



BS-WEU-Serie
Hardware-Beschreibung

2022 September

INDEX

<u>1. Einleitung</u>	12
1.1. Vorwort	13
1.2. Kundenzufriedenheit	13
1.3. Kundenresonanz	13
1.4. Kurzbeschreibung	14
1.5. Lieferumfang	15
<u>2. Inbetriebnahme</u>	16
2.1. Schritt 1 - Sicherheitshinweise	17
2.2. Schritt 2 - Anschluss der Stromversorgung	18
2.3. Schritt 3 - Verbinden mit dem PC oder Netzwerk	19
2.3.1. Verbindung via USB	19
2.3.2. Verbindung via Ethernet	19
2.3.3. Verbindung via WiFi	20
2.4. Schritt 4 - Installation der Software und Treiber	21
2.5. Schritt 5 - Anschluss der I/O Steckverbinder	22
2.6. Schritt 6 - Funktionstest	24
<u>3. Hardware Beschreibung</u>	25
3.1. Allgemeine technische Daten	26
3.1.1. BS-WEU-08-R8	26
3.1.2. BS-WEU-016-R16	27
3.1.3. BS-WEU-032-R32	28
3.1.4. BS-WEU-048-R48	29
3.1.5. BS-WEU-064-R64	30
3.1.6. BS-WEU-AD16-DA4	31
3.1.7. BS-WEU-AD16-DA4-OR16	32
3.2. Übersichtsbilder	33
3.2.1. BS-WEU-08-R8	33
3.2.2. BS-WEU-016-R16	34

INDEX

3.2.3. BS-WEU-032-R32 bis BS-WEU-064-R64	34
3.2.4. BS-WEU-AD16	35
3.3. Schnittstellen	36
3.3.1. USB	36
3.3.2. Ethernet	36
3.3.3. CAN	37
3.3.4. RS-232	38
3.3.5. Schnittstellen nachrüsten	39
3.4. LEDs	40
3.4.1. Definition der LEDs	40
3.4.2. Blinkverhalten der LEDs	41
3.5. Schalter & Taster	46
3.5.1. Funktionen der DIP-Schalter	46
3.5.2. Funktionen des CFG-Tasters	50
3.6. Analoge Ausgänge	51
3.6.1. Technische Daten	51
3.6.2. Pinbelegung	52
3.6.3. Anschlussbeispiel für Spannungsausgabe bei einem AD16-DA4	53
3.7. Analoge Eingänge	54
3.7.1. Technische Daten	54
3.7.2. Pinbelegung	55
3.7.3. Anschlussbeispiele für Spannungsmessung bei einem AD16-DA4	56
3.7.4. Anschlussbeispiel für Strommessung bei einem AD16-DA4	57
3.8. Blockschaltbild eines AD16-DA4	59
3.9. Digitale Ausgänge	60
3.9.1. Technische Daten	60
3.9.2. Pinbelegung	61
3.9.3. Anschlussbeispiel für Relais	62
3.9.4. Anschlussbeispiel für MOSFET	63
3.9.5. Blockschaltbild eines Relais-Moduls	64

INDEX

3.9.6. Blockschaltbild eines MOSFET-Moduls	65
3.9.7. Timeout Funktion	66
3.10. Digitale Eingänge	67
3.10.1. Technische Daten	67
3.10.2. Pinbelegung	68
3.10.3. Anschlussbeispiel eines Optokopplers	69
3.10.4. Blockschaltbild eines Optokoppler-Moduls	70
3.10.5. Eingangsfiler	71
3.10.6. Zustandsänderungen überwachen	72
4. Software Beschreibung	73
4.1. Benutzen unserer Produkte	74
4.1.1. Ansteuerung über unsere DELIB Treiberbibliothek	74
4.1.2. Ansteuerung über mitgelieferte Testprogramme	74
4.1.3. Ansteuerung auf Protokollebene	75
4.1.4. DELIB CLI (command-line interface) für Windows	76
4.1.4.1. Konfiguration des DELIB CLI	78
4.1.4.2. DELIB CLI Beispiele	79
4.1.5. Ansteuerung über grafische Anwendungen	83
4.1.5.1. LabVIEW	83
4.1.5.2. ProfilLab	83
4.1.5.3. Licht24 Pro	84
4.1.6. Einbinden der DELIB in Programmiersprachen	85
4.1.6.1. Einbinden der DELIB in Visual-C/C++	85
4.1.6.2. Einbinden der DELIB in Visual-C/C++ (Visual Studio 2015)	87
4.1.6.3. Einbinden der DELIB in Visual-C#	90
4.1.6.4. Einbinden der DELIB in Delphi	91
4.1.6.5. Einbinden der DELIB in Visual-Basic (VB)	92
4.1.6.6. Einbinden der DELIB in Visual-Basic.NET (VB.NET)	93
4.1.6.7. Einbinden der DELIB in MS-Office (VBA)	94
4.1.6.8. Einbinden der DELIB in LabVIEW	96
4.1.6.8.1. Einbinden der DELIB in LabVIEW	96

INDEX

4.1.6.8.2. Verwendung der VIs in LabVIEW	105
4.1.6.8.3. Setzen der Modul-ID in LabVIEW	107
4.1.6.9. Einbinden der DELIB in Java	109
4.2. DELIB Treiberbibliothek	110
4.2.1. Übersicht	111
4.2.1.1. Unterstützte Programmiersprachen	113
4.2.1.2. Unterstützte Betriebssysteme	114
4.2.1.3. SDK-Kit für Programmierer	114
4.2.2. DELIB Setup	115
4.2.3. DELIB Configuration Utility	121
4.2.3.1. Einführung	121
4.2.3.2. Neue Konfiguration erstellen oder vorhandene Konfiguration bearbeiten	122
4.2.3.2.1. Modul Konfiguration USB	123
4.2.3.2.1.1, Beispiel zur Konfiguration identischer USB-Module	125
4.2.3.2.2. Modul Konfiguration Ethernet	130
4.2.3.2.2.1, Automatische Suche	135
4.2.3.2.2.2, Verschlüsselung einrichten	139
4.2.3.2.2.1, Manuelle Konfiguration	139
4.2.3.2.2.2, Automatische Konfiguration	144
4.2.3.2.2.1, Authentifizierung	145
4.2.3.2.3. Modul Konfiguration CAN	149
4.2.3.2.4. Modul Konfiguration Seriell	152
4.2.3.3. Modul testen	154
4.2.3.4. Debug Optionen einstellen	158
4.2.4. Benutzen des Modulselectors	159
4.2.5. DELIB Module Config	165
4.2.5.1. Modul Konfigurationen	165
4.2.5.1.1. Modul-Infoseite	166
4.2.5.1.2. Modul-Identifikation	167
4.2.5.1.3. LAN Netzwerkinformationen	168
4.2.5.1.4. LAN Netzwerkeinstellungen	169
4.2.5.1.5. WiFi Netzwerkinformationen	171

INDEX

4.2.5.1.6. WiFi Netzwerkeinstellungen	173
4.2.5.1.7. WiFi WPS-Verbindung	175
4.2.5.1.8. NTP-Konfiguration	176
4.2.5.1.9. Serielle Konfiguration	178
4.2.5.1.10. I/O Kanal-Namen	180
4.2.5.1.11. CAN Konfiguration	181
4.2.5.1.11.1, CAN Status	181
4.2.5.1.11.1, CAN Status Interface	181
4.2.5.1.11.2, CAN Statistik TX/RX	182
4.2.5.1.11.2, CAN Main	183
4.2.5.1.11.1, CAN Main Interface	183
4.2.5.1.11.2, CAN Main I/O Init	185
4.2.5.1.11.3, CAN TX/RX Modi	187
4.2.5.1.11.1, CAN TX Mode	187
4.2.5.1.11.2, CAN RX Mode	189
4.2.5.2. I/O-Test	190
4.2.5.2.1. Timeout Test-Funktion	190
4.2.5.2.2. Digital In	191
4.2.5.2.3. Digital Out	192
4.2.5.2.4. Analog In	194
4.2.5.2.5. Analog Out	195
4.2.6. DELIB Module Demo	197
4.2.6.1. Auswahl des Moduls	198
4.2.6.2. Allgemein	199
4.2.6.2.1. Module Info	201
4.2.6.3. Digital Input	202
4.2.6.4. Digital Output	203
4.2.6.5. Analog Input	204
4.2.6.6. Analog Output	205
4.2.7. CAN Configuration Utility	207
4.2.7.1. Auswahl des Moduls	209
4.2.7.2. Neue Konfiguration Erstellen, Laden, Speichern	211
4.2.7.3. Konfiguration auf das Modul übertragen	213

INDEX

4.2.7.4. Statistiken vom Modul abfragen	214
4.2.7.5. Konfiguration	217
4.2.7.5.1. Modul Konfiguration	218
4.2.7.5.2. I/O Konfiguration	219
4.2.7.5.3. TX-Konfiguration	222
4.2.7.5.3.1, Beispiel Interval	224
4.2.7.5.3.2, Beispiel Trigger	225
4.2.7.5.4. RX-Konfiguration	226
4.2.7.5.4.1, Beispiel RX-DA	228
4.2.7.5.4.2, Beispiel RX-DO	229
4.2.7.6. Aufbau der CAN-Pakete	230
4.2.7.6.1. Digitale Eingänge	230
4.2.7.6.2. Digitale Ausgänge	231
4.2.7.6.3. Digitale Eingangszähler (16-Bit)	232
4.2.7.6.4. Digitale Eingangszähler (48-Bit) - 32-Bit Paket	233
4.2.7.6.5. Digitale Eingangszähler (48-Bit) - 64-Bit Paket	234
4.2.7.6.6. Analoge Ein- / Ausgänge	235
4.2.7.6.6.1, Analoge Eingänge	235
4.2.7.6.6.2, Analoge Ausgänge	237
4.2.7.6.6.3, Beispiele	238
4.2.7.6.7. Temperatur Eingänge	241
4.2.7.6.8. Stepper	243
4.2.7.6.8.1, Command-Liste	243
4.2.7.6.8.2, Werte für par 1 zu Befehl SET_MOTORCHARACTERISTIC	246
4.2.7.6.8.3, Werte für par 1 zu Befehl GO_REFSWITCH	249
4.2.7.6.8.4, Beispiel	250
4.2.8. DT-Flasher	251
4.2.8.1. Über DEDITEC-Firmware	252
4.2.8.2. Auswahl des Moduls	252
4.2.8.3. Firmware Update durchführen	254
4.2.8.3.1. Flash-Files manuell aktualisieren	256
4.3. DELIB Sample Sources (Windows Programmbeispiele)	257
4.3.1. Installation DELIB Sample Sources	258

INDEX

4.3.2. Benutzung der DELIB Sample Sources	262
4.3.2.1. Schritt 1 - Produktauswahl	262
4.3.2.2. Schritt 2 - Kategorieauswahl	264
4.3.2.3. Schritt 3 - Programmiersprachenauswahl	265
4.3.2.4. Schritt 4 - Quellcode	266
4.4. DELIB für Linux	269
4.4.1. Verwenden der DELIB Treiberbibliothek für Linux	272
4.4.1.1. Delib USB-Sample in Linux	272
4.4.1.2. Delib ETH-Sample in Linux	275
4.4.2. DELIB CLI (command-line interface) für Linux	279
4.4.2.1. Konfiguration des DELIB CLI	282
4.4.2.2. DELIB CLI Beispiele	285
<u>5. DELIB API Referenz</u>	287
5.1. Verfügbare DEDITEC Modul IDs	288
5.2. Verwaltungsfunktionen	291
5.2.1. DapiOpenModule	291
5.2.2. DapiCloseModule	292
5.2.3. DapiGetDELIBVersion	292
5.2.4. DapiSpecialCMDGetModuleConfig	293
5.2.5. DapiOpenModuleEx	296
5.2.6. DapiScanAllModulesAvailable	298
5.3. Fehlerbehandlung	299
5.3.1. DapiGetLastError	299
5.3.2. DapiGetLastErrorText	300
5.3.3. DapiClearLastError	301
5.3.4. DapiGetLastErrorByHandle	302
5.3.5. DapiClearLastErrorByHandle	303
5.4. A/D Wandler Funktionen	304
5.4.1. DapiADSetMode	304
5.4.2. DapiADGetMode	306
5.4.3. DapiADGet	306

INDEX

5.4.4. DapiADGetVolt	307
5.4.5. DapiADGetmA	307
5.4.6. DapiSpecialADReadMultipleAD	308
5.5. D/A Ausgänge verwalten	310
5.5.1. DapiDASetMode	310
5.5.2. DapiDAGetMode	312
5.5.3. DapiDASet	313
5.5.4. DapiDASetVolt	314
5.5.5. DapiDASetmA	315
5.5.6. DapiSpecialCmd_DA	316
5.6. Digitale Eingänge lesen	318
5.6.1. DapiDIGet1	318
5.6.2. DapiDIGet8	318
5.6.3. DapiDIGet16	319
5.6.4. DapiDIGet32	320
5.6.5. DapiDIGet64	321
5.6.6. DapiDIGetFF32	322
5.6.7. DapiDIGetCounter	323
5.6.8. DapiSpecialCounterLatchAll	324
5.6.9. DapiSpecialCounterLatchAllWithReset	325
5.6.10. DapiSpecialDIFilterValueSet	326
5.6.11. DapiSpecialDIFilterValueGet	327
5.6.12. Dapi_Special_DI_FF_Filter_Value_Set	328
5.6.13. Dapi_Special_DI_FF_Filter_Value_Get	329
5.7. Digitale Ausgänge verwalten	330
5.7.1. DapiDOSet1	330
5.7.2. DapiDOSet8	330
5.7.3. DapiDOSet16	331
5.7.4. DapiDOSet32	332
5.7.5. DapiDOSet64	333
5.7.6. DapiDOSet1_WithTimer	334

INDEX

5.7.7. DapiDOReadback32	335
5.7.8. DapiDOReadback64	335
5.7.9. DapiDOSetBit32	336
5.7.10. DapiDOClrBit32	337
5.8. Ausgabe-Timeout verwalten	338
5.8.1. DapiSpecialCMDTimeout	338
5.8.1.1. DapiSpecialTimeoutSetValueSec	341
5.8.1.2. DapiSpecialTimeoutActivate	342
5.8.1.3. DapiSpecialTimeoutActivateAutoReactivate	343
5.8.1.4. DapiSpecialTimeoutActivateSecureOutputs	344
5.8.1.5. DapiSpecialTimeoutDeactivate	345
5.8.1.6. DapiSpecialTimeoutGetStatus	346
5.8.1.7. DapiSpecialTimeoutDoValueMaskWRSet32	347
5.8.1.8. DapiSpecialTimeoutDoValueMaskRDSet32	348
5.8.1.9. DapiSpecialTimeoutDoValueMaskWRClr32	349
5.8.1.10. DapiSpecialTimeoutDoValueMaskRDClr32	350
5.8.1.11. DapiSpecialTimeoutDoValueLoadDefault	351
5.9. CAN Runtime Funktionen	352
5.9.1. RunTimeVarWriteToModule	352
5.10. Testfunktionen	368
5.10.1. DapiPing	368
5.11. Register Schreib-Befehle	369
5.11.1. DapiWriteByte	369
5.11.2. DapiWriteWord	370
5.11.3. DapiWriteLong	371
5.11.4. DapiWriteLongLong	372
5.12. Register Lese-Befehle	373
5.12.1. DapiReadByte	373
5.12.2. DapiReadWord	374
5.12.3. DapiReadLong	375
5.12.4. DapiReadLongLong	376

INDEX

5.13. Programmier-Beispiel	377
5.14. Delib Übersichtstabelle	379
<u>6. Anhang</u>	382
6.1. Kontakt / Support	383
6.2. Umwelt und Entsorgung	383
6.3. Revisionen	384
6.4. Urheberrechte und Marken	384

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

Unsere BS-WEU Produkte sind eine Weiterentwicklung der ursprünglichen BS-Serie. WEU steht hierbei für WIFI, Ethernet und USB. Im Bereich der PC-Messtechnik eignen sich diese Module hervorragend für den Aufbau umfangreicher Automatisierungsprojekte, Steuerungsaufgaben oder Messverfahren.

Es können sowohl analoge als auch digitale Datensignale erfasst oder ausgegeben werden. Die Bereitstellung und Verarbeitung dieser Signale geschieht durch die Kundenapplikation auf dem Steuer PC. Als einfache Programmierschnittstelle bietet sich hier zum Beispiel unsere DELIB API an, oder eine direkte Kommunikation über unser Ethernet-Protokoll.

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

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



1.5. Lieferumfang

Folgende Artikel sind im Lieferumfang enthalten:

- BS-WEU Modul
- 2 poliger Steckverbinder für den Stromanschluss
- 16 oder 18 poliger Steckverbinder mit Auswerfmechanik
- Hilfswerkzeug zum Anschluss der I/O Steckverbinder
- USB Kabel 1,5m
- Installations-CD mit Handbüchern und Treibern

Inbetriebnahme



2. Inbetriebnahme

2.1. Schritt 1 - Sicherheitshinweise

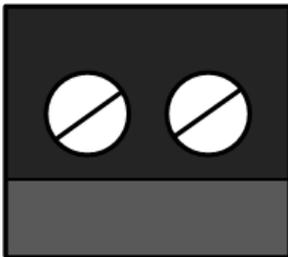
Bitte machen Sie sich vor der Inbetriebnahme Ihres DEDITEC Produktes mit diesem Handbuch vertraut und lesen Sie sich die nachfolgenden Punkte genau durch:

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

2.2. Schritt 2 - Anschluss der Stromversorgung

Wählen Sie ein geeignetes Netzteil* mit ausreichender Leistung von mindestens 5 Watt und einer Ausgangsspannung von zum Beispiel +7VDC oder +24VDC.

Die Stromversorgung wird an der 2-poligen steckbaren Schraubklemme angeschlossen. Bitte beachten Sie dabei die Polarität, wie unten abgebildet. Links V+ und rechts V-.



* Ein passendes Industrie-Netzteil kann als Zubehör bei uns erworben werden.

2.3. Schritt 3 - Verbinden mit dem PC oder Netzwerk

2.3.1. Verbindung via USB

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

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

Siehe Kapitel → **Beispiel zur Konfiguration identischer USB-Module**

2.3.2. Verbindung via Ethernet

Anschluss via Ethernet über einen Switch oder Hub:

Verbinden Sie das Modul mit einem Patch Kabel mit Ihrem Ethernet Switch oder Hub. Das Modul ist im Auslieferungszustand auf DHCP eingestellt (DIP Schalter 1 = ON). Es wird die nächste freie IP Adresse automatisch Ihrem Modul zugeordnet.

Anschluss via Ethernet direkt an den PC:

Verbinden Sie das Modul mit einem Ethernet Crossover Kabel mit Ihrem PC. Stellen Sie den DIP Schalter 1 auf dem Modul auf "OFF". Das Modul startet nun mit der im Modul gespeicherten IP Adresse → **werksseitige IP 192.168.1.1**

2.3.3. Verbindung via WiFi

Um das Modul via WiFi mit Ihrem Router zu verbinden, halten Sie den **"CFG-Taster"** ca. 7 Sekunden gedrückt.

Betätigen Sie anschließend den WPS-Knopf an Ihrem Router.

Das Modul verbindet sich nun automatisch mit Ihrem PC-Netzwerk.

Die **"WIFI-LED"** blinkt nach erfolgreicher Verbindung alle 5 Sekunden 1-mal.

Mehr Informationen zum Blinkverhalten finden Sie im Kapitel

"Blinkverhalten der LEDs".

Mehr Informationen zu Blinkverhalten und Funktionen des Tasters siehe Kapitel

"Funktionen des Tasters"

2.4. Schritt 4 - Installation der Software und Treiber

Installation unter Windows:

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

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

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

Installation unter Linux:

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

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

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

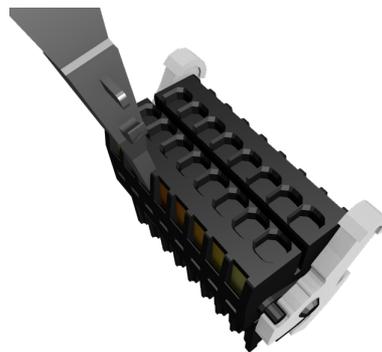
2.5. Schritt 5 - Anschluss der I/O Steckverbinder

Für den Leitungsanschluss an die I/O Steckverbinder benötigen Sie ein Hilfswerkzeug, welches im Lieferumfang enthalten ist.

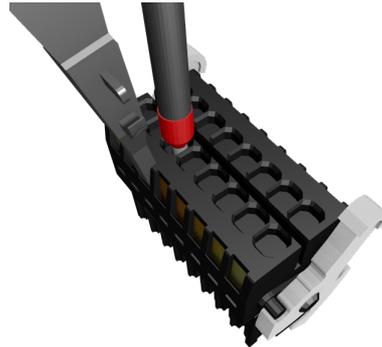


Gehen Sie zum Anschluss bitte wie folgt vor:

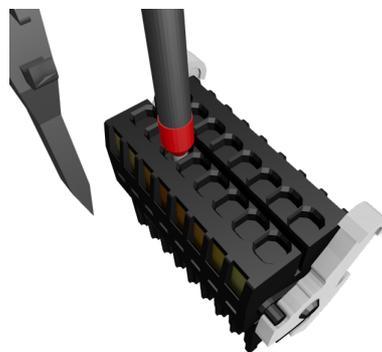
- Betätigungswerkzeug in Leiteranschlussrichtung kräftig (nach unten) in die seitliche Öffnung stecken.



- Die Anschlussleitung 6-7mm abisolieren und in den geöffneten Klemmkontakt stecken.



- Betätigungswerkzeug wieder herausziehen. Überprüfen Sie anschließend, ob die Leitung fest in der Klemme sitzt!



2.6. Schritt 6 - Funktionstest

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

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

Hardware Beschreibung



3. Hardware Beschreibung

3.1. Allgemeine technische Daten

3.1.1. BS-WEU-08-R8



Elektrische Daten:

Versorgungsspannung: 7V DC .. 24V DC

Umgebung:

Umgebungstemperatur: +10..+50 °C

Luftfeuchtigkeit: 90 %

Betauung: Nicht erlaubt

Mechanik:

Abmessungen in mm (LxBxH): 119 x 105 x 74,5

Befestigung: Hutschiene TS 35 x 7,5 mm

Mögliche IO Varianten:

Variante 1: 8 Eingänge / 8 Ausgänge

Variante 2: 16 Eingänge

Variante 3: 16 Ausgänge

3.1.2. BS-WEU-016-R16



Elektrische Daten:

Versorgungsspannung: 7V DC .. 24V DC

Umgebung:

Umgebungstemperatur: +10..+50 °C

Luftfeuchtigkeit: 90 %

Betauung: Nicht erlaubt

Mechanik:

Abmessungen in mm (LxBxH): 152 x 105 x 74,5

Befestigung: Hutschiene TS 35 x 7,5 mm

Mögliche IO Varianten:

Variante 1: 16 Eingänge / 16 Ausgänge

Variante 2: 32 Eingänge

Variante 3: 32 Ausgänge

3.1.3. BS-WEU-032-R32



Elektrische Daten:

Versorgungsspannung: 7V DC .. 24V DC

Umgebung:

Umgebungstemperatur: +10..+50 °C

Luftfeuchtigkeit: 90 %

Betauung: Nicht erlaubt

Mechanik:

Abmessungen in mm (LxBxH): 258 x 105 x 74,5

Befestigung: Hutschiene TS 35 x 7,5 mm

Mögliche IO Varianten:

Variante 1: 32 Eingänge / 32 Ausgänge

Variante 2: 64 Eingänge

Variante 3: 64 Ausgänge

3.1.4. BS-WEU-048-R48



Elektrische Daten:

Versorgungsspannung: 7V DC .. 24V DC

Umgebung:

Umgebungstemperatur: +10..+50 °C

Luftfeuchtigkeit: 90 %

Betauung: Nicht erlaubt

Mechanik:

Abmessungen in mm (LxBxH): 364 x 105 x 74,5

Befestigung: Hutschiene TS 35 x 7,5 mm

Mögliche IO Varianten:

Variante 1: 48 Eingänge / 48 Ausgänge

Variante 2: 96 Eingänge

Variante 3: 96 Ausgänge

3.1.5. BS-WEU-064-R64



Elektrische Daten:

Versorgungsspannung: 7V DC .. 24V DC

Umgebung:

Umgebungstemperatur: +10..+50 °C

Luftfeuchtigkeit: 90 %

Betauung: Nicht erlaubt

Mechanik:

Abmessungen in mm (LxBxH): 469 x 105 x 74,5

Befestigung: Hutschiene TS 35 x 7,5 mm

Mögliche IO Varianten:

Variante 1: 64 Eingänge / 64 Ausgänge

Variante 2: 128 Eingänge

Variante 3: 128 Ausgänge

3.1.6. BS-WEU-AD16-DA4



Elektrische Daten:

Versorgungsspannung: 7V DC .. 24V DC

Umgebung:

Umgebungstemperatur: +10..+50 °C

Luftfeuchtigkeit: 90 %

Betauung: Nicht erlaubt

Mechanik:

Abmessungen in mm (LxBxH): 152 x 105 x 74,5

Befestigung: Hutschiene TS 35 x 7,5 mm

Mögliche IO Varianten:

Variante 1: AD16

Variante 2: AD16 + DA4

3.1.7. BS-WEU-AD16-DA4-OR16



Elektrische Daten:

Versorgungsspannung: 7V DC .. 24V DC

Umgebung:

Umgebungstemperatur: +10..+50 °C

Luftfeuchtigkeit: 90 %

Betauung: Nicht erlaubt

Mechanik:

Abmessungen in mm (LxBxH): 258 x 105 x 74,5

Befestigung: Hutschiene TS 35 x 7,5 mm

Mögliche IO Varianten:

Variante 1: AD16 + DA4 + 16 Eingänge +
16 Ausgänge

Variante 2: AD16 + DA4 + 32 Eingänge

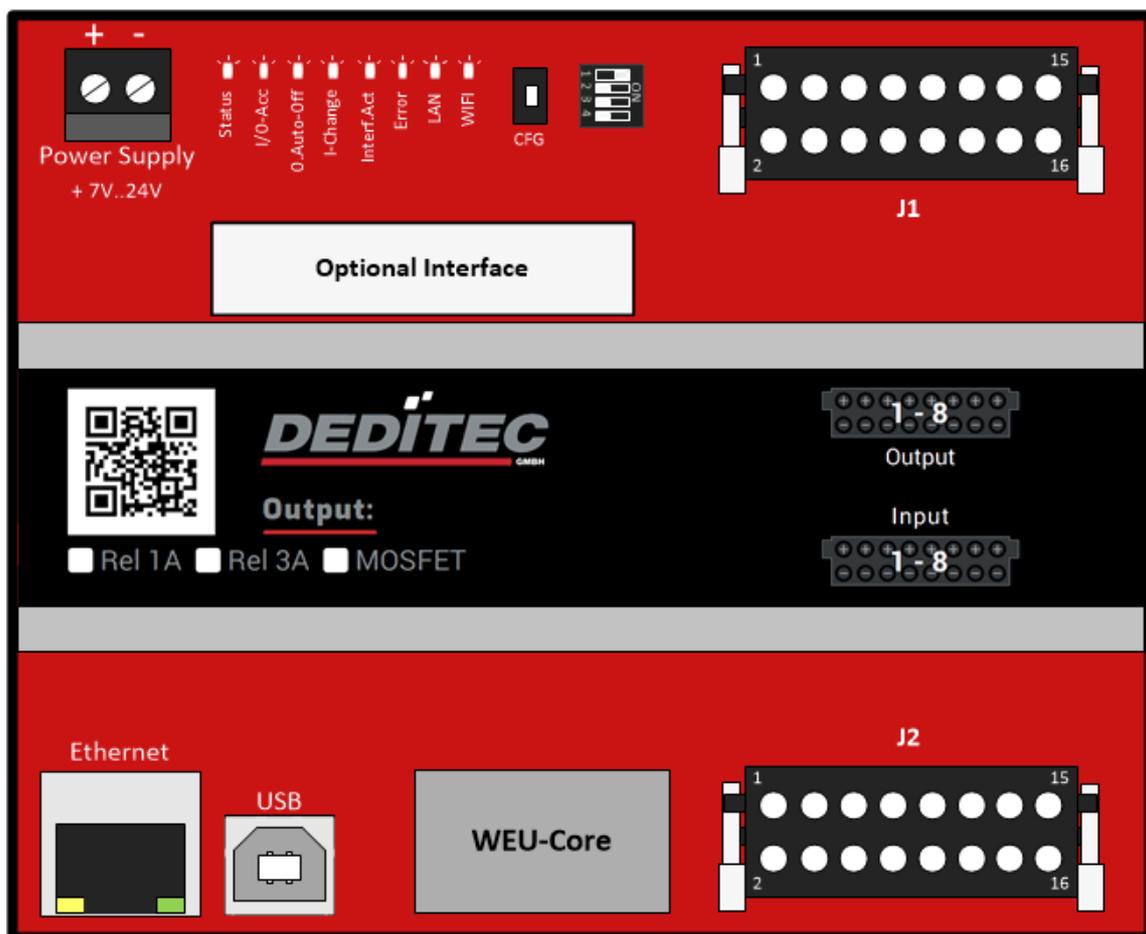
Variante 3: AD16 + DA4 + 32 Ausgänge

3.2. Übersichtsbilder

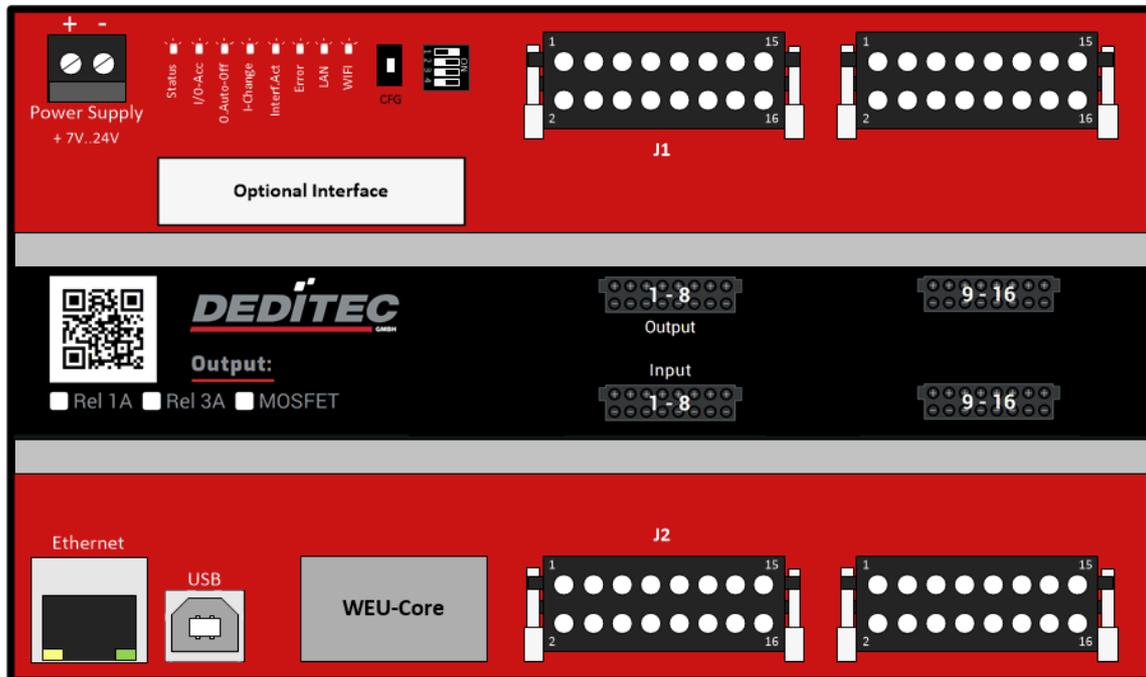
Den Leiterplattenaufbau Ihres Moduls können Sie sich auf den folgenden Übersichtsbildern ansehen.

Informationen zu den Beschreibungen und Funktionen der einzelnen Komponenten, können Sie den nachfolgenden Kapiteln entnehmen.

3.2.1. BS-WEU-08-R8



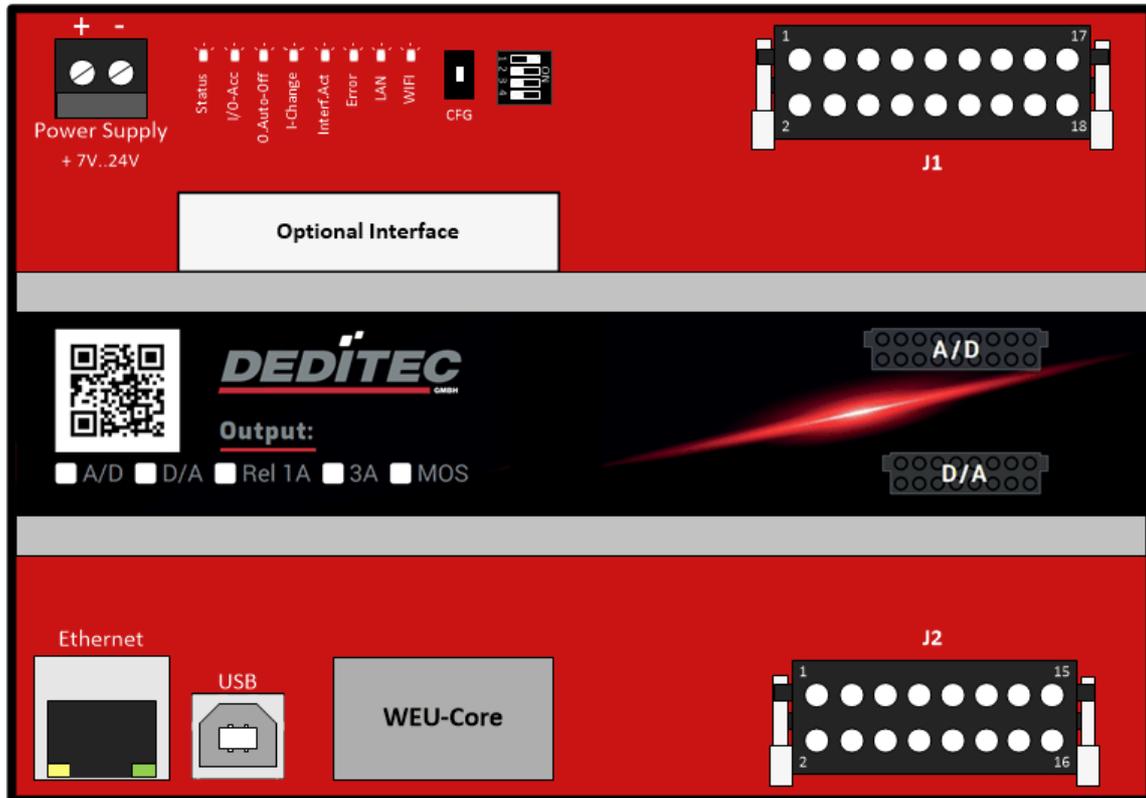
3.2.2. BS-WEU-016-R16



3.2.3. BS-WEU-032-R32 bis BS-WEU-064-R64



3.2.4. BS-WEU-AD16



3.3. Schnittstellen

Die Standard Schnittstelle auf dem Modul ist USB oder Ethernet. Optional kann das Gerät mit einer CAN oder RS-232 Schnittstelle erweitert werden.

3.3.1. USB

Technische Daten:

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

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

** durchschnittliche Zeit für 32-Bit Zugriffe

3.3.2. Ethernet

Technische Daten:

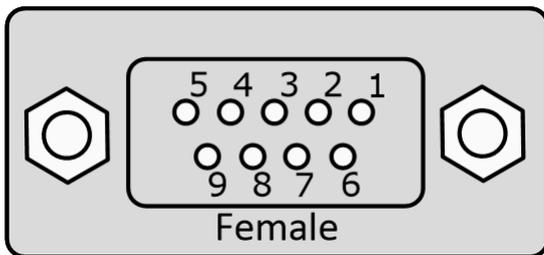
Standard:	Ethernet 100/10Mbit
Verbindungsaufbau	Ethernet / LAN Kabel / WiFi
Zugriffszeit PC auf Modul*:	Wire: 1,56 ms / Wireless: 4,06 ms

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

3.3.3. CAN

Pinbelegung D-SUB Buchse:

CAN-L	Pin 2
CAN-H	Pin 7
GND	Pin 5



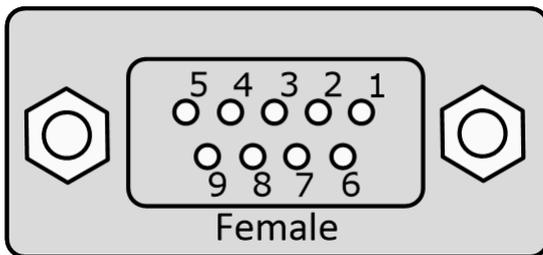
Technische Daten:

Standard:	ISO 11898
Geschwindigkeit:	1 Mbit/s, 500 Kbit/s, 250 Kbit/s, 125 Kbit/s, 100 Kbit/s, 50 Kbit/s, 20 Kbit/s, 10 Kbit/s
Verbindungsaufbau:	Offenes CAN Protokoll
Galvanische Trennung:	bis 1kV rms
Besonderheiten:	Automatisches Verarbeiten von CAN Paketen (Auto RX/TX mode)

3.3.4. RS-232

Pinbelegung D-SUB Buchse:

TX	Pin 2
RX	Pin 3
GND	Pin 5



Technische Daten:

Standard:	RS-232
Geschwindigkeit:	bis 115200 Baud

3.3.5. Schnittstellen nachrüsten

Das Gerät kann nachträglich mit einem CAN- oder RS-232 Interface Adapter bestückt werden. Beachten Sie hierbei bitte das Kapitel → **Sicherheitshinweise**

Für die Nachrüstung gehen Sie bitte folgendermaßen vor:

- Trennen Sie das Modul von der Stromzufuhr.
- Entfernen Sie eine der seitlichen Abdeckungen des Gehäuses durch lösen der vier Kreuzschrauben.
- Ziehen Sie nun die Leiterplatte vorsichtig aus dem Gehäuse und legen Sie diese auf einen festen und trockenen Untergrund.
- Stecken Sie anschließend den Interface-Adapter auf die 10-polige Stiftleiste, so dass sich die beiden Montageblöcke oberhalb der Befestigungslöcher befinden.
- Schrauben Sie den Adapter nun auf der Unterseite der Leiterplatte fest und ziehen danach die Schrauben der beiden Montageblöcke nochmals fest an.
- Der Einbau erfolgt in umgekehrter Reihenfolge. Achten Sie darauf, die Leiterplatte in die oberste Führungsschiene zu schieben.

Nach der Installation kann das Modul direkt in Betrieb genommen werden. Die Schnittstellen-Adapter werden automatisch initialisiert.

3.4. LEDs

Auf dem Modul befinden sich eine Vielzahl an Leuchtdioden. Es gibt 8 Status-LEDs und zusätzlich für jeden digitalen I/O Kanal jeweils eine LED.

3.4.1. Definition der LEDs

LED Status:

Signalisiert, ob sich das Modul im Bootloader oder in der Anwendung befindet.
→ siehe Kapitel "**Blinkverhalten LEDs**"

LED I/O Acc:

Blinkt, wenn eine Abfrage der I/O Kanäle stattfindet, oder die CPU Daten an diese sendet.

LED O.Auto-Off:

Leuchtet, wenn die Ausgänge durch ein Timeout Ereignis deaktiviert wurden.

LED I.Change:

Leuchtet, wenn an einem digitalen Eingang ein Signal erkannt wurde, welches zwischen den Auslesetakten aufgetreten ist.

LED Interf.Act:

Blinkt bei aktiver Kommunikation über eine der Schnittstellen.

LED Error:

Leuchtet, wenn ein Systemfehler im Modul aufgetreten ist.

LED LAN:

Zeigt den aktuellen Status der LAN-Verbindung an. → siehe Kapitel "**Blinkverhalten LEDs**"

LED WIFI:

Zeigt den aktuellen Status der WIFI-Verbindung an. → siehe Kapitel "**Blinkverhalten LEDs**"

I/O LEDs:

Leuchten bei anliegender Signalspannung an den digitalen Eingängen bzw. bei eingeschalteten digitalen Ausgängen.

3.4.2. Blinkverhalten der LEDs

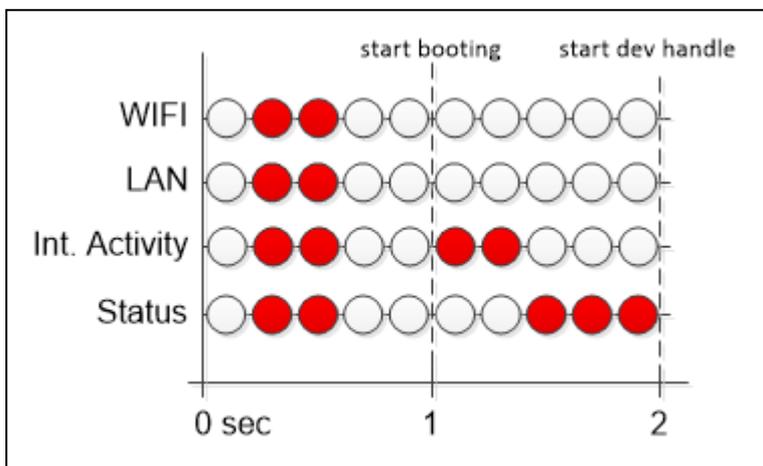
Nachfolgend werden die Blinksequenzen der Status LEDs dargestellt.

LED ON ●

LED OFF ○

1. Bootvorgang

Der Bootvorgang startet direkt nach dem Einschalten der Spannungsversorgung.

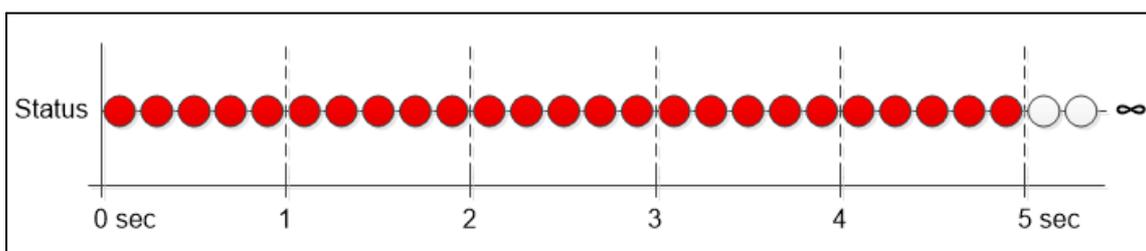


Die Bootvorgang-Sequenz wird einmalig durchlaufen.

2. Applikation oder Bootloader

2.1 Applikation

Der Bootvorgang wurde erfolgreich durchlaufen und das Produkt befindet sich nun in der Applikation. Das Produkt ist betriebsbereit.



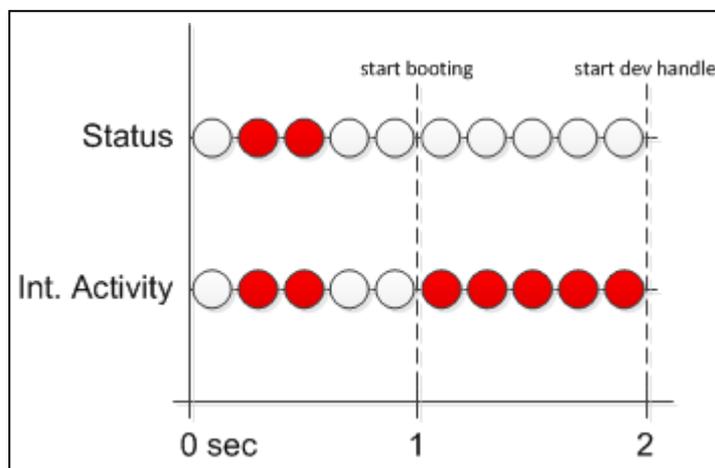
Die Status-LED leuchtet 5 Sekunden und erlischt für etwa 300ms. Die Applikations-Sequenz wiederholt sich.

2.2. Bootloader

Das Produkt befindet sich nach dem Bootvorgang im Bootloader. Die Anwendung wurde nicht geladen. Dies deutet auf einen Fehler in der Firmware hin.

Eine Aktualisierung der Firmware kann das Problem in der Regel beheben → **Firmware Update durchführen**

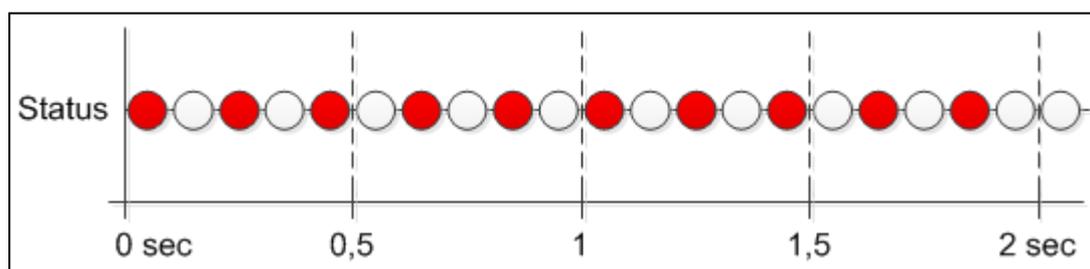
Die Status-LED leuchtet 2 Sekunden und erlischt für etwa 300ms. Die Bootloader-Sequenz wiederholt sich.



3. Bootloader erzwungen

Das Produkt kann mit dem DIP-Schalter 4 in den Bootloader gezwungen werden. → **siehe Kapitel "DIP Schalter"**

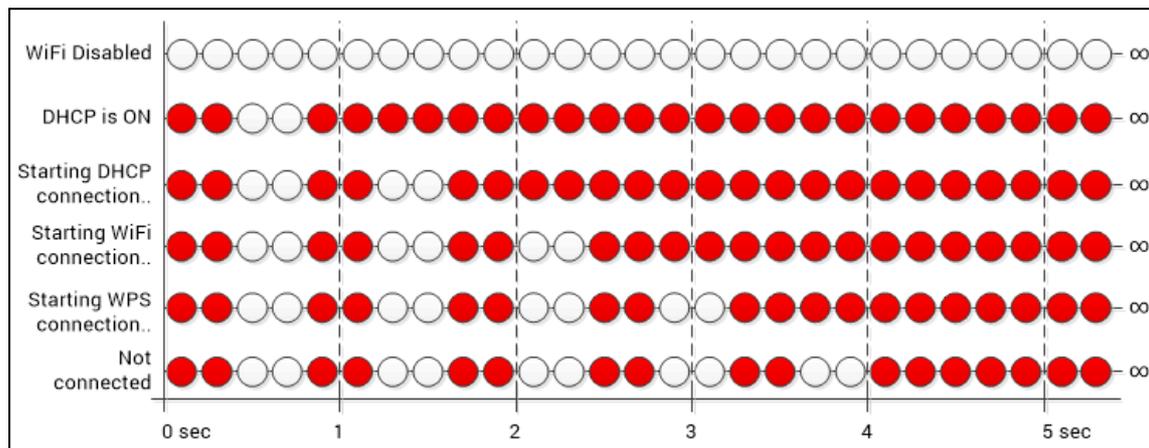
Diese Funktion dient nur zu Service-Zwecken.



Die Status-LED blinkt abwechselnd 2 Sekunden lang. Diese Sequenz wird nur einmalig durchlaufen.

4. LED WIFI

Die WIFI-LED zeigt den aktuellen Status der WIFI-Verbindung an.



WiFi Disabled

Die WiFi-Funktion ist deaktiviert. Die LED bleibt aus.

DHCP is ON

Eine Verbindung via WiFi wurde erfolgreich hergestellt. Die LED erlischt in ca. 5 Sekunden 1-mal.

Starting DHCP connection..

Es wird versucht eine Verbindung per DHCP herzustellen. Die LED erlischt in ca. 5 Sekunden 2-mal.

Starting WiFi connection..

Es wird versucht eine Verbindung per WiFi herzustellen. Die LED erlischt in ca. 5 Sekunden 3-mal.

Starting WPS connection..

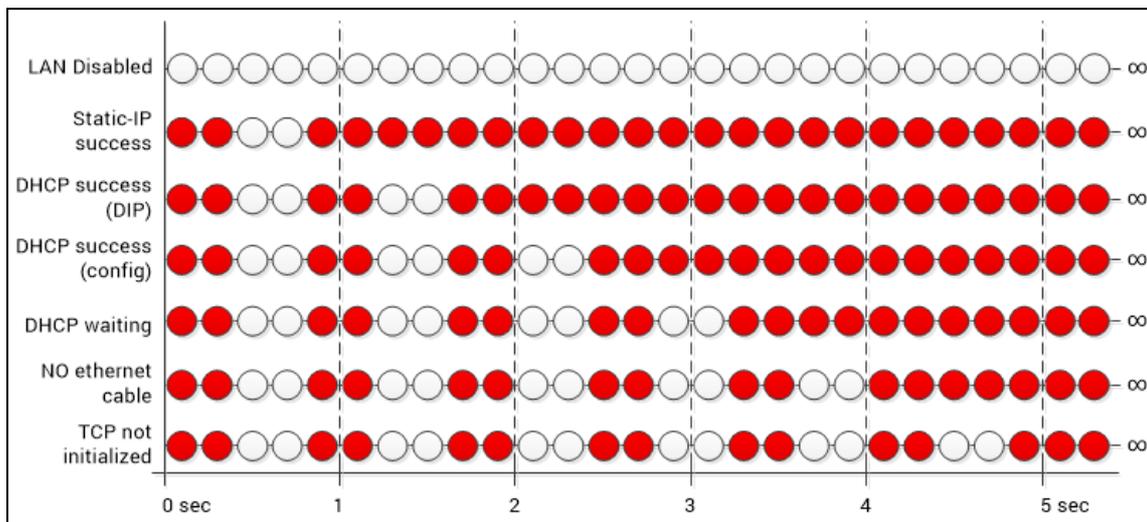
Es wird versucht eine Verbindung per WPS-Funktion herzustellen. Die LED erlischt in ca. 5 Sekunden 4-mal.

Not connected

Es kann keine Verbindung hergestellt werden. Die LED erlischt in ca. 5 Sekunden 5-mal.

5. LED-LAN

Die LAN-LED zeigt den aktuellen Status der LAN-Verbindung an.



LAN Disabled

Die LAN-Funktion ist deaktiviert. Die LED bleibt aus.

Static-IP Success

Eine erfolgreiche Verbindung via statischer IP wurde hergestellt. Die LED erlischt in ca. 5 Sekunden 1-mal.

DHCP success (DIP)

Eine erfolgreiche Verbindung via DHCP wurde hergestellt. DHCP wurde dabei über den DHCP-DIP-Schalter auf dem Board Ihres Moduls aktiviert. Die LED erlischt in ca. 5 Sekunden 2-mal.

DHCP success (config)

Eine erfolgreiche Verbindung via DHCP wurde hergestellt. DHCP wurde dabei über Software (**Modul Config** oder **Configuration Utility**) aktiviert. Die LED erlischt in ca. 5 Sekunden 3-mal.

DHCP waiting

Es wird versucht eine Verbindung über DHCP herzustellen. Die LED erlischt in ca. 5 Sekunden 4-mal.

NO ethernet cable

Das Ethernet-Kabel wurde entfernt oder steckt nicht richtig in der Ethernet-Buchse. Die LED erlischt in ca. 5 Sekunden 5-mal.

TCP not initialized

Es konnte keine Verbindung hergestellt werden. Die LED erlischt in ca. 5 Sekunden 6-mal.

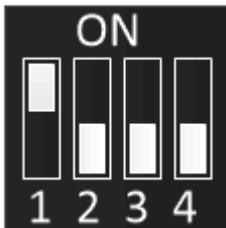
3.5. Schalter & Taster

3.5.1. Funktionen der DIP-Schalter

Mit Hilfe der DIP-Schalter können folgende Funktionen aktiviert oder deaktiviert werden:

- DIP 1: ON = DHCP on, OFF = DHCP off
- DIP 2: ON = Schreibschutz on, OFF = Schreibschutz off
- DIP 3: ON = Werkseinstellungen on, OFF = Werkseinstellungen off
- DIP 4: Bootloader Mode (Firmware wird gelöscht !!!)

DIP-Schalter Stellungen im Auslieferungszustand:



Hinweise:

Änderungen an den DIP-Schaltern werden erst nach einem Neustart des Moduls übernommen !

Die DIP-Schalter Einstellungen werden immer vor den Software-Einstellungen priorisiert !

Erläuterungen der DIP Schalter Funktionen:

DIP 1	Modus / Erklärung
ON	DHCP ist aktiviert Die Netzwerkeinstellungen IP, Subnetzmaske, DNS-Domain sowie Gateway werden über einen DHCP-Server aus Ihrem Netzwerk bezogen. DHCP wird erzwungen auch wenn DHCP per Software deaktiviert wurde.
OFF	DHCP deaktiviert Die im Module-Configuration-Memory hinterlegten Netzwerkeinstellungen werden verwendet. Mithilfe der DELIB-Software können diese Einstellungen bearbeitet und abgespeichert werden.

DIP 2	Modus / Erklärung
ON	Schreibschutz aktiviert Es können keine Konfigurationen der Netzwerkeinstellungen über das DELIB-Configuration Utility oder das Module Config vorgenommen werden.
OFF	Schreibschutz deaktiviert Es können Konfigurationen der Netzwerkeinstellungen über das DELIB-Configuration Utility oder dem Module Config vorgenommen werden.

DIP 3	Modus / Erklärung		
ON	<p>Werkseinstellungen aktiviert</p> <p>Die im Module-Configuration-Memory gespeicherten Werte (IP-Adresse, Gateway, Subnetzmaske, DHCP) werden beim Modulstart ignoriert.</p> <p>Das Modul startet mit den folgenden Werkseinstellungen.</p> <table border="0" data-bbox="453 779 1362 1249"> <tr> <td data-bbox="453 779 900 1249"> <p>Ethernet/LAN</p> <p>Boardname: Produktabhängig (z.B.: WEU-O8-R8)</p> <p>Write Protection: OFF</p> <p>DHCP: OFF</p> <p>IP: 192.168.1.1</p> <p>Subnetzmaske: 255.255.255.0</p> <p>Gateway: 192.168.1.254</p> <p>Port: 9912</p> </td> <td data-bbox="900 779 1362 1249"> <p>WiFi</p> <p>Boardname: Produktabhängig (z.B.: WEU-O8-R8_W)</p> <p>WLAN active: OFF</p> <p>Router name: DefaultSSID</p> <p>Password: DefaultPWD</p> </td> </tr> </table> <p>Die im Module-Configuration-Memory gespeicherten Werte werden nicht verändert.</p> <p>Wichtig!</p> <p>Ist ebenfalls der DHCP-DIP-Schalter aktiviert, werden die Netzwerkeinstellungen des DHCP-Servers verwendet.</p>	<p>Ethernet/LAN</p> <p>Boardname: Produktabhängig (z.B.: WEU-O8-R8)</p> <p>Write Protection: OFF</p> <p>DHCP: OFF</p> <p>IP: 192.168.1.1</p> <p>Subnetzmaske: 255.255.255.0</p> <p>Gateway: 192.168.1.254</p> <p>Port: 9912</p>	<p>WiFi</p> <p>Boardname: Produktabhängig (z.B.: WEU-O8-R8_W)</p> <p>WLAN active: OFF</p> <p>Router name: DefaultSSID</p> <p>Password: DefaultPWD</p>
<p>Ethernet/LAN</p> <p>Boardname: Produktabhängig (z.B.: WEU-O8-R8)</p> <p>Write Protection: OFF</p> <p>DHCP: OFF</p> <p>IP: 192.168.1.1</p> <p>Subnetzmaske: 255.255.255.0</p> <p>Gateway: 192.168.1.254</p> <p>Port: 9912</p>	<p>WiFi</p> <p>Boardname: Produktabhängig (z.B.: WEU-O8-R8_W)</p> <p>WLAN active: OFF</p> <p>Router name: DefaultSSID</p> <p>Password: DefaultPWD</p>		
OFF	<p>Werkseinstellungen deaktiviert</p> <p>Das Module Startet mit dem im Module-Configuration-Memory konfigurierten Einstellungen.</p>		

DIP 4	Modus / Erklärung
ON	<p>Bootloader aktiviert</p> <p>Das Modul bleibt zwangsweise im Bootloader. Die Firmware wird nicht gestartet.</p> <p>ACHTUNG!!!</p> <p>Wird das Modul mit DIP 4 auf ON gestartet, wird die Firmware des Modules gelöscht und muss neu installiert/geflashed werden.</p> <p>Alle im Module-Configuration-Memory gespeicherten Einstellungen bleiben erhalten.</p>
OFF	<p>Bootloader deaktiviert</p> <p>Das Modul startet normal.</p>

3.5.2. Funktionen des CFG-Tasters

Mit Hilfe des auf dem Modul befindlichen CFG-Tasters lassen sich verschiedene WiFi-Modi einstellen.

- WiFi enable
- WiFi disable
- WPS Start

Halten Sie den Taster so lange gedrückt, bis die Status LED anfängt zu blinken. Lassen Sie den Taster los, um den aktuellen Modus zu übernehmen.

Die Blinkfrequenz zeigt an, welcher Modus aktuell ausgewählt wurde.

nach x Sekunden	Blinkverhalten Status-LED	Modus
0 1 2	aus	
3 4 5 6	schnelles Blinken	WiFi enable
7 8 9 10 11 12	langsames Blinken	WPS Start
13 14 15, ...	ganz langsames Blinken	WiFi disable

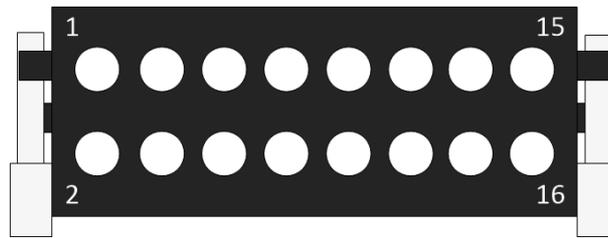
3.6. Analoge Ausgänge

3.6.1. Technische Daten

Technische Daten DA Wandler

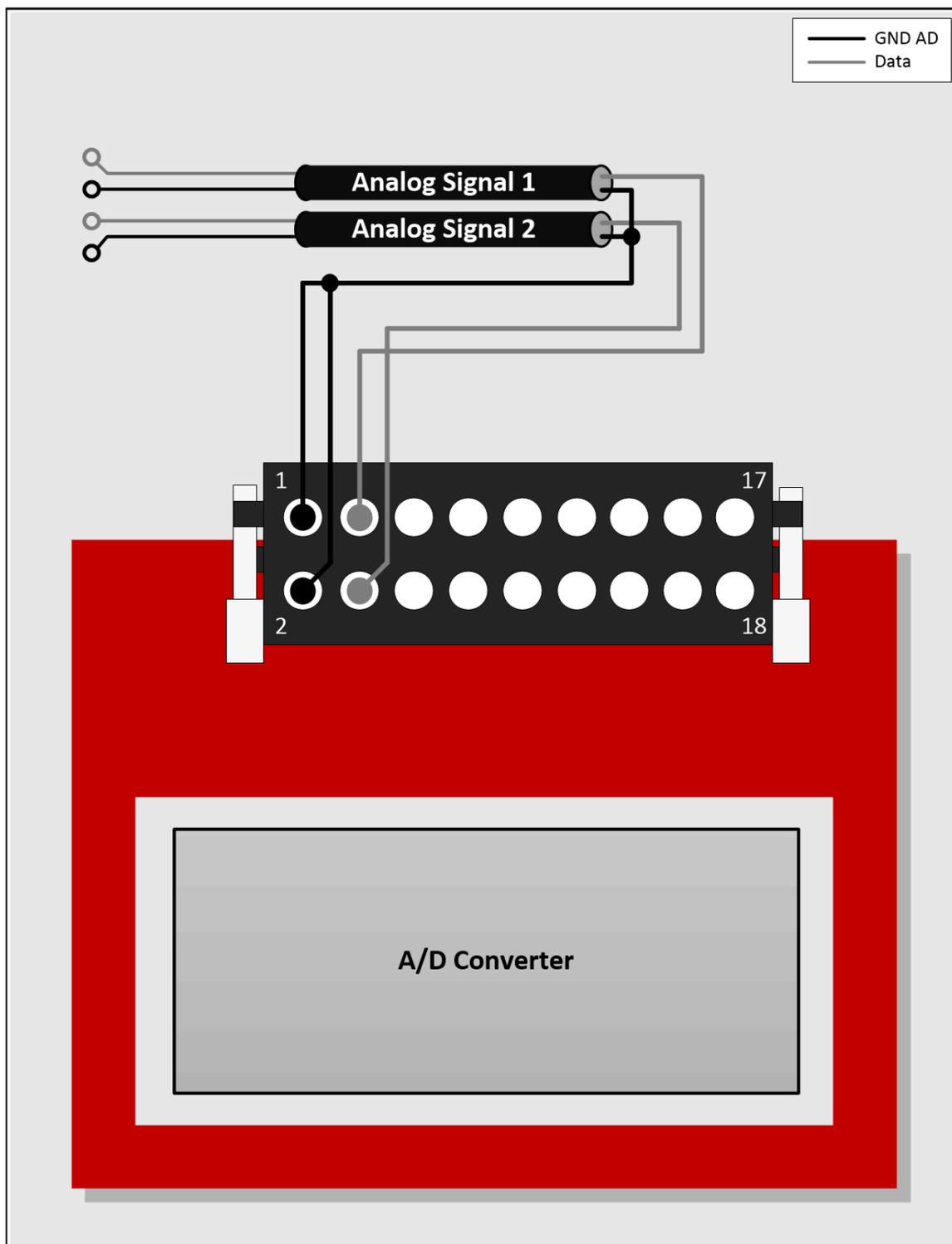
Anzahl Kanäle:	4 mit je 16-Bit Auflösung
Spannungsbereiche:	0-5V, 0-10V, $\pm 5V$, $\pm 10V$
Galvanische Trennung:	max. 500V
R Last:	2 kOhm
Relative Accuracy:	Min: -16 LSB / Max: +16 LSB
Bipolar Zero Error (T = 25°C):	± 4 ppm FSR/°C
Zero-Scale Error (T = 25°C):	± 2 ppm FSR/°C
Full-Scale Error Temp. Drift:	± 1 ppm/°C

3.6.2. Pinbelegung



Analog Output Channel	Pin	Analog Output Channel	Pin
AGND	1	Output Channel 1	2
AGND	3	Output Channel 2	4
AGND	5	Output Channel 3	6
AGND	7	Output Channel 4	8
nicht belegt	9	nicht belegt	10
nicht belegt	11	nicht belegt	12
nicht belegt	13	nicht belegt	14
nicht belegt	15	nicht belegt	16

3.6.3. Anschlussbeispiel für Spannungsausgabe bei einem AD16-DA4



3.7. Analoge Eingänge

3.7.1. Technische Daten

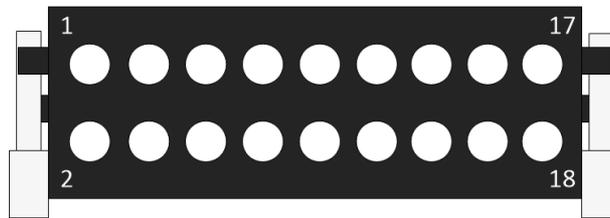
Technische Daten AD Wandler

Anzahl Kanäle:	16 mit je 16-Bit Auflösung
Spannungsbereiche:	0-5V, 0-10V, ±5V, ±10V
Strombereich:	0..50mA
Galvanische Trennung:	max. 500V zur Spannungsversorgung des Moduls
Eingangswiderstand:	> 500kOhm
Integral Linearity Error:	Min: -1.5 LSB / Max: +1.5 LSB
Bipolar Full-Scale Error:	Min: – 50 LSB / Max: +50 LSB
Unipolar Full-Scale Error:	Min: – 70 LSB / Max: +70 LSB
Genauigkeit:	+3 ppm/C°
Zero Error Temp. Drift:	+1 ppm/C°
Full-Scale Error Temp. Drift:	+1 ppm/°C
Wandlungsrate:	4µs

Hinweis!

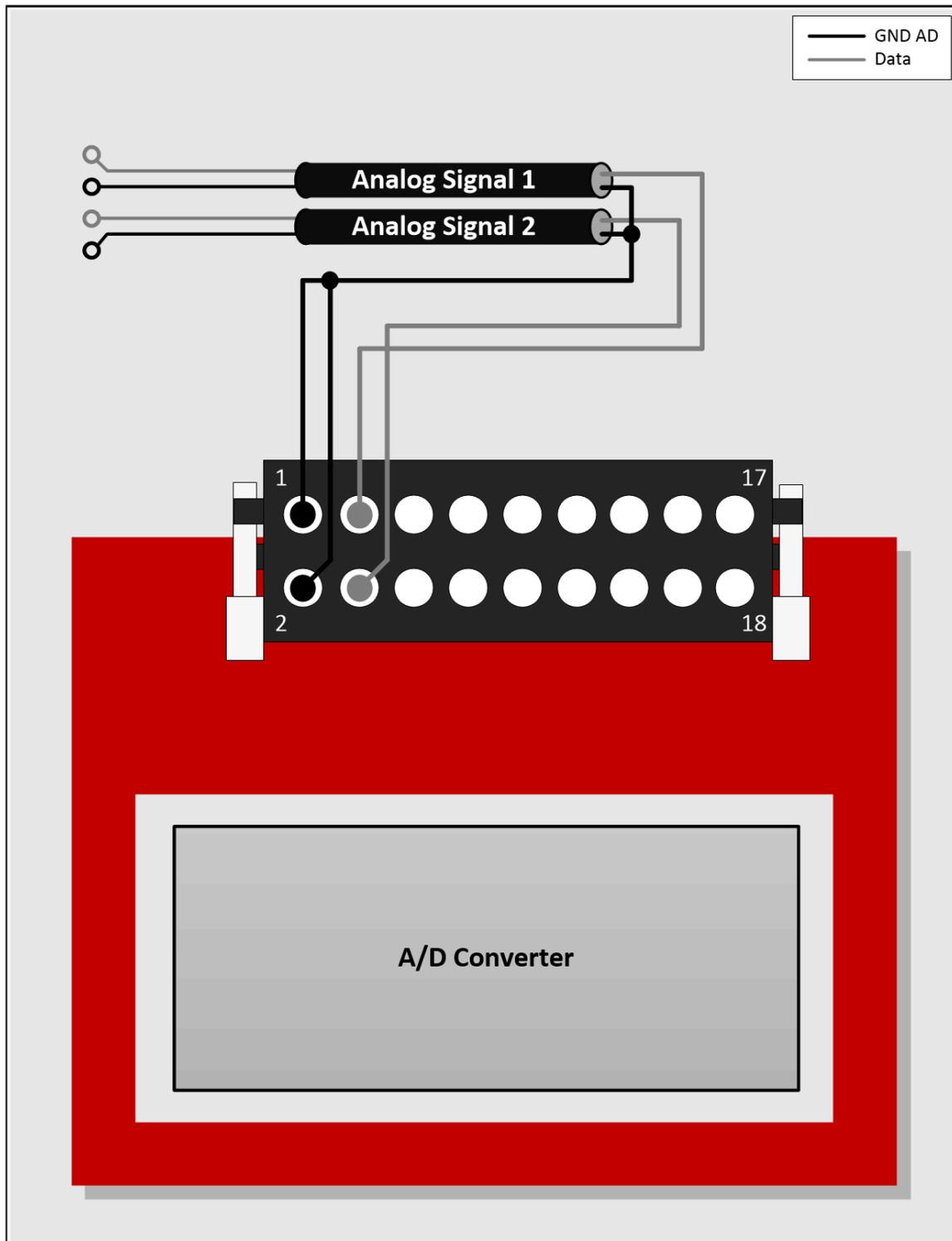
Für die Strommessung müssen separate Shunt Widerstände auf das Modul gesteckt werden. Diese sind als Zubehör bei uns erhältlich. Die Steckplätze dafür befinden sich links neben dem 18 poligen Steckverbinder. Es kann entweder nur Spannung oder nur Strom gemessen werden.

3.7.2. Pinbelegung

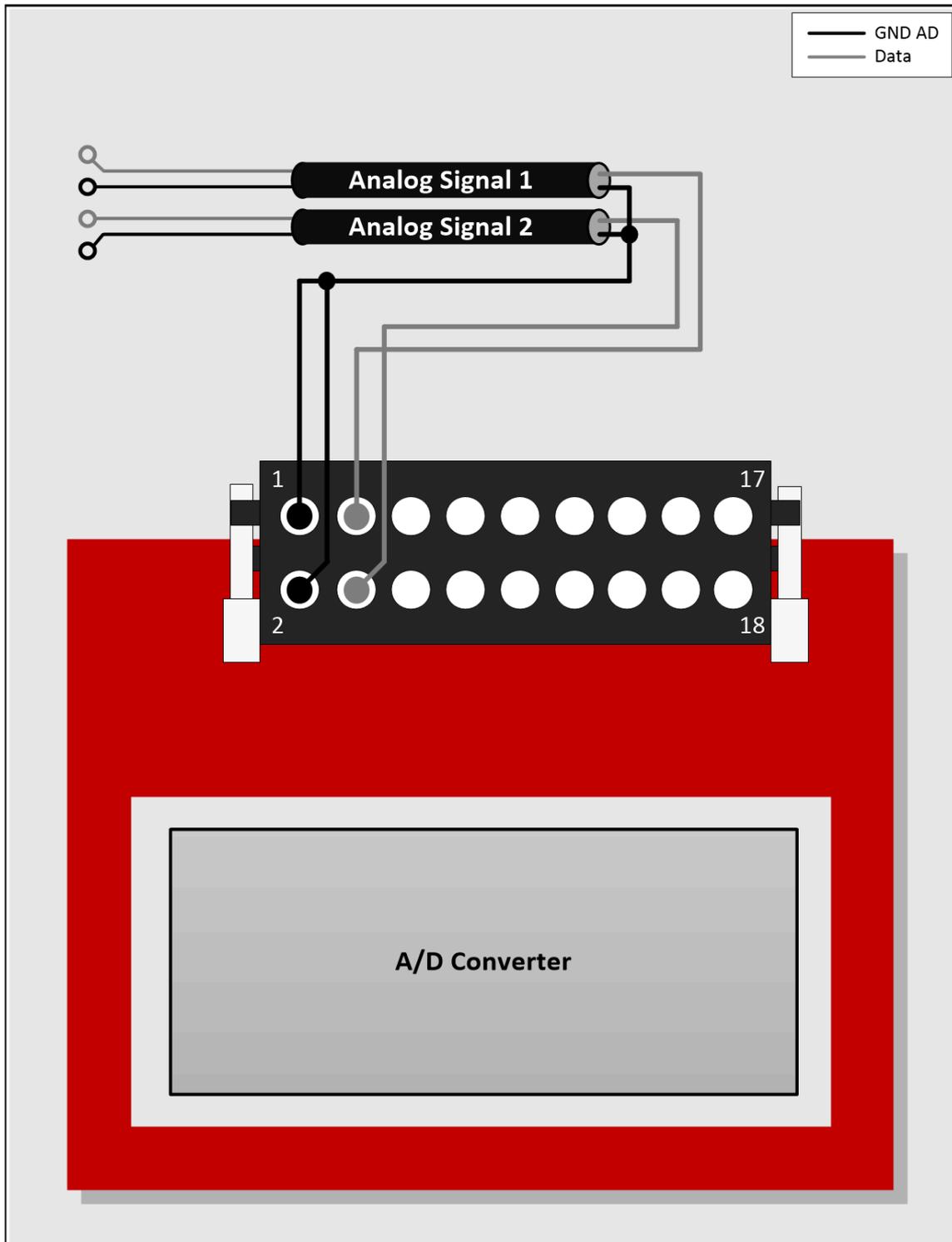


Analog Input Channel	Pin	Analog Input Channel	Pin
AGND	1	AGND	2
Input Channel 1	3	Input Channel 2	4
Input Channel 3	5	Input Channel 4	6
Input Channel 5	7	Input Channel 6	8
Input Channel 7	9	Input Channel 8	10
Input Channel 9	11	Input Channel 10	12
Input Channel 11	13	Input Channel 12	14
Input Channel 13	15	Input Channel 14	16
Input Channel 15	17	Input Channel 16	18

3.7.3. Anschlussbeispiele für Spannungsmessung bei einem AD16-DA4



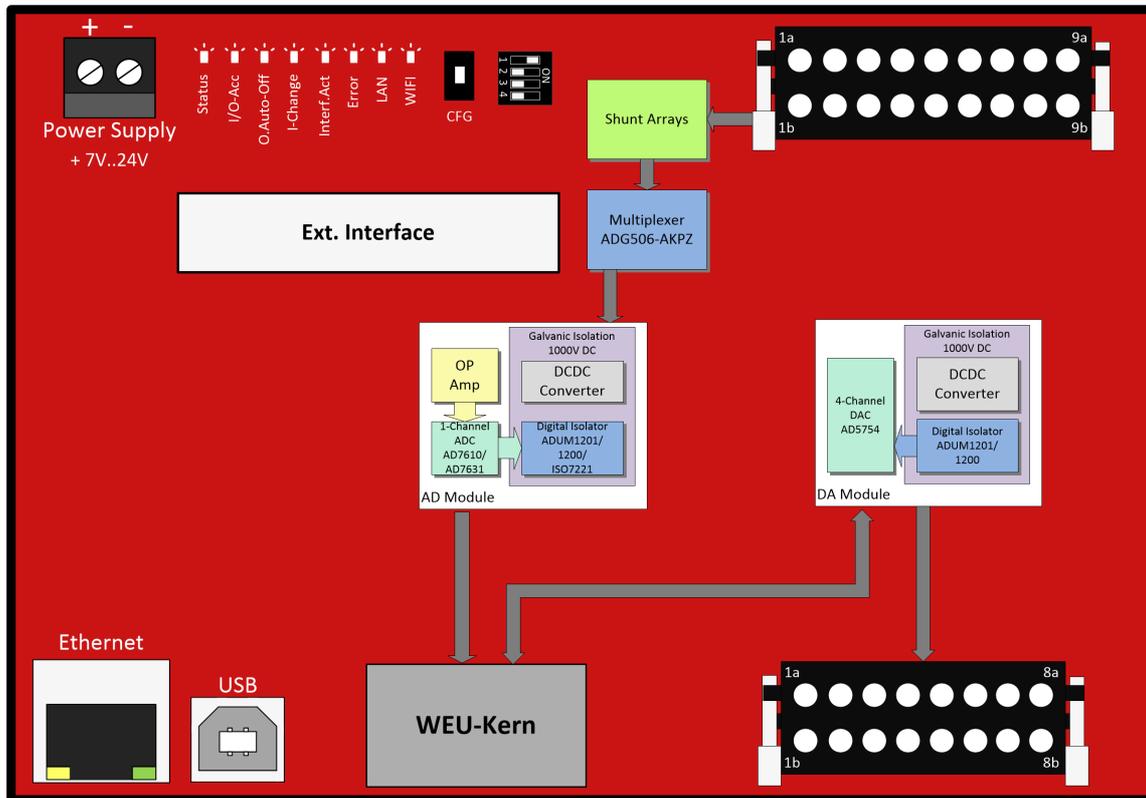
3.7.4. Anschlussbeispiel für Strommessung bei einem AD16-DA4



Hinweis:

Für die Strommessung müssen separate Shunt Widerstände auf das Modul gesteckt werden. Diese sind als Zubehör bei uns erhältlich. Die Steckplätze dafür befinden sich links neben dem 18 poligen Steckverbinder.

3.8. Blockschaltbild eines AD16-DA4



3.9. Digitale Ausgänge

3.9.1. Technische Daten

Technische Daten Relais 1A

Anzahl Kanäle pro Block:	8
Typ:	Schließer (NO)
Max. Schaltspannung:	36V AC / DC
Max. Schaltstrom:	0.5A AC / DC
Max. Transportstrom:	1A AC / DC
Max. Schaltleistung:	10W

Technische Daten Relais 3A

Anzahl Kanäle pro Block:	8
Typ:	Schließer (NO)
Max. Schaltspannung:	48V AC / DC
Max. Schaltstrom:	3A AC / DC
Max. Transportstrom:	3A AC / DC
Max. Schaltleistung:	90W

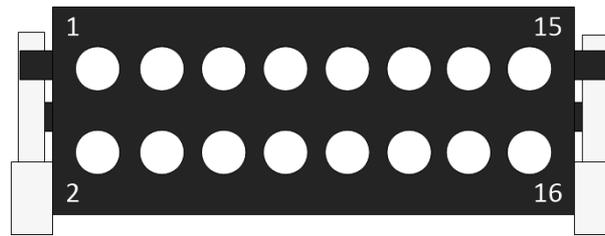
Technische Daten MOSFET 2A

Anzahl Kanäle pro Block:	8
Typ:	P-Ch.
Max. Schaltspannung:	48V DC
Min. Schaltspannung:	2.8V
Max. Schaltstrom:	2A DC
Max. Schaltleistung:	60W

Zusätzliche Funktionen

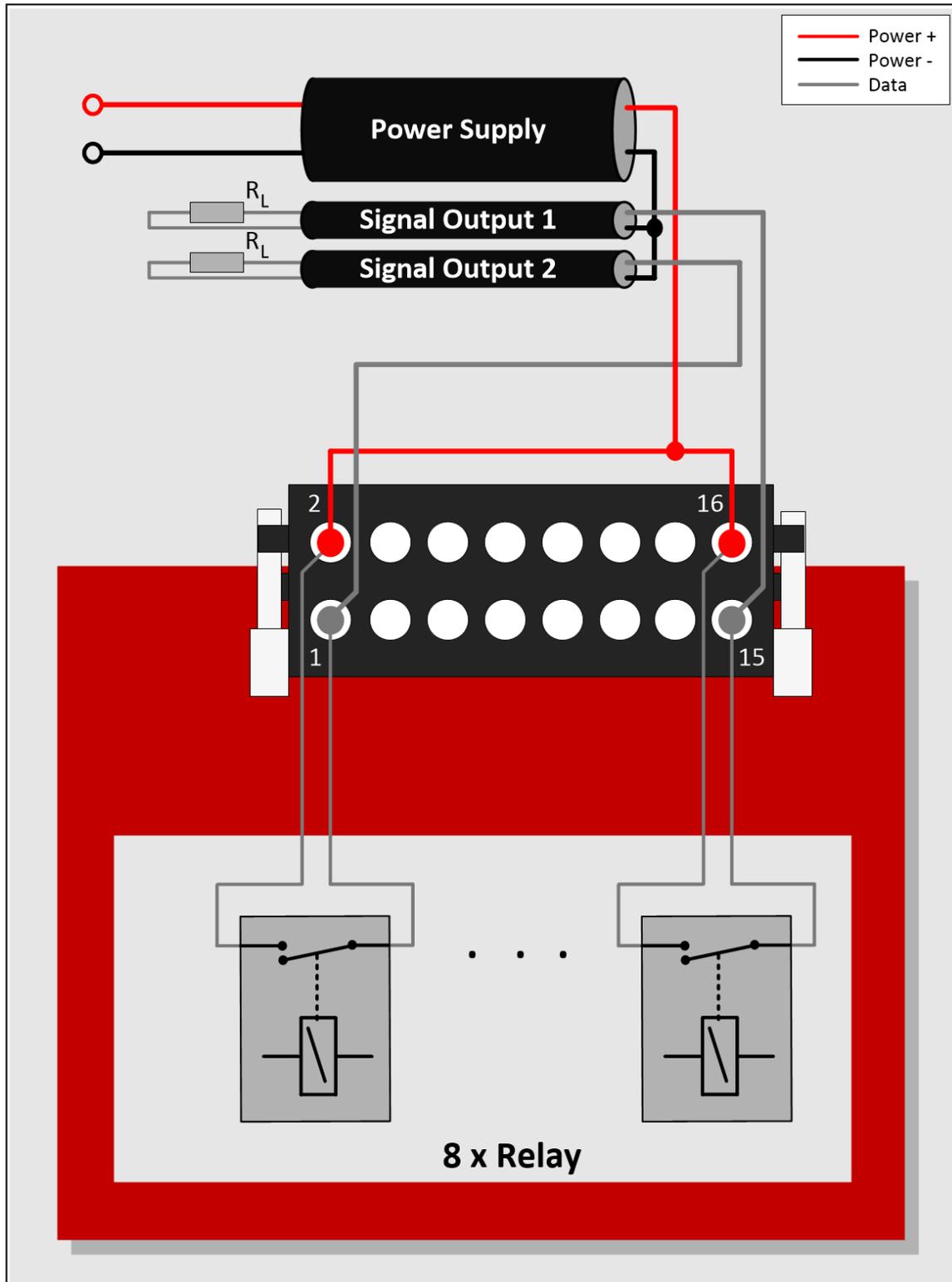
Timeout Schutz

3.9.2. Pinbelegung

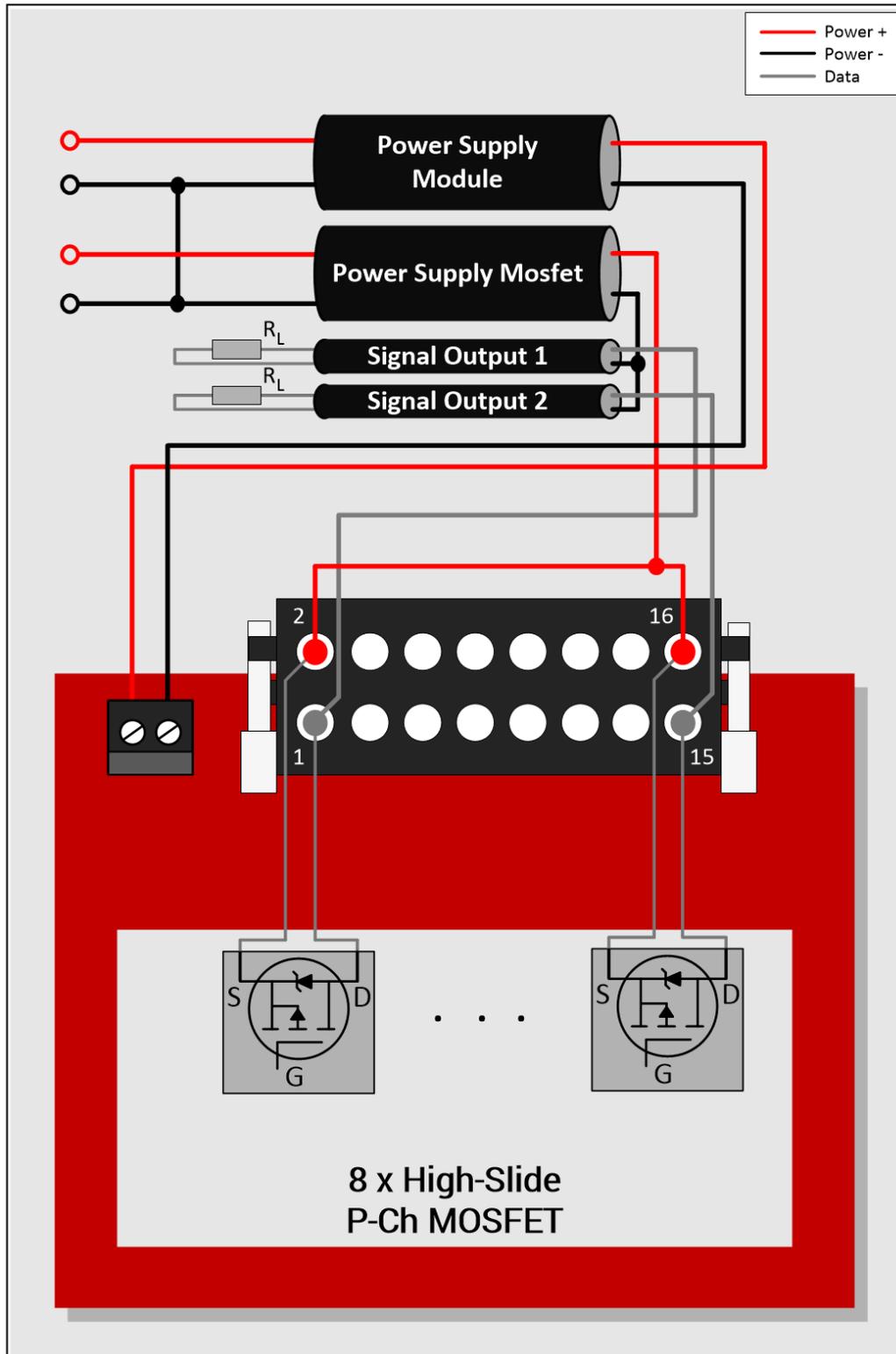


Digital Output Channel	Pin	
Output Channel 1	1	2
Output Channel 2	3	4
Output Channel 3	5	6
Output Channel 4	7	8
Output Channel 5	9	10
Output Channel 6	11	12
Output Channel 7	13	14
Output Channel 8	15	16

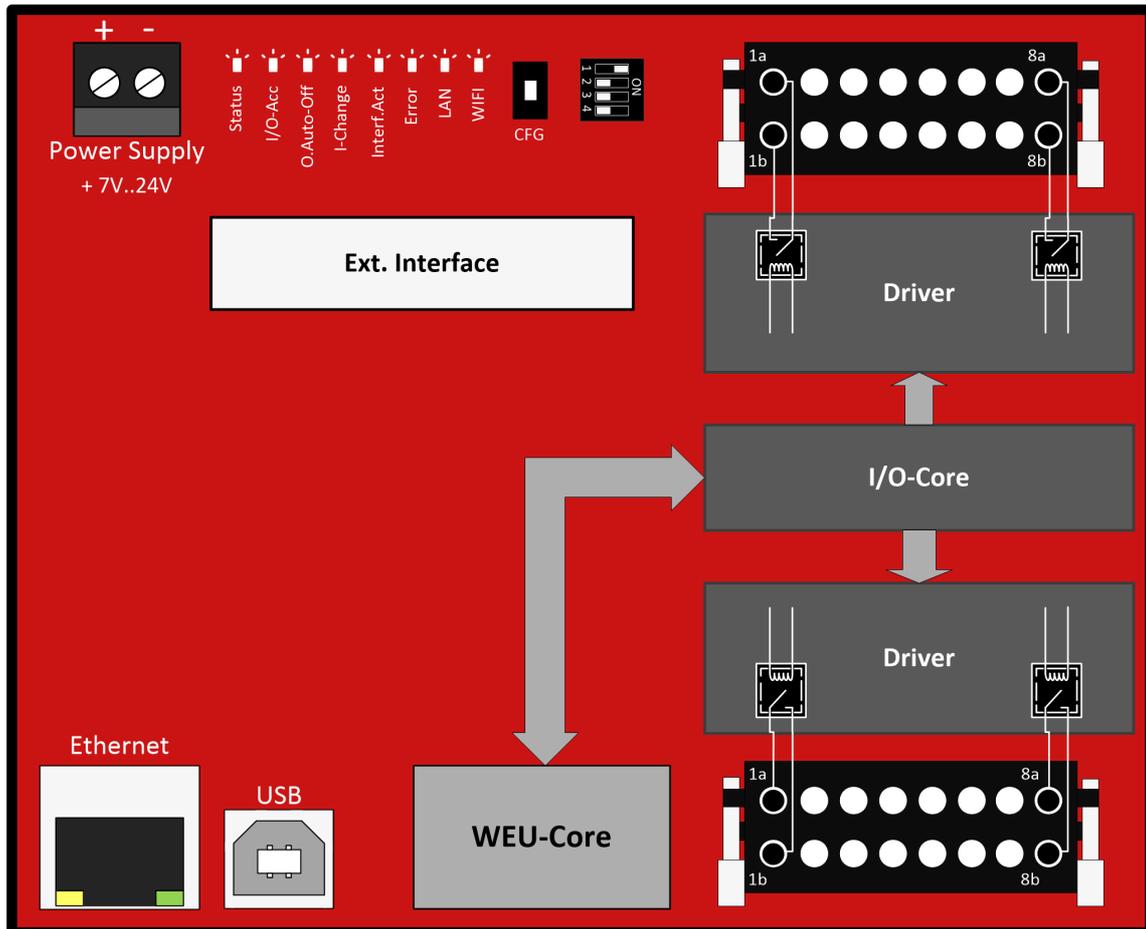
3.9.3. Anschlussbeispiel für Relais



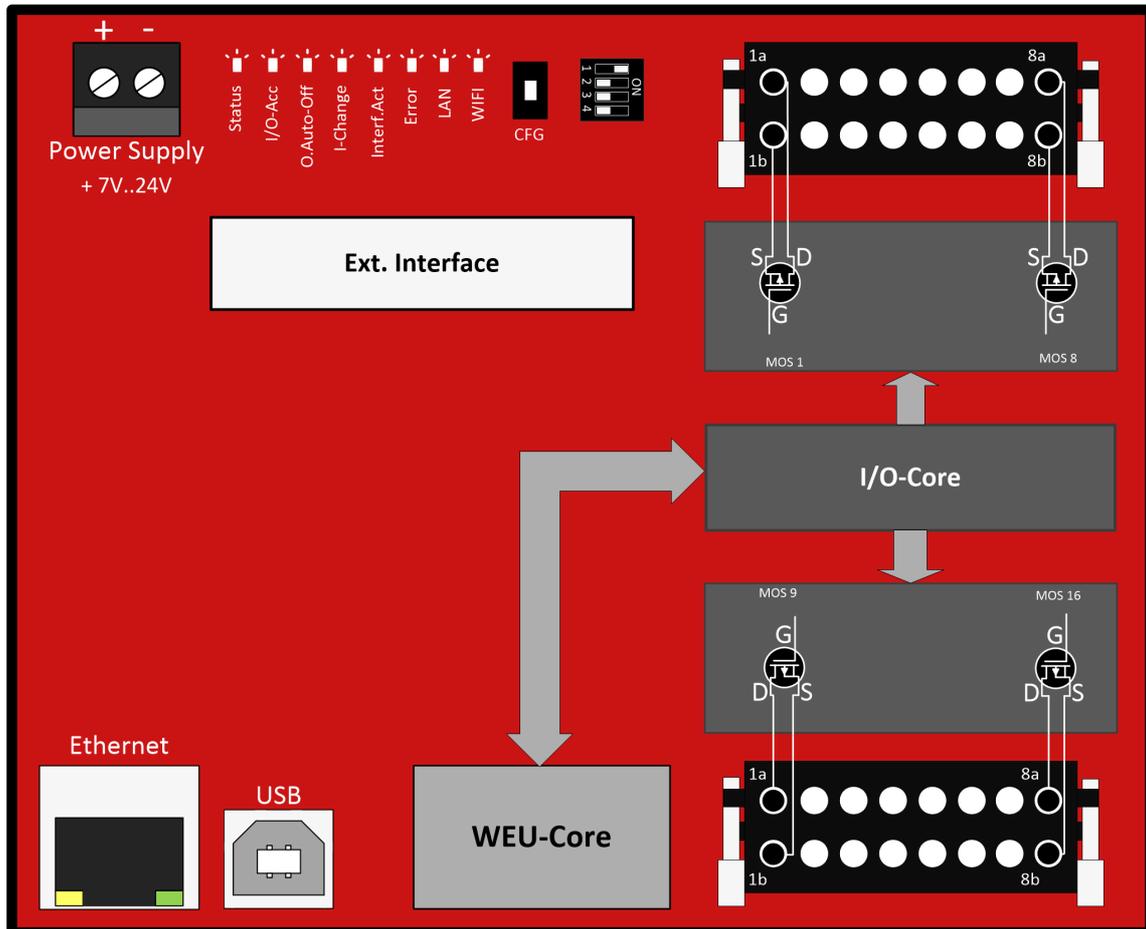
3.9.4. Anschlussbeispiel für MOSFET



3.9.5. Blockschaltbild eines Relais-Moduls



3.9.6. Blockschaltbild eines MOSFET-Moduls



3.9.7. Timeout Funktion

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

Funktionen:

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

Näheres siehe Kapitel → **Ausgabe-Timeout verwalten**

3.10. Digitale Eingänge

3.10.1. Technische Daten

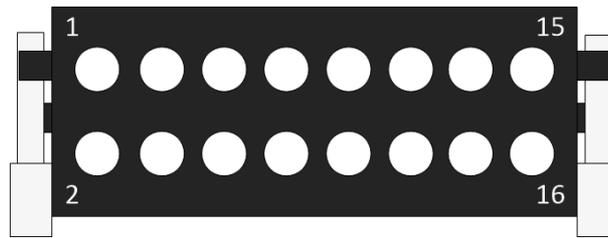
Technische Daten Optokoppler

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

Zusätzliche Funktionen

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

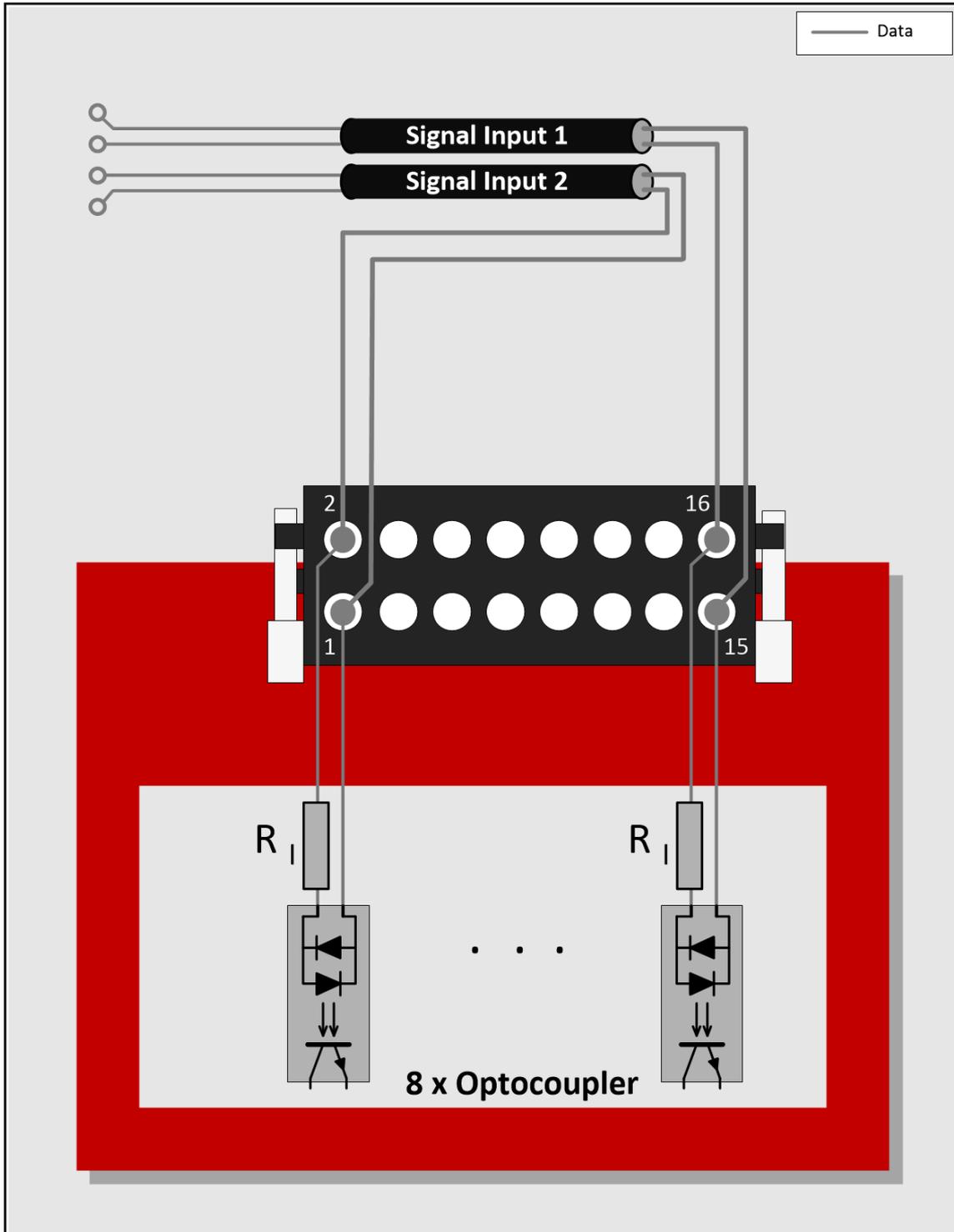
3.10.2. Pinbelegung



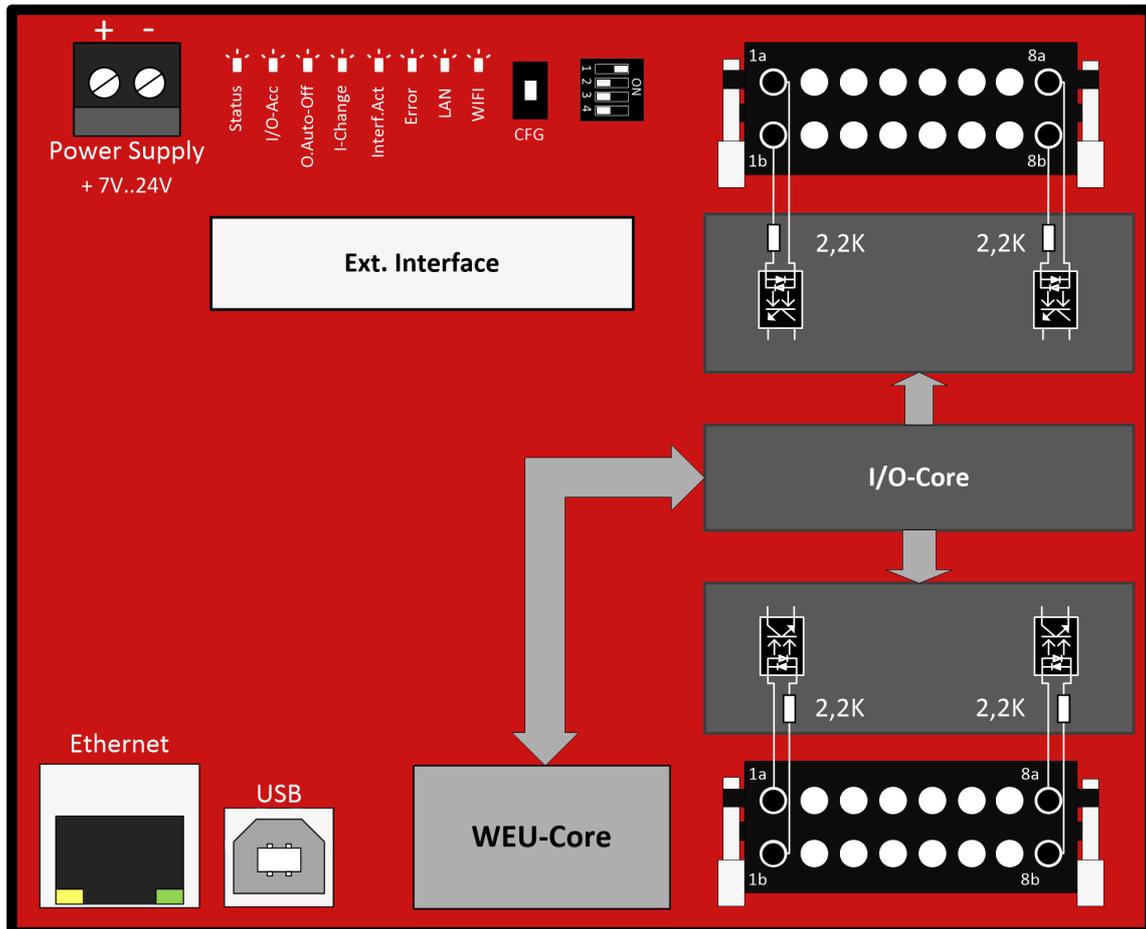
Digital Input Channel	Pin	
Input Channel 1	1	2
Input Channel 2	3	4
Input Channel 3	5	6
Input Channel 4	7	8
Input Channel 5	9	10
Input Channel 6	11	12
Input Channel 7	13	14
Input Channel 8	15	16

3.10.3. Anschlussbeispiel eines Optokopplers

Beim Anschluss einer Gleichspannungsquelle kann die Polarität vernachlässigt werden.



3.10.4. Blockschaltbild eines Optokoppler-Moduls

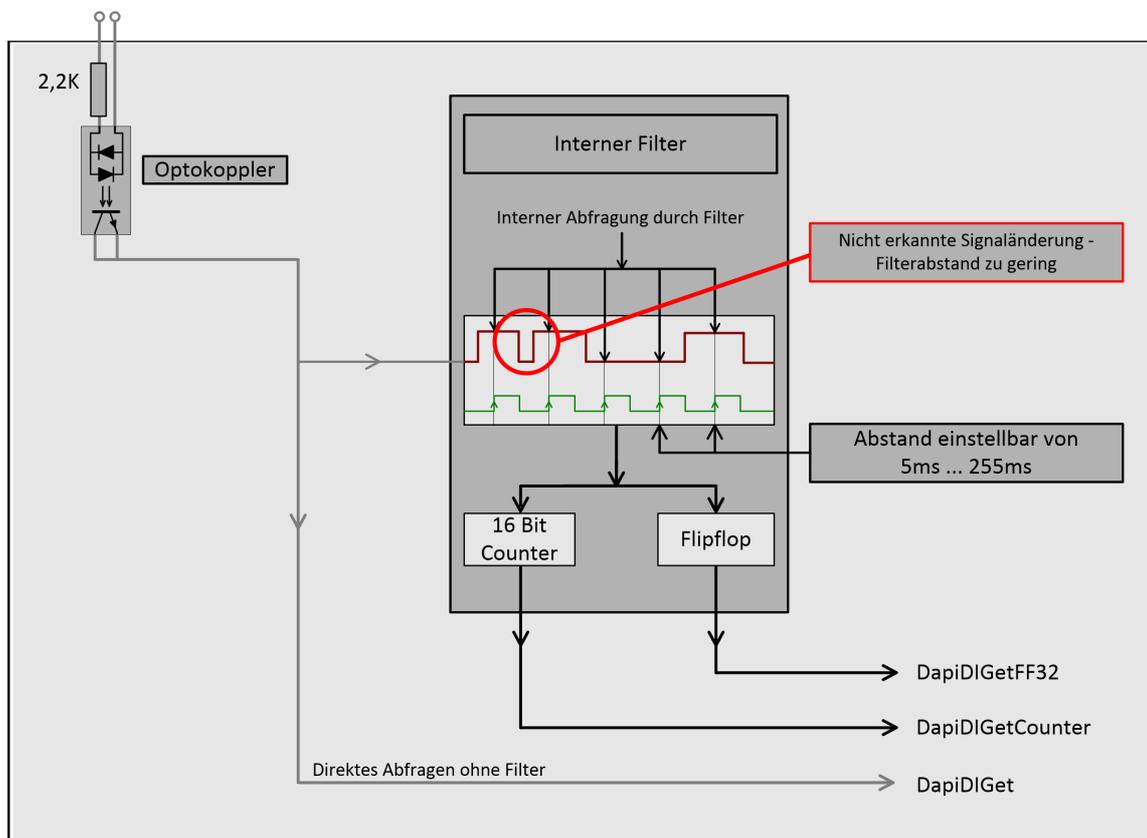


3.10.5. Eingangsfiler

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

Siehe auch Kapitel → **Digitale Eingänge Lesen**

Schematische Ansicht des Filters:



3.10.6. Zustandsänderungen überwachen

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

Siehe Kapitel: → **Digitale Eingänge lesen** → **DapiDiGetFF32**

Software Beschreibung



4. Software Beschreibung

4.1. Benutzen unserer Produkte

4.1.1. Ansteuerung über unsere DELIB Treiberbibliothek

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

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

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

4.1.2. Ansteuerung über mitgelieferte Testprogramme

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

Für ausführliche Informationen siehe Kapitel **DELIB_Module_Config**.

4.1.3. 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

Online HTML-Manual:

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.

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

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

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

Definition (Windows)

delib_cli command channel [value / unit ["nounit"]]

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

Parameter

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

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

Return-Wert

Zustand der gelesenen digitalen Eingänge

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

Zustand der FlipFlips der digitalen Eingänge

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

Zustand der gelesenen analogen Eingänge

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

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

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

4.1.4.1. Konfiguration des DELIB CLI

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

Konfiguration unter Windows

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

Inhalt der "delib_cli.cfg":

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

moduleID

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

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

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

moduleNR

Die moduleNR wird im DELIB Configuration Utility vergeben.

Diese Nummer dient zur Identifizierung identischer Hardware.

Der Standardwert ist die 0.

RO-ETH_ipAddress

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

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

4.1.4.2. DELIB CLI Beispiele

Digitale Ausgänge

```
delib_cli DO1 17 1
```

→ schaltet das 18. digitale Relais an

```
delib_cli DO1 3 0
```

→ schaltet das 4. digitale Relais aus

```
delib_cli DO8 0 255
```

→ schaltet die digitalen Relais 1 bis 8 an

```
delib_cli DO16 0 0
```

→ schaltet die digitalen Relais 1 bis 16 aus

```
delib_cli DO16 16 65535
```

→ schaltet die digitalen Relais 17 bis 32 an

```
delib_cli DO32 0 4294967295
```

→ schaltet die digitalen Relais 1 bis 32 an

Digitale Eingänge

```
delib_cli DI1 3
```

Beispiel eines Rückgabewertes: 1

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

```
delib_cli DI8 0 hex
```

Beispiel eines Rückgabewertes: **0xC8**

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

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

```
delib_cli DI16 0 hex
```

Beispiel eines Rückgabewertes: **0xE0C0**

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

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

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 angehängen 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

4.1.5. Ansteuerung über grafische Anwendungen

4.1.5.1. LabVIEW

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

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

4.1.5.2. ProfiLab

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

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

Die folgenden I/Os werden unterstützt:

Digitale Ein-/Ausgänge

- Relais
- MOSFET
- Optokoppler
- Bistabile-Relais

Analoge Ein-/Ausgänge

- Analog zu digital Wandler
- Digital zu analog Wandler

TTL-I/Os

- 8/32/64 TTL-Kanäle

4.1.5.3. Licht24 Pro

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

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

4.1.6. Einbinden der DELIB in Programmiersprachen

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

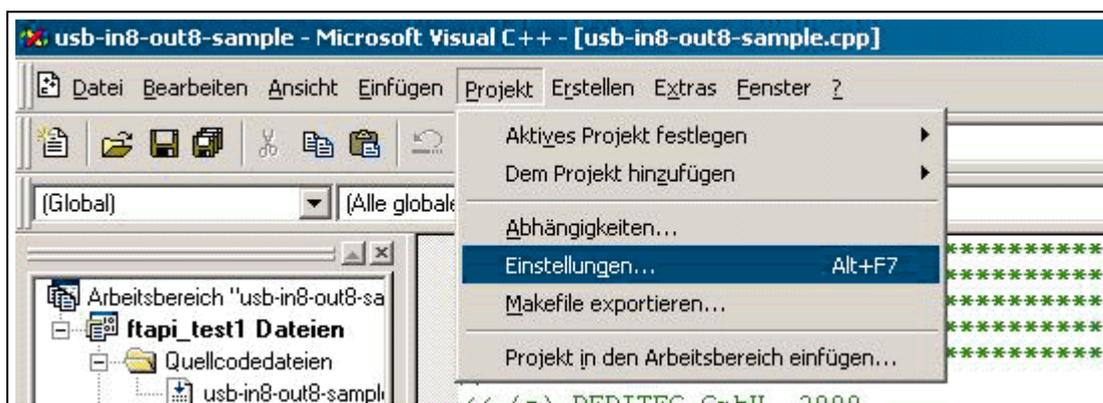
Zur Erleichterung für Verweise auf das DELIB-Include und das DELIB-Lib Verzeichnis werden bei installation der DELIB Umgebungsvariablen definiert.

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

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

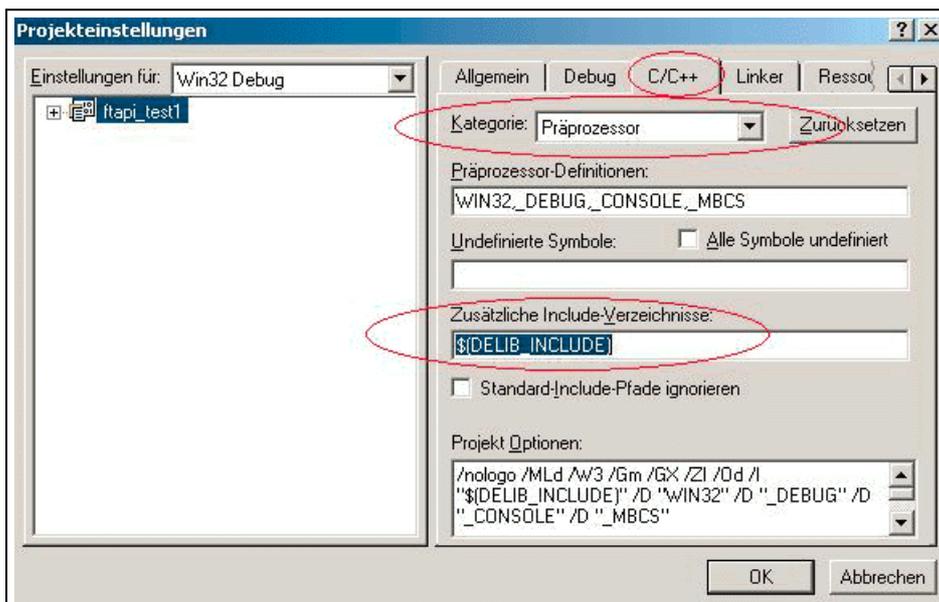
Diese werden im Folgenden in den Projekteinstellungen des Compilers eingetragen.

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



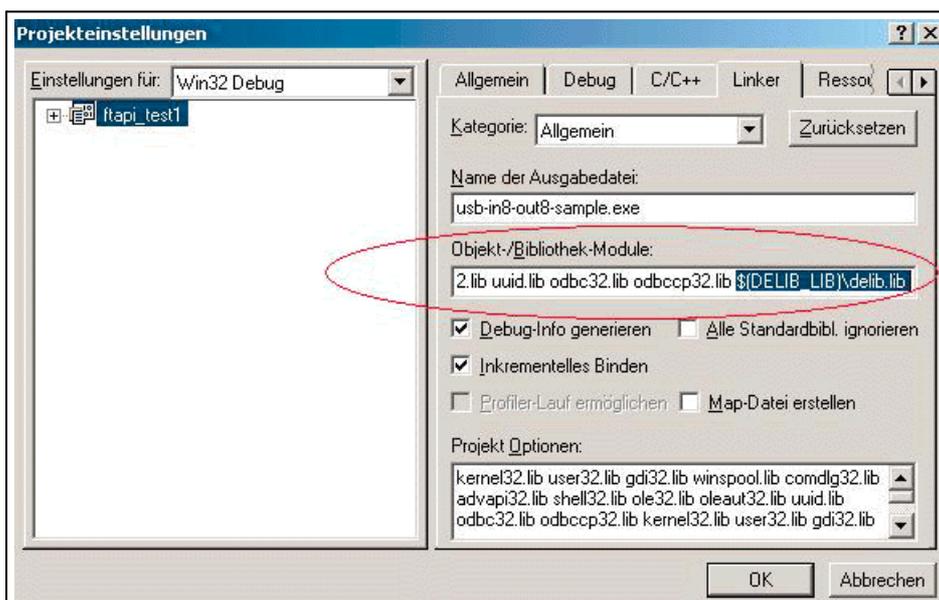
DELIB.H Eintrag in den Visual-C/C++ Projekt Einstellungen

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



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

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



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

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

32 Bit DELIB Installation

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

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

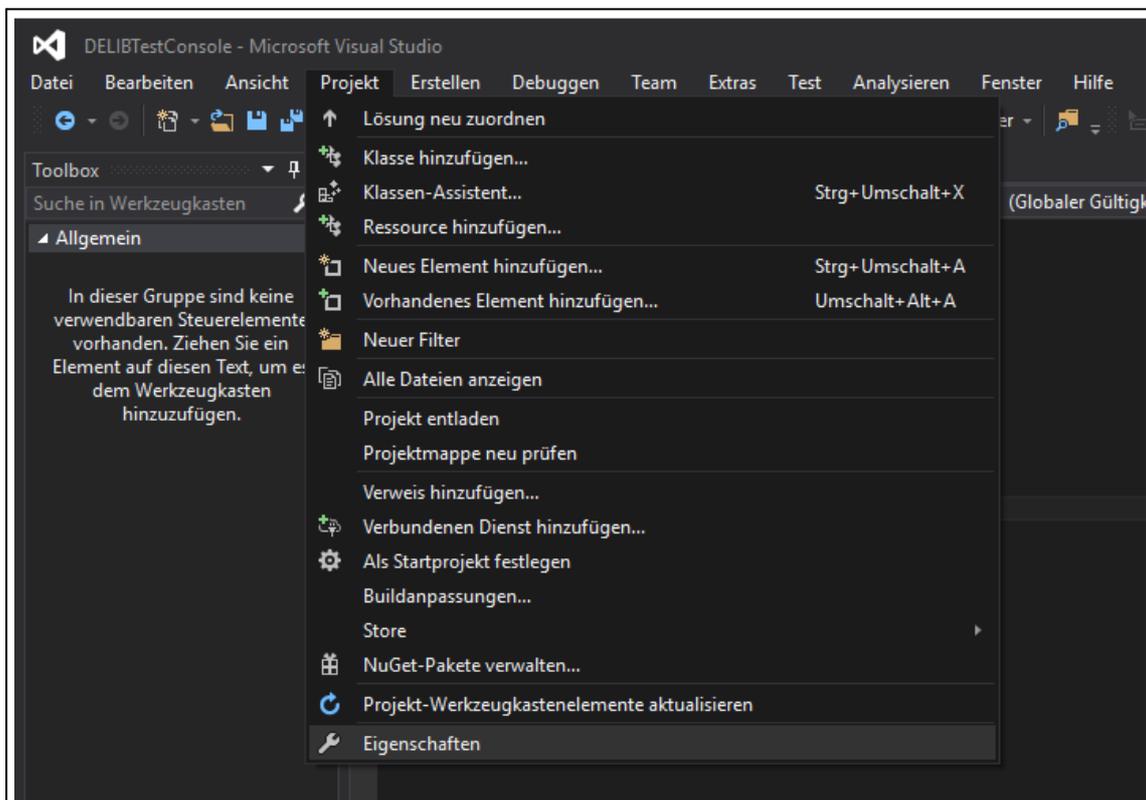
64 Bit DELIB Installation

DELIB64_LIB = C:\Programme\DEDITEC\DELIB64\lib

DELIB64_INCLUDE = C:\Programme\DEDITEC\DELIB64\include

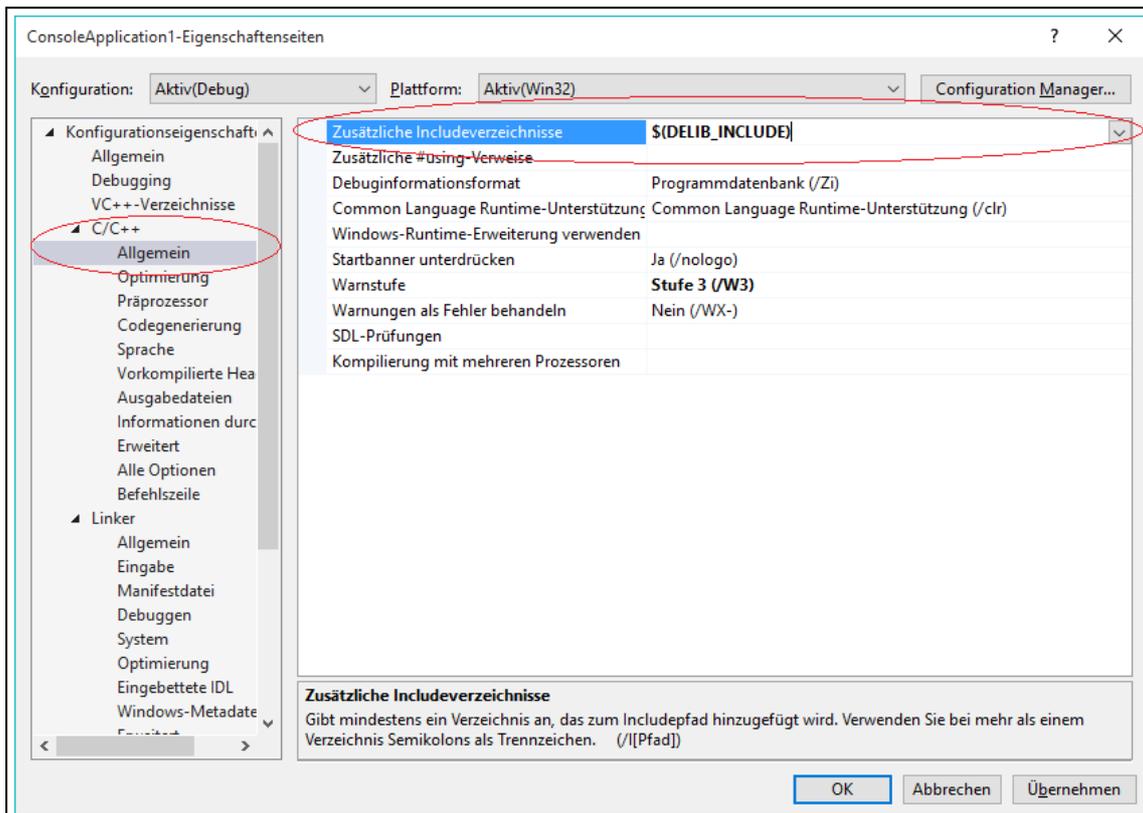
Diese werden im Folgenden in den Projekteinstellungen des Compilers eingetragen.

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



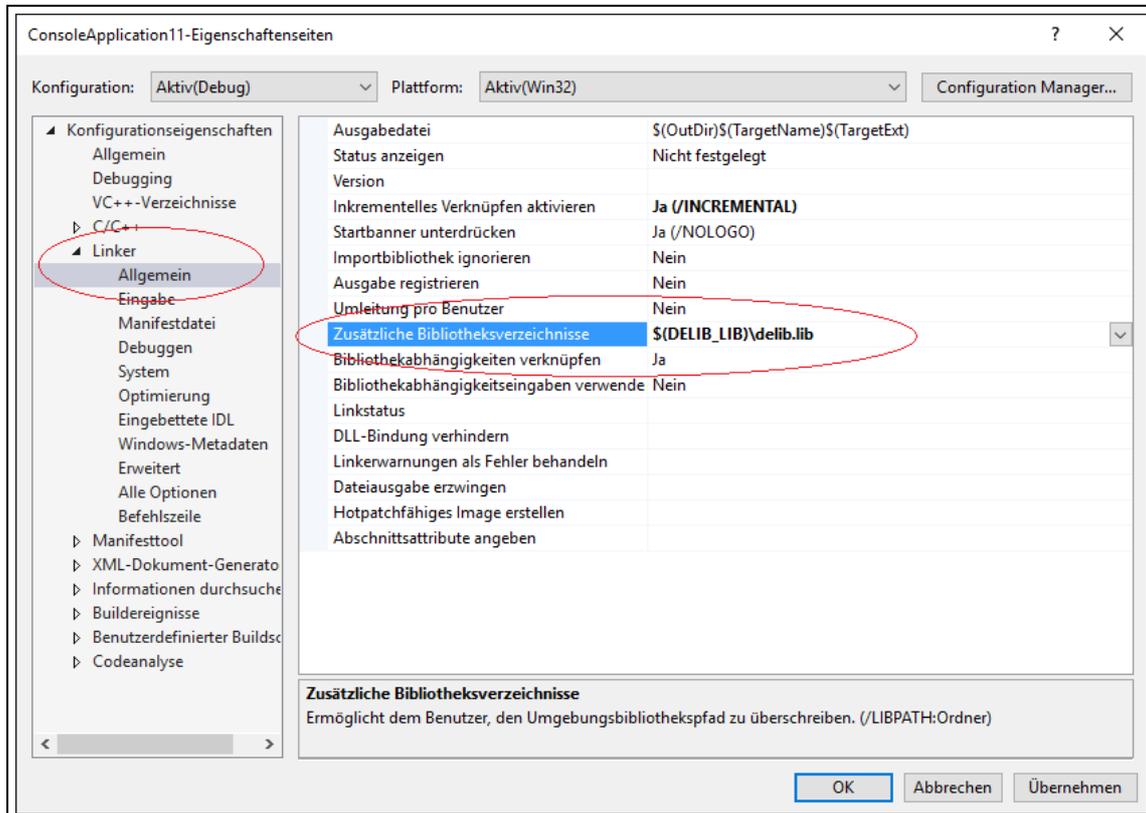
DELIB.H Eintrag in den Visual-C/C++ Projekt Einstellungen

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



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

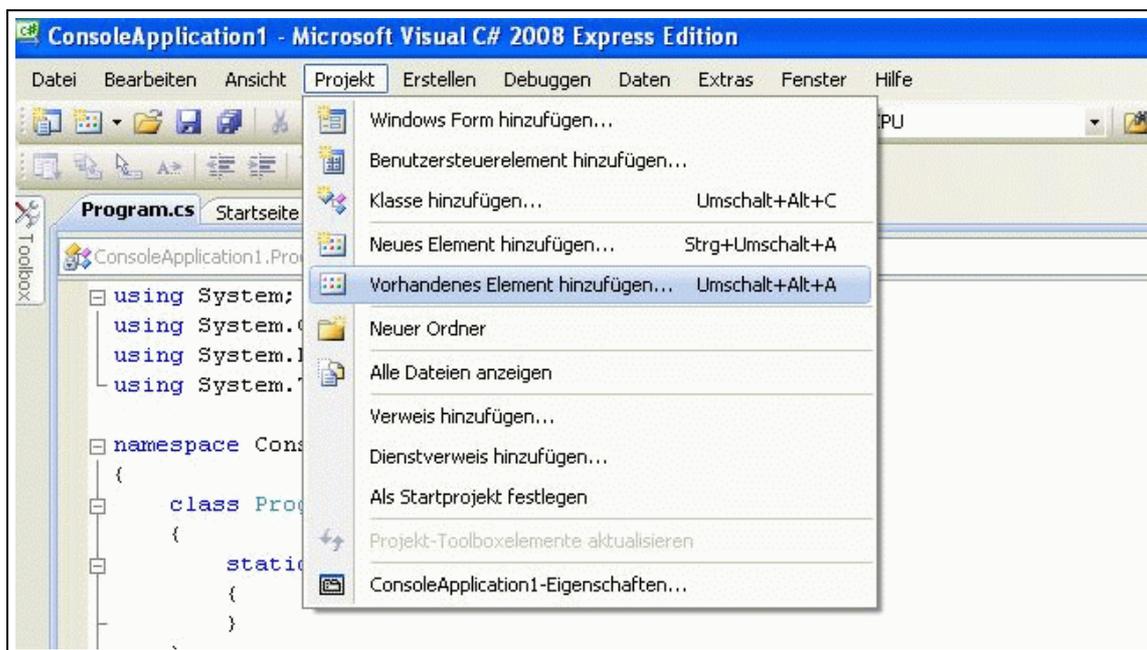
Unter dem Reiter "Linker" bei "Allgemein" "\$ (DELIB_LIB)\delib.lib" eintragen.



4.1.6.3. Einbinden der DELIB in Visual-C#

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

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



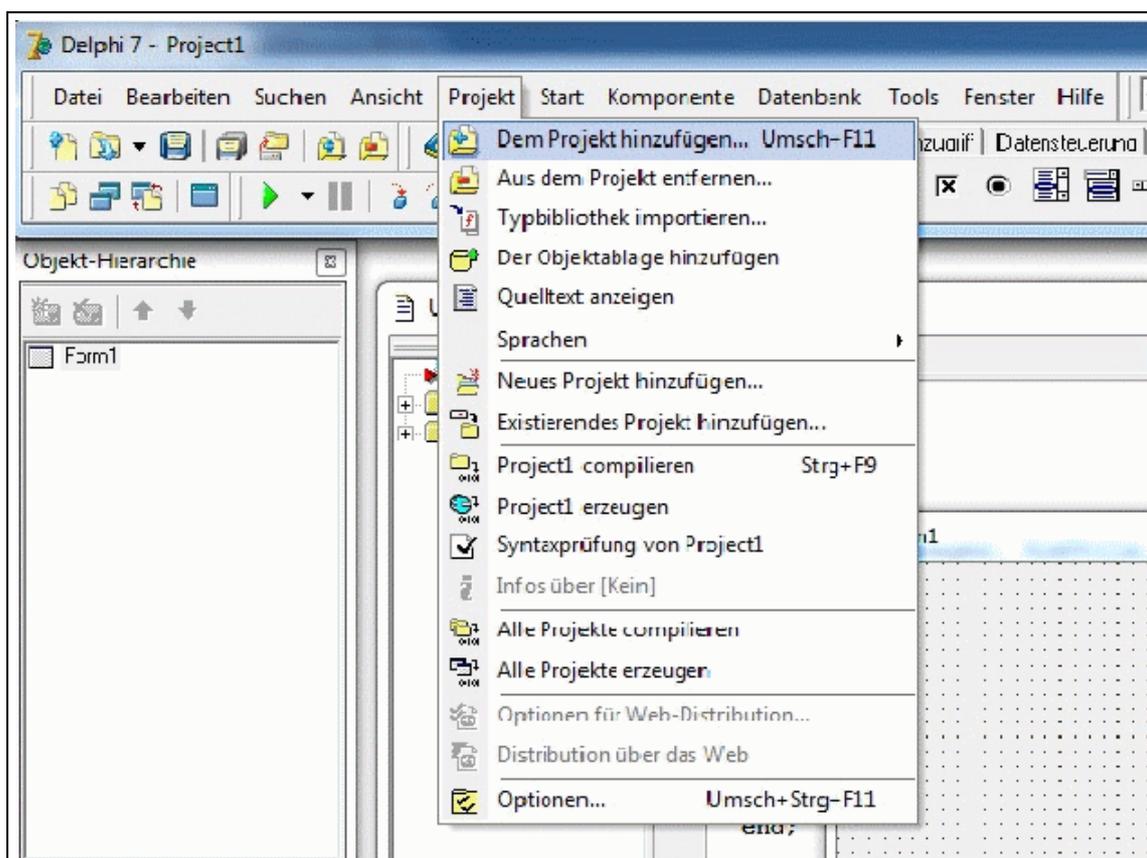
Folgenden Verweis in Ihrem Programm hinzufügen:

using DeLib;

4.1.6.4. Einbinden der DELIB in Delphi

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

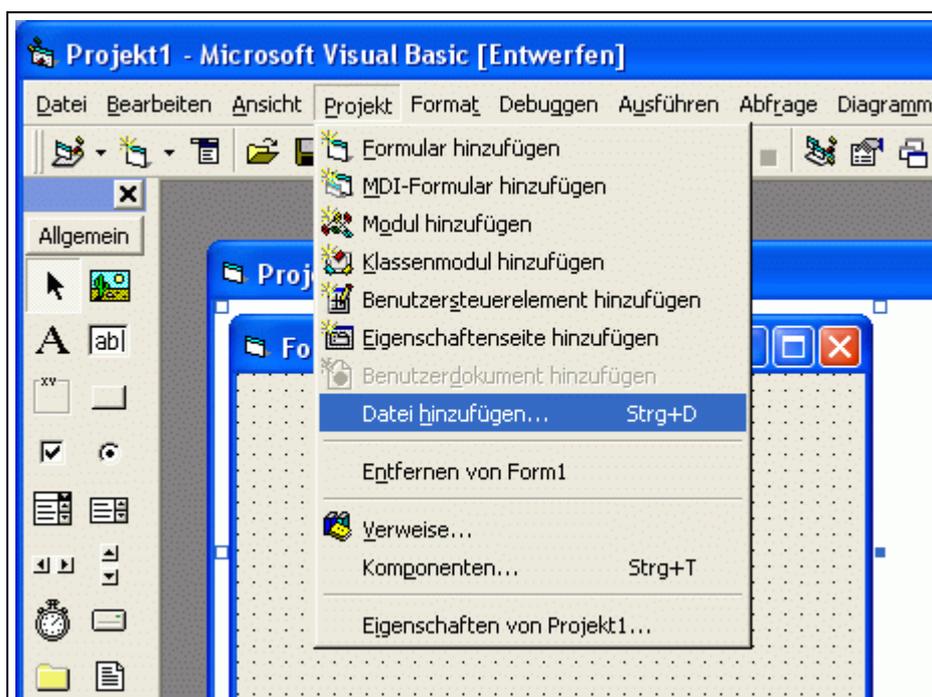
Delphi starten und über das Menue "Projekt → dem Projekt hinzufügen" im
Verzeichnis C:\Programme\DEDITEC\DELIB\include\ die Datei delib.pas zum
Importieren öffnen.



4.1.6.5. Einbinden der DELIB in Visual-Basic (VB)

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

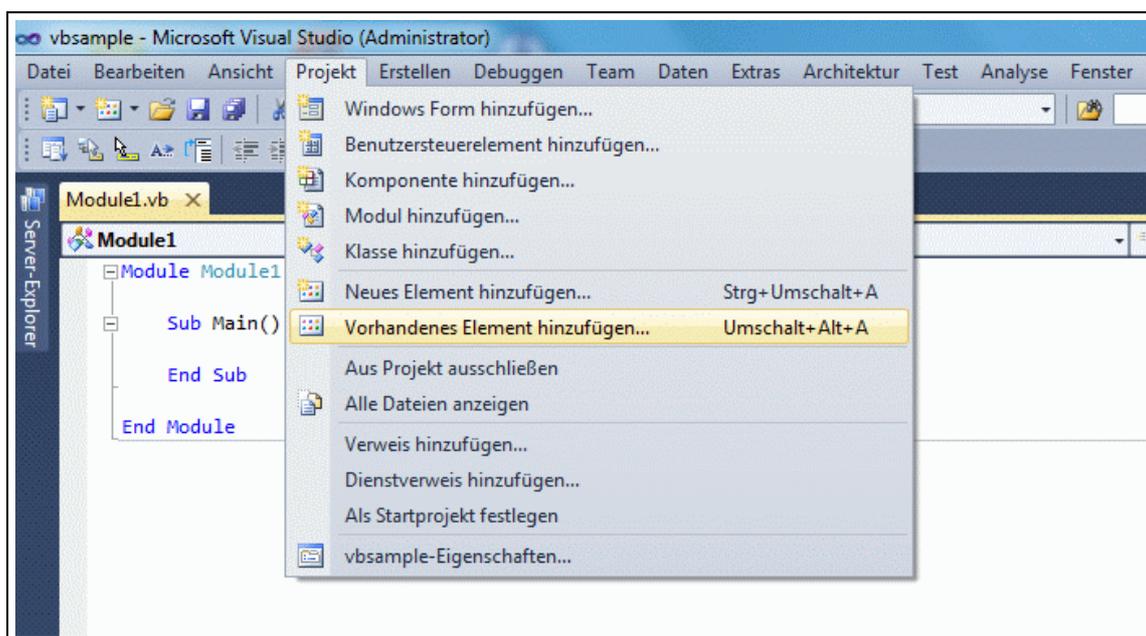
Visual Basic starten und über das Menu "Projekt → Datei hinzufügen..." im
Verzeichnis C:\Programme\DEDITEC\DELIB\include\ die Datei delib.bas zum
Importieren öffnen.



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

Die benötigte Datei für VB.NET befindet sich im Verzeichnis
C:\Programme\DEDITEC\DELIB\include.

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

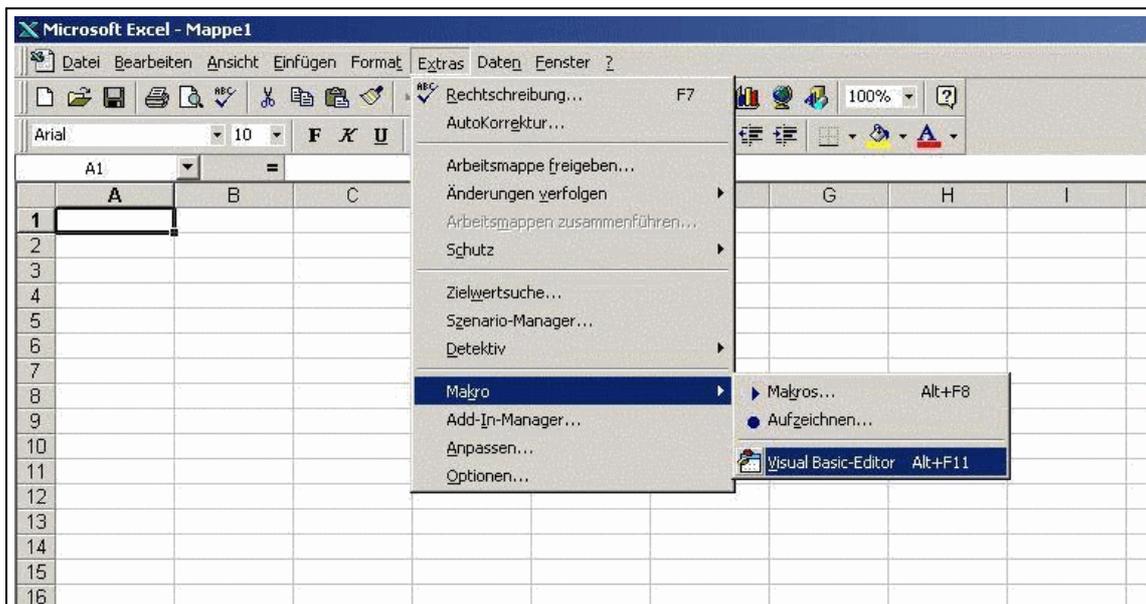


4.1.6.7. Einbinden der DELIB in MS-Office (VBA)

Die benötigte Datei für VBA befindet sich im Verzeichnis

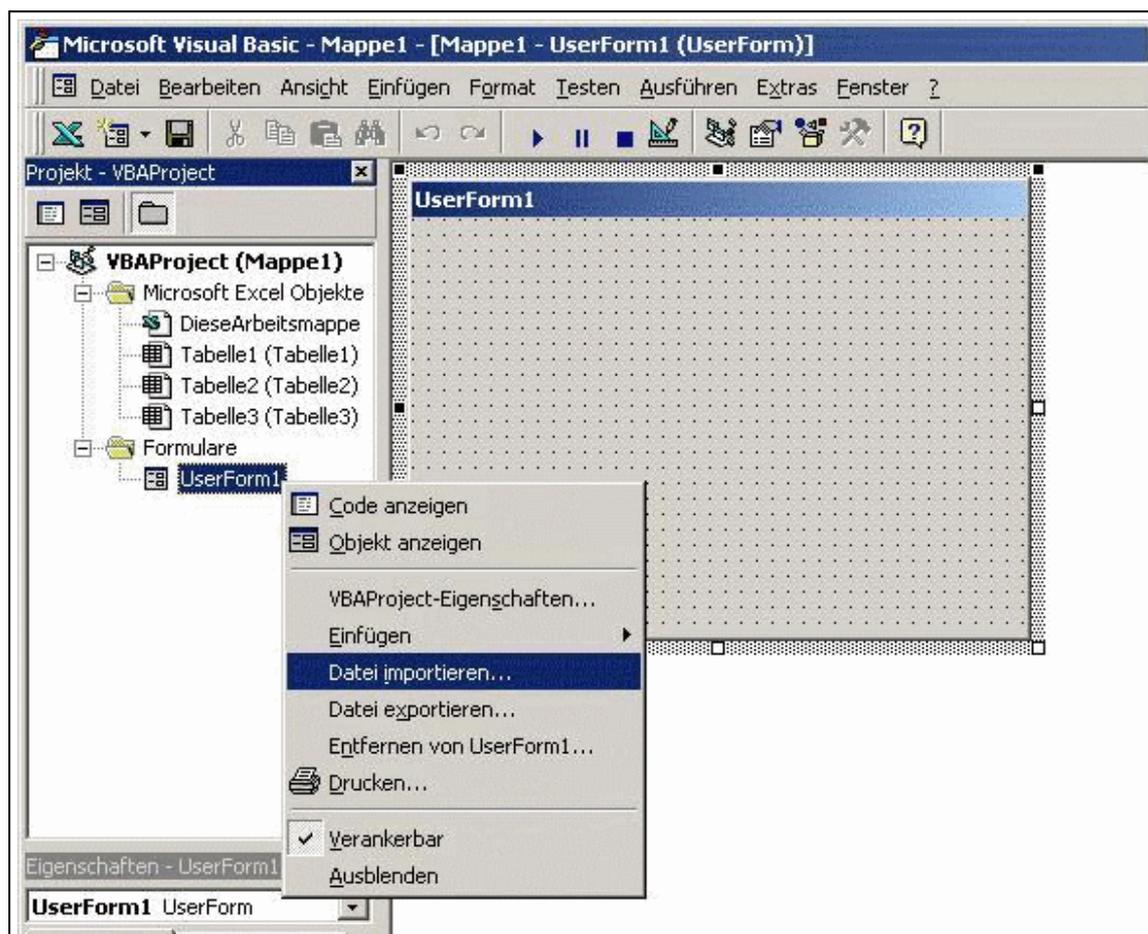
C:\Programme\DEDITEC\DELIB\include.

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



Erstellen der UserForm

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



4.1.6.8. Einbinden der DELIB in LabVIEW

4.1.6.8.1. Einbinden der DELIB in LabVIEW

Das LabVIEW-Beispielprogramm "Deditec_Modul_Control.vi" ist keine EXE-Datei und benötigt deshalb zur Ausführung die LabVIEW Entwicklungsumgebung.

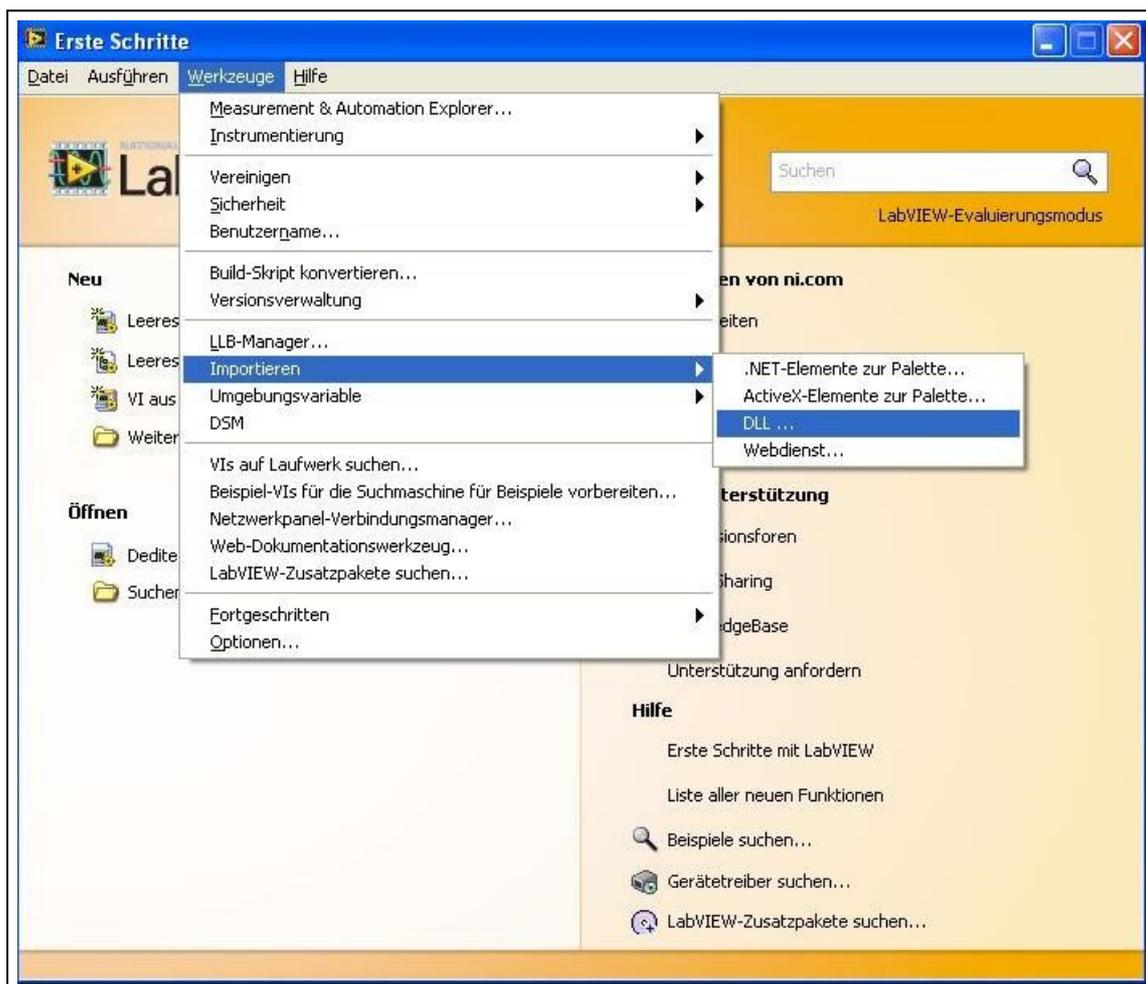
Beschreibung der Einbindung der "delib.dll" in LabVIEW Version 11

- Die benötigten Dateien für LabVIEW befinden sich im Verzeichnis

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

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

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



- Im nächsten Fenster über die 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

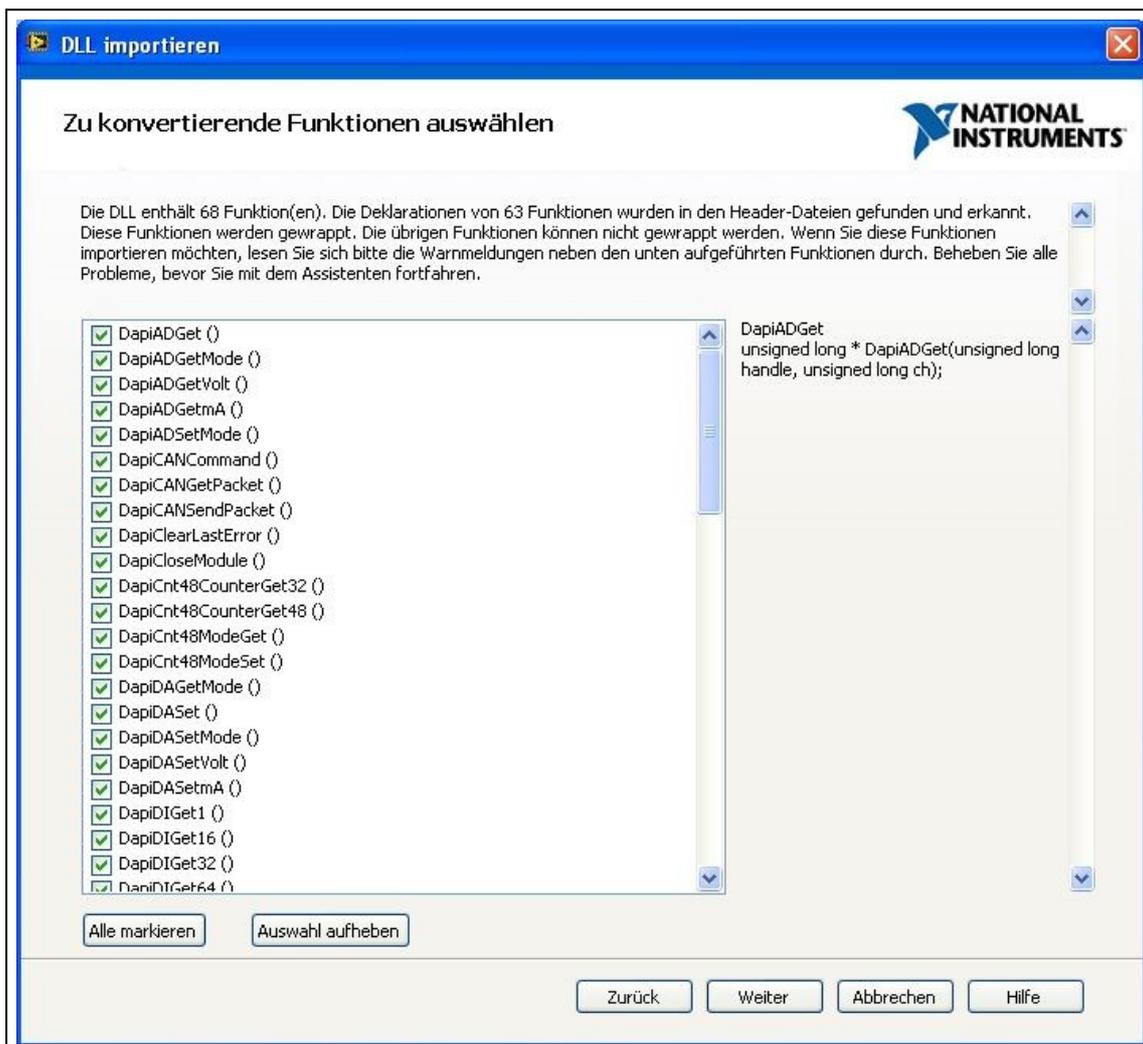
Bibliothek (*.dll)
C:\WINDOWS\system32\delib.dll

DLL befindet sich nicht auf lokalem Computer

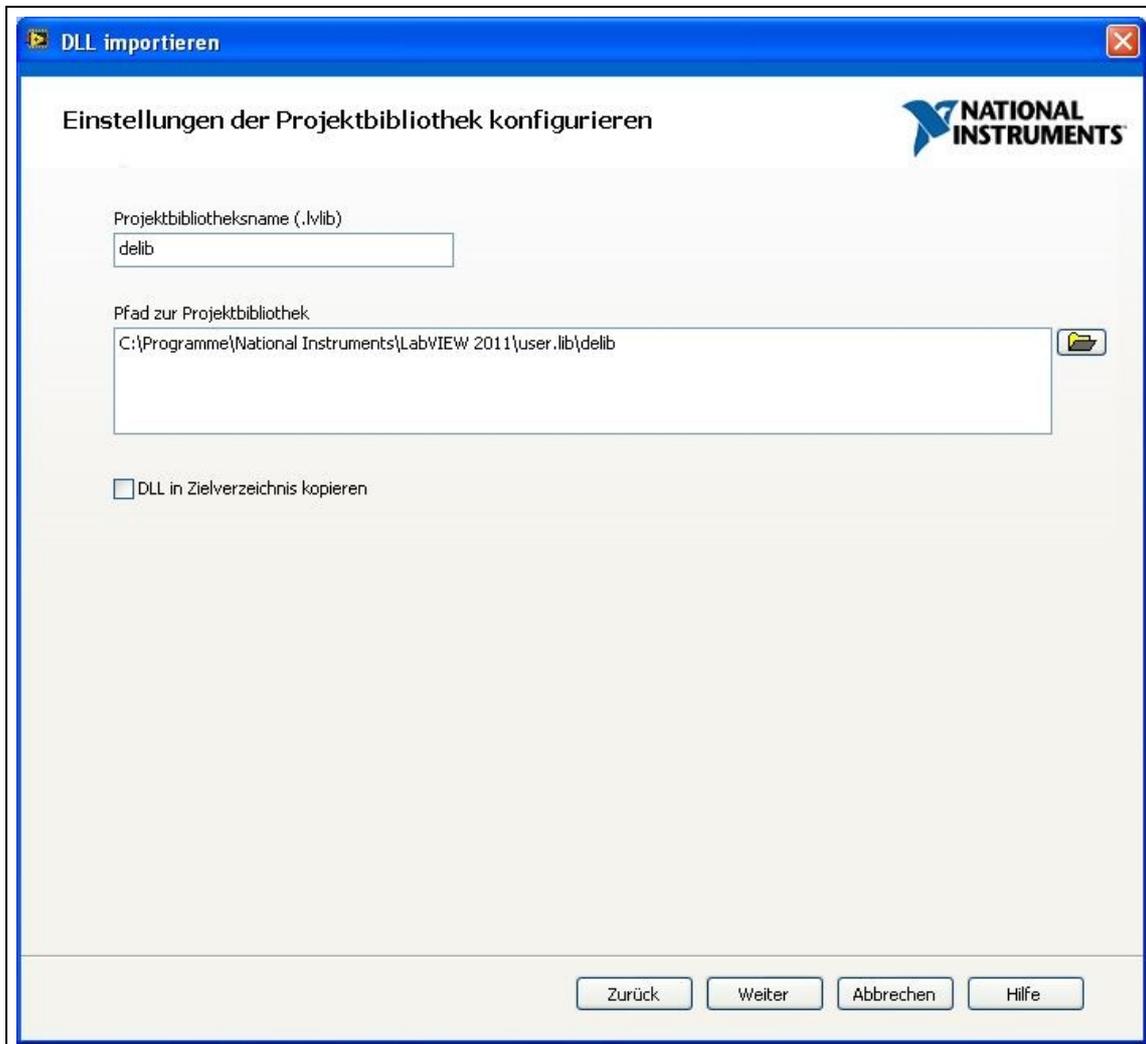
Header-Datei (.h)
C:\Programme\DEDITEC\DELIB\include\delib.h

Zurück Weiter Abbrechen Hilfe

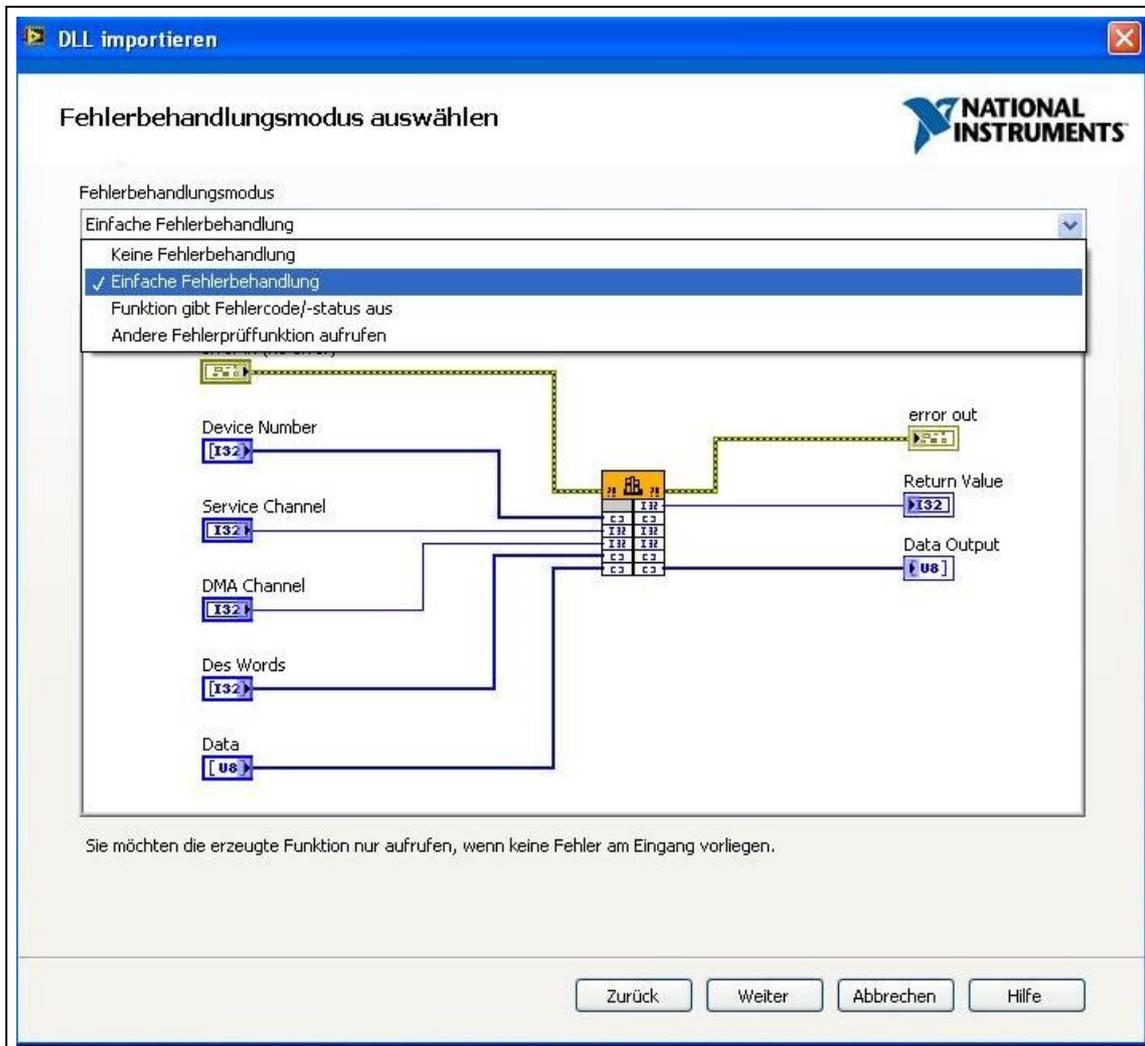
- Nochmals auf "Weiter" klicken um fortzufahren.
- Die Header-Datei wird nun analysiert. Anschließend fahren Sie im folgendem Fenster wieder mit "Weiter" fort.



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



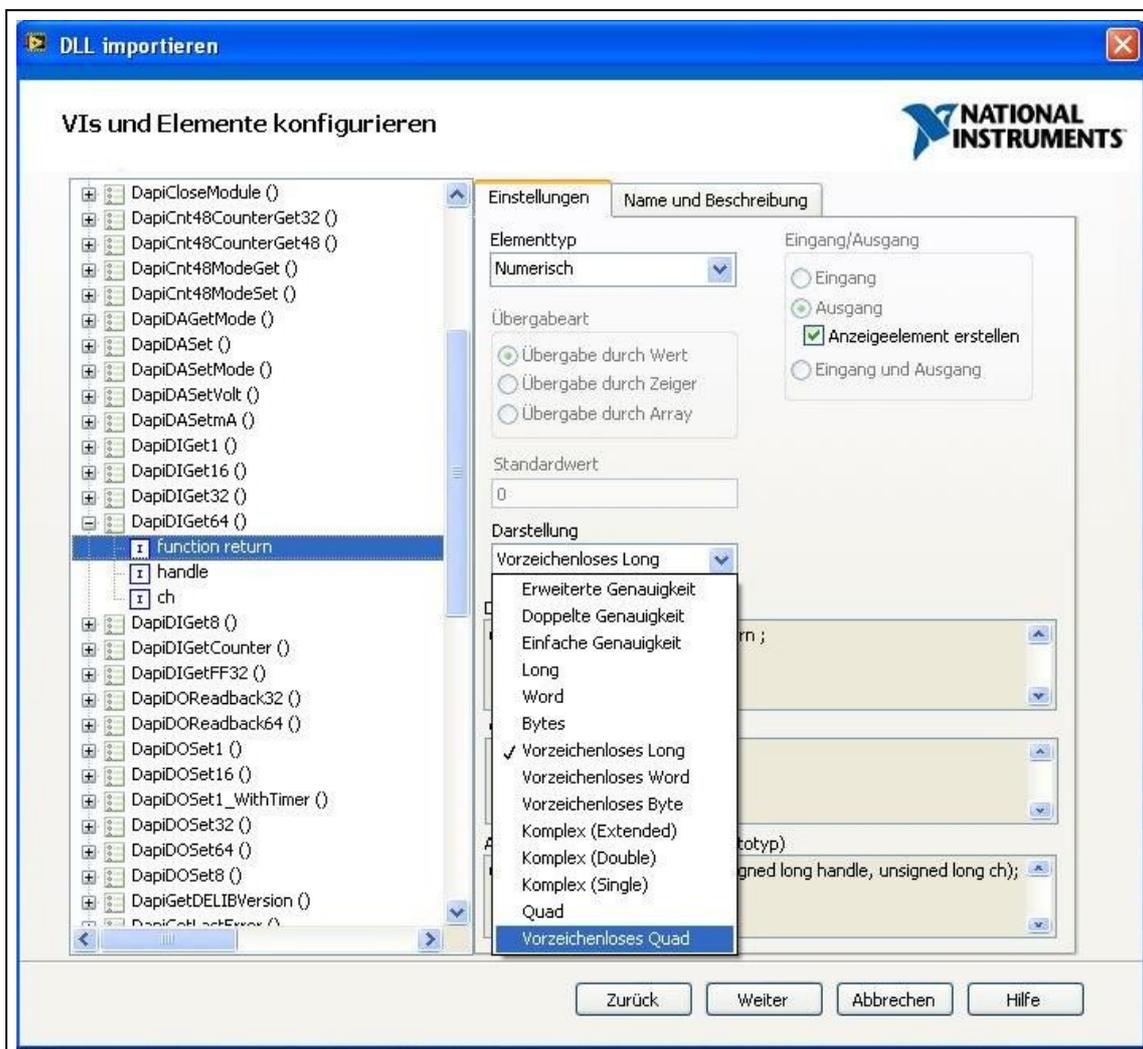
- Im folgendem Fenster wählen Sie im Drop-Down-Menü die Option "Einfache Fehlerbehandlung" aus und fahren mit "Weiter" fort.



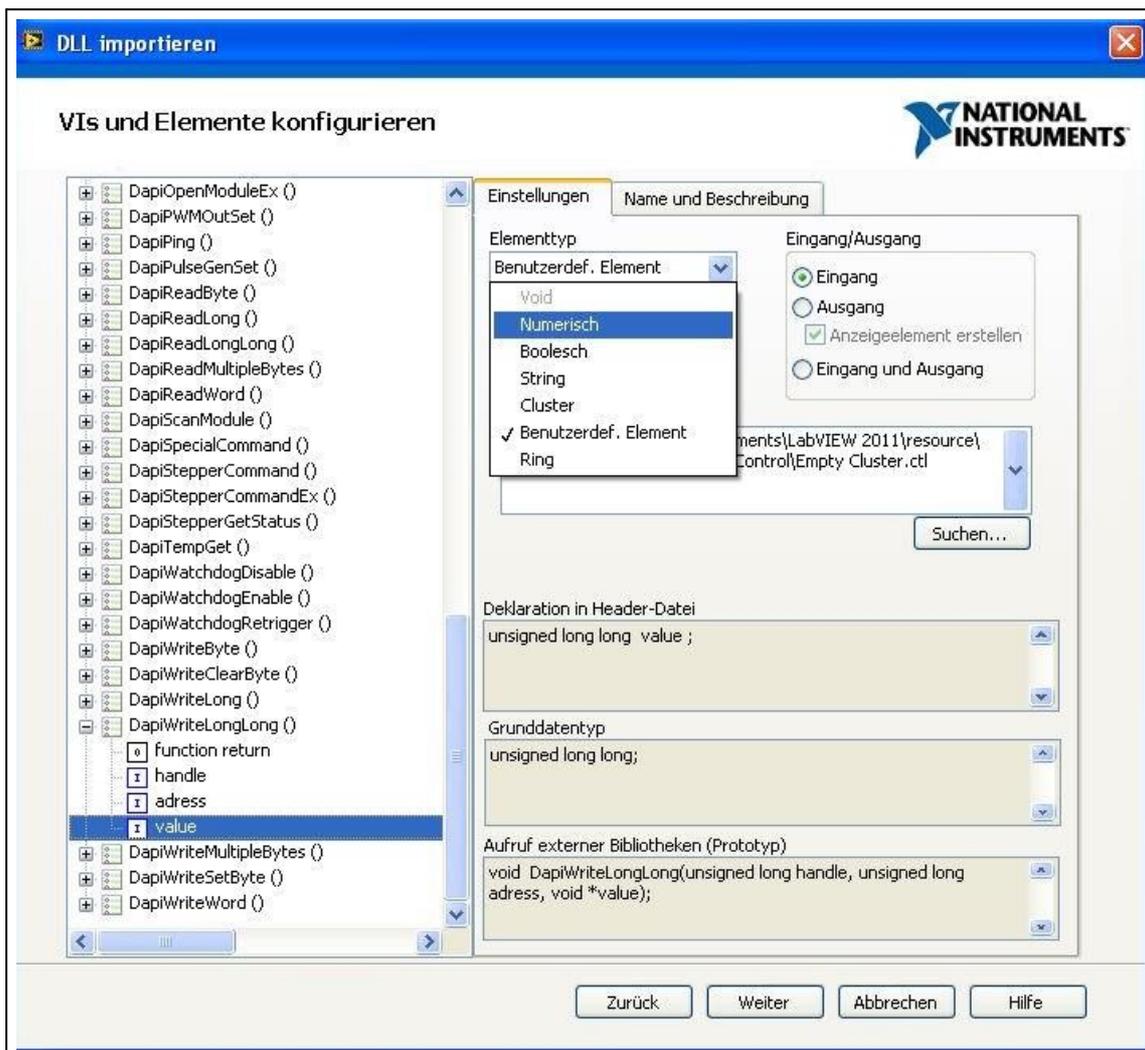
- Bei VIs die mit 64-Bit Werten arbeiten, muss die Darstellung von "Vorzeichenloses Long" in "Vorzeichenloses Quad" geändert werden.

- Folgende VIs müssen bearbeitet werden:

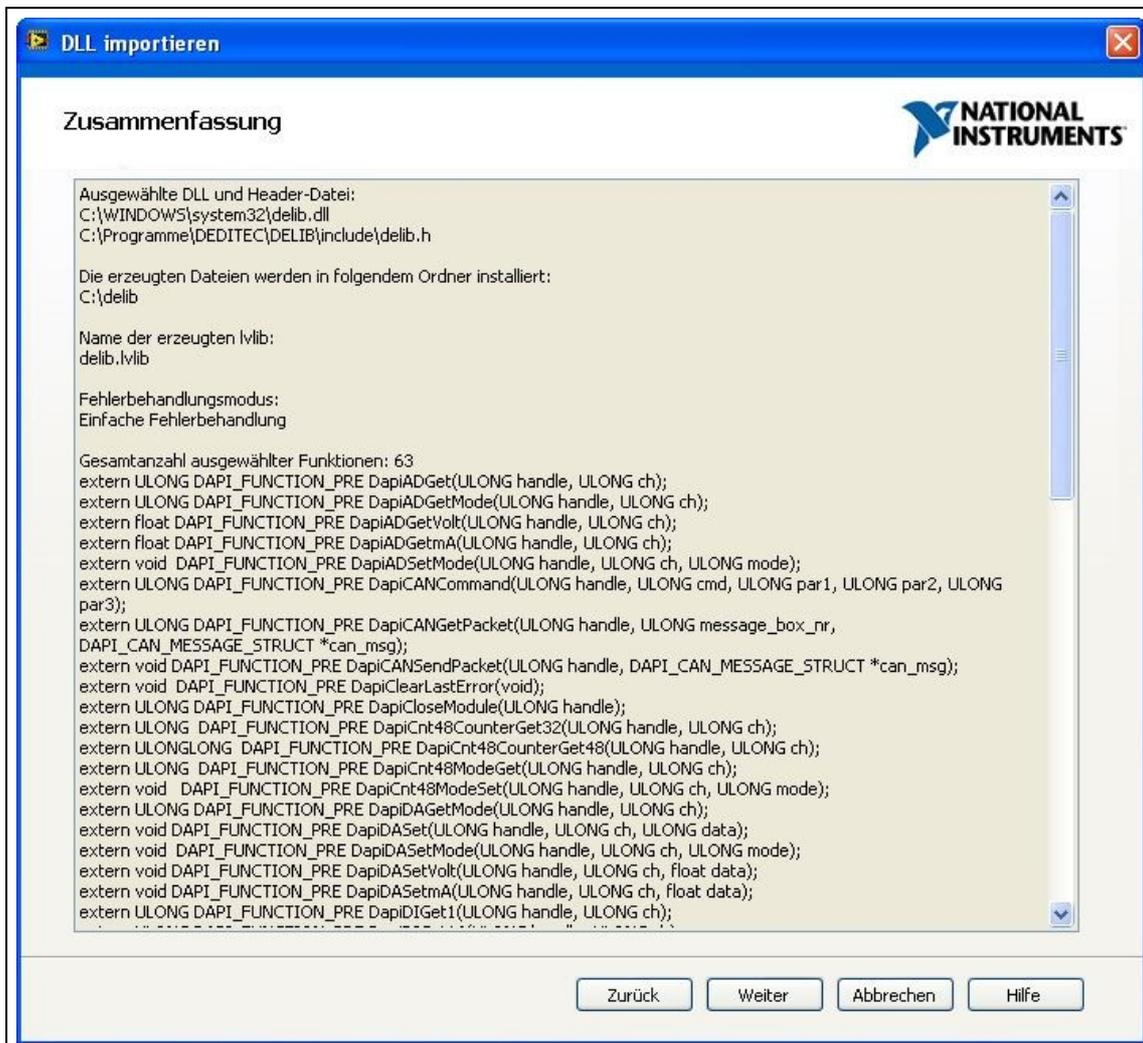
- DapiCNT48CounterGet48 (function return)
- DapiDIGet64 (function return)
- DapiDOSet64 (data)
- DapiDOReadBack64 (function return)



- 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.

4.1.6.8.2. Verwendung der VIs in LabVIEW

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

Diese Defines werden in LabVIEW nicht unterstützt.

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

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

Die Definition für die Funktion lautet:

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

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

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

Beispielcode in C/C++:

DapiADSetMode(handle, 0, ADDA_MODE_UNIPOL_5V);

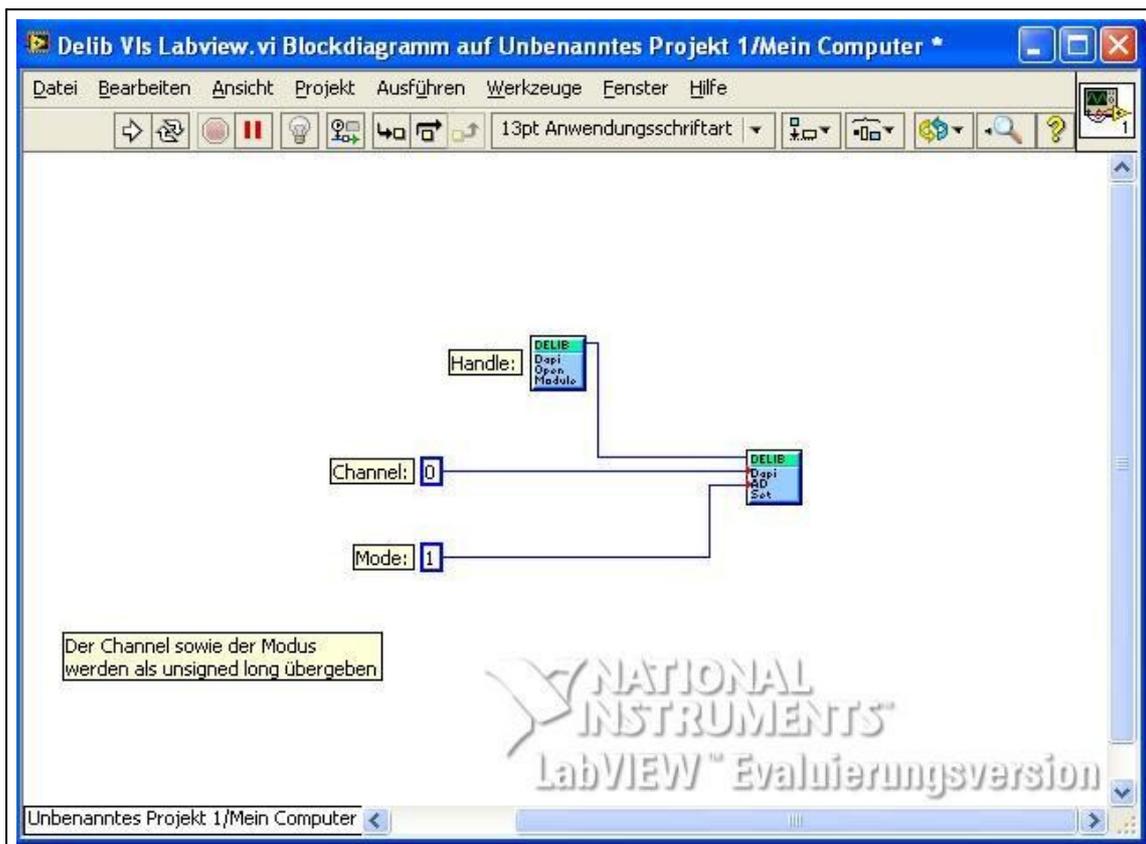
Alternativ kann man auch folgende Schreibweise verwenden:

DapiADSetMode(handle, 0, 0x01);

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

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

In LabVIEW könnte die Funktion dann so aussehen:



4.1.6.8.3. Setzen der Modul-ID in LabVIEW

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

Die Verbindung zum Modul wird mittels der Funktion DapiOpenModule hergestellt.

Die Definition für diese Funktion lautet:

ULONG DapiOpenModule(ULONG moduleID, ULONG nr);

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

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

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

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

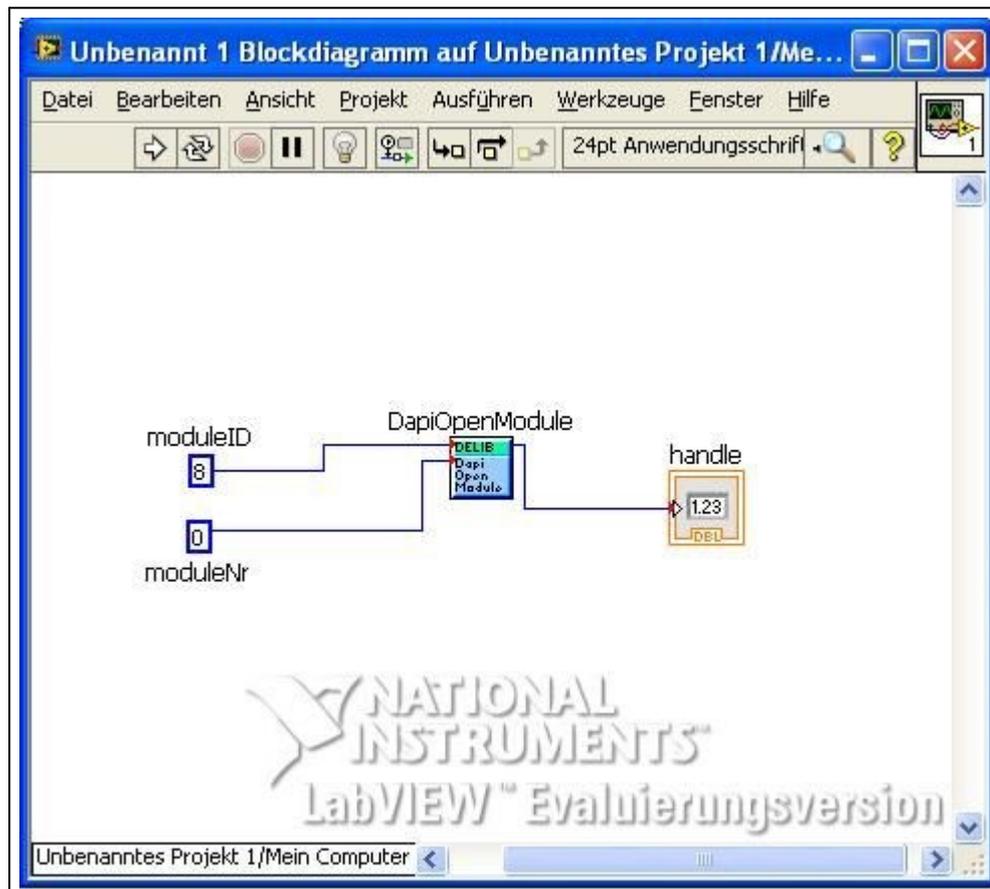
```
handle = DapiOpenModule(RO_ETH, 0); // öffnet ein RO-ETH-Modul mit  
Modul-Nr 0.
```

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:



4.1.6.9. Einbinden der DELIB in Java

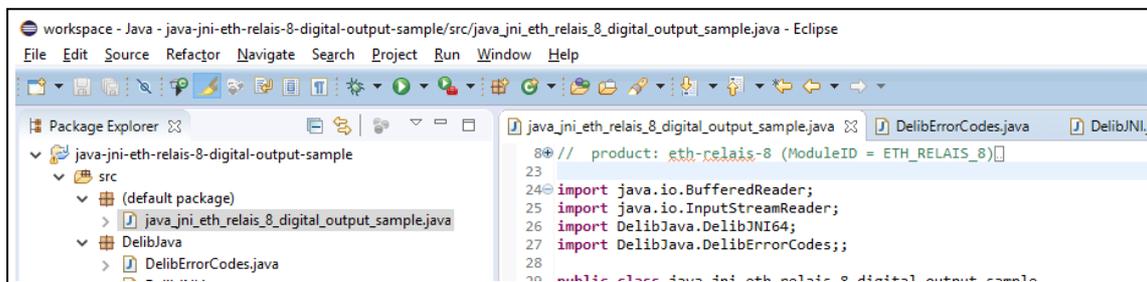
Die benötigten Dateien für Java befinden sich, je nach DELIB-Installation, in folgendem Verzeichnis

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

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

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

Anschließend müssen die verwendeten Module noch importiert werden.



The screenshot shows the Eclipse IDE interface. The Package Explorer on the left displays a project named 'java-jni-eth-relais-8-digital-output-sample' with a 'src' folder containing 'java_jni_eth_relais_8_digital_output_sample.java', 'DelibJava', and 'DelibErrorCodes.java'. The main editor window shows the source code of 'java_jni_eth_relais_8_digital_output_sample.java' with the following imports:

```
24 import java.io.BufferedReader;
25 import java.io.InputStreamReader;
26 import DelibJava.DelibJNI64;
27 import DelibJava.DelibErrorCodes;;
28
```

4.2. DELIB Treiberbibliothek

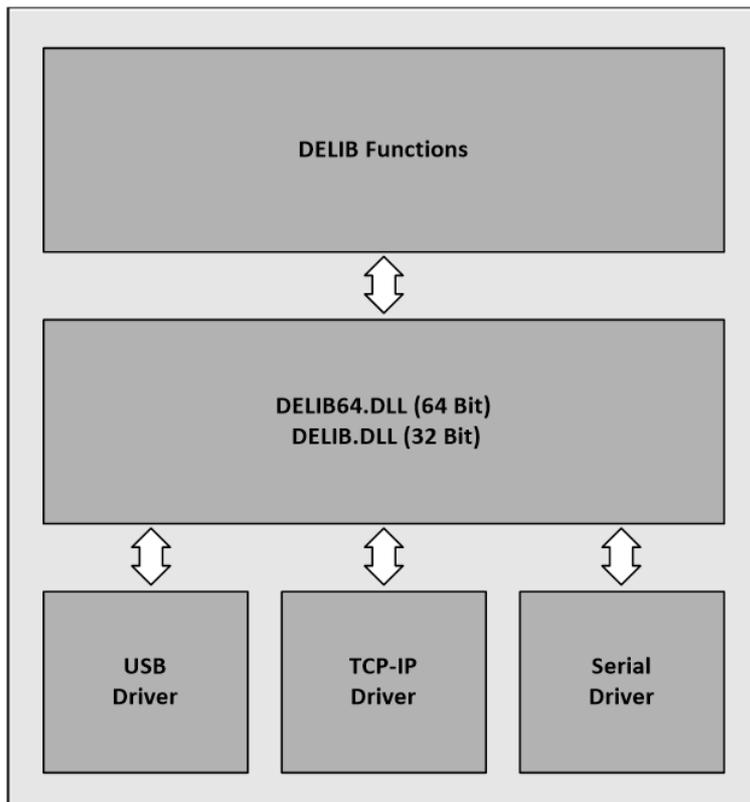
Die DELIB Treiberbibliothek enthält die DELIB-API und verschiedene Programme für den Konfigurationstest unserer Produkte.

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

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

4.2.1. Übersicht

Die folgende Abbildung erläutert den Aufbau der DELIB Treiberbibliothek



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

- Betriebssystem unabhängig
- Programmiersprachen unabhängig
- Produkt unabhängig

Diese Versionen der Treiberbibliothek bieten wir an:

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

DELIB Treiberbibliothek ETH

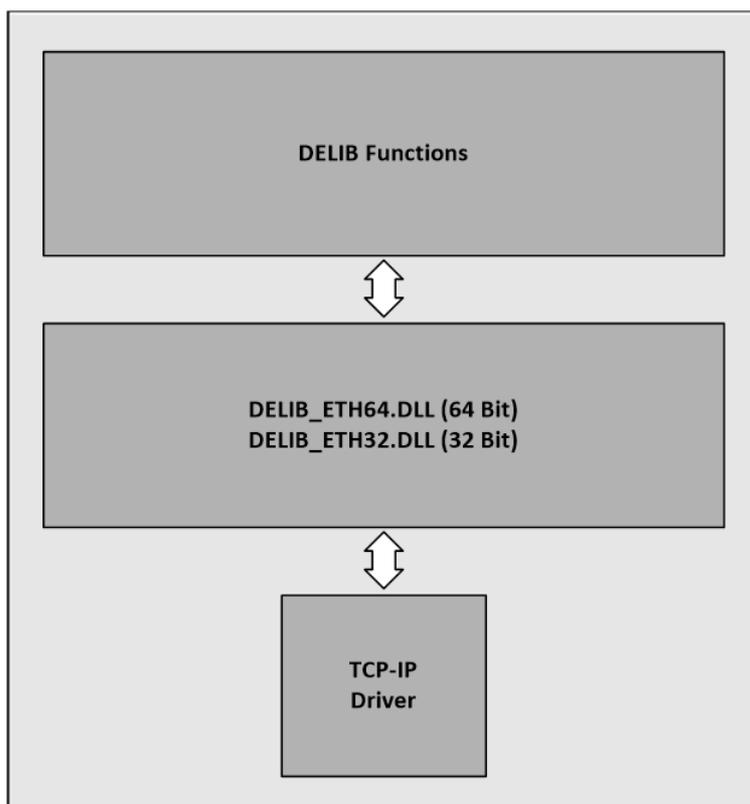
Während die DELIB für ALLE Produkte zur Verfügung steht, wird bei der DELIB-ETH auf keine weiteren Treiber zugegriffen (wie z. B. USB).

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.



4.2.1.1. Unterstützte Programmiersprachen

Die folgenden Programmiersprachen werden von der DELIB Treiberbibliothek unterstützt:

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

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

4.2.1.2. Unterstützte Betriebssysteme

Die folgende Betriebssysteme sind mit unserer DELIB Treiberbibliothek kompatibel:

32-Bit:

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

64-Bit:

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

4.2.1.3. SDK-Kit für Programmierer

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

4.2.2. DELIB Setup

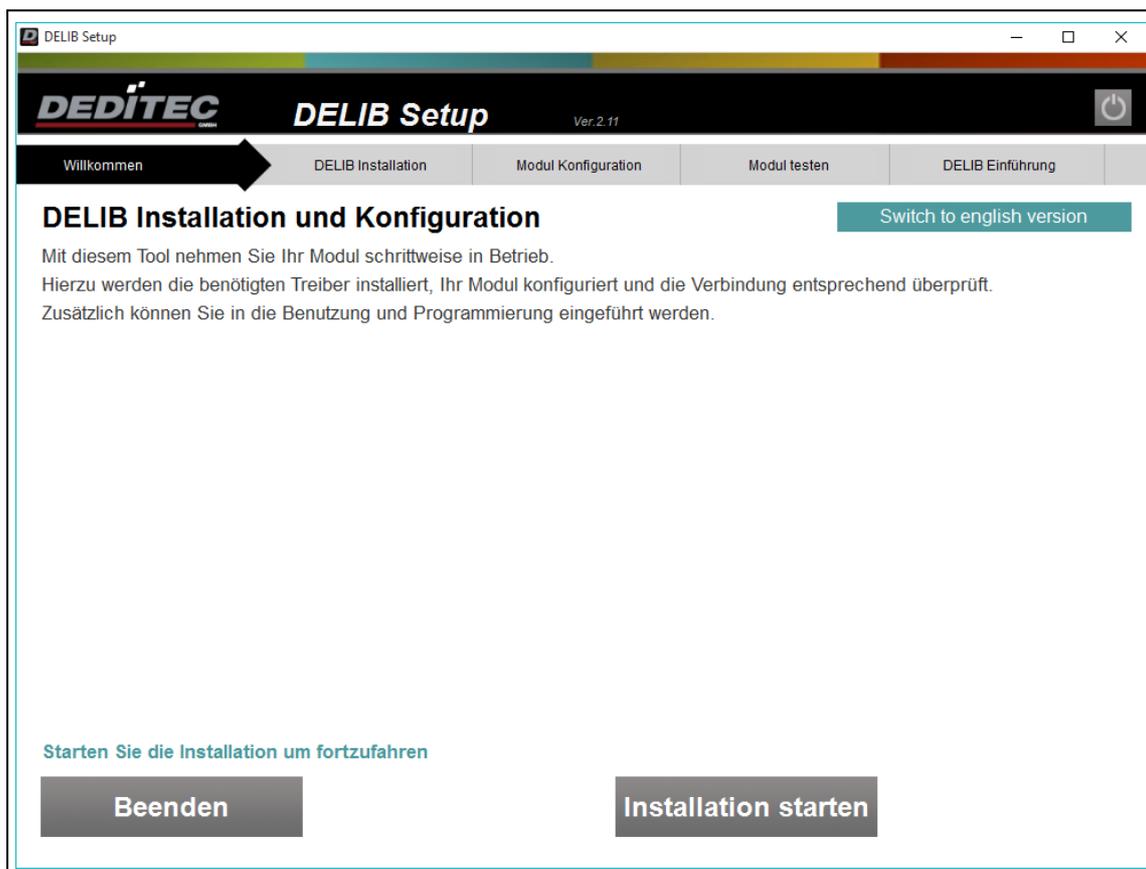
Das DELIB Setup führt Sie durch die Installation unserer DELIB Treiberbibliothek.

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

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

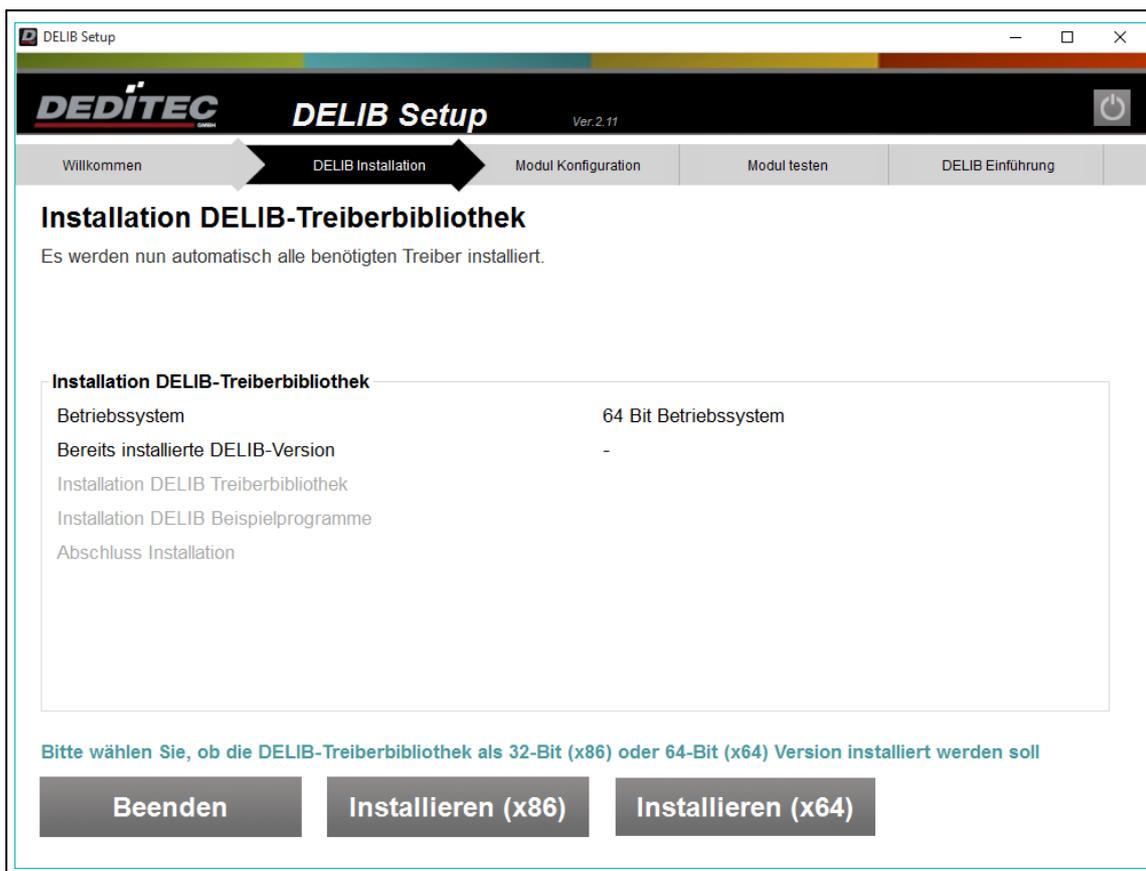
Link: <https://www.deditec.de/delib>

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

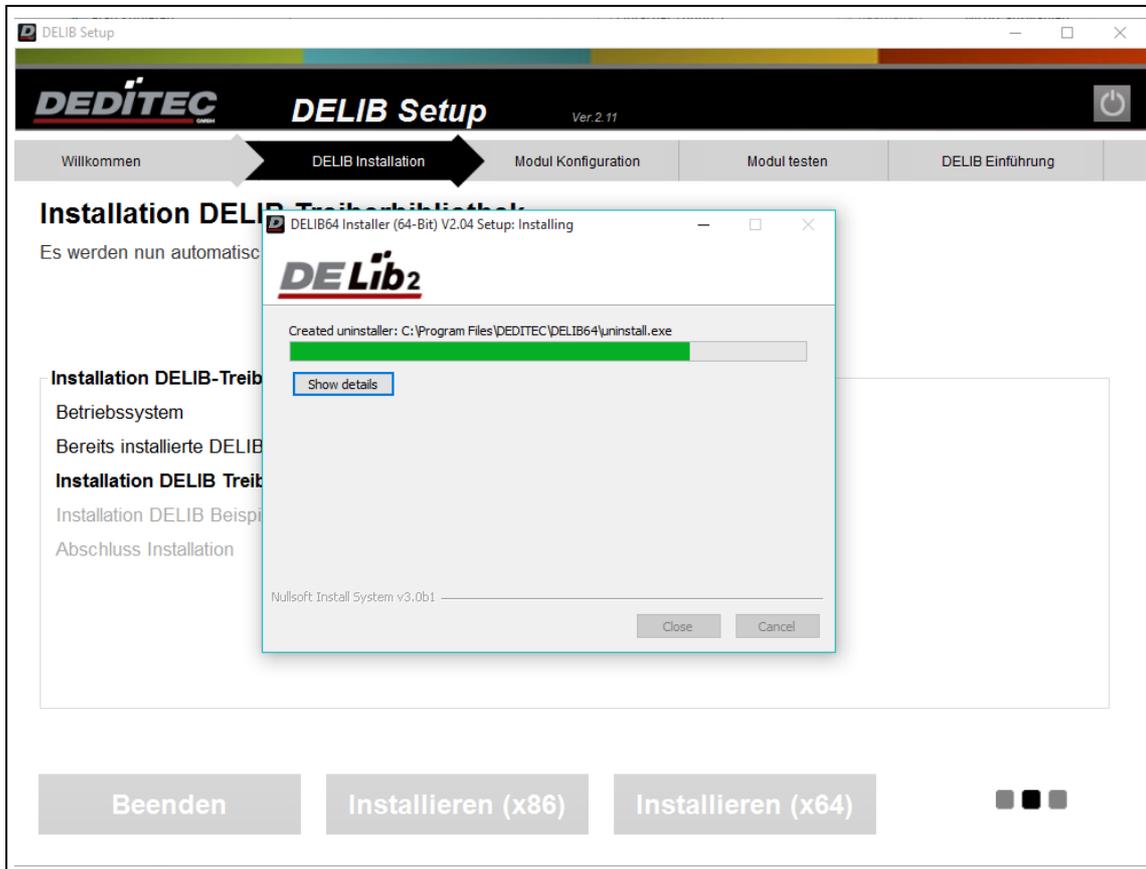


Das DELIB Setup prüft das Betriebssystem und ob bereits Versionen der DELIB Treiberbibliothek installiert sind.

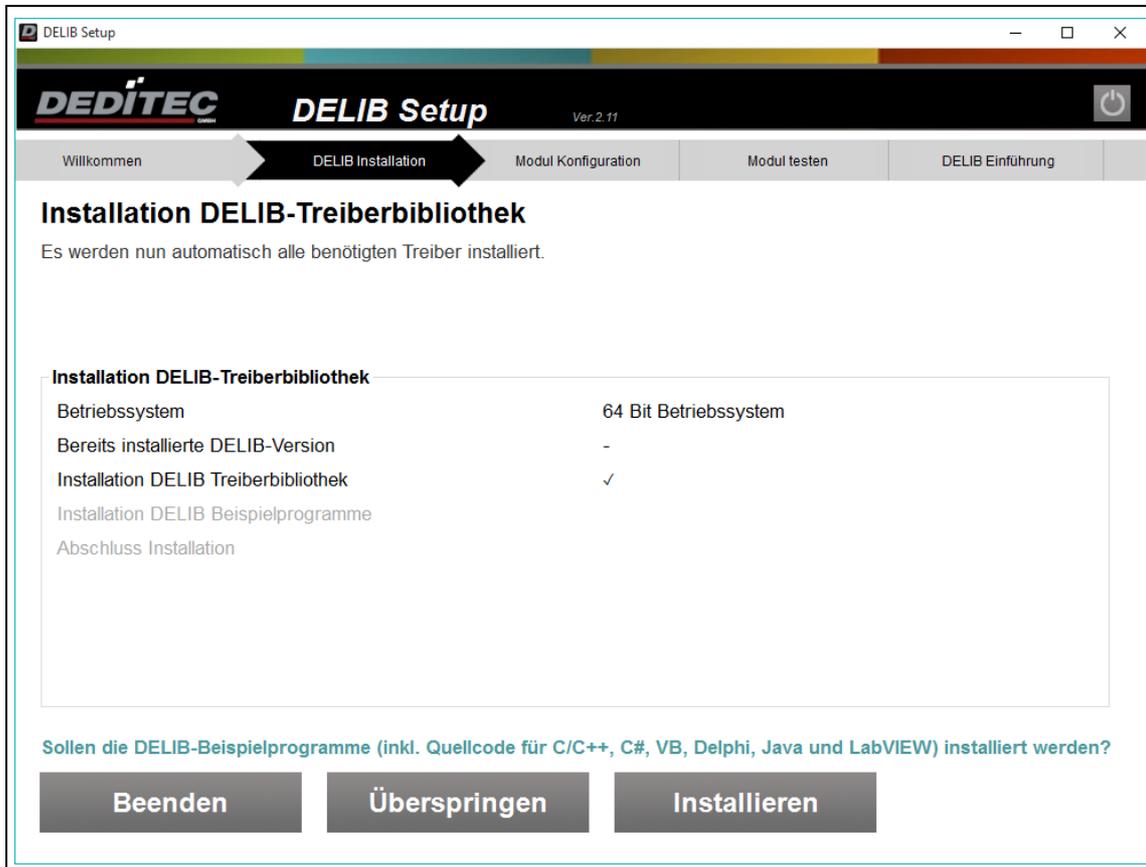
Anschließend können Sie wählen ob Sie die 32-Bit oder 64-Bit Version der DELIB Treiberbibliothek installieren möchten.



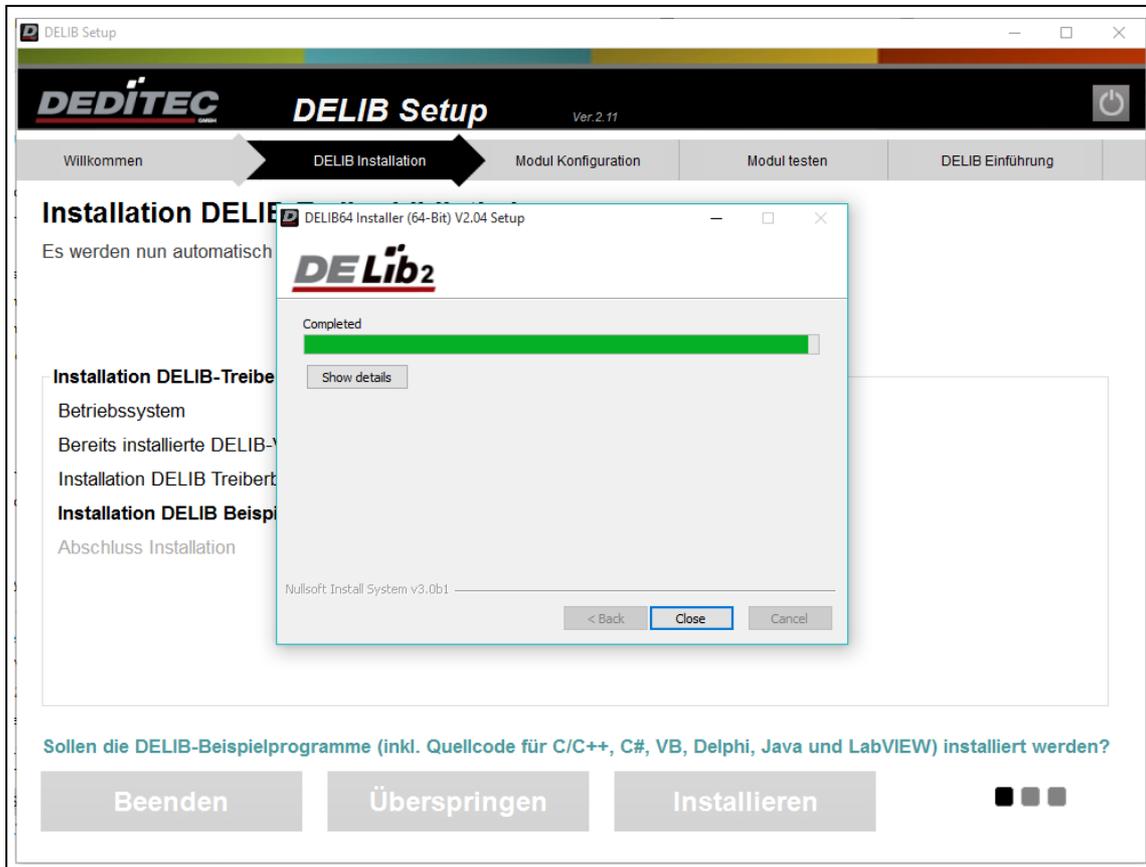
Installationsfortschritt der Treiberbibliothek.



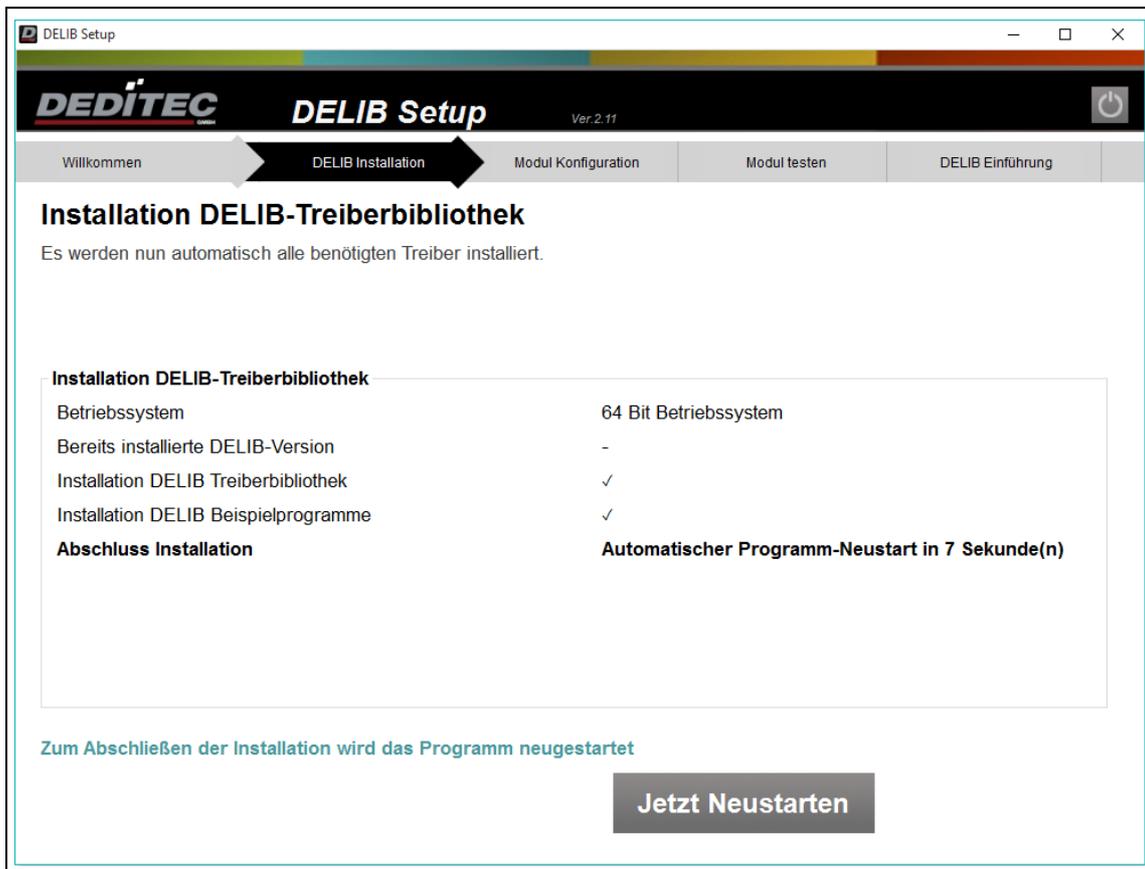
Zusätzlich können Sie wählen, ob Sie die DELIB Beispielprogramme installieren.



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.



4.2.3. DELIB Configuration Utility

4.2.3.1. Einführung

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

Die Konfiguration ist für die erste Inbetriebnahme erforderlich.

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

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

Standardpfad:

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

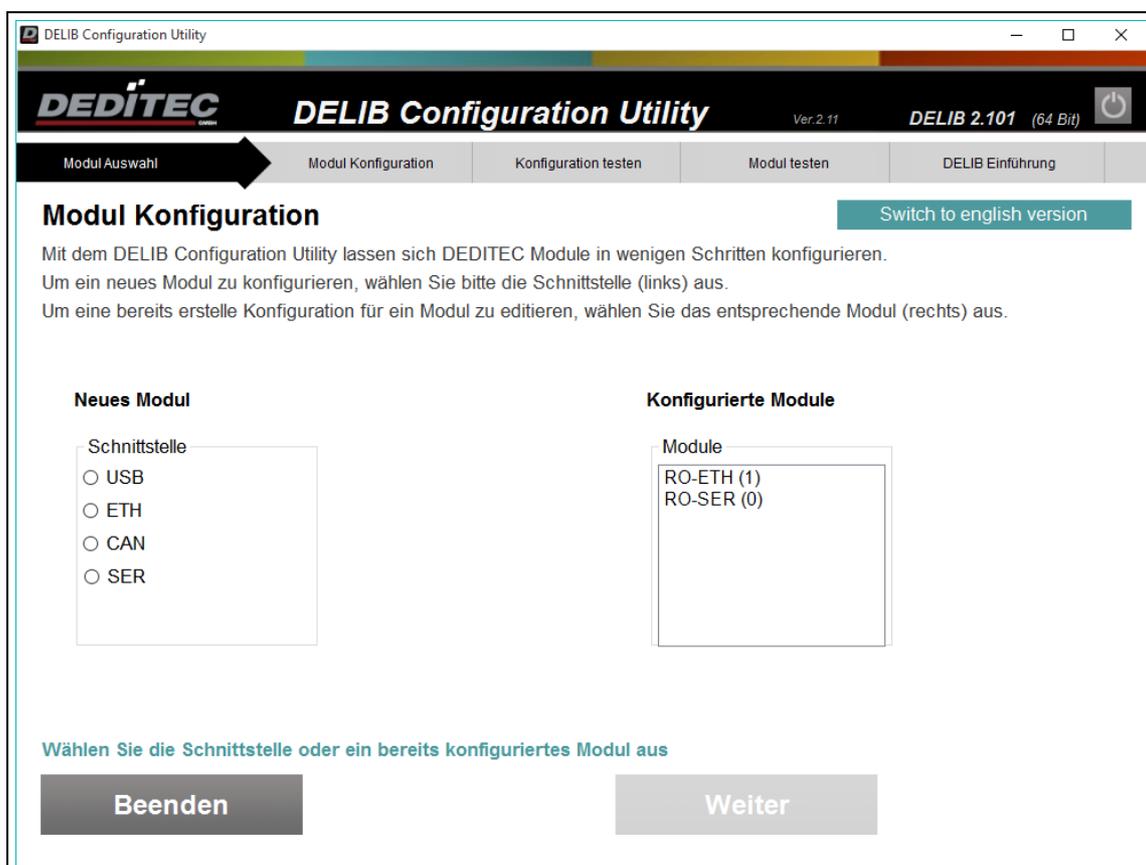
64-Bit: C:\Program Files\DEDITEC\DELIB64\programs\delib-configuration-utility_x64.exe

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

4.2.3.2. Neue Konfiguration erstellen oder vorhandene Konfiguration bearbeiten

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

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

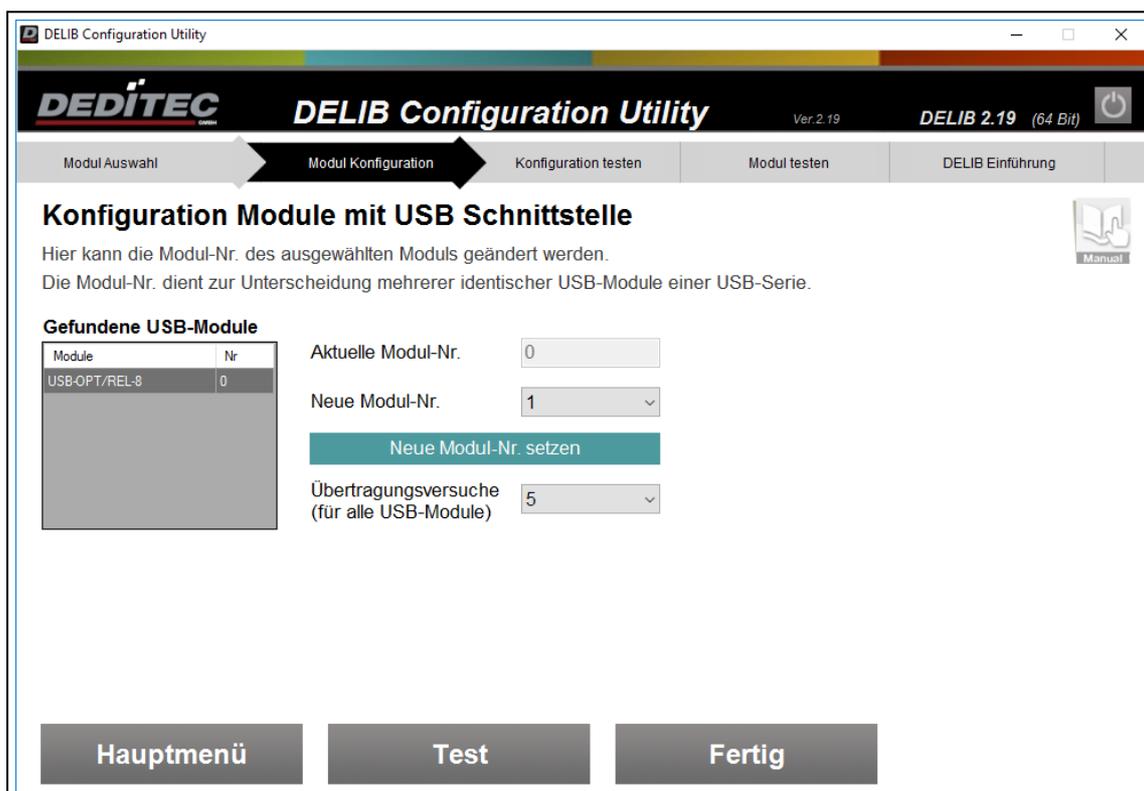


4.2.3.2.1. Modul Konfiguration USB

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

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

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



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.

4.2.3.2.1.1. Beispiel zur Konfiguration identischer USB-Module

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

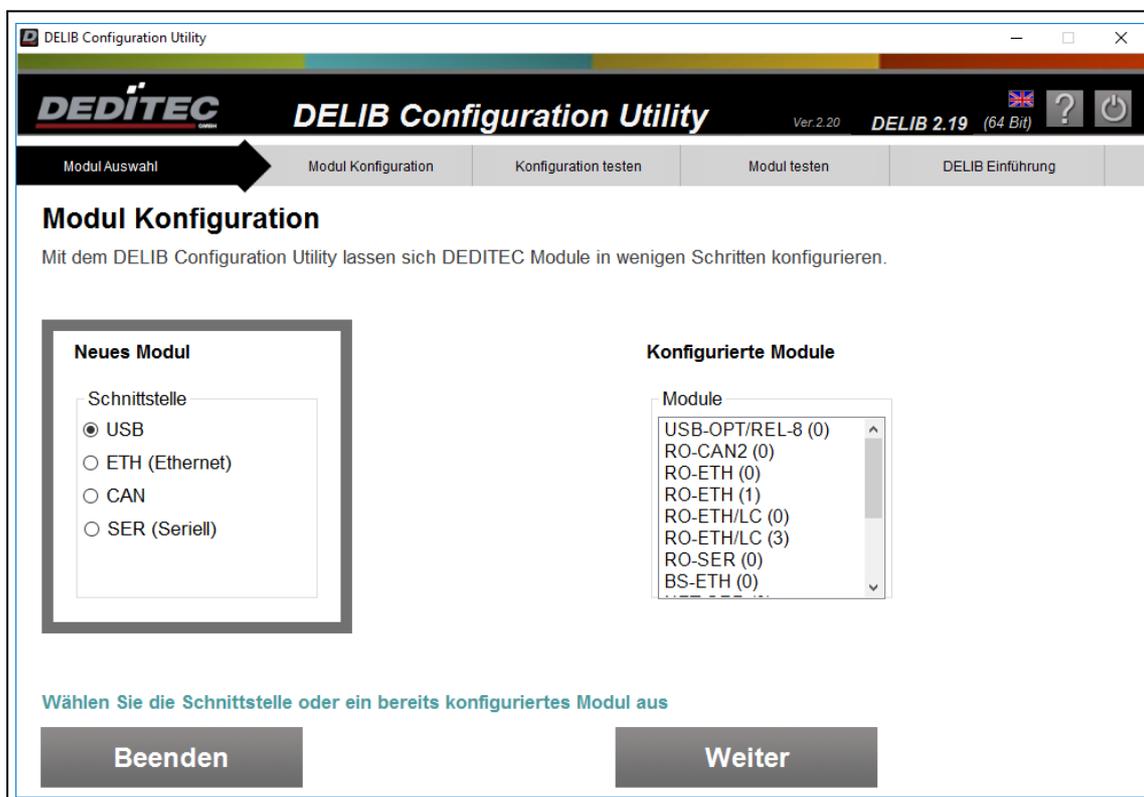
Befindet sich nur ein USB-Modul, oder mehrere USB-Module mit unterschiedlicher Modul-ID (z.B. RO-USB-016 und USB-RELAIS-8) im System, ist keine Konfiguration nötig, da die Produkte über die ID eindeutig identifizierbar sind.

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".

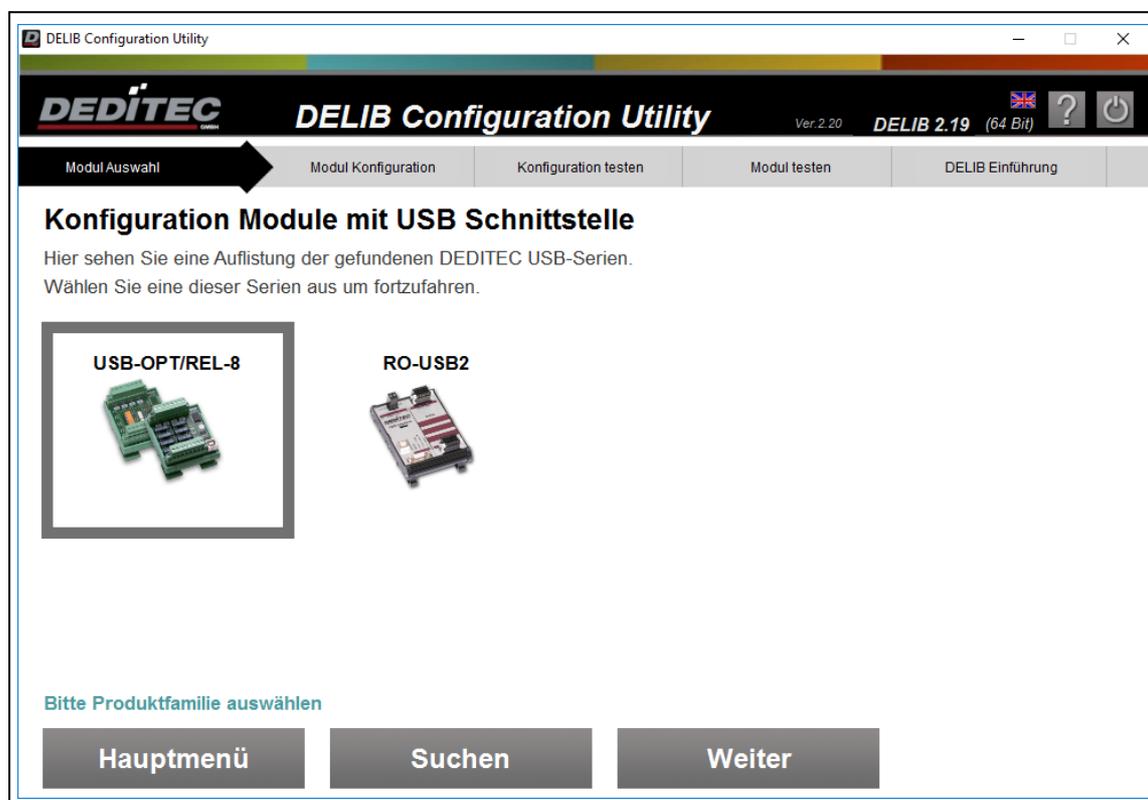


Schritt 2

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

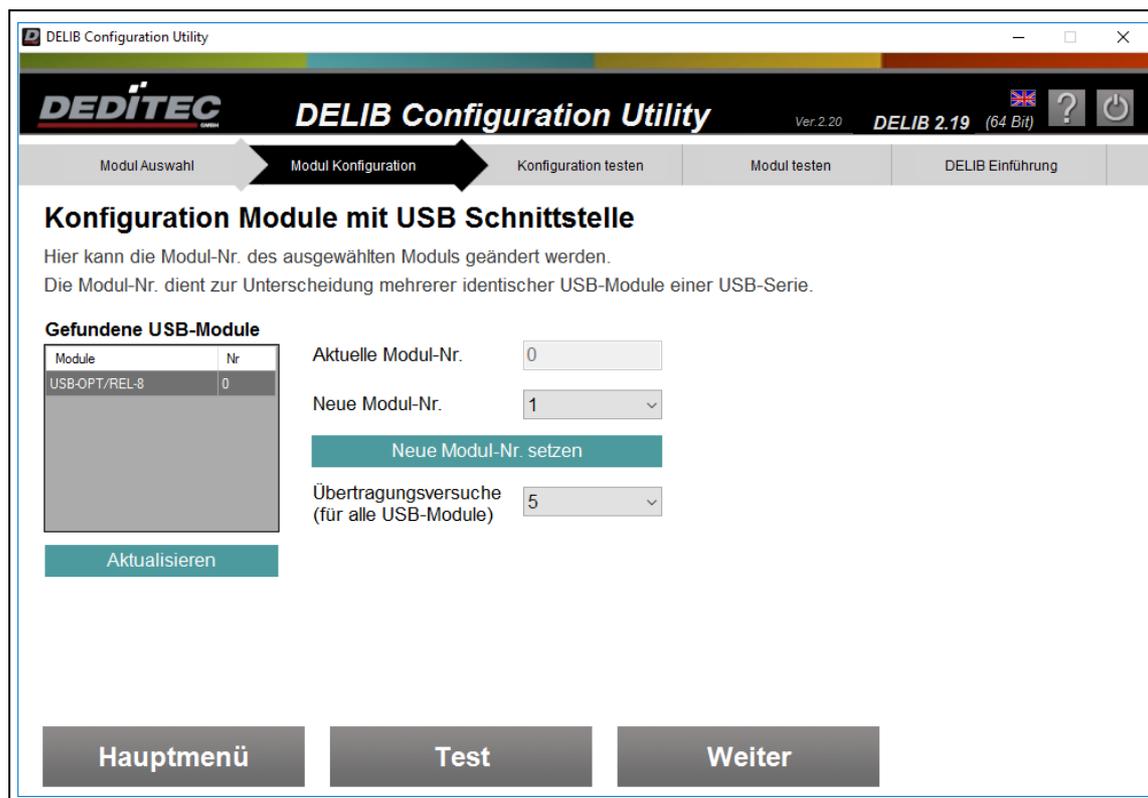
In diesem Beispiel sind Module der RO-USB2-Serie und USB-OPT/REL-8-Serie angeschlossen.

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.

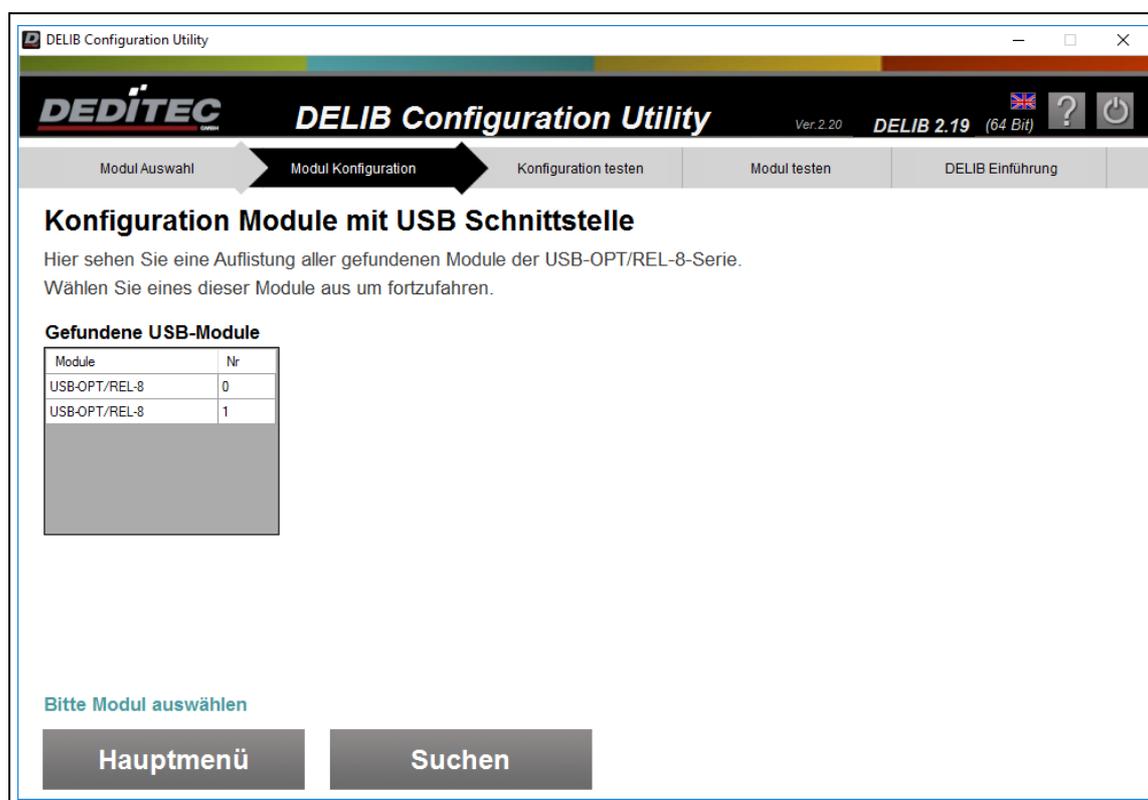


Schritt 4

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.

4.2.3.2.2. Modul Konfiguration Ethernet

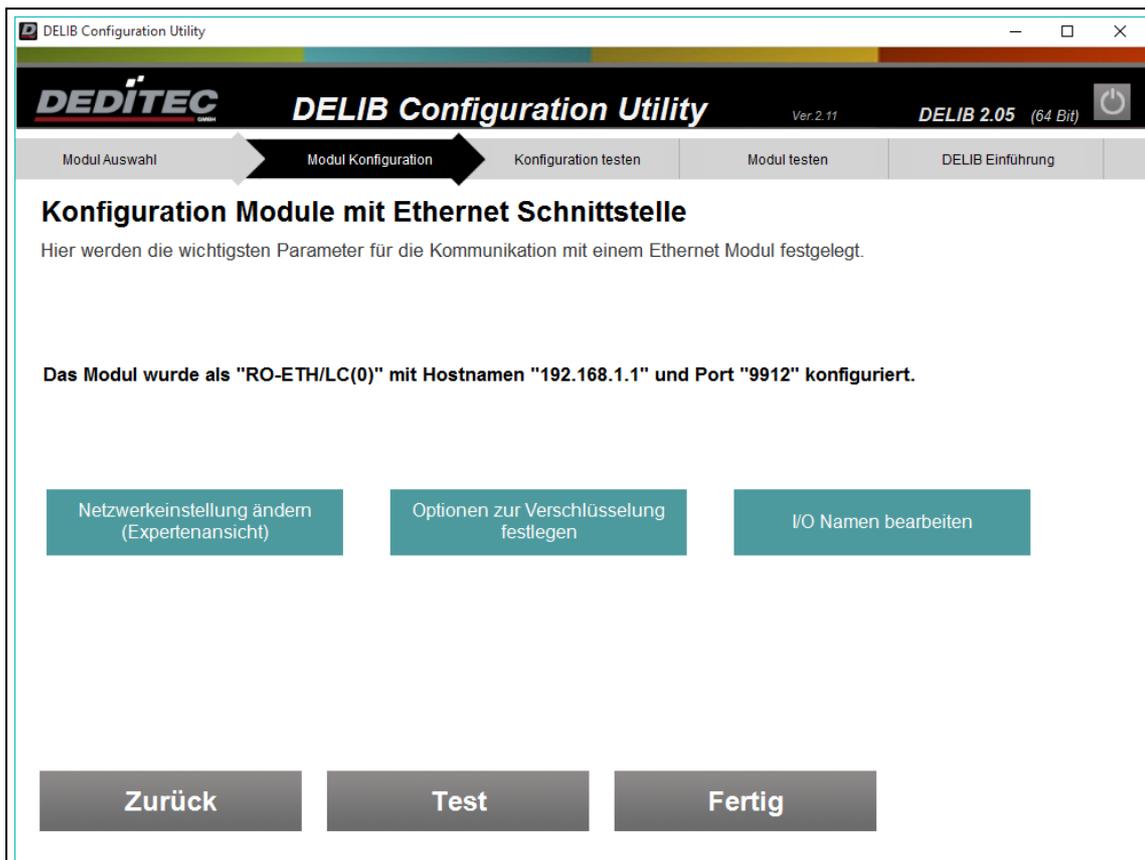
Bei der Konfiguration von Ethernet Produkten kann das Netzwerk automatisch nach DEDITEC Produkten durchsucht werden.

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 'DELIB Configuration Utility', version 'Ver. 2.11', and 'DELIB 2.05 (64 Bit)'. A progress bar at the top indicates the current step is 'Modul Konfiguration'. The main content area is titled 'Konfiguration Module mit Ethernet Schnittstelle' and contains the following text: 'Hier werden die wichtigsten Parameter für die Kommunikation mit einem Ethernet Modul festgelegt.' and 'Das Modul wurde als "RO-ETH/LC(0)" mit Hostnamen "192.168.1.1" und Port "9912" konfiguriert.' Below this, there are several interactive elements: a button 'Netzwerkeinstellung ändern (Expertenansicht)', a 'Verschlüsselung' dropdown menu set to 'deaktiviert', a 'Passwort' input field, a button 'I/O Namen bearbeiten', and a button 'Einstellungen zur Verschlüsselung auf das Modul übertragen'. At the bottom, there 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.

4.2.3.2.2.1. 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.

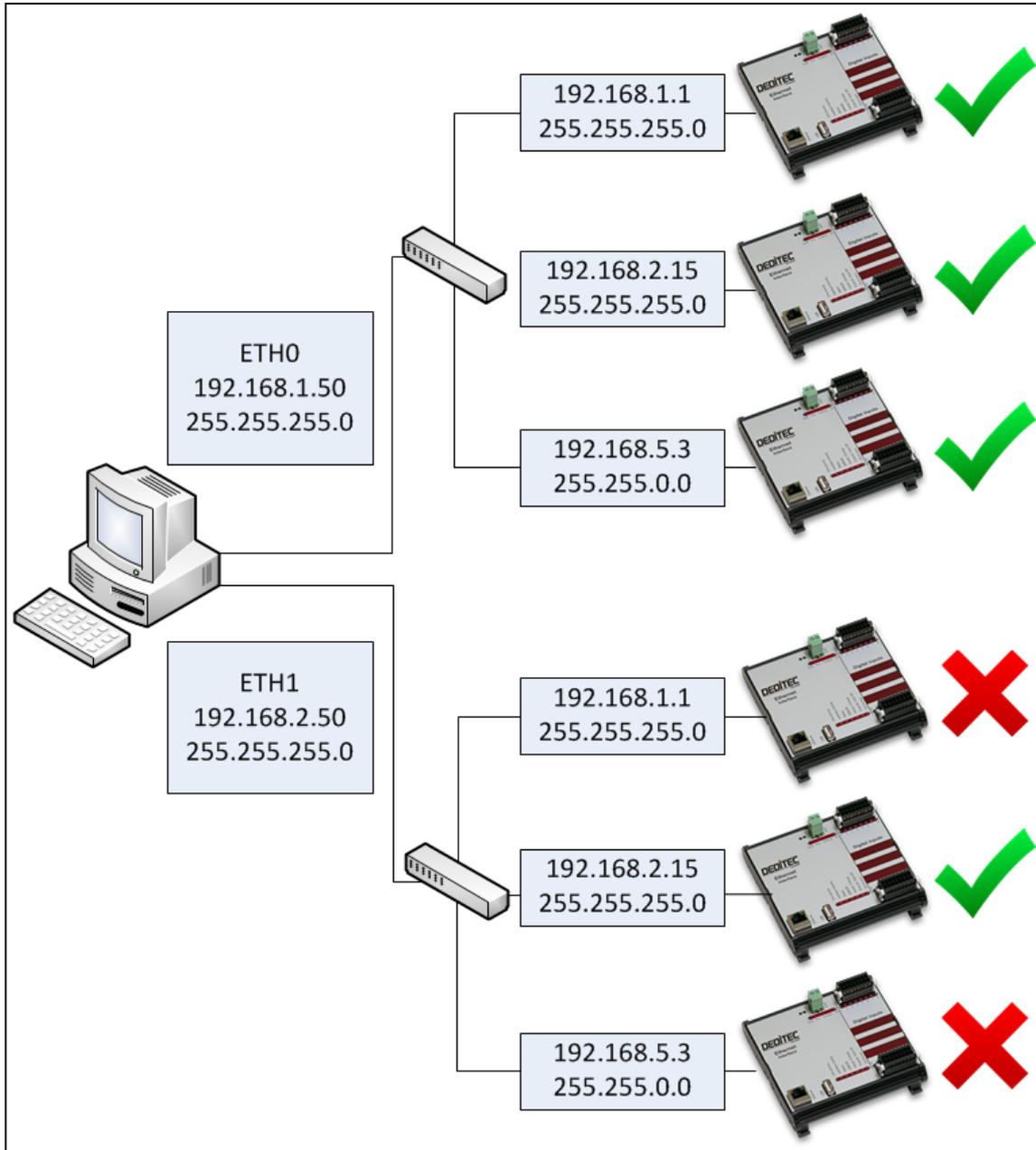
Achtung:

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.

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 items: 'Modul Auswahl' (highlighted), 'Modul Konfiguration', 'Konfiguration testen', 'Modul testen', and 'DELIB Einführung'. The main content area is titled 'Konfiguration Module mit Ethernet Schnittstelle' and contains the following text: 'Hier sehen Sie eine Auflistung der gefundenen Module im Netzwerk. Wählen Sie eines dieser Module aus um fortzufahren.' Below this is a table titled 'DEDITEC Ethernet Module im Netzwerk' with the following data:

MAC	DHCP	IP	Modulname	Produkttyp	WP
00:C0:D5:02:00:72	-	192.168.1.144	RO-ETH-LC	RO-ETH/LC(26)	WP
00:C0:D5:02:00:46	DHCP	192.168.1.127	ETH-O8-R8-Conrad	ETH-OPT/REL-8(27)	-
00:C0:D5:02:00:34	-	192.168.1.24	BS-ETH	BS-ETH(22)	-
00:C0:D5:01:04:2D	-	192.168.1.219	DEDITEC RO-ETH-Conrad	RO-ETH(8)	-
00:C0:D5:FF:FF:FF	-	192.168.1.25	BS-ETH-Conrad	BS-ETH(22)	WP
00:C0:D5:01:10:8F	-	192.168.1.222	DEDITEC RO-ETH Webc...	RO-ETH(8)	-

At the bottom, there is a prompt 'Bitte Modul auswählen' and three buttons: 'Hauptmenü', 'Suchen', and 'Weiter'.

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.

4.2.3.2.2. 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.

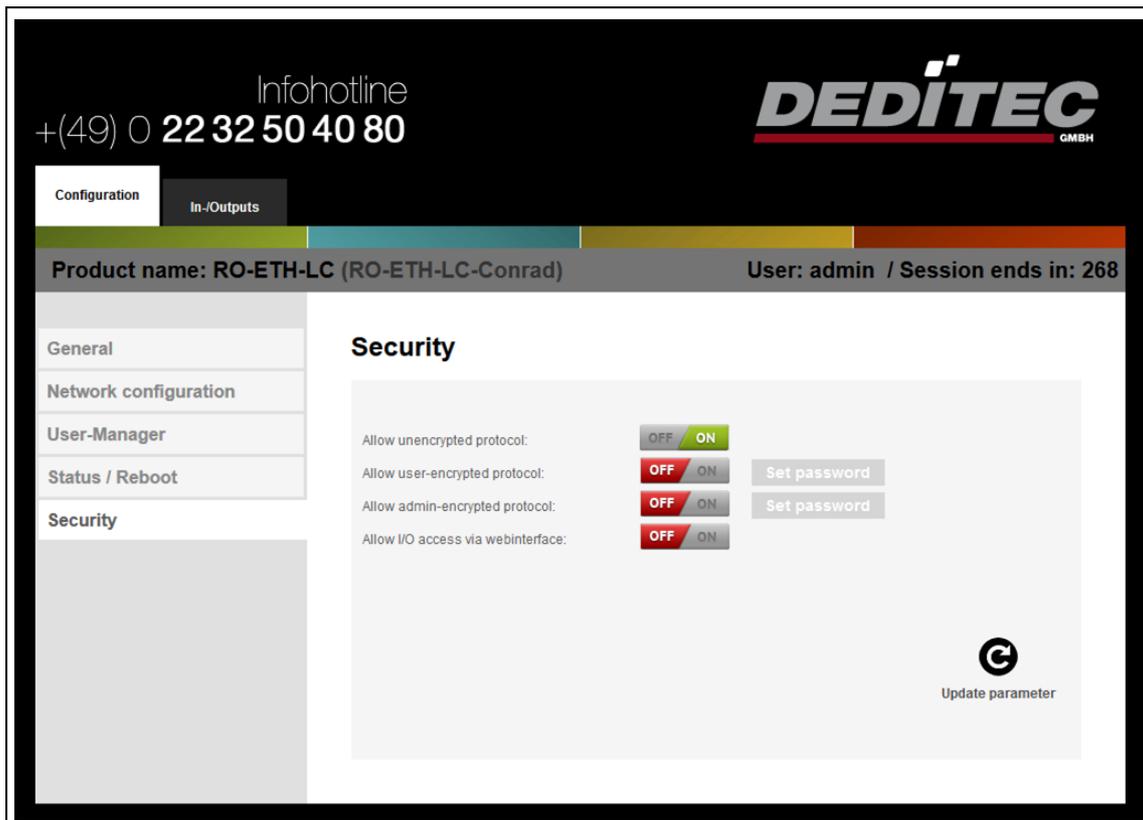
Manuelle Konfiguration

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**)



The screenshot displays the DEDITEC web interface. At the top left, there is an 'Infohotline' number: +(49) 0 22 32 50 40 80. The DEDITEC 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: 268'. On the left side, there is a navigation menu with options: 'General', 'Network configuration', 'User-Manager', 'Status / Reboot', and 'Security' (selected). The main content area is titled 'Security' and contains four settings, each with an 'OFF' and 'ON' toggle switch:

- Allow unencrypted protocol: OFF (selected) / ON
- Allow user-encrypted protocol: OFF (selected) / ON, with a 'Set password' button to the right.
- Allow admin-encrypted protocol: OFF (selected) / ON, with a 'Set password' button to the right.
- Allow I/O access via webinterface: OFF (selected) / ON

At the bottom right of the settings area, there is a circular refresh icon and the text 'Update parameter'.

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 management interface. At the top left, there is an 'Infohotline' number: +(49) 0 22 32 50 40 80. The DEDITEC GMBH logo is in the top right. Below the header, there are navigation tabs for 'Configuration' and 'In-/Outputs'. A status bar shows 'Product name: RO-ETH-LC (RO-ETH-LC-Conrad)' and 'User: admin / Session ends in: 223'. A left sidebar contains menu items: 'General', 'Network configuration', 'User-Manager', 'Status / Reboot', and 'Security'. The main content area is titled 'Security' and contains several configuration options with toggle switches and input fields:

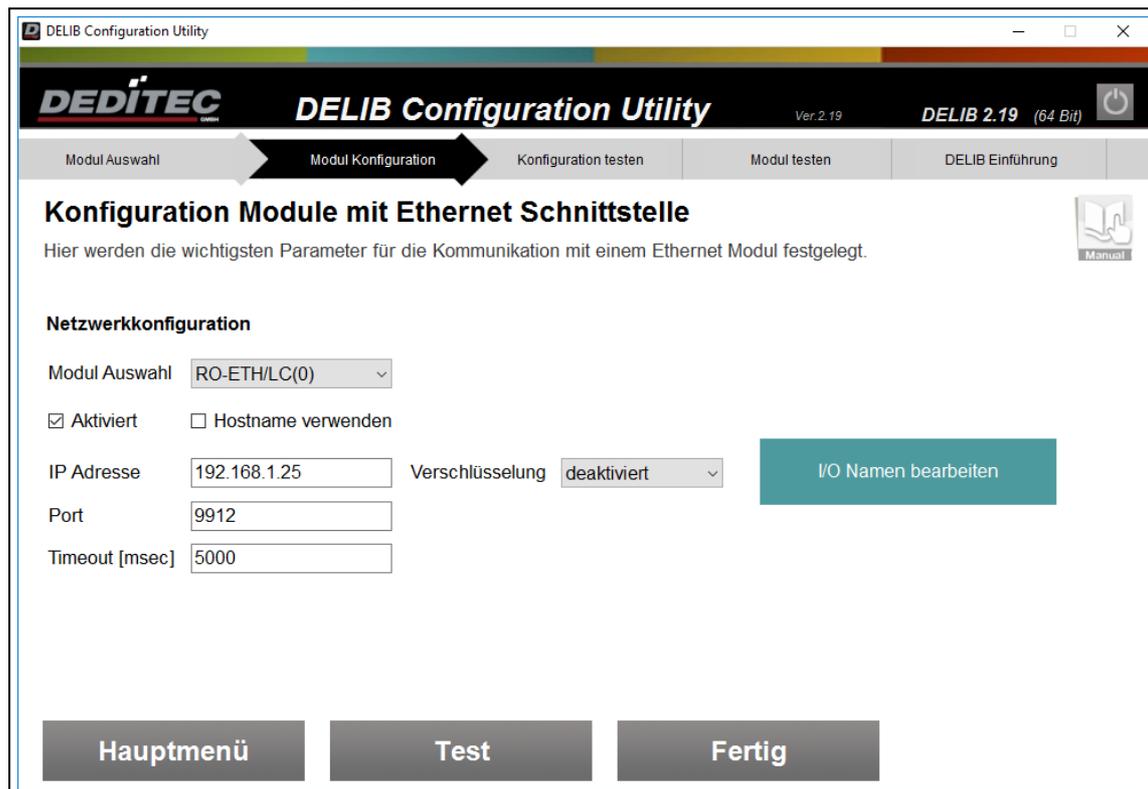
- Allow unencrypted protocol: OFF (selected) / ON
- Allow user-encrypted protocol: OFF (selected) / ON, with a 'Set password' button
- Allow admin-encrypted protocol: OFF (selected) / ON
- New password: [password input field]
- Confirm password: [password input field]
- Allow I/O access via webinterface: OFF (selected) / ON

At the bottom right of the configuration area, there is a circular refresh icon and the text 'Update parameter'.

Schritt 2 - Konfiguration PC-Seite

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

Hier bei Verschlüsselung "Admin" auswählen.



The screenshot shows the 'DELIB Configuration Utility' window. The title bar reads 'DELIB Configuration Utility'. The main header features the 'DEDITEC' logo, the product name 'DELIB Configuration Utility', the version 'Ver. 2.19', and 'DELIB 2.19 (64 Bit)'. A navigation bar at the top contains five tabs: 'Modul Auswahl', 'Modul Konfiguration' (which is active and highlighted with a black arrow), 'Konfiguration testen', 'Modul testen', and 'DELIB Einführung'. Below the navigation bar, the main content area is titled '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 of this text is a 'Manual' icon. Under the heading 'Netzwerkkonfiguration', there are several configuration options: 'Modul Auswahl' is a dropdown menu set to 'RO-ETH/LC(0)'; 'Aktiviert' is checked, while 'Hostname verwenden' is unchecked; 'IP Adresse' is a text input field containing '192.168.1.25'; 'Port' is a text input field containing '9912'; 'Timeout [msec]' is a text input field containing '5000'; and 'Verschlüsselung' is a dropdown menu set to 'deaktiviert'. A teal button labeled 'I/O Namen bearbeiten' is positioned to the right of the encryption dropdown. At the bottom of the window, there are three large, dark grey buttons: 'Hauptmenü', 'Test', and 'Fertig'.

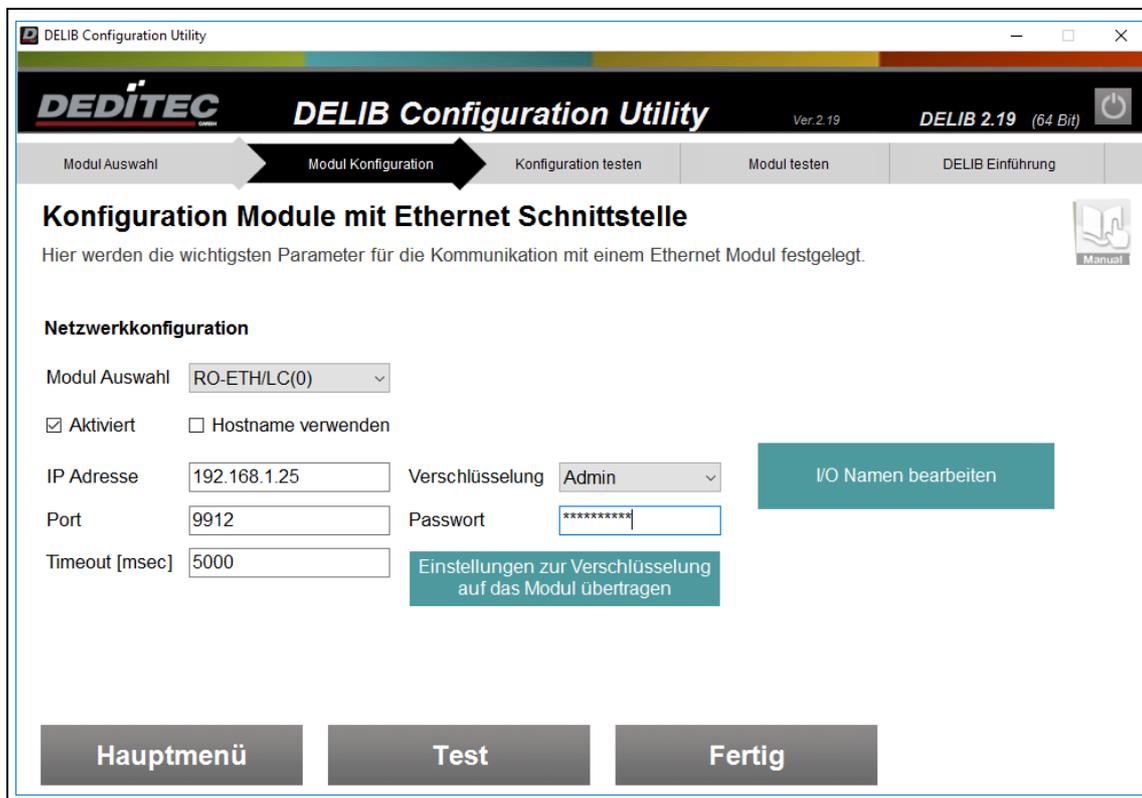
Im neu erscheinenden Passwort Feld das Passwort aus Schritt 1 eintragen.
Mit Test kann die neue verschlüsselten Admin-Modus Kommunikation getestet werden.

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 product name 'DELIB Configuration Utility', the version 'Ver. 2.19', and 'DELIB 2.19 (64 Bit)'. A progress bar at the top indicates the current step: 'Konfiguration testen', with other steps being 'Modul Auswahl', 'Modul Konfiguration', 'Modul testen', and 'DELIB Einführung'. The main content area is titled 'Konfiguration Module mit Ethernet Schnittstelle' and includes a sub-header 'Netzwerkkonfiguration'. The configuration fields are: 'Modul Auswahl' (dropdown: RO-ETH/LC(0)), 'Aktiviert' (checked checkbox), 'Hostname verwenden' (unchecked checkbox), 'IP Adresse' (text: 192.168.1.25), 'Verschlüsselung' (dropdown: Admin), 'Port' (text: 9912), 'Passwort' (text: *****), and 'Timeout [msec]' (text: 5000). A teal button 'I/O Namen bearbeiten' is visible. A teal tooltip box contains the text 'Einstellungen zur Verschlüsselung auf das Modul übertragen'. At the bottom, a green status message reads 'Modul-Kommunikation ist OK! Aktuelle Modul-Firmware ist 2.20'. Three large buttons are at the bottom: 'Hauptmenü', 'Test', and 'Fertig'.

Automatische Konfiguration

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



The screenshot shows the 'DELIB Configuration Utility' window. The title bar indicates 'Ver. 2.19' and 'DELIB 2.19 (64 Bit)'. The main menu includes 'Modul Auswahl', 'Modul Konfiguration', 'Konfiguration testen', 'Modul testen', and 'DELIB Einführung'. The current step is 'Konfiguration Module mit Ethernet Schnittstelle'. Below this, there is a section for 'Netzwerkkonfiguration' with the following fields: 'Modul Auswahl' (RO-ETH/LC(0)), 'Aktiviert' (checked), 'Hostname verwenden' (unchecked), 'IP Adresse' (192.168.1.25), 'Port' (9912), 'Timeout [msec]' (5000), 'Verschlüsselung' (Admin), and 'Passwort' (masked with asterisks). A teal button labeled 'I/O Namen bearbeiten' is visible. A teal tooltip box contains the text 'Einstellungen zur Verschlüsselung auf das Modul übertragen'. At the bottom, there are three buttons: 'Hauptmenü', 'Test', and '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.

The screenshot shows the 'DELIB Configuration Utility' window. The title bar includes the DEDITEC logo, the application name 'DELIB Configuration Utility', version 'Ver. 2.19', and 'DELIB 2.19 (64 Bit)'. The main menu has five tabs: 'Modul Auswahl', 'Modul Konfiguration' (active), 'Konfiguration testen', 'Modul testen', and 'DELIB Einführung'. The current screen is titled 'Konfiguration Module mit Ethernet Schnittstelle' and contains the following elements:

- Netzwerkkonfiguration**
- Modul Auswahl: RO-ETH/LC(0)
- Aktiviert, Hostname verwenden
- IP Adresse: 192.168.1.25, Verschlüsselung: Admin
- Port: 9912, Passwort: *****
- Timeout [msec]: 5000
- Buttons: 'I/O Namen bearbeiten', 'Einstellungen zur Verschlüsselung auf das Modul übertragen', 'Authentifizierung abbrechen' (highlighted in teal)
- Footer: 'Schritt 1: DIP-Schalter 2 (Schreibschutz) des ETH-Moduls auf Stellung 1 (0=OFF, 1=ON) setzen', 'Hauptmenü', 'Test', 'Fertig', and three small squares.

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)'. A progress bar at the top indicates the current step: 'Modul Konfiguration'. Below the header, a navigation bar contains buttons for 'Modul Auswahl', 'Modul Konfiguration', 'Konfiguration testen', 'Modul testen', and 'DELIB Einführung'. The main content area is titled 'Konfiguration Module mit Ethernet Schnittstelle' and includes a sub-section 'Netzwerkkonfiguration'. This section contains several input fields: 'Modul Auswahl' (set to 'RO-ETH/LC(0)'), 'Aktiviert' (checked), 'IP Adresse' (192.168.1.25), 'Port' (9912), and 'Timeout [msec]' (5000). There are also dropdown menus for 'Verschlüsselung' (Admin) and 'Passwort' (masked with asterisks). A 'Manual' icon is visible in the top right. At the bottom, a blue button reads 'Authentifizierung abbrechen'. A status bar at the very bottom contains 'Hauptmenü', 'Test', and 'Fertig' buttons, along with three indicator lights.

DELIB Configuration Utility

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

Modul Auswahl Modul Konfiguration Konfiguration testen Modul testen DELIB Einführung

Konfiguration Module mit Ethernet Schnittstelle

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

Netzwerkkonfiguration

Modul Auswahl: RO-ETH/LC(0)

Aktiviert Hostname verwenden

IP Adresse: 192.168.1.25 Verschlüsselung: Admin

Port: 9912 Passwort: *****

Timeout [msec]: 5000

Einstellungen zur Verschlüsselung auf das Modul übertragen

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

Authentifizierung abbrechen

Hauptmenü Test Fertig

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

The screenshot shows the 'DELIB Configuration Utility' window. The title bar reads 'DELIB Configuration Utility'. The main header features the 'DEDITEC' logo, the product name 'DELIB Configuration Utility', the version 'Ver. 2.19', and 'DELIB 2.19 (64 Bit)'. A navigation bar contains five steps: 'Modul Auswahl', 'Modul Konfiguration' (highlighted with a black arrow), 'Konfiguration testen', 'Modul testen', and 'DELIB Einführung'. A 'Manual' icon is visible on the right.

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

Buttons: 'I/O Namen bearbeiten', 'Einstellungen zur Verschlüsselung auf das Modul übertragen'

Authentifizierung erfolgreich

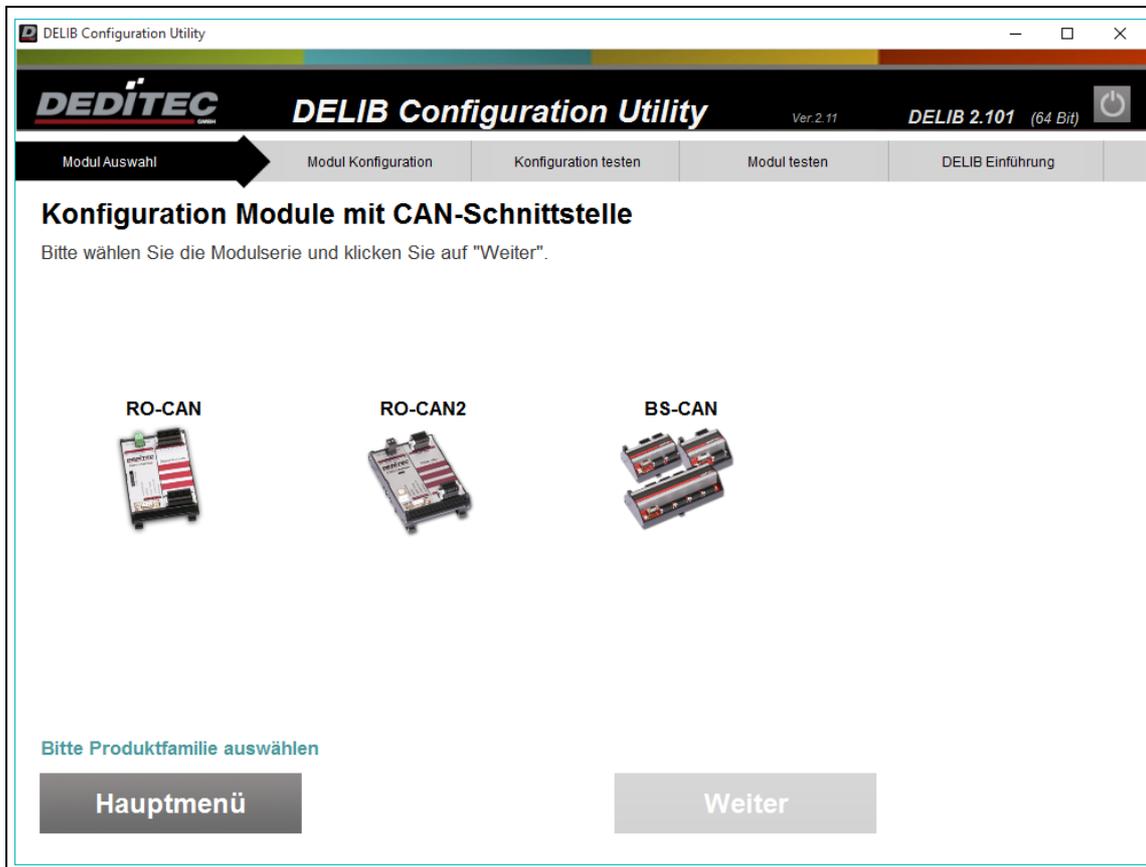
Buttons: 'Hauptmenü', 'Test', 'Fertig', and a three-dot menu icon.

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

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

4.2.3.2.3. Modul Konfiguration CAN

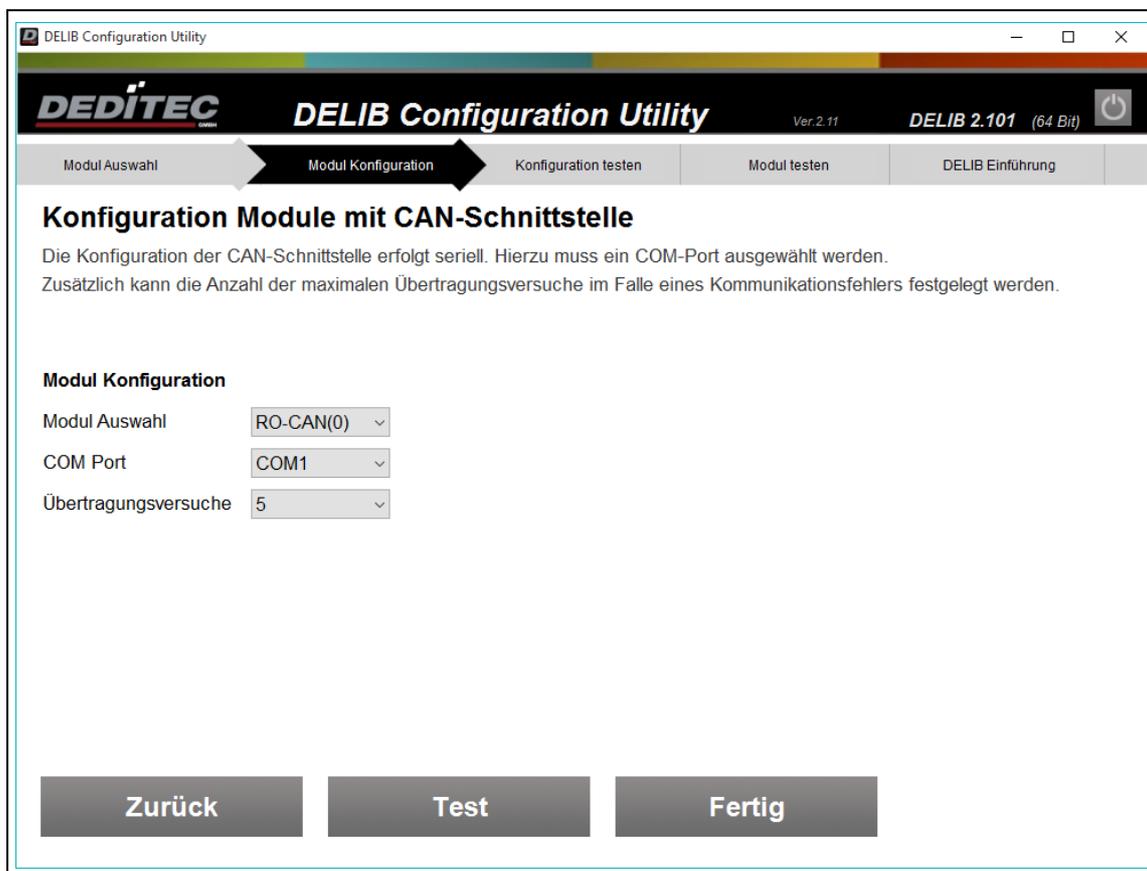
Wenn Sie eine neue CAN-Konfiguration erstellen, müssen Sie zuerst die Produktfamilie auswählen.



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.

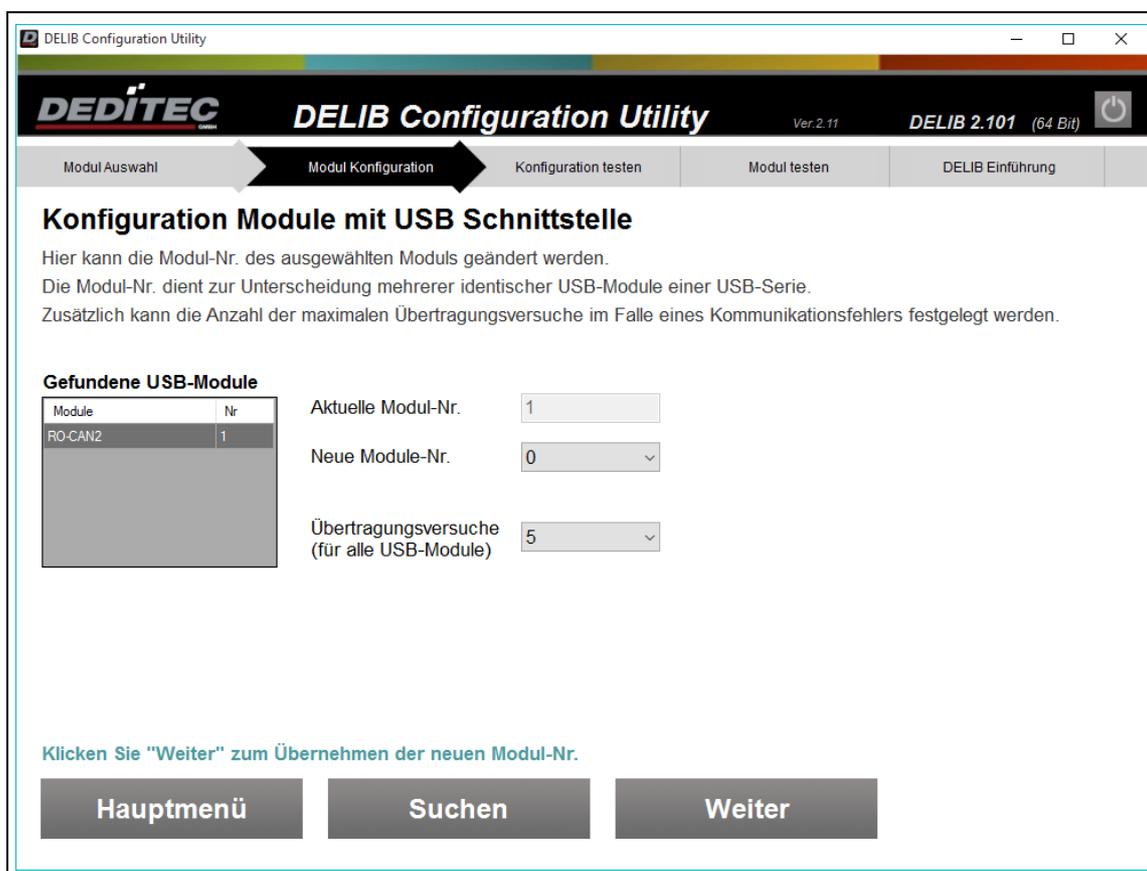
Mit dem Button "Test" wird geprüft, ob eine Kommunikation über den angegebenen COM Port möglich ist.



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.

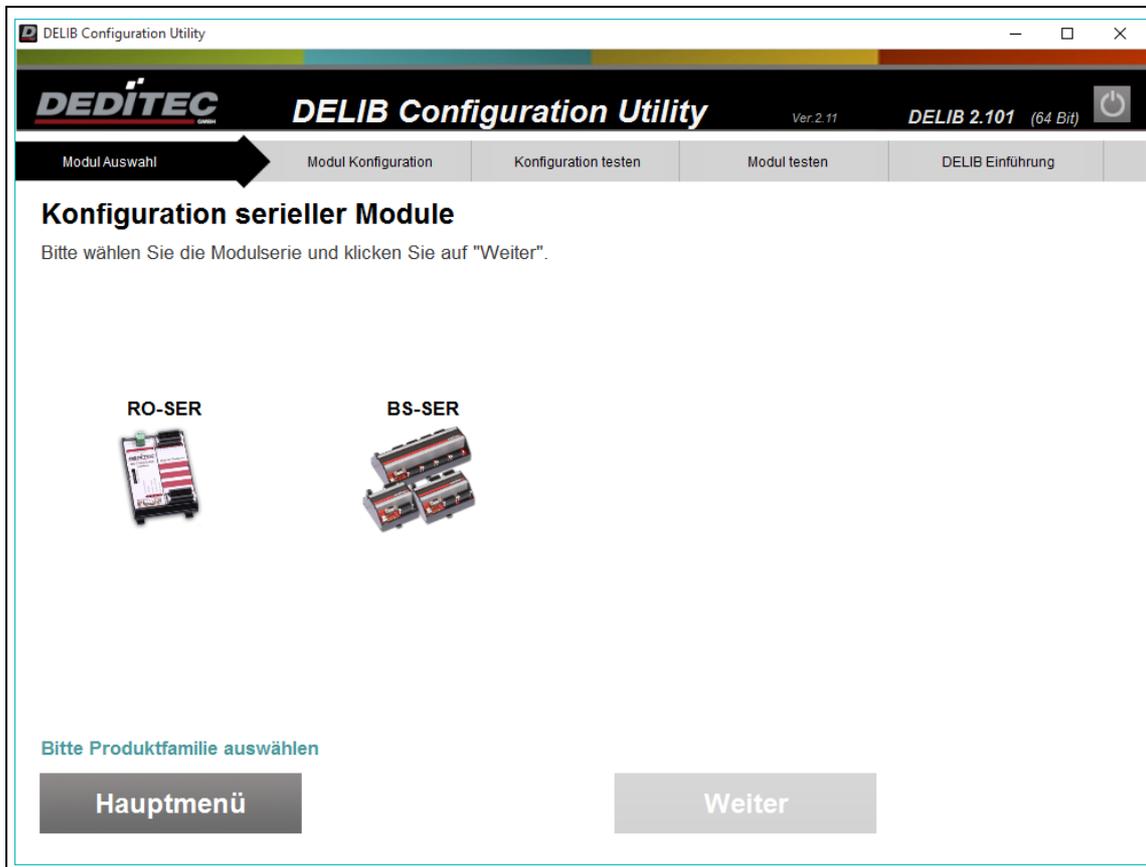
siehe Kapitel **Beispiel zur Konfiguration identischer USB-Module**

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



4.2.3.2.4. 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.

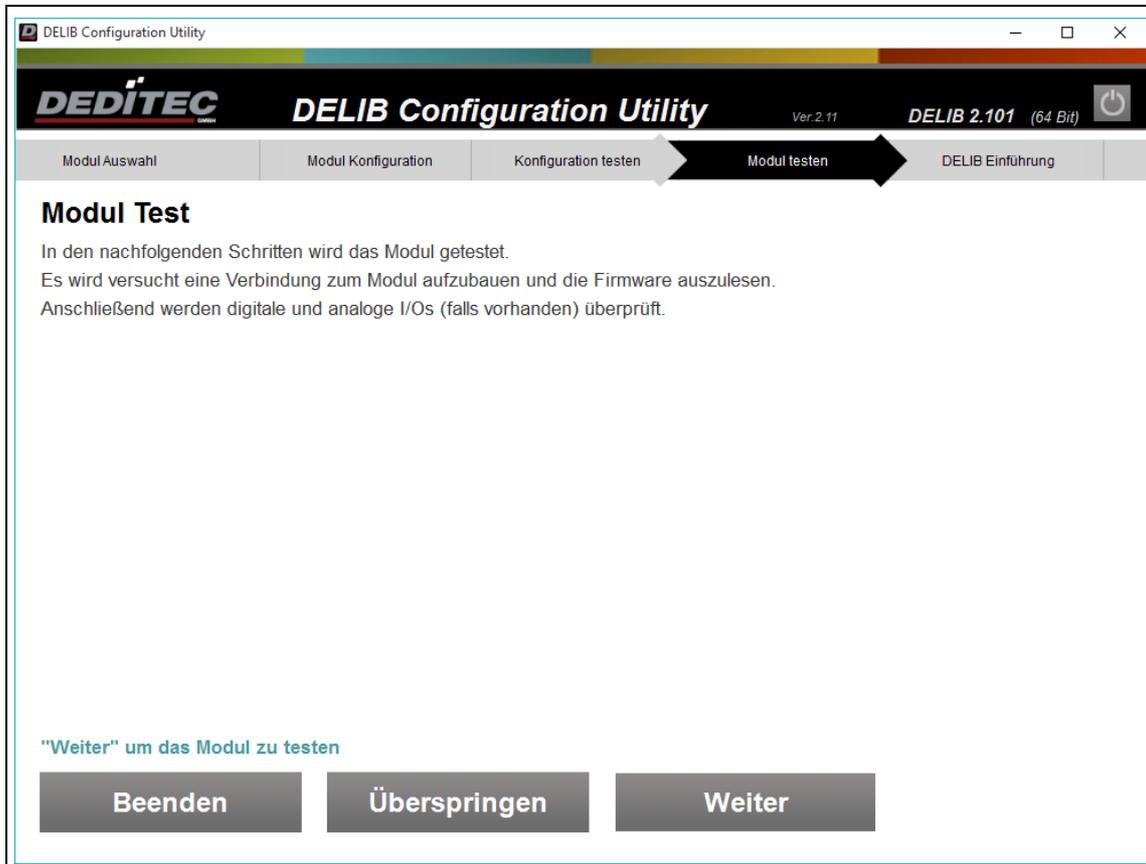
Zusätzlich muss die Anzahl der Übertragungsversuche im Falle eines Kommunikationsfehlers angegeben werden.

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

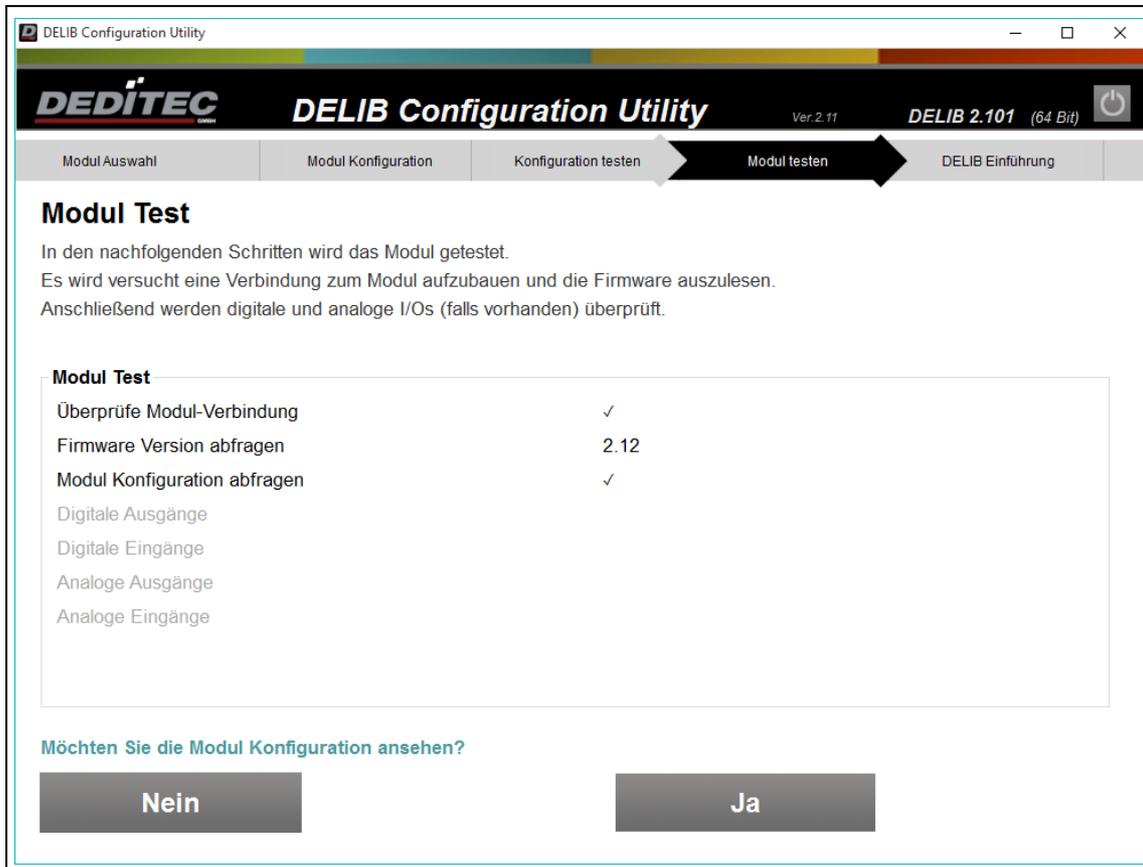
The screenshot shows the 'DELIB Configuration Utility' window. The title bar includes the logo 'DEDITEC', the application name 'DELIB Configuration Utility', the version 'Ver. 2.11', and the product name 'DELIB 2.101 (64 Bit)'. The main menu has five steps: 'Modul Auswahl', 'Modul Konfiguration', 'Konfiguration testen', 'Modul testen', and 'DELIB Einführung'. The 'Modul Konfiguration' step is currently active. The main content area is titled 'Konfiguration serieller Module' and contains the instruction: 'Hier werden die wichtigsten Parameter für die Kommunikation mit einem seriellen Modul festgelegt.' Below this, there are three configuration fields: 'Modul Konfiguration' (header), 'Modul Auswahl' (dropdown menu showing 'RO-SER(0)'), 'COM Port' (dropdown menu showing 'COM1'), and 'Übertragungsversuche' (dropdown menu showing '5'). At the bottom of the window, there are three buttons: 'Zurück', 'Test', and 'Fertig'.

4.2.3.3. Modul testen

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



Test der Firmware und Anzeige der Modul-Info.



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

DT_ModuleInfo X

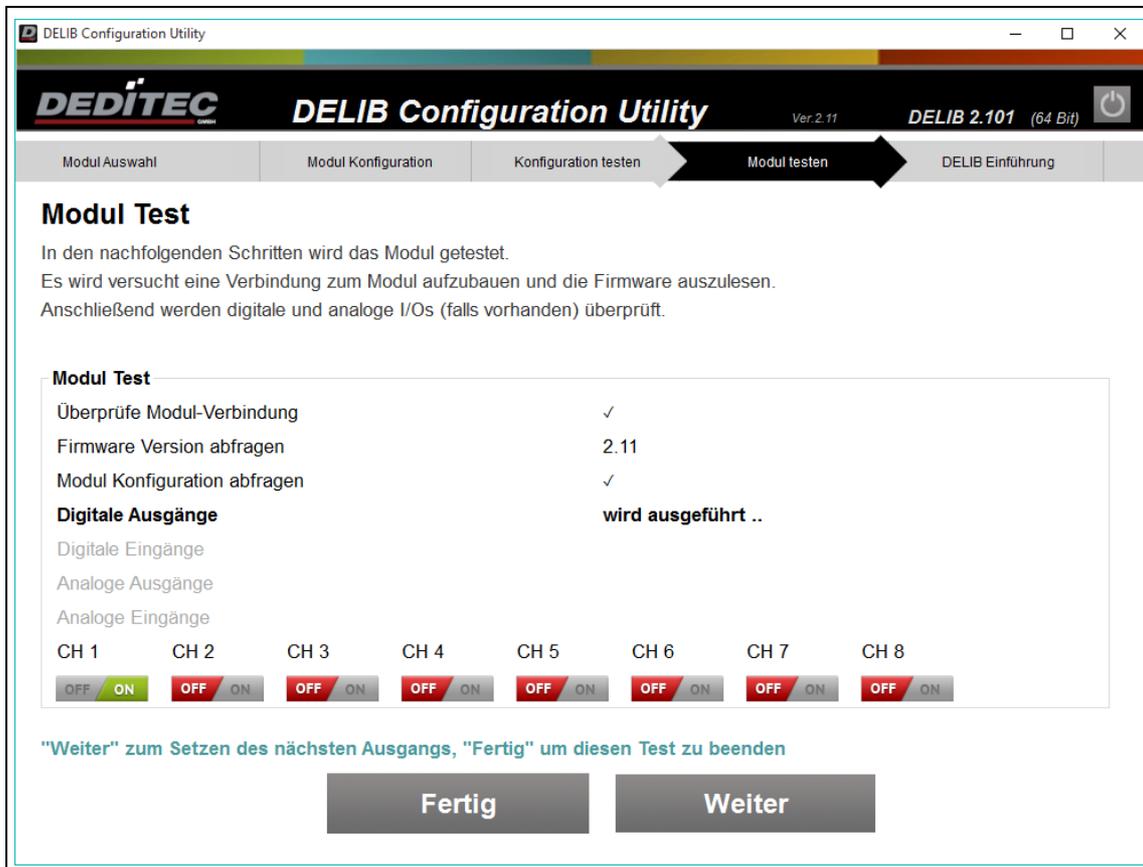
General	Digital I/O	Analog I/O	Special
SW_FEATURE_1 e3373007	Digital Inputs 8	Analog Inputs 16	Stepper 2
HW_INTERFACE_1 01000003	Digital Outputs 8	Analog Outputs 4	
Firmware-Revision 2.11	Digital In-/Outputs 0	Temperature Inputs 4	
Main-Module: RO	Digital Input FlipFlops 8		
	Digital Input Counter 8		
	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-O8_R8	FW:2.10
Sub: 2 - RO-STEPPER2	FW:1.29
Sub: 3 - RO-PT100	FW:1.03

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

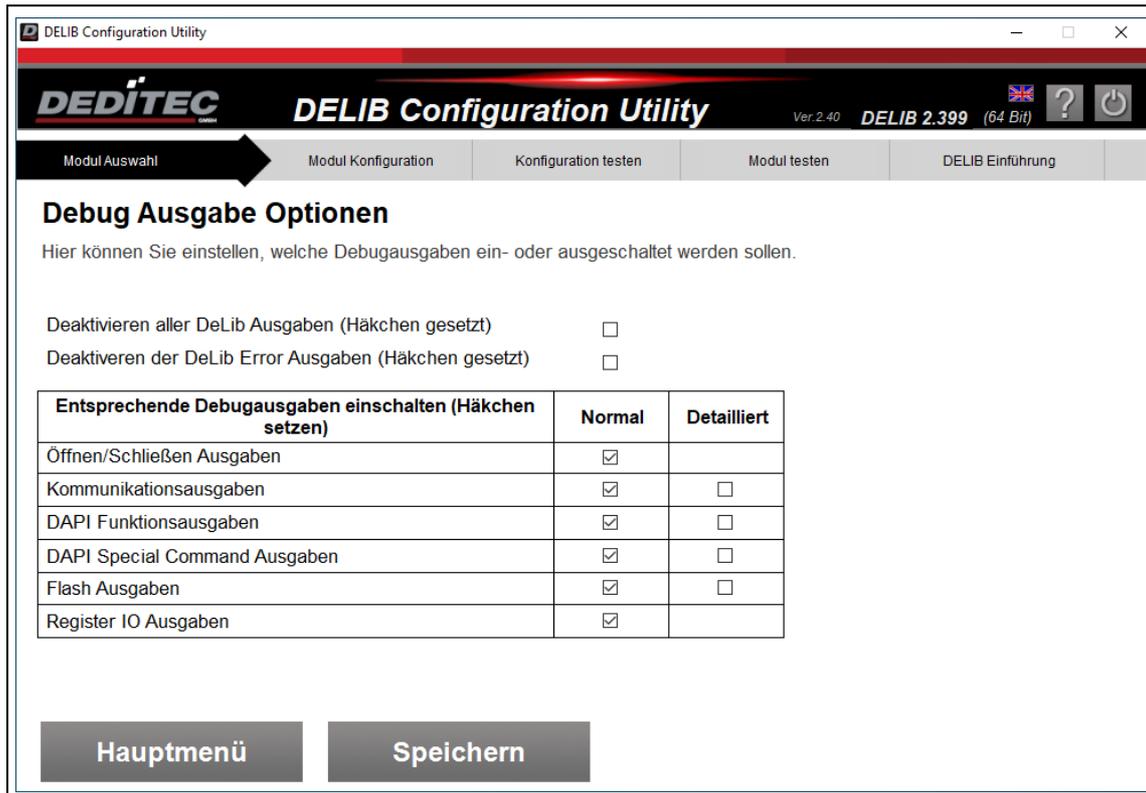


Wurden alle Tests erfolgreich durchlaufen, ist das Produkt einsatzbereit.

4.2.3.4. Debug Optionen einstellen

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

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



The screenshot shows the 'DELIB Configuration Utility' window. The title bar includes the DEDITEC logo, the application name 'DELIB Configuration Utility', version 'Ver. 2.40', and 'DELIB 2.399 (64 Bit)'. The main menu has five items: 'Modul Auswahl' (highlighted with a black arrow), 'Modul Konfiguration', 'Konfiguration testen', 'Modul testen', and 'DELIB Einführung'. The 'Debug Ausgabe Optionen' section contains the following text and controls:

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

Deaktivieren aller DeLib Ausgaben (Häkchen gesetzt)

Deaktivieren der DeLib Error Ausgaben (Häkchen gesetzt)

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

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

4.2.4. Benutzen des Modulselectors

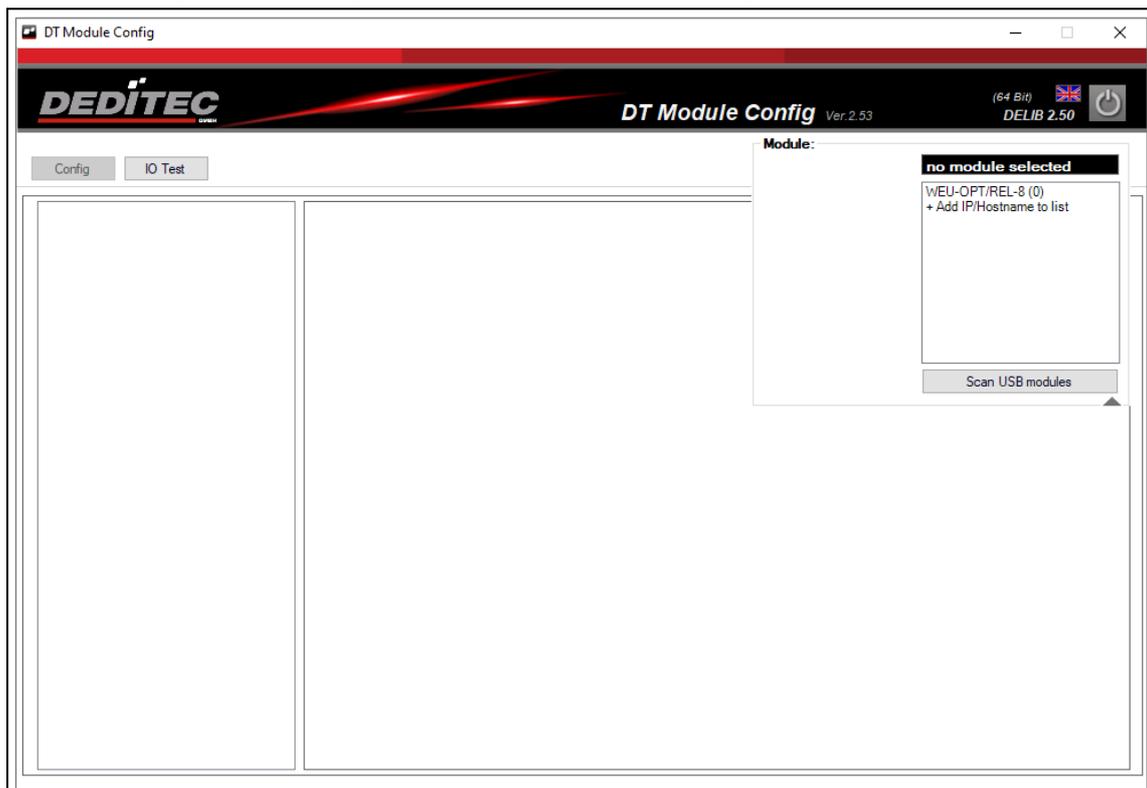
Um unsere Produkte mit der DEDITEC-Software benutzen zu können, müssen diese über den Modul Selector ausgewählt werden.

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

1. per USB

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

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

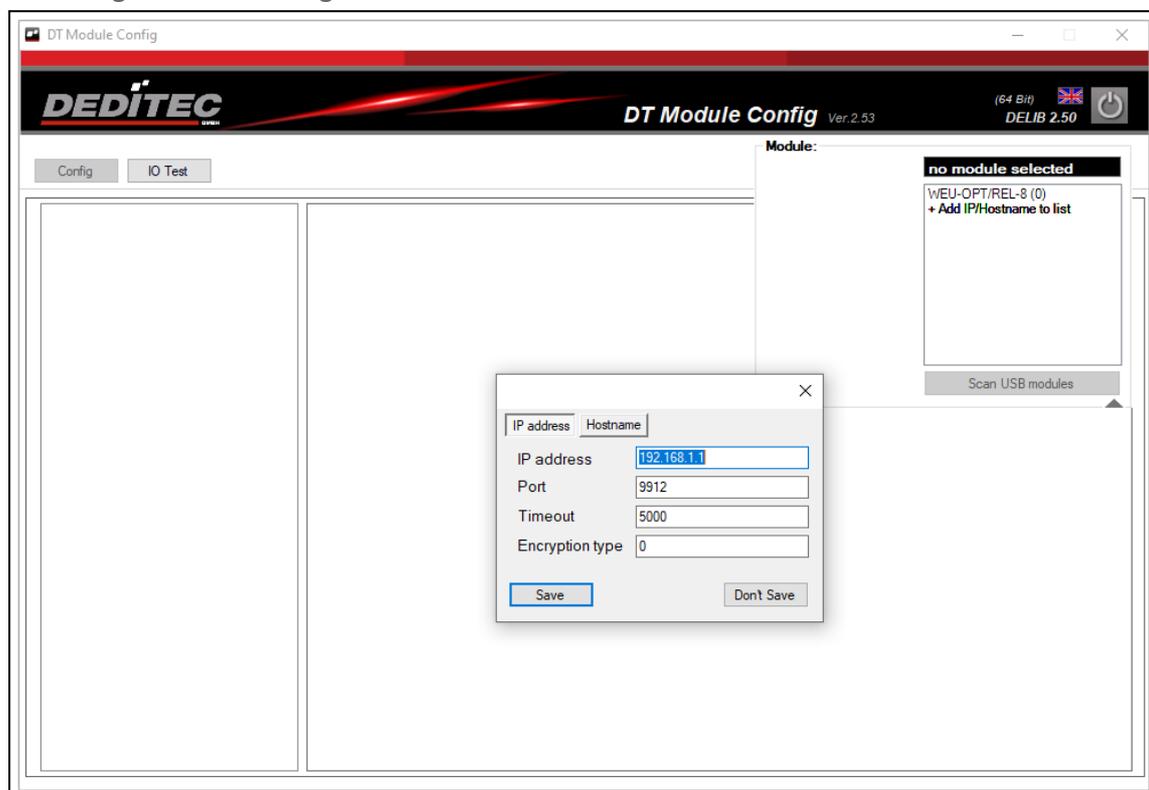


2. per Ethernet

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

Fragen Sie hierfür gegebenenfalls Ihren Systemadministrator.

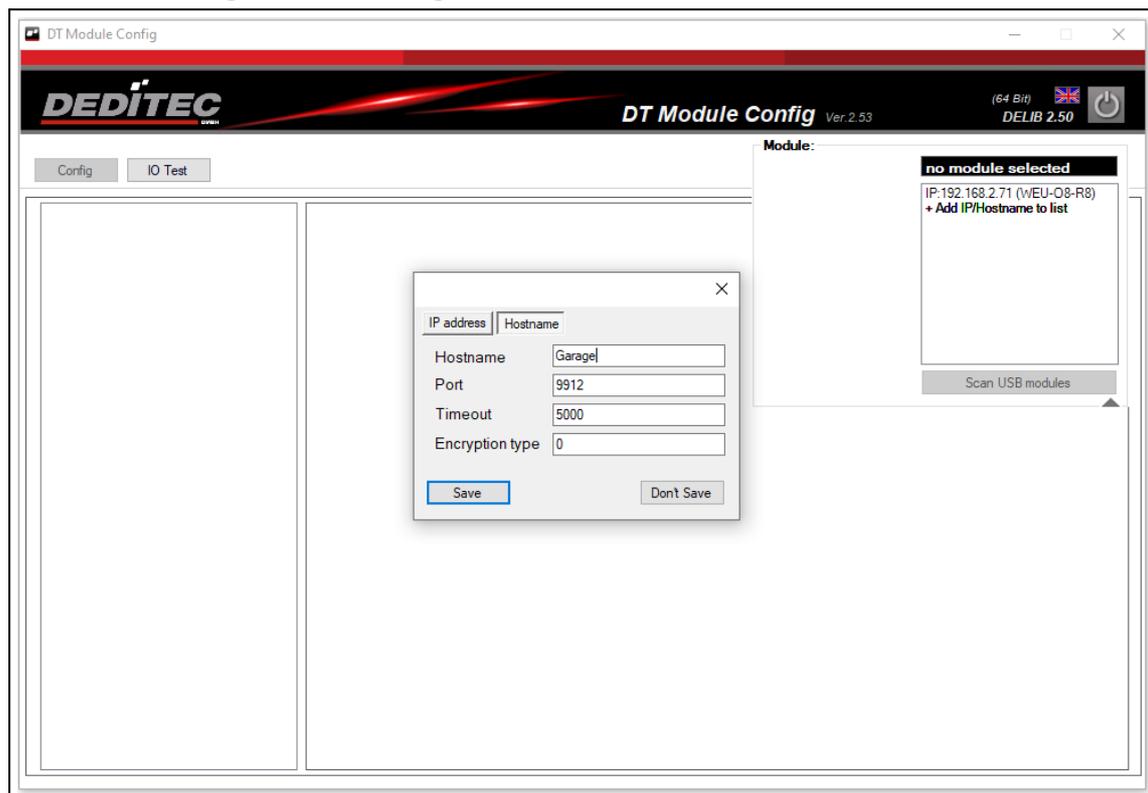
Klicken Sie im Modul Selector auf "Add IP/Hostname to list", tragen Sie dort die automatisch erhaltene IP-Adresse unter dem Reiter "IP Adresse" ein und bestätigen Sie die Eingabe mit "Save".



Ist Ihr Modul im DHCP-Modus (siehe Kapitel: **LAN Netzwerkkonfiguration**) können Sie 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. per 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.

4. per WPS (nur bei WEU-Modulen)

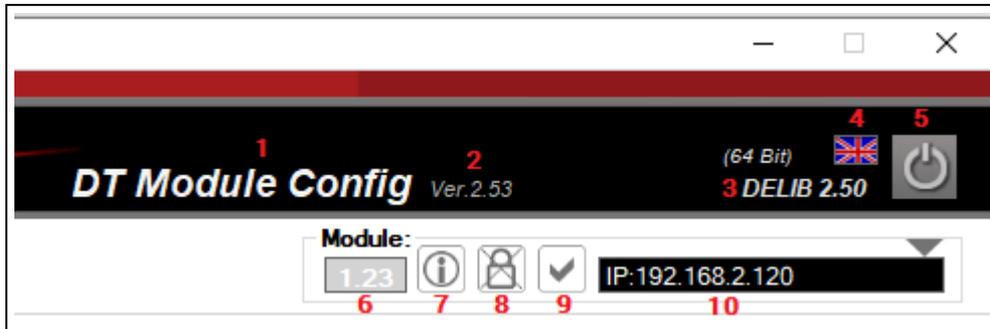
Ist das Modul noch nicht mit dem PC-Netzwerk verbunden, führen Sie den Verbindungsaufbau, wie im Kapitel CFG-Taster beschrieben, durch.

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

Fragen Sie hierfür gegebenenfalls Ihren Systemadministrator.

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

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



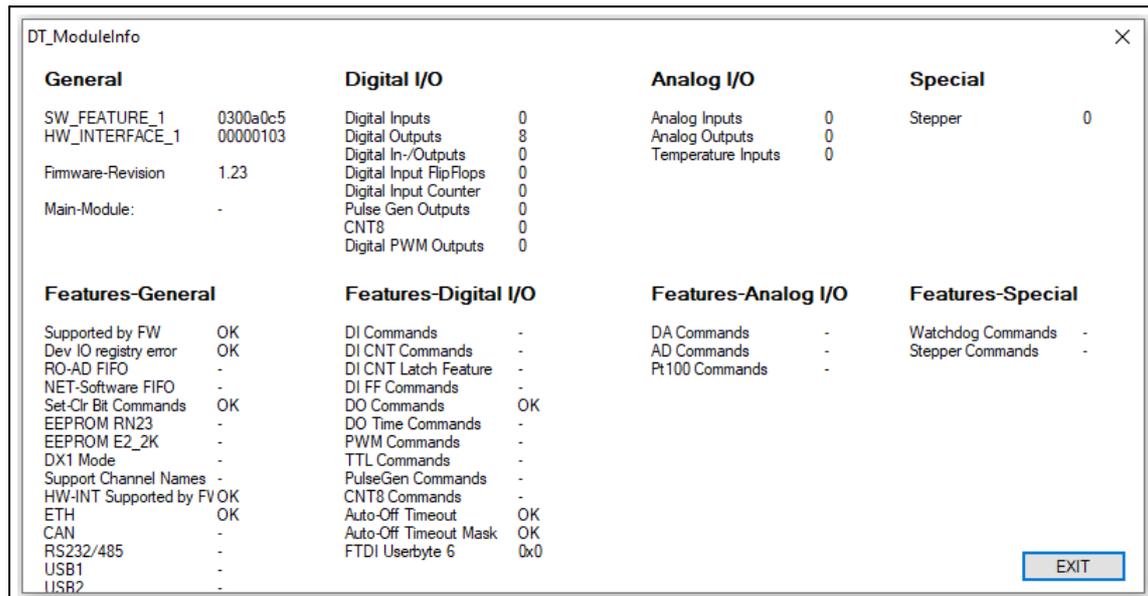
Beschreibung:

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 Fahnenymbol 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.



General		Digital I/O		Analog I/O		Special	
SW_FEATURE_1	0300a0c5	Digital Inputs	0	Analog Inputs	0	Stepper	0
HW_INTERFACE_1	00000103	Digital Outputs	8	Analog Outputs	0		
Firmware-Revision	1.23	Digital In-/Outputs	0	Temperature Inputs	0		
Main-Module:	-	Digital Input FlipFlops	0				
		Digital Input Counter	0				
		Pulse Gen Outputs	0				
		CNT8	0				
		Digital PWM Outputs	0				
Features-General		Features-Digital I/O		Features-Analog I/O		Features-Special	
Supported by FW	OK	DI Commands	-	DA Commands	-	Watchdog Commands	-
Dev IO registry error	OK	DI CNT Commands	-	AD Commands	-	Stepper Commands	-
RO-AD FIFO	-	DI CNT Latch Feature	-	Pt100 Commands	-		
NET-Software FIFO	-	DI FF Commands	-				
Set-Clr Bit Commands	OK	DO Commands	OK				
EEPROM RN23	-	DO Time Commands	-				
EEPROM E2_2K	-	PWM Commands	-				
DX1 Mode	-	TTL Commands	-				
Support Channel Names	-	PulseGen Commands	-				
HW-INT Supported by FV	OK	CNT8 Commands	-				
ETH	OK	Auto-Off Timeout	OK				
CAN	-	Auto-Off Timeout Mask	OK				
RS232/485	-	FTDI Userbyte 6	0x0				
USB1	-						
USB2	-						

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

4.2.5. DELIB Module Config

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

4.2.5.1. Modul Konfigurationen

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

4.2.5.1.1. Modul-Infoseite

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

Info
Hier finden Sie Informationen zu Ihren Moduleigenschaften

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

FormCfmModuleOverview (Ver. 1.01)

Modul-Name

Zeigt den Namen des aktuell verwendeten DEDITEC Modules an.

Modul-ID

Zeigt die ID Ihres verwendeten 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.

4.2.5.1.2. Modul-Identifikation

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

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

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

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

Dieser Vorgang wird durch drücken von "Stop" beendet.

Identifikation

Hier können Sie Ihr Modul identifizieren

Identify module

Modulidentifikation läuft!

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

- bei unseren ETH_LC-Modulen blinkt die 'Int.Act' LED
- bei unseren WEU-Modulen blinkt die 'Status-LED'

Das Bedienen der 'Stop' Taste beendet diesen Vorgang.

FormCfmModulIdent (Ver. 1.01)

4.2.5.1.3. LAN Netzwerkinformationen

Alle wichtigen LAN Netzwerkinformationen auf einen Blick.

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

Info

Hier finden Sie die Netzwerkinformationen des ausgewählten Ethernet-Produktes

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

Auto Refresh

Form für Netzwerk LAN Info (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.

4.2.5.1.4. LAN Netzwerkeinstellungen

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

Konfiguration

Hier können Sie die Netzwerkeinstellungen des ausgewählten Ethernet-Produktes ändern

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

Form/ToNetwork/ANConfin/Ver. 1.011

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

4.2.5.1.5. 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.

4.2.5.1.6. 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="i"/>

Form für Netzwerk WiFi Config (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:

Werkseinstellungen	
Board name	Modul abhängig
WLAN active	off
Routername	TESTssid
Password	TESTpwd

4.2.5.1.7. 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

Router name

Password

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_AVer_1.011

4.2.5.1.8. 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"/>

FirmComNetwork NTP (Ver. 1.01)

NTP service active

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

Server

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

Port

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

Timezone

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

Werkseinstellungen laden

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

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

4.2.5.1.9. 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"/>

FormCtoSerial (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.

Registermodus

Deaktivieren Sie den Registermodus um den Textmodus zu aktivieren.

4.2.5.1.10. 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"/>

FormCfoSubmoduleIIONames (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.

4.2.5.1.11. CAN Konfiguration

4.2.5.1.11.1. CAN Status

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

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

Interface

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

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

Auto Refresh

FomCfoCANStatus (Ver. 1.00)

CAN-Baudrate

Zeigt die aktuell eingestellte Baud rate Ihres CAN-Interfaces an

DT-CAN-CMD-MODE- Module-Address

DT-CAN-CMD-MODE - Response-Address

CAN is active

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

Use EXT ID

CAN config mode selection

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

Statistik TX/RX

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

DT-CAN-CMD-MODE		Frames total
TX		0
RX		0

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

Auto Refresh

FormCfoCANStatistic (Ver. 1.00)

4.2.5.1.11.2. CAN Main

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

Mit Hilfe dieser Einstellungen lässt sich das CAN Interface konfigurieren.

Interface

Hier können Sie Einstellungen am CAN-Interface vornehmen.

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

FormCfaCANConfiaStd (Ver. 1.00)

Baudrate

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

Address Mode

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

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

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

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

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

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

I/O-Init

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

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

Einstellung speichern

FomCfaCANConfialO (Ver. 1.00)

A/D Mode

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

A/D Filter

Hier kann der A/D Filter eingestellt werden.

D/A Mode

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

Counter Mode

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

Timeout-Mode

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

Output Timeout

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

CNT48 Mode

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

CNT48 Submode

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

CNT48 Filter

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

4.2.5.1.11.3. CAN TX/RX Modi

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

Hier können Sie Einstellungen an den TX-Paketen vornehmen.

TX-Mode[1]

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

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

FormCfCANConfiaTX (Ver. 1.00)

Activate

Aktiviert diesen TX-Mode

Trigger Mode

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

Intervall

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

Address Mode

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

Send to CAN-ID[hex]

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

Data to send

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

Hier können Sie Einstellungen an den RX-Paketen vornehmen.

RX-Mode[1]

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

Activate	<input checked="" type="checkbox"/>
Address Mode	11 Bit Mode
Receive at CAN-ID [hex]	100
Data to send	Digital Out 1-64

FomCfCANConfiaRX (Ver. 1.00)

Activate

Aktiviert diesen RX-Mode

Address Mode

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

Receive at CAN-ID[hex]

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

Data to send

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

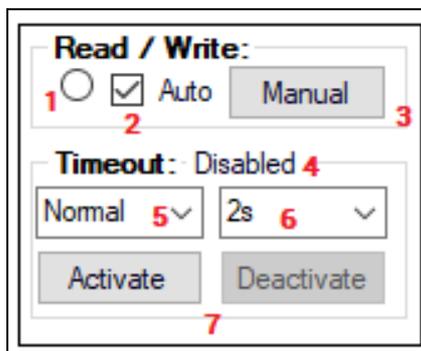
4.2.5.2. I/O-Test

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

4.2.5.2.1. Timeout Test-Funktion

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

Mit diesen Funktionen lässt sich ein Timeout-Fall auslösen.



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

4.2.5.2.2. Digital In

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

Digital In [0 .. 15]

Hier können Sie den Status der digitalen Eingangskanäle ablesen

	State on/off	Counter	FlipFlop		State on/off	Counter	FlipFlop
Kanal 0	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="checkbox"/>	Kanal 8	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="checkbox"/>
Kanal 1	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="checkbox"/>	Kanal 9	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="checkbox"/>
Kanal 2	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="checkbox"/>	Kanal 10	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="checkbox"/>
Kanal 3	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="checkbox"/>	Kanal 11	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="checkbox"/>
Kanal 4	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="checkbox"/>	Kanal 12	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="checkbox"/>
Kanal 5	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="checkbox"/>	Kanal 13	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="checkbox"/>
Kanal 6	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="checkbox"/>	Kanal 14	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="checkbox"/>
Kanal 7	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="checkbox"/>	Kanal 15	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="checkbox"/>

Read with reset

Auto Refresh

FormIODigitalIn (Ver. 1.10)

State on/off

Zeigt den aktuellen Zustand der einzelnen Eingangskanäle.

Counter

Zeigt die Zählerstände der Eingangszähler an.

FlipFlop

Zeigt die Änderung der Eingangszustände seit dem letzten Auslesen an.

Read with reset

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

4.2.5.2.3. Digital Out

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

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

Digital Out [0 .. 15]

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

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

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

Auto Refresh

FormIODigitalOut (Ver. 1.10)

On/Off

Schaltet das jeweilige Ausgangsrelais an oder aus.

Readback

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

Switch all states OFF / Switch all states ON

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

Kanäle schalten mit Timer-Funktion

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

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

Invert DO-Timer

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

4.2.5.2.4. Analog In

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

Analog In [0 .. 15]

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

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

Set mode for all channels

Mode Readback

Set filter for all channels

Filter Readback

Auto Refresh

FormIOAnalogIn (Ver. 1.10)

Value

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

Set mode for all channels

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

Mode Readback

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

Set filter for all channels

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

Filter Readback

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

4.2.5.2.5. Analog Out

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

Analog Out [0 .. 13]

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

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

Set mode for all channels
 ▾

Auto Refresh

FormIOAnalogOut (Ver. 1.10)

Value

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

Mode

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

Readback

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

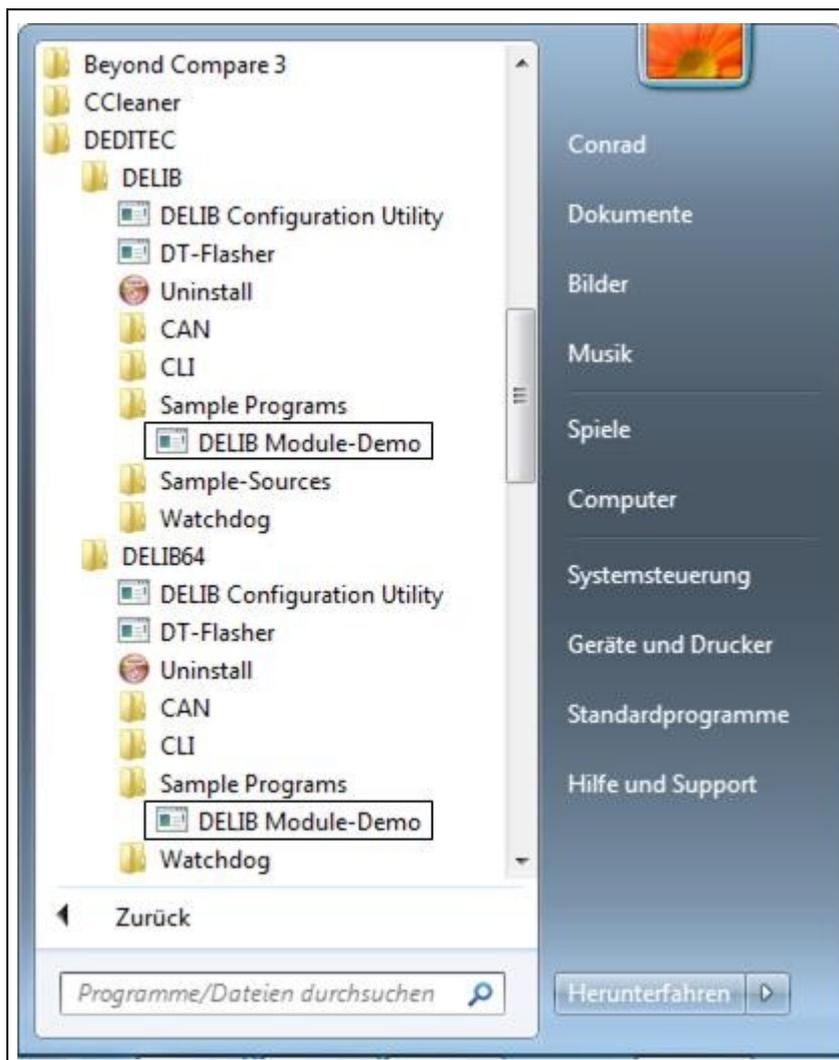
Set mode for all channels

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

4.2.6. DELIB Module Demo

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

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

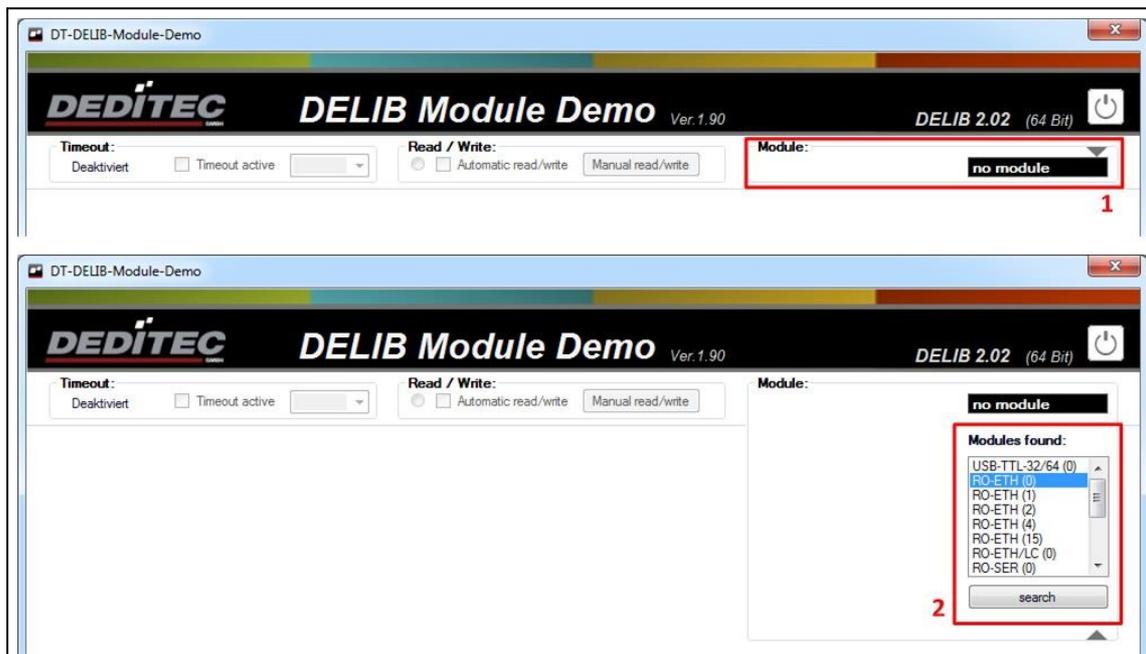


Das Programm DELIB Module Demo ist ein All-in-One Tool mit dem sämtliche I/Os aller Produkte aus unserem S&R Bereich gesteuert und getestet werden können.

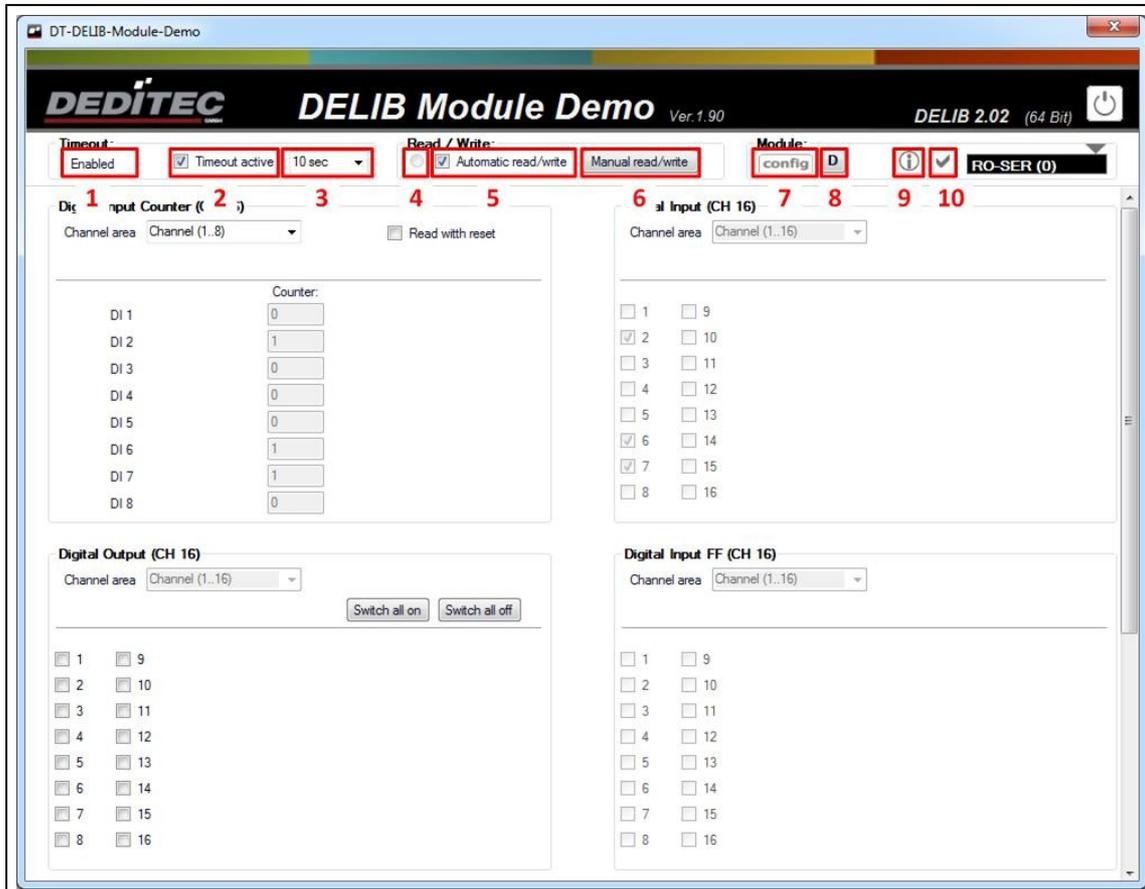
4.2.6.1. Auswahl des Moduls

Bei Programmstart muss ein Modul ausgewählt werden.

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



4.2.6.2. Allgemein



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

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

4.2.6.2.1. Module Info

DT_ModuleInfo

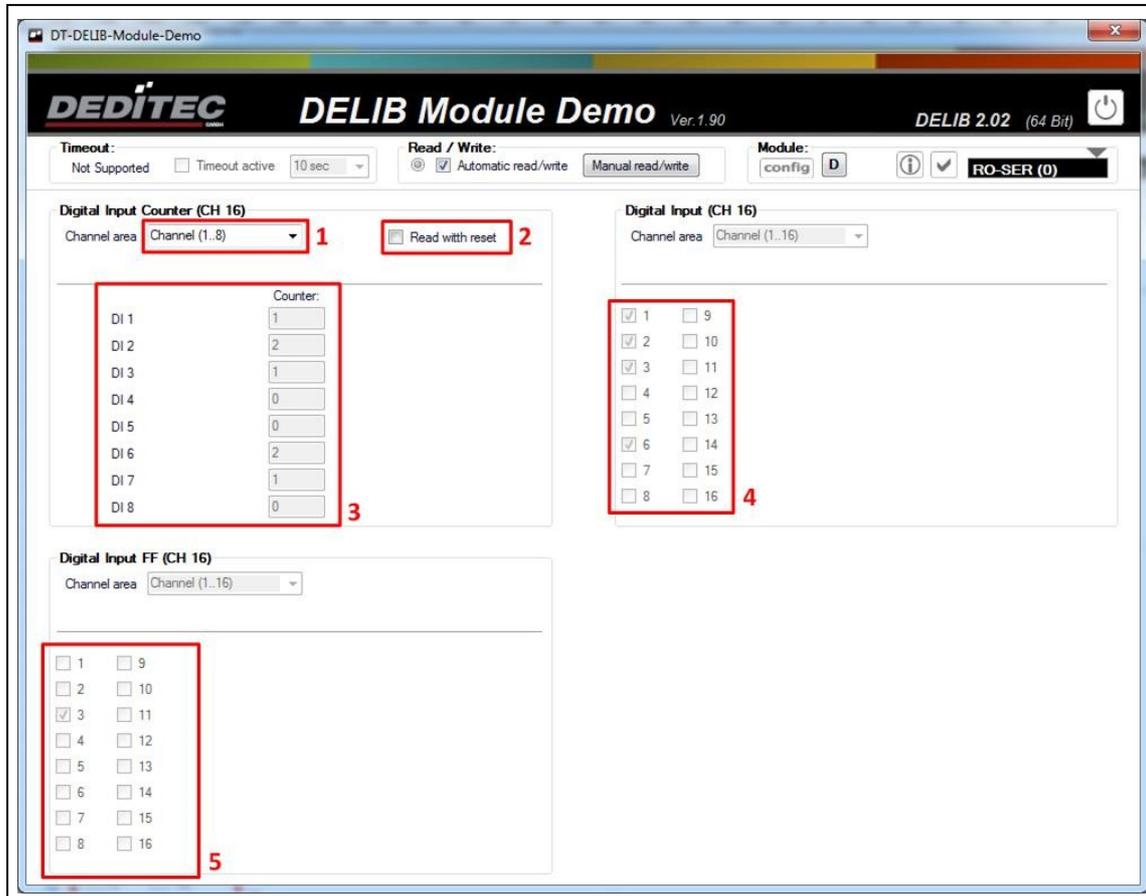
General	Digital I/O	Analog I/O	Special
SW_FEATURE_1 e300000f	Digital Inputs 16	Analog Inputs 0	Stepper 0
HW_INTERFACE_1 01000031	Digital Outputs 16	Analog Outputs 0	
Firmware-Revision 2.01	Digital In-/Outputs 0	Temperature Inputs 0	
Main-Module: RO	Digital Input FlipFlops 16		
	Digital Input Counter 16		
	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 -	Watchdog Commands -
AD FIFO OK	DI CNT Commands OK	AD Commands -	Stepper Commands -
Set-Clr Bit Commands OK	DI CNT Latch Feature -	Pt100 Commands -	
EEPROM Commands OK	DI FF Commands OK		
EEPROM E2_2K -	DO Commands OK		
DX1 Mode -	DO Time Commands -		
Support Channel Names -	PWM Commands -		
HW-INT Supported by FV OK	TTL Commands -		
ETH -	PulseGen Commands -		
CAN -	CNT8 Commands -		
RS232 -	Auto-Off Timeout Commar OK		
RS485 OK			
USB OK			

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

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

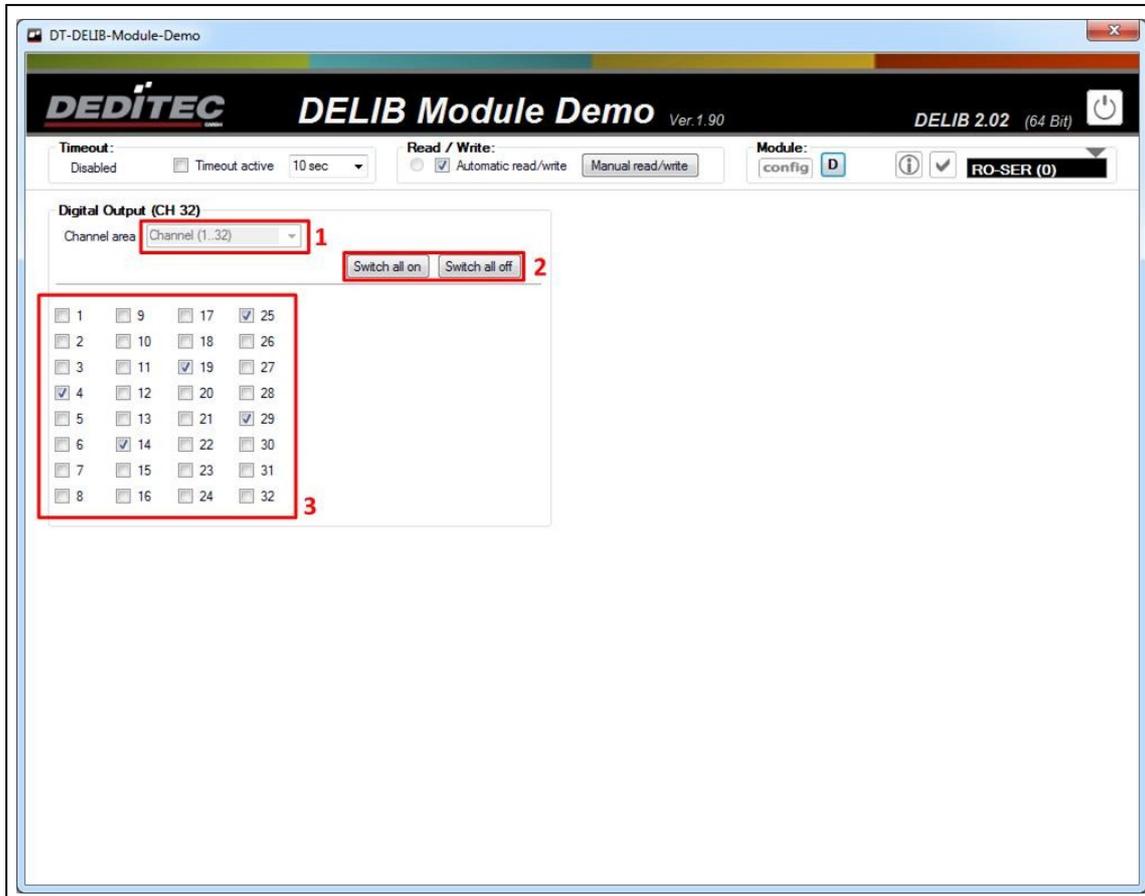
4.2.6.3. Digital Input



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

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

4.2.6.4. Digital Output



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

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

4.2.6.5. Analog Input

The screenshot shows the 'DELIB Module Demo' software interface. At the top, it displays 'Ver. 1.90' and 'DELIB 2.02 (64 Bit)'. The interface is divided into two main sections: 'Analog Output (CH 8)' and 'Analog Input (CH 32)'. The 'Analog Output' section includes a 'Channel area' dropdown set to 'Channel (1..8)', a 'Range' dropdown set to '0 - 5V', and buttons for 'EEPROM save', 'EEPROM load', 'Default load', and 'Disable Outputs'. Below this is a table of output channels (DA 1 to DA 8) with 'Value' and 'Readback' columns. The 'Analog Input' section includes a 'Channel area' dropdown set to 'Channel (1..16)', a 'Range' dropdown set to '+10V', and a 'Module' dropdown set to 'RO-SER (0)'. Below this is a table of input channels (AD 1 to AD 16) with 'Value' columns. Red boxes highlight the 'Channel (1..16)' dropdown, the '+10V' dropdown, and the value '-2.751 V' in the AD 2 row. A red number '3' is placed next to the AD 2 value.

Channel	Value	Readback
DA 1	4,000	4,000 V
DA 2	4,000	4,000 V
DA 3	4,500	4,500 V
DA 4	4,000	4,000 V
DA 5	2,500	2,500 V
DA 6	2,500	2,500 V
DA 7	2,500	2,500 V
DA 8	2,500	2,500 V

Channel	Value	Channel	Value
AD 1	-2,735 V	AD 9	-2,735 V
AD 2	-2,751 V	AD 10	-2,742 V
AD 3	-2,732 V	AD 11	-2,736 V
AD 4	-2,741 V	AD 12	-2,741 V
AD 5	-2,738 V	AD 13	-2,749 V
AD 6	-2,744 V	AD 14	-2,752 V
AD 7	-2,741 V	AD 15	-2,746 V
AD 8	-2,747 V	AD 16	-2,745 V

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.

4.2.6.6. Analog Output

The screenshot shows the 'DELIB Module Demo' software interface. The title bar indicates 'DT-DELIB-Module-Demo'. The main header features the 'DEDITEC' logo, 'DELIB Module Demo Ver. 1.90', and 'DELIB 2.02 (64 Bit)'. Below the header, there are control panels for 'Timeout' (Disabled, Timeout active, 10 sec), 'Read / Write' (Automatic read/write, Manual read/write), and 'Module' (config, D, RO-SER (0)).

The 'Analog Output (CH 8)' section is highlighted with red boxes and numbers 1 through 8. It includes a 'Channel area' dropdown set to 'Channel (1..8)' (1), a 'Range' dropdown set to '0 - 5V' (2), and four buttons: 'EEPROM save' (3), 'EEPROM load' (4), 'Default load', and 'Disable Outputs' (6). Below these are eight rows of data for channels DA 1 to DA 8. Each row has two columns for 'Value' and 'Unit' (V). The values for DA 5 are 2.500 (7) and 2.500 (8), both highlighted with red boxes.

The 'Analog Input (CH 32)' section shows a 'Channel area' dropdown set to 'Channel (1..16)' and a 'Range' dropdown set to '+10V'. It displays a table of input values for channels AD 1 to AD 16, with values ranging from -2.735 V to -2.745 V.

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

1. Auswahl des Kanal-Bereichs, der angezeigt werden soll.
2. Auswahl des Spannungs-/Strombereichs in dem die Werte ausgegeben werden sollen. Wird der Modus nicht unterstützt, erscheint die Meldung "illegal" rechts neben dem Drop-Down Menü.
3. Die aktuell ausgegebenen Spannungen/Ströme werden im Modul gespeichert. Bei Modul-Start werden diese direkt ausgegeben.
4. Lädt die Spannungen/Ströme die im Modul gespeichert sind.

5. Setzt die Spannungen/Ströme, die bei Modul-Start ausgegeben werden, auf Werkseinstellung zurück.
6. Deaktiviert die Ausgänge. Es werden keine Spannungen/Ströme ausgegeben.
7. An D/A-Kanal 5 sollen 2,5V ausgegeben werden(Soll-Wert).
8. Liest D/A-Kanal 5 zurück (Ist-Wert).

4.2.7. 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:

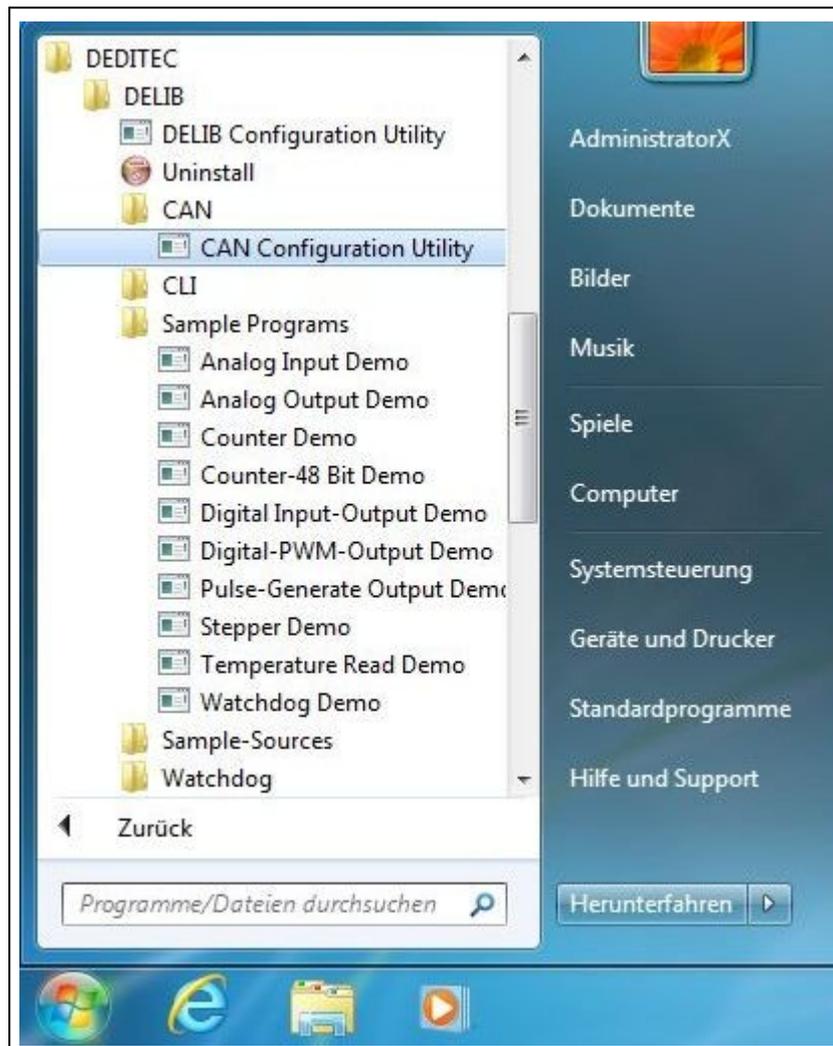
- Baudrate
- Adress-Modus
- Modul-ID
- Response-ID
- Modi, mit denen Submodule gestartet werden
- Automatischer Sendemodus (TX-Modus)
- Automatischer Empfangsmodus (RX-Modus)

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

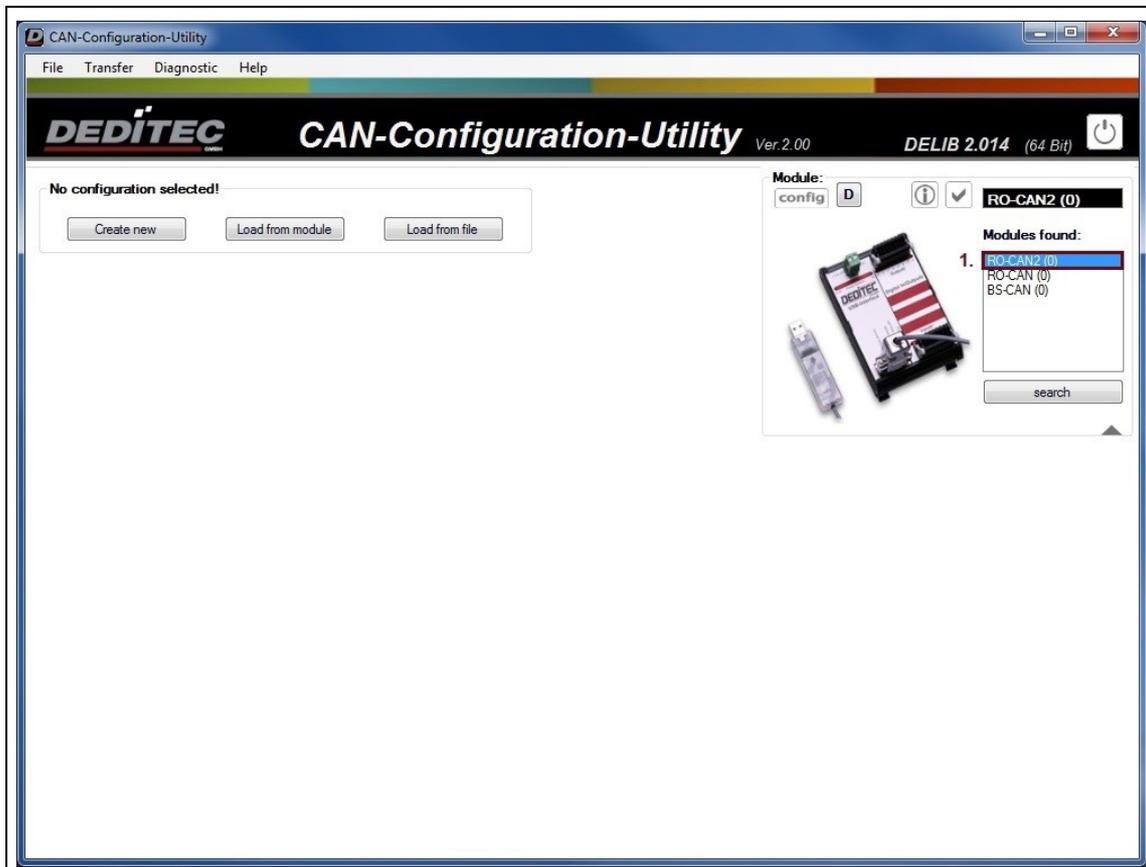
4.2.7.1. Auswahl des Moduls

Starten Sie das CAN-Configuration-Utility über:

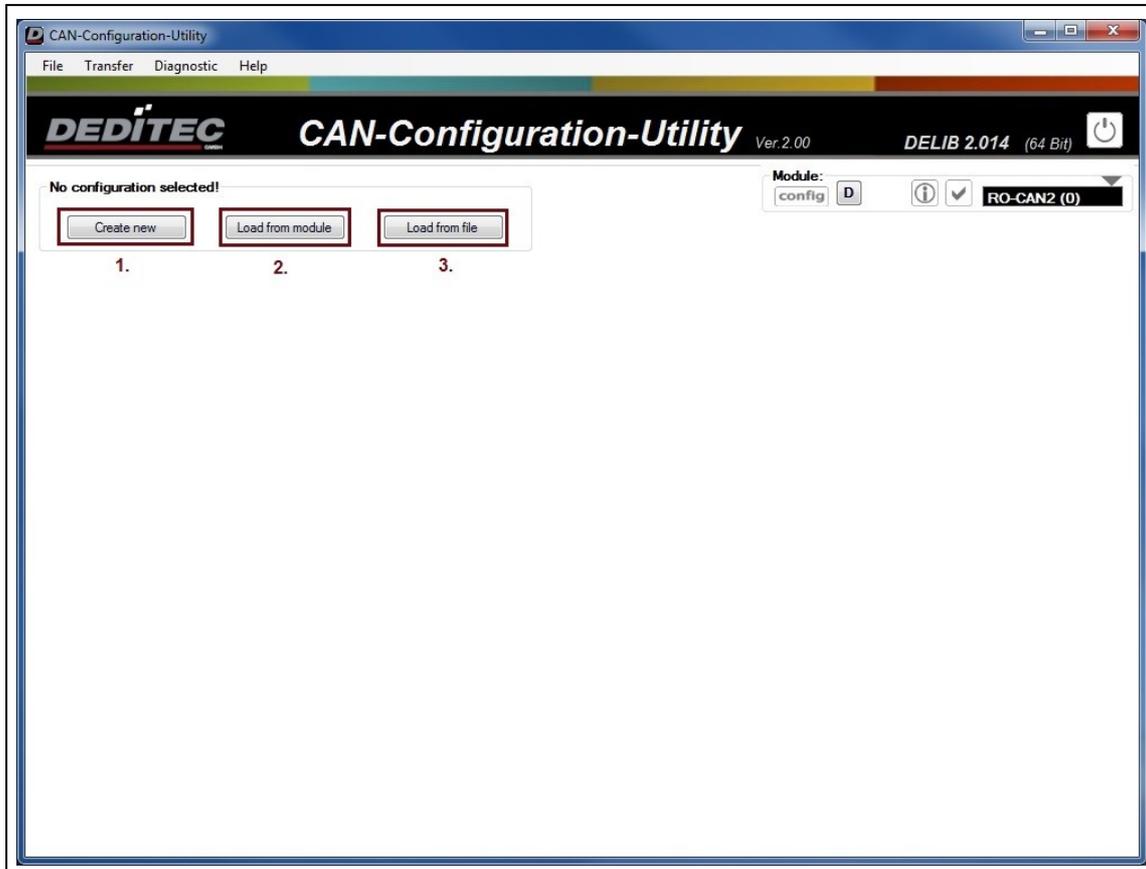
Start → Programme → DEDITEC → DELIB → CAN



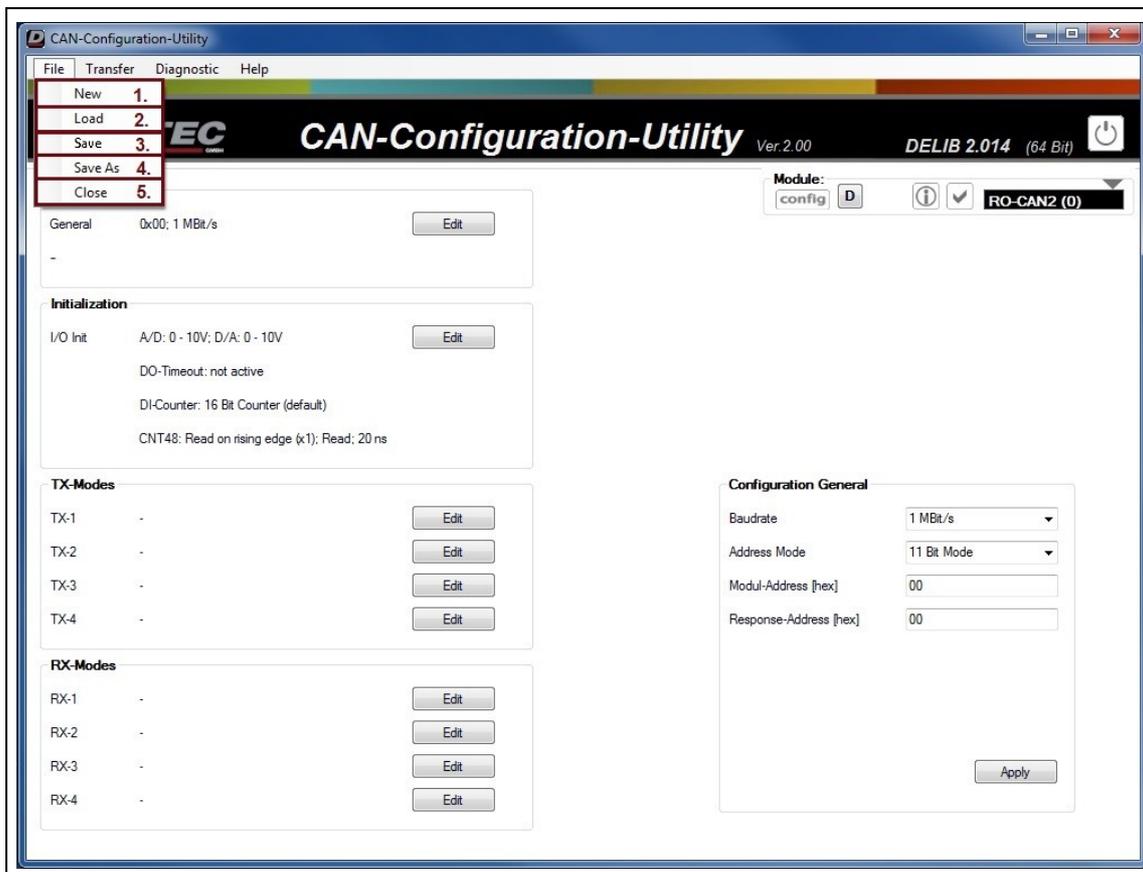
1. In der "Module-Selection" ein entsprechendes CAN-Modul (z.B. das RO-CAN2) auswählen.



4.2.7.2. Neue Konfiguration Erstellen, Laden, Speichern



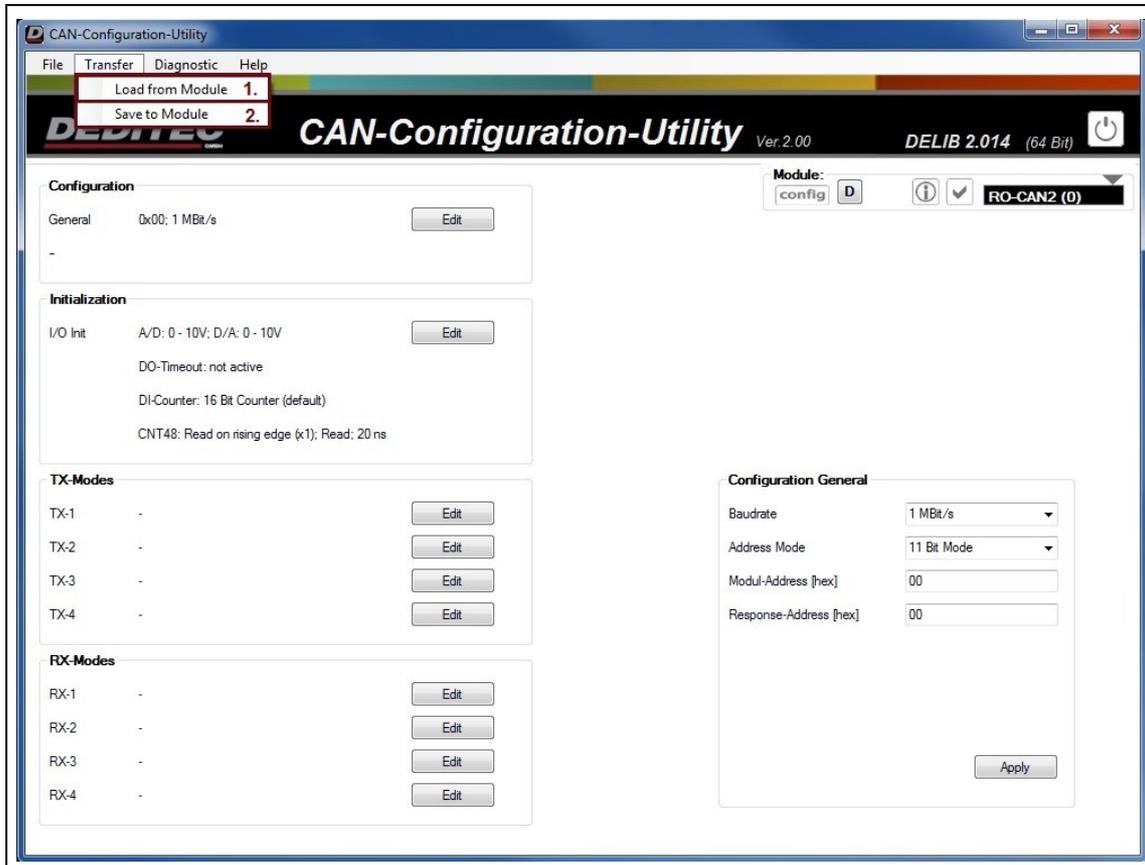
1. Über "Create new" kann eine neue CAN-Konfiguration erstellt werden.
2. Mit "Load from Module" kann die aktuell auf dem Modul vorhandene Konfiguration ausgelesen werden.
3. "Load from File" erlaubt es, zuvor auf einem Datenträger gespeicherte CAN-Konfigurationsdateien zu öffnen.



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

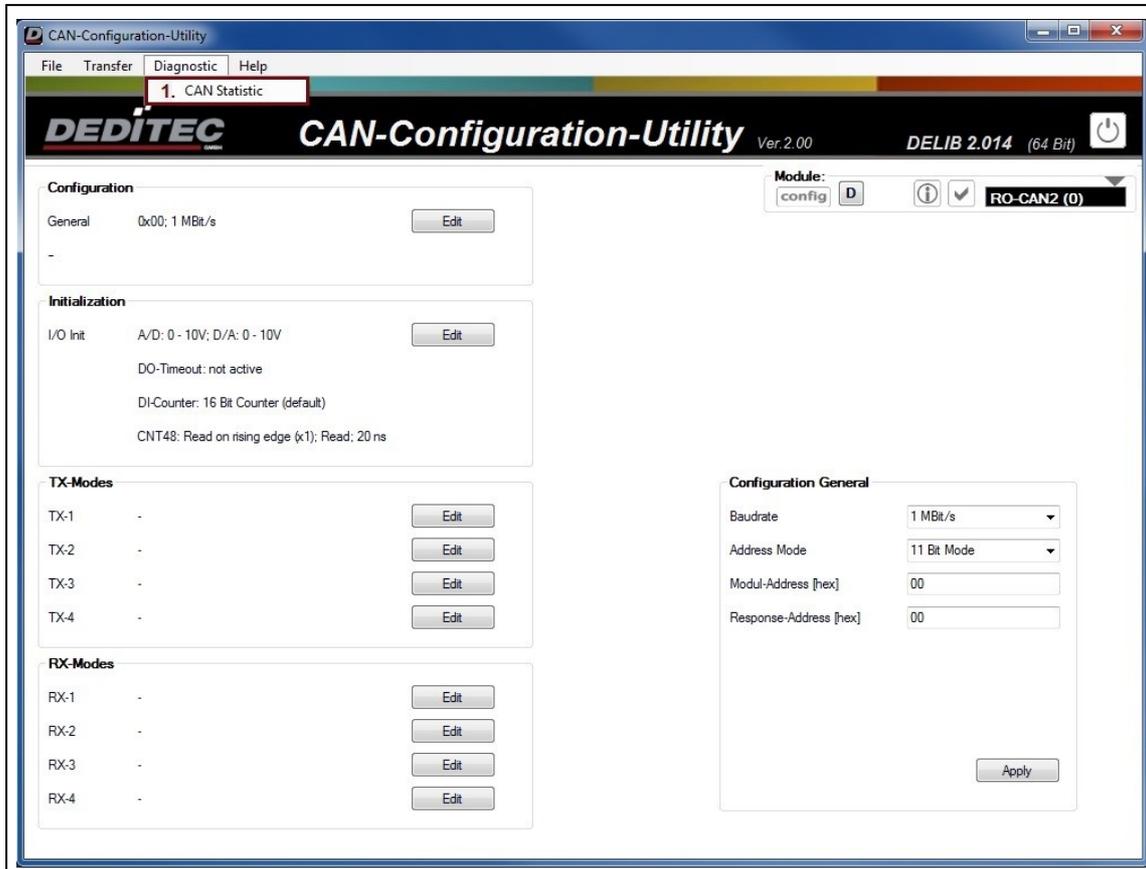
1. Über den Menüpunkt "New" kann eine neue CAN-Konfiguration erstellt werden.
2. "Load" bietet die Möglichkeit, zuvor gespeicherte CAN-Konfigurationsdateien zu öffnen.
3. Wurde eine CAN-Konfigurationsdatei geöffnet, so kann über "Save" die Datei überschrieben und aktuelle Änderungen gespeichert werden.
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.

4.2.7.3. Konfiguration auf das Modul übertragen



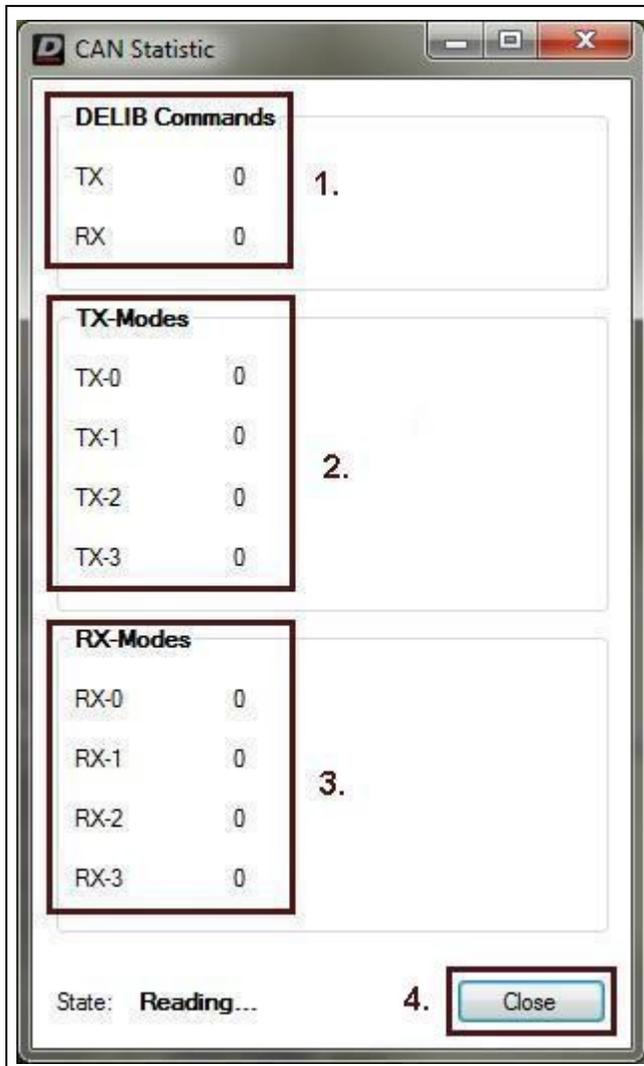
1. Über "Load from Module" kann die aktuelle Konfiguration des ausgewählten Moduls ausgelesen werden.
2. "Save to Module" überträgt die aktuelle CAN-Konfiguration auf das ausgewählte Modul.

4.2.7.4. 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.

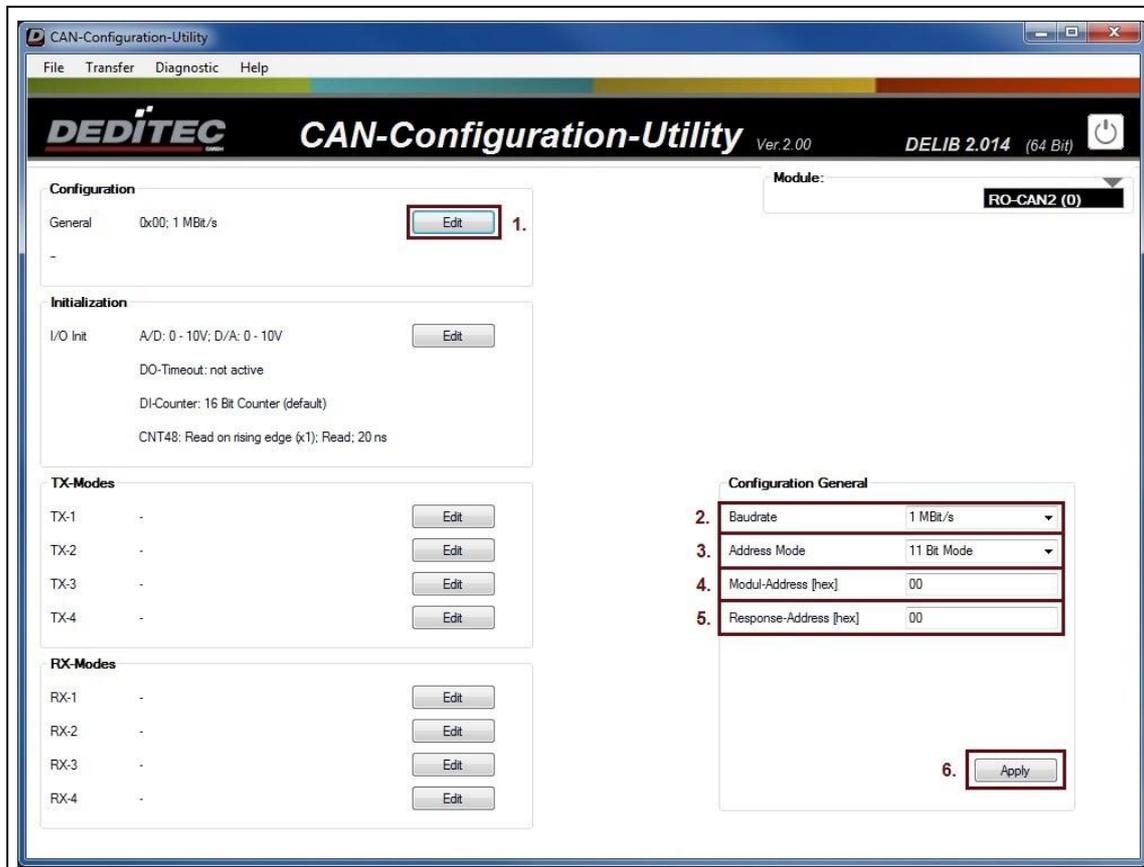
4.2.7.5. Konfiguration

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.

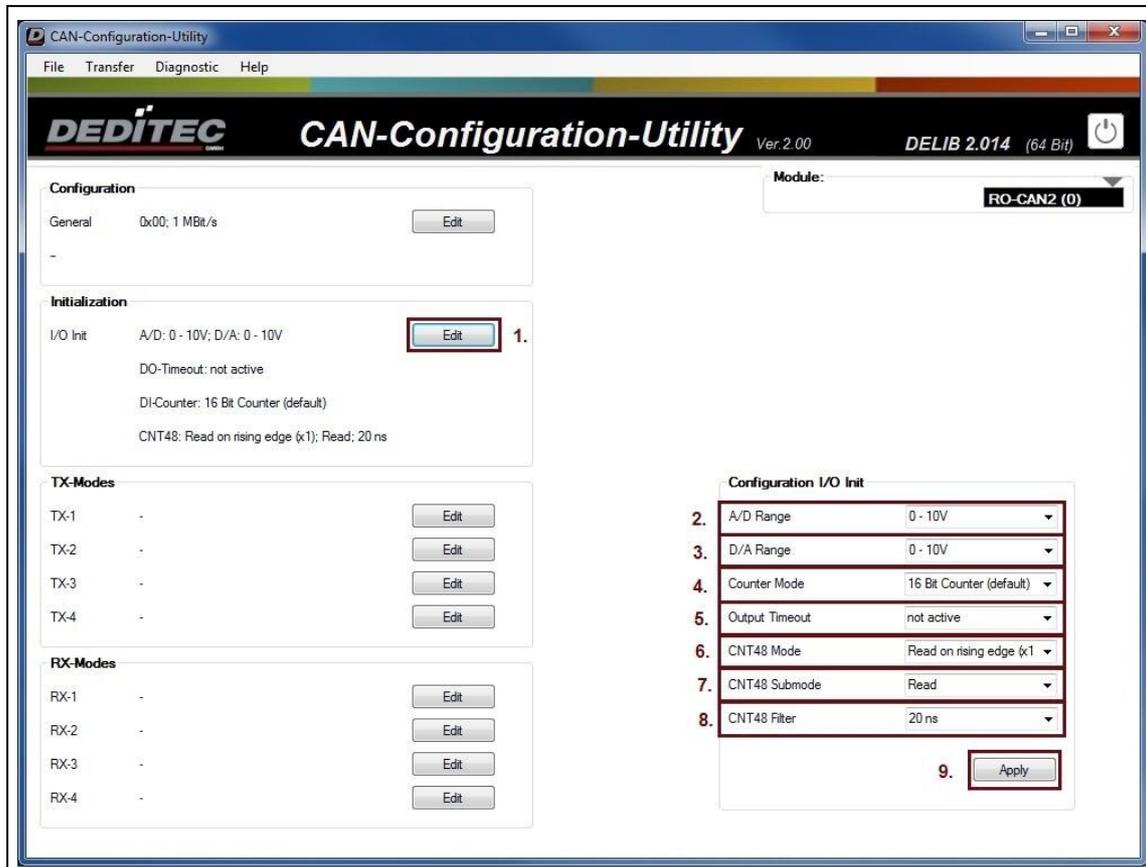
Auf den nachfolgenden Seiten wird gezeigt, wie die verschiedenen Modi konfiguriert werden.

4.2.7.5.1. 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.

4.2.7.5.2. I/O Konfiguration



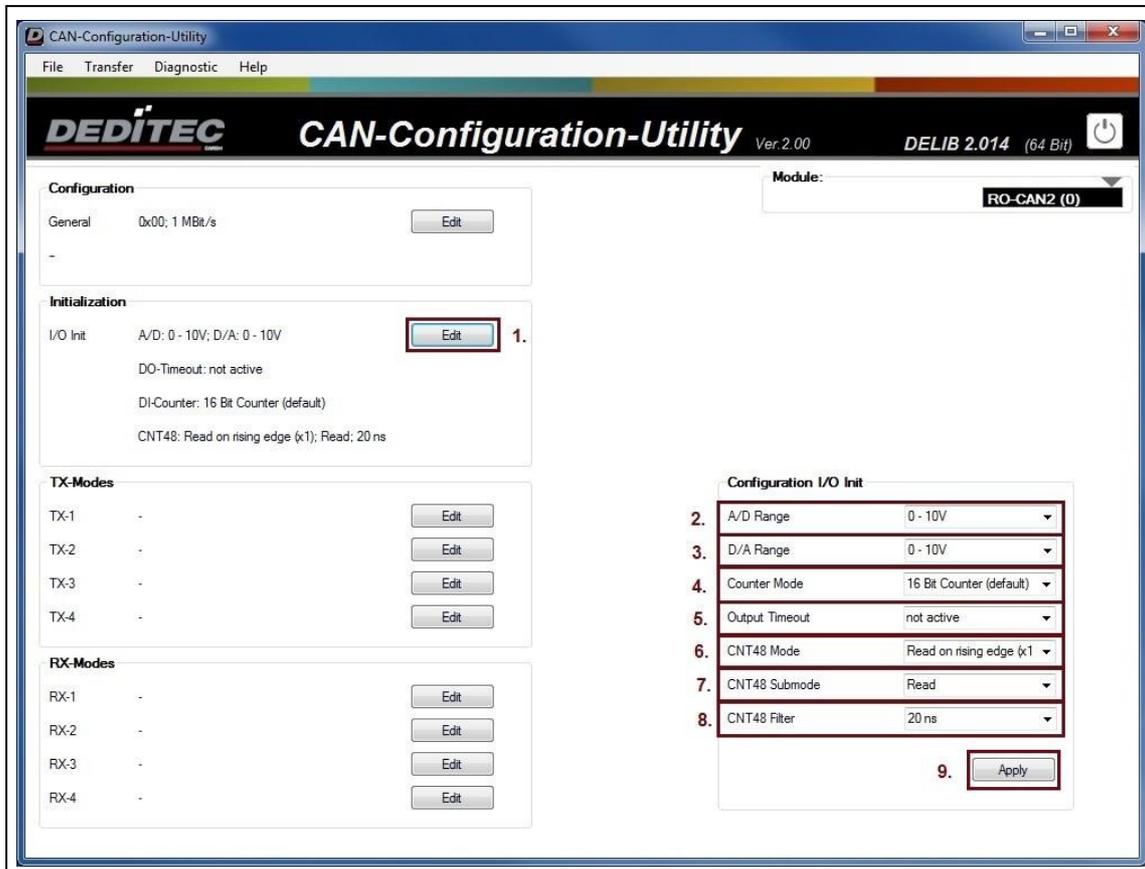
Diese Einstellungen dienen der Konfiguration der angeschlossenen Submodule. Es kann der jeweilige Filter / Modus eingestellt werden, in dem die angeschlossenen Submodule gestartet werden.

Hinweis:

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

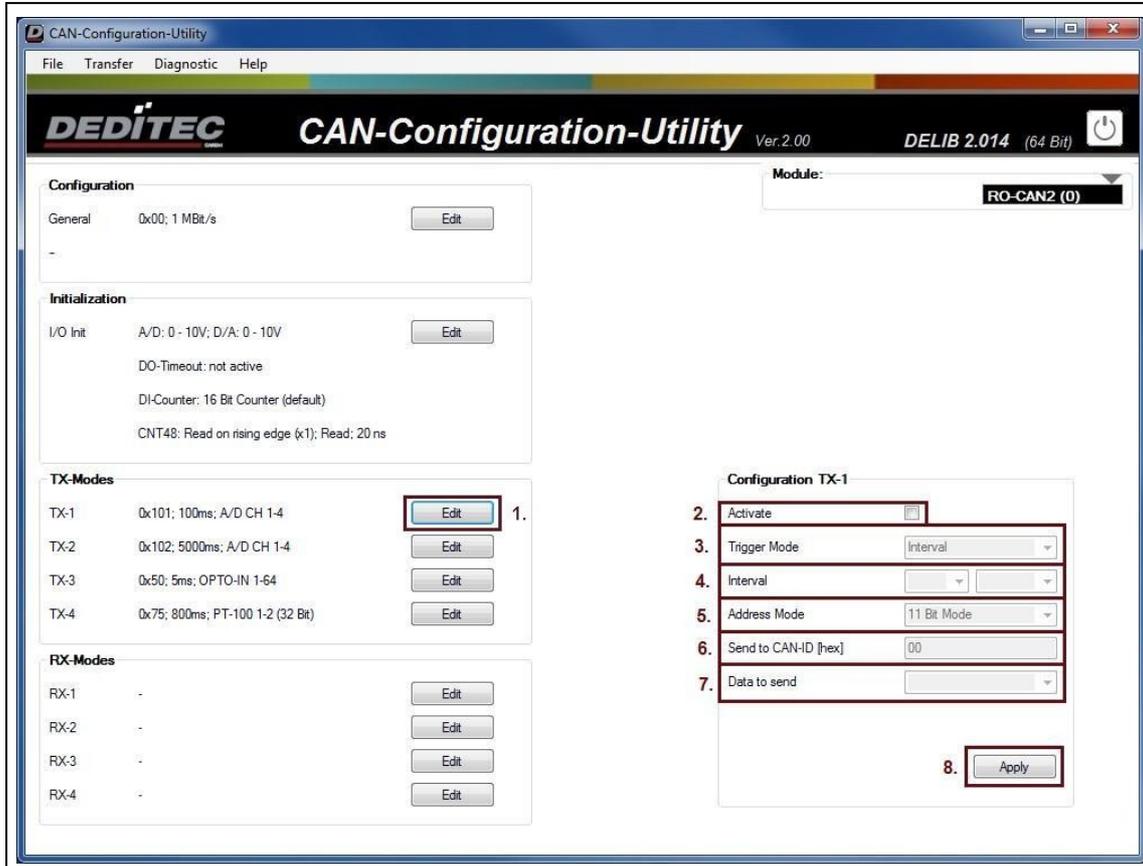
1. Über den Menüpunkt "Edit" wird ein Zusatzfenster geöffnet, in dem die Konfiguration vorgenommen werden kann.
2. Der Wertebereich gibt den Bereich an, in dem analoge Signale, digital (z.B. im Bereich 0-10V) umgesetzt werden.
3. Der Wertebereich gibt den Bereich an, in dem digitale Signale, analog (z.B. 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.

4.2.7.5.3. 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.

4.2.7.5.3.1. 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.

4.2.7.5.3.2. 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

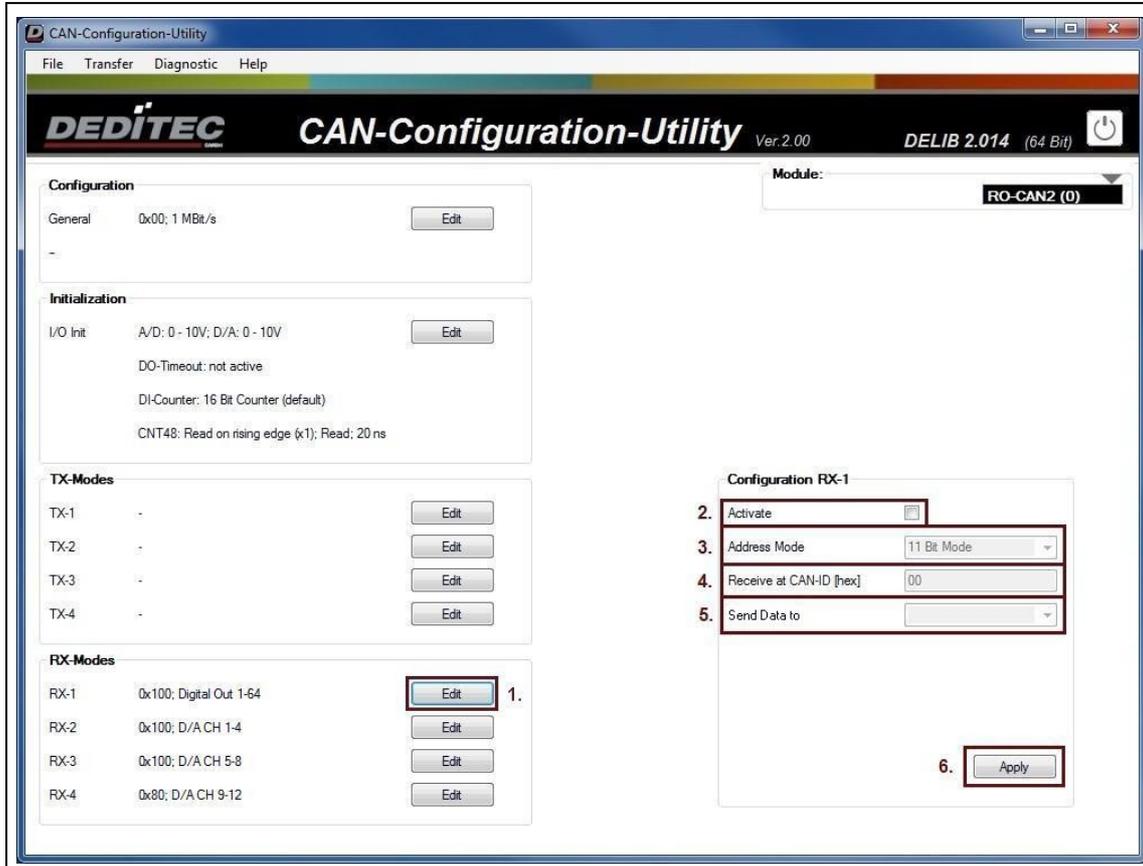
Activate	<input checked="" type="checkbox"/>
Trigger Mode	RX-Event
Interval	
Address Mode	11 Bit Mode
Send to CAN-ID [hex]	100
Data to send	A/D CH 1-4

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).

4.2.7.5.4. 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.

4.2.7.5.4.1. Beispiel RX-DA

Configuration RX-1

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

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.

4.2.7.5.4.2. Beispiel RX-DO

Configuration RX-1

Activate

Address Mode

Receive at CAN-ID [hex]

Data to send

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.

4.2.7.6. 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.

4.2.7.6.1. 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)

4.2.7.6.2. 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)

4.2.7.6.3. 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)

4.2.7.6.4. 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.

4.2.7.6.5. 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

4.2.7.6.6. Analoge Ein- / Ausgänge

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

4.2.7.6.6.2. 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.

4.2.7.6.6.3. Beispiele

Beispiel Spannungsbereich $\pm 10V$

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 = $(0x4711 * (10 * 2) / 0xFFFF) - 10 = -4,45 V$

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.

4.2.7.6.7. 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 (0 [dez] = illegaler Wert / Sensor nicht verbunden, 1 [dez] = Faktor 10, 2 [dez] = Faktor 100)
	2-7	nicht benutzt
4	0	Status PT100 Sensor Kanal 1 (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 (0 [dez] = illegaler Wert / Sensor nicht verbunden, 1 [dez] = Faktor 10, 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

4.2.7.6.8. 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)

4.2.7.6.8.1. Command-Liste

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

Kommando DAPI_STEPPEER_CMD_	mit	Wert (hex)	Bedeutung
SET_FREQUENCY		6 (hex)	Einstellung der Motorsollfrequenz
SET_FREQUENCY_DIRECTLY		7 (hex)	Einstellung der Motorfrequenz
GET_FREQUENCY		8 (hex)	Abfrage der Motorfrequenz
FULLSTOP		9 (hex)	Sofortiges Anhalten des Motors
STOP		10 (hex)	Anhalten des Motors (Bremsrampe wird eingehalten)
GO_REFSWITCH		11 (hex)	Anfahren einer Referenzposition
DISABLE		14 (hex)	Ein-/Ausschalten des Motors
MOTORCHARACTERISTIC_LOAD_DEFAULT		15 (hex)	Setzen der Motorcharakteristik auf Defaultwert
MOTORCHARACTERISTIC_EEPROM_SAVE		16 (hex)	Speichern der Motorcharakteristik im EEPROM
MOTORCHARACTERISTIC_EEPROM_LOAD		17 (hex)	Laden der Motorcharakteristik aus dem EEPROM

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

4.2.7.6.8.2. Werte für par 1 zu Befehl SET_MOTORCHARACTERISTIC

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

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

4.2.7.6.8.3. Werte für par 1 zu Befehl GO_REFSWITCH

Parameter (DAPI_STEPPER_GO_REFSW ITCH_PAR_ ...)	Wert (dez)	Bezeichnung
REF1	1	Anfahren des Referenzschalter 1
REF2	2	Anfahren des Referenzschalter 2
REF_LEFT	4	Anfahren der linken Kante des Referenzschalters
REF_RIGHT	8	Anfahren der rechten Kante des Referenzschalters
REF_GO_POSITIVE	16	Start des Motors nach rechts
REF_GO_NEGATIVE	32	Start des Motors nach links
SET_POS_0	64	Nullen der Motorposition

4.2.7.6.8.4. Beispiel

Programmbeispiel

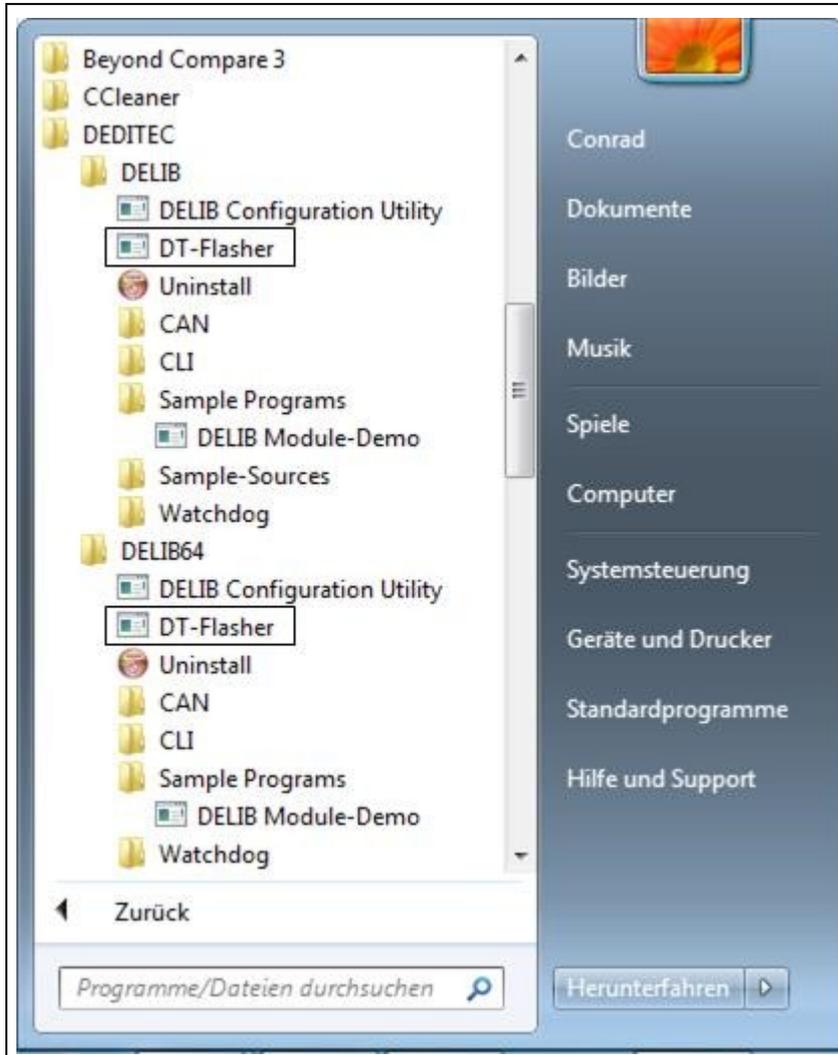
```
DapiStepperCommand(handle, 1,  
DAPI_STEPPER_CMD_GO_POSITION, 3200, 0, 0, 0);
```

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

4.2.8. 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.

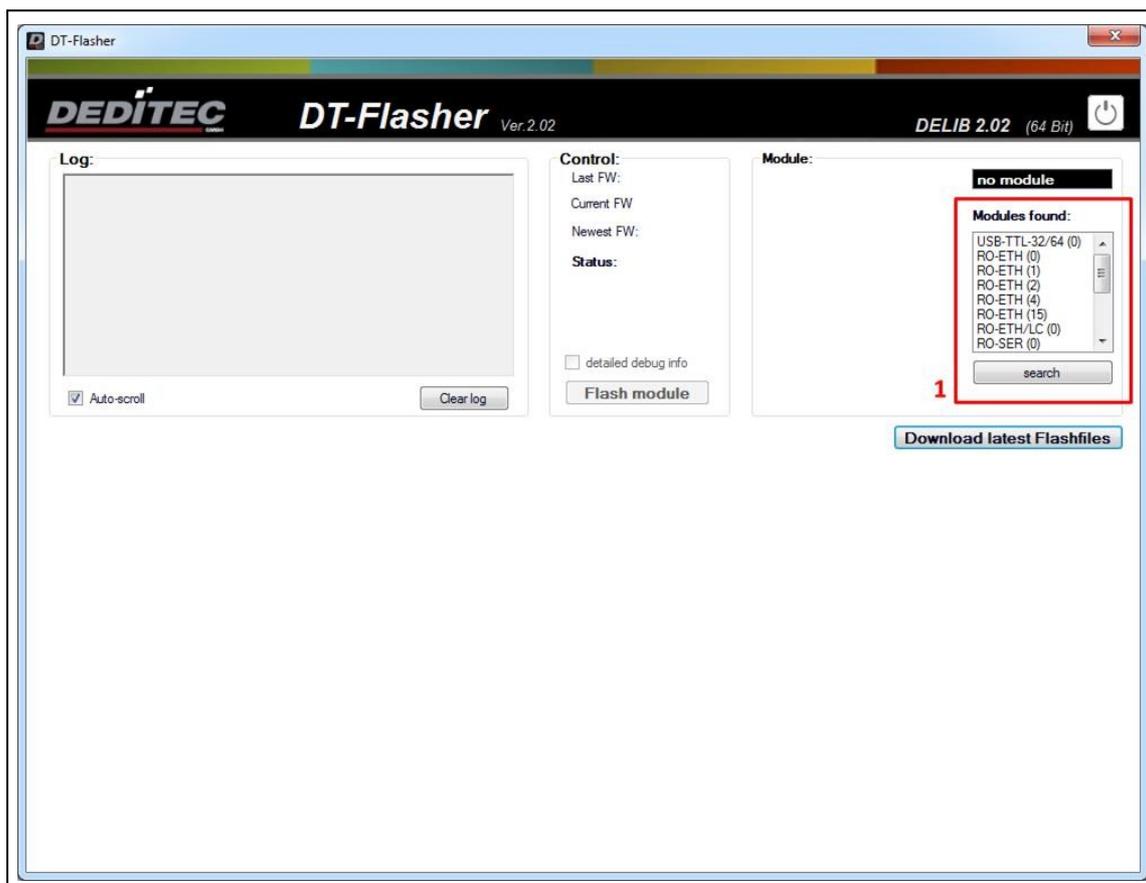
4.2.8.1. Über DEDITEC-Firmware

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:

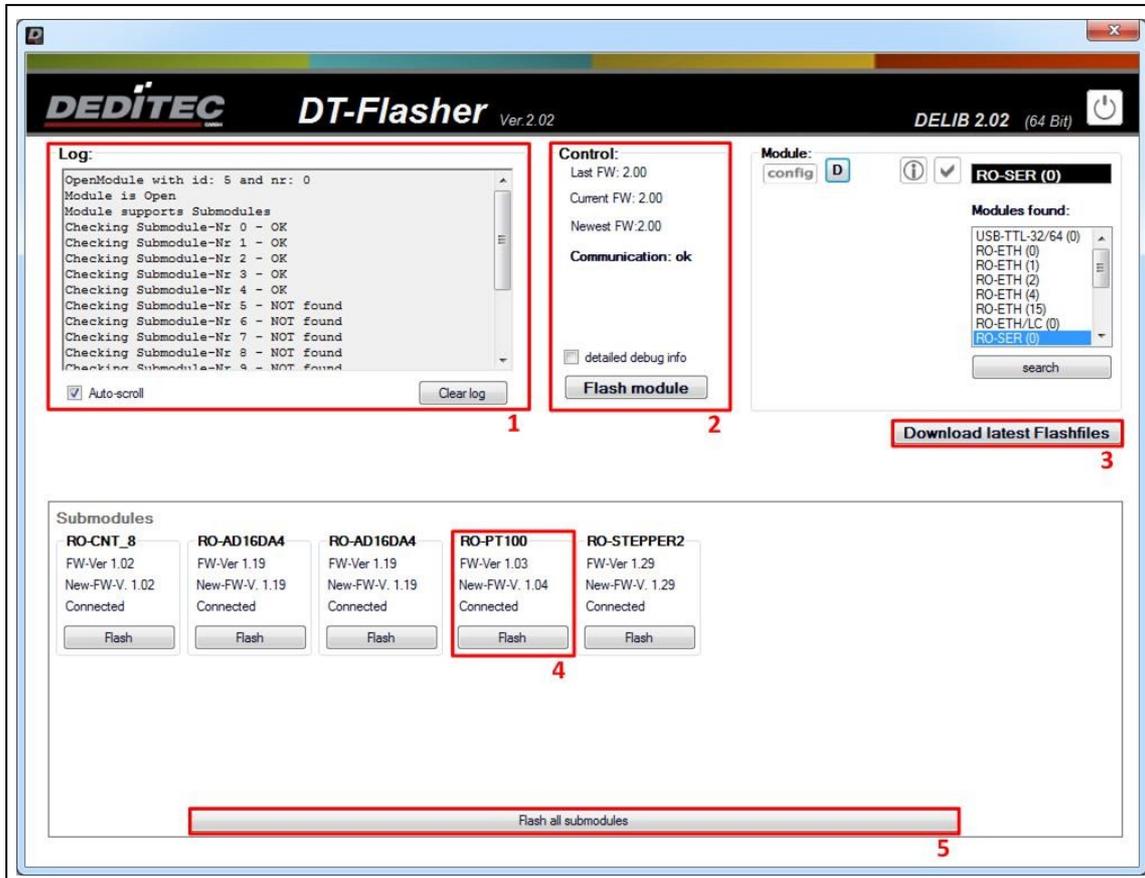
Da neue Firmware-Versionen in der Regel neue Funktionen für Ihr Produkt "freischalten", empfehlen wir daher ein regelmäßiges Firmware-Update Ihrer DEDITEC Produkte.

4.2.8.2. 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"

4.2.8.3. 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.

4.2.8.3.1. 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

Schritt 2

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

x86

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

x64

C:\Program Files\DEDITEC\DELIB\programs

4.3. DELIB Sample Sources (Windows Programmbeispiele)

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

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

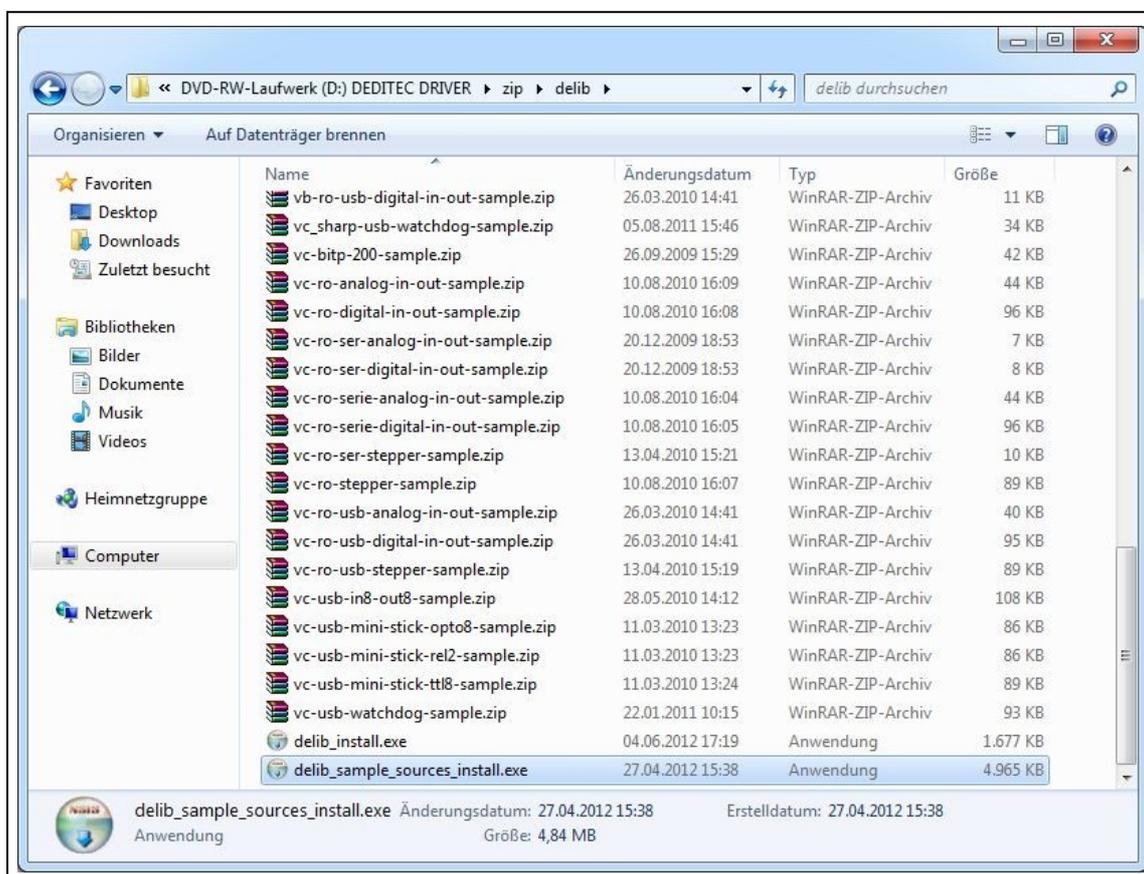
- C
- C++
- C#
- Delphi
- VisualBasic
- VB.NET
- MS-Office
- LabVIEW
- Java

4.3.1. Installation DELIB Sample Sources

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

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

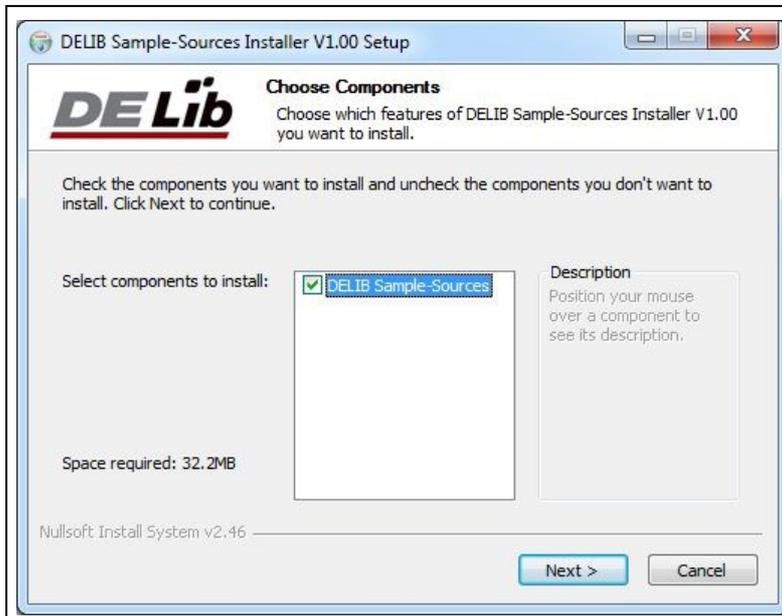
Eine aktuelle Version der Sample Sources finden Sie auch im Internet unter <http://www.deditec.de/de/delib>



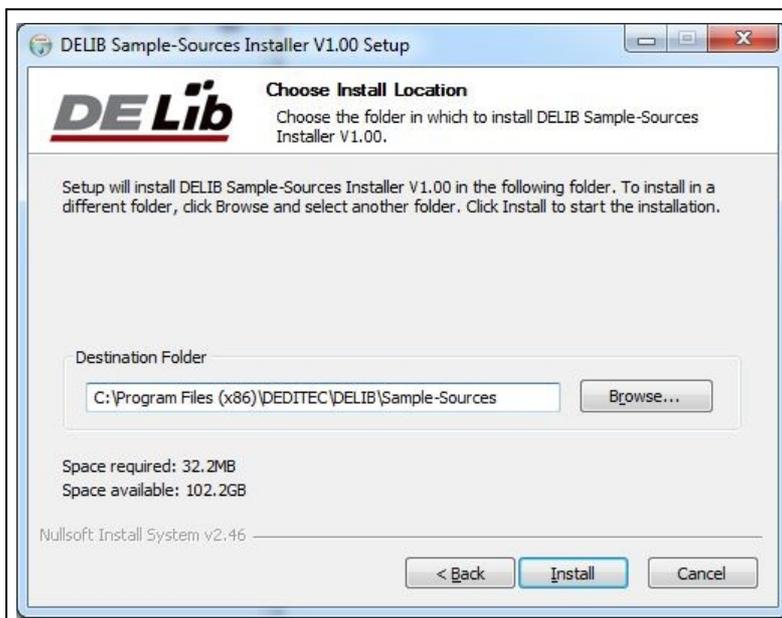
Startbild des DELIB Sample Sources Installer



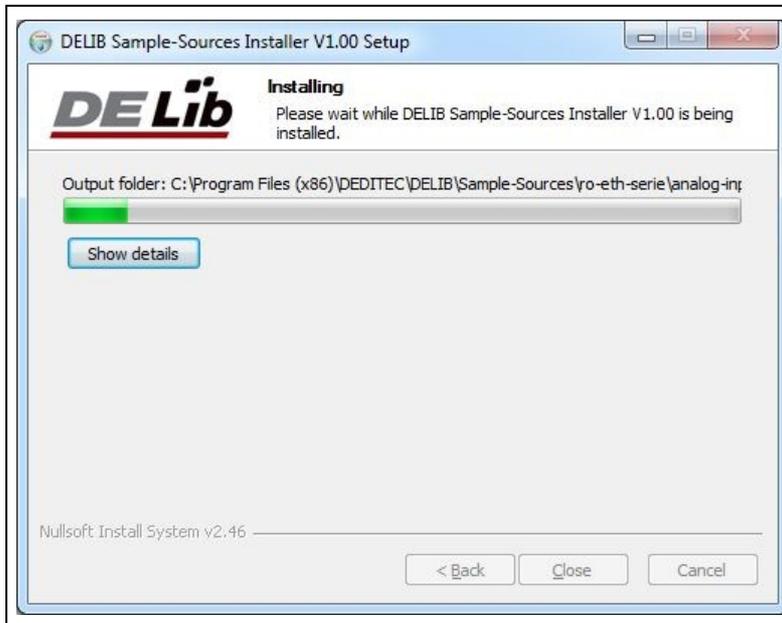
Drücken Sie Next.



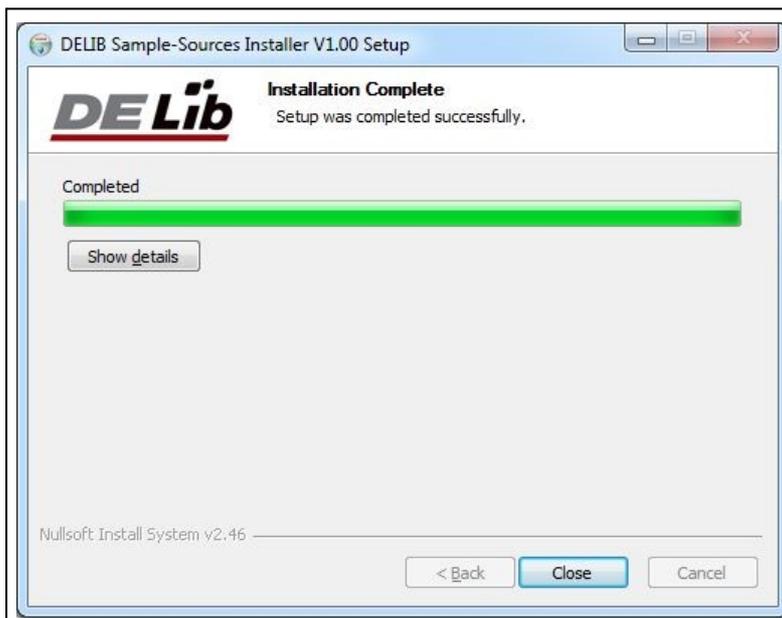
Wählen Sie den Installationsordner und drücken Sie Install.



Die DELIB Sample Sources werden nun installiert.



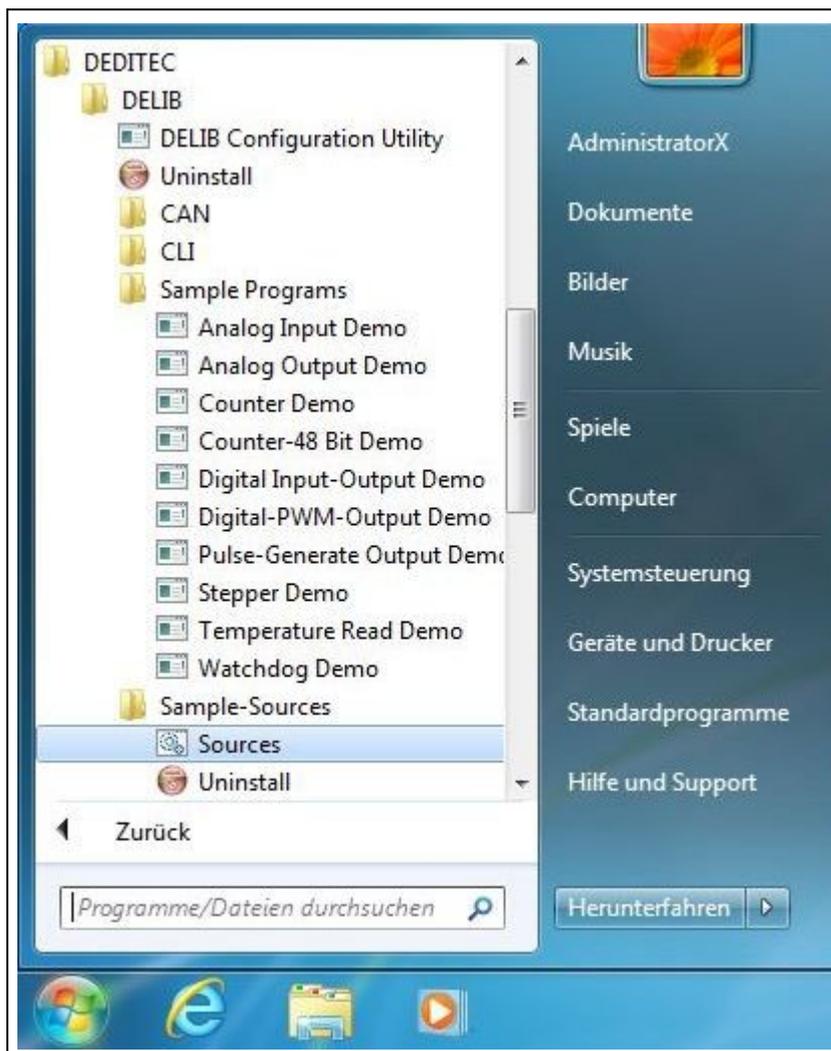
Die DELIB Sample Sources wurden erfolgreich installiert. Drücken Sie Close um die Installation zu beenden.



4.3.2. Benutzung der DELIB Sample Sources

Nach Installation der DELIB Sample Sources finden Sie diese unter

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



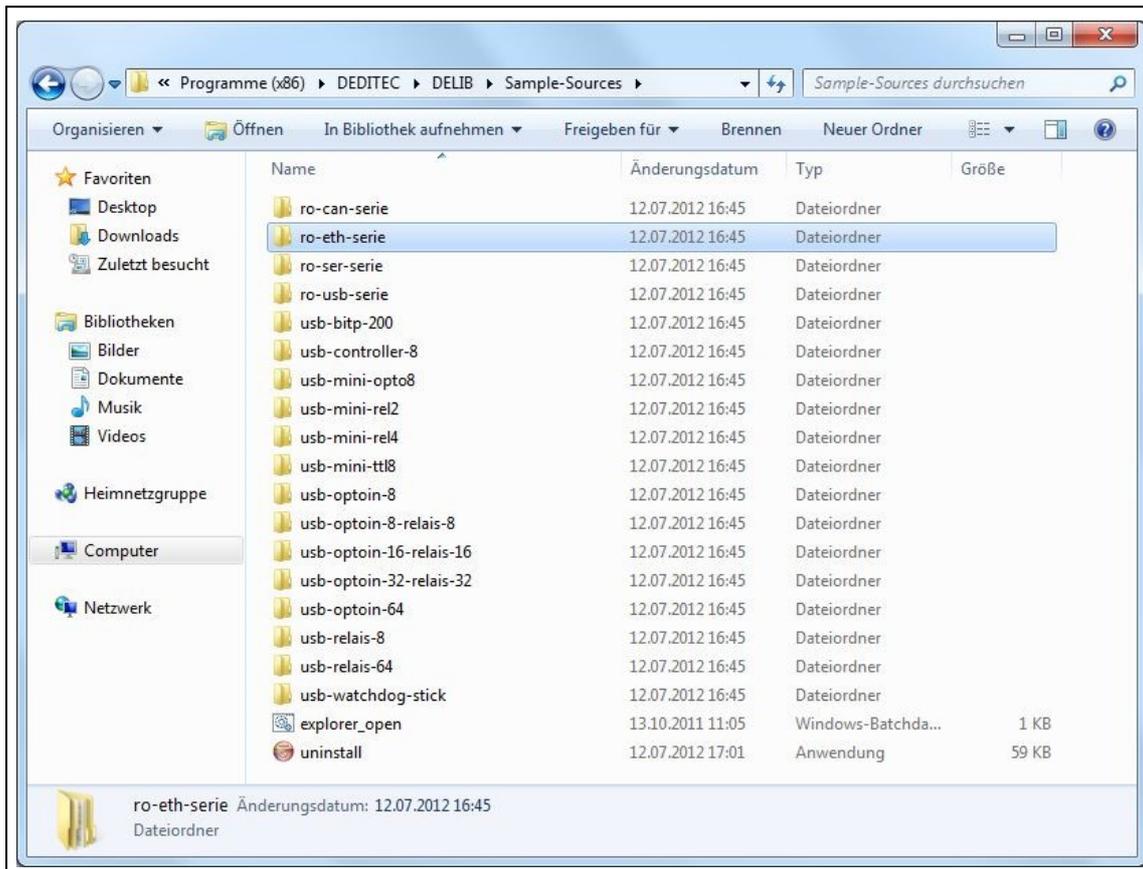
Nun öffnet sich der Windows-Explorer mit einer Übersicht aller Produkte für die ein Beispielprogramm verfügbar ist.

4.3.2.1. Schritt 1 - Produktauswahl

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

Visual-C.

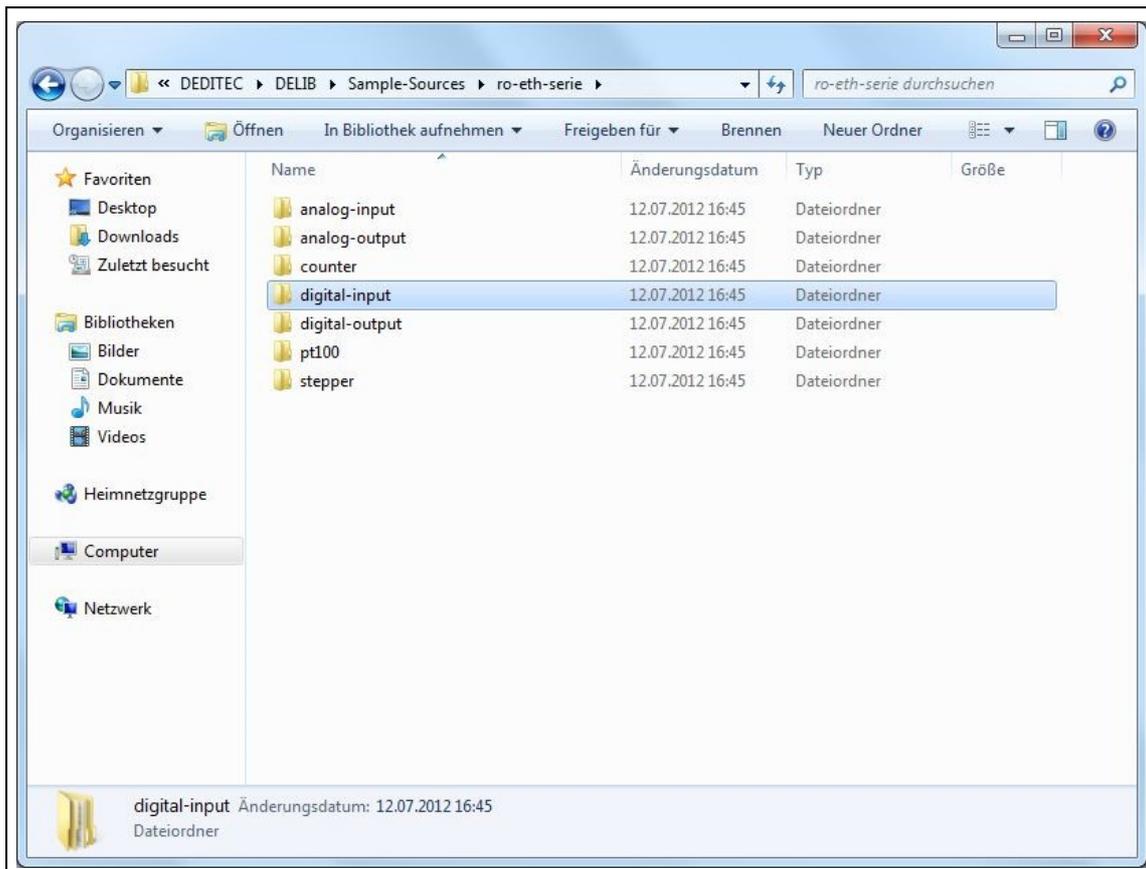
Da es sich um ein RO-ETH-Produkt handelt, wählen bzw. öffnen Sie den Ordner ro-eth-serie.



4.3.2.2. Schritt 2 - Kategorieauswahl

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

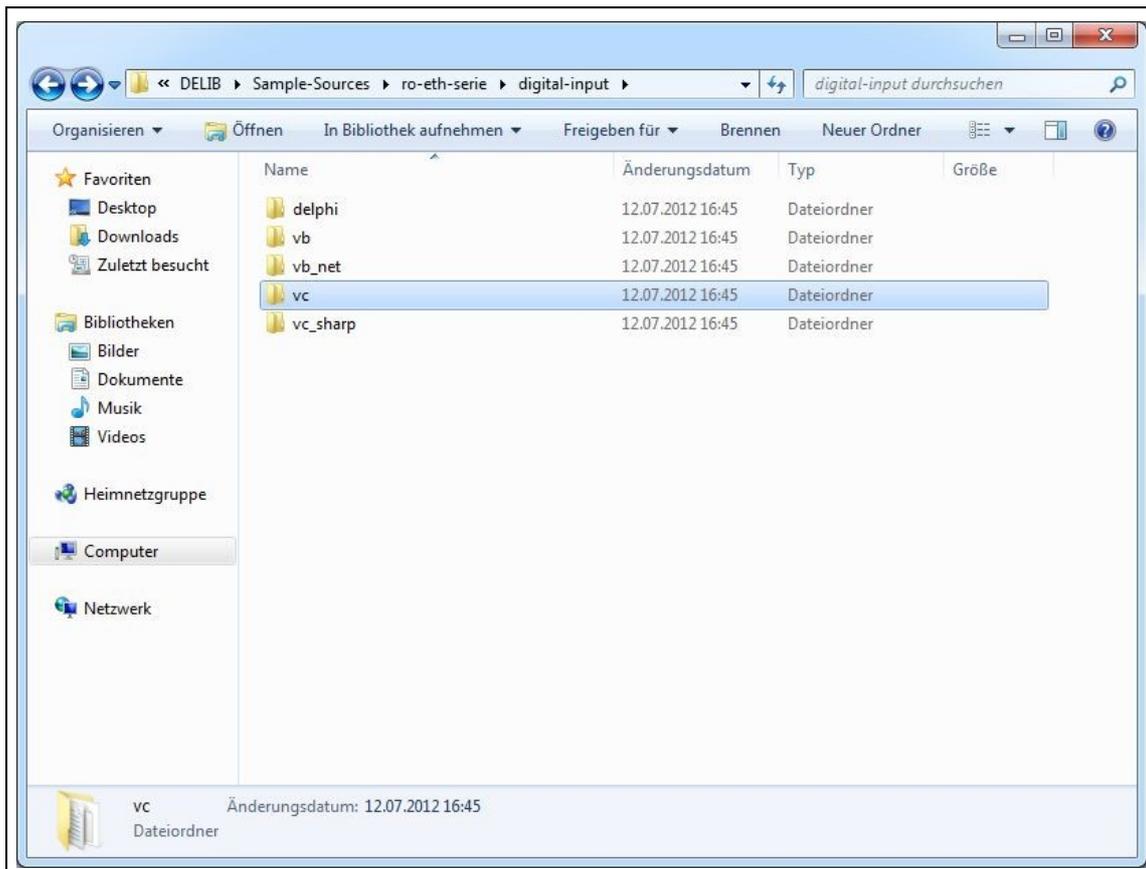
Da wir uns in diesem Beispiel auf die digitalen Eingänge konzentrieren, wählen Sie die Kategorie digital-input



4.3.2.3. Schritt 3 - Programmiersprachenauswahl

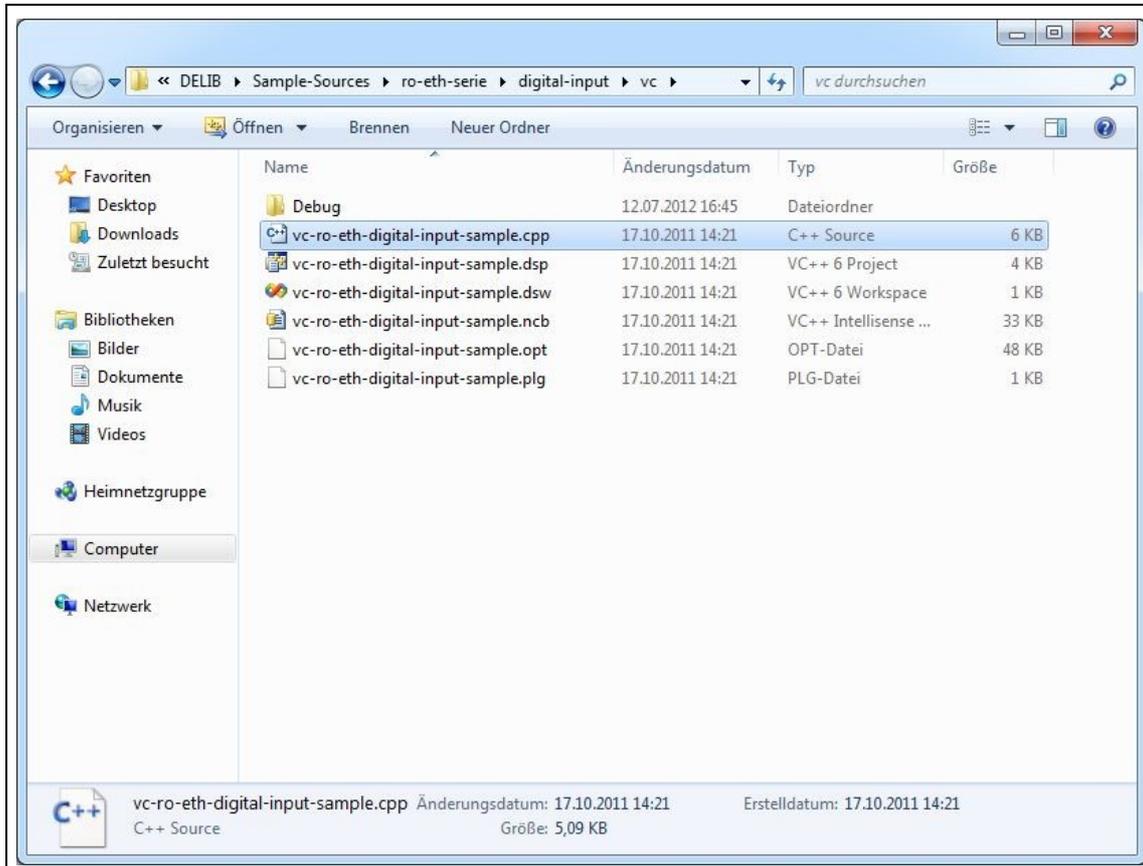
In diesem Schritt sehen Sie alle verfügbaren Programmierbeispiele der gewählten Kategorie, sortiert nach Programmiersprachen.

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

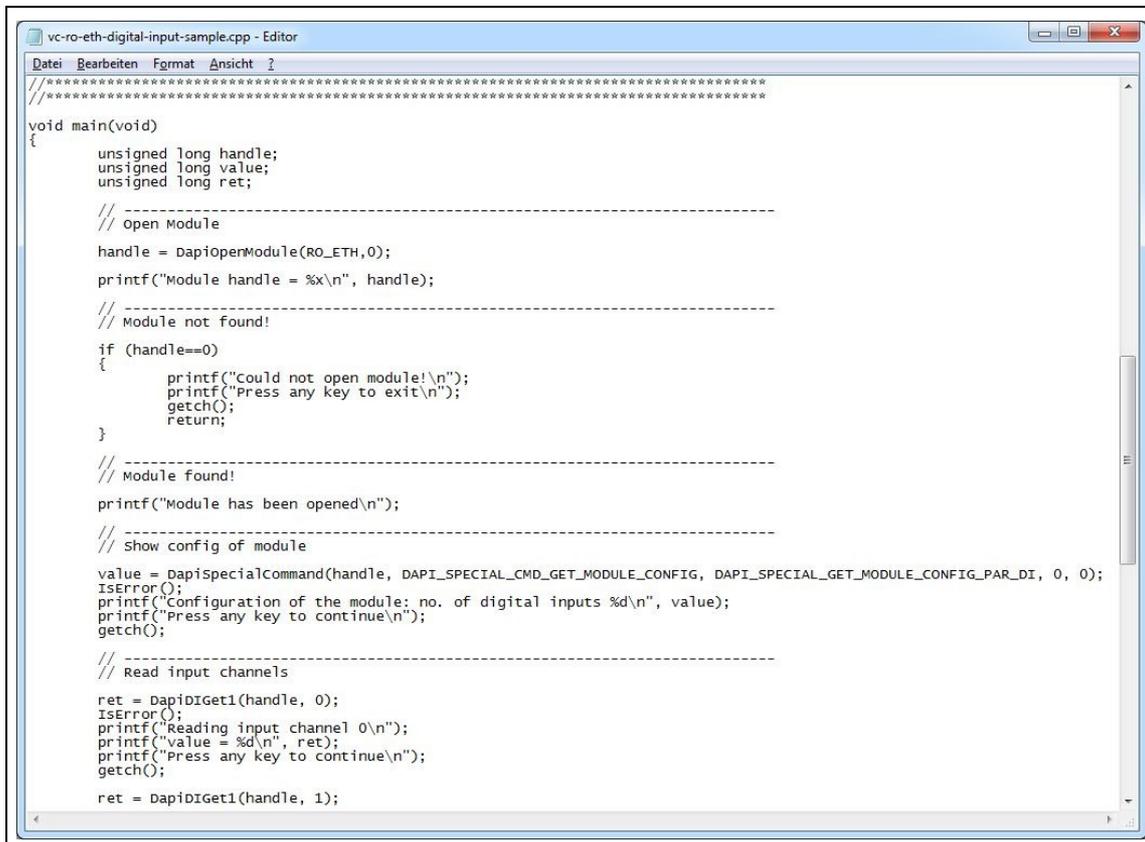


4.3.2.4. Schritt 4 - Quellcode

Nach Auswahl der Programmiersprache erhalten Sie folgende Übersicht:



Den Quellcode des 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

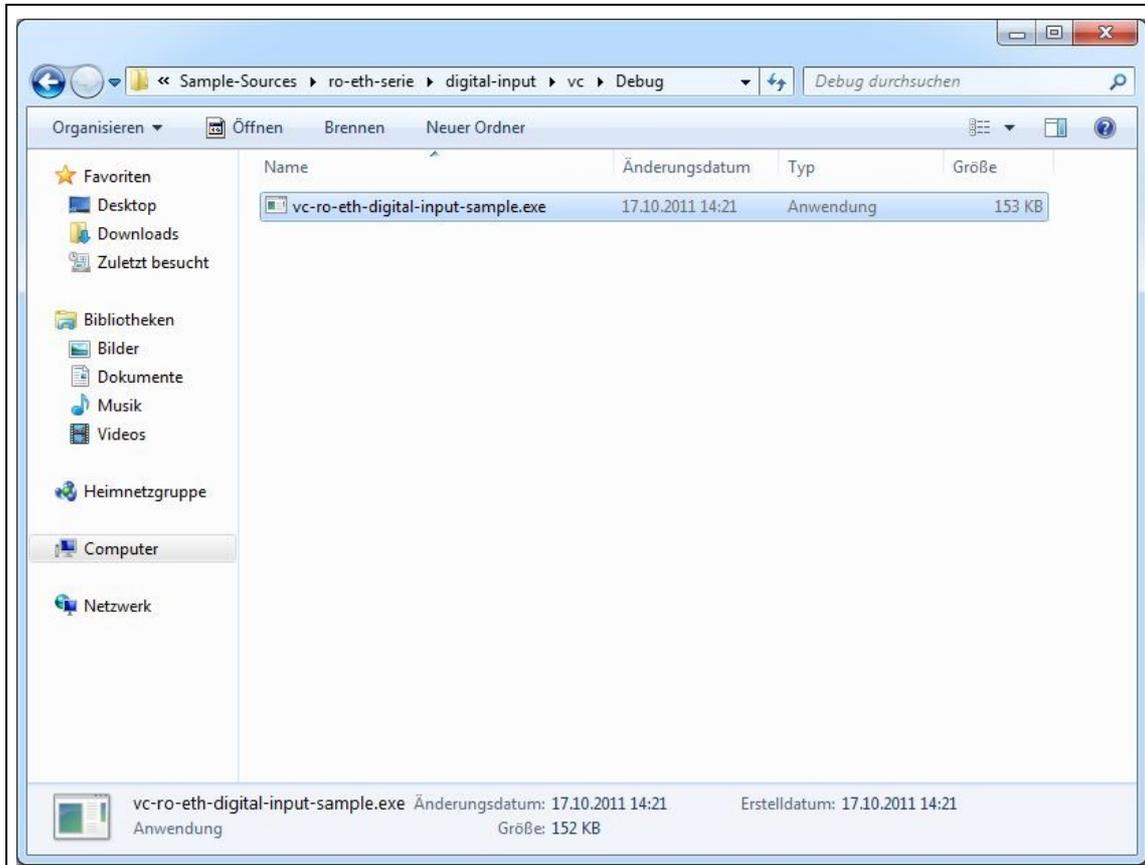
    value = DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_GET_MODULE_CONFIG, DAPI_SPECIAL_GET_MODULE_CONFIG_PAR_DI, 0, 0);
    ISError();
    printf("Configuration of the module; no. of digital inputs %d\n", value);
    printf("Press any key to continue\n");
    getch();

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

    ret = DapiDIGet1(handle, 0);
    ISError();
    printf("Reading input channel 0\n");
    printf("value = %d\n", ret);
    printf("Press any key to continue\n");
    getch();

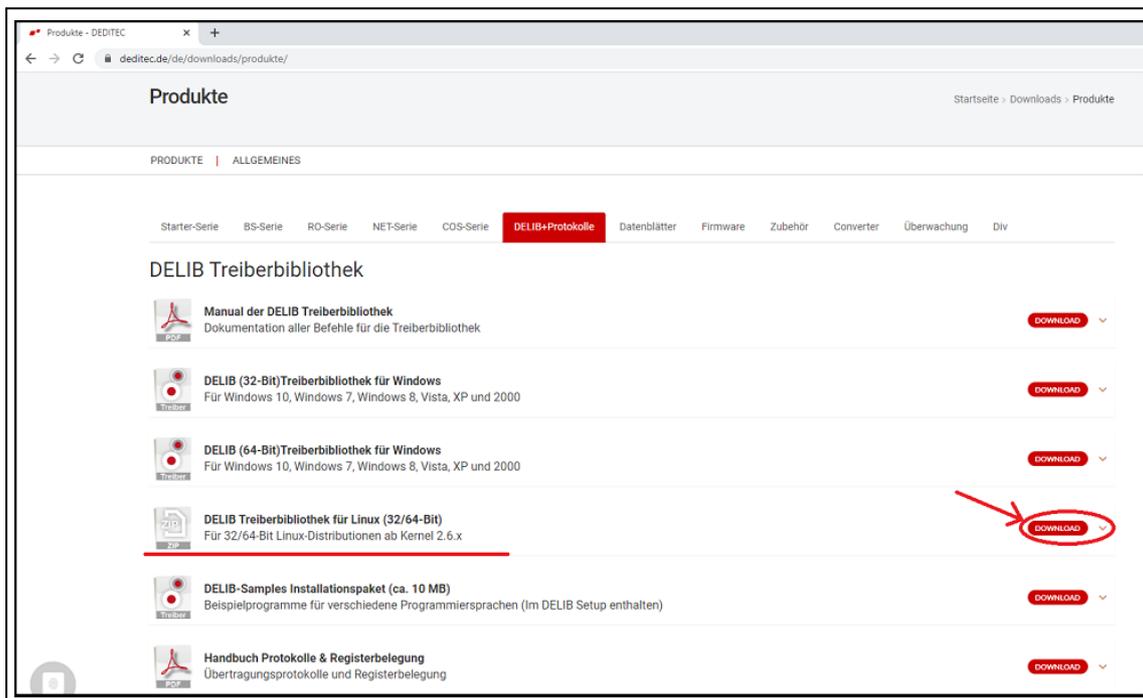
    ret = DapiDIGet1(handle, 1);
}
```

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

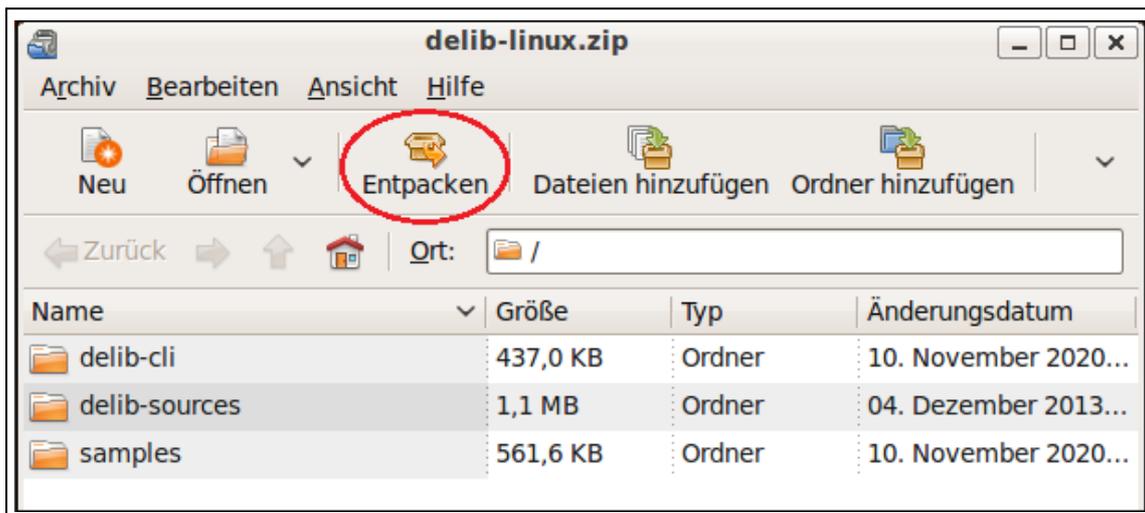


4.4. DELIB für Linux

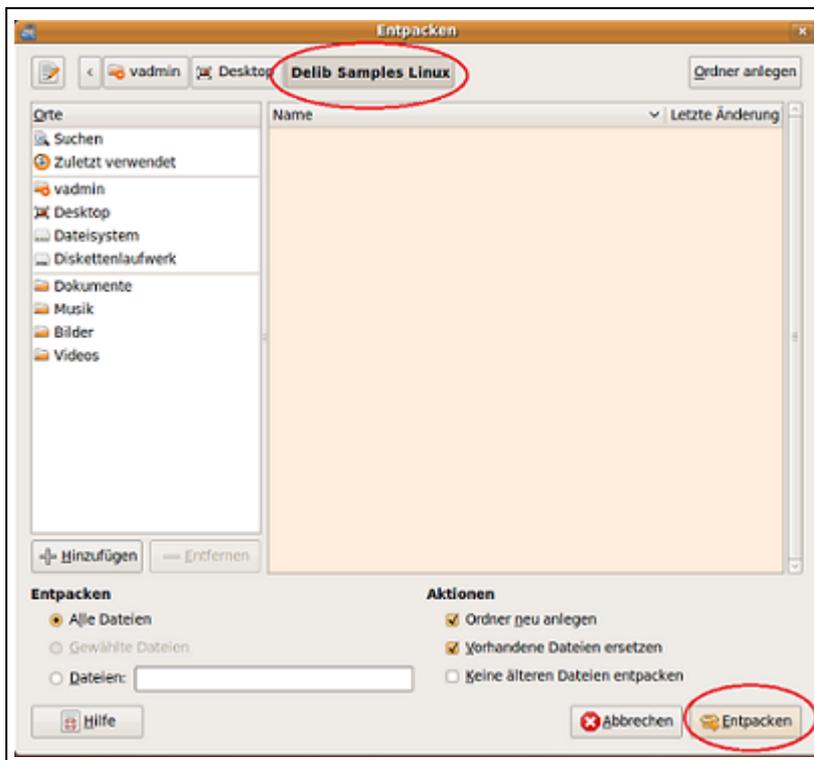
Laden Sie sich die Delib Linux Treiberbibliothek unter "www.deditec.de/de/downloads/produkte/" im Reiter „DELIB+Protokolle“ oder unter "www.deditec.de/media/zip/delib/delib-linux.zip" direkt auf ihr Linux-System.



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.



4.4.1. Verwenden der DELIB Treiberbibliothek für Linux

4.4.1.1. Delib USB-Sample in Linux

Voreinstellungen

In diesem Programmbeispiel wird ein USB_RELAIS_8 Modul angesprochen. Sollten Sie ein anderes Modul verwenden, müssen Sie in der Datei

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

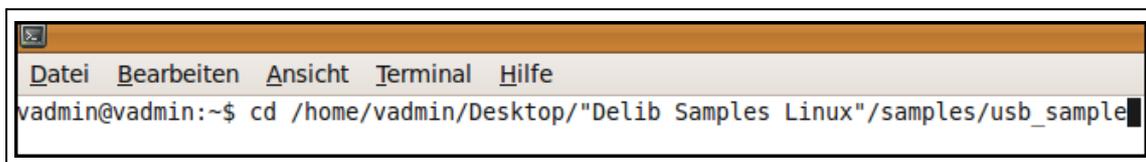
```
23
24 #include <stdio.h>
25 #include <stdlib.h>
26 #include <unistd.h>
27
28 #include "../delib-sources/delib/library/delib/delib.h"
29
30 int main()
31 {
32     ULONG i;
33     ULONG handle=0;
34
35     printf("\n\n");
36     printf("-----\n");
37     printf("-----\n");
38     printf("-----\n");
39     printf("WICHTIG !!!!\n");
40     printf("Dieses Programm bitte mit admin-Rechten ausfuehren\n");
41     printf("Also: sudo ./delib-test-digital-io <return>\n");
42     printf("-----\n");
43     printf("-----\n");
44     printf("-----\n");
45     printf("\n\n");
46
47     printf("-----\n");
48     printf("Try to open USB_RELAIS_8\n");
49     handle = DapiOpenModule(USB_RELAIS_8, 0);
50
51     if(handle == 0)
52     {
53         // Module not found
54         printf("Handle = 0x%x\n", (unsigned long) handle);
55         return 0;
56     }
57
58     printf("Handle = 0x%x\n", (unsigned long) handle);
59
```

Kompilieren des USB-Samples

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

"`cd /<Verzeichnispfad>`" zunächst in das `"/samples/usb_sample"` Verzeichnis.

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

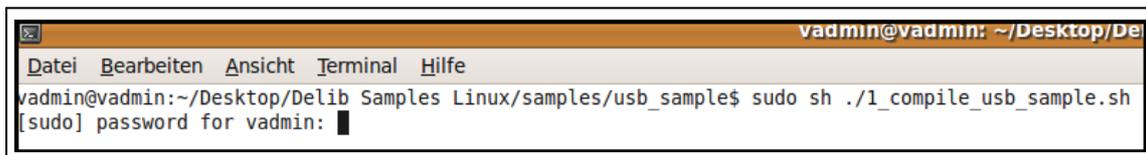


```
Terminal
Datei Bearbeiten Ansicht Terminal Hilfe
vadmin@vadmin:~$ cd /home/vadmin/Desktop/"Delib Samples Linux"/samples/usb_sample
```

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.

4.4.1.2. Delib ETH-Sample in Linux

Voreinstellungen

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

„./samples/ethernet_sample/source/eth_sample.c“ ändern (siehe Bild unten).

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

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

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

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

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

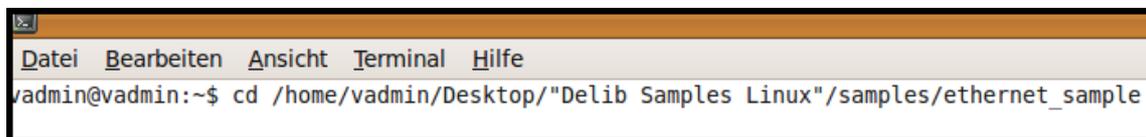
Auskommentieren,
falls keine digitalen
Eingänge vorhanden

Kompilieren des ETH-Samples

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

"`cd /<Verzeichnispfad>`" zunächst in das `"/samples/ethernet_sample"` Verzeichnis.

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

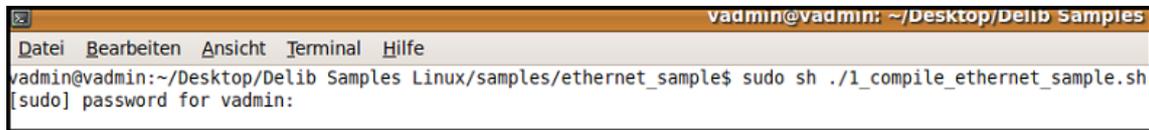


```
vadmin@vadmin:~$ cd /home/vadmin/Desktop/"Delib Samples Linux"/samples/ethernet_sample
```

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.



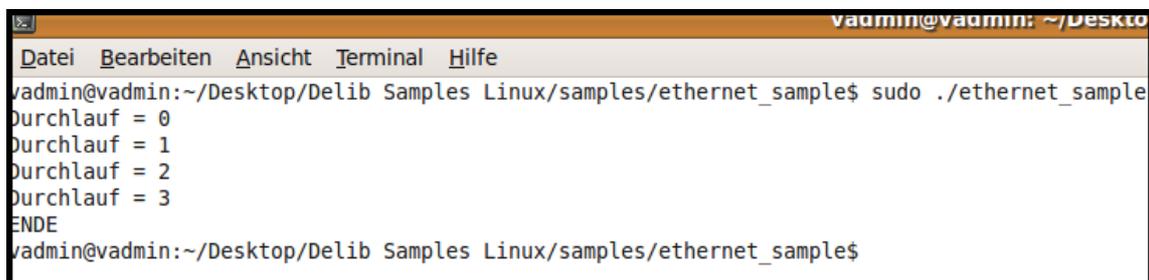
```
vadmin@vadmin: ~/Desktop/Delib Samples
Datei Bearbeiten Ansicht Terminal Hilfe
vadmin@vadmin:~/Desktop/Delib Samples Linux/samples/ethernet_sample$ sudo sh ./1_compile_ethernet_sample.sh
[sudo] password for vadmin:
```

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.

4.4.2. DELIB CLI (command-line interface) für Linux

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

Definition für USB-Module (Linux)

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

Definition für ETH-Module (Linux)

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

Hinweis:

Die einzelnen Parameter werden nur durch ein Leerzeichen getrennt.

Groß und Kleinschreibung wird hierbei nicht beachtet.

Parameter

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

4.4.2.1. Konfiguration des DELIB CLI

Voreinstellungen

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

Sie finden die "delib_cli.cfg" im Verzeichnis "/delib_cli/".

Inhalt der "delib_cli.cfg":

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

moduleID

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

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

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

moduleNR

Die moduleNR wird im DELIB Configuration Utility vergeben.

Diese Nummer dient zur Identifizierung identischer Hardware.

Der Standardwert ist 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.

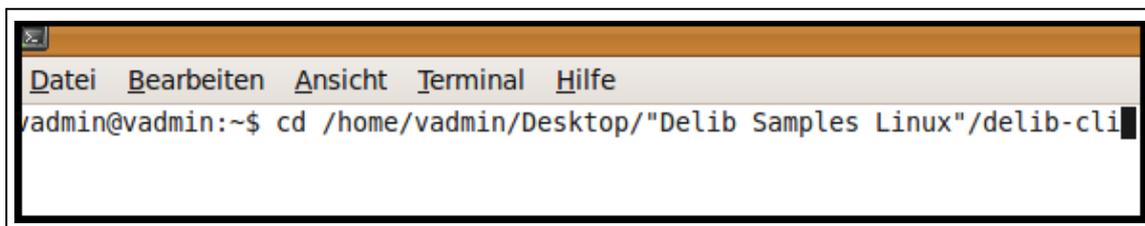
Kompilieren des Delib-CLI-Samples

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 Ordnernamen Leerzeichen enthalten sein, geben Sie diese wie im unteren Beispiel dargestellt

in " " an.

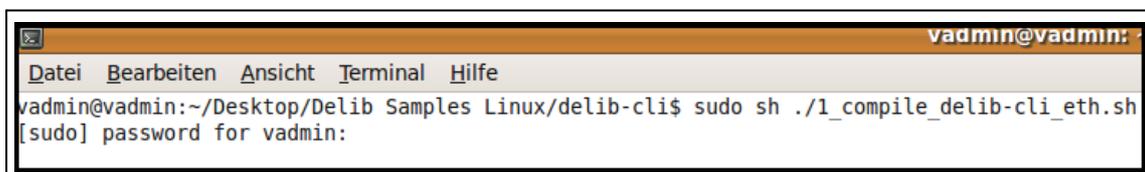


```
vadmin@vadmin:~$ cd /home/vadmin/Desktop/"Delib Samples Linux"/delib-cli
```

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

- ETH - "1_compile_delib-cli_eth.sh"
- USB - "2_compile_delib-cli_usb.sh"

Geben Sie, falls nötig, Ihr Benutzerkennwort ein.

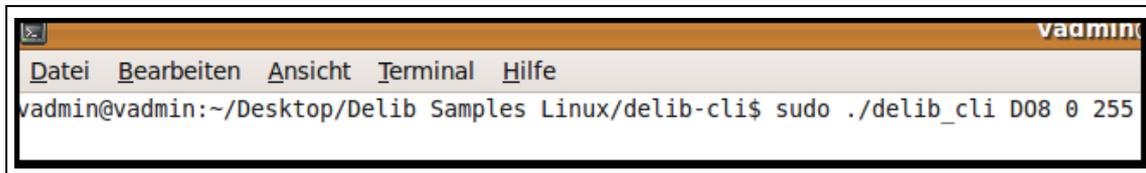


```
vadmin@vadmin:~/Desktop/Delib Samples Linux/delib-cli$ sudo sh ./1_compile_delib-cli_eth.sh  
[sudo] password for vadmin:
```

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

"sudo ./delib_cli [command] [channel] [value | unit ["nounit"]]" ausführen.

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

A terminal window with a brown title bar containing the text 'vadmin'. The menu bar includes 'Datei', 'Bearbeiten', 'Ansicht', 'Terminal', and 'Hilfe'. The main content area shows the command 'vadmin@vadmin:~/Desktop/Delib Samples Linux/delib-cli\$ sudo ./delib_cli D08 0 255' followed by a cursor.

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

4.4.2.2. DELIB CLI Beispiele

Digitale Ausgänge

```
sudo delib_cli DO1 17 1
```

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

```
sudo delib_cli DO1 3 0
```

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

Digitale Eingänge

```
sudo delib_cli DI1 3
```

Beispiel eines Rückgabewertes: **1**

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

```
sudo delib_cli DI8 0 hex
```

Beispiel eines Rückgabewertes: **0xFF**

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

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

```
sudo delib_cli FF 0
```

Beispiel eines Rückgabewertes: **192**

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

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

```
sudo delib_cli FF 32
```

Beispiel eines Rückgabewertes: **65535**

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

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

```
sudo delib_cli FF 0 hex
```

Beispiel eines Rückgabewertes: **0xD00**

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

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

Analoge Ausgänge

```
sudo delib_cli AO 7 4711
```

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

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

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

Analoge Eingänge

```
sudo delib_cli AI 2
```

Beispiel eines Rückgabewertes: **1234**

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

```
sudo delib_cli AI 2 hex
```

Beispiel eines Rückgabewertes: **0x1FA**

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

DELIB API Referenz



5. DELIB API Referenz

5.1. 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

5.2. Verwaltungsfunktionen

5.2.1. 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;
}
```

5.2.2. 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);
```

5.2.3. 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)
```

5.2.4. 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
```

5.2.5. 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);
```

5.2.6. 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 gefundenen 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;
}
```

5.3. Fehlerbehandlung

5.3.1. 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;
}
```

5.3.2. 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;
}
```

5.3.3. 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;
}
```

5.3.4. 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;
}
```

5.3.5. DapiClearLastErrorByHandle

Beschreibung

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

Definition

```
void DapiClearLastErrorByHandle();
```

Parameter

handle=Dies ist das Handle eines geöffneten Moduls.

Return-Wert

Keine

Programmierbeispiel

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

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

        DapiClearLastErrorByHandle(handle);

        return TRUE;
    }

    return FALSE;
}
```

5.4. A/D Wandler Funktionen

5.4.1. DapiADSetMode

Beschreibung

Dieser Befehl setzt den Modus für einen D/A Wandler.

Definition

void DapiDASetMode(ULONG handle, ULONG ch, ULONG mode);

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

ch=Gibt den Kanal des D/A Wandlers an (0 ..)

mode=Gibt den Modus für den D/A Wandler an (siehe delib.h)

Return-Wert

Keiner

Bemerkung

Folgende Modi werden unterstützt:

(diese sind abhängig von dem verwendeten D/A-Modul)

Unipolare Spannungen:

Modus	Wertebereich
ADDA_MODE_UNIPOL_10V	0V .. 10V
ADDA_MODE_UNIPOL_5V	0V .. 5V
ADDA_MODE_UNIPOL_2V5	0V .. 2,5V

Bipolare Spannungen:

Modus	Wertebereich
ADDA_MODE_BIPOL_10V	-10V .. +10V
ADDA_MODE_BIPOL_5V	-5V .. +5V
ADDA_MODE_BIPOL_2V5	-2,5V .. +2,5V

Ströme:

Modus	Wertebereich
ADDA_MODE_0_20mA	0 .. 20 mA
ADDA_MODE_4_20mA	4 .. 20 mA
ADDA_MODE_0_24mA	0 .. 24 mA
ADDA_MODE_0_25mA	0 .. 25 mA
ADDA_MODE_0_50mA	0 .. 50 mA

5.4.2. DapiADGetMode

Beschreibung

Dieser Befehl liest den eingestellten Modus eines A/D Wandlers zurück. Modus-Beschreibung siehe DapiADSetMode.

Definition

ULONG DapiADGetMode(ULONG handle, ULONG ch);

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

ch=Gibt den Kanal des A/D Wandlers an (0 ..)

Return-Wert

Modus des A/D Wandlers

5.4.3. DapiADGet

Beschreibung

Dieser Befehl liest einen Datenwert von einen Kanal eines A/D Wandlers.

Definition

ULONG DapiADGet(ULONG handle, ULONG ch);

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

ch=Gibt den Kanal des A/D Wandlers an (0 ..)

Return-Wert

Wert vom A/D Wandler in Digits

5.4.4. DapiADGetVolt

Beschreibung

Dieser Befehl liest einen Datenwert von einem Kanal eines A/D Wandlers in Volt.

Definition

float DapiADGetVolt(ULONG handle, ULONG ch);

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

ch=Gibt den Kanal des A/D Wandlers an (0 ..)

Return-Wert

Wert vom A/D Wandler in Volt

5.4.5. DapiADGetmA

Beschreibung

Dieser Befehl liest einen Datenwert von einem Kanal eines A/D Wandlers in mA.

Definition

float DapiADGetmA(ULONG handle, ULONG ch);

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

ch=Gibt den Kanal des A/D Wandlers an (0 ..)

Return-Wert

Wert vom A/D Wandler in mA.

Bemerkung

Dieser Befehl ist Modul abhängig. Er funktioniert natürlich nur, wenn das Modul auch den Strom-Modus unterstützt.

5.4.6. DapiSpecialADReadMultipleAD

Beschreibung

Dieser Befehl speichert die Werte bestimmter, benachbarter Kanäle eines A/D Wandlers gleichzeitig in einen Zwischenpuffer. So können anschließend die Werte nacheinander ausgelesen werden.

Vorteil hierbei ist, dass die A/D-Werte zum Einen gleichzeitig gepuffert werden, zum Anderen können die Werte mehrerer AD-Kanäle (im Vergleich zu den Befehlen DapiADGetVolt, DapiADGetmA oder DapiADGet) anschließend deutlich schneller abgefragt werden.

Definition

```
void DapiSpecialCommand(ULONG handle, DAPI_SPECIAL_CMD_AD,  
DAPI_SPECIAL_AD_READ_MULTIPLE_AD, ULONG start_ch, ULONG end_ch);
```

Parameter

handle=Dies ist das Handle eines geöffneten Moduls.

start_ch=Gibt den Start-Kanal des A/D Wandlers an, ab dem die Werte gepuffert werden (0, 1, 2, ..).

end_ch=Gibt den End-Kanal des A/D Wandlers an, bis zu dem die Werte gepuffert werden (0, 1, 2, ..).

Return-Wert

Keiner.

Bemerkung

Die Werte, die mit Befehl DapiSpecialADReadMultipleAD gepuffert wurden, können anschließend mit den Befehlen DapiADGetVolt, DapiADGetmA oder DapiADGet gelesen werden. Damit auch wirklich der gepufferte Wert gelesen wird, muss bei diesen Funktionen der Parameter "ch" mit 0x8000 logisch "oder" verknüpft werden (siehe Beispiele).

Programmierbeispiel

```
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_AD,  
DAPI_SPECIAL_AD_READ_MULTIPLE_AD, 0, 15);  
// Puffert die Werte von AD-Kanal 0..15  
  
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_AD,  
DAPI_SPECIAL_AD_READ_MULTIPLE_AD, 0, 63);  
// Puffert die Werte von AD-Kanal 0..63  
  
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_AD,  
DAPI_SPECIAL_AD_READ_MULTIPLE_AD, 16, 31);  
// Puffert die Werte von AD-Kanal 16..31  
  
value = DapiADGetVolt(handle, 0x8000 | 0);  
// Gibt den gepufferten Wert von AD-Kanal 0 in Volt zurück.  
  
value = DapiADGetmA(handle, 0x8000 | 15);  
// Gibt den gepufferten Wert von AD-Kanal 15 in mA zurück.  
  
value = DapiADGet(handle, 0x8000 | 63);  
// Gibt den gepufferten Wert von AD-Kanal 63 in Digits zurück.
```

5.5. D/A Ausgänge verwalten

5.5.1. DapiDASetMode

Beschreibung

Dieser Befehl setzt den Modus für einen D/A Wandler.

Definition

```
void DapiDASetMode(ULONG handle, ULONG ch, ULONG mode);
```

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

ch=Gibt den Kanal des D/A Wandlers an (0 ..)

mode=Gibt den Modus für den D/A Wandler an (siehe delib.h)

Return-Wert

Keiner

Bemerkung

Folgende Modi werden unterstützt:

(diese sind abhängig von dem verwendeten D/A-Modul)

Unipolare Spannungen:

Modus	Wertebereich
ADDA_MODE_UNIPOL_10V	0V .. 10V
ADDA_MODE_UNIPOL_5V	0V .. 5V
ADDA_MODE_UNIPOL_2V5	0V .. 2,5V

Bipolare Spannungen:

Modus	Wertebereich
ADDA_MODE_BIPOL_10V	-10V .. +10V
ADDA_MODE_BIPOL_5V	-5V .. +5V
ADDA_MODE_BIPOL_2V5	-2,5V .. +2,5V

Ströme:

Modus	Wertebereich
ADDA_MODE_0_20mA	0 .. 20 mA
ADDA_MODE_4_20mA	4 .. 20 mA
ADDA_MODE_0_24mA	0 .. 24 mA
ADDA_MODE_0_25mA	0 .. 25 mA
ADDA_MODE_0_50mA	0 .. 50 mA

5.5.2. DapiDAGetMode

Beschreibung

Dieser Befehl liest den eingestellten Modus eines D/A Wandlers zurück.

Definition

ULONG DapiDAGetMode(ULONG handle, ULONG ch);

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

ch=Gibt den Kanal des D/A Wandlers an (0 ..)

Return-Wert

Modus des D/A Wandlers

5.5.3. DapiDASet

Beschreibung

Dieser Befehl übergibt ein Datenwert an einen Kanal eines 16-Bit D/A Wandlers.

Definition

void DapiDASet(ULONG handle, ULONG ch, ULONG data);

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

ch=Gibt den Kanal des D/A Wandlers an (0 ..)

data=Gibt den Datenwert an, der geschrieben wird (16-Bit Datenwert -> Datenwertebereich: 0-65535)

Return-Wert

Keiner

Programmierbeispiel

```
DapiDASet(handle, 0, 65535);  
// Setzt den 1. Ausgang des D/A Wandlers auf maximalen Wert des  
gewählten Modus.  
//(bei ausgewähltem Modus ADDA_MODE_UNIPOL_10V wird der 1. Ausgang  
des D/A Wandlers auf 10V gesetzt)
```

5.5.4. DapiDASetVolt

Beschreibung

Dieser Befehl setzt eine Spannung an einen Kanal eines D/A Wandlers.

Definition

void DapiDASetVolt(ULONG handle, ULONG ch, float data);

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

ch=Gibt den Kanal des D/A Wandlers an (0 ..)

data=Gibt die Spannung an, die eingestellt werden soll [V]

Return-Wert

Keiner

Programmierbeispiel

```
DapiDASetVolt(handle, 0, 5,4321);  
// Setzt den 1. Ausgang des D/A Wandlers auf 5,4321 V
```

5.5.5. DapiDASetmA

Beschreibung

Dieser Befehl setzt einen Strom an einen Kanal eines D/A Wandlers.

Definition

void DapiDASetmA(ULONG handle, ULONG ch, float data);

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

ch=Gibt den Kanal des D/A Wandlers an (0 ..)

data=Gibt den Strom an, der geschrieben wird [mA]

Return-Wert

Keiner

Bemerkung

Dieser Befehl ist Modul abhängig. Er funktioniert natürlich nur, wenn das Modul auch den Strom-Modus unterstützt.

5.5.6. DapiSpecialCmd_DA

Beschreibung

Dieser Befehl setzt die Spannungswerte bei einem Kanal beim Einschalten bzw. nach einem Timeout eines D/A Wandlers (EEPROM-Konfiguration).

Definition

```
void DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_DA, cmd, ch, 0);
```

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

ch=Gibt den Kanal des D/A Wandlers an (0, 1, 2, ..)

Zurücksetzen der Einstellungen auf Default Konfiguration

cmd=DAPI_SPECIAL_DA_PAR_DA_LOAD_DEFAULT

Speichern der Konfiguration in das EEPROM

cmd=DAPI_SPECIAL_DA_PAR_DA_SAVE_EEPROM_CONFIG

Laden der Konfiguration aus dem EEPROM

cmd=DAPI_SPECIAL_DA_PAR_DA_LOAD_EEPROM_CONFIG

Return-Wert

Keiner

Bemerkung

DAPI_SPECIAL_CMD_DA_PAR_DA_LOAD_DEFAULT

Mit diesem Befehl wird die Default Konfiguration eines D/A Wandlers geladen. Der D/A Wandler hat jetzt als Ausgabespannung 0V.

DAPI_SPECIAL_DA_PAR_DA_SAVE_EEPROM_CONFIG

Mit diesem Befehl wird die aktuelle D/A Wandler Einstellung (Spannung/Stromwert, Enable/Disable und D/A Wandler Modus) in das EEPROM gespeichert.

DAPI_SPECIAL_DA_PAR_DA_LOAD_EEPROM_CONFIG

Mit diesem Befehl wird der D/A Wandler, mit der im EEPROM gespeicherten Konfiguration, gesetzt.

Programmierbeispiel

```
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_DA,  
DAPI_SPECIAL_DA_PAR_DA_LOAD_DEFAULT, 1, 0);  
//Zurücksetzen der EEPROM-Konfiguration auf Default Konfiguration bei  
Kanal 1.  
  
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_DA,  
DAPI_SPECIAL_DA_PAR_DA_SAVE_EEPROM_CONFIG, 3, 0);  
//Speichern der D/A Wandler Einstellungen in das EEPROM bei Kanal 3.  
  
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_DA,  
DAPI_SPECIAL_DA_PAR_DA_LOAD_EEPROM_CONFIG, 2, 0);  
//Setzen des D/A Wandlers, mit der im EEPROM gespeicherten Konfiguration  
bei Kanal 2.
```

5.6. Digitale Eingänge lesen

5.6.1. DapiDIGet1

Beschreibung

Dieser Befehl liest einen einzelnen digitalen Eingang.

Definition

ULONG DapiDIGet1(ULONG handle, ULONG ch);

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

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

Return-Wert

Zustand des Eingangs (0/1)

5.6.2. DapiDIGet8

Beschreibung

Dieser Befehl liest gleichzeitig 8 digitale Eingänge.

Definition

ULONG DapiDIGet8(ULONG handle, ULONG ch);

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

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

Return-Wert

Zustand der gelesenen Eingänge

5.6.3. DapiDIGet16

Beschreibung

Dieser Befehl liest gleichzeitig 16 digitale Eingänge.

Definition

ULONG DapiDIGet16(ULONG handle, ULONG ch);

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

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

Return-Wert

Zustand der gelesenen Eingänge

5.6.4. DapiDIGet32

Beschreibung

Dieser Befehl liest gleichzeitig 32 digitale Eingänge.

Definition

ULONG DapiDIGet32(ULONG handle, ULONG ch);

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

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

Return-Wert

Zustand der gelesenen Eingänge

Programmierbeispiel

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

5.6.5. DapiDIGet64

Beschreibung

Dieser Befehl liest gleichzeitig 64 digitale Eingänge.

Definition

ULONG DapiDIGet64(ULONG handle, ULONG ch);

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

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

Return-Wert

Zustand der gelesenen Eingänge

5.6.6. DapiDIGetFF32

Beschreibung

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

Definition

ULONG DapiDIGetFF32(ULONG handle, ULONG ch);

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

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

Return-Wert

Zustand von 32 Eingangszustandsänderungen

5.6.7. DapiDIGetCounter

Beschreibung

Dieser Befehl liest den Eingangszähler eines digitalen Eingangs.

Definition

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

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

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

mode=0 (Normale Zählfunktion)

mode=DAPI_CNT_MODE_READ_WITH_RESET (Zähler auslesen und direktes Counter resettet)

mode=DAPI_CNT_MODE_READ_LATCHED (Auslesen des gespeicherten Zählerstandes)

Return-Wert

Angabe des Zählerwertes

Programmierbeispiel

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

5.6.8. DapiSpecialCounterLatchAll

Beschreibung

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

So können anschließend alle Zählerstände des Latches nacheinander ausgelesen werden.

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

Definition

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

Parameter

Keine

Return-Wert

Keiner

Bemerkung

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

DELIB Übersichtstabelle entnehmen.

Programmierbeispiel

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

5.6.9. DapiSpecialCounterLatchAllWithReset

Beschreibung

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

Zusätzlich werden die Zählerstände der Eingangszähler im Anschluss zurückgesetzt.

Definition

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

Parameter

Keine

Return-Wert

Keiner

Bemerkung

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

DELIB Übersichtstabelle entnehmen.

Programmierbeispiel

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

5.6.10. DapiSpecialDIFilterValueSet

Beschreibung

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

Definition

```
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_DI,  
DAPI_SPECIAL_DI_FILTER_VALUE_SET, ULONG time_ms, 0);
```

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

time_ms=Zeitintervall [ms], indem digitale Eingangskanäle gelesen werden.

Bemerkung

Standardwert: 0ms

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

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

Programmierbeispiel

```
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_DI,  
DAPI_SPECIAL_DI_FILTER_VALUE_SET, 5, 0);  
// Setzt das Zeitintervall auf 5ms  
  
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_DI,  
DAPI_SPECIAL_DI_FILTER_VALUE_SET, 150, 0);  
// Setzt das Zeitintervall auf 150ms
```

5.6.11. DapiSpecialDIFilterValueGet

Beschreibung

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

Definition

```
ULONG DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_DI,  
DAPI_SPECIAL_DI_FILTER_VALUE_GET, 0, 0);
```

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

Return-Wert

Zeit [ms]

Bemerkung

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

DELIB Übersichtstabelle entnehmen.

Programmierbeispiel

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

5.6.12. Dapi_Special_DI_FF_Filter_Value_Set

Beschreibung

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

Definition

```
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_DI,  
DAPI_SPECIAL_DI_FF_FILTER_VALUE_SET, ULONG time_ms, 0);
```

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

time_ms=Zeitintervall [ms], indem digitale Eingangskanäle abgetastet werden.

Bemerkung

Dieser Befehl unterstützt nur Impulszeiten zwischen 5ms und 255ms.

Wird keine Zeit gesetzt, ist der Default-Wert 100ms.

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

DELIB Übersichtstabelle entnehmen.

Programmierbeispiel

```
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_DI,  
DAPI_SPECIAL_DI_FF_FILTER_VALUE_SET, 5, 0);  
// Setzt das Zeitintervall auf 5ms  
  
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_DI,  
DAPI_SPECIAL_DI_FF_FILTER_VALUE_SET, 150, 0);  
// Setzt das Zeitintervall auf 150ms
```

5.6.13. Dapi_Special_DI_FF_Filter_Value_Get

Beschreibung

Dieser Befehl gibt den vorher festgelegten Wert des Zeitintervalls zur Abtastung der Eingangs-Flip-Flops und der Eingangs-Zähler in [ms] zurück.

Definition

*ULONG DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_DI,
DAPI_SPECIAL_DI_FF_FILTER_VALUE_GET, 0, 0);*

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

Return-Wert

Zeit [ms]

Bemerkung

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

DELIB Übersichtstabelle entnehmen.

Programmierbeispiel

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

5.7. Digitale Ausgänge verwalten

5.7.1. DapiDOSet1

Beschreibung

Dieser Befehl setzt einen einzelnen Ausgang.

Definition

void DapiDOSet1(ULONG handle, ULONG ch, ULONG data);

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

ch=Gibt die Nummer des zu setzenden Ausgangs an (0 ..)

data=Gibt den Datenwert an, der geschrieben wird (0 / 1)

Return-Wert

Keiner

5.7.2. DapiDOSet8

Beschreibung

Dieser Befehl setzt gleichzeitig 8 digitale Ausgänge.

Definition

void DapiDOSet8(ULONG handle, ULONG ch, ULONG data);

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

ch=Gibt die Nummer des Ausgangs an, ab dem geschrieben werden soll (0, 8, 16, 24, 32, ..)

data=Gibt die Datenwerte an, die geschrieben werden

Return-Wert

Keiner

5.7.3. DapiDOSet16

Beschreibung

Dieser Befehl setzt gleichzeitig 16 digitale Ausgänge.

Definition

void DapiDOSet16(ULONG handle, ULONG ch, ULONG data);

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

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

data=Gibt die Datenwerte an, die geschrieben werden

Return-Wert

Keiner

5.7.4. DapiDOSet32

Beschreibung

Dieser Befehl setzt gleichzeitig 32 digitale Ausgänge.

Definition

void DapiDOSet32(ULONG handle, ULONG ch, ULONG data);

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

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

data=Gibt die Datenwerte an, die geschrieben werden

Return-Wert

Keiner

Programmierbeispiel

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

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

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

5.7.5. DapiDOSet64

Beschreibung

Dieser Befehl setzt gleichzeitig 64 digitale Ausgänge.

Definition

```
void DapiDOSet64(ULONG handle, ULONG ch, ULONG data);
```

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

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

data=Gibt die Datenwerte an, die geschrieben werden

Return-Wert

Keiner

5.7.6. DapiDOSet1_WithTimer

Beschreibung

Diese Funktion setzt einen Digitalausgang (ch) auf einen Wert (data - 0 oder 1) für eine bestimmte Zeit in ms.

Definition

```
void DapiDOSet1_WithTimer(ULONG handle, ULONG ch, ULONG data, ULONG time_ms);
```

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

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

data=Gibt die Datenwerte an, die geschrieben werden

time_ms=Gibt die Zeit an, in der der Ausgang gesetzt wird [ms]

Return-Wert

Keiner

Bemerkung:

Dieser Befehl wird nur von unserem RO-08-R8 Modul unterstützt.

Dieser Befehl verliert seine Gültigkeit, sofern er mit anderen Werten überschrieben wird.

Möchte man den Befehl deaktivieren, dann muss er mit time_ms=0 überschrieben werden.

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

DELIB Übersichtstabelle entnehmen.

Programmierbeispiel

```
DapiDOSet1_WithTimer(handle, 2, 1, 1000);  
//Setting channel 2 for 1000msec to 1
```

5.7.7. DapiDOReadback32

Beschreibung

Dieser Befehl liest die 32 digitalen Ausgänge zurück.

Definition

ULONG DapiDOReadback32(ULONG handle, ULONG ch);

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

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

Return-Wert

Zustand von 32 Ausgängen.

5.7.8. DapiDOReadback64

Beschreibung

Dieser Befehl liest die 64 digitalen Ausgänge zurück.

Definition

ULONG DapiDOReadback64(ULONG handle, ULONG ch);

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

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

Return-Wert

Zustand von 64 Ausgängen.

5.7.9. DapiDOSetBit32

Beschreibung

Mit diesem Befehl können Ausgänge gezielt auf 1 geschaltet werden, ohne die Zustände der benachbarten Ausgänge zu ändern.

Definition

```
void DapiDOSetBit32(uint handle, uint ch, uint data);
```

Parameter

handle = Dies ist das Handle eines geöffneten Moduls

ch = Gibt die Nummer des Ausgangs an, ab dem geschrieben werden soll

data = Gibt den Datenwert an, der geschrieben werden soll (bis zu 32 Bit)

Return-Wert

Keiner

Bemerkung:

Nur die Bits mit einer Wertigkeit von 1 im data Parameter werden vom Befehl berücksichtigt.

Programmierbeispiel

```
data = 0x1; // Ausgang 0 wird auf 1 gesetzt, der Zustand von Ausgang  
1-31 bleibt unberührt  
DapiDOSetBit32(handle, 0, data);  
data = 0xf; // Ausgang 0-3 wird auf 1 gesetzt, der Zustand von Ausgang  
4-31 bleibt unberührt  
DapiDOSetBit32(handle, 0, data);  
data = 0xff; // Ausgang 0-7 wird auf 1 gesetzt, der Zustand von  
Ausgang 8-31 bleibt unberührt  
DapiDOSetBit32(handle, 0, data);  
data = 0xff000000; // Ausgang 23-31 wird auf 1 gesetzt, der Zustand  
von Ausgang 0-22 bleibt unberührt  
DapiDOSetBit32(handle, 0, data);
```

5.7.10. DapiDOClrBit32

Beschreibung

Mit diesem Befehl können Ausgänge gezielt auf 0 geschaltet werden, ohne die Zustände der benachbarten Ausgänge zu ändern.

Definition

```
void DapiDOClrBit32(uint handle, uint ch, uint data);
```

Parameter

handle = Dies ist das Handle eines geöffneten Moduls

ch = Gibt die Nummer des Ausgangs an, ab dem geschrieben werden soll

data = Gibt den Datenwert an, der geschrieben werden soll (bis zu 32 Bit)

Return-Wert

Keiner

Bemerkung:

Nur die Bits mit einer Wertigkeit von 1 im data Parameter werden vom Befehl berücksichtigt.

Programmierbeispiel

```
data = 0x1; // Ausgang 0 wird auf 0 gesetzt, der Zustand von Ausgang  
1-31 bleibt unberührt  
DapiDOSetBit32(handle, 0, data);  
data = 0xf; // Ausgang 0-3 wird auf 0 gesetzt, der Zustand von Ausgang  
4-31 bleibt unberührt  
DapiDOSetBit32(handle, 0, data);  
data = 0xff; // Ausgang 0-7 wird auf 0 gesetzt, der Zustand von  
Ausgang 8-31 bleibt unberührt  
DapiDOSetBit32(handle, 0, data);  
data = 0xff000000; // Ausgang 23-31 wird auf 0 gesetzt, der Zustand  
von Ausgang 0-22 bleibt unberührt  
DapiDOSetBit32(handle, 0, data);
```

5.8. Ausgabe-Timeout verwalten

5.8.1. DapiSpecialCMDTimeout

Beschreibung

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

Es gibt seit 2021 drei unterschiedliche Timeout-Methoden.

"normalen" Timeout

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

Vorgehensweise für den Timeout-Befehl:

Der Timeout wird per Befehl aktiviert.

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

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

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

"auto reactivate" Timeout

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

Vorgehensweise für den Timeout-Befehl:

Der Timeout wird per Befehl aktiviert.

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

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

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

"secure outputs" Timeout

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

Vorgehensweise für den Timeout-Befehl:

Der Timeout wird per Befehl aktiviert.

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

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

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

Definition

DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_TIMEOUT, cmd, par1, par2);

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

cmd = auszuführende Funktion

par1 = Wert, der an die Funktion übergeben wird

par2 = Wert, der an die Funktion übergeben wird

5.8.1.1. DapiSpecialTimeoutSetValueSec

Beschreibung

Dieser Befehl dient zum Setzen der Timeout-Zeit.

Definition

DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_TIMEOUT, cmd, par1, par2);

Parameter

cmd = DAPI_SPECIAL_TIMEOUT_SET_VALUE_SEC

par1 = Sekunden [s]

par2 = Millisekunden [100ms] (Wert 6 = 600ms)

Bemerkung

Der zulässige Wertebereich der Zeitangabe liegt zwischen 0,1 Sekunden und 6553 Sekunden

Programmierbeispiel

```
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_TIMEOUT,  
DAPI_SPECIAL_TIMEOUT_SET_VALUE_SEC, 3, 7);  
//Die Zeit des Timeouts wird auf 3,7sek gesetzt.
```

5.8.1.2. DapiSpecialTimeoutActivate

Beschreibung

Dieser Befehl aktiviert den "normalen" Timeout.

Nach dem Timeout-Ereignis werden..

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

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

Erst wieder, wenn er wieder aktiviert wurde.

Definition

DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_TIMEOUT, cmd, 0, 0);

Parameter

cmd = DAPI_SPECIAL_TIMEOUT_ACTIVATE

Programmierbeispiel

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

5.8.1.3. DapiSpecialTimeoutActivateAutoReactivate

Beschreibung

Dieser Befehl aktiviert den "auto reactivate" Timeout.

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

Nach dem Timeout-Ereignis werden..

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

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

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

Definition

DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_TIMEOUT, cmd, 0, 0);

Parameter

cmd = DAPI_SPECIAL_TIMEOUT_ACTIVATE_AUTO_REACTIVATE

Programmierbeispiel

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

5.8.1.4. DapiSpecialTimeoutActivateSecureOutputs

Beschreibung

Dieser Befehl aktiviert den "secure" Timeout.

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

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

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

Nach dem Timeout-Ereignis werden..

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

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

Timeouts oder Deaktivieren des Timeouts können die Ausgänge beschrieben werden.

Definition

DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_TIMEOUT, cmd, 0, 0);

Parameter

cmd = DAPI_SPECIAL_TIMEOUT_ACTIVATE_SECURE_OUTPUTS

Programmierbeispiel

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

5.8.1.5. DapiSpecialTimeoutDeactivate

Beschreibung

Dieser Befehl deaktiviert den Timeout.

Definition

DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_TIMEOUT, cmd, 0, 0);

Parameter

cmd = DAPI_SPECIAL_TIMEOUT_DEACTIVATE

Programmierbeispiel

```
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_TIMEOUT,  
DAPI_SPECIAL_TIMEOUT_DEACTIVATE, 0, 0);  
//Der Timeout wird deaktiviert.
```

5.8.1.6. DapiSpecialTimeoutGetStatus

Beschreibung

Dieser Befehl dient dem Auslesen des Timeout-Status.

Definition

```
ULONG DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_TIMEOUT,  
DAPI_SPECIAL_TIMEOUT_GET_STATUS, 0, 0);
```

Parameter

cmd = DAPI_SPECIAL_TIMEOUT_GET_STATUS

Return-Wert

Return = 0 (Timeout ist deaktiviert)

Werte für den "normalen" Timeout

Return = 1 (Timeout "normal" ist aktiviert)

Return = 2 (Timeout "normal" hat stattgefunden)

Werte für den "auto reactivate" Timeout

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

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

Werte für den "secure" Timeout

Return = 5 (Timeout "secure" ist aktiviert)

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

Programmierbeispiel

```
unsigned long status = DapiSpecialCommand(handle,  
DAPI_SPECIAL_CMD_TIMEOUT,  
DAPI_SPECIAL_TIMEOUT_GET_STATUS, 0, 0);  
printf("Status = %lu\n", status);  
//Abfrage des Timeout-Status mit Ausgabe.
```

5.8.1.7. DapiSpecialTimeoutDoValueMaskWRSet32

Beschreibung

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

Definition

DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_TIMEOUT, cmd, ch, par2);

Parameter

cmd = DAPI_SPECIAL_TIMEOUT_DO_VALUE_MASK_WR_SET32

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

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

Programmierbeispiel

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

5.8.1.8. DapiSpecialTimeoutDoValueMaskRDSet32

Beschreibung

Dieser Befehl dient dem Auslesen der übergebenen Werte.

Definition

ULONG DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_TIMEOUT, cmd, 0, 0);

Parameter

cmd = DAPI_SPECIAL_TIMEOUT_DO_VALUE_MASK_RD_SET32

Return-Wert

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

Programmierbeispiel

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

5.8.1.9. DapiSpecialTimeoutDoValueMaskWRClr32

Beschreibung

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

Definition

DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_TIMEOUT, cmd, ch, par2);

Parameter

cmd = DAPI_SPECIAL_TIMEOUT_DO_VALUE_MASK_WR_CLR32

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

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

Programmierbeispiel

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

5.8.1.10. DapiSpecialTimeoutDoValueMaskRDClr32

Beschreibung

Dieser Befehl dient dem Auslesen der übergebenen Werte.

Definition

ULONG DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_TIMEOUT, cmd, 0, 0);

Parameter

cmd = DAPI_SPECIAL_TIMEOUT_DO_VALUE_MASK_RD_CLR32

Return-Wert

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

Programmierbeispiel

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

5.8.1.11. DapiSpecialTimeoutDoValueLoadDefault

Beschreibung

Setzt die SET- und CLR-Werte auf den Default-Wert zurück.

(SET-Wert = 0, CLR-Wert = FFFFFFFF)

Definition

DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_TIMEOUT, cmd, 0, 0);

Parameter

cmd = DAPI_SPECIAL_TIMEOUT_DO_VALUE_LOAD_DEFAULT

Programmierbeispiel

```
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_TIMEOUT,  
DAPI_SPECIAL_TIMEOUT_DO_VALUE_LOAD_DEFAULT, 0, 0);  
//SET- und CLR-Werte werden auf den Default-Wert gesetzt.
```

5.9. CAN Runtime Funktionen

5.9.1. RunTimeVarWriteToModule

Beschreibung

Bei dem Start des Moduls werden die Einstellungen aus dem **Module-Configuration-Memory** geladen und benutzt. Mit Hilfe dieser Befehle lassen sich die Einstellungen während der Laufzeit verändern und auslesen.

Sie werden jedoch nicht in das **Module-Configuration-Memory** gespeichert und gehen daher nach Modulneustart verloren.

Parameter

handle = Dies ist das Handle eines geöffneten Moduls

par = Laufzeitenvariable, die beschrieben oder ausgelesen werden soll

index = Gibt den Index des TX/RX-Paketes an [Wertebereich 0-7]

value = Der Wert um den die Laufzeitenvariable geändert werden soll. Bei der Read-Funktion wird hier eine Referenz übergeben

Bemerkung

Das value muss immer als Hex-Wert angegeben werden. Der Return-Wert ist ebenfalls in Hex. Eine Auflistung von Module, die diese Funktionen unterstützen, können Sie unserer **Delib Übersichtstabelle** entnehmen.

Definition

Für ein besseres Verständnis unserer Beispiele, verwenden wir für das Schreiben die Funktion **RunTimeVarWriteToModule** und für das Lesen **RunTimeVarReadFromModule**

Der darin befindliche Quellcode lautet wie folgt:

//Lesen der Werte

```
public static uint RunTimeVarReadFromModule(uint handle, uint par,
uint index, ref uint value)
{
    byte[] dummy_buff = new byte[] { 0 };
    uint u0 = 0;

    if(DT.Delib.DapiSpecialCommandExt(handle,
        DT.Ext.DAPI_SPECIAL_CMDEXT_CAN_RD_RUNTIME_VALUE,
        par, index, value, ref u0, ref u0,
        dummy_buff, 0, dummy_buff, 0, dummy_buff, 0, ref u0) !=
        DT.RETURN_OK)
    {
        return DT.Error.DAPI_ERR_DEV_CONFIG_READ_ERROR;
    }
    return DT.Error.DAPI_ERR_NONE;
}
```

//Schreiben der Werte

```
public static uint RunTimeVarWriteToModule(uint handle, uint par,
uint index, uint value)
{
    byte[] dummy_buff = new byte[] { 0 };
    uint u0 = 0;

    if(DT.Delib.DapiSpecialCommandExt(handle,
        DT.Ext.DAPI_SPECIAL_CMDEXT_CAN_WR_RUNTIME_VALUE,
        par, index, value, ref u0, ref u0, ref u0,
        dummy_buff, 0, dummy_buff, 0, dummy_buff, 0, ref u0) !=
        DT.RETURN_OK)
    {
        return DT.Error.DAPI_ERR_DEV_CONFIG_READ_ERROR;
    }
    return DT.Error.DAPI_ERR_NONE;
}
```

par = DAPI_SPECIAL_CMDEXT_CAN_RUNTIME_DEV_BAUDRATE

Mit diesem Befehl kann die Baudrate des Interface eingestellt/ausgelesen werden.

Baudrate	Value
1 MBit/s	0x00
500 KBit/s	0x01
250 KBit/s	0x02
125 KBit/s	0x03
100 KBit/s	0x04
50 KBit/s	0x05
20 KBit/s	0x06
10 KBit/s	0x07

Programmierbeispiel

```
RunTimeVarWriteToModule(handle,  
DAPI_SPECIAL_CMDEXT_CAN_RUNTIME_DEV_BAUDRATE, 0, 0x01);  
// Hier wird die Baudrate auf 500 KBit/s gesetzt.
```

```
uint val = 0;  
RunTimeVarReadFromModule(handle,  
DAPI_SPECIAL_CMDEXT_CAN_RUNTIME_DEV_BAUDRATE, 0, ref  
val);  
// Hier wird die Baudrate an die Variable val übergeben.
```

par = DAPI_SPECIAL_CMDEXT_CAN_RUNTIME_DEV_USEEXTID

Mit diesem Befehl kann der Bit-Mode eingestellt/ausgelesen werden.

useExtID	Value
11 Bit Mode	0x00
29 Bit Mode	0x01

Programmierbeispiel

```
RunTimeVarWriteToModule(handle,  
DAPI_SPECIAL_CMDEXT_CAN_RUNTIME_DEV_USEEXTID, 0, 0x00);  
// Hier wird die Ext-ID des Interfaces auf den 11 Bit Mode gesetzt.
```

```
uint val = 0;  
RunTimeVarReadFromModule(handle,  
DAPI_SPECIAL_CMDEXT_CAN_RUNTIME_DEV_USEEXTID, 0, ref  
val);  
// Hier wird der verwendete Bit Mode der Variable val übergeben.
```

par = DAPI_SPECIAL_CMDEXT_CAN_RUNTIME_TX_IS_ACTIVE

Mit diesem Befehl kann der Trigger-Mode eingestellt/ausgelesen werden.

Bei dem Nutzen des "Interval Mode (0x01)" kann zusätzlich über den Interval-Befehl eingestellt werden, in welchem Zeitintervall die TX-Pakete gesendet werden sollen.

Trigger Mode	Value
OFF	0x00
Interval Mode	0x01
RX-Event	0x02
Fast as possible	0x03

Programmierbeispiel

```
RunTimeVarWriteToModule(handle,  
DAPI_SPECIAL_CMDEXT_CAN_RUNTIME_TX_IS_ACTIVE, 1, 0x00);  
// Hier wird der Trigger Mode des TX-Paketes[1] auf OFF gesetzt.
```

```
uint val = 0;  
RunTimeVarReadFromModule(handle,  
DAPI_SPECIAL_CMDEXT_CAN_RUNTIME_TX_IS_ACTIVE, 0, ref  
val);  
// Hier wird der Trigger-Mode Status des TX-Paketes[0] der Variable val  
übergeben.
```

par = DAPI_SPECIAL_CMDEXT_CAN_RUNTIME_TX_INTERVAL

Mit diesem Befehl kann das Interval eingestellt/ausgelesen werden.

Interval count	Value Bit [4..7]
1	0x01
2	0x02
3	0x03
4	0x04
.. 9	.. 0x09

Interval unit	Value Bit [0..3]
* 1 ms	0x01
* 10 ms	0x02
* 100 ms	0x03
* 1 sec	0x04

Beispiel

Ein Interval von 700ms entspricht einem value von 0x73

Ein Interval von 40ms entspricht einem value von 0x42

Programmierbeispiel

```
RunTimeVarWriteToModule(handle,  
DAPI_SPECIAL_CMDEXT_CAN_RUNTIME_TX_INTERVAL, 1,  
(((0x02 << 4) & 0xf0) | (0x04 & 0x0f)));  
// Hier wird das Interval des TX-Paketes[1] auf 40ms eingestellt.
```

```
uint val = 0;  
RunTimeVarReadFromModule(handle,  
DAPI_SPECIAL_CMDEXT_CAN_RUNTIME_TX_INTERVAL, 0, ref  
val);  
// Hier wird das Interval des TX-Paketes[0] der Variable val übergeben.
```

par = DAPI_SPECIAL_CMDEXT_CAN_RUNTIME_TX_USE_EXT_ID

Mit diesem Befehl kann der Bit-Mode eingestellt/ausgelesen werden.

useExtID	Value
11 Bit Mode	0x00
29 Bit Mode	0x01

Programmierbeispiel

```
RunTimeVarWriteToModule(handle,  
DAPI_SPECIAL_CMDEXT_CAN_RUNTIME_TX_USE_EXT_ID, 1,  
0x00);  
// Hier wird die Ext-ID des TX-Paketes[1] auf den 11 Bit Mode gesetzt.
```

```
uint val = 0;  
RunTimeVarReadFromModule(handle,  
DAPI_SPECIAL_CMDEXT_CAN_RUNTIME_TX_USE_EXT_ID, 0, ref  
val);  
// Hier wird der verwendete Bit Mode des TX-Paketes[0] der Variable val  
übergeben.
```

par = DAPI_SPECIAL_CMDEXT_CAN_RUNTIME_TX_CANID

Mit diesem Befehl kann die CAN-ID eingestellt/ausgelesen werden.

Programmierbeispiel

```
RunTimeVarWriteToModule(handle,  
DAPI_SPECIAL_CMDEXT_CAN_RUNTIME_TX_CANID, 1, 0x1e);  
// Hier wird die CAN-ID des TX-Paketes[1] auf die 30 gesetzt.
```

```
uint val = 0;  
RunTimeVarReadFromModule(handle,  
DAPI_SPECIAL_CMDEXT_CAN_RUNTIME_TX_CANID, 0, ref val);  
// Hier wird der verwendete CAN-ID des TX-Paketes[0] der Variable val  
übergeben.
```

par = DAPI_SPECIAL_CMDEXT_CAN_RUNTIME_TX_MODE

Mit diesem Befehl kann der TX-Mode eingestellt/ausgelesen werden.

TX-Mode	Value
OPTO-IN 1-64	0x01
OPTO-IN 65-128	0x24
OPTO-IN 129-192	0x25
OPTO-IN 193-256	0x26
A/D CH 1-4 (16 Bit)	0x02
A/D CH 5-8 (16 Bit)",	0x03
A/D CH 9-12 (16 Bit)	0x04
A/D CH 13-16 (16 Bit)	0x05
A/D CH 17-20 (16 Bit)	0x06
A/D CH 21-24 (16 Bit)	0x07
A/D CH 25-28 (16 Bit)	0x08
A/D CH 29-32 (16 Bit)	0x09
Counter16 1-4 (16 Bit)	0x0a
Counter16 5-8 (16 Bit)	0x0b
Counter16 9-12 (16 Bit)	0x0c
Counter16 13-16 (16 Bit)	0x0d
Counter16 17-20 (16 Bit)	0x0e
Counter16 21-24 (16 Bit)	0x0f
Counter16 25-28 (16 Bit)	0x10

TX-Mode	Value
Counter16 29-32 (16 Bit)	0x11
Cnt48 1-2 (32 Bit)	0x12
Cnt48 3-4 (32 Bit)	0x13
Cnt48 5-6 (32 Bit)	0x14
Cnt48 7-8 (32 Bit)	0x15
PT-100 1-2 (32 Bit)	0x16
PT-100 3-4 (32 Bit)	0x17
PT-100 5-6 (32 Bit)	0x18
PT-100 7-8 (32 Bit)	0x19
Cnt48 1 (64 Bit)	0x1a
Cnt48 2 (64 Bit)	0x1b
Cnt48 3 (64 Bit)	0x1c
Cnt48 4 (64 Bit)	0x1d
Testcounter 8 bit	0x1e
DO Readback 1-64	0x1f
DO Readback 1-32	0x23
Custom1	0x20
Custom2	0x21
Custom3	0x22

Programmierbeispiel

```
RunTimeVarWriteToModule(handle,  
DAPI_SPECIAL_CMDEXT_CAN_RUNTIME_TX_MODE, 1, 0x0f);  
// Hier wird der Modus des TX-Paketes[1] auf den TX-Mode "Counter16 21-24  
(16 Bit)" gesetzt
```

```
uint val = 0;  
RunTimeVarReadFromModule(handle,  
DAPI_SPECIAL_CMDEXT_CAN_RUNTIME_TX_MODE, 0, ref val);  
// Hier wird der verwendete TX-Mode des TX-Paketes[0] der Variable val  
übergeben.
```

par = DAPI_SPECIAL_CMDEXT_CAN_RUNTIME_RX_IS_ACTIVE

Mit diesem Befehl wird das RX-Paket aktiviert/deaktiviert

Trigger Mode	Value
OFF	0x00
ON	0x01

Programmierbeispiel

```
RunTimeVarWriteToModule(handle,  
DAPI_SPECIAL_CMDEXT_CAN_RUNTIME_RX_IS_ACTIVE, 1, 0x00);  
// Hier wird das RX-Paket[1] auf OFF gesetzt.
```

```
uint val = 0;  
RunTimeVarReadFromModule(handle,  
DAPI_SPECIAL_CMDEXT_CAN_RUNTIME_RX_IS_ACTIVE, 0, ref  
val);  
// Hier wird der Status des RX-Paketes[0] der Variable val übergeben.
```

par = DAPI_SPECIAL_CMDEXT_CAN_RUNTIME_RX_USE_EXT_ID

Mit diesem Befehl kann der Bit-Mode eingestellt/ausgelesen werden.

UseExtID	Value
11 Bit Mode	0x00
29 Bit Mode	0x01

Programmierbeispiel

```
RunTimeVarWriteToModule(handle,  
DAPI_SPECIAL_CMDEXT_CAN_RUNTIME_RX_USE_EXT_ID, 1,  
0x00);  
// Hier wird die Ext-ID des RX-Paketes[1] auf den 11 Bit Mode gesetzt.
```

```
uint val = 0;  
RunTimeVarReadFromModule(handle,  
DAPI_SPECIAL_CMDEXT_CAN_RUNTIME_RX_USE_EXT_ID, 0, ref  
val);  
// Hier wird der verwendete Bit Mode des RX-Paketes[0] der Variable val  
übergeben.
```

par = DAPI_SPECIAL_CMDEXT_CAN_RUNTIME_RX_CANID

Mit diesem Befehl kann die CAN-ID eingestellt/ausgelesen werden.

Programmierbeispiel

```
RunTimeVarWriteToModule(handle,  
DAPI_SPECIAL_CMDEXT_CAN_RUNTIME_RX_CANID, 1, 0x1e);  
// Hier wird die CAN-ID des RX-Paketes[1] auf die 30 gesetzt.
```

```
uint val = 0;  
RunTimeVarReadFromModule(handle,  
DAPI_SPECIAL_CMDEXT_CAN_RUNTIME_RX_CANID, 0, ref val);  
// Hier wird der verwendete CAN-ID des RX-Paketes[0] der Variable val  
übergeben.
```

par = DAPI_SPECIAL_CMDEXT_CAN_RUNTIME_RX_MODE

Mit diesem Befehl kann RX-Mode eingestellt/ausgelesen werden.

RX-Mode	Value
Digital Out 1-64	0x01
D/A CH 1-4	0x02
D/A CH 5-8	0x03
D/A CH 9-12	0x04
D/A CH 13-16	0x05
D/A CH 17-20	0x06
D/A CH 21-24	0x07
D/A CH 25-28	0x08
D/A CH 29-32	0x09
Stepper No. 1	0x0a
Stepper No. 2	0x0b
Stepper No. 3	0x0c
Stepper No. 4	0x0d
Stepper No. 5	0x0e
Stepper No. 6	0x0f
Stepper No. 7	0x10
Stepper No. 8	0x11
D/A CH 1-4 (custom)	0x12
D/A CH 5-8 (custom)	0x13

RX-Mode	Value
D/A CH 9-12 (custom)	0x14
D/A CH 13-16 (custom)	0x15
D/A CH 17-20 (custom)	0x16
D/A CH 21-24 (custom)	0x17
D/A CH 25-28 (custom)	0x18
D/A CH 29-32 (custom)	0x19
Trigger Auto TX 1	0x1a
Trigger Auto TX 2	0x1b
Trigger Auto TX 3	0x1c
Trigger Auto TX 4	0x1d
Custom1	0x1e
Custom2	0x1f
Custom3	0x20
PWM CH 1-8	0x21
PWM CH 9-16	0x22
PWM CH 17-24	0x23
PWM CH 25-32	0x24
PWM CH 33-40	0x25
PWM CH 41-48	0x26
PWM CH 49-56	0x27

RX-Mode	Value
PWM CH 57-64	0x28
Trigger Auto TX 5	0x29
Trigger Auto TX 6	0x2a
Trigger Auto TX 7	0x2b
Trigger Auto TX 8	0x2c

Programmierbeispiel

```
RunTimeVarWriteToModule(handle,
DAPI_SPECIAL_CMDEXT_CAN_RUNTIME_RX_MODE, 1, 0x0f);
// Hier wird der Modus des RX-Paketes[1] auf den RX-Mode "Stepper No. 6"
gesetzt
```

```
uint val = 0;
RunTimeVarReadFromModule(handle,
DAPI_SPECIAL_CMDEXT_CAN_RUNTIME_RX_MODE, 0, ref val);
// Hier wird der verwendete RX-Mode des RX-Paketes[0] der Variable val
übergeben.
```

5.10. Testfunktionen

5.10.1. DapiPing

Beschreibung

Dieser Befehl prüft die Verbindung zu einem geöffneten Modul.

Definition

ULONG DapiPing(ULONG handle, ULONG value);

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

value=Übergebener Testwert, im Wertebereich von 0-255 (8-Bit), an das Modul

Return-Wert

Hier muß der mit "value" übergebene Testwert zurückkommen

5.11. Register Schreib-Befehle

5.11.1. DapiWriteByte

Beschreibung

Dieser Befehl führt einen direkten Register Schreibbefehl auf das Modul aus.

Definition

void DapiWriteByte(ULONG handle, ULONG adress, ULONG value);

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

adress=Adresse, auf die zugegriffen werden soll

value=Gibt den Datenwert an, der geschrieben wird (8 Bit)

Return-Wert

Keiner

Bemerkung

Dies sollte nur von erfahrenen Programmieren benutzt werden. So kann auf alle zur Verfügung stehenden Register direkt zugegriffen werden.

5.11.2. DapiWriteWord

Beschreibung

Dieser Befehl führt einen direkten Register Schreibbefehl auf das Modul aus.

Definition

```
void DapiWriteWord(ULONG handle, ULONG adress, ULONG value);
```

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

adress=Adresse, auf die zugegriffen werden soll

value=Gibt den Datenwert an, der geschrieben wird (16 Bit)

Return-Wert

Keiner

Bemerkung

Dies sollte nur von erfahrenen Programmieren benutzt werden. So kann auf alle zur Verfügung stehenden Register direkt zugegriffen werden.

5.11.3. DapiWriteLong

Beschreibung

Dieser Befehl führt einen direkten Register Schreibbefehl auf das Modul aus.

Definition

void DapiWriteLong(ULONG handle, ULONG adress, ULONG value);

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

adress=Adresse, auf die zugegriffen werden soll

value=Gibt den Datenwert an, der geschrieben wird (32 Bit)

Return-Wert

Keiner

Bemerkung

Dies sollte nur von erfahrenen Programmieren benutzt werden. So kann auf alle zur Verfügung stehenden Register direkt zugegriffen werden.

5.11.4. DapiWriteLongLong

Beschreibung

Dieser Befehl führt einen direkten Register Schreibbefehl auf das Modul aus.

Definition

void DapiWriteLongLong(ULONG handle, ULONG adress, ULONGLONG value);

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

adress=Adresse, auf die zugegriffen werden soll

value=Gibt den Datenwert an, der geschrieben wird (64 Bit)

Return-Wert

Keiner

Bemerkung

Dies sollte nur von erfahrenen Programmieren benutzt werden. So kann auf alle zur Verfügung stehenden Register direkt zugegriffen werden.

5.12. Register Lese-Befehle

5.12.1. DapiReadByte

Beschreibung

Dieser Befehl führt einen direkten Register Lese-Befehl auf das Modul aus.

Definition

ULONG DapiReadByte(ULONG handle, ULONG adress);

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

adress=Adresse, auf die zugegriffen werden soll

Return-Wert

Inhalt des zu lesenden Registers (8 Bit)

Bemerkung

Dies sollte nur von erfahrenen Programmieren benutzt werden. So kann auf alle zur Verfügung stehenden Register direkt zugegriffen werden.

5.12.2. DapiReadWord

Beschreibung

Dieser Befehl führt einen direkten Register Lese-Befehl auf das Modul aus.

Definition

ULONG DapiReadWord(ULONG handle, ULONG adress);

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

adress=Adresse, auf die zugegriffen werden soll

Return-Wert

Inhalt des zu lesenden Registers (16 Bit)

Bemerkung

Dies sollte nur von erfahrenen Programmieren benutzt werden. So kann auf alle zur Verfügung stehenden Register direkt zugegriffen werden.

5.12.3. DapiReadLong

Beschreibung

Dieser Befehl führt einen direkten Register Lese-Befehl auf das Modul aus.

Definition

ULONG DapiReadLong(ULONG handle, ULONG adress);

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

adress=Adresse, auf die zugegriffen werden soll

Return-Wert

Inhalt des zu lesenden Registers (32 Bit)

Bemerkung

Dies sollte nur von erfahrenen Programmieren benutzt werden. So kann auf alle zur Verfügung stehenden Register direkt zugegriffen werden.

5.12.4. DapiReadLongLong

Beschreibung

Dieser Befehl führt einen direkten Register Lese-Befehl auf das Modul aus.

Definition

ULONGLONG DapiReadLongLong(ULONG handle, ULONG adress);

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

adress=Adresse, auf die zugegriffen werden soll

Return-Wert

Inhalt des zu lesenden Registers (64 Bit)

Bemerkung

Dies sollte nur von erfahrenen Programmieren benutzt werden. So kann auf alle zur Verfügung stehenden Register direkt zugegriffen werden.

5.13. 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 ;
}

```

5.14. Delib Übersichtstabelle

Befehle	Verfügbar für
DAPI_SPECIAL_CMD_SET_DIR_DX_1	USB-MINI-TTL8
DAPI_SPECIAL_CMD_SET_DIR_DX_8	USB-MINI-TTL8 USB-TTL32 USB-TTL64 ETH-TTL64
DAPI_SPECIAL_CMD_GET_DIR_DX_1	wird nicht unterstützt
DAPI_SPECIAL_CMD_GET_DIR_DX_8	wird nicht unterstützt

Befehle	Verfügbar für	Geht nicht bei
DAPI_SPECIAL_CMD_TIMEOUT DAPI_SPECIAL_TIMEOUT_DO_VALUE_ MASK_WR_SET32 DAPI_SPECIAL_TIMEOUT_DO_VALUE_ MASK_RD_SET32 DAPI_SPECIAL_TIMEOUT_DO_VALUE_ MASK_WR_CLR32 DAPI_SPECIAL_TIMEOUT_DO_VALUE_ MASK_RD_CLR32 DAPI_SPECIAL_TIMEOUT_DO_VALUE_ LOAD_DEFAULT	ETH-TTL64 ETH-RELAIS8 USB-RELAIS8 RO-SERIE BS-SERIE NET-SERIE USB-TTL-64	USB-Mini-Stick
DAPI_SPECIAL_TIMEOUT_SET_VALUE_ SEC DAPI_SPECIAL_TIMEOUT_ACTIVATE DAPI_SPECIAL_TIMEOUT_DEACTIVATE DAPI_SPECIAL_TIMEOUT_GET_STATUS	alle Module	

Befehl	Starter USB*	Starter ETH**	RO Serie	BS Serie	NET Serie	Sonstiges
DAPI_SPECIAL_COUNTER_ LATCH_ALL			X			
DAPI_SPECIAL_COUNTER_ LATCH_ALL_WITH_RESET			X			
DapiDOSet1_WithTimer			X			
DAPI_SPECIAL_CMD_SW_FIFO DAPI_SPECIAL_SW_FIFO_INIT_ AND_CLEAR ... DAPI_SPECIAL_SW_FIFO_ IO_DEACTIVATE					X	
DAPI_SPECIAL_CMD_AD DAPI_SPECIAL_RO_AD_ FIFO_ACTIVATE ... DAPI_SPECIAL_RO_AD_ FIFO_INIT			X			

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

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

Befehl	Starter USB*	Starter ETH**	RO Serie	BS Serie	NET Serie	Sonstiges
DAPI_SPECIAL_DI_FF_FILTER DAPI_SPECIAL_DI_FF_FILTER_ VALUE_SET DAPI_SPECIAL_DI_FF_FILTER_ VALUE_GET	5-255	1-255	1-255	1-255	1-255	
DAPI_SPECIAL_DI_FILTER DAPI_SPECIAL_DI_FILTER_ VALUE_SET DAPI_SPECIAL_DI_FILTER_ VALUE_GET	x	0, 1-254	0, 1-254	0, 1-254	0, 1-254	
DAPI_SPECIAL_CMD_GET_ INTERNAL_STATISTIC	x	x	x	x		

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

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

Befehle	Verfügbar für
DAPI_SPECIAL_CMDEXT_CAN_WR_RUNTIME_ _VALUE DAPI_SPECIAL_CMDEXT_CAN_RD_RUNTIME_ VALUE	NET-CPU-PRO, BS-WEU, RO-ETH-LC

Anhang



6. Anhang

6.1. Kontakt / Support

Wenn Sie Fragen zum Produkt haben oder Unterstützung bei der Inbetriebnahme brauchen, erreichen Sie uns unter folgenden Rufnummern:

Support Software

Tel. +49 (0) 22 32 / 50 40 8 – 20

Support Hardware

Tel. +49 (0) 22 32 / 50 40 8 – 30

Support via E-mail

support@deditec.de

6.2. Umwelt und Entsorgung

Sie können das defekte oder veraltete Produkt am Ende seiner Lebensdauer wieder an uns zurück senden. Als Hersteller und Vertreiber von Elektronikbaugruppen übernehmen wir für Sie die fachgerechte Entsorgung nach den geltenden gesetzlichen Bestimmungen. Nutzen Sie hierfür am besten unser Rücksendeformular auf der Homepage:

[Rücksendeformular](#)

6.3. Revisionen

Rev 3.01 DEDITEC Design Update 2022

Rev 3.00 DEDITEC Design Update 2021

6.4. Urheberrechte und Marken

Linux ist eine registrierte Marke von Linus Torvalds.

USB ist eine registrierte Marke von USB Implementers Forum Inc.

LabVIEW ist eine registrierte Marke von National Instruments.

Intel ist eine registrierte Marke von Intel Corporation.

AMD ist eine registrierte Marke von Advanced Micro Devices, Inc.

ProfiLab ist eine registrierte Marke von ABACOM Ingenieurbüro GbR.

ispVM System ist eine registrierte Marke von Lattice Semiconductor Corporation

Windows, Visual-C/C++, -C#, -Basic, -Basic.NET und Visual-Studio sind registrierte Marken von Microsoft Corporation.

Delphi ist eine registrierte Marke von Borland Software Corporation.

Java ist eine registrierte Marke von Oracle Corporation.