



METACOM Datentreiber **Telegrammdefinition**

Dokument T98024-11

Die in diesen Unterlagen enthaltenen Angaben können ohne vorherige Ankündigung geändert werden. Dieses Handbuch darf ohne vorherige schriftliche Genehmigung von Schille Informationssysteme weder als Ganzes noch in Auszügen vervielfältigt oder an Dritte weitergegeben werden.

© Copyright 1998-2001 Schille Informationssysteme. Alle Rechte vorbehalten

Dokument T98024-11

Geänderte Ausgabe vom September 2001

Die Dokumentation beschreibt METACOM-Datentreiber (DOS und NT) ab Version 9.00

Technische Änderungen vorbehalten

Alle erwähnten Produkte sind Warenzeichen oder eingetragene Warenzeichen ihrer jeweiligen Hersteller.

1. Einführung	7
2. Definition der Telegramm Daten	7
2.1 'T' - TRANSMIT : Ereignisgesteuerte Sendedefinition	8
2.1.1 Beispiele :	8
2.1.2 Feld <i>Antwort</i>	8
2.1.3 Feld <i>Verzögerung</i>	8
2.1.4 Feld <i>Adresse</i>	9
2.1.5 <i>Definitionen</i>	9
2.1.5.1 Konstanten	10
2.1.5.2 Variablen	11
BN adresse:länge - binär	11
HX adresse:länge - hexadezimal	11
AL adresse:formatlänge ASCII, linksbündig, ein Zeichen (1 Byte)	12
AR adresse:formatlänge ASCII, rechtsbündig, ein Zeichen (1 Byte)	12
AZ adresse:formatlänge ASCII, mit Vornullen, ein Zeichen (1 Byte)	12
AL adresse:formatlänge ASCII, linksbündig, zwei Zeichen (2 Byte = 1 Wort)	13
AR adresse:formatlänge ASCII, rechtsbündig, zwei Zeichen (2 Byte = 1 Wort)	13
AZ adresse:formatlänge ASCII, mit Vornullen, zwei Zeichen (2 Byte = 1 Wort)	13
ALL adresse:formatlänge ASCII, linksbündig, vier Zeichen (4 Byte = 1 Langwort)	14
ARL adresse:formatlänge ASCII, rechtsbündig, vier Zeichen (4 Byte = 1 Langwort)	14
AZL adresse:formatlänge ASCII, mit Vornullen, vier Zeichen (4 Byte = 1 Langwort)	14
BS adresse:länge ASCII, Bitsatz, der Inhalt wird als Folge von '0' und '1' definiert	15
VZ adresse:formatlänge ASCII, mit Vornullen, ein Zeichen (1 Byte) mit Vorzeichen	15
VZ adresse:formatlänge ASCII, mit Vornullen, zwei Zeichen (1 Wort) mit Vorzeichen	15
VZL adresse:formatlänge ASCII, mit Vornullen, 4 Zeichen (1 Langwort) mit Vorzeichen	16
2.1.6 Versenden von Daten ohne Protollrahmen	16
2.2 'C' - CYCLED : Zyklische Sendedefinition	17
2.2.1 Beispiele :	17
2.2.2 Feld <i>Antwort</i>	17
2.2.3 Feld <i>Zykluszeit</i>	17
2.2.4 Feld <i>Verzögerung</i>	17
2.2.5 <i>Definitionen</i>	17
2.2.6 Versenden von Daten ohne Protollrahmen	17
2.3 'A' - ANSWER : Antwortdefinition	18
2.3.1 Beispiele :	18
2.3.2 Feld <i>Antwort</i>	18
2.3.2 Feld <i>Adresse</i>	18
2.3.3 <i>Definitionen</i>	19
2.3.3.1 Zeichen ohne Relevanz	19
2.3.3.2 Konstanten	19
2.3.3.3 Speicherzuweisungen	20
BN adresse:länge - binär	20
HX adresse:länge - hexadezimal	20
AL adresse:formatlänge ASCII, linksbündig, ein Zeichen (1 Byte)	20
AR adresse:formatlänge ASCII, rechtsbündig, ein Zeichen (1 Byte)	20
AZ adresse:formatlänge ASCII, mit Vornullen, ein Zeichen (1 Byte)	21
AL adresse:formatlänge ASCII, linksbündig, zwei Zeichen (2 Byte = 1 Wort)	21
AR adresse:formatlänge ASCII, rechtsbündig, zwei Zeichen (2 Byte = 1 Wort)	21
AZ adresse:formatlänge ASCII, mit Vornullen, zwei Zeichen (2 Byte = 1 Wort)	21
ALL adresse:formatlänge ASCII, linksbündig, vier Zeichen (4 Byte = 1 Langwort)	22
ARL adresse:formatlänge ASCII, rechtsbündig, vier Zeichen (4 Byte = 1 Langwort)	22
AZL adresse:formatlänge ASCII, mit Vornullen, vier Zeichen (4 Byte = 1 Langwort)	22
BS adresse:länge ASCII, Bitsatz, der Inhalt wird als Folge von '0' und '1' interpretiert	22
VZ adresse:formatlänge ASCII, mit Vornullen, ein Zeichen (1 Byte) mit Vorzeichen	23

VZWadresse:formatlänge	ASCII, mit Vornullen, zwei Zeichen (1 Wort) mit Vorzeichen	23
VZLadresse:formatlänge	ASCII, mit Vornullen, 4 Zeichen (1 Langwort) mit Vorzeichen	23
2.4 'R' - RECEIVE : Empfangsdefinition		24
2.4.1 Feld <i>Adresse</i>		24
2.4.2 <i>Definitionen</i>		25
2.4.2.1 Zeichen ohne Relevanz		25
2.4.2.2 Konstanten		25
2.4.2.3 Speicherzuweisungen		25
2.5 'W' Zyklische Sendedefinition mit Bedingung		25
2.5.1 Beispiel :		25
2.5.2 Feld <i>Antwort</i>		25
2.5.3 Feld <i>Zykluszeit</i>		25
2.5.4 Feld <i>Verzögerung</i>		25
2.5.5 <i>Bedingung</i>		25
2.5.6 <i>Definitionen</i>		25
2.6 'X' - EXTENDED : Treiberspezifische Informationen und Verbindungstest		26
2.6.1 Beispiel :		26
2.7 'xR' - RAW : Senden von Rohdaten		26
2.7.1 Beispiel :		26
2.8 'Yn' - AUTOCONFIG : Sonderbefehle der Schnittstellenassistenten		26
3. Konfiguration der Schnittstelle		27
3. Konfiguration der Schnittstelle		27
3.1 Eintrag Port		27
3.2 Eintrag StatusPortLow und StatusPortHigh (DOS-Plattform)		28
3.3 Eintrag Address (DOS-Plattform)		28
3.4 Eintrag IRQ (DOS-Plattform)		29
3.5 Eintrag Parity		29
3.6 Eintrag Handshake		29
3.7 Eintrag Baudrate		29
3.8 Eintrag DataBits		29
3.9 Eintrag StopBits		29
3.10 Eintrag FIFO (DOS-Plattform)		30
4. Sonstige Konfigurationen		30
4.1 Eintrag LoadFrom		30
4.2 Eintrag SaveTo		30
4.3 Eintrag MemorySize		30

1. Einführung

Die *Metacom*-Datentreiber stellen eine neue Qualität der PVis-Datentreiber dar. Über eine einheitliche, also geräteunabhängige 'Definitionssprache' erfolgt die Anlage der Dateninhalte. Der Treiber selbst sorgt nur für die Behandlung des jeweils spezifizierten Protokolls, also beispielsweise der Empfangsbestätigungen oder der Generierung von Prüfsummen.

Metacom-Treiber können mehrfach installiert werden, d.h. an mehreren Schnittstellen können Gegenstellen angeschlossen werden. Hierzu ist nur die Treiberdatei umzubenennen und nochmals zu installieren, das Leitsystem sorgt selbstständig für die korrekte Einbindung unter dem neuen Treibernamen.

Die Definition der Telegrammdatei, also des Inhalts der Kommunikation zwischen Leitsystem und Gegenstelle erfolgt über eine Definitionsdatei. Der Name der Datei bildet sich aus dem Treibernamen und der Erweiterung *.DEF*. Die Datei muß sich im PVis-Arbeitsverzeichnis befinden, d.h. dem Verzeichnis, in dem auch die Programmdatei *PVIS.EXE* bzw. *WINPVIS.EXE* liegt. Die Definitionen sind einfache, im ASCII-Format anzulegende Befehle und können mit einem Texteditor bearbeitet werden. Jede Zeile enthält eine Definition, wobei eine Zeile maximal 255 Zeichen lang sein darf. Die Anzahl der Definitionen ist nicht begrenzt. Innerhalb einer Definition werden die Anweisungen durch ein Komma getrennt. Eine weitere Möglichkeit zur Anlage der Definitionsdatei ist die Verwendung eines Tabellenkalkulationsprogrammes (StarCalc, Lotus 123, Excel etc.) und der Export in eine ASCII- bzw. CSV-Datei (Satztrennung durch Komma und Zeichensatz MS-DOS bei der DOS-Plattform angeben!).

Als Schnittstellen können bei *Metacom*-Datentriibern die vier Standardschnittstellen (COM1 bis COM4), Schnittstellenkarten mit abweichenden Portadressen und Multiserial-Karten vom Typ AST und ARNET verwendet werden. Für die Kommunikation stehen damit bis zu 16 unabhängige serielle Schnittstellen zur Verfügung, die mit jeweils bis zu 115.000 Baud betrieben werden. Auf der NT-Plattform stehen die Ports COM1 bis COM 255 über die betriebssysteminternen Treiber zur Verfügung. Neben seriellen Schnittstellentreibern stehen unter der NT-Plattform auch TCP/IP-basierende Schnittstellentreiber zur Verfügung, deren Dateninhalt auch über die *Metacom*-Spzifikation beschrieben wird.

2. Definition der Telegrammdatei

Im folgenden werden die Befehle der Definitionen beschrieben. Die angegebenen Beispiele sind allgemein gehalten und entsprechen nicht dem wirklichen Dateninhalt des Treibers. Dieser ist in der jeweiligen Treiberdokumentation beschrieben, bzw. aus den Handbüchern der Gegenstelle ersichtlich.

2.1 'T' - TRANSMIT : Ereignisgesteuerte Sendedefinition

Ereignisgesteuerte Sendungen reagieren auf Änderungen im Datenbereich des Treibers. Die Sendung erfolgt unmittelbar oder nach einer vorgegebenen Verzögerung. Werden Daten von der Gegenstelle nach der Sendung erwartet, so können diese als Antwort definiert werden.

1	2	3	4	5 - ...
Kennung 'T'	Antwort	Verzögerung	Adresse	Definitionen

2.1.1 Beispiele :

T,0,0,100:1,'T','e','s','t'

Bei Änderung des untersten Bits (00000001) auf Adresse 100 wird, ohne Erwartung eines Antworttelegrammes der Text 'Test' gesendet.

T,10,2000,500:8-0,'T','e','s','t'

Bei Löschen des höchstwertigen Bits (10000000) auf Adresse 500 wird, in Erwartung der 10. definierten Antwortdefinition der Text 'Test' nach einer Verzögerung von 2 sec gesendet.

T,100,0,0:8-1,BN1000:50

Bei Setzen des höchstwertigen Bits (10000000) auf Adresse 0 wird, in Erwartung der 100. definierten Antwortdefinition der Speicherinhalt von Adresse 1000 bis 1049 binär gesendet.

T,100,0,100,HX100:8,HX200:8

Bei einer Änderung des Inhaltes von Adresse 0 wird, in Erwartung der 100. definierten Antwortdefinition der Speicherinhalt von Adresse 100 bis 107 und von Adresse 200 bis 207 hexadezimal (ASCII) gesendet.

T,100,0,0:8-1&200=10,BN1000:50

Bei Setzen des höchstwertigen Bits (10000000) auf Adresse 0 und einem Wert von 10 auf der Adresse 200 wird, in Erwartung der 100. definierten Antwortdefinition der Speicherinhalt von Adresse 1000 bis 1049 binär gesendet.

2.1.2 Feld Antwort

In diesem Feld ist zu spezifizieren, ob eine bestimmte Antwort der Gegenstelle erwartet wird. Die Auswahl erfolgt durch Eintragen der Definitionsnummer (>0, siehe 2.3 Antwortdefinition).

2.1.3 Feld Verzögerung

Soll die Sendung nicht unmittelbar nach der Auslösung (siehe Feld Adresse) erfolgen, so kann hier eine Verzögerungszeit in ms angegeben werden. Die Auflösung beträgt etwa 20 ms.

2.1.4 Feld Adresse

Die Bedingung zur Auslösung der Sendung wird in diesem Feld spezifiziert. Durch Angabe einer Adresse und, optional eines Bits und der Flanke, wird die zu überprüfende Speicherstelle im Treiber angegeben. Die Speicherstellen können vom Leitsystem (Datenpunkte) oder durch eingehende Telegramme (Antwortdefinition 'A' und Empfangsdefinition 'R') verändert werden. Weiterhin kann eine zusätzliche Adresse auf einen Wert hin geprüft werden.

Die Syntax lautet :

Adresse

oder

Adresse:Bitmaske

oder

Adresse:Bitmaske-Flanke

bzw.

Adresse&Zweitadresse=Wert

oder

Adresse:Bitmaske&Zweitadresse=Wert

oder

Adresse:Bitmaske-Flanke&Zweitadresse=Wert

Die Adresse muß innerhalb des Speicherbereichs des Treibers liegen, als Bit ist ein Wert von 1 bis 8 einzutragen. Die Flanke ist als -0 (Löschen des Bits) oder -1 (Setzen des Bits) anzugeben.

Beispiele :

100	Auslösung bei Änderung des Speicherinhalts auf Adresse 100
100:4	Auslösung bei Änderung des 4. Bits (00001000) auf Adresse 100
100:8-1	Auslösung bei Setzen des 8. Bits auf Adresse 100
100:8-0	Auslösung bei Löschen des 8. Bits auf Adresse 100

Die Prüfung der Zweitadresse erfolgt in der Form
Zweitadresse=Wert.

Das Telegramm wird nur verschickt, wenn, nach Auslösung durch die 1. Adressedefinition, der *Wert* an der *Zweitadresse* vorgefunden wird.

2.1.5 Definitionen

Der zu sendende Dateninhalt wird in den Definitionen angegeben. Die Anzahl der Definitionen einer Sendung ist auf 128 begrenzt. Neben konstanten Werten können Speicherbereiche des Treibers als Variablen in unterschiedlichen Formaten gesendet werden. Die Definitionen können beliebig kombiniert werden.

2.1.5.1 Konstanten

Konstanten sind 1 Zeichen lang und können somit einen Wertebereich von 0 bis 255 einnehmen. Zwischen folgenden Formaten kann gewählt werden :

nn numerischer Wert
\$hh numerischer Wert, hexadezimaler Format
'c' ASCII-Zeichen

Beispiel :
0,100,\$F0,'A','z'

Zeichenketten können auch zusammengefaßt werden :

'the quick brown fox jumps over the lazy dog' ist ebenfalls möglich. Das Hochkomma kann innerhalb des Textes nicht verwendet werden und ist durch Einfügen des Wertes 96 oder \$60 zu ersetzen.

2.1.5.2 Variablen

Variablen stellen Speicherinhalte des Treibers dar und sind durch folgende Syntax anzugeben :

typ[adresse]:[länge]

Folgende Typen stehen zur Auswahl :

BNadresse:länge - binär

Die Daten ab der angegebenen Adresse bis zur Adresse+Länge-1 werden binär ausgegeben.

Beispiele :

- BN100:1 Der Wert von Adresse 100 wird binär gesendet.
Der gesendete Dateninhalt ist ein Zeichen lang.
- BN100:64 Der Wert von Adresse 100 bis 163 wird binär gesendet.
Der gesendete Dateninhalt ist 64 Zeichen lang.

HXadresse:länge - hexadezimal

Die Daten ab der angegebenen Adresse bis zur Adresse+Länge-1 werden hexadezimal als ASCII-Zeichen ausgegeben. Jede Speicherstelle wird durch zwei Zeichen (00 bis FF) repräsentiert.

Beispiele :

- HX100:1 Der Wert von Adresse 100 wird hexadezimal gesendet.
Der gesendete Dateninhalt ist zwei Zeichen lang.

Zeichen	1	2
Wert 0	0	0
Wert 255	'F'	'F'

- HX100:64 Der Wert von Adresse 100 bis 163 wird hexadezimal gesendet.
Der gesendete Dateninhalt ist 128 Zeichen lang.

Zeichen	1	2	3	4	5	6	7	8	...	127	128
Adresse	100		101		102		103			163	
Wert 0	'0'	'0'	'0'	'0'	'0'	'0'	'0'	'0'		'0'	'0'
Wert 255	'F'	'F'	'F'	'F'	'F'	'F'	'F'	'F'		'F'	'F'

ALBadresse:formatlänge ASCII, linksbündig, ein Zeichen (1 Byte)

Ein Wert an der angegebenen Adresse wird numerisch im ASCII-Format ausgegeben.
Die Ausgabe erfolgt linksbündig formatiert mit der angegebenen Formatlänge. Der Wertebereich ist 0 bis 255.

Beispiel :

ALB100:5 Der 8-Bit Wert von Adresse 100 wird numerisch gesendet.
Der gesendete Dateninhalt ist fünf Zeichen lang.

Zeichen	1	2	3	4	5
Wert 0	0				
Wert 255	2	5	5		

ARBadresse:formatlänge ASCII, rechtsbündig, ein Zeichen (1 Byte)

Ein Wert an der angegebenen Adresse wird numerisch im ASCII-Format ausgegeben.
Die Ausgabe erfolgt rechtsbündig formatiert mit der angegebenen Formatlänge. Der Wertebereich ist 0 bis 255.

Beispiel :

ARB100:5 Der 8-Bit Wert von Adresse 100 wird numerisch gesendet.
Der gesendete Dateninhalt ist fünf Zeichen lang.

Zeichen	1	2	3	4	5
Wert 0					0
Wert 255			2	5	5

AZBadresse:formatlänge ASCII, mit Vornullen, ein Zeichen (1 Byte)

Ein Wert an der angegebenen Adresse wird numerisch im ASCII-Format ausgegeben.
Die Ausgabe erfolgt rechtsbündig mit Vornullen formatiert in der angegebenen Formatlänge.
Der Wertebereich ist 0 bis 255.

Beispiel :

AZB100:5 Der 8-Bit Wert von Adresse 100 wird numerisch gesendet.
Der gesendete Dateninhalt ist fünf Zeichen lang.

Zeichen	1	2	3	4	5
Wert 0	0	0	0	0	0
Wert 255	0	0	2	5	5

ALWadresse:formatlänge ASCII, linksbündig, zwei Zeichen (2 Byte = 1 Wort)

Ein Wert an der angegebenen Adresse wird numerisch im ASCII-Format ausgegeben.
Die Ausgabe erfolgt linksbündig formatiert mit der angegebenen Formatlänge. Der Wertebereich ist 0 bis 65535.

Beispiel :

ALW100:5 Der 16-Bit Wert von Adresse 100 und 101 wird numerisch gesendet.
Der gesendete Dateninhalt ist fünf Zeichen lang.

Zeichen	1	2	3	4	5
Wert 0	0				
Wert 65535	6	5	5	3	5

ARWadresse:formatlänge ASCII, rechtsbündig, zwei Zeichen (2 Byte = 1 Wort)

Ein Wert an der angegebenen Adresse wird numerisch im ASCII-Format ausgegeben.
Die Ausgabe erfolgt rechtsbündig formatiert mit der angegebenen Formatlänge. Der Wertebereich ist 0 bis 65535.

Beispiel :

ARW100:5 Der 16-Bit Wert von Adresse 100 und 101 wird numerisch gesendet.
Der gesendete Dateninhalt ist fünf Zeichen lang.

Zeichen	1	2	3	4	5
Wert 0					0
Wert 65535	6	5	5	3	5

AZWadresse:formatlänge ASCII, mit Vornullen, zwei Zeichen (2 Byte = 1 Wort)

Ein Wert an der angegebenen Adresse wird numerisch im ASCII-Format ausgegeben.
Die Ausgabe erfolgt rechtsbündig mit Vornullen formatiert in der angegebenen Formatlänge.
Der Wertebereich ist 0 bis 65535.

Beispiel :

AZW100:5 Der 16-Bit Wert von Adresse 100 und 101 wird numerisch gesendet.
Der gesendete Dateninhalt ist fünf Zeichen lang.

Zeichen	1	2	3	4	5
Wert 0	0	0	0	0	0
Wert 65535	6	5	5	3	5

ALLadresse:formatlänge ASCII, linksbündig, vier Zeichen (4 Byte = 1 Langwort)
 Ein Wert an der angegebenen Adresse wird numerisch im ASCII-Format ausgegeben.
 Die Ausgabe erfolgt linksbündig formatiert in der angegebenen Formatlänge. Der Wertebereich ist 0 bis 2147483647!

Beispiel :

ALL100:10 Der 32-Bit Wert von Adresse 100 bis 103 wird numerisch gesendet.
 Der gesendete Dateninhalt ist zehn Zeichen lang.

Zeichen	1	2	3	4	5	6	7	8	9	10
Wert 0	0									
Wert 2147483647	2	1	4	7	4	8	3	6	7	7

ARLadresse:formatlänge ASCII, rechtsbündig, vier Zeichen (4 Byte = 1 Langwort)
 Ein Wert an der angegebenen Adresse wird numerisch im ASCII-Format ausgegeben.
 Die Ausgabe erfolgt rechtsbündig formatiert in der angegebenen Formatlänge. Der Wertebereich ist 0 bis 2147483647!

Beispiel :

ARL100:10 Der 32-Bit Wert von Adresse 100 bis 103 wird numerisch gesendet.
 Der gesendete Dateninhalt ist zehn Zeichen lang.

Zeichen	1	2	3	4	5	6	7	8	9	10
Wert 0										0
Wert 2147483647	2	1	4	7	4	8	3	6	7	7

AZLadresse:formatlänge ASCII, mit Vornullen, vier Zeichen (4 Byte = 1 Langwort)
 Ein Wert an der angegebenen Adresse wird numerisch im ASCII-Format ausgegeben.
 Die Ausgabe erfolgt rechtsbündig mit Vornullen formatiert in der angegebenen Formatlänge.
 Der Wertebereich ist 0 bis 2147483647!

Beispiel :

AZL100:10 Der 32-Bit Wert von Adresse 100 bis 103 wird numerisch gesendet.
 Der gesendete Dateninhalt ist zehn Zeichen lang.

Zeichen	1	2	3	4	5	6	7	8	9	10
Wert 0	0	0	0	0	0	0	0	0	0	0
Wert 2147483647	2	1	4	7	4	8	3	6	7	7

BSadresse:länge ASCII, Bitsatz, der Inhalt wird als Folge von '0' und '1' definiert
 Ein Wert an der angegebenen Adresse bis zu Adresse+Länge-1 wird als Bitsatz im ASCII-Format ausgegeben. Der Wertebereich ist 0 (=‘00000000’) bis 255 (=‘11111111’).

Beispiel :

BS100:2 Der Wert von Adresse 100 bis 101 wird als Bitsatz gesendet.
 Der gesendete Dateninhalt ist sechzehn Zeichen lang.

Zeichen	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Wert 0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Wert 255	1	1	1	1	1	1	7	1	1	1	1	1	1	1	1	1

VZBadresse:formatlänge ASCII, mit Vornullen, ein Zeichen (1 Byte) mit Vorzeichen
 Ein Wert an der angegebenen Adresse wird numerisch mit Vorzeichen im ASCII-Format ausgegeben. Das Vorzeichen ist im höchstwertigen Bit (10000000 = -1).
 Die Ausgabe erfolgt rechtsbündig mit Vornullen formatiert in der angegebenen Formatlänge.
 Der Wertebereich ist -128 bis 127.

Beispiel :

VZB100:5 Der 8-Bit Wert von Adresse 100 wird numerisch gesendet.
 Der gesendete Dateninhalt ist fünf Zeichen lang.

Zeichen	1	2	3	4	5
Wert -128	-	0	1	2	8
Wert 0	+	0	0	0	0
Wert 127	+	0	1	2	7

VZWadresse:formatlänge ASCII, mit Vornullen, zwei Zeichen (1 Wort) mit Vorzeichen
 Ein Wert an der angegebenen Adresse wird numerisch mit Vorzeichen im ASCII-Format ausgegeben. Das Vorzeichen ist im höchstwertigen Bit (1000000000000000 = -1).
 Die Ausgabe erfolgt rechtsbündig mit Vornullen formatiert in der angegebenen Formatlänge.
 Der Wertebereich ist -32768 bis 32767.

Beispiel :

VZW100:6 Der 16-Bit Wert von Adresse 100 und 101 wird numerisch gesendet.
 Der gesendete Dateninhalt ist sechs Zeichen lang.

Zeichen	1	2	3	4	5	6
Wert -32768	-	3	2	7	6	8
Wert 0	+	0	0	0	0	0
Wert 32767	+	3	2	7	6	7

VZLadresse:formatlänge ASCII, mit Vornullen, 4 Zeichen (1 Langwort) mit Vorzeichen
 Ein Wert an der angegebenen Adresse wird numerisch mit Vorzeichen im ASCII-Format ausgegeben. Das Vorzeichen ist im höchstwertigen Bit (10000000...0 = -1).
 Die Ausgabe erfolgt rechtsbündig mit Vornullen formatiert in der angegebenen Formatlänge.
 Der Wertebereich ist -2147483648 bis 2147483647.

Beispiel :

VZL100:11 Der 32-Bit Wert von Adresse 100 bis 103 wird numerisch gesendet.
 Der gesendete Dateninhalt ist 11 Zeichen lang.

Zeichen	1	2	3	4	5	6	7	8	9	10	11
Wert -2147483648	-	2	1	4	7	4	8	3	6	4	8
Wert 0	+	0	0	0	0	0	0	0	0	0	0
Wert 2147483647	+	2	1	4	7	4	8	3	6	4	7

2.1.6 Versenden von Daten ohne Protollrahmen

Sind Telegrammdateien ohne Protokollrahmen wie etwa Startzeichen, Endezeichen oder Prüfsummen zu versenden, kann dies durch ein 'TR' (transmitt raw) anstatt 'T' erfolgen.

2.2 'C' - CYCLED : Zyklische Sendedefinition

Zyklische Sendungen werden ab dem Hochfahren der Visualisierung zyklisch ausgelöst. Die Sendung erfolgt unmittelbar oder nach einer vorgegebenen Start-Verzögerung. Werden Daten von der Gegenstelle nach der Sendung erwartet, so können diese als Antwort definiert werden.

1	2	3	4	5 - ...
Kennung 'C'	Antwort	Zykluszeit	Verzögerung	Definitionen

2.2.1 Beispiele :

C,0,2000,0,'T','e','s','t'

Alle 2 Sekunden wird der Text 'Test' ohne Erwartung eines Antworttelegrammes gesendet.

C,10,5000,2000,'T','e','s','t'

2 Sekunden nach dem Hochfahren der Visualisierung wird alle 5 Sekunden der Text 'Test' in Erwartung des Antworttelegrammes 10 gesendet.

2.2.2 Feld Antwort

In diesem Feld ist zu spezifizieren, ob eine bestimmte Antwort der Gegenstelle erwartet wird. Die Auswahl erfolgt durch Eintragen der Definitionsnummer (>0, siehe 2.3 *Antwortdefinition*).

2.2.3 Feld Zykluszeit

Der Zyklus der Sendungen ist hier anzugeben. Die Auflösung beträgt etwa 20 ms.

2.2.4 Feld Verzögerung

Soll die Sendung nicht unmittelbar nach dem Hochfahren erfolgen, so kann hier eine einmalige Verzögerungszeit in ms angegeben werden. Die Auflösung beträgt etwa 20 ms.

2.2.5 Definitionen

Die Definitionen sind wie unter 'T'-Transmit (2.1.5) anzugeben.

2.2.6 Versenden von Daten ohne Protollrahmen

Sind Telegrammdateien ohne Protokollrahmen wie etwa Startzeichen, Endezeichen oder Prüfsummen zu versenden, kann dies durch ein 'CR' (cycled raw) anstatt 'C' erfolgen.

2.3 'A' - ANSWER : Antwortdefinition

Über Antwortdefinitionen können Antworttelegramm auf gesendete Daten definiert werden. Über eine eindeutige Definitionsnummer erfolgt die Zuordnung der Antwort zu den Sendungen (Feld *Antwort* bei Transmit und Cycled). Bei Empfang der Antwort kann eine definierte Speicherstelle mit einem Wert belegt und der Dateninhalt über die Definitionen dem Treiber-Speicherbereich zugewiesen werden.

1	2	3	4 - ...
Kennung 'A'	Antwort	Adresse	Definitionen

2.3.1 Beispiele :

A,10,100:255,'O','K'

Die Antwort hat die Definitionsnummer 10 und ist gültig, wenn von der Gegenstelle der Text 'OK' empfangen wird. Es wird bei Empfang die Speicherstelle 100 mit dem Wert 255 gesetzt.

A,500,,X,'F','E','H','L','E','R',' ','AZB100:2

Die Antwort hat die Definitionsnummer 500 und ist gültig, wenn von der Gegenstelle der Text '_FEHLER nn' empfangen wird. Das Zeichen X kennzeichnet ein Zeichen, das nicht beachtet wird. Der zweistellige Fehler nn wird als Byte an der Adresse 100 gespeichert. Es wird bei Empfang keine Speicherstelle gesetzt.

2.3.2 Feld *Antwort*

In diesem Feld ist die eindeutige Definitionsnummer (>0, siehe 2.1 und 2.2 Feld *Antwort*) anzugeben.

2.3.2 Feld *Adresse*

In diesem Feld ist eine Adresse und ein Wert anzugeben, der bei erfolgreichem Empfang geschrieben wird. Soll dies nicht erfolgen, so muß der Eintrag leer (,,,,,) bleiben. Anstatt der Angabe eines Wertes kann die Zieladresse auch inkrementiert werden (n+1), dies geschieht durch Eintrag des Zeichens '!'. Alternativ kann eine Anweisungsliste eingegeben werden, die bei Empfang der Definition unmittelbar ausgeführt wird. Besteht für diese Adresse eine Sendedefinition, so wird diese unmittelbar ausgeführt. Die Sendedefinition wird durch ein vorangestelltes Ausrufezeichen ignoriert (s.u.).

Die Syntax lautet :

Adresse:Wert

oder

Adresse:!

oder

@anweisungsliste

Beispiel :

100:255 Adresse 100, Wert 255

!100:255 Adresse 100, Wert 255, vorhandene Sendedefinition nicht ausführen

100:!
Der Wert von Adresse 100 wird um Eins erhöht

@test Die Anweisungsliste 'test.awl' wird ausgeführt.

Alternativ kann eine Adresse in einem Fremdtreiber beschrieben werden. Dies geschieht durch Angabe des Treibernamens vor der Adressangabe :

Treibername/Adresse:Wert
oder
Treibername/Adresse:l

Beispiel:

HARDWARE/100:255 Im Treiber Hardware wird bei Empfang die Adresse 100 inkrementiert.

2.3.3 Definitionen

Der zu empfangende Dateninhalt wird in den Definitionen angegeben. Die Anzahl der Definitionen einer Antwort ist auf 128 begrenzt. Neben konstanten Werten können Speicherbereiche des Treibers als Variablen in unterschiedlichen Formaten eingelesen werden. Zeichen, deren Inhalt nicht relevant oder variabel ist, können von der Erkennung ausgeschlossen werden. Die Definition ist nur gültig, wenn alle Angaben erfolgreich erkannt wurden. Die Definitionen können beliebig kombiniert werden.

2.3.3.1 Zeichen ohne Relevanz

Empfangene Zeichen, die unbeachtet bleiben sollen, sind durch ein **X** zu kennzeichnen. Die Erkennung ignoriert daraufhin diese Position. Ebenfalls ist die Angabe X:nn gültig, die eine Anzahl nn von Zeichen ignoriert.

Beispiel :

A,500,,X,X,X,X,'T','e','s','t'

Die ersten vier Zeichen der Daten werden ignoriert, der Empfang ist erfolgreich, wenn daraufhin der Text 'Test' folgt (Beispiel '1234Test' oder 'WXYZTest' sind gültig).

A,500,,X:4,'T','e','s','t' entspricht dem o.a. Beispiel.

2.3.3.2 Konstanten

Konstanten sind 1 Zeichen lang und können somit einen Wertebereich von 0 bis 255 einnehmen. Der Empfang gilt nur als erfolgreich, wenn an der definierten Stelle die Konstante vorhanden ist. Zwischen folgenden Formaten kann gewählt werden :

nn numerischer Wert
\$hh numerischer Wert, hexadezimaler Format
'c' ASCII-Zeichen

Beispiel :

A,500,, 'A',100,\$0F,

Zeichenketten können auch zusammengefaßt werden :

'the quick brown fox jumps over the lazy dog' ist ebenfalls möglich. Das Hochkomma kann innerhalb des Textes nicht verwendet werden und ist durch Einfügen des Wertes 96 oder \$60 zu ersetzen.

2.3.3.3 Speicherzuweisungen

Die empfangenen Daten können in verschiedenen Formaten dem Speicherbereich des Treibers zugewiesen werden. Die Syntax ist wie folgt :

typ[adresse]:[länge]

Folgende Typen stehen zur Auswahl :

BNadresse:länge - binär

Die Daten werden ab der angegebenen Adresse bis zur Adresse+Länge-1 binär eingetragen.

Beispiele :

- BN100:1 Der Wert wird auf die Adresse 100 geschrieben.
 Der empfangene Dateninhalt ist ein Zeichen lang.
- BN100:64 Die Daten werden von Adresse 100 bis 163 eingetragen.
 Der empfangene Dateninhalt ist 64 Zeichen lang.

HXadresse:länge - hexadezimal

Die Daten werden als ASCII-Zeichen im hexadezimalen Format interpretiert. Jede Speicherstelle wird durch zwei Zeichen (00 bis FF) repräsentiert.

Beispiele :

- HX100:1 Der hexadezimale Wert wird auf die Adresse 100 geschrieben.
 Der empfangene Dateninhalt ist zwei Zeichen lang.

ALBadresse:formatlänge ASCII, linksbündig, ein Zeichen (1 Byte)

Ein Wert an der Position wird als numerisches ASCII-Format interpretiert und auf die angegebene Adresse geschrieben,
Die Daten sind linksbündig formatiert in der angegebenen Formatlänge. Der Wertebereich ist 0 bis 255.

Beispiel :

- ALB100:5 Die 5 Stellen langen Zeichen werden als 8-Bit Wert auf Adresse 100
 geschrieben
 Der empfangene Dateninhalt ist fünf Zeichen lang.

ARBadresse:formatlänge ASCII, rechtsbündig, ein Zeichen (1 Byte)

Ein Wert an der Position wird als numerisches ASCII-Format interpretiert und auf die angegebene Adresse geschrieben,
Die Daten sind rechtsbündig formatiert in der angegebenen Formatlänge. Der Wertebereich ist 0 bis 255.

Beispiel :

- ARB100:5 Die 5 Stellen langen Zeichen werden als 8-Bit Wert auf Adresse 100
 geschrieben
 Der empfangene Dateninhalt ist fünf Zeichen lang.

AZBadresse:formatlänge ASCII, mit Vornullen, ein Zeichen (1 Byte)

Ein Wert an der Position wird als numerisches ASCII-Format interpretiert und auf die angegebene Adresse geschrieben,

Die Daten sind rechtsbündig mit Vornullen formatiert in der angegebenen Formatlänge. Der Wertebereich ist 0 bis 255.

Beispiel :

AZB100:5 Die 5 Stellen langen Zeichen werden als 8-Bit Wert auf Adresse 100 geschrieben
Der empfangene Dateninhalt ist fünf Zeichen lang.

ALWadresse:formatlänge ASCII, linksbündig, zwei Zeichen (2 Byte = 1 Wort)

Ein Wert an der Position wird als numerisches ASCII-Format interpretiert und auf die angegebene Adresse und Adresse +1 geschrieben,

Die Daten sind linksbündig formatiert in der angegebenen Formatlänge. Der Wertebereich ist 0 bis 65525.

Beispiel :

ALW100:5 Die 5 Stellen langen Zeichen werden als 16-Bit Wert auf Adresse 100 und 101 geschrieben
Der empfangene Dateninhalt ist fünf Zeichen lang.

ARWadresse:formatlänge ASCII, rechtsbündig, zwei Zeichen (2 Byte = 1 Wort)

Ein Wert an der Position wird als numerisches ASCII-Format interpretiert und auf die angegebene Adresse und Adresse +1 geschrieben,

Die Daten sind rechtsbündig formatiert in der angegebenen Formatlänge. Der Wertebereich ist 0 bis 65525.

Beispiel :

ARW100:5 Die 5 Stellen langen Zeichen werden als 16-Bit Wert auf Adresse 100 und 101 geschrieben
Der empfangene Dateninhalt ist fünf Zeichen lang.

AZWadresse:formatlänge ASCII, mit Vornullen, zwei Zeichen (2 Byte = 1 Wort)

Ein Wert an der Position wird als numerisches ASCII-Format interpretiert und auf die angegebene Adresse und Adresse +1 geschrieben,

Die Daten sind rechtsbündig mit Vornullen formatiert in der angegebenen Formatlänge. Der Wertebereich ist 0 bis 65525.

Beispiel :

AZW100:5 Die 5 Stellen langen Zeichen werden als 16-Bit Wert auf Adresse 100 und 101 geschrieben
Der empfangene Dateninhalt ist fünf Zeichen lang.

ALLadresse:formatlänge ASCII, linksbündig, vier Zeichen (4 Byte = 1 Langwort)

Ein Wert an der Position wird als numerisches ASCII-Format interpretiert und auf die angegebene Adresse bis Adresse +3 geschrieben,

Die Daten sind linksbündig formatiert in der angegebenen Formatlänge. Der Wertebereich ist 0 bis 2147483647.

Beispiel :

ALL100:10 Die 10 Stellen langen Zeichen werden als 32-Bit Wert auf Adresse 100, 101, 102 und 103 geschrieben
Der empfangene Dateninhalt ist 10 Zeichen lang.

ARLadresse:formatlänge ASCII, rechtsbündig, vier Zeichen (4 Byte = 1 Langwort)

Ein Wert an der Position wird als numerisches ASCII-Format interpretiert und auf die angegebene Adresse bis Adresse +3 geschrieben,

Die Daten sind rechtsbündig formatiert in der angegebenen Formatlänge. Der Wertebereich ist 0 bis 2147483647.

Beispiel :

ARL100:10 Die 10 Stellen langen Zeichen werden als 32-Bit Wert auf Adresse 100, 101, 102 und 103 geschrieben
Der empfangene Dateninhalt ist 10 Zeichen lang.

AZLadresse:formatlänge ASCII, mit Vornullen, vier Zeichen (4 Byte = 1 Langwort)

Ein Wert an der Position wird als numerisches ASCII-Format interpretiert und auf die angegebene Adresse bis Adresse +3 geschrieben,

Die Daten sind rechtsbündig mit Vornullen formatiert in der angegebenen Formatlänge. Der Wertebereich ist 0 bis 2147483647.

Beispiel :

AZL100:10 Die 10 Stellen langen Zeichen werden als 32-Bit Wert auf Adresse 100, 101, 102 und 103 geschrieben
Der empfangene Dateninhalt ist 10 Zeichen lang.

BSadresse:länge ASCII, Bitsatz, der Inhalt wird als Folge von '0' und '1' interpretiert

Ein Wert an der Position wird als Bitsatz im ASCII-Format interpretiert und auf die angegebene Adresse geschrieben. Der Wertebereich ist 0 (= '00000000') bis 255 (= '11111111').

Beispiel :

BS100:2 Die 16 Stellen lange Zeichenkette wird als Bitsatz erkannt und auf Adresse 100 bis 101 geschrieben.
Der empfangene Dateninhalt ist sechzehn Zeichen lang.

VZBadresse:formatlänge ASCII, mit Vornullen, ein Zeichen (1 Byte) mit Vorzeichen
Ein Wert an der Position wird numerisches, vorzeichenbehaftetes ASCII-Format interpretiert und auf die angegebene Adresse geschrieben,
Die Daten sind rechtsbündig mit Vornullen formatiert in der angegebenen Formatlänge. Der Wertebereich ist -128 bis +127.

Beispiel :

VZB100:10 Die 10 Stellen langen Zeichen werden als 8-Bit Wert auf Adresse 100 geschrieben
Der empfangene Dateninhalt ist 10 Zeichen lang.

VZWadresse:formatlänge ASCII, mit Vornullen, zwei Zeichen (1 Wort) mit Vorzeichen
Ein Wert an der Position wird numerisches, vorzeichenbehaftetes ASCII-Format interpretiert und auf die angegebene Adresse und Adresse +1 geschrieben,
Die Daten sind rechtsbündig mit Vornullen formatiert in der angegebenen Formatlänge. Der Wertebereich ist -32768 bis 32767.

Beispiel :

VZW100:10 Die 10 Stellen langen Zeichen werden als 16-Bit Wert auf Adresse 100 und Adresse 101 geschrieben
Der empfangene Dateninhalt ist 10 Zeichen lang.

VZLadresse:formatlänge ASCII, mit Vornullen, 4 Zeichen (1 Langwort) mit Vorzeichen
Ein Wert an der Position wird numerisches, vorzeichenbehaftetes ASCII-Format interpretiert und auf die angegebene Adresse bis Adresse +3 geschrieben,
Die Daten sind rechtsbündig mit Vornullen formatiert in der angegebenen Formatlänge. Der Wertebereich ist -2147483648 bis 2147483647.

Beispiel :

VZL100:11 Die 11 Stellen langen Zeichen werden als 32-Bit Wert auf Adresse 100 bis 103 geschrieben
Der empfangene Dateninhalt ist 11 Zeichen lang.

2.4 'R' - RECEIVE : Empfangsdefinition

Unaufgefordert gesendete Daten der Gegenstelle werden über die Kennung 'R' definiert. Bei Empfang der Daten kann eine definierte Speicherstelle mit einem Wert belegt und der Dateninhalt über die Definitionen dem Treiber-Speicherbereich zugewiesen werden.

1	2	3 - ...
Kennung 'R'	Adresse	Definitionen

R,100:255,'O','K'

Die Daten sind gültig, wenn von der Gegenstelle der Text 'OK' empfangen wird. Es wird bei Empfang die Speicherstelle 100 mit dem Wert 255 gesetzt.

R,,X,'F','E','H','L','E','R',' ','AZB100:2

Die Daten sind gültig, wenn von der Gegenstelle der Text '_FEHLER nn' empfangen wird. Das Zeichen X kennzeichnet ein Zeichen, das nicht beachtet wird. Der zweistellige Fehler nn wird als Byte an der Adresse 100 gespeichert. Es wird bei Empfang keine Speicherstelle gesetzt.

2.4.1 Feld Adresse

In diesem Feld ist eine Adresse und ein Wert anzugeben, der bei erfolgreichem Empfang geschrieben wird. Soll dies nicht erfolgen, so muß der Eintrag leer (,,,,,) bleiben. Anstatt der Angabe eines Wertes kann die Zieladresse auch inkrementiert werden (n+1), dies geschieht durch Eintrag des Zeichens '!'. Alternativ kann eine Anweisungsliste eingegeben werden, die bei Empfang der Definition unmittelbar ausgeführt wird. Besteht für diese Adresse eine Sendedefinition, so wird diese unmittelbar ausgeführt. Die Sendedefinition wird durch ein vorangestelltes Ausrufezeichen ignoriert (s.u.).

Adresse:Wert

oder

Adresse:l

oder

@anweisungsliste

Beispiel :

100:255 Adresse 100, Wert 255

!100:255 Adresse 100, Wert 255, vorhandene Sendedefinition nicht ausführen

100:l Der Wert von Adresse 100 wird um Eins erhöht

@test Die Anweisungsliste 'test.awl' wird ausgeführt.

Alternativ kann eine Adresse in einem Fremdtreiber beschrieben werden. Dies geschieht durch Angabe des Treibernamens vor der Adressangabe :

Treibername/Adresse:Wert

oder

Treibername/Adresse:l

Beispiel:

HARDWARE/100:255 Im Treiber Hardware wird bei Empfang die Adresse 100 inkrementiert.

2.4.2 Definitionen

Der zu empfangende Dateninhalt wird in den Definitionen angegeben. Die Anzahl der Definitionen eines Empfangs ist auf 128 begrenzt. Neben konstanten Werten können Speicherbereiche des Treibers als Variablen in unterschiedlichen Formaten eingelesen werden. Zeichen, deren Inhalt nicht relevant oder variabel ist, können von der Erkennung ausgeschlossen werden. Die Definition ist nur gültig, wenn alle Angaben erfolgreich erkannt wurden. Die Definitionen können beliebig kombiniert werden.

2.4.2.1 Zeichen ohne Relevanz

Zeichen ohne Relevanz sind wie unter 2.3.3.1 anzulegen.

2.4.2.2 Konstanten

Die Konstanten sind wie unter 2.3.3.2 anzulegen.

2.4.2.3 Speicherzuweisungen

Die Speicherzuweisungen sind wie unter 2.3.3.3 anzulegen.

2.5 'W' Zyklische Sendedefinition mit Bedingung

Zyklische Sendungen werden ab dem Hochfahren der Visualisierung zyklisch ausgelöst, sofern die Bedingung erfüllt ist. Die Sendung erfolgt unmittelbar oder nach einer vorgegebenen Start-Verzögerung. Werden Daten von der Gegenstelle nach der Sendung erwartet, so können diese als Antwort definiert werden.

1	2	3	4	5	6 - ...
Kennung 'W'	Antwort	Zykluszeit	Verzögerung	Bedingung	Definitionen

2.5.1 Beispiel :

$W,0,2000,0,100=1,'T','e','s','t'$

Alle 2 Sekunden wird der Text 'Test' ohne Erwartung eines Antworttelegrammes gesendet, sofern an der Adresse 100 der Wert 1 steht.

2.5.2 Feld Antwort

In diesem Feld ist zu spezifizieren, ob eine bestimmte Antwort der Gegenstelle erwartet wird. Die Auswahl erfolgt durch Eintragen der Definitionsnummer (>0, siehe 2.3 Antwortdefinition).

2.5.3 Feld Zykluszeit

Der Zyklus der Sendungen ist hier anzugeben. Die Auflösung beträgt etwa 20 ms.

2.5.4 Feld Verzögerung

Soll die Sendung nicht unmittelbar nach dem Hochfahren erfolgen, so kann hier eine einmalige Verzögerungszeit in ms angegeben werden. Die Auflösung beträgt etwa 20 ms.

2.5.5 Bedingung

Die Bedingung bildet sich aus der Adresse und einem Wert, getrennt durch das Gleichheitszeichen '='. Ohne Bedingung verhält sich die Definition wie unter 'C'.

Beispiel

$100=0$ Der Inhalt der Adresse 100 muß 0 sein.

2.5.6 Definitionen

Die Definitionen sind wie unter 'T'-Transmit (2.1.5) anzugeben.

2.6 'X' - EXTENDED : Treiberspezifische Informationen und Verbindungstest

Treiberspezifische Daten sowie die Abfrage einer Verbindung zur Gegenstelle können an einer bestimmten Treiberadresse abgelegt werden. Diese Definition ist einmalig zu tätigen, ist sie nicht vorhanden, werden keine Informationen zur Verfügung gestellt.

Die Abfrage der Verbindung erfolgt über das unterste Bit an der definierten Adresse. Ist es auf Eins gesetzt, so besteht eine Verbindung zur Gegenstelle. Der Ausfall der Verbindung kann nur bei Treiber-Dialogen geprüft werden, d.h. wenn nach dem Versenden eine erwartete Antwort oder eine treiberspezifische Bestätigung (ACK o.ä) ausbleibt. Nach wiederholtem Ausbleiben einer Antwort wird das Bit auf Null gesetzt.

1	2
Kennung 'X'	Adresse

2.6.1 Beispiel :

X,300

Die treiberspezifischen Informationen werden ab Adresse 300 abgelegt.

2.7 'xR' - RAW : Senden von Rohdaten

Die Sendebefehle 'T', 'C' und 'W' können durch ein zusätzliches Zeichen 'R' ('TR', 'CR' und 'WR') dazu veranlasst werden, die definierten Telegramm Daten im Rohformat ohne den treiberspezifischen Protokollrahmen zu versenden. Diese Funktion ist insbesondere bei Treibern notwendig, die einen unterschiedlichen Send- und Empfangsrahmen haben.

2.7.1 Beispiel :

TR,0,0,100:1,'T','e','s','t',13

Bei Änderung des untersten Bits (00000001) auf Adresse 100 wird, ohne Erwartung eines Antworttelegrammes der Text 'Test' mit einem Wagenrücklauf (CR-Carriage Return) gesendet.

2.8 'Yn' - AUTOCONFIG : Sonderbefehle der Schnittstellenassistenten

Der Sonderbefehl 'Yn' ('Y0'...) wird für die Schnittstellenassistenten benötigt. Veränderungen in diesen Definitionen sollten nicht manuell vorgenommen werden.

3. Konfiguration der Schnittstelle

Die Konfiguration erfolgt über die Datei *STARTUP.INI* im PVis-Arbeitsverzeichnis. In den jeweiligen Treiberdefinitionen sind alle Einstellungen vorzunehmen, bzw. werden bei der Installation des Treibers vorbesetzt :

Beispiel (für serielle Schnittstellen):

[Treibername]

Port=COM2

StatusPortLow=0 **wird nur in der DOS-Plattform ausgewertet!**

StatusPortHigh=0 **wird nur in der DOS-Plattform ausgewertet!**

Address=0 **wird nur in der DOS-Plattform ausgewertet!**

IRQ=INT3 **wird nur in der DOS-Plattform ausgewertet!**

Parity=NONE

Handshake=NONE

Baudrate=9600

DataBits=8

StopBits=1

FIFO=8 **wird nur in der DOS-Plattform ausgewertet!**

3.1 Eintrag Port

Die zu verwendene serielle Schnittstelle ist anzugeben. Die NT-Plattform unterstützt alle im System angemeldeten Schnittstellen.

Mögliche Einträge sind :

COM1 bis COM32 bei DOS, COM1 bis COM255 bei NT

Die Einträge haben dabei folgende Bedeutung (nur DOS-Plattform):

COM1	Standard COM-1, Adresse im BIOS (3F8h)	
COM2	Standard COM-2, Adresse im BIOS (2F8h)	
COM3	Standard COM-3, Adresse im BIOS (3E8h)	
COM4	Standard COM-4, Adresse im BIOS (2E8h)	
COM5	AST/Multi-Seriell/Interrupt-Karte 1, Port 1	StatusPortLow angeben! s.u.
COM6	AST/Multi-Seriell/Interrupt-Karte 1, Port 2	
COM7	AST/Multi-Seriell/Interrupt-Karte 1, Port 3	
COM8	AST/Multi-Seriell/Interrupt-Karte 1, Port 4	
COM9	AST/Multi-Seriell/Interrupt-Karte 2, Port 1	StatusPortHigh angeben! s.u.
COM10	AST/Multi-Seriell/Interrupt-Karte 2, Port 2	
COM11	AST/Multi-Seriell/Interrupt-Karte 2, Port 3	
COM12	AST/Multi-Seriell/Interrupt-Karte 2, Port 4	
COM13	zusätzliche Schnittstelle mit Adressangabe, Port 1	
COM14	zusätzliche Schnittstelle mit Adressangabe, Port 2	
COM15	zusätzliche Schnittstelle mit Adressangabe, Port 3	
COM16	zusätzliche Schnittstelle mit Adressangabe, Port 4	
COM17	ARNET-kompatibel Multiseriell, Port 1	StatusPortLow angeben! s.u.
COM18	ARNET-kompatibel Multiseriell, Port 2	
COM19	ARNET-kompatibel Multiseriell, Port 3	
COM20	ARNET-kompatibel Multiseriell, Port 4	
COM21	ARNET-kompatibel Multiseriell, Port 5	
COM22	ARNET-kompatibel Multiseriell, Port 6	
COM23	ARNET-kompatibel Multiseriell, Port 7	
COM24	ARNET-kompatibel Multiseriell, Port 8	

Achtung Folgende Kombinationen sind möglich :

COM1 bis COM4 und AST Karte 1 (COM 5-8) und COM13 bis COM16
oder
COM1 bis COM4 und AST Karte 1 (COM 5-8) & 2 (COM 9-12) oder
ARNET (COM 17-24)

Es darf kein Port mehrfach verwendet werden.

3.2 Eintrag StatusPortLow und StatusPortHigh (DOS-Plattform)

Hier sind die Adressen der Statusports der multiseriellen Schnittstellen einzutragen :

AST-kompatible Schnittstellenkarten

StatusPortLow Statusport-Adresse der 1. Karte (4 Ports)
StatusPortHigh Statusport-Adresse der 2. Karte (4 Ports)

ARNET-kompatible Schnittstellenkarten

StatusPortLow Statusport-Adresse der Karte (8 Ports)

Die Statusports sind der Dokumentation der Schnittstellenkarte zu entnehmen.

Beispiel für eine AST-kompatible / multiserielle Karte :

```
[Treibername1]
Port=COM5
StatusPortLow=$240
StatusPortHigh=0
Address=$200
IRQ=INT7
```

...

```
[Treibername2]
Port=COM6
StatusPortLow=$240
StatusPortHigh=0
Address=$208
IRQ=INT7
```

...

```
[Treibername3]
Port=COM7
StatusPortLow=$240
StatusPortHigh=0
Address=$210
IRQ=INT7
```

...

3.3 Eintrag Address (DOS-Plattform)

Hier ist die Adresse der seriellen Schnittstelle anzugeben, sofern unter **Port** nicht der Eintrag COM1 bis COM4 steht. Der Wert kann dezimal (nnnn) oder hexadezimal (\$hhh) angegeben werden. Die Adresse der Schnittstellen COM1 bis COM4 wird aus dem BIOS des Rechners gelesen.

3.4 Eintrag IRQ (DOS-Plattform)

Dieser Eintrag bestimmt den Interrupt der Schnittstelle.

Mögliche Einträge sind :

INT3
INT4
INT5
INT6
INT7
INT9
INT10
INT11
INT12
INT13
INT14
INT15

Die Interrupts pro Schnittstelle bzw. pro multiserieller Karte müssen unterschiedlich sein. Beachten Sie die Belegung der Interrupts durch andere Karten im PC.

3.5 Eintrag Parity

Die Erzeugung des Paritätsbits der seriellen Datenübertragung wird hiermit festgelegt.

Mögliche Einträge sind :

NONE	keine Parität
EVEN	gerade Parität
ODD	ungerade Parität

3.6 Eintrag Handshake

Sofern die Übertragung mit einem Handshake erfolgt, so ist dieser Eintrag anzuwenden.

Mögliche Einträge sind :

NONE	kein Handshake
XOFF/XON	Handshake über XOFF bzw. XON - Zeichen
CTS/RTS	Handshake über Steuersignale CTS bzw. RTS

3.7 Eintrag Baudrate

Die Geschwindigkeit der Datenübertragung erfolgt über diesen Eintrag. Es ist ein numerischer Wert zwischen 300 und 115000 anzugeben. Standard ist 9600 (=9600 Baud).

3.8 Eintrag DataBits

Die Anzahl der Datenbits kann zwischen 7 und 8 über dieses Feld gewählt werden (Eintrag 7 oder 8).

3.9 Eintrag StopBits

Die Anzahl der Stopbits kann zwischen 1 und 2 über dieses Feld gewählt werden (Eintrag 1 oder 2).

3.10 Eintrag FIFO (DOS-Plattform)

Über diesen Eintrag wird der Empfangspuffer der heutzutage üblichen Schnittstellenbausteine bestimmt. Der Eintrag ist vom Datenaufkommen abhängig und gegebenenfalls auszuprobieren.

Mögliche Einträge sind :

- 0** kein Empfangspuffer
- 4** Empfangspuffer 4 Bytes
- 8** Empfangspuffer 8 Bytes
- 14** Empfangspuffer 14 Bytes

4. Sonstige Konfigurationen

Neben der Schnittstellenkonfiguration können in der *STARTUP.INI* noch weitere Einstellungen vorgenommen werden, die sich wie folgt darstellen :

Beispiel:

[Treibername]

LoadFrom=DATEN.DTA

SaveTo=DATEN.DTA

MemorySize=1024

4.1 Eintrag LoadFrom

Dieser Eintrag gibt eine optionale Datei an, aus der der Speicherinhalt des Treibers vor dem Hochfahren geladen wird. Er steht im Zusammenhang mit dem Eintrag SaveTo (s. 4.2). Der Eintrag kann eine beliebige Datei mit Pfadangabe spezifizieren.

4.2 Eintrag SaveTo

Dieser Eintrag gibt eine optionale Datei an, in der der Speicherinhalt des Treibers vor dem Herunterfahren geladen wird. Er steht im Zusammenhang mit dem Eintrag LoadFrom (s. 4.1). Der Eintrag kann eine beliebige Datei mit Pfadangabe spezifizieren.

4.3 Eintrag MemorySize

Dieser optionale Eintrag ermöglicht es, den Speicherbereich des Treibers zu ändern. Insbesondere in Netzwerkanwendungen kann es von Vorteil sein, daß der Speicherbereich so klein wie möglich gehalten wird, um den Datenaustausch zu vermindern. Es sind Werte zwischen 16 und 65520 (in Bytes) möglich.