



HICAD

Design Automation
(Makrotechnik)

THE WORLD OF CAD AND PDM SOLUTIONS

UNLIMITED PERFORMANCE



Inhalt

1	Arbeiten mit Makros	5
1.1	Allgemeine Informationen.....	5
1.2	Hinweise zum Textmenümodus	8
1.3	Makro erstellen	9
1.3.1	Makroaufbau	10
1.3.2	Makrobefehle	10
1.3.3	Das freie Argument #.....	11
1.3.4	Makroerstellung beenden	12
1.4	Makro aufrufen	12
1.5	Makro bearbeiten.....	13
1.5.1	Makro überschreiben.....	13
1.5.2	Makro übernehmen	14
1.5.3	Makro testen/korrigieren.....	15
1.5.4	Kompilieren von Makros	19
2	Allgemeines zur Makrosprache	20
2.1	Einteilung der HCGS-Befehle in Gruppen.....	20
2.2	Die Argumentgruppen	22
2.3	Variablen.....	25
2.3.1	Benutzerspezifische Variable	25
2.3.2	Punkt- und Linienvariablen	25
2.3.3	Die Systemvariablen.....	26
2.4	Arithmetische Ausdrücke.....	33
2.4.1	Rechenoperatoren	33
2.4.2	Grundfunktionen	34
2.5	Logische Vergleichsausdrücke.....	35
2.6	Logische Variable	37
2.7	Stringausdrücke.....	39
2.7.1	Numerische Variable in String umwandeln	39
2.7.2	Stringoperationen	40
2.7.3	Stringfunktionen.....	40

2.8	Der Variablenspeicher.....	42
3	Die Befehle der Makrosprache	43
3.1	Alphabetische Befehlsübersicht.....	45
3.1.1	Antwort.....	47
3.1.2	APAUS / APEIN.....	49
3.1.3	CALL.....	50
3.1.4	COPY.....	52
3.1.5	DEL.....	54
3.1.6	DISTANZ	55
3.1.7	ECHO	57
3.1.8	END	59
3.1.9	FAA / FAE.....	60
3.1.10	FOR ... NEXT	61
3.1.11	GOTO	63
3.1.12	IF..ELSE..IFEND	65
3.1.13	IGNORE.....	67
3.1.14	INTEGER.....	68
3.1.15	MAKRO.....	70
3.1.16	MAUS/MEIN	72
3.1.17	MKDIR	73
3.1.18	OPEN INPUTCLOSE OPEN OUTPUT CLOSE	74
3.1.19	OPTION	77
3.1.20	PFD.....	80
3.1.21	POINT.....	81
3.1.22	REAL	83
3.1.23	REM.....	84
3.1.24	REPEAT	86
3.1.25	SAUS / SEIN.....	87
3.1.26	START	89
3.1.27	STRING	90
3.1.28	SZAUS / SZEIN	92
3.1.29	UDA / UDE	93
3.1.30	VAI	94

3.1.31	VAR.....	95
3.1.32	Variablenzuweisung	97
3.1.33	WAIT	99
3.1.34	WARTE.....	100
3.1.35	WAUS/WEIN.....	101
3.1.36	WERT	102
3.1.37	WHILE...WHEND.....	103
3.1.38	WINKEL	105
3.1.39	ZAA / ZAE	107
3.2	Beispiel	109
4	Weitere Hinweise.....	114
4.1	DXF-Dateien.....	114
4.2	Makrovariable ZDSP	114
4.3	Texte selektieren	114
4.4	Text-Tools.....	115
4.5	True-Type-Font.....	115
4.6	Objektcursor	115
4.7	Blechabwicklung.....	115
4.8	Bemaßung	116
4.9	Skizziercursor	116
4.10	Bauteilkennung.....	116
4.11	Betriebssystemaufruf.....	116
4.12	Werkstoffschraffur	116
4.13	Export in STEP/MTA oder IGES/CATIA.....	117
4.14	Temporärer Pfad	117
4.15	Makrovariablen für UNDO.....	118
5	Fehlermeldungen	120
6	Makrobibliotheken.....	122
7	Die Datei FILEGRUP.DAT	123
	Index.....	125

1 Arbeiten mit Makros

1.1 Allgemeine Informationen

Der Schlüssel zur Automatisierung und Beschleunigung von Konstruktionsvorgängen liegt in den sehr leistungsfähigen Makro- und Variantentechniken von HiCAD.

Die HiCAD-Makrotechnik ermöglicht die Reproduktion und damit die Wiederholung häufig wiederkehrender HiCAD-Abläufe. Sie beruht auf der HiCAD-Makrosprache HCGS (HiCAD Grafische Sprache) und arbeitet nach dem Teach-in-Prinzip. Das bedeutet, dass beliebig komplexe HiCAD-Vorgänge während ihrer interaktiven Ausführung in der Makrosprache aufgezeichnet werden. Zur Korrektur und Bearbeitung dieser aufgezeichneten Makros steht ein Makroeditor zur Verfügung, mit dem Makros auch getestet werden können.

HiCAD-Makros können in das Bildschirmmenü integriert werden, so dass Sie als Anwender die HiCAD-Benutzeroberfläche um Ihre speziellen Anforderungen erweitern können.

Die Erstellung von Makros wird vom HiCAD-Makroentwicklungssystem HC-MES weitgehend automatisch vorgenommen.

Um ein Makro zu erstellen, geben Sie zunächst den Makronamen an und führen dann im Prinzip die Befehlsfolge, die im Makro ablaufen soll, einmal praktisch in HiCAD aus, indem Sie die entsprechenden Funktionen über die Menüs aufrufen. Alle aufgerufenen Funktionen sowie die zugehörigen Eingaben werden im angegebenen Makro protokolliert. Bei Funktionen, die eine Benutzereingabe anfordern, muss festgelegt werden, wie diese beim Ablauf zu interpretieren ist, ob als vordefinierte Eingabe oder als Abfrage.

Die einzelnen Funktionen werden in ihrer Reihenfolge solange in das zu erstellende Makro aufgenommen, bis Sie die Makroerstellung beenden.

Der HiCAD-Makrointerpreter kann neben den durch das **HC-MES** erstellten Befehlen eine Vielzahl weiterer Befehle interpretieren und ausführen. Diese zusätzlichen Befehle müssen Sie explizit in das Makro eintragen. Sie reichen von einfachen Befehlen, die nur eine spezifische Benutzermitteilung auf dem Bildschirm ausgeben, bis hin zu komplexen Sprachkonstruktionen, die aus höheren Programmiersprachen bekannt sind (Schleifen, logische Abfragen usw.).

Dazu können Sie das Makro nachträglich mit dem integrierten Makroeditor oder einem Texteditor bearbeiten.

Die Befehlsfolge, die im Makro ablaufen soll, führen Sie jedoch nicht wie bei der normalen HiCAD-Konstruktion in den üblichen Icon-Menüs aus, sondern im Textmenümodus. Dieser Modus wird bei der Makroerstellung und -bearbeitung automatisch aktiviert. In diesem Modus erfolgt die Funktionsauswahl über spezielle Textmenüs, in denen jedoch dieselben Funktionen zur Verfügung stehen, die Sie auch sonst gewohnt sind.

Nach dem Aufruf der Funktion **Makro erstellen** wird der Textmenümodus aktiviert und das Hauptmenü angezeigt. Von hier aus erreichen Sie alle anderen HiCAD-Funktionen.

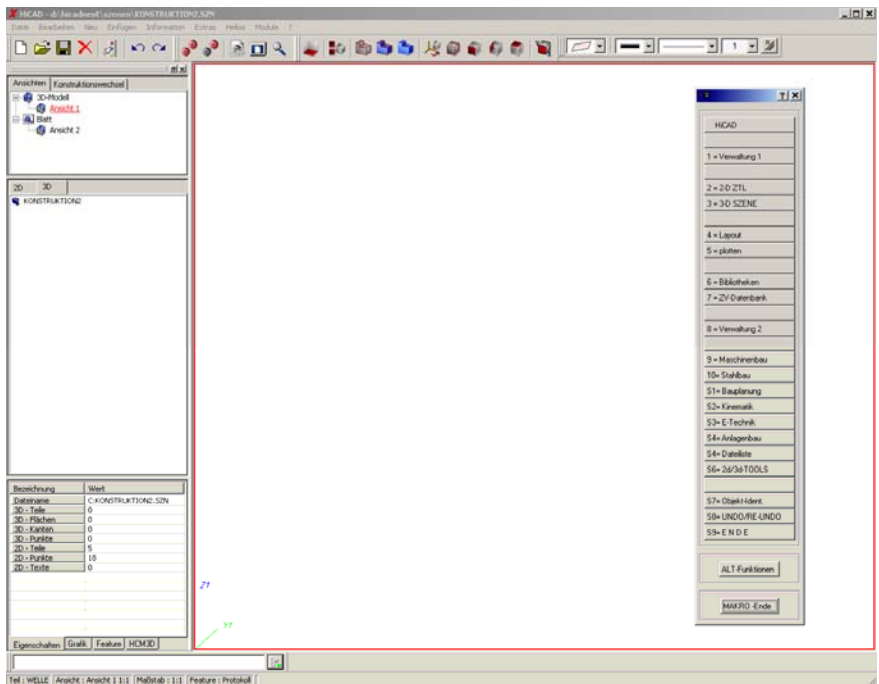


Abb. 1 Benutzeroberfläche im Textmenümodus

1.2 Hinweise zum Textmenümodus

Bitte beachten Sie bei der Makro- und Variantenerstellung grundsätzlich die folgenden Hinweise:

- Um eine neue 2D- oder 3D-Zeichnung zu erstellen, wählen Sie immer die Funktionen **3D-Szene** und dann **Neu anlegen**. Dies gilt auch dann, wenn Sie eine reine 2D-Konstruktion erstellen wollen, da HiCAD nicht zwischen 2D- und 3D-Konstruktionsdateien unterscheidet.
- Um ein neues 2D-Teil anzulegen, wählen Sie nacheinander die Funktionen **2D-ZTL**, **ZTL bearbeiten** und **Hauptteil neu** bzw. **Sub.-Teil neu** (für Nebenteile).
- Um ein neues 3D-Teil anzulegen, wählen Sie nacheinander die Funktionen **3D-Szene**, **Bearbeiten** und **Hauptteil neu** bzw. **Teil neu** (für Nebenteile).
- Sonderfunktionen, wie die Zoom-Funktionen, Funktionen für die Ansichtengenerierung, Rasterfunktionen oder das Plotten- und Drucken erreichen Sie durch Auswahl des Eintrages **ALT-Funktionen**.
- Mit **Makro-Ende** beenden Sie die Makroerstellung.
- Um zum vorhergehenden Menü zurück zu schalten, drücken Sie die mittlere oder die rechte Maustaste. Alternativ können Sie auch die ESC-Taste der Tastatur verwenden.

Eine Übersicht der wichtigsten Textmenüs finden Sie auf der Klappkarte am Ende des Buches.

1.3 Makro erstellen

Zur Erstellung eines Makros benutzen Sie die Funktion **Makro erstellen**. Sie aktivieren diese Funktion mit den Tasten

Strg + 7

HiCAD fordert nun die Eingabe des Makronamens an. Anschließend wird automatisch das Makroentwicklungssystem eingeschaltet, in den Textmenümodus gewechselt und das Haupttextmenü angezeigt.

Ist unter dem angegebenen Namen bereits ein Makro vorhanden, dann haben Sie die Möglichkeit, dieses Makro zu bearbeiten.

siehe 1.5

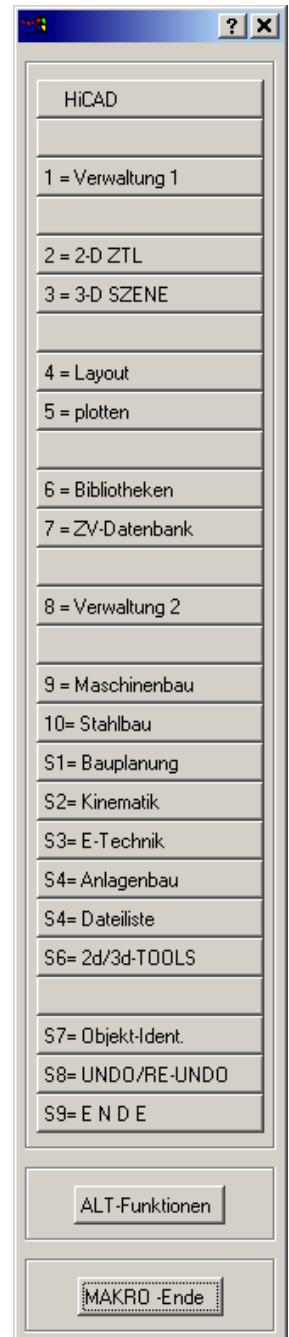


Abb. 2 Hauptmenü

1.3.1 Makroaufbau

■ Makroname

Die Dateinamenserweiterung für Makros ".MAC" wird von HiCAD automatisch gesetzt.

■ Makrorahmen

Jedes Makro wird durch eine Anfangs- und Endanweisung gekennzeichnet.

START <i>menüebene</i> . . Anweisungen . END

menüebene ist dabei die Nummer der Menüebene, in der die Makroerstellung begonnen wurde. Beim Erstellen eines Makros wird diese Nummer von HiCAD automatisch gesetzt.

siehe 3.1.8 und 3.1.26, Start-/End-Befehl

1.3.2 Makrobefehle

Nach Aufruf der Funktion **Makro erstellen** und der Eingabe eines neuen Makronamens werden alle von nun an aktivierten Funktionen protokolliert. Dies gilt auch für fehlerhafte Eingaben. Aus diesem Grund ist eine sorgfältige Vorgehensweise erforderlich. Bevor Sie das Makro erstellen, sollten Sie sich daher den gewünschten Befehlsablauf genau überlegen.

Neben den Befehlen, die HiCAD während der Makroerstellung automatisch in das Makro einfügt werden, können Sie eine Vielzahl weiterer Befehle einfügen, z.B.

- Schleifenanweisungen,
- logische Abfragen mit Sprunganweisungen,
- Variablendeklarationen usw.

Befehle dieser Art müssen Sie jedoch explizit in das Makro eintragen. Dies kann nachträglich mit dem Makro-Editor oder einem beliebigen anderen Texteditor erfolgen.

siehe 1.5.3 und 3

1.3.3 Das freie Argument

Anstelle einer Wert-, Text- oder Namenseingabe kann bei der Makroerstellung auch das so genannte freie Argument, das Zeichen #, benutzt werden. Auf diese Weise werden Makros variabler gehalten.

Beim späteren Aufruf wird das Makro dann an dieser Stelle unterbrochen und HiCAD fordert die Eingabe des Wertes, Textes oder Namens an. Während der Makroerstellung muss zwar nach der Eingabe des freien Argumentes in vielen Fällen ein Wert, ein Text oder ein Name angegeben werden, damit der Erstellungsprozess ordnungsgemäß durchgeführt werden kann, in das Makro selbst wird jedoch nur das freie Argument eingetragen.

Ein Spezialfall ist die Punkteingabe. Wenn Sie während der Makroerstellung die Punktoption **RET** verwenden, wird das Zeichen # automatisch in das Makro eingetragen.

Die Benutzung des freien Argumentes soll an einem kleinen Beispiel verdeutlicht werden.

Beispiel:

Es soll ein Makro erstellt werden, das ein neues 2D-Teil anlegt. Der Name des Teils soll vom Anwender beliebig angegeben werden können. Dazu werden nacheinander dann folgende Schritte ausgeführt.

- Aufruf der Funktion **Makro erstellen**
- Eingabe des Makronames
- Auswahl von **2D-ZTL** und **ZTL bearbeiten**
- Auswahl von **Hauptteil neu** und **Teil anlegen**
- Eingabe des Teilennamens: **# BAUT1**
- Auswahl von **MAKRO-Ende**

Das dadurch erstellte Makro hat folgende Inhalt:

```

REM   HiCAD
START 59
REM   HiCAD 2 = 2-D ZTL
OPTION 2 59
REM   ZEICHENTEILE 3 = ZTL bearbeit.
OPTION 3 1
REM   BAUTEILE 8 = Hauptteil neu
OPTION 8 2
REM   NEUES TEIL 1 = Teil anlegen
OPTION 1 3
STRING #
END

```

Wird dieses Makro aufgerufen, so wird der Name eines Teils angefordert und dann unter dem eingegebenen Namen ein neues Teil angelegt.

Voraussetzung ist, dass eine Konstruktion geladen ist.

1.3.4 Makroerstellung beenden

Sie beenden die Makroerstellung durch Auswahl des Eintrages **MAKRO-Ende** im Haupttextmenü.

Der Makrostatus ist damit beendet.

1.4 Makro aufrufen

Zum Aufruf eines Makros wählen Sie Funktion **Makro aufrufen** mit der Taste

Strg+8

Nach der Eingabe oder Auswahl des Namens wird das Makro gestartet.

Ein Makro kann während der Abarbeitung mit der Taste **ESC** abgebrochen werden.

1.5 Makro bearbeiten

Um ein vorhandenes Makro zu bearbeiten, aktivieren Sie die Funktion **Makro erstellen** mit **Strg + 7**, und wählen das gewünschte Makro. In einem PopUp-Menü werden anschließend die möglichen Bearbeitungsarten zur Auswahl angeboten. Wählen Sie die gewünschte Bearbeitungsart.

Bei Auswahl von **abbrechen**, wird keine Aktion ausgeführt. Die Makrodatei wird sofort wieder geschlossen.

Bei Bearbeitung eines bereits vorhandenen Makros erzeugt HiCAD automatisch eine Sicherungskopie, mit dem Namen des Originals und der Dateinamenserweiterung .BAK. Frühere Sicherungskopien des gleichen Makros werden dadurch überschrieben. Diese Backup-Sicherungen sind nicht als Sicherung für die zu kompilierenden Makros geeignet, sondern eher als eine Sicherung der Editiervorgänge im Makro.

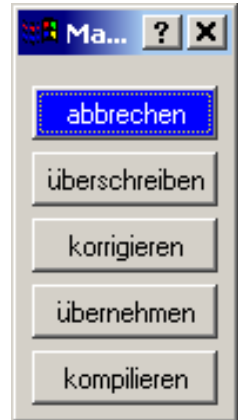


Abb. 3 Makro bearbeiten

1.5.1 Makro überschreiben

Mit dieser Funktion überschreiben Sie das gewählte Makro, d.h. der Inhalt der Datei wird gelöscht und die nachfolgend ausgeführten Funktionen werden in der Datei protokolliert.



Es wird keine Sicherungskopie der Ausgangsdatei angelegt!

1.5.2 Makro übernehmen

Mit dieser Funktion ist es möglich, Teile des Makros oder auch das gesamte Makro unverändert zu kopieren, um es anschließend um neue Befehle zu erweitern. Dazu müssen Sie die Anzahl der zu kopierenden Zeilen eingeben.



Abb. 4 Makro übernehmen

Als Vorgabewert wird die Gesamtanzahl der im Makro vorhandenen Befehlszeilen angeboten, so dass das gesamte Makro kopiert werden kann. In den häufigsten Fällen wird ein bestehendes Makro mit dieser Funktion um zusätzliche Befehle erweitert. Wird eine Zahl n eingegeben, die größer als 0 ist, dann werden die ersten n Zeilen des Makros übernommen. Ist n kleiner als 0, dann werden die letzten n Zeilen aus dem Makro gelöscht, d.h. alle Zeilen des Makros bis auf die letzten n Zeilen werden übernommen.

Das Makro läuft bei diesem Übernahmeprozess parallel dazu ab. Schleifen, bedingte Anweisungen und auch CALL-Befehle werden dabei berücksichtigt. Nachdem alle übernommenen Befehle abgearbeitet sind, wird in den Makroerstellungsmodus umgeschaltet. Sie haben dann die in Abschnitt 1.3 beschriebenen Möglichkeiten.



Es wird eine Sicherungskopie der Ausgangsdatei angelegt mit der Dateinamenserweiterung .BAK .

1.5.3 Makro testen/korrigieren

Mit dieser Funktion wird das Makro seitenweise im Makroeditor angezeigt.

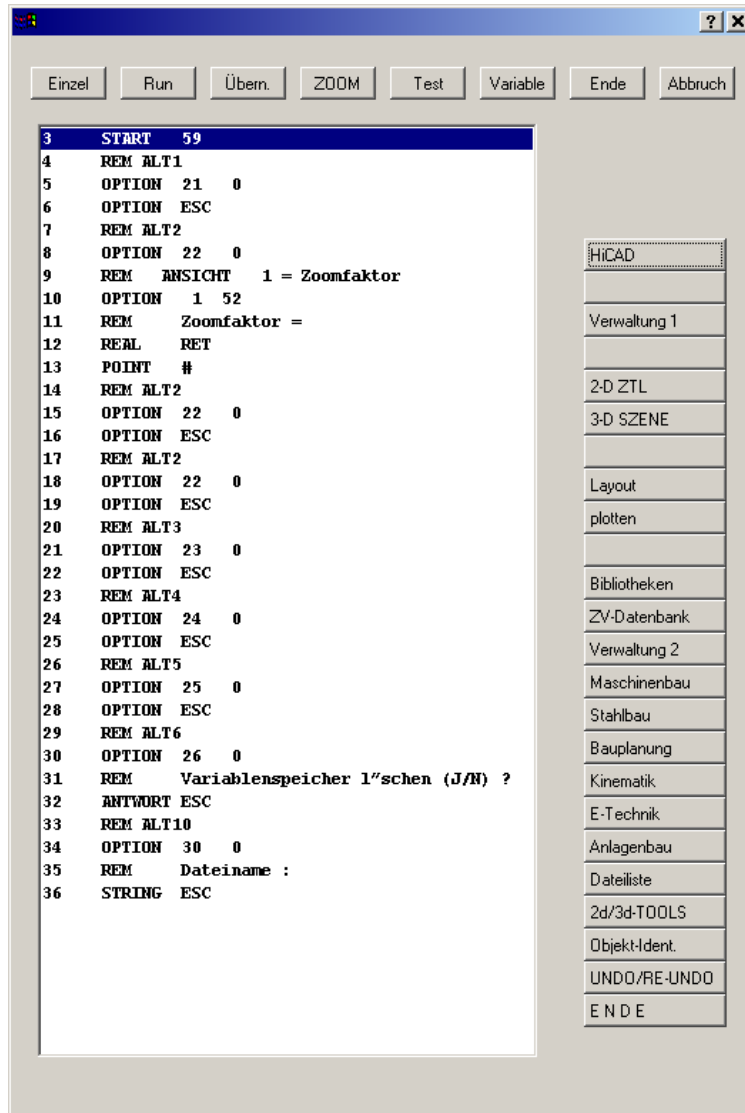


Abb. 5 Der Makroeditor

Dieses Fenster dient im Prinzip dazu, ein Makro zu testen

Die aktuelle Zeile eines Makros ist jeweils blau gekennzeichnet. Zu Beginn ist dies immer die Zeile mit dem **START**-Befehl.

Im oberen Teil des Makroeditors stehen verschiedene Funktionen zur Makrobearbeitung zur Verfügung, die Sie mit dem Cursor auswählen können.

1.5.3.1 Einzel

Mit dieser Funktion schalten Sie in den Einzelschrittmodus um. Mit **RUN** wird die aktuelle Zeile ausgeführt. Anschließend wird die nächste ausführbare Zeile zur aktuellen Makrozeile.

1.5.3.2 Zoom

Über dieses Feld haben Sie die Möglichkeit, die Makrobearbeitung temporär zu unterbrechen und z.B. einen anderen Bildausschnitt zu wählen oder ein Raster zu definieren. Nach Auswahl dieser Funktion wird ein entsprechendes PopUp-Menü eingeblendet.

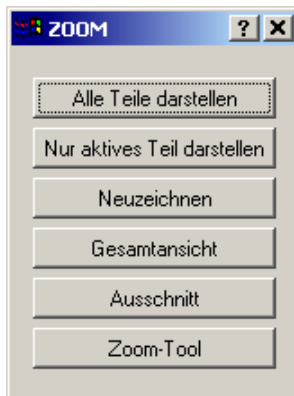


Abb. 6 Die Zoom-Funktionen

Nach Auswahl einer der hier verfügbaren Zoom-Funktionen wird wieder in die Makrobearbeitung zurückgeschaltet.

1.5.3.3 Run

Das Makro wird bis zur aktuellen, d. h. bis zur blau markierten Zeile ausgeführt. Im Einzelschrittmodus wird nur die aktuelle Zeile ausgeführt.

1.5.3.4 Test

Mit dieser Funktion können Sie prüfen, ob die Makrobefehle korrekt sind. Falsche Makrobefehle werden entsprechend markiert und können dann korrigiert werden.

1.5.3.5 Übernehmen

Mit dieser Funktion übernehmen Sie das Makro komplett bis zum Ende. Anschließend ist der Makroerstellungsmodus aktiv.

1.5.3.6 Ende

Diese Funktion arbeitet ähnlich wie die Funktion **Übernehmen**, jedoch wird das Makro hier nur bis zur aktuellen Zeile (blau markiert) übernommen. Anschließend ist der Makroerstellungsmodus aktiv.

1.5.3.7 Abbruch

Mit dieser Funktion wird die Makrobearbeitung abgebrochen. Die Änderungen werden nicht gesichert.

1.5.3.8 Variable

Mit dieser Funktion können Sie während der Makrobearbeitung Variablen ändern oder kontrollieren. Dazu werden die aktuell definierten numerischen Variablen und die Textvariablen angezeigt. Um eine Variablen zu ändern, wählen Sie diese aus und geben dann den neuen Wert oder Text ein. Mit **Übernehmen** wird der neue Wert für ein Makro übernommen.

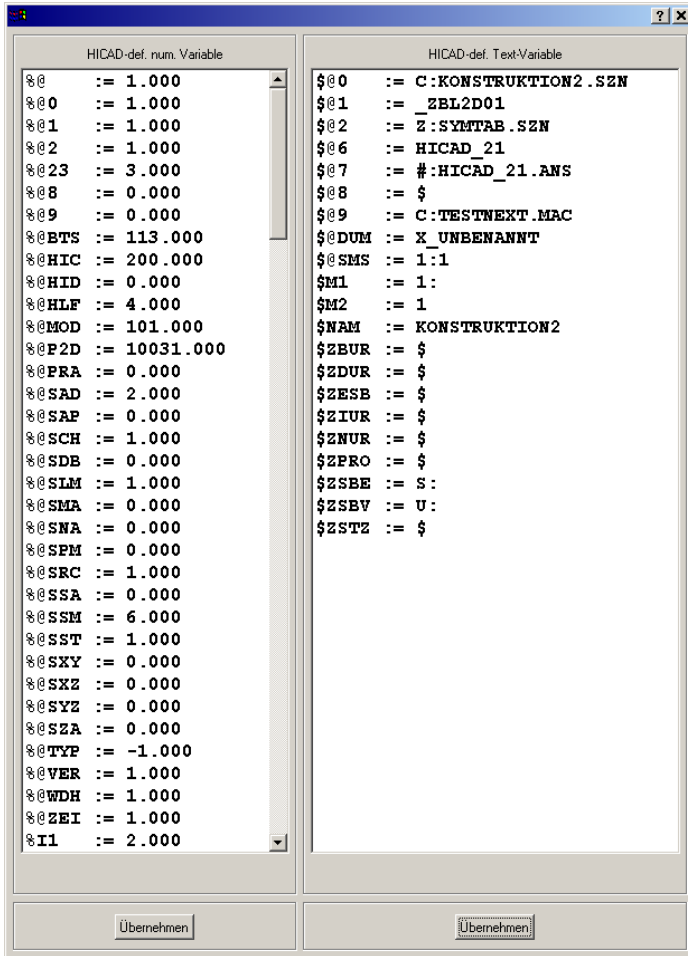


Abb. 7 Variablen kontrollieren und ändern

1.5.4 Kompilieren von Makros

Wird ein Makro zur Ausführung aufgerufen, dann muss zuerst die ASCII-Datei mit den Makrobefehlen eingelesen und in ein kompaktes Format für die anschließende Ausführung umgewandelt, d.h. kompiliert werden.

In diesem kompakten Format sind die einzelnen HCGS-Befehle durch ein einziges Byte verschlüsselt, Kommentarzeilen fehlen z.B. vollständig. Diese komprimierte Version benötigt einen Bruchteil des Speichers des Makros im lesbaren ASCII-Format

Diese kompilierte Version des Makros kann auch in der Datenbank abgelegt werden. Damit ist einerseits eine beträchtliche Speicherersparnis gegeben, aber auch die Ladezeiten des Makros verringern sich dementsprechend.

Ein Kompilieren von Makros ist vor allem für umfangreiche und sehr oft verwendete Makros empfehlenswert.

Kompilierte Makros können nicht mehr korrigiert bzw. erweitert werden.

Die Version mit dem lesbaren Quelltext im ASCII-Format ist nicht mehr verfügbar. Aus diesem Grunde sollten Sie von den Originalversionen zu kompilierender Makros in jedem Fall Sicherungen anlegen.

Diese Funktion ist optional erhältlich.

2 Allgemeines zur Makrosprache

Jeder HCGS-Befehl besteht aus zwei Teilen und hat den folgenden Aufbau:

Schlüsselwort Argumente

Sowohl die Schlüsselwörter als auch die Argumente lassen sich dabei in verschiedene Gruppen unterteilen.

2.1 Einteilung der HCGS-Befehle in Gruppen

Initialisierung	START, END Diese Befehle kennzeichnen Anfang und Ende des Makros und dürfen nur einmal auftreten.
HiCAD-Funktionen	OPTION Aktivierung von HiCAD-Funktionen
Skalare Eingabe	STRING, REAL, INTEGER, ANTWORT Diese Befehle werden vom HiCAD-Formelinterpreter benötigt.
Geometrische Eingabe	POINT, DISTANZ, WINKEL Diese Befehle werden bei Grafikeingaben benötigt.
Untermakro-Aufrufe	CALL, MAKRO Diese Befehle rufen aus dem aktuellen Makro heraus ein weiteres Makro auf.
Ausgabesteuerung MEIN,	APEIN, APAUS, ECHO, HFEIN, HFAUS, MAUS, SEIN, SAUS, SZEIN, SZAUS, WEIN, WAUS, WAIT, WARTE, UDA, UDE, ZAE, ZAA Diese Befehle steuern u.a. die Ausgaben im Textfenster.

Makrosteuerung	FOR...NEXT, WHILE...WHEND, REPEAT...UNTIL, GOTO, IF...THEN...ELSE...IFEND, IGNORE Mit diesen Befehlen werden Sprunganweisungen und Schleifen festgelegt.
Wertzuzuweisung	% , \$, VAI, VAR, PFD, DEL Mit diesen Befehlen erfolgt eine Wertzuzuweisung an Variable.
Dateiprozeduren	OPEN, INPUT, OUTPUT, CLOSE, COPY, MKDIR Mit diesen Befehlen werden ASCII-Dateien gelesen und geschrieben, Dateien kopiert und Verzeichnisse angelegt.
Kommentare	REM Mit diesem Befehl können Kommentare zum besseren Verständnis eines Makros eingefügt werden. Auf die Ausführung des Makros haben diese Befehle keine Auswirkung.

2.2 Die Argumentgruppen

Die in den HCGS-Befehlen möglichen Argumente lassen sich in folgende Gruppen unterteilen:

- Nummern** Nummern sind ganzzahlige, positive Werte.
- Dateinamen** Dateinamen sind Zeichenketten, die für einen gültigen Dateinamen stehen. HiCAD-Filegruppen (siehe 7), z.B. C:, sind erlaubt. Die Dateierweiterung wird abhängig vom geforderten Dateityp gesetzt (z.B. .MAC).
- Freies Argument** Das so genannte freie Argument wird durch das Zeichen # dargestellt. Tritt es in einem Makro auf, wird die Makroausführung an dieser Stelle unterbrochen und vom Benutzer eine Eingabe gefordert. Diese wird in der Regel durch einen entsprechenden Benutzerführungstext angezeigt und ist jeweils abhängig von dem Befehl in dem das freie Argument benutzt wird. Das freie Argument ist bei allen Eingabebefehlen erlaubt (Skalare und geometrische Eingaben).
- Konstanten** Konstanten sind aus beliebigen ASCII-Zeichen bestehende Zeichenketten, die keine Variablen enthalten dürfen. Leerzeichen werden in diesem Fall nicht als Separatoren, sondern als Bestandteile der Konstanten aufgefasst. Man unterscheidet:
- **ganzzahlige Konstante**
Integerwerte, d.h. ganze Zahlen ohne Dezimalstellen
 - **Realkonstante**
Zahlenwerte mit mindestens einer Vorkomma-
stelle und Dezimalpunkt '.' (Ein Komma als De-
zimalzeichen ist nicht zulässig !!).
 - **Stringkonstante**
Zeichenkette aus numerischen und alphanume-
rischen Zeichen.

Variablen

Bei benutzerspezifischen Variablen werden String- und Realvariablen unterschieden.

Stringvariable oder alphanumerische Variable beginnen immer mit dem \$-Zeichen und beinhalten eine beliebige, maximal 60-stellige Zeichenkette. Der Name der Variablen (ohne das \$-Zeichen) kann maximal 31-stellig sein und muss mit einem Buchstaben beginnen.

Bei **numerischen Variablen** wird nicht unterschieden zwischen Integer- und Realvariablen. Wird eine Integereingabe verlangt und eine reelle Variable als Argument eingegeben, so wird automatisch auf die nächste ganze Zahl gerundet. Das %-Zeichen kennzeichnet diese numerische Umwandlung in einen String. Bei Verwendung in Argumenten darf sonst kein % vor der Variable stehen. Der Name der Variablen kann maximal 31-stellig sein. Das erste Zeichen muss ein Buchstabe sein.

Groß-/Kleinschreibung wird bei Variablen nicht unterschieden.

siehe Abschnitt 2.3

arithmetische Ausdrücke

Anstelle eines Zahlenwertes kann in der Regel auch ein arithmetischer Ausdruck angegeben werden. Dieser kann auch Variable enthalten und liefert als Ergebnis einen Zahlenwert.

siehe Abschnitt 2.4

logische Ausdrücke

Ein logischer Ausdruck ist ein Vergleich zweier Werte oder arithmetischer Ausdrücke und kann auch Variable enthalten. Das Ergebnis eines logischen Ausdrucks ist entweder "wahr" oder "falsch".

siehe Abschnitt 2.5

feste Argumente

RET steht für RETURN. Bei Eingabe von RET wird der angebotene Wert übernommen.

JA ist nur beim ANTWORT-Befehl erlaubt. 'J' und '1' sind gleichwertige Eingaben.

NEIN komplementär zu JA. NEIN und '0' sind gleichwertig.

**Kontroll-
argumente**

ESC	entspricht dem ESC in HiCAD (mittlere Maustaste)
ZEI	darf nur im DISTANZ-Befehl vorkommen und entspricht der Angabe von 'z' innerhalb einer HiCAD-Eingabe
LLL	steht für <i>Letzte Linie Löschen</i>
LLA	steht für <i>Letzte Löschung Aufheben</i> (LLL und LLA sind nur im POINT-Befehl erlaubt)

Punktoption

spezielles Argument für den POINT-Befehl.

Erlaubte Punktoptionen sind: A, K, R, P, W, D, N, SP, L. Nicht protokolliert werden I, S, S2,M, M2, Z, F, O usw., also Punktoptionen , die sich auf Linien beziehen.

siehe auch Abschnitt 2.3.2

2.3 Variablen

2.3.1 Benutzerspezifische Variable

In HiCAD stehen Ihnen **numerische** und **alphanumerische Variablen** zur Verfügung.

Die Namen numerischer Variablen müssen mit einem Buchstaben beginnen, **ausgenommen ist der Buchstabe Z (außer ZA...)** sowie **L0 bis L9 und P0 bis P9**. Der Name einer numerischen Variablen darf inkl. des Buchstabens max. 31 Zeichen lang sein, anderenfalls wird er abgeschnitten.

Die Namen alphanumerischer Variablen müssen mit dem Zeichen \$ gefolgt von einem Buchstaben (mit Ausnahme von Z) beginnen, z.B. \$A1, \$A128 oder \$AB. Zwischen Groß- und Kleinbuchstaben wird nicht unterschieden. Der Name einer alphanumerischen Variablen darf inkl. Des \$-Zeichens und des ersten Buchstabens max. 10-stellig sein.

Wird der Wert an eine benutzerspezifische Variable nicht im Makro selbst zugewiesen, so wird zunächst der Variablenspeicher entsprechend geprüft. Ist der Variablen bisher kein Wert zugewiesen, dann wird der Makrolauf angehalten und eine Eingabe für diese Variable angefordert.

Ist der angegebenen Variablen bereits ein Wert zugewiesen, dann wird dieser als Defaultwert angezeigt. Sie können diesen Wert mit RETURN übernehmen oder die Variable mit END löschen.

2.3.2 Punkt- und Linienvariablen

Innerhalb der benutzerspezifischen Variablen haben die **Punktvariablen P0,...,P9** und die **Linienvariablen L0,...,L9** eine besondere Bedeutung. Über diese Variablen lassen sich vordefinierte grafische Informationen und Linienelemente ansprechen. Bei Punkt- und Linienvariablen erfolgt die Zuweisung grafisch, d.h. durch Bestimmung eines Punktes.

2.3.3 Die Systemvariablen

Neben den benutzerspezifischen Variablen stehen in HiCAD eine Reihe von Systemvariablen zur Verfügung, die wichtige Datenwerte enthalten und für eine effektive Makroentwicklung von außerordentlicher Bedeutung sind.

Die Systemvariablen können in der Makrotechnik gleichwertig zu den Benutzervariablen verwendet werden. Sie können diese Variablen mit ihrem Variablennamen ansprechen und so z.B. in arithmetischen Ausdrücken verwenden. Obwohl auch eine *Wertzuweisung* an diese Systemvariablen grundsätzlich möglich ist, sollten Sie damit eher vorsichtig umgehen, da diese Variablen bei entsprechenden Aktionen immer wieder verändert werden.

Systemvariablen werden hauptsächlich in den Sonderfunktionen gesetzt, die Sie mit der Funktion **Information** aufrufen können.

Eine Liste der Systemvariablen finden Sie im Anhang. Systemvariable, deren Namen mit ZA.. beginnen, sind anwenderdefinierte Systemvariable (Ausnahme ZA). Diese Variablen werden beim Löschen des Variablenspeichers nicht gelöscht.

Haben Sie Variablen das aktuelle Datum oder die aktuelle Zeit zugewiesen, z.B.

`$u:=TIM$` oder `$u:=DAT$`,

dann sind anschließend die folgenden Systemvariablen definiert:

ZSTU	aktuelle Stunde	ZJAR	aktuelles Jahr
ZMIN	aktuelle Minute	ZMON	aktueller Monat
ZSEC	aktuelle Sekunde	ZTAG	aktueller Tag

Bei der Variablen **ZJAR** wird die Jahreszahl immer 4-stellig geschrieben.

Beispiel:

<code>\$XX:=TIM\$</code>	<code>%T1:=ZSTU</code>
<code>%T2:=ZMIN</code>	<code>%T3:=ZSEC</code>

Nachfolgend finden Sie eine Auflistung der in HiCAD 2D und 3D zur Verfügung stehenden Systemvariablen.

2D-Variable

Z		Zentrifugales Trägheitsmoment
Z0		Fläche einer Figur oder einer geschlossenen Kontur
Z1		Umfang bzw. Länge eines Strecken- /Linien- /Konturzuges
Z2		Winkel (bei jeder Winkeleingabe gesetzt) Spezielle Bedeutung bei variabler Bemaßung haben: Z0 Typ: 1=Strecke, 2=Winkel, 3=Kreis, 4=Kreisbogen Z1 Art: 1=Einzel, 2=Kette, 3=Stufe, 4=Bezug, 5=Halbsch., 6=Hoeohen. Z2 Modus: 0=normal, 1=linienparallel
Z3		Länge eines Linienelementes bzw. Distanz (bei jeder Distanzeingabe gesetzt)
Z4		Statisches Moment (Drehmoment) bzgl. Y – Achse Nach Aufruf der Funktion Bemaßungsinfo: Achsenlage bei Streckenmaßen (1=x-parallel, 2=y-parallel, 0=nicht achsenparallel)
Z5		Statisches Moment (Drehmoment) bzgl. X – Achse
Z6		Trägheitsmoment bzgl. Y – Achse
Z7		Trägheitsmoment bzgl. X – Achse
Z8	X	Koordinaten des zuletzt angesprochenen Punktes bzw. des Anfangspunktes eines identifizierten Linienelementes
Z9	Y	
ZA		Aktueller Zoomfaktor
ZB		Anzahl der verspeicherten Passpunkte (beim Laden von Figuren gesetzt);
ZC	X	Koordinaten des Endpunktes eines identifizierten Linienelementes (wird bei jeder Linienidentifikation gesetzt)
ZD	Y	
ZCX	X	Koordinaten des Identifizierungspunktes bei der Identifizierung von Linien
ZCY	Y	
ZE		Statisches Linienmoment bzgl. X – Achse
ZH		Statisches Linienmoment bzgl. Y – Achse
ZF		Figur-Maßstab
ZG		Anzahl der Grafikschrime
ZGA1– ZGA4		Koordinaten der letzten Gesamtansicht
ZI		
ZJ	X	Koordinaten des Linienschwerpunktes eines Strecken-/Linien- /Konturzuges
ZK	Y	
ZL	X	Koordinaten des Mittelpunktes eines Kreisbogens
ZM	Y	
		Nach Aufruf der Funktion Bemaßungsinfo: X- bzw. Y-Koordinate der Maßzahl
ZN		Ergebnis bei Anforderung einer Auswertung (Taschenrechnerfunktion)
ZP	X	Koordinaten des Bildschirm – Mittelpunktes
ZQ	Y	

ZS		Punktbezeichnung des identif. Punktes oder Art des Passpunktes ZS=-1 ! ZS=-2 ? ZS=-3 ??
ZSC1, ZSC2		Schraffurcode 1, Schraffurcode 2
ZSD1, ZSD2		Abstand Schraffurcode1/2
ZSW1, ZSW2		Winkel Schraffurcode 1/2
ZT		Datenbank EIN/AUS [1/0]
ZU		2D/3D – Schalter (wird nur bei Makrostart belegt, 2=2D, 3=3D)
ZV		Wert der letzten numerischen Eingabe
ZW	X	Koordinaten der linken unteren Ecke der Gesamtansicht des Zeichenteiles (ungekürzte Werte) – gesetzt bei "Ganzes Zeichenteil" bzw. "Ausgewählte Figur"
ZX	Y	
ZY	X	Koord. der rechten oberen Ecke der Gesamtansicht des Zeichenteiles (ungekürzte Werte) – gesetzt bei "Ganzes Zeichenteil" bzw. "Ausgewählte Figur"
ZZ	Y	

3D-Variable

Z8	X-Koordinate Anfangspunkt Kante oder isolierter Punkt
Z9	Y-Koordinate Anfangspunkt Kante oder isolierter Punkt
ZB	Z-Koordinate Anfangspunkt Kante oder isolierter Punkt
ZBTP	Polares Flächenträgheitsmoment
ZBWY	Widerstandsmoment y bez. X-SP-Achse (SP=Schwerpunkt)
ZBWX	Widerstandsmoment x bez. Y-SP-Achse
ZC	X-Koordinate Endpunkt Kante
ZD	Y-Koordinate Endpunkt Kante
ZE	Z-Koordinate Endpunkt Kante
ZCTY	Kurventyp 0-Ebene, 1-Kreis, 2-Ellipse, 3-Freiform
ZEDZ	Anzahl identifizierter Kanten hintereinander liegend ≥ 0
ZFCZ	Anzahl identifizierter Facetten hintereinander liegend ≥ 0
ZFF.	Variablen für Freiformgeometrien
ZFFI	Die Indizierung für Freiformflächenkörper ist 0=nicht aktiv, 1=aktiv
ZFFJ	-1 oder 0 Es liegt kein indizierter Freiformflächenkörper vor oder die Indizierung kann nicht verwendet werden. >0 Index des Freiformflächenkörpers
ZFFN	Default-Generierung eines leeren Körpers für die Freiformflächen ist 0 nicht erfolgt/nicht notwendig 1 erfolgt
ZFSX	X-Koordinate Konturzug oder Flächenschwerpunkt
ZFSY	Y-Koordinate Konturzug oder Flächenschwerpunkt
ZFSZ	Z-Koordinate Konturzug oder Flächenschwerpunkt
ZFTR	0 oder 1, wenn beim Modellieren von Teile die Triangulation abgefragt wird
ZFTY	Flächentyp 0-Ebene, 2-Kugel, 3-Zylinder, 4-Kegel, 5-Torus, 6-Freiformfläche, <0-Hohlfläche
ZG8	X-Koordinate des zuletzt unter Info-3D-Punkt gewählten Punktes oder X-Koordinate des Anfangspunktes der zuletzt unter Info 3D-LE gewählten Kante
ZG9	Y-Koordinate des zuletzt unter Info-3D-Punkt gewählten Punktes oder Y-Koordinate des Anfangspunktes der zuletzt unter Info 3D-LE gewählten Kante
ZGB	Z-Koordinate des zuletzt unter Info-3D-Punkt gewählten Punktes oder Z-Koordinate des Anfangspunktes der zuletzt unter Info 3D-LE gewählten Kante
ZGC	X-Koordinate des Endpunktes der zuletzt unter Info 3D-LE gewählten Kante
ZGD	Y-Koordinate des Endpunktes der zuletzt unter Info 3D-LE gewählten Kante
ZGE	Z-Koordinate des Endpunktes der zuletzt unter Info 3D-LE gewählten Kante
ZHKV	Existenz einer Hyperkante 0=vorhanden, 1=nicht vorhanden
ZIAS	Nummer der aktuellen Ansicht
ZIAZ	Anzahl der definierten Ansichten
ZKTY	Teiletyp eines Körpers
ZKV.	Variable für Voreinstellungen bei Kantenzügen
ZKZL	Kantenzuglänge

ZKZE	Lineares Verbinden bei Kantenzügen 0 = gleiche Punkte-Anzahl der beiden Kantenzüge 1 = verschiedene Anzahl der beiden Kantenzüge
ZKEN	Bauteilkennung (siehe 4.10)
ZKZN	Anzahl der Punkte bei Kantenzug
ZKZV	-1 Interaktive Abfrage beim Verbinden von Kantenzügen 0 kürzeste Verbindung 3 Anfang -> Anfang 1 Ende -> Anfang 4 Anfang -> Ende 2 Ende -> Ende 5 Offene Kantenzügen schließen
ZMAX	X-Koordinate Maximumpunkt Hüllquader
ZMAY	Y-Koordinate Maximumpunkt Hüllquader
ZMAZ	Z-Koordinate Maximumpunkt Hüllquader
ZMIX	X-Koordinate Minimumpunkt Hüllquader
ZMIY	Y-Koordinate Minimumpunkt Hüllquader
ZMIZ	Z-Koordinate Minimumpunkt Hüllquader
ZMPX	X-Koordinate Mittelpunkt 3D
ZMPY	Y-Koordinate Mittelpunkt 3D
ZMPZ	Z-Koordinate Mittelpunkt 3D
ZOFL	Konturzug, Fläche, Oberfläche Volumina
ZOTY	Objekttyp für Identifizierung 1 Kantenzugkörper 2 FF-Gerüst-Komplex 0 Sonst
ZPAX	X-Koordinate Anfangspunkt Vektormenü
ZPAY	Y-Koordinate Anfangspunkt Vektormenü
ZPAZ	Z-Koordinate Anfangspunkt Vektormenü
ZPEX	X-Koordinate Endpunkt Vektormenü
ZPEY	Y-Koordinate Endpunkt Vektormenü
ZPEZ	Z-Koordinate Endpunkt Vektormenü
ZPRO	Nummer des aktiven Projektes (Datenbank)
ZRA1	Kreis, Kugel, Zylinderradius, großer Torusradius
ZRA2	kleiner Torusradius
ZSUB	Flag für Teilegenerierung 0-Hauptteilgenerierung 1-Nebenteilgenerierung
ZYAX	X-Koordinate Achsvektor (Zylinder, Kegel, Torus)
ZYAY	Y-Koordinate Achsvektor (Zylinder, Kegel, Torus)
ZYAZ	Z-Koordinate Achsvektor (Zylinder, Kegel, Torus)
ZVKX	X-Koordinate Vektor mit Betrag bei Vektormenü
ZVKY	Y-Koordinate Vektor mit Betrag bei Vektormenü
ZVKZ	Z-Koordinate Vektor mit Betrag bei Vektormenü
ZVNX	X-Koordinate Konturzug oder Flächennormalenvektor
ZVNY	Y-Koordinate Konturzug oder Flächennormalenvektor
ZVNZ	Z-Koordinate Konturzug oder Flächennormalenvektor
ZVOL	Volumen eines Teils
ZV SX	X-Koordinate Volumenschwerpunkt
ZV SY	Y-Koordinate Volumenschwerpunkt
ZV SZ	Z-Koordinate Volumenschwerpunkt
ZWKG	Kegelwinkel

ZXP0	Polyeder- oder analytisches Modell -1 Modell unbestimmt 10 analytisches Modell (exakt)	0 Polyeder 11 analytisches Modell (approximiert)
ZXP2	aktives Modell 0=analytisch, 1=Polyeder	
ZPX3	Polyederapproximation	
ZXPC	Feinheit von Cons- und FFF-Kurven	
ZXPG	Generierung von Gitterlinien bei der Polyederapproximation von Freiformflächen 0 nicht für ebene Freiformflächen 1 auch für ebene Freiformflächen	
ZXPO	Approximation von CONSES -1 nur für ebene Freiformflächen 1 auch zwischen Gitterlinien	
ZXPU	Feinheit von u-Gitterlinien (FFF)	
ZXPV	Feinheit von v-Gitterlinien (FFF)	
Z3DK	Statusvariable zum automatischen Makrostart (Umschaltung 2D/3D) 0 kein Makrostart 1 allgem. Zylinder 2 Rotationsteil 3 Bohren/Schneiden 4 Ausnehmen 5 Zerschneiden 6 Schnittansicht	

Numerische Systemvariable

@		Umrechnungsfaktor der Maßeinheit in Millimeter
@0		Umrechnungsfaktor Naturkoordinaten in Millimeter (@/@2)
@1		Umrechnungsfaktor Millimeter in Naturkoordinaten (@2/@)
@2		Maßstabszahl
@3	A	Koeffizienten A, B und C der Hesseschen Normalform einer Geraden: $A*x +$
@4	B	$B*y = C$ mit $A*A + B*B = 1$.
@5	C	Diese Variablen werden gesetzt, wenn mit der Funktion 4 innerhalb von ALT 3 eine Strecke identifiziert wird;
@5		Radius eines identifizierten Kreises
@6	X	Koordinaten des Schwerpunktes einer Figur oder eines geschlossenen Lini-
@7	Y	enzuges;
@8	X	Koordinaten des Bezugspunktes der Konstruktion (Kürzungswert);
@9	Y	
@SKZ		Skizziercursor ein-(ausschalten) 0=aus, anderer Wert oder nicht belegt = ein (siehe auch 4.9)

Neben den numerischen Systemvariablen kennt HiCAD auch alphanumerische Systemvariablen, die mit \$@ bis \$@9 usw. angesprochen werden.

Alphanumerische Systemvariable

\$@		Formatstring für die Umwandlung von Zahlen in Texte (F8.2), (I3), (F12.6) usw. Dies entspricht der Fortran-Formatanweisung.
\$@0		Name des aktiven Zeichenteiles
\$@1		Name der aktiven Figur
\$@2		Name der zuletzt geladenen Symboltabelle
\$@3		Text eines identifizierten Textbezugspunktes (wird in ALT 3 und in der Textinformation gesetzt)
\$@4		Name der Digitizerzuordnung Tablett/Bildschirm
\$@9		letzte Texteingabe
\$@AWV		aktuell eingestelltes Verfahren für die Blechabwicklung, z.B. F = Faktorenverfahren (DIN). Das Ergebnis ist identisch mit dem Kennungsbuchstaben in der Datei <i>verfahren.DAT</i> , wobei <i>verfahren</i> der Name der entsprechenden DIN ist, z.B. DIN 6935.DAT. Diese Dateien liegen im Unterverzeichnis <i>MAKROABW</i> .
\$@BMH		Info über Maßhilfslinien, wird bei der Funktion Bemaßungsinfo auf 1,2,3 bzw. 11,12,14 gesetzt, wenn die Maßhilfslinien ausgeblendet sind bzw. der Schalter Z-Bemaßung gesetzt ist.

2.4 Arithmetische Ausdrücke

In numerischen Eingaben können beliebige arithmetische Ausdrücke verwendet werden. Die Weiterverarbeitung dieser Ausdrücke kann auch an logische Bedingungen geknüpft werden.

Unter einem **arithmetischen Ausdruck** wird dabei eine im Sinne der Algebra eine gültige Konstruktion aus numerischen Konstanten, numerischen Variablen, arithmetischen Operatoren und Klammern verstanden, wobei die maximale Länge eines Ausdrucks durch 60 Zeichen beschränkt ist. Als Klammern dürfen nur runde Klammern verwendet werden, eckige oder geschwungene Klammern sind nicht erlaubt. Die Verwendung mehrerer ineinander geschachtelter Klammern ist zulässig, die Klammerung muss jedoch paarig sein, d.h. zu jeder öffnenden muss auch eine schließende Klammer vorhanden sein.

2.4.1 Rechenoperatoren

Folgende Operatoren können benutzt werden:

+	Addition
-	Subtraktion
*	Multiplikation
/	Division
^	Exponentialoperator

Anstelle des Operators ^ kann auch der Funktionsname XHY (x hoch y) verwendet werden. Die Auswertung eines arithmetischen Ausdrucks erfolgt nach den üblichen mathematischen Regeln.

2.4.2 Grundfunktionen

In einem Ausdruck ist auch die Verwendung der wichtigsten mathematischen Funktionen und Stringfunktionen möglich.

ABS	Absolutwert
ACOS	Arcus Cosinus
AINT	Integer (ganzzahliger Anteil)
ARC	Umwandlung Gradmaß in Bogenmaß
ASC	ASCII – Code eines Zeichens
ASIN	Arcus Sinus
ATAN	ArcusTangens
COS	Cosinus
COSH	Cosinus Hyperbolicus
EXP	Exponentialfunktion
GRD	Umwandlung Bogenmaß in Gradmaß
LEN	Länge eines Strings
LOG	natürlicher Logarithmus
LOG10	dekadischer Logarithmus
NINT	Nächster Integerwert (rundet auf nächste ganze Zahl)
SIG	Signum (Vorzeichenfunktion: -1, 0 oder 1)
SIN	Sinus
SINH	SinusHyperbolicus
SQR	Quadrat
SQRT	Quadratwurzel
TAN	Tangens
TANH	Tangens Hyperbolicus
VAL	String in Zahl umwandeln

Beispiel: `%C:=SQRT (A*A + B*B)` \longrightarrow $A^2 + B^2$

Obwohl die trigonometrischen Funktionen im Allgemeinen nur Werte im Bogenmaß verarbeiten, werden innerhalb des HiCAD-Formelinterpreters Eingaben und Ergebnisse im Gradmaß, abhängig von der eingestellten Winkeleinheit, verarbeitet.

Zur Umrechnung zwischen Grad- und Bogenmaß können die Funktionen ARC und GRD verwendet werden.

2.5 Logische Vergleichsausdrücke

Unter einem **logischen Vergleichsausdruck** versteht man einmal einen einfachen arithmetischen Vergleich

$$a1 \text{ op } a2$$

wobei $a1$ und $a2$ Konstanten, Variablen oder arithmetische Ausdrücke sind, die mit einem Vergleichsoperator **op** verknüpft werden, **op** kann dabei einer der folgenden Operatoren sein:

=	gleich	< >	ungleich
<	kleiner als	>	größer als
<=	kleiner oder gleich	>=	größer oder gleich

Ein logischer Vergleichsausdruck kann aber auch aus zwei einfachen arithmetischen Vergleichen mit Hilfe der logischen Operatoren **AND** und **OR** gebildet werden. Darüber hinaus kann ein logischer Ausdruck auch durch NOT negiert werden.

Logische Vergleichsausdrücke sind z.B. dann von Bedeutung, wenn die Wertzuweisung an eine Variable von bestimmten Bedingungen abhängig gemacht werden soll.

Dies soll anhand von Beispielen erläutert werden. Folgende Variablen seien gesetzt:

```
%a1:=5
```

```
%a2:=1
```

```
$TEXT:=ABCDEF
```

Beispiel 1:

```
IF a1 <= 5 THEN  
. .  
IFEND
```

Da a1 den Wert 5 hat, ist der logische Ausdruck "a<=5" wahr, d.h. die Befehle bis zum IFEND werden ausgeführt.

Beispiel 2:

```
WHILE a2 < 10  
. .  
%a2:=a2+1  
WHEND
```

Hier wird die Schleifenvariable am Ende eines Schleifendurchlaufes jeweils um 1 erhöht. Damit ist der logische Ausdruck "a2<10" für die ersten 9 Schleifendurchläufe wahr. Danach wird die Schleife beendet.

Beispiel 3:

```
IF NOT a1 <= 5 THEN  
. .  
IFEND
```

Da a1 den Wert 5 hat, ist der logische Ausdruck "NOT a<=5" falsch, d.h. die Befehle bis zum IFEND werden nicht ausgeführt.

Beispiel 4:

```
IF $TEXT="ABCDEF" THEN  
. .  
IFEND
```

Da TEXT den Wert ABCDEF hat, ist der logische Ausdruck richtig, d.h. die Befehle bis zum IFEND werden ausgeführt.

Beispiel 5:

```
IF $TEXT(1:3)="ABC" THEN  
. .  
IFEND
```

Da die ersten drei Buchstaben von TEXT die Zeichenkette ABC bilden, ist der logische Ausdruck richtig, d.h. die Befehle bis zum IFEND werden ausgeführt.

2.6 Logische Variable

Neben logischen Ausdrücken bietet HiCAD auch die Möglichkeit, logische Variable zu benutzen, die gerade in Schleifen und bei IF-Bedingungen sinnvoll eingesetzt werden können. Es stehen die folgenden logischen Variablen zur Verfügung:

3D	WAHR, wenn der Makrostart im 3D erfolgt.
BEMA	WAHR, wenn Bemaßungen dargestellt werden
DVORHD	WAHR, wenn bei Dateizugriffen der Datensatz vorhanden ist
ESC	WAHR, wenn die ESC-Taste (rechte Maustaste) betätigt wird
FEATURE	WAHR, wenn das aktive 3D-Teil ein Featureprotkoll hat. Dabei wird berücksichtigt, dass das Protokoll evtl zu einem übergeordneten Teil gehört (z.B. Blechlasche)
FEHL	WAHR, wenn ein Fehler aufgetreten ist, z.B. in einer POINT-Anweisung
INT	WAHR, wenn die INT-Taste (mittlere Maustaste) betätigt wird
ISOP	WAHR, wenn der Schalter ISOLIERTE PUNKTE auf EIN steht
JA	WAHR, wenn bei einer J/N-Abfrage mit JA geantwortet wird.
NEIN	WAHR, wenn bei J/N-Abfrage mit NEIN geantwortet wird.
PBEZ	WAHR, wenn Punktbezeichnungen eingeschaltet sind
PESC	WAHR, wenn eine Punkteingabe mit END bestätigt wird.
PINT	WAHR, wenn bei einer Punkteingabe INT gewählt wird
SCHR	WAHR, wenn der Schalter SCHRAFFUR J/N auf Ja steht
SYMB	WAHR, wenn der Schalter SYMBOLE EIN/AUS auf EIN steht
TEXT	WAHR, wenn der Schalter TEXTE EIN/AUS auf EIN steht
VALD	WAHR, wenn verallgemeinerte Liniendarstellungen eingeschaltet sind
VORHD	WAHR, wenn bei Dateizugriffen die Datei vorhanden ist WAHR, wenn bei einer Teileauswahl das Teil vorhanden ist WAHR, wenn bei der Bemaßung zwischen den Maßhilfslinien genügend Platz für die Maßzahl ist. WAHR, wenn eine numerische Variable bereits gesetzt ist (WERT-Befehl).

Beispiel 1

```
ANTWORT #  
IF JA THEN  
.  
.  
IFEND
```

Wird auf die Ja/Nein-Abfrage mit JA geantwortet, dann hat die logische Variable JA den Wert WAHR, d.h. die IF-Bedingung ist erfüllt und die Anweisungen bis zum IFEND werden ausgeführt.

Beispiel 2

```
IF NOT VORHD GOTO 99  
.  
.  
99:END
```

Ist ein neues Teil gewählt worden, z.B. durch Angabe des Namens und ist dieses Teil nicht vorhanden, dann wird das Makro beendet.

Beispiel 3

```
IF FEATURE THEN  
.  
.  
IFEND
```

Hat das aktive Teil ein Feature-Protokoll, dann hat die logische Variable FEATRUE den Wert WAHR, d.h. die IF-Bedingung ist erfüllt und die Anweisungen bis zum IFEND werden ausgeführt.

2.7 Stringausdrücke

Alphanumerische Variablen sowie die Systemvariablen können ebenfalls in einem Ausdruck benutzt werden. Der Formelinterpreter unterscheidet, ob die eingegebenen alphanumerischen Zeichen als Stringvariablen oder Stringkonstanten zu interpretieren sind. Ferner können numerische in alphanumerische Variable umgewandelt werden. Es können auch Funktionsaufrufe mit Stringvariablen durchgeführt werden, die numerische Ergebnisse liefern.

2.7.1 Numerische Variable in String umwandeln

Wird einer Stringvariablen im Makro eine numerische Variable zugewiesen, dann wird der Inhalt dieser numerischen Variable in einen String umgewandelt, z.B.

```
$A:=%B
```

Bei dieser Prozedur hat die Systemvariable \$@ eine besondere Bedeutung. Der Inhalt dieser Systemvariablen wird als FORTRAN-Format aufgefasst und zur Umwandlung verwendet. Der Formatstring muss in runde Klammern eingeschlossen sein und kann neben einem numerischen Format auch noch andere Formatanweisungen, beispielsweise einen Zusatztext in Form eines Hollerith-Formates enthalten. Die Verwendung der Variablen \$@ kann vor allem für *Beschriftungen* in der Zeichnung vorteilhaft sein.

Wird der Variablen \$@ beispielsweise der String

```
$@:=(4HF1.: , F8.2 , 5H m^^2 )
```

zugeordnet und dann der Inhalt der Systemvariablen Z0 als Text in die Zeichnung eingefügt, so erscheint die Ausgabe in der Form:

```
Fl.: 148.25 m2
```

Die Zuordnung

```
$@:=(I4)
```

würde beispielsweise eine vierstellige Integerzahl erzeugen.

Die Systemvariable \$@ sollte anschließend wieder gelöscht werden.

Ein String, der mit dem % – Zeichen beginnt, dem aber kein Variablenname folgt, wird ohne Änderung als Stringkonstante übernommen.

2.7.2 Stringoperationen

Strings können in HiCAD auch verknüpft werden. Die Verknüpfung erfolgt mit dem Operator +, wobei zwischen den Operanden und dem +-Zeichen kein Leerzeichen stehen darf.

Beispiele:

\$A:=Text1

\$B:=Text2

\$TEXT:=ABCDEF

\$C:=\$A+\$B → Text1Text2

\$C:="Dies ist "+\$A → Dies ist Text1

\$NAME:=\$TEXT(4:5) → DE

2.7.3 Stringfunktionen

Folgende Stringfunktionen stehen in HiCAD zur Verfügung:

- ASC**(string) ASCII-Code des ersten Zeichens des Strings
- LEN**(string) Länge des Strings. Der String kann dabei eine Stringkonstante oder eine Stringvariable sein. Das Ergebnis ist numerisch.
- CHR**\$(num) ASCII-Zeichen, das dem Ergebnis des numerischen Ausdrucks *num* entspricht. Der Wert von *num* darf nur ganzzahlige Werte zwischen 0 und 255 annehmen.
- VAL**(string) Umwandeln eines Strings in einen numerischen Wert. Voraussetzung ist, dass der Inhalt des Strings numerisch interpretiert werden kann. Der an die Funktion übergebene String darf maximal 19 Zeichen lang sein und keine Exponentialschreibweise (1.0E+8 o.ä.) enthalten.
- LTU**\$(string) Umwandeln des Strings in Großbuchstaben. Der String kann dabei eine Stringkonstante oder eine Stringvariable sein.
- UTL**\$(string) Umwandeln des Strings in Kleinbuchstaben.

- IDX(\$A,\$B)** Teilstring suchen. Der Inhalt der Stringvariablen \$B wird im Inhalt der Stringvariablen \$A gesucht. Ist \$B in \$A vorhanden, dann liefert die Funktion den Anfangsindex von \$B, d. h. die Position an der der Teilstring gefunden wurde. Ist \$B in \$A nicht vorhanden, dann liefert die Funktion den Wert 0. Nach Leerzeichen kann nicht gesucht werden. Werden anstelle der Variablen Stringkonstante verwendet, dann dürfen diese kein Komma enthalten.
- TIM\$** aktuelle Zeit in der Form HH:MM:SS. Das Ergebnis ist ein String.
- DAT\$** aktuelles Datum in der Form JJ:MM:TT. Das Ergebnis ist ein String.

Beispiele:

Folgende Variablen seien gesetzt: \$A:=TEXT, %A1:=84, \$B:=EX

%I1:=LEN(\$A)	→	4
%I1:=ASC(\$A)	→	84
\$C:=CHR\$(A1)	→	T
\$C:=TIM\$	→	13:52:18
\$C:=DAT\$	→	95:11:24
\$C:=UTL\$(\$A)	→	text
%I1:=IDX(\$A,\$B)	→	2

2.8 Der Variablenspeicher

Mit der Funktion **Variable** können Sie benutzerdefinierte Variablen aus dem Variablenspeicher löschen oder neue Variablen definieren.

Sie aktivieren diese Funktion durch Auswahl von **Information/Datenstrukturtest/Variablenspeicher** und



Variablenspeicher löschen

im Bildschirmmenü.

Mit **ja** werden alle benutzerdefinierten Variablen aus dem Variablenspeicher gelöscht.

Um neue Variablen zu definieren, wählen Sie die Funktionen

- numerische Variable setzen,
- Stringvariable setzen,
- Punktvariable setzen oder
- Linienvariable setze

im Menü **Information/Datenstrukturtest/Variablenspeicher**.

3 Die Befehle der Makrosprache

Makros werden in der Makrosprache HCGS erstellt, deren Elemente speziell auf HiCAD abgestimmt sind. Im Folgenden finden Sie die Beschreibung der HCGS-Befehle in alphabetischer Reihenfolge. Ein Befehl wird nach folgendem Muster beschrieben:

1. Der Befehl (Schlüsselwort)

Für die Makrosprache sind nur die ersten 3 Stellen signifikant, auch wenn immer der komplette Befehlsname für das Makro generiert wird.

2. Die Funktion

Eine Kurzbeschreibung der Befehls.

3. Die Syntax

Verbindliche Schreibweisen für Befehle und Argumente.

4. Die Argumente

Eine Auflistung aller erlaubten Argumente. Sind Argumente durch "/" voneinander getrennt, so muss genau eines dieser Argumente verwendet werden. In eckige Klammern eingeschlossene Argumente sind optional. Groß geschriebene Argumente sind feste Ausdrücke und müssen buchstabengetreu eingegeben werden; sind sie klein und kursiv geschrieben, sind es Platzhalter (z.B. Variablen). Argumente müssen durch ein Leerzeichen untereinander und auch vom Befehl getrennt sein. Eine Ausnahme ist der REM-Befehl.

Hier sind nur Besonderheiten aufgeführt, die bei den Argumenten zu beachten sind. Eine ausführliche Beschreibung der verschiedenen Argumente finden Sie im Abschnitt 2.

5. Hinweise

Hier finden Sie ausführliche Hinweise zum aktuellen Befehl.

6. Generierung durch das HiCAD-Entwicklungssystem

Hier wird angegeben, welche Befehle während der Makroerstellung automatisch generiert werden. Befehle, die nicht automatisch generiert werden können, müssen Sie

- entweder während der Makroerstellung mit dem Befehl **Makrozeile einfügen**
- oder nachträglich mit dem Makroeditor einfügen.

7. Beispiel

Hier finden Sie Anwendungsbeispiele zum aktuellen Befehl.



Hinweise:

- Jeder Befehl muss im Makro in einer neuen Zeile beginnen und darf sich auch nur über diese eine Zeile erstrecken. Leerzeichen werden ignoriert.
- Die nachfolgende Beispiele wurden ohne Datenbank erstellt.

3.1 Alphabetische Befehlsübersicht

APEIN, APAUS	Autopilot ein/aus
ANTWORT	Antwort bei Ja/Nein-Abfragen
CALL	Untermakro aufrufen
CLOSE	Datei schließen
COPY	Kopierbefehl für Makrotechnik
DEL	Löschen von Variablen
DISTANZ	Angabe einer Distanz relativ zum aktuellen Punkt
ECHO	Anzeigen eines frei definierbaren Benutzertextes im Textfenster
END	Makroende
FOR ... NEXT	Schleife mit einer bestimmten Anzahl von Durchläufen
GOTO	Sprunganweisung
IF ... IFEND	IF-Abfrage
IGNORE	Überspringen von Befehlen
INPUT	aus Datei lesen
INTEGER	Eingabe einer ganzen Zahl
MAKRO	Untermakro aufrufen
MAUS , MEIN	Kantenzugmarkierung ein-/ausschalten
MKDIR	Verzeichnis anlegen
OPEN	Datei öffnen
OPTION	Funktion aus Menü wählen
OUTPUT	in Datei schreiben
PFD	Pfad ermitteln
POINT	Punktbestimmung
REAL	Eingabe einer reellen Zahl
REM	Kommentar
REPEAT .. UNTIL	Bedingungsschleife
SAUS, SEIN	Benutzerführungstexte ein-/ausschalten
START	Makroanfang
SZEIN, SZAUS	Statuszeilenupdate

STRING	Texteingabe
UDA, UDE	3D-Undosicherung ein/aus
VAI	Eingabe von Integer-Variablen
VAR	Variableneingabe
WAIT, WARTE	Pause
WAUS, WEIN	Pause ein-/ausschalten
WERT	prüft, ob eine numerische Variable vorhanden ist
WHILE ... WHEND	Bedingungsschleife
WINKEL	Winkeleingabe
ZAE, ZAA	Bildaufbau abbrechen
%variable:=	Wertzuweisung an numerische Variable
\$variable:=	Wertzuweisung an eine Stringvariable

3.1.1 Antwort

■ Funktion

Alternativauswahl bei Ja/Nein-Abfrage

■ Syntax

ANTWORT JA / NEIN / # / ESC / 0 / 1

■ Argumente

JA/1 *siehe feste Argumente*

NEIN/0 *siehe feste Argumente*

siehe freies Argument

ESC *siehe Kontrollargument*

■ Hinweise

Der ANTWORT-Befehl dient zur Eingabe bei Ja/Nein-Abfragen.

RET ⇒ ANTWORT 0

JA ⇒ ANTWORT 1

NEIN ⇒ ANTWORT 0

Vorsicht ist hier bei der Verwendung des freien Arguments # geboten !

■ Generierung durch HiCAD-Entwicklungssystem

Der Befehl wird automatisch generiert, wenn in HiCAD eine Ja/Nein-Abfrage angefordert wird.

■ Beispiele

Dieses Makro speichert die aktuelle Konstruktion und lädt eine neue, deren Namen Sie eingeben müssen.

```

REM   HiCAD-Next VN:1201
START 59
REM   HiCAD 3 = 3-D SZENE
OPTION 3 59
REM   3D - SZENE 2 = Laden
OPTION 2 101
REM   Aktive Konstruktion speichern ?
ANTWORT 1
STRING RET
OPTION ESC
END

```

Mit dem folgenden Makro wird das aktive 3D-Teil gelöscht. Die Löschung wird hier automatisch mit JA bestätigt. Wird anstelle von 1 das freie Argument # angegeben, dann erfolgt eine Abfrage.

```
REM   HICAD-Next VN:1201
START   59
REM   HiCAD   3 = 3-D SZENE
OPTION  3 59
REM   3D - SZENE   3 = Bearbeiten
OPTION  3 101
REM   TEILE   7 = Löschen
OPTION  7 102
REM   O.K. ?
ANTWORT 1
END
```

3.1.2 APAUS / APEIN

■ Funktion

Aus-/Einschalten des Autopiloten

■ Syntax

APAUS

APEIN

■ Argumente

keine

■ Generierung durch HiCAD-Entwicklungssystem

Es ist keine automatische Generierung möglich.

■ Beispiel

Mit dem folgenden Makro wird ein Rechteck erzeugt. Vor der Bestimmung des ersten Eckpunktes wird der Autopilot ausgeschaltet, nach der Bestimmung des zweiten Eckpunktes wieder eingeschaltet.

```
REM   LINIEN
START   6
REM   LINIEN   4 = Rechteck
APAUS
OPTION   4   6
POINT   #
POINT   #
APEIN
END
```

3.1.3 CALL

■ Funktion

Mit diesem Befehl wird aus dem aktuellen Makro heraus ein anderes Makro aufgerufen, abgearbeitet und anschließend das aktuelle Makro fortgesetzt.

■ Syntax

CALL *dateiname*

■ Argumente

dateiname

Die Datei *dateiname* muss ein gültiges Makro mit HCGS-Befehlen sein.

■ Generierung durch HiCAD-Makroentwicklungssystem

Es ist keine automatische Generierung möglich.

■ Beispiel

In diesem Beispiel wird eine Konstruktion geladen. Dann wird mit dem CALL-Befehl das Makro **TEXTE** aufgerufen, das abfragt, ob Texte dargestellt werden sollen oder nicht und einen Bildneuaufbau durchführt. Anschließend wird das aktuelle Makro fortgesetzt.

```

REM   HICAD-Next VN:1201
REM           HiCAD
START   59
HNEXT
REM   HiCAD   3 = 3-D SZENE
OPTION  3 59
REM   3D - SZENE   2 = Laden
OPTION  2 101
STRING *
OPTION  ESC
REM   BERECHNUNG   2 = Eigene Verf.
OPTION  2 159
CALL C:TEXTE
REM   ZEICHENTEILE   3 = ZTL bearbeit.
OPTION  3 1
REM   BAUTEILE   8 = Hauptteil neu
OPTION  8 2
REM   NEUES TEIL   1 = Teil anlegen
OPTION  1 3
STRING RET
REM   BAUTEILE   1 = Linien
OPTION  1 2
REM   LINIEN   2 = Streckenzug
OPTION  2 6
POINT #
END

```

Makro TEXTE

```

REM   HICAD-Next VN:1201
START   59
REM ALT1
OPTION  21 0
REM   NEUZEICHNUNG   6 = Texte      E/A
OPTION  6 51
REM   Texte darstellen ?
ANTWORT #
OPTION  ESC
REM ALT1
OPTION  21 0
REM   NEUZEICHNUNG   1 = alle Teile
OPTION  1 51
END

```

3.1.4 COPY

■ Funktion

Kopieren von Dateien

■ Syntax

`COPY dateiname1 dateiname2`

`COPY $variable1 $variable2`

■ Argumente

dateiname1 dateiname2

dateiname1 ist der Name der zu kopierenden Datei, *dateiname2* der Name der Kopie. Den Dateinamen kann ggf. eine Pfadangabe oder eine Filegruppe gemäß der Datei FILEGRUP.DAT vorangestellt werden. Die Gesamtlänge des Copy-Befehls beträgt hier 60 Zeichen.

siehe auch MKDIR und PFD

\$variable1 \$variable2

Hier werden die Dateinamen (ggf. mit Pfad) aus den Textvariablen *\$variable1* und *\$variable2* gelesen. Der Inhalt der beiden Variablen kann jeweils 60 Zeichen lang sein.

■ Generierung durch HiCAD-Makroentwicklungssystem

Es ist keine automatische Generierung möglich.

■ Beispiel 1

Die Datei TEST.DAT soll aus dem Verzeichnis C:\HICADNEXT\SZENEN in das Verzeichnis E:\DATEN kopiert werden.

```
REM   HICAD-Next VN:1201
START 1
$A:=c:\hcadnext\szenen\TEST.DAT
$B:=e:\Daten\TEST.DAT
COPY $A $B
END
```

■ Beispiel 2

Die Datei TEST.DAT soll aus der Filegruppe C in die Filegruppe Z kopiert werden. Diese Filegruppen sind in der Datei FILEGRUP.DAT im HiCAD-Unterverzeichnis sys definiert.

```
REM   HICAD-Next VN:1201
START 1
$A:=C:TEST.DAT
$B:=Z:TESTNEU.DAT
COPY $A $B
END
```

3.1.5 DEL

■ Funktion

Löschen von numerischen Variablen und Textvariablen

■ Syntax

DEL *%variable*

DEL *\$variable*

■ Argumente

%variable oder *\$variable*

■ Hinweise

Beachten Sie, dass auch Systemvariable mit dieser Funktion gelöscht werden können.

■ Generierung durch HiCAD-Makroentwicklungssystem

Es ist keine automatische Generierung möglich.

■ Beispiel

```
REM   HICAD-Next VN:1201
START 1
$A:=Makrotechnik
$B:=technik
%I1:=IDX($A,$B)
OPEN c:Test
OUTPUT %I1
CLOSE
DEL $B
DEL %I1
END
```


3.1.6 DISTANZ

■ Funktion

Angabe einer Distanz, relativ zu einem identifizierten Punkt

■ Syntax

DISTANZ *realkonstante* / *arithm.ausdruck* / *RET* / # / *ZEI* / *END*

■ Argumente

RET Beim Makroablauf wird der angezeigte Defaultwert übernommen (festes Argument). Dieses Argument wird automatisch gesetzt, wenn während der Makroerstellung der angezeigte Defaultwert übernommen wurde.

Der Wert wird beim Makroablauf abgefragt (freies Argument).

ZEI Dieses Argument wird automatisch eingetragen, wenn während der Makroerstellung **z** eingegeben, d.h. eine Distanz aus der Zeichnung entnommen wurde. Der nächste Befehl im Makro lautet dann:

OPTION *nummer* 23

Dies entspricht der Aktivierung der Funktion **Distanz** im Menü **Information**. Die weitere Befehlsfolge hängt dann von *nummer* ab. Mit **ZEI** wird also in einem Makro der gleiche Ablauf simuliert, der durch die Eingabe von 'z' an entsprechender Stelle beim normalen Arbeiten mit HiCAD hervorgerufen wird.

■ Hinweise

Der DISTANZ-Befehl wird an Stellen generiert, an denen im Textfenster die Meldung *DISTANZ* erscheint. Dies ist insbesondere bei den Punktoptionen R, P, PX, PY der Fall, nach dem ein neuer Bezugspunkt definiert worden ist.

■ Generierung durch HiCAD-Makroentwicklungssystem

Der Befehl wird dort, wo die Eingabe einer Distanz erwartet wird, automatisch generiert.

■ Beispiel

Das angegebene Beispiel zeichnet eine Linie, wobei die y-Distanz aus der Zeichnung übernommen wird.

```
REM          BAUTEILE
START      2
REM          BAUTEILE
OPTION     1  2          1 = Linien
REM          LINIEN      2 = Streckenzug
OPTION     2  6
POINT     #
POINT    R # z
REM      X Distanz:
DISTANZ   45
REM      Y Distanz:
DISTANZ ZEI
REM          DISTANZEN      6 = Länge LE
OPTION     6  23
POINT     #
REM      Distanz negativ setzen (J/N) ?
ANTWORT   0
POINT     ESC
END
```

3.1.7 ECHO

■ Funktion

Anzeigen eines Benutzerführungstextes im Textfenster

■ Syntax

ECHO [*benutzerführungstext*]

■ Argumente

benutzerführungstext

Die Ausgabe dieses Strings erfolgt ohne einen Zeilenwechsel als Abschluss, d.h. die nächste Ausgabe im Textfenster wird direkt hinter *benutzerführungstext* gesetzt. Es lässt sich somit keine Maximallänge von *benutzerführungstext* angeben, da sie vom anschließend ausgegebenen Text abhängt. Sollte die Ausgabe einmal zu lang geraten sein, kann im Anschluss nach der Makroausführung der Bildschirm neu aufgebaut werden.

Sie haben auch die Möglichkeit einen Benutzerführungstext aus HiCAD anzugeben, der in der Datei MAKROTXT.DAT vorhanden ist. Dazu muss diese Datei in das Makro-Verzeichnis kopiert werden. Der Zugriff auf eine Zeile aus dieser Datei erfolgt durch Angabe des Strings \$100*n*

wobei *n* die Nummer einer Zeile in der Datei MAKROTXT.DAT ist, z.B. ECHO \$10021

In diesem Fall würde der Text aus Zeile 21 angezeigt.

■ Hinweise

Der ECHO-Befehl dient einmal zur Ausgabe von zusätzlicher Information zum normalen Benutzerführungstext. Dieser ausgegebene Text kann aber auch mit dem Befehl SAUS abgeschaltet werden. So haben Sie die Möglichkeit, Ihre eigenen Benutzerführungstexte zu verwenden.

■ Generierung durch HiCAD-Makroentwicklungssystem

Es ist keine automatische Generierung möglich.

Geklammerte ECHO- und VAR-Befehle

ECHO- und VAR-Befehle können auch geklammert verwendet werden. Damit wird erreicht, dass mehrere Benutzerführungstexte oder die Eingabe von Variablen während des Makroablaufes in einem PopUp-Menü angezeigt werden. Dabei ist folgendes zu beachten:

- Kommentare innerhalb der Klammer werden berücksichtigt.
- Es werden nur die ECHO-Befehle innerhalb der Klammer berücksichtigt, die vor der ersten VAR-Anweisung stehen. Spätere ECHO-Befehle werden ignoriert.
- Alle unmittelbar vor der Klammer stehenden ECHO-Befehle werden berücksichtigt.

Beispiel:

Das Beispielmakro bewirkt die Anzeige des abgebildeten PopUp-Menüs.

```
REM HICAD-NT : VN:07
REM HICAD-Next VN:1201
START 59
(
echo Variable A = Hoehe in cm
echo Variable B = Breite in cm
var %a A
var %b B
)
OPTION ESC
END
```

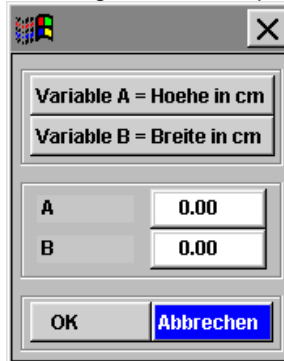


Abb. 8 Geklammerte VAR- und ECHO-Anweisungen

3.1.8 END

■ Funktion

Beendet die Ausführung eines Befehlsmakros

■ Syntax

END

■ Argumente

keine

■ Hinweise

Der END-Befehl ist im Allgemeinen der letzte Befehl eines Makros. Dem END-Befehl folgende Befehle werden ignoriert. Bei Angabe des Argumentes 1 ist nach Ende des Makros die Menüebene aktiv, in der das Makro aufgerufen wurde. Wird kein Argument angegeben, dann ist anschließend die Menüebene aktiv, in dem die Makroerstellung beendet wurde.

■ Generierung durch HiCAD-Makroentwicklungssystem

Der Befehl wird beim Beenden der Makroerstellung automatisch generiert.

■ Beispiel

Allgemeiner Rahmen eines Makros

```
REM          ZEICHENTEILE
START      1
END
```

3.1.9 FAA / FAE

- **Funktion**
Unterdrückung aller WAIT-Befehle in einem Makro / Einschalten der WAIT-Befehle
- **Syntax**
FAA
FAE
- **Argumente**
keine
- **Generierung durch HiCAD-Makroentwicklungssystem**
Es ist keine automatische Generierung möglich.
- **Hinweise**
Der Befehl FAA unterdrückt alle WAIT-Befehle eines Makros, auch solche, die WAUS ignorieren, so dass keine Meldungen auf dem Schirm erscheinen

3.1.10 FOR ... NEXT

■ Funktion

Schleife mit bestimmter Anzahl von Durchläufen

■ Syntax

FOR *schleifenvariable*:= *untergrenze* TO *obergrenze*

.

.

. (HCGS Befehle)

.

.

NEXT *schleifenvariable*

■ Argumente

schleifenvariable

Hierbei handelt es sich um eine Realvariable, die vorher nicht explizit definiert werden muss. Ihr wird zu Beginn der Schleife der Wert *untergrenze* zugewiesen. Trifft die Schleife auf den NEXT-Befehl, erhöht sich der Wert *schleifenvariable* um 1.

untergrenze, obergrenze

Diese Werte bestimmen die Anzahl der Schleifendurchläufe. Sie können ganzzahlige Konstante, Realvariable oder arithmetische Ausdrücke sein. Sollte das Ergebnis eines Ausdrucks keinen ganzzahligen Wert liefern, dann wird auf die nächstkleinere ganze Zahl abgerundet.

■ Hinweise

Es ist erlaubt, den Argumenten *schleifenvariable* und *obergrenze* innerhalb der Schleife einen Wert zuzuweisen, wobei diese Zuweisung natürlich einen Einfluss auf die Anzahl der Durchläufe hat. Weiterhin besteht die Möglichkeit, *untergrenze* und *obergrenze* durch Variablen oder arithmetische Ausdrücken zu belegen. Nach jedem NEXT-Befehl wird überprüft, ob der Wert von *schleifenvariable* kleiner oder gleich dem Wert des Argumentes *obergrenze* ist. Wenn ja, wird die Schleife erneut durchlaufen. Anderenfalls wird das Makro mit dem Befehl fortgesetzt, der dem NEXT-Befehl folgt.

FOR...NEXT Schleifen dürfen ineinander verschachtelt sein. Die Abarbeitung der Schleifen erfolgt von innen nach außen.

■ Generierung durch HiCAD-Makroentwicklungssystem

Es ist keine automatische Generierung möglich.

■ Beispiel

Das folgende Beispiel erzeugt 10 Linien in den in HiCAD verfügbaren Farben und setzt anschließend wieder die Defaultparameter.

```

REM          LINIEN
START      6
REM          LINIEN
OPTION     9 6          9 = Par. Vorgabe
REM          LINIEN-PARAM.
OPTION     7 18        7 = Param.sichern
OPTION     ESC
%a:=9
FOR %i:= 0 TO a
REM          LINIEN          9 = Par. Vorgabe
OPTION     9 6
REM          LINIEN-PARAM.   4 = Farbe
OPTION     4 18
REM          Farbe
INTEGER i
OPTION     ESC
REM          LINIEN          2 = Streckenzug
OPTION     2 6
POINT     A 100 100+10*i
POINT     R 100 0
POINT     ESC
NEXT i
REM          LINIEN          9 = Par. Vorgabe
OPTION     9 6
REM          LINIEN-PARAM.   8 = Param.zurück
OPTION     8 18
END

```


3.1.11 GOTO

■ Funktion

unbedingte Sprunganweisung, d.h. Fortsetzung des Makros an einer bestimmten Stelle

■ Syntax

GOTO *label*

label:HCGS-Befehl

■ Argumente

label

Dieser Wert gibt die Position an, mit der das Makro fortgesetzt werden soll. *label* selbst ist eine Markierung, die irgendwo innerhalb des Makros auftreten und von mehreren GOTO-Befehlen angesteuert werden kann. *label* muss eine Zahl zwischen 1 und 9999 sein! Der Zahl muss ein Doppelpunkt folgen und anschließend ein gültiger Makrobefehl. Ohne diese beiden Angaben ist *label* unwirksam, d.h. der entsprechende GOTO-Befehl wird nicht ausgeführt.

■ Hinweise

Der GOTO-Befehl ist eine unbedingte Sprunganweisung, d.h. er wird in jedem Fall ausgeführt. Der Sprung zu einer Makrozeile, die dem GOTO-Befehl vorausgeht, kann zu einer Endlosschleife führen. Dieser Fall macht nur Sinn in Verbindung mit dem IF-Befehl. Sollte es versehentlich zu einer Endlosschleife kommen, dann brechen Sie den Makrolauf ab. Wird gerade eine Benutzereingabe angefordert, geben Sie zusätzlich END ein.

■ Generierung durch HiCAD-Makroentwicklungssystem

Es ist keine automatische Generierung möglich.

■ Beispiel

Dieses Beispiel zeigt eine klassische Endlosschleife und ist daher nicht zur Nachahmung empfohlen.

```
REM          LINIEN
START      6
90:REM          LINIEN          1 = Isolierte Pkt
OPTION    1   6
POINT
REM Symbolnummer ( 0 oder 1-9999 ) (INT=grafisch):
INTEGER RET
POINT ESC
GOTO 90
END
```

siehe auch Beispiel zur IF-Anweisung

3.1.12 IF..ELSE..IFEND

■ Funktion

Bedingte Ausführung von Makrobefehlen

■ Syntax

IF *logischer Ausdruck* THEN

.

. (HCGS-Befehle)

ELSE

. (HCGS-Befehle)

.

IFEND

■ Argumente

logischer Ausdruck

Der angegebene Ausdruck muss den im Abschnitt 17.3 beschriebenen Vereinbarungen genügen.

■ Hinweise

Bei der Benutzung des IF-Befehls stehen zwei Möglichkeiten zur Verfügung:

- bedingte Befehlsausführung mit IF...GOTO
Liefert der angegebene Ausdruck das Ergebnis "wahr", dann wird der GOTO-Befehl ausgeführt. Ist das Ergebnis "falsch", wird der dem IF-Befehl folgende Befehl ausgeführt.
- IF..THEN.. ELSE ..IFEND
Hier können mehrere Makrobefehle zu einem Block zusammengefasst werden, deren Ausführung vom logischen Ausdruck des IF-Befehls abhängig ist. Ist der logische Ausdruck "wahr", wird der gesamte Block (hinter THEN bis zum ELSE, falls angegeben) ausgeführt. Ist der Ausdruck "falsch", dann wird der Block hinter dem ELSE ausgeführt. Wurde keine ELSE-Anweisung angegeben, dann wird mit dem Befehl nach dem IFEND fortgefahren. Eine Schachtelung von IF-IFEND Blöcken ist möglich, es ist aber darauf zu achten, dass jeder Block mit einem IFEND-Befehl abgeschlossen wird.
- Wertzuweisungen können nach einer einzeiligen IF-Anweisung erfolgen (ähnlich dem GOTO-Befehl). Beispiel: *if (x>100) %y:=10*

■ Generierung durch HiCAD-Makroentwicklungssystem

Es ist keine automatische Generierung möglich.

■ Beispiel

Im Folgenden wurde das Beispiel aus dem GOTO-Befehl um eine Abbruchbedingung erweitert.

```

REM          LINIEN
START      6
REM  BAUTEILE 1 = Linien
OPTION    1  2
90:REM          LINIEN          1 = Isolierte Pkt
OPTION    1  6
POINT #
REM Symbolnummer ( 0 oder 1-9999 ) (INT=grafisch):
INTEGER RET
POINT #
IF pesc GOTO 99
GOTO 90
99:END

```

Im nächsten Beispiel werden die Textparameter in Abhängigkeit vom Maßstab gesetzt. Dabei wird die Systemvariable @2 zur Abfrage der Maßstabszahl benutzt.

```

REM          BAUTEILE
START      2
REM          BAUTEILE          S6= Texte
OPTION    16  2
IF @2 <= 10 THEN
%h :=3.5
%s :=2
IFEND
IF @2 >10 THEN
%h :=2.5
%s :=4
IFEND
REM          TEXT          3 = Textparameter
OPTION    3  25
REM          PARAMETER          3 = Schrifthöhe
OPTION    3  26
REM  Schrifthöhe (mm):
REAL      H
REM          PARAMETER          2 = Schriftart
OPTION    2  26
REM  Schriftart:
INTEGER S
OPTION    ESC
END

```

3.1.13 IGNORE

■ **Funktion**

Überspringen von Befehlen

■ **Syntax**

IGNORE

■ **Argumente**

keine

■ **Hinweise**

Wird dieser Befehl in einem Makro gefunden, dann werden alle Befehle bis zum nächsten im Makro auftretenden OPTION-Befehl interaktiv abgefragt.

■ **Generierung durch HiCAD-Makroentwicklungssystem**

Es ist keine automatische Generierung möglich.

3.1.14 INTEGER

■ Funktion

Eingabeanforderung einer ganzen Zahl

■ Syntax

INTEGER *ganzzahlige konstante* / *arithm. ausdruck* / RET / # / ESC

■ Argumente

ganzzahlige konstante

Der hierfür angegebene Wert muss im jeweils erlaubten Wertebereich liegen, dies ist Funktionsabhängig.

arithmetischer Ausdruck

Sollte das Ergebnis des Ausdrucks keine ganze Zahl liefern, so wird auf die nächste ganze Zahl gerundet.

ESC

Das Argument ESC ist nur in einigen Fällen erlaubt, meistens um eine wiederholte Eingabe abzuschließen. Ist ESC eingegeben worden, obwohl es nicht erlaubt war, wird das Befehlsmakro mit einer Fehlermeldung unterbrochen. Dabei kann es auch vorkommen, dass die Ausführung des Makros anschließend abgebrochen wird.

■ Hinweise

Dieser Befehl fordert die Eingabe einer ganzen Zahl an, z.B. wenn im Textfenster eine Auswahl zu treffen ist. Der INTEGER-Befehl wird aber auch verwendet, wenn bei einer Parametereingabe nur ganze Zahlen erlaubt sind, z.B. Schichtnummer.

■ Generierung durch HiCAD-Makroentwicklungssystem

Wird während der Makroerstellung die Eingabe einer ganzen Zahl angefordert, dann wird der Befehl automatisch generiert.

■ Beispiele

Hier wird die Schichtnummer auf 2 gesetzt und die Auswahl der Farbe angefordert.

```

REM          LINIEN
START      6
REM          LINIEN
OPTION     9 6          9 = Par. Vorgabe
REM          LINIEN-PARAM.
OPTION     3 18        3 = Schichtnummer
REM        Schicht
INTEGER 2
REM          LINIEN-PARAM.
OPTION     4 18        4 = Farbe
REM        Farbe
OPTION     4 18
INTEGER #
OPTION    ESC
END

```

Hier werden Textparameter geändert. Die Eingabe der Schriftart wird mit RET bestätigt.

```

REM          BAUTEILE
START      2
REM          BAUTEILE
OPTION     16 2        S6= Texte
REM          TEXT
OPTION     3 25        3 = Textparameter
REM          PARAMETER
OPTION     2 26        2 = Schriftart
REM        Schriftart:
INTEGER RET
REM          PARAMETER
OPTION     3 26        3 = Schrifthöhe
REM        Schrifthöhe (mm):
REAL      RET
END

```

3.1.15 MAKRO

■ Funktion

Mit diesem Befehl wird aus dem aktuellen Makro heraus ein Unter-makro aufgerufen. Das aktuelle Makro wird dadurch beendet.

■ Syntax

MAKRO *dateiname*

■ Argumente

dateiname

Die Datei *dateiname* muss ein gültiges HCGS-Makro sein.

■ Hinweise

Der MAKRO-Befehl bewirkt den Sprung in das Makro *dateiname* und die Ausführung der in diesem Makro enthaltenen Befehle. Das Makro *dateiname* muss dabei ein eigenständiges Makro sein, das auch wieder einen MAKRO-Befehl enthalten kann.

■ Generierung durch HiCAD-Makroentwicklungssystem

Es ist keine automatische Generierung möglich.

■ Beispiel

In diesem Beispiel wird eine Konstruktion geladen, ohne die aktuelle zu speichern. Dann wird mit dem MAKRO-Befehl das Makro TEXTE aufgerufen, das abfragt, ob Texte dargestellt werden sollen oder nicht. Das aktuelle Makro wird dadurch beendet, d.h. alle dem MAKRO-Befehl folgenden Anweisungen werden nicht berücksichtigt. In diesem Fall wäre also der CALL-Befehl dem MAKRO-Befehl vorzuziehen.


```

REM   HICAD-Next VN:1201
REM           HiCAD
START   59
HNEXT
REM   HiCAD   3 = 3-D SZENE
OPTION  3 59
REM   3D - SZENE   2 = Laden
OPTION  2 101
STRING *
OPTION  ESC
REM   BERECHNUNG   2 = Eigene Verf.
OPTION  2 159
MAKRO C:TEXTE
REM   ZEICHENTEILE   3 = ZTL bearbeit.
OPTION  3 1
REM   BAUTEILE   8 = Hauptteil neu
OPTION  8 2
REM   NEUES TEIL   1 = Teil anlegen
OPTION  1 3
STRING RET
REM   BAUTEILE   1 = Linien
OPTION  1 2
REM   LINIEN   2 = Streckenzug
OPTION  2 6
POINT #
END

```

Makro TEXTE

```

REM   HICAD-Next VN:1201
START   59
REM ALT1
OPTION  21 0
REM   NEUZEICHNUNG   6 = Texte      E/A
OPTION  6 51
REM   Texte darstellen ?
ANTWORT #
OPTION  ESC
REM ALT1
OPTION  21 0
REM   NEUZEICHNUNG   1 = alle Teile
OPTION  1 51
END

```

3.1.16 MAUS/MEIN

■ Funktion

Aus-/Einschalten der Kantenzugmarkierung (3D)

Aus-/Einschalten der Markierungssymbole beim Identifizieren von Linienelementen (2D)

■ Syntax

MAUS
MEIN

■ Argumente

keine

■ Hinweise

Bei der Markierung von 3D-Kantenzügen werden die einzelnen Kanten normalerweise abwechselnd magenta/grün dargestellt. Mit dem Schalter MAUS/MEIN können Sie diese Darstellung aus-/einschalten. Der Defaultzustand ist MEIN.

■ Generierung durch HiCAD-Makroentwicklungssystem

Es ist keine automatische Generierung möglich.

■ Beispiel

```

REM      BEARBEITEN
START 106
REM      BEARBEITEN          2 = Kantenzüge
OPTION  2 106
REM      3D KANTENZUG       S5= Kntzug bearb.
OPTION  15 137
MAUS
REM      KANTENZÜGE        4 = Kntzug. mark.
OPTION  4 133
POINT   #
REM      Markierung beibehalten (J/N) ?
ANTWORT 1
POINT   ESC
END

```

3.1.17 MKDIR

■ Funktion

Anlegen eines Verzeichnisses

■ Syntax

MKDIR *verzeichnisname*

■ Argumente

Verzeichnisname

Hier ist der Name des neuen Verzeichnisses nach den üblichen Konventionen anzugeben.

■ Generierung durch HiCAD-Makroentwicklungssystem

Es ist keine automatische Generierung möglich.

■ Beispiel

In diesem Beispiel wird im HiCAD-Verzeichnis ein Unterverzeichnis *Szenen* angelegt und die Datei TEST.SZN aus dem ZTL-Verzeichnis in das neue Verzeichnis kopiert.

```
REM   HICAD-Next VN:1201
START 1
MKDIR \HICADNEXT\Szenenbak
COPY  \HICADNEXT\SZENEN\TEST.SZN HICADNEXT\SZENENBAK\TEST.SZN
END
```

3.1.18 **OPEN INPUTCLOSE** **OPEN OUTPUT CLOSE**

■ **Funktion**

Lesen/Schreiben von ASCII-Dateien

■ **Syntax**

Lesen aus Datei

```
OPEN dateiname
INPUT variable
INPUT variable
.
.
.
CLOSE
```

Schreiben in Datei

```
OPEN dateiname
OUTPUT variable
OUTPUT variable
.
.
.
CLOSE
```

■ **Argumente**

dateiname

Dieses Argument ist der Name der Datei aus der gelesen bzw. in die geschrieben wird. Beim Lesen muss die Datei vorhanden sein, beim Schreiben wird sie ggf. neu angelegt. Dem Dateinamen kann eine Pfadangabe vorangestellt werden. Die Namenserweiterung der Datei muss **.DAT** sein.

variable

Dieses Argument bestimmt, was aus der Datei gelesen bzw. in die Datei geschrieben wird. Dabei kann es sich um feste Zahlenwerte, um Texte, um den Inhalt von Variablen oder um Ergebnisse von Ausdrücken handeln.

■ Hinweise

Lesen

Mit dem OPEN-Befehl wird die angegebene Datei geöffnet. Aus dieser Datei können nun Zahlenwerte, Texte oder auch Punktkoordinaten eingelesen werden. Diese Werte, Texte oder Koordinaten werden über die INPUT-Anweisung jeweils einer Variablen zugeordnet. Mit der ersten INPUT-Anweisung wird die erste Zeile der Datei gelesen, mit der zweiten INPUT-Anweisung die zweite Zeile usw.. Der Zeileninhalt wird dabei jeweils der mit der entsprechenden INPUT-Anweisung angegebenen Variablen zugewiesen. Konnte einer Variablen kein Wert zugewiesen werden, wird die logische Variable VORHD auf FALSCH gesetzt, anderenfalls auf WAHR. Diese Variable kann speziell dazu verwendet werden das Ende einer ASCII-Datei zu erkennen.

Da der Variablentyp in einer INPUT-Anweisungen wechseln kann, muss die Datei natürlich entsprechend aufgebaut sein. Eine Besonderheit sind Punkt- und Linienvariablen. Sollen einer solchen Variablen über die INPUT-Anweisung Koordinaten zugewiesen werden, dann muss die entsprechende Zeile der ASCII-Datei beide Werte getrennt durch mindestens ein Leerzeichen enthalten.

Wenn alle benötigten Werte aus der Datei eingelesen sind, muss diese mit dem CLOSE-Befehl geschlossen werden. Eine geöffnete Datei wird automatisch geschlossen, wenn eine weitere Datei geöffnet wird oder die Makroausführung beendet wird.

Schreiben

Hier wird mit der OPEN-Anweisung zunächst die zu erzeugende Datei zum Schreiben geöffnet. Mit der OUTPUT-Anweisung können dann Inhalte von Variablen, Ergebnisse von Ausdrücken sowie feste Werte oder Texte in diese Datei geschrieben werden. Analog zum Lesen erfolgt auch das Schreiben zeilenweise. Eine Ausnahme sind auch hier wieder Punkt- und Linienvariable. Ihre Koordinaten werden in eine Zeile geschrieben, wobei die einzelnen Koordinatenwerte jeweils durch ein Leerzeichen getrennt sind. Auch hier muss die Datei mit CLOSE geschlossen werden.

Es ist nicht erlaubt, eine Datei gleichzeitig zum Lesen und Schreiben zu öffnen.

■ Generierung durch HiCAD-Makroentwicklungssystem

Es ist keine automatische Generierung möglich.

■ Beispiel

In diesem Beispiel werden die Stringvariable A1 und A2 aus der Datei BSP.DAT gelesen.

```
REM          ZEICHENTEILE
START      1
OPEN C:BSP
input $A1
input $a2
close
END
```

Datei BSP.DAT

```
Text1
Text2
Text3
Text4
```

Nach Ablauf des Makros haben die Variablen den Inhalt:

A1=Text1 und A2=Text2

In folgenden soll der Inhalt der Punktvariablen P0, P1 und P2 in die Datei BSP1.DAT geschrieben werden.

P0=100 100

P1=200.75 300

P2=400 570

```
REM          ZEICHENTEILE
START      1
OPEN C:BSP1
OUTPUT P0
OUTPUT P1
OUTPUT P2
CLOSE
END
```

Datei BSP1.DAT

```
100 100
200.75 300
400 570
```

3.1.19 OPTION

■ Funktion

Auswahl einer Menüfunktion aus der geraden aktiven Menüebene

■ Syntax

OPTION *menünummer*/ ESC *menüebene*

■ Argumente

menünummer

Aus der aktiven Menüebene wird die Funktion *menünummer* aktiviert.

Sollte *menünummer* eine Zahl von 21-26 sein, wird eine der Sonderfunktionen (ALT-Taste + 1 - 6) angesprochen. Es besteht folgende Zuordnung zwischen *menünummer* und Sonderfunktion (nur die ersten 6 Sonderfunktionen sind in einem Makro erlaubt):

<i>menünummer</i>	Sonderfunktion (Taste)
21	Zeichnen (ALT 1)
22	Zoom (ALT 2)
23	Information (ALT 3)
24	Bildschirmdruck (ALT 4)
25	Raster (ALT 5)
26	Variable (ALT 6)

In diesem Fall ist das 2. Argument *menüebene* 0.

ESC (END)

Mit dem Kontrollargument ESC (END) wird in der Menüstruktur eine Ebene hochgesprungen. Das Argument *menüebene* ist in diesem Fall inaktiv.

menüebene

Jedes Menü hat eine Menüebenen-Nummer. Im OPTION-Befehl gibt diese Nummer die Menüebene vor, aus der die Funktion *menünummer* gewählt wird. Dies befreit aber nicht davon, in der Auswahl hierarchisch vorzugehen. Ein Sprung innerhalb der Menüstruktur ist nicht möglich, z.B. von Menü 1 - **Zeichenteil** in Menü 6 - **Linien**, es muss erst das dazwischenliegende Menü 2 - **Bauteile** aufgerufen werden. *menünummer* wird automatisch bei Generierung des OPTION-Befehles gefunden.

■ Hinweise

Dies ist der mächtigste Befehl der Makrosprache HCGS. Er bewirkt die Aktivierung der Funktion *menünummer* aus der gerade aktiven Menüebene. Dies ist in der Regel der 2. Befehl gleich nach der START-Anweisung. **Die Nummer von *menüebene* muss bei diesem ersten OPTION-Befehl mit dem Argument der START-Anweisung übereinstimmen.** Befindet man sich nach einem Menüaufruf wiederum auf einer Menüebene, muss allerdings ein weiterer OPTION-Befehl folgen. Wird danach eine Eingabe über das Textfenster erwartet, muss ein Eingabebefehl aus den Gruppen **Skalare oder geometrische Eingabe** folgen. Mit ESC als Argument springt man innerhalb der Menüstruktur eine Stufe hoch. Befindet man sich schon in der obersten Stufe, ist der Befehl wirkungslos. Das Gleiche gilt für den Fall, dass *menüebene* nicht mit der Nummer des Menüs übereinstimmt, in der man sich gerade befindet.

■ Generierung durch HiCAD-Makroentwicklungssystem

Der Befehl wird automatisch generiert, wenn eine Menüfunktion ausgewählt wird.

■ Beispiel

Im Beispiel wird eine neue Konstruktion angelegt ohne die Konstruktionseigenschaften zu ändern. Anschließend wird ein neues 2D-Teil angelegt und die Funktion **Streckenzug** aktiviert.


```
REM HICAD-Next VN:1201
START 59
HNEXT
REM HiCAD 3 = 3-D SZENE
OPTION 3 59
REM 3D - SZENE 1 = Neu anlegen
OPTION 1 101
STRING #
REM Ändern ?
ANTWORT 0
OPTION ESC
REM BERECHNUNG 2 = Eigene Verf.
OPTION 2 159
REM ZEICHENTEILE 3 = ZTL bearbeit.
OPTION 3 1
REM BAUTEILE 8 = Hauptteil neu
OPTION 8 2
REM NEUES TEIL 1 = Teil anlegen
OPTION 1 3
STRING #
REM BAUTEILE 1 = Linien
OPTION 1 2
REM LINIEN 2 = Streckenzug
OPTION 2 6
Point #
END
```

3.1.20 PFD

■ Funktion

HiCAD-Pfad gemäß FILEGRUP.DAT ermitteln

■ Syntax

PFD *filegruppe*

PFD &:*dateierweiterung*

■ Argumente

filegruppe

Hier ist der Name einer Filegruppe anzugeben. Die Funktion liefert dann den Pfad, der dieser Filegruppe in der Datei FILEGRUP.DAT zugeordnet ist. Der ermittelte Pfad wird der Systemvariablen \$@9 zugewiesen.

dateierweiterung

Wird als Argument eine Dateinamenserweiterung angegeben, dann wird die Variable \$@9 mit dem Windows-Pfad belegt, der für diesen Dateityp verwendet wird.

■ Generierung durch HiCAD-Makroentwicklungssystem

Es ist keine automatische Generierung möglich.

■ Beispiel

Hier wird der zur Filegruppe c: gehörende Pfad ermittelt und dann der Variablen \$A zugewiesen.

```
REM   HiCAD-Next VN:1201
START 1
PFD c:
$A:=$@9
END
```

Hier wird der Pfad, in dem die Szenendateien gespeichert werden, der Variablen \$A zugewiesen.

```
REM   HiCAD-Next VN:1203
START 1
PFD &: .SZA
$A:=$@9
END
```

3.1.21 POINT

■ Funktion

Eingabe einer Punktoption, wenn gefordert

■ Syntax

POINT *punktoption* / # / ESC / END/ LLL / LLA

■ Argumente

punktoption

Nur ein Teil der in HiCAD verfügbaren Punktoptionen ist uneingeschränkt als Argument erlaubt und zwar diejenigen, die einen Punkt ohne eine weitere Punkt-/Linienbestimmung eindeutig festlegen. Dies sind die Optionen:

A, K, R, P, N, L

Sind zur eindeutigen Festlegung Koordinatenwerte erforderlich, so sind diese als weitere Argumente, durch Leerzeichen getrennt, hinter *punktoption* anzufügen, z.B.

POINT A 100.0 155.55

Als Argumente sind hier neben Konstanten auch arithmetische Ausdrücke zugelassen.

An dieser Stelle sei darauf hingewiesen, dass in einem Makro die Option K anstelle der Option A bevorzugt eingesetzt werden sollte.

Die restlichen Punktoptionen sind in einem Makro nur in Verbindung mit den Punktvariablen P0 bis P9 oder den Linienelementvariablen L0 bis L9 erlaubt, die vorher mit der ALT-Funktion **Variablenspeicher** oder der VAR-Anweisung festgelegt wurden. Werden diese Punktoptionen

I, S, S2, M, M2, F, T, O

während der Generierung mit dem HiCAD-Makroentwicklungssystem verwendet, wird als Argument des POINT Befehls das freie Argument eingetragen.

LLL Diese Argumente sind nur in Zusammenhang mit dem
LLA POINT-Befehl erlaubt. Sie entsprechen der Eingabe letz-
te

Linie löschen (LLL) und letzte Löschung aufheben (LLA) und werden bei Verwendung dieser Befehle während der Makroerstellung automatisch als Argumente eingetragen.

■ Hinweis

Die Punktoption \$X (Fangradius) wird bei der Makroerstellung nicht mitprotokolliert. Soll z.B. zur Anforderung einer Punktangabe eine dafür spezifische Mitteilung ausgegeben werden, so muss diese vor dem entsprechenden Befehl zur Punkteingabe im Makro stehen.

■ Generierung durch HiCAD-Makroentwicklungssystem

Der Befehl wird automatisch generiert, wenn die Eingabe eines Punktes angefordert wird.

■ Beispiel

In diesem Beispiel werden zur Bestimmung eines Folgepunktes bei der Funktion Streckenzug die Punktvariablen P0 und P1 definiert und benutzt.

```

REM          LINIEN
START      6
REM          LINIEN          2 = Streckenzug
OPTION     2  6
POINT     #
POINT     A 0 100
POINT     P 45 50
POINT     ESC
REM ALT6
OPTION     26  0
REM       Variablenspeicher löschen (J/N) ?
ANTWORT   0
REM       Name der numerischen Variablen :
STRING    ESC
REM       Name der Text-Variablen :
STRING    ESC
REM       Punkt :
INTEGER   0
POINT     #
REM       Punkt :
INTEGER   1
POINT     #
REM       Punkt :
INTEGER   ESC
REM       Linienelement :
INTEGER   ESC
REM          LINIEN          2 = Streckenzug
OPTION     2  6
POINT     p0
POINT     p1
POINT     ESC
END

```

3.1.22 REAL

■ Funktion

Eingabeanforderung einer reellen Zahl

■ Syntax

REAL *reelle Konstante* / *arithm.Ausdruck* / RET / #

■ Argumente

RET Beim Makroablauf wird der angezeigte Defaultwert übernommen (festes Argument).

Der Wert wird beim Makroablauf abgefragt (freies Argument).

■ Hinweise

Der REAL-Befehl ist im Wesentlichen analog zum INTEGER-Befehl, nur dass hier anstelle einer ganzen Zahl eine reelle Zahl eingegeben wird.

■ Generierung durch HiCAD-Makroentwicklungssystem

Wenn die Eingabe einer reellen Zahl angefordert wird.

■ Beispiele

Hier wird der Zoomfaktor geändert.

```
REM HICAD-Next VN:1201
START 59
REM ALT2
OPTION 22 0
REM ANSICHT 1 = Zoomfaktor
OPTION 1 52
REM Zoomfaktor =
REAL #
POINT A 50 50
END
```

3.1.23 REM

■ Funktion

Einbau von Kommentarzeilen

■ Syntax

REM [*zeichenreihe*]

■ Argumente

zeichenreihe

Die Länge des Strings *zeichenreihe* ist auf eine Bildschirmzeile beschränkt, in der Regel also 80 Zeichen. Dies ist der einzige HCGS-Befehl bei dem das Argument nicht durch ein Leerzeichen vom Befehl getrennt sein muss.

■ Hinweise

Der REM-Befehl hat keinerlei Auswirkung auf die Makro-Ausführung. Er dient einzig und allein zur Dokumentation und kann an jeder beliebigen Stelle innerhalb des Makros eingefügt werden.

■ Generierung durch HiCAD-Makroentwicklungssystem

Es wird vor jede automatisch generierte Anweisung eine Kommentarzeile gesetzt (Ausnahme: POINT). Im Fall des OPTION-Befehls besteht der Kommentar aus zwei Teilen. Der erste Teil enthält die Menübezeichnung, gefolgt von der gewählten Menüfunktion. Bei allen anderen Befehlen wird der Benutzerführungstext aus dem Textfenster übernommen.

■ Beispiel

Hier wurde das Beispiel zum Befehl OPTION erweitert und zwar um die Funktion **Farbe setzen**. Die Linienfarbe wird hier auf grün gesetzt.

```
REM   HICAD-Next VN:1201
START 59
HNEXT
REM   HICAD 3 = 3-D SZENE
OPTION 3 59
REM   3D - SZENE 1 = Neu anlegen
OPTION 1 101
STRING #
REM   Ändern ?
ANTWORT 0
OPTION ESC
REM   BERECHNUNG 2 = Eigene Verf.
OPTION 2 159
REM   ZEICHENTEILE 3 = ZTL bearbeit.
OPTION 3 1
REM   BAUTEILE 8 = Hauptteil neu
OPTION 8 2
REM   NEUES TEIL 1 = Teil anlegen
OPTION 1 3
STRING #
REM   BAUTEILE 1 = Linien
OPTION 1 2
REM   LINIEN 9 = Par. Vorgabe
OPTION 9 6
REM   LINIEN-PARAM. 4 = Farbe
OPTION 4 18
REM   Farbe grün
INTEGER 1
OPTION ESC
REM   LINIEN 2 = Streckenzug
OPTION 2 6
Point #
END
```

3.1.24 REPEAT

■ Funktion

Wiederholtes Durchlaufen einer Schleife mit HCGS-Befehlen, solange die Schleifenbedingung nicht erfüllt ist. Die Bedingung wird am Ende der Schleife getestet und somit mindestens einmal durchlaufen.

■ Syntax

REPEAT

.

. HCGS-Befehle

.

UNTIL *logischer Ausdruck*

■ Argumente

logischer Ausdruck

Im Gegensatz zu den anderen HCGS-Befehlen folgt das Argument hier nicht unmittelbar dem Befehl, sondern erst im Zusatzbefehl, der das Schleifenende markiert. Ansonsten gelten die Bedingungen des WHILE-Befehls.

■ Hinweise

Der REPEAT-Befehl steht im engen Zusammenhang mit dem WHILE-Befehl. Im Unterschied zum WHILE-Befehl wird die Schleifenbedingung jedoch hier erst am Ende der Schleife abgefragt. Dieser Befehl findet oft Verwendung bei Funktionen, die eine mehrfache Punkteingabe ermöglichen.

■ Generierung durch HiCAD-Makroentwicklungssystem

Es ist keine automatische Generierung möglich.

■ Beispiele

Dieses Makro entspricht im Wesentlichen dem HiCAD-Makro, das der Funktion **Streckenzug** hinterlegt ist.

```

REM          LINIEN
START      6
REM          LINIEN          2 = Streckenzug
OPTION     2  6
REPEAT
POINT #
UNTIL PESC
END

```


3.1.25 SAUS / SEIN

■ Funktion

Aus-/Einschalten der Benutzerführungstexte im Textfenster und der Teilerrechnungen

■ Syntax

SAUS
SEIN

■ Argumente

keine

■ Hinweise

- Bei der Ausführung eines Makros werden normalerweise alle Benutzerführungstexte der im Makro benutzten Funktionen im Textfenster angezeigt. Dies führt, speziell bei langen Makros, zu erheblichen Verzögerungen bei der Ausführung. Mit SAUS lässt sich die Ausgabe dieser Texte unterdrücken, mit SEIN wieder einschalten. Das Einschalten der Texte ist beispielsweise notwendig, damit beim Arbeiten mit dem freien Argument die entsprechende Eingabeanforderung angezeigt wird. Die Ausgabe von ECHO- und WAIT-Befehlen wird hierdurch nicht beeinflusst. Ein SEIN am Ende eines Makros ist nicht erforderlich, da die Textausgabe dann automatisch wieder eingeschaltet wird.

Der Aufruf eines Untermakros mit der Anweisung CALL setzt automatisch SAUS auf SEIN zurück.

- Wenn in HiCAD eine Bearbeitungsebene aktiv ist und man eine Ausnehmung anlegt, dann wird bei der Eingabe der Ausnehmungstiefe vorher in die Dynamik gesprungen, d.h. man kann mit der Maus dynamisch die Tiefe der Ausnehmung festlegen. Nach Abtippen des Cursors (RET) kann der angezeigte Wert übernommen oder geändert werden.

Beim Setzen von Ausnehmungen über ein Makro, springt HiCAD während des Makroablaufes ebenfalls in die Dynamik, es wird aber auf jeden Fall der im Makro eingetragene Wert als Tiefe übernommen. Der Sprung in die Dynamik kann durch Benutzen des Makrobefehls SAUS verhindert werden. Dies funktioniert auch beim Erstellen eines allgemeinen Zylinders.

■ Generierung durch HiCAD-Makroentwicklungssystem

Es ist keine automatische Generierung möglich.

■ Beispiele

Dieses Makro entspricht dem Beispiel des REPEAT-Befehls, hier wird jedoch die Ausgabe der Benutzerführungstexte ausgeschaltet.

```
REM          LINIEN
START      6
SAUS
REM          LINIEN          2 = Streckenzug
OPTION     2  6
REPEAT
POINT #
UNTIL PESC
SEIN
END
```

3.1.26 START

■ Funktion

Makroanfang

■ Syntax

START *menüebene*

■ Argumente

menüebene

Dieses Argument gehört zur Gruppe von Nummern, wobei in diesem Fall nur die folgenden Menünummern und -ebenen als Startpunkt eines Befehlsmakros erlaubt sind.

59	Hauptmenü		
1	Zeichenteil	101	Szene (3D)
2	Bauteile (2D)	102	Teile (3D)
6	Linien	106	Bearbeiten (3D)

■ Hinweise

Die durch den START-Befehl generierte Startprozedur erlaubt es, ein Makro von jeder beliebigen Stelle des hierarchisch strukturierten HiCAD-Menübaumes aufzurufen. Durch die Startprozedur wird automatisch auf die Menüebene gesprungen, die beim Erstellen des Makros aktiv war. Der START-Befehl darf in einem Makro nur einmal enthalten sein und muss als erster Befehl eines Makros auftreten (Ausnahme: ein REM-Befehl darf vorher auftreten).

■ Generierung durch HiCAD-Makroentwicklungssystem

Der Befehl wird bei Aufruf der Funktion **Makro erstellen** automatisch generiert.

■ Beispiele

REM		ZEICHENTEIL
START	1	
.		

REM		BAUTEILE
START	2	
.		

REM		LINIEN
START	6	
.		

3.1.27 STRING

■ Funktion

Eingabeanforderung einer Zeichenreihe

■ Syntax

STRING *zeichenreihe* / *variable* / RET / # / INT/ ESC

■ Argumente

zeichenreihe

Die Länge des Strings hängt von der Eingabe ab, die an dieser Stelle erwartet wird. Beispielsweise sind es bei Text für die Textbeschriftung maximal 60 Zeichen, bei Datei- oder Teilnamen dagegen nur 8. Sollte der String zu lang sein, wird er automatisch auf die Maximallänge gekürzt.

variable

Es sind sowohl Stringvariablen als auch Realvariablen erlaubt. Es ist aber nur eine Variable als Argument erlaubt. Realvariablen, mit einem %-Zeichen eingeleitet, werden automatisch in einen String umgewandelt. Die Formatvariable \$@ wird dabei ausgewertet.

RET Beim Makroablauf wird der angezeigte Defaultwert übernommen (festes Argument).

Der Wert wird beim Makroablauf abgefragt (freies Argument).

INT Dieses Argument kann z.B. benutzt werden, um Figuren durch Identifikation von Linien zu bestimmen.

ESC hat einen Abbruch der Texteingabe zur Folge

■ Generierung durch HiCAD-Makroentwicklungssystem

Der Befehl wird automatisch generiert, wenn die Eingabe eines Textes angefordert wird.

■ Beispiel

In diesem Beispiel werden verschiedene Argumente des STRING-Befehls anhand der Funktion **Text einfügen** demonstriert.

```
REM          BAUTEILE
START      2
$A:=variable
%A :=5.25
REM          BAUTEILE          S6= Texte
OPTION    16      2
REM          TEXT              1 = Texte einfüg.
OPTION    1      25
POINT     #
REM       Beschriftungswinkel:
WINKEL   RET
REM       Text:
STRING Konstante
POINT     #
REM       Beschriftungswinkel:
WINKEL   RET
REM       Text:
STRING $A
POINT     #
REM       Text:
STRING RET
POINT     ESC
END
```

3.1.28 SZAUS / SZEIN

■ **Funktion**

Aus-/Einschalten des Statuszeilen-Update

■ **Syntax**

SZAUS

SZEIN

■ **Argumente**

keine

■ **Generierung durch HiCAD-Makroentwicklungssystem**

Es ist keine automatische Generierung möglich.

■ **Hinweis**

Die Informationen im Statusfenster des Bildschirmmenüs werden auch während eines Makrolaufes aktualisiert, z.B. beim Wechsel des aktiven Teils. Diese Aktualisierung kann während eines Makrolaufes evtl. störend sein. Durch Verwendung der Funktion SZAUS kann die Aktualisierung vorübergehend unterbunden werden.

3.1.29 UDA / UDE

■ Funktion

3D-Undo-Sicherung ein-/ausschalten

■ Syntax

UDE

UDA

■ Argumente

Keine

■ Hinweise

Für die HiCAD-Undo-Funktion werden die aufgerufen Funktionen intern gesichert. Diese Sicherung können Sie mit der UDA-Funktion während des Makrolaufes ausschalten bzw. mit der UDE-Funktion wieder einschalten.

■ Generierung durch HiCAD-Makroentwicklungssystem

Es ist keine automatische Generierung möglich.

3.1.30 VAI

■ Funktion

Benutzereingabe für den Wert einer Integer-Variablen während des Makroablaufes

■ Syntax

VAI % *variable* [*benutzerführungstext*]

■ Argumente

% *variable*

Hiermit wird einer Integervariablen ein Wert zugewiesen. Die Eingabe ist in der Regel eine ganze Zahl, kann aber auch wiederum eine Realvariable oder ein arithmetischer Ausdruck sein, wobei aber immer auf die nächste ganze Zahl gerundet wird.

benutzerführungstext

benutzerführungstext ist eine Zeichenreihe, deren Länge 50 Zeichen nicht überschreiten sollte. Die Angabe ist optional, aber empfehlenswert, da der Text dem Anwender signalisiert, dass eine Eingabe verlangt wird.

■ Generierung durch HiCAD-Makroentwicklungssystem

Es ist keine automatische Generierung möglich.

■ Beispiel

In diesem Beispiel wird eine beliebige Anzahl von Kreisen mit einem bestimmten Radius erzeugt. Die Anzahl und der Radius werden vom Makro abgefragt.

```

REM          LINIEN
START      6
VAI %a Anzahl der Kreise
VAR %r Radius der Kreise
FOR %i:= 1 TO a
REM          LINIEN                      5 = Kreis/-bogen
OPTION     5 6
REM          KREISBOGEN                  5 = MP- Radius
OPTION     5 11
POINT     #
REM      Radius (INT für Punkt auf Kreis) :
DISTANZ  r
POINT    ESC
NEXT I
END

```


3.1.31 VAR

■ Funktion

Benutzereingabe für den Wert einer Variablen während des Makroablaufes

■ Syntax

VAR % / \$ *variable* / Ln / Pn / [*benutzerführungstext*]

■ Argumente

% variable

Hiermit wird einer Realvariablen ein Wert zugewiesen. Die Eingabe ist in der Regel eine reelle Zahl, kann aber auch wiederum eine Realvariable oder ein arithmetischer Ausdruck sein.

\$variable

Hiermit wird einer Stringvariablen eine Zeichenreihe zugewiesen. Die Länge der Zeichenreihe hängt wiederum von ihrer Verwendung ab (vgl. STRING-Befehl).

Pn, Ln

Hiermit werden Punkt- und Linienvariablen durch Identifizierung zugewiesen.

benutzerführungstext

benutzerführungstext ist eine Zeichenreihe, deren Länge 50 Zeichen nicht überschreiten sollte. Die Angabe ist optional, aber empfehlenswert, da der Text dem Anwender signalisiert, dass eine Eingabe verlangt wird.

■ Hinweise

Der VAR-Befehl vereinfacht die Wertzuweisung an Variablen entschieden gegenüber der Eingabe über die Sonderfunktion **Variable** (Taste STRG+6), da hier gezielt eine einzige Variable abgefragt werden kann. Der Ablauf des Makros wird an dieser Stelle angehalten um die Benutzereingabe zu ermöglichen. Sollte die betreffende Variable schon definiert sein und einen gültigen Wert besitzen, so wird dieser zur Übernahme angeboten. Durch Eingabe von RETURN wird der Wert übernommen.

■ Generierung durch HiCAD-Makroentwicklungssystem

Es ist keine automatische Generierung möglich.

Lesen Sie auch die Hinweise zu geklammerten ECHO- und VAR-Befehlen (siehe 3.1.7).

■ Beispiel 1

```

REM          LINIEN
START      6
VAR %a Anzahl der Kreise
VAR %r Radius der Kreise
FOR %i:= 1 TO a
REM          LINIEN                    5 = Kreis/-bogen
OPTION      5 6
REM          KREISBOGEN                5 = MP- Radius
OPTION      5 11
POINT      #
REM          Radius (INT für Punkt auf Kreis) :
DISTANZ    r
POINT      ESC
NEXT I
END

```

■ Beispiel 2

```

REM          Punkt- und Linienvariablen
START      6
VAR P0 1.Punkt eingeben
VAR P1 2.Punkt eingeben
VAR L0 Linie identifizieren
REM          LINIEN                    5 = Kreis/-bogen
OPTION      5 6
REM          KREISBOGEN                7 = P-P-LE
OPTION      7 11
POINT      P0
POINT      P1
POINT      L0
REM          Kreis O.K. (J/N) ?
ANTWORT    1
POINT      ESC
END

```

3.1.32 Variablenzuweisung

■ Funktion

Wertzuweisung an eine Variable innerhalb des Makros

■ Syntax

%variable:= ausdruck Numerische Variable
\$variable:= ausdruck Stringvariable

■ Argumente

ausdruck

ausdruck kann sowohl eine Konstante, eine Variable oder ein arithmetischer Ausdruck sein, ist aber abhängig von der Art der Variable, der der Wert zugewiesen wird. Hier ist zwischen numerischen Variablen (%) und Stringvariablen (\$) zu unterscheiden

numerische Variablen

Die Wertzuweisung an eine numerische Variable wird mit einem %-Zeichen eingeleitet. Als *ausdruck* sind hier ganzzahlige und reelle Konstanten, numerische Variablen und arithmetische Ausdrücke erlaubt. Nicht erlaubt ist die Zuweisung einer Stringvariablen, auch wenn darin nur eine reelle Zahl steht.

Stringvariablen

Die Wertzuweisung an eine Stringvariable wird mit einem \$-Zeichen eingeleitet. Für *ausdruck* kann zunächst eine Zeichenreihe stehen, die jedoch nicht in Hochkommata (" ") eingeschlossen sein darf, da diese mit in die Variable übernommen werden.

Neben einer Stringvariablen wird auch die Verkettung von zwei, aber nur zwei, Stringvariablen akzeptiert in der Form \$a+\$b. Es ist darauf zu achten, dass vor und hinter dem +- Zeichen keine Leerstellen eingefügt werden. Anstelle der ersten Stringvariablen, und nur da, ist auch eine Zeichenreihe erlaubt, diese muss hier dann allerdings in Hochkommata eingeschlossen sein.

Letzte Möglichkeit für *ausdruck* ist die Angabe einer numerischen Variable. Sie wird durch das %-Zeichen gekennzeichnet und dann automatisch in einen String umgewandelt. Wie auch beim STRING-Befehl wird bei dieser Umwandlung die Formatvariable \$@ ausgewertet.

■ Hinweise

Mit dieser Art der Wertzuweisung haben Sie die Möglichkeit, ein Makro programmieretechnisch zu verändern, ohne die Sonderfunktion **Variab-
le** (Taste STRG+6) aufrufen zu müssen. Allen benutzerspezifischen Variablen dürfen hier Werte zugewiesen werden. Es ist zwar auch möglich, die Systemvariablen zu verändern, diese sollte aber in erster Linie dem System überlassen werden.

■ Generierung durch HiCAD-Makroentwicklungssystem

Es ist keine automatische Generierung möglich.

■ Beispiel 1

```

REM          LINIEN
START      6
$@:=(29hFlaeche der geschl. Kontur:, f10,2, 5h mm^^2)
$a := %z0
OPTION    ESC
REM          BAUTEILE                S6= Texte
OPTION    16      2
REM          TEXT                    1 = Texte einfüg.
OPTION    1      25
POINT     #
REM          Beschriftungswinkel:
WINKEL    RET
REM          Text:
STRING    $a
POINT     ESC
OPTION    ESC
END

```

```

REM          LINIEN
START      6
VAR %a Anzahl der Kreise
%i:=0
WHILE i < a
REM          LINIEN                5 = Kreis/-bogen
OPTION    5      6
REM          KREISBOGEN            5 = MP- Radius
OPTION    5      11
POINT     #
REM          Radius (INT für Punkt auf Kreis) :
DISTANZ   #
POINT     ESC
%i:=i+1
WHEND
END

```

3.1.33 WAIT

■ Funktion

Anzeigen eines Benutzerführungstextes im Textfenster und Unterbrechung der Makroausführung. Die Ausführung wird fortgesetzt mit einem RETURN.

■ Syntax

WAIT [*benutzerführungstext*]

■ Argumente

benutzerführungstext

Dieses Argument ist ähnlich dem Argument des Befehls ECHO. An dieser Stelle kann *benutzerführungstext* eine maximale Länge von 66 Zeichen haben, da für die Ausgabe eine ganze Zeile im Textfenster reserviert ist. Sollte die Ausgabe trotzdem einmal länger geraten sein als 66 Zeichen, kann aber auch hier mit einem EGA-Reset (unter Sonderfunktion **ZOOM**) der Bildschirm neu aufgebaut werden.

Sie haben auch die Möglichkeit einen Benutzerführungstext aus HiCAD anzugeben, der in der Datei MAKROTXT.DAT vorhanden ist. Dazu muss diese Datei in das Makro-Verzeichnis kopiert werden. Auf eine Zeile aus dieser Datei wird durch Angabe des Strings **\$100n** zugegriffen, wobei n die Nummer einer Zeile in der Datei MAKROTXT.DAT ist, z.B. **WAIT \$10021**

In diesem Fall würde der Text aus Zeile 21 angezeigt.

■ Hinweise

Der WAIT-Befehl ist dem ECHO-Befehl verwandt und dient ebenfalls zur Ausgabe zusätzlicher Informationen im Textfenster während des Makroablaufes. Zusätzlich zum ECHO-Befehl wird hier nach Ausgabe von *benutzerführungstext* der Ablauf angehalten. Dies ist durch das Anzeigen eines Sterns (*) hinter der Ausgabe gekennzeichnet. Nach Betätigen der RETURN-Taste wird das Makro fortgesetzt. Andere Aktionen sind zu diesem Zeitpunkt nicht möglich.

■ Generierung durch HiCAD-Makroentwicklungssystem

Es ist keine automatische Generierung möglich.

■ Beispiel

```

REM          ZEICHENTEILE
START      1
WAIT Zeichnen einer neuen Strecke
REM          ZEICHENTEILE          3 = ZTL bearbeit.
OPTION     3  1
REM          BAUTEILE              1 = Linien
OPTION     1  2
REM          LINIEN                2 = Streckenzug
OPTION     2  6
POINT     #
POINT     #
POINT     ESC
END

```

3.1.34 WARTE

■ Funktion

Anhalten des Makros für eine bestimmte Anzahl von Sekunden.

■ Syntax

WARTE *n*

■ Argumente

n

Dieser Wert bestimmt die Dauer (in Sekunden), für die das Makro angehalten werden soll.

■ Generierung durch HiCAD-Makroentwicklungssystem

Es ist keine automatische Generierung möglich.

■ Beispiel

```

REM          LINIEN
START      6
WARTE 20
REM          LINIEN          9 = Par. Vorgabe
OPTION     9  6
REM          LINIEN-PARAM.    6 = Defaultparam.
OPTION     6  18
OPTION     ESC
END

```

3.1.35 WAUS/WEIN

■ Funktion

Aus-/Einschalten des Wait-Status

■ Syntax

WAUS
WEIN

■ Argumente

keine

■ Hinweise

Im Gegensatz zu SAUS/SEIN ist hier der ausgeschaltete Zustand (WAUS) die Voreinstellung. Das bedeutet, dass bei rein informativen Ausgaben im Textfenster, bei denen lediglich mit einem RETURN geantwortet werden kann, das Makro nicht angehalten wird. Gekennzeichnet sind solche Ausgaben durch einen Stern (*) hinter dem Ausgabertext, z.B. bei der Funktion [Defaultparameter setzen](#) in verschiedenen Funktionen. Soll an diesen Stellen nun die Prozedur anhalten, muss das Makro vorher ein *WEIN* enthalten.

■ Generierung durch HiCAD-Makroentwicklungssystem

Es ist keine automatische Generierung möglich.

■ Beispiel

```

REM          LINIEN
START      6
WEIN
REM          LINIEN          9 = Par. Vorgabe
OPTION    9  6
REM          LINIEN-PARAM.   6 = Defaultparam.
OPTION    6  18
OPTION   ESC
END

```

Die Ausgabe der Fehlermeldung bei erfolgloser Datensuche in der Datenbank kann über die Kommandos [WEINWAUS](#) beeinflusst werden.

3.1.36 WERT

■ Funktion

prüft, ob eine bestimmte numerische Variable oder Textvariable vorhanden ist

■ Syntax

WERT *%variable*

WERT *\$variable*

■ Argumente

%variable

\$variable

Es wird geprüft, ob diese Variable definiert ist. Ist dies der Fall, dann wird die logische Variable VORHD auf WAHR gesetzt, anderenfalls auf FALSCH.

3.1.37 WHILE...WHEND

■ Funktion

Wiederholtes Durchlaufen einer Schleife solange die Schleifenbedingung erfüllt ist. Die Bedingung wird zu Beginn der Schleife getestet.

■ Syntax

WHILE *logischer Ausdruck*

```
.  
.   HCGS-Befehl  
.    
WHEND
```

■ Argumente

logischer Ausdruck

Jeweils zu Beginn des Schleifendurchlaufes wird überprüft, ob das Ergebnis des angegebenen Ausdrucks "wahr" ist. Ist er "falsch" wird mit dem HCGS-Befehl fortgefahren, der dem WHEND-Befehl folgt. Es ist darauf zu achten, dass alle in *logischer Ausdruck* verwendeten Variablen vorher definiert werden.

■ Hinweise

Bei einer WHILE-Schleife muss zu Beginn die Anzahl der Durchläufe nicht feststehen. Sie findet daher Verwendung in Fällen, bei denen das Beenden der Schleife durch 'nicht vorhersehbare Ereignisse' notwendig wird. Typisch hierfür ist eine Benutzereingabe während des Makroablaufes. Dadurch lässt sich die WHILE-Schleife allgemeiner gestalten als die FOR-Schleife.

■ Generierung durch HiCAD-Makroentwicklungssystem

Es ist keine automatische Generierung möglich.

■ Beispiel

Dieses Makro löscht alle in der aktuellen Konstruktion vorhandenen 3D-Sechskantschrauben nach DIN 931 - M 42 X 130 - ST, sofern sie nicht umbenannt wurden. Dabei wird als Schleifenbedingung anstelle eines logischen Ausdrucks die Systemvariable VORHD benutzt. Dadurch wird die Schleife sofort beendet, wenn kein Teil mit dem Namen DIN 931 - M 42 X 130 - ST vorhanden ist.

```
REM   HICAD-Next VN:1201
START 59
REM   HiCAD 3 = 3-D SZENE
OPTION 3 59
REM   3D - SZENE 3 = Bearbeiten
OPTION 3 101
REM   TEILE 10= Aktivieren
OPTION 10 102
STRING DIN 931 - M 42 X 130 - ST
WHILE vorhd
REM   TEILE 7 = Löschen
OPTION 7 102
REM   O.K. ?
ANTWORT 1
REM   TEILE 10= Aktivieren
OPTION 10 102
STRING DIN 931 - M 42 X 130 - ST
WHEND
ECHO Alle Schrauben nach DIN561 gelöscht oder keine vorhanden!!
END
```

3.1.38 WINKEL

■ Funktion

Angabe eines Winkels, relativ zu einer Bezugsgeraden

■ Syntax

WINKEL *reelle Konstante / arithmetischer Ausdruck / # / ZEI / ESC*

■ Argumente

RET Beim Makroablauf wird der angezeigte Defaultwert übernommen (festes Argument).

Der Wert wird beim Makroablauf abgefragt (freies Argument).

ZEI Dieses Argument wird automatisch eingetragen, wenn während der Makroerstellung die Eingabe **z** erfolgt ist, d.h. ein Winkel aus der Zeichnung entnommen wurde. Der nächste Befehl im Makro lautet dann:

OPTION *nummer 24*

Dies entspricht der Aktivierung der Funktion **Winkel** im Menü **Information**. Die weitere Befehlsfolge hängt dann von *nummer* ab. Mit ZEI wird also in einem Makro der gleiche Ablauf simuliert, der durch die Eingabe von 'z' an entsprechender Stelle beim Arbeiten mit HiCAD hervorgerufen wird.

■ Hinweise

Dieser Befehl ist analog dem DISTANZ-Befehl und wird an Stellen benötigt, die eine Winkeleingabe erfordern (Punktoptionen P, PX, PY, WX, WY)

■ Generierung durch HiCAD-Makroentwicklungssystem

Der Befehl wird automatisch generiert, wenn die Eingabe eines Winkels angefordert wird.

■ Beispiel

```
REM          LINIEN
START      6
REM          LINIEN          2 = Streckenzug
OPTION    2  6
POINT     #
POINT     P Z
WINKEL  ZEI
REM          WINKEL          3 = Wink.LE-Achse
OPTION    3  24
POINT     #
REM          Plus rechter Winkel (4)  - Minus rechter Winkel (5)
INTEGER  RET
REM          Distanz :
DISTANZ  50
POINT     ESC
END
```

3.1.39 ZAA / ZAE

■ Funktion

Kompletter Bildneuaufbau / Abbruch des Bildneuaufbaus

■ Syntax

ZAA

ZAE

■ Hinweise

Der Befehl ZAE bewirkt einen Abbruch des Bildneuaufbaus, z.B. bei der Option **Gesamtansicht**. Bei großen Zeichnungen kann dadurch die Ablaufgeschwindigkeit des Makros erhöht werden. Mit ZAA erfolgt wieder ein kompletter Bildneuaufbau.

■ Generierung durch HiCAD-Makroentwicklungssystem

Es ist keine automatische Generierung möglich.

■ Beispiel 1

Dieses Makro bricht den Bildneuaufbau bei der Funktion **Neuzeichnung Alle Teile** ab. Bei der Funktion Gesamtansicht erfolgt der Bildneuaufbau wieder komplett.

```
REM          BAUTEILE
START      2
REM ALT1
ZAE
OPTION 21  0
REM  NEUZEICHNUNG  1 = alle Teile
OPTION  1  51
REM ALT2
ZAA
OPTION 22  0
REM  ANSICHT      5 = Gesamtansicht
OPTION  5  52
END
```

■ Beispiel 2

Dieses Makro bricht den Bildneuaufbau bei der Funktion **Neuzeichnung Alle Teile** ab. Auch bei der Funktion **Gesamtansicht** erfolgt der Bildneuaufbau nicht vollständig.

```
REM          BAUTEILE
START      2
REM ALT1
ZAE
OPTION 21  0
REM  NEUZEICHNUNG  1 = alle Teile
OPTION  1  51
REM ALT2
OPTION 22  0
REM  ANSICHT      5 = Gesamtansicht
OPTION  5  52
END
```

3.2 Beispiel

Die Makrotechnik soll anhand eines einfachen Beispiels erklärt werden. Es soll ein neues Hauptteil erzeugt werden, das die 2D-Teile der aktuellen enthält. Dieses neue Hauptteil wird anschließend verschoben.

Folgende Voraussetzung sollten erfüllt sein:

- Es ist eine Konstruktion geladen.
- Der Variablenspeicher ist gelöscht worden.

Anschließend sind die folgenden Schritte auszuführen:

- Aktivierung der Funktion **Makro erstellen** (Tasten STRG+7)
- Makronamen angeben, z.B. *TEILKOP*
- Aufruf der Sonderfunktion **Neuzeichnen**
- ALT-Funktionen und Wahl der Funktion **Alle Teile**
- ALT-Funktionen und Aufruf der Sonderfunktion **ZOOM**
- Wahl der Funktion **Gesamtansicht**
- Aktivierung des Menüs **2D-ZTL**
- Aktivierung des Menüs **ZTL bearbeiten**
- Wahl der Funktion **Hauptteil neu**
- Wahl der Funktion **Teil anlegen**
- Eingabe des Teilnamens, z. B. KOPIE1
- Umschalten in das Menü **Linien**
- Auswahl der Funktion **LE kopieren**
- Bestimmung der Linienelemente durch Angabe eines Rechtecks
- Eingabe der linken, unteren Ecke durch eine Punktbestimmung
- Eingabe der rechten, oberen Ecke durch eine Punktbestimmung
- **2xEND** zum Hochschalten der Menüstruktur in das Menü **Bauteile**
- Wahl der Funktion **transformieren**
- Wahl der Funktion **verschieben**
- Angabe des *Passpunktes* auf dem Teil
- Angabe des *Passpunktes* in der Zeichnung
- Beenden der Makroerstellung durch Wahl der Funktion **MAKRO-Ende**

Das auf diese Weise generierte Makro sieht nun folgendermaßen aus:

```
REM HICAD-Next VN:1201
START 59
REM ALT1
OPTION 21 0
REM NEUZEICHNUNG 1 = alle Teile
OPTION 1 51
REM ALT2
OPTION 22 0
REM ANSICHT 5 = Gesamtansicht
OPTION 5 52
REM HICAD 2 = 2-D ZTL
OPTION 2 59
REM ZEICHENTEILE 3 = ZTL bearbeit.
OPTION 3 1
REM BAUTEILE 8 = Hauptteil neu
OPTION 8 2
REM NEUES TEIL 1 = Teil anlegen
OPTION 1 3
STRING KOPIEL
REM BAUTEILE 1 = Linien
OPTION 1 2
REM LINIEN S5= LE kopieren
OPTION 15 6
REM im Rechteck (4) - Kontur (5) - LE-Abschnitt (6) :
INTEGER 4
POINT #
POINT #
POINT ESC
OPTION ESC
REM BAUTEILE 2 = transform.
OPTION 2 2
REM TRANSFORM. 1 = verschieben
OPTION 1 27
POINT #
POINT #
END
```


Dieses Makro soll nun wie folgt korrigiert werden:

- Die Bestimmung des Auswahlrechtecks für die Funktion **Linien kopieren** soll beim Makroablauf automatisch und in Abhängigkeit von den Koordinaten der Gesamtansicht erfolgen. Diese sind in den Systemvariablen
 - ZW, ZX für die linke, untere Ecke der Gesamtansicht und
 - ZY und ZZ für die rechte, obere Ecke der Gesamtansicht.

hinterlegt.

Die Koordinaten des Auswahlrechtecks sollen dann wie folgt errechnet werden

$u1 = x\text{-Koord. der linken, unteren Ecke} = ZW - 1$

$u2 = y\text{-Koord. der linken, unteren Ecke} = ZX - 1$

$o1 = x\text{-Koord. der rechten, oberen Ecke} = ZY + 1$

$o2 = y\text{-Koord. der rechten, oberen Ecke} = ZZ + 1$

Die entsprechenden Makroanweisungen lauten:

```
%u1:=ZW -1  
%u2:=ZX -1  
%o1:=ZY +1  
%o2:=ZZ +1
```

Diese Zeilen werden in das Makro vor dem Anlegen des neuen Hauptteils eingefügt. Die Variablen u1, u2, o1 und o2 werden dann in den **POINT**-Befehlen zur Bestimmung des Rechtecks verwendet. Als Punktbestimmungsfunktion wird **ABSOLUT X,Y** verwendet.

```
REM HICAD-Next VN:1201
START 59
REM ALT1
OPTION 21 0
REM NEUZEICHNUNG 1 = alle Teile
OPTION 1 51
REM ALT2
OPTION 22 0
REM ANSICHT 5 = Gesamtansicht
OPTION 5 52
REM HicAD 2 = 2-D ZTL
OPTION 2 59
REM ZEICHENTEILE 3 = ZTL bearbeit.
OPTION 3 1
%u1:=ZW -1
%u2:=ZX -1
%o1:=ZY +1
%o2:=ZZ +1
REM BAUTEILE 8 = Hauptteil neu
OPTION 8 2
REM NEUES TEIL 1 = Teil anlegen
OPTION 1 3
STRING KOPIE1
REM BAUTEILE 1 = Linien
OPTION 1 2
REM LINIEN S5= LE kopieren
OPTION 15 6
REM im Rechteck (4) - Kontur (5) - LE-Abschnitt (6) :
INTEGER 4
POINT A u1 u2
POINT A o1 o2
POINT ESC
OPTION ESC
REM BAUTEILE 2 = transform.
OPTION 2 2
REM TRANSFORM. 1 = verschieben
OPTION 1 27
POINT #
POINT #
END
```

- Der Name des neuen Teils soll nun nach dem Makrostart abgefragt werden und variabel sein. Dazu wird am Anfang des Makros ein **VAR-Anweisung** gesetzt:

VAR \$n Name der Kopie:

Die Variable wird dann anstelle der Namenseingabe KOPIE1 in die STRING-Anweisung eingefügt.

```

REM   HICAD-Next VN:1201
START  59
VAR $N Name der Kopie:
REM ALT1
OPTION  21  0
REM   NEUZEICHNUNG  1 = alle Teile
OPTION  1  51
REM ALT2
OPTION  22  0
REM   ANSICHT  5 = Gesamtansicht
OPTION  5  52
REM   HiCAD  2 = 2-D ZTL
OPTION  2  59
REM   ZEICHENTEILE  3 = ZTL bearbeit.
OPTION  3  1
%u1:=ZW -1
%u2:=ZX -1
%o1:=ZY +1
%o2:=ZZ +1
REM   BAUTEILE  8 = Hauptteil neu
OPTION  8  2
REM   NEUES TEIL  1 = Teil anlegen
OPTION  1  3
STRING $N
REM   BAUTEILE  1 = Linien
OPTION  1  2
REM   LINIEN  S5= LE kopieren
OPTION  15  6
REM   im Rechteck (4) - Kontur (5) - LE-Abschnitt (6) :
INTEGER 4
POINT A u1 u2
POINT A o1 o2
POINT  ESC
OPTION  ESC
REM   BAUTEILE  2 = transform.
OPTION  2  2
REM   TRANSFORM.  1 = verschieben
OPTION  1  27
POINT  #
POINT  #
END

```

4 Weitere Hinweise

4.1 DXF-Dateien

Beim Erstellen von DXF-Dateien über ein Benutzermakro kann das Vorhandensein einer DXF-Datei nicht mit der Variablen VORHD gesteuert werden. Stattdessen müssen hier die Variablen JA bzw. NEIN verwendet werden. Beispiel: IF JA THEN.....

Die DXF-Layer werden im Allgemeinen den HiCAD-Schichten zugeordnet. Zur besseren Kennzeichnung der übernommenen HiCAD-Schichten werden ab HiCAD 16 die DXF-Layernamen zusätzlich übergeben und in HiCAD unter Linienparametern angezeigt. Dies erfolgt jetzt nur noch, wenn bei der DXF/DWG-Übernahme ACADHCAD.DAT die Kennung 'LAYNA 1' gesetzt ist.

4.2 Makrovariable ZDSP

In der Textmenüoberfläche können Sie über die Funktionsfolge:

2D - Bearbeiten - Schnittstelle - akt.FIG -

jetzt ein Teil aktivieren, wenn sein Index auf der Makrovariablen ZDSP liegt. ZDSP wird beispielsweise bei der Funktion **Bemaßungsinfo** gesetzt.

4.3 Texte selektieren

Im Textmenümodus von HiCAD, in den während der Makroerstellung/-bearbeitung umgeschaltet wird, gibt es die Funktion **Text selektieren**. Da ab der Version 16 die Texthöhe mit höherer Stellenanzahl gespeichert wird, ist es sinnvoll, eine Toleranz für die Texthöhe anzugeben, innerhalb der Texte selektiert werden. Diese Toleranz ist in der Systemdatei sys\TXTPAR.DAT einstellbar.

Diese Einstellung kann beim Einlesen von DXF-Dateien sinnvoll sein.

4.4 Text-Tools

Mit der Funktion **Text-Nr. Shift** im **Text/Text-Tools**-Menü des Makrorekorders können aus Ziffern bestehende Texte mit einheitlichem Textcode um einen anzugebenden Betrag erhöht werden (z.B. für 2D-Pos.-Nr.).

4.5 True-Type-Font

Im Makroablauf kann ein True Type-Font als Textparameter angegeben werden.

Voraussetzung ist, dass der entsprechende Font in der Systemdatei *TTFONT.DAT* eingetragen ist. Wählt man dann bei der Parametereinstellung des Fonts INT, kann die Angabe des TT-Fonts gemäß der Reihenfolge in der o.g. Systemdatei erfolgen.

4.6 Objektcursor

Wenn Sie im Makroablauf vor eine Info-Funktion den Ausdruck **%@idt:=0**, setzen, dann wird der Objektcursor für diese Funktion abgeschaltet. Das Abschalten des Objektcursors während des Makroablaufs ohne Benutzereingaben, kann zu einer erheblichen Beschleunigung des Makroablaufs führen.

4.7 Blechabwicklung

Die Systemvariable **\$@AWV** liefert als Ergebnis das aktuell eingestellte Verfahren für die Blechabwicklung, z.B. F = Faktorenverfahren (DIN). Das Ergebnis ist identisch mit dem Kennungsbuchstaben in der Datei *verfahren.DAT*, wobei *verfahren* der Name der entsprechenden DIN ist, z.B. DIN 6935.DAT. Diese Dateien liegen im Unterverzeichnis *MAKROABW*.

4.8 Bemaßung

Im Transformationsmenü kann man über den Menüpunkt **Bem.Trans.** die bauteilübergreifende Bemaßung ein- und ausschalten. Das Ausschalten kann erforderlich sein, wenn ein Maßfußpunkt an einem Punkt liegt, der in mehreren Teilen enthalten ist, das Maß bei der Verschiebung eines Teils aber nicht mitgehen soll.

4.9 Skizziercursor

Über die numerische Makrovariable @SKZ lässt sich steuern, ob bei Funktionen außer der Skizziertechnik selbst (z.B. Rechteckfunktion, Transformation etc.) der Skizziercursor verwendet werden soll. Hat diese Variable den Wert 0, ist der Skizziercursor ausgeschaltet, hat die Variable einen anderen Wert oder ist sie unbelegt, ist der Skizziercursor aktiv. Die Variable kann z.B. im Startmakro INITAL.MAC gesetzt werden.

4.10 Bauteilkennung

Beim Anlegen bzw. Aktivieren eines 3D-Teils wird die Systemvariable **%ZKEN** mit einer eindeutigen Bauteilkennung belegt. Ein 3D-Teil lässt sich anschließend mit dieser Kennung identifizieren, indem man bei der Bauteilauswahl als Namen **X__IDKENN** (2 Unterstriche) und in der anschließenden Abfrage die vorher bekannte Kennung angibt.

4.11 Betriebssystemaufruf

Bei Betriebssystemaufrufen wird zwischen den Zeichen / und \ unterschieden (bis HiCAD 2 1104 fand eine Umwandlung von / in \ statt).

4.12 Werkstoffschraffung

Zur Definition einer Werkstoffschraffung wird für den 1. Schraffungurcode der Wert 9999 eingegeben; hiernach erfolgt die Abfrage der einzelnen Schraffungurparameter wie Symbolnummer etc.

4.13 Export in STEP/MTA oder IGES/CATIA

Beim Schreiben von Makros zur Ausgabe von STEP/MTA und IGES/CATIA Dateien muss am Anfang des Makros manuell die Makrovariable **STYP** gesetzt werden, ansonsten wird das Makro nicht ordentlich ausgeführt.

- STEP %STYP:= 1
- MTA %STYP:= 2
- IGES %STYP:= 1
- CATIA %STYP:= 2

4.14 Temporärer Pfad

In Makros lässt sich jetzt der temporäre FILEGRUP.DAT-Pfad |: verwenden. Dieser logische Pfad wird nicht in die FILEGRUP.DAT eingetragen, kann aber benutzt werden, um beliebige Windows-Pfade einem HICAD-Pfad zuzuordnen.

Dieser Pfad muss nicht gesichert/ zurückgesichert werden, sondern muss im Makro jeweils belegt werden.

Um diesen Pfad vorzubelegen, wählen Sie während der Makroaufzeichnung die Funktion VERWALTUNG → Pfad ändern im Textmenümodus. Anschließend geben Sie als Dateigruppe | ein (| auf der <, > Taste) und als Pfad den gewünschten Windows-Pfad, z.B. D:\USERDATEN.

```

REM HICAD-Next VN:1210
REM   HiCAD
START 59
HNEXT
REM HiCAD 1 = Verwaltung 1
OPTION 1 59
REM VERWALTUNG 4 = Pfad ändern
OPTION 4 8
REM   Dateigruppe (neuer Default-Pfad) :
STRING |
REM   Dateiverzeichnis:
STRING RET
END

```

4.15 Makrovariablen für UNDO

Es stehen neue Makrovariablen für die Undo-Funktionalität zur Verfügung.

ZUNG Globales Undo

- 0 Die UNDO-Funktionalität ist generell ausgeschaltet, d.h. es besteht keine Möglichkeit, einen vorherigen Zustand zurück zu erhalten.

Bei ausgeschaltetem UNDO haben die Variablen **ZUNA** (UNDO aktiv/inaktiv) und **ZUNP** (UNDO-Pause) keine Bedeutung

- 1 Die UNDO-Funktionalität ist eingeschaltet

ZUNA UNDO aktiv/inaktiv

- 0 UNDO inaktiv
Inaktiv bedeutet, dass keine UNDO-Sicherung durchgeführt wird. Bei inaktivem UNDO hat die Variable **ZUNP** (UNDO-Pause) keine Bedeutung

- 1 UNDO aktiv

ZUNP UNDO-Pause (UNDO temporär ausschalten)

- 0 UNDO pausiert nicht
- 1 UNDO pausiert
Das Pausieren bewirkt, dass keine UNDO-Sicherung stattfindet. Der Zustand vor dem Pausieren kann jedoch jederzeit wiederhergestellt werden, da vor dem Pausieren eine Szenen-UNDO-Sicherung durchgeführt wird. Wenn das Pausieren wieder aufgehoben wird, finden wieder UNDO-Sicherungen statt.

Die Pause wird mit dem Makrobefehl **UDA/UDE** aktiviert/deaktiviert.

Bitte beachten Sie:

- Bitte verwenden Sie den Makrobefehl UDA/UDE Befehl mit äußerster Vorsicht, da sichergestellt sein muss, dass zwischen UDA und UDE nur „temporäre“ Bauteile erzeugt/ bearbeitet werden, die nach dem „UDE“ wieder den Zustand vor dem UDA haben !!
- Eine UNDO-Sicherung kann dann durchgeführt werden, wenn ZUNG = 1, ZUNA = 1 und ZUNP = 0.
- In der Menüleiste kann ein Laden der UNDO-/REDO-Sicherung durchgeführt werden, wenn das entsprechende Symbol nicht ausgegraut ist.

5 Fehlermeldungen

Folgende Fehler können beim Erstellen oder Ablauf eines Makros auftreten:

Fehlercode	Fehlerursache
-2	Leere HCGS Prozedur kann nicht korrigiert werden
-1	HCGS Prozedur ist nicht vorhanden
1	Der HC ZED erwartet einen OPTION-Befehl an Stelle des gefundenen Befehls
3	CALL- oder MAKRO-Befehl: Datei nicht vorhanden
10	Unzulässige Menüebene
20	Fehler in den Argumenten einer ZUSATZ Anweisung
21	In HCGS Prozedur fehlt eine ZUSATZ Anweisung
30	unbekannter Datensatz (Dateiname) im DATFIL-Befehl
40	Fehler in den Argumenten einer LINE oder WIED Anweisung
41	In HCGS Prozedur fehlt eine LINE oder WIED Anweisung
99	unbekannter HCGS-Befehl
100	unerlaubtes Argument im Start-Befehl
101	HCGS Prozedur ohne START-Befehl bzw. HCGS-Befehl vor START-Befehl
102	START-Befehl ohne Parameter
103	Fehlerhaftes Argument im END-Befehl
110	Fehlende Argumente im HCGS-Befehl
111	Fehlerhafte Argumente im OPTIONS-Befehl
201	VAR Px oder VAR Lx Befehl innerhalb einer Punkteingabe
202	Unbekannte Variable in VAR-Befehl
205	Unbekannte Variable in Variablenzuweisung
206	Fehler im arithmetischen Ausdruck bei Variablenzuweisung
999	Kompilierte Makroprozedur kann nicht korrigiert werden

Fehlercode	Fehlerursache
1001	Leeres Befehlsmakro
1002	HCGS Prozedur findet keinen Platz im dafür vorgesehenen Speicherbereich
1003	HCGS Prozedur ist zu groß für Korrektur
1004	Mehr als 99 Sprungadressen (Labels)
1005	Unerlaubte Sprungadresse (Label ist größer als 9999)
1006	Fehlende Sprungadresse (Label)
1007	Zu viele FOR, WHILE oder REPEAT Schleifen
1008	Falscher oder fehlender Variablenname bei NEXT
1009	Falsch geschachtelte FOR Schleifen
1010	NEXT ohne FOR bzw. unbekannter Beginn der Schleife
1011	UNTIL ohne REPEAT
1012	WHEND ohne WHILE
1019	Schleifen-Ende ohne Anfangs-Befehl
1020	Zu tiefe Verschachtelung für CALL Aufrufe
1021	FOR - Anweisung nicht korrekt
1022	Unzulässige Variablenzuweisung
1023	REPEAT-Schleife nicht beendet
1024	WHILE-Schleife nicht beendet
1025	FOR-Anweisung nicht beendet
1026	IF-Anweisung nicht beendet
1027	Hierarchie-Fehler
1028	unzulässiger Makrobefehl
1029	Unzulässige Variablenprüfung
1030	HiCAD-Funktionen ein/ausblenden nur unmittelbar vor 'OPTION' erlaubt

6 Makrobibliotheken

HiCAD bietet die Möglichkeit, Makros in Bibliotheken zusammenzufassen. Diese Bibliotheken können in das Bildschirmmenü integriert werden. Die zur Bibliothek gehörenden Makros können dann über ein PopUp-Menü schnell und komfortabel aufgerufen werden.

Die Funktionen zur Erstellung und Bearbeitung von Makrobibliotheken finden Sie auf der Registerkarte **Konstruktion** unter **Teil einfügen > Anwenderbibliothek**. Die ausführliche Beschreibung dieser Funktion finden Sie in der HiCAD Online-Hilfe im Abschnitt *Grundlagen > Anpassen von HiCAD*.

7 Die Datei FILEGRUP.DAT

Bei einigen Funktionen, z.B. beim Erstellen von Vorschaubildern für Varianten und innerhalb von Makros, wird die Systemdatei **FILEGRUP.DAT** benötigt. In dieser Datei sind alle für HiCAD intern relevanten Verzeichnisse voreingestellt. Diese Datei wird bei der Installation von HiCAD im Programmverzeichnis, d.h. im Unterverzeichnis `exe`, abgelegt und beim Start von HiCAD gelesen.

```
C:/HICAD/sys
A:C:/HICAD/szenen
B:C:/HICAD/norm
C:C:/HICAD/szenen D:C:/HICAD/ega
E:C:/HICAD/makrolib
F:C:/HICAD/szenen
G:C:/HICAD/szenen
H:C:/HICAD/szenen
I:C:/HICAD/szenen
J:C:/HICAD/ray
K:C:/HICAD/material
L:C:/HICAD/szenen
.
.
```

Abb. 9 Auszug aus der Datei FILEGRUP.DAT

Jede Zeile der Datei enthält genau eine gültige Pfadangabe. Die ersten beiden Zeichen jeder Zeile enthalten den Namen der Filegruppe, d.h. die Kurzbezeichnung des Pfades in HiCAD, z.B. A:. Danach folgt die vollständige Pfadangabe, die maximal 80 Zeichen lang sein kann.

Die erste Zeile der Datei **mu**ss immer die Pfadbezeichnung des HiCAD-Systemverzeichnisses enthalten, wobei in diesem Fall an Stelle der 2-stelligen Kurzbezeichnung Leerzeichen stehen müssen. In den Folgezeilen stehen die Pfadbezeichnungen der einzelnen HiCAD-Datenverzeichnisse.

Die in der FILEGRUP.DAT eingetragenen Verzeichnisse können über das Menü **Einstellungen** durch Auswahl des Eintrags **Verzeichnisse** geändert werden.

Änderbar sind alle Verzeichnisse, die mit einem geöffneten Schloss markiert sind. Um ein Verzeichnis zu ändern, klicken Sie auf das entsprechende Feld und wählen dann das gewünschte Verzeichnis aus.

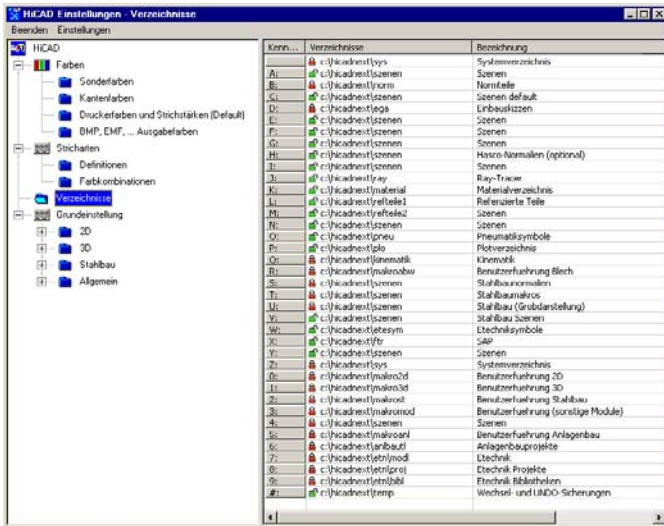


Abb. 10 Verzeichnisse ändern

Die beim Verlassen des Menüs **Einstellungen** eingetragenen Verzeichnisse bleiben auch beim Beenden von HiCAD erhalten, d.h. sie sind beim nächsten Start von HiCAD die Default-Einstellungen. Diese Einstellungen werden in der Datei FILEGRUP.DAT gespeichert.

Die bei der Installation von HiCAD eingestellten Verzeichnisse werden beim ersten Aufruf der Funktion **Verzeichnisse** in der Datei FILEGRUP.ORI gesichert. Diese Einstellungen können Sie jederzeit wiederherstellen, indem Sie unter **Einstellungen** den Eintrag **Defaulteinstellung laden** wählen.

Beachten Sie bitte:

Unabhängig von den Einstellungen in der FILEGRUP.DAT können Sie ihre Konstruktion natürlich wie bei allen Windows-Programmen in/aus jedem beliebigen Verzeichnis speichern/laden. Der hier voreingestellte Pfad für HiCAD-Konstruktionen (Szenen) wird bei einem Neustart von HiCAD als Voreinstellung verwendet. Während einer Arbeitssitzung wird beim Laden von Dateien immer der zuletzt verwendete Pfad angezeigt.

Index

- A**
- Abbruch 17
 - Antwort 47
 - APAUS 49
 - APEIN 49
 - Argumentgruppen 22
 - arithmetische Ausdrücke ... 33
 - arithmetischer Ausdruck 23
- B**
- Bauteilkennung 116
 - Bemaßung ein-/ausschalten 116
 - Betriebssystemaufruf 116
 - Blechabwicklung 115
- C**
- Call 50
 - CATIA 117
 - CLOSE 74
 - COPY 52
- D**
- Dateinamen 22
 - DEL 54
 - DISTANZ 55
 - DXF-Dateien 114
- E**
- ECHO 57
 - Einstellungen
 - Verzeichnisse 123
 - Einzelschrittmodus 16
 - END 59
 - Ende 17
 - Export
 - IGES/CATIA 117
 - STEP/MTA 117
- F**
- FAA 60
 - FAE 60
 - Fehlermeldungen 120
 - feste Argumente 23
 - FILEGRUP.DAT 123
 - FOR 61
 - freies Argument 11
 - Freies Argument 22
- G**
- GOTO 63, 65
 - Grundfunktionen 34
- H**
- HCGS-Befehle 20, 43
 - HC-MES 5
- I**
- IF 65
 - IFEND 65
 - IGES 117
 - IGNORE 67
 - INPUT 74
 - INTEGER 68
- K**
- Konstanten 22
 - Kontrollargumente 24
- L**
- Linienvariablen 25
 - logische Variable 37
 - logischer Ausdruck 23
 - logischer Vergleichs-
ausdruck 35
- M**
- Makro
 - kompilieren 19
 - MAKRO 70
 - Makro aufrufen 12
 - Makro bearbeiten 13
 - Makro erstellen 9
 - Makro korrigieren 15
 - Makro testen 17
 - Makro übernehmen 14, 17
 - Makro überschreiben 13
 - Makroaufbau 10
 - Makrobefehle 10
 - Makrobeispiel 109
 - Makrobibliotheken 122
 - Makroentwicklungssystem .. 5

Makroerstellung beenden ..	12
Makroerstellungsmodus	9
Makroname	10
Makrorahmen	10
Makrosprache.....	20
mathematische Funktionen	34
MAUS	72
MEIN	72
MKDIR.....	73
MTA.....	117
N	
NEXT	61
Nummern.....	22
O	
Objektcursor abschalten..	115
OPEN	74
OPTION.....	77
OUTPUT.....	74
P	
Pfad	
temporär.....	117
Pfadangaben	123
PFD	80
POINT	81
Punktoption	24
Punktvariable.....	25
R	
REAL	83
Rechenoperatoren.....	33
REM	84
REPEAT	86
Run.....	16
S	
SAUS.....	87
SEIN	87
Skizziercursor	116
START	89
STEP	117
STRING.....	90
Stringausdruck	39
Stringfunktionen.....	34, 40
Stringoperationen	40
STYP	117
Systemdateien	
FILEGRUP.DAT	123
SZ AUS.....	92
SZEIN.....	92
T	
Temporärer Pfad	117
Text selektieren	114
Text-Tools.....	115
THEN.....	65
True Type-Font.....	115
U	
UDA	93
UDE	93
UNDO-Variablen.....	118
V	
VAI.....	94
VAR	95
Variable	17
benutzerspezifische.....	25
Variablen	23
Variablenspeicher	42
Variablenzuweisung.....	97
Verzeichnisse	
voreinstellen	123
W	
WAIT	99
WARTE.....	100
WAUS.....	101
WEIN	101
Werkstoffschraffur	116
WERT	102
WHILE	103
WINKEL.....	105
Z	
ZAA.....	107
ZAE.....	107
ZDSP	114
Zoom-Funktionen.....	16

© 2009 ISD ® Software und Systeme GmbH alle Rechte vorbehalten

Dieses Handbuch sowie die darin beschriebene Software werden unter Lizenz zur Verfügung gestellt und dürfen nur in Übereinstimmung mit den Lizenzbedingungen verwendet oder kopiert werden. Der Inhalt dieses Handbuches dient ausschließlich zur Information, kann ohne Vorankündigung verändert werden und ist nicht als Verpflichtung von ISD Software und Systeme GmbH anzusehen. Die ISD Software und Systeme GmbH gibt keine Gewähr oder Garantie hinsichtlich der Richtigkeit oder Genauigkeit der Angaben in dieser Dokumentation. Kein Teil dieser Dokumentation darf, außer durch das Lizenzabkommen ausdrücklich erlaubt, ohne vorherige, schriftliche Genehmigung von ISD Software und Systeme GmbH reproduziert, in Datenbanken gespeichert oder in irgendeiner Form übertragen werden.

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

Herausgeber und Redaktion: © 2009 ISD ® Software und Systeme GmbH

ISD Software und Systeme GmbH
Hauert 4
D-44227 Dortmund
Tel. +49-(0)231-9793-0
Fax +49-(0)231-9793-101
info@isdgroup.de
www.isdgroup.de

ISD Berlin
Paradiesstraße 208a
D-12526 Berlin
Tel. +49-(0)30-634178-0
Fax +49-(0)30-634178-10
berlin@isdgroup.de
www.isdgroup.de

ISD Hannover
Ahrensburger Straße 3
D-30659 Hannover
Tel. +49-(0)511-616803-40
Fax +49-(0)511-616803-41
hannover@isdgroup.de
www.isdgroup.de

ISD Nürnberg
Nordostpark 7
D-90411 Nürnberg
Tel. +49-(0)911-95173-0
Fax +49-(0)911-95173-10
nuernberg@isdgroup.de
www.isdgroup.de

ISD Ulm
Wilhelmstraße 25
D-89073 Ulm
Tel. +49-(0)731-96855-0
Fax +49-(0)731-96855-10
ulm@isdgroup.de
www.isdgroup.de

ISD Austria GmbH
Hafenstraße 47-51
A-4020 Linz
Tel. +43-(0)732-9015-1800
Fax +43-(0)732-9015-1829
info@isdgroup.at
www.isdgroup.at

ISD Benelux b.v.
Het Zuiderkruis 33
NL-5215 MV 's-Hertogenbosch
Tel. +31-(0)73-61538-88
Fax +31-(0)73-61538-99
info@isdgroup.nl
www.isdgroup.nl

ISD East Europe Sp.z.o.o.
Ul. Fortuny 6B
01-339 Warszawa, Polen
Tel. +48-(0)22-86205-29
Fax +48-(0)22-86205-20
info@isdgroup.pl
www.isdgroup.pl

ISD Italia s.r.l.
Viale Zanotti, 76
I-27027 Gropello Cairoli (Pavia)
Tel. +39-(0)382-815772
Fax +39-(0)382-826112
info@isdgroup.it
www.isdgroup.it

ISD Japan Co. Ltd.
Daiwajisho-Building 4th floor
74-1 Yamashita-cho Naka-Ku,
Yokohama, Kanagawa, 231-0023 Japan
Tel. +81-(0)45-315-9605
Fax +81-(0)45-315-9607
info@isdgroup.jp
www.isdgroup.jp

ISD Schweiz AG
Rosenweg 2
CH-4500 Solothurn
Tel. +41-(0)32-62413-40
Fax +41-(0)32-62413-42
info@isdgroup.ch
www.isdgroup.ch



VERWALTUNG

- 1 = File kopieren
- 2 = File löschen
- 3 = File umben.
- 4 = Pfad ändern
- 5 = Pfadzuordnung
- 6 = Linienarten
- 7 = Schraffurart
- 8 = Zeichensatz
- 9 = Attrib.-Namen

DEFAULTPARAM.

- 10= Maßeinheit
- S1= Linienpar.
- S2= Textparameter
- S3= Bemaßungspar.
- S4= Plotterparam.
- S5= Menü/Tablett
- S6= Allgem.Param.
- S7= 2D-Makroleiste
- S8= 3D-Makroleiste

ALT-Funktionen

MAKRO -Ende

VERWALTUNG 2

FREIHANDSYMB.

- 1 = EIN/AUS
- 2 = definieren
- 3 = anzeigen
- 4 = NKS Koord.
- 5 = Exp-Form.Ausg
- 6 = Iconleisten
- 7 = Makrodatei
- 8 = Systemdatei
- 9 = Makrofeld
- 10= Modul-Lizenz
- S1= Applikation.
- S2= ISD-Service
- S3= Wörterbuch

ALT-Funktionen

MAKRO -Ende

HiCAD

- 1 = Verwaltung 1
- 2 = 2-D ZTL
- 3 = 3-D SZENE
- 4 = Layout
- 5 = plotten
- 6 = Bibliotheken
- 7 = ZV-Datenbank
- 8 = Verwaltung 2
- 9 = Maschinenbau
- 10= Stahlbau
- S1= Bauplanung
- S2= Kinematik
- S3= E-Technik
- S4= Anlagenbau
- S4= Dateiliste
- S6= 2d/3d-TOOLS
- S7= Objekt-Ident.
- S8= UNDO/RE-UNDO
- S9= E N D E

ALT-Funktionen

MAKRO -Ende

ZEICHENTEILE

- 1 = ZTL anlegen
- 2 = ZTL laden
- 3 = ZTL bearbeit.
- 4 = ZTL speichern
- 5 = Stückliste
- 6 = Datenaustaus.
- 7 = Symboltabelle
- 8 = Detail
- 9 = Dateiverz.
- 10= Zusatzinfo.
- S1= NC
- S2= Maß-System
- S3= Variante-geo.
- S4= Variante-par.
- S5= Variante-HCM
- S6= ZTL wechseln
- S7= ZTL löschen
- S8= CAQ
- S9= Scannen
- S10=ZTL-TOOLS

ALT-Funktionen

MAKRO -Ende

BAUTEILE

- 1 = Linien
- 2 = transform.
- 3 = Baut.-Param.
- 4 = wiederholen
- 5 = Attrib.-Sel.
- 6 = Bt. speichern
- 7 = Bt. löschen
- 8 = Hauptteil neu
- 9 = Sub-Teil neu
- 10= Bt. auswählen
- S1= Bt. markieren
- S2= Bemaßung
- S3= spez. Linien
- S4= Schraffur
- S5= LE sortieren
- S6= Texte
- S7= Hilfslinien
- S8= Bt.-Gruppen
- S9= Schnittstelle
- S10=BT.-TOOLS

ALT-Funktionen

MAKRO -Ende

LINIEN

- 1 = Isolierte Pkt
- 2 = Streckenzug
- 3 = Parallele
- 4 = Rechteck
- 5 = Kreis/-bogen
- 6 = Kegelschnitte
- 7 = LE löschen
- 8 = LE korrig.
- 9 = Par. Vorgabe
- 10= Par. ändern
- S1= Tangente
- S2= Streckensymm.
- S3= Winkelsymmetr
- S4= LE entnehmen
- S5= LE kopieren
- S6= Pktbezeichng
- S7= Symbol setzen
- S8= Strzug Datei
- S9= Freihandlinie
- S10=LE-TOOLS

ALT-Funktionen

MAKRO -Ende

