



DELIB

DEDITEC Treiber Bibliothek

2025 Januar

INDEX

1. Software Beschreibung	13
1.1. Benutzen unserer Produkte	14
1.1.1. Ansteuerung über unsere DELIB-Treiberbibliothek	14
1.1.2. Ansteuerung über mitgelieferte Testprogramme	14
1.1.3. Ansteuerung auf Protokollebene	15
1.1.4. DELIB CLI (command-line interface) für Windows	16
1.1.4.1. Konfiguration des DELIB CLI	18
1.1.4.2. DELIB CLI Beispiele	19
1.1.5. Ansteuerung über grafische Anwendungen	23
1.1.5.1. LabVIEW	23
1.1.5.2. ProfiLab	23
1.1.5.3. Licht24 Pro	24
1.1.6. Einbinden der DELIB in Programmiersprachen	25
1.1.6.1. Einbinden der DELIB in Visual-C/C++	25
1.1.6.2. Einbinden der DELIB in Visual-C/C++ (Visual Studio 2015)	27
1.1.6.3. Einbinden der DELIB in Visual-C#	30
1.1.6.4. Einbinden der DELIB in Delphi	31
1.1.6.5. Einbinden der DELIB in Visual-Basic (VB)	32
1.1.6.6. Einbinden der DELIB in Visual-Basic.NET (VB.NET)	33
1.1.6.7. Einbinden der DELIB in MS-Office (VBA)	34
1.1.6.8. Einbinden der DELIB in LabVIEW	36
1.1.6.8.1. Einbinden der DELIB in LabVIEW	36
1.1.6.8.2. Verwendung der VIs in LabVIEW	45
1.1.6.8.3. Setzen der Modul-ID in LabVIEW	47
1.1.6.9. Einbinden der DELIB in Java	49
1.2. DELIB-Treiberbibliothek	50
1.2.1. Übersicht	50
1.2.1.1. Unterstützte Programmiersprachen	53
1.2.1.2. Unterstützte Betriebssysteme	54
1.2.1.3. SDK-Kit für Programmierer	54
1.2.2. DELIB Setup	55

INDEX

1.2.3. ICT-Tool	60
1.3. DELIB Sample Sources (Windows Programmbeispiele)	60
1.3.1. Installation DELIB Sample Sources	61
1.3.2. Benutzung der DELIB Sample Sources	64
1.3.2.1. Schritt 1 - Produktauswahl	64
1.3.2.2. Schritt 2 - Kategorieauswahl	66
1.3.2.3. Schritt 3 - Programmiersprachenauswahl	67
1.3.2.4. Schritt 4 - Quellcode	68
1.4. DELIB für Linux	71
1.4.1. Verwenden der DELIB-Treiberbibliothek für Linux	74
1.4.1.1. Delib USB-Sample in Linux	74
1.4.1.2. Delib ETH-Sample in Linux	77
1.4.2. DELIB CLI (command-line interface) für Linux	81
1.4.2.1. Konfiguration des DELIB CLI	84
1.4.2.2. DELIB CLI Beispiele	87
1.5. Weboberfläche	88
1.5.1. Konfiguration	90
1.5.1.1. Allgemein	90
1.5.1.2. Netzwerk Konfiguration	91
1.5.1.3. Benutzer Manager	92
1.5.1.4. Status	95
1.5.1.5. Sicherheit	96
1.5.2. Ein-/Ausgänge	98
1.5.2.1. Allgemein	98
1.5.2.2. Digitale Eingänge	99
1.5.2.3. Digitale Eingänge Zähler	101
1.5.2.4. Digitale Ausgänge	103
1.5.2.5. Analoge Eingänge	105
1.5.2.6. Analoge Ausgänge	106
1.5.2.7. Konfiguration	107
<u>2. ICT-Tool</u>	110

INDEX

2.1. Allgemeine Beschreibungen	111
2.2. Modul hinzufügen	116
2.2.1. USB Module	118
2.2.2. Ethernet Module	119
2.2.2.1. Ethernet Standard Methode	120
2.2.2.2. Ethernet Direct Methode	123
2.2.2.3. Search via Broadcast	126
2.2.3. CAN Module	128
2.2.4. SER Module	129
2.3. Modul konfigurieren	131
2.3.1. Modul Infoseite	132
2.3.2. Modul Identifikation	133
2.3.3. Modul Neustart	134
2.3.4. USB Konfiguration	135
2.3.5. LAN Netzwerkinformationen	136
2.3.6. LAN Netzwerkeinstellungen	138
2.3.7. WiFi Netzwerkinformationen	140
2.3.8. WiFi Netzwerkeinstellungen	142
2.3.9. WiFi WPS-Verbindung	144
2.3.10. NTP-Konfiguration	145
2.3.11. TCP-Verschlüsselung	147
2.3.12. WEB-Login	151
2.3.13. Serielle Konfiguration	152
2.3.14. I/O Kanal-Namen	155
2.3.15. D/A-Startwerte	156
2.3.16. CAN-Konfiguration	157
2.3.16.1. CAN-Status Interface	157
2.3.16.2. CAN-Statistik TX/RX	159
2.3.16.3. CAN Main Interface	160
2.3.16.4. CAN Main I/O Init	161
2.3.16.5. CAN TX-Mode	163
2.3.16.6. CAN RX-Mode	165

INDEX

2.3.17. Watchdog-Konfiguration	166
2.4. M2M Konfiguration	167
2.4.1. Statistik	168
2.4.2. Empfangsmodule	170
2.4.3. Aktion	171
2.5. Event-Control	173
2.5.1. Statistik	174
2.5.2. Konfiguration Events	175
2.5.3. Konfiguration Aktionen	177
2.6. Modul testen	178
2.6.1. Timeout Test-Funktion	178
2.6.2. Digital Out	179
2.6.3. Digital TTL	181
2.6.4. Digital In	183
2.6.5. Digital In Counter	184
2.6.6. Analog In	186
2.6.7. Analog Out	187
2.6.8. Temp	189
2.6.9. Grafische Darstellung	190
2.6.10. CAN Runtime Parameter	192
2.6.11. Puls Generator	193
2.6.12. PWM Out	194
2.6.13. Watchdog Schaltverhalten	196
2.7. Modul diagnostizieren	197
2.7.1. Digitaler Kabelrückführungstest	197
2.7.2. Analoger Kabelrückführungstest	200
2.7.3. Zugriffszeitentests	202
2.7.4. A/D-Rauschen Graph	204
2.7.5. A/D-Lesen mit Abweichung	205
2.7.6. FIFO In/Out	206
2.7.7. Main Loop	208

INDEX

2.7.8. Module-Config-Memory	209
2.8. Firmware-Update	210
2.8.1. Über DEDITEC-Firmware	210
2.8.2. Flash-Files per ICT-Tool aktualisieren	211
2.8.3. Firmware Update durchführen	212
2.9. Einstellungen	214
2.9.1. Allgemein	214
2.9.2. Updates	215
2.9.3. Fehlerprotokoll	216
2.9.4. DELIB-DebugView-Global	217
2.9.5. DELIB-DebugView-Details	218
3. <u>DELIB API Referenz</u>	219
3.1. Verfügbare DEDITEC-Modul-IDs	220
3.2. Verzeichnisstruktur der DELIB	224
3.2.1. Include Verzeichnis	226
3.2.2. Library-Verzeichnis	226
3.2.3. Library-Verzeichnis für Borland	226
3.2.4. Umgebungsvariablen	226
3.3. Verwaltungsfunktionen	226
3.3.1. DapiOpenModule	226
3.3.2. DapiCloseModule	228
3.3.3. DapiGetDELIBVersion	228
3.3.4. DapiSpecialCMDGetModuleConfig	229
3.3.5. DapiOpenModuleEx	232
3.3.6. DapiScanAllModulesAvailable	234
3.4. Fehlerbehandlung	235
3.4.1. DapiGetLastError	235
3.4.2. DapiGetLastErrorText	236
3.4.3. DapiClearLastError	237
3.4.4. DapiGetLastErrorByHandle	238

INDEX

3.4.5. DapiClearLastErrorByHandle	239
3.5. Digitale Eingänge lesen	240
3.5.1. DapiDlGet1	240
3.5.2. DapiDlGet8	240
3.5.3. DapiDlGet16	241
3.5.4. DapiDlGet32	242
3.5.5. DapiDlGet64	243
3.5.6. DapiDlGetFF32	244
3.5.7. DapiDlGetCounter	245
3.5.8. DapiSpecialCounterLatchAll	246
3.5.9. DapiSpecialCounterLatchAllWithReset	247
3.5.10. DapiSpecialDlFilterValueSet	248
3.5.11. DapiSpecialDlFilterValueGet	249
3.5.12. Dapi_Special_DI_FF_Filter_Value_Set	250
3.5.13. Dapi_Special_DI_FF_Filter_Value_Get	251
3.6. Digitale Ausgänge verwalten	252
3.6.1. DapiDOSet1	252
3.6.2. DapiDOSet8	252
3.6.3. DapiDOSet16	253
3.6.4. DapiDOSet32	254
3.6.5. DapiDOSet64	255
3.6.6. DapiDOSet1_WithTimer	256
3.6.7. DapiDOReadback32	257
3.6.8. DapiDOReadback64	257
3.6.9. DapiDOSetBit32	258
3.6.10. DapiDOClrBit32	259
3.7. Digitale Zähler Funktionen für RO-Counter Module	260
3.7.1. DapiCnt48ModeSet	260
3.7.2. DapiCnt48ModeGet	266
3.7.3. DapiCnt48CounterGet32	267
3.7.4. DapiCnt48CounterGet48	268

INDEX

3.7.5. DapiSpecialCNT48ResetSingle	269
3.7.6. DapiSpecialCNT48ResetGroup8	270
3.7.7. DapiSpecialCNT48LatchGroup8	271
3.7.8. DapiSpecialCNT48DIGet1	272
3.8. PWM Funktionen	273
3.8.1. DapiPWMOutSet	273
3.8.2. DapiPWMOutReadback	275
3.8.3. DAPI_SPECIAL_PWM_FREQ_SET	276
3.8.4. DAPI_SPECIAL_PWM_FREQ_READBACK	277
3.9. Pulsgenerator Ausgänge verwalten	278
3.9.1. DapiPulseGenSet	278
3.10. A/D-Wandler Funktionen	280
3.10.1. DapiADSetMode	280
3.10.2. DapiADGetMode	283
3.10.3. DapiADGet	283
3.10.4. DapiADGetVolt	284
3.10.5. DapiADGetmA	284
3.10.6. DapiSpecialADReadMultipleAD	285
3.10.7. DapiSpecialADFilterSet	287
3.10.8. DapiSpecialADFilterGet	290
3.11. D/A-Ausgänge verwalten	292
3.11.1. DapiDASetMode	292
3.11.2. DapiDAGetMode	294
3.11.3. DapiDASet	295
3.11.4. DapiDASetVolt	296
3.11.5. DapiDASetmA	297
3.11.6. DapiSpecialCmd_DA	298
3.12. PT100 Funktionen	300
3.12.1. DapiTempGet	300
3.13. TTL-Ein-/Ausgangs Richtungen setzen mit DapiSpecialCommand	301
3.13.1. DAPI_SPECIAL_CMD_SET_DIR_DX_1	301

INDEX

3.13.2. DAPI_SPECIAL_CMD_SET_DIR_DX_8	303
3.13.3. DAPI_SPECIAL_CMD_GET_DIR_DX_8	304
3.14. Watchdog Funktionen	305
3.14.1. DapiWatchdogEnable	305
3.14.2. DapiWatchdogDisable	306
3.14.3. DapiWatchdogRetrigger	307
3.15. Schrittmotoren Funktionen	308
3.15.1. Befehle mit DapiStepperCommand	308
3.15.1.1. DAPI_STEPPER_CMD_GO_POSITION	308
3.15.1.2. DAPI_STEPPER_CMD_GO_POSITION_RELATIVE	309
3.15.1.3. DAPI_STEPPER_CMD_SET_POSITION	310
3.15.1.4. DAPI_STEPPER_CMD_SET_FREQUENCY	311
3.15.1.5. DAPI_STEPPER_CMD_GET_FREQUENCY	312
3.15.1.6. DAPI_STEPPER_CMD_SET_FREQUENCY_DIRECTLY	313
3.15.1.7. DAPI_STEPPER_CMD_STOP	314
3.15.1.8. DAPI_STEPPER_CMD_FULLSTOP	315
3.15.1.9. DAPI_STEPPER_CMD_DISABLE	316
3.15.1.10. DAPI_STEPPER_CMD_SET_MOTORCHARACTERISTIC	317
3.15.1.11. DAPI_STEPPER_CMD_GET_MOTORCHARACTERISTIC	324
3.15.1.12. DAPI_STEPPER_CMD_MOTORCHARACTERISTIC_EEPROM_SAVE	333
3.15.1.13. DAPI_STEPPER_CMD_MOTORCHARACTERISTIC_EEPROM_LOAD	334
3.15.1.14. DAPI_STEPPER_CMD_MOTORCHARACTERISTIC_LOAD_DEFAULT	335
3.15.1.15. DAPI_STEPPER_CMD_GO_REFSWITCH	336
3.15.1.16. DAPI_STEPPER_CMD_GET_CPU_TEMP	338
3.15.1.17. DAPI_STEPPER_CMD_GET_MOTOR_SUPPLY_VOLTAGE	339
3.15.2. Status abfragen mit DapiStepperGetStatus	340

INDEX

3.15.2.1. DAPI_STEPPER_STATUS_GET_ACTIVITY	340
3.15.2.2. DAPI_STEPPER_STATUS_GET_POSITION	341
3.15.2.3. DAPI_STEPPER_STATUS_GET_SWITCH	342
3.15.3. DapiStepperCommandEx	343
3.16. CAN Runtime Funktionen	344
3.16.1. RunTimeVarWriteToModule	344
3.17. Software FIFO verwalten	360
3.17.1. DapiSpecialCMDSWFifo	360
3.17.1.1. DapiSpecialSWFifoInitAndClear	361
3.17.1.2. DapiSpecialSWFifoSetIOModule	362
3.17.1.3. DapiSpecialSWFifoGetIOModule	363
3.17.1.4. DapiSpecialSWFifoIOModActivate	364
3.17.1.5. DapiSpecialSWFifoIOModDeactivate	364
3.17.1.6. DapiSpecialSWFifoIOModSetChannel	365
3.17.1.7. DapiSpecialSWFifoIOModGetChannel	366
3.17.1.8. DapiSpecialSWFifoIOModSetFrequencyHz	367
3.17.1.9. DapiSpecialSWFifoIOModGetFrequencyHz	368
3.17.1.10. DapiSpecialSWFifoIOModSetMode	369
3.17.1.11. DapiSpecialSWFifoIOModGetMode	370
3.17.1.12. DapiSpecialSWFifoIOModIOActivate	371
3.17.1.13. DapiSpecialSWFifoIOModIODeactivate	371
3.17.1.14. DapiSpecialSWFifoIOModGetBytesPerSample	372
3.17.1.15. DapiSpecialSWFifoIOModGetBytesAvailable	373
3.17.1.16. DapiSpecialSWFifoIOModGetBytesFree	374
3.17.1.17. DapiSpecialSWFifoIOModGetStatTXCnt	375
3.17.1.18. DapiSpecialSWFifoIOModGetStatTXErrCnt	376
3.17.1.19. DapiSpecialSWFifoIOModGetStatRXCnt	377
3.17.1.20. DapiSpecialSWFifoIOModGetStatRXErrCnt	378
3.17.1.21. DapiSpecialSWFifoGetStatus	379
3.17.1.22. DapiSpecialSWFifoGetStatusBytesOut	381
3.17.1.23. DapiSpecialSWFifoGetBytesFree	382
3.17.1.24. DapiSpecialSWFifoGetActivity	383

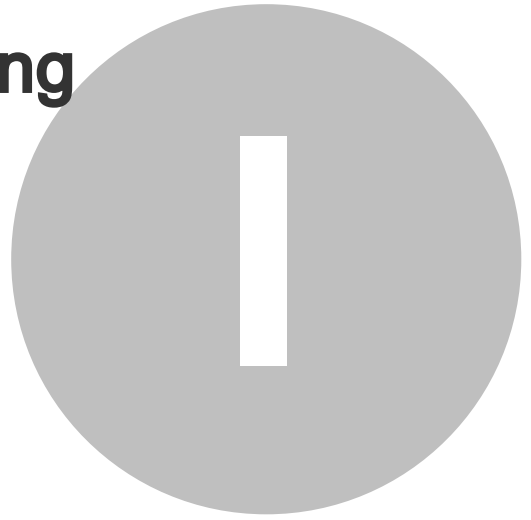
INDEX

3.17.1.25. DapiSpecialSWFifoGetInstanceType	384
3.17.2. DapiWriteFifo	386
3.17.3. DapiReadFifo	387
3.18. Ausgabe-Timeout verwalten	391
3.18.1. DapiSpecialCMDTimeout	391
3.18.1.1. DapiSpecialTimeoutSetValueSec	394
3.18.1.2. DapiSpecialTimeoutActivate	395
3.18.1.3. DapiSpecialTimeoutActivateAutoReactivate	396
3.18.1.4. DapiSpecialTimeoutActivateSecureOutputs	397
3.18.1.5. DapiSpecialTimeoutDeactivate	398
3.18.1.6. DapiSpecialTimeoutGetStatus	399
3.18.1.7. DapiSpecialTimeoutDoValueMaskWRSet32	400
3.18.1.8. DapiSpecialTimeoutDoValueMaskRDSet32	401
3.18.1.9. DapiSpecialTimeoutDoValueMaskWRClr32	402
3.18.1.10. DapiSpecialTimeoutDoValueMaskRDClr32	403
3.18.1.11. DapiSpecialTimeoutDoValueLoadDefault	404
3.19. Testfunktionen	405
3.19.1. DapiPing	405
3.20. Register Schreib-Befehle	406
3.20.1. DapiWriteByte	406
3.20.2. DapiWriteWord	407
3.20.3. DapiWriteLong	408
3.20.4. DapiWriteLongLong	409
3.21. Register Lese-Befehle	410
3.21.1. DapiReadByte	410
3.21.2. DapiReadWord	411
3.21.3. DapiReadLong	412
3.21.4. DapiReadLongLong	413
3.22. Programmier-Beispiel	414
3.23. Delib Übersichtstabelle	416
<u>4. Sicherheitsmaßnahmen</u>	419

INDEX

4.1. Schreibschutz	420
4.2. TCP-Verschlüsselung	420
4.3. Weboberfläche	420
<u>5. Anhang</u>	421
5.1. Revisionen	422
5.2. Urheberrechte und Marken	424

Software Beschreibung



1. Software Beschreibung

1.1. Benutzen unserer Produkte

1.1.1. 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.

1.1.2. Ansteuerung über mitgelieferte Testprogramme

Mit unserem ICT-Tool können Sie unsere Steuer- & Regelungstechnik Produkte ohne großen Konfigurationsaufwand auf Funktionalität testen.

Für ausführliche Informationen siehe Kapitel **ICT-Tool**.

1.1.3. 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.

1.1.4. 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 Flip-Flops 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

1.1.4.1. 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 ICT-Tool 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 ICT-Tool sowie über die Weboberfläche des Moduls eingestellt werden.

1.1.4.2. DELIB CLI Beispiele

Digitale Ausgänge

```
delib_cli DO1 17 1
```

→ schaltet das 18. digitale Relais an

```
delib_cli DO1 3 0
```

→ schaltet das 4. digitale Relais aus

```
delib_cli DO8 0 255
```

→ schaltet die digitalen Relais 1 bis 8 an

```
delib_cli DO16 0 0
```

→ schaltet die digitalen Relais 1 bis 16 aus

```
delib_cli DO16 16 65535
```

→ schaltet die digitalen Relais 17 bis 32 an

```
delib_cli DO32 0 4294967295
```

→ schaltet die digitalen Relais 1 bis 32 an

Digitale Eingänge

```
delib_cli DI1 3
```

Beispiel eines Rückgabewertes: 1

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

```
delib_cli DI8 0 hex
```

Beispiel eines Rückgabewertes: **0xC8**

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

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

```
delib_cli DI16 0 hex
```

Beispiel eines Rückgabewertes: **0xE0C0**

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

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

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

```
delib_cli DI8 0 hex nunit
```

Beispiel eines Rückgabewertes: **FF**

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

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

```
delib_cli FF 0
```

Beispiel eines Rückgabewertes: **192**

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

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

```
delib_cli FF 32
```

Beispiel eines Rückgabewertes: **65535**

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

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

```
delib_cli FF 0 hex
```

Beispiel eines Rückgabewertes: **0xD00**

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

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

Analoge Ausgänge

`delib_cli AO 7 4711`

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

`delib_cli AO 6 0x4711`

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

`delib_cli AO 7 3.7V`

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

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

`delib_cli AO 7 13.3mA`

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

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

Analoge Eingänge

```
delib_cli AI 2
```

Beispiel eines Rückgabewertes: **1234**

→ liest den Wert des 3. analogen Eingangs als Dezimalzahl

```
delib_cli AI 2 hex
```

Beispiel eines Rückgabewertes: **0x1FA**

→ liest den Wert des 3. analogen Eingangs als Hexadezimalzahl

```
delib_cli AI 2 V
```

Beispiel eines Rückgabewertes: **12.500000V**

→ liest die Spannung des 3. analogen Eingangs als Kommazahl

```
delib_cli AI 2 mA
```

Beispiel eines Rückgabewertes: **20.551600mA**

→ liest den Strom des 3. analogen Eingangs als Kommazahl

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

```
delib_cli AI 3 hex nunit
```

Beispiel eines Rückgabewertes: **1FA**

→ liest den Wert des 4. analogen Eingangs als Hexadezimalzahl

```
delib_cli AI 3 V nunit
```

Beispiel eines Rückgabewertes: **12.500000**

→ liest die Spannung des 4. analogen Eingangs als Kommazahl

1.1.5. Ansteuerung über grafische Anwendungen

1.1.5.1. 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

1.1.5.2. 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

1.1.5.3. 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>

1.1.6. Einbinden der DELIB in Programmiersprachen

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

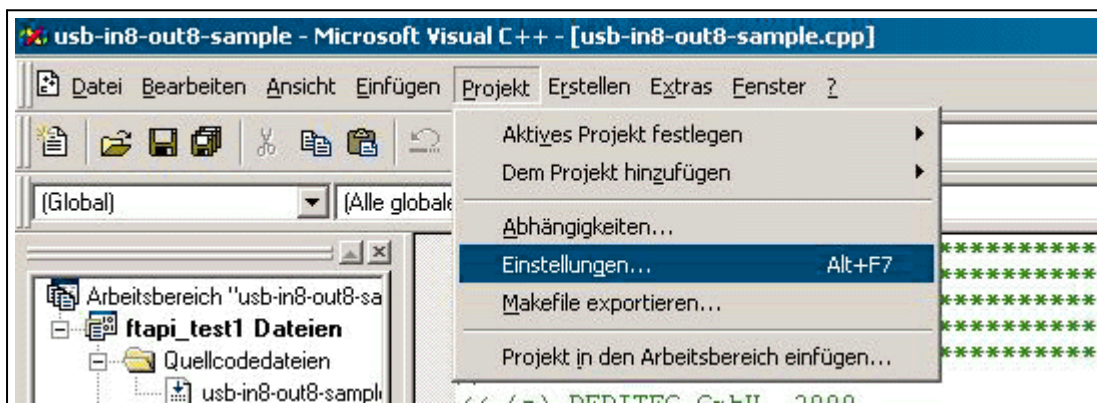
Zur Erleichterung für Verweise auf das DELIB-Include und das DELIB-Lib Verzeichnis werden bei installation der DELIB Umgebungsvariablen definiert.

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

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

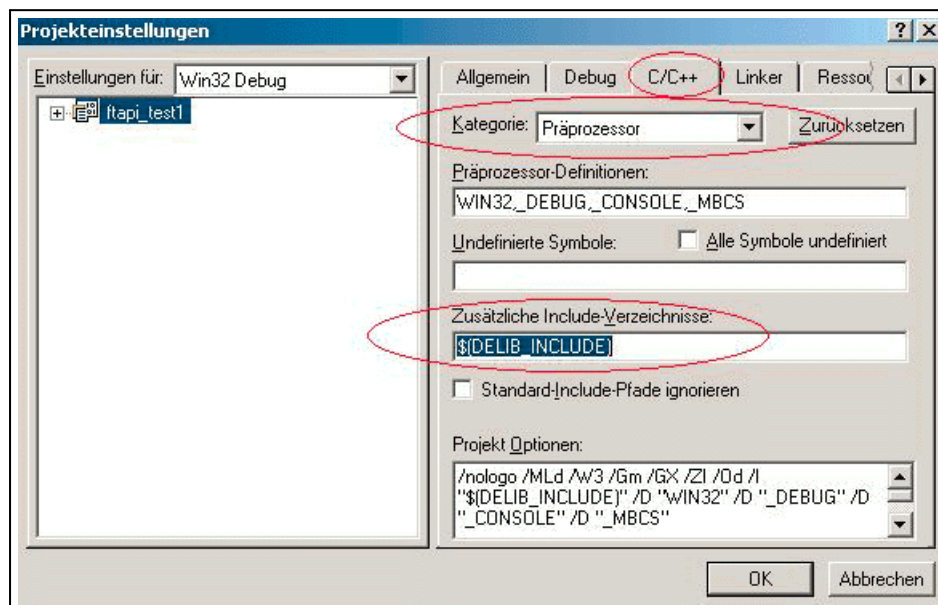
Diese werden im Folgenden in den Projekteinstellungen des Compilers eingetragen.

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



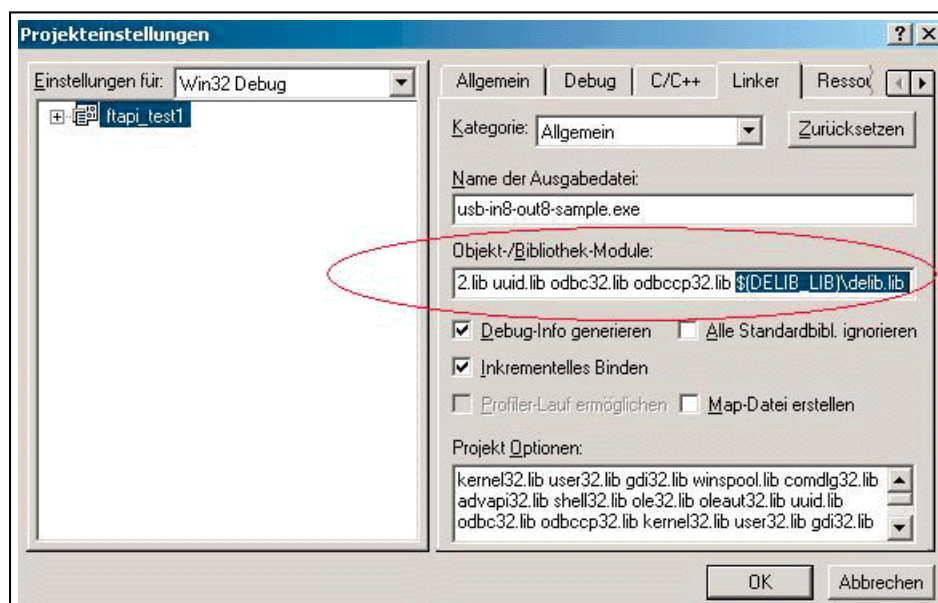
DELIB.H Eintrag in den Visual-C/C++ Projekt Einstellungen

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.



1.1.6.2. 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

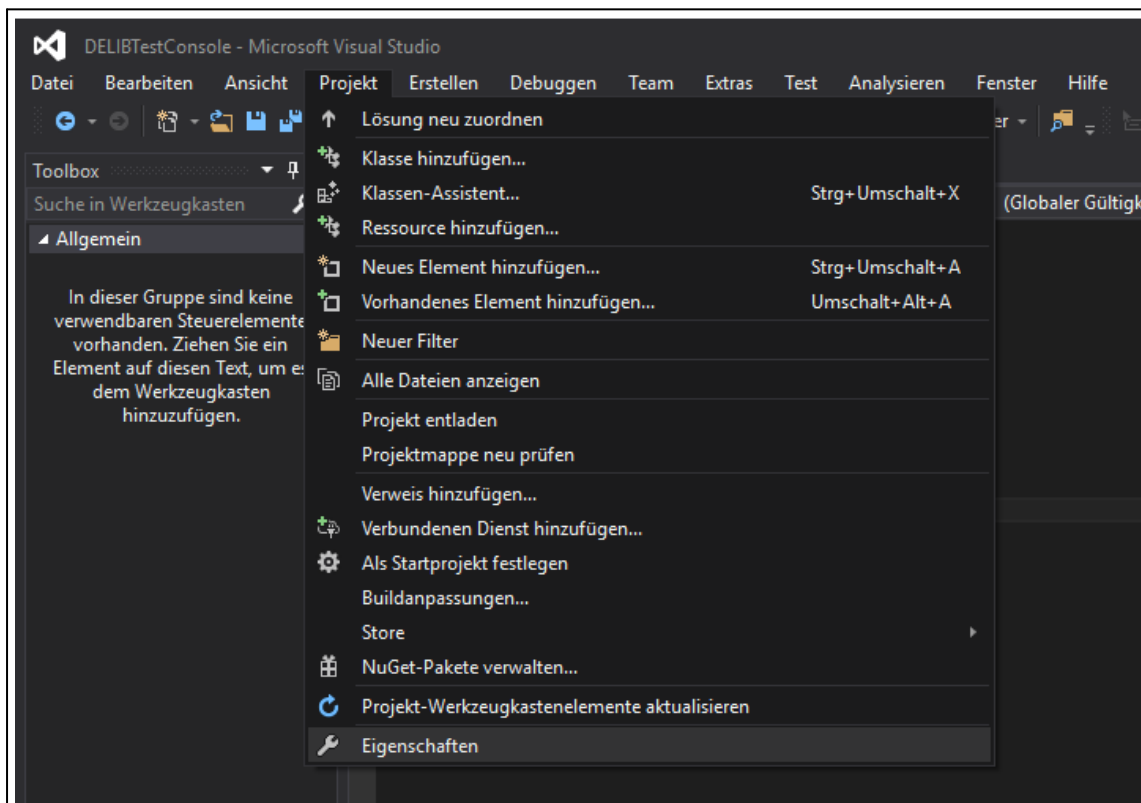
64 Bit DELIB Installation

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

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

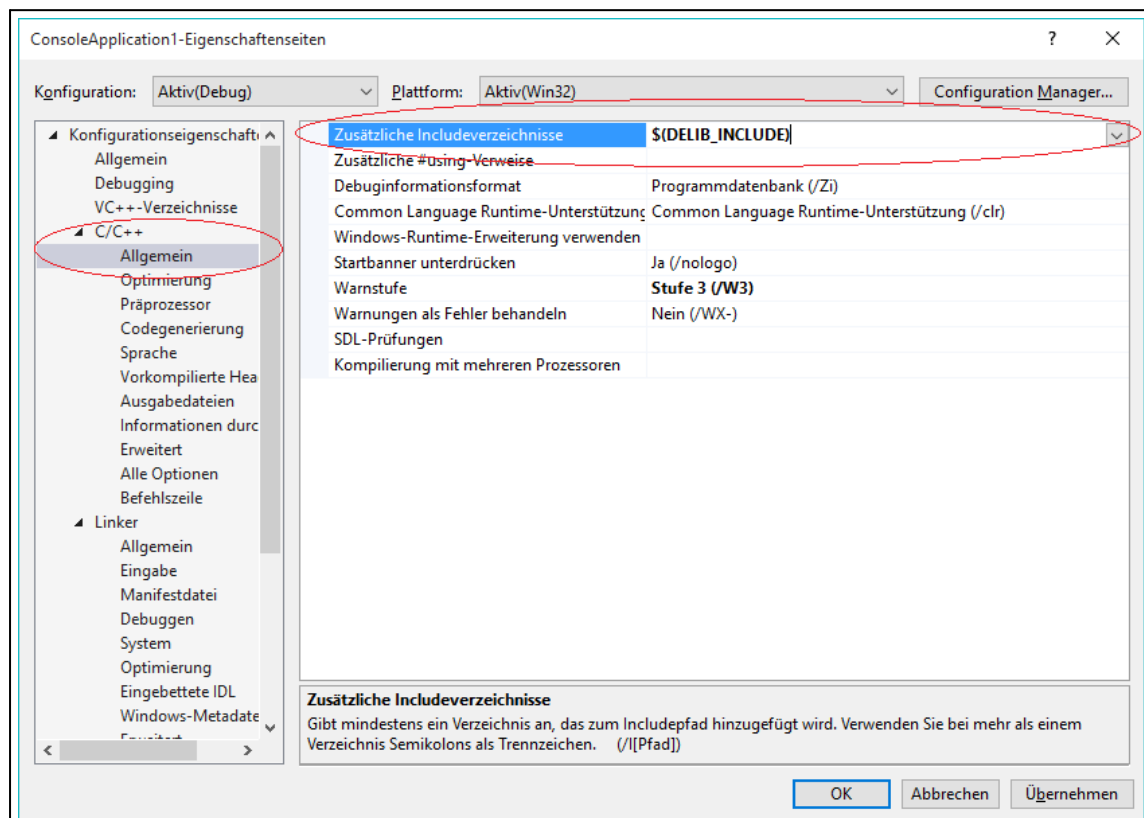
Diese werden im Folgenden in den Projekteinstellungen des Compilers eingetragen.

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



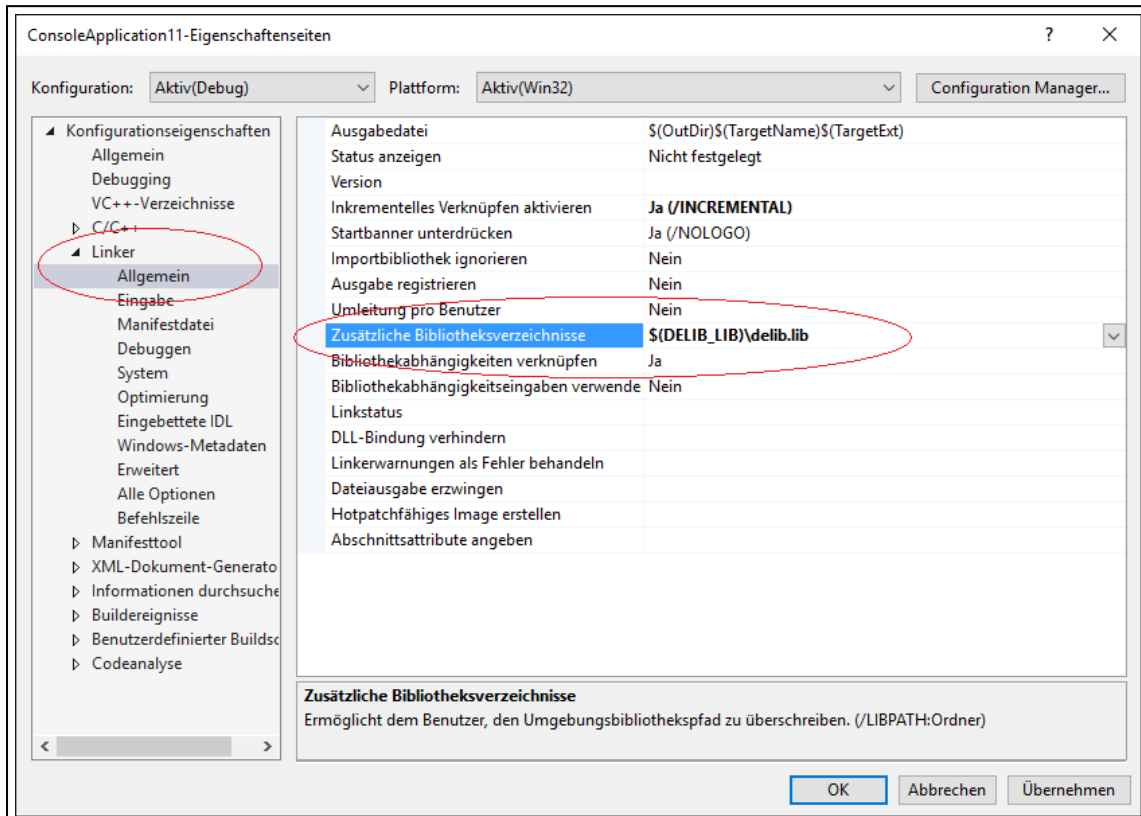
DELIB.H Eintrag in den Visual-C/C++ Projekt Einstellungen

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

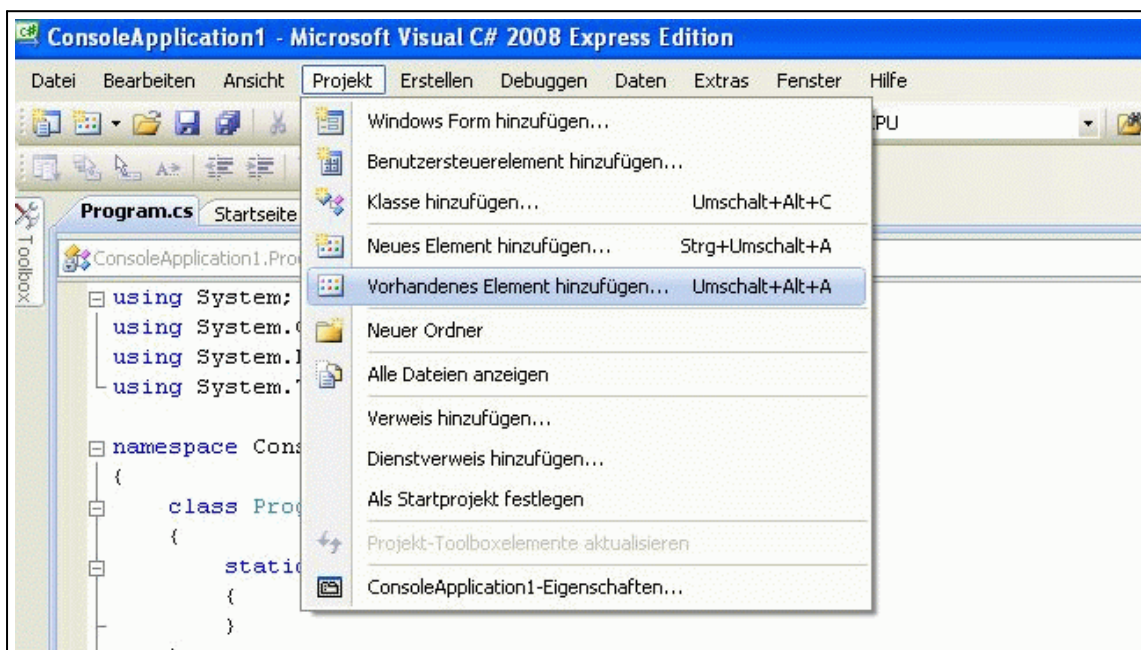
Unter dem Reiter "Linker" bei "Allgemein" "\$ (DELIB_LIB)\delib.lib" eintragen.



1.1.6.3. 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:

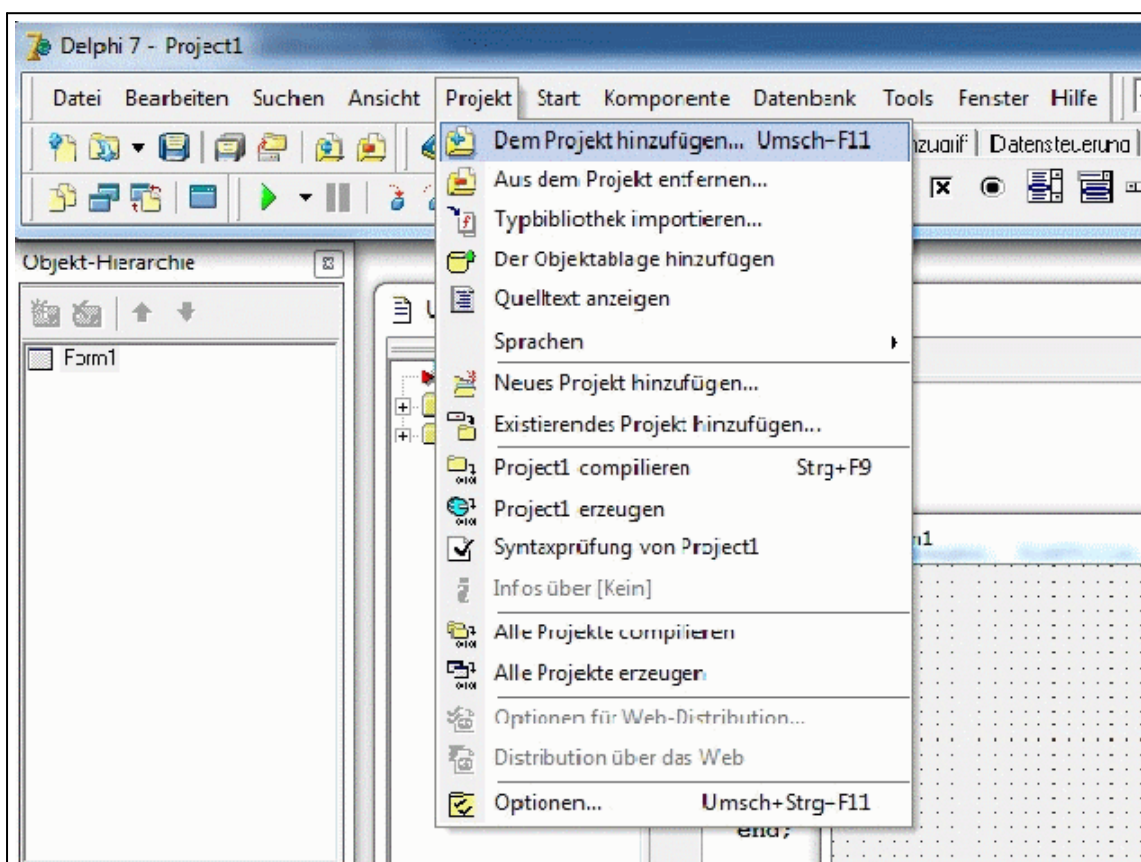
using DeLib;

1.1.6.4. Einbinden der DELIB in Delphi

Die benötigte Datei für Delphi befindet sich im Verzeichnis

C:\Programme\DEDITEC\DELIB\include.

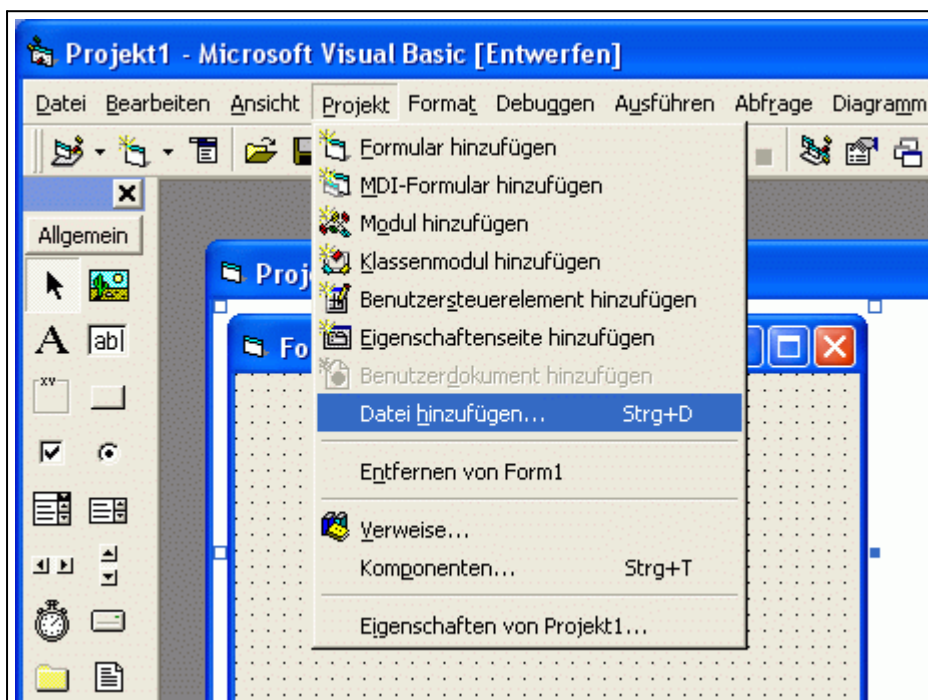
Delphi starten und über das Menue "Projekt → dem Projekt hinzufügen" im Verzeichnis C:\Programme\DEDITEC\DELIB\include\ die Datei delib.pas zum Importieren öffnen.



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

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

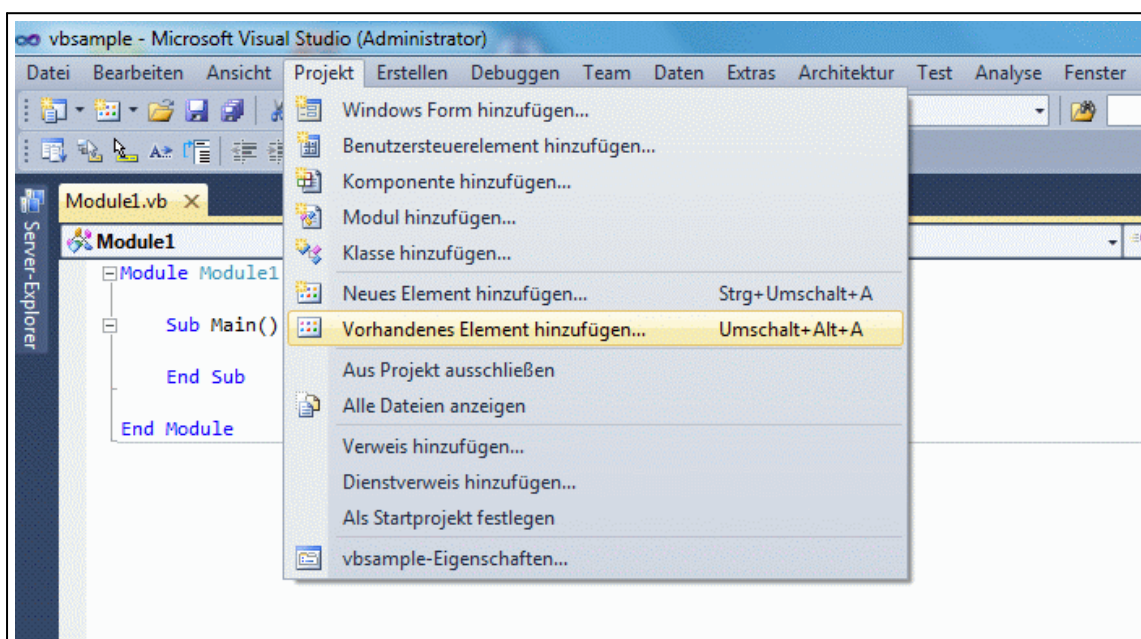
Visual Basic starten und über das Menue "Projekt → Datei hinzufügen..." im Verzeichnis C:\Programme\DEDITEC\DELIB\include\ die Datei delib.bas zum Importieren öffnen.



1.1.6.6. 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.

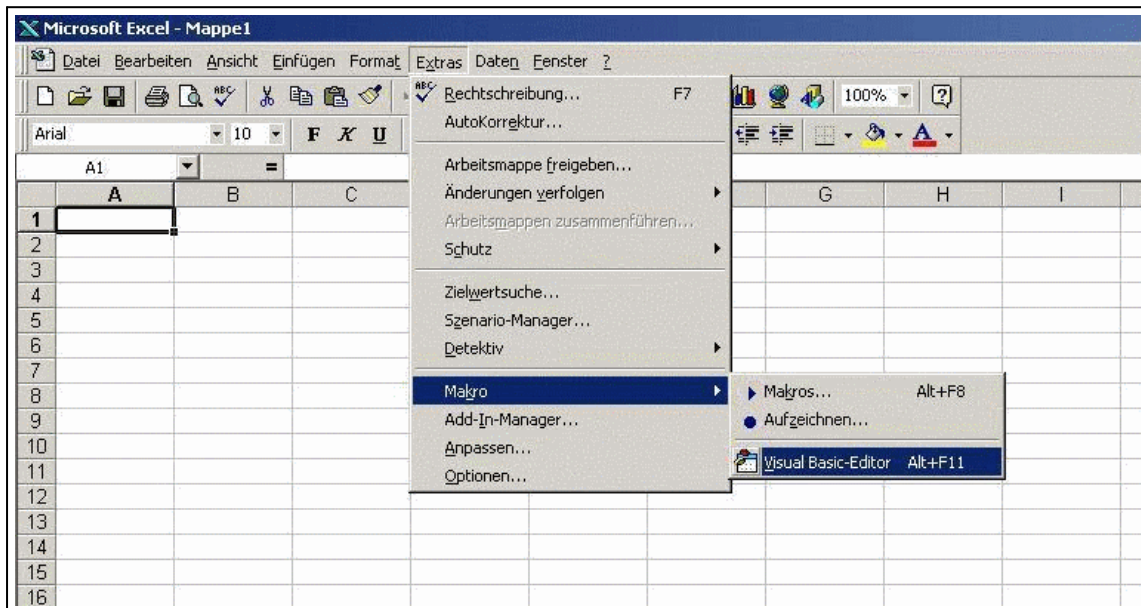


1.1.6.7. Einbinden der DELIB in MS-Office (VBA)

Die benötigte Datei für VBA befindet sich im Verzeichnis

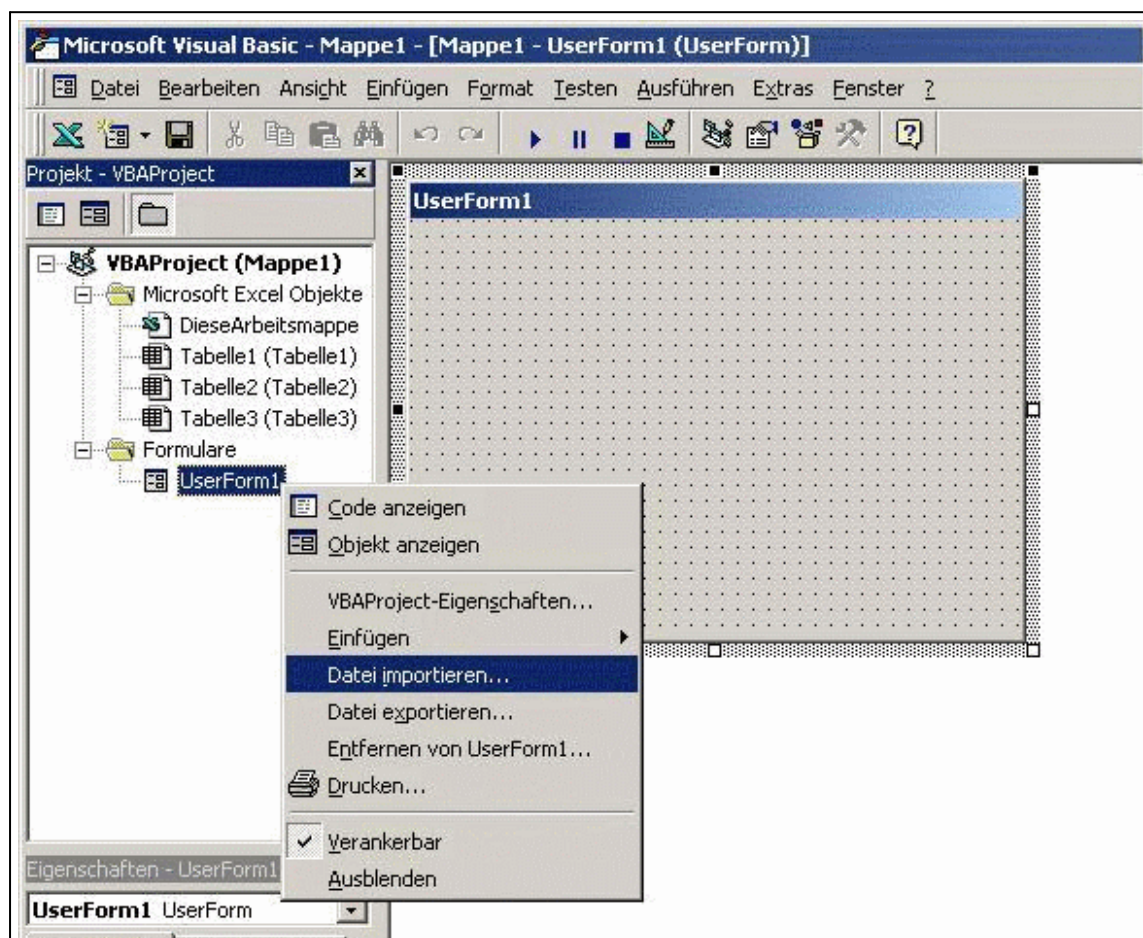
C:\Programme\DEDITEC\DELIB\include.

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.



1.1.6.8. Einbinden der DELIB in LabVIEW

1.1.6.8.1. Einbinden der DELIB in LabVIEW

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

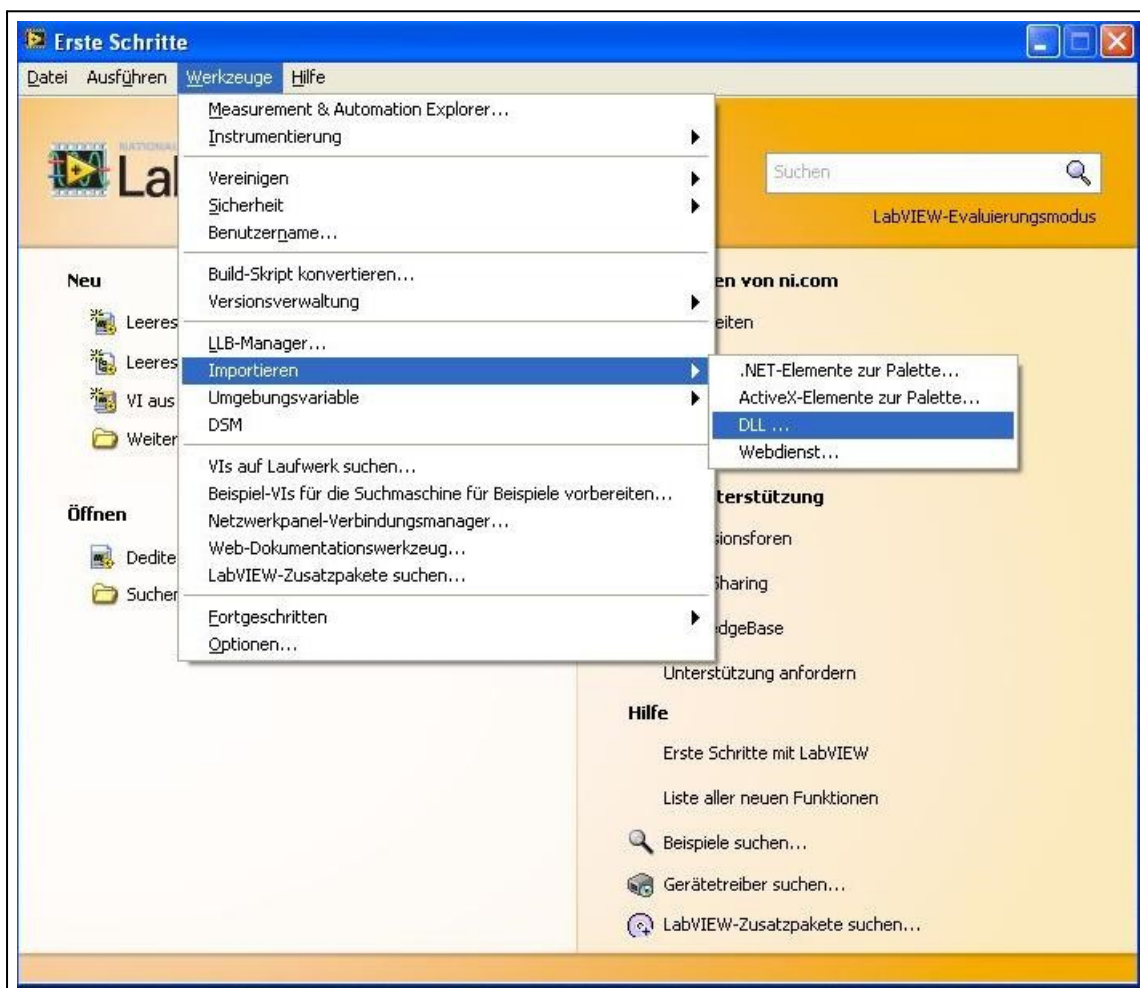
Beschreibung der Einbindung der "delib.dll" in LabVIEW Version 11

- Die benötigten Dateien für LabVIEW befinden sich im Verzeichnis

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

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

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



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

DLL importieren

Erstellungs- oder Aktualisierungsmodus angeben

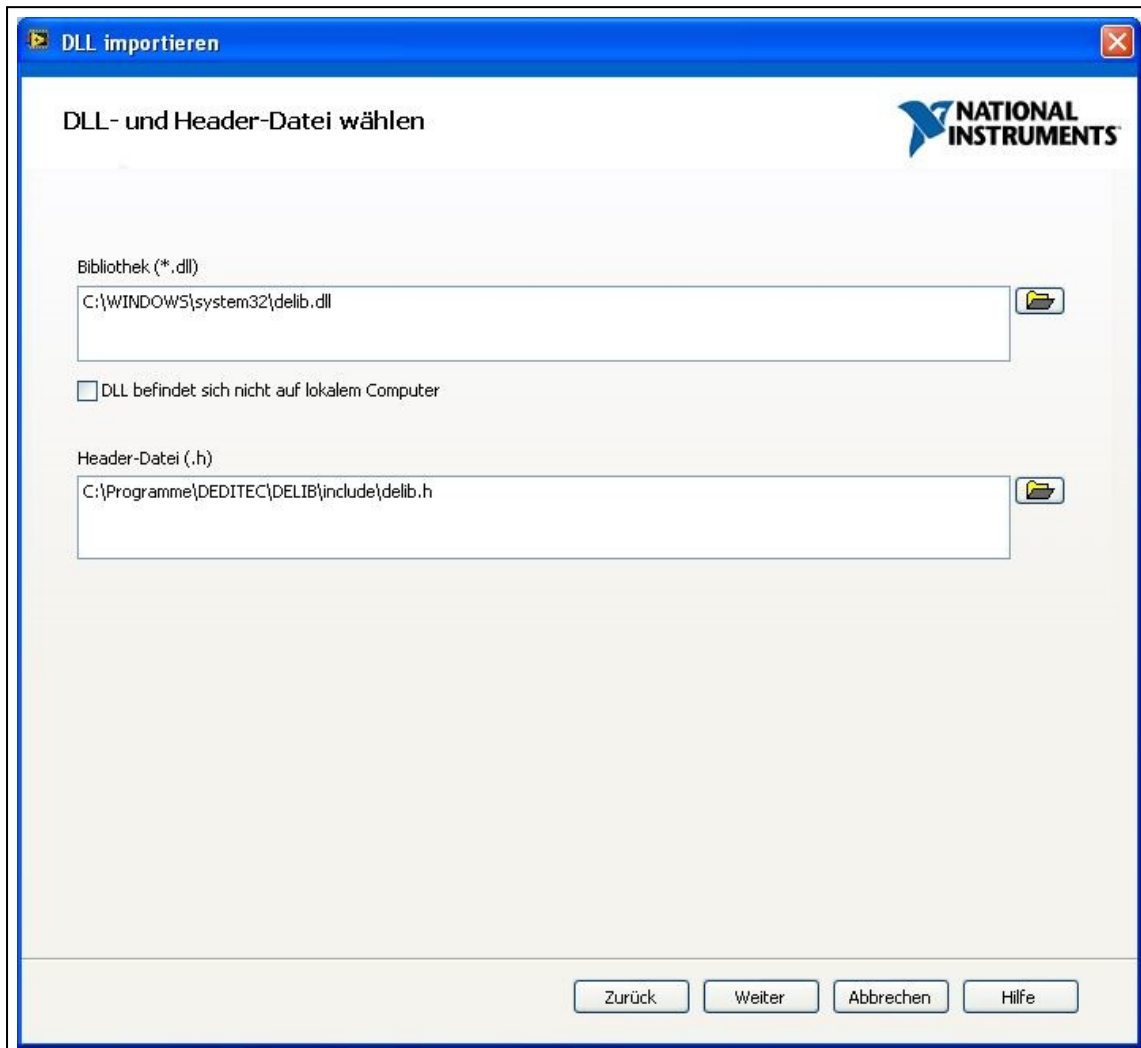
☒ VIs für DLL erstellen
Erzeugt VIs basierend auf der vorliegenden DLL- und Header-Datei.

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

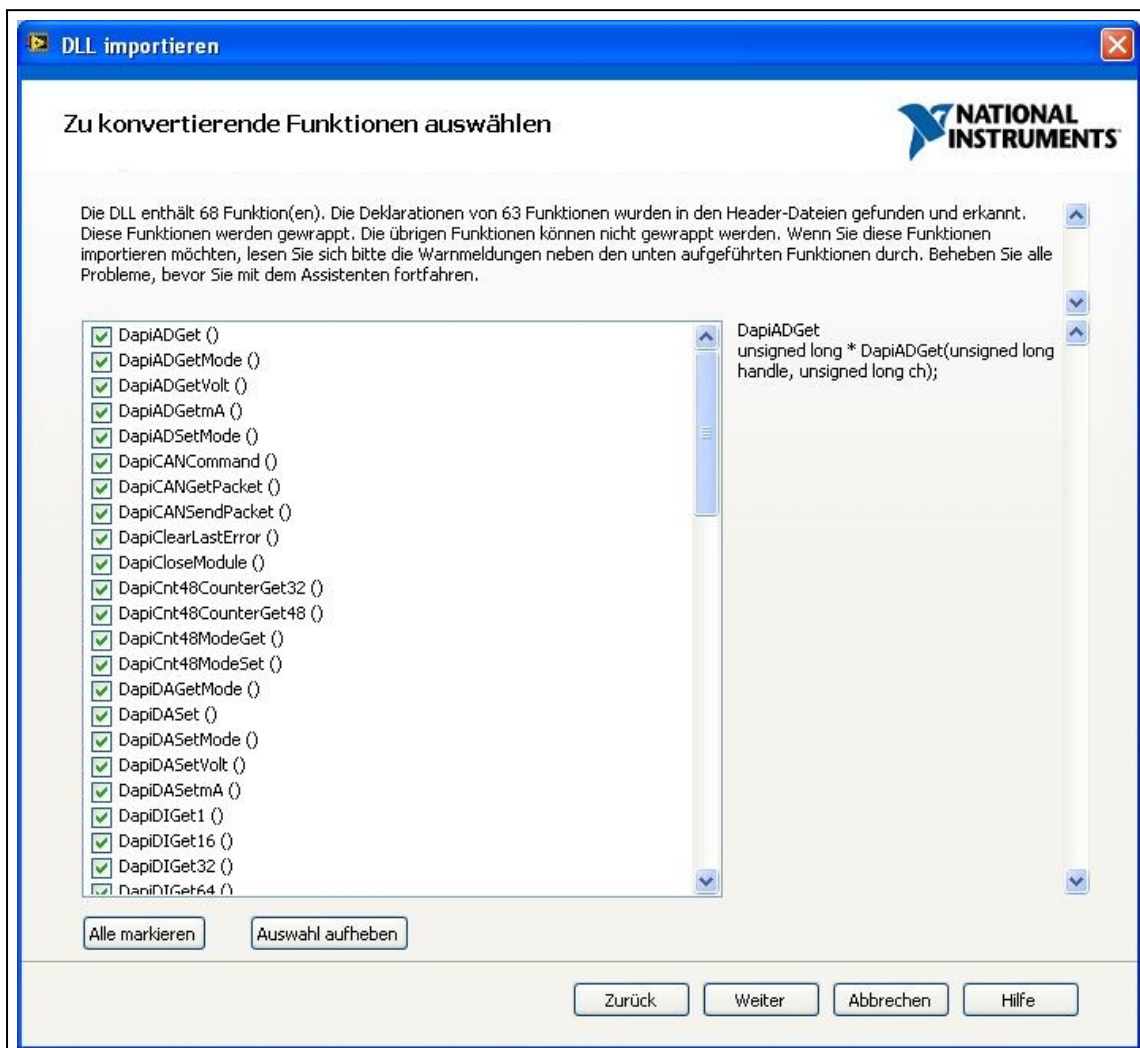
Projekt	DLL-Datei	Datum

Zurück Weiter Abbrechen Hilfe

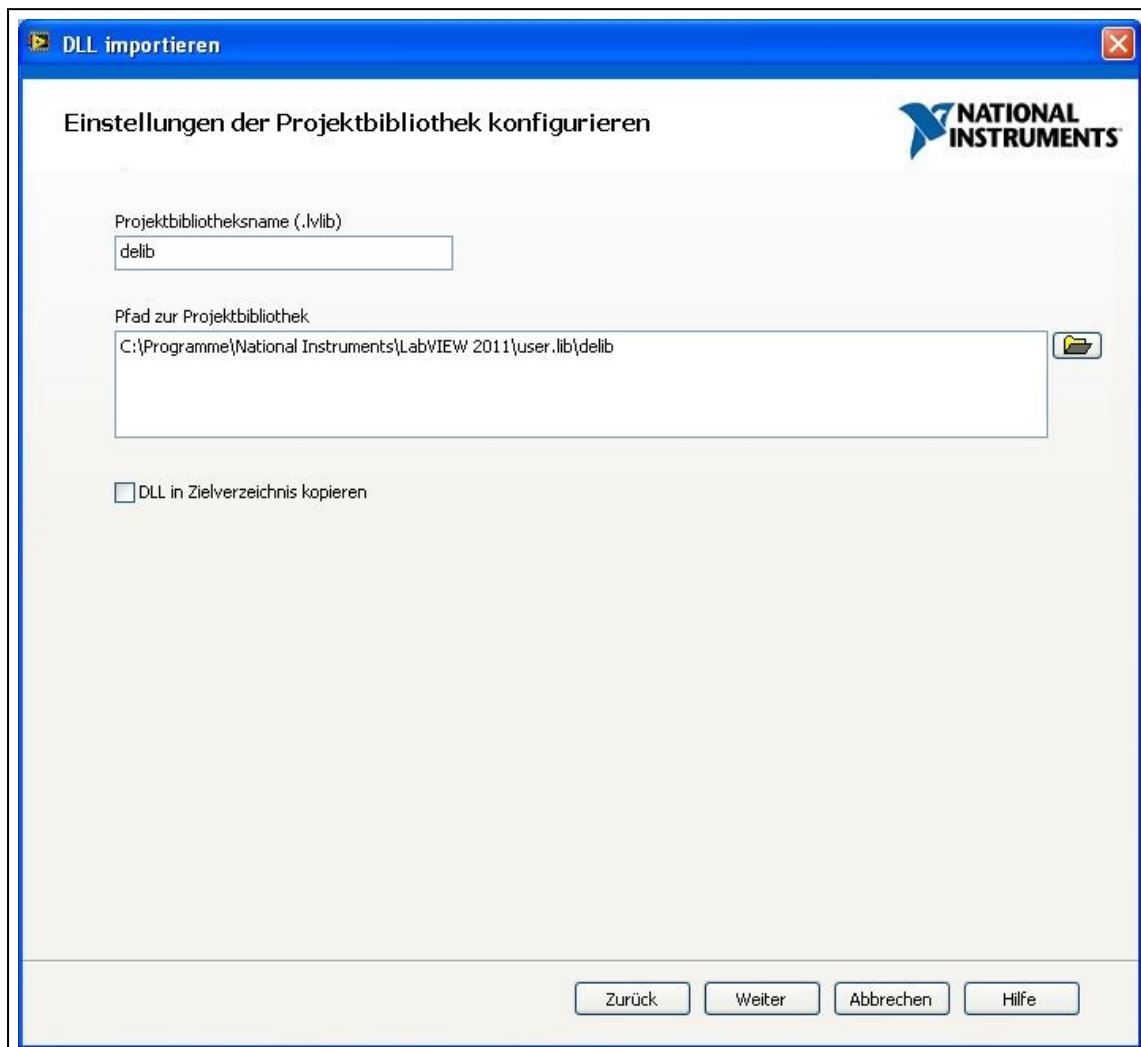
- Im nächsten Fenster über die Browser-Buttons den Speicherort der delib.dll und der delib.h Datei angeben und mit "Weiter" fortfahren.



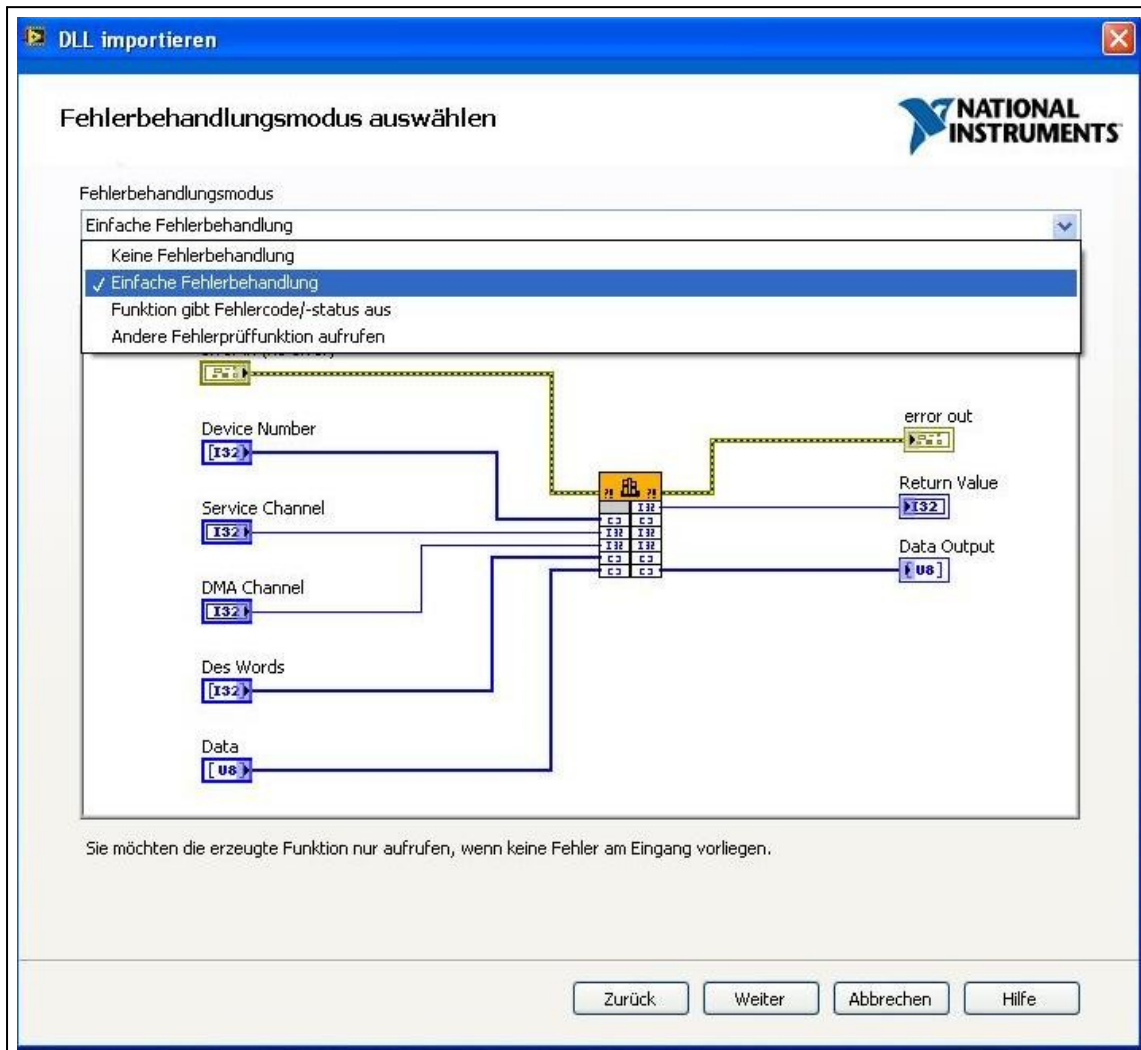
- Nochmals auf "Weiter" klicken um fortzufahren.
- Die Header-Datei wird nun analysiert. Anschließend fahren Sie im folgendem Fenster wieder mit "Weiter" fort.



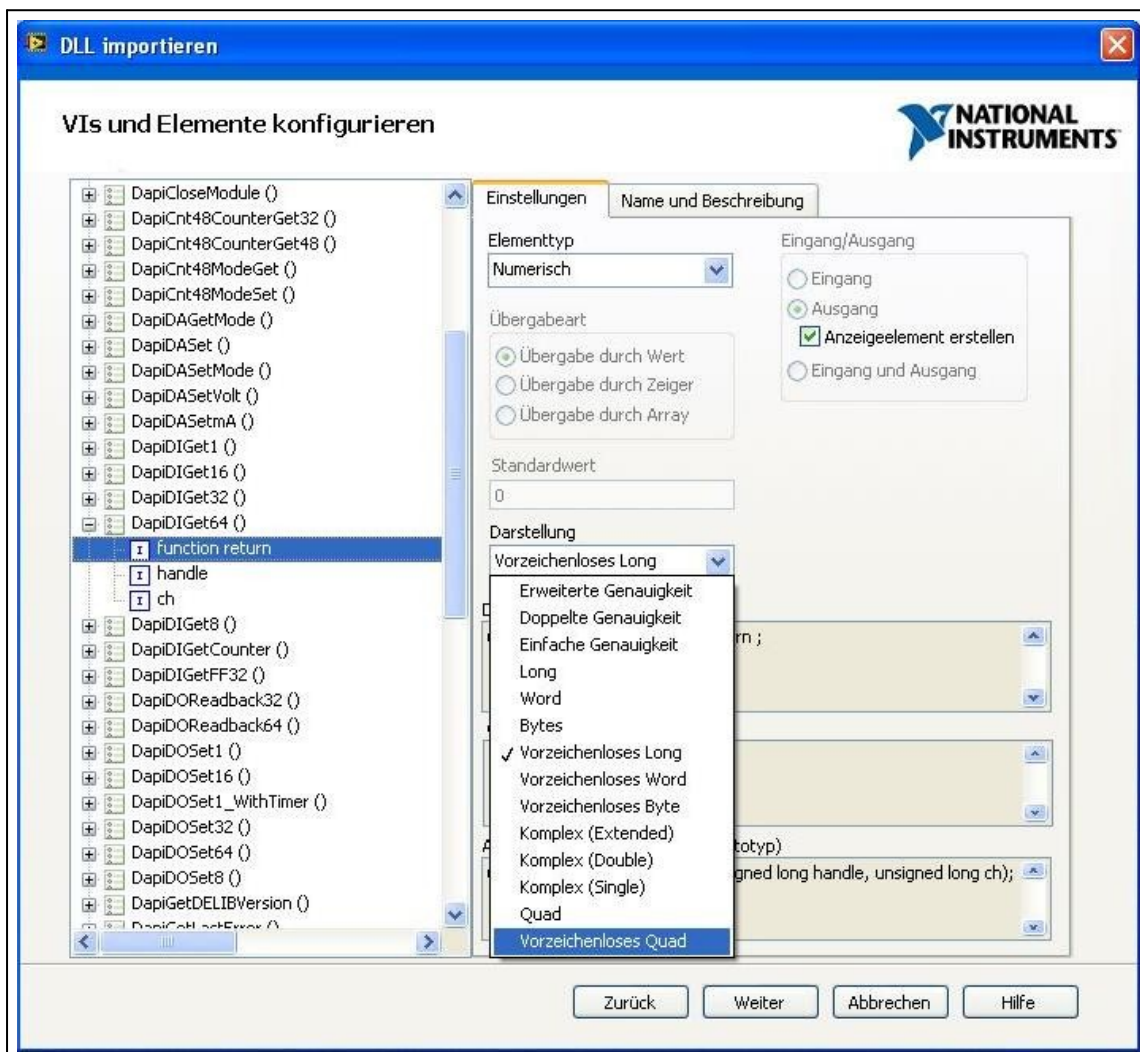
- Den weiteren Anweisungen folgen, bzw. die Konfiguration und den Speicherort für die VIs anpassen.



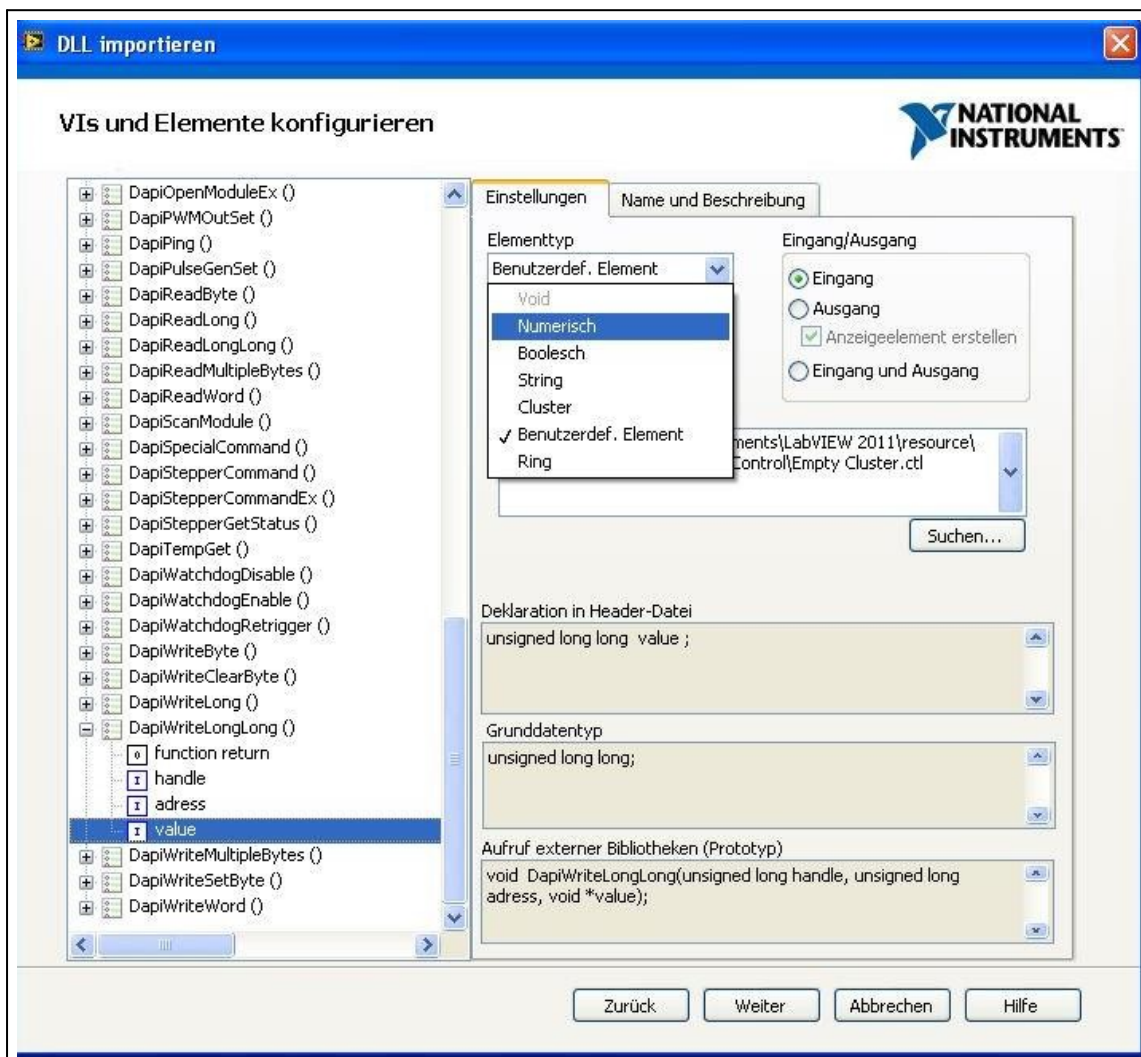
- Im folgendem Fenster wählen Sie im Drop-Down-Menü die Option "Einfache Fehlerbehandlung" aus und fahren mit "Weiter" fort.



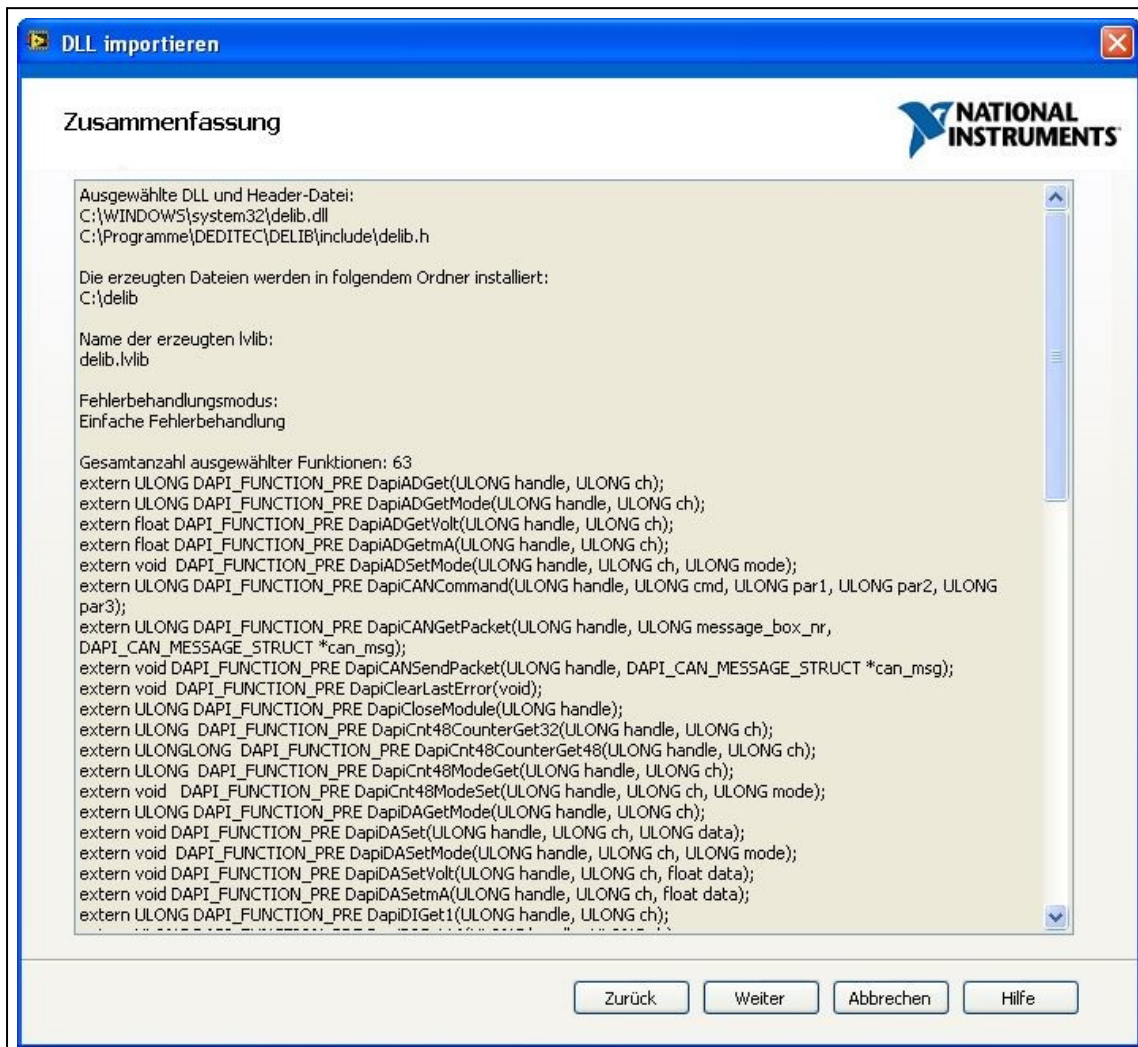
- Bei VIs die mit 64-Bit Werten arbeiten, muss die Darstellung von "Vorzeichenloses Long" in "Vorzeichenloses Quad" geändert werden.
- Folgende VIs müssen bearbeitet werden:
 - DapiCNT48CounterGet48 (function return)
 - DapiDIGet64 (function return)
 - DapiDOSet64 (data)
 - DapiDOReadBack64 (function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



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



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



- Die VIs werden nun erzeugt und können verwendet werden.

1.1.6.8.2. 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);
```

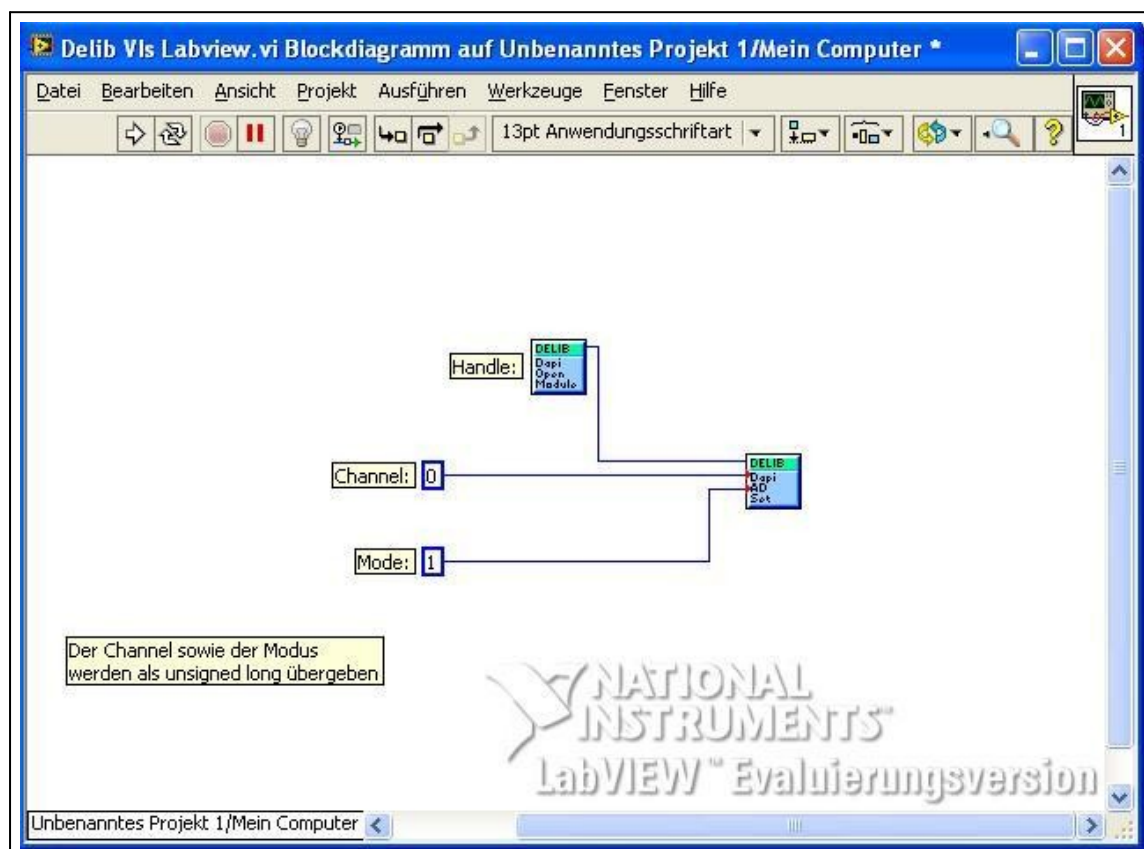
Alternativ kann man auch folgende Schreibweise verwenden:

```
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:



1.1.6.8.3. 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:

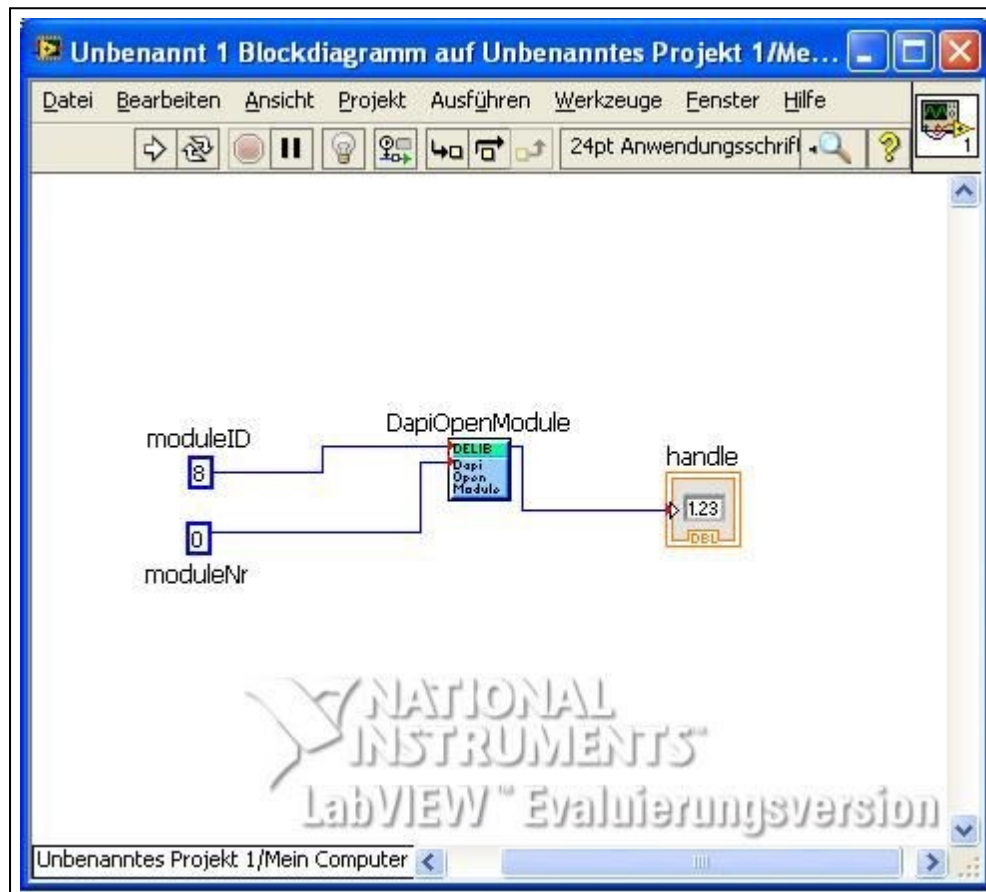
```
handle = DapiOpenModule(RO_ETH, 0); // öffnet ein RO-ETH-Modul mit Modul-Nr 0.
```

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:



1.1.6.9. Einbinden der DELIB in Java

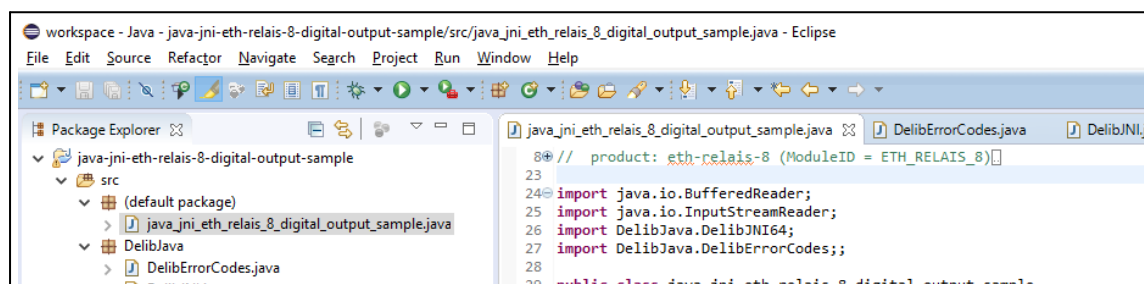
Die benötigten Dateien für Java befinden sich, je nach DELIB-Installation, in folgendem Verzeichnis

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.



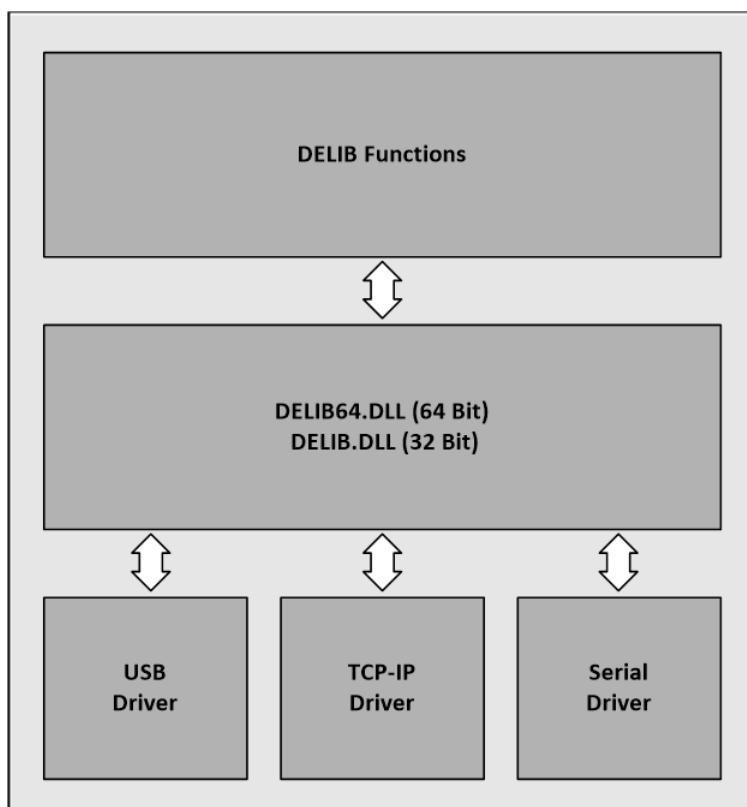
1.2. DELIB-Treiberbibliothek

Mit unserem neuen ICT-Tool, lässt sich Ihr Modul ganz unkompliziert konfigurieren, testen, diagnostizieren und flashen.

Eine ausführlichere Beschreibung zu den einzelnen Funktionen, finden Sie im Kapitel **ICT-Tool**.

1.2.1. 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

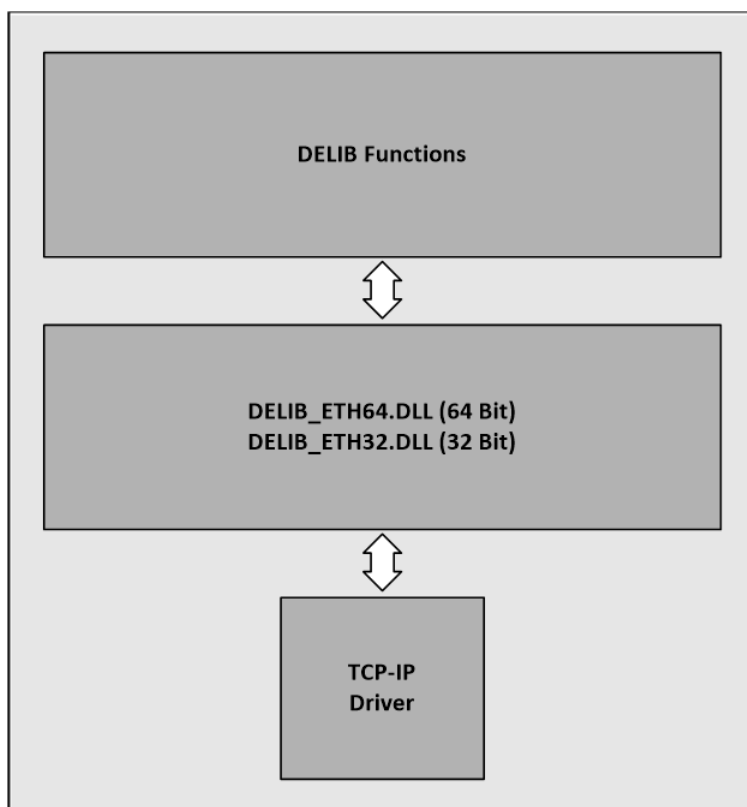
Während die DELIB für ALLE Produkte zur Verfügung steht, wird bei der DELIB-ETH auf keine weiteren Treiber zugegriffen (wie z. B. USB).

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.



1.2.1.1. 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.

1.2.1.2. 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

1.2.1.3. 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.

1.2.2. DELIB Setup

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

Anschließend werden Sie durch den Konfigurationsprozess und den Funktionstest für unsere verschiedenen Produkte geführt.

Unsere aktuelle Version des DELIB-Setups für Linux und Windows in der 32-Bit- oder 64-Bit-Version können Sie von unserer Homepage herunterladen.

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

Das DELIB-Setup führt Sie Schritt für Schritt durch die Installation der DELIB-Treiberbibliothek.

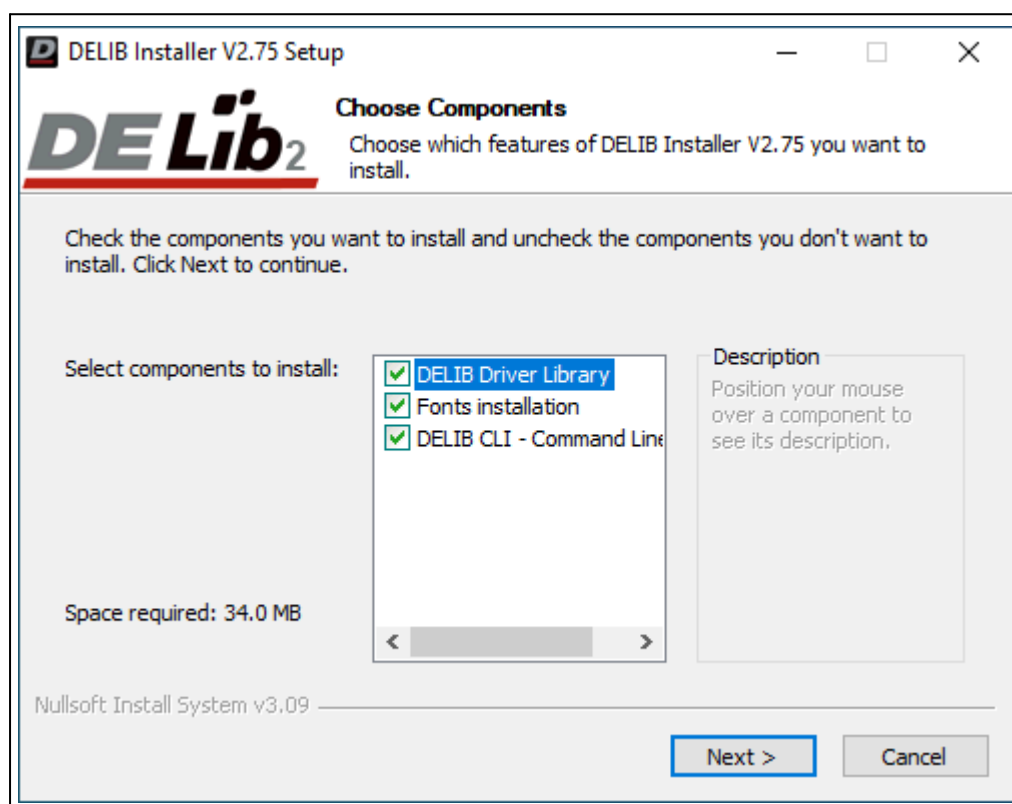


Zusätzlich zur DELIB-Treiberbibliothek haben Sie die Möglichkeit, unsere "DELIB CLI - Command Line" Software mit zu installieren.

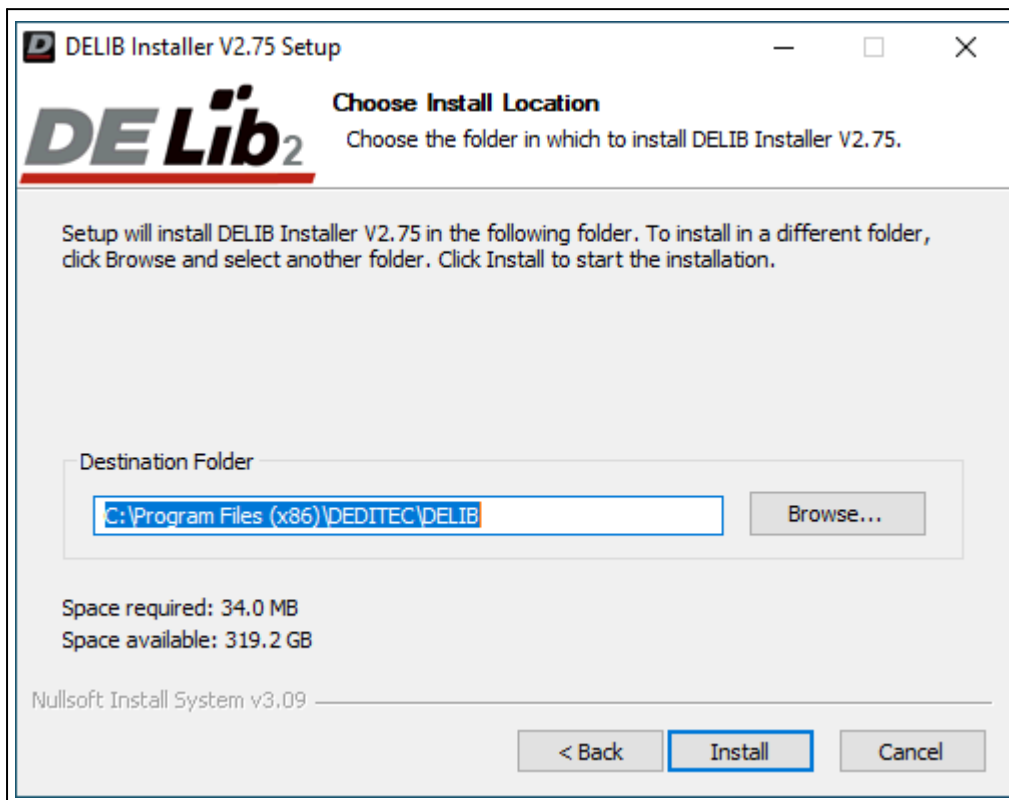
Mit diesem Programm können Sie Befehle direkt über die Kommandozeile ausführen. Mehr Infos dazu finden Sie im folgenden Kapitel:

DELIB CLI (command-line interface) für Windows

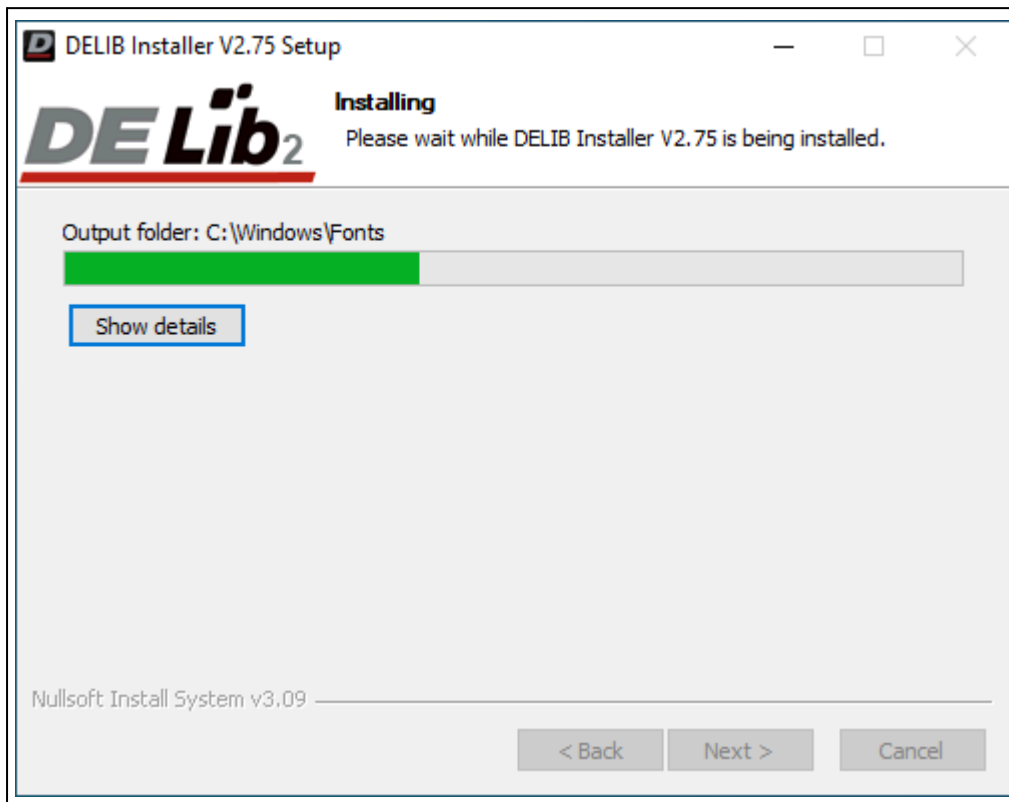
DELIB CLI (command-line interface) für Linux



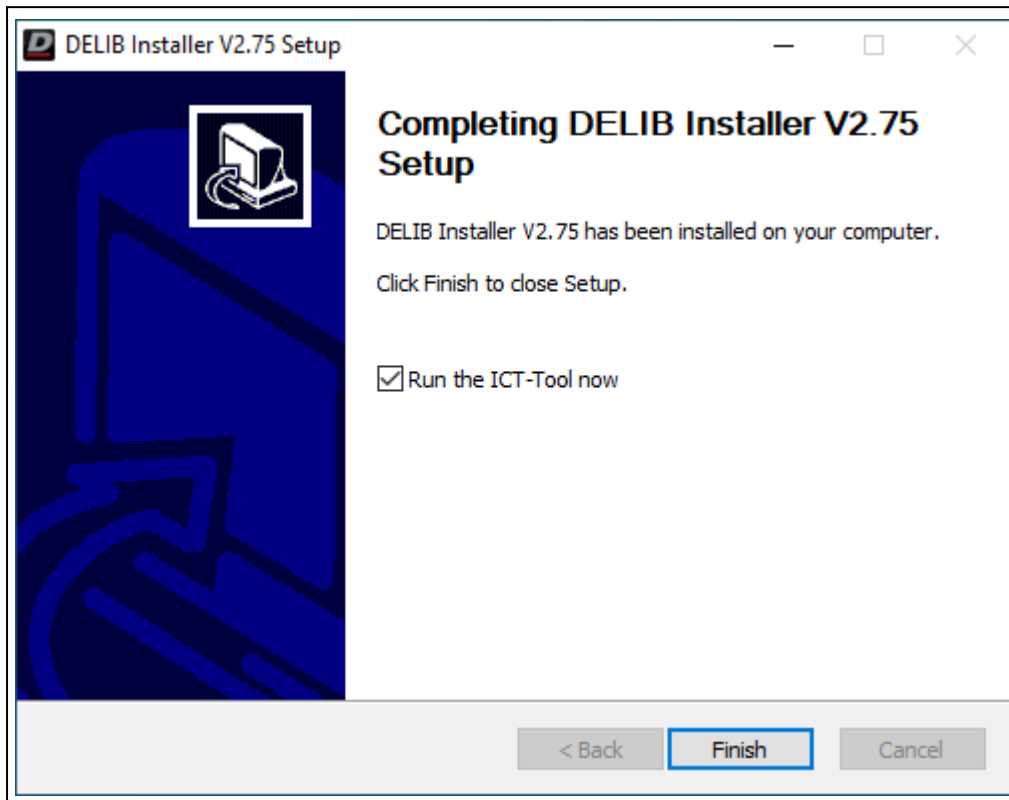
Wählen Sie den gewünschten Installationspfad und bestätigen Sie mit der Schaltfläche "Installieren".



Fortschritt der Installation der Treiberbibliothek.



Nach Abschluss der Installation schließen Sie das Setup mit der Schaltfläche "Finish".



Mit "Run the ICT-Tool now" können Sie die Option, das ICT-Tool direkt nach dem Beenden des DELIB-Setup zu starten.

Zusätzlich finden Sie das ICT-Tool und das DELIB-CLI nun in Ihrem Windows-Verzeichnis "Start".

1.2.3. ICT-Tool

Mit unserem neuen ICT-Tool, lässt sich Ihr Modul ganz unkompliziert konfigurieren, testen, diagnostizieren und flashen.

Eine ausführlichere Beschreibung zu den einzelnen Funktionen, finden Sie im Kapitel **ICT-Tool**.

1.3. 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

1.3.1. Installation DELIB Sample Sources

Die DELIB Sample Sources können Sie über ein eigenständiges Setup installieren.

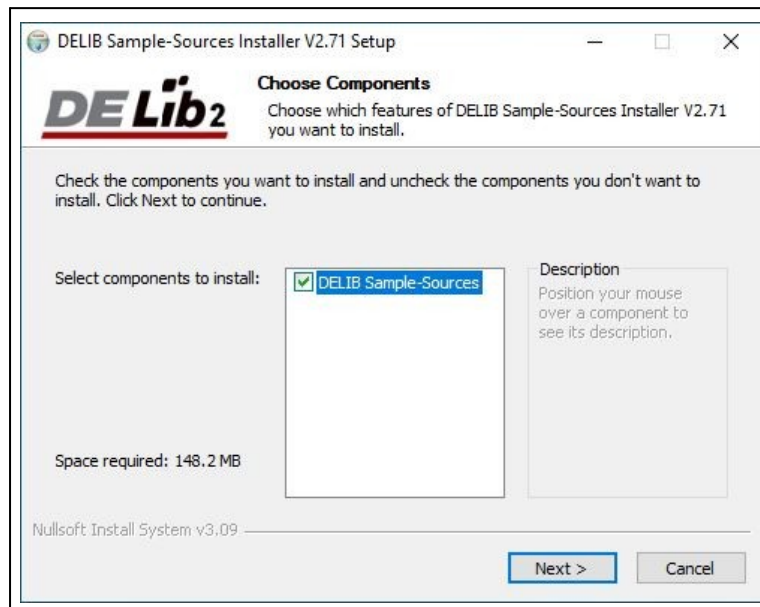
Eine aktuelle Version der Sample Sources finden Sie auch im Internet unter folgendem Link im Reiter Samples:

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

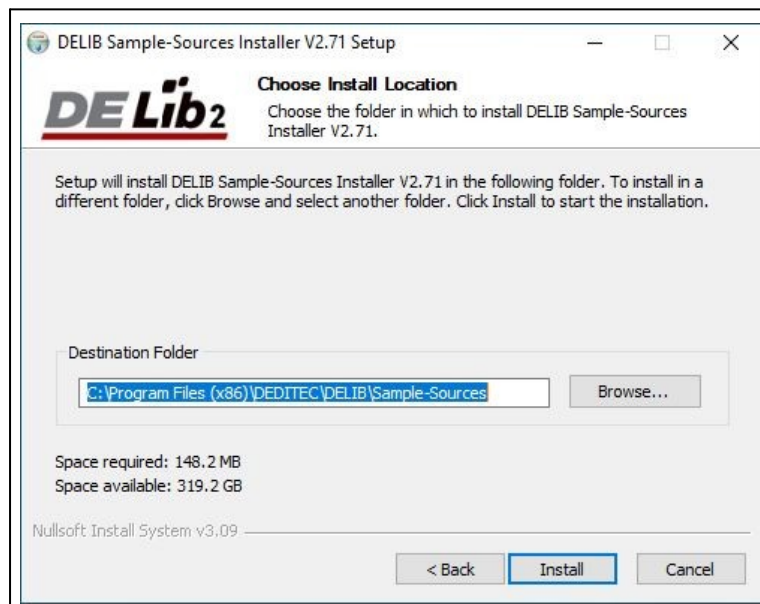
Startbild des DELIB Sample Sources Installer



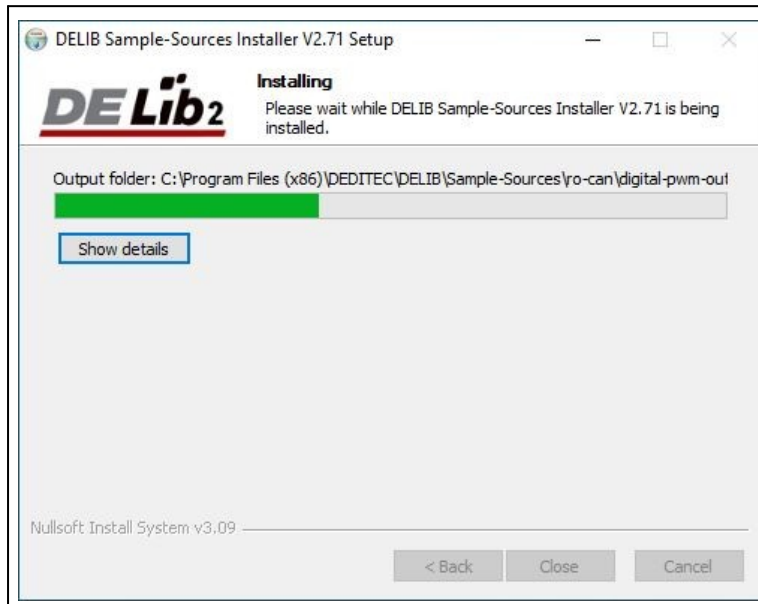
Drücken Sie Next.



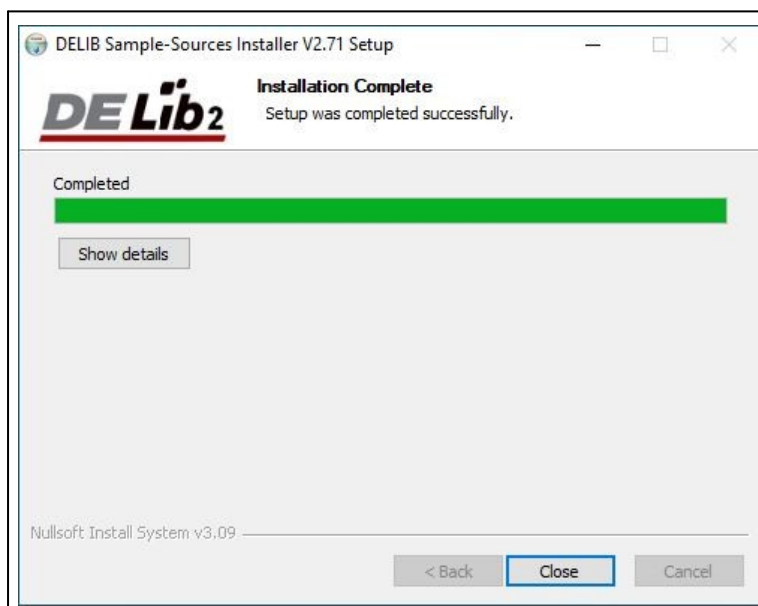
Wählen Sie den Installationsordner und drücken Sie Install.



Die DELIB Sample Sources werden nun installiert.



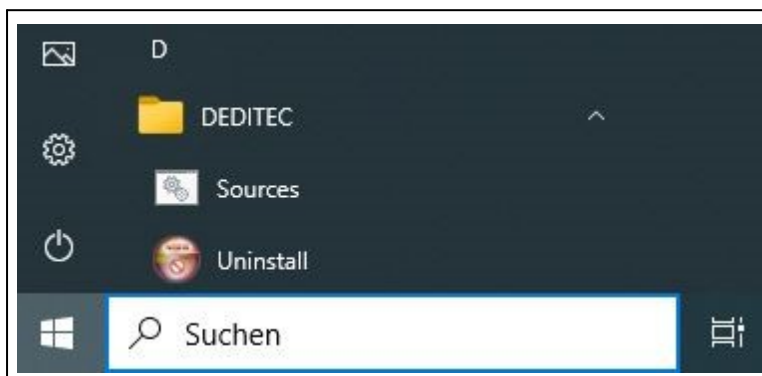
Die DELIB Sample Sources wurden erfolgreich installiert. Drücken Sie Close um die Installation zu beenden.



1.3.2. Benutzung der DELIB Sample Sources

After installing the DELIB Sample Sources you can find them under

Start → Programs → DEDITEC → Sources

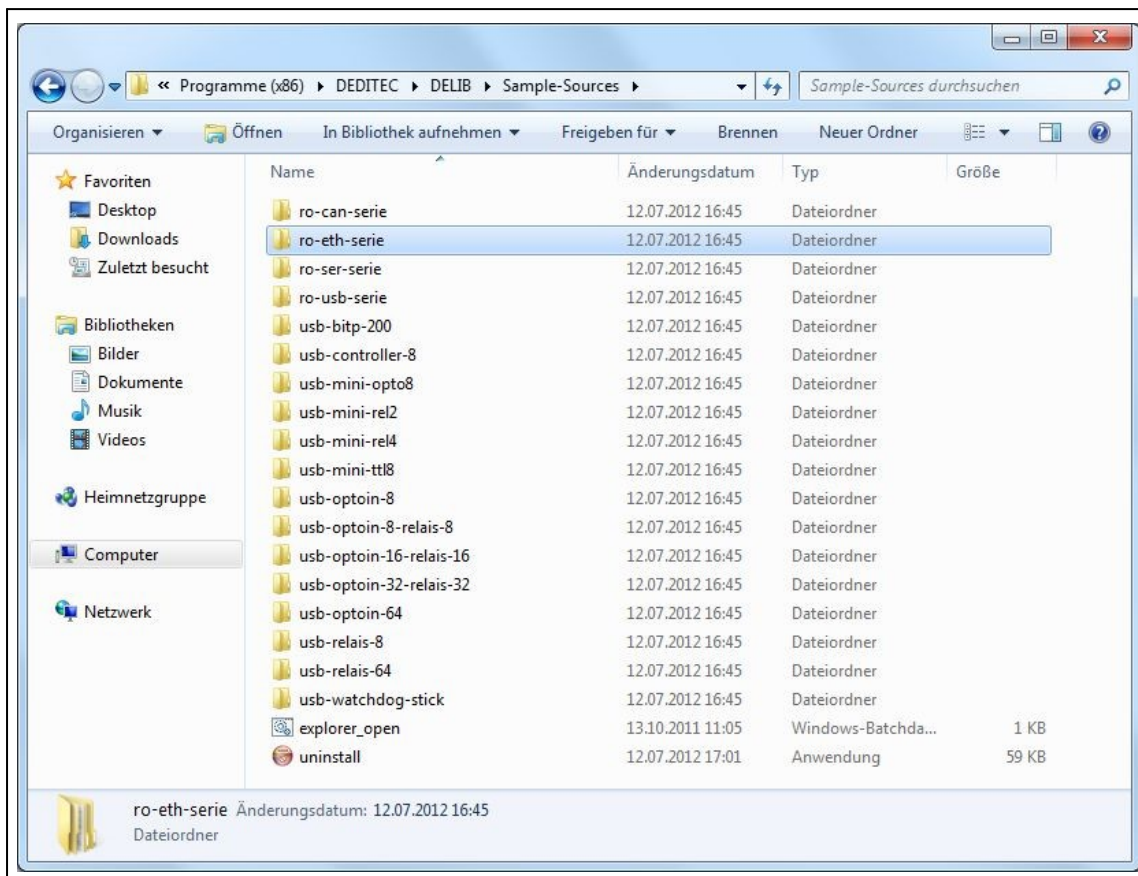


Now the Windows Explorer opens with an overview of all products for which a sample program is available.

1.3.2.1. 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.

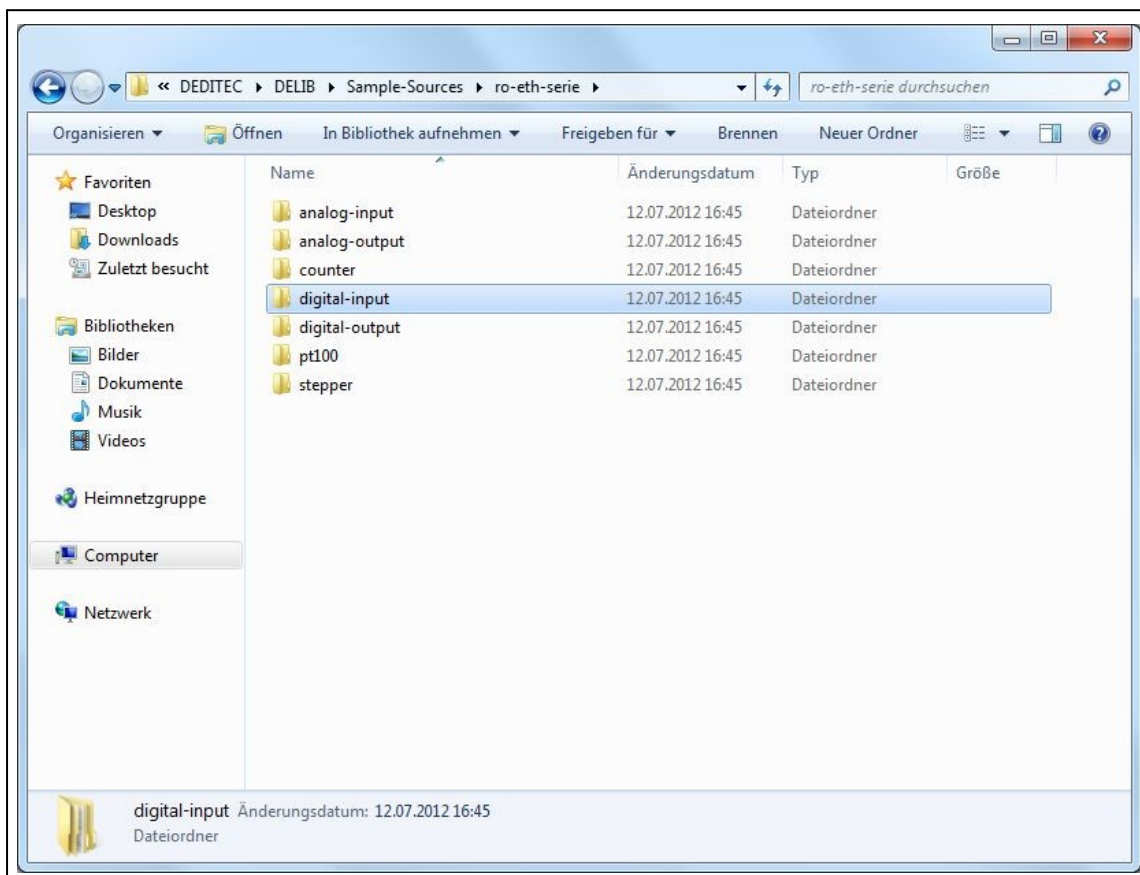
Da es sich um ein RO-ETH-Produkt handelt, wählen bzw. öffnen Sie den Ordner ro-eth-serie.



1.3.2.2. Schritt 2 - Kategorieauswahl

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

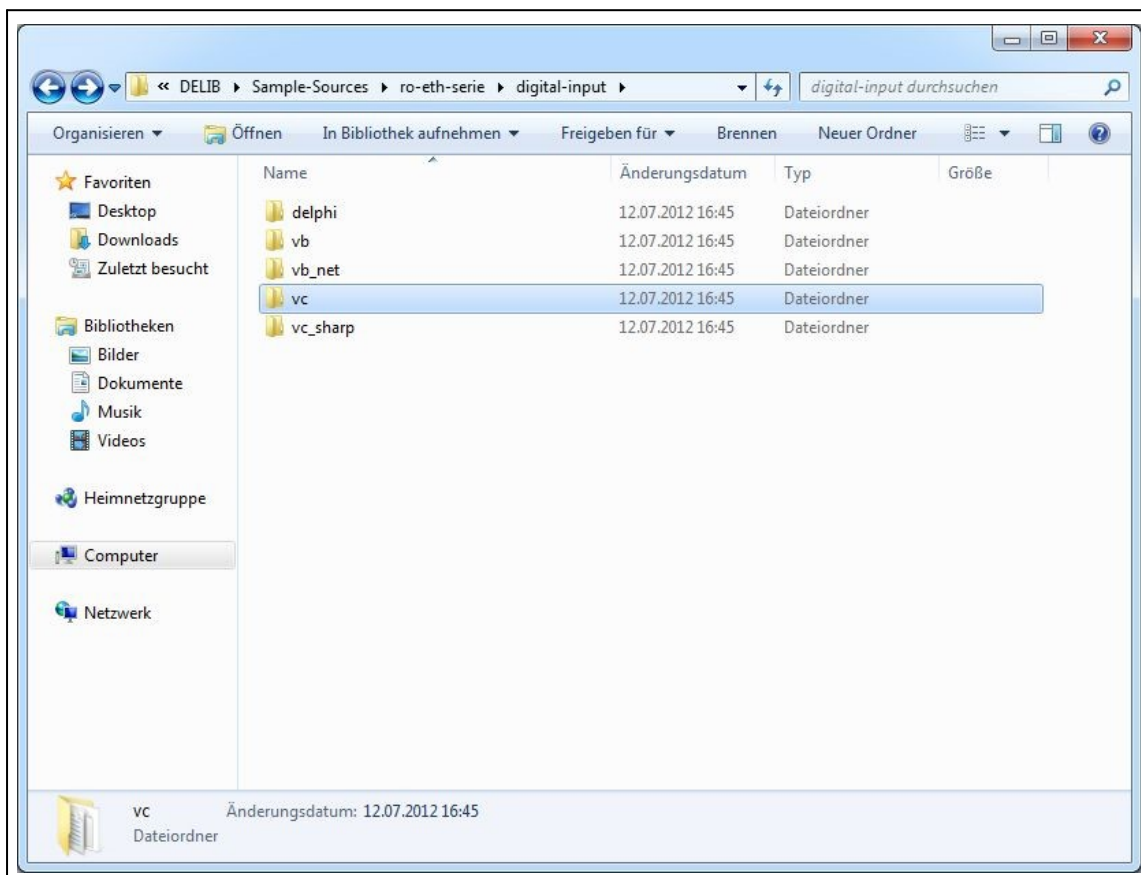
Da wir uns in diesem Beispiel auf die digitalen Eingänge konzentrieren, wählen Sie die Kategorie digital-input



1.3.2.3. Schritt 3 - Programmiersprachenauswahl

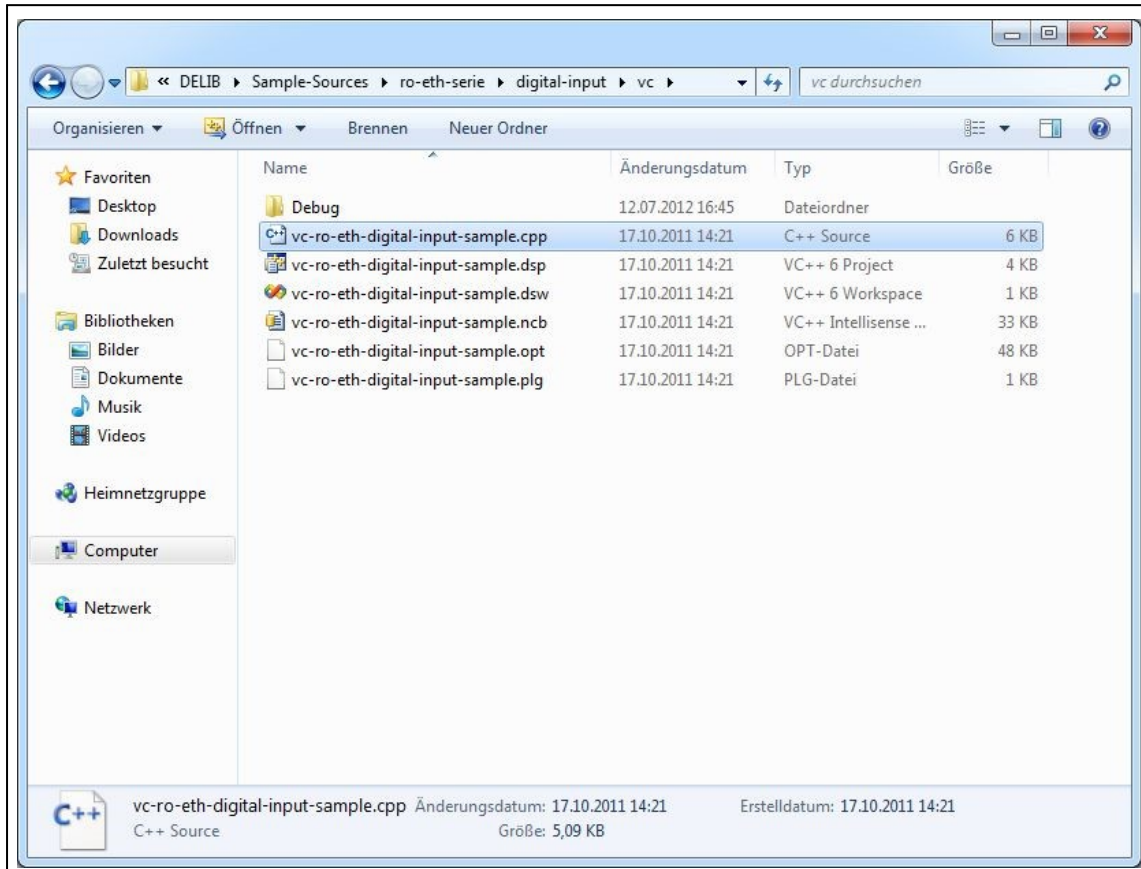
In diesem Schritt sehen Sie alle verfügbaren Programmierbeispiele der gewählten Kategorie, sortiert nach Programmiersprachen.

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

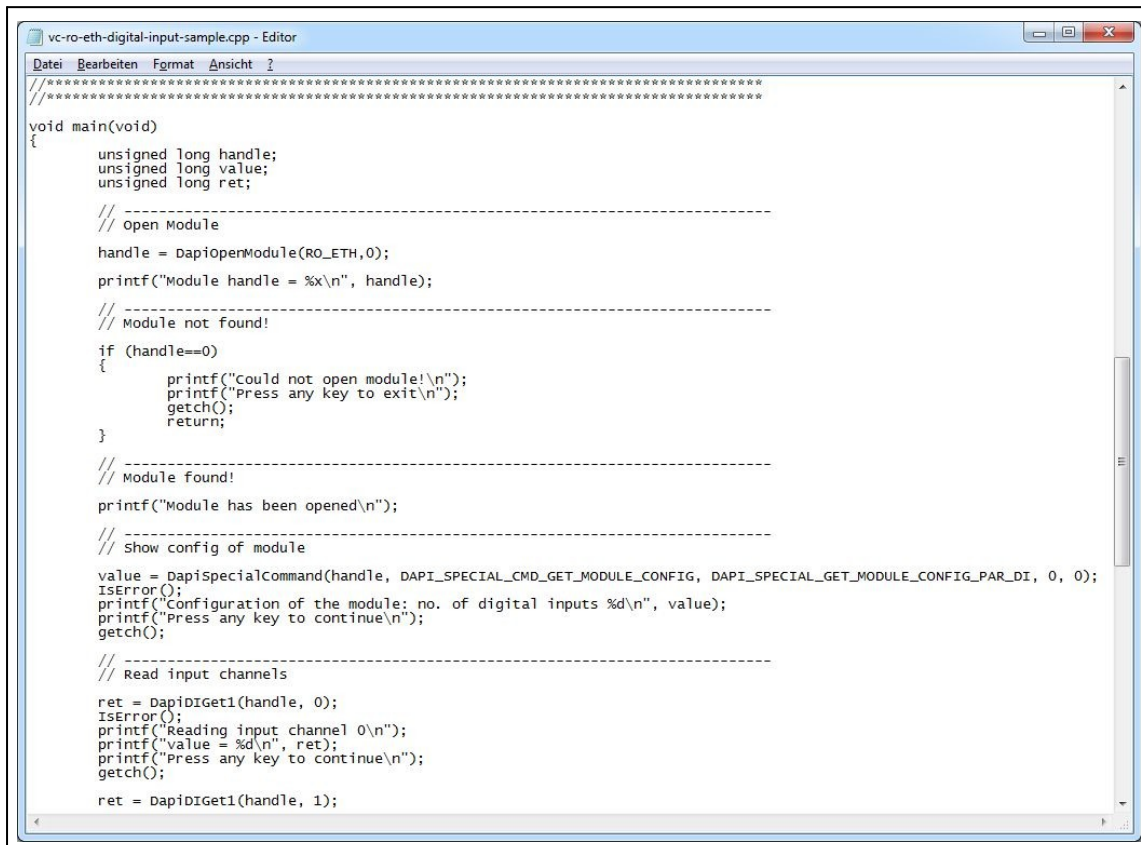


1.3.2.4. Schritt 4 - Quellcode

Nach Auswahl der Programmiersprache erhalten Sie folgende Übersicht:



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



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

    // -----
    // Open Module

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

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

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

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

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

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

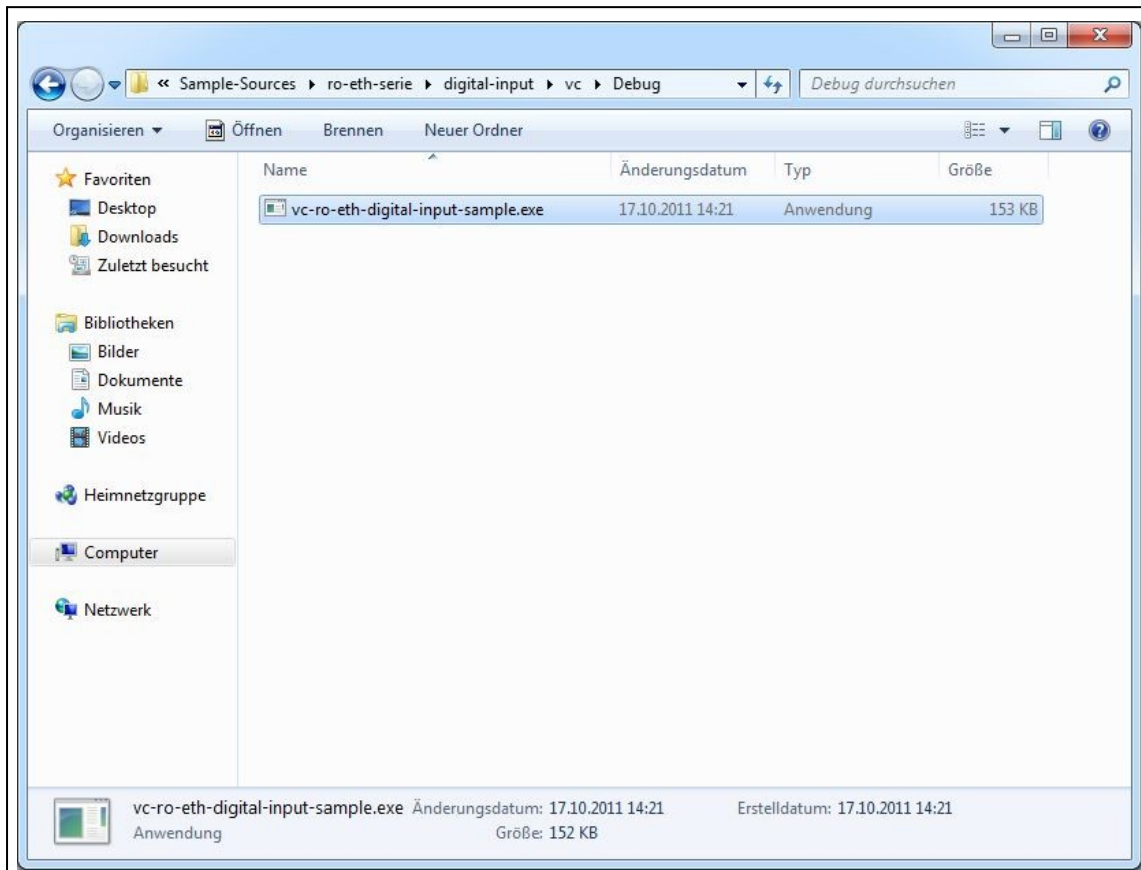
    value = DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_GET_MODULE_CONFIG, DAPI_SPECIAL_GET_MODULE_CONFIG_PAR_DI, 0, 0);
    ISError();
    printf("Configuration of the module: no. of digital inputs %d\n", value);
    printf("Press any key to continue\n");
    getch();

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

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

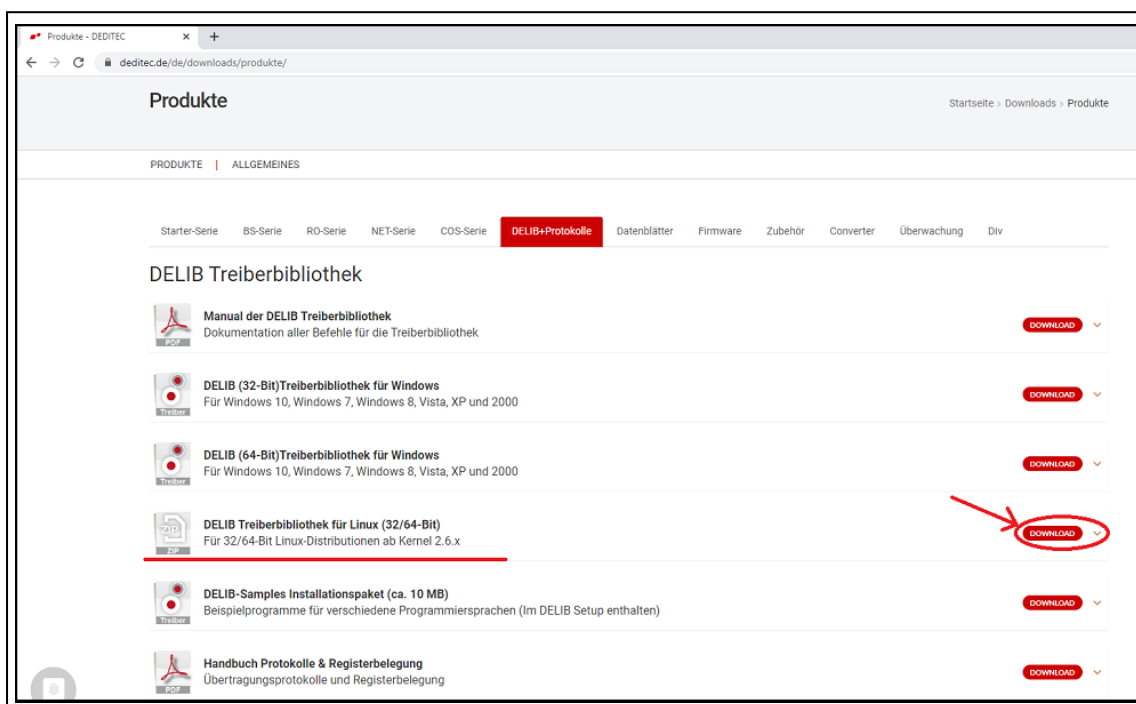
    ret = DapiDIGet1(handle, 1);
```

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

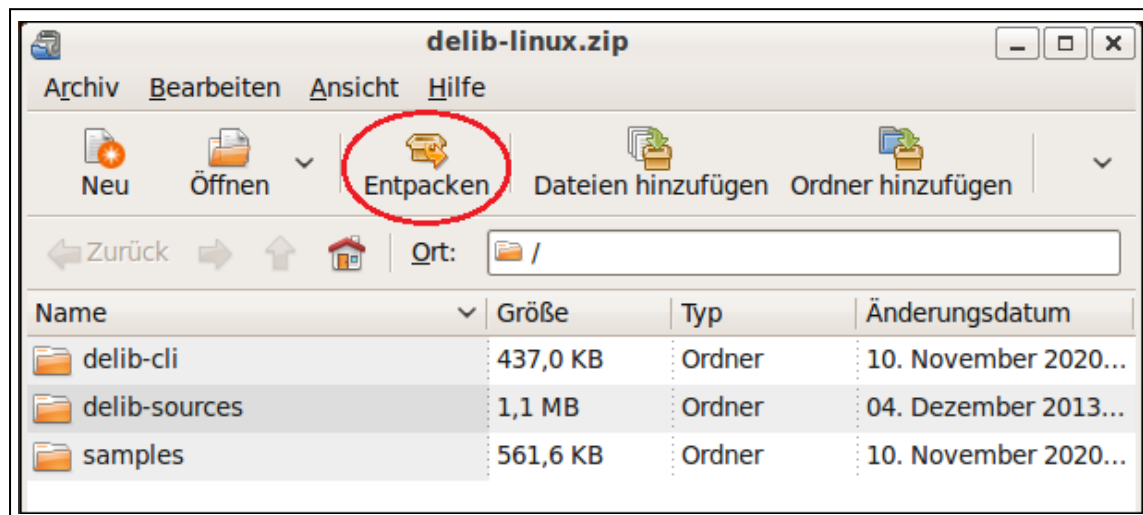


1.4. DELIB für Linux

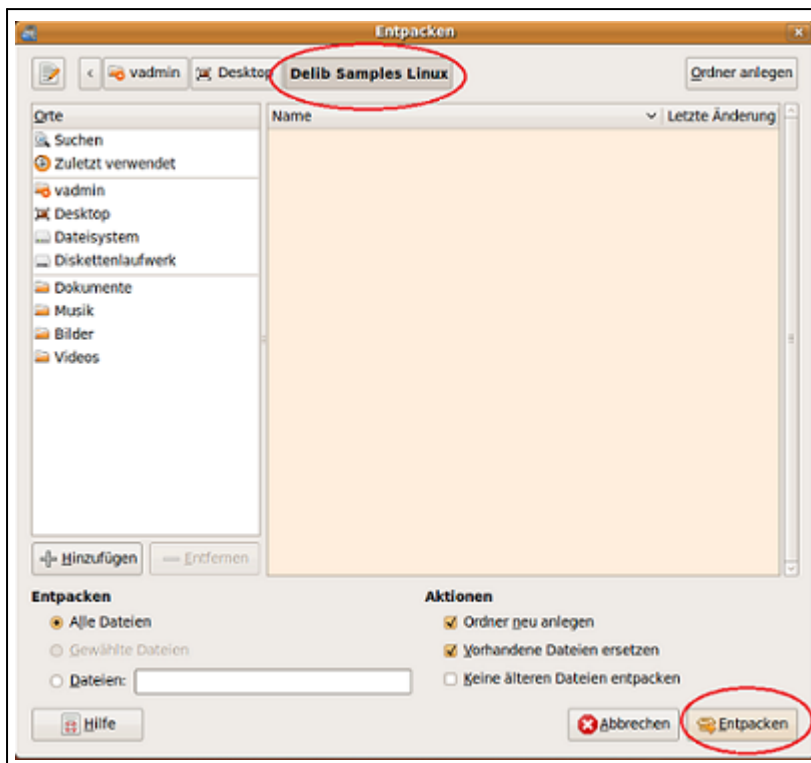
Laden Sie sich die Delib Linux Treiberbibliothek unter "www.deditec.de/de/downloads/produkte/" im Reiter „DELIB+Protokolle“ oder unter "www.deditec.de/media/zip/delib/delib-linux.zip" direkt auf ihr Linux-System.



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.



1.4.1. Verwenden der DELIB-Treiberbibliothek für Linux

1.4.1.1. Delib USB-Sample in Linux

Voreinstellungen

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

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

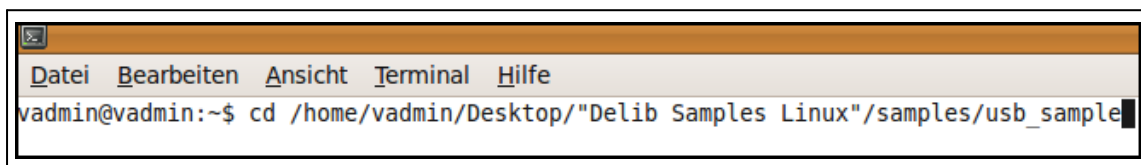
```
23
24 #include <stdio.h>
25 #include <stdlib.h>
26 #include <unistd.h>
27
28 #include "../../delib-sources/delib/library/delib/delib.h"
29
30 int main()
31 {
32     ULONG i;
33     ULONG handle=0;
34
35     printf("\n\n");
36     printf("-----\n");
37     printf("-----\n");
38     printf("-----\n");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



```

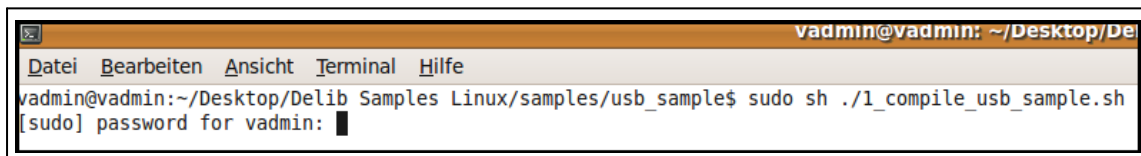
Datei Bearbeiten Ansicht Terminal Hilfe
vadmin@vadmin:~$ cd /home/vadmin/Desktop/"Delib Samples Linux"/samples/usb_sample

```

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:

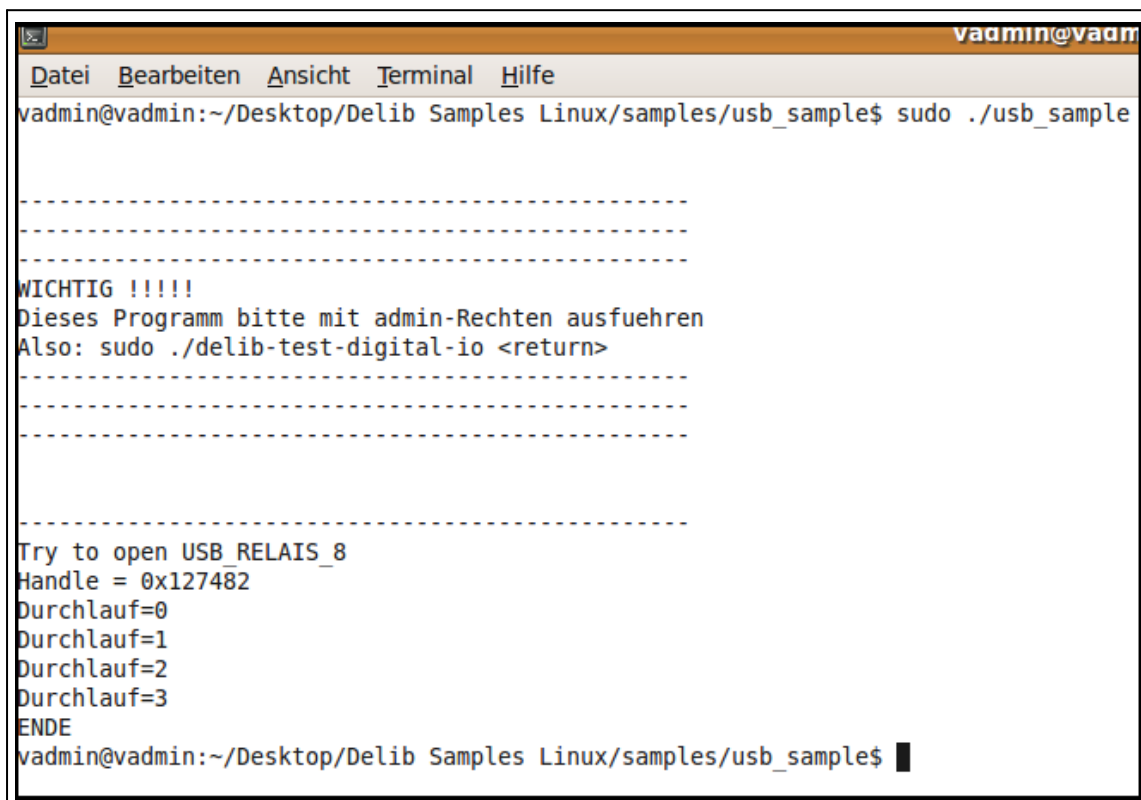
```

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.

1.4.1.2. Delib ETH-Sample in Linux

Voreinstellungen

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

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

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

Die Konfiguration der ETH-Module können über das ICT-Tool, 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
```

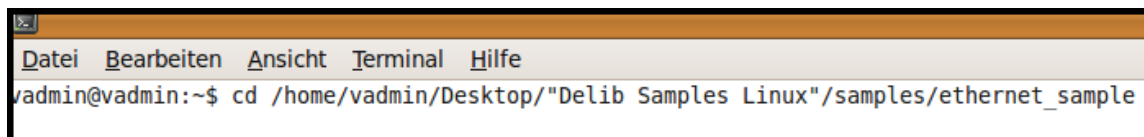
Auskommentieren,
falls keine digitalen
Eingänge vorhanden

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.

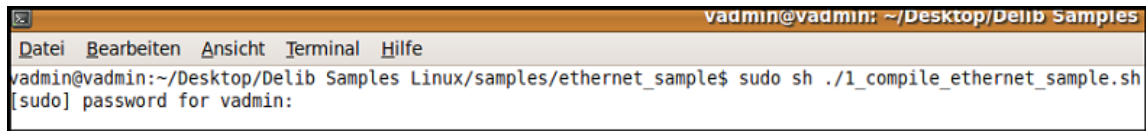
A screenshot of a terminal window with a menu bar containing 'Datei', 'Bearbeiten', 'Ansicht', 'Terminal', and 'Hilfe'. The command prompt shows 'vadmin@vadmin:~\$ cd /home/vadmin/Desktop/"Delib Samples Linux"/samples/ethernet_sample'.

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

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

- Möchten Sie das Modul über eine unverschlüsselte TCP Verbindung ansteuern, verwenden Sie die Datei „1_compile_ethernet_sample.sh“
- Möchten Sie das Modul über eine verschlüsselte TCP Verbindung ansteuern, verwenden Sie die Datei „2_compile_ethernet_sample_with_encryption.sh“

Geben Sie, falls nötig, Ihr Benutzerkennwort ein.



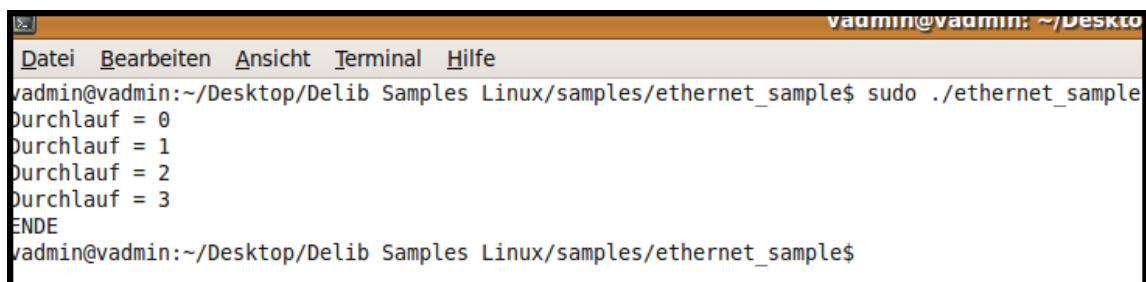
```
vadmin@vadmin: ~/Desktop/Delib Samples
Datei Bearbeiten Ansicht Terminal Hilfe
vadmin@vadmin:~/Desktop/Delib Samples Linux/samples/ethernet_sample$ sudo sh ./1_compile_ethernet_sample.sh
[sudo] password for vadmin:
```

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

Es wurde die Datei "ethernet_sample" dem Verzeichnis hinzugefügt.

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

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



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

Das Programm wird nun ausgeführt.

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

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

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

Definition für USB-Module (Linux)

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

Definition für ETH-Module (Linux)

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

Hinweis:

Die einzelnen Parameter werden nur durch ein Leerzeichen getrennt.

Groß und Kleinschreibung wird hierbei nicht beachtet.

Parameter

Befehl	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

1.4.2.1. Konfiguration des DELIB CLI

Voreinstellungen

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

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

Inhalt der "delib_cli.cfg":

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

moduleID

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

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

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

moduleNR

Die moduleNR wird im ICT-Tool 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.

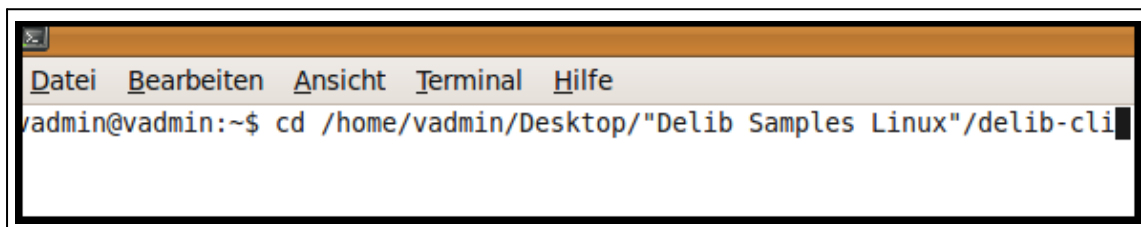
Die IP-Adresse der ETH-Module können über das ICT-Tool sowie über die Weboberfläche des Moduls eingestellt werden.

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.

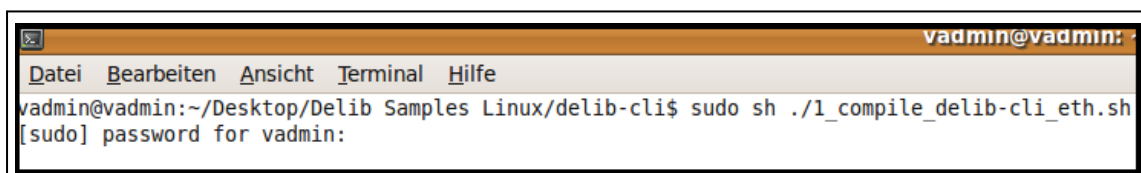


```
admin@vadmin:~$ cd /home/vadmin/Desktop/"Delib Samples Linux"/delib-cli
```

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.

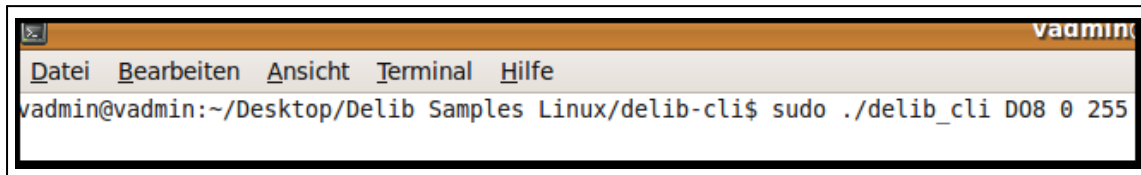


```
admin@vadmin:~/Desktop/Delib Samples Linux/delib-cli$ sudo sh ./1_compile_delib-cli_eth.sh
[sudo] password for vadmin:
```

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 command prompt shows 'vadmin@vadmin:~/Desktop/Delib Samples Linux/delib-cli\$' followed by the command 'sudo ./delib_cli D08 0 255'.

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

1.4.2.2. DELIB CLI Beispiele

Digitale Ausgänge

```
sudo delib_cli DO1 17 1
```

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

```
sudo delib_cli DO1 3 0
```

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

Digitale Eingänge

```
sudo delib_cli DI1 3
```

Beispiel eines Rückgabewertes: **1**

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

```
sudo delib_cli DI8 0 hex
```

Beispiel eines Rückgabewertes: **0xFF**

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

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

```
sudo delib_cli FF 0
```

Beispiel eines Rückgabewertes: **192**

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

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

```
sudo delib_cli FF 32
```

Beispiel eines Rückgabewertes: **65535**

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

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

```
sudo delib_cli FF 0 hex
```

Beispiel eines Rückgabewertes: **0xD00**

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

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

Analoge Ausgänge

```
sudo delib_cli AO 7 4711
```

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

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

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

Analoge Eingänge

```
sudo delib_cli AI 2
```

Beispiel eines Rückgabewertes: **1234**

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

```
sudo delib_cli AI 2 hex
```

Beispiel eines Rückgabewertes: **0x1FA**

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

1.5. Weboberfläche

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



The screenshot shows a standard web browser security warning. At the top, it says 'Authentifizierung erforderlich' with a close button (X). Below this, it states: 'Für http://192.168.1.1 sind ein Nutzernamen und ein Passwort erforderlich.' and 'Die Verbindung zu dieser Website ist nicht sicher.' There are two input fields: 'Nutzername:' and 'Passwort:'. At the bottom, there are two buttons: 'Anmelden' and 'Abbrechen'.

Der integrierte Web-Server des Moduls erfordert eine Authentifizierung

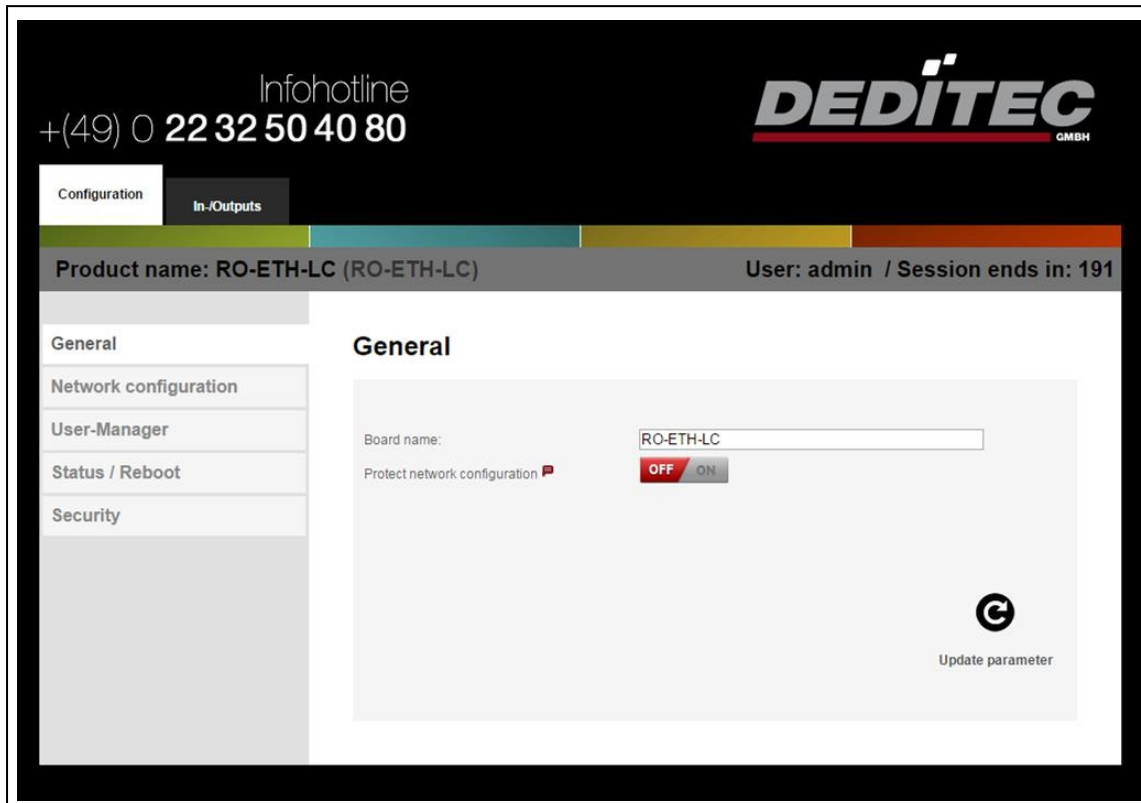
um das Modul vor unberechtigten Zugriffen zu schützen.

Standardmäßig ist folgender Benutzer eingerichtet:

Benutzername	Passwort	Rechte
admin	admin	Administratoren-Rechte

1.5.1. Konfiguration

1.5.1.1. Allgemein



Boardname

Das ist der Name des Moduls, welcher auch im ICT-Tool 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 ICT-Tool) wird in diesem Fall verhindert.

1.5.1.2. Netzwerk Konfiguration

The screenshot displays the DEDITEC web interface. At the top, 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 with menu items: 'General', 'Network configuration', 'User-Manager', 'Status / Reboot', and 'Security'. The 'Network configuration' menu item is selected. The main content area is titled 'Network configuration' and contains the following fields and controls:

- MAC: 00:C0:D5:02:00:0D
- Obtain IP address automatically (DHCP): **OFF** / ON
- IP-address: 192.168.1.25
- Netmask: 255.255.255.0
- Std-GW: 192.168.1.254
- 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.

1.5.1.3. Benutzer Manager

The screenshot shows the DEDITEC web interface. 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 two tabs: 'Configuration' and 'In-Outputs'. The 'Configuration' tab is active. Below the tabs, there is a status bar showing 'Product name: RO-ETH-LC (RO-ETH-LC)' and 'User: admin / Session ends in: 196'. On the left side, there is a sidebar with a list of menu items: 'General', 'Network configuration', 'User-Manager', 'Status / Reboot', and 'Security'. The 'User-Manager' item is selected. The main content area is titled 'User-Manager'. It shows 'Modul Status: OK'. Below this, there is a section for 'Webinterface requires login' with a toggle switch set to 'ON'. There is a 'Username:' field with the value 'admin' and a 'Set password' button. Below that, there is a 'Session valid time' field with the value '200' and the unit 'sec'. A notice states: 'Notice: Changes apply after board restart'. At the bottom right of the main content area, 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 minutes. 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 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 seconds and a notice that changes apply after a 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.

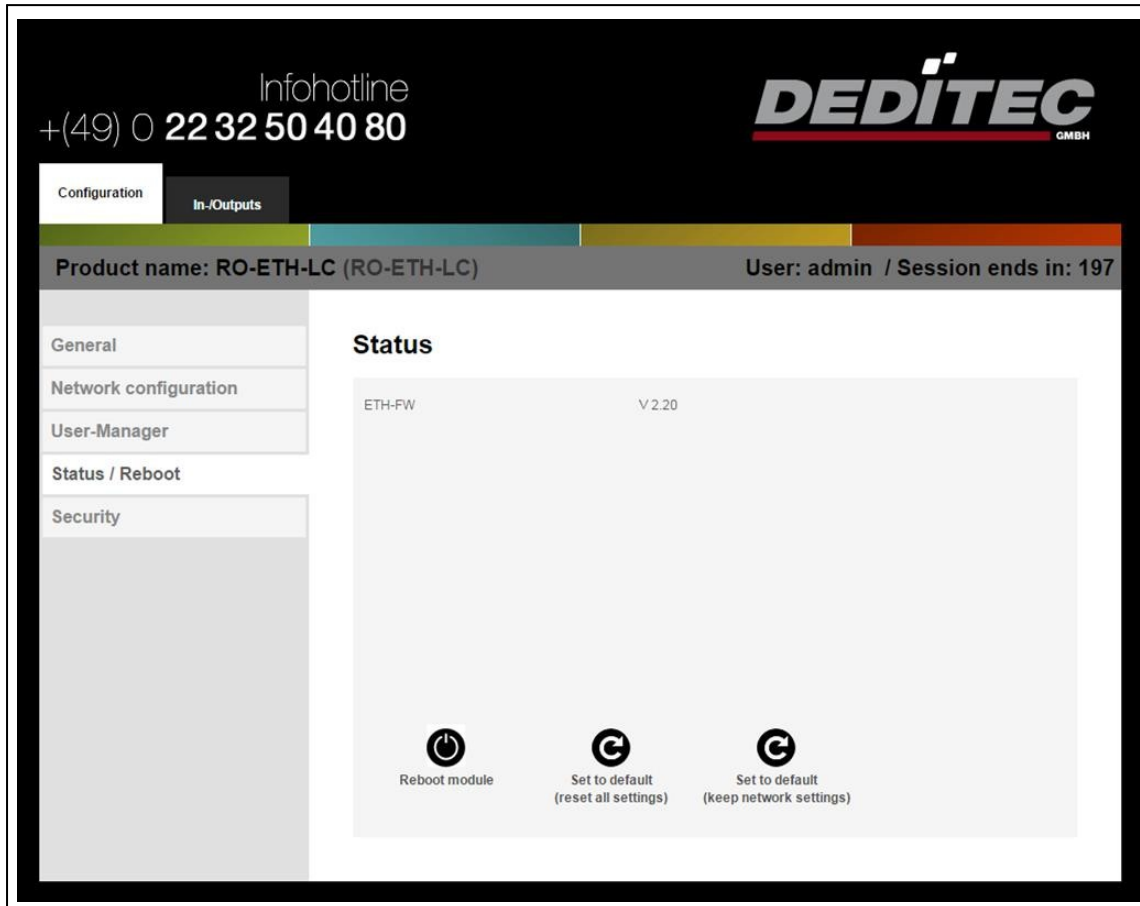
Update

Aktiviert die Zugriffsberechtigungen des aktuellen Benutzerkontos.

Password

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

1.5.1.4. Status



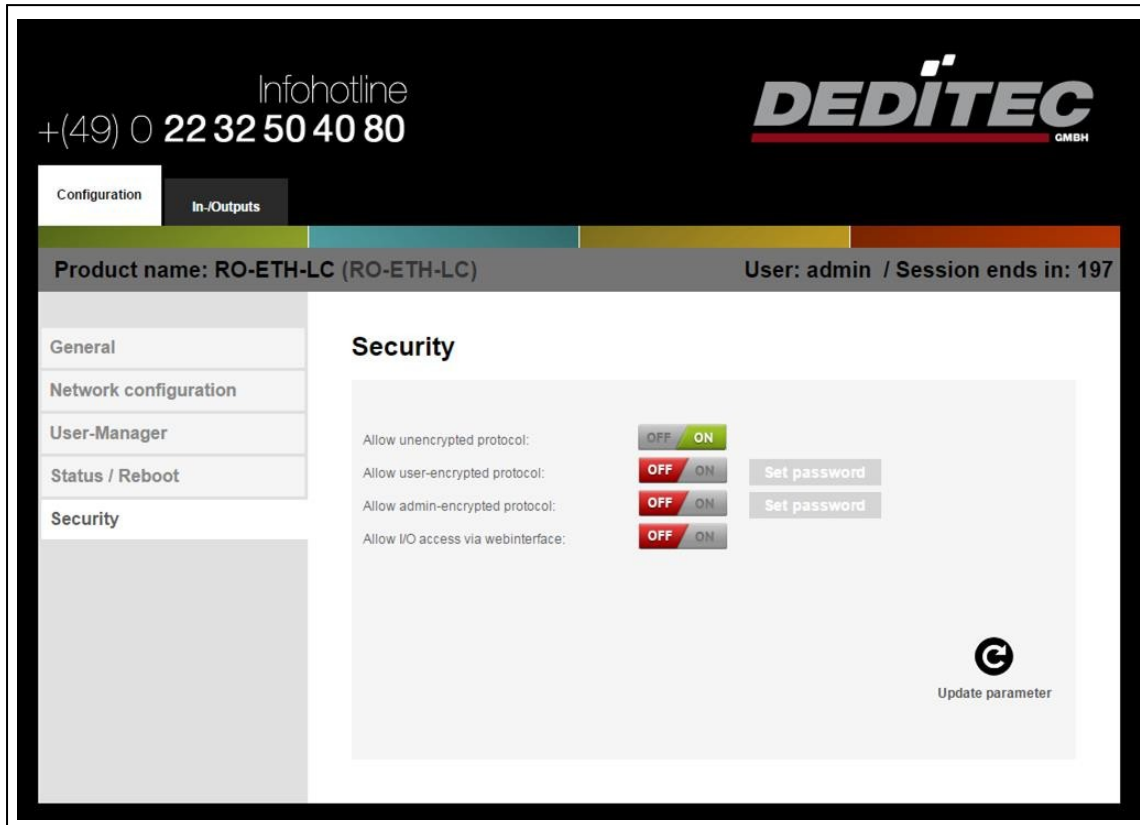
Auf der Status Seite sehen Sie die Revisionsnummern der wichtigsten System Prozesse.

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.

1.5.1.5. 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

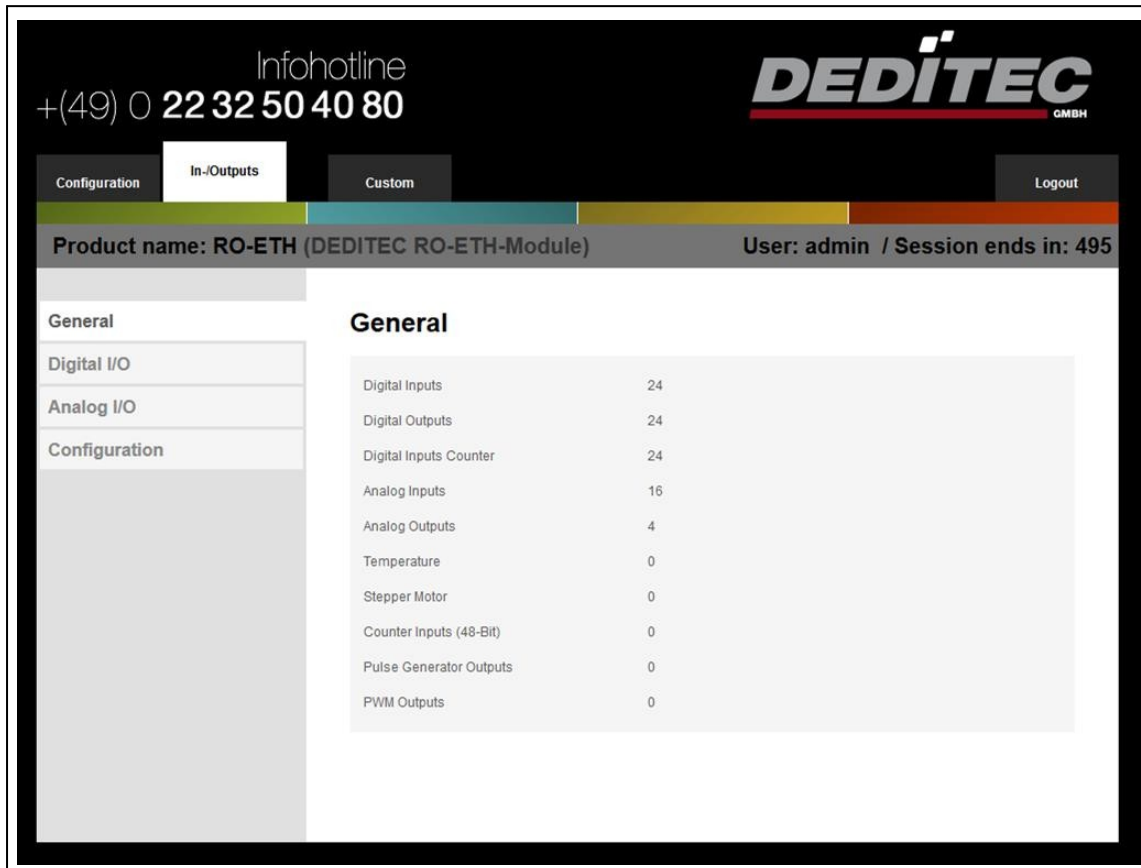
Diese Option bestimmt, ob angeschlossene Ein-/Ausgänge von der Weboberfläche aus gelesen/geschaltet werden können.

Achtung:

Wenn das Passwort zur User- oder Admin-Verschlüsselung geändert wird, muss dieses auch im ICT-Tool nachgetragen werden.

1.5.2. Ein-/Ausgänge

1.5.2.1. 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

1.5.2.2. 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.

1.5.2.3. 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

I/O 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

1.5.2.4. 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.

1.5.2.5. 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. The navigation bar has tabs for 'Configuration', 'In-/Outputs', 'Custom', and 'Logout'. The main content area is titled 'Analog Inputs' and displays a table of 16 channels. The 'Select channel area' dropdown is set to 'CH 0..15' and the 'A/D Mode' is set to '0..10 V'. The table lists channels 0 through 15, each with a name 'Analog Input X' and a value of '0.000 V'. An 'EDIT CH NAMES' button is located at the bottom of the table.

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

1.5.2.6. 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

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

1.5.2.7. 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.



ICT-Tool



2. ICT-Tool

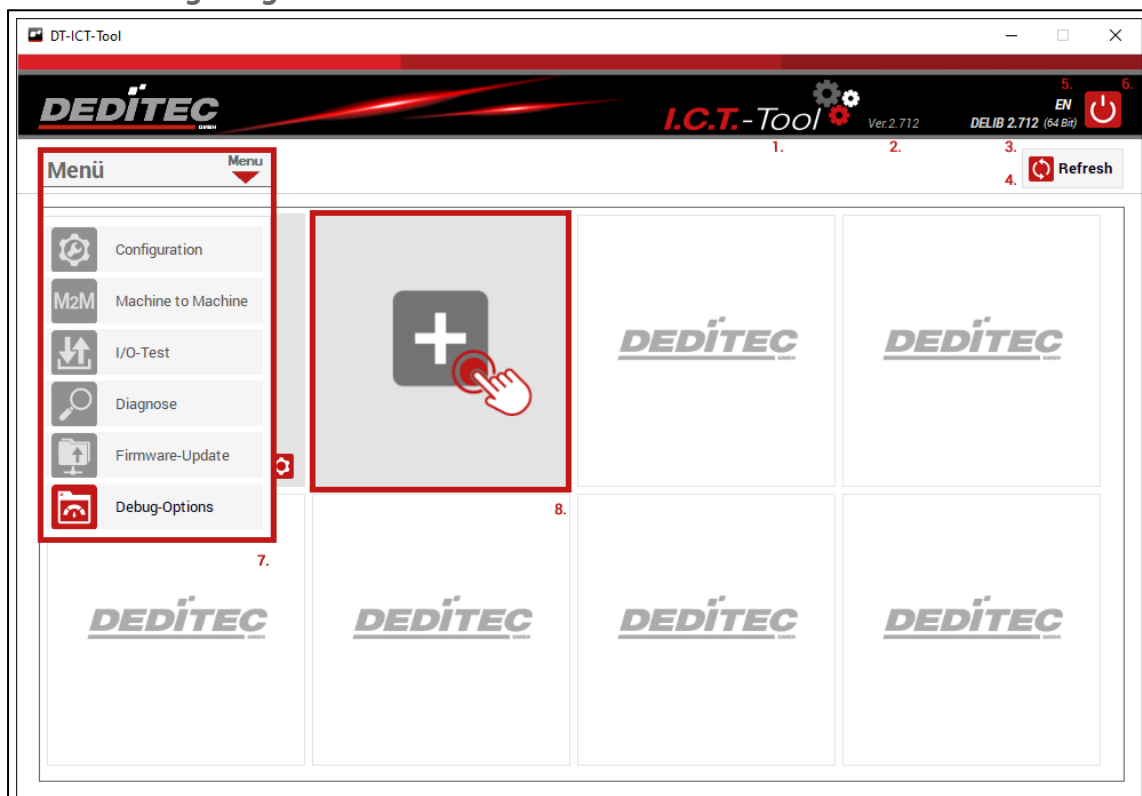
Mit Hilfe des ICT-Tools haben Sie die volle Kontrolle über Ihre DEDITEC-Module. Mit dieser All-in-one Software,

können Sie Ihr Modul flashen, konfigurieren und testen ohne zusätzliche Software. Dies ist besonders wichtig, für die erste Inbetriebnahme des Moduls.

In den folgenden Kapiteln, werden die verschiedenen Bereiche des ICT-Tools genauer dargestellt.

2.1. Allgemeine Beschreibungen

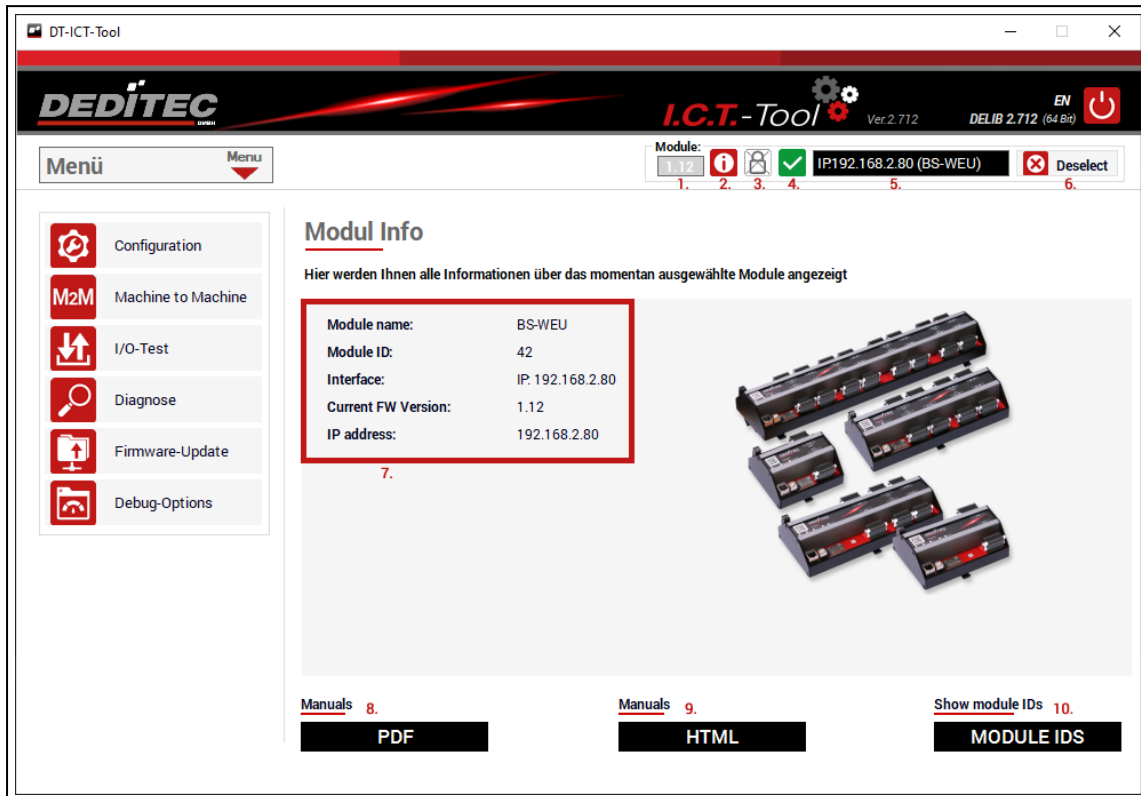
Beschreibung Programmstart:



1. Zeigt den Namen der verwendeten DEDITEC-Software an
2. Zeigt die derzeit genutzte Versionsnummer der Software an
3. Zeigt die derzeit genutzte DELIB-Version an
4. Hier wird die Liste der hinzugefügten Module aktualisiert
5. Durch einen Klick auf das EN oder DE lässt sich die Sprache zwischen englisch und deutsch ändern

6. Schließt das Programm
7. Im Module Selector, können Sie zwischen den einzelnen Bereichen der Software wechseln (es muss erst ein Modul hinzugefügt werden)
8. Mit Hilfe des + Symbols können Sie ein Modul zur Software hinzufügen. Mehr Informationen dazu finden Sie im Kapitel: **Modul hinzufügen**

Beschreibung bei erfolgreicher Modulverbindung



1. Zeigt die aktuelle Firmware-Version des Interface-Moduls an
2. Ruft das Informationsfenster auf (s. unten). Dort finden Sie alle I/O-Spezifischen Informationen
3. Zeigt an, ob das Modul verschlüsselt oder unverschlüsselt ist
4. Zeigt an, ob eine Verbindung zum Modul besteht
5. Zeigt die IP-Adresse des Moduls (via Ethernet) oder die Modulnummer (via USB) an
6. Schließt das Modul und bringt Sie zurück in den Moduleselector
7. Zeigt die aktuelle Netzwerk-Konfiguration, Firmware und Modul-ID Ihres Moduls an
8. Öffnet das Handbuch Ihres Moduls lokal als PDF
9. Öffnet das Handbuch Ihres Moduls im Browser als HTML-Version
10. Zeigt Ihnen alle verfügbaren Modul-IDs an. Diese wird für die

Programmierung von eigenen Projekten benötigt.

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.

2.2. Modul hinzufügen

Um unsere Produkte mit dem ICT-Tool benutzen zu können, müssen diese im Startbildschirm des Programms hinzugefügt werden.

USB-Module

Bei Modulen mit USB-Schnittstelle geschieht das automatisch per Plug and Play. Die Module werden direkt im Startbildschirm angezeigt und lassen sich sofort verwenden.

CAN-Module

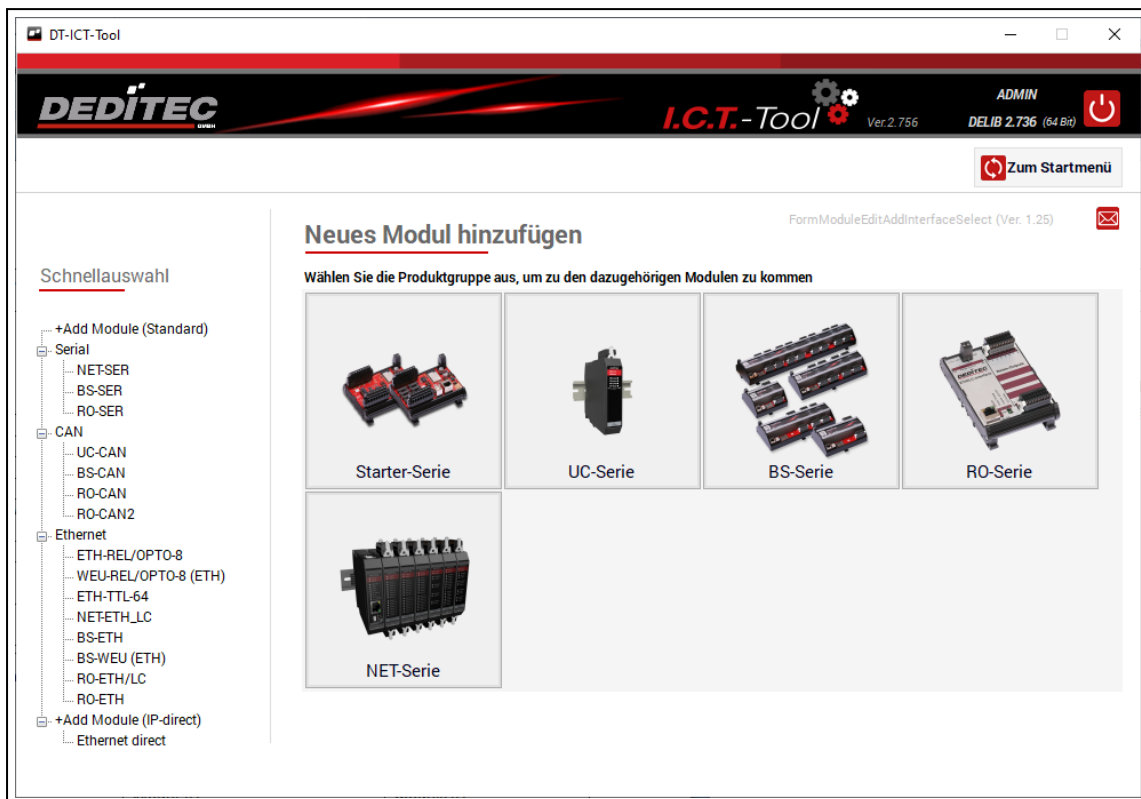
Da CAN-Module ebenfalls über die USB-Schnittstelle konfiguriert werden, sind auch hier keine weiteren Einstellungen notwendig.

Ausgenommen davon sind älteren Module der RO-CAN1-Serie, die wie unsere Ethernet und seriellen Module eingebunden werden.

Ethernet- und Serielle-Module

Mit einem Klick auf das + Symbol können Module mit einer Ethernet- oder Seriellen-Schnittstelle dem ICT-Tool hinzufügen werden.

Im folgenden Fenster haben Sie die Möglichkeit das Modul über die Schnellauswahl am linken Bildschirmrand direkt auszuwählen oder Sie können über die Serien-Vorschaubilder zum gewünschten Produkt navigieren.



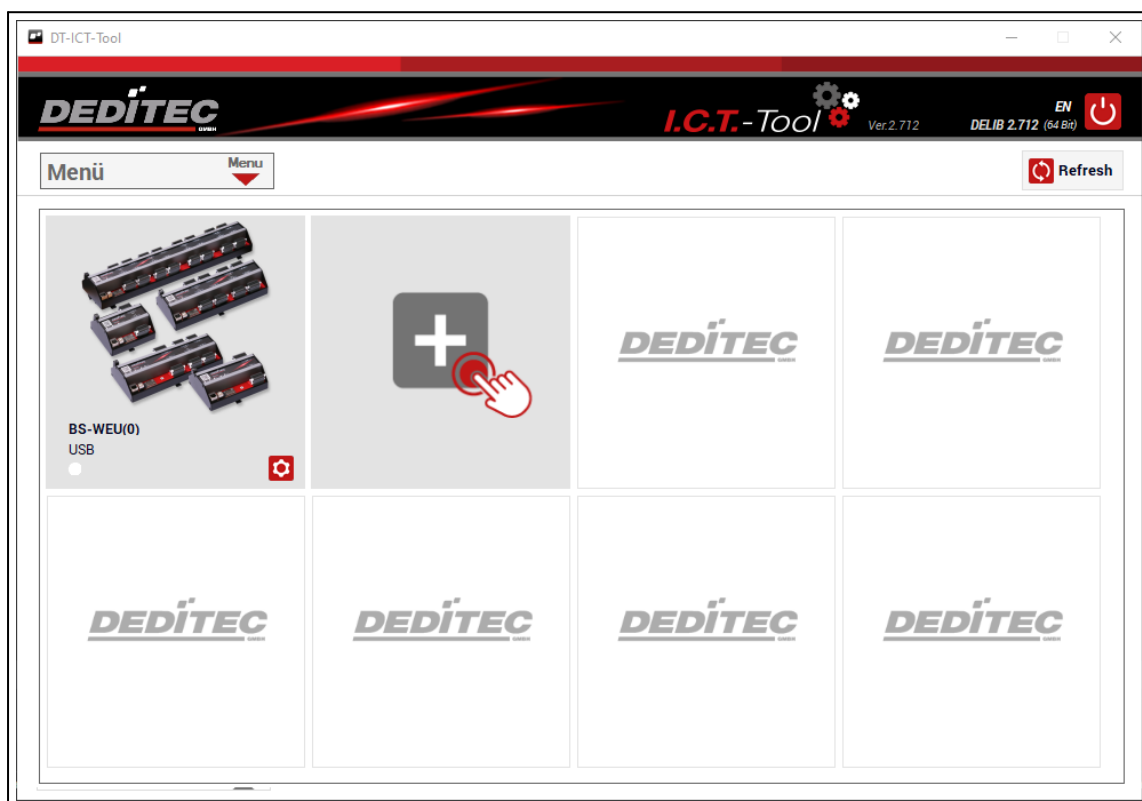
In den folgenden Kapiteln finden Sie nähere Informationen zum Einbinden und Konfigurieren der Schnittstelle Ihres Moduls.

2.2.1. USB Module

Der Anschluss von USB-Modulen funktioniert über Plug and Play. Die Module werden automatisch erkannt und können sofort verwendet werden.

Bei der Verwendung mehrerer USB-Modulen der gleichen Serie, muss zusätzlich die Modulnummer des Moduls geändert werden.

Weitere Informationen dazu finden Sie im Kapitel: **USB-Konfiguration**



2.2.2. Ethernet Module

Um Ethernet Module benutzen zu können, haben Sie die Möglichkeit diese entweder als Registry-Eintrag oder direkt per IP hinzuzufügen. Folgende Tabelle veranschaulicht die jeweiligen Vor- und Nachteile.

	Ethernet (Standard)	Ethernet (IP-Direct)
Generelles	<ul style="list-style-type: none"> • ggf. Administrationsrechte zum Schreiben benötigt werden 	<ul style="list-style-type: none"> • es werden keine Administrationsrechte benötigt
Installation	<ul style="list-style-type: none"> • Installation der benötigten .DLL- und Treiberdateien erfolgt durch das DELIB-Setup 	<ul style="list-style-type: none"> • Keine Installation nötig. Die entsprechende .DLL muss lediglich im Programmverzeichnis liegen
Module konfigurieren	<ul style="list-style-type: none"> • einfache Konfiguration durch das ICT-Tool • Netzwerkeinstellungen, die zum Öffnen des Moduls verwendet werden sollen, werden in die Windows-Registry eingetragen 	<ul style="list-style-type: none"> • Einfache Konfiguration durch das ICT-Tool • Modulkonfiguration am Rechner nicht nötig, da die IP-Adresse direkt in die Anwendungssoftware eingetragen wird
Ansprechen der Module über eigene Software	<ul style="list-style-type: none"> • Einfache Einbindung in Projekte, da ModulID und ModulNR benutzt wird (DapiOpenModule) • Anwendungssoftware muss nicht die IP-Adresse kennen, da diese auf dem Rechner in der Registry gespeichert ist. Die Verteilung von Software auf unterschiedliche Rechner ist dadurch einfacher 	<ul style="list-style-type: none"> • DapiOpenModuleEx öffnet direkt das Modul mit Angabe von IP-Adresse, Timeout und optional Encryption-Type
Anwendungen von Drittanbietern	<ul style="list-style-type: none"> • DapiOpenModule-Befehl benötigt nur 2 Parameter, daher einfacher einzubinden 	<ul style="list-style-type: none"> • gestaltet sich oft schwieriger aus unserer Erfahrung her

2.2.2.1. Ethernet Standard Methode

Hier können Sie Ihr Modul über einen Registry-Eintrag hinzufügen.

Schnellauswahl

Vorhanden

Hier werden bereits hinzugefügte Module aufgelistet. Die Zahl in Klammern entspricht der zugewiesenen Modul-Nummer.

Neues Modul

Hier werden alle noch unbenutzten Modul-Nummern aufgelistet. Die Zahl in Klammern entspricht dabei der zugewiesenen Modul-Nummer.

Konfiguration

Modulstatus

Mit dieser Option können Sie ein Ethernet-Modul dem ICT-Tool hinzufügen. Ist das Häkchen nicht gesetzt, wird das Modul wieder entfernt.

IP/Hostname

Hier können Sie die IP-Adresse des Moduls angeben. Ist das Häkchen bei "use Hostname" , gesetzt, wird statt einer IP-Adresse, der eingetragene Hostname verwendet.

Port

Hier wird die Portnummer des Moduls eingetragen. Der Knopf "Set default port" schreibt den Standard Port "9912" in das Feld.

Timeout [msec]

Der Timeout gibt an, wie lange versucht wird, eine Verbindung zum Modul herzustellen. Ist diese Zeit zu gering eingestellt, kann unter Umständen der Timeout getriggert werden, bevor das Modul ordnungsgemäß geöffnet wurde. Die Dauer der Verbindungsversuche, nach einem Verbindungsabbruch zum Modul, wird ebenfalls mit dem Timeout angegeben. Der Knopf "set default timeout" schreibt den Standard Timeout-Wert "5000" in das Feld.

Encryption type

Mit dem Encryption type können Sie einstellen, ob die Modulkommunikation unverschlüsselt oder verschlüsselt (User/Admin) geöffnet werden soll.

Mehr Informationen dazu finden Sie im Kapitel: **TCP-Verschlüsselung**

Modul identifizieren

Mit der Identifikations Funktion können Sie feststellen, welches Ihrer Module momentan durch das ICT-Tool angesteuert wird.

Dies ist besonders hilfreich, bei der gleichzeitigen Verwendung mehrerer Module.

Durch Betätigen der "Identifizieren" Schaltfläche fängt eine Kontroll LED auf dem Modul an, schnell zu blinken.

Die Kontroll-LED variiert je nach Modul-Serie:

- NET-Serie: "Int.Act"-LED
- RO-Serie: "Int.Act"-LED
- BS-WEU-Serie: "Status"-LED
- Starter-Serie: "Status"-LED

Suche nach DEDITEC Modulen im Netzwerk

Über die Search-Funktion wird das vorhandene Netzwerk nach angeschlossenen DEDITEC Modulen gescannt.

Die Ergebnisse des Suchlaufs werden anschließend in einer Listenansicht dargestellt. Mehr dazu im Kapitel: **Search via Broadcast**

SPEICHERN/TEST

Hier können Sie geänderte Einstellungen speichern und die Kommunikation mit dem Modul testen.

(es werden ggf. Administrationsrechte benötigt)

BEENDEN

Hier gelangen Sie zurück ins Startmenü.

2.2.2.2. Ethernet Direct Methode

Hier können Sie Ihr Modul direkt über die IP-Adresse hinzufügen.

The screenshot shows the 'Ethernet Konfiguration (IP-direct)' window in the DEDITEC I.C.T.-Tool. The window has a title bar 'DT-ICT-Tool' and a header with the DEDITEC logo, 'I.C.T.-Tool Ver. 2.756', 'ADMIN', and 'DELIB 2.738 (64 Bit)'. A 'Zum Startmenü' button is in the top right. On the left is a 'Schnellwahl' (Quick Selection) tree with categories: '+Add Module (Standard)', 'Serial' (NET-SER, BS-SER, RO-SER), 'CAN' (UC-CAN, BS-CAN, RO-CAN, RO-CAN2), 'Ethernet' (ETH-REL/OPTO-8, WEU-REL/OPTO-8 (ETH), ETH-TTL-64, NET-ETH_LC, BS-ETH, BS-WEU (ETH), RO-ETH/LC, RO-ETH), '+Add Module (IP-direct)', and 'Ethernet direct'. The main area is titled 'Ethernet Konfiguration (IP-direct)' with a subtitle 'Ethernet Einstellungen für Zugriff auf das Modul ändern.' Below this are input fields: 'IP / Hostname' (192.168.1.1), 'Port' (9912), 'Timeout [msec]' (5000), and 'Verschlüsselungsmodus' (Deaktiviert). There are buttons for 'Hostname verwenden' (unchecked), 'Standard Port setzen', 'Standard Timeout setzen', and 'Modul identifizieren'. At the bottom is a button 'Suche nach DEDITEC Modulen im Netzwerk'. At the very bottom are three buttons: 'ZURÜCK', 'SPEICHERN/TEST', and 'BEENDEN'.

Konfiguration

IP/Hostname

Hier können Sie die IP-Adresse des Moduls angeben. Ist das Häkchen bei "use Hostname" , gesetzt, wird statt einer IP-Adresse, der eingetragene Hostname verwendet.

Port

Hier wird die Portnummer des Moduls eingetragen. Der Knopf "Set default port" schreibt den Standard Port "9912" in das Feld.

Timeout [msec]

Der Timeout gibt an, wie lange versucht wird, eine Verbindung zum Modul herzustellen. Ist diese Zeit zu gering eingestellt, kann unter Umständen der Timeout getriggert werden, bevor das Modul ordnungsgemäß geöffnet wurde. Die Dauer der Verbindungsversuche, nach einem Verbindungsabbruch zum Modul, wird ebenfalls mit dem Timeout angegeben. Der Knopf "set default timeout" schreibt den Standard Timeout-Wert "5000" in das Feld.

Encryption type

Mit dem Encryption type können Sie einstellen, ob die Modulkommunikation unverschlüsselt oder verschlüsselt (User/Admin) geöffnet werden soll.

Mehr Informationen dazu finden Sie im Kapitel: **TCP-Verschlüsselung**

Modul identifizieren

Mit der Identifikations Funktion können Sie feststellen, welches Ihrer Module momentan durch das ICT-Tool angesteuert wird.

Dies ist besonders hilfreich, bei der gleichzeitigen Verwendung mehrerer Module.

Durch Betätigen der "Identifizieren" Schaltfläche fängt eine Kontroll LED auf dem Modul an, schnell zu blinken.

Die Kontroll-LED variiert je nach Modul-Serie:

- NET-Serie: "Int.Act"-LED
- RO-Serie: "Int.Act"-LED
- BS-WEU-Serie: "Status"-LED
- Starter-Serie: "Status"-LED.

Suche nach DEDITEC Modulen im Netzwerk

Über die Search-Funktion wird das vorhandene Netzwerk nach angeschlossenen DEDITEC Modulen gescannt.

Die Ergebnisse des Suchlaufs werden anschließend in einer Listenansicht dargestellt. Mehr dazu im Kapitel: **Search via Broadcast**

SPEICHERN/TEST

Hier können Sie geänderte Einstellungen speichern und die Kommunikation mit dem Modul testen.

(es werden ggf. Administrationsrechte benötigt)

BEENDEN

Hier gelangen Sie zurück ins Startmenü.

2.2.2.3. Search via Broadcast

Über die Search-Funktion wird das vorhandene Netzwerk nach angeschlossenen DEDITEC Modulen gescannt.

Die Ergebnisse des Suchlaufs werden anschließend in einer Listenansicht dargestellt.

The screenshot shows the DEDITEC I.C.T.-Tool interface. The top bar includes the DEDITEC logo, 'I.C.T.-Tool Ver. 2.756', and user information 'ADMIN DELIB 2.738 (64 Bit)'. A 'Zum Startmenü' button is in the top right. The left sidebar has a 'Schnellauswahl' section with a tree view of modules: Serial (NET-SER, BS-SER, RO-SER), CAN (UC-CAN, BS-CAN, RO-CAN, RO-CAN2), Ethernet (ETH-REL/OPTO-8, WEU-REL/OPTO-8 (ETH), ETH-TTL-64, NET-ETH_LC, BS-ETH, BS-WEU (ETH), RO-ETH/LC, RO-ETH), and Add Module (IP-direct) (Ethernet direct). The main area is titled 'Modulsuche' and contains the text 'Hier können Sie nach verfügbaren Module suchen.' Below this is a table of detected modules:

MAC	DHCP	IP	ProductName	ProductType	WP	Select	Ident	Edit
40:91:51:2F24:93	True	192.168.2.126	BS-WEU	BS-WEU(42)	False	Select	Ident	Edit
40:91:51:2F86:57	True	192.168.2.131	BS-WEU	BS-WEU(42)	False	Select	Ident	Edit
00:C0:D5:FFFF:FF	True	192.168.2.159	NET-ETH-LC	NET-ETH_LC(32)	False	Select	Ident	Edit
00:C0:D5:02:0C:....	True	192.168.2.168	NET-ETH-LC	NET-ETH_LC(32)	False	Select	Ident	Edit

Below the table is a button 'Aktualisiere Liste mit DEDITEC Modulen im Netzwerk'. At the bottom are 'ZURÜCK' and 'BEENDEN' buttons.

Select

Mit Hilfe der Auswahl wird das entsprechende Modul in das Konfigurations Menü übernommen.

Ident

Mit Hilfe der Identify-Funktion kann das angesprochene Modul durch eine schnell blinkende Status LED identifiziert werden.

Dies ist besonders hilfreich, bei der Verwendung von mehreren Modulen.

Edit

Hier können Sie direkt die Netzwerkeinstellungen des ausgewählten Moduls ändern und in das Modul speichern.

Aktualisiere Liste mit DEDITEC Modulen im Netzwerk

Die Liste der gefundenen DEDITEC Module im Netzwerk wird aktualisiert.

2.2.3. CAN Module

CAN-Module werden zur Konfiguration über die USB-Schnittstelle mit dem PC verbunden.

Einstellungen am CAN-Interface und den TX- und RX-Paketen können im Konfigurationsbereich vorgenommen werden.

Weitere Informationen dazu finden Sie im Kapitel: **CAN-Konfiguration**

2.2.4. SER Module

Hier können Sie den COM-Port Ihres Moduls einer Modulnummer zuweisen. Konfigurationen der seriellen Schnittstelle können im Konfigurationsbereich vorgenommen werden. Dort lässt sich auch einstellen, ob Sie das Modul über die serielle Schnittstelle mit dem ICT-Tool benutzen wollen.

Weitere Informationen dazu finden Sie im Kapitel: **Serielle Konfiguration**

The screenshot shows the DT-ICT-Tool interface. At the top, there's a header with the DEDITEC logo, 'I.C.T.-Tool' branding, version 'Ver. 2.756', and user information 'ADMIN' and 'DELIB 2.738 (64 Bit)'. A 'Zum Startmenü' button is in the top right. The main content area is titled 'Zuweisung der COM-Ports' with a subtitle 'Wählen Sie in der linken Spalte die gewünschte Modulnummer aus.' On the left, under 'Schnellauswahl', there's a tree view with '+Ergänze Modul' and 'Neues Modul' containing a list of modules from 'BS-SER(0)' to 'BS-SER(15)'. The main configuration area on the right shows 'Modulname' as 'BS-SER (0)', 'Modulstatus' with a checked 'Aktivieren' checkbox, 'COM-Port' set to 'COM10', and 'Kommunikations-Wiederholversuche im Fehlerfall' set to '5'. At the bottom, there are three buttons: 'ZURÜCK', 'SPEICHERN/TEST', and 'BEENDEN'.

Schnellauswahl

Vorhanden

Hier werden bereits hinzugefügt Module aufgelistet. Die Zahl in Klammern entspricht der zugewiesenen Modulnummer

Neues Modul

Hier werden alle noch unbenutzten Modulnummern aufgelistet. Die Zahl in Klammern entspricht dabei der zugewiesenen Modulnummer.

Konfiguration

Modulname

Zeigt das ausgewählte Modul an.

Modulstatus

Mit dieser Option können Sie ein serielles Modul dem ICT-Tool hinzufügen. Ist das Häkchen nicht gesetzt, wird das Modul wieder entfernt.

COM-Port

Hier können Sie den COM-Port Ihres Moduls einstellen.

Kommunikationsversuche im Fehlerfall

Hier können Sie einstellen, wie viele Zugriffsversuche im Fehlerfall durchgeführt werden sollen.

SPEICHERN/TEST

Hier können Sie geänderte Einstellungen speichern und die Kommunikation mit dem Modul testen.

(es werden ggf. Administrationsrechte benötigt)

BEENDEN

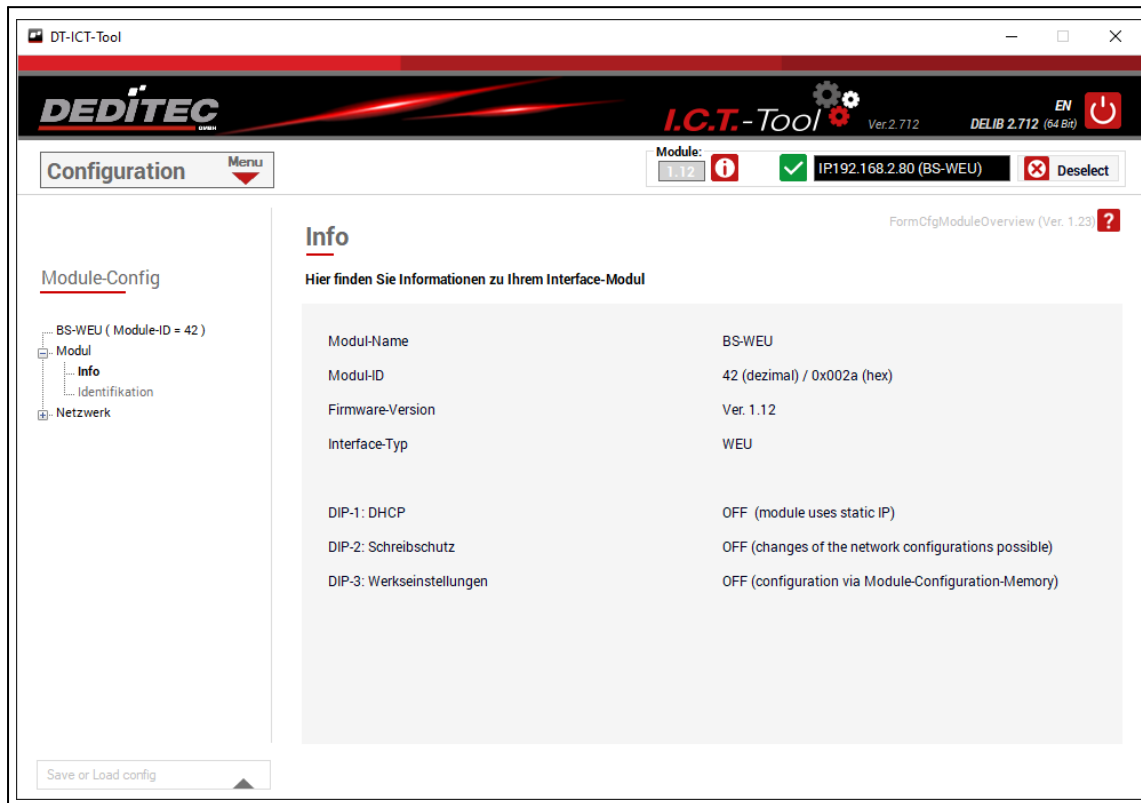
Hier gelangen Sie zurück ins Startmenü.

2.3. Modul konfigurieren

Im Bereich Konfiguration können Einstellungen des Moduls eingesehen oder geändert werden.

2.3.1. Modul Infoseite

Mit dem ICT-Tool lässt sich Ihr Modul nicht nur schnell und einfach konfigurieren, Sie können sich auch viele wichtige Modulinformationen auf nur einen Blick anzeigen lassen.



Modul Name

Zeigt den Namen des aktuell verwendeten DEDITEC-Moduls an.

Modul-ID

Zeigt die ID Ihres verwendeten Moduls an. Diese wird unter Umständen für das Programmieren eigener Software mit DELIB-Befehlen benötigt.

Firmware-Version

Zeigt die aktuelle auf dem Modul installierte Firmware Version an.

DIP-Schalterstellungen

Zeigt die aktuelle DIP-Schalterstellung Ihrer ETH-Schnittstelle an.

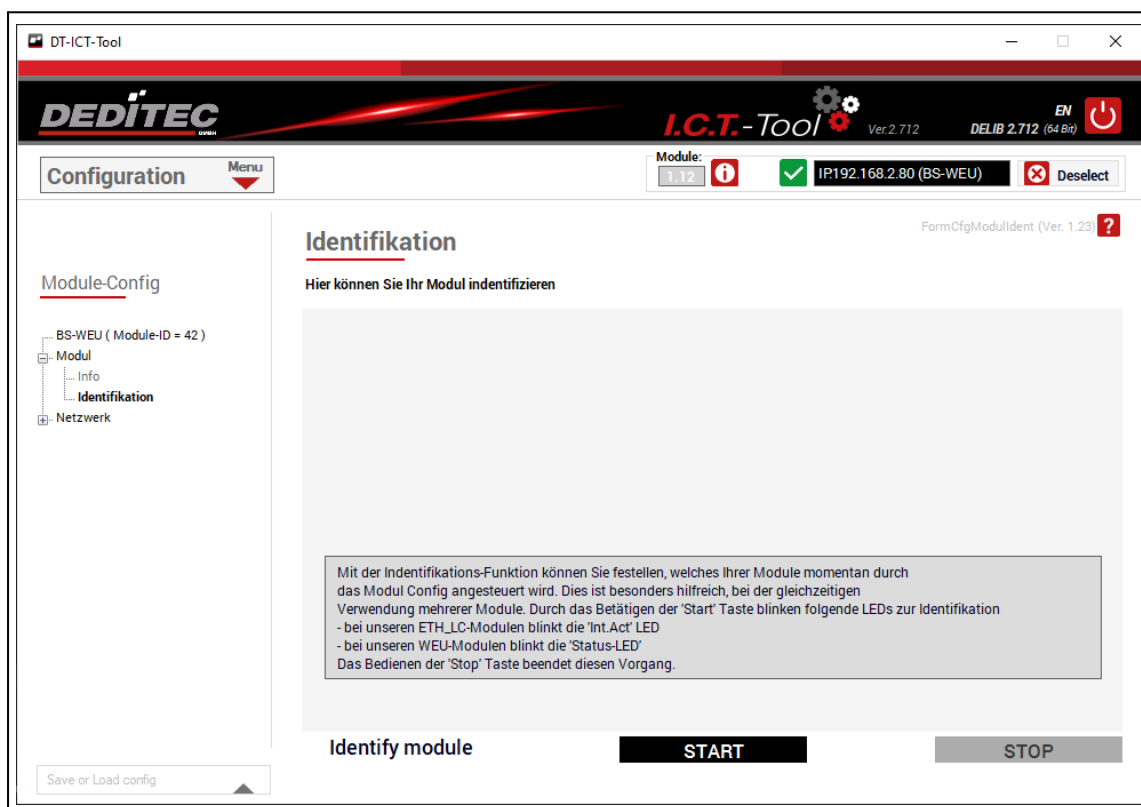
2.3.2. Modul Identifikation

Mit der Identifikations Funktion können Sie feststellen, welches Ihrer Module momentan durch das ICT-Tool angesteuert wird.

Dies ist besonders hilfreich, bei der gleichzeitigen Verwendung mehrerer Module. Durch Betätigen der "Identifizieren" Schaltfläche fängt eine Kontroll LED auf dem Modul an, schnell zu blinken.

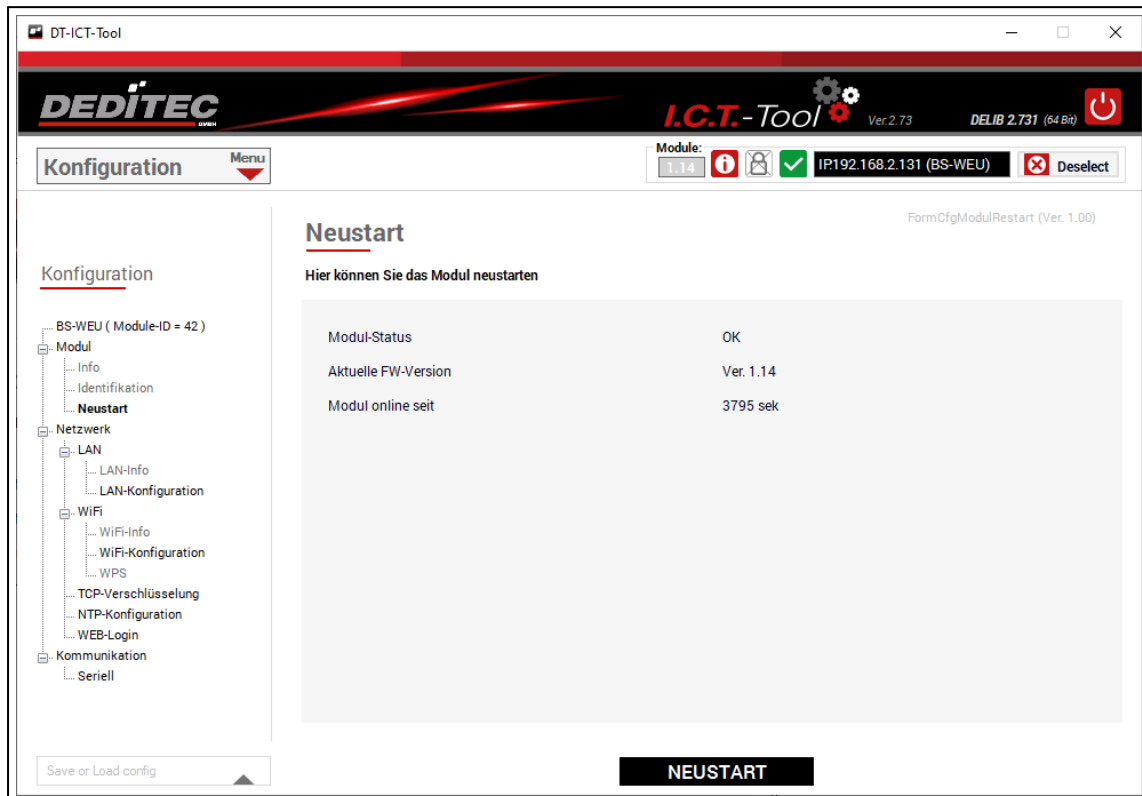
Die Kontroll-LED variiert je nach Modul-Serie:

- NET-Serie: "Int.Act"-LED
- RO-Serie: "Int.Act"-LED
- BS-WEU-Serie: "Status"-LED
- Starter-Serie: "Status"-LED



2.3.3. Modul Neustart

Hier können Sie Ihr Modul neustarten.



Modul Status

Zeigt an, ob nach dem Neustart die Verbindung zu dem Modul wieder erfolgreich aufgebaut wurde.

- OK: Verbindung wurde erfolgreich aufgebaut
- Fehler: Modul nach dem Neustart nicht gefunden

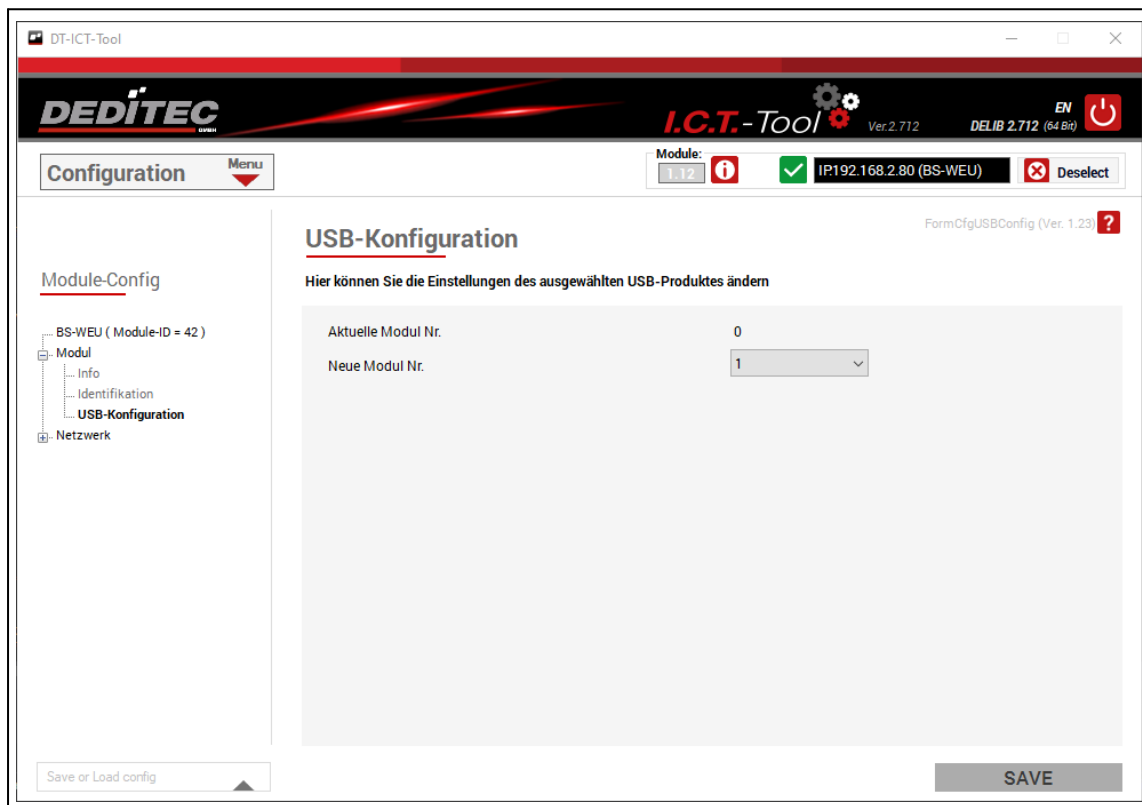
2.3.4. USB Konfiguration

Hier können Sie die Modulnummer Ihres USB-Moduls ändern.

Diese ist notwendig, wenn Sie mehrere Module des gleichen Typs in Betrieb nehmen möchten.

Um das richtige Modul öffnen zu können, wird neben der Modul-ID auch die entsprechende Modulnummer benötigt.

Die aktuell verwendete Modulnummer wird neben dem Namen des Moduls in Klammern dargestellt: BS-WEU(0)



Aktuelle Modulnummer

Zeigt an, welche Modulnummer aktuell zugewiesen ist.

Neue Modulnummer

Hier können Sie dem Modul eine neue Modulnummer zuweisen.

2.3.5. LAN Netzwerkinformationen

Hier finden Sie alle wichtigen LAN-Netzwerkinformationen für Ihr Modul auf einen Blick.

The screenshot shows the 'I.C.T.-Tool' interface for configuring a DEDITEC module. The main window is titled 'DT-ICT-Tool'. The top bar includes the DEDITEC logo, the tool name 'I.C.T.-Tool', version 'Ver. 2.712', and a power button icon. Below the bar, there's a 'Configuration' tab and a 'Menu' button. A status bar shows 'Module: 1.12' with a green checkmark and 'IP192.168.2.80 (BS-WEU)' with a red 'X' and 'Deselect' button. The left sidebar shows a tree view with 'Module-Config' expanded, containing 'Modul', 'Netzwerk', 'LAN', 'LAN-Info', 'LAN-Konfiguration', 'WiFi', 'TCP-Verschlüsselung', 'NTP-Konfiguration', and 'WEB-Login'. The main area is titled 'LAN-Info' with a subtitle 'Hier finden Sie die Netzwerkinformationen des ausgewählten Ethernet-Produktes'. It contains a form with the following fields: 'MAC address' (58:BF:25:E8:D3:B7), 'Board name (Hostname in DHCP mode)' (BS-WEU), 'LAN status' (Static-IP success), 'DHCP active' (checkbox), 'IP address' (192.168.2.80), 'Subnet mask' (255.255.255.0), 'Default gateway' (192.168.2.254), and 'TCP port' (9912). At the bottom left, there's a 'Save or Load config' button.

MAC-Adresse

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

Boardname

Zeigt den aktuellen Boardname Ihres Moduls an.

LAN-Status

Hier wird der Verbindungsstatus Ihres angeschlossenen Moduls angezeigt.

Wenn der Status „Query not supported (FW update)“ angezeigt wird, benötigt Ihr Modul eine aktuellere Firmware.

DHCP aktiv

Zeigt an, ob DHCP per DIP-Schalter oder Software aktiviert ist.

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

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

2.3.6. LAN Netzwerkeinstellungen

Hier können Sie die LAN-Netzwerkeinstellungen Ihres Moduls konfigurieren.

The screenshot shows the 'DT-ICT-Tool' window with the 'DEDITEC' logo and 'I.C.T.-Tool' branding. The 'Configuration' menu is active, and the 'Module' dropdown is set to '1.12'. The 'IP192.168.2.80 (BS-WEU)' is selected, and the 'Deselect' button is visible. The 'LAN-Konfiguration' section is titled 'Hier können Sie die Netzwerkeinstellungen des ausgewählten Ethernet-Produktes ändern'. It contains the following fields and options:

Field/Option	Value
Board name (Hostname in DHCP mode)	BS-WEU
DHCP active	<input type="checkbox"/>
IP address	192.168.2.80
Subnet mask	255.255.255.0
Default gateway	192.168.2.254
TCP port	9912

At the bottom, there are buttons for 'Save or Load config', 'LOAD DEFAULT', and 'SAVE'.

Boardname

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

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

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

Mehr Infos zum Anbinden des Moduls via Boardname finden Sie im Kapitel: **Benutzung des Moduleselectors**

DHCP aktiv

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

Der Boardname wird als Hostname verwendet. Sollte DHCP via DIP-Schalter aktiviert sein, wird diese Einstellung ignoriert.

IP-Adresse, Subnetz-Maske, Std. Gateway und TCP-Port

Diese Einstellungen werden verwendet, wenn DHCP deaktiviert ist.

Werkseinstellungen laden

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

Werkseinstellungen	
Boardname	Modul abhängig
DHCP	off
IP-Adresse	192.168.1.1
Subnet mask	255.255.255.0
Default gateway	192.168.1.254
TCP Port	9912

2.3.7. WiFi Netzwerkinformationen

Hier finden Sie alle wichtigen WiFi-Netzwerkinformationen Ihres Moduls auf einen Blick.

The screenshot shows the 'WiFi-Info' configuration page in the DEDITEC I.C.T.-Tool. The interface includes a top header with the DEDITEC logo, 'I.C.T.-Tool' branding, and version information (Ver. 2.712, DELIB 2.712 (64 Bit)). A 'Configuration' menu is visible on the left. The main content area is titled 'WiFi-Info' and contains a list of network parameters with their current values:

Parameter	Value
MAC address	00:00:00:00:00:00
Board name (Hostname in DHCP mode)	BS-WEU_W
WLAN status	WiFi Disabled
WLAN active	<input type="checkbox"/>
IP address	0.0.0.0
Subnet mask	0.0.0.0
Default gateway	0.0.0.0
TCP port	9912
Router name	DefaultSSID
Password	DefaultPWD

At the bottom of the page, there is a 'Save or Load config' button and an 'Auto Refresh' checkbox.

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 aktiv

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.

Router-Password

Zeigt das verwendete Routerpasswort an.

2.3.8. WiFi Netzwerkeinstellungen

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

The screenshot shows the 'DT-ICT-Tool' window with the 'DEDITEC' logo and 'I.C.T.-Tool' branding. The 'Configuration' menu is active, and the 'WiFi-Konfiguration' page is displayed. The left sidebar shows a tree view with 'Module-Config' expanded, leading to 'WiFi-Konfiguration'. The main area contains the following settings:

- Board name (Hostname in DHCP mode): BS-WEU_W
- WLAN active: ☐
- Router name: DefaultSSID
- Password: DefaultPWD
- TCP port: 9912

At the bottom, there are buttons for 'Save or Load config', 'LOAD DEFAULT', and 'SAVE'. The status bar at the top right shows 'Module: 1.12', a green checkmark, and the IP address 'IP192.168.2.171'.

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 aktiv

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

Router-Name

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

Router-Passwort

Hier können Sie das Router-Passwort des verwendeten Routers eintragen.

TCP-Port

Hier wird der verwendete TCP-Port angezeigt. 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	DefaultSSID
Password	DefaultPWD

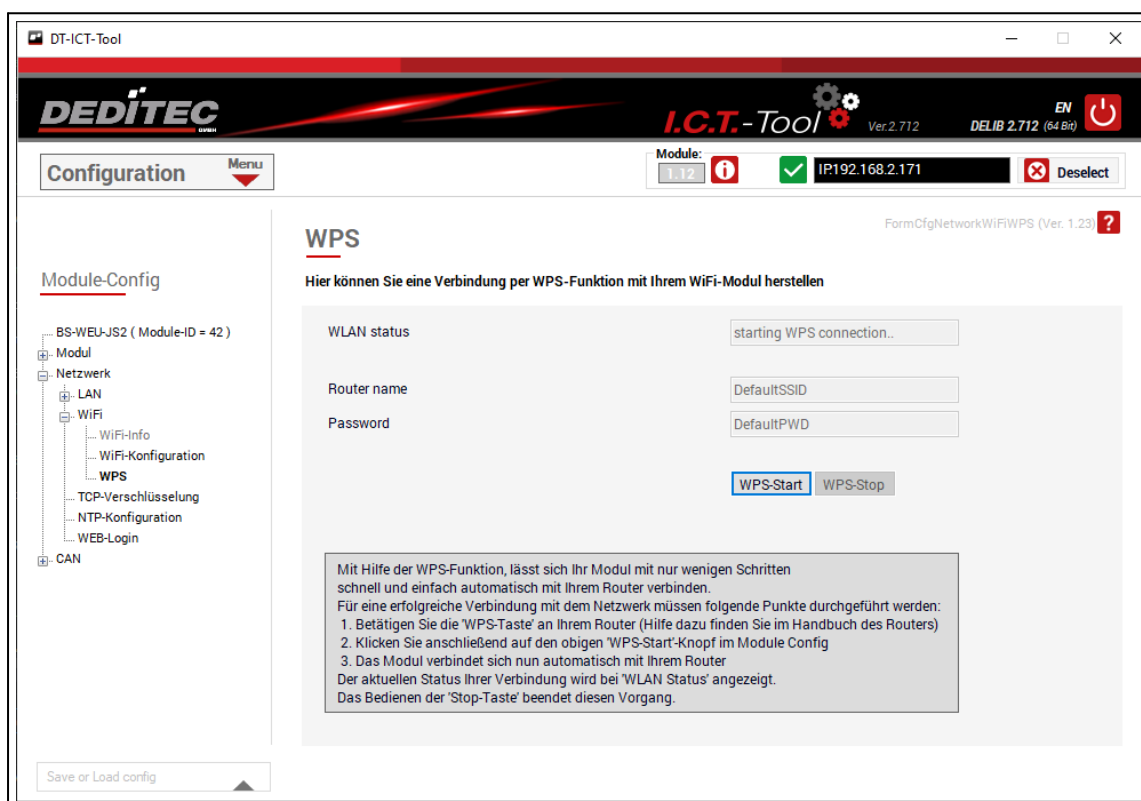
2.3.9. WiFi WPS-Verbindung

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

Drücken Sie dafür auf den "WPS-Start" Knopf im ICT-Tool (s.Bild unten) **oder** drücken und halten Sie den Config-Taster auf Ihrem Modul für 7 Sekunden gedrückt.

Der WPS-Status sollte jetzt auf "starting WPS connection" wechseln. Starten Sie nun den WPS-Suchlauf auf Ihrem Router.

Bei einer erfolgreichen Verbindung wechselt der WLAN-Status auf "DHCP is ON". Zusätzlich erscheint der verwendete Router-Name und das Passwort.



2.3.10. NTP-Konfiguration

Hier können Änderungen am NTP-Service vorgenommen werden.

The screenshot shows the 'NTP-Konfiguration' window within the DEDITEC I.C.T.-Tool. The interface includes a top header with the DEDITEC logo, 'I.C.T.-Tool Ver. 2.712', and 'DELIB 2.712 (64 Bit)'. A 'Configuration' menu is visible. On the left, a 'Module-Config' tree shows the navigation path: BS-WEU-JS2 (Module-ID = 42) > Modul > Netzwerk > LAN > WiFi > TCP-Verschlüsselung > NTP-Konfiguration. The main area is titled 'NTP-Konfiguration' with a subtitle 'Hier können Sie Änderungen am NTP Service vornehmen'. It contains the following settings:

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

At the bottom, there are buttons for 'LOAD DEFAULT' and 'SAVE', and a 'Save or Load config' dropdown.

NTP-Service aktiv

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.

Zeitzone

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 aktiv	on
Server	0.de.pool.ntp.org
Port	123
Zeitzone	(GMT) Greenwich Mean Time: Dublin, Edinburgh, Lisbon, London

2.3.11. TCP-Verschlüsselung

1. Kurzbeschreibung

Mit dem DEDITEC-Verschlüsselungsmodus kann die gesamte TCP-Kommunikation vom Steuer-PC zum DEDITEC-Modul verschlüsselt werden. Somit wird ein nicht autorisierter Zugriff auf das Modul durch Fremdpersonen verhindert.

Die Verschlüsselung selbst, erfolgt mittels Passwortvergabe im ICT-Tool und findet dann automatisch in der DELIB-Treiberbibliothek statt. Auf dem DEDITEC-Modul werden die Daten anschließend wieder selbstständig entschlüsselt.

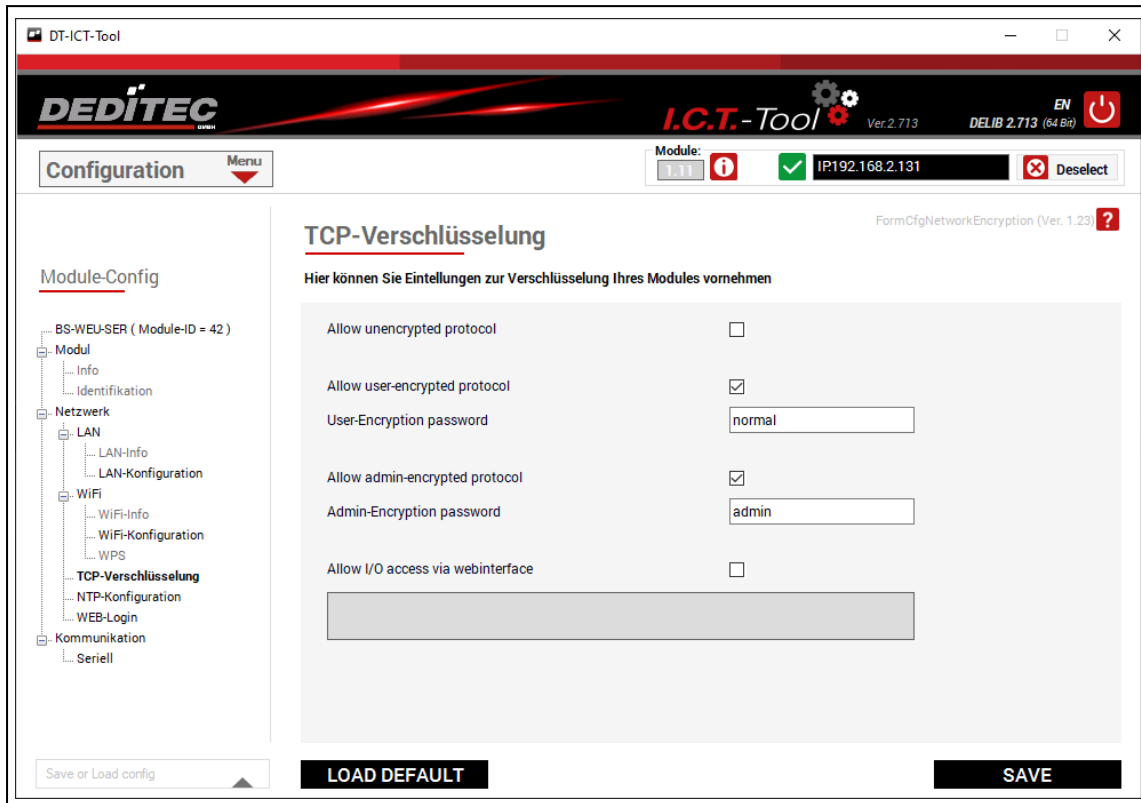
2. Modul konfigurieren

Das Modul kann mit 3 unterschiedlichen Zugriffsrechten versehen werden:

Verschlüsselungstyp	Bedeutung
Unverschlüsseltes Protokoll	offener Zugriff auf alle Funktionen
Benutzer verschlüsseltes Protokoll	geschützter Zugriff nur auf die I/Os
Admin verschlüsseltes Protokoll	geschützter Zugriff auf alle Funktionen

- Wählen Sie zunächst das Modul im ICT-Tool via USB oder Ethernet-Anbindung aus
- Wechseln Sie in das Menü Konfiguration -> Netzwerk -> TCP Verschlüsselung
- Wählen Sie nun die gewünschten Verschlüsselungsoptionen aus und vergeben Sie bei Bedarf die entsprechenden Passwörter für den User- und Admin-Zugriff***
- Speichern Sie die eingegebenen Daten auf das Modul

*** **Hinweis:** Ist ausschließlich die Option „Benutzerverschlüsseltes Protokoll“ aktiviert, können via Ethernetanbindung keine Modulkonfigurationen mehr vorgenommen werden.



Geschieht die Konfiguration via Ethernetanbindung, stehen zwei Möglichkeiten der Datenübertragung zur Auswahl:

Übertragungsart	Bedeutung
Normal	Unverschlüsselte Übertragung des Passwortes
Temp-Admin	Verschlüsselte Übertragung des Passwortes

Mit Hilfe des temporären Admin-Modus, wird das gesetzte Passwort verschlüsselt in das Modul übertragen. **Gleichzeitig muss am Modul ein Freigabe-Taster oder DIP-Schalter betätigt werden.** Dies soll sicherstellen, dass keine unbefugte Person von außen, die Verschlüsselungseinstellungen am Modul verändert.

3. Einrichten des Moduls am Arbeitsplatz

Nachdem die Verschlüsselungsoptionen ins Modul gespeichert wurden, gehen Sie zur Modulauswahl in das Hauptmenü des ICT-Tools zurück. Klicken Sie nun auf das kleine Zahnradsymbol des verschlüsselten Moduls.

Im Feld „Encryption type“ wählen Sie nun die Zugriffsrechte aus, mit der das Modul zukünftig angesprochen werden soll. Danach tragen Sie dort das entsprechende User- oder Admin-Passwort ein, welches zuvor in das Modul gespeichert wurde.

Nach dem Speichern der Einstellungen startet das Modul mit den festgelegten Zugriffsrechten.

The screenshot shows the DT-ICT-Tool interface. The top bar includes the DEDITEC logo, the I.C.T.-Tool title, version 2.713, and a language selector set to EN. Below the bar, a status bar indicates 'no module selected' with a 'Deselect' button. The main window is titled 'IP Adresse bearbeiten' (Edit IP Address) and contains the following elements:

- Left Sidebar (Schnellauswahl):** A tree view showing module categories: Serial (NET-SER, BS-SER, RO-SER), CAN (UC-CAN, BS-CAN, RO-CAN, RO-CAN2), Ethernet (ETH-REL/OPTO-8, WEU-REL/OPTO-8 (ETH), ETH-TTL-64, NET-ETH_LC, BS-ETH, BS-WEU (ETH), RO-ETH/LC), and Ethernet direct.
- Main Configuration Area:**
 - Module name:** BS-WEU-SER
 - IP / Hostname:** 192.168.2.131 (with a 'use Hostname' checkbox).
 - Port:** 9912 (with a 'default port' label and a gear icon).
 - Timeout [msec]:** 5000 (with a 'default timeout' label and a gear icon).
 - Encryption type:** User (with a dropdown menu and a password field containing '*****').
 - Buttons:** 'Search via Broadcast', 'Identify Module', 'Start', and 'Stop'.
- Log Section:** A 'Log:' label above a large text area for logging, with an 'Auto-scroll' checkbox and a 'Clear log' button.
- Bottom Buttons:** 'SPEICHERN/TEST' and 'BEENDEN'.

4. Wiederherstellungsoptionen

Sollte kein Zugriff mehr auf das Modul möglich sein, weil bspw. die Passwörter verloren gegangen sind, können Sie die Verschlüsselungseinstellungen wie folgt ändern:

- Per USB-Verbindung können Sie jederzeit die Einstellungen an den Zugriffsrechten im ICT-Tool vornehmen.
- Per Webinterface können Sie jederzeit die Einstellungen an den Zugriffsrechten im ICT-Tool vornehmen. Hierfür benötigen Sie lediglich die entsprechenden Login-Daten.
- Stellen Sie den DIP-Schalter 3 bei einem WEU-Modul auf "ON", bei allen anderen ETH_LC Modulen auf "OFF" und führen Sie anschließend einen Spannungsreset durch, um das Modul mit den Werkseinstellungen zu laden. Danach den DIP-Schalter 3 wieder auf OFF stellen. Beachten Sie jedoch, dass sich unter Umständen die IP Einstellungen an Ihrem Modul geändert haben!

2.3.12. WEB-Login

Hier können Sie Einstellungen an der Weboberfläche des Moduls vornehmen.

The screenshot shows the 'WEB-Login' configuration window within the DEDITEC I.C.T.-Tool. The window title is 'DT-ICT-Tool'. The top bar features the DEDITEC logo, 'I.C.T.-Tool' with a gear icon, 'Ver. 2.712', 'DELIB 2.712 (64 Bit)', and a power button icon. Below the bar, there's a 'Configuration' tab and a 'Menu' button. On the right, a status bar shows 'Module: 1.12' with an info icon, a green checkmark, and 'IP: 192.168.2.171' with a 'Deselect' button. The main content area is titled 'WEB-Login' and includes the subtitle 'Hier können Sie die Einstellungen des Webinterface ändern'. On the left, a 'Module-Config' tree shows a hierarchy: BS-WEU-JS2 (Module-ID = 42) > Modul > Netzwerk > LAN > WiFi > TCP-Verschlüsselung > NTP-Konfiguration > WEB-Login > CAN. The main configuration area contains four fields: 'Session Time' (set to 200 seconds), 'Account name' (set to 'admin'), 'Account password' (masked with dots, with a 'show password' checkbox), and 'Confirm password'. At the bottom, there are buttons for 'Save or Load config', 'LOAD DEFAULT', and 'SAVE'.

Session Time

Gibt an, nach wie viel Sekunden Inaktivität auf der Weboberfläche der Benutzer automatisch abgemeldet wird.

Einstellung von 10 - 65535 Sekunden möglich.

Account name

Gibt den Benutzernamen für die Anmeldung auf der Weboberfläche an.

Bitte beachten Sie Groß- und Kleinschreibung.

Account password

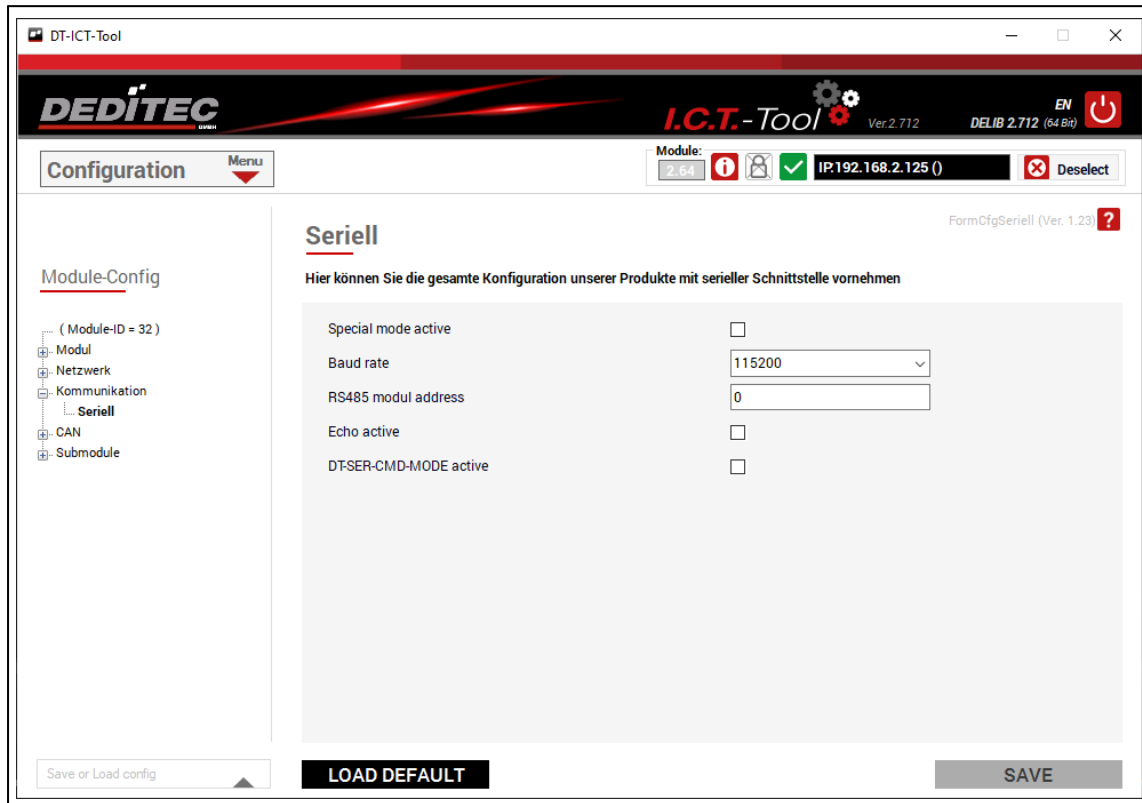
Gibt das Passwort für die Anmeldung auf der Weboberfläche an.

Aus Sicherheitsgründen muss das Passwort zusätzlich durch Wiedereingabe bestätigt werden.

Bitte beachten Sie Groß- und Kleinschreibung.

2.3.13. Serielle Konfiguration

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



Vorzugsmodus (Special mode)

Im Vorzugsmodus wird das Modul automatisch mit folgenden Einstellungen betrieben:

Baudrate: 115200

RS485 Modul Adresse: 0

Echo aktiviert: Off

DT-SER-CMD-MODE: On

Baudrate

Ist der Vorzugsmodus deaktiviert, kann die Baudrate der Kommunikation wie folgt 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.

DT-SER-CMD-MODE

Deaktivieren Sie den Registermodus um den Textmodus zu aktivieren.

Um über die serielle Schnittstelle auf das ICT-Tool zugreifen zu können, muss diese Option aktiviert sein.

Werkseinstellungen laden

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

Werkseinstellungen	
Special mode aktivieren	OFF
Baudrate	115200
RS485 Modul-Adresse	0
Echo aktivieren	OFF
DT-SER-CMD-MODE aktivieren	OFF

2.3.14. I/O Kanal-Namen

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

The screenshot shows the 'I/O Kanal-Namen' configuration window in the DEDITEC I.C.T.-Tool. The window title is 'DT-ICT-Tool'. The top bar includes the DEDITEC logo, 'I.C.T.-Tool Ver. 2.712', 'DELIB 2.712 (64 Bit)', and a power button icon. Below the top bar, there's a 'Configuration' tab and a 'Menu' button. The main area is titled 'I/O Kanal-Namen' with a subtitle 'Hier können Sie die Namen der einzelnen I/O-Kanäle ändern und speichern'. On the left, a 'Module-Config' tree shows the hierarchy: (Module-ID = 32) > Modul > Netzwerk > Kommunikation > CAN > Submodule > 0 - NET-REL16 > Info > I/O Kanal-Namen. The main content area displays 16 channels (Ch. 0 to Ch. 15) with text input fields for their names. All fields are pre-filled with 'Kanal X' (e.g., 'Kanal 0' for Ch. 0, 'Kanal 15' for Ch. 15). At the bottom, there are buttons for 'Save or Load config', 'LOAD DEFAULT', and 'SAVE'. A small 'FormCfglIONames (Ver. 1.23)' help icon is visible in the top right corner of the main area.

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

Hinweis:

Der Kanalname darf maximal 16 Zeichen lang sein.

Standard Kanalnamen festlegen

Schreibt den oben abgebildeten Text als Namensvorschlag in die Textfelder.

Zum Übernehmen auf das Modul muss zusätzlich gespeichert werden.

2.3.15. D/A-Startwerte

Hier können Sie den D/A-Modus und -Wert der jeweiligen Kanäle Ihres Modules einstellen, mit denen das Modul gestartet werden soll.

The screenshot shows the 'DT-ICT-Tool' window with the 'DEDITEC' logo and 'I.C.T.-Tool Ver. 2.712' header. The 'Configuration' menu is active. On the left, the 'Module-Config' tree shows 'Module-ID = 32' expanded, with submodules '0 - NET-REL16', '1 - NET-REL16', '2 - NET-AD16_16', and '3 - NET-DA4/8'. The 'D/A Default Werte' option is selected under 'Info'. The main area is titled 'D/A Default Werte' and contains the text 'Hier können Sie die Konfiguration des ausgewählten D/A-Submodules ändern'. Below this is a table with 8 rows, each representing an output channel. Each row has a 'Mode' column with a dropdown menu set to '16 Bit / 0-5V' and a 'Value' column with a text input field and a unit 'V'. The values are: 0,99998; 1,99996; 2,99995; 3,99993; 4,99992; 4,99992; 4,99992; 4,99992. Below the table is a button 'Set all values to 0'. At the bottom right is a 'SAVE' button. At the bottom left is a 'Save or Load config' button.

	Mode	Value
A. Output 0	16 Bit / 0-5V	0,99998 V
A. Output 1	16 Bit / 0-5V	1,99996 V
A. Output 2	16 Bit / 0-5V	2,99995 V
A. Output 3	16 Bit / 0-5V	3,99993 V
A. Output 4	16 Bit / 0-5V	4,99992 V
A. Output 5	16 Bit / 0-5V	4,99992 V
A. Output 6	16 Bit / 0-5V	4,99992 V
A. Output 7	16 Bit / 0-5V	4,99992 V

D/A-Modus

Hier können Sie den D/A-Modus einstellen, den der jeweilige Kanal beim Modulstart haben soll.

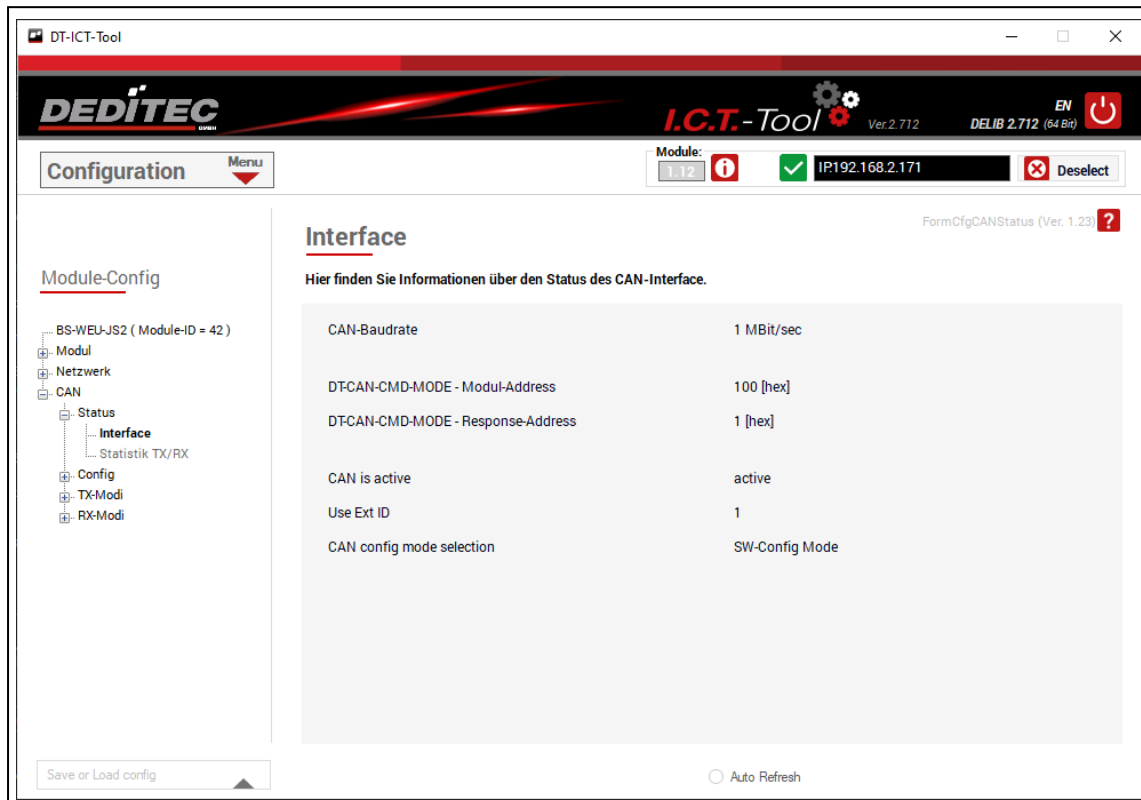
Wert

Hier können Sie den D/A-Wert einstellen, den der jeweilige Kanal beim Modulstart haben soll.

2.3.16. CAN-Konfiguration

2.3.16.1. CAN-Status Interface

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



CAN-Baudrate

Zeigt die aktuell eingestellte Baud rate Ihres CAN-Interfaces an

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

Legt fest, unter welcher Adresse das Modul im CAN-Bus identifiziert wird.

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

Legt fest, an welche Modul-Adresse eine Bestätigung gesendet wird, sobald ein Paket empfangen wurde.

CAN ist aktiv

(wird nur angezeigt, wenn das Modul diese Funktion unterstützt)

Use EXT ID

Zeigt die verwendete extended ID des Interface-Moduls an

CAN config mode selection

Zeigt den verwendeten CAN-Konfigurations-Modus an

2.3.16.2. CAN-Statistik TX/RX

Hier werden alle wichtigen Statistiken zu den TX- und RX-Paketen angezeigt.

The screenshot shows the DT-ICT-Tool interface. The top bar includes the DEDITEC logo, the I.C.T.-Tool logo with version 2.712, and a power button icon. The main menu on the left shows a tree structure with 'Module-Config' selected. The right pane displays the 'Statistik TX/RX' configuration and statistics. The configuration section shows 'Module: 1.12' and 'IP: 192.168.2.171'. The statistics section shows 'FormCfgCANStatistic (Ver. 1.23)' and a table of TX and RX statistics.

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

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

Folgende TX-Informationen werden dargestellt:

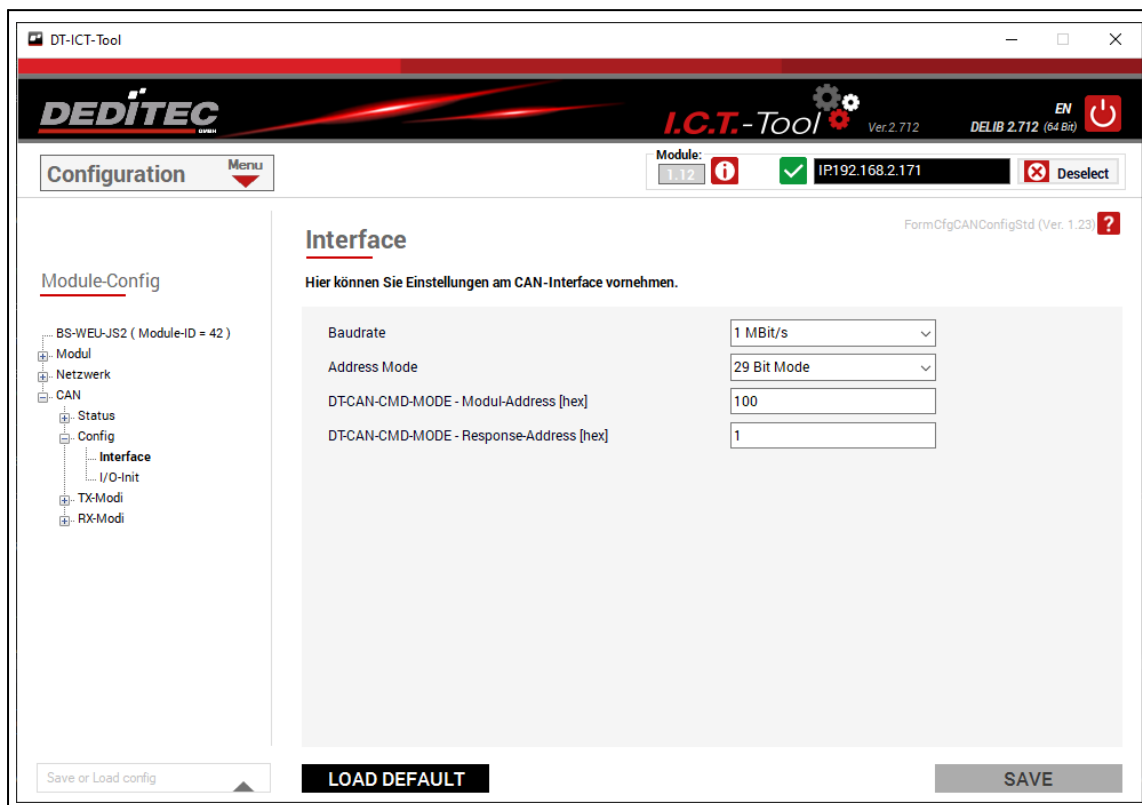
- Gesamtanzahl gesendeter Pakete
- Anzahl gesendeter Pakete pro Sekunde
- Durchschnittliche Zeit, die für das versenden eines Paketes benötigt wird

Folgende RX-Informationen werden dargestellt:

- Gesamtanzahl empfangener Pakete

2.3.16.3. CAN Main Interface

Mit Hilfe dieser Einstellungen lässt sich das CAN Interface konfigurieren.



Baudrate

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

Address Mode

Der Address Mode gibt vor, wie viele Bits zur Adressierung verwendet werden.

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

Legt fest, unter welcher Adresse das Modul im CAN-Bus identifiziert wird.

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

Legt fest, an welche Modul-Adresse eine Bestätigung gesendet wird, sobald ein Paket empfangen wurde.

2.3.16.4. CAN Main I/O Init

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

The screenshot shows the 'I/O-Init' configuration window in the DT-ICT-Tool. The window has a title bar 'DT-ICT-Tool' and a header with the 'DEDITEC' logo, 'I.C.T.-Tool' version 2.712, and 'DELIB 2.712 (64 Bit)' with an 'EN' button. Below the header is a 'Configuration' menu and a 'Module:' field showing '1.12' and 'IP:192.168.2.171' with a 'Deselect' button. The main area is titled 'I/O-Init' and contains the text 'Hier können Sie die CAN-Einstellungen der angeschlossenen Submodule konfigurieren.' Below this is a table of settings:

A/D Mode	16 Bit / ± 20V
A/D Filter	None
D/A Mode	16 Bit / ± 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

At the bottom of the window are buttons for 'Save or Load config', 'LOAD DEFAULT', and 'SAVE'. A sidebar on the left shows a tree view with 'Module-Config' expanded, showing 'BS-WEU-JS2 (Module-ID = 42)' and its sub-items: 'Modul', 'Netzwerk', 'CAN', 'Status', 'Config', 'Interface', 'I/O-Init', 'TX-Modi', and 'RX-Modi'.

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

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. 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 20 ns und 5 ms ausgewählt werden.

2.3.16.5. CAN TX-Mode

Hier können Sie Einstellungen an den TX-Paketen vornehmen.

The screenshot shows the 'DT-ICT-Tool' window with the 'DEDITEC' logo and 'I.C.T.-Tool' branding. The interface is in German. The left sidebar shows a tree view under 'Konfiguration' with 'TX-Mode[1]' selected. The main area is titled 'TX-Mode[1]' and contains the following settings:

- Aktivieren:** ☒
- Triggermodus:** TX-Event[1] (dropdown)
- Intervall:** 1 (dropdown) * 1 sec (dropdown)
- Erweiterte ID verwenden:** 29 Bit Mode (dropdown)
- Sende zu CAN-ID [hex]:** 123 (text input)
- Modus:** DO Readback 1-64 (dropdown)

At the bottom, there are buttons for 'STANDARD LADEN' and 'SPEICHERN', and a 'Save or Load config' button on the left.

Activate

Aktiviert diesen TX-Mode.

Triggermodus

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

Interval

Wird der Interval-Modus eingestellt, können Sie zusätzlich angeben, in welchem Zeitintervall das TX-Paket gesendet werden soll.

Use extended ID

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]

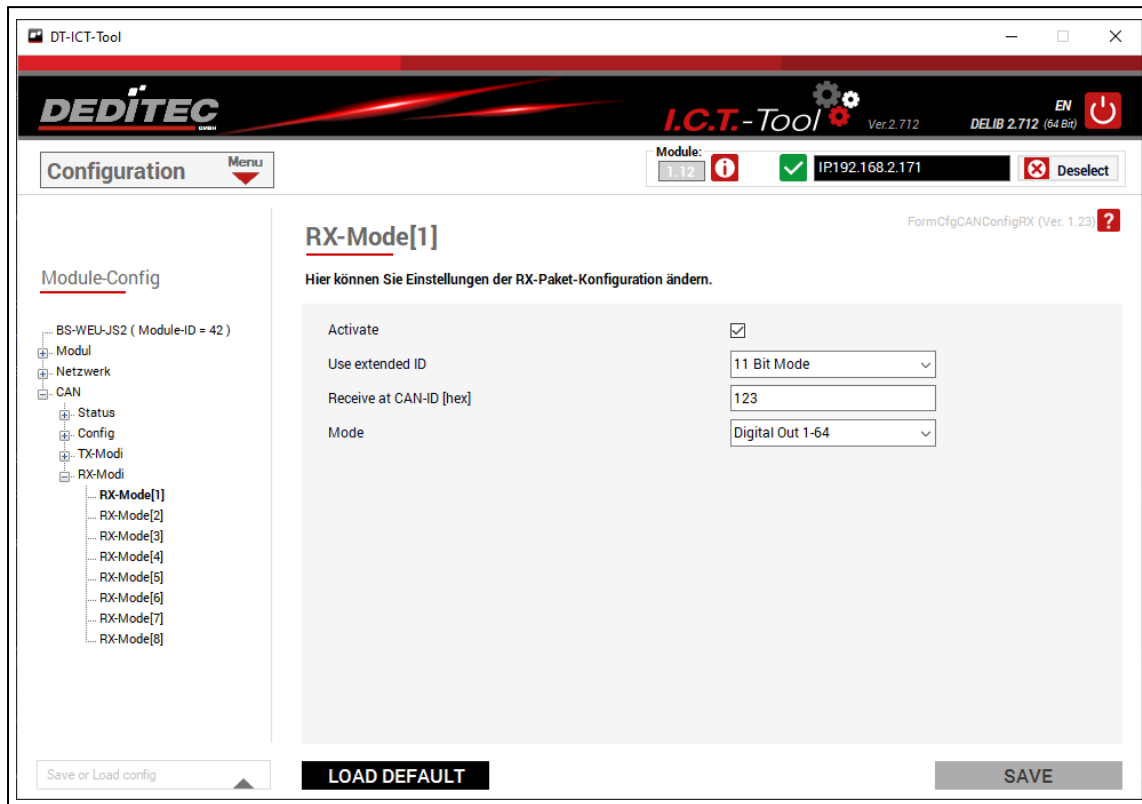
Gibt an, an welche Adresse die CAN-Pakete gesendet werden sollen.

Mode

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

2.3.16.6. CAN RX-Mode

Hier können Sie Einstellungen an den RX-Paketen vornehmen.



Activate

Aktiviert diesen RX-Mode.

Use extended ID

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

Receive at CAN-ID [hex]

Gibt an, auf welcher Adresse die Pakete empfangen werden sollen.

Mode

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.

2.3.17. Watchdog-Konfiguration

Hier können Sie Einstellungen an Ihrem Watchdog-Stick vornehmen.

The screenshot shows the 'Watchdog Parameter' configuration window in the DT-ICT-Tool. The window has a title bar 'DT-ICT-Tool' and a header with 'DEDITEC', 'I.C.T.-Tool Ver. 2.712', 'DELIB 2.712 (64 Bit)', and a power button icon. Below the header is a 'Configuration' menu and a 'Module:' dropdown set to '2.00'. A green checkmark icon is next to 'USB-Watchdog (0)', and a red 'X' icon is next to 'Deselect'. The main content area is titled 'Watchdog Parameter' and includes the text 'Hier können Sie die Watchdog-Konfigurationen ändern'. It features a 'Global Watchdog Timeout' field set to '5000 ms' and a 'Load default' button. Below this is a table with columns 'State', 'Relais 1', 'Relais 2', and 'Going to the state'. The table has five rows: 'Deactivated', 'Activated', 'Retriggered', 'Timeout', and 'direct Relais Setting'. The 'Relais 1' and 'Relais 2' columns contain dropdown menus. The 'Going to the state' column contains text descriptions and functions. At the bottom, there is a 'Save or Load config' button and a large 'SAVE' button.

State	Relais 1	Relais 2	Going to the state
Deactivated	ON	OFF	- Power On - DELIB Function: DapiWatchdogDisable
Activated	ON	OFF	- DELIB Function: DapiWatchdogEnable
Retriggered	ON	OFF	- DELIB Function: DapiWatchdogRetrigger
Timeout			- TIMEOUT occurred
direct Relais Setting	ON	OFF	
after	5000 ms	5000 ms	
Relais switch	ON	OFF	

Deactivated

Überwachung Relais 1 und 2 nicht aktiv

Activated

Überwachung Relais 1 und 2 aktiv

Retriggered

Watchdog Zeit Relais 1 und 2 wird zurückgesetzt

Timeout

Watchdog Zeit Relais 1 und 2 wurde nicht zurückgesetzt

Relais Setting after x msec

Zeitverzögertes Schalten der Relais nach einem Timeout

2.4. M2M Konfiguration

M2M steht für „Machine-to-Machine“ und stellt eine direkte Kommunikation zwischen zwei oder mehreren Geräten dar.

Es ermöglicht z.B. die automatisierte Übertragung von Messwerten oder eine Fernüberwachung von Maschinen und Anlagen.

Die Möglichkeit einer M2M Übertragung zwischen unseren DEDITEC Modulen wird von zwei Schnittstellen unterstützt, Ethernet und CAN.

Bei der M2M-Kommunikation über Ethernet muss beachtet werden, dass das Sendemodul explizit über die M2M-Funktion verfügen muss.

Als Empfangsmodule können jedoch alle DEDITEC-Produkte mit einer Ethernet-Schnittstelle verwendet werden.

Die M2M-Kommunikation über die CAN-Schnittstelle wird von all unseren CAN-Modulen unterstützt.

Einstellungen können im CAN-Konfigurationsbereich des ICT-Tools vorgenommen werden.

Mehr Informationen dazu finden Sie im Kapitel: **CAN-Konfiguration**

2.4.1. Statistik

Hier erhalten Sie einen Überblick über die Aktivität von Sende- und Empfangsmodul

The screenshot shows the DT-ICT-Tool interface. At the top, there's a header with the DEDITEC logo, I.C.T.-Tool version 2.756, and ADMIN status. Below the header, a dropdown menu shows 'M2M'. To the right, there's a 'Module:' section with a list of modules, including 'IP192.168.2.131 (BS-WEU)' which is selected. A 'Deselect' button is next to it. The main content area is titled 'Übersicht' and contains a table with columns for 'Aktion (Sendemodul)' and 'Empfangsmodul'. The table has 8 rows of data, all showing 0 actions and 0 errors. A sidebar on the left shows a tree view with 'BS-WEU', 'Status', 'Übersicht', 'Empfangsmodule', and 'Aktionen'. At the bottom, there's a 'Save or Load config' button and an 'Auto Refresh' checkbox.

Aktion (Sendemodul)					Empfangsmodul		
Nr.	Aktiv	Anzahl Aktionen	Aktionen/Sek	DELIB-Fehler	Verbunden	IP-Adresse	Verb. Versuche
1	-	0	0	0	-	0.0.0.0	0
2	-	0	0	0	-	0.0.0.0	0
3	-	0	0	0	-	0.0.0.0	0
4	-	0	0	0	-	0.0.0.0	0
5	-	0	0	0	-	0.0.0.0	0
6	-	0	0	0	-	0.0.0.0	0
7	-	0	0	0	-	0.0.0.0	0
8	-	0	0	0	-	0.0.0.0	0

Folgende Sendemodul-Informationen werden dargestellt:

- Status der Aktivität: Wenn das Häkchen gesetzt ist, wird diese Aktion gerade ausgeführt.
- Gesamtzahl der durchgeführten Aktionen. Wird nach einem Neustart des Moduls auf 0 zurückgesetzt.
- Anzahl der Aktionen pro Sekunde.
- Anzahl der DELIB-Fehler, die bei der Ausführung von Befehlen aus der DEDITEC-Treiberbibliothek auftreten. Hilfestellung für das Debuggen von Fehlern finden Sie im Kapitel: **DELIB-DebugView-Global**

Die folgenden Informationen zum Empfängermodul werden angezeigt:

- Verbunden: Wenn das Häkchen gesetzt ist, sind die Sender- und Empfängermodul miteinander verbunden.
- IP-Adresse des Empfängermoduls.
- Anzahl der Verbindungsversuche.

2.4.2. Empfangsmodule

Hier können Sie die Netzwerkeinstellungen des Empfangsmoduls einstellen.

The screenshot shows the DT-ICT-Tool interface. At the top, there's a header with 'DEDITEC' and 'I.C.T.-Tool Ver. 2.756'. Below the header, a dropdown menu is set to 'M2M'. To the right, there's a status bar showing 'Module: 1.17', a green checkmark, and 'IP192.168.2.131 (BS-WEU)' with a 'Deselect' button. The main area is titled 'Ziel [1]' and contains the text 'Hier können Sie die Netzwerkeinstellungen des Empfangsmoduls eingeben'. On the left, a tree view shows 'Machine to Machine' with sub-items: 'BS-WEU', 'Status', 'Empfangsmodule' (expanded), 'Ziel [1]', 'Ziel [2]', 'Ziel [3]', 'Ziel [4]', and 'Aktionen'. The configuration fields for 'Ziel [1]' are: 'Bezeichnung' (Garagentor), 'TCP-IP-Adresse/Host' (192.168.2.90), 'Hostname verwenden' (checkbox), 'TCP-Port' (9912, with a 'Standard Port setzen' button), 'TCP-Timeout [msec]' (5000, with a 'Standard Timeout setzen' button), and 'Verschlüsselungsmodus' (Deaktiviert). At the bottom, there are buttons for 'Save or Load config', 'STANDARD LADEN', and 'SPEICHERN'.

Bezeichnung

Geben Sie hier den gewünschten Namen für das Empfangsmodul an.

IP / Port

Hier können Sie die Netzwerkeinstellungen des Empfangsmoduls angeben.

Timeout[msec]

Der Timeout gibt an, wie lang versucht werden soll eine Verbindung zum Modul herzustellen. Ist diese Zeit zu gering eingestellt, kann unter Umständen der Timeout getriggert werden, bevor das Modul ordnungsgemäß geöffnet wurde. Die Dauer der Verbindungsversuche nach einem Verbindungsabbruch mit dem Modul wird ebenfalls mit dem Timeout angegeben.

Verschlüsselungstyp/ -passwort

Ist auf dem Empfangsmodul die TCP-Verschlüsselung aktiviert, stellen Sie hier bitte den entsprechenden Verschlüsselungstyp ein, mit dem dazugehörigen Passwort.

Mehr Informationen dazu finden Sie im Kapitel: **TCP-Verschlüsselung**

2.4.3. Aktion

Mit einer Aktion definieren Sie, welches Eingangssignal vom Sendemodul an welchen Ausgang eines Empfangsmoduls gesendet werden soll.

In diesem Bildbeispiel werden alle 100ms DI-Daten von Kanal 3(CH Start) bis Kanal 8 ((CH Start) + (CH Count)) an die Kanäle 0-5 des Ziel Moduls gesendet.

DT-ICT-Tool

DEDITEC

I.C.T.-Tool Ver. 2.756 ADMIN DELIB 2.738 (64 Bit)

M2M

Module: 1.17 IP192.168.2.131 (BS-WEU) Deselect

FormM2MJobs (Ver. 1.30)

Aktion [1]

Hier können Sie einstellen, welche Eingänge zum Empfangsmodul gesendet werden sollen.

Modus: Digitale Eingänge

Sub-Modus: Normal

Empfangsmodul: Ziel: 0 "Garagentor" (192.168.2.90)

Intervall: 1 * 100ms

Sendemodul Startkanal (Erster Kanal ist 0): 3

Anzahl der zu sendenden Kanäle: 5

Empfangsmodul Startkanal (Erster Kanal ist 0): 0

Save or Load config

STANDARD LADEN

SPEICHERN

Aktion

Hier können Sie angeben, ob Sie digitale oder analoge Eingangssignale senden möchten.

Modus

Im normalen Modus werden die aktuellen Zustände der Eingangskanäle an das Empfangsmoduls übertragen.

Im Testmodus werden abwechselnd High / Low Pakete an das Empfangsmoduls übertragen. Somit können Sie Ihre Sende-/Empfangskonfiguration einfacher überprüfen.

Empfangsmodul

Hier können Sie auswählen, welches Empfangsmodul die Informationen erhalten soll. Das Empfangsmodul wird im Reiter "Empfangsmodule" definiert.

Intervall

Stellen Sie ein, in welchen Zeitabständen das Sendemodul die Informationen verschicken soll.

Sendemodul Startkanal

Stellen Sie hier ein, ab welchem Eingangskanal die Informationen an das Empfangsmodul verschickt werden sollen.

Anzahl der zu sendenden Kanäle

Geben Sie hier die Anzahl der Eingangskanäle an, die an das Empfangsmodul übertragen werden sollen, beginnend mit Ihrem Eintrag im Feld "Sendemodul Startkanal"

Empfangsmodul Startkanal

Stellen Sie hier ein, ab welchem Ausgangskanal des Empfangsmodul, die Informationen vom Sendemodul ausgegeben werden sollen.

2.5. Event-Control

Mit dem Event-Control lassen sich Aktionen konfigurieren, die vom Modul automatisch ausgeführt werden, sobald eine bestimmte Bedingung erfüllt wird. Sowohl die Ereignisbedingung als auch die Aktion lassen sich separat einstellen.

Welche Ereignisse und Aktionen konfiguriert werden können, ist Modul abhängig.

Hilfe zum Konfigurieren des Events finden Sie hier: **Konfiguration Events**

Hilfe zum Konfigurieren der Aktion, finden Sie hier: **Konfiguration Aktionen**

Folgende Modul-Serien werden vom Event-Control unterstützt:

- BS-WEU-Serie
- BS-USB-Serie
- NET-Serie
- UC-Serie
- CANBox

Folgende Modul-Serien werden vom Event-Control **nicht** unterstützt:

- Starter-Serie
- RO-Serie
- COS-Serie

2.5.1. Statistik

Hier werden verschiedene Informationen zu den Events und Aktionen angezeigt.

The screenshot displays the DT-ICT-Tool interface. The top bar shows the DEDITEC logo, the I.C.T.-Tool version (Ver. 2.756), and the user 'ADMIN' with a 'DELIB 2.738 (64 Bit)' status. The 'Event-Control' section is active, showing a tree view on the left with 'BS-WEU-CAN' expanded. The main area displays 'Events 1 - 8' with a table of events and actions.

Events				Aktionen		
Nr.	Event	Timer	Anz. Event-Aktionen	Nr.	Anz. Aktionen	Anz. Else-Aktionen
1	DO-Readback	4998	0	1	1221	-
2	DI	0	14020	1	1221	-
3	-	-	-	-	-	-
4	-	-	-	-	-	-
5	-	-	-	-	-	-
6	-	-	-	-	-	-
7	-	-	-	-	-	-
8	-	-	-	-	-	-

2.5.2. Konfiguration Events

Mit Hilfe eines Events, wird eine bestimmte Aktion ausgeführt, sobald eine Bedingung erfüllt ist.

So können z.B. automatisch Spannungen über den D/A-Wandler ausgegeben oder Relais geschaltet werden, wenn eine zuvor definierte Schaltschwelle an den A/D-Eingängen erreicht wird,

oder ein Signal an einem digitalen Eingang erfasst wird.

Auch eine Zeitspanne wie lang die Bedingung erfüllt sein muss, um eine Aktion auszuführen, kann vorgegeben werden.

Diese Einstellungen werden direkt in das **Modul-Configuration-Memory** des Modul gespeichert.

Abhängig von der Ausstattung Ihres Produktes, stehen Ihnen unterschiedliche Modi zur Auswahl:

"DI"

High oder Low Signal eines digitalen Eingangs.

"DI-Flip-Flop"

Bei einer Zustandsänderung des digitalen Eingangs, wird der Flip-Flop des Kanals gesetzt und anschließend wieder zurückgesetzt.




"DO-Readback"

Der Status des Ausgangs-Kanals wird ausgelesen.




"A/D"



Die aktuell anliegende Spannung des analogen Eingangs wird ausgelesen.

DT-ICT-Tool



Ver. 2.756
ADMIN
DELIB 2.738 (64 Bit)


Event-Control

Module: 1.17


BS-WEU (0)

Deselect

FormECEvents (Ver. 1.30)



Event-Control

BS-WEU
(Module - ID = 0x29)

Status

Konfiguration Events

Event [1]
Event [2]
Event [3]
Event [4]
Event [5]
Event [6]
Event [7]
Event [8]
Event [9]
Event [10]
Event [11]
Event [12]
Event [13]
Event [14]
Event [15]
Event [16]

Konfiguration Aktionen

Save or Load config

Event [1]

Hier kann das Event konfiguriert werden

Modus	A/D
Kanal	0
A/D-Modus	16 Bit / 0-10V
Bedingung	innerhalb Wertebereich von Wert 1 und Wert 2
Wert 1	4 V
Wert 2	7 V
Minstdauer der Bedingung	3 * 10ms
Auszuführende Aktion, wenn Bedingung erfüllt	Aktion[1] TX: TX[0], Zeitlimit: 900 ms

STANDARD LADEN

SPEICHERN

2.5.3. Konfiguration Aktionen

Eine Aktion legt fest, was nach Erfüllen einer Bedingung innerhalb eines Events passieren soll.

Abhängig von der Ausstattung Ihres Produktes, können unterschiedliche Modi definiert werden:

"DO Ausgang setzen"

Schaltet einen digitalen Ausgang ein oder aus.

"D/A Ausgang setzen"

Gibt Spannung an einem analogen Ausgang aus.

"CAN Paket senden"

Das eingestellte CAN TX-Event wird ausgelöst. Für diese Funktion muss der Triggermodus im CAN TX-Paket auf "TX-EVENT" gesetzt werden.

Mehr Informationen zur Konfiguration des TX-Mode finden Sie im Kapitel: **CAN TX-Mode**.

The screenshot displays the DT-ICT-Tool interface. At the top, a status bar shows the user 'ADMIN' and the version 'Ver. 2.756'. Below this, a navigation bar includes a dropdown menu for 'Event-Control' and a status area showing 'Module: 1.17' and 'IP192.168.2.75'. The main workspace is divided into two sections. On the left, a sidebar titled 'Event-Control' contains a tree view with the following structure: 'BS-WEU-CAN' (Module - ID = 0x2A), 'Status', 'Konfiguration Events', and 'Konfiguration Aktionen'. Under 'Konfiguration Aktionen', a list of actions from 'Aktion [1]' to 'Aktion [16]' is visible. The right section, titled 'Aktion [1]', contains the instruction 'Hier kann die Aktion konfiguriert werden'. Below this, three configuration fields are shown: 'Modus' set to 'CAN Paket senden', 'CAN TX-Event[1..8] auslösen' set to 'CAN TX-Event [1]', and 'Zeitlimit für erneuten Aufruf' set to '9' and '* 100ms'. At the bottom of the interface, there are two buttons: 'STANDARD LADEN' and 'SPEICHERN'.

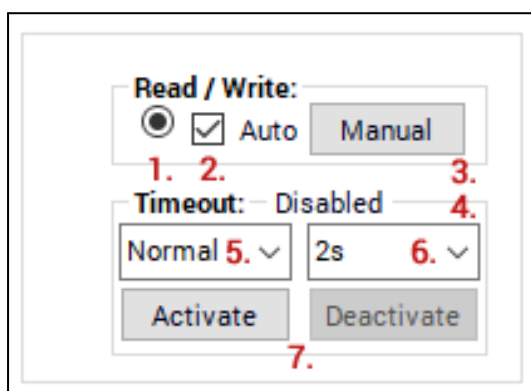
2.6. Modul testen

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

2.6.1. Timeout Test-Funktion

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

Mit diesen Funktionen lässt sich ein Timeout-Fall auslösen.

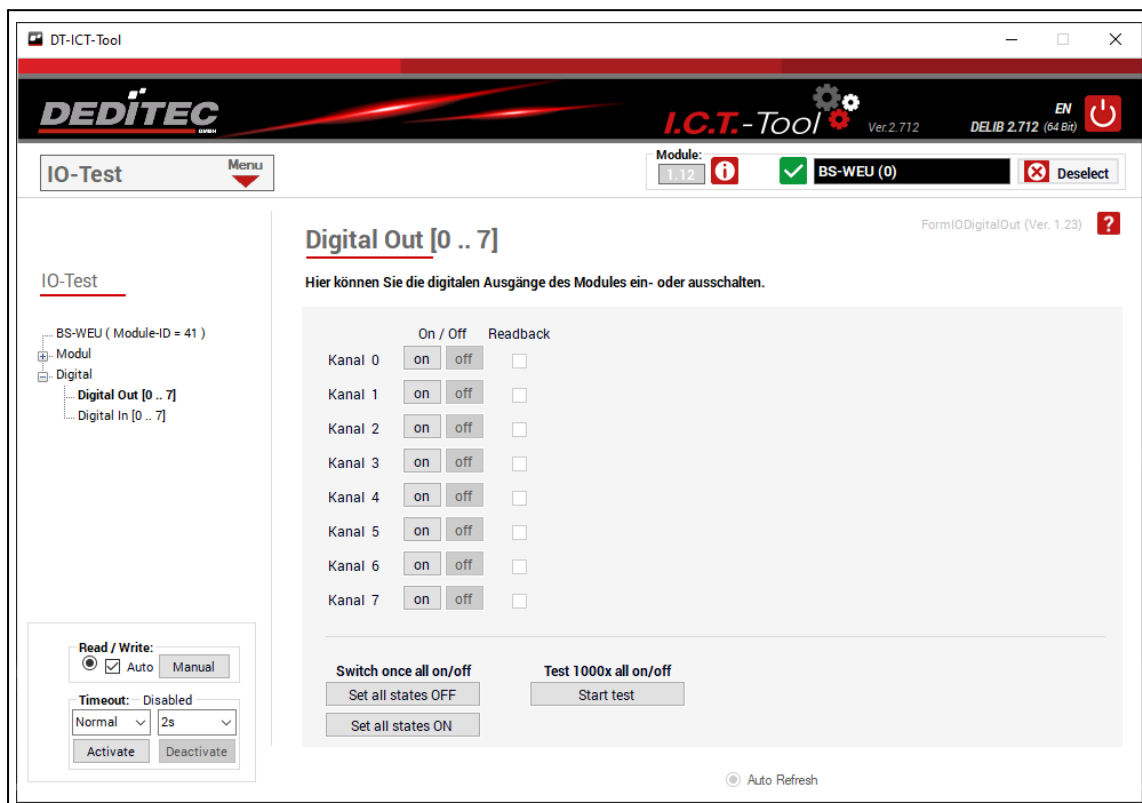


- 1 Das Feld zeigt durch wiederholtes Blinken an, ob eine Verbindung zum Modul besteht. Bleibt das Feld leer, ist die Kommunikation unterbrochen.
- 2 Durch das Entfernen des Häkchens, wird die Verbindung unterbrochen und somit ein Timeout-Fall ausgelöst. Ein erneutes Setzen des Hakens stellt die Verbindung 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 gewünschten 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.

2.6.2. Digital Out

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

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



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 dem Knopf "set" starten Sie den Timer.

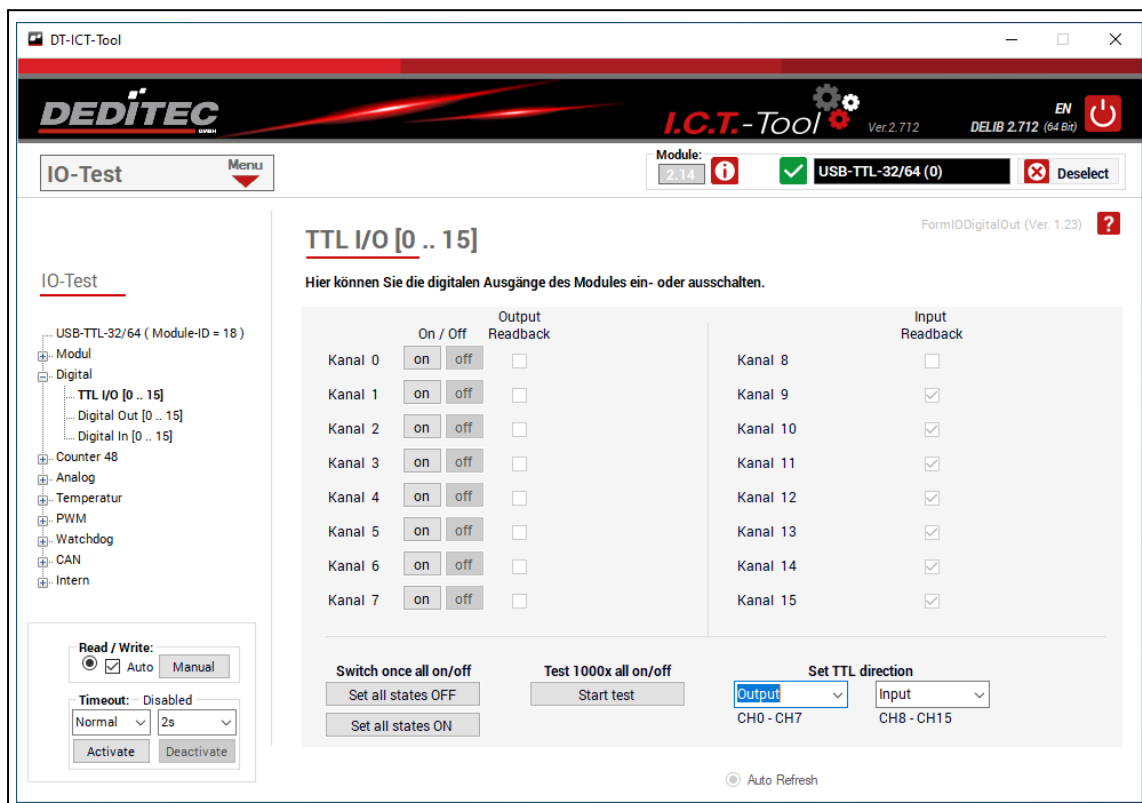
Invert DO-Timer

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

2.6.3. Digital TTL

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).



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 / Test 1000x all on/off

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

Mit dem Testknopf "Test 1000x ll on/off" werden die Ausgänge des Modules automatisch 1000x ein- und wieder ausgeschaltet.

Set TTL direction

Hier können Sie die Richtung (Input/Output) Ihres Modules 8-Kanalweise einstellen.

Diese Einstellungen bleiben bis Modulneustart erhalten.

2.6.4. Digital In

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

The screenshot shows the DT-ICT-Tool interface. At the top, there's a header with the DEDITEC logo, I.C.T.-Tool version 2.712, and a power button icon. Below the header, there's a navigation bar with 'IO-Test' selected. The main content area is titled 'Digital In [0 .. 7]' and includes a description: 'Hier können Sie den Status der digitalen Eingangskanäle ablesen.' Below this is a table with columns 'State on/off', 'Counter', and 'FlipFlop'. The table lists channels from Kanal 0 to Kanal 7. Each channel has a checkbox for 'State on/off', a text input for 'Counter' (all showing '0'), and a checkbox for 'FlipFlop'. At the bottom, there's a checkbox for 'Read with reset'. The interface also includes a sidebar with a tree view showing the module structure and an 'Auto Refresh' button at the bottom right.

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

Read with reset ☐

Auto Refresh

State on/off

Zeigt den aktuellen Zustand der einzelnen Eingangskanäle.

Counter

Zeigt die Zählerstände der Eingangszähler an.

FlipFlop

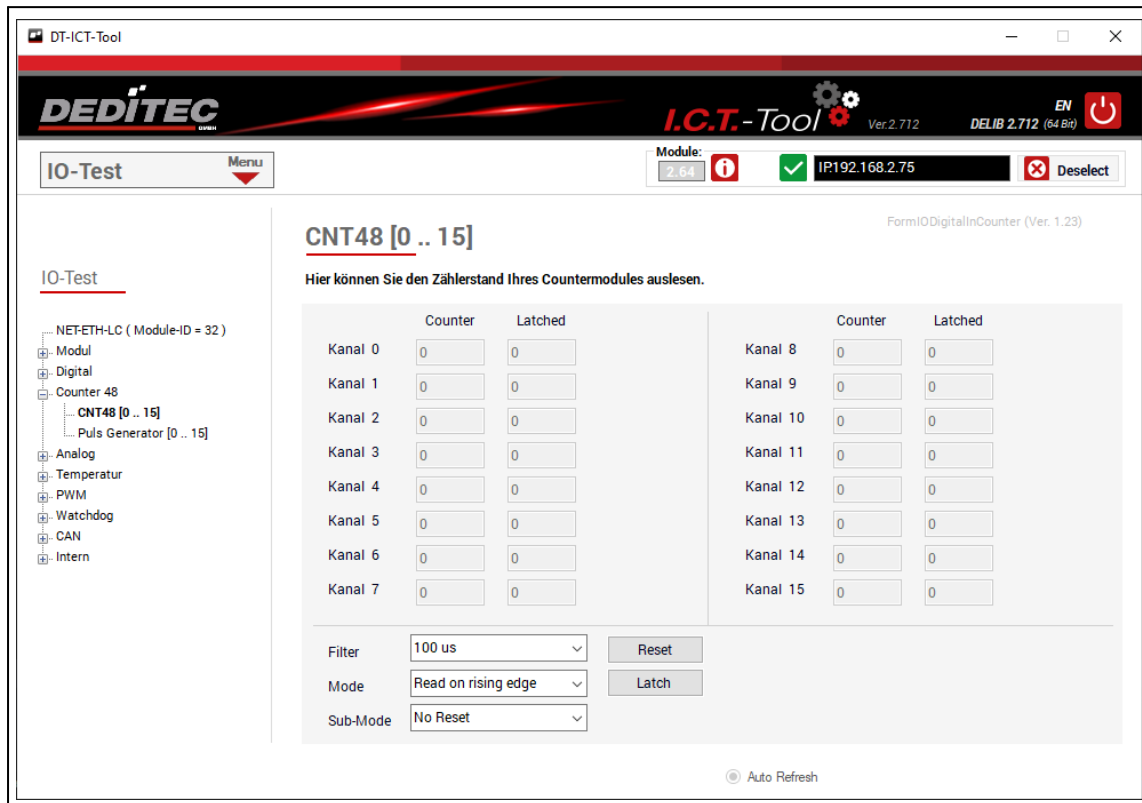
Zeigt die Änderung der Eingangszustände seit dem letzten Auslesen an.

Read with reset

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

2.6.5. Digital In Counter

Hier können Sie Ihr Digital In Counter Modul testen



Counter

Zeigt den aktuellen Zählerstand des Kanals an.

Latched

Zeigt den aktuellen Zählerstand des Zwischenspeichers an.

Filter

Hier kann der Eingangsfilter eingestellt werden.

Mode

Hier kann der Counter-Modus eingestellt werden.

Sub-Mode

Hier kann der Counter-Sub-Modus eingestellt werden.

Reset

Hier werden alle Zählerstände zurück gesetzt.

Latch

Liest alle Zählerstände aus dem Zwischenspeicher aus und stellt sie dar.

2.6.6. Analog In

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

The screenshot shows the DT-ICT-Tool software interface. At the top, there's a header with the DEDITEC logo, I.C.T.-Tool logo, version 2.712, and a power button icon. Below the header, there's a navigation bar with 'IO-Test' selected. The main area is titled 'Analog In [0 .. 15]' and contains a table of 16 channels (Kanal 0 to Kanal 15) with their respective values. Below the table, there are two sections: 'Set mode for all channels' and 'Set filter for all channels'. The 'Set mode for all channels' section has a dropdown menu set to '16 Bit / ± 10V' and a 'Set' button. The 'Set filter for all channels' section has a dropdown menu set to '0' and a 'Set' button. At the bottom right, there is an 'Auto Refresh' checkbox.

Kanal	Value
Kanal 0	0,000 V
Kanal 1	-0,002 V
Kanal 2	-0,001 V
Kanal 3	-0,002 V
Kanal 4	-0,001 V
Kanal 5	0,000 V
Kanal 6	-0,002 V
Kanal 7	0,000 V
Kanal 8	0,002 V
Kanal 9	-0,001 V
Kanal 10	-0,001 V
Kanal 11	-0,002 V
Kanal 12	-0,001 V
Kanal 13	0,000 V
Kanal 14	0,000 V
Kanal 15	-0,001 V

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.

2.6.7. Analog Out

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

The screenshot shows the DT-ICT-Tool software interface. At the top, there's a header with the DEDITEC logo, I.C.T.-Tool branding, version 2.712, and a power button icon. Below the header, there's a navigation bar with 'IO-Test' selected. The main content area is titled 'Analog Out [0 .. 7]' and includes a subtitle 'Hier können Sie die analogen Ausgänge des Modules setzen.' Below this, there's a table with columns 'Value', 'Mode', and 'Readback'. The table lists 8 channels (Kanal 0 to Kanal 7). Each channel has a 'Value' input field, a 'Mode' dropdown menu, and a 'Readback' field. The 'Value' fields contain values like 0,99998, 1,99996, 2,99995, 3,99993, 4,99992, 4,99992, 4,99992, 4,99992. The 'Mode' dropdowns are all set to '16 Bit / 0-5V'. The 'Readback' fields show the same values as the 'Value' fields. Below the table, there are two sections: 'Set mode for all channels' with a dropdown set to '16 Bit / 0-5V' and a 'Set' button, and 'Set test values for all channels' with a 'Test Values' button. On the left side, there's a sidebar with a tree view showing the module structure: (Module-ID = 32), Modul, Digital, Analog, Analog In [0 .. 15], Analog Out [0 .. 7], and CAN. Below the sidebar, there's a 'Read / Write' section with radio buttons for 'Auto' (selected) and 'Manual', and a 'Timeout' section with a dropdown set to 'Normal' and a '2s' value. There are also 'Activate' and 'Deactivate' buttons. At the bottom right, there's an 'Auto Refresh' checkbox.

	Value	Mode	Readback
Kanal 0	0,99998	16 Bit / 0-5V	0,99998 V
Kanal 1	1,99996	16 Bit / 0-5V	1,99996 V
Kanal 2	2,99995	16 Bit / 0-5V	2,99995 V
Kanal 3	3,99993	16 Bit / 0-5V	3,99993 V
Kanal 4	4,99992	16 Bit / 0-5V	4,99992 V
Kanal 5	4,99992	16 Bit / 0-5V	4,99992 V
Kanal 6	4,99992	16 Bit / 0-5V	4,99992 V
Kanal 7	4,99992	16 Bit / 0-5V	4,99992 V

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.

2.6.8. Temp

Hier können Sie die angelegten Temperaturen und den Status des Sensors auslesen.

The screenshot shows the DT-ICT-Tool interface. The top header features the DEDITEC logo, the I.C.T.-Tool logo, and version information (Ver. 2.712, DELIB 2.712 (64 Bit)). The main content area is titled "Temp [0 .. 15]" and includes a subtitle "Hier können Sie die angelegten Temperaturen kanalweise auslesen." Below this is a table with two columns of temperature channels (Kanal 0 to 15). The first column shows Kanal 0 with a value of 24.52 °C, while the rest of the channels show "Sensor offen". The second column shows all channels from 8 to 15 as "Sensor offen". The interface also includes a sidebar with a tree view of modules, a top menu bar, and a bottom status bar with an "Auto Refresh" button.

Kanal	Value	Unit	Status
Kanal 0	24.52	°C	
Kanal 1	-	°C	Sensor offen
Kanal 2	-	°C	Sensor offen
Kanal 3	-	°C	Sensor offen
Kanal 4	-	°C	Sensor offen
Kanal 5	-	°C	Sensor offen
Kanal 6	-	°C	Sensor offen
Kanal 7	-	°C	Sensor offen
Kanal 8	-	°C	Sensor offen
Kanal 9	-	°C	Sensor offen
Kanal 10	-	°C	Sensor offen
Kanal 11	-	°C	Sensor offen
Kanal 12	-	°C	Sensor offen
Kanal 13	-	°C	Sensor offen
Kanal 14	-	°C	Sensor offen
Kanal 15	-	°C	Sensor offen

Value

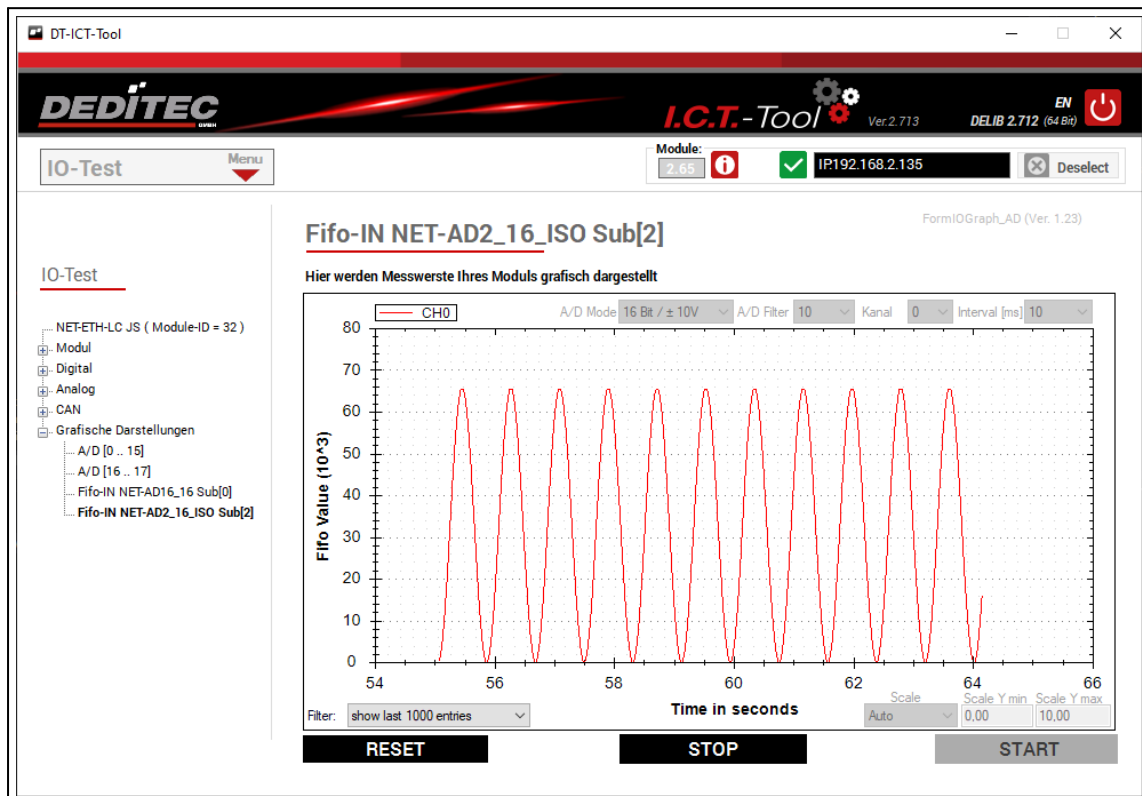
Gibt die aktuell angelegte Temperatur wieder.

Status

Zeigt den aktuellen Status des Temperatursensors an.

2.6.9. Grafische Darstellung

Hier können Sie sich gelesene A/D-, Temperatur- und Fifo-Werte grafisch darstellen lassen.



A/D-Mode

Hier können Sie den A/D-Mode des A/D-Wandlers angeben

A/D-Filter

Hier können Sie angeben, welcher A/D-Filter verwendet werden soll

Kanal

Gibt an, welcher Kanal grafisch dargestellt werden soll

Interval [ms]

Gibt an, in welchem Intervall die Daten gelesen werden sollen

Filter

Gibt an, wie viele Daten dargestellt werden sollen

Start / Stop

Startet oder beendet die grafische Ausgabe

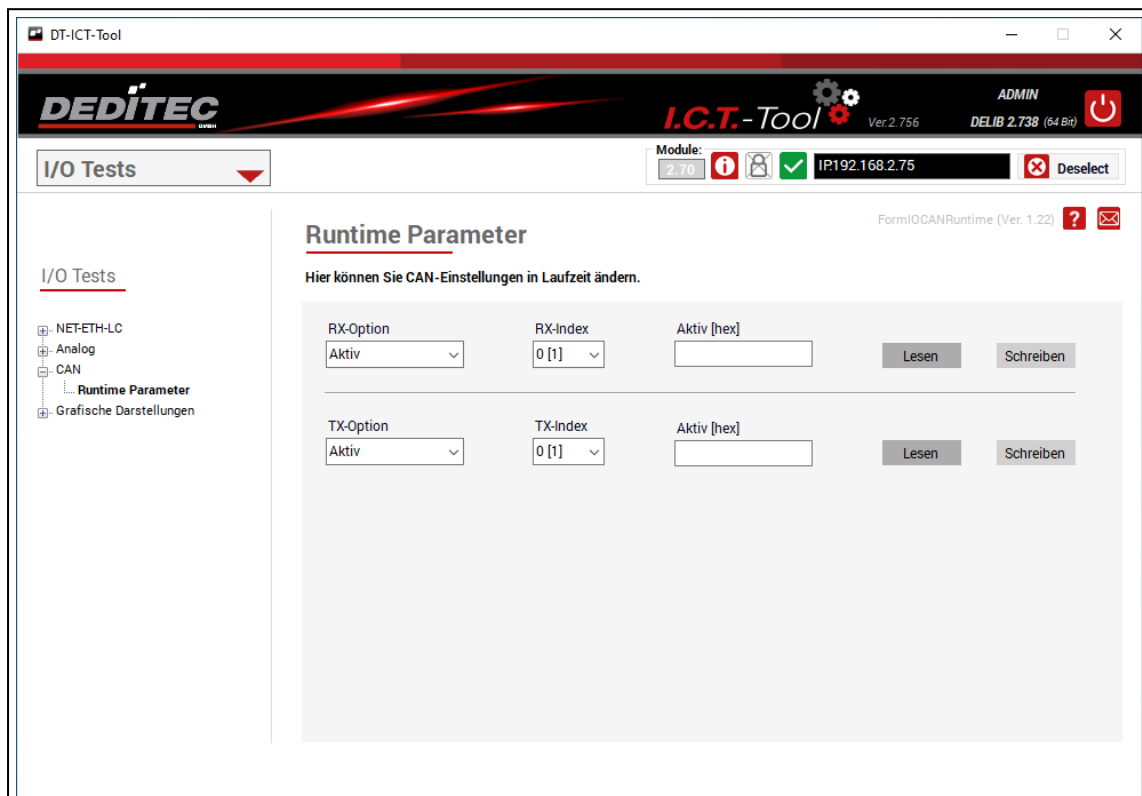
Reset

Löscht die aktuelle Ausgabe

2.6.10. CAN Runtime Parameter

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

Diese Änderungen werden nicht in das Modul gespeichert und gehen nach einem Modulneustart verloren.



Read

Liebt die aktuelle Konfiguration des jeweiligen Parameters aus

Write

Speichert die aktuelle Konfiguration des jeweiligen Parameters

RX-/TX-Option

Hier können Sie auswählen, welchen Parameter sie ändern möchten.

Der neue Wert muss in HEX angegeben werden.

2.6.11. Puls Generator

Hier können Sie den Pulse Generator testen.

The screenshot shows the DT-ICT-Tool interface. At the top, there's a header with the DEDITEC logo, I.C.T.-Tool Ver. 2.712, and a power button icon. Below the header, there's a navigation bar with 'IO-Test' and a 'Menu' button. The main area is titled 'Puls Generator [0 .. 15]' and includes the text 'Hier können Sie den Pulse Generator testen.' Below this, there are two tables for configuring pulse generators for 16 channels (Kanal 0 to Kanal 15). Each channel has input fields for Low-Time [ns], High-Time [ns], No. of Pulses, and a 'Go' button. The 'Frequency' column is also present but empty. On the left side, there's a tree view showing the module structure: NET-ETH-LC (Module-ID = 32), Modul, Digital, Counter 48, CNT48 [0 .. 15], Puls Generator [0 .. 15], Analog, Temperatur, PWM, Watchdog, CAN, and Intern. The 'Puls Generator [0 .. 15]' item is selected.

	Low-Time [ns]	High-Time [ns]	No. of Pulses	Frequency
Kanal 0	0	0	0	Go
Kanal 1	0	0	0	Go
Kanal 2	0	0	0	Go
Kanal 3	0	0	0	Go
Kanal 4	0	0	0	Go
Kanal 5	0	0	0	Go
Kanal 6	0	0	0	Go
Kanal 7	0	0	0	Go
Kanal 8	0	0	0	Go
Kanal 9	0	0	0	Go
Kanal 10	0	0	0	Go
Kanal 11	0	0	0	Go
Kanal 12	0	0	0	Go
Kanal 13	0	0	0	Go
Kanal 14	0	0	0	Go
Kanal 15	0	0	0	Go

Low-Time[ns], High-Time[ns], No. of Pulses

Die Low-Time, High-Time und die Anzahl der zu generierenden Pulse kann hier eingestellt werden.

2.6.12. PWM Out

Hier können Sie Einstellungen an Ihrem PWM-Modul vornehmen.

The screenshot shows the DT-ICT-Tool interface. The top bar includes the DEDITEC logo, I.C.T.-Tool Ver. 2.712, and a power button. The main window is titled 'IO-Test' and shows a tree view on the left with the following structure:

- NET-ETH-LC (Module-ID = 32)
 - Modul
 - Digital
 - Counter 48
 - Analog
 - Temperatur
 - PWM
 - PWM Out [0 .. 15]**
 - Watchdog
 - CAN
 - Intern

The main content area is titled 'PWM Out [0 .. 15]' and contains the text: 'Hier können Sie Einstellungen an der Frequenz der PWM-Kanäle ändern.' Below this is a table with 16 channels, each with a 'Value' and 'Readback' column. The 'Value' column for Kanal 0 is set to 100%, while the others are 0%.

Kanal	Value	Readback
Kanal 0	100 %	0 %
Kanal 1	0 %	0 %
Kanal 2	0 %	0 %
Kanal 3	0 %	0 %
Kanal 4	0 %	0 %
Kanal 5	0 %	0 %
Kanal 6	0 %	0 %
Kanal 7	0 %	0 %
Kanal 8	0 %	0 %
Kanal 9	0 %	0 %
Kanal 10	0 %	0 %
Kanal 11	0 %	0 %
Kanal 12	0 %	0 %
Kanal 13	0 %	0 %
Kanal 14	0 %	0 %
Kanal 15	0 %	0 %

Below the table, there are settings for 'Frequency' (set to 250Hz) and 'Readback' (set to 2s). There are also buttons for 'Set', 'Readback', 'Activate', and 'Deactivate'.

Value

Hier können Sie das PWM-Verhältnis eines Kanals setzen.

PWM-Verhältnis von 0% bis 100% in 1% Schritten

Readback (Value)

Gibt das PWM-Verhältnis des jeweiligen Kanals wieder.

Frequency

Hier können Sie die PWM-Frequenz des Moduls einstellen

Readback (Frequency)

Gibt die eingestellte PWM-Frequenz wieder.

Set mode for all channels

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

2.6.13. Watchdog Schaltverhalten

Hier können Sie das Schaltverhalten des Watchdog-Sticks testen

The screenshot shows the 'DT-ICT-Tool' window with the 'IO-Test' menu open. The 'Watchdog Parameter' section is active, displaying controls for enabling, disabling, and retriggering the watchdog. It also includes a table for status and counter values, and an option for automatic retriggering.

Watchdog Parameter	
Watchdog enable	Status: 0
Watchdog disable	Watchdog-Timeout-Value: 5000
Watchdog retrigger	Watchdog-Counter: 0
	Relais-Counter: 0
	Watchdog-Retrigger-Counter: 0

Watchdog auto retrigger
☐ enable Intervall

Auto Refresh

Watchdog enable

Aktiviert den Watchdog

Watchdog disable

Deaktiviert den Watchdog

Watchdog retrigger

Setzt den Watchdog-Stick-Status zurück

Watchdog auto retrigger

Setzt den Status des Watchdog Sticks automatisch nach der angegebenen Zeit zurück

2.7. Modul diagnostizieren

Im Bereich Diagnose, können Sie die Funktionalität Ihrer I/Os diagnostizieren. Dabei können Sie Funktionstests laufen lassen und sich die Auswertung Darstellen lassen.

Für die Kabelrückführungstests wird eine zusätzlich Verkabelung benötigt. Diese Kabel können Sie sich selbst konfigurieren, oder direkt über unseren Shop bestellen.

2.7.1. Digitaler Kabelrückführungstest

Bei diesem Test, werden digitale Ausgänge 8-Blockweise geschaltet. Dabei wird die Spannung dann via Rücklesekabel an den jeweiligen 8 digitalen Eingang geschickt und dann ausgelesen.

Wichtig dabei ist, dass die ausgewählten I/Os den der Verkabelten Ein- und Ausgängen entsprechen. **Dieser Test funktioniert nur mit zusätzlichem Testkabel.**

The screenshot shows the DT-ICT-Tool software interface. The top bar includes the DEDITEC logo, the I.C.T.-Tool version (Ver. 2.712), and the DELIB 2.712 (64 Bit) version. The main menu on the left lists 'Diagnose', 'Statistik', 'Kabelrückföhrtest', 'Digital', and 'Zugriffszeiten Tests'. The 'Digital' section is currently selected. The main area displays the 'Digital' test configuration. It includes a header with the title 'Digital' and a description: 'Hier können Sie digitale Ausgangssignale mit Hilfe eines zusätzlichen Rückföhrungskabels auf die digitalen Eingänge zuröcklesen.' Below this, there are three columns of settings: 'DO State', 'DI State', and 'Data write/read'. Each column has a list of channels (CH1 to CH8) with checkboxes for selection. There are also dropdown menus for 'ch area' (1-8) and 'Test mit delay' (0 ms). At the bottom, there are 'Test Start' and 'Test Stop' buttons. The right side of the interface shows a list of test results, including 'Total count', 'Readback error count', 'Kabel-Verify error count', 'Delib error count', 'Delay time', 'Min delay', 'Max delay', and 'Total test duration time'.

DO State	DI State	Data write
CH1 <input type="checkbox"/>	CH1 <input type="checkbox"/>	Data write -
CH2 <input type="checkbox"/>	CH2 <input type="checkbox"/>	Data read -
CH3 <input type="checkbox"/>	CH3 <input type="checkbox"/>	Total count -
CH4 <input type="checkbox"/>	CH4 <input type="checkbox"/>	Readback error count -
CH5 <input type="checkbox"/>	CH5 <input type="checkbox"/>	Kabel-Verify error count -
CH6 <input type="checkbox"/>	CH6 <input type="checkbox"/>	Delib error count -
CH7 <input type="checkbox"/>	CH7 <input type="checkbox"/>	Delay time <input type="text"/>
CH8 <input type="checkbox"/>	CH8 <input type="checkbox"/>	Min delay -
ch area <input type="text" value="1 - 8"/>	ch area <input type="text" value="1 - 8"/>	Max delay -

Test mit delay ms

Total test duration time -

DO-Status

Zeigt den Status des aktuellen Ausganges an. Das Schaltet passiert automatisch.

DI-Status

Zeigt den Status des aktuellen Einganges an. Das Schaltet passiert automatisch.

ch area

Gibt an welche I/Os für den Test verwendet werden sollen.

delay (in ms)

Hier können Sie die Verzögerungszeit einstellen, die zwischen dem Schreibe- und Lesebefehl gewartet werden soll.

Dies wird bei Relais benötigt, die eine längere Schaltzeit haben.

Data write / Data read

Zeigt an welcher Wert geschrieben und zurück gelesen wurde. Dies passiert automatisch.

Kontrollieren Sie bei Abweichungen ggf. Ihre Verkabelung

Total count

Zeigt an wie viele Befehle ausgeführt wurden.

Readback error count

Zeigt an, ob es während dem Test einen Fehler beim direkten zurück lesen des geschriebenen Wertes gab. Dieser Wert ist unabhängig von der Verkabelung.

Kabel-Verify error count

Zeigt an, ob es während dem Test eine Abweichung zwischen dem via Rückführungskabel gesendeten und empfangen Wert gab.

Delib error count

Zeigt an, ob es während dem Test einen Fehler beim Ausführen eines DELIB-Befehls gab.

Delay Time, min, max

Zeigt die Zeit an, die für einen DapiDOSet8, DapiDOReadBack64 und DapiDIGet8 Befehl benötigt wird.

Min/Max zeigt dabei den geringsten bzw. höchsten Wert während der Tests an.

2.7.2. Analoges Kabelrückführungstest

Bei diesem Test werden Spannungen im Bereich von +9/-9 V auf die ausgewählten D/A-Kanäle gegeben und mit Hilfe eines Rückführungskabel an den Entsprechenden A/D-Kanälen ausgelesen.

Wichtig dabei ist, dass die ausgewählten I/Os den der Verkabelten Ein- und Ausgängen entsprechen.

Dieser Test funktioniert nur mit zusätzlichem Testkabel.

Bei diesem Test kann ein zusätzliches A/D Modul als Referenzmodul benutzt werden.

The screenshot shows the DT-ICT-Tool software interface. The top bar includes the DEDITEC logo, the I.C.T.-Tool version 2.712, and a power button. The main menu on the left lists various diagnostic tests, with 'Analog' selected. The 'Analog' test configuration panel is displayed, showing settings for D/A channel (D/A 0), Delay time in ms (0), A/D channel (A/D 0), and A/D filterlevel (10). Below these settings, there are fields for 'D/A to A/D duration', 'Total count', and 'Delib error count', all currently showing '-'. A table of voltage ranges and their corresponding maximum deviations is also present. At the bottom, there are 'Test Start' and 'Test Stop' buttons, and a checkbox for 'Mit dieser Option kann ein 2. A/D Modul für den Readback verwendet werden' (checked), with a dropdown menu showing 'BS-WEU (0)'.

D/A channel	Delay time in ms	A/D channel	A/D filterlevel
D/A 0	0	A/D 0	10

D/A to A/D duration	Total count	Delib error count
-	-	-

Voltage Range	max. deviation
< 10 mV	-
> 10 mV / < 100 mV	-
> 100 mV	-

Total test duration time: -

Mit dieser Option kann ein 2. A/D Modul für den Readback verwendet werden: ☒

BS-WEU (0)

D/A-Kanal

Gibt den D/A-Kanal an, der für den Test verwendet werden soll

Delay time in ms

Gibt die Zeit an, die zwischen einem Schreib- und Lesebefehl gewartet werden soll. Dies wird bei langsameren A/D-Wandlern benötigt.

A/D-Kanal

Gibt den A/D-Kanal an, der für den Test verwendet werden soll

A/D-Filterlevel

Zeigt an, welcher A/D-Filterlevel für diesen Test verwendet werden soll

D/A to A/D duration

Zeigt an, wieviel Zeit für einen DapiDASet und DapiADGet Befehl benötigt wird (inkl. delay Zeit)

Total count

Zeigt an wie viele Befehle ausgeführt wurden.

Delib error count

Zeigt an, ob es während dem Test einen Fehler beim Ausführen eines DELIB-Befehls gab.

[< 10 mV], [>10 mV / < 100 mV], [> 100 mv]

Zeigt an, wie oft der jeweilige Abweichungsbereich aufgetreten ist.

Max. deviation

Zeigt die höchste aufgetretene Abweichung innerhalb eines Abweichungsbereich an.

2.7.3. Zugriffszeitentests

Abhängig von der Modulkonfiguration können Sie unterschiedliche Zugriffszeitentests durchführen. Neben den Registerzugriffen, können auch die Zugriffe auf die digitalen und analogen I/Os Ihres Moduls getestet werden.

Dabei wird die Zeit gemessen, die benötigt wird, um Befehle an das Modul zuzusenden oder um Werte vom Modul abzufragen.

Der Test wird automatisch mit unterschiedlichen Bits oder Kanälen durchgeführt. Je nach verwendeter Schnittstelle kann es zu unterschiedlichen Zeitmessungen kommen.

The screenshot shows the DT-ICT-Tool interface. The top bar includes the DEDITEC logo, the I.C.T.-Tool version (2.712), and the module information (DELIB 2.712 (64 Bit)). The main window is titled 'Diagnose' and shows a tree view on the left with 'Zugriffszeiten Register' selected. The main area displays the test results for the 'Zugriffszeiten Register' test. The results are organized into a table with columns for '1 Bit', '8 Bit', '16 Bit', '32 Bit', '64 Bit', 'RB Err Cnt', and 'DELIB Err'. The table is divided into three sections: 'Read', 'Write', and 'Write + Readback'. Each section shows minimum, maximum, and average values for the specified bit widths. The 'Test with delay' is set to 0 ms, and the 'Total count' is 120. The 'Test runs' are set to 10. The 'Elapsed time' is 00:00:01.

	1 Bit	8 Bit	16 Bit	32 Bit	64 Bit	RB Err Cnt	DELIB Err
Read	min.	0,273ms	0,386ms	0,546ms	0,568ms		
	Ø	0,620ms	0,462ms	0,627ms	0,761ms		0
	max.	2,094ms	0,560ms	0,756ms	1,974ms		
	Access / sec	1613,6	2163,8	1595,7	1313,5		
Write	min.	0,429ms	0,391ms	0,368ms	0,268ms		
	Ø	0,532ms	0,464ms	0,461ms	0,509ms		0
	max.	1,177ms	0,592ms	0,558ms	0,814ms		
	Access / sec	1880,9	2155,1	2170,6	1964,7		
Write + Readback	min.	0,422ms	0,297ms	0,407ms	0,261ms		
	Ø	0,510ms	0,478ms	0,715ms	0,449ms	0	0
	max.	0,660ms	0,573ms	9,224ms	0,588ms		
	Access / sec	1962,3	2093,4	1398,4	2227,7		

Testauswertung

Hier wird die geringste, höchste und die durchschnittliche benötigte Zeit ermittelt. Zusätzlich wird die Zugriffszeit pro Sekunde ermittelt.

Test with delay

Hier können Sie einstellen, mit welcher Verzögerungszeit zwischen den Befehlen der Test ausgeführt werden soll

Total Count

Gibt an, wieviele Befehle ausgeführt wurden

Test runs

Hier können Sie auswählen, wie oft Befehle Pro Bit oder Kanal ausgeführt werden soll

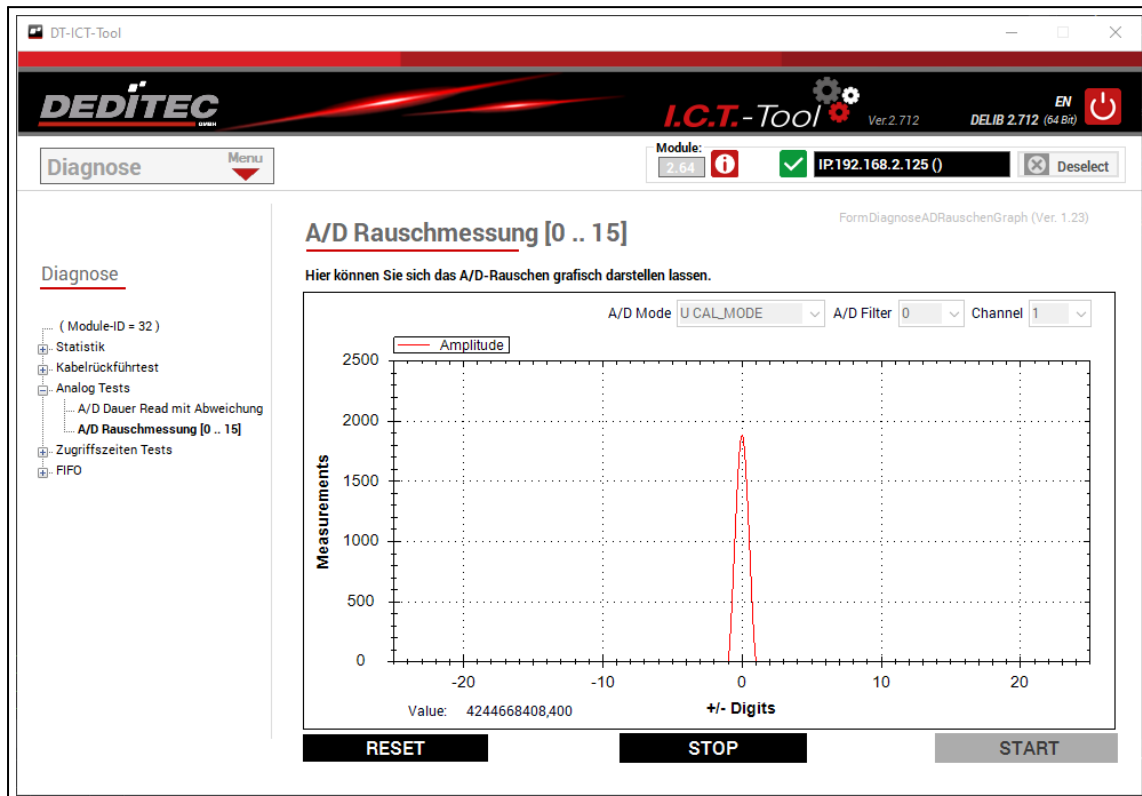
Endless run

Ist dieses Häkchen gesetzt, fängt der Test nach jedem Durchlauf erneut von vorne an.

Die Auswertungen werden dabei gespeichert und angepasst.

2.7.4. A/D-Rauschen Graph

Hier können Sie das Rauschen eines A/D-Kanals grafisch darstellen.



A/D-Mode

Gibt an, welcher A/D-Modus für den Test verwendet werden soll

A/D-Filter

Gibt an, welcher A/D-Filter für den Test verwendet werden soll

Channel

Gibt an, welcher Kanal angezeigt werden soll

Start

Startet die grafische Ausgabe

Stop

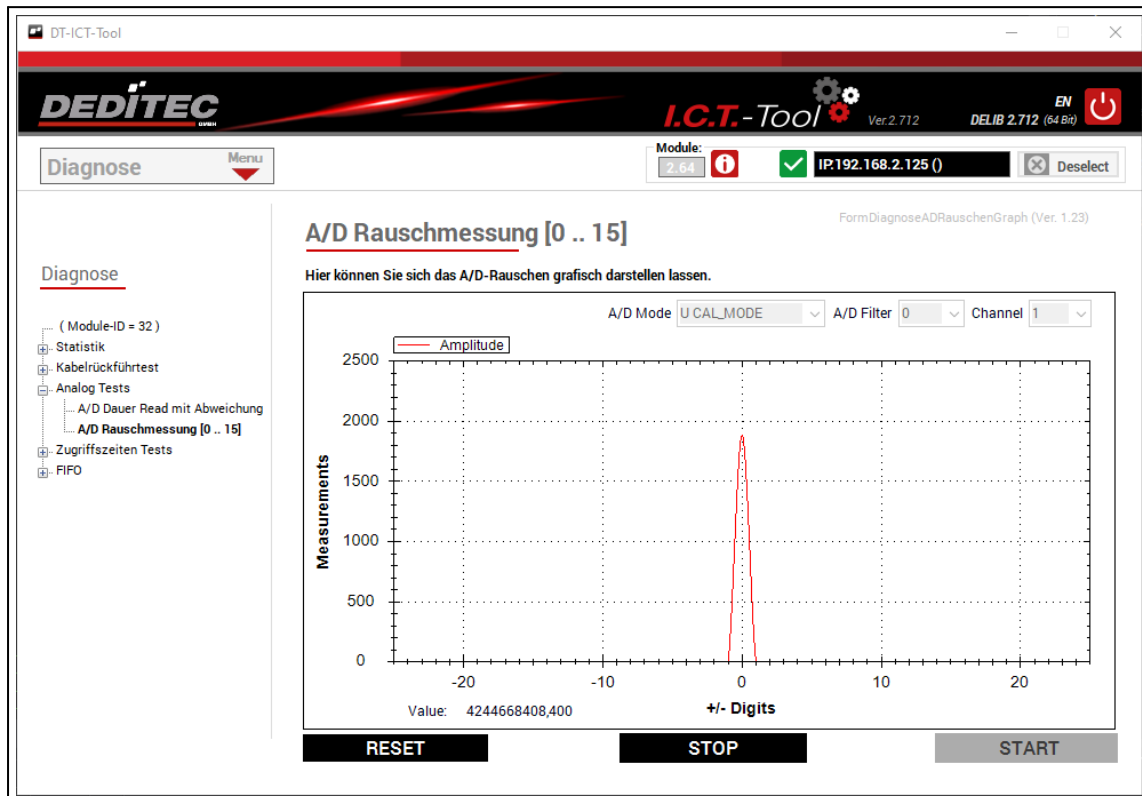
Beendet die grafische Ausgabe

Reset

Setzt die grafische Ausgabe zurück

2.7.5. A/D-Lesen mit Abweichung

Hier können Sie das Rauschen eines A/D-Kanals grafisch darstellen.



A/D-Mode

Gibt an, welcher A/D-Modus für den Test verwendet werden soll

A/D-Filter

Gibt an, welcher A/D-Filter für den Test verwendet werden soll

Channel

Gibt an, welcher Kanal angezeigt werden soll

Start

Startet die grafische Ausgabe

Stop

Beendet die grafische Ausgabe

Reset

Setzt die grafische Ausgabe zurück

2.7.6. FIFO In/Out

Hier können Sie den FIFO-In oder FIFO-Out des Moduls testen

DT-ICT-Tool

DEDITEC ADMIN **I.C.T.-Tool** Ver. 2.754 **DELIB 2.733 (64 Bit)**

Diagnose Menu Module: 2.67 IP192.168.2.149 Deselect

FIFO-In FormDiagnoseFifoInOut (Ver. 1.23)

Hier werden Werte aus dem Fifo des Moduls angefordert

FIFO-Instanz	4	5	6	7
Submodul-Nr.	0	1	2	3
Startkanal	0	0	0	0
Endkanal	0	0	0	0
Frequenz	300	300	300	300
FIFO-Modus	TS_0 TESTCTR32	TS_0 TESTCTR32	TS_0 TESTCTR32	TS_0 TESTCTR32
TODO	TS+HEX VAL	TS+HEX VAL	TS+HEX VAL	TS+HEX VAL
Aktivieren	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FIFO-In Zähler	0	0	0	0
Bytes frei	0	0	0	0
Bytes gelesen	0	0	0	0
Bytes in Modul	0	0	0	0
Submodul Bytes frei	0	0	0	0
Submodul TX-Zähler	0	0	0	0
Submodul RX-Zähler	0	0	0	0
FIFO-Status	0	0	0	0

Start Stop

FIFO-Instanz

Zeigt an, welche Instanz benutzt werden soll (FIFO-Out: 0-3, FIFO-In: 4-7)

Submodul-Nr.

Gibt an, welches Submodul für das FIFO lesen/schreiben benutzt werden soll

Startkanal

Gibt den Startkanal für den FIFO an

Endkanal

Gibt den Endkanal für den FIFO an

Frequenz

Gib an, in welcher Frequenz der FIFO gelesen/geschrieben werden soll

FIFO-Modus

Gibt an welcher Modus für den FIFO benutzt werden soll

TODO

Wie der FIFO gelesen/geschrieben werden soll

2.7.7. Main Loop

Hier werden Informationen zum Main Loop Ihres Moduls angezeigt

The screenshot shows the DEDITEC ADMIN I.C.T.-Tool interface. The top bar includes the DEDITEC logo, 'ADMIN', 'I.C.T.-Tool', 'Ver. 2.754', 'DELIB 2.733 (64 Bit)', and a power button. Below the bar, there's a 'Diagnose' button and a 'Menu' dropdown. The 'Module:' section shows 'BS-WEU (0)' with a 'Deselect' button. The left sidebar lists the navigation menu: 'Diagnose', 'BS-WEU (Module-ID = 42)', 'Statistik', 'Module-Config-Memory', 'Main Loop', 'Haupt-Modul', 'Kabelrückföhrtest', and 'Zugriffszeiten Tests'. The main content area is titled 'Haupt-Modul' and contains the text 'Hier wird die Hauptschleife des Modules überprüf'. Below this is a table with the following data:

Modul-Typ	Hauptmodul	
Durchlauf-Anzahl (min/max)	1144	2765
Durchlauf-Dauer [ms] (min/max)	0 ms	532 ms
Auswertzeit [sek]	14 sek	
Verwendeter Sitzungsspeicher	6792 bytes	

At the bottom right of the table area is a 'WERTE LÖSCHEN' button. At the bottom left, there is an 'Auto Refresh' checkbox.

Modul-Typ

Zeigt an ob es sich um ein Interface/Submodule handelt.

Durchlauf Anzahl (min/max)

Zeigt an, wie oft die Mainloop seit Modulstart durchlaufen wurde.

Durchlauf-Dauer [ms] (min/max)

Zeigt an, wie lang ein Mainloop-Durchlauf dauert.

Auswertungsdauer [sek]

Zeigt an, seit wie vielen Sekunden die Messung läuft.

Verwendeter Sitzungsspeicher

Zeigt an, wie hoch der verwendete Sitzungsspeicher ist.

2.7.8. Module-Config-Memory

Hier werden Informationen zum **Module-Config-Memory** dargestellt

The screenshot shows the DEDITEC ADMIN I.C.T.-Tool interface. The top bar includes the DEDITEC ADMIN logo, the I.C.T.-Tool version (Ver. 2.754), and the DELIB 2.733 (64 Bit) status. The main menu on the left lists 'Diagnose', 'Statistik', 'Speicher Übersicht', 'Main Loop', 'Kabelrückföhrtest', and 'Zugriffszeiten Tests'. The 'Speicher Übersicht' page is active, showing a table of memory usage for the 'UC-CAN (1)' module. The table lists 'Mod_cfg_mem(EEPROM) Verzeichniseinträge' (5), 'Mod_cfg_mem(EEPROM) Freie Bytes' (1360 bytes), and 'Verwendeter Speicherplatz' (5464 bytes). The page also includes a 'Deselect' button and an 'Auto Refresh' checkbox.

Mod_cfg_mem(EEPROM) Verzeichniseinträge	5
Mod_cfg_mem(EEPROM) Freie Bytes	1360 bytes
Verwendeter Speicherplatz	5464 bytes

Mod_cfg_mem(EEPROM) Verzeichniseinträge

Gibt an wie viel Verzeichniseinträge im **Module-Config-Memory** existieren

Mod_cfg_mem(EEPROM) Freie Bytes

Gibt die Anzahl der freien Bytes im **Module-Config-Memory** an

Verwendeter Speicherplatz

Gibt die Anzahl der belegten Bytes im Speicher an

2.8. Firmware-Update

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, kann der im ICT-Tool integrierte Flasher verwendet werden. 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.

Bitte beachten Sie, dass der Flashvorgang nur über eine Schnittstelle ausgeführt werden. Dies kann je nach Produktreihe unterschiedlich sein.

Flashfiles manuell installieren

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

2.8.2. Flash-Files per ICT-Tool aktualisieren

Hier können Sie die aktuelle Firmware für Ihr Modul downloaden.

Unter "Online firmware" können Sie erkennen, wann die Firmware zuletzt auf unserer Homepage aktualisiert wurde.

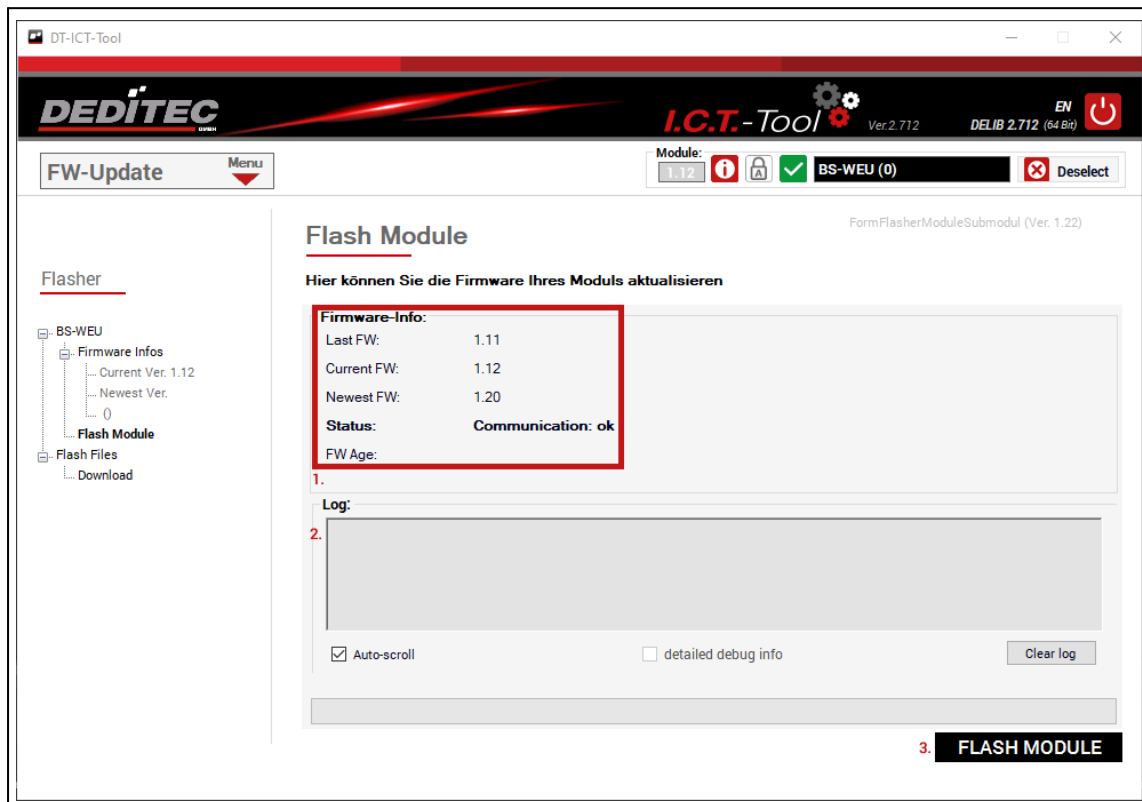
The screenshot shows the 'Firmware-Download' section of the DEDITEC I.C.T.-Tool. The interface includes a sidebar with a tree view under 'Flasher' containing 'BS-WEU', 'Firmware Infos' (with sub-items 'Current Ver. 1.12' and 'Newest Ver. 0'), 'Flash Module', 'Flash Files', and 'Download'. The main content area is titled 'Firmware-Download' and includes the text 'Hier können Sie die Firmware Ihres Moduls von deditec.de downloaden'. Below this is a table with two rows: 'Local firmware' (file not found) and 'Online firmware' (25.01.2024). At the bottom, there is a 'Log:' section with a text area, an 'Auto-scroll' checkbox, a 'Clear log' button, and a large 'DOWNLOAD' button.

Firmware Type	Update Date
Local firmware	file not found
Online firmware	25.01.2024

2.8.3. Firmware Update durchführen

Hier können Sie die Firmware Ihres Interface- und Submoduls aktualisieren.

Um von den neuesten Funktionen zu profitieren und Fehler vorzubeugen, empfehlen wir Ihnen Ihr DEDITEC-Produkt stets Up-to-date zu halten.



1. Hier finden Sie alle nötigen Informationen zu den verschiedenen Firmware Versionen. Unter "Newest FW" wird Ihnen angezeigt, ob es eine neuere Firmware-Version für Ihr Modul gibt.

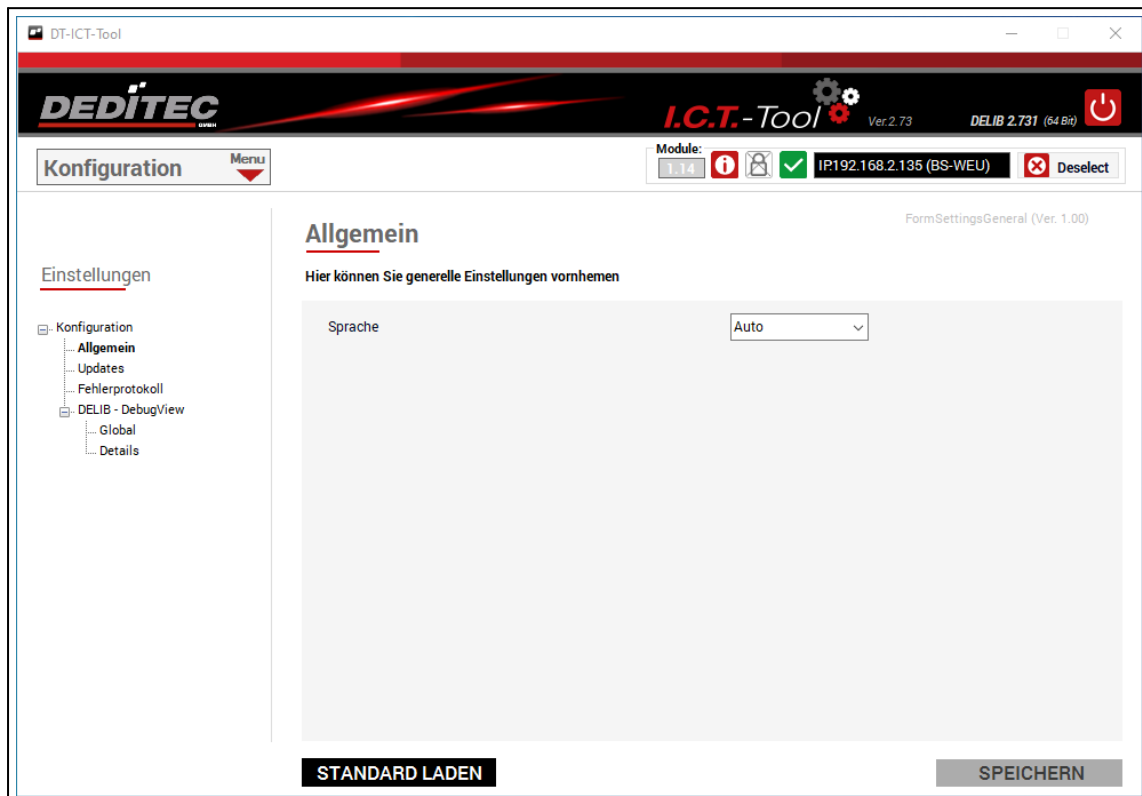
2. Hier werden alle wichtigen Informationen, Fehler- und Statusmeldungen, die während des Flashvorgangs aufgetreten sind, eingetragen.

3. Mit dem "Flash Module" Knopf wird der Flashvorgang gestartet.

2.9. Einstellungen

2.9.1. Allgemein

Hier können Sie allgemeine Einstellungen am ICT-Tool vornehmen.



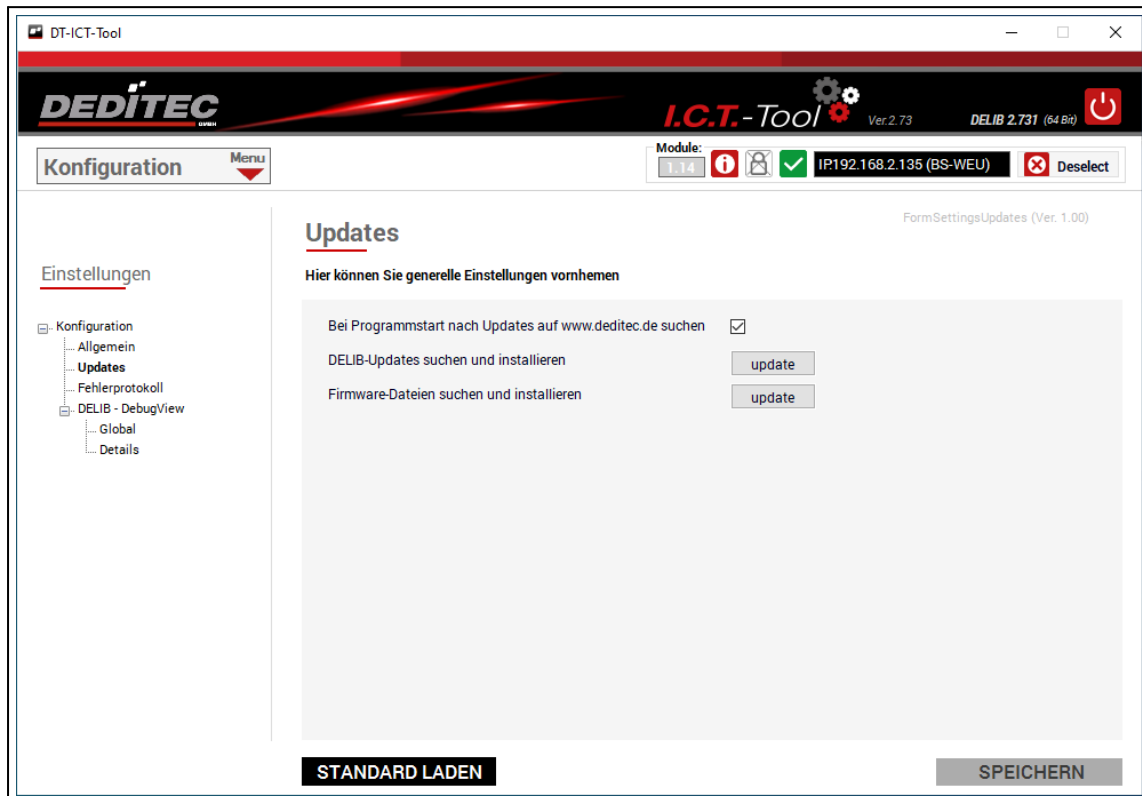
Sprache

Gibt an in welcher Sprache das ICT-Tool dargestellt werden soll.

Aktuell kann nur zwischen den Sprachen Deutsch und Englisch gewählt werden.

2.9.2. Updates

Hier können Sie nach aktuellen DELIB- und Firmware-Updates suchen.



Bei Programmstart nach Updates auf www.deditec.de suchen

Gibt an, ob bei Programmstart nach Updates geprüft werden soll.

Dafür ist eine funktionierende Internetverbindung notwendig.

DELIB-Updates suchen und installieren

Hier können Sie nach aktuellen DELIB-Updates suchen.

Firmware-Dateien suchen und installieren

Hier können Sie nach aktuellen Firmware-Dateien suchen.

2.9.3. Fehlerprotokoll

Hier können Sie Einstellungen am deditec_debug.log vornehmen.

Dieser wird unter dem folgenden Pfad abgespeichert:

C:\Users\?USER?\AppData\Local\DEDITEC

The screenshot shows the 'Fehlerprotokoll' (Error Log) settings page in the DEDITEC I.C.T.-Tool. The interface includes a sidebar with 'Einstellungen' (Settings) and a main area with five settings:

- Schreiben von Debug-Ausgaben in eine debug.log aktivieren ☒
- Log-Benachrichtigung aktivieren ☒
- Automatisches Löschen des debug.log aktivieren ☐
- Löschen des debug.log nach.. Tagen
- Maximale Log Größe mb

Buttons at the bottom: STANDARD LADEN, SPEICHERN.

Schreiben von Debug-Ausgabe in eine debug.log aktivieren

Gibt an, ob Debug-Ausgaben in eine externe .log geschrieben werden sollen.

Log-Benachrichtigung aktivieren

Hier können Sie einstellen, ob Sie benachrichtigt werden wollen, sollte das .log zu groß oder zu alt sein.

Automatisches Löschen des debug.log aktivieren

Hier können Sie einstellen, ob das debug.log automatisch gelöscht werden soll, sollte die Größe oder das Alter der Datei überschritten werden.

Löschen des debug.log nach..

Hier können Sie einstellen nach wie vielen Tagen das Log gelöscht werden soll.

Maximale Log Größe

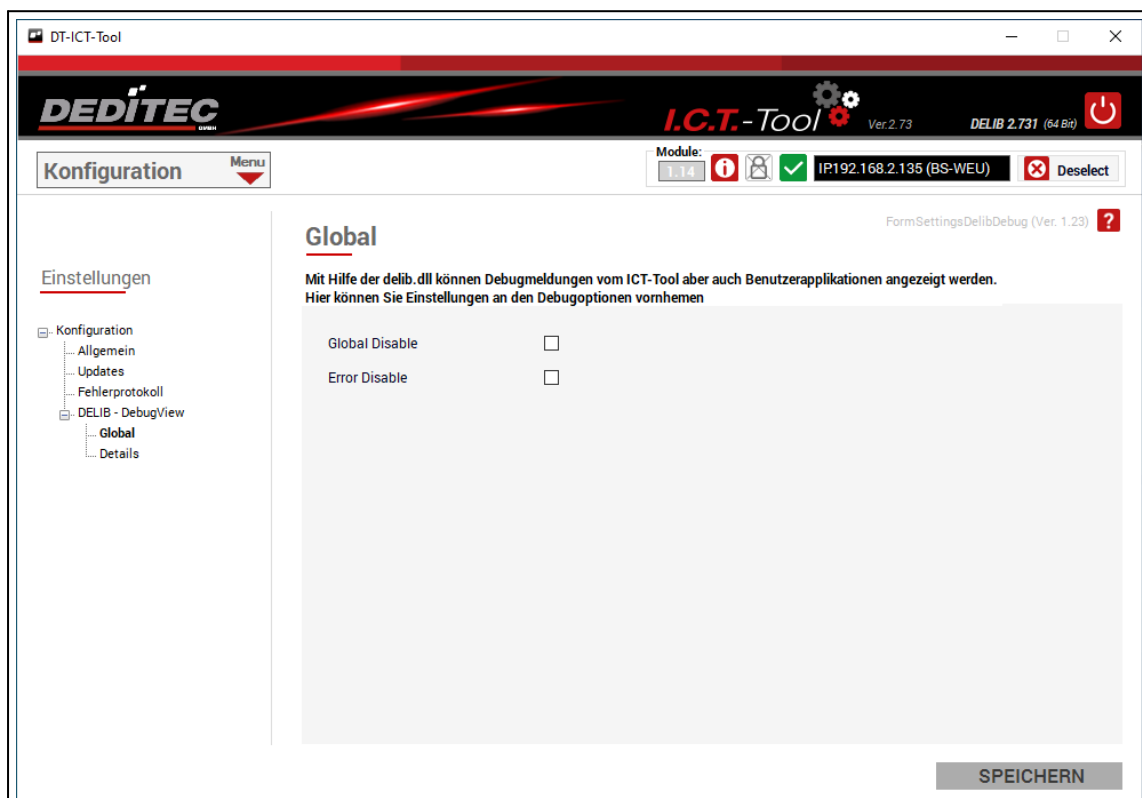
Hier können Sie einstellen wie groß das Log sein darf, bevor es gelöscht wird.

2.9.4. DELIB-DebugView-Global

Hier können Sie die Debug-Ausgabe aktivieren. Zusätzlich können Sie einstellen, ob auftretende Fehler angezeigt werden sollen.

Dies ist besonders in einem Fehlerfall sehr hilfreich, um so das Modul schneller Debuggen zu können.

Die Informationen werden in eine separate .log Datei geschrieben und können außerdem von Überwachungstools wie z.B. DEBUGView ausgegeben werden.



2.9.5. DELIB-DebugView-Details

Hier können Sie einstellen, welche und wie detailliert Informationen aufgezeichnet werden sollen.

Dies ist besonders in einem Fehlerfall sehr hilfreich, um so das Modul schneller Debuggen zu können.

Die Informationen werden in eine separate .log Datei geschrieben und können außerdem von Überwachungstools wie z.B. DEBUGView ausgegeben werden.

The screenshot shows the DEDITEC I.C.T.-Tool interface. The title bar reads "DT-ICT-Tool". The main header features the DEDITEC logo and "I.C.T.-Tool" with version "Ver. 2.73" and "DELIB 2.731 (64 Bit)". A "Konfiguration" menu is on the left. The right side shows a status bar with "Module: 1.14", a green checkmark, and "IP: 192.168.2.135 (BS-WEU)" with a "Deselect" button. The main content area is titled "Details" and contains the text "Hier können Sie einstellen, welche Debugausgaben ein- oder ausgeschaltet werden sollen". Below this is a table with columns "Description", "Normal", and "Detailed".

Description	Normal	Detailed
Open Close	<input checked="" type="checkbox"/>	
Communication	<input checked="" type="checkbox"/>	<input type="checkbox"/>
DAPI Functions	<input checked="" type="checkbox"/>	<input type="checkbox"/>
DAPI Special Commands	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Flash	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Register IO	<input checked="" type="checkbox"/>	
FTD2XX Command	<input checked="" type="checkbox"/>	<input type="checkbox"/>

A "SPEICHERN" button is located at the bottom right of the configuration area.

DELIB API Referenz



3. DELIB API Referenz

3.1. Verfügbare DEDITEC-Modul-IDs

Hier finden Sie eine Auflistung mit allen verfügbaren Modul IDs.

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

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

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

Modul Name	ID
USB_RELAIS_8	14
USB_OPTOIN_8_RELAIS_8	15
USB_OPTOIN_16_RELAIS_16	16
USB_OPTOIN_32	16
USB_RELAIS_32	16
USB_OPTOIN_32_RELAIS_32	17
USB_OPTOIN_64	17
USB_RELAIS_64	17
BS_USB_8	15
BS_USB_16	16
BS_USB_32	17
USB_TTL_32	18
USB_TTL_64	18
RO_ETH_INTERN	19
BS_SER	20
BS_CAN	21
BS_ETH	22
NET_ETH	23
RO_CAN2	24
RO_USB2	25
RO_ETH_LC	26

Modul Name	ID
ETH_RELAIS_8	27
ETH_OPTOIN_8	27
ETH_O4_R4_ADDA	28
ETHERNET_MODULE	29
ETH_TTL_64	30
NET_USB2	31
NET_ETH_LC	32
NET_USB1	33
NET_SER	34
NET_CAN_OPEN	35
NET_RAS_PI	36
USB_CANOPEN_STICK	37
ETH_CUST_0	38
WEU_RELAIS_8	39
WEU_OPTO_8	39
WEU_E_RELAIS_8	40
BS_WEU	41
BS_WEU_E	42
UC_USB	43
UC_CAN	44
BS_USB2	45

Modul Name	ID
CAN_BOX	46

3.2. 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

3.2.1. 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.

3.2.2. Library-Verzeichnis

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

3.2.3. 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.

3.2.4. 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

3.3. Verwaltungsfunktionen

3.3.1. 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;
}
```

3.3.2. 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);
```

3.3.3. 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)
```


3.3.4. 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
```

3.3.5. 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 ICT-Tool 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);
```

3.3.6. 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;  
}
```

3.4. Fehlerbehandlung

3.4.1. 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;
}
```

3.4.2. DapiGetLastErrorText

Beschreibung

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

Definition

*ULONG DapiGetLastErrorText(unsigned char * msg, unsigned long msg_length);*

Parameter

msg = Buffer für den zu empfangenden Text

msg_length = Länge des Text Buffers

Programmierbeispiel

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

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

        DapiClearLastError();

        return TRUE;
    }

    return FALSE;
}
```


3.4.3. DapiClearLastError

Beschreibung

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

Definition

```
void DapiClearLastError();
```

Parameter

Keine

Return-Wert

Keine

Programmierbeispiel

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

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

        DapiClearLastError();

        return TRUE;
    }

    return FALSE;
}
```

3.4.4. DapiGetLastErrorByHandle

Beschreibung

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

Definition

ULONG DapiGetLastErrorByHandle(ULONG handle);

Parameter

handle=Dies ist das Handle eines geöffneten Moduls.

Return-Wert

Fehler Code

0=kein Fehler. (siehe delib_error_codes.h)

Programmierbeispiel

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

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

        DapiClearLastErrorByHandle(handle);

        return TRUE;
    }

    return FALSE;
}
```

3.4.5. DapiClearLastErrorByHandle

Beschreibung

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

Definition

```
void DapiClearLastErrorByHandle();
```

Parameter

handle=Dies ist das Handle eines geöffneten Moduls.

Return-Wert

Keine

Programmierbeispiel

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

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

        DapiClearLastErrorByHandle(handle);

        return TRUE;
    }

    return FALSE;
}
```

3.5. Digitale Eingänge lesen

3.5.1. 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)

3.5.2. 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

3.5.3. 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

3.5.4. 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();
```

3.5.5. 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

3.5.6. 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

3.5.7. 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
```

3.5.8. 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);
```

3.5.9. 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);
```

3.5.10. 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
```

3.5.11. 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.
```

3.5.12. 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
```

3.5.13. 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.
```

3.6. Digitale Ausgänge verwalten

3.6.1. 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

3.6.2. 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

3.6.3. 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

3.6.4. 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();
```

3.6.5. 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

3.6.6. DapiDOSet1_WithTimer

Beschreibung

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

Definition

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

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

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

data=Gibt die Datenwerte an, die geschrieben werden

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

Return-Wert

Keiner

Bemerkung:

Dieser Befehl wird nur von unserem RO-O8-R8 Modul unterstützt.

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

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

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

DELIB Übersichtstabelle entnehmen.

Programmierbeispiel

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

3.6.7. 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.

3.6.8. 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.

3.6.9. 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);
```

3.6.10. 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);
```

3.7. Digitale Zähler Funktionen für RO-Counter Module

3.7.1. 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_X2	1		X
DAPI_CNT48_MODE_COUNT_RISING_EDGE_X4	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
DAPI_CNT48_SUBMODE_RESET_WITH_READ	1	x

Submode	Wert[hex]	HW-Reset verfügbar
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 Eingangsfilter (mit einer oder Verknüpfung) kombiniert werden. Hierzu stehen folgende Eingangsfilter 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).
```

3.7.2. 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
```

3.7.3. 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
```

3.7.4. 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
```


3.7.5. 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
```

3.7.6. 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
```

3.7.7. 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
```

3.7.8. 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
```

3.8. PWM Funktionen

3.8.1. 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)
```

3.8.2. 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
```

3.8.3. 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
```


3.8.4. 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
```

3.9. Pulsgenerator Ausgänge verwalten

3.9.1. 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 ($t[\text{ns}] / 10 - 1$)

par2 = High-Zeit des Impulses ($t[\text{ns}] / 10 - 1$)

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.

3.10. A/D-Wandler Funktionen

3.10.1. DapiADSetMode

Beschreibung

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

Definition

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

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

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

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

Return-Wert

Keiner

Bemerkung

Folgende Modi werden unterstützt:

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

Unipolare Spannungen (gilt nicht für Module der CAN-Box-Serie)

Modus	Wertebereich
DAPI_ADDA_MODE_16BIT_UNIPOL_1 0V	0V .. 10V
DAPI_ADDA_MODE_16BIT_UNIPOL_5 V	0V .. 5V
DAPI_ADDA_MODE_16BIT_UNIPOL_2 V5	0V .. 2,5V

Unipolare Spannungen der CAN-Box-Serie

Modus	Wertebereich
DAPI_ADDA_MODE_16BIT_UNIPOL_4 0V (nur CAN-Box 24V-Version)	0V .. 40V
DAPI_ADDA_MODE_16BIT_UNIPOL_6 0V (nur CAN-Box 48V-Version)	0V .. 60V

Bipolare Spannungen:

Modus	Wertebereich
DAPI_ADDA_MODE_16BIT_BIPOL_10V	-10V .. +10V
DAPI_ADDA_MODE_16BIT_BIPOL_5V	-5V .. +5V
DAPI_ADDA_MODE_16BIT_BIPOL_2V5	-2,5V .. +2,5V

Ströme:

Modus	Wertebereich
DAPI_ADDA_MODE_16BIT_0_20mA	0 .. 20 mA
DAPI_ADDA_MODE_16BIT_4_20mA	4 .. 20 mA
DAPI_ADDA_MODE_16BIT_0_24mA	0 .. 24 mA
DAPI_ADDA_MODE_16BIT_0_50mA	0 .. 50 mA

3.10.2. 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

3.10.3. 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

3.10.4. 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

3.10.5. 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.

3.10.6. 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.
```

3.10.7. DapiSpecialADFilterSet

Erklärung

A/D-Wandler mit höherer Auflösung sind sehr präzise Messinstrumente, die bereits kleinste Änderungen am Messsignal erfassen können. In der Praxis kommt es häufig vor, dass elektrische oder elektromagnetische Störgrößen über die Anschlussleitung auf das Messsignal einwirken und dieses abfälschen. Mit Hilfe eines A/D-Filters lassen sich die Messsignale glätten und optimieren.

Der von uns eingesetzte Filter auf Software-Basis ist bereits im Microcontroller integriert worden, durch den der A/D-Wandler angesteuert wird. Per Softwarebefehl können verschiedene Filterlevel eingestellt und auch zurückgelesen werden.

Das Filterprinzip basiert auf einer Mittelwertbildung aus einer Summe von Messungen. Je höher das Filterlevel gewählt wird, desto mehr Messungen werden zur Mittelwertbildung herangezogen. Bedeutet aber auch, dass sich die Abtastrate des A/D-Wandlers mit steigendem Filterlevel reduziert.

Beschreibung

Mit diesem Befehl lässt sich das A/D-Filterlevel Ihres A/D-Moduls festlegen. Je nach Modul, kann die Höhe des einstellbaren Filterlevels variieren.

Bei einem Modul mit mehreren Submodulen kann das A/D-Filterlevel für jedes Submodul einzeln bestimmt werden.

NET-Interface	Submodule Nr.0	Submodule Nr.1	Submodule Nr.2	Submodule Nr.3	Submodule Nr. ..
---------------	----------------	----------------	----------------	----------------	------------------

Wird als Filterlevel eine 0 übergeben, wird der Filterlevel deaktiviert.

Definition

```
void DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_AD, cmd, par1, par2);
```

Parameter

handle=Dies ist das Handle eines geöffneten Moduls.

cmd=DAPI_SPECIAL_AD_FILTER_SET

par1=((filterlevel & 0xff) << 8) | (submodule_nr & 0xff)

filterlevel: gibt an, welches Filterlevel gesetzt werden soll

*submodule_nr: gibt an, auf welchem Submodule der Filter gesetzt werden soll.
Sollte das Modul keine Submodule besitzen wird der Wert 0xff übergeben*

par2 = 0

Return-Wert

Keiner.

Bemerkung

Bei jedem Neustart des Moduls ist der A/D-Filter zunächst ausgeschaltet und kann bei Bedarf aktiviert werden. Es empfiehlt sich daher, den A/D-Filter in der Initialisierung Ihres Projektes festzulegen.

Programmierbeispiel

```
filterlevel = 10
submodule_nr = 0
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_AD,
DAPI_SPECIAL_AD_FILTER_SET, ((filterlevel & 0xff) << 8)
| (submodule_nr & 0xff) , 0);
// A/D-Filter des Submoduls 0 wird auf 10 gesetzt

filterlevel = 5
submodule_nr = 3
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_AD,
DAPI_SPECIAL_AD_FILTER_SET, ((filterlevel & 0xff) << 8)
| (submodule_nr & 0xff) , 0);
// A/D-Filter des Submoduls 3 wird auf 5 gesetzt

filterlevel = 6
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_AD,
DAPI_SPECIAL_AD_FILTER_SET, ((filterlevel & 0xff) << 8)
| (0xff & 0xff) , 0);
// A/D-Filter des Hauptmoduls wird auf 6 gesetzt (nur bei Modulen ohne
Submodul)

filterlevel = 2
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_AD,
DAPI_SPECIAL_AD_FILTER_SET, ((filterlevel & 0xff) << 8)
| (0xff & 0xff) , 0);
// A/D-Filter des Hauptmoduls wird auf 2 gesetzt (nur bei Modulen ohne
Submodul)
```

3.10.8. DapiSpecialADFilterGet

Beschreibung

Mit diesem Befehl kann die A/D-Filterlevel des A/D-Moduls ausgelesen werden.

Bei einem Modul mit Submodulen muss die Submodulnummer übertragen werden.

Definition

```
void DT.Delib.DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_AD, cmd, par1,  
    par2);
```

Parameter

handle=Dies ist das handle eines offenen Moduls.

cmd=*DAPI_SPECIAL_AD_FILTER_GET*

par1=(submodule_nr & 0xff)

*submodule_nr: gibt das Submodul an, aus dem der Filter gelesen werden soll.
Wenn das Modul keine Submodule hat, wird der Wert 0xff übertragen*

par2 = 0

Return-Wert

A/D-Filterlevel

Programmierbeispiel

```
unsigned long filterlevel = DapiSpecialCommand(handle,  
DAPI_SPECIAL_CMD_AD, DAPI_SPECIAL_AD_FILTER_GET, (0 &  
0xff) , 0);  
// A/D-Filter des Submoduls 0 wird ausgelesen  
  
unsigned long filterlevel = DapiSpecialCommand(handle,  
DAPI_SPECIAL_CMD_AD, DAPI_SPECIAL_AD_FILTER_GET, (5 &  
0xff) , 0);  
// A/D-Filter des Submoduls 5 wird ausgelesen  
  
unsigned long filterlevel = DapiSpecialCommand(handle,  
DAPI_SPECIAL_CMD_AD, DAPI_SPECIAL_AD_FILTER_GET, (0xff  
& 0xff) , 0);  
// A/D-Filter des Hauptmoduls wird ausgelesen (nur bei Modulen ohne  
Submodul)
```

3.11. D/A-Ausgänge verwalten

3.11.1. 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

3.11.2. 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

3.11.3. 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)
```

3.11.4. 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
```

3.11.5. 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.

3.11.6. 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.
```

3.12. PT100 Funktionen

3.12.1. 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
```


3.13. TTL-Ein-/Ausgangs Richtungen setzen mit DapiSpecialCommand

3.13.1. DAPI_SPECIAL_CMD_SET_DIR_DX_1

Beschreibung

Dieser Befehl setzt die Richtung von 8 hintereinanderliegenden TTL-Ein/Ausgängen (1-Bit weise).

Definition

```
void DapiSpecialCommand(ULONG handle, DAPI_SPECIAL_CMD_SET_DIR_DX_1,  
    ULONG ch, ULONG dir, 0);
```

Parameter

handle = Dies ist das Handle eines geöffneten Moduls

ch = Muss immer 0 sein!

dir = Gibt die Richtung für 8 Kanäle an (1=output / 0=input) / Bit 0 steht für Kanal 0, Bit 1 für Kanal 1 ...

Bemerkung

Nicht kompatibel mit USB-TTL-32/64.

Verwenden Sie für diese Module den DAPI_SPECIAL_CMD_SET_DIR_DX_8 Befehl.

Programmierbeispiel

```
DapiSpecialCommand(handle,  
    DAPI_SPECIAL_CMD_SET_DIR_DX_1, 0, 0x01 , 0);  
// Set Dir of TTL-I/O CH0 to output, others to input  
DapiSpecialCommand(handle,  
    DAPI_SPECIAL_CMD_SET_DIR_DX_1, 0, 0x02 , 0);  
// Set Dir of TTL-I/O CH1 to output, others to input  
DapiSpecialCommand(handle,  
    DAPI_SPECIAL_CMD_SET_DIR_DX_1, 0, 0x04 , 0);  
// Set Dir of TTL-I/O CH2 to output, others to input  
DapiSpecialCommand(handle,  
    DAPI_SPECIAL_CMD_SET_DIR_DX_1, 0, 0x08 , 0);  
// Set Dir of TTL-I/O CH3 to output, others to input  
DapiSpecialCommand(handle,  
    DAPI_SPECIAL_CMD_SET_DIR_DX_1, 0, 0x10 , 0);  
// Set Dir of TTL-I/O CH4 to output, others to input
```

```
DapiSpecialCommand(handle,  
DAPI_SPECIAL_CMD_SET_DIR_DX_1, 0, 0x20 , 0);  
// Set Dir of TTL-I/O CH5 to output, others to input  
DapiSpecialCommand(handle,  
DAPI_SPECIAL_CMD_SET_DIR_DX_1, 0, 0x40 , 0);  
// Set Dir of TTL-I/O CH6 to output, others to input  
DapiSpecialCommand(handle,  
DAPI_SPECIAL_CMD_SET_DIR_DX_1, 0, 0x80 , 0);  
// Set Dir of TTL-I/O CH7 to output, others to input  
  
DapiSpecialCommand(handle,  
DAPI_SPECIAL_CMD_SET_DIR_DX_1, 0, 0x0f , 0);  
// Set Dir of TTL-I/O CH0-3 to output, others to input  
DapiSpecialCommand(handle,  
DAPI_SPECIAL_CMD_SET_DIR_DX_1, 0, 0xff , 0);  
// Set Dir of TTL-I/O CH0-7 to output, others to input
```

3.13.2. DAPI_SPECIAL_CMD_SET_DIR_DX_8

Beschreibung

Dieser Befehl setzt die Richtung von bis zu 64 hintereinanderliegenden TTL-Ein/Ausgängen (8-Bit weise).

1-Bit repräsentiert dabei 8 TTL-Ein/Ausgänge.

Definition

```
void DapiSpecialCommand(ULONG handle,  
DAPI_SPECIAL_CMD_SET_DIR_DX_8, ULONG ch, ULONG dir, 0);
```

Parameter

handle = Dies ist das Handle eines geöffneten Moduls

ch = Muss immer 0 sein!

dir = (8-Bit) gibt die Richtung für bis zu 64 hintereinanderliegende TTL-Ein/Ausgänge an. (1=output / 0=input)

Bemerkung

Nur kompatibel mit USB-TTL-32/64.

Verwenden Sie für andere TTL-Produkte den DAPI_SPECIAL_CMD_SET_DIR_DX_1 Befehl.

Programmierbeispiel

```
DapiSpecialCommand(handle,  
DAPI_SPECIAL_CMD_SET_DIR_DX_8, 0, 0x1 , 0);  
// Set Dir of TTL-I/O CH0-7 to output, others to input  
DapiSpecialCommand(handle,  
DAPI_SPECIAL_CMD_SET_DIR_DX_8, 0, 0x3 , 0);  
// Set Dir of TTL-I/O CH0-15 to output, others to input  
DapiSpecialCommand(handle,  
DAPI_SPECIAL_CMD_SET_DIR_DX_8, 0, 0xc , 0);  
// Set Dir of TTL-I/O CH16-31 to output, others to input  
DapiSpecialCommand(handle,  
DAPI_SPECIAL_CMD_SET_DIR_DX_8, 0, 0x33 , 0);  
// Set Dir of TTL-I/O CH0-15 and CH32-47 to output, others to input  
DapiSpecialCommand(handle,  
DAPI_SPECIAL_CMD_SET_DIR_DX_8, 0, 0xff , 0);  
// Set Dir of TTL-I/O CH0-63 to output, others to input
```

3.13.3. DAPI_SPECIAL_CMD_GET_DIR_DX_8

Beschreibung

Dieser Befehl liest die Richtung von bis zu 64 hintereinanderliegenden TTL-Ein/Ausgängen (8-Bit weise).

Definition

ULONG DapiSpecialCommand(ULONG handle,
DAPI_SPECIAL_CMD_GET_DIR_DX_8, ULONG ch, ULONG dir, 0);

Parameter

handle = Dies ist das Handle eines geöffneten Moduls

ch = Muss immer 0 sein!

dir = Muss immer 0 sein!

Bemerkung

Nur kompatibel mit USB-TTL-32/64.

Return-Wert

Richtungszustand von 64 Kanälen.

Bit 0: Richtung von TTL 0-7	/ 1=Output, 0=Input
Bit 1: Richtung von TTL 8-15	/ 1=Output, 0=Input
Bit 2: Richtung von TTL 16-23	/ 1=Output, 0=Input
Bit 3: Richtung von TTL 24-31	/ 1=Output, 0=Input
Bit 4: Richtung von TTL 32-39	/ 1=Output, 0=Input
Bit 5: Richtung von TTL 40-47	/ 1=Output, 0=Input
Bit 6: Richtung von TTL 48-55	/ 1=Output, 0=Input
Bit 7: Richtung von TTL 56-63	/ 1=Output, 0=Input

Programmierbeispiel

```
ULONG ret = DapiSpecialCommand(handle,  
DAPI_SPECIAL_CMD_GET_DIR_DX_8, 0, 0, 0);  
// Liest die Richtung von 64 Kanälen aus
```

3.14. Watchdog Funktionen

3.14.1. DapiWatchdogEnable

Beschreibung

Diese Funktion aktiviert den Watchdog.

Definition

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

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

Return-Wert

Keiner

Programmierbeispiel

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

3.14.2. DapiWatchdogDisable

Beschreibung

Diese Funktion deaktiviert den Watchdog.

Definition

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

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

Return-Wert

Keiner

Programmierbeispiel

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

3.14.3. DapiWatchdogRetrigger

Beschreibung

Diese Funktion retriggert den Watchdog-Timer.

Definition

```
void DapiWatchdogRetrigger(ULONG handle);
```

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

Return-Wert

Keiner

Programmierbeispiel

```
DapiWatchdogRetrigger(handle) ;  
//Retriggert den Watchdog-Timer
```

3.15. Schrittmotoren Funktionen

3.15.1. Befehle mit DapiStepperCommand

3.15.1.1. DAPI_STEPPER_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_STEPPER_CMD_GO_POSITION, position, 0, 0, 0);

Programmierbeispiel

```
DapiStepperCommand(handle, motor,  
DAPI_STEPPER_CMD_GO_POSITION, go_pos_par, 0, 0, 0);
```


3.15.1.2. 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.
```

3.15.1.3. DAPI_STEPPEER_CMD_SET_POSITION

Beschreibung

Dieses Kommando dient zum setzen der Motorposition. Die Auflösung beträgt 1/16 Vollschriff. Dieses Kommando darf nur bei angehaltenem Motor verwendet werden.

Definition

```
DapiStepperCommand(handle, motor, DAPI_STEPPEER_CMD_SET_POSITION, par1,  
0, 0, 0);
```

Parameter

par1 = Motorposition

3.15.1.4. 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]

3.15.1.5. 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]

3.15.1.6. 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. Schritverluste 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]

3.15.1.7. 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);

3.15.1.8. 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);
```

3.15.1.9. DAPI_STEPPEER_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_STEPPEER_CMD_DISABLE, par1, 0, 0, 0);
```

Parameter

par1 = Disablemode (0=Normale Funktion / 1=Disable)

3.15.1.10. 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 Endschalter1)

par2=3 = (ENDSW2 - LED leuchtet bei geschlossenen Endschalter2)

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 Enschanter 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 Referenzschalterschalter1

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_MOTORCHAR_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_AFTERENDSWITCH, 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_TOOFFSET, 300, 0, 0);
```

// Einstellung der Geschwindigkeit, mit der zum optionalen Offset angefahren wird.

3.15.1.11. 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 Endschalter1)

return=3 (ENDSW2 - LED leuchtet bei geschlossenen Endschalter2)

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.

3.15.1.12. DAPI_STEPPER_CMD_MOTORCHARACTERISTIC_EEPROM_SAVE

Beschreibung

Es wird die aktuelle Motorcharakteristik des Motors ins EEPROM abgespeichert.

Definition

```
DapiStepperCommand(handle, motor,  
DAPI_STEPPER_CMD_MOTORCHARACTERISTIC_EEPROM_SAVE, 0, 0, 0, 0);
```

3.15.1.13. 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);
```

3.15.1.14. 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 Endschaltes1: nicht invertiert
- Funktion des Endschaltes2: 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

3.15.1.15. 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);
```

3.15.1.16. 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]

3.15.1.17. 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]

3.15.2. Status abfragen mit DapiStepperGetStatus

3.15.2.1. 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);
```


3.15.2.2. 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);
```

3.15.2.3. 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);
```

3.15.3. 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.

3.16. CAN Runtime Funktionen

3.16.1. RunTimeVarWriteToModule

Beschreibung

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

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

Parameter

handle = Dies ist das Handle eines geöffneten Moduls

par = Laufzeitenvariable, die beschrieben oder ausgelesen werden soll

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

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

Bemerkung

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

Definition

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

Der darin befindliche Quellcode lautet wie folgt:

//Lesen der Werte

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

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

//Schreiben der Werte

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

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

par = DAPI_SPECIAL_CMDEXT_CAN_RUNTIME_DEV_BAUDRATE

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

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

Programmierbeispiel

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

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

par = DAPI_SPECIAL_CMDEXT_CAN_RUNTIME_DEV_USEEXTID

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

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

Programmierbeispiel

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

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

par = DAPI_SPECIAL_CMDEXT_CAN_RUNTIME_TX_IS_ACTIVE

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

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

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

Programmierbeispiel

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

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


par = DAPI_SPECIAL_CMDEXT_CAN_RUNTIME_TX_INTERVAL

Mit diesem Befehl kann das Interval eingestellt/ausgelesen werden.

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

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

Beispiel

Ein Interval von 700ms entspricht einem value von 0x73

Ein Interval von 40ms entspricht einem value von 0x42

Programmierbeispiel

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

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

par = DAPI_SPECIAL_CMDEXT_CAN_RUNTIME_TX_USE_EXT_ID

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

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

Programmierbeispiel

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

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

par = DAPI_SPECIAL_CMDEXT_CAN_RUNTIME_TX_CANID

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

Programmierbeispiel

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

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

par = DAPI_SPECIAL_CMDEXT_CAN_RUNTIME_TX_MODE

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

TX-Mode	Value
OPTO-IN 1-64	0x01
OPTO-IN 65-128	0x24
OPTO-IN 129-192	0x25
OPTO-IN 193-256	0x26
A/D CH 1-4 (16 Bit)	0x02
A/D CH 5-8 (16 Bit)",	0x03
A/D CH 9-12 (16 Bit)	0x04
A/D CH 13-16 (16 Bit)	0x05
A/D CH 17-20 (16 Bit)	0x06
A/D CH 21-24 (16 Bit)	0x07
A/D CH 25-28 (16 Bit)	0x08
A/D CH 29-32 (16 Bit)	0x09
Counter16 1-4 (16 Bit)	0x0a
Counter16 5-8 (16 Bit)	0x0b
Counter16 9-12 (16 Bit)	0x0c
Counter16 13-16 (16 Bit)	0x0d
Counter16 17-20 (16 Bit)	0x0e
Counter16 21-24 (16 Bit)	0x0f
Counter16 25-28 (16 Bit)	0x10
Counter16 29-32 (16 Bit)	0x11

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

Programmierbeispiel

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

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

par = DAPI_SPECIAL_CMDEXT_CAN_RUNTIME_RX_IS_ACTIVE

Mit diesem Befehl wird das RX-Paket aktiviert/deaktiviert

Trigger Mode	Value
OFF	0x00
ON	0x01

Programmierbeispiel

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

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

par = DAPI_SPECIAL_CMDEXT_CAN_RUNTIME_RX_USE_EXT_ID

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

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

Programmierbeispiel

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

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

par = DAPI_SPECIAL_CMDEXT_CAN_RUNTIME_RX_CANID

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

Programmierbeispiel

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

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


par = DAPI_SPECIAL_CMDEXT_CAN_RUNTIME_RX_MODE

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

RX-Mode	Value
Digital Out 1-64	0x01
D/A CH 1-4	0x02
D/A CH 5-8	0x03
D/A CH 9-12	0x04
D/A CH 13-16	0x05
D/A CH 17-20	0x06
D/A CH 21-24	0x07
D/A CH 25-28	0x08
D/A CH 29-32	0x09
Stepper No. 1	0x0a
Stepper No. 2	0x0b
Stepper No. 3	0x0c
Stepper No. 4	0x0d
Stepper No. 5	0x0e
Stepper No. 6	0x0f
Stepper No. 7	0x10
Stepper No. 8	0x11
D/A CH 1-4 (custom)	0x12
D/A CH 5-8 (custom)	0x13
D/A CH 9-12 (custom)	0x14

RX-Mode	Value
D/A CH 13-16 (custom)	0x15
D/A CH 17-20 (custom)	0x16
D/A CH 21-24 (custom)	0x17
D/A CH 25-28 (custom)	0x18
D/A CH 29-32 (custom)	0x19
Trigger Auto TX 1	0x1a
Trigger Auto TX 2	0x1b
Trigger Auto TX 3	0x1c
Trigger Auto TX 4	0x1d
Custom1	0x1e
Custom2	0x1f
Custom3	0x20
PWM CH 1-8	0x21
PWM CH 9-16	0x22
PWM CH 17-24	0x23
PWM CH 25-32	0x24
PWM CH 33-40	0x25
PWM CH 41-48	0x26
PWM CH 49-56	0x27
PWM CH 57-64	0x28
Trigger Auto TX 5	0x29

RX-Mode	Value
Trigger Auto TX 6	0x2a
Trigger Auto TX 7	0x2b
Trigger Auto TX 8	0x2c

Programmierbeispiel

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

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

3.17. Software FIFO verwalten

3.17.1. DapiSpecialCMDSWFifo

Beschreibung

Dieser Befehl dient zum Einstellen des Software FIFO.

Es wird unterschieden zwischen FIFO-OUT (**fifo_instance 0 - 3**) für D/A- und DO-Anwendungen und FIFO-IN (**fifo_instance 4 - 7**) für A/D- und DI-Anwendungen.

Weitere Informationen zu den einzelnen Befehlen, finden Sie in den folgenden Kapiteln und unserem Fifo In/Out Programmierbeispiel.

Definition

```
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_SW_FIFO, cmd, fifo_instance, par2);
```

Parameter

handle = Dies ist das Handle eines geöffneten Moduls

cmd = auszuführende Funktion

fifo_instance = Gibt die Instanz des Software FIFO an [0..7]

par2 = Wert, der an die Funktion übergeben wird

Bemerkung

Definieren Sie als erstes immer das Submodul mit dem DapiSpecialSWFifoSetSubmodule-Befehl!

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

3.17.1.1. DapiSpecialSWFifoInitAndClear

Beschreibung

Dieser Befehl löscht vorhandene Daten aus dem Software FIFO-Speicher und bringt den FIFO-Mechanismus in den Ausgangszustand zurück.

Definition

```
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_SW_FIFO, cmd, fifo_instance, 0);
```

Parameter

cmd = DAPI_SPECIAL_SW_FIFO_INIT_AND_CLEAR

fifo_instance = Gibt die Instanz des Software FIFO an

Wichtig

Das Interface und die Submodule haben separate Software-Fifos, deshalb muss der DAPI_SPECIAL_SW_FIFO_INIT_AND_CLEAR Befehl auf beides angewendet werden.

Programmierbeispiel

```
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_SW_FIFO,  
DAPI_SPECIAL_SW_FIFO_INIT_AND_CLEAR, fifo_instance, 0);  
//Vorhandene Daten werden aus dem FIFO-Speicher gelöscht.
```

3.17.1.2. DapiSpecialSWFifoSetIOModule

Beschreibung

Dieser Befehl gibt an, an welches NET-I/O-Modul die Daten des Software FIFO übergeben werden.

Die Submodulreihenfolge die an das Interface-Module angeschlossen ist, bestimmt welche Submodulnummer Sie angeben müssen.

NET-Interface	Submodule Nr. 0	Submodule Nr. 1	Submodule Nr. 2	Submodule Nr. 3	Submodule Nr. ...
---------------	-----------------	-----------------	-----------------	-----------------	-------------------

Definition

DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_SW_FIFO, cmd, fifo_instance, par2);

Parameter

cmd = DAPI_SPECIAL_SW_FIFO_SET_IOMODULE

fifo_instance = Gibt die Instanz des Software FIFO an

par2 = gibt die Nummer des I/O-Moduls an (0, 1, 2, 3, ...)

Programmierbeispiel

```
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_SW_FIFO,  
DAPI_SPECIAL_SW_FIFO_SET_IOMODULE, fifo_instance, 2);  
//Der Software FIFO überträgt die Daten an NET-I/O-Moduls 2.
```

3.17.1.3. DapiSpecialSWFifoGetIOModule

Beschreibung

Dieser Befehl gibt die Nummer des NET-I/O-Moduls, auf welches die Daten übertragen werden, zurück.

Definition

```
ULONG DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_SW_FIFO, cmd,  
    fifo_instance, 0);
```

Parameter

cmd = DAPI_SPECIAL_SW_FIFO_GET_IOMODULE

fifo_instance = Gibt die Instanz des Software FIFO an

Programmierbeispiel

```
unsigned long ret = DapiSpecialCommand(handle,  
    DAPI_SPECIAL_CMD_SW_FIFO,  
    DAPI_SPECIAL_SW_FIFO_GET_IOMODULE, fifo_instance, 0);  
printf("Submodule = %lu\n", ret);  
//Die Nummer des NET-I/O-Moduls wird ausgelesen und dargestellt.
```

3.17.1.4. DapiSpecialSWFifoIOModActivate

Beschreibung

Dieser Befehl aktiviert die FIFO-Datenübertragung innerhalb der NET-Module.

Definition

```
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_SW_FIFO, cmd, fifo_instance, 0);
```

Parameter

cmd = DAPI_SPECIAL_SW_FIFO_IOMOD_ACTIVATE

fifo_instance = Gibt die Instanz des Software FIFO an

Programmierbeispiel

```
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_SW_FIFO,  
DAPI_SPECIAL_SW_FIFO_IOMOD_ACTIVATE, fifo_instance, 0);  
//Das automatische Ausgeben der Fifo-Datenübertragung wird aktiviert.
```

3.17.1.5. DapiSpecialSWFifoIOModDeactivate

Beschreibung

Dieser Befehl deaktiviert die FIFO-Datenübertragung innerhalb der NET-Module.

Definition

```
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_SW_FIFO, cmd, fifo_instance, 0);
```

Parameter

cmd = DAPI_SPECIAL_SW_FIFO_IOMOD_DEACTIVATE

fifo_instance = Gibt die Instanz des Software FIFO an

Programmierbeispiel

```
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_SW_FIFO,  
DAPI_SPECIAL_SW_FIFO_IOMOD_DEACTIVATE, fifo_instance,  
0);  
//Das automatische Ausgeben der Fifo-Datenübertragung wird deaktiviert.
```


3.17.1.6. DapiSpecialSWFifoIOModSetChannel

Beschreibung

Dieser Befehl gibt unter Angabe von Start- und Endkanal an, in welche Kanäle die Daten des FIFO übertragen werden sollen.

Definition

DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_SW_FIFO, cmd, fifo_instance, ch);

Parameter

cmd = DAPI_SPECIAL_SW_FIFO_IOMOD_SET_CHANNEL

fifo_instance = Gibt die Instanz des Software FIFO an

ch = Angabe des Start- und Endkanals

Programmierbeispiel

```
unsigned long ch_start = 0; //Start with Channel 0
unsigned long ch_end = 1; //End with Channel 1

DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_SW_FIFO,
DAPI_SPECIAL_SW_FIFO_IOMOD_SET_CHANNEL, fifo_instance,
((ch_end << 8) & 0xff00) | (ch_start & 0xff);
//Der Start- und Endkanal wird festgelegt
```

3.17.1.7. DapiSpecialSWFifoIOModGetChannel

Beschreibung

Dieser Befehl zeigt die Kanäle, in welche die Daten übertragen werden.

Definition

```
ULONG DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_SW_FIFO, cmd,  
    fifo_instance, 0);
```

Parameter

cmd = DAPI_SPECIAL_SW_FIFO_IOMOD_GET_CHANNEL

fifo_instance = Gibt die Instanz des Software FIFO an

Return-Wert

Nummer der Kanäle

Bit 0-7 Startkanal

Bit 8-15 Endkanal

Programmierbeispiel

```
unsigned long ret = DapiSpecialCommand(handle,  
    DAPI_SPECIAL_CMD_SW_FIFO,  
    DAPI_SPECIAL_SW_FIFO_IOMOD_GET_CHANNEL, fifo_instance,  
    0);  
printf("Channel = %lu\n", ret);  
//Zeigt auf in welche Submodulkanäle die Daten  
übertragen werden.
```

3.17.1.8. DapiSpecialSWFifoIOModSetFrequencyHz

Beschreibung

Dieser Befehl gibt an, mit welcher Frequenz (in Hertz) ..

.. bei Eingabe gelesen wird.

.. bei Ausgabe geschrieben wird.

Definition

DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_SW_FIFO, cmd, fifo_instance, par2);

Parameter

cmd = DAPI_SPECIAL_SW_FIFO_IOMOD_SET_FREQUENCY_HZ

fifo_instance = Gibt die Instanz des Software FIFO an

par2 = Frequenz in Hertz (Hz)

Bemerkung

Zulässiger Wertebereich: min 1Hz, max. abhängig vom verwendeten Modul

Programmierbeispiel

```
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_SW_FIFO,  
DAPI_SPECIAL_SW_FIFO_IOMOD_SET_FREQUENCY_HZ,  
fifo_instance, 10);  
//Setzt das Frequenzintervall auf 10Hz.
```

3.17.1.9. DapiSpecialSWFifoIOModGetFrequencyHz

Beschreibung

Dieser Befehl gibt die eingestellte Frequenz des NET-I/O-Moduls in Hertz wieder.

Definition

```
ULONG DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_SW_FIFO, cmd,
fifo_instance, 0);
```

Parameter

cmd = DAPI_SPECIAL_SW_FIFO_IOMOD_GET_FREQUENCY_HZ

fifo_instance = Gibt die Instanz des Software FIFO an

Return-Wert

Frequenz in Hertz (Hz)

Programmierbeispiel

```
unsigned long ret = DapiSpecialCommand(handle,
DAPI_SPECIAL_CMD_SW_FIFO,
DAPI_SPECIAL_SW_FIFO_IOMOD_GET_FREQUENCY_HZ,
fifo_instance, 0);
printf("Frequency = %lu (Hz)\n", ret);
//Zeigt das vorher eingestellte Frequenz des NET-I/O-Moduls an.
```

3.17.1.10. DapiSpecialSWFifoIOModSetMode

Beschreibung

Dieser Befehl setzt den Software FIFO-Mode.

Definition

DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_SW_FIFO, cmd, fifo_instance, par2);

Parameter

cmd = DAPI_SPECIAL_SW_FIFO_IOMOD_SET_MODE

fifo_instance = Gibt die Instanz des Software FIFO an

par2 = Software FIFO Mode	Wert(hex)
DAPI_SPECIAL_SW_FIFO_MODE_IN_AD16	0x40
DAPI_SPECIAL_SW_FIFO_MODE_IN_AD16_TS	0xc0
DAPI_SPECIAL_SW_FIFO_MODE_IN_AD18	0x41
DAPI_SPECIAL_SW_FIFO_MODE_IN_AD18_TS	0xc1
DAPI_SPECIAL_SW_FIFO_MODE_IN_DI8	0x60
DAPI_SPECIAL_SW_FIFO_MODE_IN_DI8_TS	0xe0
DAPI_SPECIAL_SW_FIFO_MODE_IN_DI16	0x61
DAPI_SPECIAL_SW_FIFO_MODE_IN_DI16_TS	0xe1

Bemerkung

Der Software FIFO-Mode kann sowohl mit als auch ohne Timestamp (TS) eingestellt werden.

Programmierbeispiel

```
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_SW_FIFO,  
DAPI_SPECIAL_SW_FIFO_IOMOD_SET_MODE, fifo_instance, 0);  
//Der Software FIFO-Mode wird gesetzt.
```

3.17.1.11. DapiSpecialSWFifoIOModGetMode

Beschreibung

Dieser Befehl gibt den vorher eingestellten FIFO-Mode wieder. Aktuell wird dieser in der Firmware noch nicht unterstützt.

Definition

```
ULONG DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_SW_FIFO, cmd,  
fifo_instance, 0);
```

Parameter

cmd = DAPI_SPECIAL_SW_FIFO_IOMOD_GET_MODE

fifo_instance = Gibt die Instanz des Software FIFO an

Return-Wert

FIFO Software Mode

Programmierbeispiel

```
unsigned long ret = DapiSpecialCommand(handle,  
DAPI_SPECIAL_CMD_SW_FIFO,  
DAPI_SPECIAL_SW_FIFO_IOMOD_GET_MODE, fifo_instance, 0);  
printf("Mode = %lu\n", ret);  
//Gibt den vorher eingestellten FIFO Mode wieder.
```

3.17.1.12. DapiSpecialSWFifoIOModIOActivate

Beschreibung

Dieser Befehl aktiviert die FIFO- I/O Ein-/Ausgabe.

Definition

```
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_SW_FIFO, cmd, fifo_instance, 0);
```

Parameter

cmd = DAPI_SPECIAL_SW_FIFO_IOMOD_IO_ACTIVATE

fifo_instance = Gibt die Instanz des Software FIFO an

Programmierbeispiel

```
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_SW_FIFO,  
DAPI_SPECIAL_SW_FIFO_IOMOD_IO_ACTIVATE, fifo_instance,  
0);  
//Das automatische Ausgeben des FIFO an das Modul wird aktiviert.
```

3.17.1.13. DapiSpecialSWFifoIOModIODeactivate

Beschreibung

Dieser Befehl deaktiviert die FIFO- I/O Ein-/Ausgabe.

Definition

```
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_SW_FIFO, cmd, fifo_instance, 0);
```

Parameter

cmd = DAPI_SPECIAL_SW_FIFO_IOMOD_IO_DEACTIVATE

fifo_instance = Gibt die Instanz des Software FIFO an

Programmierbeispiel

```
DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_SW_FIFO,  
DAPI_SPECIAL_SW_FIFO_IOMOD_IO_DEACTIVATE,  
fifo_instance, 0);  
//Das automatische Ausgeben des FIFO an das Modul wird deaktiviert.
```

3.17.1.14. DapiSpecialSWFifoIOModGetBytesPerSample

Beschreibung

Dieser Befehl gibt an wie viel Bytes für das Schreiben in den D/A-Wandler notwendig sind.

Beispiel: Werden bei einem 16-Bit (2Byte) D/A-Wandler 3 D/A-Kanäle beschrieben, werden also 3x2 Bytes pro Sample benötigt. Der Wert 6 wird wiedergegeben.

Bei einem A/D-Wandler kommt ein zusätzliches Byte für den Timestamp dazu.

Definition

```
ULONG DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_SW_FIFO, cmd,  
    fifo_instance, 0);
```

Parameter

cmd = DAPI_SPECIAL_SW_FIFO_IOMOD_GET_BYTES_PER_SAMPLE

fifo_instance = Gibt die Instanz des Software FIFO an

Return-Wert

Benötigte Bytes pro Sample

Programmierbeispiel

```
unsigned long ret = DapiSpecialCommand(handle,  
    DAPI_SPECIAL_CMD_SW_FIFO,  
    DAPI_SPECIAL_SW_FIFO_IOMOD_GET_BYTES_PER_SAMPLE,  
    fifo_instance, 0);  
printf("Benötigte Bytes = %lu\n", ret);  
//Ausgabe der notwendigen Bytes pro Sample.
```


3.17.1.15. DapiSpecialSWFifoIOModGetBytesAvailable

Beschreibung

Dieser Befehl gibt an, wie viel Bytes zum schreiben zur Verfügung stehen.

Definition

```
ULONG DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_SW_FIFO, cmd,
    fifo_instance, 0);
```

Parameter

cmd = DAPI_SPECIAL_SW_FIFO_IOMOD_GET_BYTES_AVAILABLE

fifo_instance = Gibt die Instanz des Software FIFO an

Return-Wert

Anzahl der Bytes

Programmierbeispiel

```
unsigned long ret = DapiSpecialCommand(handle,
    DAPI_SPECIAL_CMD_SW_FIFO,
    DAPI_SPECIAL_SW_FIFO_IOMOD_GET_BYTES_AVAILABLE,
    fifo_instance, 0);
printf("Verfügbaren Bytes = %lu\n", ret);
//Anzahl Bytes.
```

3.17.1.16. DapiSpecialSWFifoIOModGetBytesFree

Beschreibung

Dieser Befehl gibt an, wie viel Bytes im Software Fifo des Submodules noch frei sind.

Definition

```
ULONG DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_SW_FIFO, cmd,  
    fifo_instance, 0);
```

Parameter

cmd = DAPI_SPECIAL_SW_FIFO_IOMOD_GET_BYTES_FREE

fifo_instance = Gibt die Instanz des Software FIFO an

Return-Wert

Anzahl der Bytes

Programmierbeispiel

```
unsigned long ret = DapiSpecialCommand(handle,  
    DAPI_SPECIAL_CMD_SW_FIFO,  
    DAPI_SPECIAL_SW_FIFO_IOMOD_GET_BYTES_FREE,  
    fifo_instance, 0);  
printf("Freie Fifo Bytes = %lu\n", ret);  
//Anzahl Bytes.
```

3.17.1.17. DapiSpecialSWFifoIOModGetStatTXCnt

Beschreibung

Gibt an, wie viele Pakete vom Submodul an den Software Fifo gesendet wurden.

Definition

```
ULONG DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_SW_FIFO, cmd,  
fifo_instance, 0);
```

Parameter

cmd = DAPI_SPECIAL_SW_FIFO_IOMOD_GET_STAT_TX_CNT

fifo_instance = Gibt die Instanz des Software FIFO an

Return-Wert

Anzahl der Bytes

Programmierbeispiel

```
unsigned long ret = DapiSpecialCommand(handle,  
DAPI_SPECIAL_CMD_SW_FIFO,  
DAPI_SPECIAL_SW_FIFO_IOMOD_GET_STAT_TX_CNT,  
fifo_instance, 0);  
printf("Gesendete Datenpakete = %lu\n", ret);  
//Ausgabe der Bytes
```

3.17.1.18. DapiSpecialSWFifoIOModGetStatTXErrCnt

Beschreibung

Gibt an, wie oft Fehler beim Senden der Datenpakete an den Software Fifo auftraten.

Fehler treten auf, wenn zum Beispiel der Fifo überläuft.

Definition

ULONG DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_SW_FIFO, cmd, fifo_instance, 0);

Parameter

cmd = DAPI_SPECIAL_SW_FIFO_IOMOD_GET_STAT_TX_ERR_CNT

fifo_instance = Gibt die Instanz des Software FIFO an

Return-Wert

Anzahl der aufgetretenen Fehler beim Senden der Bytes

Programmierbeispiel

```
unsigned long ret = DapiSpecialCommand(handle,
DAPI_SPECIAL_CMD_SW_FIFO,
DAPI_SPECIAL_SW_FIFO_IOMOD_GET_STAT_TX_ERR_CNT,
fifo_instance, 0);
printf("Verfügbaren Bytes = %lu\n", ret);
//Ausgabe der Anzahl an aufgetretenen Fehler beim senden der Bytes
```

3.17.1.19. DapiSpecialSWFifoIOModGetStatRXCnt

Beschreibung

Gibt an, wie viele Datenpakete vom Software Fifo empfangen wurden.

Definition

```
ULONG DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_SW_FIFO, cmd,  
    fifo_instance, 0);
```

Parameter

cmd = DAPI_SPECIAL_SW_FIFO_IOMOD_GET_STAT_RX_CNT

fifo_instance = Gibt die Instanz des Software FIFO an

Return-Wert

Anzahl der empfangenen Bytes

Programmierbeispiel

```
unsigned long ret = DapiSpecialCommand(handle,  
    DAPI_SPECIAL_CMD_SW_FIFO,  
    DAPI_SPECIAL_SW_FIFO_IOMOD_GET_STAT_RX_CNT,  
    fifo_instance, 0);  
printf("Verfügbaren Bytes = %lu\n", ret);  
//Anzahl der empfangen Bytes.
```

3.17.1.20. DapiSpecialSWFifoIOModGetStatRXErrCnt

Beschreibung

Gibt an, wie oft Fehler beim Empfangen der Datenpakete Fehler auftraten. Fehler treten auf, wenn keine Daten im Fifo vorhanden sind, ein Abruf aber stattfindet.

Definition

ULONG DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_SW_FIFO, cmd, fifo_instance, 0);

Parameter

cmd = DAPI_SPECIAL_SW_FIFO_IOMOD_GET_STAT_RX_ERR_CNT

fifo_instance = Gibt die Instanz des Software FIFO an

Return-Wert

Anzahl der aufgetretenen Fehler beim Empfangen der Bytes

Programmierbeispiel

```
unsigned long ret = DapiSpecialCommand(handle,
DAPI_SPECIAL_CMD_SW_FIFO,
DAPI_SPECIAL_SW_FIFO_IOMOD_GET_STAT_RX_ERR_CNT,
fifo_instance, 0);
printf("Verfügbaren Bytes = %lu\n", ret);
//Ausgabe der Anzahl an aufgetretenen Fehler beim Empfangen der Bytes
```

3.17.1.21. DapiSpecialSWFifoGetStatus

Beschreibung

Mit diesem Befehl können Statuswerte abgerufen werden.

Definition

```
ULONG DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_SW_FIFO, cmd,
fifo_instance, 0);
```

Parameter

cmd = DAPI_SPECIAL_SW_FIFO_GET_STATUS

fifo_instance = Gibt die Instanz des Software FIFO an

Return-Wert

Befehl	Beschreibung (FIFO-Status erzeugt einen Return-Wert...)	Wert(hex)
DAPI_SPECIAL_SW_FIFO_STATUS_IS_ACTIVE	... wenn die Ausgabe des D/A-Wandlers aktiv ist	0x01
DAPI_SPECIAL_SW_FIFO_STATUS_IO_IS_ACTIVE	... wenn die Ausgabe des FIFO I/O aktiv ist	0x02
DAPI_SPECIAL_SW_FIFO_STATUS_FIFO_OVERFLOW	... wenn zuviele Daten in den FIFO geschrieben werden	0x04
DAPI_SPECIAL_SW_FIFO_STATUS_FIFO_UNDERRUN	... wenn der FIFO leer läuft (es soll eine Ausgabe am I/O-Modul stattfinden, aber es sind keine Daten im Fifo vorhanden)	0x08
DAPI_SPECIAL_SW_FIFO_STATUS_FIFO_OUT_OF_SYNC	... wenn die FIFO-Kommunikation innerhalb der Module unterbrochen ist	0x10

Programmierbeispiel

```
unsigned long ret;

ret = DapiSpecialCommand(handle,
DAPI_SPECIAL_CMD_SW_FIFO, DAPI_SPECIAL_SW_FIFO_GET_STATU
S, fifo_instance, 0);
if((ret & 0x01) != 0) {printf("is_active");}
if((ret & 0x02) != 0) {printf("io_is_active");}
if((ret & 0x04) != 0) {printf("fifo_overflow");}
if((ret & 0x08) != 0) {printf("fifo_underrun");}
if((ret & 0x10) != 0) {printf("fifo_out_of_sync");}
```


3.17.1.22. DapiSpecialSWFifoGetStatusBytesOut

Beschreibung

Dieser Befehl gibt an, wie viel Bytes gesendet wurden.

Definition

```
ULONG DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_SW_FIFO, cmd,  
    fifo_instance, 0);
```

Parameter

cmd = DAPI_SPECIAL_SW_FIFO_GET_STATUS_BYTES_OUT

fifo_instance = Gibt die Instanz des Software FIFO an

Return-Wert

Anzahl der gesendeten Bytes

Programmierbeispiel

```
unsigned long ret = DapiSpecialCommand(handle,  
    DAPI_SPECIAL_CMD_SW_FIFO,  
    DAPI_SPECIAL_SW_FIFO_GET_STATUS_BYTES_OUT,  
    fifo_instance, 0);  
printf("Bytes gesendet = %lu\n", ret);  
//Ausgabe der bereits gesendeten Bytes
```

3.17.1.23. DapiSpecialSWFifoGetBytesFree

Beschreibung

Dieser Befehl dient zum Auslesen der freien Bytes im Software FIFO-Buffer.

Definition

```
ULONG DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_SW_FIFO, cmd,  
    fifo_instance, 0);
```

Parameter

cmd = DAPI_SPECIAL_SW_FIFO_GET_BYTES_FREE

fifo_instance = Gibt die Instanz des Software FIFO an

Return-Wert

Freie Bytes des Software FIFO

Programmierbeispiel

```
unsigned long ret = DapiSpecialCommand(handle,  
    DAPI_SPECIAL_CMD_SW_FIFO,  
    DAPI_SPECIAL_SW_FIFO_GET_BYTES_FREE, fifo_instance, 0);  
printf("Freier Speicher = %lu\n", ret);  
//Ausgabe der noch freien Bytes des Speichers.
```

3.17.1.24. DapiSpecialSWFifoGetActivity

Beschreibung

Mit diesem Befehl wird der Übertragungs-Status des FIFO abgerufen (ob aktiv oder inaktiv).

Definition

```
ULONG DapiSpecialCommand(handle, DAPI_SPECIAL_SW_FIFO, cmd,  
fifo_instance, 0);
```

Parameter

cmd = DAPI_SPECIAL_SW_FIFO_GET_ACTIVITY

fifo_instance = Gibt die Instanz des Software FIFO an

Return-Wert

Return = 0 (Übertragung ist deaktiviert)

Return = 1 (Übertragung ist aktiviert)

Programmierbeispiel

```
unsigned long ret = DapiSpecialCommand(handle,  
DAPI_SPECIAL_CMD_SW_FIFO,  
DAPI_SPECIAL_SW_FIFO_GET_ACTIVITY, fifo_instance, 0);  
printf("Status = %lu\n", ret);  
//Übertragungs-Status wird abgerufen
```

3.17.1.25. DapiSpecialSWFifoGetInstanceType

Beschreibung

Mit diesem Befehl kann ausgelesen werden, bei welchem Kanal es sich um einen Ein- bzw. Ausgangskanal handelt.

Definition

```
ULONG DapiSpecialCommand(handle, DAPI_SPECIAL_CMD_SW_FIFO, cmd,  
fif_instance, 0);
```

Parameter

cmd = DAPI_SPECIAL_SW_FIFO_GET_INSTANCE_TYPE

fif_instance = Gibt die Instanz des Software FIFO an

Return-Wert

Befehl	Beschreibung	Wert(hex)
DAPI_SPECIAL_INSTANCE_TYPE_FIFO_IN	... gibt an, ob es sich bei dem Kanal um einen Eingang handelt	0x01
DAPI_SPECIAL_INSTANCE_TYPE_FIFO_OUT	... gibt an, ob es sich bei dem Kanal um einen Ausgang handelt	0x02

Programmierbeispiel

```
unsigned long ret;

for(int i=0;i!=10;++i)
{
    ret = DapiSpecialCommand(handle,
    DAPI_SPECIAL_CMD_SW_FIFO,
    DAPI_SPECIAL_SW_FIFO_GET_INSTANCE_TYPE, i, 0);
    switch(ret)
    {
        case DAPI_SPECIAL_INSTANCE_TYPE_FIFO_IN:
printf("Instance %d = FIFO_IN\n\r", i);break;
        case DAPI_SPECIAL_INSTANCE_TYPE_FIFO_OUT:
printf("Instance %d = FIFO_OUT\n\r", i);break;
        default:
printf("Instance %d = INVALID\n\r", i);break;
    }
}
//Gibt wieder ob es sich um einen Ein- oder Ausgangskanal handelt
```

3.17.2. DapiWriteFifo

Beschreibung

Dieser Befehl schreibt Datensätze in den Software FIFO.

Definition

*DapiWriteFifo(ULONG handle, ULONG fifo_instance, ULONG type, UCHAR * buffer, ULONG buffer_length);*

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

fifo_instance=Gibt die Instanz des Software FIFO an

type=Gibt den FIFO-Typ an

buffer=Buffer für den zu sendenden Datensatz

buffer_length=Länge des Buffers

Programmierbeispiel

```
DapiWriteFifo(handle, fifo_instance, type, buffer,
buffer_length);
//Schreibt den Datensatz in den Software FIFO.
```

3.17.3. DapiReadFifo

Beschreibung

Dieser Befehl liest den Software-FIFO aus.

Definition

```
ULONG DapiReadFifo(ULONG handle, ULONG fifo_instance, ULONG type, UCHAR  
* buffer, ULONG buffer_length);
```

Parameter

handle=Dies ist das Handle eines geöffneten Moduls

fifo_instance=Gibt die Instanz des Software FIFO an

type=Gibt den FIFO-Typ an

buffer=Buffer für den zu empfangenden Datensatz

buffer_length=Länge des Buffers

Return-Wert

Länge der ausgelesenen FIFO-Datenstätze

Aufbau eines FIFO-Datensatz (Beispiel mit 2 aktiven A/D-Kanälen, A/D 0 und A/D 4)

Byte	Bedeutung	Wert [hex]
0	RO_FIFO_ID_START	0xf0
1	FIFO-Typ	
2	Zeitstempel (Bit0..Bit7)	
3	Zeitstempel (Bit8..Bit15)	
4	Aktive A/D-Kanäle (Bit0..Bit7)	0x11
5	Aktive A/D-Kanäle (Bit8..Bit15)	0x00
6	A/D-Wert Kanal 0 (Bit0..Bit7)	
7	A/D-Wert Kanal 0 (Bit8..Bit15)	
8	A/D-Wert Kanal 4 (Bit0..Bit7)	
9	A/D-Wert Kanal 4 (Bit8..Bit15)	
10	RO_FIFO_ID_END	0xf1

FIFO-Datenstatz = 7 Bytes ID + (2 x Anzahl aktiver A/D-Kanäle) Bytes Daten

RO_FIFO_ID_START

Signalisiert den Anfang eines neuen FIFO-Datensatzes. Die RO_FIFO_ID_START hat immer den Wert 0xf0 [hex]

FIFO Typ

Gibt den FIFO Typ an (z.B. RO_FIFO_ID_TYPE_AD16M0 für A/D-FIFO)

Zeitstempel

Gibt den 16-Bit Zeitstempel des aktuellen Datensatzes an. Zeit-Referenz ist hierbei der Zeitpunkt der Aktivierung des FIFO.

Beim Überlauf des Zeitstempels, wird dieser auf 0 zurückgesetzt.

Aktive A/D-Kanäle

Gibt einen 16 Bit Wert für die aktuell aktiven A/D-Kanäle an. Jedes Bit steht für einen Kanal (Bit0 -> A/D0, Bit1 -> A/D1, .. Bit15 -> A/D15).

Ist das Bit gesetzt, ist der entsprechende A/D-Kanal aktiv

RO_FIFO_ID_END

Signalisiert das Ende eines FIFO-Datensatzes. Die RO_FIFO_ID_END hat immer den Wert 0xf1 [hex]

Bemerkung

Beachten Sie, dass der Software FIFO zuvor mit dem Befehl "DapiSpecialCMDAD" aktiviert, bzw. initialisiert werden muss.

Programmierbeispiel

```
bytes_received = DapiReadFifo(handle, fifo_instance,  
DAPI_FIFO_TYPE_READ_AD_FIFO, buffer, sizeof(buffer));  
//Liest den Software FIFO aus
```

3.18. Ausgabe-Timeout verwalten

3.18.1. 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

3.18.1.1. 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.
```

3.18.1.2. 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.
```

3.18.1.3. 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.
```


3.18.1.4. 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.
```

3.18.1.5. 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.
```

3.18.1.6. 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.
```

3.18.1.7. 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.
```

3.18.1.8. 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.
```

3.18.1.9. 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.
```

3.18.1.10. 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.
```

3.18.1.11. 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.
```


3.19. Testfunktionen

3.19.1. 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

3.20. Register Schreib-Befehle

3.20.1. 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.

3.20.2. 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.

3.20.3. 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.

3.20.4. 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.

3.21. Register Lese-Befehle

3.21.1. 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.

3.21.2. 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.

3.21.3. 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.
```


3.21.4. 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.

3.22. Programmier-Beispiel

```
// *****
// *****
//
// (c) DEDITEC GmbH, 2009
//
// web: http://www.deditec.de
//
// mail: vertrieb@deditec.de
//
// dtapi_prog_beispiel_input_output.cpp
//
// *****
// *****
//
// Folgende Bibliotheken beim Linken mit einbinden: delib.lib
// Dies bitte in den Projekteinstellungen
// (Projekt/Einstellungen/Linker(Objekt-
// Bibliothek-Module) .. letzter Eintrag konfigurieren
#include <windows.h>
#include <stdio.h>
#include "conio.h"
#include "delib.h"
// *****
// *****

void main(void)
{
    unsigned long handle;
    unsigned long data;
    unsigned long anz;
    unsigned long i;
    unsigned long chan;
    // -----
    // USB-Modul öffnen
    handle = DapiOpenModule(USB_Interface8,0);
    printf("USB_Interface8 handle = %x\n", handle);
    if (handle==0)
    {
        // USB Modul wurde nicht gefunden
        printf("Modul konnte nicht geöffnet werden\n");
        printf("TASTE für weiter\n");
        getch();
        return;
    }
    // Zum Testen - ein Ping senden
    // -----
    printf("PING\n");
    anz=10;
    for(i=0;i!=anz;++i)
    {
        data=DapiPing(handle, i);
        if(i==data)
        {
            // OK
            printf(".");
        }
        else
        {

```

```

// No answer
printf("E");
}
}
printf("\n");

// -----
// Einen Wert auf die Ausgänge schreiben
data = 255;
DapiWriteByte(handle, 0, data);
printf("Schreibe auf Adresse=0 daten=0x%x\n", data);
// -----
// Einen Wert auf die Ausgänge schreiben
data = 255;
DapiWriteByte(handle, 1, data);
printf("Schreibe auf Adresse=1 daten=0x%x\n", data);
// -----
// Einen Wert auf die Ausgänge schreiben
data = 255;
DapiWriteByte(handle, 2, data);
printf("Schreibe auf Adresse=2 daten=0x%x\n", data);
// -----
// Einen Wert von den Eingängen lesen
data = (unsigned long) DapiReadByte(handle, 0);
printf("Gelesene Daten = 0x%x\n", data);
// -----
// Einen A/D-Wert lesen
chan=11; // read chan. 11
data = DapiReadWord(handle, 0xff010000 + chan*2);
printf("Adress=%x, ret=%x volt=%f\n", chan, data, ((float) data) / 1024*5); //
Bei 5 Volt Ref
// -----
// Modul wieder schliessen
DapiCloseModule(handle);
printf("TASTE für weiter\n");
getch();
return ;
}

```

3.23. 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

Sicherheitsmaßnahmen



IV

4. Sicherheitsmaßnahmen

Es gibt mehrere Sicherheitsmaßnahmen, um Ihr Modul vor unautorisierten Zugriffen von außen zu schützen.

4.1. Schreibschutz

Mit dem DIP-Schalter 2, können Sie den Zugriff via Ethernet- oder USB-Schnittstelle auf das **Module-Configuration-Memory** mit dem Schreibschutz deaktivieren.

DIP-2 on = Schreibschutz aktiv

DIP-2 off = Schreibschutz deaktiviert

4.2. TCP-Verschlüsselung

Mit dem ICT-Tool können Sie die Kommunikation über TCP verschlüsseln.

Auch der Teil- oder Vollzugriff auf das **Module-Configuration-Memory** kann mit Hilfe der Admin- und User-Protokolle eingestellt werden.

Mehr Informationen dazu finden Sie im Kapitel : **TCP-Verschlüsselung**

4.3. Weboberfläche

Der Zugriff auf das Modul über die Weboberfläche, kann mit Hilfe des ICT-Tool aktiviert/deaktiviert werden.

Weitere Informationen dazu finden Sie im Kapitel: **TCP-Verschlüsselung**

Anhang



5. Anhang

5.1. Revisionen

Rev 3.01	DEDITEC Design Update 2022
Rev 3.00	DEDITEC Design Update 2021
Rev 2.20	Kapitel "Software" und "DELIB API Referenz" überarbeitet
Rev 2.19	Neue Befehle "DapiDOSetBit32" und "DapiDOClrBit32" hinzugefügt. Diverse Delib Befehle überarbeitet
Rev 2.18	Neuer A/D-Befehl "DapiSpecialADReadMultipleAD" und neues Kapitel "Software FIFO verwalten"
Rev 2.17	Kapitel "DELIB CLI (command-line interface)" ergänzt
Rev 2.16	Index hinzugefügt
Rev 2.15	Neuer CNT48 Befehl "DapiSpecialCNT48DIGet1" Ergänzung bei Befehl "DapiSpecialCMDSetDirDX8"
Rev 2.14	Ergänzung Kapitel "Software"
Rev 2.13	Ergänzung Kapitel "DELIB CLI" und Ergänzung der Parameter bei Verwaltungsfunktion "DapiSpecialCMDGetModuleConfig"
Rev 2.12	Ergänzung Kapitel "Digitale Zähler Funktionen", Kapitel "Temperatur Funktionen", Kapitel "Testprogramme" und neue Verwaltungsfunktion "DapiOpenModuleEx"
Rev 2.11	Neue DI Befehle "DapiSpecialDIFFFilterValueSet" und "DapiSpecialDIFFFilterValueGet"
Rev 2.10	Neue CNT48 Befehle "DapiCnt48ModeSet", "DapiCnt48ModeGet", "DapiCnt48CounterGet32" und "DapiCnt48CounterGet48"
Rev 2.09	Neuer Temperatur Befehl "DapiTempGet"
Rev 2.08	Ergänzung DELIB Console Execute und neue Verwaltungsfunktion

	"DapiSpecialCMDGetModuleConfig"
Rev 2.07	Neue Verwaltungsfunktion "DapiGetDELIBVersion", neuer DI Befehl "DapiSpecialCounterLatchAll", "DapiSpecialCounterLatchAllWithReset" und Ergänzung der Modi bei "DapiDIGetCounter"
Rev 2.06	Neuer DO-Befehl "DapiDOSet1_WithTimer"
Rev 2.05	Ergänzung des Watchdog-Sticks
Rev 2.04	Neues Programmierbeispiel bei Befehl "DAPI_SPECIAL_CMD_SET_DIR_DX_1" 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 2.03	Neuer Stepper Befehl "DAPI_STEPPER_CMD_GO_POSITION_RELATIVE"
Rev 2.02	Neuer D/A-Befehl "DAPI_SPECIAL_CMD_DA" und DO-Befehl "DAPI_SPECIAL_CMD_TIMEOUT_GET_STATUS"
Rev 2.01	Ergänzung der DELIB Befehle "DAPI_STEPPER_CMD_SET_MOTORCHARACTERISTIC" "DAPI_STEPPER_CMD_GET_MOTORCHARACTERISTIC" und "DAPI_STEPPER_CMD_GO_REFSWITCH"
Rev 2.00	Designanpassung
Rev 1.3	Softwareinstallation und Verzeichnisstruktur der DELIB
Rev 1.2	Ergänzung des Stepper Motors
Rev 1.1	Ergänzung von diversen AD/DA Befehlen
Rev 1.00	Erste DEDITEC Anleitung

5.2. 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.