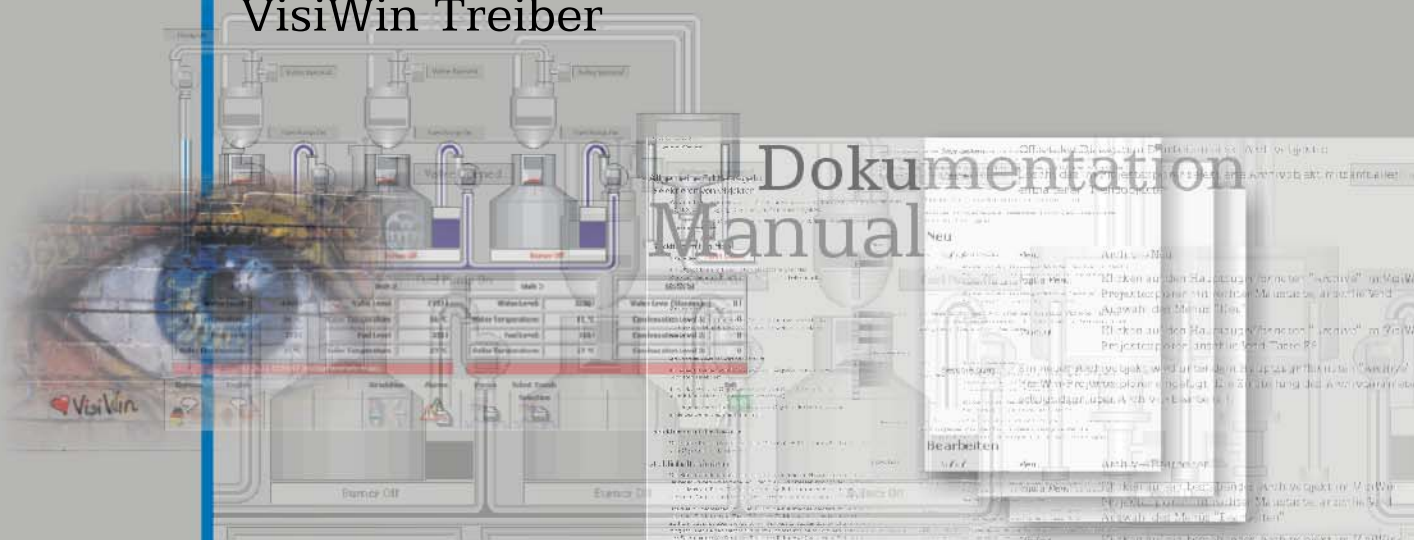


VisiWinNET 2005

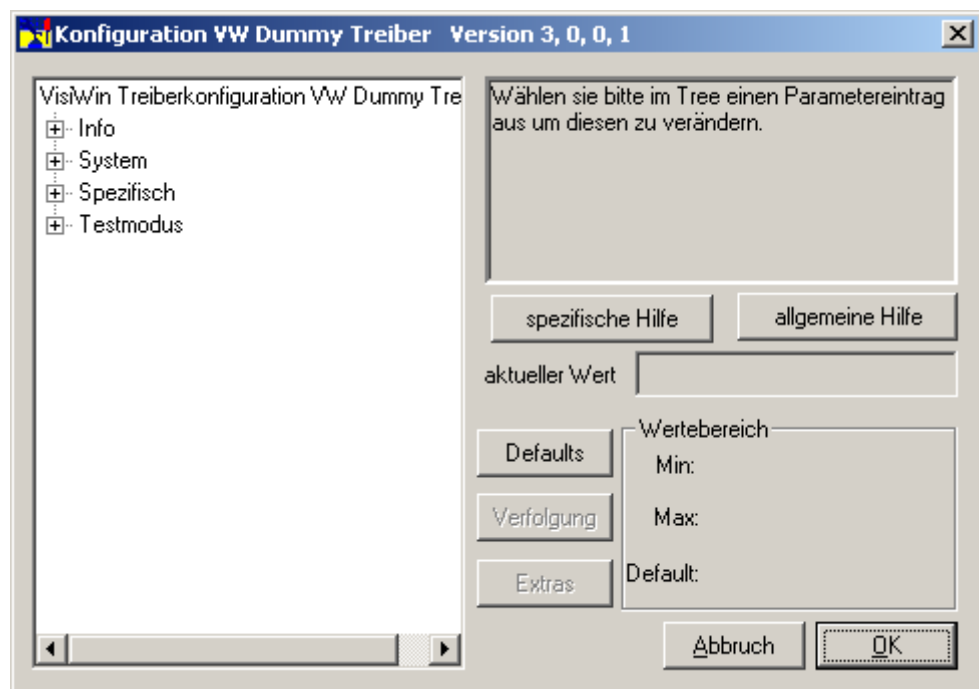
VisiWin Treiber



- VisiWin
- VisiWinNET 2005
 - Allgemeines
 - Klassenbibliothek
 - Systeme
 - Werkzeuge
 - Technische Infos
- Inosoft OPCServer
 - Grundlagen und Hilfsmittel
 - Protokolle

VisiWinNET 2005

VisiWin Treiber









Typischer Konfigurationsdialog eines VisiWin Treibers

Ohne ausdrückliche schriftliche Erlaubnis der INOSOFT GmbH darf der Inhalt des Handbuches nicht anderweitig verwendet werden.
Den Inhalt des Handbuches haben wir auf Übereinstimmung mit der beschriebenen Software geprüft. Trotzdem können Abweichungen nicht ausgeschlossen werden. Aus diesem Grund können wir für eine völlige Übereinstimmung keine Gewähr geben. Der Inhalt des Handbuches wird regelmäßig auf Änderungen geprüft. Korrekturen sind in der nachfolgenden Ausgabe vorhanden. Verbesserungsvorschläge werden dankbar angenommen.

Legende

Um auf besondere Einstellungen aufmerksam zu machen, werden in den INOSOFT-Dokumentationen folgende Symbole verwendet:

	Achtung	Absätze mit diesem Zeichen sollten Sie besonders aufmerksam lesen - und beachten.
	Hinweis	Wichtige Einstellung "additional informations".
	Tipp	Viele Wege führen nach Rom, hier finden Sie eine Abkürzung.
	In Arbeit	Funktionen in Vorbereitung, bzw. bereits implementiert, jedoch noch nicht für die Dokumentation aufbereitet.
	Beispiel ausführen	Anweisungen, die in einem Beispiel auszuführen sind.
	Beispiel beobachten	Ergebnisse, die durch die Ausführung der Beispielanweisungen zu beobachten sind.

© / ™ / ®

Windows®, Windows NT®, Windows 2000®, Windows XP®
sind eingetragene Warenzeichen der Firma Microsoft.

Weitere mit ® erwähnte Produktnamen sind Warenzeichen der betreffenden
Herstellerfirmen.

INOSOFT GmbH

VisiWinNET Version: ab 6.04.000

erstellt am 07.06.2010

Inhalt

1 Vorwort	1
2 VisiWin-Treiber	2
2.1 VisiWin-Treiber mit dem VisiWinNET Variablenkern	2
2.1.1 Prozessvariable	3
2.1.2 Strukturen	3
2.1.3 Abtaststruktur	4
2.1.4 Bildstruktur	4
3 Die Installation	5
3.1 Installation	5
4 Projekteinbindung	7
4.1 Projekteinbindung in VisiWinNET	7
5 Konfiguration	9
5.1 Starten der Konfiguration	9
5.1.1 In der Entwicklungsumgebung	9
5.1.2 Direktes Starten des Konfigurators	10
5.1.3 Start über den VisiWin-Treiber	11
5.2 Bedienung des Konfigurators	11
5.2.1 Karteikarte Standard	12
5.2.2 Parameterbereich Erweitert	13
5.2.3 Parameterbereich Datenformat	15
5.2.4 Parameterbereich Spezifisch	16
5.2.5 Parameterbereich Traces	16
5.2.6 Parameterbereich Datenüberwachung	17
5.2.7 Frei editierbare Einträge	18
6 Der VisiWin-Treiber	19
6.1 Das VisiWin-Treiber-Programm	19
6.2 Start des VisiWin-Treibers	19
6.2.1 Start über die Visualisierung	19
6.2.2 Direkter Start des VisiWin-Treibers [Testmodus]	19
6.3 Bedienung des VisiWin-Treibers	21
6.3.1 Die Fensteroberfläche	21
6.4 Anzeigen und Senden von Daten	24
6.4.1 Das Menü [Optionen]	24
6.5 Datenaustausch zur Prozessdatenbank der Applikation	26
6.6 Der Diagnose Dialog	28
6.7 Optimierung der Kommunikation	29

1 Vorwort

Zu diesem Handbuch

Dieses Handbuch enthält spezifische Informationen zu VisiWin Treibern. Es beschreibt die Eigenschaften, die alle VisiWin Treibern haben. Neben diesem Dokument existiert für jeden Treiber ein eigenes Dokument, das auf die protokollspezifischen Eigenheiten eingeht.

Fragen und Probleme

Bei technischen Fragen und Problemen wenden Sie sich bitte an Ihre zuständige INOSOFT Vertretung oder an den Support der INOSOFT GmbH unter Tel.: +49 (5221) 16 66 02 bzw. E-Mail: Support@INOSOFT.com

Häufige Fragen und Probleme werden auch auf unserer Homepage unter www.inosoft.com behandelt.

Dort finden Sie auch einen Supportbereich, in dem Sie sich mit Fragen direkt an unsere Zentrale wenden können.

2 VisiWin-Treiber

VisiWin-Treiber sind Kommunikationskomponenten, die den Datenaustausch zwischen Variablenkern und einer SPS regeln. Diese Dokumentation gibt einen Überblick über die Handhabung der VisiWin-Treiber. Bei allen Informationen wird dabei vorrangig die Verbindung der VisiWin-Treiber mit VisiWinNET beschrieben.

Das Dokument beschreibt die Merkmale, die für alle VisiWin-Treiber identisch sind. Parameter oder Besonderheiten, die abhängig vom jeweiligen Protokoll sind, werden in der spezifischen Dokumentation, die zum jedem VisiWin-Treiber gehört, beschrieben.

Mit dem Inhalt dieses Dokumentes soll zu der Installation, der Einbindung in ein VisiWinNET-Projekt, der Konfiguration des VisiWin-Treibers bis hin zur Optimierung der Kommunikation eine wesentliche Hilfestellung gegeben werden. Zudem wird Hintergrundwissen zum besseren Verständnis des Zusammenspiels zwischen VisiWin-Treiber und der Applikation zur Laufzeit vermittelt.



Die Vergangenheit hat gezeigt, dass es gerade bei der Inbetriebnahme immer wieder zu Problemen verschiedenster Art gekommen ist. Besonders die Schnittstelle zwischen Soft- und Hardware verhält sich unter realen Bedingungen oft für den Anwender unerwartet. Daher möchten wir an dieser Stelle noch einmal darauf verweisen, dass es auf alle Fälle sinnvoll ist, die Kommunikation schon zu Beginn eines Projektes realitätsnah zu testen, um nicht während der Inbetriebnahme in unnötige Schwierigkeiten zu kommen. Außerdem ist zu beachten, dass die Kommunikation sich mit einer kompletten (großen) Datenbank anders verhält als mit einer Testdatenbank, die nur einige Objekte beinhaltet (siehe Optimierung).

Ein Schaubild, das die Einbindung der VisiWin-Treiber darstellt befindet sich im Kapitel Prozessanbindung/Einführung in die Prozessanbindung.

2.1 VisiWin-Treiber mit dem VisiWinNET Variablenkern

In der historischen Entwicklungsgeschichte von VisiWin stellten die VisiWin-Treiber (vor der Entwicklung des OPC-Standards) die optimale Lösung einer flexiblen und schnellen Schnittstelle zur Anbindung an verschiedene Kommunikationspartner (SPS'en) dar. Ähnliche Konzepte verfolgten auch andere Visualisierungssysteme. Der große Nachteil dabei ist, dass das Rad immer wieder neu erfunden werden musste. Bedeutet, dass jedes Visualisierungssystem, natürlich auch VisiWin, nicht auf fertige Kommunikationskomponenten anderer Hersteller zugreifen konnte, sondern diese mit großen Aufwand selber entwickeln musste. Und auch in Zukunft wird es verschiedene Sonderfälle von Kommunikationswegen geben, wo der VisiWin-Treiber die optimale Lösung einer Kommunikation darstellt.

Durch die Entwicklung des OPC-Standards ist es möglich geworden, Kommunikationskomponenten anderer Hersteller mit seinem eigenen Produkt zu verbinden. Um die Vorteile, die dieses System bietet, zu nutzen, war es notwendig, OPC als zentrale Verbindungsschnittstelle auch in VisiWinNET zu integrieren.

Um aber auch die bisher ca. 100 entwickelten VisiWin-Treiber zu nutzen und sich nicht die Möglichkeiten zu versperren, auch zukünftig diese optimierte Softwareschnittstelle zu verwenden, wurde gefordert, diese Kommunikationskomponenten mit der OPC Spezifikation in VisiWin zu verbinden. Dies macht an vielen Stellen Schwierigkeiten, da

einige Funktionalitäten, die es im Zusammenspiel mit den VisiWin-Treibern gibt, unter OPC nicht spezifiziert sind.

Im Folgenden werden diese Punkte verdeutlicht und Lösungen vorgestellt, um den VisiWinNET Variablenkern optimal mit den VisiWin-Treibern zu nutzen.

2.1.1 Prozessvariable

Das als Prozessvariable benannte Element wird Item genannt. Ganz korrekt ist diese Zuordnung aber nicht, da die Prozessvariable nach OPC Spezifikation noch weitere Bereiche beinhaltet:

- Wert (Value). Hier befindet sich die eigentliche Abbildung des Speichers aus der SPS. Dies ist also vergleichbar mit der bisher bekannten Prozessvariablen von VisiWin32.
- Zeitstempel (Timestamp). Dieser Teil zeigt den Zeitpunkt der letzten Aktualisierung des Variablenwertes an. Der Variablenkern setzt diesen Wert auf den Zeitpunkt, zu dem er die Prozessvariable aus dem Speicherbereich des VisiWin-Treibers ausliest. Dies ist anders als in einem OPC-Server, der den Zeitstempel auf den Zeitpunkt setzt, zu dem der Variablenwert aus der SPS gelesen wurde.
- Qualität (Quality). Über diese Eigenschaft erhält der Benutzer die Information, ob der Variablenwert, der aus der SPS gelesen wurde, gültig ist oder nicht. Hat der VisiWin-Treiber zum Beispiel einen Lesefehler in dem Datenbereich, in dem sich die Prozessvariable befindet, so wird die Qualität auf "BAD" gesetzt.
- Datentyp (Datatype). Definiert den Datentyp, mit dem der Variablenwert interpretiert wird. Z.B.:VT_UI2 ist ein vorzeichenloser 16 Bit Wert (Bereich von 0 bis 65535).

2.1.2 Strukturen

Strukturen reduzieren erheblich den Protokollaufwand zwischen Visualisierung und Steuerung. In der Visualisierung werden mehrere (oder viele) Prozesswerte durch Strukturdefinitionen zu einem Datenblock zusammengefasst, der durch einen einzigen Lese-/Schreibzugriff mit der Steuerung ausgetauscht wird. Die Auflösung zu den einzelnen in der Visualisierung benötigten Elementen (Prozesswerten) erfolgt im Variablenkern.

VisiWinNET bietet folgende Möglichkeit: Zuerst ist eine Struktur zu definieren. In dieser Struktur werden nun Elemente definiert. Dabei können Elemente auch wieder Strukturen sein. Nun ist eine Prozessvariable anzulegen, das als Datentyp die Struktur verwendet. In den Steuerelementen von VisiWinNET besteht jetzt die Möglichkeit, auf die Elemente der Struktur zuzugreifen.

2.1.3 Abtaststruktur

Abtaststrukturen bieten die Möglichkeit, bei Bedarf programmatisch oder über die Visualisierungsoberfläche den Austausch von einzelnen Daten oder Datenblöcken zu steuern.

Um den Datenstrom der Kommunikation zur SPS zu optimieren, können in VisiWinNET "Gruppen" definiert werden. In diesen Gruppen befinden sich beliebige Prozessvariablen. Zur Laufzeit können die Gruppen aktiviert oder deaktiviert werden, sodass die Prozessvariablen nur dann aktualisiert werden, wenn der Benutzer es möchte. Werden keine Gruppen angelegt, so befinden sich alle definierten Variablen in einer Gruppe "Standard". Prozessvariablen werden nur dann gelesen, wenn diese aktuell in der laufenden Applikation benötigt werden. Der Variablenkern hat einen Automatismus, dies zu erkennen. So sind Gruppen nur dann aktiv, wenn ein Element daraus in der Applikation benötigt wird.

2.1.4 Bildstruktur

Bildstrukturen bilden eine Spezialfall der Abtaststruktur. Beim Laden einer Bildschirmmaske soll die Kommunikation einzelner auf dem Bild dargestellter Prozesswerte gestartet werden.

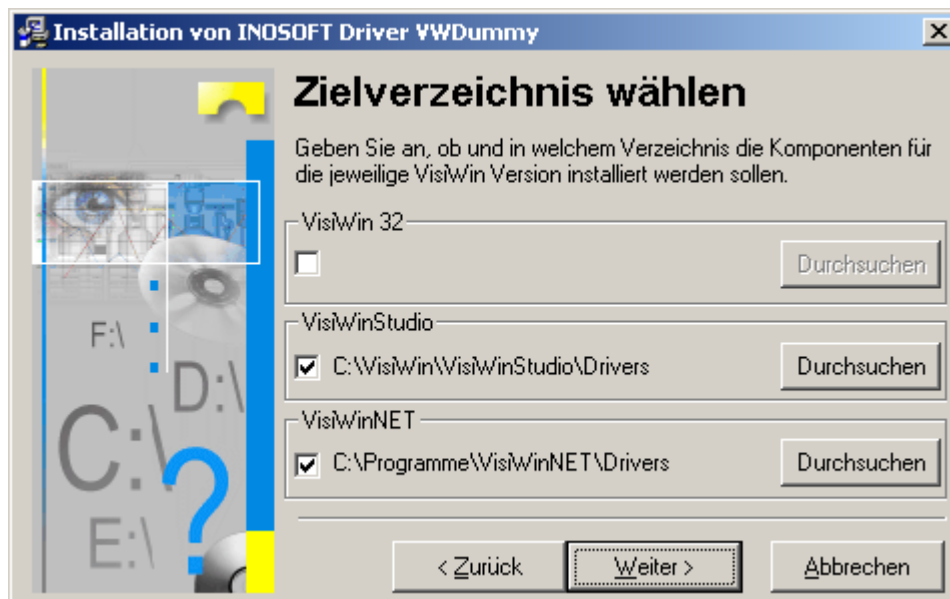
Auch hierzu lassen sich OPC-Gruppen verwenden. Hierzu sind Gruppen zu definieren, in denen sich die Prozessvariablen befinden, die beim aktivieren eines Formulars aktualisiert werden sollen. In dem "Load"-Ereignis des Formulars ist dann die OPC Gruppe zu aktivieren. In dem Ereignis "Close" ist diese Gruppe dann wieder zu deaktivieren.

3 Die Installation

3.1 Installation

Geliefert wird die Installation als SETUP.EXE. Dieses Programm kann einfach über den Explorer oder über das "Windows-Startmenü" gestartet werden.

Das Installationstool sucht nach installierten VisiWin-Versionen. Für jede installierte Version kann, wenn nötig, der Pfad für den VisiWin-Treiber ausgewählt werden.



Bei einer Installation werden folgende Dateien aus dem SETUP.EXE in das Zielverzeichnis kopiert (hier Beispiel für den VWDummy VisiWin-Treiber):

VWDummy.exe	VisiWin-Treiberprogramm
VWDummyk.txt	Textdatei, in welcher die zum VisiWin-Treiber gehörigen Parameter beschrieben sind. Diese Datei wird vom Konfigurator benötigt.
VWDummy.dep	Datei, die die Abhängigkeiten des VisiWin-Treibers definiert. Diese wird für die VisiWinNET Laufzeitgenerierung benötigt.
VWDummy.chm	Hilfdatei in der die protokollspezifischen Parameter des VisiWin-Treibers beschrieben sind. Einige VisiWin-Treiber verwenden an dieser Stelle noch ein WORD Dokument.
eventuell weitere Dateien	z.B. DLL's... Diese Dateien können auch andere Zielverzeichnisse haben.

Alle VisiWin-Treiber tragen sich in die Registry des Systems ein. Die Einträge sind bei VisiWinNET unter folgendem Pfad zu finden:

[HKEY_LOCAL_MACHINE\SOFTWARE\INOSOFT GmbH\VisiWin\5.0\Drivers\VWDummy]



Ein VisiWin-Treiber lässt sich nur installieren, wenn auf dem System schon eine VisiWinNET-Version installiert wurde.



Sollte es vorkommen, dass der VisiWin-Treiber aus der VisiWinNET Entwicklungsumgebung nicht anwählbar ist, also nicht in der Auswahlliste erscheint, so ist als erstes zu kontrollieren, ob der VisiWin-Treiber sich in die Registry ihres Systems eingetragen hat. Wenn nicht, so ist die Installation ein weiteres Mal durchzuführen.

4 Projekteinbindung

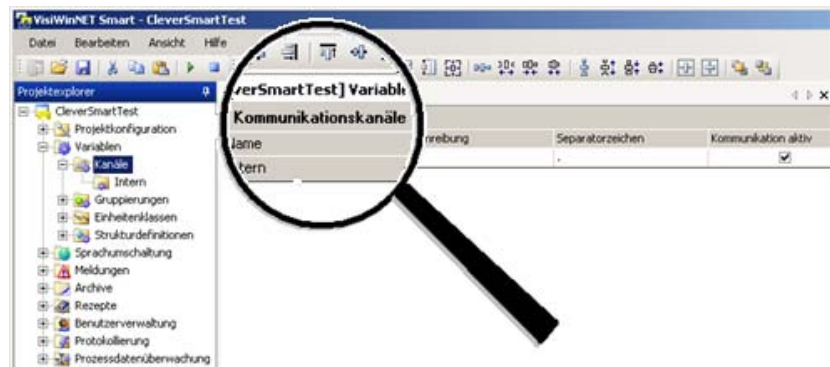
4.1 Projekteinbindung in VisiWinNET



Der Knoten "Variablen" im VisiWinNET Projektextplorer ist zu expandieren. Über einen Klick auf den darunter liegenden Knoten "Kanäle" ist der Tabelleneditor des Variableneditors zu öffnen.




Ist im Projektextplorer der Knoten "Kanäle" markiert, werden im Tabelleneditor alle Kommunikationskanäle des Projektes aufgelistet.



Auf dem Tabelleneditor ist das Kontextmenü zu öffnen (Klick mit rechter Maustaste) und hier der Eintrag "Neu" zu wählen.

Diese Aktion öffnet den Dialog "Kanal hinzufügen". Hier werden grundlegende Eigenschaften des neuen Kanals festgelegt:

- Name: Ein wahlfreier Name für den Kanal. Der Name muss innerhalb der Kommunikationskanäle des Projektes eindeutig sein. Verweise auf Prozessvariablen in der Applikation beziehen diesen Namen ein (z.B. "Ch1.w0" bezeichnet die Variable "wo" im Kommunikationskanal mit dem Namen "Ch1").
- Kanaltyp: Entweder OPC oder Treiber. Diese Einstellung ist auf "Treiber" zu stellen.
- OPC-Server/ Treiber: Über die -Schaltfläche ist der Dialog zur Auswahl einer Kommunikationskomponente zu öffnen. Hier ist der entsprechende Treiber auszuwählen.

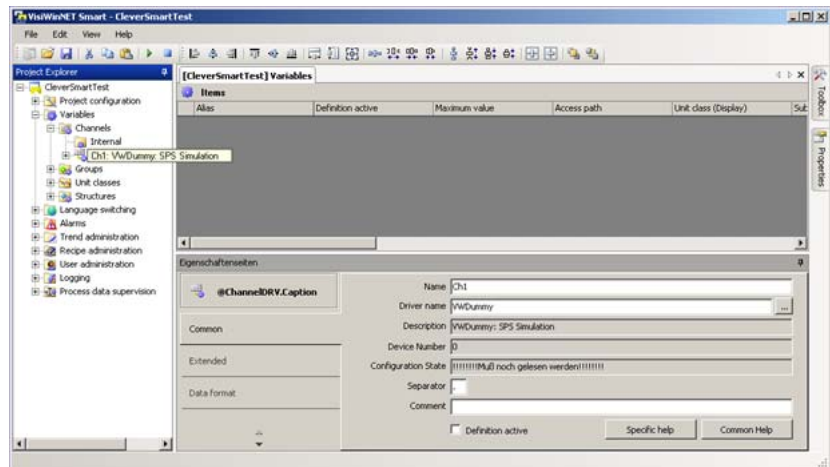
Nach dem Schließen des Dialoges ist auf den Knoten "Kanäle" im Projektextplorer zu klicken.



Die Markierung des "Kanäle"-Knotens bewirkt das Speichern der eingegebenen Daten in die Projektdatenbank.

Der neue Kommunikationskanal wird nun durch einen Knoten im Projektextplorer repräsentiert.

Wird der Knoten markiert, blendet sich automatisch die VisiWinNET-Eigenschaftenseite ein. Hier wird der vollständige Parametersatz des Kommunikationskanals zum Bearbeiten angeboten.



5 Konfiguration

5.1 Starten der Konfiguration

Es gibt mehrere Wege, die Konfiguration des VisiWin-Treibers zu starten. Die drei Möglichkeiten werden unten beschrieben.

5.1.1 In der Entwicklungsumgebung

Dies ist die "normale" Vorgehensweise, einen VisiWin-Treiber zu konfigurieren:

Nachdem in VisiWinNET ein neuer Kommunikationskanal eingefügt und die Kommunikationskomponente festgelegt wurde, werden die Konfigurationsdaten über die VisiWinNET Eigenschaftenseite eingestellt. Dazu ist zuerst der Kommunikationskanal im Projektextplorer zu markieren.



Die folgenden Karteikarten enthalten Einstellungen, die zum grundlegenden Funktionsumfang von VisiWin-Treibern gehören:

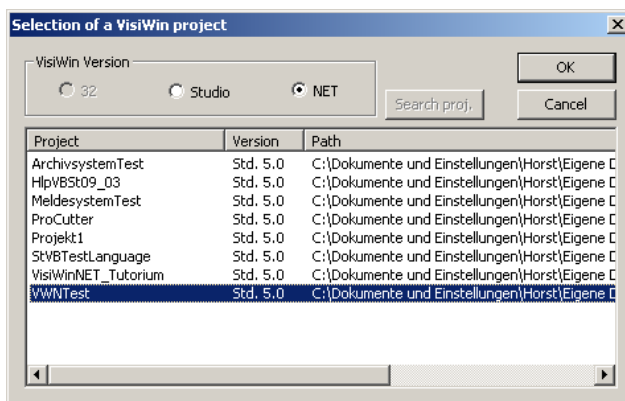
Karteikarte	Einstellungen
Erweitert	<ul style="list-style-type: none"> • Kommunikationsparametrierung (siehe Handbuch Prozessanbindung) • Informationen zum Treiber • Hauptzyklus und Zykluszwischenzeit • Testmodus
Datenformat	Einstellungen zu speziellen Datentypkonvertierungen
Traces	Überwachung von Ereignissen im Treiber und Speichern als Log-Datei.
Datenüberwachung	Legt Variablen fest, deren Werte im Treiber überwacht und in die Log-Datei geschrieben werden sollen.

Die Karteikarte "Spezifisch" enthält die individuellen Einstellungen eines VisiWin-Treibers. Weitere Informationen zu diesen Einstellungen befinden sich in der spezifischen Treiberhilfe. Diese Hilfe wird über die Schaltfläche "spezifische Hilfe" auf der Karteikarte "Standard" aufgerufen.

5.1.2 Direktes Starten des Konfigurators

Als globales Konfigurationswerkzeug liegt im Unterverzeichnis "Common" des VisiWinNET-Installationspfades die Datei "VisiWinNET.DrvConfig.exe".

Nach dem Start über den Windows-Explorer oder über das "Windows Startmenü" erscheint folgender Dialog.



Dieser Dialog umfasst die Auswahl der VisiWin-Version und des VisiWinNET-Projektes.



Dieser Dialog erscheint nur, wenn mehr als ein Treiber als Kommunikationskomponente im Projekt verwendet wird. Hier ist also der zu konfigurierende Kanal auszuwählen

5.1.3 Start über den VisiWin-Treiber

Über die Auswahl des Menüs [**Optionen\Konfiguration**] des laufenden VisiWin-Treibers wird der Konfigurator gestartet.



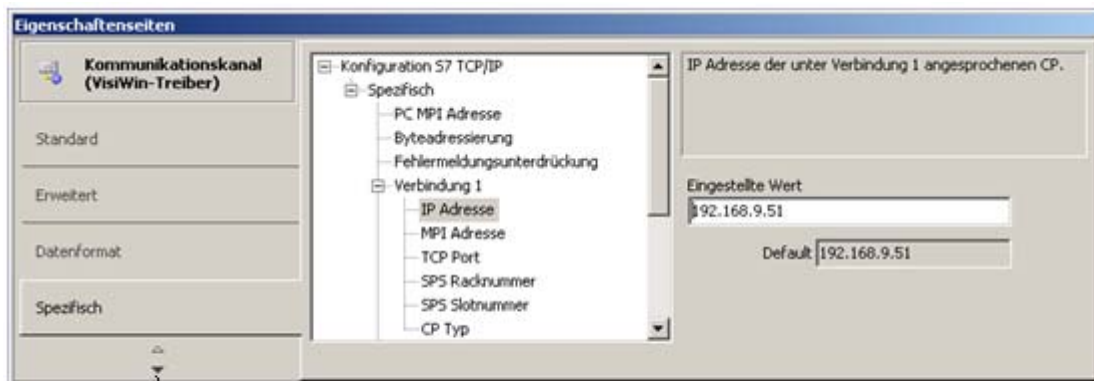
Diese Methode kann nur dann angewendet werden, wenn der VisiWin-Treiber schon einmal aus der Entwicklungsoberfläche konfiguriert wurde und der Fensterstil mindestens auf sichtbar (Parameterwert 2) steht. Zudem muss der Konfigurator und die zugehörige TXT Datei mit auf dem System installiert worden sein (dies betrifft die Laufzeitkonfiguration).



Nach Änderung der Konfiguration muss der Treiber neu gestartet werden, damit die neuen Parameter wirksam werden.

5.2 Bedienung des Konfigurators

Die Oberfläche der Konfiguration beinhaltet mehrere Karteikarten, die sich durch die Karteireiter am rechten Rand aktivieren lassen:



Bildlaufleiste für Karteireiter

Sind nicht alle Karteireiter sichtbar, weil der Dialog zu klein ist, kann über die Bildlaufleiste der Karteireiter zu den unsichtbaren Karteikarten navigiert werden.

5.2.1 Karteikarte Standard



Einige der hier aufgelisteten Einstellungen lassen sich beim Aufruf des Konfigurators außerhalb der Entwicklungsumgebung nicht ändern.

Die Karteikarte "Standard" beinhaltet Konfigurationseinstellung für die Einbindung in VisiWinNET:

Einstellung	Beschreibung
Name	Gibt den Namen des Kommunikationskanals an. Der Name des Kommunikationskanals wird beispielsweise bei der Anbindung an Steuerelemente vor die Variablennamen gehängt. Für die Treiberfunktion hat er keine Bedeutung.
Treibername	Gibt den Namen des verwendeten Treibers an.
Beschreibung	Gibt einen Kurztext für den Treiber an.
Schnittstelle	Gibt die Universalschnittstelle des Treibers im Projekt an. Die Universalschnittstelle wird beim Einfügen des Treiber in ein Projekt automatisch vergeben. Sie identifiziert die Treiberinstanz eindeutig, wenn mehrere Kommunikationskanäle im Projekt angelegt werden, die den gleichen Treiber verwenden.
Separator	Gibt das Separatorzeichen an, das die Ansicht der Variablen im Editor in Namensräume unterteilt.
Kommentar	Gibt einen wahlfreien Kommentar an
Definition aktiv	Gibt an, ob zur Laufzeit die Kommunikationskomponente (das Treiberprogramm) gestartet wird.

Die Karteikarte enthält außerdem zwei Schaltflächen:

Allgemeine Hilfe	Öffnet dieses Dokument. Hier werden die Eigenschaften beschrieben, die jeder VisiWin-Treiber beinhaltet
Spezifische Hilfe	Öffnet das treiberspezifische Dokument. Die Installation eines VisiWin-Treibers bringt immer auch eine Online-Hilfe (<Treibername>.CHM) mit, in der die spezifischen Konfigurationseinstellungen, der Adressaufbau und spezielle Fehlermeldungen beschrieben werden.

5.2.2 Parameterbereich Erweitert

Einstellung	Beschreibung
Synchrone Kommunikation ermöglichen	<p>Der Parameter "Synchrone Kommunikation ermöglichen" stellt ein, ob der Variablenkern synchrone Aufrufe bei einem VisiWin-Treiber verwendet. Dies ist insbesondere dann von Bedeutung, wenn:</p> <ul style="list-style-type: none"> • Der lesende oder schreibende Zugriff in einer Gruppe auf "synchron" eingestellt ist (→Parameter "Lesezugriffe"/"Schreibzugriffe") • Der Parameter "Startwerte nicht synchron einlesen" der Kommunikationskomponente auf "False" eingestellt ist.
	Einstellung Beschreibung
True	Synchrone Aufrufe werden verwendet.
False	Synchrone Aufrufe werden als asynchrone Aufrufe ausgeführt. Dies ist das Verhalten, das VisiWinNET standardmäßig vor der Version 4.07 zeigte.
Startwerte nicht synchron einlesen	<p>Der Parameter "Startwerte nicht synchron einlesen" bestimmt das Initialisierungsverhalten bei der ersten Datenanforderung. Der Variablenkern liest im Normalfall bei der ersten Anforderung eines Variablenwertes den Wert synchron aus der Kommunikationsschnittstelle. Damit werden Fehlzugriffe und fehlerhafte Datendarstellungen in der Applikation vermieden. In bestimmten Fällen kann dies jedoch zum Fehlerfall, bzw. zu unerwünschten Verzögerungen führen.</p>
Aktuelle Treiberversion	<p>Diese Information erhält man auch, wenn man mittels des Explorers über die rechte Maustaste die Eigenschaften des VisiWin-Treibers angezeigt werden. Das Format über den Explorer ist im Gegensatz zum hier angezeigten dreistelligen Format (x.yy.zzz) vierstellig (x.y.a.z). Der aktuelle Wert kann nur angezeigt und nicht verändert werden.</p>

Fensterstil

Der Fensterstil des Kommunikationstreibers. Es gibt dabei fünf Möglichkeiten:

0-Fenster nicht sichtbar

Der VisiWin-Treiber hat kein sichtbares Fenster. Er erscheint weder als Icon in der Taskleiste noch ist er über den Taskmanager zu erreichen.

Diese Einstellung ist für die endgültige Laufzeit auf dem Zielrechner sinnvoll, da der Bediener nicht über Windows an den VisiWin-Treiber herankommen sollte, um diesen z.B. zu beenden. Zudem wird bei dieser Einstellung am wenigsten Rechnerperformance benötigt.

1-Fenster nur als Icon

Das Fenster ist nur als Icon in der Taskleiste sichtbar.

2-Fenster sichtbar

Das Fenster des VisiWin-Treibers ist sichtbar. Hierbei besteht nicht die Möglichkeit, die Datenbereiche, die mit der SPS ausgetauscht werden, anzuzeigen oder diese vom VisiWin-Treiber aus zu verändern!

3-Fenster mit Debuginfo

Das Fenster ist sichtbar und Datenbereiche vom VisiWin-Treiber können angezeigt und verändert werden.

4-Manueller Start des VisiWin-Treibers

Der **VisiWin**-Treiber wird gestartet. Die Kommunikation muss aber extra über das Menü [**Datei**→**Start**] gestartet werden.



Wird der VisiWin-Treiber manuell und nicht über VisiWin gestartet und ist ein Fensterstil kleiner 3 eingestellt worden, so wird dieser Wert nicht übernommen. Stattdessen wird er Stil auf 3 gesetzt, damit der Treiber bedienbar ist.



Diese Einstellung sollte nicht verwendet werden, wenn der VisiWin-Treiber über VisiWin gestartet wird, da die Datenkommunikation nicht automatisch gestartet wird.

Hauptzyklus

Dieser Wert wird entweder in Milli- oder in 10 Millisekunden angegeben (siehe Infocfeld des Konfigurators).



Für den ersten Kommunikationstest sollte der Default-Wert beibehalten werden. Das Kapitel "Optimierung der Kommunikation" enthält weiterführende Information über diesen Parameter.

Zykluszwischenzeit

Dieser Wert sollte im Normalfall immer auf dem Default-Wert 1 bleiben!



Nur in Ausnahmefällen ist dieser Parameter zu verändern. Der Parameter ist **nicht auf 0 zu setzen!** Das Kapitel "Optimierung der Kommunikation" enthält auch für diesen Parameter weiterführende Information bereit.

**Testadresse/
Testanzahl**

VisiWin-Treiber beinhalten einen Testmodus. Dabei wird der Treiber direkt ohne eine Applikation gestartet. Über die Testadresse können Daten mit der Steuerung ausgetauscht werden.

In die Testadresse ist der Adress-String einzutragen. Dieser String muss den Konventionen genügen, die in der spezifischen Treiberdokumentation für das verwendete Protokoll beschrieben ist. Als Beispiel für das Siemens Protokoll AS511 (S5-PG) könnte eine mögliche Adresse

01:20:10

sein. Dies steht für: COM1:Datenbaustein 20:ab Wort 10.

Die Testlänge beschreibt die Anzahl der Worte, die der VisiWin-Treiber für den unter **Testadresse** gewählten Bereich aus der SPS auslesen soll.



Auch ohne Applikation? Trotzdem: Der Testmodus funktioniert nur, wenn schon ein VisiWin Projekt auf dem PC eingerichtet ist, in dem der VisiWin-Treiber eingebunden und parametriert wurde.

5.2.3 Parameterbereich Datenformat

Hier wird definiert, ob zwischen VisiWin-Treiber und Variablenkern Datentypen gewandelt werden sollen. Ob gewandelt werden soll, ist abhängig vom SPS Typ. Wird hier eine Datenformatänderung eingestellt, so wird diese nicht vom VisiWin-Treiber, sondern vom Variablenkern durchgeführt, da diese Änderung vom Datentyp der Prozessvariablen abhängig ist. Der VisiWin-Treiber kennt nur den Datenblock, aber nicht dessen Unterteilung in die einzelnen Prozessvariablen.

**Gleitkomma-
konvertierung
(IEEE)**

Dieser Parameter gilt speziell für Siemens S5 SPSen. Diese verwenden ein eigenes Fließkommaformat. In dem Fall, dass die Fließkommawerte nicht richtig angezeigt werden, ist dieser Parameter anzuwählen (Wert=1). Dieser Parameter muss vielfach zusammen mit dem Worttausch bei Gleitkommawerten eingeschaltet werden.

**Worttausch
Gleitkommawerte**

Das High- und Low-Word von Gleitkommawerten soll getauscht werden.

**Worttausch
Langworte**

Das High- und Low-Word von Langwortwerten soll getauscht werden.

Texte

Das High- und Low-Byte von Prozessvariablen des Datentyps Text soll getauscht werden.

Motorola Formatanpassung

Viele SPSen verwenden nicht das Intel, sondern das Motorola Datenformat. Das bedeutet, dass die Daten byteweise sowie auch wortweise vertauscht vorliegen.

Der Bytetausch wird vom VisiWin-Treiber selber vorgenommen. Eventuell ist dies, wenn nötig, bei einigen Protokollen unter Kapitel "Spezifisch" einzustellen.

5.2.4 Parameterbereich Spezifisch

Die hier dargestellten Parameter sind VisiWin-Treiber spezifisch und in dieser Dokumentation nicht aufgeführt. Stattdessen werden sie in der treiberspezifischen Dokumentation beschrieben.

5.2.5 Parameterbereich Traces

Diese Parameter dienen dazu, die Tracefunktion zu konfigurieren. Dabei gibt es verschiedene, unabhängig von einander aktivierbare Traceebenen. Zusätzlich ist die Art der Handhabung der Datei, in denen die Traces aufgezeichnet werden, konfigurierbar.

Der Name der Datei ist meist identisch oder ähnlich dem EXE Namen des VisiWin-Treibers. Die Endung lautet immer "LOG". Normalerweise befindet sich die Tracedatei im Applikationsverzeichnis. Kann dieser Pfad vom Treiber nicht festgestellt werden, z.B. wenn der VisiWin-Treiber im Testmodus gestartet wird, so wird die Datei im Root-Verzeichnis auf dem Laufwerk C:\ abgelegt.

In der Defaulteinstellung sind alle Traces deaktiviert.

Sicheres Loggen	Das sichere Loggen öffnet und schließt die Datei für jeden Zugriff. Dies bedeutet eine deutlich höhere Systembelastung, hat aber den Vorteil, dass die Informationen in der Datei bei einem Absturz nicht verloren gehen. Im Normalfall kann die Datei bis zum Programmende immer geöffnet bleiben.
Trace Texte an Spy senden	Sollen die Texte nicht in einer Datei abgespeichert werden, können sie stattdessen an den "VisiSpy" geschickt werden. "VisiSpy" ist nicht in der Standard-Installation eines Treibers enthalten. Nehmen Sie Rücksprache mit der INOSOFT GmbH.
Interne Funktionen	Interne Funktionsabläufe werden protokolliert. Achtung: Dies bedeutet ein sehr hohes Datenaufkommen.
Protokoll Funktionen	Funktionsabläufe aus der Protokollebene des VisiWin-Treibers werden protokolliert. Achtung: Dies bedeutet ein sehr hohes Datenaufkommen.
Interne Fehler	Interne Funktionsfehler des VisiWin-Treibers werden protokolliert.
Protokoll Fehler	Protokollfehler aus der Kommunikationsebene des VisiWin-Treibers werden protokolliert.
Lesebefehle	Protokolliert alle Aufträge, die Daten aus der SPS zur Applikation kopieren. Achtung: Dies kann ein sehr hohes Datenaufkommen bedeuten.
Schreibbefehle	Protokolliert alle Aufträge, die Daten von der Applikation in die SPS kopieren

Windows Messages	Protokolliert alle Windows Messages, die zur Kommunikation zwischen VisiWin-Treiber und Variablenkern dienen.
Info	Protokolliert Status- und Diagnoseinformation des VisiWin-Treibers. Zur Zeit noch nicht implementiert.
Spezifische Meldungen	Speziell für Kunden einprogrammierte Traceausgaben können hier aktiviert werden.
Schreibdaten	Die ersten 32 Datenbits bei Schreibaufträgen werden protokolliert. Der Wert wird 4-stellig hexadezimal ohne Berücksichtigung von einem Datenformat ausgegeben.
Listen Infos	Infos über die Schreib- und Leseliste, die zur Kommunikation zwischen VisiWin-Treiber und Variablenkern dient. Achtung: Dies kann ein sehr hohes Datenaufkommen bedeuten.
Dateigröße	Gibt die Größe der Dateien in kBytes an. Die Angabe von 0 besagt, dass die Datei unendlich groß werden kann. Ist die vorgegebene Dateigröße erreicht, so wird die Datei komprimiert. Die letzten 25% des Inhalts der alten Datei werden dabei aber immer beibehalten. (Achtung: Das System wird instabil, wenn es nicht mehr genug Platz auf einem Laufwerk hat)
Aufzeichnungsmodus bei unendlich	Wird die Dateigröße über den vorangegangenen Parameter nicht eingeschränkt, kann die Datei Stunden- oder tageweise gesplittet werden. Dabei wird die alte Datei unter einem Dateinamen, in dem sich das aktuelle Datum bzw. die Zeit wiederfindet im selben Pfad abgespeichert.
Datenverfolgung	Protokolliert selbst zu definierende Daten. Über die Karteikarte "Datenverfolgung" können Adressen angegeben werden, die auf Wertänderung überwacht werden.

5.2.6 Parameterbereich Datenüberwachung

Datenverfolgung	Beinhaltet eine Liste von Datenblöcken, die in der Traceausgabe aufgezeichnet werden. Dazu muss auf der Karteikarte "Traces" die Option "Datenverfolgung" aktiv sein. Über die Schaltfläche "Neu" wird eine neue Zeile zur Angabe der zu protokollierenden Adresse generiert. Ein Klick auf die neue Zeile erlaubt das Bearbeiten des Zeileninhaltes. Als Eingabe werden folgende Informationen erwartet.
Adresse	Treiberspezifische Adressangabe
Byteoffset	Offset in Byte, der das Startbyte des zu protokollierenden Datenbereiches kennzeichnet.
Byteanzahl	Länge des Datenbereiches in Byte

5.2.7 Frei editierbare Einträge

Fast alle Parameter stehen zur Zeit der Entwicklung eines VisiWin-Treibers fest. Nur selten gibt es Ausnahmen, die es verlangen zum Zeitpunkt der Konfiguration eines VisiWin-Treibers neue Parameter und dann auch deren Werte zusätzlich zu den fixen Parametern hinzuzufügen. Mit der folgenden Dialogbox hat man dazu die Möglichkeit. Sie kann durch die Schaltfläche **[Extras]** im Konfigurator aufgeschlagen werden. Diese Schaltfläche wird aber nur bei einigen VisiWin-Treibern aktiv geschaltet.

Beim Aufruf des Dialoges werden alle nicht fest programmierten Parameter in eine Liste eingetragen. Durch Anklicken eines Eintrags in der Spalte **Eintrag** oder in der Spalte **Wert** kann dieser verändert werden.

Um ein Listenfeld in den Editiermodus zu schalten, selektiert man erst die Zeile, in der ein Parameter verändert werden soll. Danach klick man mit der Maus auf den Eintrag oder den Wert, der editiert werden soll. Bis das Listenfeld in den Editiermodus schaltet vergeht eine kurze Zeitspanne.



- OK** Übernimmt die Einträge und deren Werte in die Konfiguration der VisiWin-Treibers für das aktuelle Projekt. Danach wird der Dialog beendet.
- Abbruch** Beendet den Dialog, ohne die Parameter und Werte zu übernehmen.
- Neuer Eintrag** Fügt eine neue Zeile in die Liste ein. Eintrag und Wert muss dann über die Auswahl des Listenfeldes neu gesetzt werden.
- Löschen** Löscht die selektierte Zeile aus der Liste.

6 Der VisiWin-Treiber

6.1 Das VisiWin-Treiber-Programm

Der VisiWin-Treiber transportiert Daten über ein sogenanntes Gerät zur Prozessdatenbank der Applikation bzw. umgekehrt. Dabei kann ein Gerät z.B. eine serielle Schnittstelle oder eine PC-Einsteckkarte sein, an der eine SPS angeschlossen ist.

6.2 Start des VisiWin-Treibers

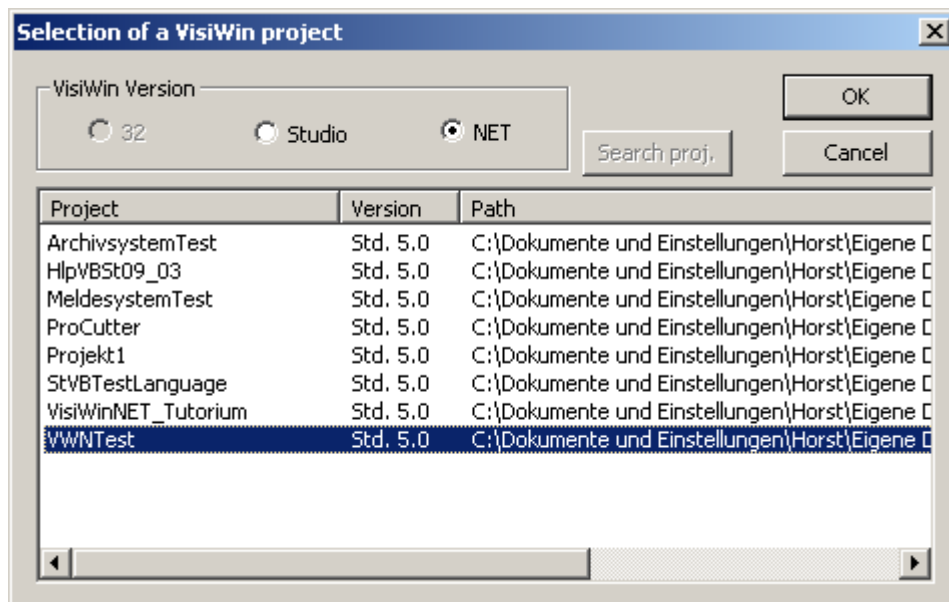
Es gibt zwei mögliche Arten, einen VisiWin-Treiber zu starten.

6.2.1 Start über die Visualisierung

Wird ein VisiWin-Treiber in eine Visualisierungsapplikation eingebunden, so wird er beim Start der Applikation durch den Variablenkern gestartet.

6.2.2 Direkter Start des VisiWin-Treibers [Testmodus]

Ist ein Projekt einmal mit einem VisiWin-Treiber konfiguriert, kann er auch direkt über den Explorer oder das Windows Startmenü gestartet werden. Der VisiWin-Treiber greift dann nicht auf die Prozessdatenbank der Applikation zu, sondern läuft im Testmodus und liest zyklisch die Testadresse. Die Parametrierung erhält der VisiWin-Treiber aus der Konfiguration, die zu der gewählten Applikation gehört.

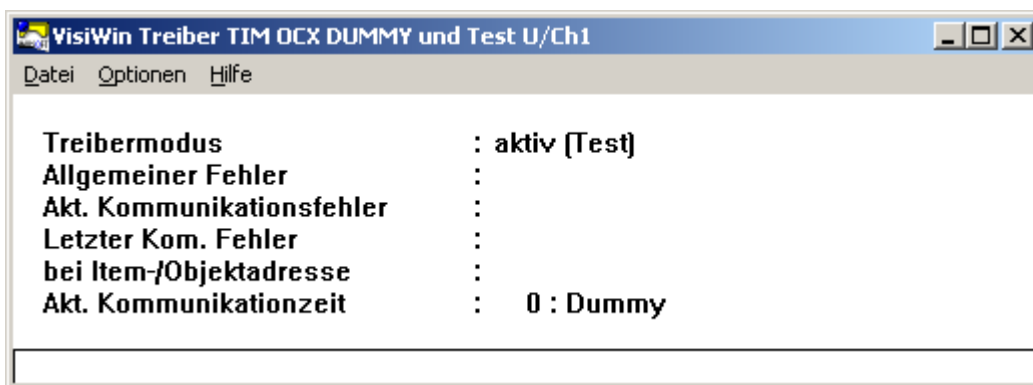


6.3 Bedienung des VisiWin-Treibers

6.3.1 Die Fensteroberfläche

Wird in der Konfiguration des VisiWin-Treibers der Parameter Fensterstil auf mindestens "**Fenster sichtbar**" gestellt, ist nach dem Start folgende Fensteroberfläche sichtbar.

In der Titelzeile wird der Name des VisiWin-Treibers, sowie die verwendete Universalschnittstelle (Kanal) ausgegeben. Die Ausgaben im Fenster werden zyklisch aktualisiert.



Treibermodus	aktiv- der VisiWin-Treiber läuft.
	nicht aktiv- der VisiWin-Treiber wurde mit dem Parameter "Fensterstil 4-manueller Start" gestartet oder er konnte die Protokollinitialisierung nicht vornehmen. In letzterem Falle wird eine Fehlermeldung ausgegeben. Hinter der Ausgabe wird ein (Test) ausgegeben, wenn der Treiber im Testmodus, also mit der Testadresse läuft.
Allgemeiner Fehler	Kommt es zu Fehlern, die nicht vom Protokoll verursacht wurden, sondern durch einen fehlerhaften internen Ablauf im VisiWin-Treiberprogramm, so wird an dieser Stelle eine Meldung mit dem entsprechenden Fehlertext ausgegeben.
Akt. Kommunikationsfehler	Hat ein Schreib- oder Leseauftrag einen protokollabhängigen Fehler erzeugt, so wird hier eine entsprechende Fehlermeldung ausgegeben. Hat die letzte Kommunikation keinen Fehler mehr hervorgerufen, so wird diese Textzeile gelöscht.
Letzter Kom. Fehler	Der zuletzt aufgetretene Kommunikationsfehler wird in diese Zeile mit Zeitstempel und fehlerverursachender Strukturadresse / Variablenadresse eingetragen.

bei Objektadresse	Die Strukturadresse / Variablenadresse, bei der der letzte Kommunikationsfehler aufgetreten ist, wird hier zur weiteren Information nochmals separat dargestellt, bis ein neuer Eintrag ausgegeben wird.
Akt. Kommunikationszeit	Die Zeit in Millisekunden, die der letzte Lese- oder Schreibauftrag gedauert hat wird an dieser Stelle ausgegeben. Dabei ist zu beachten, das Windows 95/98 Systeme eine Zeitauflösung von 1 Millisekunde, Windows XP/Vista meist eine Zeitauflösung von 10 Millisekunden haben. Einige Treiber setzten aber diese Zeitauflösung auch bei Windoes XP/Vista auf eine Millisekunde herunter, um eine optimale Performance zu erhalten.

6.3.1.1 Die Menüs

In der Menüleiste sind alle im VisiWin-Treiber verfügbaren Befehle in Gruppen unterteilt und verschiedenen Menüs zugeordnet.

6.3.1.2 Menü [Datei]

Start	Wenn der VisiWin-Treiber mit dem Fensterstil " manueller Start " parametrieret wurde, kann hier die Kommunikation gestartet werden.
Stop	Kommunikation stoppen.
Ende	Das Kommunikationsprogramm wird beendet.



Die Menüpunkte unter [**Datei**] sollten nur verwendet werden, wenn der VisiWin-Treiber direkt gestartet wurde. Ist der VisiWin-Treiber über eine Applikation gestartet worden, so wird das Starten, Stoppen und Beenden von der Applikation übernommen.

6.3.1.3 Menü [Optionen]

Konfiguration

Ist der Konfigurator und die zugehörige Parameter-Textdatei vorhanden, dann wird der Konfigurator des VisiWin-Treibers gestartet.



Sollten hier Parameter verändert werden, so werden diese erst nach einem Neustart des VisiWin-Treibers übernommen. Das Projekt muss dazu nicht neu übersetzt werden.

Ist der Konfigurator nicht installiert, so wird hier eine Dialogbox angezeigt, in welcher die Konfigurationsparameter angezeigt werden. Eine Veränderung dieser Parameter kann hierüber nicht erfolgen.

Diagnose

Öffnet einen Dialog, weitere Informationen zur Kommunikations ausgegeben werden. Siehe der Diagnose Dialog.

Abbildung

Wenn der VisiWin-Treiber mit dem Fensterstil "Fenster mit Debuginfo" gestartet wurde, erreicht man über dieses Menü ein weiteres Fenster, über das man sich die gelesenen Datenbereiche ansehen oder verändern kann.

Ansicht Log

Öffnet das Diagnosefenster des Treibers. Hier werden die in der Konfiguration unter "Traces" aktivierten Protokollierungen angezeigt.

6.3.1.4 Menü [Hilfe]

spezifische Hilfe

Steht die protokollabhängige Dokumentation im Installationsverzeichnis des VisiWin-Treibers, wird dieses Dokument aufgeschlagen. Ist es nicht vorhanden, gibt es keine Aktion.

allgemeine Hilfe

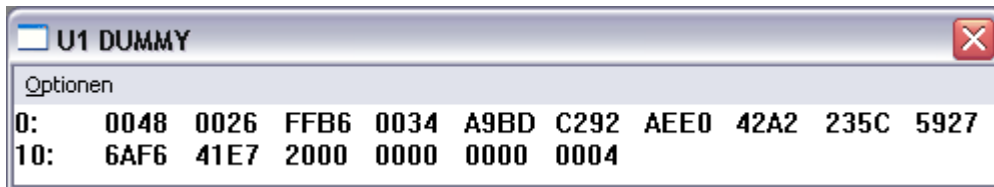
Öffnet dieses Hilfedokument.

Über mich

Zeigt eine Dialogbox mit Versionsnummer, Copyright und weitere Informationen zum VisiWin-Treiber an.

6.4 Anzeigen und Senden von Daten

Nach Aufruf des Menüs **[Optionen\Abbildung]** öffnet sich ein Fenster, in der als Speicherauszug die Nettodaten des ersten Datenbereiches aus der Prozessdatenbank angezeigt werden. Läuft der VisiWin-Treiber im Testmodus, wird der Datenbereich der Testadresse ausgegeben.



Beim Öffnen dieses Fensters ist der gesamte angezeigte Datenbereich mit dem Wert 0 vorbelegt. Erst wenn durch einen neuen Kommunikationszyklus der angezeigte Datenbereich neu ausgelesen wurde, wird auch der Inhalt des Fensters aktualisiert.

Die Titelzeile besteht aus der Nummer der Universalschnittstelle (Kanals), hier 1 und dem Namen der angezeigten Strukturadresse (hier DUMMY). Pro Zeile werden 10 hexadezimale Werte angezeigt. Der Wortoffset des ersten Nettowertes jeder Zeile wird mit Doppelpunkten getrennt vor die Zeile geschrieben.

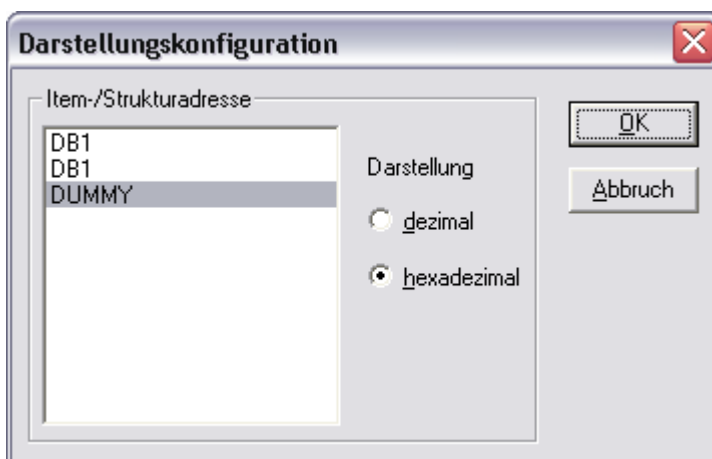


Der Aufbau der Inhalte dieses Fensters bezieht sich nicht auf die in einer Struktur definierten Elemente, sondern gibt lediglich ein Speicherabbild aus. Eine unter den Konfigurationspunkten [Datenüberwachung] eingestellte Datenkonvertierung ist hier noch nicht berücksichtigt.

6.4.1 Das Menü [Optionen]

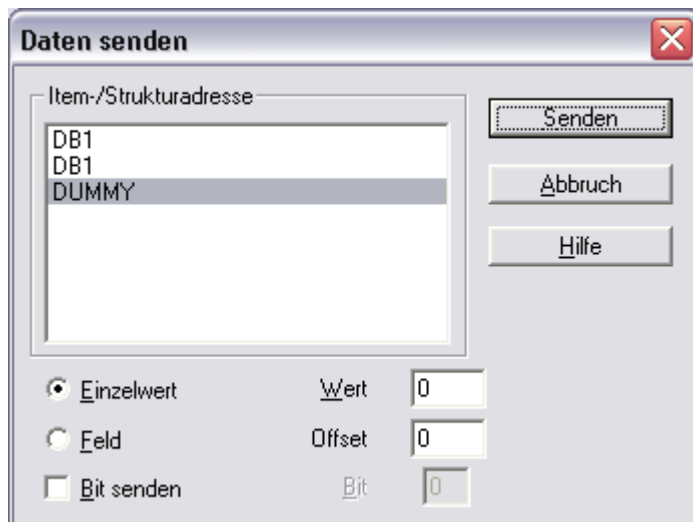
6.4.1.1 Menü [Optionen]

Öffnet eine Dialogbox in der die Darstellung im Datenausgabefenster von dezimal auf hexadezimal umgestellt werden kann. Weiterhin hat man hier die Auswahl, welche Variablenadresse / Strukturadresse ausgegeben werden soll.



6.4.1.2 Menü [Daten senden]

Über die folgende Dialogbox ist es möglich, Werte direkt über den VisiWin-Treiber an eine SPS zu senden.



Variablen/ Strukturadresse

Auswahl der Variablenadresse / Strukturadresse, an die man Daten senden möchten.

Einzelwert

Es soll ein Wort (16Bit) oder ein einzelnes Bit gesendet werden.

Feld

Es soll ein Feld mit mehreren Worten gesendet werden. Das Steuerfeld **Wert** wird dabei umbenannt in **Startwert**, das Steuerfeld **Bit** in **Anzahl** und das Steuerfeld **Bit senden** in **hochzählen**.

Bit senden

Dieses Feld (de)-aktiviert das **Wert** Kontrollfeld und (de)-aktiviert das Kontrollfeld **Bit**.

Hochzählen

Bei Auswahl von **Feld** besagt dieses Kontrollfeld, ob alle zu schreibenden Werte mit dem Wert aus **Startwert** beschrieben werden, oder ob jedes Wort um den Wert 1 inkrementiert werden soll.

Wert

Gibt den zu sendenden Wortwert vor.

Startwert

Bei Auswahl von **Feld** wird hier der Wert des ersten Wortes eingetragen.

Offset

Definiert den Wortoffset des unter **Variablen/Strukturadresse** ausgewählten Eintrages, ab dem geschrieben werden soll.

Bit

Bei **Bit senden** steht hier die Nummer des Bits, welches gesetzt werden soll Die Zählweise ist von 0 bis 15.

Anzahl

Bei Auswahl von **Feld** beschreibt dieser Parameter die Anzahl der Worte, die gesendet werden sollen.

Senden

Übergibt den Sendeauftrag an die Protokollebene des VisiWin-Treibers.

Abbruch

Schließt die Dialogbox.

Hilfe

Öffnet dieses Hilfedokument.

6.4.1.3 Menü [Ende]

Schließt das Datenausgabefenster.

6.5 Datenaustausch zur Prozessdatenbank der Applikation

Dieses Kapitel beschreibt die internen Abläufe zwischen der Prozessdatenbank einer Applikation und der Protokollebene eines VisiWin-Treibers.

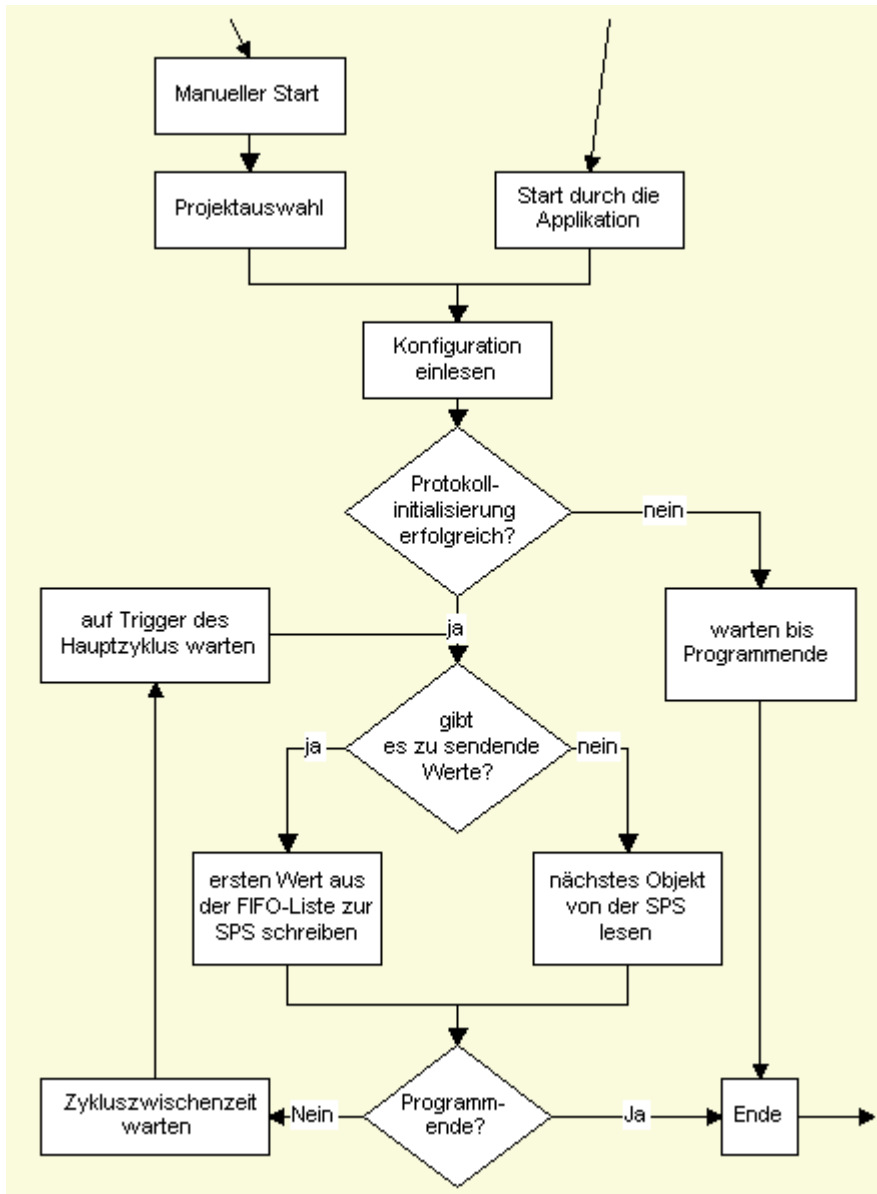
Zwischen der Prozessdatenbank und dem VisiWin-Treiber bestehen zwei Datenbereiche (speicherbasierte Dateien). Die erste Datei beinhaltet alle in einer Projektdatenbank definierten **Variablen**-/Strukturobjekte, die als Einzelvariable, Variablenarray oder Strukturvariable angelegt wurden. Diese Datei hat eine feste Größe, die von der Anzahl der Objekte abhängt. Die Größe ist nicht beeinflussbar. Der zweite Bereich ist für Prozessvariablen, deren Werte zur SPS geschrieben werden sollen. Dieser Bereich ist normalerweise leer. Nur wenn sich der Wert einer Prozessvariablen auf der Seite der Visualisierung geändert hat, wird in dieser Datei ein Objekt erzeugt. Diese Datei hat eine Vorgabegröße. Diese ist abhängig von der Anzahl der Objekte in der Datenbank. Die kleinste Größe hat Platz für 500 Objekte. Es kann unter Umständen nötig sein, diese Anzahl zu vergrößern.



Hat man z.B. eine langsame Kommunikation (seriell mit geringer Baudrate) oder wenn die Zykluszeiten von Prozessdatenbank und VisiWin-Treiber schlecht aufeinander abgestimmt wurden und viele einzelne Werte zur SPS gesendet werden (bei Rezepten), kann es vorkommen, dass der VisiWin-Treiber die Aufträge nicht schnell genug abarbeiten kann. Das hat zur Folge, dass die Schreibliste voll läuft. In diesem Fall erscheint eine Systemfehlermeldung in den Alarmsteuerelementen mit dem Namen des VisiWin-Treiberprogramms.

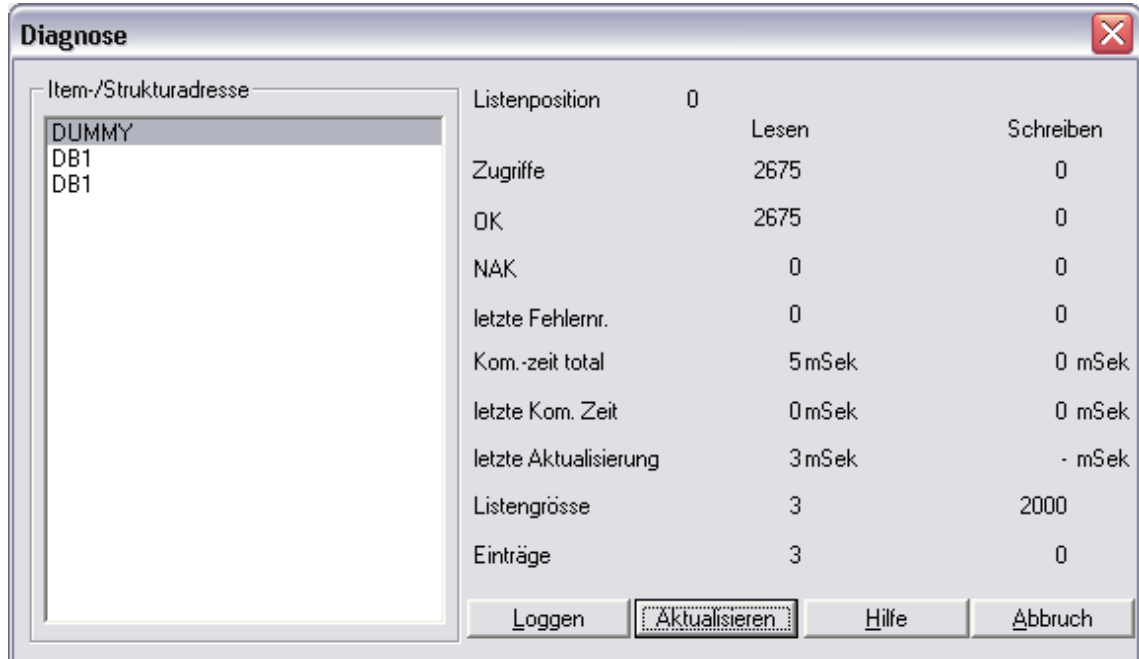
Sollte dies eintreten, so vergrößert man die Anzahl der Listeneinträge im Dialog der Laufzeitkonfiguration. Man sollte zuvor aber versuchen, die Kommunikation zu optimieren. Weitere Informationen stehen dazu im Kapitel "Optimierung der Kommunikation".

Der Programmablauf eines VisiWin-Treibers kann vereinfacht in folgendem Diagramm dargestellt werden:



6.6 Der Diagnose Dialog

Über den Menüpunkt Optionen/Diagnose öffnet sich folgender Dialog.



In der Auflistung Variablen-/Strukturadresse werden alle dem Treiber zur Zeit bekannten Objekte aufgelistet. Durch Auswählen über die Maus oder die Tastatur eines Objektes, werden dessen Daten in der rechten Seite des Dialogs ausgegeben. Analog zur Spalte Lesen, in der sich alle ausgegebenen Werte auf das Lesen von Daten aus der SPS beziehen, gibt es die Spalte Schreiben, dessen Daten die Schreibaufträge zur SPS darstellt.

Listenposition	Gibt die Position des ausgewählten Objekte in der jeweiligen Liste an (interne Information).
Zugriffe*	Gibt die Anzahl aller Lese- bzw. Schreibaufträge seit Starte des VisiWin-Treibers an.
OK*	Anzahl der Aufträge, die ohne Kommunikationsfehler abgeschlossen wurden.
NAK*	Anzahl der Aufträge, die mit einem Kommunikationsfehler abgeschlossen wurden.
Letzte Fehlernummer	Nummer des letzten aufgetretenen Kommunikationsfehlers.
Kommunikationszeit total*	Auffsummierung aller Kommunikationszeiten, die fehlerlos abgelaufen sind. Dies ist nicht 100%tig genau, da die Zeitauflösung eine Rolle spielt und das Ergebnis verfälscht.
Letzte Kommunikationszeit	Letzte Kommunikationszeit, in Millisekunden, eines Auftrages ohne Fehler.
Letzte Aktualisierung	Die letzte Aktualisierung des Objektes war vor x Millisekunden. Dabei wird die Zeit bei Auswahl des Objektes bzw. bei Betätigen der Schaltfläche Aktualisieren gemessen.

Listengröße	Größe der jeweiligen Liste (Interne Information).
Einträge	Anzahl der Einträge in die Jeweilige Liste.
* Nach einer längeren Laufzeit des VisiWin-Treiber kann es zu einem Überlauf des 32 Bit Wertes kommen.	
Loggen	Ist eine beliebige Traceoption aktiviert, so werden die zur Zeit angezeigten Werte in das Logfile aufgezeichnet.
Aktualisieren	Aktualisiert die Daten für das gewählte Objekt.
Hilfe	Öffnet dieses Dokument.
Abbruch	Schließt die Dialogbox.

6.7 Optimierung der Kommunikation

Die Optimierung der Kommunikation hängt von vielen Faktoren ab und kann daher leider nicht über ein "Patentrezept" gelöst werden. Die folgende Liste enthält Tipps, die schon bei der Erstellung der Prozessdatenbank berücksichtigt werden sollte. Einige dieser Punkte wirken sich bis zur Programmierung der SPS aus und sollten daher auch mit dem SPS-Programmierer besprochen werden.

- Es ist immer sinnvoll, die Menge der Daten die zyklisch gelesen werden, zu minimieren. Dazu verwendet man wenn möglich Bild- und Abtaststrukturen. Gruppen dienen dazu, den Datenfluss zu steuern und damit zu minimieren.
- Daraus resultiert, dass einzelne Prozessvariablen zu Elementen in Strukturen oder Array's nach ihrer Verwendung in der Applikation zusammengelegt werden sollten. Diese Strukturen und Array's sollten dann von der Aktualisierungsrate her in Gruppen zusammengefasst werden, um den Datenfluss zur SPS zu steuern.
- Bei Strukturen, Array's und Einzelvariablen, deren Daten nicht unbedingt möglichst schnell aktualisiert werden müssen, kann der Abtastzyklus erhöht werden. Setzt man z.B. den Abtastzyklus auf 7, so wird die Prozessvariable vom VisiWin-Treiber nur jeden siebten Zyklus der Laufzeit (in Millisekunden) gelesen. Bei der Verwendung ist die Prozessvariable in eine Gruppe zu definieren, dessen Aktualisierungsrate beeinflusst werden kann.
- Daten die nur in Richtung SPS geschrieben werden, sollten in Strukturen zusammengefasst werden, die das Attribut **nur schreibend / Write** oder **einmal lesen dann schreiben** haben.
- Strukturen sollten nicht unnötig viele unbenutzte Bereiche enthalten. Somit verringert sich das Datenvolumen.
- Im Allgemeinen kann man sagen, dass die Größe von Strukturen möglichst voll ausgenutzt werden sollte. Viele kleine Strukturen vergrößern den Verwaltungsaufwand und verschlechtern das Verhältnis zwischen übertragenen Nettodaten und dem Protokoll-Overhead. Hierbei ist es dann immer schwierig, einen Kompromiss zu schließen zum vorhergehenden Punkt (Größe der Strukturen). Ob es jetzt sinnvoller ist, eine Struktur mit unbenutzten Bereichen aufzubauen und dafür einen Block zu übertragen oder die Struktur aufzuteilen, hängt dann auch von der Art des Protokolls ab. Ist die Reaktionszeit der SPS schlecht oder das Protokoll selbst schnell, so empfiehlt es sich, Lücken im übertragenen Datenbereich in Kauf zu nehmen. Anderenfalls sollte man lieber die Datenbereiche aufteilen.

- Der Variablenkern ist mit der Prozessdatenbankabarbeitung sehr schnell, aber trotzdem sollte man für dessen Zyklus sinnvolle Werte einstellen. Normalerweise reichen Zykluszeiten von 200 bis 500 msec. aus, um Refreshraten von 2 bis 5 mal pro Sekunde zu erreichen. Unter Windows XP/Vista kann über den Taskmanager [**Ctrl-Alt-Del**] die Systemauslastung des VisiWinNET Manager betrachtet werden. Dieser sollte das System mit max. 20% auslasten. Ist diese Auslastung noch höher ist die Zykluszeit des Variablenkerns oder die Abtastraten der Gruppen zu hoch eingestellt. Dies wiederum verringert deutlich die Performance der VisiWin-Treiber.
- Der VisiWin-Treiber muss immer schneller als der Variablenkern eingestellt werden (min Faktor 3). Besteht gar keine Vorstellung über die Geschwindigkeit der Kommunikation mit der SPS, so kann man beim Betrieb des VisiWin-Treibers sein Fenster beobachten. In der untersten Zeile wird die Kommunikationszeit der letzten bearbeiteten Variablen/Struktur ausgegeben. Betrachtet man eine Zeit lang diese Zeiten, so stellt man fest, dass sie manchmal recht konstant sind, manchmal unterliegen sie aber auch starken Schwankungen. Grundsätzlich ist es sinnvoll vom kleinsten Wert auszugehen. Ist dieser z.B. 15 Millisekunden, so ist der Hauptzyklus des VisiWin-Treiber auf einen Wert zwischen 10 und 15 Millisekunden zu stellen. Kleinere Werte machen den VisiWin-Treiber nicht schneller, sondern vergrößern nur etwas die Systemauslastung!
- Über den Taskmanager [**Ctrl-Alt-Del**] können die laufenden Prozesse betrachtet werden. Der VisiWin-Treiber sollten dabei unter 30% CPU Auslastung liegen (normalerweise deutlich weniger, ca. 0-5%). Ist dies nicht der Fall, so sollte der Parameter **Zykluszwischenzeit** etwas erhöht werden. Damit überlässt man allen anderen laufenden Applikationen mehr CPU Zeit. Ein Wert über 10 Millisekunden sollte aber nicht eingestellt werden!