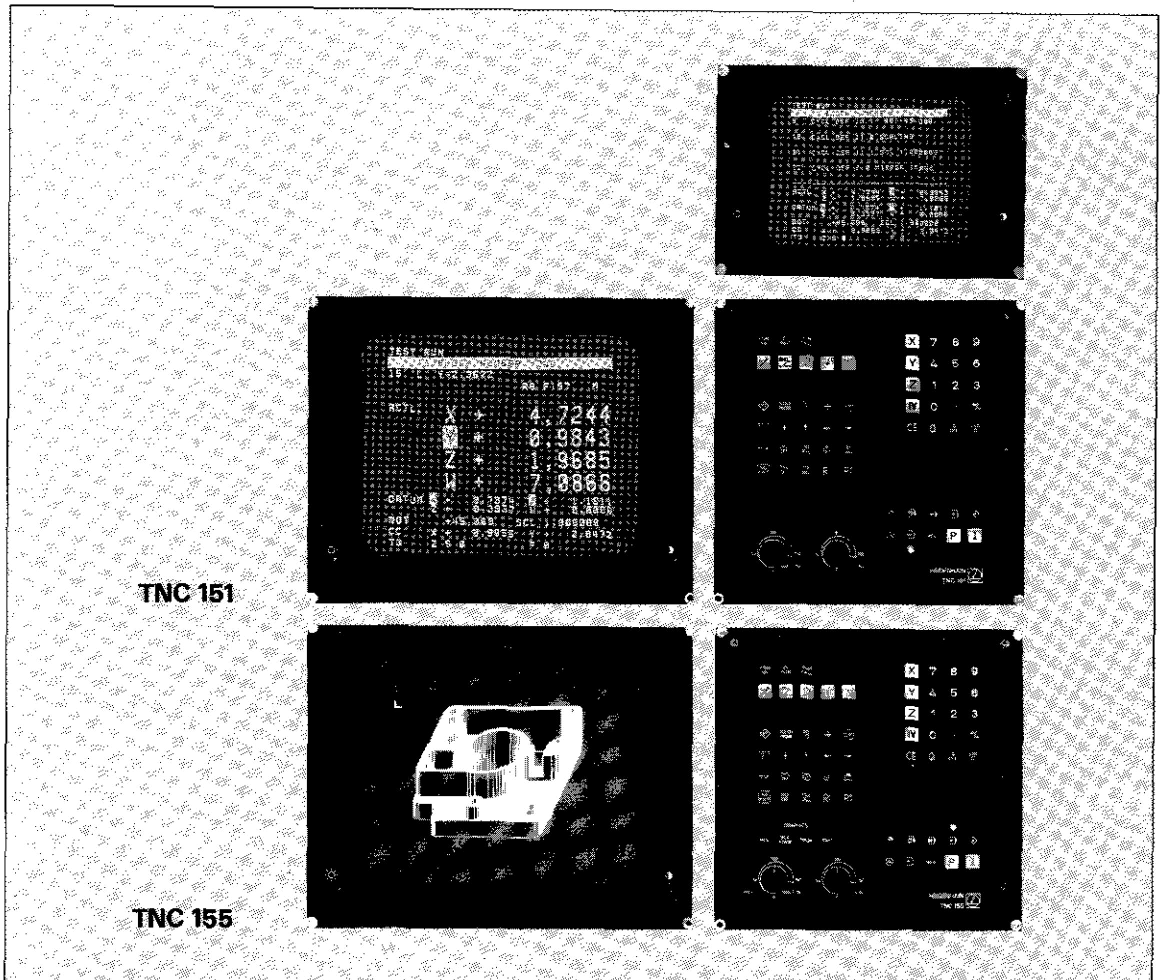
HEIDENHAIN
Optics and Electronics
Precision Graduations

Operating Manual

HEIDENHAIN TNC 151 A/TNC 151 P
HEIDENHAIN TNC 155 A/TNC 155 P

Contouring Control



July '86

This operating manual is valid for all available TNC 151/TNC 155-versions:

Transducer inputs	TNC 151/TNC 155-versions **without** separate **PLC-board(s)**	TNC 151/TNC 155-versions **with PLC-board(s)**
Sinusoidal signals	TNC 151 A/TNC 155 A TNC 151 E/TNC 155 E*	TNC 151 P/TNC 155 P TNC 151 V/TNC 155 V*
Squarewave signals	TNC 151 AR/TNC 155 AR TNC 151 ER/TNC 155 ER*	TNC 151 PR/TNC 155 PR TNC 151 VR/TNC 155 VR*

*without 3D-positioning and "transfer blockwise"

HEIDENHAIN is constantly working on further developments of its TNC-controls. It is therefore possible that details of certain control versions may deviate from the version explained in this operating manual.

Manufacturer's certificate

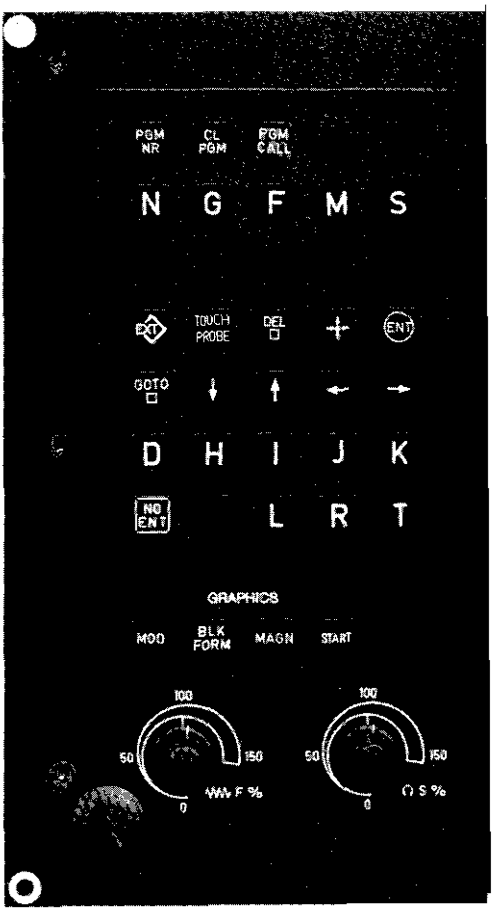
We hereby certify that the above unit is radioshielded in accordance with the West German official register decree 1046/1984.

The West German postal authorities have been notified of the issuance of this unit and have been granted admission for examination of the series regarding compliance with the regulations.

Information:

If the unit is incorporated by the user into an installation then the complete installation must comply with the above requirements.

Snap-on keyboard



Standard ISO-Keys

N Block number
G Preparatory function
F Feed rate/Dwell time with G04/ Scaling factor
M Auxiliary (Miscellaneous) function
S Spindle speed

D Parameter definition
H Angle for polar co-ordinates/ Rotational angle with G73-cycle
I X-Co-ordinate of circle centre
J Y-Co-ordinate of circle centre
K Z-Co-ordinate of circle centre

L Set label number with G98/ Jump to label number/ Tool length with G99
R Radius for polar co-ordinates Rounding-off radius with G25, G26, G27/Chamfer with G24/ Tool radius with G99
T Tool definition with G99/ Tool call

Keyboard

Program management

PGM NR Designation and recall of programs

CL PGM Clear program

CALL PGM Recall of a program within another program

Entry of workpiece contour

L Line (Linear interpolation)/Chamfers

RND Rounding of corners/Tangential contour approach and departure

CT Circle tangentially adjoining the previous contour (End position only)

CC Circle centre/pole

C Circle definition (with circle centre and arc end position)

Programming and editing

EXT External data transmission

TOUCH PROBE Touch probe functions

DEL □ Delete block

✛ Actual position data programming

ENT Enter into memory

GOTO □ ↓ ↑ ← → Search and editing routines

STOP Programmed STOP; Interruption/Discontinuation

CYCL DEF CYCL CALL Definition and recall of canned cycles

LBL SET LBL CALL Definition and recall of subprograms

NO ENT "No entry" into memory/Dialogue question "Skip-over"

TOOL DEF TOOL CALL Definition and recall of tools

R+ R- Tool radius/Tool path compensation

Graphics (TNC 155 only)

MOD Graphics modes

BLK FORM Definition of workpiece blank form and reset to blank form

MAGN Magnify

START Graphics start

Entry values and axis address

X Y Z IV Axis address

CE Clear entry

END □ End block entry

Parameter programming

Q Entry of parameter to substitute a numerical value

D DEF Definition of parameter functions

Operating modes

■ Manual operation (The control operates as a conventional digital readout)

▣ Positioning with MDI (Manual Data Input) (Block is keyed-in without entry into memory and immediately positioned)

▣ Program run in single block operation (Block-by-block positioning)

■ Automatic (complete run of program sequence)

■ Programming (Manual program entry or via the data interface)

■ Electronic handwheel

■ Program test (for checking stored program without machine movement)

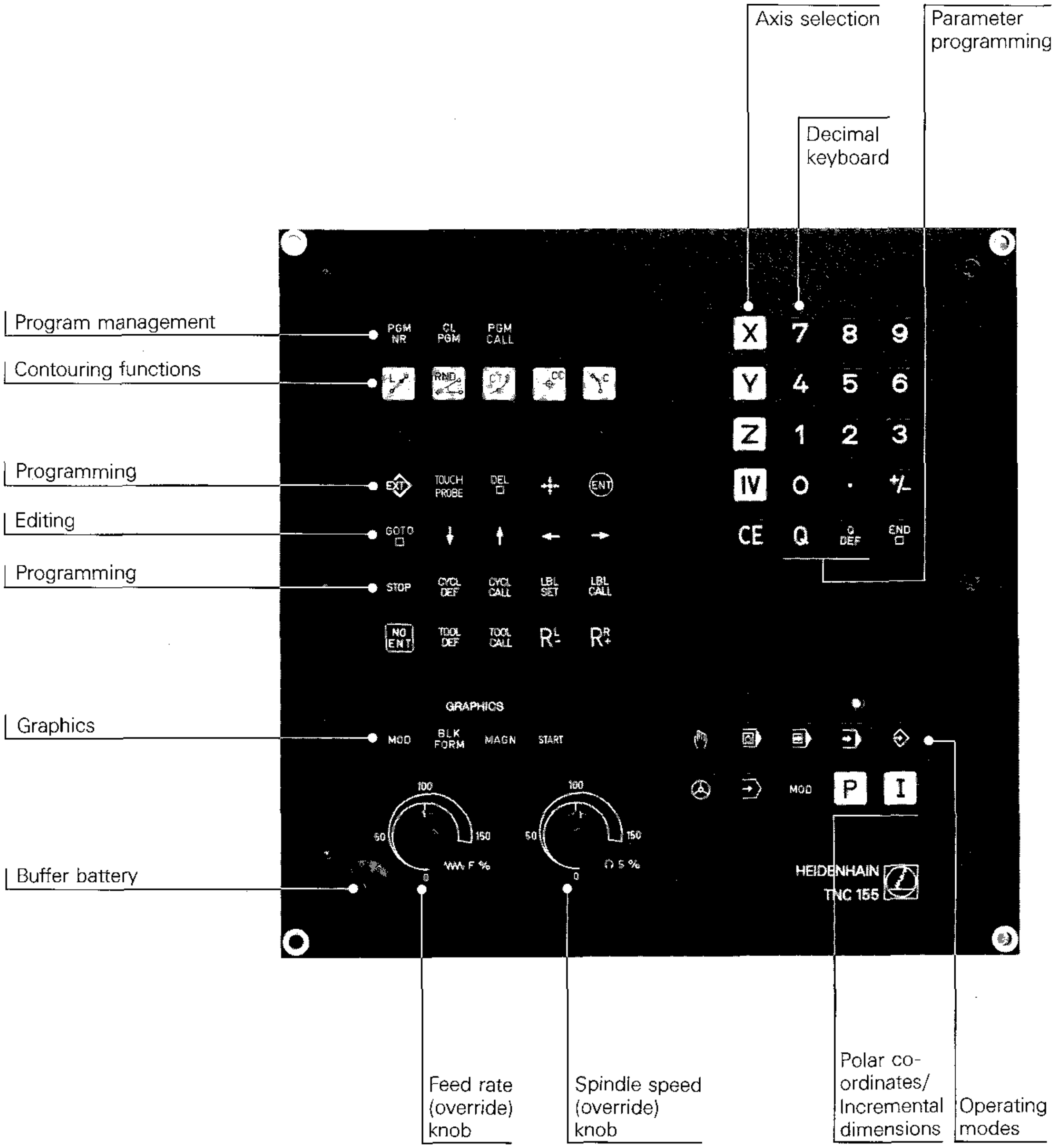
MOD Supplementary operating modes (Vacant blocks – mm/inch – Character height of position display)
Display switchover: Actual/Nominal value/Distance to go/
Trailing error. Baud rate – Safety zones – User parameters –
Code number – NC/PLC-software number
With ISO-programming: Block number increment

Polar co-ordinates/Incremental dimensions

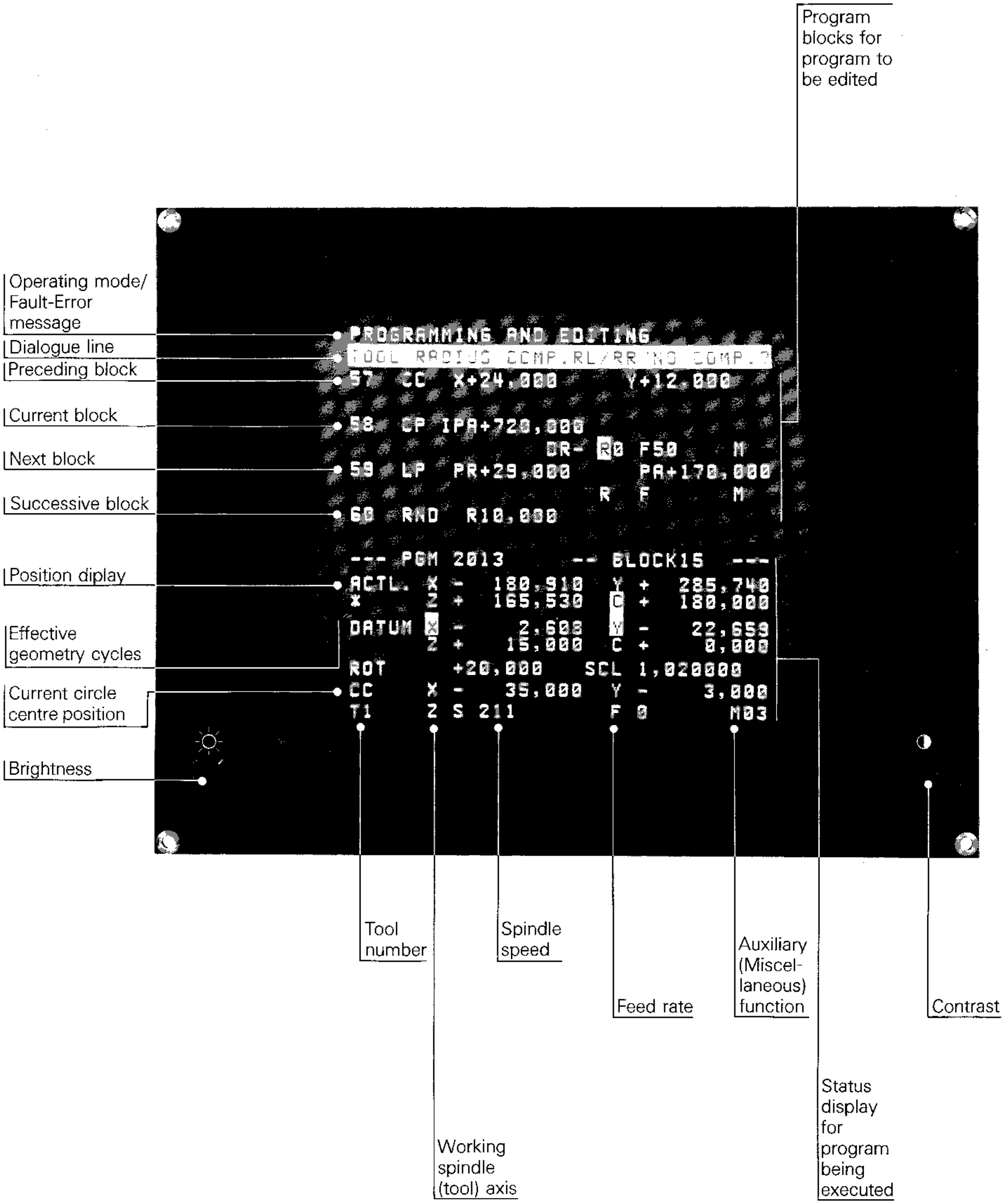
P Nominal position entry in polar co-ordinates

I Nominal position entry in incremental dimensions

Operating panel



Screen display data



```
PROGRAMMING AND EDITING
TOOL RADIUS COMP.RL/RR/NO COMP.?
57  CC  X+24,000      Y+12,000

58  CP IPA+720,000
              DR- R0 F50    M
59  LP  PR+29,000     PA+170,000
                  R  F      M
60  RND  R10,000

--- PGM 2013     -- BLOCK15  ---
ACTL. X -  180,910  Y +  285,740
X     Z +  165,530  C +  180,000
DATUM X -    2,608  Y -   22,659
      Z +   15,000  C +    0,000
ROT     +20,000  SCL 1,020000
CC    X -   35,000  Y -    3,000
T1    Z S 211       F 0      M03
```

Program blocks for program to be edited
Operating mode/ Fault-Error message
Dialogue line
Preceding block
Current block
Next block
Successive block
Position diplay
Effective geometry cycles
Current circle centre position
Brightness
Tool number
Spindle speed
Feed rate
Auxiliary (Miscel-laneous) function
Contrast
Status display for program being executed
Working spindle (tool) axis

List of contents

Introduction — E

Manual operation — M

Co-ordinate system and dimensions — K

Programming with HEIDENHAIN plain language dialogue — P

Program entry in ISO-format (G-codes) — D

Touch probe system — A

External data transmission via V.24/RS-232-C-interface — V

Technical description and specifications, Index — T

Brief description
TNC 151/TNC 155 Control

Control type

The HEIDENHAIN TNC 151/TNC 155 is a contouring control for 4 axes. Axes X, Y and Z are linear axes and axis IV can be used optionally for the connection of a rotary table or a further linear axis. The fourth axis can be switched on or off as is required.

This 4-axis control permits:
- linear interpolation in any 3 axes
- circular interpolation in two linear axes

With the aid of parameter programming, complex contours can be machined.

Program entry

Program entry can be either in
- HEIDENHAIN plain language dialogue
  or
- in standard format to ISO 6983 (G-codes).

Dialogues, entry values, the machining program, fault/error messages and position data are displayed on the VDU-screen. The program memory has a capacity for 32 programs with a total of 3100 blocks.

Entry of the machining program is either by manual key-in or "electronically" via a data interface.
The "transfer blockwise" mode permits transfer and execution of machining programs from an external data store.
During execution of a machining program, a further program may be manually entered via the background programming feature.

Magnetic tape cassette units

The HEIDENHAIN magnetic tape units ME 101/ ME 102 are available for external storage of a program on magnetic tape cassettes. These units each have two interfaces for connection of a peripheral unit (e.g. a printer) in addition to the TNC 151/TNC 155.

Brief description
TNC 151/TNC 155 Control

Program test

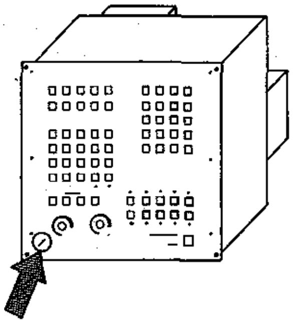
In the operating mode "program test", the TNC 151/TNC 155 checks a machining program without machine movement. Program errors are clearly displayed in plain language. A further possibility for program checking is provided by the graphics feature in which program run is simulated. Machining in the three main axes can be simulated with a constant tool axis and a cylindrical milling hob.

Programs which were compiled on the control models TNC 145 and TNC 150 are fully compatible with the TNC 151/TNC 155. Entry data is adapted to the TNC 151/TNC 155 by the control. An existing TNC 145 program library is also accepted by the TNC 151/TNC 155.

Exchange of buffer batteries

The buffer battery is the power source for the machine parameter store and the program memory of the control. It is located beneath the cover on the control panel.
If the error message
= EXCHANGE BUFFER BATTERY =
is displayed, the batteries must be exchanged.
(Upon display of the message, the memory content is retained for approx. 1 week)

Battery exchange must be made with the mains power on. [illegible]

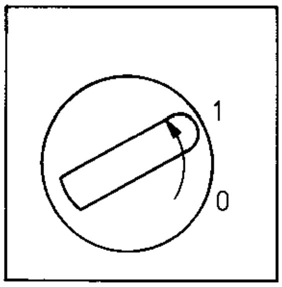


Battery type
Mignon cells, leak proof
IEC-description "LRG"
Recommended: VARTA Type 4006

Control switch-on
Traversing over reference points

Switch-on

The following symbols are used in this operating manual:
○ ≙ Buttons on the external machine operating panel.
□ ≙ Keys on the operating panel of the TNC- unit.

[redacted] ▶ 

Switch on power.

MEMORY TEST

The control checks the internal control electronics. The display is erased automatically.

POWER INTERRUPTED ▶ CE Cancel dialogue message.

RELAY EXT. DC VOLTAGE MISSING ▶ I Switch on control voltage.

PASS OVER Z-REFERENCE MARK
PASS OVER X-REFERENCE MARK
PASS OVER Y-REFERENCE MARK
PASS OVER REFERENCE MARK AXIS 4

▶ START Traverse over the reference point of each axis.
Start each axis individually.

MANUAL OPERATION

If, for exceptional reasons, the reference marks cannot be traversed over in the above sequence (e.g. due to danger of collision), proceed as follows:

Control switch-on
Traversing over reference points

```
PASS OVER Z-REFERENCE MARK
PASS OVER X-REFERENCE MARK
PASS OVER Y-REFERENCE MARK
PASS OVER REFERENCE MARK AXIS 4
```

MOD Select supplementary mode.

VACANT BLOCKS = 1654

↑ Select MOD-function "code number".

CODE NUMBER =

□ Key-in code number **84159**.

ENT Enter into memory.

```
CAUTION: SOFTWARE LIMITS INACTIVE
CODE NUMBER = 84159
PASS OVER Z-REFERENCE MARK
PASS OVER X-REFERENCE MARK
PASS OVER Y-REFERENCE MARK
PASS OVER REFERENCE MARK AXIS 4
```

X Traverse over reference point of X-axis.

Y Traverse over reference point of Y-axis.

Z Traverse over reference point of Z-axis.

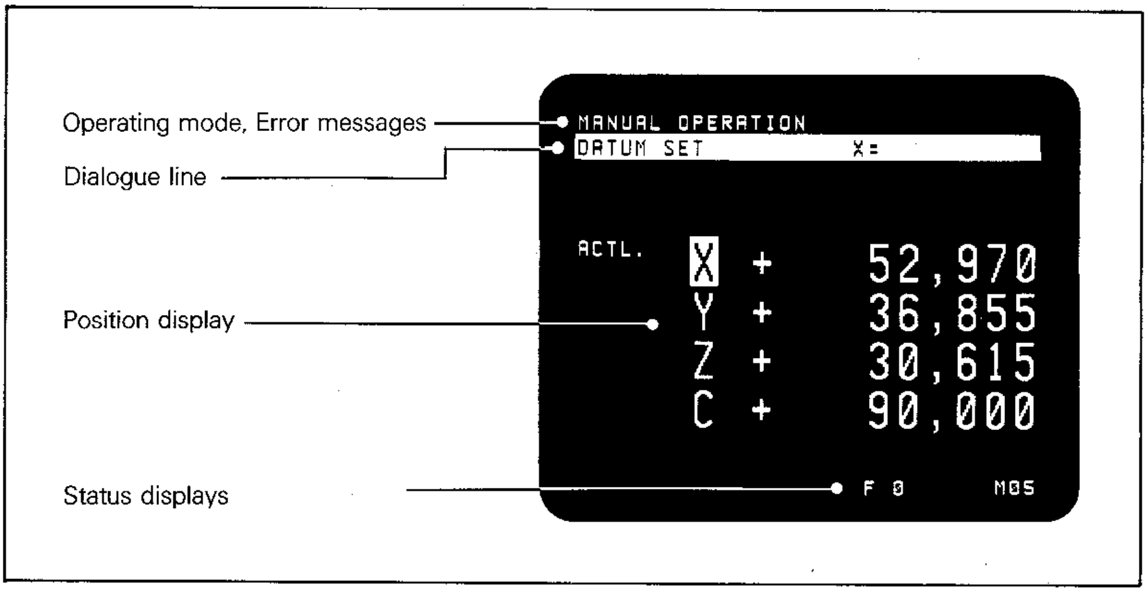
IV Traverse over reference point of IV-axis.

The reference points can be traversed over in any desired sequence, either via the axis direction buttons or via the external start button.

MANUAL OPERATION

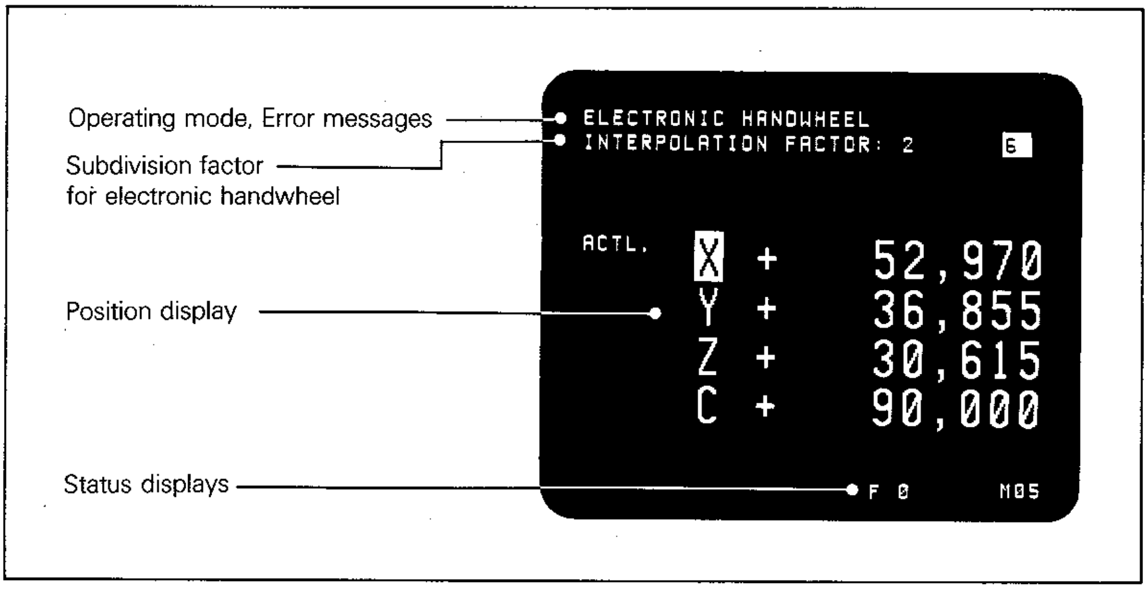
Operating modes and screen displays

Manual operation



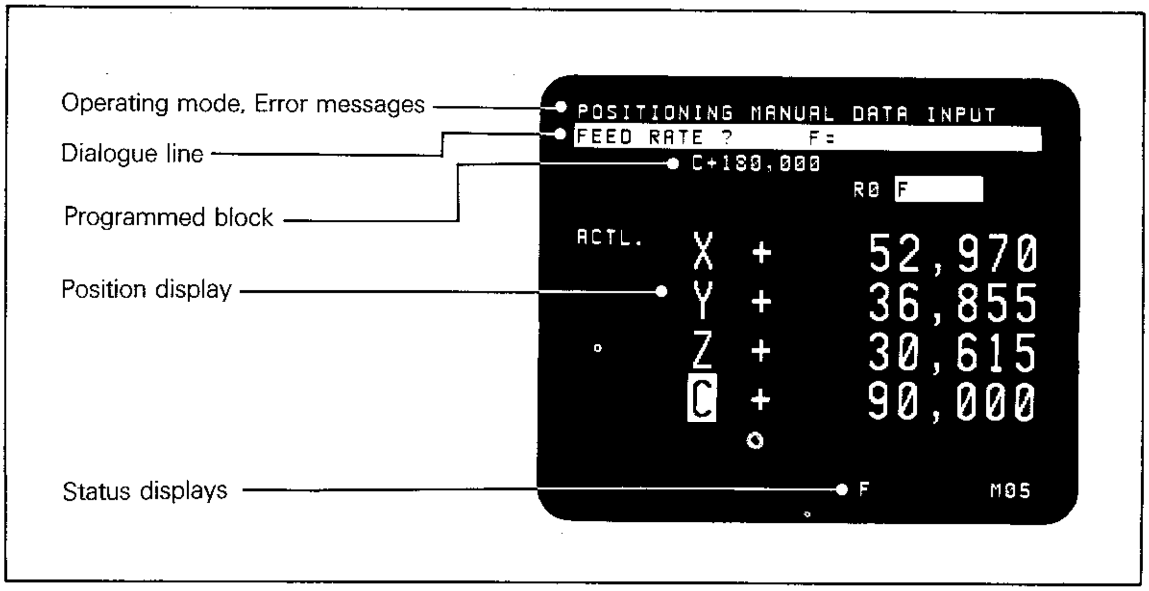
```
MANUAL OPERATION
DATUM SET                X=

ACTL.   X  +   52,970
        Y  +   36,855
        Z  +   30,615
        C  +   90,000

                    F 0        M05
```

Electronic handwheel



```
ELECTRONIC HANDWHEEL
INTERPOLATION FACTOR: 2          6

ACTL.   X  +   52,970
        Y  +   36,855
        Z  +   30,615
        C  +   90,000

                    F 0        M05
```

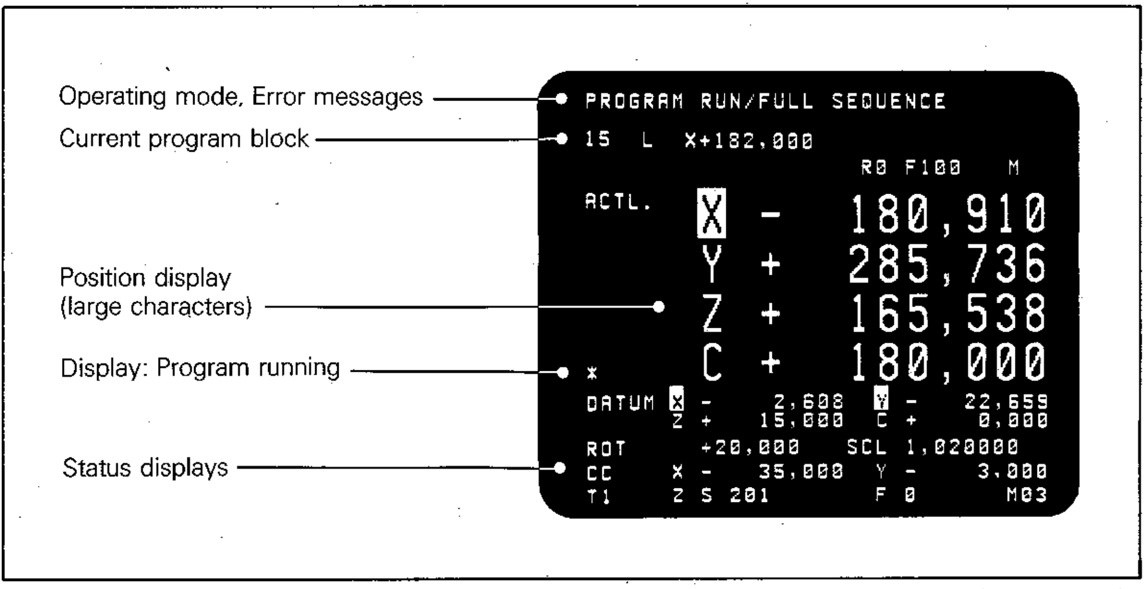
Positioning with MDI



```
POSITIONING MANUAL DATA INPUT
FEED RATE ?        F=
         C+180,000
                        R0 F
ACTL.   X  +   52,970
        Y  +   36,855
        Z  +   30,615
        C  +   90,000

                    F          M05
```

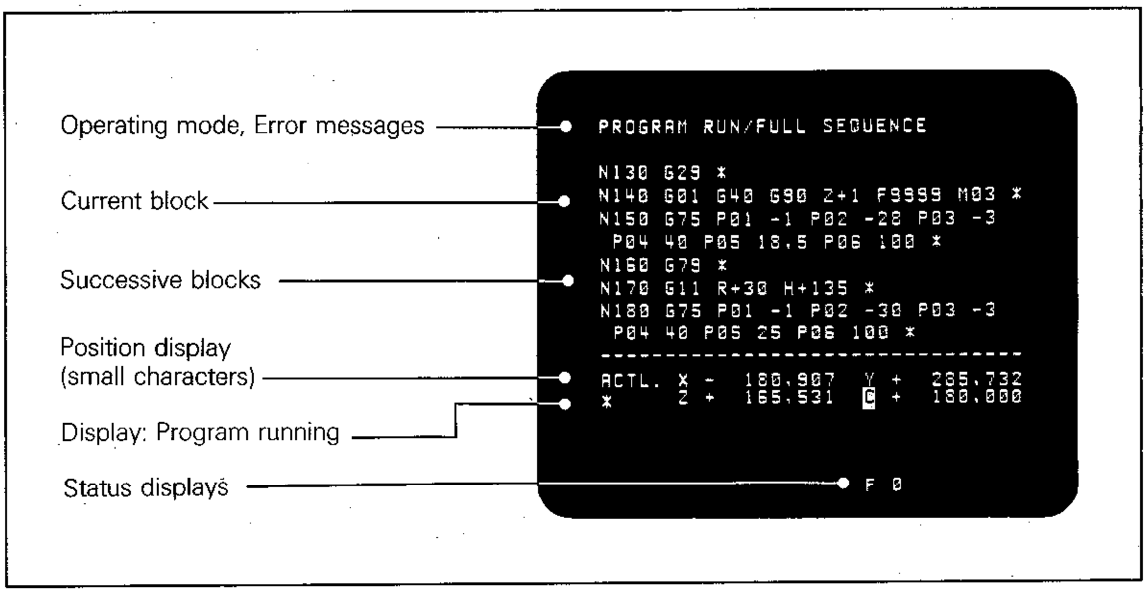
Operating modes and screen displays

Program run, single block (HEIDENHAIN-dialogue)


Operating mode, Error messages
Current program block
Position display (large characters)
Display: Program running
Status displays

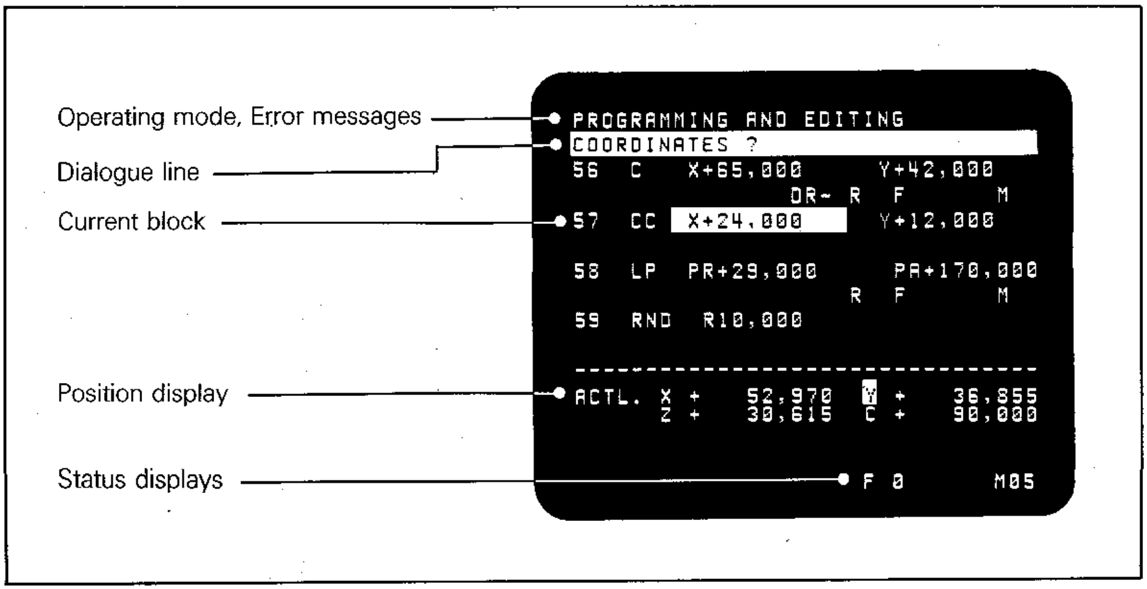
```
PROGRAM RUN/FULL SEQUENCE
15  L  X+182,000
                          R0 F100   M
ACTL.   X  -   180,910
        Y  +   285,736
        Z  +   165,538
*       C  +   180,000
DATUM X -    2,608  Y -    22,659
      Z +   15,000  C +     0,000
ROT     +20,000    SCL 1,020000
CC    X -   35,000  Y -     3,000
T1    Z S 201       F 0        M03
```

Program run, single block (ISO-Format)



```
PROGRAM RUN/FULL SEQUENCE

N130 G29 *
N140 G01 G40 G90 Z+1 F9999 M03 *
N150 G75 P01 -1 P02 -28 P03 -3
 P04 40 P05 18,5 P06 100 *
N160 G79 *
N170 G11 R+30 H+135 *
N180 G75 P01 -1 P02 -30 P03 -3
 P04 40 P05 25 P06 100 *
--------------------------------
ACTL. X -  180,907  Y +  285,732
*     Z +  165,531  C +  180,000


                    F 0
```

Programming



```
PROGRAMMING AND EDITING
COORDINATES ?
56  C   X+65,000      Y+42,000
                DR- R  F      M
57  CC  X+24,000      Y+12,000

58  LP  PR+29,000      PA+170,000
                    R  F      M
59  RND  R10,000

--------------------------------
ACTL. X +   52,970  Y +   36,855
      Z +   30,615  C +   90,000


                    F 0      M05
```

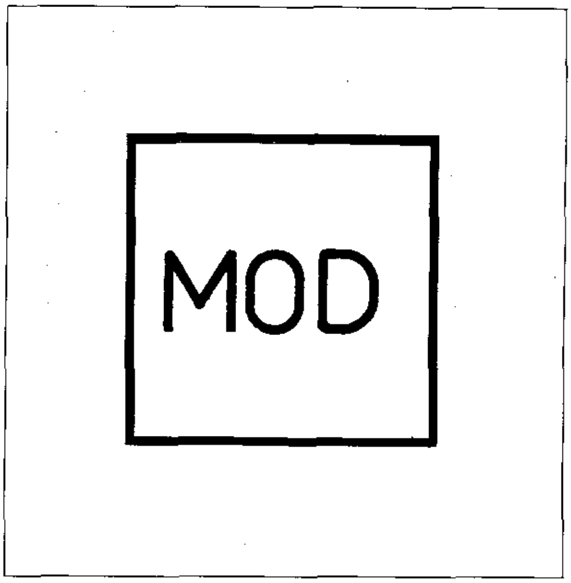
Supplementary operating modes

Introduction

In addition to the main operating modes, the TNC 151/TNC 155 also provides **supplementary operating modes** i. e. MOD*-functions.
Supplementary operating modes are addressed with the MOD -key. After pressing this key, the dialogue line displays the MOD-function "Vacant blocks".
The MOD-menu can be paged both forward and reverse via the ↓ ↑ -keys. Forward paging is also possible with the MOD -key.

Supplementary modes are cancelled with the DEL □ -key.

* MOD = abbreviation for "mode"



Limitations

With program run in the ≡▶ or →▶ -mode, the following supplementary modes can be addressed:
- ● Position display enlarged/small
- ● Vacant blocks

During display of
= POWER INTERRUPTED =
the following supplementary modes can be addressed:
- ● Code number
- ● User parameters
- ● NC-software number
- ● PLC-software number

Vacant blocks

The supplementary mode "Vacant blocks" indicates the number of vacant blocks which are still available.
When programming in ISO-format (G-codes), the number of vacant characters is displayed.

Display example:

VACANT BLOCKS = 1178

Supplementary operating modes
Addressing and cancellation of MOD-functions

Addressing

Operating mode ____________________

Dialogue initiation ____________________ MOD

VACANT BLOCKS = 1974

Select MOD-function via paging keys ▶ ↓ ↑

or MOD-key
(only forward paging possible). ▶ MOD

Cancellation

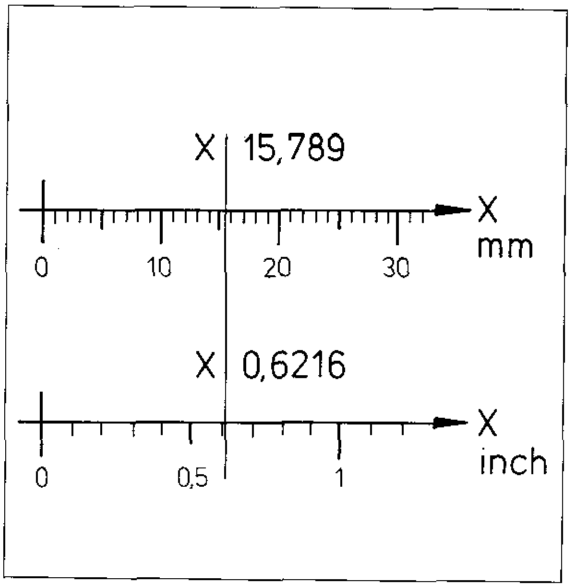
LIMIT X+ = X+ 350,000 ▶ DEL □ Leave supplementary mode

Numerical entries are to be transferred into the memory via ENT before cancellation of the MOD-function.

Supplementary operating modes

mm/inch changeover

The MOD-function mm/inch enables the operator to choose between metric and imperial display. Press ENT for changeover from mm – to – inch or vice-versa.

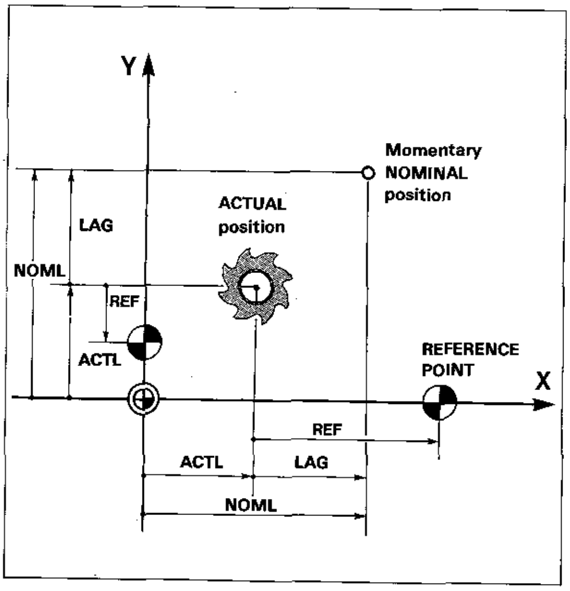


The mm or inch mode can be easily recognised by observing the number of decimal places:
X 15.789 mm-display
X 0.6216 inch-display

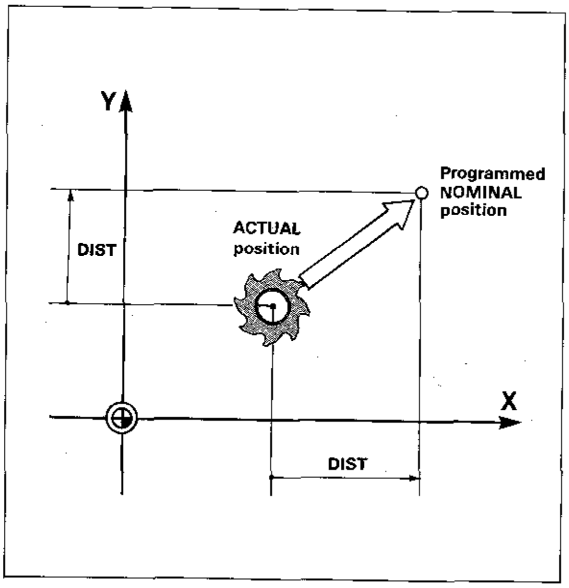
Position data display

The MOD-function "position data display" enables selection of various position data:

- ● Display of the actual position: **ACTL**
- ● Display of the distance to reference points: **REF**
- ● Display of displacement between the momentary nominal position and the actual position (trailing error or lag): **LAG**
- ● Display of the momentary nominal position as calculated by the control: **NOML**



- ● Display of the "distance to go" to the nominal position (difference between programmed nominal position and momentary actual position): **DIST**



Supplementary operating modes

Position data display

Select MOD-function.	▶ MOD ↓	

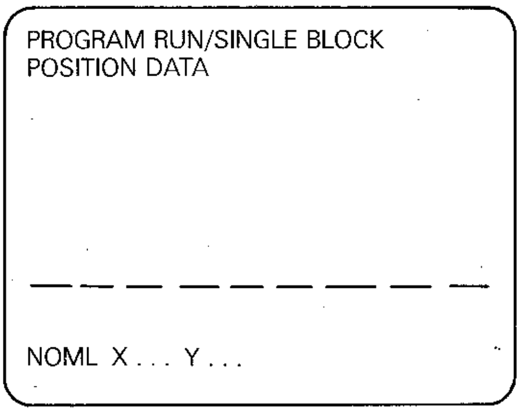
CHANGE MM/INCH

The control displays position data in mm and is to be changed to inch-mode.	▶ ENT	Switchover.

The changeover from inch-mode to mm-mode is performed in the same manner.

mm/inch changeover

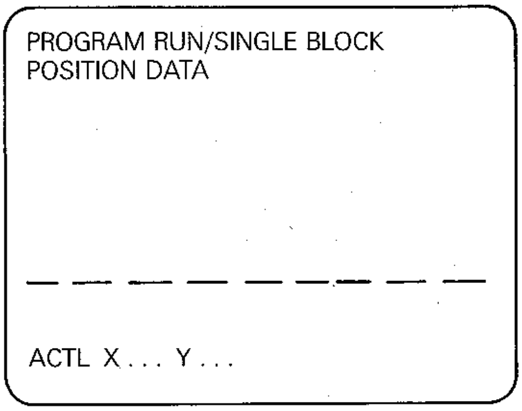
Select MOD-function.	▶ MOD ↓	



PROGRAM RUN/SINGLE BLOCK
POSITION DATA

NOML X . . . Y . . .

The display is to be switched over to "actual position":	▶ ENT	Switchover. (Repetitive pressing until ACTL is displayed)



PROGRAM RUN/SINGLE BLOCK
POSITION DATA

ACTL X . . . Y . . .

The display is to be switched over to "nominal position" again:	▶ ENT	Switchover. (Repetitive pressing until NOML is displayed)

Switchover to the modes REF, LAG and DIST is performed in the same manner.

Supplementary operating modes

Position display enlarged/small

The character height on the screen display can be converted in the operating modes: ⇛ program run single block and ⇨ automatic program run.
With display in small characters, four program blocks are also shown (previous, current, next and a successive block). With large characters, only the current block is displayed.

With ISO-programming, the position display cannot be switched over to enlarged characters. This is due to some program blocks requiring more than two lines.

Block number increment

When programming in ISO-format (G-codes), the increment from block number-to-block number can be determined via the MOD-function "Block number increment".

If the block number increment is e.g. 10, the blocks are numbered as follows:
N10
N20
N30
etc.

Entry range: 0 – 99

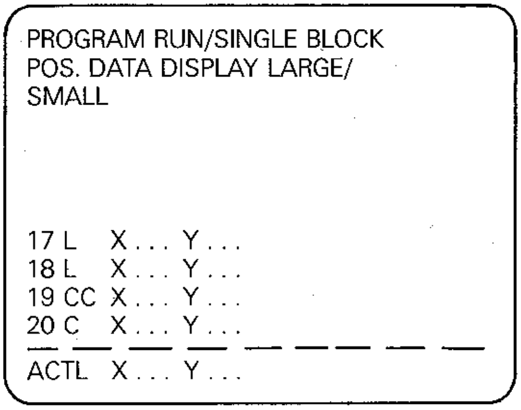
Baud rate

The MOD-function "Baud rate" indicates the data transmission rate for the data-interface (see page "Baud rate entry").

Supplementary operating modes

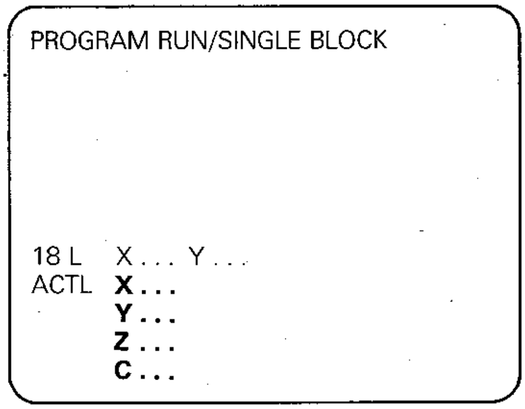
Position display enlarged/small

Select MOD-function "Position data display large/small": ▶ MOD ↓



```
PROGRAM RUN/SINGLE BLOCK
POS. DATA DISPLAY LARGE/
SMALL




17 L   X . . .  Y . . .
18 L   X . . .  Y . . .
19 CC  X . . .  Y . . .
20 C   X . . .  Y . . .
— — — — — — — — — — — —
ACTL   X . . .  Y . . .
```

Switchover of position display to large: ▶ ENT



```
PROGRAM RUN/SINGLE BLOCK






18 L   X . . .  Y . . .
ACTL   X . . .
       Y . . .
       Z . . .
       C . . .
```

Switchover from large to small is performed in the same manner.

Block number increment

Select MOD-function "Block number increment" ▶ MOD ↓

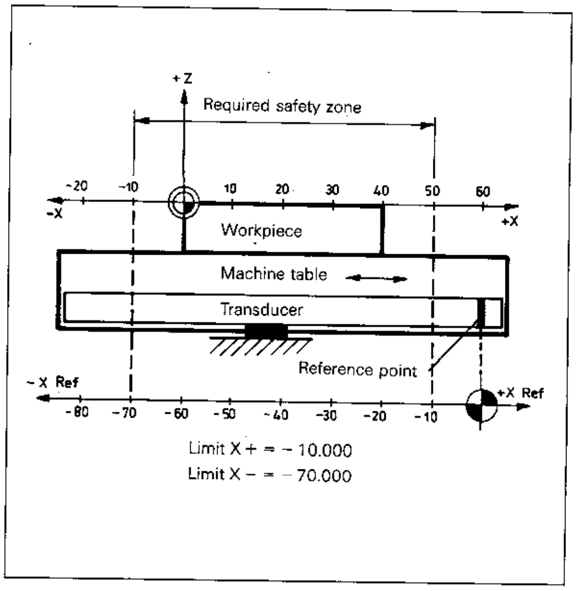
BLOCK NR. INCREMENT = ▶ □ Key-in increment size

ENT Enter into memory

Supplementary operating modes

Limit

With the MOD-function "Limit", traversing ranges can be provided with safety zones e.g. for prevention of workpiece collisions.
Maximum traversing ranges can be defined by software limits. The traversing limits of each axis are set one after the other in the + and − directions, in relation to the reference point. When determining the limit positions, the position display must be switched to REF.



Supplementary operating modes

Setting safety zones

Operating mode ✋ or ☸

When setting safety zones, switch position display to REF.

Select MOD-function "Limit": ▶ MOD ↓

▼

LIMIT X+ = + 30000,000

Traverse to limit position via axis jog buttons or electronic handwheel.

Program displayed position, e.g. – 10.000: ▶ □ Key-in X-value.

▼ ENT Enter into memory.

▼

LIMIT X+ = – 10,000

Select next MOD-function "Limit": ▶ ↓

▼

LIMIT X– = – 30000,000

Traverse to limit position via axis jog buttons or electronic handwheel.

Program displayed position, e.g. – 70.000: ▶ □ Key-in X-value.

▼ ENT Enter into memory.

▼

LIMIT X– = – 70,000

The setting of limits in the remaining traversing ranges is performed in the same manner.

If operation is without safety zones, the values + 30000.000 mm and – 30000.000 must be entered for the appropriate axes.

Supplementary operating modes

NC-Software number

This MOD-function is used for display of the software number for the TNC-Control model.

Display example:

NC: SOFTWARE NUMBER 227 020 08

PLC-Software number

This MOD-function is used for display of the software number of the integral PLC.

Display example:

PC: SOFTWARE NUMBER 228 601 01

User parameters

With this MOD-function, up to 16 machine parameters can be made available to the machine operator. User parameters are allocated by the machine tool builder. Details should be obtained from the machine tool builder.

Code number

This MOD-function can be used for
.a special routine for "reference mark approach" via code numbers or
.the cancellation of edit/erase protection for programs (refer to appropriate section)

Supplementary operating modes

User parameters

Select MOD-function "User parameters"	▶ MOD ↑	
USER PARAMETERS	▶ ENT	Enter MOD-function into memory*
	↓ ↑	Select required user parameter
	□	If necessary, change parameter
	ENT	Enter into memory

Leaving the user parameter mode

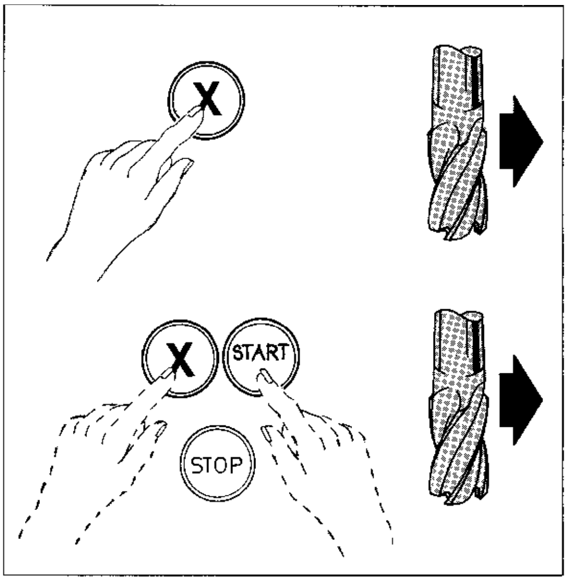
MOD-function, "User parameters" is to be cancelled:	▶ DEL □	Leave MOD-function

*If the machine tool builder has not allocated a dialogue text, the display will show USER PAR. 1

Remarks

Manual operation
Operating mode "Electronic handwheel"

In the manual operating mode [✋], the machine axes can be traversed via the axis jog buttons (X) (Y) (Z) (IV) of the machine.



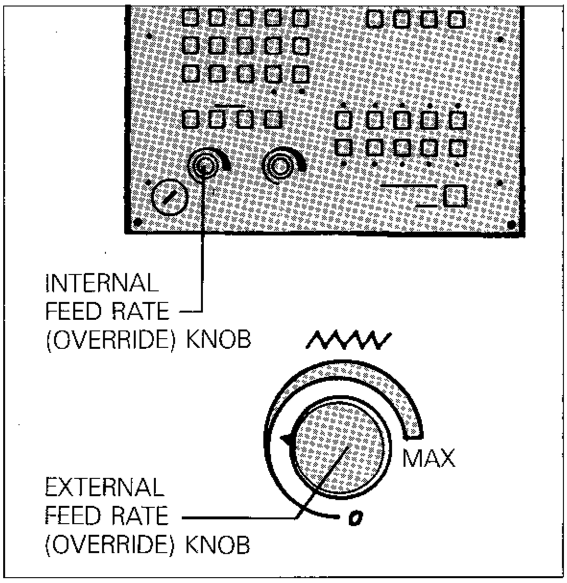
Axis jog operation

The machine axis is traversed as long as the external axis jog button is being pressed. The axis immediately stops when the button is released. A number of axes can be traversed simultaneously in jog operation.

Continuous operation

If the **external start button** is pressed simultaneously with **an axis jog button,** the selected axis traverses although the button has been released. The axis is **brought to a stop** by pressing the **external stop button.**

In the [✋]-mode, the [X] [Y] [Z] [IV] keys are used for definition of the workpiece zero datum (see "workpiece datum").


INTERNAL FEED RATE (OVERRIDE) KNOB
MAX
EXTERNAL FEED RATE (OVERRIDE) KNOB
0

Feed rate

The feed rate (traversing speed) can be set

- with the **internal feed rate override** of the control or
- with the **external feed rate override** of the machine (depending on the entered machine parameters). The feed rate value which has been set is displayed on the screen.

Spindle speed

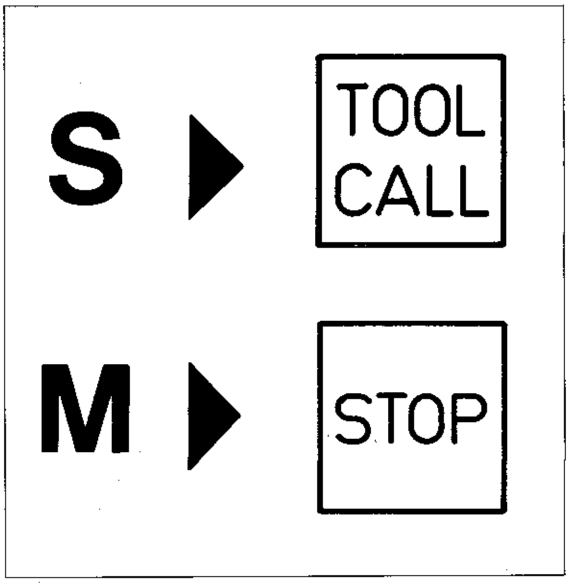
The spindle speed can be defined via the [TOOL CALL] -key (see "TOOL CALL").

With analogue output, the programmed spindle speed can be altered via the spindle override during program run.

The machine tool supplier (or manufacturer) can advise you whether your machine is equipped with a coded or analogue output for spindle speeds.

Auxiliary function

Auxiliary (miscellaneous) functions can be programmed via the [STOP]-key (see "Program stop").

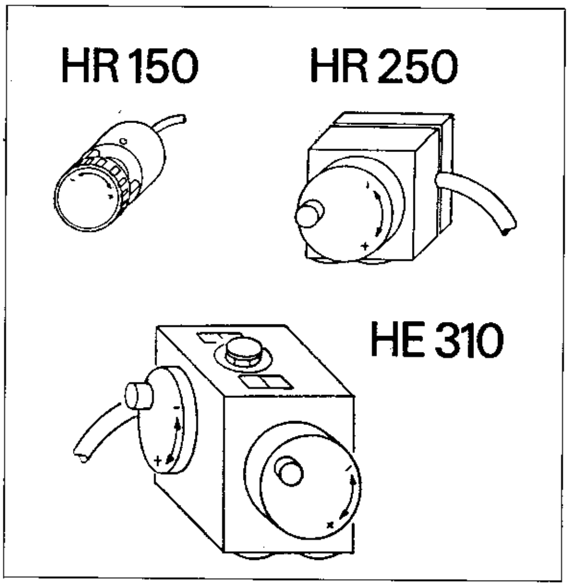


Manual operation
Operating mode "Electronic handwheel"

Versions

The control can be equipped with an electronic handwheel for assisting set-up operations. There are three versions available:

- **HR 150:** 1 Handwheel for incorporation into the machine operating panel;
- **HR 250:** 1 Handwheel in a portable unit;
- **HE 310:** 2 Handwheels in a portable unit with additional axis address keys and emergency stop button.



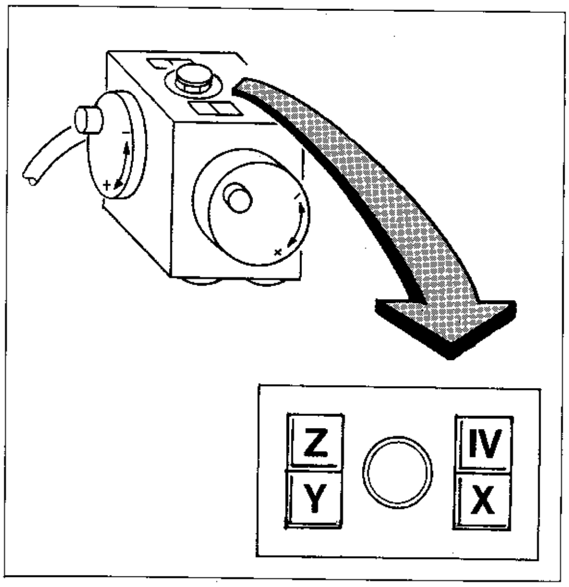
Interpolation factor

Reduction of the traversing distance for each handwheel revolution is determined by the interpolation factor (see adjacent table).

Interpolation factor	Traversing distance in mm per revolution
1	10.0
2	5.0
3	2.5
4	1.25
5	0.625
6	0.313
7	0.156
8	0.078
9	0.039
0	0.020

Operation

With **versions HR 150** and **HR 250** the handwheel is allocated to the axis via the X Y Z IV -keys.
The **version HE 310** with dual handwheels also has additional axis buttons X Y Z IV . This enables one handwheel to be switched to the X or IV-axis and the other handwheel to Y or Z.
The moving axis which is being activated by the handwheel is shown in the display in inverted characters.



In the ⊛ -mode, the machine axes can be additionally traversed via the external jog buttons (X) (Y) (Z) (IV) !

Manual operation
Operating mode "Electronic handwheel"

Operation HR 150/ HR 250

Operating mode and dialogue initiation ____ ⊛

INTERPOLATION FACTOR: 3	**0** ▶ □ ▼ ENT	Key-in required subdivisional factor, e.g. 4. Enter into memory.

▼

INTERPOLATION FACTOR: 4	**4** ▶ Y	Enter required traversing axis, e.g. Y.

The tool can now be moved in the positive or negative Y-direction with the electronic handwheel.

Operation HE 310

Operating mode and dialogue initiation ____ ⊛

INTERPOLATION FACTOR: 4	**0** ▶ □ ▼ ENT	Enter required interpolation factor, e.g. 6. Enter into memory.

▼

INTERPOLATION FACTOR: 6	**6** ▶ Z ▼ X	Enter first traversing axis at the handwheel unit, e.g. Z. Enter second traversing axis at the handwheel unit, e.g. X.

The tool can be moved in the positive or negative Z-direction with the first handwheel and in the positive or negative X-direction with the second handwheel.

Remarks

Co-ordinate system and dimensioning

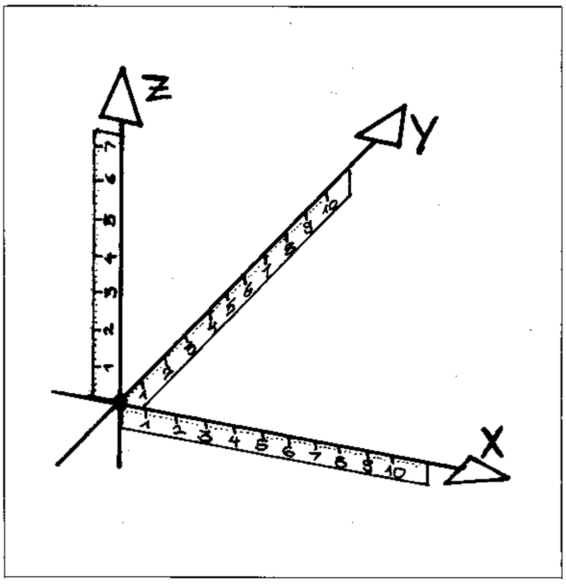
Introduction

An NC-machine is only able to machine a workpiece if all machining operations have been completely defined by the NC-program. For complete machining operation, the nominal positions of the tool – in relationship to the workpiece – must be defined within the NC-program. A reference system i.e. co-ordinate system, is necessary for defining the nominal position of the tool. Depending on the job, the TNC permits the use of either right-angled co-ordinates or polar co-ordinates.

Right-angled or Cartesian*) co-ordinate system

A right-angled co-ordinate system is formed either by two axes in a plane and 3-axes in space. These axes intersect at one point and are also perpendicular to each other. The intersecting point is referred to as the origin or zero-point of the co-ordinate system. Each axis is designated with a letter X, Y or Z.

The axes are each allocated with an imaginative scale, the zero-point of which, coincides with the origin of the co-ordinate system. The arrows indicate the positive counting directions of the scales.



* Named after the french mathematician René Descartes, lat. Renatus Cartesius (1596–1650)

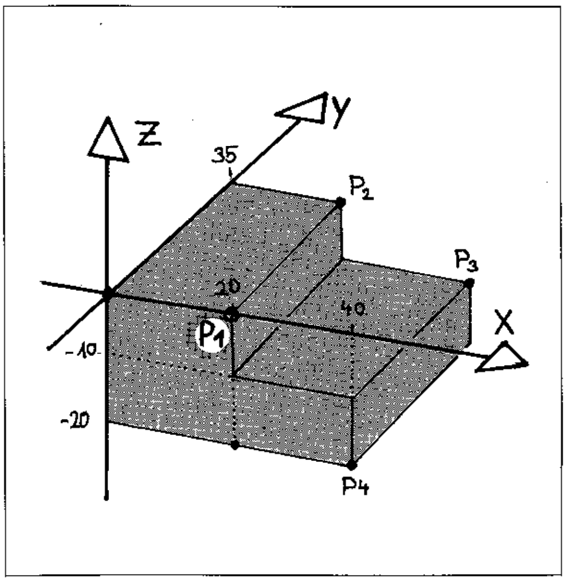
Example

With the aid of the Cartesian co-ordinates system, random points of a workpiece can be located by stating the appropriate X, Y and Z-co-ordinates:

P1 X = 20
Y = 0
Z = 0

abbreviated:
P1 (20; 0; 0)

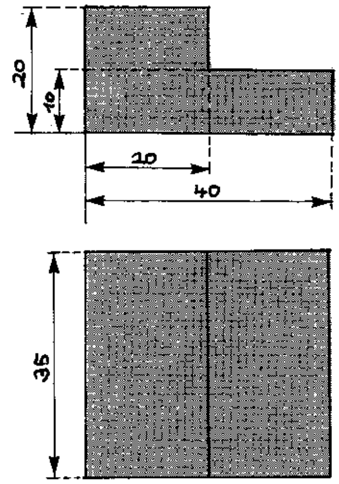
P2 (20; 35; 0)
P3 (40; 35; –10)
P4 (40; 0; –20)



Co-ordinate system and dimensioning

The Cartesian co-ordinate system is particularly convenient if the working drawing is dimensioned as per the adjacent example.

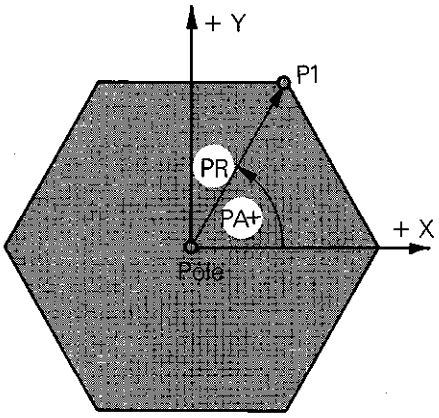
Definition of positions on workpieces incorporating circular elements or angle dimensions is easier with polar co-ordinates.



Polar co-ordinates

The polar co-ordinate system is used for defining points in one plane. System reference is via the pole (= zero-point of co-ordinate system) and the direction (= reference axis for the specific angle).

Points are described as follows:
by specifying the polar co-ordinate radius **PR** (= distance between the pole and point P1) and the angle **PA** between the reference direction (+X-axis, in the adjacent drg.) and the connecting line: pole – point P1.



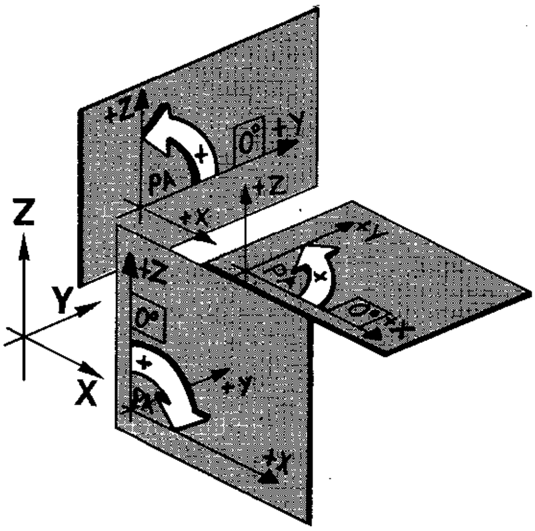
Entry range

The polar co-ordinates angle PA is entered in degrees (°).
Entry range: absolute –360° to +360°
incremental –5400° to +5400°
PA positive: Angle clockwise
PA negative: Angle counter-clockwise

Angle reference axis

The angle reference axis (0°-axis) is
the +X-axis in the XY-plane,
the +Y-axis in the YZ-plane,
the +Z-axis in the ZX-plane.

The sign for the angle PA can be determined in accordance with the adjacent drawing.

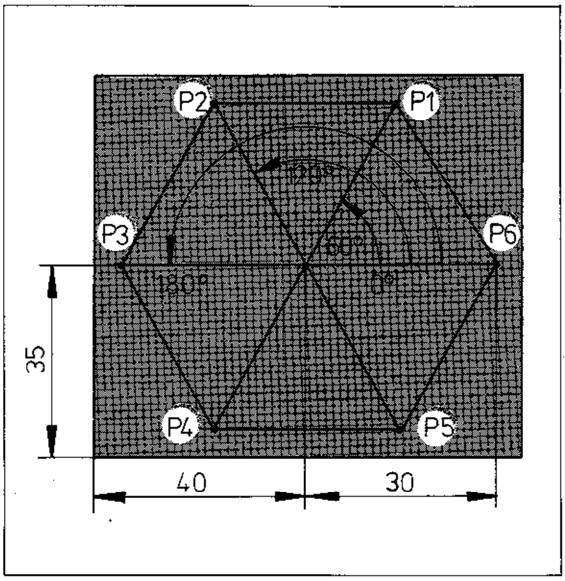


Co-ordinate system and dimensioning

Example

Point	Polar co-ord. radius PR	Polar co-ord. absolute	angle PA incremental
P1	30	60°	60°
P2	30	120°	60°
P3	30	180°	60°
P4	30	240°	60°
P5	30	300°	60°
P6	30	360°	60°

The polar co-ordinate system is particularly useful for defining a workpiece if the working drawing contains a number of angle dimensions as shown in the adjacent example.



Relative tool movement

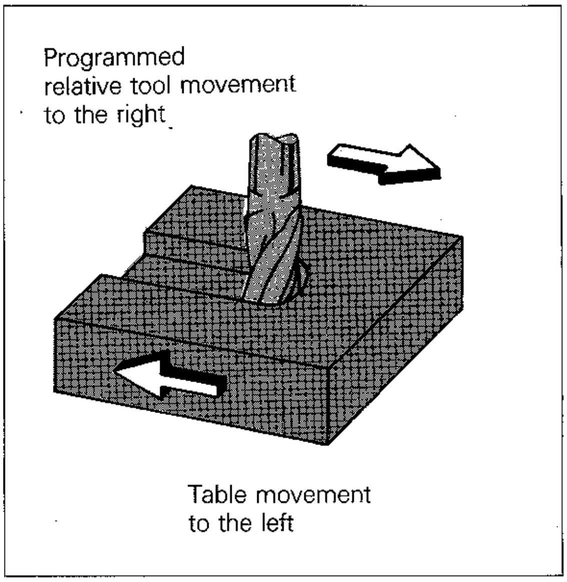
When machining a workpiece, it is irrespective whether the **tool** moves or the **workpiece** moves with the tool remaining stationary.

Only the relative movement is considered when compiling a program.

This means e.g.:
if the milling machine table carrying the workpiece traverses to the left, the relative movement of the tool is towards the right.

If table motion is upwards, the relative tool motion is downwards.

Actual tool motion only takes place if the spindle head is moving, i.e. machine movement always corresponds to the relative tool motion.



Programmed relative tool movement to the right

Table movement to the left

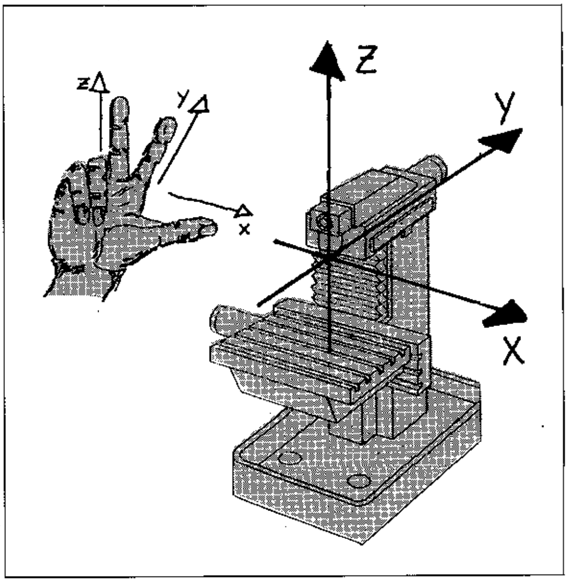
Correlation of machine slide movements and co-ordinate system

In order that workpiece co-ordinates within the machining program can be correctly interpreted by the control, two factors must be clarified:

- which slide will traverse parallel to the co-ordinate axis (correlation of machine axis to co-ordinate axis)
- which relationship exists between machine slide positions and co-ordinate data of the program.

The three main axes

The correlation of the three main co-ordinate axes to the appropriate machine slides is defined by the standard ISO 841 for various machine tools. Traversing directions can be easily remembered by applying the "right-hand rule".

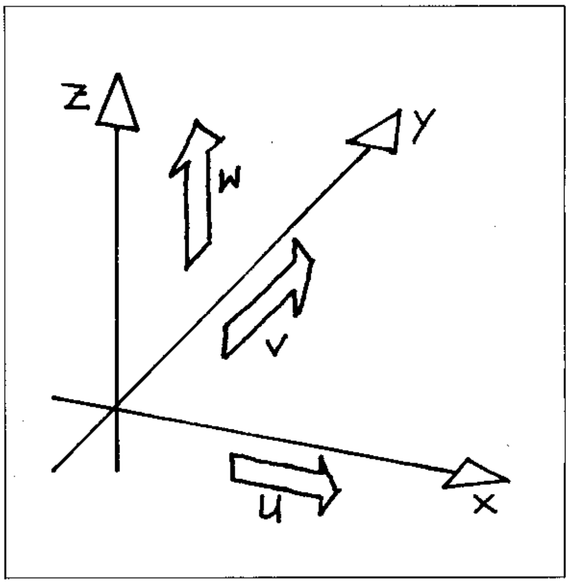


Co-ordinate system and dimensioning

The fourth axis

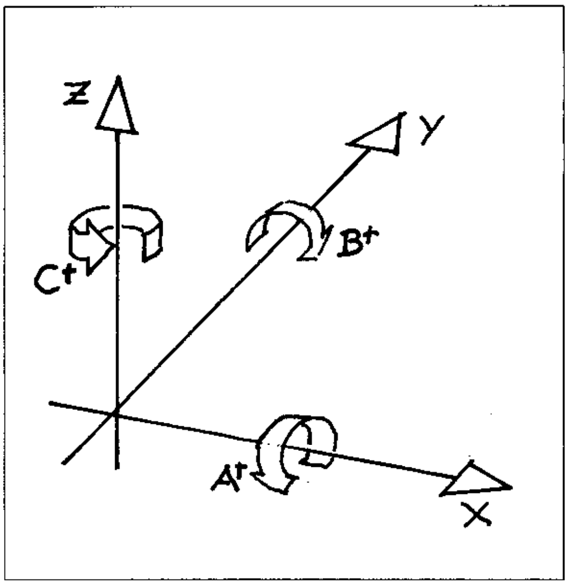
The machine tool builder will determine whether the fourth axis – when switched on – is to be used as a **rotary table** or **linear axis** (e.g. a controlled quill) and how it is to be designated on the VDU-screen.

An additional linear axis with a movement parallel to the X, Y or Z-axis is designated with U, V or W.



When programming rotary table movements, the rotation angle is entered for A, B or C-values in degrees (°).

This axis is referred to as an A, B or C-axis, each rotating about the X, Y or Z-axis.



Co-ordinate system and dimensioning

Correlation of co-ordinate system

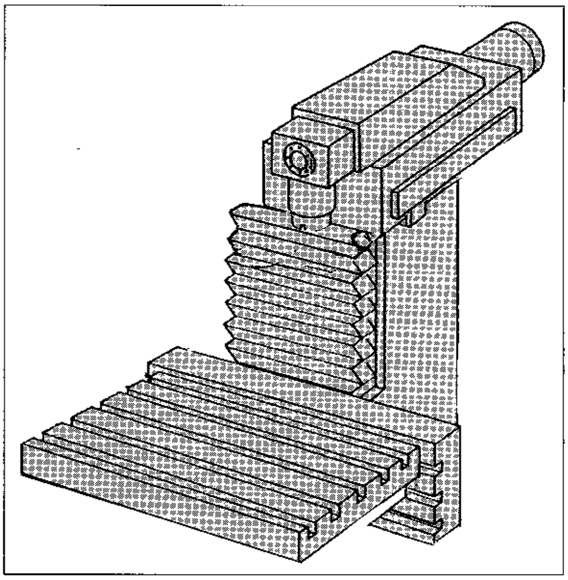
The allocation of the co-ordinate system to the machine is defined as follows:

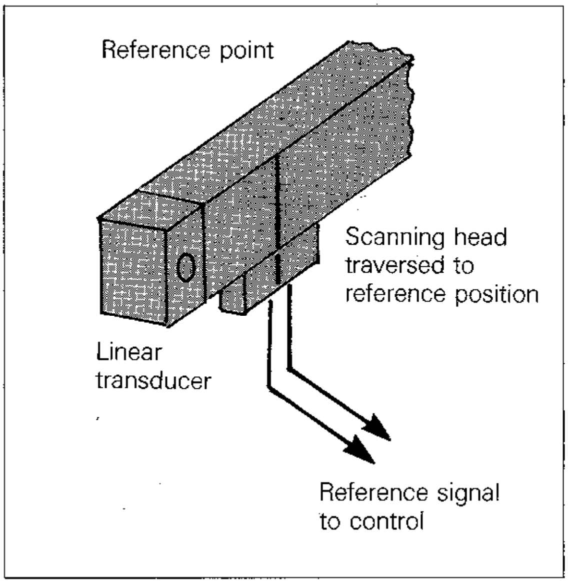
The machine slide is traversed over a defined position – the reference position (also referred to as the reference point). When crossing this point, the control receives an electrical signal from the transducer (reference signal).

On receiving the reference signal, the control allocates a certain co-ordinate value to the reference point.

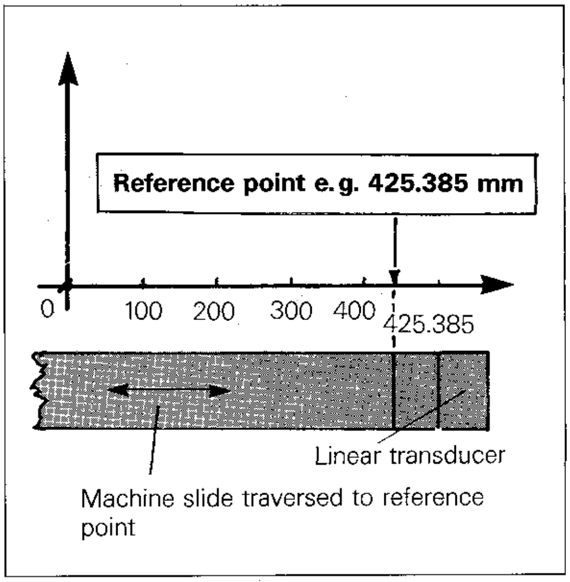
This procedure is repeated for all machine slides.

The co-ordinate system is now correlated to the machine.




Reference point
Scanning head traversed to reference position
Linear transducer
Reference signal to control

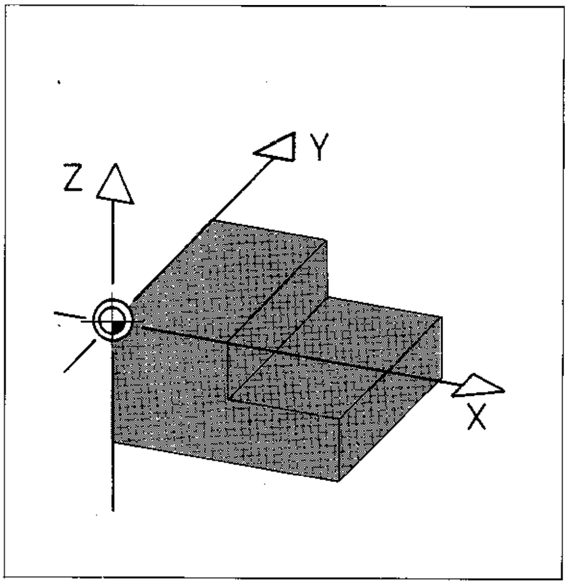
The reference points must be traversed over after every interruption of power supply, otherwise the correlation between the co-ordinate system and the machine slides is lost.
Before this procedure, all other functions are inhibited.
On crossing the reference points, the control then knows where the previous zero datum (refer to following section) and the software limits were located.


Reference point e. g. 425.385 mm
0
100
200
300
400
425.385
Linear transducer
Machine slide traversed to reference point

Co-ordinate system and dimensioning
Setting the workpiece datum

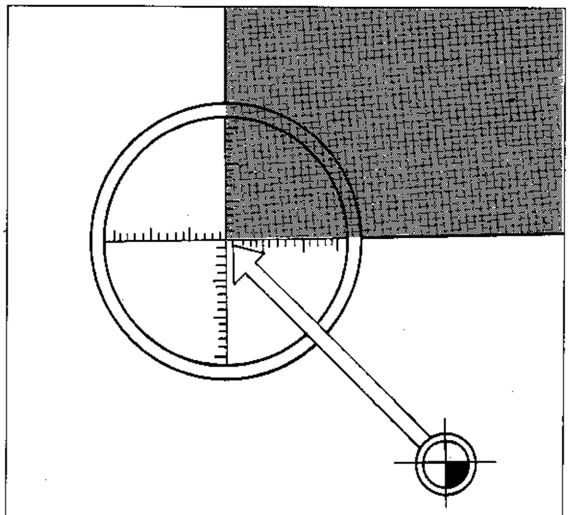
Setting the workpiece datum

To save unnecessary calculation work, the workpiece datum is located at **the point** from which all dimensioning is commenced. For safety reasons, the workpiece datum is always located at the uppermost level of the workpiece in the feed axis.



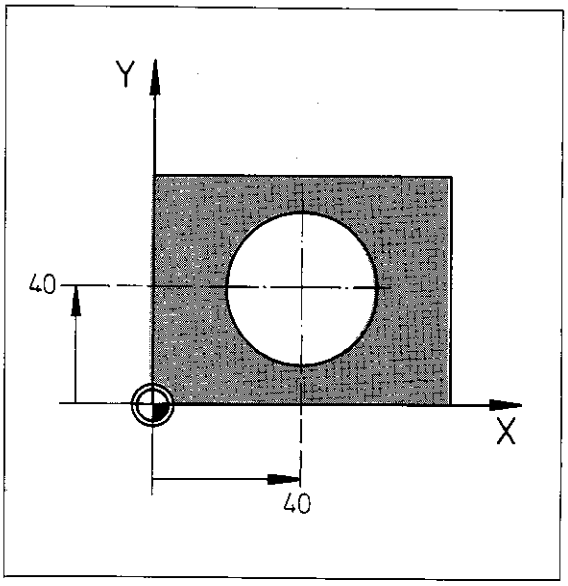
Setting the workpiece datum in the working plane with an optical edge finder

Traverse to the required location for the workpiece datum and reset both axes of the working plane to zero.


Symbol for workpiece datum

With a centring device

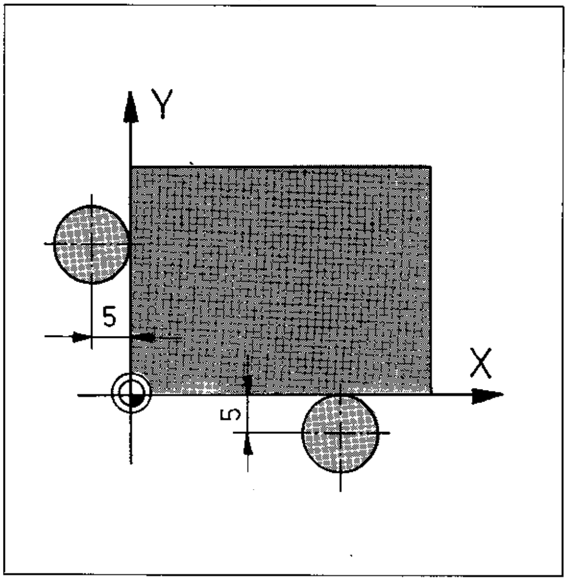
Traverse to a known position e.g. to a hole centre with the aid of the centring device. The co-ordinates of the hole centre are then entered into the control (e.g. X = 40, Y = 40). The location of the workpiece datum is then defined.



Co-ordinate system and dimensioning
Setting the workpiece datum

With touch probe or tool

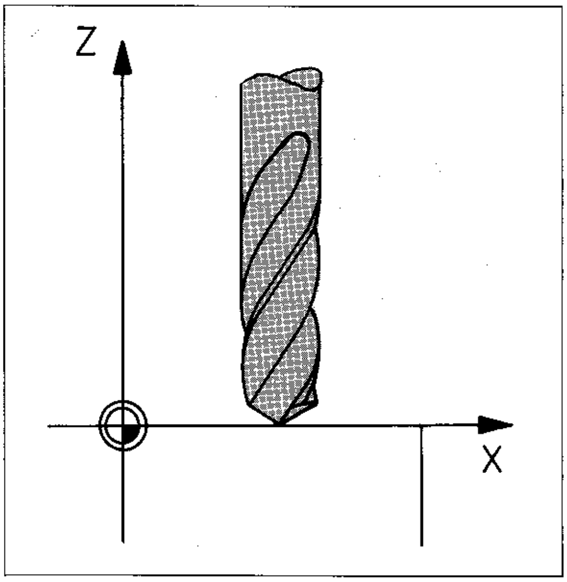
Traverse machine until the tool makes contact with the reference edges of the workpiece. When the tool touches the workpiece edge, preset the position display to the value of the tool radius with negative sign (e.g. X = −5, Y = −5).



Setting the workpiece datum in the feed axis by touching the workpiece surface

Traverse zero-tool to workpiece surface. When the tool tip touches the surface, reset position display of the feed axis to zero.

If touching of the workpiece surface is undesired, a small metal plate with a known thicknes (e.g. 0.1 mm) may be placed between the tool tip and the workpiece. Instead of zero, the thickness of the plate is entered (e.g. Z = 0.1).



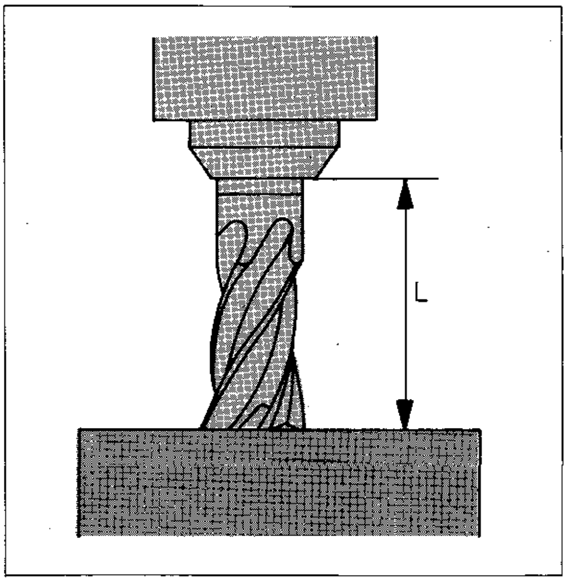
With preset tools

With preset tools, i.e. when the tool length is already known, the workpiece surface is touched with a random tool. In order to allocate the workpiece surface to the value zero, the known length L of the tool is entered as an actual position value – with positive sign – for the feed axis.

If the workpiece surface is to have a preset value differing from zero, the following value is to be entered:

(Actual value Z) = (Tool length L) + (surface position)

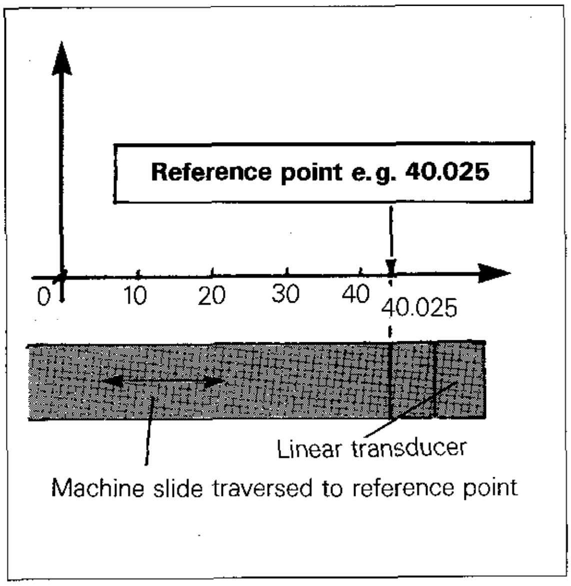
Example:
Tool length L = 100 mm
Position of workpiece surface = + 50 mm
Actual value Z = 100 mm + 50 mm = 150 mm



Co-ordinate system and dimensioning
Setting the workpiece datum

When setting the zero datum of the workpiece, definite numerical values ("REF-values") are allocated to the reference points.

The control automatically memorizes these values. After an interruption of power supply, simple reproduction of the workpiece datum is now possible by traversing over the reference points.


Reference point e.g. 40.025
0
10
20
30
40
40.025
Linear transducer
Machine slide traversed to reference point

Co-ordinate system and dimensioning
Setting the workpiece datum

Setting the workpiece datum

Operating mode ✋

The workpiece datum can only be set if the actual position is being displayed.

If necessary, select this display mode via the MOD-function.

Dialogue initiation X

DATUM SET X =

▶ □ Key-in value for X-axis.

▼ (ENT) Enter into memory.

Dialogue initiation Y

DATUM SET Y =

▶ □ Key-in value for Y-axis.

▼ (ENT) Enter into memory.

Dialogue initiation Z

DATUM SET Z =

▶ □ Key-in value for Z-axis.

▼ (ENT) Enter into memory.

Dialogue initiation IV

DATUM SET C =

▶ □ Key-in value for 4 axis.

▼ (ENT) Enter into memory.

Depending on the machine parameters which have been entered, the 4 axis is designated and displayed with A, B, C or U, V, W.

If the dialogue for datum set has been inadvertently initiated, and a datum set is not intended, the following key is to be pressed:

NO ENT when programming in HEIDENHAIN-format

DEL when programming in standard ISO (G-code) format

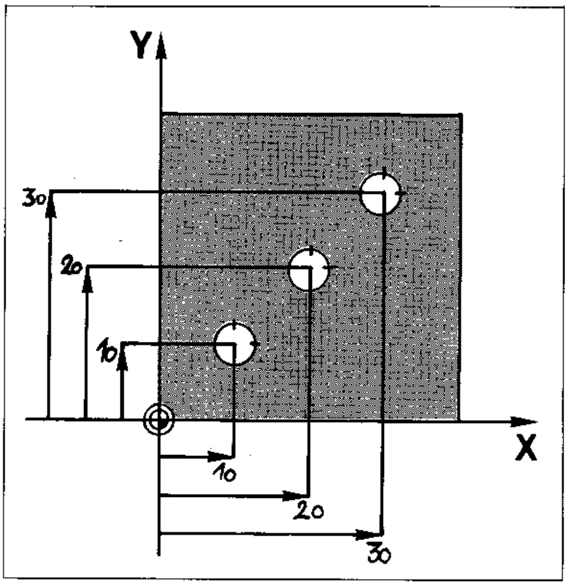
Co-ordinate system and dimensioning
Absolute/Incremental dimensions

Dimensioning

Dimensions in working drawings are either absolute or incremental dimensions.

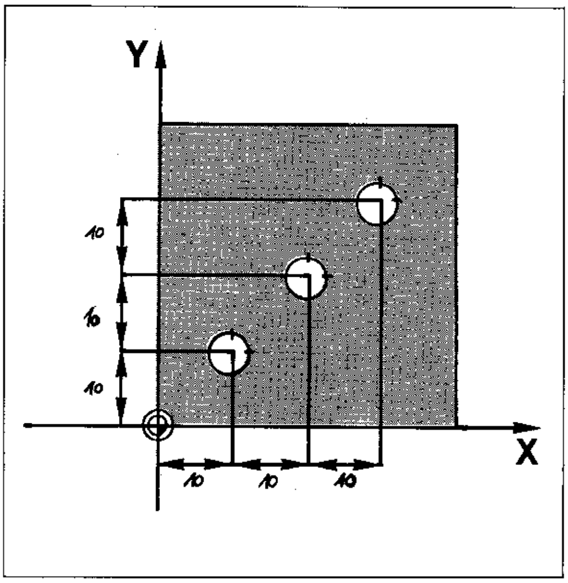
Absolute dimensions

Absolute dimensions of a machining program are referenced to a fixed absolute point e.g. the zero datum of a co-ordinate system or a workpiece datum.



Incremental dimensions

Incremental dimensions of a machining program are referenced to the previous nominal position of the tool.

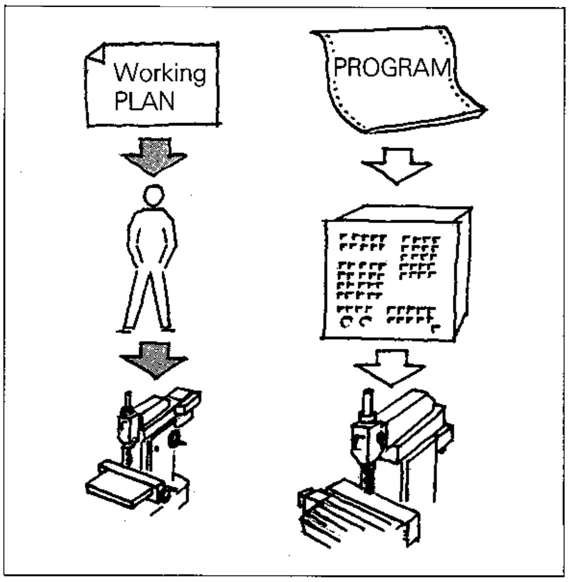


Programming
Introduction

Introduction

As with manual-operated machine tools, a working plan is also required for NC-machine tools. The sequence of operations is the same.

On manually-operated machines, each working step must be executed by the operator; however, on an NC-machine, the electronic control performs the calculation for the tool path, the co-ordination of the feed movements of the machine slides and the supervision of the spindle speed. For this, the control receives the information from a program which has been entered.



Program

The program can be simply regarded as a working plan which is written in a certain language.

Programming

Programming is the compilation and entry of such a working plan in a language which is comprehensible to the control.

Programming language

In a machining program every **NC-programming block** correspond to a working step. A block consists of **single commands.**

Examples

Programmed working step	Meaning
Y–50,000	Traverse the Y-axis slide to the position –50.000 mm
F250	Traverse the axis slide with a feed rate of 250 mm/min.
TOOL CALL 1	Call-up of compensation values for tool number 1

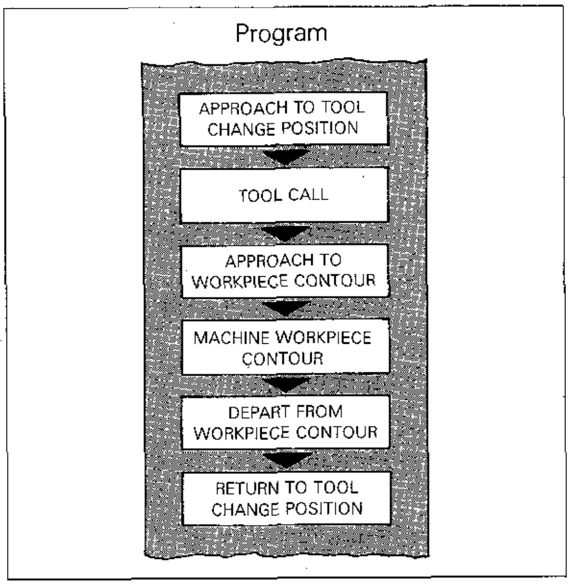
Programming
Program

Program structure

A program which is used for the manufacture of a workpiece can be subdivided into the following **sections:**

- Approach to tool change position,
- Insert tool,
- Approach to workpiece contour,
- Machine workpiece contour,
- Depart from workpiece contour
- Return to tool change position.

Each program section comprises individual program blocks.


Program
APPROACH TO TOOL CHANGE POSITION
TOOL CALL
APPROACH TO WORKPIECE CONTOUR
MACHINE WORKPIECE CONTOUR
DEPART FROM WORKPIECE CONTOUR
RETURN TO TOOL CHANGE POSITION

Block number

The control automatically allocates a block number to each block. The **block number** designates the program block within a machining program.

When erasing a block, the block number remains and the subsequent block then shifts to the allocation of the erased block.

```
 7 L     Z - 20,000
                           R0 F9999     M03
 8 L     X - 12,000        Y + 60,000
                           R0 F9999     M
 9 L     X + 20,000        Y + 60,000
                           RR F40       M
10 RND   R + 5,000
11 L     X + 50,000        Y + 20,000
                           RR F40       M
12 CC    X - 10,000        Y + 80,000
13 C     X + 70,000        Y + 51,715
                     DR +  RR F40       M
14 CC    X + 150,000       Y + 80,000
15 C     X + 90,000        Y + 20,000
                     DR +  RR F40       M
16 L     X + 120,000       Y + 20,000
                           RR F40       M
```

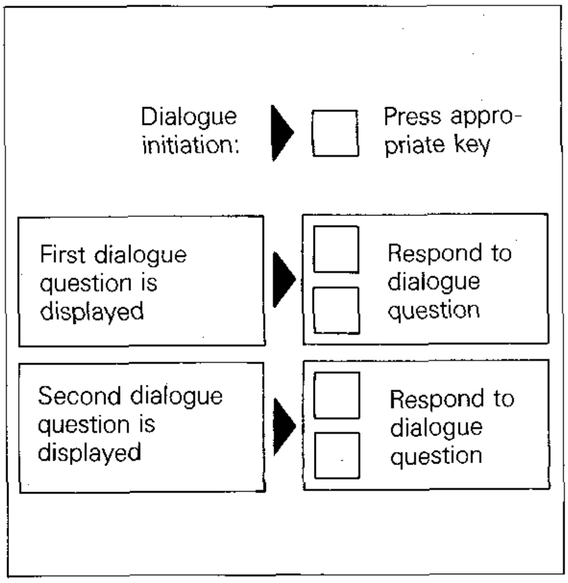
Dialogue prompting

Programming is guided by a prompting routine, i.e. during program entry, the control asks for the necessary data in plain language.

With every block, a sequence of dialogues is opened by pressing the dialogue initiation key e.g. TOOL DEF (the control subsequently asks for the tool number and then the tool length etc.).

The operator is made aware of entry errors via the plain language display. Incorrect data can be amended immediately during program entry.

Program entry is performed in the operating mode: PROGRAMMING AND EDITING →


Dialogue initiation:
Press appropriate key
First dialogue question is displayed
Respond to dialogue question
Second dialogue question is displayed
Respond to dialogue question

Programming
Responding to dialogue questions

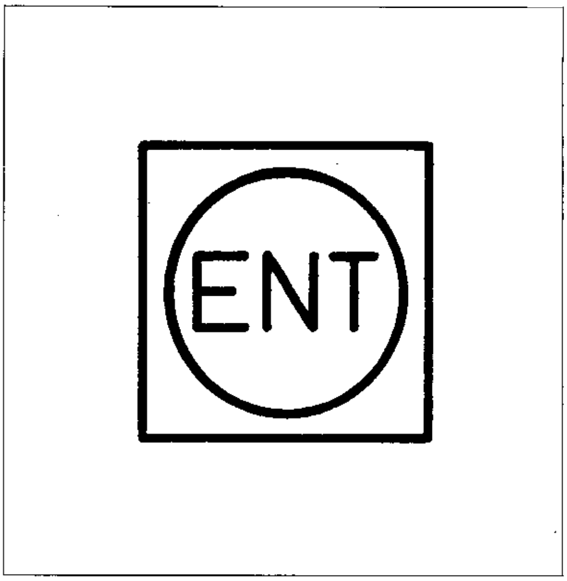
Responding to dialogue questions

Every dialogue question must be responded to. The response is displayed in the inverted character line on the screen.

After complete response of the dialogue question, the entered data is transferred into the memory by pressing ENT.

"ENT": Abbreviation for the word "enter".

When programming an axis without a numerical value (e.g. mirror image axis), the ENT -key must not be pressed.

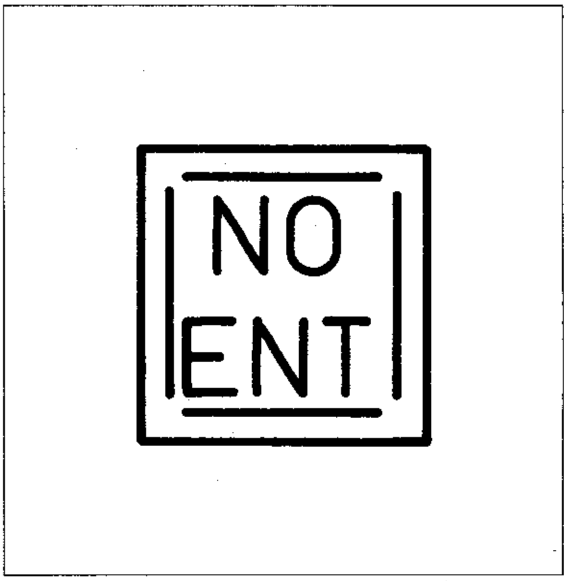


Omission of dialogue questions

Certain entry data remains identical from block-to-block, e.g. the feed rate or spindle speed. Such dialogue questions do not have to be answered and can be "skipped over" by pressing NO ENT.

The data which is already displayed in the inverted character line is erased and the next dialogue question appears.

When executing the program, the data previously entered under the appropriate address is valid.

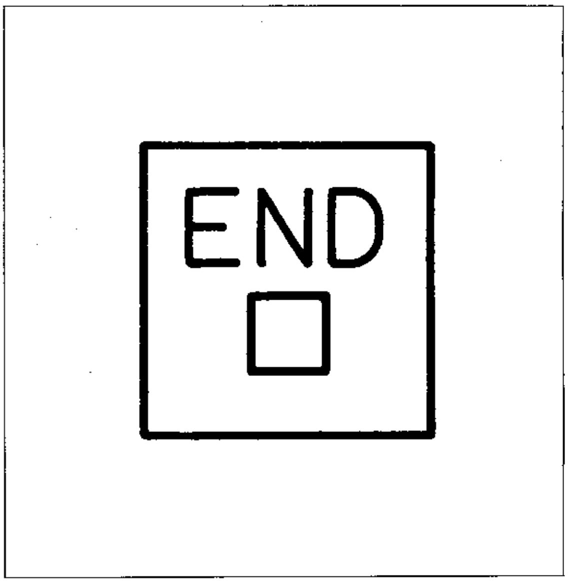


Curtailed blocks

By using the END □ -key, it is possible to curtail the programming of positioning blocks, tool calls or the cycles "datum shift" and "mirror image". The END □ -key can be pressed for transferring the data into the memory (as per ENT), or for direct access to the subsequent dialogue question (as per NO ENT).

When executing the program, the data previously entered under the appropriate address is valid.

□ is the symbol for a program block.



Programming
Entry of numerical values

Entry of numerical values

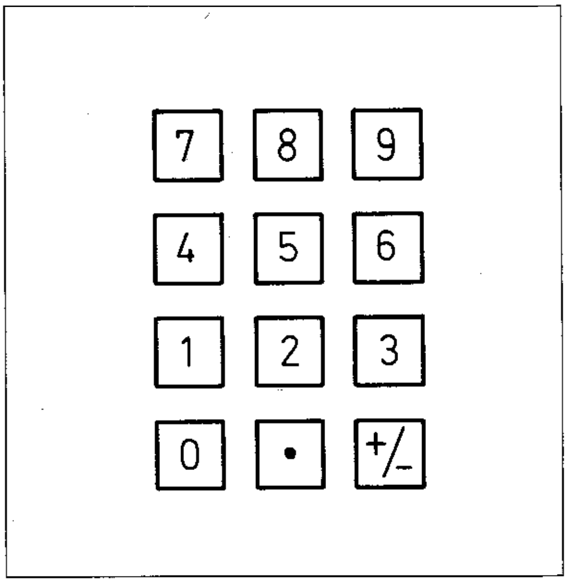
Numerical values are entered on the decimal keyboard – with decimal point and arithmetical sign. Leading zeros before the decimal point may be neglected. (The decimal point is displayed as a decimal comma)

Entry of the arithmetical sign is possible prior, during or after entry of the numerical value.

Incorrect entries can be erased by pressing the CE-key (clear entry) – before transferring into the memory – and re-entered correctly.

On pressing CE, a zero appears in the inverted character line!

If an entry is not intended, the NO ENT-key is to be pressed!



Remarks

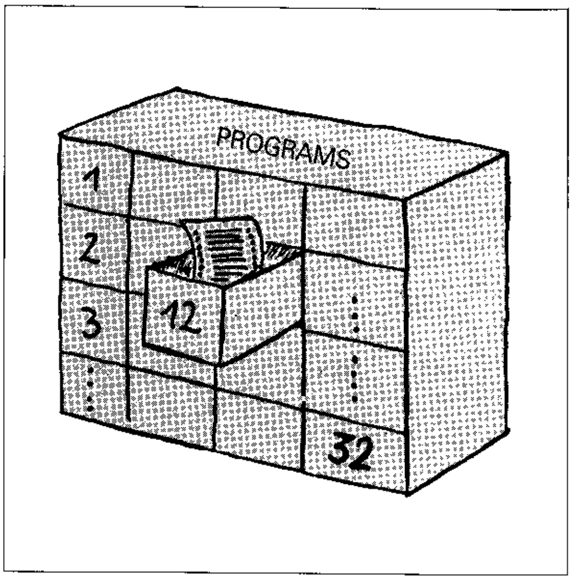
Program management

Erase/Edit protection

The control has the capability of storing up to 32 programs with a total of 3100 program blocks.

In order to differentiate between programs, each program is designated with a **program number.**

A machining program can consist of max. 999 blocks.



Protection against erasing and editing

Programs may be protected against direct intervention (e.g. program editing or erasing).

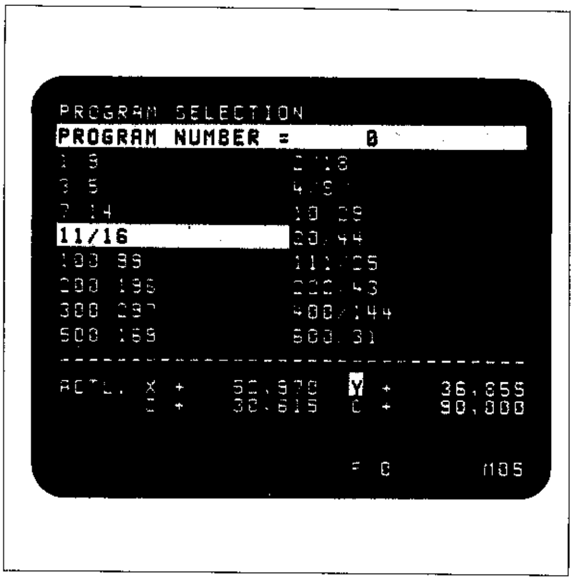
Program list

The dialogue for entry or call-up of a program number is initiated by pressing PGM NR.
The display shows the **program directory** listing all the programs which are contained in the program memory. The **program extent** is indicated behind the program number **(total number of program blocks).**

Call-up of an existing program

Programs already entered are called-up via the program number. This can be performed in two ways:

- Programs which are stored in the control memory are displayed on the screen with the appropriate program number. The number last entered or called-up is shown in inverted characters.
  The inverted character cursor can be shifted within the table of numbers by using the editing keys ↓ ↑ ← →.
  The program within the inverted character cursor is called-up by pressing ENT.
- A program may be called-up by keying-in the program number and pressing ENT.



```
PROGRAM SELECTION
PROGRAM NUMBER =        0
1  9                   2 18
3  5                   4 9
7  14                  10 29
11/16                  20 44
100 99                 111 25
200 196                222 43
300 297                400 144
500 169                600 31
--------------------------------------
ACTL. X +    50,970  Y +    36,655
      Z +    30,615  C +    90,000

                     F 0        M05
```

Program management

Entry of a new program number

Operating mode ____________________ ⇨

Dialogue initiation ____________________ PGM NR

PROGRAM SELECTION

PROGRAM NUMBER

▶ □ Enter program number (max. 8 digits).

▼ ENT Enter into memory.

▼

MM = ENT/INCH = NO ENT

▶ ENT for **dimensions in mm**

or

▶ NO ENT for **dimensions in inches**

Display example

```
0  BEGIN  PGM  12345678    MM

1  END    PGM  12345678    MM
```

The program is numbered 12345678; dimensions are in mm. When programming, the machining program is inserted between the BEGIN-block and the END-block.

Selecting an existing program number

Operating mode ____________________ ⇨ or → or ⇉ or →

Dialogue initiation ____________________ PGM NR

PROGRAM SELECTION

PROGRAM NUMBER =

Either select program number using the reverse video cursor:

▶ ↓ ↑ ← → Set cursor to desired number.

▼ ENT Enter number into memory.

Or key-in the program number:

▶ □ Key-in number.

▼ ENT Enter into memory.

Display example

```
0  BEGIN  PGM  8324    MM

1  L...
```

The beginning of the selected program appears on the screen.

Program management
Programs with edit protection

Erase/Edit protection

After program compilation, an entry can be made for erase/edit protection. Programs having protection against erasing and editing are marked with the letter P at the beginning and end of the program.

A protected program can only be erased if the erase/edit protection has been cancelled. This can be done by addressing the program and entering the code number 86357.

Program management
Programs with edit protection

Entry of erase/ edit protection

Operating mode ______________ ⇨

0 BEGIN PGM 22 MM ▶ →

Select block number 0 of program to be protected.

Press until dialogue question PGM-Protection is displayed.

▼

PGM-PROTECTION? ▶ ENT ▼ →

0 BEGIN PGM 22 MM P

Protection is programmed.

Display example

0 BEGIN PGM 22 MM P

1 L...

2 L...

Cancellation of erase/edit protection

0 BEGIN PGM 22 MM P ▶ MOD

Select program which is to have protection cancelled.

Select supplementary mode.

▼

VACANT BLOCKS 2951 ▶ ↑

Select MOD-function "Code number".

▼

CODE NUMBER = ▶ □ ▼ ENT

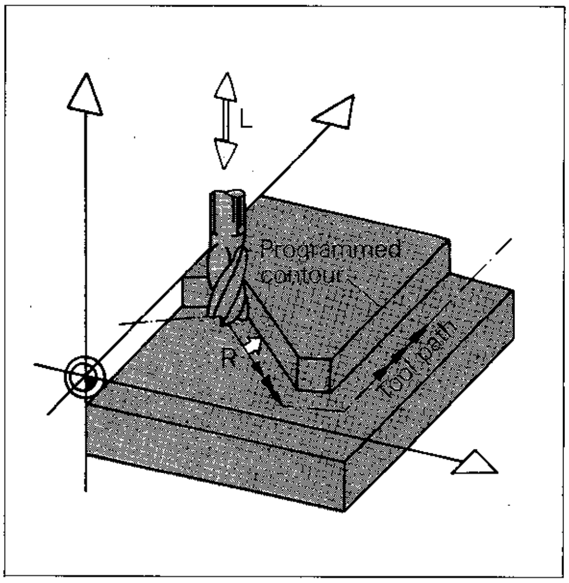
Key-in code number **86357.**

Erase/edit protection is cancelled.

Programming of tool compensation

Tool definition TOOL DEF

In order that the control can calculate a tool path which conforms to an entered workpiece contour, the tool length and radius must be entered. These data are programmed within the TOOL DEFINITION.



Tool number

Compensation (or offset) values are related to a certain tool which has a certain tool number.

Entry values for the tool number depend on the type of machine tool:

with automatic tool changer: 1 – 99,

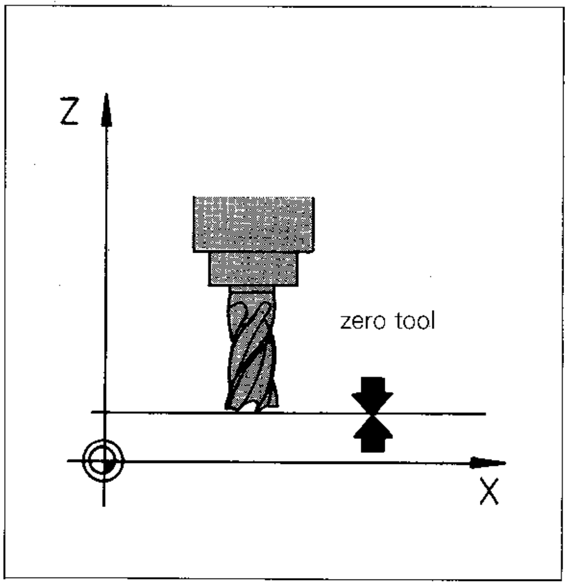
without automatic tool changer: 1 – 254.

Tool length

The **offset value** for the tool length can be determined on the machine or on a tool presetter.

If the length offset is to be determined at the machine, the workpiece zero datum ⊕ is to be defined. The tool with which the workpiece zero datum was set has the offset value 0 and is referred to as the **"zero-tool".**

Length offset values of the remaining tools correspond to the **length difference** from the zero-tool.

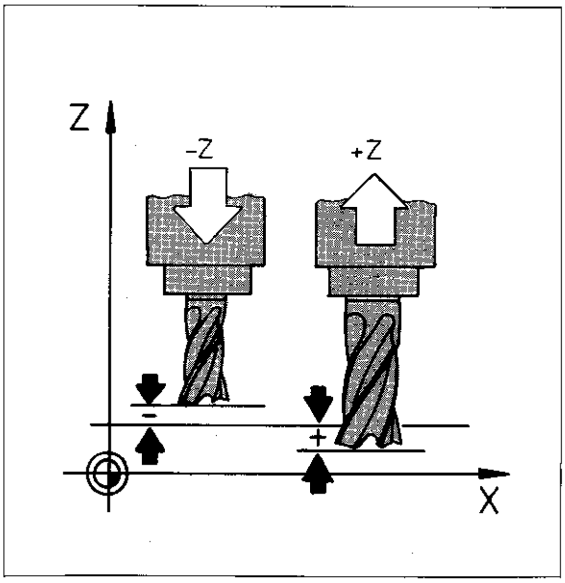


Arithmetical sign

If a tool is **shorter** than the zero-tool, the difference is programmed as a **negative** offset value.

If a tool is **longer** than the zero-tool, the difference is programmed as a **positive** offset value.

If a **tool presetter** is being used, all tool lengths are already known. The offset values are entered from a list with the correct arithmetical sign.

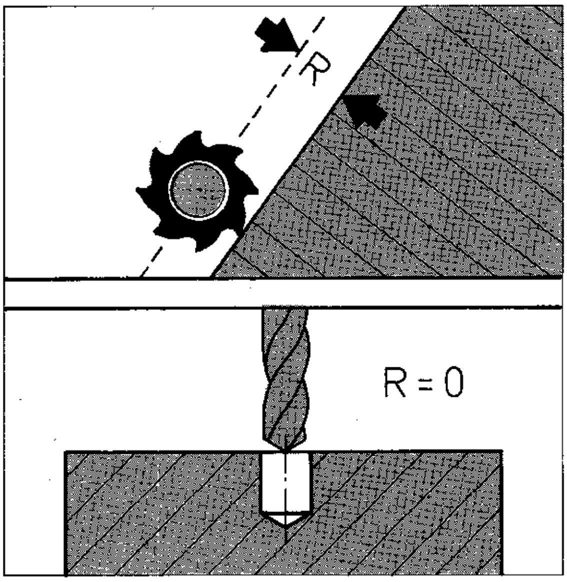


If the tool length is determined on the machine, the difference in length can be entered by pressing +.

Programming the workpiece contour

Tool radius

A tool radius offset is always entered as a positive value (exception: radius compensation with playback programming).



For drilling and boring tools, the value 0 can be entered.

Possible entry range: ± 30000.000 mm

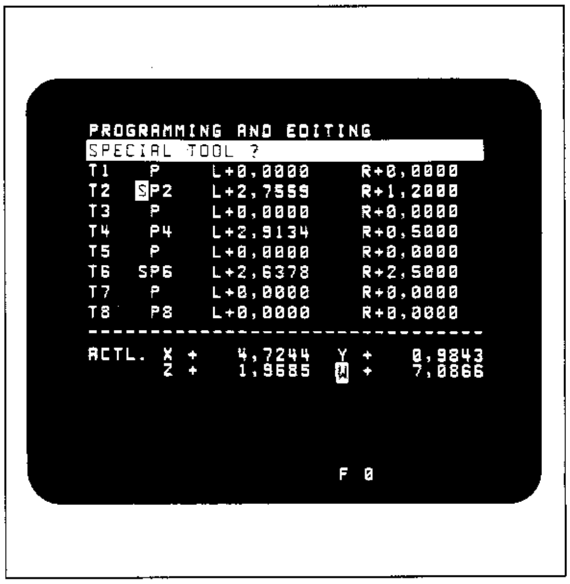
If a program is to be checked with the aid of the TNC 155 graphics function, a tool radius must always be programmed.

Programming of tool compensation

Central tool store

As of software version 03, TNC 151 and TNC 155 can activate a central tool store via machine parameters.

The central tool store is addressed via the program number 0 and can be amended, output and input in the ◈ "programming"-mode. Up to 99 tools can be stored. Each tool is entered with a tool number, length, radius and store location.



```
PROGRAMMING AND EDITING
SPECIAL TOOL ?
T1    P     L+0,0000     R+0,0000
T2   SP2    L+2,7559     R+1,2000
T3    P     L+0,0000     R+0,0000
T4    P4    L+2,9134     R+0,5000
T5    P     L+0,0000     R+0,0000
T6   SP6    L+2,6378     R+2,5000
T7    P     L+0,0000     R+0,0000
T8    P8    L+0,0000     R+0,0000
-----------------------------------
ACTL. X +    4,7244  Y +    0,9843
      Z +    1,9685  W +    7,0866

                     F 0
```

Toolchanger with random select facility

When using a toolchanger with random select, i.e. variable tool location coding, the control is responsible for the tool management. Random tool selection operates as follows: Whilst a certain tool is being used for machining, the control is already searching for the next tool to be used. When a tool change takes place, the tool last used is exchanged for the new tool. The control automatically registers the tool number and in which store location is was last placed. The tool which is to be searched for is programmed with the TOOL DEF-key. (Caution! This is a new function for the TOOL DEF-key).

Tools which, due to their size, allocate three locations, can be defined as special purpose tools. A special purpose tool is always located to a fixed location. This is programmed by setting the cursor in response to the dialogue question

SPECIAL TOOL?

and replying with ENT .

Blockwise transfer

In the „blockwise transfer"-mode, compensation values can be called-up from the central tool store.

Programming of tool compensation
Tool definition

Entry of tool offsets

Operating mode ________ ◈

Dialogue initiation ________ TOOL DEF

TOOL NUMBER?

▶ □ Key-in tool number.

▼ ENT Enter into memory.

▼

The tool number 0 must not be programmed within the TOOL DEF entry procedure. This number has already been allocated by the control internally (see "TOOL CALL").

Tool length and tool radius can also be entered via the playback procedure (see "Tool compensation with playback").

▼

TOOL LENGTH L?

▶ □ or ⌖ Key-in difference value from zero tool or enter by pressing actual position data key.

▼ ENT Enter into memory.

▼

TOOL RADIUS R?

▶ □ Key-in tool radius.

▼ ENT Enter into memory.

Display example

15 TOOL DEF 28 L + 15,780
R + 20,000

Tool No. 28 has offset values 15.780 for the length and 20 mm for the radius.

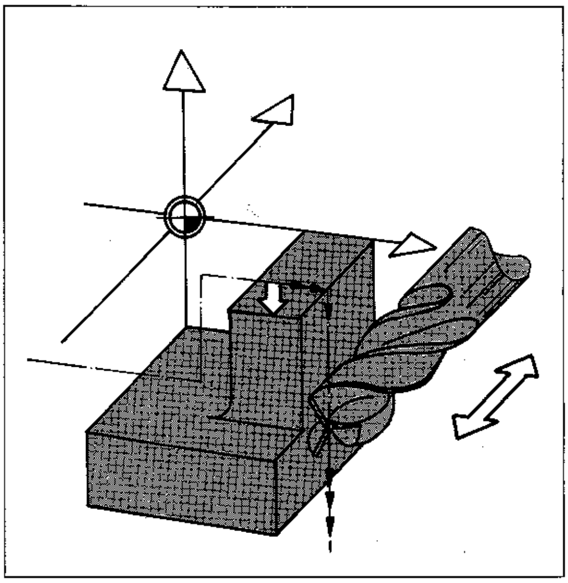
Programming of tool compensation
Tool call

Calling-up a tool TOOL CALL

With TOOL CALL, a new tool and the corresponding compensation values for length and radius are called-up.

In addition to the **tool number,** the control must also know in which axis the spindle will operate, in order to apply both-the length compensation in the correct axis-and the radius compensation in the correct plane.

After specification of the working spindle axis, the **spindle speed** must be entered.
If a spindle speed lies outside of the permissible range for the machine, the following error message is displayed during program run:
= WRONG RPM =



Tool change

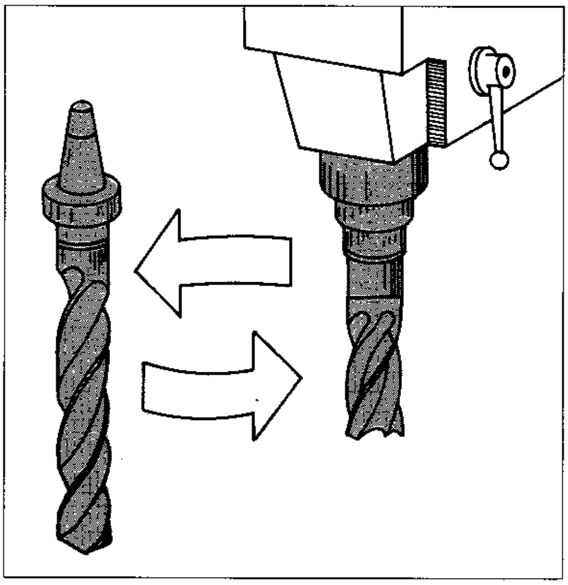
Tool change takes place in a definite **tool change position.** The control therefore positions the tool to a position with **non-compensated nominal values** for execution of tool change. For this, the compensation data for the tool currently in operation must be cancelled.

This is done via a

TOOL CALL 0:

The tool is positioned to the required non-compensated nominal position which is programmed in the following block.

Traverses to the tool change position can be executed via M91, M92 (Auxiliary functions M) or via a PLC-positioning command. (Information can be obtained from the machine tool supplier).

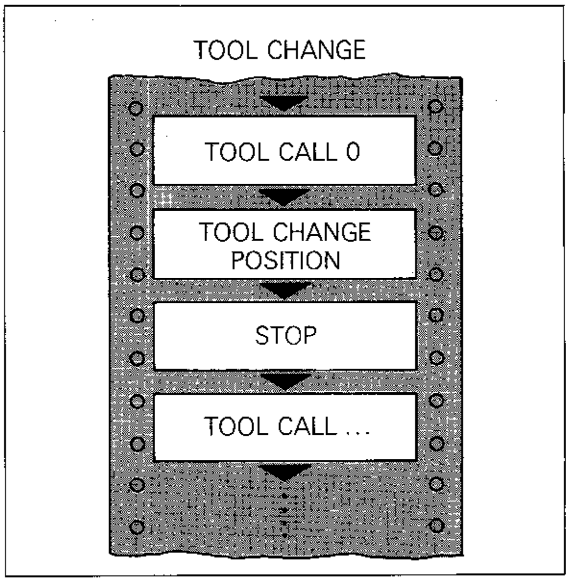


Program structure

When performing a manual tool change, the program must be stopped. A STOP-command is therefore required before the TOOL CALL-command. The program remains in a stopped condition until the external start button is pressed.

If a tool call is only programmed for the purpose of speed-change, the programmed STOP may be neglected.

An **automatic tool change** does not require a programmed STOP. Program run is continued when the tool change procedure is finalised.



Programming of tool compensation
Tool call/Program run stop

Entry of a tool call command

Operating mode ________________ ⟐

Dialogue initiation ________________ TOOL CALL

TOOL NUMBER?	▶ □	Key-in tool number.
	ENT	Enter into memory.

WORKING SPINDLE AXIS X/Y/Z?	▶ Z	Enter working spindle axis, e.g. Z.

SPINDLE SPEED S RPM?	▶ □	Key-in spindle speed (refer to table on next page).
	ENT	Enter into memory.

Display example

```
TOOL CALL 5          Z

          S 125,000
```

Tool number 5 has been called-up. The working spindle axis is operating in the Z-direction; the spindle speed is 125 rpm.

Entry of a programmed stop

Operating mode ________________ ⟐

Dialogue initiation ________________ STOP

AUXILIARY FUNCTION M?

Auxiliary function required:	▶ □	Key-in auxiliary function.
	ENT	Enter into memory.
Auxiliary function not required:	▶ NO ENT	Data entry not required.

Display example

```
18 STOP

               M
```

Program run is stopped at block No. 18.

No auxiliary function.

Tool call
Spindle speeds

Programmable spindle speeds (with coded output)

S in rpm	S in rpm	S in rpm	S in rpm	S in rpm
0	1	10	100	1000
0.112	1.12	11.2	112	1120
0.125	1.25	12.5	125	1250
0.14	1.4	14	140	1400
0.16	1.6	16	160	1600
0.18	1.8	18	180	1800
0.2	2	20	200	2000
0.224	2.24	22.4	224	2240
0.25	2.5	25	250	2500
0.28	2.8	28	280	2800
0.315	3.15	31.5	315	3150
0.355	3.55	35.5	355	3550
0.4	4	40	400	4000
0.45	4.5	45	450	4500
0.5	5	50	500	5000
0.56	5.6	56	560	5600
0.63	6.3	63	630	6300
0.71	7.1	71	710	7100
0.8	8	80	800	8000
0.9	9	90	900	9000

With coded output, the spindle speeds must lie within the standard range. If necessary, the control will round-off the value to the next highest standard value.

Programmable spindle speeds (with analogue output)

Programmed spindle speeds do not have to correspond to the values given in the table. Any desired spindle speed may be programmed provided it is not below the minimum speed and does not exceed the maximum speed.

Moreover, the "spindle override" potentiometer enables the programmed speed to be superimposed by a set %-factor.

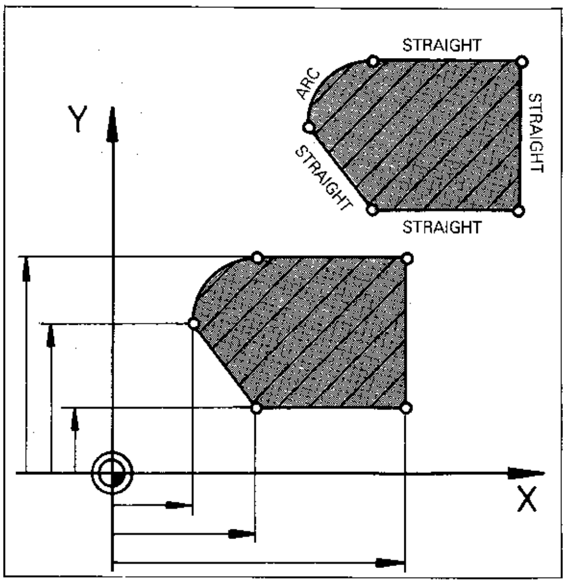
With TNC 155 as of software version 06 and TNC 151, the max. entry value with analogue output of spindle speeds has been increased to 30000 rpm.

☝ The spindle speed function will be active if the machine is equipped with an analogue or coded spindle speed output.

Programming of workpiece contours
Contour

Workpiece contour

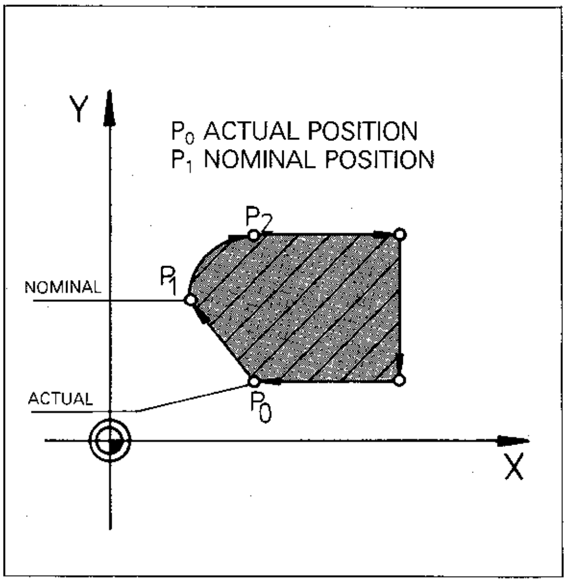
Workpiece contours which are programmed with the TNC 151/TNC 155 consist of the contour elements **straights** and **arcs.**



Construction of a workpiece contour

For construction of a workpiece contour, the control must receive information regarding the type and location of individual contour elements. Since the next machine step is determined in each program block, it is sufficient

- to enter the **co-ordinates** of the next target position and
- specify with which **type of path** (straight, arc or spiral) the next target position is to be reached.



Programming of co-ordinates

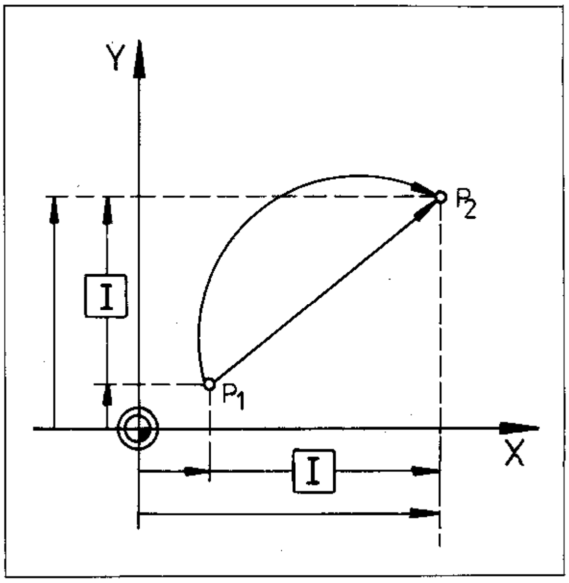
Co-ordinates can only be programmed when the **path** to the target position has already been specified.

The type of path is programmed with one of the **contouring keys** (see next page). These keys simultaneously initiate dialogue programming.

Absolute/ Incremental dimensions

If position co-ordinates are to be entered in **incremental dimensions,** the I -key must be pressed. The red indicator lamp signals that the entry has been transferred as an incremental dimension.

The I -key has an alternating function. By re-pressing the I -key, programming is reverted to **absolute dimensions** and the red indicator lamp is then off.



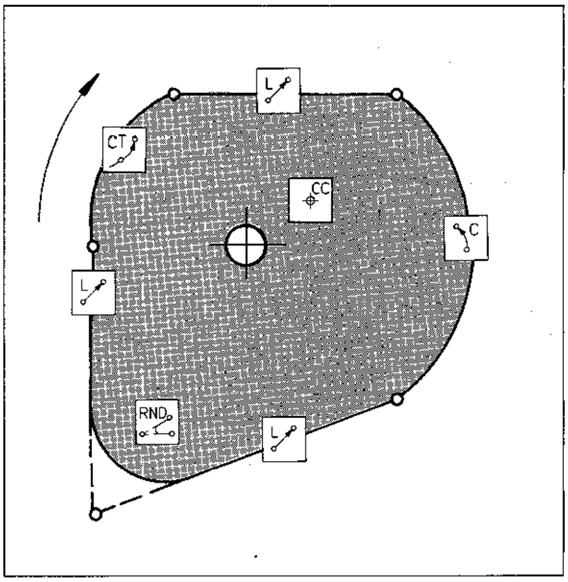
Programming of workpiece contours
Contouring keys/Cartesian co-ordinates

Contouring keys

Linear interpolation L ("Line"):
The tool follows a straight path. The end position of the straight line is programmed.

Circular interpolation C ("Circle"):
The tool follows the path of a circular arc. The end position of the circular arc is programmed.

Circle centre CC ("Circle Centre") (also as pole for polar co-ordinate programming):
For programming the circle centrepoint with circular interpolation and the pole-position for program entry in polar co-ordinates.

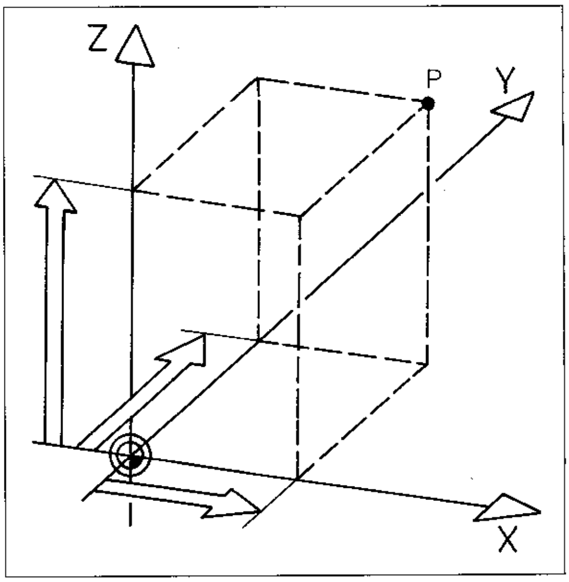


Rounding of corners RND:
The tool inserts an arc which has a tangential transition into the subsequent contour. Only the arc radius has to be programmed.

Tangential arc CT:
The tool inserts an arc which tangentially adjoins the previous contour. Only the end position of the arc has to be programmed.

Cartesian co-ordinates

A maximum of three axes (with linear interpolation) with the corresponding numerical value can be programmed. If axis IV is to be used for a rotary table (A, B or C-axis), entry is made in ° (degrees).



Programming of workpiece contours
Cartesian co-ordinates

Entry of Cartesian co-ordinates

Dialogue question:

COORDINATES?	X	Select axis, e.g. X.
	I	Incremental-Absolute?
	□	Key-in numerical value.
	Y	Enter next co-ordinate, e.g. Y and if required the third co-ordinate (max. 3 axes).
When **all co-ordinates** are entered:	ENT	Enter into memory.

Programming of workpiece contours
Polar co-ordinates/Pole

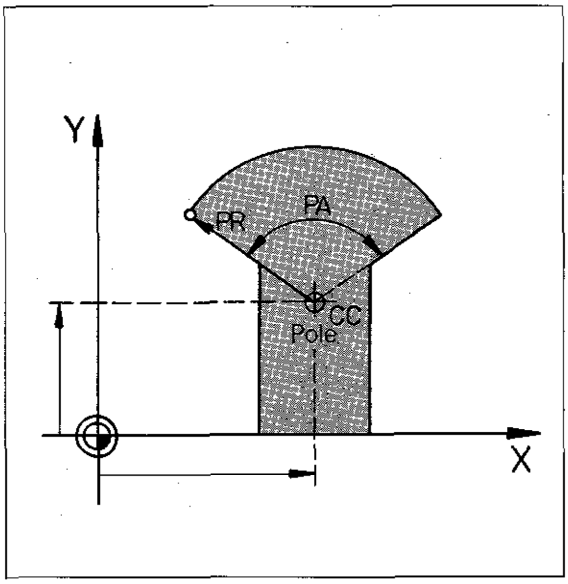
Pole CC

In the polar co-ordinates system, the datum for the polar co-ordinates is the pole.
Before entry of polar co-ordinates, the pole must be defined.

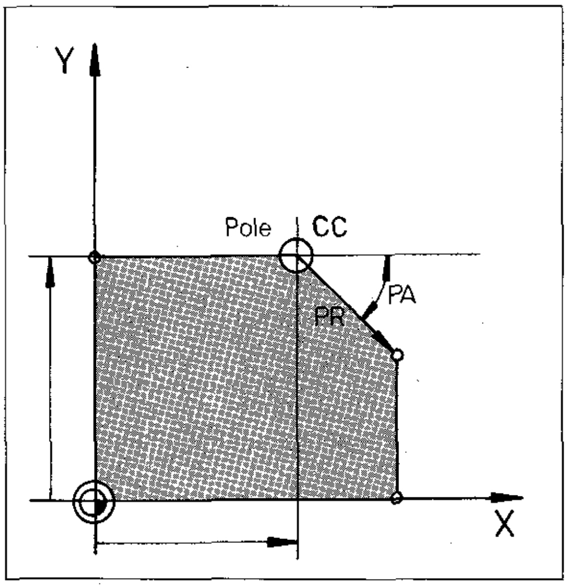
There are three ways of defining the pole:

- The pole is re-defined by using Cartesian co-ordinates.

  A CC-block is programmed with co-ordinates of the working plane.

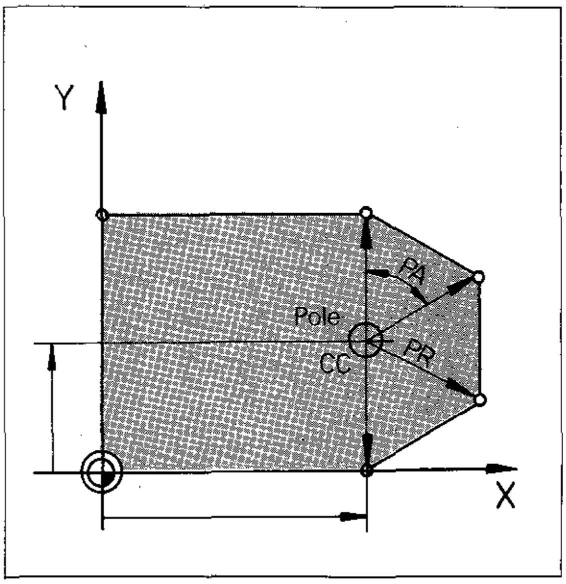


- The last nominal position is utilised as the pole.
  A CC-block is programmed. The co-ordinates last programmed are then used for the definition of the pole.



- The pole has the co-ordinates which were programmed in the last CC-block.

  A CC-block need not be programmed.



The pole can only be programmed in Cartesian co-ordinates.
CC in absolute dimensions: The pole is referenced to the workpiece datum.
CC in incremental dimensions: The pole is referenced to the previous nominal position of the tool.

Programming of workpiece contours
Polar co-ordinates/Pole

Entry of the pole

Operating mode ______________________ ◈

Dialogue initiation ______________________ CC

COORDINATES?	X	Select first axis, e.g. X.
	I	Incremental-Absolute?
	□	Key-in numerical value.
If only one co-ordinate of the last nominal value is to change, the other does not have to be entered.	Y	Select second axis, e.g. Y.
	I	Incremental-Absolute?
	□	Key-in numerical value.
	ENT	Enter into memory.

If the previous nominal position value is to be used as the pole, press NO ENT or END □.

Display example 1

```
27 CC X+10,000 IV+45,000
```

The pole has the absolute X-co-ordinate 10 and the incremental Y-co-ordinate 45.

Display example 2

```
92 L   X+20,500 Y+33,000

               R  F     M

93 CC
```

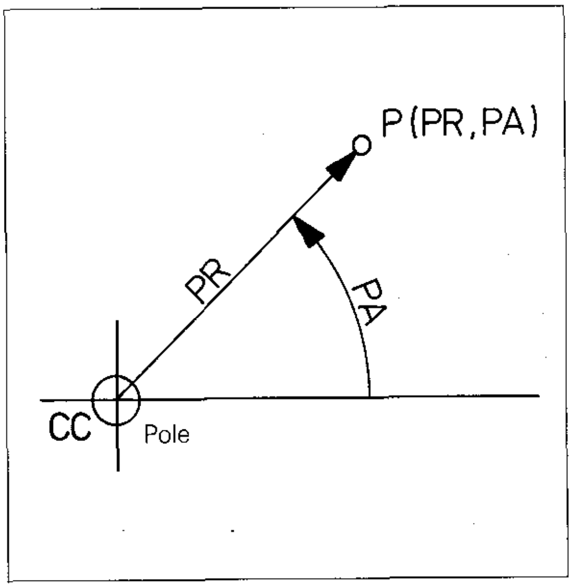
The pole in block 93 has the co-ordinates X 20.500 and Y 33.000.

Programming of workpiece contours
Polar co-ordinates

Polar co-ordinates

If required, polar co-ordinates can be used for programming positions (polar co-ordinate radius PR, polar co-ordinate angle PA).

Polar co-ordinates are always related to a **pole CC.**

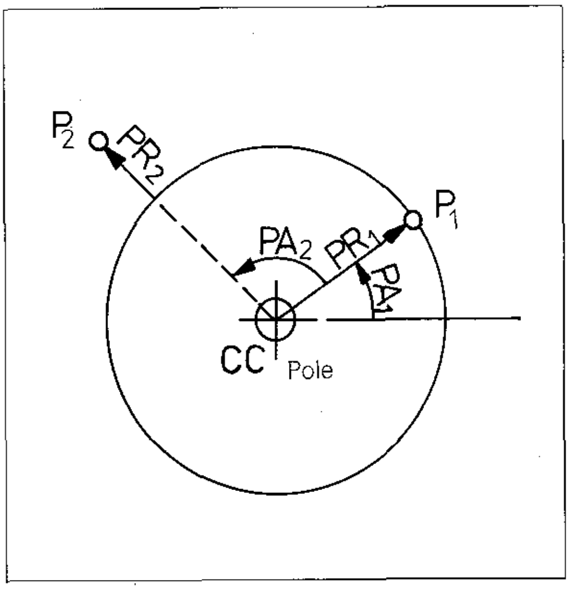


Incremental entry

With incremental entry, the polar co-ordinate radius is increased by the programmed value.

An incremental polar co-ordinate angle is referenced to the angle last entered.

Example: Point P1 has the polar co-ordinates PR1 (absolute) and PA1 (absolute). Point P2 has the polar co-cordinates PR2 (incremental) and PA2 (incremental). When programming point PR2, only the **change in radius** and **change in angle** for PA2 are entered as numerical values. Point P2 has the absolute values PR = (PR1 + PR2) and PA = (PA1 + PA2).



Programming of workpiece contours
Polar co-ordinates

Entry of polar co-ordinates

Dialogue question:

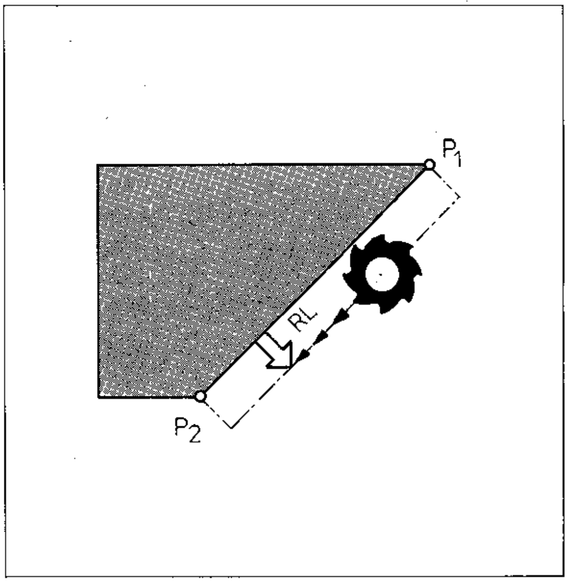
POLAR COORDINATES-RADIUS PR?	I	Incremental-Absolute?
	□	Key-in polar co-ordinates radius PR to target point.
	ENT	Enter into memory.

POLAR COORDINATES-ANGLE PA?	I	Incremental-Absolute?
	□	Key-in polar co-ordinates angle PA related to reference axis.
	ENT	Enter into memory.

Programming of workpiece contours
Radius compensation – Path compensation

Tool radius compensation

For automatic compensation of tool length and radius – as entered in the TOOL DEF block – the control must know whether the tool is located to the right of the contour, left of the contour or is directly on the contour in the feed direction.



Path compensation

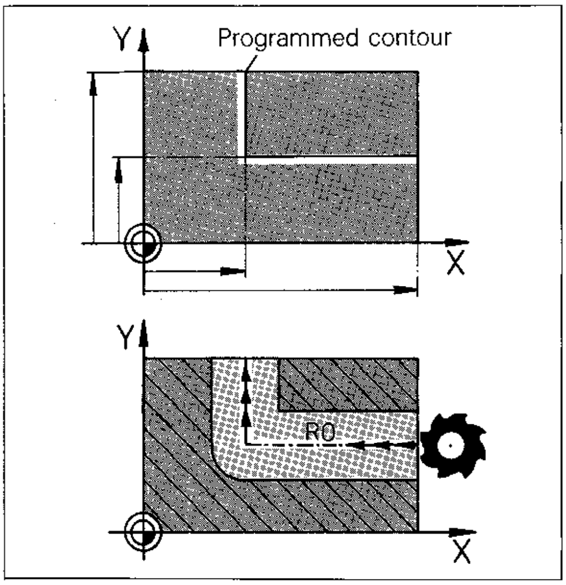
If the tool is moving with path compensation, i.e. the centrepoint of the tool is moving with the programmed radius being considered, the tool follows a path which is parallel to the workpiece contour and which is offset by the tool radius.

Programming the radius offset

Tool radius offset is programmed by pressing the keys $\boxed{R^{L}_{-}}$ and $\boxed{R^{R}_{+}}$. The red indicator lamp shows which type of tool radius compensation is being applied.

R0

If the tool is to move along the contour without consideration of a radius offset, the positioning block must be programmed without tool radius compensation.



RR

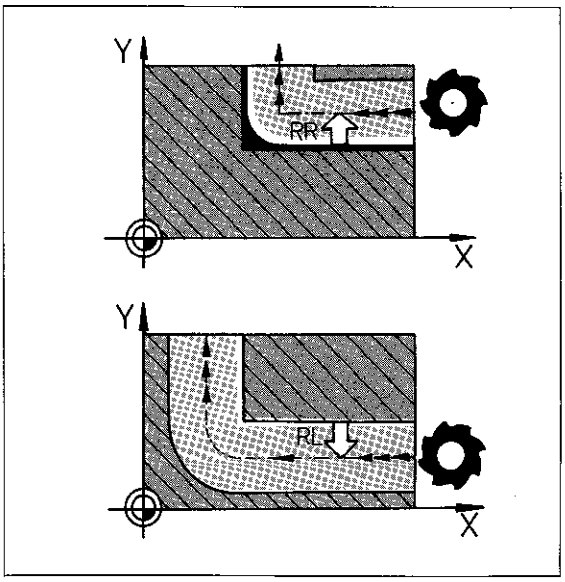
If the tool is to move on the **right-hand side** of the programmed contour with radius offset, press $\boxed{R^{R}_{+}}$.

The red indicator lamp signals that the $\boxed{R^{R}_{+}}$-function is effective.

RL

If the tool is to move on the **left-hand side** of the programmed **contour** with radius offset, press $\boxed{R^{L}_{-}}$.

The red indicator lamp signals that the $\boxed{R^{L}_{-}}$-function is effective.



Programming of workpiece contours
Radius compensation

Entry of RL or RR

Dialogue question:

TOOL RADIUS COMP. RL/RR/NO COMP.? ▶	RL RR	select radius compensation.
	▼ ENT	Enter into memory.

Entry of R0

Red indicator lamps below RR and RL **must be off.**

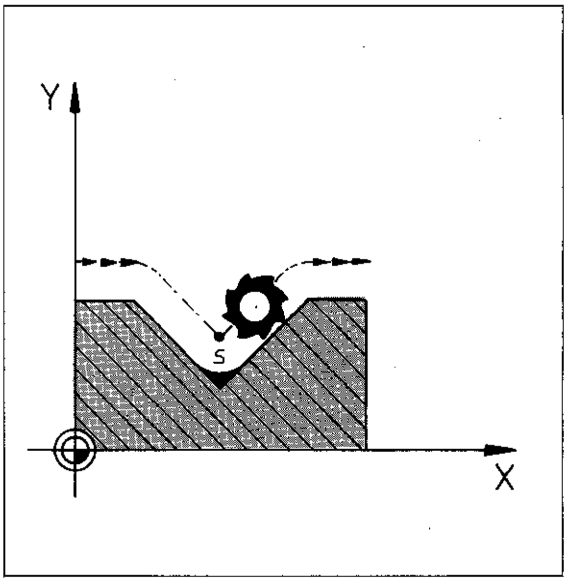
Dialogue question:

TOOL RADIUS COMP. RL/RR/NO COMP.? ▶	ENT	Enter "no compensation" into memory.

Programming of workpiece contours
Path compensation

Path compensation on internal corners

On **internal corners,** the control automatically calculates the **intersection S** of the milling tool path which is parallel to the workpiece contour. This prevents workpiece damage through back cutting.

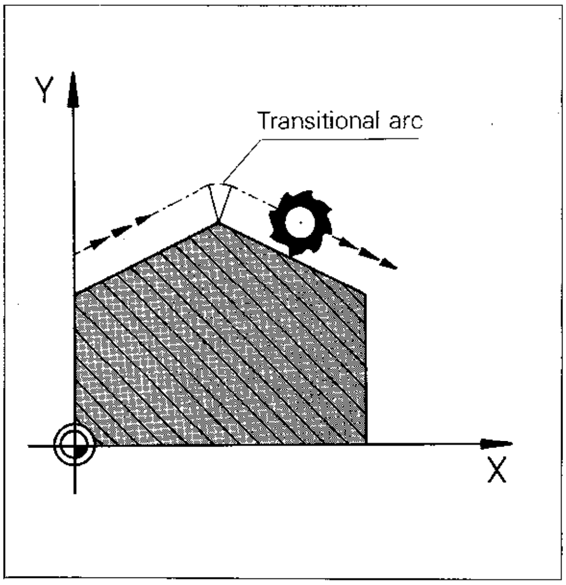


Path compensation on external corners

When radius compensation has been programmed, the control applies a **transitional arc** which enables the tool to "roll" around the corner.

In most cases, the tool is guided around the corner at a constant feed rate. If, however, the programmed feed rate is too high for the transitional arc, the feed rate is automatically reduced to a lower value (ensuring contour precision). The limit value is permanently programmed within the control.

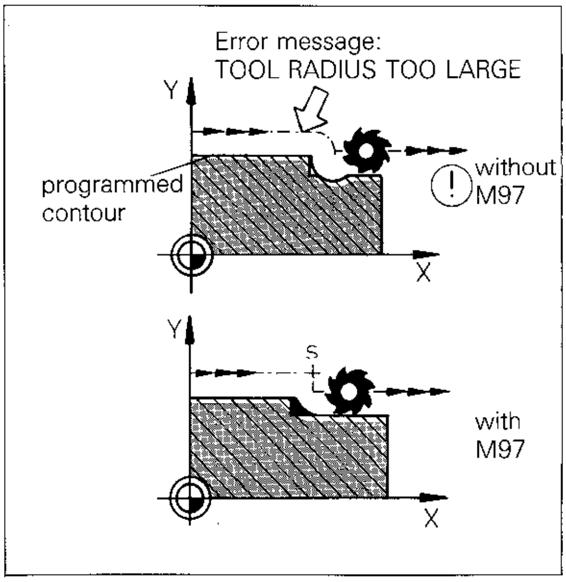
Automatic feed rate reduction can be cancelled by programming the auxiliary function M90 (see "Feed rate") if required.



Correction of path intersection with M97

If the tool radius is larger than a step within the contour, the transitional arc can cause workpiece damage on an external corner. This is then indicated by the error message = TOOL RADIUS TOO LARGE = and the corresponding positioning block is not executed.

The auxiliary function **M97** prevents the insertion of a transitional arc. The control then calculates a further **path intersection S** and guides the tool via this point, thereby preventing damage to the contour.



Path intersection correction via M97 is effective blockwise. It should be programmed into the block in which the external corner position has been programmed.

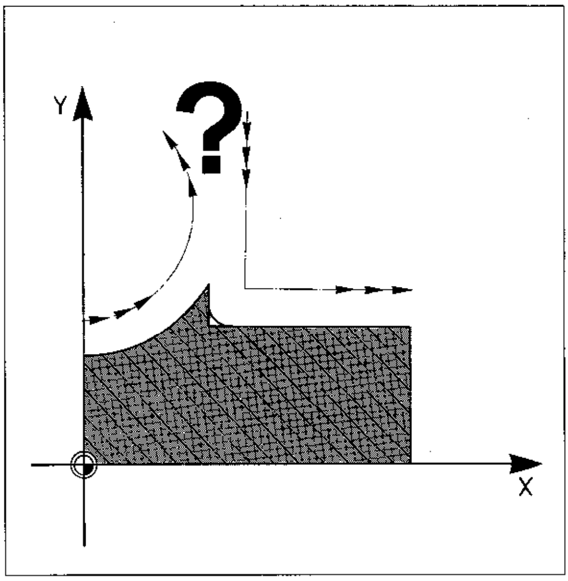
Special case with M97

In special cases, e.g. intersection of a circle and straight line, the control is unable to make an intersection with path compensation using M97.

When executing the program, the error

= TOOL RADIUS TOO LARGE =

is displayed.



Remedy

Insertion of an auxiliary positioning block which extends the end point of the arc by a length "zero". The control then performs a linear interpolation which determines the intersecting point S.

Example

16 CC Circle centrepoint
17 C Arc end position

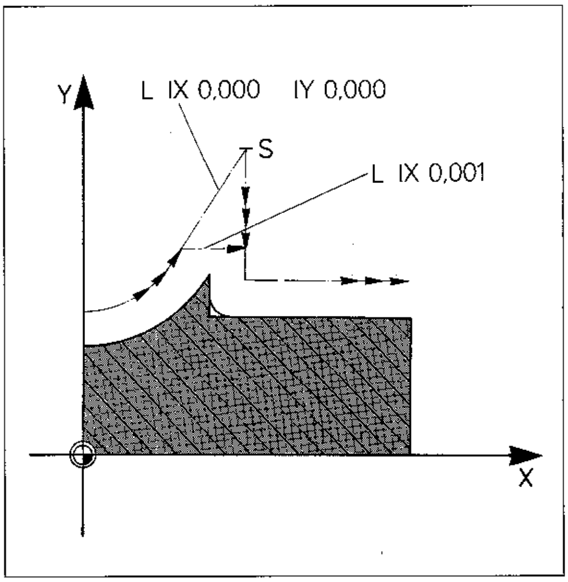
**18 L IX 0,000 IY 0,000
R F M97**

19 L straight

A straight contour element with the length zero has been programmed in block 18
or:

**18 L IX 0,001
R F M97**

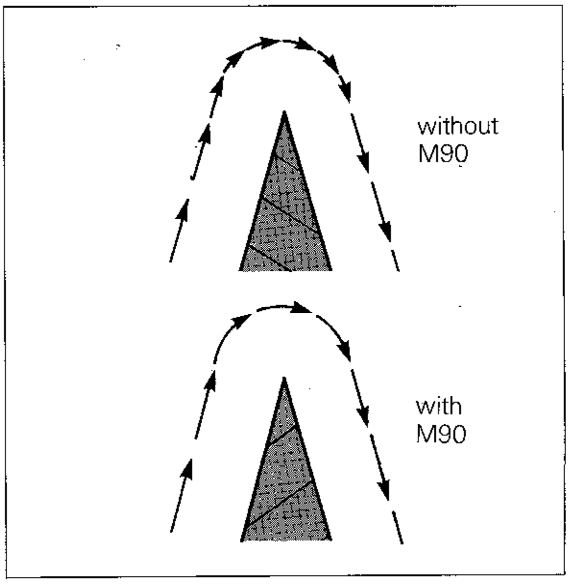
A straight contour has been programmed with a length of 0.001 mm.



Constant feed rate on external corners M90

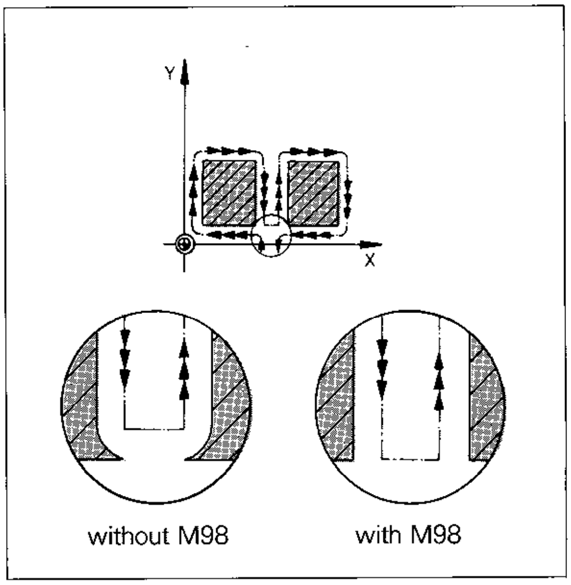
The feed rate reduction on external corners can be cancelled with the auxiliary function M90. This can however lead to a slight contour blemish. Also, excessive acceleration values can occur, i.e. the maximum acceleration defined in the machine parameters can be exceeded.

This auxiliary function depends on the machine parameters which are stored in the memory (operation with trailing error). The machine tool builder will indicate if this type of operation is possible with your control.



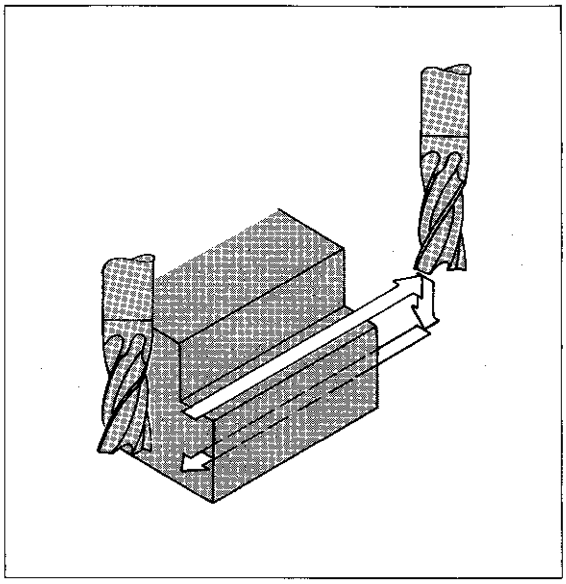
Termination of path compensation M98

The auxiliary function M98 ensures that a contour element is completely executed. If a further contour has been programmed, as shown in the adjacent example, the first contour position is approached with tool radius compensation, as a result of M98, and is completely executed (see also "Departure command").



Line-by-line milling with M98

A further example for application of M98 is line-by-line milling with downfeed in Z.



Example

```
LZ -10     R F9999      M
LX X20     Y-10 RR F20  M
L  Y110    R F          M98
LZ -20     R F9999      M
L  Y-110   RL F20       M
L  Y-10    R F          M98
```

Remarks

Programming of workpiece contours
Feed rate F/Auxiliary functions M

Feed rate

The feed rate, i.e. tool path speed is programmed in mm/min. or 0.1 inch/min.

With rotary tables (A, B or C-axis) the entry value is in °/min.

The **feed rate override** on the control operating panel can vary the feed rate from 0 to 150%.

Max. entry values (rapid) for the feed rate are

- 15999 mm/min. or
- 6299/10 inch/min.

The max. feed rate of the individual machine axes is determined through machine parameters by the machine tool builder.

For control models with software version 08: The current feed rate is indicated in the status display, below right, of the VDU-display. If this display is shown in inverted characters (light background) and the axes are no longer moving, it signifies that the feed rate has not been released by the control interface. In the event of this condition, please contact the machine tool manufacturer.

F

Auxiliary functions

For control of special machine functions (e.g. spindle "on") and tool path behaviour, auxiliary (miscellaneous) functions can be programmed. Auxiliary functions have the **address letter M** and a **code number.**

When programming, it must be noted that certain M-functions are effective at the beginning of a block (e.g. M03 spindle "on", clockwise) and others at the block-end (e.g. M05: Spindle "stop").

A list of all M-functions is given on the following pages.

M

Programming of workpiece contours
Entry of feed rate
Entry of auxiliary functions

Entry of feed rate

Dialogue question:

FEED RATE ? F =	▶ □	Key-in code number.
	▼ ENT	Enter into memory.

Entry of an auxiliary function

Dialogue question:

AUXILIARY FUNCTION M ?	▶ □	Key-in code number.
	▼ ENT	Enter into memory.

Auxiliary functions M

M-functions which affect program run

M	Function	Effective at	
		block beginning	block end
M00	Program run stop Spindle stop Coolant off		●
M02	Program run stop Spindle stop Coolant off Return jump to block 1		●
M03	Spindle on, clockwise	●	
M04	Spindle on, counter-clockwise	●	
M05	Spindle stop		●
M06	Tool change Program run stop if reqd. (depends on machine parameters entered) Spindle stop Coolant off		●
M08	Coolant on	●	
M09	Coolant off		●
M13	Spindle on, clockwise Coolant on	●	
M14	Spindle on, counter-clockwise Coolant on	●	
M30	As per M02		●
M89	Free auxiliary function or Cycle call, modally effective (depending on the machine parameters entered)	●	●
M90	Constant path feed rate on corners (see "Feed rate")	●	
M91	**Within a positioning block:** Workpiece zero datum is replaced by reference point	●	
M92	**Within a positioning block:** The set workpiece zero datum is replaced by a position which is defined by the machine tool builder using a machine parameter. (e.g. tool change position).	●	
M93	The assignment of this M-function is reserved by HEIDENHAIN	●	
M94	Rotary table axis display reduction to a value below 360°	●	
M95	Change approach behaviour (see "Approach command M95")		●
M96	Change approach behaviour (see "Approach command M96")		●
M97	Compensation of path intersection on external corners		●
M98	End compensation of path intersection		●
M99	Cycle call		●

Auxiliary functions M

Freely selectable auxiliary functions

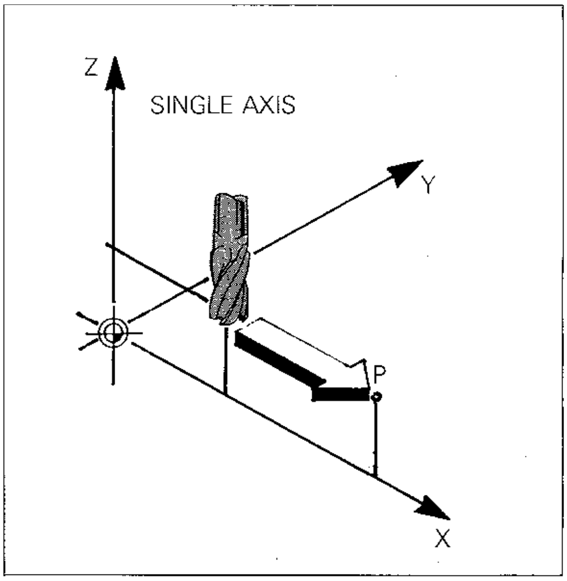
Freely selectable auxiliary functions are determined by the machine tool builder and are explained in the machine tool manual.

M	Function	Effective at	
		block beginning	block end
M01			●
M07		●	
M10			●
M11		●	
M12			●
M15		●	
M16		●	
M17		●	
M18		●	
M19		●	
M20		●	
M21		●	
M22		●	
M23		●	
M24		●	
M25		●	
M26		●	
M27		●	
M28		●	
M29		●	
M31		●	
M32			●
M33			●
M34			●
M35			●
M36		●	
M37		●	
M38		●	
M39		●	
M40		●	
M41		●	
M42		●	
M43		●	
M44		●	
M45		●	
M46		●	
M47		●	
M48		●	
M49		●	
M50		●	
M51		●	
M52			●
M53			●
M54			●
M55		●	
M56		●	
M57		●	
M58		●	
M59		●	
M60			●
M61		●	
M62		●	
M63			●
M64			●
M65			●
M66			●
M67			●
M68			●
M69			●
M70			●
M71		●	
M72		●	
M73		●	
M74		●	
M75		●	
M76		●	
M77		●	
M78		●	
M79		●	
M80		●	
M81		●	
M82		●	
M83		●	
M84		●	
M85		●	
M86		●	
M87		●	
M88		●	

Programming of workpiece contours
Straight paths

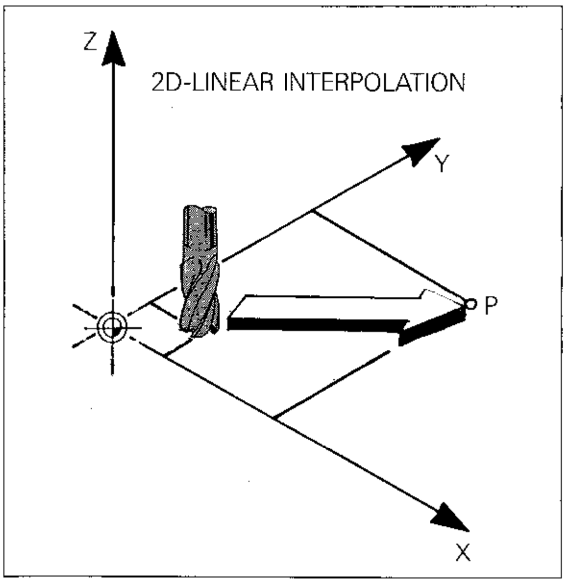
Single axis movements

If the tool moves relative to the workpiece in a straight path which is parallel to a **machine axis,** this is referred to as **single axis** positioning or machining.



2D-Linear interpolation

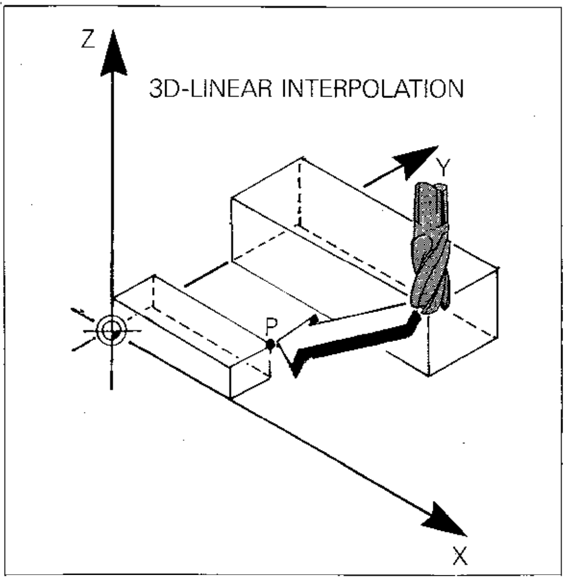
If the tool moves in a straight path in one of the **main planes** (XY, YZ, ZX), this is referred to as **2D-linear interpolation.**



3D-Linear interpolation

If the tool moves relative to the workpiece in a straight path with simultaneous traversing of **all three machine axes,** this is referred to as **3D-linear interpolation.**

Simultaneous traversing of three machine axes in a straight path is not possible with the control versions TNC 151 E/TNC 155 E/TNC 151 V/TNC 155 V.



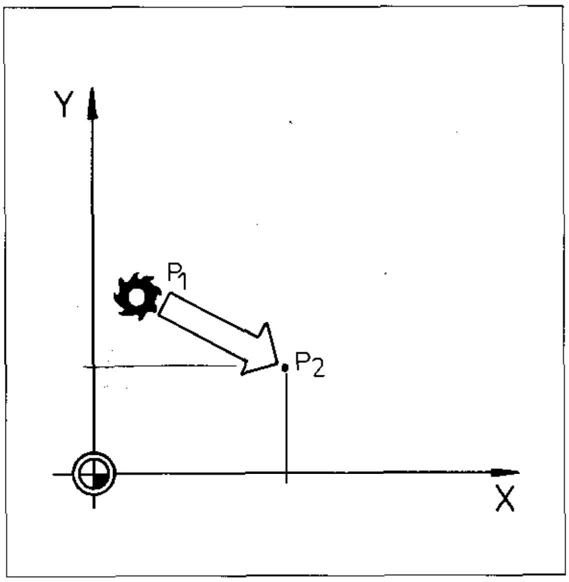
Programming of workpiece contours
Straight paths

Straight line L

The tool is to move in a straight line from the starting position P1 to the target position P2.

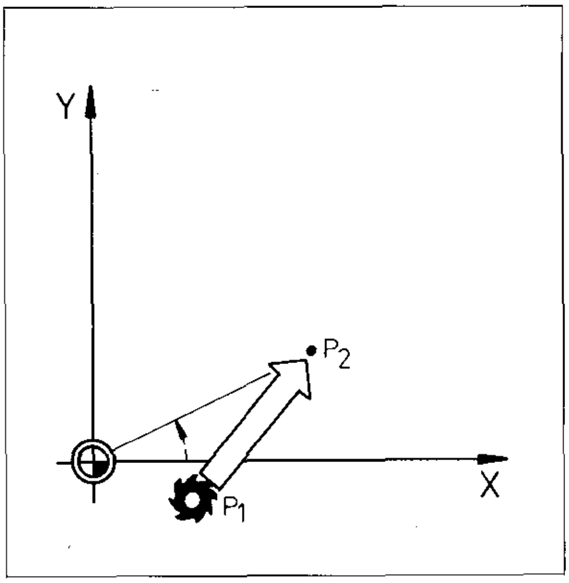
The target position P2 (nominal position) is programmed.

The nominal position P2 can be specified either in Cartesian or in polar co-ordinates.



Linear interpolation with a linear axis and angle axis

When performing linear interpolation with a linear and an angle axis, the following should be noted:
Software version 01, 02 (TNC 155)
The programmed feed rate applies to the speed of the angle axis. With rotary axis movements through small angles, the linear axis must adapt its feed rate to the rotary axis. This leads to relatively high feed rates of the linear axis and – since the feed rate of the linear axis is displayed – a correspondingly high feed rate display on the VDU-screen.
As of software version 03
(TNC 151/TNC 155)
The programmed feed rate F is interpreted as a contouring feed rate, i.e. broken down into linear and angle components as follows:



$$F(L) = \frac{F \times \Delta L}{\sqrt{(\Delta L)^2 + (\Delta W)^2}}$$

$$F(W) = \frac{F \times \Delta W}{\sqrt{(\Delta L)^2 + (\Delta W)^2}}$$

Designation:
F = programmed feed rate
F (L) = linear component of feed rate
F (W) = angle component
ΔL = Traversing distance of linear axis
ΔW = Traversing distance of angle axis

Remarks

Programming of workpiece contours
Linear interpolation/Cartesian co-ordinates

Red indicator lamp beneath P-key (for polar co-ordinates) must be off! Press P-key if necessary.

Entry in Cartesian co-ordinates

Operating mode ____________ ◈

Dialogue initiation ____________ L

COORDINATES?	X	Select axis, e.g. X.
	I	Incremental-Absolute?
	□	Key-in numerical value.
	Y	Key-in next co-ordinate, e.g. Y and if reqd. a third co-ordinate (max. 3 axes).
When keying-in of **all co-ordinates** of the target position is finalised.	ENT	Enter into memory.

TOOL RADIUS COMP. RL/RR/NO COMP.?	RL RR	If reqd., key-in radius compensation.
	ENT	Enter into memory.

FEED RATE? F =	□	If reqd., key-in feed rate.
	ENT	Enter into memory.

AUXILIARY FUNCTION M?	□	If reqd., key-in auxiliary function.
	ENT	Enter into memory.

If, after keying-in the co-ordinates, the remaining data are unchanged, the block can be curtailed by simply pressing END.

Display example

```
28 L X+20,000 IY+49,800

              RL F100 M13
```

The tool moves to position X 20.0 mm (absolute) and Y 49.8 mm (incremental) with a radius offset to the left of the contour and with a feed rate of 100 mm/min.

The coolant is switched on at the beginning and the spindle rotation is clockwise.

Remarks

Programming of workpiece contours
Linear interpolation/Polar co-ordinates

Red indicator lamp beneath P-key (for polar co-ordinates) must be on! Press P-key if necessary.

Entry in polar co-ordinates

Operating mode ____________ ⇒

Dialogue initiation ____________ (if reqd. P) L

POLAR COORDINATES RADIUS PR?	▶ I	Incremental – Absolute?
	▼ □	Key-in polar co-ordinates radius PR for end position of straight line.
	▼ ENT	Enter into memory.

▼

POLAR COORDINATES ANGLE PA?	▶ I	Incremental – Absolute?
	▼ □	Key-in polar co-ordinates angle PA for end position of straight line.
	▼ ENT	Enter into memory.

▼

TOOL RADIUS COMP. TL/RR/NO COMP.?	▶ RL RR	If reqd., key-in radius compensation.
	▼ ENT	Enter into memory.

▼

FEED RATE ? F =	▶ □	If reqd., key-in feed rate.
	▼ ENT	Enter into memory.

▼

AUXILIARY FUNCTION M ?	▶ □	If reqd., key-in auxiliary function.
	▼ ENT	Enter into memory.

If, after keying-in the co-ordinates, the remaining data are unchanged, the block can be curtailed by simply pressing END.

Display example

```
39 LP PR + 35,000 PA + 45,000
                R    F    M
```

The tool moves to a position which is 35 mm away from the last defined Pole CC; the polar co-ordinates angle is 45° (absolute). Radius compensation and feed rate are determined by the values last programmed. There is no auxiliary function.

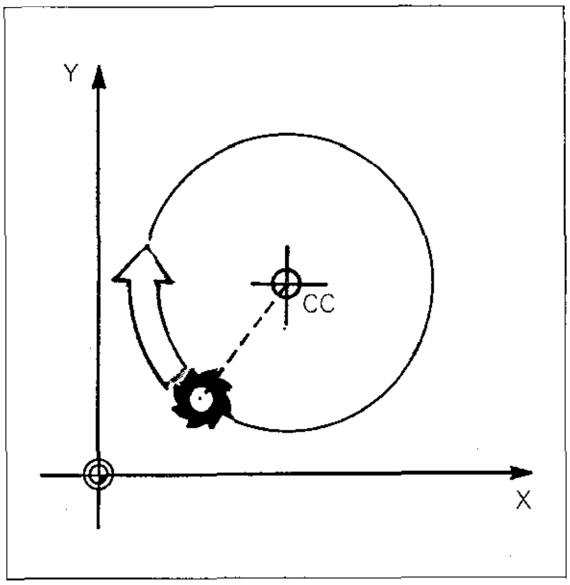
Programming of workpiece contours
Circular interpolation

Circular interpolation

The movements of two axes are simultaneously controlled such, that the relative movement of the tool to the workpiece describes a circle or an arc.

With TNC 155 an arc can be programmed in three ways:

- via the circle centrepoint and end position with the keys CC and C
- by inserting an arc with a tangential transition at both ends, via the radius only, with the RND-key.
- by adjoining the arc to the previous contour tangentially and the arc end position with the CT-key



Circle centre CC

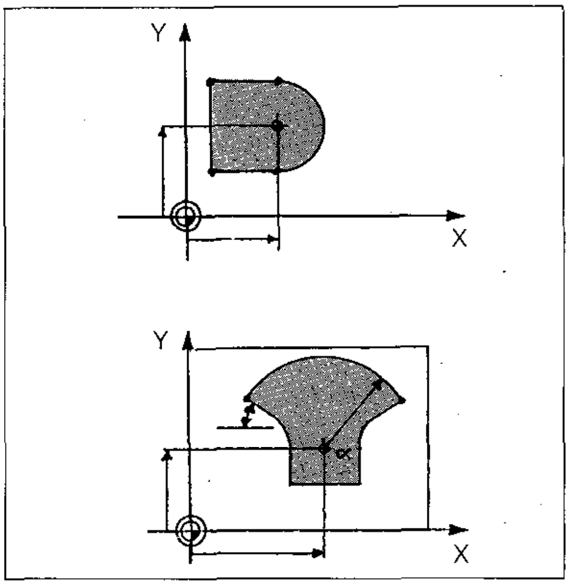
The circle centre must be defined before commencement of circular interpolation-programming with C.

Two types of programming are possible:

- The circle centre CC is defined with Cartesian co-ordinates.
- The circle centre is already defined by the co-ordinates of the last CC-block.

Entry dialogue for the circle centre is initiated with the CC-key (see "Pole").

CC in absolute dimensions: The circle centre is referenced to the workpiece datum.
CC in incremental dimensions: The circle centre is referenced to the previous nominal position of the tool.



Circular path C

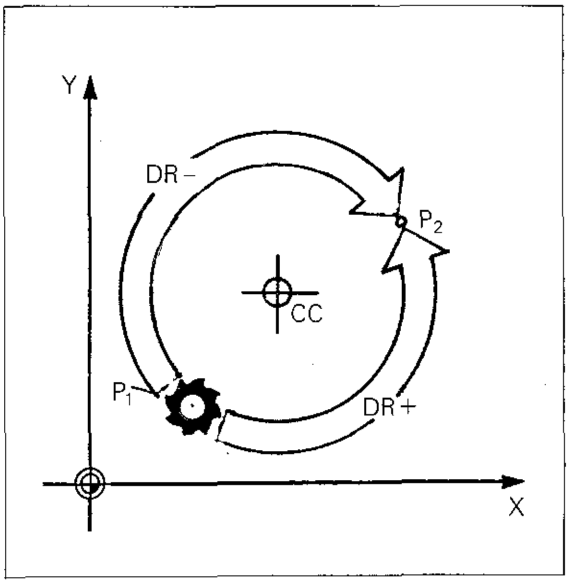
The tool is to move on a circular path from the actual position P1 to the target position P2. Only position P2 is programmed. Position P2 may be specified in Cartesian or polar co-ordinates.

Direction of rotation

For circular path movement, the control must know the **direction of rotation.** The rotation direction is either positive DR+ (counter-clockwise) or negative DR− (clockwise).

An amended contour must not be commenced with a circular path.

Error message:
PATH OFFSET WRONGLY STARTED



Programming of workpiece contours
Direction of rotation

Entry

Dialogue question:

ROTATION CLOCKWISE: DR – ?

If rotation should be clockwise:	▶ +/−	Key-in (–) rotating direction.
	▼ ENT	Enter into memory.
If rotation should be counter-clockwise:	▶ +/−	Key-in (+) rotating direction (press sign change key twice).
	▼ +/−	
	▼ ENT	Enter into memory.

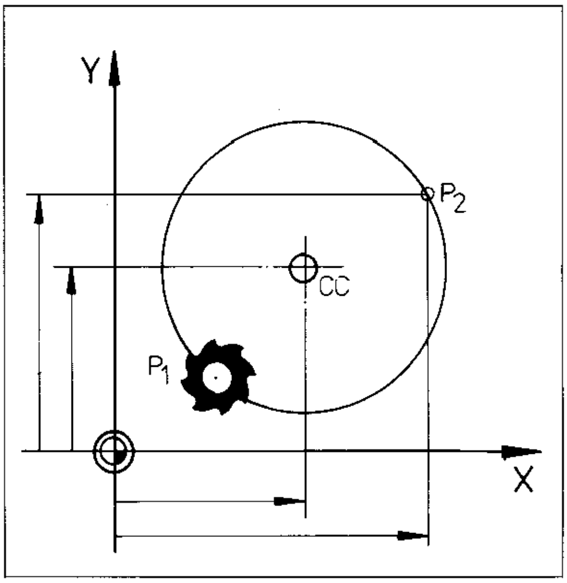
Programming of workpiece contours
Circular interpolation/Cartesian co-ordinates

Circular path programming in Cartesian co-ordinates

When programming in Cartesian co-ordinates care must be taken that the starting position and target position (new nominal position) both lie on the same circular path, i.e. both positions must have the same distance to the circle centre CC.

If this is not the case, the following error is displayed:

= CIRCLE END POS. INCORRECT =



Programming of workpiece contours
Circular interpolation/Cartesian co-ordinates

Red indicator lamp beneath P-key (for polar co-ordinates) must be off! Press P-key if necessary.

Entry in Cartesian co-ordinates

Operating mode

Dialogue initiation

COORDINATES?	X	Select axis, e.g. X.
	I	Incremental – Absolute ?
	□	Key-in numerical value.
	Y	Key-in next co-ordinate, e.g. Y.
When **all co-ordinates** of the arc end position are keyed-in:	ENT	Enter into memory.
ROTATION CLOCKWISE: DR – ?	+/–	Key-in rotating direction.
	ENT	Enter into memory.
TOOL RADIUS COMP. RL/RR/NO COMP. ?	RL RR	If reqd., key-in radius compensation.
	ENT	Enter into memory.
FEED RATE ? F =	□	If reqd., key-in feed rate.
	ENT	Enter into memory.
AUXILIARY FUNCTION M ?	□	If reqd., key-in auxiliary function.
	ENT	Enter into memory.

Display example

```
87 C X + 30,000 Y + 48,000
          DR+ RR F      M
```

The tool moves to the target position X 30.000 and Y 48.000 in a circular path in the positive rotating direction (counter-clockwise), with a tool radius offset to the right of the contour.

The feed rate corresponds to the value last programmed. There is no auxiliary function.

Programming of workpiece contours
Circular interpolation/Polar co-ordinates

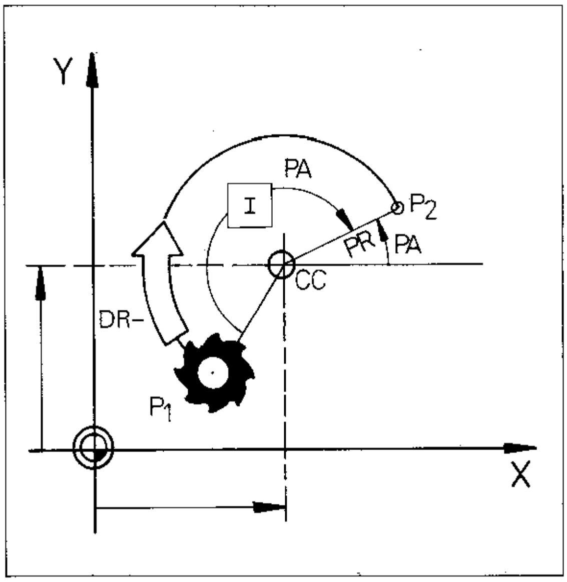
Circular path programming in polar co-ordinates

If the target position on the circular path is programmed in polar co-ordinates, it is sufficient if the target position is defined through specification of the polar co-ordinates angle PA (absolute or incremental).

The radius is already defined through the position of the tool and the programmed circle centre CC.

With circular arc programming in polar co-ordinates, the angle PA can be entered either as a positive or a negative value. The angle PA determines the end position of the arc. The traversing direction can also be programmed as a positive or a negative value. If the angle PA is specified as an incremental value, the signs of angle and the traversing direction should be indentical. In the adjacent example IPA and DR are both negative.

If the tool is located at the pole or circle centre before starting circular interpolation, the following error is displayed:
= ANGLE REFERENCE MISSING =



Programming of workpiece contours
Circular interpolation/Polar co-ordinates

Red indicator lamp beneath P-key (for polar co-ordinates) must be on! Press P-key if necessary.

Entry in polar co-ordinates

Operating mode ______ ◇

Dialogue initiation ______ (if reqd. P) C

POLAR COORDINATES ANGLE PA?	▶ I	Incremental – Absolute
	□	Key-in polar co-ordinates angle PA for circle target position.
	ENT	Enter into memory.

COORDINATES? The question COORDINATES requires an entry with "Helical interpolation" for explanation, see "Helical interpolation"	▶ NO ENT	

ROTATION CLOCKWISE: DR – ?	▶ +/−	Key-in rotating direction.
	ENT	Enter into memory.

TOOL RADIUS COMP. RL/RR/NO COMP. ?	▶ RL RR	If reqd., key-in radius compensation.
	ENT	Enter into memory.

FEED RATE ? F =	▶ □	If reqd., key-in feed rate.
	ENT	Enter into memory.

AUXILIARY FUNCTION M ?	▶ □	If reqd., key-in auxiliary function.
	ENT	Enter into memory.

Display example

```
17  CP  PA + 60,000

        DR – RL  F  M
```

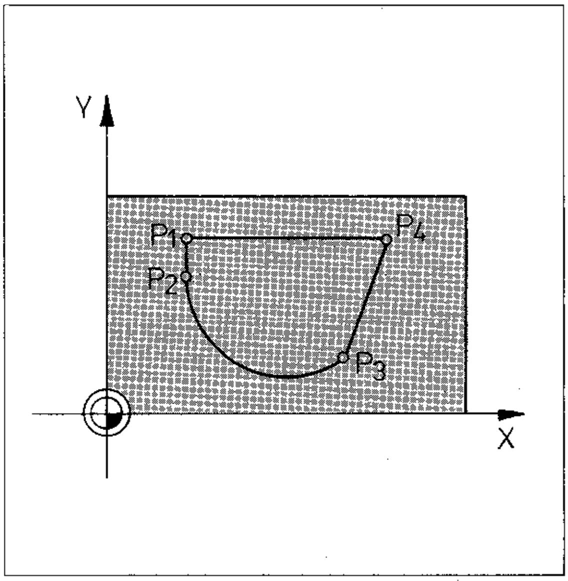
The tool moves in a circular path in the negative direction (clockwise), with a tool radius offset to the left of the programmed contour. The polar co-ordinates angle PA to the reference axis is 60°.

The feed rate corresponds to the value last programmed. There is no auxiliary function.

Programming of workpiece contours
Adjoining arcs

Arc with tangential connection

Programming of a circular path is simplified if the arc tangentially adjoins the contour. Only the **arc end position** is entered for defining the arc.



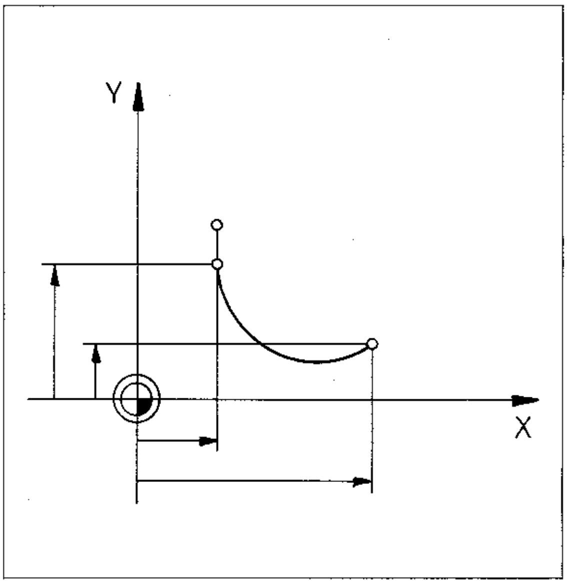
Provisions

The contour section, to which the circular path is to be adjoined, must be entered immediately before programming the adjoining arc. If the contour section is missing, the following error is displayed:

= CIRCLE END POS. INCORRECT =

Two co-ordinates must be programmed in the positioning block prior to the adjoining arc and within the block for the arc, otherwise the following error will be displayed:

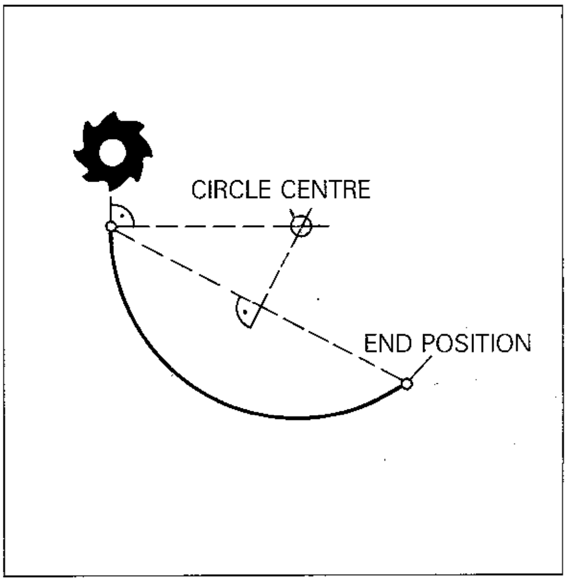
= ANGLE REFERENCE MISSING =



Geometry

With a tangential connection to the contour and an end position of the circular path, an arc **is defined exactly.**

This arc has a definite radius, a definite direction of rotation and a definite centrepoint. It is therefore unnecessary to program these items.



Entry

Only **Cartesian co-ordinates** may be programmed for the arc end position.

Dialogue is initiated by pressing CT.

Programming of workpiece contours
Adjoining arcs

Entry

Operating mode ______________________ ◈

Dialogue initiation ______________________ CT

COORDINATES?	X	Select axis, e.g. X.
	I	Incremental-Absolute?
	□	Key-in numerical value.
	Y	Key-in next co-ordinate, e.g. Y.
	I	Incremental-Absolute?
	□	Key-in numerical value.
	ENT	Enter into memory.

TOOL RADIUS COMP. RL/RR/NO COMP. ?	RL RR	If reqd., key-in radius compensation.
	ENT	Enter into memory.

FEED RATE ? F =	□	If reqd., key-in feed rate.
	ENT	Enter into memory.

AUXILIARY FUNCTION M ?	□	If reqd., key-in auxiliary function.
	ENT	Enter into memory.

☞ A full circle cannot be programmed.

Display example

```
20  CT  X + 15,800  Y + 35,000

          R    F              M
```

An arc has been tangentially adjoined. The co-ordinates of the arc end position are X 15.8/Y 35.0.

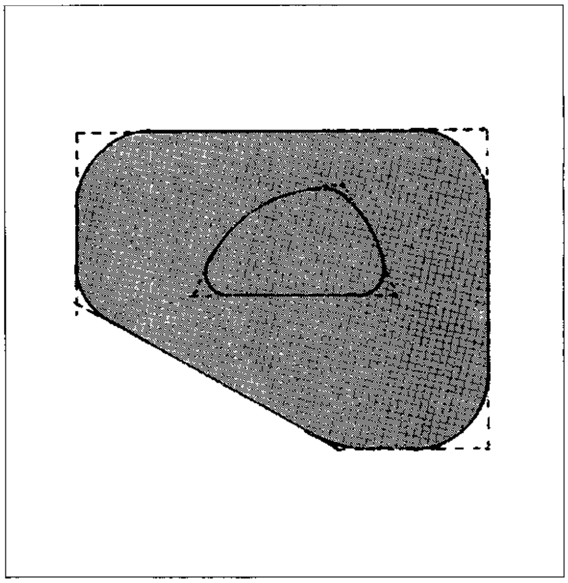
Programming of workpiece contours
Rounding of corners

Rounding of corners RND

Contour corners can be rounded-off by applying corner radii. The corner radius has a tangential transition into both the previous and subsequent contour section.

Insertion of a rounding-off radius is possible on all contour corners, i.e. corners can be formed by the following contour elements:

- Straight – Straight
- Straight – Arc of Arc-Straight
- Arc – Arc

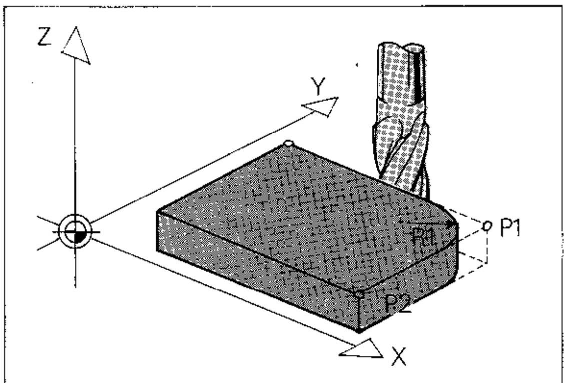


Programming hint

Application of a rounding-off radius can only be performed in a **main plane, XY, YZ or ZX.**

This means that the positioning blocks immediately before and after the "rounding-off" block must contain both co-ordinates of the working plane. If the working plane is not exactly defined (e.g. positioning blocks with X.. Y.. Z..), the following error is displayed:

= PLANE WRONGLY DEFINED =



15 Straight line to P1 (X, Y)

16 RND R 15,000

17 Straight line to P2 (X, Y)

Programming

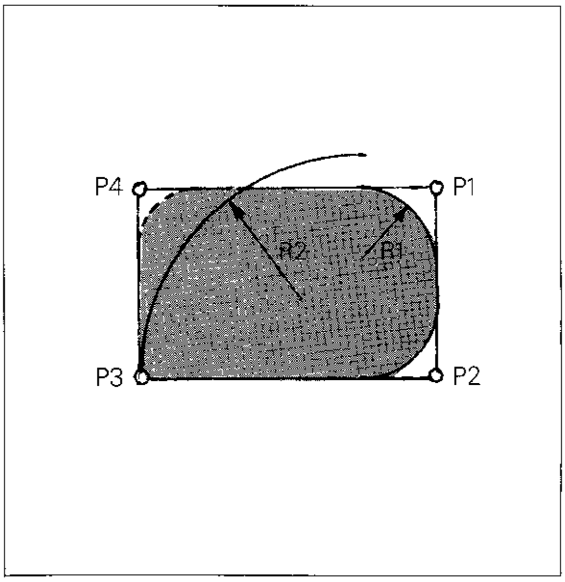
Programming of the rounding-off radius immediately follows the point P1 in which the corner is located.

The rounding-off radius is entered.

The rounding-off radius should not be too large. It should "fit between the contour elements".

If the selected radius is too large, the following error is displayed:

= ROUNDING RADIUS TOO LARGE =



Programming of workpiece contours
Rounding of corners

Before and after the rounding-off block, the contour elements must lie in the same working plane.

Entry

Operating mode ______________ ⇨

Dialogue initiation ______________ RND

ROUNDING-OFF RADIUS R?	▶ □	Key-in corner radius.
	▼ ENT	Enter into memory.

Display example

```
78  RND  R 5,000
```

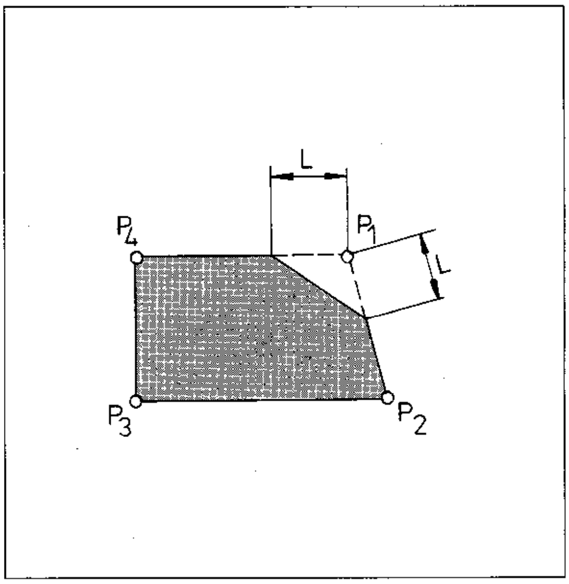
A rounding-off radius R = 5.000 mm has been inserted between the contour elements forming a corner.

Programming of workpiece contours
Chamfers

Chamfers

With TNC 151/TNC 155, chamfers with the side length L can be applied to workpieces. The L/ -key is used for programming.

The angle between points $\overline{P4P1}$ and $\overline{P1P2}$ is optional.

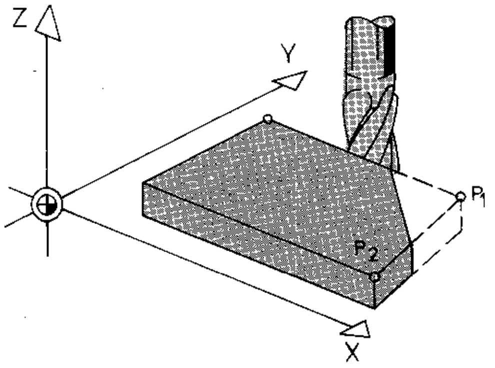


Program

Application of a chamfer may only be performed in **one of the main planes** (XY, YZ, ZX). This means that the blocks before and after the "chamfer-block" must contain both co-ordinates of the working plane.

If the working plane has not been exactly defined (e.g. a positioning block with X... Y... Z...), the following error is displayed:

= PLANE WRONGLY DEFINED =



26 Straight line to P1 (X, Y)

27 L 10,000

28 Straight line to P2 (X, Y)

Programming of workpiece contours
Chamfers

Entry

Operating mode ______ ⎆

Dialogue initiation ______ ⟋

COORDINATES ?	▶□	Key-in chamfer side length L.
	▼ ENT	Enter into memory.

Display example

88 L 7,500

A chamfer with the side length L = 7.5 mm has been applied between the contour elements forming a corner.

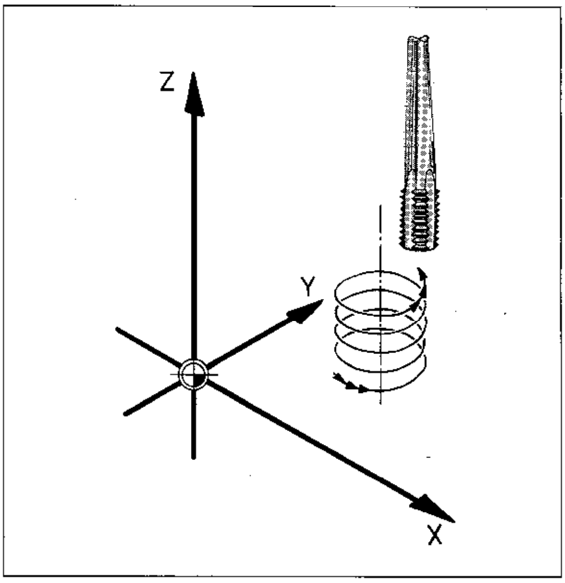
Programming of workpiece contours
Helical interpolation

Helix

With circular interpolation, two axes are simultaneously traversed such, that a circle is described in one of the main planes (XY, YZ, ZX).

If the circular interpolation is superimposed with a linear movement in the third axis (= tool axis), the tool will follow a helical path.

Helical interpolation can be used for manufacture of large-diameter, internal and external threads as well as lubrication grooves.



Helical interpolation is not possible with control versions TNC 151/TNC 155 E/TNC 151 V/ TNC 155 V.

Entry data

A helix can only be programmed in polar co-ordinates. As with circular interpolation, the **circle centre CC** must already be defined **beforehand.**

The total rotational angle of the tool (= number of thread turns Z) is entered as the **polar co-ordinates angle PA in degrees:**

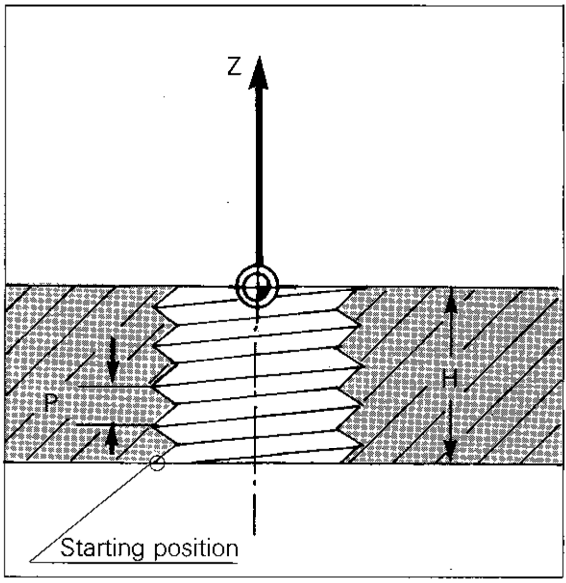
PA = Number of turns x 360°

For angles greater than 360°, PA must be specified incrementally. The total height/depth is entered in response to the dialogue request for **co-ordinates.**

This value depends on the required pitch.

H = P x A	H = Total height/depth
	P = Pitch
	A = Number of thread turns

The total height/depth can also be programmed as an absolute or incremental value.



Radius compensation

The tool radius compensation depends on
- the direction of rotation,
- the type of thread (internal/external)
- milling direction (pos./neg. axis direction):

Negative axis direction (– Z or – Y)			
Thread	Rotation direction	Radius compensation	
		internal	external
Left-hand thread	DR +	RL	RR
Right-hand thread	DR –	RR	RL

Positive axis direction (+ Z or + Y)			
Thread	Rotation direction	Radius compensation	
		internal	external
Left-hand thread	DR –	RR	RL
Right-hand thread	DR +	RL	RR

Programming of workpiece contours
Helical interpolation

Red indicator lamp beneath the P-key (for polar co-ordinates) must be on! Press P-key if necessary.

Entry

Operating mode ______ ◈

Dialogue initiation ______ (if reqd. P) ↰C

POLAR COORDINATES ANGLE PA ?	I	Incremental – Absolute ?
	□	Key-in total rotational angle.
COORDINATES ?	Z	Select feed axis.
	I	Incremental – Absolute ?
	□	Key-in height or depth.
	ENT	Enter into memory.
ROTATION CLOCKWISE: DR – ?	+/–	Key-in rotating direction.
	ENT	Enter into memory.
TOOL RADIUS COMP. RL/RR/NO COMP. ?	RL RR	Key-in radius compensation.
	ENT	Enter into memory.
FEED RATE ? F =	□	If reqd., key-in feed rate.
	ENT	Enter into memory.
AUXILIARY FUNCTION M ?	□	If reqd., key-in auxiliary function.
	ENT	Enter into memory.

Display example

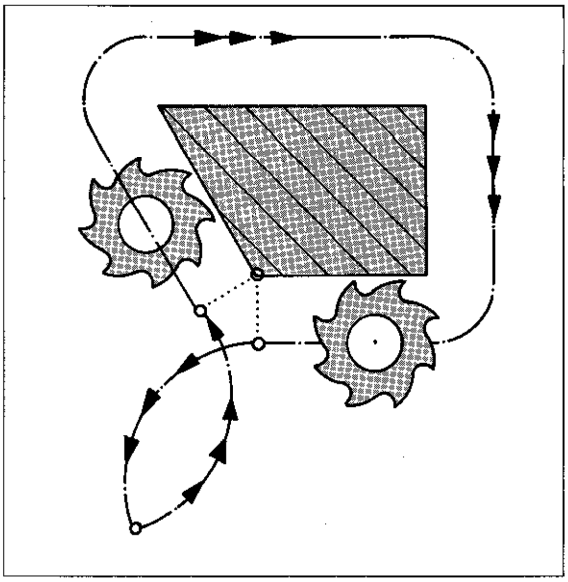
```
230  CP  IPA + 720,000  IZ + 6,000

                  DR+  RL  F  100  M
```

The tool follows a helical path in a counter-clockwise direction and completes two full turns. The total height is 6 mm, therefore resulting in a pitch of 3 mm. Tool radius offset is to the left of the contour, which means that an internal thread is being machined.

Contour approach and departure on an arc

Approach and departure on arc

Contour approach and departure on an arc has the advantage of the contour being approached to and departed from on a tangential "smooth" path. Programming for smooth tangential approach and departure is performed with RND!



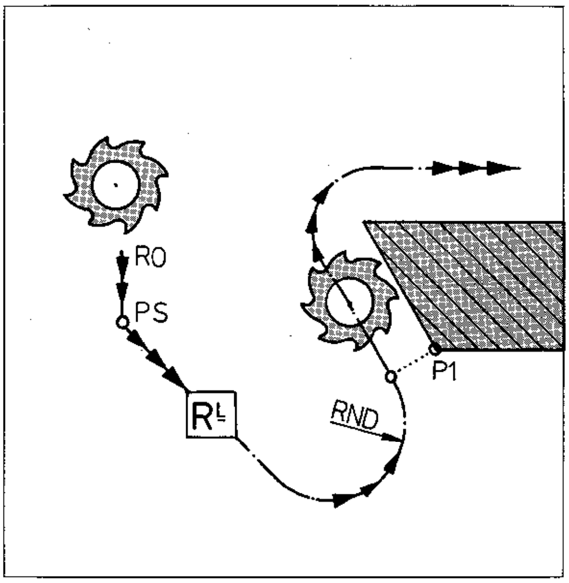
Approach (run-on)

The tool moves to the starting position PS and then towards the contour which is to be machined.

The positioning block to PS must not contain path compensation (i.e. R0).

The positioning block to the first contour position P1 contains path compensation (RR or RL).

The control recognizes that a **tangential** run-on procedure is required, since an RND-block follows the positioning block for contour position P.



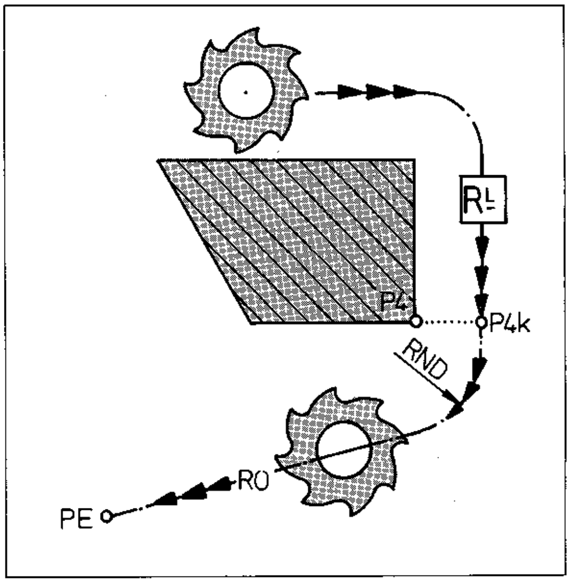
Departure (run-off)

The tool has reached the last contour position P and then proceeds to the finishing position PE.

The positioning block to P contains path compensation (RR or RL).

The position block to PE must not contain path compensation (i.e. R0).

The control recognizes that a **tangential** run-off procedure is required, since an RND-block follows the positioning block for the contour position P.



Contour approach and departure on an arc

Programming for approach (run-on)

20 L X + 100,000 Y + 50,000	Positioning block to starting position PS with **R0.**
R0 F 15999 M	
21 L X + 65,000 Y + 40,000	Positioning block to first contour position P1 with path compensation **RR.**
RR F 50 M13	
22 RND R 10,000	Specification of **tangential run-on radius.**
23 L X + 65,000 Y + 100,000	Positioning block to next contour position P2.
R F M	

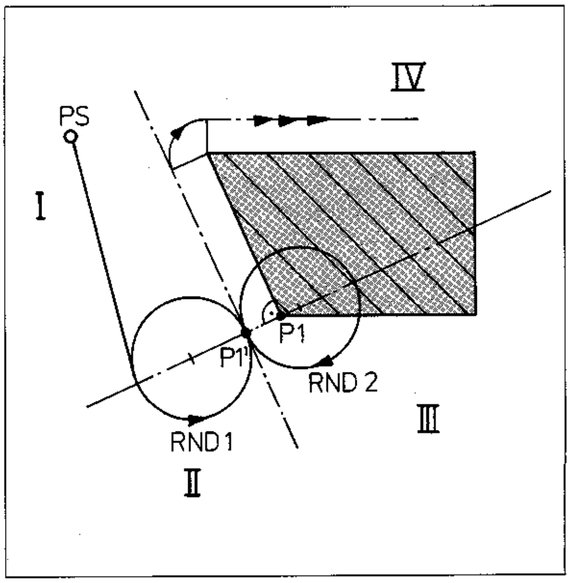
Programming for departure (run-off)

30 L X + 50,000 Y + 65,000	Positioning block to last contour position P with path compensation **RR.**
RR F 50 M	
31 RND R 15,000	Specification of **tangential run-off radius.**
32 L X + 100,000 Y + 85,000	Positioning block to finishing position PE with **R0.**
R0 F 15999 M00	Caution, when entering F15999.

A positioning block with both co-ordinates of the working plane must be programmed before and after the RND-block.

For tangential approach: The starting point PS must be located within the quadrant I,. II or III. The quadrants are formed by the starting direction in P1' and the its perpendicular (tangential direction with arcs) also passing through P1'. If the starting direction is located within quadrant IV, a clockwise arc will be formed thus damaging the workpiece.

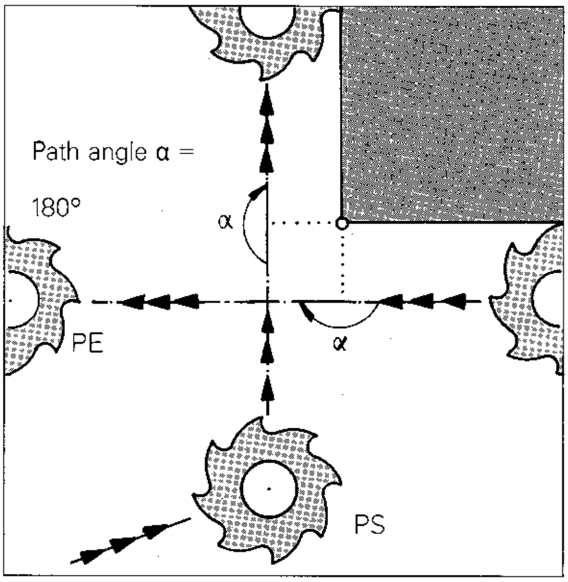
P1	= First contour position
P1'	= First compensated contour position
PS	= Starting position (with radius R0)
RND1	= Rounding-off arc for quadrants I, II
RND2	= Rounding-off arc for quadrants III, IV



Contour approach and departure in a straight path
Introduction

Contour approach and departure in a straight path

The tool is to move to the position PS and then run-on to the contour. After the machining procedure, the tool is to run-off the contour and proceed to the position PE.



Path angle α

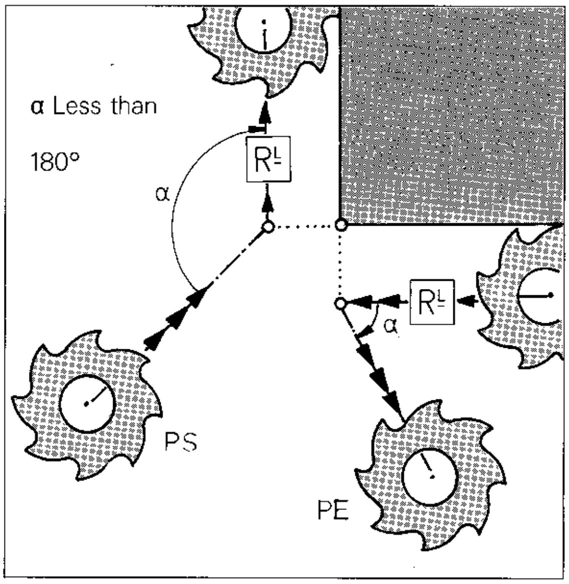
Run-on and run-off behaviour depends on the path angle **α**. This angle is related to the angle which is formed between

- the approach-straight and the first contour element and
- the departure-straight and the last contour element.

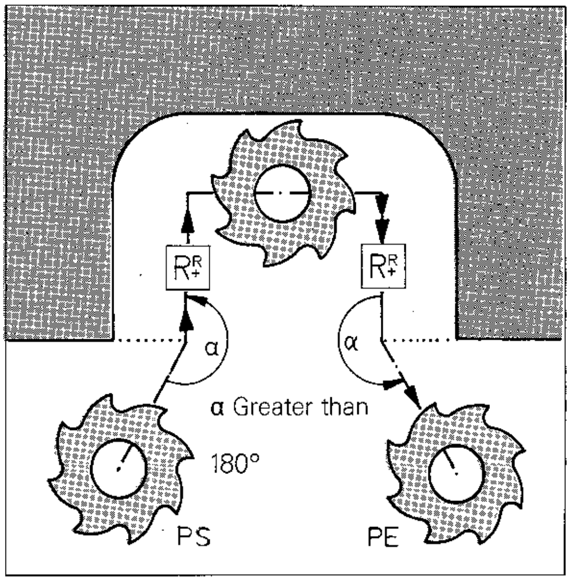
There are normally three cases which can be considered:

- Path angle **α** = 180°

- Path angle **α** less than 180°



- Path angle **α** greater than 180°

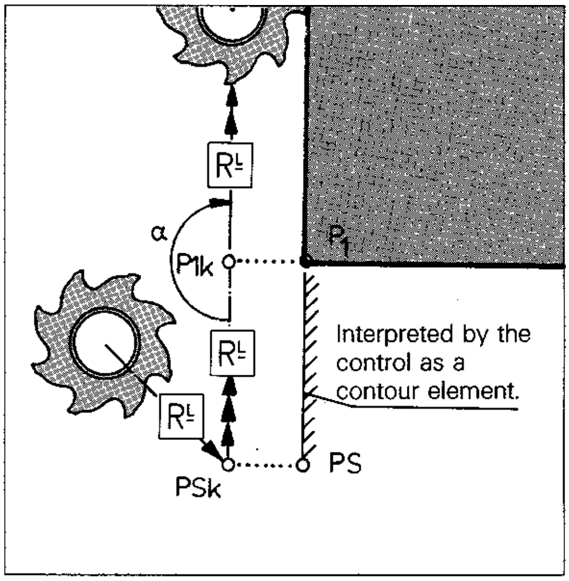


Contour approach and departure in a straight path
Path angle α equal to 180°

Path angle α = 180°

If the path angle α is equal to 180°, the starting and finishing position is located on the extension of the last position of a straight contour or the tangent of the first/last contour position with circular shaped contours.

The starting and finishing position must be programmed **with radius compensation** (RL or RR).

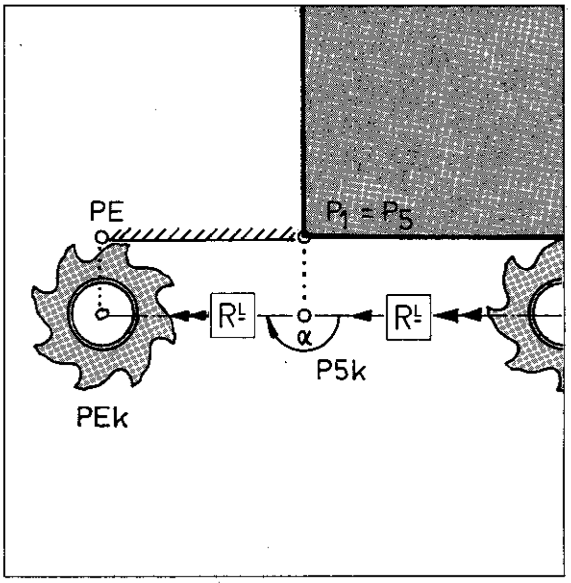


Approach (Run-on)

The tool moves in a straight path to the compensated position PSk of contour position PS and then proceeds to the position P1k on a compensated path.

Departure (Run-off)

The tool moves from the compensated position P5k of contour point P5 in a compensated path to position PEk.

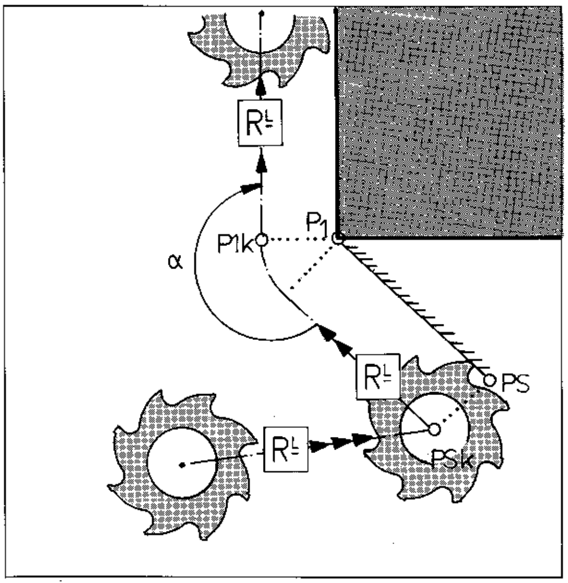


Contour approach and departure in a straight path
Path angle α greater than 180°

Path angle α greater than 180°

With α greater than 180°, the starting and finishing position must be programmed **with radius compensation** (RL or RR).

The first and last contour position is assumed as being an external corner. The control implements path compensation for an external corner and inserts a transitional arc.



Approach (Run-on)

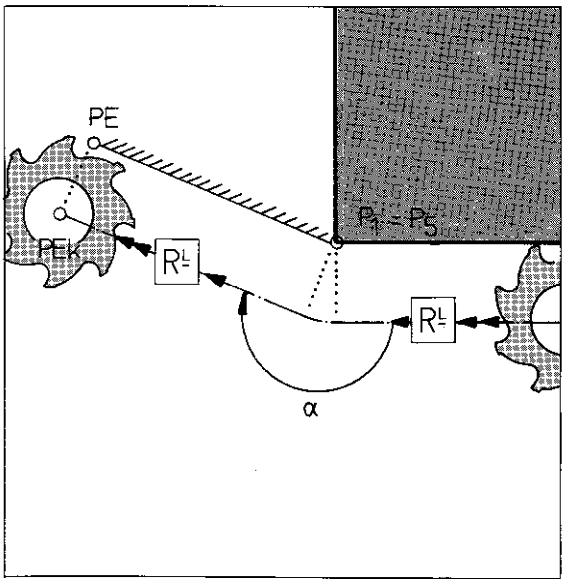
The control considers the starting position PS as being the first contour position.

The tool moves to position PSk and then on a compensated path to position P1k.

Departure (Run-off)

The control considers the finishing position PE as being the last contour position.

The tool moves to the finishing position PEk on a compensated path.

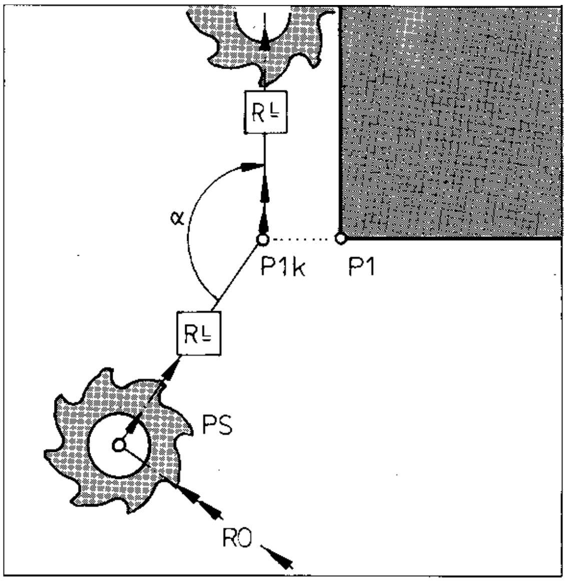


Contour approach and departure in a straight path
Path angle α less than 180°

Path angle α less than 180°

With α less than 180°, the starting and finishing position must be programmed **without compensation,** i.e. with R0.

PS and PE are positioned without path compensation.

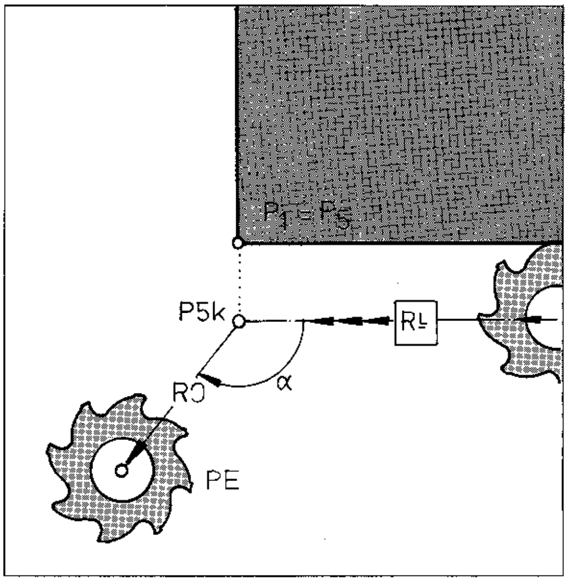


Approach (Run-on)

The tool moves from PA in a straight path to the position P1k of contour position P1.

Departure (Run-off)

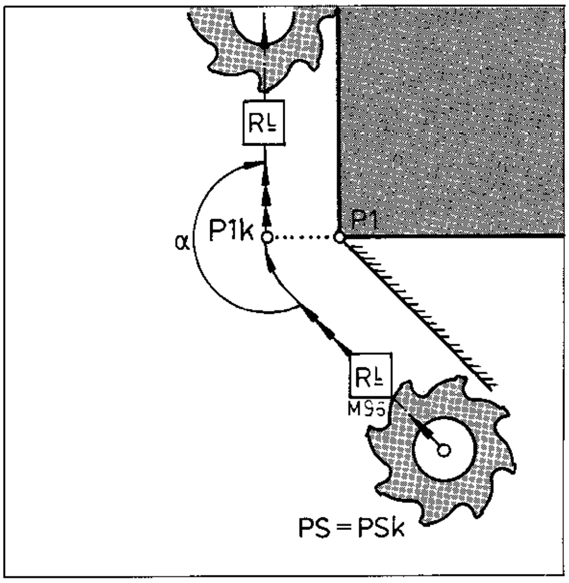
The tool moves from the compensated position P5k of contour position P1 in a straight path to the uncompensated position PE.



Contour approach and departure in a straight path
Approach command M96
Departure command M98

Approach command M96

If position PS has been programmed without tool compensation and the path angle α for contour approach is greater than 180°, contour damage will occur.
With the auxiliary function M96, the starting position PS is interpreted as a compensated position PSk.
The tool is positioned to P1k on a compensated path.
With path **angles α greater than 180°,** the auxiliary function M96 must be programmed. M96 is programmed in the block for P1.



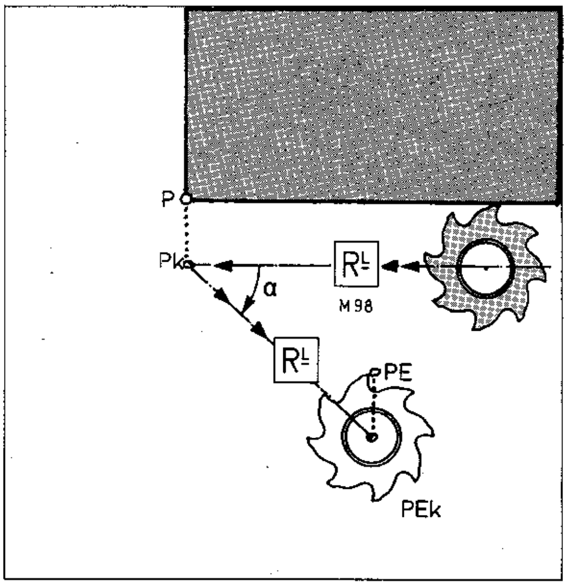
M96 is effective when normal path compensation is ineffective at the beginning of the program (R0).

If M96 is programmed with path angles α less than 180°, machining of the contour will be incomplete.

Departure command M98

If the finishing position is programmed with compensation and with a **departure angle α less than 180°,** contour machining will be incomplete.

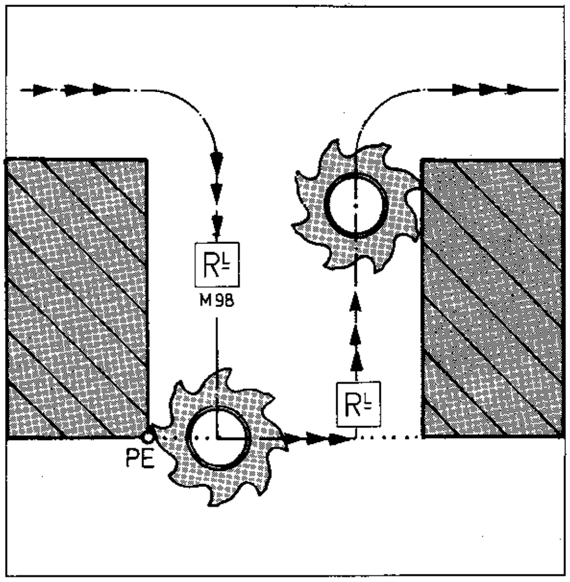
By programming M98 into the block for P, the tool is positioned directly to position Pk and then to the compensated position PEk. The direction PE-PEk corresponds to the radius offset last executed; in this example P-Pk.



Termination of path compensation M98

If further contour positions have been programmed subsequent to PE, the direction for the radius offset depends on the direction of the next contour section.

An M98 within the block for the last contour position ensures that the contour element is completely executed and that the first position of the subsequent contour is approached to with radius compensation as per the adjacent example.



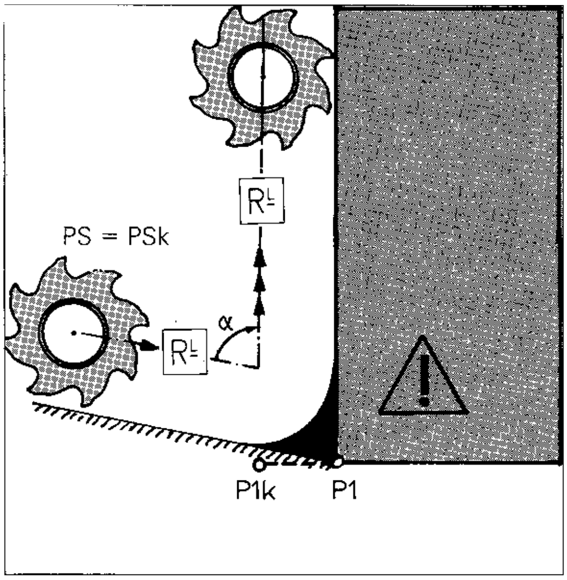
The auxiliary function M98 is only effective blockwise. In the subsequent block, M98 prevents the insertion of transitional arcs on external corners and the calculation of path intersections on internal corners. RL must be re-entered after M98.

Contour approach and departure in a straight path

Tool in start position
Approach command M95

Problem with approach angle α less than 180°

At the beginning of the program, the tool **happens to be located at the actual position PS** or the position PS has been approached with compensation (PS = PSk) and position P1k cannot be approached due to the path compensation.

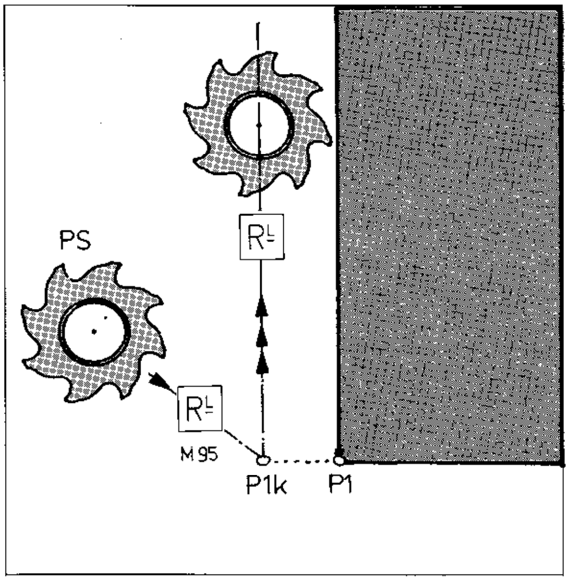


Approach command M95

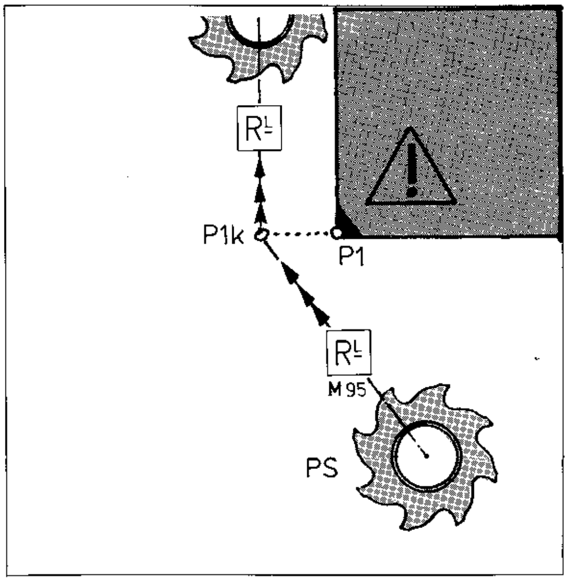
With auxiliary function M95, path compensation for the first positioning block is cancelled. The tool travels from position PS to the compensated contour P1k without path compensation.

The auxiliary function M95 is programmed when the approach angle α is less than 180°. It is programmed into the block for position P1.

M95 is only effective at the beginning of a machining program!
Cancellation of path compensation within a machining program is performed with function M98 (see "Termination of path compensation").



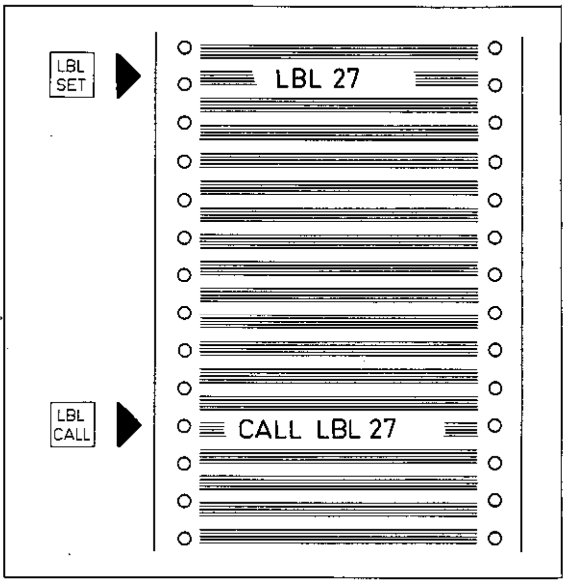
If M95 is programmed with an approach angle α exceeding 180°, contour damage will occur.



Subprograms and program part repeats
Program markers (Labels)

Label

When programming, labels with a certain number can be set to mark a program section as e. g. a subprogram (sub-routine).
Jumps can be made to such label numbers during program run (e. g. for execution of the appropriate subprogram).



Setting a label LBL SET

A label is set by pressing the LBL SET -key.

Label number

Label numbers from 0 to 254 may be allocated. Label number 0 always signifies the **end of a subprogram** (see "Subprogram") and is therefore considered as a return jump marker!

If a label number is entered which has already been allocated somewhere else within the program, the following error is displayed:

= NUMBER ALREADY ALLOCATED =.

Calling-up a label LBL CALL

Dialogue is initiated by pressing LBL CALL
With LBL CALL

- **Subprograms** can be retrieved.
- **Program part repeats** can be set.

Label number

Label number 1 – 254 may be called-up.

If the number 0 is entered, the following error is displayed:

= JUMP TO LABEL 0 NOT PERMITTED =.

Repetition REP

With **program part repeats** the question "REPEAT REP" is responded to by entering the required number of repetitions.
The question REP is responded to by pressing NO ENT for subprogram calls.

Subprograms and program part repeats
Labels

Setting a label

Operating mode ________ ◈

Dialogue initiation ________ LBL SET

LABEL NUMBER?	▶ □	Key-in label number.
	▼ ENT	Enter into memory.

Display example

```
118 LBL 27
```

Label number 27 has been allocated to block 118.

Label call

Operating mode ________ ◈

Dialogue initiation ________ LBL CALL

LABEL NUMBER?	▶ □	Key-in label number to be called-up.
	▼ ENT	Enter into memory.

▼

REPEAT REP?

If a **program part repeat** is to be entered:	▶ □	Key-in the number of repetitions.
	▼ ENT	Enter into memory.
If a **subprogram call** is to be entered:	▶ NO ENT	Entry not required.

Display example 1

```
218 CALL LBL 27 REP
```

The subprogram having label number 27 is called-up (continuation of machining with block number 118 above).

Display example 2

```
29 CALL LBL 5 REP 2/2
```

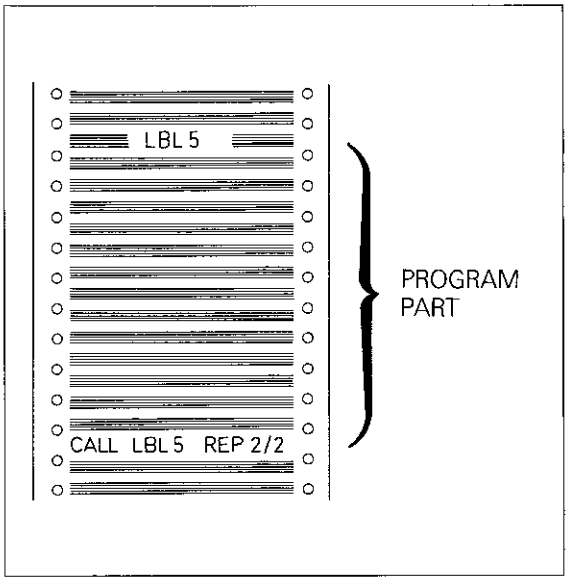
A program part is repeated two times. The number after the dash is a countdown indicating the number of repetitions which are still to be executed. This number is reduced by 1 after completion of each program part.

Subprograms and program part repeats
Program part repeat

Program part repeat

A program section which has been executed can be repeated if required. This is referred to as a program loop or **program part repeat.**

The beginning of the program part which is to be repeated is marked with a **label number.**
The end of the program part is formed by a **LBL CALL** in conjunction with the **number of repeats REP.**



Program run

The control executes the main program (including the appropriate program part) until call-up of the label number.

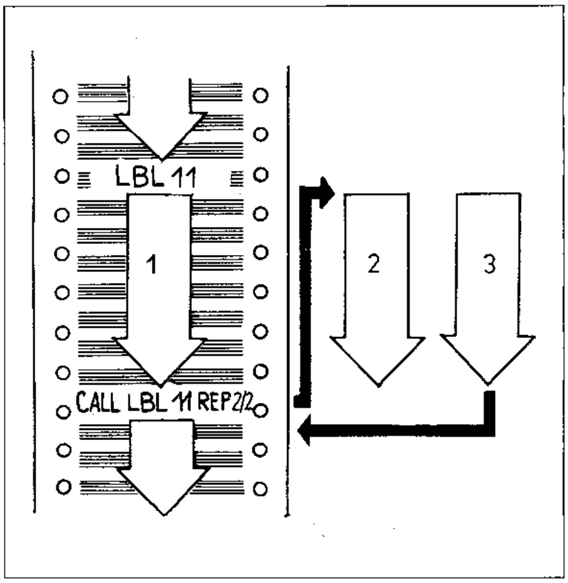
A jump is then made to the program label and the program part is repeated.

The display countdown reduces the number of repetitions by 1: REP 2/1.

After a new jump, the program part is repeated again.

When all programmed repetitions have been executed, (display: REP 2/0), the main program is continued.

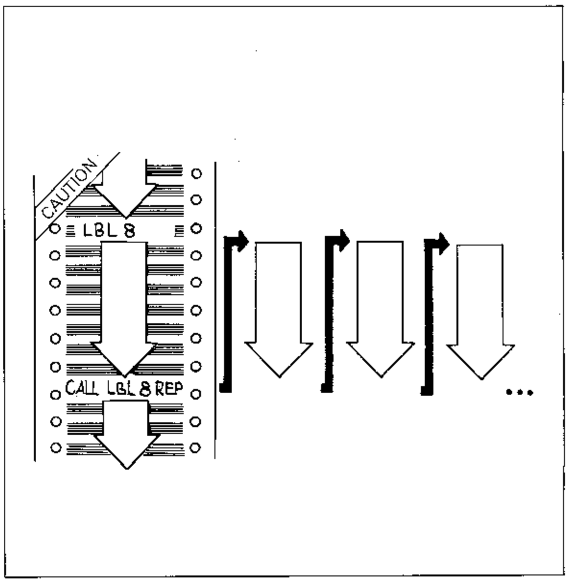
The program part is always executed (in total) by the number of programmed repetitions **plus one.**



Infinite loop

If **no entry** is made (by pressing NO ENT) in response to the question concerning the number of repeats **REP,** an endless loop will take place: the **call-up** of the label number is **repeated** constantly.

During program run and a test run, an infinite loop is indicated after 8 repetitions by the error message:
= EXCESSIVE SUBPROGRAMMING =.



Subprograms and program part repeats
Subprogram

Subprogram

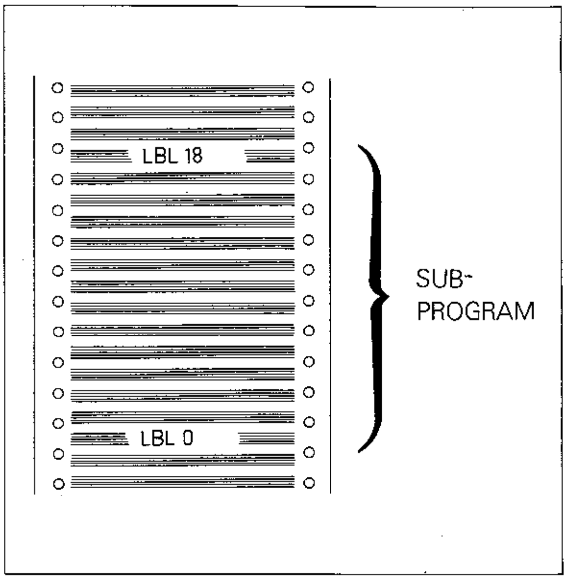
If a program part is required at another location within the machining program, this program section is referred to as a sub-routine or subprogram.

The **beginning** of the subprogram is labelled with a **label number. The end of subprogram** is always labelled with the **label number 0.**

If the end of the subprogram is not labelled with 0, the subprogram call with result into an infinite loop (see "Infinite loop").

The subprogram is retrieved via a LBL CALL command. LBL CALL can be made at any location within the program.

After execution of the subprogram, a return jump is made to the main program.



Program run

The control works through the main program until the subprogram call-up (CALL LBL 27 REP).

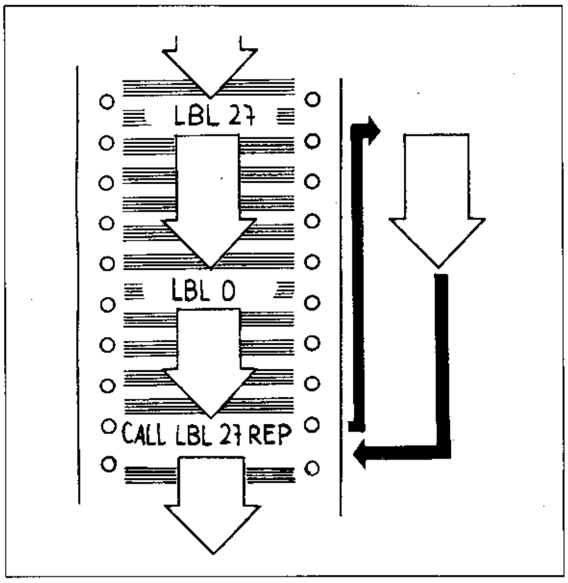
A jump is then made to the label called.

The subprogram is executed until label number 0 (subprogram end).

Finally, a return jump is made into the main program.

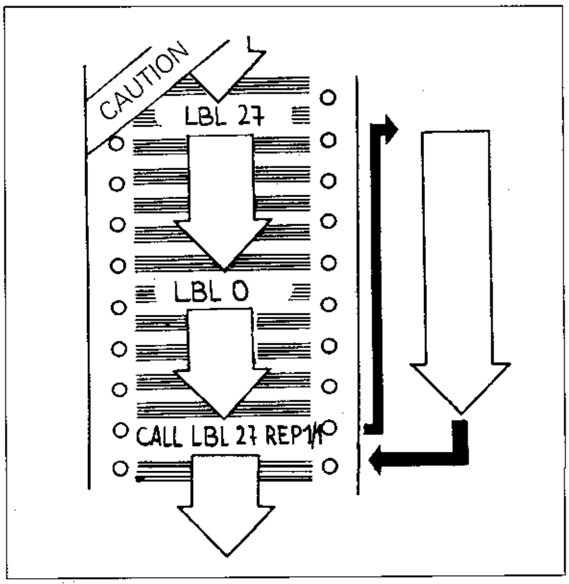
The main program is continued from the block immediately after the subprogram call.

If the subprogram lies within the main program, as in the above example, it is executed once without a call-up.



A subprogram can only be executed once via a call-up command! When retrieving a subprogram via LBL CALL, the dialogue question REPEAT REP? must be responded to by pressing NO ENT.

If a repetition is programmed e.g. REP 1/1, the section between the label number called and the CALL LBL command, is executed as a program part repeat. The label LBL 0 is not taken into account.



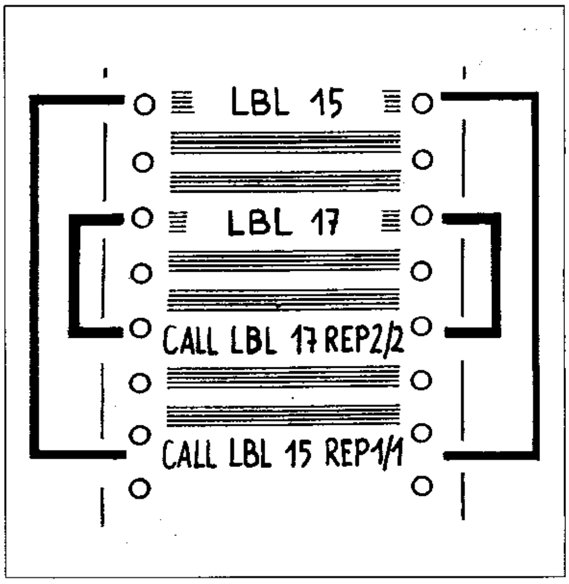
Subprograms and program part repeats
Nesting

Nesting

A further subprogram or program part repeat can be called-up within an existing subprogram or program part repeat. This procedure is referred to as **nesting.** (Illustrative example: set of boxes or tables etc. fitting one inside another).

Program parts and **subprograms** can be nested up to 8 times, i. e. the **nesting level** totals 8.

If the nesting level has been exceeded, the following error is displayed:
= EXCESSIVE SUBPROGRAMMING =.

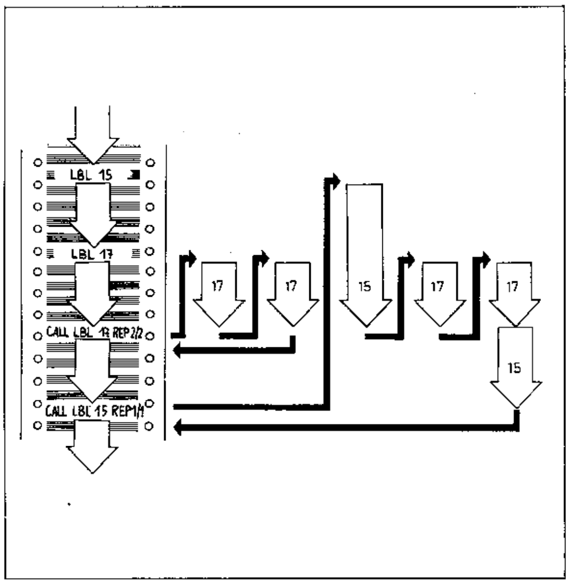


Program run with repetition

The main program is executed until a jump is made to LBL 17.

The program part is repeated twice.

Afterwards, the control continues program execution until a jump to LBL 15. The program part is repeated once until CALL LBL 17 REP 2/2 and the nested program part twice in addition. The program part last programmed is then continued to CALL LBL 17.

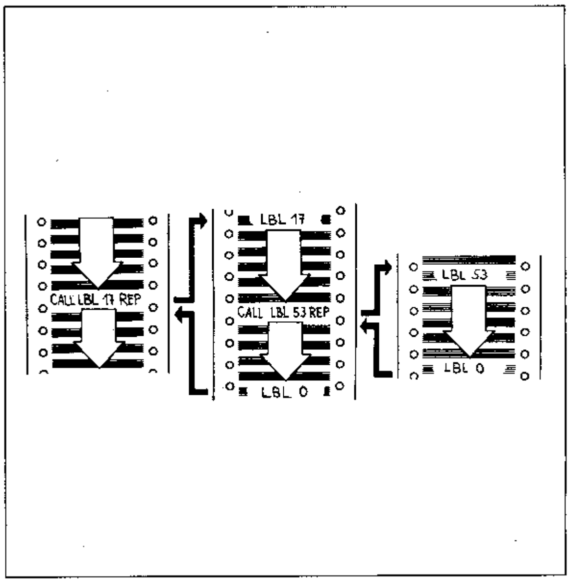


Program run with subprograms

The main program is executed until the jump command CALL LBL 17.

Afterwards, the subprogram is executed from LBL 17 to the next call-up CALL LBL 53 etc. The last subprogram within the series of nests is executed without interruption.

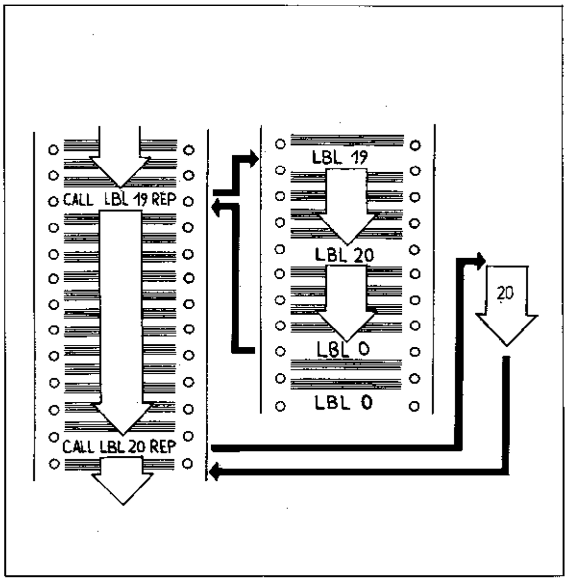
Before the end of the last subprogram (LBL 0), a return jump is made to each previous subprogram until the main program is reached again.



Subprograms and program part repeats
Nesting

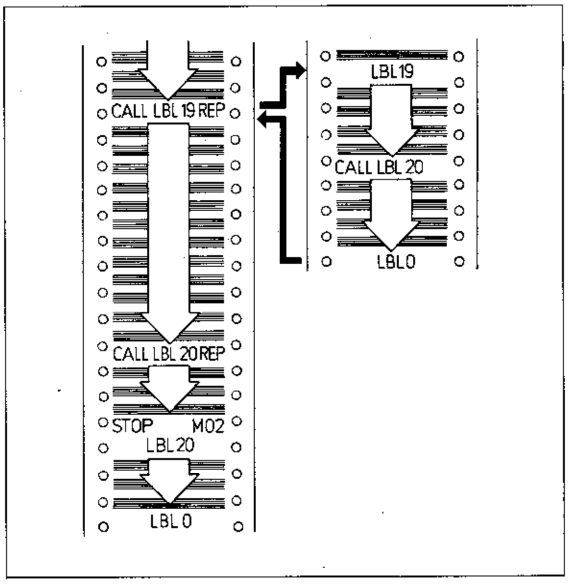
A subprogram with a subprogram

A subprogram cannot be programmed into an existing subprogram. As per the adjacent example, each of the subprograms is only executed to the label number 0.



In this case, the subprogram 20 should be programmed at the end of the main program, however separated from the main program by a STOP M02.

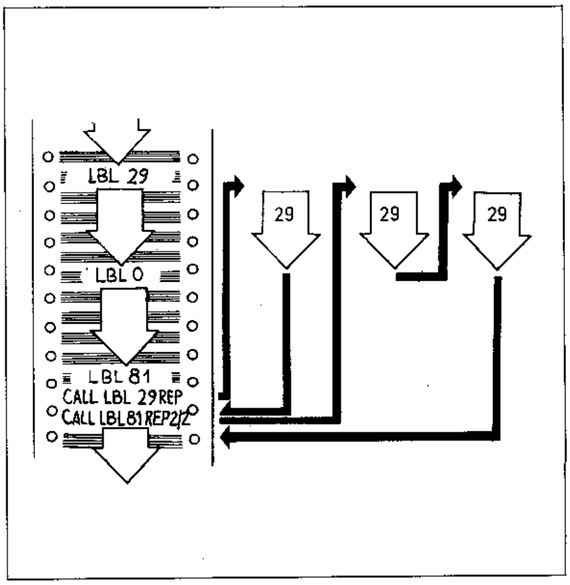
Subprogram 20 is called-up via CALL LBL 20 within subprogram 19.



Repetition of subprograms

With the aid of nesting, it is possible to repeat subprograms.

The subprogram is called-up within a program part repeat. This subprogram call is the only block of the program part repeat.
During program run, care should be taken that the subprogram is executed one time more than the number of repetitions programmed.



Program jump

A jump into another main program

Program management of the control permits a jump from one main program to another.

This enables

- home-made machining cycles to be compiled by using parameter programming (see cycle "program call")

or

- the storage of tool lists.

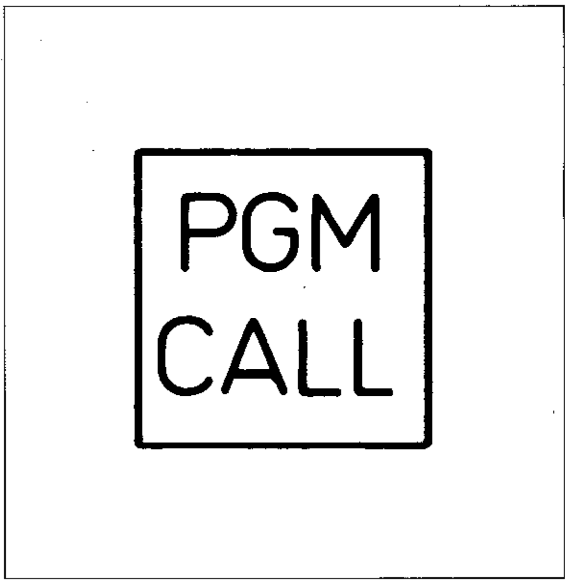
Programming of the jump is initiated with the PGM CALL -key.

If a program number, to which no program has been allocated, is entered (e.g. CALL PGM 13), the error

= PGM 13 UNAVAILABLE =

is displayed when selecting the main program via the jump command.

Max. **four nesting levels** are permitted for program calls, i.e. the nesting level is 4.



Program run example

The control executes the main program 1 until the program call command CALL PGM.

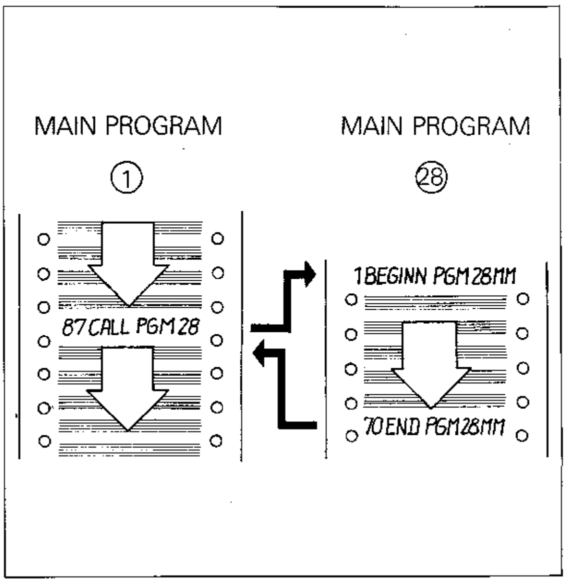
A jump is then made into the main program 28.

Program 28 is completely executed from start to finish.

A return jump is then made into main program 1.

Main program 1 is then continued from the block subsequent to the program call.

A return jump into the original main program must not be programmed within the main program which has been called-up (this forms an infinite loop.



Program jump

Entry

Operating mode ______________ ◈

Dialogue initiation ______________ PGM CALL

PROGRAM NUMBER?	▶ □	Key-in number of program to be called-up.
	▼ ENT	Enter into memory.

Display example

87 CALL PGM 28

Main program 28 has been called-up in block 87

A program call can be programmed as per a cycle call, if the program number has been entered into the cycle definition 12.

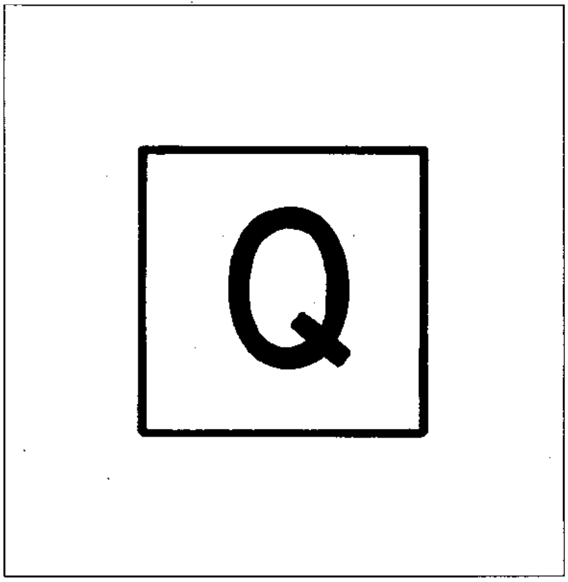
By doing this, the home-made cycles which have been prepared via parameter programming are treated as permanently programmed cycles (see cycle "program call").

Parameters

Parameters

Within a program, numerical values which are related to units of measure (co-ordinates or feed rate) can be substituted by **variable parameters** for numerical values which are either entered at a later stage or calculated by the control.

When executing the program, the control then uses the numerical value which the parameter provides in the parameter definition.



Setting parameters

Parameters are designated by the letter Q and a number between 0 and 99. Parameters may be entered with a negative sign. Positive signs do not have to be programmed. The Q -key is used for setting a parameter.

Parameter definition

The correlation of certain numerical values to the parameters is either possible directly or via mathematical and logical functions.

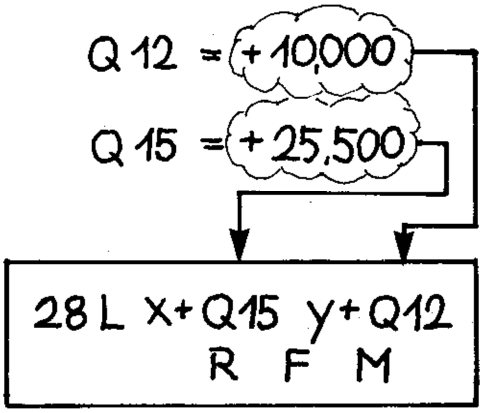
The dialogue for parameter definitions is initiated with the Q DEF -key. The adjacent **parameter functions FN** can be selected with the ↑ ↓ -keys.

FN 0:	**ASSIGN**
FN 1:	**ADDITION**
FN 2:	**SUBTRACTION**
FN 3:	**MULTIPLICATION**
FN 4:	**DIVISION**
FN 5:	**SQUARE ROOT**
FN 6:	**SINE**
FN 7:	**COSINE**
FN 8:	**ROOT SUM OF SQUARES**
FN 9:	**IF EQUAL, JUMP**
FN 10:	**IF UNEQUAL, JUMP**
FN 11:	**IF GREATER THAN, JUMP**
FN 12:	**IF LESS THAN, JUMP**

Parameter definition example

If parameters are entered instead of co-ordinates within a linear interpolation, contours can be produced which are based on mathematical functions e.g. ellipses. The contour is then formed by a large number of individual straight sections. (see also programming example "Ellipse")

With parameter programming a calculation step can take between 3 ms and 20 ms. With complex mathematical functions and high feed rates, standstill on the contour may occur.



Parameters

Setting a parameter

Dialogue question e. g.

COORDINATES?

▶ X	Select axis, e. g. X.
▼ Q	Press parameter-key.
▼ □	Key-in parameter number.
▼ +/–	If reqd., key-in sign
ENT	Enter into memory.

Display example

27 L X + Q13 Y – Q2

R F M

Parameter Q13 is an allocation for the numerical value of the X-co-ordinate.
Parameter Q2 is an allocation for the negative Y-co-ordinate value.
Q13 is for example, assigned with the value +40.000 and Q2 +19.000. The tool will therefore move to the position P (X +40.000/Y –19.000).

Parameters must be defined before call-up. At the beginning of program run, non-defined parameters are automatically assigned with the value 0.
In the above display example, the tool would traverse to the position X0/Y0.

Addressing a parameter function

Operating mode ______________________ ◈

Dialogue initiation ______________________ Q DEF

FNO: ASSIGN ▶ ↓ or ↑ Select reqd. parameter function.

If the reqd. function is in the display, e.g.

▼

FN 9: IF EQUAL, JUMP ▶ ENT Enter into memory.

The first dialogue question appears in the display (see corresponding function for response).

Parameters
Parameter functions

FN 0:
Assign

With function FN 0, a parameter is assigned with a **numerical value** or **another parameter.** Assignment is designated by a "=" sign.

Q5 = 65,432

Display:

```
18 FN 0: Q5 = +65,432
```

FN 1:
Addition

With function FN 1, a certain parameter is defined as the **sum** of two parameters or two numerical values or a parameter and a numerical value.

Q17 = Q2 + 5,000

Display:

```
12 FN 1: Q17 = +Q2
               + +5,000
```

FN 2:
Subtraction

With function FN 2, a certain parameter is defined as the **difference** between two parameters or two numerical values or a parameter and a numerical value.

Q11 = 5,000 − Q34

Display:

```
94 FN 2: Q11 = +5,000
               − +Q34
```

FN 3:
Multiplication

With function FN 3, a certain parameter is defined as the **product** of two parameters or two numerical values or a parameter and a numerical value.

Q21 = Q1 x 60,0

Display:

```
85 FN 3: Q21 = +Q1
               * +60,000
```

FN 4:
Division

With function FN 4, a certain parameter is defined as the **quotient** of two parameters or two numerical values or a parameter and a numerical value.
(**DIV:** abbrevation for **division**)

Q12 = Q2 / 62

Display:

```
73 FN 4: Q12 = +Q2
           DIV   +62,000
```

FN 5:
Square root

With function FN 5, a certain parameter is defined as the **square root** of a parameter or a numerical value.
(**SQRT:** abbrevation for **square root**)

Q98 = $\sqrt{2}$

Display:

```
69 FN 5: Q98 = SQRT +2
```

Parameters
Parameter functions

Programming example FN 1

Operating mode ________________

Dialogue initiation ________________ Q DEF ↓

FN 1: ADDITION	ENT	Enter function into memory.

PARAMETER NUMBER FOR RESULT?	□	Key-in parameter number.
	ENT	Enter into memory.

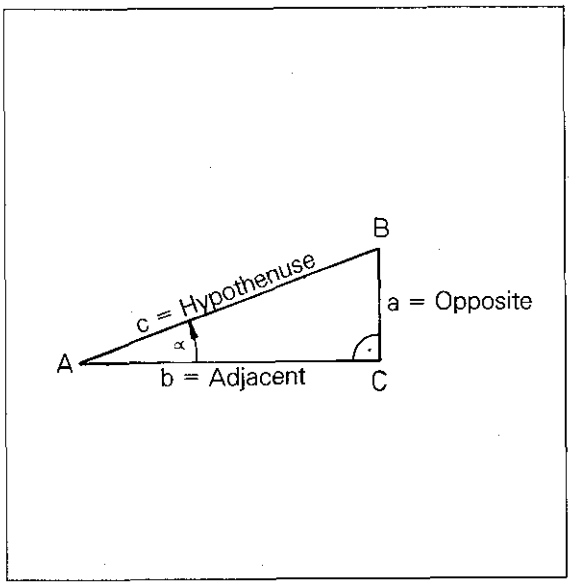
FIRST VALUE/PARAMETER?		
If a numerical value is assigned:	□	Key-in value.
	ENT	Enter into memory.
If a parameter is assigned:	Q	Press parameter key.
	□	Key-in parameter number.
	ENT	Enter into memory.

SECOND VALUE / PARAMETER?		
If a numerical value is assigned:	□	Key-in value.
	ENT	Enter into memory.
If a parameter is assigned:	Q	Press parameter key.
	□	Key-in parameter number.
	ENT	Enter into memory.

Parameters
Parameter functions

Trigonometrical functions

Sine and cosine functions form a mathematical relationship between an angle and a side length of a right-angled triangle. Trigonometrical functions are programmed with
FN 6: sine and
FN 7: cosine



Definition of trigonometrical functions

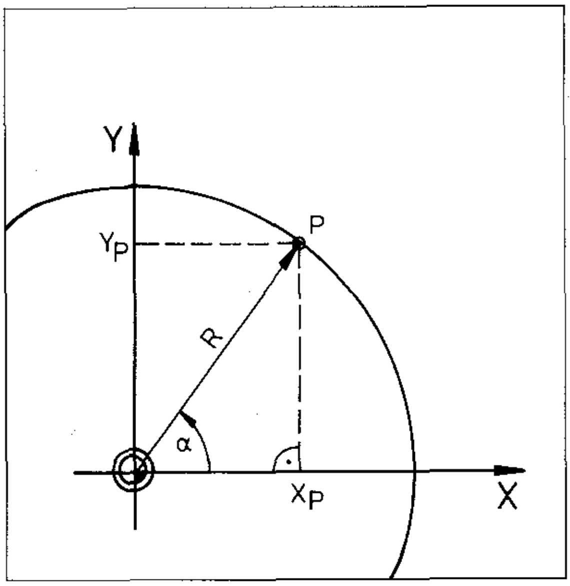
$$\sin \alpha = \frac{\text{Opposite side}}{\text{Hypothenuse}} = \frac{a}{c}$$

$$\cos \alpha = \frac{\text{Adjacent side}}{\text{Hypothenuse}} = \frac{b}{c}$$

Trigonometrical functions within a right-angled triangle

$X_P = R \times \cos \alpha$

$Y_P = R \times \sin \alpha$



FN 6: sine

With function FN 6 sine, a certain parameter is defined as the **sine** of an angle (in degrees (°)). The angle can be a numerical value or a parameter.

```
Q10 = sin Q8

Display:

113 FN 6: Q10 = SIN + Q8
```

FN 7: cosine

With function FN 7 cosine, a certain parameter is defined as the **cosine** of an angle (in degrees (°)). The angle can be a numerical value or a parameter.

```
Q81 = cos (–Q55)

Display:

911 FN 7: Q81 = COS – Q55
```

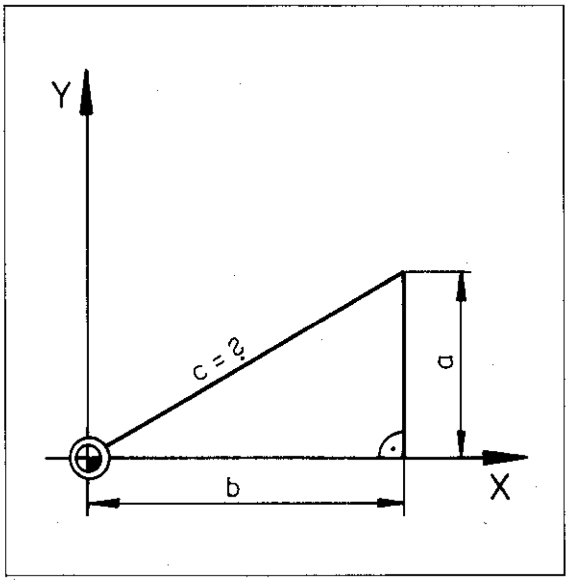
Parameters
Parameter functions

Length of a distance

Parameter function FN 8: root of sum of square, is used for determining the **length of a distance** within a right-angled triangle.

The Pythagoras theorem states:

$a^2 + b^2 = c^2$ or $\mathbf{c = \sqrt{a^2 + b^2}}$



FN 8: Root of sum of squares

With function FN 8, root of sum of squares, a certain parameter is defined as the **square root** of the sum of the squares of two numerical values or parameters.

(**LEN** = abbreviation for length).

$$\mathbf{Q3 = \sqrt{30^2 + Q45^2}}$$

Display:

```
56 FN 8: Q3 = +30,000
         LEN   +Q45
```

Parameters
Parameter functions

If-jump

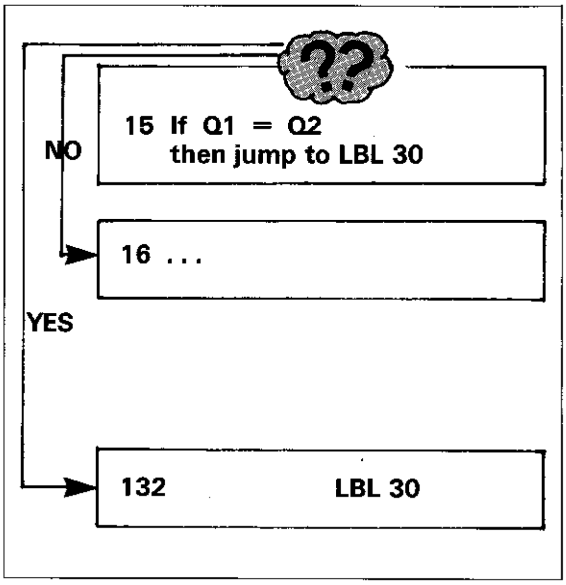
With parameter functions F 9 to F 12, a parameter can be compared with another parameter or with a numerical value.
Depending on the result of such a comparison, a jump can be made to a certain program label.

The equations are:

- First parameter is equal to a value or a second parameter, e.g. **Q1 = Q3**
- First parameter is different to a value or a second parameter, e.g. **Q1 ≠ Q3**
- First parameter is greater than a value or a second parameter, e.g. **Q1 > Q3**
- First parameter is less than a value or a second parameter, e.g. **Q1 < Q3**

=	equal
≠	unequal
>	greater than
<	less than

If one of these equations is satisfied, a **jump** is then made to a certain program label.
If the equation is not satisfied, the program is continued with the block which follows.



FN 9: If equal, jump

When programming the function FN 9, "If equal, jump", a jump to a program label is only made if a certain parameter is **equal to** another parameter or a numerical value.

IF = **If** or **when**
EQU = abbreviation for **equal**
GOTO = **"go to"** (proceed to)

If: Q2 = 360
then jump to LBL 30!

Display:

```
 47 FN 9: IF + Q2

     EQU + 360,000   GOTO LBL 30
```

Parameters
Parameter functions

Entry
Example FN 9

Operating mode ______ ⬗

Dialogue initiation ______ Q DEF ↑

FN 9: IF EQUAL, JUMP	▶ ENT	Enter function into memory.

FIRST VALUE / PARAMETER	▶ Q	Press parameter key.
	□	Key-in parameter number.
	ENT	Enter into memory.

SECOND VALUE / PARAMETER		
If the parameter set above is to be compared with a value.	▶ □	Key-in numerical value.
	ENT	Enter into memory.
If the parameter set above is to be compared with another parameter.	▶ Q	Press parameter key.
	□	Key-in parameter number.
	ENT	Enter into memory.

LABEL NUMBER?	▶ □	Key-in label number for jump.
	ENT	Enter into memory.

Display data is shown with the appropriate function on the following page.

FN 10: If unequal, jump

When programming, the function FN 10: "If unequal, jump", a jump to a label number is only made if a certain parameter is **unequal** to a numerical value or another parameter.

(**NE** = abbreviation for **not equal**).

If Q3 ≠ Q10,
then jump to LBL 2!

Display:

```
 38 FN 10: IF + Q3

          NE + Q10   GOTO LBL 2
```

FN 11: If greater than, jump

When programming the function FN 11: "If greater than, jump", a jump to a label number is only made if a certain parameter is **greater** than a numerical value or another parameter.

(**GT** = abbreviation for **greater than**).

If Q8 > 360,
then jump to LBL 17!

Display:

```
 28 FN 11: IF + Q8

          GT + 360,000   GOTO LBL 17
```

FN 12: If less than, jump

When programming the function FN 12: "If less than, jump", a jump to a label number is only made if a certain parameter is **less** than a numerical value or another parameter.

(**LT** = abbreviation for **less than**).

If Q6 < Q5,
then jump to LBL 3!

Display:

```
 24 FN 12: IF + Q6

          LT + Q5   GOTO LBL 3
```

Remarks

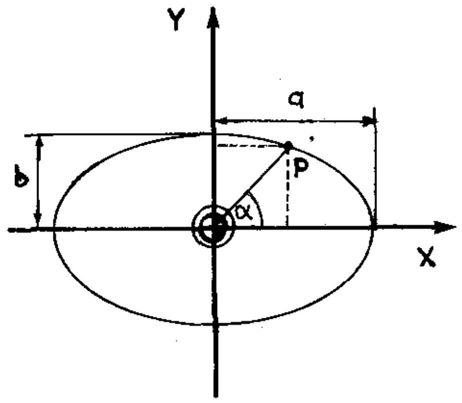
Parameters
Parameter programming (Example)

Programming with parameters can be explained in the example of an ellipse.

Geometry

The **ellipse** is described according to the adjacent formula (math. parameter form of the ellipse).

Every angle α has an X and Y-co-ordinate. Beginning at α = 0° and proceeding to α = 360° in small increments, a number of individual points are obtained forming an ellipse. These points are adjoined by straight lines to form a closed contour.

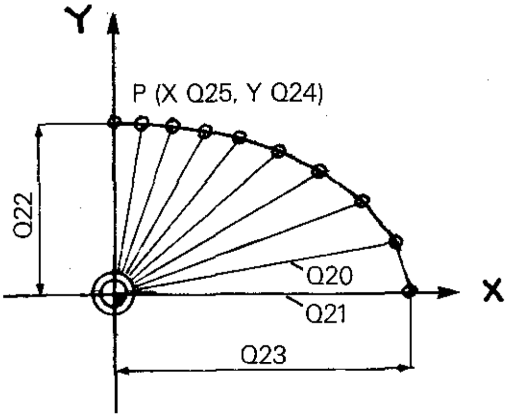


$X = a \cdot \cos \alpha$
$Y = b \cdot \sin \alpha$
a, b : semi-axes of the ellipse

Parameter definition

The program consists of 4 main sections:
- Parameter definition
- Positioning (linear interpolation) for milling of ellipse
- Increase of angular step
- Parameter comparison and program continuation until the ellipse is completed.

The following are defined as parameters:
- **Angular step Q20:** The angle is to increase in increments of 2°; Q20 = + 2.000
- **Starting angle Q21:** The first contour point has the angle 0°; Q21 = 0.000
- **Semi-axis in X-direction Q23:** Q23 = +50.000
- **Semi-axis in Y-direction Q22:** Q22 = +30.000
- **X-co-ordinate Q25:** The numerical value of the C-co-ordinate is assigned to parameter Q25.
- **Y-co-ordinate Q24:** The numerical value of the Y-co-ordinate is assigned to parameter Q24.



Parameters Q25 and Q24 are defined according to the above mentioned formula:

(X=) Q25 = Q23 ✱ cos Q21;
(Y=) Q24 = Q22 ✱ sin Q21;

Both equations must be converted, since they cannot be entered in this way, therefore:

first: Q14 = sin Q21
Q15 = cos Q21
then: Q24 = Q14 ✱ Q22
Q25 = Q15 ✱ Q23

```
Q20 = +  2.000
Q21 = +  0.000
Q22 = +30.000
Q23 = +50.000
```

```
Q14 = SIN  + Q21
Q15 = COS + Q21
Q24 = + Q14  ✱ + Q22
Q25 = + Q15  ✱ + Q23
```

Parameters

Parameter programming (Example)

Positioning block Milling of the ellipse is programmed within a block with linear interpolation.

```
Q20 = +  2,000
Q21 = +  0,000
Q22 = +30,000
Q23 = +50,000

Q14 = SIN  + Q21
Q15 = COS + Q21
Q24 = + Q14  *  + Q22
Q25 = + Q15  *  + Q23

L       X + Q25       Y + Q24
                      RR  F200    M
```

Increase of angular step New angle Q21 =
Old angle Q21 + angular step Q20

```
Q20 = +  2,000
Q21 = +  0,000
Q22 = +30,000
Q23 = +50,000

Q14 = SIN  + Q21
Q15 = COS + Q21
Q24 = + Q14  *  + Q22
Q25 = + Q15  *  + Q23

L       X + Q25       Y + Q24
                      RR  F200    M

Q21 = + Q21 + + Q20
```

Parameter comparison and program repeat For a repetition, a label must be set prior to the parameter definition for Q25 and Q24: LBL 1.

The repetition is governed by the following condition:

If angle Q21 is less than 360,1°, (however greater than 360°, but smaller than 360° plus the angular step) then jump to LBL 1:

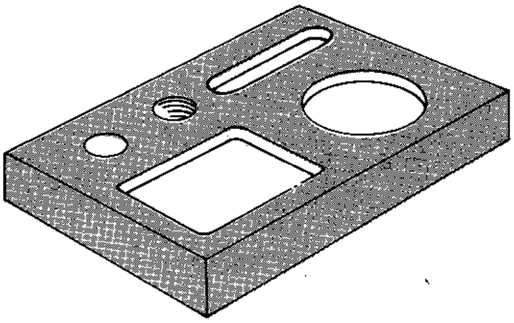
IF + Q21
LT + 360,100 GOTO LBL 1.

```
Q20 = +  2,000
Q21 = +  0,000
Q22 = +30,000
Q23 = +50,000

LBL 1

Q14 = SIN  + Q21
Q15 = COS + Q21
Q24 = + Q14  *  + Q22
Q25 = + Q15  *  + Q23

L       X + Q25       Y + Q24
                      RR  F200    M

Q21  = + Q21  + + Q20

IF + Q21
LT + 360,100 GOTO LBL 1
```

Canned cycles
Introduction

Canned cycles

To simplify and speed-up programming, reoccurring machining routines and certain co-ordinate transformations are pre-programmed as fixed – or canned – cycles. E.g. the milling of pockets or the shifting of a workpiece datum to another location.



Cycle definition

With the cycle definition, the control is informed of the necessary data for the cycle, e.g. side length of the pocket. Dialogue for cycle definition is initiated with the CYCL DEF-key. Cycles can be addressed with the ↓ ↑ -keys.

Breakdown of available cycles

Cycles 1 to 5 are **machining cycles,** i.e. machining routines are executed on the workpiece.
With cycle 9, a **dwell time** can be programmed and a program can be called-up via cycle 12.
The remaining cycles are used for various types of **co-ordinate transformations.**

CYCL DEF 1	Peck drill	Machining cycles
CYCL DEF 2	Tapping	
CYCL DEF 3	Slot milling	
CYCL DEF 4	Pocket milling	
CYCL DEF 5	Circular pocket	
CYCL DEF 7	Datum shift	Co-ordinate trans-formations
CYCL DEF 8	Mirror image	
CYCL DEF 10	Co-ordinate system rotation	
CYCL DEF 11	Scaling	
CYCL DEF 12	Program call	
CYCL DEF 9	Dwell time	

Cycles for co-ordinate transformations effect an end to tool compensation.

Cycle call

The cycle call enables the cycle and the dwell time which has been previously defined, to be executed.
Co-ordinate transformations do not require a special call-up; they are active immediately after cycle definition.
There are three programming possibilities for cycle call:

- Call-up with a CYCL CALL-block
- Call-up via auxiliary function M99
- Call-up via auxiliary function M89 (depending on the machine parameters entered)

Call-up M89 is modally effective, this means a call-up of the machining cycle last programmed is made with each subsequent positioning block. M89 is cancelled either by the entry of M99 or a CYCL CALL-block.

A cycle call is only effective for the last machining cycle which was defined.

Canned cycles
Cycle definition
Cycle call

Definition of a cycle

Operating mode ______________________ ◈

Dialogue initiation ______________________ CYCL DEF

CYCL DEF 1 PECKING	▶ ↓ ↑	Select required machining cycle.
The cycle is displayed e.g. ▼		
CYCL DEF 4 POCKET MILLING	▶ ENT	Enter cycle into memory.
The display shows the first dialogue question of the selected cycle. (See appropriate cycle definition for response).		

Call-up of a cycle

Operating mode ______________________ ◈

Dialogue initiation ______________________ CYCL CALL

AUXILIARY FUNCTION M?	▶ □	If reqd., key-in auxiliary function.
	▼ ENT	Enter into memory.

Display example

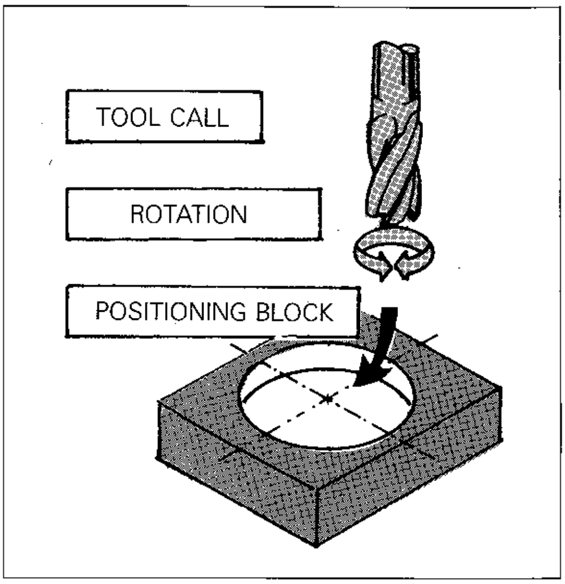
95 CYCL CALL	The cycle last defined is called-up.
M03	The spindle rotates clockwise.

Canned Cycles
Machining cycles
Preparatory measures

Provisions

The following must be programmed **prior to a cycle call:**

- **Tool call:** for definition of the **working spindle axis** and **spindle speed**
- **Auxiliary function:** for specification of the rotating direction
- **Positioning block to start position:** of machining cycle.



Error messages

The **absence of a tool call** is indicated by

= TOOL CALL MISSING =.

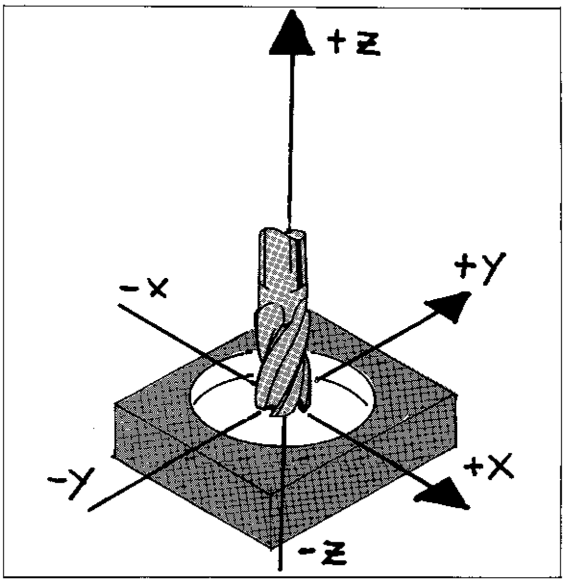
The **absence of the spindle rotating direction** is indicated by

= SPINDLE ROTATES MISSING =.

Dimensioning

Specification of dimensions within the cycle definition are always referenced to the **starting position** of the tool and are always incremental.

The I -key does not have to be pressed!

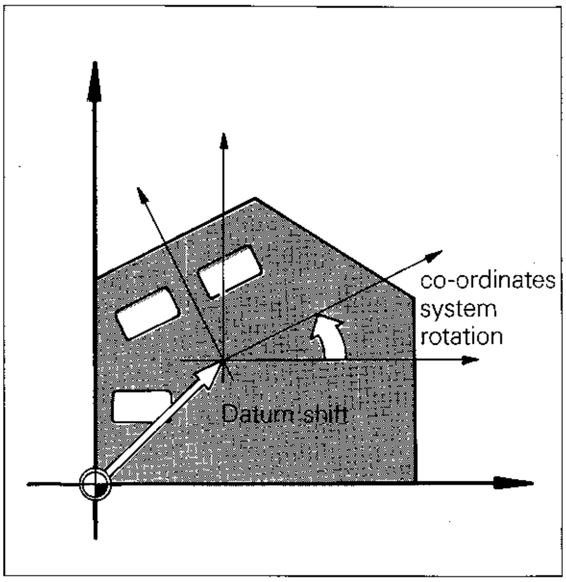


Machining cycles must (as opposed to coordinate transformations) be called up for execution.

Canned cycles
Co-ordinate transformations

General

Co-ordinate transformations alter the co-ordinate system which was determined with the workpiece zero. These cycles are effective immediately after the definition and a cycle call is therefore unnecessary.



Cancellation of a cycle

Co-ordinate transformations remain active until they are cancelled. This can be done either with a new cycle definition-with which the original condition is programmed-or with the auxiliary function M02, M30 or with the block END PGM...MM (depending on the entered machine parameter 173).

Canned cycles
Peck-drilling

Entry data

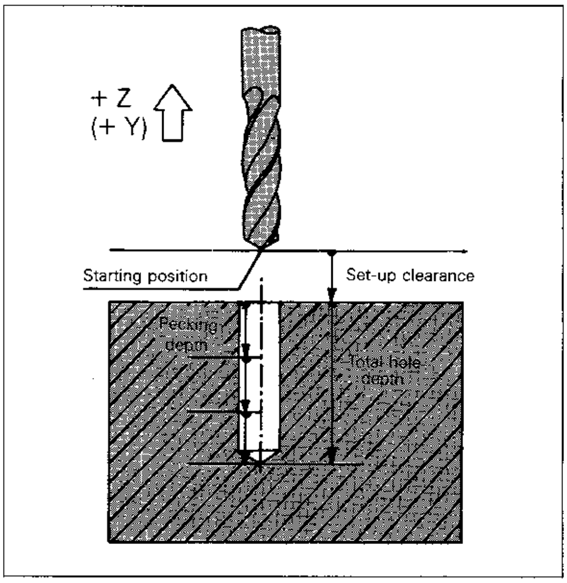
Set-up clearance: Distance between tool tip (starting position) and workpiece surface.
Arithmetical sign:
- ● in positive axis direction +
- ● in negative axis direction –

Total hole depth: Distance between workpiece surface and base of hole (tip of drill-taper).
See safety clearance for arithmetical sign.

Pecking depth: Depth of single penetration during pecking action.
See safety clearance for arithmetical sign.

Dwell time: Duration of tool standstill time upon reaching the total hole depth for chip breaking.

Feed rate: Feed speed of tool axis during operation.

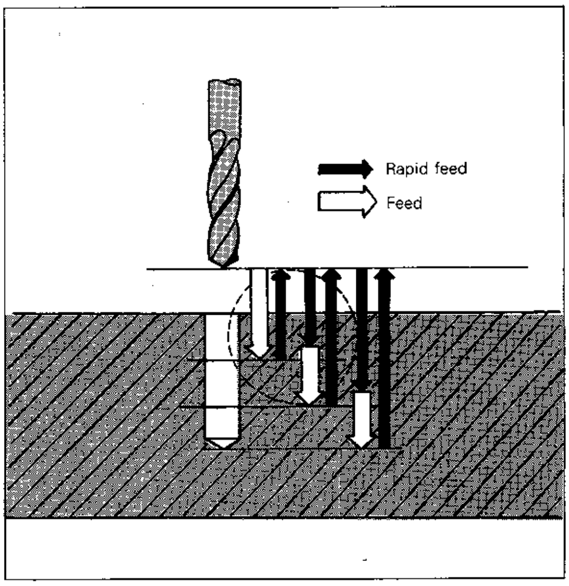


Procedure

From the **starting position,** the tool penetrates the work for the first pecking depth at the programmed **feed rate.** After reaching the first pecking depth, the tool is retracted to the starting position in rapid and then makes a new plunge taking the advanced stop distance into account.

The tool makes a further penetration by the pecking depth and then retracts again etc.

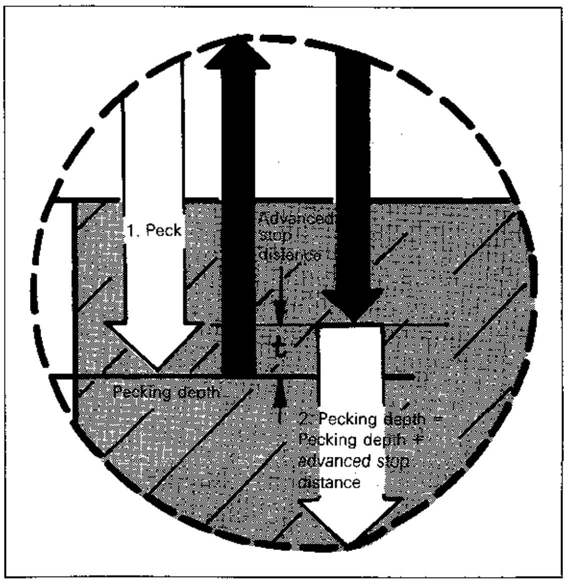
Pecking action is repeated until the programmed **hole depth** is reached. At the end of the cycle and after duration of the dwell time, the tool returns to the starting position.



Advanced stop

The advanced stop distance t is automatically determined by the control:
- ● with a drilling depth of up to 30 mm:
  t = 0.6 mm

- ● with a drilling depth exceeding 30 mm the following formula applies:
  t = drilling depth/50 whereby the max. distance is limited to 7 mm:
  t_{max} = 7 mm.



Canned cycles
Peck-drilling

Cycle definition

Operating mode ______________________ ◇

Dialogue initiation ______________________ CYCL DEF

CYCL DEF 1 PECKING	ENT	Enter cycle into memory.
SET-UP CLEARANCE?	□	Key-in set-up clearance.
	+/–	Key-in correct sign.
	ENT	Enter into memory.
TOTAL HOLE DEPTH?	□	Key-in hole depth.
	+/–	Key-in correct sign.
	ENT	Enter into memory.
PECKING DEPTH?	□	Key-in pecking depth.
	+/–	Key-in correct sign.
	ENT	Enter into memory.
DWELL TIME IN SECONDS?	□	Key-in dwell time at hole base.
	ENT	Enter into memory.
FEED RATE? F =	□	Key-in feed rate for pecking.
	ENT	Enter into memory.

The safety clearance, total hole depth and pecking depth must all have the same sign.

Remarks

Canned cycles
Peck-drilling

Display example

110 CYCL DEF 1.0 PECKING	The pecking cycle allocates 6 program blocks
111 CYCL DEF 1.1 SET-UP –2,000	Set-up clearance
112 CYCL DEF 1.2 DEPTH –30,000	Total hole depth
113 CYCL DEF 1.3 PECKG –20,000	Pecking depth
114 CYCL DEF 1.4 DWELL –0,000	Dwell time
115 CYCL DEF 1.5 F80	Feed rate

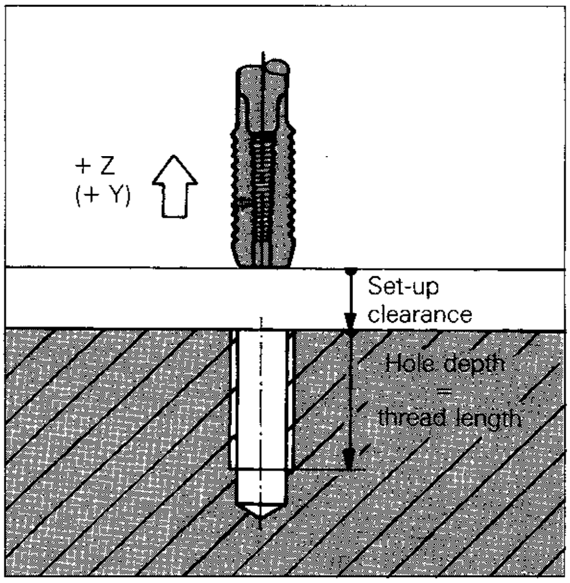
Canned cycles
Tapping

The cycle

The chuck must be able to compensate for the tolerances between the feed rate and the rotating speed as well as the deceleration in spindle rotation.

A **chuck with length compensation** is necessary for the tapping cycle.

After a cycle call, **the spindle override becomes ineffective** and **the feed rate override** is only active within a **limited range.** The limits have been set by the machine tool builder via parameters.



Entry data

Set-up clearance: (see cycle 1)
(approx. value: ca. 4 x thread pitch)

Total hole depth (= thread length): Distance between the workpiece surface and end of the thread. See set-up clearance for sign.

Dwell time: Duration between change of spindle rotation and retraction of tool

The entry value for the dwell time can be obtained from the machine tool builder.

Feed rate: Penetration speed during thread cutting.

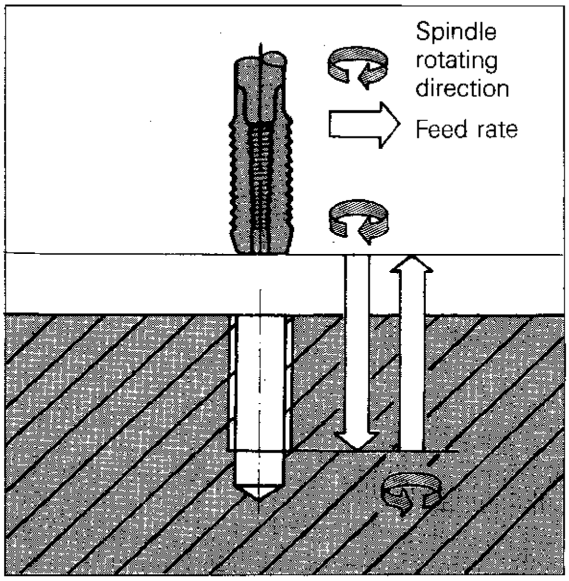
The feed rate value for the tapping cycle is determined with the following formula:

F = S x P

F: Feed rate
S: Spindle rpm
P: Thread pitch

Procedure

The thread is cut in one operation. When the tool reaches the **total hole depth,** the direction of spindle rotation is changed after a duration which has already been programmed within the machine parameters. After the programmed **dwell time** has ellapsed, the tool is retracted to the starting position.



Canned cycles
Tapping

Cycle definition

Operating mode ____________ ◈

Dialogue initiation ____________ CYCL DEF ↓

CYCL DEF 2 TAPPING	▶ ENT	Enter cycle into memory.

▼

SET-UP? CLEARANCE?	▶ □	Key-in set-up clearance.
	▼ +/–	Key-in correct sign.
	▼ ENT	Enter into memory.

▼

TOTAL HOLE DEPTH?	▶ □	Key-in thread depth.
	▼ +/–	Key-in correct sign.
	▼ ENT	Enter into memory.

▼

DWELL TIME IN SECS?	▶ □	Key-in dwell time between spindle rotation change-over and rotation.
	▼ ENT	Enter into memory.

▼

FEED RATE? F =	▶ □	Enter calculated feed rate.
	▼ ENT	Enter into memory.

The set-up clearance and the hole depth must have the same arithmetical sign!

Display example

80 CYCL DEF 2.0 TAPPING	The cycle definition "tapping" allocates 5 program blocks
81 CYCL DEF 2.1 SET-UP –2,000	Set-up clearance
82 CYCL DEF 2.2 DEPTH –30,000	Total hole depth
83 CYCL DEF 2.3 DWELL 0,000	Dwell time
84 CYCL DEF 2.4 F 160	Feed rate

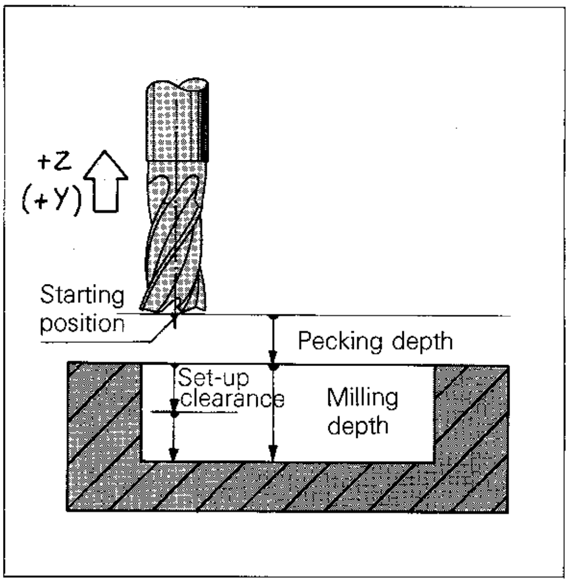
Canned cycles
Slot milling

The cycle

"Slot milling" is a combined rough/fine cut cycle. The slot is parallel to an axis of the current co-ordinate system which may have to be rotated if necessary, (see cycle 10, Co-ordinate system rotation).

Entry data

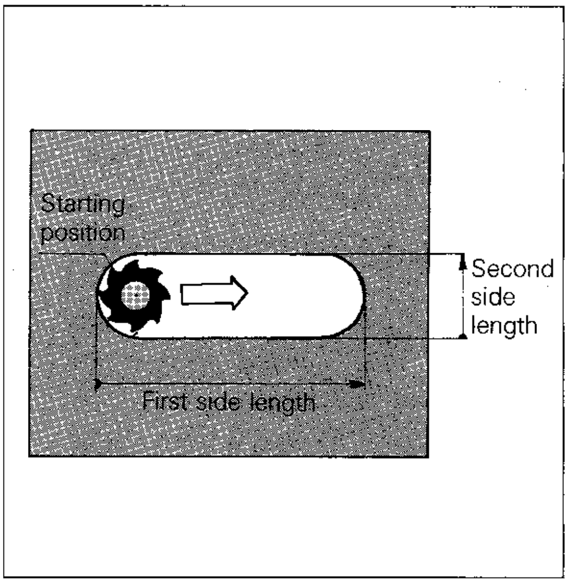
Set-up clearance: see cycle 1
Milling depth (= depth of slot): Distance between workpiece surface and base of slot. Arithmetical sign – see set-up clearance.
Pecking depth: Depth of plunge when penetrating workpiece. Arithmetical sign – see set-up clearance.
Feed rate for pecking: Feed rate when tool penetrates workpiece.



First side length: Finished length of slot. The programmed sign must correspond to the milling direction:
If milling is in the positive direction when commencing from the starting position:
positive sign.
If milling is in the negative direction when commencing from the starting position:
negative sign.
Second side length: Finished slot width. The sign is always positive.

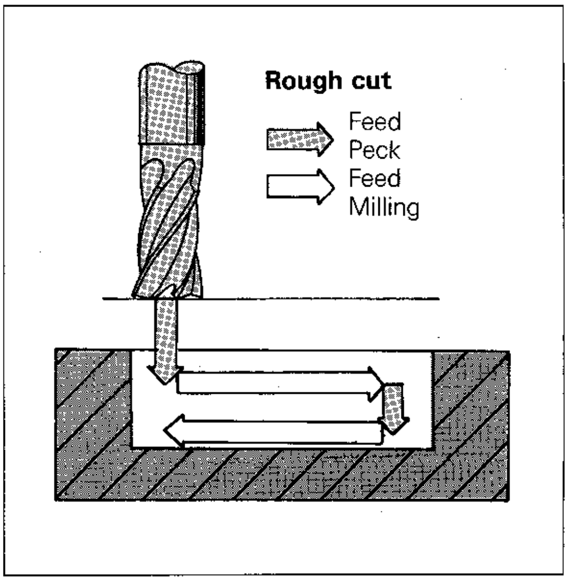
The slot width must always be greater than the mill diameter!

Feed rate: Feed rate of tool motion in the working plane.



Procedure

Rough cut cycle: From the **starting position,** the tool penetrates the workpiece. The slot is then milled in the length direction. After the next peck, the slot is milled in the opposite direction. The procedure is repeated until the programmed **milling depth** is reached.

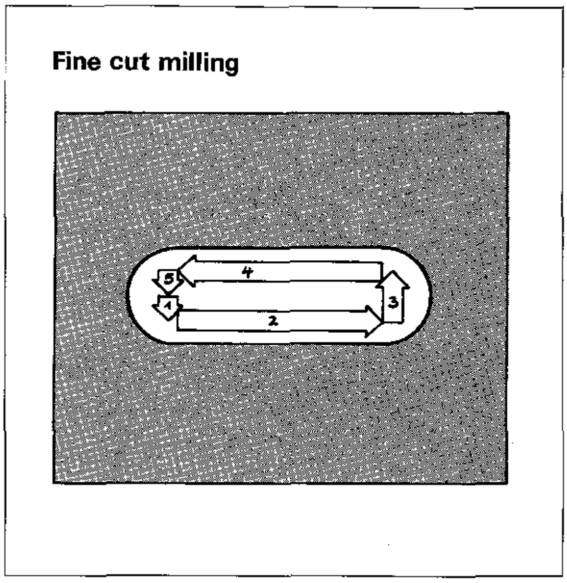


Canned cycles
Slot milling

Procedure

Fine cut milling: The control positions the milling cutter in the transverse direction at the base of the slot for the final finish cut of the contour in **down-cut** milling.
If the number of pecks was odd, the tool returns to the starting position at the set-up height.

Due to the fine cut, a small straight section is formed at the ends of the slot.

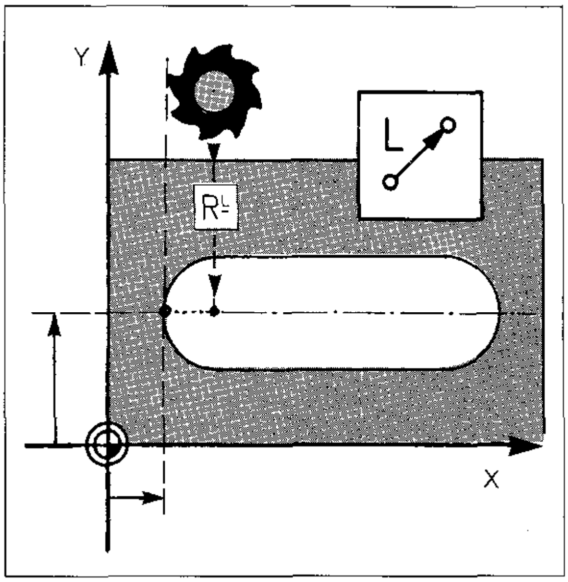

Fine cut milling

Starting position

The starting position for the slot milling cycle must be positioned exactly; taking the tool radius into account.

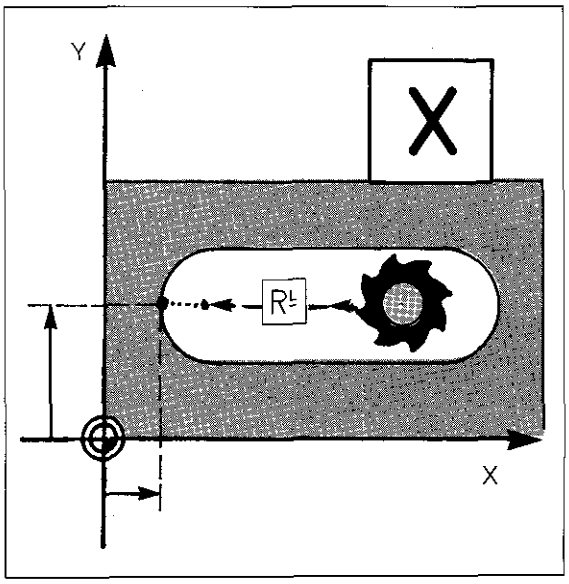
Contour approach with a linear interpolation block

The slot is approached at right-angles to the length direction with tool path offset RL/RR and auxiliary function M98.



Approach with a single axis positioning block

The slot is approached in the length direction with tool radius compensation R–/R+.



Remarks

Canned cycles
Slot milling

Cycle definition

Operating mode ______________________ ◈

Dialogue initiation ______________________ CYCL DEF ↓

CYCL DEF 3 SLOT MILLING	▶ ENT	Enter cycle into memory.

▼

SET-UP CLEARANCE?	▶ □	Key-in set-up clearance.
	▼ +/−	Key-in correct sign.
	▼ ENT	Enter into memory.

▼

MILLING DEPTH?	▶ □	Key-in milling depth.
	▼ +/−	Key-in correct sign.
	▼ ENT	Enter into memory.

▼

PECKING DEPTH?	▶ □	Key-in pecking depth.
	▼ +/−	Key-in correct sign.
	▼ ENT	Enter into memory.

▼

FEED RATE FOR PECKING?	▶ □	Key-in feed rate tool penetration.
	▼ ENT	Enter into memory.

▼

FIRST SIDE LENGTH?	▶ X	Specify axis direction for slot length, e.g. X.
	▼ □	Key-in slot length.
	▼ +/−	Key-in correct sign.
	▼ ENT	Enter into memory.

▼

Remarks

Canned cycles
Slot milling

SECOND SIDE LENGTH?	Y	Specify axis for slot width, e.g. Y.
	□	Key-in slot width with positive sign.
	ENT	Enter into memory.

FEED RATE? F =	□	Key-in feed rate for milling of slot.
	ENT	Enter into memory.

The set-up clearance, milling depth and pecking depth must all have the same sign

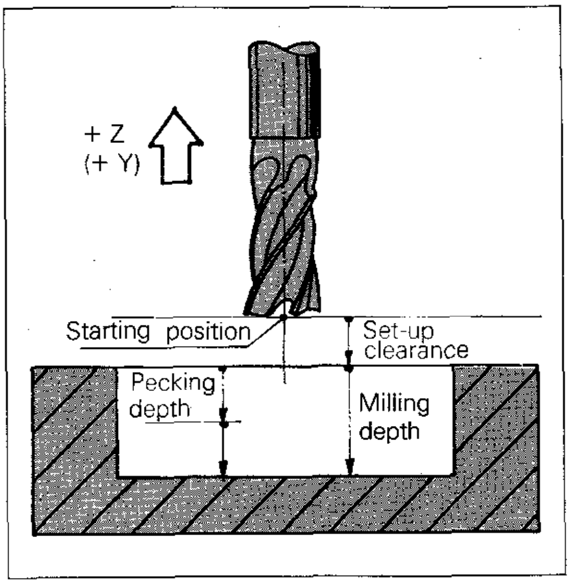
Display example

100 CYCL DEF 3.0 SLOT MILLING	The slot milling cycle allocates 7 program blocks
101 CYCL DEF 3.1 SET-UP −2,000	Set-up clearance
102 CYCL DEF 3.2 DEPTH −40,000	Milling depth
103 CYCL DEF 3.3 PECKING −20,000	Pecking depth
F80	Feed rate for pecking
104 CYCL DEF 3.4 X −120,000	Length of slot
105 CYCL DEF 3.5 Y + 21,000	Width of slot
106 CYCL DEF 3.6 F100	Feed rate

Canned cycles
Pocket milling

The cycle

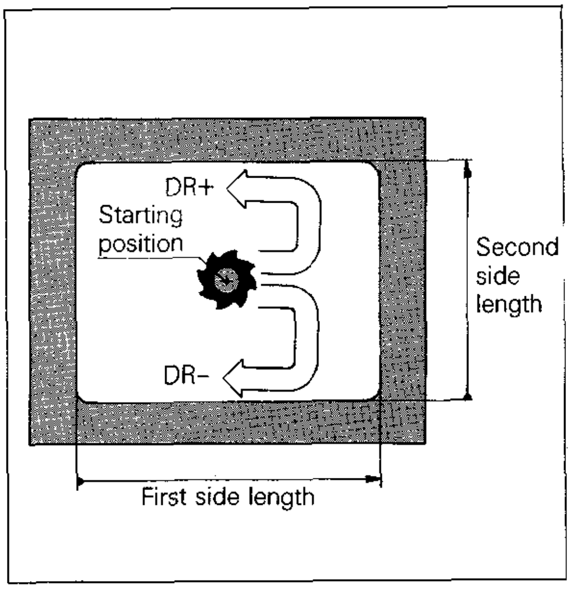
The pocket milling cycle can be performed as a **rough cut** or **fine cut** cycle. Sides of the pocket are located parallel to the axes of the current co-ordinate system. If necessary, the co-ordinate system is to be rotated (see cycle 10 "co-ordinate system rotation")



Entry data

Set-up clearance: see cycle 1
Milling depth: (= depth of pocket): Distance between the workpiece surface and the base of the pocket.
See set-up clearance for sign.
Pecking depth: Penetration depth of tool.
See set-up clearance for sign.

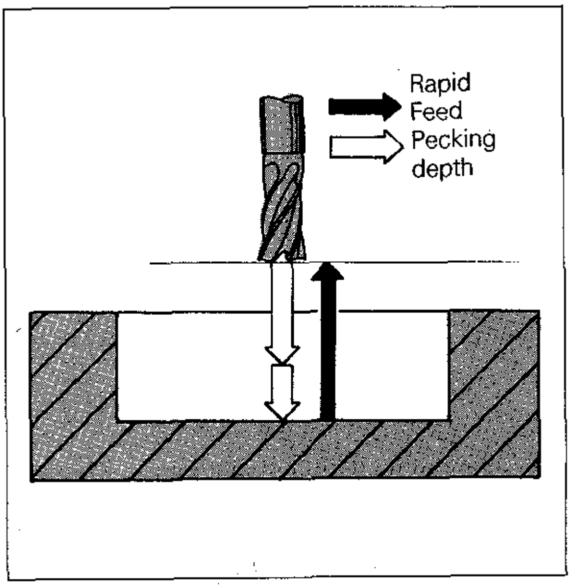
Feed rate for pecking: Feed rate when tool penetrates workpiece.
First side length: Length of pocket parallel to the first main axis in the working plane. The sign is always positive.
Second side length: Width of pocket. The sign is also positive.
Feed rate: Feed rate of tool motion in the working plane.
Rotation: Rotation direction of cutter path.
DR+: positive rotation (counter-clockwise); down-cut milling
DR–: negative rotation (clockwise); up-cut milling



Procedure

The tool penetrates the work at the **starting position** (centre of pocket). The milling tool then follows the path as indicated. The starting direction of the tool path is the positive axis direction of the longest side, i.e. if the longest side is parallel to the X-axis, the tool will move in the positive X-direction.

The corner radii of the rectangular pocket correspond to the tool radius.

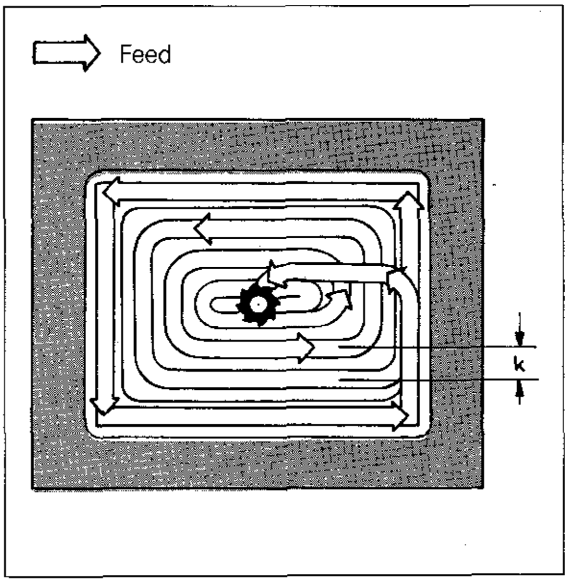


Canned cycles
Pocket milling

Procedure

When milling rectangular pockets, the tool always starts in the positive Y-direction. The rotating direction depends on the **rotation** which has been programmed (here DR+). The stepover distance is always k (or less).

The procedure is repeated until the programmed milling depth is reached. Finally, the tool is retracted to the starting position.



Stepover

The control calculates the stepover k according to the following formula:

$$k = K \times R$$

k: stepover
K: Factor defined by machine tool builder (via machine parameter)
R: Radius of mill

Remarks

Canned cycles
Pocket milling

Cycle definition

Operating mode ________ ◈

Dialogue initiation ________ CYCL DEF ↓

CYCL DEF 4 POCKET MILLING	ENT	Enter cycle into memory.
SET-UP CLEARANCE?	□	Key-in set-up clearance.
	+/−	Key-in correct sign.
	ENT	Enter into memory.
MILLING DEPTH?	□	Key-in milling depth.
	+/−	Key-in correct sign.
	ENT	Enter into memory.
PECKING DEPTH?	□	Key-in pecking depth.
	+/−	Key-in correct sign.
	ENT	Enter into memory.
FEED RATE FOR PECKING?	□	Key-in feed rate for tool penetration.
	ENT	Enter into memory.
FIRST SIDE LENGTH?	X	Specify axis of first side length e.g. X.
	□	Key-in first side length with positive sign.
	ENT	Enter into memory.
SECOND SIDE LENGTH?	Y	Specify axis of second side length e.g. Y.
	□	Key-in second side length with positive sign.
	ENT	Enter into memory.

Remarks

Canned cycles
Pocket milling

FEED RATE? F =	□	Key-in feed rate for milling of pocket.
	ENT	Enter into memory.

ROTATION CLOCKWISE: DR–?	+/–	Key-in tool path rotation.
	ENT	Enter into memory.

Set-up clearance, milling depth and pecking depth must all have the same sign.

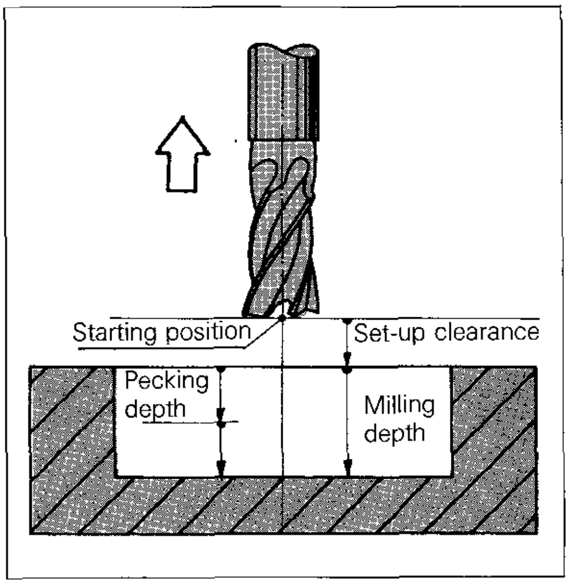
Display example

250 CYCL DEF 4.0 POCKET MILLING	The cycle definition pocket milling allocates 7 program blocks
251 CYCL DEF 4.1 SET-UP –2,000	Set-up clearance
252 CYCL DEF 4.2 DEPTH –30,000	Milling depth
253 CYCL DEF 4.3 PECKING –10,000	Pecking depth
F80	Feed rate for pecking
254 CYCL DEF 4.4 X +80,000	First side length
255 CYCL DEF 4.5 Y +40,000	Second side length
256 CYCL DEF 4.6 F 100 DR+	Feed rate/Path rotation

Canned cycles
Circular pocket milling

The cycle

The circular pocket cycle is a **rough cut** and **fine cut** cycle.

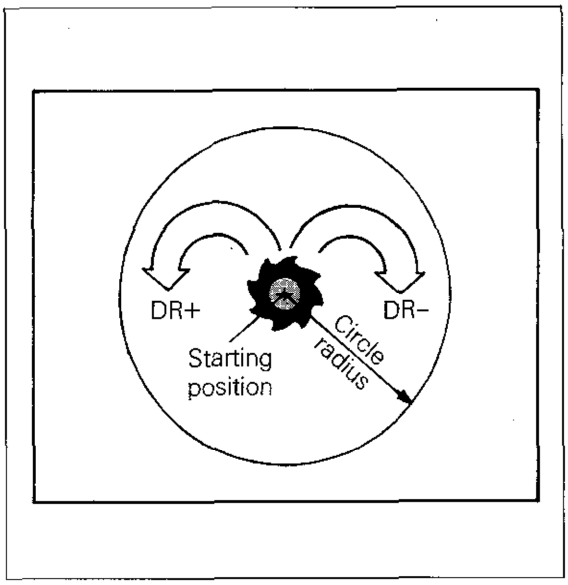


Entry data

Set-up clearance: see cycle 1.
Milling depth: (= depth of pocket): Distance between workpiece surface and base of pocket. See set-up clearance for sign.
Pecking depth: Penetration depth of tool. See set-up clearance for sign.
Feed rate for pecking: Feed rate when tool penetrates workpiece.
Circle radius: Radius of circular pocket.
Feed rate: Feed rate of tool motion in the working plane.

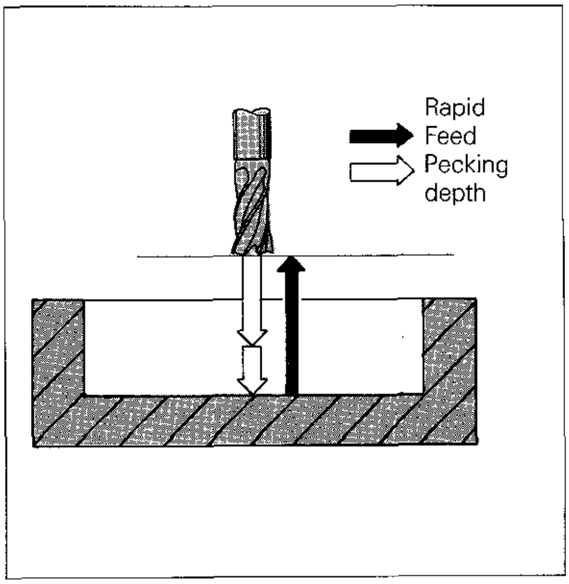
Rotation: Rotating direction of cutter path

DR+: positive rotation (counter-clockwise); down-cut milling
DR−: negative rotation (clockwise); up-cut milling



Procedure

The tool penetrates the work at the **starting** position (centre of pocket). The cutter then follows a spiral-shaped path, the rotation of which, depends on the programmed **rotation** (here DR+).
The starting direction of the cutter is
- the Y+direction for the X, Y-plane
- the X+direction for the X, Z-plane
- the Z+direction for the Y, Z-plane

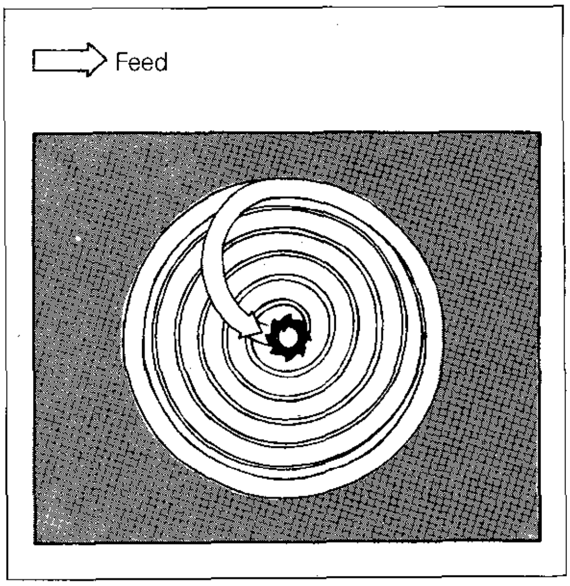


Canned cycles
Circular pocket milling

Procedure

The stepover distance is max. "k" (see cycle "Pocket milling")

The procedure is repeated until the programmed **milling depth** is reached.
Finally, the tool is retracted to the starting position.



Remarks

Canned cycles
Circular pocket milling

Cycle definition

Operating mode ______________ ⇨

Dialogue initiation ______________ CYCL DEF ↓

CYCL DEF 5 CIRCULAR POCKET	ENT	Enter cycle into memory.
SET-UP CLEARANCE?	□	Key-in set-up clearance.
	+/−	Key-in correct sign.
	ENT	Enter into memory.
MILLING DEPTH?	□	Key-in milling depth.
	+/−	Key-in correct sign.
	ENT	Enter into memory.
PECKING DEPTH?	□	Key-in pecking depth.
	+/−	Key-in correct sign.
	ENT	Enter into memory.
FEED RATE FOR PECKING?	□	Key-in feed rate for tool penetration.
	ENT	Enter into memory.
CIRCLE RADIUS?	□	Key-in radius of pocket.
	ENT	Enter into memory.
FEED RATE? F =	□	Key-in feed rate for milling.
	ENT	Enter into memory.

Remarks

Canned cycles
Circular pocket milling

ROTATION CLOCKWISE: DR–?	▶ +/–	Key-in rotation for tool path.
	▼ ENT	Enter into memory.

Set-up clearance, milling depth and pecking depth must all have the same sign.

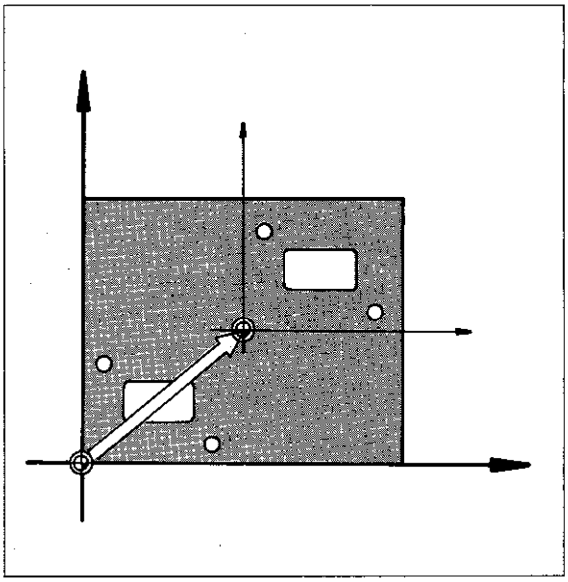
Display example

40 CYCL DEF 5.0 CIRCULAR POCKET	The cycle definition circular pocket allocates 6 program blocks.
41 CYCL DEF 5.1 SET-UP –2,000	Set-up clearance
42 CYCL DEF 5.2 DEPTH –60,000	Milling depth
43 CYCL DEF 5.3 PECKING –20,000	Pecking depth
F80	Feed rate for pecking
44 CYCL DEF 5.4 RADIUS –120,000	Radius of pocket
45 CYCL DEF 5.5 F100 DR–	Feed rate/Rotating direction

Canned cylces
Datum shift

The cycle

This cycle is for displacement of the workpiece datum to another location within the co-ordinate system. Machining procedures such as slot milling or pocket milling can be performed at different locations on the job without having to reprogram.



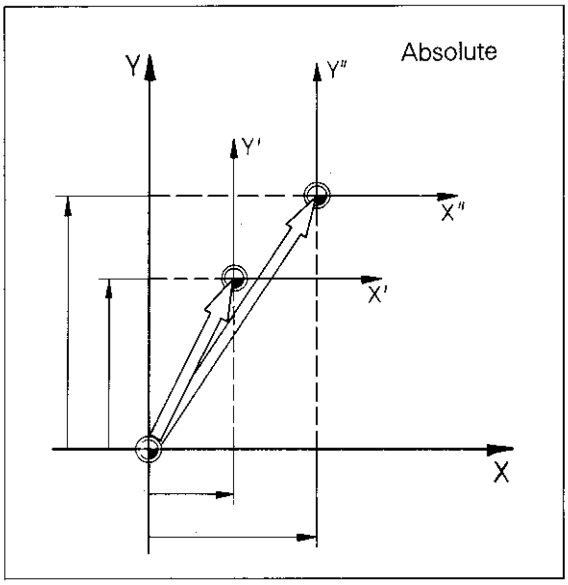
Datum shift

Datum shift, only requires the entry of the new co-ordinates for the datum. The co-ordinate system with its **X, Y, Z** and **IV**-axes is then relocated about the new datum. All subsequent co-ordinate entries are then related to the new datum.

Incremental/ Absolute

Co-ordinates can be entered with the cycle definition as follows:

- **Absolute:** Co-ordinates of the new datum are referenced to the original datum. (= workpiece datum originally set ⊕).

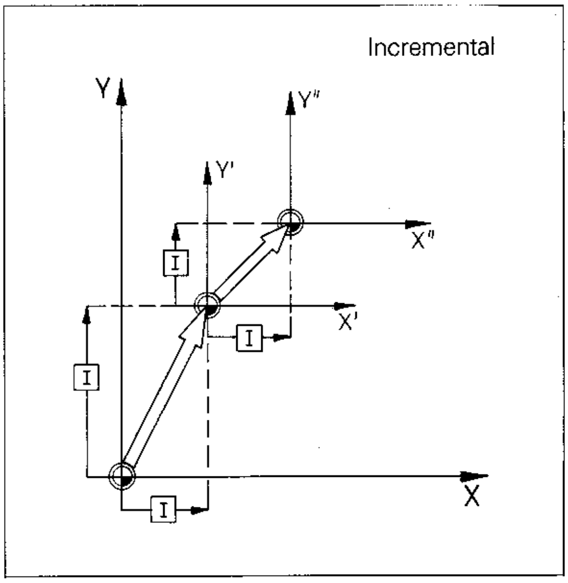


- **Incremental:** Co-ordinates of the new datum are referenced to the datum which was last valid. The last datum can also be a shifted datum.

Cancellation of a datum shift

A datum shift is cancelled as follows:

- Entry of an absolute datum shift with the co-ordinates X 0.000/Y 0.000/Z 0.000/ IV 0.000;
- Entry of auxiliary function M02, M30 or the block END PGM ... MM (depending on the machine parameter entered).



Canned cycles
Datum shift

Cycle definition

Operating mode ______ ⇨

Dialogue initiation ______ CYCL DEF ↑

CYCL DEF 7 DATUM SHIFT	▶ ENT	Enter cycle into memory.

▼

DATUM SHIFT?	▶ X	Select axis, e.g. X
	▼ I	Incremental-Absolute?
	▼ □	Key-in co-ordinates of new datum.
With datum shift, numerical values can be allocated to **all axes X, Y, Z, IV.**	▼ Y	
After keying-in co-ordinates of new datum:	▶ ENT	Enter into memory.

The datum shift cycle is immediately effective after cycle definition. The shift is related to the real datum as displayed in the status display.

Display example

```
10  CYCL DEF  7.0  DATUM SHIFT
11  CYCL DEF  7.1  X +20,000
12  CYCL DEF  7.2  Y +10,000
13  CYCL DEF  7.3  Z +10,000
14  CYCL DEF  7.4  C +90,000
```

The cycle definition "datum shift" allocates 5 program blocks.

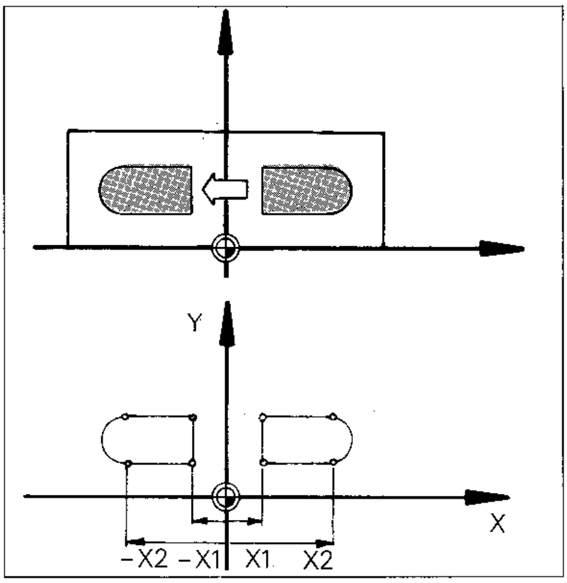
Canned cycles
Mirror image

The cycle

When mirror imaging an axis at the zero datum, the direction of the axis is changed and the arithmetical signs of all co-ordinates are reversed. The result is a programmed contour or hole pattern in a mirror (or reflected) image. Mirror image is only possible in the working plane, either by reversing one axis or both simultaneously.

Mirror image axis

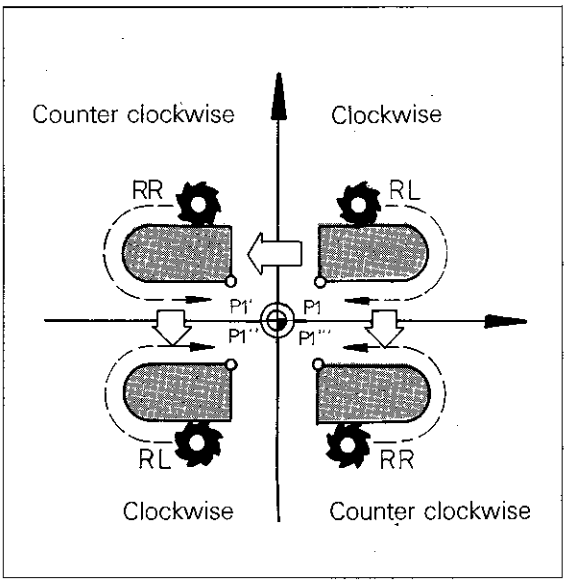
Mirror image programming requires the entry of the axis or axes to be reversed. The co-ordinates of the respective axis are then reversed within the program.
If the tool axis has been inadvertently mirror imaged, the following error message is displayed:
= MIRROR IMAGE ON TOOL AXIS =



Machining direction

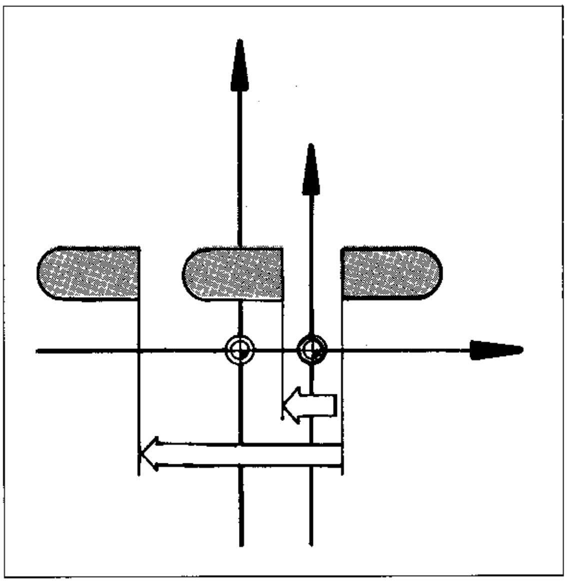
Mirror image in one axis: The machining direction is also reversed when the signs of the co-ordinates have been reversed. If a contour was originally milled in a counter-clockwise direction, the mirror image will affect clockwise milling. The machining direction is, however, maintained for canned cycles.

Mirror image in two axes: The contour which has been mirror imaged in one axis is subjected to further mirror imaging in a second axis. The machining direction is reversed once again, i.e. the original direction therefore remains.



Zero datum

When programming, care should be taken that the co-ordinate axis for mirror imaging lies exactly between the mirrored contour and the contour which is to be mirror imaged. If necessary, a datum shift should be programmed before the cycle definition.



Cancellation of mirror image

The mirror image cycle can be cancelled as follows:

- Entry of the mirror image cycle using NO ENT as the response to the dialogue questions.
- Entry of auxiliary function M02, M30 or the block END PGM...MM (depending on the machine parameter entered).

Canned cycles
Mirror image

Cycle definition

Operating mode ______________ ⬗

Dialogue initiation ______________ CYCL DEF ↑

CYCL DEF 8 MIRROR IMAGE	▶ ENT	Enter cycle into memory.

▼

MIRROR IMAGE AXIS?	▶ X	Key-in axis to be mirror imaged.
If mirror imaging is **simultaneous in two axes:**	▼ Y	Key-in second axis, e.g. Y.
	▼ END □	Enter axes into memory and terminate entry routine.

Entry of axis directions or axes without numerical values must always be terminated with the END □ key.
If axis entry is finalised with ENT, the following error is displayed:
= WRONG AXIS PROGRAMMED =

The mirror image cycle is immediately effective with the cycle definition! The mirror imaged axes are displayed in the status display for datum shift with inverted characters (light background).

Display example

```
120 CYCL DEF 8.0 MIRROR IMAGE

121 CYCL DEF 8.1 X
```

The cycle definition "mirror image" allocates 2 program blocks.

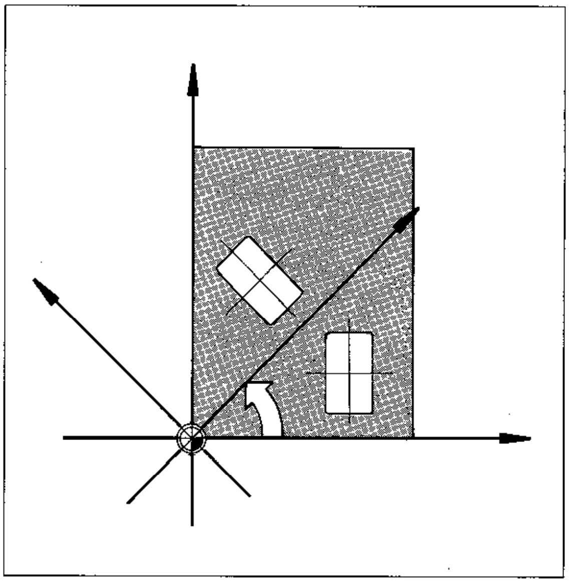
Mirror image axis X.
X-co-ordinates of the subsequent program blocks are reversed.

Canned cycles
Co-ordinates system rotation

The cycle

The co-ordinate system of the working plane can be rotated about the zero datum within a program.

This is convenient e.g. for the milling of repetitive pockets, the sides of which, are not parallel to the original co-ordinate axes.



Rotation angle

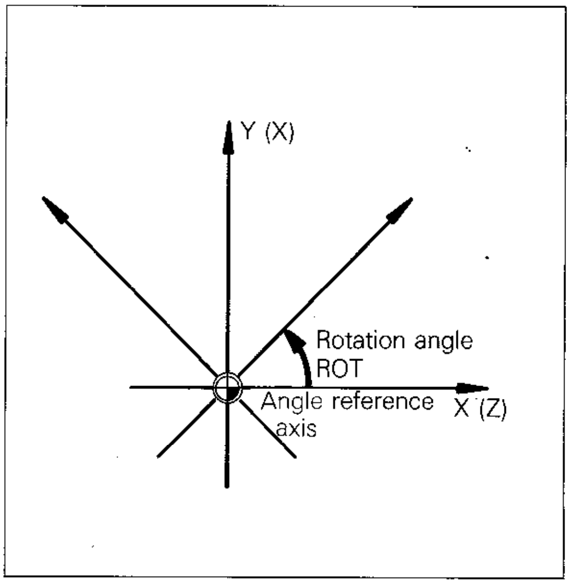
The rotation is entered by programming the **rotation angle ROT.**
The rotation angle is always referenced to the zero datum of the co-ordinate system – the centre of rotation – and the **reference axis** for absolute programming is

- **+ X-axis** for the **X, Y-plane**
- **+ Y-axis** for the **Y, Z-plane**
- **+ Z-axis** for the **Z, X-plane**

All co-ordinate entries which follow the rotation are then referenced to the datum and the rotated co-ordinate system.
The rotation angle may also be entered incrementally.

Entry range

The rotation angle is entered in degrees (°).
Entry range: from – 360° to + 360°



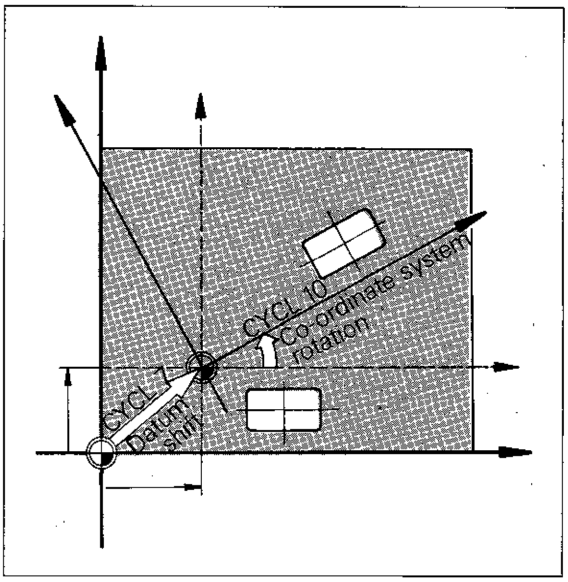
Co-ordinate system rotation and datum shift

The co-ordinate system rotation cycle can be combined with the datum shift cycle by simply programming them consecutively. A simultaneous shift and rotation of the co-ordinate system is therefore made possible.

Cancellation of co-ordinate system rotation

The co-ordinate system rotation cycle can be cancelled as follows:

- Rotation entry with an angle 0° (ROT 0)
- Entry of auxiliary function M02, M30 or the block END PGM... MM (depending on the machine parameter entered).



Canned cycles
Co-ordinate system rotation

Cycle definition

Operating mode ______________________ ⇨

Dialogue initiation ______________________ CYCL DEF ↑

CYCL DEF 10 ROTATION	▶ ENT	Enter cycle into memory.
ROTATION?	▶ □	Key-in rotational angle.
	ENT	Enter into memory.

The co-ordinate system rotation cycle is immediately effective after the definition! The absolute rotary angle is displayed in the status display by ROT ...

Display example

```
184  CYCL DEF  10.0  ROTATION

185  CYCL DEF  10.1  ROT +45,000
```

The cycle definition "co-ordinate system rotation" allocates 2 program blocks

Rotational angle in (°)

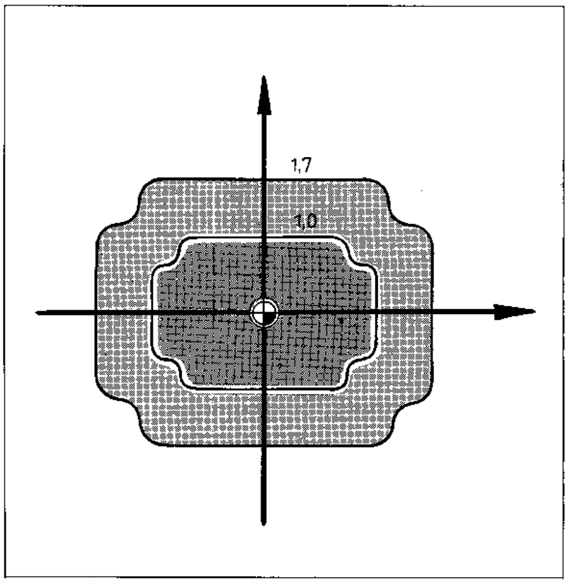
Canned cycles
Scaling

The cycle

Contours within the working plane can be increased or decreased in size.

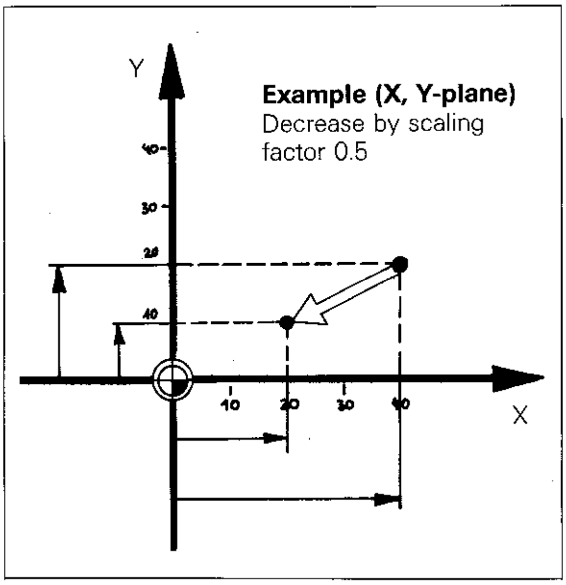
Geometrically similar shapes can be machined without re-programming, and the control can take e.g. shrinkage dimensions into account.

Depending on the machine parameters entered, the scaling factor is effective in the working plane or in the three main axes.
This function can be explained by the machine tool builder.



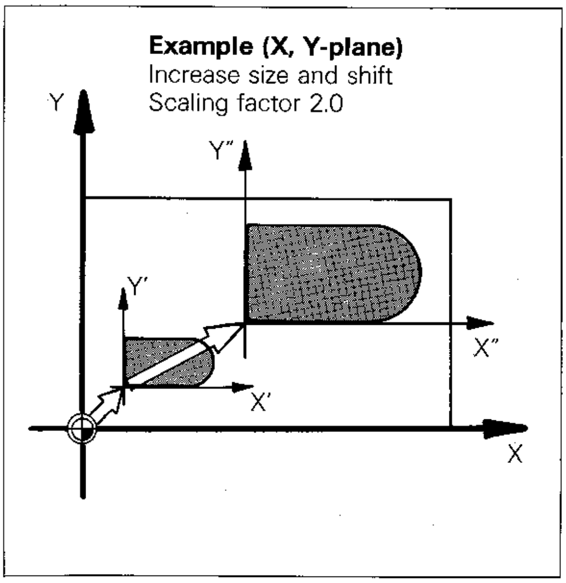
Scaling factor

For increase or decrease in sizes, a scaling factor SCL has to be programmed. The control multiplies all co-ordinates and radii-within the working plane-which are executed with the cycle.
Entry range: 0 to 99.999999


Example (X, Y-plane)
Decrease by scaling factor 0.5

Location of zero datum

With increase or decrease of a contour size, the position of the co-ordinate system datum remains the same. If a scaled contour is required at another location, a datum shift or a co-ordinate system rotation must be programmed beforehand.
Before programming a scaling factor, it is advisable to set the datum at the corner-point of a contour. This saves calculation work.


Example (X, Y-plane)
Increase size and shift
Scaling factor 2.0

Cancelling of the scaling factor

The scaling cycle can be cancelled as follows:
- Entry of a scaling cycle with factor 1.0
- Entry of auxiliary function M02, M30 or the block END PGM...MM (depending on the machine parameter entered).

Canned cycles
Scaling

Cycle definition

Operating mode ______________________ ⇾

Dialogue initiation ______________________ CYCL DEF ↑

CYCL DEF 11 SCALING	▶ ENT	Enter cycle into memory.

▼

FACTOR?	▶ □	Key-in scaling factor.
	▼ ENT	Enter into memory.

The scaling cycle is immediately effective after the definition! The scaling factor is displayed in the status display by SCL.

Display example

```
12  CYCL DEF  11.0  SCALING

13  CYCL DEF  11.1  SCL 0,750000
```

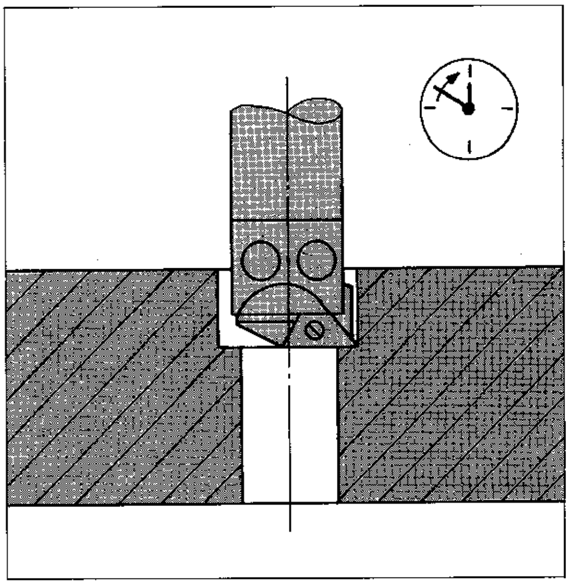
The scaling cycle allocates 2 program blocks.

By entering the scaling factor 0.75, all subsequent dimensions are decreased in size by 0.75.

Canned cycles
Dwell time

The cycle

A dwell time can be used within a program to pause the feed whilst the spindle is still running e.g. for chip breaking during internal boring. The dwell time cycle is performed immediately after the cycle definition.



Entry range

The dwell time is entered in seconds.
Entry range: 0.000 s – 19999.999 s

Entry of 19999.999 corresponds to a working pause of 5.5 hours!

Canned cycles
Dwell time

Cycle definition

Operating mode ______________ ⇨

Dialogue initiation ______________ CYCL DEF ↑

CYCL DEF 9 DWELL TIME	▶ ENT	Enter cycle into memory.
DWELL TIME IN SECS.	▶ □	Key-in reqd. dwell time.
	ENT	Enter into memory.

The "dwell time" cycle is immediately executed after the definition!

Display example

```
97  CYCL DEF  9.0  DWELL TIME

98  CYCL DEF  9.1  DWELL 10,000
```

The dwell time cycle allocates 2 program blocks.

Canned cycles
Freely programmable cycles (Program call)

The cycle

The "program call" cycle permits simple call-up of programs (with CYCL CALL M89 and M99) which have been compiled with the aid of parameter functions, e.g. zig-zag milling. These freely programmable cycles therefore have the same status as pre-programmed canned cycles.

Canned cycles
Freely programmable cycles (Program call)

Cycle definition

Operating mode ______________ ◈

Dialogue initiation ______________ CYCL DEF ↑

CYCL DEF 12 PGM CALL	▶ ENT	Enter cycle into memory
PROGRAM NUMBER?	▶ □	Key-in program number
	▼ ENT	Enter into memory

Display example

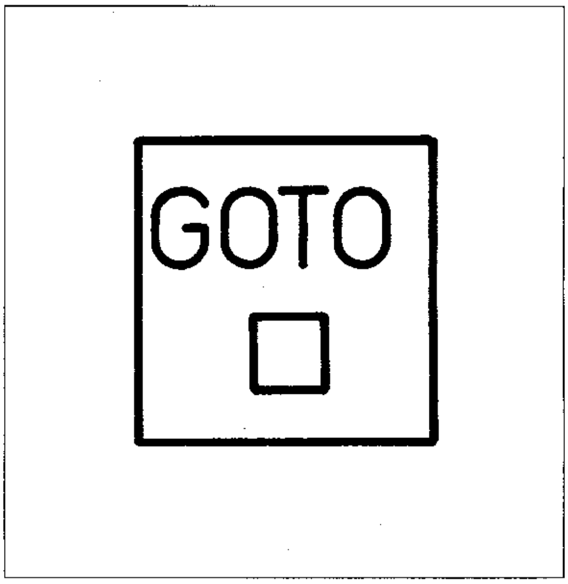
```
5 CYCL DEF 12.0 PGM CALL

6 CYCL DEF 12.1 PGM 23
```

The cycle which has been called-up is programmed within program number 23.

Programm editing

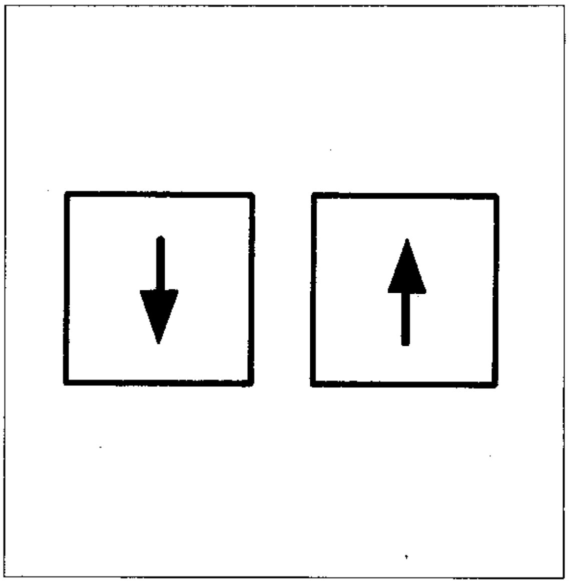
Editing

Editing deals with the checking, amendment and extension of a program.
Editing functions permit easy search and correction of program blocks and words via simple key-in.



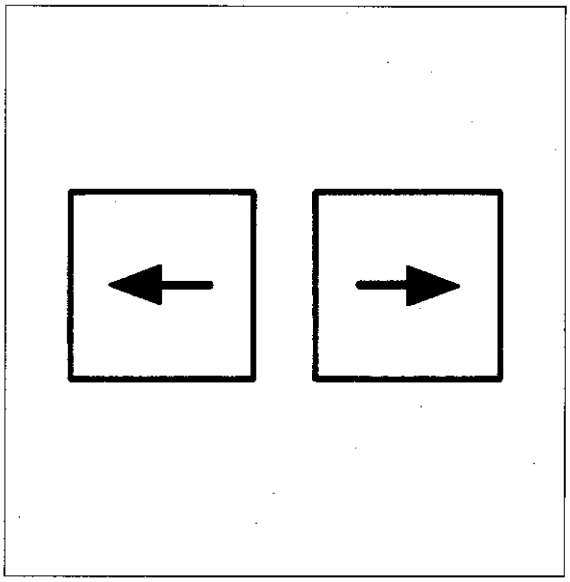
Block call-up

A certain block is addressed with the [GOTO □]-key.



Program paging

Block-to-block paging is performed with the [↓] [↑] -keys.
[↑] -key: Jump to next lowest block number
[↓] -key: Jump to next highest block number



Block word editing

The [←] [→]-keys are used for setting a **cursor** in reversed video – within the current block.
The cursor is set to the word which is to be edited.

The cursor can only be operated in the [◇]-mode.
Cursor operation must be started with the [→]-key!

Program editing
Block call-up

Block call-up

Operating mode ______ ◈ ⇛ ⇥ ⇢

Dialogue initiation ______ GOTO □

GOTO: NUMBER =	□	Key-in block number.
	▼ ENT	Enter into memory.

Editing block words

Operating mode ______ ◈

A word within the current block is to be edited:	▶ →	Set cursor to word for editing.

▼

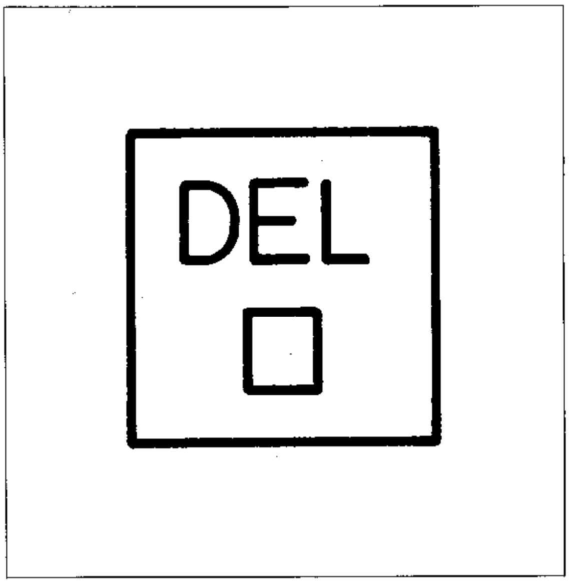
A dialogue question is displayed e.g.

COORDINATES?	▶ X ▼ □	Edit entry value.
If editing is finalised:	▶ END □	Enter block into memory (or shift cursor-out of screen-to the right or left).
If a further word is to be edited:	▶ ← →	Set cursor to word for editing.

Program editing
Deletion and insertion of blocks

Block deletion

The current block within a program can be erased by pressing DEL □.
DEL = abbreviation for "delete"
Block deletion is only possible in the ⇨-mode.



When erasing single blocks, care should be taken that only the current block is being erased. It is advisable to call-up the block by its number.

After deletion, the block with the next-lowest block number shifts into the location of the erased block.
Subsequent block numbers are automatically shifted.

Cycle definition or part program deletion

When deleting a cycle definition or a program part, the last block of the definition or program part is called-up. The DEL □-key is then pressed repeatedly until all the blocks of the definition or program part have been erased.

Inserting a block

New blocks can be inserted at any desired location within the program. Only the block which **immediately follows** the location of insertion is called-up. Subsequent block numbers are automatically shifted.

If the storage capacity of the program memory is exceeded, the following error is displayed:
= PROGRAM MEMORY EXCEEDED =
This error also appears if it is attempted to insert a block subsequent to the END-block of the program (Program end is shown in the current block).

Editing during programming

Entry errors during programming can be amended in three ways:

CE Entry value is erased and "0" appears.

NO ENT The entry value is completely erased.

Program amendments
Block deletion

Deleting a block

Operating mode ______________________ ◈

The current program block is to be deleted.	▶ DEL □	Press for deletion

Program editing
Search routines
Clear program

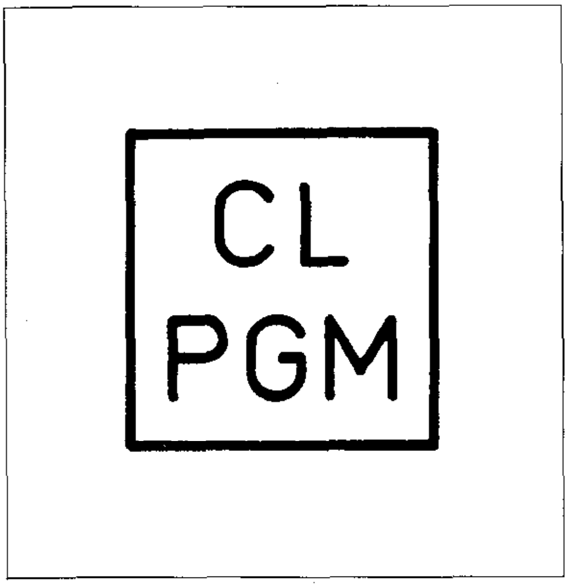
Searching for certain addresses

Blocks containing certain addresses can be easily found by using the paging keys ↓ ↑. The cursor is set to the word with the search address with → or ← and the program is paged with the ↓ and/or ↑ -key. Only those blocks are displayed which contain the word address being searched for.
The search routine is only possible in the ◈ -mode.

Clearing a complete program

The CL PGM -key initiates the dialogue for clearing the program.
On pressing this key, the program menu is displayed in inverted characters. The cursor can be shifted with the ↓ ↑ ← → -keys.

Only the program to which the cursor has been set, can be erased.



Program editing
Search routines
Clear program

Finding certain search addresses

Operating mode ______________ ⇨

All blocks containing the address M are to be displayed:

▶ ↓ ↑ Select a block containing the search address

▼ → Set cursor to a word with the search address.

▼

AUXILIARY FUNCTION M?

▶ ↓ ↑ Call-up blocks containing the search address.

☞ Setting of the cursor is initiated with the → -key.

Clearing a program

Operating mode ______________ ⇨

Dialogue initiation ______________ CL PGM

CLEAR PROGRAM = ENT/END = NOENT

If the program is to be cleared:

▶ ↓ ↑ ← → Set cursor to program number.

▼ ENT Clear program.

Do not clear program or terminate clear program-routine.

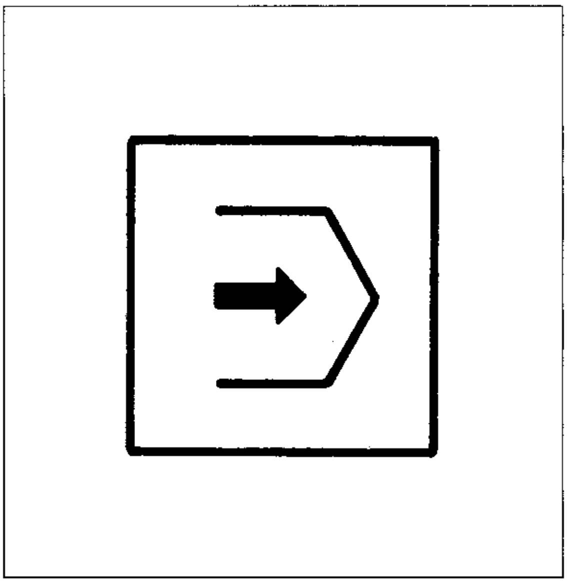
▶ NO ENT

Program Test

Program test

Before machining, the program can be subjected to a test for geometrical errors, without machine movement. The control calculates the program sequence as per a normal program run.
The program test is interrupted with an error message.

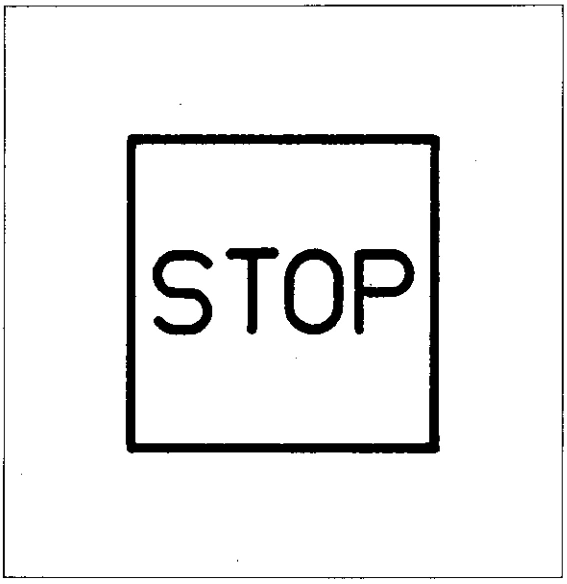
The ⇨-mode key initiates the program test.



Stopping the test run

A test run can be stopped at any point by pressing STOP.

A program test run automatically stops at every programmed stop. Continuation of test run must then be re-started (see next page).



Program test

Starting a program test

Operating mode ________________________ [→)]

TO BLOCK NUMBER =

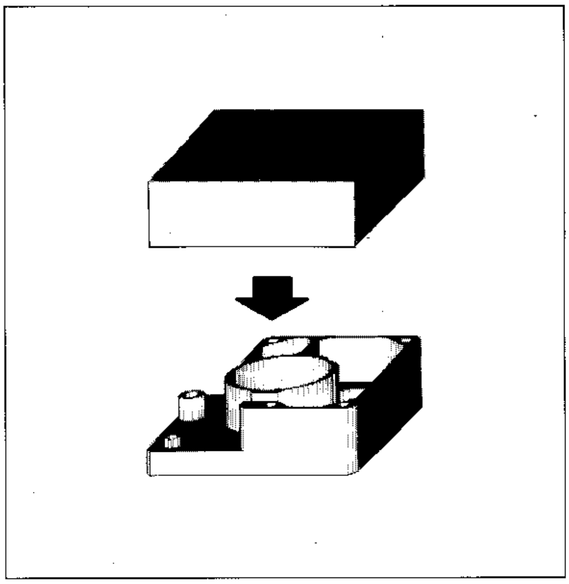
Test to be executed until a certain block number:	▶ □	Key-in block number..
	▼ ENT	Enter into memory.
Test complete program.	▶ NO ENT	

Graphics*
Blank form definition

Graphic image

Machining programs can be graphically simulated on the VDU-screen. For checking the machining program, the production of a workpiece can be displayed. The machine remains stationary during graphics display.

The workpiece blank is always displayed as a cuboid (if this is not the case, the workpiece-blank has to be programmed separately).



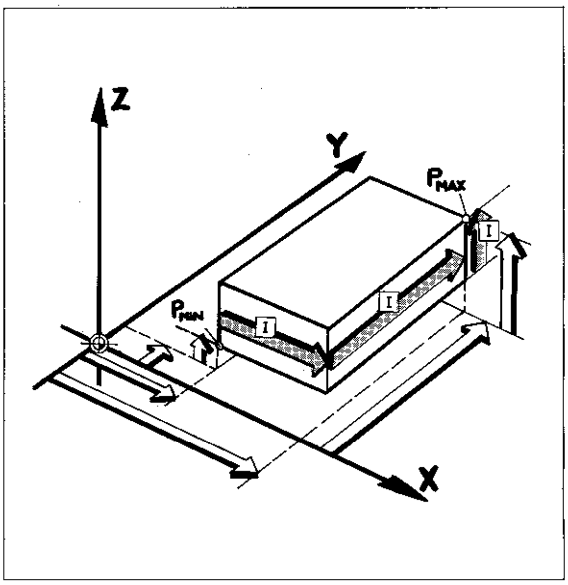
Workpiece milling can be simulated in the three main axes – with the same tool axis – using a cylindrical shaft end mill. Simulation of helical interpolation is not possible.

Definition of the "blank form"

To obtain a workpiece image in graphics, the shape of the blank form must be defined, i.e.

- its **position in relation to the co-ordinate system**
  and
- the programming of its **dimensions.**

The specification of **two corner positions** is sufficient for definition of the cuboid. These are referred to as the minimum point (PMIN) and maximum point (PMAX) (points with the minimum and maximum co-ordinates).
PMIN may only be entered in absolute dimensions, whereas PMAX may be entered either absolute or incremental!



The blank form data is stored within the appropriate machining program and is available when the program is selected.
Definition of the cuboid is advisable **at the beginning** of the program. This enables the BLK FORM-blocks to be found more rapidly when changing the sizes of the blank.
Dialogue, is initiated with the BLK FORM-key.

BLK FORM = abbreviation for BLANK FORM (initial shape of the blank)

The maximum overall dimensions of the blank are 14000 mm x 14000 mm x 14000 mm.

*The graphics feature is only available with the TNC 155-versions.

Graphics
Cuboid corner points – BLANK FORM

Entry of corner points

Operating mode ______________ ⇨

Dialogue initiation ______________ BLK FORM

WORKING SPINDLE AXIS X/Y/Z?	▶ Z	Key-in spindle axis.

▼

DEF BLK FORM: MIN-CORNER?	▶ □	Key-in numerical value for X-co-ordinate.
	ENT	Enter into memory.
	□	Key-in numerical value for Y-co-ordinate.
	ENT	Enter into memory.
	□	Key-in numerical value for Z-co-ordinate.
	ENT	Enter into memory.

▼

DEF BLK FORM: MAX-CORNER?	▶ I	Incremental/Absolute?
	□	Key-in numerical value for X-co-ordinate.
	ENT	Enter into memory.
	I	Incremental/Absolute?
	□	Key-in numerical value for Y-co-ordinate.
	ENT	Enter into memory.
	I	Incremental/Absolute?
	□	Key-in numerical value for Z-co-ordinate.
	ENT	Enter into memory.

Display example

```
1 DEF BLK FORM MIN  X+  0,000
           Y+   0,000 Z-15,000
2 DEF BLK FORM MAX  X+80,000
           Y+100,000 Z+ 0,000
```

The blank is positioned parallel to the main axes.

P_{MIN} has the co-ordinates X0, Y0, Z – 15.

P_{MAX} has the co-ordinates X80, Y100, Z0.

Graphics
Image projections

Operating mode Graphics

A machining program can be graphically displayed in the operating modes.

⇨ PROGRAM RUN/FULL SEQUENCE

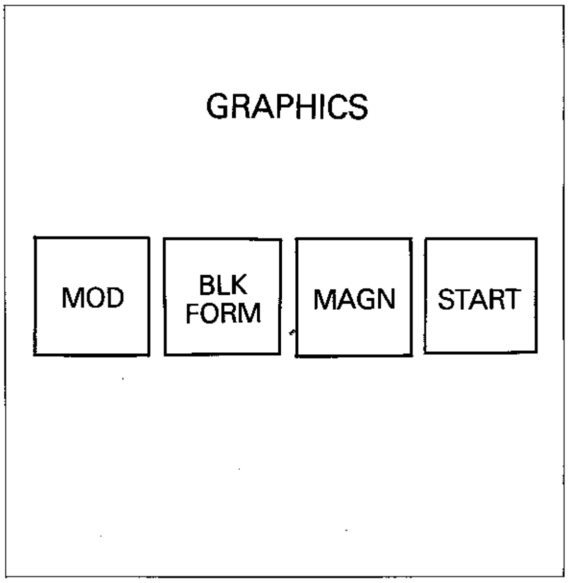
and

⊟ PROGRAM RUN/SINGLE BLOCK

Graphics display is only possible if the program is stored within the memory.

A menu showing the types of image projections which are available is called-up by pressing the MOD-key **twice.**

The ↑ ↓-keys are used for setting the cursor to the required projection mode. Press ENT for transfer into memory.



GRAPHICS

MOD | BLK FORM | MAGN | START

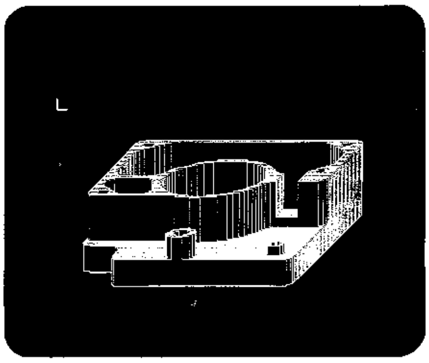
Image projection

These are four types of image projection.

3D-View

Program execution is display in a three-dimensional image. The workpiece can be rotated about the vertical axis by pressing → ← or tilted about the horizontal axis by pressing ↑ ↓.

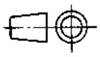
The attitude of the co-ordinate system is indicated by an angle in the top left-hand corner of the screen (working plane).

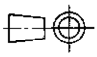


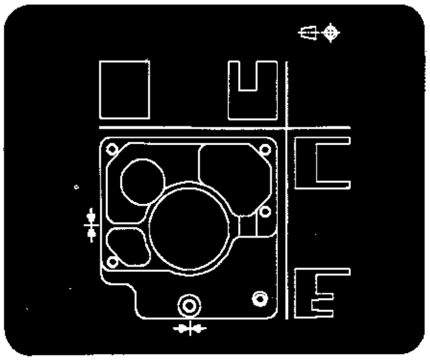
View in three planes

Program execution is displayed with a plan view and two cross-sections, similar to a working drawing. The sectional planes can be shifted with the → ← ↑ ↓-keys.

As of software version 06:
The view in three planes can be switched from the standard DIN-projection to the U.S.-standard third angle projection. A symbol to DIN 6 indicates the projection as follows:

DIN-standard 

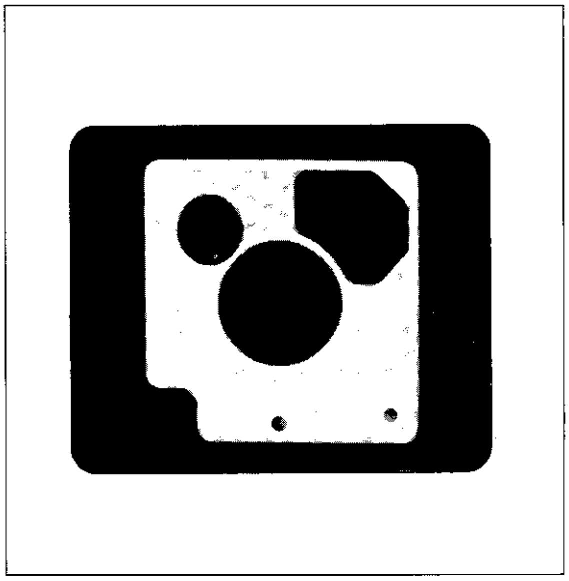
U.S.-standard 



Graphics
Image projections

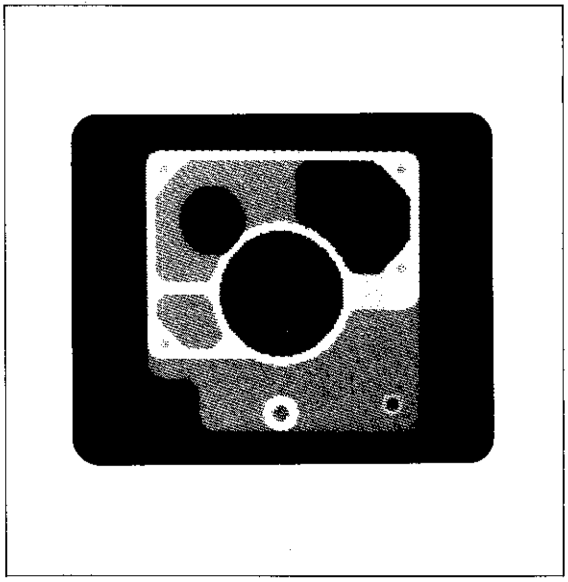
Plan view 1

The program is executed in a plan view with **5 grades of depth shading.** The darker – the deeper.



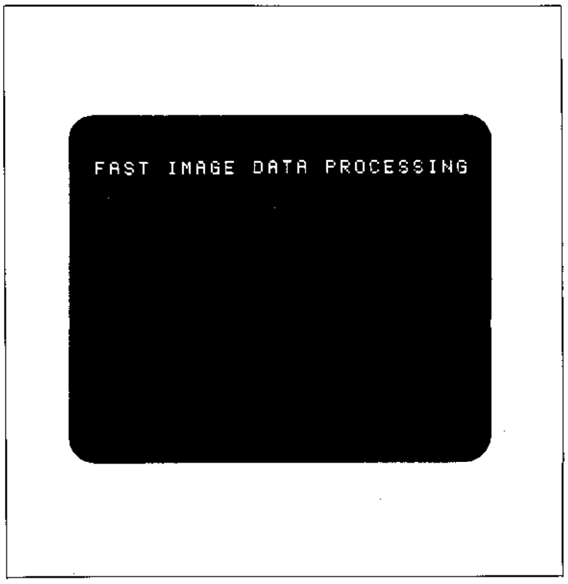
Plan view 2

As per plan view 1, however with **17 grades of depth shading.** The image resolution in the other two axes is however, less superior.



Fast image generation

A finished workpiece can be displayed on the screen after **fast image data processing.** The control "develops" the workpiece in accordance with the program without displaying the various stages of progress.
Only the block number is displayed.



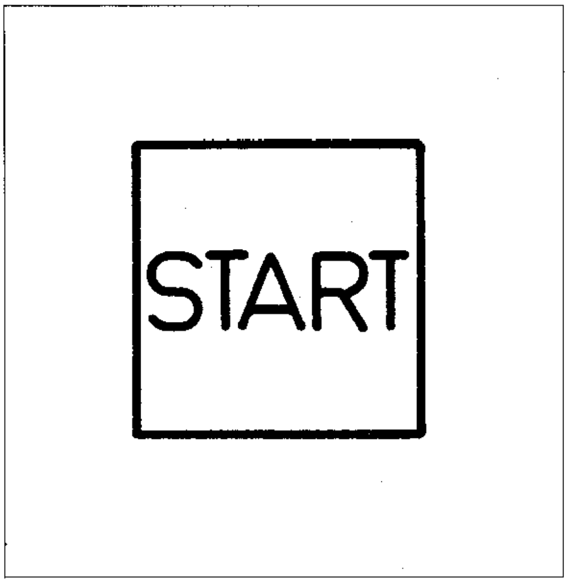
FAST IMAGE DATA PROCESSING

Graphics
Operation

Start

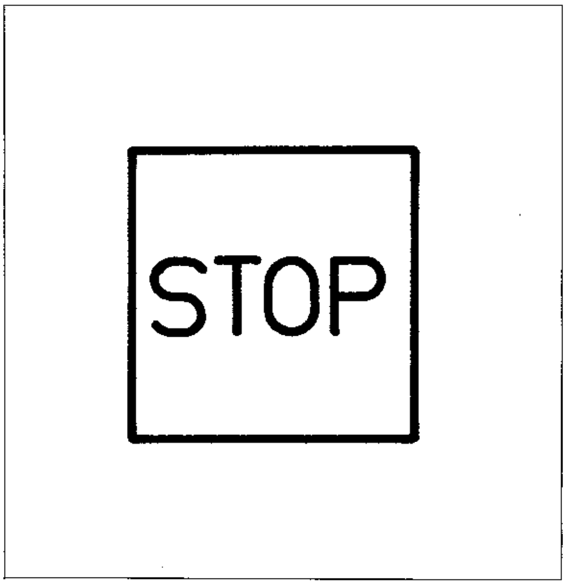
After selecting the required graphics mode, program run is started by pressing START.

Prior to the first axis movement a tool call must be programmed to define the tool axis. Specification of the spindle axis in the BLK FORM definition is insufficient for graphics. If the tool axis is missing, the error = PGM-SECTION CANNOT BE SHOWN = is displayed after the graphics start. This error is also displayed if a fourth axis or helical interpolation was programmed.



Stop

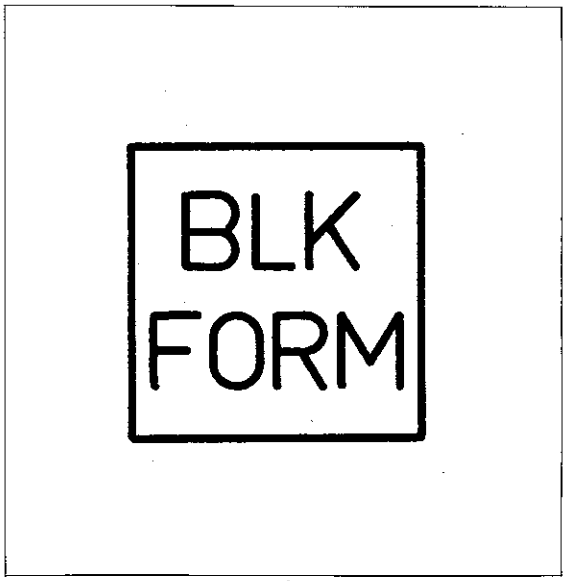
Graphics simulation can be stopped at any time by pressing STOP. The current block is however, completed.



Reset to blank form

After stopping graphics program run, the displayed workpiece can be reset to the blank form (original cuboid) by pressing BLK FORM.

If workpiece production is to be re-simulated graphically, a jump should be made to the beginning of the program by pressing GOTO □.



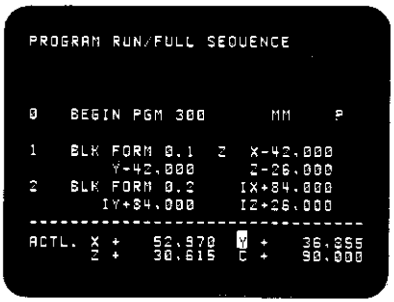
Graphics
Graphics start

Starting of graphics

Operating mode ______ ⇥ or ⇥

Select required program.

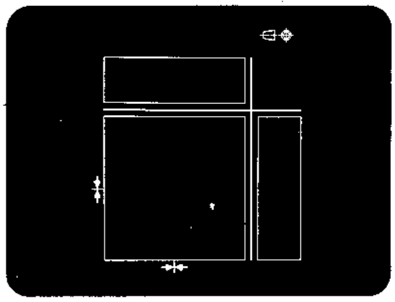
PGM NR
□
ENT



```
PROGRAM RUN/FULL SEQUENCE


0    BEGIN PGM 300          MM    P

1    BLK FORM 0.1  Z   X-42.000
         Y-42.000      Z-26.000
2    BLK FORM 0.2     IX+84.000
        IY+84.000     IZ+26.000
--------------------------------------
ACTL. X +    52.970  Y +     36.855
      Z +    30.615  C +     90.000
```

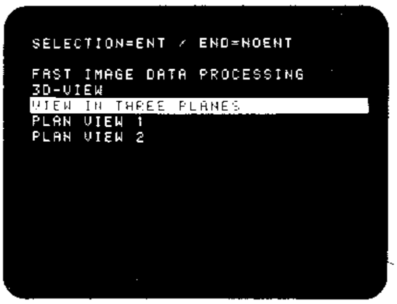
Select graphics mode.

MOD



Call graphics menu.

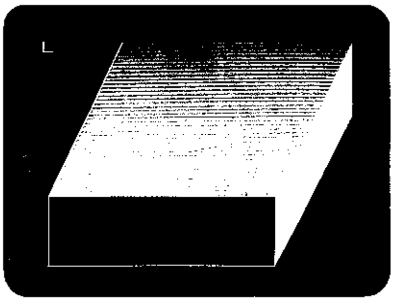
MOD



```
SELECTION=ENT / END=NOENT

FAST IMAGE DATA PROCESSING
3D-VIEW
VIEW IN THREE PLANES
PLAN VIEW 1
PLAN VIEW 2
```

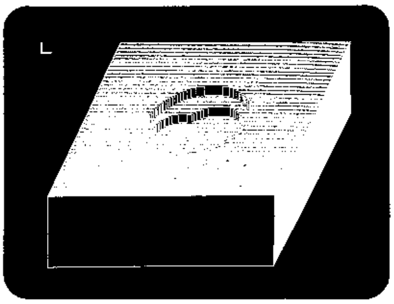
Set cursor to required projection, e.g. 3D-view

↓ ↑
ENT



Start graphics sequence.

START



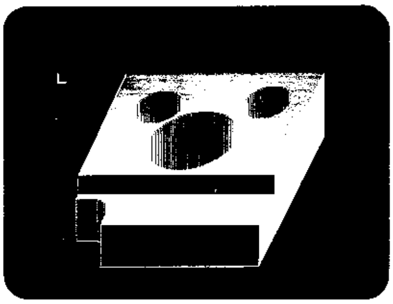
Remarks

Graphics
Graphics start

Graphics stop/ start

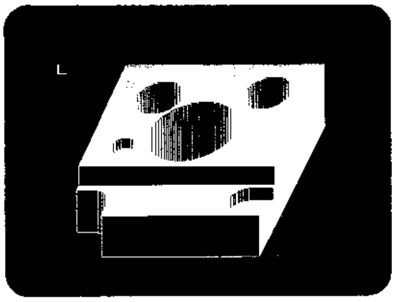
Stop graphics sequence.

STOP



Re-start graphics sequence.

START



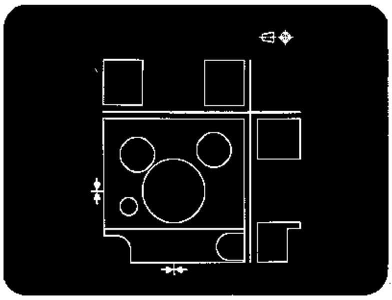
Remarks

Graphics
View in three planes

Shifting the sectional planes

Stop graphics sequence.

▶ STOP ▶



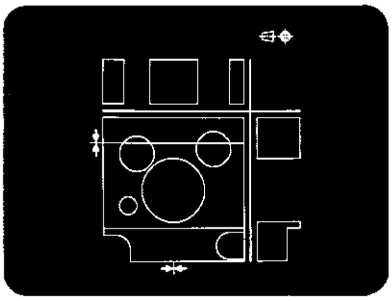
Shift horizontal section cursor
e.g. upwards:

Either press ↑ several times

or move section cursor continuously.

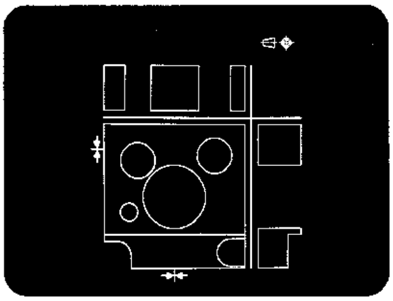
Repeated pressing of the ENT -key enables faster shifting of the section cursor.

▶ ↑ ▶
ENT
ENT



Stop shifting.

▶ STOP ▶

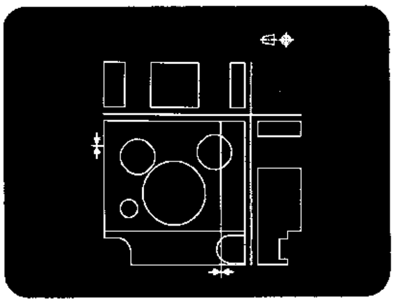


Shift vertical section cursor e.g.
towards the right:

Either press → several times or move section cursor continuously.

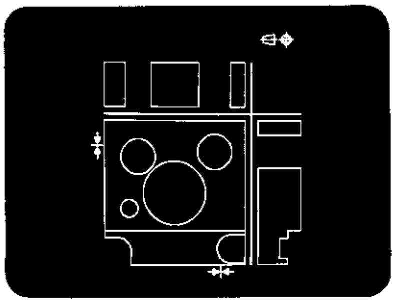
Repeated pressing of the ENT -key enables faster shifting of the section cursor.

▶ → ▶
ENT
ENT



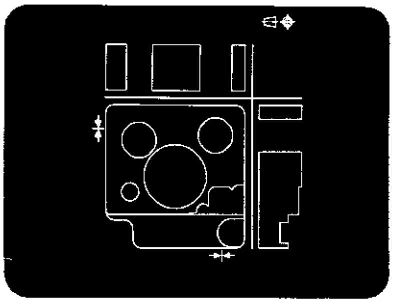
Stop shifting.

▶ STOP ▶



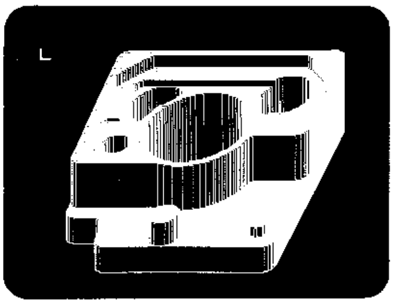
Graphics
View in three planes

Start graphics sequence ▶ START ▶



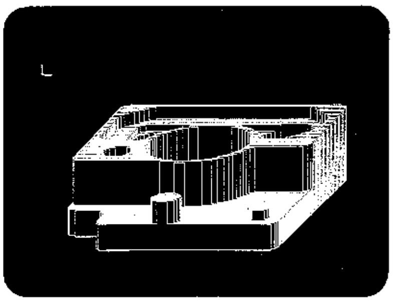
Rotation and tilting

Stop graphics sequence. ▶ STOP ▶



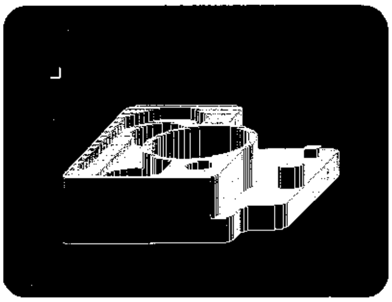
▼

Tilt image, e.g. upwards. ▶ ↓ ▶



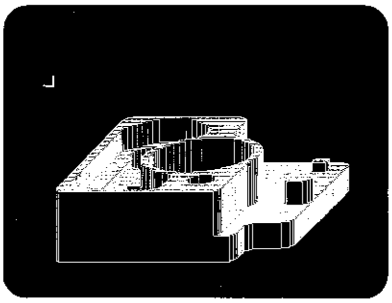
▼

Rotate image, e.g. to the right. ▶ ← ▶



▼

Start graphics sequence. ▶ START ▶

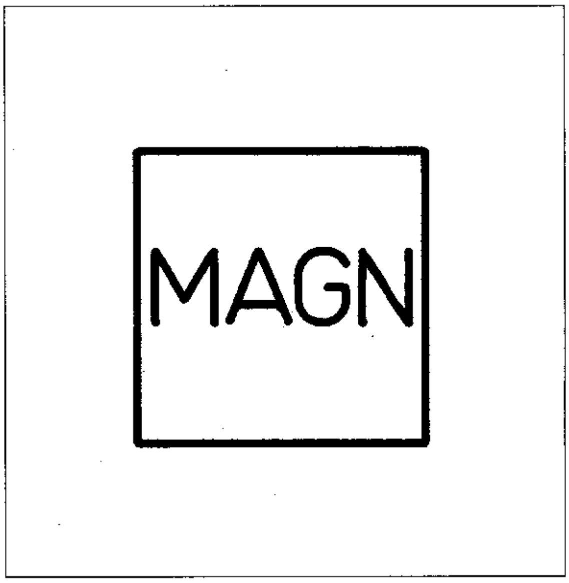


Graphics
Magnify

Magnifying function

The magnifying function is used for enlarging any desired detail of the workpiece.

Definition of the detail to be magnified is only possible in the 3D-graphics mode

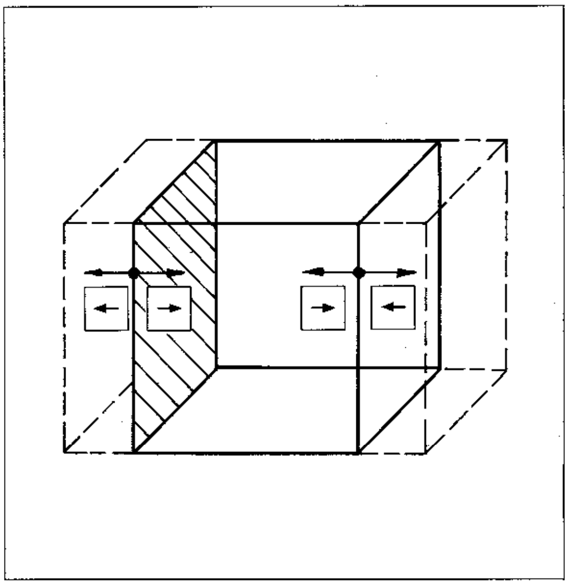


Limitation of workpiece detail

A workpiece detail is limited by means of cuboid frame which appears in the top left-hand corner of the screen after pressing MAGN .

The hatched face can be shifted left and right (or forwards/reverse, upwards/downwards) with the → ← -keys.

Continuous shifting is performed with the ENT - key and stopped by pressing STOP .



Definition of next limit

The next limit (right-hand face) is defined with ↑ .

By doing this, all faces can be selected and shifted one at a time.

The ↓ -key enables a return-jump to the previous face.

Entry of selected detail

After defining the last limiting face (upper face), the detail can be entered by re-pressing ↑ and finally ENT .

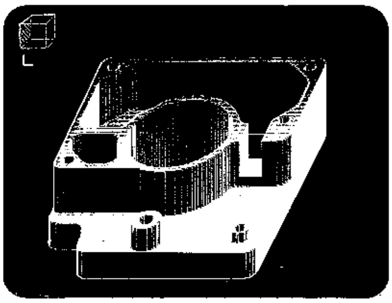
The display then shows an enlarged cuboid blank of the detail. The magnified detail, complete with contour is obtained with a graphics run start in any one of the normal graphics modes.

Graphics
Magnify

Limitation of detail and magnification

Control is in the 3D-graphics mode.

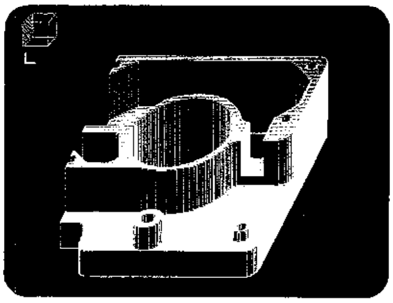
Select magnify-function. ▶ MAGN ▶



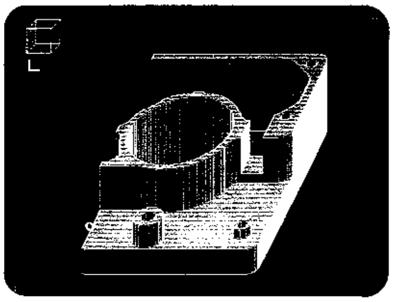
Shift left-hand limiting face e.g. to the right:

Either press → repeatedly (jogging) ▶ → ▶

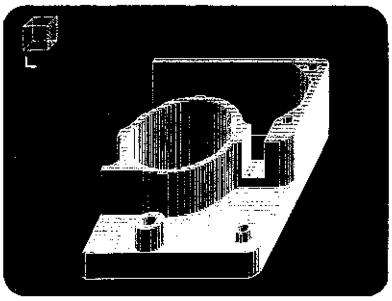
By pressing ENT several times, shifting is accelerated. ENT ▼ ENT



Stop shifting and enter ▶ STOP ▶

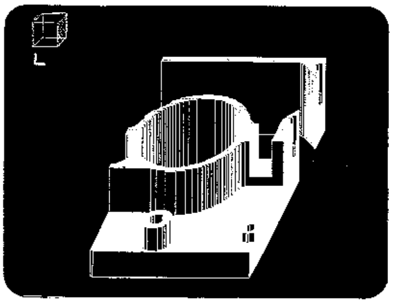


Select next limiting face (right) ▶ ↑ ▶



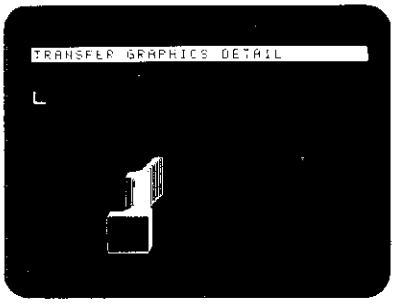
Graphics
Magnify

Shift this face and remaining faces as explained above.



Once the last face, has been shifted (upper face):

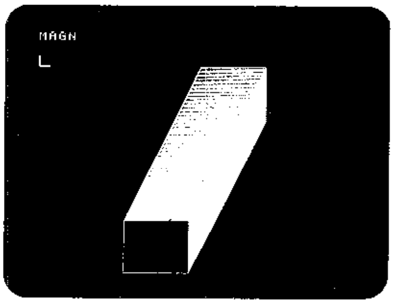
▶ ↑ ▶



TRANSFER GRAPHICS DETAIL

TRANSFER GRAPHICS DETAIL

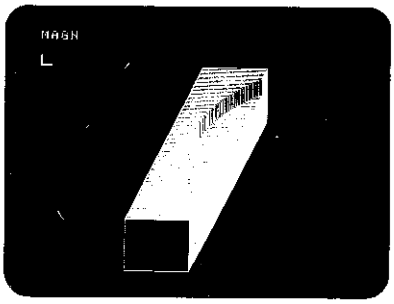
▶ ENT ▶



MAGN

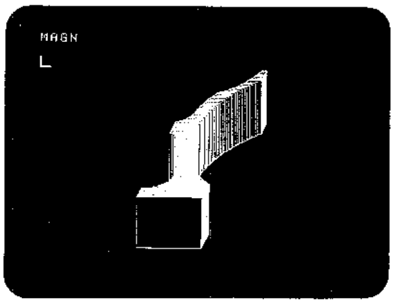
Start program run

▶ START ▶



MAGN

Workpiece execution is simulated. The display only shows the detail which has been defined.



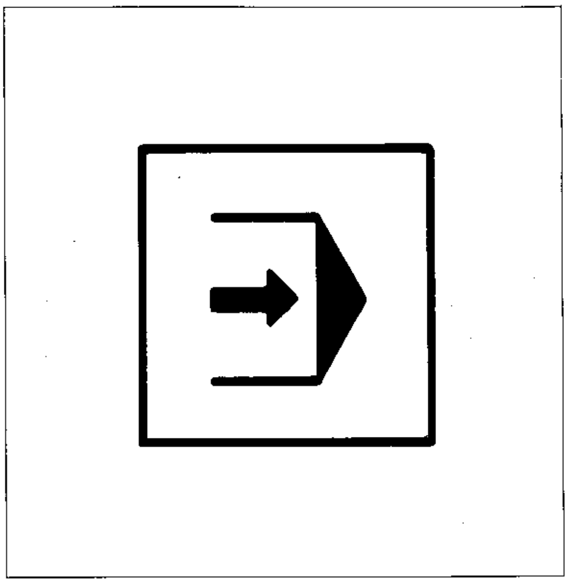
MAGN

Remarks

Program run
Modes

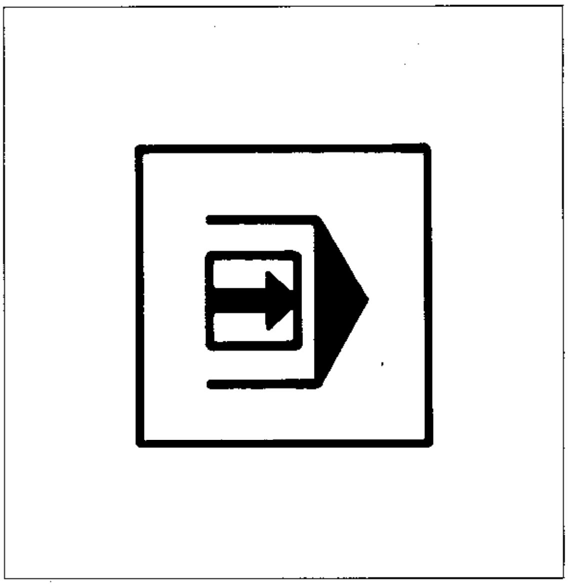
Program run/ full sequence

In the operating mode "program run/full sequence" →, the control executes the program automatically until program end. Program run is only interrupted if a "stop" has been programmed. Only in this case, does program run have to be re-started.



Program run/ Single block

In the operating mode "program run/single block" ⊟, the contro executes the stored program block by block. After execution of each block, program run must be re-started.

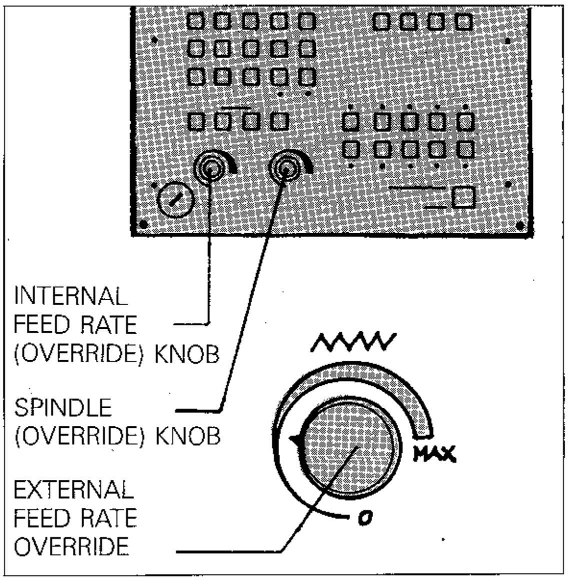


Feed rate

The programmed feed rate can be altered

- via the **internal feed rate override** and/or
- via the **external feed rate override** of the machine.

This depends however, on how the control has been adapted to the machine by the machine tool builder.


INTERNAL FEED RATE (OVERRIDE) KNOB
SPINDLE (OVERRIDE) KNOB
EXTERNAL FEED RATE OVERRIDE
MAX
0

Spindle speed

With analogue output, spindle speeds can be varied via the **spindle override.**

Program run
Start

Before machining the first workpiece, the workpiece datum must be set.

Starting program run/ Single block

Operating mode ______________ ⇒

The first program block is displayed.	▶ START	Execute first block.

▼

The second program block is displayed.	▶ START	Execute second block.

▼

⋮

Starting program run/ full sequence

Operating mode ______________ →

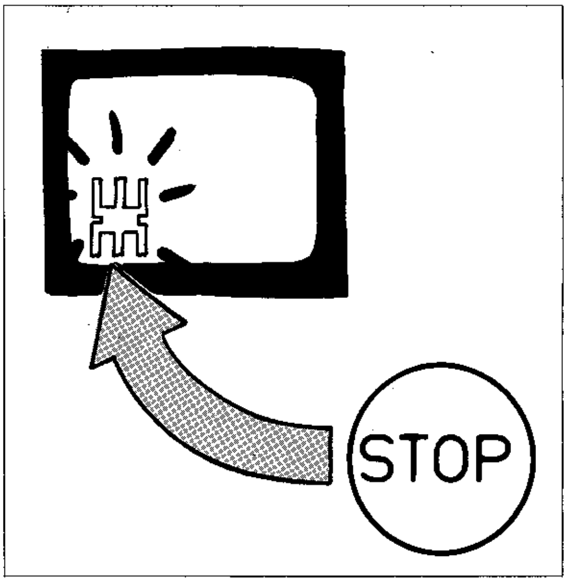
The first program block is displayed.	▶ START	Execute full program.

The control executes the program until a programmed stop or program end.

Program run
Interruption and Termination

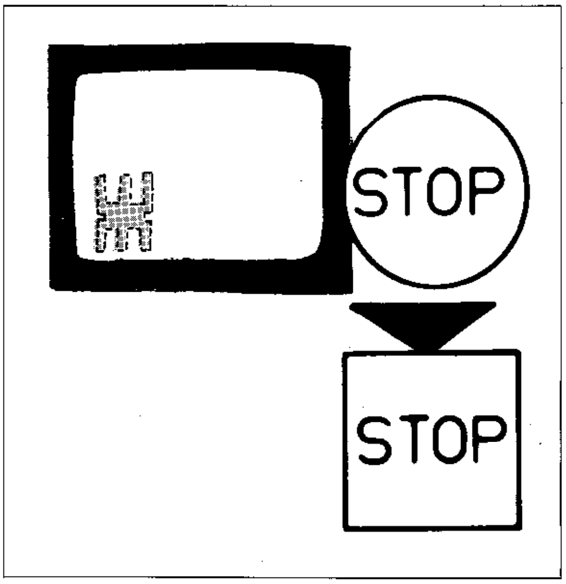
Interruption

If the control is in the ⇒-mode (program run/ full sequence or ⇨-mode (program run/single block), the program can be interrupted at any time by pressing the external stop button. Program interruption is indicated by a flashing ✱ -character (means control in operation) in the display.



Termination

Before switching over to another mode, program run must be interrupted and terminated (exception: program execution with background programming.
This is performed with the external stop-button and the stop-key of the control. With interruption, the ✱-character disappears.

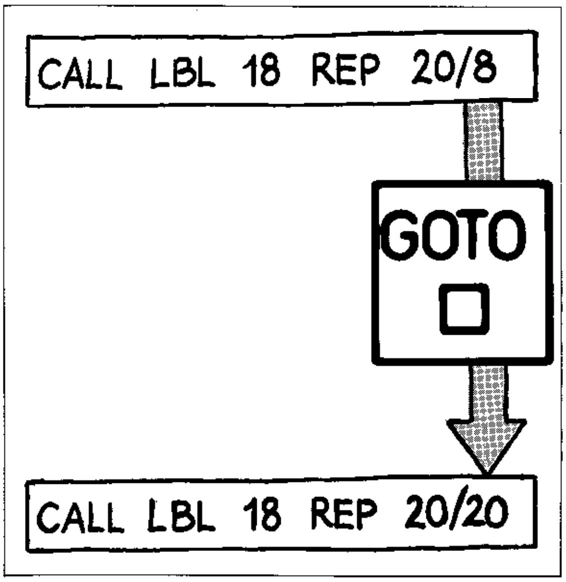


Upon termination, the following program data are stored:

- **Tool** last called-up
- **Co-ordinate transformations:** datum shift, mirror image, co-ordinate system rotation, scaling
- **Circle centre/Pole CC** last valid
- **Canned cycle** last defined
- Current status of **program part repeats**
- Return jump address with **subprograms**

If interruption takes place within a **subprogram** or a **program part repeat** and a block is addressed with GO TO, the countdown for the program part repeat sequence is reset. The return jump address is cancelled with subprograms.

If the number of repeats which are still to be executed or return jump address is to be retained, the program blocks should be addressed with the ↓ ↑ -keys.



Program run
Interruption and termination

Interruption of program run

Operating mode ______ ⇥ or →

The running program is to be interrupted:	▶ STOP	Interrupt program run

▼

The display character ✱ flashes (control in operation).

Termination of program run

Operating mode ______ ⇥ or →

The running program is to be terminated:	▶ STOP	Terminate program run
	▼ STOP	Interrupt program run

▼

Display of the ✱ -character ceases.

☞ When executing the program in ISO-format, the function of the internal START-key is taken over by the DEL -key.

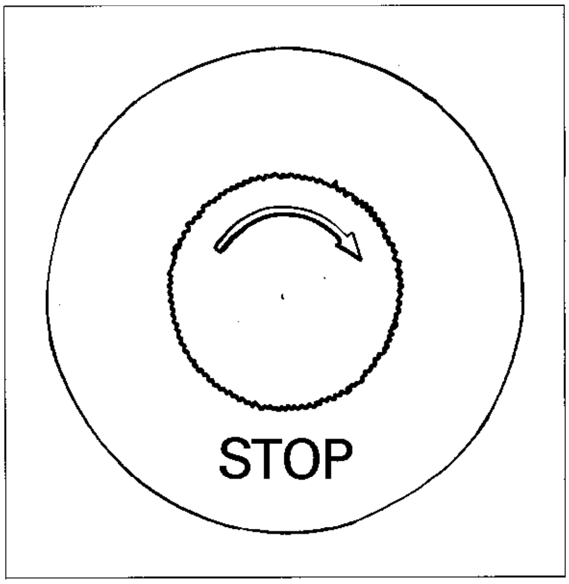
Program run
Interruption and termination

Emergency stop

In an emergency situation, the machine and the control can be switched off by pressing one of the emergency stop buttons. This is displayed by the control with

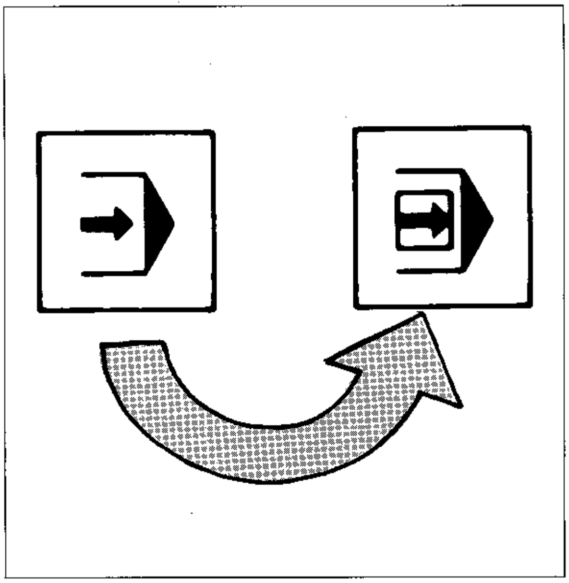
= EMERGENCY STOP =

For a new switch-on, the emergency stop button must be turned clockwise. Switch power on again and cancel display message by pressing CE . After backing-off the tool, operation may continue.



Changeover from "full sequence to single block"

If program run/full sequence → has been selected, a changeover to single block operation ▤ is possible during program run. After execution of the current block, program run is ended. Changeover during subprograms or program part repeats takes place when the call-up or number of repetitions has been completed.

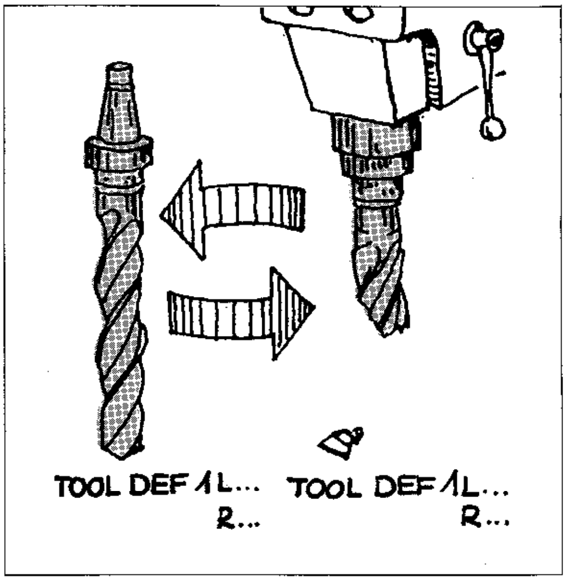


Program run
Re-entry after termination

Re-entry

A program can be re-started after an interruption or termination. To prevent workpiece damage, the **following provisions** must be made:

- the **tool** must move to the **position** it was at prior to interruption;
- the program must be re-started with the **block** in which interruption took place;
- if the **tool** has been changed due to a **tool break,** the **new tool data** (tool definition) must be entered and the tool is then re-called in the MDI-mode. The workpiece must then be touched again by the tool.



Error messages

If:

- the program has been paged after interruption (↓ , ↑),
- no block has been addressed with GOTO □ ,
- the program has not been re-started at the block which was interrupted,

  the following error is displayed:

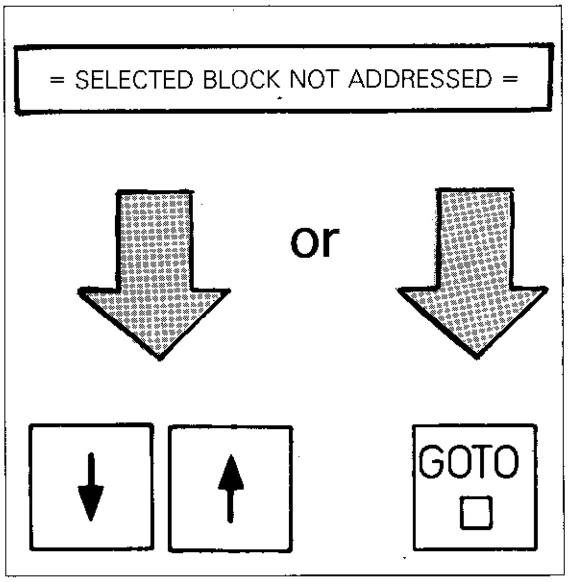
  = SELECTED BLOCK NOT ADDRESSED =

Remedy

The block which was interrupted is to be addressed by

- pressing ↑ or ↓ or
- pressing GOTO □ and entering the block number.

Caution when using the GOTO □ -key, see "Program run termination".

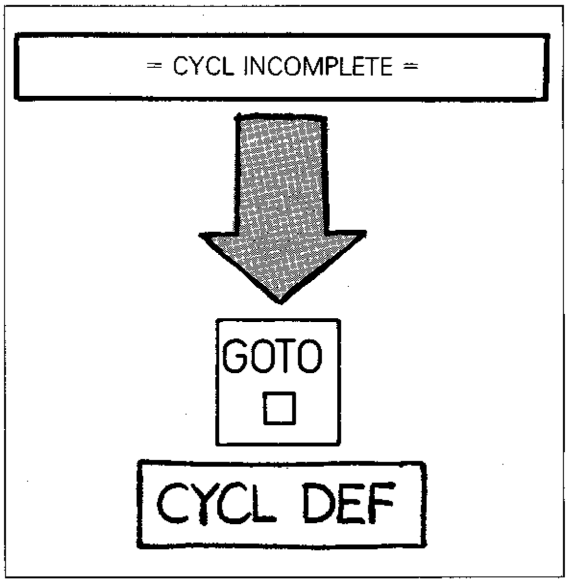


= SELECTED BLOCK NOT ADDRESSED =

Program run
Re-entry

If, after interruption of program run, a block is inserted or erased, the **cycle definition** last displayed is no longer active. With a new start, the following error is displayed before the cyle call:

= CYCL INCOMPLETE =



Remedy

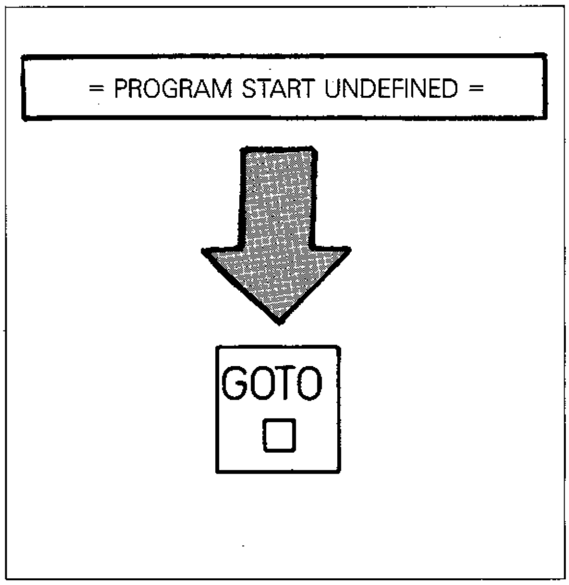
The last cycle definition must be executed before the cycle call. Addressing of the cycle definition **must** be made with the GOTO □-key!

Caution when using the GOTO □-key! see "Program run, Termination".

If program is re-started:
- ● with an amended incremental block or
- ● with a positioning block with only one co-ordination or
- ● within a canned cycle,

the following error is displayed

= PROGRAM START UNDEFINED =



Remedy

Either the program must be amended correspondingly, or a previous block is to be addressed via GOTO □.

Caution when using GOTO □-key! see "Program run, Termination".

A canned cycle must be re-started.
The "tapping cycle" may not be repeated in the same position!

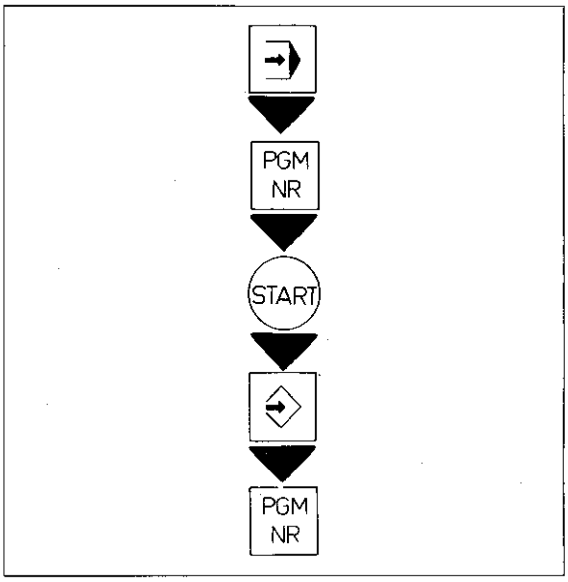
Program run with background programming

Screen display

The control permits execution of a programm via → and simultaneous entry or editing of a further program in the ◈-mode.

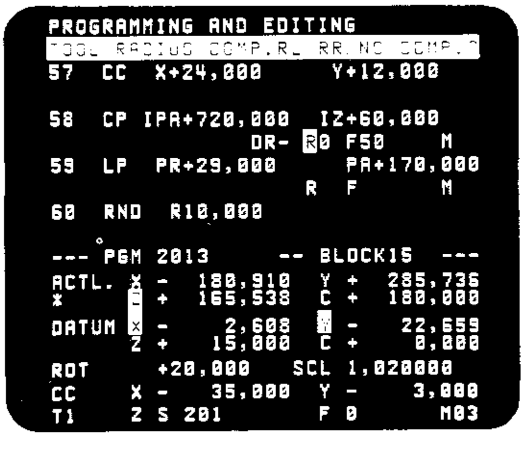
Procedure

The program to be executed must be called-up and started (operating mode →). Afterwards, the program which is to be compiled in the ◈-mode (or already stored), is defined and called see "Program call".



Screen display

Program entry is shown in the upper half of the screen and program run is displayed in the lower half. Contrary to the normal display for program run, only the program number and the current block is displayed. Position data and status displays (active cycles for co-ordinate transformations, tool, spindle rpm, feed rate and auxiliary function) are displayed as normal.



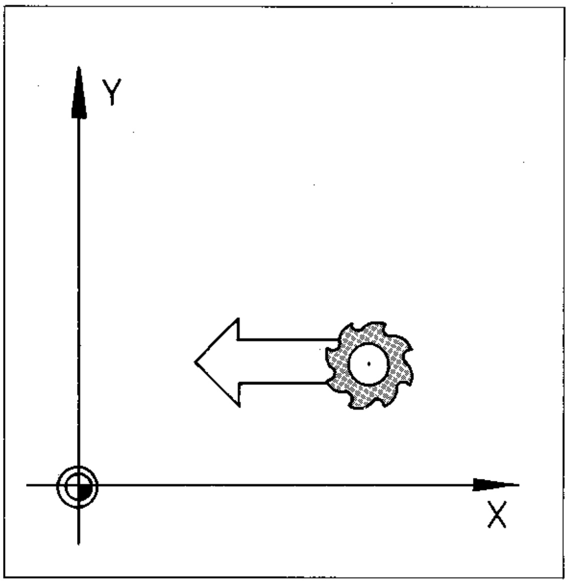
```
PROGRAMMING AND EDITING
TOOL RADIUS COMP.RL RR.NO COMP.?
57  CC  X+24,000       Y+12,000

58  CP IPA+720,000  IZ+60,000
                DR- R0 F50     M
59  LP  PR+29,000       PA+170,000
                    R  F       M
60  RND  R10,000

--- PGM 2013      -- BLOCK15   ---
ACTL. X -  180,910  Y +  285,736
*     Z +  165,538  C +  180,000
DATUM X -    2,608  Y -   22,659
      Z +   15,000  C +    0,000
ROT     +20,000   SCL 1,020000
CC    X -   35,000  Y -    3,000
T1    Z S 201       F 0      M03
```

Remarks

Single axis machining
Programming via axis address keys

Dialogue initiation

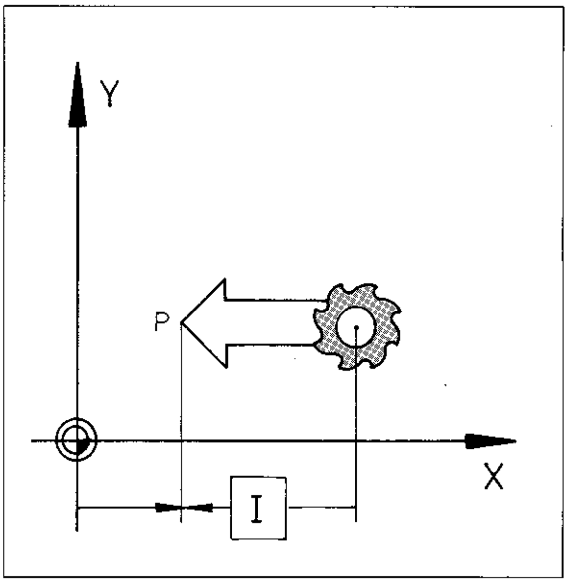
Entry of single axis positioning blocks can be simplified:
Entry dialogue is immediately initiated with the axis address keys $\boxed{X}\,\boxed{Y}\,\boxed{Z}\,\boxed{IV}$.



Nominal position value

The co-ordinate of the appropriate axis is entered as the **nominal position.** The numerical value can be specified either as an absolute value (i.e. referenced to the workpiece datum) or an incremental value (referenced to the last nominal position).

In both cases, the tool moves from its momentary actual position to the target position, in a path which is parallel to the selected axis.



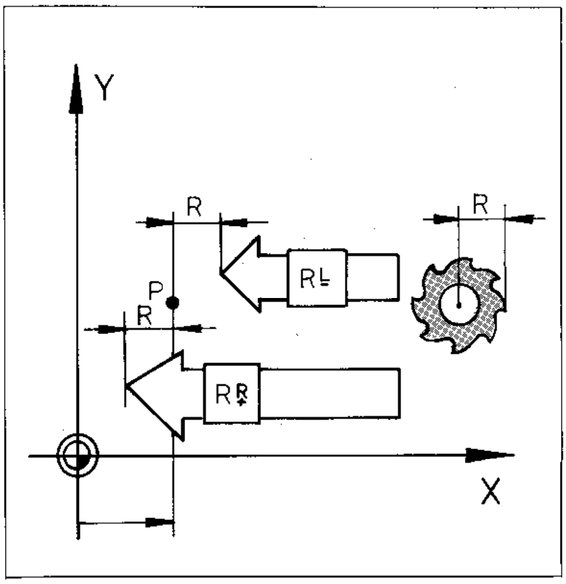
Tool radius compensation

When programming, the tool radius compensation is to be understood as follows:

- The traversing distance is **decreased** by the tool radius, $\boxed{R^{L}_{-}}$-key; display R–.
- The traversing distance is **increased** by the tool radius, $\boxed{R^{R}_{+}}$-key; display R+.
- The tool traversed to the programmed nominal position; display R0.

If R+/R– is programmed for the position of the **tool axis, no compensation** is considered.

When using the **IV axis** as **rotary axis,** tool radius compensation is also neglected.



Single axis machining
Programming via axis address keys

Single axis positioning blocks with tool compensation R+/R– or positioning blocks with RR/RL may not be entered consecutively into a program.

```
WRONG

16  L  X+15,000  Y+20,000
         RR   F     M03

17      Y+40,000
         R-    F100 M

18  L  X+50,000  Y+57,000
         RR   F     M
```

Single axis positioning blocks, which have been entered via axis keys, may be inserted between positioning blocks with R0 (no compensation) which have been programmed via contouring functions.

```
CORRECT

18  L  X+15,000  Y+20,000
         RO   F     M

19  L  X+10,000  Y+10,000
         RO   F     M

20     X+40,000
         R+   F     M

21  L  X+50,000  Y+20,000
         RO   F     M
```

Single axis machining
Programming via axis address keys

Entry of single axis movements

Operating mode ________________

Dialogue initiation ________________ X or Y or Z or IV

POSITION VALUE?	I	Incremental-Absolute?
	□	Key-in numerical value.
	ENT	Enter into memory.

TOOL RADIUS COMP. R+/R–/NO COMP.?	R^{L}_{-} R^{R}_{+}	If reqd. key-in radius compensation.
	ENT	Enter into memory.

FEED RATE? F =	□	If reqd., key-in feed rate.
	ENT	Enter into memory.

AUXILIARY FUNCTION M?	□	If reqd., key-in auxiliary function.
	ENT	Enter into memory.

Display example

```
119  IX  +46,000

                 R+   F60     M03
```

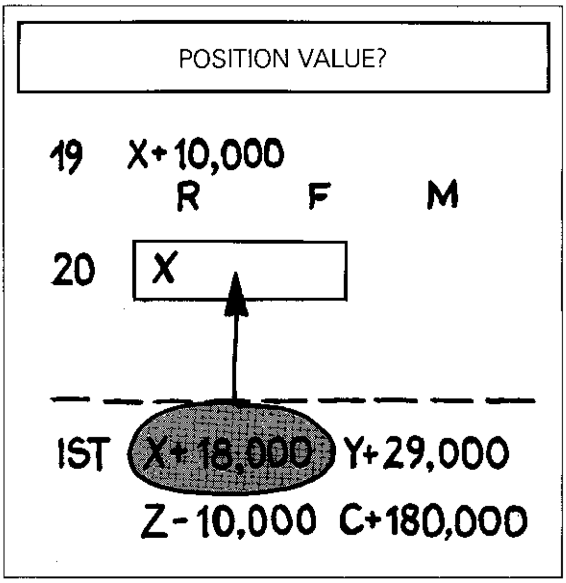
In block No. 119 the tool is moved by + 46.0 mm parallel to the X-axis plus the tool radius. The feed rate is 60 mm/min. and the spindle rotates clockwise.

Single axis machining
Playback programming

Playback

If the tool has been positioned manually (handwheel or via axis key), the actual position data can be transferred into the program as a nominal position. This type of programming is referred to as playback.

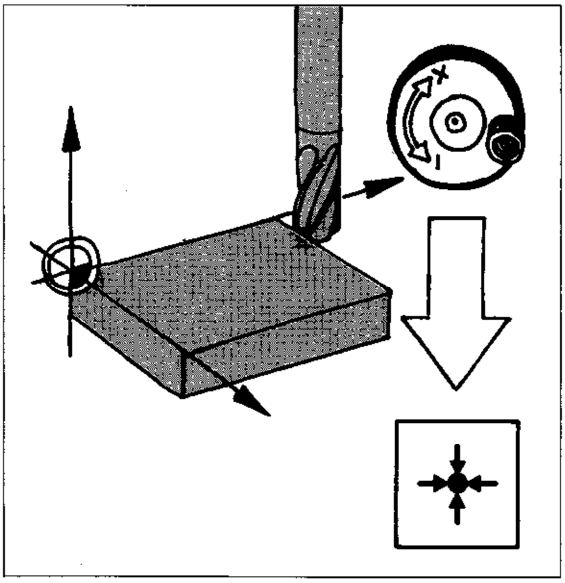
Playback programming is only advisable with single axis operation. This type of programming should be avoided on complex contours.



```
POSITION VALUE?

19   X+10,000
       R        F       M

20   X

IST  X+18,000  Y+29,000
     Z-10,000  C+180,000
```

Procedure

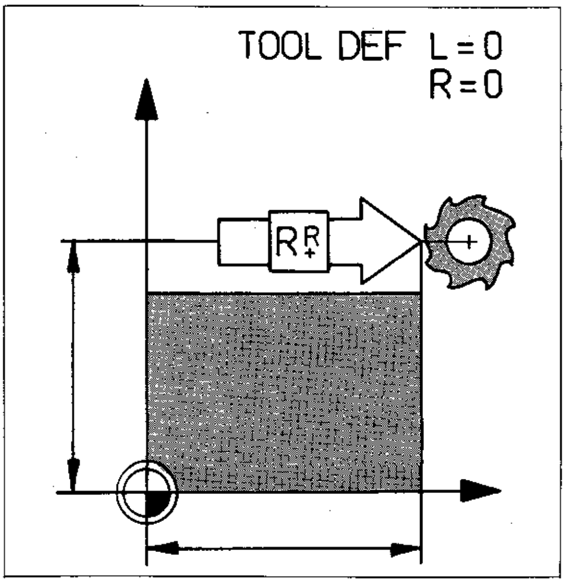
The tool is positioned to the required position either via the electronic handwheel or the axis key. In the ◇-mode, the actual position value is transferred as a nominal position value by pressing ⌖.



Tool radius compensation

The actual position value already contains the length and radius data for the tool which was used. Therefore, the compensation values L = 0 and R = 0 must be entered in the tool definition.

When programming positioning blocks with playback, the correct tool radius compensation R+ or R− or R0 is to be entered. In the event of a tool break or tool change, the new tool data can be considered.



Single axis machining
Playback programming

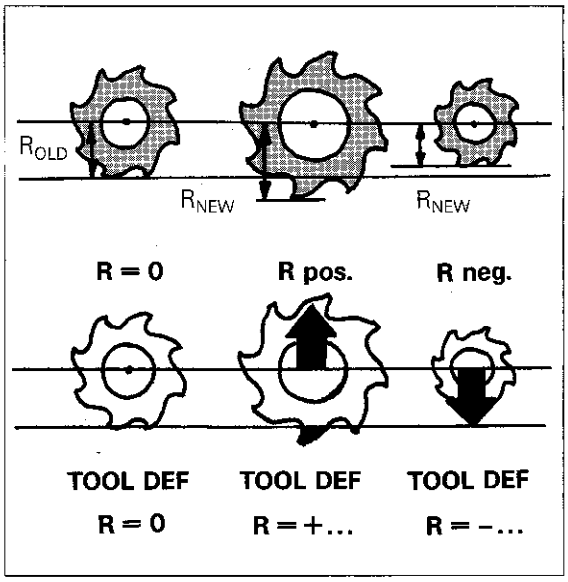
Tool compensation

The new compensation values are determined as follows:

$R = R_{NEW} - R_{OLD}$

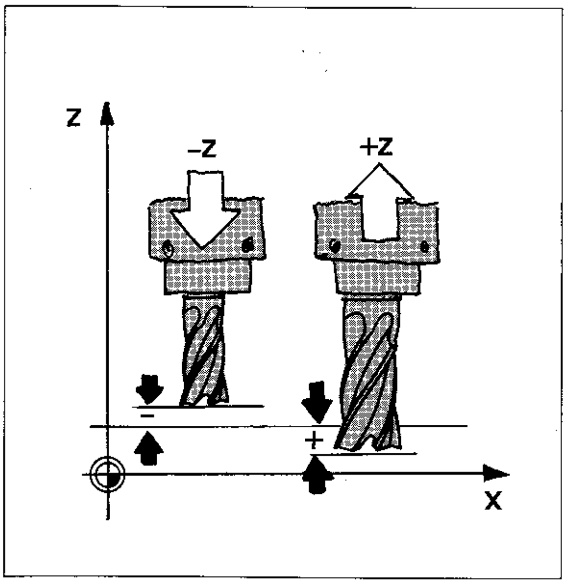
R Radius compensation value for TOOL DEF
R_{NEW} Radius of new tool
R_{OLD} Radius of original tool

The new compensation values are entered into the tool definition of the original tool (R = 0, L = 0).
A compensation value can be **positive** or **negative,** depending on the radius of the new tool being larger (+) or smaller (−).



Length compensation

The compensation value for the new tool length is determined as per TOOL DEF. In this case, the "zero tool" is the original tool.



Remarks

Single axis machining
Playback programming

Entry Example

Operating mode ______ ◈

Dialogue initiation ______ X or Y or Z or IV

POSITION VALUE?	▶ Ⓧ	Traverse tool to required position.
	⌖	Transfer position data.
	ENT	Enter into memory.

TOOL RADIUS COMP. R+/R–/NO COMP.?	RL− RR+	Key-in radius compensation, if reqd.
	ENT	Enter into memory.

FEED RATE? =	▶ □	Key-in feed rate, if reqd.
	ENT	Enter into memory.

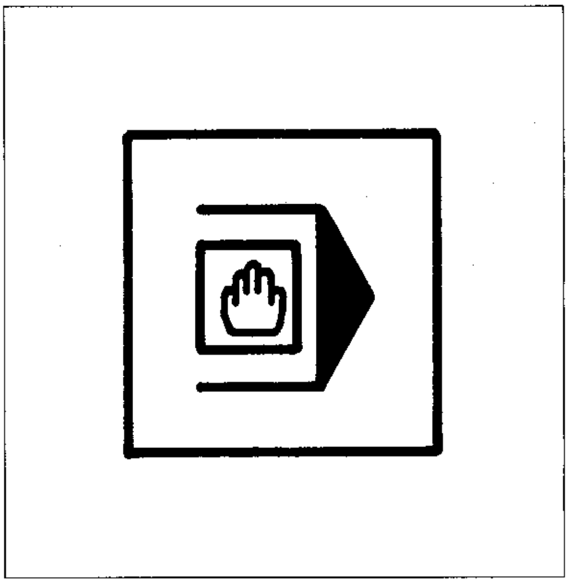
AUXILIARY FUNCTION M?	▶ □	If reqd., key-in auxiliary function.
	ENT	Enter into memory.

Program entry can be terminated in advance by pressing END □

Single axis machining
Positioning with MDI

Positioning

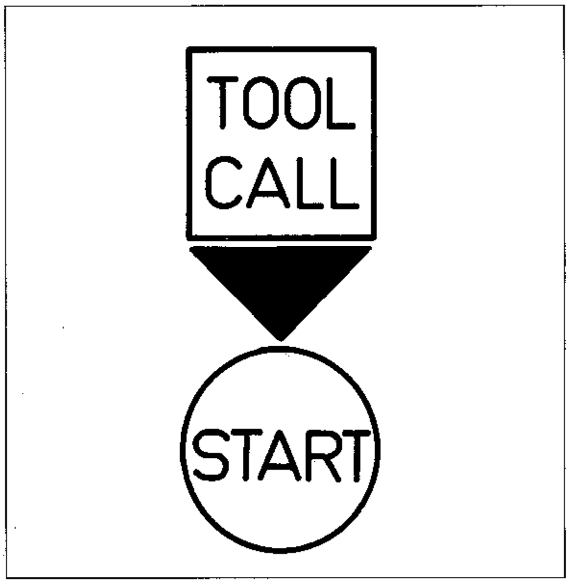
The operating mode "positioning with MDI" permits entry and execution of **single axis** positioning blocks without transfer of data into the control memory. After entry, the block must be immediately executed by pressing the external start button.



If a block contains incremental dimensions, it can be repeated as often as required.

Tool call

If a tool definition TOOL DEF already exists in the control memory, the appropriate tool may be called-up via TOOL CALL in the -mode.
The new tool data is then effective.
Tool call is executed via the external start button.

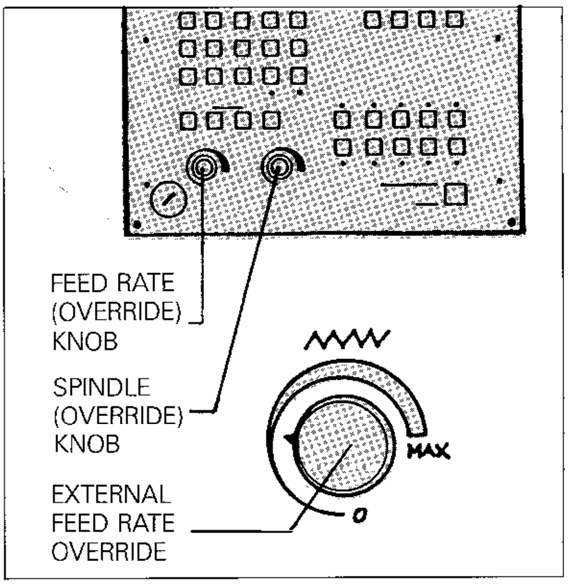


Feed rate

The programmed feed rate can be varied via the
- **internal feed rate override** and/or
- the **external feed rate override** of the machine, depending on how the control has been adapted to the machine by the machine tool builder.

Spindle speed

The programmed spindle speed can be varied via the **spindle override** (only with analogue output of spindle speed).



Single axis machining
Positioning with MDI

Example of position entry

Operating mode ________________________

Dialogue initiation ________________________ X or Y or Z or IV

POSITION VALUE?	I	Incremental/Absolute?
	□	Key-in numerical value.
	ENT	Enter into memory.
TOOL RADIUS COMP. R+/R–/NO COMP.?	R- R+	If reqd., key-in radius compensation.
	ENT	Enter into memory.
FEED RATE? F =	□	If reqd., key-in feed rate.
	ENT	Enter into memory.
AUXILIARY FUNCTION M?	□	If reqd., key-in auxiliary function.
	ENT	Enter into memory.
BLOCK COMPLETE	START	Start positioning block

Program entry can be terminated in advance by pressing END.

Remarks

Single axis machining
Positioning with MDI

Example of tool call

Operating mode ____________________

Dialogue initiation ____________________ TOOL CALL

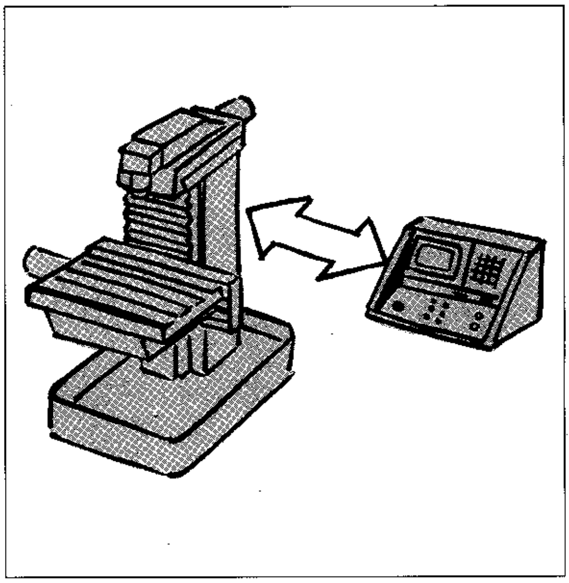
TOOL NUMBER?	□ ENT	Key-in tool number. Enter into memory.
WORKING SPINDLE AXIS X/Y/Z?	Z	Key-in axis, e.g. Z
SPINDLE SPEED S RPM =?	□ ENT	Key-in spindle rpm Enter into memory
BLOCK COMPLETE	START	Start tool call

If a central tool memory is not being used, the applicable tool definition must be given in the → mode.

Machine parameters

Machine parameters

In order that the machine can perform the control commands correctly, the control must be aware of the specific data of the machine e.g. traverses, accelerations etc. These data are determined by the machine tool builder by using machine parameters.

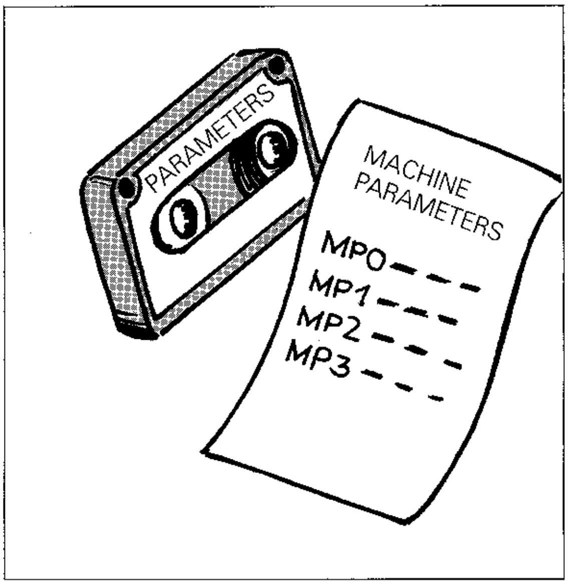


Programming

Machine parameters are entered during the initial commissioning procedure of the control. This can be done via an external data carrier (e.g. ME-cassette with stored machine parameters) or by keying-in the values manually.
After an interruption of power with either empty or missing buffer batteries, the machine parameters must be re-entered. In this case, they are requested by the control dialogue.

User-Parameter

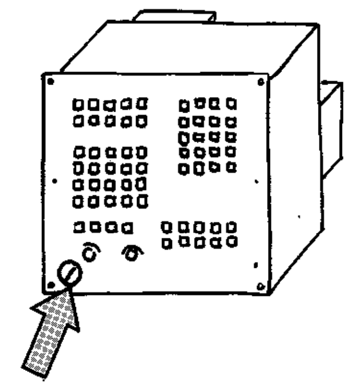
Certain machine parameters are accessible when using the MOD-mode; e.g. for switching over from HEIDENHAIN plain language to the ISO-programming language.
The machine user-parameters which are accessible via MOD are determined by the machine tool builder, who can give detailed information.



Buffer batteries

The buffer batteries are the power source for the machine parameter memory and the program memory. It is located beneath the cover on the control panel.
If the message
= EXCHANGE BUFFER BATTERY =
is displayed, the batteries must be exchanged (the batteries last for approx. 1 week after display of the above message).

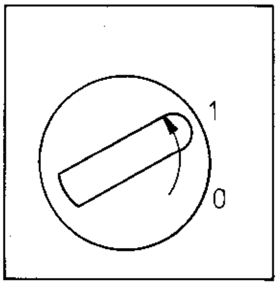
Exchange of batteries should be performed with the mains power switched on. The TNC-memories are then supplied with power. If the batteries are exchanged with the power off, all memories are erased.
Machines parameters must then be re-entered!



Battery type
Mignon cells, leak proof
IEC-description "LRG"
Recommended: VARTA Type 4006

Machine parameters

Entry via magnetic tape



Switch on power.

MEMORY TEST

The control checks the internal control electronics. This display message is automatically cleared.

EXCHANGE BUFFER BATTERY

Insert new buffer battery

CE — Clear message

OPERATION PARAMETERS ERASED

CE — Clear message

MACHINE PARAMETER PROGRAMMING

MACHINE PARAMETER MP 0?

MP0: 0

Insert magnetic tape containing parameters

TNC — Select operating mode on ME

EXT — Start external data transmission

MACHINE PARAMETER PROGRAMMING

EXTERNAL DATA INPUT

MP0: 0

Machine parameters are automatically programmed.

Machine parameters

When all parameters are entered:

POWER INTERRUPTED	▶ CE	Clear message
NC: PROGRAM MEMORY ERASED	▶ CE	Clear message
RELAY EXT. DC VOLTAGE MISSING	▶ Ⓘ	Switch-on control voltage

Finally, reference points must be traversed over.
The control is now operational.

Machine parameters

Manuel entry

Switch on mains power.

MEMORY TEST

The control checks the internal control electronics.
Display is automatically erased.

EXCHANGE BUFFER BATTERY ▶ Insert new batteries.

CE Clear message.

OPERATION PARAMETERS ERASED ▶ CE Clear message.

MACHINE PARAMETER PROGRAMMING

MACHINE PARAMETER MP 0?

MP0: 0 ▶ □ Key-in machine parameter.
MP 0 according to table.

ENT Enter into memory.

After parameter entry, the display automatically shifts to the next parameter.

Press ENT after every parameter entry.

When all machine parameters are entered:

POWER INTERRUPTED ▶ CE Clear message.

NC: PROGRAM MEMORY ERASED ▶ CE Clear message.

RELAY EXT. DC VOLTAGE MISSING ▶ I Switch on control voltage.

Finally, the reference points must be traversed over.
The control is then operational.

Machine parameters

Machine parameter number	Entry value	Machine parameter number	Entry value	Machine parameter number	Entry value
MP 00					
MP 01		MP 41		MP 81	
MP 02		MP 42		MP 82	
MP 03		MP 43		MP 83	
MP 04		MP 44		MP 84	
MP 05		MP 45		MP 85	
MP 06		MP 46		MP 86	
MP 07		MP 47		MP 87	
MP 08		MP 48		MP 88	
MP 09		MP 49		MP 89	
MP 10		MP 50		MP 90	
MP 11		MP 51		MP 91	
MP 12		MP 52		MP 92	
MP 13		MP 53		MP 93	
MP 14		MP 54		MP 94	
MP 15		MP 55		MP 95	
MP 16		MP 56		MP 96	
MP 17		MP 57		MP 97	
MP 18		MP 58		MP 98	
MP 19		MP 59		MP 99	
MP 20		MP 60		MP 100	
MP 21		MP 61		MP 101	
MP 22		MP 62		MP 102	
MP 23		MP 63		MP 103	
MP 24		MP 64		MP 104	
MP 25		MP 65		MP 105	
MP 26		MP 66		MP 106	
MP 27		MP 67		MP 107	
MP 28		MP 68		MP 108	
MP 29		MP 69		MP 109	
MP 30		MP 70		MP 110	
MP 31		MP 71		MP 111	
MP 32		MP 72		MP 112	
MP 33		MP 73		MP 113	
MP 34		MP 74		MP 114	
MP 35		MP 75		MP 115	
MP 36		MP 76		MP 116	
MP 37		MP 77		MP 117	
MP 38		MP 78		MP 118	
MP 39		MP 79		MP 119	
MP 40		MP 80		MP 120	

Machine parameters

Machine parameter number	Entry value
MP 121	
MP 122	
MP 123	
MP 124	
MP 125	
MP 126	
MP 127	
MP 128	
MP 129	
MP 130	
MP 131	
MP 132	
MP 133	
MP 134	
MP 135	
MP 136	
MP 137	
MP 138	
MP 139	
MP 140	
MP 141	
MP 142	
MP 143	
MP 144	
MP 145	
MP 146	
MP 147	
MP 148	
MP 149	
MP 150	
MP 151	
MP 152	
MP 153	
MP 154	
MP 155	
MP 156	
MP 157	
MP 158	
MP 159	
MP 160	

Machine parameter number	Entry value
MP 161	
MP 162	
MP 163	
MP 164	
MP 165	
MP 166	
MP 167	
MP 168	
MP 169	
MP 170	
MP 171	
MP 172	
MP 173	
MP 174	
MP 175	
MP 176	
MP 177	
MP 178	
MP 179	
MP 180	
MP 181	
MP 182	
MP 183	
MP 184	
MP 185	
MP 186	
MP 187	
MP 188	
MP 189	
MP 190	
MP 191	
MP 192	
MP 193	
MP 194	
MP 195	
MP 196	
MP 197	
MP 198	
MP 199	
MP 200	

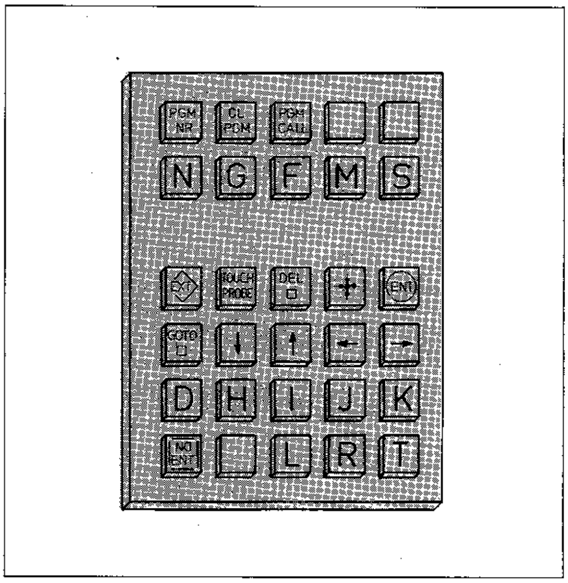
Machine parameter number	Entry value
MP 201	
MP 202	
MP 203	
MP 204	
MP 205	
MP 206	
MP 207	
MP 208	
MP 209	
MP 210	
MP 211	
MP 212	
MP 213	
MP 214	
MP 215	
MP 216	
MP 217	
MP 218	
MP 219	
MP 220	
MP 221	
MP 222	
MP 223	
MP 224	
TNC 155, as of software version 03 and TNC 151:	
MP 225	
MP 226	
MP 227	
MP 228	
MP 229	
MP 230	
MP 231	
MP 232	
MP 233	
MP 234	
TNC 155, as of software version 06 and TNC 151:	
MP 235	
MP 236	

Remarks

Program entry in ISO-format
Introduction

Snap-on keyboard

The TNC 151/TNC 155 permits program entry in either the HEIDENHAIN-conception with operator prompting via plain language dialogue or to standard format as per ISO 6983. Programming in ISO-format is advantageous when programming from an external computer.
An overlay keyboard with standard key-designations is provided for ISO-programming. The keyboard is simply placed over the existing keyboard. It is secured via small magnets.
The snap-on keyboard is immediately effective after **switchover** from HEIDENHAIN plain language dialogue to standard format.



The internal STOP-key is occupied by the D-key. In ISO-operation the internal STOP-function is performed with the DEL □-key.

Program entry

Program entry in ISO-format is partially dialogue-guided. Entry sequence for single block word information is optional. The control automatically arranges these commands into the correct order at the end of each block entry.
Errors in program entry and program execution are displayed in plain language.

Block structure, Positioning blocks

Positioning blocks may contain:

- 8 **G-functions** of different groups (see G-functions) and an additional G90 or G91 before each co-ordinate;
- 3 **co-ordinates** (X, Y, Z, IV) and an additional Circle Centre/Pole-co-ordinates (I, J, K);
- 1 **Feed rate** (max. 5 digits);
- 1 **auxiliary function** M
- 1 **spindle rpm** S (max. 4 digits);
- 1 **tool number** (max. 3 digits).

Block structure Canned cycles

Block with canned cycles may contain:

- all **individual data** for the cycle (cycle parameter P);
- 1 **auxiliary function** M;
- 1 **spindle rpm** S;
- 1 **tool number** (see G-functions) (tool call);
- 1 **positioning block;**
- 1 **feed rate** F;
- 1 **cycle call;**

Error messages

Errors within block structure are indicated during block entry, e.g.:
= G-CODE GROUP ALREADY ASSIGNED =
or, after end of block entry, e.g.
= BLOCK FORMAT INCORRECT =

Program entry in ISO-format
Control switchover

Switchover from HEIDENHAIN-programming to ISO

Switchover from HEIDENHAIN-programming language to ISO-format is performed via machine parameters. These machine parameters can be altered via the MOD-function "user parameters". "User parameters" are defined by the machine tool builder who can give you detailed information.

Program entry in ISO-format
Control switchover

Operating mode ______________ □ optional

Dialogue initiation ______________ MOD

VACANT BLOCKS: 1638	▶ MOD ↑	Select MOD-function "User parameters".
USER PARAMETERS	▶ ENT ↓ ↑	Select required user parameter.
= Dialogue as provided by machine tool builder =		
Program entry in HEIDENHAIN-format:	▶ 0	
or	DEL □	Leave supplementary mode.
Program entry in ISO-format:	▶ 1	
	DEL □	Leave supplementary mode.
POWER INTERRUPTED	▶ CE	Clear message.
RELAY EXT. DC VOLTAGE MISSING	▶ I	Switch on control voltage.

Finally, the reference points must be traversed over. The control is then operational.

When switching over the control, plain language programs are automatically converted to ISO-format and vice-versa.

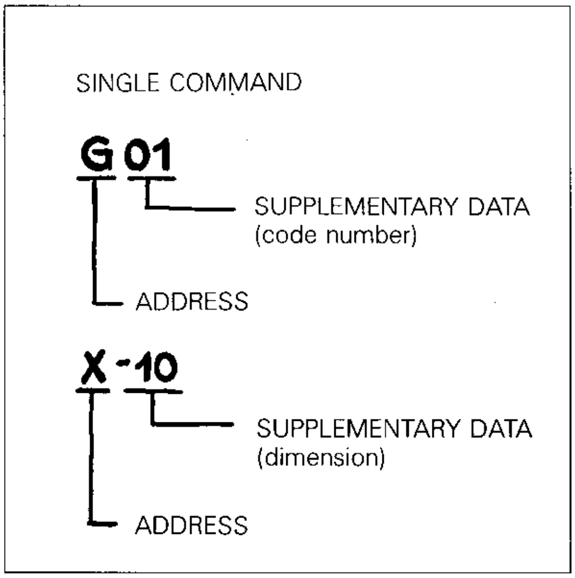
When switching over from ISO-format to plain language-format please note the following: Modal functions (e.g. G01) are only converted within the block in which it was originally programmed. The plain language symbol (e.g. L) is only altered within the original block. All subsequent plain language blocks then display * instead.

- ● K signifies Cartesian co-ordinates
- ● P signifies polar co-ordinates
- ● F MAX signifies rapid traverse.

Program entry in ISO-format
Operating the control

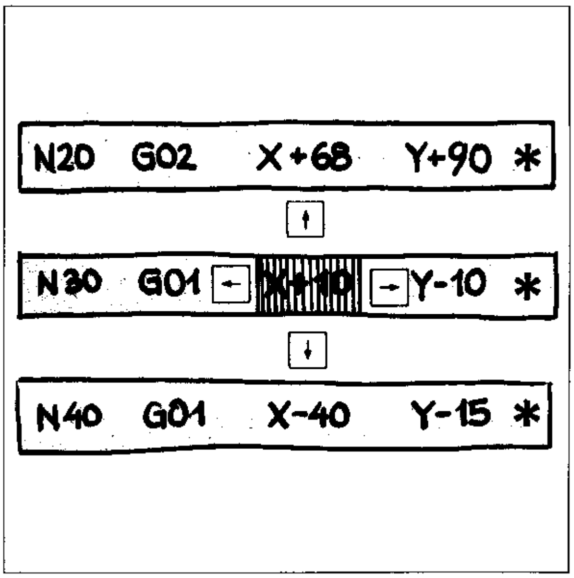
Entry of single commands

Single commands consist of an **address** and supplementary data.
A single command is entered by first pressing the address letter and the supplementary data via the decimal keyboard.
Single command entry is automatically finalised with the address letter of the following command.
If block entry can be curtailed, simply press END □.

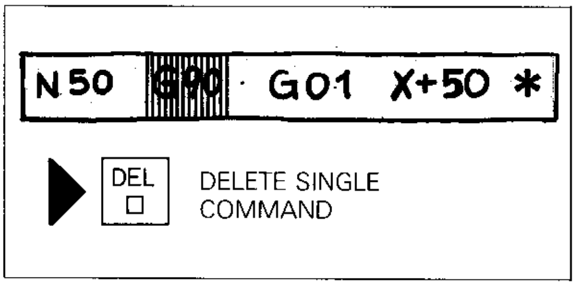

SINGLE COMMAND
G 01
SUPPLEMENTARY DATA (code number)
ADDRESS
X -10
SUPPLEMENTARY DATA (dimension)
ADDRESS

Editing

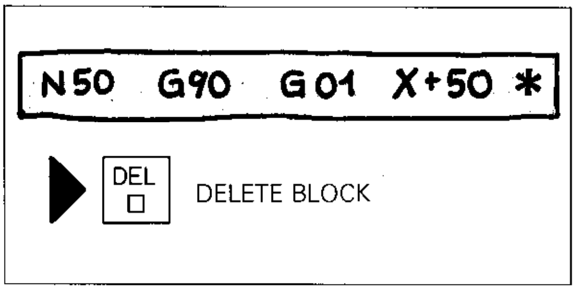
Program editing can be performed immediately after a block entry or entry of the complete program. The keys GOTO □ ↓ ↑ ← → are used for editing (see "Program editing").
As opposed to HEIDENHAIN plain language format, the cursor can be set in ISO-format by pressing → or ←.
If **the cursor** is located at a single command within a block, the ↓ ↑ -keys may be used for the search routine. Editing is ended by shifting the ← -key out of the display towards block beginning or → towards block end or press END □.
Supplementary data which has been inadvertently entered can be cleared with the CE -key.



On pressing CE a zero appears within the reverse video cursor.
The zero can be overwritten.



Erroneously entered **address letters** or **complete commands** are deleted with DEL □.



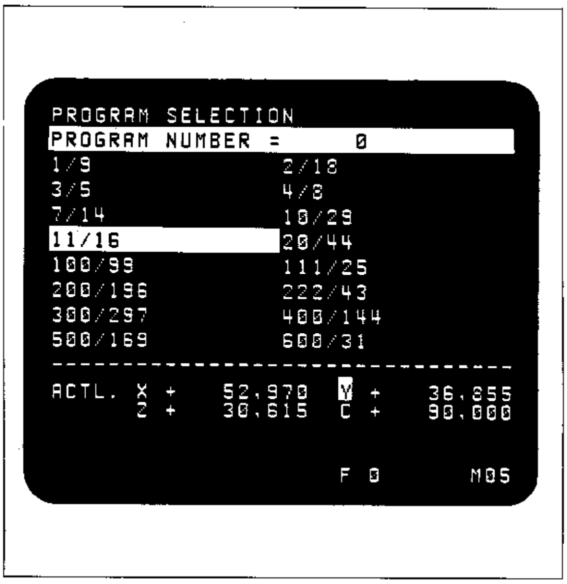
The cursor must be set to the single command for deletion!
If the cursor does not appear within a block, DEL □ deletes the complete block.

Program entry in ISO-format
Program management

Program management

The control can store up to **32 programs** with a total of **3100 program blocks.**
Entry of a new program or call-up of an existing program is performed via the PGM NR-key (see "Program call").

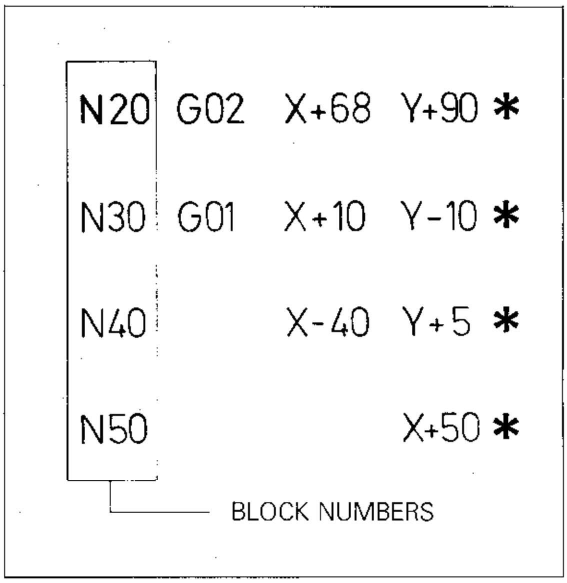
Within the program library, the number of allocated characters is indicated after the program number e.g. 20/444.



```
PROGRAM SELECTION
PROGRAM NUMBER =        0
1/9                 2/18
3/5                 4/8
7/14                10/29
11/16               20/44
100/99              111/25
200/196             222/43
300/297             400/144
500/169             600/31
--------------------------------
ACTL. X +   52,970  Y +   36,855
      Z +   30,615  C +   90,000

                    F 0      M05
```

Block number

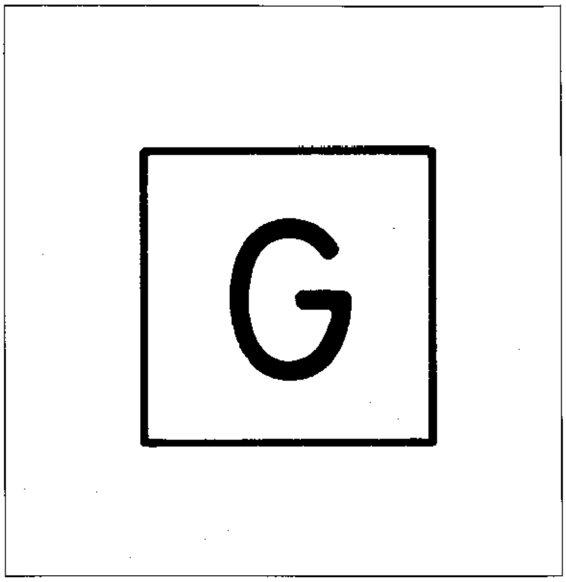
A block number comprises the **address N** and the block number.
It can be set **manually** via the N-key or **automatically** by the control.
The increment between the block numbers can be determined with the MOD-function ("Block number increment".
The control executes the program according to the block entry sequence. The actual block number has no influence on the sequence of execution.
With **program editing,** blocks with any block number may be inserted between two existing program blocks.



```
N20  G02  X+68  Y+90 *

N30  G01  X+10  Y-10 *

N40       X-40  Y+5  *

N50             X+50 *
```

Program entry in ISO-format
G-functions

Categories



Preparatory G-functions normally deal with tool path behaviour. They have the address **G** and a two-digit code number.
G-functions are split into the following groups:

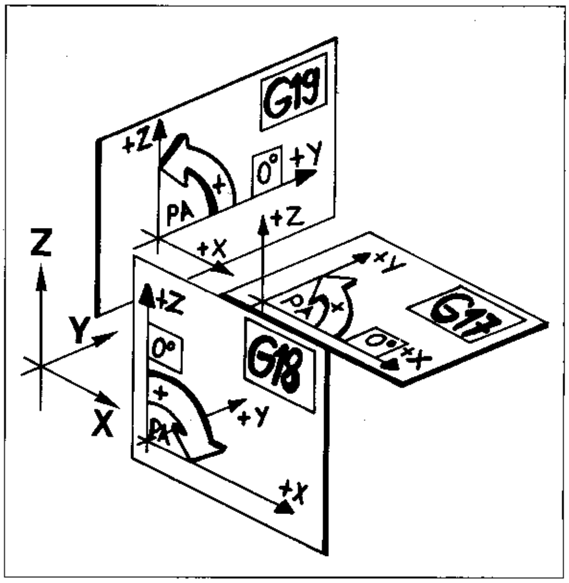
- **G-functions for positioning procedures**
  Target position in Cartesian co-ordinates
  G00-G07

  Target position in polar co-ordinates
  G10-G15

- **G-functions for cycles**
  Machining cycles:
  Drilling cycles G83-G84
  Milling cycles G74-G78

  Cycles for co-ordinate transformations
  Cycles G28/G54/G72/G73

Cycle, Dwell time G04
Freely programmable cycles (Program call) G39



- **G-functions for selecting the working plane**
  G17 Plane XY, Tool axis Z,
  Angle reference axis X
  G18 Plane ZX, Tool axis Y,
  Angle reference axis Z
  G19 Plane YZ, Tool axis X,
  Angle reference axis Y
  G20 Tool axis IV

- **G-functions for chamfering, rounding of corners and tangential contour approach**
  G24 – G27

- **G-functions for path compensation**
  G40 – G44

- **Remaining G-functions**

G29	Transfer of last nominal position value as pole
G30	Blank form definition for graphics; Min. point
G31	Blank form definition for graphics; Max. point
G38	Corresponds to a STOP-block in HEIDENHAIN-format
G50	Clear Edit protection (at program beginning)
G70	Dimensions in inch (at program beginning)
G71	Dimensions in mm (at program beginning)
G79	Cycle call
G90	Absolute dimensions
G91	Incremental dimensions
G98	Label set
G99	Tool definition
G51	Next tool number when using a central tool store
G55	Touch probe function "Surface = Datum"

Program entry in ISO-format
G-functions

Entry of G-functions

A program block may only comprise G-functions from the different groups, e.g.

N101 G01 G90...G41

Several G-functions from one group would be contradictory, e.g.

N105 G02 G03...

During program entry, the control indicates this kind of error with the message
= G-CODE GROUP ALREADY ASSIGNED =

If a code number which is unknown to the control, is allocated to the G-address, the control will indicate
= ILLEGAL G-CODE =

The first program block must contain a G-function from each of the following groups:
G17, G18, G19, G20
G00, G01, G02, G03, G06 etc.
G40, G41, G42, G43, G44
G90, G91
There is no **automatic setting!**

Program entry in ISO-format
Dimensions in inch/mm
Erase/Edit protection

Dimensions in inch/mm

G70 Dimensions in inch (dialogue-guided)

G71 Dimensions in mm (dialogue-guided)

After dialogue initiation with the PGM NR-key and response to the dialogue question:

PROGRAM NUMBER

the following dialogue question is displayed:

MM = G71 / INCH = G70

Respond to dialogue question by entering G71 or G70.

Block structure (example)

% 2 G71

%	Program beginning
2	Program number
G71	Dimensions in mm

Erase/Edit protection

G50 Erase/Edit protection (dialogue guided)

If the dialogue question PGM PROTECTION? is selected via the ← →- keys with the first block (e.g. % 2 G71) of a completely entered program, protection against erasing and editing can be provided by entering G50

Block structure (example)

% 2 G71 G50

%	Program beginning
2	Program number
G71	Dimensions in mm
G50	Edit/Erase protection

Edit/Erase protection is cancelled by entering the code number 86357.

Explanation, see "Erase/Edit protection."

Program entry in ISO-format
Tool definition/Tool call

Tool definition

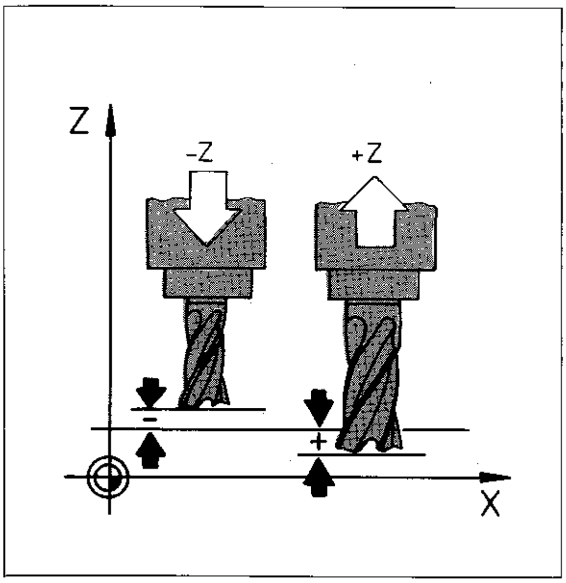
G99 Tool definition (dialogue-guided)

Block structure (example)

G99 T1 L+0 R+20

G99	Tool definition
T...	Tool number
L...	Tool length compensation
R...	Tool radius compensation

Explanation see "tool definition"



The tool definition allocates one program block

Tool call

T Tool call

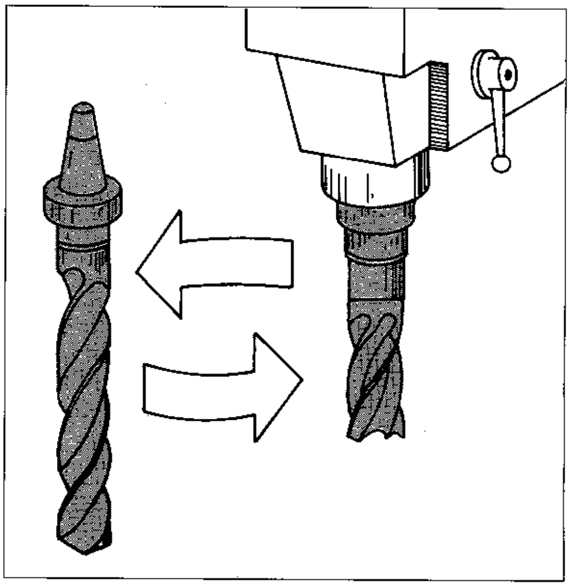
In addition to the tool call, the working plane (G17/G18/G19) and the spindle rpm must be defined.
G17 may not be programmed within a contour, since this would automatically lead to an end of path compensation.

Program structure (example)

T1 G17 S1000

T...	Tool call and tool number
G17	Working plane XY, Tool axis Z
S...	Spindle rpm

For explanation see "tool call".



Next tool

With TNC 155 as of software version ...... 02 and TNC 151.

G51 Next tool when using a central tool store

Block structure (example)

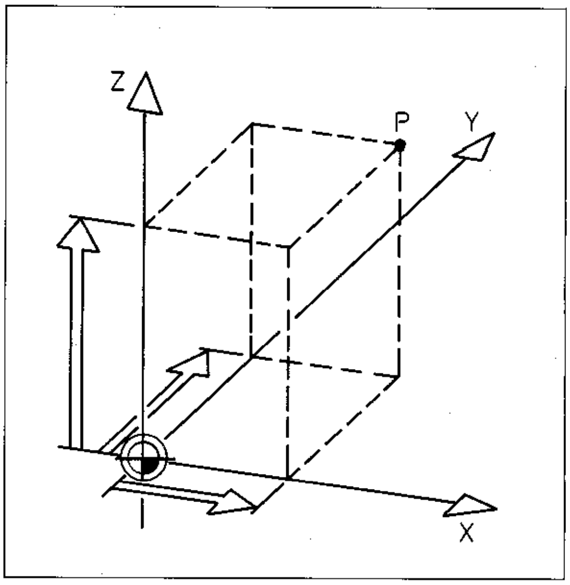
G51 T1

G51	next tool
T...	tool number

Program entry in ISO-format
Dimensions

Cartesian co-ordinates

Cartesian co-ordinates are programmed via the X Y Z IV -keys. With linear interpolation, max. 3 co-ordinates may be specified for the target position and 2 co-ordinates for circular interpolation.



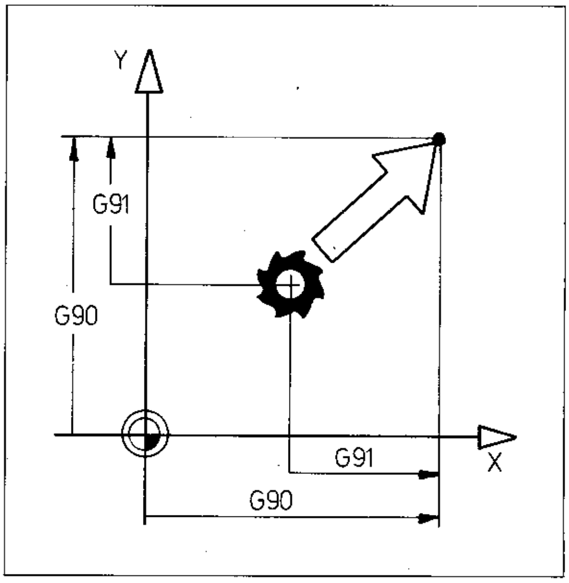
Incremental/ Absolute dimensions

The G-functions G90 – absolute dimensions and G91 – incremental dimensions are **modally** effective, e.g. they are permanently effective until they are superseded through another G-function (G91 or G90).

When specifying **co-ordinates in absolute dimensions** the G-function **G90 – absolute** must be entered (or made effective) before the appropriate co-ordinate.

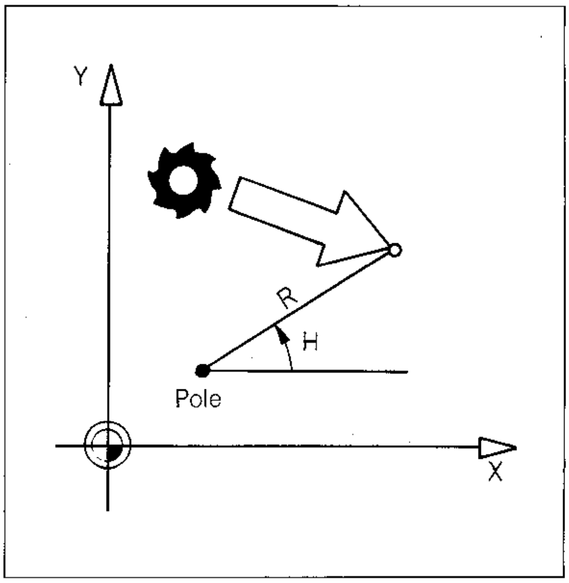
When specifying **co-ordinates in incremental dimensions** the G-function **G91 – incremental** must be entered (or made effective) prior to the appropriate co-ordinate.

G90 or G91 must be programmed before the first co-ordinate. If this is neglected, the following error is displayed
= PROGRAM START UNDEFINED =



Polar co-ordinates

Polar co-ordinates are programmed with the H -key (polar co-ordinates angle H) and the R -key (polar co-ordinates radius).
The pole must be defined before entry of polar co-ordinates.



Program entry in ISO-format
Dimensions

Pole/ Circle centre

The pole/circle centre is always defined by two Cartesian co-ordinates. The axis designations for these co-ordinates are

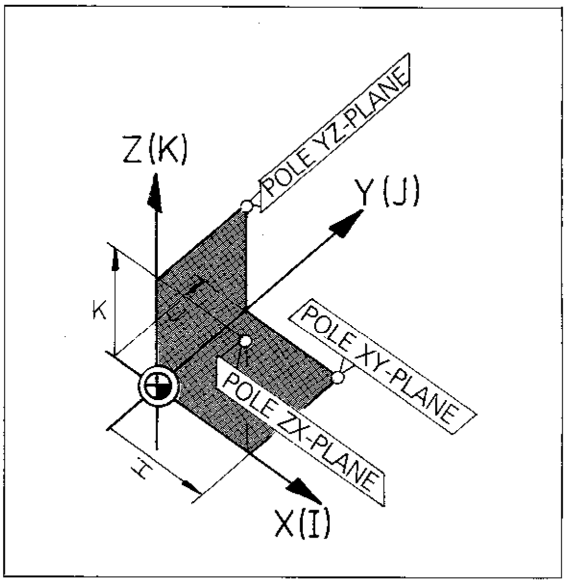
- I: for the X-axis
- J: for the Y-axis
- K: for the Z-axis

The pole/circle centre must be located in the appropriate working plane:

Working plane	Co-ordinates of pole/circle centre
X, Y-plane	I, J
Y, Z-plane	J, K
Z, Y-plane	K, I

Co-ordinate entry is via the keyboard,

I J K

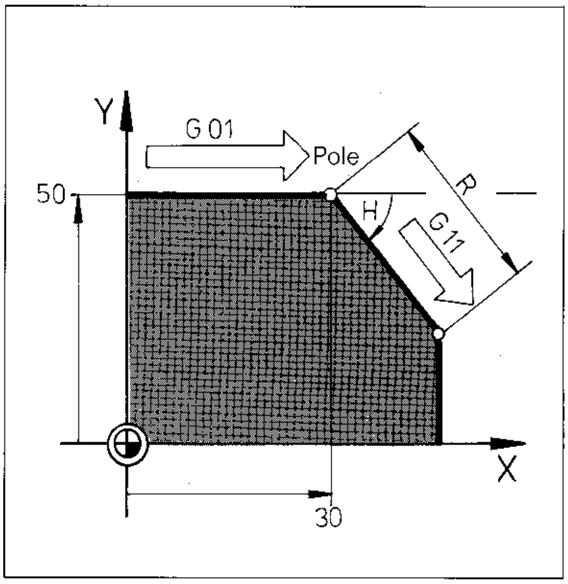


Pole definition G29

If the last nominal position value is to be transferred as a pole, the entry of the G20-function is sufficient.

Example:

```
N30   G01   G90   X+30   Y+50
N40   G29   G11   R+50   H-45
```



Program entry in ISO-format
Linear interpolation

Target position in Cartesian co-ordinates

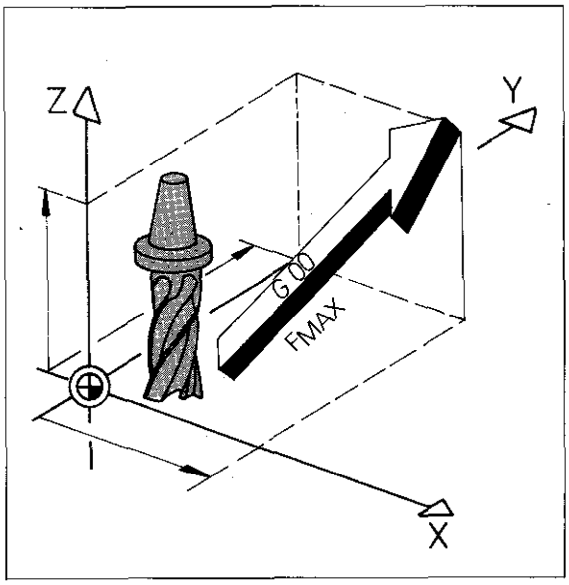
G00 Linear interpolation, Cartesian **in rapid.**

Block structure (example):

G00 G90 X+80 Y+50 Z+10

G00	Linear interpolation, Cartesian in rapid
G90	Absolute dimensions
X...	X-co-ordinate of target position
Y...	Y-co-ordinate of target position
Z...	Z-co-ordinate of target position

Simultaneous traversing of three axes in a straight line is not possible with the control versions TNC 151 E/TNC 155 E/TNC 151 V/ TNC 155 V.

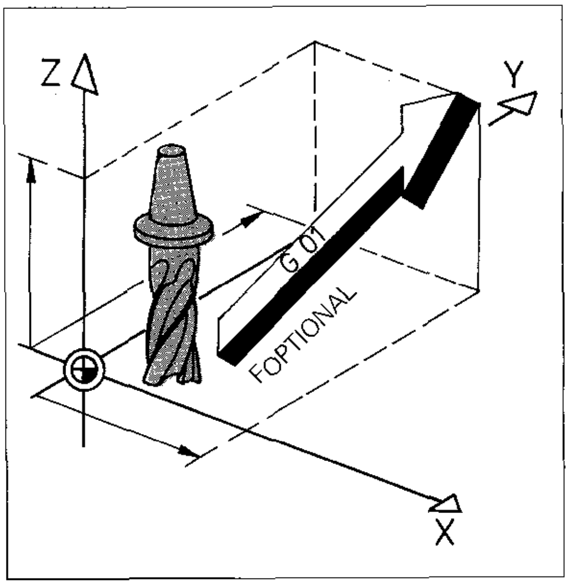


G01 Linear interpolation, Cartesian

Block structure (example):

G01 G91 X+80 Y+50 Z+10 F150

G01	Linear interpolation, Cartesian
G91	Incremental dimensions
X...	X-co-ordinate of target position
Y...	Y-co-ordinate of target position
Z...	Z-co-ordinate of target position
F...	Feed rate



Single axis positioning

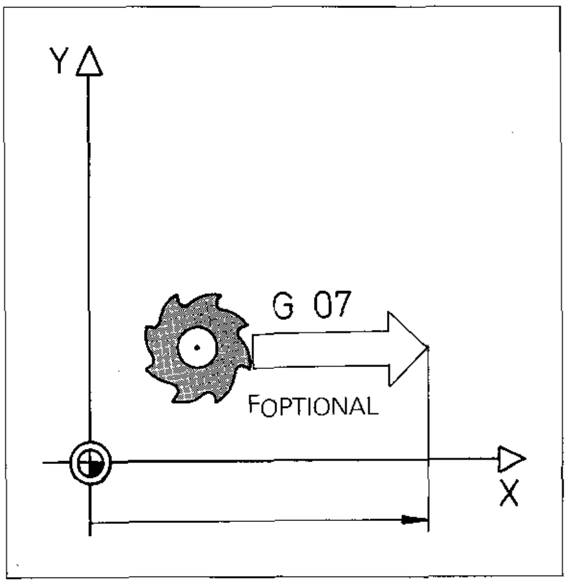
G07 Single axis movement

Block structure (example):

G07 G90 X+40 F190

G07	Single axis positioning block
G90	Absolute dimensions
X...	Co-ordinate of target position
F...	Feed rate

G07 is effective blockwise only!



Program entry in ISO-format
Linear interpolation

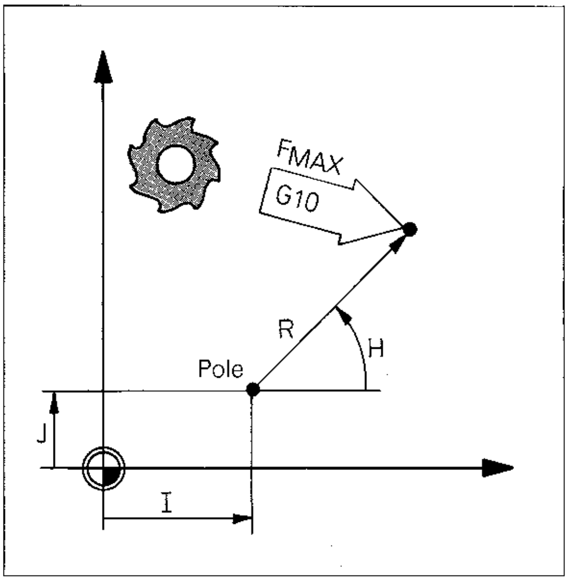
Target position in polar co-ordinates

G10 Linear interpolation, polar, **in rapid.**

Block structure (example):

G90 I+20 J+10 G10 R+30 H+45

G90	Absolute dimensions
I...	X-co-ordinate of pole
J...	Y-co-cordinate of pole
G10	Linear interpolation, polar, in rapid
R...	Polar co-ordinates radius to target
H...	Polar co-ordinates radius to target

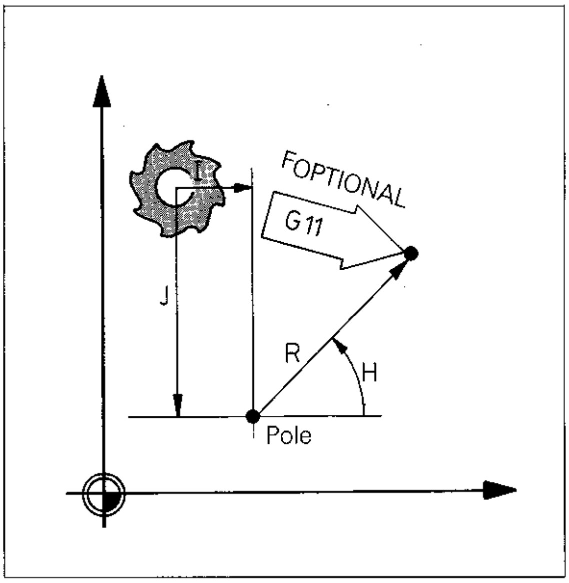


G11 Linear interpolation, polar.

Block structure (example):

G91 I+10 J–30 G11 G90 R+30 H+45 F150

G91	Incremental dimensions
I...	X-co-ordinate of pole
J...	Y-co-cordinate of pole
G11	Linear interpolation, polar
G90	Absolute dimensions
R...	Polar co-ordinates radius to target
H...	Polar co-ordinates angle to target
F...	Feed rate



Program entry in ISO-format
Circular interpolation

Target position in Cartesian co-ordinates

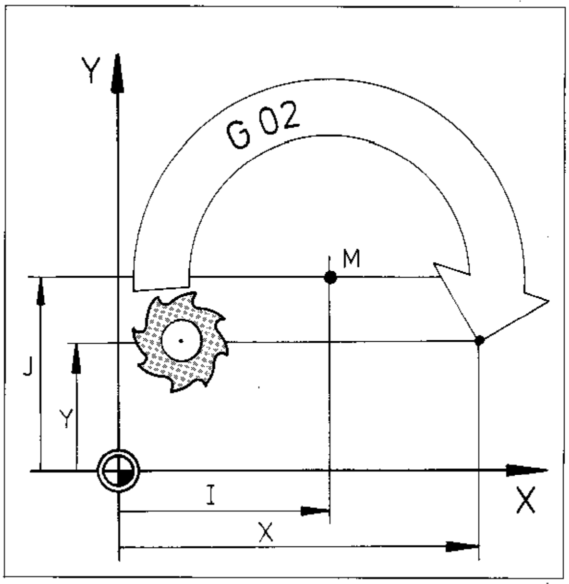
G02 Circular interpolation, Cartesian, **clockwise**

Block structure (example):

Previous block: Approach to arc starting point

G90 I+30 J+30 G02 X+69 Y+23 F150

G90	Absolute dimensions
I...	X-co-ordinate of circle centre
J...	Y-co-ordinate of circle centre
G02	Circular interpolation, Cartesian, clockwise
X...	X-co-ordinate of target position
Y...	Y-co-ordinate of target position
F...	Feed rate



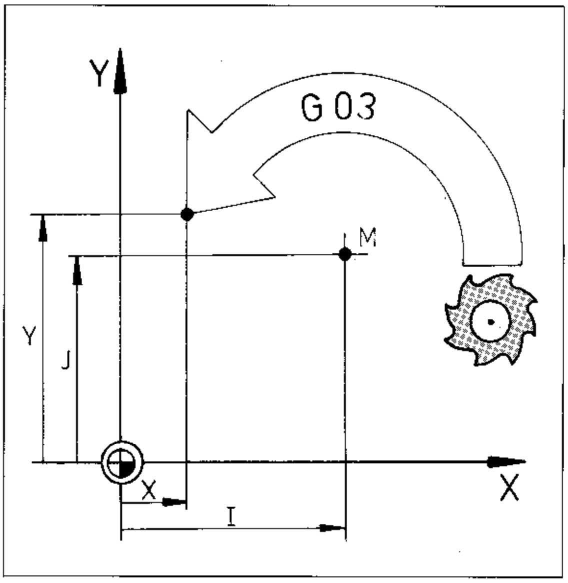
G03 Circular interpolation, Cartesian, **counter-clockwise**

Block structure (example):

Previous block: Approach to arc starting point

G90 I+30 J+28 G03 X–12 Y+32 F150

G90	Absolute dimensions
I...	X-co-ordinate of circle centre
J...	Y-co-ordinate of circle centre
G03	Circular interpolation, Cartesian, clockwise
X...	X-co-ordinate of target position
Y...	Y-co-ordinate of target position
F...	Feed rate



G05 Circular interpolation, Cartesian, **without specification of rotation**

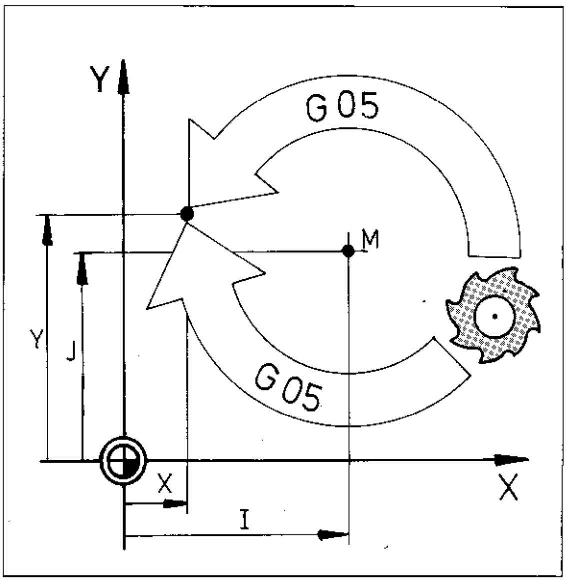
Block structure (example):

Previous block: Approach to arc starting point

G90 I+22 J+20 G05 X+5 Y+30 F150

G90	Absolute dimensions
I...	X-co-ordinate of circle centre
J...	Y-co-ordinate of circle centre
G05	Circular interpolation, Cartesian, without specification of rotation
X...	X-co-ordinate of target position
Y...	Y-co-ordinate of target position
F...	Feed rate

Before circular interpolation with G05/G15, a circular interpolation procedure with specification of rotation must already have been executed, otherwise the following message is displayed: = PROGRAM START UNDEFINED =



Program entry in ISO-format
Circular interpolation

Target position in polar co-ordinates

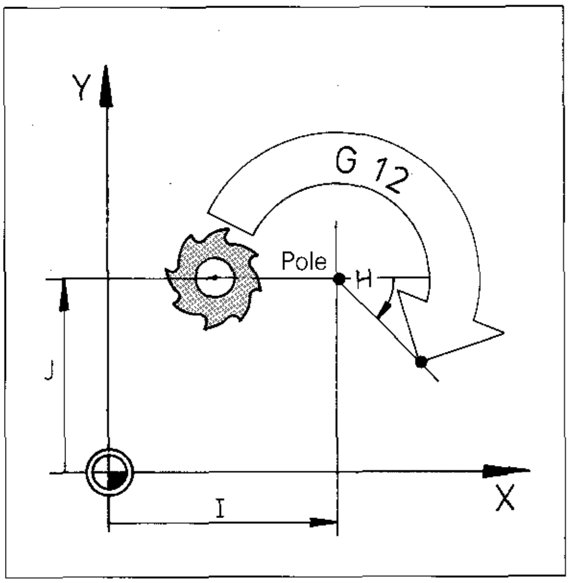
G12 Circular interpolation, polar, **clockwise**

Block structure (example):

Previous block: Approach to arc starting point

G90 I+50 J+40 G12 H–45 F150

G90	Absolute dimensions
I...	X-co-ordinate of pole/circle centre
J...	Y-co-ordinate of pole/circle centre
G12	Circular interpolation, polar, clockwise
H...	Polar co-ordinates angle to target
F...	Feed rate



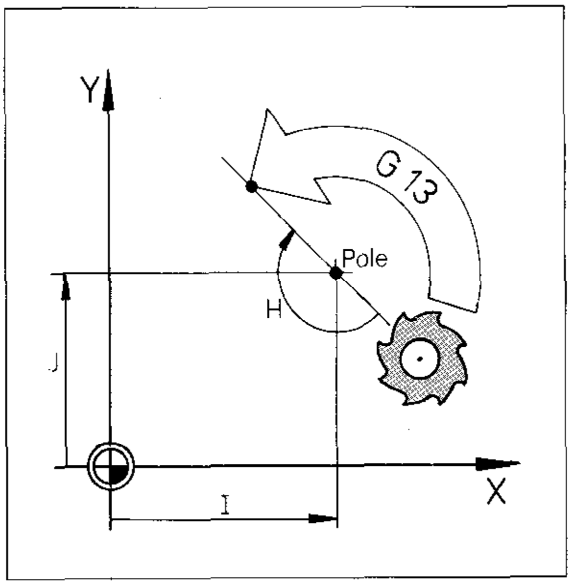
G13 Circular interpolation, polar, **counter-clockwise**

Block structure (example):

Previous block: Approach to arc starting point

G90 I–30 J+25 G13 H–180 F150

G90	Absolute dimensions
I...	X-co-ordinate of pole/circle centre
J...	Y-co-ordinate of pole/circle centre
G13	Circular interpolation, polar, counter-clockwise
H...	Polar co-ordinates angle to target
F...	Feed rate



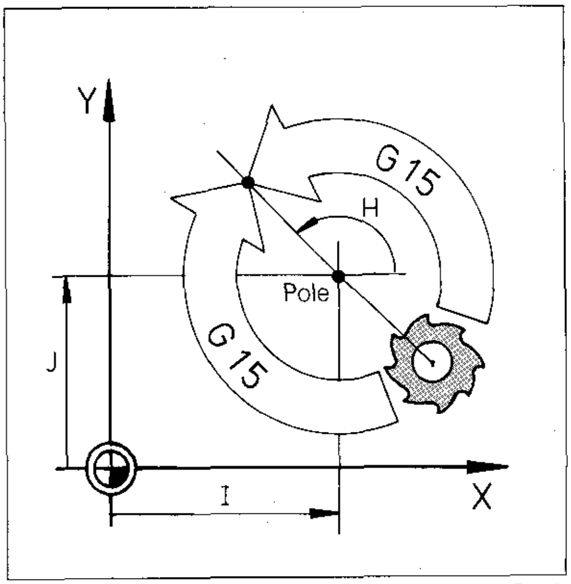
G15 Circular interpolation, polar, **without specification of rotation** (see also function G05)

Block structure (example):

Previous block: Approach to arc starting point

G90 I+50 J+40 G15 H+120 F150

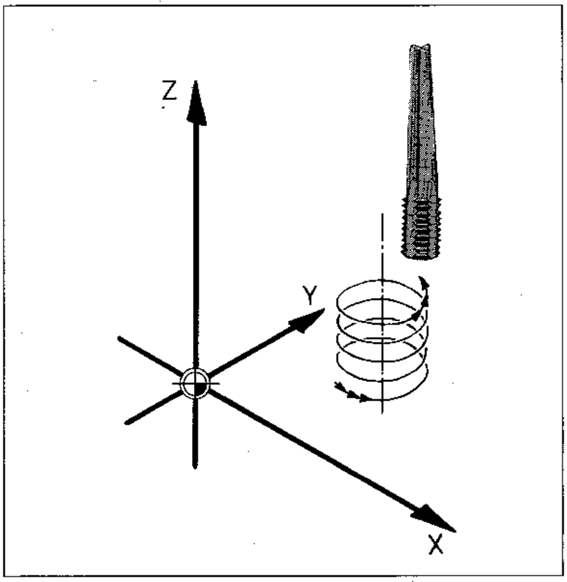
G90	Absolute dimensions
I...	X-co-ordinate of pole/circle centre
J...	Y-co-ordinate of pole/circle centre
G15	Circular interpolation, polar, without specification of rotation
H...	Polar co-ordinates angle to target
F...	Feed rate



Program entry in ISO-format
Helical interpolation
Tangential arcs

Helical interpolation

Helical interpolation is the combination of circular interpolation in the working plane and a superimposed linear movement in the tool axis. For further explanation, see "Helical interpolation".



Helical interpolation is not possible with control versions TNC 151 E/TNC 155 E/TNC 151 V/ TNC 155 V.

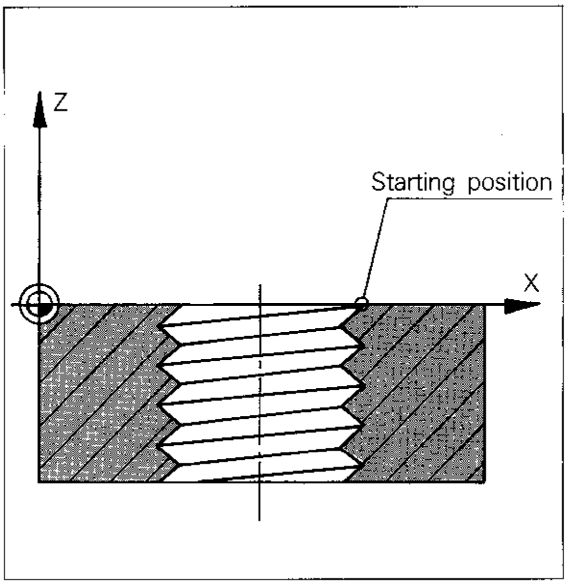
G12...Z Helical interpolation, **clockwise**

G13...Z Helical interpolation, **counter-clockwise**

Block structure (example):

G90 I+15 J+45 G12 G91 H+1080 Z-5

G90	Absolute dimensions
I...	X-co-ordinate of pole/circle centre
J...	Y-co-ordinate of pole/circle centre
G12	Circular interpolation, polar, clockwise
G91	Incremental dimensions
H...	Polar co-ordinates-angle = rotation angle
Z...	Height co-ordinate of helix



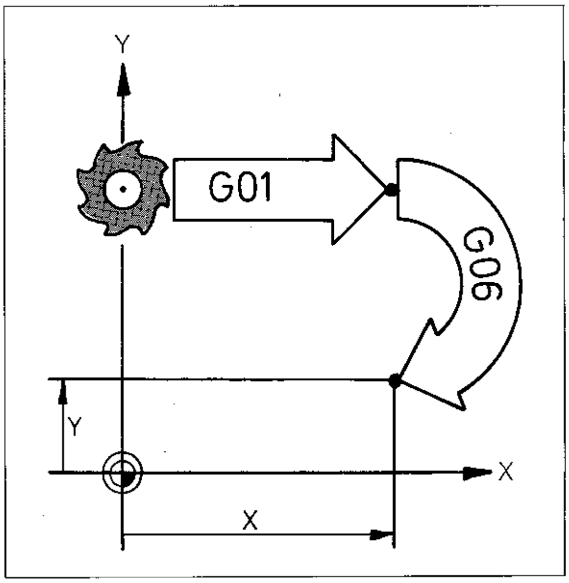
Tangential arc

G06 Circular interpolation, Cartesian, the arc tangentially adjoins the previous contour. A circle centre is not required.

Block structure (example):

G06 G90 X+50 Y+10

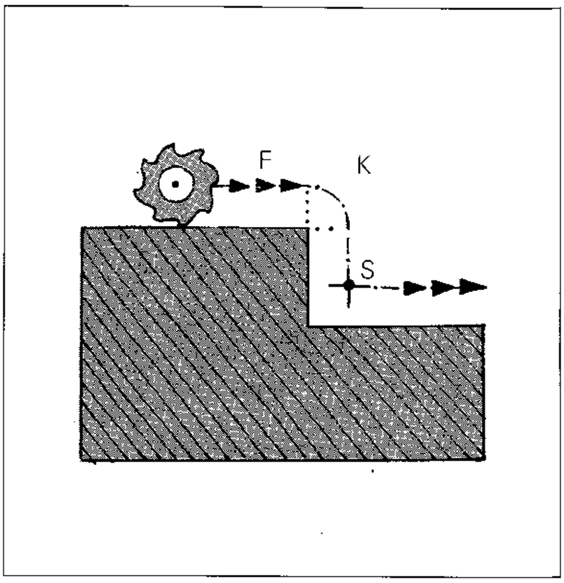
G06	Circular interpolation, Cartesian, tangential connection to contour
G90	Absolute dimensions
X...	X-co-ordinate of target position
Y...	Y-co-ordinate of target position



Program entry in ISO-format
Tool path compensation

Correction of the tool path

With **tool path compensation,** the tool moves to either the left or the right of the contour in the feed direction.
The offset corresponds to the **tool radius.**
A transitional arc K is automatically inserted on **external corners.**
With **internal corners,** the control automatically calculates a **path intersection S** so that unwanted recesses are prevented.

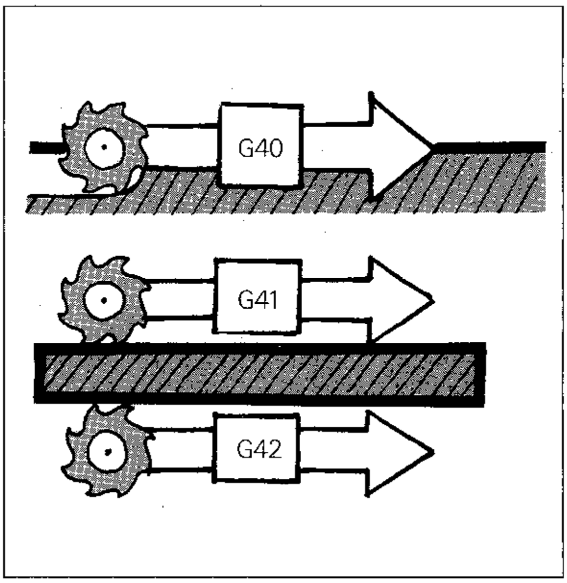


Tool path compensation

Tool path compensation is also programmed via G-functions. These G-functions are **modally effective,** i.e. they are active until they are superseded by another G-function.

Tool path compensation can be entered into **every positioning block.**

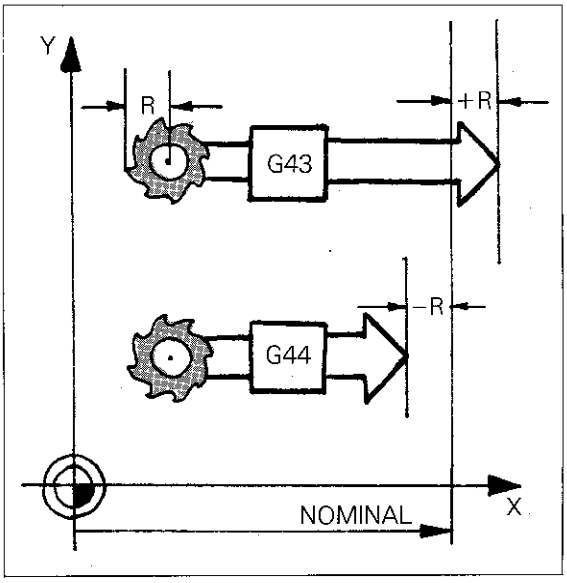
G40 The tool traverses exactly **on** the programmed contour, (cancellation of path compensation G41/G42/G43/G44).
G41 The tool path is offset to the **left** of the contour.
G42 The tool path is offset to the **right** of the contour.



Tool radius compensation with single axis positioning blocks

With single axis positioning blocks, the tool path is either increased or decreased by the tool radius.

G43 Tool path is increased
G44 Tool path is decreased



Program entry in ISO-format
Rounding of corners/Chamfers

Chamfers

G24 Chamfers

Program structure

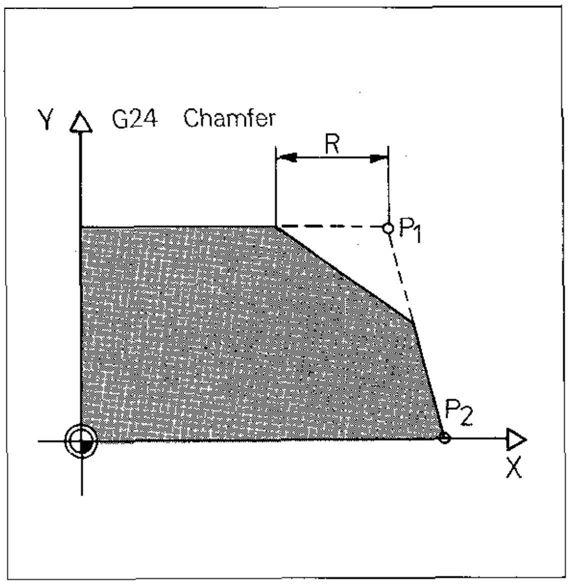
N25 G01 X... Y... (Position P1)

N26 G24 R... (Chamfer)

N27 G01 X... Y... (Position P2)

G24 may also be programmed into the block for the corner which is to be chamfered.

Explanation, see "Chamfer".



Rounding of corners

G35 Rounding of corners

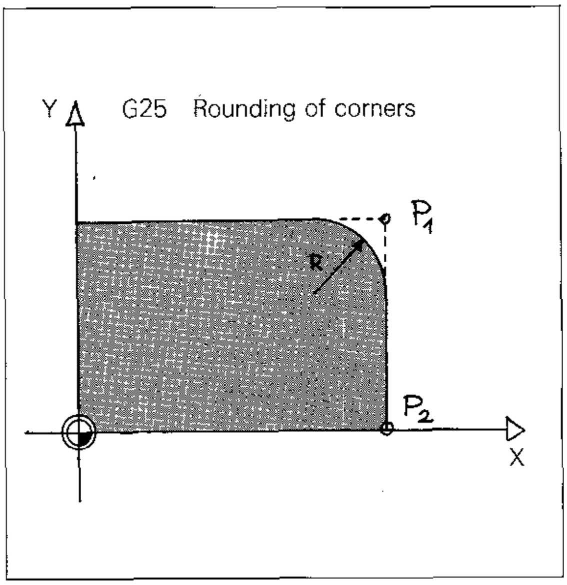
Program structure

N15 G01 X... Y... (Position P1)

N16 G25 R... (Corner radius)

N17 G01 X... Y... (Position P2)

G25 may also be programmed into the block for P1.
Explanation see "Rounding of corners".



A positioning block with both co-ordinates of the working plane must be programmed before and after the block for rounding-off or chamfering.

Program entry in ISO-format
Tangential contour approach and departure

Tangential approach (run-on)

G26 Contour approach (run-on) on a tangential arc to the first contour element (dialogue-guided).

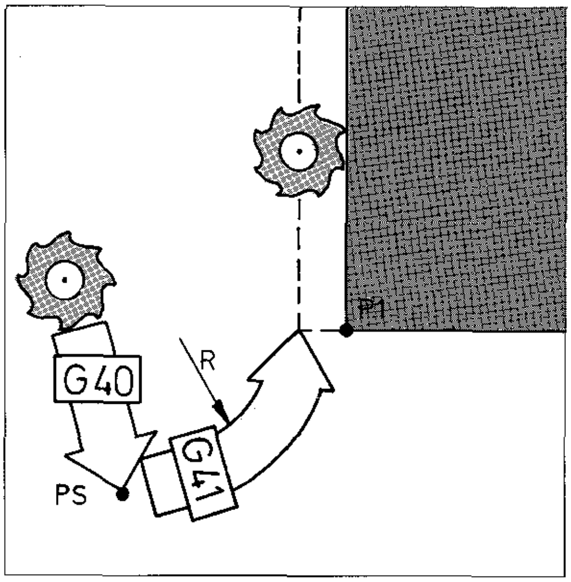
Program structure

N25 G40 G01 X... Y... (Position PS)

N26 G41 X... Y... (Position P1)

N27 G26 R... (arc)

The G26-function may also be programmed into the positioning block for the first contour position P1.
Explanation, see "Contour approach on an arc".



Tangential departure (run-off)

G27 Departure from the contour on an arc which is tangential to the last contour element (dialogue-guided).

Program structure

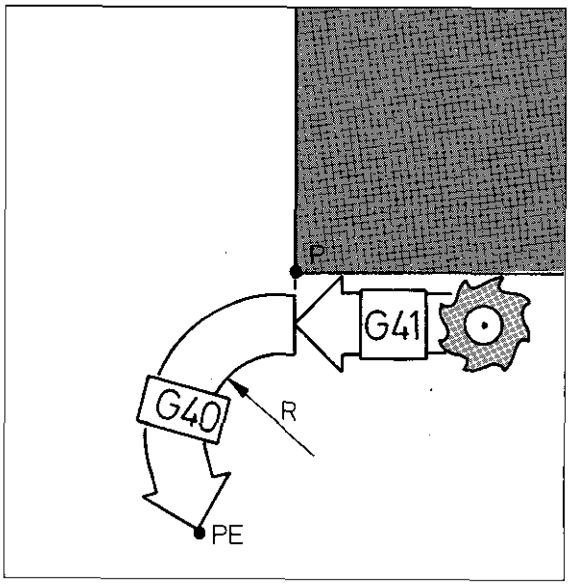
N35 G41 G01 X... Y... (Position P)

N36 G27 R... (arc)

N37 G40 X... Y... (Position PE)

The G27-function may also be programmed into the positioning block for the last contour position P1.

Explanation, see "Contour departure on an arc".



Program entry in ISO-format
Canned cycles
Machining cycles

Categories

Canned cycles are grouped into
- **Machining cycles** (for workpiece machining)
- **Co-ordinate transformations** (cycles for variations within the co-ordinate system)
- **Dwell time**
- **Freely programmable cycles**

Machining cycles are defined by G-functions and must therefore be called-up after cycle definition with either G79-cycle call – or M99 cycle call or M89 modal cycle call. This also applies to the freely programmable cycles.

Co-ordinate transformations
Are immediately effective after the definition via a G-function and therefore require no call-up. This also applies to the dwell time cycle.

Programmable **machining cycles** (dialogue-guided):

G83	Peck-drilling
G84	Tapping

G74	Slot milling
G75	Pocket milling, clockwise
G76	Pocket milling, counter-clockwise
G77	Circular pocket milling, clockwise
G78	Circular pocket milling, counter-clockwise

Programmable **co-ordinate transformations** (partially dialogue-guided):

G28	Mirror image
G54	Datum shift
G72	Scaling
G73	Co-ordinate system rotation

Further cycles (dialogue-guided)

G04	Dwell time

G39	Freely programmable cycles (program call)

Program entry in ISO-format
Canned cycles
Machining cycles

Peck-drilling

G83 Peck-drilling (dialogue-guided)

Block structure (example):

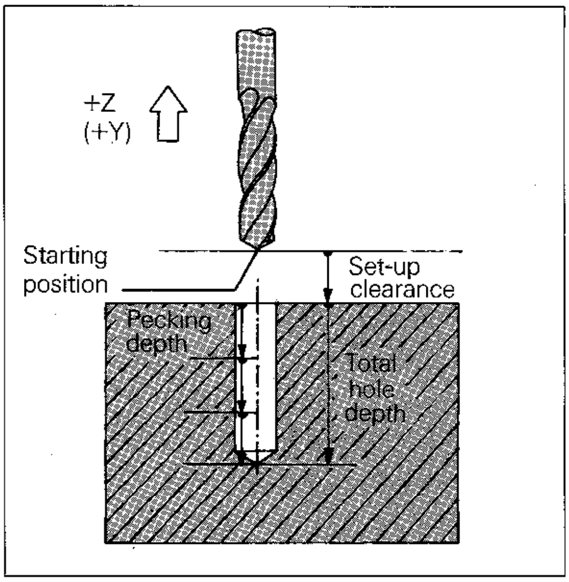
G83 P01-2 P02-20 P03-10

P04 0 P05 150

G83	Peck-drilling
P01	Set-up clearance
P02	Total hole depth
P03	Pecking depth
P04	Dwell time
P05	Feed rate

Explanation of cycle parameters and cycle procedure see "Pecking".

Cycle parameters P01/P02/P03 must have the same sign!



Tapping

G84 Tapping (dialogue-guided)

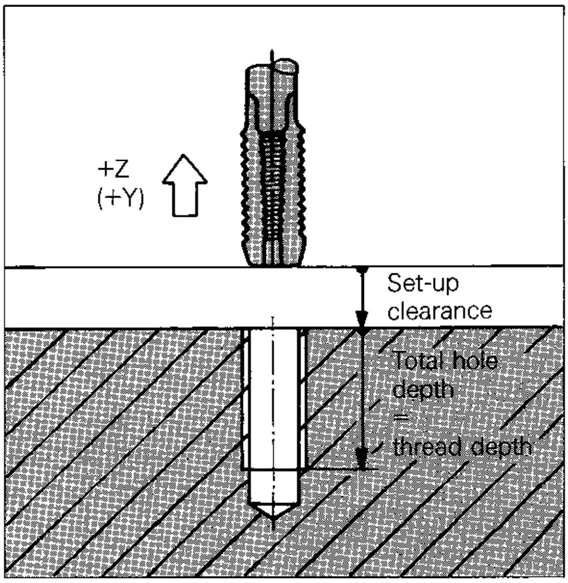
Block structure (example):

P84 P01-2 P02-20 P03 0 P04 80

G84	Tapping
P01	Set-up clearance
P02	Total hole depth (thread depth)
P03	Dwell time
P04	Feed rate

Explanation of cycle parameters and cycle procedure, see "Tapping".

Cycle parameters P01/P02 must have the same sign.



Slot milling cycle G74

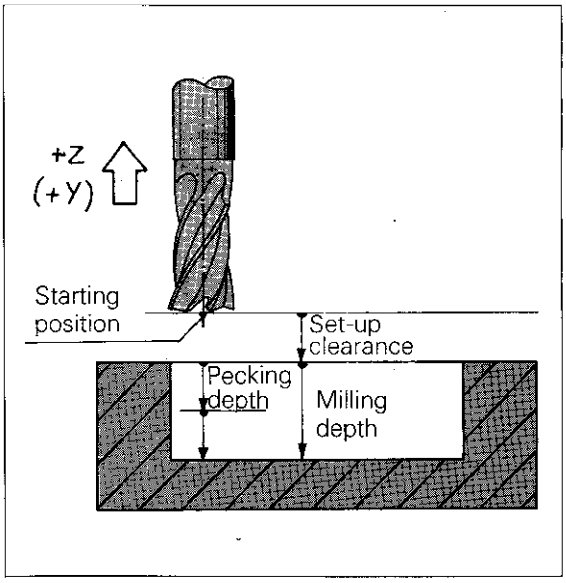
G74 Slot milling (dialogue-guided)

Block structure (example):

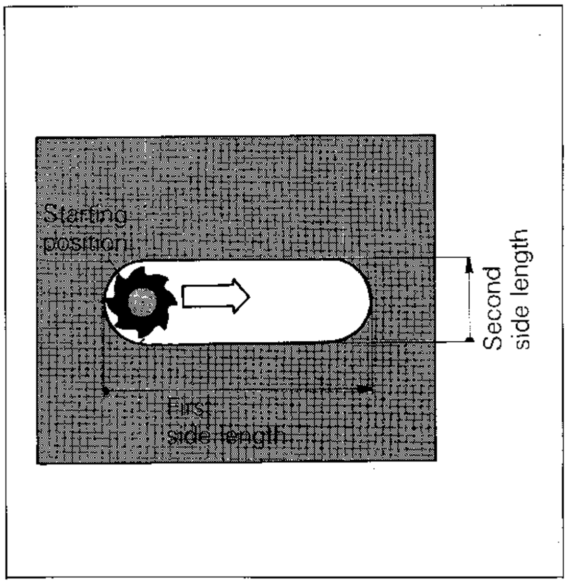
```
G74  P01-2  P02-20  P03-10  P04  80

P05  X+50  P06  Y+10  P07  150
```

G74	Slot milling
P01	Set-up clearance
P02	Milling depth
P03	Pecking depth
P04	Feed rate for pecking
P05	Length-axis and first side length
P06	Width-axis and second side length
P07	Feed rate

Explanation of cycle parameters and cycle procedure, see "Slot milling".



Cycle parameters P01/P02/P03 must have the same sign!



Pocket milling

G75 Pocket milling, **clockwise** (dialogue-guided)

G76 Pocket milling, **counter-clockwise** (dialogue-guided)

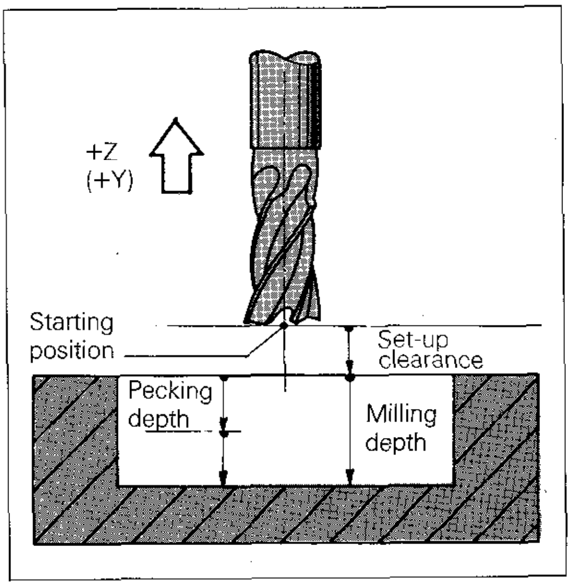
Block structure (example G76):

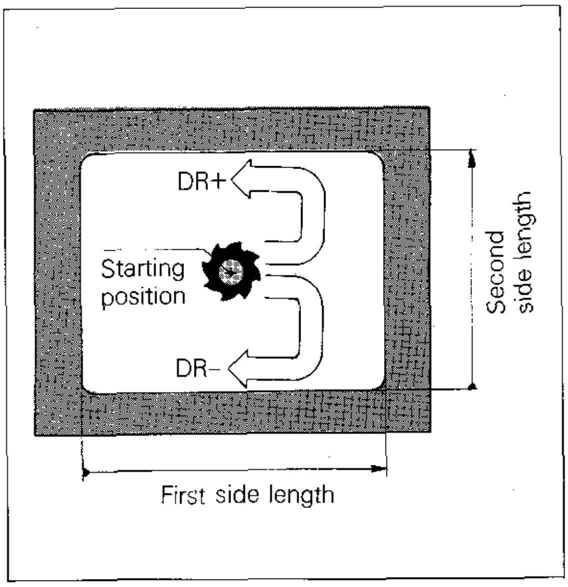
```
G76  P01-2  P02-20  P03-10  P04  80

P05  X+90  P06  Y+50  P07  150
```

G76	Pocket milling, counter-clockwise
P01	Set-up clearance
P02	Milling depth
P03	Pecking depth
P04	Feed rate for pecking
P05	First axis direction and side length
P06	Second axis direction and side length
P07	Feed rate

Explanation of cycle parameters and cycle procedure, see "Pocket milling".

Cycle parameters P01/P02/P03 must have the same sign!
Cycle parameters P05 and P06 must have a positive sign!





Program entry in ISO-format
Machining cycles

Circular pocket

G77 Circular pocket milling, **clockwise** (dialogue-guided)

G78 Circular pocket milling, **counter-clockwise** (dialogue-guided)

Block structure (example G78):

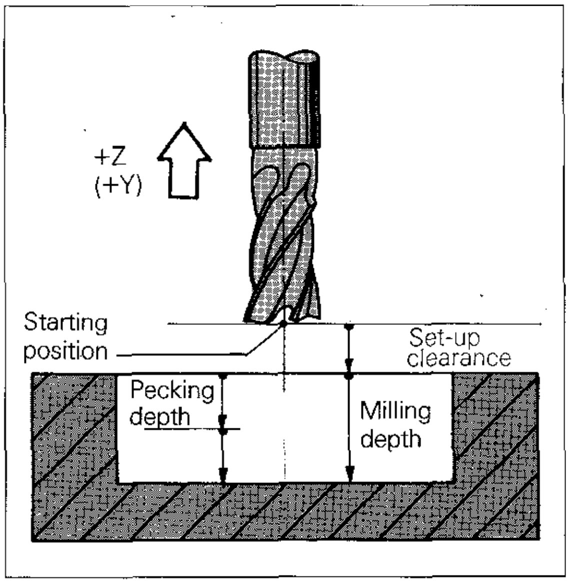
G78 P01-2 P02-20 P03-10 P04 80

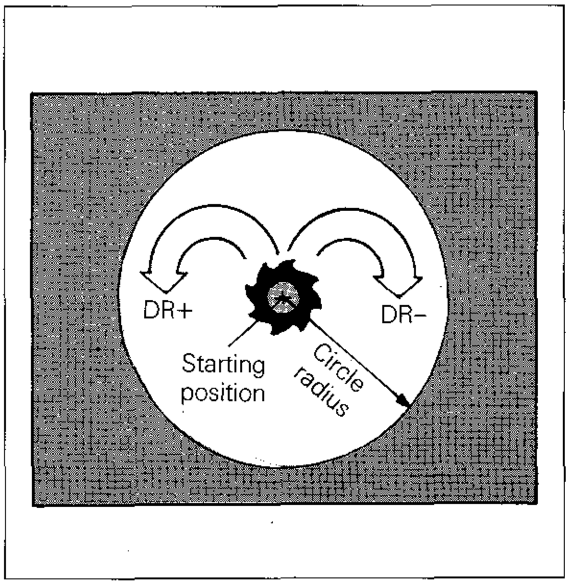
P05 90 P06 150

G78	Circular pocket, counter-clockwise
P01	Set-up clearance
P02	Milling depth
P03	Pecking depth
P04	Feed rate for pecking
P05	Circle radius
P06	Feed rate

Explanation of cycle parameters and cycle procedure, see "Circular pocket milling".

Cycle parameters P01/P02/P03 must have the same sign!





Program entry in ISO-format
Co-ordinate transformations

Mirror image

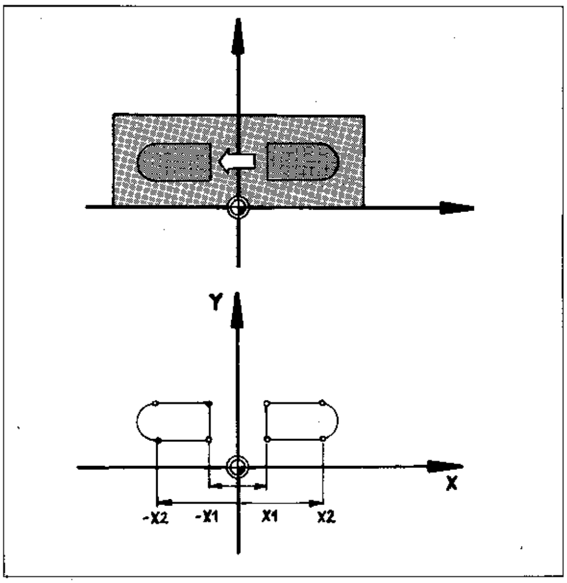
G28 Mirror image

Block structure (example):

G28 X

G28	Mirror image
X	Mirror image axis

Two axes may be mirror imaged simultaneously; the mirror imaging of the tool axis is not possible.
Explanation of cycle, see "Mirror image".



Datum shift

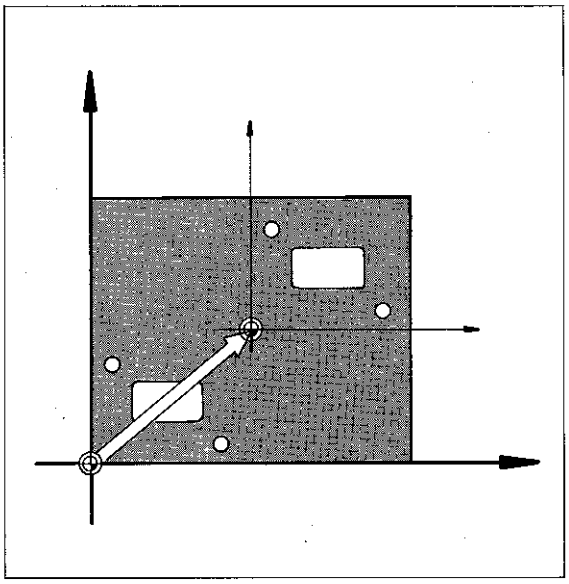
G54 Datum shift

Block structure (example):

G54 G90 X+50 G91 Y+15 Z–10

G54	Datum shift
G90	Absolute dimensions
X...	Datum shift, X-axis
G91	Incremental dimensions
Y...	Datum shift, Y-axis
Z...	Datum shift, Z-axis

Explanation of cycle, see "Datum shift".



Scaling

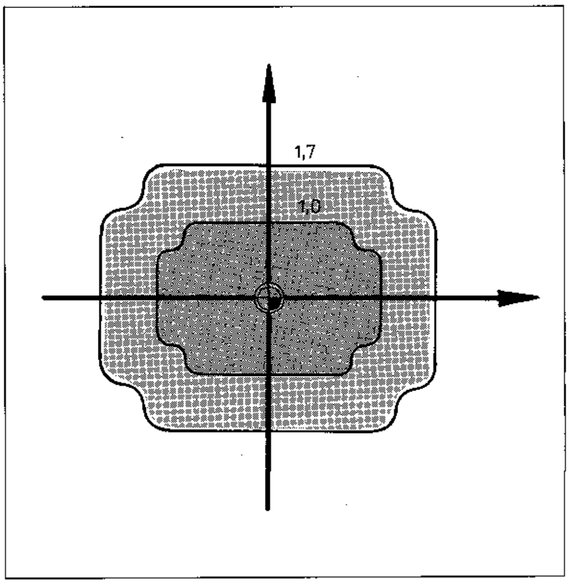
G72 Scaling (dialogue guided)

Block structure (example):

G72 F 1.7

G72	Scaling cycle
F...	Scaling factor

Explanation of cycle, see "Scaling".



Program entry in ISO-format
Co-ordinate transformations
Dwell time, Freely programmable cycle

Co-ordinate system rotation

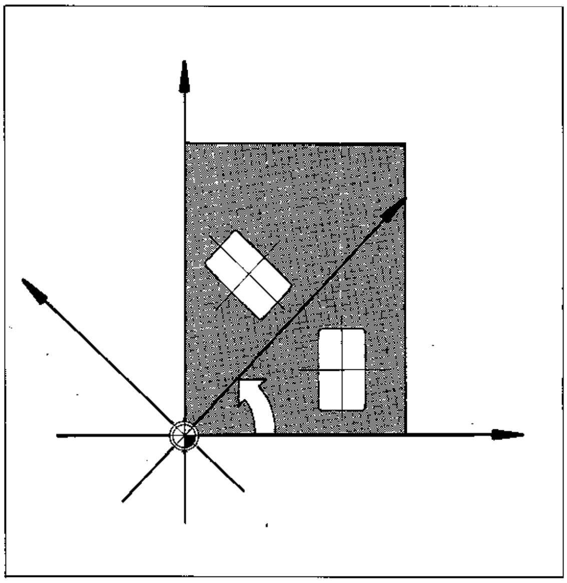
G73 Co-ordinate system rotation (dialogue-guided)

Block structure (example):

G90 G73 H+120 G17

G90	Absolute dimensions
G73	Co-ordinate system rotation
H...	Rotation angle
G17	Plane selection for angle reference axis

Explanation of cycle, see "Co-ordinate system rotation".



Dwell time

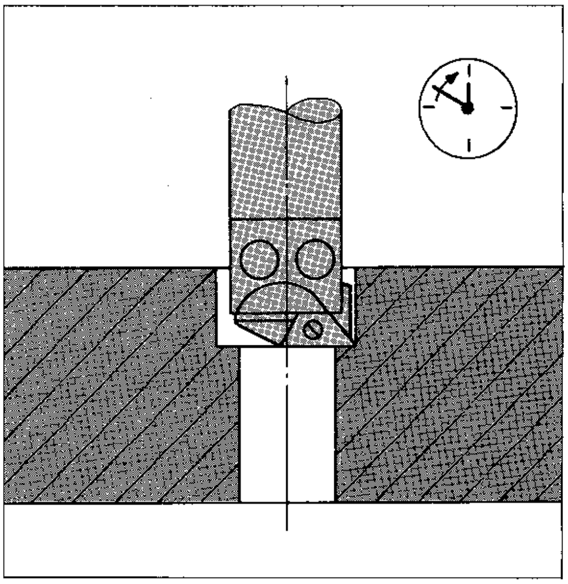
G04 Dwell time (dialogue-guided)

Block structure (example):

G04 F5

G04	Dwell time cycle
F...	Dwell time in secs.

Explanation of cycle, see „Dwell time".



Freely programmable cycle (Program call)

G39 Freely programmable cycle (dialogue guided)

Block structure (example):

G39 P01 12

G39	Freely programmable cycle (Program call)
P01	Program number

Explanation of cycle, see "Freely programmable cycle".

Program entry in ISO-format
Touch probe functions

Workpiece surface as datum

With TNC 155 as of software version ...... 06 and with TNC 151

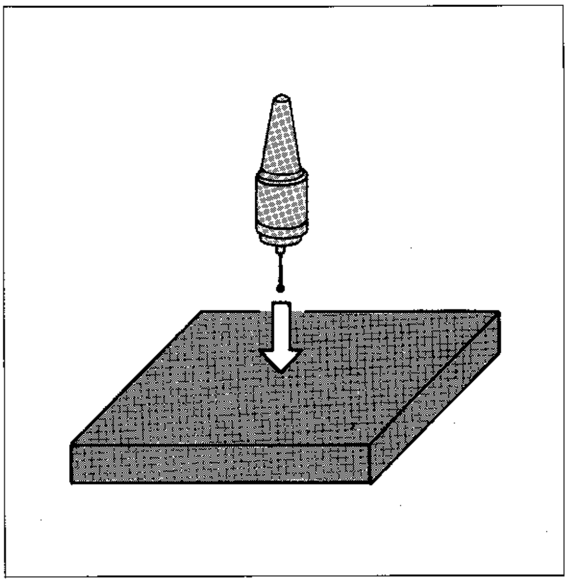
G 55 Touch probe function: Workpiece surface as datum (see "Touch probe system")

Block structure (example):

G55 P01 10 P02 Z– P03 G90

X+50.000 Y+50.000 Z–20.000

G55	Workpiece surface as datum
P01	Parameter number for result
P02	Approach axis and approach direction
P03	Probing point



Program entry in ISO-format
Subprograms and program part repeats

Label number

A **label number** is programmed with the command G98 L... This jump command may be programmed within any program block which does not contain a **label call.**

A **jump command** is programmed with the address L and a label number.

A jump command G98 L... and a label call L... should not be programmed within the same block.

Program label:

$$\textbf{N35}\quad \underbrace{\textbf{G98}\quad \textbf{L15}}\quad \textbf{G01...}$$

Label number 15

Label call:

N45 L15...

Part program

A part program is designated by G98 L... (label number) at the beginning.

The end of the program part repeat has a call-up L.... With **program part repeats,** the number of repetitions is entered after the label number. The label number and the repetition number are separated by a decimal point $\boxed{\cdot}$

e.g. L15.8, call-up label 15,
8 repetitions of program part

Program part:

N35 G98 L15 G01...

⋮

Program part repeat:

N70 L15.8

Subprogram

A subprogram is designated at the beginning by G98 L... (label number). It is ended with G98 L0 (label number 0).

A **subprogram call-up** is also made with the address L and the label number.

With a subprogram call, no repetitions should be programmed.

Subprogram:

N75 G98 L19 G00...

⋮

N90 G98 L0

Subprogram call:

N150 L19

Program entry in ISO-format
Jump into another main program/STOP-block

Jump into another main program

Programming of a jump into another main program is performed with the PGM CALL-key.
The control displays a jump into e.g. PGM 29 as follows:

N127 % 29

Further explanations, see "Program call".

For controls with software version 08:

STOP-block

G38 corresponds to a STOP-block in HEIDENHAIN plain language format.

Block structure example:

G38

Program entry in ISO-format
Parameter programming

Setting parameters

Parameters are markers fo numerical values which are related to units of measure. They are designated by the letter Q and a numeral. Entry (= setting) is performed with the Q -key.

D00 ≙ Assign
D01 ≙ Addition
D02 ≙ Subtraction
D03 ≙ Multiplication
D04 ≙ Division
D05 ≙ Square root
D06 ≙ Sine
D07 ≙ Cosine
D08 ≙ Root sum of square
D09 ≙ If equal, jump
D10 ≙ If unequal, jump
D11 ≙ If greater than, jump
D12 ≙ If less than, jump

Parameter definition

The assignment of a certain value or the correlation of a value through mathematical or logical functions is referred to as the **parameter definition.** A parameter definition consists of an **address D** and a code number (see adjacent table).
Entry of parameter definitions is dialogue-guided.

Block structure

A parameter definition requires one program block.
Individual **block elements** of a parameter definition comprise the **letter P** and a **number** (see also cycle parameter with canned cycles). The significance of these elements depends on the sequence within the block, which also depends on the entry dialogue. For checking, it is advisable to shift the cursor → ← within the block. The dialogue question is then displayed for each block element.

Program entry in ISO-format
Parameter programming

Example 1:

Q98 = $\sqrt{+2}$

D05 Q98 P01 +2

D05 Square root

Q98 Parameter to which result is assigned
P01 Parameter or numerical value within the square root

Example 2:

Q12 = Q2x62

D03 Q12 P01 +Q2 P02 +62

D03 Multiplication
Q12 Parameter to which result is assigned
P01 First factor (parameter or numerical value)
P02 Second factor (parameter or numerical value)

Example 3:

IF Q6 < Q5, jump to LBL 3

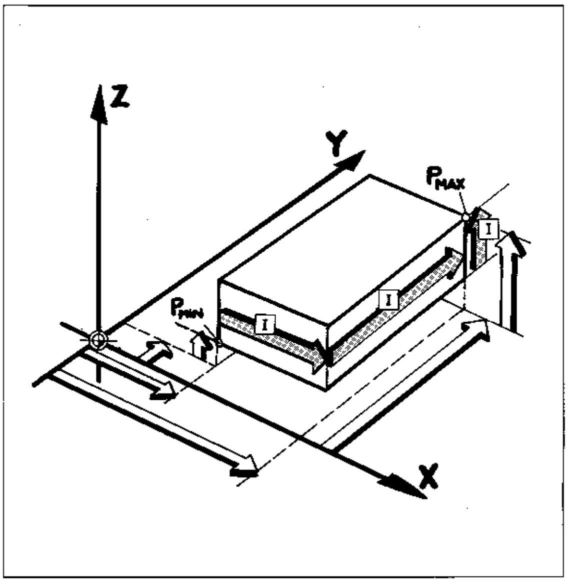
D12 P01 +Q6 P02 +Q5 P03 3

D12 If less than, jump
P01 First comparison value or parameter
P02 Second comparison value or parameter
P03 Label number (jump address)

Program entry to ISO-format
Graphics-Definition of BLANK FORM

Definition of blank

A workpiece blank (BLANK FORM) is defined by the points P_{MIN} and P_{MAX} (see "Blank form" (Graphics).
In addition to P_{MIN}, the tool axis must be specified via G17/G18/G19. If this has been neglected, the following error is displayed:
= BLK FORM DEFINITON INCORRECT =



Entry P_{MIN}

G30 Definition of P_{MIN} (entry only in absolute)

Block structure (example):

G30 G17 X+5 Y+5 Z–10

G30 Definition P_{MIN} (entry only in absolute)
G17 Plane definition and tool axis
X... X-co-ordinate of P_{MIN}
Y... Y-co-ordinate of P_{MIN}
Z... Z-co-ordinate of P_{MIN}

The function G90 (absolute dimensions) can be neglected if G30 has been programmed!

Entry P_{MAX}

G31 Definition of P_{MAX} (entry in either absolute or incremental)

Block structure (example):

G31 G91 X+95 Y+95 Z+10

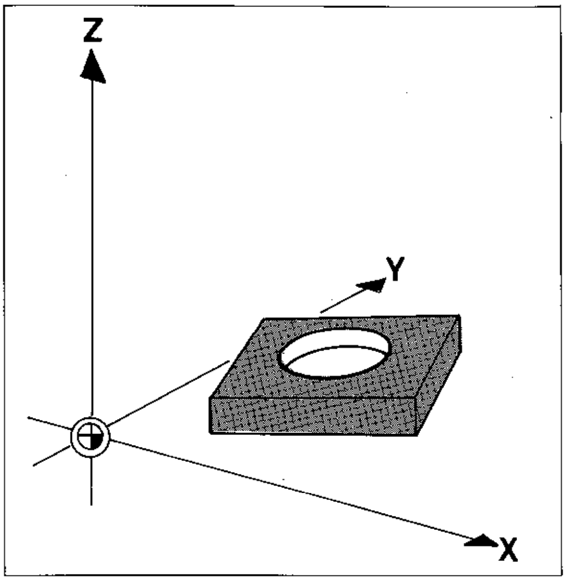
G31 Definition P_{MAX}
G91 Incremental dimensions
X... X-co-cordinate of P_{MAX}
Y... Y-co-cordinate of P_{MAX}
Z... Z-co-cordinate of P_{MAX}

Graphic simulation of workpiece machining can be stopped by pressing DEL □ .

Touch probe
Introduction

Touch probe

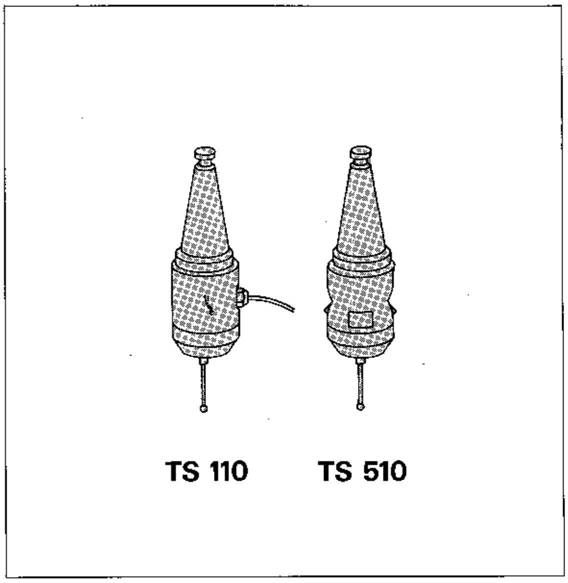
In conjunction with a HEIDENHAIN touch probe system, the TNC 155 as of software version ...... 06 and TNC 151 – control can detect deviations of workpiece attitude after the work has been clamped to the machine table. These deviations are stored and automatically compensated for during workpiece machining.
This dispenses with alignment procedures during workpiece set-up. A programmable probing function permits workpiece measurement either before or during machining. For example, the surfaces of cast workpieces with different heights can be probed in order that the correct depths can be obtained with subsequent machining.
Positional changes due to the temperature increase of the machine can be compensated at certain intervals of time.



Versions

Touch probe systems are available in two versions:
Touch probe 110 system with cable connection: Transmission of probe signals and operating voltage via a connecting cable.
The touch probe system 110 comprises the touch probe TS 110 and the mating electronics unit APE 110.
Touch probe 510 system with infra-red transmission and battery-power.
The touch probe system 510 comprises the touch probe TS 510 and the mating electronics unit APE 510 (including the transmitter/receiver unit).

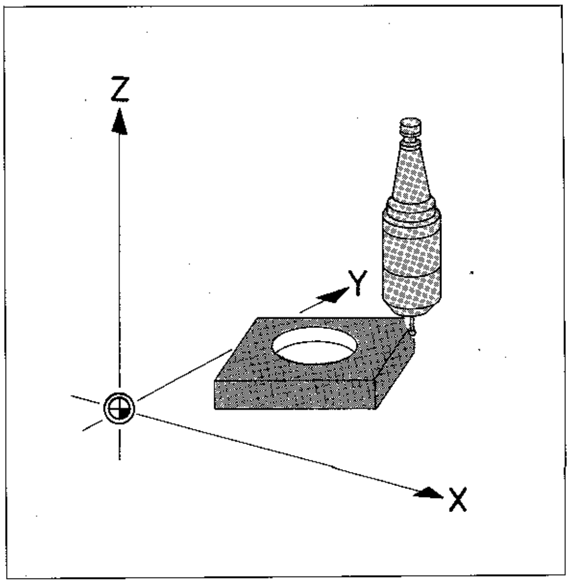
Each version has a standard tool shank enabling it to be inserted into the tool chuck. The probing head is interchangeable. Batteries for the TS 510 system with infra-red transmission have a life of 8 h in probing operation and 1 month in standby operation.



TS 510 has a transmitter and receiver window on one side (for the triggering signal) and a transmitter window displaced at 180°. When probing, the side with the transmitter/receiver window must face the transmitter/receiver-unit. The transmitter window which is displaced by 180° is not required for use with HEIDENHAIN-controls.

Operation

The touch probe is traversed to a side or the upper surface of the workpiece. The feed rate for probing and the max. probing distance has been set by the machine tool manufacturer via machine parameters. The probe signals physical contact with the workpiece to the control. The control then stores the co-ordinates of the probed points.
Workpiece surfaces, corners and circle centres can be easily determined with the touch probe and set as reference surfaces or datum points.



Touch probe
Dialogue initiation/Error messages

Dialogue initiation

The touch probe system is operational in the operating modes

⊗ electronic handwheel

✋ manual

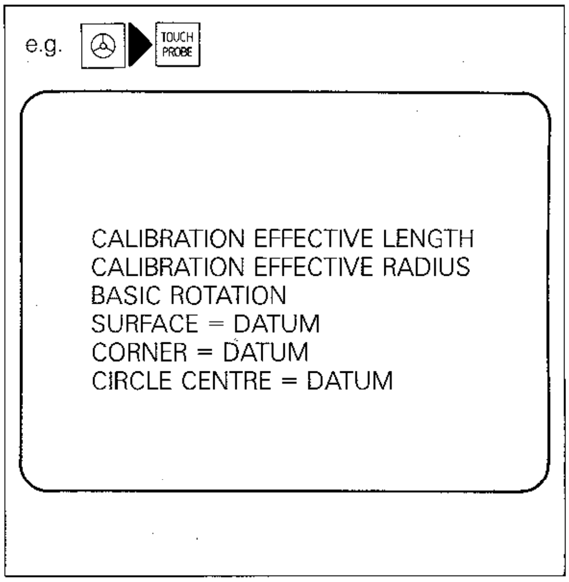
⇒ → single block/automatic program run

Dialogue is opened with the TOUCH PROBE-key.

In the ⊗ or ✋ mode the adjacent menu of **touch probe functions** is displayed.

The desired function is selected via the ↓ ↑-keys and transferred by pressing ENT.

In the ⇔-mode the dialogue for the touch probe function "workpiece surface = datum" after dialogue initiation with TOUCH PROBE.



e.g. ⊗ ▶ TOUCH PROBE

CALIBRATION EFFECTIVE LENGTH
CALIBRATION EFFECTIVE RADIUS
BASIC ROTATION
SURFACE = DATUM
CORNER = DATUM
CIRCLE CENTRE = DATUM

Cancellation of touch probe functions

Touch probe functions can be ended at any time by pressing NO ENT. The control then returns to the previous operating mode.

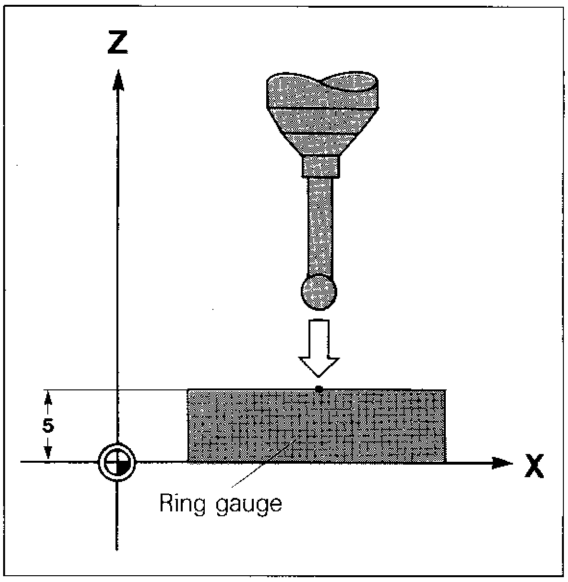
Error messages

If the touch probe is unable to find a suitable probing point within the defined travel (via machine parameters) or if a probing point is already reached when a touch probe function is started, the following error is displayed:
= TOUCH POINT INACCESSIBLE =
Touch probe systems with **infra-red transmission** have to be set such, that the transmitter/receiver window (i.e. the side with two windows) is adjusted to the evaluation electronics. Insufficient adjustment or an interruption of the transmission range (e.g. splash shield) initiates the following error message:
= PROBE SYSTEM NOT READY =
If the battery voltage for the infra-red version drops by a certain value, the following error is displayed:
= EXCHANGE TOUCH PROBE BATTERY =

Touch probe
Calibration of effective length

Introduction

The effective length of the probing stylus and the effective radius of the stylus tip can be determined with the aid of the control.
The necessary data are automatically calculated by the control via the probing functions "Calibration of effective length" and "Calibration of effective radius".
The length and the radius are stored by the control and are automatically taken into account during probing operations.
Compensation values can be entered at any time via the control keyboard.



Auxiliary equipment

For calibration of the effective radius, a ring gauge with a known height and internal radius is required. The ring gauge must be clamped to the machine table.

Effective length

The effective length is determined by probing a reference plane. On touching the surface, the touch probe is withdrawn to its starting position in rapid traverse.
Display of the effective length is activated upon selection of the next calibration.

Before calibrating the effective length, set the reference surface with the zero-tool.

Touch probe
Calibration of effective length

Entry

Operating mode ____________ ✋ or ⊛

Dialogue initiation ____________ TOUCH PROBE ↑

CALIBRATION EFFECTIVE LENGTH	▶ ENT	Enter touche probe function

▼

CALIBRATION EFFECTIVE LENGTH		
Z+ Z–		
TOOL AXIS = Z	▶ Y	If reqd. enter tool axis.
DATUM +0.000		
EFFECT. PROBE RADIUS = 0.000		
EFFECTIVE LENGTH = 0.000		

▼

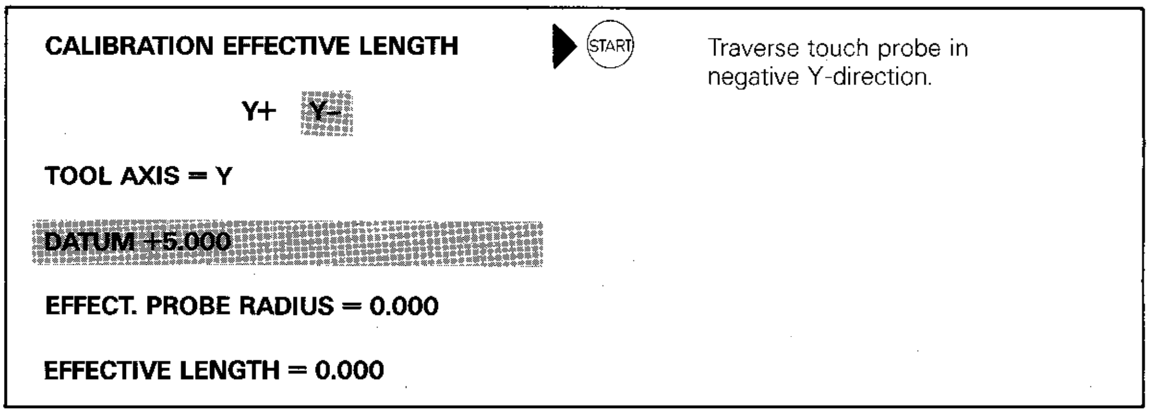
CALIBRATION EFFECTIVE LENGTH	▶ X Y Z ▼	Traverse touch probe to within the vicinity of the reference plane.
Y+ Y–		
TOOL AXIX = Y		
DATUM +0.000	▶ ↓ ▼	If reqd. enter datum: select "datum"
EFFECT. PROBE RADIUS = 0.000	□ ▼	Enter datum in the tool axis, e.g. + 5.0 mm
EFFECTIVE LENGTH = 0.000	ENT	Enter into memory

▼

CALIBRATION EFFECTIVE LENGTH		
Y+ Y–	▶ ← →	If reqd. select traversing direction of touch probe, here Y–.
TOOL AXIS = Y		
DATUM +5.000		
EFFECT. PROBE RADIUS = 0.000		
EFFECTIVE LENGTH = 0.000		

▼

Touch probe
Calibration of effective length



CALIBRATION EFFECTIVE LENGTH

Y+ Y–

TOOL AXIS = Y

DATUM +5.000

EFFECT. PROBE RADIUS = 0.000

EFFECTIVE LENGTH = 0.000

▶ START

Traverse touch probe in negative Y-direction.

After touching the surface, the touch probe is retracted to its starting position in rapid traverse.

MANUAL OPERATION

The control automatically switches to the display "Manual operation" or "Electronic handwheel"

Display of the calibrated length is activated after selection of the next calibration.

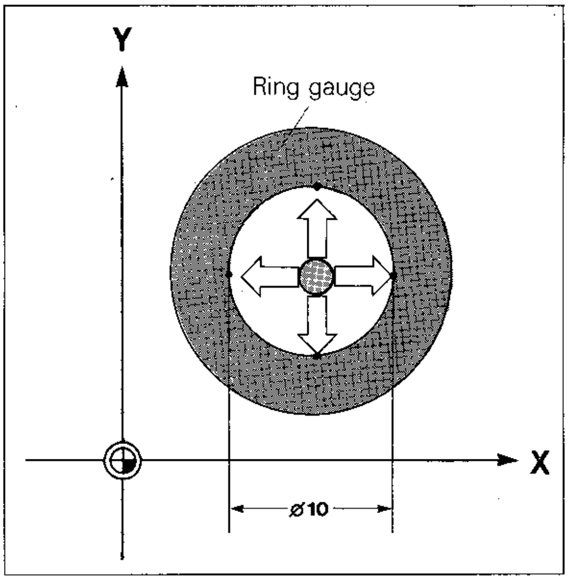
Remarks

Touch probe
Calibration of effective probe radius

Effective radius

The touch probe tip must be located within the bore of the ring gauge. Calculation of the effective radius is performed by touching 4 points of the bore. The traversing directions are specified by the control, e.g. X+, X–, Y+, Y– (tool axis = Z).
After every touch sequence the touch probe is retracted to its starting position. The control displays the co-ordinates of all touch points.

The effective radius is displayed after re-selection of the calibration.



Touch probe
Calibration of effective probe radius

Entry

Operating mode ______ ✋ or ⎈

Dialogue initiation ______ TOUCH PROBE ↑

CALIBRATION EFFECTIVE RADIUS	▶ ENT	Enter touch probe function.

▼

CALIBRATION EFFECTIVE RADIUS	▶ ↓	Select "Radius ring gauge".
X+ X– Y+ Y–	▼ □	Enter ring gauge radius, e.g. 10.0 mm
TOOL AXIS = Z	▼ ENT	Enter into memory
RADIUS RING GAUGE = 0.000		If reqd. enter another tool axis (see "effective length")
EFFECT. PROBE RADIUS = 0.000		
EFFECTIVE LENGTH = 8.455		

▼

CALIBRATION EFFECTIVE RADIUS	▶ X Y Z	Traverse to approximate centre of ring gauge.
X+ X– Y+ Y–	▼ ← →	Select traversing direction of touch probe, e.g. X+.
TOOL AXIS = Z		
RADIUS RING GAUGE = 10.000		
EFFECT. PROBE RADIUS = 0.000		
EFFECTIVE LENGTH = 8.455		

▼

CALIBRATION EFFECTIVE RADIUS	▶ START	Traverse touch probe in the positive X-axis.
X+ X– Y+ Y–		
TOOL AXIS = Z		
RADIUS RING GAUGE = 10.000		
EFFECT. PROBE RADIUS = 0.000		
EFFECTIVE LENGTH = 8.455		

▼

Touch probe
Calibration of effective radius

After touching the ring gauge, the touch probe is retracted to its starting position in rapid traverse.

CALIBRATION EFFECTIVE RADIUS

X+ X– Y+ Y–

X (touch point) Y (touch point)

Z (touch point) C (touch point)

▶ ← → Select next traversing direction of touch probe, e.g. X–.

CALIBRATION EFFECTIVE RADIUS

X+ X– Y+ Y–

X (touch point) Y (touch point)

Z (touch point) C (touch point)

▶ START Traverse touch probe in negative X–direction.

After touching the ring gange, the touch probe is retracted to its starting position in rapid traverse.

The control displays the actual values of the second touch point beneath the values of the first point.

Finally, the ring gauge is touched in the positive and negative Y-direction.

After this procedure:

MANUAL OPERATION

The control automatically switches to the display "Manual operation" or "Electronic handwheel".

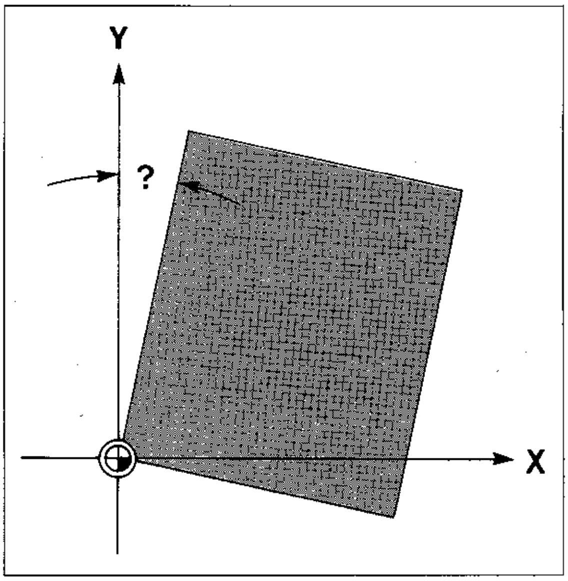
Display of the calibrated probe radius is activated after re-selection of the calibration in the appropriate line.

Remarks

Touch probe
Basic rotation

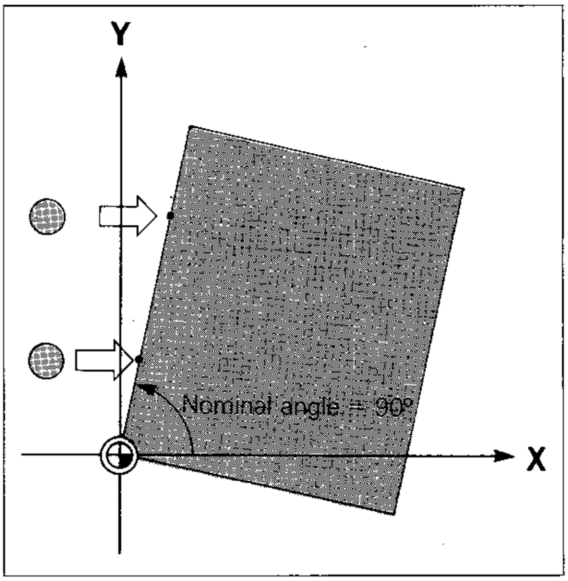
Description

The touch probe function "basic rotation" is used for detecting the angular misalignment of the workpiece attitude after it has been clamped and non-aligned to the machine table.



Procedure

The touch probe traverses to a side face of the workpiece from two different starting positions. The traversing directions are pre-determined, e.g. X+, X–, Y+, Y– (Tool axis = Z).
After touching the side face the touch probe returns to the appropriate starting position in rapid traverse.
The control stores the co-ordinates of the touch points and calculates the angular deviation. For compensation of this deviation, the control must know the "nominal angle" of this side face. The nominal angle is entered into the line after "ROTATION ANGLE".



Touch probe
Basic rotation

Entry

Operating mode ________________ ✋ or ⊕

Dialogue initiation ________________ TOUCH PROBE ↑

BASIC ROTATION

▶ ENT — Enter touch probe function.

BASIC ROTATION

X+ X– Y+ Y–

ROTATION ANGLE = 0.000

▶ □ Enter angle attitude of side faces to be probed, e.g. Y-axis: + 90°.

ENT — Enter into memory.

BASIC ROTATION

X+ X– Y+ Y–

ROTATION ANGLE = + 90.000

▶ X Y Z — Traverse to first starting position

← → — select traversing direction, e.g. X+

BASIC ROTATION

X+ X– Y+ Y–

ROTATION ANGLE = + 90.000

▶ START — Traverse touch probe in positive X-direction.

After touching the side face, the touch probe is returned to its starting position in rapid traverse.

BASIC ROTATION

X+

X (touch point) **Y (touch point)**

Z (touch point) **C (touch point)**

ROTATION ANGLE = + 90.000

▶ X Y — Traverse touch probe to second starting position.

Touch probe
Basic rotation

BASIC ROTATION

X+

X (touch point) **Y (touch point)**

Z (touch point) **C (touch point)**

ROTATION ANGLE = + 90.000

▶ START Traverse touch probe in positive X-direction.

After touching the side face, the touch probe is returned to the second starting position in rapid traverse.

MANUAL OPERATION

The control automatically switches to the display "Manual operation" or "Electronic handwheel".

Display of the calibrated rotation angle is activated after re-selection of the basic rotation.

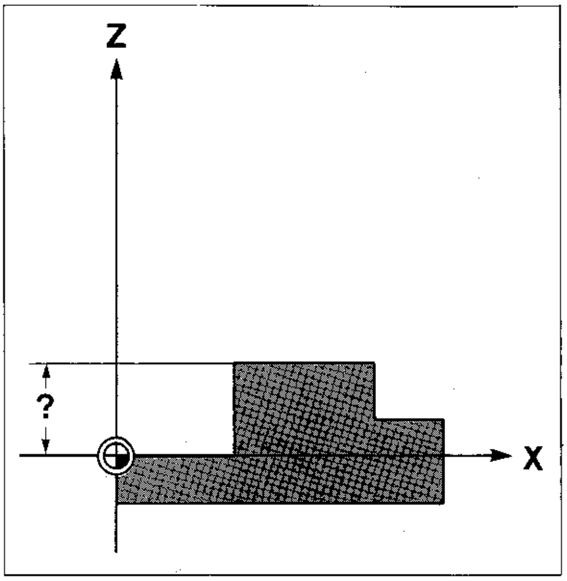
"Basic rotation" is cancelled by selecting the touch probe function "Basic rotation" and entry of the rotation angle 0° via the keyboard.

If a "basic rotation" was programmed, the status display indicates **ROT** in inverted characters (bright background).
This display remains as long as the "basic rotation" is stored within the memory. The "basic rotation" is not erased in the event of power switch-off.

Touch probe
Surface = Datum

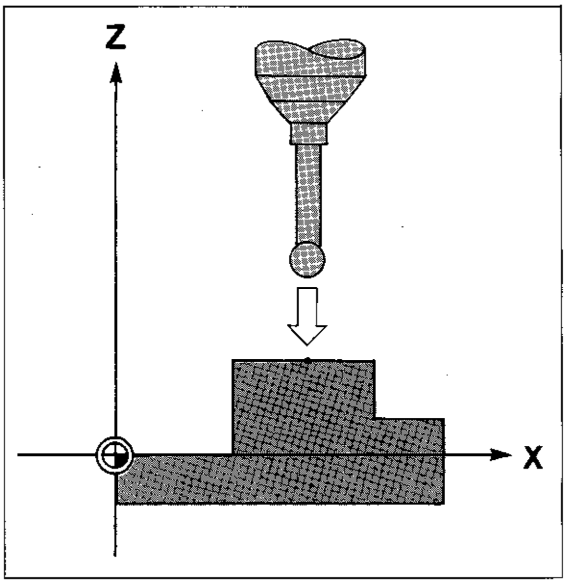
Description

On workpieces which have been clamped parallel to the axes, the upper surface or a side face can be set as a datum by using the touch probe function "Surface = Datum".
During machining, the control then references all subsequent nominal position values to this surface.



Procedure

The touch probe is traversed to the surface or face in question.
After touching the surface, the touch probe is returned to the starting position in rapid traverse.
The control stores the co-ordinates of the touch point in the traversing axis and displays the value in the display line "DATUM".
Any value may be allocated to the touch point by using the control keyboard.



Touch probe
Surface = Datum

Entry

Operating mode ____________ ✋ or ⊛

Dialogue initiation ____________ TOUCH PROBE

SURFACE = DATUM	▶ ENT	Enter touch probe function.

▼

SURFACE = DATUM	▶ X Y Z	Traverse to starting position
X+ X– Y+ Y– Z+ Z– C+ C–	▼ ← →	select traversing direction, e.g. Z–.

▼

SURFACE = DATUM	▶ START	Traverse touch probe in the negative Z-direction.
X+ X– Y+ Y– Z+ Z– C+ C–		

▼

After touching the surface, the touch probe is returned to its starting position in rapid traverse.

▼

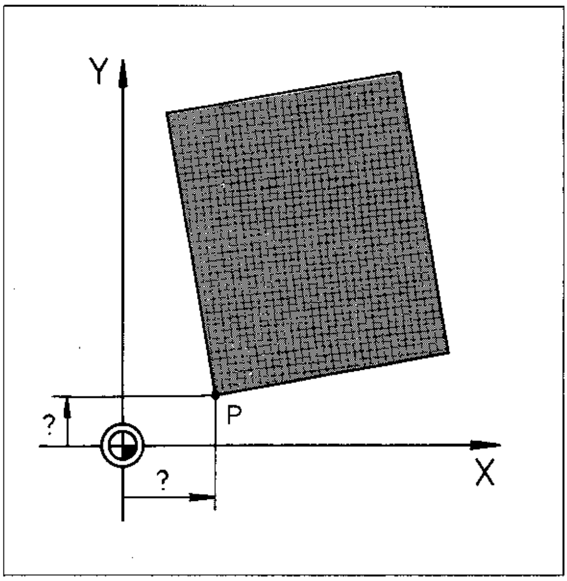
SURFACE = DATUM		
X (touch point) Y (touch point)		
Z (touch point) C (touch point)		
DATUM Z – 18.125	▶ □	If reqd. enter random datum value.
	▼ ENT	Enter into memory.

Remarks

Touch probe
Corner = Datum

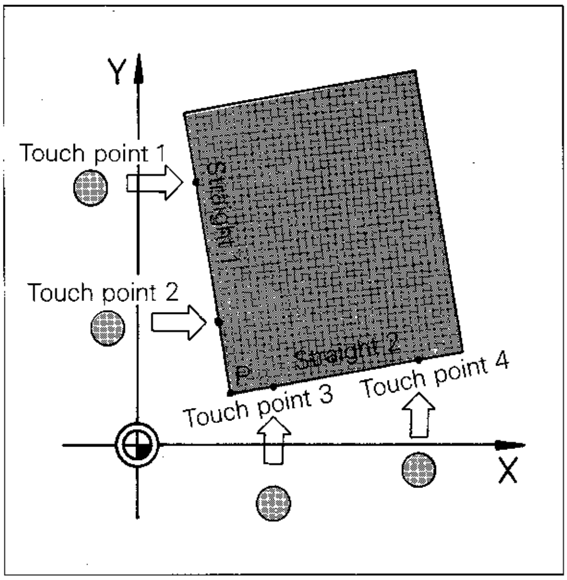
Description

With the touch probe function "Corner = Datum", the control calculates the co-ordinates of the corner point of a clamped workpiece.
The calculated value can be used as a datum for subsequent machining. All nominal position values are then referenced to this point.

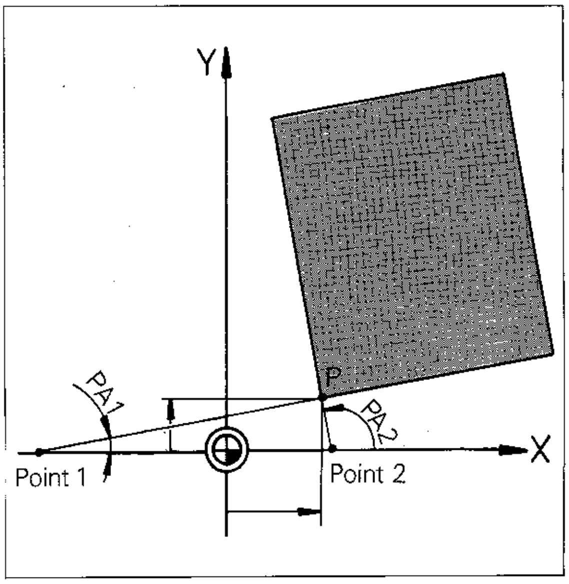


Procedure

The touch probe touches two intersecting faces of a workpiece from two independent starting points for each face. The traversing directions are given:
X+, X–, Y+, Y– (Tool axis = Z).
After touching the side face, the touch probe is returned to the starting position in rapid traverse. The control stores the co-ordinates of the touch points and calculates two straight lines. The intersection of these lines is the required corner point.



The control display indicates the co-ordinates of the corner point. The calculated lines are indicated beneath by a point of each line and the appropriate angle PA.
Instead of the calculated corner point, a datum value may be set via the control keyboard.
If a "Basic rotation" was calculated prior to the "Corner = Datum"-function, the straight line data which was defined for the "Basic rotation" may be utilized for the "Corner = Datum"-function.



Touch probe
Corner = Datum

Entry

Operating mode ________________ ✋ or ⊕

Dialogue initiation ________________ TOUCH PROBE ↓

CORNER = DATUM	▶ ENT	Enter touch probe function.

▼

CORNER = DATUM	▶ X Y Z	Traverse to first starting position
X+ X– Y+ Y–	▼ ← →	select traversing direction, e.g. X+.

▼

CORNER = DATUM	▶ START	Traverse touch probe in the positive X-direction.
X+ X– Y+ Y–		

▼

After touching the side face, the touch probe is returned to its starting position in rapid traverse.

▼

CORNER = DATUM	▶ X Y	Traverse to next starting position.
X+		
X (touch point 1) Y (touch point 1)		
Z (touch point 1) C (touch point 1)		

▼

CORNER = DATUM	▶ START	Traverse touch probe in positive X-direction.
X+		
X (touch point 1) Y (touch point 1)		
Z (touch point 1) C (touch point 1)		

▼

After touching the side face the touch probe is returned to its starting position in rapid traverse.

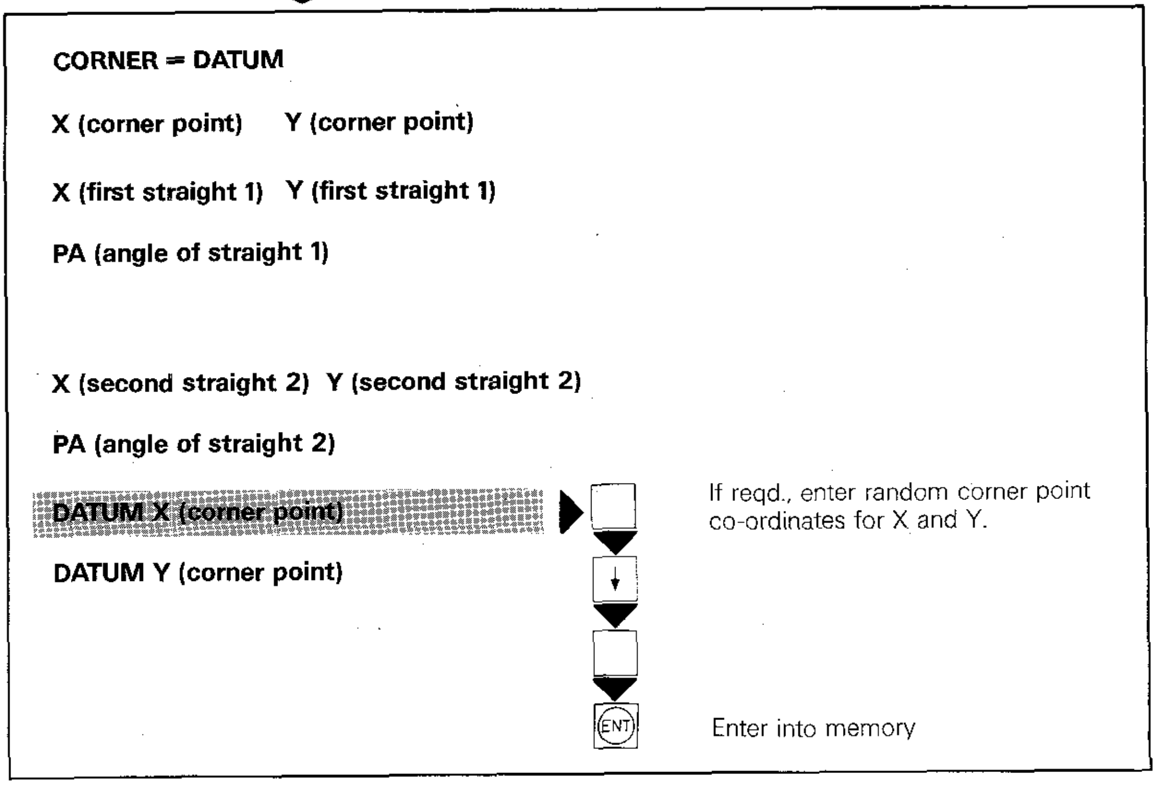
The control displays the actual values of the second touch point beneath the values of the first point. In addition, the first straight line is indicated by a random point on the straight line and direction angle.

▼

Touch probe
Corner = Datum

Finally, the second side face is to be probed from two different starting positions.

On completion of this:



CORNER = DATUM

X (corner point) Y (corner point)

X (first straight 1) Y (first straight 1)

PA (angle of straight 1)

X (second straight 2) Y (second straight 2)

PA (angle of straight 2)

DATUM X (corner point) □ If reqd., enter random corner point co-ordinates for X and Y.

DATUM Y (corner point) ↓

□

ENT Enter into memory

Remarks

Touch probe
Corner = Datum

Entry immediately after a "Basic rotation"

Operating mode ____________ ✋ or ⎈

Dialogue initiation ____________ TOUCH PROBE ↓

CORNER = DATUM ▶ ENT Enter touch probe function

▼

CORNER = DATUM

TOUCH POINTS OF BASIC ROTATION?

X (straight 1) Y (straight 1)

PA (angle of straight)

If touch points for the basic rotation are to be utilized: ▶ ENT Enter data

If touch points for the basic rotation are not to be utilized: ▶ NO ENT No enter

▼

Afterwards, probe second side face as described above.

▼

CORNER = DATUM ▶ ____________

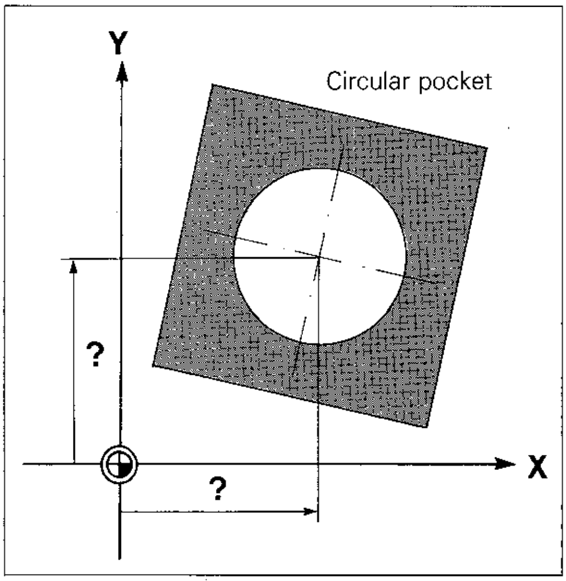
X+ X– Y+ Y–

Remarks

Touch probe
Circle centre = Datum

Description

The centrepoint co-ordinates of a clamped workpiece with cylindrical surfaces (bore, circular pocket or external cylinder) can be determined by the touch probe function "circle centre = Datum".
The calculated centrepoint can be used as a datum for subsequent machining. All position values can then be referenced to this position.



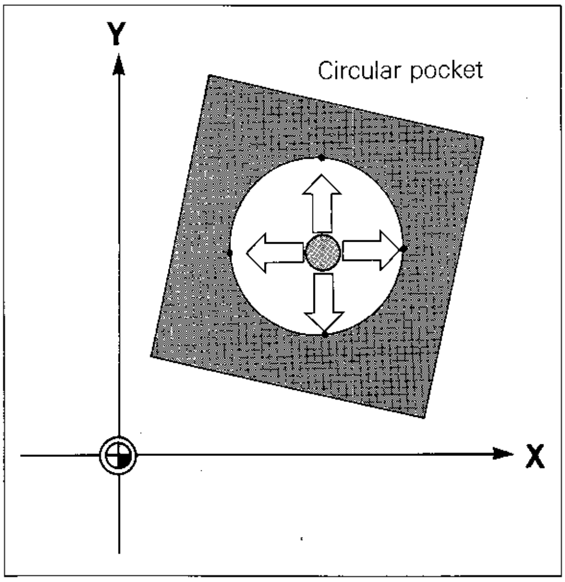
Procedure

With internal bores, the touch probe must have access into the bore.

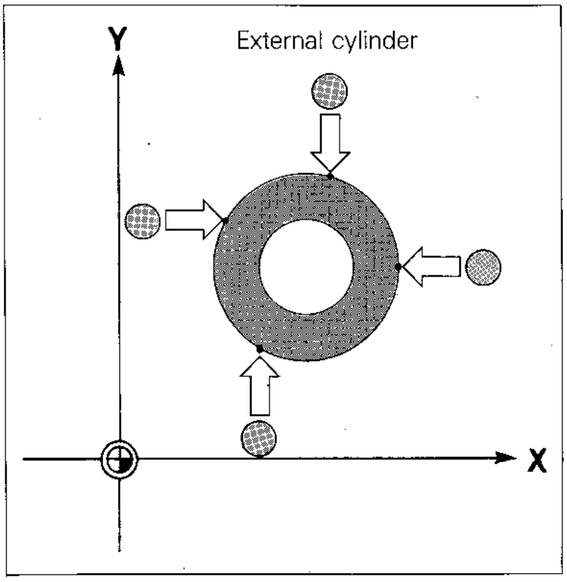
The circle centre is determined by touching 4 independent points on the circumference of the bore or external cylinder. Traversing directions are predetermined, e.g. X+, X–, Y+, Y– (tool axis = Z).

After every touch procedure, the touch probe is retracted to the starting position in rapid traverse. The control calculates the co-ordinates of all four points and then derives the co-ordinates of the centrepoint.

The display indicates the co-ordinates of the circle centre and the radius PR.



Instead of the calculated centrepoint co-ordinates a random datum may also be set via the control keyboard.



Touch probe
Circle centre = Datum

Entry

Operating mode ________________ ✋ or ⊛

Dialogue initiation ________________ TOUCH PROBE ↓

CIRCLE CENTRE = DATUM	▶ ENT	Enter touch probe function.

▼

CIRCLE CENTRE = DATUM	▶ X Y Z	Traverse to first starting position.
X+ X– Y+ Y–	▼ ← →	Select traversing direction, e.g. X+.

▼

CIRCLE CENTRE = DATUM **X+ X– Y+ Y–**	▶ START	Traverse touch probe in positive X-direction.

▼

After touching the cylindrical surface, the touch probe is returned to the starting position in rapid traverse.

▼

CIRCLE CENTRE = DATUM **X+ X– Y+ Y–** **X (touch point 1) Y (touch point 1)** **Z (touch point 1) C (touch point 1)**	▶ ← →	Select next traversing direction, e.g. X–.

▼

CIRCLE CENTRE = DATUM **X+ X– Y+ Y–** **X (touch point 1) Y (touch point 1)** **Z (touch point 1) C (touch point 1)**	▶ START	Traverse touch probe in negative X-direction.

▼

After touching the cylindrical surface, the touch probe is returned to its starting position in rapid traverse.

The control displays the actual values of touch point 2.

▼

Touch probe
Circle centre = Datum

Afterwards, two further points of the cylindrical surface are traversed to in positive and negative Y-directions.

When this is completed:

CIRCLE CENTRE = DATUM

X (centrepoint) **Y (centrepoint)**

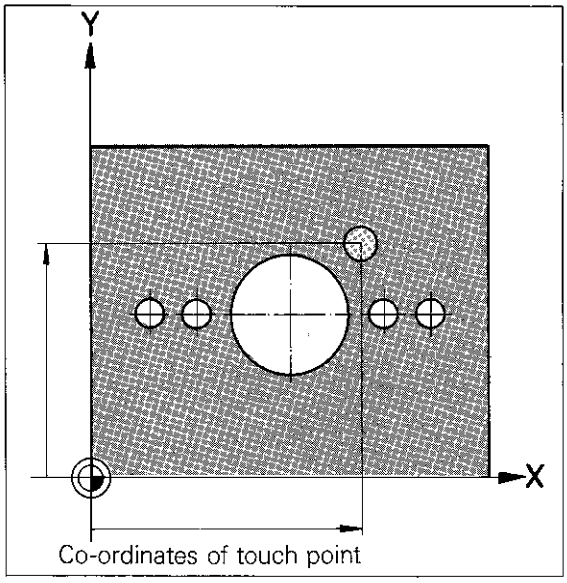
PR (circle radius)

DATUM X (centrepoint)	▶ □	If reqd. key-in random co-ordinates for X and Y.
DATUM Y (centrepoint)	↓	
	□	
	ENT	Enter into memory.

Touch probe
Programmable touch probe function "Surface = Datum"

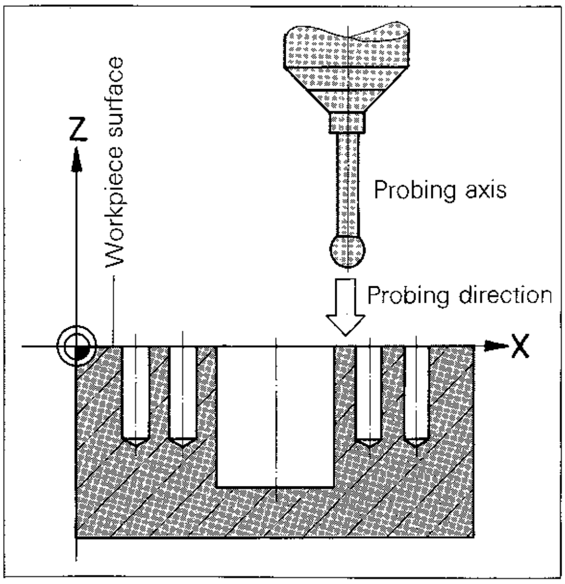
Description

Before or during workpiece machining it is possible to probe a workpiece surface in controlled operation. As an example, the surface of cast workpieces with varying heights can be touched in order to ensure that the correct depth is obtained with subsequent machining. Furthermore, positional changes due to temperature increases of the machine and workpiece can also be detected and compensated.



Programming

Programming is initiated via the TOUCH PROBE-key. The control then asks for the parameter number to which the result of the touch probe calibration is to be allocated. After entry of the probing axis and probing direction, the nominal position value for execution of the touch probe cycle is to be entered. The programmed touch probe cycle allocates two program blocks.



Procedure

The touch probe traverses in rapid to the nominal position (touch point) which has been programmed in the touch probe cycle, however only to the safety clearance before the position. The safety clearance is determined by the machine tool builder via a machine parameter.

Afterwards, the workpiece is traversed in the probing axis and probing direction with the feed rate for measurement until the surface is touched. After touching, the touch probe returns to the starting position in rapid traverse.

To compensate deviations of attitude in the workpiece surface, the zero-datum must be shifted in the probing axis by the stored Q-value via a datum shift procedure. The measured value can, e.g. be utilized as a length compensation value in a tool definition.

Touch probe
Programmable touch probe function "Surface = Datum"

Entry

Operating mode ______________ ◈

Dialogue initiation ______________ TOUCH PROBE

PARAMETER NUMBER FOR RESULT?	▶ □	Key-in parameter number.
	▼ ENT	Enter into memory.

▼

PROBING AXIS/PROBING DIRECTION?	▶ Z	Key-in probing axis, e.g. Z.
	▼ +/−	Key-in probing direction.
	▼ ENT	Enter into memory.

▼

POSITION VALUE?	▶ X	Key-in co-ordinates of touch point: Select axis, e.g. X.
	▼ I	Incremental-Absolute?
	▼ □	Key-in numerical value.
	▼ Y	Select next axis, e.g. Y.
	⋮	
After entry of all co-ordinates;	▼ ENT	Enter into memory

Display example

```
32 TCH PROBE 0.0 REF. PLANE

                 Q 10 Z–

33 TCH PROBE 0.1 X + 10.000

           Y + 20.000 Z + 0.000
```

The X-, Y-plane is probed in the negative Z-direction. The measured value is stored under the parameter allocation Q10. The nominal touch point has the co-ordinates X 10.000/Y 20.000/Z 0.000.

Remarks

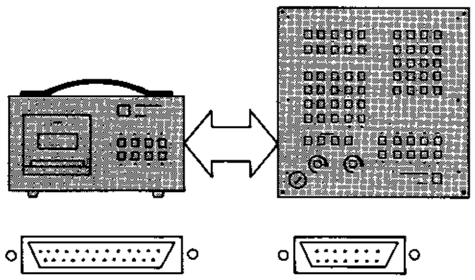
External data transmission

Interface V.24/RS-232-C

The TNC 151/TNC 155 is equipped with a **V.24-data interface (RS-232-C)** for read-in and read-out of programs in plain language or ISO-format.

This means that programs within the TNC 155-memory can be transferred via this interface to an **external storage unit,** e.g. magnetic tape unit, or another **peripheral unit,** e.g. a printer. Data can also be transferred from an external storage unit into the control.

The interface connection is located at the rear of the control.



Baud rate

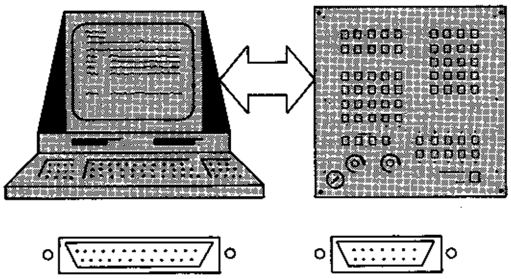
The **data transmission rate** (= Baud rate) for external storage units is automatically set to 2400 Baud.
Data units with other Baud rates can also be connected (see adjacent table); but for this, the Baud rate of the control must be re-programmed.

Possible Baud rates:
110 Baud
150 Baud
300 Baud
600 Baud
1200 Baud
2400 Baud
4800 Baud
9600 Baud

1 Baud = 1 bit/sec.

Transfer blockwise

The TNC 151/TNC 155 can receive machining programs from an external station via the V.24 data interface. The external station has the superior function of a host computer governing program management, program assignment and the transmission.



External data transmission
Magnetic tape unit

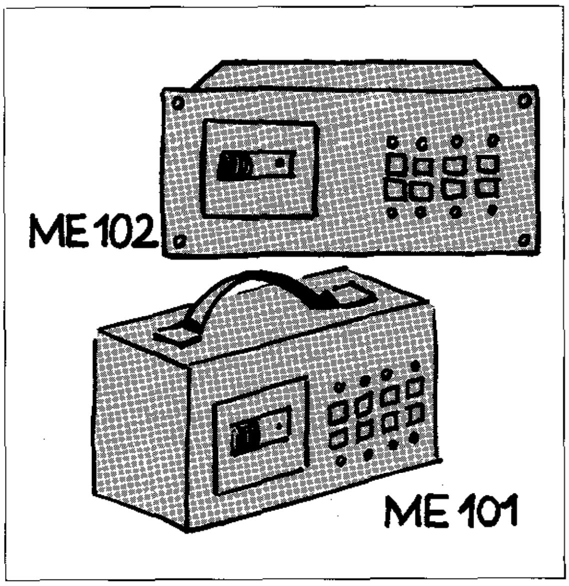
Magnetic tape unit

The magnetic tape unit is used for external program storage or transfer of programs which have been compiled on an off-line programming station.

There are two versions available:

ME 101: Portable unit for use on several machines

ME 102: Pendant type for permanent installation on one machine



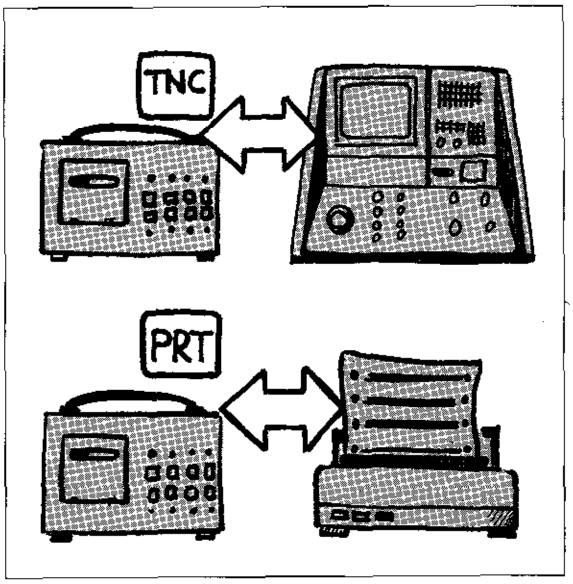
Connections

ME 101 and ME 102 each have two V.24-data interfaces with the designations **TNC** and **PRT.**

TNC-connection: for connection of magnetic tape unit-control.

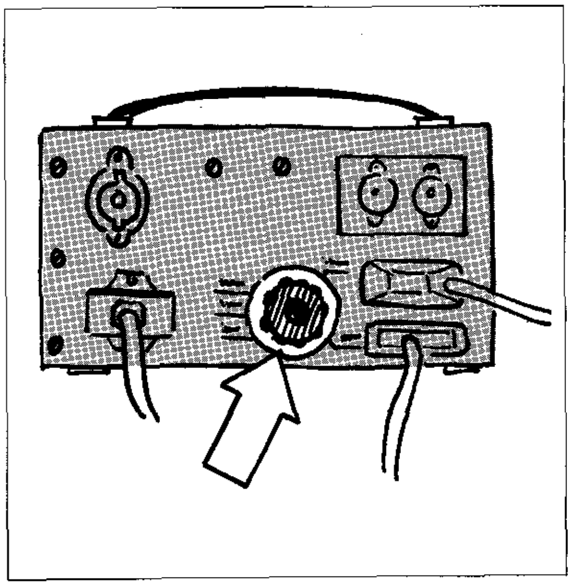
PRT-connection: for connection of magnectic tape unit – to – peripheral unit

These interfaces permit the connection of a second unit in addition to the TNC-control.



Transmission rate

The data transmission rate between the **TNC-control** and the **magnetic tape unit** has been set to 2400 Baud. The transmission rate between a **peripheral unit** and the **magnetic tape unit** can be adapted via the selector switch on the rear of the magnetic tape unit.
Possible Baud rates:
110 / 150 / 300 / 600 / 1200 / 2400 Baud



External data transmission
Changing the Baud rate

Entry of Baud rate

Operating mode ______ □ optional

Dialogue initiation ______ MOD

VACANT BLOCKS = ... ▶ MOD ↓ Page supplementary modes until BAUD RATE is displayed.

▼

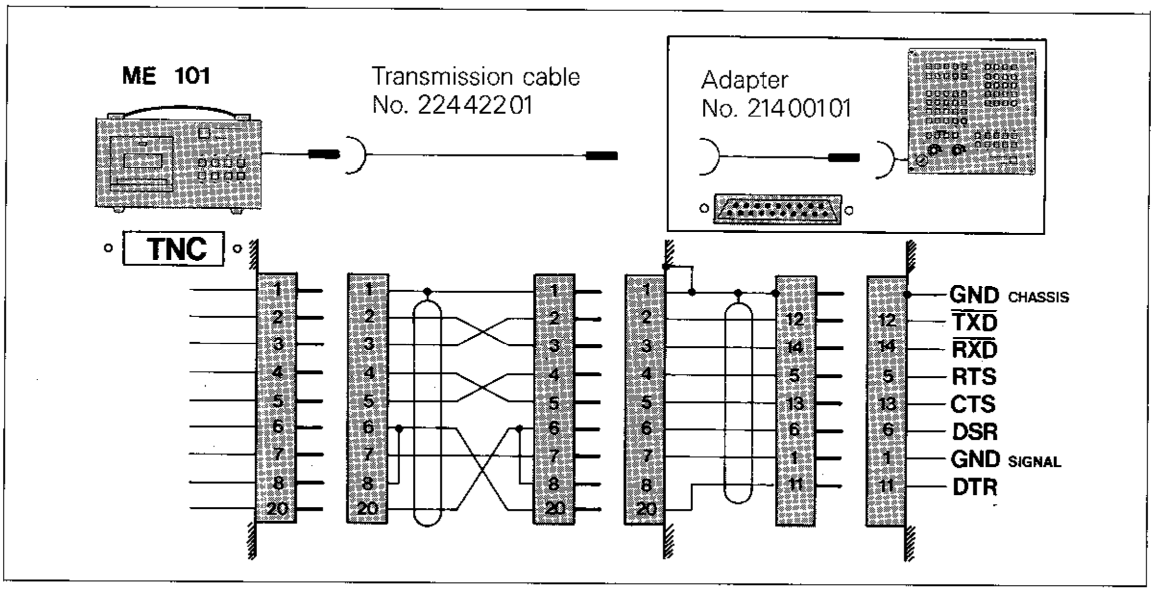
BAUD RATE = 2400 ▶ □ Key-in Baud rate according to table.

▼ ENT Enter into memory.

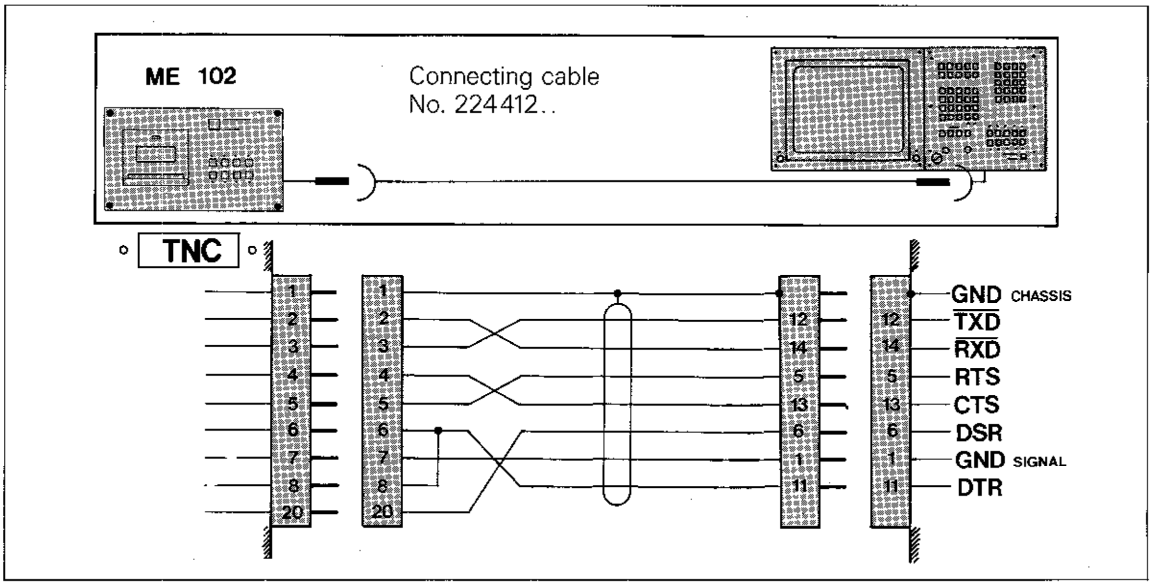
Entry of the new Baud rate can be transferred with the MOD or the ↓ ↑ -keys.

External data transmission
Cables and connections

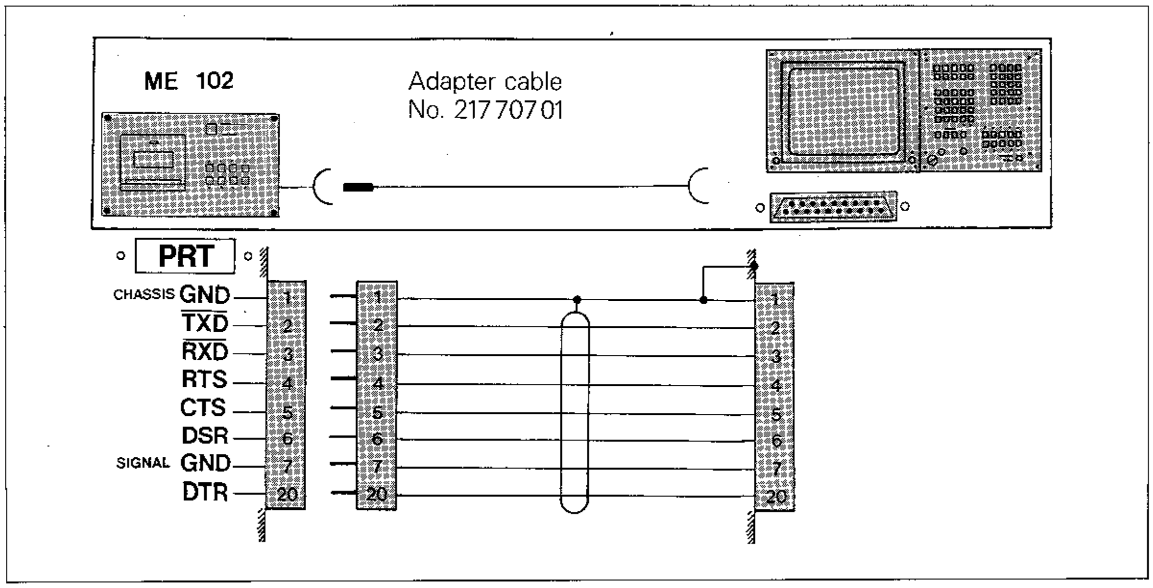
Magnetic tape unit ME 101 – TNC



Magnetic tape unit ME 102 – TNC

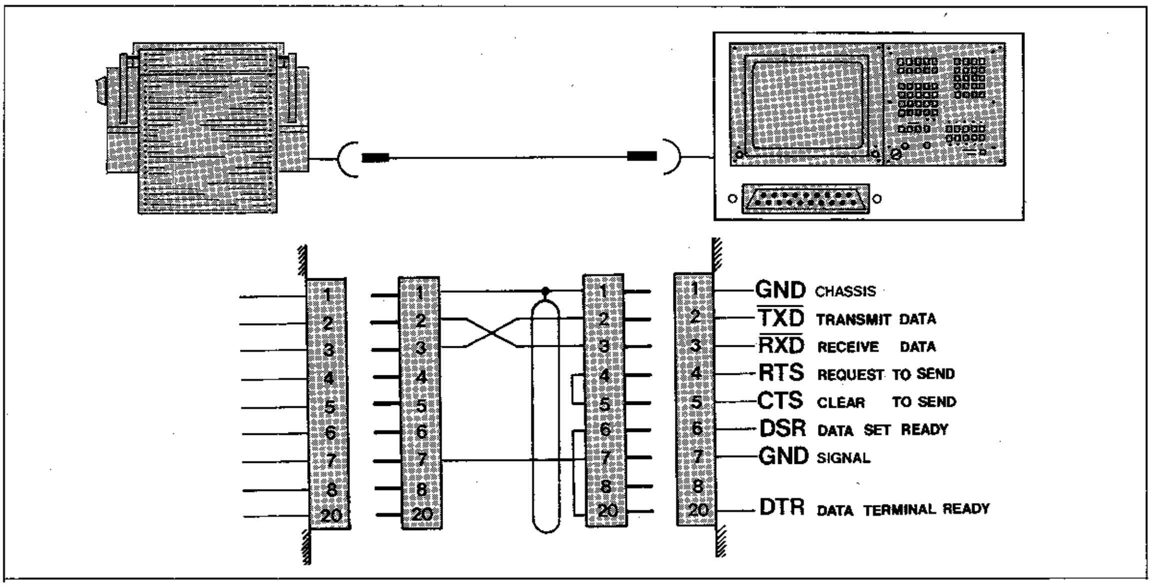


Magnetic tape unit ME 102 – PRT



External data transmission
Cables and connections

Magnetic tape unit/TNC – Peripheral unit (e.g. printer)



External data transmission Operation

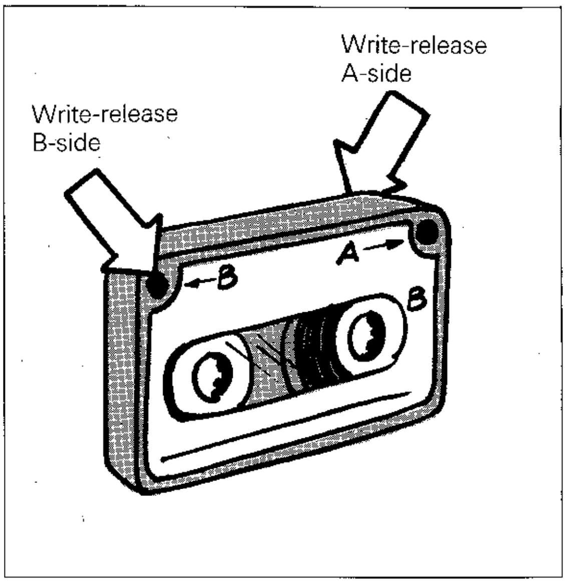
Data transmission ME ↔ TNC

Program management of the control permits the **transfer of individual programs** from tape to the TNC and vice-versa.
Max. 32 programs can be stored on one side of a magnetic tape cassette. If a program which exceeds this capacity is read-in or read-out, the following message is displayed:

= EXCHANGE CASSETTE – ME START =

After exchanging the cassette and re-starting the magnetic tape unit via START the remaining program blocks are transferred.

Data transmission is only possible when the write-release plug is in the cassette.

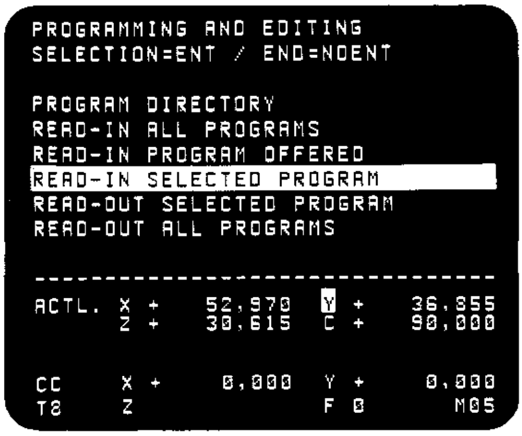


Dialogue initiation

Data transmission can only be performed in the programming mode ⟐. Dialogue for the transfer direction (tape → TNC or TNC → tape) is initiated with the EXT-key. The display indicates the adjacent transfer modes for selection.

The cursor can be set to the required mode via the ↓ ↑-keys. Mode start is activated by pressing ENT.

Mode cancellation is performed with NO ENT.



```
PROGRAMMING AND EDITING
SELECTION=ENT / END=NOENT

PROGRAM DIRECTORY
READ-IN ALL PROGRAMS
READ-IN PROGRAM OFFERED
READ-IN SELECTED PROGRAM
READ-OUT SELECTED PROGRAM
READ-OUT ALL PROGRAMS

-------------------------------------
ACTL.  X +    52,970  Y +    36,855
       Z +    30,615  C +    90,000

CC     X +     0,000  Y +     0,000
T8     Z              F 0       M05
```

Interruption of data transmission

Data transmission which has been started can be interrupted by pressing DEL □ on the TNC and STOP on the ME-unit. After interruption of transmission, the following error message is displayed:
= ME: PROGRAM INCOMPLETE =
After cancellation of the message via CE, the menu of data transmission modes is displayed.

External data transmission
External data store → TNC

Program directory

Operating mode ______ ◈

Transmission (keys on ME-unit) ______ ←8 TNC

Dialogue initiation ______ EXT ↑

PROGRAM DIRECTORY ▶ ENT Enter mode into memory

▼

EXTERNAL DATA INPUT

Magnetic tape is started

▼

END = NOENT

10 15 600

All programs which are stored on the magnetic tape are displayed, but not transmitted.

Leave mode if desired: ▶ NO ENT Leave mode

▼

PROGRAMMING AND EDITING

The control is in the PROGRAMMING AND EDITING mode.

External data transmission
External data store → TNC

Read-in all programs:

Operating mode ______ ⊕

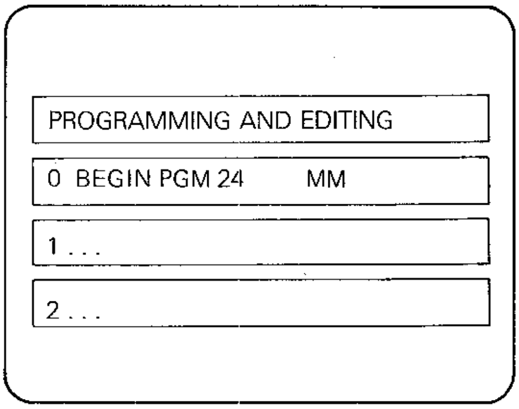
Transmission (keys on ME-unit) ______ ←β TNC

Dialogue initiation ______ EXT ↑

READ-IN ALL PROGRAMS ▶ ENT Enter mode into memory

EXTERNAL DATA INPUT

Magnetic tape is started



```
PROGRAMMING AND EDITING
0  BEGIN PGM 24        MM
1 . . .
2 . . .
```

All programs which are stored on the tape are within the TNC-memory. The program with the highest program number is displayed.

External data transmission
External data store → TNC

Read-in program offered

Operating mode ______________________ ⇨

Transmission (keys on ME-unit)______________ ←ϐ TNC

Dialogue initiation ____________________ EXT ↑

READ-IN PROGRAM OFFERED ▶ ENT Enter mode into memory

▼

EXTERNAL DATA INPUT

Magnetic tape is started

▼

ENTRY = ENT/OVERREAD = NOENT

22

If offered program is to be transferred. ▶ ENT Enter program into memory.

If offered program should **not be** transferred ▶ NO ENT Jump to next program

▼

ENTRY = ENT/OVERREAD = NOENT

24

The control displays all programs which are stored on the tape, one after the other. After display of the program with the highest number, the control jumps automatically back to the PROGRAMMING AND EDITING mode.

External data transmission
External data store → TNC

Read-in selected program

Operating mode ____________________ ⇨

Transmission (keys on ME-unit) ____________ ←8 TNC

Dialogue initiation ____________________ EXT

READ-IN SELECTED PROGRAM ▶ ENT Enter mode into memory.

▼

PROGRAM NUMBER = ▶ □ Key-in reqd. program number.

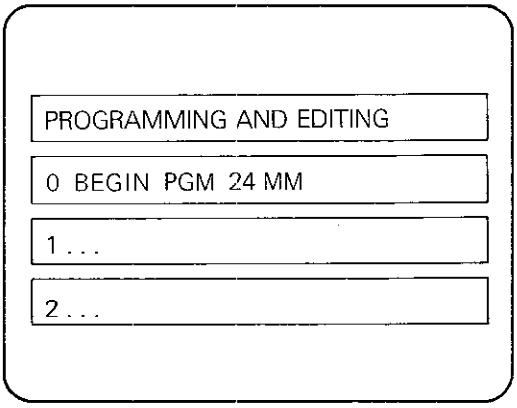
▼ ENT Enter into memory.

▼

EXTERNAL DATA INPUT

Magnetic tape is started.

▼



```
PROGRAMMING AND EDITING
0 BEGIN PGM 24 MM
1 . . .
2 . . .
```

The program offered is now in the TNC-memory and being displayed.

External data transmission
TNC → External data store

Read-out selected program

Operating mode ______ ⇔

Transmission (keys on ME-unit) ______ TNC →8

Dialogue initiation ______ EXT ↓

READ-OUT SELECTED PROGRAM ▶ ENT — Enter mode into memory.

▼

EXTERNAL DATA OUTPUT

Magnetic tape is started and stops after output of screen message.

▼

OUTPUT = ENT/END = NOENT ▶ ↓ ↑ ← → — Set cursor to reqd. program number.

1 **12** 13

14 15 24

▼ ENT — Transfer the selected program to the tape.

▼

EXTERNAL DATA OUTPUT

Magnetic tape is started and stops after transfer of program.

▼

OUTPUT = ENT/END = NOENT

1 12 **13**

14 15 24

The cursor is set to the next program number.

If the mode is to be cancelled. ▶ NO ENT — Cancel mode.

▼

PROGRAMMING AND EDITING

The control is now in the PROGRAMMING AND EDITING mode.

External data transmission
TNC → External data store

Read-out all programs

Operating mode ______________________ ⟐

Transmission (keys on ME-unit) ___________ TNC →8

Dialogue initiation ____________________ EXT ↓

READ-OUT ALL PROGRAMS ▶ ENT Enter mode into memory.

▼

EXTERNAL DATA OUTPUT

Magnetic tape is started and transmission begins.

After data transmissior, the control is in the PROGRAMMING AND EDITING mode.

Remarks

External data transmission
Transfer blockwise

Execution from an external store

In the "transfer blockwise mode", machining programs can be transferred and executed from an external store via the series data interface V.24-(RS-232-C). It is therefore possible to execute programs which exceed the storage capacity of the control.

Data interface

The data interface is programmable via machine parameters. A detailed description of the interface signals and necessary software adaptation of the computer is given in the manual "Interface description TNC 151/TNC 155".

Starting of "Transfer blockwise"

Data transmission from an external store can be started with the EXT-key in the modes: "Single block/Automatic program run" and "Test run". The control stores the program blocks in the memory available and interrupts data transmission if the memory capacity is exceeded.

The display shows no program blocks until either the available memory is full or the complete program has been transferred.

Although program blocks are not being displayed, program run can be started by pressing the external START-button.

When operating via an external store, only short positionings are normally executed. In order to prevent an unnecessary interruption after starting, a substantial buffer of program blocks should be stored. It is therefore advantageous to wait until the available memory is full.

After starting, the executed blocks are automatically erased and further blocks are called-up from the external store.

External data transmission
Transfer blockwise

Overreading program blocks

If GOTO □ is pressed and a block number entered prior to the starting of "transfer blockwise", all blocks prior to the entered block number are overread.

Interruption of program execution

Interruption of execution is possible:

- by pressing the external stop button and internal STOP-key.

The display TRANSFER BLOCKWISE remains after interruption of execution. It is erased if

- a new program number is called-up
  or
- a mode changeover is made from single block/Automatic program run to another operating mode.

Program structure

In the "transfer blockwise" mode the following applies for program structure:

- Program calls, Subprogram calls, Program part repeats and certain program jumps cannot be executed.
- Only the last defined tool can be called-up. (exception: Operation with a central tool store).

Block number

The program which is being transferred may contain blocks with numbers greater than 999. The block numbers do not have to be consecutive, but should not exceed the number 65534. With plain language programs, 4-digit block numbers are displayed in 2 lines on the screen.

External data transmission
Transfer blockwise

Starting of "Transfer blockwise"

Operating mode ______ ⇛ ⇒ ⇢

Dialogue initiation ______ EXT

PROGRAM NUMBER

▶ □ Key-in reqd. program number

▼ ENT Enter into memory

▼

TRANSFER BLOCKWISE

Wait until the screen displays the first blocks.

▶ START Execute program

Interruption of "Transfer blockwise"

TRANSFER BLOCKWISE

Program run is to be interrupted

▶ STOP Interrupt program run

▼ STOP Terminate program run

In the ⇒ -mode, the program which has been started can be interrupted by switching over to the ⇛ -mode.

External data transmission
Output of TNC 155 graphics in hardcopy

This is possible with the TNC 155 only (as of software version 03)

A machining program of the TNC 155 can be scrutenised with the aid of the graphics feature. The graphics image on the VDU-screen can be output via the V.24 (RS-232-C) interface and printed in hardcopy.

The external printer can be adapted to the TNC 155 via machine parameters 226 to 233. The printing procedure is started by pressing the EXT -key whilst the required graphics image is being displayed.

The following entry values are applicable to the **Texas Instruments-Printer OMNI 800/Model 850** for machine parameters 226 to 233:

Parameter No.	Entry value
226	1819
227	17200
228	6977
229	2060
230	1290
231	6990
232	2
233	0

Machine parameter 222 must be set to 169
For controls with software version 08:
With output of graphics, the control automatically switches to 8 data bits. Therefore, for parameter 222 the normal value 168 can be entered for ME 101 and ME 102.

Following entry values apply to the **EPSON Matrix printer:**

Parameter No.	Entry value
226	1819
227	17217
228	6963
229	5642
230	1290
231	6987
232	2
233	0

Remarks

Technical description/Specifications

Control versions

TNC 151
with visual display unit BE 111 (9-inch monochrome) or BE 211 (12-inch monochrome) including PLC for external machine adaptation
TNC 151 A
without separate PLC-I/O-boards
TNC 151 P
inputs and outputs on 1 or 2 separate PLC-I/O-boards
TNC 155
with visual display unit BE 411 (12-inch monochrome) including PLC for machine adaptation
TNC 155 A
without separate PLC-I/O-boards
TNC 155 P
inputs and outputs on 1 or 2 separate PLC-I/O-boards

Control type

Contouring control for 4 axes
Linear interpolation in 3 out of 4 axes, Circular interpolation in 2 out of 4 axes, Helical interpolation
Program entry and display either with HEIDENHAIN-plain language dialogue or to ISO 6983 standard format (G-codes), mm/inch instant conversion for entry values and displays
Display step 0.005 mm or 0.0002 inch or optionally 0.001 mm or 0.0001 inch
Nominal positions (absolute or incremental) in Cartesian or Polar co-ordinates
Entry step down to 0.001 mm or 0.0001 inch or 0.001°

Operator-prompting and displays

Plain language dialogue and fault/error indication (in various languages),
Display of current program block, previous block and 2 successive blocks
Actual position/Nominal position/Target distance/Trailing error display and status display for all important program data

Program memory

Buffered semiconductor store for 32 NC-programs; Programmable erase/edit protection;
TNC 151
Optional 1200 or 3100 blocks
TNC 155
3100 blocks

Central tool store

Up to 99 tools for automatic random select toolchangers with variable tool location coding

Operating modes

Manual/Electronic handwheel: Control operates as a digital readout
Positioning with MDI: Positioning block is keyed-in (without entry into memory) and immediately positioned
Program run in single block: Block-by-block positioning with individual press of button
Automatic mode: After press of button, complete run of program sequence until "programmed STOP" or program end.
Programming (also during program run)
a) with linear or circular interpolation:
Manually (MDI) to program list or workpiece drawing
or externally via V.24/RS-232-C data interface (e.g. Magnetic tape unit ME 101/102 from HEIDENHAIN or other peripheral unit)
b) with single axis operation: additionally by entering actual position data (playback) during conventional manual machining.
Transfer blockwise: On line operation with a host computer. Programs which exceed the memory capacity of the control can be transferred from the host computer in data blocks and simultaneously executed.
Additional operating modes: mm/inch, character height for position display, Safety zones, User-parameters (defined by machine tool builder)
Displays for: Vacant blocks, Actual/Nominal position/Target distance/Trailing error, Baud rate, Block number increment (with ISO-programming)

Technical description/Specifications

Programmable functions	Linear chamfer Circular path by circle centre and end point of circular arc/Circular path with tangential run-on by end point of circular arc/Circular path with tangential transition on both ends by radius only. Tangential contour approach and departure Tool number, tool length and radius compensation Spindle speed Rapid traverse Feed rate Call-up of programs into other programs (4 x nesting) Subprograms/Program part repeats (8 x nesting) Canned cycles for: Pecking, Tapping, Slot milling, Rectangular pocket milling, Circular pocket Co-ordinate transformations: Datum shift, Co-ordinate system rotation, Mirror image, Scaling Dwell time Auxiliary functions M Program Stop
Parameter programming	Mathematical functions (=, +, –, x, ÷, sine, cosine, $\sqrt{\ }$, $\sqrt{a^2 + b^2}$) Parameter comparison (=, +, >, <)
Program test without machine movement	**TNC 151/TNC 155:** Analytical program test without graphics **TNC 155 only:** Graphics simulation of machining program Display modes: in three planes, view with depth shading, 3D-view
Program editing	Editing of block-words, insertion of program blocks, deletion of program blocks; Search routines or finding blocks with common criteria within a program.
Program run continuation after interruption	The control simplifies continuation of program run by storing all important program data.
Touch probe functions	For setting-up operation in the "manual" or "electronic handwheel" mode. Detection of workpiece attitude on the machine table through point probing. Definition of a corner position or centrepoint and workpiece rotation. Programmable: Setting of a workpiece surface as datum.
Data interface	Standard series interface to CCITT-recommendation V.24/EIA-standard RS-232-C Programmable Baud rates: 110, 150, 300, 600, 1200, 2400, 4800, 9600 Baud Extended interface with control character and block check character BCC for "transfer blockwise"-mode and "execution of machining programs".
Monitoring system	The control monitors the functioning of important electronic subassemblies including positioning systems, position transducers and important machine functions. If a fault is discovered via this monitoring system, it is indicated in plain language on the visual display unit (VDU) and the machine emergency stop is activated.
Reference mark evaluation	After a power failure, automatic re-generation of datum setting by traversing over transducer reference mark.
Max. traversing distance	± 30 m or 1181 inches
Max. traversing speed	16 m/min. or 630 inches/min.
Feed rate and spindle override	Two potentiometers on the control panel

Technical description/Specifications

Position transducers	HEIDENHAIN incremental linear transducers or rotary encoders Signal cycle 0.02 mm or 0.01 mm or 0.1 mm (with R-Version via EXE)
Limit switches	Software-controlled limit switches for axis movements (X+/X–/Y+/Y–/Z+/Z– and IV+/IV–). Each traversing range is entered as a machine parameter. Additional programmable safety zones.
Integral PLC for machine adaptation	1000 user-markers (without power failure protection) 1000 user-markers (with power failure protection) 1024 fixed allocated markers 16 counters, 32 timers **Inputs/outputs for TNC 151 A/TNC 155 A:** 23 inputs (24 V =, ca. 10 mA) 24 outputs (24 V =, max. 50 mA) **PLC board for TNC 151 P/TNC 155 P:** 63 (+63) inputs (24 V =, ca. 10 mA) **PL100:** 31 (+31) outputs (24 V =, max. 1.2 A) **PL110:** 25 (+25) outputs (24 V =, max. 1.2 A) + 3 (+3) bipolar output pairs (15 V =, 300 mA) External power supply for PLC: 24 V = + 10%/– 15% Option: specific macro-commands for toolchanger (fixed or variable tool location coding)
Control inputs TNC 151/TNC 155 (with standard-PLC-program)	Transducers X, Y, Z, IV Electronic handwheel (HR 150 or HR 250) or 2 electronic handwheels (HE 310) Start, Stop, Rapid traverse Feedback signal: "Auxiliary function completed" Feed rate release Manual activation (opens positioning loop) Feedback signal; emergency stop-supervision Reference end position X, Y, Z, IV Reference pulse inhibit X, Y, Z, IV Machine traverse buttons X, Y, Z, IV External feed rate potentiometer
Control outputs TNC 151/TNC 155 (with standard-PLC-program)	1 analogue output each for X, Y, Z, IV (with automatic offset-adjustment) One analogue output for S Axis release X, Y, Z, IV "Control in operation" M-strobe signal S-strobe signal T-strobe signal 8 outputs for M, S- and T-functions coded "Coolant off"; "Coolant on" "Spindle counter-clockwise" "Spindle stop" "Spindle clockwise" Spindle lock on Control in "automatic" operating mode Emergency stop
Mains power supply	Selectable 100/120/140/200/220/240 V + 10%/– 15%, 48...62 Hz
Power consumption	**TNC 151** ca. 60 W (with 9 or 12-inch VDU) **TNC 155** Logic and control unit ca. 45 W, VDU ca. 40 W
Ambient temperature	Operation 0...45° C (32...113° F), Storage –30...70° C (–22...158° F)

Technical description/Specifications

Weight

Control TNC 151/TNC 155: 12 kg (26 lb.)
Visual display unit BE 111 (9 inch): 6,8 kg (15 lb.)
Visual display unit BE 211/BE 411 (12 inch): 10 kg (24 lb.),
PLC-board PL 100/PL 110: 1.2 kg (2.6 lb.) (TNC 151 P/TNC 155 P)

Technical description/Specifications

With infra-red transmission
TS 510

Triggering 3D-touch probe
Probing reproducibility better than 1 µm
Probing speed max. 3 m/min.
Stylus with deliberate fracturing point
Ball tip material: ruby
Shank and stylus versions to customer specifications

Infra-red transmission
2 signal transmitters (at 0° and 180°)
1 starting signal receiver (at 0°)
Possible signal beam direction to spindle axis (please specify when ordering): 90/60/30°
Distance: 3D-touch probe – transmitter/receiver unit 500...2000 mm

Operating voltage:
4 micro-sized Ni-Cd-batteries
Max. operating duration per charge:
Measuring operation 8 hours; standby operation 1 month
Standard supply: Second battery set and external charging unit (220 V, 50 Hz)

Protection: IP 55 – DIN 40050/IEC 529

Interface to NC control
The interface comprises a transmitter and receiver unit including matching electronics

SE 510
Transmitter and receiver unit:
Diameter 80 mm; Length 49 mm
Cable length 3 m
Protection: IP 66 – DIN 40050/IEC 529

APE 510
Matching electronics:
Within aluminium diecas: housing: LxWxH 175x80x57 mm
Max. cable length 20 m
Protection: IP 64 – DIN 40050/IEC 529

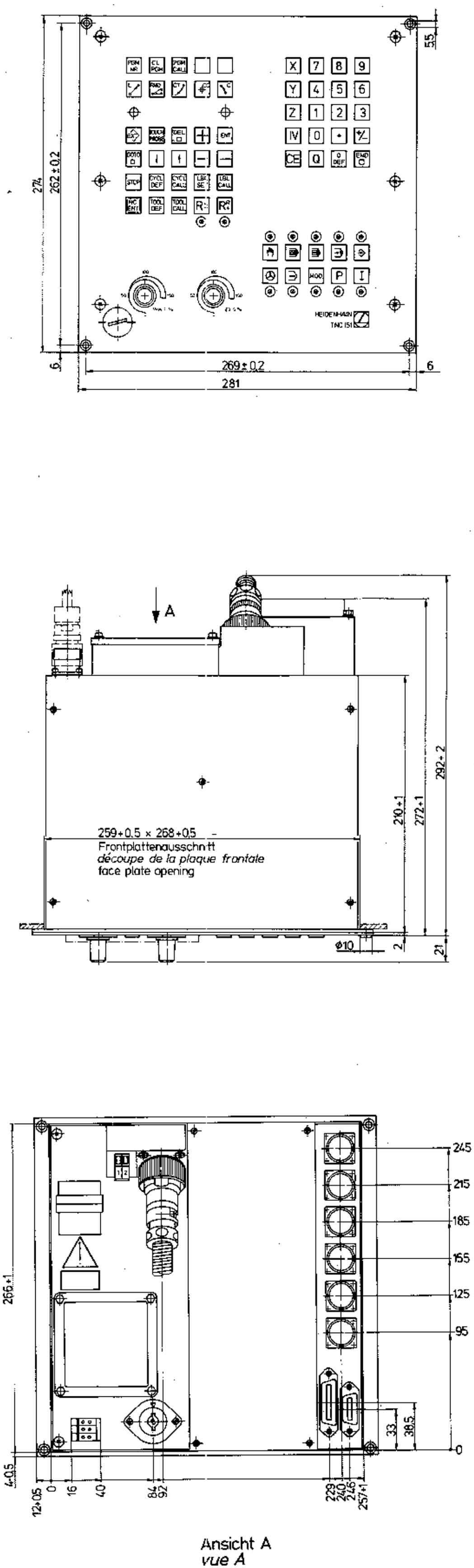
With cable
TS 110

Triggering 3D-touch probe
Technical specifications as per 3D-touch probe for infra-red transmission however, without infra-red transmitter/receiver
Max. cable length 3 m

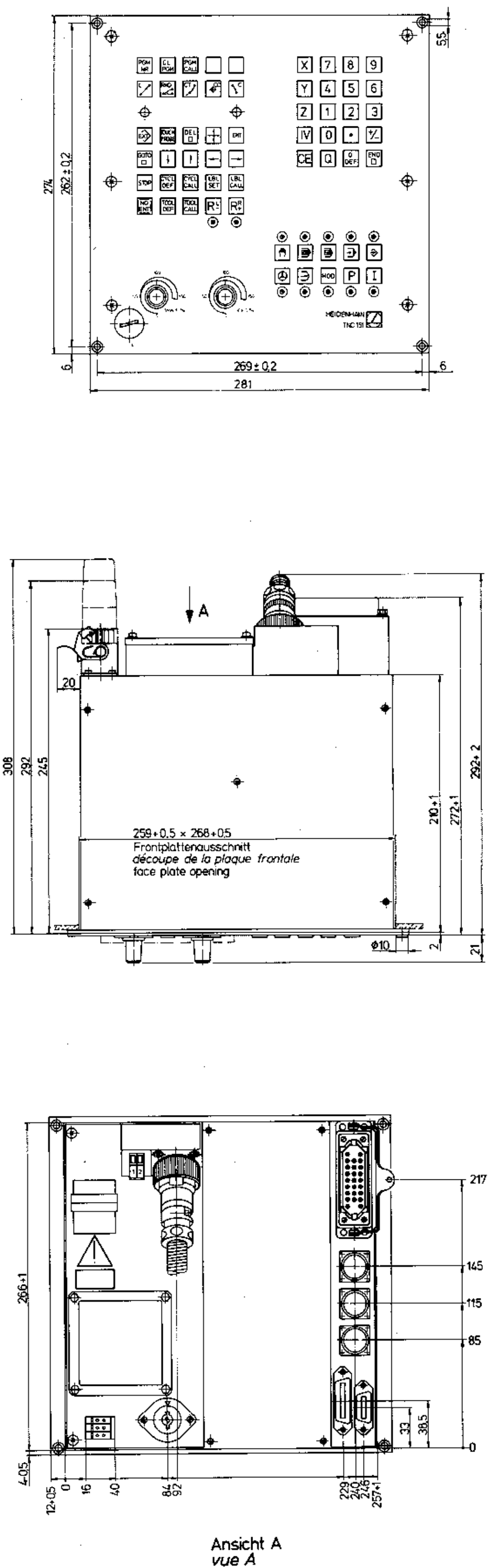
APE 110
Matching electronics
Within aluminium diecast housing: LxWxH 175x80x57 mm
Max. cable length 20 m
Protection: IP 64 – DIN 40050/IEC 529

Dimensions
Logic/Operating unit
TNC 151 A/P TNC 151 AR/PR
TNC 151 E/V TNC 151 ER/VR

Dimensions in mm



Ansicht A
vue A
view A



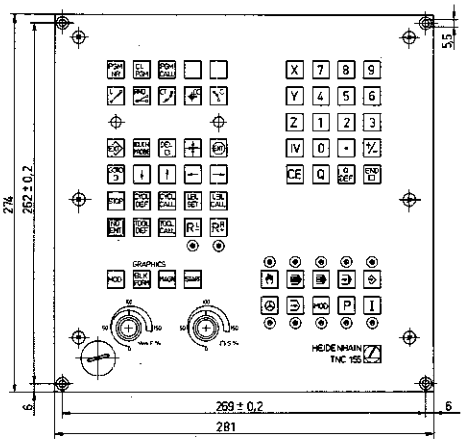
Ansicht A
vue A
view A

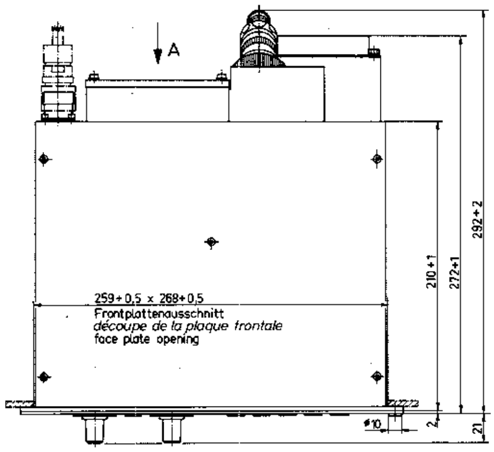
Dimensions

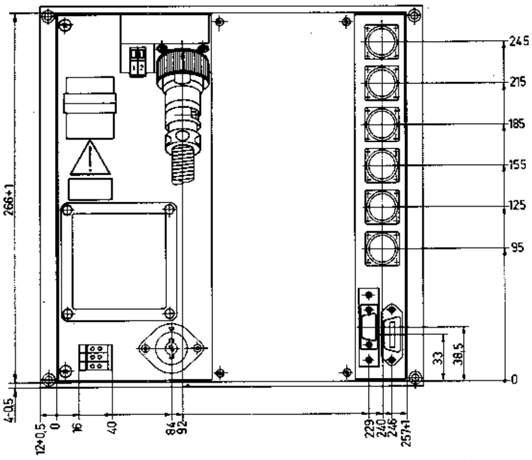
Logic/Operating unit

TNC 155 A/P
TNC 155 E/V

Dimensions in mm

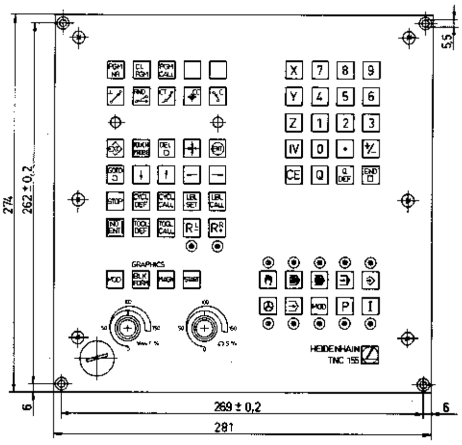


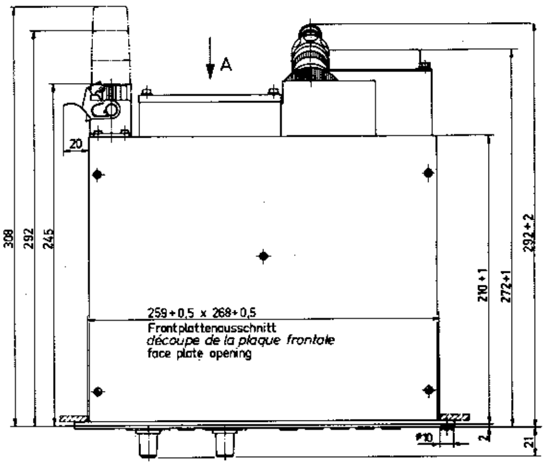


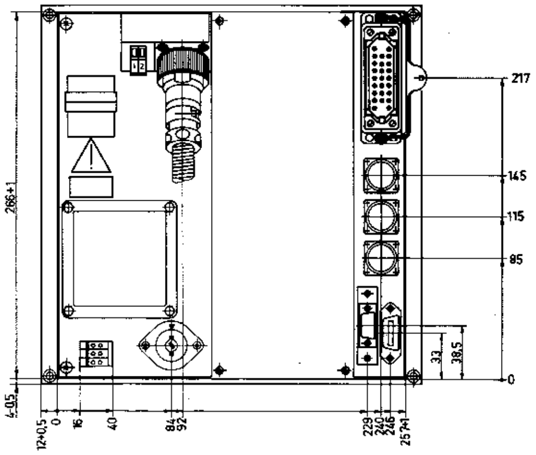


Ansicht A
vue A
view A

TNC 155 AR/PR
TNC 155 ER/VR



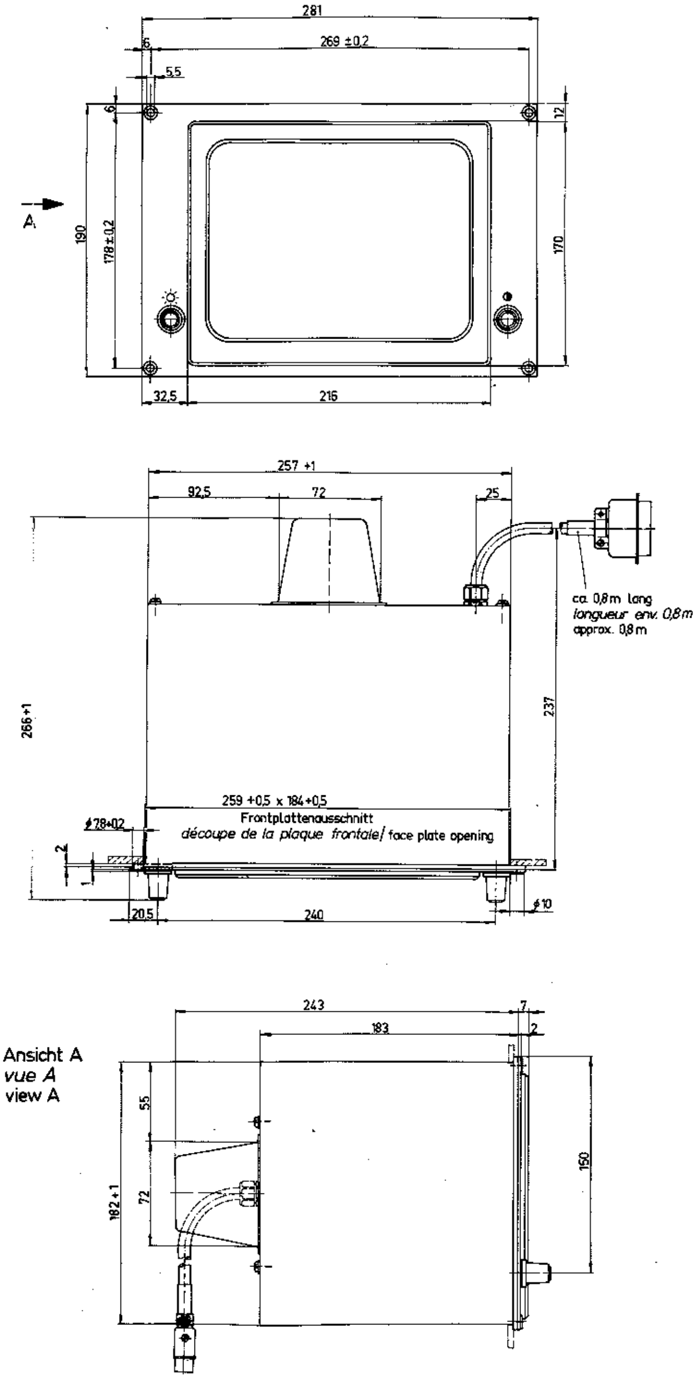




Ansicht A
vue A
view A

Dimensions
Visual display unit BE 111 (9 inches)

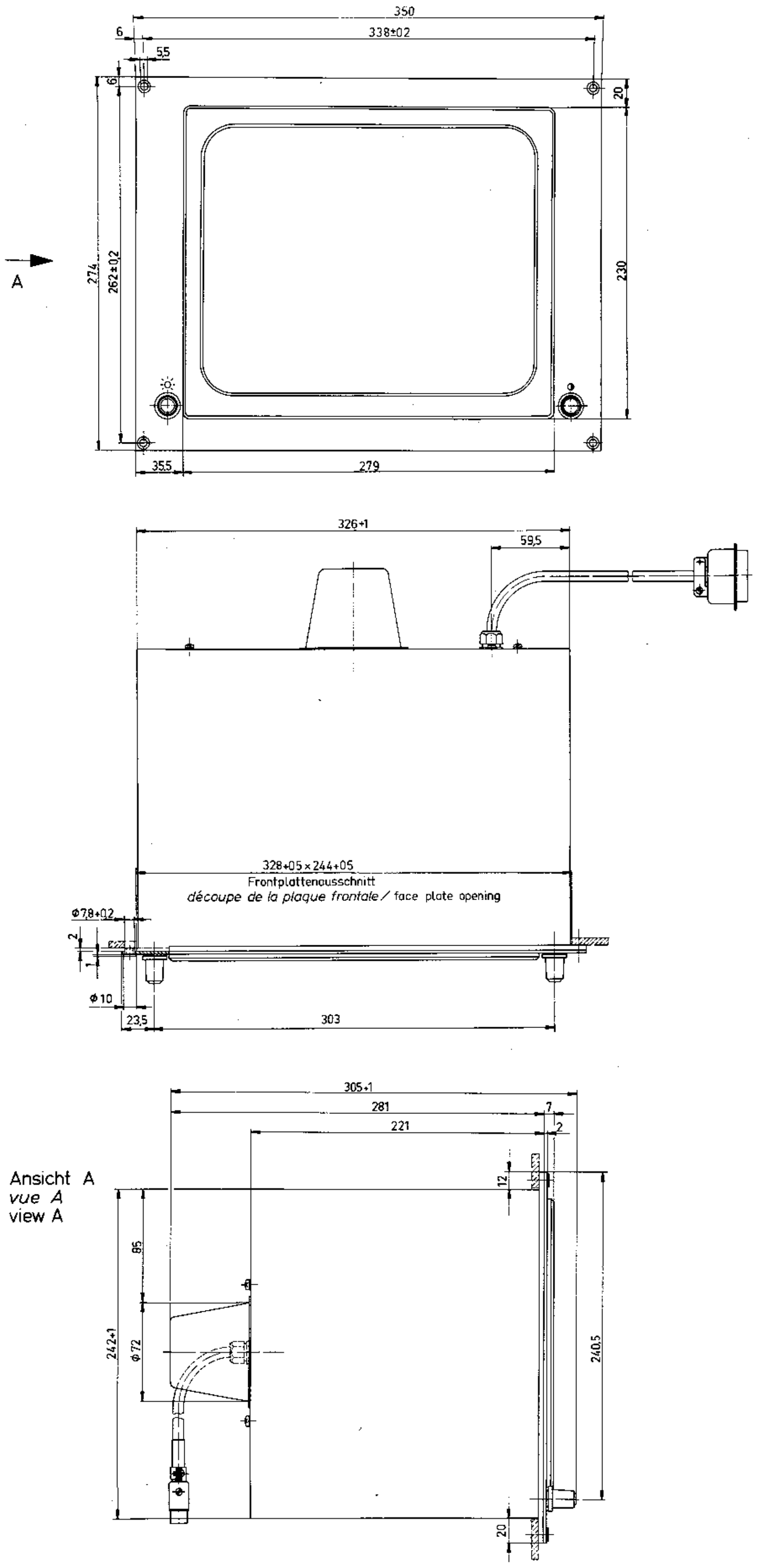
Dimensions in mm



Dimensions

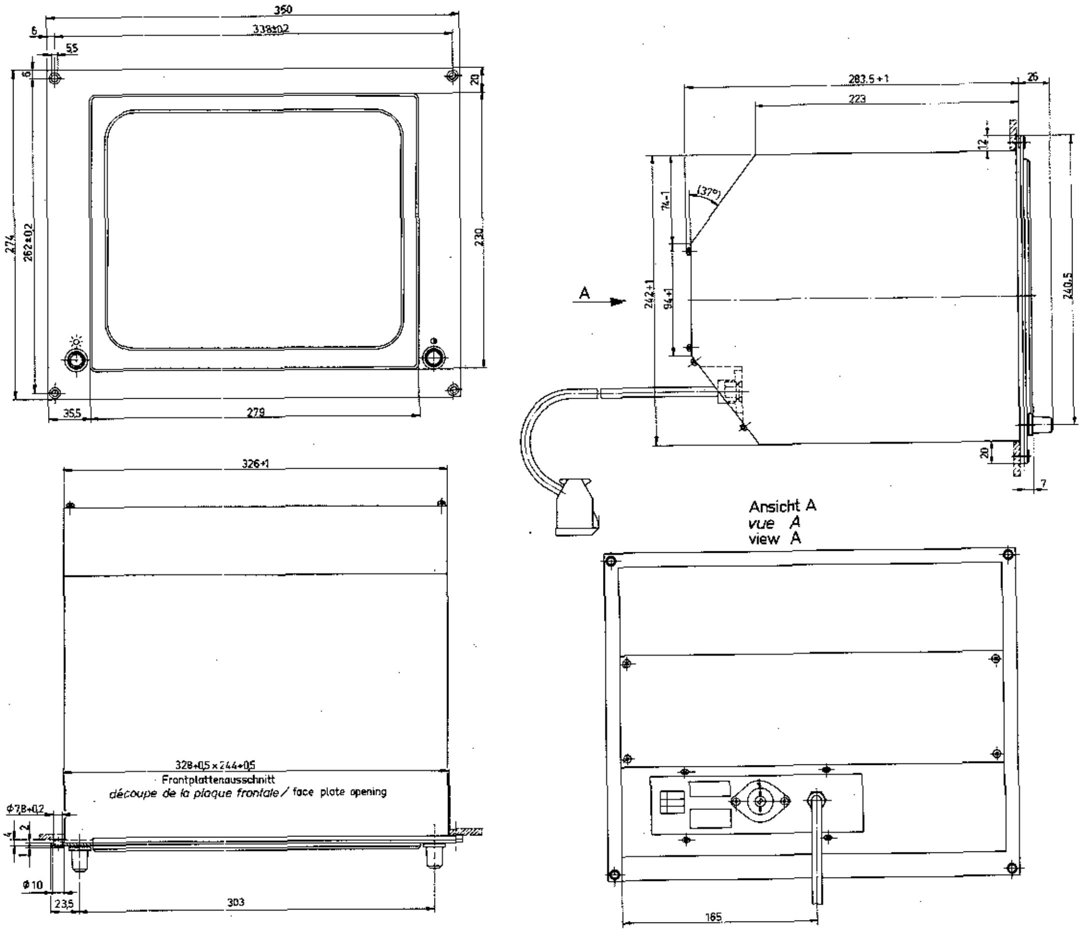
Visual display unit BE 211 (12 inches)

Dimensions in mm



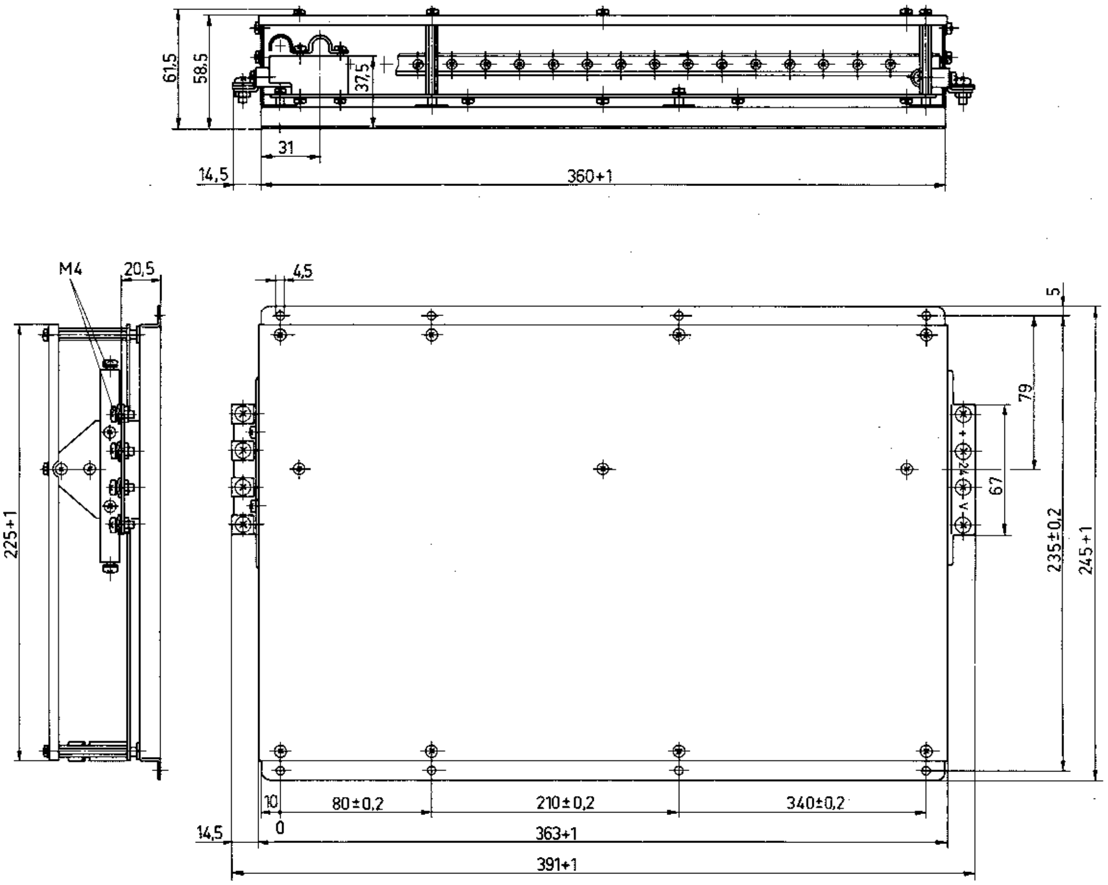
Dimensions
Visual display unit BE 411

Dimensions in mm



Dimensions
PLC-Board PL 100/PL 110

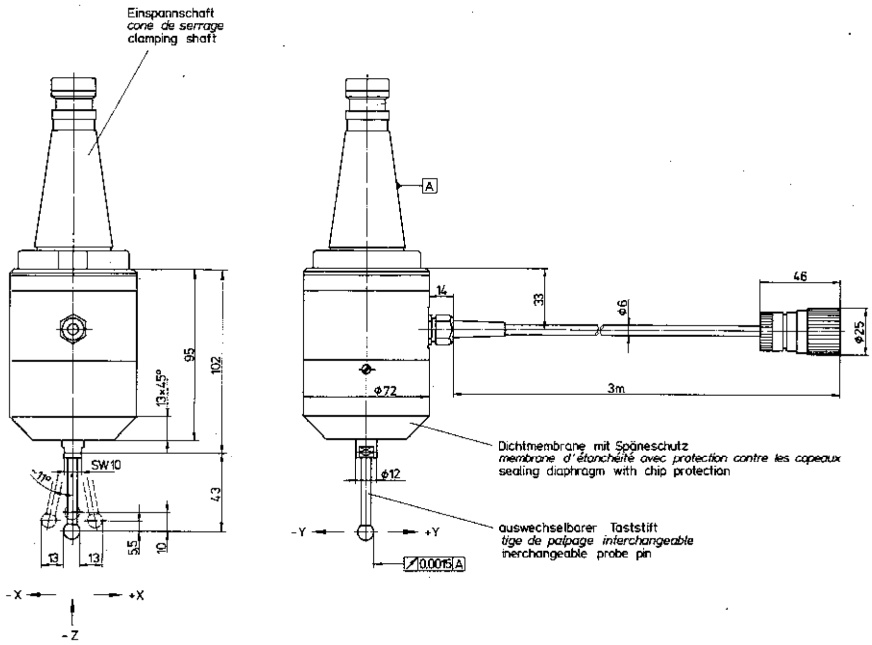
Dimensions in mm



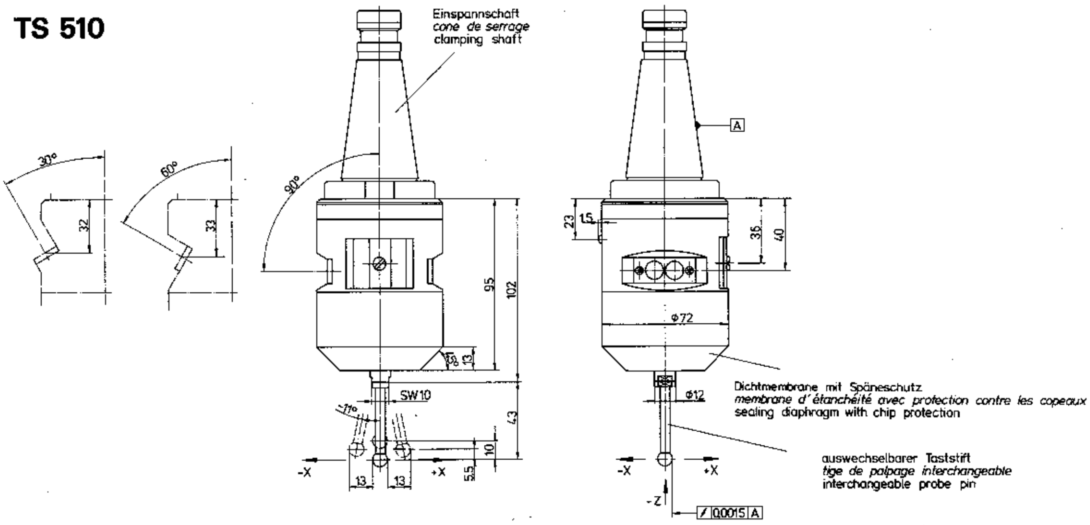
Dimensions
Touch probe system

Dimensions in mm

TS 110


Einspannschaft
cone de serrage
clamping shaft
A
46
14
33
φ6
φ25
95
102
13x45°
φ72
3m
SW10
11°
43
5,5
10
13
13
φ12
Dichtmembrane mit Späneschutz
membrane d'étanchéité avec protection contre les copeaux
sealing diaphragm with chip protection
auswechselbarer Taststift
tige de palpage interchangeable
inerchangeable probe pin
0,0015 A
-X
+X
-Z
-Y
+Y

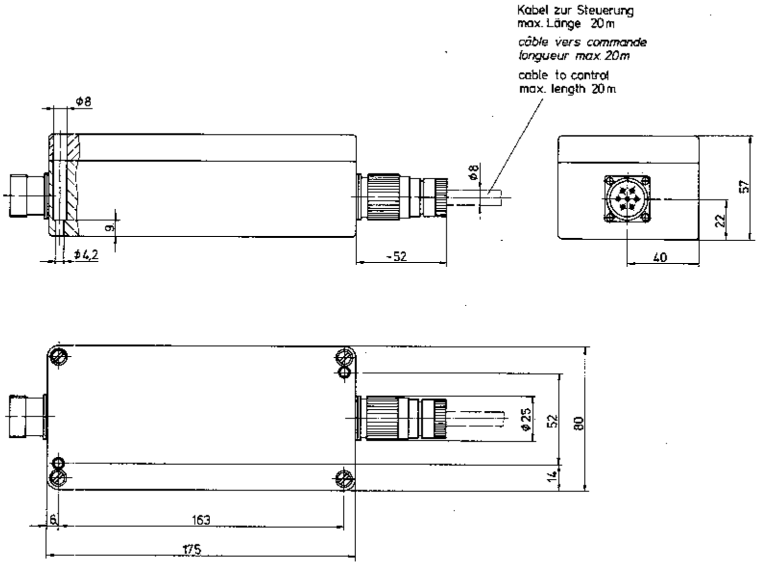
TS 510


Einspannschaft
cone de serrage
clamping shaft
30°
32
60°
33
90°
A
23
1,5
35
40
95
102
3x1
13
φ72
SW10
11°
43
10
5,5
13
13
-X
+X
φ12
Dichtmembrane mit Späneschutz
membrane d'étanchéité avec protection contre les copeaux
sealing diaphragm with chip protection
auswechselbarer Taststift
tige de palpage interchangeable
interchangeable probe pin
-X
+X
-Z
0,0015 A

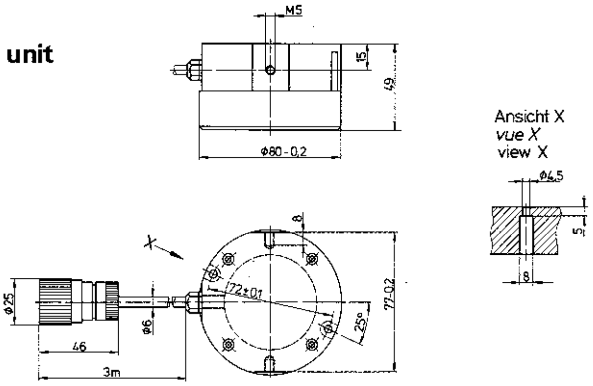
Dimensions
Touch probe system

Dimensions in mm

APE 110
APE 510



Transmitter/Receiver unit



Index

A

Absolute dimensions K10, P17, P22, D10
● ISO D10
● Plain language P17
Adjoining arcs P46
● ISO D16
● Plain language D47
Advanced stop distance t P86
Angle reference axis K2
Approach command M95 P61
Approach command M96 P60
Approach command M98 P60
Arc with tangential connection (see Adjoining arcs) P46
Auxiliary functions M P30
● freely selectable P33
● which affect program run P32

B

Basic rotation A11
– entry A12
Baud rate E12
– entry V2
Blank (Graphics) P130
BLK FORM (Blank form) P130, P133
Block call-up P122
Block deletion P124
Block insertion P124
Block number P2
Block number increment E12, D5
Buffer battery V16

C

C (see Circular interpolation) P40
Calibration A3
– effective length A3
– entry A4
– effective radius A7
– entry A8
Canned cycles P82
CC (see Circle centre/Pole) P20, P40
CE-key P4
Chain dimensions K10, P17, P22, D10
Chamfers P50
● ISO D18
● Plain language P51
Circle centre P20, P40
● ISO D15
● Plain language P21
Circle centre = Datum A23
– entry A24
Circular interpolation P40
● ISO D14, D15, D16
● Plain language P43, P45
Circular path P40
● ISO D14, D15, D16
● Plain language P43, P45
Circular pocket P104
● ISO D24
● Plain language V4
Code number E16
Connecting cable (ME) V4

Index

C continued

Contour approach in a straight path P56
● Path angle ″ equal to 180° P57
● Path angle ″ greater than 180° P59
● Path angle ″ less than 180° P58
Contour approach on an arc P54
● ISO D19
● Plain language P55
Contouring keys P18
Conversion mm/inch E10
Cosine (Parameter definition) P74
CT (see Adjoining arcs) P46
Co-ordinates K1, P17
● Cartesian K1, P18
● Polar (see Polar co-ordinates) K2, P22
● Programming P19, P23
● Right-angled K1, P18
Co-ordinate axes K1
Co-ordinate system K1
Co-ordinate system rotation P114
● ISO D26
● Plain language P115
Co-ordinate transformations P82
Corner = Datum A17
– entry A18
Cycle P82
– call P82
– cancellation P85
– definition P82
– deletion P124
– parameters D21

D

D (Address letter) D30
Data transmission V1
Datum, setting K9
Datum shift P110
● ISO D25
● Plain language P111
Deletion of block P124
Departing from a contour in a straight path P56
● Path angle equal to 180° P57
● Path angle greater than 180° P59
● Path angle less than 180° P58
Departing from a contour on an arc P54
● ISO D19
● Plain language P55
Departure command M98 P60
Dialogue prompting P2
Direction of rotation P40
● Angle K2, P114
● Circular interpolation P40
● Circular pocket milling P104
● Pocket milling (rectangular) P98
DR (see Direction of rotation) P40
Dwell time P118
● ISO D26
● Plain language P119
● Within a machining cycle P86

Index

E

Edit (see Erase/Edit protection) P6
● ISO D8
● Plain language P142
Editing of block words P122
Electronic handwheel M2
Ellipse (programming example) P80
Emergency stop P150
END-key P3
Enlargement (Scaling) P117
- Graphics P142
ENT-key P3
Erase (see Erase/Edit protection) P6
● ISO D8
● Plain language P142
Erase/Edit protection P6

F

F (Address) D25, D26
F (see Feed rate) P30, D12
Fast image data processing (Graphics) P133
Feed rate P30, D12
● within a canned cycle P86
● override M1, P146, P162
- constant on external corners M90 P27
FN (see Parameter-function) P70
Freely programmable cycles (Program call) P120
● ISO D26
● Plain language P121

G

G (Address) D6
G-functions D6
GOTO (see **B**lock call-up and **J**ump) P122
Graphics P130
● Start P134, P135
● Stop P134, P137

H

H (Address) D13
Helical interpolation P52
● ISO D16
● Plain language P53

I

I (Address) D13
If equal, jump P76
If greater than, jump P78
If – jump P76
If less than, jump P78
If unequal, jump P78
Increase (Scaling) P117
Incremental dimensions K10, P17, P22, D10
● ISO D10
● Plain language P17
Infinite loop P64
Inserting a block P124
Interpolation factor (Electronic handwheel) M2
Interruption of power E4

Index

J

J (Address) ______ D13
Jump (conditional) ______ P76
● ISO ______ D31
● Plain language ______ P77
Jump (unconditional) ______ P62
● ISO ______ D28
● Plain language ______ P63

K

K (see Address) ______ D13
k (see **S**tepover) ______ P99
Keyboard ______ D1
● for ISO (Fold-out page) ______ D1
● for plain language (Fold-out page) ______ D1

L

L (see **L**inear interpolation) ______ P34
Label ______ P62
● call ______ P62
● number ______ P62
● set ______ P62
LBL ______ P63
LBL CALL ______ P63
LBL SET ______ P63
Limits (safety zones) ______ E14
Linear interpolation ______ P63
● Three-dimensional (3D) ______ P34
● Two-dimensional (2D) ______ P34

M

M (Address) ______ P30
Machine axes ______ K3
Machining cycles ______ P82, P84
● ISO ______ D19
● Plain language ______ P83
Machine parameters ______ V16
● table of ______ P168
Magnetic tape unit ______ V3
MAGN (see Magnify) ______ P142
Magnify (Graphics) ______ P142
Manual operation ______ M1
ME (see **M**agnetic tape unit) ______ V3
M-function ______ P30
Milling depth ______ P92, P98, P104
Mirror image ______ P112
● ISO ______ D25
● Plain language ______ P113
Miscellaneous function ______ P30
mm/inch changeover ______ E10
MOD-function ______ E8
MP (see **M**achine parameters) ______ V16

N

N (Address) ______ D5
NC: Software number ______ E16
Nesting ______ P66
NO ENT-key ______ P3

Index

O

Override M1, P146, P162
– feed ride M1, P146, P162
– spindle speed M1, P146, P162

P

P (Address) (see **C**ycle parameter and Parameter definition) D21, D30
P (Display and key) (see **P**olar co-ordinates) P23
Paging P122
– within definitions P83
– within parameter definition P71
– within cycle program P122
Parameters P70
– definition P70
● ISO D30
● Plain language P71
– function P70
– setting P70
● ISO D30
● Plain language P71
Path angle P56
Path/Radius compensation P24
● ISO D17
● Plain language P25
– Correction of path intersection P26
– on external corners P26
– on internal corners P26
– Termination M98 P28, P60
– with single axis positioning blocks P155
● ISO D17
● Plain language P157
Peck-drilling P86
● ISO D21
● Plain language P87
Pecking depth P86, P98
Peripheral unit V1
Plan view (Graphics) P133
Playback P158
PLC: Software number E16
Pocket milling P98
● ISO D23
● Plain language P101
Polar co-ordinates K2, P22
– Angle P22
● ISO D10
● Plain language P23
– Radius P22
● ISO D10
● Plain language P23
Pole P20
● ISO D13, D15
● Plain language P21
Position display E10
Position display enlarged/small E12
Positioning with MDI P162
Power interrupted E4
Program P1
– amendments P122
– call P6
– call (cycle) P120
● ISO D26
● Plain language P121

Index

P continued

– clearing P126
– corrections (see **P**rogram editing) P122
– directory V7
– edit protection P6, P8
– editing P3, P122
– entry P6
  ● ISO D1
  ● Plain language P7
– erase protection P126
– jump P7
  ● if (conditional) P76
  ●● ISO D31
  ●● Plain language P77
  ● into another program P68
  ●● ISO D29
  ●● Plain language P69
  ● unconditional P62
  ●● ISO D28
  ●● Plain language P63
– label P62
  ● ISO D28
  ● Plain language P63
– length P6
– number P6
– part repeat P64
  ● ISO D28
  ● Plain language P64
– protection P7
– run P146
  ● automatic P146, D150
  ● continuation P148, P151, P152
  ● interruption P148
  ● single block P146, P150
  ● termination P148
stop P15
– supervision (see **P**rogram test and **S**earch routines) P128, P126
– test P128
Program entry in ISO-format D1
Program management P6, D5
Program menu P6, V7
Protection (**E**rase/**E**dit) P6
  ● ISO D8
  ● Plain language P142

Q

Q (Address) P70
Q DEF-key P70

R

R (Address) P11, P22, P48, D10, D18
Radius compensation P24
– with normal program blocks P24
– with single axis milling P155
Rectangular Pocket (see **P**ocket milling) P98
Read-in program offered V9
Read-in selected program V10
Read-in tape contents V8
Read-out all programs V12
Read-out selected program V11
Reduction (Scaling) P117
Reference mark K5
– passing over E4

Index

R continued

Reference position ____ K5
Reference signal ____ K5
Relative tool movement ____ K3
REP (see **P**rogram repeat) ____ P64
Repetition ____ P64, P67
RND (see **R**ounding of corners) ____ P49
ROT (see **R**otation angle) ____ P115
Rotation angle ROT ____ P114
Rounding of corners ____ P48
● ISO ____ D18
● Plain language ____ P49
Rounding-off radius ____ P48
Run-off contour ____ P54, P56
Run-on contour ____ P54, P56

S

S (Address) ____ P15, D9
Scaling ____ D25, P116, P117
Scaling factor ____ P116
● ISO ____ D25
● Plain language ____ P117
SCL (see Scaling factor) ____ P117
Screen display (opposite)____ E1
Screen displays (modes ____ E6
Search routines ____ P126
Set-up clearance ____ P86
Sine (Parameter definition) ____ P74
Single axis machining ____ P155
● ISO ____ D12
● Plain language ____ P157
Slot milling ____ P92
● ISO ____ D22
● Plain language ____ P95
Snap-on keyboard for ISO ____ D1
Spindle axis ____ P14
Spindle override ____ M1, P146, P162
Spindle rotation direction (M-function) ____ P32, P84
Square root (Parameter definition) ____ P72
● root of sum of square (Pythagoras) ____ P75
● square root ____ P72
Standard format (see **P**rogram entry in ISO-format) ____ D1
Stepover k ____ P99
STOP ____ P15
Straight paths ____ P34
● ISO ____ D12, D13
● Plain language ____ P37, P39
Subprogram ____ P65
– repetiton ____ P67
Subroutines (see **S**ubprogram) ____ P65
Supplementary operating modes ____ E8
Surface = Datum A14, A26
– ISO ____ D27
– entry ____ D27
– Plain language ____ A14, A26
– entry ____ A15, A27
Switch-on (control) ____ E4
Switchover ISO/Plain language and vice-versa ____ D3

Index

T

T (Address) D9
t (see **A**dvanced stop) P86
Tapping P90
● ISO D21
● Plain language P91
Termination of path compensation M98 P60
Three dimensional (3D)-inpolation (see **L**inear interpolation) P34
Three dimensional (3D)-view P132
Tool P10
– call P14
● ISO D9
● Plain language P15
– compensation P10
● ISO D9
● Plain language P13, P15
● with playback P159
– change P14
– definition P10
● ISO D9
● Plain language P13
– length P10
● ISO D9
● Plain language P13
– number P10, P14
● ISO D9
● Plain language P13, P15
– radius P11
● ISO D9
● Plain language P13
TOOL axis P14
TOOL CALL P14
TOOL CALL 0 P14
TOOL DEF P10
Total hole depth (pecking) P86
Touch probe A1
Touch probe function, general A2
TOUCH PROBE-key A2
Transducer K5
Transfer blockwise (data) V1
Transmission rate for data (see **B**aud rate) V2
Two-D, (2D)-linear interpolation (see **L**inear interpolation) P34

U

Unconditional jump P62
● ISO D28
● Plain language P63
User parameters E16

V

Vacant blocks E8
View in three planes (Graphics) P132

Index

W

Working spindle axis ____ P14
Workpiece ____ P17
– contour ____ P17
– datum (setting) ____ K6, K9
Write release ____ V6

X

Y

Z

Zero tool ____ P10

Error messages

ANGLE REFERENCE MISSING P42, P46

BLK FORM DEFINITION INCORRECT D32
BLOCK FORMAT INCORRECT D1

CIRCLE END POS. INCORRECT P42, P46
CYCL INCOMPLETE P152

EMERGENCY STOP P150
EXCESSIVE SUBPROGRAMMING P64, P66
EXCHANGE BUFFER BATTERY E3, P166
EXCHANGE TOUCH PROBE BATTERY A2

G-CODE GROUP ALREADY ASSIGNED D1, D7

ILLEGAL G-CODE D7

MIRROR IMAGE ON TOOL AXIS P112

PATH OFFSET WRONGLY STARTED P40
PLANE WRONGLY DEFINED P48, P50
PGM-SECTION CANNOT BE SHOWN P134
POWER INTERRUPTED D3, E4, E8
PROBE SYSTEM NOT READY A2
PROGRAM MEMORY EXCEEDED P124
PROGRAM START UNDEFINED P152, D10, D14

RELAY EXT. DC VOLTAGE MISSING D3, E4
ROUNDING RADIUS TOOL LARGE P48

SELECTED BLOCK NOT ADDRESSED P151
SPINDLE ROTATES MISSING P84

TOOL CALL MISSING P84
TOUCH POINT INACCESSIBLE A2

WRONG AXIS PROGRAMMED P113
WRONG RPM P14

Auxiliary functions M

M	Function	Active at block	
		beginning	end
M00	Program run stop Spindle stop Coolant off		●
M02	Program run stop Spindle stop Coolant off Return jump to block 1		●
M03	Spindle on CW	●	
M04	Spindle on CCW	●	
M05	Spindle stop		●
M06	Tool change Program run stop (depending on machine parameters entered) Spindle stop Coolant on		●
M08	Coolant on	●	
M09	Coolant on		●
M13	Spindle on CW Coolant on	●	
M14	Spindle on CCW Coolant on	●	
M30	as per M02		●
M89	Cycle call (modally active)	●	
M90	Constant contouring speed corners (see "contouring speed")	●	
M91	**Within a positioning block:** Workpiece datum is substituted by reference point	●	
M92	**Within a positioning block:** Workpiece datum is substituted by a position which has been defined by the machine tool builder via machine parameter, e.g. tool change position	●	
M93	M-function assigment reserved for HEIDENHAIN		
M94	Reduction of display value of C-axis to value below 360°	●	
M95	Change of approach behaviour (see "Approach behaviour M95")		●
M96	Change of approach behaviour (see "Approach behaviour M96"		●
M97	Path intersection correction on external corners		●
M98	End path compensation		●
M99	Cycle call		●

Letter addresses (ISO)

Letter address	Function	Entry range	
		Numerals	Parameter
%	Program beginning or program call	0 – 99999999	–
A	Rotary axis about X-axis	± 30000.000	Q0 – Q99
B	Rotary axis about Y-axis	± 30000.000	Q0 – Q99
C	Rotary axis about Z-axis	± 30000.000	Q0 – Q99
D	Parameter-Definition (Program-Parameter Q)	0 – 12	–
F	Feed (rate) code	0 – 15999	Q0 – Q99
F	Dwell time with G04	0 – 19999.999	Q0 – Q99
F	Scaling factor with G72	0 – 99.999	–
G	Preparatory function	0 – 99	–
H	Angle for polar co-ordinates	± 5400.000	Q0 – Q99
		± 360.000	Q0 – Q99
H	Rotational angle with G73	± 360.000	Q0 – Q99
I	X-Co-ordinate of circle centre/Pole	± 30000.000	Q0 – Q99
J	Y-Co-ordinate of circle centre/Pole	± 30000.000	Q0 – Q99
K	Z-Co-ordinate of circle centre/Pole	± 30000.000	Q0 – Q99
L	Set label number with G98	0 – 254	–
L	Jump to label number	1 – 254.65535	–
L	Tool length with G99	± 30000.000	Q0 – Q99
M	Auxiliary (Miscellaneous) function	0 – 99	–
N	Block number	1 – 9999	–
		1 – 65534	–
P	Cycle parameter in machining cycles	01 – 07	–
P	Parameter in parameter definition	01 – 03	–
Q	Program parameter	0 – 99	–
R	Radius for polar co-ordinates	± 30000.000	Q0 – Q99
R	Rounding-off radius with G25/G26/G27	0 – 19999.999	Q0 – Q99
R	Chamfer length with G24	0 – 19999.999	Q0 – Q99
R	Tool radius with G99	± 30000.000	Q0 – Q99
S	Spindle speed	0 – 9000.000	–
		0 – 30000.000	–
T	Tool definition with G99	0 – 254	–
T	Tool call	0 – 254	–
U	Additional linear axis parallel to X-axis	± 30000.000	Q0 – Q99
V	Additional linear axis parallel to Y-axis	± 30000.000	Q0 – Q99
W	Additional linear axis parallel to Z-axis	± 30000.000	Q0 – Q99
X	X-Axis command	± 30000.000	Q0 – Q99
Y	Y-Axis command	± 30000.000	Q0 – Q99
Z	Z-Axis command	± 30000.000	Q0 – Q99
*	End of block	–	–

Program entry in ISO-format

G-codes	
G00	Linear interpolation, Cartesian, in rapid
G01	Linear interpolation, Cartesian
G02	Circular interpolation, Cartesian CW
G03	Circular interpolation, Cartesian CCW
G05	Circular interpolation, Cartesian, without direction data
G06	Circular interpolation, Cartesian, tangential contour connection
● G07	Single axis block
G10	Linear interpolation, polar, in rapid
G11	Linear interpolation, polar
G12	Circular interpolation, polar CW
G13	Circular interpolation, polar CCW
G15	Circular interpolation, polar, without direction data
● G04	Dwell
G28	Mirror image
● G39	Designates program, call-up via G79
G54	Datum shift
G72	Scaling
G73	Co-ordinate system (plane) rotation
G74	Slot milling
G75	Rectangular pocket milling CW
G76	Rectangular pocket milling CCW
G77	Circular pocket milling CW
G78	Circular pocket milling CCW
G83	Pecking
G84	Tapping
G17	XY-plane designation, Tool axis Z
G18	ZY-plane designation, Tool axis Y
G19	YZ-plane designation, Tool axis X
G20	Tool axis IV
● G24	Chamfer with R
● G25	Rounding of corners with R
● G26	Tangential contour approach (run-on) with R
● G27	Tangential contour depart (run-off) with R
G40	No tool compensation
G41	Tool radius compensation to contour, offset left
G42	Tool radius compensation to contour, offset right
G43	Tool length compensation positive
G44	Tool length compensation negative
G50	Erase/edit protection
● G79	Cycle call
G90	Absolute dimensioning
G91	Incremental dimensioning
● G29	Transfer of last nominal position value as pole
G30	Blank form definition for graphics – min. point
G31	Blank form definition for graphics – max. point
G70	Dimensioning in inches (at program beginning)
G71	Dimensioning in millimetres (at program beginning)
● G98	Assign label number
● G99	Tool definition
● G51	Next tool number when using the central tool memory
● G55	Touch probe function: Workpiece surface as datum
● G38	Corresponds to a STOP-block in HEIDENHAIN-format

● = G-codes which are only effective blockwise

HEIDENHAIN
DR. JOHANNES HEIDENHAIN GmbH
D-8225 Traunreut · Tel. (0 86 69) 31-0

225 041 24 · 15 · 7/86 · H · Printed in West Germany · Subject to technical modifications