



KfK 3216
Januar 1982

GIPSY-Handbuch

Band II

G. Enderle, K. H. Bechler, H. Grimme, W. Hieber, F. Katz
Institut für Reaktorentwicklung

Kernforschungszentrum Karlsruhe

KERNFORSCHUNGSZENTRUM KARLSRUHE
Institut für Reaktorentwicklung

KfK 3216

G I P S Y - Handbuch

Band II

G. Enderle
K.H. Bechler
H. Grimme 1)
W. Hieber 2)
F. Katz

1) Bayrische Motorenwerke AG, München

2) Maschinenfabrik Augsburg-Nürnberg AG, Augsburg

Als Manuskript vervielfältigt
Für diesen Bericht behalten wir uns alle Rechte vor

Kernforschungszentrum Karlsruhe GmbH
ISSN 0303-4003

Zusammenfassung

Band II des GIPSY-Handbuches enthält die vollständige Information zur Anwendung der in den Jahren 1980 und 1981 neu in GIPSY aufgenommenen Fähigkeiten. Es sind dies: Bemaßungsroutinen, Konturen, Schraffuren, Einlesen von AGF-Plotfiles, Aufnahme der Hershey-Text-Zeichensätze und die interaktive GIPSY-Anwendung.

Abstract

GIPSY User Manual Volume II

Volume II of the GIPSY user manual contains the complete information necessary for application of new GIPSY-features that have been integrated into the GIPSY drafting system in 1980 und 1981. Those features are:

Dimensioning procedures, profiles, hatching, interpretation of the AGF-plotfile, integration of the Hershey-character fonts and the interactive application of GIPSY.

Die vorliegende Erweiterung von GIPSY konnte nur in Zusammenarbeit mit Anwendern innerhalb und außerhalb des KfK verwirklicht werden. Insbesondere wurde ein wesentlicher Teil der Arbeiten von Mitarbeitern der Firmen BMW (Bayrische Motorenwerke AG, München) und MAN (Maschinenfabrik Augsburg-Nürnberg AG, Augsburg) durchgeführt. Auch das Erkennen und Beseitigen von Fehlern war nur durch die konstruktive Zusammenarbeit mit den Anwendern möglich.

I n h a l t

	Seite
1. Einleitung	3
2. Maßroutinen	5
2.1 Übersicht über die Bemaßungsfunktionen	5
2.1.1 Längenmaße	5
2.1.2 Winkelmaße	7
2.1.3 Radienmaße	8
2.1.4 Zusatzroutinen	9
2.2 Parameter der Bemaßungsroutinen	10
2.3 Aufruf der Bemaßungsfunktionen	15
2.3.1 Einteilung der Maßroutinen in Gruppen	15
2.3.2 Längenmaße	16
2.3.3 Winkelmaße	21
2.3.4 Radienmaße	23
2.3.5 Zusatzroutinen	25
3. Konturen und Schraffuren	29
3.1 Linienzüge	29
3.2 Erzeugung von Linienzügen, Konturen und Schraffuren	31
3.3 Rundungen und Fasen	34
4. Einlesen von AGF-Plotfiles (GIPFI)	36
4.1 Plotfile-Identifikation und -Aufbau	36
4.2 GIPFI-Routinen	37
4.3 Fehlermeldungen	38
5. Anwendung der Hershey-Zeichensätze	39
5.1 Die Zeichensätze	39
5.2 Ergänzungen in GIPSY	40
5.3 AGF-Plotfile	59
5.3.1 Kodierung der Zeichensatz-Nummer im Plotfile	59
5.3.2 Ausgabe des AGF-Plotfiles	59

	Seite
6. Interaktive Anwendung von GIPSY(IGIPSY)	60
6.1 Aufruf des Programms	61
6.2 Die Bildstruktur in GIPSY	63
6.3 Die IGIPSY-Menüs	64
6.4 Die Funktionen	71
6.5 Anwendung der IGIPSY-Version 1	81
7. Erweiterung der Plotfile-Interpreterer	82
8. Alphabetischer Handbuchteil	83
Literatur	191

.....

1. Einleitung

Das GIPSY-Handbuch / 1 / hat die Anwendung des graphischen Systems GIPSY / 2 / wesentlich vereinfacht. Sowohl für das Einführen neuer Anwender als auch als Nachschlagewerk für routinierte Benutzer hat es sich bewährt. Seit seinem Erscheinen im März 1980 wurden jedoch in GIPSY eine Reihe von Fähigkeiten ausgebaut oder neu integriert, die bisher nur in einzelnen Berichten dokumentiert waren. Das vorliegende GIPSY-Handbuch Band II soll die neuen GIPSY-Fähigkeiten zusammenfassend im Stile des GIPSY-Handbuches beschreiben.

Die neuen GIPSY-Fähigkeiten sind:

- Maßroutinen

Im Rahmen der Zusammenarbeit zwischen KfK und MAN, Augsburg, wurden seitens MAN Unterprogramme zur Bemaßung technischer Zeichnungen entwickelt. Diese Unterprogramme wurden von BMW, München, erweitert und verbessert und dann von KfK in GIPSY integriert.

- Konturen und Schraffuren

Im zweidimensionalen Bereich wurde GIPSY um Funktionen zum Behandeln von Flächen erweitert. Die Erweiterungen erlauben die problemgerechte Beschreibung zweidimensionaler Flächen, der Kontur solcher Flächen und von Schraffuren innerhalb von Flächen. Rundungen und Fasen können an Eckpunkten einer Kontur angebracht werden. Diese Fähigkeiten sind zur Erzeugung technischer Zeichnungen erforderlich.

- Einlesen von AGF-Plotfile (GIPFI)

Schon bisher konnte man mit GIPSY Bilder auf den AGF-Plotfile / 3 / ausgeben, um sie dann mit Hilfe von verschiedenen Plotfile-Interpretierern auf Zeichengeräten oder Bildschirmen darzustellen. Mit dem GIPSY-Plotfileinterpretierer (GIPFI) ist

es jetzt möglich, den Inhalt eines AGF-Plotfiles ganz oder teilweise in GIPSY einzulesen und einer GIPSY-Kollektion zuzuweisen. Es können ganze Plotfilebilder gelesen werden; Teilbilder können fortlaufend gelesen oder übergangen werden, sie können auch nach ihrem Namen ausgewählt werden.

- Aufnahme der Hershey-Zeichensätze

Die vierzehn Hershey-Zeichensätze dienen zum Erzeugen von ansprechenden Darstellungen von Textzeichen auf Bildschirm und Zeichengeräten. Die Zeichensätze wurden in GIPSY, im AGF-Plotfile und in den Plotfile-Interpretierern verfügbar gemacht.

- Interaktive Anwendung von GIPSY(IGIPSY)

Das Programm IGIPSY ermöglicht die interaktive Anwendung von GIPSY. An einem graphischen Arbeitsplatz IBM 3277-GA, der an die IBM-Großrechenanlage angeschlossen ist, können Bilder vom AGF-Plotfile eingelesen oder aus Grundelementen aufgebaut, dargestellt, transformiert, geändert und auf den Plotfile ausgegeben werden.

In den folgenden Kapiteln werden die einzelnen GIPSY-Erweiterungen beschrieben. Kapitel 8 enthält einen alphabetischen Handbuchteil mit den neuen oder erweiterten GIPSY-Funktionen. Bei Erweiterung von GIPSY-Fähigkeiten wird auf die entsprechenden Kapitel des GIPSY-Handbuches Band I (GHB) verwiesen.

2. Maßroutinen

Die GIPSY-Maßroutinen dienen der Bemaßung technischer Zeichnungen. Die Bemaßungsroutinen sind in 3 Gruppen unterteilt, nämlich solche für Längenmaße, Winkelmaße und Radienmaße.

Um die Bemaßungsroutinen in die GIPSY-Sprache zu integrieren, wurden sie als GIPSY-Funktionen realisiert, die als Ergebnis eine Kollektion zurückliefern. Man kann also schreiben:

```
SET collection = MASS..(...);
```

Dies ermöglicht das Sammeln aller Bemaßungen in einer Kollektion, auf die dann weitere GIPSY-Operationen angewandt werden können (z.B. CHANGE oder EDIT von Attributen).

Einige globale Variable der Maßroutinen.

können durch das CHANGE-Statement geändert werden, weswegen auch die CHANGE-Anweisung erweitert wurde.

2.1 Übersicht über die Bemaßungsfunktionen

Es wurde bewußt darauf verzichtet, eine generelle Bemaßungsroutine zu erstellen, die alle Möglichkeiten umfaßt, da sonst eine unübersichtliche, lange Parameterliste die Folge gewesen wäre. Das führte zu einer Unterteilung in Längen-, Winkel- und Radienbemaßung. Diese werden im folgenden dargestellt.

2.1.1 Längenmaße

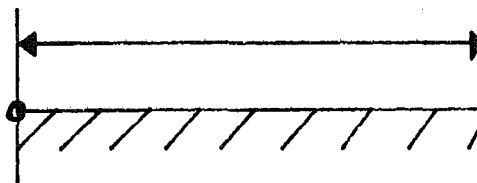
Da die Längenmaße in einer Fertigungszeichnung hauptsächlich unten, rechts, oben oder links angeordnet sind, wurden deshalb namentlich eigene Routinen erstellt. Zusätzlich wurden diese Routinen in verschiedenen Maßtypen unterteilt (zweistellige Zahl im Namen, z.B.Ø1 = zwei gleich lange Maßhilfslinien mit Maßzahl in Maßlücke).

Analog dazu besteht eine Gruppe von Routinen für Längenmaße unter beliebigem Winkel. Demnach haben die Längenmaßroutinen folgenden Programmaufbau:

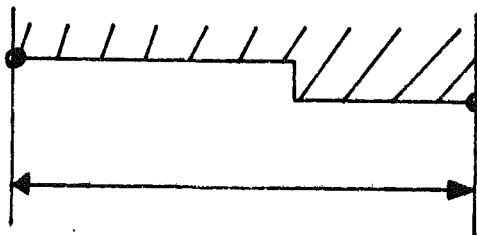
Masslinie innerhalb Massluecke	Von aussen angesetzte Pfeile	Beliebiger Winkel, Masslinie innerhalb Massluecke	Beliebiger Winkel, von aussen ange- setzte Pfeile
MØ1U (U/R/O/L) MØ2 " MØ3 " MØ4 " MØ5 " MØ6 "	M11 (U/R/O/L) M12 " M13 " M14 " M15 " M16 "	MØ1G MØ2G MØ3G MØ4G MØ5G MØ6G	M11G M12G M13G M14G M15G M16G

Die Typen 1 bis 6 haben folgende Bedeutung:

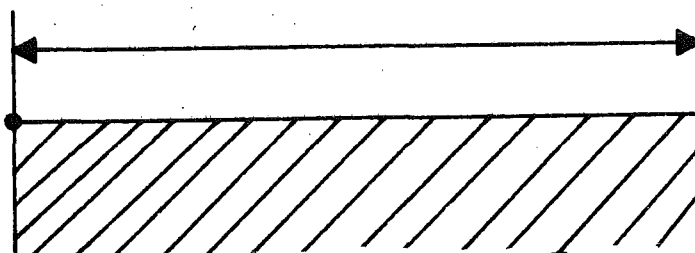
MØ1 und M11: Längenmaß mit 2 gleich langen Maßhilfslinien



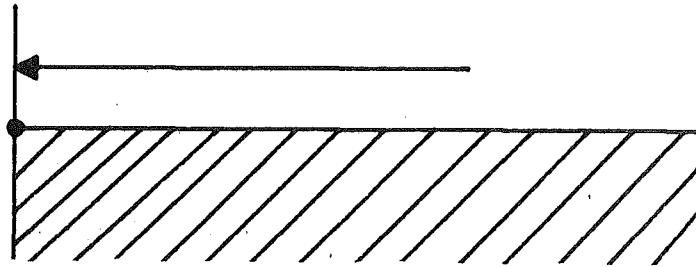
MØ2 und M12: Längenmaß mit zwei unterschiedlich langen Maßhilfslinien



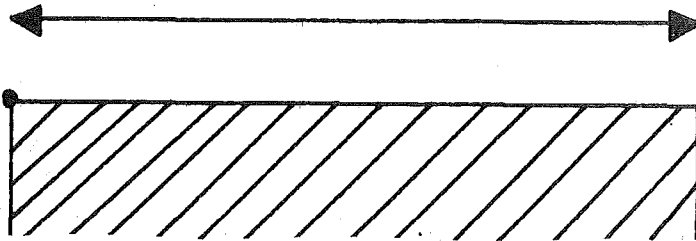
MØ3 und M13: Längenmaß mit zwei Maßpfeilen und einer Maßhilfslinie



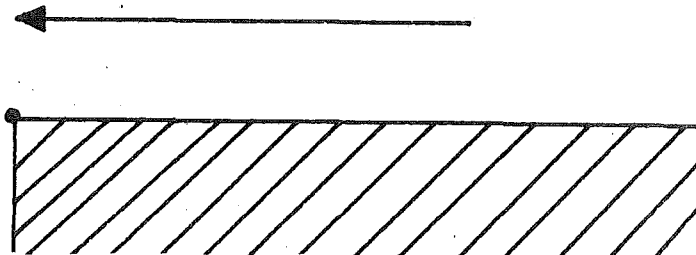
MO4 und M14: Längenmaß mit einem Maßpfeil und einer Maßhilfslinie



MO5 und M15: Längenmaß mit zwei Maßpfeilen aber ohne Maßhilfslinie



MO6 und M16: Längenmaß mit einem Maßpfeil ohne Maßhilfslinie



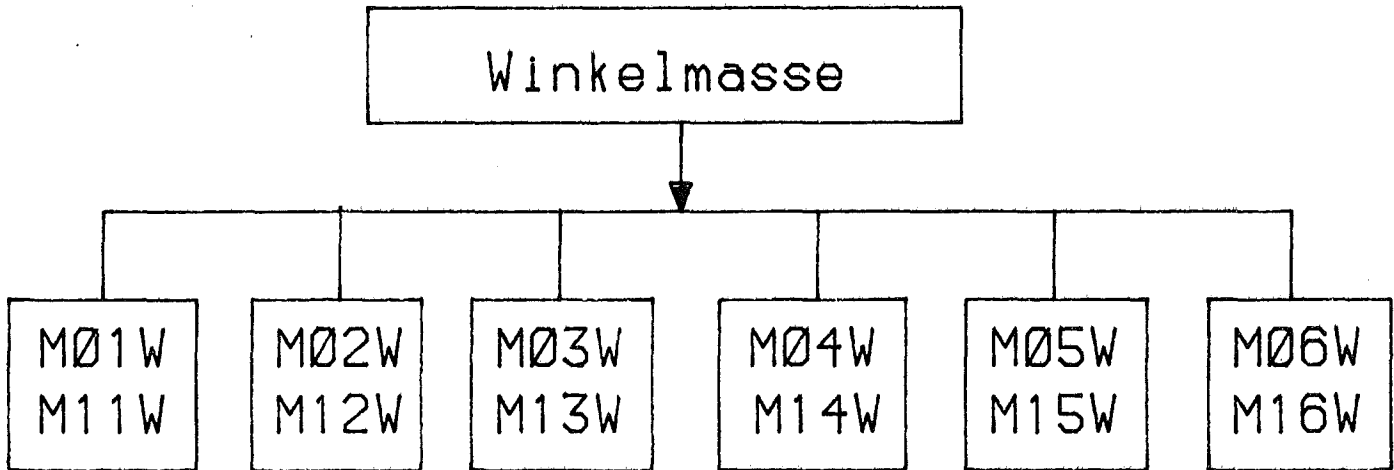
2.1.2 Winkelmaße

Der Programmaufbau ist analog dem Aufbau der Längenmaßroutine. Bei den Winkelmaßen wird unter sechs Typen unterschieden:

- MØ1W Winkelmaß mit zwei gleich langen Maßhilfslinien bis zum Mittelpunkt
- MØ2W Winkelmaß mit unterschiedlich langen Maßhilfslinien
- MØ3W Winkelmaß mit nur einer Maßhilfslinie
- MØ4W Winkelmaß ohne Maßhilfslinien
- MØ5W Winkelmaß mit nur einer Maßhilfslinie und einem Maßpfeil

- M06W Winkelmaß mit nur einem Maßpfeil

Analog dazu Routinen mit von außen angesetzten Pfeilen (M11W, M12W, M13W, M14W, M15W, M16W).

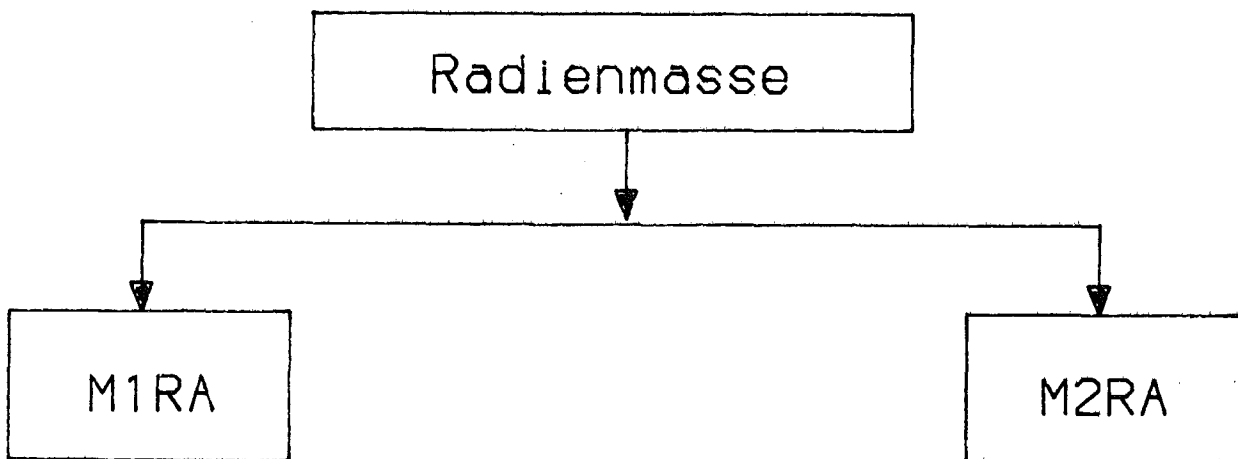


2.1.3 Radianmaße

Die Radianmaße sind in zwei Typen unterteilt:

- M1RA Radianmaß ohne Mittelpunktskennzeichnung, d.h. mit "R" vor Maßzahl
- M2RA Radianmaß mit Mittelpunktskennzeichnung

Der Programmaufbau ist wieder entsprechend der Längen- und Winkelmaße.



2.1.4 Zusatzroutinen

Eine Reihe von Routinen dient zur Darstellung weiterer Symbole, die in technischen Zeichnungen benötigt werden:

MLFT	Lage- und Formtoleranzen nach DIN 7184
MBEZ	Bezugsangabe für Lage- und Formtoleranzen
MBAZ	Bearbeitungszeichen nach DIN 1302
MTEX	allgemeiner Text
MWSK	Werkstückkanten
MARW	Maßpfeil
MEZH	Einzelheitvergrößerung
MSVL	Schnittlinienverlauf

2.2 Parameter der Bemaßungsroutinen

Alle Bemaßungsroutinen benutzen die Möglichkeit, die im GIPSY-Programm meist zuvor bestimmten Konturpunkte namentlich in der Parameterliste als Referenzpunkt der Bemaßung mit zu übergeben. Außerdem wurde darauf geachtet, die Parameterliste möglichst identisch in ihrer Reihenfolge und Bedeutung aufzubauen.

Die allgemeine Parameterbeschreibung unterliegt folgender Systematik:

Syntax:

M... (P1, Maßzahl, [Lage], [Start-Winkel], [MH-Länge],
[Zusatz-Information], [Platzierung])

gilt für Mxyz mit $y \neq 2$ bzw.

M... (P1, P2, Maßzahl, [Lage], [Start-Winkel], [MH-Länge],
[Zusatz-Information], [Platzierung])

bei Mxyz mit $y = 2$.

Erläuterung:

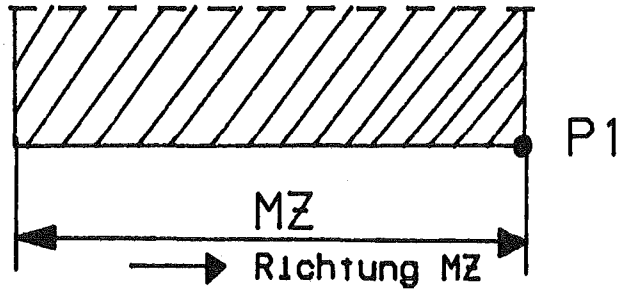
M... : Name der Maßroutine des gewünschten Bemaßungstyps,
z.B. MØ1U, M16W...

P1, P2 : Namen von zuvor erzeugten Konturpunkten. Diese
Punkte müssen graphische Objekte vom Typ POINT2 sein.

Längenmaß:

Lage : Lage des Konturpunktes P1.
Kennung ist ein Zeichen entweder 'L' (für links)
oder 'R' (für rechts). 'R' bedeutet, daß P1 in Rich-
tung Maßtext plaziert ist, wenn dieser in der Maßlücke
stehen würde.

Beispiel: Lage = 'R'

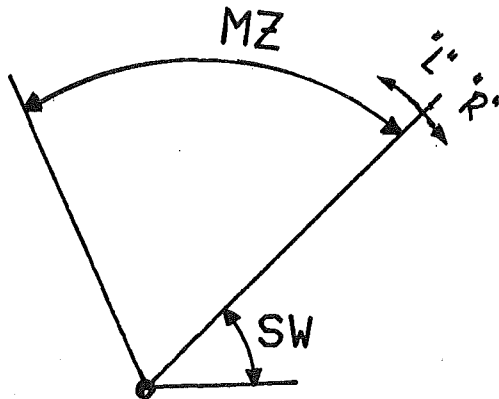


Winkelmaße:

Ausgehend vom Start-Winkel die Richtung des Maßwinkels im mathematisch positiven Sinn.

Beispiel:

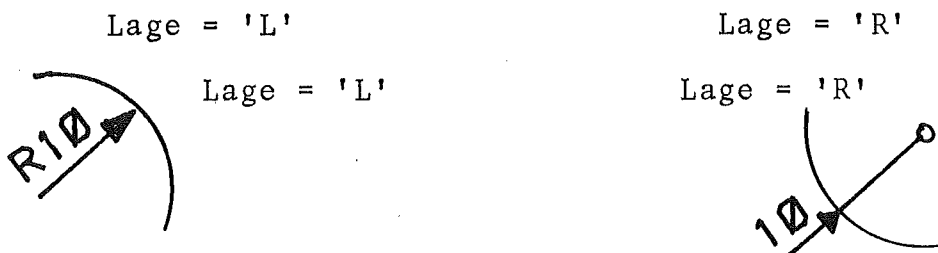
Lage = 'L'



Radienmaß:

Lage = 'L' = Bemaßung innerhalb (vom Mittelpunkt zum Bogen)

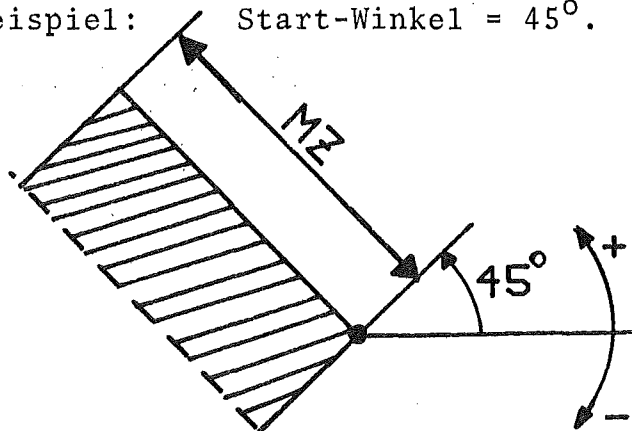
Beispiel:



Start-Winkel: Längenmaß unter beliebigen Winkel:

Startwinkel der Maßhilfslinie.

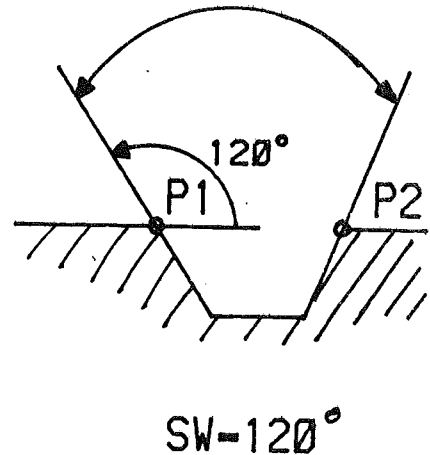
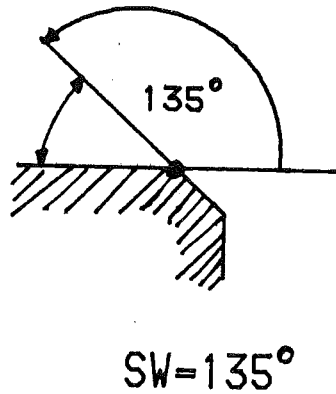
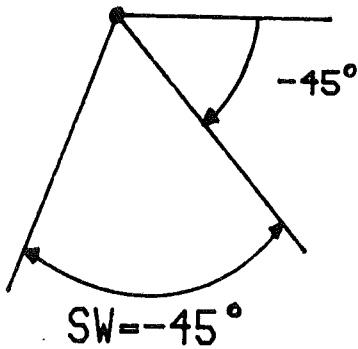
Beispiel: Start-Winkel = 45°.



Winkelmaße:

Bei Angabe von zwei Konturpunkten ist Startwinkel SW am Punkt P1 anzugeben.

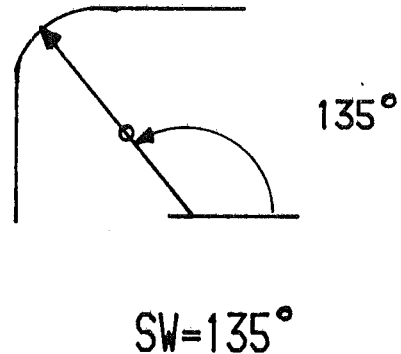
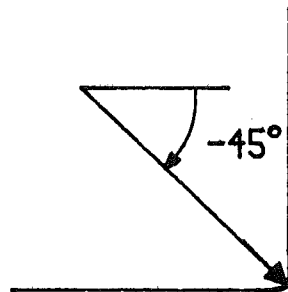
Beispiel:



Radienmaß:

Startwinkel ist immer Vektorenrichtung vom Mittelpunkt zur Pfeilspitze.

Beispiel:





[MH-Länge]

: Maßhilfs-Linienlänge von Konturpunkt P1 aus. Angabe muß in CM erfolgen. Die CM-Angabe wurde aus Gründen der gedanklich praktischen Abstandsabschätzung ("Denken in Zentimetern") gewählt.

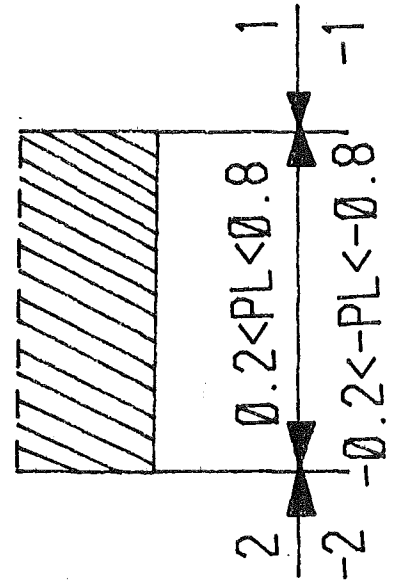
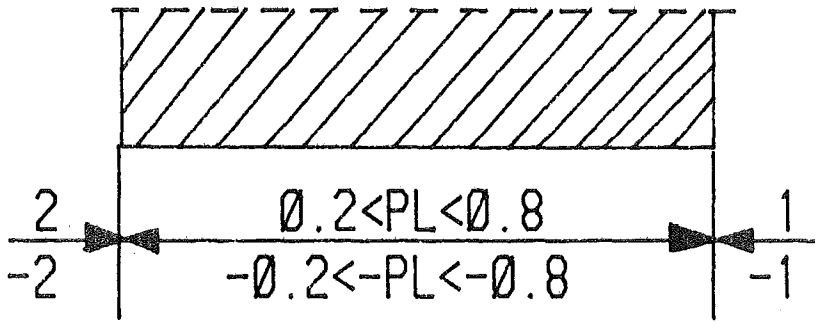
Maßzahl : Sie bedeutet zugleich Maßlückengröße und Maßtext.

Zusatz-Information : Eine Zeichenkette von der maximalen Länge von 256 Zeichen, wobei zur Steuerung das Trennungszeichen § mit anschließendem Kennbuchstabe(n) gewählt wurde. Es gibt folgende Informationsarten :

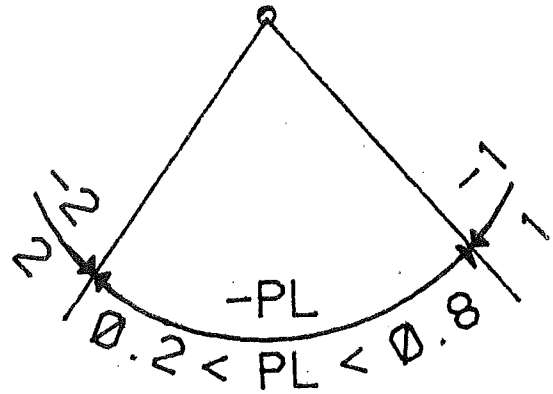
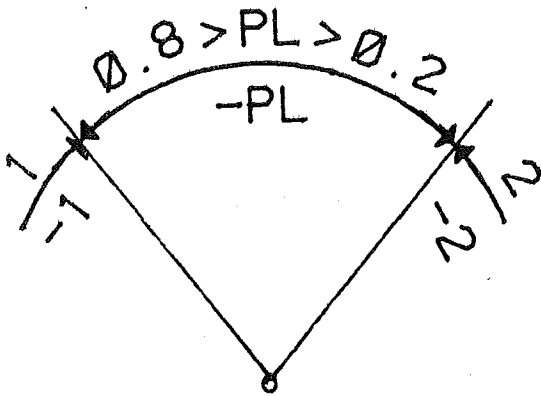
\$TV....	Beliebiger Text vor der Maßzahl.
\$TN....	Beliebiger Text nach der Maßzahl.
\$F....	Fasenangabe, z.B. \$F45 bei 45°
\$TK....	Teilkreisangabe bei Winkelmaßen z.B. \$TK6 bei 6x6∅°
\$D....	Bei Durchmesserzeichen vor Maßzahl
\$OA....	Oberes Abmaß, z.B. \$O+∅1 oder \$OH7
\$UA....	Unteres Abmaß, z.B. \$U∅.15 oder \$U-∅.2
\$OU....	Für oberes und unteres Abmaß, z.B. \$OU∅.1 ergibt <u>+∅.1</u>
\$M...	Für Gewindeangaben, z.B. \$M2 bei M12x2
\$WM...	Bei Winkelmaßen, wenn Angabe in Grad, Minuten und Sekunden erfolgen soll
\$MMC...	maximal Material Bedingung (M)
\$PTZ...	projizierte Toleranzzone (P)
\$PM...	Prüfmaß 
\$HM....	Hilfsmaß ()
\$TM....	theoretisches Maß 
\$BM....	Bei Winkelmaßen, wenn Angabe im Bogenmaß erfolgen soll
\$UM....	unmaßstäbliches Maß <u> </u>

Plazierung: Grundsätzlich wird versucht, die Maßzahl zwischen die Maßhilfslinien zu setzen. Dazu dienen die Nachkommastellen. Sollte die Maßzahl nicht in die Maßlücke passen, so wird die Ziffer vor dem Komma interpretiert.

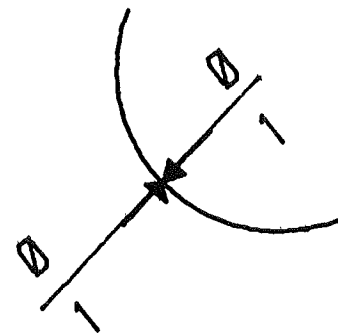
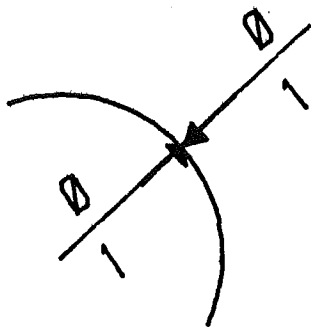
Bei Längenmaß



Bei Winkelmaß



Bei Radienmaß



2.3 Aufruf der Bemaßungsfunktionen

2.3.1 Einteilung der Maßroutinen in Gruppen

MØ1U, MØ1R, MØ1O, MØ1L

MØ2U, MØ2R, MØ2O, MØ2L

MØ3U, MØ3R, MØ3O, MØ3L

Gruppe I MØyz

MØ4U, MØ4R, MØ4O, MØ4L

MØ5U, MØ5R, MØ5O, MØ5L

MØ6U, MØ6R, MØ6O, MØ6L

M11U, M11R, M11O, M11L

M12U, M12R, M12O, M12L

M13 , M13R, M13O, M13L

Gruppe II M1yz

M14U, M14R, M14O, M14L

M15 , M15R, M15O, M15L

M16 , M16R, M16O, M16L

MØ1G, MØ2G, MØ3G, MØ4G, MØ5G, MØ6G,

Gruppe III MØxG

M11G, M12G, M13G, M14G, M15G, M16G,

Gruppe IV M1xG

Obige Tabelle umfaßt sämtliche Längenbemaßungen. Gruppe I (MØyz mit z=U,R,O,L) sind Längenbemaßungen, bei denen die Maßlinie innerhalb der Maßlücke gekennzeichnet wird. Bei II (M1YZ mit =U,R,O,L) werden die Maßpfeile von außen angesetzt. z gibt die Lage des Maßtextes bezüglich des Konturpunktes an; dabei steht U für unten, R für rechts, O für oben und L für links. III und IV sind Längenbemaßungen mit beliebigem Startwinkel. Bei III wird die Maßlinie innerhalb der Maßlücke gezeichnet, bei IV werden von außen Pfeile angesetzt.

Längenbemaßungen mit $y=2$ (Mxyz) beziehen sich auf Zeichnungen mit 2 Konturpunkten. In allen anderen Fällen ($y \neq 2$) existiert nur ein Konturpunkt.

MØ1W, M11W, MØ2W, M12W

V MxyW

MØ3W, M13W, MØ4W, M14W

MØ5W, M15W, MØ6W, M16W

Gruppe V umfaßt die Winkelbemaßung. MØ1W und M11W haben 2 gleich lange Maßhilfslinien zum Mittelpunkt. Bei MØ2W und M12W sind die Maßhilfslinien unterschiedlich lang. MØ3W und M13W haben nur eine Maßhilfslinie und MØ4W sowie M14W keine Maßhilfslinien. Bei $x=1$ (M1yW) sind die Pfeile von außen angesetzt, bei $y=2$ liegen 2 Konturpunkte vor. ($y \neq 2$ nur ein Konturpunkt).

Schließlich gibt es noch die Radienmaße

M1RA, M2RA

VI

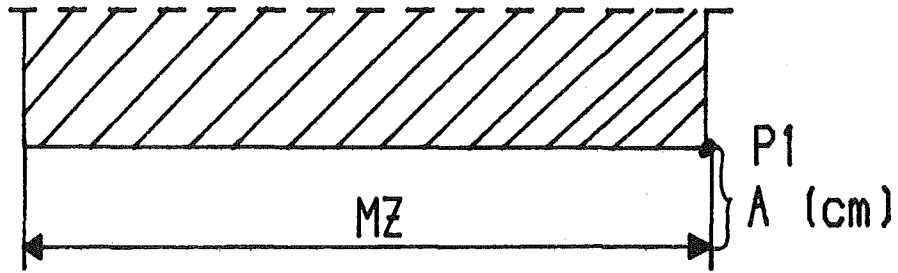
Dabei steht M1RA für Radienmaße ohne und M2RA für solche mit Mittelpunktkennzeichnung.

2.3.2 Längenmaße

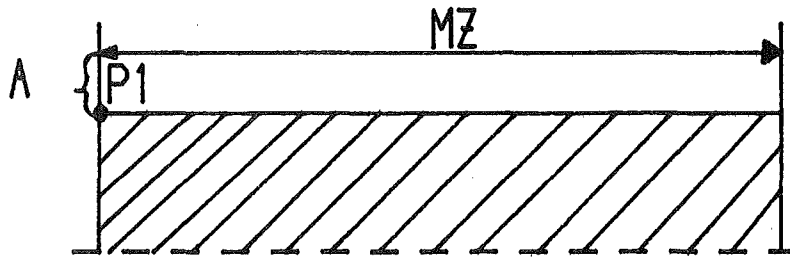
Es bestehen 64 namentlich unterschiedliche Längenmaßroutinen. Sie sind unterteilt durch ihre Lage (z.B. unten, rechts, oben, links, oder unter beliebigem Winkel wie MØ1U, MØ1R, MØ1O, oder MØ1G) und durch ihren unterschiedlichen Aufbau (z.B. Maß mit zwei gleich langen Maßhilfslinien und Maßlinie innerhalb der Maßlücke - MØ1U, oder nur Maßlinie innerhalb Maßlücke ohne Maßhilfslinien - MØ4U).

Skizzen zu den verschiedenen Maßtypen folgen:

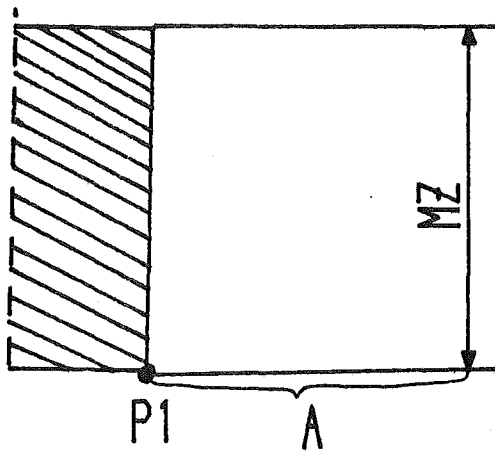
MØ1U(P1 ,MZ ,'R' ,A ,' ' ,Ø.) ;



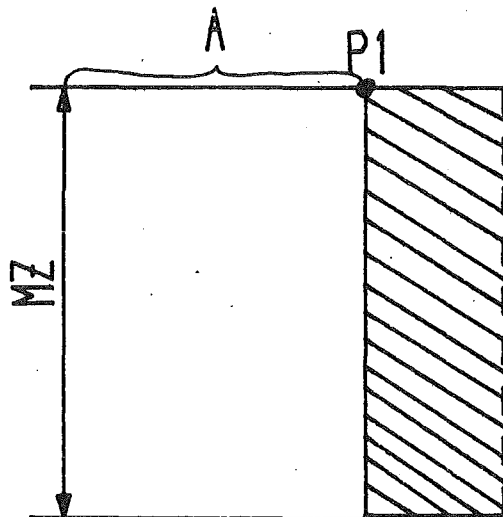
MØ1O(P1 ,MZ ,'L' ,A ,' ' ,Ø.) ;



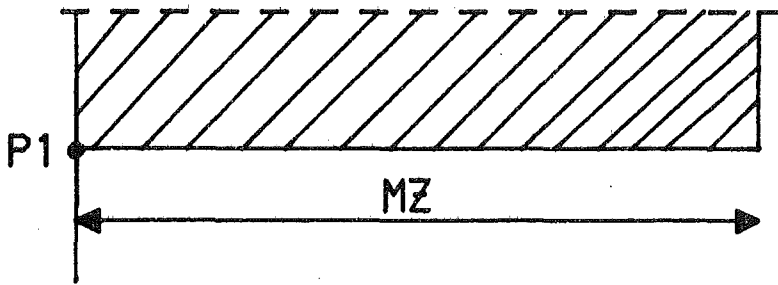
MØ1R(P1 ,MZ ,'L' ,A ,' ' ,Ø.) ;



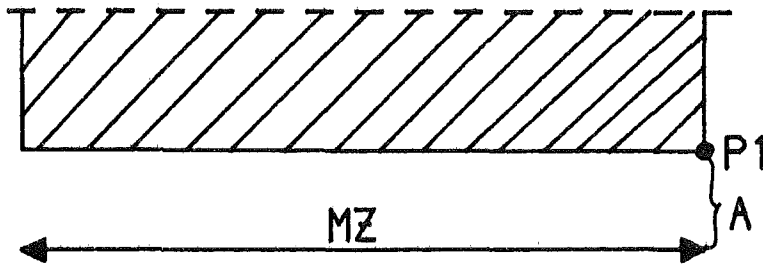
MØ1L(P1 ,MZ ,'R' ,A ,' ' ,Ø.) ;



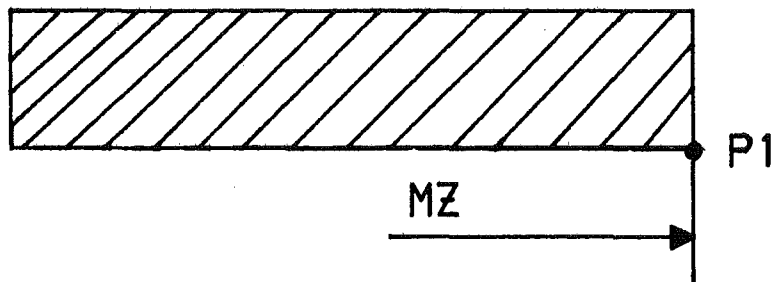
MØ3U(P1 ,MZ , 'L' ,A , ' ' ,Ø.) ;



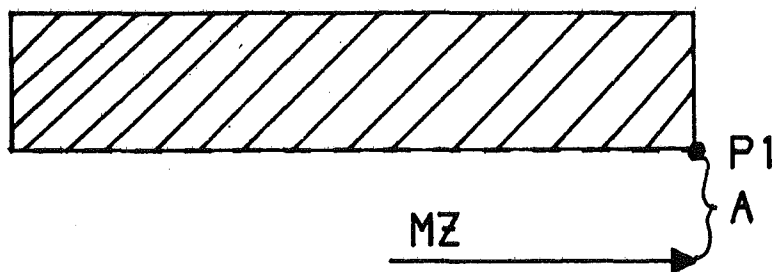
MØ4U(P1 ,MZ , 'R' ,A , ' ' ,Ø.) ;



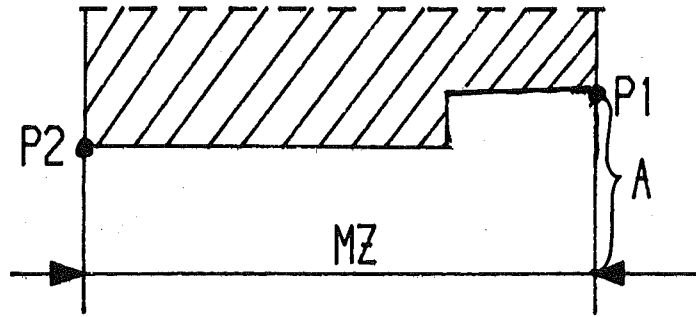
MØ5U(P1 ,MZ , 'L' ,A , ' ' ,Ø.) ;



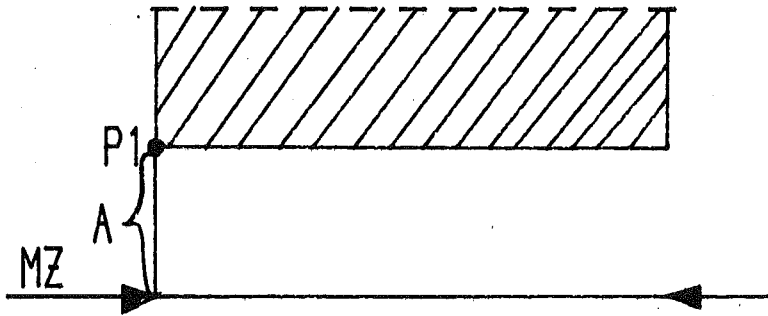
MØ6U(P1 ,MZ , 'R' ,A , ' ' ,Ø.) ;



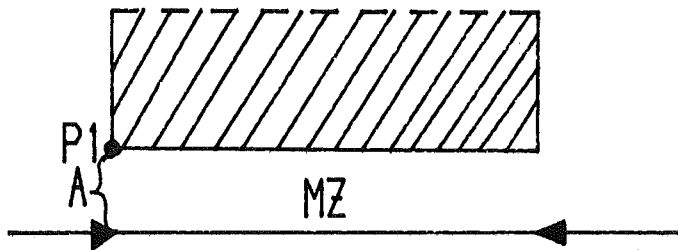
M12U(P1 ,P2 ,MZ ,'R' ,A ,' ' ,Ø.) ;



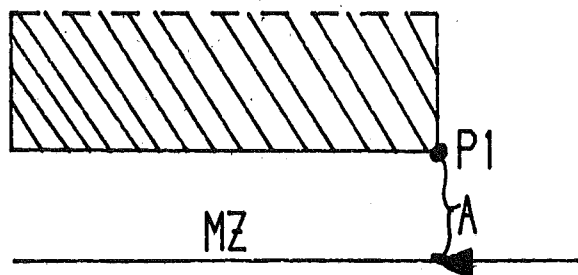
M13U(P1 ,MZ ,'L' ,A ,' ' ,2) ;



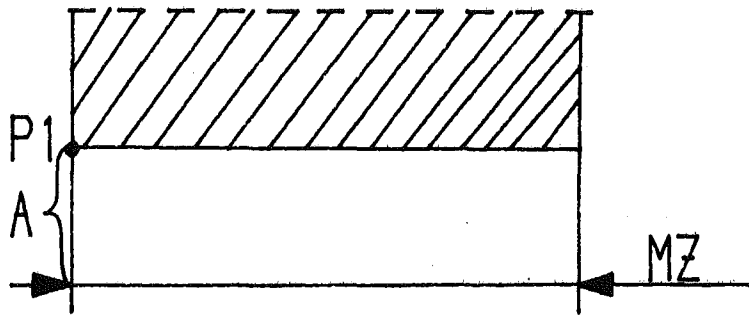
M14U(P1 ,MZ ,'L' ,A ,' ' ,Ø.) ;



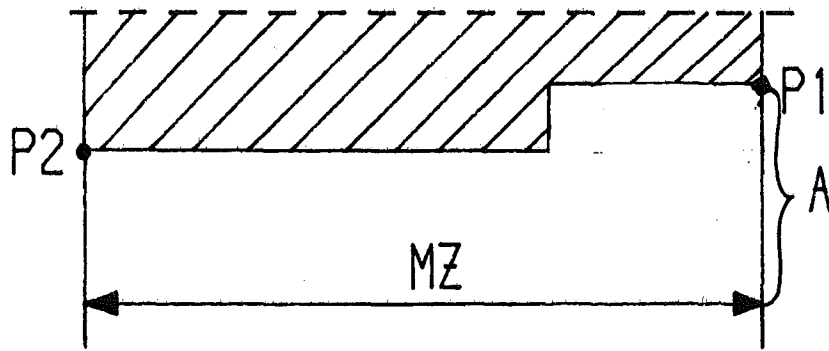
M15U(P1 ,MZ ,'R' ,A ,' ' ,Ø.) ;



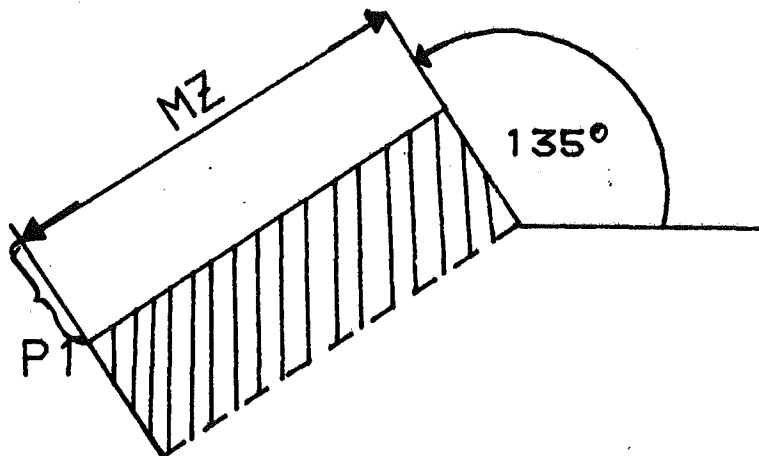
M11U(P1 ,MZ , 'L' , A , ' ' , 1) ;



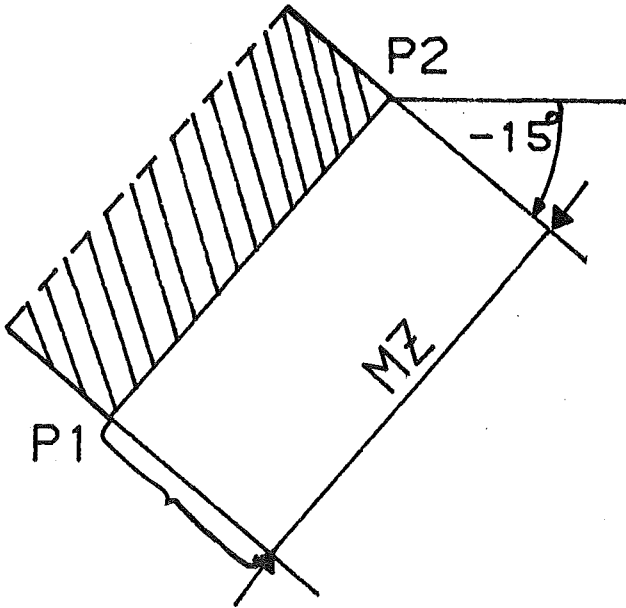
M02U(P1 ,P2 ,MZ , 'R' , A , ' ' , 0.) ;



M01G(P1 ,MZ , 'L' , 135. , A , ' ' , 0.)



M12G (P1 ,P2 ,MZ , 'L' , -15. ,A , ' ' ,Ø.)



Die Bedeutung der Maßtypen MØ1G bis M16G ist analog zu MØ1U bis M16U. Testbeispiele finden sich im alphabetischen Handbuchteil S.

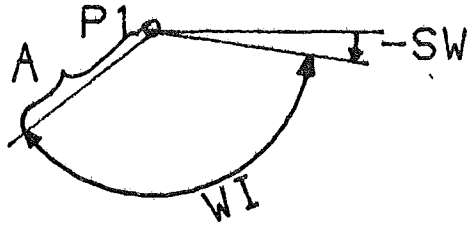
2.3.3 Winkelmaße

Die zwölf Winkelmaßroutinen sind in zwei Gruppen unterteilt. MØ1W bis MØ6W sind Winkelmaße, deren Maßlinie zwischen den Maßhilfslinien-Schenkeln ist. M11W bis M16W sind Routinen für Winkelmaße mit von außen angesetzten Pfeilen.

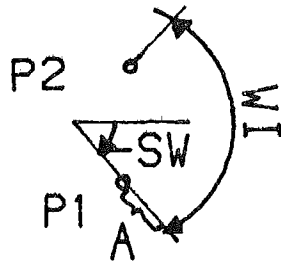
Testbeispiele für die Winkelmaße finden sich im alphabetischen Handbuchteil S. 148 ff.

Skizzen zu den verschiedenen Maßtypen:

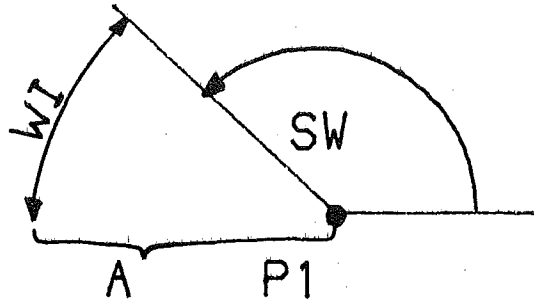
MØ1W (P1 ,WI , 'R' ,SW ,A , ' ' ,Ø.)



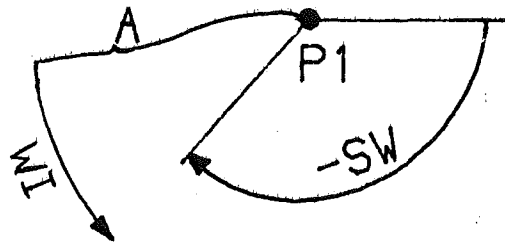
MØ2W (P1 ,P2 ,WI , 'L' ,SW ,A , ' ' ,Ø)



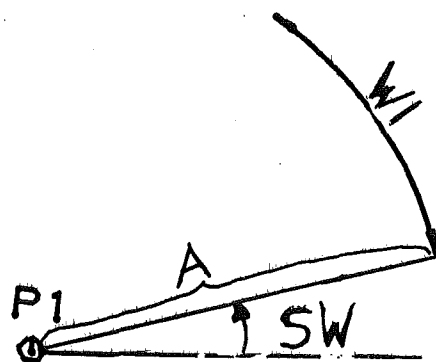
MØ3W (P1 ,WI , 'L' ,SW ,A , ' ' ,Ø.)



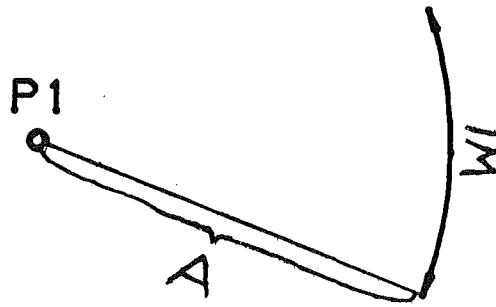
MØ4W (P1 ,WI , 'R' ,SW ,A , ' ' ,Ø.)



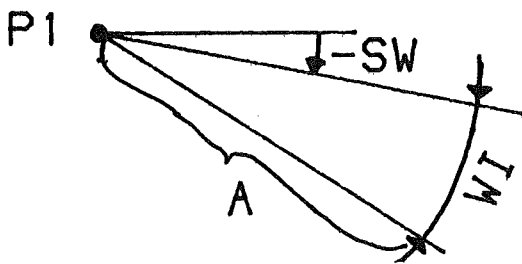
MØ5W (P1 ,WI , 'L' ,SW ,A , ' ' ,Ø.)



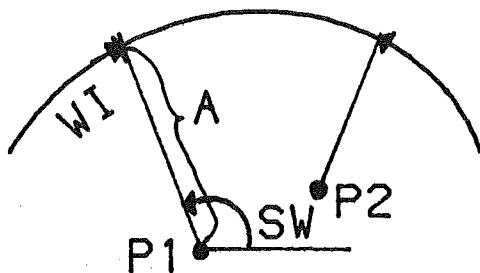
M06W(P1 ,WI , 'R' ,SW ,A , ' ' ,Ø.)



M11W(P1 ,WI , 'R' ,SW ,A , ' ' ,Ø.)



M12W(P1 ,P2 ,WI , 'R' ,SW ,A , ' ' , -1)



M13W bis M16W sind im Aufbau und Bedeutung analog zu M03W bis M06W.

2.3.4 Radianmaße

Bei den Radianmaßen wird unter 2 Typen unterschieden. M1RA, wenn die Lage des Mittelpunktes nicht gekennzeichnet ist (Großbuchstabe R vor Maßzahl) und M2RA mit gekennzeichnetem Mittelpunkt.

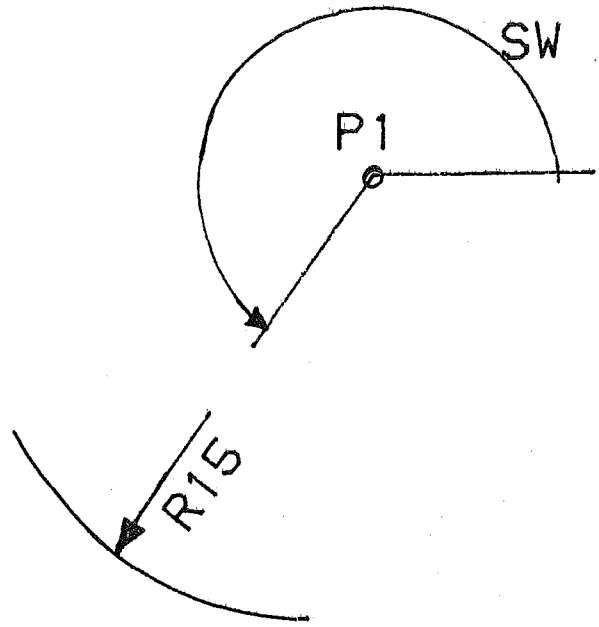
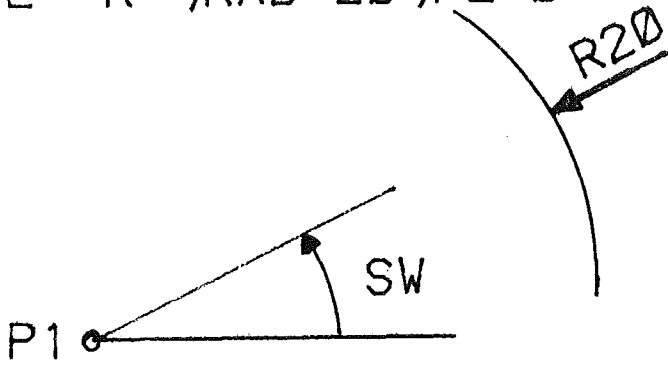
Testbeispiele für Radianmaßroutine siehe im alphabetischen Handbuchteil S.146.

Skizzen der Radienmaße:

M1RA (P1 ,RAD ,L ,SW ,' ' ,PL)

L = 'R' ,RAD=20 ,PL=0

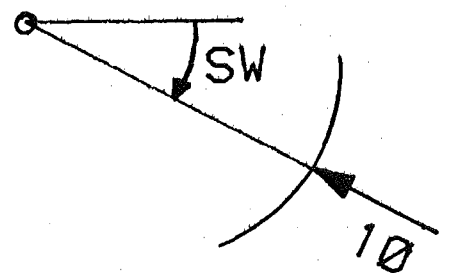
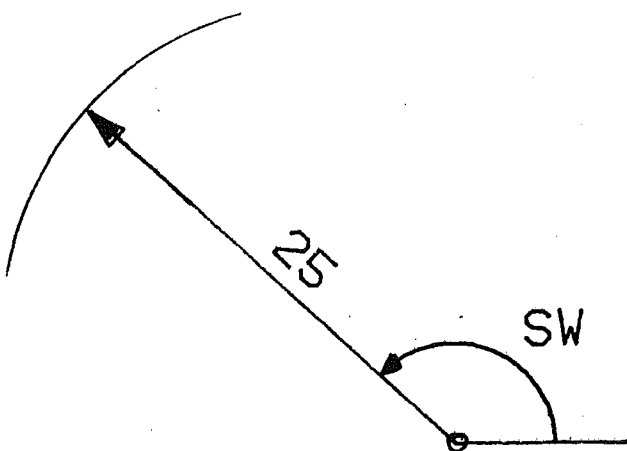
L = 'L' ,RAD=15 ,PL=1



M2RA (P1 ,RAD ,L ,SW ,' ' ,PL)

L = 'L' ,RAD=25 ,PL=0

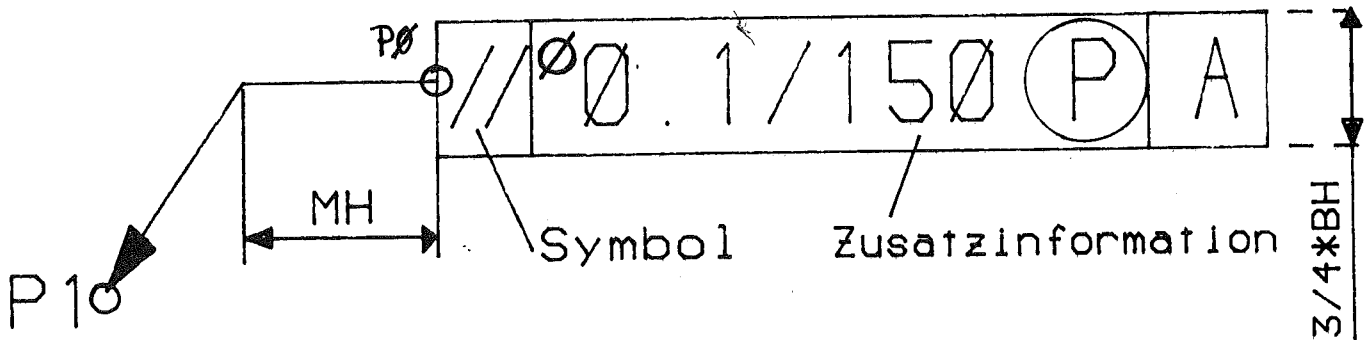
L = 'R' ,RAD=10 ,PL=1



2.3.5 Zusatzroutinen

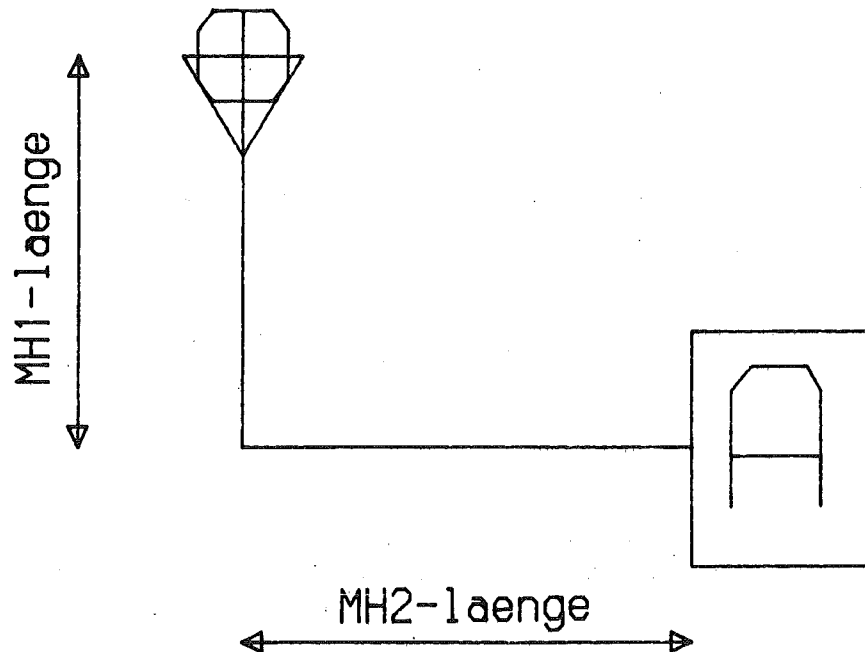
Form- und Lagetoleranzen werden zusätzlich zu Maßtoleranzen angegeben, um Funktion und Austauschbarkeit zu gewährleisten.

MLFT(P1, PØ, [MH-Länge], [Symbol], [Zusatz-Information])



Bezugsbuchstabe für Lage- und Formtoleranzen

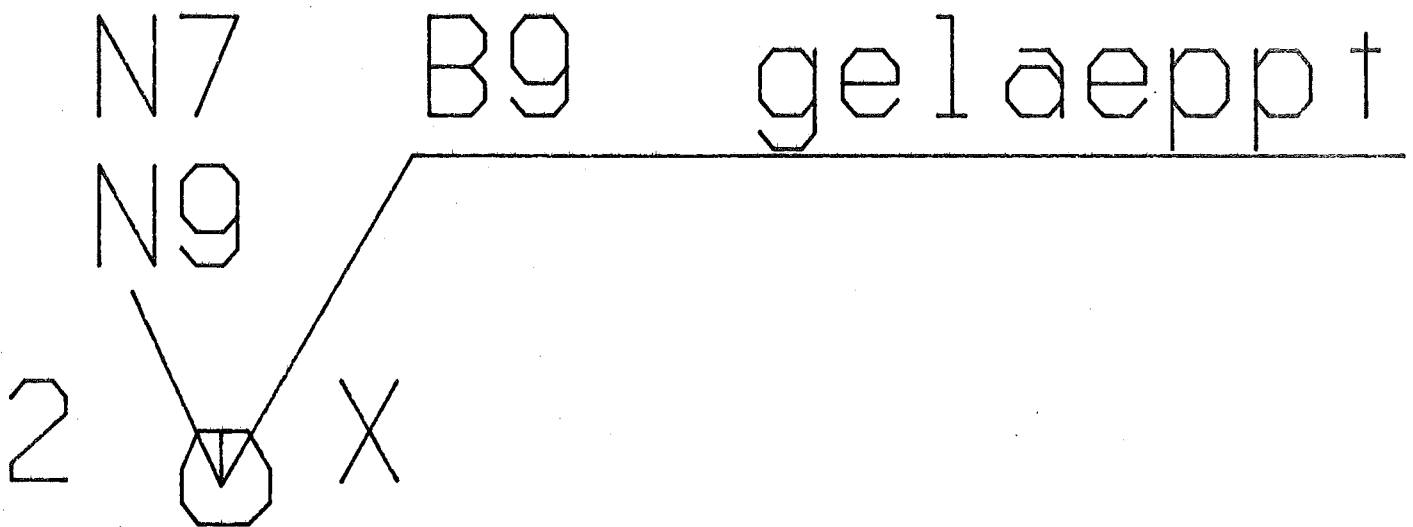
MBEZ(PØ, [Lage], [Winkel], [MH1-Länge], [Bezugsbuchstabe],
MH2-Länge)



Bearbeitungszeichen nach DIN 1302

Diese Routine gilt für Symbole und die zusätzlichen Angaben für die Kennzeichnung der Oberflächenbeschaffenheit in technischen Zeichnungen.

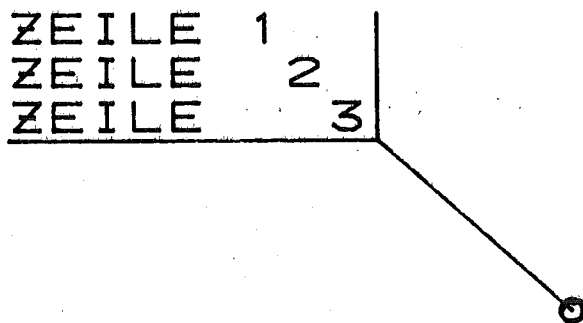
MBAZ(PØ, [Lage], [Zusatz-Information])



Allgemeiner Text

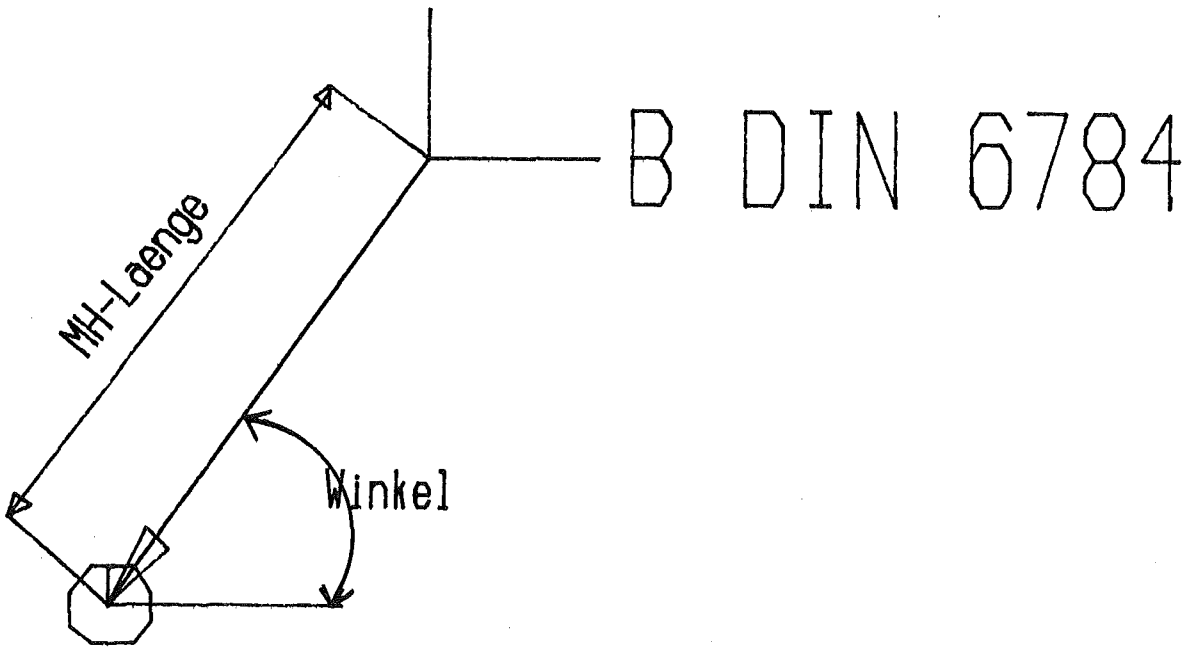
Diese Routine dient zum Plazieren von Text, der auf bestimmte Elemente verweisen soll.

MTEX(P1,P2, [Zusatz-Information])



Werkstückkanten nach DIN 6784

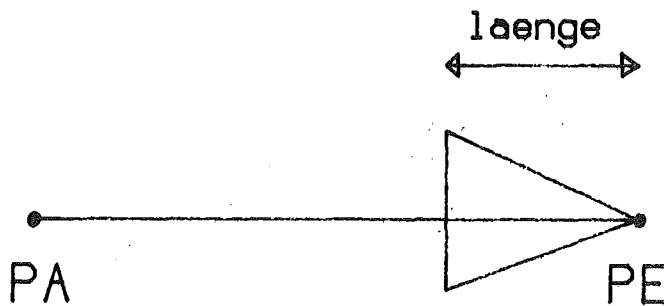
MWSK(PØ, [Lage] , [Winkel] , [MH-Länge] , [Zusatz-Information])



Maßpfeil

Die Routine bildet einen Normpfeil nach untenstehender Abbildung ab.

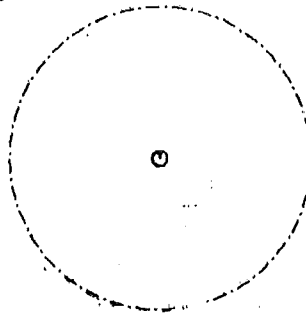
MARW(PA,PE, [Länge])



Einzelheitvergrößerung

Die Routine zeichnet um einen angegebenen Punkt einen Kreis mit angegebenem Durchmesser. Bezugsbuchstabe zur Kennzeichnung der Einzelheit kann entsprechend positioniert werden.

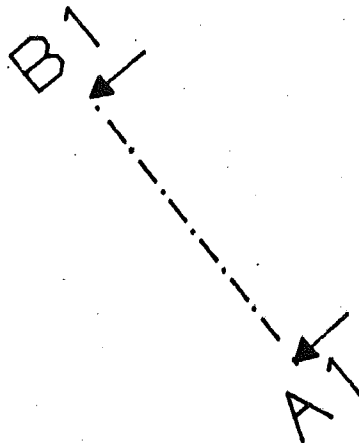
MEZH(PM,R, [Lage] , [Buchstabe]) B



Schnittlinienverlauf

Zur Kennzeichnung eines Schnittverlaufes wird ein Linienzug mit einer Kennzeichnung aufbereitet.

MSVL(PF, [Anzahl] , [Lage] , [Zusatz-Information])



3. Konturen und Schraffuren

Die in diesem Kapitel beschriebenen 2D-Erweiterungen erlauben die problemgerechte Beschreibung zweidimensionaler Flächen, der Kontur solcher Flächen und von Schraffuren innerhalb von Flächen. Rundungen und Fasen können an Eckpunkten einer Kontur angebracht werden. Diese Fähigkeiten sind zur Erzeugung technischer Zeichnungen erforderlich.

3.1 Linienzüge

Ein Linienzug ist ein zweidimensionales graphisches GIPSY-Objekt. Die bestehenden Objekttypen POINT2, POLY2, CIRCLE2, ARC2, AXIS2 und TEXT2 im zweidimensionalen Bereich werden um den neuen Objekttyp

LINE2

erweitert. Ein LINE2-Objekt ist aus geraden Strecken und Kreisbögen aufgebaut, die miteinander verbunden sind. Der letzte Punkt wird mit dem ersten verbunden, so daß ein flächenhaftes Gebilde entsteht (siehe Abb. 3.1). Eine Materialrichtung gibt an, ob beim Durchlaufen der Teilstrecken vom Anfang zum Ende links oder rechts die Fläche mit Material erfüllt ist. Im folgenden werden die Eigenschaften des LINE2-Objektes näher beschrieben. Die zugehörigen Abschnitte des GIPSY-Handbuches sind jeweils angegeben.

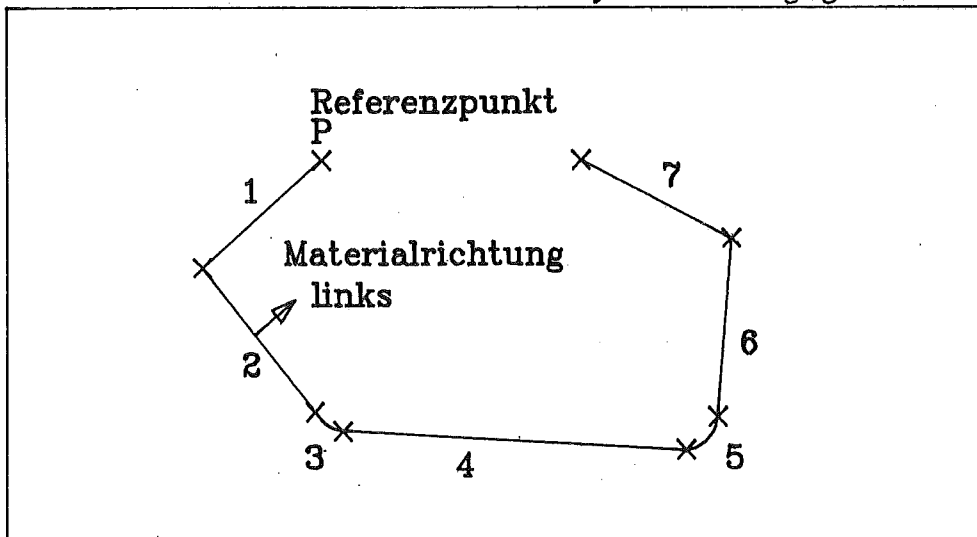


Abb.3.1: Line2-Objekt

Referenzpunkt und Attribute (Kap.4, S.37 GHB)

Referenzpunkt eines LINE2-Objektes ist der erste Punkt, d.h. der Startpunkt der ersten Teilstrecke bzw. des ersten Bogens. Zusätzlich zu den Attributen SYMBOL, LINETYPE, LENGTH, HEIGHT, COLOR, LINEWIDTH, PEN, OPEN/CLOSED und EVERY besitzt das LINE2-Objekt das Attribut

MATERIAL.

Es kann zu MAT abgekürzt werden und gibt die Materialrichtung an. Das Attribut kann durch die EDIT-Anweisung zugewiesen werden.

```
EDIT MAT( { LEFT } ) OF(line2-objekt);
```

Beim Erzeugen einer LINE2 wird die Standard-Materialrichtung benutzt, die durch eine CHANGE STANDARD-Anweisung veränderbar ist:

```
CHANGE STANDARD MAT( { LEFT } );
```

Nach ENTER GIPSY steht der Standardwert auf LEFT.

Deklaration von LINE2: (Kap.4.5.1, S.46 GHB)

```
DCL [ level ] ident [ dim ] LINE2 [ storage class ];
```

Zu beachten ist insbesondere, daß hier, ähnlich wie bei Kollektionen, keine obere Grenze für die Anzahl der Teilobjekte in der Deklaration angegeben werden muß (im Gegensatz zu POLY2).

3.2 Erzeugung von Linienzügen, Konturen und Schraffuren

(Kap.5.1.1, S.58 GHB)

Linienzüge werden durch die LINE2-Funktion erzeugt, Schraffuren durch die SHADE2-Funktion. Die HATCH2-Funktion liefert als ihr Ergebnis eine Kollektion von Linien (POLY2). Die Umrandung einer aus dem Schnitt, der Vereinigung oder der Differenz mehrerer Flächen entstehenden Ergebnisfläche wird durch die CONT2-Funktion ermittelt. Ergebnis dieser Funktion ist eine Kollektion von LINE2-Objekten (siehe Abb.3.2).

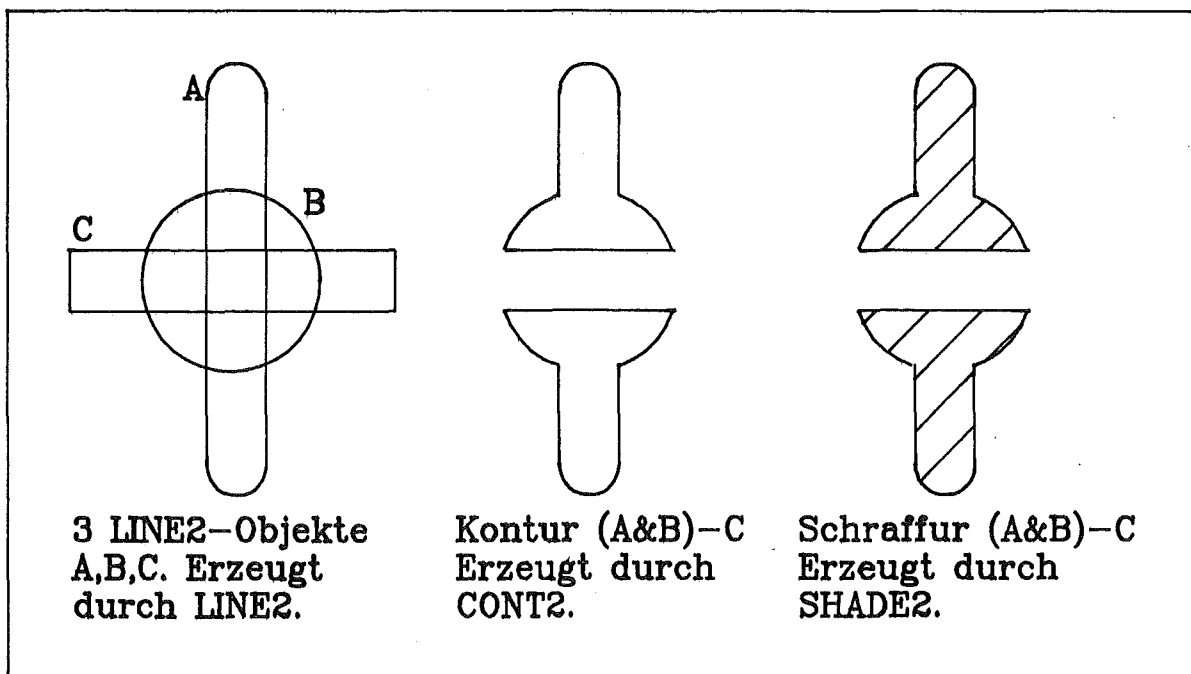


Abb. 3.2: Linienzug, Kontur und Schraffur

Die LINE2-Funktion zum Erzeugen von Linienzügen enthält eine ganze Reihe von verschiedenen Möglichkeiten, den Linienzug zu beschreiben. Dabei wird die Bewegung eines Stiftes (oder eines Werkzeuges) entlang der Kontur nachvollzogen, ähnlich wie es bei einigen NC-Sprachen (Sprachen zur Programmierung numerisch gesteuerter Werkzeugmaschinen) gebräuchlich ist (Beispiel siehe Abb.3.3). Außerdem kann in eine LINE2 eine beliebige Anzahl von

Polygonzügen, Kreisbögen und Punkten aufgenommen werden. Bei Polygonzügen und Kreisbögen ist dabei die Richtung (Reihenfolge vom Anfangs- zum Endpunkt) wichtig, siehe alphabetischen Handbuchteil.

```
SET L=LINE2 (POINT2 (x1,y1),RIGHT(5),UP(3),RIGHT(R) ANGLE(-30),  
            ROUNDT0(A,4)TURN(60),LTAN(C));
```

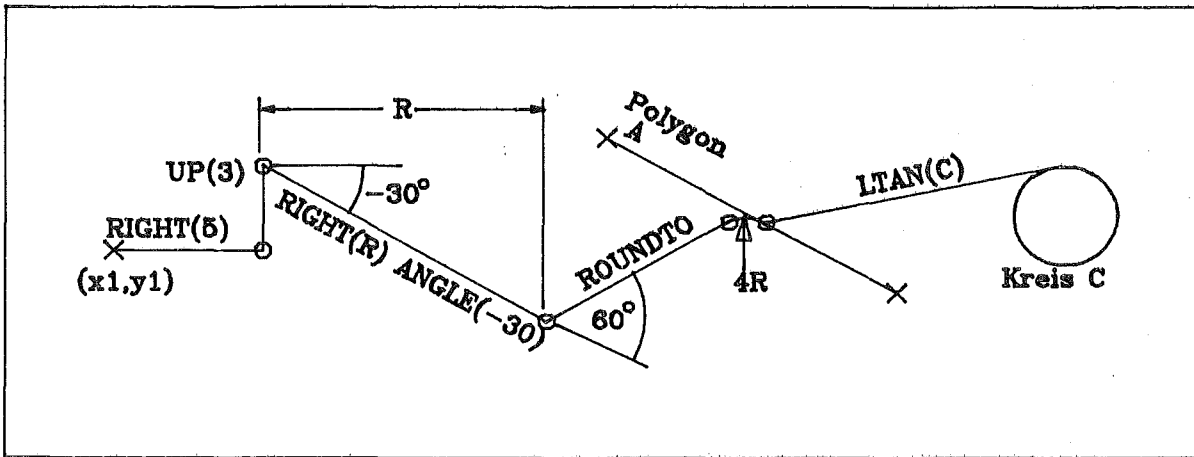


Abb. 3.3: Beispiel für Line2-Operationen

POLY2

Die Funktion POLY2 kann auch auf LINE2-Objekte angewandt werden. Sie wandelt den Linienzug in einen Polygonzug um. Die Kreisbögen werden dabei entsprechend dem durch ARC (siehe S.93 GIPSY-Handbuch) gegebenen Winkelinkrement in Polygonstücke angewandelt.

Objekttransformationen (Kap.5.2.1, S.66 GHB)

Die 2D-Transformationen können alle auch auf LINE2-Objekte angewandt werden. Die Transformation wird nacheinander auf alle Teile (Bögen und Strecken) angewandt.

Sonstige Funktionen (Kap.5.3, S.76 GHB)

LINE2-Objekte können wie andere 2D-Objekte in Kollektionen aufgenommen werden, so daß die Funktionen CARDCOL und NCOLL wie bisher angewandt werden können. Die Funktion OTYP liefert für LINE2-Objekte die Zeichenkette 'LINE2'.

Die COORD-Funktion liefert die Koordinaten des Referenzpunktes, also des ersten Punktes, falls der Parameter "select" nicht angegeben wird. Sonst werden die Koordinaten des "select"-ten Punktes geliefert. Dabei zählen alle Stützpunkte von Strecken mit, bei Bögen jeweils der Anfangs- und der Endpunkt.

(Siehe Beispiel Abb.3.4)

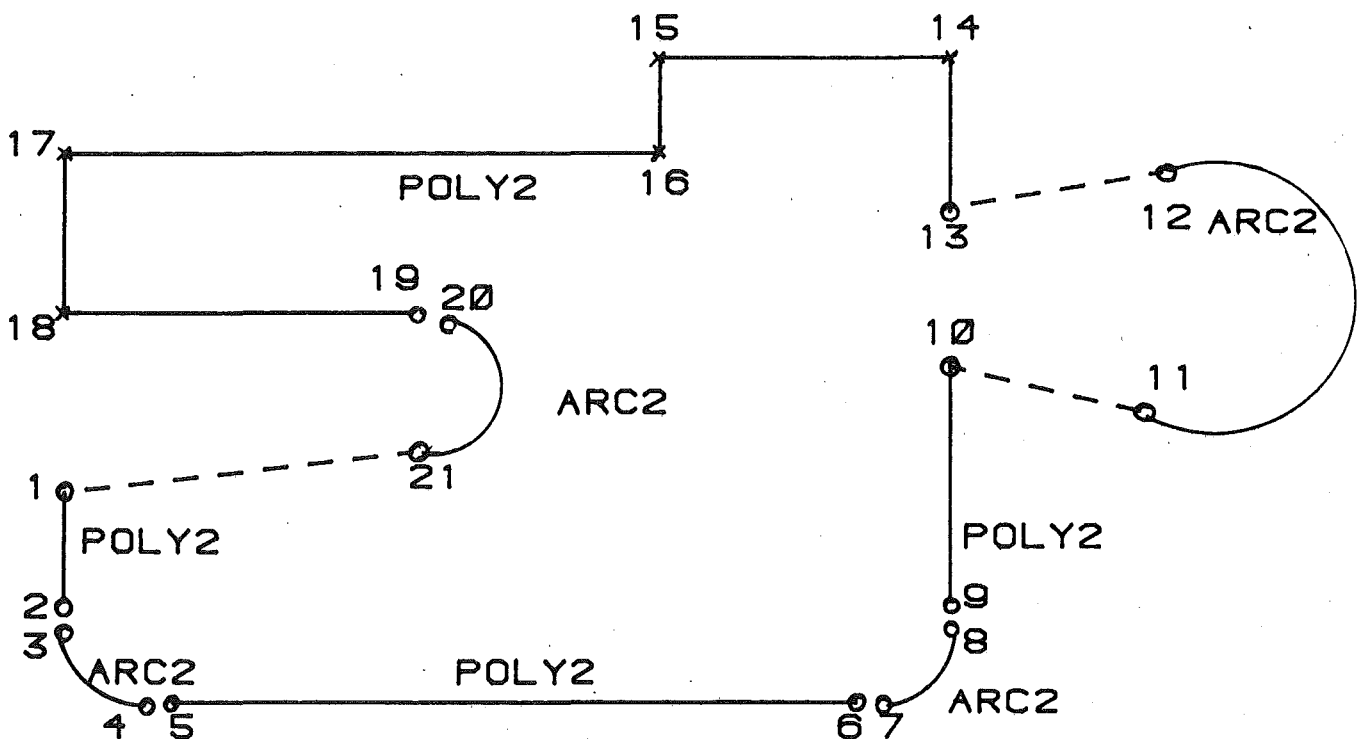


Abb.3.4: Zählung der Punkte eines Linienzuges.

SET COORD (Kap.6.2.2, S.79 GHB)

Die Koordinatenwerte des ersten Punktes oder diejenigen des "select"-ten Punktes können auch für LINE2-Objekte mit dieser Anweisung geändert werden.

PRINT (Kap.6.4, S.86 GHB)

PRINT(LINE2-object) liefert für Linienzüge den Ausdruck der Attributwerte und aller Teilobjekte (Bögen und Strecken). Die Anweisungen SET, PLOT, SAVE, VIEW werden in der üblichen Weise verwendet.

EMPTY (Kap.6.2.5, S.85 GHB)

Wie Kollektionen müssen LINE2-Objekte geleert werden, wenn sie nicht mehr benötigt werden.

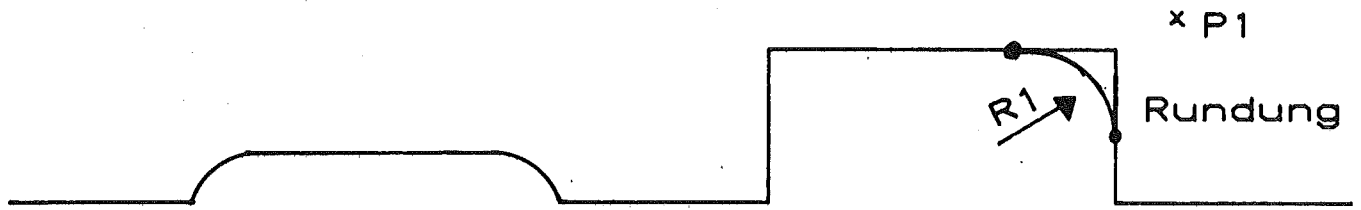
Beispiel:

```
BEGIN
    DCL A LINE2;
        verwende A;
    EMPTY(A);
END;
```

3.3 Rundungen und Fasen

Die Anweisungen ROUND und CHAMFER werden verwendet, um an eine existierende Fläche (LINE2) eine Rundung oder Fase anzubringen. Dabei wird ein Punkt P angegeben, in dessen Nähe die Rundung oder Fase anzubringen ist. Es wird derjenige Endpunkt der Fläche ausgewählt, der am nächsten bei P liegt. Die so gefundene Ecke wird dann durch die Rundung oder Fase ersetzt (siehe Abb.3.5).

ROUND L1 RADIUS(R1) AT(P1);



CHAMFER L2 LENGTH(M2) AT(P2);

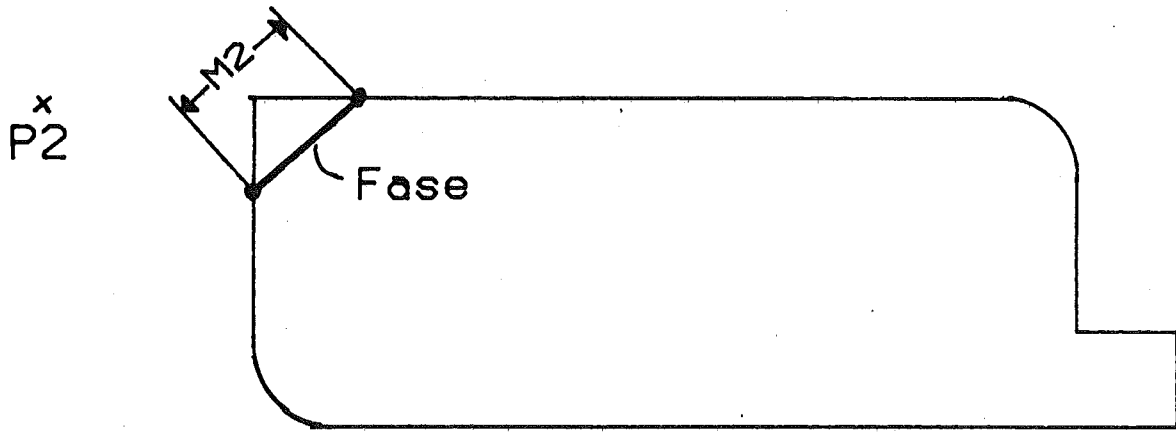


Abb.3.5: Rundung und Fase

4. Einlesen von AGF-Plotfiles (GIPFI)

Mit dem GIPSY-Plotfileinterpretierer (GIPFI) ist es möglich, den Inhalt eines AGF-Plotfiles ganz oder teilweise in GIPFI einzulesen und einer GIPSY-Kollektion zuzuweisen. Es können ganze Plotfilebilder gelesen werden; Teilbilder können fortlaufend gelesen oder übergangen (skip) werden, sie können auch nach ihrem Namen ausgewählt werden.

Das Eröffnen, Einlesen und Schließen eines Plotfiles erfolgt durch eine Reihe von Routinen, die per CALL aus einem GIPSY-Programm heraus aufgerufen werden können. Sie sind nur zwischen "ENTER GIPSY" und "END GIPSY" ansprechbar. Der Plotfile muß in der Job-Control-Language durch eine DD-Karte definiert sein.

4.1 Plotfile-Identifikation und -Aufbau

Beim Öffnen eines Input-Plotfiles wird eine PL/1-Struktur vom Typ BASED allokiert. Diese Struktur übernimmt die Information in den gelesenen Plotfile-Headers, sie enthält den Filenamen und eventuell den Titel, sowie Daten zur Steuerung des Programmablaufs. Sind mehrere Plotfiles geöffnet, dann ist für jeden solch eine Struktur angelegt. Sie tritt im Benutzerhauptprogramm nicht auf. Die Verbindung zwischen Hauptprogramm und Plotfileidentifikationsstruktur erfolgt über eine Pointervariable, die vom Benutzer deklariert wird. Durch Aufruf einer Openroutine wird sie adressiert. Beim Schließen des Files erhält sie schließlich die PL/1-NULL-Adresse.

Gemäß den AGF-Plotfilekonventionen können formatiert und unformatiert dargestellte Daten mit unterschiedlicher Genauigkeit gelesen werden. Sie werden mit einfacher Genauigkeit weitergegeben. Das heißt für Integerzahlen 2 Byte- und für Realzahlen 4 Byte-Wortlänge. Für alle Koordinaten muß die Dimension gleich zwei sein.

4.2 GIPFI-Routinen

Öffnen und Schließen des Plotfiles:

```
CALL PFOPEN(QQ,ptr, ifile);
```

```
CALL PFCLOSE(QQ,ptr);
```

"ptr": Pointer-Variable

"ifile": PL/1-File, einzulesender AGF-Plotfile.

Lesen der Plotfile-Datensätze:

```
CALL PFREAD(QQ, coll, ptr,  
            picture, mode, segment, sequence);
```

coll: GIPSY-Kollektion, in die der Plotfile-Inhalt
gelesen wird

ptr: POINTER-Variable

picture: Bildnummer des Plotfiles

mode: Modus, entscheidet darüber, was wie gelesen wird

segment: Name eines Objektes (Teilbild) auf dem Plotfile

sequence: Anzahl von Teilbildern

Die genaue Bedeutung der Parameter ist im alphabetischen
Handbuchteil S.172 ff beschrieben.

Beispiel:

```
// EXEC REGENT
```

```
//P.SYSIN DD*
```

```
LESE: PROC OPTIONS(MAIN)REGENT;
```

```
ENTER GIPSY;
```

(Fortsetzung siehe nächste Seite)

```
DCL PFOPEN ENTRY(PTR,PTR,FILE)EXT;
DCL PFCLOSE ENTRY(PTR,PTR)EXT;
DCL PFREAD ENTRY(PTR,PTR,PTR, BIN FIXED(15),
    BIN FIXED(15), CHAR(*) VARYING, BIN FIXED(15))EXT;
DCL K1 COLL;
DCL PAGF PTR;
DCL PLOTFILE;
CALL PFOPEN(QQ,PAGF,PLOTFILE);
DO I=1 TO 10;
    CALL PFREAD(QQ,K1,PAGF,I,1,1,0);
    OPEN PLOTFILE DIN A(4);
    PLOT(K1);
END;
CALL PFCLOSE(QQ,PAGF);
END GIPSY;
//G.PLOTFILE DD DSN=....
```

4.3 Fehlermeldungen (Kap.10 GHB)

Die Nachrichtenausgabe und Fehlerbehandlung erfolgt über die REGENT-Nachrichtenverwaltung. Folgende Meldungen können die GIPSY-Plotfileinterpretationsroutinen während der Ausführung des Programms erzeugen:

```
I NO SUBPICTURE WITH NAME.....FOUND
W SUBPICTURE HEADER MISSING
W END OF PLOTFILE REACHED
W NO RASTERGRAPHICS IN GIPSY
W MODE.....IS NOT IMPLEMENTED
E PLOTFILE HEADER MISSING
E PICTURE HEADER MISSING
E INVALID DATA ON PLOTFILE
E PLOTFILE VERSION 1 NOT SUPPORTED
```

5. Anwendung der Hershey-Zeichensätze

Die Hershey-Zeichensätze (Hershey-Fonts) werden als eine Kollektion von 1300 kalligraphischen Schriftzeichen vom amerikanischen National Bureau of Standards vertrieben / 4 /. Die Zeichen gehören zum lateinischen, griechischen und kyrillischen Alphabet. Darüberhinaus gibt es noch etwa 5000 chinesische Schriftzeichen, die aber im vorliegenden Fall nicht benutzt wurden. Eine Auswahl der Zeichen wurde in 14 verschiedene Mengen aufgeteilt, die Zeichensatz (engl.: "font") genannt werden.

Diese Zeichensätze können im graphischen System GIPSY für TEXT- und TEXT2-Objekte verwendet werden und sie sind in allen mit dem AGF-Plotfile verbundenen Programmen verfügbar.

5.1 Die Zeichensätze

Die Zeichensätze werden als simplex, duplex, triplex oder komplex bezeichnet, je nachdem wieviele parallele Linien zur Darstellung eines Zeichensegmentes dienen. Eine weitere Unterscheidung wird vorgenommen in romanische, gotische, griechische, kyrillische Buchstaben sowie in Kursiv- und Schreibschrift. Jedes Alphabet besteht aus vier Komponenten: Großbuchstaben, Kleinbuchstaben, Ziffern und Sonderzeichen.

Die vierzehn Zeichensätze sind:

	Deutsche Bezeichnung	Englische Bezeichnung
1.	Simplex romanisch	Simplex Roman
2.	Duplex romanisch	Duplex Roman
3.	Komplex romanisch	Complex Roman
4.	Triplex romanisch	Triplex Roman
5.	Komplex kursiv	Complex Italic
6.	Triplex kursiv	Triplex Italic
7.	Simplex Schreibschrift	Simplex Script
8.	Komplex Schreibschrift	Complex Script

Deutsche Bezeichnung	Englische Bezeichnung
9. Simplex griechisch	Simplex Greek
10. Komplex griechisch	Complex Greek
11. Englisch gotisch	Gothic English
12. Deutsch gotisch	Gothic German
13. Italienisch gotisch	Gothic Italian
14. Komplex kyrillisch	Complex Cyrillic

Abb.5.1 gibt einen Überblick über die 14 Zeichensätze ohne Sonderzeichen. Die Abbildungen 5.3 bis 5.16 geben eine detaillierte Darstellung der Zeichensätze, zusammen mit der Zuordnung von maschineninterner Zeichendarstellung im EBCDIC-Code und der externen Zeichen-Representation. Bei denjenigen Zeichen, die auf Tastaturen von Bildschirmen und Lochern und auf den Druckern vorhanden sind, wird die EBCDIC-Standardzuordnung benutzt. Lediglich Sonderzeichen und deutsche Umlaute wurden z.T. davon abweichend zugeordnet. Auf den Stellen 0 bis 63 (hexadezimal 00 bis 3F) wurden die von der Plotter-Software und vom GIPSY-Zeichengenerator bekannten Symbole in alle Zeichensätze eingefügt, damit sich die Basisroutinen weitgehend identisch mit der SYMBOL-Routine der Plottersoftware verhalten.

Die Transkription der griechischen und kyrillischen Alphabete ist in Abb.5.17 dargestellt. Das griechische Alphabet enthält weniger, das kyrillische mehr Zeichen als das lateinische. Der GIPSY-Standard-Zeichengenerator erhält die Zeichensatz Nummer 0 (Abb.5.2).

5.2 Ergänzungen in GIPSY

In GIPSY erhielten die TEXT- und TEXT2-Objekte ein weiteres Attribut, FONT, das als Wert die ganzen Zahlen 0 bis 14 annehmen kann. Damit wird einer der Hershey-Zeichensätze 1 bis 14 ausgewählt oder mit 0 der GIPSY-Standard-Zeichensatz angesprochen. Das FONT-Attribut wird wie die anderen Attribute auch

F1	abcdefghijklmnopqrstuvwxy ^z	ABCDEFGHIJKLMN ^{OP} QRSTU ^{VW} XYZ	0123456789
F2	abcdefghijklmnopqrstuvwxy ^z	ABCDEFGHIJKLMN ^{OP} QRSTU ^{VW} XYZ	0123456789
F3	abcdefghijklmnopqrstuvwxy ^z	ABCDEFGHIJKLMN ^{OP} QRSTU ^{VW} XYZ	0123456789
F4	abcdefghijklmnopqrstuvwxy ^z	ABCDEFGHIJKLMN ^{OP} QRSTU ^{VW} XYZ	0123456789
F5	abcdefghijklmnopqrstu ^{vwx} yz	ABCDEFGHIJKLMN ^{OP} QRSTU ^{VW} XYZ	0123456789
F6	abcdefghijklmnopqrstu ^{vwx} yz	ABCDEFGHIJKLMN ^{OP} QRSTU ^{VW} XYZ	0123456789
F7	abcdefghijklmnopqrstu ^{vwx} yz	ABCDEFGHIJKLMN ^{OP} QRSTU ^{VW} XYZ	0123456789
F8	abcdefghijklmnopqrstu ^{vwx} yz	ABCDEFGHIJKLMN ^{OP} QRSTU ^{VW} XYZ	0123456789
Φ9	αβχδεφγηι κλμνωπ ρστυψξυζ	ΑΒΧΔΕΦΓΗΙ ΚΛΜΝΩΠ ΡΣΤΟΤΨΞΘΖ	0123456789
Φ10	αβχδεφγηι κλμνωπ ρστυψξυζ	ΑΒΧΔΕΦΓΗΙ ΚΛΜΝΩΠ ΡΣΤΟΤΨΞΘΖ	0123456789
Φ11	abcdefghijklmnopqrstu ^{vwx} yz	ABCDEFGHIJKLMN ^{OP} QRSTU ^{VW} XYZ	0123456789
Φ12	abcdefghijklmnopqrstu ^{vwx} yz	ABCDEFGHIJKLMN ^{OP} QRSTU ^{VW} XYZ	0123456789
Φ13	abcdefghijklmnopqrstu ^{vwx} yz	ABCDEFGHIJKLMN ^{OP} QRSTU ^{VW} XYZ	0123456789
Φ14	абчдефгхияк ^{лмно} пю ^{рсту} жвхыц	АБЧДЕФГХИЯК ^{ЛМНО} ПЮ ^{РСТУ} ВЖХЫЦ	0123456789

Abb. 5.1: Übersicht über die Hershey-Zeichensätze.

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	☐	⊙	△	+	×	◇	♠	⊗	Z	Y	♠	*	⊗		☆	-
10	{	{	^	≡	→	≠	±	—	—	—	∫	∩	∪	~	≈	
20	}	}	μ	π	φ	θ	ψ	x	ω	λ	∞	δ	ε	γ		
30	Σ	÷	≤	≥	△	□	□	\	↑	√	†	‡	←	×	↑	↓
40											⊙	◦	<	{	+	
50	&										! ◻	\$	*	}	◦ ◻	◻
60	-	/									∞	◻	%	—	>	? ◻
70											◻	#	⊙	∇	=	◻
80		a	b	c	d	e	f	g	h	i						
90		j	k	l	m	n	o	p	q	r						
A0		~	s	t	u	v	w	x	y	z						
B0																
C0	{	A	B	C	D	E	F	G	H	I						
D0	{	J	K	L	M	N	O	P	Q	R						
E0	{		S	T	U	V	W	X	Y	Z						
F0	{	0	1	2	3	4	5	6	7	8	9					
	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F

Abb.5.2: Zuordnung des Zeichensatzes 0, Standard GIPSY

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
00	□	⊙	△	+	×	◇	♣	⌘	Z	Y	⊖	*	⊗		☆	-	0F
10	⌋		^	≡	→		≠	±				∫	∩	v	~	≈	1F
20	⌋	{	μ	π	φ	⊖	ψ	x	ω	λ	α	δ	ε	η			2F
30	∑	÷	≤	≥	△	□]	\	↑	√	†	‡	←	x	↑	↓	3F
40													<)	+		4F
50	&									!	\$	*)		;	?	5F
60	-	/													>	?	6F
70									:	#	@	'			=	=	7F
80		a	b	c	d	e	f	g	h	i							8F
90		j	k	l	m	n	o	p	q	r							9F
A0		o	s	t	u	v	w	x	y	z							AF
B0																	BF
C0		A	B	C	D	E	F	G	H	I	ä	ö	ü	β			CF
D0		J	K	L	M	N	O	P	Q	R							DF
E0			S	T	U	V	W	X	Y	Z	Ä	Ö	Ü				EF
F0	0	1	2	3	4	5	6	7	8	9							FF
	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	

Abb.5.3: Zuordnung des Zeichensatzes 1, simplex romanisch

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	☐	⊙	△	+	×	◇	⋈	⌘	Z	Y	⌘	*	⌘		☆	-
10	}	{	^	≡	→	≠	±	—	—		∫	∩	∪	~	≈	
20	∑	∕	μ	π	φ	θ	ψ	x	ω	λ	α	δ	ε	η		
30	∑	÷	≤	≥	△	□]	\	↑	√	†	‡	←	×	↑	↓
40													<	}	+	
50	&								!	\$	*				:	;
60	-	/							:	,	#	@	'	>	<	?
70									:	,	#	@	'	=	=	"
80		a	b	c	d	e	f	g	h	i						
90		j	k	l	m	n	o	p	q	r						
A0		s	t	u	v	w	x	y	z							
B0																
C0		A	B	C	D	E	F	G	H	I	ä	ö	ü	β		
D0		J	K	L	M	N	O	P	Q	R						
E0			S	T	U	V	W	X	Y	Z	Ä	Ö	Ü			
F0	0	1	2	3	4	5	6	7	8	9						

Abb. 5.4: Zuordnung des Zeichensatzes 2, duplex romanisch

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	□	○	△	+	×	◇	↑	×	Z	Y	⊠	*	⊗		☆	-
10	⌋		^	≡	→	≠	±				∫	∩	∨	~	≈	
20	⌌	{	μ	π	φ	⊖	ψ	x	ω	λ	∞	δ	€	γ		
30	∑	÷	≤	≥	△	□	□	\	↑	√	†	‡	←	×	↑	↓
40													<)	+	
50	&								!		\$	*			;	
60	-	/													>	?
70									:		#	@	'		=	"
80		a	b	c	d	e	f	g	h	i						
90		j	k	l	m	n	o	p	q	r						
A0		o	s	t	u	v	w	x	y	z						
B0																
C0		A	B	C	D	E	F	G	H	I	ä	ö	ü	β		
D0		J	K	L	M	N	O	P	Q	R						
E0			S	T	U	V	W	X	Y	Z	Ä	Ö	Ü			
F0	0	1	2	3	4	5	6	7	8	9						

Abb. 5.5: Zuordnung des Zeichensatzes 3, komplex romanisch

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	□	○	△	+	×	◇	♣	⌘	Z	Y	⌘	*	⌘		☆	-
10	⌋	⌈	^	≡	→	≠	±				∫	∩	∪	~	≈	
20	⌌	{	μ	π	φ	⊖	ψ	x	ω	λ	α	δ	ε	γ		
30	∑	÷	≤	≥	△	□	□	\	↑	√	†	‡	←	×	↑	↓
40													<	}	+	
50	&								!	\$	*				:	
60	-	/													>	?
70									:	,	#	@	'		=	=
80		a	b	c	d	e	f	g	h	i						
90		j	k	l	m	n	o	p	q	r						
A0		ø	s	t	u	v	w	x	y	z						
B0																
C0		A	B	C	D	E	F	G	H	I	ä	ö	ü	β		
D0		J	K	L	M	N	O	P	Q	R						
E0			S	T	U	V	W	X	Y	Z	Ä	Ö	Ü			
F0	0	1	2	3	4	5	6	7	8	9						

Abb. 5.6: Zuordnung des Zeichensatzes 4, triplex romanisch

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
00	☐	⊖	△	+	×	◇	⋈	⌘	Z	Y	⊠	*	⊗		☆	-	0F
10	⌋		^	≡	→	≠	±		—	—		⌋	∩	v	~	≈	1F
20	⌋	{	μ	π	φ	⊖	ψ	x	ω	λ	α	δ	ε	γ			2F
30	∑	⊖	≤	≥	△	□	□	\	↑	√	†	‡	←	x	↑	↓	3F
40													<	(+		4F
50	&									!	\$	*)		:		5F
60	-	/													>	?	6F
70										:	#	@	'		=	"	7F
80		<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>	<i>i</i>							8F
90		<i>j</i>	<i>k</i>	<i>l</i>	<i>m</i>	<i>n</i>	<i>o</i>	<i>p</i>	<i>q</i>	<i>r</i>							9F
A0		<i>s</i>	<i>t</i>	<i>u</i>	<i>v</i>	<i>w</i>	<i>x</i>	<i>y</i>	<i>z</i>								AF
B0																	BF
C0		<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>F</i>	<i>G</i>	<i>H</i>	<i>I</i>	<i>ä</i>	<i>ö</i>	<i>ü</i>	<i>β</i>			CF
D0		<i>J</i>	<i>K</i>	<i>L</i>	<i>M</i>	<i>N</i>	<i>O</i>	<i>P</i>	<i>Q</i>	<i>R</i>							DF
E0		<i>S</i>	<i>T</i>	<i>U</i>	<i>V</i>	<i>W</i>	<i>X</i>	<i>Y</i>	<i>Z</i>	<i>Ä</i>	<i>Ö</i>	<i>Ü</i>					EF
F0	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>							FF
	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	

Abb. 5.7: Zuordnung des Zeichensatzes 5, komplex Kursiv

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	☐	⊙	△	+	×	◇	♠	⌘	Z	Y	⌘	*	⌘		☆	-
10	}	{	^	≡	→	≠	±	—	—	∞	δ	€	γ	~	≈	
20	Σ	÷	≤	≥	△	□	□	\	↑	√	†	‡	←	×	↑	↓
30																
40																
50	&									!	\$	*)	;	+	
60	-	/								,	#	@	'	>	>	?
70										:				=	=	'
80																
90																
A0																
B0																
C0																
D0																
E0																
F0																
	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>	<i>i</i>							
	<i>j</i>	<i>k</i>	<i>l</i>	<i>m</i>	<i>n</i>	<i>o</i>	<i>p</i>	<i>q</i>	<i>r</i>							
	<i>s</i>	<i>t</i>	<i>u</i>	<i>v</i>	<i>w</i>	<i>x</i>	<i>y</i>	<i>z</i>								
	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>F</i>	<i>G</i>	<i>H</i>	<i>I</i>	<i>ä</i>	<i>ö</i>	<i>ü</i>	<i>β</i>			
	<i>J</i>	<i>K</i>	<i>L</i>	<i>M</i>	<i>N</i>	<i>O</i>	<i>P</i>	<i>Q</i>	<i>R</i>							
	<i>S</i>	<i>T</i>	<i>U</i>	<i>V</i>	<i>W</i>	<i>X</i>	<i>Y</i>	<i>Z</i>	<i>Ä</i>	<i>Ö</i>	<i>Ü</i>					
	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>						

Abb. 5.8: Zuordnung des Zeichensatzes 6, triplex Kursiv

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	□	⊙	△	+	×	◇	♣	⊗	Z	Y	♠	*	⊗		☆	-
10	{	{	^	≡	→	≠	±	-	-	-	∫	∩	∪	∩	∪	≈
20	∑	{	μ	π	φ	⊖	ψ	x	ω	λ	α	δ	ε	γ		
30	∑	÷	≤	≥	△	□	□	\	↑	√	†	‡	←	×	↑	↓
40													<)	+	-
50	&								!		\$	*)		,	∧
60	-	/							:		,	#	@	'	≡	∩
70									:		,	#	@	'	≡	∩
80		a	b	c	d	e	f	g	h	i						
90		j	k	l	m	n	o	p	q	r						
A0		s	t	u	v	w	x	y	z							
B0																
C0		A	B	C	D	E	F	G	H	I	ä	ö	ü	β		
D0		J	K	L	M	N	O	P	Q	R						
E0		S	T	U	V	W	X	Y	Z	Ä	Ö	Ü				
F0	0	1	2	3	4	5	6	7	8	9						

Abb. 5.9: Zuordnung des Zeichensatzes 7, simplex Schreibschrift

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
00	□	⊙	△	+	×	◇	♠	⌘	Z	Y	⌘	*	⌘		☆	-	0F
10	}	{	^	≡	→	≠	±	-	-			⌘	⌘	v	~	≈	1F
20			μ	π	φ	⊖	ψ	x	ω	λ	α	δ	ε	γ			2F
30	Σ	÷	≤	≥	△	□	□	\	↑	√	†	‡	←	x	↑	↓	3F
40													<)	+		4F
50	&									!	\$	*)		;	?	5F
60	-	/													>	?	6F
70										:	#	@	'		=	!"	7F
80		a	b	c	d	e	f	g	h	i							8F
90		j	k	l	m	n	o	p	q	r							9F
A0		s	t	u	v	w	x	y	z								AF
B0																	BF
C0		A	B	C	D	E	F	G	H	I	ä	ö	ü	β			CF
D0		J	K	L	M	N	O	P	Q	R							DF
E0		S	T	U	V	W	X	Y	Z	Ä	Ö	Ü					EF
F0		0	1	2	3	4	5	6	7	8	9						FF
	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	

Abb. 5.10: Zuordnung des Zeichensatzes 8, komplex Schreibschrift

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	□	⊖	△	+	×	◇	⋈	⌘	Z	Y	⊠	*	⊗		☆	—
10	⌋		^	≡	→	≠	±		—	—		∩	∪	~	≈	
20	⌋	{	μ	π	φ	⊖	ψ	x	ω	λ	α	δ	ε	η		
30	∑	÷	≤	≥	△	□	□	\	↑	√	†	‡	←	x	↑	↓
40													<	(+	
50	&									!	\$	*)		;	
60	—	/													>	?
70										:	#	@	'		=	=
80		α	β	χ	δ	ε	φ	γ	η	ι						
90			κ	λ	μ	ν	ω	π		ρ						
A0	°	σ	τ	ο	υ	ψ	ξ	ϑ	ς							
B0																
C0		A	B	X	Δ	E	Φ	Γ	H	I						
D0			K	Λ	M	N	Ω	Π		P						
E0			Σ	T	O	Υ	Ψ	Ξ	Θ	Z						
F0	0	1	2	3	4	5	6	7	8	9						

Abb. 5.11: Zuordnung des Zeichensatzes 9, simplex griechisch

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	☐	⊙	△	+	×	◇	⋈	⌘	Ζ	Υ	⊘	*	∑		☆	—
10	}	{	^	≡	→	≠	±	—	—	—	—	∫	⊃	√	~	≈
20	∑	⊖	μ	π	φ	θ	ψ	χ	ω	λ	α	δ	ε	η		
30	∑	⊖	≤	≥	△	□	□	\	↑	√	†	‡	←	×	↑	↓
40													<)	+	
50	&										!	\$	*)	;	?
60	—	/													>	:"
70											:	#	@	'	≡	"
80		α	β	χ	δ	ε	φ	γ	η	ι						
90		κ	λ	μ	ν	ω	π		ρ							
A0	ο	σ	τ	ο	υ	ψ	ξ	υ	ζ							
B0																
C0		Α	Β	Χ	Δ	Ε	Φ	Γ	Η	Ι						
D0		Κ	Λ	Μ	Ν	Ω	Π		Ρ							
E0		Σ	Τ	Ο	Υ	Ψ	Ξ	Θ	Ζ							
F0	0	1	2	3	4	5	6	7	8	9						

Abb. 5.12: Zuordnung des Zeichensatzes 10, komplex griechisch

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	□	○	△	+	×	◇	♠	⌘	Z	Y	♁	*	⊗		☆	-
10	{	{	^	≡	→	≠	±	—	—	—	∩	∪	∩	∪	~	≈
20	∑	∫	μ	π	φ	⊖	ψ	x	ω	λ	α	δ	ε	γ		
30	∑	÷	≤	≥	△	□	□	\	↑	√	†	‡	←	×	↑	↓
40													<	}	+	—
50	&									!	\$	*	}		:	—
60	—	/								:	#	@	'		>	?
70										:	#	@	'		=	?
80		a	b	c	d	e	f	g	h	i						
90		j	k	l	m	n	o	p	q	r						
A0		o	s	t	u	v	w	x	y	z						
B0																
C0		A	B	C	D	E	F	G	H	I	ä	ö	ü	β		
D0		J	K	L	M	N	O	P	Q	R						
E0		S	T	U	V	W	X	Y	Z	Ä	Ö	Ü				
F0	0	1	2	3	4	5	6	7	8	9						

Abb. 5.13: Zuordnung des Zeichensatzes 11, englisch gotisch

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	☐	⊙	△	+	×	◇	♠	⌘	Z	Y	⌘	*	⌘		☆	-
10	}	{	^	≡	→	≠	±	—	—		∫	⊃	v	~	≈	
20			μ	π	φ	⊖	ψ	x	w	λ	α	δ	ε	η		
30	∑	÷	≤	≥	△	□	□	\	↑	√	†	‡	←	x	↑	↓
40													<	}	+	
50	&								!	\$	*				:	
60	-	/													>	?
70									:	#	@	'			=	"
80		a	b	c	d	e	f	g	h	i						
90		j	k	l	m	n	o	p	q	r						
A0		o	s	t	u	v	w	x	y	z						
B0																
C0		U	B	C	D	E	F	G	H	I	ä	ö	ü	β		
D0		S	R	Q	M	N	O	P	Q	R						
E0			G	T	U	B	W	X	Y	Z	ü	ü				
F0	0	1	2	3	4	5	6	7	8	9						

Abb. 5.14: Zuordnung des Zeichensatzes 12, deutsch gotisch

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
00	□	○	△	+	×	◇	♠	⌘	Z	Y	⌘	*	⌘		☆	-	0F
10	{	{	^	≡	→	≠	±	—	—			∫	∩	∨	~	≈	1F
20	∑	∫	μ	π	φ	θ	ψ	χ	ω	λ	α	δ	ε	γ			2F
30	∑	÷	≤	≥	△	□	□	\	↑	√	†	‡	←	×	↑	↓	3F
40													<)	+		4F
50	&									!	\$	*			;	?	5F
60	-	/								:	#	@	'		=	'	6F
70																	7F
80		a	b	c	d	e	f	g	h	i							8F
90		j	k	l	m	n	o	p	q	r							9F
A0		o	s	t	u	v	w	x	y	z							AF
B0																	BF
C0		H	B	C	D	E	F	G	H	I	ä	ö	ü	β			CF
D0		J	K	L	M	N	O	P	Q	R							DF
E0		S	C	U	V	W	X	Y	Z	3	Ä	Ö	Ü				EF
F0	0	1	2	3	4	5	6	7	8	9							FF

Abb. 5.15: Zuordnung des Zeichensatzes 13, italienisch gotisch

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
00	☐	⊙	△	+	×	◇	♠	⌘	Z	Y	⌘	*	⌘		☆	-	0F
10	}	{	^	≡	→	≠	±	-	-	-	⌘	⌘	∨	~	≈		1F
20	}	{	μ	π	φ	⊖	ψ	x	ω	λ	α	δ	ε	η			2F
30	Σ	÷	≤	≥	△	□	□	\	↑	√	†	‡	←	×	↑	↓	3F
40													<)	+		4F
50	&								!	\$	*)	;	:	:	:	5F
60	-	/							:	,	#	@	'	>	?	!"	6F
70									:	#	@	'	=	"	"	"	7F
80		а	б	ч	д	е	ф	г	ш	и							8F
90		я	к	л	м	н	о	п	ю	р							9F
A0		°	с	т	у	ж	в	х	ы	ц							AF
B0																	BF
C0		А	Б	Ч	Д	Е	Ф	Г	Ш	И	З	Ь	Щ	Ъ			CF
D0		Я	К	Л	М	Н	О	П	Ю	Р							DF
E0			С	Т	У	В	Ж	Х	Ы	Ц	З	Ь	Щ				EF
F0	0	1	2	3	4	5	6	7	8	9							FF

Abb. 5.16: Zuordnung des Zeichensatzes 14, komplex kyrillisch

a b c d e f g h i
j k l m n o p q r
s t u v w x y z

A B C D E F G H I ä ö ü
J K L M N O P Q R
S T U V W X Y Z Ä Ö Ü

α β χ δ ε φ γ η ι
κ λ μ ν ω π ρ
σ τ ο υ ψ ξ ϑ ζ

Α Β Χ Δ Ε Φ Γ Η Ι
Κ Λ Μ Ν Ω Π Ρ
Σ Τ Ο Υ Ψ Ξ Θ Ζ

а б ч д е ф г ши
я к л м н о п ю р
с т у ж в х ы ц

А Б Ч Д Е Ф Г Ш И З Ъ Щ
Я К Л М Н О П Ю Р
С Т У В Ж Х Ы Ц З Ъ Щ

Abb.5.17: Transkription des griechischen und kyrillischen Alphabetes

durch die EDIT- und die CHANGE-STANDARD-Anweisung gesteuert. Der Standardwert für FONT ist 0. Bei der Ausgabe auf Zeichengerät oder Bildschirm wird der zugewiesene Zeichensatz zur Erzeugung der Schriftzeichen verwendet. Bei Ausgabe auf den Plotfile (SAVE-Anweisung) wird die Zeichensatz-Nummer mit auf den Plotfile ausgegeben, die Umsetzung in Vektoren erfolgt erst bei der Plotfile-Ausgabe.

Anweisungen:

EDIT FONT(i) OF FOR (text-object);
Setzt die Zeichensatz-Nummer für den Text "text-object" auf i.

CHANGE STANDARD FONT(i);
Setzt die Standard-Zeichensatz-Nummer auf i. Diese Nummer wird nun für alle folgenden TEXT- oder TEXT2-Objekte verwendet.

RESET STANDARD FONT;
Setzt den Standard-Zeichensatz auf 0 zurück.

Beispiel:

```
DCL (T1,T2,T3) TEXT2;  
SET T1 = TEXT2 (POINT2(10,10), 0, 'TEXT1');  
PLOT(T1);                               /*Zeichensatz 0 */  
EDIT FONT(4) OF(T1);  
PLOT(T1);                               /* Zeichensatz 4 */  
CHANGE STANDARD FONT(11);  
SET T2 = TEXT2 (POINT2(10,50), 0, 'TEXT2');  
PLOT(T2);                               /* Zeichensatz 11 */  
RESET STANDARD FONT;  
SET T3 = TEXT2 (POINT2(10,80), 0, 'TEXT3');  
PLOT(T3);                               /* Zeichensatz 0 */
```


5.3 AGF-Plotfile (Kap.7.3 S.136 GHB)

5.3.1 Kodierung der Zeichensatz-Nummer im Plotfile

(Kap.7.3.2 S.150 GHB)

Die Zeichensatz-Information war auch bisher schon im Plotfile-Kennsatz HEAD0046 (Schriftform) vorhanden. Die Font-Nummer hat danach folgende Bedeutung:

- 1: Standard
- 2: Romanisch
- 3: Italic
- 4: Griechisch
- 5: Mathematische Symbole
- >=6: Implementierungsabhängig

Da diese Numerierung den Hershey-Fonts 1-14 nicht direkt entspricht, wurde folgende Zuordnung vorgenommen:

Plotfile-Font = Hershey-Nummer + 100.

Also hat z.B. der Hershey-Font 4 auf dem Plotfile die Nummer 104. Bei der Plotfile-Ausgabe wird diese Konvention natürlich ebenfalls berücksichtigt. Falls ein Plotfile, der z.B. von einer anderen Institution übernommen wurde, die Font-Nummer 1-4 enthält, werden folgende Zeichensätze bei der Ausgabe ausgewählt:

- 1: Standard
- 2: Hershey-Font 3
- 3: Hershey-Font 5
- 4: Hershey-Font 10.

5.3.2 Ausgabe des AGF-Plotfiles

Alle Programme, die den AGF-Plotfile verarbeiten, berücksichtigen jetzt die Zeichensatz-Information und wählen bei der Ausgabe den entsprechenden Font aus. Im einzelnen:

PEPLOT - Ausgabe auf Plotter (Kap.7.3.5 S.170 GHB)

PFTEK14/PFTEK15-Ausgabe auf Tektronix 4014/4015 (Kap.7.3.3 S.155 GHB)

PFMOVIE-Ausgabe auf den Mikروفilm-Plotter bei KFA Jülich (Kap.
7.3.4 S.168 GHB)

6. Interaktive Anwendung von GIPSY (IGIPSY)

Das Programm IGIPSY (Interaktives GIPSY) ermöglicht die interaktive Nutzung der GIPSY-Fähigkeiten an dem graphischen Arbeitsplatz IBM 3277-GA. Der Arbeitsplatz besteht aus einem alphanumerischen (A/N) und einem graphischen Bildschirm, die gemeinsam über einer Leitung an den IBM-Großrechner angeschlossen sind (siehe Abb.6.1). Der Einsatz erfolgt unter TSO.

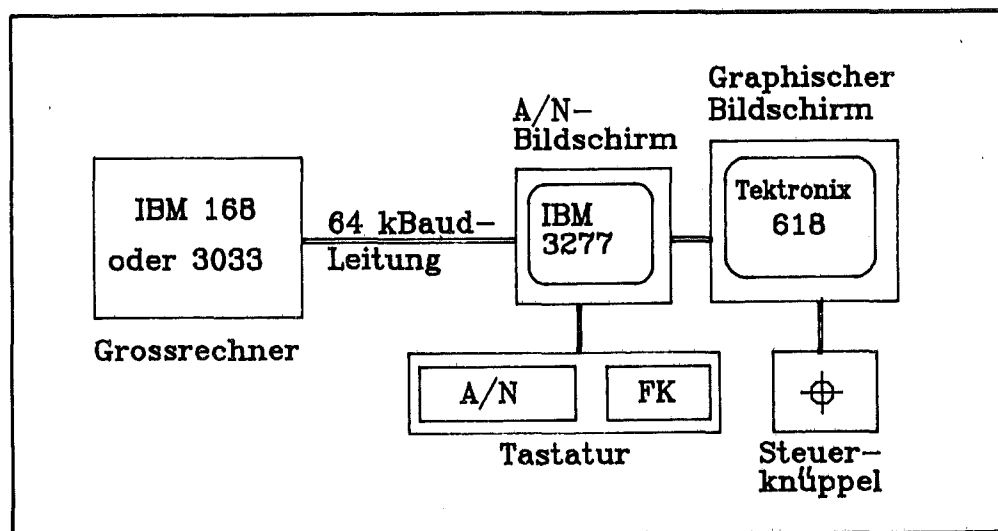


Abb. 6.1: Graphischer Arbeitsplatz IBM 3277-GA

Der A/N-Bildschirm besitzt die für SPF-Terminals übliche Tastatur mit 12 Funktionstasten (FK). Der graphische Bildschirm ist eine Tektronix-Speicherröhre Typ 618 mit einer Größe von 30 mal 40 cm. Zur Eingabe von Koordinatenwerten ist ein Steuerknüppel angeschlossen. Das IGIPSY-Programm nutzt einerseits das REGENT-Subsystem GIPSY und andererseits die IBM-DSM- (Dual Screen Manager-) Software zur Aussteuerung des graphischen Arbeitsplatzes.

Die gegenwärtig verfügbare Version 1 unterstützt folgende Fähigkeiten:

- Einlesen von Bildern vom AGF-Plotfile
- Ausgabe von Bildern auf dem AGF-Plotfile
- Aufbau von Bildern aus Punkten, Polygonzügen, Texten, Kreisen und Kreisbögen

- Einfügen von Bildern vom AGF-Plotfile in das aktuelle Bild
- Verändern von Bildern durch:
 - Löschen von Teilbildern,
 - Verschieben, Drehen oder Skalieren von Teilbildern oder des Gesamtbildes
- Duplizieren eines Bildes oder eines Teilbildes
- Verändern der Abbildungstransformation (Window/Viewport)
- Ausgabe auf eine Hardcopy-Einheit.

6.1 Aufruf des Programmes

Vor dem Aufruf von IGIPSY müssen alle AGF-Plotfiles, mit denen später gearbeitet werden soll, allokiert werden, z.B.

```
ALLOC F(DD1) DA(PLOTFILE.DATA)
```

für einen sequentiellen Dataset, oder:

```
ALLOC F(DD2) DA(PFILE.DATA(BILD1))
```

für ein PDS-Member. Es ist empfehlenswert, neben allen Plotfiles, von denen gelesen werden soll, auch einen Ausgabe-Plotfile und einen temporären Plotfile zum Speichern von Zwischenergebnissen zu allokiieren. Anschließend ist das Programm per CALL aufzurufen:

```
CALL 'REGENT.MODS(IGIPSY)' 'ISA(50K)'
```

Eine Kommandoprozedur mit dem Namen IGIPSY allokiert eine temporäre Datei mit dem DD-Namen DDTEMP und, falls ein Datasetname angegeben wurde, diesen Dataset mit dem DD-Namen DD und ruft anschließend IGIPSY auf:

```
PROC O DSN()
CONTROL MAIN NOMSG
FREE ATTRLIST(ATT1)
FREE F(DDTEMP)
IF &DSN = THEN FREE F(DD)
CONTROL MSG
IF &DSN = THEN ALLOC F(DD) DA(&DSN.) SHR
ATTRIB ATT1 BLKSIZE(3120) LRECL(80) RECFM(F B)
ALLOC F(DDTEMP) NEW SPACE(20 20) TRACK USING(ATT1)
CALL 'REGENT.MODS(IGIPSY)' 'ISA(50K)'
END
```

Das Programm IGIPSY meldet sich mit dem Hauptmenü am A/N-Bildschirm. Für das Arbeiten mit IGIPSY ist eine Region von 1024K erforderlich, die beim LOGON mit dem LOGON-Parameter SIZE(1024) angefordert werden muß.

Anmerkung für die IGIPSY-Anwendung im KfK:

Das Programm befindet sich auf der Bibliothek 'TSO147.REGENT.LOAD', da die REGENT.MODS vom TSO nicht angesprochen werden kann.

Die Kommandoprozedur IGIPSY kann folgendermaßen aufgerufen werden:

IRECMD

IGIPSY oder IGIPSY DSN(datasetname)

Zur Zeit gibt es einen Fehler im TSO, der dazu führt, daß die größere Region nur dann vom TSO richtig verstanden wird, wenn die LOGON-Zeile komplett eingegeben wird, ohne daß LOGON-Parameter durch Prompten nachgefordert werden. Richtig ist also z.B.:

LOGON IRE..../....PROC(.) A(...) SIZE(1024)

6.2 Die Bildstruktur in IGIPSY

Das Programm IGIPSY behandelt Bilder, die in Objekte gegliedert sein können. Jedes Objekt besteht aus einem oder mehreren Grundelementen (Primitives), siehe Abb.6.2.

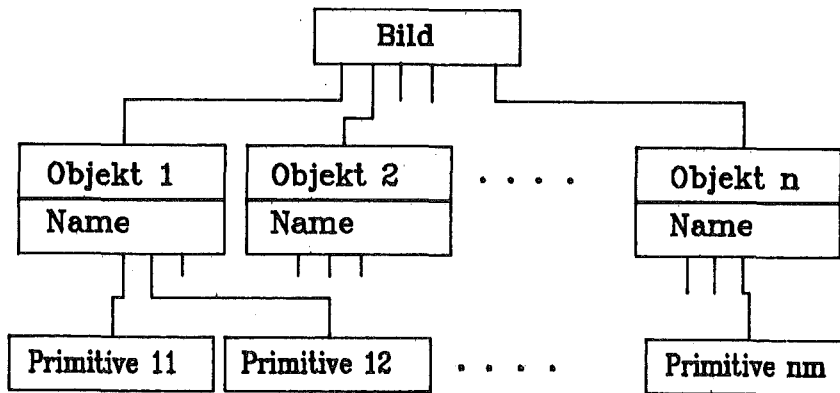


Abb. 6.2: IGIPSY - Bildstruktur

Zu einer Zeit kann nur ein Bild bearbeitet werden. Es wird entweder aus Objekten und Grundelementen am Bildschirm aufgebaut oder von einem Plotfile als ganzes eingelesen. Ein Bild kann auch nur als ganzes auf einen Plotfile ausgegeben werden. Ein Bild kann auch als Einheit transformiert oder gelöscht werden. Bilder können von einem Plotfile in das aktuelle Bild eingefügt werden.

Die Objekte eines Bildes haben Namen mit einer Länge von 8 Zeichen. Diese Namen können beim Bildaufbau angegeben werden (START-OBJECT-Funktion). Beim Einlesen vom Plotfile werden die Objektnamen aus den Teilbildnamen des Plotfiles erzeugt. Z.B. erzeugen die Anweisungen "SAVE; PLOT(KURVE);" in GIPSY den Teilbildnamen "KURVE" auf dem Plotfile und somit den Objektnamen "KURVE" in IGIPSY. Ein Objekt kann selektiv transformiert oder gelöscht werden. Die Objektauswahl erfolgt dabei namentlich oder durch Zeigen auf dem Bildschirm mit Hilfe des Fadenkreuzes. Diese Art der Eingabe wird PICKER-Eingabe genannt.

Stets ist entweder das ganze Bild, ein Objekt oder ein Grundelement die aktuelle Einheit, die gerade manipuliert wird. Z.B. ist nach der Erzeugung eines Grundelementes dieses die aktuelle Einheit, nach START OBJECT ist das neu begonnene Objekt die aktuelle Einheit, nach dem Lesen oder Einfügen eines Bildes von einem Plotfile das eingelesene Bild. Eine nachfolgende Transformation (TRANSLATE, ROTATE, SCALE) oder ein Löschen (DELETE) ohne nähere Spezifikation bezieht sich auf die aktuelle Einheit. Sie wird in den EDIT- und 2D-Menüs unten links auf dem A/N-Bildschirm angezeigt.

6.3 Die IGIPSY-Menüs

IGIPSY wird über Menüs auf dem A/N-Bildschirm bedient. Abb.6.3 zeigt die Menü-Hierarchie. Die gestrichelt gezeichneten Menüs sind noch nicht implementiert.

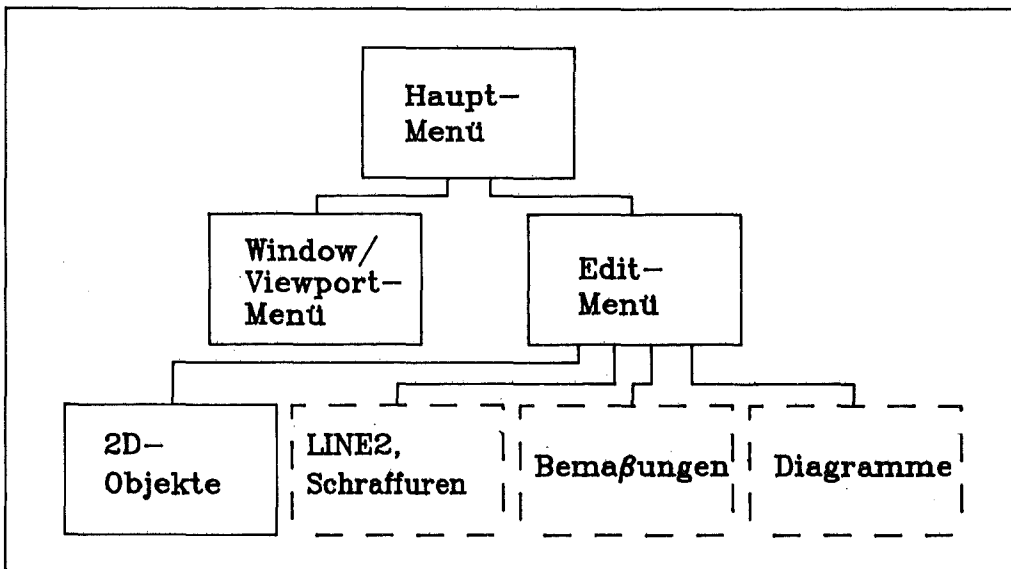


Abb. 6.3: Menü-Hierarchie

Die Abbildungen 6.4 bis 6.7 zeigen den Aufbau der Menüs auf dem A/N-Bildschirm. In den einzelnen Menüs können folgende Funktionen durchgeführt werden:

Hauptmenü:

- | | |
|-----------------|---|
| DELETE PICTURE | - Löschen des Bildes |
| GET PICTURE | - Holen eines Bildes vom Plotfile |
| SAVE PICTURE | - Schreiben eines Bildes auf dem Plotfile |
| EDIT/DRAFTING | - Umschalten in das Menü zum Zeichnen und Editieren von Bildern |
| RENEW PICTURE | - Löschen Bildschirm und Neuzeichnen des Bildes |
| WINDOW/VIEWPORT | - Umschalten in das Window/Viewport-Menü |
| END | - Ende des Programms, zurück ins TSO |

Window/Viewport-Menü:

- | | |
|-----------------|---|
| SET VIEWPORT | - Setzen des Viewports |
| SET WINDOW | - Setzen des Windows |
| RESET VIEWPORT | - Zurücksetzen des Viewports auf Standardwert |
| RESET WINDOW | - Zurücksetzen des Windows auf Standardwert |
| ZOOM IN | - Auf das Bild zufahren |
| ZOOM OUT | - Vom Bild wegfahren |
| UP | - Nach oben fahren |
| DOWN | - Nach unten fahren |
| RIGHT | - Nach rechts fahren |
| LEFT | - Nach links fahren |
| SET PICK RADIUS | - Picker-Radius ändern |
| END | - Zurück ins Hauptmenü |

Edit-Menü

- | | |
|----------------|--|
| START OBJECT | - Beginn neues Objekt |
| INSERT PICTURE | - Einfügen eines Bildes vom Plotfile ins aktuelle Bild |

- DELETE - Löschen des Bildes, eines Objektes oder eines Grundelementes
- COPY - Duplizieren des Bildes, eines Objektes oder eines Grundelementes
- TRANSLATE - Verschieben des Bildes, eines Objektes oder eines Grundelementes
- ROTATE - Verdrehen des Bildes, eines Objektes oder eines Grundelementes
- SCALE - Skalieren des Bildes, eines Objektes oder eines Grundelementes
- RENEW PICTURE - Löschen Bildschirm und Neuzeichnen des Bildes
- END - Zurück ins Hauptmenü

2D - Menü

- DELETE - Löschen des Bildes, eines Objektes oder eines Grundelementes
- TRANSLATE - Verschieben des Bildes, eines Objektes oder eines Grundelementes
- ROTATE - Verdrehen des Bildes, eines Objektes oder eines Grundelementes
- SCALE - Skalieren des Bildes, eines Objektes oder eines Grundelementes
- RENEW PICTURE - Löschen Bildschirm und Neuzeichnen des Bildes
- END - Zurück
- POINT2 - Erzeugen eines Punktes
- POLYGON2 - Erzeugen eines Polygonzuges
- TEXT2 - Erzeugen einer Zeichenkette
- ARC2 - Erzeugen eines Kreisbogens
- CIRCLE2 - Erzeugen eines Kreises
- SWAP SUBMENUES - Umschalten in andere Edit-Submenüs
(Da es zur Zeit nur das eine Edit-Submenü für 2D-Objekte gibt, ist diese Funktion nicht sinnvoll anzuwenden).

Die Auswahl einer Funktion kann grundsätzlich auf 2 Arten erfolgen:

1. Durch Drücken der entsprechenden Funktionstaste. Die Nummern der Tasten stehen in den Menüzeilen links, unten rechts auf dem A/N-Bildschirm ist die Bedeutung der Funktionstaste noch einmal dargestellt. Z.B. hat die Taste Nr.3 stets die Bedeutung "ENDE".
2. Durch Positionieren des A/N-Cursors auf die entsprechende Zeile. Bei manchen Kommandos ist eine Auswahl innerhalb der Zeile möglich, diese Auswahl kann nur durch Positionieren des Cursors erfolgen.

GIPSY VERSION 3.2 - INTERACTIVE DRAFTING SYSTEM

PICTURE-MENUE

COMMANDS:

```
5 DELETE PICTURE
4 GET PICTURE      DDNAME: ?      NO: ?
7 SAVE PICTURE     DDNAME: ?      NO:      1
8 EDIT/DRAFTING
6 RENEW PICTURE
9 WINDOW/VIEWPT
3 END
```

```
|-----|
|FUNCTION KEYS|
|-----|
|HLP      END|
|GET DEL REN|
|SAV EDT WIN|
|-----|
```

Abb. 6.4: Hauptmenü

GIPSY VERSION 3.2 - INTERACTIVE DRAFTING SYSTEM WINDOW-SUBMENU

COMMANDS:

2	SET VIEWPORT	0	4095	0	3071
6	SET WINDOW	0	400	0	300
9	RESET VIEWPORT				
12	RESET WINDOW				
4	ZOOM IN	1			
5	ZOOM OUT	1			
7	UP				
8	DOWN				
10	RIGHT				
11	LEFT				
	SET PCK RADIUS	50			
3	END				

FUNCTION KEYS

HLP SVI END
ZIN ZOT SWN
UP DWN RWI
LFT RGT RVI

Abb. 6.5: Window/Viewport-Menü

GIPSY VERSION 3.2 - INTERACTIVE DRAFTING SYSTEM EDIT-SUBMENU

COMMANDS:

2	START OBJECT	NAME:			
	INSERT PICTURE	DDNAME: ?		NO: ?	
5	DELETE			PIC OBJ: (ACT +)	PRI(ACT +)
4	COPY			PIC OBJ: (ACT +)	PRI(ACT +)
10	TRANSLATE	X	Y	PIC OBJ: (ACT +)	PRI(ACT +)
11	ROTATE	A		PIC OBJ: (ACT +)	PRI(ACT +)
12	SCALE	FX	FY	PIC OBJ: (ACT +)	PRI(ACT +)
6	RENEW PICTURE				
3	END				

SUBMENUES:

7 2D-OBJECTS
8 LINE2/SHADE2
9 DIMENSIONS
DIAGRAMS

FUNCTION KEYS

HLP STA END
COP DEL REN
2D L2 DIM
TRA ROT SCA

EDITING:

OBJECT NAME:

Abb. 6.6: Edit-Menü

GIPSY VERSION 3.2 - INTERACTIVE DRAFTING SYSTEM

2D-SUBMENU

COMMANDS:

5	DELETE			PIC OBJ: (ACT +)	PRI (ACT +)
10	TRANSLATE	X	Y	PIC OBJ: (ACT +)	PRI (ACT +)
11	ROTATE	A		PIC OBJ: (ACT +)	PRI (ACT +)
12	SCALE	FX	FY	PIC OBJ: (ACT +)	PRI (ACT +)
6	RENEW PICTURE				
3	END				
2	POINT2	X	Y	H=5	SYM
4	POLYGON2	X1	Y1	H=5	SYM EVRY LTYP
7	TEXT2	X	Y	H=5	W=5 A=0.0 S=0.0 F=0
	ARC2	+3PTS	+MP/2PTS	+MP/R/2A	
	CIRCLE2	XM	YM	R	3PTS MP/PP

SWAP SUBMENUES:

- 8 LINE2/SHADE2
- DIMENSIONS
- DIAGRAMS

FUNCTION KEYS

HLP PT2 END
PY2 DEL REN
TX2 L2 DIM
TRA ROT SCA

EDITING:

Abb. 6.7: 2D-Menü

Die zweite Bildschirmseite ist für Meldungen und Aufforderungen (Prompts) freigehalten, z.B. "ENTER TEXT STRING" oder "PICTURE DELETED". Das Positionieren des Cursors kann mit den üblichen Tasten ↑, ↓, →, ←, ↵, →|, |← erfolgen. Bei "Tab" (→|) geht der Cursor auf das nächste Eingabefeld des Bildschirms, die Taste "nächste Zeile" (↵) ist wie bei SPF vorteilhaft zur Zeilenauswahl zu benutzen. Die Taste "ERASEEOF" löscht den Rest eines Eingabefeldes (wie im SPF). Wurde irrtümlich durch Drücken der "ERASE INPUT"-Taste oder durch Operateur-Meldungen der Bildschirminhalt zerstört, kann er jederzeit durch die PA2- (CNCL)-Taste wieder restauriert werden.

Koordinateneingabe über Fadenkreuz

Bei Eingabe von Koordinatenwerten mit dem Steuerknüppel erscheint zur Kontrolle das Fadenkreuz auf dem graphischen Bildschirm. Eine Koordinate wird eingegeben, indem mit dem Steuerknüppel das Fadenkreuz an die gewünschte Stelle positioniert wird, und danach eine alphanumerische Taste gedrückt wird (z.B. A, S, D, 5, /, "). Bei Folgen von Eingabewerten, wie bei der Erzeugung von Polygonzügen oder beim Skalieren mit Hilfe des Fadenkreuzes, gilt dabei das "E" (wie ENDE) als Abschlußkennzeichen. Bei einem Polygonzug mit 3 Punkten werden also die beiden ersten Punkte etwa mit dem "A" und der dritte, letzte mit dem "E" akzeptiert.

Werden Koordinatenwerte oder Längen über den A/N-Bildschirm in Form von Zahlen eingegeben, z.B. "100" oder "22.6", dann haben diese Angaben stets die Einheit Millimeter. Dies entspricht der Angabe "CHANGE UNITS LENGTH(MM);" in GIPSY. Die einzige Ausnahme sind die Werte für Viewport und Picker-Radius, die in Bildpunkten angegeben werden.

PICKER-Eingabe

Auch die PICKER-Eingabe (das Zeigen auf ein Objekt auf dem Bildschirm) erfolgt mit dem Steuerknüppel und dem Fadenkreuz. Das gewünschte Objekt wird angefahren und wieder durch Drücken einer A/N-Taste akzeptiert, dabei darf nicht das "E" benutzt werden. Dann blinkt das ausgewählte Objekt, die Auswahl muß durch nochmaliges Drücken quittiert werden. Ist das blinkende Objekt nicht das richtige, so kann durch Drücken von "E" der Pick-Vorgang abgebrochen werden, oder durch Verschieben des Fadenkreuzes und zweimaliges Drücken einer Taste kann ein neuer Pick-Versuch gemacht werden.

Darstellungs-Modi

Der graphische Bildschirm kennt zwei Darstellungsmodi: Gespeicherte Darstellung (Stored Mode) oder bewegliche Darstellung (Refresh-Mode). Gespeicherte Bildteile können nur dadurch geändert werden, daß der ganze Bildschirm gelöscht und danach

das Bild neu gezeichnet wird. Bildteile dagegen, die beweglich dargestellt sind, können selektiv transformiert oder gelöscht werden, ohne den restlichen Bildschirminhalt zu beeinflussen. IGIPSY versucht, stets die aktuelle Einheit im beweglichen Modus darzustellen. Da die Kapazität des Refresh-Speichers jedoch sehr beschränkt ist, kann dieses Konzept oft nicht durchgehalten werden. Es wird dann nur ein Teil der aktuellen Einheit im beweglichen Modus dargestellt, um z.B. bei Transformationen eine Kontrollmöglichkeit zu haben.

Eine HELP-Funktion ähnlich dem SPF-Help wurde vorgesehen, aber in der Testversion noch nicht implementiert. Sie wird stets über die Funktionstaste PF1 angesprochen.

6.4 Die Funktionen

Im folgenden Kapitel werden die einzelnen Funktionen des IGIPSY-Programmes genauer beschrieben:

Hauptmenü:

DELETE PICTURE (PF5) Löscht das ganze Bild. Das Löschen muß durch nochmaliges Drücken der PF5- oder der ENTER-Taste bestätigt werden.

GET PICTURE (PF4) Holt ein Bild vom Plotfile. Dazu muß der Plotfile vor dem IGIPSY-Aufruf allokiert werden. Der DD-Name und die gewünschte Bildnummer sind in die entsprechenden Felder auf dem A/N-Bildschirm einzutragen. Ist der gleiche Plotfile vorher als Ausgabe-Plotfile benutzt worden (in SAVE PICTURE), so wird er abgeschlossen und von vorne eingelesen. Beim Einlesen wird das auf dem Plotfile gespeicherte Window in IGIPSY übernommen. Dabei wird jedoch das Höhe/Breite-Verhältnis gewahrt. Die auf dem Plotfile vorhandenen Objektnamen (z.B. erzeugt durch 'PLOT(object)') werden als Objektnamen in IGIPSY übernommen.

SAVE PICTURE (PF7) Schreibt ein Bild auf einen Plotfile.

Der Plotfile muß vor dem IGIPSY-Aufruf allokiert werden. Der DD-Name ist in das entsprechende Feld auf dem Bildschirm einzutragen. Die Bildnummer wird von IGIPSY eingesetzt, sie fängt bei 1 an und wird bei jedem SAVE eins hochgezählt. D.h. ein Ausgabe-Plotfile wird stets vom Anfang an geschrieben. Wird er zwischendurch geschlossen, indem er mit der GET-Funktion wieder eingelesen wird, oder indem ein anderer Ausgabe-Plotfile zwischenzeitlich benutzt wurde, wird er beim nächsten Mal wieder von vorn überschrieben. Ist der Ausgabe-Plotfile beim SAVE gleichzeitig der aktuelle Eingabe-Plotfile, so wird er geschlossen und von vorn neu beschrieben. Das aktuelle Window wird auf den Plotfile mit ausgegeben.

EDIT/DRAFTING (PF8) Umschalten in das Edit-Menü

RENEW PICTURE (PF6) Der Bildschirm wird gelöscht und das aktuelle Bild wird anschließend im Speicher-Modus neu gezeichnet.

WINDOW/VIEWPORT (PF9) Umschalten in das Window/Viewport-Menü

END (PF3) Ende des Programms, zurück ins TSO

Window/Viewport-Menü

SET VIEWPORT (PF2) Setzen des Viewports. Der Viewport ist ein Rechteckbereich auf dem Bildschirm, in den das Bild gezeichnet wird. Der minimale und maximale x- und y-Wert werden angezeigt und können verändert werden. Die x-Werte müssen zwischen 0 und 4095 und die y-Werte zwischen 0 und 3071 liegen. Die Standardwerte nutzen den vollen Bildschirm für das Bild aus. Steht der A/N-Cursor bei Aufruf der Funktion auf einem der Eingabefelder für die Viewport-Werte, werden diese Werte benutzt. Sonst, oder wenn die Funktion durch Drücken der Funktionstaste 2 ausgewählt wurde, werden die Werte über Steuerknüppel und Fadenkreuz eingelesen, erst die linke untere, dann die rechte obere Ecke. Die neuen Werte werden dann angezeigt.

SET WINDOW (PF6) Setzen des Windows. Das Window ist ein Rechteckbereich im Anwender-Koordinatenraum, der nur durch die Zahlendarstellung der Real-Zahlen begrenzt ist. Durch die Abbildungstransformation wird das Window auf den Viewport abgebildet. Der minimale und maximale x- und y-Wert werden angezeigt und können verändert werden. Die Standardwerte sind 0 bis 40 cm in x-Richtung und 0 bis 30 cm in y-Richtung, entsprechen also der Bildschirmgröße. Dadurch ergibt sich eine maßstäbliche Abbildung des Bildes auf den Bildschirm. Steht der A/N-Cursor bei Aufruf der Funktion auf einem der Eingabefelder für die Window-Werte, werden diese Werte benutzt. Sonst, oder wenn die Funktion durch Drücken der Funktionstaste 6 ausgewählt wurde, werden die Werte über Steuerknüppel und Fadenkreuz eingelesen, erst die linke untere, dann die rechte obere Ecke. Die neuen Werte werden dann angezeigt. Beim Einlesen eines Bildes vom Plotfile (GET PICTURE), jedoch nicht beim Einfügen eines Bildes (INSERT PICTURE) werden die Window-Werte vom Plotfile übernommen, was u.U. zu einer Verzerrung des Bildes führen kann.

RESET VIEWPORT (PF9) Zurücksetzen des Viewports auf den Standardwert (0-4095, 0-3071).

RESET WINDOW (PF12) Zurücksetzen des Windows auf den Standardwert (0-400 mm, 0-300 mm).

ZOOM IN (PF4) Auf das Bild zufahren. Der angezeigte Zoomwert ist das Verhältnis der Standard-Window-Größe zum aktuellen-Window. Ein ZOOM-IN-Wert von 2 bedeutet also die Vergrößerung des Bildes um den Faktor 2. Dies wird durch Verkleinern des Windows erreicht. Beim Drücken von Taste PF4 wird das aktuelle Window jedesmal um den Faktor 1.5 verkleinert. Die neuen Werte von ZOOM-IN, ZOOM-OUT und WINDOW werden angezeigt.

ZOOM OUT (PF5) Vom Bild wegfahren. Der angezeigte Zoomwert ist das Verhältnis der aktuellen Window-Größe zum Standard-Window. Der ZOOM-OUT-Wert ist also der Kehrwert des ZOOM-IN-Wertes. Die beiden Funktionen wurden getrennt, damit mit den beiden Funktionstasten 4 und 5 bequem auf das Bild zu- und von ihm weggefahren werden kann. Ein ZOOM-OUT-Wert von 2 bedeutet also die Verkleinerung des Bildes um den Faktor 2. Dies wird durch Vergrößern des Windows erreicht. Beim Drücken von Taste PF5 wird das aktuelle Window jedesmal um den Faktor 1.5 vergrößert. Die neuen Werte von ZOOM-IN, ZOOM-OUT und WINDOW werden angezeigt.

UP (PF7) Nach oben fahren. Diese Funktion schiebt das Window in die positive y-Richtung, so daß Bildteile über der oberen Bildschirmbegrenzung sichtbar werden. Die Funktion ist somit analog zu der "UP"-Funktion im SPF-Edit. Das Window wird bei jedem Drücken der Taste PF7 um $1/3$ seiner y-Ausdehnung verschoben. Die neuen Window-Werte werden angezeigt.

DOWN (PF8) Nach unten fahren. Diese Funktion schiebt das Window in die negative y-Richtung, so daß Bildteile unter der unteren Bildschirmbegrenzung sichtbar werden. Die Funktion ist somit analog zu der "DOWN"-Funktion im SPF-Edit. Das Window wird bei jedem Drücken der Taste PF8 um $1/3$ seiner y-Ausdehnung verschoben. Die neuen Window-Werte werden angezeigt.

RIGHT (PF10) Nach rechts fahren. Diese Funktion schiebt das Window in die positive x-Richtung, so daß Bildteile rechts vor der rechten Bildschirmbegrenzung sichtbar werden. Das Window wird bei jedem Drücken der Taste PF10 um $1/3$ seiner x-Ausdehnung verschoben. Die neuen Window-Werte werden angezeigt.

LEFT (PF11) Nach links fahren. Diese Funktion schiebt das Window in die negative x-Richtung, so daß Bildteile links von der linken Bildschirmbegrenzung sichtbar werden. Das Window wird bei jedem Drücken der Taste PF10 um 1/3 seiner x-Ausdehnung verschoben. Die neuen Window-Werte werden angezeigt.

SET PICK RADIUS Picker-Radius ändern. Beim Picken mit Steuerknüppel und Fadenkreuz werden alle Objekte identifiziert, deren Abstand vom Fadenkreuz kleiner oder gleich dem Picker-Radius ist. Der Standardwert ist 50 Bildpunkte, also etwa 5 mm. Bei sehr dichten Bildern muß der Wert zur sicheren Unterscheidung verschiedener Bildteile kleiner gewählt werden.

END (PF3) Zurück ins Hauptmenü.

Edit-Menü

START OBJECT (PF2) Beginn neues Objekt. Der Namen des neuen Objektes kann eingetragen werden. Er kann maximal 8 Zeichen lang sein. Der Standardwert für das erste Objekt ist 'OBJ00001'. Die Objektnamen werden beim SAVE PICTURE mit auf den Plotfile ausgegeben.

INSERT PICTURE Einfügen eines Bildes vom Plotfile ins aktuelle Bild. Dazu muß der Plotfile vor dem IGIPSY-Aufruf allokiert werden. Der DD-Name und die gewünschte Bildnummer sind in die entsprechenden Felder auf dem A/N-Bildschirm einzutragen. Das eingefügte Bild wird zur aktuellen Einheit. Ist der gleiche Plotfile vorher als Ausgabe-Plotfile benutzt worden (in SAVE PICTURE), so wird er abgeschlossen und von vorne eingelesen. Wie bei GET PICTURE werden die Objektnamen vom Plotfile in IGIPSY übernommen. Bei mehrfachem INSERT des gleichen Bildes enthält dann das IGIPSY-Objekt die Objektelemente ebenfalls mehrfach.

DELETE (PF5) Löschen des Bildes, eines Objektes oder eines Grundelementes. Die Auswahl der zu löschenden Einheit wird folgendermaßen festgestellt: Steht der A/N-Cursor in der DELETE-Zeile auf "PIC", dann wird das Bild (PICTURE) gelöscht. Steht der Cursor auf "OBJ:(ACT)", dann wird das aktuelle Objekt gelöscht. Steht der Cursor im Feld "OBJ: (ACT+)" auf dem Plus "+", dann wird das zu löschende Objekt über die Picker-Eingabe ausgewählt. Obwohl die Picker-Eingabe als Echo nur das gepickte Grundelement blinken läßt, wird das ganze Objekt gelöscht, zu dem das gepickte Element gehört. Steht der Cursor auf dem "OBJ:....."-Feld und ist dort statt "(ACT+)" ein Name eingetragen, z.B. "OBJ: KURVE", so wird das Objekt mit dem angegebenen Namen gelöscht. Steht der Cursor auf "PRI(ACT", so wird das aktuelle Grundelement gelöscht. Steht der Cursor im Feld "PRI(ACT+)" auf dem Plus "+", dann wird das zu löschende Element über die Picker-Eingabe ausgewählt. Steht aber der Cursor nicht auf einem dieser Felder in der DELETE-Zeile, oder wurde die Funktion durch Drücken der Taste PF5 aufgerufen, dann wird die aktuelle Einheit gelöscht, die unten am Bildschirm, nach "EDITING:" angezeigt wird. Das Löschen wird erst durchgeführt, nachdem es durch nochmaliges Drücken der PF5- oder der ENTER-Taste bestätigt wird.

COPY (PF4) Duplizieren eines Bildes, eines Objektes oder eines Grundelementes. Das ganze Bild, ein Objekt oder ein Grundelement wird dupliziert. Die neue Kopie wird zur aktuellen Einheit und kann durch TRANSLATE, ROTATE oder SCALE transformiert werden. Die Auswahl erfolgt wieder durch Positionieren des Cursors am A/N-Bildschirm, mit der zusätzlichen Möglichkeit der Objektauswahl über die Picker-Eingabe.

TRANSLATE (PF10) Verschieben des Bildes, eines Objektes oder eines Grundelementes. Die Auswahl der zu transformierenden Einheit geschieht wie bei der DELETE-Funktion. Der Verschiebefaktor kann auf zwei Arten angegeben werden:

Über Eingabe in den beiden Feldern "X" und "Y" am A/N-Bildschirm, oder über das Fadenkreuz. Letzteres erscheint dann, wenn die "X"- und "Y"-Felder nicht verändert wurden. Die zu verschiebende Einheit erscheint im beweglichen Modus und kann positioniert werden. Die ausgewählte Einheit wird zur aktuellen Einheit.

ROTATE (PF11) Verdrehen des Bildes, eines Objektes oder eines Grundelementes. Die Auswahl der zu transformierenden Einheit geschieht wie bei der DELETE-Funktion. Der Drehwinkel kann auf zwei Arten angegeben werden: Über Eingabe im Feld "A" am A/N-Bildschirm oder über das Fadenkreuz. Letzteres erscheint dann, wenn das "A"-Feld nicht verändert wurde. Die zu drehende Einheit erscheint im beweglichen Modus. Bei der Fadenkreuz-Eingabe ist nur die x-Koordinate von Bedeutung. Bildschirmmitte bedeutet 0 Grad, ganz rechts -180 Grad und ganz links +180 Grad Drehwinkel. Nach jedem Tastendruck auf eine A/N-Taste wird das veränderte Bild gezeigt. Das Ende der Funktion wird durch die "E"-Taste angezeigt. Die ausgewählte Einheit wird zur aktuellen Einheit.

SCALE (PF12) Skalieren des Bildes, eines Objektes oder eines Grundelementes. Die Auswahl der zu transformierenden Einheit geschieht wie bei der DELETE-Funktion. Die Skalierungsfaktoren können auf zwei Arten angegeben werden: Über Eingabe in den Feldern "FX" und "FY" am A/N-Bildschirm oder über das Fadenkreuz. Letzteres erscheint dann, wenn die Felder "FX" und "FY" nicht verändert wurden. Die zu skalierende Einheit erscheint im beweglichen Modus. Der Abstand des Fadenkreuzes von der Bildschirmmitte ergibt die Skalierungsfaktoren in x- und y-Richtung. Vergrößern heißt also: Fadenkreuz nach rechts oder nach oben; Ein Verfahren des Fadenkreuzes über die Bildmitte hinweg nach links oder unten führt zum Spiegeln der ausgewählten Einheit. Nach jedem Tastendruck auf eine A/N-Taste wird das veränderte Bild gezeigt. Das Ende der Funktion wird durch die "E"-Taste angezeigt. Die ausgewählte Einheit wird zur aktuellen Einheit.

RENEW PICTURE (PF6) Löschen Bildschirm und Neuzeichnen des
Bildes (wie im Hauptmenü).

END (PF3) Zurück ins Hauptmenü.

2D-Menü

DELETE (PF5) Löschen des Bildes, eines Objektes oder
eines Grundelementes (wie im Edit-Menü).

TRANSLATE (PF10) Verschieben des Bildes, eines Objektes
oder eines Grundelementes (wie im Edit-
Menü).

ROTATE (PF11) Verdrehen des Bildes, eines Objektes
oder eines Grundelementes (wie im Edit-
Menü).

SCALE (PF12) Skalieren des Bildes, eines Objektes
oder eines Grundelementes (wie im Edit-
Menü).

RENEW PICTURE (PF6) Löschen Bildschirm und Neuzeichnen des
Bildes (wie im Hauptmenü).

END (PF3) Zurück ins Edit-Menü.

Bei allen Objekt-Eingabe-Operationen im 2D-Submenü (POINT2, POLY2,
TEXT2, CIRCLE2, ARC2) kann die Operation jederzeit mit Funktions-
taste 3 (END-Taste) abgebrochen werden.

POINT2 (PF2) Erzeugen eines Punktes. Die Punktekoor-
dinaten können in den Feldern "X" und "Y" in mm angegeben wer-
den. Sind diese beiden Felder unverändert, erscheint das Faden-
kreuz. Die Symbolgröße kann im Feld "H=5" und die Symbolnummer

im Feld "SYM" angegeben werden, z.B. "12" "7" für Höhe 12 mm und Symbol Nummer 7. Standardwerte sind (Felder unverändert) 5 mm und Nummer 1. Der Punkt wird zur aktuellen Einheit.

POLYGON2 (PF4) Erzeugen eines Polygonzuges. Die Punktekordinaten des ersten Punktes können in den Feldern "X1" und "Y1" in mm angegeben werden. Sind diese beiden Felder unverändert, erscheint das Fadenkreuz. Bei der Fadenkreuz-Eingabe werden solange Punkte angefordert, bis als Quittung des letzten Punktes die "E"-Taste gedrückt wurde. Wurde der erste Punkt in den Feldern "X1" und "Y1" eingegeben, werden die Folgekoordinaten ebenfalls vom A/N-Bildschirm über die Eingabezeile unten am Bildschirm angefordert. Jeweils ein Zahlenpaar, getrennt durch Blank oder Komma, ist einzugeben. Die Werte werden durch die "ENTER"-Taste übermittelt, der letzte Punkt wird statt dessen mit einer beliebigen Funktionstaste quittiert. Die Linienart kann durch Eintragen von "SOL" (SOLID, durchgezogen), "DAS" (DASHED, gestrichelt), "CEN" (CENTER, Mittellinie), "MAR" (MARKED, Kurve mit Symbolen), "DOT" (DOTTED, Symbole ohne Kurve) oder "HAN" (HAND, Freihandlinie) gewählt werden. Für MARKED und DOTTED können noch die Symbole näher beschrieben werden. Im Feld "EVRY" ist das GIPSY- "EVERY"-Attribut einzutragen. Die Symbolgröße kann im Feld "H=5" und die Symbolnummer im Feld "SYM" angegeben werden, z.B. "12" "7" für Höhe 12 mm und Symbol Nummer 7. Standardwerte sind (Felder unverändert) 5 mm, Nummer 1, EVERY=1, SOLID. Der Polygonzug wird zur aktuellen Einheit.

TEXT2 (PF7) Erzeugen einer Zeichenkette. Die Zeichenkette wird über die Eingabezeile unten am Bildschirm angefordert. Die Anfangskordinaten des Textes können in den Feldern "X" und "Y" in mm angegeben werden. Sind diese beiden Felder unverändert, erscheint das Fadenkreuz und der Text kann positioniert werden. Die Zeichengröße kann im Feld "H=5" und "W=5" (Höhe, Breite), der Winkel im Feld "A=0.0"

und die Neigung der Zeichen im Feld "S=0.0" angegeben werden, z.B. "45" , "15" für Winkel 45 Grad und Neigung 15 Grad nach rechts. Im Feld "F=0" kann durch eine Zahl von 0 bis 14 einer der Hershey-Zeichensätze oder der Standard-Zeichengenerator ausgewählt werden. Es ist zu beachten, daß die Zeichensätze zum Teil sehr viele Vektoren erzeugen, die bei IGIPSY von der GA-Software im Arbeitsspeicher gehalten werden. Das kann sehr schnell zu einem fehlerhaften Programmbruch mit der Meldung "INSUFFICIENT SPACE" führen.

Standardwerte sind (Felder unverändert) 5 mm Zeichenhöhe und -breite, Winkel und Neigung 0 Grad und Zeichensatz 0. Der Text wird zur aktuellen Einheit.

ARC2 Erzeugen eines Kreisbogens. Der Kreisbogen kann durch Angabe dreier Punkte, durch Mittelpunkt und 2 Punkte oder durch Mittelpunkt, Radius und zwei Winkel angegeben werden. Die Auswahl erfolgt durch Anfahren des entsprechenden Feldes in der ARC2-Zeile. Wird dabei auf das Plus "+" positioniert, etwa in "+3PTS", dann werden die Punktekoordinaten über das Fadenkreuz eingelesen, sonst werden sie über die Eingabezeile am A/N-Bildschirm angefordert. Der Bogen wird zur aktuellen Einheit.

CIRCLE2 Erzeugen eines Kreises. Der Kreis kann durch Angabe dreier Punkte, durch Mittelpunkt und einen Punkt oder durch Mittelpunkt und Radius angegeben werden. Die Auswahl erfolgt durch Anfahren des entsprechenden Feldes in der CIRCLE2-Zeile. Die Koordinaten des Mittelpunktes und der Radius können in den Feldern "XM", "YM" und "R" in mm angegeben werden. Sind diese Felder unverändert, erscheint das Fadenkreuz. Der Kreis wird zur aktuellen Einheit.

SWAP SUBMENUES Umschalten in andere Edit-Submenüs. Da es zur Zeit nur das eine Edit-Submenü für 2D-Objekte gibt, ist diese Funktion nicht sinnvoll anzuwenden. Später wird sie dazu dienen, von einem Edit-Submenü ins andere zu gelangen, ohne jedesmal den "Umweg" über das Edit-Menü machen zu müssen.

6.5.4 Anwendung des IGIPSY-Version 1

Die jetzt freigegebene IGIPSY-Version 1 ist noch nicht in allen Teilen vollständig und sie enthält sicher auch noch Fehler. Die Meldung von Fehlern, die möglichst genaue Fehlerbeschreibung und, wenn möglich, auch die Angabe, wie der Fehler herbeigeführt werden kann, ist wichtig zum Erreichen einer sicheren, stabilen Version des Programmes. Aber auch Anregungen und Wünsche zum Konzept des Programms, zur Behandlung der Kommunikation und zum Funktionsumfang sind in dieser Testphase erforderlich, so daß die Weiterentwicklung im Sinne der optimalen Deckung der Anwender-Anforderungen erfolgen kann.

7. Erweiterungen der Plotfile-Interpreterer

PFPLOT-Plotfile auf Plotter (Kap.7.3.5 S.170 GHB)

- a) Die Eingabeparameter WIN='1'B, XLUW=..., XROW=..., YROW=... können jetzt benutzt werden. Damit kann ein Bildausschnitt ausgewählt werden (Window).
- b) Der Eingabeparameter CLIPP= '1' '0'B wurde hinzugefügt. Für CLIPP='1'B wird am Window abgeschnitten. CLIPP='1'B ist der Standardwert (dadurch kann die CPU-Zeit ansteigen!).

PFTEK14/15-Plotfile auf Tektronix 4014/4015 (Kap.7.3.2 S.155 GHB)

- a) Neues Kommando PRINT

PRINT $\left\{ \begin{array}{l} \underline{W}INDOW \\ \underline{V}IEWPORT \\ \underline{N}UMBER \\ \underline{E}XTENT \end{array} \right\}$

Druckt das aktuelle Window, den aktuellen Viewport, die aktuelle Bildnummer oder das maximale Window (Extent, wird bei WINDOW MAX oder ALL verwendet) auf den Bildschirm.

- b) Neues Kommando ZOOM

ZOOM f vergrößert bzw. verkleinert das aktuelle Bild um den Faktor f. Im Gegensatz zu SCALE geht ZOOM immer vom aktuellen Bild aus, wirkt also kumulativ.

- c) Neues Kommando FRAME

FRAME ON OFF schaltet das Zeichnen der Bildrahmen (OPEN-PLOT-Rahmen und Viewport) ein oder aus.

- d) Erweiterung des WINDOW-Kommandos

WINDOW CCROSSHAIR erlaubt die Eingabe des Windows mit Hilfe des Fadenkreuzes. Es wird zunächst die linke untere Ecke, dann die rechte obere Ecke angefordert. Das erlaubt die einfache Auswahl von Ausschnitten.

8. Alphabetisches Handbuch (Kap.9 S.219 GHB)

Das Kapitel 8 enthält alle neuen GIPSY-Funktionen, Anweisungen und Objekte in alphabetischer Reihenfolge. Die Syntax und die Bedeutung wird jeweils angegeben. Beispiele verdeutlichen die Anwendung der GIPSY-Funktionen und GIPSY-Anweisungen.

Änderungen an schon bestehenden GIPSY-Funktionen sind durch eine Kennzeichnung am Rand kenntlich gemacht.

```

BOGEN PROC OPTIONS(MAIN) RECENT(INIT,NODA,PLOT-STATOS
DCL PU1 POINT2,
DCL B01 ARC2,
DCL (W11,W12,RAD DEC FLOAT),
OPEN PLOT SIZE(8 CM,8 CM),
SET PU1 = POINT2(30.,10.),
W11 = 0.,
W12 = 135.,
RAD = 25.,
SET B01 = ARC2(PU1,W11,W12,RAD),
PRINT(B01), PLOT(B01),
SET B01 = ARC2(PU1,W12,W11,RAD),
PLOT(SHIFT2(B01,0.,20.)),

SET PU1 = POINT2(20.,5.),
SET PU2 = POINT2(5.,40.),
SET PU3 = POINT2(55.,30.),
/* 2 PUNKT,PUNKT,RAD,CROSSER-,KLEINER BOGEN */

DCL PU2 POINT2,
OPEN PLOT SIZE(8 CM,8 CM),
RAD = 25.,
SET B01 = ARC2(PU1,PU2,RAD,'SMALL'),
PRINT(B01), PLOT(B01),
SET B01 = ARC2(PU1,PU2,RAD,'LARGE'),
PLOT(SHIFT2(B01,0.,10.)),

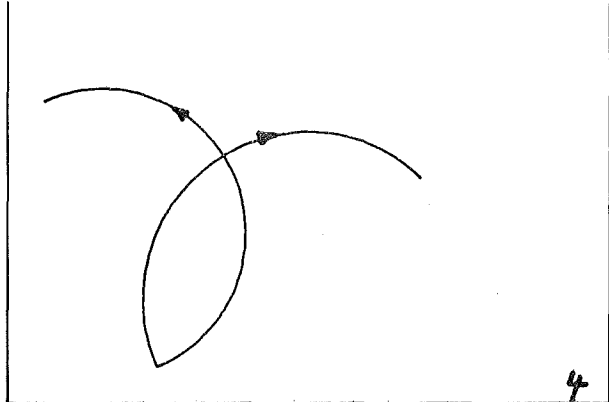
/* 3. PUNKT,PUNKT,PUNKT */

DCL PU3 POINT2,
OPEN PLOT SIZE(8 CM,8 CM),
SET B01 = ARC2(PU1,PU3,PU2),
/* KLEINER BOGEN PU1-PU2 WIRD
GEZEICHNET */
PRINT(B01), PLOT(B01),
SET B01 = ARC2(PU1,PU2,PU3),
PLOT(SHIFT2(B01,0.,10.)),

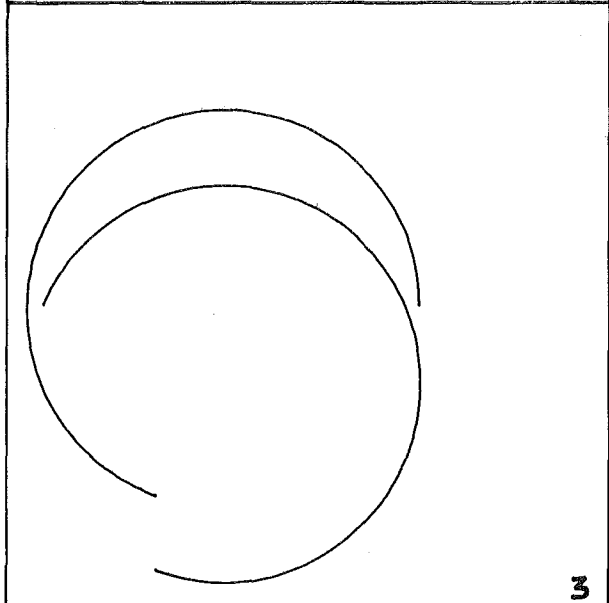
/* 4. PUNKT,PUNKT,BOGENLAENGE. */

DCL BOG DEC FLOAT,
OPEN PLOT SIZE(8 CM,8 CM),
BOG = 60.,
SET B01 = ARC2(PU1,PU2,BOG),
PRINT(B01), PLOT(B01),
SET B01 = ARC2(PU3,PU1,BOG),
PLOT(B01),
END,
END GIPSY,
MESSAGE ACTIVE DEBUG,
FINISH,
END BOGEN,

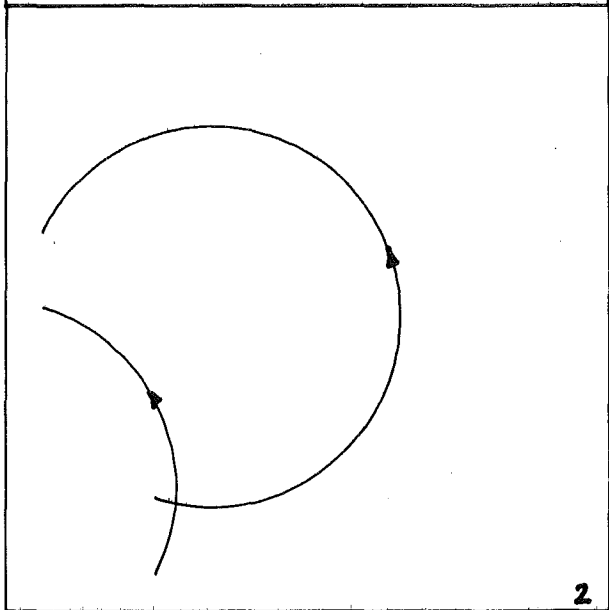
```



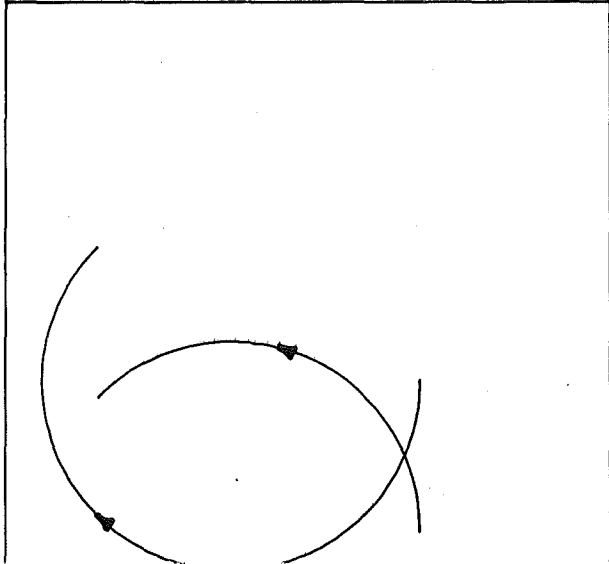
4



3



2



(S.223 GHB)

ARC2
Funktion
Deklaration

Syntax

```
ARC2(point2, avalue, avalue, radex)
ARC2(point2, point2, radex, { 'LARGE'
                             'SMALL' } )
ARC2(point2, point2, point2)
ARC2(point2, point2, lvalue)
```

Erläuterung

Um einem Kreisbogen die bestimmenden Attribute zuzuweisen, sind hier 4 Möglichkeiten gegeben:

- 1) point2 ist der Mittelpunkt des Kreisbogens, sein Radius ist durch radex gegeben, seine Länge durch die Winkelangaben. Der Kreisbogen wird von der ersten zur zweiten Winkelangabe im mathematischen positiven Sinn durchlaufen, falls radex positiv ist und im negativen Sinn für radex < 0.
- 2) Der Kreisbogen ist definiert durch die Angabe von Anfangs- und Endpunkt, den Bogenradius sowie durch die Angabe, ob das kleinere Bogenstück ('SMALL') oder das größere ('LARGE') gemeint ist. Die Krümmung muß wieder, wenn man von Punkt 1 zu Punkt 2 geht, mathematisch positiv sein, falls radex positiv ist, und negativ für radex < 0. Winkel werden in Grad (DEG) oder Radian (RAD) angegeben.
- 3) Es wird ein Kreisbogen erzeugt, ausgehend von Punkt 1 über Punkt 2 nach Punkt 3.
- 4) Ein Kreisbogen wird durch Anfangs- und Endpunkt und seine Länge bestimmt. Die Krümmung ist wieder positiv für lvalue positiv und negativ, falls lvalue negativ ist.

Deklaration

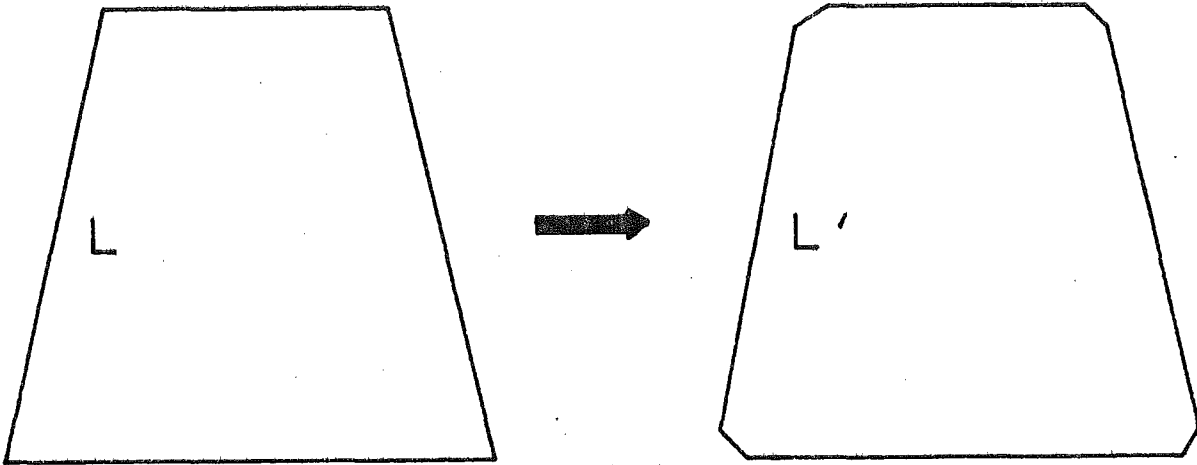
```
DCL [level] ident [dim] ARC2
    [storage class];
```

Siehe DCL.

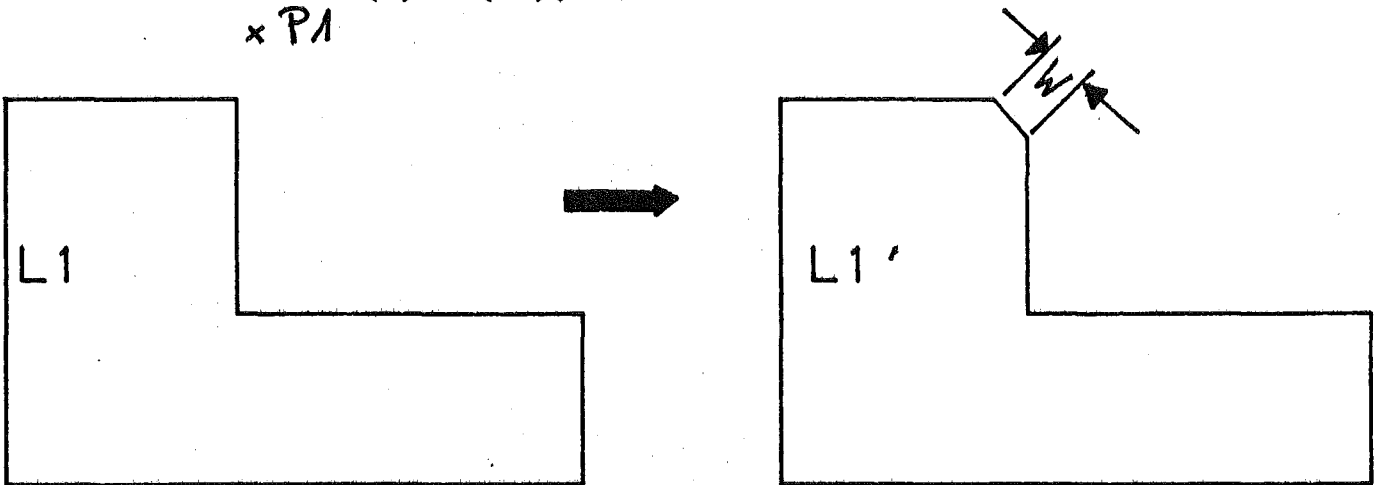
Beispiele:

DCL(L,L1) LINE2;

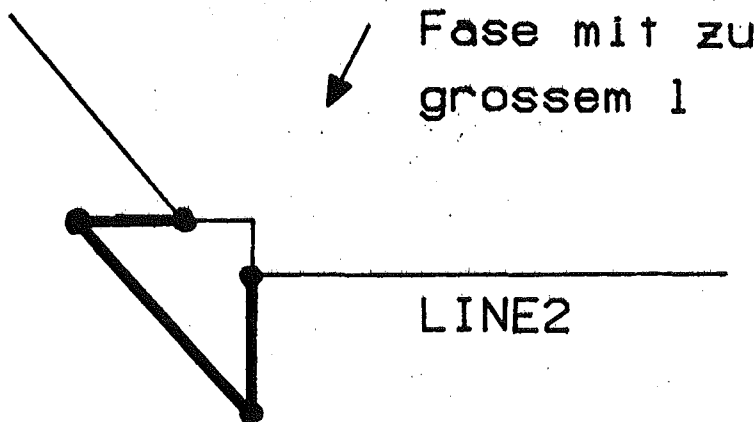
CHAM L LENGTH(3 MM);



CHAM L1 WIDTH(w) AT(P1);
x P1



Achtung! Falsche Anwendung führt evtl. zu unerwarteten Resultaten:



CHAMFER

Statement

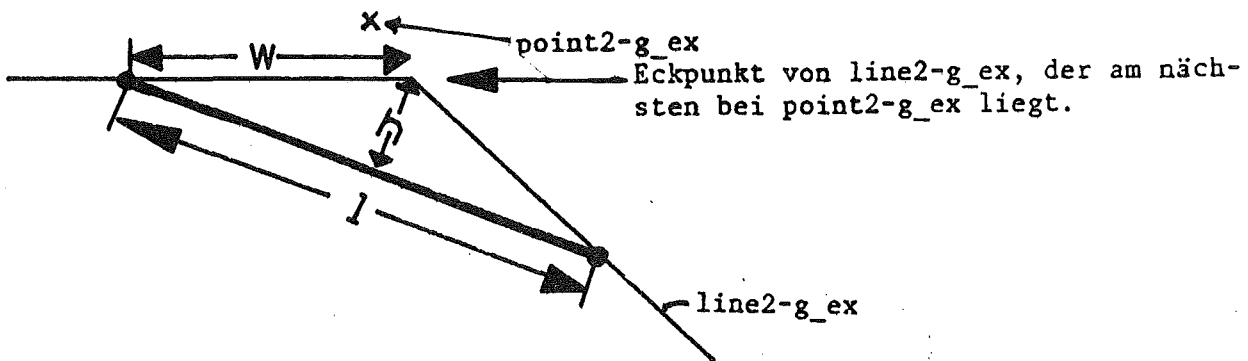
Syntax

$$\left\{ \begin{array}{l} \text{CHAMFER} \\ \text{CHAM} \end{array} \right\} \text{line2-g_ex} \left\{ \begin{array}{l} \text{LENGTH} \quad (l) \\ \text{HEIGHT} \quad (h) \\ \text{WIDTH} \quad (w) \end{array} \right\} \left[\left\{ \begin{array}{l} \text{NEAR} \\ \text{AT} \end{array} \right\} \text{point2-g_ex} \right];$$

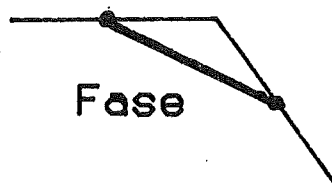
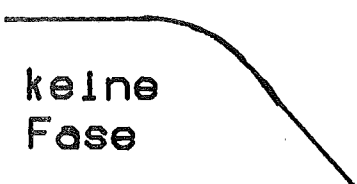
Erläuterung

An demjenigen Eckpunkt des Objektes "line2-g_ex", der am nächsten bei "point2-g_ex" liegt, wird eine Fase angebracht. Bei gleichem Abstand wird der erste Punkt ausgewählt. Die Fase hat an beiden Seiten gleiche Winkel, so daß sie zusammen mit den ursprünglichen Seitenlinien ein gleichschenkeliges Dreieck bildet. Wird point2-g_ex nicht angegeben, werden alle Ecken abgeschragt.

- l : lvalue, Länge der Fase
- h : lvalue, Höhe der Fase
- w : lvalue, Schenkellänge des durch die Fase abgeschnittenen gleichschenkeligen Dreiecks.



Eine Ecke ist der Berührungspunkt zweier gerader Linienstücke:



CHANGE

Beispiele

```
CHANGE MASFAC(0.5),
      STANDARD FONT(2),
      UNITS LENGTH(MM),
      PROJECTION PARALLEL(0,0,1),
      PROJECTION PI_N(0,0,1),
      INVISIBLE LINETYPE(DASHED),
      STANDARD PEN(5),
      INVISIBLE PEN(4),
      MAXPEN(8),
      ARC(5GRAD);
CHANGE UNITS ANGLE(RADIAN);
CHANGE INCREMENT LINEAR(2CM),
      INCREMENT ANGLE(10DEG);
CHANGE STANDARD MAT(RIGHT);
```

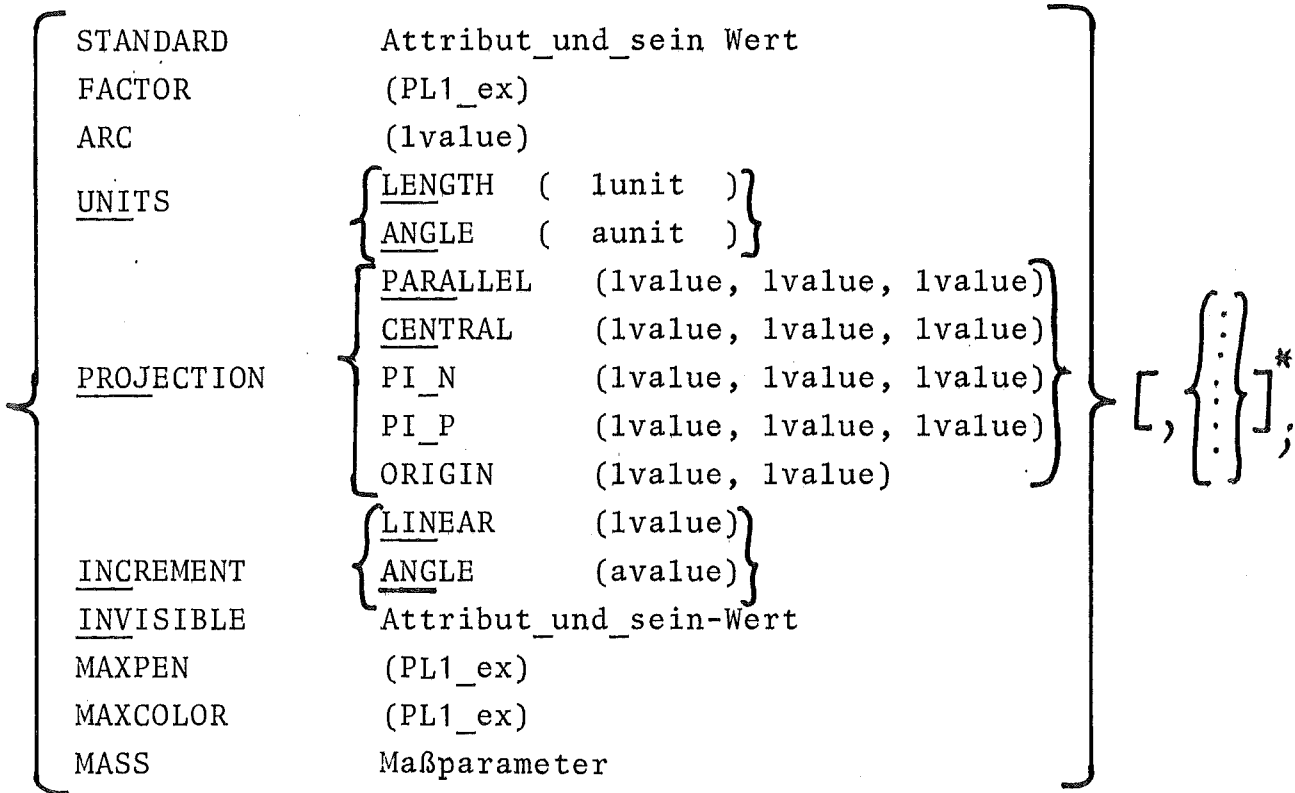
(S.235 GHB)

CHANGE

Statement

Syntax

CHANGE



Erläuterung

CHANGE ändert die Systemvariablen für die

- Standardeinheiten (Längen-, Winkelmaß)
- Projektionsart (parallel, zentral,....)
- Inkremente bei Kurvenberechnung
- Standardattribute (Linientyp, Farbe, Texthöhe,....)
- Skalierung (FACTOR)
- Umwandlung von Kreisen und Bögen in Polygone,
- Parameter für Maßroutinen (MASS)

Standard: siehe CHANGE STANDARD

UNITS: " " UNITS

PROJEKTION: " " PROJECTION

INVISIBLE: " " INVISIBLE

ARC,FACTOR,INCREMENT,MAXPEN,MAXCOLOR:siehe CHANGE ARC

MASS: siehe CHANGE MASS

Beispiele

CHANGE MASS MASSTAB(0.5);

CHANGE MASS TRACE(1);

CHANGE MASS BH(2.5);

CHANGE

MASS

Statement

Syntax

CHANGE MASS {
 MASFAC(m)
 MASTAB (m)
 BH (h)
 TRACE (t)
 MTRACE (t)

m: PL1_ex, Maßstabsfaktor, Standardwert 1

h: PL1_Ex, Schrifthöhe der Bemaßung, in cm.
Standardwert: 0.35 cm.

t: PL1_ex, Trace-Steuerparameter,
0,1,2,3, Standardwert 0 (kein Trace)

Es gibt 3 Variable, die durch das Change Statement explizit vom Benutzer gesetzt werden können bzw. durch RESET MASS auf ihren Standardwert zurückgesetzt werden.

Diese sind:

1. MASFAC bzw. MASTAB
2. TRACE bzw. MTRACE
3. BH

MASFAC (Abkürzung MASF) verändert die Zeichnungskontur um den gewünschten Faktor. Skaliert werden die erforderlichen Referenzpunkte und Maßlinienpunkte, nicht dagegen die Höhe der Beschriftung und die Länge der Maßhilfslinien.

Mit TRACE bzw. MTRACE läßt sich Testoutput erzeugen, dessen Umfang mit zunehmendem Wert (0,1,3) größer wird.

Mit BH läßt sich die Höhe der Beschriftungen ändern.

Das Zurücksetzen der Maßparameter geschieht durch RESET MASS;

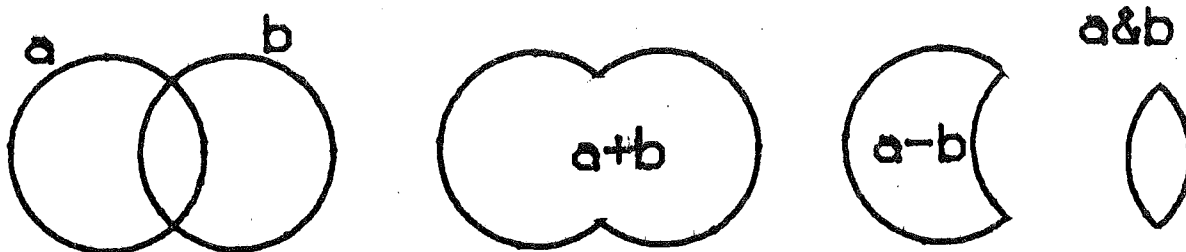
Erläuterung

- 1) Die Standardattribute erhalten neue Werte
- 2) Diese Standardattribute sind wirksam bei der Erzeugung von Punkten, Texten und Linienzügen aufgrund von Builtin-Funktion sowie bei der Erzeugung von sichtbaren Umrißlinien und Schnittlinien von Körpern bei der REDUCE-Operation (explizit oder implizit in der PLOT-Anweisung).
- 3) Wird ein Attribut wiederholt, so gilt die letzte Ausgabe.
- 4) Standardwerte sind

SYMBOL(1)	Punktsymbol
SLANT(0 GRAD)	Neigung von Text-Buchstaben
HEIGHT (5 MM)	Höhe von Punktsymbolen und Buchstaben
WIDTH(5 MM)	Breite von Textbuchstaben
EVERY(1)	Jeder Punkt eines Polygons wird markiert
DISTANCE(20 MM)	Skalenstrichabstand bei Achsen
OPEN	Polygonzüge offen
LINETYPE(SOLID)	Linienzüge durchgezogen
LENGTH(5 MM)	Strichabstand unterbrochener Linien
PEN(0)	Stiftauswahl nicht benutzt
COLOR(0)	Farbe irrelevant
LINEWIDTH(0.0)	Strichstärke nicht benutzt
FONT(0)	Standard-GIPSY-Zeichensatz
MAT(LEFT)	Materialrichtung links
- 5) Umriß- und Schnittlinien werden stets als OPEN erzeugt unabhängig von der Standardattributangabe.
- 6) Bei Ausgabe graphischer Objekte auf graphischen Geräten mit der PLOT-Anweisung kann es vorkommen, daß bestimmte Darstellungsattribute nicht realisiert werden können. Der Gerätetreiber wird dann in einer gerätespezifischen Weise darauf reagieren.

Erläuterung

Die CONT2-Funktion liefert als ihr Ergebnis die Umrandung (Kontur) einer durch "cont-expression" beschriebenen Fläche. Das Ergebnis ist eine Kollektion von LINE2-Objekten. Vor Beginn der Berechnung werden alle Objekte vom Typ POLY2, CIRCLE2, ARC2 zu Objekten von Typ LINE2 gewandelt. Dabei wird die Standard-Materialrichtung genommen, bei Kreisen und Kreisbögen innen. Alle LINE2-Objekte werden als geschlossen angesehen, d.h. der erste mit dem letzten Punkt verbunden. Damit ist die "cont-expression" eine Folge von geschlossenen Konturen, die durch "+", "-", "&" und Klammern verknüpft bzw. durch " " negiert sind. " " bedeutet Umkehrung der Materialrichtung. Umrandet wird nur der Bereich, der vom "Material" erfüllt ist. Die von Material erfüllte Fläche wird durch die Mengenoperationen "+", "-", "&" beschrieben:



Die Reihenfolge der Abarbeitung ist von links nach rechts, falls keine Klammern vorhanden sind:

a + b - c & d

 1. 2. 3.

Klammern bestimmen die Priorität der Abarbeitung:

a + (b - (c & d))
 3. 2. 1.

Falls die durch "cont-Expression" beschriebene Fläche das Clipp-Fenster schneidet, wird sie am Clipp-Fenster abgeschnitten. Siehe auch SHADE2 und LINE2.

Syntax

CONT2(cont-expression)

cont-Expression ::= cont-term |

cont-expression operator cont-term

cont-term ::= cont-obj | ¬cont-obj

cont-obj ::= g_ex | (cont-expression)

g_ex vom Typ LINE2, POLY2, CIRCLE2,
ARC2 oder COLL

operator ::= + | - | &

Beispiel

Kreis K und Linienzug L

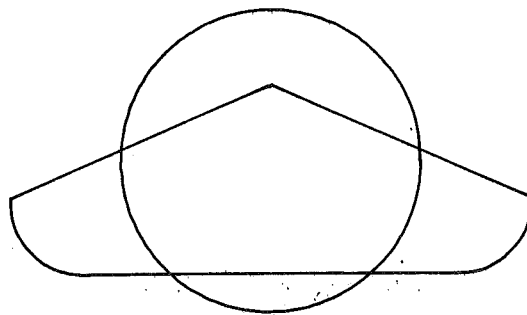
DCL (K,L)LINE2;

SET K=LINE2(CIRCLE2(PPOINT2(0,0),2,0));

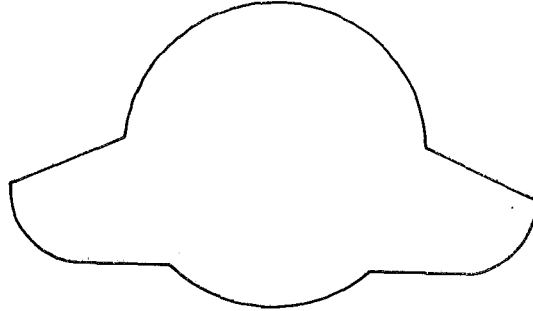
SET L=LINE2(POLY2(PPOINT2(0,1),PPOINT2(-3.5,-0.5)),

ROUND(1,90)ANGLE(-90),RIGHT(5),ROUND(1,90));

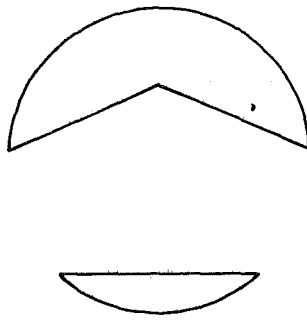
PLOT(K,L);



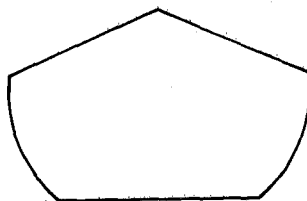
PLOT (CONT2 (K+L));



PLOT (CONT2 (K-L));



PLOT (CONT2 (K&L));



CONT2

Beispiel

```
PLOT (CONT2 (L-K));
```



Beispiele

```
DCL (A1,B1,C1) LINE2;
```

```
DCL HLINE(13) POLY(2);
```

```
    VLINE(5) POLY2(2);
```

Damit könnte man ein Netz von 12x4 Maschen erzeugen.

```
DCL KREIS POLY2(12);
```

```
N = 20; M = 50;
```

```
BEGIN;
```

```
    DCL A(0:N,10) POLY2(M);
```

```
    CALL SUB(A);
```

```
    SUB: PROC(AA);
```

```
        DCL AA(x,x)POLY2(x)PARAMETER;
```

```
        SET AA(I,J) = POLY2(.....);
```

```
        ⋮
```

```
    END SUB;
```

```
END;
```

```
DCL 1 S,
```

```
    2 S1 TEXT2(20),
```

```
    2 S2 COLL,
```

```
    2 S3 (5x10,100)POINT2;
```

```
DCL PP(100,100) POLY2(100) BASED(PQ);
```

```
ALLOC POLY2 PP;
```

```
    ⋮
```

```
FREE POLY2 PP;
```

Falsch wäre:

```
DCL 1 A, 2 P POINT; /* P ist ohne Qualifikation nicht  
                    eindeutig */
```

```
DCL 1 B, 2 P POINT;
```


(S.261 GHB)

DCL
DECLARE
2D-Objekte

Syntax

DCL [level] ident [dim] type [storage class] ;

Erläuterung

level : level innerhalb von PL/1-Strukturen

ident : Name des Objektes nach PL/1-Regeln, mit maximal
20 Zeichen
Namen von GIPSY-Objekttypen und GIPSY-Builtin-
Funktionen sind verboten

dim : Dimension nach PL/1-Regeln
Maximal 6 Dimensionen sind erlaubt

type : POINT2 | CIRCLE2 | ARC2 | LINE2 |
POLY2(max-point-number) |
TEXT2(max-char-number) |
AXIS2(max-char-number) | COLL | COLLECTION

storage class: AUTOMATIC | BASED | PARAMETER

CONTROLLED ist verboten, ebenso wie die Attribute
STATIC und EXTERNAL

max-point-number: p11_ex, maximale Punktzahl des Polygonzuges

max-char-number: p11_ex, maximale Länge der Zeichenkette des
Textes oder der Achsenbeschriftung

- 1) Für BASED-Objekte gibt es besondere ALLOC- und FREE-Statements
- 2) Objekte, die Parameter von Unterprogrammen sind, müssen das
Attribut PARAMETER tragen. Bei Parameter-Strukturen genügt
die Angabe von PARAMETER auf Level 1.
- 3) Alle Namen von GIPSY-Objekten müssen auch ohne Strukturquali-
fikation eindeutig sein.
- 4) Siehe Kapitel 4

Beispiel

```
DCL C(10) COLLECTION;  
DO J = 1 TO 5;  
EDIT HEIGHT(3 MM), WIDTH(4 MM), SYMBOL(J) FOR(C(J));  
END;
```

/x Die Kollektionen C(1) bis C(5) werden editiert.

Alle Texte erhalten die Höhe 3 mm und die Breite 4 mm.

Der Symboltyp variiert von Kollektion zu Kollektion x/.

```
DCL TT TEXT2(50);  
EDIT FONT(4) FOR(TT);  
DCL LL LINE2;  
EDIT MAT(RIGHT) OF(LL);
```

(S.265 GHB)

EDIT
Statement

Syntax

EDIT attribut [,attribut] $\left\{ \begin{array}{l} \text{OF} \\ \text{FOR} \end{array} \right\}$ (g_obj [,g_obj]^{*});

attribut = SYMBOL(integer) |
 HEIGHT(1value) |
 EVERY(integer) |
 OPEN | CLOSED |
 LINETYPE [(] linetype [)] |
 LENGTH(1value) |
 PEN(integer) |
 COLOR(integer) |
 LINEWIDTH(1value) |
 WIDTH(1value) |
 SLANT(avalue) |
 FONT(integer) |
 MAT (RIGHT | LEFT)

Erläuterung

- 1) Die Darstellungsattribute der Punkt, Texte, Linien in den graphischen Objekten der FOR-Option werden neu festgelegt.
- 2) Wenn für eines der Objekte eines der Attribute irrelevant ist, so ist es wirkungslos.
- 3) Siehe CHANGE STANDARD und CHANGE INVISIBLE

Beispiel

```
DCL CO COLLECTION;  
SUB: PROC(C);  
DCL C COLLECTION PARAMETER;  
DCL S SPACE(3), B BODY, T TEXT2;  
etc.  
BUILD S = SPACE(.....);  
BUILD B = BODY(.....);  
SET C = COLLECTION(REDUCE(B),T);  
EMPTY(B);  
END SUB;  
CALL SUB(CO);  
PLOT(CO);  
EMPTY(CO);  
  
DCL (L1,L2) LINE2;  
⋮  
EMPTY (L1,L2);
```

(S.267 GHB)

EMPTY

Statement

Syntax

```
EMPTY(g_obj [, g_obj]* );
```

Erläuterung

1) Graphische Objekte vom Typ

COLLECTION |

SPACE |

BODY |

LINE2

werden "geleert" (in den Grundzustand versetzt).

2) Bei allen anderen Objekttypen keine Wirkung.

3) Diese Anweisung ist bei Objekten vom Typ COLLECTION, LINE2 oder BODY erforderlich. Sie muß vor Verlassen des Blockes, in dem die Objekte deklariert sind, ausgeführt werden. Wird dies versäumt, so bleiben im Arbeitsspeicher unzugängliche Daten liegen, die unnötig Platz verbrauchen.

Syntax

HATCH2(winkel, abstand, shade-expression)

winkel : avalue, Winkel der Schraffur

abstand: lvalue, Abstand der Schraffurlinien

shade-expression ::= shade-term |
shade expression operator shade-term

shade-term ::= shade-obj | ¬shade-obj
::= g_ex | (shade-expression)
g_ex vom Typ LINE2, POLY2, CIRCLE2,
ARC2 oder COLL

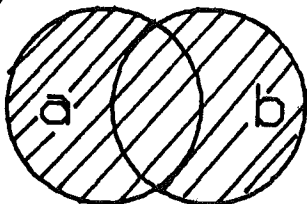
operator ::= + | - | &

Erläuterung

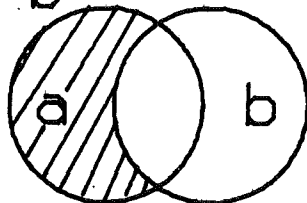
Vor Beginn der Schraffur werden alle Objekte vom Typ POLY2, CIRCLE2, ARC2 zu Objekten von Typ LINE2 gewandelt. Dabei wird die Standard-Materialrichtung genommen, bei Kreisen und Kreisbögen innen. Alle LINE2-Objekte werden als geschlossen angesehen, d.h. der erste mit dem letzten Punkt verbunden.

Damit ist die "shade-expression" eine Folge von geschlossenen Konturen, die durch "+", "-", "&" und Klammern verknüpft bzw. durch " " negiert sind. " " bedeutet Umkehrung der Materialrichtung. Schraffiert wird nur der Bereich, der von "Material" erfüllt ist. Die von Material erfüllte Fläche wird durch die Mengenoperationen "+", "-", "&" beschrieben:

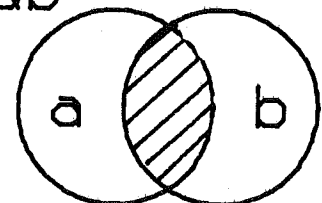
a+b



a-b



a&b



Die Reihenfolge der Abarbeitung ist von links nach rechts, falls keine Klammern vorhanden sind:

a + b - c & d

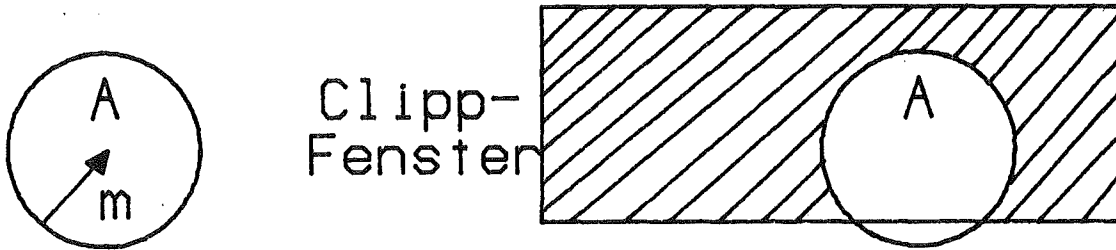
1. 2. 3.

Klammern bestimmen die Priorität der Abarbeitung:

$$a + (b - (c \& d))$$

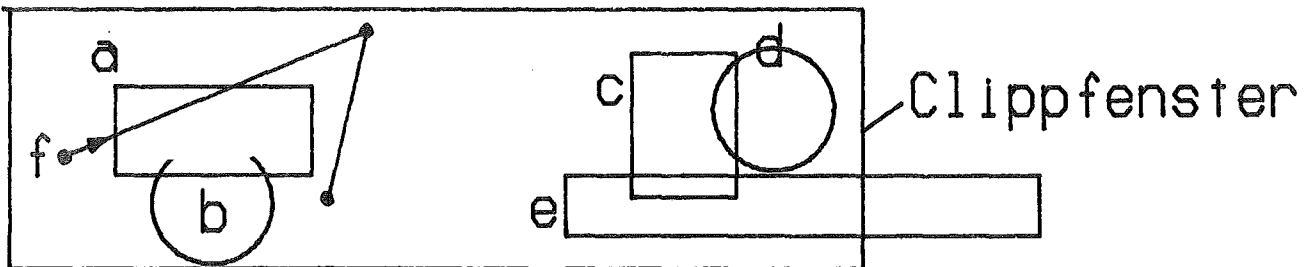
3. 2. 1.

Die Schraffur wird am gerade gültigen CLIPP-Fenster abgeschnitten, z.B. HATCH2(\neg A)



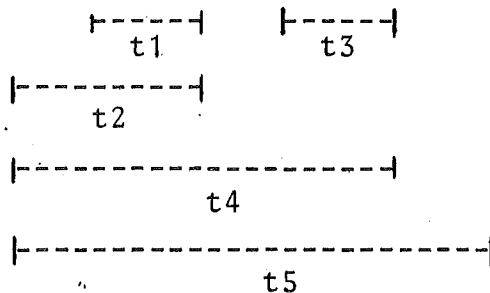
Die HATCH2-Funktion liefert als ihr Ergebnis eine Kollektion von Linien (POLY2). Ist abstand 0, so wird in die Kollektion zusätzlich als LINE2-Objekt(e) die Kontur der erzeugten Fläche aufgenommen. Für abstand 0 wird als Schraffurstrichabstand abstand genommen. In diesem Fall besteht die Ergebniskollektion nur aus den Schraffurlinien, ohne die Umrandung (siehe auch CONT2).

Beispiel

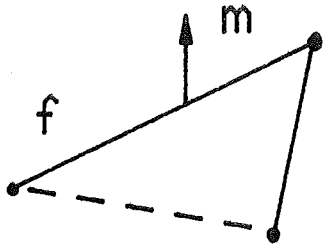


DCL c COLL:

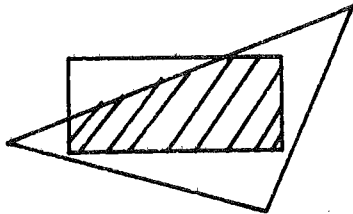
```
SET c = HATCH2(45,1,b + (a &  $\neg$ f) + (c + e) - d);
```



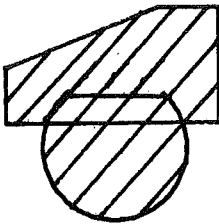
Vorbereitung: Aus POLY2 f und Bogen b LINE2 machen.



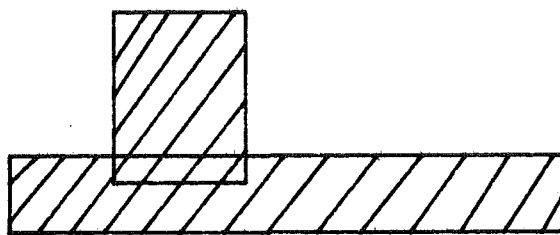
1. Operation: $a \& \neg f \rightarrow t1$



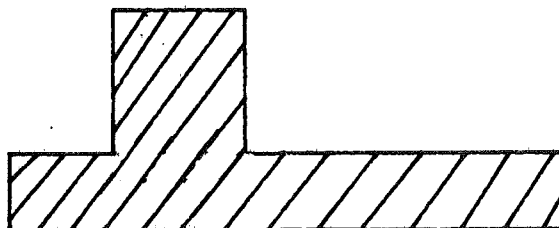
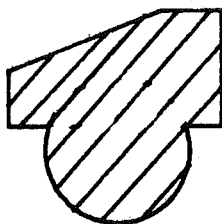
2. Operation: $b + t1 \rightarrow t2$



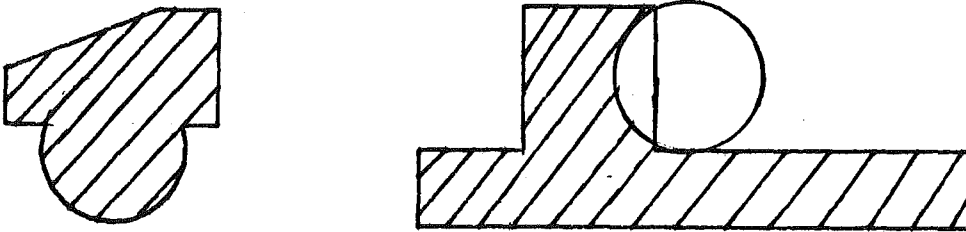
3. Operation: $c + e \rightarrow t3$



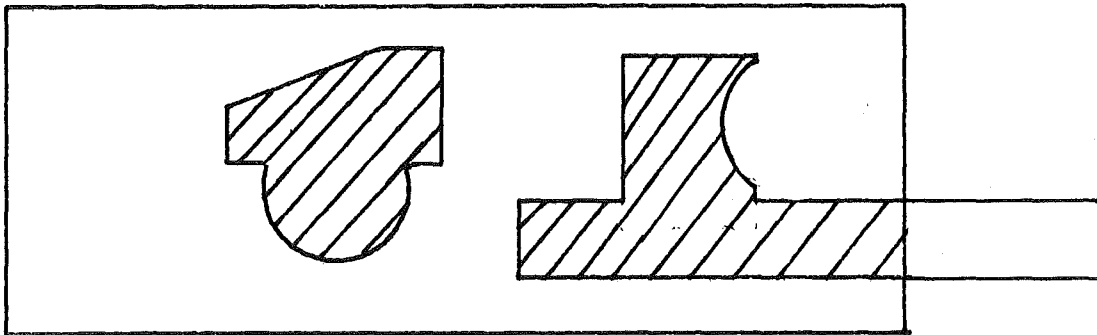
4. Operation: $t2 \& t3 \rightarrow t4$



5. Operation: t4-d → t5

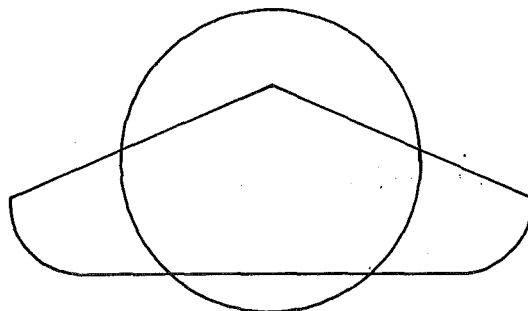


6. Operation: t5&Bildfenster (CLIPP)

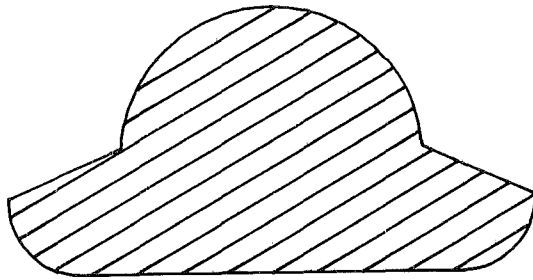


b) Kreis K und Linienzug L

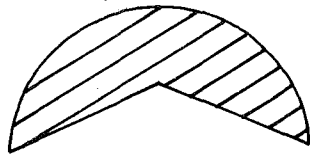
```
DCL (K,L) LINE2;  
SET K=LINE2(CIRCLE2(POINT2(0,0),2.0));  
SET L=LINE2(POLY2(POINT2(0,1),POINT2(-3.5,-0.5)),  
    ROUND(1,90)ANGLE(-90),RIGHT(5),  
    ROUND(1,90));  
PLOT(K,L);
```



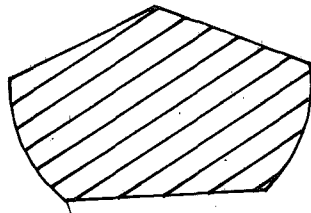
PLOT (HATCH2 (30, 3MM, K+L));



PLOT (HATCH2 (30, 3MM, K-L));



PLOT (HATCH2 (30, 3MM, K&L));



PLOT (HATCH2 (30, 3MM, L-K));



Syntax

```
LINE2( objekt [ , objekt moveoperation ]** )  
objekt ::= point2 | poly2 | line2 | arc2 | circle2  
moveoperation ::= move [ set ]  
set ::= SET( point2-reference )  
move ::= UP( dy ) [ angle ]  
      | DOWN( dy ) [ angle ]  
      | RIGHT dx ) [ angle ]  
      | LEFT( dx ) [ angle ]  
      | UPTO( point2 ) [ angle ]  
      | DOWNTO( point2 ) [ angle ]  
      | RIGHTTO( point2 ) [ angle ]  
      | LEFTTO( point2 ) [ angle ]  
      | LEGTH( laenge ) [ angle ]  
      | ROUND( radius , avalue ) [ angle ]  
      | ROUNDTO( poly2, radius ) [ angle ]  
      | RTAN( circle2 | arc2 )  
      | LTAN( circle2 | arc2 )  
      | NORMAL( poly2 )  
  
angle ::= ANGLE( avalue ) | TURN( avalue )  
dx , dy , laenge , radius ::= lvalue  
point2, poly2 ::= g_ex
```

Bezüglich "avalue", "lvalue", "g_ex" siehe S.33, 229 und 289 GHB.

Standardwerte für angle:

ANGLE(90 GRAD) für UP, DOWN, UPTO, DOWNTO
ANGLE(0) für LEFT, RIGHT, LEFTTO, RIGHTO
TURN(0) für ROUND, ROUNDTO

Deklaration

DCL [level] ident [dim] LINE2 [storage class]:

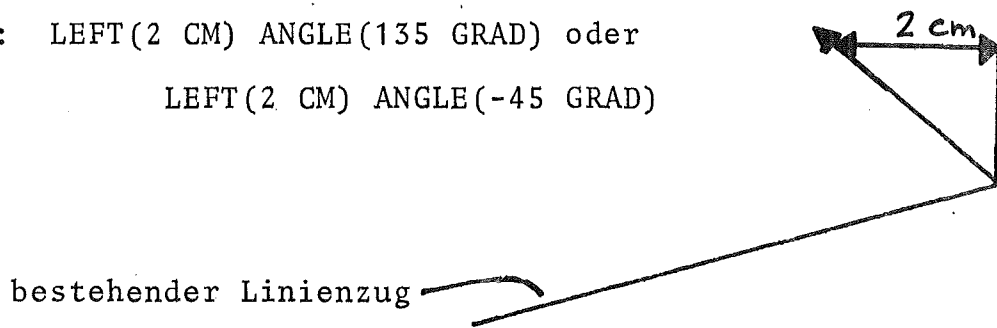
Erläuterung

- In der angegebenen Reihenfolge werden die Objekte aneinandergefügt bzw. die Operationen ausgeführt. Bei den Moveoperationen kann die SET-Option dazu benutzt werden, den erreichten Punkt in " point2-reference " zu speichern (ebenfalls in der angegebenen Reihenfolge).
- Auch Kreise und geschlossene Polygonzüge werden in das LINE2-Objekt aufgenommen. Das Objekt enthält danach eine geschlossene Schleife.
- Polygonzüge, Kreisbögen, Linienzüge, die aufgenommen werden, erhalten die Standard-Darstellungsattribute. Auch die Verbindungslinie, die (außer beim ersten Objekt) erzeugt wird, enthält ebenso wie die durch " move " erzeugten Teile die gültigen Standardattribute. Zu beachten ist, daß die Punkte in den Objekten die richtige Reihenfolge haben (siehe S.
- Die Option ANGLE(avalue) gibt die Richtung des neu erzeugten Abschnittes (bzw. seiner Tangente) im Anfangspunkt an, bezogen auf das Koordinatensystem.
- Die Option TURN ("avalue") gibt die Richtung des neu erzeugten Abschnittes (bzw. seiner Tangente) im Anfangspunkt an, bezogen auf die Richtung des dort endenden vorhergehenden Abschnittes. Ausnahme: wenn nur ein Punkt vorliegt, wirkt TURN wie ANGLE.

- bei UP, DOWN, LEFT, RIGHT, UPTO, DOWNT, LEFTTO, RIGHTTO wird avalue in der TURN- bzw. ANGLE-Option modulo π genommen. In diesen Fällen ist

$$\begin{Bmatrix} \text{ANGLE} \\ \text{TURN} \end{Bmatrix} (\text{avalue} + n \cdot \pi) = \begin{Bmatrix} \text{ANGLE} \\ \text{TURN} \end{Bmatrix} (\text{avalue})$$

Beispiel: LEFT(2 CM) ANGLE(135 GRAD) oder
LEFT(2 CM) ANGLE(-45 GRAD)

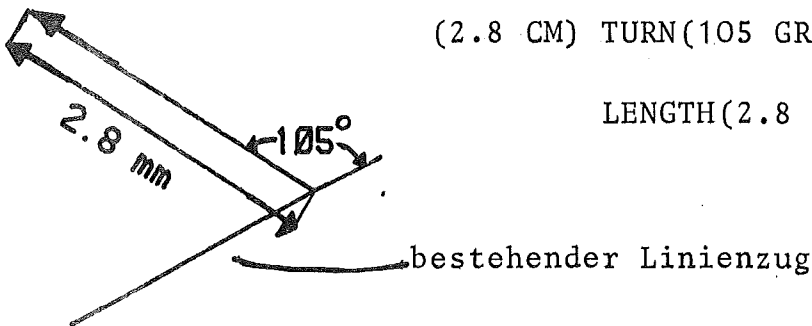


- bei LENGTH, ROUND wird <avalue> modulo 2π genommen

LENGTH

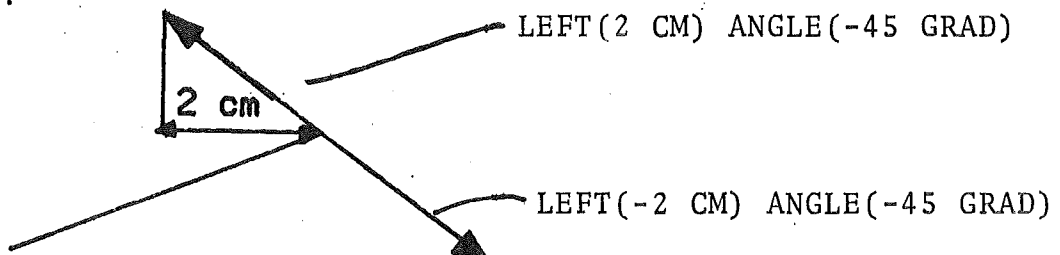
(2.8 CM) TURN(105 GRAD)

LENGTH(2.8 CM) TURN(-75 GRAD)



- bei ROUND wird <avalue> in der TURN oder ANGLE-Option modulo π genommen (siehe unten).
- UP, DOWN, RIGHT, LEFT zieht eine Gerade im angegebenen Winkel soweit, daß der angegebene Betrag in Y bzw. X-Richtung zurückgelegt wird. Negative Beträge kehren die Richtung um.

Beispiel:



Nicht realisierbare Bewegungen sind ein Fehler.

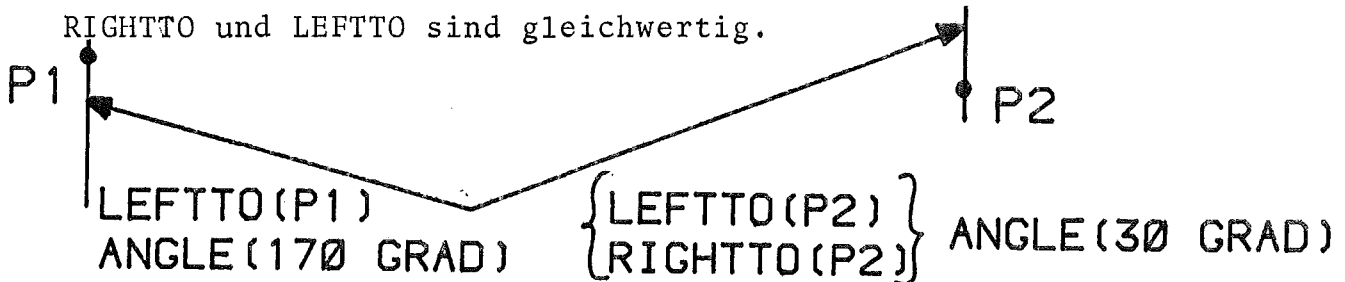
Beispiel: UP(dy) ANGLE(0)

- UPTO, DOWNTO zieht eine Gerade in der angegebenen Richtung bis zur x-Koordinate des angegebenen Punktes.

UPTO und DOWNTO sind gleichwertig.

- RIGHTO, LEFTTO zieht eine Gerade in der angegebenen Richtung bis zur x-Koordinate des angegebenen Punktes.

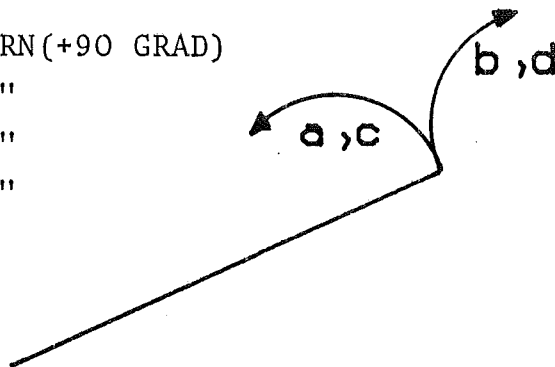
RIGHTTO und LEFTTO sind gleichwertig.



- ROUND erzeugt einen arc2 mit Anfangstangente in der angegebenen Richtung. Das Vorzeichen von radius gibt im mathematischen Sinne die Richtung des Bogenstückes an. avalue gibt an, wie weit der Bogen zu ziehen ist. Von avalue wird nur der Betrag verwendet.

Beispiel:

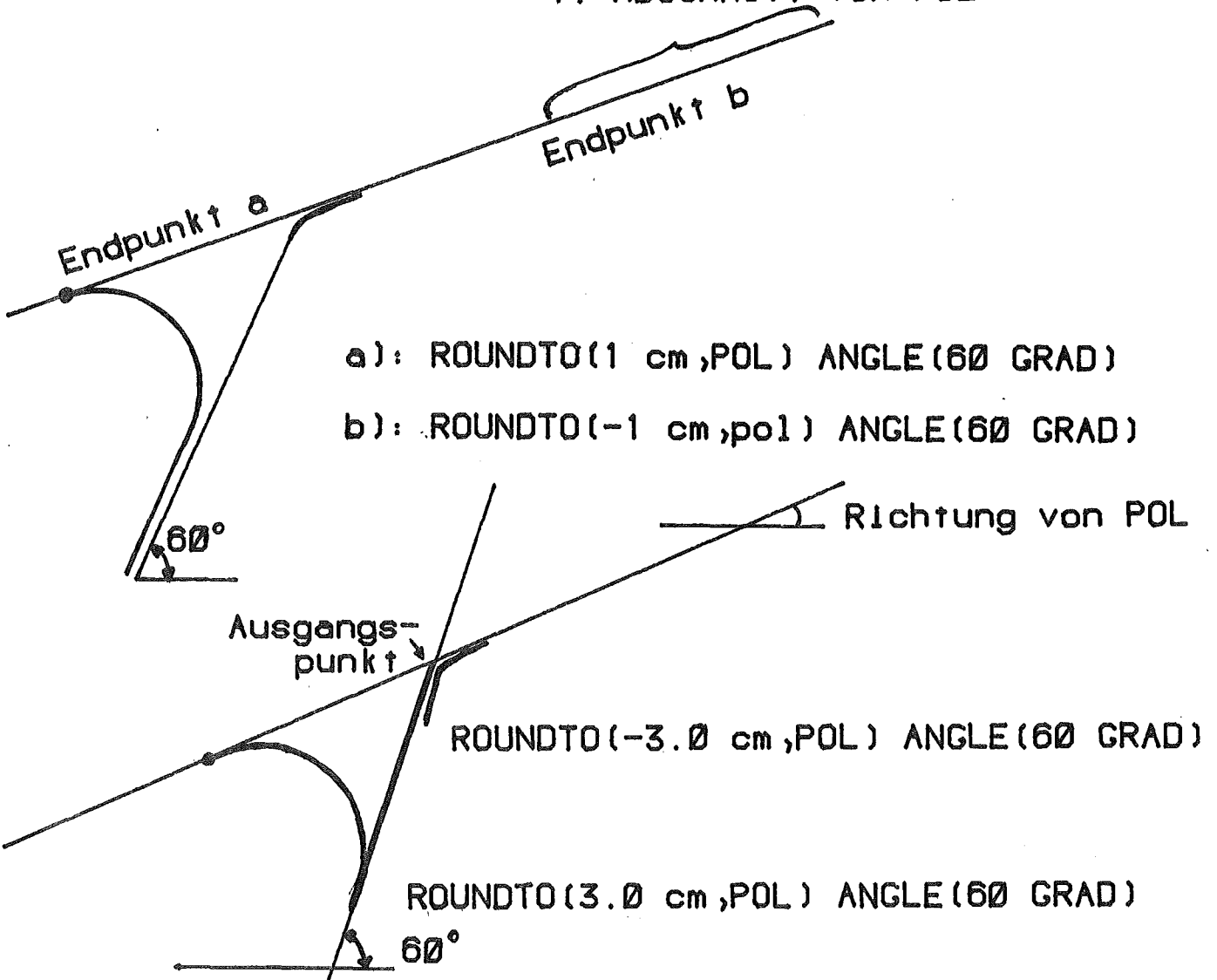
- a) ROUND(2 CM, 90 GRAD) TURN(+90 GRAD)
- b) ROUND(-2 CM, 90 GRAD) "
- c) ROUND(2 CM, -90 GRAD) "
- d) ROUND(-2 CM, -90 GRAD) "



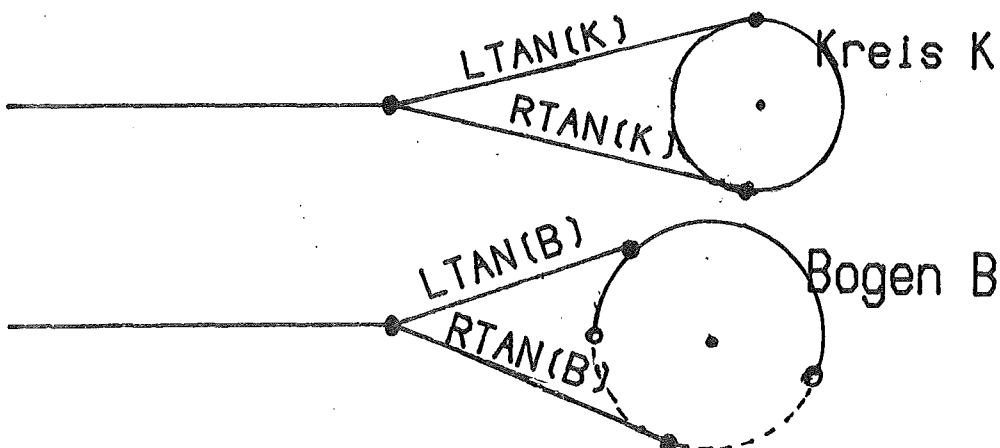
- ROUND erzeugt ein Geradenstück im angegebenen angle und einen Kreisbogen mit Radius radius, der dieses Geradenstück und diejenige Gerade, die vom ersten Abschnitt des poly2 definiert ist, berührt. Das Bogenstück hat die Richtung, die vom Vorzeichen von radius angegeben ist. Der Bogen ist stets kleiner als 180 Grad und liegt stets auf derjenigen Seite der von

von poly2 definierten Geraden, auf der auch der Ausgangspunkt liegt. (Liegt dieser Punkt auf der Geraden, so wird die Anwendung mit Fehlermeldung ignoriert).

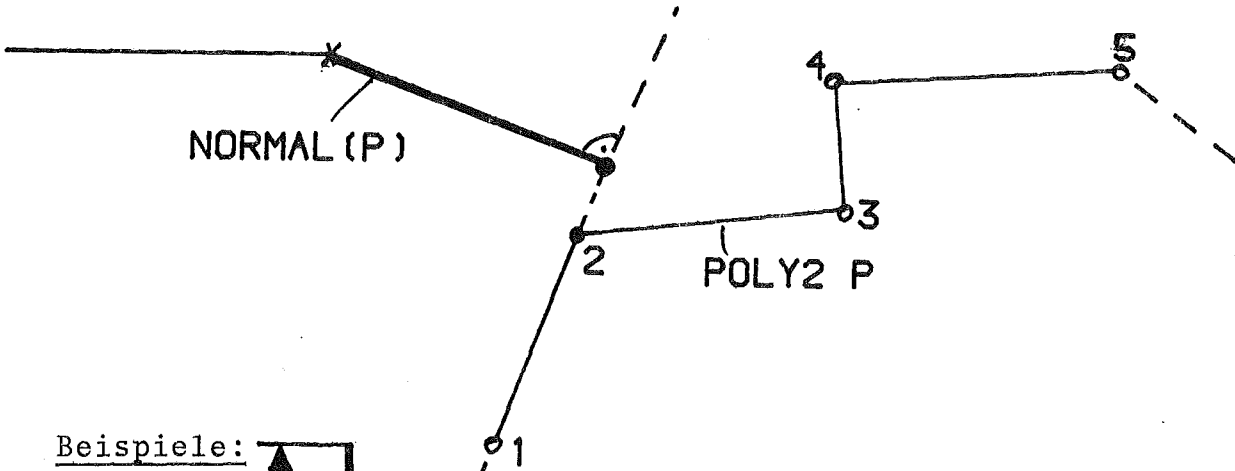
1. Abschnitt von POL



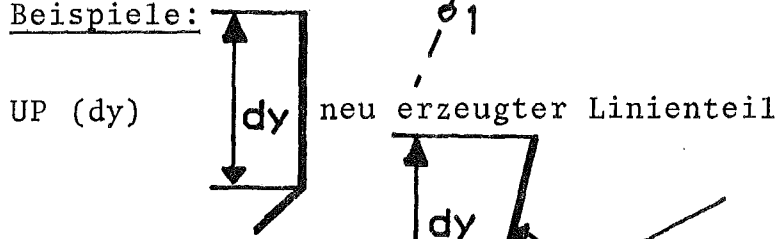
- RTAN/LTAN erzeugt die rechte/linke Tangente an den angegebenen Kreis oder Kreisbogen. Fehler: Anfangspunkt innerhalb des Kreises. Tangenten an Kreisbögen werden so erzeugt, als sei der Bogen zu einem vollständigen Kreis erweitert.



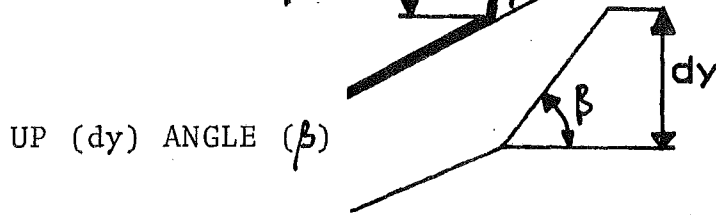
- NORMAL fällt das Lot vom aktuellen Punkt aus auf die Gerade, die durch die ersten zwei Punkte des POLY2-Objektes geht.



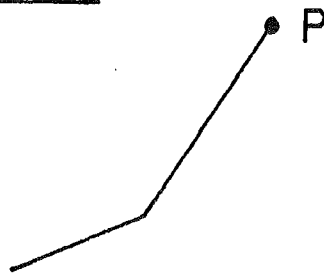
Beispiele:



UP (dy) TURN (β)

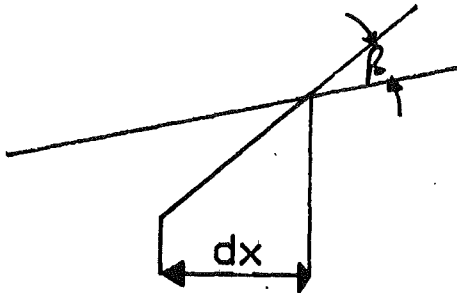


UP (dy) $\left\{ \begin{matrix} \text{ANGLE}(\alpha) \\ \text{TURN}(\beta) \end{matrix} \right\}$ SET (P)



Entsprechend für DOWN, LEFT, RIGHT

LEFT(dx) TURN(β)



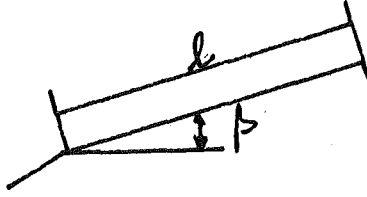
DOWNTO(P) ANGLE(β)



Ebenso für UPTO, RIGHTTO, LEFTTO

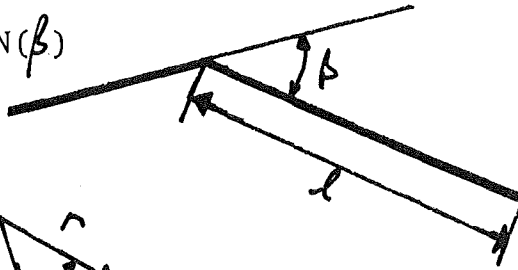
LENGTH(l)

ANGLE(β)

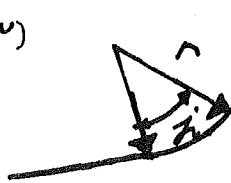


LENGTH(l)

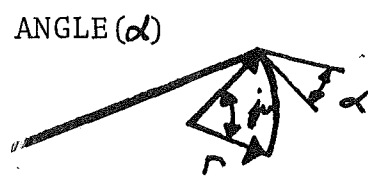
TURN(β)



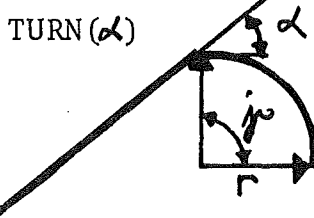
ROUND(r, μ)



ROUND($-r, \mu$)

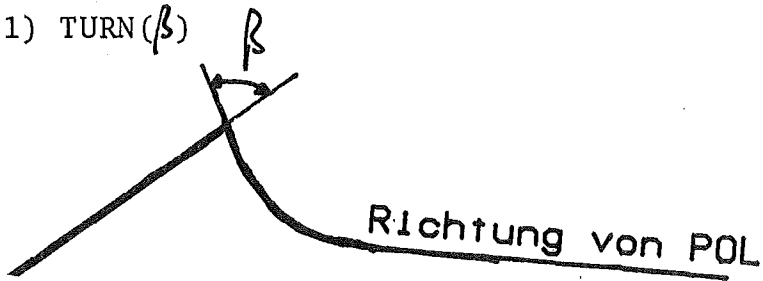


ROUND($-r, \mu$)

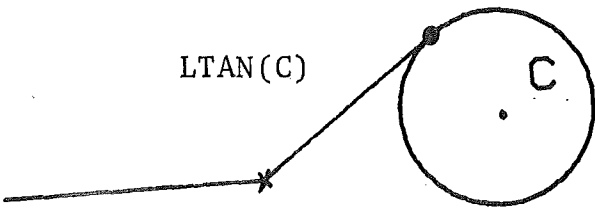


ROUNDTO($r, \text{pol1}$)

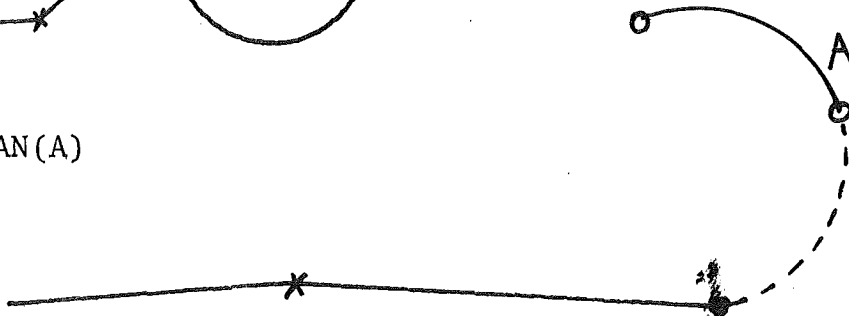
TURN(β)



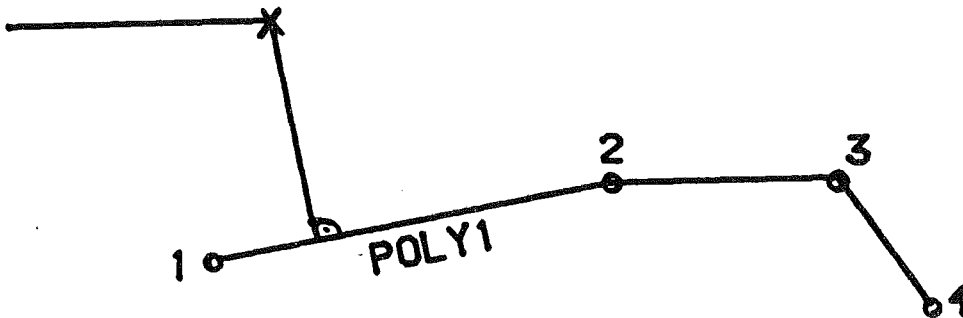
LTAN(C)



RTAN(A)

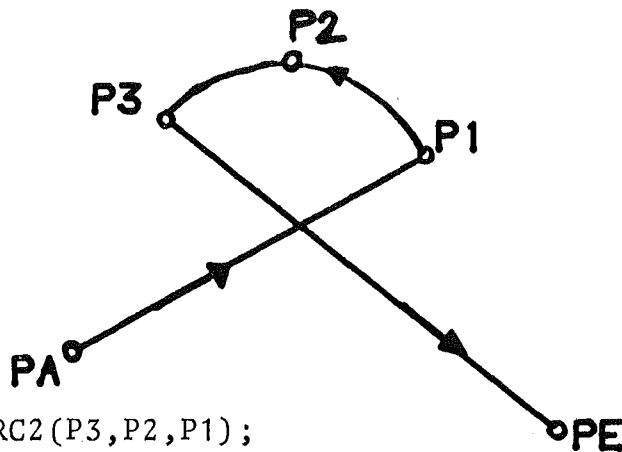


NORMAL(POLY1)

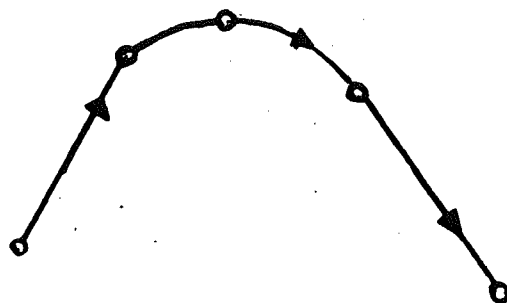


Aufnahme von Bögen: Zu beachten ist die Bogenrichtung (siehe auch ARC2-Funktion).

```
DCL A ARC2;
DCL L LINE2;
SET A = ARC2(P1,P2,P3);
SET L = LINE2(PA,A,PE);
```



```
SET A = ARC2(P3,P2,P1);
SET L = LINE2(PA,A,PE);
```



Syntax

MØ1U }
MØ1R } (P1, maßzahl, [lage], [abstand], [info], [plz])
MØ10 }
MØ1L }

Erläuterung

P1 : Konturpunkt, Typ POINT2
lage : Lage des Konturpunktes, 'L' oder 'R'
abstand : Abstand des Maßtextes vom Konturpunkt in cm
(Länge der Maßhilfslinie)
maßzahl : Maßtext und Maßlückengröße
info : Zusatzinformation, s.Kap. 2.2, S. 13
plz : Platzierung der Maßzahl, Kap. 2.2, S. 13 ff.

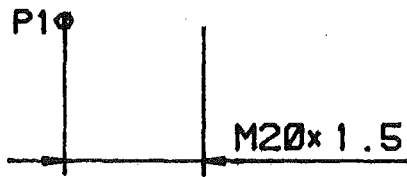
Standardwerte

info : "(Leerstring)
abstand : 1.cm
plz : 0.
lage : 'R' bei MØ1U
'L' " MØ1R
'L' " MØ10
'R' " MØ1L

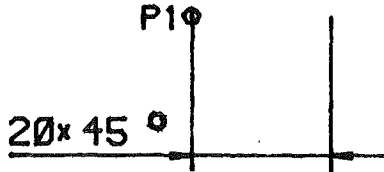
MØ1U(P1 ,LAGE ,MH ,MZ ,ZI ,PL

Beispiele

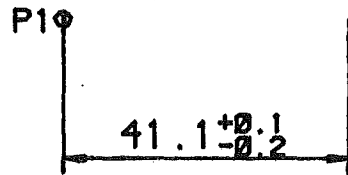
<p>⌀ 110.134</p>	L	1.0	110.1344	◆S3◆PTZSHM	-0.5
<p>50x45 ◦ ABGESCHRAEGT</p>	L	1.0	50.00	◆FX45◆TNABGESCHRAEGT	2.5
<p>⌀110.5 H11</p>	L	1.0	110.52	◆D◆PM◆0AH11	0.95
<p>110.35 M 0.2</p>	L	1.0	110.352	◆S2◆OU0.2◆MMC◆TM	0.35
<p>110.15 M 0.2</p>	L	1.0	110.153	◆OU0.2◆S2◆MMC	0.3
<p>POLIERT 110.2^{+0.2}/_{-0.15}</p>	L	1.0	110.153	◆TVPOLIERT◆OA0.2◆UA0.15	0.8
<p>110.1x45 ◦ ABGESCHRAEGT</p>	R	1.0	110.06	◆F45◆TNABGESCHRAEGT	0.5
<p>110.134 SEITE 1</p>	L	1.0	110.134	◆S3◆TNSEITE 1	0.0



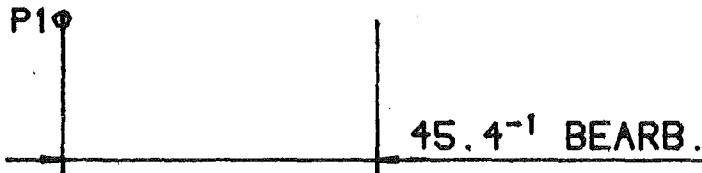
M01U(P1,20.0,'L',2.,'\$MX1.5',1.2)



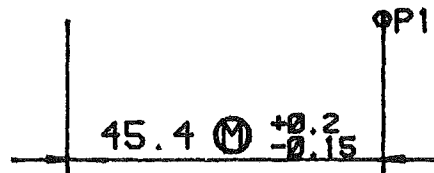
M01U(P1,20.0,'L',2.,'\$FX45',2)



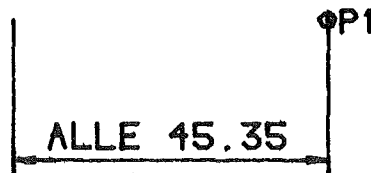
M01U(P1,41.15,'L',2.,'\$OA0.1\$UA0.2',1.5)



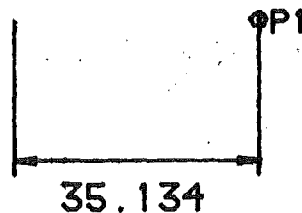
M01U(P1,45.42,'L',2.,'\$OA-1\$TNBEARB.',0.5)



M01U(P1,45.42,'R',2.,'\$OA0.2\$UA-0.15\$MMC',2)

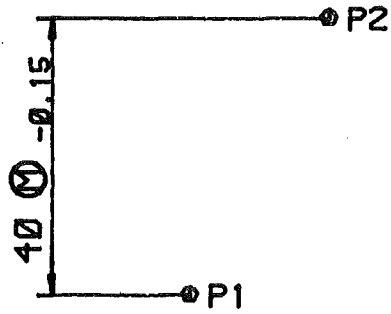


M01U(P1,45.352,'R',2.,'\$S2\$TVALLE\$TH',2)

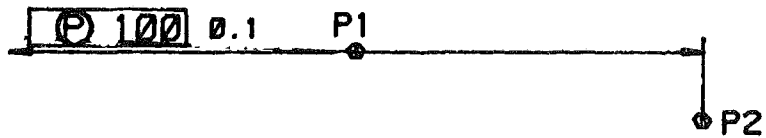


M01U(P1,35.1344,'R',2.,'\$S3\$TLINKS',-3)

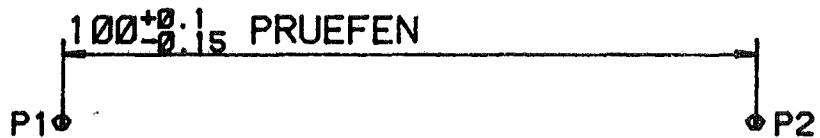
Beispiele



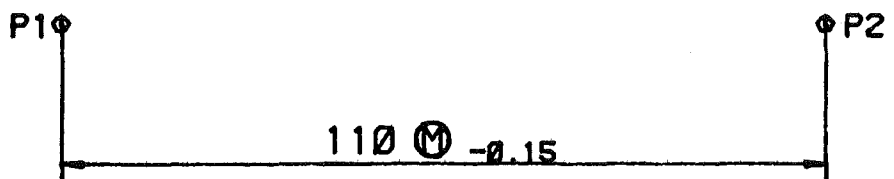
M02L(P1 ,P2 ,40.0 , 'L' ,2. , '\$UA0.15\$MMC' ,0.5)



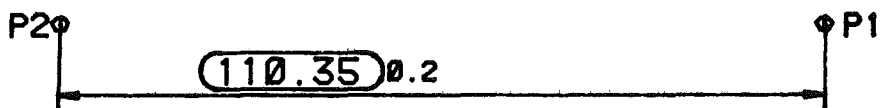
M020(P2 ,P1 ,100. , 'R' ,1. , '\$OU0.1\$TM\$PTZ' ,1.2)



M020(P1 ,P2 ,100. , 'L' ,1. , '\$OA0.1\$UA0.15\$TNPRUEFEN' ,1.2)



M02U(P1 ,P2 ,110.0 , 'L' ,2. , '\$UA0.15\$MMC' ,0.5)



M02U(P1 ,P2 ,110.352 , 'R' ,1. , '\$S2\$OU0.2\$PM' ,2.35)

MO2
Maßroutine
Längenmaß

Syntax

MØ2U }
MØ2R } (P1,P2, maßzahl, [lage], [abstand], [info], [plz])
MØ2O }
MØ2L }

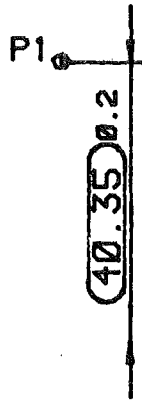
Erläuterung

P1, P2 : Erster und zweiter Konturpunkt. Typ POINT2
lage : Lage des Konturpunktes P1.s.Kap.2.3.2
abstand : Abstand des Maßtextes von den Konturpunkten in
cm. (Länge der Maßhilfslinie)
maßzahl : Maßtext und Maßlückengröße
info : Zusatzinformation, s.Kap.2.2, S.13.
plz : Platzierung der Maßzahl, Kap. 2.2, S.13 ff.

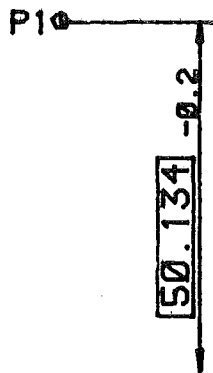
Standardwerte

info : "
abstand : 1 cm
plz : 0.
lage : 'R' bei MØ2U
'L' " MØ2R
'L' " MØ2O
'R' " MØ2L

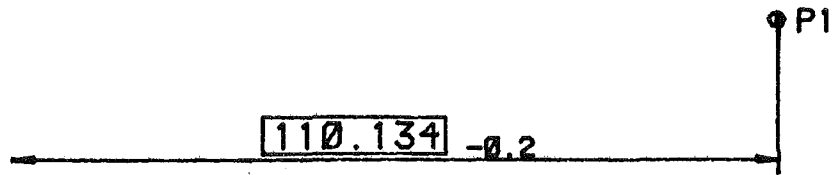
Beispiele



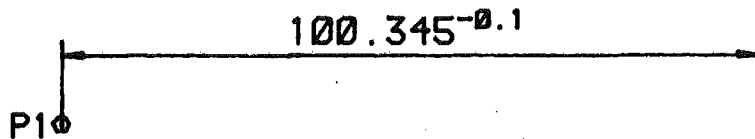
```
M03R(P1,40.352,'R',1.,'$S2$OU0.2$PM',2.35)
```



```
M03R(P1,50.1344,'R',2.,'$S3$UA0.2$TM',0.)
```



```
M03U(P1,110.1344,'R',2.,'$S3$UA0.2$TM',0.)
```



```
M03O(P1,100.345,'L',1.0,'$S3$OA-0.1',1.0)
```


Syntax

MØ3U }
MØ3R } (P1, maßzahl, [lage], [abstand], [info], [plz])
MØ3O }
MØ3L }

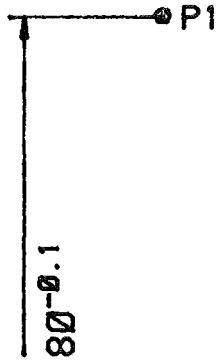
Erläuterung

P1 : Konturpunkt, Typ POINT2
lage : Lage des Konturpunktes P1.
abstand : Abstand des Maßtextes von dem Konturpunkt
in cm. (Länge der Maßhilslinie)
maßzahl : Maßtext und Maßlückengröße
info : Zusatzinformation, s.Kap.2.2, S.13
plz : Platzierung der Maßzahl, Kap. 2.2, S.13 ff.

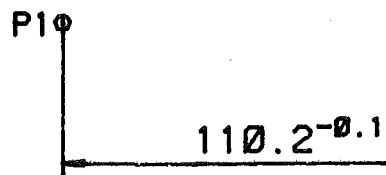
Standardwerte

info : "
abstand : 1 cm
plz : 0.
lage : 'R' bei MØ3U
'L' " MØ3R
'L' " MØ3O
'R' " MØ3L

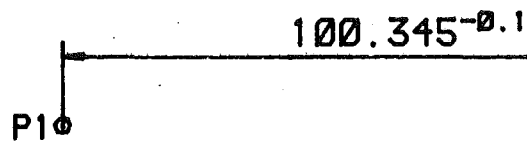
Beispiele



MØ4L(P1 ,80. , 'R' ,2. , 'ØA-Ø.1' , -0.5)



MØ4U(P1 ,110.152 , 'L' ,2. , 'ØA-Ø.1ØPOLIERT' , 0.3)



MØ40(P1 ,100.345 , 'L' ,1.0 , 'ØS3ØA-Ø.1' , 1.0)

MO4
Maßroutine
Längenmaß

Syntax

MØ4U }
MØ4R } (P1, maßzahl, [lage], [abstand], [info], [plz])
MØ4O }
MØ4L }

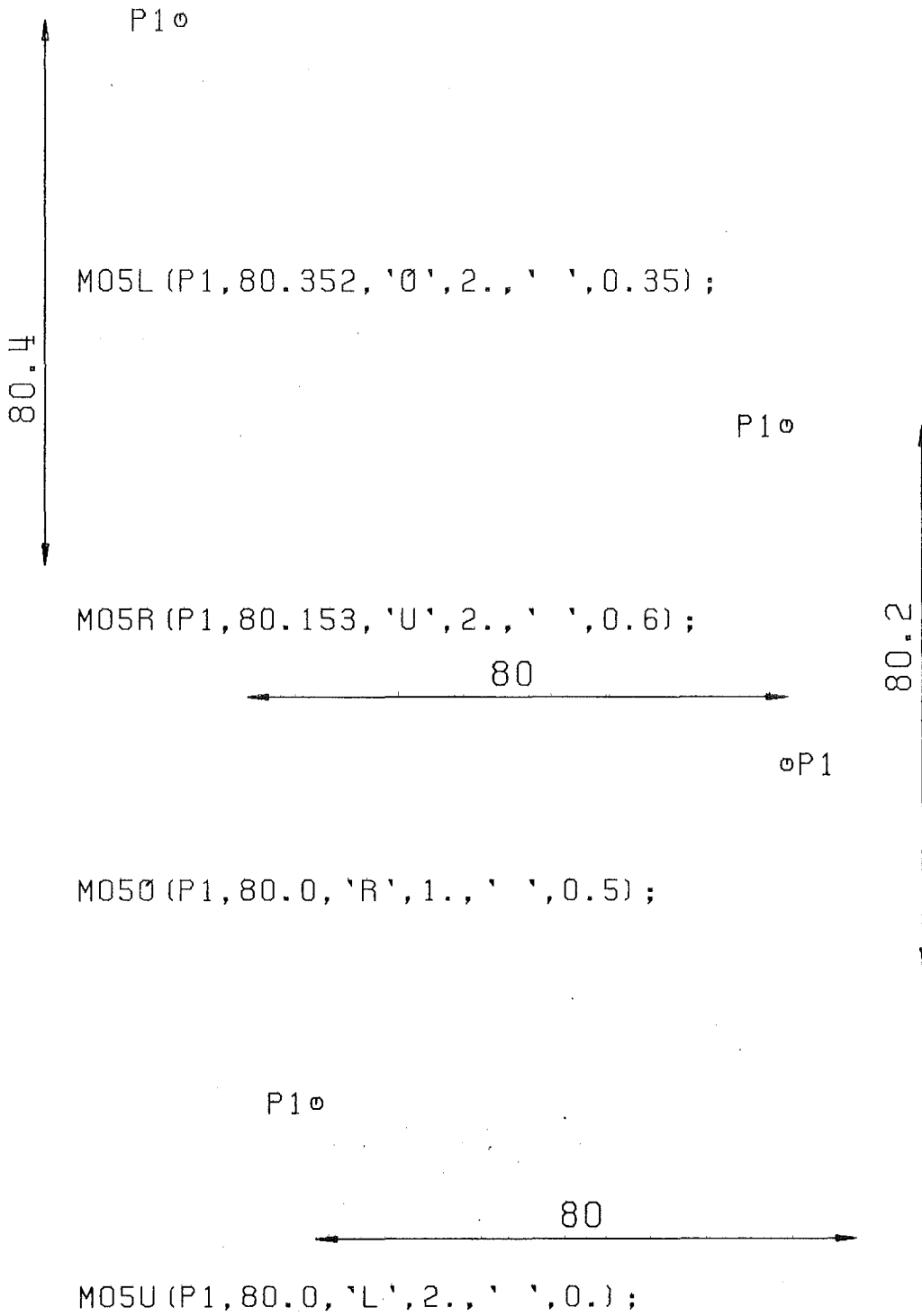
Erläuterung

P1 : Konturpunkt. Typ POINT2
lage : Lage des Konturpunktes, 'L' oder 'R'
abstand : Abstand des Maßtextes vom Konturpunkt in cm.
(Länge der Maßhilfslinie)
maßzahl : Maßtext und Maßlückengröße
info : Zusatzinformation, s.Kap. 2.2, S.13
Plz : Platzierung der Maßzahl, Kap. 2.2, S. 13 ff.

Standardwerte:

info : "
abstand : 1 cm
plz : 0.
lage : 'R' bei MØ4U
'L' " MØ4R
'L' " MØ4O
'R' " MØ4L

Beispiele



M05
Maßroutine
Längemaß

Syntax

MØ5U }
MØ5R } (P1, maßzahl, [lage], [abstand], [info], [plz])
MØ50 }
MØ5L }

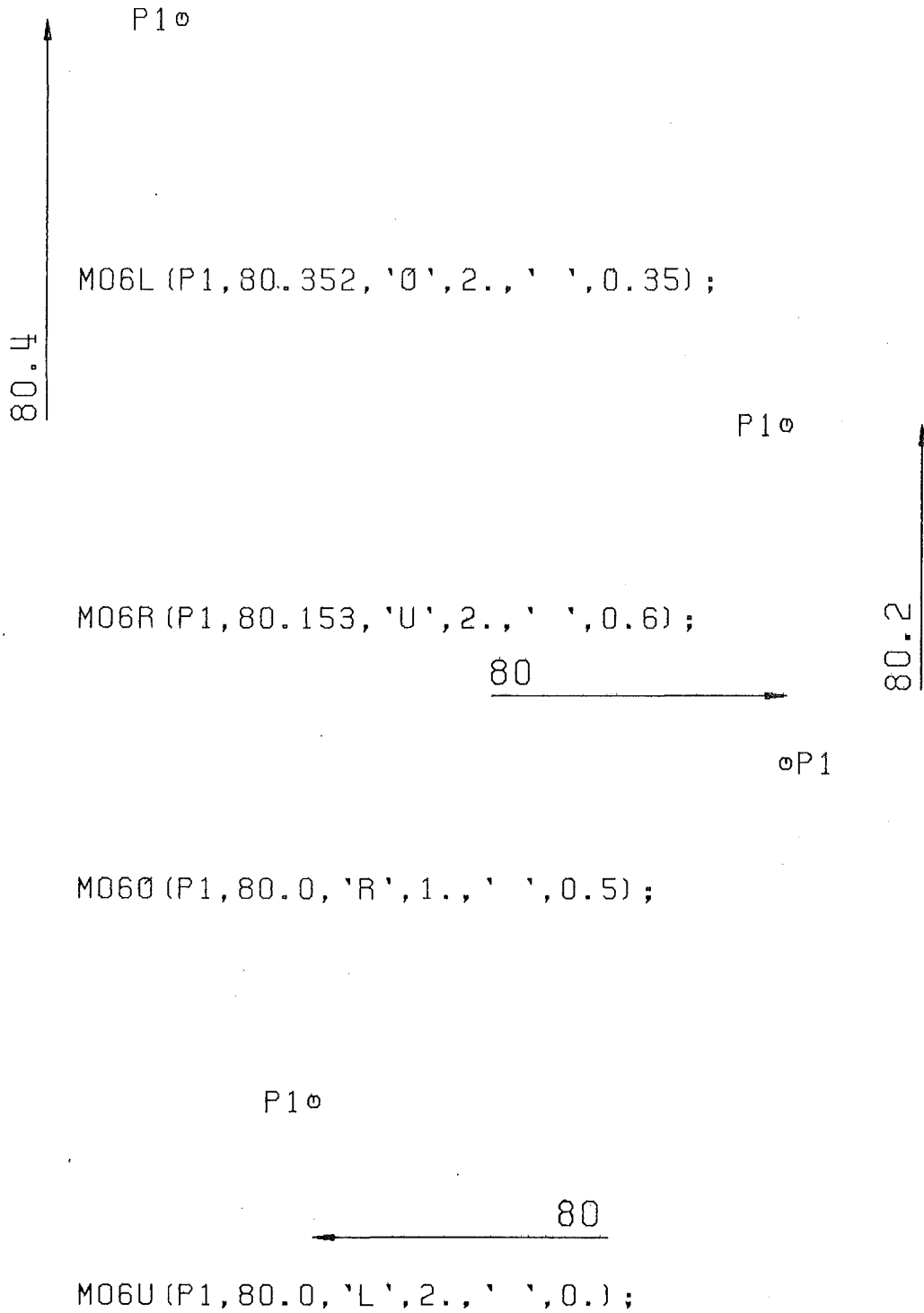
Erläuterung

P1 : Konturpunkt, Typ POINT2
lage : Lage des Konturpunktes, 'L' oder 'R'
abstand : Abstand des Maßtextes vom Konturpunkt in cm.
(Länge der Maßhilfslinie)
maßzahl : Maßtext und Maßlückengröße
info : Zusatzinformation, siehe Kap.2.2, S.13,
plz : Platzierung der Maßzahl, siehe Kap.2.2, S.13 ff.

Standardwerte

info : "(Leerstring)
abstand : 1 cm
plz : 0.
lage : 'R' bei MØ5U
'L' " MØ5R
'L' " MØ50
'R' " MØ5L

Beispiele



Syntax

MØ6U }
MØ6R } (P1, maßzahl, [lage], [abstand], [info], [plz])
MØ6O }
MØ6L }

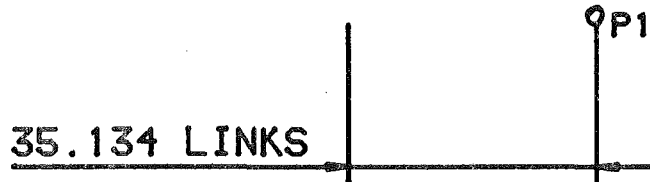
Erläuterung

P1 : Konturpunkt, Typ POINT2
lage : Lage des Konturpunktes 'L' oder 'R'
abstand : Abstand des Maßtextes vom Konturpunkt in cm.
(Länge der Maßhilfslinie)
maßzahl : Maßtext und Maßlückengröße
info : Zusatzinformation, siehe Kap. 2.2, S.13.
plz : Platzierung der Maßzahl, siehe Kap. 2.2, S.13 ff.

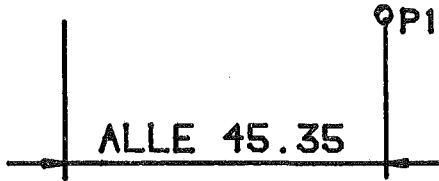
Standardwerte

info : "(Leerstring)
abstand : 1 cm
plz : 0.
lage : 'R' bei MØ6U
'L' " MØ6R
'L' " MØ6O
'R' " MØ6L

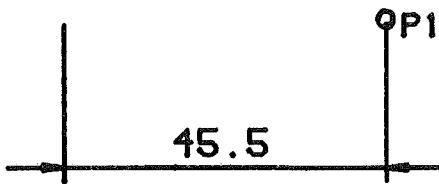
Beispiele



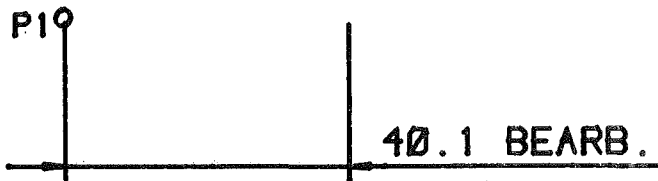
M11U(P1,35.1344,'R',2.,'\$S3\$TNLINKS',-3),



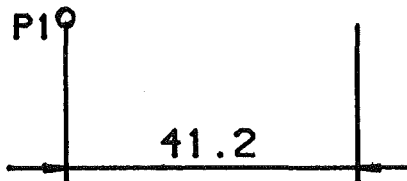
M11U(P1,45.352,'R',2.,'\$S2\$TVALLE',2),



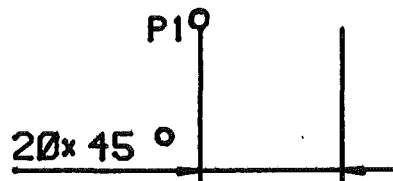
M11U(P1,45.52,'R',2.,'\$00.2\$U-0.15',2),



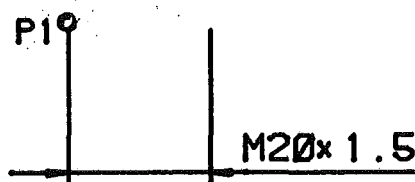
M11U(P1,40.06,'L',2.,'\$0-1\$TNBEARB.',



M11U(P1,41.153,'L',2.,'\$00.1\$U0.2',1),



M11U(P1,20.0,'L',2.,'\$FX45',2),



M11U(P1,20.0,'L',2.,'\$MX1.5',1),

Syntax

M11U }
M11R } (P1, maßzahl, [lage], [abstand], [info], [plz])
M11O }
M11L }

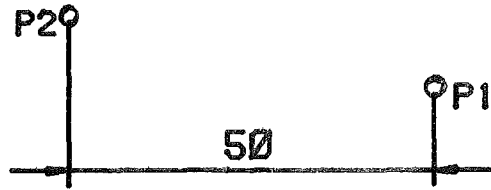
Erläuterung

P1 : Konturpunkt. Typ POINT2
lage : Lage des Konturpunktes P1.
abstand : Abstand des Maßtextes von dem Konturpunkt in cm.
(Länge der Maßhilfslinie)
maßzahl : Maßtext und Maßlückengröße
info : Zusatzinformation, s.Kap.2.2, S.13.
plz : Platzierung der Maßzahl, Kap. 2.2, S.13 ff.

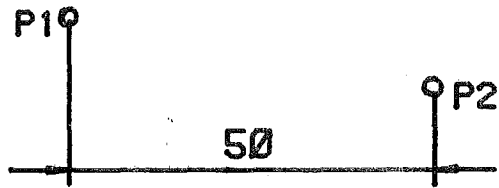
Standardwerte

info : "
abstand : 1 cm
lage : 'L' bei M11U 1.
'R' " M11R 2.
'R' " M11O 2.
'L' " M11L 1.

Beispiele



M12U(P1 ,P2 ,50.0 , 'R' , 1. , ' ' , 0.5) ,



M12U(P1 ,P2 ,50.0 , 'L' , 2. , ' ' , 0.5) ,

M12
Maßroutine
Längemaß

Syntax

M12U }
M12R } (P1, P2, maßzahl, [lage], [abstand], [info], [plz])
M12O }
M12L }

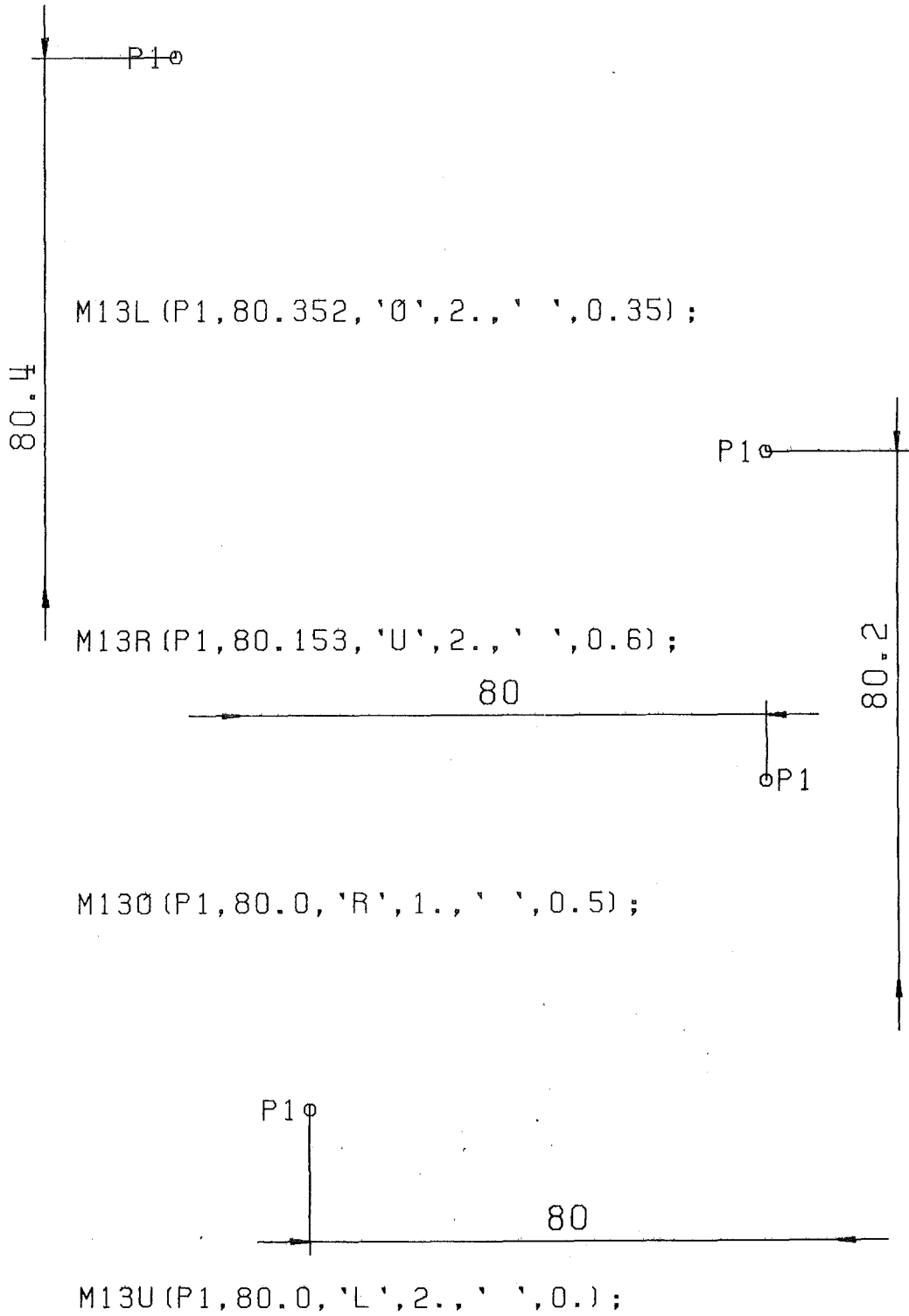
Erläuterung

P1, P2 : Erster und zweiter Konturpunkt. Typ POINT2
lage : Lage des Konturpunktes P1, s.Kap.2.2, S.
abstand : Abstand des Maßtextes von den Konturpunkten in cm.
(Länge der Maßhilfslinie)
maßzahl : Maßtexte und Maßlückengröße
info : Zusatzinformation, s. Kap. 2.2, S.13.
plz : Platzierung der Maßzahl, Kap. 2.2, S.13 ff.

Standardwerte

info : "
abstand : 1 cm
plz : 0.
lage : 'R' bei M12U
'L' " M12R
'L' " M12O
'R' " M12L

Beispiele



Syntax

M13U }
M13R } (P1, maßzahl, [lage], [abstand], [info], [plz])
M13O }
M13L }

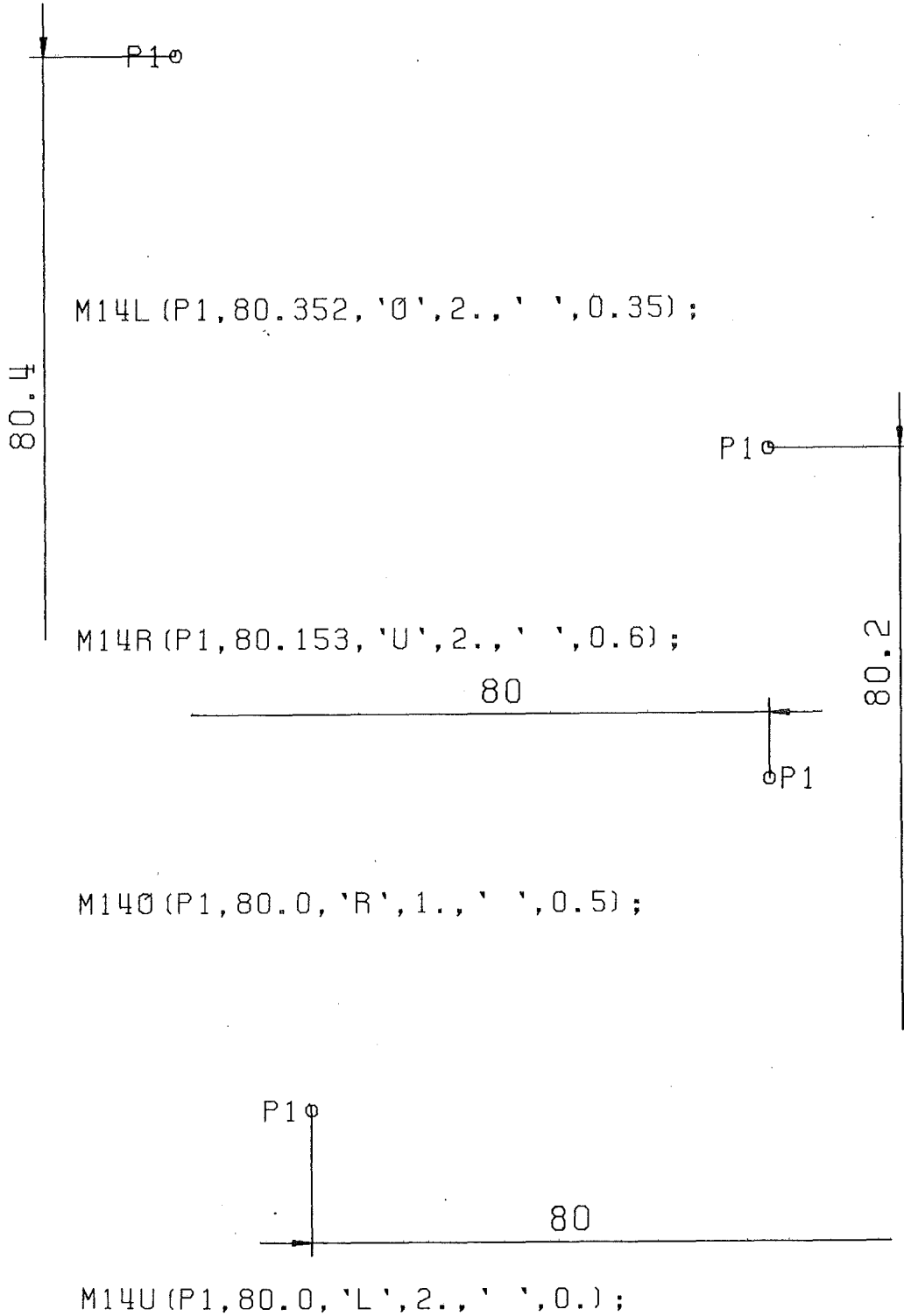
Erläuterung

- P1 : Konturpunkt. Typ POINT2
lage : Lage des Konturpunktes P1.
abstand : Abstand des Maßtextes von dem Konturpunkt in cm.
(Länge der Maßhilfslinie)
maßzahl : Maßtext und Maßlückengröße
info : Zusatzinformation, s.Kap. 2.2, S.13.
plz : Platzierung der Maßzahl, Kap. 2.2, S.13 ff.

Standardwerte

info : "
abstand : 1 cm
lage : 'L' bei M13U 2.
'R' " M13R 1.
'R' " M13O 1.
'L' " M13L 2.

Beispiele



M14
Maßroutine
Längenmaß

Syntax

M14U }
M14R } (P1, maßzahl, [lage], [abstand], [info], [plz])
M14O }
M14L }

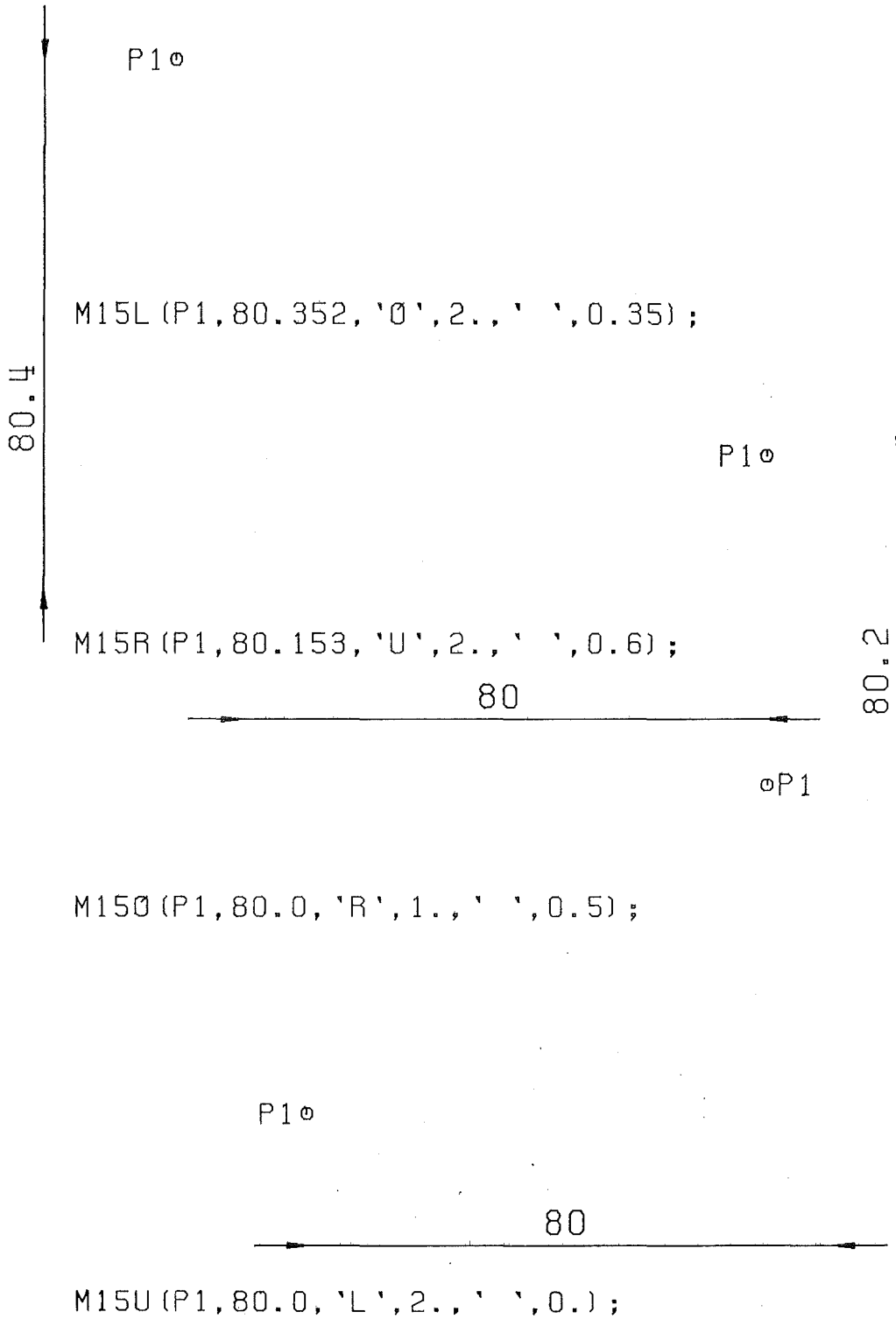
Erläuterung

P1 : Konturpunkt. Typ POINT2
lage : Lage des Konturpunktes P1.
abstand : Abstand des Maßtextes von dem Konturpunkt in cm.
(Länge der Maßhilfslinie)
maßzahl : Maßtext und Maßlückengröße
info : Zusatzinformation, s.Kap.2.2, S.13.
plz : Platzierung der Maßzahl, Kap. 2.2, S.13 ff.

Standardwerte

info : "
abstand : 1 cm
plz : 0.
lage : 'L' bei M14U
'R' " M14R
'R' " M14O
'L' " M14L

Beispiele



Syntax

M15U }
M15R } (P1, Maßzahl, [lage], [abstand], [info], [plz])
M15O }
M15L }

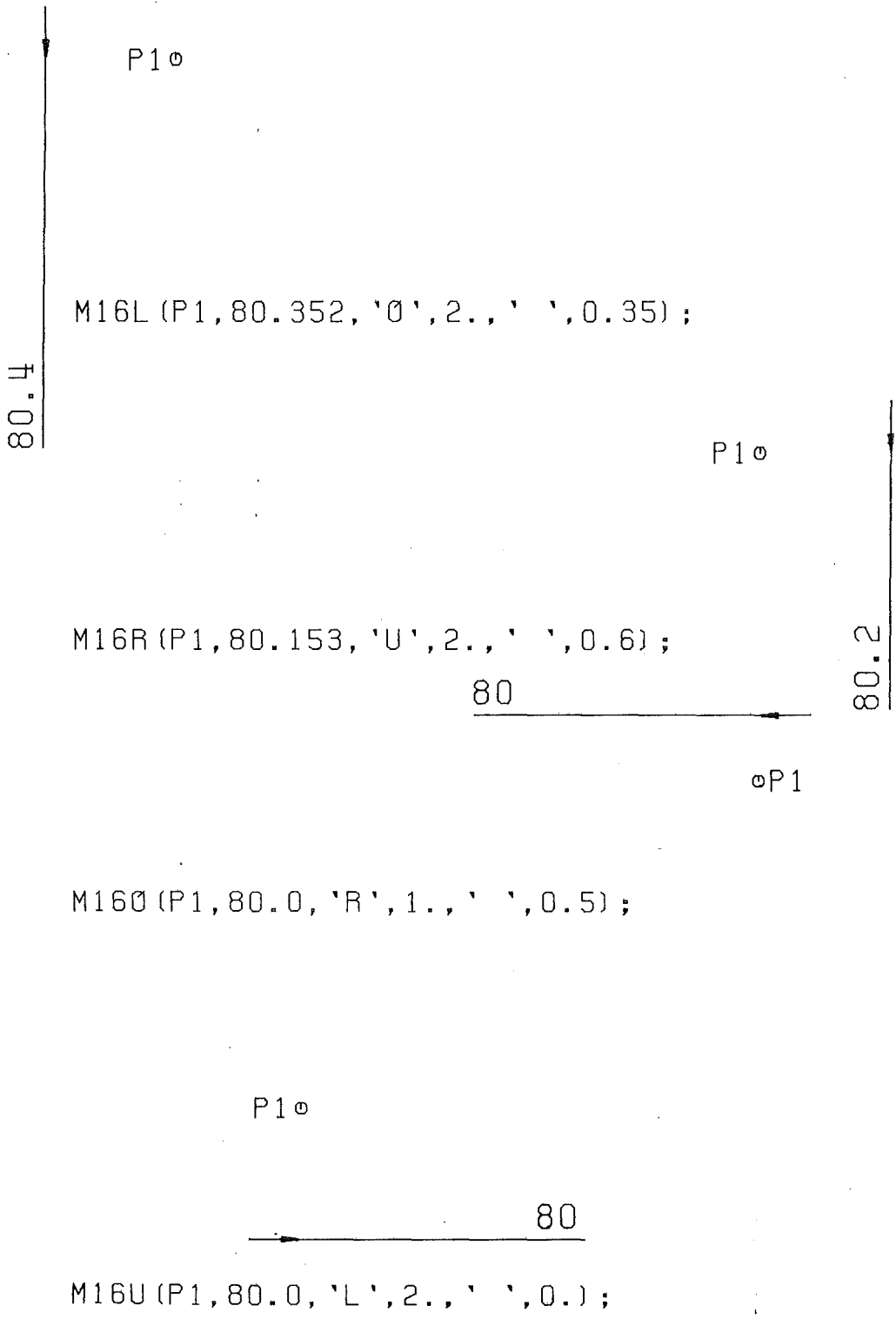
Erläuterung

P 1 : Konturpunkt, Typ POINT2
lage : Lage des Konturpunktes P1.
abstand : Abstand des Maßtextes von dem Konturpunkt in cm.
(Länge der Maßhilfslinie)
maßzahl : Maßtext und Maßlückengröße
info : Zusatzinformation, s.Kap. 2.2, S.13.
plz : Platzierung der Maßzahl, Kap. 2.2, S.13 ff.

Standardwerte

info : "
abstand : 1 cm
plz : 0.
lage : 'R' bei M15U
'L' " M15R
'L' " M15O
'R' " M15L

Beispiele



M16
Maßroutine
Längenmaß

Syntax

M16U }
M16R } (P1, maßzahl, [lage], [abstand], [info], [plz]);
M16O }
M16L }

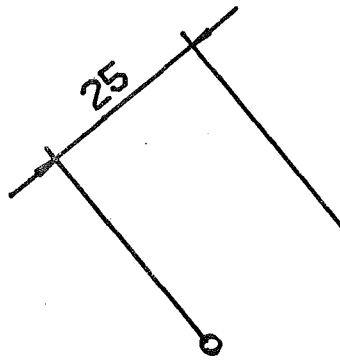
Erläuterung

P1 : Konturpunkt, Typ POINT2
lage : Lage des Konturpunktes P1.
abstand : Abstand des Maßtextes von dem Konturpunkt in cm.
(Länge der Maßhilfslinie)
maßzahl : Maßtext und Maßlückengröße
info : Zusatzinformation, s.Kap. 2.2, S.13.
plz : Platzierung der Maßzahl, Kap. 2.2, S.13 ff.

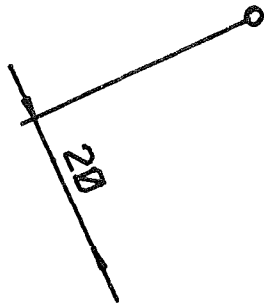
Standardwerte

info : "
abstand : 1 cm
plz : 0.
lage : 'R' bei M16U
'L' " M16R
'L' " M16O
'R' " M16L

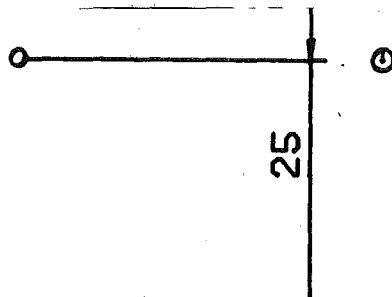
Beispiele



```
M11G(P(1),25., 'L',130.,3.3, ' ',0.),
```

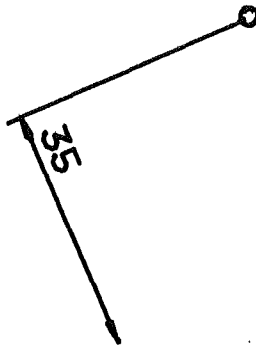


```
M13G(P(2),20., 'L',205.,3.3, ' ',0.),
```

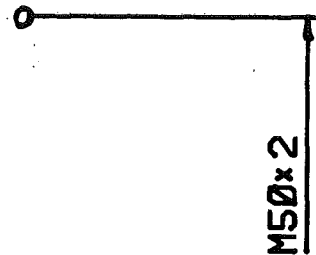


```
M14G(P(5),25., 'R',0.,4., ' ',0.),
```

```
M15G(P(6),25., 'R',0.,4., ' ',0.),
```



```
M03G(P(2),35., 'L',205.,3.5, ' ',0.2),
```



```
M04G(P(5),50., 'R',0.,4., 'M2',0.),
```

MxxG
Maßroutine
Längenmaß

Syntax

M01G }
M03G }
M04G }
M05G }
M06G } (P1, maßzahl, [lage], [winkel], [abstand], [info],
M11G } [plz])
M13G }
M14G }
M15G }
M16G }

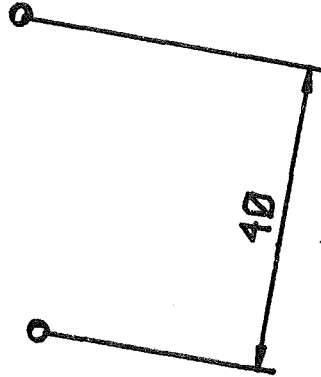
Erläuterung

P1 : Konturpunkt. Typ POINT2
lage : Lage des Konturpunktes, 'L' oder 'R'.
winkel : Startwinkel der Maßhilfslinie
abstand : Abstand des Maßtextes vom Konturpunkt.
info : Zusatzinformation, s.Kap. 2.2, S.13.
plz : Platzierung der Maßzahl, Kap.2.2, S.13 ff.

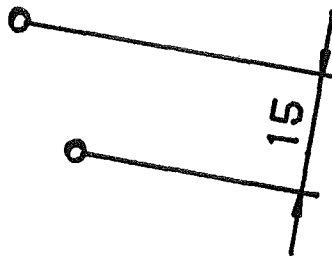
Standardwerte

winkel : 0
lage : 'L'
info : "
plz : 0
abstand : 1 cm

Beispiel



M02G(P(3),P(4),10., 'R',-10,3.8., ' ',0.7),



M12G(P(3),P(4),15., 'R',-10.,3.8., ' ',0.2),

M02G
M12G
Maßroutine
Längenmaß

Syntax

M02G }
M12G } (P1, P2, maßzahl, [lage], [winkel], [abstand],
[info], [plz])

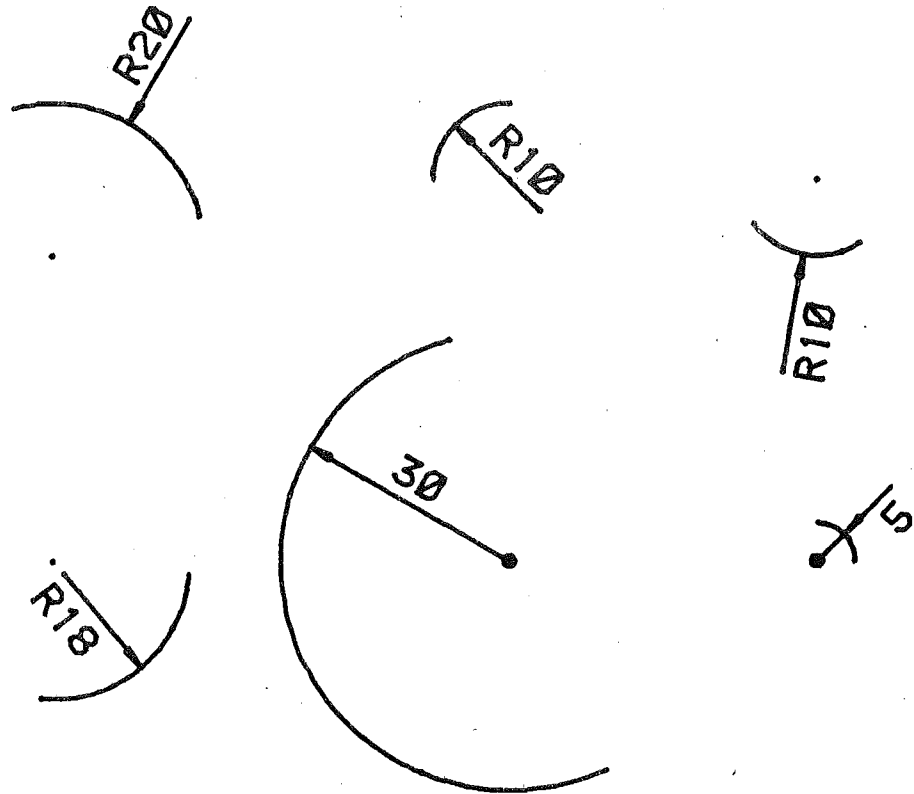
Erläuterung

P1, P2 : Erster und zweiter Konturpunkt. Typ POINT2
lage : Lage des Konturpunktes P1. Kap. 2.2, S
winkel : Startwinkel der Maßhilfslinie
abstand : Abstand des Maßtextes vom Konturpunkt
info : Zusatzinformation, s.Kap. 2.2, S.13.
plz : Platzierung der Maßzahl, Kap. 2.2, S.13 ff.

Standardwerte

winkel : 0
lage : 'R' bei M02G
'L' " M12G
abstand : 1 cm
info : "
plz : 0.

Beispiele



AUFRUFE VON LINKS OBEN NACH RECHTS UNTEN:

MIRA(P(1),20,'R',60,' ',0),

MIRA(P(2),10,'L',135,' ',0),

MIRA(P(3),10,'R',260,' ',1),

MIRA(P(4),18,'L',310,' ',1),

M2RA(P(5),30,'L',150,' ',0),

M2RA(P(6),5,'R',45,' ',1),

M1RA
M2RA
Maßroutine
Radienmaß

Syntax

M1RA }
M2RA } (P1, radius, [lage], [startwinkel], [info], [plz])

Erläuterung

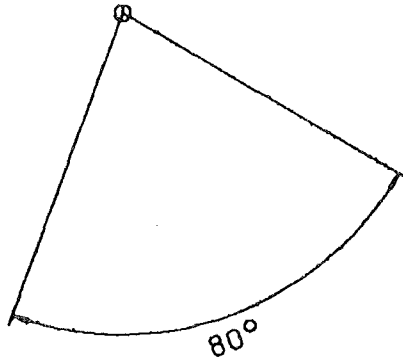
P1 : Konturpunkt. Typ POINT2
radius : Radius, gleichzeitig Maßzahl
lage : Lage des Konturpunktes, 'L' oder 'R'
startwinkel: Startwinkel der Maßhilfslinie
info : Zusatzinformation, s. Kap.2.2, S.13
plz : Platzierung der Maßzahl, s.Kap.2.2, S. 13 ff.

Standardwerte

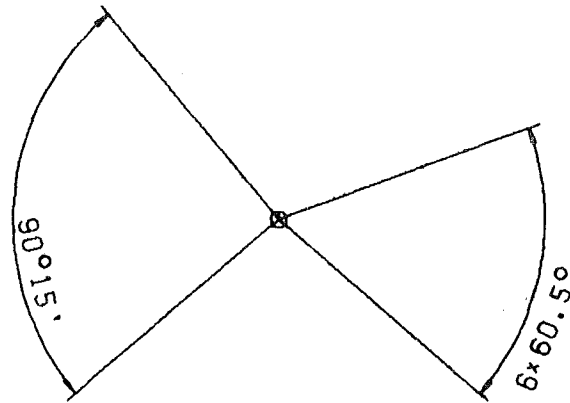
lage : 'L'
startwinkel: 0
info : "
plz : 0.

Beispiele

MO1W(P(1),80.,'R',-30.,6.0,' ',0.);

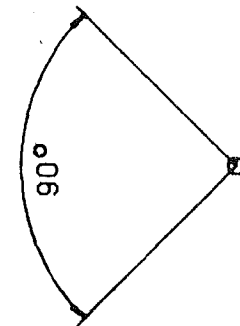
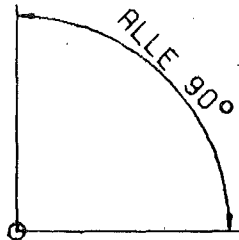


MO1W(P(3),90.25,'L',130.,5.0,'*WM',-0.7);



MO1W(P(2),60.55,'R',20.,5.0,'*TK8',0.3);

MO1W(P(4),90.,'R',90.,4.0,'*TVALLE',0.);



MO1W(P(5),90.,'L',135.,4.0,' ',-0.5);

MxxW
Maßroutine
Winkelmaß

Syntax

MO1W
MO3W
MO4W
MO5W
MO6W
M11W
M13W
M14W
M15W
M16W

(P1, winkel, [lage], [startwinkel], [abstand],
[info], [plz])

Erläuterung

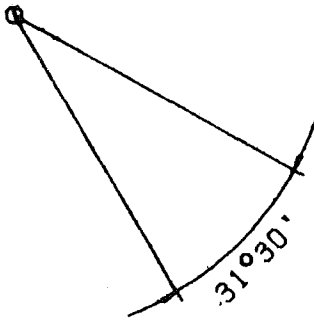
P1 : Konturpunkt. Typ POINT2
winkel : Winkel, gleichzeitig Maßzahl
lage : Lage des Konturpunktes, 'L' oder 'R'.
startwinkel: Startwinkel der Maßhilfslinie
abstand : Abstand des Maßtextes vom Konturpunkt
info : Zusatzinformation, s.Kap.2.2, S.13.
plz : Platzierung der Maßzahl, Kap.2.2, S.13 ff.

Standardwerte

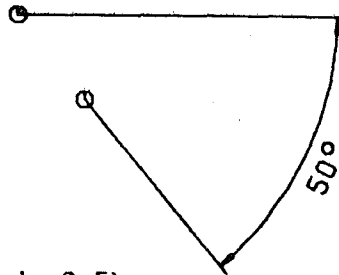
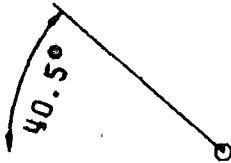
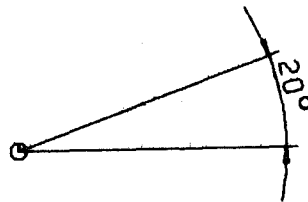
startwinkel : 0
lage : 'L'
info : "
plz : 0.
abstand : 1 cm

Beispiele

M11W(P1,30.5,'R',-30,6,'\$HM',0.);

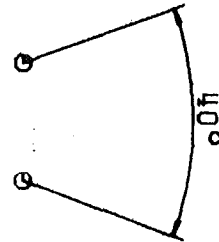
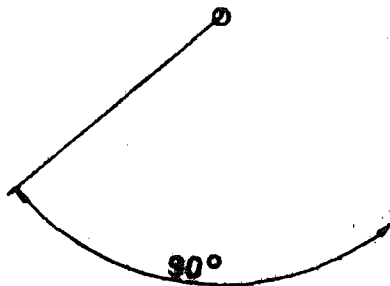


M11W(P1,20,'L',0.5,'',1.);



M03W(P1,40.5,'R',160,4,'',-0.5);

M02W(P1,P2,50,'R',0.6,'',0);



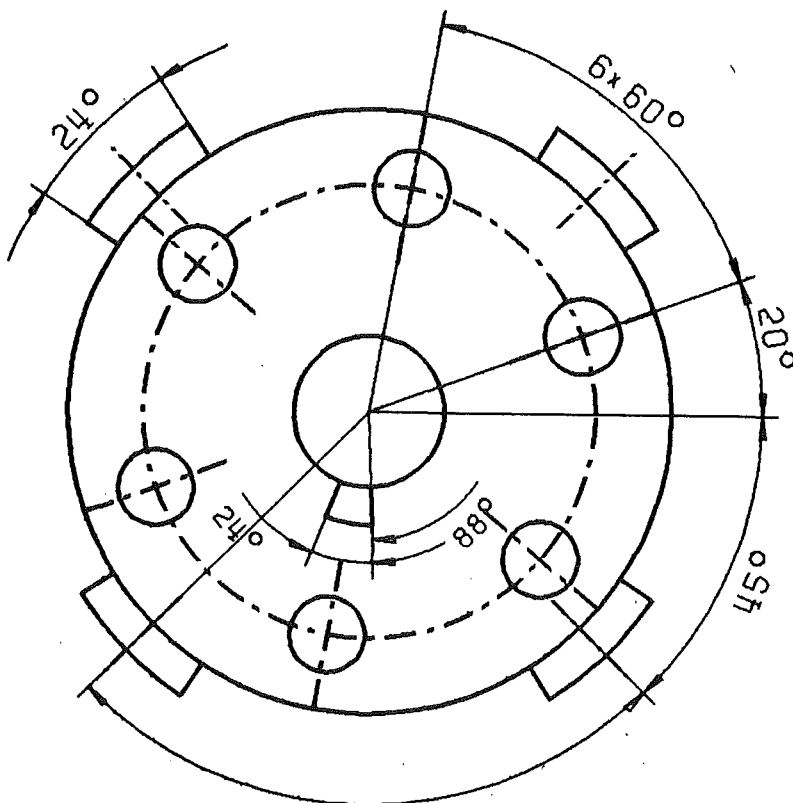
M04W(P1,90,'L',220,5,'',-0.5);

M02W(P1,P2,40,'R',20,3,'',0);

MxxW

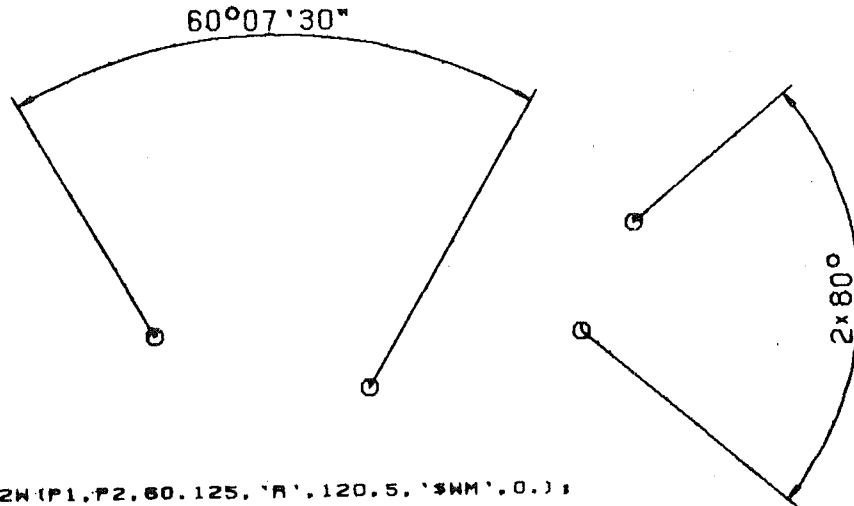
Beispiele

```
CHANGE STANDARD PEN(4);  
CHANGE UNITS LENGTH(MM);  
PLOT(DREHTEIL);  
CHANGE STANDARD PEN(1);  
SET P1=POINT2(100,100);  
SET M1=M01W(P1,60,'L',20,5.2,'$TK6',0.);  
SET M2=M03W(P1,20,'L',00,5.2,'',0.);  
SET P13=POINT2(140,100);  
SET P14=POINT2(135,100);  
SET P14=ROT2(P14,-45 DEG,P1);  
SET M3=M02W(P13,P14,45,'R',00,1.2,'',0.);  
SET M4=M03W(P1,90,'L',-135,5.2,'$TN4x am Umkreis',0.);  
SET P8=POINT2(140,100);  
SET P8=ROT2(P8,147 DEG,P1);  
SET P9=ROT2(P8,-24 DEG,P1);  
SET M7=M12W(P8,P9,24,'R',147,1.2,'',0.);  
CHANGE MASS BH(0.20);  
SET P3=POINT2(110,100);  
SET P3=ROT2(P3,-88 DEG,P1);  
SET P4=ROT2(P3,-24 DEG,P1);  
SET M5=M02W(P3,P4,24,'R',-88,1.0,'',2.);  
SET M6=M04W(P1,88,'R',00,1.7,'',0.);  
PLOT(M1,M2,M3,M4,M5,M6,M7);
```



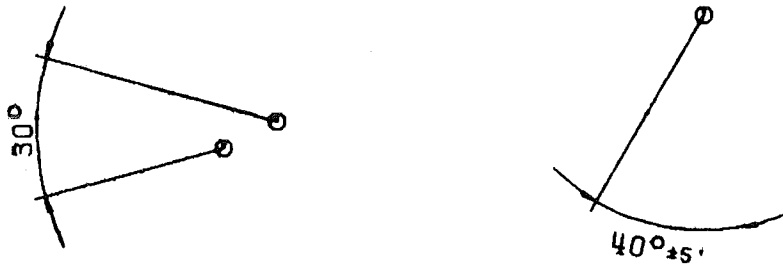
90° 4x am Umkreis

Beispiele



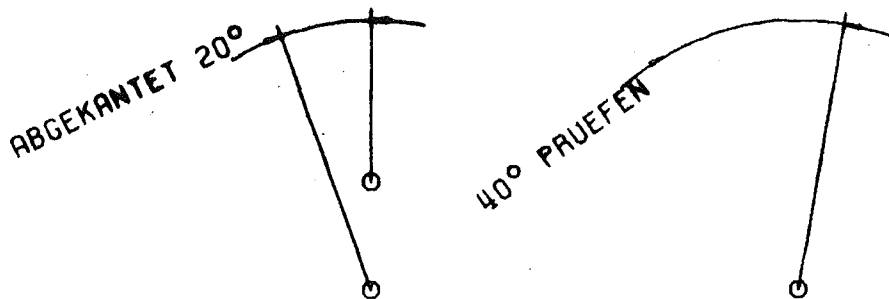
M02W (P1, P2, 60, 125, 'R', 120, 5, '\$WM', 0.);

M02W (P1, P2, 80, 'L', -40, 5, '\$TK2', -0.5);



M12W (P1, P2, 30, 'L', 165, 4, 5, ' ', -2);

M13W (P1, 40, 'L', 240, 4, 'US', -1);



M12W (P1, P2, 20, 'R', 110, 5, '\$TVABGEKANTET', 0);

M14W (P1, 40, 'L', 80, 5, '1' '\$TNPRUEFEN', -1);

MO2W
M12W
Maßroutine
Winkelmaß

Syntax

MO2W }
M12W } (P1, P2, winkel, [lage], [startwinkel], [abstand],
[info], [plz])

Erläuterung

P1, P2 : Erster und zweiter Konturpunkt. Typ POINT2
winkel : Winkel, gleichzeitig Maßzahl
lage : Lage des Konturpunktes P1. Kap. 2.2, S.
startwinkel: Startwinkel der Maßhilfslinie
abstand : Abstand des Maßtextes vom Konturpunkt
info : Zusatzinformation, s.Kap. 2.2, S.13.
plz : Platzierung der Maßzahl, Kap.2.2, S.13 ff.

Standardwerte

lage : 'R' bei MØ2W
'L' " M12W
abstand : 1 cm
info : "
plz : 0.

MARW
Bemaßung
Zusatzroutine

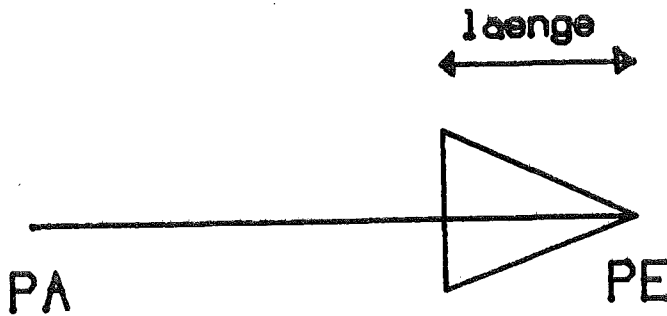
Syntax

```
CALL MARW( QQ, coll, PA, PE, länge);
```

Erläuterung

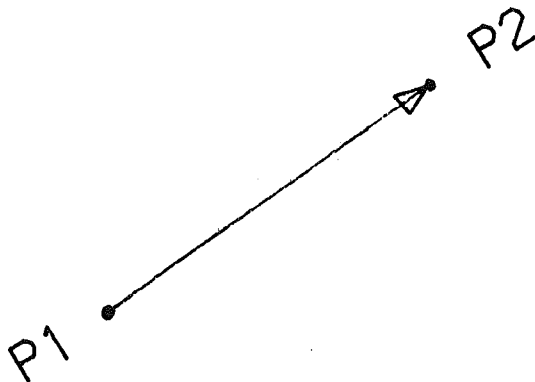
Maßpfeil

- coll : zurückgelieferte COLLECTION
- PA : Startpunkt der Pfeillinie, graph. Objekt vom Typ POINT2
- PE : Pfeilspitze, graph. Objekt vom Typ POINT2
- länge : Länge der Pfeilspitze in mm



Beispiel

```
CALL MARW(QQ,K1,P1,P2,5.);
```



Syntax

```
CALL MEZH(QQ, coll, PM, r, lage, buchstabe);
```

Erläuterung

Einzelheitvergrößerung

Die Routine zeichnet um einen angegebenen Punkt einen Kreis mit angegebenem Durchmesser. Bezugsbuchstabe zur Kennzeichnung der Einzelheit kann entsprechend positioniert werden.

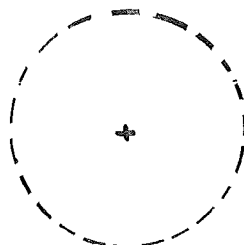
coll : zurückgelieferte COLLECTION
PM : Mittelpunkt des Kreises, graph.Objekt vom Typ POINT2
r : Radius des Kreises in cm
lage : Lage des Bezugsbuchstabens
 'L' = links
 'R' = rechts
 'O' = oben
 'U' = unten

buchstabe: ein Buchstabe zur Kennzeichnung der Einzelheit

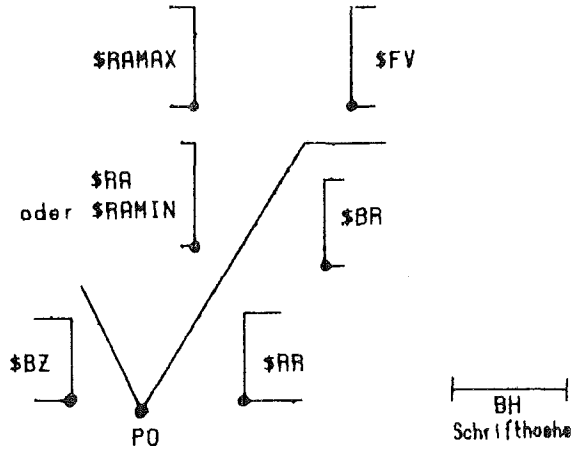
Beispiel

```
CALL MEZH(QQ, K1, P0, 2.0, 'L', 'X');
```

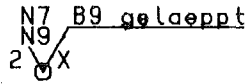
Einzelheit



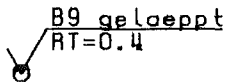
Beispiele



```
CALL Mbaz (QQ,MCOL,P1,'L','$BR2.5');
```



```
CALL Mbaz (QQ,MCOL,P1,'0','$RAMAXN9$RAMINN7$RR3$BZ2
$FVB9 gelaeppt');
```



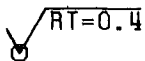
```
CALL Mbaz (QQ,MCOL,P1,'0','$FVB9 gelaeppt$BRRt=0.4');
```



```
CALL Mbaz (QQ,MCOL,P1,'0','$SN$RA3.2');
```



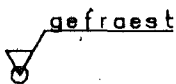
```
CALL Mbaz (QQ,MCOL,P1,'0','$BZ2');
```



```
CALL Mbaz (QQ,MCOL,P1,'0','$BRRt=0.4');
```



```
CALL Mbaz (QQ,MCOL,P1,'0','$BR2.5');
```



```
CALL Mbaz (QQ,MCOL,P1,'0','$SM$FVgefraest');
```

MBAZ
Bemaßung
Zusatzroutine

Syntax

CALL MBAZ(QQ, coll, PO, lage, info);

Erläuterung

Diese Routine gilt für Symbole und die zusätzlichen Angaben für die Kennzeichnung der Oberflächenbeschaffenheit in technischen Zeichnungen.

coll	:	zurückgelieferte Collection
PØ	:	Position des Bearbeitungszeichens, graph.Objekt vom Typ POINT2.
lage	:	Lage des Symbols, 'O' und 'L'
info	:	
§SG		Grundsymbol (Voreinstellung)
§SN		nicht materialabtrennende Bearbeitung
§SM		materialabtrennende Bearbeitung
§SZ...		Bearbeitungszugabe
§RA...		Mittenrauwert
§RAMAX...		max.Mittenrauwert
§RAMIN...		min.Mittenrauwert
§FV.....		Fertigungsverfahren
§BR.....		Bezugsstrecke und/oder andere Rauheitsmeßgrößen
§RR.		Rillenrichtung
1	=	
2	⊥	
3	X	
4	M	
5	C	
6	R	
§WH....		Kennzeichen für mehrmalige Wiederholung des Zeichens

✓
✓

MH1-länge : s.Abb.

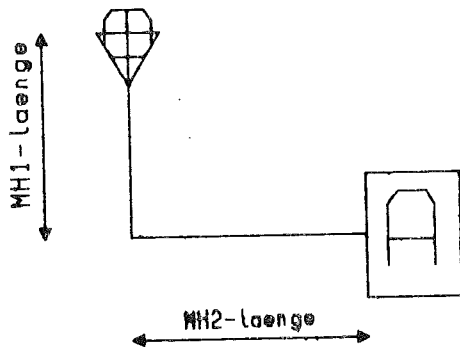
Länge in cm

bezugsbuchstabe:

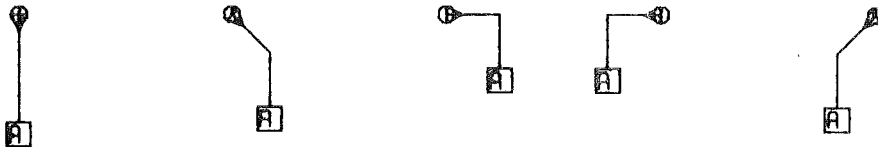
§B nach dem Kennzeichen §B wird nur ein Zeichen interpretiert.

MH2-länge : s.Abb.

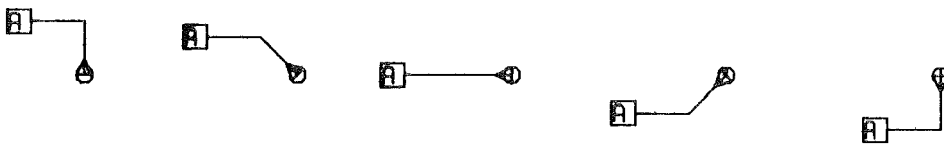
Länge in cm



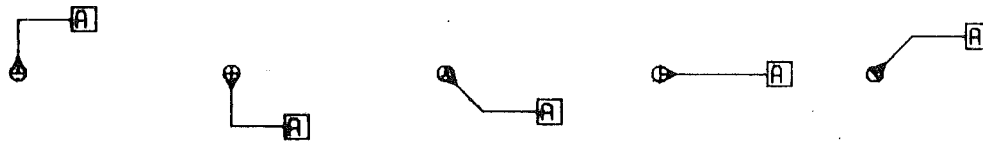
Beispiele



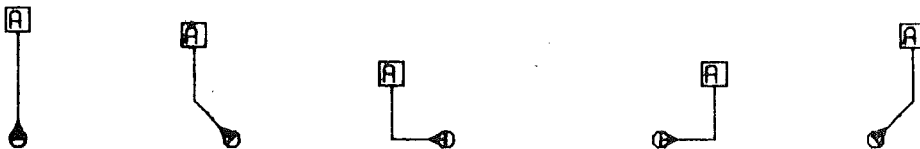
```
CALL MBEZ (QQ,MCOL,P1,'U',WINKEL,1.,'$BA',1.);
```



```
CALL MBEZ (QQ,MCOL,P1,'L',WINKEL,1.,'$BA',1.);
```



```
CALL MBEZ (QQ,MCOL,P1,'R',WINKEL,1.,'$BA',1.);
```



```
CALL MBEZ (QQ,MCOL,P1,'O',WINKEL,1.,'$BA',1.);
```

Syntax

CALL MBEZ(QQ,coll,PO,lage,winkel,MH1-länge,bezugsbuchstabe,
MHZ-länge);

Erläuterung

Bezugsbuchstabe für Lage- und Formtoleranzen

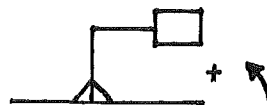
- coll : Zurückgelieferte COLLECTION
- PØ : Punkt auf der Kontur für Bezugsdreieck, Objekt vom Typ POINT2
- lage : Lage des Bezugsbuchstabens
 - 'L' links vom Bezugspunkt
 - 'R' rechts vom Bezugspunkt
 - 'O' oben vom Bezugspunkt
 - 'U' unten vom Bezugspunkt
- winkel : Winkel der Konturlinie, auf der der Bezugspunkt liegt

Lage='L'



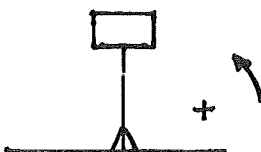
Winkel=0°

Lage='R'



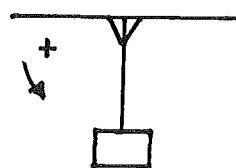
Winkel=0°

Lage='O'



Winkel=0°

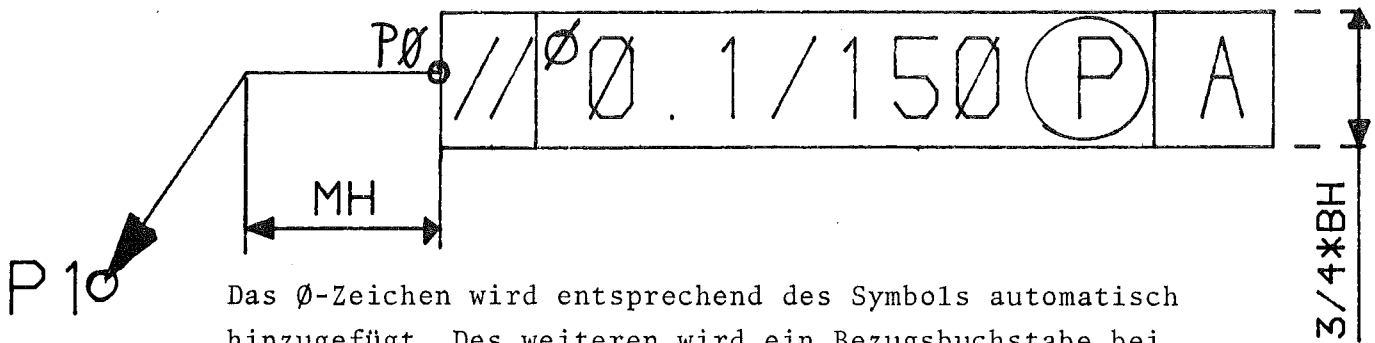
Lage='U'



Winkel=0°

Symbol :

1		Geradheit
2		Ebenheit
3		Rundheit
4		Zylinderform
5		Linienform
6		Flächenform
7		Parallelität für tolerierte Achsen
8		Parallelität für tolerierte Flächen
9		Rechtwinkeligkeit
10		Neigung (Winkeligkeit)
11		Position
12		Symmetrie
13		Koaxialität
14		Planlauf, Rundlauf



Das \varnothing -Zeichen wird entsprechend des Symbols automatisch hinzugefügt. Des weiteren wird ein Bezugsbuchstabe bei Formtoleranz unterdrückt.

MLFT
Bemaßung
Zusatzroutine

Syntax

CALL MLFT(QQ, coll, P1, P0, MH-länge, symbol, info);

Erläuterung

Form- und Lagetoleranzen werden zusätzlich zu Maßtoleranzen angegeben, um Funktion und Austauschbarkeit zu gewährleisten.

coll : Zurückgelieferte COLLECTION

P1 : Punkt auf der Kontur für den Hinweispfeil,
graph. Objekt vom Typ POINT2

P0 : Position der Form- und Lagetoleranzen,
graph.Objekt vom Typ POINT2

MH-länge : > 0 siehe Abbildung
= 0 keine Verbindung zw.P0 und P1

info:

\$M... Toleranzwert z.B. 0.03

\$MMCM Maximal Material-Bedingung auf Toleranzwert
bezogen






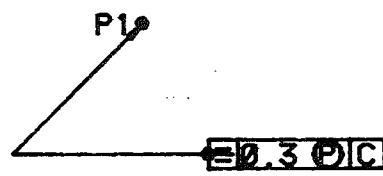
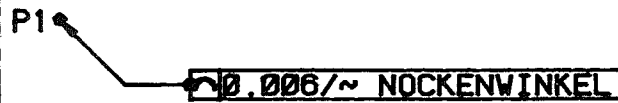
\$PTZ Projektierte Toleranzen

\$MMCB Maximal Material Bedingung auf Bezugsbuch-
staben bezogen

\$B.... Bezugsbuchstaben

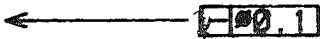
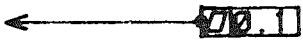
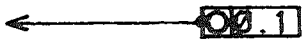
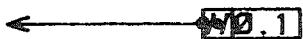
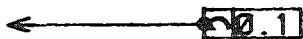
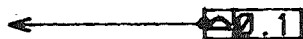
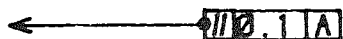
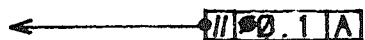
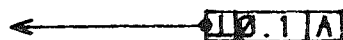
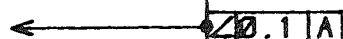
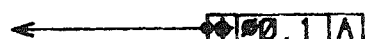
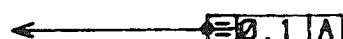
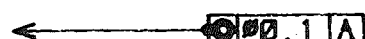

\$T.... Beschreibender Text

CALL MLFT (QQ ,MCOL ,P1 ,P2 ,MH ,SYMB ,ZI) ;

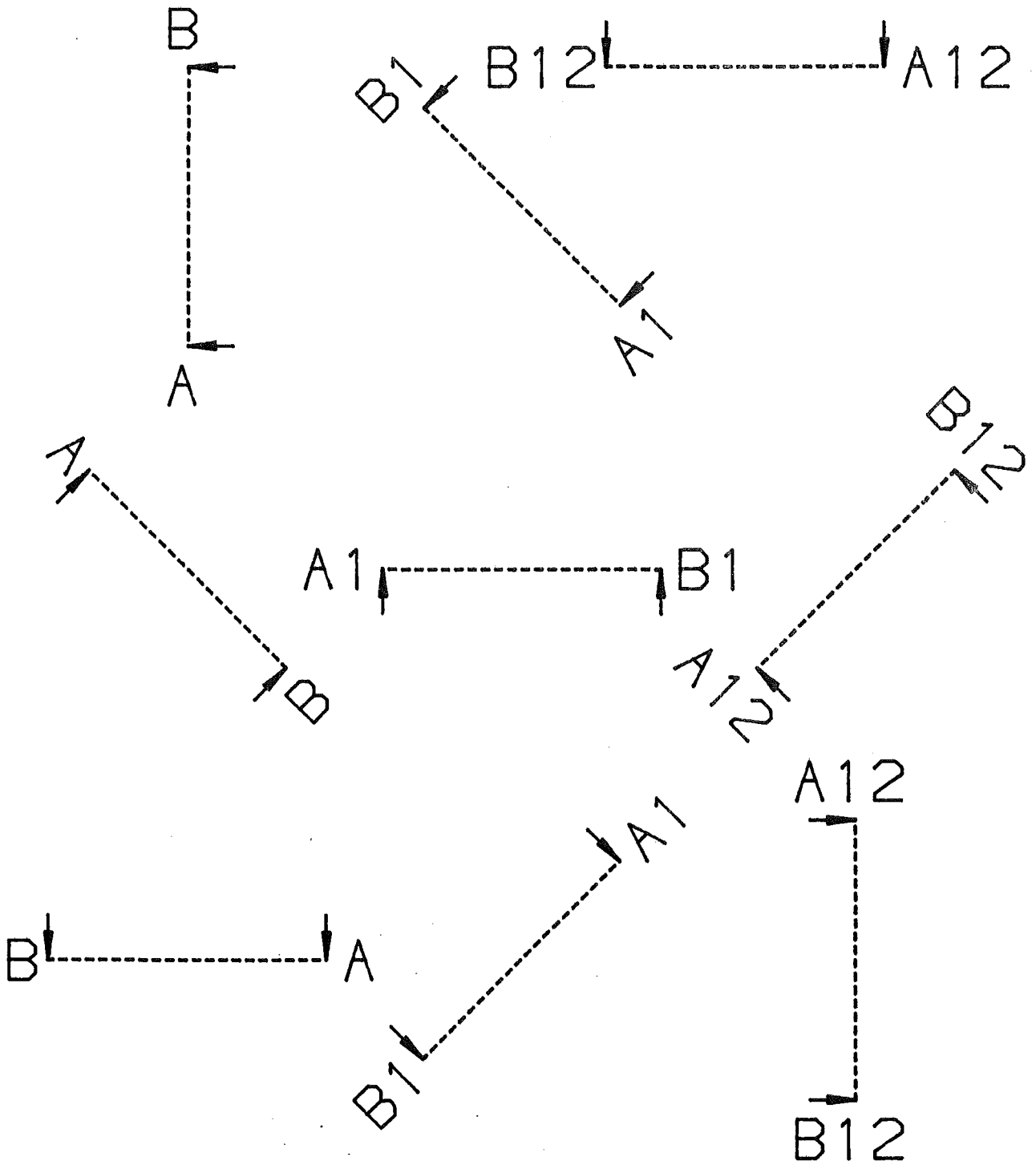
	MH (CM)	SYMB	ZI (ZUSATZINFORMATION)
	1.	7	⊕M0.1⊕BA
	1.	9	⊕M0.1/150⊕BA⊕PTZ
	1.	14	⊕M0.02⊕BA
	0.	9	⊕M0.015⊕BA⊕TNICHT KONVEX
	1.5	11	⊕M0.1⊕MMCM⊕BAC⊕MMCB
	3.0	12	⊕M0.3⊕BC⊕PTZ
	1.0	5	⊕M0.006/~ NOCKENWINKEL⊕BAB

LAGE- U. FORMTOLERANZEN NACH DIN 7184

CALL MLFT (QQ ,MCOL ,P1 ,P2 ,MH ,SYMB ,ZI) ;

	MH (CM)	SYMB	ZI (ZUSATZINFORMATION)
← 	1.	1	\$MØ.1\$BA
← 	1.	2	\$MØ.1\$BA
← 	1.	3	\$MØ.1\$BA
← 	1.	4	\$MØ.1\$BA
← 	1.	5	\$MØ.1\$BA
← 	1.	6	\$MØ.1\$BA
← 	1.	7	\$MØ.1\$BA
← 	1.	8	\$MØ.1\$BA
← 	1.	9	\$MØ.1\$BA
← 	1.	10	\$MØ.1\$BA
← 	1.	11	\$MØ.1\$BA
← 	1.	12	\$MØ.1\$BA
← 	1.	13	\$MØ.1\$BA
← 	1.	14	\$MØ.1\$BA

LAGE- U. FORMTOLERANZEN NACH DIN 7184



CALL MSVL (QQ ,MCOL ,PT ,2 ,'R' ,'\$AA\$EB') ,

SCHNITTVERLAUF

MSVL
Bemaßung
Zusatzroutinen

Syntax

```
CALL MSVL(QQ, coll, PF, anzahl, lage, info);
```

Erläuterung

Zur Kennzeichnung eines Schnittverlaufes wird ein Linienzug mit einer Kennzeichnung aufbereitet.

coll : zurückgelieferte COLLECTION

PF : Punktfeld für den Linienzug, graph. Objekt vom Typ POINT2

anzahl : Anzahl Punkte im Punktfeld (muß ≥ 2 sein)

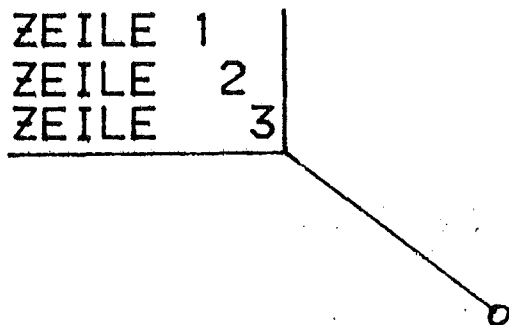
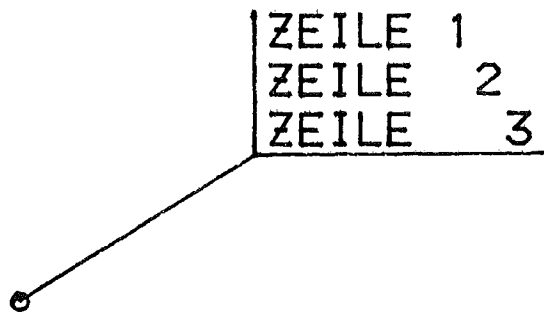
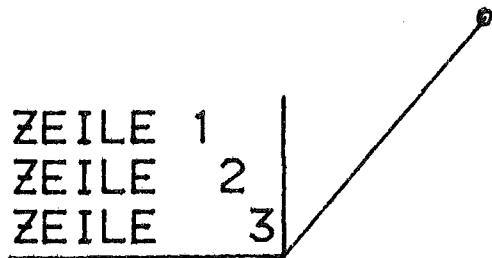
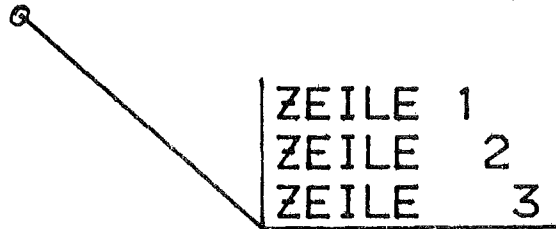
lage : Lage der Ansichtspfeile für den Schnitt

Lage = 'L': Pfeile liegen links der Linie,
vom 1. Punkt zum n-ten Punkt

Lage = 'R': Pfeile liegen rechts der Linie
vom 1. Punkt zum n-ten Punkt

info : '\$A...' Text am 1. Punkt
'\$A...' Text am letzten Punkt

Beispiele



CALL MTEX(QQ,MV_COL,P1,P2,'\$1ZEILE 1 \$ZEILE 2\$ZEILE 3');

ALLGEMEINER TEXT

MTEX
Bemaßung
Zusatzroutine

Syntax

```
CALL MTEX(QQ, coll, P1, P2, info);
```

Erläuterung

Diese Routine dient zum Plazieren von Text, der an bestimmte Elemente verweisen soll.

coll : Zurückgelieferte COLLECTION

P1 : Beginn der Hilfslinie, graph.Objekt
 vom Typ POINT2

P2 : Position des Textes, graph.Objekt
 vom Typ POINT2

info :

§1.... Text in Zeile 1

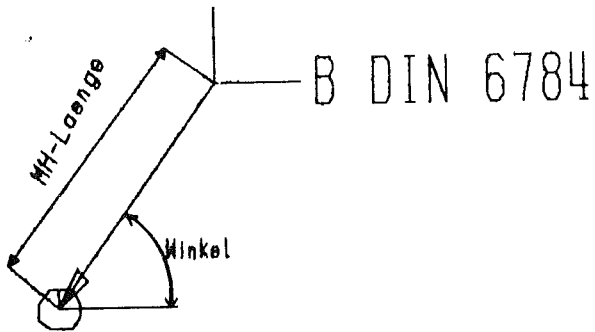
§2.... Text in Zeile 2

 ⋮

§5.... Text in Zeile 5

Oder es kann ein Maß anstelle der Formbuchstaben unter Beachtung der Zeichen \pm mit zusätzlicher Maßangabe an das Symbol gesetzt werden.

z.B.: $\$T0,5 \quad \pm 0,5$
 $\$T-0,1 \quad - 0,1$



Beispiele

\perp B DIN 6784

CALL MWSK (QQ, MCØL, P1, 'B', 45, 2, '\$TA');

\perp B DIN 6784

\perp B DIN 6784

CALL MWSK (QQ, MCØL, P1, 'H', 45, 2, '\$TB');

\perp C DIN 6784

\perp C DIN 6784

CALL MWSK (QQ, MCØL, P1, 'V', 45, 2, '\$TC');

\perp *0.5 \perp *0.5

CALL MWSK (QQ, MCØL, P1, 'B', 45, 2, '\$T0.5');

Syntax

CALL MWSK(QQ, coll, PO, lage, winkel, MH-länge, info);

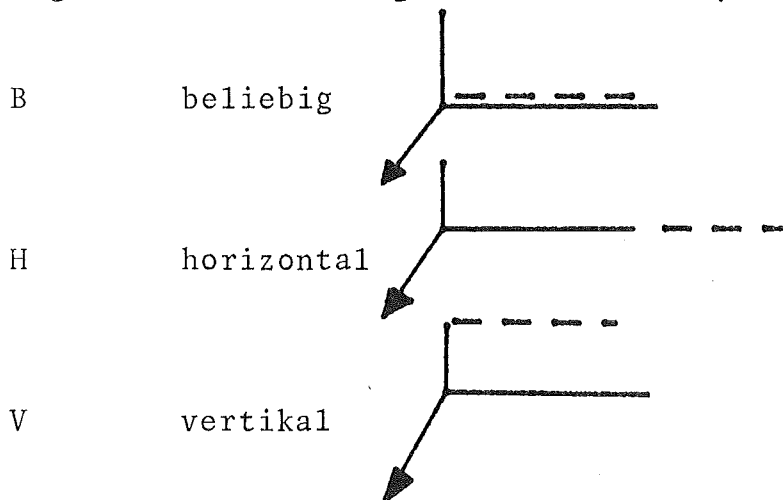
Erläuterung

Werkstückkanten nach DIN 6784

coll : zurückgelieferte COLLECTION

PØ : bezeichnet die Pfeilspitze des Zeichens,
graphisches Objekt vom Typ POINT2

lage : Lage der Beschriftung innerhalb des Systems



winkel : Winkel, unter dem der Pfeil von der Pfeilspitze
zum Symbol verläuft

MH-Länge: Länge des Pfeils

info : \$T... Beginn des Textes für das Symbol. Die Buch-
staben A,B,C,D bezeichnen den Kantenzustand
nach DIN 6784. Beginnt der Text mit den obigen
4 Buchstaben, dann wird der String 'DIN 6784'
angehängen.

z.B.: \$A1→A1 DIN 6784

Beispiel

DCL P1 POINT, P7 BALL;

DCL WAS CHAR(32) VAR INIT('');

SET P1=POINT(0,1,1);

DCL P PTR;

DCL LL LINE2;

1. WAS=OTYP(P1);/xWAS='POINT';x/

2. WAS=OTYP(P7);/xWAS='BALL_UNINIT';x/

3. WAS=OTYP(P);/xWAS='ILLEGAL';x/

4. WAS=OTYP(LL);/xWAS='LINE2';x/

(S.303 GHB)

OTYP

Funktion

Syntax

OTYP(g_obj)

Erläuterung

Mit der GIPSY-Builtin Funktion OTYP kann der Typ eines GIPSY-Objektes abgefragt werden. OTYP liefert einen Charakterstring (CHAR(32)VARYING) und zwar

1. 'ILLEGAL' - falls kein GIPSY-Objekt vorliegt -
2. 'POINT', 'TEXT', 'POLY', 'CIRCLE', 'PLANE', 'BALL',
'CYLINDER', 'CONE', 'SPACE', 'COLL', 'POINT2', 'TEXT2',
'POLY2', 'CIRCLE2', 'ARC2', 'AXIS2', 'BODY', 'ARC', 'LINE2',

je nachdem, welches dieser GIPSY-Objekte OTYP übergeben
wurde bzw.
3. 'UNDEFINED' - in allen übrigen Fällen.

Im 2.Fall wird zusätzlich '_UNIT' zurückgeliefert, falls das GIPSY-Objekt nicht initialisiert war, also z.B. 'POINT2_UNINIT'.

PFOPEN
PFCLOSE

Beispiel

```
GIPFI: PROC OPTIONS(MAIN) REGENT(INIT,NODA,MPOOL=500000);
  ENTER GIPSY; SAVE;
  MESSAGE ACTIVE I;
  ON ERROR CALL PLIDUMP('TFNB','GIPFI');
  DCL KENNUNG TEXT2(60);

  DCL PFOPEN ENTRY(PTR,PTR,FILE) EXTERNAL;
  DCL PFOPTL ENTRY(PTR,PTR,FILE,CHAR(x)) EXTERNAL;

  DCL PFCLOSE ENTRY(PTR,PTR) EXTERNAL;
  DCL PFREAD ENTRY(PTR,PTR,PTR,BIN FIXED(15),BIN FIXED(15),
    CHAR(x) VARYING,BIN FIXED(15)) EXTERNAL;

  DCL PIFIL PTR;
  DCL IFIL FILE STREAM;
  DCL KBILD COLLECTION;
  DCL (BILDNR,LESART,SEGANZ) BIN FIXED(15);
  DCL SEGNAME CHAR(20) VARYING;

CALL PFOPEN(QQ,PIFIL,IFIL);

  PUT SKIP LIST(' LIES BILD2                MODE 1 xxxxx');
  BILDNR = 2;
  LESART = 1;
  SEGANZ = 0;
  SEGNAME = '';
CALL PFREAD(QQ,PIFIL,KBILD,BILDNR,LESART,SEGNAME,SEGANZ);

  OPEN PLOT DIN A(4) BROAD;
  PLOT (KBILD);

CALL PFCLOSE(QQ,PIFIL);

DCL TFIL CHAR(8) VARYING INIT('IFIL1');
CALL PFOPTL(QQ,PIFIL,IFIL,TFIL);

  PUT SKIP LIST(' LIES BILD2                MODE 1 xxxxx');
  BILDNR = 3;
  LESART = 1;
  SEGANZ = 0;
  SEGNAME = '';
CALL PFREAD(QQ,PIFIL,KBILD,BILDNR,LESART,SEGNAME,SEGANZ);

  PLOT (KBILD);

CALL PFCLOSE(QQ,PIFIL);

MESSAGE ACTIVE DEBUG;;
END GIPSY;
END GIPFI;
```

PFOPEN
PFCLOSE
Prozeduren

Syntax

```
CALL PFOPEN(QQ,pfile, ifile);  
CALL PFOPTL(QQ,pfile, ifile, tfile);  
CALL PFCLOSE(QQ,pfile);
```

Erläuterung

PFOPEN öffnet den Plotfile ohne TITLE-Option
PFOPTL öffnet den Plotfile mit TITLE-Option
PFCLOSE schließt den durch pfile identifizierten Plotfile.
 Mehrfaches Schließen des gleichen Files führt nicht
 zu Fehlerreaktionen.
pfile ist eine Pointervariable, die den geöffneten Plotfile
 identifiziert
ifile ist der PL/1-File, der den Plotfile enthält. Er muß
 ungeöffnet an die Openroutine übergeben werden.
tfile ist ein optionaler DDNAME für den Plotfile.

Deklarationen

```
DCL PFOPEN ENTRY(OTR,PTR,FILE)EXTERNAL;  
DCL PFOPTL ENTRY(PTR,PTR,FILE,CHAR(*))EXTERNAL;  
DCL PFCLOSE ENTRY(PTR,PTR) EXTERNAL;
```

Beispiel

```
GIPFI: PROC OPTIONS(MAIN) REGENT(INIT,NODA,MPOOL=200000);
  ENTER GIPSY;   SAVE;
  DCL PFOPEN ENTRY(PTR,PTR,FILE) EXTERNAL;
  DCL PFCLOSE ENTRY(PTR,PTR) EXTERNAL;
  DCL PFREAD ENTRY(PTR,PTR,PTR,BIN FIXED(15),BIN FIXED(15),
                  CHAR(x),BIN FIXED(15)) EXTERNAL;

  DCL (PIFIL1,PPOP) PTR;
  DCL (IFIL1,IFIL2) FILE STREAM;
  DCL KBILD(60) COLLECTION;
  DCL (OUT,CPOP) COLLECTION;
  DCL (BILDNR,LESART,SEGANZ) BIN FIXED(15);
  DCL SEGNAME CHAR(20) VARYING;
  CALL PFOPEN(QQ,PIFIL1,IFIL1);

  /x ES WERDEN 60 BILDER IN KBILD GEBRACHT UND NACH UMFORMUNG x/
  /x AUF EINEM BILD GEPLOTTET. SIEHE ABB.1 x/

  DO I=1 TO 60;
    BILDNR = I;
    LESART = I;
    SEGANZ = 0;
    SEGNAME ' ';

    CALL PFREAD(QQ,PIFIL1,KBILD(I),BILDNR,LESART,SEGNAME,SEGANZ);

    SET KBILD(I)=SCALE2(KBILD(I),2,2);/x 0 x/

  END;
  CALL PFCLOSE(QQ,PIFIL1);
  OPEN PLOT SIZE(35 CM, 24 CM);
  DO I=0 TO 59;
    JY=I/10+1;      JY = 8. - JY;
    JX=MOD(I,10)+1;
    SET OUT=SHIFT2(KBILD(I+1),JX*2.5 CM,JY*2.5 CM);
    PLOT(OUT);
    EMPTY(OUT);
  END;
  END GIPSY;
  END GIPFI;
```

PFREAD
Modus 1
Prozedur

Syntax

```
CALL PFREAD(QQ,pfile,pictnr,mode,segment,sequence);
```

Erläuterung

PFREAD liest vom Plotfile

QQ ist der REGENT-Common Pointer

coll ist eine GIPSY-Kollektion. Sie liefert die GIPSY-Objekte zurück, die durch PFREAD vom Plotfile gelesen werden. Die Kollektion coll wird beim Eintritt in PFREAD durch die GIPSY-EMPTY-Funktion leergemacht.

pfile Pointervariable, die den Plotfile identifiziert.

pictnr Nummer des Bildes auf dem Plotfile

mode =1! Es wird das Bild der Nummer npict gelesen. Falls das Bild nicht vorhanden ist, wird eine leere Kollektion zurückgeliefert.

segment, sequence werden bei mode=1 ignoriert.

Deklaration

```
DCL PFREAD ENTRY(PTR,PTR,PTR, BIN FIXED(15), BIN FIXED(15),  
                CHAR(x), BIN FIXED(15)) EXTERNAL;
```

```
DCL coll COLLECTION,  
      pfile PTR  
      npict BIN FIXED(15),  
      mode BIN FIXED(15),  
      segment CHAR(n)VARYING,  
      sequence BIN FIXED(15);
```

Beispiel

```
GIPFI: PROC OPTIONS(MAIN) REGENT(INIT,NODA,MPOOL=200000);
  ENTER GIPSY;  SAVE;
  MESSAGE ACTIVEI;
  DCL KENNUNG TEXT2(60);
  ON ERROR CALL PLIDUMP('TENB','GIPFI');

  DCL PFOPEN ENTRY(PTR,PTR,FILE) EXTERNAL;
  DCL PFCLOSE ENTRY(PTR,PTR) EXTERNAL;
  DCL PFREAD ENTRY(PTR,PTR,BIN FIXED(15);BIN FIXED(15);
    CHAR(*) VARYING,BIN FIXED(15)) EXTERNAL;

  DCL PIFIL1 PTR;
  DCL IFIL1 FILE STREAM;
  DCL KBILD1 COLLECTION;
  DCL (BILDNR,LESART,SEGANZ) BIN FIXED(15);
  DCL SEGNAME CHAR(20) VARYING;

  CALL PFOPEN(QQ,PIFIL1,IFIL1);

  PUT SKIP LIST(
    ' MODE2  LIES 1.SEGMENT AUS 1.BILD AUF FILE1 xxxxxxxx');
  BILDNR = 1;
  LESART = 2;
  SEGANZ = 1;
  SEGNAME = '';
  CALL PFREAD(QQ,PIFIL1,KBILD1,BILDNR,LESART,SEGNAME,SEGANZ);

  SET KENNUNG = TEXT2(POINT2(2CM,2CM),0.,
    'BILD2 MODE2 LESE SEGMENT1AUF FILE1');
  SET KBILD1 = COLLECTION(KBILD1,KENNUNG);
  OPEN PLOT BIN A(4) BROAD;
  PLOT (KBILD1);

  CALL PFCLOSE(QQ,PIFIL1);

  MESSAGE ACTIVE DEBUG;
  END GIPSY;
END GIPSY;
```

PFREAD
Modus 2
Prozedur

Syntax

```
CALL PFREAD(QQ,coll,pfile,pictnr,mode,segment,sequence);
```

Erläuterung

QQ,coll,pfile,pictnr siehe Modus 1

mode = 2: lies eines oder mehrere der folgenden Teilbilder
segment ist ein Charakterstring, er liefert den Teilbildnamen
des zuerst gelesenen Teilbildes (Plotfilebezeichnung:
Segment)

Anmerkung: Die GIPSY-Anweisung PLOT(name); erzeugt auf dem
Plotfile ein Teilbild mit Namen = "name".

sequence gibt an, wieviele Teilbilder gelesen werden sollen.
Bei Bildende werden die letzten noch vorhandenen Teilbilder
zurückgeliefert.

Deklaration

siehe Modus 1

PFREAD
Modus 3

Beispiel

```
GIPFI: PROCOPTIONS(MAIN) REGENT(INIT,NODA,MPOOL=500000);
  ENTER GIPSY;  SAVE;
    DCLKENNUNG TEXT2((0);
    ON ERROR CALL PLIDUMP('TFNB','GIPFI');

    DCL PFOPEN ENTRY(PTR,PTR,FILE) EXTERNAL;
    DCL PFCLOSE ENTRY(PTR,PTR) EXTERNAL;
    DCL PFREAD ENTRY(PTR,PTR,PTR,BIN FIXED(15),BIN FIXED(15),
                     CHAR(x) VARYING,BIN FIXED(15)) EXTERNAL;

    DCL PIFIL1 PTR;
    DCL IFIL1 FILE STREAM;
    DCL KBILD1 COLLECTION;
    DCL (BILDNR,LESART,SEGANZ)BIN FIXED(15);
    DCL SEGNAME CHAR(20) VARYING;

  CALL PFOPEN(QQ,PIFIL1,IFIL1);

  PUT SKIP LIST(
    ' SKIP ERSTES SEGMENT,LIES NAECHSTES AUF FILE1 xxxxxx');
  BILDNR = 1;
  LESART = 3;
  SEGANZ = 1;
  SEGNAME = '';
  CALL PFREAD(QQ,PIFIL1,KBILD1,BILDNR,LESART,SEGNAME,SEGANZ);

  LESART = 2;
  CALL PFREAD(QQ,PIFIL1,KBILD1,BILDNR,LESART,SEGNAME,SEGANZ);

  SET KENNUNG = TEXT2(POINT2(2CM,2CM),0.,
    'BILD MODE 3+2 SKIP SEGMENT1,LESE SEGMENT');
  SET KBILD; = COLLECTION(KBILD1,KENNUNG);
  OPEN PLOT DIN A(4) BROAD;
  PLOT (KBILD1);

  CALL PFCLOSE(QQ,PIFIL1);

  MESSAGE ACTIVE DEBUG;
  END GIPSY;
END GIPFI;
```


PFREAD
Modus 3
Prozedur

Syntax

```
CALL PFREAD(QQ, coll, pfile, pictnr, mode, segment, sequence);
```

Erläuterung

QQ, coll, pfile, pictnr, siehe Modus 1

mode = 3: Hier werden Segmente überlesen. Es wird eine leere
Kollektion zurückgegeben.

segment liefert den Namen des ersten ignorierten Teilbildes.

sequence gibt an, wieviele Teilbilder überlesen werden.

Deklaration

siehe Modus 1

Beispiel

```
GIPFI: PROC OPTIONS(MAIN) REGENT(INIT,NODA,MPOOL=500000);
  ENTER GIPSY;   SAVE;
    DCL KENNUNG TEXT2(60);

    DCL PFOBEN ENTRY(PTR,PTR,FILE) EXTERNAL;
    DCL PFCLOSE ENTRY(PTR,PTR) EXTERNAL;
    DCL PFREAD ENTRY(PTR,PTR,PTR,BIN FIXED(15),BIN FIXED(15),
                     CHAR(x) VARYING,BIN FIXED(15)) EXTERNAL;
    DCL PIFIL1 PTR;
    DCL IFIL1 FILE STREAM;
    DCL KBILD1 COLLECTION;
    DCL (BILDNR,LESART,SEGANZ) BIN FIXED(15);
    DCL SEGNAME CHAR(20) VARYING;

  CALL PFOPEN(QQ,PIFIL1,IFIL1);

  PUT SKIP LIST(
    ' LIES NAECHSTES SEGMENT MIT NAMEN "CDRUC1" xxxxx);
  LESART = 4;
  SEGNAME = 'CDRUC1';
  BILDNR = 1;
  SEGANZ = 1;
  CALL PFREAD(QQ,PIFIL1,KBILD1,LESART,SEGNAME,SEGANZ);

  SET KENNUNG = TEXT2(POINT2(2CM,2CM),0.,
    'BILD4 MODE3 LIES SEGMENT MIT NAMEN "CDRUC1"');
  SET KBILD1 = COLLECTION(KBILD1,KENNUNG);
  OPEN PLOT DIN A(4) BROAD;
  PLOT (KBILD1);
  PUT SKIP LIST(SEGNAME);

  CALL PFCLOSE(QQ,PIFL1);

  END GIPSY;
END GIPFI;
```

PFREAD
Modus 4
Prozedur

Syntax

```
CALL PFREAD(QQ,coll,pfile,pictnr,mode,segment,sequence);
```

Erläuterung

QQ,coll,pfile,pictnr, siehe Modus 1

mode = 4: Es wird ein Teilbild mit einem bestimmten, vorgegebenen Namen gesucht, beziehungsweise gelesen.

segment übergibt als Charakterstring den Namen des Teilbildes.

sequence ist ohne Bedeutung.

Falls das Teilbild nicht gefunden wurde, wird eine leere Kollektion zurückgeliefert.

Deklaration

siehe Modus 1

Beispiel

```
GIPFI: PROC OPTIONS(MAIN) REGENT(INIT,NODA,MPOOL=500000);
  ENTER GIPSY;   SAVE;
    DCL KENNUNG TEXT"(60);
    ON ERROR CALL PLIDUMP('TFNB','GIPFI');

    DCL PFOPEN ENTRY(PTR,PTR,FILE) EXTERNAL;
    DCL PFCLOSE ENTRY(PTR,PTR) EXTERNAL;
    DCL PFREAD ENTRY(PTR,PTR,BIN FIXED(15),BIN FIXED(15),
                     CHAR(x) VARYING,BIN FIXED(15)) EXTERNAL;
    DCL PIFIL1 PTR;
    DCL IFIL1 FILE STREAM;
    DCL KBILD1 COLLECTION;
    DCL (BILDNR,LESART,SEGANZ) BIN FIXED(15);
    DCL SEGNAME CHAR(20) VARYING;

  CALL PFOPEN(QQ,PIFIL1,IFIL1);

  PUT SKIP LIST(
    ' LIES ALLE SEGMENTE MIT NAMEN "UEBE1" MODE5   xxxxx');
  LESART = 5;
  SEGNAME = 'UEBE1';
  BILDNR = 1;
  SEGANZ = 0;
  PUT SKIP LIST('SEGNAME VOR DEM LESEN:',SEGNAME);
  CALL PFREAD(QQ,PIFIL1,KBILD1,BILDNR,LESART,SEGNAME,SEGANZ);

  PUT SKIP LIST('SEGNAME NACH DEM LESEN:',SEGNAME);
  SET KENNUNG = TEXT2(POINT2(2CM,2CM),0.);
  'BILD7 MODE5 LIES ALLE SEGMENTE MIT NAMEN UEBE1');
  SET KBIL1 = COLLECTION(KBILD1,KENNUNG);
  OPEN PLOT DIN A(4) BROAD;
  PLOT (KBILD1);

  CALL PFCLOSE(QQ,PIFIL1);

END GIPSY;
```

PFREAD
Modus 5
Prozedur

Syntax

```
CALL PFREAD(QQ,coll,pfile,pictnr,mode,segment,sequence);
```

Erläuterung

QQ,coll,pfile,pictnr siehe Modus 1

mode = 5: Alle Teilbilder eines Bildes, die einen vorgegebenen Namen besitzen, werden gelesen.

segment übergibt den Namen der gesuchten Teilbilder.

sequence ist ohne Bedeutung.

Falls kein Teilbild mit dem gegebenen Namen existiert, wird eine leere Kollektion zurückgeliefert.

Deklaration

siehe Modus 1

PFREAD

Modus 6

Beispiel

```
GIPFI: PROC OPTIONS(MAIN) REGENT(INIT,NODA,MPOOL=500000);
  ENTER GIPSY;   SAVE;
  MESSAGE ACTIVE I;
  DCL KENNUNG TEXT2(60);
  ON ERROR CALL PLIDUMP('TFNB','GIPFI');

  DCL PFOPEN ENTRY(PTR,FILE) EXTERNAL;
  DCL PFCLOSE ENTRY(PTR,PTR) EXTERNAL;
  DCL PFREAD ENTRY(PTR,PTR,BIN FIXED(15),BIN FIXED(15);
                  CHAR(*) VARYING,BIN FIXED(15)) EXTERNAL;
  DCL PIFIL1 PTR;
  DCL IFIL1 FILE STREAM;
  DCL KBILD1 COLLECTION;
  DCL (BILDNR,LESART,SEGANZ)BIN FIXED(15);
  DCL SEGNAME CHAR(20) VARYING;

  CALL PFOPEN(QQ,PIFIL1,IFIL1);

  PUT SKIP LIST(
    ' LIES ALLE SEGMENTE BIS ZUM BILDENDE   MODE6   xxxxx');
  LESART = 6;
  SEGNAME = 'XXXXXX';
  BILDNR = 1;
  SEGANZ = 0;
  PUT SKIP LIST('SEGNAME VOR DEM LESEN:',SEGNAME);
  SET KENNUNG = TEXT2(POINT2(2CM,2CM),0.,
'BILD6 MODE6 LIES ALLE SEGMENTE BIS BILDENDE');
  SET KBILD1 = COLLECTION(KBILD1,KENNUNG);
  OPEN PLOT DIN A(4) BROAD;
  PLOT (KBILD;);

  CALL PFCLOSE(QQ,PIFIL1);

  END GIPSY;
END GIPFI;
```

PFREAD
Modus 6
Prozedur

Syntax

```
CALL PFREAD(QQ,coll,pfile,pictnr,mode,segment,sequence);
```

Erläuterung

QQ,coll,pfile,pictnr siehe Modus 1

mode = 6: Es werden fortlaufend alle Teilbilder bis zum
Bildende gelesen.

segment liefert den Namen des zuerst gelesenen Teilbildes.

sequence ist ohne Bedeutung.

Falls die Bildnummer nicht existiert, wird eine leere Kollektion
zurückgeliefert.

Deklaration

siehe Modus 1

Beispiel

```
DCL C CIRCLE,  
  A ARC2,  
  P1 POLY2(10),  
  P2 POLY2(10),  
  PP POLY2(1000),  
  K COLL, LL LINE2,  
  (PK1, PK2, PK3, PK4) POINT2;
```

```
SET C=.....
```

Wertezuweisung an A, P1, P2, PK1, PK2, PK3, PK4, LL

```
SET PP=POLY2(PK1, P2, C, A, PK3, P1, LL);  
EDIT CLOSED OF(PP);  
PLOT(PP);  
SET K=COLL(POINT2(10,12),A, PK1, A, PK2);  
SET PP=POLY2(K);  
PLOT(PP);  
SET PP=POLY2(K, P2,PP,P1, A, C, K);  
PLOT(PP);  
SET PP=POLY2(COLL(P1,P2),POINT2(x,y),INT(P1,P2),ARC(PK1,PK2,PK3));  
PLOT(PP);
```


(GHB S.315)

POLY2

Funktion

Deklaration

Syntax

POLY2(g_ex [,g_ex] *)

Erläuterung

Erzeugt ein 2D-Polygonzug aus 2D-Punkten, Polygonzügen, Kreisen und Kreisbögen. Objekte, die nicht erlaubt sind (z.B.Text, Ebene, 3D-Objekte) werden mit Fehlermeldung übergangen.

Es werden auch Kreise und Kreisbögen akzeptiert und daraus Polygonzüge erzeugt, die den Parametern entsprechen, welche beim Zeichnen benutzt werden. (Kreise werden beim Zeichnen durch Polygonzüge dargestellt). Auch Kollektionen von Objekten der Typen POINT2, POLY2, ARC2, CIRCLE2 und LINE2 sind erlaubt.

Deklaration

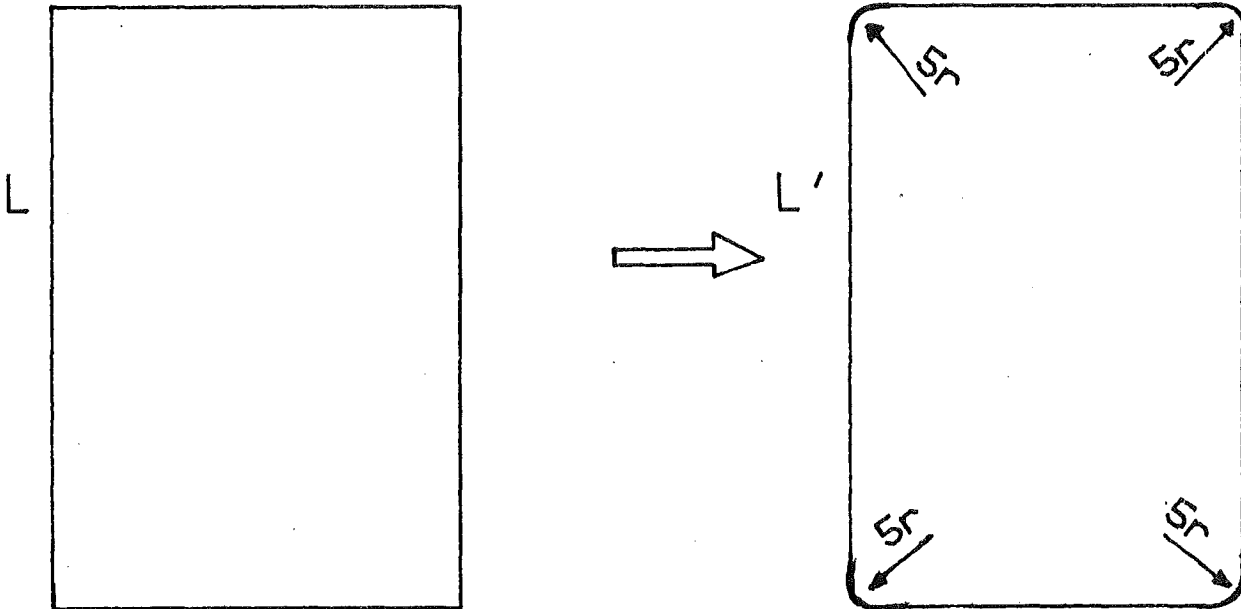
DCL [level] ident [dim] POLY2 (length) [storage class] ;

length: Maximale Anzahl von Polygon-Stützpunkten.

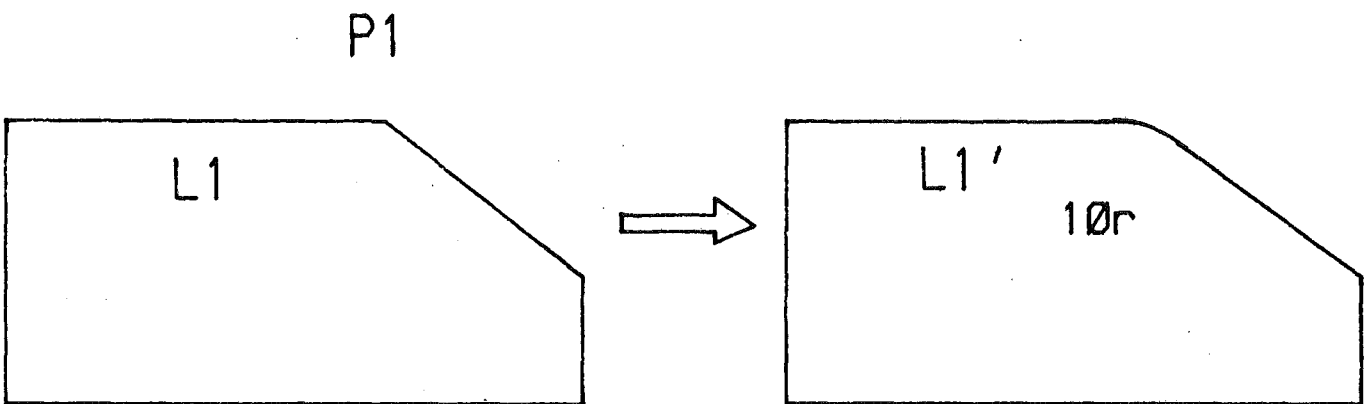
Siehe DCL

Beispiele

ROUND L RADIUS(5);



ROUND L1 RAD(10) NEAR(P1);



Literatur

- / 1 / G.Enderle, K.H.Bechler, F.Katz, K.Leinemann, W.Olbrich,
E.G.Schlechtendahl, K.Stölting:
GIPSY-Handbuch, KfK 2878 und KfK-CAD 146, 1980
- / 2 / R. Schuster:
System und Sprache zur Behandlung graphischer Information
im rechnergestützten Entwurf (GIPSY), KfK 2305, 1976
- / 3 / G.Enderle, I.Giese, M.Krause, H.P.Meinzer:
Der AGF-Plotfile - Eine Datei zum Speichern und
Transportieren graphischer Information.
KfK 2776, 1979
- / 4 / Norman M.Walcott: A Contribution to Computer Typesetting
Techniques (National Bureau of Standards, Washington,
D.C.20234, Special Publication 424, April '76)