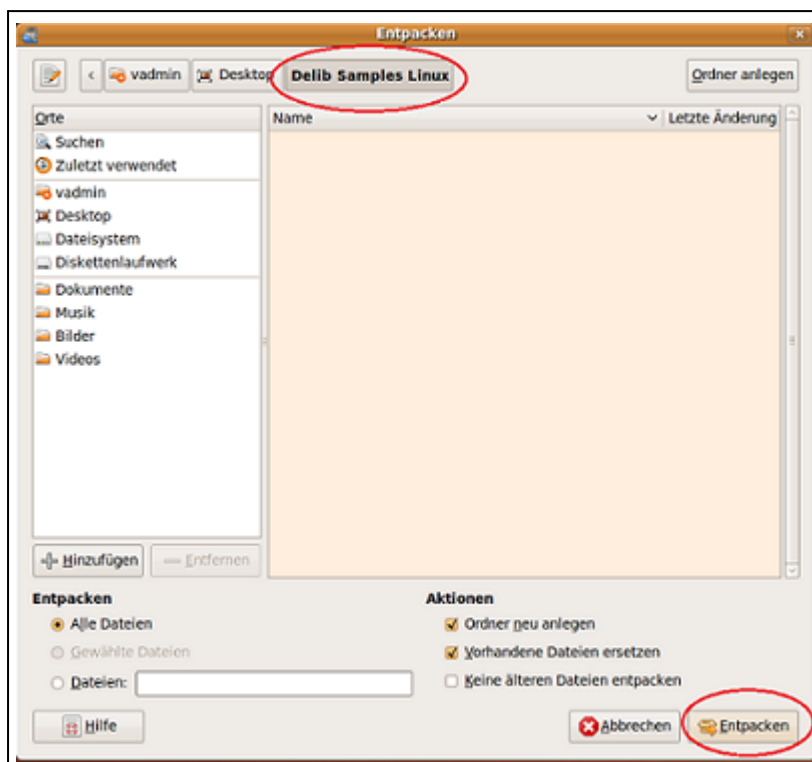
Laden Sie sich die Delib Linux Treiberbibliothek unter "[www.deditec.de/de/downloads/produkte/](http://www.deditec.de/de/downloads/produkte/)" im Reiter „DELIB+Protokolle“ oder unter "[www.deditec.de/media/zip/delib/delib-linux.zip](http://www.deditec.de/media/zip/delib/delib-linux.zip)" direkt auf ihr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



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



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



### 3.4.1. DELIB Sample Sources (Windows Programmbeispiele) 2

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

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

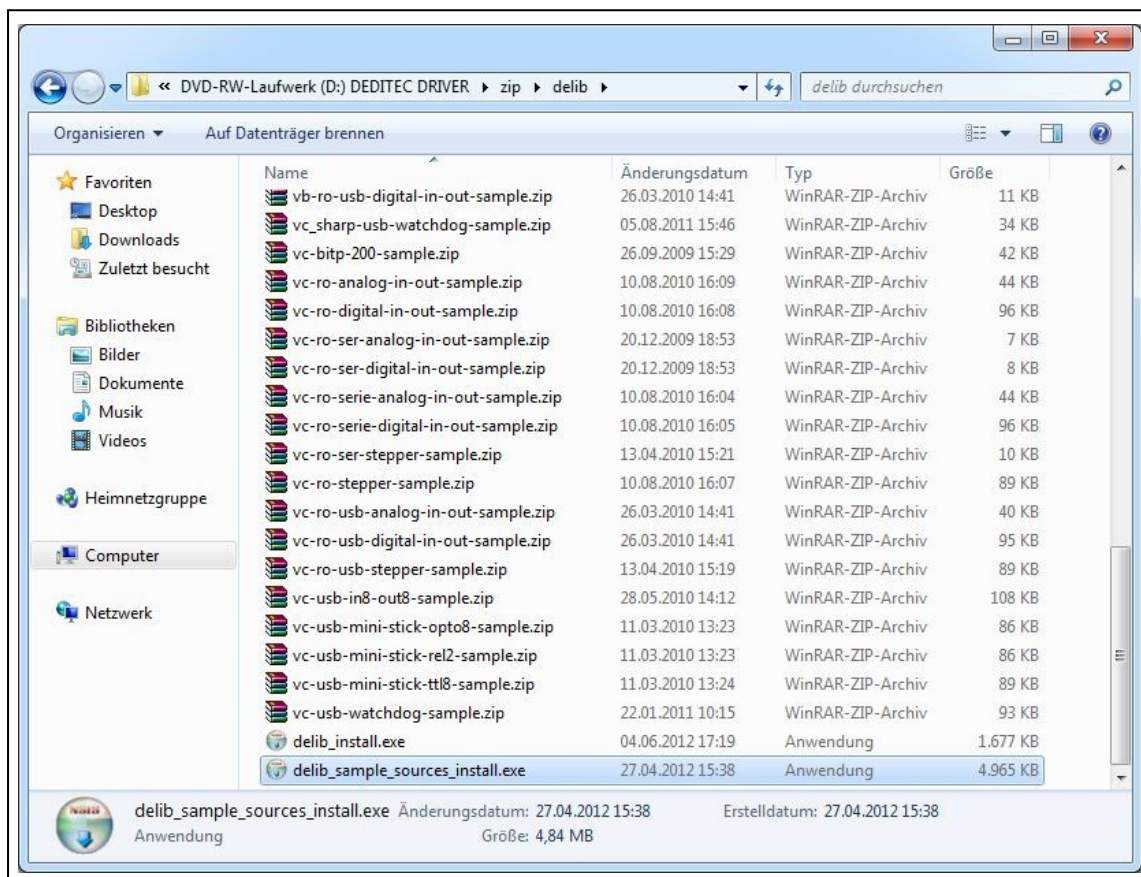
- C
- C++
- C#
- Delphi
- VisualBasic
- VB.NET
- MS-Office
- LabVIEW
- Java

### 3.4.1.1. Installation DELIB Sample Sources 2

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

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

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

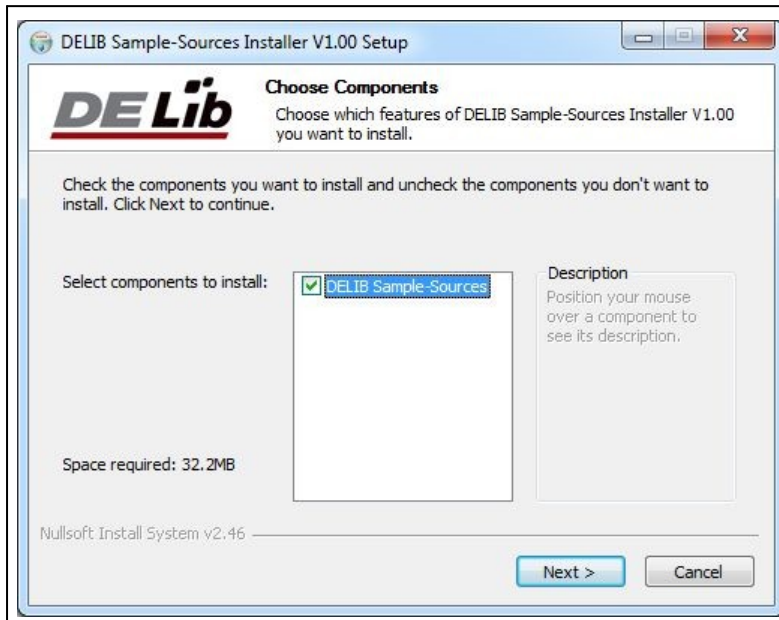


Startbild des DELIB Sample Sources Installer

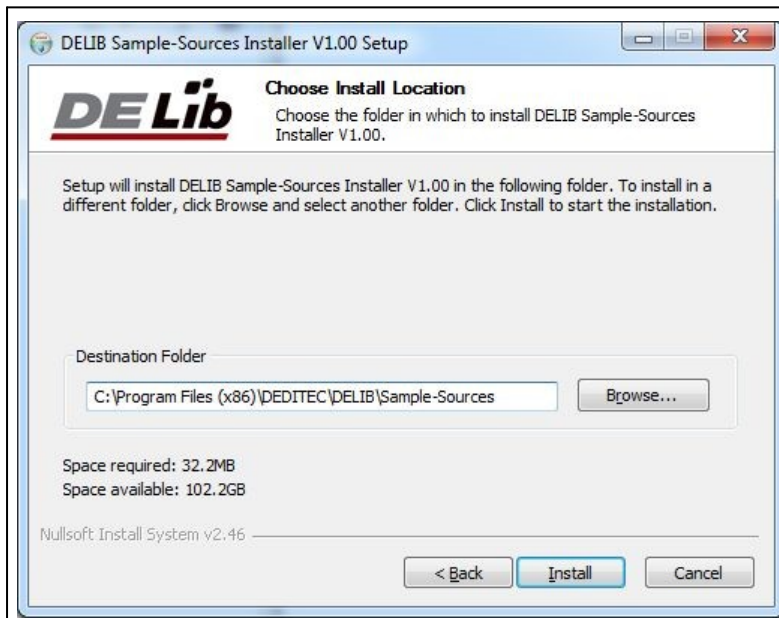




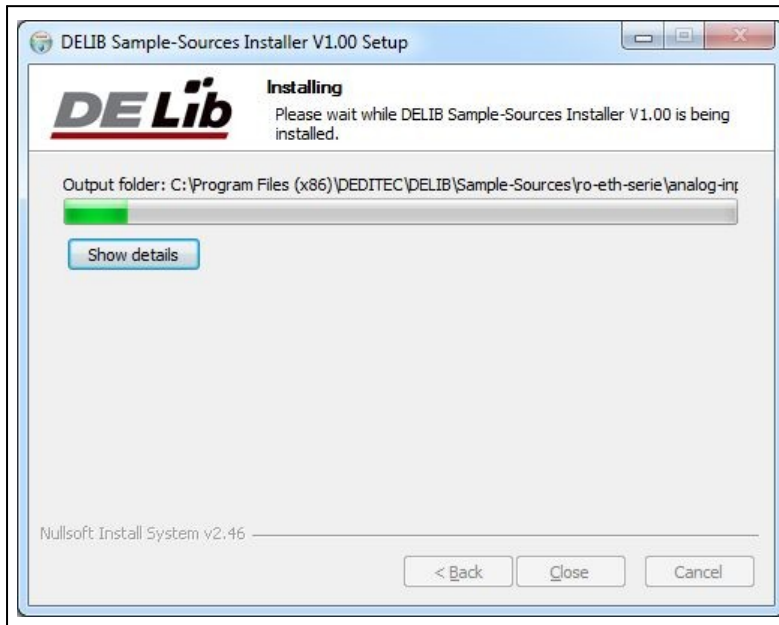
Drücken Sie Next.



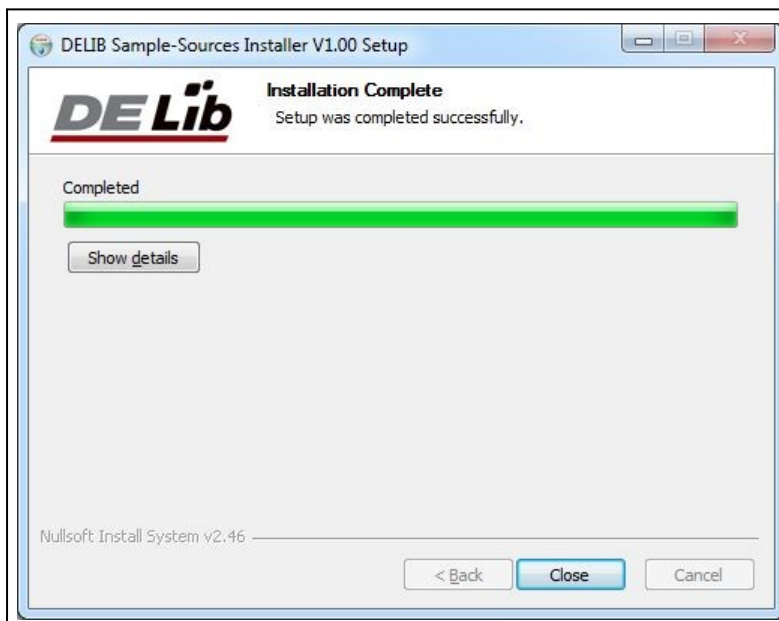
Wählen Sie den Installationsordner und drücken Sie Install.



Die DELIB Sample Sources werden nun installiert.



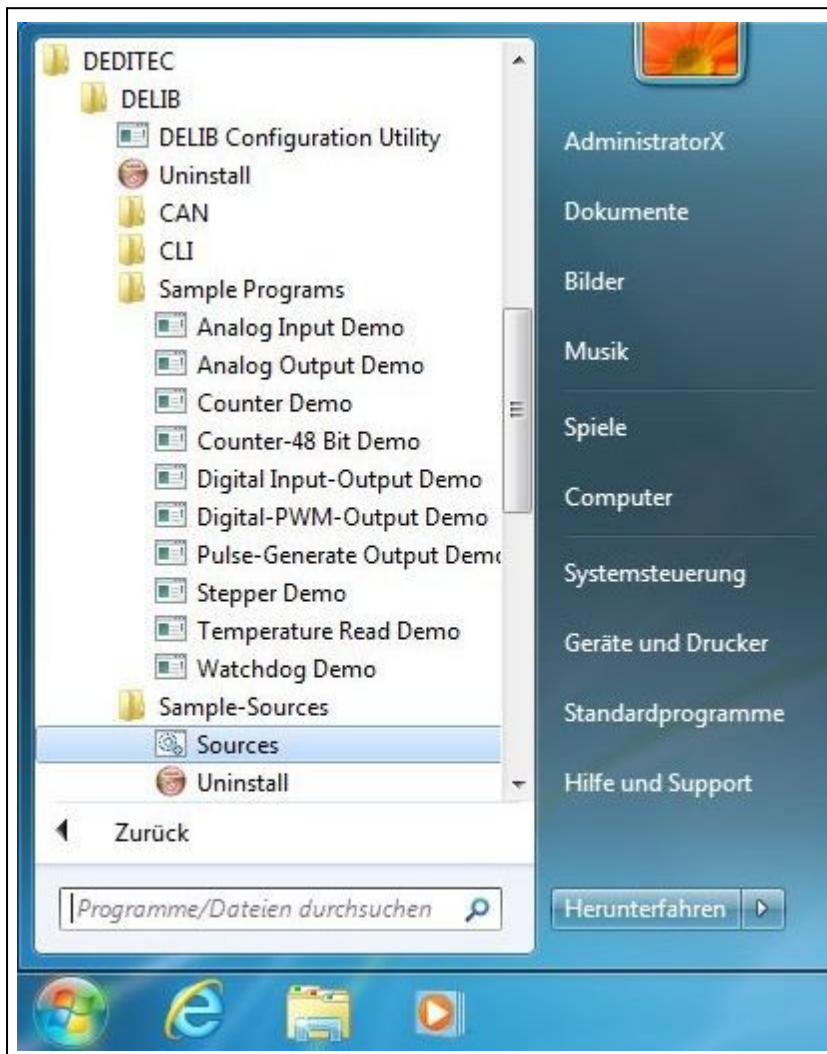
Die DELIB Sample Sources wurden erfolgreich installiert. Drücken Sie Close um die Installation zu beenden.



### 3.4.1.2. Benutzung der DELIB Sample Sources 2

Nach Installation der DELIB Sample Sources finden Sie diese unter

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

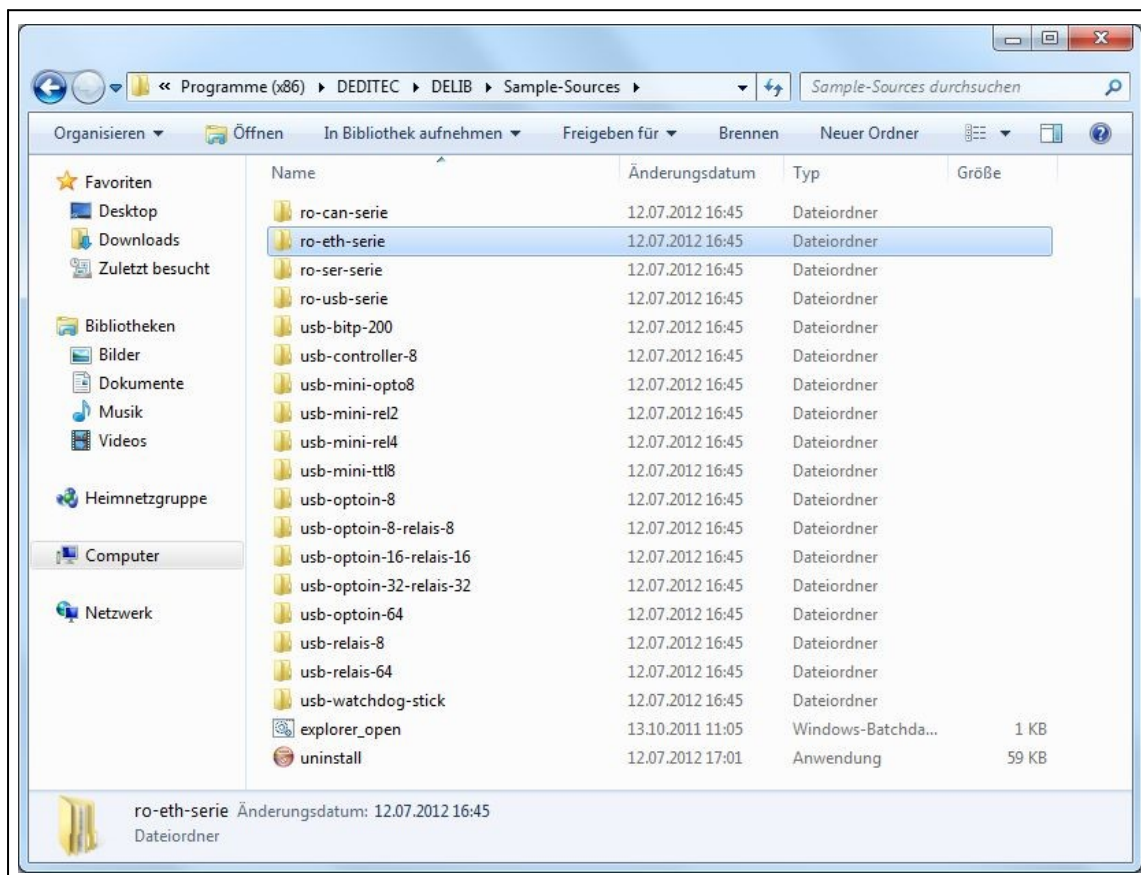


Nun öffnet sich der Windows-Explorer mit einer Übersicht aller Produkte für die ein Beispielprogramm verfügbar ist.

### 3.4.1.2.1. Schritt 1 - Produktauswahl 2

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

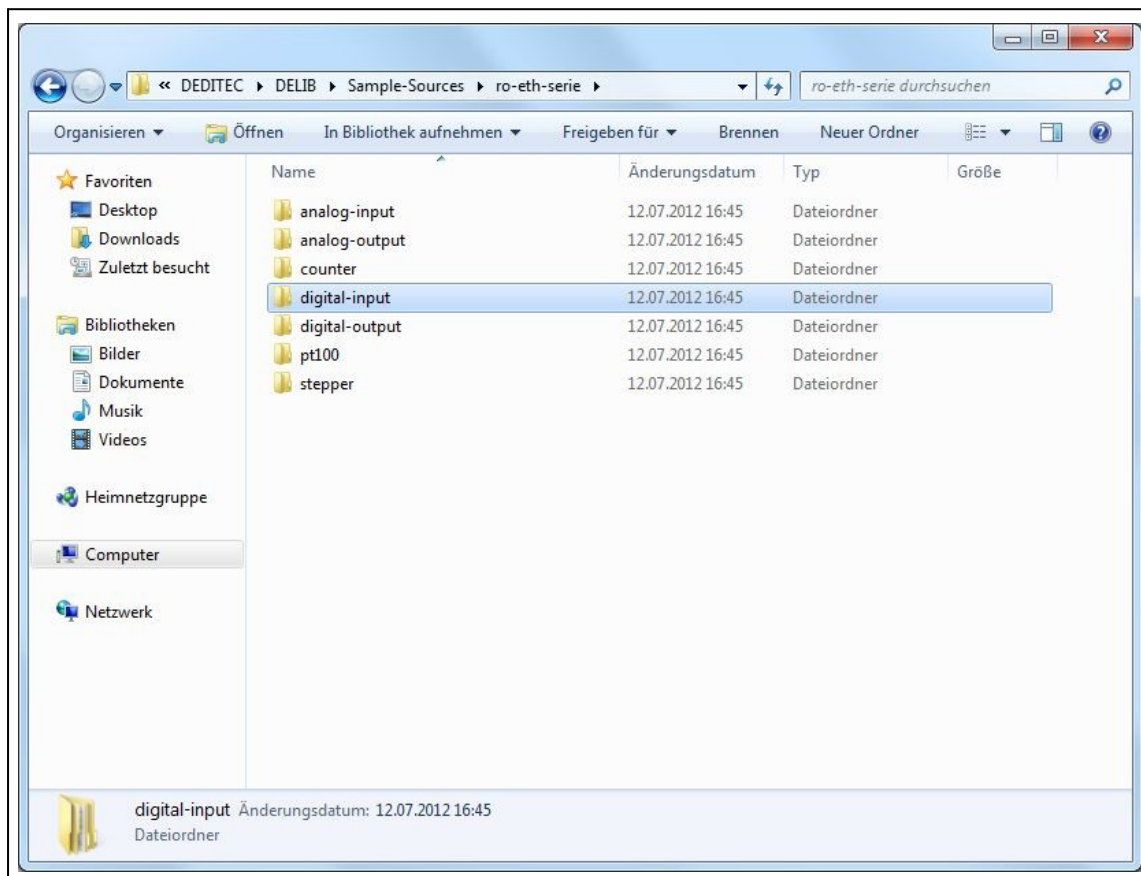
Da es sich um ein RO-ETH-Produkt handelt, wählen bzw. öffnen Sie den Ordner ro-eth-serie.



### 3.4.1.2.2. Schritt 2 - Kategorieauswahl 2

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

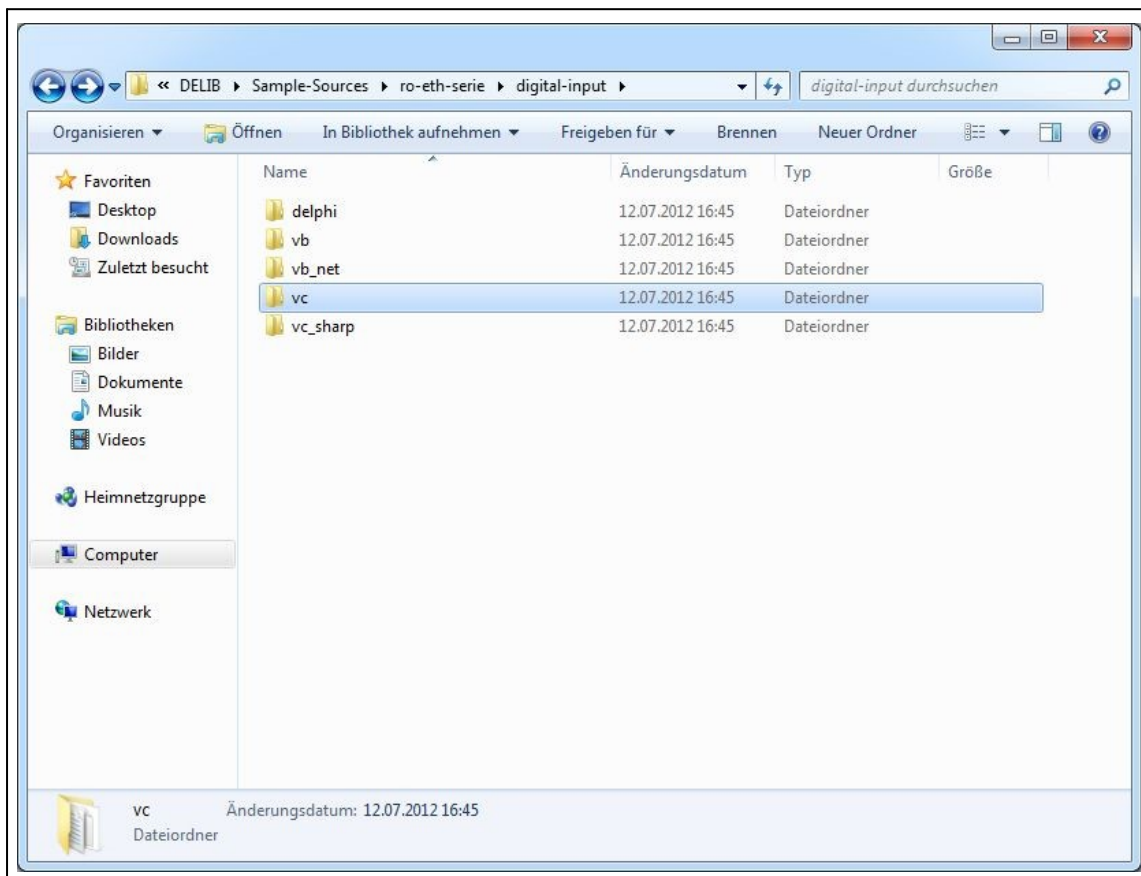
Da wir uns in diesem Beispiel auf die digitalen Eingänge konzentrieren, wählen Sie die Kategorie digital-input



### 3.4.1.2.3. Schritt 3 - Programmiersprachenauswahl 2

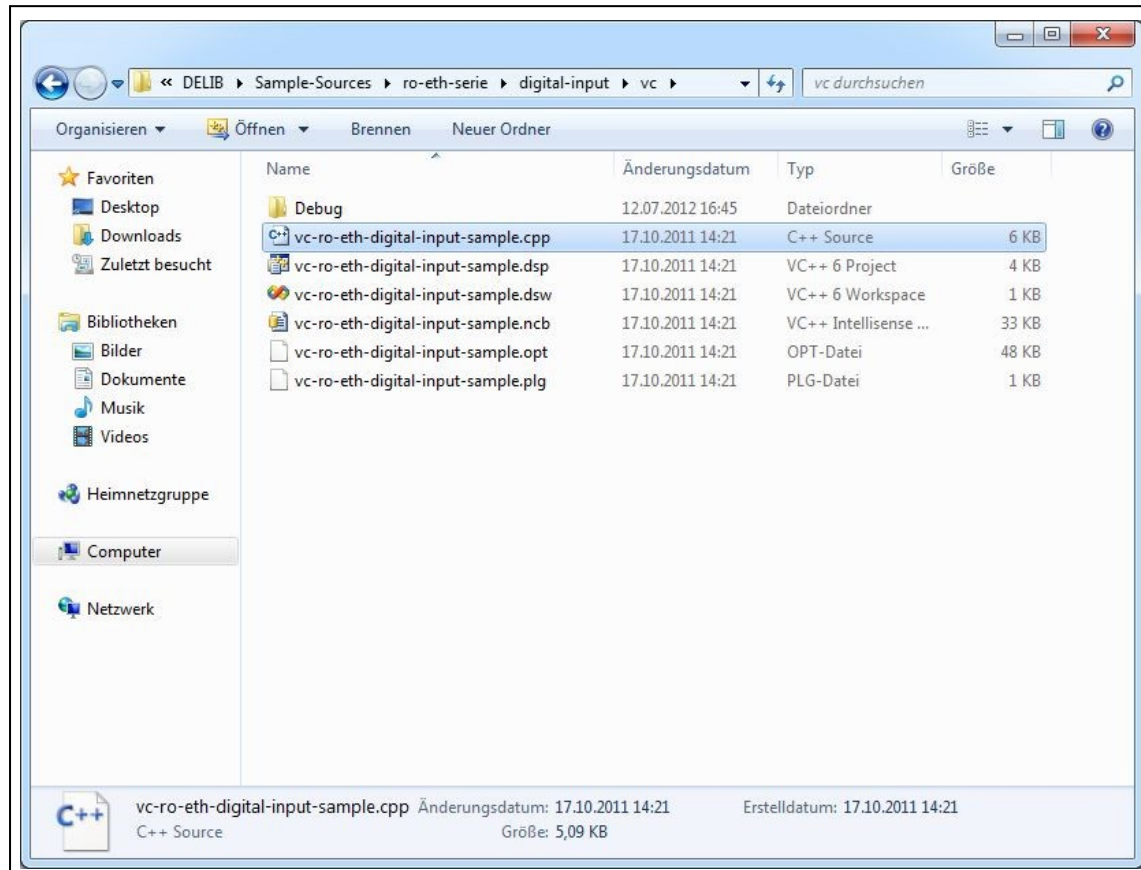
In diesem Schritt sehen Sie alle verfügbaren Programmierbeispiele der gewählten Kategorie, sortiert nach Programmiersprachen.

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



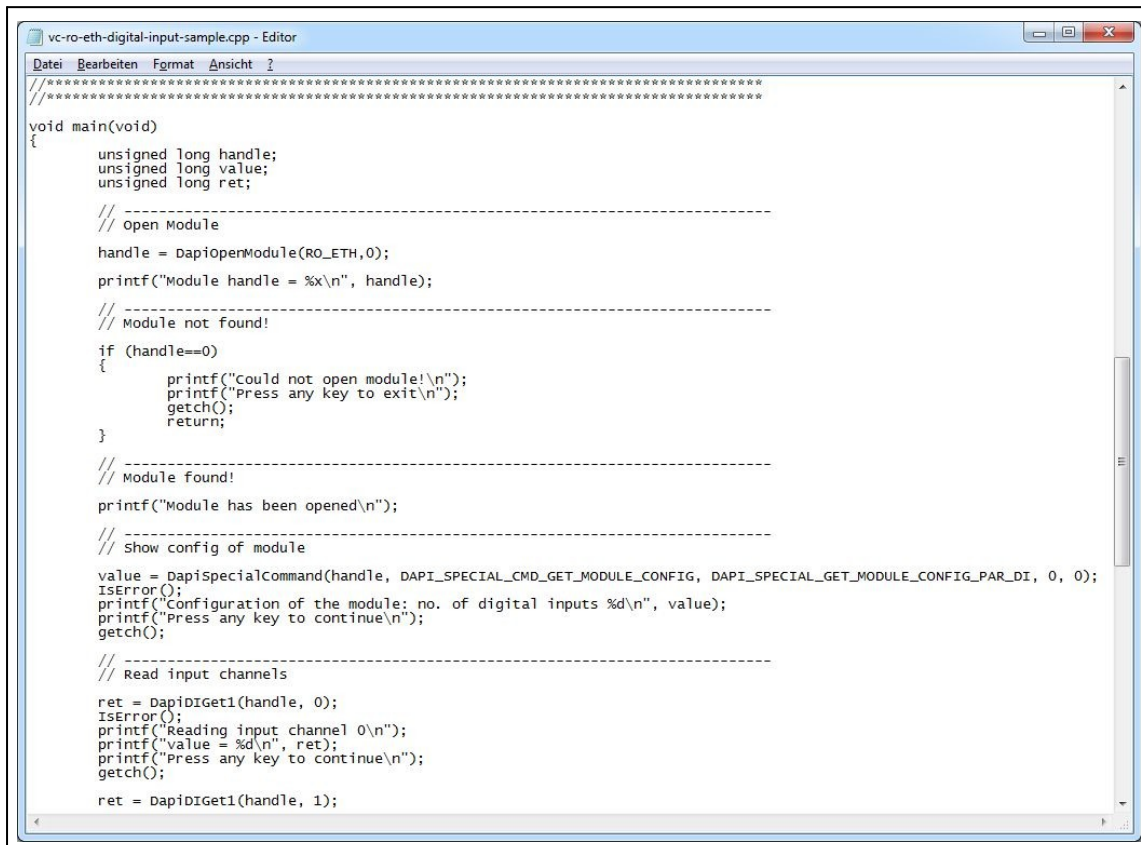
#### 3.4.1.2.4. Schritt 4 - Quellcode 2

Nach Auswahl der Programmiersprache erhalten Sie folgende Übersicht:





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



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

    // -----
    // Open Module

    handle = DapiOpenModule(RO_ETH,0);
    printf("Module handle = %x\n", handle);

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

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

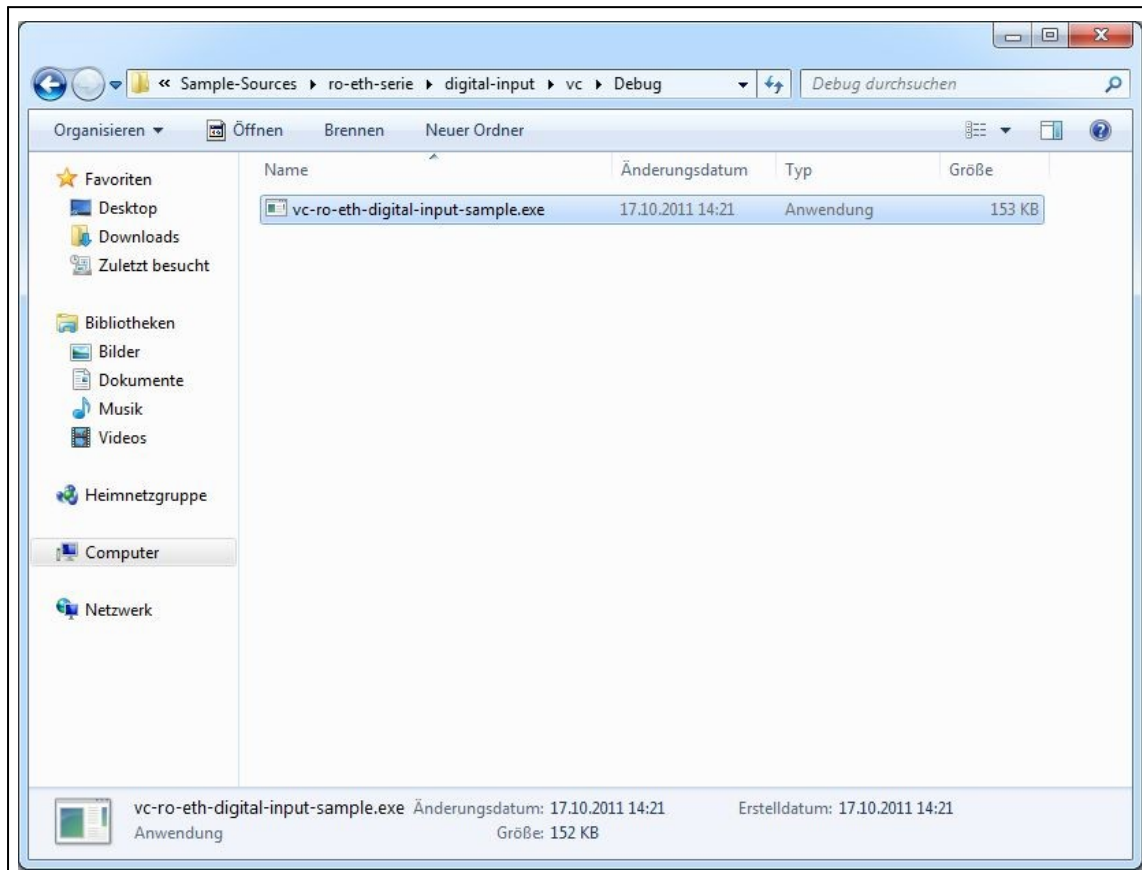
    // -----
    // Show config of module
    value = DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_GET_MODULE_CONFIG, DAPI_SPECIAL_GET_MODULE_CONFIG_PAR_DI, 0, 0);
    if (value < 0)
    {
        printf("Configuration of the module: no. of digital inputs %d\n", value);
        printf("Press any key to continue\n");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

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

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



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



## 3.4.2. Verwenden der DELIB Treiberbibliothek für Linux

### 3.4.2.1. Delib USB-Sample in Linux

#### Voreinstellungen

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

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

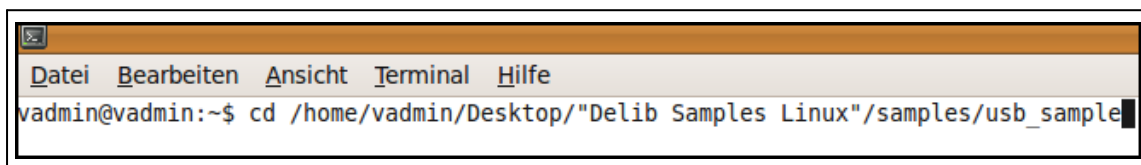
```
23
24 #include <stdio.h>
25 #include <stdlib.h>
26 #include <unistd.h>
27
28 #include "../delib-sources/delib/library/delib/delib.h"
29
30 int main()
31 {
32     ULONG i;
33     ULONG handle=0;
34
35     printf("\n\n");
36     printf("-----\n");
37     printf("-----\n");
38     printf("-----\n");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

## Kompilieren des USB-Samples

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

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

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

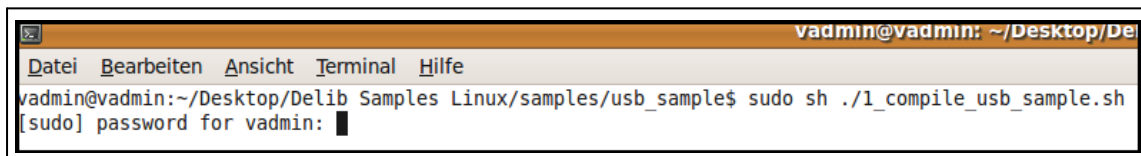


```
Terminal
Datei Bearbeiten Ansicht Terminal Hilfe
vadmin@vadmin:~$ cd /home/vadmin/Desktop/"Delib Samples Linux"/samples/usb_sample
```

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

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

Geben Sie, falls nötig, Ihr Benutzerkennwort ein.



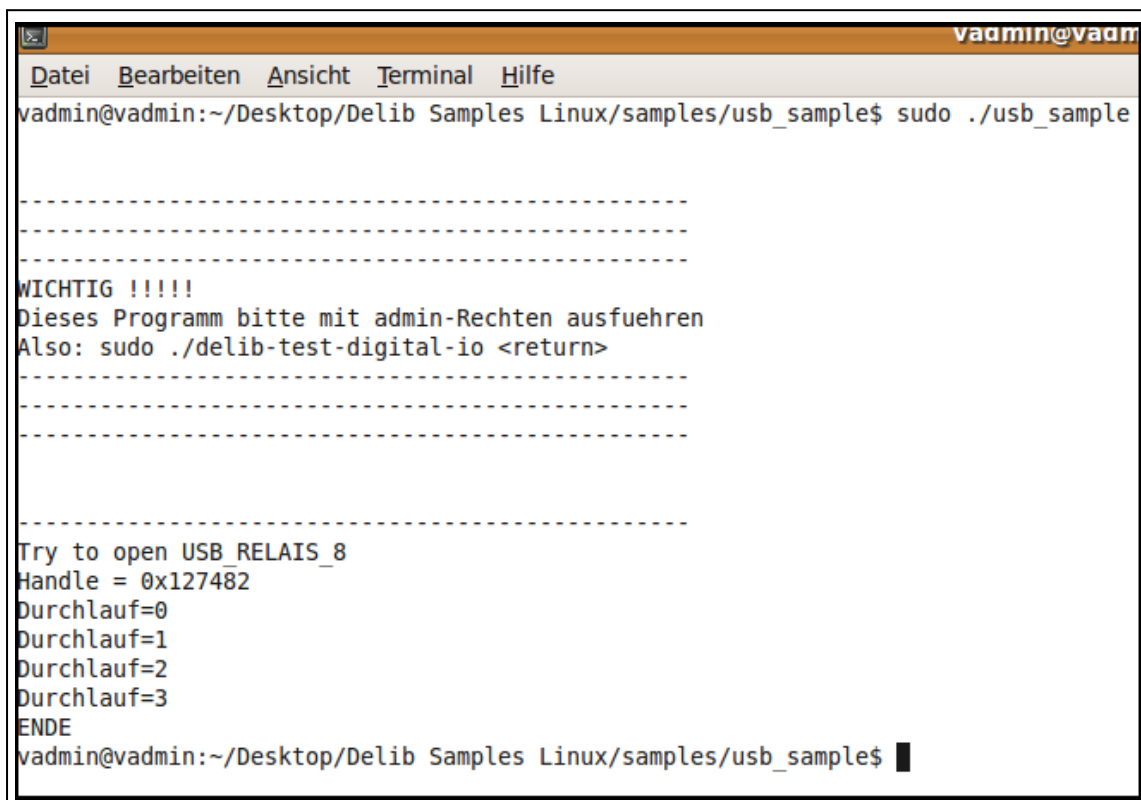
```
vadmin@vadmin: ~/Desktop/Del
Datei Bearbeiten Ansicht Terminal Hilfe
vadmin@vadmin:~/Desktop/Delib Samples Linux/samples/usb_sample$ sudo sh ./1_compile_usb_sample.sh
[sudo] password for vadmin:
```

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

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

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

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



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

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

-----

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

Das Programm wird nun ausgeführt.

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

### 3.4.2.2. Delib ETH-Sample in Linux

#### Voreinstellungen

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

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

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

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

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

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

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

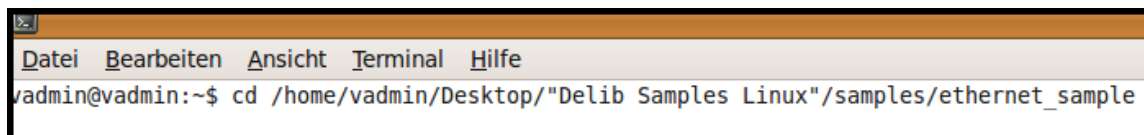
Auskommentieren,  
falls keine digitalen  
Eingänge vorhanden

## Kompilieren des ETH-Samples

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

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

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

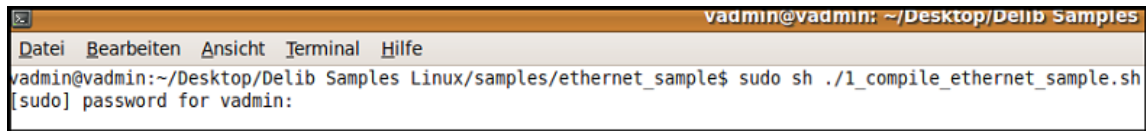
A screenshot of a terminal window with a menu bar containing 'Datei', 'Bearbeiten', 'Ansicht', 'Terminal', and 'Hilfe'. The terminal text shows the command 'vadmin@vadmin:~\$ cd /home/vadmin/Desktop/"Delib Samples Linux"/samples/ethernet\_sample' being entered.

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

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

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

Geben Sie, falls nötig, Ihr Benutzerkennwort ein.



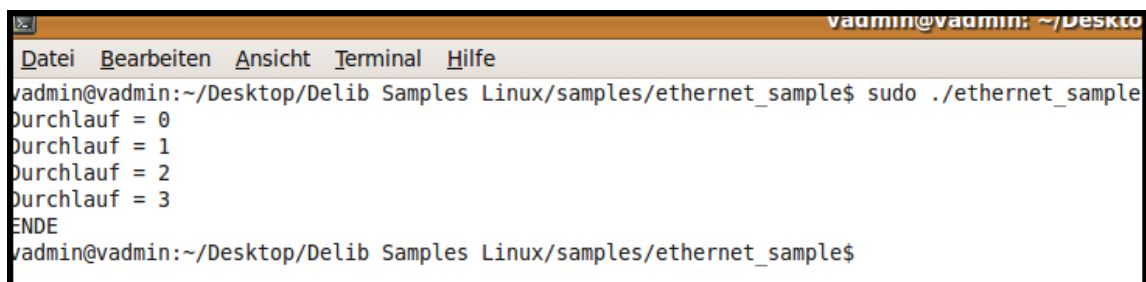
```
vadmin@vadmin: ~/Desktop/Delib Samples
Datei Bearbeiten Ansicht Terminal Hilfe
vadmin@vadmin:~/Desktop/Delib Samples Linux/samples/ethernet_sample$ sudo sh ./1_compile_ethernet_sample.sh
[sudo] password for vadmin:
```

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

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

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

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



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

Das Programm wird nun ausgeführt.

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



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

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

Definition für USB-Module (Linux)

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

Definition für ETH-Module (Linux)

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

#### **Hinweis:**

Die einzelnen Parameter werden nur durch ein Leerzeichen getrennt.

Groß und Kleinschreibung wird hierbei nicht beachtet.

## Parameter

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

### **Return-Wert**

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

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

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

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

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

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

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

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

### 3.4.3.1. Konfiguration des DELIB CLI

#### Voreinstellungen

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

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

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

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

#### moduleID

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

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

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

#### moduleNR

Die moduleNR wird im DELIB Configuration Utility vergeben.

Diese Nummer dient zur Identifizierung identischer Hardware.

Der Standardwert ist 0.

#### RO-ETH\_ipAddress

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

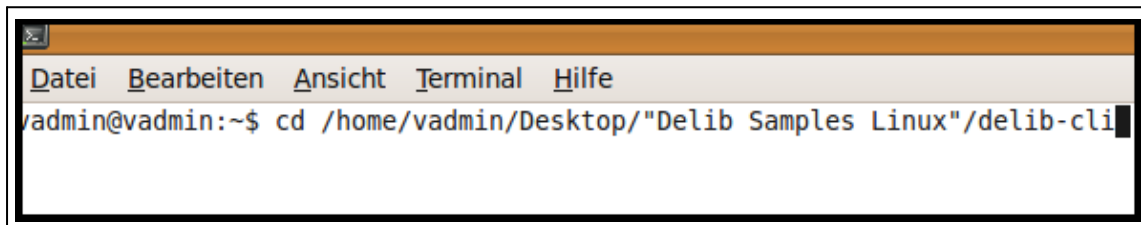
Die IP-Adresse der ETH-Module können über das DELIB Configuration Utility sowie über die Weboberfläche des Moduls eingestellt werden.

## Kompilieren des Delib-CLI-Samples

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

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

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

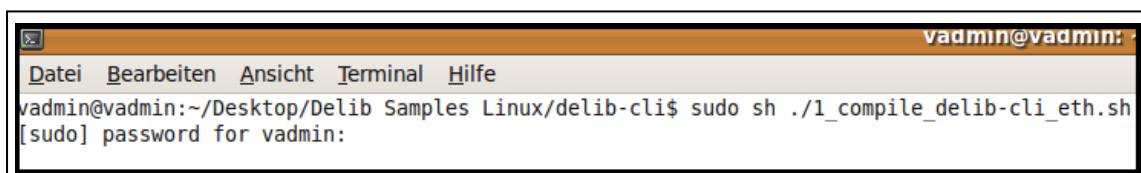


```
File Bearbeiten Ansicht Terminal Hilfe
vadmin@vadmin:~$ cd /home/vadmin/Desktop/"Delib Samples Linux"/delib-cli
```

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.

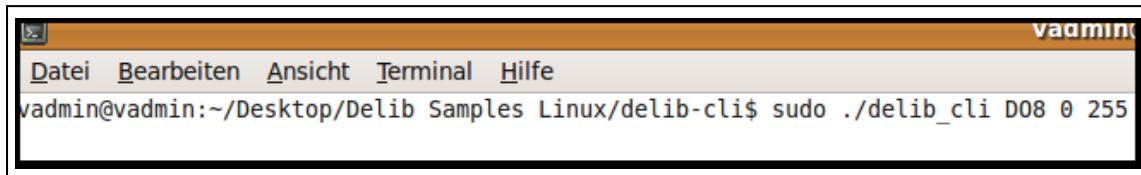


```
vadmin@vadmin:
File Bearbeiten Ansicht Terminal Hilfe
vadmin@vadmin:~/Desktop/Delib Samples Linux/delib-cli$ sudo sh ./1_compile_delib-cli_eth.sh
[sudo] password for vadmin:
```

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".



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

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

### 3.4.3.2. DELIB CLI Beispiele

#### Digitale Ausgänge

```
sudo delib_cli DO1 17 1
```

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

```
sudo delib_cli DO1 3 0
```

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

#### Digitale Eingänge

```
sudo delib_cli DI1 3
```

Beispiel eines Rückgabewertes: **1**

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

```
sudo delib_cli DI8 0 hex
```

Beispiel eines Rückgabewertes: **0xFF**

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

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

```
sudo delib_cli FF 0
```

Beispiel eines Rückgabewertes: **192**

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

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

```
sudo delib_cli FF 32
```

Beispiel eines Rückgabewertes: **65535**

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

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

```
sudo delib_cli FF 0 hex
```

Beispiel eines Rückgabewertes: **0xD00**

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

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

### Analoge Ausgänge

```
sudo delib_cli AO 7 4711
```

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

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

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

### Analoge Eingänge

```
sudo delib_cli AI 2
```

Beispiel eines Rückgabewertes: **1234**

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

```
sudo delib_cli AI 2 hex
```

Beispiel eines Rückgabewertes: **0x1FA**

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



## **DELIB API Referenz**



## 4. DELIB API Referenz

### 4.1. Verfügbare DEDITEC Modul IDs

Hier finden Sie eine Auflistung mit allen verfügbaren Modul IDs.

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

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

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

| Modul Name              | ID |
|-------------------------|----|
| USB_RELAIS_8            | 14 |
| USB_OPTOIN_8_RELAIS_8   | 15 |
| USB_OPTOIN_16_RELAIS_16 | 16 |
| USB_OPTOIN_32           | 16 |
| USB_RELAIS_32           | 16 |
| USB_OPTOIN_32_RELAIS_32 | 17 |
| USB_OPTOIN_64           | 17 |
| USB_RELAIS_64           | 17 |
| BS_USB_8                | 15 |
| BS_USB_16               | 16 |
| BS_USB_32               | 17 |
| USB_TTL_32              | 18 |
| USB_TTL_64              | 18 |
| RO_ETH_INTERN           | 19 |
| BS_SER                  | 20 |
| BS_CAN                  | 21 |
| BS_ETH                  | 22 |
| NET_ETH                 | 23 |
| RO_CAN2                 | 24 |
| RO_USB2                 | 25 |

| Modul Name        | ID |
|-------------------|----|
| RO_ETH_LC         | 26 |
| ETH_RELAIS_8      | 27 |
| ETH_OPTOIN_8      | 27 |
| ETH_O4_R4_ADDA    | 28 |
| ETHERNET_MODULE   | 29 |
| ETH_TTL_64        | 30 |
| NET_USB2          | 31 |
| NET_ETH_LC        | 32 |
| NET_USB1          | 33 |
| NET_SER           | 34 |
| NET_CAN_OPEN      | 35 |
| NET_RAS_PI        | 36 |
| USB_CANOPEN_STICK | 37 |
| ETH_CUST_0        | 38 |
| WEU_RELAIS_8      | 39 |
| WEU_OPTO_8        | 39 |
| WEU_E_RELAIS_8    | 40 |
| BS_WEU            | 41 |
| BS_WEU_E          | 42 |

## 4.2. Verwaltungsfunktionen

### 4.2.1. DapiOpenModule

#### Beschreibung

Diese Funktion öffnet ein bestimmtes Modul.

#### Definition

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

#### Parameter

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

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

nr=0 → 1. Modul

nr=1 → 2. Modul

#### Return-Wert

handle=Entsprechender Handle für das Modul

handle=0 → Modul wurde nicht gefunden

#### Bemerkung

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

#### Programmierbeispiel

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

### 4.2.2. DapiCloseModule

#### Beschreibung

Dieser Befehl schließt ein geöffnetes Modul.

#### Definition

*ULONG DapiCloseModule(ULONG handle);*

#### Parameter

handle=Dies ist das Handle eines geöffneten Moduls.

#### Return-Wert

Keiner

#### Programmierbeispiel

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

### 4.2.3. DapiGetDELIBVersion

#### Beschreibung

Diese Funktion gibt die installierte DELIB-Version zurück.

#### Definition

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

#### Parameters

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

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

#### Return-Wert

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

#### Programmierbeispiel

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

#### 4.2.4. DapiSpecialCMDGetModuleConfig

##### Beschreibung

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

##### Definition

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

##### Parameter

handle=Dies ist der handle eines offenen Moduls

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

#### **Return value**

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

return=number of digital input channels

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

return=number of digital input flip-flops

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

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

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

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

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

return=number of digital output channels

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

return=number of digital pulse generator outputs

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

return=number of digital PWM outputs



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

return=number of digital input/output channels

### Querying the number of analog input channels

return=number of analog input channels

### Querying the number of analog output channels

return=number of analog output channels

### Query number of temperature channels

return=number of temperature channels

### Query number of stepper channels

return=number of stepper channels

### Programmierbeispiele

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

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

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

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

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

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

#### 4.2.5. DapiOpenModuleEx

##### Beschreibung

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

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

##### Definition

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

##### Parameter

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

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

nr = 0 → 1. Modul

nr = 1 → 2. Modul

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

##### Return-Wert

handle = Entsprechender Handle für das Modul

handle = 0 → Modul wurde nicht gefunden

### Bemerkung

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

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

Universelle Ethernet moduleID

### Die moduleID:

ETHERNET\_MODULE = 29

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

### Encryption Type

Folgende Encryption Types stehen zur Verfügung:

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

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

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

### Programmierbeispiel

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

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

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

#### 4.2.6. DapiScanAllModulesAvailable

##### Beschreibung

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

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

##### Definition

*ULONG DapiScanAllModulesAvailable(uint nr)*

##### Parameter

nr = 0: Es wird nach allen am USB-Bus angeschlossenen Module gesucht

nr = i: Auslesen der einzelnen angeschlossenen Module

##### Return-Wert

Gibt die Anzahl der gefunden Module wieder.

##### Programmierbeispiel

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

## 4.3. Fehlerbehandlung

### 4.3.1. DapiGetLastError

#### Beschreibung

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

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

#### Definition

*ULONG DapiGetLastError();*

#### Parameter

Keine

#### Return-Wert

Fehler Code

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

#### Programmierbeispiel

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

### 4.3.2. DapiGetLastErrorText

#### Beschreibung

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

#### Definition

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

#### Parameter

msg = Buffer für den zu empfangenden Text

msg\_length = Länge des Text Buffers

#### Programmierbeispiel

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

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

        DapiClearLastError();

        return TRUE;
    }

    return FALSE;
}
```

### 4.3.3. DapiClearLastError

#### Beschreibung

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

#### Definition

*void DapiClearLastError();*

#### Parameter

Keine

#### Return-Wert

Keine

#### Programmierbeispiel

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

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

        DapiClearLastError();

        return TRUE;
    }

    return FALSE;
}
```

#### 4.3.4. DapiGetLastErrorByHandle

##### Beschreibung

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

##### Definition

*ULONG DapiGetLastErrorByHandle(ULONG handle);*

##### Parameter

handle=Dies ist das Handle eines geöffneten Moduls.

##### Return-Wert

Fehler Code

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

##### Programmierbeispiel

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

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

        DapiClearLastErrorByHandle(handle);

        return TRUE;
    }

    return FALSE;
}
```



### 4.3.5. DapiClearLastErrorByHandle

#### Beschreibung

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

#### Definition

*void DapiClearLastErrorByHandle();*

#### Parameter

handle=Dies ist das Handle eines geöffneten Moduls.

#### Return-Wert

Keine

#### Programmierbeispiel

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

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

        DapiClearLastErrorByHandle(handle);

        return TRUE;
    }

    return FALSE;
}
```

## 4.4. Watchdog Funktionen

### 4.4.1. DapiWatchdogEnable

#### Beschreibung

Diese Funktion aktiviert den Watchdog.

#### Definition

```
void DapiWatchdogEnable(ULONG handle);
```

#### Parameter

handle=Dies ist das Handle eines geöffneten Moduls

#### Return-Wert

Keiner

#### Programmierbeispiel

```
DapiWatchdogEnable(handle);  
//Aktiviert den Watchdog
```

#### 4.4.2. DapiWatchdogDisable

##### Beschreibung

Diese Funktion deaktiviert den Watchdog.

##### Definition

```
void DapiWatchdogDisable(ULONG handle);
```

##### Parameter

handle=Dies ist das Handle eines geöffneten Moduls

##### Return-Wert

Keiner

##### Programmierbeispiel

```
DapiWatchdogDisable(handle);  
//Deaktiviert den Watchdog
```

### 4.4.3. DapiWatchdogRetrigger

#### Beschreibung

Diese Funktion retriggert den Watchdog-Timer.

#### Definition

*void DapiWatchdogRetrigger(ULONG handle);*

#### Parameter

handle=Dies ist das Handle eines geöffneten Moduls

#### Return-Wert

Keiner

#### Programmierbeispiel

```
DapiWatchdogRetrigger(handle);  
//Retriggert den Watchdog-Timer
```

## 4.5. Testfunktionen

### 4.5.1. DapiPing

#### Beschreibung

Dieser Befehl prüft die Verbindung zu einem geöffneten Modul.

#### Definition

*ULONG DapiPing(ULONG handle, ULONG value);*

#### Parameter

handle=Dies ist das Handle eines geöffneten Moduls

value=Übergebener Testwert, im Wertebereich von 0-255 (8-Bit), an das Modul

#### Return-Wert

Hier muß der mit "value" übergebene Testwert zurückkommen

## 4.6. Register Schreib-Befehle

### 4.6.1. DapiWriteByte

#### Beschreibung

Dieser Befehl führt einen direkten Register Schreibbefehl auf das Modul aus.

#### Definition

*void DapiWriteByte(ULONG handle, ULONG adress, ULONG value);*

#### Parameter

handle=Dies ist das Handle eines geöffneten Moduls

adress=Adresse, auf die zugegriffen werden soll

value=Gibt den Datenwert an, der geschrieben wird (8 Bit)

#### Return-Wert

Keiner

#### Bemerkung

Dies sollte nur von erfahrenen Programmieren benutzt werden. So kann auf alle zur Verfügung stehenden Register direkt zugegriffen werden.

### 4.6.2. DapiWriteWord

#### Beschreibung

Dieser Befehl führt einen direkten Register Schreibbefehl auf das Modul aus.

#### Definition

*void DapiWriteWord(ULONG handle, ULONG adress, ULONG value);*

#### Parameter

handle=Dies ist das Handle eines geöffneten Moduls

adress=Adresse, auf die zugegriffen werden soll

value=Gibt den Datenwert an, der geschrieben wird (16 Bit)

#### Return-Wert

Keiner

#### Bemerkung

Dies sollte nur von erfahrenen Programmieren benutzt werden. So kann auf alle zur Verfügung stehenden Register direkt zugegriffen werden.

### 4.6.3. DapiWriteLong

#### Beschreibung

Dieser Befehl führt einen direkten Register Schreibbefehl auf das Modul aus.

#### Definition

*void DapiWriteLong(ULONG handle, ULONG adress, ULONG value);*

#### Parameter

handle=Dies ist das Handle eines geöffneten Moduls

adress=Adresse, auf die zugegriffen werden soll

value=Gibt den Datenwert an, der geschrieben wird (32 Bit)

#### Return-Wert

Keiner

#### Bemerkung

Dies sollte nur von erfahrenen Programmieren benutzt werden. So kann auf alle zur Verfügung stehenden Register direkt zugegriffen werden.



#### 4.6.4. DapiWriteLongLong

##### Beschreibung

Dieser Befehl führt einen direkten Register Schreibbefehl auf das Modul aus.

##### Definition

*void DapiWriteLongLong(ULONG handle, ULONG adress, ULONGLONG value);*

##### Parameter

handle=Dies ist das Handle eines geöffneten Moduls

adress=Adresse, auf die zugegriffen werden soll

value=Gibt den Datenwert an, der geschrieben wird (64 Bit)

##### Return-Wert

Keiner

##### Bemerkung

Dies sollte nur von erfahrenen Programmieren benutzt werden. So kann auf alle zur Verfügung stehenden Register direkt zugegriffen werden.

## 4.7. Register Lese-Befehle

### 4.7.1. DapiReadByte

#### Beschreibung

Dieser Befehl führt einen direkten Register Lese-Befehl auf das Modul aus.

#### Definition

*ULONG DapiReadByte(ULONG handle, ULONG adress);*

#### Parameter

handle=Dies ist das Handle eines geöffneten Moduls

adress=Adresse, auf die zugegriffen werden soll

#### Return-Wert

Inhalt des zu lesenden Registers (8 Bit)

#### Bemerkung

Dies sollte nur von erfahrenen Programmieren benutzt werden. So kann auf alle zur Verfügung stehenden Register direkt zugegriffen werden.

### 4.7.2. DapiReadWord

#### Beschreibung

Dieser Befehl führt einen direkten Register Lese-Befehl auf das Modul aus.

#### Definition

*ULONG DapiReadWord(ULONG handle, ULONG adress);*

#### Parameter

handle=Dies ist das Handle eines geöffneten Moduls

adress=Adresse, auf die zugegriffen werden soll

#### Return-Wert

Inhalt des zu lesenden Registers (16 Bit)

#### Bemerkung

Dies sollte nur von erfahrenen Programmieren benutzt werden. So kann auf alle zur Verfügung stehenden Register direkt zugegriffen werden.

### 4.7.3. 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.

#### Programmbeispiel

```
char v0, v1, v2, v3;
uint ver;
float fw_ver;

ver = (uint)DapiReadLong(handle, 0xffff4);

v3 = (char)((ver >> 24) & 0xff);
v2 = (char)((ver >> 16) & 0xff);
v1 = (char)((ver >> 8) & 0xff);
v0 = (char)((ver >> 0) & 0xff);

fw_ver = (((float)v0) - '0') * 10 + (((float)v1) - '0')
+ (((float)v2) - '0') / 10 + (((float)v3) - '0') / 100;
// Hier wird die Firmware-Version des Modules ausgelesen.
```

#### 4.7.4. DapiReadLongLong

##### Beschreibung

Dieser Befehl führt einen direkten Register Lese-Befehl auf das Modul aus.

##### Definition

*ULONGLONG DapiReadLongLong(ULONG handle, ULONG adress);*

##### Parameter

handle=Dies ist das Handle eines geöffneten Moduls

adress=Adresse, auf die zugegriffen werden soll

##### Return-Wert

Inhalt des zu lesenden Registers (64 Bit)

##### Bemerkung

Dies sollte nur von erfahrenen Programmieren benutzt werden. So kann auf alle zur Verfügung stehenden Register direkt zugegriffen werden.

## 4.8. Programmier-Beispiel

```
// *****  
// *****  
//  
// (c) DEDITEC GmbH, 2009  
//  
// web: http://www.deditec.de  
//  
// mail: vertrieb@deditec.de  
//  
// dtapi_prog_beispiel_input_output.cpp  
//  
// *****  
// *****  
//  
// Folgende Bibliotheken beim Linken mit einbinden: delib.lib  
// Dies bitte in den Projekteinstellungen  
// (Projekt/Einstellungen/Linker(Objekt-  
// Bibliothek-Module) .. letzter Eintrag konfigurieren  
#include <windows.h>  
#include <stdio.h>  
#include "conio.h"  
#include "delib.h"  
// *****  
// *****  
  
void main(void)  
{  
    unsigned long handle;  
    unsigned long data;  
    unsigned long anz;  
    unsigned long i;  
    unsigned long chan;  
    // -----  
    // USB-Modul öffnen  
    handle = DapiOpenModule(USB_Interface8,0);  
    printf("USB_Interface8 handle = %x\n", handle);  
    if (handle==0)  
    {  
        // USB Modul wurde nicht gefunden  
        printf("Modul konnte nicht geöffnet werden\n");  
        printf("TASTE für weiter\n");  
        getch();  
        return;  
    }  
    // Zum Testen - ein Ping senden  
    // -----  
    printf("PING\n");  
    anz=10;  
    for(i=0;i!=anz;++i)  
    {  
        data=DapiPing(handle, i);  
        if(i==data)  
        {  
            // OK  
            printf(".");  
        }  
        else  
        {  

```

```

// No answer
printf("E");
}
}
printf("\n");

// -----
// Einen Wert auf die Ausgänge schreiben
data = 255;
DapiWriteByte(handle, 0, data);
printf("Schreibe auf Adresse=0 daten=0x%x\n", data);
// -----
// Einen Wert auf die Ausgänge schreiben
data = 255;
DapiWriteByte(handle, 1, data);
printf("Schreibe auf Adresse=0 daten=0x%x\n", data);
// -----
// Einen Wert auf die Ausgänge schreiben
data = 255;
DapiWriteByte(handle, 2, data);
printf("Schreibe auf Adresse=2 daten=0x%x\n", data);
// -----
// Einen Wert von den Eingängen lesen
data = (unsigned long) DapiReadByte(handle, 0);
printf("Gelesene Daten = 0x%x\n", data);
// -----
// Einen A/D Wert lesen
chan=11; // read chan. 11
data = DapiReadWord(handle, 0xff010000 + chan*2);
printf("Adress=%x, ret=%x volt=%f\n", chan, data, ((float) data) / 1024*5); //
Bei 5 Volt Ref
// -----
// Modul wieder schliessen
DapiCloseModule(handle);
printf("TASTE für weiter\n");
getch();
return ;
}

```

## 4.9. 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<br>USB-TTL32<br>USB-TTL64<br>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<br>DAPI_SPECIAL_TIMEOUT_DO_VALUE_MASK_WR_SET32<br>DAPI_SPECIAL_TIMEOUT_DO_VALUE_MASK_RD_SET32<br>DAPI_SPECIAL_TIMEOUT_DO_VALUE_MASK_WR_CLR32<br>DAPI_SPECIAL_TIMEOUT_DO_VALUE_MASK_RD_CLR32<br>DAPI_SPECIAL_TIMEOUT_DO_VALUE_LOAD_DEFAULT | ETH-TTL64<br>ETH-RELAIS8<br>USB-RELAIS8<br>RO-SERIE<br>BS-SERIE<br>NET-SERIE<br>USB-TTL-64 | USB-Mini-Stick |
| DAPI_SPECIAL_TIMEOUT_SET_VALUE_SEC<br>DAPI_SPECIAL_TIMEOUT_ACTIVATE<br>DAPI_SPECIAL_TIMEOUT_DEACTIVATE<br>DAPI_SPECIAL_TIMEOUT_GET_STATUS  | alle Module  |                |



| Befehl   | Starter<br>USB* | Starter<br>ETH** | RO<br>Serie | BS<br>Serie | NET<br>Serie | Sonstiges |
|--|-----------------|------------------|-------------|-------------|--------------|-----------|
| DAPI_SPECIAL_COUNTER_ LATCH_ALL  |                 |                  | x           |             |              |           |
| DAPI_SPECIAL_COUNTER_ LATCH_ALL_WITH_RESET   |                 |                  | x           |             |              |           |
| DapiDOSet1_WithTimer   |                 |                  | x           |             |              |           |
| DAPI_SPECIAL_CMD_SW_FIFO<br>DAPI_SPECIAL_SW_FIFO_INIT_<br>AND_CLEAR<br>...<br>DAPI_SPECIAL_SW_FIFO_<br>IO_DEACTIVATE |                 |                  |             |             | x            |           |
| DAPI_SPECIAL_CMD_AD<br>DAPI_SPECIAL_RO_AD_<br>FIFO_ACTIVATE<br>...<br>DAPI_SPECIAL_RO_AD_<br>FIFO_INIT               |                 |                  | x           |             |              |           |

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

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

| Befehl  | Starter<br>USB* | Starter<br>ETH** | RO<br>Serie | BS<br>Serie | NET<br>Serie | Sonstiges |
|---|-----------------|------------------|-------------|-------------|--------------|-----------|
| DAPI_SPECIAL_DI_FF_FILTER<br>DAPI_SPECIAL_DI_FF_FILTER_<br>VALUE_SET<br>DAPI_SPECIAL_DI_FF_FILTER_<br>VALUE_GET | 5-255           | 1-255            | 1-255       | 1-255       | 1-255        |           |
| DAPI_SPECIAL_DI_FILTER<br>DAPI_SPECIAL_DI_FILTER_<br>VALUE_SET<br>DAPI_SPECIAL_DI_FILTER_<br>VALUE_GET          | x               | 0,<br>1-254      | 0,<br>1-254 | 0,<br>1-254 | 0,<br>1-254  |           |
| DAPI_SPECIAL_CMD_GET_<br>INTERNAL_STATISTIC   | x               | x                | x           | x           |              |           |

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

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

| Befehle   | Verfügbar für                        |
|---|--------------------------------------|
| DAPI_SPECIAL_CMDEXT_CAN_WR_RUNTIME_<br>_VALUE<br>DAPI_SPECIAL_CMDEXT_CAN_RD_RUNTIME_<br>VALUE | NET-CPU-PRO,<br>BS-WEU,<br>RO-ETH-LC |

# Anhang

---



## **5. Anhang**

### **5.1. Kontakt / Support**

Wenn Sie Fragen zum Produkt haben oder Unterstützung bei der Inbetriebnahme brauchen, erreichen Sie uns unter folgenden Rufnummern:

#### **Support Software**

Tel. +49 (0) 22 32 / 50 40 8 – 20

#### **Support Hardware**

Tel. +49 (0) 22 32 / 50 40 8 – 30

#### **Support via E-mail**

[support@deditec.de](mailto:support@deditec.de)

### **5.2. 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](#)

### 5.3. Revisionen

|          |  |
|----------|--|
| Rev 3.00 | DEDITEC Design Update  |
| Rev 2.02 | Kapitel "Software" und "DELIB API Referenz" überarbeitet   |
| Rev 2.01 | Index hinzugefügt  |
| Rev 2.00 | Designänderung   |
| Rev 1.02 | Watchdog Configuration Utility,<br>neue Verwaltungsfunktion "DapiGetDELIBVersion",<br>neue Watchdog-Special Funktionen<br>"DapiSpecialWatchdogGetStatus",<br>"DapiSpecialWatchdogGetTimeoutMsec",<br>"DapiSpecialWatchdogGetWDCounterMsec",<br>"DapiSpecialWatchdogGetTimeoutRelaisCounterMsec",<br>"DapiSpecialWatchdogSetTimeoutRel1CounterMsec" und<br>"DapiSpecialWatchdogSetTimeoutRel2CounterMsec" |
| Rev 1.01 | Kapitel Anwendungsbeispiele hinzugefügt  |
| Rev 1.00 | Erste Version  |

## 5.4. 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.