



RO-SERIE

Hardware-Beschreibung

2023 September

INDEX

<u>1. Einleitung</u>	17
1.1. Vorwort	18
1.2. Kundenzufriedenheit	18
1.3. Kundenresonanz	18
1.4. Kurzbeschreibung	19
1.5. Lieferumfang	20
<u>2. Inbetriebnahme</u>	21
2.1. Schritt 1 - Sicherheitshinweise	22
2.2. Schritt 2 - Anschluss der Stromversorgung	23
2.3. Schritt 3 - Verbinden mit dem PC oder Netzwerk	24
2.3.1. Verbindung via USB	24
2.3.2. Verbindung via Ethernet	24
2.4. Schritt 4 - Installation der Software und Treiber	25
2.5. Schritt 5 - Anschluss der I/O Steckverbinder	26
2.6. Schritt 6 - Funktionstest	28
<u>3. Hardware Beschreibung</u>	29
3.1. Interfaces	30
3.1.1. RO-ETH	30
3.1.1.1. Hardware Beschreibung	30
3.1.1.1.1. Übersichtsbild	30
3.1.1.1.2. Technische Daten	31
3.1.1.1.3. Taster auf dem Modul	32
3.1.1.1.4. Kontroll LED's	33
3.1.1.1.4.1, Definition der LEDs	34
3.1.1.1.4.2, Blinkverhalten der LEDs	36
3.1.1.2. Konfiguration des Moduls	39
3.1.1.2.1. Konfiguration über das DELIB Configuration Utility	39
3.1.1.2.2. Konfiguration über den internen Web-Server des Moduls	39
3.1.1.2.3. Auslieferungszustand	40

INDEX

3.1.1.3. Grundkonfiguration wiederherstellen	41
3.1.1.3.1. IP Adresse zurücksetzen	41
3.1.1.3.2. Firmware zurücksetzen	42
3.1.2. RO-ETH/LC	43
3.1.2.1. Hardware Beschreibung	43
3.1.2.1.1. Übersichtsbild	43
3.1.2.1.2. Technische Daten	44
3.1.2.1.3. DIP-Schalter	45
3.1.2.1.4. Kontroll LED's	49
3.1.2.1.4.1, Definition der LEDs	49
3.1.2.1.4.2, Blinkverhalten der LEDs	51
3.1.2.2. Konfiguration des Moduls	54
3.1.2.2.1. Konfiguration über das DELIB Configuration Utility	54
3.1.3. RO-USB	55
3.1.3.1. Hardware Beschreibung	55
3.1.3.1.1. Übersichtsbild	55
3.1.3.1.2. Technische Daten	59
3.1.3.1.3. Kontroll LED's	60
3.1.3.1.3.1, Definition der LEDs	61
3.1.3.1.4. Konfiguration über das DELIB Configuration Utility	61
3.1.4. RO-CAN	62
3.1.4.1. Hardware Beschreibung	62
3.1.4.1.1. Übersichtsbild	62
3.1.4.1.2. Technische Daten	65
3.1.4.1.3. Kontroll LED's	66
3.1.4.1.3.1, Definition der LEDs	67
3.1.4.2. Konfiguration des Moduls	69
3.1.4.2.1. DIP-Schalter	70
3.1.4.2.2. Der Vorzugsmodus	72
3.1.4.2.3. Software-Modus	72
3.1.4.2.4. DIP-Schalter-Modus	73
3.1.4.2.4.1, Einstellen der Übertragungsgeschwindigkeit (Bitrate)	73
3.1.4.2.4.2, Einstellen der CAN-Moduladresse	74

INDEX

3.1.5. RO-SER	77
3.1.5.1. Hardware Beschreibung	77
3.1.5.1.1. Übersichtsbild	77
3.1.5.1.2. Technische Daten	81
3.1.5.1.3. Kontroll LED's	82
3.1.5.1.3.1, Definition der LEDs	83
3.1.5.2. Konfiguration des Moduls	84
3.1.5.2.1. DIP-Schalter	84
3.1.5.2.2. Der Vorzugsmodus	86
3.1.5.2.3. DIP-Schalter-Modus	86
3.1.5.2.4. Serieller Textmodus	87
3.1.5.2.4.1, Übersicht Schreibbefehle	88
3.1.5.2.4.2, Übersicht Lesebefehle	90
3.1.5.2.5. Echo aktivieren	92
3.1.5.2.6. Einstellen der Baudrate	92
3.1.5.2.7. Einstellen der Moduladresse (nur bei RS-485)	93
3.1.5.2.8. Konfiguration über das DELIB Configuration Utility	94
3.1.6. Schnittstellen	95
3.1.6.1. USB	95
3.1.6.2. Ethernet	95
3.1.6.3. CAN	96
3.1.6.4. RS-232	97
3.1.6.5. RS-485	98
3.1.6.6. Wählen der Schnittstellenvariante RS-232 / RS-485 / CAN	99
3.2. Submodule	100
3.2.1. RO-REL16	100
3.2.1.1. Übersichtsbild	100
3.2.2. RO-REL8_UM	102
3.2.2.1. Übersichtsbild	102
3.2.2.2. Bistabile Relais Ausgänge	103
3.2.3. RO-OPT8-REL8	104
3.2.3.1. Übersichtsbild	104

INDEX

3.2.4. RO-OPT16	106
3.2.4.1. Übersichtsbild	106
3.2.4.2. Ändern der Eingangsspannung	108
3.2.5. RO-MOS16	109
3.2.5.1. Übersichtsbild	109
3.2.6. RO-PWM	111
3.2.6.1. Übersichtsbild	111
3.2.7. Digitale Ausgänge	113
3.2.7.1. Technische Daten	113
3.2.7.2. Pinbelegung eines RO-REL16	114
3.2.7.3. Pinbelegung eines RO-REL8_UM	115
3.2.7.4. Pinbelegung eines RO-MOS16_P und RO-PWM16_P	116
3.2.7.5. Anschlussbeispiel eines Relais-Moduls	118
3.2.7.6. Anschlussbeispiel eines Umschaltrelais-Moduls	119
3.2.7.7. Anschlussbeispiel eines MOSFET-Moduls	120
3.2.7.8. Timeout Funktion	121
3.2.8. Digitale Eingänge	122
3.2.8.1. Technische Daten	122
3.2.8.2. Pinbelegung RO-O16	123
3.2.8.3. Pinbelegung RO-O8-R8	124
3.2.8.4. Anschlussbeispiel RO-O16	126
3.2.8.5. Anschlussbeispiel RO-O8-R8	127
3.2.8.6. Eingangsfilter	129
3.2.8.7. Zustandsänderungen überwachen	130
3.2.9. Analoge Ein-/Ausgabemodule	131
3.2.9.1. RO-AD16-DA4	131
3.2.9.1.1. Übersichtsbild	132
3.2.9.1.2. Technische Daten	134
3.2.9.1.3. Pinbelegung	135
3.2.9.1.3.1, A/D Steckverbinder (18pol)	135
3.2.9.1.3.2, D/A Steckverbinder (10pol)	136
3.2.9.2. RO-AD16	137
3.2.9.2.1. Übersichtsbild	137

INDEX

3.2.9.2.2. Technische Daten	139
3.2.9.2.3. Pinbelegung	140
3.2.9.2.3.1, A/D Steckverbinder (18pol)	140
3.2.9.3. RO-AD16_ISO	141
3.2.9.3.1. Übersichtsbild	141
3.2.9.3.2. Technische Daten	143
3.2.9.3.3. Pinbelegung	144
3.2.9.3.3.1, A/D Steckverbinder (18pol)	144
3.2.9.4. RO-DA4/RO-DA8	145
3.2.9.4.1. Übersichtsbild	145
3.2.9.4.2. Technische Daten	146
3.2.9.4.3. Pinbelegung	147
3.2.9.4.3.1, D/A Steckverbinder (10pol)	147
3.2.9.5. RO-DA2_ISO	148
3.2.9.5.1. Übersichtsbild	148
3.2.9.5.2. Technische Daten	150
3.2.9.5.3. Pinbelegung	151
3.2.9.5.3.1, D/A Steckverbinder (10pol)	151
3.2.10. Stepper Modul	153
3.2.10.1. Übersichtsbild	153
3.2.10.2. Technische Daten	155
3.2.10.3. Ansteuerung	155
3.2.10.4. Pinbelegung	156
3.2.10.4.1. Stepper Steckverbinder (10pol)	156
3.2.11. RO-CNT8	157
3.2.11.1. Übersichtsbild	157
3.2.11.2. Technische Daten	159
3.2.11.3. Counter Modul Modi	160
3.2.11.4. Blockschaltbild	161
3.2.11.5. Pinbelegung	162
3.2.11.5.1. Pinbelegung RO-CNT8	162
3.2.12. RO-CNT/IGR	163
3.2.12.1. Übersichtsbild	163

INDEX

3.2.12.2. Technische Daten	165
3.2.12.3. Blockschaltbild	166
3.2.12.4. Auswertung Phi 0/Phi 90° (Grafisch)	167
3.2.12.5. LEDs	168
3.2.12.6. Eingänge	169
3.2.12.6.1. Referenzieren	169
3.2.12.6.2. Visuelle Kontrolle der Eingänge	169
3.2.12.7. Pinbelegung	170
3.2.13. RO-PT100-4 / RO-PT100-8	171
3.2.13.1. Übersichtsbild	171
3.2.13.2. Einführung	173
3.2.13.3. Technische Daten	174
3.2.13.4. Blockschaltbild	175
3.2.13.5. Pinbelegung	176
3.3. Abmessung der RO-Serie	178
3.3.1. Draufsicht	178
3.3.2. Seitenansicht	179
4. Software Beschreibung	180
4.1. Benutzen unserer Produkte	181
4.1.1. Ansteuerung über unsere DELIB Treiberbibliothek	181
4.1.2. Ansteuerung über mitgelieferte Testprogramme	181
4.1.3. Ansteuerung auf Protokollebene	182
4.1.4. DELIB CLI (command-line interface) für Windows	183
4.1.4.1. Konfiguration des DELIB CLI	185
4.1.4.2. DELIB CLI Beispiele	186
4.1.5. Ansteuerung über grafische Anwendungen	190
4.1.5.1. LabVIEW	190
4.1.5.2. ProfiLab	190
4.1.5.3. Licht24 Pro	191
4.1.6. Einbinden der DELIB in Programmiersprachen	192
4.1.6.1. Einbinden der DELIB in Visual-C/C++	192

INDEX

4.1.6.2. Einbinden der DELIB in Visual-C/C++ (Visual Studio 2015)	194
4.1.6.3. Einbinden der DELIB in Visual-C#	197
4.1.6.4. Einbinden der DELIB in Delphi	198
4.1.6.5. Einbinden der DELIB in Visual-Basic (VB)	199
4.1.6.6. Einbinden der DELIB in Visual-Basic.NET (VB.NET)	200
4.1.6.7. Einbinden der DELIB in MS-Office (VBA)	201
4.1.6.8. Einbinden der DELIB in LabVIEW	203
4.1.6.8.1. Einbinden der DELIB in LabVIEW	203
4.1.6.8.2. Verwendung der VIs in LabVIEW	212
4.1.6.8.3. Setzen der Modul-ID in LabVIEW	214
4.1.6.9. Einbinden der DELIB in Java	216
4.2. DELIB Treiberbibliothek	217
4.2.1. Übersicht	218
4.2.1.1. Unterstützte Programmiersprachen	221
4.2.1.2. Unterstützte Betriebssysteme	222
4.2.1.3. SDK-Kit für Programmierer	222
4.2.2. DELIB Setup	223
4.2.3. DELIB Configuration Utility	229
4.2.3.1. Einführung	229
4.2.3.2. Neue Konfig erstellen oder vorhandene Konfig bearbeiten	230
4.2.3.2.1. Modul Konfiguration USB	231
4.2.3.2.1.1, Beispiel zur Konfiguration identischer USB-Module	233
4.2.3.2.2. Modul Konfiguration Ethernet	238
4.2.3.2.2.1, Automatische Suche	243
4.2.3.2.2.2, Verschlüsselung einrichten	247
4.2.3.2.2.1, Manuelle Konfiguration	247
4.2.3.2.2.2, Automatische Konfiguration	252
4.2.3.2.2.1, Authentifizierung	253
4.2.3.2.3. Modul Konfiguration CAN	257
4.2.3.2.4. Modul Konfiguration Seriell	260
4.2.3.3. Modul testen	262

INDEX

4.2.3.4. Debug Optionen einstellen	266
4.2.4. Benutzung des Modulselectors	267
4.2.4.1. via USB	267
4.2.4.2. via Ethernet	268
4.2.4.3. via WiFi / WPS	270
4.2.4.4. Modul Info	271
4.2.5. DELIB Module Config	273
4.2.5.1. Modul Konfigurationen	273
4.2.5.1.1. Modul-Infoseite	274
4.2.5.1.2. Modul-Identifikation	275
4.2.5.1.3. LAN Netzwerkinformationen	276
4.2.5.1.4. LAN Netzwerkeinstellungen	277
4.2.5.1.5. WiFi Netzwerkinformationen	279
4.2.5.1.6. WiFi Netzwerkeinstellungen	281
4.2.5.1.7. WiFi WPS-Verbindung	283
4.2.5.1.8. NTP-Konfiguration	284
4.2.5.1.9. Serielle Konfiguration	286
4.2.5.1.10. I/O Kanal-Namen	288
4.2.5.1.11. CAN Konfiguration	289
4.2.5.1.11.1, CAN Status	289
4.2.5.1.11.1, CAN Status Interface	289
4.2.5.1.11.2, CAN Statistik TX/RX	290
4.2.5.1.11.2, CAN Main	291
4.2.5.1.11.1, CAN Main Interface	291
4.2.5.1.11.2, CAN Main I/O Init	293
4.2.5.1.11.3, CAN TX/RX Modi	295
4.2.5.1.11.1, CAN TX Mode	295
4.2.5.1.11.2, CAN RX Mode	297
4.2.5.2. I/O-Test	298
4.2.5.2.1. Timeout Test-Funktion	298
4.2.5.2.2. Digital In	299
4.2.5.2.3. Digital Out	300
4.2.5.2.4. Analog In	302

INDEX

4.2.5.2.5. Analog Out	303
4.2.6. DELIB Module Demo	305
4.2.6.1. Auswahl des Moduls	306
4.2.6.2. Allgemein	307
4.2.6.2.1. Module Info	309
4.2.6.3. Digital Input	310
4.2.6.4. Digital Output	311
4.2.6.5. Analog Input	312
4.2.6.6. Analog Output	313
4.2.7. CAN Configuration Utility	314
4.2.7.1. Auswahl des Moduls	316
4.2.7.2. Neue Konfiguration Erstellen, Laden, Speichern	318
4.2.7.3. Konfiguration auf das Modul übertragen	320
4.2.7.4. Statistiken vom Modul abfragen	321
4.2.7.5. Konfiguration	324
4.2.7.5.1. Modul Konfiguration	325
4.2.7.5.2. I/O Konfiguration	326
4.2.7.5.3. TX-Konfiguration	329
4.2.7.5.3.1, Beispiel Interval	331
4.2.7.5.3.2, Beispiel Trigger	332
4.2.7.5.4. RX-Konfiguration	333
4.2.7.5.4.1, Beispiel RX-DA	335
4.2.7.5.4.2, Beispiel RX-DO	336
4.2.7.6. Aufbau der CAN-Pakete	337
4.2.7.6.1. Digitale Eingänge	337
4.2.7.6.2. Digitale Ausgänge	338
4.2.7.6.3. Digitale Eingangszähler (16-Bit)	339
4.2.7.6.4. Digitale Eingangszähler (48-Bit) - 32-Bit Paket	340
4.2.7.6.5. Digitale Eingangszähler (48-Bit) - 64-Bit Paket	341
4.2.7.6.6. Analoge Ein- / Ausgänge	342
4.2.7.6.6.1, Analoge Eingänge	342
4.2.7.6.6.2, Analoge Ausgänge	344
4.2.7.6.6.3, Beispiele	345

INDEX

4.2.7.6.7. Temperatur Eingänge	348
4.2.7.6.8. Stepper	350
4.2.7.6.8.1, Command-Liste	350
4.2.7.6.8.2, Werte für par 1 zu Befehl SET_MOTORCHARACTERISTIC	353
4.2.7.6.8.3, Werte für par 1 zu Befehl GO_REFSWITCH	356
4.2.7.6.8.4, Beispiel	356
4.2.8. DT-Flasher	357
4.2.8.1. Über DEDITEC-Firmware	358
4.2.8.2. Auswahl des Moduls	358
4.2.8.3. Firmware Update durchführen	359
4.2.8.3.1. Flash-Files manuell aktualisieren	361
4.3. DELIB Sample Sources (Windows Programmbeispiele)	362
4.3.1. Installation DELIB Sample Sources	363
4.3.2. Benutzung der DELIB Sample Sources	367
4.3.2.1. Schritt 1 - Produktauswahl	368
4.3.2.2. Schritt 2 - Kategorieauswahl	369
4.3.2.3. Schritt 3 - Programmiersprachenauswahl	370
4.3.2.4. Schritt 4 - Quellcode	371
4.4. DELIB für Linux	374
4.4.1. DELIB Sample Sources (Windows Programmbeispiele) 2	377
4.4.1.1. Installation DELIB Sample Sources 2	378
4.4.1.2. Benutzung der DELIB Sample Sources 2	382
4.4.1.2.1. Schritt 1 - Produktauswahl 2	383
4.4.1.2.2. Schritt 2 - Kategorieauswahl 2	384
4.4.1.2.3. Schritt 3 - Programmiersprachenauswahl 2	385
4.4.1.2.4. Schritt 4 - Quellcode 2	386
4.4.2. Verwenden der DELIB Treiberbibliothek für Linux	389
4.4.2.1. Delib USB-Sample in Linux	389
4.4.2.2. Delib ETH-Sample in Linux	392
4.4.3. DELIB CLI (command-line interface) für Linux	396
4.4.3.1. Konfiguration des DELIB CLI	399
4.4.3.2. DELIB CLI Beispiele	402

INDEX

4.5. Weboberfläche	403
4.5.1. Konfiguration	405
4.5.1.1. Allgemein	405
4.5.1.2. Netzwerk Konfiguration	406
4.5.1.3. Benutzer Manager	407
4.5.1.4. Status	410
4.5.1.5. Sicherheit	411
4.5.2. Ein-/Ausgänge	413
4.5.2.1. Allgemein	413
4.5.2.2. Digitale Eingänge	414
4.5.2.3. Digitale Eingänge Zähler	416
4.5.2.4. Digitale Ausgänge	418
4.5.2.5. Analoge Eingänge	420
4.5.2.6. Analoge Ausgänge	421
4.5.2.7. Konfiguration	422
<u>5. Verzeichnisstruktur der DELIB</u>	425
5.1. Include Verzeichnis	428
5.2. Library-Verzeichnis	428
5.3. Library-Verzeichnis für Borland	428
5.4. Umgebungsvariablen	428
<u>6. DELIB API Referenz</u>	429
6.1. Verwaltungsfunktionen	430
6.1.1. DapiOpenModule	430
6.1.2. DapiCloseModule	431
6.1.3. DapiGetDELIBVersion	431
6.1.4. DapiSpecialCMDGetModuleConfig	432
6.1.5. DapiOpenModuleEx	435
6.1.6. DapiScanAllModulesAvailable	437
6.2. Fehlerbehandlung	438
6.2.1. DapiGetLastError	438

INDEX

6.2.2. DapiClearLastError	439
6.2.3. DapiGetLastErrorByHandle	440
6.2.4. DapiClearLastErrorByHandle	441
6.2.5. DapiGetLastErrorText	442
6.3. Digitale Eingänge lesen	443
6.3.1. DapiDIGet1	443
6.3.2. DapiDIGet8	443
6.3.3. DapiDIGet16	444
6.3.4. DapiDIGet32	445
6.3.5. DapiDIGet64	446
6.3.6. DapiDIGetFF32	447
6.3.7. DapiDIGetCounter	448
6.3.8. DapiSpecialCounterLatchAll	449
6.3.9. DapiSpecialCounterLatchAllWithReset	450
6.3.10. DapiSpecialDIFilterValueSet	451
6.3.11. DapiSpecialDIFilterValueGet	452
6.3.12. Dapi_Special_DI_FF_Filter_Value_Set	453
6.3.13. Dapi_Special_DI_FF_Filter_Value_Get	454
6.4. Digitale Ausgänge verwalten	455
6.4.1. DapiDOSet1	455
6.4.2. DapiDOSet8	455
6.4.3. DapiDOSet16	456
6.4.4. DapiDOSet32	457
6.4.5. DapiDOSet64	458
6.4.6. DapiDOSet1_WithTimer	459
6.4.7. DapiDOReadback32	460
6.4.8. DapiDOReadback64	460
6.4.9. DapiDOSetBit32	461
6.4.10. DapiDOClrBit32	462
6.5. Digitale Zähler Funktionen	463
6.5.1. DapiCnt48ModeSet	463

INDEX

6.5.2. DapiCnt48ModeGet	469
6.5.3. DapiCnt48CounterGet32	470
6.5.4. DapiCnt48CounterGet48	471
6.5.5. DapiSpecialCNT48ResetSingle	472
6.5.6. DapiSpecialCNT48ResetGroup8	473
6.5.7. DapiSpecialCNT48LatchGroup8	474
6.5.8. DapiSpecialCNT48DIGet1	475
6.6. PWM Funktionen	476
6.6.1. DapiPWMOutSet	476
6.6.2. DapiPWMOutReadback	478
6.6.3. DAPI_SPECIAL_PWM_FREQ_SET	479
6.6.4. DAPI_SPECIAL_PWM_FREQ_READBACK	480
6.7. Pulsgenerator Ausgänge verwalten	481
6.7.1. DapiPulseGenSet	481
6.8. A/D Wandler Funktionen	483
6.8.1. DapiADSetMode	483
6.8.2. DapiADGetMode	485
6.8.3. DapiADGet	485
6.8.4. DapiADGetVolt	486
6.8.5. DapiADGetmA	486
6.8.6. DapiSpecialADReadMultipleAD	487
6.9. D/A Ausgänge verwalten	489
6.9.1. DapiDASetMode	489
6.9.2. DapiDAGetMode	491
6.9.3. DapiDASet	492
6.9.4. DapiDASetVolt	493
6.9.5. DapiDASetmA	494
6.9.6. DapiSpecialCmd_DA	495
6.10. PT100 Funktionen	497
6.10.1. DapiTempGet	497
6.11. Schrittmotoren Funktionen	498

INDEX

6.11.1. Befehle mit DapiStepperCommand	498
6.11.1.1. DAPI_STEPPER_CMD_GO_POSITION	498
6.11.1.2. DAPI_STEPPER_CMD_GO_POSITION_RELATIVE	499
6.11.1.3. DAPI_STEPPER_CMD_SET_POSITION	500
6.11.1.4. DAPI_STEPPER_CMD_SET_FREQUENCY	501
6.11.1.5. DAPI_STEPPER_CMD_GET_FREQUENCY	502
6.11.1.6. DAPI_STEPPER_CMD_SET_FREQUENCY_DIRECTLY	503
6.11.1.7. DAPI_STEPPER_CMD_STOP	504
6.11.1.8. DAPI_STEPPER_CMD_FULLSTOP	505
6.11.1.9. DAPI_STEPPER_CMD_DISABLE	506
6.11.1.10. DAPI_STEPPER_CMD_SET_MOTORCHARACTERISTIC	507
6.11.1.11. DAPI_STEPPER_CMD_GET_MOTORCHARACTERISTIC	514
6.11.1.12. DAPI_STEPPER_CMD_MOTORCHARACTERISTIC_EEPROM_SAVE	523
6.11.1.13. DAPI_STEPPER_CMD_MOTORCHARACTERISTIC_EEPROM_LOAD	524
6.11.1.14. DAPI_STEPPER_CMD_MOTORCHARACTERISTIC_LOAD_DEFAULT	525
6.11.1.15. DAPI_STEPPER_CMD_GO_REFSWITCH	526
6.11.1.16. DAPI_STEPPER_CMD_GET_CPU_TEMP	528
6.11.1.17. DAPI_STEPPER_CMD_GET_MOTOR_SUPPLY_VOLTAGE	529
6.11.2. Status abfragen mit DapiStepperGetStatus	530
6.11.2.1. DAPI_STEPPER_STATUS_GET_ACTIVITY	530
6.11.2.2. DAPI_STEPPER_STATUS_GET_POSITION	531
6.11.2.3. DAPI_STEPPER_STATUS_GET_SWITCH	532
6.11.3. DapiStepperCommandEx	533
6.12. Ausgabe Timeout-Schutz-Funktion verwalten	534
6.12.1. DapiSpecialCMDTimeout	534
6.12.1.1. DapiSpecialTimeoutSetValueSec	537

INDEX

6.12.1.2. DapiSpecialTimeoutActivate	538
6.12.1.3. DapiSpecialTimeoutActivateAutoReactivate	539
6.12.1.4. DapiSpecialTimeoutActivateSecureOutputs	540
6.12.1.5. DapiSpecialTimeoutDeactivate	541
6.12.1.6. DapiSpecialTimeoutGetStatus	542
6.12.1.7. DapiSpecialTimeoutDoValueMaskWRSet32	543
6.12.1.8. DapiSpecialTimeoutDoValueMaskRDSet32	544
6.12.1.9. DapiSpecialTimeoutDoValueMaskWRClr32	545
6.12.1.10. DapiSpecialTimeoutDoValueMaskRDClr32	546
6.12.1.11. DapiSpecialTimeoutDoValueLoadDefault	547
6.13. Testfunktionen	548
6.13.1. DapiPing	548
6.14. Register Schreib-Befehle	549
6.14.1. DapiWriteByte	549
6.14.2. DapiWriteWord	550
6.14.3. DapiWriteLong	551
6.14.4. DapiWriteLongLong	552
6.15. Register Lese-Befehle	553
6.15.1. DapiReadByte	553
6.15.2. DapiReadWord	554
6.15.3. DapiReadLong	555
6.15.4. DapiReadLongLong	556
6.16. DELIB Übersichtstabelle	557
<u>7. Anhang</u>	560
7.1. Kontakt / Support	561
7.2. Umwelt und Entsorgung	561
7.3. Revisionen	562
7.4. Urheberrechte und Marken	564

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 Produkte der RO-Serie sind vielseitig einsetzbar. Durch ein Stecksystem können Submodule individuell verbunden werden, somit wird der Einsatz dieser Module noch flexibler. 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 bspw. unsere DELIB API an, oder eine direkte Kommunikation über unser Ethernet-Protokoll.

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

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



1.5. Lieferumfang

Folgende Artikel sind im Lieferumfang enthalten:

- RO-Modul
- 2 pol. Steckverbinder für den Stromanschluss
- 10, 16 oder 18 pol. 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

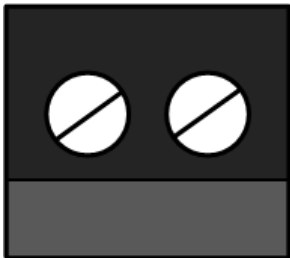
Bitte machen Sie sich vor der Inbetriebnahme Ihres DEDITEC Produktes mit diesem Handbuch vertraut und lesen Sie sich die nachfolgenden Punkte genau durch:

- 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 Leiterplatine. 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**

2.4. 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>

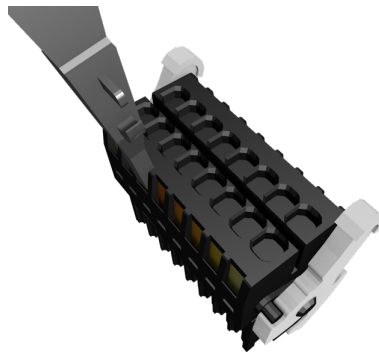
2.5. Schritt 5 - Anschluss der I/O Steckverbinder

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

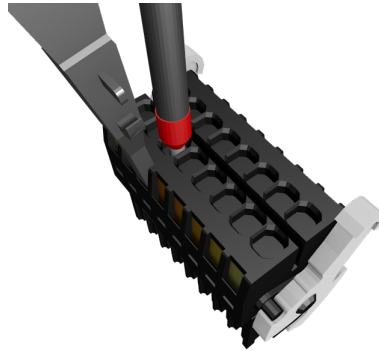


Gehen Sie zum Anschluss bitte wie folgt vor:

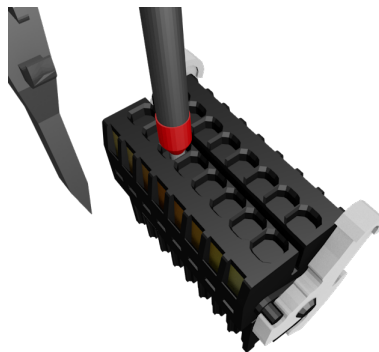
- Betätigungswerkzeug in Leiteranschlussrichtung kräftig (nach unten) in die seitliche Öffnung stecken.



- 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

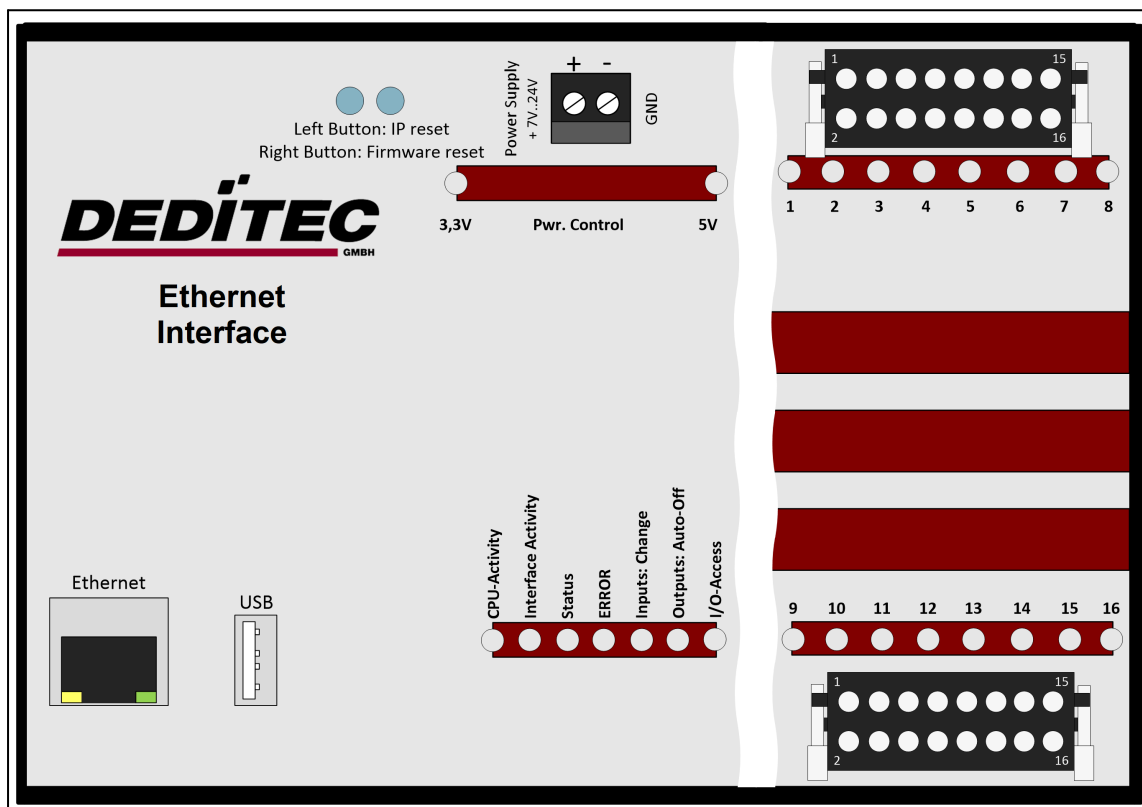
3.1. Interfaces

3.1.1. RO-ETH

3.1.1.1. Hardware Beschreibung

3.1.1.1.1. Übersichtsbild

Die Abbildung zeigt das Steuermodul mit Ethernet Interface (links) in Kombination mit einem Ein- /Ausgabemodul (rechts).



3.1.1.1.2. Technische Daten

Elektrische Daten:

Versorgungsspannung: 7V DC .. 24V DC

Leistungsaufnahme: max. 5W

Umgebung:

Umgebungstemperatur: +10..+50 °C

Luftfeuchtigkeit: 90 %

Betauung: Nicht erlaubt

Mechanik:

Abmessungen in mm (LxBxH): L* x 122 x 51,5
*die Länge variiert mit der Art und Anzahl angeschlossener Submodule

Befestigung: Hutschiene TS 35 x 7,5 mm

Optional:

galv. Trennung: als Zubehör erhältlich

3.1.1.1.3. Taster auf dem Modul

Linker Taster:

IP Adresse auf Default Werte zurücksetzen
(siehe Kapitel "IP Adresse zurücksetzen")

Rechter Taster:

Firmware in den Auslieferungszustand zurücksetzen
(siehe Kapitel "Firmware zurücksetzen")

3.1.1.1.4. Kontroll LED's

Auf dem RO-ETH Modul befinden sich eine Reihe von Kontroll LED's. Sie dienen zur einfachen optischen Zustandsanzeige von diversen Funktionen.

Nach dem Einschalten des Moduls muss folgende Blinksequenz im Normalbetrieb auftreten:

- ca. 20 Sek nach Einschalten des Moduls blinken LED 1 und 2 zweimal kurz hintereinander auf. -> Betriebssystem erfolgreich geladen.
- Anschließend leuchtet die 3. LED dauerhaft und LED 1 blinkt. -> Modul ist betriebsbereit.

3.1.1.1.4.1. Definition der LEDs

LED	Bezeichnung	Erklärung
oben	3,3V	Interne 3,3V Versorgungsspannung vorhanden.
oben	5V	Interne 5V Versorgungsspannung vorhanden.
1	CPU Activity	2x blinken + lange Pause. Betriebssystem meldet: Status OK
2	Interface Activity	Kommunikation über Ethernet aktiv.
3	Status	LED leuchtet -> Modul ist betriebsbereit
4	ERROR	Fehler bei der ETH Übertragung (näheres siehe Dokument "ETH Protokoll").
5	Inputs: Change	Zustandswechsel zwischen 2 Auslesetakten wurde erfasst.
6	Outputs: Auto-Off	Sämtliche Ausgänge wurden auf Grund des Timeout sicherheitshalber abgeschaltet.
7	I/O Access	Zugriff der CPU auf Ein- und Ausgänge der

LED	Bezeichnung	Erklärung
		angeschlossenen Module.

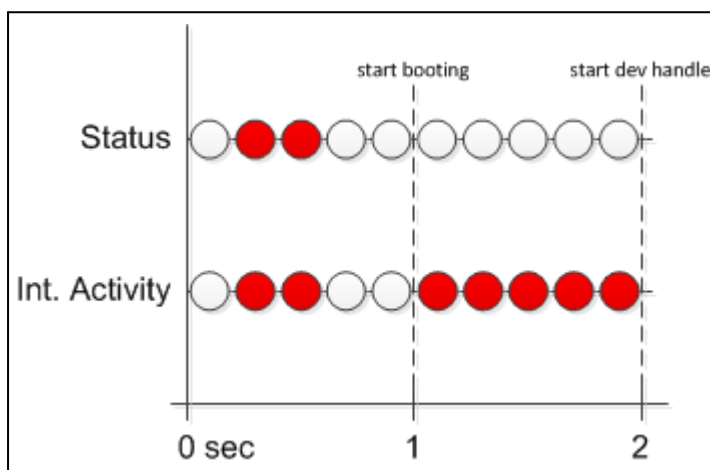
3.1.1.1.4.2. Blinkverhalten der LEDs

Nachfolgend werden die Blinksequenzen der Status LEDs dargestellt.



1. Bootvorgang

Der Bootvorgang startet direkt nach dem Einschalten der Spannungsversorgung.

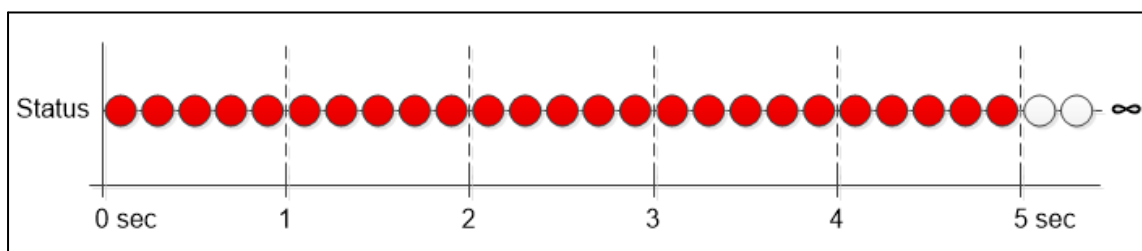


Die Bootvorgang-Sequenz wird einmalig durchlaufen.

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.



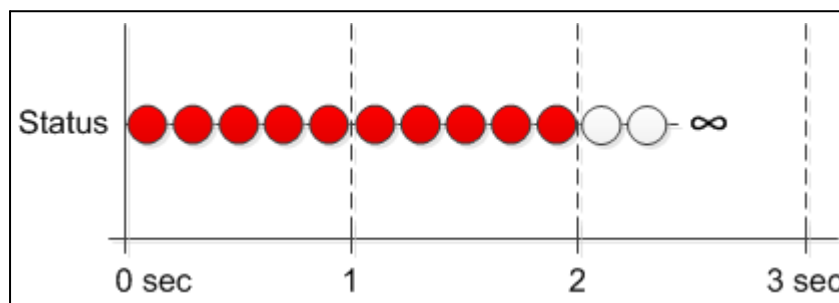
Die Status-LED leuchtet 5 Sekunden und erlischt für etwa 300ms. Die Applikations-Sequenz wiederholt sich.

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.

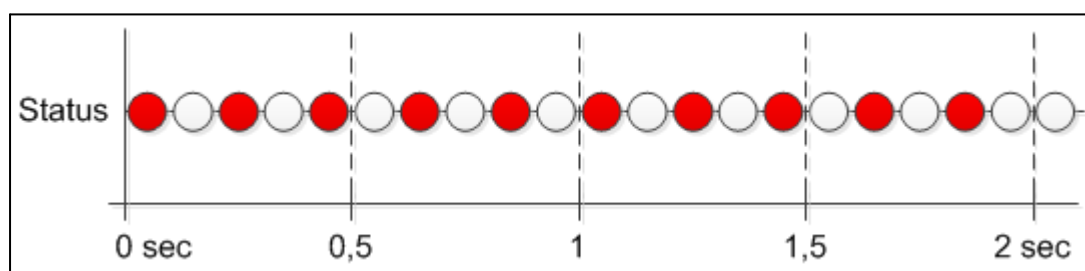


3. Bootloader erzwungen

Das Produkt kann mit dem DIP-Schalter 4 in den Bootloader gezwungen werden.

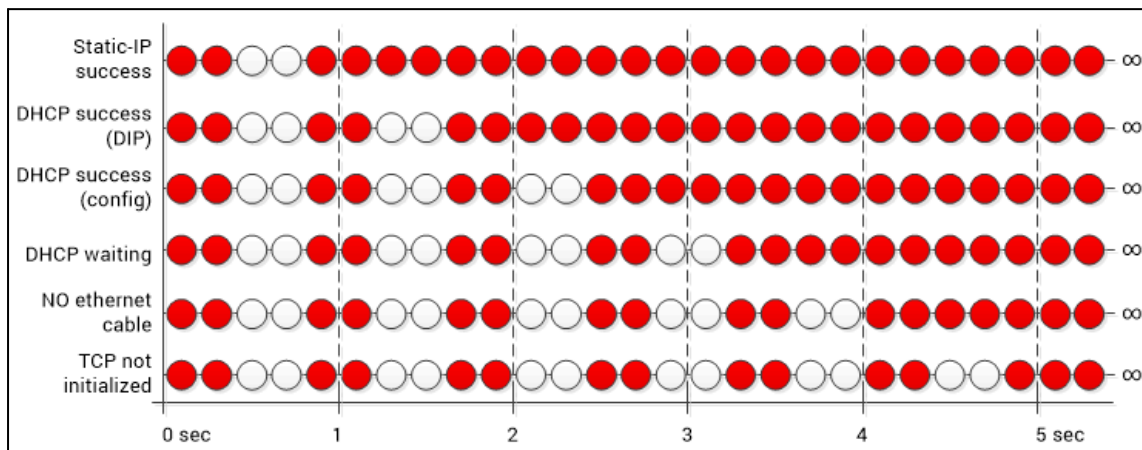
→ **siehe Kapitel "DIP Schalter"**

Diese Funktion dient nur zu Service-Zwecken.



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

4. I/F.Status LED



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.

3.1.1.2. Konfiguration des Moduls

3.1.1.2.1. Konfiguration über das DELIB Configuration Utility

Die Konfiguration ist im Kapitel **Konfiguration Ethernet-Schnittstelle** beschrieben.

3.1.1.2.2. Konfiguration über den internen Web-Server des Moduls

Das RO-ETH-Modul hat einen eigenen Web-Server über den die Konfiguration ebenfalls vorgenommen werden kann.

The screenshot shows the DEDITEC web configuration interface. At the top, there is a header with the DEDITEC logo and contact information: "Infohotline + (49) 0 22 32 50 40 80". Below the header, there are navigation tabs: "Configuration", "In-/Outputs", "Custom", and "Logout". The "Configuration" tab is selected. Below the tabs, there is a status bar showing "Product name: RO-ETH (DEDITEC RO-ETH Module)" and "User: gast / Session ends in: 684". On the left side, there is a sidebar menu with options: "General", "Network configuration", "Network time (NTP)", "HTTP-Server", "Mail-Server", "User-Manager", "FW-Update", "Log's", "Status / Reboot", and "Security". The "Network configuration" option is selected. The main content area is titled "Network configuration" and contains several input fields for network settings: "IP-address:" (192.168.1.1), "Netmask:" (255.255.255.0), "Std-GW:" (192.168.1.254), "DNS:" (192.168.1.254), and "TCP port:" (9912). Below these fields, there is a notice: "Notice: Changing TCP port requires board reboot". At the bottom right of the main content area, there is a circular icon with a 'C' and the text "Update parameter".

3.1.1.2.3. Auslieferungszustand

Im Auslieferungszustand hat das Ethernet Modul folgende Einstellungen:

IP Adresse: 192.168.1.1

Der Auslieferungszustand kann jederzeit durch Betätigen des linken Tasters wiederhergestellt werden.

siehe Kapitel: **IP Adresse zurücksetzen**

IP Adresse	192.168.1.1
Port	9912
Subnetz Maske	255.255.255.0
Standard Gateway	192.168.1.254

3.1.1.3. Grundkonfiguration wiederherstellen

3.1.1.3.1. IP Adresse zurücksetzen

Als Default Wert wird folgende IP Adresse verwendet: **192.168.1.1**

Linker Taster:

IP Adresse auf Default Werte zurücksetzen (192.168.1.1):

Um die IP Adresse zurückzusetzen, gehen Sie wie folgt vor:

- Taster mindestens. 5 Sek lang drücken
- Danach müssen die linken beiden LED's "CPU Activity" und "Interface Activity" viermal hintereinander aufblinken (Übernahmebestätigung)
- Anschließend hat das Modul folgende Einstellungen:

IP Adresse	192.168.1.1
Port	9912
Subnetz Maske	255.255.255.0
Standard Gateway	192.168.1.254

3.1.1.3.2. Firmware zurücksetzen

Um die Firmware auf den Default Wert zurückzusetzen, gehen Sie wie folgt vor:

Rechter Taster:

Firmware in den Auslieferungszustand zurücksetzen

Um die Firmware in den Auslieferungszustand zurückzusetzen, gehen Sie wie folgt vor:

- Taster mindestens 10 Sek lang drücken
- Danach müssen die linken drei LED's "CPU Activity", "Interface Activity" und "Status" viermal hintereinander aufblinken (Übernahmebestätigung)
- Anschließend startet das Modul neu.

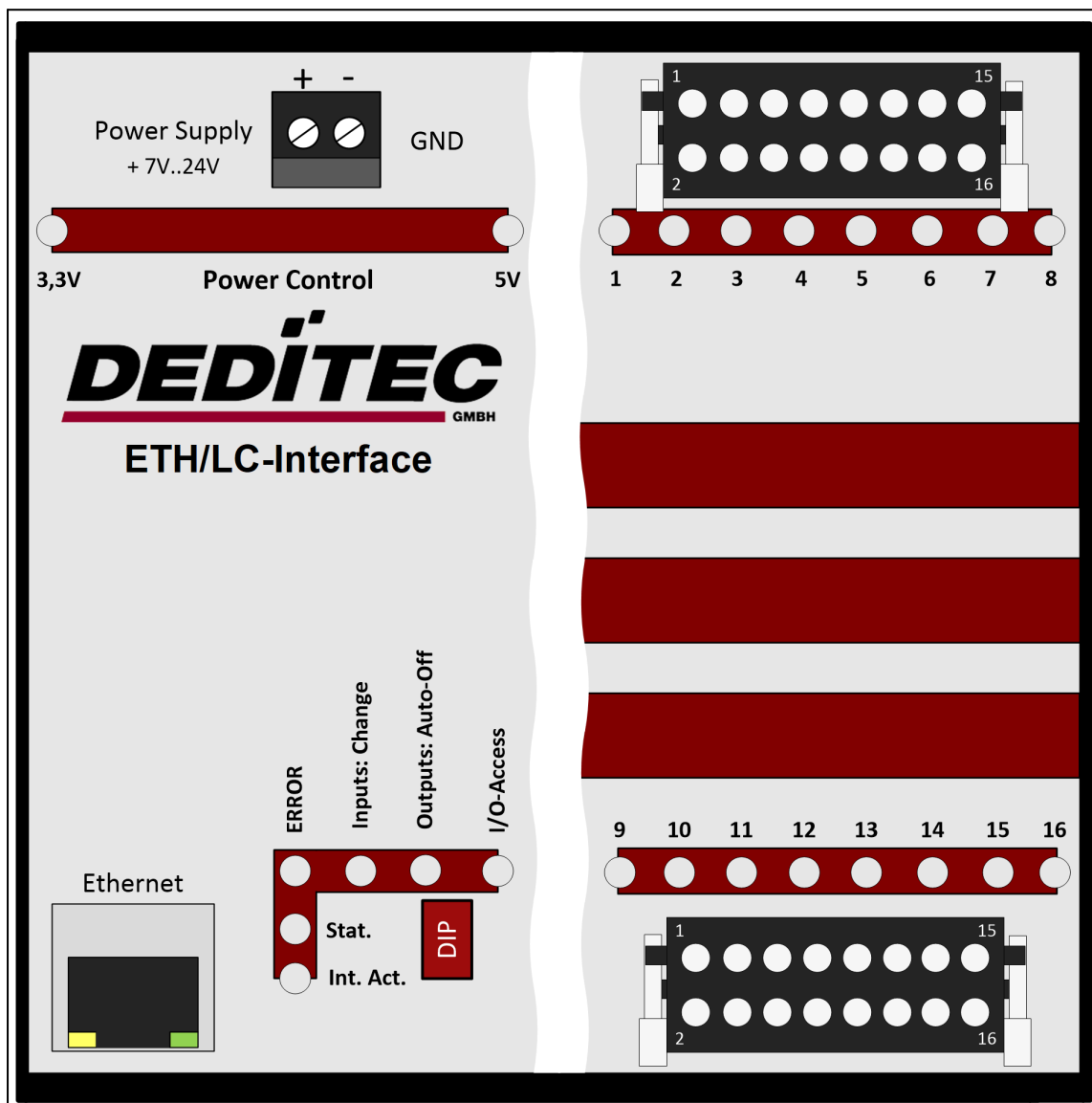
Die Firmware und Konfiguration des Auslieferungzustandes, sind jetzt wieder aktiv!

3.1.2. RO-ETH/LC

3.1.2.1. Hardware Beschreibung

3.1.2.1.1. Übersichtsbild

Die Abbildung zeigt das Steuermodul mit Ethernet Interface (links) in Kombination mit einem Ein- /Ausgabemodul (rechts).



3.1.2.1.2. Technische Daten

Elektrische Daten:

Versorgungsspannung: 7V DC .. 24V DC

Leistungsaufnahme: max. 5W

Umgebung:

Umgebungstemperatur: +10..+50 °C

Luftfeuchtigkeit: 90 %

Betauung: Nicht erlaubt

Mechanik:

Abmessungen in mm (LxBxH): L* x 122 x 51,5
*die Länge variiert mit der Art und Anzahl angeschlossener Submodule

Befestigung: Hutschiene TS 35 x 7,5 mm

Optional:

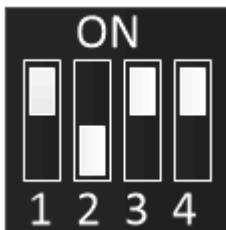
galv. Trennung: als Zubehör erhältlich

3.1.2.1.3. 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 off, OFF = Werkseinstellungen on
- DIP 4: Bootloader Mode (nur für Service-Zwecke)

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 Schalter 3	Modus / Erklärung
ON	Werkseinstellungen deaktiviert Das Module Startet mit dem im Module-Configuration-Memory konfigurierten Einstellungen.
OFF	Werkseinstellungen aktiviert Die im Module-Configuration-Memory gespeicherten Werte (IP-Adresse, Gateway, Subnetzmaske, DHCP) werden beim Modulstart ignoriert. Das Modul startet mit den folgenden Werkseinstellungen. Ethernet/LAN Boardname: Produktabhängig (z.B.: BS-ETH) Write Protection: OFF DHCP: OFF IP: 192.168.1.1 Subnetzmaske: 255.255.255.0 Gateway: 192.168.1.254 Port: 9912 Die im Module-Configuration-Memory gespeicherten Werte werden nicht verändert. Wichtig! Ist ebenfalls der DHCP-DIP-Schalter aktiviert, werden die Netzwerkeinstellungen des DHCP-Servers verwendet.

DIP Schalter 4	Modus / Erklärung
ON	Bootloader deaktiviert Das Modul startet normal.
OFF	Bootloader aktiviert Das Modul bleibt zwangsweise im Bootloader. Die Firmware wird nicht gestartet.

3.1.2.1.4. Kontroll LED's

Auf dem RO-ETH/LC Modul befinden sich eine Reihe von Kontroll LED's. Sie dienen zur einfachen optischen Zustandsanzeige von diversen Funktionen.

3.1.2.1.4.1. Definition der LEDs

LED	Bezeichnung	Erklärung
oben	3,3V	Interne 3,3V Versorgungsspannung vorhanden.
oben	5V	Interne 5V Versorgungsspannung vorhanden.
1	CPU Activity	2x blinken + lange Pause. Betriebssystem meldet: Status OK
2	Interface Activity	Kommunikation über Ethernet aktiv.
3	Status	LED leuchtet -> Modul ist betriebsbereit
4	ERROR	Fehler bei der ETH Übertragung (näheres siehe Dokument "ETH Protokoll").
5	Inputs: Change	Zustandswechsel zwischen 2

LED	Bezeichnung	Erklärung
		Auslesetakten wurde erfasst.
6	Outputs: Auto-Off	Sämtliche Ausgänge wurden auf Grund des Timeout sicherheitshalber abgeschaltet.
7	I/O Access	Zugriff der CPU auf Ein- und Ausgänge der angeschlossenen Module.

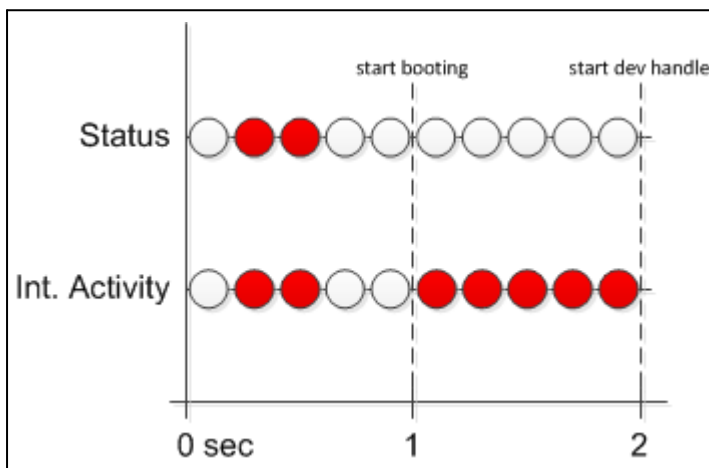
3.1.2.1.4.2. Blinkverhalten der LEDs

Nachfolgend werden die Blinksequenzen der Status LEDs dargestellt.



1. Bootvorgang

Der Bootvorgang startet direkt nach dem Einschalten der Spannungsversorgung.

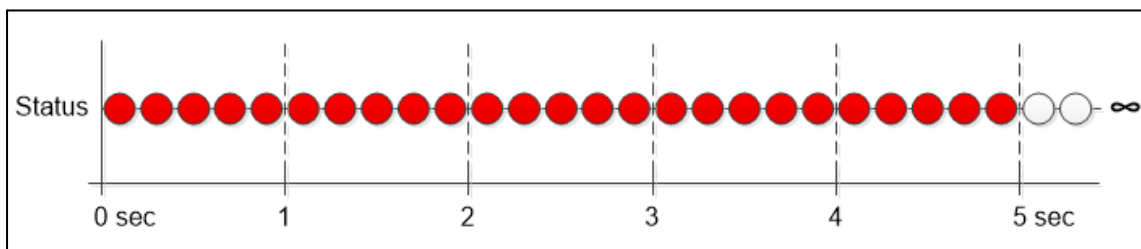


Die Bootvorgang-Sequenz wird einmalig durchlaufen.

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.



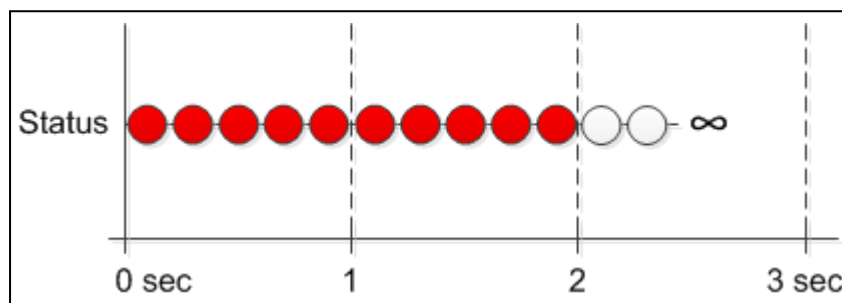
Die Status-LED leuchtet 5 Sekunden und erlischt für etwa 300ms. Die Applikations-Sequenz wiederholt sich.

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.

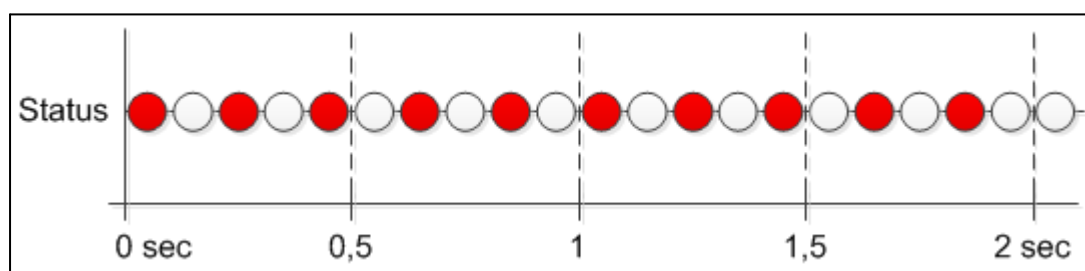


3. Bootloader erzwungen

Das Produkt kann mit dem DIP-Schalter 4 in den Bootloader gezwungen werden.

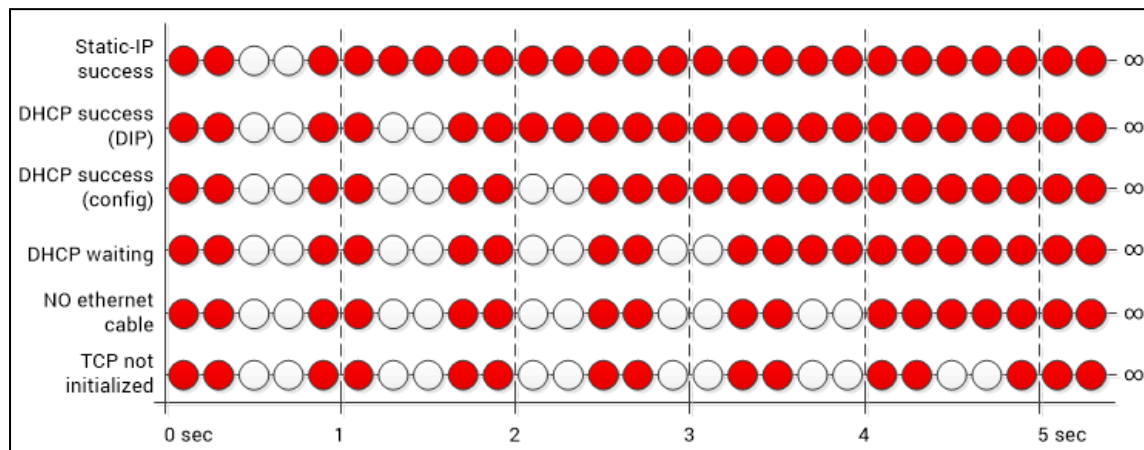
→ **siehe Kapitel "DIP Schalter"**

Diese Funktion dient nur zu Service-Zwecken.



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

4. I/F.Status LED



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.

3.1.2.2. Konfiguration des Moduls

3.1.2.2.1. Konfiguration über das DELIB Configuration Utility

Die Konfiguration ist im Kapitel **Konfiguration ETH-Schnittstelle** beschrieben.

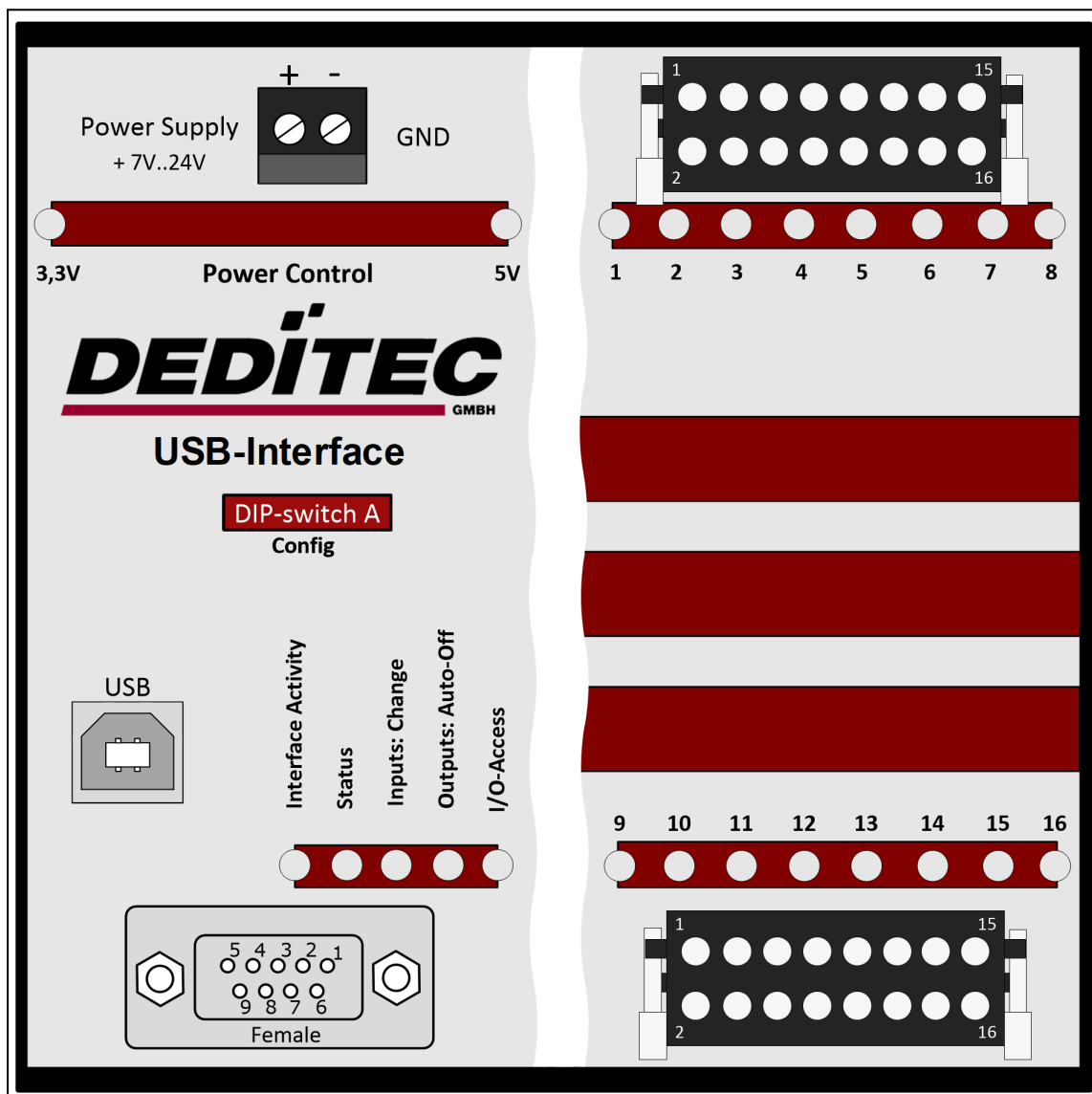
3.1.3. RO-USB

3.1.3.1. Hardware Beschreibung

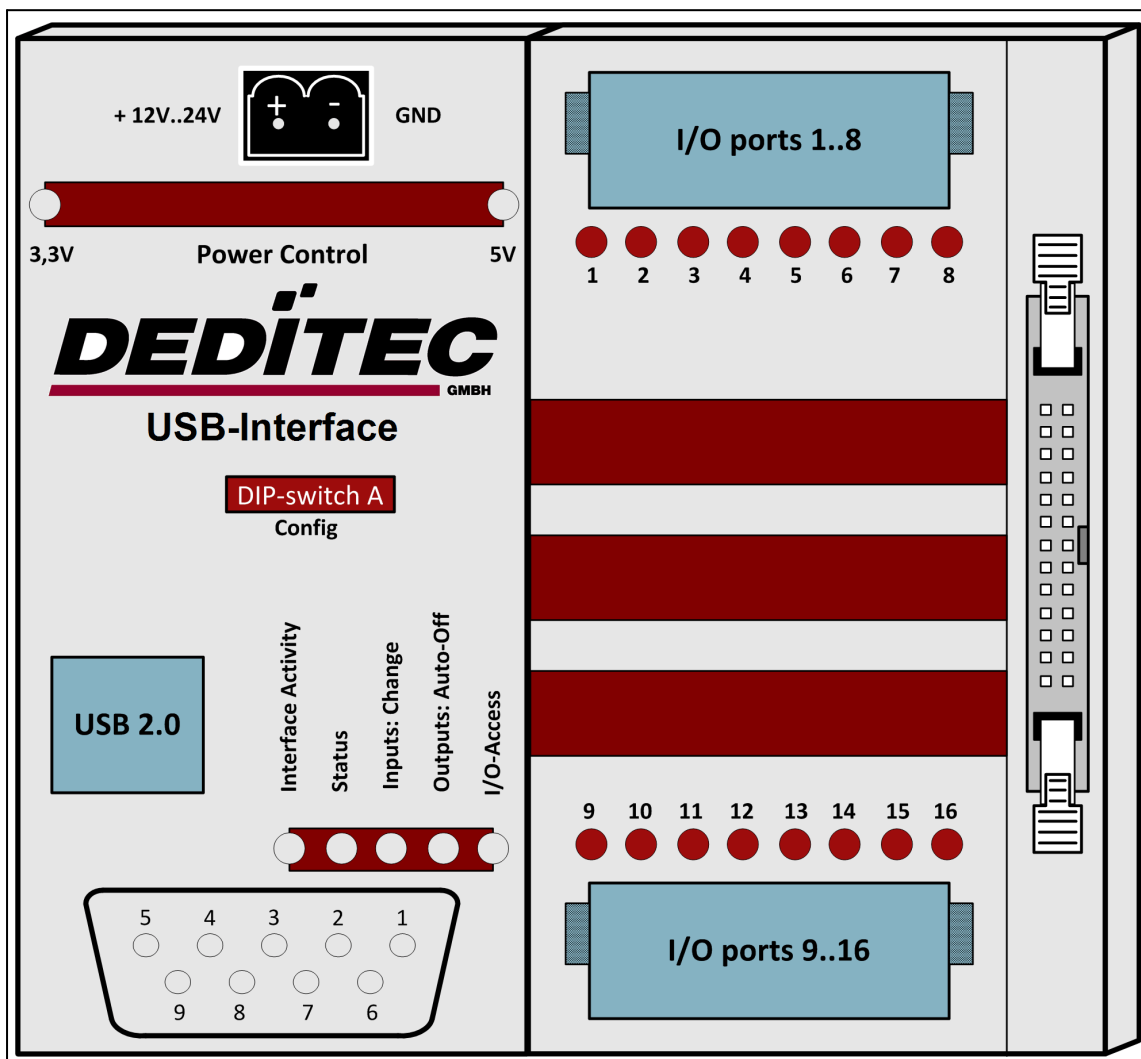
3.1.3.1.1. Übersichtsbild

Ser Interface (Revision 2.0)

Die Abbildung zeigt das Steuermodul mit USB Interface (links) in Kombination mit einem Ein-/Ausgabemodul (rechts). Für die Anbindung an den USB Bus wird ein USB A/B Kabel verwendet. Ein entsprechendes Kabel ist im Lieferumfang enthalten.

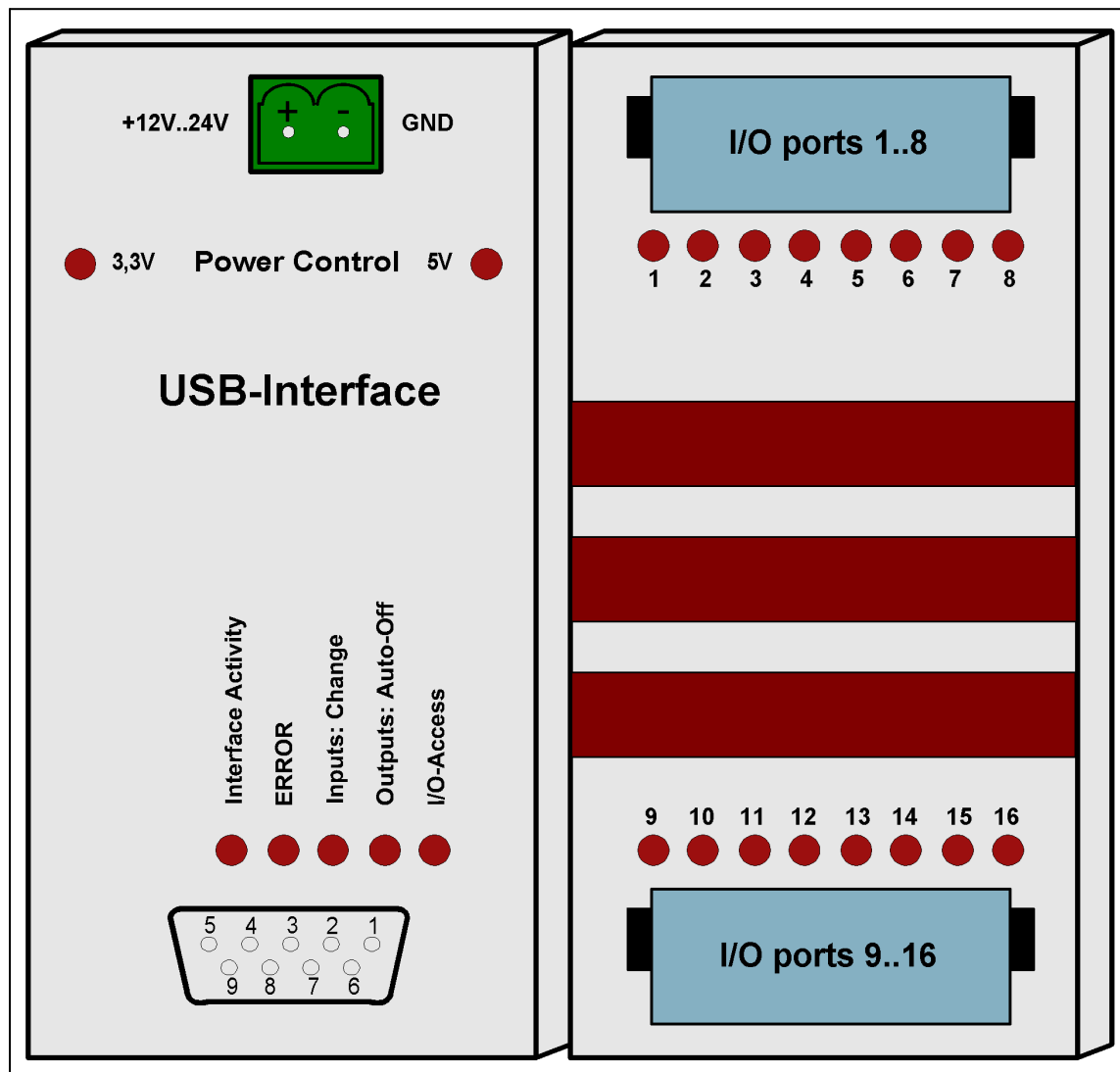


Die untere Abbildung zeigt das Steuermodul mit USB Interface (links) in Kombination mit einem I/O-Modul des flexiblen Steckverbindersystems (rechts). Für die Anbindung an den USB Bus wird ein USB A/B Kabel verwendet. Ein entsprechendes Kabel ist im Lieferumfang enthalten.

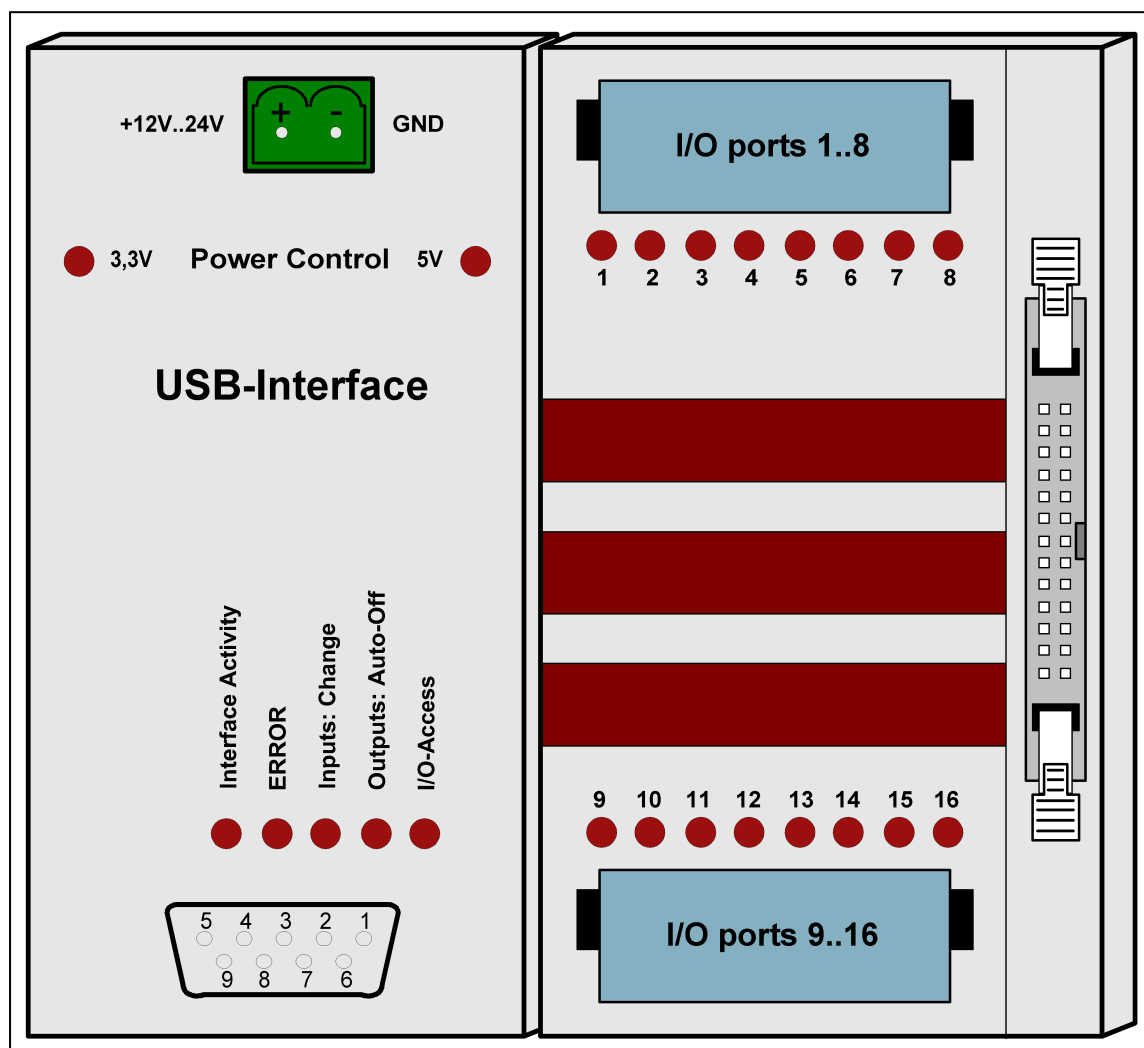


Seriell Interface (Revision 1.0)

Die Abbildung zeigt das Steuermodul mit USB Interface (links) in Kombination mit einem Ein- /Ausgabemodul (rechts). Für die Anbindung an den USB Bus, wird ein passendes Adaptermodul in Form eines USB Sticks mitgeliefert.



Die untere Abbildung zeigt das Steuermodul mit USB Interface (links) in Kombination mit einem I/O-Modul des flexiblen Steckverbindersystems (rechts). Für die Anbindung an den USB Bus, wird ein passendes Adaptermodul in Form eines USB Sticks mitgeliefert.



3.1.3.1.2. Technische Daten

Elektrische Daten:

Versorgungsspannung: 7V DC .. 24V DC

Leistungsaufnahme: max. 5W

Umgebung:

Umgebungstemperatur: +10..+50 °C

Luftfeuchtigkeit: 90 %

Betauung: Nicht erlaubt

Mechanik:

Abmessungen in mm (LxBxH): L* x 122 x 51,5
*die Länge variiert mit der Art und Anzahl angeschlossener Submodule

Befestigung: Hutschiene TS 35 x 7,5 mm

Optional:

galv. Trennung: als Zubehör erhältlich

3.1.3.1.3. Kontroll LED's

Auf dem USB Modul befinden sich eine Reihe von Kontroll LED's. Sie dienen zur einfach optischen Zustandsanzeige von diversen Funktionen.

Im Betrieb sollte nach dem Einschalten des Moduls folgende Blinksequenz auftreten:

alle fünf LED blinken kurz auf

rechte LED (I/O Access) blinkt kurz auf

alle fünf LED blinken kurz auf

3.1.3.1.3.1. Definition der LEDs

Bezeichnung	Erklärung
3,3V	Interne 3,3V Versorgungsspannung vorhanden.
5V	Interne 5V Versorgungsspannung vorhanden.
Interface Activity	Kommunikation über den USB-Bus aktiv.
ERROR	Fehler bei der USB-Übertragung (näheres siehe Dokument "USB Protokoll").
Inputs: Change	Zustandswechsel zwischen 2 Auslesetakten wurde erfasst.
Outputs: Auto-Off	Sämtliche Ausgänge wurden auf Grund des Timeout sicherheitshalber abgeschaltet.
I/O-Access	Zugriff der CPU auf Ein- und Ausgänge der angeschlossenen Module.

3.1.3.1.4. Konfiguration über das DELIB Configuration Utility

Die Konfiguration ist im Kapitel **Konfiguration USB-Schnittstelle** beschrieben.

3.1.4. RO-CAN

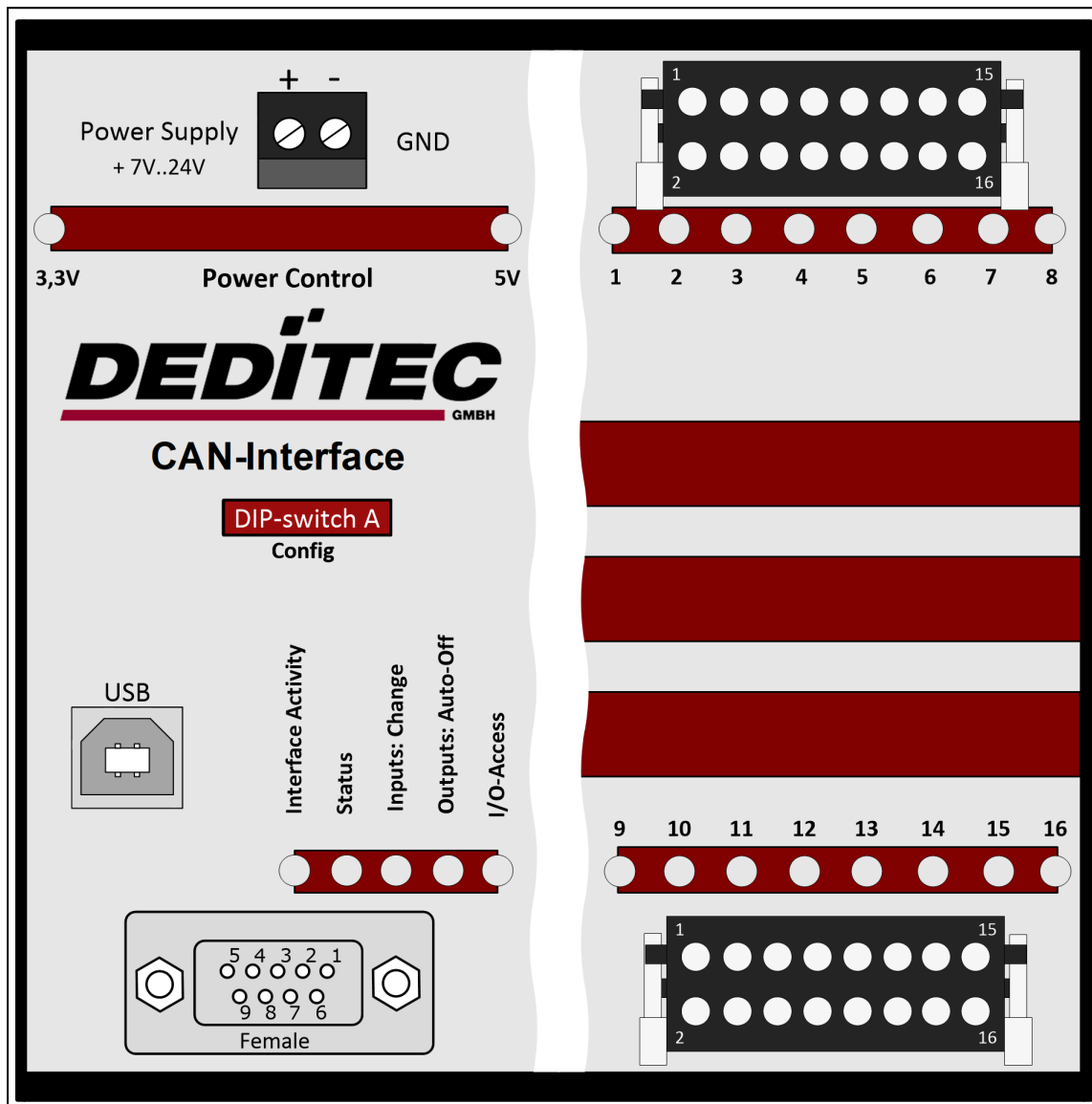
3.1.4.1. Hardware Beschreibung

3.1.4.1.1. Übersichtsbild

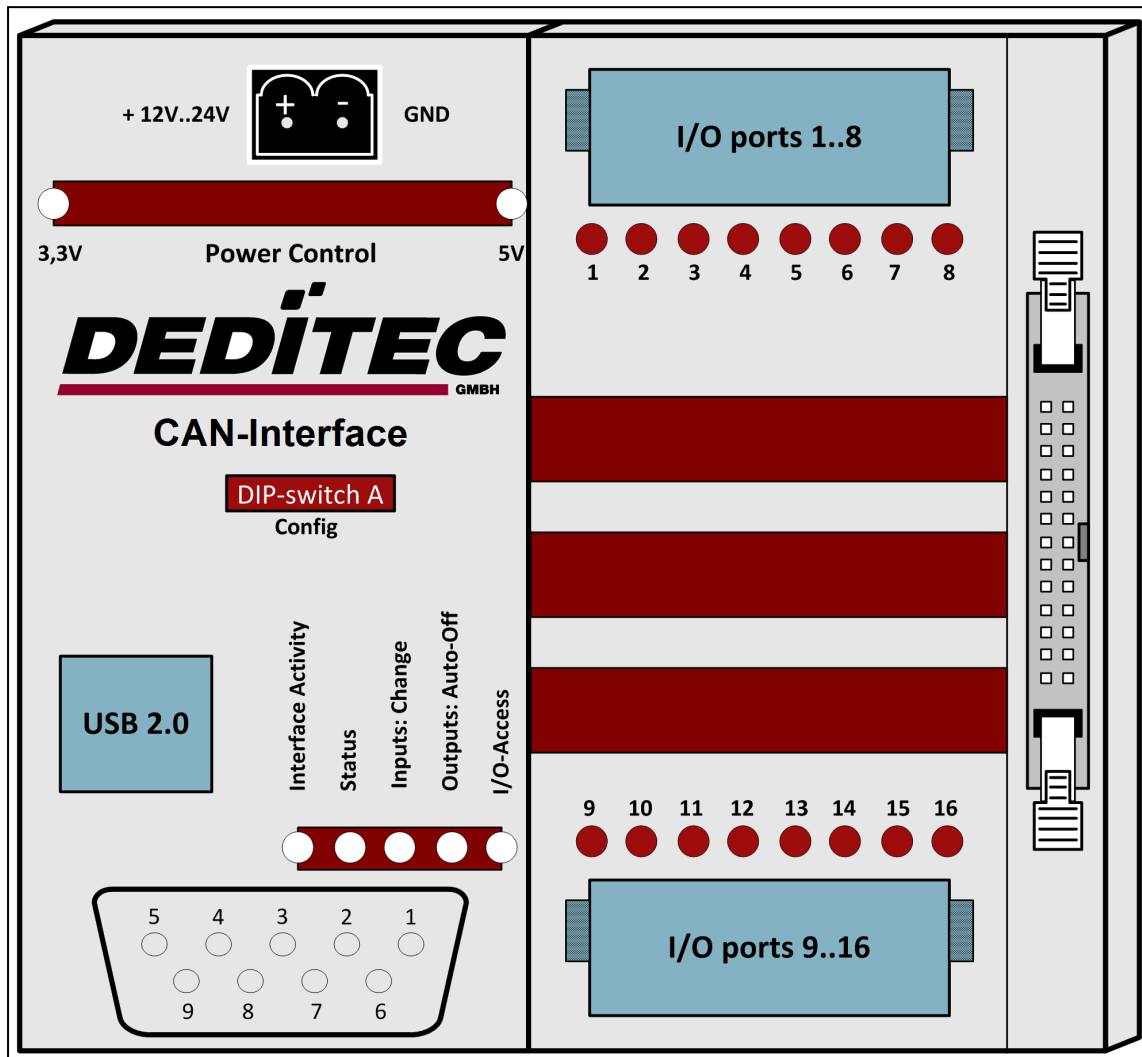
RO-CAN Interface (Revision 2.0)

Die Abbildung zeigt das Steuermodul mit CAN Interface (links) in Kombination mit einem Ein- /Ausgabemodul (rechts).

Das Modul verfügt über eine zusätzliche USB-Schnittstelle zur Konfiguration.

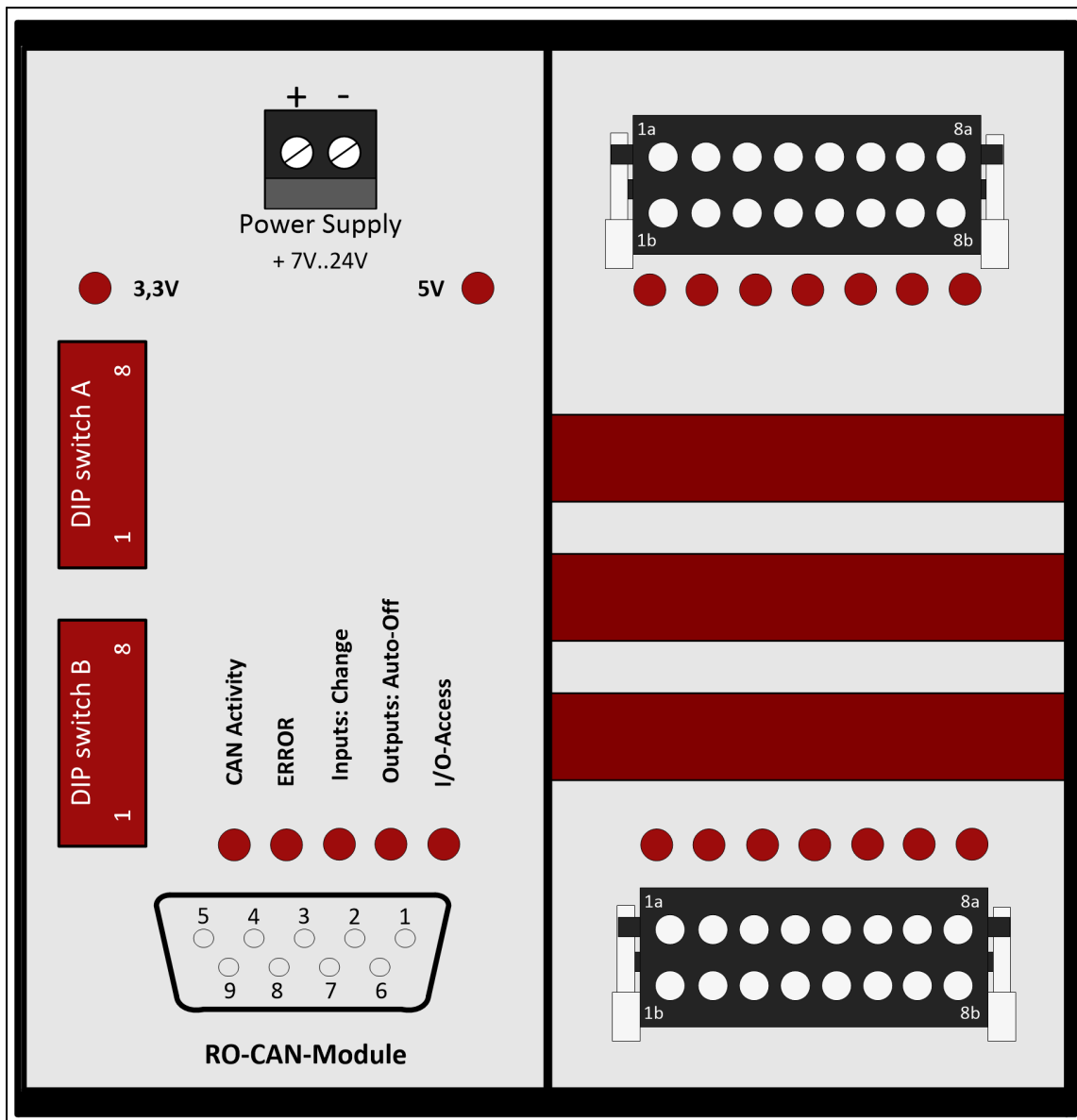


Die Abbildung zeigt das Steuermodul mit CAN Interface (links) in Kombination mit einem I/O-Modul des flexiblen Steckverbindersystems (rechts).



RO-CAN Interface (Revision 1.0)

Die Abbildung zeigt das Steuermodul mit CAN Interface (links) in Kombination mit einem Ein- /Ausgabemodul (rechts).



3.1.4.1.2. Technische Daten

Elektrische Daten:

Versorgungsspannung: 7V DC .. 24V DC

Leistungsaufnahme: max. 5W

Umgebung:

Umgebungstemperatur: +10..+50 °C

Luftfeuchtigkeit: 90 %

Betauung: Nicht erlaubt

Mechanik:

Abmessungen in mm (LxBxH): L* x 122 x 51,5
*die Länge variiert mit der Art und Anzahl angeschlossener Submodule

Befestigung: Hutschiene TS 35 x 7,5 mm

Optional:

galv. Trennung: als Zubehör erhältlich

Besonderheiten:

Ab Revision 2.0: USB-Schnittstelle zur Konfiguration über PC

3.1.4.1.3. Kontroll LED's

Auf dem CAN Modul befinden sich eine Reihe von Kontroll LED's. Sie dienen zur einfachen optischen Zustandsanzeige von diversen Funktionen.

Im Dip-Schaltermodus oder Softwaremodus sollte nach dem Einschalten des Moduls folgende Blinksequenz auftreten:

alle fünf LEDs blinken kurz auf

rechte LED (I/O Access) blinkt kurz auf

Im "Vorzugsmodus" sollte nach dem Einschalten des Moduls folgende Blinksequenz auftreten:

alle fünf LEDs blinken kurz auf

rechte LED (I/O Access) blinkt kurz auf

alle fünf LEDs blinken kurz auf

3.1.4.1.3.1. Definition der LEDs

RO-CAN Interface (Revision 2.0)

Bezeichnung	Erklärung
3,3V	Interne 3,3V Versorgungsspannung vorhanden.
5V	Interne 5V Versorgungsspannung vorhanden.
Interface Activity	Kommunikation über den CAN Bus aktiv.
Status	Modul befindet sich in der Applikation oder Bootloader
Inputs: Change	Zustandswechsel zwischen 2 Auslesetakten wurde erfasst.
Outputs: Auto-Off	Sämtliche Ausgänge wurden auf Grund des Timeout sicherheitshalber abgeschaltet.
I/O Access	Zugriff der CPU auf Ein- und Ausgänge der angeschlossenen Module.

RO-CAN Interface (Revision 1.0)

Bezeichnung	Erklärung
3,3V	Interne 3,3V Versorgungsspannung vorhanden.
5V	Interne 5V Versorgungsspannung vorhanden.
CAN Activity	Kommunikation über den CAN Bus aktiv.
ERROR	Fehler bei der CAN Übertragung (näheres siehe Dokument "CAN Protokoll").
Inputs: Change	Zustandswechsel zwischen 2 Auslesetakten wurde erfasst.
Outputs: Auto-Off	Sämtliche Ausgänge wurden auf Grund des Timeout sicherheitshalber abgeschaltet.
I/O Access	Zugriff der CPU auf Ein- und Ausgänge der angeschlossenen Module.

3.1.4.2. Konfiguration des Moduls

Um ein Modul in ein bestehendes Bus System zu integrieren, muss zunächst eine freie Moduladresse vergeben und die korrekte Übertragungsgeschwindigkeit eingestellt werden. Zum schnellen Einstieg kann der Vorzugsmodus verwendet werden.

3.1.4.2.1. DIP-Schalter

Einige Einstellungen lassen sich einfach mit Hilfe von DIP-Schaltern konfigurieren. Es kann der Vorzugs-, der Software- oder der DIP-Schalter-Modus konfiguriert werden

DIP Schalter A8	DIP Schalter A7	Erklärung
ON	ON	Vorzugsmodus: Blinksequenz beim Start, 100KHz, CAN-ID=0x100, Response-Module- Addr=1, keine extended ID's
ON	OFF	Nur für SERVICE- Zwecke: Applikation wird nicht gestartet. Zwangsweise im Bootloader
OFF	ON	Software-Modus: Konfiguration über das CAN-Configuration- Utility
OFF	OFF	DIP-Schalter-Modus: Konfiguration über DIP- Schalter, Response- Module-Addr=1, CAN 2.0A

Nur im DIP-Schalter-Modus

DIP Schalter	Erklärung
A6 bis A4	Einstellen der Übertragungsrate
A3 bis B1	Einstellen der CAN Adresse (Revision 1.0)
A3 bis A1	Einstellen der CAN Adresse (Revision 2.0)

Hinweis:

Die Stellungen der DIP-Schalter werden einmalig beim Starten des Moduls abgefragt.

3.1.4.2.2. Der Vorzugsmodus

Der Vorzugsmodus dient dazu, das Gerät schnell und einfach auf festgelegte Standardwerte zu setzen. Dies ist hilfreich bei einer schnellen und einfachen Inbetriebnahme des Moduls. Eine Fehleranalyse oder erste Inbetriebnahme wird somit erleichtert.

Werden die Schalter A7 und A8 auf "on" gestellt, gelangt man in diesen Modus. Alle anderen DIP-Schalter sind nun deaktiviert.

Das Modul arbeitet daraufhin mit folgenden Einstellungen:

Adressierung mit 11 Bit

100Kbit/sec Bitrate

CAN-Adresse = 0x100

Response-Modul-Addr = 1 (an diese Adresse werden die Antworten zurückgesendet)

3.1.4.2.3. Software-Modus

Die Konfiguration ist im Kapitel **Konfiguration CAN-Schnittstelle** beschrieben.

3.1.4.2.4. DIP-Schalter-Modus

In diesem Modus muss das Interface-Modul über die DIP Schalter konfiguriert werden. Hierfür müssen die DIP Schalter A7=OFF und A8=OFF sein. Der Adressraum ist 11 Bit breit (CAN 2.0A). Die Modul-Adresse ist, je nach Revision des CAN-Interfaces, über DIP A3..A1 (Revision 2.0), bzw. DIP A3..A1 und B8..B1 (Revision 1.0) einstellbar. Die Response-Modul-Addr ist 1 (an diese Adresse werden die Antworten zurückgesendet)

3.1.4.2.4.1. Einstellen der Übertragungsgeschwindigkeit (Bitrate)

Je nachdem welche Reichweite der CAN Bus hat, werden unterschiedliche Übertragungsgeschwindigkeiten erreicht. Hierzu lassen sich folgende Bitraten per DIP-Schalter einstellen. Andere Bitraten sind unter Umständen nur per Software realisierbar.

DIP	1Mbit	500K	250K	125K	100K	50K	20K	10K
A6	On	On	On	On	Off	Off	Off	Off
A5	On	On	Off	Off	On	On	Off	Off
A4	On	Off	On	Off	On	Off	On	Off

3.1.4.2.4.2. Einstellen der CAN-Moduladresse

Jedes Gerät, welches sich im CAN Netz befindet, benötigt eine feste Adresse um direkt angesprochen werden zu können. Hierzu lassen sich, je nach Revision des CAN-Interfaces, bis zu 8 (Revision 2.0), bzw. 2047 (Revision 1.0) unterschiedliche Adressen per DIP-Schalter einstellen.

RO-CAN Interface (Revision 2.0)

DIP Schalter	Bit	Wertigkeit ON	Wertigkeit OFF
A3	Bit 10	1024	0
A2	Bit 9	512	0
A1	Bit 8	256	0

Verfügbare Adressen

Adresse	DIP A3	DIP A2	DIP A1
0	Off	Off	Off
256	Off	Off	On
512	Off	On	Off
768	Off	On	On
1024	On	Off	Off
1280	On	Off	On
1536	On	On	Off
1792	On	On	On

RO-CAN Interface (Revision 1.0)

DIP Schalter	Bit	Wertigkeit ON	Wertigkeit OFF
A3	Bit 10	1024	0
A2	Bit 9	512	0
A1	Bit 8	256	0
B8	Bit 7	128	0
B7	Bit 6	64	0
B6	Bit 5	32	0
B5	Bit 4	16	0
B4	Bit 3	8	0
B3	Bit 2	4	0
B2	Bit 1	2	0
B1	Bit 0	1	0

Beispiele

DIP Schalter	Adresse 0	Adresse 117	Adresse 588
A3	Off	Off	Off
A2	Off	Off	On
A1	Off	Off	Off
B8	Off	Off	Off
B7	Off	On	On
B6	Off	On	Off
B5	Off	On	Off
B4	Off	Off	On
B3	Off	On	On
B2	Off	Off	Off
B1	Off	On	Off

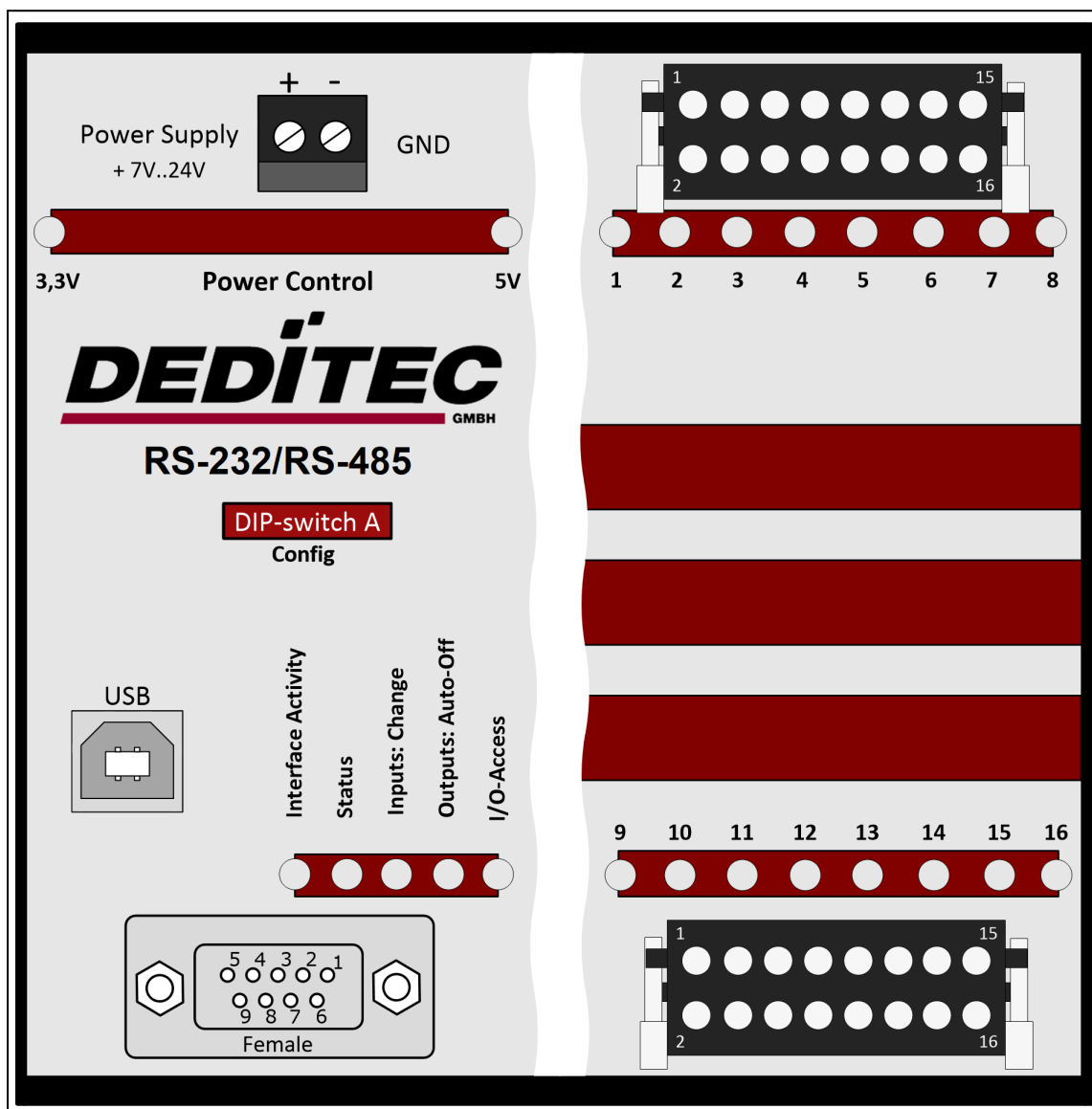
3.1.5. RO-SER

3.1.5.1. Hardware Beschreibung

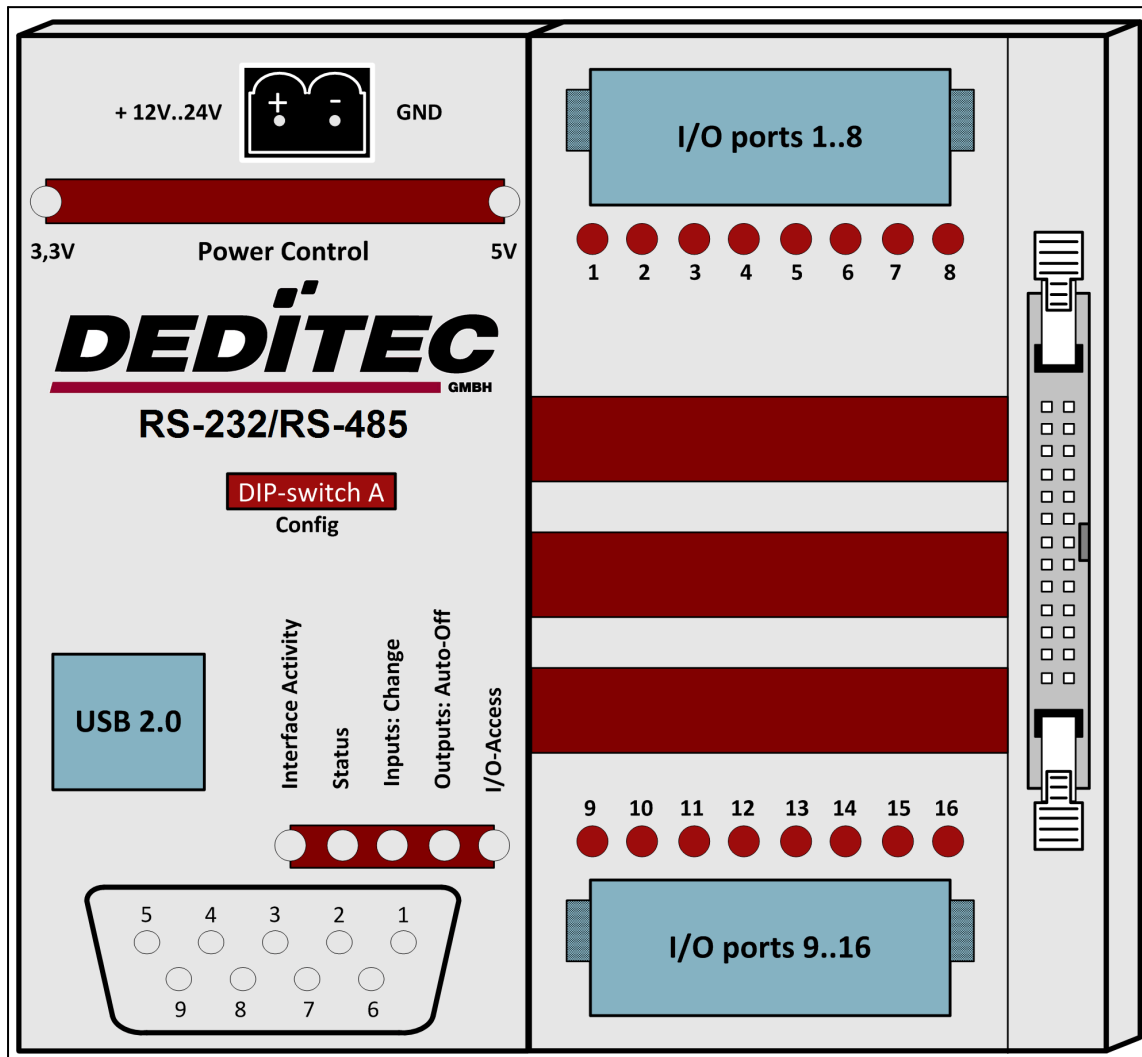
3.1.5.1.1. Übersichtsbild

RO-SER Interface (Revision 2.0)

Die Abbildung zeigt das Steuermodul mit RS-232/RS-485 Interface (links) in Kombination mit einem Ein- /Ausgabemodul (rechts).

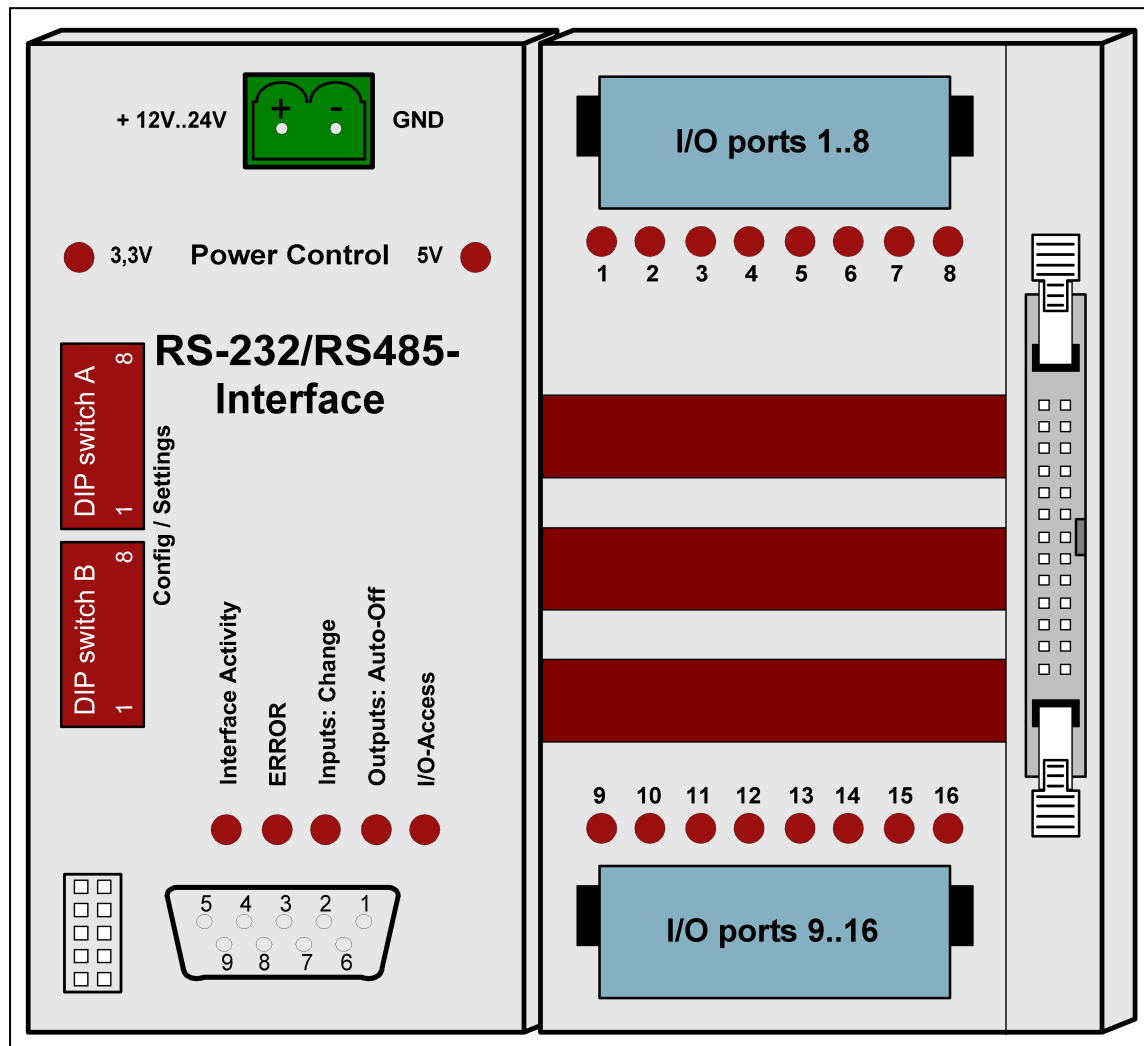


Die Abbildung zeigt das Steuermodul mit RS-232/RS-485 Interface (links) in Kombination mit einem I/O-Modul des flexiblen Steckverbindersystems (rechts).

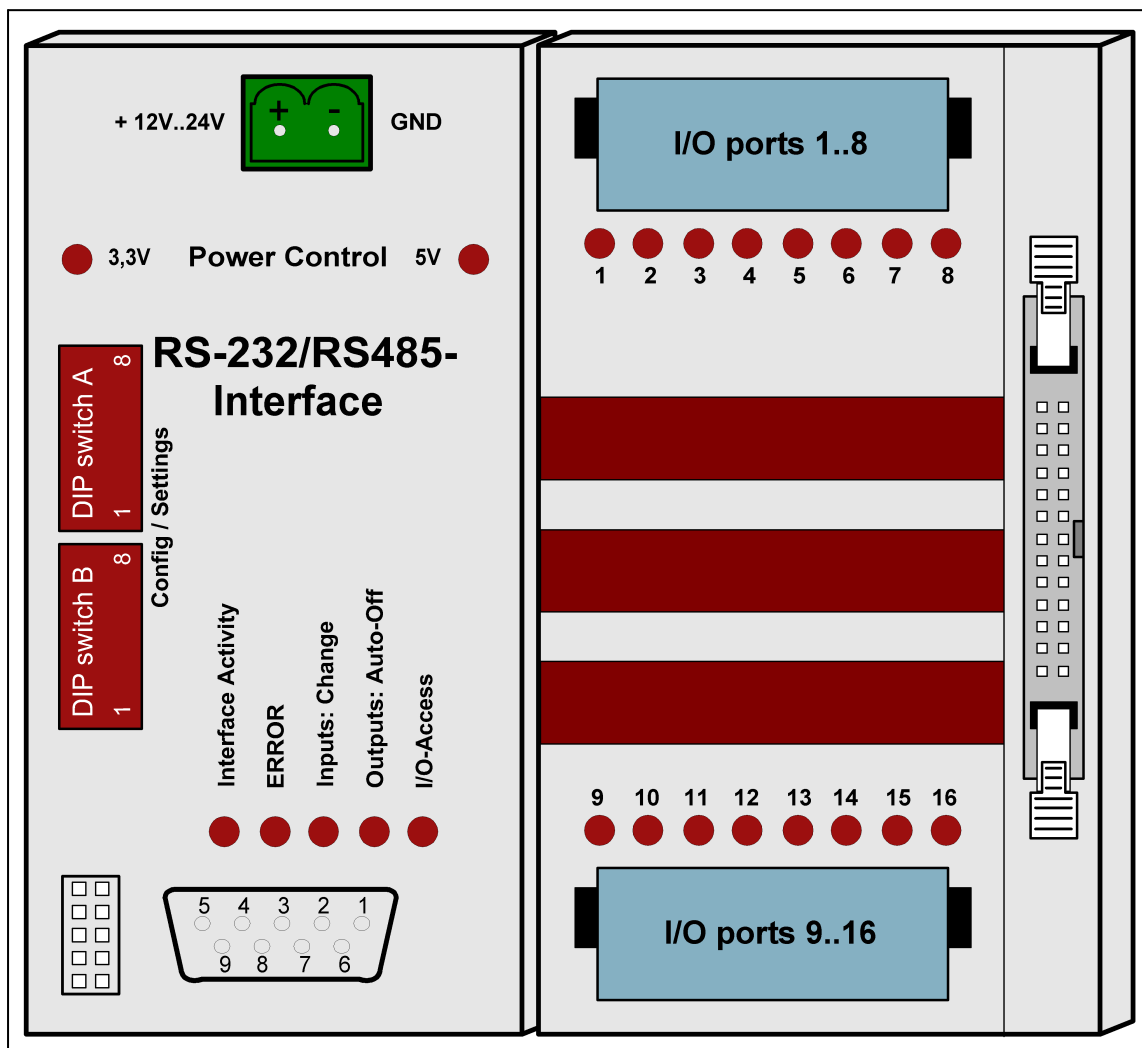


RO-SER Interface (Revision 1.0)

Die Abbildung zeigt das Steuermodul mit RS-232/RS-485 Interface (links) in Kombination mit einem Ein- /Ausgabemodul (rechts).



Die Abbildung zeigt das Steuermodul mit RS-232/RS-485 Interface (links) in Kombination mit einem I/O-Modul des flexiblen Steckverbindersystems (rechts).



3.1.5.1.2. Technische Daten

Elektrische Daten:

Versorgungsspannung: 7V DC .. 24V DC

Leistungsaufnahme: max. 5W

Umgebung:

Umgebungstemperatur: +10..+50 °C

Luftfeuchtigkeit: 90 %

Betauung: Nicht erlaubt

Mechanik:

Abmessungen in mm (LxBxH): L* x 122 x 51,5
*die Länge variiert mit der Art und Anzahl angeschlossener Submodule

Befestigung: Hutschiene TS 35 x 7,5 mm

Optional:

galv. Trennung: als Zubehör erhältlich

Besonderheiten:

Ab Revision 2.0: USB-Schnittstelle zur Konfiguration über PC

3.1.5.1.3. Kontroll LED's

Auf dem RS-232/RS-485 Modul befinden sich eine Reihe von Kontroll LED's. Sie dienen zur einfach optischen Zustandsanzeige von diversen Funktionen.

Im Normalbetrieb sollte nach dem Einschalten des Moduls folgende Blinksequenz auftreten:

alle fünf LEDs blinken kurz auf
rechte LED (I/O Access) blinkt kurz auf

Im "Vorzugsmodus" sollte nach dem Einschalten des Moduls folgende Blinksequenz auftreten:

alle fünf LEDs blinken kurz auf
rechte LED (I/O Access) blinkt kurz auf
alle fünf LEDs blinken kurz auf

3.1.5.1.3.1. Definition der LEDs

Bezeichnung	Erklärung
3,3V	Interne 3,3V Versorgungsspannung vorhanden.
5V	Interne 5V Versorgungsspannung vorhanden.
RS-232/RS-485 Activity	Kommunikation über den seriellen Bus aktiv.
ERROR	Fehler bei der seriellen Übertragung (näheres siehe Dokument "Seriellles Protokoll").
Inputs: Change	Zustandswechsel zwischen 2 Auslesetakten wurde erfasst.
Outputs: Auto-Off	Sämtliche Ausgänge wurden auf Grund des Timeout sicherheitshalber abgeschaltet.
I/O-Access	Zugriff der CPU auf Ein- und Ausgänge der angeschlossenen Module.

3.1.5.2. Konfiguration des Moduls

Um ein Modul in ein bestehendes Bussystem zu integrieren, ist es zunächst erforderlich, eine freie Moduladresse und die entsprechende Bitrate zu vergeben. Für den schnellen Betrieb des Systems kann alternativ der "Special Mode" genutzt werden.

3.1.5.2.1. DIP-Schalter

Einige Einstellungen lassen sich einfach mit Hilfe von DIP Schaltern konfigurieren. Es kann der Vorzugs-, der Text- oder der DIP-Schalter-Modus konfiguriert werden

DIP Schalter A8	DIP Schalter A7	Erklärung
ON	ON	Vorzugsmodus: Blinksequenz beim Start, 115k Baudrate, Modul- Adresse=0
ON	OFF	Nur für SERVICE- Zwecke: Applikation wird nicht gestartet. Zwangsweise im Bootloader
OFF	ON	DIP-Schalter-Modus
OFF	OFF	Serieller Textmodus

Nur im seriellen Text- oder DIP-Schalter-Modus

DIP Schalter A6	Erklärung
ON	Echo = ON, serielle empfangene Zeichen werden zurückgegeben
OFF	Echo = OFF

DIP Schalter	Erklärung
A5	Reserviert
A4 bis A1	Einstellen der Baudrate

Hinweis:

Die Stellungen der DIP-Schalter werden einmalig beim Starten des Moduls abgefragt.

3.1.5.2.2. Der Vorzugsmodus

Der Vorzugsmodus dient dazu, das Gerät schnell und einfach auf festgelegte Standardwerte zu setzen. Dies ist hilfreich bei einer schnellen und einfachen Inbetriebnahme des Moduls. Eine Fehleranalyse oder erste Inbetriebnahme wird somit erleichtert.

Werden die Schalter A7 und A8 auf "on" gestellt, gelangt man in diesen Modus. Alle anderen DIP-Schalter sind nun deaktiviert.

Das Modul arbeitet daraufhin mit folgenden Einstellungen:

115K Baud

Moduladresse = 0 (deaktiviert)

Echo = Off

3.1.5.2.3. DIP-Schalter-Modus

In diesem Modus wird die Konfiguration der seriellen Schnittstelle (Baudrate und Echo) direkt per DIP-Schalter festgelegt.

Um in den DIP-Schalter-Modus zu gelangen, muss DIP-Schalter A7 auf "on" und A8 auf "off" geschaltet werden.

Die Kommunikation mit dem Modul muss hierbei über das serielle Protokoll erfolgen (weitere Informationen finden Sie hierzu im Handbuch "[Übertragungsprotokolle + Registerbelegung](#)").

Hinweis:

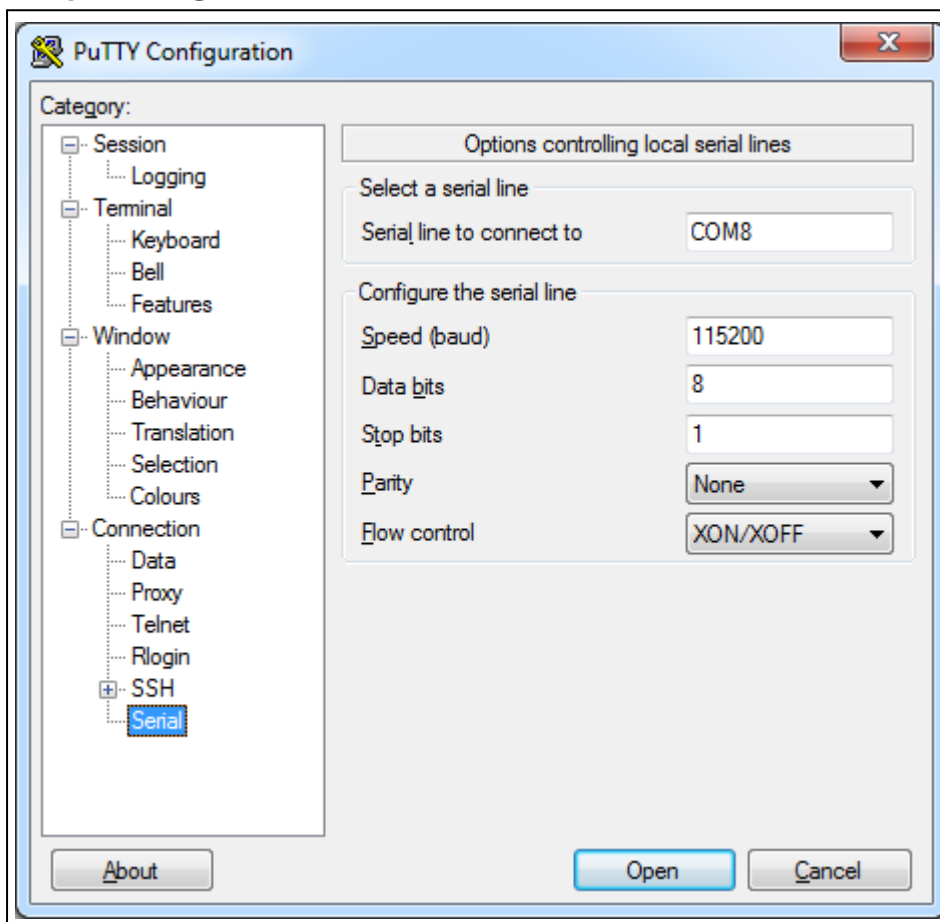
Im DIP-Schalter-Modus kann das Modul nicht über die DELIB-Treiberbibliothek angesprochen werden.

3.1.5.2.4. Serieller Textmodus

Der serielle Textmodus erlaubt die Ansteuerung von digitalen und analogen Ein-/Ausgänge, sowie Lese- und Schreibzugriffe auf Register, direkt per "Klar-Text" Kommandos über ein beliebiges Terminal Programm (z.B. PuTTY).

Hierzu müssen DIP-Schalter A7 und A8 auf "off" geschaltet werden. Da es sich beim Textmodus um einen erweiterten DIP-Schalter-Modus handelt, muss die Baudrate (DIP A1..A4) und das Echo (DIP A6) ebenfalls per DIP-Schalter konfiguriert werden.

Beispielkonfiguration PuTTY



Hinweis:

Im Textmodus kann das Modul nicht über die DELIB-Treiberbibliothek oder das serielle Protokoll angesprochen werden.

3.1.5.2.4.1. Übersicht Schreibbefehle

Definition

cmd1 + cmd2 + par1 + "=" + par2

Übersicht

Beschreibung	cmd1	cmd2	par1 [dez]	par2 [dez]
Digitalen Ausgang setzen	S (Set)	D (Digital)	Kanal- nummer	1 Bit Wert (0=Aus, 1=An)
Analogen Ausgang setzen		A (Analog)		2 Byte Wert (0 .. 65535)
Register schreiben (1 Byte)	W (Write)	B (Byte)	Register	1 Byte Wert (0 .. 255)
Register schreiben (2 Byte)		W (Word)		2 Byte Wert (0 .. 65535)

Return-Wert

ERROR oder OK

Beispiele

```
sd5=1
OK
// schaltet digitalen Ausgang 5 an (entspricht DapiDOSet1(handle, 5, 1))

sd2=0
OK
// schaltet digitalen Ausgang 2 aus (entspricht DapiDOSet1(handle, 2, 0))

sa3=4096
OK
// setzt den Datenwert 0x1000 an analogen Ausgang 3 (entspricht
DapiDASet(handle, 3, 0x1000))

wb8194=64
OK
// schreibt den Datenwert 0x40 in Register 0x2002 (entspricht
DapiWriteByte(handle, 0x2002, 0x40))

ww8192=12288
OK
// schreibt den Datenwert 0x3000 in Register 0x2000 (entspricht
DapiWriteWord(handle, 0x2000, 0x3000))
```

3.1.5.2.4.2. Übersicht Lesebefehle

Definition

cmd1 + cmd2 + par1

Übersicht

Beschreibung	cmd1	cmd2	par1 [dez]	Return-Wert
Digitalen Eingang lesen	G (Get)	D (Digital)	Kanal- nummer	1 Bit Wert (0=Aus, 1=An)
Analogen Eingang lesen		A (Analog)		2 Byte Wert (0 .. 65535)
Register lesen (1 Byte)	R (Read)	B (Byte)	Register	1 Byte Wert (0 .. 255)
Register lesen (2 Byte)		W (Word)		2 Byte Wert (0 .. 65535)

Return-Wert

ERROR oder Wert

Beispiele

```
gd3
1
// liest digitalen Eingang 3 (entspricht DapiDIGet1(handle, 3))

ga2
4711
// liest analogen Ausgang 2 (entspricht DapiADGet(handle, 2))

rb8194
64
// liest Register 0x2002 (entspricht DapiReadByte(handle, 0x2002))

rw8192
12288
// liest Register 0x2000 (entspricht DapiReadWord(handle, 0x2000))
```

3.1.5.2.5. Echo aktivieren

Serielle empfangene Zeichen sollen auf den Bildschirm zurückgegeben werden (ON=ja, OFF=nein).

3.1.5.2.6. Einstellen der Baudrate

Die untere Tabelle zeigt, welche Baudraten mit den 4 DIP Schaltern (A1 bis A4) eingestellt werden können.

Baudrate	DIP A4	DIP A3	DIP A2	DIP A1
1,25 Mbit	On	On	On	On
625 Kbit	On	On	On	Off
250 Kbit	On	On	Off	On
125 Kbit	On	On	Off	Off
115200 Bit	On	Off	On	On
57600 Bit	On	Off	On	Off
50000 Bit	On	Off	Off	On
38400 Bit	On	Off	Off	Off
19200 Bit	Off	On	On	On
9600 Bit	Off	On	On	Off
4800 Bit	Off	On	Off	On
2400 Bit	Off	On	Off	Off
1200 Bit	Off	Off	On	On
600 Bit	Off	Off	On	Off
300 Bit	Off	Off	Off	On

3.1.5.2.7. Einstellen der Moduladresse (nur bei RS-485)

Da im RS-485 Betrieb mehrere Module an einem BUS System angeschlossen werden können, ist es erforderlich jedem Modul eine eigene Adresse zuzuweisen. Diese kann mit den DIP Schaltern B1 bis B8 zwischen 0 und 255 eingestellt werden. Bei Modulnummer 0 wird die Modulnummer ignoriert.

Baudrate	Bit	Wertigkeit ON	Wertigkeit OFF
DIP Schalter B8	Bit 7	128	0
DIP Schalter B7	Bit 6	64	0
DIP Schalter B6	Bit 5	32	0
DIP Schalter B5	Bit 4	16	0
DIP Schalter B4	Bit 3	8	0
DIP Schalter B3	Bit 2	4	0
DIP Schalter B2	Bit 1	2	0
DIP Schalter B1	Bit 0	1	0

Beispiele:

Baudrate	Adresse 0	Adresse 25	Adresse 237
DIP Schalter B8	Off	Off	On
DIP Schalter B7	Off	Off	On
DIP Schalter B6	Off	Off	On
DIP Schalter B5	Off	On	Off
DIP Schalter B4	Off	On	On
DIP Schalter B3	Off	Off	On
DIP Schalter B2	Off	Off	Off
DIP Schalter B1	Off	On	On

Hinweis:

Der RS-485 Modus steht nur in der Revision 1.0 zur Verfügung.

3.1.5.2.8. Konfiguration über das DELIB Configuration Utility

Die Konfiguration ist im Kapitel **Konfiguration RS-232/RS-485-Schnittstelle** beschrieben.

3.1.6. Schnittstellen

3.1.6.1. 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

3.1.6.2. Ethernet

Technische Daten:

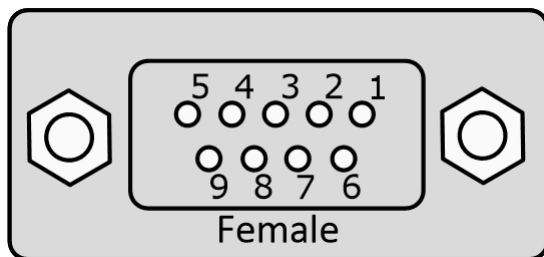
Standard:	Ethernet 100/10Mbit
Verbindungsaufbau	Ethernet / LAN Kabel
Zugriffszeit PC auf Modul*:	Wire: 1,56 ms

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

3.1.6.3. CAN

Pinbelegung D-SUB Buchse:

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



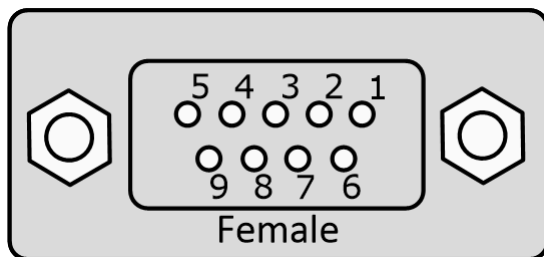
Technische Daten:

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)

3.1.6.4. RS-232

Pinbelegung D-SUB Buchse:

TX	Pin 2
RX	Pin 3
GND	Pin 5



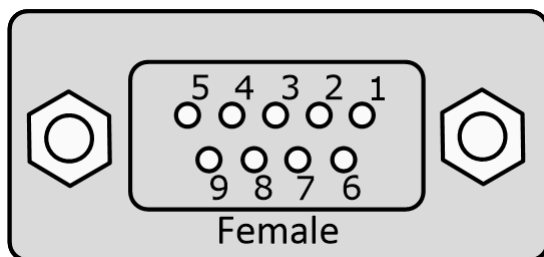
Technische Daten:

Standard:	RS-232
Geschwindigkeit:	bis 115200 Baud

3.1.6.5. RS-485

Pinbelegung D-SUB Buchse:

RS-485 B	Pin 2
RS-485 A	Pin 7
GND	Pin 5



Technische Daten:

Standard:	RS-485
Geschwindigkeit:	bis 115200 Baud

3.1.6.6. Wählen der Schnittstellenvariante RS-232 / RS-485 / CAN

Bei Auslieferung befindet sich das Schnittstellenmodul standardmäßig im CAN Modus. Möchten Sie RS-232 oder RS-485 verwenden, kontaktieren Sie bitte unsere Support-Hotline (s. Kapitel: **Kunden-Support**).

Rechts neben dem D-SUB Stecker befindet sich eine 14pol. Stiftleiste mit den dazugehörigen Jumpers. In der nachfolgenden Tabelle wird gezeigt, welche Jumper gesteckt werden müssen.

Stiftleiste	Schnittstelle	Jumper setzen
	RS-232	Pin1 & Pin3 Pin2 & Pin4
	RS-485	Pin3 & Pin5 Pin4 & Pin6 Pin7 & Pin8
	Abschlusswiderstand 120 Ohm optional	Pin9 & Pin10
	CAN	Pin 11 & Pin 12 Pin 13 & Pin 14

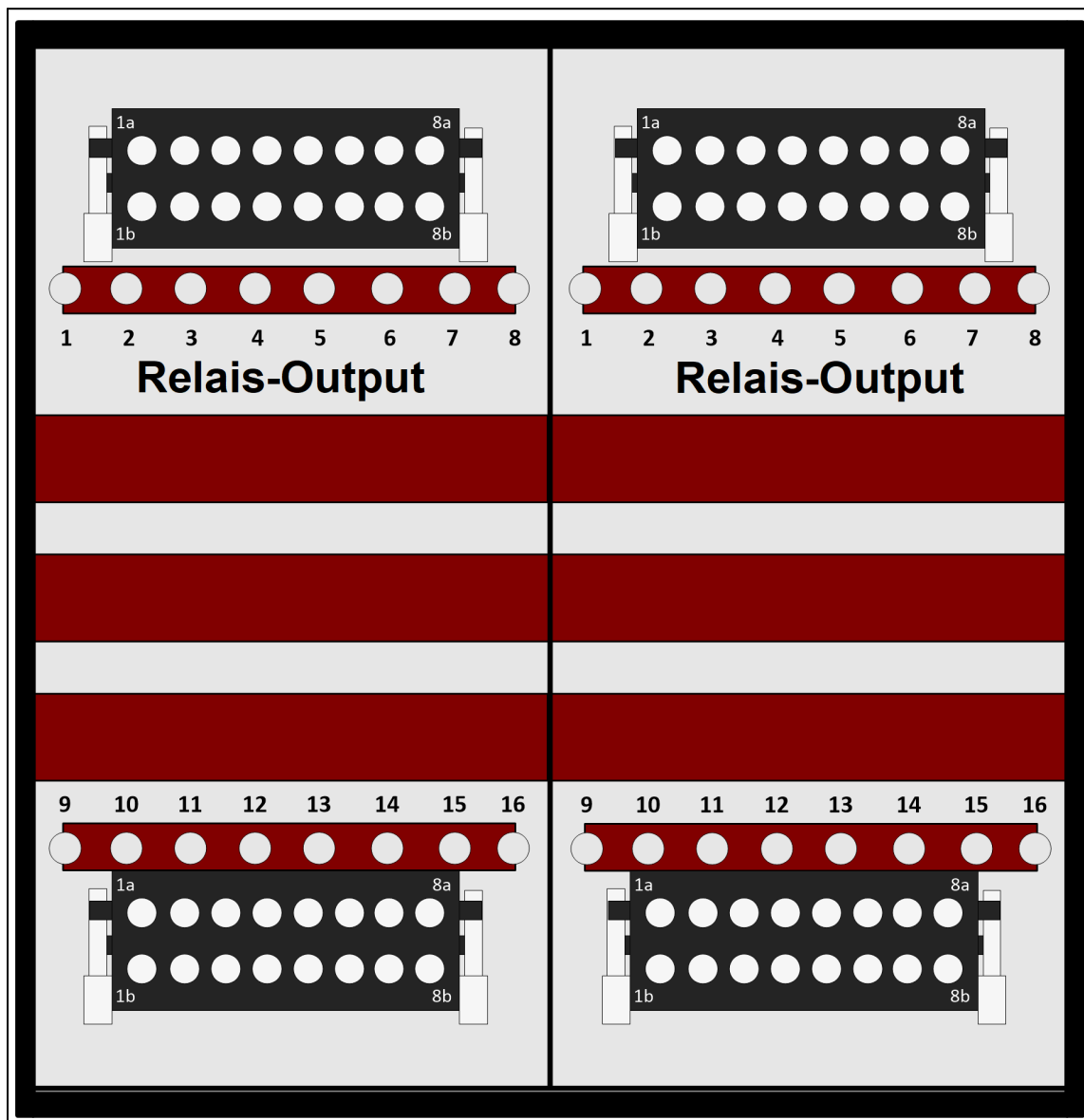
RS-232	RS-485	CAN
		

3.2. Submodule

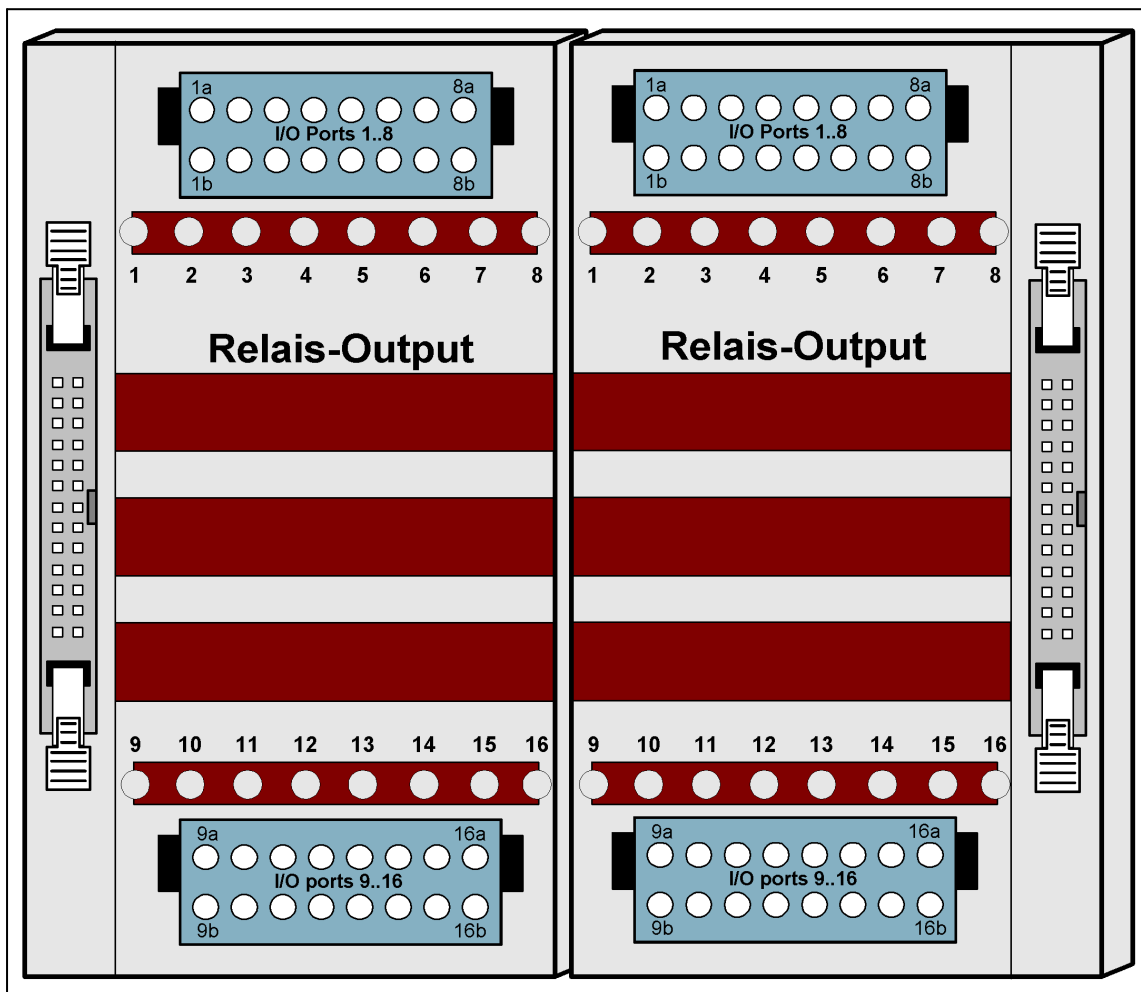
3.2.1. RO-REL16

3.2.1.1. Übersichtsbild

Die Abbildung zeigt zwei Module nebeneinander mit entsprechender Durchnummerierung der Anschlussklemmen.



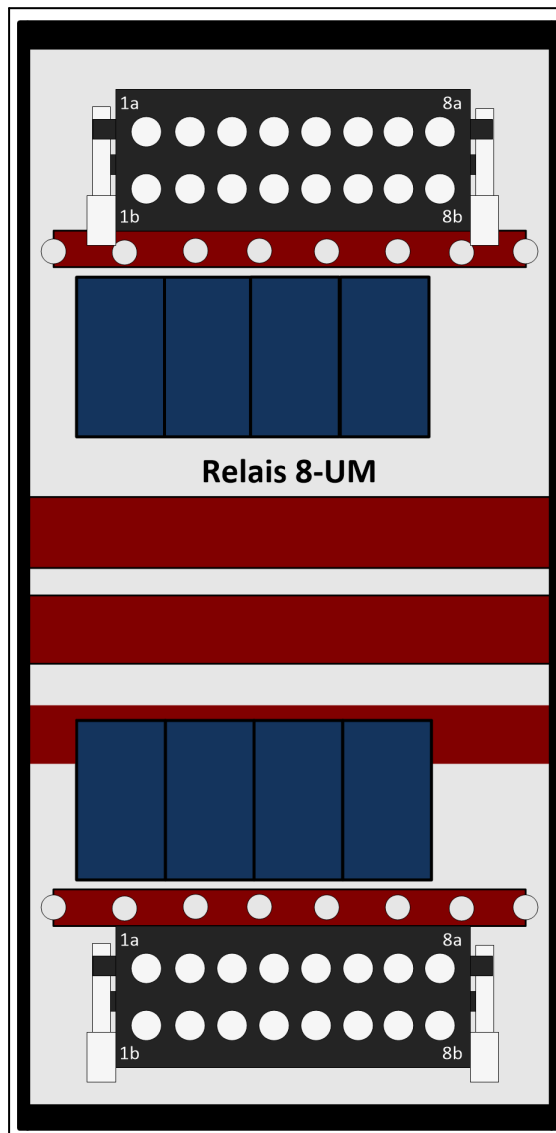
Untere Abbildung zeigt ein I/O-Modul des flexiblen Steckverbindersystems mit 32 Ausgängen und entsprechender Durchnummerierung der Anschlussklemmen. An den äußeren Enden des Moduls befindet sich jeweils eine 26pol. Wannenstiftleiste mit Verriegelungshebeln. Somit können mehrere Moduleinheiten mit Hilfe eines Flachbandkabels miteinander verbunden werden.



3.2.2. RO-REL8_UM

3.2.2.1. Übersichtsbild

Die Abbildung zeigt ein RO-REL8_UM Modul mit entsprechender Durchnummerierung der Anschlussklemmen.



3.2.2.2. Bistabile Relais Ausgänge

Durch die bistabilen Relais, die in diesem Modul verbaut werden, wird die Schaltstellungen der Relais auch bei einem Verlust der Spannungsversorgung oder einem Neustart des Moduls beibehalten.

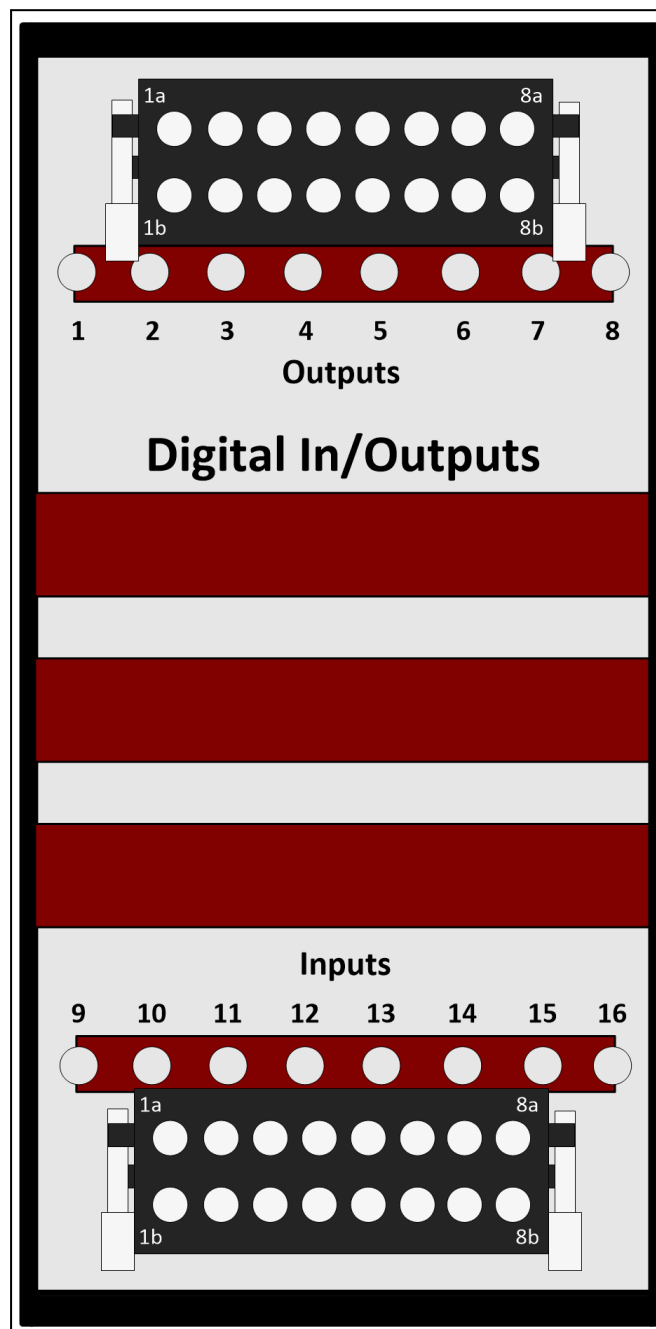
Jedes dieser bistabilen Relais verfügt über einen Wechslerkontakt mit zwei LEDs als Zustandsanzeige.

Die bistabilen Relais sind durch ihre spezielle Eigenschaft vielfältig einsetzbar. Anwendungsbeispiele sind z.B. in der Automobilbranche zu finden aber auch in der Netztechnik finden sie großen Absatz. Zusätzlich ist der Sicherheitsaspekt dieser Relais enorm, durch das beibehalten der Schaltstellungen wird einem fehlerhaftem Schalten vorgebeugt.

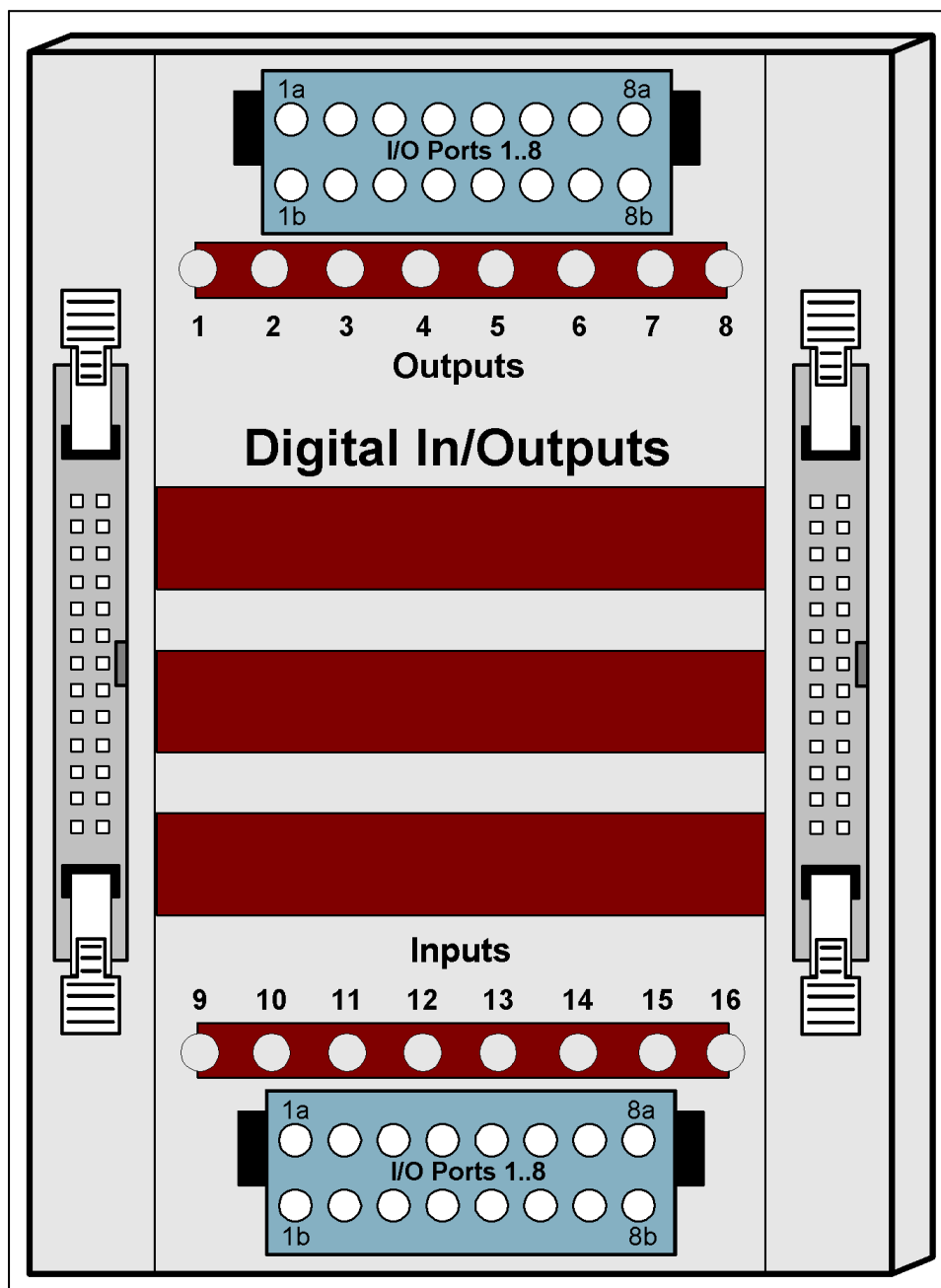
3.2.3. RO-OPT8-REL8

3.2.3.1. Übersichtsbild

Die Abbildung zeigt ein RO-O8-R8 Modul mit entsprechender Durchnummerierung der Anschlussklemmen.



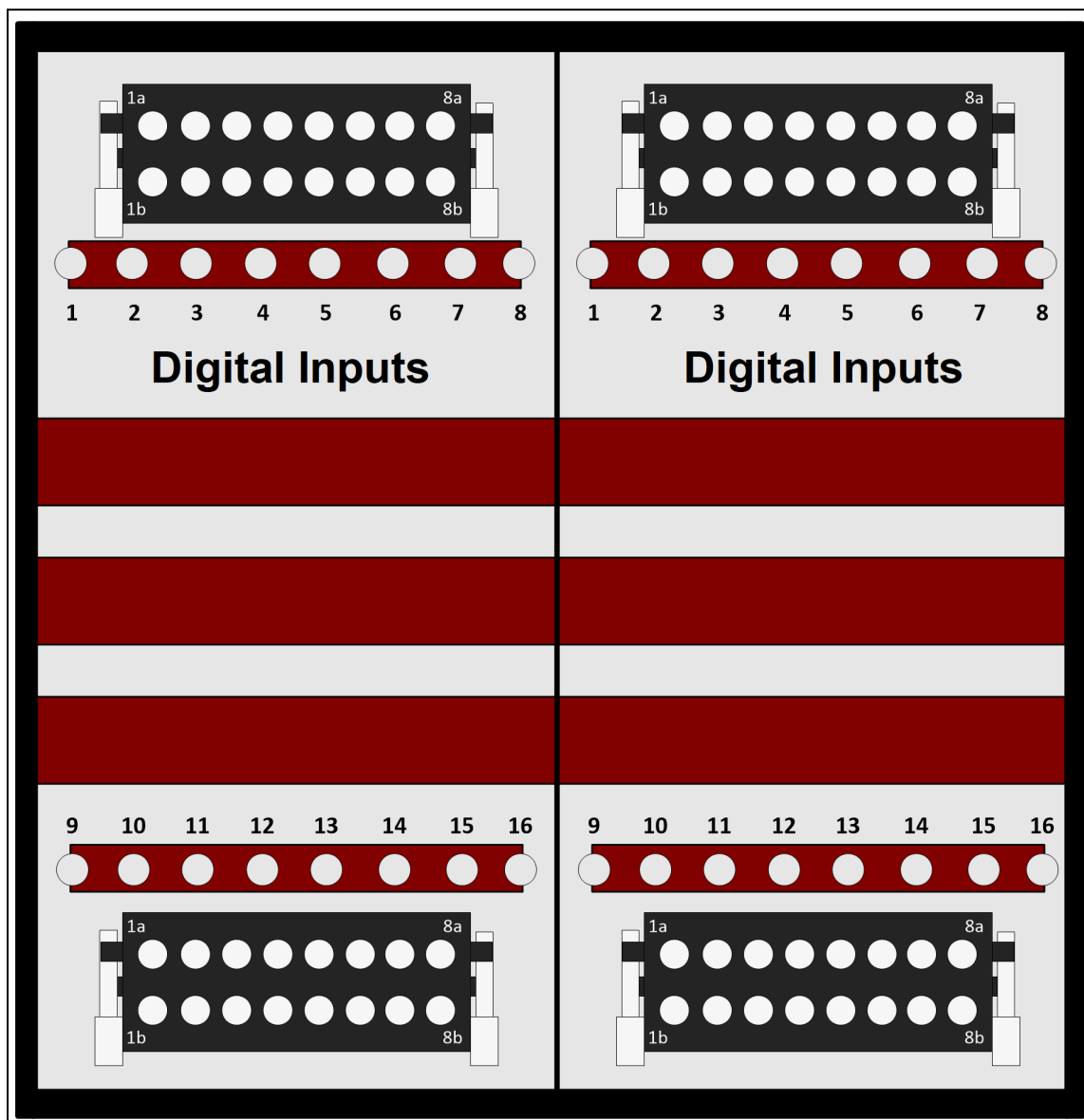
Untere Abbildung zeigt ein RO-08-R8 Modul des flexiblen Steckverbindersystems mit entsprechender Durchnummerierung der Anschlussklemmen. An den äußeren Enden des Moduls befindet sich jeweils eine 26pol. Wannenstiftleiste mit Verriegelungshebeln. Somit können mehrere Moduleinheiten mit Hilfe eines Flachbandkabels miteinander verbunden werden.



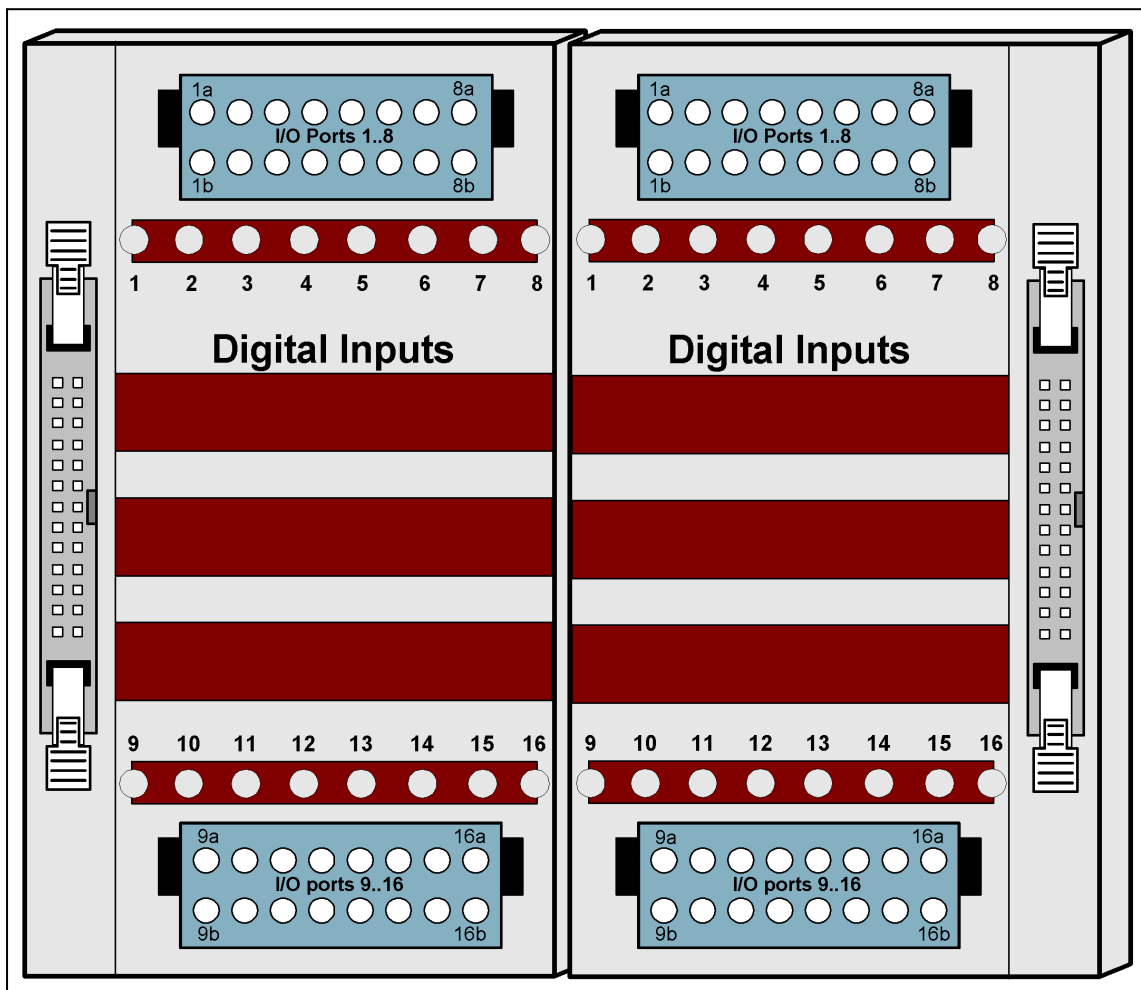
3.2.4. RO-OPT16

3.2.4.1. Übersichtsbild

Die Abbildung zeigt zwei Module nebeneinander mit entsprechender Durchnummerierung der Anschlussklemmen.



Untere Abbildung zeigt ein I/O-Modul des flexiblen Steckverbindersystems mit 32 Eingängen und entsprechender Durchnummerierung der Anschlussklemmen. An den äußeren Enden des Moduls befindet sich jeweils eine 26pol. Wannenstiftleiste mit Verriegelungshebeln. Somit können mehrere Moduleinheiten mit Hilfe eines Flachbandkabels miteinander verbunden werden.



3.2.4.2. Ändern der Eingangsspannung

Eingangsspannungs-bereich	5V – 15V	15V – 30V
Widerstandswert	1K	2K2

Um den Eingangsspannungsbereich anzupassen, müssen pro Steckverbinder 2 Vorwiderstände ausgetauscht werden. D.h., man kann jeweils 4 Eingangskanäle (1-4, 5-8, 9-12 und 13-16) mit unterschiedlicher Spannung einspeisen. Die einzelnen Schritte hierfür werden im Folgenden erläutert.

Hinweis!

Vor Öffnen des Gerätes bitte folgendes beachten:

Netzstecker ziehen!

Elektronische Bauteile nicht berühren, da diese durch elektrostatische Entladung zerstört werden können! Vor dem Arbeiten ggf. geerdete Gehäuse oder Heizkörper berühren.

Seitenelement des Moduls entfernen, dazu die drei Kreuzschrauben lösen

Leiterplatte samt Frontplatte seitlich aus dem Gehäuse herausziehen

Frontplatte abheben

Die Eingangsmodule verfügen alle über einreihige Sockelleisten, in denen die Eingangswiderstände für den Spannungsbereich eingesteckt sind

Widerstände nun vorsichtig aus den Sockelstreifen herausziehen

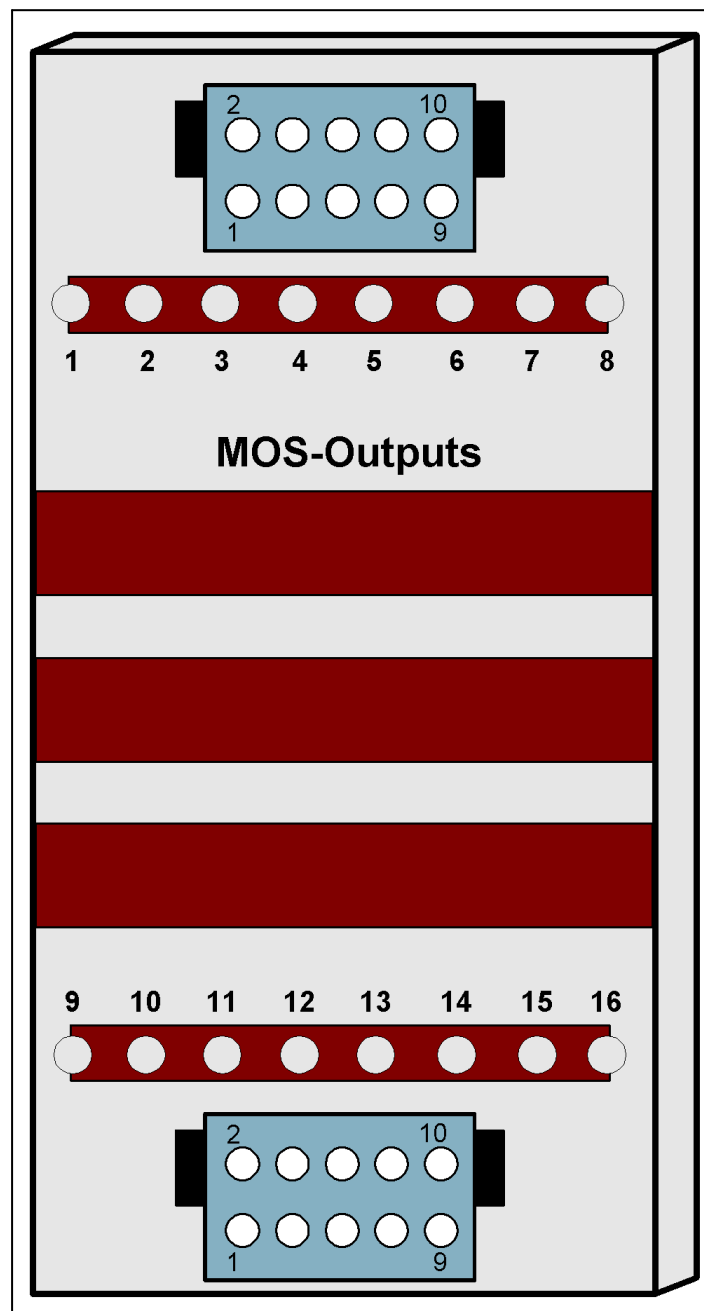
Neue Widerstände einsetzen

Einbau erfolgt in umgekehrter Reihenfolge

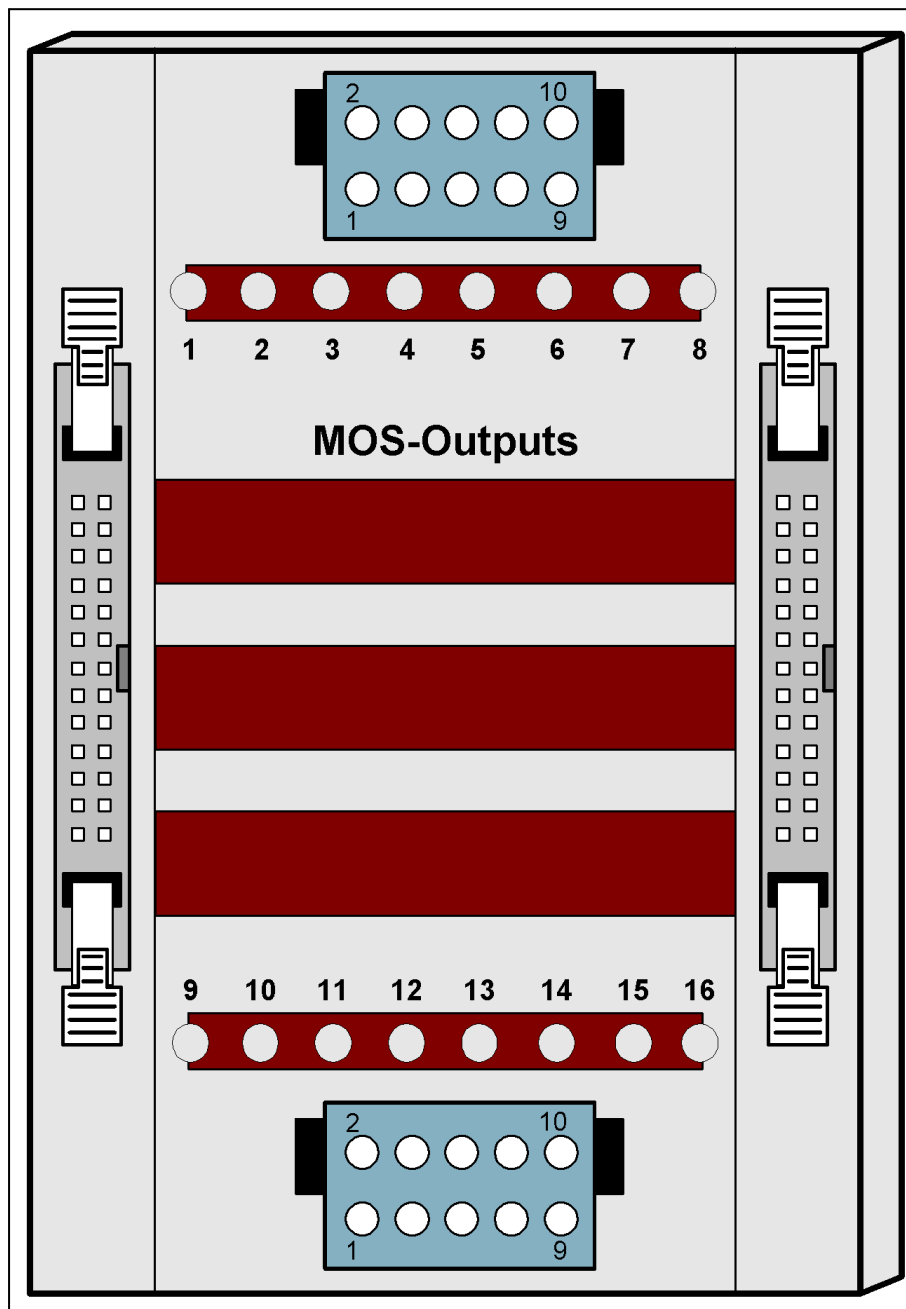
3.2.5. RO-MOS16

3.2.5.1. Übersichtsbild

Die Abbildung zeigt ein Modul mit entsprechender Durchnummerierung der Anschlussklemmen.



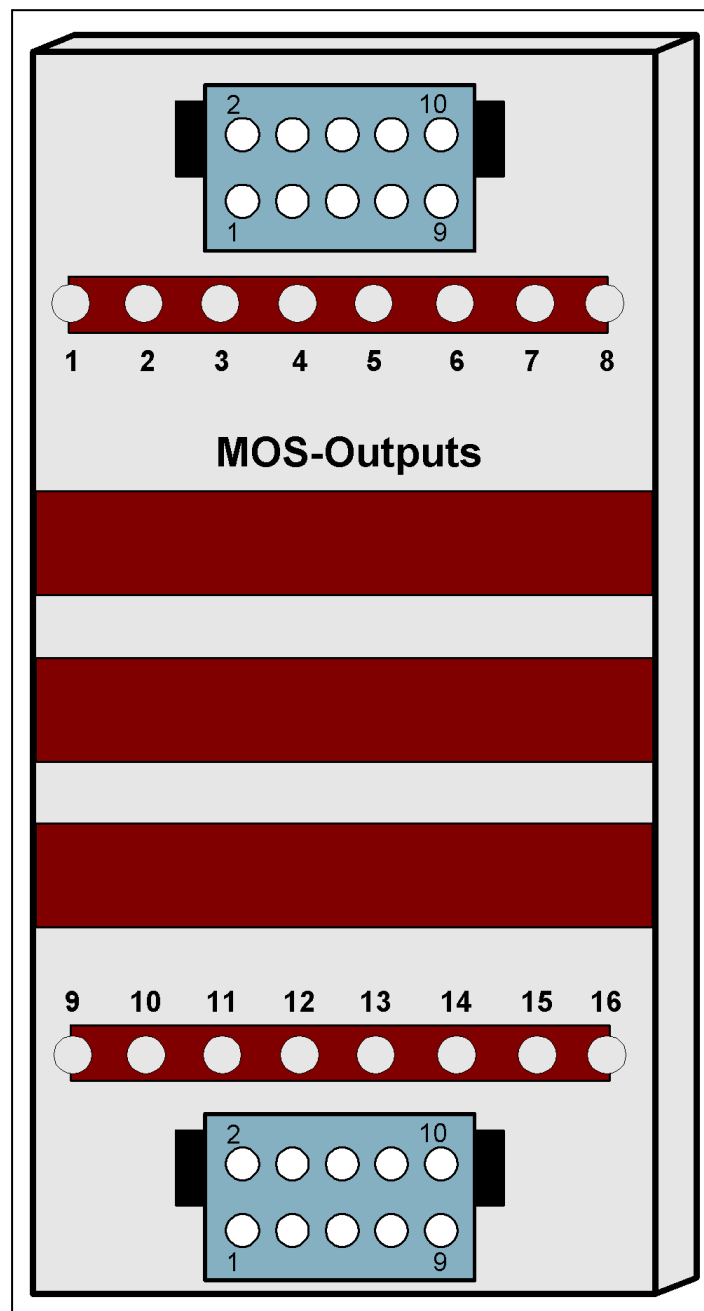
Untere Abbildung zeigt ein I/O-Modul des flexiblen Steckverbindersystems mit 16 Ausgängen und entsprechender Durchnummerierung der Anschlussklemmen. An den äußeren Enden des Moduls befindet sich jeweils eine 26pol. Wannenstiftleiste mit Verriegelungshebeln. Somit können mehrere Moduleinheiten mit Hilfe eines Flachbandkabels miteinander verbunden werden.



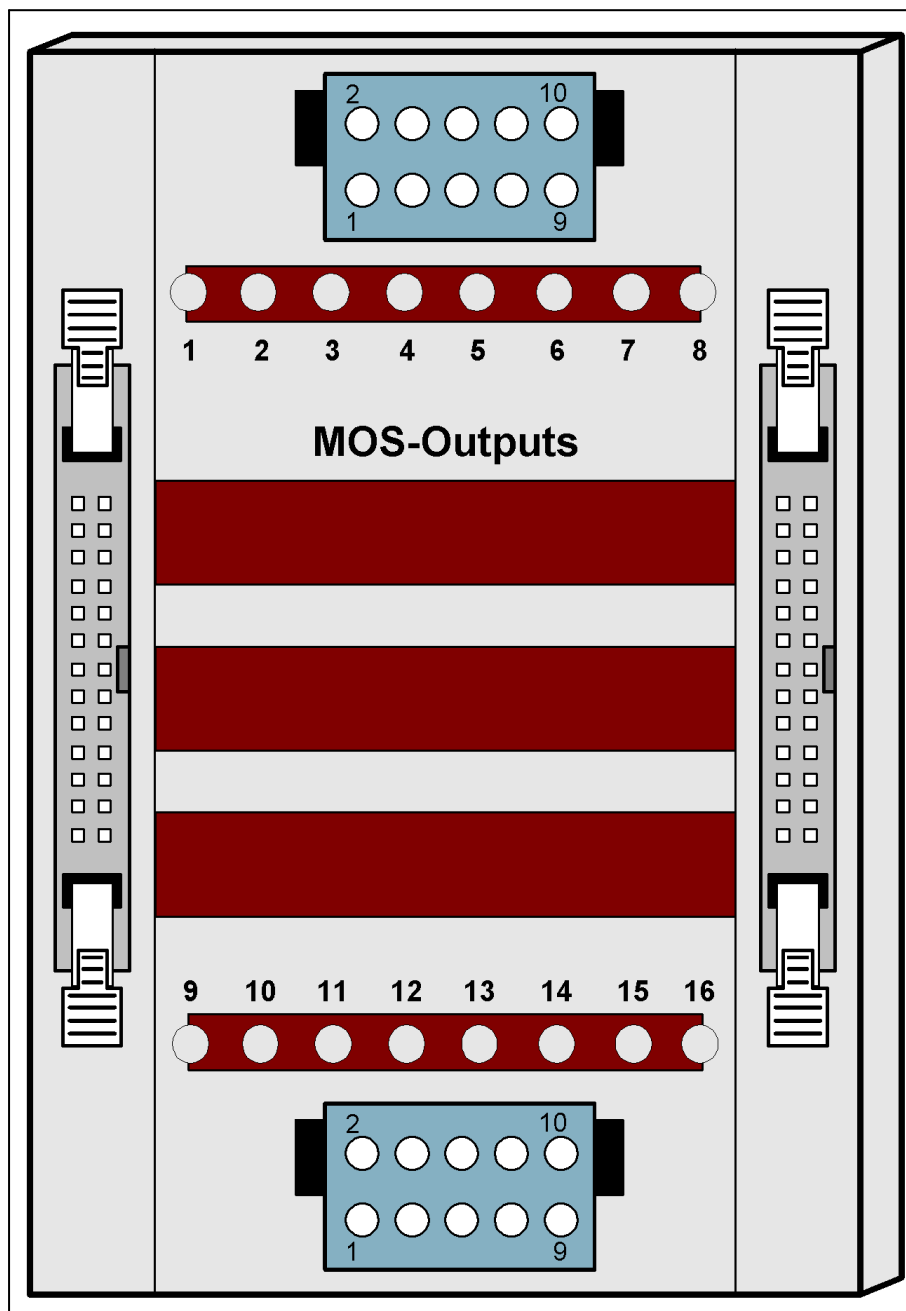
3.2.6. RO-PWM

3.2.6.1. Übersichtsbild

Die Abbildung zeigt ein Modul mit PWM-Ausgängen nebeneinander mit entsprechender Durchnummerierung der Anschlussklemmen.



Die untere Abbildung zeigt ein I/O-Modul des flexiblen Steckverbindersystems mit 16 MOSFET-Ausgänge mit Pulsweitenmodulation und entsprechender Durchnummerierung der Anschlussklemmen. An den äußeren Enden des Moduls befindet sich jeweils eine 26pol. Wannenstiftleiste mit Verriegelungshebeln. Somit können mehrere Moduleinheiten mit Hilfe eines Flachbandkabels miteinander verbunden werden.



3.2.7. Digitale Ausgänge

3.2.7.1. 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 Bistabile Relais

Anzahl Kanäle pro Block:	8
Typ:	Schließer (CO)
Max. Schaltspannung:	30V DC
Max. Schaltstrom:	2A DC
Max. Schaltleistung:	60W DC
Isolationwiderstand:	1000 MOhm @500V DC
Schalt- und Abfallzeit:	max. 4ms

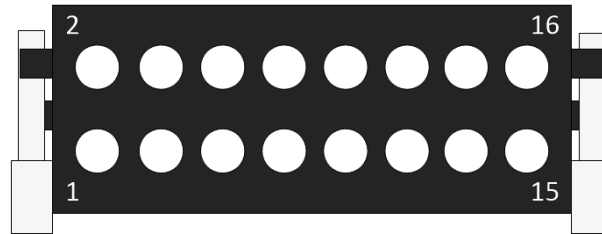
Technische Daten MOSFET 2A

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

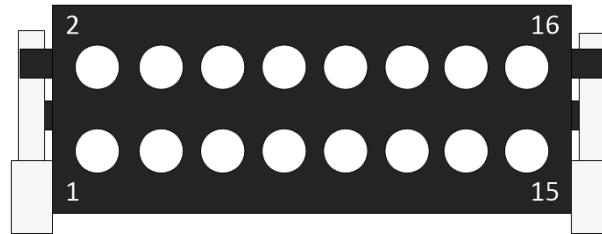
Timeout Schutz

3.2.7.2. Pinbelegung eines RO-REL16



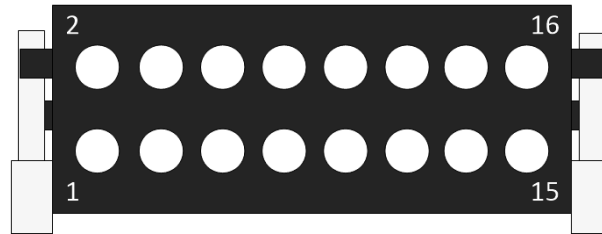
Digital Output Channel	Pin	
Output Channel 1	1	2
Output Channel 2	3	4
Output Channel 3	5	6
Output Channel 4	7	8
Output Channel 5	9	10
Output Channel 6	11	12
Output Channel 7	13	14
Output Channel 8	15	16

3.2.7.3. Pinbelegung eines RO-REL8_UM



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

3.2.7.4. Pinbelegung eines RO-MOS16_P und RO-PWM16_P



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

Bei den Optokopplerausgängen muss auf die richtige Polarität beim Anschluss geachtet werden (**siehe Bild unten**)

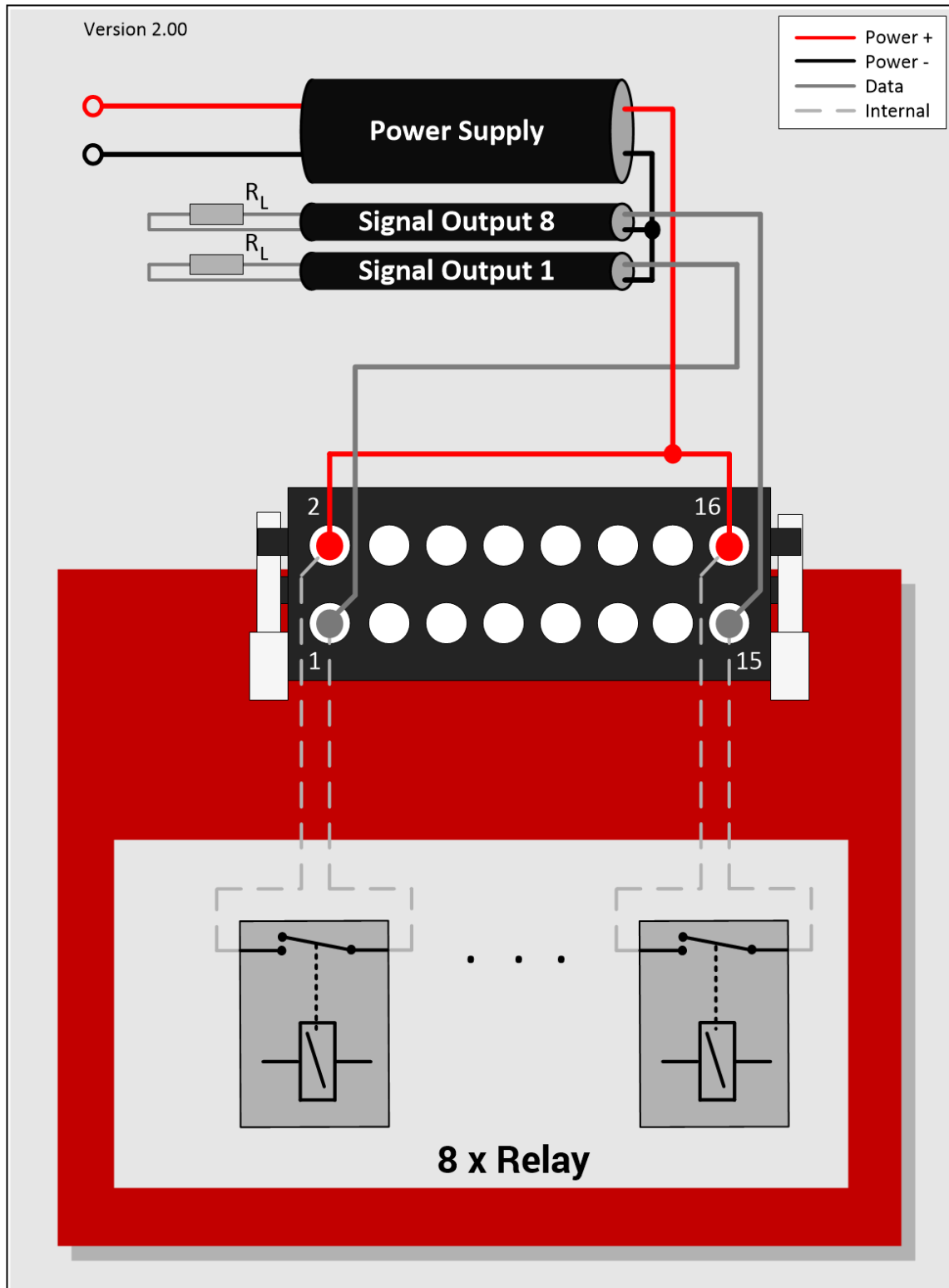
Achtung:

Jeder Steckverbinder verfügt über je 4 VCC+ Eingänge, die jeweils für eine Belastung von max.10A ausgelegt sind.

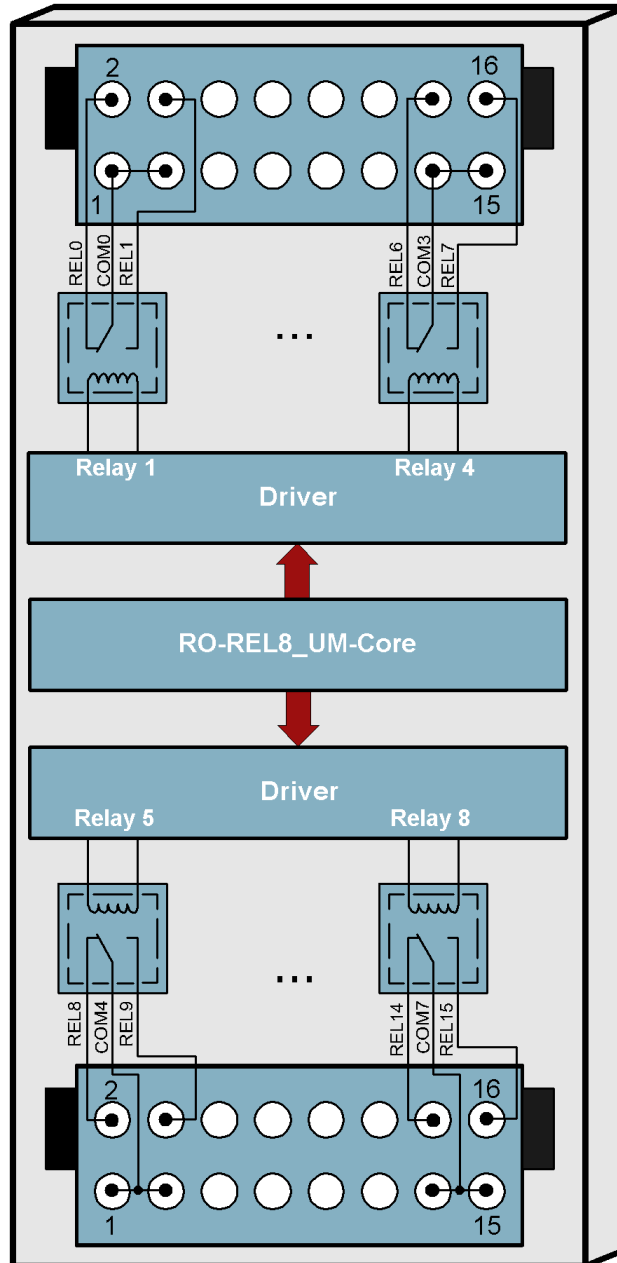
Sollte die Summenbelastung des Steckverbinders beispielsweise 30A sein, müssen 3 VCC+ Eingänge belegt werden.

Summenbelastung	Benötigte VCC+ Eingänge
<= 10A	1
<= 20A	2
<= 30A	3
<= 40A	4

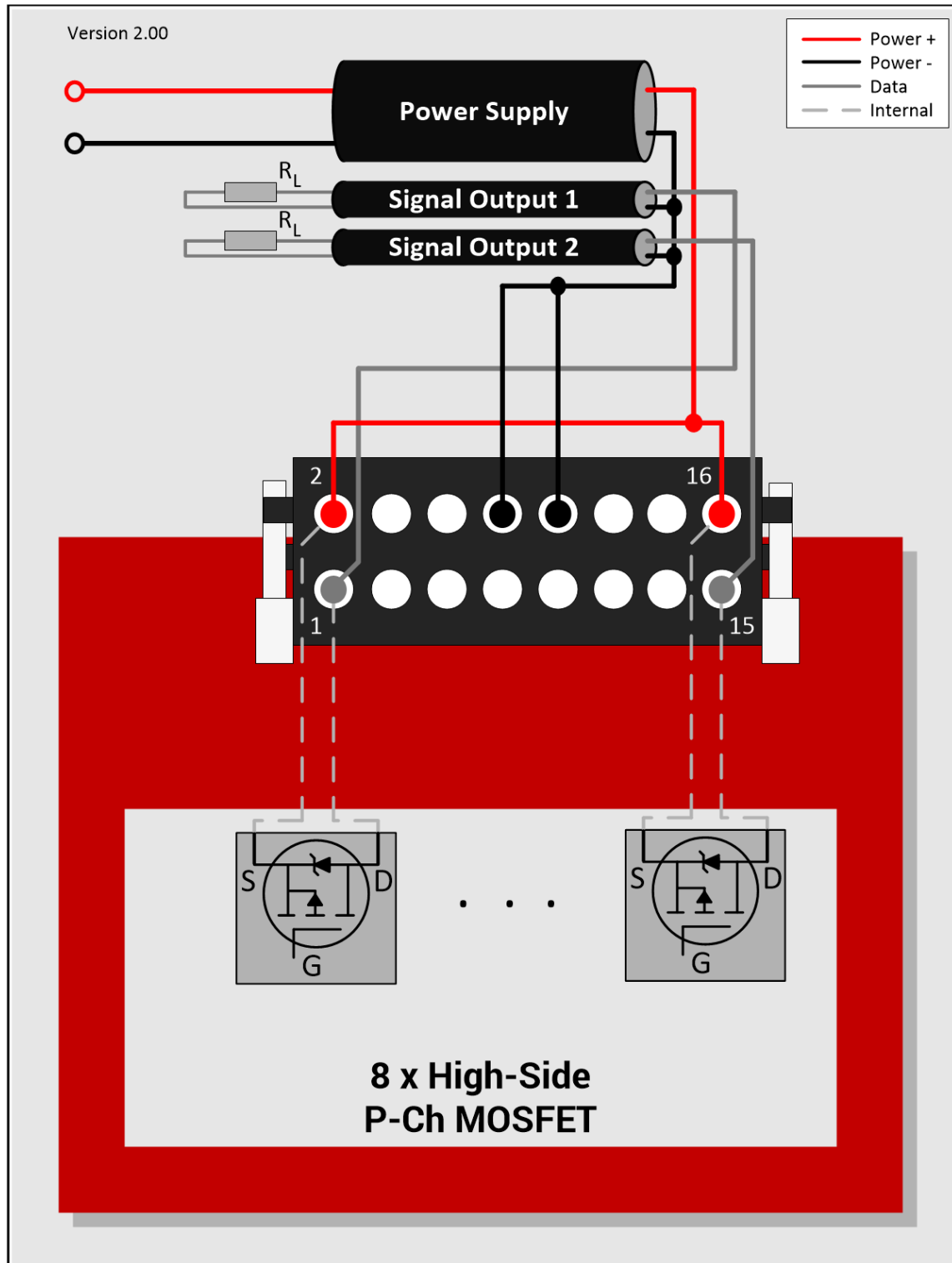
3.2.7.5. Anschlussbeispiel eines Relais-Moduls



3.2.7.6. Anschlussbeispiel eines Umschaltrelais-Moduls



3.2.7.7. Anschlussbeispiel eines MOSFET-Moduls



J:\Produkte\90-Anschlussbeispiele Produkte\MOS-PWM\wago-16_NET_COS\vsd\

3.2.7.8. 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**

3.2.8. Digitale Eingänge

3.2.8.1. Technische Daten

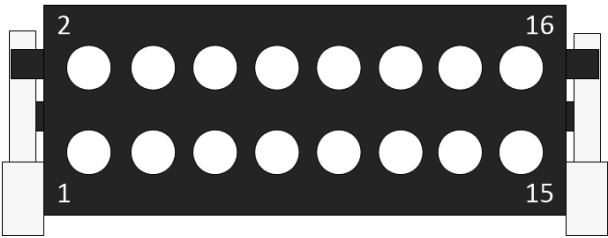
Technische Daten Optokoppler

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

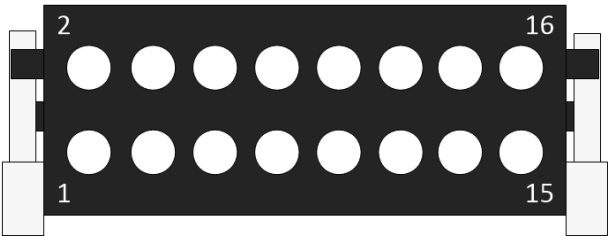
- 16 Bit Zähler je Kanal.
- Maximal mögliche Zählvorgänge: 65535 / Kanal. Nach Speicherüberlauf Reset auf Null.
- Interne Zähllogik bis 10kHz mit Latch Funktion.
- Programmierbarer Filter für Eingangskanäle (Flip-Flop und Counter): Mindest low bzw. high Impulsdauer: 5ms...255ms.
- Erkennt einen Wechsel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

3.2.8.2. Pinbelegung R0-016



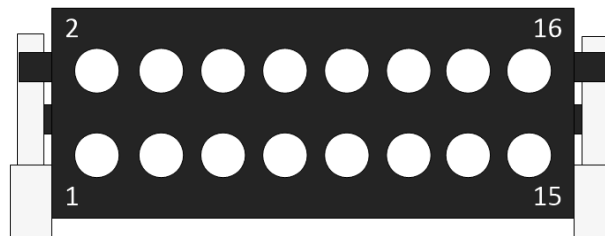
Digital Input Channel	Pin	
Input Channel 1	1	2
Input Channel 2	3	4
Input Channel 3	5	6
Input Channel 4	7	8
Input Channel 5	9	10
Input Channel 6	11	12
Input Channel 7	13	14
Input Channel 8	15	16

3.2.8.3. Pinbelegung RO-08-R8



Relais Ausgänge (Stecker oben)

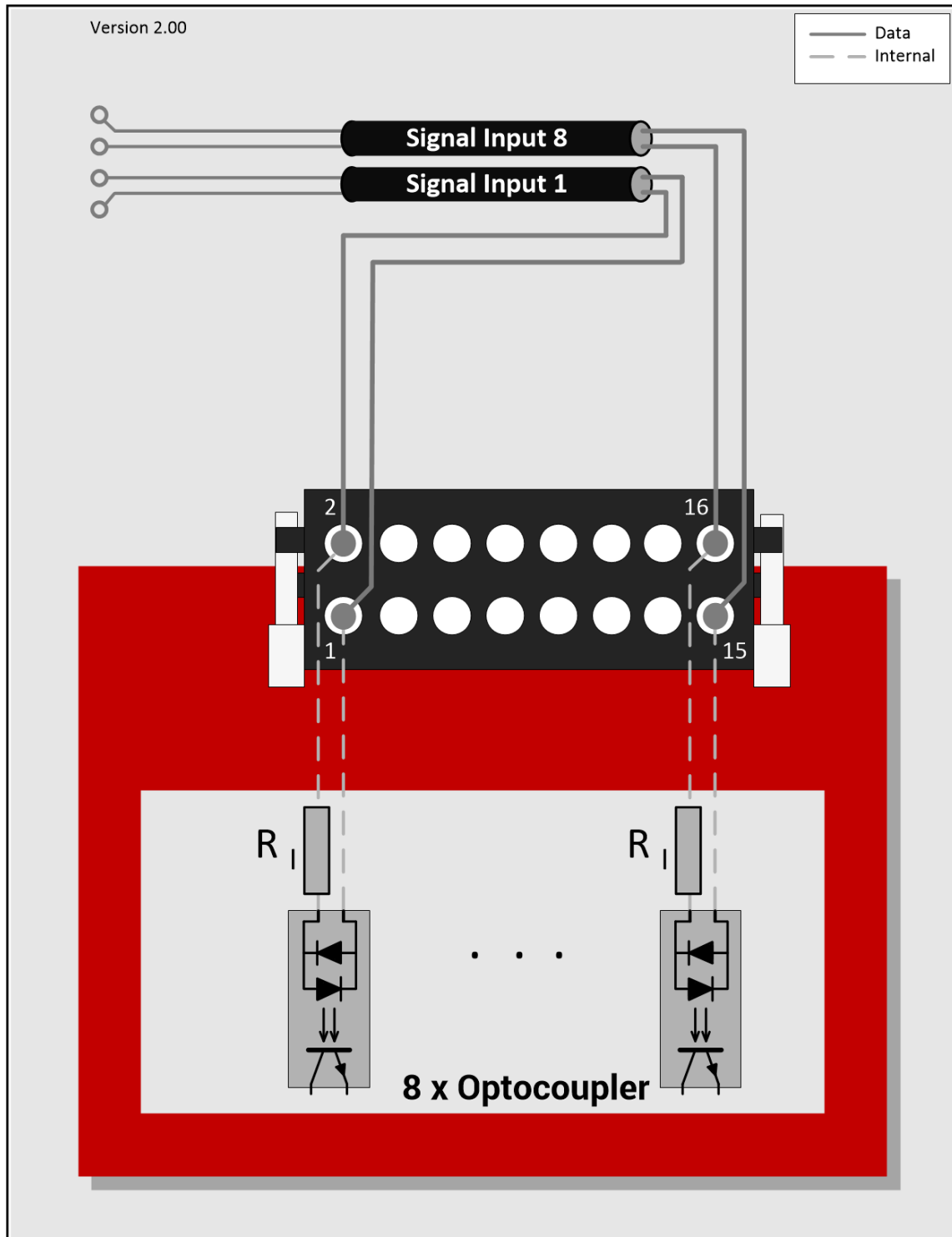
Digital Output Channel	Pin	
Output Channel 1	1	2
Output Channel 2	3	4
Output Channel 3	5	6
Output Channel 4	7	8
Output Channel 5	9	10
Output Channel 6	11	12
Output Channel 7	13	14
Output Channel 8	15	16



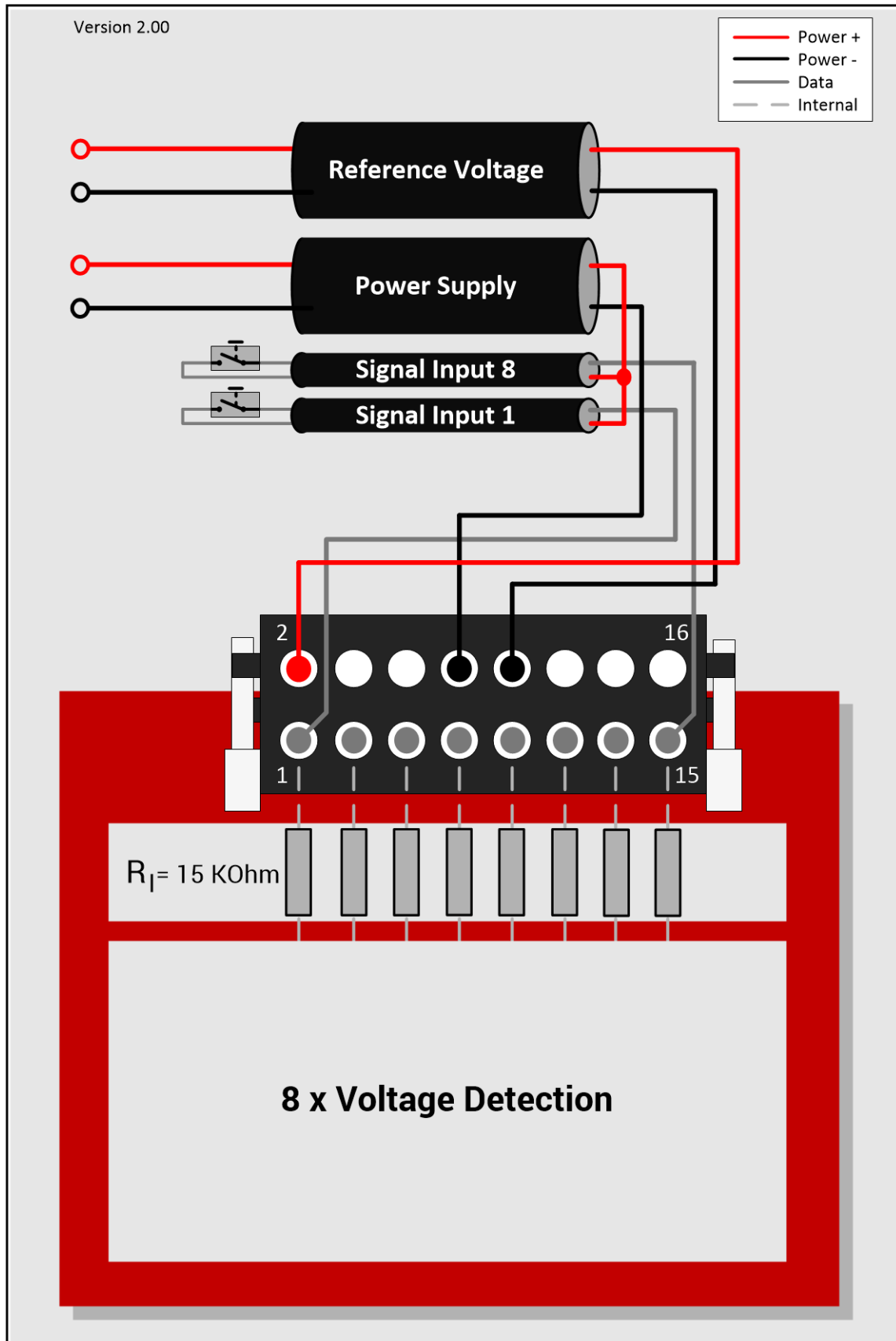
Optokoppler Eingänge (Stecker unten)

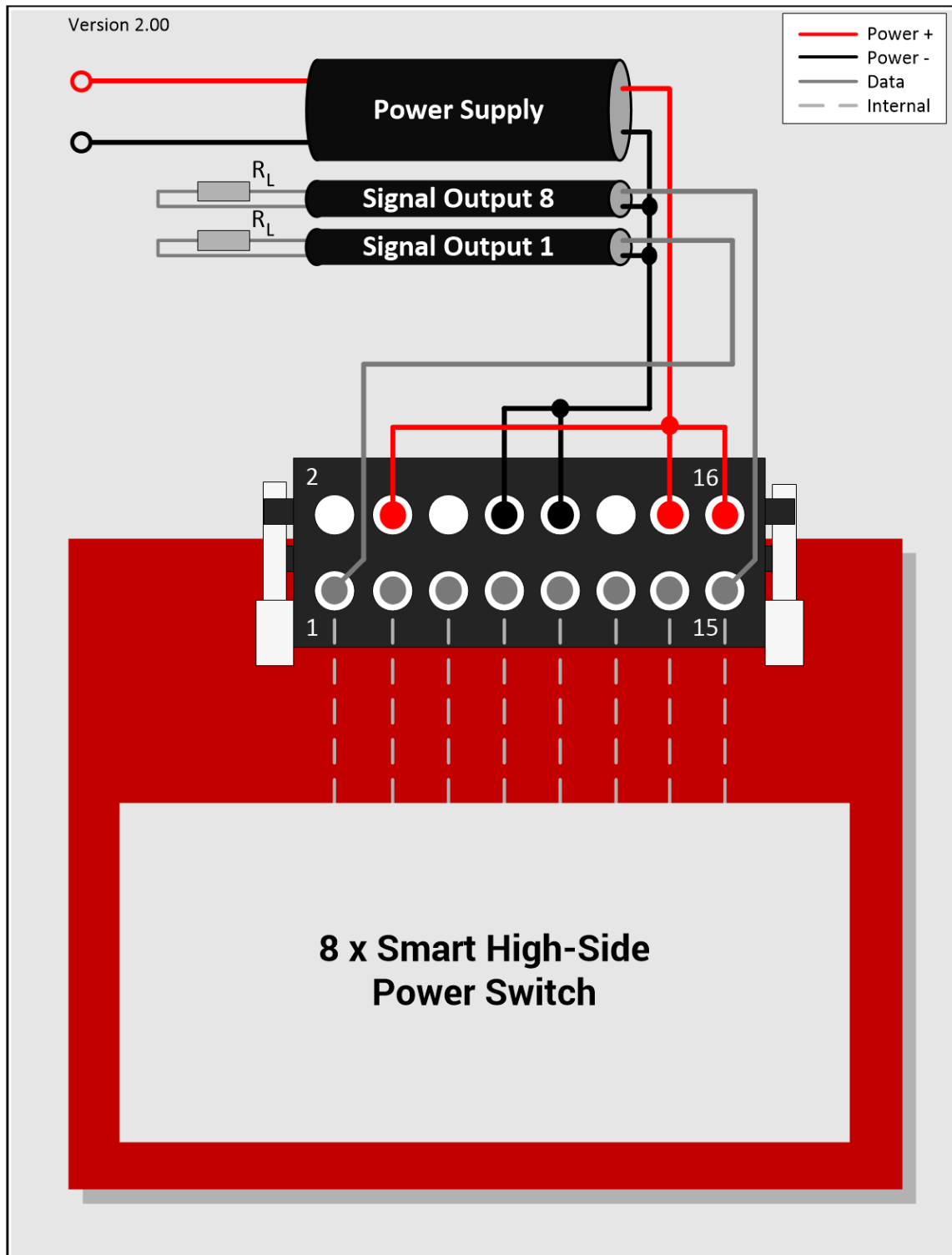
Digital Input Channel	Pin	
Input Channel 1	1	2
Input Channel 2	3	4
Input Channel 3	5	6
Input Channel 4	7	8
Input Channel 5	9	10
Input Channel 6	11	12
Input Channel 7	13	14
Input Channel 8	15	16

3.2.8.4. Anschlussbeispiel R0-016



3.2.8.5. Anschlussbeispiel R0-08-R8



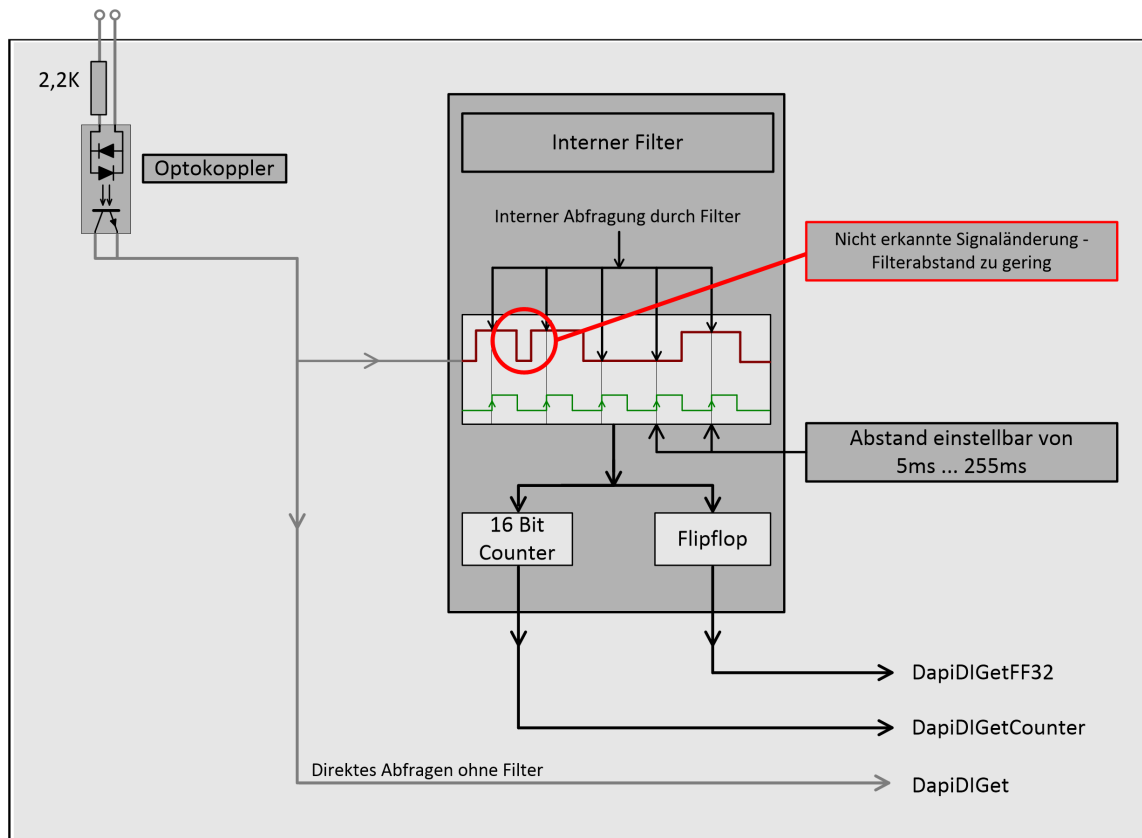


3.2.8.6. EingangsfILTER

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

Siehe auch Kapitel → **Digitale Eingänge Lesen**

Schematische Ansicht des Filters:



3.2.8.7. 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**

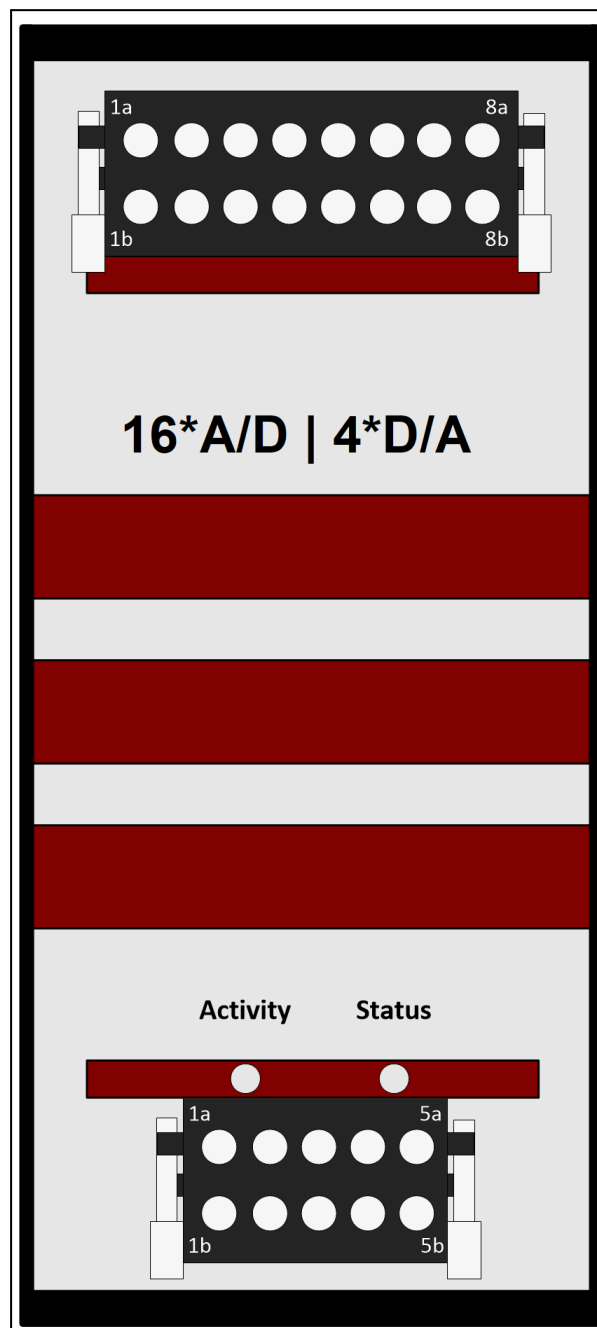
3.2.9. Analoge Ein-/Ausgabemodule

3.2.9.1. RO-AD16-DA4

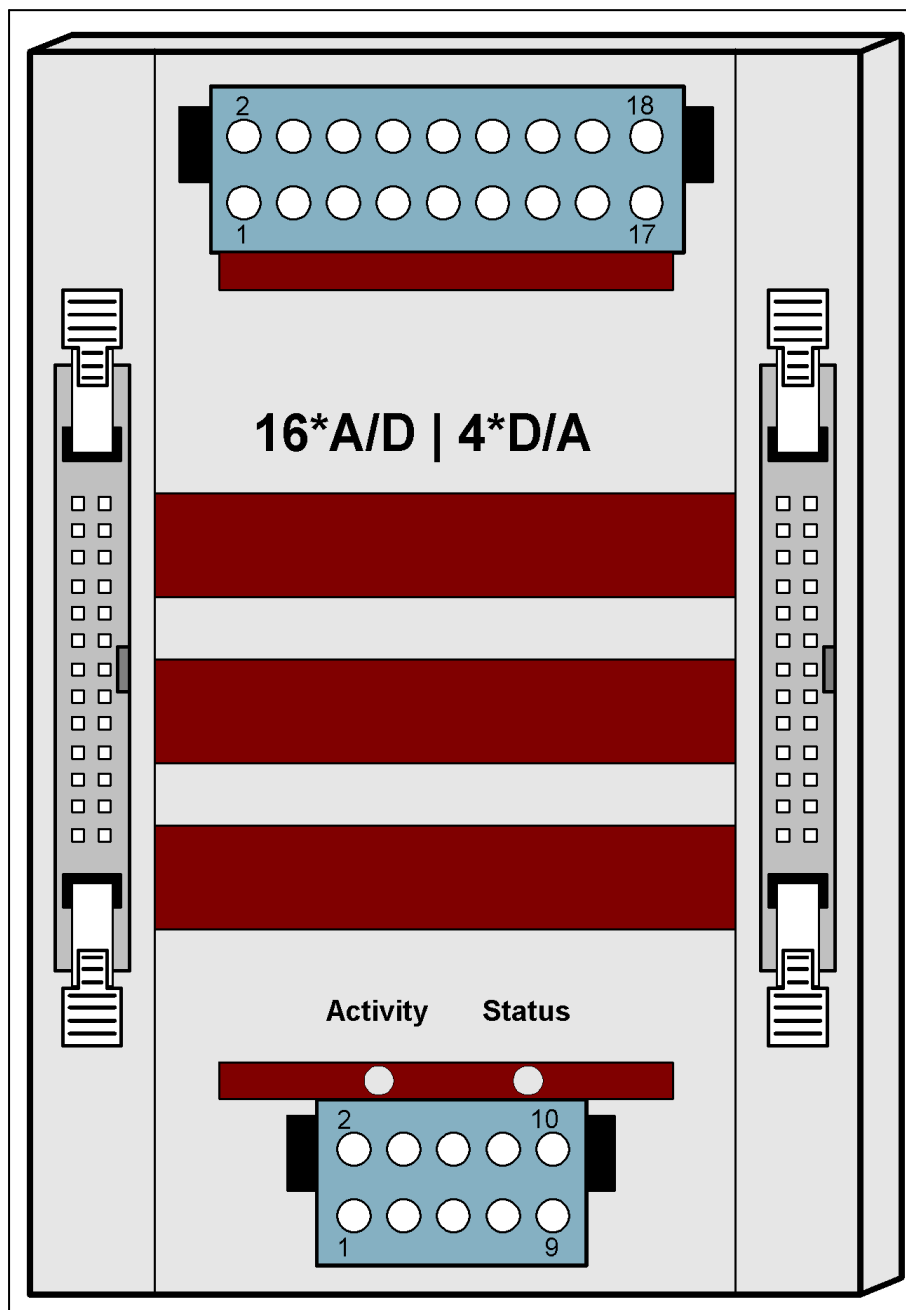
Mit 16 A/D Eingängen stellt dieses Modul eine gute Basis zur Umwandlung von Spannungen in digitale Werte dar. Zusätzlich dank der 4 D/A Ausgänge ist es möglich digitale Werte in Spannungen umzuwandeln.

3.2.9.1.1. Übersichtsbild

Die Abbildung zeigt ein Modul mit entsprechender Durchnummerierung der Anschlussklemmen.



Die Abbildung zeigt ein A/D-Modul des flexiblen Steckverbindersystems mit entsprechender Durchnummerierung der Anschlussklemmen.



3.2.9.1.2. Technische Daten

A/D Eingänge

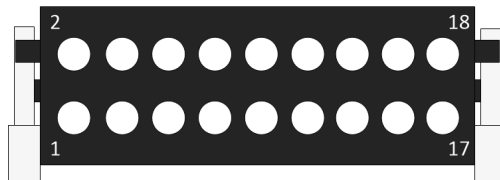
Modus U:	(Spannung)
Unipolar:	0-5V, 0-10V
Bipolar:	+5V, +10V
Modus I:	(Strom)
Bereich:	0-20mA (optional)

D/A Ausgänge

Modus U:	(Spannung)
Unipolar:	0-5V, 0-10V
Bipolar:	+5V, +10V

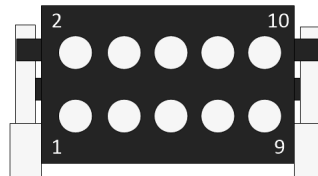
3.2.9.1.3. Pinbelegung

3.2.9.1.3.1. A/D Steckverbinder (18pol)



Pin	Analog Channel	Input	Pin	Analog Channel	Input
1	AGND		2	AGND	
3	AD1		4	AD0	
5	AD3		6	AD2	
7	AD5		8	AD4	
9	AD7		10	AD6	
11	AD9		12	AD8	
13	AD11		14	AD10	
15	AD13		16	AD12	
17	AD15		18	AD14	

3.2.9.1.3.2. D/A Steckverbinder (10pol)



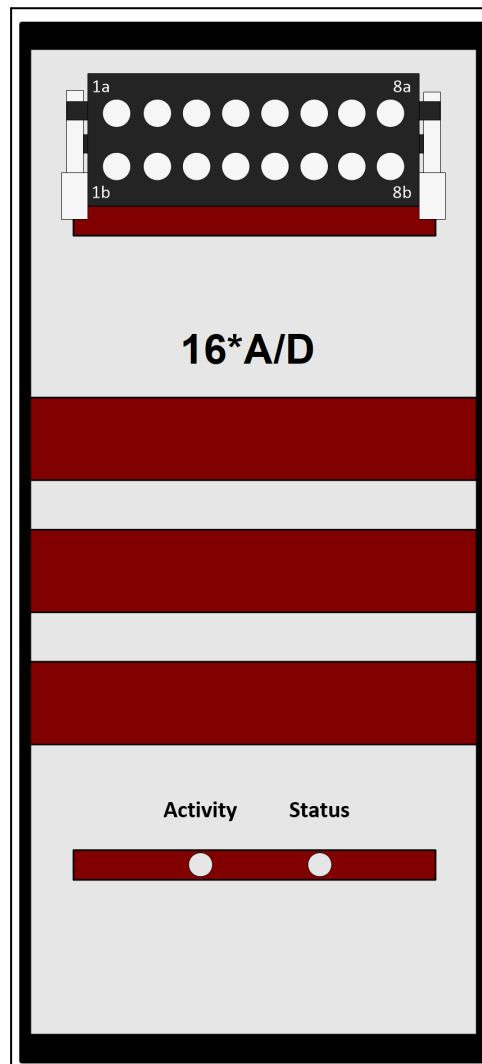
Pin	Analog Channel	Output	Pin	Analog Channel	Output
1	AGND		2	DA0	
3	AGND		4	DA1	
5	AGND		6	DA2	
7	AGND		8	DA3	
9	AGND		10	AGND	

3.2.9.2. RO-AD16

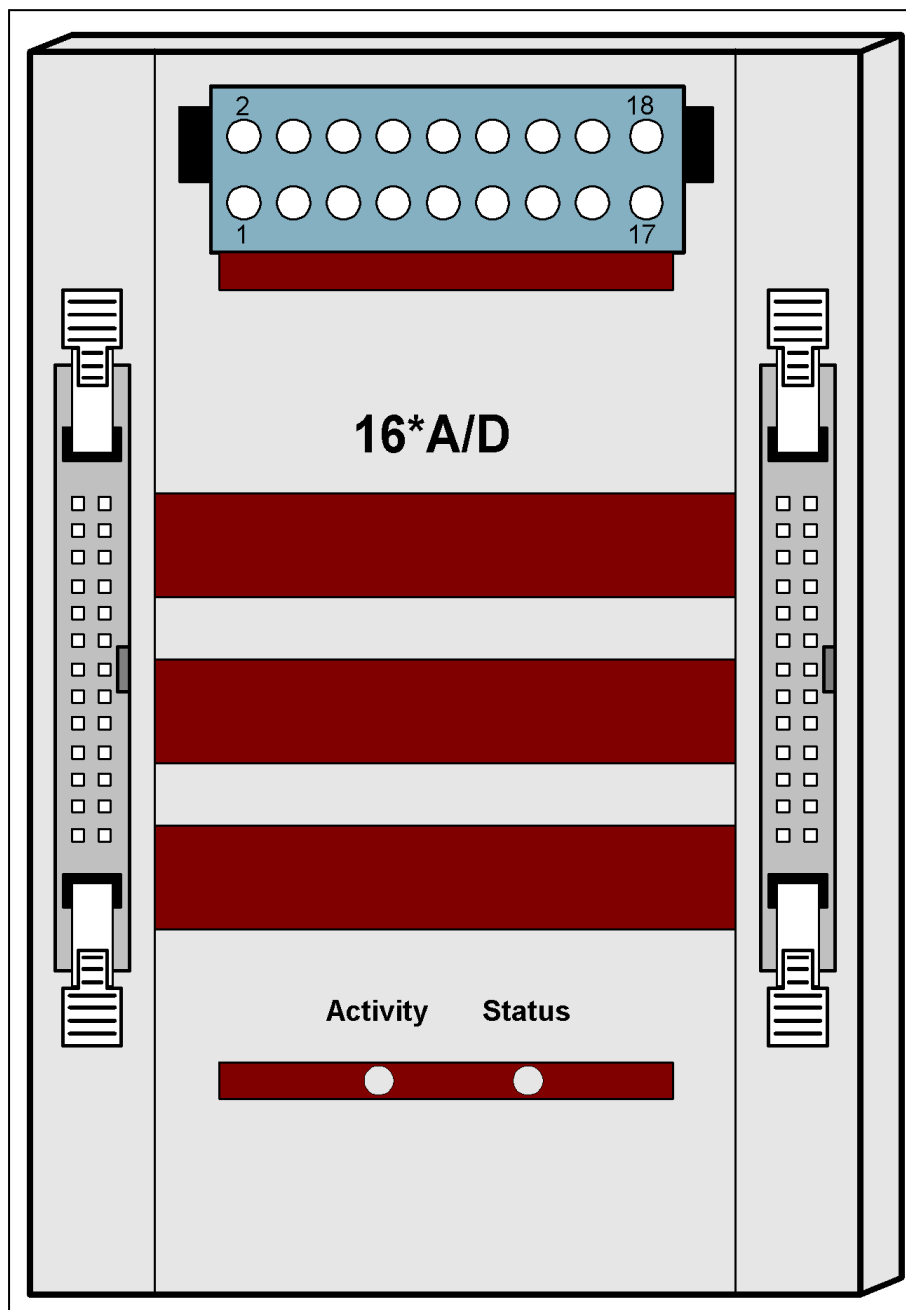
Mit 16 A/D Eingangs-Kanäle stellt dieses Modul eine gute Basis zur Umwandlung von Spannungen in digitale Werte dar.

3.2.9.2.1. Übersichtsbild

Die Abbildung zeigt ein Modul mit entsprechender Durchnummerierung der Anschlussklemmen.



Die Abbildung zeigt ein I/O-Modul des flexiblen Steckverbindersystems mit entsprechender Durchnummerierung der Anschlussklemmen.



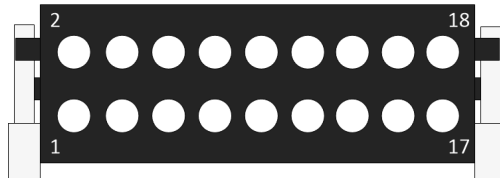
3.2.9.2.2. Technische Daten

A/D Eingangs-Kanäle

Modus U:	(Spannung)
Unipolar:	0-5V, 0-10V
Bipolar:	+5V, +10V
Modus I:	(Strom)
Bereich:	0-20mA (optional)

3.2.9.2.3. Pinbelegung

3.2.9.2.3.1. A/D Steckverbinder (18pol)



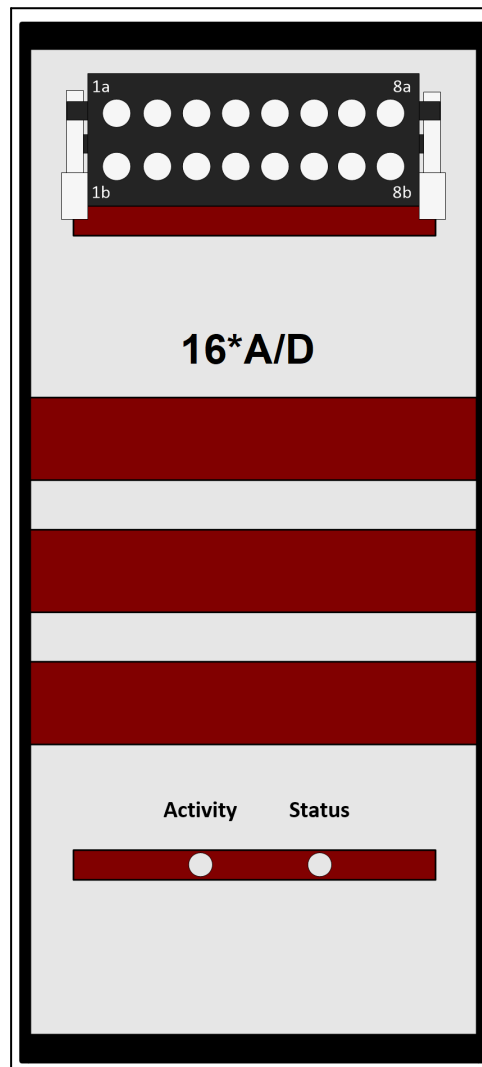
Pin	Analog Channel	Input	Pin	Analog Channel	Input
1	AGND		2	AGND	
3	AD1		4	AD0	
5	AD3		6	AD2	
7	AD5		8	AD4	
9	AD7		10	AD6	
11	AD9		12	AD8	
13	AD11		14	AD10	
15	AD13		16	AD12	
17	AD15		18	AD14	

3.2.9.3. RO-AD16_ISO

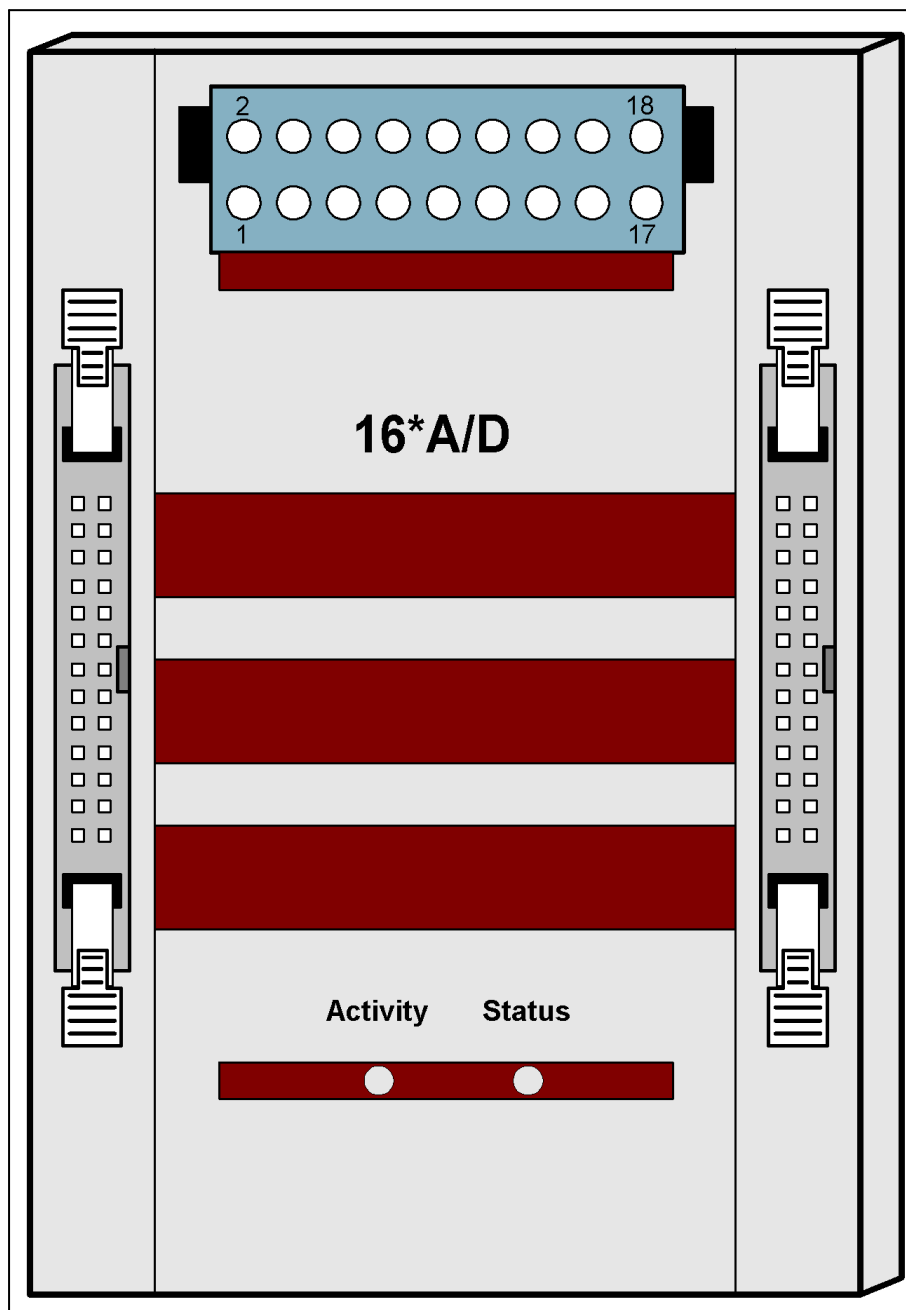
Mit 16 A/D Eingangs-Kanälen (galvanisch getrennt) stellt dieses Modul eine gute Basis zur Umwandlung von Spannungen in digitale Werte dar.

3.2.9.3.1. Übersichtsbild

Die Abbildung zeigt ein Modul mit entsprechender Durchnummerierung der Anschlussklemmen.



Die Abbildung zeigt ein I/O-Modul des flexiblen Steckverbindersystems mit entsprechender Durchnummerierung der Anschlussklemmen.



3.2.9.3.2. Technische Daten

A/D Eingangs-Kanäle

Galvanisch getrennt

Modus U: (Spannung)

Unipolar: 0-5V, 0-10V

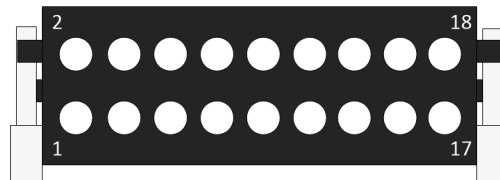
Bipolar: +5V, +10V

Modus I: (Strom)

Bereich: 0-20mA (optional)

3.2.9.3.3. Pinbelegung

3.2.9.3.3.1. A/D Steckverbinder (18pol)



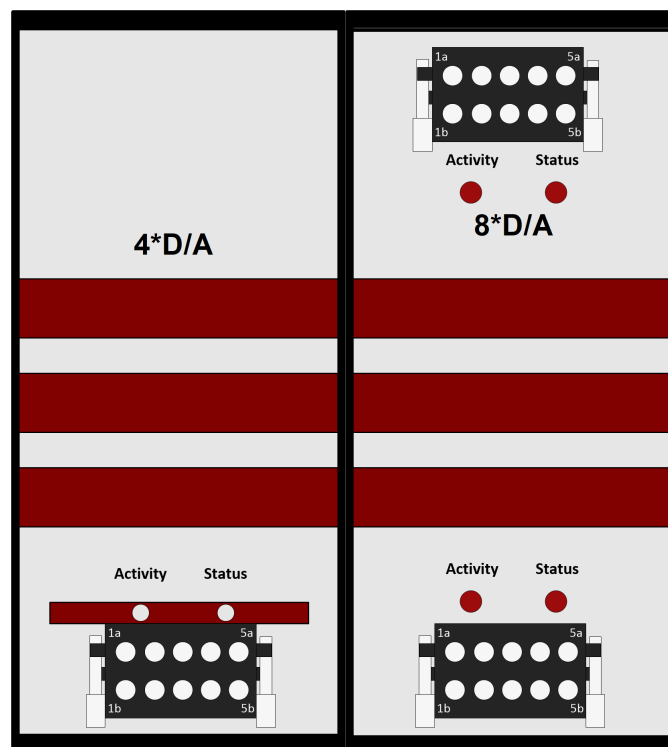
Pin	Analog Channel	Input	Pin	Analog Channel	Input
1	AGND		2	AGND	
3	AD1		4	AD0	
5	AD3		6	AD2	
7	AD5		8	AD4	
9	AD7		10	AD6	
11	AD9		12	AD8	
13	AD11		14	AD10	
15	AD13		16	AD12	
17	AD15		18	AD14	

3.2.9.4. RO-DA4/RO-DA8

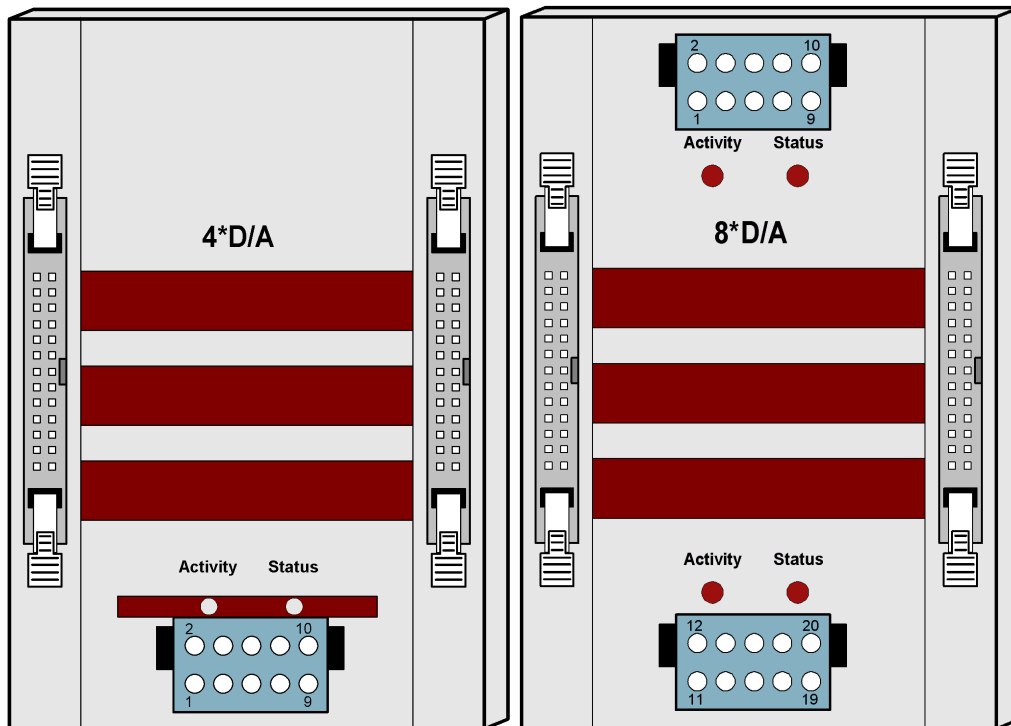
Mit 4/8 D/A Ausgängen stellen dieses Module eine gute Basis zur Umwandlung von digitalen Werten in Spannungen dar.

3.2.9.4.1. Übersichtsbild

Die Abbildung zeigt ein Modul mit entsprechender Durchnummerierung der Anschlussklemmen bei einem D/A 4.



Die Abbildung zeigt ein I/O-Modul des flexiblen Steckverbindersystems mit entsprechender Durchnummerierung der Anschlussklemmen.



3.2.9.4.2. Technische Daten

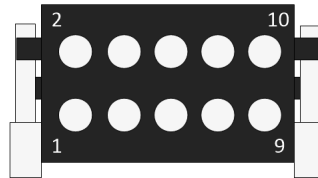
D/A Ausgänge

Unipolar: 0-5V, 0-10V

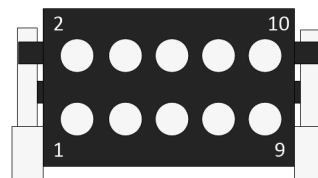
Bipolar: +5V, +10V

3.2.9.4.3. Pinbelegung

3.2.9.4.3.1. D/A Steckverbinder (10pol)



Pin		Pin	
1	AGND	2	DA0
3	AGND	4	DA1
5	AGND	6	DA2
7	AGND	8	DA3
9	AGND	10	AGND



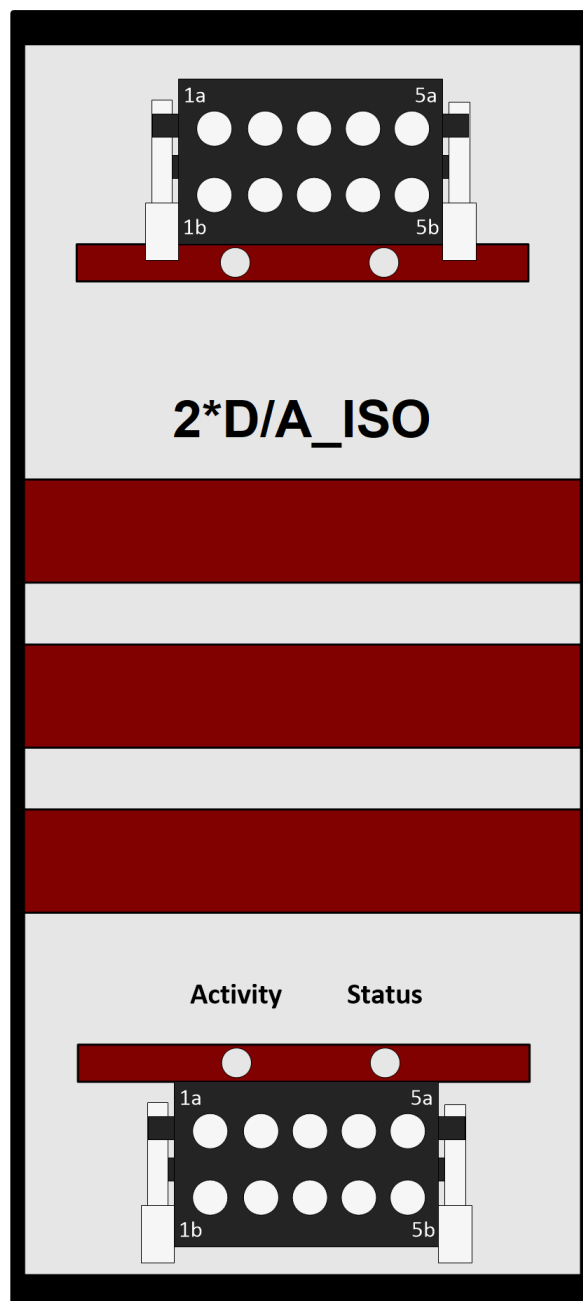
Pin		Pin	
11	AGND	12	DA4
13	AGND	14	DA5
15	AGND	16	DA6
17	AGND	18	DA7
19	AGND	20	AGND

3.2.9.5. RO-DA2_ISO

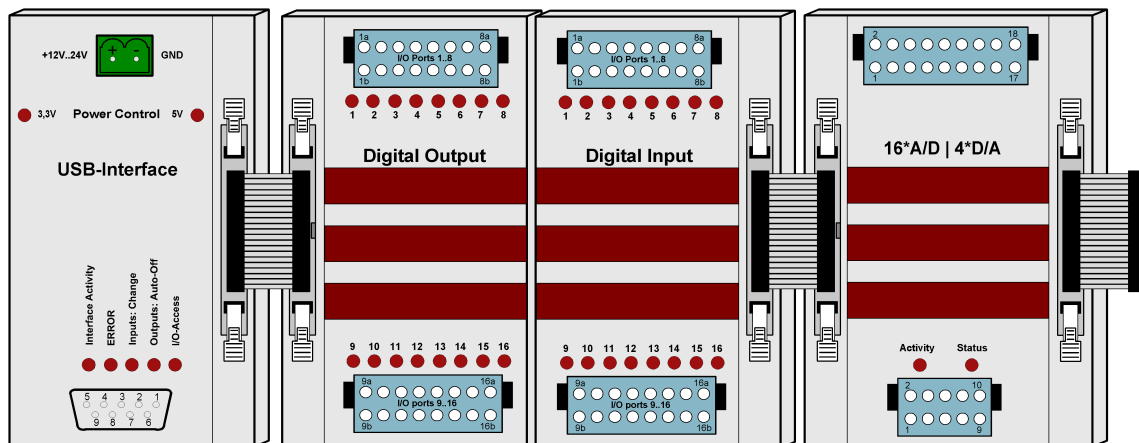
Mit 2 galvanisch getrennten D/A Ausgängen stellt dieses Modul eine gute Basis zur Umwandlung von digitalen Werten in Spannungen dar.

3.2.9.5.1. Übersichtsbild

Die Abbildung zeigt ein Modul mit entsprechender Durchnummerierung der Anschlussklemmen.



Die Abbildung zeigt ein I/O-Modul des flexiblen Steckverbindersystems mit entsprechender Durchnummerierung der Anschlussklemmen.



3.2.9.5.2. Technische Daten

D/A Ausgänge

Timeout Schutz

Modus U: (Spannung)

Unipolar: 0V-5V, 0V-10V

Bipolar: +5V, +10V

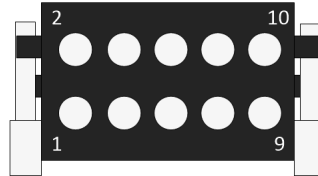
Modus I: (Strom)

0-20mA, 4-20mA, 0-24mA

3.2.9.5.3. Pinbelegung

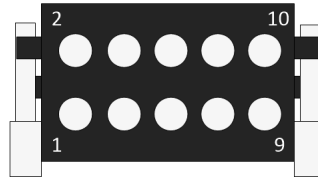
3.2.9.5.3.1. D/A Steckverbinder (10pol)

Steckverbinder oben:



Pin		Pin	
1	VOUT_A	2	+Vsense_A
3	VOUT_A	4	+Vsense_A
5	AGND	6	-Vsense_A
7	AGND	8	-Vsense_A
9	AGND	10	IOUT_A

Steckverbinder unten:

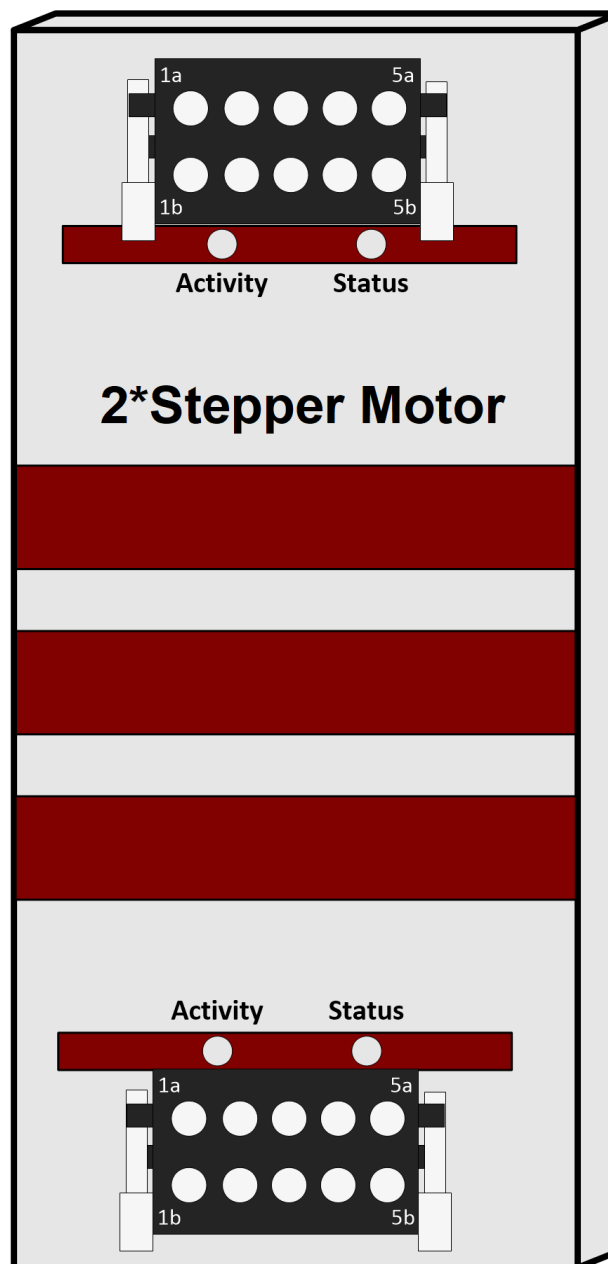


Pin		Pin	
1	VOUT_B	2	+Vsense_B
3	VOUT_B	4	+Vsense_B
5	AGND	6	-Vsense_B
7	AGND	8	-Vsense_B
9	AGND	10	IOOUT_B

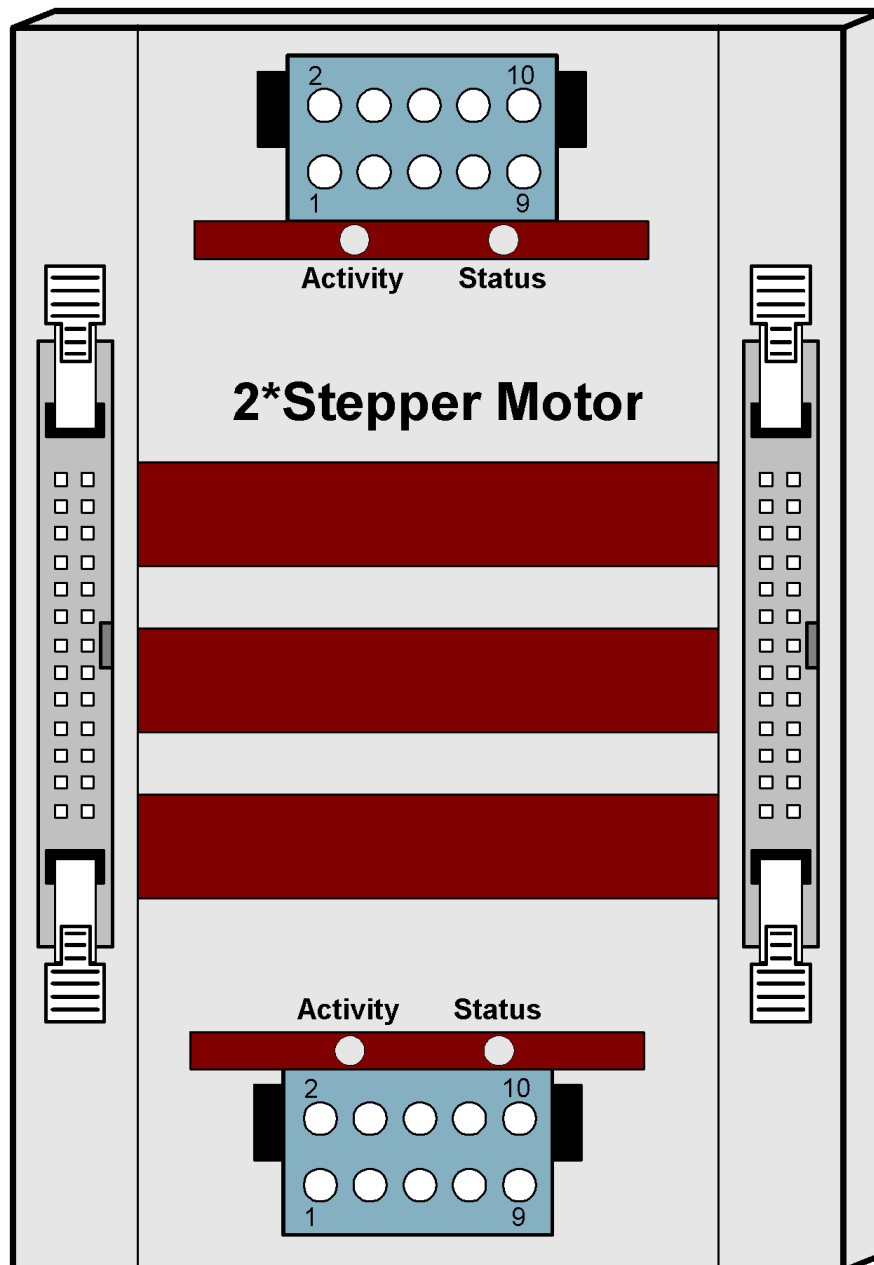
3.2.10. Stepper Modul

3.2.10.1. Übersichtsbild

Die Abbildung zeigt ein Modul mit entsprechender Durchnummerierung der Anschlussklemmen.



Die Abbildung zeigt ein Stepper-Modul des flexiblen Steckverbindersystems mit entsprechender Durchnummerierung der Anschlussklemmen.



3.2.10.2. Technische Daten

Allgemeines:

- Modul für zwei Schrittmotoren
- Komfortables Steckverbindersystem mit Auswerfmechanik
- Kann problemlos mit anderen Modulen der RO Serie kombiniert werden

Einstellbare Parameter:

- Start-/ Stopfrequenz
- Maximale Schrittfrequenz
- Beschleunigungsrampe
- Bremsrampe
- Phasenstrom
- Haltestrom
- Haltezeit

3.2.10.3. Ansteuerung

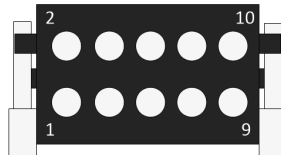
Alle Parameter lassen sich bequem über die DELIB Library einstellen.

Zwei Referenzschalter dienen zum Anfahren zu Referenzpositionen. Weitere zwei Endschalter sorgen für ein sicheres Anhalten und ausschließliches Fahren in entgegengesetzter Richtung, wenn diese betätigt werden.

3.2.10.4. Pinbelegung

3.2.10.4.1. Stepper Steckverbinder (10pol)

Pinbelegung eines Steckverbinders und zugleich eines Schrittmotors:



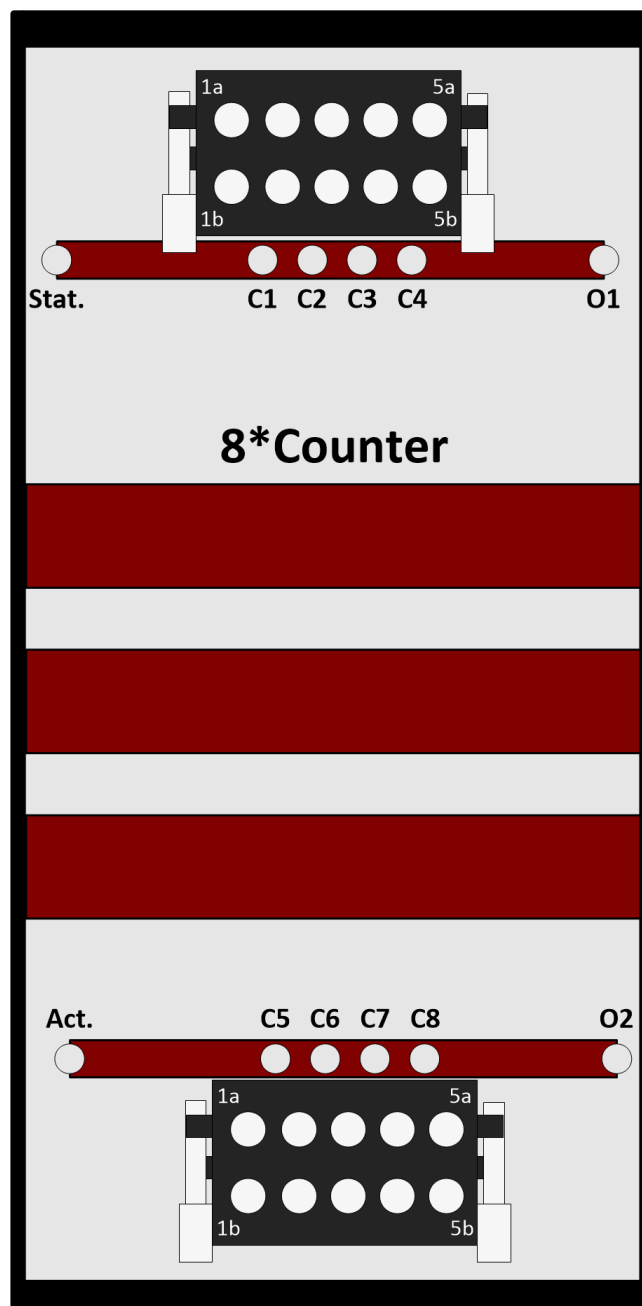
Pin		Pin	
1	24 V (Motorversorgung)	2	0 V (Motorversorgung)
3	Phase 1 (+)	4	Referenzschalter 2*)
5	Phase 1 (-)	6	Referenzschalter 1*)
7	Phase 2 (+)	8	Endschalter 2*)
9	Phase 2 (-)	10	Endschalter 1*)

*) Die Schalter sind gegen 24 V zu verschalten.

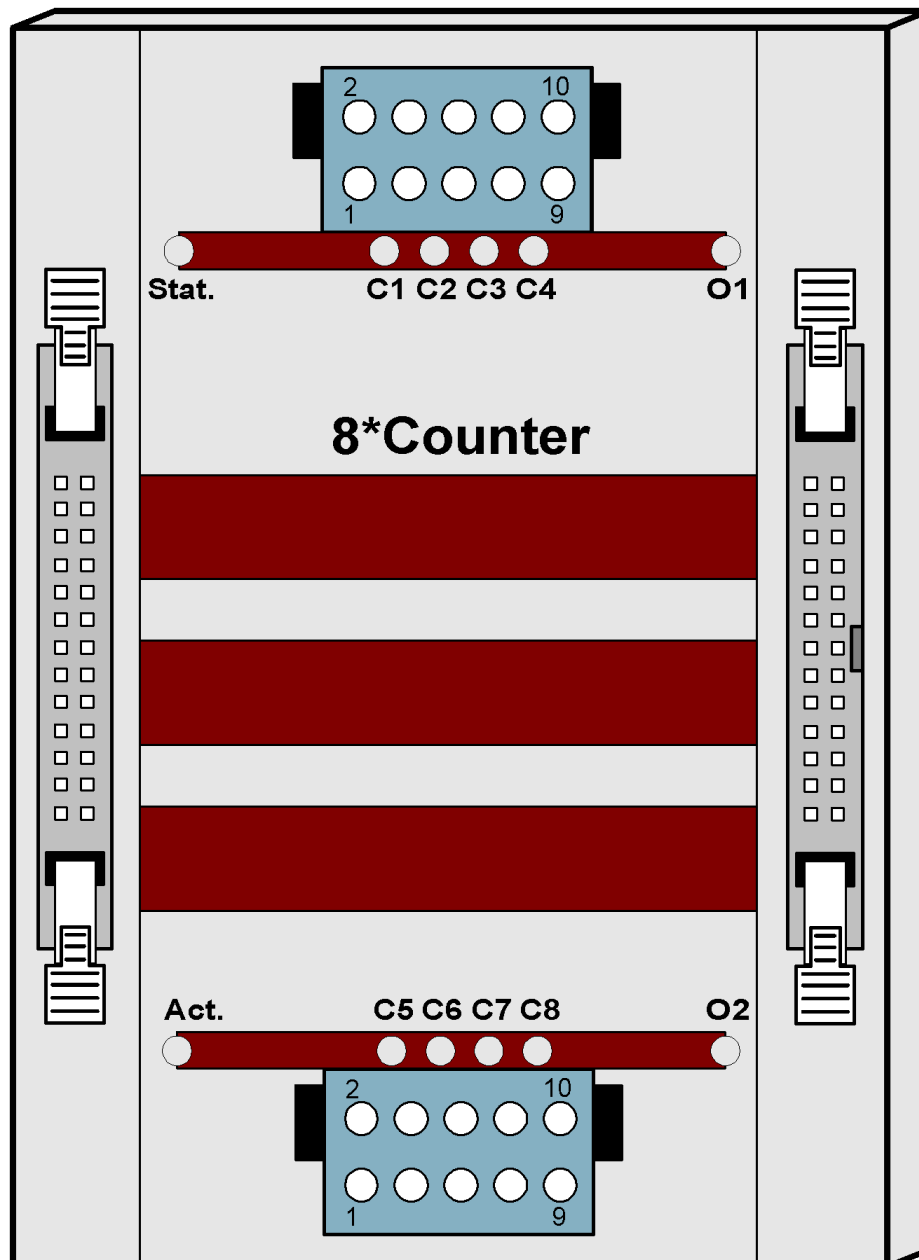
3.2.11. RO-CNT8

3.2.11.1. Übersichtsbild

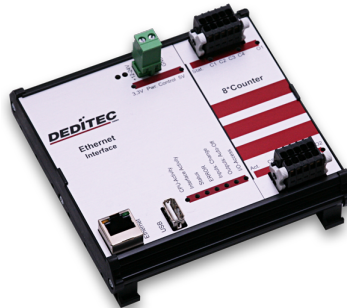
Die Abbildung zeigt ein Modul mit entsprechender Durchnummerierung der Anschlussklemmen.



Die Abbildung zeigt ein Counter-8 Modul des flexiblen Steckverbindersystems mit entsprechender Durchnummerierung der Anschlussklemmen.



3.2.11.2. Technische Daten



- 8 Eingangszähler
- max 20 MHz Eingangsfrequenz
- 32-Bit Zähler
- 15V-30V galv. getrennte Eingänge / Umkonfigurierbar auf 5V-12V
- Jeder Eingangskanal kann mit einem Filter versehen werden
- Jeder Eingangskanal zeigt mit einer LED an, dass ein Eingangszustand erfasst wurde
- 2 Ausgangszähler
- Max 5 MHz Ausgangsfrequenz
- 32-Bit Zähler
- Es kann die Anzahl der zu erzeugenden Impulse eingestellt werden (0 = unendlich viele)
- Einstellung der low-Zeit (10ns Schritte)
- Einstellung der high-Zeit (10ns Schritte)
- PWM Ausgänge mit z.B. folgenden Modi:
  - 1kHz - Frequenz PWM Verhältnis mit 0,01% Auflösung
  - 10kHz - Frequenz PWM Verhältnis mit 0,1% Auflösung
  - 100kHz - Frequenz PWM Verhältnis mit 1% Auflösung
- Anzeige-LEDs Activity LED (kurzes Aufblinken bei Schnittstellenkommunikation)
- Status LED (kurzes Aufblinken bei Zugriff auf D/A Wandler)
- Steckverbinder 10 pol. steckbare Klemmleisten für jeweils einen D/A-Ausgang

3.2.11.3. Counter Modul Modi

Das Counter Modul besitzt folgende 5 Modi:

Modus 0: Zählen von Impulsen Mit diesem Modus lassen sich Impulse anhand von einer steigenden Flanke zählen. Ein zurücksetzen ist hierbei per Software möglich.

Modus 1: Zählen von Impulsen mit "reset bei read"

Mit diesem Modus lassen sich Impulse anhand einer steigenden Flanke zählen. Beim Auslesen des Zählers wird der Zähler gleichzeitig auf Null zurückgesetzt.

Modus 2: Periodendauer Bestimmung

In diesem Modus wird die Periodendauer "T" des angelegten Eingangssignals gemessen. Ein interner 100MHz Zähler dient hierbei als Zeitbasis. Bei einer steigenden Flanke des Eingangssignals wird der 32 Bit Zähler in einen Zwischenspeicher gelegt (Latch) und kann dann ausgewertet werden, während das nächste Signal wieder gezählt wird.

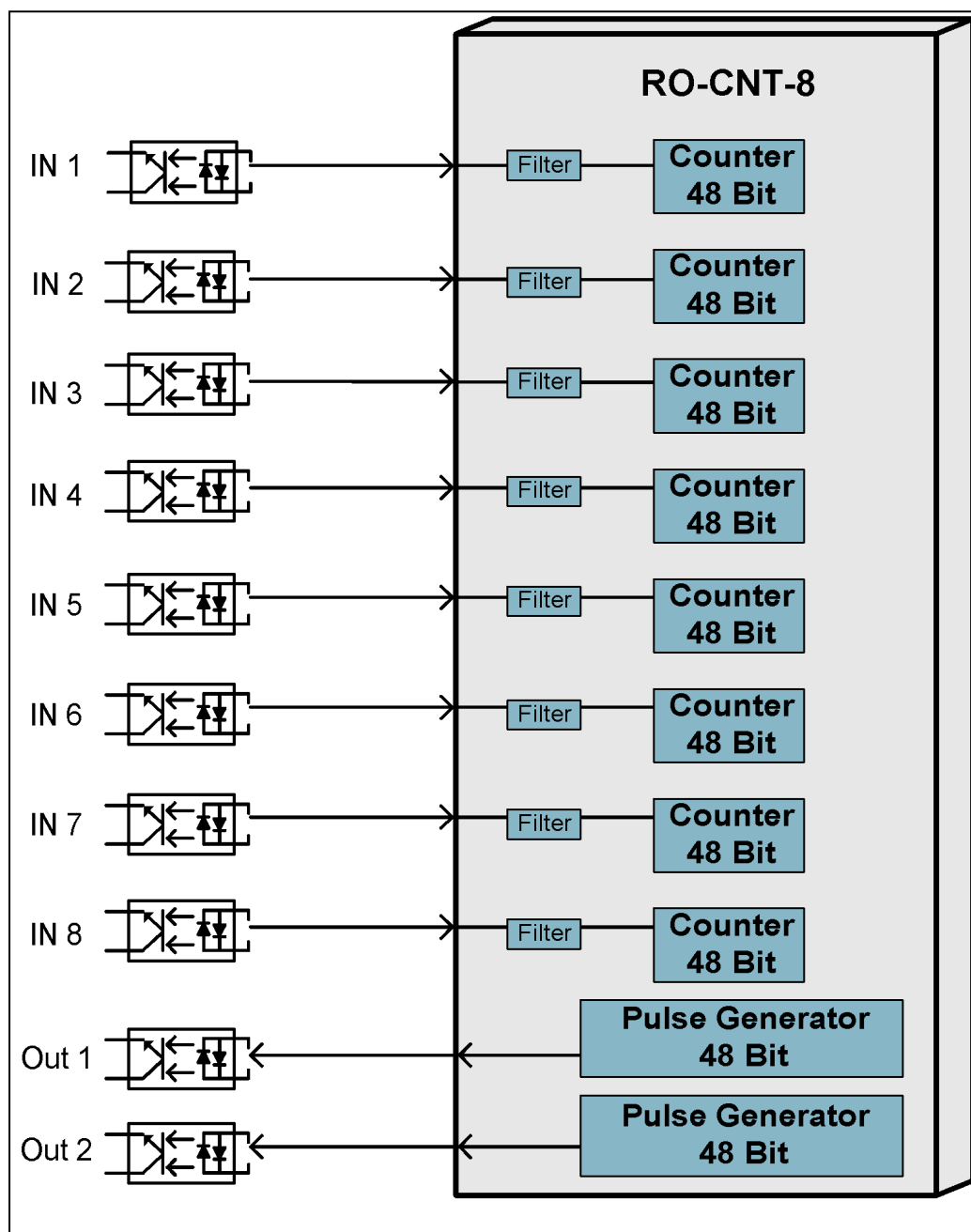
Modus 3: Anzahlmessung/Frequenzmessung

Mit diesem Modus lässt sich die Anzahl der steigenden Flanken des Eingangssignals innerhalb einer bestimmten Zeit messen. Hier lassen sich die Zeiten 1ms, 10ms, 100ms, 1sec. per Software konfigurieren. So lassen sich z.B. Frequenzmessungen durchführen.

Modus 4: PWM Messungen

Mit diesem Modus werden mit zwei unterschiedlichen Zählern die "high"-Zeit und die "low"-Zeit eines Signales gemessen. Dadurch lässt sich das high / low-Verhältnis bestimmen (PWM). Durch die Verwendung eines internen 100MHz Zählers lassen sich hier äusserst genaue PWM Verhältnisse bestimmen.

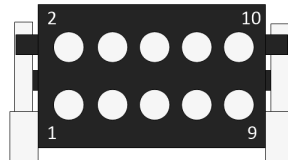
3.2.11.4. Blockschaltbild



3.2.11.5. Pinbelegung

3.2.11.5.1. Pinbelegung RO-CNT8

Steckverbinder 10 pol. steckbare Klemmleiste für jeweils 4 * Counter-Eingänge

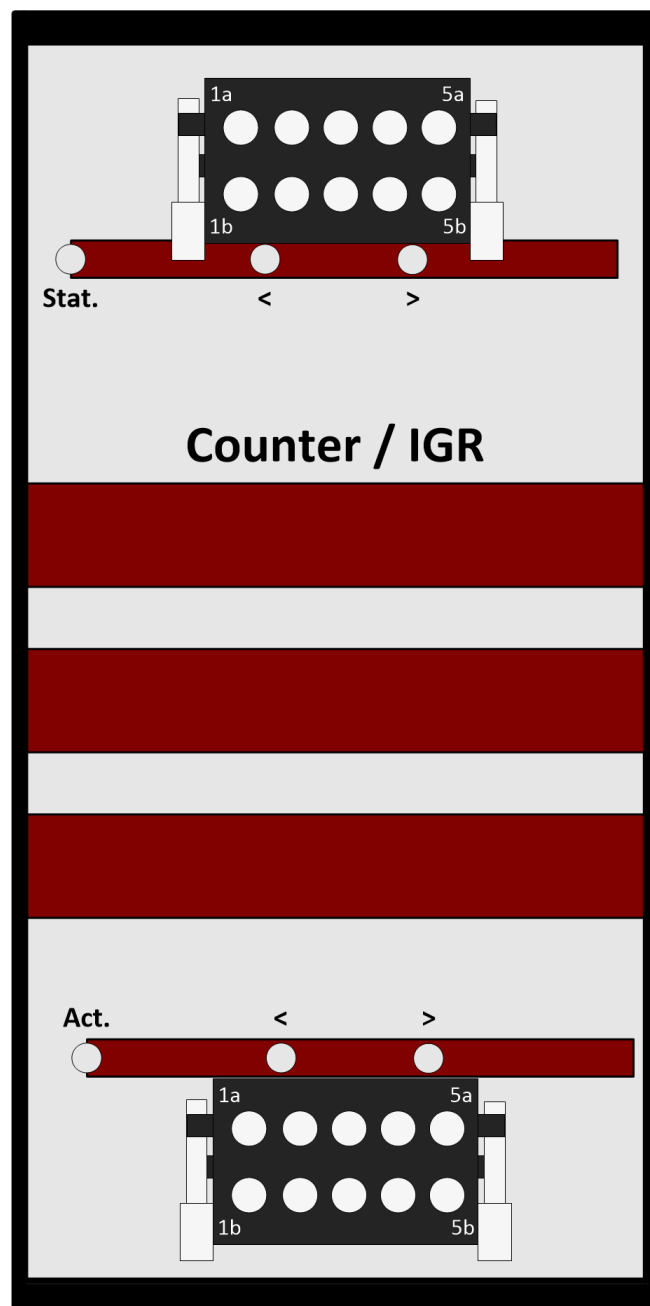


Pin	Belegung
1	Inputchannel 1-
2	Inputchannel 1+
3	Inputchannel 2-
4	Inputchannel 2+
5	Inputchannel 3-
6	Inputchannel 3+
7	Inputchannel 4-
8	Inputchannel 4+
9	GND
10	CNT-OUT +5V

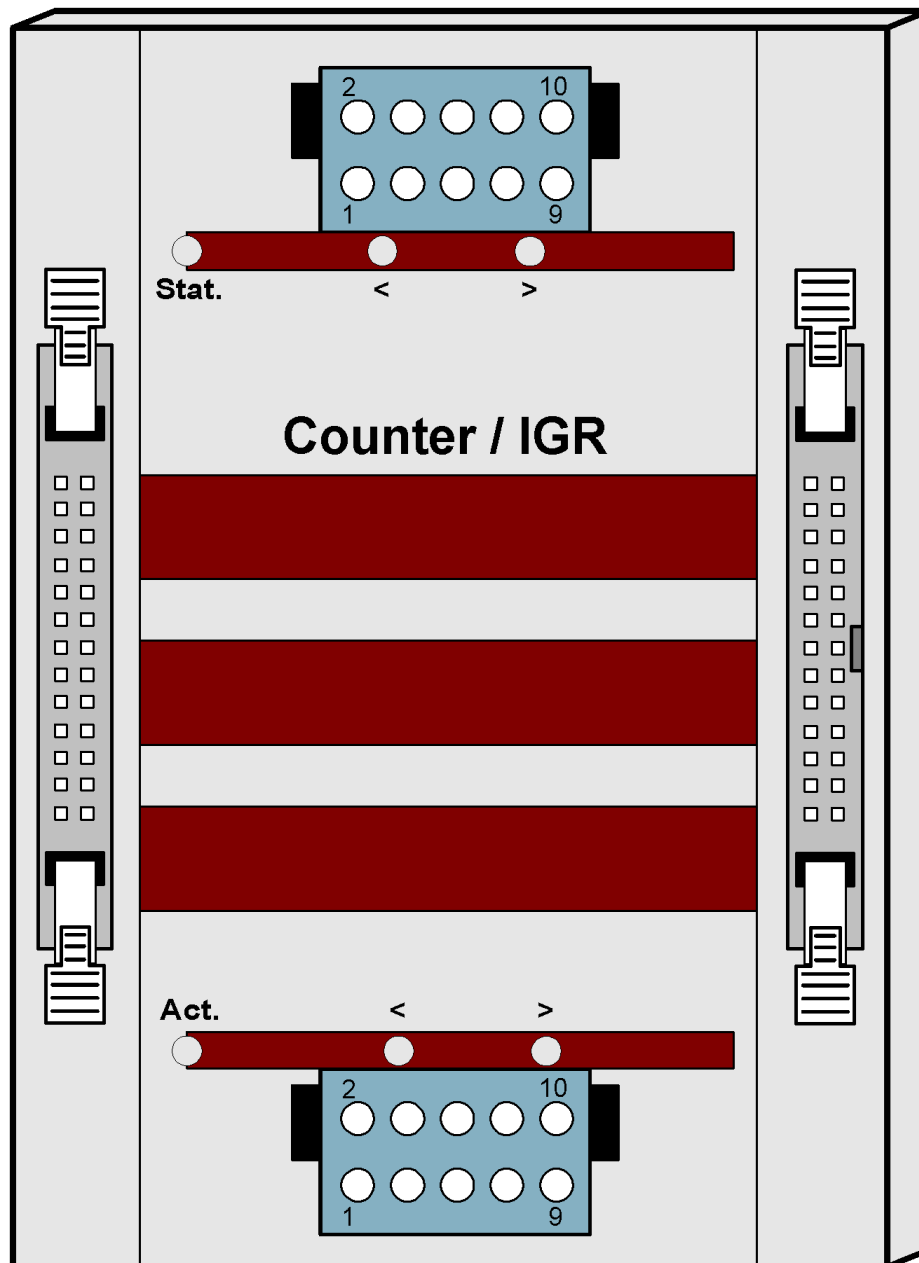
3.2.12. RO-CNT/IGR

3.2.12.1. Übersichtsbild

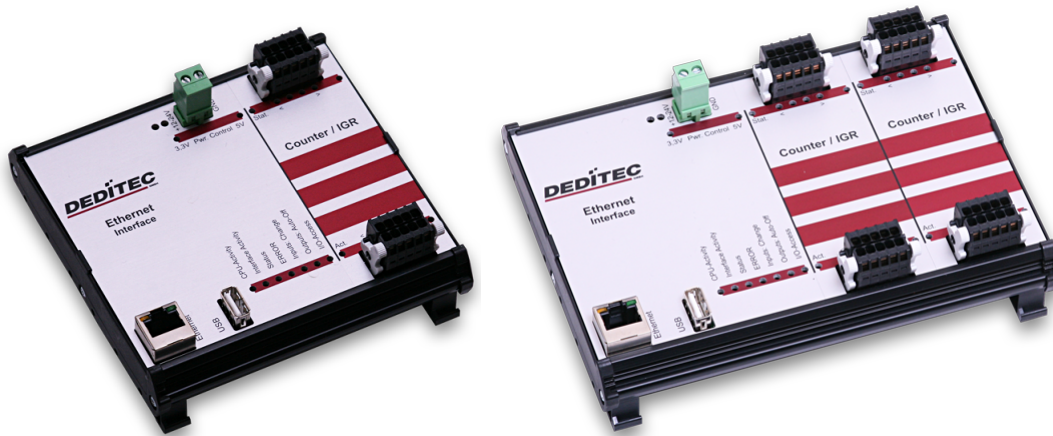
Die Abbildung zeigt ein Modul mit entsprechender Durchnummerierung der Anschlussklemmen.



Die Abbildung zeigt ein Counter/IGR Modul des flexiblen Steckverbindersystems mit entsprechender Durchnummerierung der Anschlussklemmen.

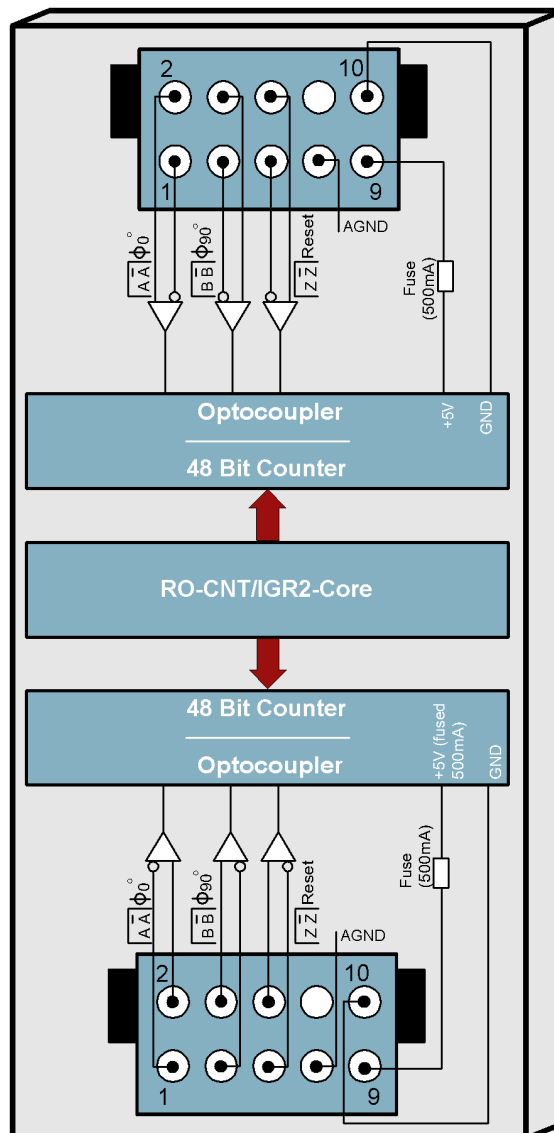


3.2.12.2. Technische Daten

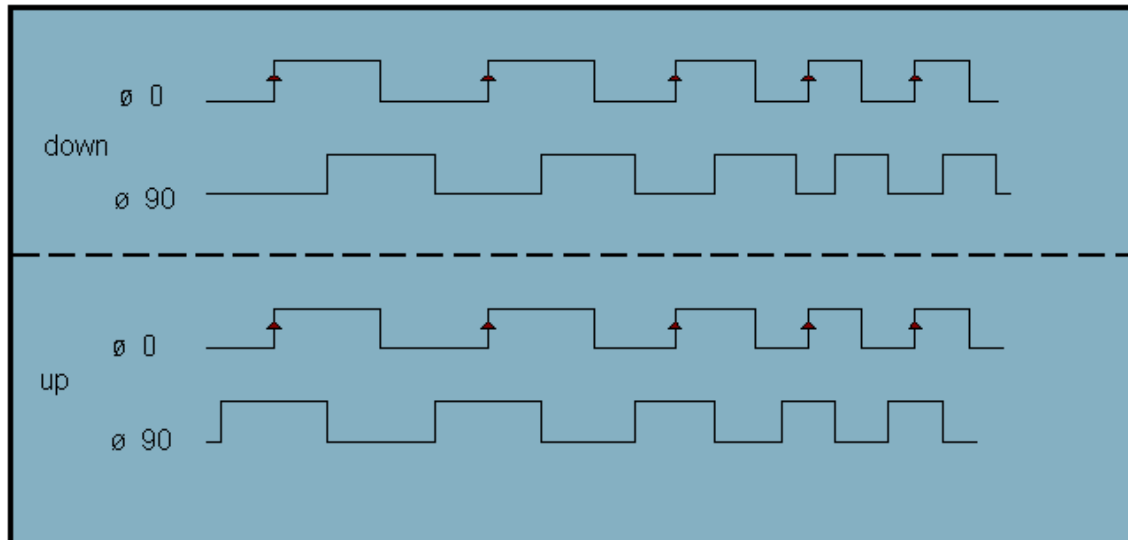


- 48Bit Counter
- Max 10MHz Phi 0/Phi 90° Frequenz
- Galvanisch getrennt (A, B, Z der Eingänge beider Inkrementalgeber Eingänge untereinander und auch zur 24V Modulversorgung)
- Anzeige-LEDs Activity LED (kurzes Aufblinken bei Schnittstellenkommunikation)
- Status LED (kurzes Aufblinken bei Zugriff auf die Inkrementalgeber)
- Richtungs LED's (Leuchten je nach ausgewählter Drehrichtung des Inkrementaldrehgebers)
- Steckverbinder 10 pol. steckbare Klemmleisten für jeweils einen Inkrementalgeber
- Betriebstemperatur 10°C...+50°C

In dem Blockschaltbild wird deutlich, wie das Modul aufgebaut ist.



3.2.12.4. Auswertung Phi 0/Phi 90° (Grafisch)



Diese Grafik verdeutlicht wann das Zählmodul hoch/runter zählt.

Ist bei $\Phi 90^\circ$ der Pegel low so wird runter gezählt. Ist der Pegel high so wird hoch gezählt.

3.2.12.5. LEDs

Activity LED:

- kurzes Aufblinken bei Schnittstellenkommunikation

Status LED:

- kurzes Aufblinken bei Zugriff vom Interface auf die Inkrementalgeber

Richtungs LEDs:

- linke Richtungs LED blinkt kurz auf wenn das Zählermodul runter zählt.
- rechte Richtungs LED blinkt kurz auf wenn das Zählermodul hoch zählt.
- beide Richtungs LEDs blinken kurz auf wenn der Zählerstand zurückgesetzt wird.

Counter LED:

- blinkt auf wenn ein Zählvorgang statt findet.

3.2.12.6. Eingänge

3.2.12.6.1. Referenzieren

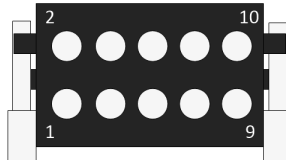
Durch das Neustarten des Zählermoduls wird die absolute Position nicht erkannt und die aktuelle Position auf 0 gesetzt. Dabei können Fehlmessungen entstehen. Durch Endlagenschaltern können Referenzfahrten eingestellt werden bei denen eine bestimmte Position angefahren wird. Dies führt zur genauen Kalibrierung des Drehgebers. Inkrementelle Drehgeber geben pro voller Umdrehung noch einen Referenzimpuls (Index) auf einem dritten Kanal aus. Einige Inkrementalgeber haben auch eine abstandskodierte Referenzmarke.

3.2.12.6.2. Visuelle Kontrolle der Eingänge

Über LED wird der Zustand als auch die Drehrichtung jedes Inkrementalgebers direkt angezeigt. Fehler in der Verdrahtung sind somit schneller erkennbar.

3.2.12.7. Pinbelegung

Steckverbinder 10 pol. steckbare Klemmleiste für jeweils 1 * Inkrementalgeber-Eingang

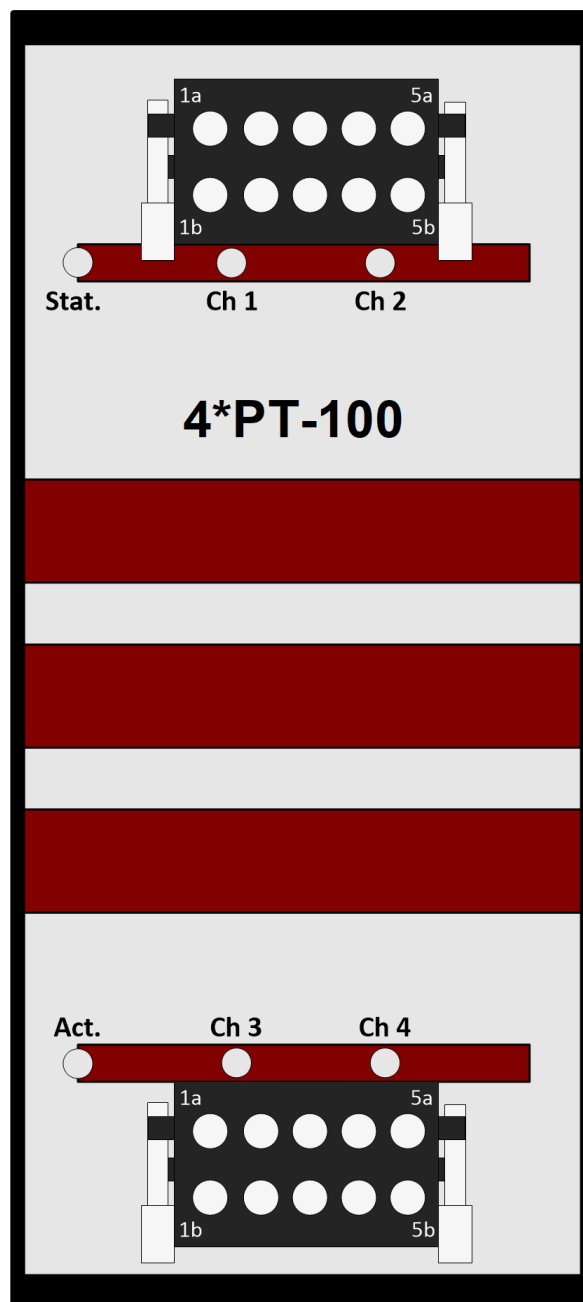


Pin	Belegung	Funktion	Galvanische Trennung
1	Inputchannel A-	Phi 0°	Galvanisch getrennt
2	Inputchannel A+		
3	Inputchannel B-	Phi 90°	
4	Inputchannel B+		
5	Inputchannel Z-	Reset	
6	Inputchannel Z+		
7	GND-Analog	GND zu Pin 1-6	
8	N.C.		
9	GND	GND zu Pin 10	keine Galvanische Trennung
10	+5V	Fused (500mA)	

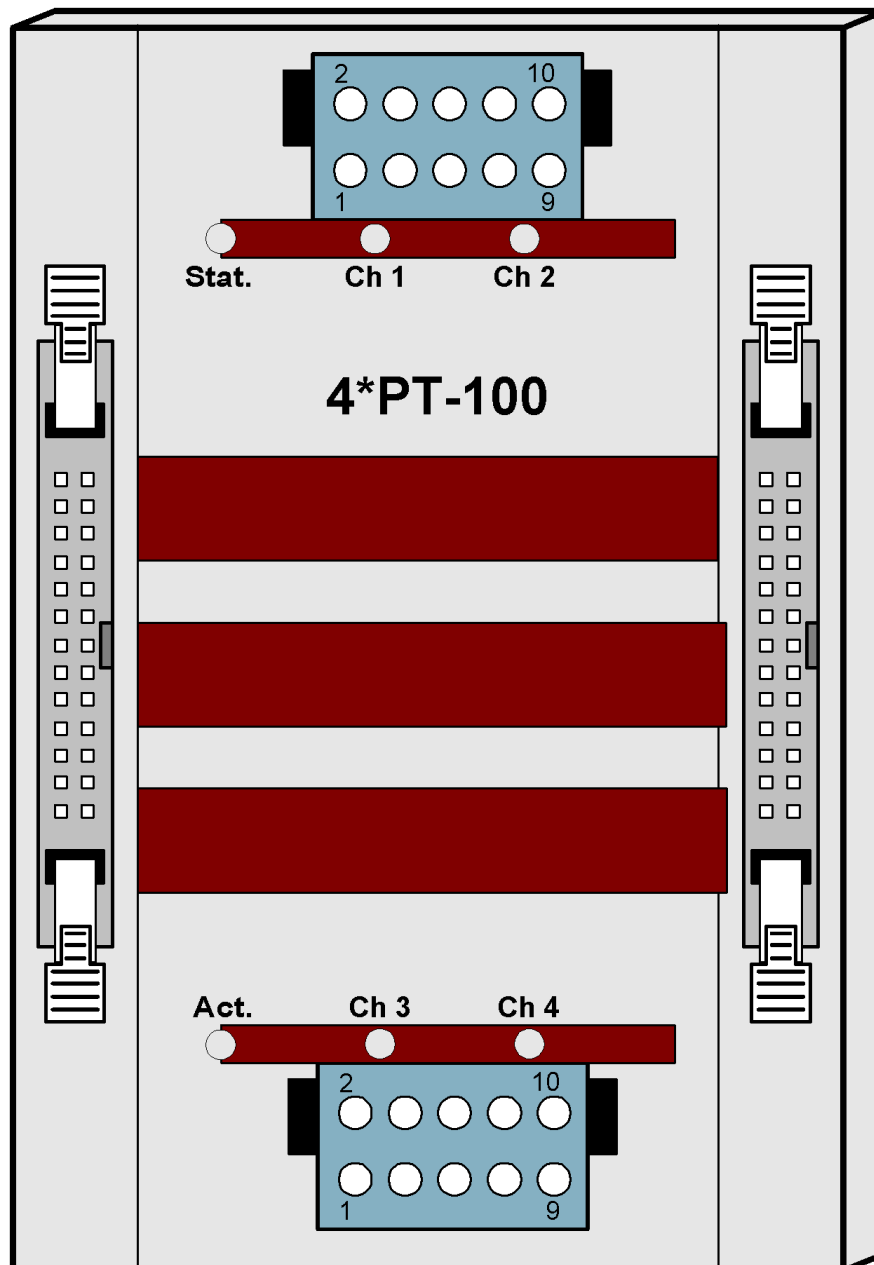
3.2.13. RO-PT100-4 / RO-PT100-8

3.2.13.1. Übersichtsbild

Die Abbildung zeigt ein Modul mit entsprechender Durchnummerierung der Anschlussklemmen.



Die Abbildung zeigt ein Counter-8 Modul des flexiblen Steckverbindersystems mit entsprechender Durchnummerierung der Anschlussklemmen.



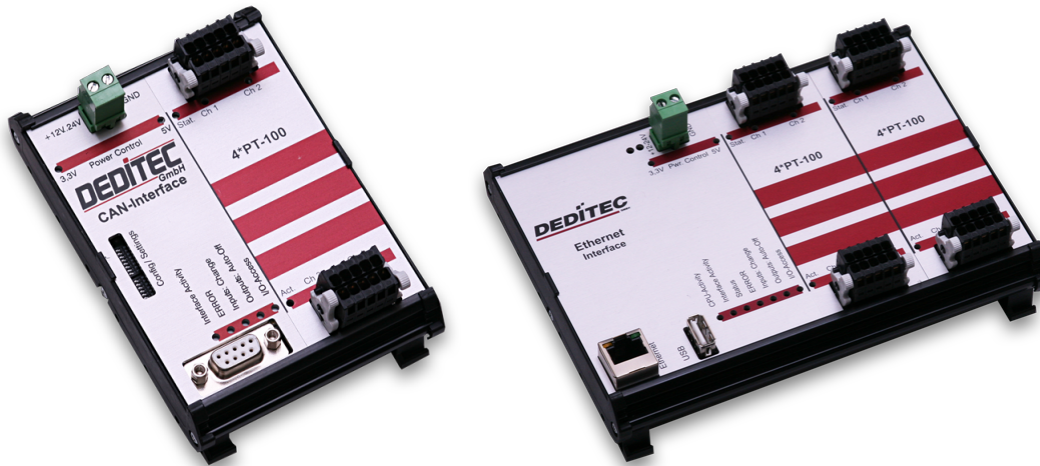
3.2.13.2. Einführung

Über den Zwei- oder Dreileiteranschluss lässt sich ein PT100-Sensor anschließen, mit dem eine sehr genaue Messung durch den integrierten 24 Bit A/D Wandler möglich ist.

Je nach Sensor liegt der Temperaturmessungsbereich zwischen -200°C und 850°C. Der Leitungswiderstand wird durch eine zusätzliche Fühlerleitung ermittelt und ausgeglichen, dadurch ist der Einsatz einer viel längeren Anschlussleitung möglich.

Als Anschlussklemmen kommen servicefreundliche Steckleisten mit Verriegelungsschutz und Auswerfmechanik zum Einsatz. Diese ermöglichen ein schnelles, nachträgliches Umstecken der angeschlossenen Anlagen. Der Leitungsanschluß selbst erfolgt über ein schraubenloses Stecksystem.

3.2.13.3. Technische Daten



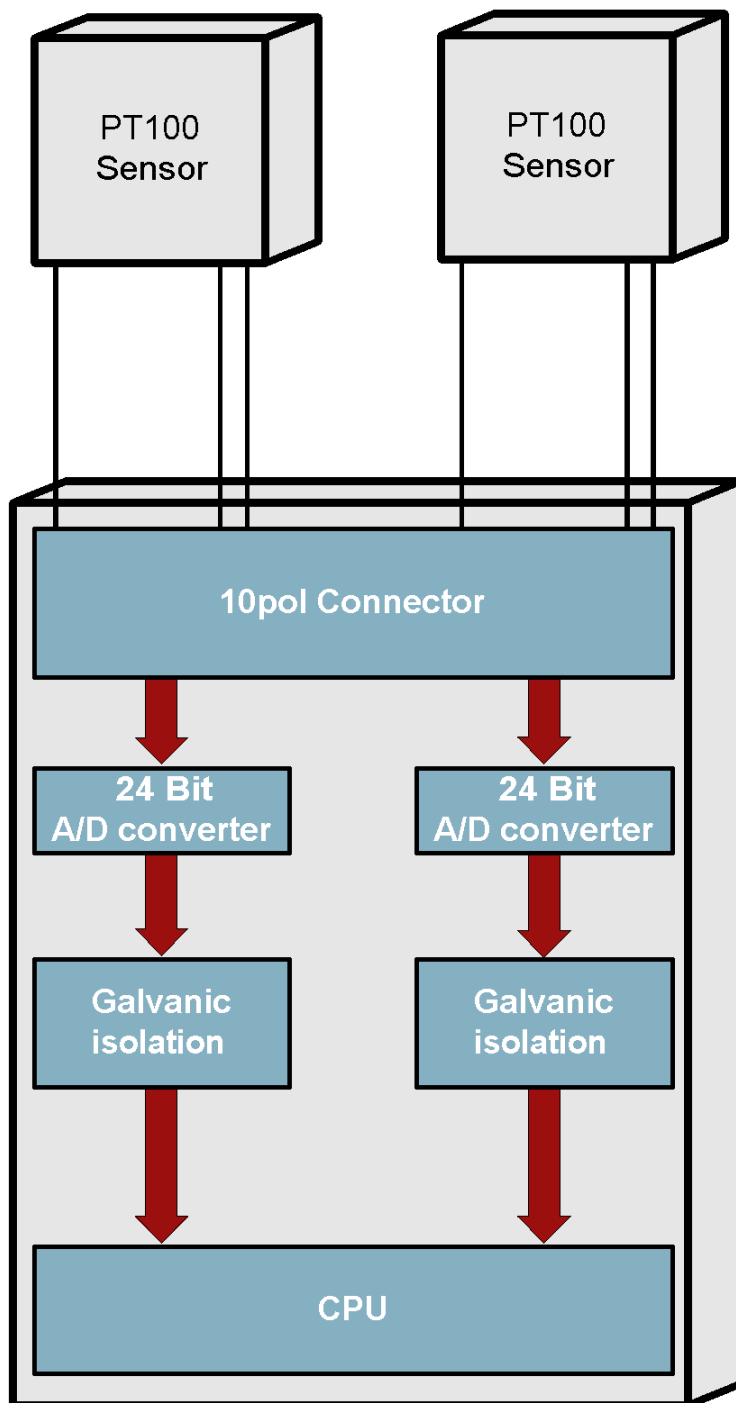
- 24 Bit A/D Wandler für eine genaue Messung
- Der Temperaturmessbereich liegt zwischen -200°C bis 850°C.
- Galvanisch getrennte Eingänge.
- Anzeige-LEDs Activity LED (kurzes Aufblinken bei Schnittstellenkommunikation)
- Status LED (kurzes Aufblinken bei Zugriff auf PT100)
- Anschluss LED 1 je Eingang (kurzes Aufblinken bei einem angeschlossenen PT100 Sensor. Leuchtet bei nicht belegten Eingängen oder bei einem defekten PT100 Sensor).
- Steckverbinder 10 pol. steckbare Klemmleiste für jeweils 2 * PT100-Eingänge
- Kann problemlos mit anderen Modulen der RO Serie kombiniert werden

Produktspezifische Daten:

Produkt	Eingänge
RO-PT100-4	4 * PT100 Eingänge
RO-PT100-8	8 * PT100 Eingänge

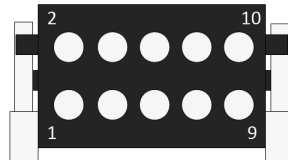
3.2.13.4. Blockschaltbild

In dem Blockschaltbild wird deutlich, wie das Modul aufgebaut ist.



3.2.13.5. Pinbelegung

Steckverbinder 10 pol. steckbare Klemmleiste für jeweils 2 * PT100-Eingänge



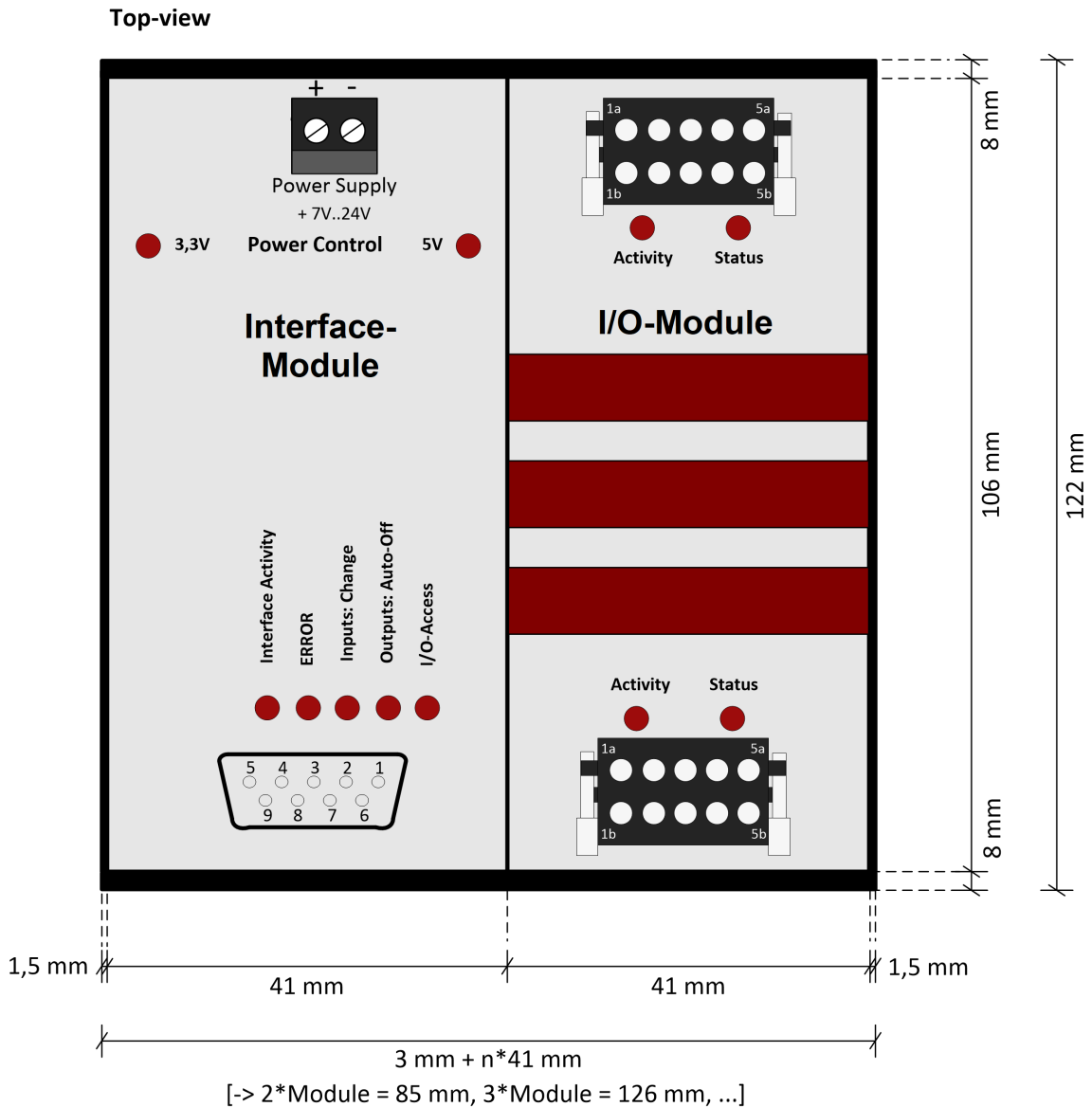
Pin	Belegung
1	AIN-_A
2	AIN+_A
3	AIN-_A
4	REFIN+_A
5	-
6	-
7	AIN-_B
8	AIN+_A
9	AIN-_B
10	REFIN+_B

Zweikanal Konfiguration: Der Temperatursensor wird an Pin 1 (AIN- A) und Pin2 (AIN+ A) angeschlossen. Pin 3 (AIN-A) und Pin 4 (REFIN + A) müssen hierbei überbrückt werden.

Dreikanal Konfiguration: Der Temperatursensor wird an Pin 1 (AIN- A) und Pin2 (AIN+ A) angeschlossen. Pin 4 (REFIN+ A) dient hierbei als Fühlerleitung und muss zwischen Temperatursensor und Pin 1 (AIN- A) angeklemt werden.

3.3. Abmessung der RO-Serie

3.3.1. Draufsicht



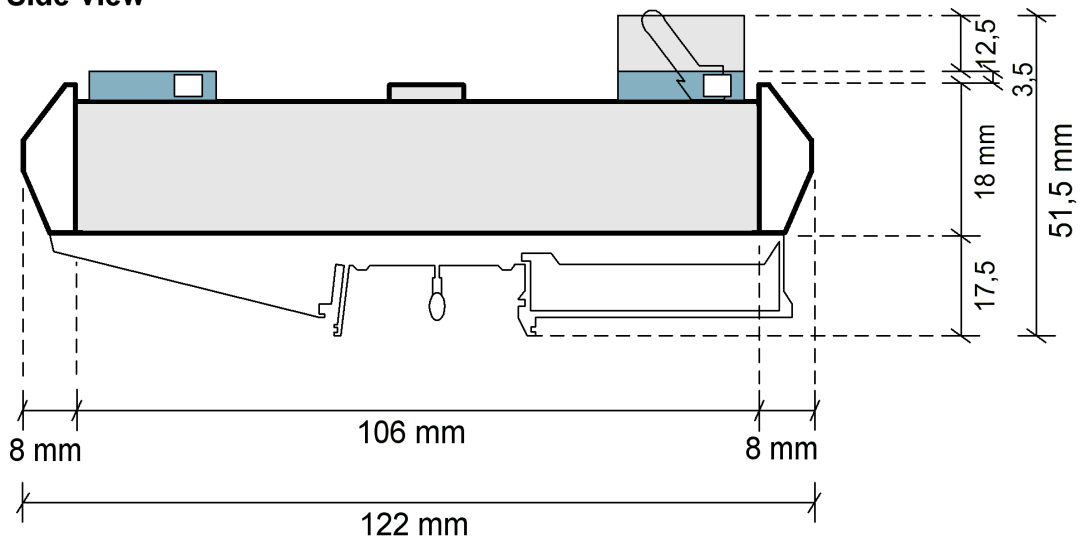
Hinweis:

Die Breite des RO-ETH-Interfaces entspricht der doppelten Breite eines RO-CAN-, RO-SER- oder RO-USB-Interfaces.

So ergibt sich eine Breite von $3 \text{ mm} + 82 \text{ mm} + n \cdot 41 \text{ mm}$

3.3.2. Seitenansicht

Side-view



Software Beschreibung



IV

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 angehängt werden:

```
delib_cli DI8 0 hex nunit
```

Beispiel eines Rückgabewertes: **FF**

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

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

```
delib_cli FF 0
```

Beispiel eines Rückgabewertes: **192**

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

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

```
delib_cli FF 32
```

Beispiel eines Rückgabewertes: **65535**

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

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

```
delib_cli FF 0 hex
```

Beispiel eines Rückgabewertes: **0xD00**

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

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

Analoge Ausgänge

`delib_cli AO 7 4711`

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

`delib_cli AO 6 0x4711`

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

`delib_cli AO 7 3.7V`

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

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

`delib_cli AO 7 13.3mA`

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

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

Analoge Eingänge

```
delib_cli AI 2
```

Beispiel eines Rückgabewertes: **1234**

→ liest den Wert des 3. analogen Eingangs als dezimalzahl

```
delib_cli AI 2 hex
```

Beispiel eines Rückgabewertes: **0x1FA**

→ liest den Wert des 3. analogen Eingangs als hexadezimalzahl

```
delib_cli AI 2 V
```

Beispiel eines Rückgabewertes: **12.500000V**

→ liest die Spannung des 3. analogen Eingangs als kommazahl

```
delib_cli AI 2 mA
```

Beispiel eines Rückgabewertes: **20.551600mA**

→ liest den Strom des 3. analogen Eingangs als kommazahl

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

```
delib_cli AI 3 hex nunit
```

Beispiel eines Rückgabewertes: **1FA**

→ liest den Wert des 4. analogen Eingangs als hexadezimalzahl

```
delib_cli AI 3 V nunit
```

Beispiel eines Rückgabewertes: **12.500000**

→ liest die Spannung des 4. analogen Eingangs als kommazahl

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++

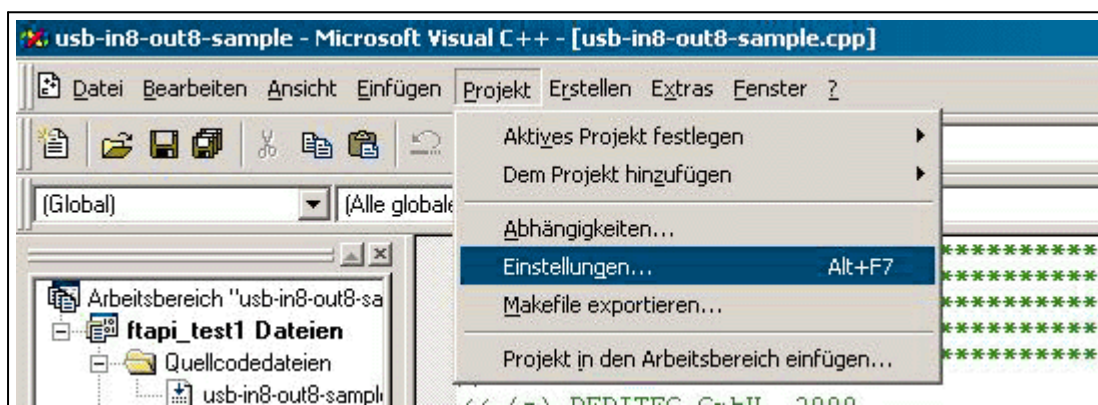
Zur Erleichterung für Verweise auf das DELIB-Include und das DELIB-Lib Verzeichnis werden bei installation der DELIB Umgebungsvariablen definiert.

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

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

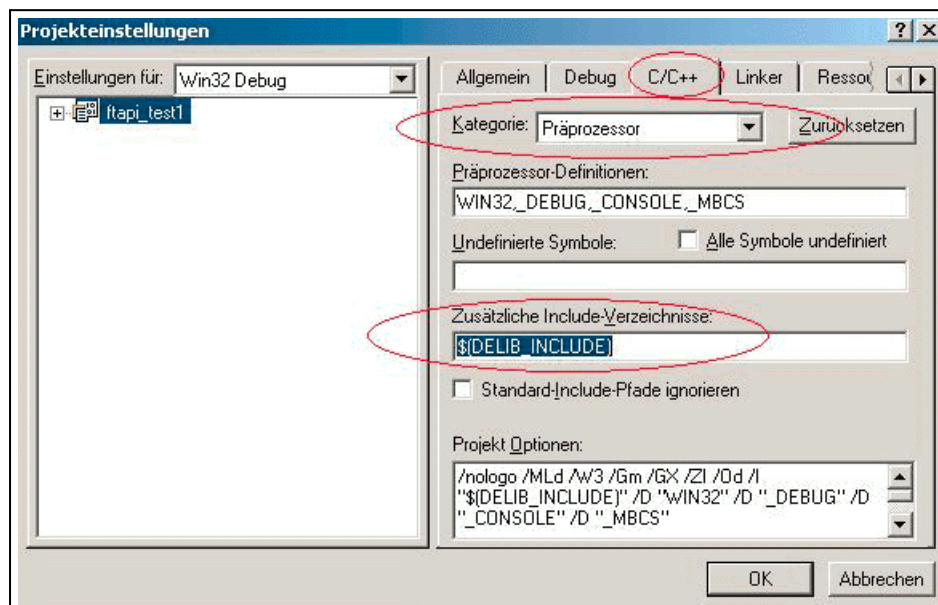
Diese werden im Folgenden in den Projekteinstellungen des Compilers eingetragen.

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



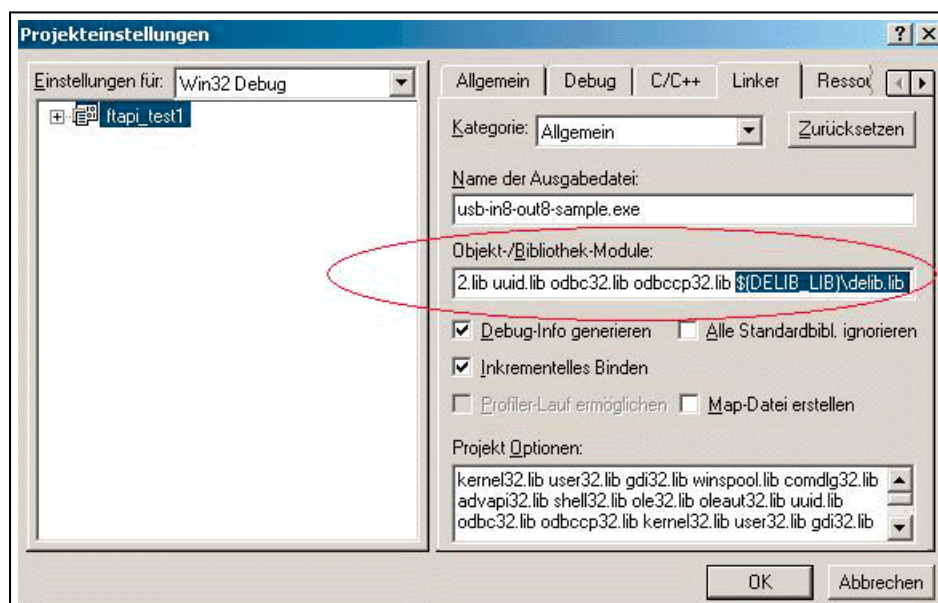
DELIB.H Eintrag in den Visual-C/C++ Projekt Einstellungen

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

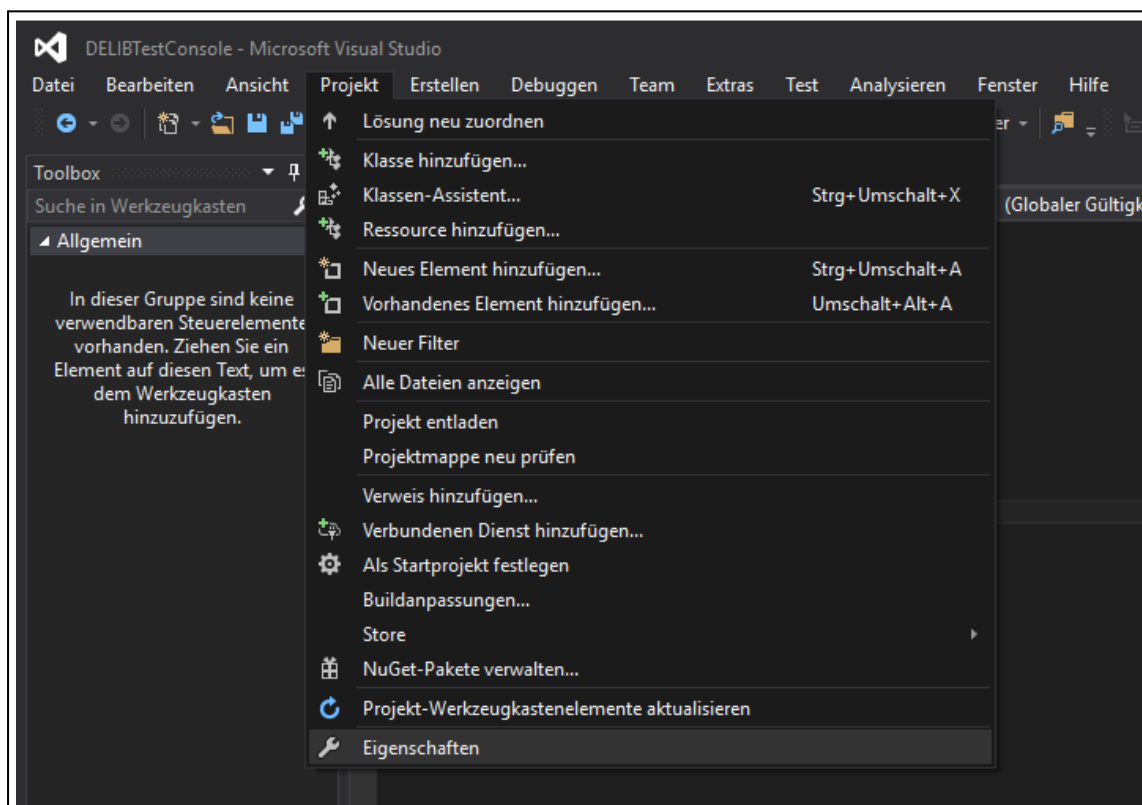
64 Bit DELIB Installation

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

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

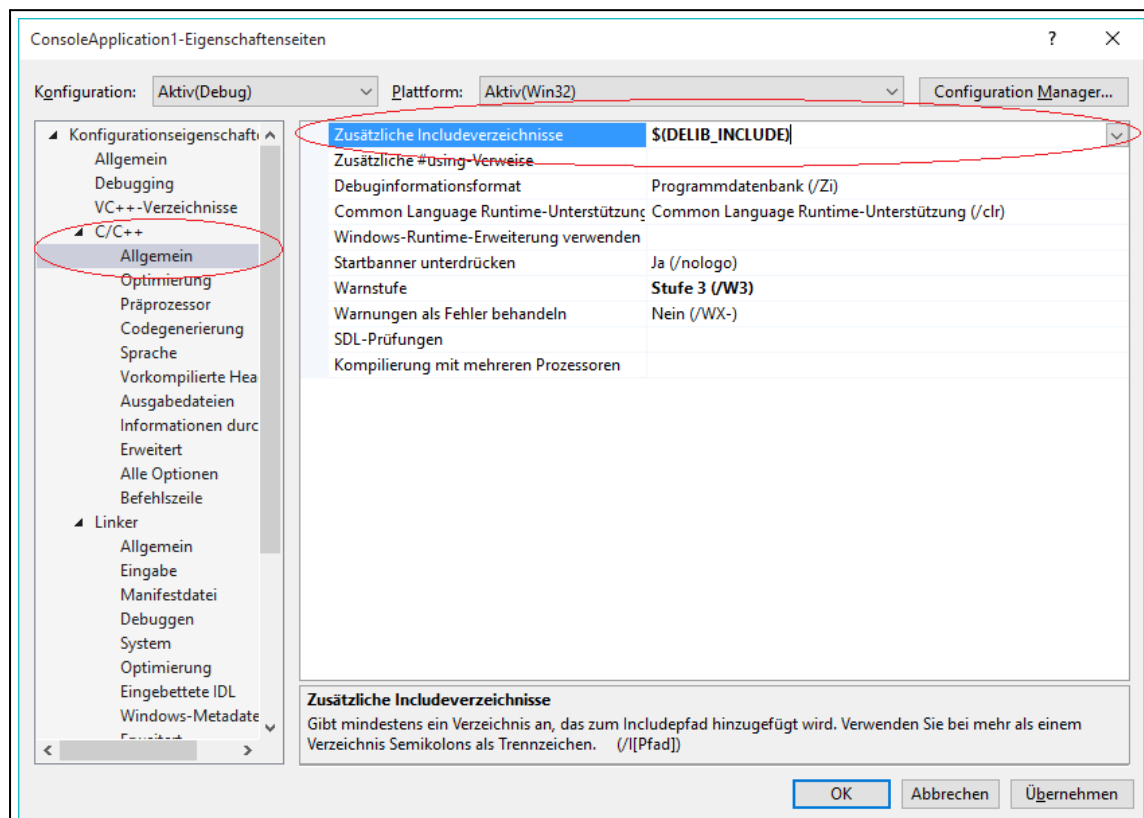
Diese werden im Folgenden in den Projekteinstellungen des Compilers eingetragen.

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



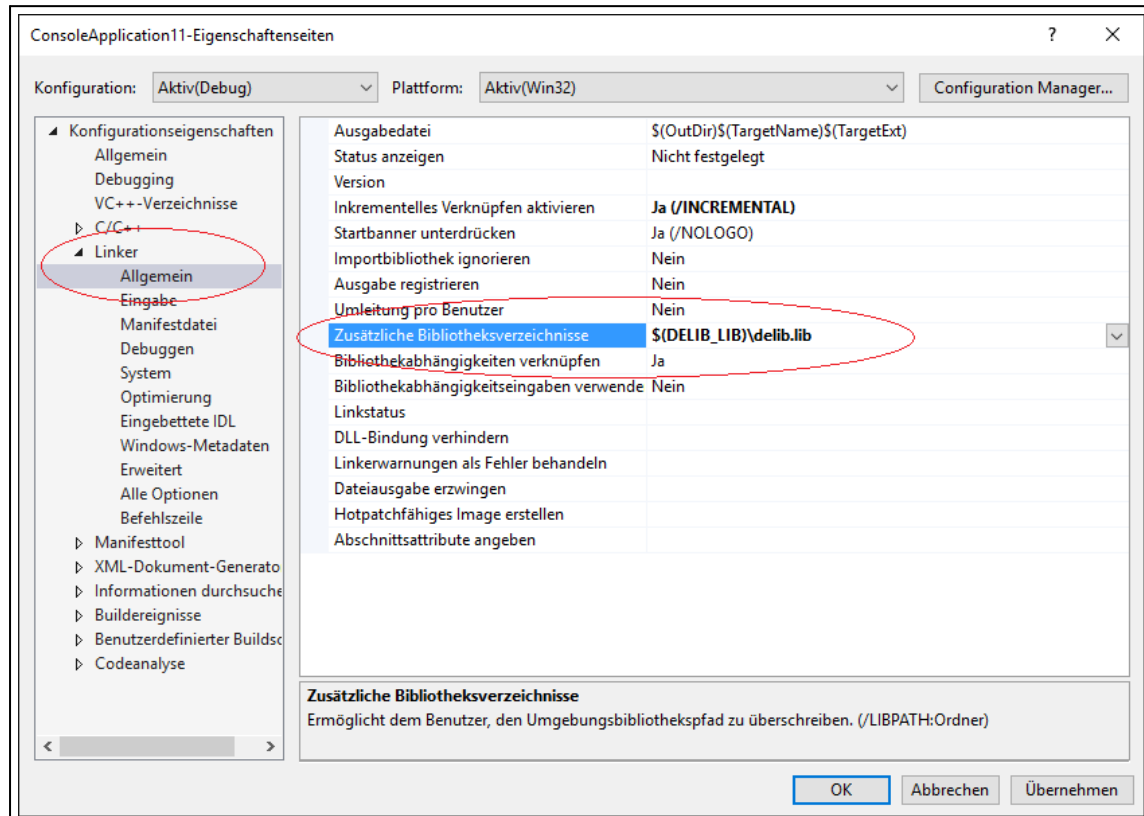
DELIB.H Eintrag in den Visual-C/C++ Projekt Einstellungen

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

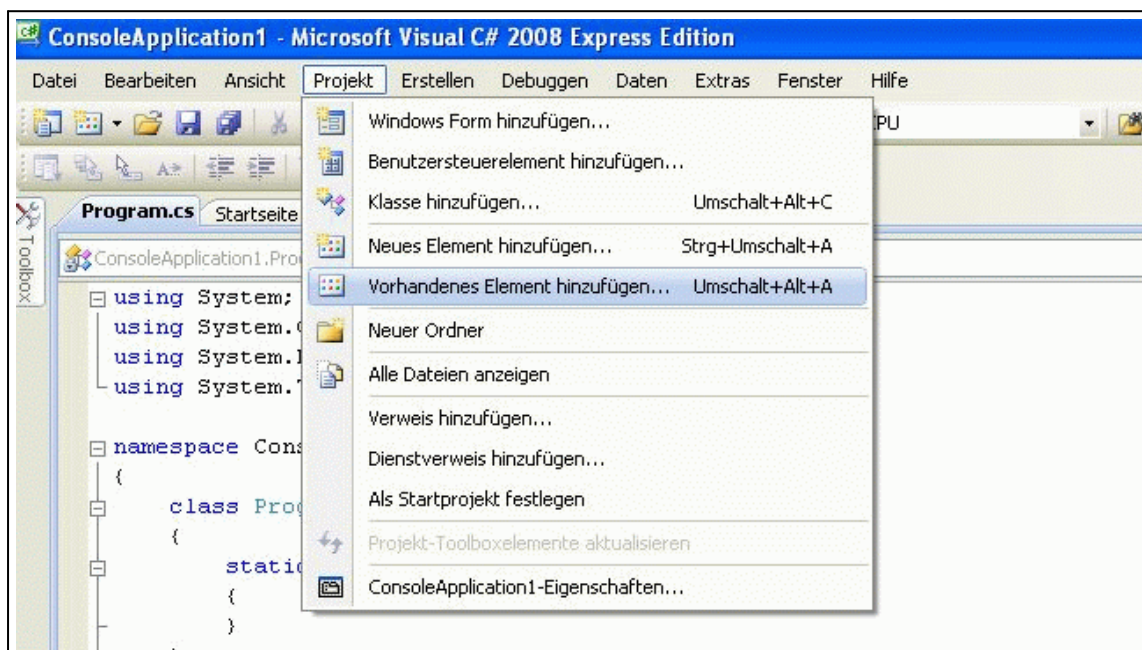
Unter dem Reiter "Linker" bei "Allgemein" "\$ (DELIB_LIB)\delib.lib" eintragen.



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 Menue "Projekt → Vorhandenes Element hinzufügen" im Verzeichnis C:\Programme\DEDITEC\DELIB\include\ die Datei delib.cs zum Importieren öffnen.



Folgenden Verweis in Ihrem Programm hinzufügen:

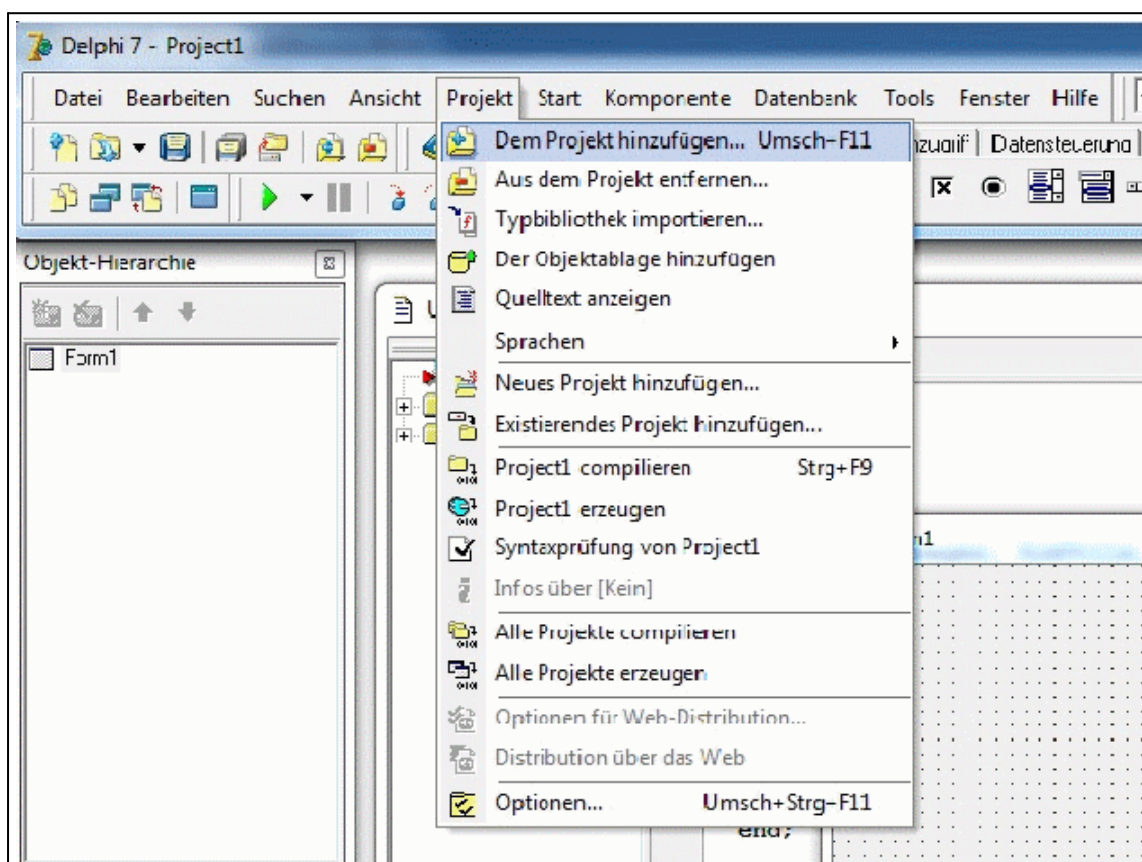
using DeLib;

4.1.6.4. Einbinden der DELIB in Delphi

Die benötigte Datei für Delphi befindet sich im Verzeichnis

C:\Programme\DEDITEC\DELIB\include.

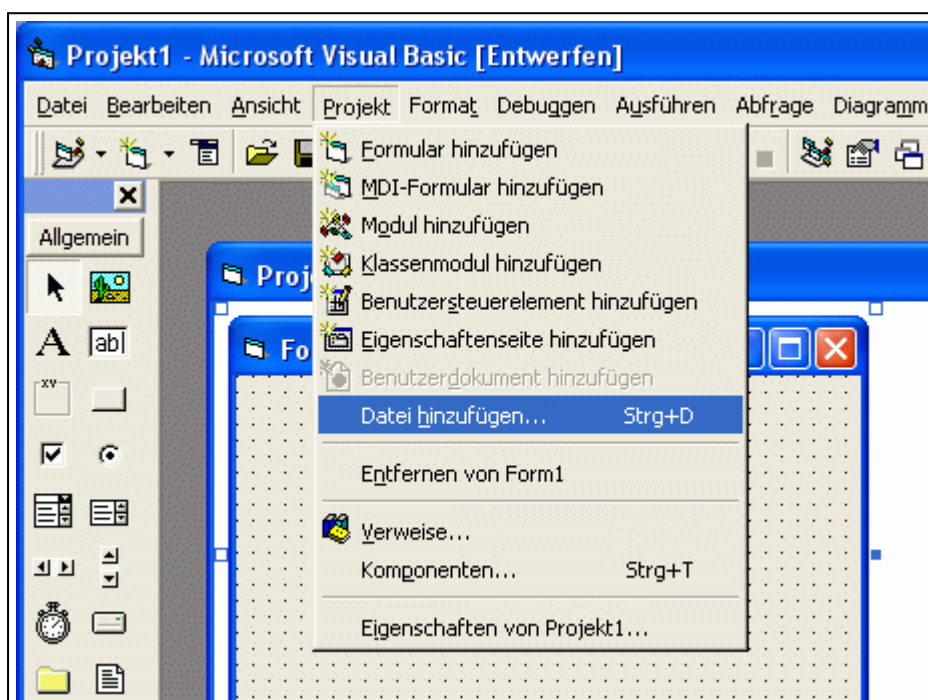
Delphi starten und über das Menue "Projekt → dem Projekt hinzufügen" im Verzeichnis C:\Programme\DEDITEC\DELIB\include\ die Datei delib.pas zum Importieren öffnen.



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

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

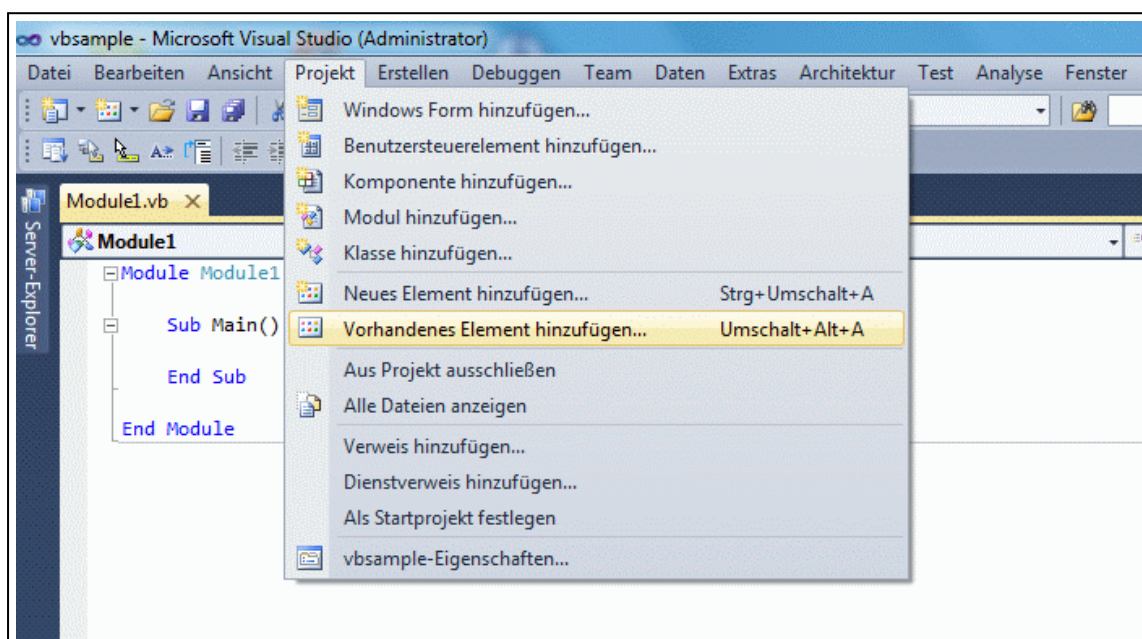
Visual Basic starten und über das Menue "Projekt → Datei hinzufügen..." im Verzeichnis C:\Programme\DEDITEC\DELIB\include\ die Datei delib.bas zum Importieren öffnen.



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

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

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

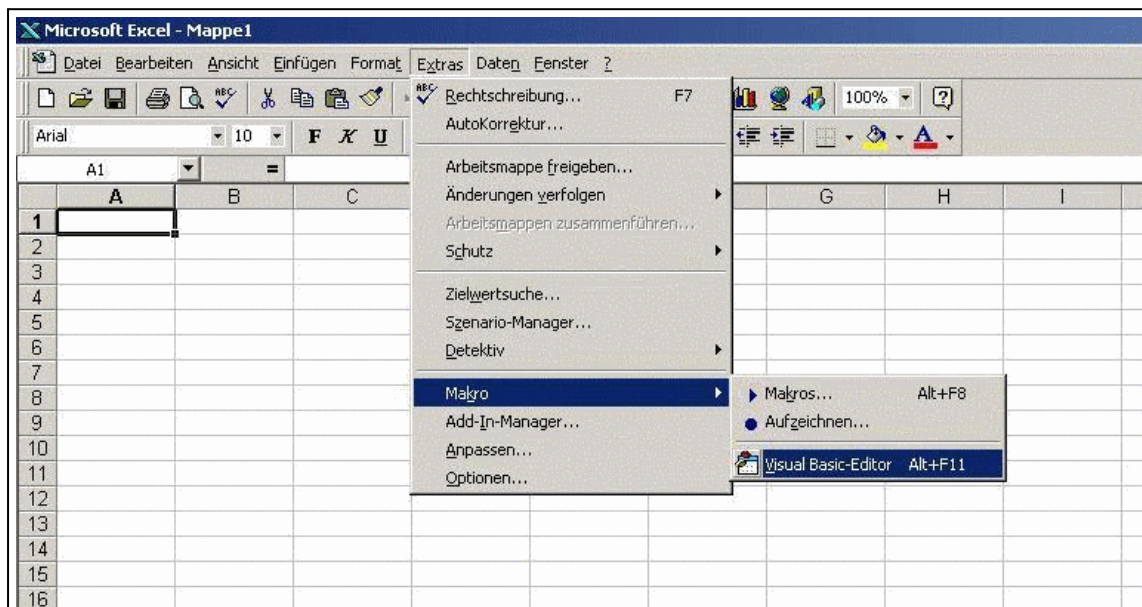


4.1.6.7. Einbinden der DELIB in MS-Office (VBA)

Die benötigte Datei für VBA befindet sich im Verzeichnis

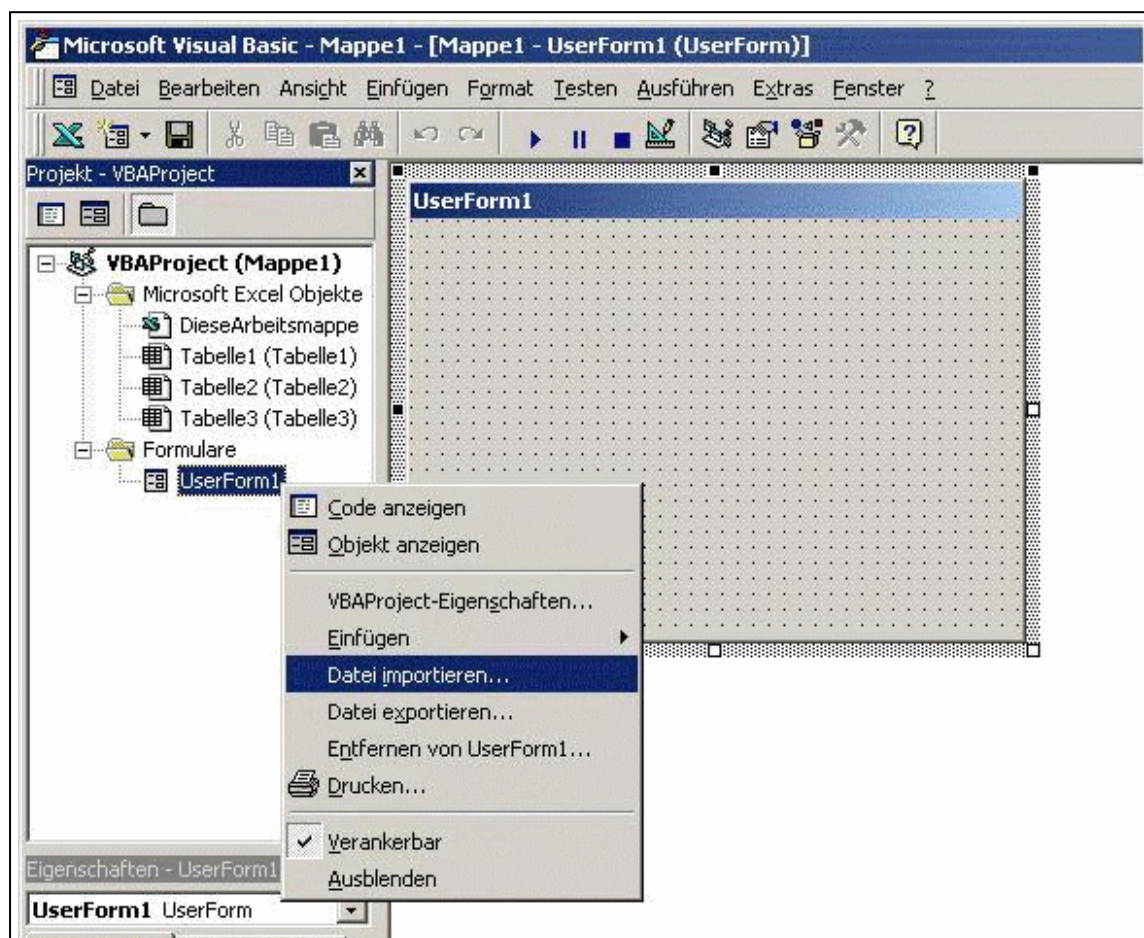
C:\Programme\DEDITEC\DELIB\include.

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



Erstellen der UserForm

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



4.1.6.8. Einbinden der DELIB in LabVIEW

4.1.6.8.1. Einbinden der DELIB in LabVIEW

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

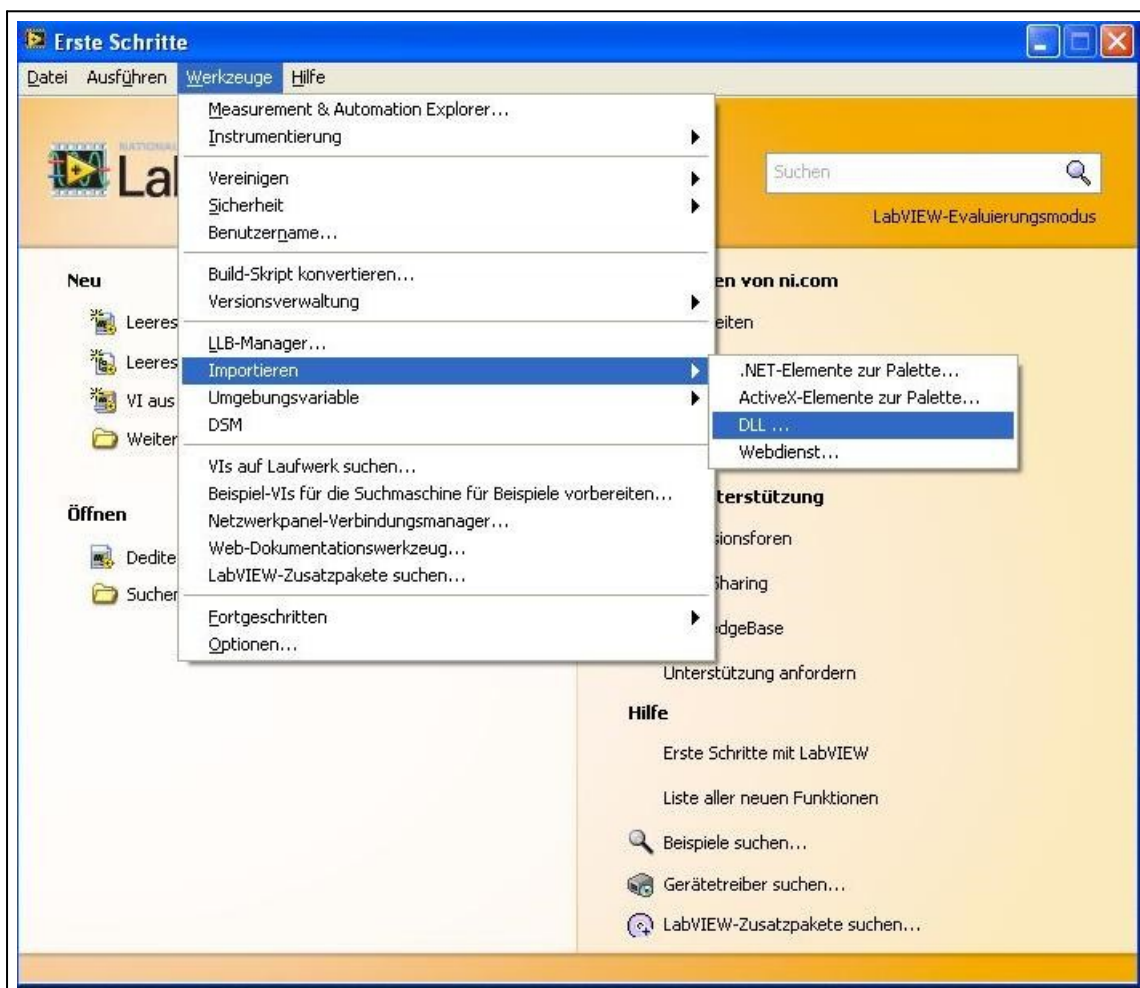
Beschreibung der Einbindung der "delib.dll" in LabVIEW Version 11

- Die benötigten Dateien für LabVIEW befinden sich im Verzeichnis

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

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

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



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

[illegible]

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

DLL importieren

DLL- und Header-Datei wählen

NATIONAL INSTRUMENTS

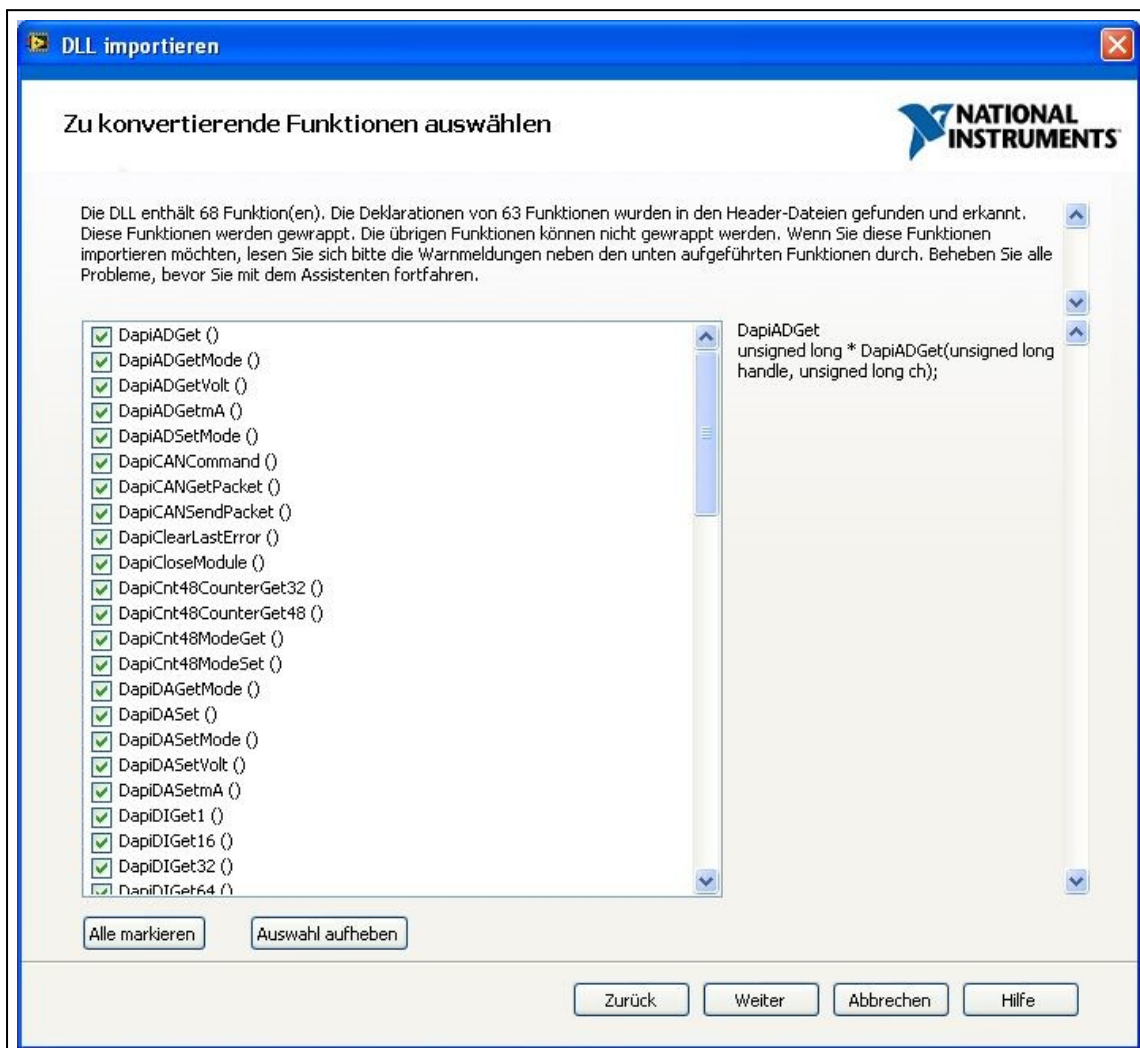
Bibliothek (*.dll)
C:\WINDOWS\system32\delib.dll

☐ DLL befindet sich nicht auf lokalem Computer

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

Zurück Weiter Abbrechen Hilfe

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



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

DLL importieren

Einstellungen der Projektbibliothek konfigurieren

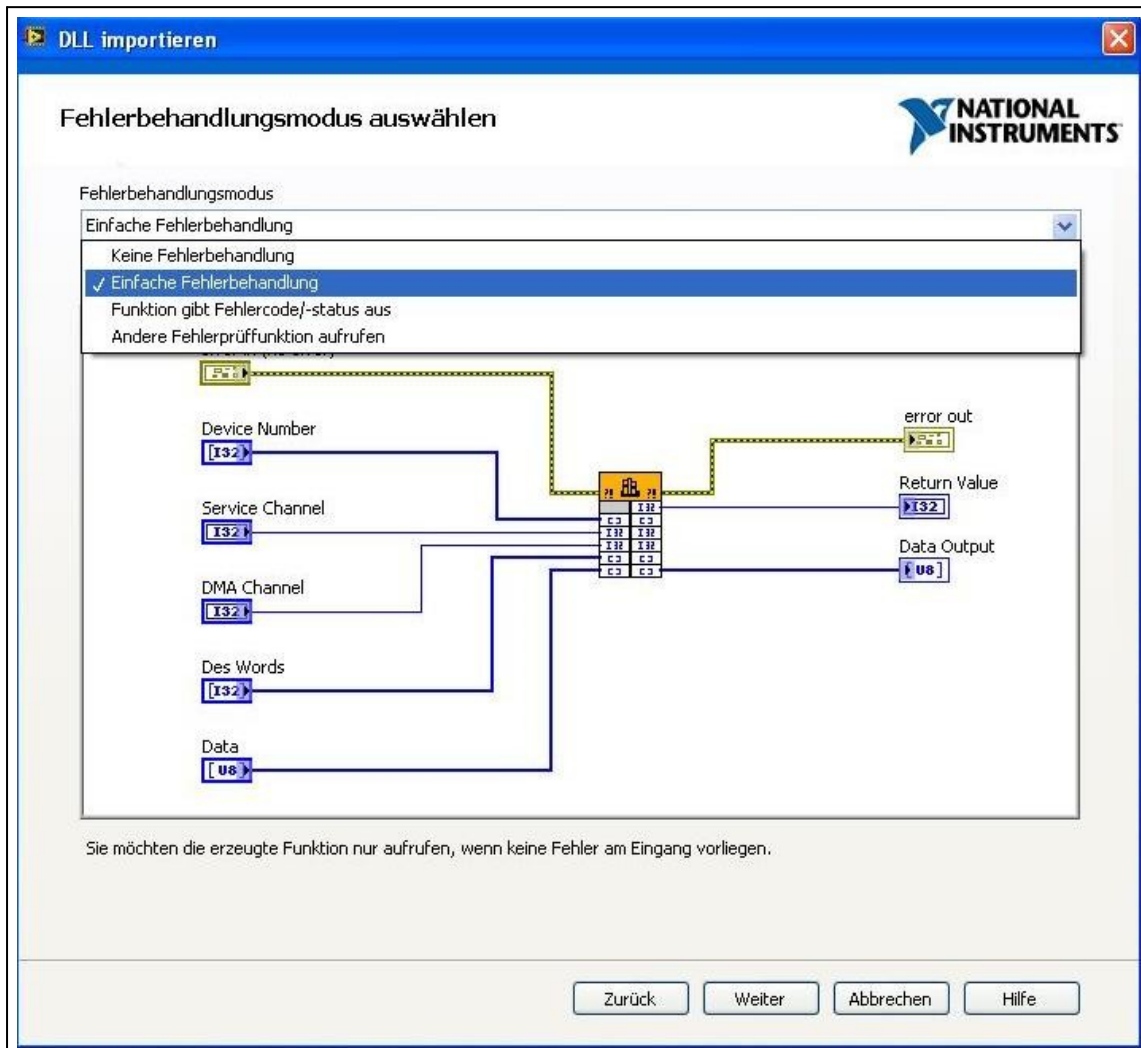
Projektbibliotheksname (.lvlib)
delib

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

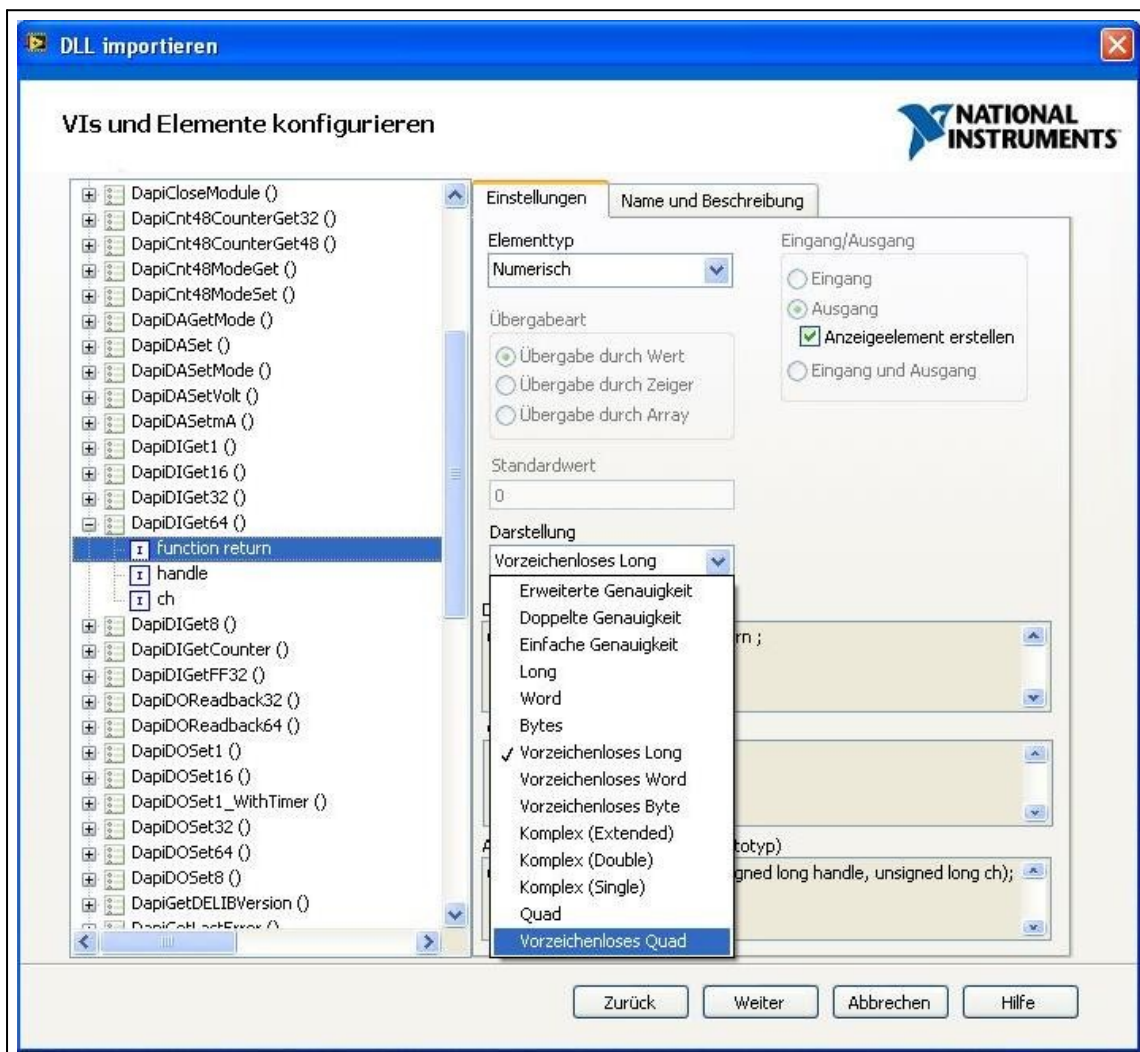
☐ DLL in Zielverzeichnis kopieren

Zurück Weiter Abbrechen Hilfe

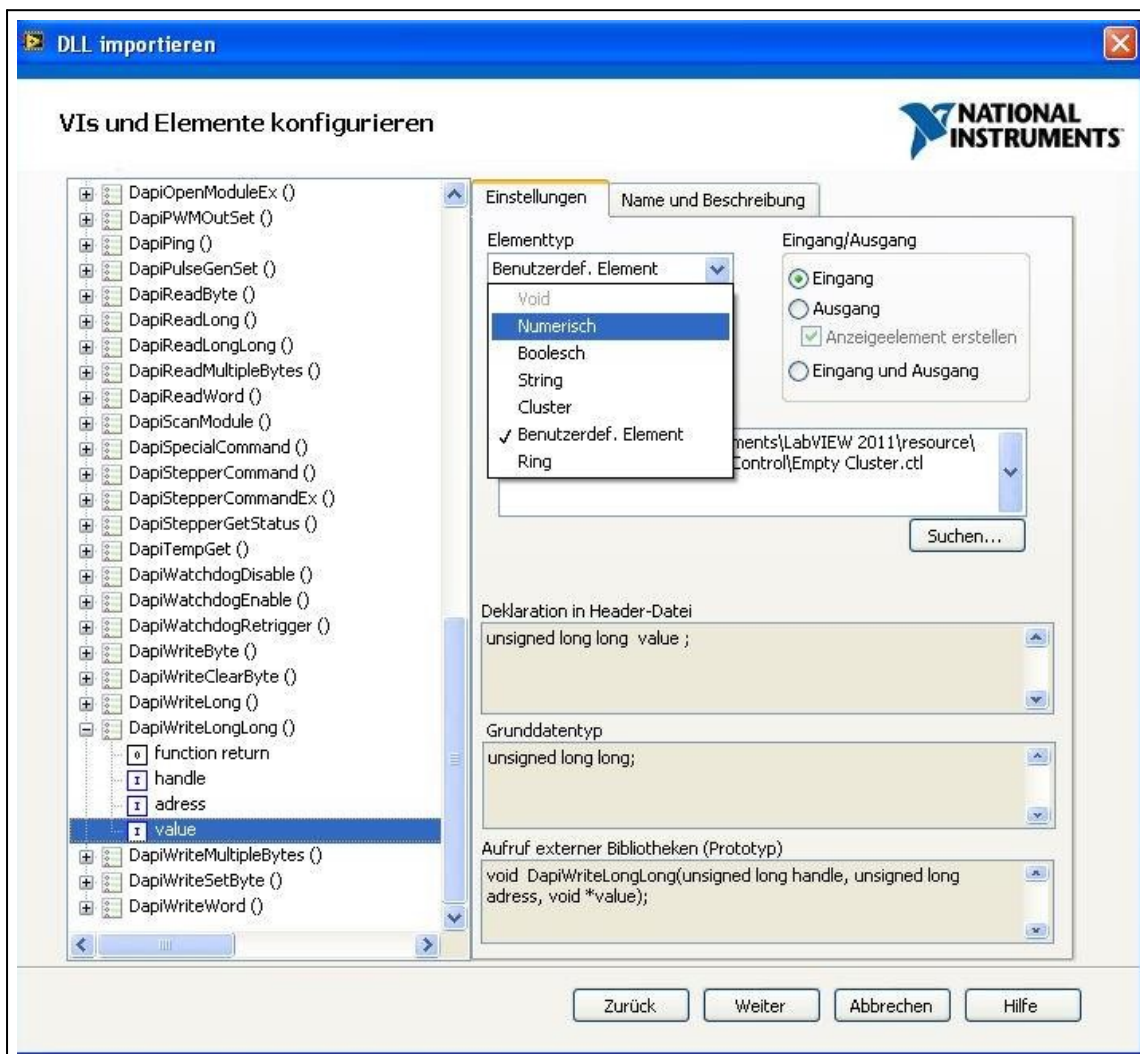
- Im folgendem Fenster wählen Sie im Drop-Down-Menü die Option "Einfache Fehlerbehandlung" aus und fahren mit "Weiter" fort.



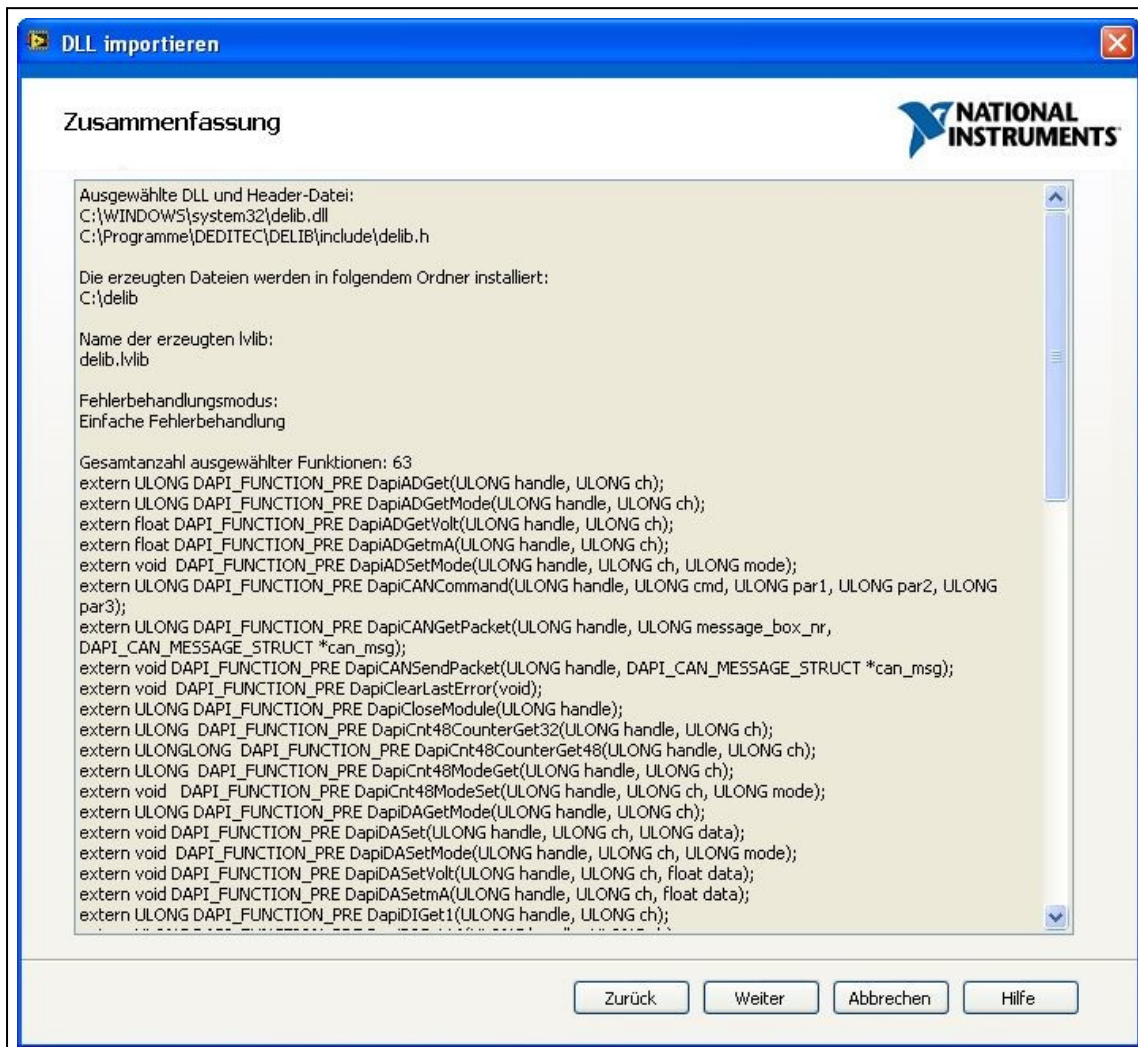
- Bei VIs die mit 64-Bit Werten arbeiten, muss die Darstellung von "Vorzeichenloses Long" in "Vorzeichenloses Quad" geändert werden.
- Folgende VIs müssen bearbeitet werden:
  - DapiCNT48CounterGet48 (function return)
  - DapiDIGet64 (function return)
  - DapiDOSet64 (data)
  - DapiDOReadBack64 (function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



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



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



- Die VIs werden nun erzeugt und können verwendet werden.

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);
```

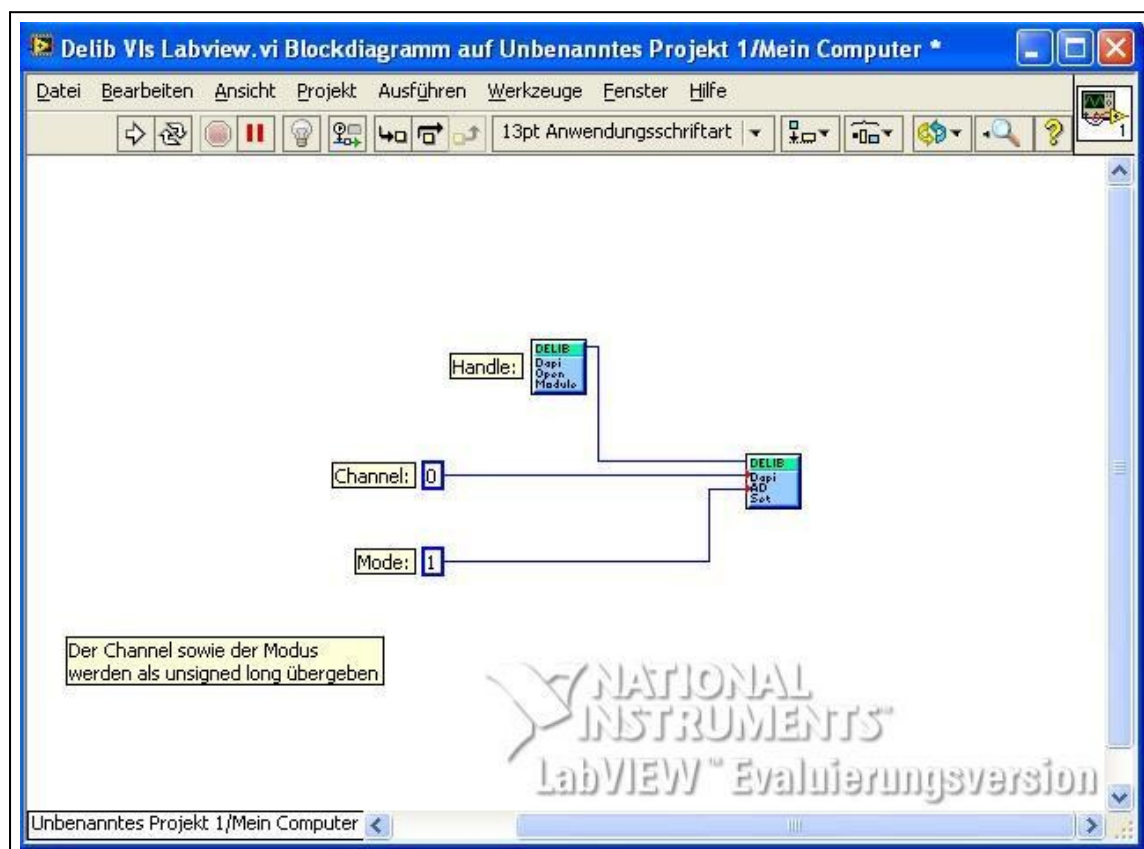
Alternativ kann man auch folgende Schreibweise verwenden:

```
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_RELAI8_8 15 // USB-OPTOIN-8-RELAIS-8  
#define USB_OPTOIN_16_RELAI8_16 16 // USB-OPTOIN-16-RELAIS-16  
#define USB_OPTOIN_32 16 // USB-OPTOIN-16-RELAIS-16  
#define USB_RELAI8_32 16 // USB-OPTOIN-16-RELAIS-16  
#define USB_OPTOIN_32_RELAI8_32 17 // USB-OPTOIN-32-RELAIS-32  
#define USB_OPTOIN_64 17 // USB-OPTOIN-32-RELAIS-32  
#define USB_RELAI8_64 17 // USB-OPTOIN-32-RELAIS-32  
#define USB_TTL_32 18 // USB-TTL-32  
#define USB_TTL_64 18 // USB-TTL-64  
  
#define MAX_NR_OF_MODULES 18
```

Beispiel in C:

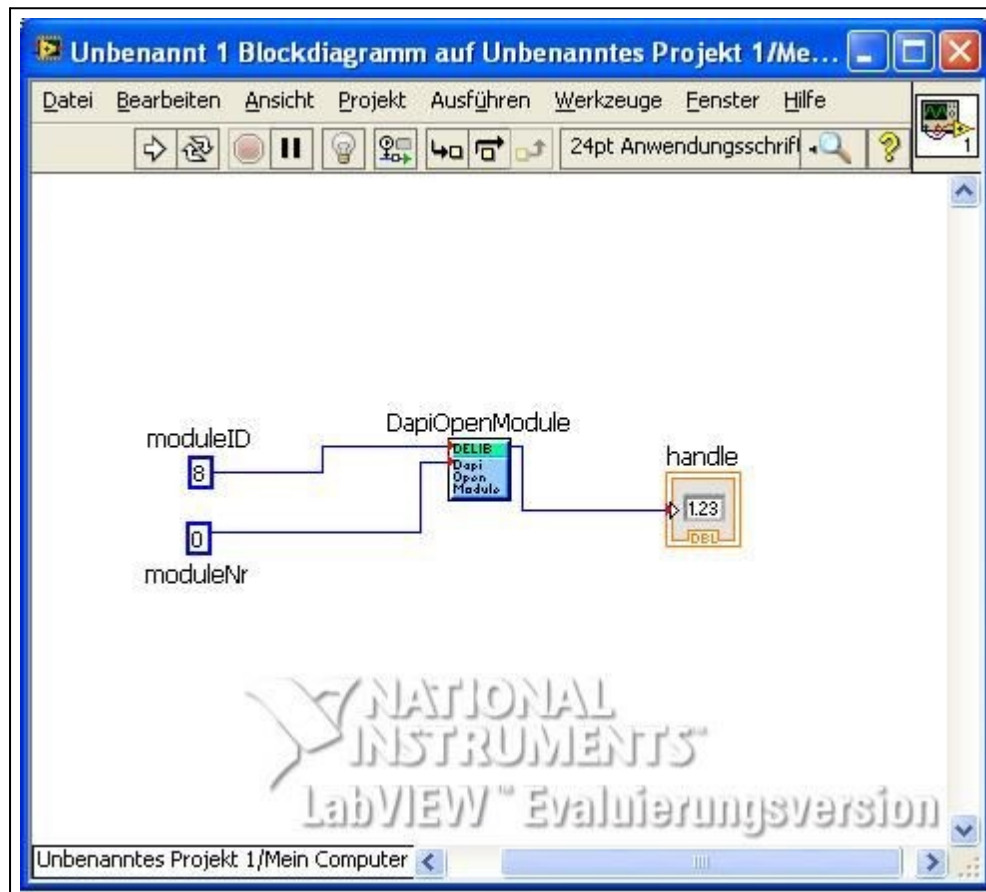
```
handle = DapiOpenModule(RO_ETH, 0); // öffnet ein RO-ETH-Modul mit Modul-Nr 0.
```

Alternativ kann man auch folgende Schreibweise verwenden:

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

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

Beispiel in Labview:



4.1.6.9. Einbinden der DELIB in Java

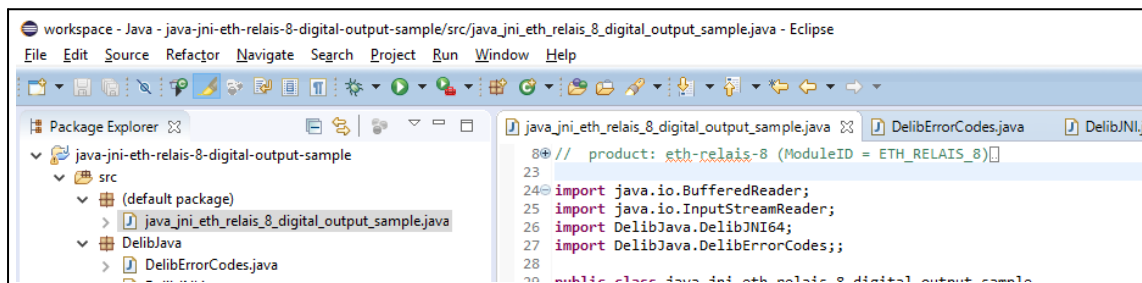
Die benötigten Dateien für Java befinden sich, je nach DELIB-Installation, in folgendem Verzeichnis

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

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

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

Anschließend müssen die verwendeten Module noch importiert werden.



4.2. DELIB Treiberbibliothek

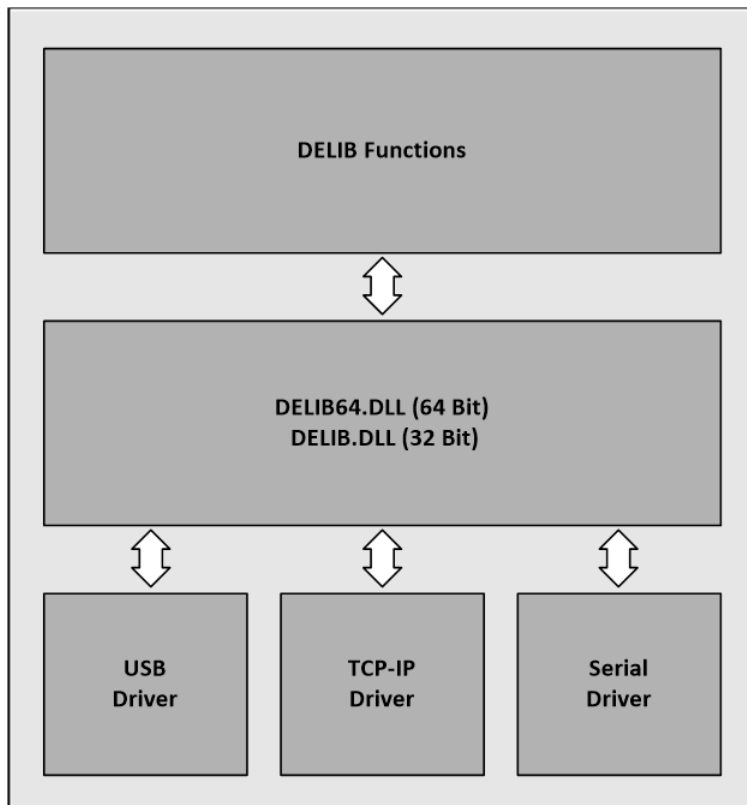
Die DELIB Treiberbibliothek enthält die DELIB-API und verschiedene Programme für den Konfigurationstest unserer Produkte.

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

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

4.2.1. Übersicht

Die folgende Abbildung erläutert den Aufbau der DELIB Treiberbibliothek



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

- Betriebssystem unabhängig
- Programmiersprachen unabhängig
- Produkt unabhängig

Diese Versionen der Treiberbibliothek bieten wir an:

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

DELIB Treiberbibliothek ETH

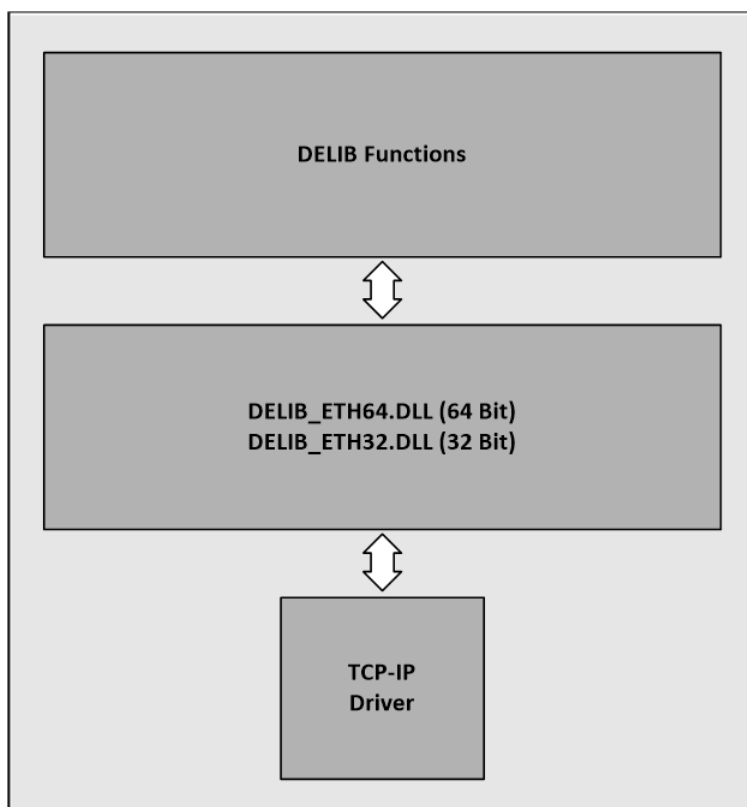
Während die DELIB für ALLE Produkte zur Verfügung steht, wird bei der DELIB-ETH auf keine weiteren Treiber zugegriffen (wie z. B. USB).

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

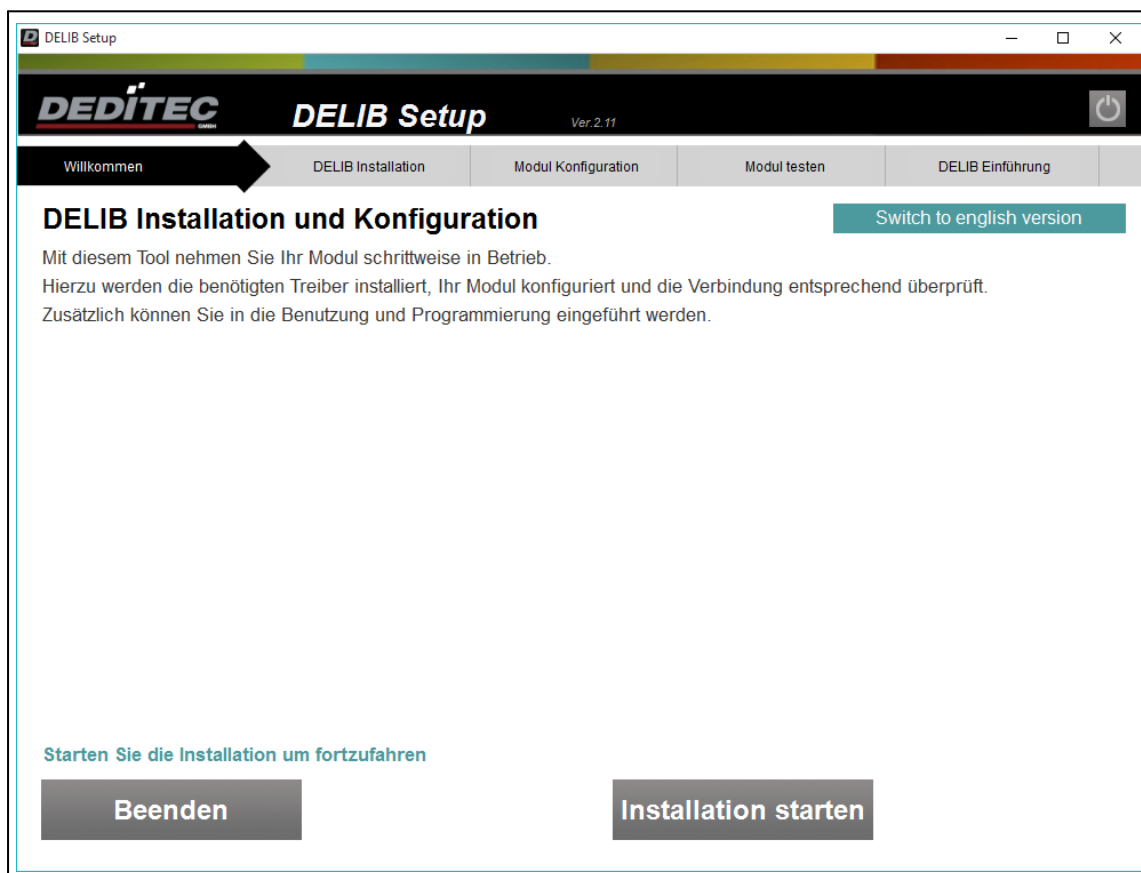
Das DELIB Setup führt Sie durch die Installation unserer DELIB Treiberbibliothek.

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.

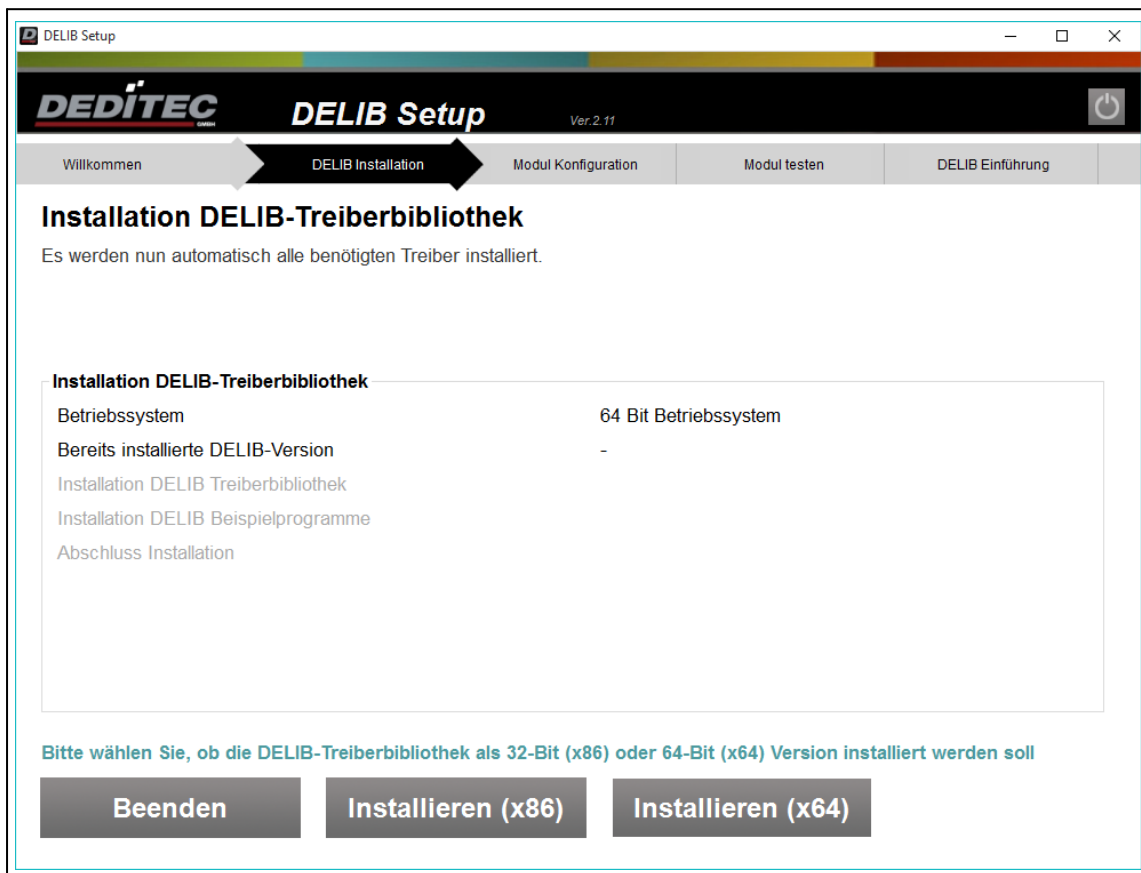
Link: <https://www.deditec.de/delib>

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

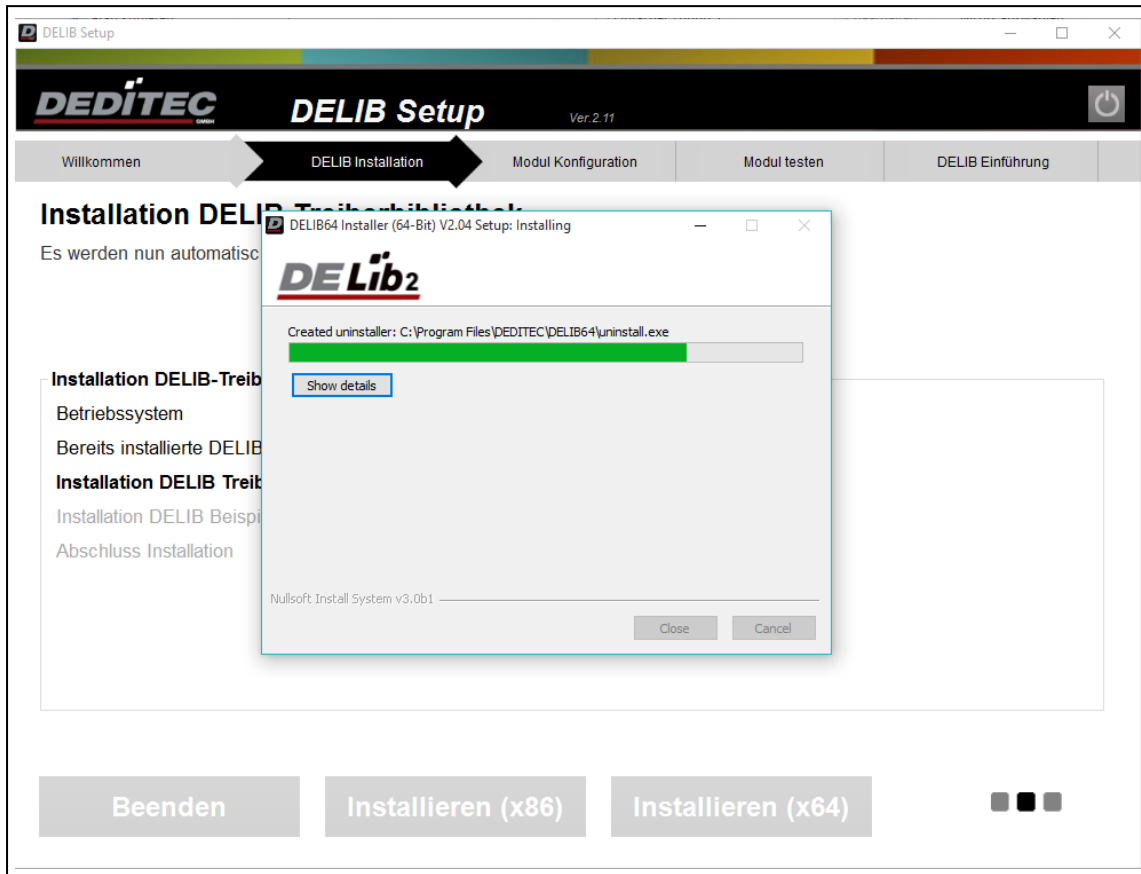


Das DELIB Setup prüft das Betriebssystem und ob bereits Versionen der DELIB Treiberbibliothek installiert sind.

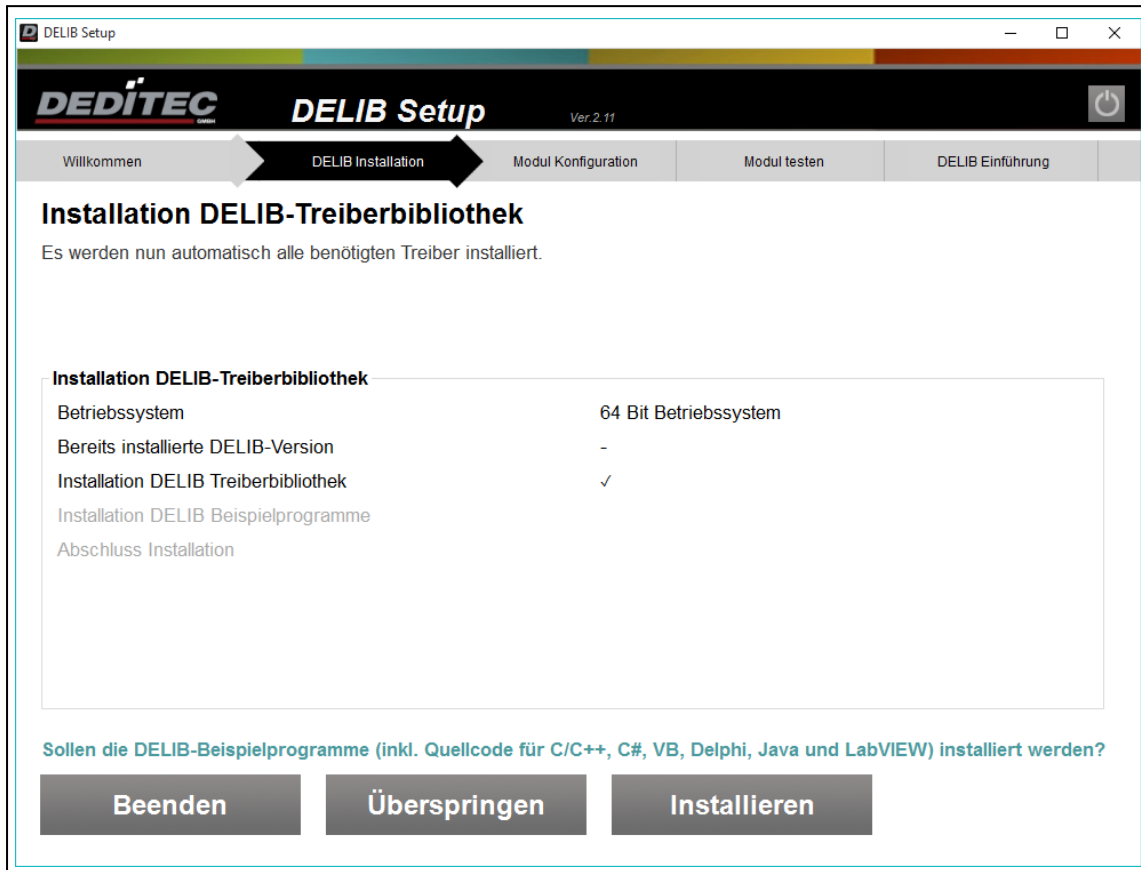
Anschließend können Sie wählen ob Sie die 32-Bit oder 64-Bit Version der DELIB Treiberbibliothek installieren möchten.



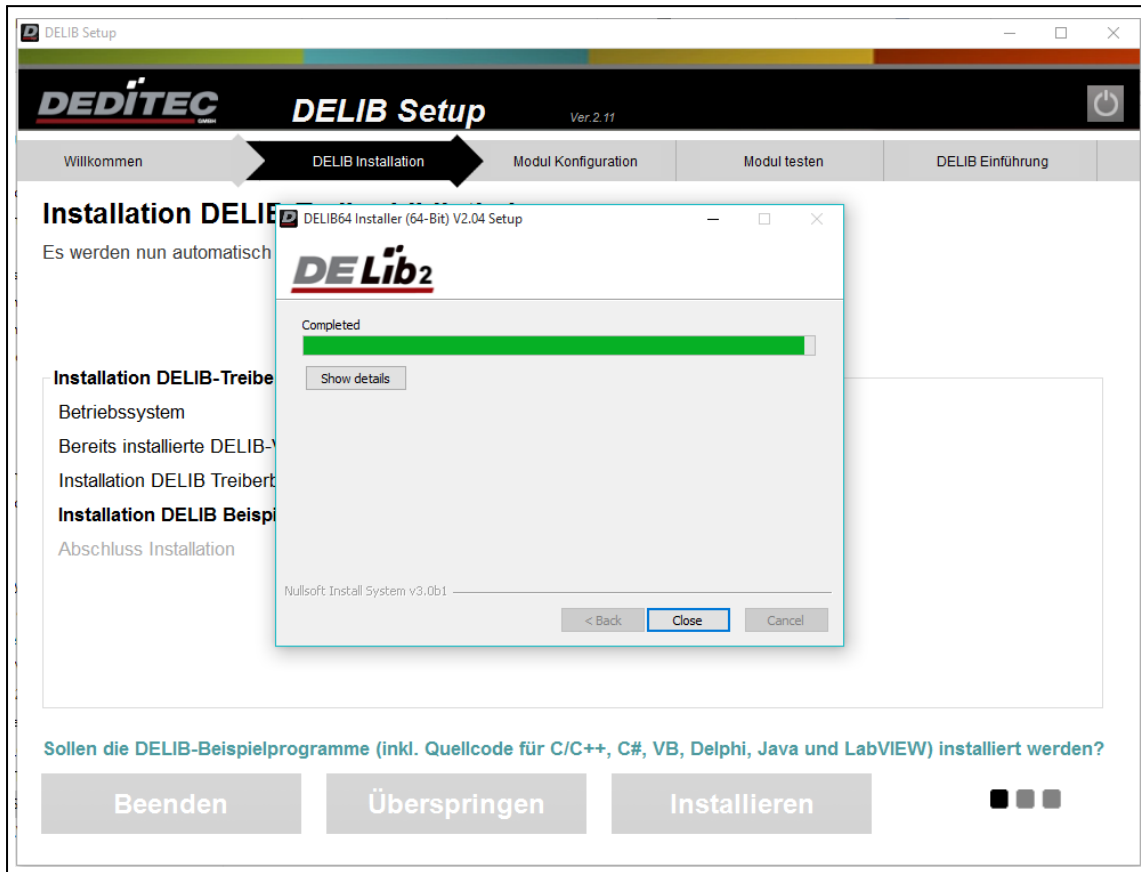
Installationsfortschritt der Treiberbibliothek.



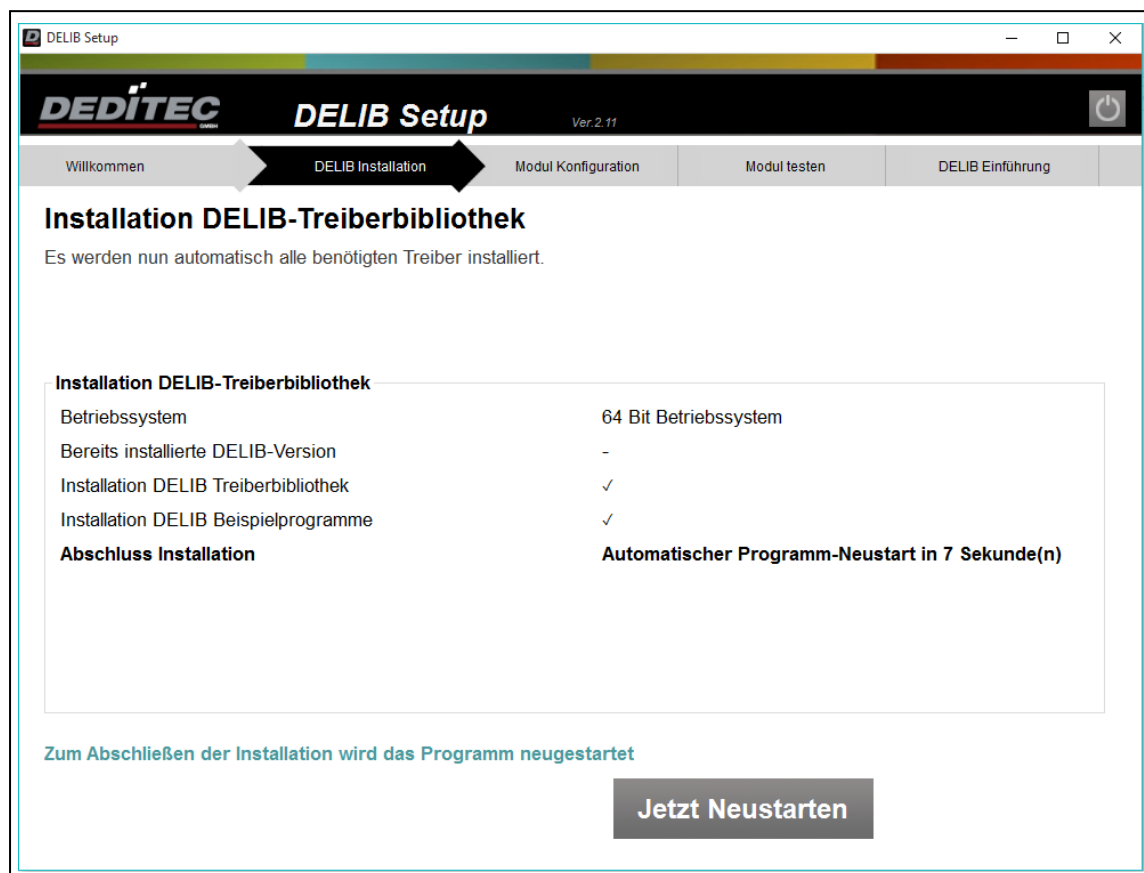
Zusätzlich können Sie wählen, ob Sie die DELIB Beispielprogramme installieren.



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 Konfig erstellen oder vorhandene Konfig bearbeiten

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

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

The screenshot shows the 'DELIB Configuration Utility' window. The title bar indicates 'Ver. 2.11' and 'DELIB 2.101 (64 Bit)'. The main menu includes 'Modul Auswahl', 'Modul Konfiguration', 'Konfiguration testen', 'Modul testen', and 'DELIB Einführung'. The 'Modul Konfiguration' tab is active, showing instructions in German and a 'Switch to english version' button. Under 'Neues Modul', there are radio buttons for 'Schnittstelle' (USB, ETH, CAN, SER). Under 'Konfigurierte Module', a list shows 'RO-ETH (1)' and 'RO-SER (0)'. At the bottom, there are 'Beenden' and 'Weiter' buttons.

DELIB Configuration Utility

Ver. 2.11 DELIB 2.101 (64 Bit)

Modul Auswahl Modul Konfiguration Konfiguration testen Modul testen DELIB Einführung

Modul Konfiguration

Switch to english version

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

Neues Modul

Schnittstelle

- ☐ USB
- ☐ ETH
- ☐ CAN
- ☐ SER

Konfigurierte Module

Module

- RO-ETH (1)
- RO-SER (0)

Wählen Sie die Schnittstelle oder ein bereits konfiguriertes Modul aus

Beenden Weiter

4.2.3.2.1. Modul Konfiguration USB

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

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

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

The screenshot shows the 'DELIB Configuration Utility' window. The title bar reads 'DELIB Configuration Utility'. The main header features the 'DEDITEC' logo, the title 'DELIB Configuration Utility', the version 'Ver. 2.19', and 'DELIB 2.19 (64 Bit)' with a power icon. Below the header is a navigation bar with five tabs: 'Modul Auswahl', 'Modul Konfiguration' (which is active and highlighted with a black arrow), 'Konfiguration testen', 'Modul testen', and 'DELIB Einführung'. The main content area is titled 'Konfiguration Module mit USB Schnittstelle'. Below this title, it says: 'Hier kann die Modul-Nr. des ausgewählten Moduls geändert werden. Die Modul-Nr. dient zur Unterscheidung mehrerer identischer USB-Module einer USB-Serie.' To the right of this text is a 'Manual' icon. On the left, under the heading 'Gefundene USB-Module', there is a table with two columns: 'Module' and 'Nr.'. The table contains one entry: 'USB-OPT/REL-8' with the number '0'. To the right of the table, there are three input fields: 'Aktuelle Modul-Nr.' with the value '0', 'Neue Modul-Nr.' with a dropdown menu showing '1', and 'Übertragungsversuche (für alle USB-Module)' with a dropdown menu showing '5'. A teal button labeled 'Neue Modul-Nr. setzen' is positioned between the 'Neue Modul-Nr.' and 'Übertragungsversuche' fields. At the bottom of the window, there are three large buttons: 'Hauptmenü', 'Test', and 'Fertig'.

Module	Nr.
USB-OPT/REL-8	0

Aktuelle Modul-Nr. 0

Neue Modul-Nr. 1

Übertragungsversuche (für alle USB-Module) 5

Neue Modul-Nr. setzen

Hauptmenü Test Fertig

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

Mit dem Punkt "Neue Module-Nr" kann dem Produkt eine neue Nummer zwischen 0 und 255 zugewiesen werden. Im Auslieferungszustand haben alle Produkte die Modul-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-O16 und USB-RELAIS-8) im System, ist keine Konfiguration nötig, da die Produkte über die ID eindeutig identifizierbar sind.

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

Schritt 1

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

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

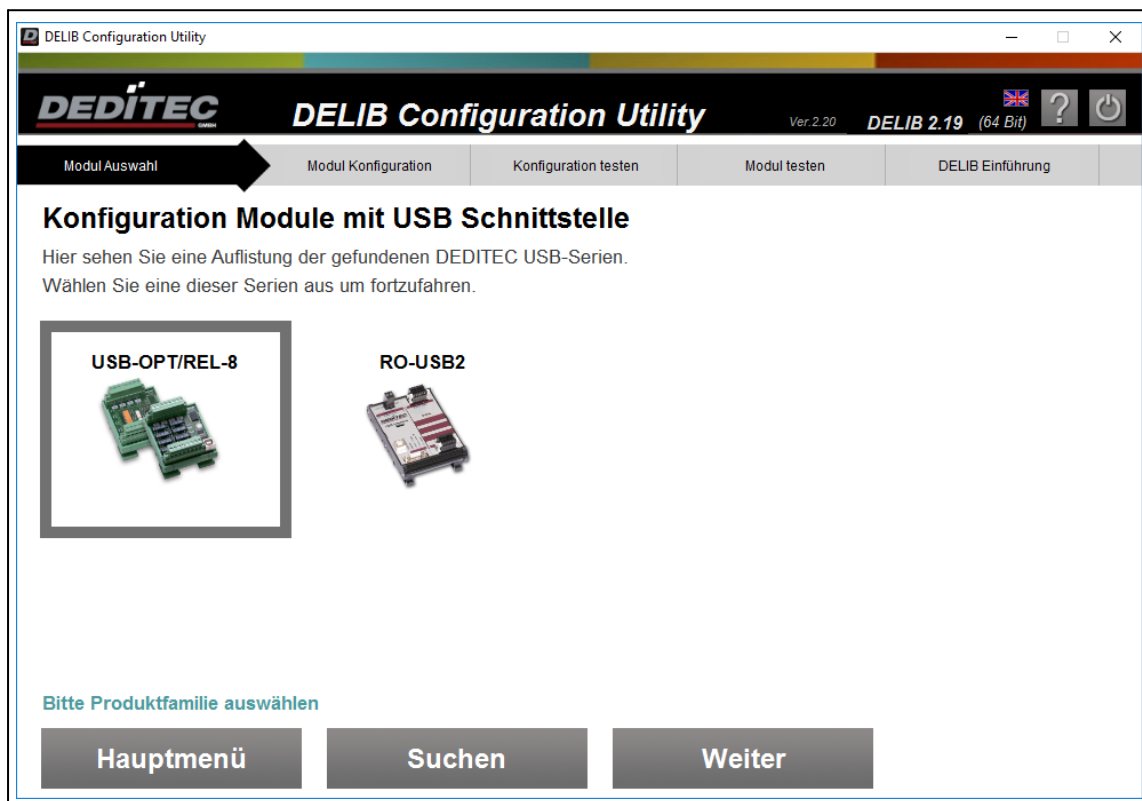
The screenshot shows the 'DELIB Configuration Utility' window. The title bar includes the DEDITEC logo, the application name, version 'Ver. 2.20', and 'DELIB 2.19 (64 Bit)'. The main menu has five tabs: 'Modul Auswahl' (selected), 'Modul Konfiguration', 'Konfiguration testen', 'Modul testen', and 'DELIB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

Schritt 2

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

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

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



Schritt 3

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

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

Module	Nr
USB-OPT/REL-8	0

Below the table is an 'Aktualisieren' button. To the right of the table are the following controls:

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

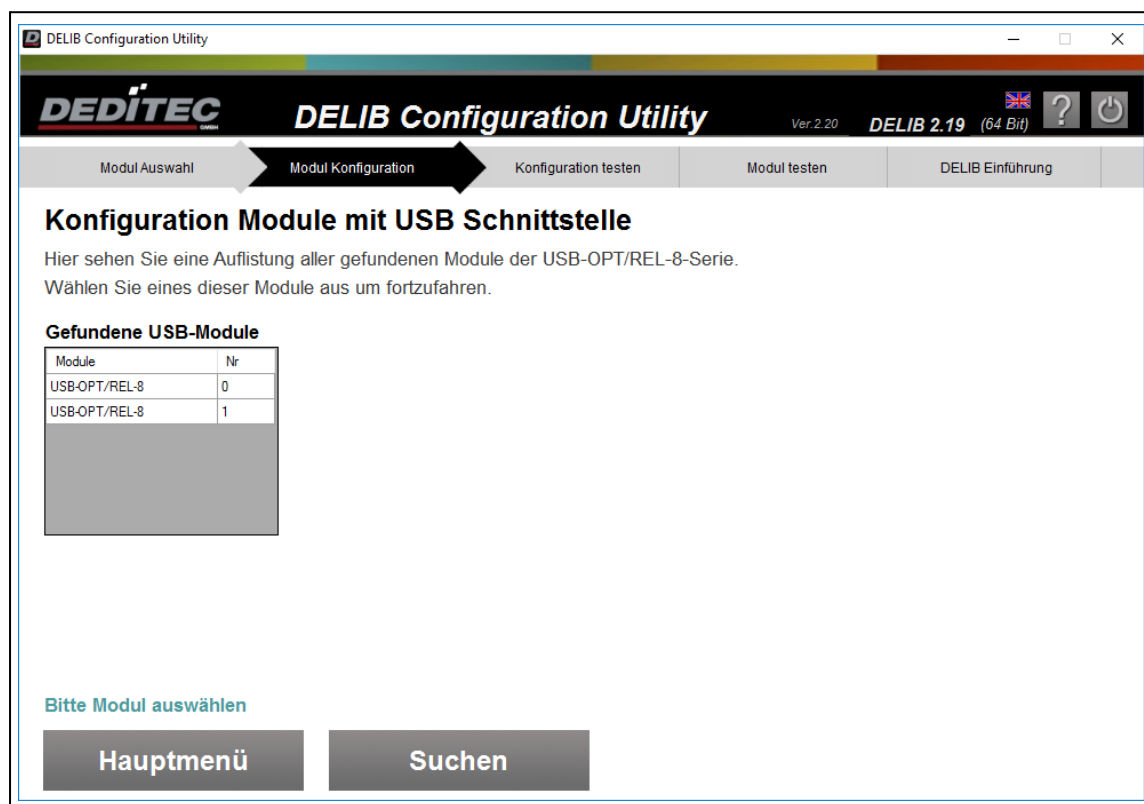
At the bottom of the window are three large buttons: 'Hauptmenü', 'Test', and 'Weiter'.

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:

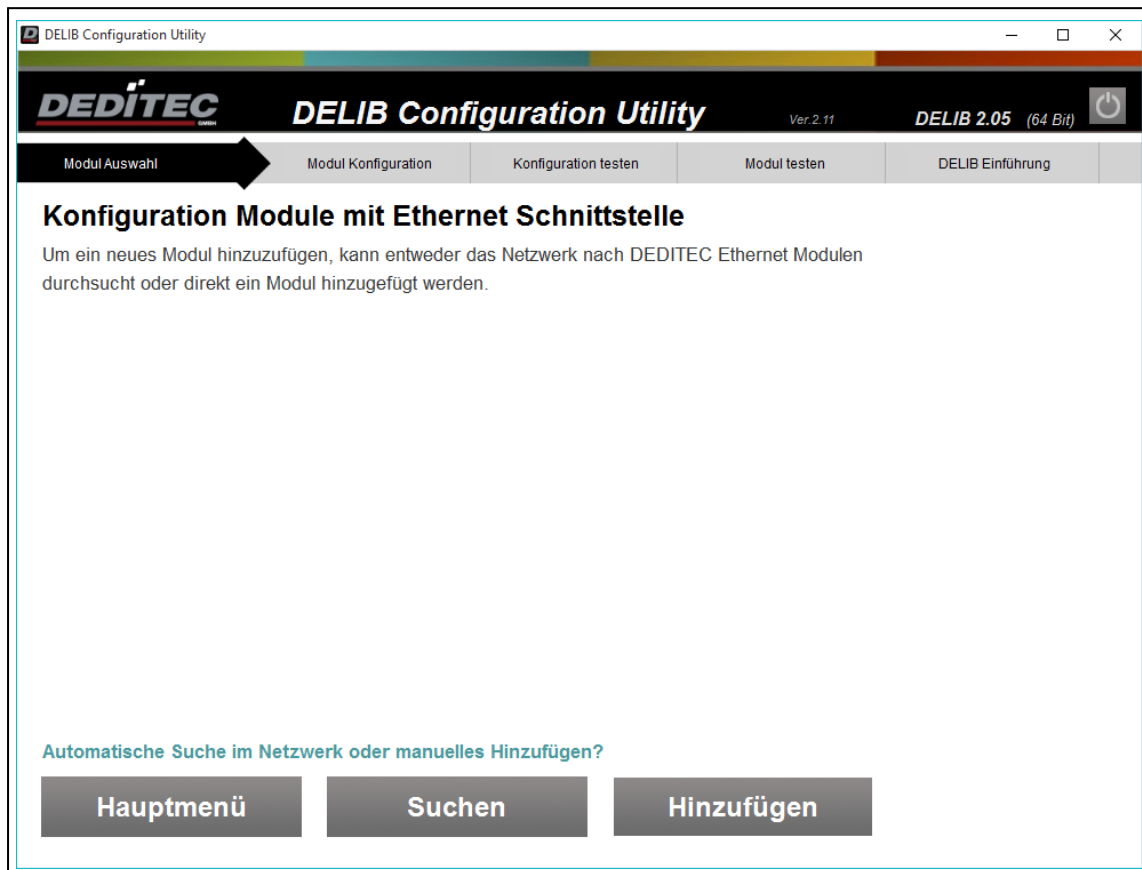
Alle Module haben als Werkseinstellung die NR "0".

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

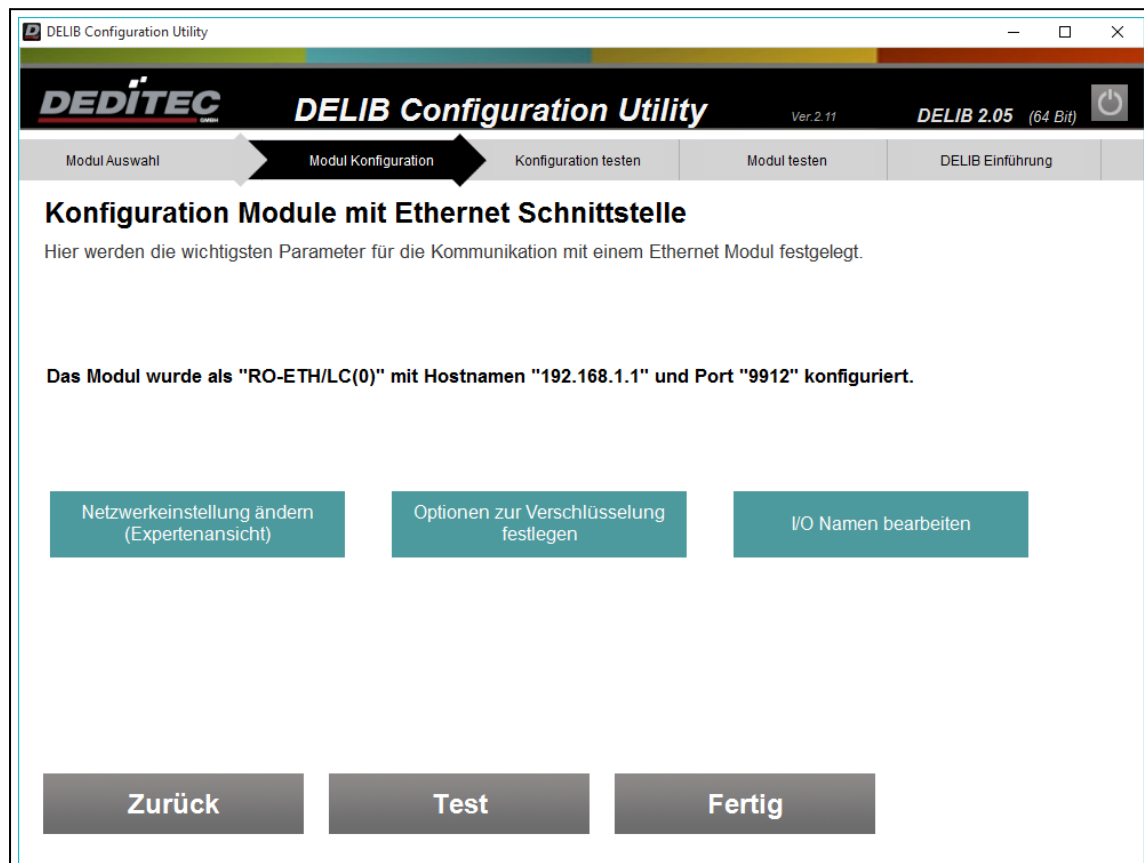
Bei der Konfiguration von Ethernet Produkten kann das Netzwerk automatisch nach DEDITEC Produkten durchsucht werden.

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

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

DELIB Configuration Utility

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

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

Konfiguration Module mit Ethernet Schnittstelle

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

I/O Namen Konfiguration

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

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

Zurück Speichern

Hinweis

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

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.

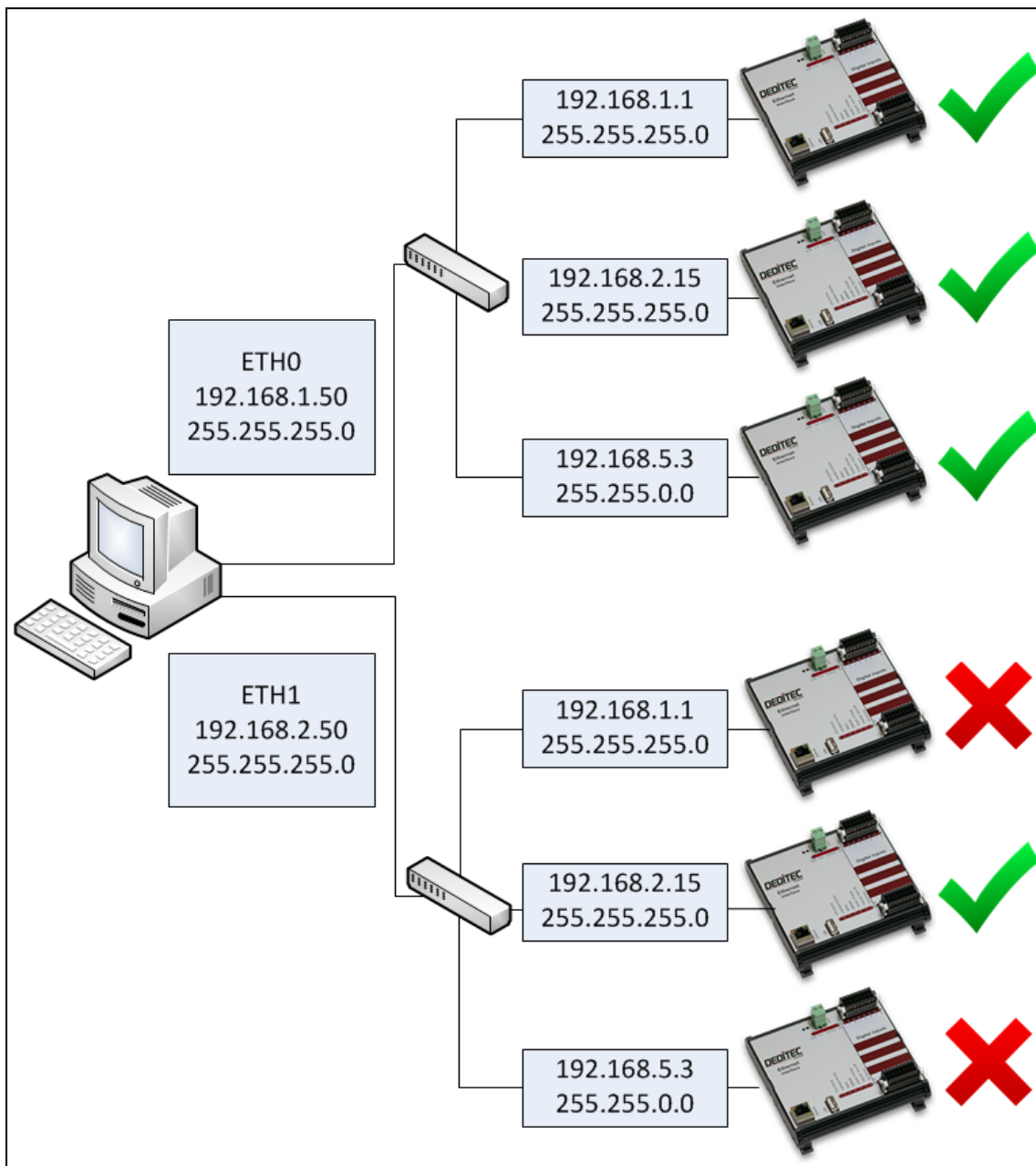
Achtung:

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.

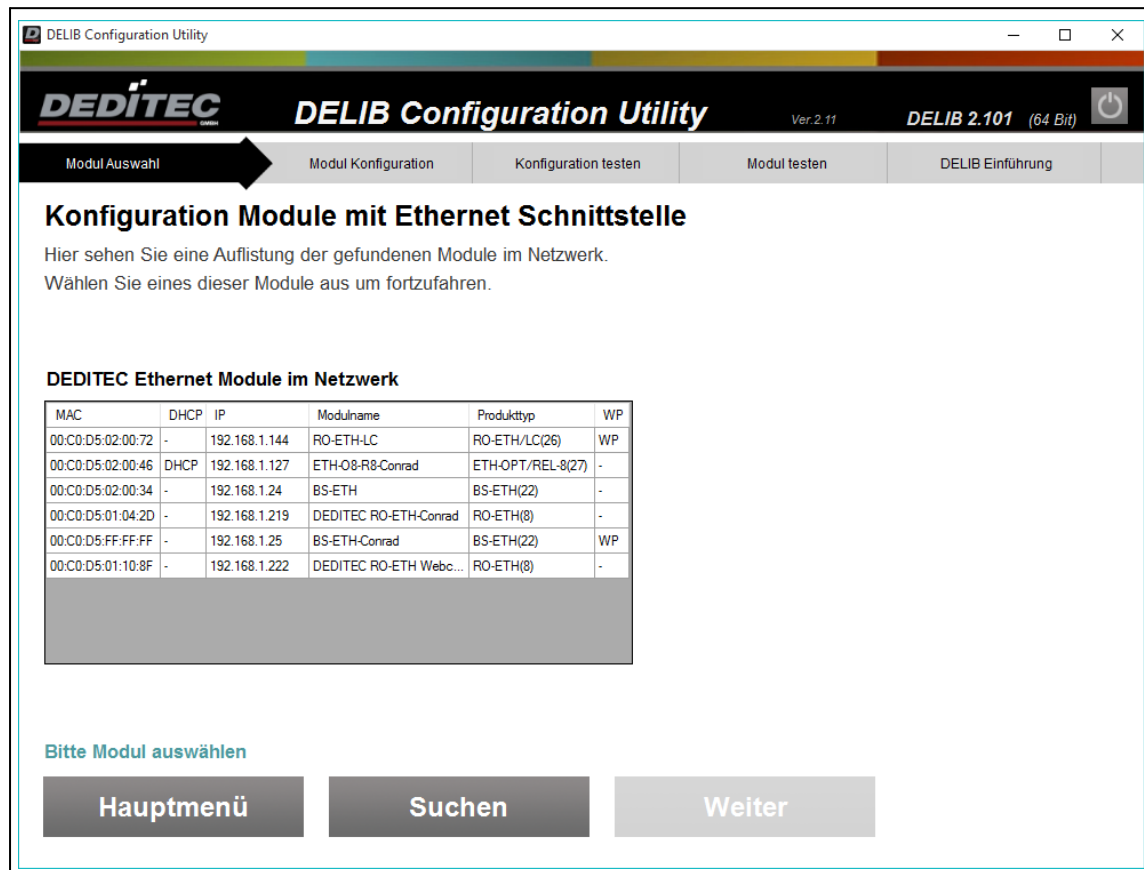
Zur Modul-Einrichtung empfiehlt es sich daher, den WLAN-Adapter kurzzeitig zu deaktivieren.

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.



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**

DELIB Configuration Utility Ver. 2.11 DELIB 2.05 (64 Bit)

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

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.

4.2.3.2.2.2. Verschlüsselung einrichten

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

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

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

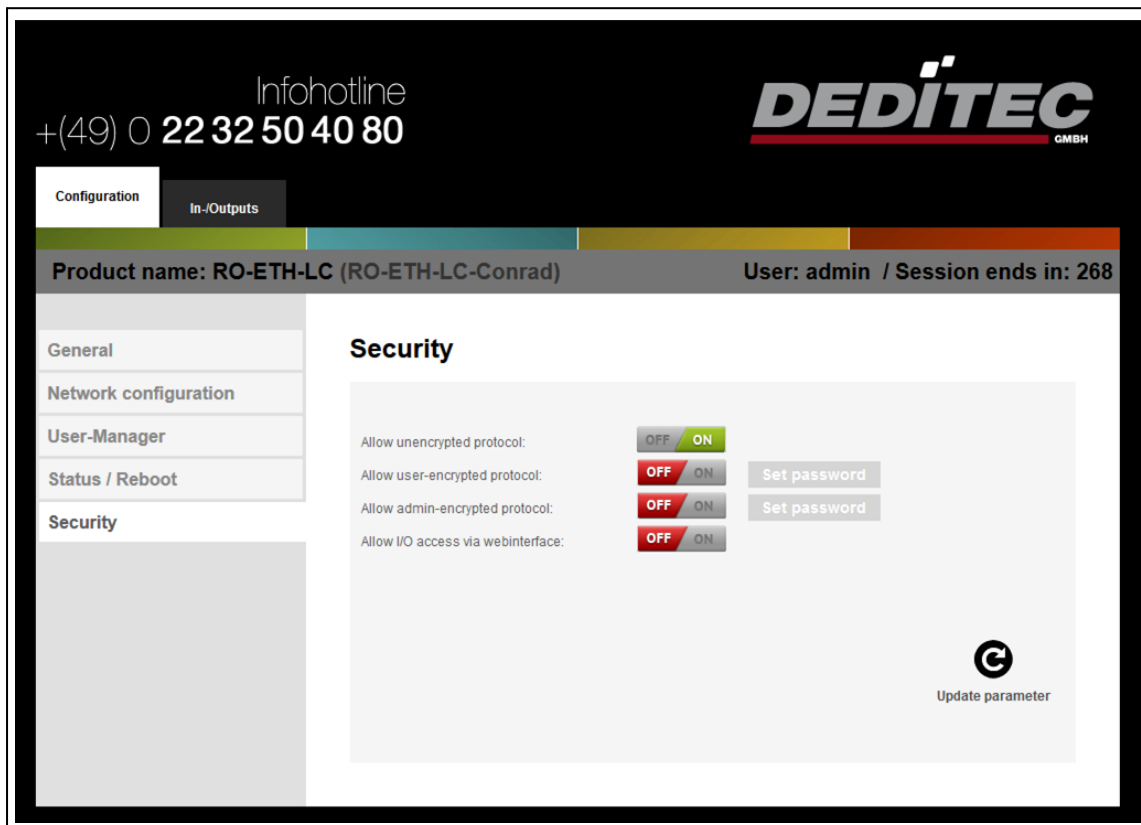
Manuelle Konfiguration

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

Schritt 1 - Konfiguration Modul-Seite

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

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



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

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

Hinweis

Bei der Eingabe des Passwortes sind alle Zeichen erlaubt.

The screenshot displays the DEDITEC web interface. At the top left, the infohotline number is + (49) 0 22 32 50 40 80. The DEDITEC GMBH logo is at the top right. Below the header, there are 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'. On the left sidebar, the 'Security' menu item is selected. The main content area is titled 'Security' and contains several settings: 'Allow unencrypted protocol' (OFF/ON), 'Allow user-encrypted protocol' (OFF/ON), 'Allow admin-encrypted protocol' (OFF/ON), 'New password' (password field), 'Confirm password' (password field), and 'Allow I/O access via webinterface' (OFF/ON). A 'Set password' button is located next to the password fields. At the bottom right, there is a circular arrow icon labeled '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 interface has a dark header with the 'DEDITEC' logo, the title 'DELIB Configuration Utility', the version 'Ver. 2.19', and 'DELIB 2.19 (64 Bit)' with a power icon. Below the header is a navigation bar with five tabs: 'Modul Auswahl', 'Modul Konfiguration' (active), 'Konfiguration testen', 'Modul testen', and 'DELIB Einführung'. The main content area is titled 'Konfiguration Module mit Ethernet Schnittstelle' and includes a sub-header 'Netzwerkkonfiguration'. It contains several input fields: 'Modul Auswahl' (dropdown menu showing 'RO-ETH/LC(0)'), 'Aktiviert' (checked checkbox), 'Hostname verwenden' (unchecked checkbox), 'IP Adresse' (text field with '192.168.1.25'), 'Port' (text field with '9912'), 'Timeout [msec]' (text field with '5000'), and 'Verschlüsselung' (dropdown menu showing 'deaktiviert'). A teal button labeled 'I/O Namen bearbeiten' is positioned to the right of the encryption dropdown. At the bottom, there are three large buttons: 'Hauptmenü', 'Test', and 'Fertig'. A small 'Manual' icon is visible in the top right corner of the main content area.

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

Das DELIB Configuration Utility kann nun geschlossen werden.

The screenshot shows the 'DELIB Configuration Utility' window. The title bar reads 'DELIB Configuration Utility'. The main header features the 'DEDITEC' logo, the title 'DELIB Configuration Utility', the version 'Ver. 2.19', and 'DELIB 2.19 (64 Bit)' with a power icon. Below the header is a navigation bar with five tabs: 'Modul Auswahl', 'Modul Konfiguration', 'Konfiguration testen' (which is highlighted with a black arrow), 'Modul testen', and 'DELIB Einführung'. The main content area is titled 'Konfiguration Module mit Ethernet Schnittstelle'. Below this title, it says 'Hier werden die wichtigsten Parameter für die Kommunikation mit einem Ethernet Modul festgelegt.' and there is a 'Manual' icon. Under the heading 'Netzwerkconfiguration', there are several input fields: 'Modul Auswahl' (a dropdown menu showing 'RO-ETH/LC(0)'), 'Aktiviert' (checked checkbox), 'Hostname verwenden' (unchecked checkbox), 'IP Adresse' (text field with '192.168.1.25'), 'Verschlüsselung' (dropdown menu showing 'Admin'), 'Port' (text field with '9912'), 'Passwort' (password field with '*****'), and 'Timeout [msec]' (text field with '5000'). To the right of these fields is a teal button labeled 'I/O Namen bearbeiten'. Below the input fields is a teal button labeled 'Einstellungen zur Verschlüsselung auf das Modul übertragen'. At the bottom of the main content area, a green status message reads 'Modul-Kommunikation ist OK! Aktuelle Modul-Firmware ist 2.20'. At the very bottom of the window are three large buttons: 'Hauptmenü', 'Test', and 'Fertig'.

Automatische Konfiguration

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

DELIB Configuration Utility

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

Modul Auswahl Modul Konfiguration Konfiguration testen Modul testen DELIB Einführung

Konfiguration Module mit Ethernet Schnittstelle

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

Netzwerkkonfiguration

Modul Auswahl: RO-ETH/LC(0)

☒ Aktiviert ☐ Hostname verwenden

IP Adresse: 192.168.1.25 Verschlüsselung: Admin

Port: 9912 Passwort: *****

Timeout [msec]: 5000

I/O Namen bearbeiten

Einstellungen zur Verschlüsselung auf das Modul übertragen

Hauptmenü Test Fertig

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

Hinweis

Bei der Eingabe des Passwortes sind alle Zeichen erlaubt.

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

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

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

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

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

DELIB Configuration Utility

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

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

Konfiguration Module mit Ethernet Schnittstelle

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

Netzwerkconfiguration

Modul Auswahl: RO-ETH/LC(0)

☒ Aktiviert ☐ Hostname verwenden

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

Port: 9912 Passwort: *****

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

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

Authentifizierung abbrechen

Hauptmenü Test Fertig

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

The screenshot shows the 'DELIB Configuration Utility' window. The title bar reads 'DELIB Configuration Utility'. The main header features the 'DEDITEC' logo, the product name 'DELIB Configuration Utility', the version 'Ver. 2.19', and 'DELIB 2.19 (64 Bit)' with a power icon. A navigation bar 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 section 'Konfiguration Module mit Ethernet Schnittstelle' is displayed, followed by the instruction: 'Hier werden die wichtigsten Parameter für die Kommunikation mit einem Ethernet Modul festgelegt.' A 'Manual' icon is visible on the right. Under the heading 'Netzwerkconfiguration', there are several input fields and checkboxes: 'Modul Auswahl' is a dropdown menu showing 'RO-ETH/LC(0)'; 'Aktiviert' is checked, while 'Hostname verwenden' is unchecked; 'IP Adresse' is '192.168.1.25'; 'Port' is '9912'; 'Timeout [msec]' is '5000'; 'Verschlüsselung' is a dropdown menu showing 'Admin'; and 'Passwort' is masked with '*****'. A button 'I/O Namen bearbeiten' is located to the right of the encryption settings. Below these fields, a button reads 'Einstellungen zur Verschlüsselung auf das Modul übertragen'. At the bottom of the configuration area, a teal button says 'Authentifizierung abbrechen'. A status bar at the very bottom contains three buttons: 'Hauptmenü', 'Test', and 'Fertig'. To the right of these buttons, a teal box contains the text 'Schritt 2: DIP-Schalter 2 (Schreibschutz) des ETH-Moduls auf Stellung 0 (0=OFF, 1=ON) setzen'.

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 title 'DELIB Configuration Utility', the version 'Ver. 2.19', and 'DELIB 2.19 (64 Bit)' with a power icon. A progress bar at the top indicates the current step is 'Modul Konfiguration', with other steps being 'Modul Auswahl', 'Konfiguration testen', 'Modul testen', and 'DELIB Einführung'. The main content area is titled 'Konfiguration Module mit Ethernet Schnittstelle' and includes a sub-header 'Netzwerkconfiguration'. Below this, there are input fields for 'Modul Auswahl' (set to 'RO-ETH/LC(0)'), 'IP Adresse' (192.168.1.25), 'Port' (9912), and 'Timeout [msec]' (5000). There are also checkboxes for 'Aktiviert' (checked) and 'Hostname verwenden' (unchecked), a 'Verschlüsselung' dropdown (set to 'Admin'), and a 'Passwort' field (masked with asterisks). A button 'I/O Namen bearbeiten' is visible. A message 'Authentifizierung erfolgreich' is displayed in green. At the bottom, there are three buttons: 'Hauptmenü', 'Test', and 'Fertig', followed by three small square icons.

DELIB Configuration Utility

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

Modul Auswahl Modul Konfiguration Konfiguration testen Modul testen DELIB Einführung

Konfiguration Module mit Ethernet Schnittstelle

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

Netzwerkconfiguration

Modul Auswahl: RO-ETH/LC(0)

☒ Aktiviert ☐ Hostname verwenden

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

Port: 9912 Passwort: *****

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

Authentifizierung erfolgreich

Hauptmenü Test Fertig

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

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

DELIB Configuration Utility

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

Modul Auswahl Modul Konfiguration Konfiguration testen Modul testen DELIB Einführung

Konfiguration Module mit Ethernet Schnittstelle

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

Netzwerkconfiguration

Modul Auswahl: RO-ETH/LC(0)

☒ Aktiviert ☐ Hostname verwenden

IP Adresse: 192.168.1.25 Verschlüsselung: Admin

Port: 9912 Passwort: *****

Timeout [msec]: 5000

I/O Namen bearbeiten

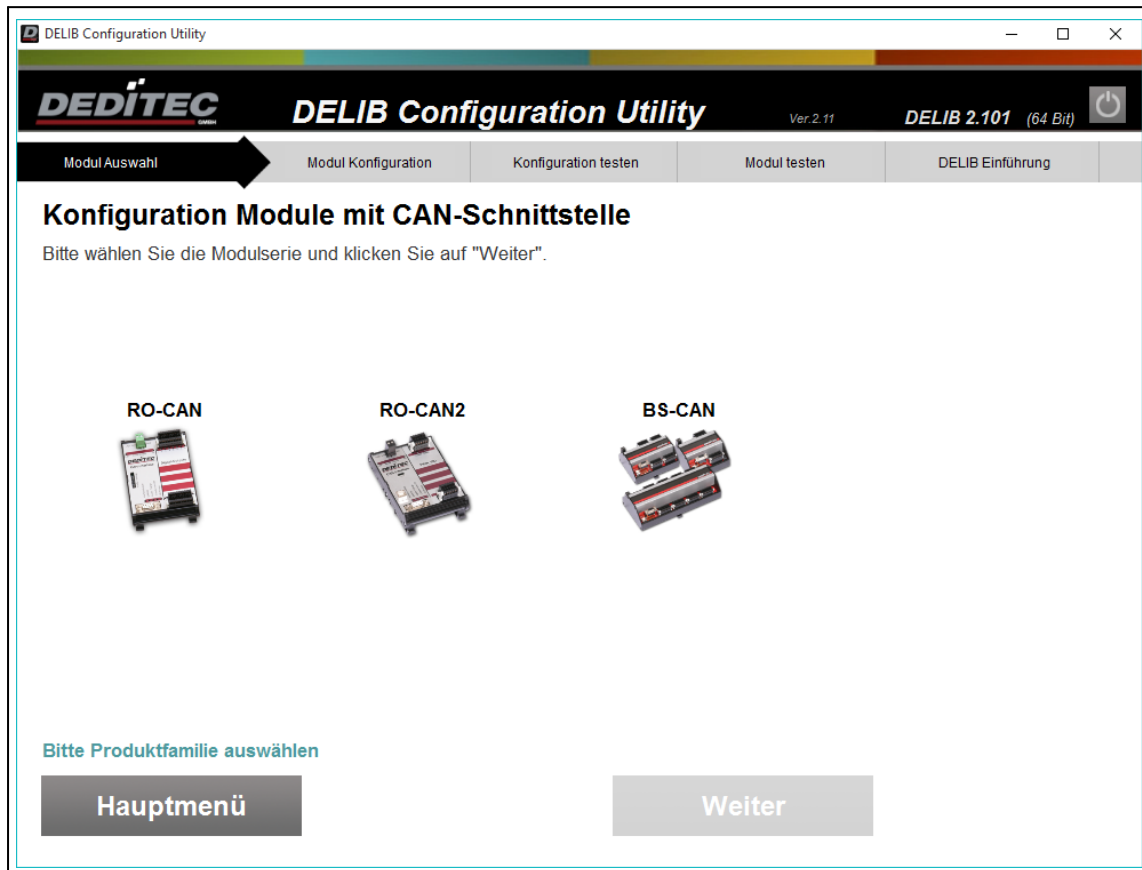
Einstellungen zur Verschlüsselung auf das Modul übertragen

Einstellungen zur Verschlüsselung erfolgreich übertragen

Hauptmenü Test Fertig

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.

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

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

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

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

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

Below the text, there is a section titled 'Gefundene USB-Module' containing a table:

Module	Nr
RO-CAN2	1

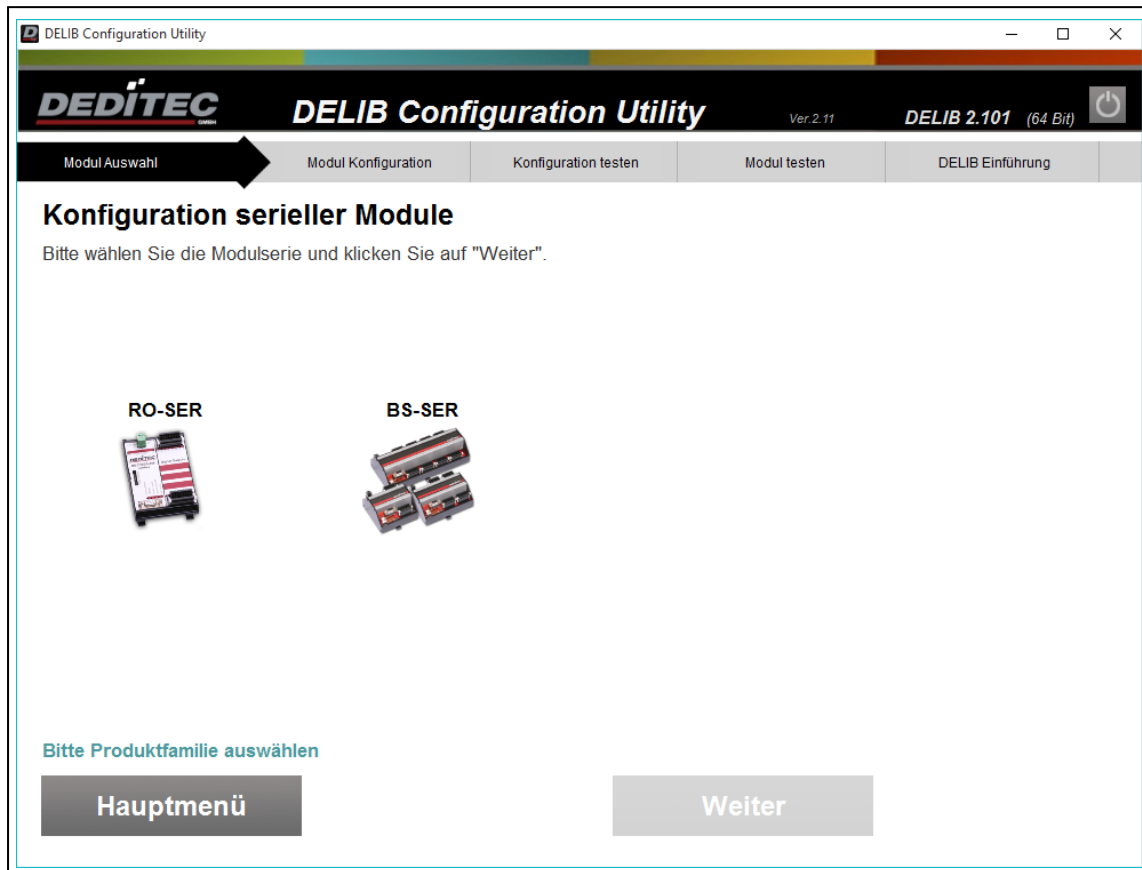
To the right of the table are three input fields:

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

At the bottom, a blue text prompt says 'Klicken Sie "Weiter" zum Übernehmen der neuen Modul-Nr.' Below this are three buttons: 'Hauptmenü', 'Suchen', and 'Weiter'.

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.

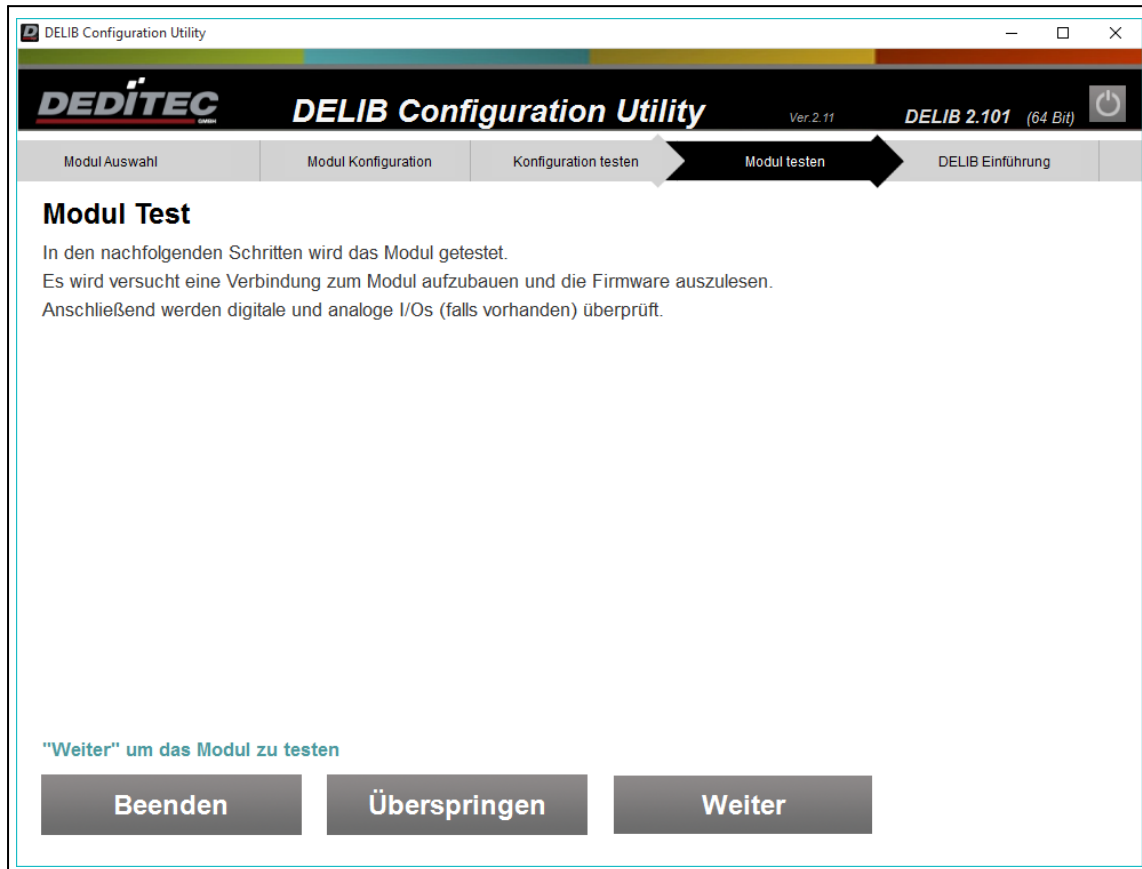
Zusätzlich muss die Anzahl der Übertragungsversuche im Falle eines Kommunikationsfehlers angegeben werden.

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

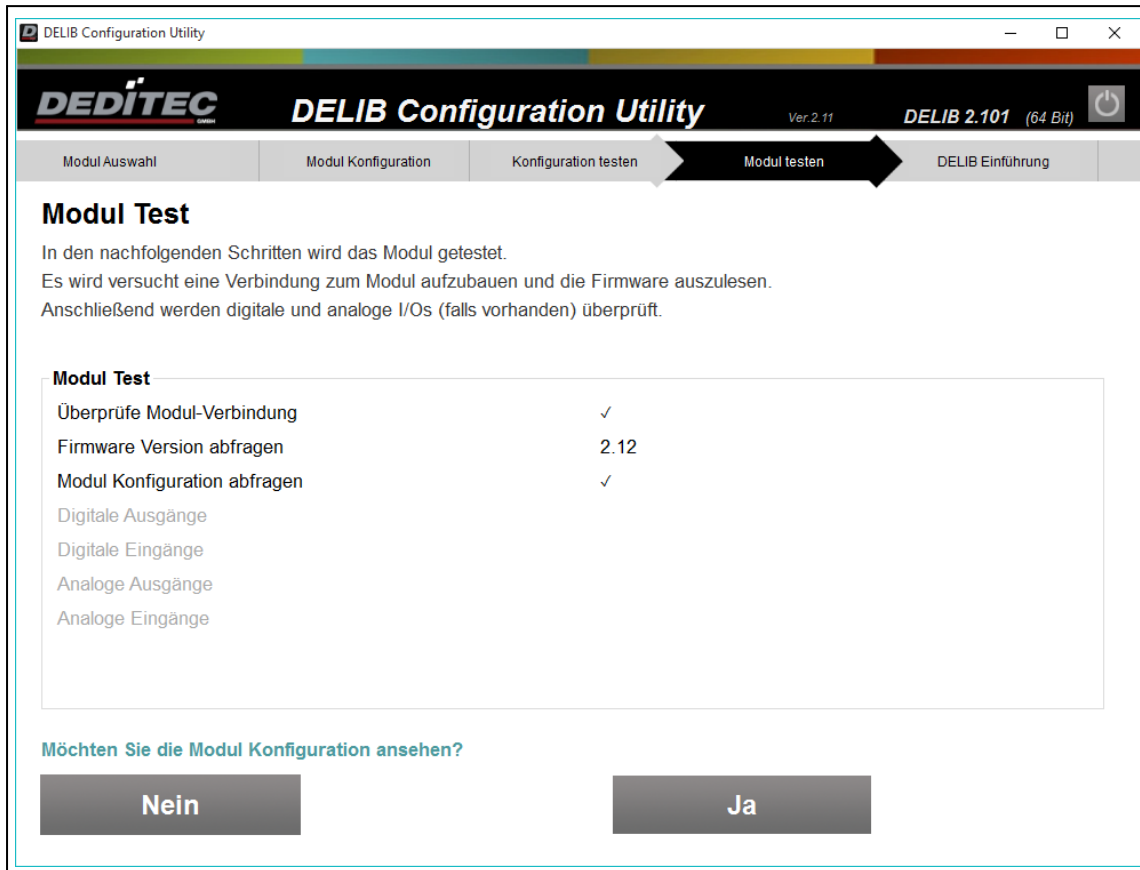
The screenshot shows the 'DELIB Configuration Utility' window. The title bar reads 'DELIB Configuration Utility'. The main header features the 'DEDITEC' logo, the product name 'DELIB Configuration Utility', the version 'Ver.2.11', and the device model 'DELIB 2.101 (64 Bit)' with a power icon. Below the header is a navigation bar with five tabs: 'Modul Auswahl', 'Modul Konfiguration' (which is the active tab and highlighted with a black arrow), 'Konfiguration testen', 'Modul testen', and 'DELIB Einführung'. 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.' Under the heading 'Modul Konfiguration', there are three configuration items: 'Modul Auswahl' with a dropdown menu showing 'RO-SER(0)', 'COM Port' with a dropdown menu showing 'COM1', and 'Übertragungsversuche' with a dropdown menu showing '5'. At the bottom of the window, there are three large 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

×

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		

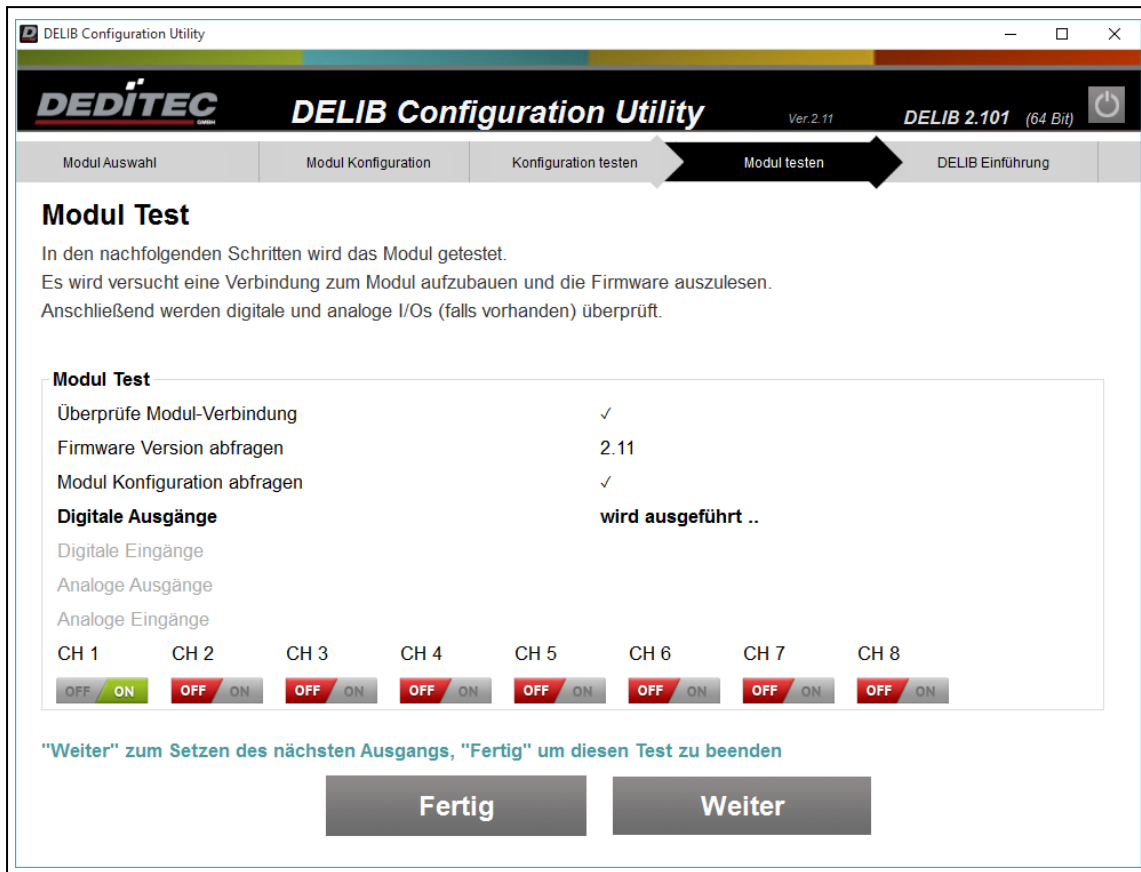
Features-General	Features-Digital I/O	Features-Analog I/O	Features-Special
Supported by FW OK	DI Commands OK	DA Commands OK	Watchdog Commands -
Dev IO registry error OK	DI CNT Commands OK	AD Commands OK	Stepper Commands OK
AD FIFO OK	DI CNT Latch Feature -	Pt100 Commands OK	
Set-Clr Bit Commands OK	DI 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-Q8_R8	FW:2.10
Sub: 2 - RO-STEPPER2	FW:1.29
Sub: 3 - RO-PT100	FW:1.03

EXIT

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

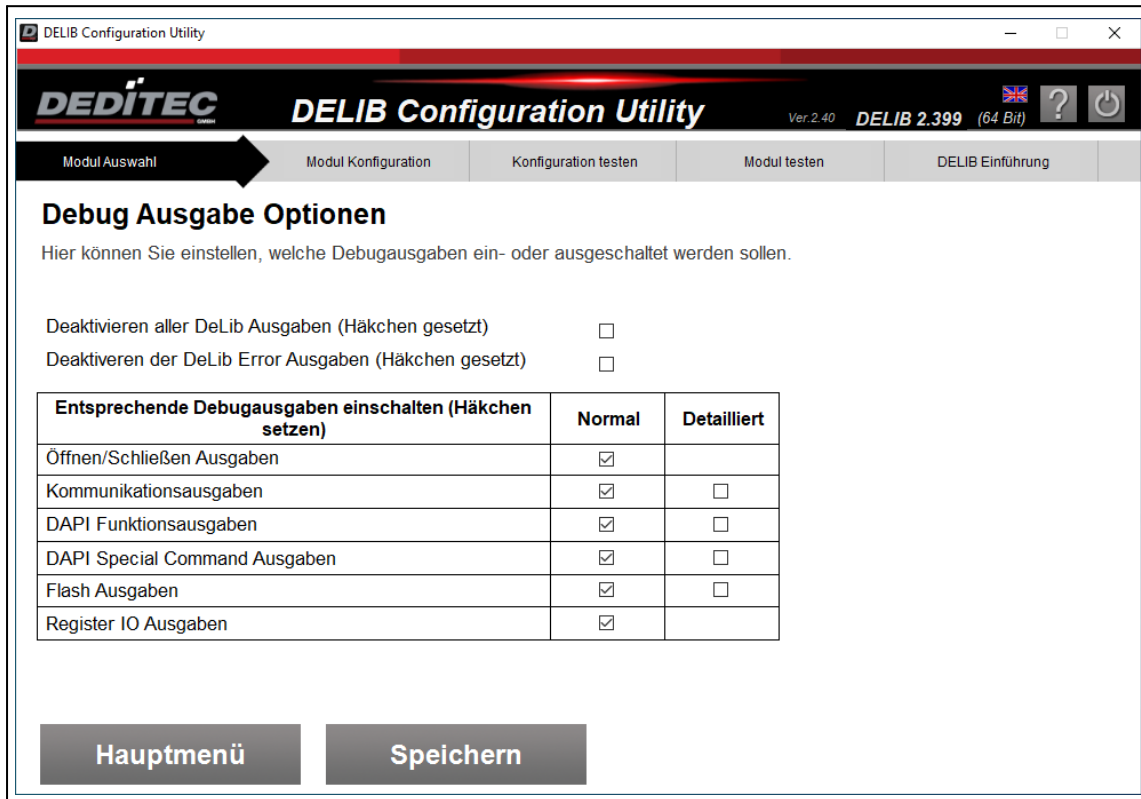


Wurden alle Tests erfolgreich durchlaufen, ist das Produkt einsatzbereit.

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

Entsprechende Debugausgaben einschalten (Häkchen setzen)	Normal	Detailliert
Öffnen/Schließen Ausgaben	☒	
Kommunikationsausgaben	☒	☐
DAPI Funktionsausgaben	☒	☐
DAPI Special Command Ausgaben	☒	☐
Flash Ausgaben	☒	☐
Register IO Ausgaben	☒	

4.2.4. Benutzung des Moduleselectors

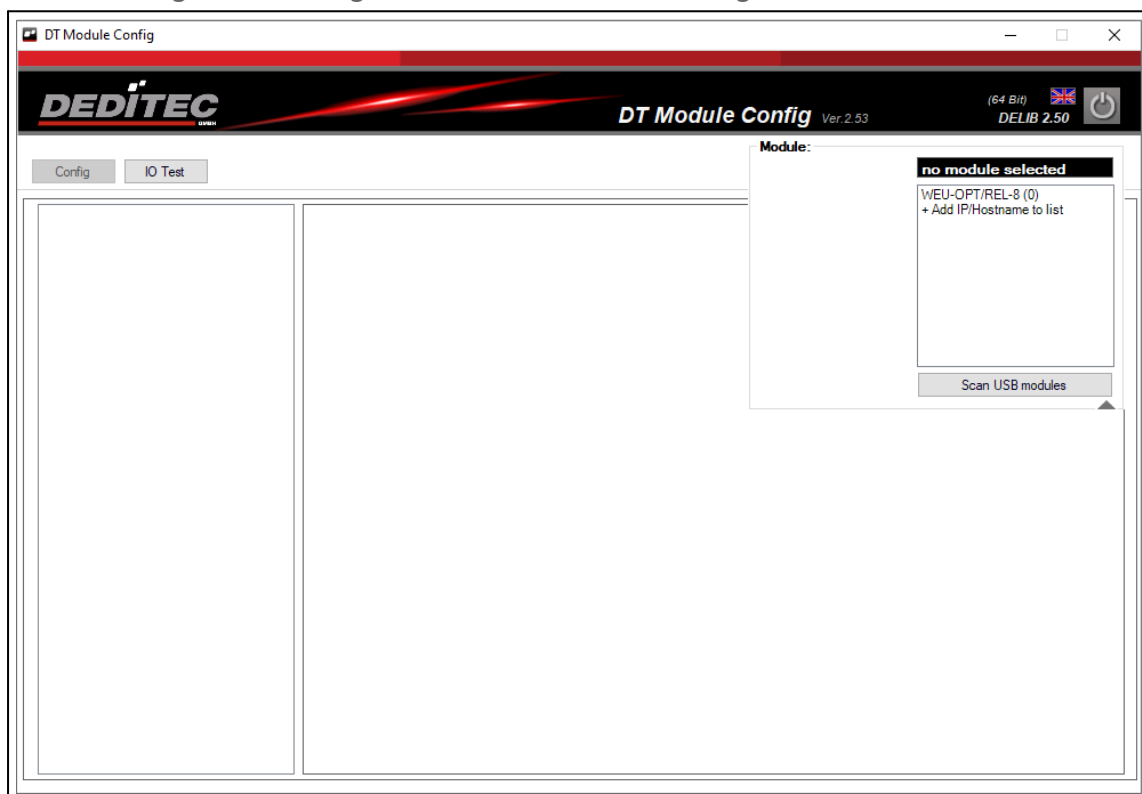
Um unsere Produkte mit der DEDITEC-Software benutzen zu können, müssen diese über den Modul Selector ausgewählt werden.

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

4.2.4.1. via USB

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

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

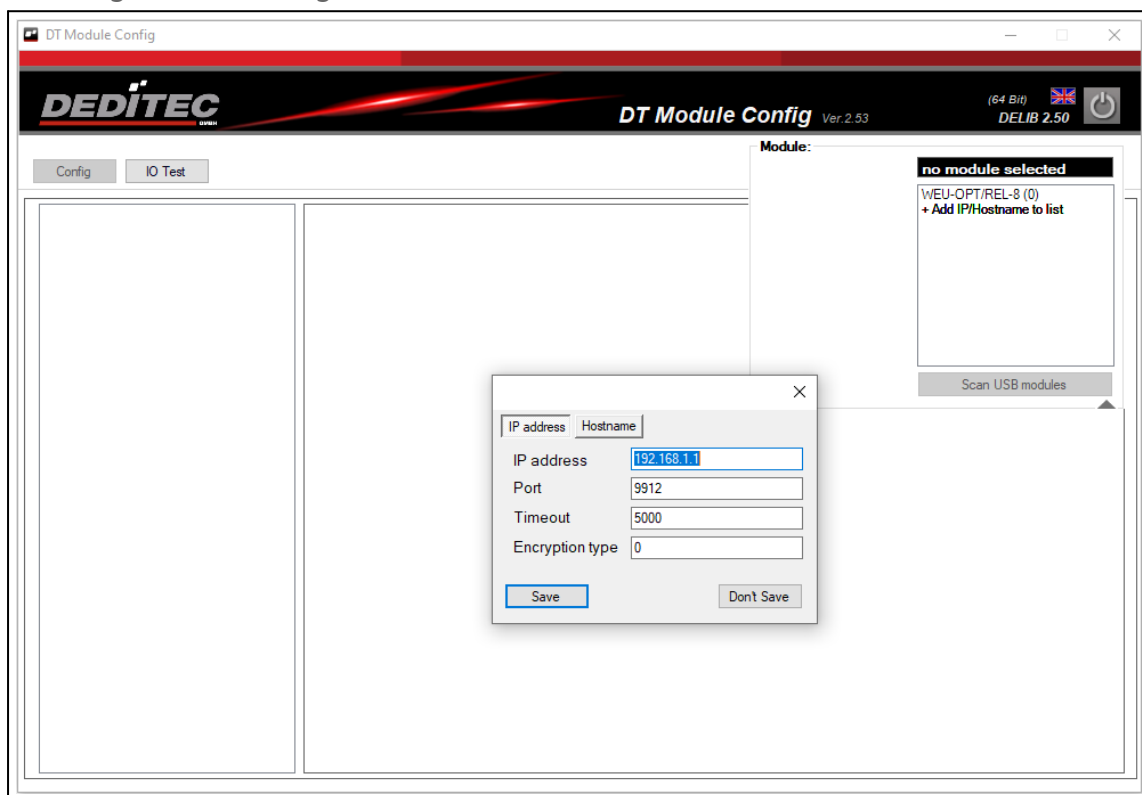


4.2.4.2. via Ethernet

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

Fragen Sie hierfür gegebenenfalls Ihren Systemadministrator.

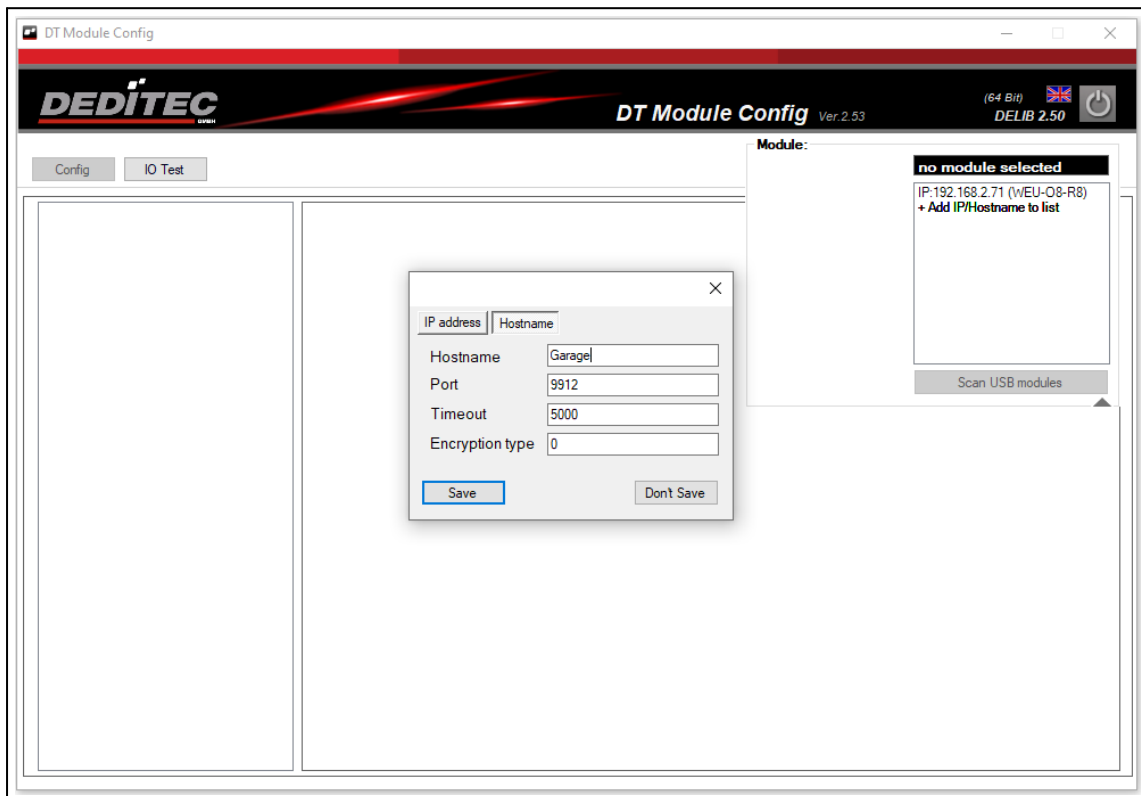
Klicken Sie im Modul Selector auf "Add IP/Hostname to list", tragen Sie dort die automatisch erhaltene IP-Adresse unter dem Reiter "IP Adresse" ein und bestätigen Sie die Eingabe mit "Save".



Ist Ihr Modul im DHCP-Modus (siehe Kapitel: **LAN 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".



4.2.4.3. via WiFi / WPS

WiFi (nur bei WEU-Modulen)

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

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

WPS (nur bei WEU-Modulen)

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

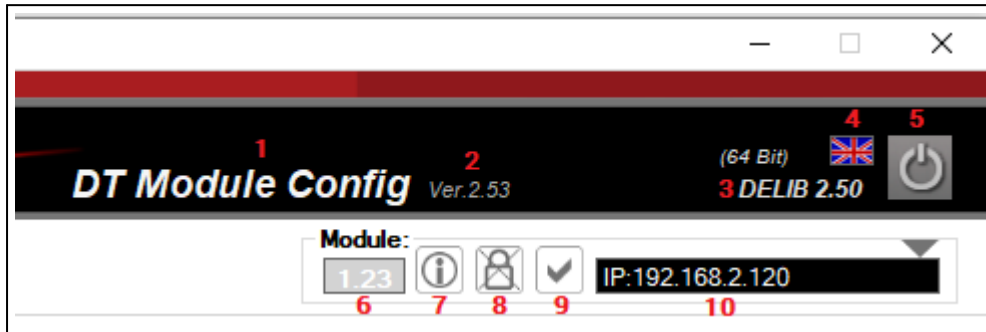
Klicken Sie danach im Modul Selector auf "Add IP/Hostname to list", tragen Sie dort die automatisch erhaltene IP-Adresse unter dem Reiter "IP Adresse" ein und bestätigen Sie die Eingabe mit "Save".

Fragen Sie hierfür gegebenenfalls Ihren Systemadministrator.

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

4.2.4.4. Modul Info

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



Beschreibung:

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

Informationsfenster

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

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

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

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

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

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

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

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

FormConfigModuleOverview (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.

FormConfModulIdent (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-08-R8
LAN status	Static-IP success
DHCP active	☐
IP address	192.168.2.71
Subnet mask	255.255.255.0
Default gateway	192.168.2.254
TCP port	9912

☐ Auto Refresh

FormOfnNetwork LANInfo (Ver. 1.01)

MAC-Adresse

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

Board Name

Zeigt den aktuellen Board Name Ihres Modules an.

LAN-Status

Hier wird der Verbindungsstatus Ihres angeschlossenen Moduls angezeigt.

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

DHCP active

Zeigt an, ob das Modul über DHCP verbunden ist.

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

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

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)	WEU-08-R8
Network configuration protection active 	☐
DHCP active	☐
IP address	192.168.2.71
Subnet mask	255.255.255.0
Default gateway	192.168.2.254
TCP port	9912

Form("ToNetwork1_AbMConfin_Ver. 1.01")

Board Name

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

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

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

Mehr Infos zum Anbinden des Moduls per Board Name

siehe Kapitel: **Benutzung des Modulselectors**

Network configuration protection active

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

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

DHCP active

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

Der Board Name wird als Hostname verwendet.

IP address, Subnet mask, Default gateway und TCP port

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

Werkseinstellungen laden

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

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

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	☒
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

FormToNetworkWiFiInfo (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)	WEU-08-R8_W
WLAN active	☐
Router name	TESTssid
Password	TESTpwd
TCP port	9912 
Restart network	

Form für Netzwerk-WiFi Konfig (Ver. 1.01)

Board name

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

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

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

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

WLAN active

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

Router name

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

Fragen Sie hierfür gegebenenfalls Ihren Systemadministrator.

Password

Hier können Sie das Routerpasswort des verwendeten Routers eingetragen werden.

Fragen Sie hierfür gegebenenfalls Ihren Systemadministrator.

TCP port

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

Werkseinstellungen laden

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

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

starting WPS connection..

Router name

Password

WPS-Start

WPS-Stop

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

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

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

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

☒ Auto Refresh

FormCfgNetworkWiFiWPS (Ver. 1.01)

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	☒
Server	0.de.pool.ntp.org
Port	123
Timezone	(GMT) Greenwich Mean Time: Dublin, Edinburgh

Werkseinstellung laden

Einstellung speichern

Form für Netzwerk 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	☐
Baud rate	625000
RS485 modul address	0
Echo active	☐
Register modus active	☐

FormC3aSenell (Ver 1.01)

Vorzugsmodus (Special mode)

Im Vorzugsmodus wird das Modul automatisch mit folgenden Einstellungen betrieben:

Baud rate: 115200

Modul-Nr. 0

Echo = Off

Register-Mode = On

Baud rate

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

625000

250000

125000

115200

57600

50000

38400

19200

9600

4800

2400

1200

600

300

RS485 Modul-adresse

Adresse für die Identifikation im RS485 Bus.

Echo

Seriell empfangene Zeichen werden vom Modul zurückgesendet.

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	Kanal 0	Ch. 8	Kanal 8
Ch. 1	Kanal 1	Ch. 9	Kanal 9
Ch. 2	Kanal 2	Ch. 10	Kanal 10
Ch. 3	Kanal 3	Ch. 11	Kanal 11
Ch. 4	Kanal 4	Ch. 12	Kanal 12
Ch. 5	Kanal 5	Ch. 13	Kanal 13
Ch. 6	Kanal 6	Ch. 14	Kanal 14
Ch. 7	Kanal 7	Ch. 15	Kanal 15

Set default channel name

Einstellung speichern

FormCfnSubmoduleIOnames (Ver. 1.01)

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

Hinweis:

Der Kanalname darf maximal 16 Zeichen lang sein.

Set default channel name

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

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

FormCfoCANStatus (Ver. 1.00)

CAN-Baudrate

Zeigt die aktuell eingestellte Baud rate Ihres CAN-Interfaces an

DT-CAN-CMD-MODE- Module-Address

DT-CAN-CMD-MODE - Response-Address

CAN is active

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

Use EXT ID

CAN config mode selection

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

Statistik TX/RX

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

DT-CAN-CMD-MODE

Frames total

TX0

RX0

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

☒ Auto Refresh

FormCfoCANStatistic (Ver. 1.00)

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	1 MBit/s
Address Mode	11 Bit Mode
DT-CAN-CMD-MODE - Modul-Address [hex]	FFFFFF
DT-CAN-CMD-MODE - Response-Address [hex]	FFFFFF

Einstellung speichern

FormCfaCANConfiaStd (Ver. 1.00)

Baudrate

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

Address Mode

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

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

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

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

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

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

I/O-Init

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

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

Einstellung speichern

FormCfaCANConfIO (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	☒
Trigger Mode	Interval
Interval	1 * 1ms
Address Mode	11 Bit Mode
Send to CAN-ID [hex]	200
Data to send	OPTO-IN 1-64

Werkseinstellung laden

Einstellung speichern

FormCfCANConfiaTX (Ver. 1.00)

Activate

Aktiviert diesen TX-Mode

Trigger Mode

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

Intervall

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

Address Mode

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

Send to CAN-ID[hex]

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

Data to send

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

Hier können Sie Einstellungen an den RX-Paketen vornehmen.

RX-Mode[1]

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

Activate	☒
Address Mode	11 Bit Mode
Receive at CAN-ID [hex]	100
Data to send	Digital Out 1-64

Werkseinstellung laden

Einstellung speichern

FormCfoCANConfioRX (Ver. 1.00)

Activate

Aktiviert diesen RX-Mode

Address Mode

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

Receive at CAN-ID[hex]

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

Data to send

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

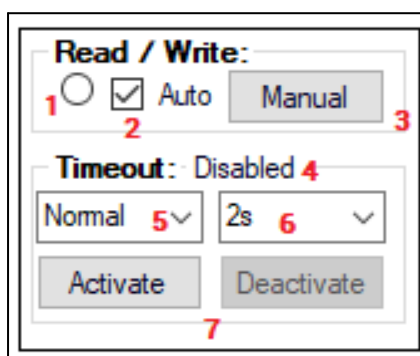
4.2.5.2. I/O-Test

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
Kanal 0	☐	0	☐
Kanal 1	☐	0	☐
Kanal 2	☐	0	☐
Kanal 3	☐	0	☐
Kanal 4	☐	0	☐
Kanal 5	☐	0	☐
Kanal 6	☐	0	☐
Kanal 7	☐	0	☐

	State on/off	Counter	FlipFlop
Kanal 8	☐	0	☐
Kanal 9	☐	0	☐
Kanal 10	☐	0	☐
Kanal 11	☐	0	☐
Kanal 12	☐	0	☐
Kanal 13	☐	0	☐
Kanal 14	☐	0	☐
Kanal 15	☐	0	☐

☐ 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	on off	☐	0 s set		Kanal 8	on off	0 s set
Kanal 1	on off	☐	0 s set		Kanal 9	on off	0 s set
Kanal 2	on off	☐	0 s set		Kanal 10	on off	0 s set
Kanal 3	on off	☐	0 s set		Kanal 11	on off	0 s set
Kanal 4	on off	☐	0 s set		Kanal 12	on off	0 s set
Kanal 5	on off	☐	0 s set		Kanal 13	on off	0 s set
Kanal 6	on off	☐	0 s set		Kanal 14	on off	0 s set
Kanal 7	on off	☐	0 s set		Kanal 15	on off	0 s 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	0,027 V	Kanal 8	0,027 V
Kanal 1	0,026 V	Kanal 9	0,025 V
Kanal 2	0,025 V	Kanal 10	0,024 V
Kanal 3	0,025 V	Kanal 11	0,025 V
Kanal 4	0,025 V	Kanal 12	0,024 V
Kanal 5	0,025 V	Kanal 13	0,024 V
Kanal 6	0,025 V	Kanal 14	0,025 V
Kanal 7	0,025 V	Kanal 15	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
Kanal 0	0	16 Bit / 0-10V ▾	0 V
Kanal 1	1	16 Bit / 0-10V ▾	0,999908 V
Kanal 2	2	16 Bit / 0-10V ▾	1,999969 V
Kanal 3	3	16 Bit / 0-10V ▾	2,999878 V
Kanal 4	4	16 Bit / 0-10V ▾	3,999939 V
Kanal 5	5	16 Bit / 0-10V ▾	5 V
Kanal 6	6	16 Bit / 0-10V ▾	5,999908 V
Kanal 7	7	16 Bit / 0-10V ▾	6,999969 V

	Value	Mode	Readback
Kanal 8	8	16 Bit / 0-10V ▾	7,999878 V
Kanal 9	9	16 Bit / 0-10V ▾	8,999939 V
Kanal 10	10	16 Bit / 0-10V ▾	9,999847 V
Kanal 11	11	16 Bit / 0-10V ▾	9,999847 V
Kanal 12	12	16 Bit / 0-10V ▾	9,999847 V
Kanal 13	13	16 Bit / 0-10V ▾	9,999847 V

Set mode for all channels

16 Bit / 0-10V ▾

☒ 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.

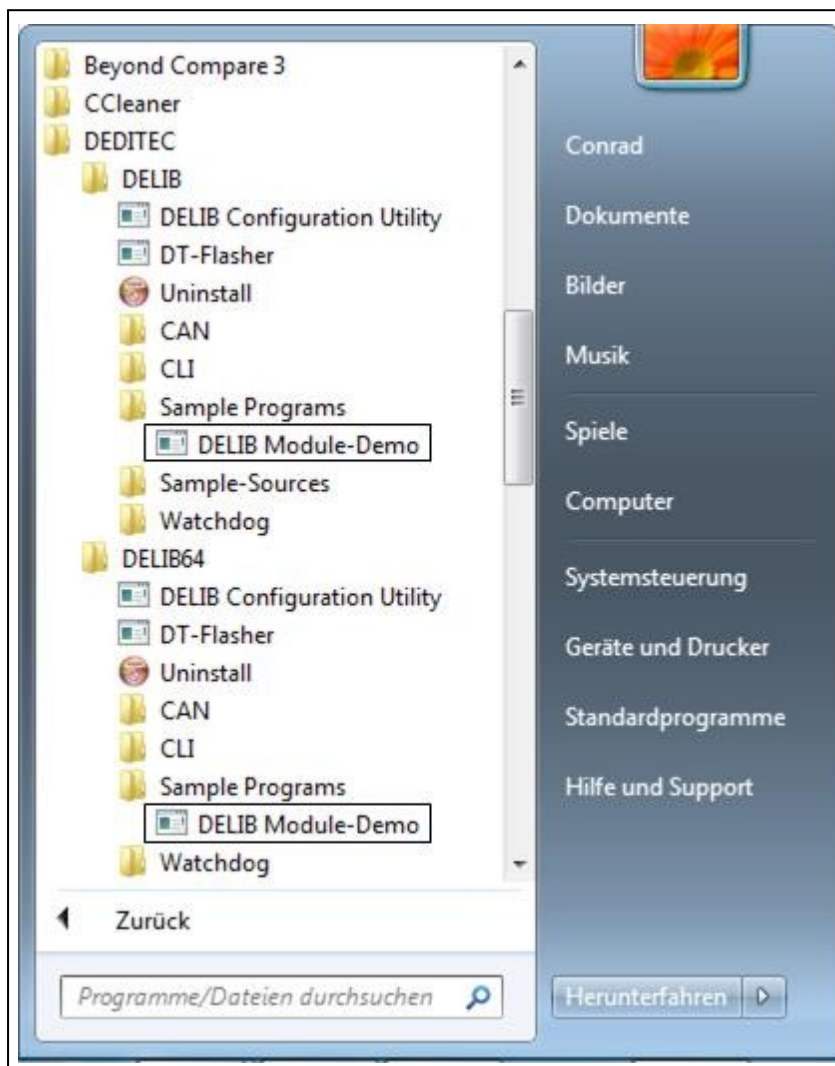
Set mode for all channels

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.

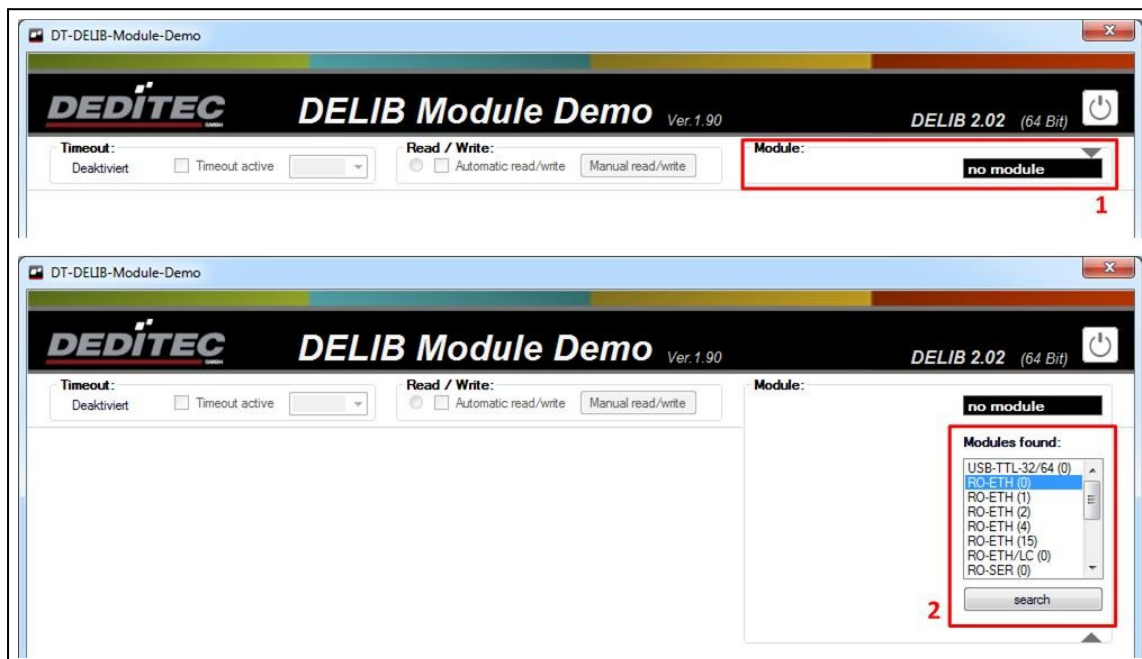


Das Programm DELIB Module Demo ist ein All-in-One Tool mit dem sämtliche I/Os aller Produkte aus unserem S&R Bereich gesteuert und getestet werden können.

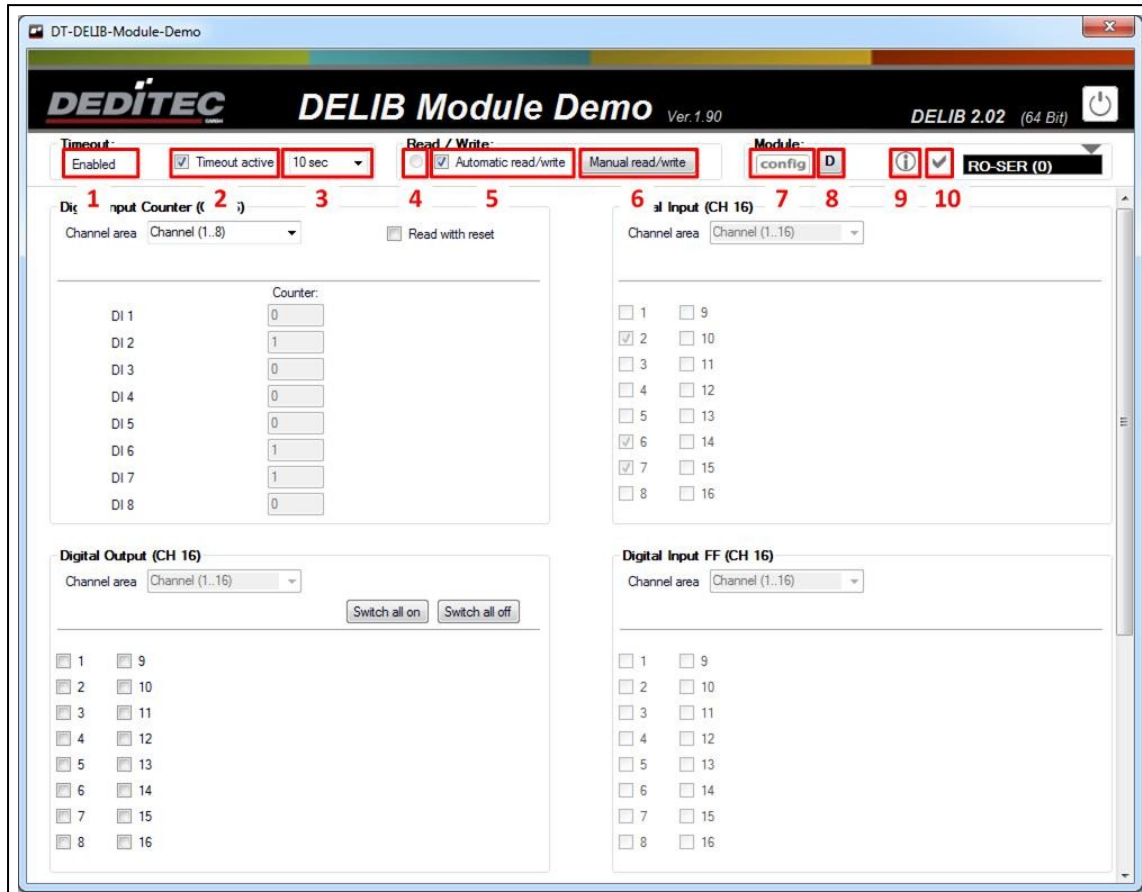
4.2.6.1. Auswahl des Moduls

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.



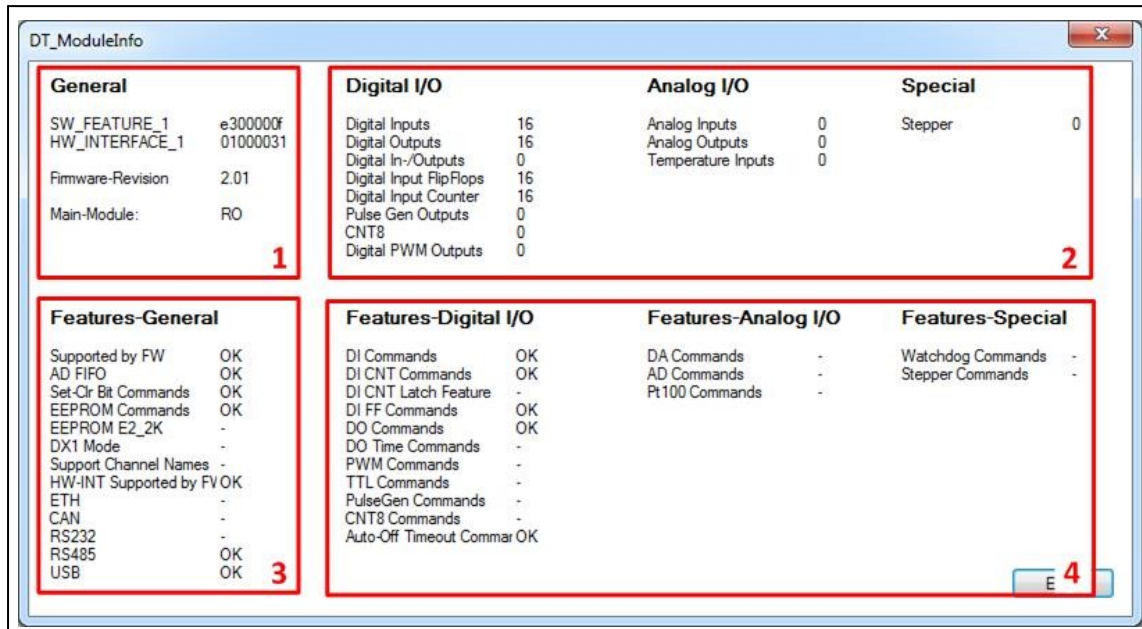
4.2.6.2. Allgemein



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.

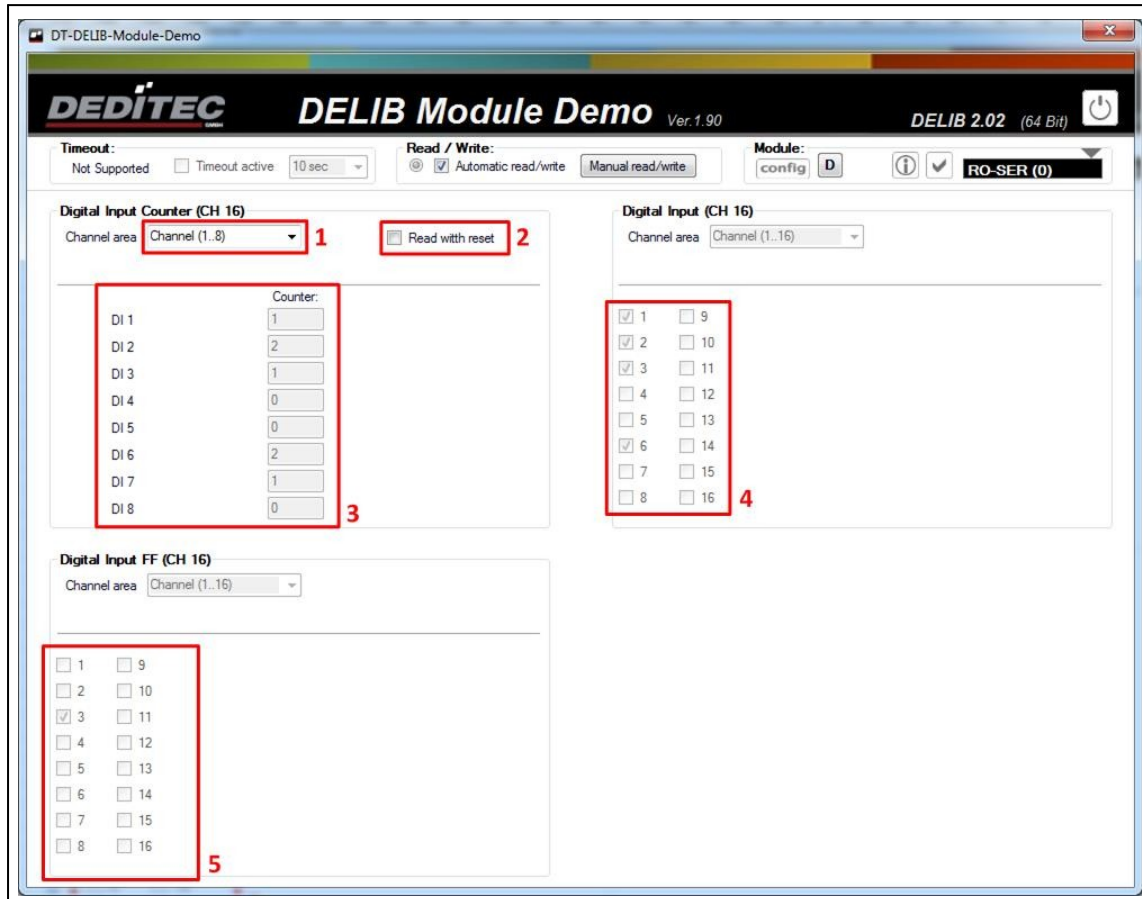
4.2.6.2.1. Module Info



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.

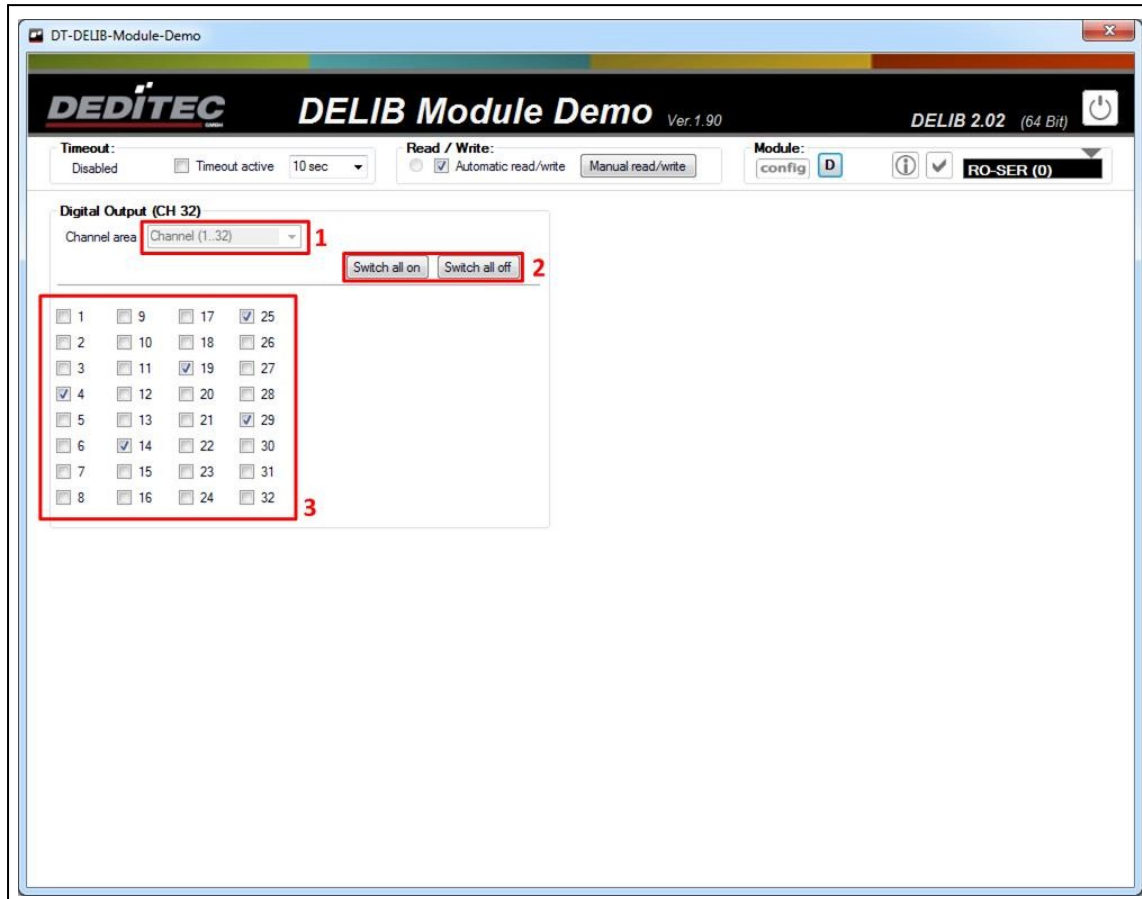
4.2.6.3. Digital Input



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).

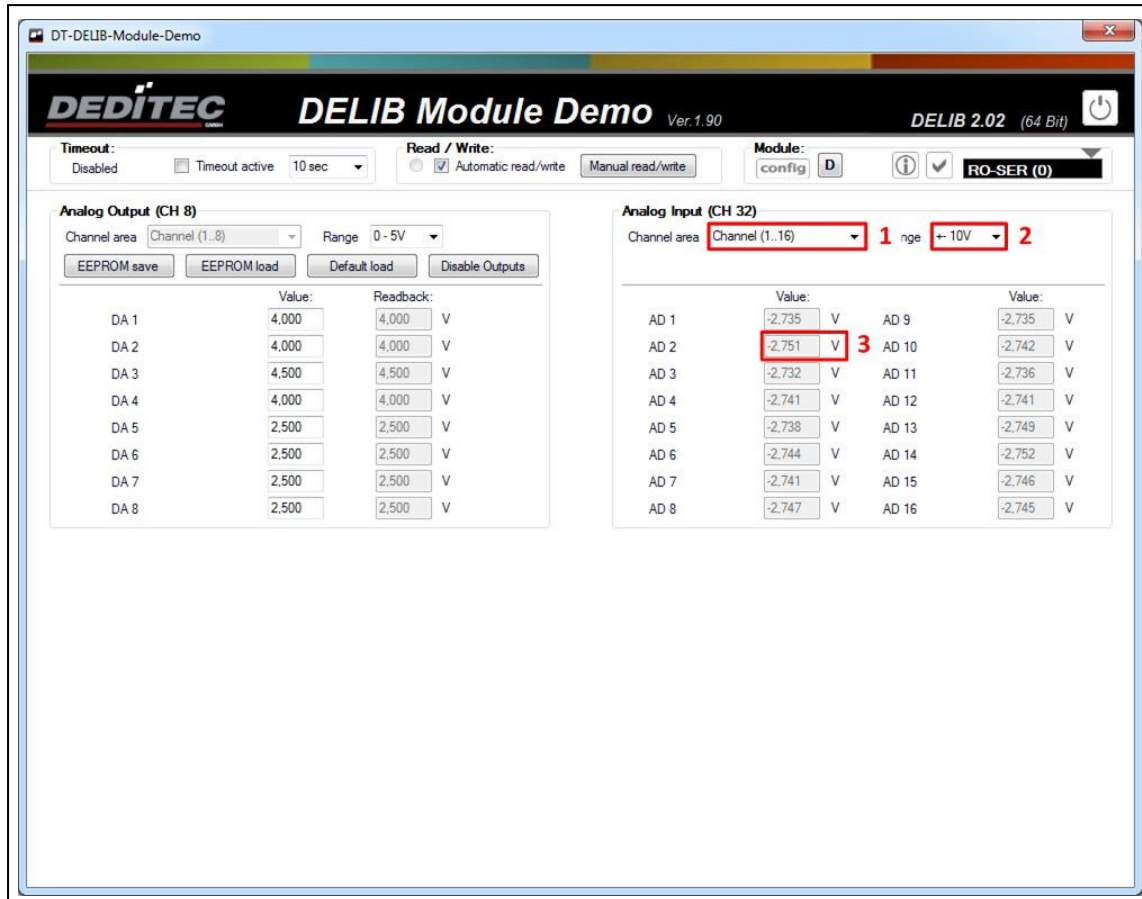
4.2.6.4. Digital Output



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.

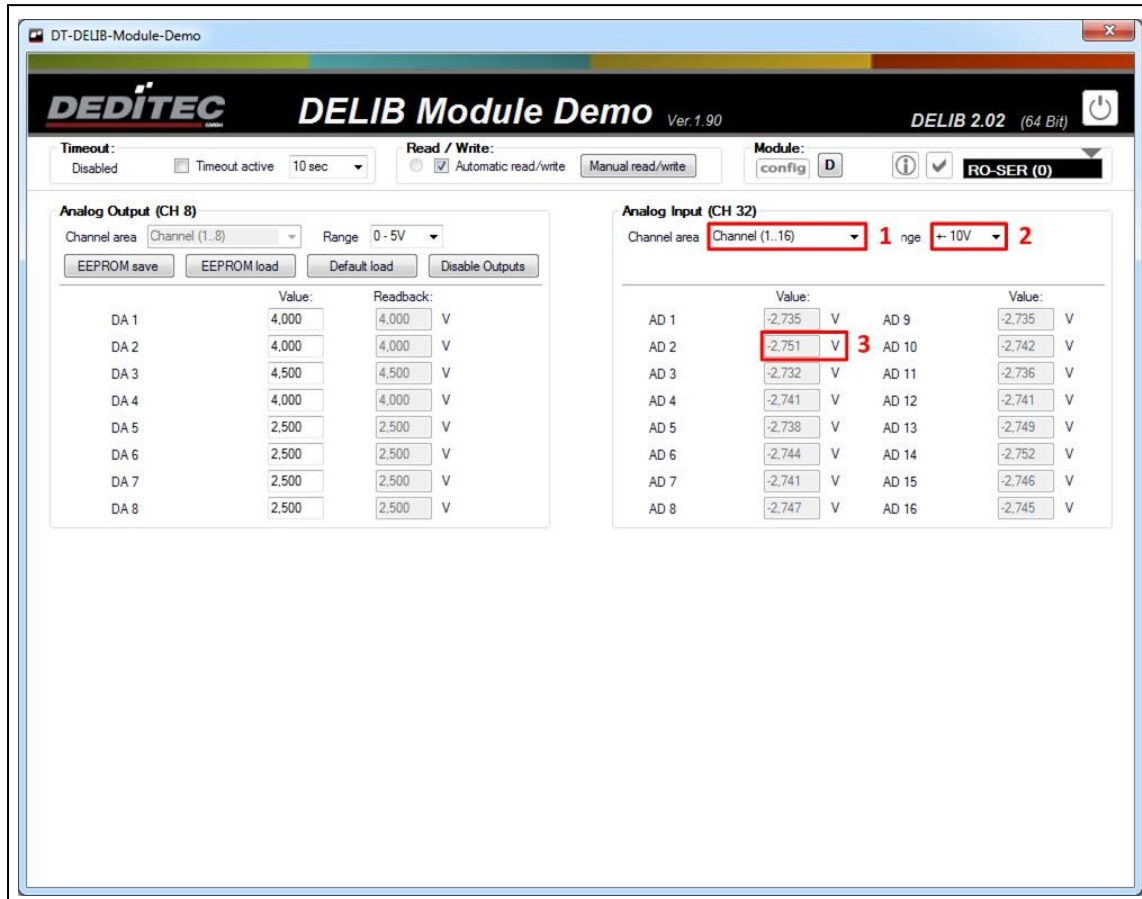
4.2.6.5. Analog Input



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



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.

4.2.7. 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:

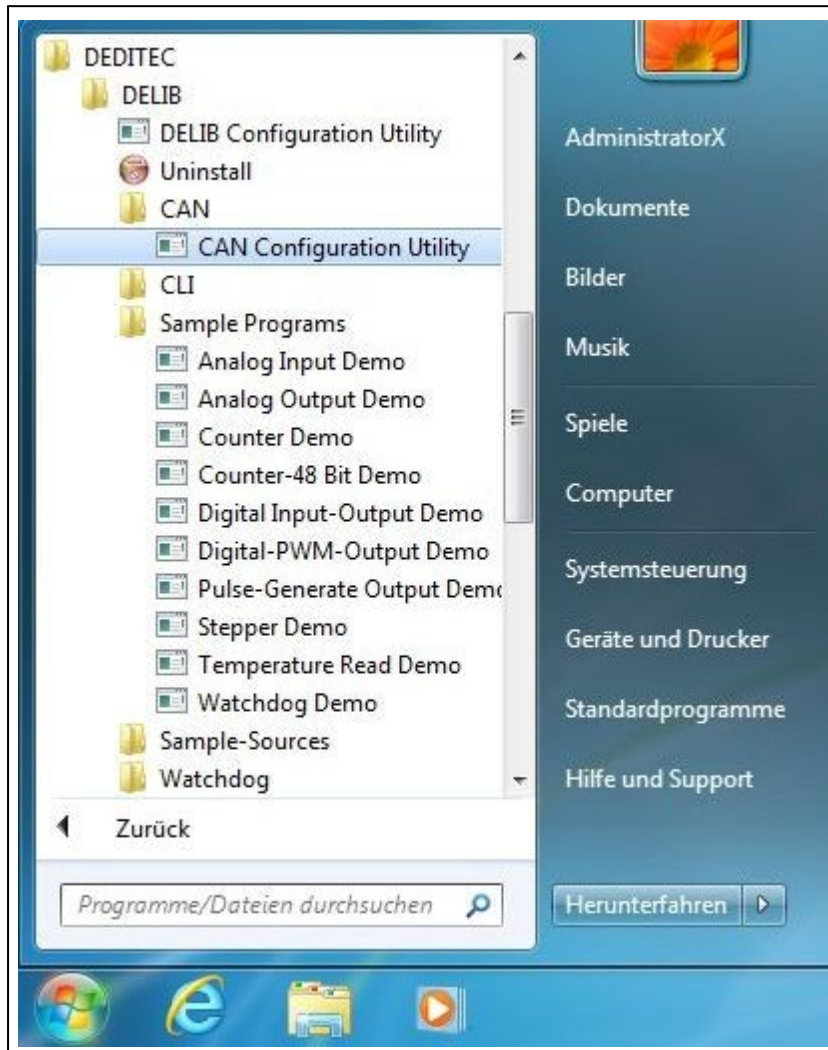
- Baudrate
- Adress-Modus
- Modul-ID
- Response-ID
- Modi, mit denen Submodule gestartet werden
- Automatischer Sendemodus (TX-Modus)
- Automatischer Empfangsmodus (RX-Modus)

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

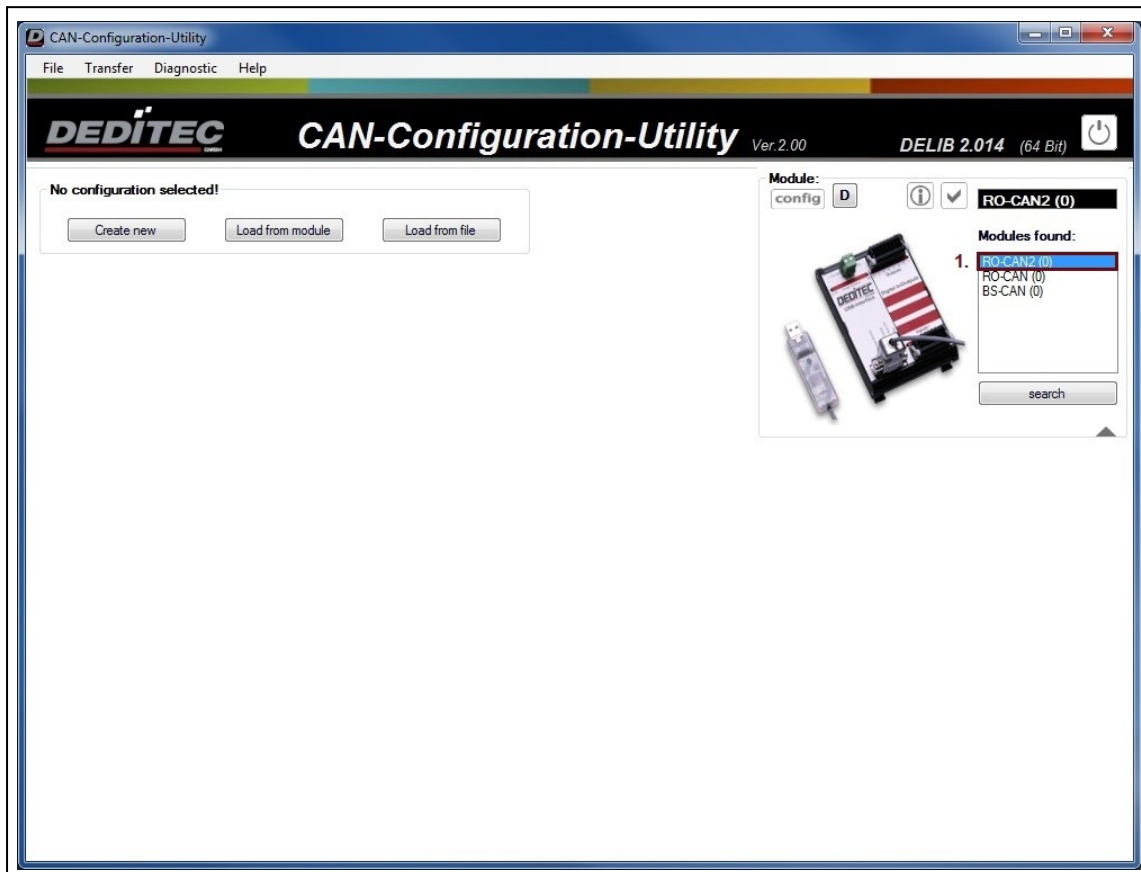
4.2.7.1. Auswahl des Moduls

Starten Sie das CAN-Configuration-Utility über:

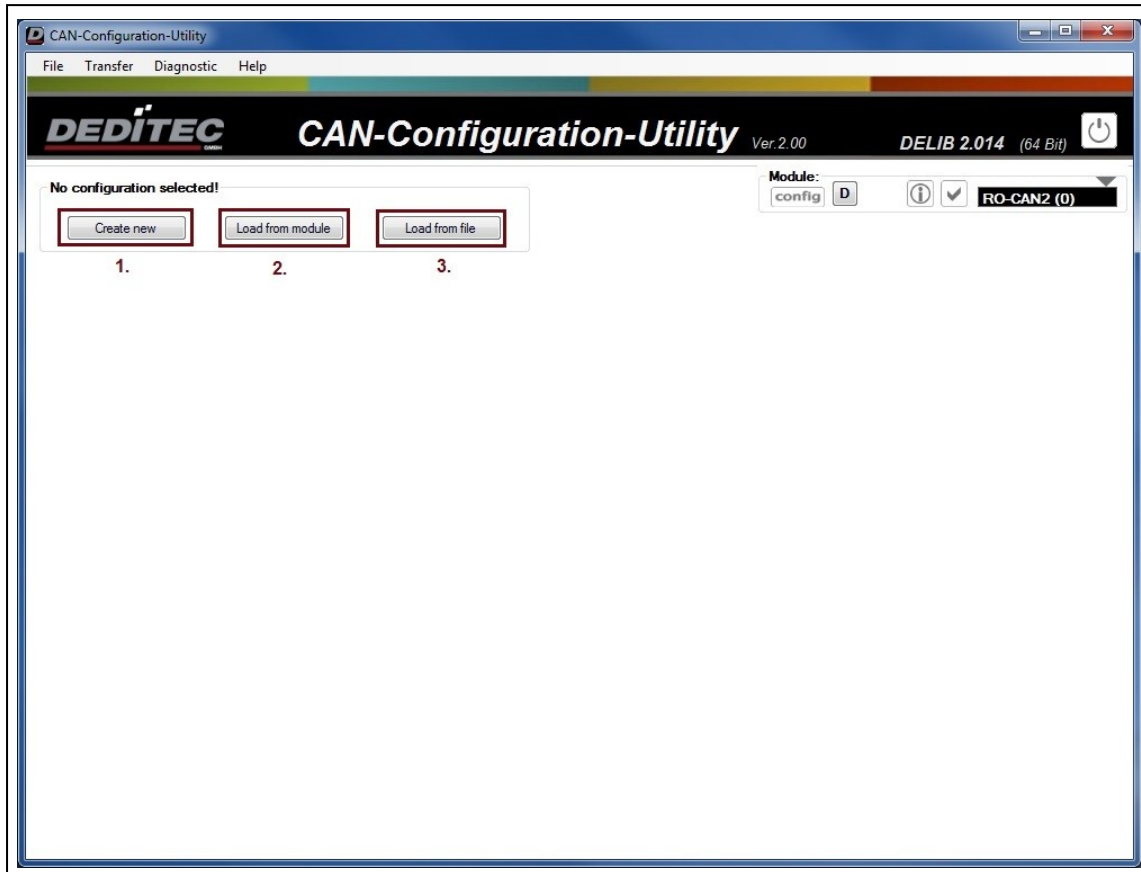
Start → Programme → DEDITEC → DELIB → CAN



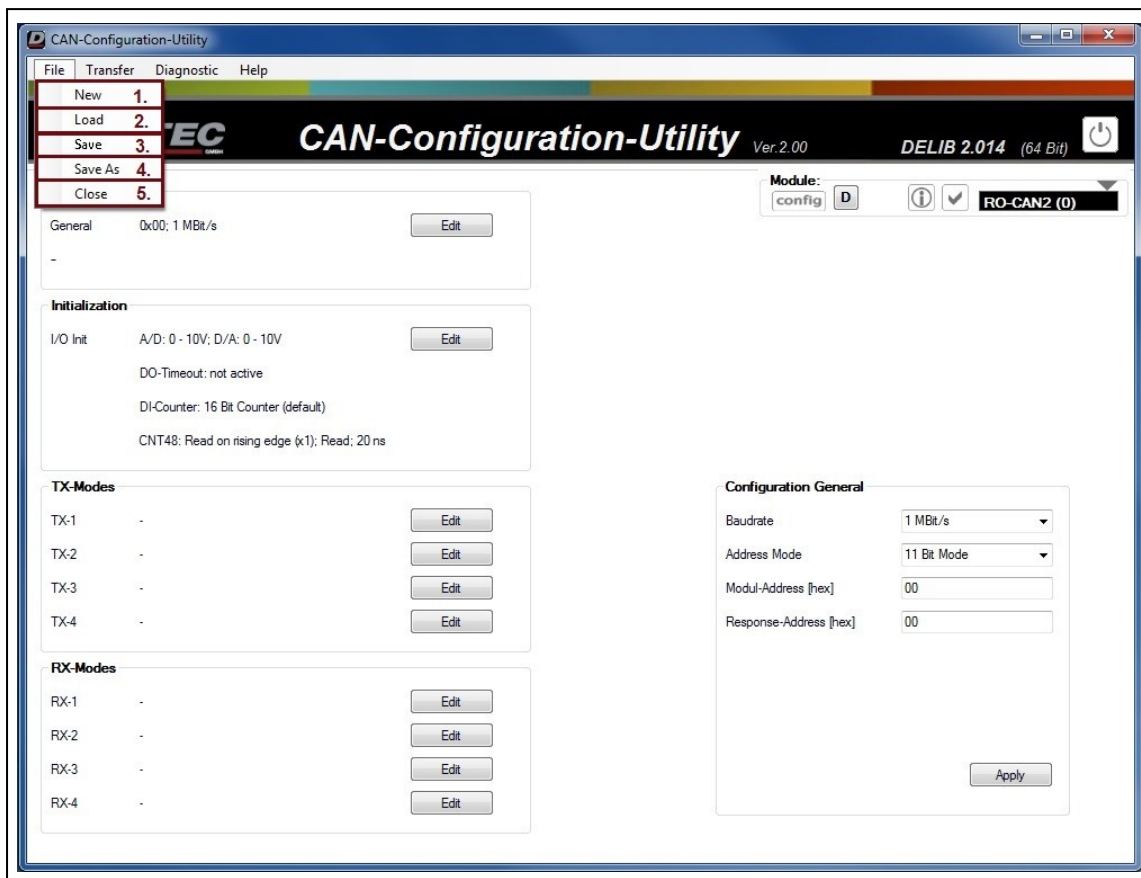
1. In der "Module-Selection" ein entsprechendes CAN-Modul (z.B. das RO-CAN2) auswählen.



4.2.7.2. Neue Konfiguration Erstellen, Laden, Speichern



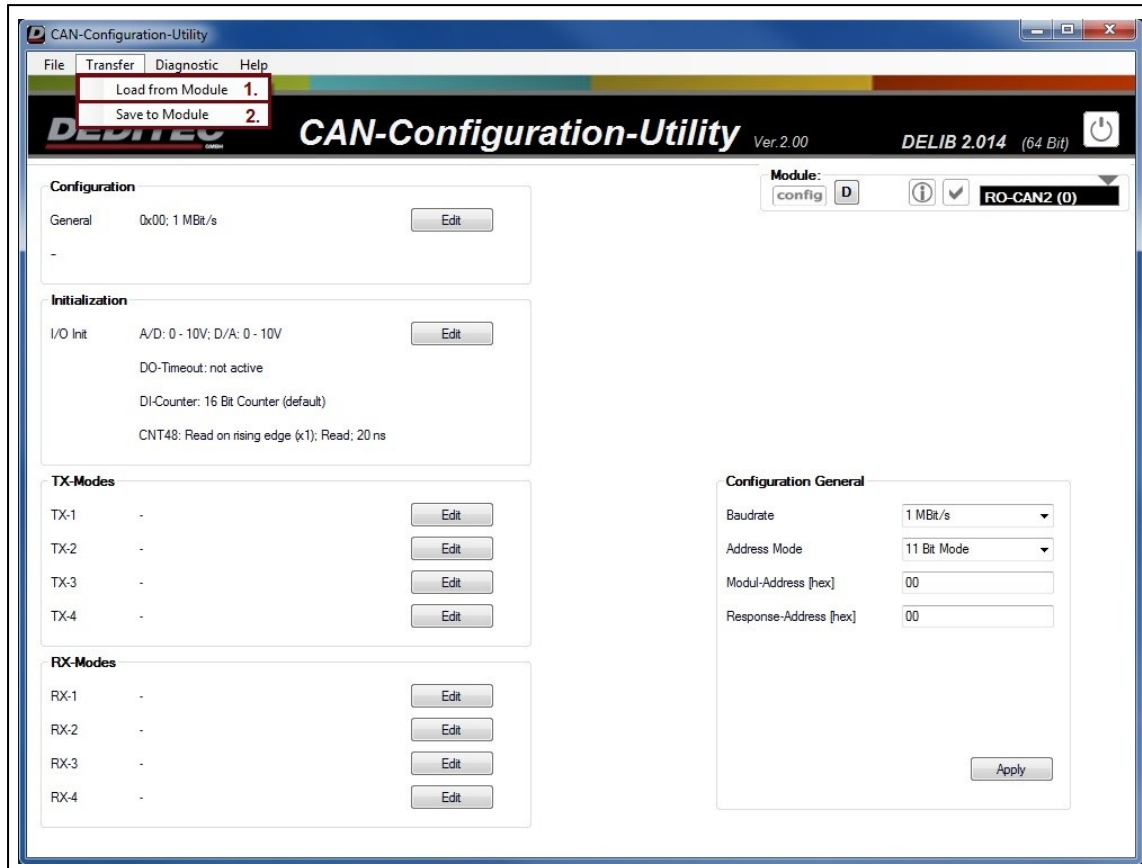
1. Über "Create new" kann eine neue CAN-Konfiguration erstellt werden.
2. Mit "Load from Module" kann die aktuell auf dem Modul vorhandene Konfiguration ausgelesen werden.
3. "Load from File" erlaubt es, zuvor auf einem Datenträger gespeicherte CAN-Konfigurationsdateien zu öffnen.



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

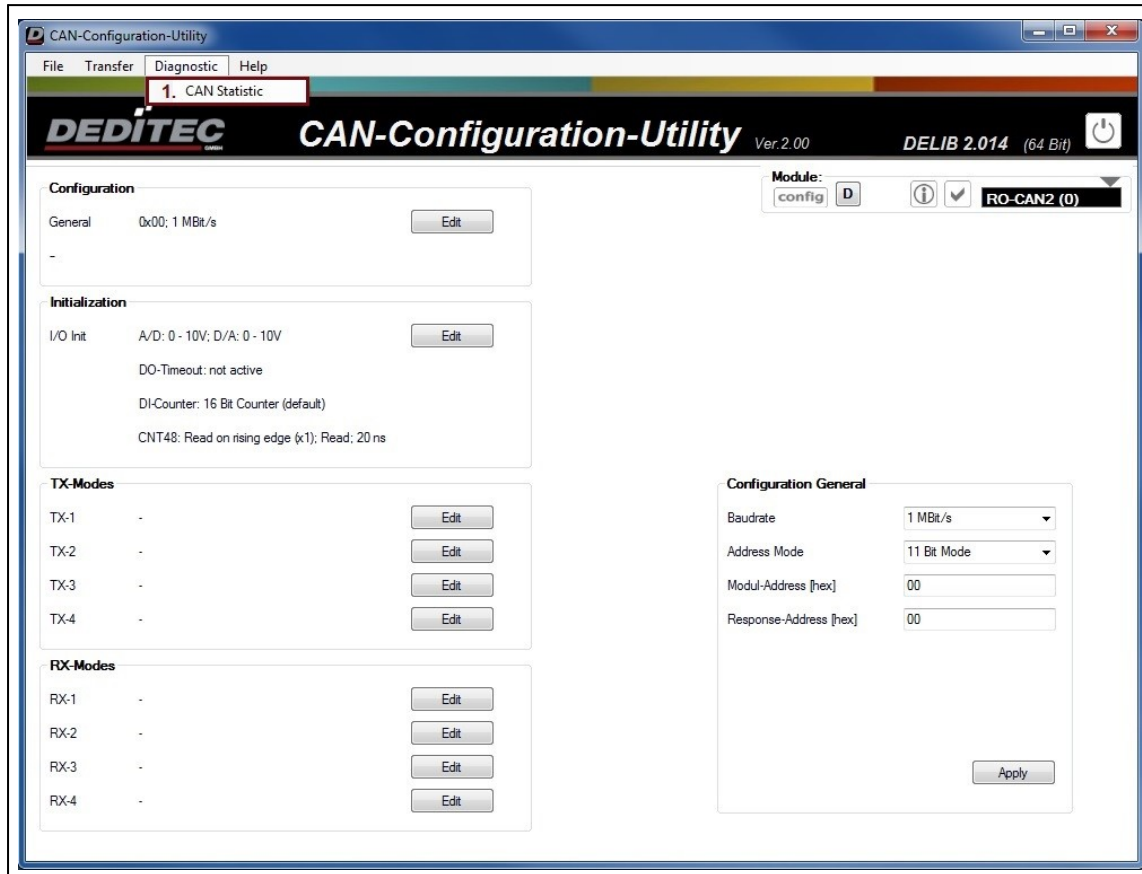
1. Über den Menüpunkt "New" kann eine neue CAN-Konfiguration erstellt werden.
2. "Load" bietet die Möglichkeit, zuvor gespeicherte CAN-Konfigurationsdateien zu öffnen.
3. Wurde eine CAN-Konfigurationsdatei geöffnet, so kann über "Save" die Datei überschrieben und aktuelle Änderungen gespeichert werden.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



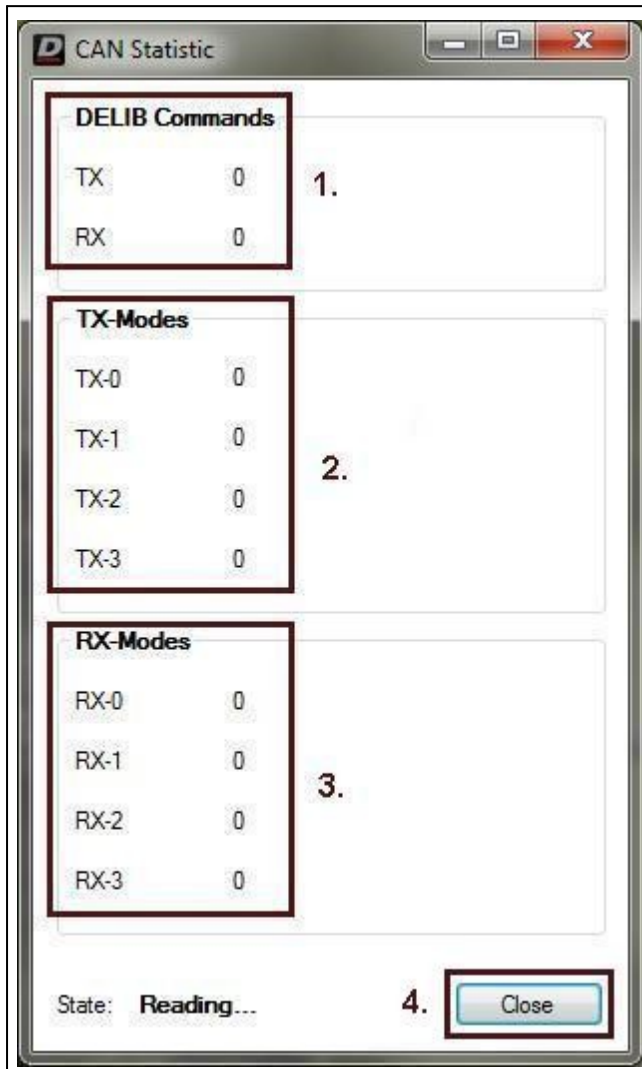
1. Über "Load from Module" kann die aktuelle Konfiguration des ausgewählten Moduls ausgelesen werden.
2. "Save to Module" überträgt die aktuelle CAN-Konfiguration auf das ausgewählte Modul.

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.

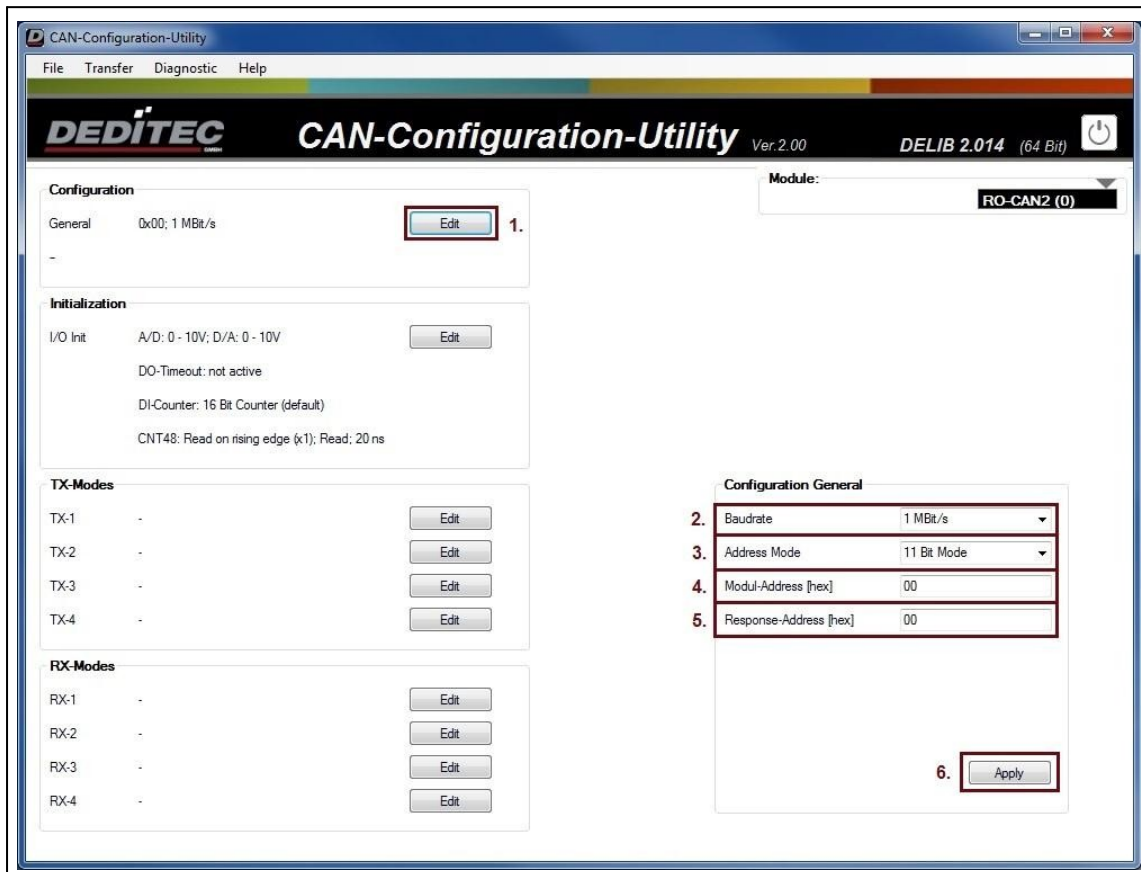
4.2.7.5. Konfiguration

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.

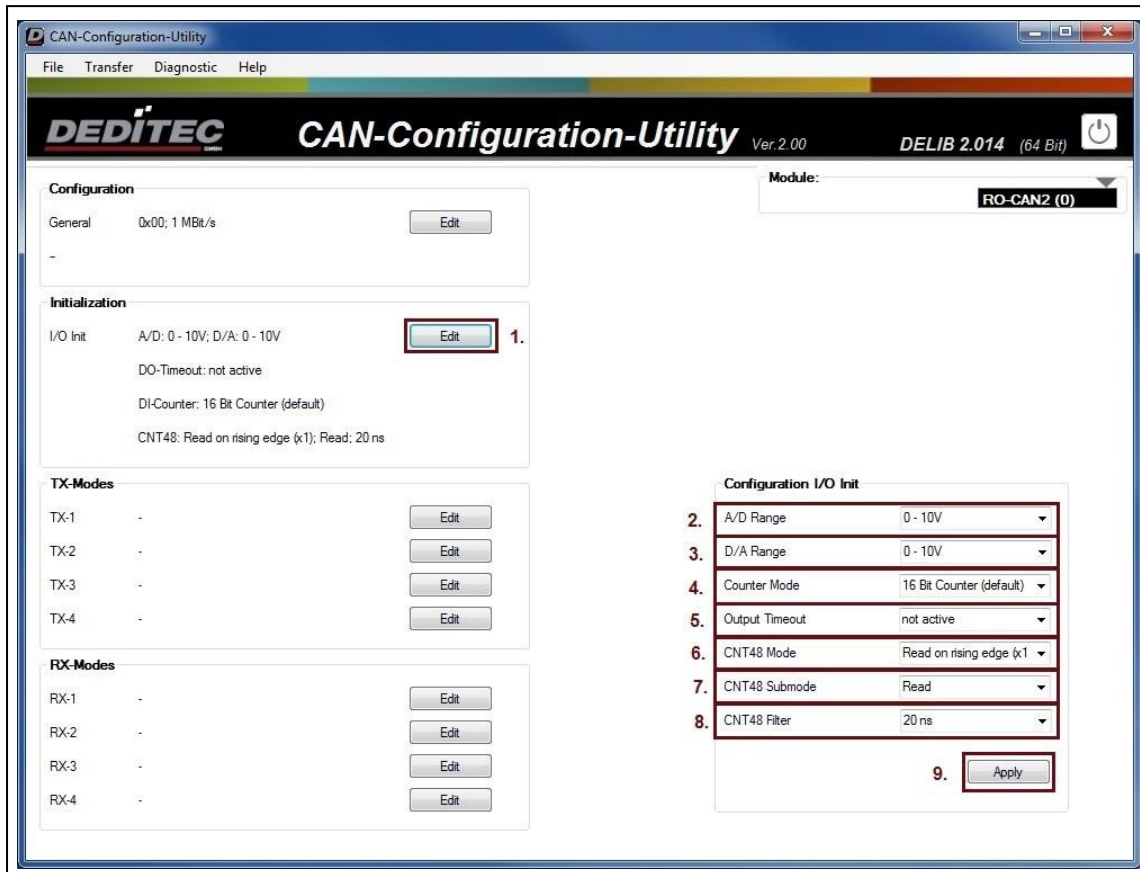
Auf den nachfolgenden Seiten wird gezeigt, wie die verschiedenen Modi konfiguriert werden.

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.

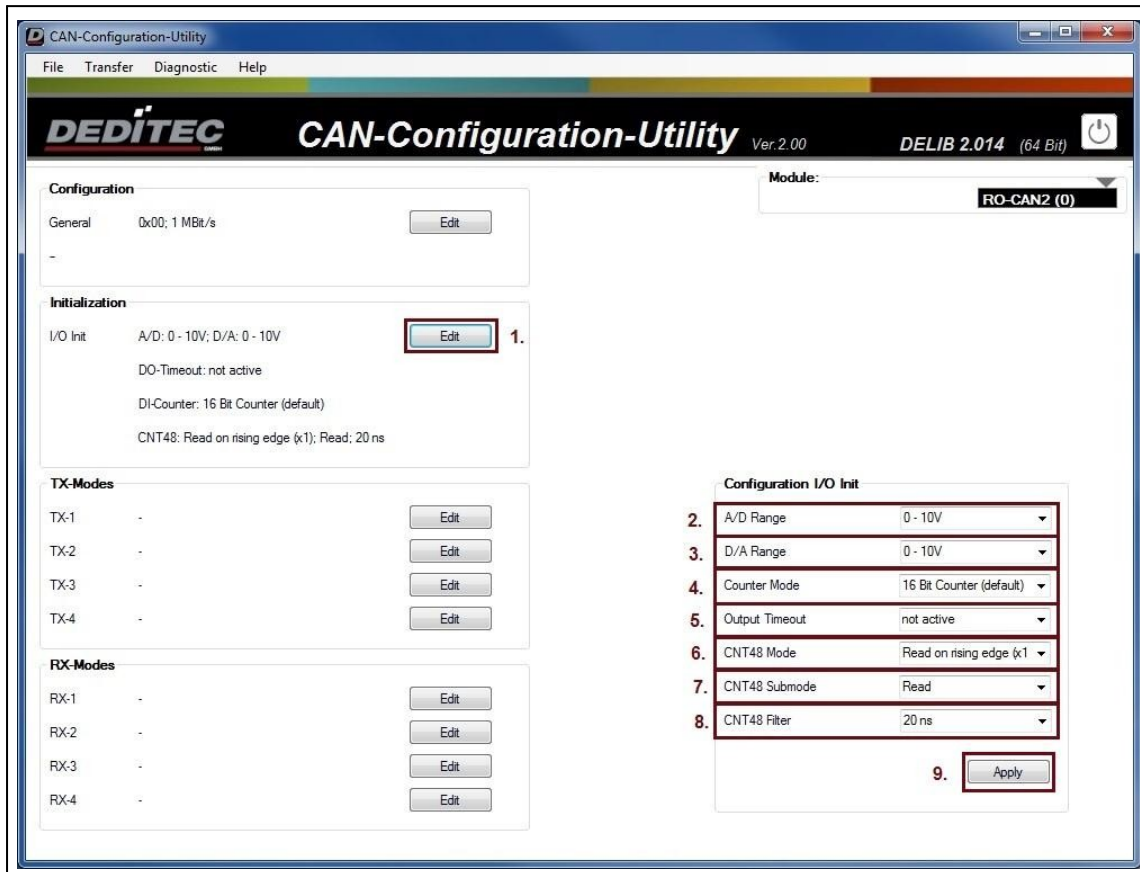
Hinweis:

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

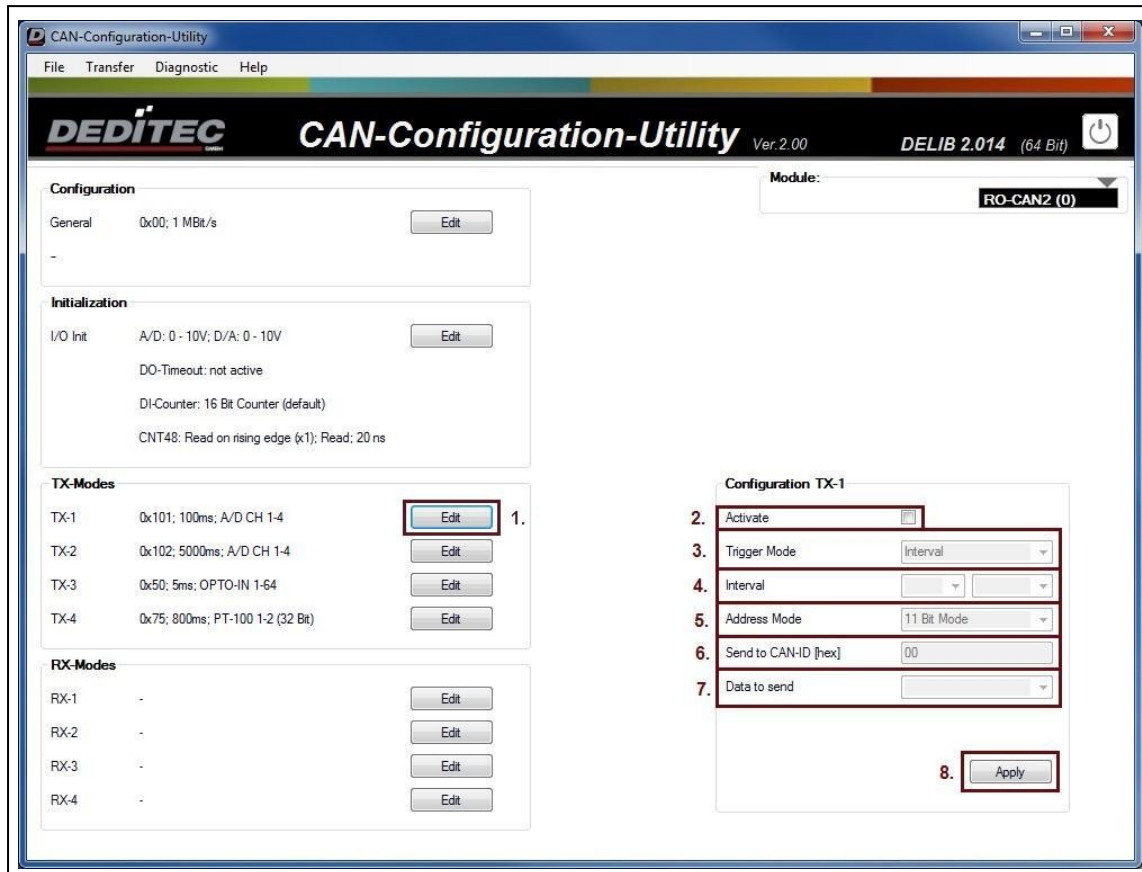
mit 16-Bit hochgezählt werden, oder mit je 8-Bit hoch- und heruntergezählt werden.

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	☒
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

☒

Address Mode

11 Bit Mode

▼

Receive at CAN-ID [hex]

200

Data to send

Trigger Auto TX 1

▼

Apply

Configuration TX-1

Activate

☒

Trigger Mode

RX-Event

▼

Interval

▼

▼

Address Mode

11 Bit Mode

▼

Send to CAN-ID [hex]

100

Data to send

A/D CH 1-4

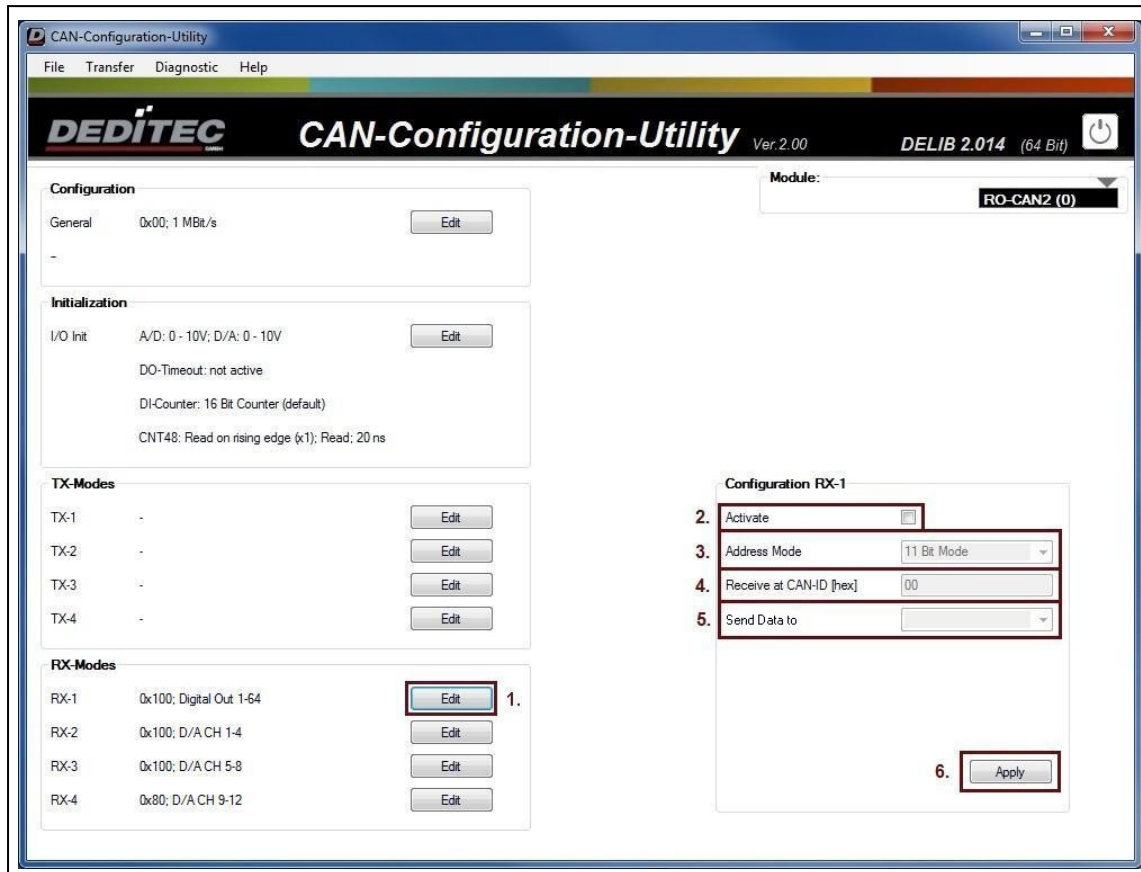
▼

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	☒
Address Mode	11 Bit Mode
Receive at CAN-ID [hex]	201
Data to send	D/A CH 5-8

Apply

Das Beispiel beinhaltet folgende Einstellungen:

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

4.2.7.5.4.2. Beispiel RX-DO

Configuration RX-1

Activate ☒

Address Mode 11 Bit Mode

Receive at CAN-ID [hex] 100

Data to send Digital Out 1-64

Apply

Das Beispiel beinhaltet folgende Einstellungen:

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

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 $\pm 10\text{V}$

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

Beispiel zur Berechnung:

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

Wertebereich = +/-10 V

Berechnung:

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

Beispiel Spannungsbereich 0-5V

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

Beispiel zur Berechnung:

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

Wertebereich = 0-5 V

Berechnung:

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

Beispiel Strombereich

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

Beispiel zur Berechnung:

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

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

Berechnung:

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

Hinweis:

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

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_STEPPER_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_EEPR OM_SAVE	16 (hex)	Speichern der Motorcharakteristik im EEPROM
MOTORCHARACTERISTIC_EEPR OM_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

4.2.7.6.8.4. Beispiel

Programmbeispiel

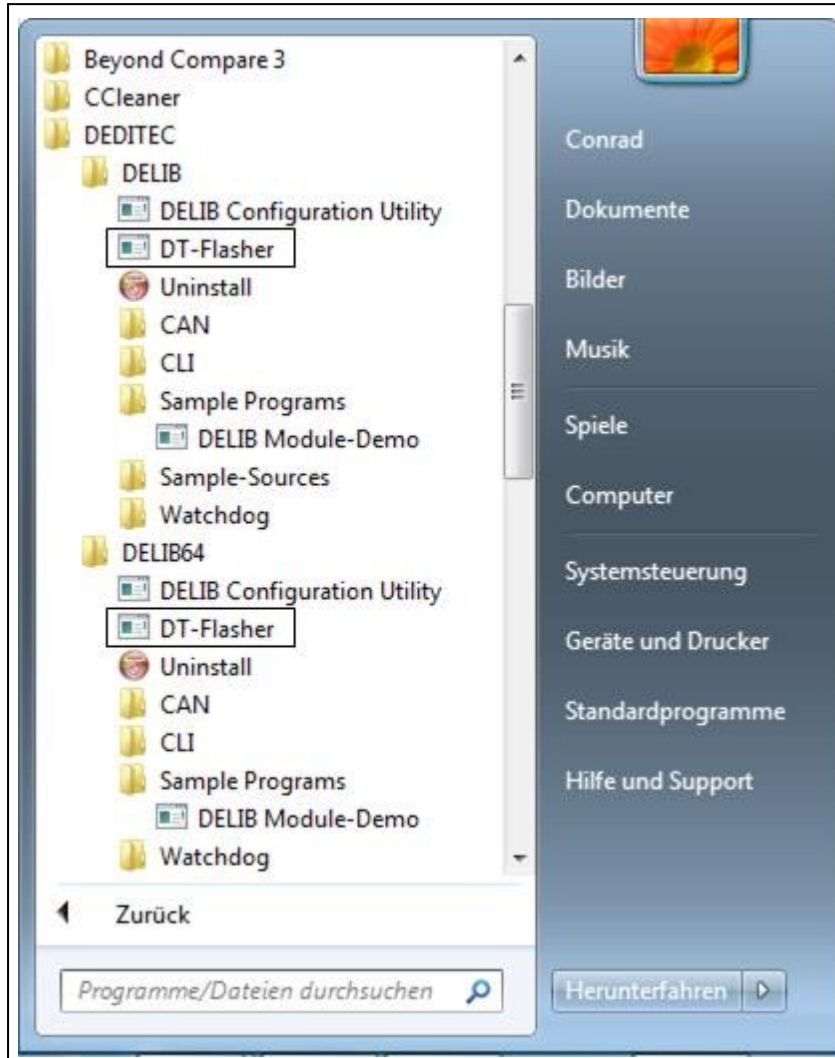
```
DapiStepperCommand(handle, 1,  
DAPI_STEPPER_CMD_GO_POSITION, 3200, 0, 0, 0);
```

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.

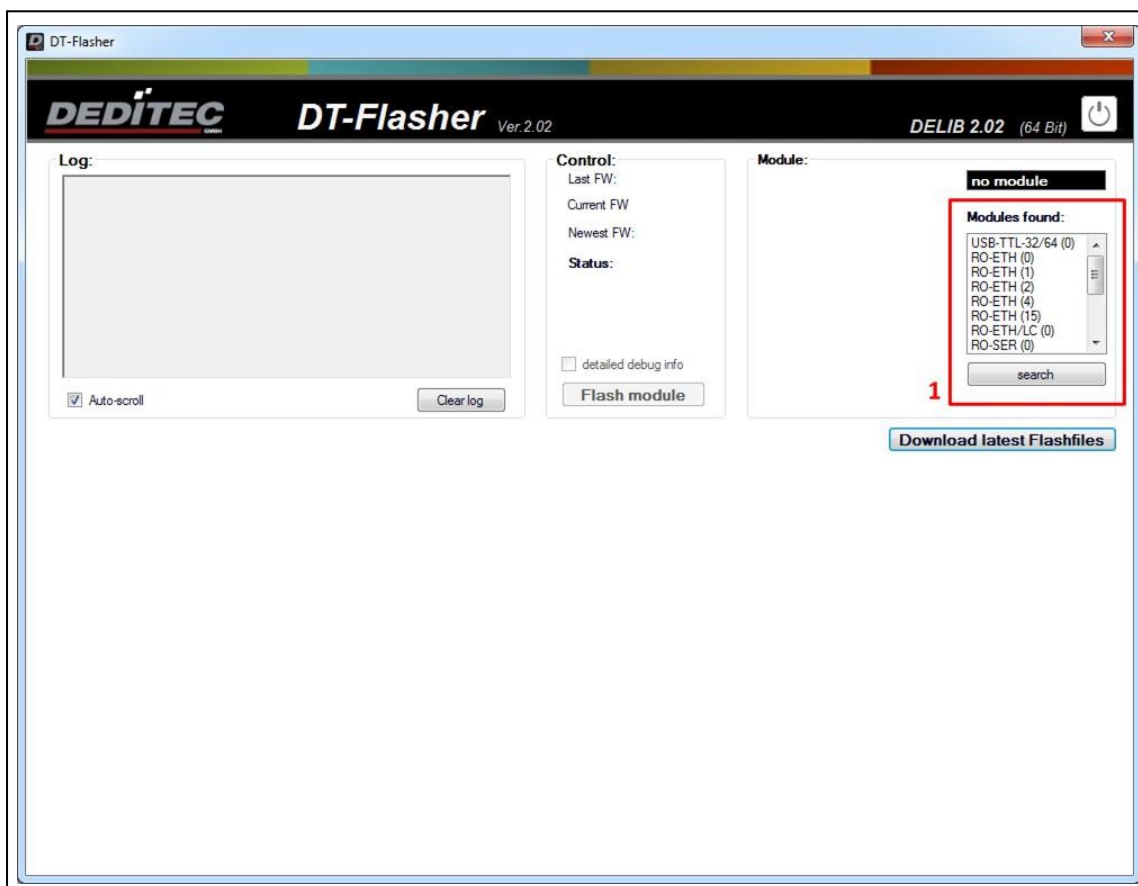
4.2.8.1. Über DEDITEC-Firmware

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:

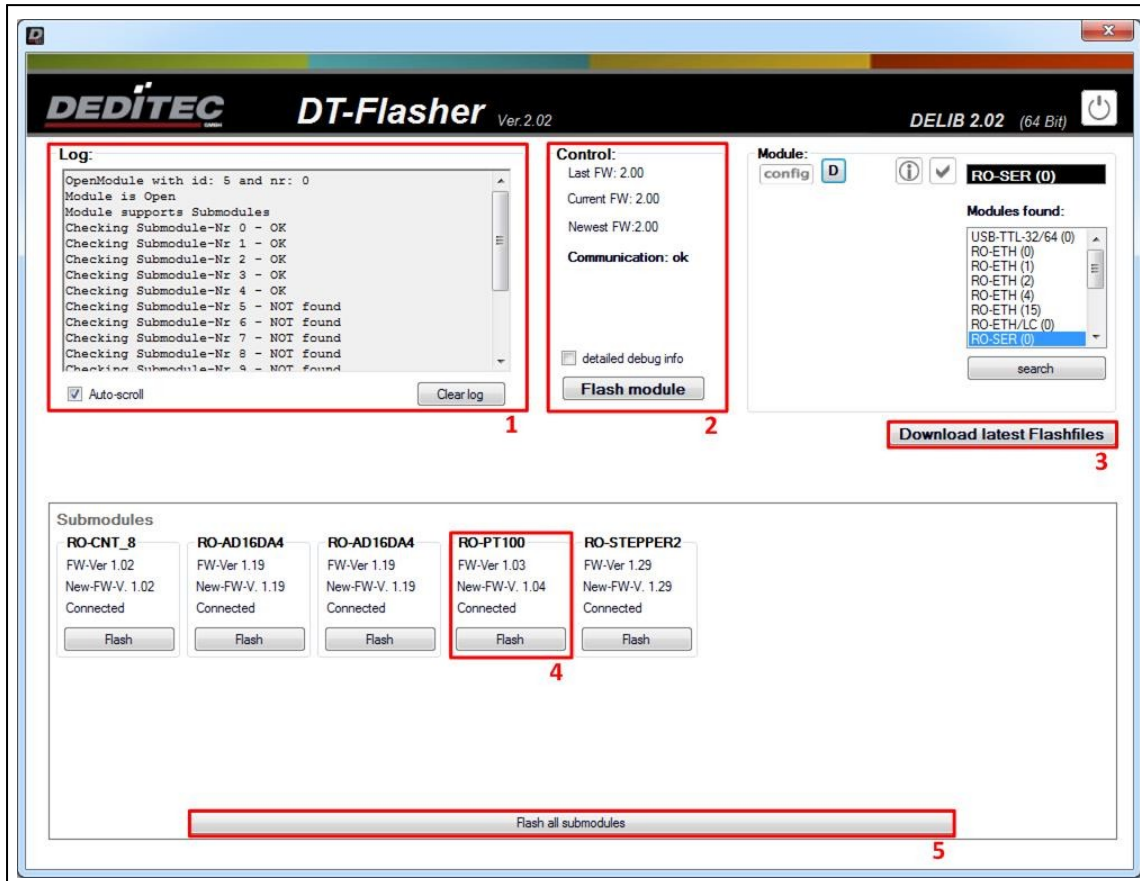
Da neue Firmware-Versionen in der Regel neue Funktionen für Ihr Produkt "freischalten", empfehlen wir daher ein regelmäßiges Firmware-Update Ihrer DEDITEC Produkte.

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:

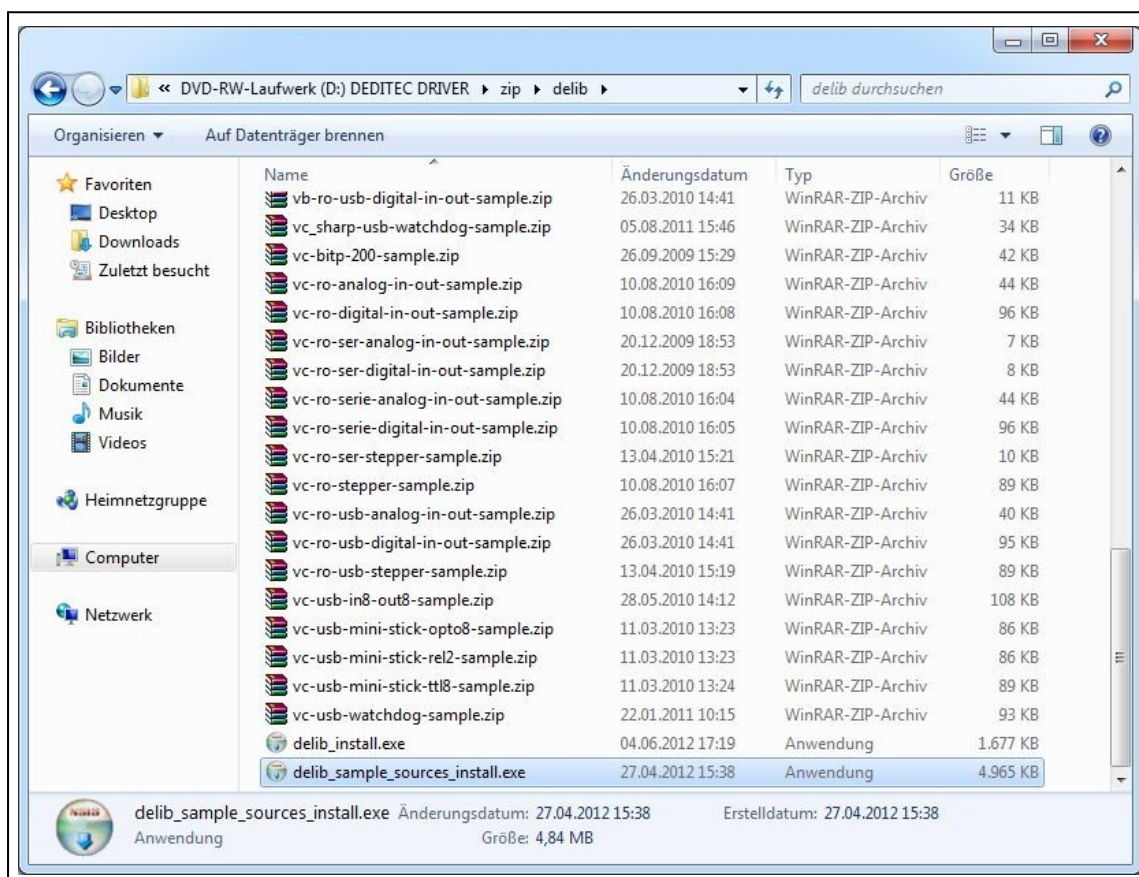
- C
- C++
- C#
- Delphi
- VisualBasic
- VB.NET
- MS-Office
- LabVIEW
- Java

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`.

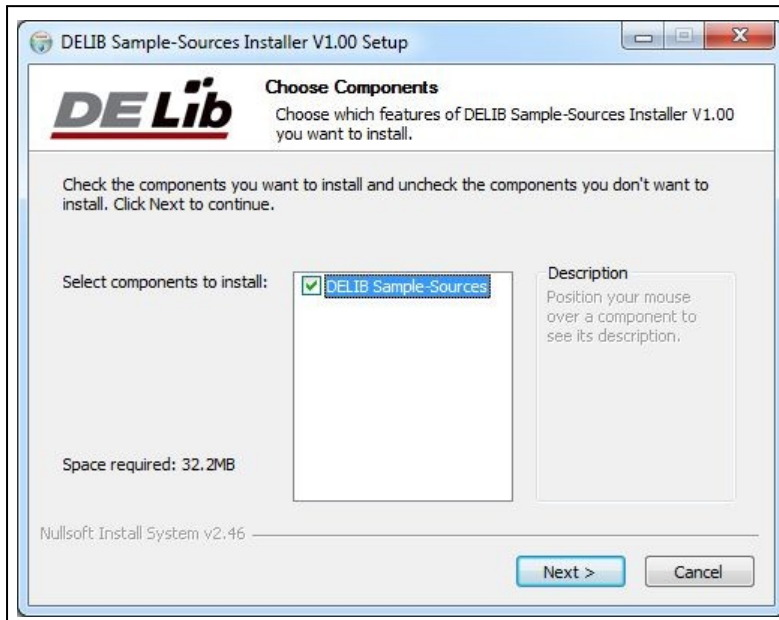
Eine aktuelle Version der Sample Sources finden Sie auch im Internet unter <http://www.deditec.de/de/delib>



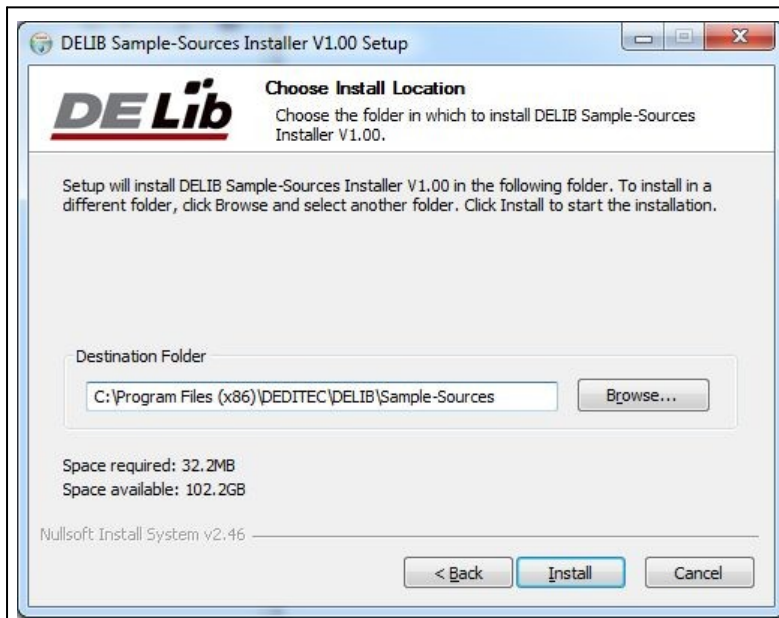
Startbild des DELIB Sample Sources Installer



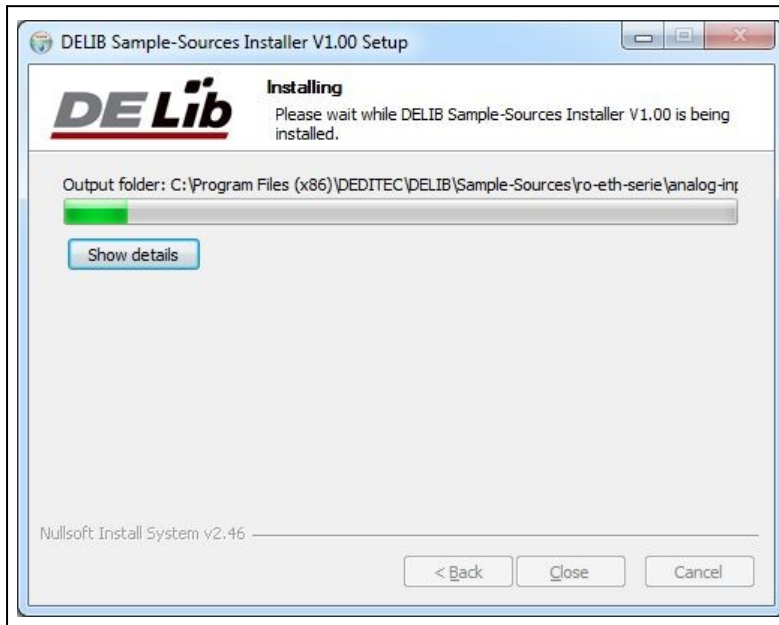
Drücken Sie Next.



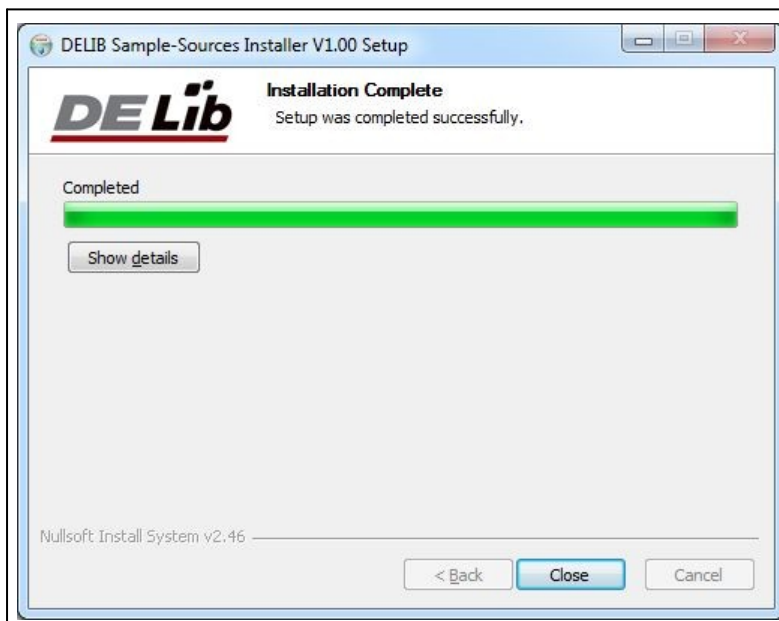
Wählen Sie den Installationsordner und drücken Sie Install.



Die DELIB Sample Sources werden nun installiert.



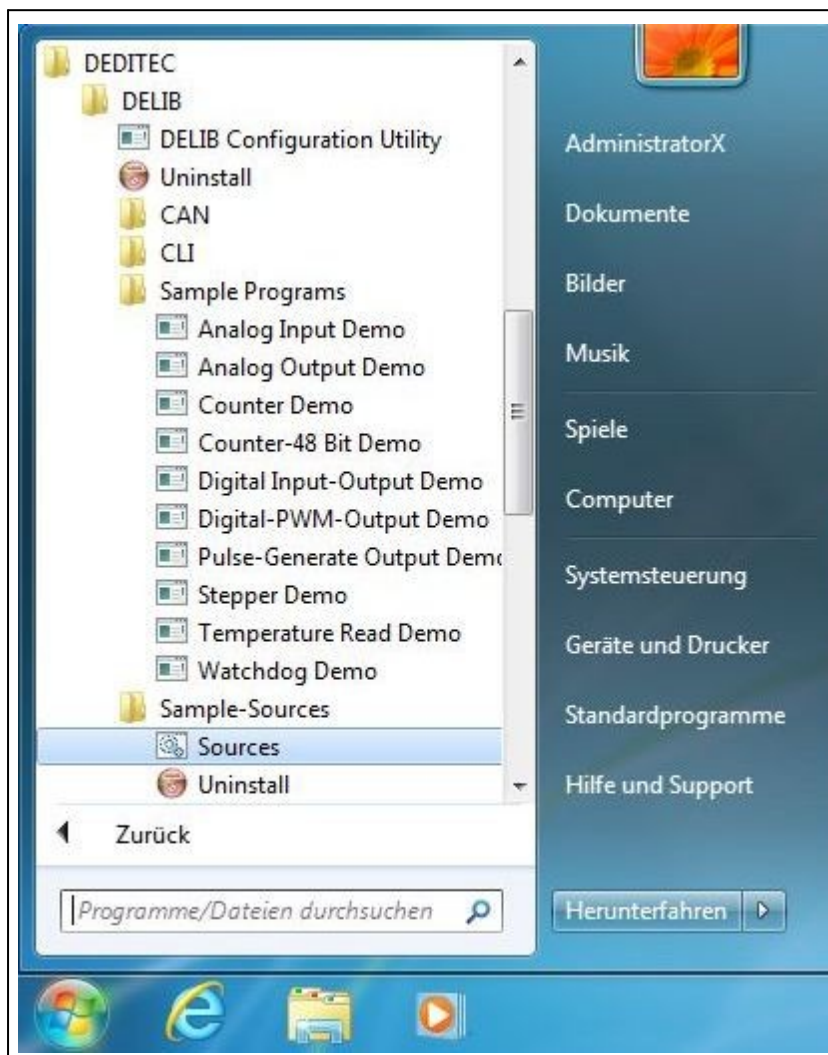
Die DELIB Sample Sources wurden erfolgreich installiert. Drücken Sie Close um die Installation zu beenden.



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

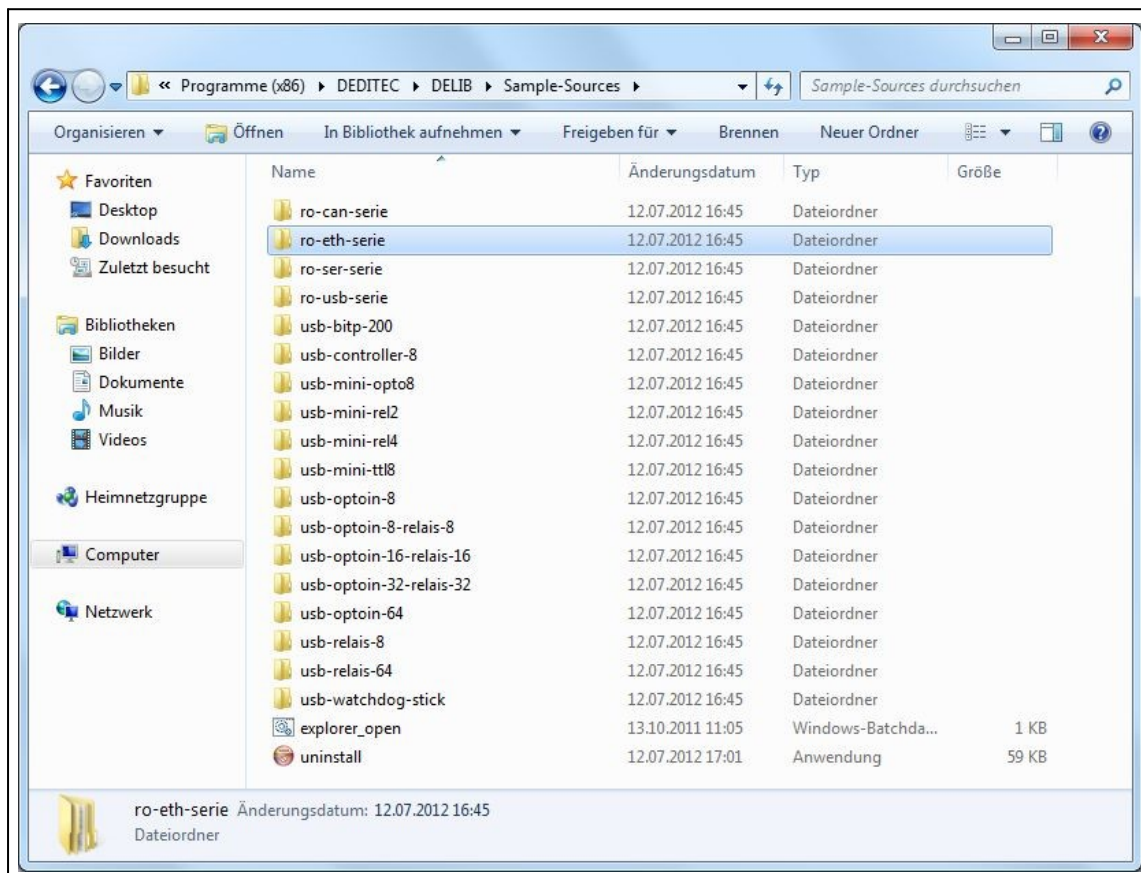


Nun öffnet sich der Windows-Explorer mit einer Übersicht aller Produkte für die ein Beispielprogramm verfügbar ist.

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.

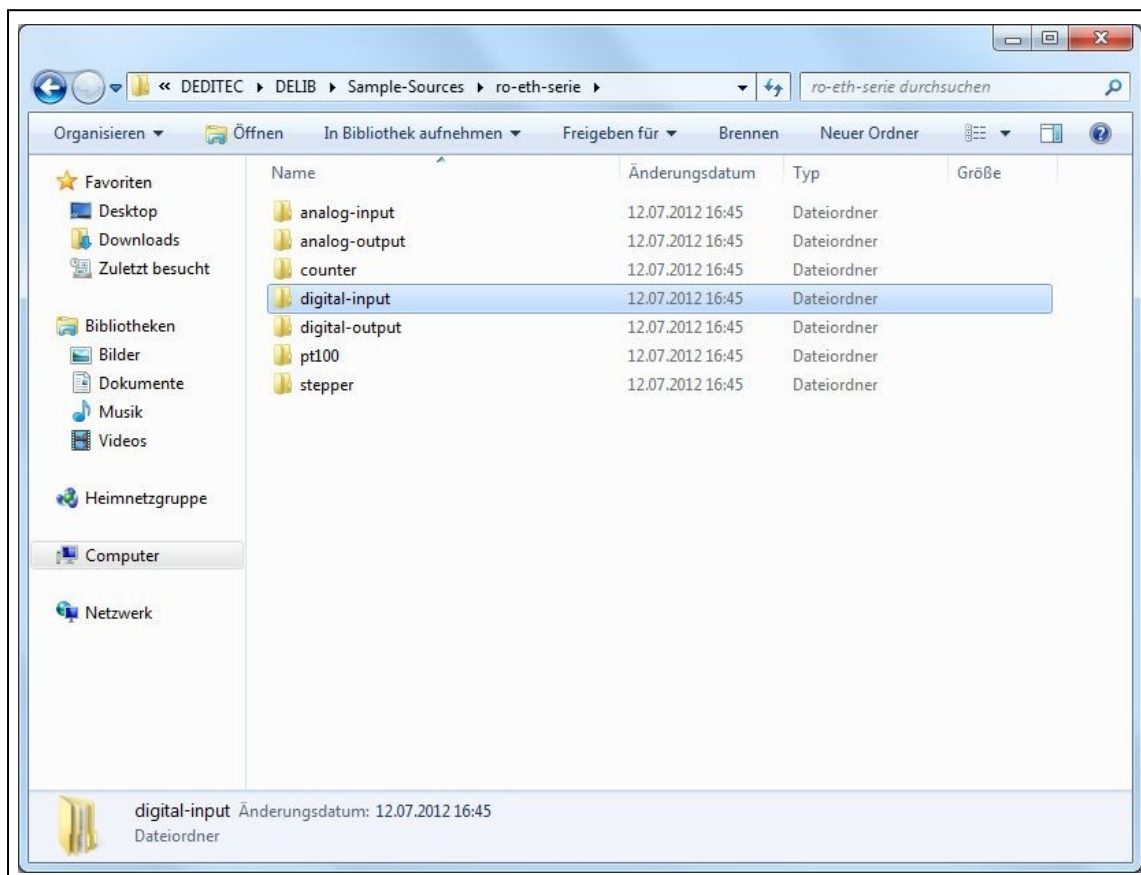
Da es sich um ein RO-ETH-Produkt handelt, wählen bzw. öffnen Sie den Ordner ro-eth-serie.



4.3.2.2. Schritt 2 - Kategorieauswahl

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

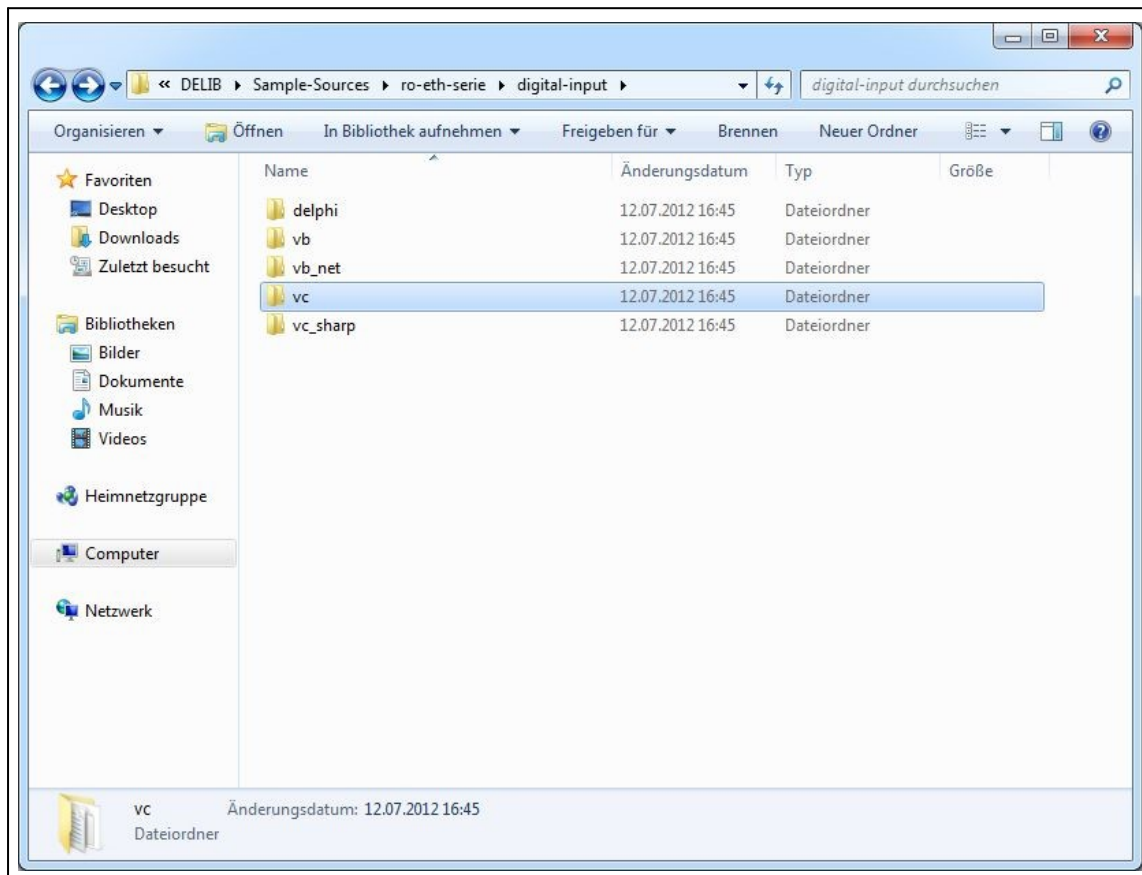
Da wir uns in diesem Beispiel auf die digitalen Eingänge konzentrieren, wählen Sie die Kategorie digital-input



4.3.2.3. Schritt 3 - Programmiersprachenauswahl

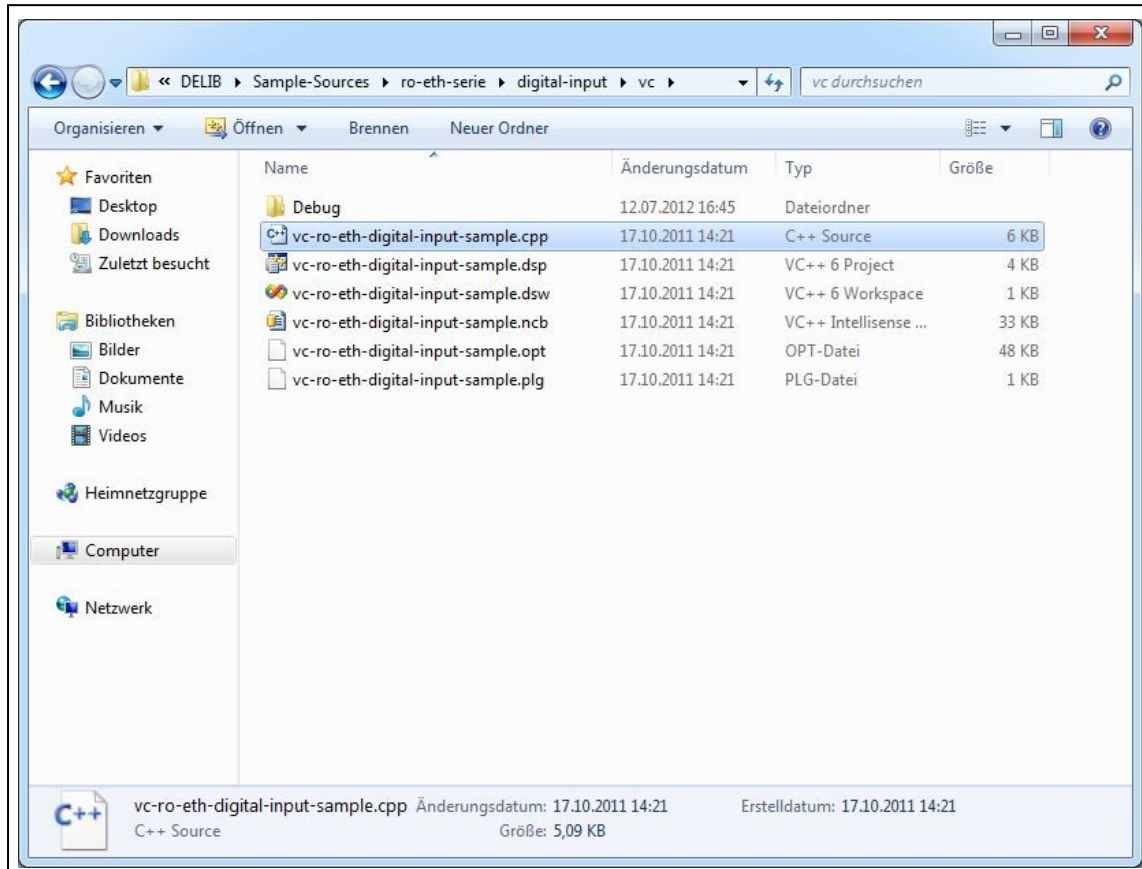
In diesem Schritt sehen Sie alle verfügbaren Programmierbeispiele der gewählten Kategorie, sortiert nach Programmiersprachen.

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

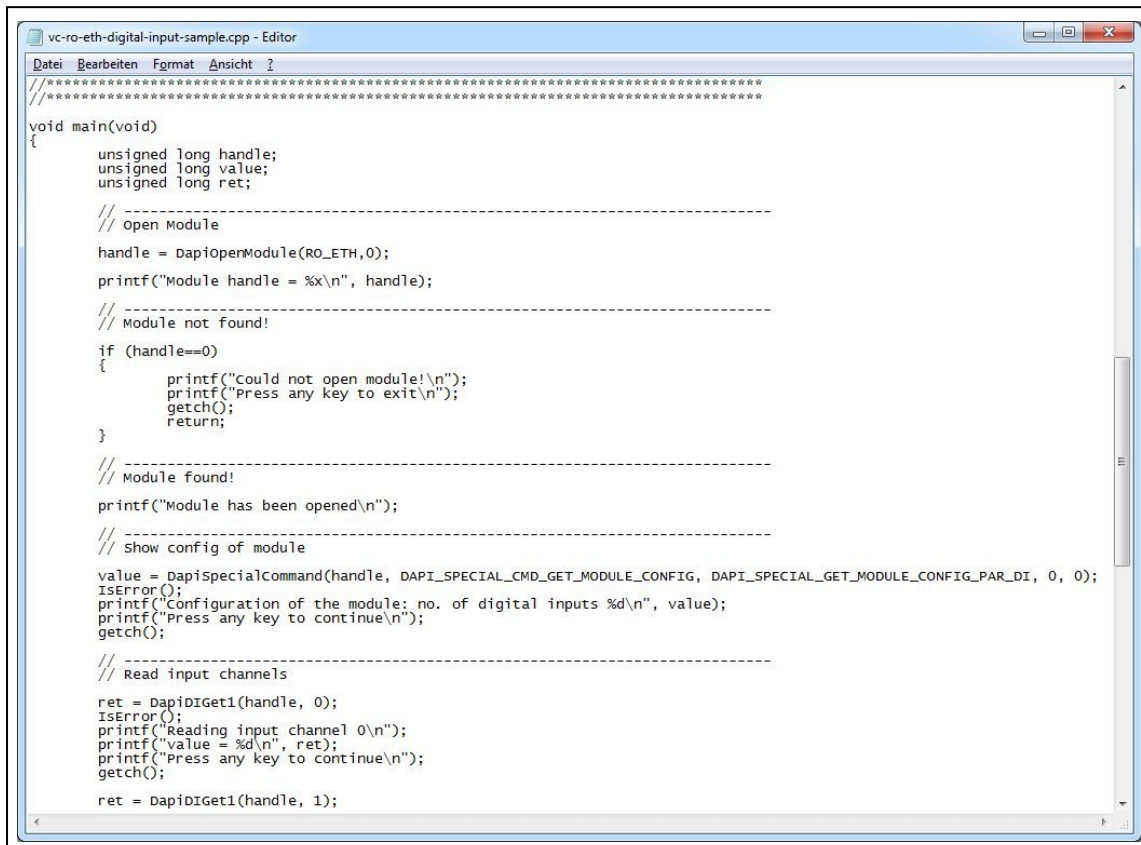


4.3.2.4. Schritt 4 - Quellcode

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

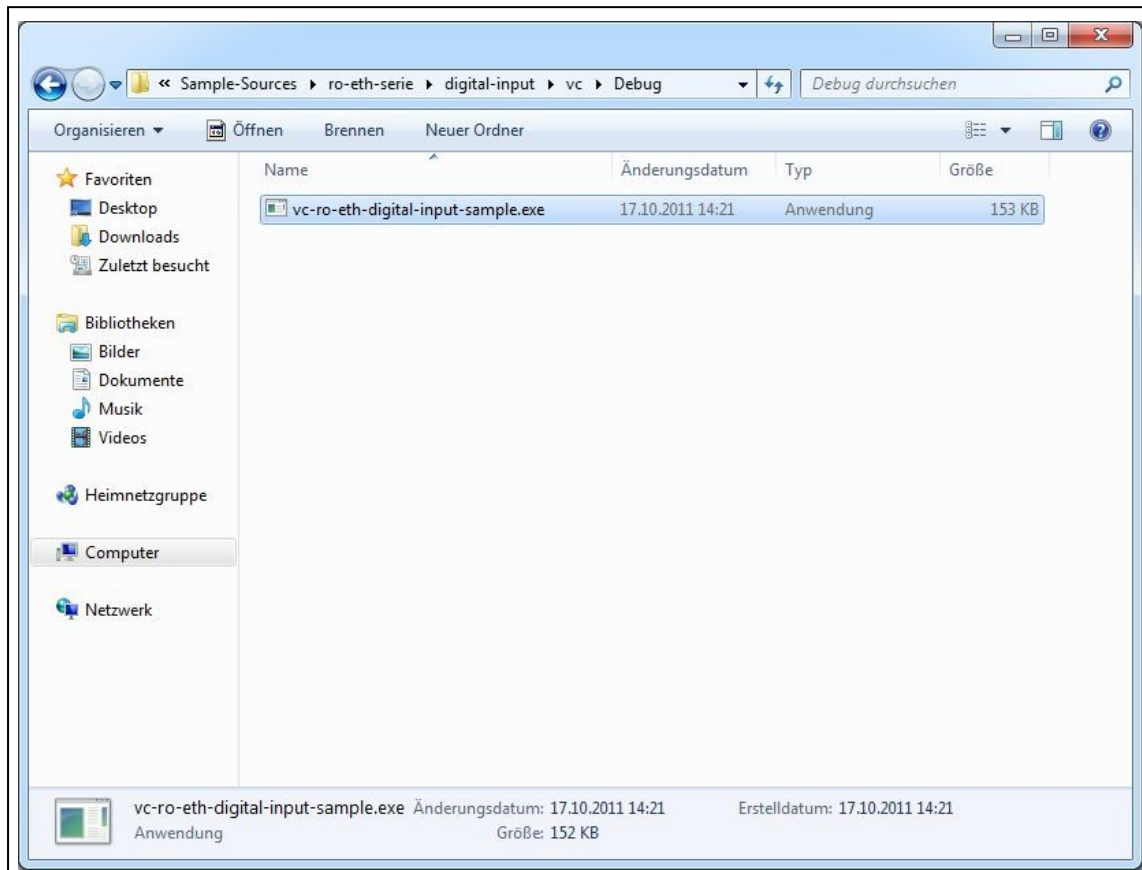
    value = DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_GET_MODULE_CONFIG, DAPI_SPECIAL_GET_MODULE_CONFIG_PAR_DI, 0, 0);
    if (value < 0)
    {
        printf("Configuration of the module: no. of digital inputs %d\n", value);
        printf("Press any key to continue\n");
        getch();
    }

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

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

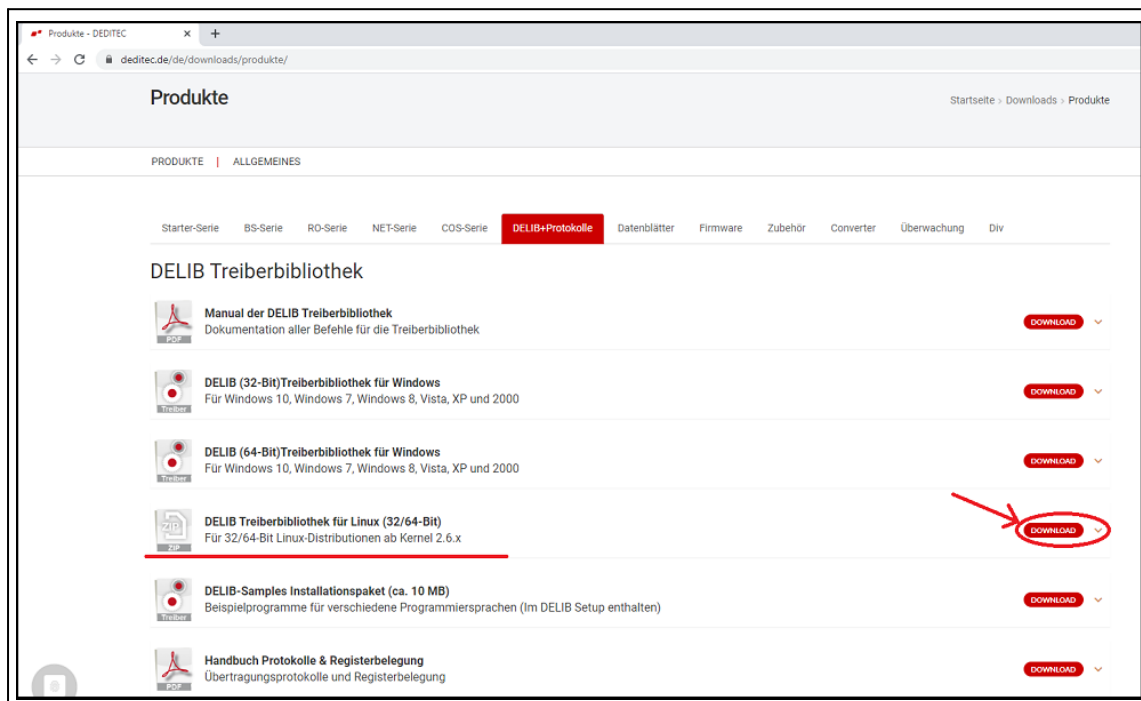
    ret = DapiGetI(handle, 1);
```


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

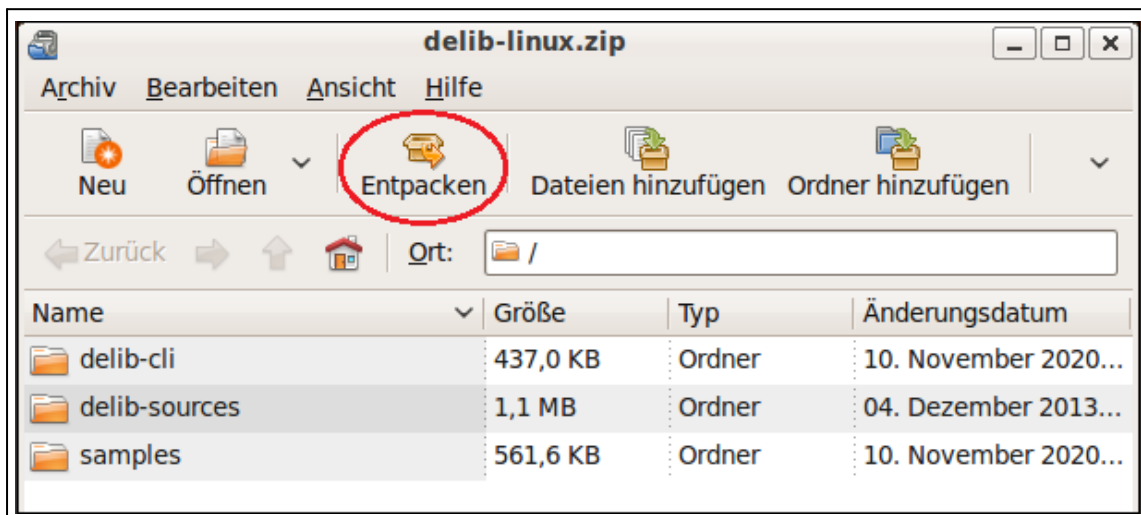


4.4. DELIB für Linux

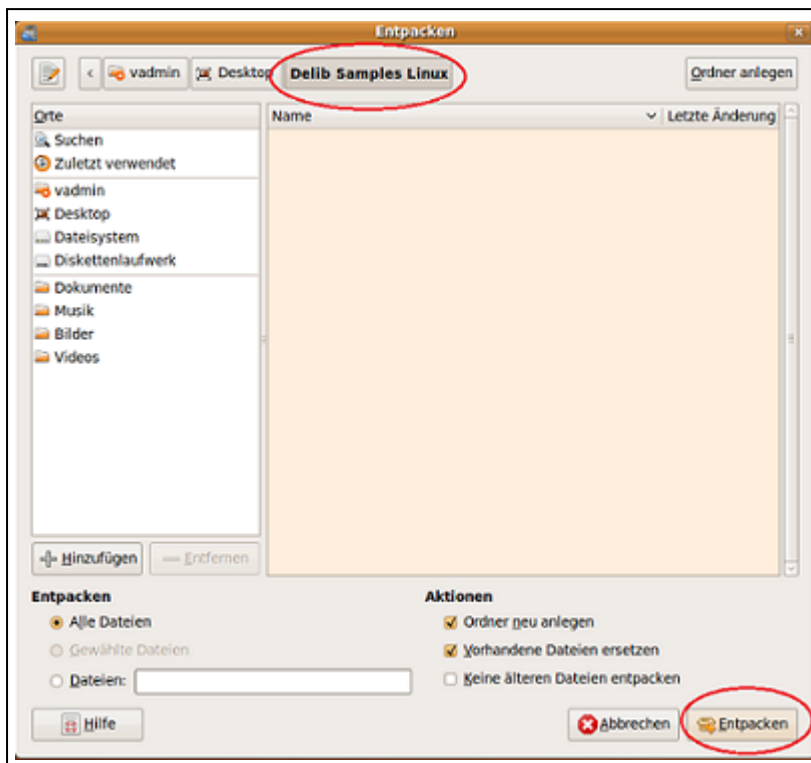
Laden Sie sich die Delib Linux Treiberbibliothek unter "www.deditec.de/de/downloads/produkte/" im Reiter „DELIB+Protokolle“ oder unter "www.deditec.de/media/zip/delib/delib-linux.zip" direkt auf ihr Linux-System.



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. DELIB Sample Sources (Windows Programmbeispiele) 2

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

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

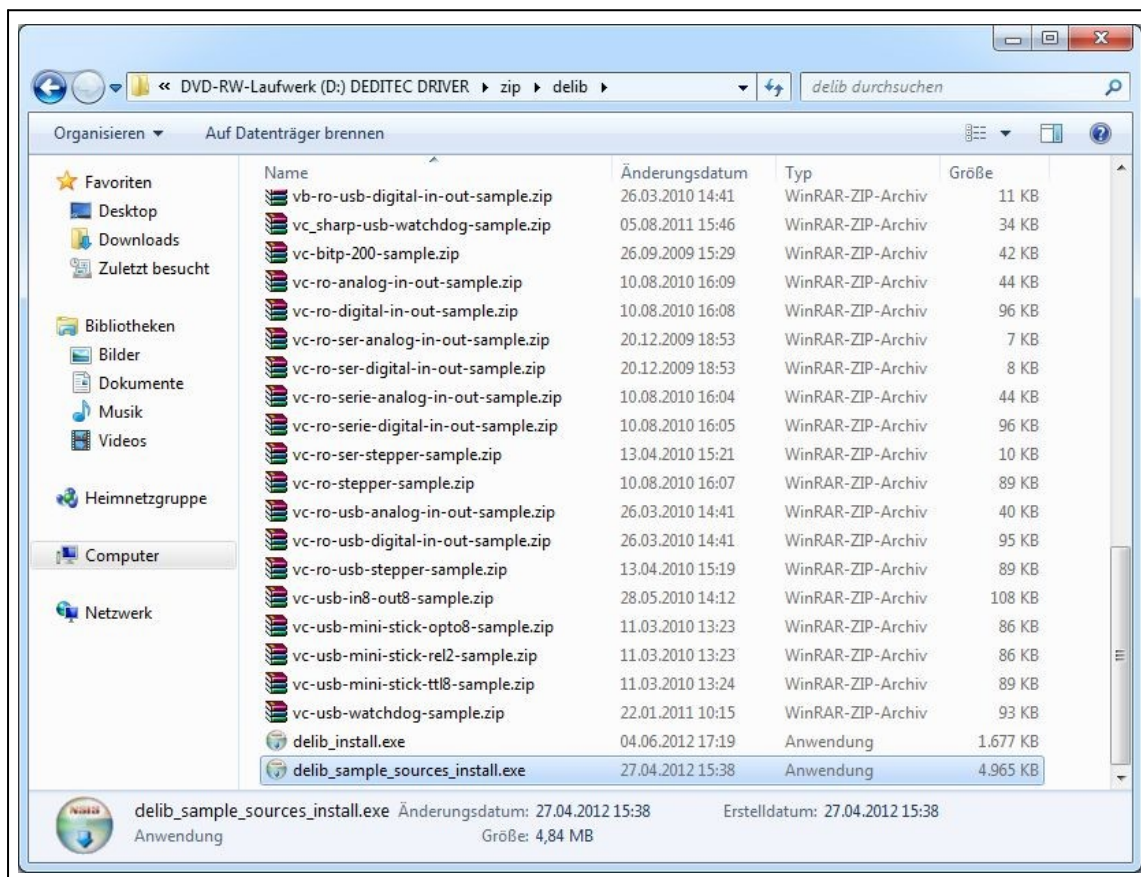
- C
- C++
- C#
- Delphi
- VisualBasic
- VB.NET
- MS-Office
- LabVIEW
- Java

4.4.1.1. Installation DELIB Sample Sources 2

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

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

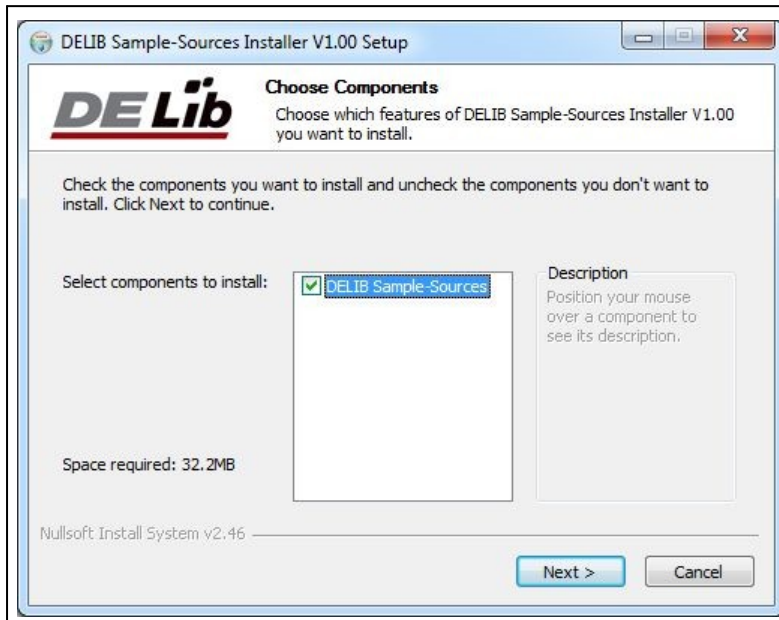
Eine aktuelle Version der Sample Sources finden Sie auch im Internet unter <http://www.deditec.de/de/delib>



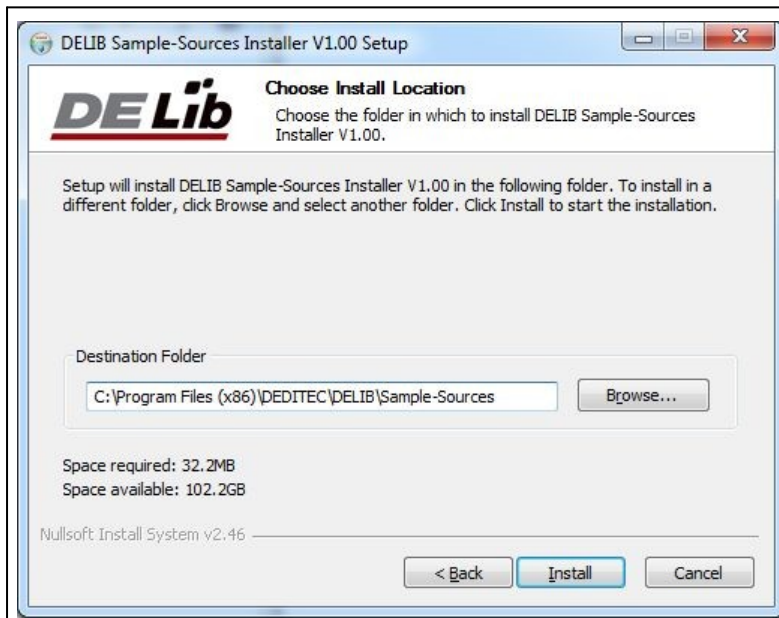
Startbild des DELIB Sample Sources Installer



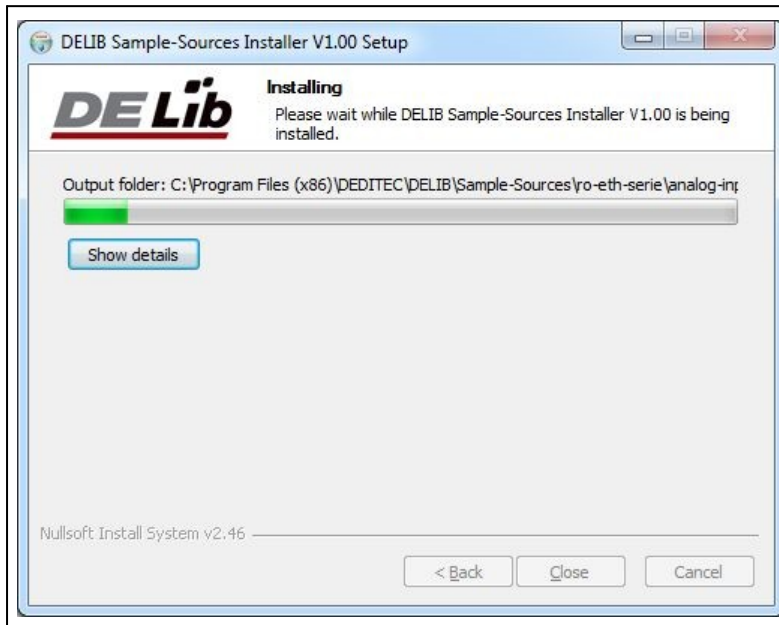
Drücken Sie Next.



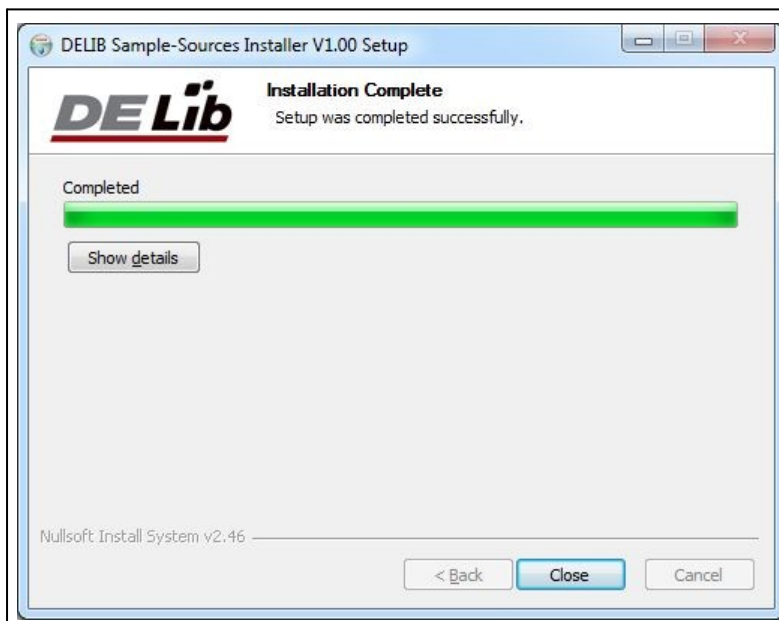
Wählen Sie den Installationsordner und drücken Sie Install.



Die DELIB Sample Sources werden nun installiert.



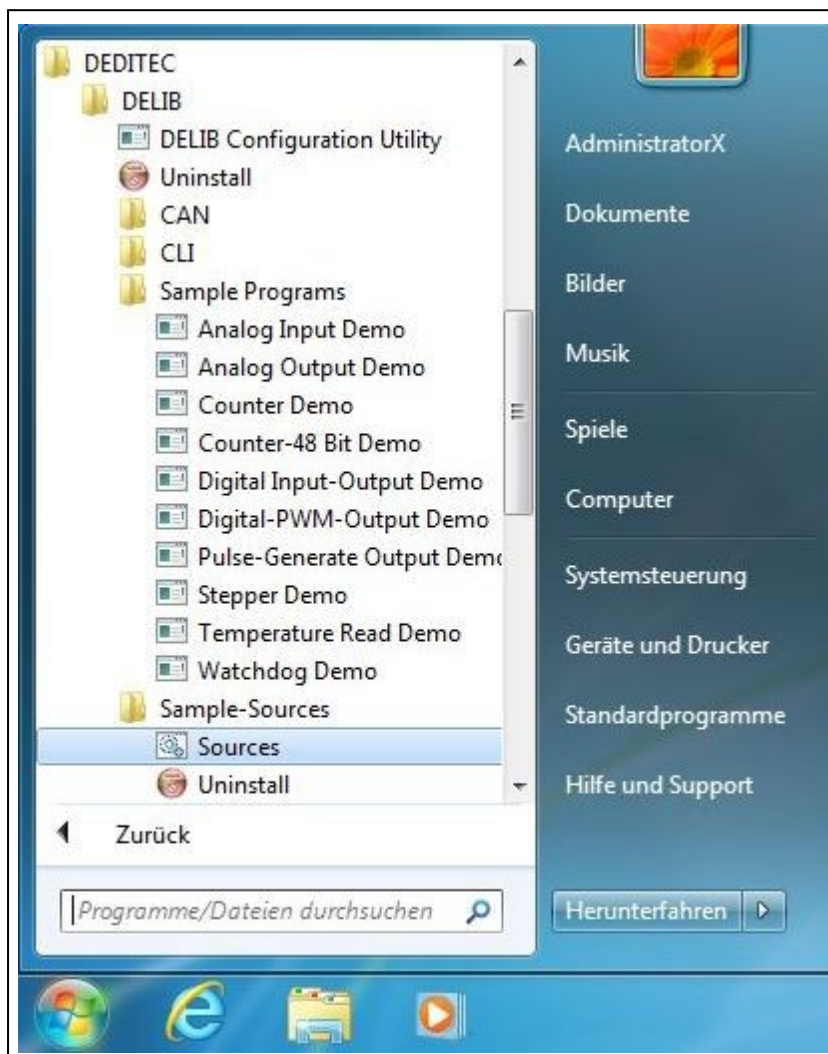
Die DELIB Sample Sources wurden erfolgreich installiert. Drücken Sie Close um die Installation zu beenden.



4.4.1.2. Benutzung der DELIB Sample Sources 2

Nach Installation der DELIB Sample Sources finden Sie diese unter

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

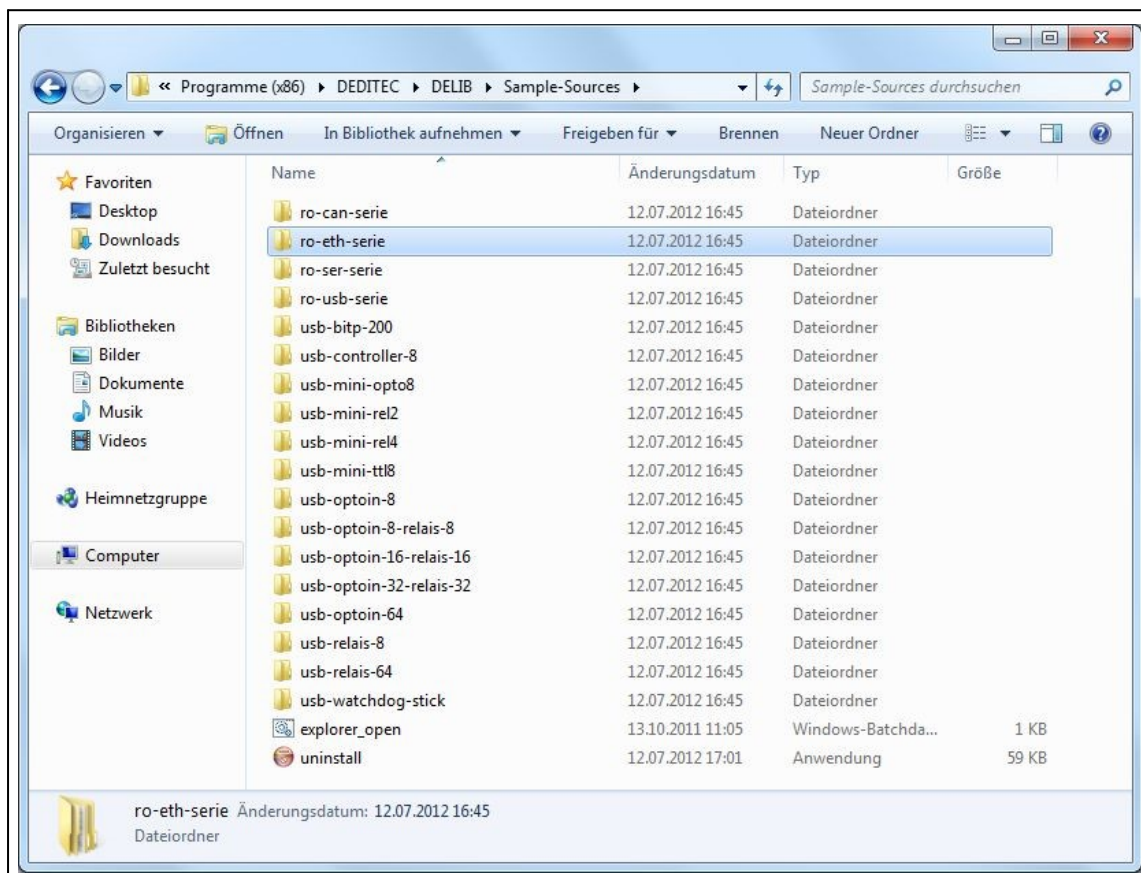


Nun öffnet sich der Windows-Explorer mit einer Übersicht aller Produkte für die ein Beispielprogramm verfügbar ist.

4.4.1.2.1. Schritt 1 - Produktauswahl 2

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

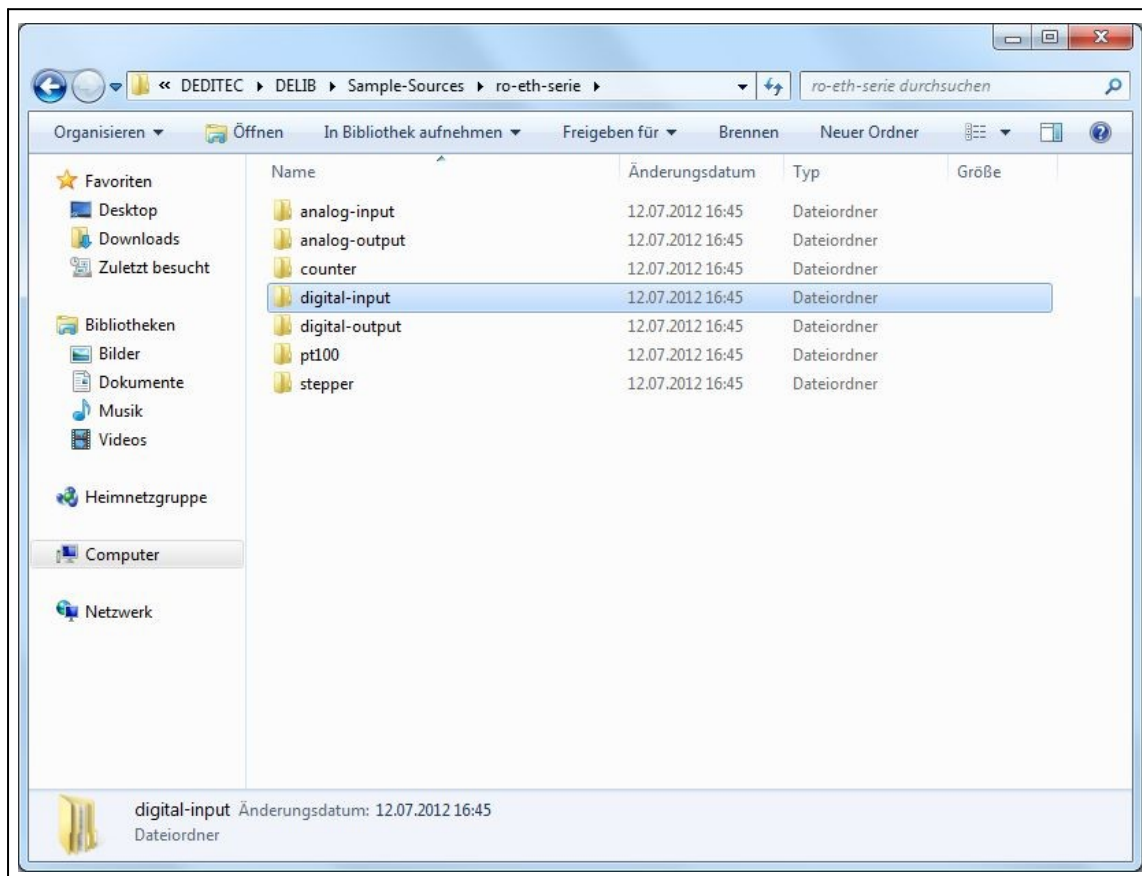
Da es sich um ein RO-ETH-Produkt handelt, wählen bzw. öffnen Sie den Ordner ro-eth-serie.



4.4.1.2.2. Schritt 2 - Kategorieauswahl 2

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

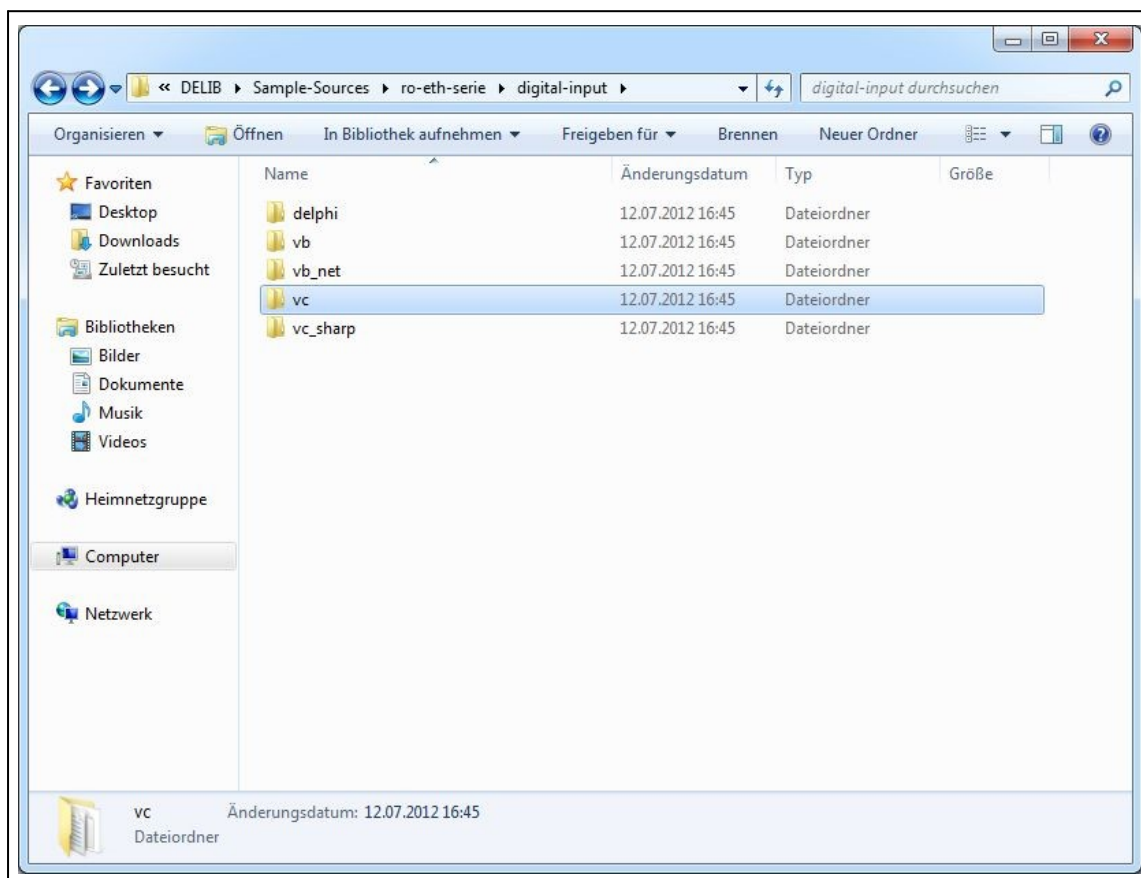
Da wir uns in diesem Beispiel auf die digitalen Eingänge konzentrieren, wählen Sie die Kategorie digital-input



4.4.1.2.3. Schritt 3 - Programmiersprachenauswahl 2

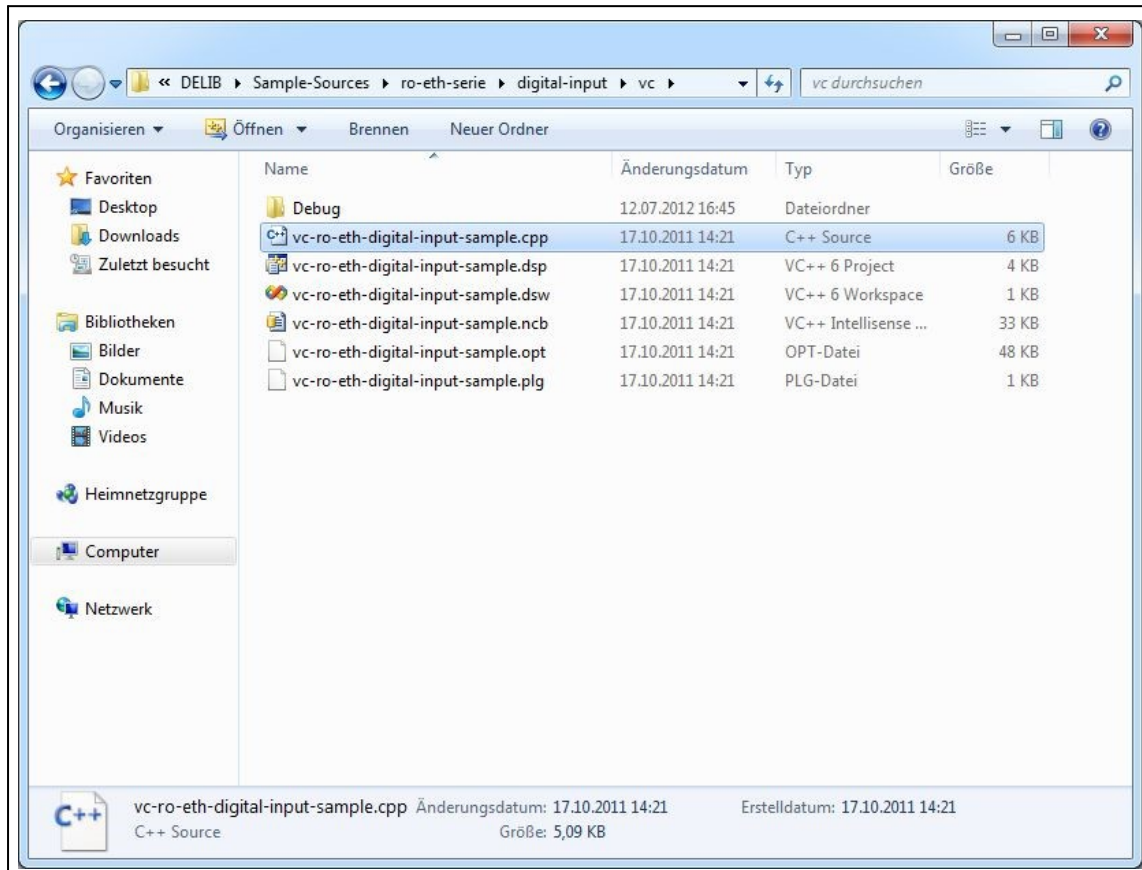
In diesem Schritt sehen Sie alle verfügbaren Programmierbeispiele der gewählten Kategorie, sortiert nach Programmiersprachen.

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

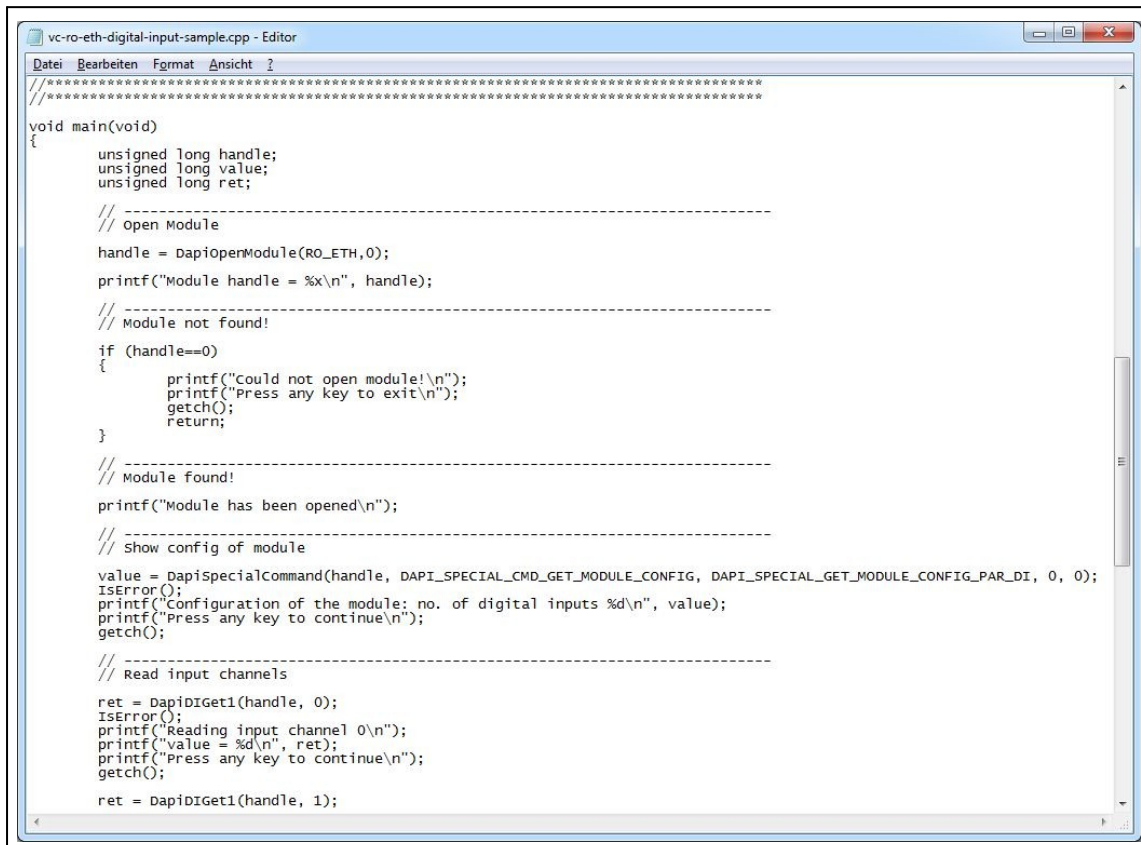


4.4.1.2.4. Schritt 4 - Quellcode 2

Nach Auswahl der Programmiersprache erhalten Sie folgende Übersicht:



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



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

    // -----
    // Open Module

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

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

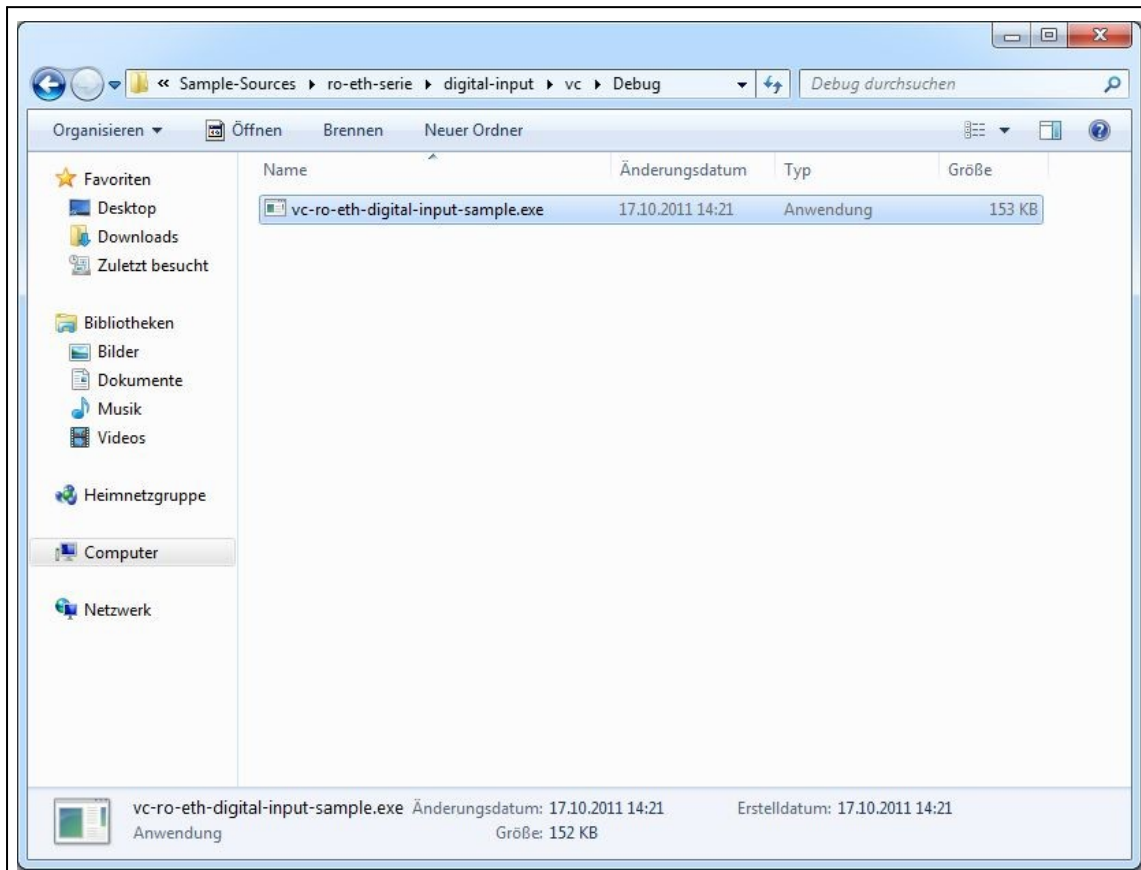
    // -----
    // Module found!
    printf("Module has been opened\n");

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

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

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


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



4.4.2. Verwenden der DELIB Treiberbibliothek für Linux

4.4.2.1. Delib USB-Sample in Linux

Voreinstellungen

In diesem Programmbeispiel wird ein USB_REL AIS_8 Modul angesprochen. Sollten Sie ein anderes Modul verwenden, müssen Sie in der Datei

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

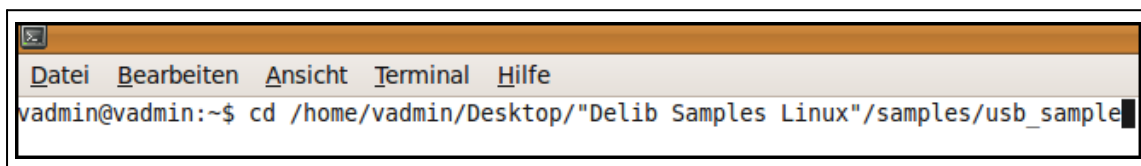
```
23
24 #include <stdio.h>
25 #include <stdlib.h>
26 #include <unistd.h>
27
28 #include "../delib-sources/delib/library/delib/delib.h"
29
30 int main()
31 {
32     ULONG i;
33     ULONG handle=0;
34
35     printf("\n\n");
36     printf("-----\n");
37     printf("-----\n");
38     printf("-----\n");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

Kompilieren des USB-Samples

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

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

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



```

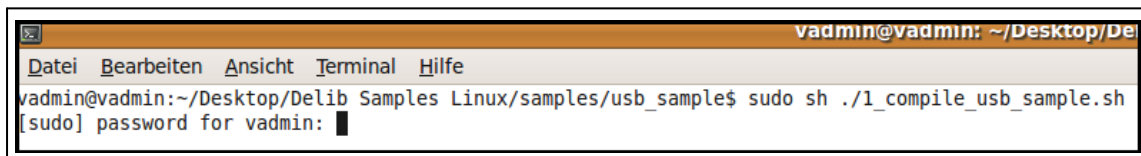
Datei Bearbeiten Ansicht Terminal Hilfe
vadmin@vadmin:~$ cd /home/vadmin/Desktop/"Delib Samples Linux"/samples/usb_sample

```

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

„sudo sh ./1_compile_usb_sample.sh“.

Geben Sie, falls nötig, Ihr Benutzerkennwort ein.



```

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

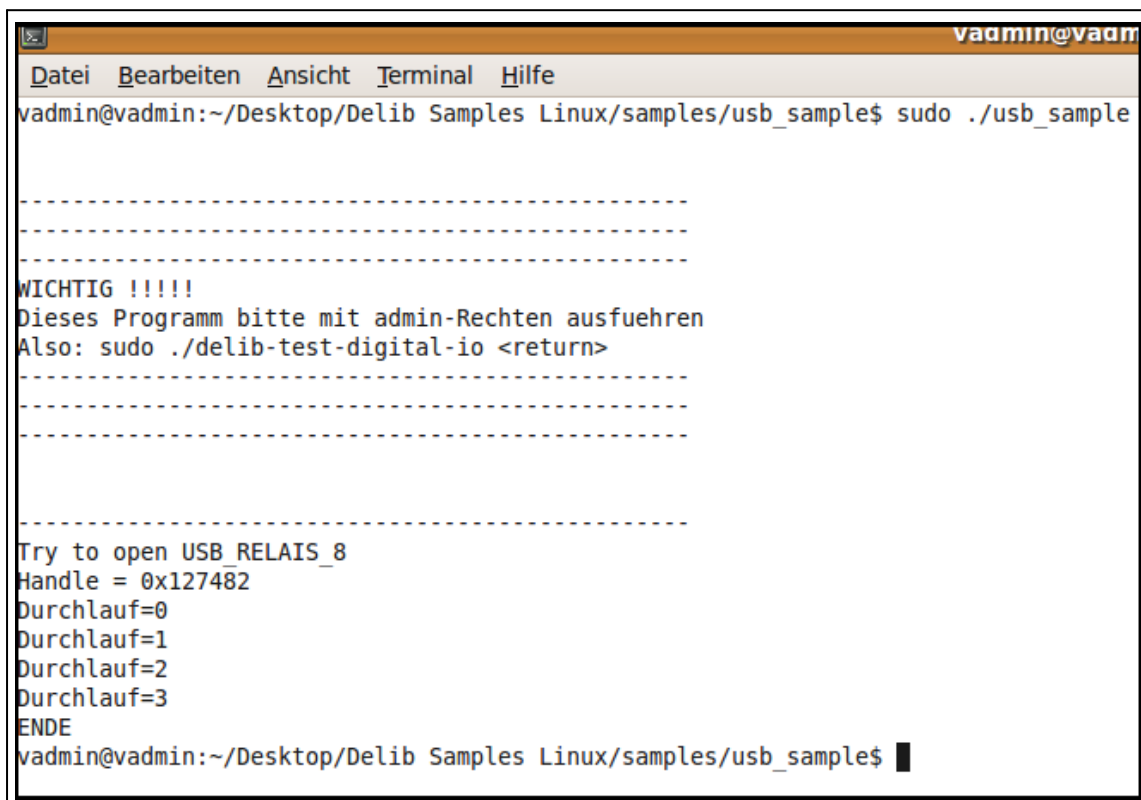
```

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

Es wurde die Datei "usb_sample" dem Verzeichnis hinzugefügt.

Jetzt können Sie das Beispielprogramm mit "sudo ./usb_sample" ausführen.

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



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

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

-----

Try to open USB_REL AIS_8
Handle = 0x127482
Durchlauf=0
Durchlauf=1
Durchlauf=2
Durchlauf=3
ENDE
vadmin@vadmin:~/Desktop/Delib Samples Linux/samples/usb_sample$
```

Das Programm wird nun ausgeführt.

In diesem Beispiel werden alle digitalen Ausgänge des USB_REL AIS_8 in einer Schleife an und wieder ausgeschaltet.

4.4.2.2. Delib ETH-Sample in Linux

Voreinstellungen

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

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

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

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

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

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

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

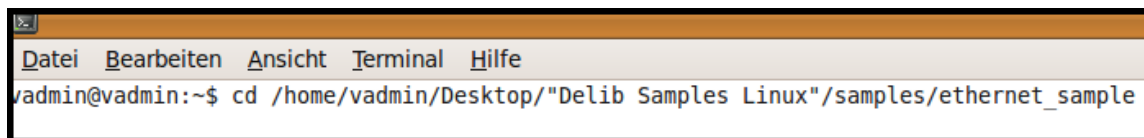
Auskommentieren,
falls keine digitalen
Eingänge vorhanden

Kompilieren des ETH-Samples

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

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

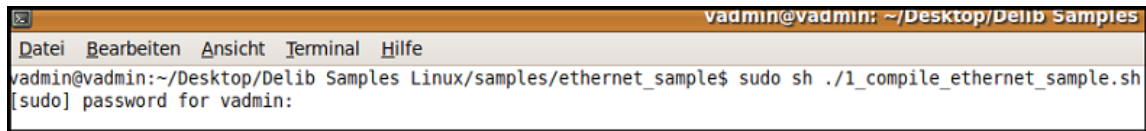
Tipp: Sollten in Ihrem Ordnernamen Leerzeichen enthalten sein, geben Sie diese wie im unteren Beispiel dargestellt in " " an.

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

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.



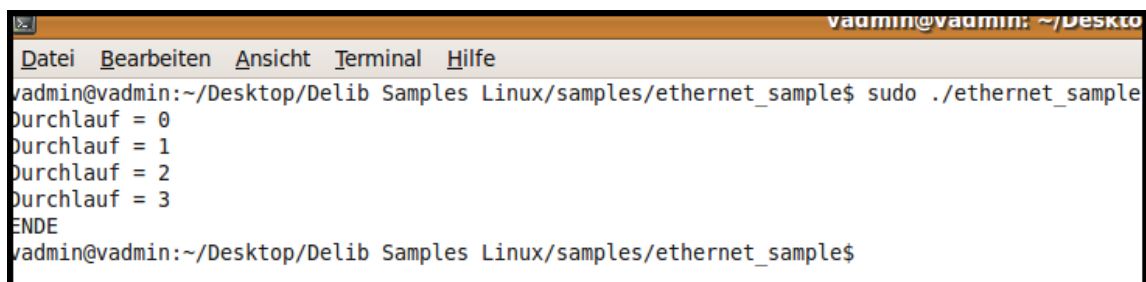
```
vadmin@vadmin: ~/Desktop/Delib Samples
Datei Bearbeiten Ansicht Terminal Hilfe
vadmin@vadmin:~/Desktop/Delib Samples Linux/samples/ethernet_sample$ sudo sh ./1_compile_ethernet_sample.sh
[sudo] password for vadmin:
```

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.

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

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

Definition für USB-Module (Linux)

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

Return-Wert

Zustand der gelesenen digitalen Eingänge

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

Zustand der FlipFlips der digitalen Eingänge

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

Zustand der gelesenen analogen Eingänge

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

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

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

4.4.3.1. Konfiguration des DELIB CLI

Voreinstellungen

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

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

Inhalt der "delib_cli.cfg":

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

moduleID

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

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

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

moduleNR

Die moduleNR wird im DELIB Configuration Utility vergeben.

Diese Nummer dient zur Identifizierung identischer Hardware.

Der Standardwert ist 0.

RO-ETH_ipAddress

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

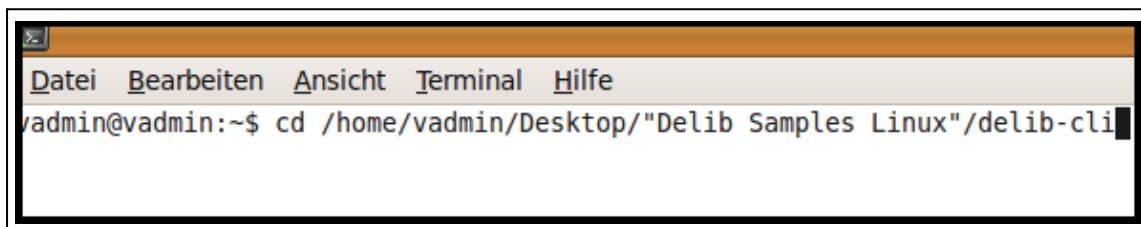
Die IP-Adresse der ETH-Module können über das DELIB Configuration Utility sowie über die Weboberfläche des Moduls eingestellt werden.

Kompilieren des Delib-CLI-Samples

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

"cd /<Verzeichnispfad>" zunächst in das "../delib_cli/" Verzeichnis.

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

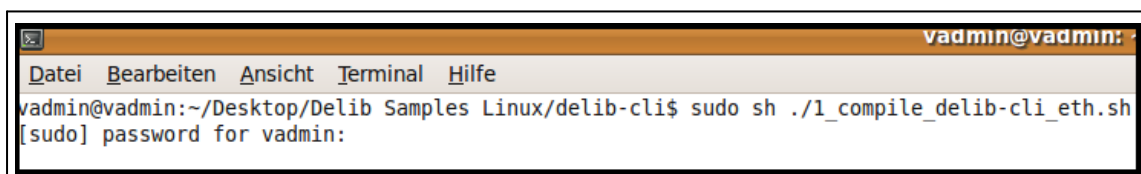


```
admin@vadmin:~$ cd /home/vadmin/Desktop/"Delib Samples Linux"/delib-cli
```

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

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

Geben Sie, falls nötig, Ihr Benutzerkennwort ein.

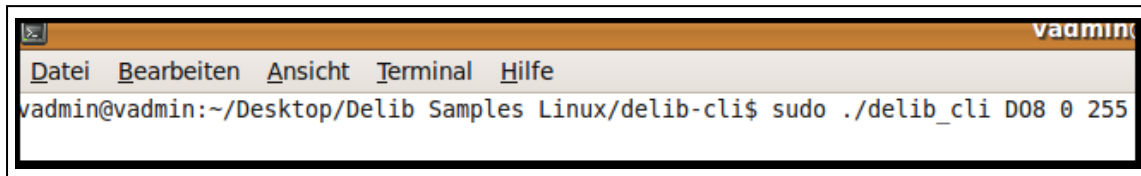


```
admin@vadmin:~/Desktop/Delib Samples Linux/delib-cli$ sudo sh ./1_compile_delib-cli_eth.sh
[sudo] password for vadmin:
```

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

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

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

A terminal window with a title bar containing a close button icon and the text 'vadmin'. The menu bar includes 'Datei', 'Bearbeiten', 'Ansicht', 'Terminal', and 'Hilfe'. The terminal text shows the user 'vadmin' at host 'vadmin' in the directory '~/Desktop/Delib Samples Linux/delib-cli' executing the command 'sudo ./delib_cli D08 0 255'.

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

4.4.3.2. DELIB CLI Beispiele

Digitale Ausgänge

```
sudo delib_cli DO1 17 1
```

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

```
sudo delib_cli DO1 3 0
```

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

Digitale Eingänge

```
sudo delib_cli DI1 3
```

Beispiel eines Rückgabewertes: **1**

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

```
sudo delib_cli DI8 0 hex
```

Beispiel eines Rückgabewertes: **0xFF**

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

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

```
sudo delib_cli FF 0
```

Beispiel eines Rückgabewertes: **192**

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

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

```
sudo delib_cli FF 32
```

Beispiel eines Rückgabewertes: **65535**

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

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

```
sudo delib_cli FF 0 hex
```

Beispiel eines Rückgabewertes: **0xD00**

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

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

Analoge Ausgänge

```
sudo delib_cli AO 7 4711
```

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

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

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

Analoge Eingänge

```
sudo delib_cli AI 2
```

Beispiel eines Rückgabewertes: **1234**

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

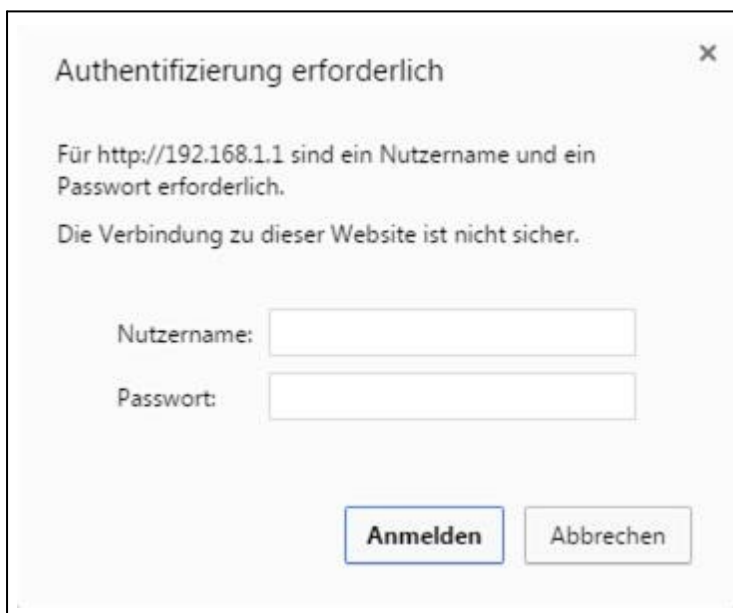
```
sudo delib_cli AI 2 hex
```

Beispiel eines Rückgabewertes: **0x1FA**

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

4.5. Weboberfläche

Geben Sie die IP-Adresse (Auslieferungszustand 192.168.1.1) des Moduls in einem Internet-Browser ein.



The screenshot shows a web browser authentication dialog box with the title "Authentifizierung erforderlich" and a close button (X). The text inside the dialog reads: "Für http://192.168.1.1 sind ein Nutzernamen und ein Passwort erforderlich." and "Die Verbindung zu dieser Website ist nicht sicher." Below this text are two input fields: "Nutzername:" and "Passwort:". At the bottom of the dialog are two buttons: "Anmelden" and "Abbrechen".

Der integrierte Web-Server des Moduls erfordert eine Authentifizierung

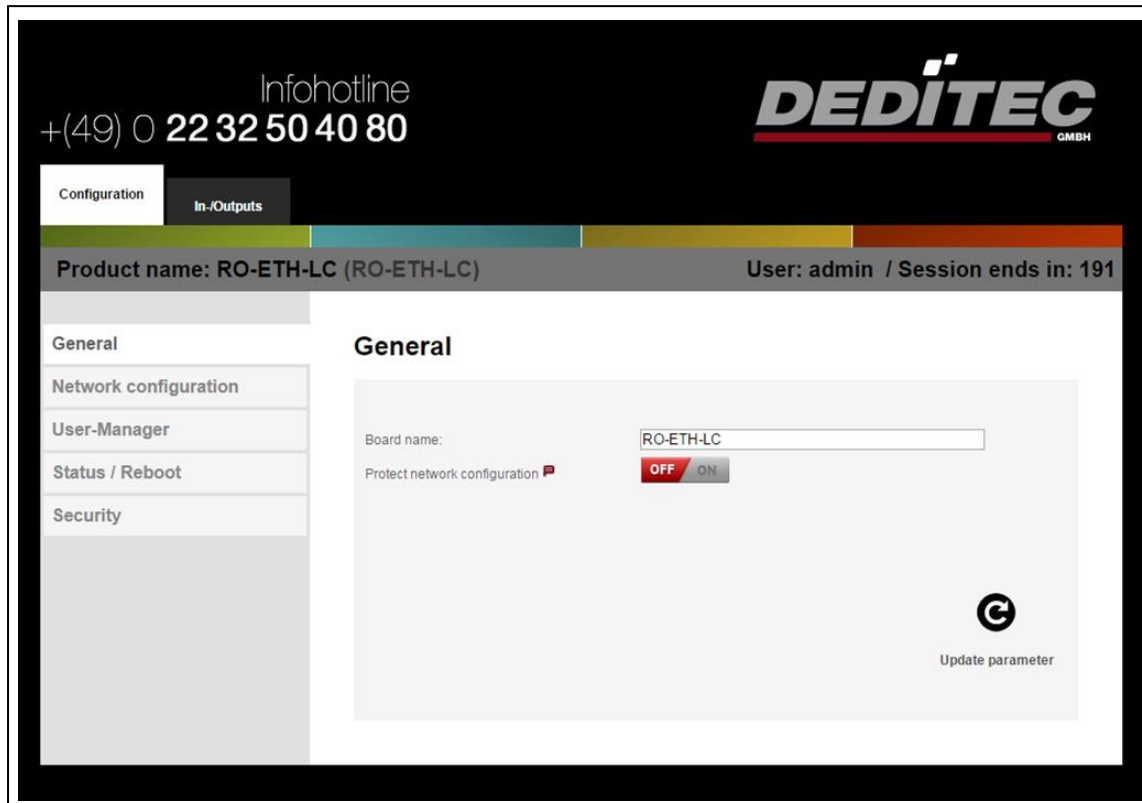
um das Modul vor unberechtigten Zugriffen zu schützen.

Standardmäßig ist folgender Benutzer eingerichtet:

Benutzername	Passwort	Rechte
admin	admin	Administratoren-Rechte

4.5.1. Konfiguration

4.5.1.1. Allgemein



Boardname

Das ist der Name des Moduls, welcher auch im DELIB-Configuration-Utility angezeigt wird.

Dieser Name dient zur Identifizierung mehrerer DEDITEC-Ethernet-Module im Netzwerk.

Protect network configuration

Ist diese Option aktiviert, kann die Netzwerk Konfiguration nur über die Weboberfläche des Moduls geändert werden.

Ein Ändern dieser Konfiguration (z.B. über das DELIB-Configuration Utility) wird in diesem Fall verhindert.

4.5.1.2. Netzwerk Konfiguration

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' and 'In-Outputs'. The 'Configuration' tab is active. Below the tabs, a status bar shows 'Product name: RO-ETH-LC (RO-ETH-LC)' and 'User: admin / Session ends in: 197'. On the left side, there is a sidebar menu with options: 'General', 'Network configuration', 'User-Manager', 'Status / Reboot', and 'Security'. The 'Network configuration' option is selected. The main content area is titled 'Network configuration' and contains several input fields: 'MAC' (00:C0:D5:02:00:0D), 'Obtain IP address automatically (DHCP)' (a toggle switch set to 'OFF'), 'IP-address' (192.168.1.25), 'Netmask' (255.255.255.0), 'Std-GW' (192.168.1.254), and 'TCP port' (9912). Below these fields, a notice states: 'Notice: Changing TCP port requires board reboot'. At the bottom right of the configuration area, there is a circular refresh icon and the text 'Update parameter'.

Hier kann die Netzwerk Konfiguration des Moduls geändert werden.

Wird diese Konfiguration geändert, werden Sie automatisch auf die neue IP-Adresse weitergeleitet, sofern diese vom PC erreichbar ist.

Eine Änderung des Ports erfordert einen Neustart des Moduls.

4.5.1.3. Benutzer Manager

The screenshot shows the DEDITEC web interface for the 'User-Manager' module. At the top, there is an 'Inf hotline' number: +(49) 0 22 32 50 40 80. The DEDITEC logo is in the top right corner. Below the header, there are tabs for 'Configuration' and 'In-Outputs'. A status bar indicates 'Product name: RO-ETH-LC (RO-ETH-LC)' and 'User: admin / Session ends in: 196'. On the left, a sidebar lists menu items: 'General', 'Network configuration', 'User-Manager' (selected), 'Status / Reboot', and 'Security'. The main content area is titled 'User-Manager' and shows 'Modul Status: OK'. It contains a toggle switch for 'Webinterface requires login' (set to 'ON'), a 'Username:' field with 'admin' entered, a 'Set password' button, and a 'Session valid time' field with '200' entered and 'sec' as the unit. A notice states 'Changes apply after board restart'. At the bottom right, there is a circular refresh icon and the text 'Update parameter'.

Remove

Löscht das entsprechende Benutzerkonto.

Add User

Erstellt ein neues Benutzerkonto. Die Eingabemaske fordert Sie auf einen Benutzernamen und Passwort einzugeben.

Session valid time

Gibt die Zeit an, wie lange eine Anmeldung gültig ist. Läuft diese Zeit ab, muss der Nutzer sich neu anmelden.

Achtung:

Die Änderung dieser Zeit benötigt einen Neustart des Moduls.

Klicken Sie auf ein Benutzerkonto (z.B. gast) um die Einstellungen zu ändern.

The screenshot shows the DEDITEC web interface. At the top, there is an infohotline number +49 0 22 32 50 40 80 and the DEDITEC GMBH logo. Below this is a navigation bar with tabs for Configuration, In-/Outputs, Custom, and Logout. A status bar indicates the product name as RO-ETH (DEDITEC RO-ETH-Module) and the user as admin, with a session ending in 125 seconds. On the left, a sidebar lists various configuration options: General, Network configuration, Network time (NTP), HTTP-Server, Mail-Server, User-Manager (selected), FW-Update, Log's, Status / Reboot, and Security. The main content area is titled 'User-Manager' and shows the status 'requesting userlist succeeded'. It displays a list of users, with 'gast' selected. The 'gast' user's configuration is shown in a form with fields for Rights (CONFIG rd, IO rd), available rights (IO_wr, CONFIG_wr, ADMINISTRATION), and set rights (IO_rd, CONFIG_rd). There are buttons for Remove, edit, and update. A Password field is also present with a set button. Below the user list, there is a Session valid time field set to 500 sec and a Notice: Changes apply after board restart. The DEDITEC logo and 'Update parameter' text are at the bottom right.

Edit

Ändert die Zugriffsberechtigungen des aktuellen Benutzerkontos. Klicken Sie auf eine Zugriffsberechtigung um diese entweder aus- oder abzuwählen.

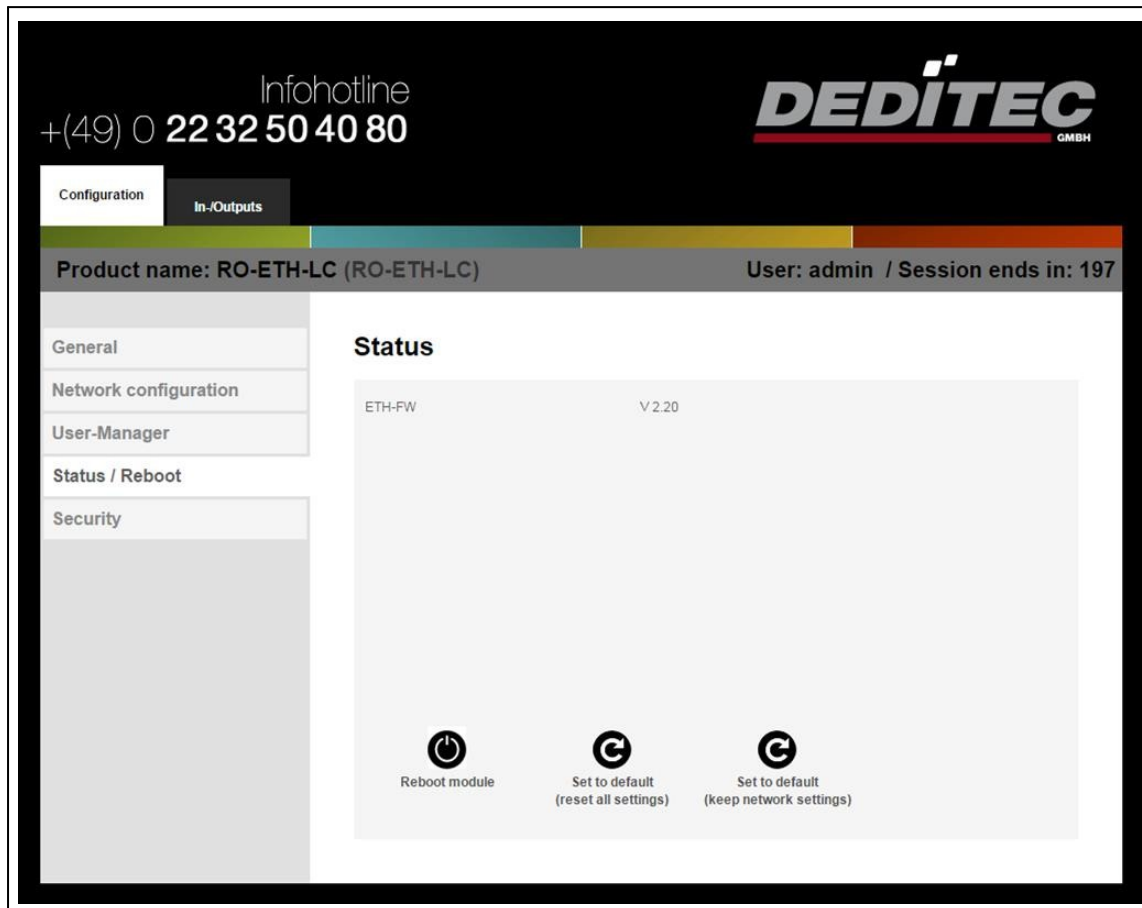
Update

Aktiviert die Zugriffsberechtigungen des aktuellen Benutzerkontos.

Password

Hier kann für das aktuelle Benutzerkonto ein neues Passwort gesetzt werden, welches mit set bestätigt werden muss.

4.5.1.4. Status



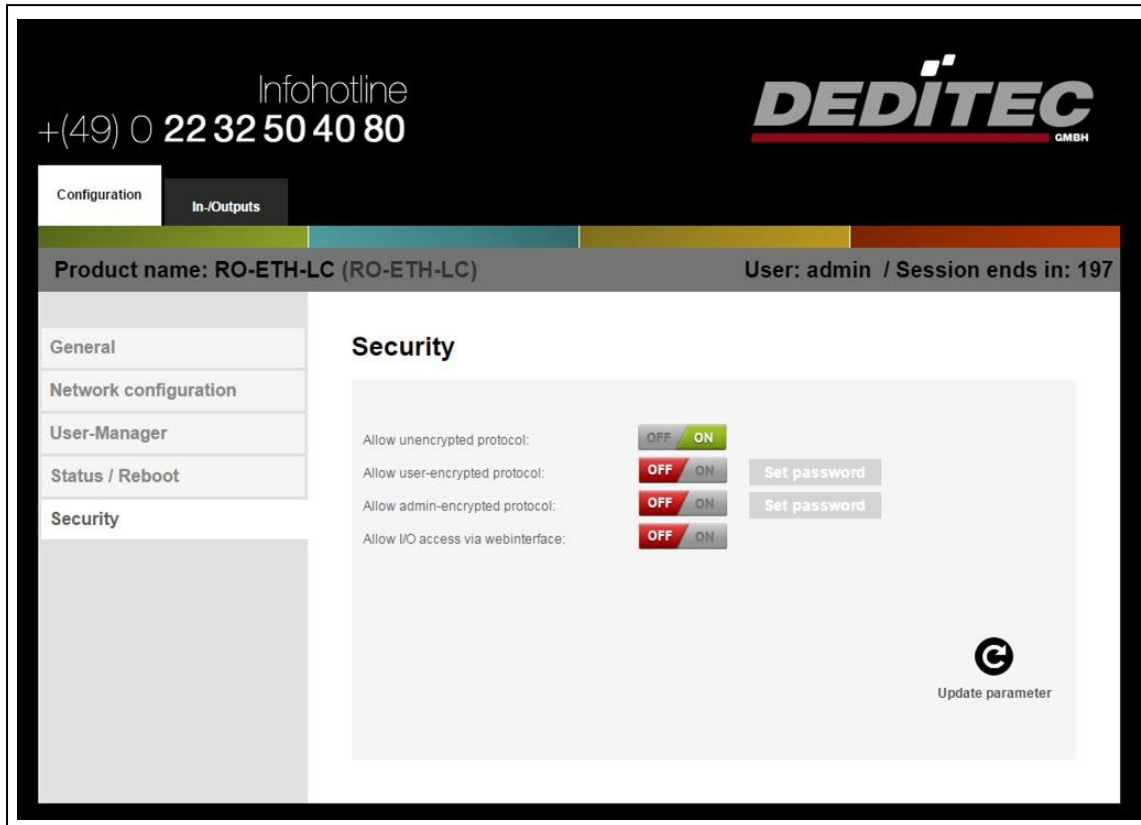
Auf der Status Seite sehen Sie die Revisionsnummern der wichtigsten System Prozesse.

Darüber hinaus, kann an dieser Stelle das Modul per Klick neugestartet, oder die Netzwerk-Einstellungen auf Werkseinstellungen zurückgesetzt werden.

Achtung

Das Neustarten des Moduls, sowie das Zurücksetzen auf Werkseinstellung benötigen Administratoren Berechtigungen.

4.5.1.5. Sicherheit



Allow unencrypted protocol

Diese Option bestimmt, ob ein Zugriff auf das Modul mit einem unverschlüsseltem Protokoll erlaubt wird.

Allow user-encrypted protocol

Diese Option bestimmt, ob ein Zugriff auf das Modul mit einem User-verschlüsseltem Protokoll erlaubt wird.

Dieses Protokoll verfügt über eingeschränkte Zugriffs-Rechte und empfiehlt sich für Benutzer, deren Kommunikation verschlüsselt aber keine System Einstellungen ändern sollen.

Allow admin-encrypted protocol

Diese Option bestimmt, ob ein Zugriff auf das Modul mit einem Admin-verschlüsseltem Protokoll erlaubt wird.

Dieses Protokoll wird benötigt um beispielsweise die Namen der angeschlossenen Ein-/Ausgänge zu lesen oder zu ändern.

Allow I/O access via webinterface

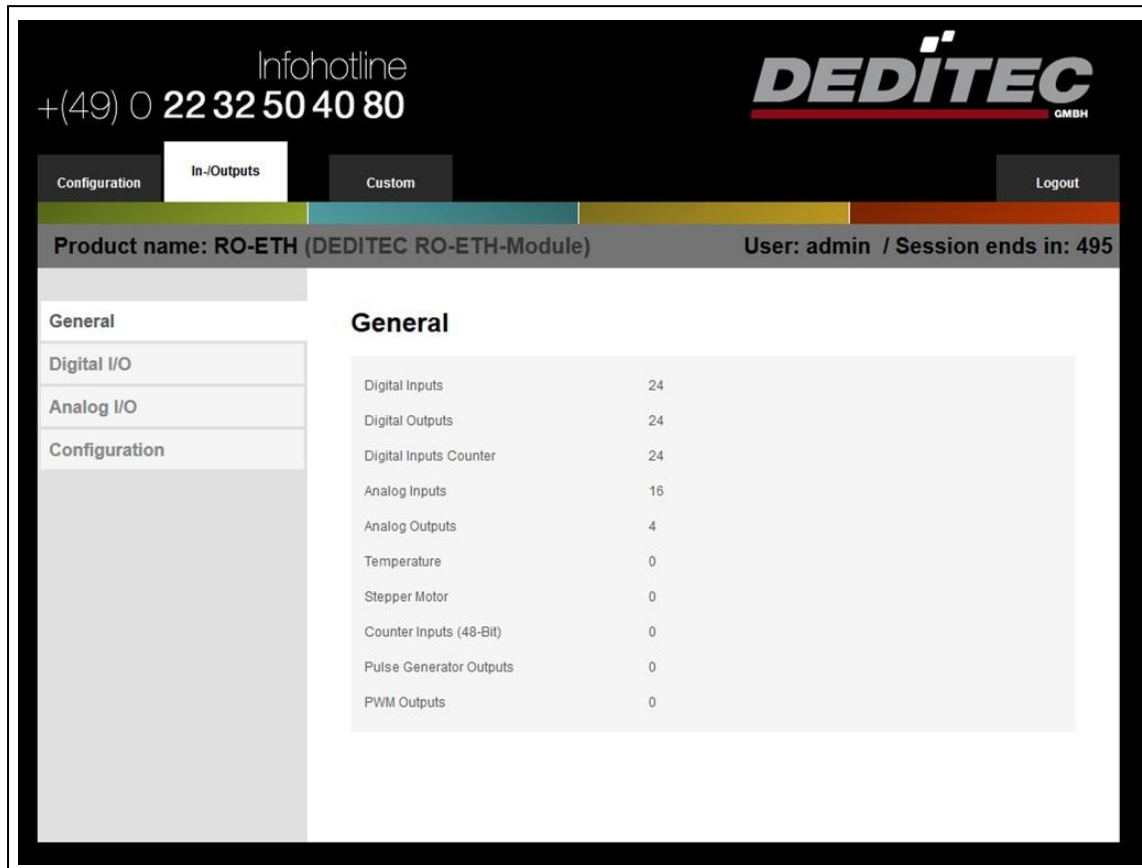
Diese Option bestimmt, ob angeschlossene Ein-/Ausgänge von der Weboberfläche aus gelesen/geschaltet werden können.

Achtung:

Wenn das Passwort zur User- oder Admin-Verschlüsselung geändert wird, muss dieses auch im DELIB-Configuration-Utility nachgetragen werden.

4.5.2. Ein-/Ausgänge

4.5.2.1. Allgemein



Infohotline
+(49) 0 22 32 50 40 80

DEDITEC
GMBH

Configuration In-/Outputs Custom Logout

Product name: RO-ETH (DEDITEC RO-ETH-Module) User: admin / Session ends in: 495

General

Digital Inputs	24
Digital Outputs	24
Digital Inputs Counter	24
Analog Inputs	16
Analog Outputs	4
Temperature	0
Stepper Motor	0
Counter Inputs (48-Bit)	0
Pulse Generator Outputs	0
PWM Outputs	0

Hier sehen Sie eine Auflistung der angeschlossenen Ein-/Ausgänge

4.5.2.2. Digitale Eingänge

Infohotline
+(49) 0 22 32 50 40 80

DEDITEC
GMBH

Configuration

In-/Outputs

Custom

Logout

Product name: RO-ETH (DEDITEC RO-ETH-Module)

User: admin / Session ends in: 355

General

Digital I/O

Digital Inputs

Digital Inputs Counter

Digital Outputs

Analog I/O

Configuration

Digital Inputs

I/O Access to the module: ☐ Activity

Status from modul: OK

Select channel area: CH 0..15

CH	Name	State
0	Digital Input 1	OFF ON
1	Digital Input 2	OFF ON
2	Digital Input 3	OFF ON
3	Digital Input 4	OFF ON
4	Digital Input 5	OFF ON
5	Digital Input 6	OFF ON
6	Digital Input 7	OFF ON
7	Digital Input 8	OFF ON
8	Digital Input 9	OFF ON
9	Digital Input 10	OFF ON
10	Digital Input 11	OFF ON
11	Digital Input 12	OFF ON
12	Digital Input 13	OFF ON
13	Digital Input 14	OFF ON
14	Digital Input 15	OFF ON
15	Digital Input 16	OFF ON

EDIT CH NAMES

Auf dieser Seite werden in regelmäßigen Abständen die Eingänge des Moduls gelesen.

Select channel area

Hier kann der angezeigte Kanal-Bereich (16er Blöcke) festgelegt werden.

State

Zustand der einzelnen Eingänge

Edit CH names

Zur besseren Übersicht, kann hier jedem Kanal ein Name zugeordnet werden.

4.5.2.3. Digitale Eingänge Zähler

Infohotline
+(49) 0 22 32 50 40 80

DEDITEC
GMBH

Configuration In-/Outputs Custom Logout

Product name: RO-ETH (DEDITEC RO-ETH-Module) User: admin / Session ends in: 336

General
Digital I/O
Digital Inputs
Digital Inputs Counter
Digital Outputs
Analog I/O
Configuration

Digital Inputs Counter

IO Access to the module: ☒ Activity
Status from modul: OK

Select channel area: CH 0..15

CH	Name	Counter value
0	Digital Input 1	1
1	Digital Input 2	1
2	Digital Input 3	1
3	Digital Input 4	0
4	Digital Input 5	1
5	Digital Input 6	1
6	Digital Input 7	1
7	Digital Input 8	0
8	Digital Input 9	0
9	Digital Input 10	0
10	Digital Input 11	0
11	Digital Input 12	0
12	Digital Input 13	0
13	Digital Input 14	0
14	Digital Input 15	0
15	Digital Input 16	0

EDIT CH NAMES RESET

Auf dieser Seite werden in regelmäßigen Abständen die Eingangszähler des Moduls gelesen.

Select channel area

Hier kann der angezeigte Kanal-Bereich (16er Blöcke) festgelegt werden.

Counter value

Aktueller Stand der Eingangszähler

Edit CH names

Zur besseren Übersicht, kann hier jedem Kanal ein Name zugeordnet werden.

Reset

Resettet alle Eingangszähler des aktuellen Kanal-Bereichs

4.5.2.4. Digitale Ausgänge

Infob hotline
+(49) 0 22 32 50 40 80

DEDITEC GMBH

Configuration In-/Outputs Custom Logout

Product name: RO-ETH (DEDITEC RO-ETH-Module) User: admin / Session ends in: 295

General
Digital I/O
Digital Inputs
Digital Inputs Counter
Digital Outputs
Analog I/O
Configuration

Digital outputs

I/O Access to the module ☒ Activity
Status from modul: OK

Select channel area CH 0..15

CH	Name	State	Readback
0	Digital Output 1	OFF ON	OFF ON
1	Digital Output 2	OFF ON	OFF ON
2	Digital Output 3	OFF ON	OFF ON
3	Digital Output 4	OFF ON	OFF ON
4	Digital Output 5	OFF ON	OFF ON
5	Digital Output 6	OFF ON	OFF ON
6	Digital Output 7	OFF ON	OFF ON
7	Digital Output 8	OFF ON	OFF ON
8	Digital Output 9	OFF ON	OFF ON
9	Digital Output 10	OFF ON	OFF ON
10	Digital Output 11	OFF ON	OFF ON
11	Digital Output 12	OFF ON	OFF ON
12	Digital Output 13	OFF ON	OFF ON
13	Digital Output 14	OFF ON	OFF ON
14	Digital Output 15	OFF ON	OFF ON
15	Digital Output 16	OFF ON	OFF ON

EDIT CH NAMES ALL OFF ALL ON

Auf dieser Seite werden in regelmäßigen Abständen die Ausgänge des Moduls zurückgelesen. Zusätzlich kann jeder Kanal einzeln oder der aktuelle Kanal-Bereich ein-/ausgeschaltet werden.

Select channel area

Hier kann der angezeigte Kanal-Bereich (16er Blöcke) festgelegt werden.

State

Diesen Kanal oder alle Kanäle ein-/ausschalten

Readback

Aktueller Zustand des Ausgangs

Edit CH names

Zur besseren Übersicht, kann hier jedem Kanal ein Name zugeordnet werden.

4.5.2.5. Analoge Eingänge

The screenshot shows the DEDITEC web interface for the RO-ETH module. The top header includes the company name 'DEDITEC GMBH' and contact information '+49) 0 22 32 50 40 80'. The navigation bar has tabs for 'Configuration', 'In-/Outputs', 'Custom', and 'Logout'. The main content area is titled 'Product name: RO-ETH (DEDITEC RO-ETH-Module)' and 'User: admin / Session ends in: 212'. On the left, a sidebar menu lists 'General', 'Digital I/O', 'Analog I/O', 'Analog Inputs', 'Analog Outputs', and 'Configuration'. The 'Analog Inputs' section is active, showing a table of 16 channels. Above the table, there are settings for 'I/O Access to the module' (set to 'Activity'), 'Status from modul' (set to 'OK'), 'Select channel area' (set to 'CH 0..15'), and 'A/D Mode' (set to '0..10 V'). At the bottom of the table is a button labeled 'EDIT CH NAMES'.

CH	Name	Value
0	Analog Input 1	0.000 V
1	Analog Input 2	0.000 V
2	Analog Input 3	0.000 V
3	Analog Input 4	0.000 V
4	Analog Input 5	0.000 V
5	Analog Input 6	0.000 V
6	Analog Input 7	0.000 V
7	Analog Input 8	0.000 V
8	Analog Input 9	0.000 V
9	Analog Input 10	0.000 V
10	Analog Input 11	0.000 V
11	Analog Input 12	0.000 V
12	Analog Input 13	0.000 V
13	Analog Input 14	0.000 V
14	Analog Input 15	0.000 V
15	Analog Input 16	0.000 V

Auf dieser Seite werden in regelmäßigen Abständen die analogen Eingänge des Moduls gelesen.

Select channel area

Hier kann der angezeigte Kanal-Bereich (16er Blöcke) festgelegt werden.

A/D Mode

Aktueller A/D-Modus des Kanal-Bereichs

Edit CH names

Zur besseren Übersicht, kann hier jedem Kanal ein Name zugeordnet werden.

Value

Aktueller Analog Wert

4.5.2.6. Analoge Ausgänge

Infoc hotline
+(49) 0 22 32 50 40 80

DEDITEC
GMBH

Configuration In-/Outputs Custom Logout

Product name: RO-ETH (DEDITEC RO-ETH-Module) User: admin / Session ends in: 239

General
Digital I/O
Analog I/O
Analog Inputs
Analog Outputs
Configuration

Analog Outputs

I/O Access to the module ☒ Activity
Status from modul: OK

Select channel area: CH 0..3

D/A Mode: +/-10 V +/-10 V

CH	Name	Value	Set value	Readback
0	Analog Output 1	1.23 V	SET	1.230 V
1	Analog Output 2	4.56 V	SET	4.560 V
2	Analog Output 3	7.89 V	SET	7.890 V
3	Analog Output 4	9.87 V	SET	9.870 V

EDIT CH NAMES

Auf dieser Seite können analoge Ausgänge gesetzt werden.

Select channel area

Hier kann der angezeigte Kanal-Bereich (16er Blöcke) festgelegt werden.

D/A Mode

Aktueller D/A-Modus des Kanal-Bereichs

Edit CH names

Zur besseren Übersicht, kann hier jedem Kanal ein Name zugeordnet werden.

Value

Analog Wert, der ausgegeben werden soll.

Readback

Aktueller Analog Wert

4.5.2.7. Konfiguration

Infohotline
+(49) 0 22 32 50 40 80

DEDITEC GMBH

Configuration In-/Outputs Custom Logout

Product name: RO-ETH (DEDITEC RO-ETH) User: admin / Session ends in: 1225

General
Digital I/O
Analog I/O
Temperature
Stepper Motor
Configuration

Configuration

Status from modul: OK

Select I/O type: Digital Input
Select channel area: CH 0..7

CH	Name
0	Digital Input 01
1	Digital Input 02
2	Digital Input 03
3	Digital Input 04
4	Digital Input 05
5	Digital Input 06
6	Digital Input 07
7	Digital Input 08

SAVE

Auf dieser Seite können sämtliche Namen der angeschlossenen I/O-Einheiten bearbeitet werden.

Select I/O type

Hierüber kann der I/O-Typ ausgewählt werden.

Select channel area

Sind mehr Ein-/Ausgänge angeschlossen, als auf dieser Form dargestellt werden kann, kann hierüber der Kanalbereich ausgewählt werden.

Save

Hiermit werden die Namen für diesen Kanalbereich gespeichert.

Hinweis:

Der Kanalname darf maximal 16 Zeichen lang sein.

Verzeichnisstruktur der DELIB



5. Verzeichnisstruktur der DELIB

Nach erfolgreicher Installation liegt folgender Verzeichnisbaum vor:

\$DELIB_DIR

- |
- | - > include   Includes für Programmiersprachen
- | - > lib   Library
- | - > lib\bc   Borland Compiler Library
- | - > prod_pics   Produktbilder
- | - > programs   Modul-Testprogramme
- + - > USB-Driver   Treiber für USB-Module

Bitte beachten Sie, dass der "\$DELIB_DIR" Ordner, je nach Betriebssystem und DELIB-Version, variieren kann.

DELIB Installation	Windows Installation	Pfad
32 Bit	32 Bit	C:\Programme\DEDITEC\DELIB\
32 Bit	64 Bit	C:\Programme (x86)\DEDITEC\DELIB\
64 Bit	64 Bit	C:\Programme\DEDITEC\DELIB64\

Zudem werden im Windows System Ordner folgende Dateien installiert:

\$SYSDIR\delib.dll, bzw. \$SYSDIR\delib64.dll (32 Bit, bzw. 64 Bit DELIB Version)

\$SYSDIR\delibJNI.dll, bzw. \$SYSDIR\delibJNI64.dll (32 Bit, bzw. 64 Bit DELIB Version)

\$SYSDIR\ftbusui.dll

\$SYSDIR\ftd2xx.dll

\$SYSDIR\FTLang.dll

\$SYSDIR\drivers\ftdibus.sys

Bitte beachten Sie, dass der "\$SYSDIR" Ordner, je nach Betriebssystem und DELIB-Version, variieren kann.

DELIB Installation	Windows Installation	Pfad
32 Bit	32 Bit	C:\Windows\System32
32 Bit	64 Bit	C:\Windows\SysWOW64
64 Bit	64 Bit	C:\Windows\System32

5.1. Include Verzeichnis

Das für die DELIB angelegte Include-Verzeichnis enthält die Dateien, welche die entsprechenden Library-Funktionen beschreiben. Diese sind für die Programmiersprachen C (.h), Delphi (.pas) und Visual Basic (.bas) gegeben.

5.2. Library-Verzeichnis

Hierin befindet sich die Datei "DELIB.lib". Sie dient als Bindeglied für das Compilieren von eigenen Programmen, die die "DELIB.dll" benutzen.

5.3. Library-Verzeichnis für Borland

Für Borland Compiler gibt es eine separate DELIB.lib, die sich im Unterverzeichnis "bc" befindet. Diese dient ebenfalls als Bindeglied für das Compilieren von eigenen Programmen, die die "DELIB.dll" benutzen.

5.4. Umgebungsvariablen

Zwei Umgebungsvariablen weisen auf wichtige Verzeichnisse hin, die Dateien für die Programmiersprachen C, Delphi und Visual Basic enthalten.

"DELIB_INCLUDE" zeigt auf das Include-Verzeichnis.

%DELIB_INCLUDE% à c:\Programme\DEDITEC\DELIB\include"

"DELIB_LIB" zeigt auf das Library-Verzeichnis.

%DELIB_LIB% à c:\ Programme\DEDITEC\DELIB\lib

DELIB API Referenz



6. DELIB API Referenz

6.1. Verwaltungsfunktionen

6.1.1. DapiOpenModule

Beschreibung

Diese Funktion öffnet ein bestimmtes Modul.

Definition

ULONG DapiOpenModule(ULONG moduleID, ULONG nr);

Parameter

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

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

nr=0 → 1. Modul

nr=1 → 2. Modul

Return-Wert

handle=Entsprechender Handle für das Modul

handle=0 → Modul wurde nicht gefunden

Bemerkung

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

Programmierbeispiel

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

6.1.2. DapiCloseModule

Beschreibung

Dieser Befehl schließt ein geöffnetes Modul.

Definition

ULONG DapiCloseModule(ULONG handle);

Parameter

handle=Dies ist das Handle eines geöffneten Moduls.

Return-Wert

Keiner

Programmierbeispiel

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

6.1.3. DapiGetDELIBVersion

Beschreibung

Diese Funktion gibt die installierte DELIB-Version zurück.

Definition

ULONG DapiGetDELIBVersion(ULONG mode, ULONG par);

Parameters

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

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

Return-Wert

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

Programmierbeispiel

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

6.1.4. DapiSpecialCMDGetModuleConfig

Beschreibung

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

Definition

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

Parameter

handle=Dies ist der handle eines offenen Moduls

Querying the number of digital input channels

par=DAPI_SPECIAL_GET_MODULE_CONFIG_PAR_DI

Query number of digital input flip-flops

par=DAPI_SPECIAL_GET_MODULE_CONFIG_PAR_DI_FF

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

par=DAPI_SPECIAL_GET_MODULE_CONFIG_PAR_DI_COUNTER

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

par=DAPI_SPECIAL_GET_MODULE_CONFIG_PAR_CNT48

Querying the number of digital output channels

par=DAPI_SPECIAL_GET_MODULE_CONFIG_PAR_DO

Querying the number of digital pulse generator outputs

par=DAPI_SPECIAL_GET_MODULE_CONFIG_PAR_PULSE_GEN

Querying the number of digital PWM outputs

par=DAPI_SPECIAL_GET_MODULE_CONFIG_PAR_PWM_OUT

Querying the number of digital input/output channels

par=DAPI_SPECIAL_GET_MODULE_CONFIG_PAR_DX

Querying the number of analog input channels

par=DAPI_SPECIAL_GET_MODULE_CONFIG_PAR_AD

Querying the number of analog output channels

par=DAPI_SPECIAL_GET_MODULE_CONFIG_PAR_DA

Query number of temperature channels

par=DAPI_SPECIAL_GET_MODULE_CONFIG_PAR_TEMP

Query number of stepper channels

par=DAPI_SPECIAL_GET_MODULE_CONFIG_PAR_STEPPER

Return value

Querying the number of digital input channels

return=number of digital input channels

Query number of digital input flip-flops

return=number of digital input flip-flops

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

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

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

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

Querying the number of digital output channels

return=number of digital output channels

Querying the number of digital pulse generator outputs

return=number of digital pulse generator outputs

Querying the number of digital PWM outputs

return=number of digital PWM outputs

Querying the number of digital input/output channels

return=number of digital input/output channels

Querying the number of analog input channels

return=number of analog input channels

Querying the number of analog output channels

return=number of analog output channels

Query number of temperature channels

return=number of temperature channels

Query number of stepper channels

return=number of stepper channels

Programmierbeispiele

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

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

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

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

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

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

6.1.5. DapiOpenModuleEx

Beschreibung

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

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

Definition

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

Parameter

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

nr = Gibt an, welches 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);
```


6.1.6. DapiScanAllModulesAvailable

Beschreibung

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

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

Definition

ULONG DapiScanAllModulesAvailable(uint nr)

Parameter

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

nr = i: Auslesen der einzelnen angeschlossenen Module

Return-Wert

Gibt die Anzahl der gefunden Module wieder.

Programmierbeispiel

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

6.2. Fehlerbehandlung

6.2.1. DapiGetLastError

Beschreibung

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

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

Definition

```
ULONG DapiGetLastError();
```

Parameter

Keine

Return-Wert

Fehler Code

0=kein Fehler. (siehe delib_error_codes.h)

Programmierbeispiel

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

6.2.2. 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;
}
```

6.2.3. 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;
}
```

6.2.4. 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;
}
```

6.2.5. 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;
}
```

6.3. Digitale Eingänge lesen

6.3.1. DapiDIGet1

Beschreibung

Dieser Befehl liest einen einzelnen digitalen Eingang.

Definition

ULONG DapiDIGet1(ULONG handle, ULONG ch);

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

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

Return-Wert

Zustand des Eingangs (0/1)

6.3.2. DapiDIGet8

Beschreibung

Dieser Befehl liest gleichzeitig 8 digitale Eingänge.

Definition

ULONG DapiDIGet8(ULONG handle, ULONG ch);

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

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

Return-Wert

Zustand der gelesen Eingänge

6.3.3. DapiDIGet16

Beschreibung

Dieser Befehl liest gleichzeitig 16 digitale Eingänge.

Definition

ULONG DapiDIGet16(ULONG handle, ULONG ch);

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

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

Return-Wert

Zustand der gelesenen Eingänge

6.3.4. DapiDIGet32

Beschreibung

Dieser Befehl liest gleichzeitig 32 digitale Eingänge.

Definition

ULONG DapiDIGet32(ULONG handle, ULONG ch);

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

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

Return-Wert

Zustand der gelesenen Eingänge

Programmierbeispiel

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

6.3.5. DapiDIGet64

Beschreibung

Dieser Befehl liest gleichzeitig 64 digitale Eingänge.

Definition

ULONG DapiDIGet64(ULONG handle, ULONG ch);

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

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

Return-Wert

Zustand der gelesenen Eingänge

6.3.6. DapiDIGetFF32

Beschreibung

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

Definition

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

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

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

Return-Wert

Zustand von 32 Eingangszustandsänderungen

6.3.7. DapiDIGetCounter

Beschreibung

Dieser Befehl liest den Eingangszähler eines digitalen Eingangs.

Definition

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

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

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

mode=0 (Normale Zählfunktion)

mode=DAPI_CNT_MODE_READ_WITH_RESET (Zähler auslesen und direktes Counter resettet)

mode=DAPI_CNT_MODE_READ_LATCHED (Auslesen des gespeicherten Zählerstandes)

Return-Wert

Angabe des Zählerwertes

Programmierbeispiel

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

6.3.8. DapiSpecialCounterLatchAll

Beschreibung

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

So können anschließend alle Zählerstände des Latches nacheinander ausgelesen werden.

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

Definition

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

Parameter

Keine

Return-Wert

Keiner

Bemerkung

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

Programmierbeispiel

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

6.3.9. 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);
```

6.3.10. DapiSpecialDIFilterValueSet

Beschreibung

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

Definition

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

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

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

Bemerkung

Standardwert: 0ms

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

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

Programmierbeispiel

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

6.3.11. DapiSpecialDIFilterValueGet

Beschreibung

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

Definition

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

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

Return-Wert

Zeit [ms]

Bemerkung

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

DELIB Übersichtstabelle entnehmen.

Programmierbeispiel

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


6.3.12. 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
```

6.3.13. 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.
```

6.4. Digitale Ausgänge verwalten

6.4.1. DapiDOSet1

Beschreibung

Dieser Befehl setzt einen einzelnen Ausgang.

Definition

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

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

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

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

Return-Wert

Keiner

6.4.2. DapiDOSet8

Beschreibung

Dieser Befehl setzt gleichzeitig 8 digitale Ausgänge.

Definition

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

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

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

data=Gibt die Datenwerte an, die geschrieben werden

Return-Wert

Keiner

6.4.3. DapiDOSet16

Beschreibung

Dieser Befehl setzt gleichzeitig 16 digitale Ausgänge.

Definition

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

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

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

data=Gibt die Datenwerte an, die geschrieben werden

Return-Wert

Keiner

6.4.4. DapiDOSet32

Beschreibung

Dieser Befehl setzt gleichzeitig 32 digitale Ausgänge.

Definition

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

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

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

data=Gibt die Datenwerte an, die geschrieben werden

Return-Wert

Keiner

Programmierbeispiel

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

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

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

6.4.5. DapiDOSet64

Beschreibung

Dieser Befehl setzt gleichzeitig 64 digitale Ausgänge.

Definition

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

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

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

data=Gibt die Datenwerte an, die geschrieben werden

Return-Wert

Keiner

6.4.6. DapiDOSet1_WithTimer

Beschreibung

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

Definition

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

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

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

data=Gibt die Datenwerte an, die geschrieben werden

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

Return-Wert

Keiner

Bemerkung:

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

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

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

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

DELIB Übersichtstabelle entnehmen.

Programmierbeispiel

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

6.4.7. DapiDOReadback32

Beschreibung

Dieser Befehl liest die 32 digitalen Ausgänge zurück.

Definition

ULONG DapiDOReadback32(ULONG handle, ULONG ch);

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

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

Return-Wert

Zustand von 32 Ausgängen.

6.4.8. DapiDOReadback64

Beschreibung

Dieser Befehl liest die 64 digitalen Ausgänge zurück.

Definition

ULONG DapiDOReadback64(ULONG handle, ULONG ch);

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

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

Return-Wert

Zustand von 64 Ausgängen.

6.4.9. DapiDOSetBit32

Beschreibung

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

Definition

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

Parameter

handle = Dies ist das Handle eines geöffneten Moduls

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

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

Return-Wert

Keiner

Bemerkung:

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

Programmierbeispiel

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

6.4.10. DapiDOClrBit32

Beschreibung

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

Definition

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

Parameter

handle = Dies ist das Handle eines geöffneten Moduls

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

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

Return-Wert

Keiner

Bemerkung:

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

Programmierbeispiel

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

6.5. Digitale Zähler Funktionen

6.5.1. DapiCnt48ModeSet

Beschreibung

Dieser Befehl setzt einen Zählmodus (optional auch Submodus) und Eingangsfiler für einen bestimmten Eingangszählerkanal.

Definition

```
void DapiCnt48ModeSet(ULONG handle, ULONG ch, ULONG mode);
```

Parameter

handle = Dies ist das Handle eines geöffneten Moduls

ch = Nummer des Eingangszählerkanals, dessen Modus gesetzt werden soll (0, 1, 2, 3, ..)

mode = Gibt den Modus an

Übersicht verfügbare Zähler Modi

Mode	Wert[hex]	CNT8	CNT/IGR
DAPI_CNT48_MODE_COUNT_RISING_EDGE	0	X	X
DAPI_CNT48_MODE_COUNT_RISING_EDGE_X 2	1		X
DAPI_CNT48_MODE_COUNT_RISING_EDGE_X 4	2		X
DAPI_CNT48_MODE_T	D	X	
DAPI_CNT48_MODE_FREQUENCY	E	X	
DAPI_CNT48_MODE_PWM	F	X	

Submode	Wert[hex]	HW-Reset verfügbar
DAPI_CNT48_SUBMODE_NO_RESET	0	x

Submode	Wert[hex]	HW-Reset verfügbar
DAPI_CNT48_SUBMODE_RESET_WITH_READ	1	x
DAPI_CNT48_SUBMODE_NO_RESET	2	
DAPI_CNT48_SUBMODE_RESET_WITH_READ	3	

Mögliche Werte für mode

*mode=DAPI_CNT48_MODE_COUNT_RISING_EDGE |
DAPI_CNT48_SUBMODE_NO_RESET*

In diesem Modus wird bei steigender Flanke gezählt.

*mode=DAPI_CNT48_MODE_COUNT_RISING_EDGE |
DAPI_CNT48_SUBMODE_RESET_WITH_READ*

In diesem Modus wird bei steigender Flanke gezählt. Zusätzlich wird bei jedem Lesevorgang der Zähler resettet.

*mode=DAPI_CNT48_MODE_COUNT_RISING_EDGE |
DAPI_CNT48_SUBMODE_RESET_ON_CH_7*

In diesem Modus wird bei steigender Flanke gezählt. Zusätzlich kann der Zähler über ein externes Signal (letzter Kanal des Moduls = 1) resettet werden.

*mode=DAPI_CNT48_MODE_COUNT_RISING_EDGE |
DAPI_CNT48_SUBMODE_LATCH_COMMON*

Mit dem Befehl "**DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_CNT48, DAPI_SPECIAL_CNT48_LATCH_GROUP8, 0, 0)**" werden alle 8 Zählerstände der Eingangszähler gleichzeitig in einen Zwischenspeicher (Latch) geschrieben. Mit diesem Modus kann dann der gelatchte Zählerstand ausgelesen werden.

mode=DAPI_CNT48_MODE_T

Mit diesem Modus wird die Periodendauer T gemessen. Als Basis hierbei dient ein 100 MHz Zähler.

mode=DAPI_CNT48_MODE_FREQUENCY

Bei diesem Modus lässt sich die Anzahl der steigenden Flanken innerhalb einer Sekunde (=Frequenz) messen.

mode=DAPI_CNT48_MODE_PWM

Mit diesem Modus werden die "high" und "low" Zeit eines Signals gemessen. Dadurch kann anschließend das Verhältnis bestimmt werden (PWM).

Zusätzlich können alle Eingangszähler mit einem Eingangsfiler (mit einer oder-Verknüpfung) kombiniert werden. Hierzu stehen folgende Eingangsfiler zur Verfügung:

DAPI_CNT48_FILTER_20ns

DAPI_CNT48_FILTER_100ns

DAPI_CNT48_FILTER_250ns

DAPI_CNT48_FILTER_500ns

DAPI_CNT48_FILTER_1us

DAPI_CNT48_FILTER_2_5us

DAPI_CNT48_FILTER_5us

DAPI_CNT48_FILTER_10us

DAPI_CNT48_FILTER_25us

DAPI_CNT48_FILTER_50us

DAPI_CNT48_FILTER_100us

DAPI_CNT48_FILTER_250us

DAPI_CNT48_FILTER_500us

DAPI_CNT48_FILTER_1ms

DAPI_CNT48_FILTER_2_5ms

DAPI_CNT48_FILTER_5ms

Return-Wert

Keiner

Bemerkung

Dieser Befehl wird nur von unserem Modul RO-CNT8 unterstützt.

Programmierbeispiel

```
DapiCnt48ModeSet(handle, 0, DAPI_CNT48_MODE_COUNT_RISING_EDGE |  
DAPI_CNT48_SUBMODE_RESET_WITH_READ | DAPI_CNT48_FILTER_20ns);  
//Eingangszählerkanal 0 zählt alle Impulse <= 20ns bei steigender Flanke.  
Zusätzlich wird der Zähler nach Abfrage resettet.  
DapiCnt48ModeSet(handle, 1, DAPI_CNT48_MODE_COUNT_RISING_EDGE |  
DAPI_CNT48_SUBMODE_RESET_ON_CH_7 | DAPI_CNT48_FILTER_500us);  
//Eingangszählerkanal 1 zählt alle Impulse <= 500us bei steigender Flanke.  
Dieser Zähler kann mit einem externen Signal (ch7 = 1) resettet werden.  
DapiCnt48ModeSet(handle, 2, DAPI_CNT48_MODE_PWM |  
DAPI_CNT48_FILTER_5ms);  
//Eingangszählerkanal 2 misst alle low-/high Zeiten <= 5ms. Anschließend  
wird das Verhältnis bestimmt (PWM).
```


6.5.2. DapiCnt48ModeGet

Beschreibung

Dieser Befehl liest den Zählmodus eines bestimmten Eingangszählerkanals zurück.

Definition

ULONG DapiCnt48ModeGet(ULONG handle, ULONG ch);

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

ch=Nummer des Eingangszählerkanals, dessen Modus ausgegeben werden soll
(0, 1, 2, 3, ..)

Return-Wert

Zählmodus des Eingangszählerkanals.

(Nähere Informationen / Beschreibung der Bits -> siehe delib.h oder Manual "RO-Registerbelegung")

Bemerkung

Dieser Befehl wird nur von unserem Modul RO-CNT8 unterstützt.

Programmierbeispiel

```
value = DapiCnt48ModeGet(handle, 0)
//Gibt den Zählmodus von Eingangszählerkanal 0 zurück
value = DapiCnt48ModeGet(handle, 3)
//Gibt den Zählmodus von Eingangszählerkanal 3 zurück
```

6.5.3. DapiCnt48CounterGet32

Beschreibung

Dieser Befehl liest die ersten 32 Bit eines 48 Bit Eingangszählers.

Definition

ULONG DapiCnt48CounterGet32(ULONG handle, ULONG ch);

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

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

Return-Wert

Ausgabe des Zählerwertes.

Bemerkung

Dieser Befehl wird nur von unseren Modulen RO-CNT8 und RO-CNT/IGR unterstützt.

Programmierbeispiel

```
value = DapiCnt48CounterGet32(handle, 0);  
//gibt den Wert von Eingangszählerkanal 0 aus  
value = DapiCnt48CounterGet32(handle, 3);  
//gibt den Wert von Eingangszählerkanal 3 aus
```

6.5.4. DapiCnt48CounterGet48

Beschreibung

Dieser Befehl liest einen 48 Bit Zähler eines Eingangszählerkanals.

Definition

ULONGLONG DapiCnt48CounterGet48(ULONG handle, ULONG ch);

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

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

Return-Wert

Ausgabe des Zählerwertes.

Bemerkung

Dieser Befehl wird nur von unseren Modulen RO-CNT8 und RO-CNT/IGR unterstützt.

Programmierbeispiel

```
value = DapiCnt48CounterGet48(handle, 0);  
//gibt den Wert von Eingangszählerkanal 0 aus  
value = DapiCnt48CounterGet48(handle, 3);  
//gibt den Wert von Eingangszählerkanal 3 aus
```

6.5.5. DapiSpecialCNT48ResetSingle

Beschreibung

Dieser Befehl resettet den Zählerstand eines einzelnen Eingangszählers.

Definition

```
void DapiSpecialCommand(ULONG handle, DAPI_SPECIAL_CMD_CNT48,  
DAPI_SPECIAL_CNT48_RESET_SINGLE, ULONG ch, 0)
```

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

ch=Gibt die Nummer des Eingangszählers an, dessen Zählerstand resettet werden soll (0, 1, 2, ..)

Return-Wert

Keiner

Bemerkung

Dieser Befehl wird nur von unserem Modul RO-CNT8 unterstützt.

Programmierbeispiel

```
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_CNT48,  
DAPI_SPECIAL_CNT48_RESET_SINGLE, 0, 0)  
// Zählerstand von Eingangszähler 0 wird resettet  
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_CNT48,  
DAPI_SPECIAL_CNT48_RESET_SINGLE, 1, 0)  
// Zählerstand von Eingangszähler 1 wird resettet
```

6.5.6. DapiSpecialCNT48ResetGroup8

Beschreibung

Dieser Befehl resettet gleichzeitig die Zählerstände von 8 Eingangszählern.

Definition

```
void DapiSpecialCommand(ULONG handle, DAPI_SPECIAL_CMD_CNT48,  
DAPI_SPECIAL_CNT48_RESET_GROUP8, ULONG ch, 0)
```

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

ch=Gibt die Nummer des Eingangszählers an, ab dem die Zählerstände von 8 Eingangszählern resettet werden (0, 8, 16, ...)

Return-Wert

Keiner

Bemerkung

Dieser Befehl wird nur von unserem Modul RO-CNT8 unterstützt.

Programmierbeispiel

```
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_CNT48,  
DAPI_SPECIAL_CNT48_RESET_GROUP8, 0, 0)  
// Zählerstände der Eingangszähler 0-7 werden resettet  
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_CNT48,  
DAPI_SPECIAL_CNT48_RESET_GROUP8, 8, 0)  
// Zählerstände der Eingangszähler 8-15 werden resettet
```

6.5.7. DapiSpecialCNT48LatchGroup8

Beschreibung

Dieser Befehl speichert die Zählerstände von 8 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 (langsam) einzeln nacheinander ausgelesen werden können.

Definition

```
void DapiSpecialCommand(ULONG handle, DAPI_SPECIAL_CMD_CNT48,  
DAPI_SPECIAL_CNT48_LATCH_GROUP8, ULONG ch, 0)
```

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

ch=Gibt die Nummer des Eingangszählers an, ab dem die Zählerstände von 8 Eingangszählern gelatched werden (0, 8, 16, ...)

Return-Wert

Keiner

Bemerkung

Dieser Befehl wird nur von unserem Modul RO-CNT8 unterstützt.

Hinweis

Bitte beachten Sie, dass nur die Zählerstände der Eingangszähler gelatched werden,

bei denen zuvor der Modus "DAPI_CNT48_SUBMODE_LATCH_COMMON" gesetzt wurde. (-> **DapiCnt48ModeSet**)

Programmierbeispiel

```
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_CNT48,  
DAPI_SPECIAL_CNT48_LATCH_GROUP8, 0, 0)  
// Zählerstände der Eingangszähler 0-7 werden gelatched  
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_CNT48,  
DAPI_SPECIAL_CNT48_LATCH_GROUP8, 8, 0)  
// Zählerstände der Eingangszähler 8-15 werden gelatched
```

6.5.8. DapiSpecialCNT48DIGet1

Beschreibung

Dieser Befehl liest den Eingangszustand (0/1) eines digitalen Eingangszählerkanals.

Definition

ULONG DapiSpecialCommand(ULONG handle, DAPI_SPECIAL_CMD_CNT48, DAPI_SPECIAL_CNT48_DI_GET1, ULONG ch, 0)

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

ch=Gibt die Nummer des Eingangszählerkanals an, dessen Eingangszustand gelesen werden soll (0, 1, 2, 3, ..)

Return-Wert

Zustand des Eingangszählers (0/1)

Bemerkung

Dieser Befehl wird nur von unserem Modul RO-CNT8 unterstützt.

Programmierbeispiel

```
value = DapiSpecialCommand(handle,
DAPI_SPECIAL_CMD_CNT48, DAPI_SPECIAL_CNT48_DI_GET1, 0,
0)
// Liest Eingangszustand von Eingangszählerkanal 1
value = DapiSpecialCommand(handle,
DAPI_SPECIAL_CMD_CNT48, DAPI_SPECIAL_CNT48_DI_GET1, 1,
0)
// Liest Eingangszustand von Eingangszählerkanal 2
```

6.6. PWM Funktionen

6.6.1. DapiPWMOutSet

Beschreibung

Dieser Befehl setzt das PWM Verhältnis eines PWM-Kanals

Definition

```
void DapiPWMOutSet(ULONG handle, ULONG ch, float data);
```

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

ch=Gibt die Nummer des Ausgangs an, der gesetzt werden soll

data=PWM-Verhältnis in von 0% bis 100% in 1% Schritten

Kleinstes PWM-Verhältnis ist abhängig von der PWM-Frequenz

10Hz data muss $\geq 0\%$ sein

100Hz data muss $\geq 2\%$ sein

250Hz data muss $\geq 3\%$ sein

1000Hz data muss $\geq 9\%$ sein

Return-Wert

Keiner

Programmierbeispiel

```
DapiPWMOutSet(handle, 0, 50);  
// Setzt das PWM Verhältnis des ersten Kanals auf 50% (50% high, 50% low)  
DapiPWMOutSet(handle, 1, 100);  
// Setzt das PWM Verhältnis des zweiten Kanals auf 100% (100% high, 0%  
low)
```

6.6.2. DapiPWMOutReadback

Beschreibung

Dieser Befehl liest das PWM-Verhältnis eines PWM-Kanals

Definition

float DapiPWMOutReadback(ULONG handle, ULONG ch);

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

ch=Gibt die Nummer des Ausgangs an, der gelesen werden soll

Return-Wert

PWM Verhältnis des Kanals von 0% bis 100%

Programmierbeispiel

```
float data = DapiPWMOutReadback(handle, 0);  
// Liest das PWM Verhältnis des ersten Kanals  
float data = DapiPWMOutReadback(handle, 2);  
// Liest das PWM Verhältnis des zweiten Kanals
```

6.6.3. DAPI_SPECIAL_PWM_FREQ_SET

Beschreibung

Dieser Befehl setzt die PWM Frequenz des Moduls

Definition

```
void DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_PWM, cmd, par1,  
par2);
```

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

cmd=DAPI_SPECIAL_PWM_FREQ_SET

par1=channel area 0 (ch 0-15), 16 (ch 16-31) ... usw.

par2=Frequenz = DAPI_PWM_FREQUENCY_10HZ,
DAPI_PWM_FREQUENCY_100HZ, DAPI_PWM_FREQUENCY_250HZ oder
DAPI_PWM_FREQUENCY_1000Hz

Return-Wert

Keiner

Programmierbeispiel

```
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_PWM,  
DAPI_SPECIAL_PWM_FREQ_SET, 0,  
DAPI_PWM_FREQUENCY_100HZ);  
// Setzt die PWM Frequenz des Moduls auf 100Hz
```

6.6.4. DAPI_SPECIAL_PWM_FREQ_READBACK

Beschreibung

Dieser Befehl liest die aktuelle PWM Frequenz des Moduls

Definition

```
void DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_PWM, cmd, par1,  
par2);
```

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

cmd=DAPI_SPECIAL_PWM_FREQ_READBACK

par1=0

par2=0

Return-Wert

uint = DAPI_PWM_FREQUENCY_10HZ, DAPI_PWM_FREQUENCY_100HZ,
DAPI_PWM_FREQUENCY_250HZ oder DAPI_PWM_FREQUENCY_1000Hz

Programmierbeispiel

```
uint frequency = DapiSpecialCommand(handle,  
DAPI_SPECIAL_CMD_PWM, DAPI_SPECIAL_PWM_FREQ_READBACK,  
0, 0);  
// Liest die PWM Frequenz des Moduls
```

6.7. Pulsgenerator Ausgänge verwalten

6.7.1. DapiPulseGenSet

Beschreibung

Dieser Befehl generiert eine gewisse Anzahl an Impulsen mit vorgegebenen low- und high-Zeiten.

Er wird nur von unseren RO-CNT8 Modulen unterstützt.

Definition

```
void DapiPulseGenSet(ULONG handle, ULONG ch, ULONG mode, ULONG par0,  
    ULONG par1, ULONG par2);
```

Parameter

handle = Dies ist das Handle eines geöffneten Moduls

ch = Gibt die Nummer des zu setzenden Ausgangs an (0, 1, 2, ..)

mode = Modus, mit dem Impulse generiert werden (muss immer 0 sein).

par0 = Anzahl der zu erzeugenden Impulse (par0 = 0 -> unendlich viele Impulse)

par1 = Low-Zeit des Impulses (t[ns] / 10 - 1)

par2 = High-Zeit des Impulses (t[ns] / 10 - 1)

Beispiel für das Einstellen der low-/high-Zeit (Par1/Par2)

500ns -> $500 / 10 - 1 = 49(\text{dez})$

7µ -> $7000 / 10 - 1 = 699(\text{dez})$

2,5ms -> $2500000 / 10 - 1 = 249,999(\text{dez})$

Return-Wert

keiner

Programmierbeispiel

```
DapiPulseGenSet(handle, 0, 0, 10, 29, 69);  
// generiert 10 Impulse an Pulsgenerator-Ausgang 0 mit einer low-Zeit von  
300ns und einer high-Zeit von 700ns.  
DapiPulseGenSet(handle, 3, 0, 100, 7999, 9999);  
// generiert 100 Impulse an Pulsgenerator-Ausgang 3 mit einer low-Zeit von  
80µs und einer high-Zeit von 100µs.  
DapiPulseGenSet(handle, 10, 0, 0, 249999, 299999);
```

// generiert unendlich viele Impulse an Pulsgenerator-Ausgang 10 mit einer low-Zeit von 2,5ms und einer high-Zeit von 3ms.

6.8. A/D Wandler Funktionen

6.8.1. 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

6.8.2. 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

6.8.3. 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

6.8.4. DapiADGetVolt

Beschreibung

Dieser Befehl liest einen Datenwert von einen 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

6.8.5. DapiADGetmA

Beschreibung

Dieser Befehl liest einen Datenwert von einen 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.

6.8.6. 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.
```

6.9. D/A Ausgänge verwalten

6.9.1. 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

6.9.2. 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

6.9.3. 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)
```


6.9.4. 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
```

6.9.5. 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.

6.9.6. 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 Ausgangsspannung 0V.

DAPI_SPECIAL_DA_PAR_DA_SAVE_EEPROM_CONFIG

Mit diesem Befehl wird die aktuelle D/A Wandler Einstellung (Spannung/Strom-Wert, 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.
```

6.10. PT100 Funktionen

6.10.1. DapiTempGet

Beschreibung

Dieser Befehl liest einen Temperatur Eingang.

Definition

float DapiTempGet(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

Temperatur [°C]

Programmierbeispiel

```
ret=DapiTempGet (handle, 0)  
//gibt die Temperatur von Kanal 0 zurück
```

6.11. Schrittmotoren Funktionen

6.11.1. Befehle mit DapiStepperCommand

6.11.1.1. DAPI_STEPPEER_CMD_GO_POSITION

Beschreibung

Hiermit wird eine bestimmte Position angefahren. Dieses Kommando darf nur ausgeführt werden, wenn der Motor nicht "disabled" ist und kein Go_Position oder Go_Referenz ausgeführt wird.

Definition

DapiStepperCommand(handle, motor, DAPI_STEPPEER_CMD_GO_POSITION, position, 0, 0, 0);

Programmierbeispiel

```
DapiStepperCommand(handle, motor,  
DAPI_STEPPEER_CMD_GO_POSITION, go_pos_par, 0, 0, 0);
```

6.11.1.2. DAPI_STEPPER_CMD_GO_POSITION_RELATIVE

Beschreibung

Hiermit wird eine relative Position angefahren. Im Gegensatz zum Befehl GO_POSITION, der eine absolute Position anfährt, wird hier die momentane Position berücksichtigt. Dieses Kommando darf nur ausgeführt werden, wenn der Motor nicht "disabled" ist und kein Go_Position oder Go_Referenz ausgeführt wird.

Definition

```
void DapiStepperCommand(handle, motor,  
DAPI_STEPPER_CMD_GO_POSITION_RELATIVE, go_pos_rel_par, 0, 0, 0);
```

Programmierbeispiel

```
DapiStepperCommand(handle, motor,  
DAPI_STEPPER_CMD_GO_POSITION_RELATIVE, 100, 0, 0, 0);  
//Motor fährt, von der aktuellen Position aus gesehen, 100 Schritte nach  
rechts.
```

6.11.1.3. DAPI_STEPPER_CMD_SET_POSITION

Beschreibung

Dieses Kommando dient zum setzen der Motorposition. Die Auflösung beträgt 1/16 Vollschr. Dieses Kommando darf nur bei angehaltenem Motor verwendet werden.

Definition

```
DapiStepperCommand(handle, motor, DAPI_STEPPER_CMD_SET_POSITION, par1,  
0, 0, 0);
```

Parameter

par1 = Motorposition

6.11.1.4. DAPI_STEPPER_CMD_SET_FREQUENCY

Beschreibung

Dieses Kommando dient zur Einstellung der Motorsollfrequenz. Die Motorfrequenzregelung übernimmt dabei die Einhaltung der Beschleunigungs- / Bremsrampe. Schrittverluste treten nicht auf. Die Motorsollfrequenz ist bezogen auf Vollschrittbetrieb. Über das Vorzeichen wird die Richtung ausgewählt. Die Motorsollfrequenz darf nicht über der Maxfrequenz liegen, ansonsten wird das Kommando abgelehnt.

Bei geschlossenem Endschalter1 läßt sich nur in positive Richtung verfahren, bei geschlossenem Endschalter2 läßt sich nur in negative Richtung verfahren, ansonsten wird das Kommando abgelehnt.

Definition

DapiStepperCommand(handle, motor, DAPI_STEPPER_CMD_SET_FREQUENCY, par1, 0, 0, 0);

Parameter

par1 = Motorsollfrequenz [Hz]

6.11.1.5. DAPI_STEPPER_CMD_GET_FREQUENCY

Beschreibung

Dieses Kommando dient zum Abfragen der Motorfrequenz. Dieses Kommando darf immer verwendet werden.

Definition

```
DapiStepperCommand(handle, motor, DAPI_STEPPER_CMD_GET_FREQUENCY,  
par1, 0,0,0);
```

Return-Wert

Motorfrequenz [Hz]

6.11.1.6. DAPI_STEPPER_CMD_SET_FREQUENCY_DIRECTLY

Beschreibung

Dieses Kommando dient zur Einstellung der Motorfrequenz. Die Motorfrequenzregelung übernimmt dabei keine Funktion. Für die Einhaltung der Beschleunigungs- / Bremsrampe ist der Anwender verantwortlich. Schrittverluste können bei Nichteinhaltung auftreten.

Die Motorfrequenz ist bezogen auf Vollschrittbetrieb. Über das Vorzeichen wird die Richtung ausgewählt.

Die Frequenz darf nicht über der Maxfrequenz liegen.

Definition

```
DapiStepperCommand(handle, motor,  
DAPI_STEPPER_CMD_SET_FREQUENCY_DIRECTLY, par1, 0,0,0);
```

Parameter

par1 = Motorfrequenz [Hz]

6.11.1.7. DAPI_STEPPEER_CMD_STOP

Beschreibung

Dieses Kommando dient zum Anhalten des Motors, die Bremsrampe wird dabei eingehalten.

Definition

DapiStepperCommand(handle, motor, DAPI_STEPPEER_CMD_STOP, 0, 0, 0, 0);

6.11.1.8. DAPI_STEPPER_CMD_FULLSTOP

Beschreibung

Dieses Kommando dient zum sofortigen Anhalten des Motors, die Bremsrampe wird dabei nicht eingehalten. Die Motorposition kann vielleicht danach nicht mehr stimmen, da der Motor unkontrolliert angehalten wird.

Definition

DapiStepperCommand(handle, motor, DAPI_STEPPER_CMD_FULLSTOP, 0, 0, 0, 0);

Programmierbeispiel

```
DapiStepperCommand(handle, motor,  
DAPI_STEPPER_CMD_FULLSTOP, 0, 0, 0, 0);
```

6.11.1.9. DAPI_STEPPER_CMD_DISABLE

Beschreibung

Dieses Kommando dient zum deaktivieren/enaktivieren des Motors, der Motor verfährt dann nicht mehr/oder wieder. Dieses Kommando darf nur bei Motorstillstand benutzt werden.

Definition

```
DapiStepperCommand(handle, motor, DAPI_STEPPER_CMD_DISABLE, par1, 0, 0, 0);
```

Parameter

par1 = Disablemode (0=Normale Funktion / 1=Disable)

6.11.1.10. DAPI_STEPPER_CMD_SET_MOTORCHARACTERISTIC

Beschreibung

Hiermit werden neue Motor Konfigurationen gesetzt.

Definition

```
DapiStepperCommand(handle, motor,  
DAPI_STEPPER_CMD_SET_MOTORCHARACTERISTIC, par1, par2, 0, 0);
```

Parameter

Parameter-Stepmode setzen

par1=DAPI_STEPPER_MOTORCHAR_PAR_STEPMODE

par2=0 (Vollschrittbetrieb)

par2=1 (Halbschrittbetrieb)

par2=2 (Viertelschrittbetrieb)

par2=3 (Achtelschrittbetrieb)

par2=4 (Sechzehntelschrittbetrieb)

Parameter-GO-Frequency setzen

par1=DAPI_STEPPER_MOTORCHAR_PAR_GOFREQUENCY

par2=Geschwindigkeit [Vollschritt / s] - bezogen auf Vollschritt Frequenz -
(Maximalwert=5000)

Parameter-Start-Frequency setzen

par1=DAPI_STEPPER_MOTORCHAR_PAR_STARTFREQUENCY

par2=Startfrequenz [Vollschritt / s] - bezogen auf Vollschritt Frequenz -
(Maximalwert=5000)

Parameter-Stop-Frequency setzen

par1=DAPI_STEPPER_MOTORCHAR_PAR_STOPFREQUENCY

par2=Stopfrequenz [Vollschritt / s] - bezogen auf Vollschritt Frequenz -
(Maximalwert=5000)

Parameter-Max-Frequency setzen

par1=DAPI_STEPPER_MOTORCHAR_PAR_MAXFREQUENCY

par2=Maximale Frequenz [Vollschritt / s] - bezogen auf Vollschritt Frequenz - (Maximalwert=5000)

Parameter-Accelerationslope setzen

par1=DAPI_STEPPER_MOTORCHAR_PAR_ACCELERATIONSLOPE

par2=Beschleunigungsrampe [Vollschritt / 10ms] - (Maximalwert=1000)

Parameter-Decelerationslope setzen

par1=DAPI_STEPPER_MOTORCHAR_PAR_DECELERATIONSLOPE

par2= Bremsrampe [Vollschritt / 10ms] - (Maximalwert=1000)

Parameter-Phasecurrent setzen

par1=DAPI_STEPPER_MOTORCHAR_PAR_PHASECURRENT

par2=Phasenstrom [mA] - (Maximalwert = 1500)

Parameter-Hold-Phasecurrent setzen

par1=DAPI_STEPPER_MOTORCHAR_PAR_HOLDPHASECURRENT

par2=Phasenstrom bei Motorstillstand [mA] - (Maximalwert=1500)

Parameter-Hold-Time setzen

par1=DAPI_STEPPER_MOTORCHAR_PAR_HOLDTIME

par2=Zeit in der der Haltestrom fließt nach Motorstop [ms]

par2=-1 / FFFF hex / 65535 dez (Zeit unendlich)

Parameter-Status-LED-Mode setzen

par1=DAPI_STEPPER_MOTORCHAR_PAR_STATUSLEDMODE

par2=Betriebsart der Status-LED

par2=0 = (MOVE - LED leuchtet bei Motorbewegung)

par2=1 = (HALT - LED leuchtet bei Motorstillstand)

par2=2 = (ENDSW1 - LED leuchtet bei geschlossenen Endschanter1)

par2=3 = (ENDSW2 - LED leuchtet bei geschlossenen Endschanter2)

par2=4 = (REFSW1 - LED leuchtet bei geschlossenen Referenzschalterschalter1)

par2=5 = (REFSW2 - LED leuchtet bei geschlossenen Referenzschalterschalter2)

Parameter-Invert-END-Switch1 setzen

par1=DAPI_STEPPER_MOTORCHAR_PAR_INVERT_ENDSW1

par2=Invertiere Funktion des Endschalter1 (0=normal / 1=invertieren)

Parameter-Invert-END-Switch2 setzen

par1=DAPI_STEPPER_MOTORCHAR_PAR_INVERT_ENDSW2

par2=Invertiere Funktion des Endschalter2 (0=normal / 1=invertieren)

Parameter-Invert-Ref-Switch1 setzen

par1=DAPI_STEPPER_MOTORCHAR_PAR_INVERT_REFSW1

par2=Invertiere Funktion des Referenzschalterschalter1 (0=normal / 1=invertieren)

Parameter-Invert-Ref-Switch2 setzen

par1=DAPI_STEPPER_MOTORCHAR_PAR_INVERT_REFSW2

par2=Invertiere Funktion des Referenzschalterschalter2 (0=normal / 1=invertieren)

Parameter-Invert-direction setzen

par1=DAPI_STEPPER_MOTORCHAR_PAR_INVERT_DIRECTION

par2=Invertiere alle Richtungsangaben (0=normal / 1=invertieren)

Parameter-Endswitch-Stopmode setzen

par1=DAPI_STEPPER_MOTORCHAR_PAR_ENDSWITCH_STOPMODE

par2=Einstellen des Stopverhaltens (0=Fullstop / 1=Stop)

Parameter-GoReferenceFrequency setzen

(ACHTUNG: Dieser Parameter wird nicht mehr unterstützt!)

par1=DAPI_STEPPER_MOTORCHAR_PAR_GOREFERENCEFREQUENCY

Bemerkung

Dieser Parameter wird durch die nachfolgenden drei Parametern vollständig ersetzt.

Parameter-GoReferenceFrequencyToEndSwitch setzen

par1=DAPI_STEPPER_MOTORCHAR_PAR_GOREFERENCEFREQUENCY_TOENDSWITCH

par2=Geschwindigkeit, mit der der Enshalter angefahren wird (Frequenz [Vollschritt / s] - (Maximalwert=5000))

Parameter GoReferenceFrequencyAfterEndSwitch setzen

par1=DAPI_STEPPER_MOTORCHAR_PAR_GOREFERENCEFREQUENCY_AFTERE
NDSWITCH

par2=Geschwindigkeit, mit der vom Enschanter abgefahren wird (Frequenz
[Vollschritt / s] - (Maximalwert=5000))

Parameter GoReferenceFrequencyToOffset setzen

par1=DAPI_STEPPER_MOTORCHAR_PAR_GOREFERENCEFREQUENCY_TOOFFS
ET

par2=Geschwindigkeit, mit der der optionale Offset angefahren wird (Frequenz
[Vollschritt / s] - (Maximalwert=5000))

Programmierbeispiel

```
DapiStepperCommand(handle, motor,  
DAPI_STEPPER_CMD_SET_MOTORCHARACTERISTIC,  
DAPI_STEPPER_MOTORCHAR_PAR_STEPMODE, 4,0,0);  
// Schrittmode (Voll-, Halb-, Viertel-, Achtel-, Sechszehntelschritt)  
  
DapiStepperCommand(handle, motor,  
DAPI_STEPPER_CMD_SET_MOTORCHARACTERISTIC,  
DAPI_STEPPER_MOTORCHAR_PAR_GOFREQUENCY, 1000,0,0);  
// Schrittmode (Voll-, Halb-, Viertel-, Achtel-, Sechszehntelschritt)  
  
DapiStepperCommand(handle, motor,  
DAPI_STEPPER_CMD_SET_MOTORCHARACTERISTIC,  
DAPI_STEPPER_MOTORCHAR_PAR_STARTFREQUENCY, 100,0,0);  
// Startfrequenz [Vollschritt / s]  
  
DapiStepperCommand(handle, motor,  
DAPI_STEPPER_CMD_SET_MOTORCHARACTERISTIC,  
DAPI_STEPPER_MOTORCHAR_PAR_STOPFREQUENCY, 100,0,0);  
// Stopfrequenz [Vollschritt / s]  
  
DapiStepperCommand(handle, motor,  
DAPI_STEPPER_CMD_SET_MOTORCHARACTERISTIC,  
DAPI_STEPPER_MOTORCHAR_PAR_MAXFREQUENCY, 3500,0,0);  
// maximale Frequenz [Vollschritt / s]  
  
DapiStepperCommand(handle, motor,  
DAPI_STEPPER_CMD_SET_MOTORCHARACTERISTIC,  
DAPI_STEPPER_MOTORCHAR_PAR_ACCELERATIONSLOPE, 20,0,0);
```

// Beschleunigung in [Vollschritten / ms]

```
DapiStepperCommand(handle, motor,  
DAPI_STEPPER_CMD_SET_MOTORCHARACTERISTIC,  
DAPI_STEPPER_MOTORCHAR_PAR_DECELERATIONSLOPE, 20,0,0);
```

// Bremsung in [Vollschritten / ms]

```
DapiStepperCommand(handle, motor,  
DAPI_STEPPER_CMD_SET_MOTORCHARACTERISTIC,  
DAPI_STEPPER_MOTORCHAR_PAR_PHASECURRENT, 750,0,0);
```

// Phasenstrom [mA]

```
DapiStepperCommand(handle, motor,  
DAPI_STEPPER_CMD_SET_MOTORCHARACTERISTIC,  
DAPI_STEPPER_MOTORCHAR_PAR_HOLDPHASECURRENT, 500,0,0);
```

// Phasenstrom bei Motorstillstand [mA]

```
DapiStepperCommand(handle, motor,  
DAPI_STEPPER_CMD_SET_MOTORCHARACTERISTIC,  
DAPI_STEPPER_MOTORCHAR_PAR_HOLDTIME, 15000,0,0);
```

// Zeit in der der Haltestrom fließt nach Motorstop [s]

```
DapiStepperCommand(handle, motor,  
DAPI_STEPPER_CMD_SET_MOTORCHARACTERISTIC,  
DAPI_STEPPER_MOTORCHAR_PAR_STATUSLEDMODE, 0,0,0);
```

// Betriebsart der Status-LED

```
DapiStepperCommand(handle, motor,  
DAPI_STEPPER_CMD_SET_MOTORCHARACTERISTIC,  
DAPI_STEPPER_MOTORCHAR_PAR_INVERT_ENDSW1, 0,0,0);
```

// invertiere Funktion des Endschalter1

```
DapiStepperCommand(handle, motor,  
DAPI_STEPPER_CMD_SET_MOTORCHARACTERISTIC,  
DAPI_STEPPER_MOTORCHAR_PAR_INVERT_ENDSW2, 0,0,0);
```

// invertiere Funktion des Endschalter2

```
DapiStepperCommand(handle, motor,  
DAPI_STEPPER_CMD_SET_MOTORCHARACTERISTIC,  
DAPI_STEPPER_MOTORCHAR_PAR_INVERT_REFSW1, 0,0,0);
```

// invertiere Funktion des Referenzschalters

```
DapiStepperCommand(handle, motor,  
DAPI_STEPPER_CMD_SET_MOTORCHARACTERISTIC,
```

```

DAPI_STEPPER_MOTORCHAR_PAR_INVERT_REFSW2, 0,0,0);
// invertiere Funktion des Referenzschalterschalter2

DapiStepperCommand(handle, motor,
DAPI_STEPPER_CMD_SET_MOTORCHARACTERISTIC,
DAPI_STEPPER_MOTORCHAR_PAR_INVERT_DIRECTION, 0,0,0);
// invertiere alle Richtungsangaben

DapiStepperCommand(handle, motor,
DAPI_STEPPER_CMD_SET_MOTORCHARACTERISTIC,
DAPI_STEPPER_MOTORCHAR_PAR_ENDSWITCH_STOPMODE, 0,0,0);
// einstellen des Stopverhaltens

DapiStepperCommand(handle, motor,
DAPI_STEPPER_CMD_SET_MOTORCHARACTERISTIC, DAPI_STEPPER_M
OTORCHAR_PAR_GOREFERENCEFREQUENCY_TOENDSWITCH,
100,0,0);
// Einstellung der Geschwindigkeit, mit der zum Endschalter angefahren wird.

DapiStepperCommand(handle, motor,
DAPI_STEPPER_CMD_SET_MOTORCHARACTERISTIC,
DAPI_STEPPER_MOTORCHAR_PAR_GOREFERENCEFREQUENCY_AFTEREN
DSWITCH , 200,0,0);
// Einstellung der Geschwindigkeit, mit der vom Endschalter abgefahren wird.

DapiStepperCommand(handle, motor,
DAPI_STEPPER_CMD_SET_MOTORCHARACTERISTIC,
DAPI_STEPPER_MOTORCHAR_PAR_GOREFERENCEFREQUENCY_TOOFFSE
T, 300,0,0);
// Einstellung der Geschwindigkeit, mit der zum optionalen Offset angefahren
wird.

```

6.11.1.11. DAPI_STEPPER_CMD_GET_MOTORCHARACTERISTIC

Beschreibung

Hiermit wird der Motorspezifische Parameter ausgelesen. Dieses Kommando darf immer benutzt werden. Es teilt sich in Unterkommandos auf, die analog den Parametern von DAPI_STEPPER_CMD_SET_MOTORCHARACTERISTIC sind.

Definition

```
DapiStepperCommand(handle, motor,  
DAPI_STEPPER_CMD_GET_MOTORCHARACTERISTIC, par1, 0, 0, 0);
```

Parameter

Parameter-Stepmode abfragen

```
par1=DAPI_STEPPER_MOTORCHAR_PAR_STEPMODE
```

Parameter-GO-Frequency abfragen

```
par1=DAPI_STEPPER_MOTORCHAR_PAR_GOFREQUENCY
```

Parameter-Start-Frequency abfragen

```
par1=DAPI_STEPPER_MOTORCHAR_PAR_STARTFREQUENCY
```

Parameter-Stop-Frequency abfragen

```
par1=DAPI_STEPPER_MOTORCHAR_PAR_STOPFREQUENCY
```

Parameter-Max-Frequency abfragen

```
par1=DAPI_STEPPER_MOTORCHAR_PAR_MAXFREQUENCY
```

Parameter-Accelerationslope abfragen

```
par1=DAPI_STEPPER_MOTORCHAR_PAR_ACCELERATIONSLOPE
```

Parameter-Decelerationslope abfragen

```
par1=DAPI_STEPPER_MOTORCHAR_PAR_DECELERATIONSLOPE
```

Parameter-Phasecurrent abfragen

```
par1=DAPI_STEPPER_MOTORCHAR_PAR_PHASECURRENT
```

Parameter-Hold-Phasecurrent abfragen

par1=DAPI_STEPPER_MOTORCHAR_PAR_HOLDPHASECURRENT

Parameter-Hold-Time abfragen

par1=DAPI_STEPPER_MOTORCHAR_PAR_HOLDTIME

Parameter-Status-LED-Mode abfragen

par1=DAPI_STEPPER_MOTORCHAR_PAR_STATUSLEDMODE

Parameter-Invert-END-Switch1 abfragen

par1=DAPI_STEPPER_MOTORCHAR_PAR_INVERT_ENDSW1

Parameter-Invert-END-Switch2 abfragen

par1=DAPI_STEPPER_MOTORCHAR_PAR_INVERT_ENDSW2

Parameter-Invert-Ref-Switch1 abfragen

par1=DAPI_STEPPER_MOTORCHAR_PAR_INVERT_REFSW1

Parameter-Invert-Ref-Switch2 abfragen

par1=DAPI_STEPPER_MOTORCHAR_PAR_INVERT_REFSW2

Parameter-Invert-direction abfragen

par1=DAPI_STEPPER_MOTORCHAR_PAR_INVERT_DIRECTION

Parameter-Endswitch-Stopmode abfragen

par1=DAPI_STEPPER_MOTORCHAR_PAR_ENDSWITCH_STOPMODE

Parameter-GoReferenceFrequency abfragen

(ACHTUNG: Dieser Parameter wird nicht mehr unterstützt!)

par1=DAPI_STEPPER_MOTORCHAR_PAR_GOREFERENCEFREQUENCY

Bemerkung:

Dieser Parameter wird durch die nachfolgenden drei Parametern vollständig ersetzt.

Parameter-GoReferenceFrequencyToEndSwitch abfragen

par1=DAPI_STEPPER_MOTORCHAR_PAR_GOREFERENCEFREQUENCY_TOENDSWITCH

Parameter GoReferenceFrequencyAfterEndSwitch abfragen

par1=DAPI_STEPPER_MOTORCHAR_PAR_GOREFERENCEFREQUENCY_AFTERE

NDSWITCH

Parameter GoReferenceFrequencyToOffSet abfragen

par1=DAPI_STEPPER_MOTORCHAR_PAR_GOREFERENCEFREQUENCY_TOOFFS
ET

Return-Wert

Parameter-Stepmode ablesen

par1=DAPI_STEPPER_MOTORCHAR_PAR_STEPMODE

return=0 (Vollschrittbetrieb)

return=1 (Halbschrittbetrieb)

return=2 (Viertelschrittbetrieb)

return=3 (Achtelschrittbetrieb)

return=4 (Sechzehntelschrittbetrieb)

Parameter-GO-Frequency

par1=DAPI_STEPPER_MOTORCHAR_PAR_GOFREQUENCY

return=Geschwindigkeit [Vollschritt / s] - bezogen auf Vollschritt

Parameter-Start-Frequency

par1=DAPI_STEPPER_MOTORCHAR_PAR_STARTFREQUENCY

return=Startfrequenz [Vollschritt / s]

Parameter-Stop-Frequency

par1=DAPI_STEPPER_MOTORCHAR_PAR_STOPFREQUENCY

return=Stopfrequenz [Vollschritt / s]

Parameter-Max-Frequency

par1=DAPI_STEPPER_MOTORCHAR_PAR_MAXFREQUENCY

return=maximale Frequenz [Vollschritt / s]

Parameter-Accelerationslope

par1=DAPI_STEPPER_MOTORCHAR_PAR_ACCELERATIONSLOPE

return=Beschleunigungsrampe [Vollschritten / ms]

Parameter-Decelerationslope

par1=DAPI_STEPPER_MOTORCHAR_PAR_DECELERATIONSLOPE

return= Bremsrampe [Vollschritten / ms]

Parameter-Phasecurrent

par1=DAPI_STEPPER_MOTORCHAR_PAR_PHASECURRENT

return=Phasenstrom [mA]

Parameter-Hold-Phasecurrent

par1=DAPI_STEPPER_MOTORCHAR_PAR_HOLDPHASECURRENT

return= Phasenstrom bei Motorstillstand [mA]

Parameter-Hold-Time

par1=DAPI_STEPPER_MOTORCHAR_PAR_HOLDTIME

return=Zeit in der der Haltestrom fließt nach Motorstop [ms]

return=-1 / FFFF hex / 65535 dez (Zeit unendlich)

Parameter-Status-LED-Mode

par1=DAPI_STEPPER_MOTORCHAR_PAR_STATUSLEDMODE

return=Betriebsart der Status-LED

return=0 (MOVE - LED leuchtet bei Motorbewegung)

return=1 (HALT - LED leuchtet bei Motorstillstand)

return=2 (ENDSW1 - LED leuchtet bei geschlossenen Endschanter1)

return=3 (ENDSW2 - LED leuchtet bei geschlossenen Endschanter2)

return=4 (REFSW1 - LED leuchtet bei geschlossenen Referenzschalterschalter1)

return=5 (REFSW2 - LED leuchtet bei geschlossenen Referenzschalterschalter2)

Parameter-Invert-END-Switch1

par1=DAPI_STEPPER_MOTORCHAR_PAR_INVERT_ENDSW1

return=Endschalter1 wird invertiert (0=normal / 1=invertieren)

Parameter-Invert-END-Switch2

par1=DAPI_STEPPER_MOTORCHAR_PAR_INVERT_ENDSW2

return=Endschalter2 wird invertiert (0=normal / 1=invertieren)

Parameter-Invert-Ref-Switch1

par1=DAPI_STEPPER_MOTORCHAR_PAR_INVERT_REFSW1

return=Referenzschalterschalter1 wird invertiert (0=normal / 1=invertieren)

Parameter-Invert-Ref-Switch2

par1=DAPI_STEPPER_MOTORCHAR_PAR_INVERT_REFSW2

return=Referenzschalterschalter2 wird invertiert (0=normal / 1=invertieren)

Parameter-Invert-direction

par1=DAPI_STEPPER_MOTORCHAR_PAR_INVERT_DIRECTION

return=Richtungsangaben werden invertiert (0=normal / 1=invertieren)

Parameter-Endswitch-Stopmode

par1= DAPI_STEPPER_MOTORCHAR_PAR_ENDSWITCH_STOPMODE

return=Einstellung des Stopverhaltens (0=Fullstop / 1=Stop)

Parameter-GoReferenceFrequencyToEndSwitch

par1=DAPI_STEPPER_MOTORCHAR_PAR_GOREFERENCEFREQUENCY_TOENDSWITCH

return=Frequenz [Vollschritt / s]

Parameter GoReferenceFrequencyAfterEndSwitch abfragen

par1=DAPI_STEPPER_MOTORCHAR_PAR_GOREFERENCEFREQUENCY_AFTERE
NDSWITCH

return=Frequenz [Vollschritt / s]

Parameter GoReferenceFrequencyToOffset abfragen

par1=DAPI_STEPPER_MOTORCHAR_PAR_GOREFERENCEFREQUENCY_TOOFFS
ET

return=Frequenz [Vollschritt / s]

Programmierbeispiel

```
value = DapiStepperCommand(handle, motor,  
DAPI_STEPPER_CMD_GET_MOTORCHARACTERISTIC,  
DAPI_STEPPER_MOTORCHAR_PAR_STEPMODE, 0, 0, 0);  
// Schrittmode (Voll-, Halb-, Viertel-, Achtel-, Sechszehntelschritt)  
  
value = DapiStepperCommand(handle, motor,  
DAPI_STEPPER_CMD_GET_MOTORCHARACTERISTIC,  
DAPI_STEPPER_MOTORCHAR_PAR_GOFREQUENCY, 0, 0, 0);  
// Schrittmode bei Motorstop (Voll-, Halb-, Viertel-, Achtel-,  
Sechszehntelschritt)
```

```

value = DapiStepperCommand(handle, motor,
DAPI_STEPPER_CMD_GET_MOTORCHARACTERISTIC,
DAPI_STEPPER_MOTORCHAR_PAR_STARTFREQUENCY, 0, 0, 0);
// Startfrequenz [Vollschritt / s]

value = DapiStepperCommand(handle, motor,
DAPI_STEPPER_CMD_GET_MOTORCHARACTERISTIC,
DAPI_STEPPER_MOTORCHAR_PAR_STOPFREQUENCY, 0, 0, 0);
// Stopfrequenz [Vollschritt / s]

value = DapiStepperCommand(handle, motor,
DAPI_STEPPER_CMD_GET_MOTORCHARACTERISTIC,
DAPI_STEPPER_MOTORCHAR_PAR_MAXFREQUENCY, 0, 0, 0);

// maximale Frequenz [Vollschritt / s]

value = DapiStepperCommand(handle, motor,
DAPI_STEPPER_CMD_GET_MOTORCHARACTERISTIC,
DAPI_STEPPER_MOTORCHAR_PAR_ACCELERATIONSLOPE, 0, 0, 0);
// Beschleunigung in [Vollschritten / ms]

value = DapiStepperCommand(handle, motor,
DAPI_STEPPER_CMD_GET_MOTORCHARACTERISTIC,
DAPI_STEPPER_MOTORCHAR_PAR_DECELERATIONSLOPE, 0, 0, 0);
// Bremsung in [Vollschritten / ms]

value = DapiStepperCommand(handle, motor,
DAPI_STEPPER_CMD_GET_MOTORCHARACTERISTIC,
DAPI_STEPPER_MOTORCHAR_PAR_PHASECURRENT, 0, 0, 0);
// Phasenstrom [mA]

value = DapiStepperCommand(handle, motor,
DAPI_STEPPER_CMD_GET_MOTORCHARACTERISTIC,
DAPI_STEPPER_MOTORCHAR_PAR_HOLDPHASECURRENT, 0, 0, 0);
// Phasenstrom bei Motorstillstand [mA]

value = DapiStepperCommand(handle, motor,
DAPI_STEPPER_CMD_GET_MOTORCHARACTERISTIC,
DAPI_STEPPER_MOTORCHAR_PAR_HOLDTIME, 0, 0, 0);
// Zeit in der der Haltestrom fließt nach Motorstop [s]

value = DapiStepperCommand(handle, motor,
DAPI_STEPPER_CMD_GET_MOTORCHARACTERISTIC,
DAPI_STEPPER_MOTORCHAR_PAR_STATUSLEDMODE, 0, 0, 0);

```

// Betriebsart der Status-LED

```
value = DapiStepperCommand(handle, motor,  
DAPI_STEPPER_CMD_GET_MOTORCHARACTERISTIC,  
DAPI_STEPPER_MOTORCHAR_PAR_INVERT_ENDSW1, 0, 0, 0);
```

// invertiere Funktion des Endschaltes1

```
value = DapiStepperCommand(handle, motor,  
DAPI_STEPPER_CMD_GET_MOTORCHARACTERISTIC,  
DAPI_STEPPER_MOTORCHAR_PAR_INVERT_ENDSW2, 0, 0, 0);
```

// invertiere Funktion des Endschaltes2

```
value = DapiStepperCommand(handle, motor,  
DAPI_STEPPER_CMD_GET_MOTORCHARACTERISTIC,  
DAPI_STEPPER_MOTORCHAR_PAR_INVERT_REFSW1, 0, 0, 0);
```

// invertiere Funktion des Referenzschalters1

```
value = DapiStepperCommand(handle, motor,  
DAPI_STEPPER_CMD_GET_MOTORCHARACTERISTIC,  
DAPI_STEPPER_MOTORCHAR_PAR_INVERT_REFSW2, 0, 0, 0);
```

// invertiere Funktion des Referenzschalters2

```
value = DapiStepperCommand(handle, motor,  
DAPI_STEPPER_CMD_GET_MOTORCHARACTERISTIC,  
DAPI_STEPPER_MOTORCHAR_PAR_INVERT_DIRECTION, 0, 0, 0);
```

// invertiere alle Richtungsangaben

```
value = DapiStepperCommand(handle, motor,  
DAPI_STEPPER_CMD_GET_MOTORCHARACTERISTIC,  
DAPI_STEPPER_MOTORCHAR_PAR_ENDSWITCH_STOPMODE, 0, 0,  
0);
```

// einstellen des Stopverhaltens

```
value = DapiStepperCommand(handle, motor,  
DAPI_STEPPER_CMD_GET_MOTORCHARACTERISTIC,  
DAPI_STEPPER_MOTORCHAR_PAR_GOREFERENCEFREQUENCY_TOENDSW  
ITCH, 0,0,0);
```

// Abfrage der Geschwindigkeit, mit der der Endschaltes angefahren wird.

```
value = DapiStepperCommand(handle, motor,  
DAPI_STEPPER_CMD_GET_MOTORCHARACTERISTIC,  
DAPI_STEPPER_MOTORCHAR_PAR_GOREFERENCEFREQUENCY_AFTEREN  
DSWITCH, 0,0,0);
```

// Abfrage der Geschwindigkeit, mit der vom Endschalter abgefahren wird.

```
value = DapiStepperCommand(handle, motor,  
DAPI_STEPPER_CMD_GET_MOTORCHARACTERISTIC,  
DAPI_STEPPER_MOTORCHAR_PAR_GOREFERENCEFREQUENCY_TOOFFSE  
T, 0,0,0);
```

// Abfrage der Geschwindigkeit, mit der der optionale Offset angefahren wird.

6.11.1.12. DAPI_STEPPEER_CMD_MOTORCHARACTERISTIC_EEPROM_SAVE

Beschreibung

Es wird die aktuelle Motorcharakteristik des Motors ins EEPROM abgespeichert.

Definition

```
DapiStepperCommand(handle, motor,  
DAPI_STEPPEER_CMD_MOTORCHARACTERISTIC_EEPROM_SAVE, 0, 0, 0, 0);
```

6.11.1.13. DAPI_STEPPER_CMD_MOTORCHARACTERISTIC_EEPROM_LOAD

Beschreibung

Es wird die Motorcharakteristik des Motors aus dem EEPROM geladen.

Definition

```
DapiStepperCommand(handle, motor,  
DAPI_STEPPER_CMD_MOTORCHARACTERISTIC_EEPROM_LOAD, 0, 0, 0, 0);
```


6.11.1.14. DAPI_STEPPER_CMD_MOTORCHARACTERISTIC_LOAD_DEFAULT

Beschreibung

Es wird die Motorcharakteristik des Motors auf Defaultwerte zurück gesetzt.

Definition

```
DapiStepperCommand(handle, motor,  
DAPI_STEPPER_CMD_MOTORCHARACTERISTIC_LOAD_DEFAULT, 0, 0, 0, 0);
```

Bemerkung

Die Defaultwerte sind folgende:

- Stepmode Vollschrift
- Schrittfrequenz bei GoPosition [Vollschrift / s]: 1000 Hz
- Startfrequenz [Vollschrift / s]: 200Hz
- Stopfrequenz [Vollschrift / s]: 200Hz
- Maximale Schrittfrequenz [Vollschrift / s]: 3000Hz
- Beschleunigungsrampe [Hz/10ms]: 10Hz/10ms
- Bremsrampe [Hz/10ms]: 10Hz/10ms
- Phasenstrom 0..1,5A [1mA]: 750mA
- Haltestrom 0..1,5A [1mA]: 500mA
- Haltezeit 0..unendlich [ms]: 15000ms
- Status_LEDfunktion: Move
- Funktion des Endschalter1: nicht invertiert
- Funktion des Endschalter2: nicht invertiert
- Funktion des Referenzschalter1: nicht invertiert
- Funktion des Referenzschalter2: nicht invertiert
- Funktion aller Richtungsangaben: nicht invertiert
- Endschaltermode: Fullstop
- Schrittfrequenz bei GoReferenz [Vollschrift / s]: 1000 Hz

6.11.1.15. DAPI_STEPPER_CMD_GO_REFSWITCH

Beschreibung

Der Motor fährt zur Referenzposition.

Definition

DapiStepperCommand(handle, motor, DAPI_STEPPER_CMD_GO_REFSWITCH, par1, par2, par3, 0);

Parameter

Mögliche Werte für par1: (werden mehrere benötigt, müssen die einzelnen addiert werden)

DAPI_STEPPER_GO_REFSWITCH_PAR_REF1

DAPI_STEPPER_GO_REFSWITCH_PAR_REF2

DAPI_STEPPER_GO_REFSWITCH_PAR_REF_LEFT

DAPI_STEPPER_GO_REFSWITCH_PAR_REF_RIGHT

DAPI_STEPPER_GO_REFSWITCH_PAR_REF_GO_POSITIVE

DAPI_STEPPER_GO_REFSWITCH_PAR_REF_GO_NEGATIVE

DAPI_STEPPER_GO_REFSWITCH_PAR_SET_POS_0

par2=Motorpositionsoffset (1/16 Vollschrift)

par3=Timeoutzeit [ms]

Bemerkung

Anfahren des Referenzschalters

Zunächst fährt der Motor zur Referenzposition 1 oder 2 (siehe par1).

Hierbei kann angegeben werden, ob der Referenzschalter 1 (DAPI_STEPPER_GO_REFSWITCH_PAR_REF1) oder der Referenzschalter 2 (DAPI_STEPPER_GO_REFSWITCH_PAR_REF2) angefahren wird. Dabei läßt sich die Richtung wählen in die der Motor startet. Mit dem Parameter DAPI_STEPPER_GO_REFSWITCH_PAR_REF_GO_NEGATIVE wird nach links und mit dem Parameter DAPI_STEPPER_GO_REFSWITCH_PAR_REF_GO_POSITIVE wird nach rechts gestartet.

Hierbei wird die Geschwindigkeit GOREFERENCEFREQUENCY_TOENDSWITCH benutzt (siehe **DapiStepperCommand_SetMotorcharacteristic**).

Herausfahren aus dem Referenzschalter

Danach fährt der Motor mit der Geschwindigkeit GOREFERENCEFREQUENCY_AFTERENDSWITCH aus der Referenzposition heraus. Dabei läßt sich wählen, ob der Motor die rechte oder linke Seite des Referenzschalters anfährt. Mit dem Parameter DAPI_STEPPER_GO_REFSWITCH_PAR_REF_LEFT wird die linke Kante angefahren und mit dem Parameter DAPI_STEPPER_GO_REFSWITCH_PAR_REF_RIGHT wird die rechte Kante angefahren.

Optionales Anfahren eines Offsets

Nach dem Herausfahren aus dem Referenzschalter kann noch ein Offset angefahren werden. Falls dieser Parameter nicht = 0 ist (par2), fährt der Motor zu diesem Offset mit der Geschwindigkeit GOREFERENCEFREQUENCY_TOOFFSET.

Nullen der Position des Motors

Mit dem Parameter DAPI_STEPPER_GO_REFSWITCH_PAR_SET_POS_0 kann zusätzlich eingestellt werden, ob der Motor jetzt die Position 0 bekommt.

Programmierbeispiel

```
DapiStepperCommand(handle, motor,
DAPI_STEPPER_CMD_GO_REFSWITCH,
DAPI_STEPPER_GO_REFSWITCH_PAR_REF1 +
DAPI_STEPPER_GO_REFSWITCH_PAR_REF_LEFT +
DAPI_STEPPER_GO_REFSWITCH_PAR_REF_GO_POSITIVE +
DAPI_STEPPER_GO_REFSWITCH_PAR_SET_POS_0, 0, 15000, 0);
```

6.11.1.16. DAPI_STEPPER_CMD_GET_CPU_TEMP

Beschreibung

Die Temperatur des CPU wird abgefragt.

Definition

```
ULONG DapiStepperCommand(handle, motor,  
DAPI_STEPPER_CMD_GET_CPU_TEMP, 0, 0, 0, 0);
```

Parameter

cmd=DAPI_STEPPER_CMD_GET_CPU_TEMP

Return-Wert

Temperatur [°C]

6.11.1.17. DAPI_STEPPER_CMD_GET_MOTOR_SUPPLY_VOLTAGE

Beschreibung

Hiermit wird die Versorgungsspannung des Motors abgefragt.

Definition

```
DapiStepperCommand(handle, motor,  
DAPI_STEPPER_CMD_GET_MOTOR_SUPPLY_VOLTAGE, 0, 0, 0, 0);
```

Parameter

cmd=DAPI_STEPPER_CMD_GET_MOTOR_SUPPLY_VOLTAGE

Return-Wert

Motorversorgungsspannung in [mV]

6.11.2. Status abfragen mit DapiStepperGetStatus

6.11.2.1. DAPI_STEPPER_STATUS_GET_ACTIVITY

Beschreibung

Hiermit werden verschiedene Statusinformationen (z.B. die Aktivität des Motorstroms, etc.) abgefragt.

Definition

```
ULONG DapiStepperGetStatus(handle, motor,  
DAPI_STEPPER_STATUS_GET_ACTIVITY);
```

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

motor=Nummer des anzusprechenden Motors

Return-Wert

Bit	Command	Beschreibung
0	DISABLE	Motor darf nicht verfahren
1	MOTORSTROMACTIV	Motorstrom ist aktiv
2	HALTESTROMACTIV	Haltestrom ist aktiv
3	GOPOSITIONACTIV	GoPosition ist aktiv
4	GOPOSITIONBREMSSEN	GoPosition Bremsung ist aktiv
5	GOREFERENZACTIV	GoReference ist aktiv

Programmierbeispiel

```
ret = DapiStepperGetStatus(handle, motor,  
DAPI_STEPPER_STATUS_GET_ACTIVITY);
```

6.11.2.2. DAPI_STEPPER_STATUS_GET_POSITION

Beschreibung

Hiermit wird eine bestimmte Position abgelesen.

Definition

ULONG DapiStepperGetStatus(handle, motor, cmd);

Parameter

cmd=DAPI_STEPPER_STATUS_GET_POSITION

Return-Wert

Es wird die aktuelle Motorposition in 1/16 Schritteinheiten zurückgegeben

Programmierbeispiel

```
value = DapiStepperGetStatus(handle, motor,  
DAPI_STEPPER_STATUS_GET_POSITION);
```

6.11.2.3. DAPI_STEPPER_STATUS_GET_SWITCH

Beschreibung

Hiermit wird der Zustand der Schalter abgefragt.

Definition

ULONG DapiStepperGetStatus(handle, motor, cmd);

Parameter

cmd=DAPI_STEPPER_STATUS_GET_SWITCH

Return-Wert

Es wird der Zustand der Schalter zurückgeliefert:

Bit0: ENDSCHALTER1; 1 = Endschalter1 ist geschlossen

Bit1: ENDSCHALTER2; 1 = Endschalter2 ist geschlossen

Bit2: REFSCHALTER1; 1 = Referenzschalter1 ist geschlossen

Bit3: REFSCHALTER2; 1 = Referenzschalter2 ist geschlossen

Programmierbeispiel

```
pos = DapiStepperGetStatus(handle, motor,  
DAPI_STEPPER_STATUS_GET_SWITCH);
```


6.11.3. DapiStepperCommandEx

Beschreibung

Dieser erweiterte Befehl steuert Schrittmotoren an.

Definition

```
ULONG DapiStepperCommandEx(ULONG handle, ULONG motor, ULONG cmd,  
ULONG  
par1, ULONG par2, ULONG par3, ULONG par4, ULONG par5, ULONG par6, ULONG  
par7);
```

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

motor=Nummer des anzusprechenden Motors

cmd=Erweitertes Kommando

par1..7=Erweiterte kommandoabhängige Parameter (s. Bemerkung)

Bemerkung

Siehe delib.h für die erweiterten Kommandos und den zugehörigen Parametern.

6.12. Ausgabe Timeout-Schutz-Funktion verwalten

6.12.1. 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

6.12.1.1. 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.
```

6.12.1.2. 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.
```

6.12.1.3. 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.
```

6.12.1.4. 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.
```


6.12.1.5. 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.
```

6.12.1.6. 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.
```

6.12.1.7. 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.
```

6.12.1.8. 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.
```

6.12.1.9. 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.
```

6.12.1.10. 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.
```

6.12.1.11. 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.
```

6.13. Testfunktionen

6.13.1. 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

6.14. Register Schreib-Befehle

6.14.1. 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.

6.14.2. 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.

6.14.3. 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.

6.14.4. 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.

6.15. Register Lese-Befehle

6.15.1. 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.

6.15.2. 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.

6.15.3. DapiReadLong

Beschreibung

Dieser Befehl führt einen direkten Register Lese-Befehl auf das Modul aus.

Definition

ULONG DapiReadLong(ULONG handle, ULONG adress);

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

adress=Adresse, auf die zugegriffen werden soll

Return-Wert

Inhalt des zu lesenden Registers (32 Bit)

Bemerkung

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

Programmbeispiel

```
char v0, v1, v2, v3;
uint ver;
float fw_ver;

ver = (uint)DapiReadLong(handle, 0xffff4);

v3 = (char)((ver >> 24) & 0xff);
v2 = (char)((ver >> 16) & 0xff);
v1 = (char)((ver >> 8) & 0xff);
v0 = (char)((ver >> 0) & 0xff);

fw_ver = (((float)v0) - '0') * 10 + (((float)v1) - '0')
+ (((float)v2) - '0') / 10 + (((float)v3) - '0') / 100;
// Hier wird die Firmware-Version des Modules ausgelesen.
```

6.15.4. 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.

6.16. 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



7. Anhang

7.1. 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

7.2. 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](#)

7.3. Revisionen

Rev 3.00	DEDITEC Design Update
Rev 2.20	Aktualisiert. Kapitel Software überarbeitet. Weboberfläche sowie DELIB Module Config hinzugefügt
Rev 2.08	Kapitel "Software" und "DELIB API Referenz" überarbeitet Kapitel "Ethernet Interface 2.0", "Ethernet Interface L/C", "CAN Interface 2.0", "RS-232/RS-485 Interface 2.0" und "USB Interface 2.0" hinzugefügt.
Rev 2.07	Kapitel "REL8_UM" hinzugefügt
Rev 2.06	Kapitel "Firmware Update" hinzugefügt, Kapitel "Software" überarbeitet
Rev 2.05	PT100 Modul ins Manual aufgenommen
Rev 2.04	Kapitel Abmessung der RO-Serie hinzugefügt
Rev 2.03	Neue Verwaltungsfunktion "DapiGetDELIBVersion", neuer DI Befehl "DapiSpecialCounterLatchAll", "DapiSpecialCounterLatchAllWithReset" und Ergänzung der Modi bei "DapiDIGetCounter"
Rev 2.02	Erweiterung um Kapitel "RO-O8-R8" und neuer DO-Befehl "DapiDOSet1_WithTimer"
Rev 2.01	Neuer Stepper Befehl "DAPI_STEPPER_CMD_GO_POSITION_RELATIVE" Ergänzung des Return-Werts bei Befehl "DAPI_STEPPER_STATUS_GET_ACTIVITY" Ergänzung des Parameters Hold-Time (Zeit unendlich) bei Befehl "DAPI_STEPPER_CMD_SET_MOTORCHARACTERISTIC"
Rev 2.00	Designänderung
Rev 1.7	Manual wurde um Ethernet Interface erweitert
Rev 1.6	Kapitel 2.2.2.3 um CAN/SER Adapter erweitert
Rev 1.5	Änderung Abschnitt CAN-Interface Einstellungen
Rev 1.4	Erweiterung Kapitel Stepper
Rev 1.3	Änderung Konfiguration seriell / CAN

Rev 1.2	Ergänzung RO-AD16_ISO
Rev 1.1	Software Installation
Rev 1.01	Erweiterung flexibles Steckverbindersystem Überarbeitung der Blockschaltbilder
Rev 1.00	Erste Anleitung

7.4. 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.