

Inhaltsverzeichnis zur Anleitung zum TOOL4

Erläuterungen zur synthetischen Programmierung	Seite 4-2
CLEN	Seite 4-3
CONV	Seite 4-4
CONVL\$	Seite 4-5
CONVS\$	Seite 4-6
ELAST	Seite 4-7
ELINE	Seite 4-8
ELIST\$	Seite 4-9
ENTER	Seite 4-10
Beispiele zur synthetischen Programmierung	Seite 4-11
Erläuterungen zu statistischen Berechnungen	Seite 4-14
AVGX / AVGY	Seite 4-15
CORR	Seite 4-16
GRA	Seite 4-17
SDNX / SDNY	Seite 4-18
SDVX / SDVY	Seite 4-19
SEG	Seite 4-20
STAT	Seite 4-21
STATCLR	Seite 4-22
STATIN	Seite 4-23
STATOFF	Seite 4-24
STATON	Seite 4-25
STATOUT	Seite 4-26
STX / STY	Seite 4-27
Beispiele zu statistischen Berechnungen	Seite 4-28
Liste der Instruktionen	Seite 4-30

Erläuterungen zur synthetischen Programmierung *****

Einführung

Unter "synthetischer Programmierung" ist die Erzeugung oder Veränderung eines ausführbaren Programmes durch ein anderes zu verstehen. Die Befehle des TRAMsoft TOOL4 ermöglichen das Lesen und Speichern von BASIC-Programm-Zeilen während der Programmausführung.

BASIC-Schlüsselworte

Im SHARP PC-1500/A werden aus Platzgründen alle BASIC-Schlüsselworte wie PRINT, INPUT, GOSUB usw. in sogenannte 2-Byte-Token übersetzt und als solche im Speicher abgelegt. Variablen-Namen, Texte und Sonderzeichen werden dagegen im ASCII-Code gespeichert (siehe Anhang A-3/A-4). Wird ein Programm in die Anzeige gebracht oder auf einem Drucker ausgegeben, so werden die Token wieder in die entsprechenden Zeichenfolgen zurück übersetzt.

Eine in Token übersetzte Programm-Zeile kann, wie eine BASIC-Variable, maximal 80 Bytes lang sein. Eine zurück übersetzte Programm-Zeile kann dagegen wesentlich länger werden. Die Funktion ELIST\$ liefert deshalb die unübersetzte Programm-Zeile, damit diese einer Variablen zugewiesen werden kann. Daneben stehen Funktionen zur Verfügung, um die Token in einer Programm-Zeile aufzufinden und in die zugehörige Zeichenfolge zu übersetzen.

Voraussetzungen

Die Anweisungen zur synthetischen Programmierung beziehen sich immer auf das "aktive Programm-Modul" (siehe Allgemeines, Seite iii).

Es können nur Programm-Zeilen verändert, gelöscht oder hinzugefügt werden, die sich im Speicher hinter der zur Zeit ausgeführten Programm-Zeile befinden. Diese Bedingung ist erfüllt, wenn

- sich die zu bearbeitende Programm-Zeile und die zur Zeit ausgeführte Programm-Zeile im gleichen Programm-Modul befinden und die zu bearbeitende Programm-Zeile eine höhere Zeilennummer aufweist als die ausgeführte Programm-Zeile

oder

- sich die zu bearbeitende Programm-Zeile in einem Programm-Modul befindet, das nachträglich mit MERGE, FLOAD P (TOOL 2), DLOAD P (Floppy) oder APPEND (TOOL 1) zum Programm mit der zur Zeit ausgeführten Programm Zeile hinzugefügt wurde.

CLEN

```
*****  
*  
* Liefert die Länge einer übersetzten Programm-Zeile. *  
*  
*****
```

Syntax



Ergebnis: numerischer Wert

Parameter-Beschreibung: - [Programm-Zeile]: Nicht übersetzte BASIC-Zeile.

Hinweise: - Enthält die Programm-Zeile kein BASIC-Schlüsselwort, so liefern CLEN und LEN das gleiche Resultat.

- CLEN kann manuell oder programmkontrolliert ausgeführt werden.
- Beispiel siehe Seite 4-11.

CONV

```
*****  
*  
* Liefert die Anzahl Elemente in einer Programm-Zeile. *  
*  
*****
```

Syntax



Ergebnis: numerischer Wert

Parameter-Beschreibung: - [Programm-Zeile]: Nicht übersetzte BASIC-Zeile.

Erläuterung

Ein Element ist:

- Ein einzelnes Zeichen eines Variablen-Namen oder Textes, sowie Sonderzeichen
- Ein Token eines BASIC-Schlüsselwortes

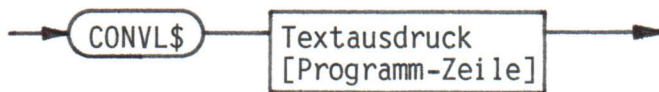
Hinweise: - Das Resultat der Funktion CONV kann als Schleifenzähler beim schrittweisen Uebersetzen einer Programm-Zeile verwendet werden.

- CONV kann manuell oder programmkontrolliert ausgeführt werden.
- Beispiel siehe Seite 4-11.

CONVL\$

```
*****  
* Uebersetzt alle Token in einer Programm-Zeile in die *  
* entsprechenden Zeichenfolgen. *  
*****
```

Syntax



Ergebnis: Text

Parameter-Beschreibung: - [Programm-Zeile]: Nicht übersetzte BASIC-Zeile.

- Hinweise:
- CONVL\$ übersetzt alle Token in die entsprechenden Zeichenfolgen, alle übrigen Zeichen bleiben unverändert.
 - Token von nicht angeschlossenen Zusatzgeräten zum PC-1500 werden wie bei LIST, bzw. LLIST mit einer "Wellenlinie" übersetzt.
 - CONVL\$ kann manuell oder programmkontrolliert ausgeführt werden.
 - Beispiel siehe Seite 4-11.

Besondere Fehlermeldung: - ERROR 67: Die übersetzte Programm-Zeile würde länger als 80 Zeichen.

CONVS\$

```
*****  
* Uebersetzt ein einzelnes Element in einer Programm-  
* Zeile.  
*  
*****
```

Syntax



Ergebnis: Text

Parameter-Beschreibung: - [Programm-Zeile]: Nicht übersetzte BASIC-Zeile.
- [Position]: Stelle des zu übersetzenden Elementes.

Hinweise: - CONVS\$ übersetzt ein einzelnes Element in einer Programm-Zeile.
- Ein Element ist ein einzelnes Zeichen oder ein Token.
- Ein einzelnes Zeichen bleibt unverändert.
- Ein Token wird in die entsprechende Zeichenfolge übersetzt.
- Token von nicht angeschlossenen Zusatzgeräten zum PC-1500 werden wie bei LIST, bzw. LLIST mit einer "Wellenlinie" übersetzt.
- CONVS\$ kann manuell oder programmkontrolliert ausgeführt werden.
- Beispiel siehe Seite 4-11.

ELAST

```
*****  
*  
* Liefert die Nummer der letzten Programm-Zeile.  
*  
*****
```

Syntax



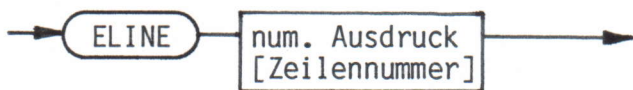
Ergebnis: numerischer Wert

- Hinweise: - ELAST bezieht sich immer auf das aktive Programm-Modul.
- ELAST kann manuell oder programmkontrolliert ausgeführt werden.
- Beispiel siehe Seite 4-11.

ELINE

```
*****  
*  
* Liefert die Nummer der angegebenen oder nächst höheren *  
* Programm-Zeile. *  
*  
*****
```

Syntax



Ergebnis: numerischer Wert

Parameter-Beschreibung: - [Zeilennummer]: num. Ausdruck im Bereich der möglichen Zeilennummern.

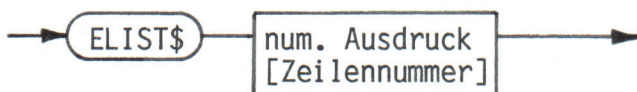
- Hinweise:
- ELINE dient dazu, die Zeilennummer der nachfolgenden Programmzeile festzustellen.
 - ELINE bezieht sich immer auf das aktive Programm-Modul.
 - ELINE kann manuell oder programmkontrolliert ausgeführt werden.
 - Beispiel siehe Seite 4-11.

Besondere Fehlermeldung: - ERROR 11: Die angegebene Zeilennummer ist größer als die höchste Zeilennummer im aktiven Programm-Modul.

ELIST\$

```
*****  
*  
* Liefert die unübersetzte Programm-Zeile. *  
*  
*****
```

Syntax



Ergebnis: Text

Parameter-Beschreibung: - [Zeilennummer]: Zeilennummer der aus dem Programm-Speicher auszulesenden BASIC-Zeile. Existiert die angegebene Zeilennummer nicht, so wird die Zeile mit der nächst höheren Zeilennummer ausgelesen.

Hinweise: - ELIST\$ bezieht sich immer auf das aktive Programm-Modul.
- ELIST\$ kann manuell oder programmkontrolliert ausgeführt werden.
- Beispiel siehe Seite 4-11.

Besondere Fehlermeldung: - ERROR 11: Die angegebene Zeilennummer ist größer als die höchste Zeilennummer im aktiven Programm-Modul.