

CANopen



BEDIENUNGSANLEITUNG

SUPPLYCOM APPLICATION INTERFACE

Stand 06/2005

SysDesign GmbH
Säntisstrasse 25
D-88079 Kressbronn

Telefon: +49 7543 9620-0
Telefax: +49 7543 9620-22
Internet: www.SysDesign.info

Inhaltsübersicht:

1	GENERELLES	3
1.1	Nutzungsbedingungen	3
1.2	Produkt-Support	4
2	FUNKTIONSBESCHREIBUNG	5
2.1	Unterstützte Netzgeräte	5
2.2	Unterstützte Kommunikationsarten	5
3	INSTALLATIONSHINWEISE	6
3.1	Systemvoraussetzungen	6
3.2	Installation	6
3.3	Hinweise für SW-Entwickler	6
4	HANDHABUNG	7
4.1	Anzeige	7
4.2	Hinweise zur Bedienung	8
5	BESCHREIBUNG DER TREIBER-DLL	9
5.1	Einleitung	9
5.2	Treiberaufbau	9
5.3	API Beschreibung.....	10
5.4	Aufrufreihenfolge	13
5.5	Programmierbeispiele	14
6	STICHWORTVERZEICHNIS	15

1 Generelles

Bitte lesen Sie sich vor Inbetriebnahme die vorliegende Bedienungsanleitung aufmerksam durch. Bewahren Sie sie stets gut auf, damit Sie jederzeit darauf zugreifen können, auch falls am Einsatzort kein Internet-Zugang zur Verfügung steht.



Beachten Sie insbesondere die nachstehenden Nutzungs- und Gewährleistungsbedingungen, denen Sie durch die Installation des Treibers ausdrücklich zustimmen. Wenn Sie mit den Bestimmungen nicht einverstanden sind, dürfen Sie die Software nicht verwenden

1.1 Nutzungsbedingungen

§1 Allgemeines

Dieser Endbenutzer-Lizenzvertrag (nachfolgend "Vertrag") ist eine rechtliche Vereinbarung zwischen einem Anwender als einer natürlichen oder juristischen Person (nachfolgend "Anwender" genannt) und der SysDesign GmbH (nachfolgend "SysDesign" genannt), über die entsprechende Verwendung des SupplyCom Application Interface, einschließlich der möglichen Verwendung von Programmbeschreibungen und Bedienungsanleitungen.

Mit Installation der Software erklären Sie als Anwender Ihr ausdrückliches Einverständnis, an die Bestimmungen dieses Vertrages gebunden zu sein.

§2 Lizenzerteilung

Das SupplyCom Application Interface wird kostenfrei zur Verfügung gestellt. Dem Anwender werden an der Software unentgeltlich Nutzungsrechte eingeräumt.

Bei Einhaltung der Geschäftsbedingungen sind Sie berechtigt zur Erstellung und Nutzung einer beliebigen Anzahl Kopien der entsprechenden Software. Die Erstellung von Kopien und eine Weitergabe an Dritte darf nur unentgeltlich, unter Beibehaltung der Marke und des Copyright-Vermerks sowie unter Hinweis auf diese Nutzungsbedingungen erfolgen.

§3 Copyright

Die Software ist Eigentum der SysDesign GmbH. SysDesign behält sich sämtliche in diesem Vertrag nicht ausdrücklich gewährten Rechte, Rechtstitel und Nutznießung an der Software vor. Die Software darf außerhalb des durch diesen Vertrag geregelten Gültigkeitsbereichs nicht ohne schriftliche Genehmigung verwendet, kopiert, angepasst oder vertrieben werden.

§4 Einschränkungen

Sie dürfen die Software nicht weiterverkaufen, vermieten, leasen, verleihen oder Nutzungslizenzen vergeben. Sie dürfen die Software nicht rückentwickeln (Reverse Engineering), dekompileieren oder disassemblieren. Darüber hinaus ist es nicht zulässig, die Software oder das schriftliche Material ohne die Zustimmung von SysDesign anderweitig anzupassen oder zu verändern oder auf dieser Software oder dem schriftlichen Material basierende Produkte zu erstellen. Dies gilt auch für beliebige Kopien der Software.

§5 Gewährleistungsausschluss

Die Software wird ohne Mängelgewähr unentgeltlich bereitgestellt. Es findet daher weder ein kaufrechtliches noch sonstiges Gewährleistungsrecht Anwendung.

Der Anwender trägt das Risiko sowie die volle Verantwortung für die Auswahl der Software zum Erzielen der von ihm angestrebten Ergebnisse sowie für die Installation und die Verwendung der Software wie auch für die durch die Verwendung der Software erzielten Ergebnisse.

SysDesign schließt jede Gewährleistung sowohl hinsichtlich einer Ausführung ohne Fehler, Unterbrechungen oder anderweitige Störungen wie auch hinsichtlich der Kompatibilität mit beliebiger Hardware oder Software aus. SysDesign schließt auch stillschweigende Gewährleistungen für die Software und die begleitenden schriftlichen Materialien aus, einschließlich Gewährleistungen der allgemeinen Gebrauchstauglichkeit, der Eignung für einen bestimmten Zweck und der Nichtverletzung der Rechte Dritter.

§6 Haftungsbeschränkung

SysDesign übernimmt keine Haftung für Folgeschäden. Der Anwender trägt alle Kosten für Schäden selbst, die durch die Verwendung der Software oder durch die Verwendung der darin enthaltenen oder mit Hilfe der Software erstellten Informationen oder durch die Interaktion (oder fehlerhafte Interaktion) mit beliebiger anderer Hardware oder Software von SysDesign oder eines Drittanbieters entstehen.

Eine Haftung von SysDesign und seiner Händler für Schäden jedweder Art wird ausgeschlossen, die aus der Verwendung oder der Unmöglichkeit der Verwendung oder der Inkompatibilität der Software entstehen, einschließlich aber nicht begrenzt auf jegliche Folgeschäden wie Verlust von Geschäftsgewinnen, Geschäftsunterbrechungen, Verlust von Geschäftsinformationen, Störungen, Beeinträchtigungen oder Ausfall von Hardware oder Software, Reparaturkosten, Zeitwert oder andere finanzielle Verluste. Dies gilt auch, wenn auf die Möglichkeit solcher Schäden hingewiesen wurde.

§7 Grundlagen

Grundlage dieses Vertrages bildet das Recht der Bundesrepublik Deutschland. Gerichtsstand ist Sitz von SysDesign. Sollten einzelne Bestimmungen unwirksam sein, so bleibt hierdurch die Wirksamkeit der restlichen Bestimmungen im übrigen unberührt. Ergänzungen dieses Vertrages bedürfen der Schriftform.

Achtung

	Für Schäden am Gerät, die durch unsachgemäße Bedienung resultierend aus der Nichtbeachtung der Bedienungsanleitung oder durch Missachtung der vorstehenden Hinweise entstehen, besteht kein Gewährleistungsanspruch.
	Es wird grundsätzlich keine Haftung für Folgeschäden übernommen.
	Gegen eine Gebühr können Sie den Programmcode der Software offengelegt bekommen und Ihre Nutzungsrechte auf die freie Anpassung sowie die Verwendung für eigene Produkte erweitern. Die Bedienoberfläche ist als virtuelles Instrument verfügbar, das sich direkt in NI LabView® integrieren und beliebig anpassen lässt. Wenden Sie sich dazu an den Hersteller.

1.2 Produkt-Support

Diese Bedienungsanleitung, Gerätedateien sowie weiterer Produkt-Support steht Ihnen jederzeit im Internet zur Verfügung unter:

www.SupplyCom.de

Wenden Sie sich bei Unklarheiten, Hinweisen und Fragen immer zunächst an die Vertriebsstelle, bei der Sie das Produkt erworben haben.

2 Funktionsbeschreibung

Das vorliegende Application Interface ist eine Bedienoberfläche für das SupplyCom Kommunikationsmodul. Man kann damit ein angeschlossenes Netzteil vom PC aus direkt einstellen und betreiben.

Die Software läuft als eigenständige Anwendung und kann somit auf jedem PC unter kompatibelem Microsoft Windows Betriebssystem installiert werden.

Die Software besteht aus dem Gerätetreiber, der den dll-Funktionsumfang zur Verfügung stellt, und der eigentlichen Bedienoberfläche in Form eines virtuellen Instruments (VI).

Die Funktionalität des Gerätetreibers kann nach Installation aus einer beliebigen Windows-Anwendung heraus angesprochen werden.



Die Bedienoberfläche ist gegen Gebühr als virtuelles Instrument (VI) verfügbar, das sich direkt in NI LabView® oder LabWindows/CVI® integrieren und beliebig anpassen lässt. Wenden Sie sich dazu an den Hersteller.

2.1 Unterstützte Netzgeräte

Aktuell unterstützt werden:

Hersteller	Modell
Delta Elektronik	alle Standard-Modelle

Eine Erweiterung der unterstützten Hersteller und Modelle ist möglich. Bitte wenden Sie sich hierzu an den Hersteller.

2.2 Unterstützte Kommunikationsarten

Aktuell unterstützt werden:

Schnittstelle	Umfang
RS-232	Baudrates: 1.200/2.400/4.800/9.600/19.200 kBs
CANopen	Kommunikation über folgende Schnittstellenkarten der Marke Vector <ul style="list-style-type: none"> • CANCardXL • CANcaseXL • CANboardXL • CANboardXL pxi

Die Kommunikation über PROFIBUS wird von dem Application Interface derzeit nicht unterstützt.



Eine Erweiterung der unterstützten Schnittstellen und Kommunikationshardware ist möglich. Bitte wenden Sie sich hierzu an den Hersteller.

3 Installationshinweise

3.1 Systemvoraussetzungen

Die Software wurde in folgender Systemumgebung entwickelt und getestet. Die Kompatibilität mit anderen Systemumgebungen wurde nicht bewertet.

System	Eigenschaften	
PC	Prozessor	Pentium PIII
	Arbeitsspeicher	256 MB
	Betriebssystem	Microsoft Windows 2000
sonstige Anforderungen	Größe des Installers:	ca. 10 MB
	Speicherbedarf:	Anwendung selbst ca. 650 kB zusätzlich NI LabView® Runtime Engine
	vorhandene Schnittstelle	freie Schnittstelle gemäß Kapitel 2.2

3.2 Installation

Führen Sie die Datei "Setup.exe" aus und folgen Sie den Anweisungen.
Es werden folgende Komponenten installiert:

- Netzteil.exe
- Netzteil.ini (ohne Funktion)
- supplycom.dll (im Unterverzeichnis "data")
- supplycom.lib (im Unterverzeichnis "data")
- supplycom.h (im Unterverzeichnis "data")
- vxlapi.dll (im Unterverzeichnis "data", Version 5.1.1.0)
- LabView Run-Time-Engine

3.3 Hinweise für SW-Entwickler

	Die Bedienoberfläche ist gegen Gebühr als virtuelles Instrument (VI) verfügbar, das sich direkt in NI LabView® oder LabWindows/CVI® integrieren und beliebig anpassen lässt. Wenden Sie sich dazu an den Hersteller.
	Zur Einbindung des VI in der LabView®-Entwicklungsumgebung ist darauf zu achten, dass die supplycom.dll im gleichen Verzeichnis wie "Netzteil.vi" verfügbar ist. Ansonsten wird beim Öffnen des VI nach einer dll-Datei verlangt.
	Zur Einbindung des Supplycom-Treibers in C/C++-Projekte stehen supplycom.dll, supplycom.lib und supplycom.h zur Verfügung. Diese Dateien finden sich nach der Installation im Unterverzeichnis "data". Die dll-Funktionen sind im Kapitel "Beschreibung der Treiber-DLL" nachstehend beschrieben.
	Gegen eine Gebühr können Sie den Programmcode der Software offengelegt bekommen und Ihre Nutzungsrechte auf die freie Anpassung sowie die Verwendung für eigene Produkte erweitern. Wenden Sie sich hierzu an den Hersteller.

4 Handhabung

4.1 Anzeige

Die Bedienoberfläche ist gemäß Bild 4-1 gestaltet.

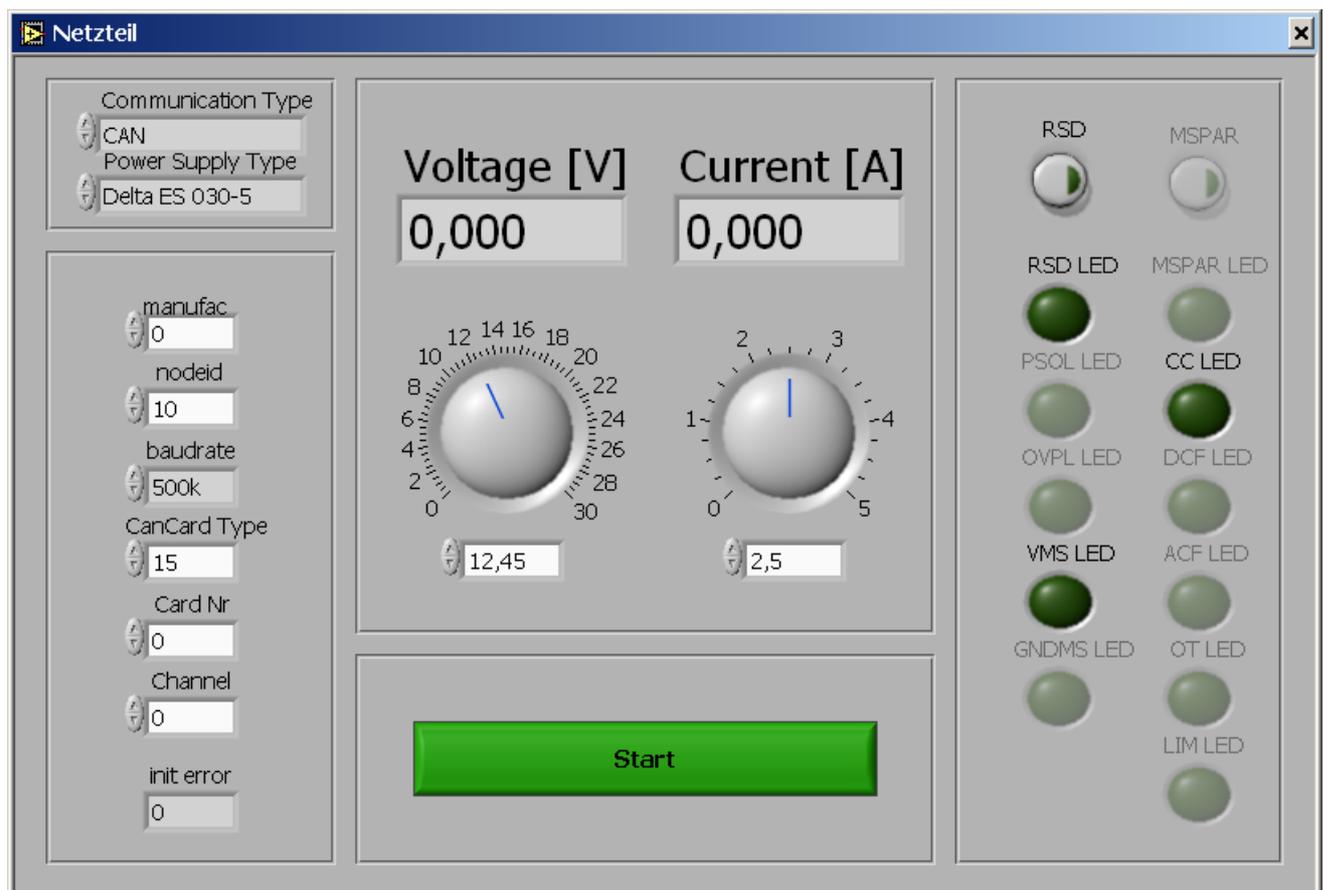


Bild 4-1: Die Bedienoberfläche - auch als virtuelles Instrument (VI)

Links befinden sich die Initialisierungseinstellungen. Oben kann die gewünschte Kommunikationsart und das erforderliche Netzgerätprofil ausgewählt werden. Die zur gewählten Kommunikationsart gehörenden Parameter werden im unteren Bereich angezeigt bzw. eingestellt.



Die Einstellungen müssen mit der Konfiguration des SupplyCom übereinstimmen. Näheres zur Konfiguration des SupplyCom finden Sie in der entsprechenden Bedienungsanleitung.

In der Mitte befinden sich Drehknöpfe zum Einstellen von Strom und Spannung mit den dazugehörigen Anzeigen sowie dem Start-Button.

Rechts befinden sich die digitalen Ein- und Ausgänge des Netzteils. Die im aktuellen Netzteilprofil nicht unterstützten Signale werden blass dargestellt.

4.2 Hinweise zur Bedienung

4.2.1 Einstellungen

Nachdem die Applikation gestartet wurde, muss die Kommunikationsart und der Netzteiltyp ausgewählt werden.

Für CAN-Betrieb sind die Parameter "Manufacturer", "NodeID", "Baudrate", "HwType", "HwIndex" und "Channel" einzustellen, für RS232-Betrieb die Parameter "Comport" und "Baudrate".

4.2.2 Betrieb

Danach kann man mit **"Start"** in den Betriebsmodus gehen.

Mit **"Stop"** wird der Betrieb beendet.



Unten links ist die Anzeige für den Initialisierungsstatus. Bei Start des Betriebsmodus muss dieser auf "0" springen. Andere Werte deuten auf eine fehlerhafte Initialisierung hin.



Durch drücken des "Stop"-Buttons werden die Sollwerte für Strom und Spannung am Netzgerät auf Null gesetzt.

Spannung und Strom werden entweder mit den beiden Drehknöpfen oder zur exakteren Einstellung direkt als Wert eingegeben. Die am Netzteil gemessenen Werte sind in den darüber liegenden Anzeigen ablesbar.



Beim Drehen von Spannungs- und Stromeinstellung mit der Maus ist darauf zu achten, dass der Mauszeiger beim Loslassen der linken Maustaste sich auf dem Drehknopf befindet. Sonst wird der neu eingestellte Wert nicht übernommen und gesendet.



Bei der direkten manuellen Eingabe von Werten über Tastatur mit "Return" bestätigen.
Beim Ändern der Werte über die Pfeiltasten wird der Wert sofort übernommen.

4.2.3 Programmende

Die Applikation wird durch Schließen des Fensters beendet.



Vor dem Schließen des Programms sollte der Betrieb mit "Stop" beendet werden, um Strom und Spannung auf "0" zu setzen.

5 Beschreibung der Treiber-DLL

5.1 Einleitung

Der Supplycom-Treiber besteht aus einer WIN32 DLL und einer zugehörigen Headerdatei. Er ermöglicht das Ansprechen eines einzelnen SupplyCom Moduls über CAN-Bus oder eine serielle Verbindung über RS232.

Dazu stellt er eine API aus C Funktionen bereit. Diese können z.B. von einem C-Programm aus verwendet werden (MS Visual C++ 6.0, MS .NET, NI LabWindows/CVI®), auch eine Verwendung in NI LabView® ist möglich.



Die Bedienoberfläche ist als virtuelles Instrument verfügbar, das sich direkt in LabView® integrieren und beliebig anpassen lässt. Wenden Sie sich hierfür an den Hersteller.

Der Treiber instanziert dynamisch in Abhängigkeit der eingestellten Kommunikationsart die entsprechende Klasse. Nach Außen sind nur reine C-Funktionen sichtbar.

5.2 Treiberaufbau

Der Treiber liegt in Form einer *.dll-Datei vor. Das Byte-Alignment beträgt 4 Bytes. Die Calling Convention ist C Declaration.

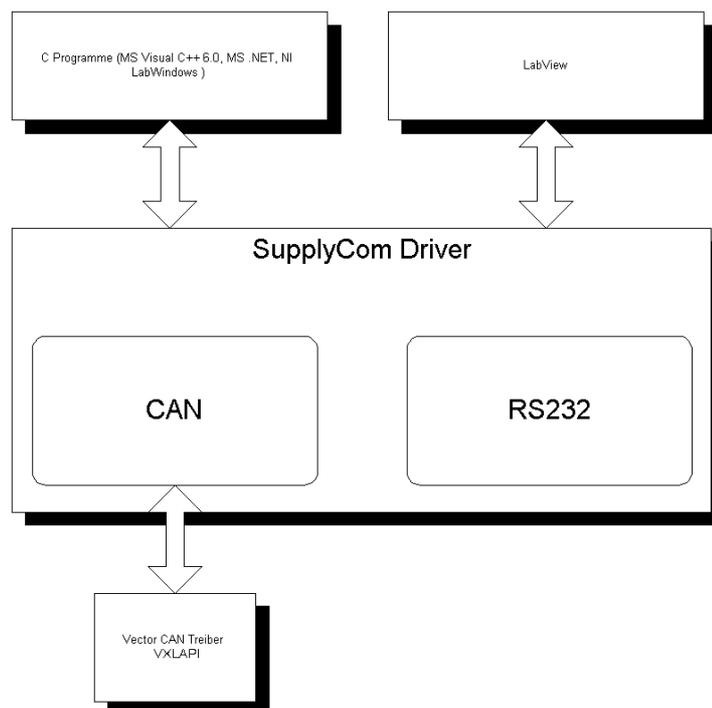


Bild 5-1: Treiberaufbau

5.3 API Beschreibung

Funktion	Beschreibung	Parameter	Rückgabewert
<pre>int init_can(int manufacturer, int nodeid, int baudrate, unsigned long hwtype, unsigned long hwindex, unsigned int channel)</pre>	<p>Initialisiert den Treiber für CAN Betrieb und führt eine hersteller- und kartenabhängige Hardware Initialisierung durch. Desweiteren versendet sie eine CAN Nachricht um das SupplyCom in den Betriebszustand zu versetzen. Muss als erste Funktion des Treibers aufgerufen werden! Nur bei Rückgabewert 0 dürfen weitere Funktionen aufgerufen werden.</p>	<p>manufacturer: 0 = Vector 1 = IXXAT 2 = MICROCONTROL 3 = HILSCHER</p> <p>nodeid: die auf dem SupplyCom konfigurierte Knotenadresse</p> <p>baudrate: Baudrate des CAN Bus</p> <p>hwtype: Modell des Herstellers. Zur Zeit nur Vector implementiert. Die Werte gelten nur innerhalb von manufacturer</p> <p>15 = CANCardXL 21 = CANCaseXL 23 = CANCaseXL_LOG 25 = CANBoardXL 27 = CANBoardXL_PXI</p> <p>hwindex: Zum Unterscheiden gleicher Hardware. Z.B. 0 = CANCardXL Nr. 1, 1 = CANCardXL Nr. 2, ...</p> <p>channel: Channel 1 = 0 Channel 2 = 1</p>	<p>0 = Initialisierung erfolgreich beendet</p> <p>ERR_FATAL = CAN Objekt konnte nicht instanziiert werden, womöglich reicht der Speicher nicht aus.</p> <p>-1 = Treiber konnte nicht geladen werden</p> <p>-2 = Hardware Konfiguration konnte nicht ausgelesen werden</p> <p>-3 = kein CAN Kanal gefunden (CAN Cabs?)</p> <p>-4 = Port konnte nicht geöffnet werden</p> <p>-5 = keine Erlaubnis CAN Parameter einzustellen</p>

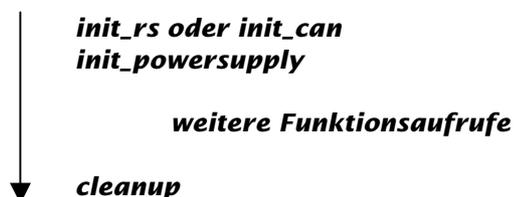
Funktion	Beschreibung	Parameter	Rückgabewert
int init_rs(int comport, int baudrate)	Initialisiert den Treiber für RS Betrieb und öffnet den angegebenen COM Port.	comport: 1 = COM 1 2 = COM 2 3 = COM 3 4 = COM 4 ... baudrate: 9600 = 9600 Baud 19200 = 19200 Baud 38400 = 38400 Baud	0 = Initialisierung erfolgreich beendet ERR_FATAL = RS232 Objekt konnte nicht instanziiert werden -1 = COM Port konnte nicht geöffnet werden
int init_powersupply(int powersupply)	Teilt dem Treiber mit, welches Netzgerät verwendet wird.	powersupply: 0 = Delta ES 015-10 1 = Delta ES 030-5 2 = Delta ES 075-2 3 = Delta ES 0300-0.45 4 = Delta ES 030-10 5 = Delta SM 15-100 6 = Delta SM 35-45 7 = Delta SM 52-30 8 = Delta SM 70-22 9 = Delta SM 120-13 10 = Delta SM 300-5 11 = Delta SM 15-200D 12 = Delta SM 30-100D 13 = Delta SM 45-70D- P087 14 = Delta SM 45-70D 15 = Delta SM 70-45D 16 = Delta SM 120-25D 17 = Delta SM 300-10D 18 = Delta SM 15-400 19 = Delta SM 30-200 20 = Delta SM 45-140 21 = Delta SM 60-100 22 = Delta SM 70-90 23 = Delta SM 120-50 24 = Delta SM 300-20 25 = Delta SM 1540-D 26 = Delta SM 7020-D 27 = Delta SM 3004-D	0 = Initialisierung erfolgreich beendet -1 = Netzgeräte Objekt konnte nicht erstellt werden

Funktion	Beschreibung	Parameter	Rückgabewert
void cleanup(void)	Löscht die angelegten Objekte und entlädt den CAN Treiber. Sollte bei Programmende aufgerufen werden.		
void set_analog(unsigned int millivolt, unsigned int milliamp)	Sendet die Stellwerte an das SupplyCom Modul.	millivolt: Angabe des neuen Spannungswertes (Netzgerät Ausgangsspannung) milliamp: Angabe des neuen Stromwertes	
int set_voltage(unsigned int millivolt)	Sendet einen neuen Spannungsstellwert an das SupplyCom Modul. Der Stellwert für den Strom bleibt unverändert.	millivolt: Angabe des neuen Spannungswertes in mV	0 = Vorgang erfolgreich
int set_current(unsigned int milliamp)	Sendet einen neuen Stromstellwert an das SupplyCom Modul. Der Stellwert für die Spannung bleibt unverändert.	milliamp: Angabe des neuen Stromwertes in mA	0 = Vorgang erfolgreich
unsigned int get_voltage(void)	Fragt das SupplyCom Modul nach dem momentan gemessenen Spannungswert und liefert diesen zurück.		Spannungswert in mV
unsigned int get_current(void)	Fragt das SupplyCom Modul nach dem momentan gemessenen Stromwert und liefert diesen zurück.		Stromwert in mA
void set_timeout(unsigned int mstimeout)	Timeout in ms. Gibt an wie lange auf eine CAN Nachricht gewartet werden soll. Der Default ist 10ms.	mstimeout: Timeout in ms	

Funktion	Beschreibung	Parameter	Rückgabewert
int set_digital(unsigned int func, char wert)	Setzt das angegebene digitale Flag(z.B. RSD) auf den Wert <wert> {0;1}	func: Angabe des digitalen Flags wert: 0 = loeschen 1 = setzen	0 = Erfolg -1 = ungueltiges Flag ERR_FATAL = kein Busobjekt / Netzgeräteeobjekt vorhanden (Init?)
int get_digital(unsigned int func)	Liest das angegebene digitale Flag.	func: Angabe des digitalen Flags	0 = das angegebene Flag ist nicht gesetzt 1 = das angegebene Flag ist gesetzt -1 = das angegebene Flag ist beim gewählten Netzgerät nicht vorhanden

5.4 Aufrufreihenfolge

Die ersten Funktionen, die aus dem Treiber aufgerufen werden müssen, sind **init_can** bzw. **init_rs**, gefolgt von **init_powersupply**. Erst danach dürfen weitere Funktionen aufgerufen werden. Zum Schluss sollte die Funktion **cleanup** aufgerufen werden.



5.5 Programmierbeispiele

Zum Linken der Beispielprogramme sind die entsprechende Lib-Dateien in der jeweiligen Entwicklungsumgebung einzubinden.

5.5.1 CAN

```
int main()
{
    unsigned int voltage = 0;
    unsigned int current = 0;

    //Vector CANCardXL mit 500kBit/s; SupplyCom Node ID: 10
    if( init_can(0, 10, 500000, 15, 0, 0) >= 0 )
    {
        //Delta SM 35-45
        init_powersupply( 6 );

        //1V 2A
        set_analog( 1000, 2000 );

        //Spannung / Strom lesen
        voltage = get_voltage();
        current = get_current();

        printf("Spannung: %d\n", voltage);
        printf("Strom: %d\n", current);

        set_voltage( 20000 );
        set_current( 40000 );

        voltage = get_voltage();
        current = get_current();

        printf("Spannung: %d\n", voltage);
        printf("Strom: %d\n", current);

        set_voltage( 0 );
        set_current( 0 );

        //RSD setzen
        set_digital( 1, 1 );

        //RSD lesen
        int digi = 0;
        digi = get_digital( 1 );

        printf("RSD: %d\n", digi );

        //RSD loeschen
        set_digital( 1, 0 );

        //Treiber freigeben
        cleanUp();
    }
    else
    {
        printf("Vector CAN Hardware konnte nicht gefunden werden!");
    }
    return 0;
}
```

5.5.2 RS232

In obenstehenden Beispiel ist lediglich die Initialisierung zu ändern:

```
if( init_rs( 1, 9600 ) >= 0 )
```

6 Stichwortverzeichnis

API	11	LabView®	7
Aufruf.....	14	LabWindows/CVI®	7
Bedienungsanleitung	4	Lizenznehmer	3
Betrieb.....	9	Netzgeräte.....	5
C/C++-Projekte	7	Produkt-Support.....	4
dll-Datei	10	Programmende	9
Einstellungen.....	9	Schnittstellenkarten	5
Endbenutzer-Lizenzvertrag	3	Support	4
Gerätetreiber	5	Systemvoraussetzungen.....	7
Installation	7	Treiber.....	10
Kommunikationsarten	5	VI	<i>Siehe</i> virtuelles Instrument
Konfiguration	8	virtuelles Instrument.....	5, 8