

DLL Version 4.37

1. Allgemeines

Die vorliegende Version des Treibers für das Messdatenerfassungssystem **DC-HI-NET** wird unter dem Dateinamen **DCHINET.DLL** in Form einer 16 Bit DLL (Dynamic Link Library) für Microsoft-Windows® und unter dem Namen **DCH_32.DLL** in Form einer 32 Bit DLL für die Betriebssysteme Windows® 95, Windows® 98 / ME, Windows NT®, Windows® 2000 / XP / VISTA, Windows 7 ausgeliefert.

Unter Zuhilfenahme der jeweiligen DLL ist es möglich, von den unterschiedlichsten Anwendungsprogrammen, welche über die Möglichkeit der Einbindung einer DLL verfügen (z.B. Microsoft Access®/Excel®), Messwerte von einem DC-HI-NET einzulesen.

2. Installation

Die auf der Diskette enthaltenen Dateien **DCHINET.DLL** bzw. **DCH_32.DLL** und **DCHINET.INI** sind in das Verzeichnis von Windows zu kopieren.

In der Konfigurationsdatei **DCHINET.INI** müssen für das an eine serielle Schnittstelle (COM1 ... COM99) angeschlossene DC-HI-NET die notwendigen Parameter eingetragen werden. Seit der Version 4.20 können auch zwei DC-HI-NET Boxen parallel betrieben werden. Dies muss jedoch entsprechend konfiguriert werden. Ab Version 4.24 können auch zweistellige COM-Nummern verwendet werden.

Standardmäßig lautet die Einstellung für den *Eingerätebetrieb*:

COM1:9600,n,8,1 (serieller Port COM1, Baudrate 9600, 8 Datenbits, keine Parität, 1 Stopbit)

In der Konfigurationsdatei können darüber hinaus für jeden Kanal und jede Box Einträge für eine Messwerttransformation enthalten sein. Dabei handelt es sich zum einen um einen Wert der Nullpunktverschiebung (Nullungsoffset, s. u.) sowie um einen Faktor (Verstärkungsfaktor). Während der jeweilige Wert für die Nullpunktverschiebung durch den Aufruf einer Treiberfunktion automatisch in die INI-Datei eingetragen werden kann, müssen die Verstärkungsfaktoren grundsätzlich per Texteditor von Hand eingetragen werden.

```
;Konfigurationsdatei für DC-HI-NET Treiber

;Anzahl der angeschlossenen DC-HI-NET Geräte
[DCHINET]
ANZ = 2

[DCHINET 1]
Schnittstelle=COM1:9600,n,8,1

[DCHINET 2]
Schnittstelle=COM2:9600,n,8,1

;Verstärkungsfaktoren für bestimmte Kanäle (Box 1)
[DC1_FAK]
1=10.0

;Verstärkungsfaktoren für bestimmte Kanäle (Box 2)
[DC2_FAK]
3=-0.11

; Nullungsoffsets für bestimmte Kanäle (Box 1)
[DC1_NULL]
1=0.005

; Nullungsoffsets für bestimmte Kanäle (Box 2)
[DC2_NULL]
2=-0.003

...
```

Aufbau der Datei "DCHINET.INI"

DLL Version 4.37

Achtung: Das **XON/XOFF-Protokoll** muss für die benötigte(n) COM-Schnittstelle(n) mit Hilfe der WINDOWS Systemsteuerung unbedingt **ausgeschaltet** werden.

Soll sich die Konfigurationsdatei **DCHINET.INI** für die 16-Bit Version der DLL nicht im Verzeichnis von WINDOWS, sondern in einem beliebigem Verzeichnis befinden, so muss in der Datei **WIN.INI** die Sektion "**DCHINET**" mit dem Eintrag "**INIPATH**", der auf den gültigen Pfad verweist, existieren.

```
...
[DCHINET]
;Parameter für das DC-HI-NET Messdatenerfassungssystem
INIPATH=C:\DCHINET\DLL
...
```

Angabe des Verzeichnisses von "DCHINET.INI" in der Datei "WIN.INI"

3. Funktionen des Treibers

Achtung: Da es seit der Version 4.20 der vorliegenden DLL möglich ist, **zwei** DC-HI-NET Interfaces **parallel** an je einer RS232-Schnittstelle zu betreiben, die Kompatibilität zu bestehenden Anwendungen jedoch beibehalten werden musste, gilt für alle nachstehenden Funktionen, welche eine Kanalnummer als Parameter aufweisen, dass **alle Aufrufe mit einer Kanalnummer ab 1000** auf das zweite DC-HI-NET Interface umgeleitet werden.

Bei Funktionen, welche **keine** Kanalnummer als Parameter erwarten, existiert für das zweite DC-HI-NET Interface jeweils eine **eigene** Funktion. Diese sind an der "2" im vorderen Teil des Funktionsnamens zu erkennen: "**HINET2_...**" bzw. "**SINGLE2_...**"

3.1. Einfache Funktionen

Zur einfachen Messwertübernahme werden folgende Funktionen, die in der 16-Bit Version gemäß der üblichen PASCAL-Aufrufkonvention implementiert wurden, bereitgestellt.

HINET_Version

Diese Funktion dient zum Auslesen des Releasestandes der DC-HI-NET- DLL. Als Parameter muss dabei ein Zeiger auf einen String übergeben werden, in welchem die Informationen geschrieben werden.

	Typ	Bedeutung
Rückgabewert	Integer 32-Bit	Immer 1 (TRUE)
Übergabeparameter (von links beginnend)	Zeichenkette (String) (by reference)	Speicherbereich (Zeichenkette, String) in welchem die Releaseinformationen zurückgegeben werden. Beispiel: "4.13 (vom 30.07.2000)"

DLL Version 4.37

HINET_Init

Initialisiert das/die DC-HI-NET gemäß den Eintragungen in der Konfigurationsdatei. Im Anschluss daran wird (jeweils) eine Nummernvergabe durchgeführt und die Anzahl der adressierbaren Kanäle zurückgeliefert. Bei einem Mehrgerätebetrieb wird die Gesamtanzahl der Kanäle aller DC-HI-NET Boxen zurückgeliefert. Tritt bei der Initialisierung ein Fehler auf, so ist der Rückgabewert 0. Die trifft auch dann zu, wenn dies bei einem Mehrgerätebetrieb mindestens bei einer Box auftritt.

	Typ	Bedeutung
Rückgabewert	Integer 32-Bit	Anzahl der adressierbaren Kanäle bzw. 0 bei einem Fehler

Übergabeparameter (von links beginnend)	keine	
--	-------	--

HINET_Init2**HINET2_Init2**

Mit Hilfe dieser Funktion können die Schnittstellenparameter für die Box 1 bzw. Box2 direkt vom aufrufenden Programm übergeben werden. Der entsprechende Eintrag aus der INI-Datei wird somit nicht ausgewertet. Der übergebene String muss jedoch das gleiche Format aufweisen, wie der Eintrag in der INI-Datei, z.B. "COM1:9600,n,8,1" (Baudrate 9600, keine Parität, 8 Datenbits und 1 Stoppbit).

Zu beachten ist jedoch, dass dieser String nur beim ersten Aufruf der Funktion interpretiert wird, da nur dieses eine Mal die Initialisierung der COM-Schnittstelle erfolgt.

	Typ	Bedeutung
Rückgabewert	Integer 32-Bit	Anzahl der adressierbaren Kanäle bzw. 0 bei einem Fehler

Übergabeparameter (von links beginnend)	Zeichenkette (String) (by reference)	Initialisierungsstring (nullterminiert!) gemäß der COM-Einstellung in der Konfigurationsdatei (vgl. Abschnitt 2)
--	--	--

Über einen (optionalen) Parameter in der Konfigurationsdatei **DCHINET.INI** kann gesteuert werden, ob bei der Initialisierung auch ein Reset des angeschlossenen DC-HI-NET-Interfaces erfolgen soll (vgl. Funktion "**HINET_Reset**"). Ob bei der Ausgabe des RESET-Befehls ein Fehler aufgetreten ist, lässt sich durch Aufruf der Funktion "**HINET_Error**" mit dem Parameter 0 bzw. 1000 als Kanalnummer erfragen. Ein Wert ungleich 0 signalisiert hierbei einen Fehler.

Fehlt der Parameter in der Konfigurationsdatei, so wird kein Reset durchgeführt.

```
;Konfigurationsdatei für DC-HI-NET Treiber
...
[DCHINET 1]
...
RESET = 0
...
[DCHINET 2]
...
RESET = 0
...
```

Konfiguration des Resetbefehls bei der Initialisierung in der Datei "DCHINET.INI"

DLL Version 4.37

HINET_IniFile

Mit Hilfe dieser Funktion kann der Benutzer den Namen (inklusive Pfad) der zu verwendeten INI-Datei angeben (vgl. 2).

	Typ	Bedeutung
Rückgabewert	Integer 32-Bit	TRUE: Datei kann geöffnet werden FALSE: Datei kann nicht geöffnet werden
Übergabeparameter (von links beginnend)	Zeichenkette (String) (by reference)	Name und Pfad (nullterminiert!) der zu verwendenden INI-Datei (z.B. C:\DCHINET\DLL\USER.INI).

HINET_Reset

Durch Aufruf dieser Funktion lässt sich im laufenden Betrieb ein Kanal gezielt zurücksetzen. Das angeschlossene DC-HI-NET quittiert diesen Befehl, um sicherzustellen, dass der Reset auch wirklich erfolgt ist.

Wird als Kanalnummer der Wert "0" übergeben, so wird der RESET-Befehl an alle Kanäle übergeben.

	Typ	Bedeutung
Rückgabewert	Integer 32-Bit 'boolean'	TRUE: Befehl erfolgreich ausgeführt FALSE: Fehler aufgetreten bzw. falsche Kanalnummer
Übergabeparameter (von links beginnend)	Integer 32-Bit (by value)	Nummer des zurückzusetzenden Kanals

Wiederum kann über einen (optionalen) Parameter "TIMEOUT_RESET" in der Konfigurationsdatei **DCHINET.INI** gesteuert werden, wie lange (in Millisekunden) auf das Quittierungssignal gewartet werden soll (s.u.). Fehlt dieser Parameter, so wird standardmäßig ein Timeout von 2000 Millisekunden (2 Sek.) verwendet. Entsprechendes gilt für den Parameter "TIMEOUT", welcher bestimmt, wie lange generell auf eine Antwort vom DCHINET gewartet werden soll.

```
;Konfigurationsdatei für DC-HI-NET Treiber
...
[DCHINET 1]
...
TIMEOUT=2000
TIMEOUT_RESET=2000
...
[DCHINET 2]
...
TIMEOUT=2000
TIMEOUT_RESET=2000
...
```

Konfiguration des Timeoutwertes in Millisekunden Allgemein und für das Warten auf die Quittierung des Resetbefehls in der Datei "DCHINET.INI"

HINET_Exit

Gibt die von der DLL belegte(n) COM-Schnittstelle(n) wieder frei. Um das/die DC-HI-NET erneut ansprechen zu können, muss neu initialisiert werden.

Auch eine ggf. mittels **HINET_IniFile** gesetzte INI-Datei wird wieder neutralisiert und muss erneut erfolgen

	Typ	Bedeutung
Rückgabewert	Integer 32-Bit	Immer 1 (TRUE)
Übergabeparameter	Keine	

HINET_Set0

Adressiert einen DC-HI-NET Kanal, liest den aktuell anliegenden Messwert und interpretiert diesen fortan als neuen Nullpunkt (Koordinatentransformation). Eine eventuell bereits bestehende Nullpunktverschiebung wird zuvor neutralisiert, so dass auf jeden Fall mit einem Absolutwert genullt wird. Ein eventuell vorhandener Verstärkungsfaktor bleibt jedoch berücksichtigt.

Durch den Aufruf dieser Funktion wird ein Kanal per Software genullt, so dass innerhalb der Funktionen **HINET_Akt**, **HINET_Min**, **HINET_Max** und vor der Rückkehr in das aufrufende Programm der eingelesene Messwert korrigiert werden muss (Subtraktion des Offsets). Die Nullungsoffsets werden je Kanal in der Konfigurationsdatei **DCHINET.INI** gespeichert (s.o.) und können dort auch von Hand modifiziert bzw. gelöscht werden.

Beim Aufruf der 4 Funktionen kann über einen Parameter (s.u.) auch angegeben werden, ob die Funktion erst bei Betätigung des Übernahmekontaktes am Messmittel bzw. des Fußtasters den Messwert einlesen und zurückkehren soll.

Wird mit Hilfe des 4. Parameters der Wert "-1" übergeben, so wird der Inhalt des 3. Parameters und nicht der aktuelle Kanalwert als neuer Nullungsoffset interpretiert.

Rückgabewert

Typ	Bedeutung
Integer 32-Bit 'boolean'	TRUE: Nullung erfolgreich FALSE: Fehler aufgetreten bzw. Warten auf Fußtaster unterbrochen

Übergabeparameter
(von links beginnend)

Integer 32-Bit (by value)	Nummer des zu adressierenden Kanals
Double (by reference)	Speicherbereich (8 Byte Fließkommazahl), in welchen der ermittelte Nullungsoffset geschrieben wird (für Kontrollzwecke oder Benutzerinformationen innerhalb des aufrufenden Programms).
Integer 32-Bit 'boolean' (by value)	Flag, das angibt, ob der Wert sofort ausgelesen werden soll ('FALSE', 0), oder erst bei Betätigung des Übernahmekontaktes bzw. des Fußtasters ("TRUE", 1). Wird der Wert "-1" übergeben so wird der Inhalt des Double-Wertes (Parameter 3) als neuer Nullwert gespeichert.

DLL Version 4.37

HINET_Akt
HINET_Min
HINET_Max
HINET_Dif

Durch den Aufruf einer dieser Funktionen wird ein DC-HI-NET Kanal adressiert und je nach Funktion folgender Messwert ermittelt:

- Der aktuelle an dem gewünschten Kanal anstehende Messwert, korrigiert bezüglich der in der INI-Datei definierten Transformation.
- Das Minimum bzw. das Maximum der seit der letzten Initialisierung an dem gewünschten Kanal erfassten Messwerte. Auch diese Werte sind transformiert.
- Die Differenz zwischen Maximum und Minimum.

Bei Aufruf einer der Funktionen kann über einen Parameter (s.u.) auch angegeben werden, ob die Funktion erst bei Betätigung des Übernahmekontaktes am Messmittel bzw. des Fußtasters den Messwert einlesen und zurückkehren soll. Ist dies der Fall, so wird während des Wartens auf den Messwert ständig die Kontrolle an das WINDOWS-System abgeben, damit die anderen Applikationen weiterlaufen können (Multitasking). Das Warten kann mit der Funktion **HINET_Break** (s.u.) unterbrochen werden. In diesem Fall wird der Wert "FALSE" zurückgeliefert.

Sollen die Messwerte von allen Kanälen gleichzeitig übermittelt werden, so ist als Kanalnummer (s.u.) der Wert 0 anzugeben. Der zweite Parameter muss dann allerdings auf ein mindestens der maximalen Kanalanzahl dimensioniertes Array von Double-Variablen zeigen, in das die Messwerte werden (beginnend beim Offset 0!) geschrieben werden.

Das Warten auf den Übernahmekontakt wird dabei ignoriert.

Der Zustand des jeweiligen Fehlerbits des adressierten Kanals bzw. der adressierten Kanäle wird gespeichert und kann im Anschluss mit den Fehlerbit-Funktionen "HINET_Error ()" erfragt werden.

Rückgabewert

Typ	Bedeutung
Integer 32-Bit 'boolean'	TRUE: Messwert erfolgreich eingelesen FALSE: Fehler aufgetreten bzw. Warten auf Fußtaster unterbrochen

Übergabeparameter
(von links beginnend)

Integer 32-Bit (by value)	Nummer des zu adressierenden Kanals bzw. 0, falls die Werte von allen Kanälen ausgelesen werden sollen (s.o.).
Double (by reference)	Speicherbereich (8 Byte Fließkommazahl), in welchen der/die Messwert(e) vom Treiber geschrieben werden soll(en) (s.o.).
Integer 32-Bit "boolean" (by value)	Flag, das angibt, ob der Wert sofort ausgelesen werden soll ("FALSE", 0), oder erst bei Betätigung des Übernahmekontaktes bzw. des Fußtasters ("TRUE", 1). Der Parameter wird ignoriert, falls alle Kanäle adressiert werden (Kanal = 0).

DLL Version 4.37

HINET_Get

Diese Funktion kombiniert die zuvor beschriebenen vier Messwertfunktionen. Die Art des gewünschten Messwertes muss über einen zusätzlichen Parameter angegeben werden.

Der Zustand des jeweiligen Fehlerbits des adressierten Kanals bzw. der adressierten Kanäle wird gespeichert und kann im Anschluss mit den Fehlerbit-Funktionen "HINET_Error ()" erfragt werden.

Rückgabewert

Typ	Bedeutung
Integer 32-Bit 'boolean'	TRUE: Messwert erfolgreich eingelesen FALSE: Fehler aufgetreten bzw. Warten auf Fußtaster unterbrochen

Übergabeparameter
(von links beginnend)

Integer 32-Bit (by value)	Art des gewünschten Messwertes: aktueller (0), maximaler (1), minimaler (2) Messwert bzw. die Differenz zwischen Maximum und Minimum (3).
Integer 32-Bit (by value)	Nummer des zu adressierenden Kanals bzw. 0, falls die Werte von allen Kanälen ausgelesen werden sollen (s.o.).
Double (by reference)	Speicherbereich (8 Byte Fließkommazahl), in welchen der/die Messwert(e) vom Treiber geschrieben werden soll(en) (s.o.).
Integer 32-Bit "boolean" (by value)	Flag, das angibt, ob der Wert sofort ausgelesen werden soll ('FALSE', 0), oder erst bei Betätigung des Übernahmekontaktes bzw. des Fußtasters ("TRUE", 1). Der Parameter wird ignoriert, falls alle Kanäle adressiert werden (Kanal = 0).

HINET_Get3

Diese Funktion entspricht im Wesentlichen der Funktion "HINET_Get ()". Der oder die Messwerte werden hierbei jedoch als String (nullterminiert!) zurückgegeben.

Der Zustand des jeweiligen Fehlerbits des adressierten Kanals bzw. der adressierten Kanäle wird gespeichert und kann im Anschluss mit den Fehlerbit-Funktionen "HINET_Error ()" erfragt werden.

Rückgabewert

Typ	Bedeutung
Integer 32-Bit	Anzahl der gültigen Zeichen im Rückgabestring bzw. 0 im Fehlerfall.

Übergabeparameter
(von links beginnend)

Integer 32-Bit (by value)	Art des gewünschten Messwertes: aktueller (0), maximaler (1), minimaler (2) Messwert bzw. die Differenz zwischen Maximum und Minimum (3).
Integer 32-Bit (by value)	Nummer des zu adressierenden Kanals bzw. 0, falls die Werte von allen Kanälen ausgelesen werden sollen (s.o.).
Zeichenkette (String) (by reference)	Speicherbereich (Zeichenkette, String) in welchem der bzw. die Messwerte - als String kodiert - zurückgegeben werden. Werden mehr als ein Messwert übergeben, so werden diese durch ein Leerzeichen (Blank) voneinander getrennt. Beispiel für 3 Messwerte: "0.123 1.234 2.345"
Integer 32-Bit "boolean" (by value)	Flag, das angibt, ob der Wert sofort ausgelesen werden soll ("FALSE", 0), oder erst bei Betätigung des Übernahmekontaktes bzw. des Fußtasters ("TRUE", 1). Der Parameter wird ignoriert, falls alle Kanäle adressiert werden (Kanal = 0).

DLL Version 4.37

HINET_Get4

Diese Funktion ermittelt einen bestimmten DC-HI-NET- Messwert und liefert diesen direkt als Funktionsergebnis zurück. Die erforderlichen Parameter werden dabei als String in der Form "Messwertart, Kanal, Warten auf Fußtaster" übergeben.

Der Zustand des jeweiligen Fehlerbits des adressierten Kanals bzw. der adressierten Kanäle wird gespeichert und kann im Anschluss mit den Fehlerbit-Funktionen "HINET_Error ()" erfragt werden.

	Typ	Bedeutung
Rückgabewert	Double	Wert des adressierten Kanals bzw. MAXDOUBLE (1.7E+308) im Fehlerfall..
Übergabeparameter (von links beginnend)	Zeichenkette (String) (by reference)	<p>Übergibt die Art des Messwertes, den betroffenen Kanal und das Warteverhalten auf den Fußtasterkontakt in einem String (s.o.). Die einzelnen Parameter (Bedeutung vgl. auch "HINET_Get ()") werden dabei durch ein Komma voneinander getrennt.</p> <p>Beispiel für aktuellen Messwert von Kanal 3 mit Warten auf Fußtaster: "0, 3, 1"</p> <p>Beispiel für aktuellen Messwert von Kanal 1 des <u>zweiten</u> Gerätes ohne Warten auf Fußtaster: "0, 1003, 0"</p>

Über einen (optionalen) Parameter in der Konfigurationsdatei **DCHINET.INI** kann gesteuert werden, ob auf einen vom DC-HI-NET erhaltenen Messwert eine Bestätigung ("Acknowledge") an das DC-HI-NET zurückgesandt werden soll. Standardmäßig ist dies ausgeschaltet (Fehlen des Parameters in der Konfigurationsdatei).

```
;Konfigurationsdatei für DC-HI-NET Treiber
...
[DCHINET 1]
...
ACK = 0
...
[DCHINET 2]
...
ACK = 0
...
```

Konfiguration eines "Acknowledge-Handshakes" beim Erhalt eines Messwerts in der Datei "DCHINET.INI"

Soll ein "Acknowledge" erfolgen, so ist der Wert "1" einzutragen, andernfalls der Wert "0".

Achtung: Dieses Feature ist nur bei bestimmten DC-HI-NET- Geräten implementiert.

DLL Version 4.37

HINET_Error

Diese Funktion ermittelt, ob bei einem vorhergehenden Auslesen von Messwerten (s.o.) bei den betroffenen Kanälen jeweils ein Fehlerbit gesetzt war. Wird als Kanalnummer dabei der Wert "0" übergeben, so gibt die Funktion die absolute Anzahl aller zuletzt durch ein gesetztes Fehlerbit betroffenen Kanäle zurück.

	Typ	Bedeutung
Rückgabewert	Integer 32-Bit "boolean" bzw. "numerisch"	TRUE: Fehlerbit beim Kanal gesetzt FALSE: kein Fehlerbit gesetzt Anzahl der Kanäle mit Fehlerbit bei Kanalnummer 0 (s.o.).
Übergabeparameter (von links beginnend)	Integer 32-Bit (by value)	Nummer des zu adressierenden Kanals

HINET_Error2

Auch diese Funktion ermittelt für alle Kanäle einer DC-HI-NET Box, ob bei einem vorhergehenden Auslesen von Messwerten (s.o.) jeweils ein Fehlerbit gesetzt war.

HINET2_Error2

Sie schreibt die Informationen dabei je Kanal fortlaufend in ein Array (beginnend beim Offset 0!), dessen Adresse als Parameter übergeben werden muss. Dieses Array aus 32-Bit Integer-Zahlen muss dabei ausreichend dimensioniert sein. Wurde bei einem Kanal ein Fehlerbit gesetzt, so wird an der entsprechenden Stelle des Arrays der Wert "1" gespeichert, ansonsten "0".

	Typ	Bedeutung
Rückgabewert	Integer 32-Bit	Anzahl der Kanäle mit Fehlerbit
Übergabeparameter (von links beginnend)	Array von Integer 32-Bit (by reference)	Adresse des ausreichend dimensionierten Arrays zum Speichern der Fehlerbitinformationen je Kanal.

HINET_Error2a

Diese Funktion entspricht der vorhergehenden Funktion "HINET_Error2 ()" bzw. "HINET2_Error2 ()". Im Unterschied dazu werden hier jedoch die Fehlerbitinformationen der einzelnen Kanäle einer DC-HI-NET Box in einem String zurückgegeben. Für ein gesetztes Fehlerbit wird eine "1" in den String geschrieben, ansonsten eine "0".

HINET2_Error2a

	Typ	Bedeutung
Rückgabewert	Integer 32-Bit	Anzahl der gültigen Zeichen im Rückgabestring bzw. 0 im Fehlerfall.
Übergabeparameter (von links beginnend)	Zeichenkette (String) (by reference)	Speicherbereich (Zeichenkette, String) in welchem die Fehlerbitinformationen der einzelnen Kanäle - als String kodiert - zurückgegeben werden. Die Fehlerbitstati für die Kanäle werden durch ein Leerzeichen (Blank) voneinander getrennt. Beispiel für 6 Kanäle: "0 1 0 1 1 0"

DLL Version 4.37

HINET_DeMinMax Diese Funktion dient zum Löschen des für den angegebenen DC-HI-NET Kanal gespeicherten minimalen und maximalen Messwerts.

	Typ	Bedeutung
Rückgabewert	Integer 32-Bit 'boolean'	TRUE: Messwert erfolgreich eingelesen FALSE: Fehler aufgetreten
Übergabeparameter (von links beginnend)	Integer 32-Bit (by value)	Nummer des zu adressierenden Kanals

HINET_Trigger Diese Funktion dient zum Feststellen, ob beim zuletzt abgeholten Messwert eines Kanals der Fußtaster bzw. Übernahmekontakt betätigt wurde.

	Typ	Bedeutung
Rückgabewert	Integer 32-Bit 'boolean'	TRUE: Fußtaster betätigt FALSE: Kein Fußtaster betätigt
Übergabeparameter (von links beginnend)	Integer 32-Bit (by value)	Nummer des zu adressierenden Kanals

HINET_Break Durch den Aufruf dieser Funktion kann das Warten auf Betätigung des Fußtasters bzw. Übernahmekontaktes unterbrochen werden. Die wartende Funktion (**HINET_Akt**, **HINET_Min**, **HINET_Max**, **HINET_Dif**, **HINET_Get** oder **HINET_Set0**) kehrt mit dem Wert "FALSE" zurück.

	Typ	Bedeutung
Rückgabewert	Integer 32-Bit 'boolean'	TRUE: Fußtaster betätigt FALSE: Kein Fußtaster betätigt
Übergabeparameter (von links beginnend)	keine	

DLL Version 4.37

3.2. Funktionen zur ereignisgesteuerten Messwerterfassung

Neben den im vorherigen Abschnitt beschriebenen einfachen Funktionen zur Messwerterfassung stehen auch Funktionen zur ereignisgesteuerten Messwerterfassung zur Verfügung. "Ereignisgesteuert" bedeutet in diesem Fall, dass das Hauptprogramm, das die DLL verwendet, den Start einer Messwerterfassung initiieren kann und anschließend mittels einer Meldung über den Erhalt des Messwertes benachrichtigt wird. Diese Art der Messwerterfassung hat den Vorteil, dass sie asynchron und völlig im Hintergrund abläuft, so dass sich das Hauptprogramm mit anderen Arbeiten beschäftigen kann, während z.B. auf die Betätigung des Übernahmekontaktes gewartet wird.

Während einer ereignisgesteuerten Messwerterfassung wird auf dem Desktop ein zu einem Icon verkleinertes Fenster mit dem Titel "DCHINET.DLL" angezeigt. Dieses darf auf keinen Fall durch den Taskmanager oder ähnliche Programme geschlossen werden. Ist die ereignisgesteuerte Messwerterfassung beendet, wird das Fenster automatisch wieder geschlossen.

Folgende Funktionen stehen für die ereignisgesteuerte Messwerterfassung zur Verfügung:

HINET_StartMess

Diese Funktion leitet für einen bestimmten DC-HI-NET Kanal eine ereignisgesteuerte Messwerterfassung ein. Über einen Parameter wird die Art des gewünschten Messwerts (AKT, MIN, MAX, DIF) angegeben. Die Funktion kehrt nach dem Start der Messung sofort zurück. Wenn der gewünschte Messwert vom DC-HI-NET übergeben bzw. der Übernahmekontakt (Fußtaster) betätigt wurde, erhält das Hauptprogramm die Nachricht **WM_DCHINET** (definiert als "WM_USER + 99") und kann den gültigen Messwert mit der Funktion **HINET_Get2** ermitteln.

	Typ	Bedeutung
Rückgabewert	Integer 32-Bit "boolean"	TRUE: Messwerterfassung erfolgreich gestartet FALSE: Messwerterfassung konnte nicht gestartet werden
Übergabeparameter (von links beginnend)	Window-Handle (by value)	Handle des Fensters, an das die Nachricht WM_DCHINET über den Erhalt des gewünschten Messwerts gesendet werden soll. Das zugehörige Fenster muss auf diese Nachricht reagieren können und die Funktion HINET_Get2 zum Auslesen des Messwerts aufrufen.
	Integer 32-Bit (by value)	Art des gewünschten Messwertes: aktueller (0), maximaler (1), minimaler (2) Messwert bzw. die Differenz zwischen Maximum und Minimum (3).
	Integer 32-Bit (by value)	Nummer des zu adressierenden Kanals
	Integer 32-Bit "boolean" (by value)	Wird dieses Flag auf den Wert "TRUE" (oder 1) gesetzt, werden von der DLL ständig Messwerte vom DC-HI-NET abgefragt und dem Hauptprogramm die Nachricht WM_DCHINET erst bei Betätigung des Übernahmekontaktes bzw. des Fußtasters übermittelt. Ansonsten (Parameter gleich "FALSE" oder 0) wird das Hauptprogramm unmittelbar nach Erhalt des ersten Messwerts vom gewünschten Kanals benachrichtigt.

HINET_StopMess

Durch den Aufruf dieser Funktion wird eine laufende ereignisgesteuerte Messung abgebrochen.

	Typ	Bedeutung
Rückgabewert	Integer 32-Bit 'boolean'	TRUE: Messung beendet FALSE: Messung konnte nicht beendet werden bzw. keine ereignisgesteuerte Messung aktiv.
Übergabeparameter	keine	

DLL Version 4.37

HINET_Get2

Diese Funktion dient zum Ermitteln des aktuell eingelesenen Messwerts von dem Kanal, der durch **HINET_StartMess** für eine ereignisgesteuerte Messung vorgesehen wurde. Im Normalfall wird **HINET_Get2** nach Erhalt der Meldung **WM_DCHINET** aufgerufen. Der Aufruf ist jedoch auch während des Wartens auf Betätigung des Übernahmekontaktes bzw. des Fußtasters möglich. In diesem Fall wird der zuletzt eingelesene Messwert übergeben. Damit können z.B. Online-Anzeigen realisiert werden.

Jedoch erst der dem Erhalt der Meldung **WM_DCHINET** nachfolgende Aufruf ermittelt den eigentlich gewünschten Messwert.

	Typ	Bedeutung
Rückgabewert	Integer 32-Bit "boolean"	TRUE: Messwert gültig FALSE: Fehler aufgetreten, Messwert nicht gültig
Übergabeparameter (von links beginnend)	Double (by reference)	Speicherbereich (8 Byte Fließkommazahl), in welchen der Messwert übergeben werden soll.

3.3. Funktionen zur Definition von Kanalwertberechnungen und Kanalverknüpfungen

Jeden DC-HI-NET Kanal können Formeln zur Berechnung bzw. mathematischer Manipulation des zugehörigen Messwerts zugeordnet werden. Mit diesen Formeln lassen sich auch auf einfache Weise Kanalverknüpfungen realisieren. In diesem Zusammenhang sind Kanalnummern bis 99 adressierbar, wobei die Kanäle ab 80 sog. *virtuelle Kanäle* darstellen. Die Ergebniswerte dieser virtuellen Kanäle können nur mit Formeln berechnet werden, da das DC-HI-NET nur maximal 78 Kanäle unterstützt. Die Werte der virtuellen Kanäle können wie Werte "normaler" Kanäle ermittelt werden (siehe Abschnitt 3.1.). Zu Beachten ist jedoch, dass bei Kanalnummern größer als 80 **nur aktuelle Messwerte** und keine Minima bzw. Maxima ermittelt werden können. Wird z.B. der MIN-Wert eines mittels einer Formel berechneten Kanals bis 80 angefordert, so kommt als Kanalwert in der Formel ebenfalls der MIN-Wert als Berechnungsgrundlage zur Anwendung.

Die Formeln zur Kanalwertberechnung haben folgenden Aufbau:

K<Kanalnummer> := <Ausdruck>

Beispiele: "K80 := K1 * K2"

Der Wert von Kanal Nr. 80 ergibt sich aus dem Produkt von Kanal 1 und Kanal 2

"K81 := (K1 + 2.01)/(K3 - 1)"

Der Wert von Kanal Nr. 81 ergibt sich aus der Division von Kanal 1 minus der Konstante 2.01 und Kanal 3 minus der Konstanten 1.

Achtung: Zur Vermeidung redundanter Kanalverknüpfungen wird auf der rechten Seite der Formel bei Kanalnummern unter 80 der direkte Wert vom DC-HI-NET zur Berechnung herangezogen und nicht eine etwaige Kanalverknüpfung.

"K1 := 10 * K1 + 2.0"

Der unmittelbare Wert von Kanal 1 wird zunächst mit 10 multipliziert und anschließend um die Konstante 2.0 erhöht.

Achtung: Bei einem Mehrgerätebetrieb dürfen bei der *Definition* von Kanalverknüpfungen keine Kanalwerte ab 1000 verwendet werden. Soll jedoch der Wert des (virtuellen) Kanals 97 der zweiten DC-HI-NET Box ermittelt werden, so ist als Kanalnummer jedoch die "1097" zu übergeben.

Mit Hilfe der folgenden Funktionen können Kanalwertberechnungsformel definiert werden:

DLL Version 4.37

HINET_DefChan**HINET2_DefChan**

Mit Hilfe dieser Funktion kann der DLL eine Formel zur Berechnung eines DC-HI-NET Kanals einer bestimmten Box übergeben werden. Wurde eine syntaktisch korrekte Formel übergeben, so wird diese dem entsprechenden Kanal zugeordnet aber nicht in der Konfigurationsdatei gespeichert, so dass sie beim nächsten Start der DLL auch nicht wieder zur Verfügung steht.

	Typ	Bedeutung
Rückgabewert	Integer 32-Bit "boolean"	TRUE: Formel gespeichert und zugeordnet FALSE: Fehlerhafte Formel
Übergabeparameter (von links beginnend)	Zeichenkette (by reference)	Zeichenkette (String), welche die Formel zur Kanalwertberechnung darstellt. Eine für den betroffenen Kanal bereits existierende Formel wird dabei überschrieben. Wird für einen Kanal 'x' ein Formelstring der Form "Kx :=", d.h. keine Berechnungsvorschrift, übergeben, so wird eine eventuell vorhandene Formel für diesen Kanal wieder gelöscht.

Sollen Formeln zur Kanalwertberechnung dauerhaft gespeichert werden und bei jedem Neustart der DLL wieder zur Verfügung stehen, so müssen sie direkt in der Konfigurationsdatei **DCHINET.INI** hinterlegt werden. Hierin befindliche Formeln werden bei jedem Neustart der DLL sofort ausgewertet und den entsprechenden Kanälen zugeordnet.

Die Kanalwertberechnungsformeln müssen in der Datei **DCHINET.INI** dabei für jede Box in einer eigenen Sektion "[DC1_DEF]" bzw. "[DC2_DEF]" mit fortlaufenden Nummern definiert werden.

```

...
[DC1_DEF]
;Formeln zur Kanalwertberechnung (z.B. Kanalverknüpfungen)
1="K80 := K1 * K2"
2="K81 := (K1 + 2.01)/(K3 - 1)"
3="K1 := 10 * K1 + 2.0"
...
[DC2_DEF]
;Formeln zur Kanalwertberechnung (z.B. Kanalverknüpfungen)
1="K80 := K1 / K2"
2="K97 := K80 * K80"
...

```

Definition der Formeln zur Kanalwertberechnung in der Datei "DCHINET.INI"

Achtung: Die ereignisgesteuerte Messmethode (vgl. Abschnitt 3.2) ist nicht auf einen Kanal anwendbar, für den eine Kanalverknüpfung definiert wurde. Es wird hierbei immer der "echte" Wert des DC-HI-NET Kanals ermittelt.

3.4. Funktionen zur Ansteuerung des DC-HI-NET Steckmoduls "DIG-I/O"

HINET_DigiOut

Mit dieser Funktion können digitale Ausgaben auf das DC-HI-NET Steckmodul "DIG-I/O" durchgeführt werden. Diese Karte besitzt 4 Gruppen zu je 4 Steuerleitungen (vgl. Handbuch Steckmodul „DIG-I/O“), was einer Gesamtanzahl von 16 Steuerleitungen entspricht. Die gewünschte Gruppe wird über den zweiten Parameter, das/die zu setzende(n) Bit(s) über den dritten Parameter angegeben.

Rückgabewert

Typ	Bedeutung
Integer 32-Bit "boolean"	TRUE: Ausgabe erfolgt FALSE: Fehler von DC-HI-NET (Fehlerbit)

Übergabeparameter
(von links beginnend)

Integer 32-Bit (by value)	Nummer des DIG-I/O Kanals
Integer 32-Bit "unsigned" (by value)	Nummer der gewünschten Gruppe.
Integer 32-Bit "unsigned" (by value)	Zu setzende Ausgabeports der angegebenen Gruppe. Die max. 4 betroffenen Ausgangsleitungen werden über das untere Halbbyte (Bits 1 ... 4) codiert, wobei jedes gesetzte Bit einer geschalteten Ausgangsleitung entspricht.

HINET_DigIn

Mit dieser Funktion können von einem in das DC-HI-NET eingebauten Steckmodul "DIG-I/O" 16 Eingangssignale ausgewertet werden. Das Ergebnis der Abfrage wird in eine 16 Bit große Maske (zweiter Parameter) geschrieben. Jedes gesetzte Bit in der Maske stellt dabei einen aktiven Eingang dar.

Wird mittels des dritten Parameter der Wert "TRUE" übergeben, so wird der Inhalt des zweiten Parameters als Bitmaske interpretiert, welche den Eingangszuständen entsprechen soll. Die Funktion kehrt dabei erst dann zurück, wenn die Eingangszustände der Maske entsprechen.

Rückgabewert

Typ	Bedeutung
Integer 32-Bit "boolean"	TRUE: kein Fehler ist aufgetreten, der Wert der Maske ist gültig FALSE: Fehler von DC-HI-NET (Fehlerbit)

Übergabeparameter
(von links beginnend)

Integer 32-Bit (by value)	Nummer des DIG-I/O Kanals
Integer 32-Bit (by reference)	Zeiger auf ein 16 Bit breites Bitfeld (Funktion s.o.).
Integer 32-Bit "boolean" (by value)	Wird der Wert "TRUE" (= 1) übergeben, so wird der zweite Parameter als Bitmaske, welche den Eingangszuständen entsprechen soll, interpretiert. Es wird dann solange gewartet, bis die Eingangszustände der Maske entsprechen.

DLL Version 4.37

3.5. Spezielle DC-HI-NET-Funktionen

HINET_Special

Diese Funktion dient zur Übermittlung spezieller Befehlsfolgen an besonders angepasste DC-HI-NET- Geräte.

	Typ	Bedeutung
Rückgabewert	Integer 32-Bit 'boolean'	TRUE: Funktion erfolgreich übermittelt FALSE: Fehler aufgetreten
Übergabeparameter (von links beginnend)	Integer 32-Bit (by value)	Nummer der Sonderfunktion (zwischen 1 und 96)
	Integer 32-Bit (by value)	Nummer des zu adressierenden Kanals
	Integer 32-Bit (by value)	Parameterwert der Sonderfunktion (zwischen 0 und 127)

HINET_Special2

Diese Funktion dient zur Übermittlung spezieller Befehlsfolgen an besonders angepasste DC-HI-NET- Geräte.

	Typ	Bedeutung
Rückgabewert	Integer 32-Bit 'boolean'	TRUE: Funktion erfolgreich übermittelt FALSE: Fehler aufgetreten
Übergabeparameter (von links beginnend)	Integer 32-Bit (by value)	Nummer der Sonderfunktion (zwischen 1 und 96)
	Integer 32-Bit (by value)	Nummer des zu adressierenden Kanals
	Zeichenkette (String) (by reference)	Adresse des Puffers (Zeichenkette, String) in welchem sich die auszugebenden Daten befinden.
	Integer 32-Bit (by value)	Anzahl der Zeichen im Puffer (zwischen 0 und 126)

DLL Version 4.37

3.6. Spezielle Funktionen zur Ansteuerung eines DC-HI-NET "SINGLE"

SINGLE_Version

Diese Funktion ermittelt der Softwareversion eines angeschlossenen DC-HI-NET "SINGLE".

SINGLE2_Version

	Typ	Bedeutung
Rückgabewert	Integer 32-Bit "boolean"	TRUE: Funktion erfolgreich übermittelt FALSE: Fehler aufgetreten, evtl. kein SINGLE angeschlossen
Übergabeparameter (von links beginnend)	Zeichenkette (String) (by reference)	Adresse des Puffers, der bei erfolgreicher Funktionsausführung die Softwareversion in Form eines nullterminierten Strings enthält. Der Größe des Puffers muss dabei mindestens 33 Byte entsprechen. Beispiel: "S6F TEST 4=15-26 5=5-10 6=11-14"

SINGLE_Revision

Diese Funktion ermittelt der Softwarerevision eines angeschlossenen DC-HI-NET SINGLE.

SINGLE2_Revision

	Typ	Bedeutung
Rückgabewert	Integer 32-Bit "boolean"	TRUE: Funktion erfolgreich übermittelt FALSE: Fehler aufgetreten, evtl. kein SINGLE angeschlossen
Übergabeparameter (von links beginnend)	Zeichenkette (String) (by reference)	Adresse des Puffers, der bei erfolgreicher Funktionsausführung die Softwarerevision in Form eines nullterminierten Strings enthält. Der Größe des Puffers muss dabei mindestens 20 Byte entsprechen. Beispiel: "V3.01 vom 01.08.01"

SINGLE_Release

Diese Funktion ermittelt der Softwarerelease eines angeschlossenen DC-HI-NET "SINGLE".

SINGLE2_Release

	Typ	Bedeutung
Rückgabewert	Integer 32-Bit "boolean"	TRUE: Funktion erfolgreich übermittelt FALSE: Fehler aufgetreten, evtl. kein SINGLE angeschlossen
Übergabeparameter (von links beginnend)	Zeichenkette (String) (by reference)	Adresse des Puffers, der bei erfolgreicher Funktionsausführung die Softwarerelease in Form eines nullterminierten Strings enthält. Der Größe des Puffers muss dabei mindestens 20 Byte entsprechen. Beispiel: "Rel 999"

DLL Version 4.37

SINGLE_SerialNr

Diese Funktion ermittelt die Seriennummer eines angeschlossenen DC-HI-NET "SINGLE".

SINGLE2_SerialNr

	Typ	Bedeutung
Rückgabewert	Integer 32-Bit "boolean"	TRUE: Funktion erfolgreich übermittelt FALSE: Fehler aufgetreten, evtl. kein SINGLE angeschlossen
Übergabeparameter (von links beginnend)	Zeichenkette (String) (by reference)	Adresse des Puffers, der bei erfolgreicher Funktionsausführung die Seriennummer in Form eines nullterminierten Strings enthält. Der Größe des Puffers muss dabei mindestens 20 Byte entsprechen. Beispiel: "300900-0769"

SINGLE_GetRS232

Diese Funktion ermittelt die RS232-Parameter der physikalischen Eingänge eines angeschlossenen DC-HI-NET "SINGLE".

SINGLE2_GetRS232

	Typ	Bedeutung
Rückgabewert	Integer 32-Bit "boolean"	TRUE: Funktion erfolgreich übermittelt FALSE: Fehler aufgetreten, evtl. kein SINGLE angeschlossen
Übergabeparameter (von links beginnend)	Integer 32-Bit (by value)	Nummer des gewünschten Eingangs (Anzahl abhängig vom SINGLE-Typ).
	Integer 32-Bit (by reference)	Zeiger auf eine 32-Bit Integervariable, zur Speicherung der aktuellen Baudrate des gewünschten Eingangs.
	Integer 32-Bit (by reference)	Zeiger auf eine 32-Bit Integervariable, zur Speicherung der aktuellen Anzahl der Datenbits des gewünschten Eingangs.
	Integer 32-Bit (by reference)	Zeiger auf eine 32-Bit Integervariable, zur Speicherung der aktuellen Anzahl der Stoppbits des gewünschten Eingangs.
	Integer 32-Bit (by reference)	Zeiger auf eine 32-Bit Integervariable, zur Speicherung der aktuellen Parität des gewünschten Eingangs: 0 keine, 1 ungerade und 2 gerade Parität.

SINGLE_GetRS232Host

Diese Funktion ermittelt die Parameter des RS232-Anschlusses zum Host eines angeschlossenen DC-HI-NET "SINGLE".

SINGLE2_GetRS232Host

	Typ	Bedeutung
Rückgabewert	Integer 32-Bit "boolean"	TRUE: Funktion erfolgreich übermittelt FALSE: Fehler aufgetreten, evtl. kein SINGLE angeschlossen
Übergabeparameter (von links beginnend)	Integer 32-Bit (by reference)	Zeiger auf eine 32-Bit Integervariable, zur Speicherung der aktuellen Baudrate des Hostanschlusses.

DLL Version 4.37**SINGLE_
GetParameter**

Diese Funktion ermittelt den Wert eines bestimmten Parameters eines physikalischen Eingangs eines angeschlossenen DC-HI-NET "SINGLE".

	Typ	Bedeutung
Rückgabewert	Integer 32-Bit "boolean"	TRUE: Funktion erfolgreich übermittelt FALSE: Fehler aufgetreten, evtl. kein SINGLE angeschlossen
Übergabeparameter (von links beginnend)	Integer 32-Bit (by value)	Nummer des gewünschten Eingangs (Anzahl abhängig vom SINGLE-Typ).
	Integer 32-Bit (by Value)	Nummer des entsprechenden Parameters: 1 bis 4
	Integer 32-Bit (by reference)	Zeiger auf eine 32-Bit Integervariable, zur Speicherung des aktuellen Parameterwertes des gewünschten Eingangs.

DLL Version 4.37

4. Aufruf der DLL-Funktionen aus C-Programmen

4.1. Prototypen der DLL-Funktionen

extern long FAR PASCAL HINET_Version	(char FAR *lpszVersion);
extern long FAR PASCAL HINET_Init	(void);
extern long FAR PASCAL HINET_Init2	(char FAR *lpszComIni);
extern long FAR PASCAL HINET2_Init2	(char FAR *lpszComIni);
extern long FAR PASCAL HINET_IniFile	(char FAR *lpszIniFile);
extern long FAR PASCAL HINET_Exit	(void);
extern long FAR PASCAL HINET_Reset	(long nChan);
extern long FAR PASCAL HINET_Get	(long nType, long nChan, double FAR *lpdftVal, long fWait);
extern long FAR PASCAL HINET_Get3	(long Type, long nChan, char FAR *lpszVal, long fWait);
extern double FAR PASCAL HINET_Get4	(char FAR *lpszPara);
extern long FAR PASCAL HINET_Akt	(long nChan, double FAR *lpdftVal, long fWait);
extern long FAR PASCAL HINET_MIN	(long nChan, double FAR *lpdftVal, long fWait);
extern long FAR PASCAL HINET_MAX	(long nChan, double FAR *lpdftVal, long fWait);
extern long FAR PASCAL HINET_Dif	(long nChan, double FAR *lpdftVal, long fWait);
extern long FAR PASCAL HINET_Set0	(long nChan, double FAR *lpdftVal, long fWait);
extern long FAR PASCAL HINET_Error	(long nChan);
extern long FAR PASCAL HINET_Error2	(long FAR *anChanError);
extern long FAR PASCAL HINET2_Error2	(long FAR *anChanError);
extern long FAR PASCAL HINET_Error2a	(char FAR *lpszChanError);
extern long FAR PASCAL HINET2_Error2a	(char FAR *lpszChanError);
extern long FAR PASCAL HINET_DelMinMax	(long nChan);
extern long FAR PASCAL HINET_Trigger	(long nChan);
extern long FAR PASCAL HINET_Break	(void);
extern long FAR PASCAL HINET_StartMess	(HWND hwMainProg, long Type, long nChan, long fWait);
extern long FAR PASCAL HINET_StopMess	(void);
extern long FAR PASCAL HINET_Get2	(double FAR *lpdftVal);
extern long FAR PASCAL HINET_DefChan	(char far *lpszDef);
extern long FAR PASCAL HINET2_DefChan	(char far *lpszDef);
extern long FAR PASCAL HINET_DigiOut	(long nChan, WORD uGroup, WORD uBits);
extern long FAR PASCAL HINET_DigIn	(long nChan, WORD far *uMask, BOOL fWait);
extern long FAR PASCAL HINET_Get3	(long nType, long nChan, char FAR *lpszVal, long fWait);
extern long FAR PASCAL HINET_Special	(long nFunc, long nChan, long nParam);
extern long FAR PASCAL HINET_Special2	(long nFunc, long nChan, char FAR *lpszBuffer, long BytesInBuffer);
extern long FAR PASCAL SINGLE_Version	(char FAR *lpszVersion);
extern long FAR PASCAL SINGLE2_Version	(char FAR *lpszVersion);
extern long FAR PASCAL SINGLE_Revision	(char FAR *lpszRevision);
extern long FAR PASCAL SINGLE2_Revision	(char FAR *lpszRevision);
extern long FAR PASCAL SINGLE_Release	(char FAR *lpszRelease);
extern long FAR PASCAL SINGLE2_Release	(char FAR *lpszRelease);
extern long FAR PASCAL SINGLE_SerialNr	(char FAR *lpnSerialNr);
extern long FAR PASCAL SINGLE2_SerialNr	(char FAR *lpnSerialNr);
extern long FAR PASCAL SINGLE_GetRS232	(long nChan, long FAR *lpnBaudrate, long FAR *lpnDatabits, long FAR *lpnStopbits, long FAR *lpnParity);

DLL Version 4.37

```
extern long FAR PASCAL SINGLE2_GetRS232 (long nChan, long FAR *lpnBaudrate,
                                         long FAR *lpnDatabits, long FAR *lpnStopbits,
                                         long FAR *lpnParity);
extern long FAR PASCAL SINGLE_GetRS232Host (long FAR *lpnBaudrate);
extern long FAR PASCAL SINGLE_GetParameter (long nChan, long nParaNr, long FAR *lpnParaVal);
```

4.2. Definitionen

#define WM_DCHINET

WM_USER + 99 // Messagekennung der DLL

4.3. Deklarationen in der Moduldefinitionsdatei für MS-WINDOWS Programme

16-Bit	32-Bit
IMPORTS	IMPORTS
DCHINET.HINET_Version	DCH_32.HINET_Version
DCHINET.HINET_Init	DCH_32.HINET_Init
DCHINET.HINET_Init2	DCH_32.HINET_Init2
DCHINET.HINET2_Init2	DCH_32.HINET2_Init2
DCHINET.HINET_Exit	DCH_32.HINET_Exit
DCHINET.HINET_Reset	DCH_32.HINET_Reset
DCHINET.HINET_Get	DCH_32.HINET_Get
DCHINET.HINET_Get2	DCH_32.HINET_Get2
DCHINET.HINET_Get3	DCH_32.HINET_Get3
DCHINET.HINET_Akt	DCH_32.HINET_Akt
DCHINET.HINET_Min	DCH_32.HINET_Min
DCHINET.HINET_Max	DCH_32.HINET_Max
DCHINET.HINET_Dif	DCH_32.HINET_Dif
DCHINET.HINET_Set0	DCH_32.HINET_Set0
DCHINET.HINET_Error	DCH_32.HINET_Error
DCHINET.HINET_Error2	DCH_32.HINET_Error2
DCHINET.HINET2_Error2	DCH_32.HINET2_Error2
DCHINET.HINET_Error2a	DCH_32.HINET_Error2a
DCHINET.HINET2_Error2a	DCH_32.HINET2_Error2a
DCHINET.HINET_DeMinMax	DCH_32.HINET_DeMinMax
DCHINET.HINET_Trigger	DCH_32.HINET_Trigger
DCHINET.HINET_Break	DCH_32.HINET_Break
DCHINET.HINET_StartMess	DCH_32.HINET_StartMess
DCHINET.HINET_StopMess	DCH_32.HINET_StopMess
DCHINET.HINET_Get2	DCH_32.HINET_Get2
DCHINET.HINET_DefChan	DCH_32.HINET_DefChan
DCHINET.HINET2_DefChan	DCH_32.HINET2_DefChan
DCHINET.HINET_DigiOut	DCH_32.HINET_DigiOut
DCHINET.HINET_DigiIn	DCH_32.HINET_DigiIn
DCHINET.HINET_Spezial	DCH_32.HINET_Spezial
DCHINET.HINET_Spezial2	DCH_32.HINET_Spezial2
DCHINET.SINGLE_Version	DCH_32.SINGLE_Version
DCHINET.SINGLE2_Version	DCH_32.SINGLE2_Version
DCHINET.SINGLE_Revision	DCH_32.SINGLE_Revision

DLL Version 4.37

16-Bit

IMPORTS

DCHINET.SINGLE2_Revision
DCHINET.SINGLE_Release
DCHINET.SINGLE2_Release

DCHINET.SINGLE_SerialNr
DCHINET.SINGLE2_SerialNr
DCHINET.SINGLE_GetRS232
DCHINET.SINGLE_GetRS232Host
DCHINET.SINGLE2_GetRS232Host
DCHINET.SINGLE_GetParameter

32-Bit

IMPORTS

DCH_32.SINGLE2_Revision
DCH_32.SINGLE_Release
DCH_32.SINGLE2_Release

DCH_32.SINGLE_SerialNr
DCH_32.SINGLE2_SerialNr
DCH_32.SINGLE_GetRS232
DCH_32.SINGLE_GetRS232Host
DCH_32.SINGLE2_GetRS232Host
DCH_32.SINGLE_GetParameter

DLL Version 4.37

5. Aufruf der DLL-Funktionen aus VisualBasic-Programmen

' Routinen zum Testen und zum Einsatz der Funktionen der DC-HI-NET Treiber-DLL

```

Declare Function HINET_Version Lib "DCH_32.DLL" (ByVal szVersion As String) As Long
Declare Function HINET_Init Lib "DCH_32.DLL" () As Long
Declare Function HINET_Init2 Lib "DCH_32.DLL" (ByVal szInit As String) As Long
Declare Function HINET2_Init2 Lib "DCH_32.DLL" (ByVal szInit As String) As Long
Declare Function HINET_IniFile Lib "DCH_32.DLL" (ByVal szIniFile As String) As Long
Declare Function HINET_Reset Lib "DCH_32.DLL" (ByVal Chan As Long) As Long
Declare Function HINET_Get Lib "DCH_32.DLL" (ByVal Typ As Long, ByVal Chan As Long,
    ByVal Messwert As Double, ByVal fWait As Long) As Long
Declare Function HINET_Get3 Lib "DCH_32.DLL" (ByVal Typ As Long, ByVal Chan As Long,
    ByVal Messwert As String, ByVal fWait As Long) As Long
Declare Function HINET_Get4 Lib "DCH_32.DLL" (ByVal szPara As String) As Double
Declare Function HINET_Akt Lib "DCH_32.DLL" (ByVal Chan As Long, ByRef Messw As Double,
    ByVal fWait As Long) As Long
Declare Function HINET_Min Lib "DCH_32.DLL" (ByVal Chan As Long, ByRef Messw As Double,
    ByVal fWait As Long) As Long
Declare Function HINET_Max Lib "DCH_32.DLL" (ByVal Chan As Long, ByRef Messw As Double,
    ByVal fWait As Long) As Long
Declare Function HINET_Dif Lib "DCH_32.DLL" (ByVal Chan As Long, ByRef Messw As Double,
    ByVal fWait As Long) As Long
Declare Function HINET_Set0 Lib "DCH_32.DLL" (ByVal Chan As Long, ByRef NullW As Double,
    ByVal fWait As Long) As Long

Declare Function HINET_Error Lib "DCH_32.DLL" (ByVal Chan As Long) As Long
Declare Function HINET_Error2 Lib "DCH_32.DLL" (ByRef ArChan As Long) As Long
Declare Function HINET2_Error2 Lib "DCH_32.DLL" (ByRef ArChan As Long) As Long
Declare Function HINET_Error2a Lib "DCH_32.DLL" (ByVal szChanError As String) As Long
Declare Function HINET2_Error2a Lib "DCH_32.DLL" (ByVal szChanError As String) As Long

Declare Function HINET_DeImin Lib "DCH_32.DLL" (ByVal Chan As Long) As Long
Declare Function HINET_DeImax Lib "DCH_32.DLL" (ByVal Chan As Long) As Long
Declare Function HINET_DeIminMax Lib "DCH_32.DLL" (ByVal Chan As Long) As Long

Declare Function HINET_Trigger Lib "DCH_32.DLL" (ByVal Chan As Long) As Long
Declare Function HINET_DigiOut Lib "DCH_32.DLL" (ByVal Chan As Long, ByVal Gruppe As Long,
    ByVal Bits As Long) As Long
Declare Function HINET_DigiIn Lib "DCH_32.DLL" (ByVal Chan As Long, ByRef Bitmaske As Long,
    ByVal fWait As Long) As Long

Declare Function HINET_Special Lib "DCH_32.DLL" (ByVal Func As Long, ByVal Chan As Long,
    ByVal Param As Long) As Long
Declare Function HINET_Special2 Lib "DCH_32.DLL" (ByVal Func As Long, ByVal Chan As Long,
    ByVal szBuffer as String, ByVal BytesInBuffer As Long) As Long

Declare Function SINGLE_Version Lib "DCH_32.DLL" (ByVal szVersion As String) As Long
Declare Function SINGLE2_Version Lib "DCH_32.DLL" (ByVal szVersion As String) As Long
Declare Function SINGLE_Revision Lib "DCH_32.DLL" (ByVal szRevision As String) As Long
Declare Function SINGLE2_Revision Lib "DCH_32.DLL" (ByVal szRevision As String) As Long
Declare Function SINGLE_Release Lib "DCH_32.DLL" (ByVal szRelease As String) As Long
Declare Function SINGLE2_Release Lib "DCH_32.DLL" (ByVal szRelease As String) As Long
Declare Function SINGLE_SerialNr Lib "DCH_32.DLL" (ByVal nSerialNr As String) As Long
Declare Function SINGLE2_SerialNr Lib "DCH_32.DLL" (ByVal nSerialNr As String) As Long

```

DLL Version 4.37

```
Declare Function SINGLE_GetRS232 Lib "DCH_32.DLL" (nChan As Long,  
    ByRef nBaudrate As Long, ByRef nDatabits As Long,  
    ByRef nStopbits As Long, ByRef nParity As Long) As Long  
Declare Function SINGLE_GetRS232Host Lib "DCH_32.DLL" (ByRef nBaudrate As Long)  
    As Long  
Declare Function SINGLE2_GetRS232Host Lib "DCH_32.DLL" (ByRef nBaudrate As Long)  
    As Long  
Declare Function SINGLE_GetParameter Lib "DCH_32.DLL" (nChan As Long, nParaNr As Long,  
    By Reference nParaVal As Long) As Long
```

STEINWALD datentechnik GmbH

Oskar-Loew-Str. 12 D-95615 Marktreidwitz
Tel.: +49 9231 9630-10 Fax: +49 9231 9630-11
eMail: info@steinwald.com
internet: <http://www.steinwald.com>

Rechtlicher Hinweis:

Microsoft; Windows; Windows NT, Excel und Access sind eingetragene Warenzeichen der Microsoft Corporation.