CC 100 M

User Handbook

Version 104

CC 100 M

User Handbook

1070 073 401-104 (89.04) GB

(S32)

Reg. Nr. 16149-03

Discretionary charge 21,30 DM

1 DESCRIPTION

2 OPERATING

4 PARAMETRIC FUNCTIONS

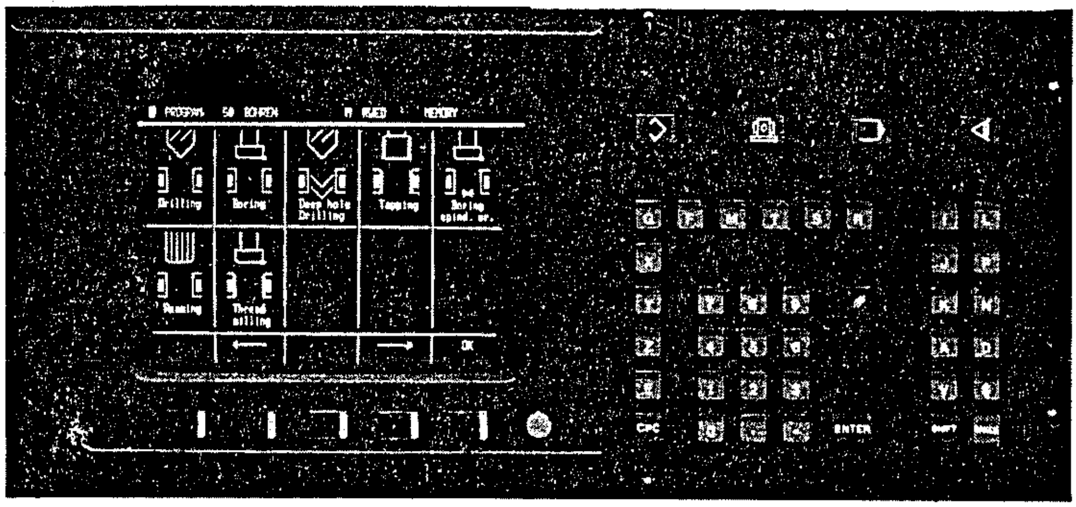
5 TECHNOLOGY

1. DESCRIPTION

CC 100 M

Full CNC continuous path
control for up to 4 numerically controlled
axes plus controlled main spindle.

Programming based on DIN 66025,
extended by graphic and arithmetic functions.



This manual is intended for the use
by the enduser of the control.

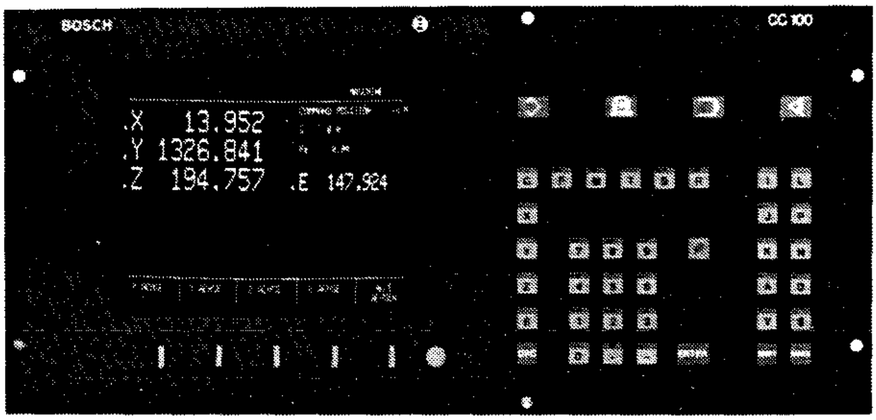
Component parts of the control, operating
elements, maintenance, working with the
data interface are described in chapter 1.

Reset conditions, the reference system, operation
of the operating panel and the manual panel, and
the technology stores are described in chapter 2.

Chapter 3 describes the conventional
programming to DIN, 3-digit G-codes and
contour cycles.

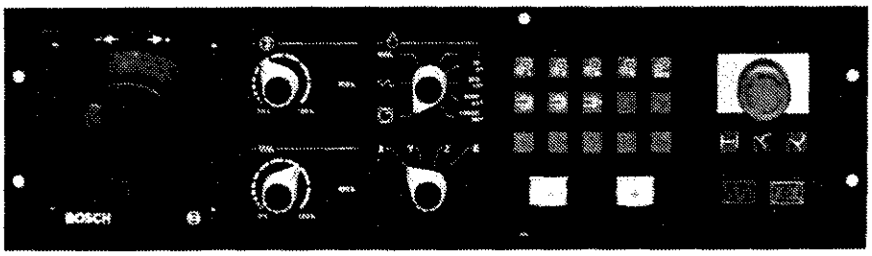
Parametric functions, user graphic, operation
of the tool compensation and special applications
are described in chapters 4 - 7.

COMPONENT PARTS



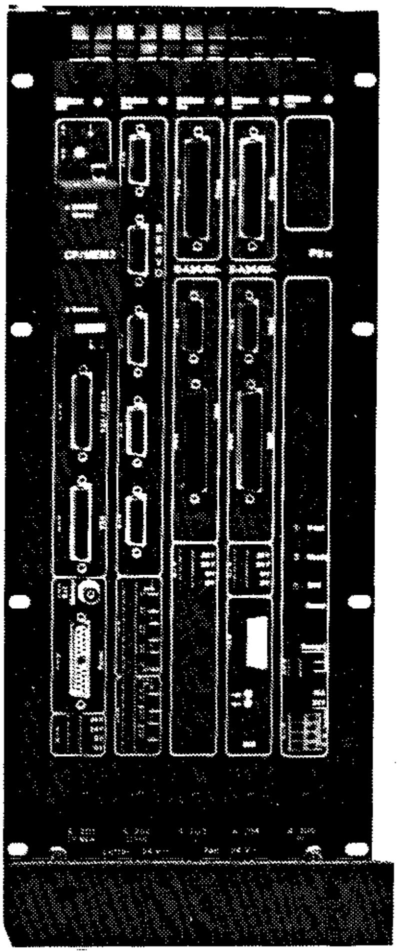
Operating Panel

graphic screen, 10", green
soft keys
main mode
input keyboard



Manual Panel

handwheel, jog buttons, override switches
customer keyboard
reentry / display distance to go
start / stop / emergency stop button



Logic Modules:

CP/MEM module:
connections for 2 serial data I/0 devices, operating panel, external VDU, battery and software module

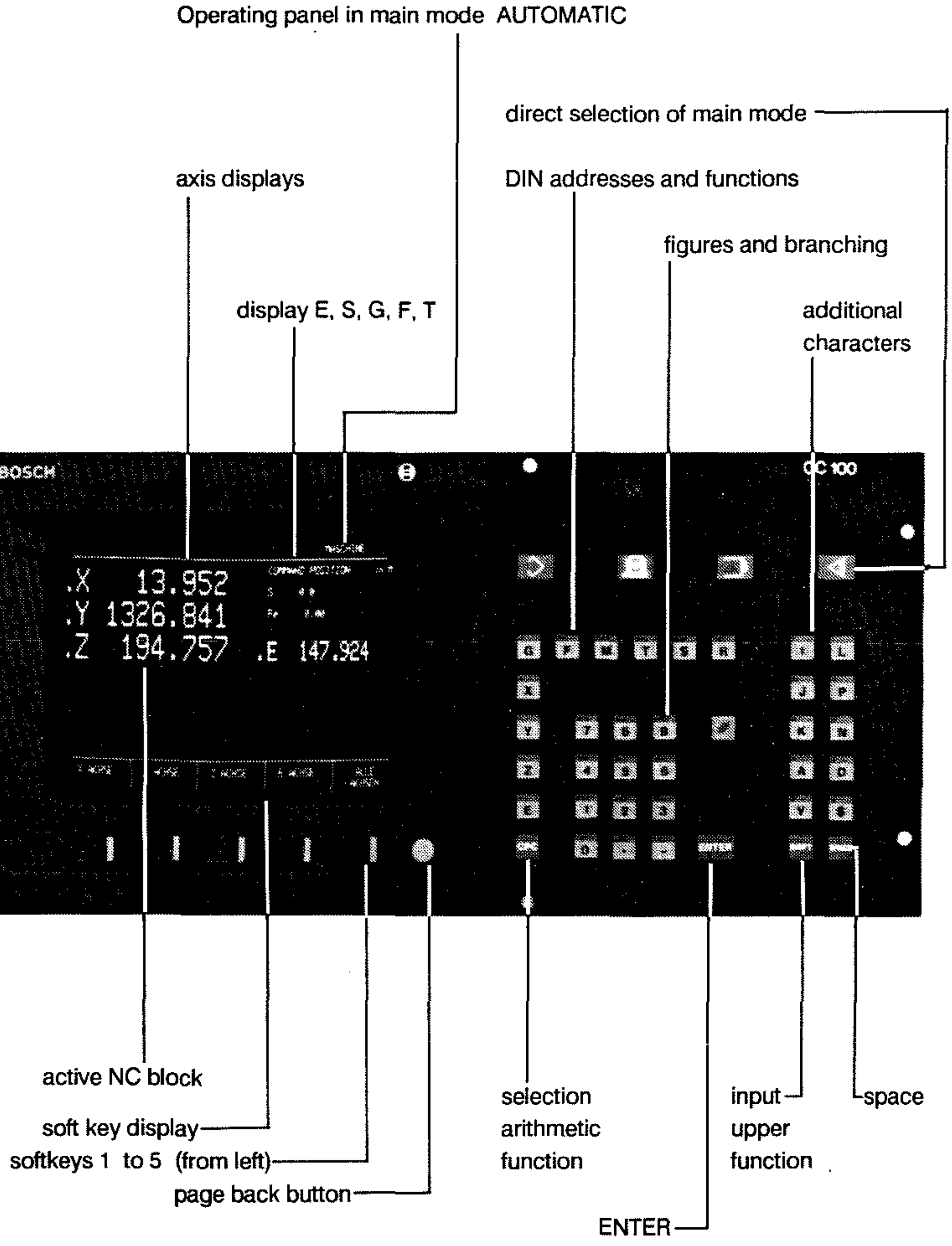
Module PS 75:
Displays for
- Ready (green)
- 24 V (green),
- internal voltage levels ok (green)
  reset button

connections for:
- ready 2
- 24V

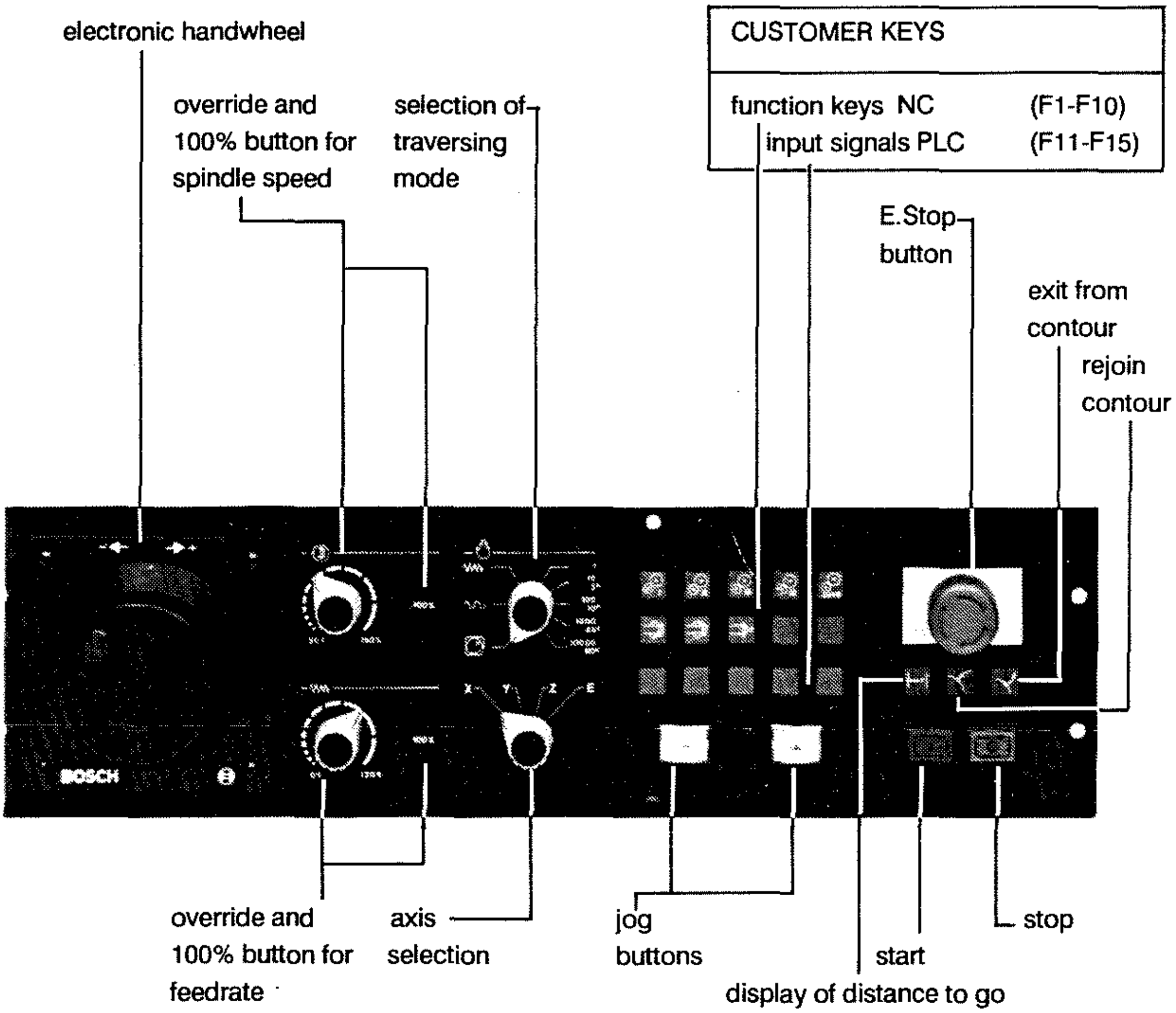
SERVO module:
connections for
5 incremental
measuring systems,
analogue outputs
time-critical signals

PIC module or
PLC connection

OPERATING PANEL



IUAL PANEL



ctions

Override potentiometers:

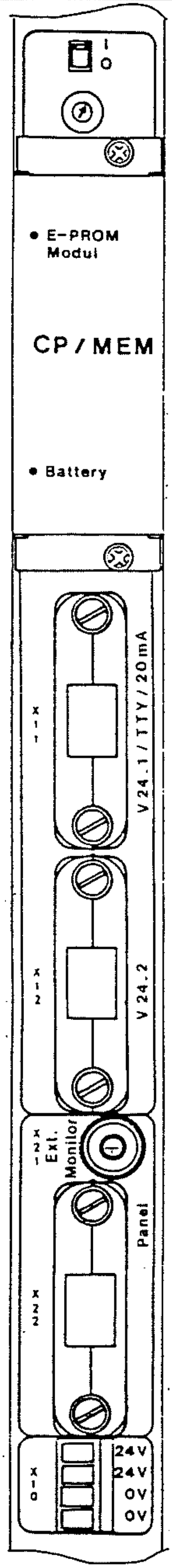
The feedrate value is set on the potentiometer in %.
The button deactivates the potentiometer (sets value to 100% when the potentiometer is set between 80 and 120%).

The potentiometers can be used in MACHINE and AUTOMATIC modes.

Customer keys:

Effective in MACHINE mode; Depression of one of the keys in the top two rows triggers an MDI function, which is stored in memory.

Bottom row for direct switching of PLC input signals.



CP/MEM

Overwrite protection switch for machine parameter area.
I = protected

"Start up" test during the runup phase;
switch position 1 = test active

battery

Buffer battery for data in RAM.
The battery must be replaced yearly.
The battery voltage is checked automatically in a 24 hr cycle and each time the control is switched on. If undervoltage is detected an error is signalled.

With normal battery discharge (no defect on PCB) a further buffer period of at least 14 days is guaranteed after the first error signal.

The CP/MEM incorporates a capacitor for the temporary buffering of the RAM data. If the battery is changed with the control switched off the period for which the supply is interrupted must not exceed 5 min. To change the battery the battery cover must be opened; the battery is fastened to the cover by means of a mounting.

Battery: 4.5 V alkali battery, part no. 107 - 913 572.

The executive system software is contained on a plug-in PCB (soft-board) under the cover.
To change the software remove the battery cover and withdraw the soft-board. (POWER OFF first!)

Serial Data Interfaces

X11
25-pole

Interface 1

V.24 or 20 mA

Desired characteristics selected by pin allocation.
Connector: sub-miniature D-type socket.

X12
25-pole

Interface 2

V.24

Second voltage interface; can be connected in addition to X 11.
Connector: sub-miniature D-type socket.

X21

Ext. VDU monitor (BAS signals)
Connection for an additional VDU with 75 Ohm.
Connector: BNC socket.

X22
20-pole

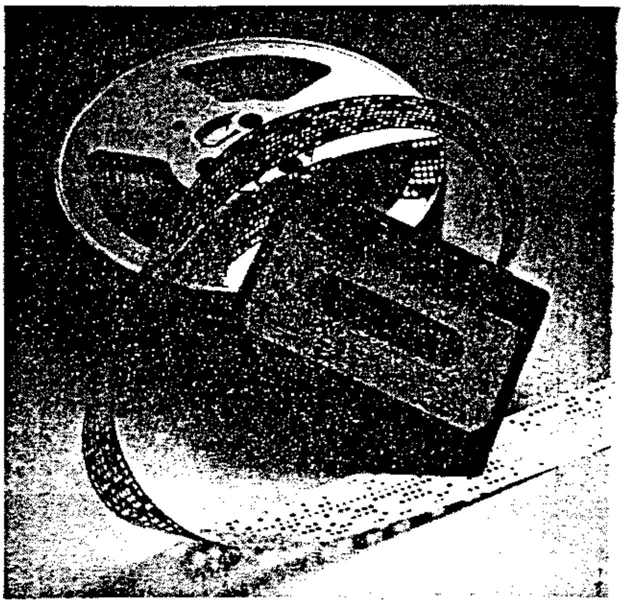
Operating panel connection.
Connection between control system and CC operating panel.
Connector: sub-miniature D-type socket with integral coaxial connector.

X10
4-pole

External 24 V **logic** DC supply (to supply the VDU in the operating panel)
Connector: Weidmüller terminal strip.
Max. cable size 1.5 mm²

INTERFACES

INTERFACES, general



The user can connect up to 2 external data terminals at the CP/MEM.

20 mA	1 device of this type can be connected to X11 (see page 1-5). This interface is particularly suitable for use where long distances are involved and/or where there is a high level of interference in the surroundings. With this type of interface one side is active (serves as source of current), the other must be made passive. This is achieved by specific pin allocations in the connections (see page 1-10, 1-11).
V.24	1 device of this type can be connected to X11 or X12. This interface allows higher transfer speeds than the TTY interface but is more susceptible to interference.

Control Signals

DTR	Data Terminal Ready: Status of readyness to receive data is output (output signal).
DSR	Data Set Ready: Status of permission to send is recognized (input signal).
Note:	Switch off handshake by means of a bridge, Pins 4 and 6 at the control side.

Data Lines

TX	Data output at the device sending the data.
RX	Receipt of data at the receiving device.

Make sure not to confuse the plugs when connecting the devices!
Only connect one device per interface (V.24/20mA) !

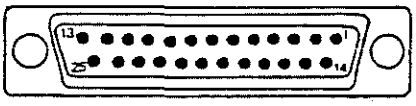
DATA FORMAT

1 start bit, 7 data bits, 1 stop bit, "even" parity bit
(1 start bit, 7 data bits, 2 stop bits, "even" parity bit for 110 Bd)

Control Characters (ASCII)

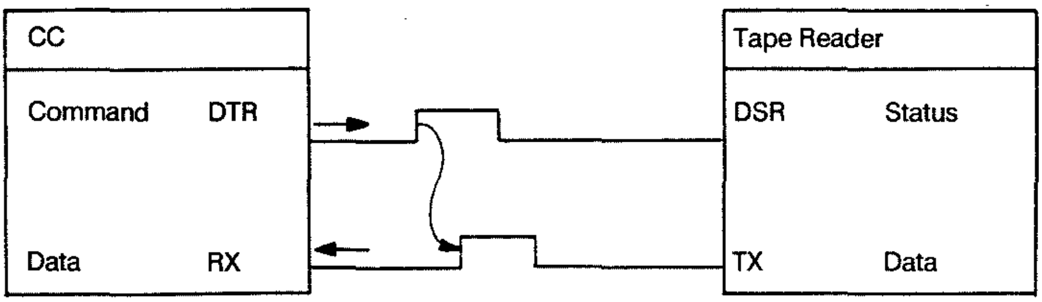
DC1	Tape reader ON or input START.
DC2	Punch ON or output START. Output comes from the controlling device. It starts the transmission.
DC3	Tape reader OFF or input STOP.
DC4	Punch OFF or output STOP. Output comes from the controlling device. It interrupts (stops) the transmission.
STX	Start of text.
ETX	End of text.
EOT	End of transmission.

Sub-miniature D-type connector
25-pole
socket on device
plug on cable

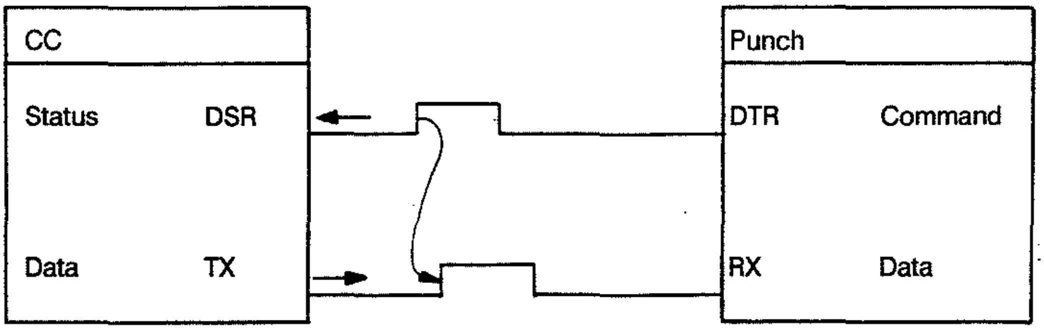


Plug: side for soldering

Reading in Data

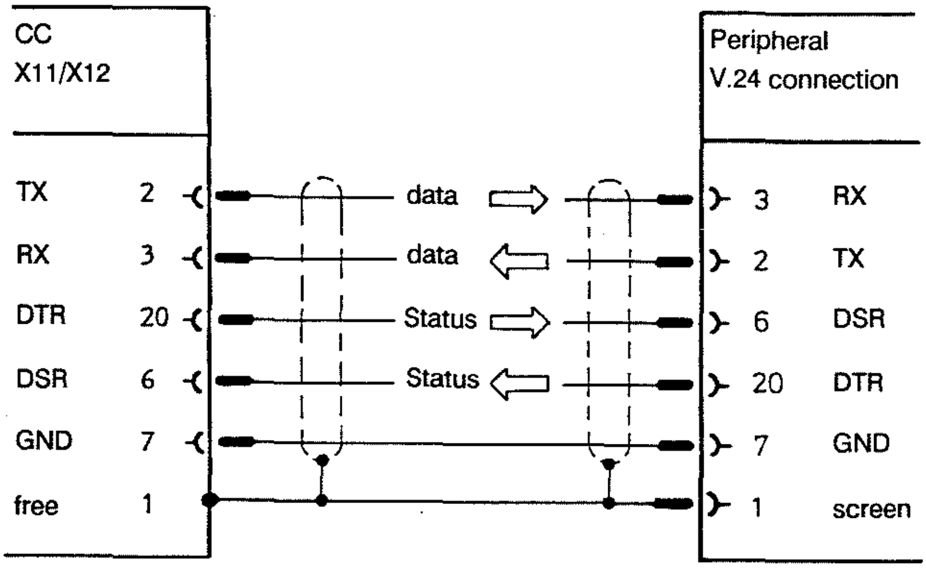


Data output



V.24 CABLE

Cable length	max. 15 m
	transfer rate max. 9600 Baud, always with handshake
Signal levels	high +3 V to +12 V low -3 V to - 12 V



Note: X12 interface does not use handshake signals.

20 mA CABLE

Cable lengths:	CC active	max. 15 m
	CC passive	max. 100 m

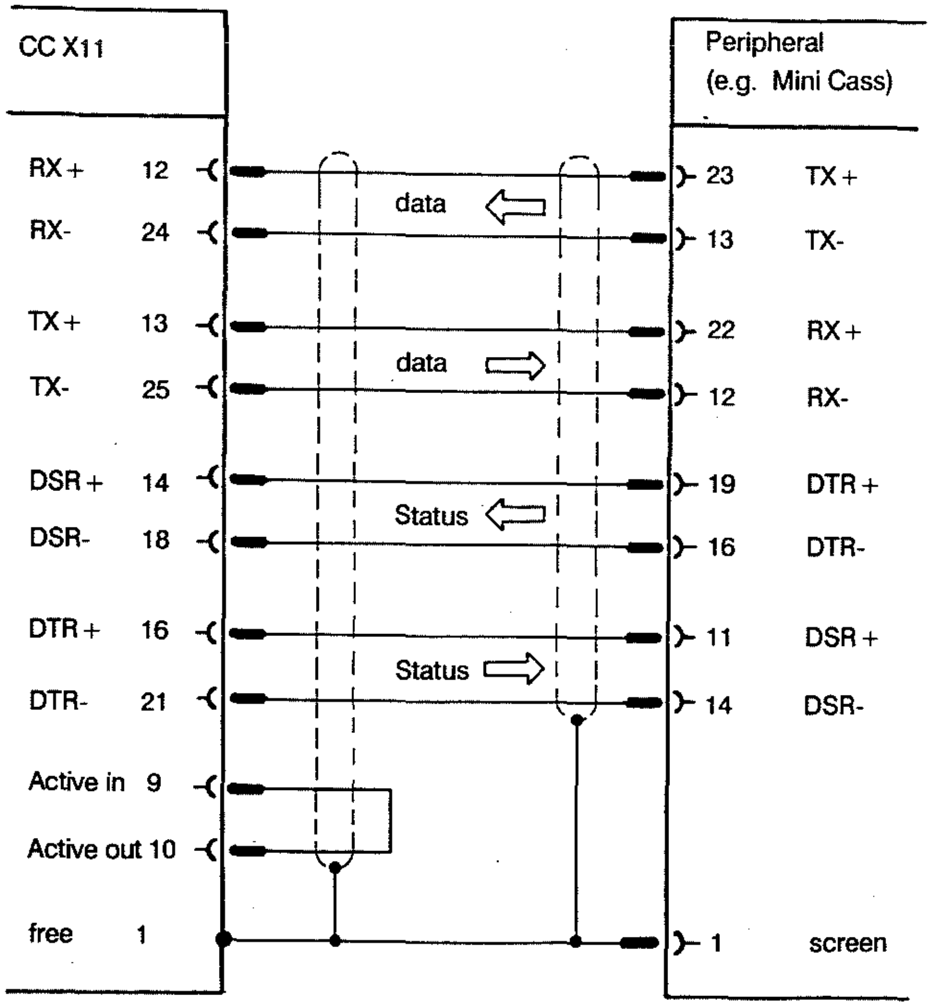
Baudrates:
max. 4800 Bd with handshake
max. 300 Bd without handshake

Signal levels:	high	approx. 20 mA
	low	approx. 0 mA

max. external voltage drop 2 V

CC active The CC serves as source of current:

Pin Allocation

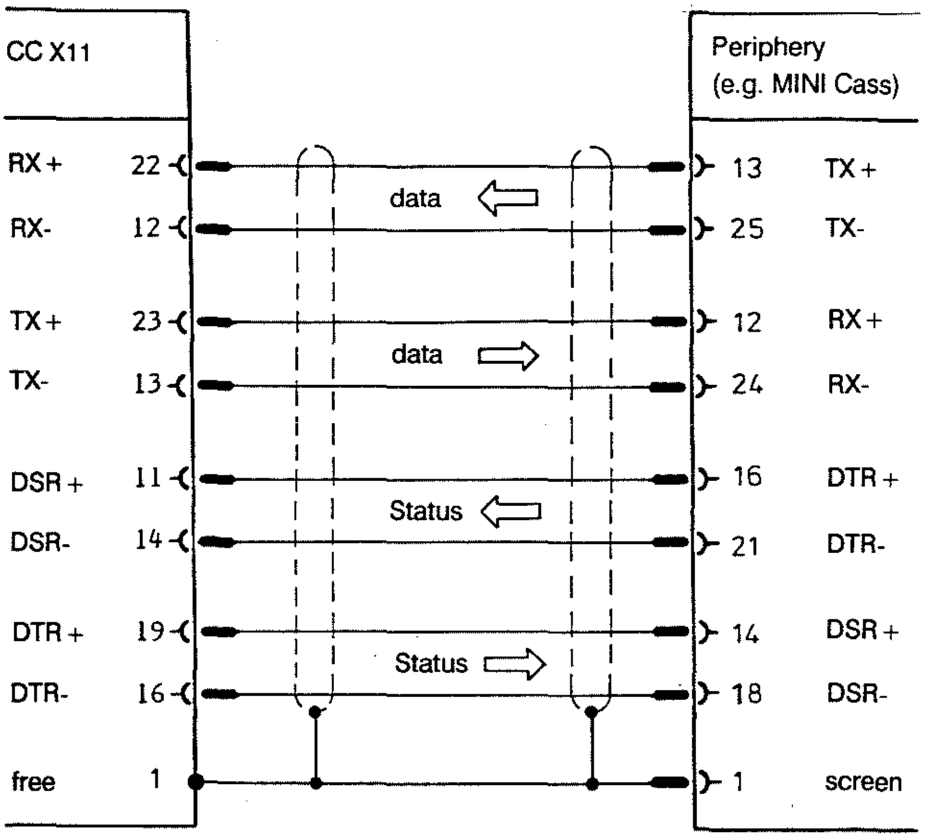


20 mA TERMINAL

CC passive

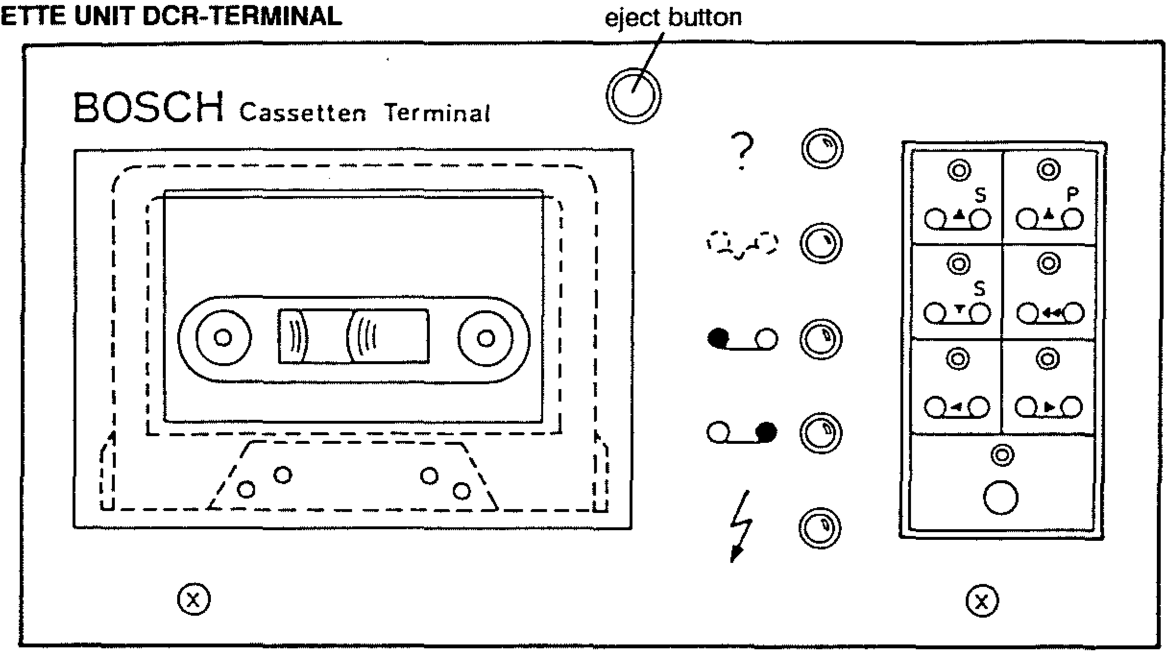
The peripheral device serves as source of current.
Max. admissible voltage drop in the control 2V.
The supply to the driving device can be up to 24V.

Pin Allocation



PERIPHERALS

CASSETTE UNIT DCR-TERMINAL



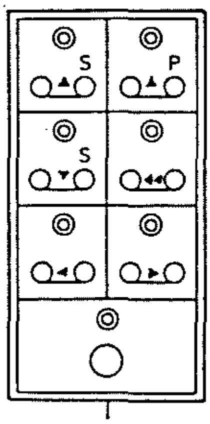
GENERAL

- recording process:
  ECMA 34
- storage capacity:
  256 KB, unformatted
- data format and baudrate
  set on back
- parallel and serial
  interface

DISPLAYS

?	◎	- device not ready
⚇	◎	- bad cassette
●‿○	◎	- beginning of recording
○‿●	◎	- end of recording
ϟ	◎	- mains and DCR switched on

OPERATING ELEMENTS



Read (DCR ⟶ serial) -

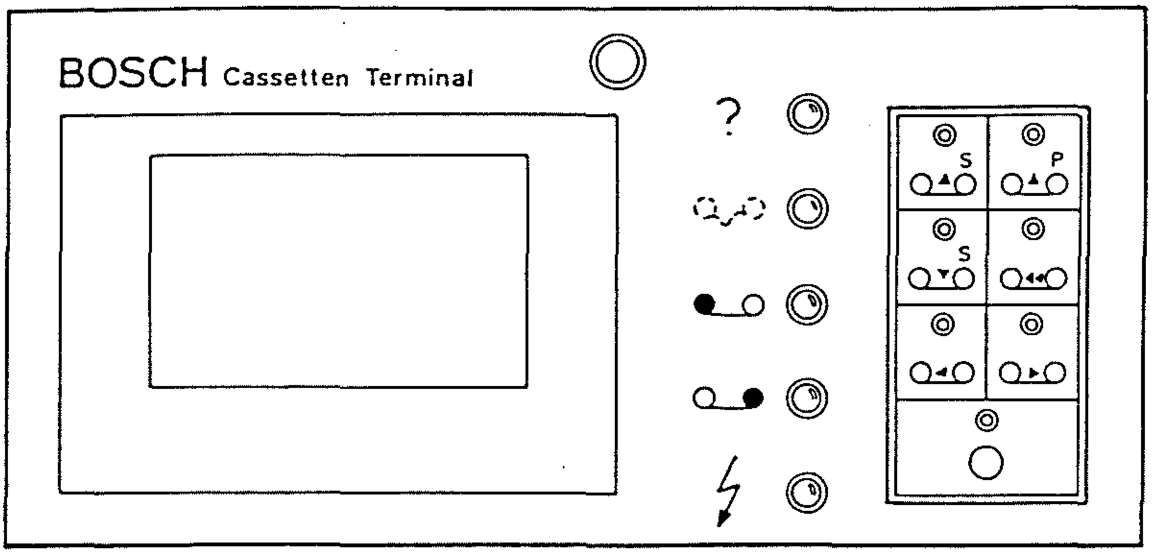
Write (serial ⟶ DCR) -

Search backwards -

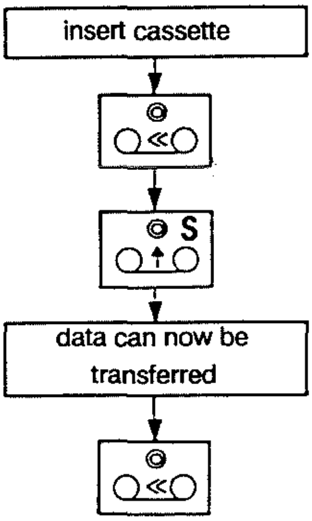
- Read (DCR ⟶ parallel)

- fast rewind

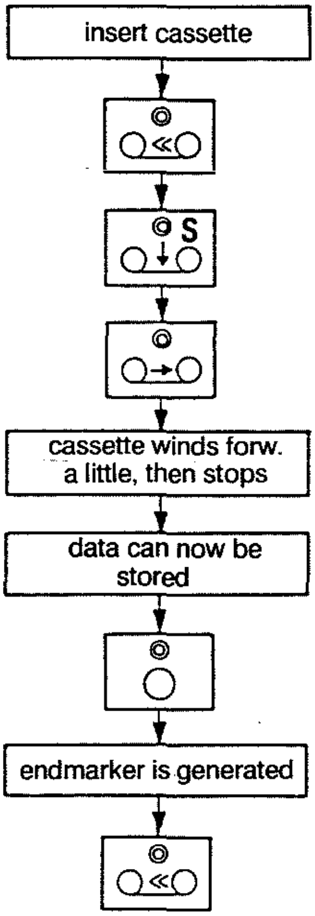
- Search forwards



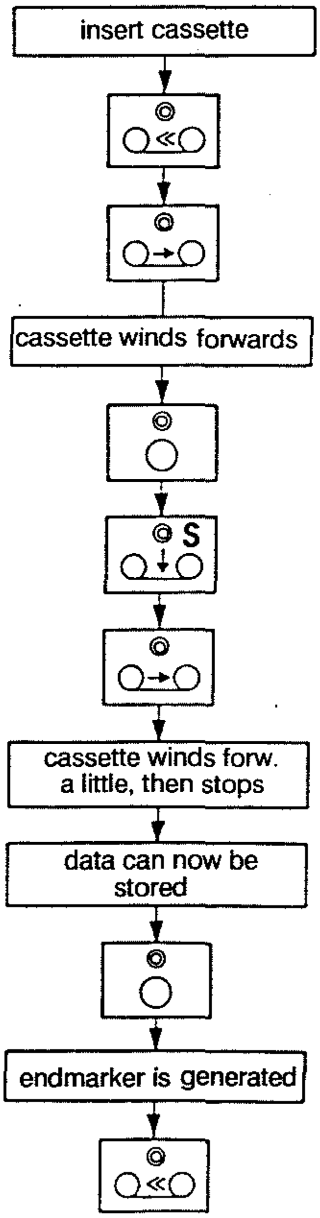
READ
(DCR ⟶ CC100 M)



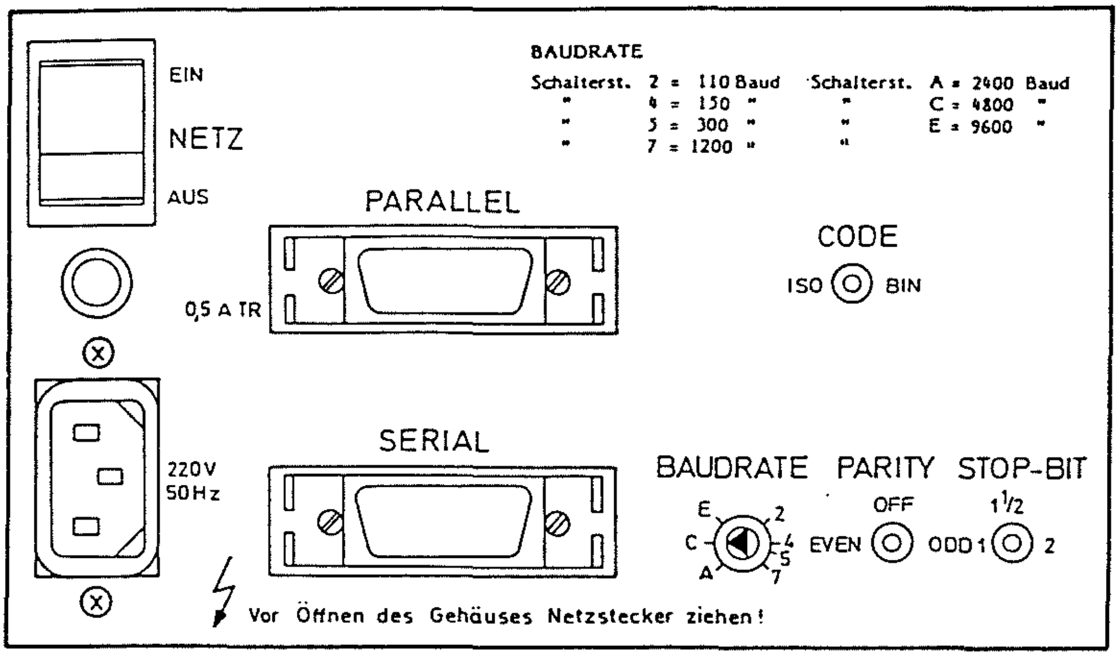
WRITE I
(CC100 M ⟶ DCR)



WRITE II
(CC100 M ⟶ DCR)



DCR Rear Panel



Settings:

1. **CODE:** BIN
2. **BAUDRATE:** C (= 4800 Bd)
3. **PARITY:** EVEN
4. **STOP BIT**: 1 (as in control)
5. Connector for use with CC 100 M is **SERIAL**
6. Cable used: 046266

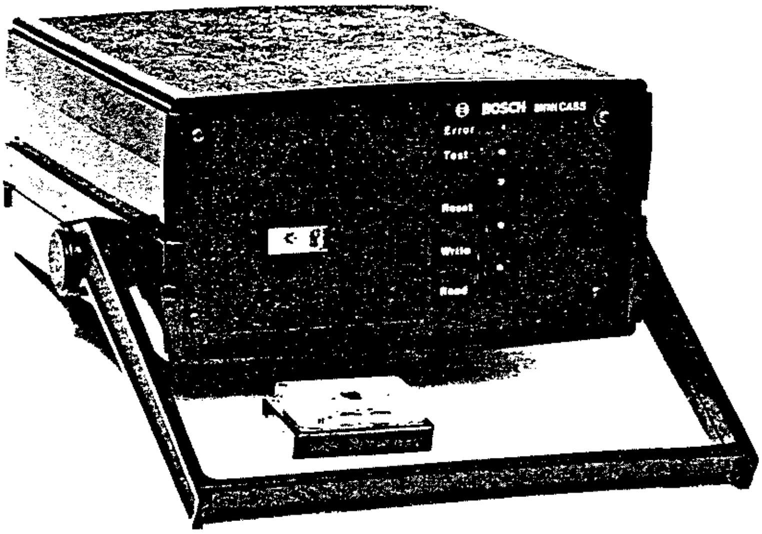
<u>Explanations:</u>

NETZ EIN/AUS - MAINS ON/OFF

Schalterst. - switch position

Vor Öffnen des Gehäuses Netzstecker ziehen! - Unplug mains cable before opening the housing !

MINI CASSETTE UNIT



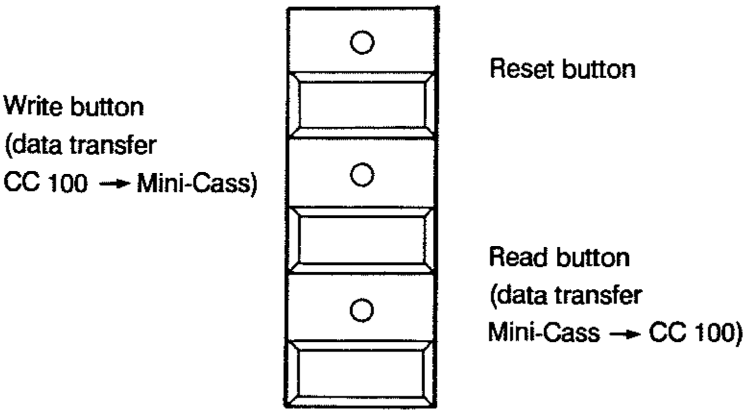
GENERAL

- recording process:
  ECMA 34
- storage capacity:
  20 KB each side
- data format and baudrate
  set on back
- automatic self-diagnosis
  after switch-on with
  "Ready" indicator
- serial interface with
  V24 or 20 mA

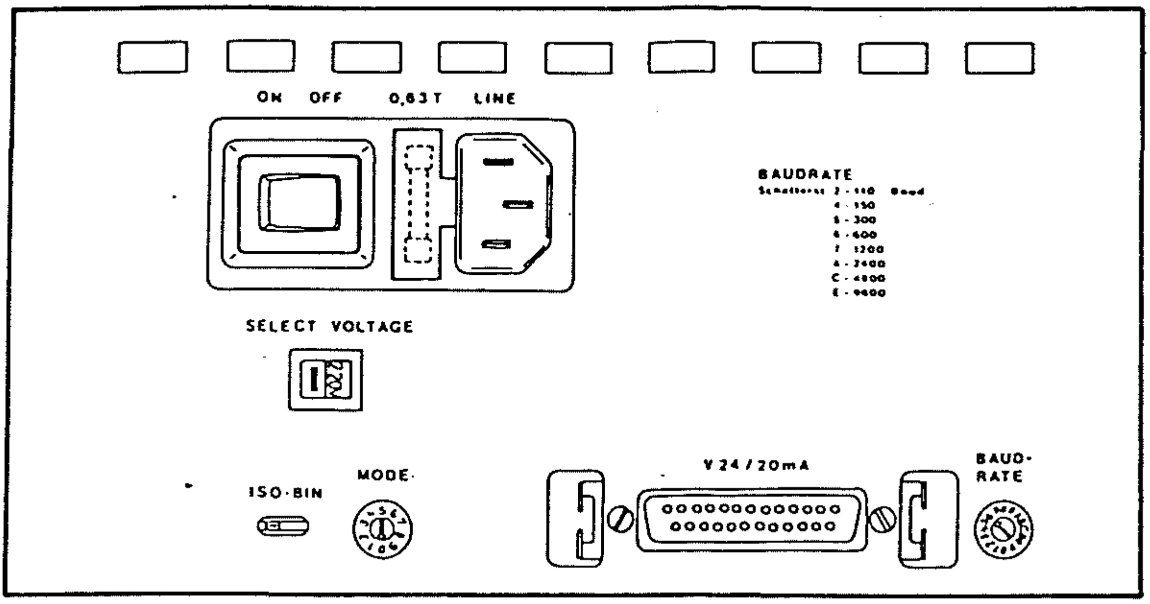
DISPLAYS

Error	○	error indicator
Test	○	ready indicator

OPERATING ELEMENTS



Rear Panel of MINI CASS



Settings:

1. code: BIN

2. **MODE:** 4

3. **BAUDRATE:** 7 (= 1200 Baud)

4. cable used: 20 mA - 2.5 m part no. 046266

Data carrier:

Digital mini-cassette LDB 400 part no. 910749

Control	Mode	Number of data bits	Parity bit	Start bit	Stop bit	Operating buttons active	Binary data
micro 5/8 CC 100/200/300	4	7	even	1	1	yes	no

PROGRAM HEADER

EXTERNAL PROGRAM PRODUCTION

The following text explains the methods by which part programs and part program type subprograms (or cycles) are produced.

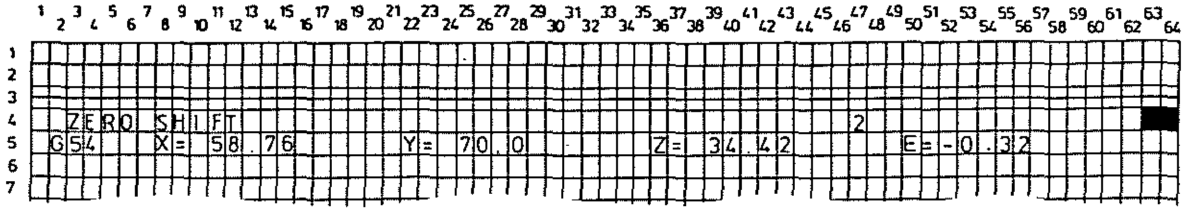
Such programs are constructed from program language elements to DIN 66025 and can be produced by one of the following methods:

1. via keyboard input, using the program editor in the NC
2. via the manual panel with 'Teach In', in the NC
3. via a programming unit onto a data carrier (paper tape, for instance), outside the NC
4. by computer, outside the NC

Programs produced outside the NC must conform to the NC machine code and the NC syntax.
In addition programs which are input from a data carrier (tape or digital cassette) or via an interface (V24/20 mA) must have a leader (header) and a trailer. Leader and trailer, the beginning of the individual program lines, as well as the program identifications of the header lines of data blocks must be provided in the correct format.

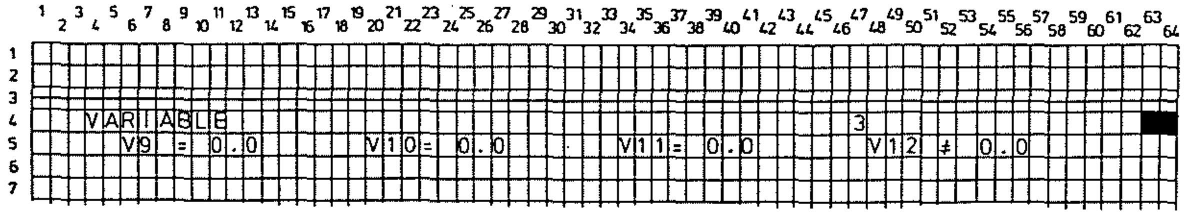
Note: When data needs to be transmitted the external data carrier must be activated before the control.

CC100 - PROGRAM HEADER (general format)

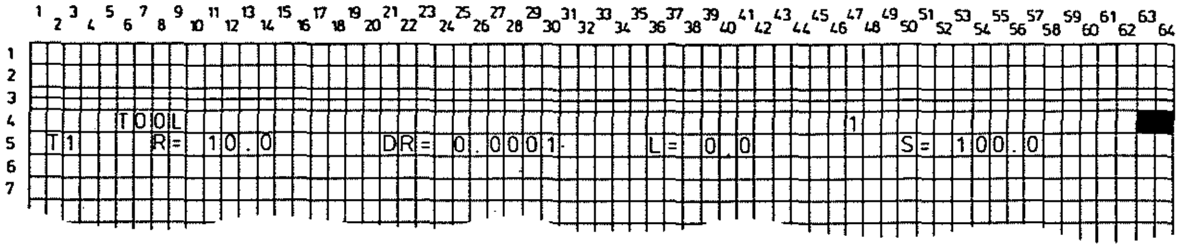


```
  ZERO SHIFT                                  2
 G54   X= 58.76       Y= 70.0       Z= 34.42     E=-0.32
```

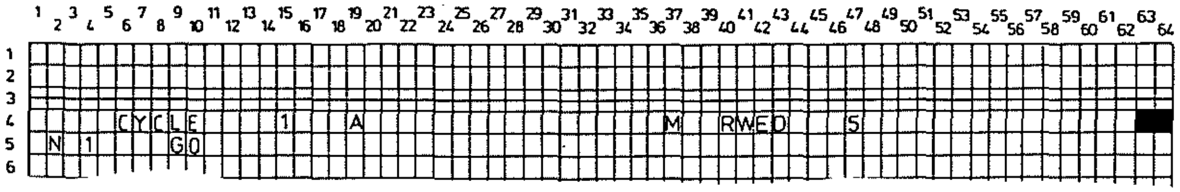
*)



```
   VARIABLE                                   3
     V9  = 0.0    V10= 0.0     V11= 0.0      V12 ≠ 0.0
```

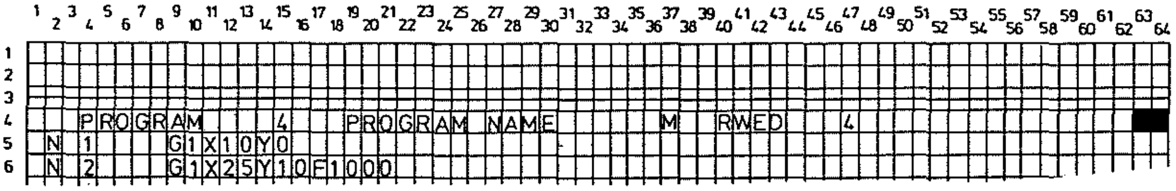


```
     TOOL                                    1
T1    R= 10.0      DR= 0.0001     L= 0.0       S= 100.0
```



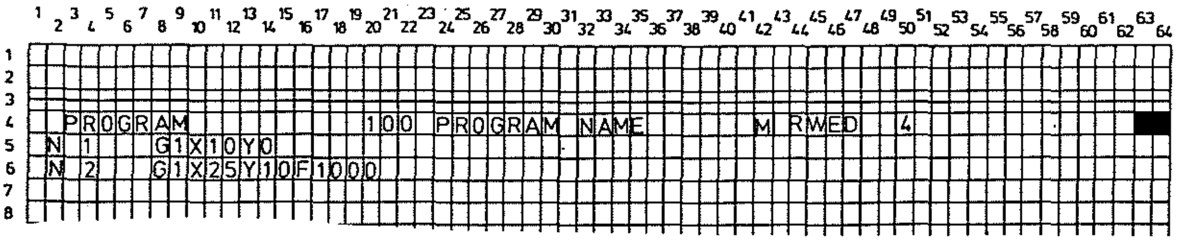
```
     CYCLE   1   A                  M  RWED   5
N 1    G0
```

Program header with 1 and 2-digit program numbers



```
   PROGRAM   4   PROGRAM NAME       M  RWED   4
N 1    G1X10Y0
N 2    G1X25Y10F1000
```

Program header with program numbers of over 2 digits



```
  PROGRAM          100 PROGRAM NAME     M RWED  4
N 1    G1X10Y0
N 2    G1X25Y10F1000
```

Note: The control characters listed below are generated as follows:

DC2 = CTRL R (device control 2)
STX = CTRL B (start of text)
ETX = CTRL C (end of text)
EOT = CTRL D (end of transmission)
DC4 = CTRL T (device control 4)

*) CR LF must be in columns 63 and 64 respectively.

Program Header - Original Print-out

Data is output by the control in this format, and the same format must be used when programming data externally (see also previous page).

Tool

IDENTIFICATION AS TOOL DATA

```
-----TOOL-----------------------------------1------------
T1    R= 18.0      DR= 0.0       L= 200.0     S= 0.0
T2    R= 0.0       DR= 0.0       L= 0.0       S= 0.0
```

Zero Shift

IDENTIFICATION AS ZERO SHIFT

```
--ZERO-SHIFT--------------------------------2------------
G54   X= 91.20052  Y= 0.0        Z= 55555.0   E= 0.0
G55   X= 0.0       Y= 0.0        Z= 0.0       E= 0.0
```

Variable

IDENTIFICATION AS VARIABLE

```
---VARIABLE-------------- ----- --------------3------------
     V1 = 45.0     V2 = 0.707106
```

PROGR. NAME

INCH/METR.

ACCESS LEVEL

R/W ENABLED

Program

```
---PROGRAM----1---REF--------------M--RWED---4------------
N 1    G879
N 2    G1 X200 F2000
N 3    M2
N 4    (PROGRAM END)
```

Cycle

IDENTIFICATION AS CYCLE

```
-----CYCLE----20--***************---M--RWED---5------------
N 1    G92 X0 Y0
N 2    M21
```

\- = space character

Identification Letters

The access level is identified as follows:

RWED read, write execute, delete permitted

RE read, execute permitted

E execute permitted (cycles only)

Dimensioning:
M = metric   I = inch

PROGRAM HEADER IN DFS FORMAT

The CC 100 program header in DFS format has been designed on the basis of the header format of the cc 200/300, in order to create uniformity in this area for the future. Specific types of files can be loaded and output.

The uniform DFS program header has the following (basic) format:

(DFS * , * file type * file number [, * [file name *] suffix*] , * access level)

At the positions indicated by an asterisk it is possible to insert one, several or no space character (s).

Different possibilities

(DFS, Pxx)
(DFS, Pxx, . suffix)
(DFS, Pxx, name . suffix)
(DFS, Pxx, . suffix, RWED)
(DFS, Pxx, name . suffix, RWED)

Explanations

- **DFS**

Identification of the program header in DFS format (defined storage).

- **File type**

Specific letters identify the file type:

P = program
C = cycle
E = text
K = compensation table (K0)
V = zero shift table (V0)
X = variables (X0)
L = machine parameters

File number
- Program numbers can contain up to 9 digits, cycle numbers up to 2 digits.

File name
The file name can contain up to 15 characters, which can be letters as well as numbers. Tables are transferred without name. The file type to be transferred is simply identified as X0, V0 or K0.

- **Suffix**

The suffix consists of one letter and determines the dimensioning method (I = inch/M = metric).
It is separated from the file name by a decimal point.

- **Access level**

The access level is defined by a 2-character code.
2-char.: RE (read, execute)
4-char.: RWED (read, write, execute, delete)

Note Input of file name, suffix and access level is not compulsory.
They are purely optional .
If no file name is programmed the suffix can be ommitted. The control will then automatically assume the dimensioning to be metric (= suffix M).
If a file name is stated in the program header the suffix must be entered too.

Examples of DFS program header for different file types

(DFS, P12)	- transfer of a single program, program number 12
(DFS, P10,.M)	- transfer of a metric program, program number 10
(DFS, C 4,TOOL CHANGE . I)	- transfer of the tool change cycle in inch format
(DFS, P1, TEST RAPID.M,RWED)	- transfer of program P1 with metric dimensions under access level RWED
(DFS, X0)	- transfer of the variable table
(DFS, K0)	" compensation table
(DFS, V0)	" zero shift table

Examples:

```
(DFS,P 1,TEST RAPID.M,RWED)

(DFS,C 79,.M,RWED)

(DFS,K 0)

(DFS,X 0)

(DFS,V 0)
```

OPERATING SEQUENCES FOR OUTPUT AND INPUT

The files to be **output** are determined via soft key and marked on the screen in reverse video:

SELECTED FILE ONLY	- Output if specific file had previously been selected.
PROGRAMS OR CYCLES	- Output if no specific file had previously been selected. Whether programs or cycles are output depends on the file type active at the time.
PROGRAMS AND CYCLES	- Selection via soft key.
FILE + TOOLS FILE + ZERO SHIFT FILE + VARIABLES	- Output of a specific file, as well as tool, zero shift or variable file.

Files to be **loaded** can be transferred several at a time in any sequence.

If loading via interface is selected in main mode MEMORY a specific number of files can be selected by soft key operation:

ALL FILES YES NO	START	PORT NO	BAUDRATE	CONTROL YES NO

How many? (1...99) ☐

CHECKSUM

- Whatever the tape format, programs can be output with or without checksum.
- The DFS program header is output without checksum.
- In each program block the checksum is inserted directly before the CR LF control character.

Position, calculation, input/output of the checksum

1) Position of the checksum

At the end of the data and before CR LF, a space,
the character ":" and then the checksum value (a 2-digit number)are written.

e.g. N-11- - - G1 CR LF becomes
N-11- - - G1-:nn CR LF

\- = space
nn = 2-digit number for the checksum

2) How to calculate the checksum

Every character between the LF of the previous line and the ":" is included into the checksum calculation. The ASCII value of each character is added up and multiples of 256 are removed until 255 or less remain, and this remainder is converted into a hexadecimal number.

e.g. N-11- - -G1-:nn CR

CODE	ASCII VALUE
N	78
-	32
1	49
1	49
-	32
-	32
-	32
G	71
1	49
-	32

456 - 256 = 200 = C8
The block will now read: N-11- - -G1-:C8 CR LF

3) Input/output of the checksum

INPUT	SK "CONTROL YES" active	- control checks syntax
	SK "CONTROL NO" active	- control checks the checksum, if it exists, otherwise it checks the syntax
OUTPUT	SK "CHECKSUM YES/NO" is called up via SK "FORMAT".	
	SK "CHECKSUM YES" active	- programs are stored **with** checksum
	SK "CHECKSUM NO" active	- programs are stored **without** checksum

2. OPERATING

MAIN MODES

SURVEY The operation of the control is subdivided into the following main modes, which are directly selectable by pushbuttons:

EDIT	MACHINE	AUTOMATIC	INFO
working with stored data	manual operation	execution of programs	additional information
display input, modification of: programs,	direct execution without storage execution of cycles	execution of stored programs, cycles	status displays, NC/IO, axis displays, error list
subprograms, cycles, tools zero shifts variables	reference axes, reference cycle, MDI, manual machine, operation, teach in	execution continuous/ block by block, variable step size, block selection, break points, reentry	deletion of: programs, variables, tool/zero shift tables control reset
input and output via data inter-faces V.24/20 mA	customer keys handwheel jog buttons	with/without path compensation tool length compensation	MTB SERVICE only for machine tool builder
baudrates	distance to go display	CPC test distance to go display	SERVICE load M-parameters
automatic generation of header lines for PROGRAM/ CYCLE etc.		milling conditions	logbook set clock mode read in text

The active main mode is displayed continuously in the top right corner of the screen.

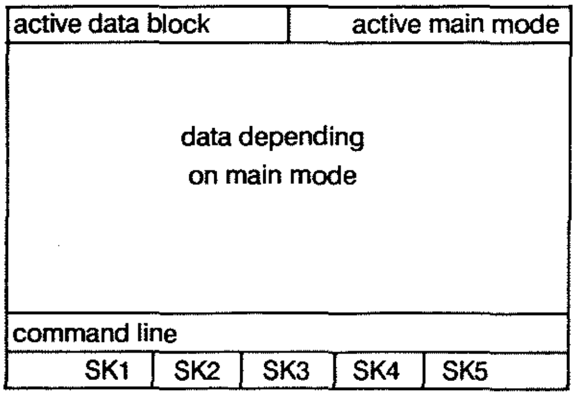
To come out of the current main mode altogether:

> Use the page back button to revert through the levels until the 1st soft key level is reached, then select new mode. Exception: For change-over MEMORY/EDIT to AUTOMATIC no paging back required.

To come out of the current main mode temporarily:

> Select a different main mode directly. The old main mode is retained in the background (display flashes) and can be reactivated by pressing the relevant mode key once more.

Subdivision of VDU Display



data blocks:

program
cycle
variable table
zero shift table
tool table

Reset Conditions

Immediately after switch-on the following modal conditions are active:

G1	linear interpolation
G17	plane X/Y
G39	programmed mirror image off
G40	radius compensation off
G53	no zero offset
G62	in position operation off
G65	programmed feedrate applies to cutter centre path
G66	feedrate and spindle speed can be modified
G68/ G69	contour transition as arc/intersection (dependent on machine parameter)
G80	no fixed cycle active
G90	absolute dimensions
G94	feedrate in mm/min
G97	direct spindle speed programming
	scale for factor 1
	no feedrate effective

These modal conditions are active in all main modes.

The G-codes which become active on switch-on are denoted with an "A" in the following descriptions, i.e. G39A.

Note

When working in AUTOMATIC or MACHINE mode the control will output the following types of messages, as and when appropriate:

MESSAGE xxx - further operation possible

ERROR xxx - further operation is inhibited

The content of the message can be displayed in INFO mode.

EDIT

Access to Data

In this main mode all user data can be handled (see EDITOR).

Selectable data blocks:

- tool table
- zero shift table
- variable table
- programs
- cycles

The menue for part programs and cycles can be paged forwards with soft key "NEXT PAGE".

Access Levels

Unauthorized accessing of the data can be prevented via softkey operation. Execution is always permitted.

The access levels are expressed as follows:

- RWED read, write, execute and delete are possible
- RE only reading and executing are possible
- E only executing is possible (cycles only)

Dimensioning

The dimensions can be selected by soft key to be in metric or inch.

Display in index and in "active datablock" line:

- M metric
- I inch

Commands

Under this SK the following functions are available in 2 levels:

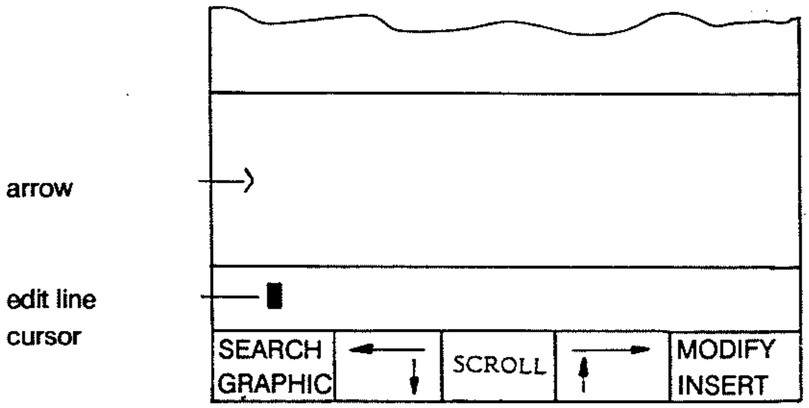
- resequence block numbers
- transfer program to a cycle
- rename a file
- inch/metric
- copy file
- file protection
- delete file

Data Interfaces

See chapter on "Data Handling"

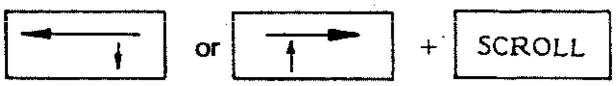
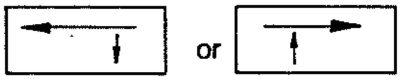
Copy

Programs stored in the memory can be duplicated with SK function "COPY". The user must enter a new file name and the control will select the file number.



Selection via SK "PROGRAMS" or "CYCLES", program name or number
SK "EDIT"

The position of the arrow indicates which line is being worked with. This block is repeated in the edit line which contains a cursor (bright rectangle)

Cursor Functions	Switch-over between MODIFY/INSERT	MODIFY INSERT
	Scrolling blocks up/down by simultaneous actuation of	
	Moving cursor sideways	
Block Selection	The cursor is placed to the right of the position at which a letter is to be inserted/modified.	
Search Functions	A characteristic string (sequence of letters, numbers and characters) from the required line is entered, i.e. G41.	ENTER
Delete	- individual character to the left of the cursor	///
Line Delete	- content of the line to the right of the cursor is deleted	MODIFY SHIFT ///
Modify	- First delete individual character,	///
	- then key in new character(s)	ENTER
Insert		INSERT
	- enter new character(s)	ENTER

MACHINE

MANUAL MACHINE OPERATION

The manual panel is always activated in MACHINE mode.

REFERENCE AXES	REFERENCE CYCLE	MDI	TEACH IN	INCH METRIC

driving to the reference point in one or several (all) axes

input of instructions for direct execution; execution with NC start e.g. G01 X100 START, see page 2 - 6

switching of dimensioning unit INCH-METRIC

reference cycle call (cycle 79) start button

TEACH IN

Recording of elements of a sample contour (see p. 2 - 7)

MDI

After SK selection of MDI one block can be executed after the relevant data has been entered. The execution is initiated with the start button.
Under the SK HELP the permanently stored drilling and milling cycles can be selected, parameterized and executed, as well as the user-definable cycles.

REFERENCE AXES	REFERENCE CYCLE	**MDI**	TEACH IN	INCH METRIC

HELP	←		→	CLEAR BLOCK

			BORING CYCLES	CONTOUR CYCLES

MTB-specific soft keys (cycles)

Note:

- It is not possible to return to previous SK levels while a block/cycle is being executed.

- G41/G42 are not permitted.

- MTB cycle PRIOTITY ROUTINE can not be called up.

- Axes which have been driven onto the software limit switches can only be moved by means of the JOG

  buttons + - in reverse direction.

When working in manual mode the type of traversing movement needs to be defined:

- With the jog buttons the axes can be traversed individually in incremental steps (of 1, 10, 100, 1000 or 10,000 increments). The max. feedrate corresponds to the limit determined by the machine parameter for manual feed (1 - 120,000 mm/min).

- The electronic handwheel can be activated for individual axes.

- Change-over between feed and rapid.

TEACH IN

Definition By tracking the outline of a sample contour with the machine the specific contour features are recorded by key actuation (soft key RECORD).
During this procedure the control stores the position values of all axes.
A circular movement is generated by positioning to three points of the circle (soft key CIRCLE COMPUTE).

MDI function As in MDI mode blocks can be keyed in. The data is transferred into memory with SK "RECORD".

Operating

Main mode MACHINE

REFERENCE AXES	REFERENCE CYCLE	MDI	TEACH IN	INCH METRIC

RECORD	←	CIRCLE COMPUTE	→	CLEAR BLOCK

Function Keys

RECORD

- Storing positions of moved axes
- Storing entered blocks
- Storing positions of blocks generated internally

CIRCLE COMPUTE

- Automatic calculation of circles
- The CC 100 calculates circle data from 3 scanned points (SK 'RECORD POINT 1', 'RECORD POINT 2' and 'RECORD POINT 3')
- Circular interpolation G2/G3 is also modal in TEACH IN mode. If a linear movement is to follow G0/G1 must be programmend: Key in G0/G1 before the linear movement and transfer into memory with SK RECORD.

CLEAR BLOCK

- Clearing blocks which have not yet been stored from the edit line.

TEACH IN

Calculation of Circles with Parameter R

The control calculates the radius R from the 3 recorded axis positions and generates the circular contour.

The current axis position is the Ist point for the calculation of the circle.

The display will show the last axis position with the calculated radius.

Display

G2/3 X... Y... R...
The block is stored with soft key RECORD.

Note

- The CC 100 automatically generates a program with the name "TEACH IN". If a program with this name is already stored in the memory, this program has the newly entered TEACH IN functions added to it.
  If several independent programs are to be generated via TEACH IN, the old program must first be renamed in EDIT mode with SK RENAME.

- Switching of the dimensioning unit INCH/METRIC during TEACH IN operation is not permitted.
  Should it be attempted an error message will be displayed:
  "inch/metric selection incorrect".

AUTOMATIC

Execution of programs and/or cycles from memory.

PROGRAM / CYCLE - Selection

The stored cycles and programs are listed in ascending numerical order. The selection is made by entering the name or the number.

OPERATING PROCEDURE BEFORE START OF PROGRAM/CYCLE

NORMAL step: no

DRY RUN RAPID	STEP	SELECT BREAKPOINT	MILLING CONDITIONS	SELECT STARTPOINT

SELECT BREAKPOINT: setting a break point in the program

SELECT STARTPOINT: setting a program start point

Note:
If a start point has been selected and the program is restarted after M30 the execution will begin at the set start point.

Selection of:
DRY RUN - test without movement
RAPID - test in rapid
NORMAL - execution as per program
active mode displayed in the prompt line.

selection of step size (1-9):
1 : single block
2 : double block
.
.
9 : ninefold block
(= stop every 9 blocks)

COLL TEST ON/OFF	WITH STOP YES/NO	LENGTH COMP ON/OFF	CUT. COMP ON/OFF	CPC TEST

COLL TEST ON/OFF: when executing programs with path compensation the tool coll. monitoring can be switched off (shortening of block cycle time)

WITH STOP YES/NO: during dry run, if YES the program stops at error found; if NO the program runs to the end; errors are displayed in INFO mode

LENGTH COMP ON/OFF, CUT. COMP ON/OFF: switching off tool compensation for test purposes

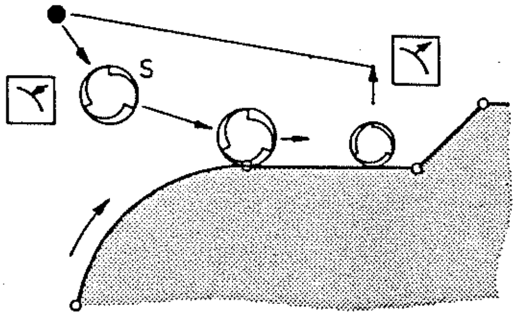
CPC TEST: CPC test facilitates the DEBUGGING of parametric programs

AUTOMATIC

INTERRUPTION / RE-ENTRY during program execution

Possibility of external intervention by the operator with tool compensation active / not active, after at least one block has been executed completely:

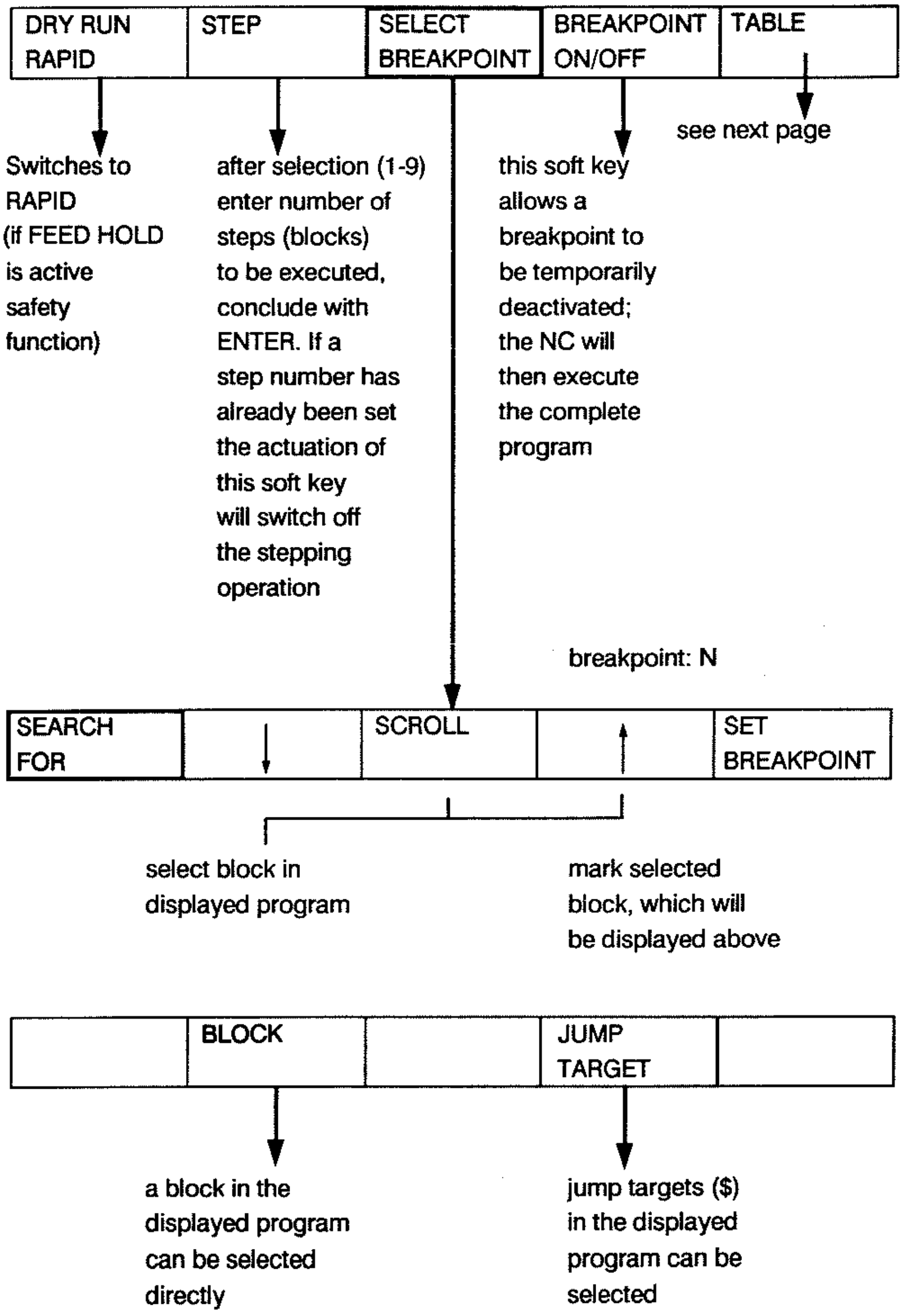
Sequence

	Response of machine and possible actions:
a) Cycle stop	feed hold is effective
b) Press	manual mode/MDI are activated
c) Manual intervention movement away from contour for measuring purposes, for instance	manual panel is active. spindle can be stopped or oriented
d) Tool change with	
- replacement by identical tool	old values are retained,
- replacement by a different tool	input of new tool data is possible (tool wear is set to 0)
Tool Change 	it is also possible to modify the active block; re-entry onto linear and circular contour elements
e) Drive to suitable position S to start re-entry	**This position must allow direct traversing onto the contour.**(no automatic evasion of obstacles)
f) Press	control drives back onto the contour, with the tool centre vertical above the beginning of the unfinished contour-program
g)	execution is resumed

Note:

- G92 must not be active (see chapter 3)
- If main mode AUTOMATIC is selected between exit and reentry the reentry operation is abandoned and the basic display for main mode AUTOMATIC is displayed. Continuation is possible via reselection of the program and CYCLE START.

OPERATING PROCEDURE AFTER CYCLE START



DRY RUN RAPID	STEP	SELECT BREAKPOINT	BREAKPOINT ON/OFF	TABLE

DRY RUN RAPID: Switches to RAPID (if FEED HOLD is active safety function)

STEP: after selection (1-9) enter number of steps (blocks) to be executed, conclude with ENTER. If a step number has already been set the actuation of this soft key will switch off the stepping operation

BREAKPOINT ON/OFF: this soft key allows a breakpoint to be temporarily deactivated; the NC will then execute the complete program

TABLE: see next page

breakpoint: N

SEARCH FOR	↓	SCROLL	↑	SET BREAKPOINT

↓ SCROLL ↑: select block in displayed program

SET BREAKPOINT: mark selected block, which will be displayed above

	BLOCK		JUMP TARGET	

BLOCK: a block in the displayed program can be selected directly

JUMP TARGET: jump targets ($) in the displayed program can be selected

After selection of block or a jump target the previous SK line will appear once more. The breakpoint should then be set.

TABLES

DRY RUN RAPID	STEP	SELECT BREAKPOINT	BREAKPOINT	**TABLE**

TOOLS	ZERO SHIFTS	VARIABLES		

Zero shifts and variables can be checked, tools can be checked and edited.

TOOLS

TOOLS	ZERO SHIFTS	VARIABLES	↓	

Tool data appears in the edit line.

TOOL NUMBER	↓	SCROLL		

Tool data can be selected directly via their number (+ ENTER) or by cursor control. The cursor is positioned on the DR value (wear). The wear value compensation value can now be updated by an incremental input. Conclude with ENTER (see p. 4 - 1).

ZERO SHIFTS

TOOLS	**ZERO SHIFTS**	VARIABLES		

Zero shift data appears in the edit line.

ZERO SHIFT NUMBER	↓	SCROLL	↑	

Direct selection via number (+ ENTER) or by cursor control (+ SCROLL).

VARIABLES

TOOLS	ZERO SHIFTS	**VARIABLES**		

VARIABLE NUMBER	↓	SCROLL	↑	

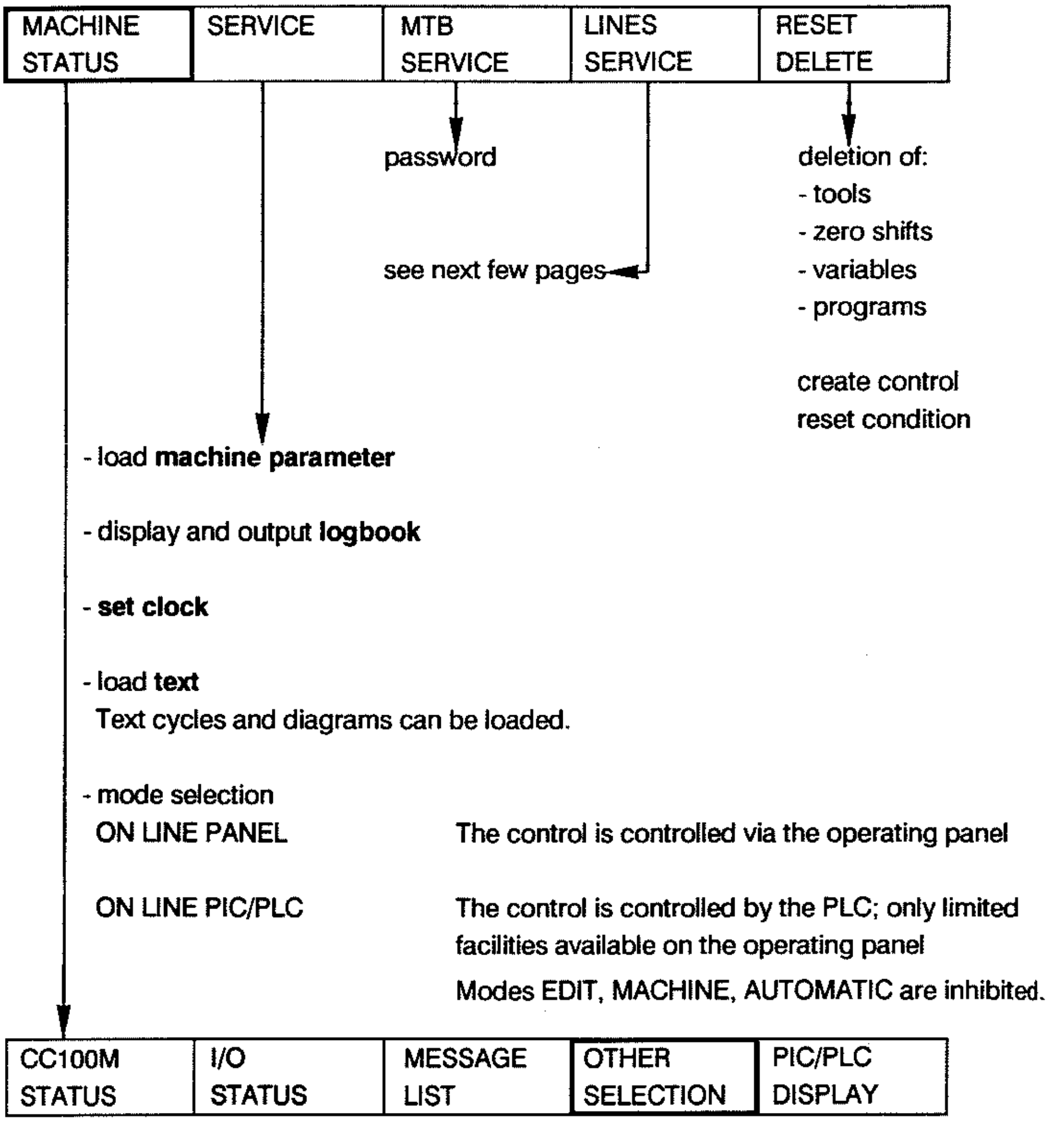
Operating and function as for zero shifts.

INFO

The **INFO** mode is subdivided into two separate sections:

- the machine tool builder section, protected by the MTB code
- the user section.

Within the user section additional information is made available to the operator.



MACHINE STATUS	SERVICE	MTB SERVICE	LINES SERVICE	RESET DELETE

password

see next few pages

deletion of:
- tools
- zero shifts
- variables
- programs

create control reset condition

- load **machine parameter**
- display and output **logbook**
- **set clock**
- load **text**
  Text cycles and diagrams can be loaded.
- mode selection

ON LINE PANEL	The control is controlled via the operating panel
ON LINE PIC/PLC	The control is controlled by the PLC; only limited facilities available on the operating panel Modes EDIT, MACHINE, AUTOMATIC are inhibited.

CC100M STATUS	I/O STATUS	MESSAGE LIST	OTHER SELECTION	PIC/PLC DISPLAY

CC 100M STATUS	- Display of the set modal functions, potentimeters, zero shifts, scale factors, SW limit switches
I/O STATUS	- Status of the CNC-PIC interface
MESSAGE LIST	- Display of the last 10 error texts with error number and error location (program, block)

selection

CC 100M STATUS	EXTERNAL STATUS	MESSAGE LIST	AXES DISPLAY	PIC/PLC DISPLAY

TABLE	LIST			

TABLE	LIST	PAGE +	PAGE -	

Display of machine status conditions, defined by MTB.

(Seperate DNC description in preparation)

AXES DISPLAY

THE FOLLOWING SOFT KEYS APPEAR:

COMMAND POSITION	LAG	MACHINE POSITION	DISTANCE TO GO	INCH METRIC

COMMAND POSITION	- The programmed position is displayed.
LAG	- The lag, (also called following error), is displayed.
MACHINE POSITION	- The actual position is displayed as long as there are neither zero shifts nor G92 active. The MACHINE POSITION results from the COMMAND POSITION minus the lag.
DISTANCE TO GO	- The difference between the programmed command position and the actual position, i.e.the distance to go, is displayed.
INCH METRIC	- The default setting is metric. The dimensioning system selected with this soft key determines the display in the other main modes; a change-over is however also possible in these modes.

PIC/PLC DISPLAY

The PIC program is displayed and the following soft keys are offered:

SEARCH	↑	↑	TABLES	TRIGGER

SEARCH

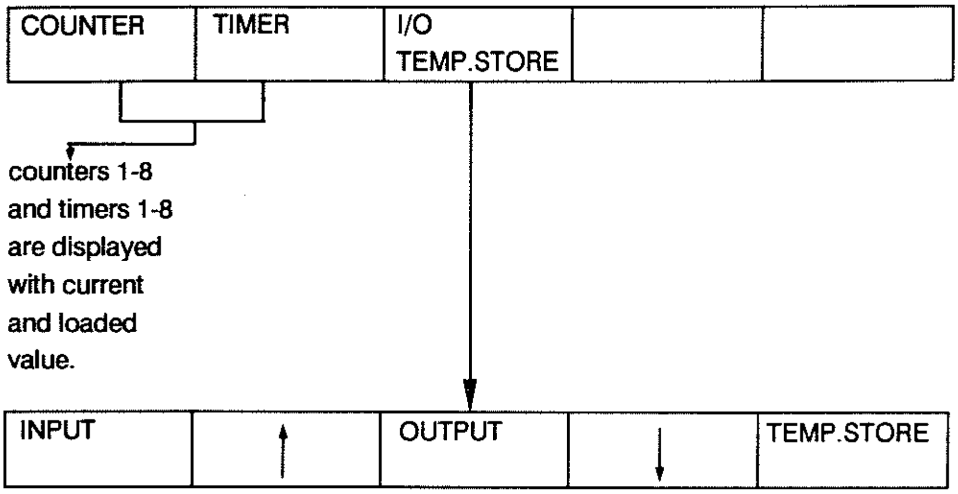
With this soft key
- addresses
- instructions (command + operator)
- commands (CMD)
- operators

can be searched for and displayed, entered either with the full number or part of the number or without the number.

If a string is not found the message STRING NOT FOUND appears in the edit line. If an instruction, a command etc. is not found the NC gives the message NOT FOUND in the edit line.

SOFT KEYS ↑ ↓ - The program display can be scrolled up and down line by line (no repeat function)

TABLES - makes the following soft keys available:



soft keys INPUT OUTPUT TEMP.STORE	- These soft keys are used to select the corresponding data or clear them from the screen. Selected data is marked by highlighting of the corresponding soft key. Data used in the NC-PLC interface are highlighted in the display. Several or all sets of data can be selected simultaneosly.
soft keys ↑ ↓	- The selected data displayed on the screen can be scrolled up or down line by line (no repeat function).
TRIGGER	- makes the following soft keys available:

LOW	_┌	HIGH	┐_	TRIGGER OFF

soft keys LOW, HIGH	- The trigger function responds to a low signal or a high signal.
soft keys _┌ ┐_	- The trigger function responds to a rising or falling edge.

If one of these soft keys is actuated the following soft keys appear:

SEARCH	↑	↓	TABLES	TRIGGER OFF

The selected trigger condition is displayed in the highlighted line at the top of the screen.

The highlighted line at the top of the screen contains the following information:

STATUS SIGNAL TYPE INSTRUCTION ADDRESS

STATUS - waiting for
(signal has not occurred yet)

- triggered
(signal has occurred)

SIGNAL TYPE as selected by soft key

- low level
- high level
- rising edge
- falling edge

INSTRUCTION - instruction marked by the cursor in the displayed program

ADDRESS - address of the displayed instruction

While the trigger function is switched on it is possible to page through the program. Soft key TRIGGER OFF switches the trigger function off. The purpose of the trigger function is the monitoring of signals which occur intermittently; it is an important aid for fault finding.

LINES SERVICE SOFTKEY LINE FOR DNC OPERATION

Lines service

PORT SET UP		DNC RESET	STATUS MASK	DNC ON OFF

(separate DNC description in preparation)

DIMENSIONING - SWITCHING BETWEEN INCH/METRIC

MEMORY mode

TOOLS	ZERO SHIFTS	VARIABLES	PROGRAMS	CYCLES

ACCESS ON/OFF		EDIT		SAVE

ACCESS ON/OFF	INCH METRIC	EDIT	LOAD	SAVE

TOOLS	ZERO	VARIABLES	PROGRAMS	CYCLES

	NEXT PAGE		LOAD	SAVE

e.g. 1 ENTER

COMMAND	NEXT PAGE	EDIT	LOAD	SAVE

		INCH METRIC		RENAME

VARIABLES can not be switched to INCH/METRIC. Whether the file types, tools and zero shifts are to be effective in metric or inch is determined by soft key.

Effect: The file types program and cycles are stored with the dimensioning index I/M. Metric is preset for new files.

MACHINE mode

In main mode MACHINE the INCH/METRIC switching is effected in the first soft key line:

REFERENCE AXES	REFERENCE CYCLE	MDI	TEACH IN	INCH METRIC

Effect: The selection is effective for all functions in MACHINE mode. The selection is retained even after a hardware reset and it also applies after a switch into INFO mode.

AUTOMATIC mode

File types such as programs and cycles are already defined with respect to the dimensioning during the generation process. The chosen dimensioning method also applies for the execution.

INFO mode

The axis measurement format (INCH/METRIC) selected in INFO mode sets the priority for the axis display in machine mode.

MACHINE STATUS	SERVICE	MTB SERVICE	LINES SERVICE	RESET DELETE

CC 100M STATUS	I/O STATUS	MESSAGE LIST	OTHER SELECTION	PIC/PLC DISPLAY

CC 100M STATUS	EXTERNAL STATUS	MESSAGE LIST	AXES DISPLAY	PIC/PLC DISPLAY

COMMAND POSITION	LAG	MACHINE POSITION	DISTANCE DISPLAY	INCH METRIC

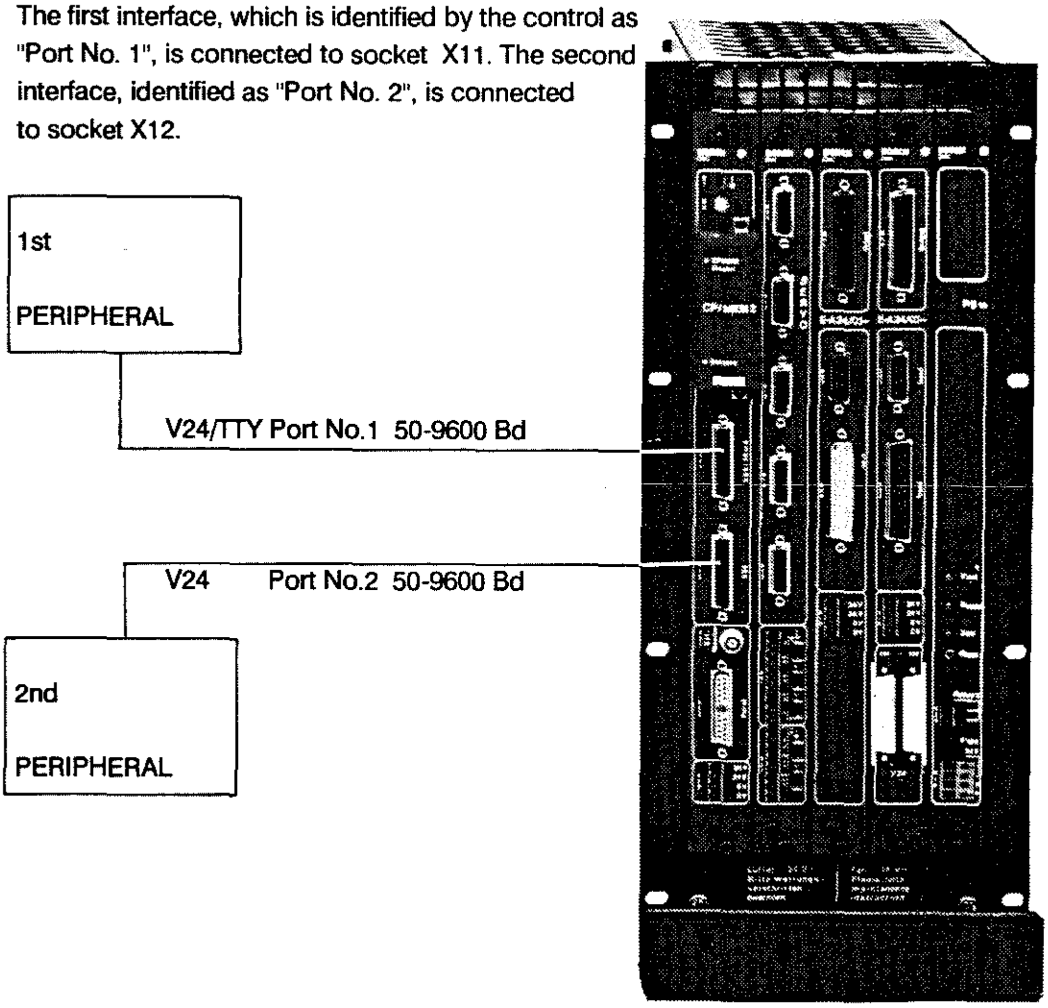
- The desired dimensioning method is selected for the particular axis display (command/position, machine position, lag, distance to go).
- On switch-on the dimensioning method last active is reactivated.

DATA HANDLING

GENERAL

LOAD / SAVE

The CC100M has two serial data interfaces, the sockets of which are located on the CP/MEM board.
The first interface, which is identified by the control as "Port No. 1", is connected to socket X11. The second interface, identified as "Port No. 2", is connected to socket X12.



Input and output of data is possible in main modes INFO and EDIT. Interface selection and parameterisation are made via soft keys.

In main mode "EDIT" the following types of data can be loaded and saved:

(soft keys:)

TOOLS	ZERO SHIFTS	VARIABLES	PROGRAMS	CYCLES

In "INFO" mode it is possible load machine parameters, M-functions, texts and graphics.
Programs, tools, zero shifts and variables can only be cleared.

LOAD

Operating procedure:

- Select main mode EDIT

- Actuate soft keys as shown below:

TOOLS	ZERO SHIFTS	VARIABLES	PROGRAMS	CYCLES

	NEXT PAGE		LOAD	SAVE

- Optional: Key in program number or name and press "ENTER".

COMMAND	NEXT PAGE	EDIT	LOAD	SAVE

ALL FILES YES NO	START	PORT NO	BAUDRATE	CONTROL YES NO

- Soft key "ALL FILES"

"YES" selected: All files on the data carrier are loaded.

"NO" selected: Only the specified number of successive files (number is requested) are loaded.

- Soft key "START": The loading operation is started; the control waits for data. After the initial actuation the soft key changes to "STOP" and can be used to stop the data transfer.

- Soft key "PORT NO": Enter port number 1 or 2. The corresponding interface (X11 or X12) will be activated.

- Soft key "BAUDRATE": Set baudrate. A list of the code numbers for the baudrates appears on the screen. The baudrate set on the control must be the same as the one set on the peripheral.

- Soft key "CONTROL YES/NO":
With CONTROL YES the syntax is checked.

With CONTROL NO only the checksum is checked, if it exists.

If the program or cycle does not contain checksums the control will carry out a syntax check.

Note

Under SK "PROGRAMS" it is also possible to load cycles, tool compensations, zero shifts and variables; the same applies for SK "CYCLES". Cycles are loaded in succession, like the programs. When the last program or cycle has been loaded the load operation is stopped. If there are tool, zero shift and variable files on the data carrier loading is stopped after each file, if an EOT signal separates the files. If the subsequent files are to be loaded too SK "START" must be actuated for each one.

Protection

When loading data via serial interfaces programs are automatically protected against overwriting. If a program is loaded which is already stored in the memory the control will ask whether to

- overwrite the existing program	(input 1)
- store the program under a new number	(input 2)
- abort the loading operation	(SK "STOP")

A program with overwrite protection can not be overwritten.
Error message: "file protected".

SAVE

Operating procedure:

- Select main mode EDIT

- Actuate soft keys as shown below:

TOOLS	ZERO SHIFTS	VARIABLES	PROGRAMS	CYCLES

	NEXT PAGE	LOAD		SAVE

- Optional: Key in program or cycle name or number and actuate "ENTER".

ANOTHER SELECTION	START	PORT NO	BAUDRATE	CHECKSUM YES/NO

The screen displays the message "SELECTED FILE ONLY" (highlighted characters)

- Soft keys "START", "PORT NO" and "BAUDRATE" are operated as for loading.

- Soft key "CHECKSUM" switches the generating of a checksum, which is to be output, on and off.

PROGRAMS + CYCLES	START	FILE + TOOLS	FILE + ZEROSHIFTS	FILE + VARIABLES

- Soft key "PROGRAMS + CYCLES" determines whether only either programs or cycles are to be saved, depending on the selection in the first SK line, or whether programs and cycles are to be output. (Display with highlighted characters.)
  The page back button resets the display to "SELECTED FILE ONLY".

- Soft keys "FILE + TOOLS", "FILE + ZEROSHIFTS" and "FILE + VARIABLES". When one of these is selected the corresponding term will be displayed in highlighted characters.

ote If one of these soft keys is selected the parameters "from" and "to" must be defined. Unless this is done no page back or other selection is possible.

The parameter ranges are as follows:

tools	1 - 48; input e.g.: 1,7,14,15,16, 23, 44
zero shifts	54 - 59; input e.g.: 54, 57, 58
variables	1 - 99, A - Z; input e.g. 7, 9,10, 25, 49,A,C,L,X

Only the numbers should be entered, not the associated letter codes. The sequence for the variables is numbers first, then letters.

Output without file selection:

TOOLS	ZERO SHIFTS	VARIABLES	PROGRAMS	CYCLES

	NEXT PAGE		LOAD	SAVE

PROGRAMS + CYCLES	START	PORT NO	BAUDRATE	FORMAT

		CHECKSUM YES NO	FORMAT DFS CC100	

Soft key "PROGRAMS + CYCLES" offers the choice of outputting programs or cycles. Either programs or cycles are preselected, depending on the choice made in the first soft key line.

omment During the output of programs and cycles the selection of the dimensioning unit "INCH" or "METRIC" is output in the program header.

elete Main mode INFO

Operating procedure:

- Activate main mode "INFO"
- Continue with soft key operation

MACHINE STATUS	SERVICE	MTB SERVICE	LINES SERVICE	RESET + DELETE

DELETE TOOLS	DELETE ZEROSHIFTS	DELETE VARIABLES	DELETE PROGRAMS	CONTROL RESET

The selected soft key is highlighted on the display.
The delete operation can be aborted with the page back button.

Caution When the "ENTER" key is pressed all programs will be deleted, even those with write protection.

Delete function in main mode EDIT

In main mode "EDIT" programs are deleted individually (or cycles, depending on the soft key selection), and only those without read/write protection can be deleted in this mode.

Operating procedure:

- Select main mode "EDIT"

- Continue with soft key operation:

TOOLS	ZERO SHIFTS	VARIABLES	PROGRAMS	CYCLES

	NEXT PAGE		LOAD	SAVE

Select program or cycle by name or number.

COMMAND	NEXT PAGE	EDIT	LOAD	SAVE

COMMAND	PROTECTION ON OFF	INCH	DELETE	RENAME

Note If an attempt is made to delete a program or cycle with read/write protection the message "file protected" will appear on the screen.

In "EDIT" mode it is not possible to delete tool data, variable data and zero shifts.

TOOLS, ZERO SHIFTS, VARIABLES

These types of data can be loaded and saved in "EDIT" mode; they can only be deleted in "INFO" mode.

Load Main mode EDIT ◇

Soft keys:

TOOLS	ZEROSHIFTS	VARIABLE	PROGRAMS	CYCLES

ACCESS ON/OFF		EDIT		SAVE

ACCESS ON/OFF	INCH METRIC	EDIT	LOAD	SAVE

The soft key "INCH/METRIC" does not appear for variables.

	START	PORT NO	BAUDRATE	

Note Data can also be loaded under "PROGRAMS" or "CYCLES"
Write protection is then not effective. The selection of inch or metric made in this way is not stored on the data carrier and must be made at the control.

Save

TOOLS	ZEROSHIFTS	VARIABLES	PROGRAMS	CYCLES

ACCESS ON/OFF		EDIT		SAVE

	START	PORT NO	BAUDRATE	FORMAT

		CHECKSUM YES NO	FORMAT DFS CC100	

The delete function in "INFO" mode works as described in chapter "Load and save programs and cycles".

Machine Parameters, Text Strings and Graphics

In "INFO" mode these types of data can only be loaded.

Operating procedure:

- Select "INFO" mode with key.
- Continue with soft key operation:

	SERVICE	MTB SERVICE	LINES SERVICE	RESET DELETE

LOAD MACH. PARAMETER	LOGBOOK	SET CLOCK	MODE	LOAD TEXT

	START	PORT NO	BAUDRATE	

Note

During the loading operation the data previously in the memory is overwritten. Enter only the appropriate data under the selected type of data, i.e. do not select soft key "LOAD TEXT" if you have previously selected LOAD MACHINE PARAMETERS.

Logbook If a logbook exists the data can be output in "INFO" mode.

Operating procedure:

- Select main mode "INFO" with key.

- Continue with soft key operation:

MACHINE STATUS	SERVICE	MTB SERVICE	LINES SERVICE	RESET DELETE

LOAD MACH. PARAMETER	LOGBOOK	SET CLOCK	MODE	LOAD TEXT

ACTIVATE LOGBOOK	LOGBOOK DISPLAY	CLEAR LOGBOOK	SAVE LOGBOOK	

	START	PORT NO	BAUDRATE	

Note If no logbook has been generated the soft key "LOGBOOK DISPLAY" will not be displayed.

3. PROGRAMMING

GENERAL

Program Production

Part programs can be produced by the following methods:

- directly at the control via	panel input in modes EDIT or MACHINE (TEACH IN) or
- at programming stations	For transmissions please note the instructions in sections: DATA INTERFACES (chapter 1) Data handling (chapter 2)

Memory Allocation

The following types of user data are stored in the control:

Memory areas	Contents
part program memory	part programs and cycles, with the relevant subprograms
technology table	tool geometry and tool wear data, cutting speeds
zero shift table	zero shifts G54 to G59
variable table	CPC varables VI-V99 and VA-VZ
machine parameter memory	machine specific data

Basic Conditions

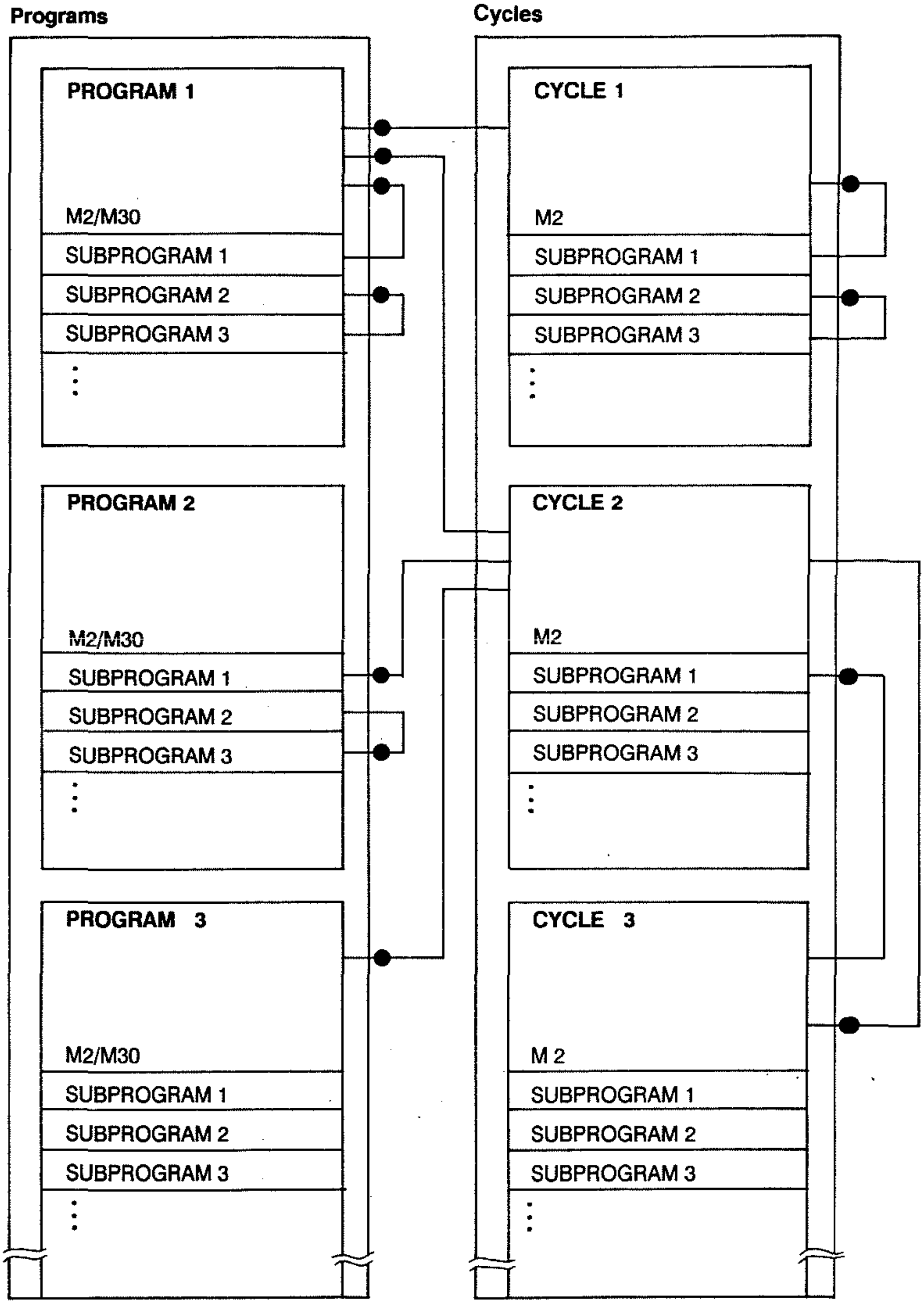
Descriptions in the programming instructions relate to the control as used on a machine tool (milling machine) with a Cartesian axis configuration within a clockwise coordinate system. Unless otherwise stated the following G-functions are assumed to be active:

G17	plane XY
G27	no field limitation
G40/T00	no tool compensation
G53	no zero shift active
G62	in position function off
G90	absolute dimensions

The reset status or the status after switching to automatic mode is indicated by 'A'.

Memory Allocation

Possibilities of Program Construction



Explanations

Up to 99 suprograms can be assigned to a program or cycle.

Main programs and their subprograms can call up cycles.
From within cycles and their suprograms other cycles and subprograms can be called up, up to a 10-fold total nesting depth.
●call-up source

PART PROGRAMS AND CYCLES

A program or a cycle describes a sequence of machining operations and is subdivided into blocks. The blocks contain preparatory functions, axis information, miscellaneous and auxiliary functions.

Block A block is made up of the block number and one or several words.

Example: N120 G0 X100 Z100 M3 S1000

block number words

The block length is variable. During external programming the words can be written in any order. The block number must be at the beginning of the block. No space characters required between blocks. But note the gap between the block number and the first word (see transmission protocol, p. 3-4).

Word A word consists of an address letter and a sequence of figures, which represent the address contents.

Only those figures which contain information need be written.

N10	G0	X5.100	Z0.500	M3	T01	or
N10	G0	X5.1	Z.5	M3	T01	

Blocks are built up from individual words which begin with an address letter.

address

Example: N 120 X 125 Z 160

address content

With DIN programming an address may only be programmend once in each block.

Block Numbers The first word of a program block is the block number. It is made up of the address letter "N" (ISO format) and a 1to 4-digit sequence of figures.

- sequence During **external** program production no block numbers need to be programmed. The control will store data in ascending order.

During **panel input** the control generates the block numbers automatically in the course of the input dialogue.

- steps Block numbers are programmed or generated in steps of 1. If additional blocks are entered via "INSERT" the control will mark these blocks with a " + ". The jump addresses remain valid after insertions or deletions sine they are marked with symbolic "labels".

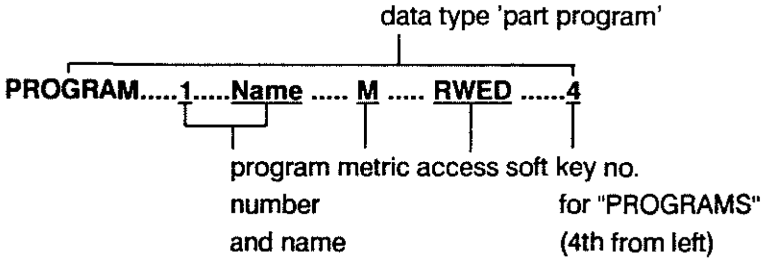
The control can store 1 or several user programs. During the programming these programs can be marked as main programs, or subprograms (SBP), or cycles.

Program

A program is defined by the

- HEADER	in the first line and
- PROGRAM END	instruction in the last line.

Header



The header line is generated automatically by the control after call-up of the program or input of the program name.

Program end

M2 program end
M30 program end - renewed execution with CYCLE START

Subprogram

Subprograms are of local character; i.e. they are always assigned to a specific program. Subprogram numbers may be used repeatedly as long as they are assigend to different programs.

A subprogram (SBP) is defined by

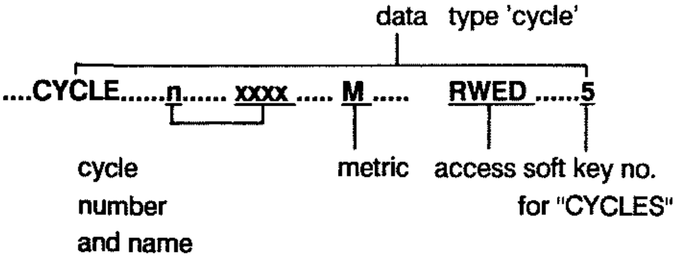
$..	up to 2-digit subprogram number in first line
G99	subprogram end in the last line

The subprogram and the main program are stored in the same file.

Cycle

Cycles are of global character.
In other words: Each cycle number may only be used once in the program memory, but can be called up from each program/subprogram or with a direct call-up.

Header



Cycle end

M2 cycle end
During panel input the headers are generated by soft key selection.

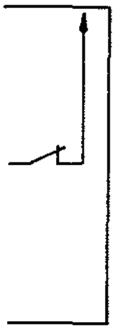
Jump Instructions

Program jumps can be used for a more efficient usage of the individual program segments.
The jump instructions relate to jump addresses (labels) which are to be previously defined. These symbolic addresses are retained even when program alterations are carried out by inserting or deleting blocks.

Programming of
G24 P x x (unconditional jump)
or
G23 P x x (conditional jump)
P = jump address number
effects branching to a program line which is marked as a jump address:
$ x x.

Example

```
N10
N11  $    2
N12  X...... Y...
N13

N20   G23   P2
N21  N...
N22  Y...
.
N24  X...
N25  G24  P2
```



sequence if
signal **OPITIONAL JUMP** = high:

N10 to N20 / N11 to N20
sequence if
signal **OPTIONAL JUMP** = low:

N10 to N25 / N11 to N25.

SBP Call-ups

The calling up of subprograms must only be possible by programming

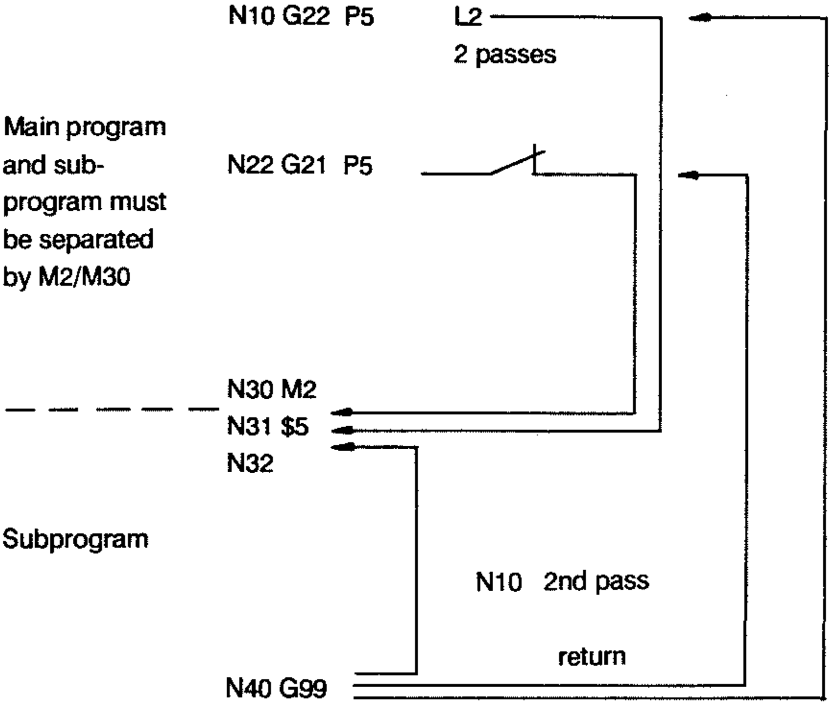
G22 P .. L.. unconditional SBP call-up or
G21 P... L.. conditional SBP call-up

P = SBP number 1 to 99
L = repetition 0 99

For this reason the subprogram call-ups G21/G22 in the main program must be separated from the subprograms themeselves by M2/M30.

One SBP can be called up repeatedly and from different places within the relevant main program.

Example



Main program and sub-program must be separated by M2/M30

Subprogram

Decisions

Subprogram calls or jumps can be linked to a condition, which can be

- the logic state of interface contacts or
- the result of a mathematical comparison (parametric functions)

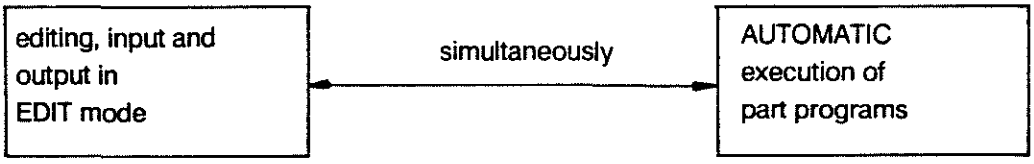
The jumps or calls are carried out if the stated condition is fulfilled. They are not carried out (and the program is continued at the next line) if the condition is not fulfilled.

PARALLEL PROGRAMMING

Definition

Parallel programming allows the control to be used in EDIT mode while an active program is being executed.
In edit mode tool data, zero shift tables, variables and part programs can be entered, edited and output.
Active programs and cycles can not be edited in parallel operation.

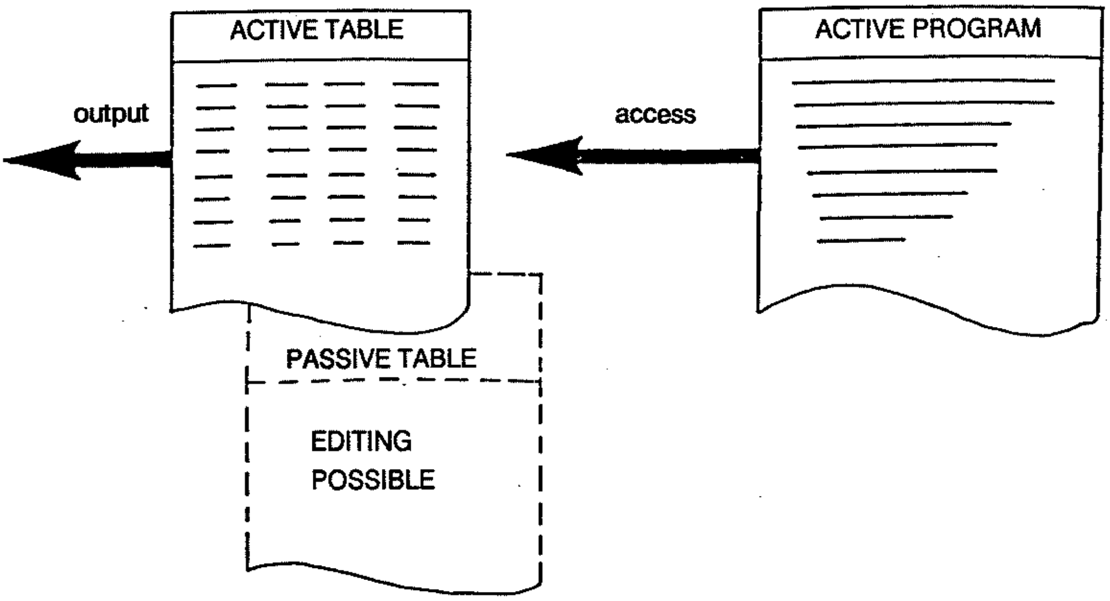
PARALLEL PROGRAMMING



Functions available in Parallel Operation

Tables

TOOL, ZERO SHIFT and VARIABLE tables can be edited, entered and output. Contents of tables which need to be accessed by the active program can not be edited in parallel operation.
A possibility does, however, exist to edit table contents during program execution. After the program has been completed the existing table is overwritten with the modifications (updated). The control generates a passive table for this purpose.



CYCLES Cycles can not be edited in parallel operation. But they can be input and output via the serial interface.

PROGRAMS Programs can be edited in parallel operation and can also be input from and output to external data carriers.

The **active** program can **not** be edited.
There is, however, the possibility of copying the active program in the memory before starting program execution.
The copied program can then be edited.

Soft key TABLES During AUTOMATIC execution of a program while in parallel operation the soft key TABLES appears.
Under this soft key it is possible to look at the tables TOOLS, ZERO SHIFTS and VARIABLES without having to come out of main mode AUTOMATIC.

DRIP FEEDING

DEFINITION Long programs which do not fit into the program memory can be loaded via interface for direct execution.

DRIP FEEDING - SINGLE ACTIVATION

Single drip feeding operation is activated via soft key

DRIP
FEEDING

IN AUTOMATIC mode (direct selection).

DRIP FEEDING - CONTINUOUSLY ACTIVE

If drip feeding is to be activated automatically when AUTOMATIC mode is selected the operator must switch to

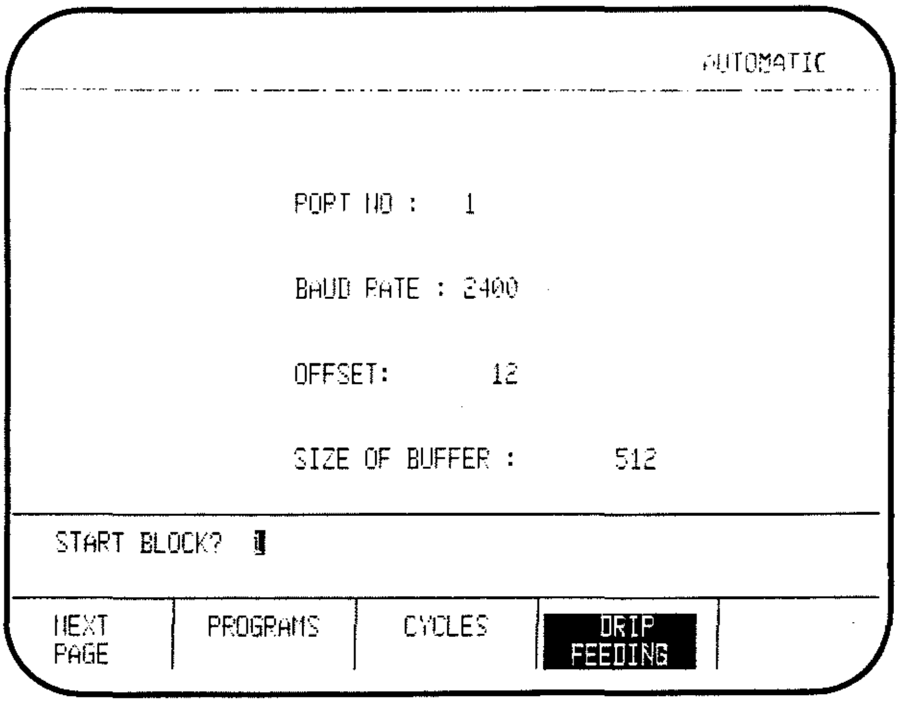
ACTIVE ON
POWER ON

(reverse video) in the 3rd soft key level (INFO mode).

DRIP FEEDING - USER INTERFACE

In AUTOMATIC mode the preset parameters for DRIP FEEDING will appear on the display once it is activated.

Example



```
                                        AUTOMATIC

              PORT NO :   1

              BAUD RATE : 2400

              OFFSET:       12

              SIZE OF BUFFER :       512

 START BLOCK?
```

The DRIP FEEDING parameters are preset in INFO mode.
The parameters do not affect the program which is to be executed.

Main mode INFO

MACHINE STATUS	SERVICE	MTB SERVICE	LINES SERVICE	RESET DELETE

	DRIP FEEDING		DNC	

ACTIVE ON POWER ON	BUFFER SIZE	PORT NO.	BAUDRATE	BLOCK OFFSET

Meaning of the DRIP FEEDING parameters

ACTIVE ON POWER ON

If this parameter is active (reverse video) the control
defaults to DRIP FEEDING mode when AUTOMATIC is selected.

BUFFER SIZE

The BUFFER SIZE parameter determines the buffer size in 0.5 kBytes, which is to be kept free for DRIP FEEDING in the part program memory of the control.
Input format: 512 bytes
Min. buffer size: 1 (= 512 bytes)
Max. buffer size: ≤max. available memory capacity (see Drip Feeding and main memory)

PORT NO.

Selection of the interface on the CP/MEM

Port 1 - V.24/20 mA	(with handshake)
Port 2 - V.24	(with or without handshake)

BAUDRATE

Setting of the baudrate.
The following baudrates are recommended (- 1800 Bd):

8 = 1800Bd
9 = 2000Bd
10 = 2400Bd
11 = 3600Bd
12 = 4800Bd
13 = 7200Bd
14 = 9600Bd

BLOCK OFFSET

This parameter is originally preset so that the program execution begins after 12 program blocks have been loaded (min.).

The setting "n" determines after how many loaded blocks the execution is to begin.

Possibilities:		
	n = -1	execution begins when the buffer is full or when M30/M2 is transferred from the DRIP FEEDING program.
	n = 0	Execution begins when 12 program blocks are loaded.
	n > 12	Execution begins when the specified number (n) of program blocks are loaded.

START POINT? Input of the block number at which DRIP FEEDING is to start (1 = beginning of the program).
NC blocks before the start point are ignored.

Note: The DRIP FEEDING parameters can only be changed in INFO mode. Port no. and baudrate are independent of the parameters as described in chapter "Data Handling".

DRIP FEEDING AND MAIN MEMORY

Part programs and cycles occupy a certain area in the part program memory; the remaining available storage capacity is used for DRIP FEEDING.

When the buffer size for DRIP FEEDING has been determined in INFO mode and DRIP FEEDING is activated in AUTOMATIC mode the control checks whether the selected buffer size does not exceed the available storage capacity. If it does an error message will be produced. If the buffer size is not defined the user can utilize the max. available storage capacity.

Input: available storage capacity (see basis display
512 in AUTOMATIC)

If the available storage capacity is not sufficient there are two possibilities:

- deletion of individual programs or cycles to increase the available storage capacity

- reduce the buffer size in INFO mode

PROGRAM EXECUTION WITH DRIP FEEDING AFTER CYCLE START

The DRIP FEEDING operation is started with Cycle Start.

During program execution only the active block is displayed on the screen.

DRY RUN RAPID	STEP	LIST		TABLE

By actuating SK LIST the 6 blocks following the active block can be listed.

Program execution is possible with the following options:

- step size in program

- rapid / dry run of the program

- starting the program at a set start point (block N)

Recommendations for achieving fast data input with drip feeding

- When the control has "some time" (e.g. long traversing path, G4 active, or FEED HOLD active) it loads data into the buffer. It is therefore advantageous to choose the buffer to be as large as possible. The control is then able to "live" on data from the buffer for those program parts where the block cycle time is critical. In this case the loading of new blocks is inhibited until only the minimum number of blocks are in the buffer. The block cycle time will then be the same as when working from memory.

- Drip feeding and checksum:

  Drip feeding programs should be transferred to the control with checksum in order to increase the speed of the transmission. Also the baudrate should not be below 1800.

Position and calculation of the checksum

(see program header in DFS format, page 1-22

Restrictions

- Jumps, subprograms and the setting of stop points are not permitted in DRIP FEEDING programs;

- Parallel programming is not possible since there are several functions active simultaneously during DRIP FEEDING:

  - automatic program execution

  - block processing

  - transfer function from external data carrier (LOAD, SAVE)

- The REENTRY function is not possible.

ADDRESSES

ADDRESS F

G1 F. . G2 F. . G3 F. . G5 F. .	F defines the path feedrate in mm/min.
G04 F. .	F takes effect as dwell in seconds.
G93 F. .	F takes effect as execution time for the programmed path section in seconds.
G94 F. .	F takes effect as feedrate in mm/min. G94 is active on switch-on. Programmable range: F0.001 to F 50 000
G95 F. .	F takes effect as feedrate in mm/rev. The programmed path feedrate is derived from the actual speed of the main spindle. G95 is used for tapping and finishing.

CONTROL RESET clears any programmed F-address.

ADDRESS T

T determines the tool number, which is to be output, and/or the tool length compensation, which is to be applied internally. T is programmed with 2 or 4 digits.

```
T .. ..
  |  L——tool number to be output
  |     to the PIC
  |     programmable range: 0 to 99
  |
  L—————compensation group of the
        technology memory to be applied
        programmable range: 0 up to max. 48
```

If T is programmed with only 2 digits these are always interpreted as the compensation group.

The operation of the tool length and tool radius compensation is described in detail under TOOL COMPENSATION, chapter 5.

ADDRESS M

Definition Output signals can be generated by means of the program.

Range of M-Functions The control itself allows all M-codes from M0 to M99 to be used. The user can utilize all M-functions which have a machine function assigned to them.

Internal Functions Listed below are a number of codes which have fixed internal functions:

Code	**Internal function**	
M0	Program stop after execution of the block. All other conditions unchanged; does not cause spindle stop. New start with next block number via CYCLE START.	
M2	Main program end, cycle end, programmed separately system then switches into program selection level **Irrespective** of the start point selection a new program will start at the beginning.	
M3	Spindle rotation, clockwise. A direction of rotation must be active when spindle speeds or gear ranges are programmed.	machine specific effect
M4	Spindle rotation, counter-clockwise, otherwise as M3.	
M5	Spindle stop, programmed separately, spindle speed and gear range remain stored internally.	
M6	Call-up of automatic tool change cycle (cycle 77)	
M13	Spindle CW coolant on	
M14	Spindle CCW coolant on	
M19	Orientation of main spindle to fixed position in degrees. M19 S . . . .: positioning to programmable position. M19 is output at the interface; address S is not	
M21	Call-up of MTB cycle 76. No output at interface.	
M22	Call-up of MTB cycle 75. No output at interface.	
M30	Program end. Mode of operation and other conditions are retained. Change of mode after reset. **Dependent** on the start point selection a restarted program after M30 will be executed from the selected start point onwards.	
M40	Automatic gear range selection (Active on switch-on, machine specific operation).	
M41–M44	Selection of fixed gear ranges 1 to 4 (machine specific operation).	
M98	SINGLE BLOCK command is not allowed for as long as M98 is active. Programmed in a block of its own.	
M99	SINGLE BLOCK is possible, i.e. M98 is cancelled. M99 is active on switch-on. Programmed in a block of its own	

External effects and further M-functions are particular to each machine and details must be provided by the machine tool builder; for instance: coolant on/off, delivery and removal of workpieces.

ADDRESS S

Definition Programmed on its own the S-address determines the spindle speed, or the position for spindle orientation.

G92 S . . When programmed in conjunction with G92 the S-address limits the maximum speed of the main spindle.

M19 S . . The spindle is oriented onto the position programmed with S (degrees). If M 19 is programmed on its own the value defined by machine parameter 111 will apply as orientation point (range 00 - 359.999°).

S . . . . . . Spindle speed in rpm. The direction of rotation (M3/M4) must have been defined.

SPINDLE SPEEDS

Definition Inputs are evaluated as follows:

With G 97 S = spindle speed directly in rpm format 4.3

The direction of rotation must be determined together with the programming of S or beforehand.
Minimum and maximum speeds are predetermined for the particular machine (M-parameters).

ADDRESS M GEAR RANGES

Machines with a gearbox which can be controlled via the CNC can operate in two ways:

Fixed Selection M41-44 One particular gear range is programmed in the user program with M41 to 44, corresponding to gear ranges 1 to 4:
The control assists with the change-over between gear ranges by the output of idling speeds, by the processing of signals relating to the gear ranges etc.
If a speed is programmed which is not achievable within the selected gear range, the control outputs the max. or min. speed possible within that range.

Automatic Selection M40 When M40 is active the control itself selects the appropriate gear range on the basis of the following criteria:

- up to 4 gear ranges with min. and max. speed values can be controlled
- output range for the speed:
  1 to 9999 rpm (MTB can restrict the range for the particular machine)
- when S is programmed the appropriate gear range is automatically selected, on the basis of the current program data
- where gear ranges overlap the control selects the lower of any two possible gear ranges (higher motor speed).

G96 + M40 A new gear range is only selected for the following block if the required speed can not be achieved in the active gear range. Idling speed is output for as long as the activation of the correct gear range has not been acknowledged.

H-ADDRESS

Hxx
Hxxxx

Definition

H-address = "FLYING OUTPUT"
As opposed to the M-address, which is output before each traversing movement, the H-address is output simulaneously with the traversing movement.
This simultaneous output prevents drops in the command value.

Use

This function can used in programs for machining operations during which any momentary stopping of the axes would result in damage to the workpiece (for instance during laser cutting).
This 4-digit auxiliary function permits additional control and switching functions for time-critical applications.

Programming

- The H-address should be regarded as an additional auxiliary function; it should not be programmed with other auxiliary functions in the same block.

- The programming format is up to 4-digit.
- Variables can be allocated to the H-address (V1 = 1212; H = V1).

Output

The H-address is output to the interface in BCD code.
If the address has 4 digits the last two are output first.

Note

H-addresses can also be used for the extension of certain functions (e.g. speed programming in dual spindle operation: S1 = 1000 rpm; H = 500 rpm).

OPERATOR INSTRUCTION PROGRAMMING

Definition The operator instruction programming facility allows the display of texts during the program execution.
These texts can be purely informative or they can give instructions to the operator.
The contents of the texts do not affect the program sequence or machine functions in any way.

Programming The text is programmed in brackets and must be written in a separate block.

Usage This facility can be used to provide documentation for the program. Since the display always shows the next block to be execution while the program is being actioned it is possible to put message up on the screen by programming M0 beforehand.

If a program block is programmed in brackets, i.e. as an operator instruction, it will not be actioned. In this way blocks can be blanked out in a program.

Example:

```
.
.
.
N5
N6
N7  M0
N8  (NOTE - SWITCH ON COOLANT
N9  X . . .  Y . . .
N10 X . . .  Y . . .
.
.
.
```

The program sequence stops in block 7 (due to M0).
The operator instruction will then be displayed.

TABLES

Tools

Up to 48 tool compensation stores are available.
Each tool compensation store comprises the following:

T1	R = uuu.uuu	DR = u.uuu	L = uuu.uuu	S = uuu
no.	tool radius (mm or inch)	tool wear max.10% of radius, max. 1 mm, increm. input	tool length (mm or inch)	cutting speed m/min or inch/min

Input dimension defined as metric or inch via soft key.

Example:

```
*****TOOL**************************************1**************
 T1     R= 16.0        DR= 0.9             L= 25.0         S= 10.0
```

* = space character

Zero Shifts

6 zero shifts are available (G54-G59). See under section "G-Functions" G53,G54-G59 for definition.

Each zero shift comprises the following:

G54	X = uuuu.uuu	Y = uuuu.uuu	Z = uuuu.uuu	E = uuuu.uuu
no. of zero shift offset	amount of			

The dimension is defined as metric or inch via soft key.

Example:

```
**ZERO*SHIFT***********************************2**************
 G54    X= 10.0           Y= 20.0           Z= 30.0         E= 40.0
```

Variables

A maximum of 125 variables are available for the writing of variable programs (V1 . . .V99 and VA . . . VZ).
Variables represent numbers of up to 7 digits.

Example:

```
***VARIABLE************************************3**************
     V1 = 116.0     V2 = 8.0        V3 = 0.6       V4 =-1.0
```

Header line:

When programming tool data, zero shifts, variables, programs, cycles and (M) parameters externally, identifying HEADERS as shown above must be provided. These **must** be written in a specific format which is explained on page 1-17.

G-FUNCTIONS

LINEAR INTERPOLATION IN RAPID G0

Definition The axes travel to the programmed position with linear interpolation. The speed is determined by machine parameter.

Feedrate No feedrate should be programmed (address F). The machine parameter values for rapid will become effective.

Interaction This mode remains modal until a different mode of motion is selected. G0 cancels modes G1, 2, 3 and 5.

Execution of the next block is not started until all axes are "IN POSITION". THE IN POSITION range is defined by machine parameter.

Positioning with G0 is possible when the main spindle is stopped.

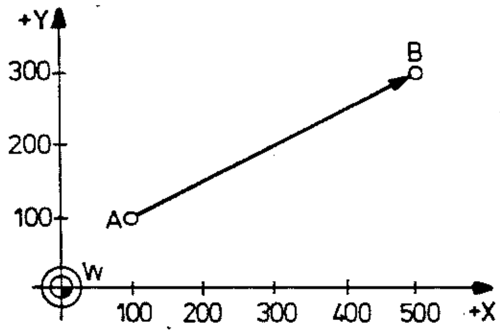
Programming **G0 X..... Y ..... Z...... E ......**
Programmable with or without axis addresses.

Path The traversing movement is linear even if the distances for the individual axes are different, or if the axes have different rapid speeds. The override potentiometer can be deactivated for G0 and AUTOMATIC by machine parameter.

Example

```
N1   G0
N2          X 100      Y 100  (starting position A)
N3    G0    X500       Y 300  (end position B)
N4    M30
```

Resulting movement with different distances in two axes:



Speeds The axis which has the longest distance to cover traverses at maximum speed. The speeds of other axes are regulated in such a way that all axes reach the programmed position simultaneously.

Note - G0 slope: Axis acceleration and deceleration during rapid traverse are controlled. by means of a command ramp. The constant acceleration parameters are programmed for the different axes via machine parameters (see Connections manual, Chapter 4).
This does not apply to the 4th axis if it is defined as a Hirth axis.

LINEAR INTERPOLATION IN FEED G1 A (A = active on switch-on)

Definition The axes traverse to the programmed point in a straight line at the active feedrate (F-word).
The movement is coordinated in such a way that all involved axes (up to 4 axes: X, Y, Z, E) reach the programmed point simultaneously.

Feedrate The programmed feedrate value (F) takes effect as the path feedrate; this means that if several axes are involved in the movement the portion of each individual axis is smaller than F.

The speed can be influenced via the feedrate override potentiometer.
If X, Y, Z and a rotary axis (E) are to traverse together, an anqular velocity is calculated for E. It is therefore advisable to use time programming G93 for movements involving both linear and rotary axes (see G2, G3, G5).

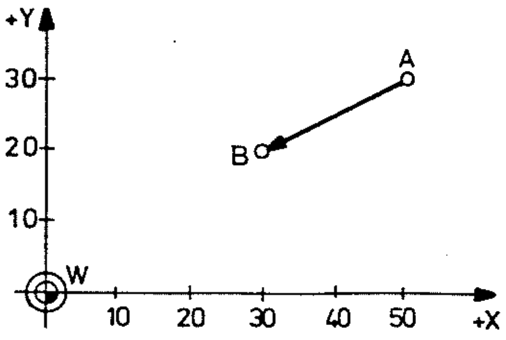
Interactions G1 cancels G0, 2, 3, 5 and is modal, as is the programmed feedrate (main address F).

Programming **G1 X.... Y.... Z.... E.... (F....)**

G1 can be programmed with or without axis information.
It must be programmed together with an F-word if no F-word is active yet.
Once a feedrate is programmed it remains effective until it is overwritten by a new value.
(Servo Error or switching off cancels the modal feedrate).
The programming of "F0" is not admissible.

Example

```
N1   G1   X50        Y30   F1000  (feedrate 1000 mm/min)
N2        X30        Y20
N3   M30
```



CIRCULAR INTERPOLATION G2, G3, G5

Definition

The axes traverse to the programmed point at the active feedrate on a circular or helical path.
The movement is coordinated in such a way that all involved axes reach the programmed point simultaneously.
Circles can only lie in parallel with one of the planes generated by two of the coordinate axes.

Feedrate

There must already be a feedrate active, or a feedrate must be programmed in the same block.

The following functions, are possible:

G64 / G65	feedrate applies to the contour / tool centre
G93	programming in time segments
G94	programming in mm/min
G95	programming in mm/rev

The achievable feedrate can be limited by the ratio between the feedrate and the contour radius, as well as the programmed distance. See F-address.
The max. feedrate is determined by machine parameter.

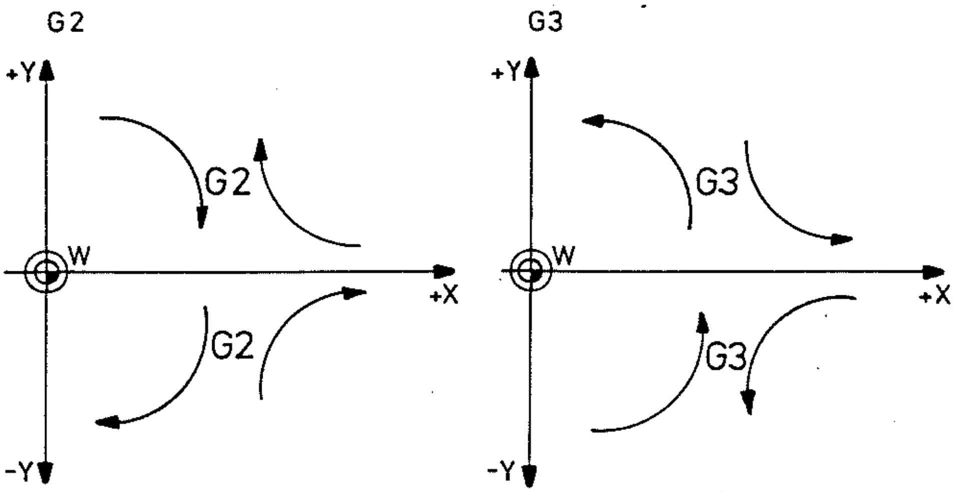
Interactions

G0/1/2/3/5 cancel each other.

Entry into Circle

G5 X... Y...	tangential entry, automatic calculation of the radius
G2/G3 X... Y... R...	any type of entry with programming of the radius
G2/G3 X... Y... I... J...	any type of entry with programming of the centre of the circle

Direction of Rotation



Any size of arc can be defined. Full circles can be programmed using I,J,K. The centre coordinates are always necessary for full circle programming.

Exit from the Circle

The are no restrictions regarding the exit from a circular contour

CIRCULAR INTERPOLATION WITH ANY TYPE OF ENTRY INTO THE CIRCLE

G2/G3 with R

Programming **G2 X.... Y.... R.... (X/Y plane)**

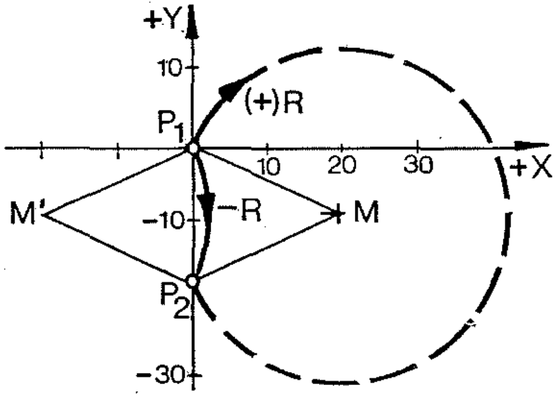
Entry into the Arc If the radius is defined during the programming any entry into the arc can be realized.

Radius R The radius is programmed by the R-address with sign.
Maximum input value: 100 m.
Negative sign: arc smaller than a semicircle.
No sign: arc larger than a semicircle.
(see examples)

Definition of the Arc Given the same starting and end points and radius 4 different arcs are possible.

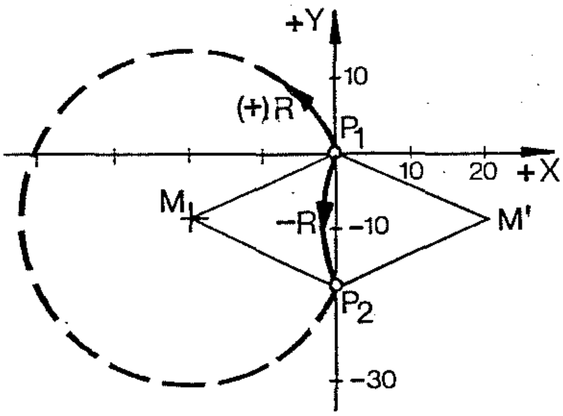
These are distinguished by determining the direction of rotation and the sign of the radius as follows:

Examples **G2 clockwise**



Programming:
N1 G0 X0 Y0 (point P1)
N2 G2 X0 Y-20 R22 F1000
(broken line circle)
or
N2 G2 X0 Y-20 R-22 F1000
(continuous line circle)
N3 M30

G3 counter-clockwise



Programming:
N1 G0 X0 Y0
N2 G3 X0 Y-20 R22 F 1000
(broken line circle)
or
N2 G3 X0 Y-20 R-22 F1000
(continuous line circle)
N3 M 30

Note No programming of full circles possible with R.

CIRCULAR INTERPOLATION WITH ANY TYPE OF ENTRY INTO THE CIRCLE

G2/G3 with I, J, K

Programming	**G2 X.... Y....**	**I....J....**	**(X/Y plane)**
	G2 X.... Z....	**I....K....**	**(X/Z plane)**
	G2 Y.... Z....	**J....K....**	**(Y/Z plane)**

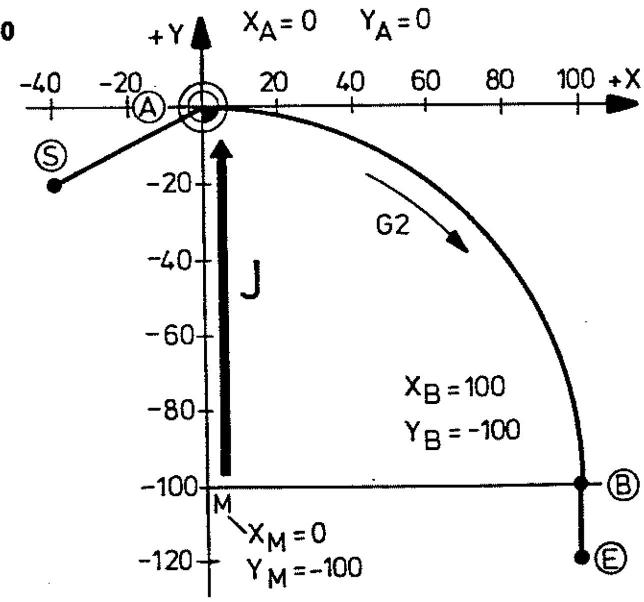
Entry into the Arc If the position of the centre of the circle is defined with I, J, (K) any type of entry onto the circular contour can be realized, as well as full circles.
NOTE: If I, J or K = 0 then this value need not be entered into the program.

Parameters of the Centre of the Circle The position of the centre of the circle is determined by I, J and K.
I, J and K are modal in effect.
X/Y,Z, as well as I, J and K are programmed in absolute or incremental dimensions.

	G90 absolute dimensions	**G91** distances to existing position
I	X position of centre, absolute	distance in X-direction $(X_M - X_A)$
J	Y position of centre, absolute	distance in Y-direction $(Y_M - Y_A)$
K	Z position of centre, absolute	distance in Z-direction $(Z_M - Z_A)$

Example starting point = A, end point B, centre of circle M

G2, G90



Calculation of centre:
$I = X_M - X_A = 0$
$J = Y_M - Y_A = -100 - 0 = -100$

Programming with G90:

```
N1 G1 X-40 Y-20  F1000  Ⓢ
N2 X0 Y0                Ⓐ
N3 G2 X100 Y-100 J-100  Ⓑ
N4 G1 X100 Y-120        Ⓔ
N5 M30
```

(G90 active on switch-on)

Programming with G91:

```
N1 G1 X-40 Y-20  F 1000  Ⓢ
N2 G91
N3 X40 Y20               Ⓐ
N4 G2 X100 Y-100 J-100   Ⓑ
N5 G1 X0    Y-20         Ⓔ
N6 M30
```

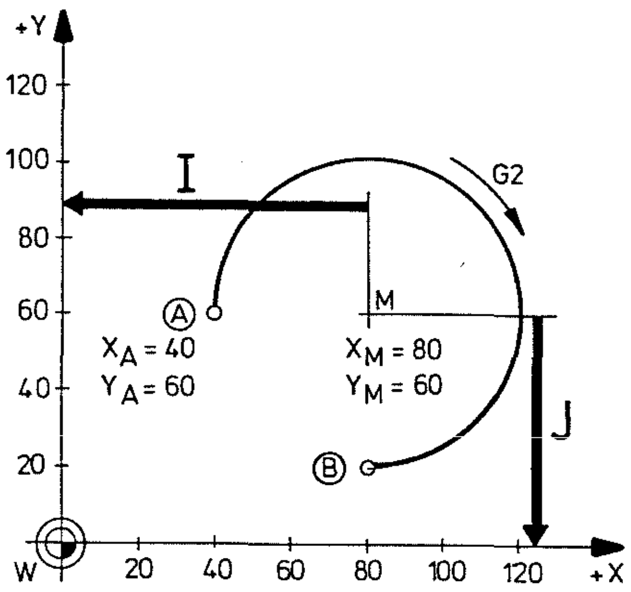
CIRCULAR INTERPOLATION G2/G3

Examples

G90 Starting point X40 Y60.
Radius: 40 mm.

A = starting point
B = end point
M = centre of circle

G2



Calculations:

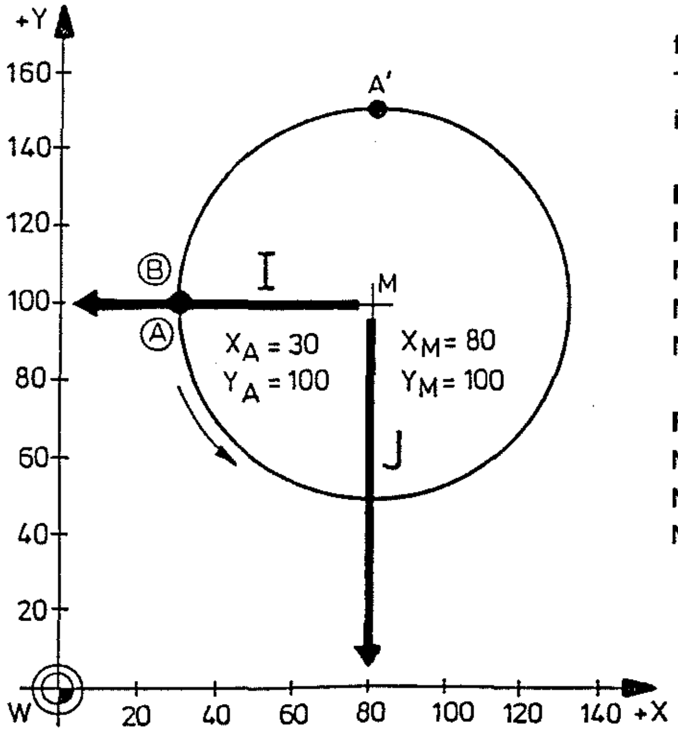
Programming with R

```
N1 G1 X40 Y60 F1000 Ⓐ
N2 G2 X80 Y20 R40   Ⓑ
N3 M30
```

Programming with I, J

```
N1 G1 X40 Y60 F1000  Ⓐ
N2 G2 X80 Y20 I80 J60 Ⓑ
N3 M30
```

G3 Full circle with a radius of 50 mm



A and B are identical for a full circle.
The circle must be subdivided into 2 parts when programming with R.

Programming with R

```
N1 G0 X30 Y100      Ⓐ
N2 G3 X80 Y150 R50  Ⓐ'
N3 X30 Y100 R-50    Ⓑ
N4 M30
```

Full circle programming with I, J

```
N1 G1 X30 Y100          Ⓐ
N2 G3 X30 Y100 I80 J100 Ⓑ
N3 M30
```

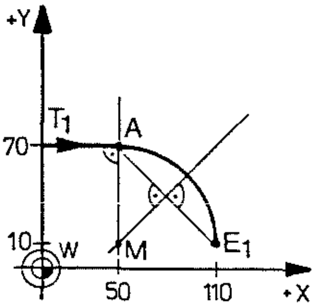
CIRCULAR INTERPOLATION WITH TANGENTIAL ENTRY G5

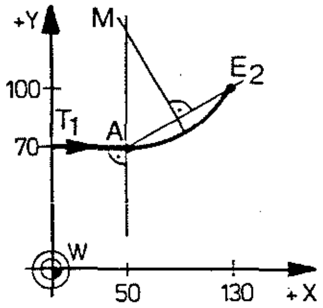
Programming G5 X.... Y....

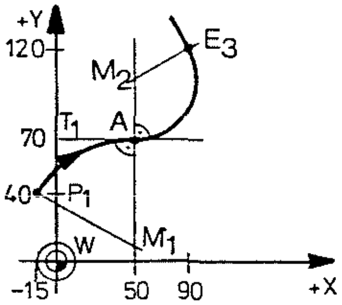
Entry into the Arc When G5 is programmed the control will calculate a tangential entry into the circular contour. No radius is programmed. Only those contour transitions are considered tangential which do not involve a reversal of direction. The control calculates the size and the position of the arc as illustrated in the following examples:

When several G5 movements follow one another the 1st entry tangent influences all subsequent contour elements with G5.

Different End Points





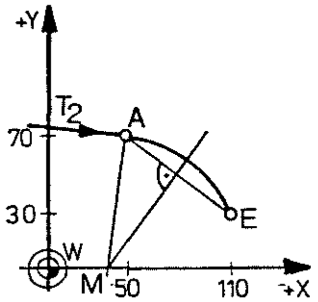


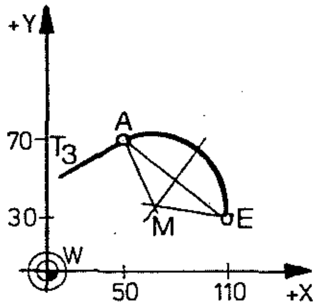
```
N1 G1 X0 Y70 F200
N2 X50
N3 G5 X110     Y 10
N4 M30
```

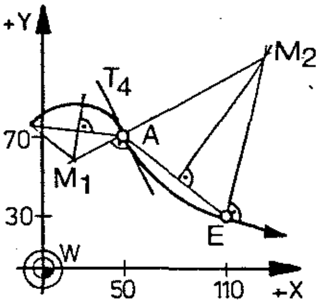
```
N1 G1 X0 Y70 F200
N2 X50
N3 G5 Y130     Y100
N4 M30
```

```
N1 G1 X-15 Y40 F200
N2 G2 X50 Y70 R-60
N3 G5 X90 Y120
N4 M30
```

Different Tangents







Restriction G5 can not be programmed in MDI or as the first block in a part program, since it would not be possible to calculate a tangent.

T_n = tangent   A = starting point of arc

M_n = centre of circle   E = end point of arc

DWELL G4

Definition The execution of the subsequent blocks is not started until the programmed time has elapsed.

Operation G4 only becomes effective in the block in which it is prgrammed and must be programmed on its own.

Modal conditions are retained.

Programming **G4 F....** F in seconds
input range 0.01 to 9 999 999.

Example

```
N12 G1  X10 Y100    F150
N13 G4              F2          2 sec.dwell
N14     Z-60
N15 G4              F 1.78      1.78 sec. dwell
N16     Z0
N17 M30
```

LINEAR INTERPROLATION IN RAPID WITH EXTENDED IN POSITION RANGE G6

Definition

In interpolation mode the control waits until an In Position range is reached before starting the interpolation for the next block.
G6 corresponds to the G0 function, but with a larger In position range (as a rule).
As opposed to the G0 IN-POS range, which is determined as a constant value in the machine parameters (see MP 49, 69, 89, 109) the IN-POS range of the G6 function is related to the max. rapid feedrate (see MP 35, 55, 75, 95):

$$\text{IN-POS range} = \frac{\text{max.rapid feedrate}}{1000}$$

The **smaller** the max. rapid feedrate determined by the machine parameters the more precise (smaller) is the IN-POS range.

After this range is reached the control stops for a short time before the interpolation for the next block is started. The length of this stop time is determined in a separate machine parameter (MP 23) and applies for all axes.

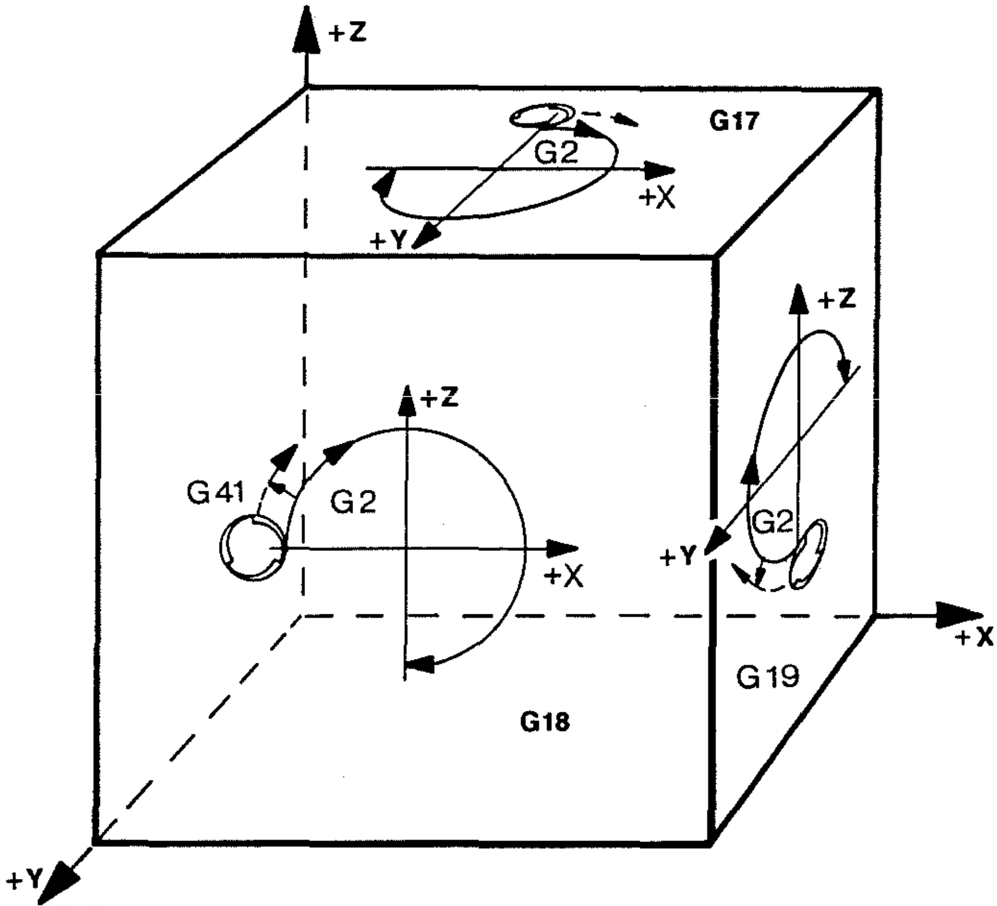
Reactivation of the "normal" IN-POS range by programming G0, G1, G2, G3 or G5.

PLANE SELECTION

G17 A X/Y plane
G18 Z/X plane
G19 Y/Z plane

Definition These G-codes are used to determine the working plane. They also influence the operation of functions G2, G3 and G5, of the tool radius compensation and of the tool length compensation.

Interactions G17, 18, 19 are modal functions and cancel each other. **The definition of a pole with G20 also effectively makes a plane selection.**



Programming A change in the working plane must be programmed before the first circular movement (G2, G3).

A change in the working plane must not be programmed while tool radius or tool length compensation (G41,G42,Txx) is active.

Plane Selection

G-code	circular interpolation tool radius comp. positioning plane for standard boring cycles	tooll length comp. feed-in axis for standard boring cycles
G17	X/Y plane	Z-axis
G18	Z/X plane	Y-axis
G19	Y/Z plane	X-axis

SETTING A POLE G20

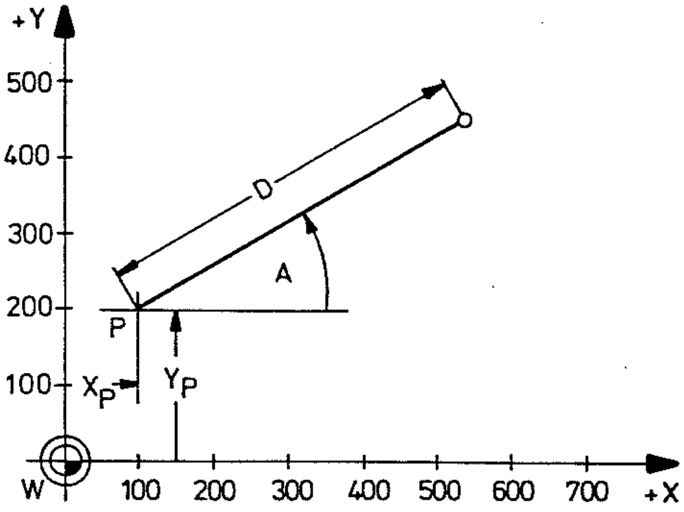
Definition The pole and the associated plane G17/18/19 are determined by 2 axis addresses, which are programmed together with G20.The pole relates to the active zero point. The setting of the pole does not produce any axis movement.

Programming with Polar Coordinates Contour points are defined by the radius and an angle. The data relates to a pole, which is to be defined, and a plane. Positions described in this way are converted within the control into command values for standard axes in a Cartesian system.

Terms

Polar plane	Plane defined by 2 cartesian axes within which the polar coordinates lie.
Pole	Centre of the polar coordinate system. Position of the pole: Without/before G20: on the active program zero point After G20: on the point defined with G20
Radius **D**	Program address assigned to the vector length.
Angle **A**	Program address assigned to the vector angle. In mathematical terms the angle relates to the active reference plane..
Reference axis for angle A	The axis **in bold print** written first in the plane selection. G17 **XY** G18 **ZX** G19 **YZ**

Operation The interpolation modes G0, 1, 2, 3, 5 etc. are not affected by this function.



P = pole

Xp = distance of the pole from the Cartesian zero point in X

Yp = distance of the pole from the Cartesian zero point in Y

A = angle

D = vector length

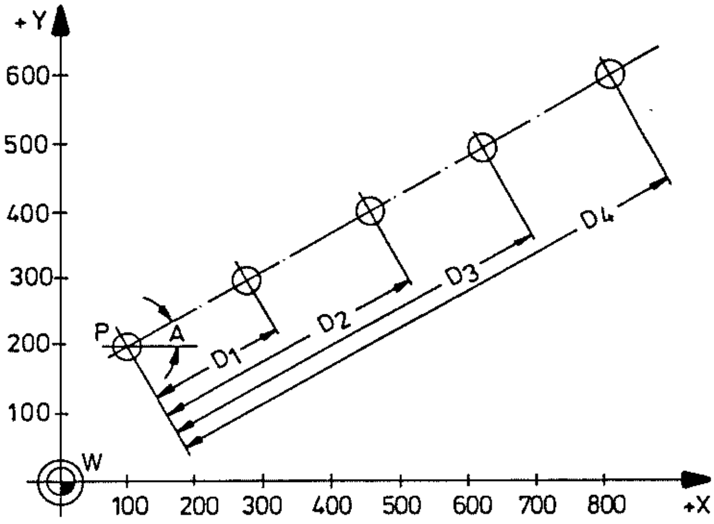
Programming:
G20 X100 Y200

Example G17 Setting a pole in plane XY (polar plane) at position X = 100 Y = 200

Effect with G91 Angle A absolute, vector length D incremental.

POLAR COORDINATES G 20

Example Machining a row of holes with G81

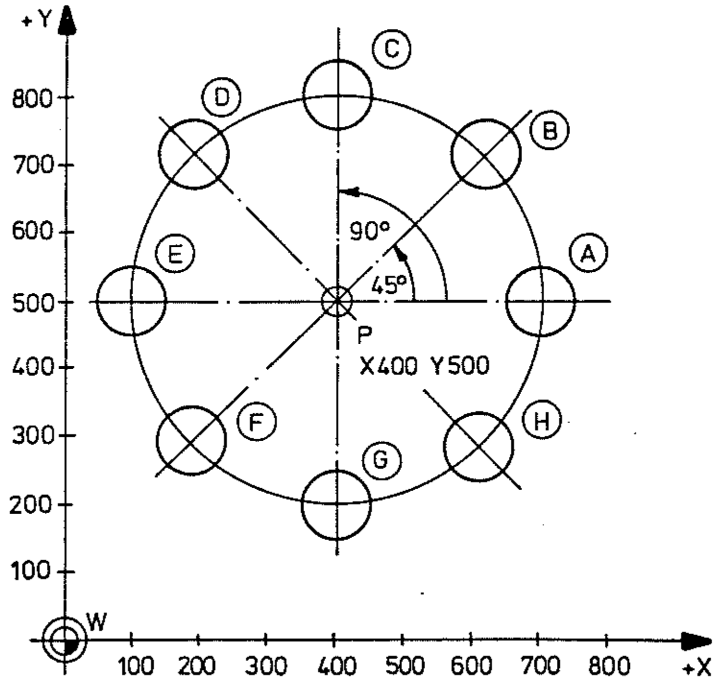


Program:

```
N1    F1000   S500   M3  T01
N2    G81     VI=80 V2=30

N3    G20     X100  Y200
N4    X100    Y200
N5    A30     D200        (D1)

N6    D400                (D2)
N7    D600                (D3)
N8    D800                (D4)
N9    M30
```

P = position of the pole

Example Machining a bolt hole pattern with G81



Programm:

```
N1    F1000   S500   M3  T01
N2    G81     VI=80  V2=30
N3    G20     X400   Y500
N4    X700    Y500         Ⓐ
N5    A45     D300         Ⓑ
N6    X400    Y800         Ⓒ
N7    A135                 Ⓓ
N8    A180                 Ⓔ
N9    A225                 Ⓕ
N10   A270                 Ⓖ
N11   A315                 Ⓗ
N12   M30
```

CONDITIONAL SUBPROGRAM CALL-UP G21

Definition The subprogram call-up is dependent on the status of I/F signal "CONDITIONAL SUBPROGRAM CALL-UP"
Any program label (marked with "$") can be used.

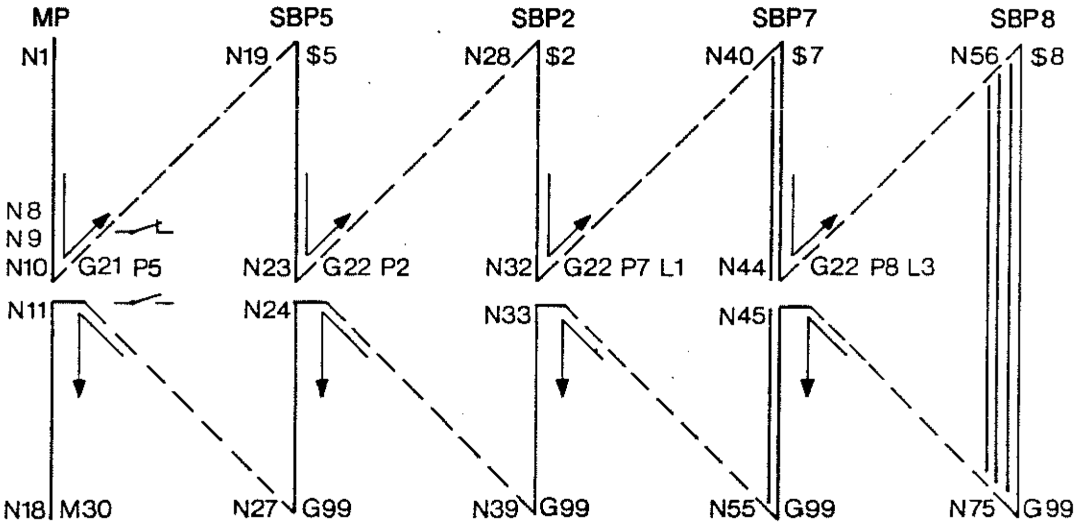
Operation The interface signal "CONDITIONAL SUBPROGRAM CALL-UP" must be present at least 3 blocks before the block in which G21 is programmed.

Status of signal "CONDITIONAL SUBPROGRAM CALL-UP":

High	The subprogram is carried out.
Low	The subprogram is not carried out. (Next block is executed.)

By using backwards jumps it is possible to produce endless program repetitions, for series production for instance.

Subprogram nesting up to 10 programs deep is possible
(nesting: one SBP calls up other subprograms).



MP = main program SBP = subprogram

Explanation of above example:
All the subprograms are only carried out if signal "CONDITIONAL SUBPROGRAM CALL-UP" is high when block 8 is read in.

General Format **G21 P... L...**

P = subprogram number ranging from 0 to 99
L = repetition factor (in addition to 1st execution) ranging from 1 to 99
input of L is dispensable

Programming Example: G21 P10 L1
SBP 10 is executed (1 + 1) = 2 times, if the signal is at high level.

G21 must not be used if tool radius compensation is active.

G21 must be programmed on its own.

SUBPROGRAM CALL-UP G22

Definition

Programs which are marked as subprograms are called up with G22
Any program label (marked "$") can be used.

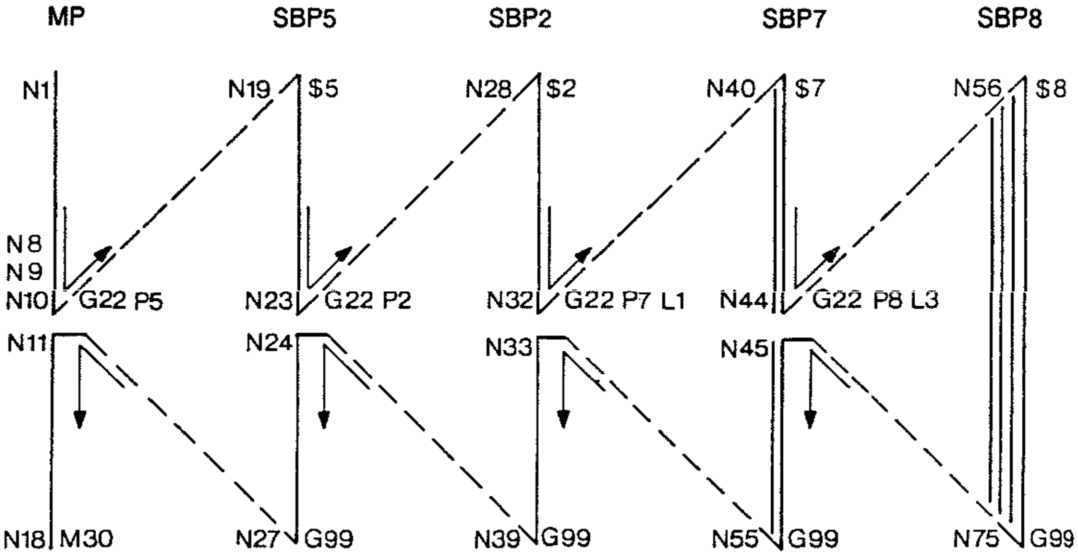
Operation

Subprograms called up with G22 P . . . (L . . .) are carried out unconditionally.
The subprograms of the CC 100 are of local character,
in other words they are always assigned to a particular main program or cycle.

Programming

Example: G22 P5 Subprogram 5 is carried out once.

Subprogram nesting up to 10 programs deep is possible
(nesting: one SBP calls up other subprograms).



MP = main program SBP = subprogram

Explanation of the above example:
On its own the call-up of SBP 8 in block 44 will produce
4 program runs (1.execution + 3 repetitions).
The preceding call-up of SBP 7 in block 32, on its own,
will produce 2 runs of SBP 7.

Total number of	MP	SBP5	SBP2	SBP7	SBP8
program runs:	1	1	1	1+1=2	2x(1+3)=8

General Format

G22 P . . . L . . .

P = subprogram number ranging from 0 to 99
L = repetition factor (in addition to first execution)
ranging from 1 to 99
input of L is dispensable
G22 must be programmed on its own.

CONDITIONAL JUMP G23

Definition

The jump is only carried out if the interface signal "OPTIONAL JUMP" is present. If this condition is not fulfilled, the subsequent block will be executed.

Operation

Any programmed label can be used as jump address. Program labels are marked with $. The interface signal "OPTIONAL JUMP" must be present at least three blocks before the block in which the jump is programmed.

Programming

G23 P . . . P = 1 to 99 for the program label

G23 must always be programmed on its own.

G23 must not be used while tool radius compensation is active.

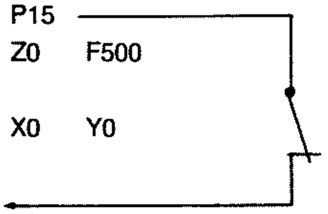
Example

Drilling holes at different positions, depending on the workpiece, if its identification triggers the I/F signal "OPTIONAL JUMP".

Signal = high Blocks 10-12, 19-24, 16-18 are executed.

Signal = low Blocks 10 to 18 are executed.

```
N1  G0    X800  Y500  Z100
N2  G23   P15
N3  G1    Z0    F500
N4  Z100
N5  G0    X0    Y0
N6  X100
N7  $15
N8  G0    Y400
N9  G1    Z-20  F400
N10 Z100
N11 G0    X0    Y0
N12 M30
```



optional jump
dependent on I/F signal

UNCONDITIONAL JUMP G24

Definition During the execution the program is not continued at the next block but at the program label defined in the jump instruction. The program label is marked with $.

Operation The jump is carried out unconditionally. By programming backwards jumps it is possible to produce endless program repetitions, for series production for instance.

Programming General format:

G24 P . . P for the program label

The programming range for P is 1 - 99.

A jump must not be programmed together with other instructions in the same block.

Example Backwards jump from the main program to the second block.

```
:
N9                                              main program 2
N10  $5
N11  G0   X50  Y100 S1000  M3
N12  G1   X52  Y98  F500
N13  G1   Y80
N14  G0   X140
:
N16
N17  G0   X80  Z120  S1000  M3
N18  G1   X0   F300
N19  X78
N20  X76  Z118
N21  G23  P6        IF High                     IF Low
N22  G24  P5
N23  $6
N24  M2
```

Explanation of above program:

Program 2 is repeated continually for as long as input "OPTIONAL JUMP" is low. As soon as this signal goes high machining is concluded with blocks 23 and 24.

FIELD LIMITATION

SETTING MINIMUM VALUES	**G25**
SETTING MAXIMUM VALUES	**G26**
CANCELLING LIMITATION	**G27 A**

Definition

The field limitation prevents the axes from being driven into areas where collisions might occur.
Unlike the limit switches these limitations must be determined separately for each program.

The axes can not position to any point with values
- **under** those programmed with **G25**
- **above** those programmed with **G26**

The input of the axis values does not produce any axis movement.

The limitation values relate to the active program zero point.
Any offset programmed with G92 X. . . Y . . . is not considered.

Operation

The limitation function is modal for all machining modes.
It takes into account tool radius compensation as well as tool wear.

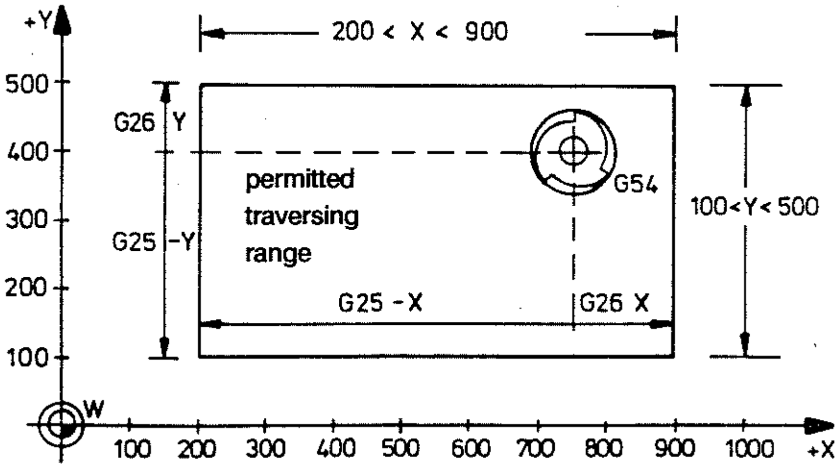
The field limitation does not become activated until the software limit switches are set and the axes have been referenced.

Programming

G25 X. . . Y. . . Z. . . E. . .
The axes must already be positioned within the field of operation.

Cancelling

The limitations set with G25 and G26 are cancelled by programming G27 X Y Z without numerical values as well as by CONTROL RESET.
The software limit switches remain valid.



Example

```
N10  G0   X750  Y400  Z300      axes position above workpiece zero point
N11  G92  X0    Y0              clamping position is taken into account
N12  G25  X-550 Y-300 Z-280     lower limit is determined
N13  G26  X150  Y100  Z200      upper limit is determined.
:         part program
N80  G27  X     Y     Z         limitation is cancelled
```

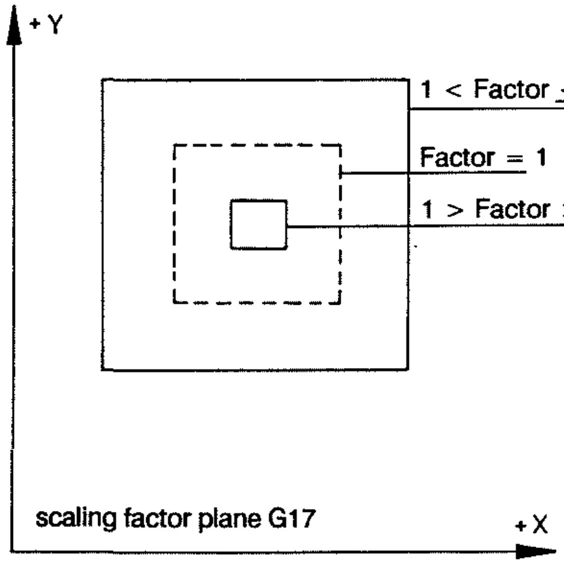
Note

The traversing field limitation set in the machine parameters can not be extended, but only be limited further with G25/G26.

SCALING FACTOR SWITCHING G 36

Definition

Modification of the scaling factor of the coordinate system.
The contour lines of a workpiece are enlarged or reduced in the specified factor area, without having to change the programming of the actual contour lines.
The scaling factor always relates to a particular plane (see next page); the two axes of a plane can not be modified separately.



Modification range of scaling factor

Format

X.xxxxx (5 digits behind decimal point)

Scaling factor data is accepted in decimal format, for instance:

X0.2 = 5-fold reduction (corresponds to reciprocal value of 5)
X5.0 = 5-fold enlargement

Display

The defined scaling factors for the different axes can be displayed in main mode INFO under the CC 100M STATUS display.

Operation

- G36 always relates to a particular plane.
  Example: The programming of the scaling factor for X automatically influences X and Y in plane G17.

- G36 is modal and can be reset with CONTROL RESET, G36 X1 (Y, Z, E) to factor 1.
  A change in plane (G17/18/19) also resets a defined scaling factor. This means that the scaling factor needs to be redefined after each plane selection.

- G36 also operates in the E-axis, if this is defined as a linear axis, whatever working plane is selected.

- Any variables called up in the program are subject to modification according to the scaling factor. The scaled values are, however, not transferred into the variable table or tool table.

- G36 does affect the contents of the zero shift table if it precedes G54-59 in the program.
  G36 does not affect any preceding zero shifts.

- If G36 is programmed in several blocks they overwrite each other. The block last prgrammed has highest priority and the programmed scaling factor remains effective until the next change in scaling factor. The scaling by means of the scaling factor is switched off by programming the scaling factor 1.

Programing

- G36 can be programmed together with main addresses F, S, T, H, but not with any other G-codes or with M-codes 6, 19, 21, 22 in one block.

- G36 is to be programmed with only one axis of the working plane; for working plane G17 this is either X **or** Y; axes Z and E can be programmed independently in the same block with a different scaling factor.

Example

```
N1  G0    X0  Y0  Z0
N2  G17               selection of X/Y plane, clearing all programmed scaling factors
N3  G36   X2          scaling factor for X and Y-axis, 2-fold magnification
N4  G0    Y50         traversing to Y100 mm
N5  G36   X1          switch off scaling
N6  M30
```

PROGRAMMABLE MIRRORING

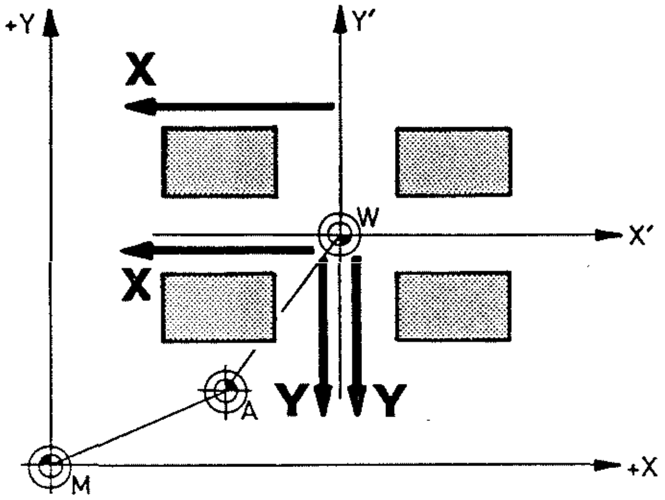
G38 switch on
G39A switch off

Definition 1 or 2 specified axis(es) is (are) mirrored within the selected plane.

The axes are programmed together with G38.

Operation The programmed workpiece positions are interpreted with inverted sign in the relevant axis. The position values are mirrored around the active zero point. This is the zero point which resulted after any possible setting of the position stores with G92, presets or zero shifts.

Reference Points



M = machine zero point
W = workpiece zero point
A = clamping zero point (G92)

X' = axis values after zero shift
Y' = axis values after zero shift

X = X-axis values are mirrored
Y = Y-axis values are mirrored

Programming

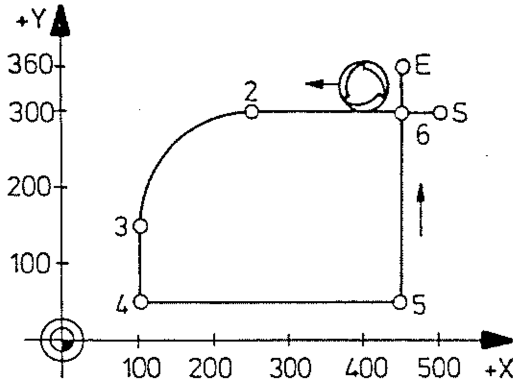
N10	G17	(G18/G19)	plane selection
N11	G38	X or Y or Z	(max. 3 axes)

N10	G39		to cancel all mirroring
N11	G39	X (Y) (Z)	selective cancelling of mirroring in particular axes

The axis addresses are always programmed without axis values.

PROGRAMMABLE MIRRORING G38 / G39

Example Original Contour G17

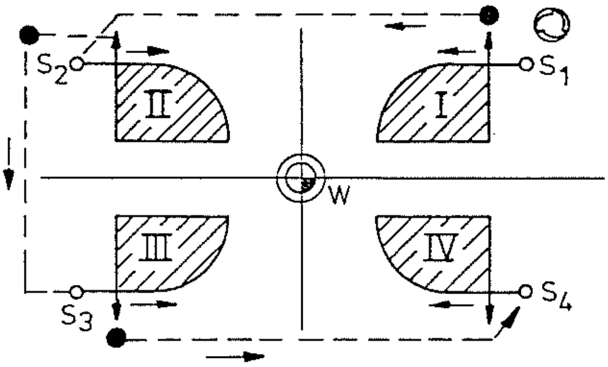


subprogram "contour":

```
N21     $1
N22     G0      X500    Y300    Start (S)
N23     G1      X450    F200
N24     X250    F500
N25     G3      X100    Y150    R-150
N26     G1      Y50
N27     X450
N28     Y360                    End (E)
N29     G99
```

N10	(G17)		execution in 1st quadrant (I) **no mirroring**
N11	(G39)		
N12	G22	P1 Y400	program call-up
N13	G38	X	execution in 2nd quadrant (II) **X-values mirrored**
N14	G22	P1 X500	program call-up
N15	G38	XY	execution in 3rd quadrant (III) **X and Y-values mirrored**
N16	G22	P1 Y400	program call-up
N17	G38	Y	execution in 4th quadrant (IV)
N18	G22	P1	**Y-values mirrored**
N19	G39		mirroring cancelled in both axes
N20	M30		

Operating Sequence during Mirroring



→ tool path for mirroring operations

●auxiliary points

Note

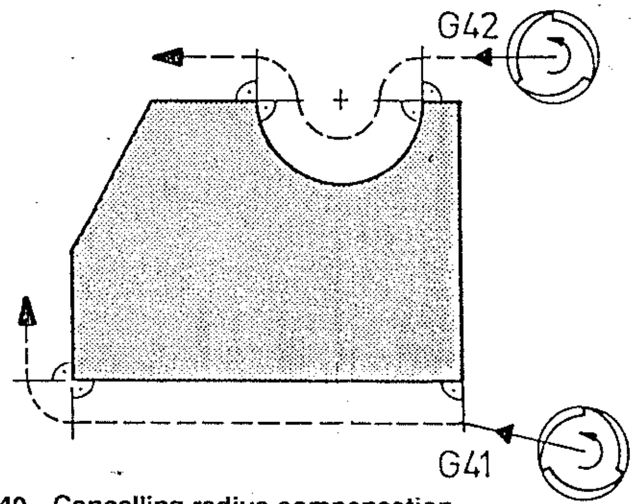
When the values are mirrored for just one axis the control converts G41 into G42 and G3 into G2 etc. internally. See also examples II and IV.

This is not the case when the values for 2 axes are mirrored. See example III.

TOOL RADIUS COMPENSATION G40 A / G41 / G42

Definition

When carrying out a part program with tool radius compensation the tool is guided along an equidistant parallel to the programmed path. Equidistant = path with a constant distance to the programmed contour. The tool length is taken into account by the call-up of the T-address.



Interactions

G40 Cancelling radius compensation.
Active on switch-on, also automatically active when switching into AUTOMATIC mode. G40 must be active at the end of each program.

G41 Call-up of tool radius compensation **to the left of the programmed contour,** viewed in the direction of the path.

G42 Call-up of tool radius compensation **to the right of the programmed contour,** viewed in the direction of the path.

G40/41/42 are modal, exclude one another, and become active with the subsequent traversing movement. G41/G42 can not be used in manual operation.

Programming

```
G41  X...       Y... (T...)
G1   X...       Y... F...
.
.
contour
.
.
G40  X... Y...
```

During the phasing in of the compensation, T must either already be active or be programmed feedrate, compensation group call-up and cancellation with a positioning movement in the involved axes (XY for G17), which is suitable for phasing in/out the compensation.

For detailed description see TOOL RADIUS COMPENSATION, Chapter 5.

Note:

- A block with axis address without traversing movement, because the axes have already positioned, is not allowed with G41/42.

Safety Consideration

- During the exit from the contour a reversal of direction of the cutter movement must be prevented. The angle under which the cutter moves away must therefore always be smaller than 90 .

ZERO SHIFT

G53 A cancel zero shift
G54 to G59
activate zero shift

Definition

By using zero shifts programs can be carried out in different places without any modification. While a zero shift is active the machine parameters are temporarily overwritten. They can be reactivated simply by programming G53.

Operation

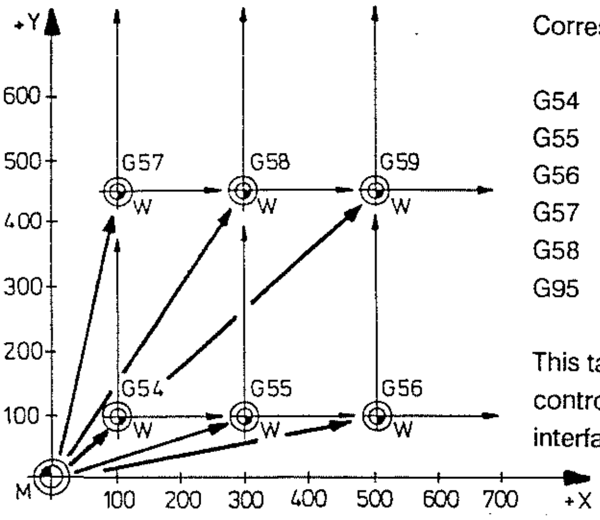
Up to 6 zero shifts can be stored in the zero shift table.
For each zero shift up to 1 value each can be stored for X, Y, Z and E.

If G54 is then called up, for instance, the control will shift the zero point to the machine coordinates which were stored under G54. In order to use a zero shift (for instance G54) the zero shift table must already have been loaded with the respective offset data.

Programming

G54 — on its own this does not produce any axis
or — movement (display changes to programmed position)
G54 X.. Y.. Z.. — the zero shift aleady applies to the position programmed in this block

Example:



Corresponding zero shift table:

G54	X100	Y100	Z70
G55	X300	Y100	Z70
G56	X500	Y100	Z70
G57	X100	Y450	Z70
G58	X300	Y450	Z70
G95	X500	Y450	Z70

This table can be loaded by manual input at the control, via parametric functions, via the serial interface, or via the BCD data bus.

N10	**G54**	1st zero shift active
N11	G22 P5	call-up of part program
	.	
	.	
N20	**G59**	6th zero shift active
N21	G22 P5	call-up of part program
N22	**G53**	machine coordinates apply one more

Condition

- No circular interpolation (G2, G3, G5) must follow immediately after an active zero shift. Operation must start or continue with linear interpolation.

- G36 modifies the contents of the zero shift table if it is written **in front of** G54-59 in the program. G36 only affects subsequent zero shifts.

- When G92 is cancelled any active zero shifts G54-59 are also reset.

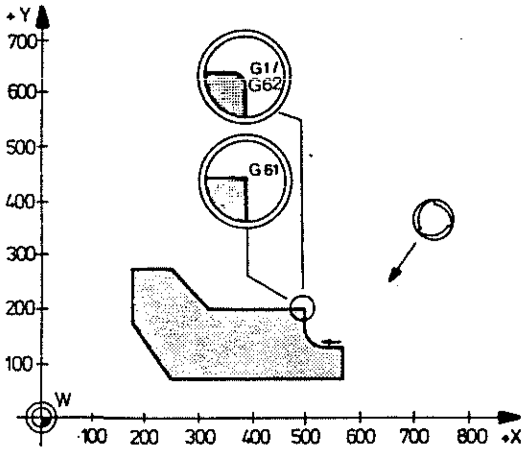
'IN POSITION' LOGIC ON G61
'IN POSITION' LOGIC OFF G62 A

Definition G61 In interpolation modes G1, 2, 3, 5 the control waits for each block until the 'In Position' window has been reached before it starts with the interpolation for the next block. The width of this window is determined by machine parameter. Once the window has been reached the control stops for a short time before interpolating the next block. The duration of this stop time is determined by a machine parameter.

Definition G62 When the 'In Position' function is switched off the control starts with the interpolation of the next block while the last path section from the previous block is being actioned. This results in a "cutting of corners", but saves time.

Operation Functions G61/G62 are modal and cancel one another.

G62 G62 is effective on switch-on



Programming G61, G62 must be programmed at the latest in the block for which they are to be effective.

Influence of machine parameters

IN POS time	-	MP23
IN POS range	-	MP49, 69, 89, 109

(see Connections manual for CC100M)

Example

```
N10   G61                        no movement
N11   G1    Y500   F200          interpolation with IN POS
or
N10   G62                        'IN POS' function off
N11   G1    Y500    F200
:
N50   G61   X200                 interpolation with IN POS in this block
```

FEEDRATE AND SPINDLE SPEED (S) 100% G63
FEEDRATE AND SPINDLE SPEED OVERRIDE VIA POT G66 A

Definition	**G66**	The position of the relevant override potentiometers on the manual panel affects the commanded values.
Definition	**G63**	Feedrate and spindle speed are set to 100% of the programmed/ entered value, whatever the position of the potentiometers.
Operation		G66 is active on switch-on. Both functions are modal and exclude one another. Override ranges: - feedrate 0 to 120% of the programmed value - spindle speed 50 to 150% of the programmed value
Programming		Can be programmed with other instructions in the same block.
Application		The override potentiometers for feedrate and spindle speed can be deactivated by means of programming.
Note re.	**G66**	The potentiometers take effect even when the maximum feedrate is programmed. If the potentiometer is set to between 100% and 120% the maximum feedrate will be exceeded.

EFFECT OF FEEDRATE

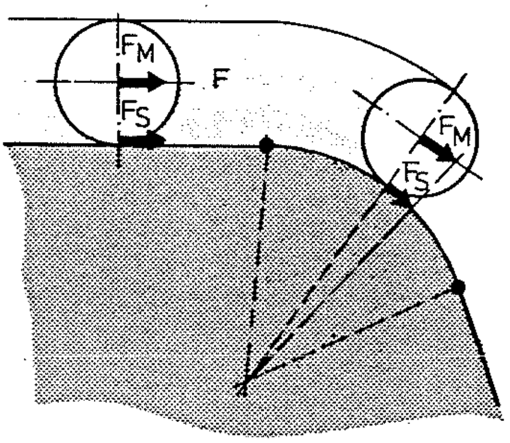
G64 cutting path
G65 A cutter centre path

Definition The feedrate determined with F relates to the cutting path of the cutter or to the cutter centre path when machining circular contour sections.

Interactions G64/65 are modal and exclude one another.

G65 The control keeps the feedrate along the cutter centre path constant. G65 is active on switch-on and is used for roughing.

Examples



F_{prog} = programmed feedrate

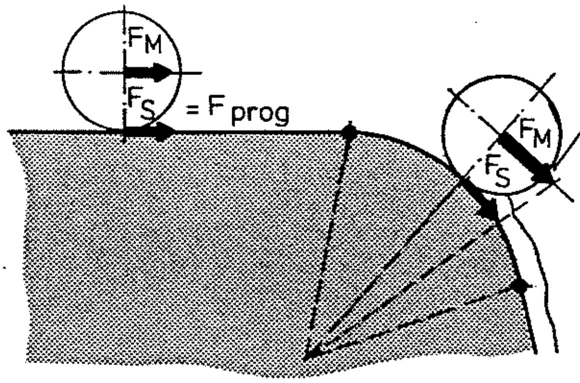
F_M = feedrate along cutter centre path

F_S = feedrate along the cutting path $F_S = F_{prog}$

In the example the feedrate effective on the actual contour is lower than the programmed value.

G64 The control keeps the feedrate along the cutting path constant. These calculations can only be carried out for arcs G2/3/5 if G41/G42 is active.

Since the speed can increase considerably on circular contours this function shoud only be used during finish milling.



increased feedrate since $F_S = F_{prog}$

The effective axis feedrate is higher than the programmed one in the above example.

CONTOUR TRANSITIONS

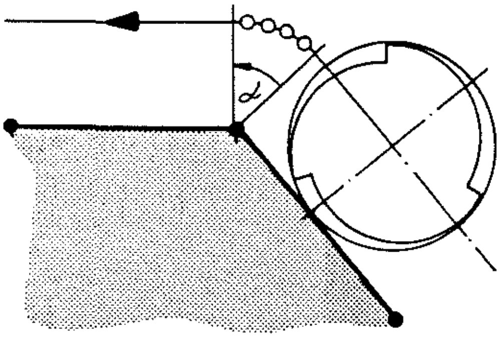
G69 **intersection**
G68 A* **arc**

Definition

If tool radius compensation is active the control must create transitions for outside corners. These transitions can either be the intersections of the equidistants or automatically generated arcs. G68/69 are modal and cancel one another.

Operation
G68

Arc
Only in conjunction with G41/42 with an angle alpha of between 0° and 180°
The arc produces a continuous transition, which usually is the best solution technologically and puts less strain on the drives due to the soft transition.



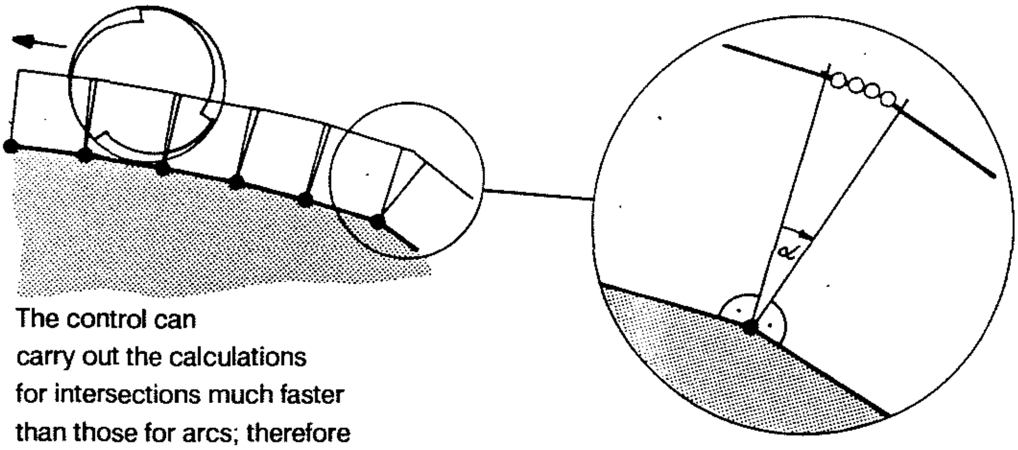
ooo generated automatically

●●● programmed points

G69

Intersection
Only in conjunction with G41/42 with an angle alpha < 90%.

With angles of ≥ 90° up to 180° the control will produce transitions as if G68 had been selected.



The control can
carry out the calculations
for intersections much faster
than those for arcs; therefore
G69 is most suitable for contours
which require extremely fast block sequences.

Programming

G68/G69 without axis information.
If G68/69 is used while path compensation is active the function must be programmed 3 blocks in advance.

Function active on switch-on

Either G68 or G69 can be defined as active on switch-on by machine parameter.

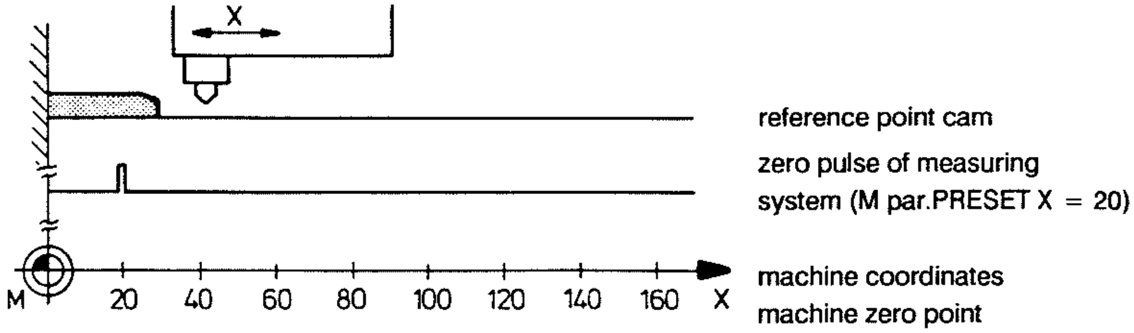
*dependent unpon machine parameter

REFERENCING G74

Definition The axes programmed in the block traverse simultaneously onto the reference point(s) at the feedrate determined by m/c parameter.

Once the reference point has been reached the axis position values are set to machine specific values (machine parameters).

Example X-axis



Interactions G74 cancels zero shifts which were activated with G54 to G59 or G92. No tool compensations must be active during G74. While G74 is carried out all modal conditions are temporarily suppressed.

Programming G 74 X Y Z E

G74 is programmed in a separate block with just the relevant axis addresses without numerical values.

Example G74 X and Z traverse to the reference point.

N7 G74 X Z

Note Further details on interactions with other functions can be found under

G25, G26	field limitation
G53-59	zero shifts
G92	setting position stores.

MEASURING PROBE INPUT G75

Definition

The control drives the measuring axes in the direction of the programmed position with linear interpolation (G1). While the axes are traversing the switching condition of the measuring probe is being monitored.
As soon as the signal becomes 1 (probe touching surface to be measured) the control responds as follows:

- it stores the current position values and
- cancels the distance to go and G75.

Once the G75 operation is completed the control automatically retracts the measuring axis to the position at which contact was made.

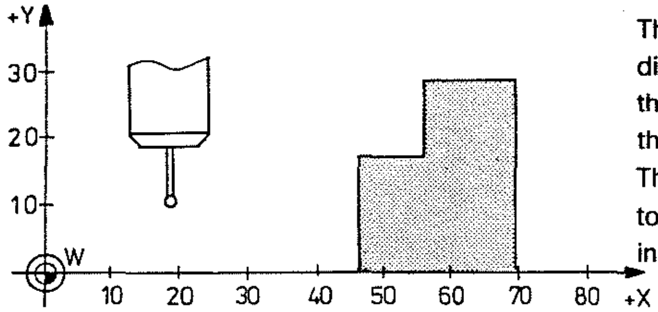
Interactions

G75 is effective only in the block in which it is programmed and automatically sets linear interpolation G1. The control only responds to the closing of the probe contact, and not to the opening of this contact. If the end point programmed with G75 is reached without the probe contact closing the program is interrupted and the error message " probe not triggered" is output.

Programming

G75 X... (and/or Y.../Z.../E.../F...)

Example



The probe is to be used to measure the distance from the two surfaces on the left hand side of the workpiece to the zero point. The axes have been driven to the starting position in a machining program.

N1	G75 X70 F200	traverse towards workpiece until probe is triggered
N2	M0	programmed stop to read position value
N3	G0 X40	reposition for next measuring operation
N4	Y20	
N5	G75 X70	traverse towards workpiece to measure distance to upper surface
N6	G0 X0	
N7	M3	

Applications

The following tasks represent some of the applications:
- part recognition
- checking workpiece accuracy
- setting reference point at surface of workpiece
- tool inspection

Note

When working with the measuring probe the tool compensations must be switched off, otherwise the following error message will be displayed: "G-code not allowed with cut. or length comp."
The feedrate should be kept moderate in order to avoid damage to the probe.

MACHINING OF BORES

G80 A **Fixed Cycles Off**
G81-87 **Fixed Cycles On**

Usage

The programming of fixed cycles to machine bores is simplified with the cycles described below.

In the course of the programming the user calls up the relevant fixed cycle. Values are entered for the variables; the variables are illustrated in the fixed cycle graphics.

Conditions

The fixed cycles can not be used while tool radius compensation is active; if necessary the tool radius compensation must be deactivated with G40.
Further conditions:
- F feedrate in mm/min
- S spindle speed
- M3/4 spindle rotation clockwise/counter-clockwise

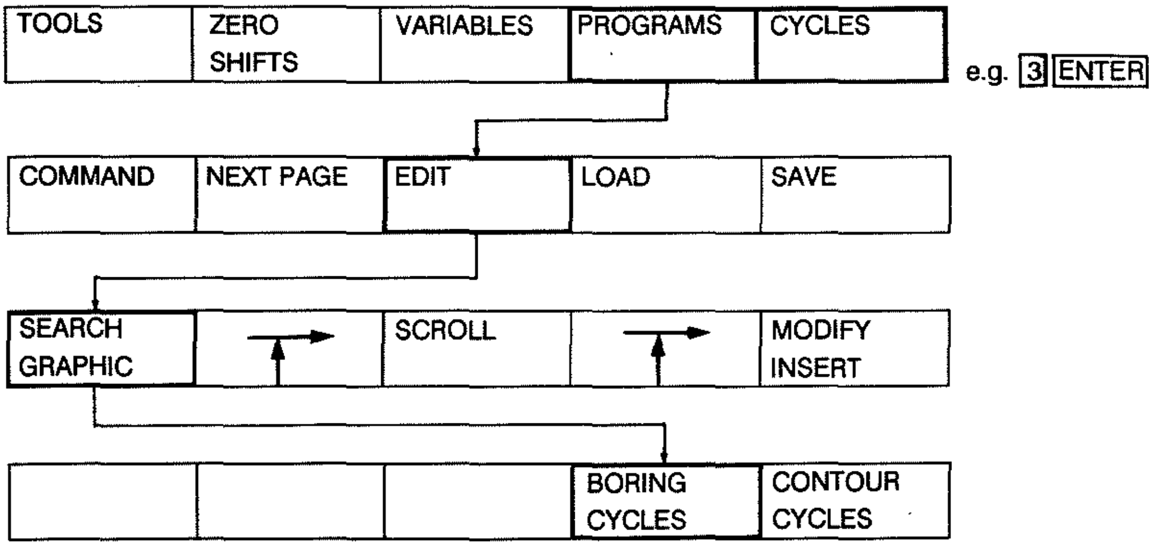
Movements in the positioning plane are all performed in rapid with IN POS operation.
The spindle is switched on with the first positioning movement.
The cycles can be used with G90 or G91 for both axis directions of the feed-in axis.

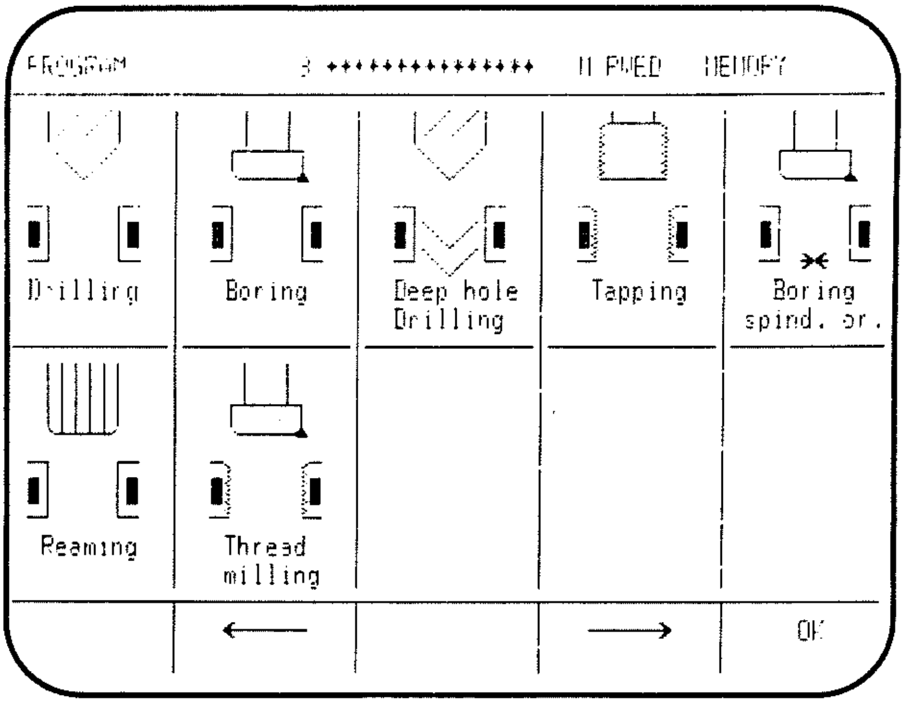
OPERATION

Fixed cycles G81 - G87 are executed in each traversing block once the programmed position has been reached. The selected function is cancelled by programming G80, M2, M30 or by selecting another cycle.

Call-up of fixed cycles and input of variables:

Select main mode EDIT





The selection of the required fixed cycle graphic is made with the ← and → keys. When the required cycle is reached (cycle name flashing) the selection is confirmed with OK. The control will then automatically transfer into the selected fixed cycle level.

The inputs for the different variables in a particular cycle can be confirmed with the keys TAB →

The cursor will then automatically jump to the next variable.

Handling of cycles
(RAM cycles, boring cycles, contour cycles)

See CC100M connections manual.

SURVEY OF FIXED MACHINING CYCLES

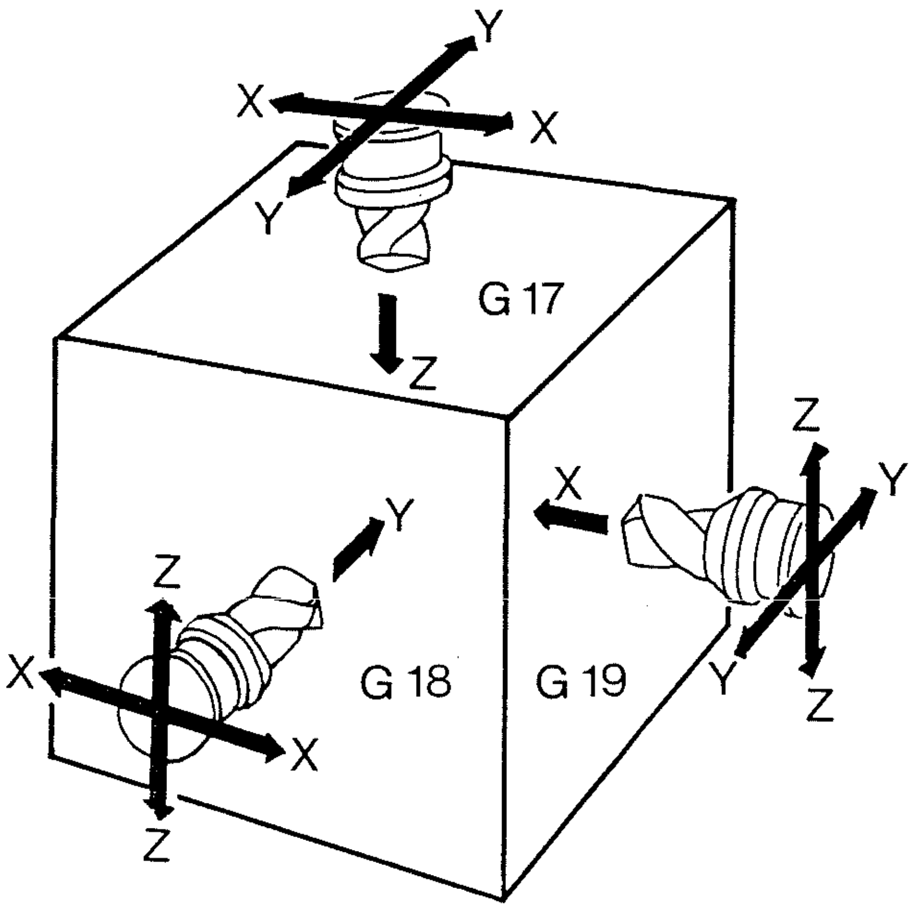
G80 - 87

Type of machining	CODE	Machining sequence		
		Feed- in movement	At depth	Retract movement
drilling	G81	M3 feed	-	rapid M3 active
boring with dwell	G82	M3 feed	dwell	rapid/ feed M3 active
deep hole drilling with swarf removal	G83	M3 (posit. in rapid +feed-in strokes in feed)	- -	swarf removal strokes in rapid M3 active
tapping with tap holder	G84	M3 (M4) feed	M4 (M3) dwell	feed M4(M3) active
boring with spindle orientation	G85	M3 feed	orientation, retract in positioning axis	rapid active
reaming	G86	M3 feed	-	feed with stop for measuring M0 M5
thread milling	G87	M3 helical interpolation	retract in positioning axis M5	rapid
cancelling fixed cycles	G80			

Note When editing fixed cycles the control will display the appropriate graphic for the active plane.

FIXED MACHINING CYCLES G80 - 87

Plane Selection The fixed cycles can be used in the 3 main planes.
The selection of the interpolation plane determines the following:



Positioning plane	Feed-in axis, positioning level, tool length compensation, workpiece surface, working depth	Code
X, Y	Z	G17
Z, X	Y	G18
Y, Z	X	G19

The setting of a pole with G20 effectively also represents a plane selection.

FIXED MACHINING CYCLES G80 - 87

Programming Technique

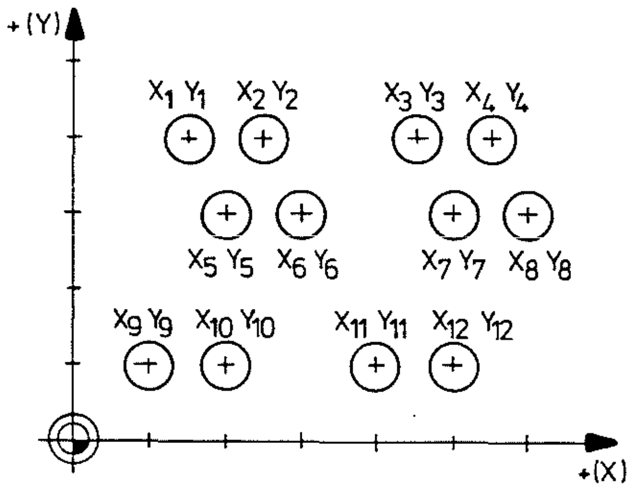
Fixed cycles simplify programming by their modal character.
Programming is rationalized particularly well if the machining can be described by a machining graphic which can be used repeatedly.
Only the different bore positions need then be programmed (see example).

Machining Graphic

The machining graphic contains the coordinates and data which remain constant.

The call-up of the machining graphic is preceded by the selection of the particular fixed cycle with the required feedrate and spindle speed etc.

Example Machining in Z (G17)



$1	
X_1	Y_1
X_2	Y_2
X_3	Y_3
.	
.	
.	
X_{11}	Y_{11}
X_{12}	Y_{12}
X_{13}	Y_{13}

Call-ups	:					
	G0		M . . .	S . . .	T . . .	
	TCH					tool change
	Z_1		F_1	S_1	M3	preconditions for drilling
G81	**G81**		V1 to	V4		
	G22	P1				call-up of **machining graphic**
	M5					
	TCH					
	Z_2		F_2	S_2		precond. for deep hole drilling
G83	**G83**		V1 to	V6		
	G22	P1				call-up of **machining graphic**
	M5					
	TCH					
	Z_3		F_3	S_3		precond. for tapping
G84	**G84**		V1 to	V5		
	G22	P1				call-up of **machining graphic**
	M5					

FIXED MACHINING CYCLES G80 - G87

Variables V

The program variables V1 to V6 are used by the fixed machining cycles. The fixed cycles use program variables V1 to V6, i.e. the contents of these parameters are modified by the call-up of a fixed cycle. When calling a fixed cycle all the relevant parameters must be defined. The variables must be programmed in one line together with the G-code for the particular fixed cycle.

Positions

Position values in the positioning plane relate to

the active zero point with G90

the previous position with G91

The data V1 to V6 for the feed-in axis are independent of G90/91 and are marked individually as

abs. = absolute values or

inc. = incremental values

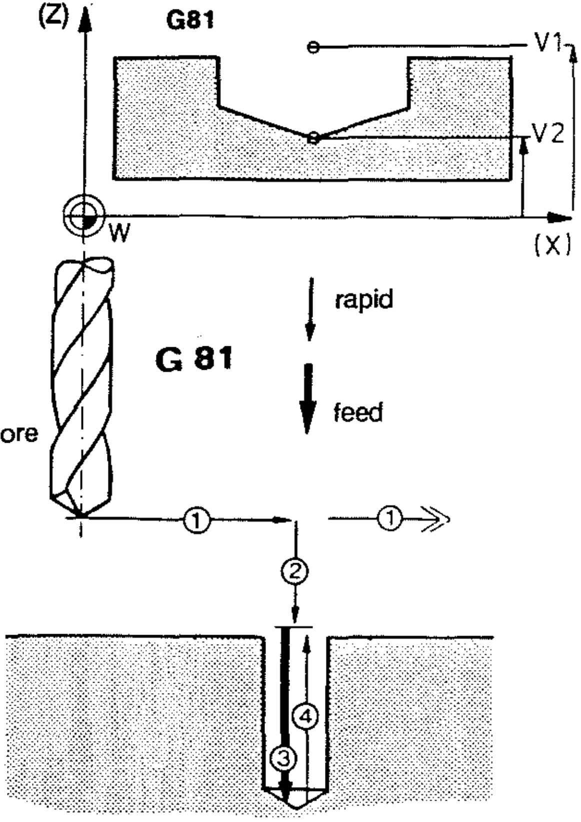
Spindle Rotation

Unless otherwise described for the particular cycle, the main spindle is switched on before the start of the movement in the positioning plane, and it is not stopped automatically after the execution of the cycle.

Safety Consideration

All fixed cycles operate with METRIC dimensions internally. If a fixed cycle is called up in an INCH program the variables are converted into metric values. After the execution of the cycle the variables will be processed in the program as INCH values.

DRILLING



Definition Drilling,
centering

G0 - - > G1 (and vice versa)

Input change-over point V1 mm abs.
depth of bore V2 mm abs.

Sequence spindle on

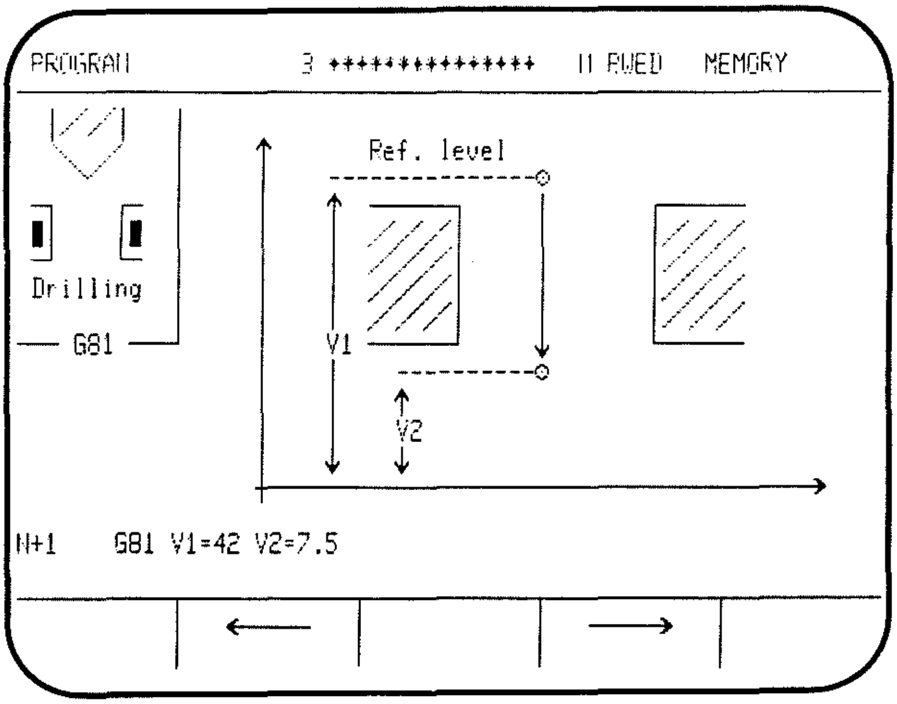
1 positioning axes drive to the centre of the bore in rapid; feed- in axis remains at traversing height

2 feed-in axis drives to V1
change-over to feed

3 feed-in axis drives to V2 in feed

4 retract to V1 in rapid

```
:
N9    T0101                                  tool selection
N10   F500      S250      M3                 preconditions
N11   G81       V1=42     V2=7.5             call-up of cycle G81
:                                            and definition of variables
N12   X125      Y175                         machining positions
N13   X128      Y204
N19   G80                                    cancellation of cycle
:
```



```
PROGRAM          3 ***************    M FWED   MEMORY

N+1    G81 V1=42 V2=7.5
```

BORING

DEFINITION Boring or end facing with dwell at the bottom of the bore

Input

change-over point G0-G1 V1 mm abs.
depth of bore V2 mm abs
dwell V3 secs.
retract G0: V4 = 0
G1 V4 = 1

Sequence

spindle on

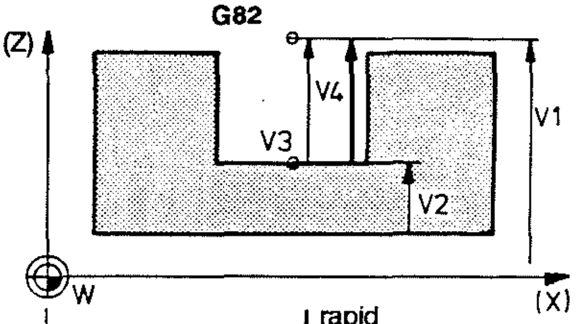
1 positioning axes drive to the centre of the bore in rapid; feed-in axis remains at traversing height

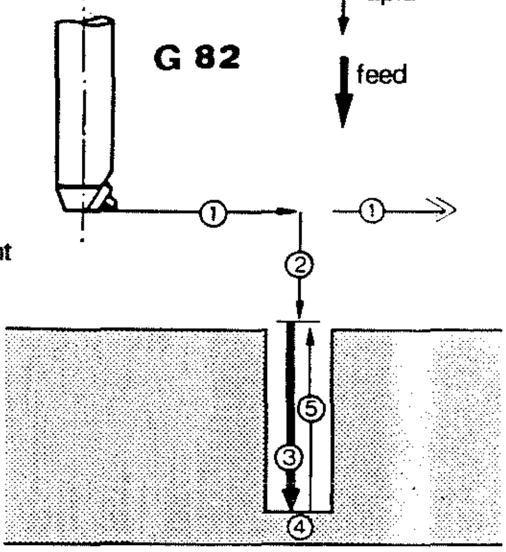
2 feed-in axis drives to V1; change-over to feed

3 feed-in axis drives to V2 in feed

4 dwell at bottom of bore for free-cutting

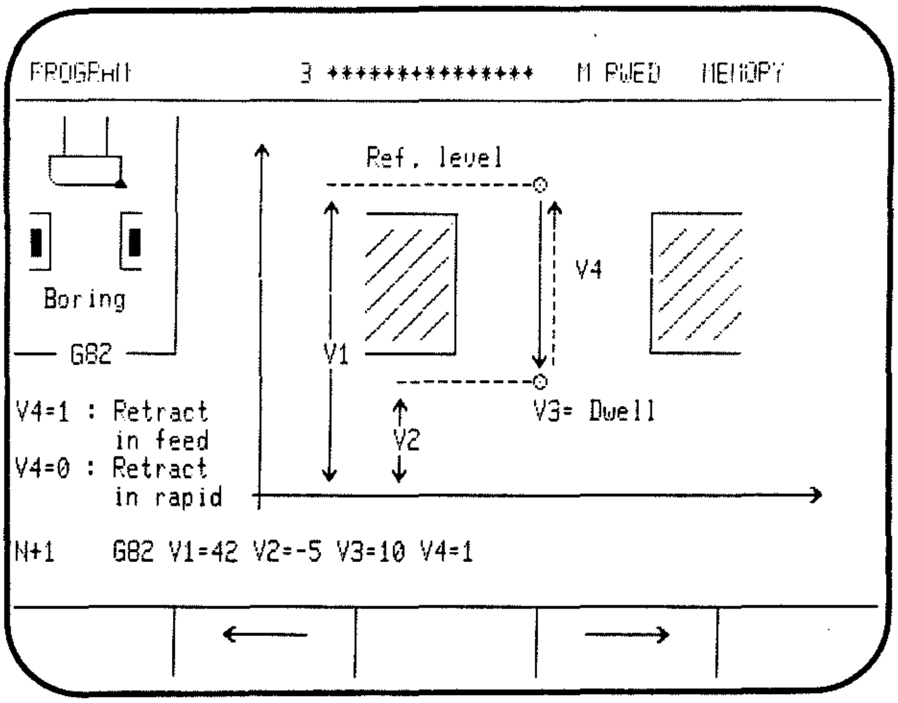
5 retract to V1 in rapid or feed





Example

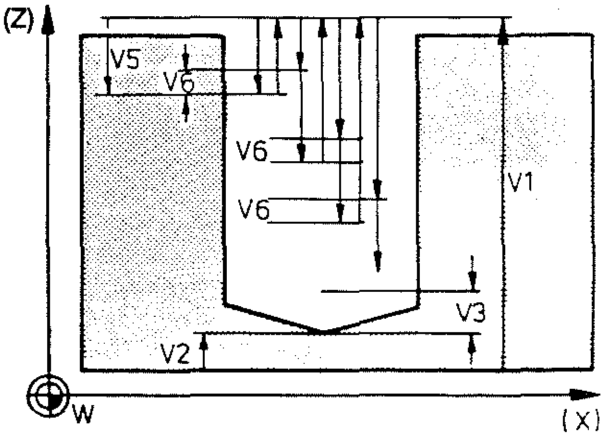
```
:
N9    T0101                                        tool selection
N10   F500    S250  M3                             conditions
N11   G82     V1=42V2=-5  V3=10  V4=1              call-up of cycle G82
N12   X75     Y109                                 machining positions
N13   X90     Y122
:
N19   G80                                          cancellation of cycle
:
```



```
V4=1 : Retract
       in feed
V4=0 : Retract
       in rapid

N+1    G82 V1=42 V2=-5 V3=10 V4=1
```

DEEP HOLE DRILLING G83

Definition Deep hole drilling in several feed-in movements with retract movements to remove swarf.



Input

change-over point	V1 mm abs.
final depth of bore	V2 mm abs.
minimum feed-in	V3 mm inc
degression factor	V4
first feed-in	V5 mm inc.
safety distance	V6 mm inc.

Sequence spindle on

1 positioning axes drive to the centre of the bore in rapid; feed-in axis remains at traversing height

2 feed-in axis drives to V1; change-over to feed

3 feed-in axis drives to depth 1 (V5*1) in feed

4 in rapid to V1

5 back to depth 1 +V6

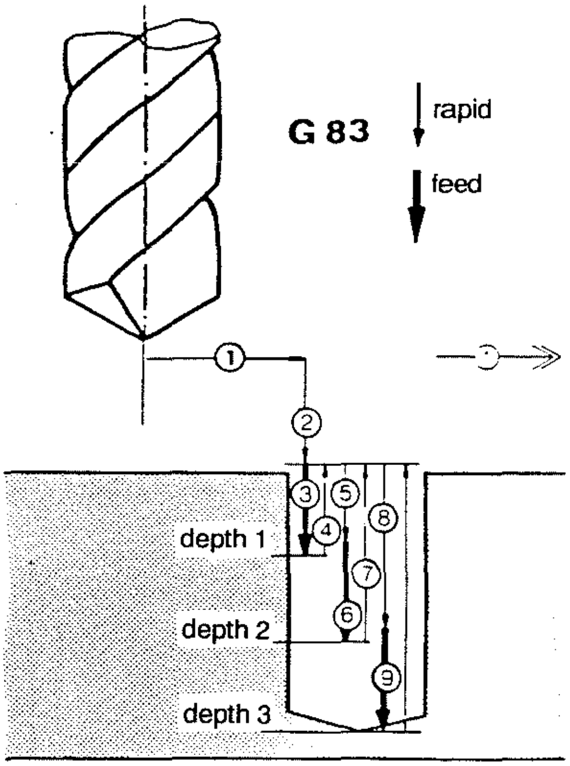
6 in feed to depth 2
V5*(1 +V4)

7 in rapid to V1

8 back to depth 2 +V6

9 in feed to depth 3
V5*(1 +V4+ $V4^2$)

ect.

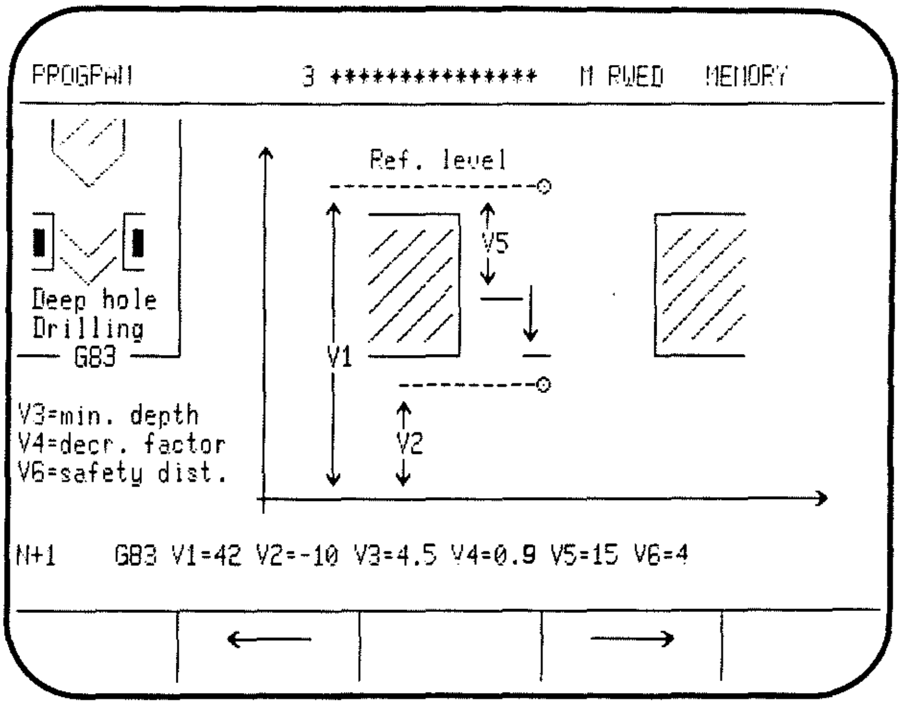


Degression Factor

The degression factor determines the individual feed-in depths for deep hole drilling. At each stage the previous feed-in depth is multiplied by the control with the degression factor in order to establish the next feed-in depth for the deep hole drilling cycle.
The final depth is approached directly during the last feed-in movement.
If the chosen degression factor or the remaining distance would produce a feed-in of less than V3 this is prevented by a corrected input for the feed-in.

Example

```
:
N4    T0101                                                  tool selection
N5    F500  S250  M3                                         conditions
N6    G83  V1=42  V2=10  V3=4.5  V4=0.9  V5=15  V6=4        cycle call-up
                                                             and definition
                                                             of variables
N7    X92   Y17                                              machining
N8    X88   Y42                                              positions
:
N19   G80                                                    cancellation of cycle
```

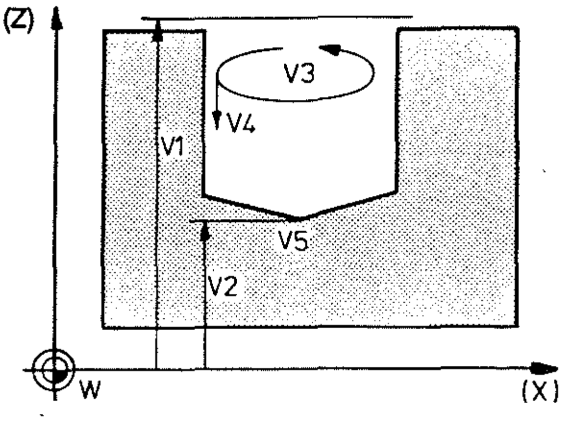


```
N+1     G83 V1=42 V2=-10 V3=4.5 V4=0.9 V5=15 V6=4
```

TAPPING G84

Definition

Tapping with central feed-in.



Input

change-over point		V1 mm abs.
depth of bore (thread)		V2 mm abs.
rotation:	M3/M4	V3 3 inward
	M4/M3	V3 4 outward
feedrate		V4 mm/rev
dwell		V5 sec.

Sequence

spindle on, single block suppressed

1 positioning axes traverse the centre of the bore in rapid; feed-in axis remains at traversing height

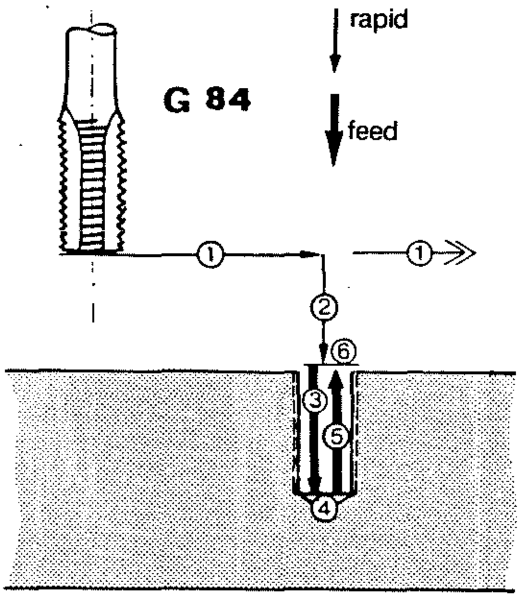
2 feed-in axis drives to V1; change-over to feed

3 feed-in axis drives to depth V2 at feedrate determined by V4

4 reversal of spindle rotation; dwell at bottom of bore

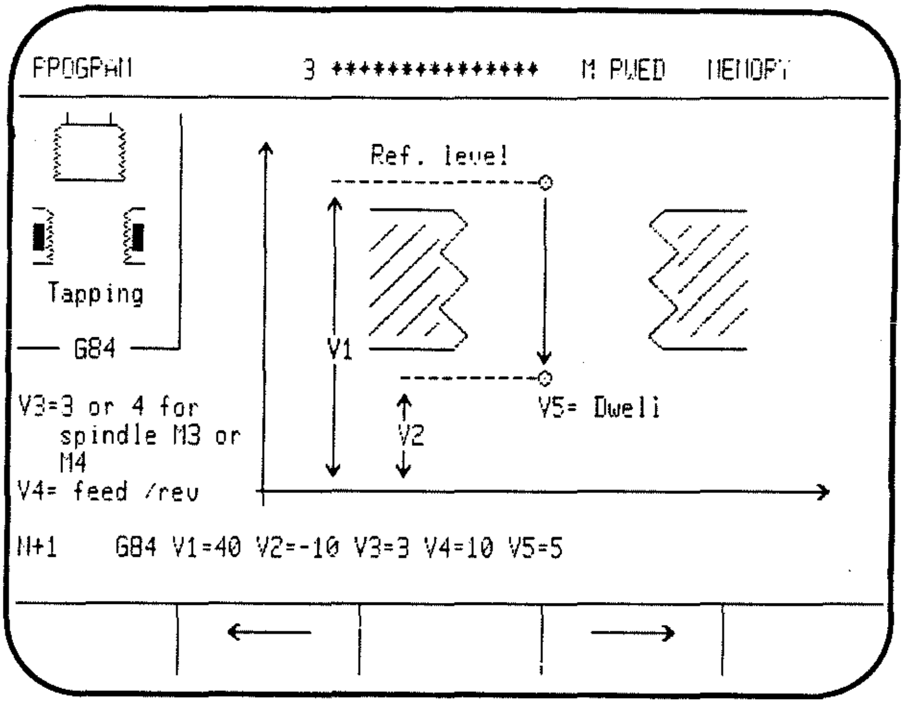
5 retract to V1 in feed

6 spindle stop single block possible again



Example

```
:
N9    T0101                                        tool selection
N10   F500  S250  M3                               conditions
N11   G84   V1=40  V2=-10  V3=3   V4=10   V5=5     call-up of cycle
                                                   G84 and definition
                                                   of variables
N12   X16   Y52                                    machining
N13   X27   Y48                                    positions
 :
N19   G80                                          cancellation
                                                   of cycle
```



```
PROGRAM          3 ***************     M PWED     MEMORY
```

```
N+1     G84 V1=40 V2=-10 V3=3 V4=10 V5=5
```

The following functions are activated:

M3	spindle rotation clockwise
M4	spindle rotation counter-clockwise
M98	single block suppressed
M99	single block possible

Note

- Feed conditions active before the call-up of the cycle are stored and reactivated automatically once the cycle has been completed.
- While G84 is active the reentry functions are not active.
- Feedrate 100% is set automatically;
  single block is suppressed automatically (M98).

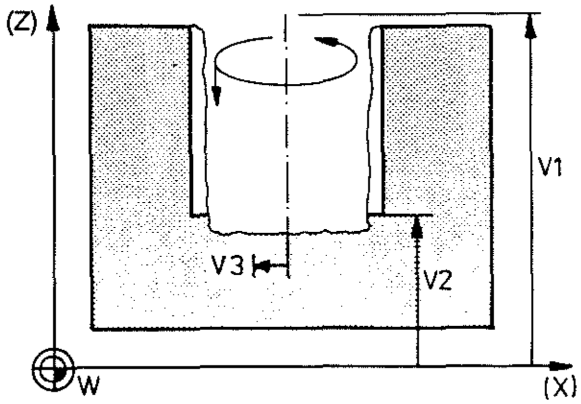
BORING G85

Definition

Boring a rough bore with a boring tool. Oriented spindle stop at the bottom of the bore with eccentric retract.

Input

change-over point	V1 mm abs.
depth of bore	V2 mm abs.
transverse movement at bottom of bore	V3 mm inc.



Sequence

spindle on

1 positioning axes drive to the centre of the bore in rapid; feed-in axis remains at traversing height

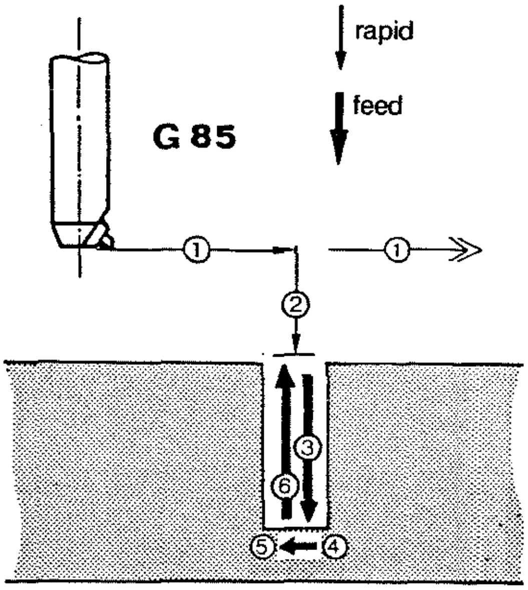
2 feed-in axis drives to V1; change-over to feed

3 feed-in axis drives to depth V2 in feed

4 oriented spindle stop, M19 at the bottom of the bore, angle = 0°

5 transverse movement of abscissa axis by distance V3 (negative axis direction)

6 eccentric retract of the feed-in axis to V1

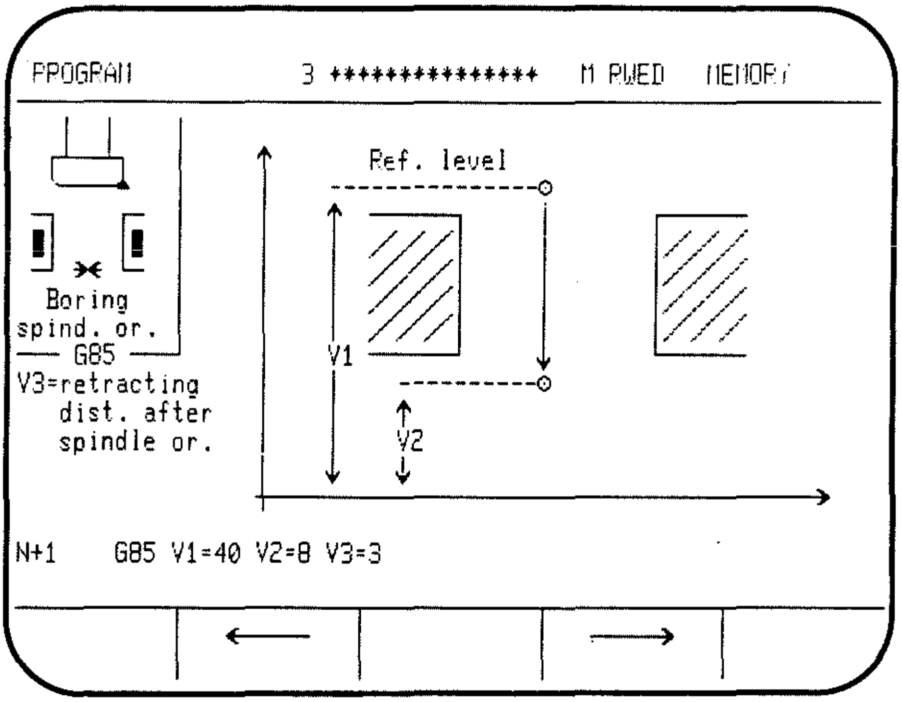


Condition

If cycle G85 is to be used an encoder is required to allow spindle orientation (M19); otherwise an error message is displayed.

Example

```
:
N9    T0101                                  tool selection
N10   F500  S250  M3                         conditions
N11   G85  V1=40  V2=8  V3=3                 call-up of cycle G85 and
                                             definition of variables
N12   X59  Y74.5                             machining positions
N13   X47  Y83
:
N19  G80                                     cancellation of cycle
N20
:
```



```
PROGRAM                3 ****************     M RWED    MEMORY
```

```
N+1     G85 V1=40 V2=8 V3=3
```

The following functions are activated:

M3 spindle rotation clockwise
F feedrate active before call-up
M19 spindle stop with orientation,
remains active after execution of cycle

REAMING G86

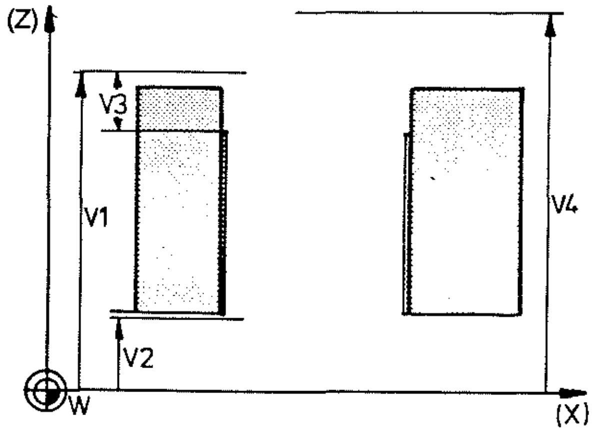
Definition

After the 1st CYCLE START the reaming bit is sunk into the workpiece for a short trial feed-in and then retracted to allow measuring.

From the second CYCLE START onwards the tool is driven to the full depth.

Input

change-over point	V1	abs.
machining depth	V2	abs.
Ist feed-in depth	V3	inc.
retract height for measuring	V4	abs.



Sequence

spindle on

1 positioning axes drive to the centre of the bore in rapid; feed-in axis remains at traversing height

2 feed-in axis drives to V1: change-over to feed

3 feed-in axis drives down by V3 to first feed-in depth in feed

4 in feed to change-over point V1

5 in rapid to retract height V4; spindle continues to rotate;

program stop, **M0 is active**

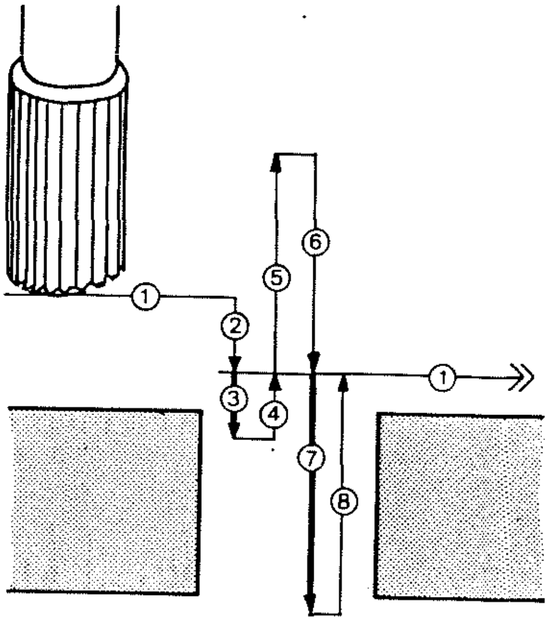
diameter of the bore can be measured, and the spindle speed corrected

After 2nd CYCLE START:

6 in rapid to V1

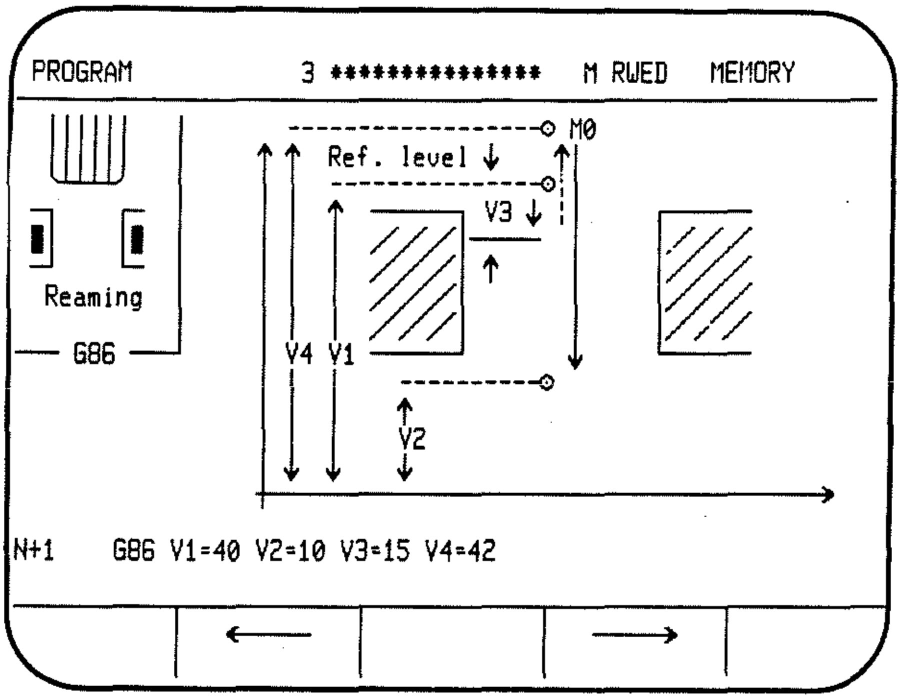
7 in feed to bottom of bore V2

8 in feed to change-over point



Example

:		
N9	T0101	tool selection
N10	F500 S250 M3	conditions
N11	G86 V1=40 V2=10 V3=15 V4=42	call-up of cycle G86 and definition of variables
N12	X97 Y102	machining positions
N13	X86 Y113	
:		
N19	G80	cancellation of cycle



N+1 G86 V1=40 V2=10 V3=15 V4=42

The following functions are activated:

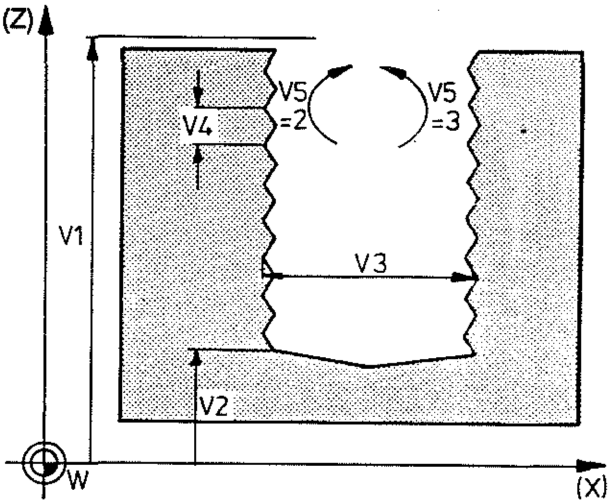
F = feedrate active call-up
S = old, possibly corrected spindle speed
M3
G0, which remains active after the execution of the cycle

THREAD MILLING G87

Definition A thread is cut by the helical motion of the tool.

Input

change-over point	V1	abs.
machining depth	V2	abs.
thread diameter	V3	abs.
thread pitch/rev.	V4	
right/left-hand thread:	V5	=2 right
	V5	=3 left



Sequence

spindle on

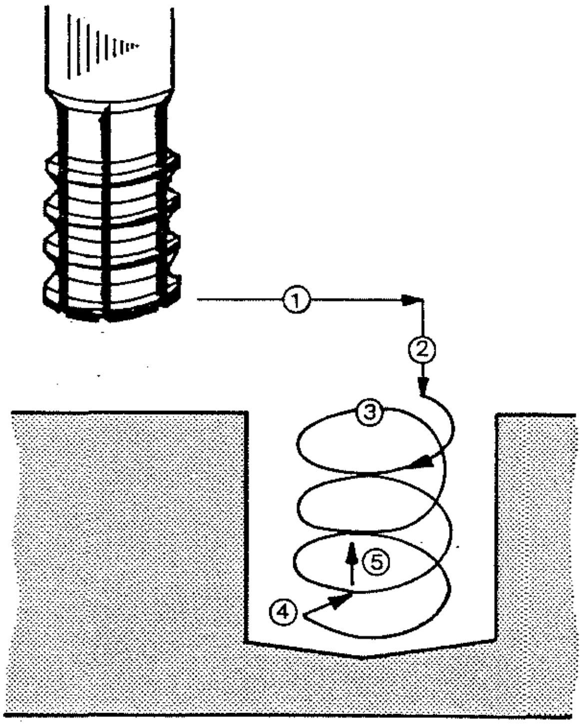
1 positioning axes drive to centre of bore in rapid; feed-in axis remains at traversing height

2 feed-in axis drives to V1; change-over to feed

3 helical interpolation in feed down to the bottom of the thread

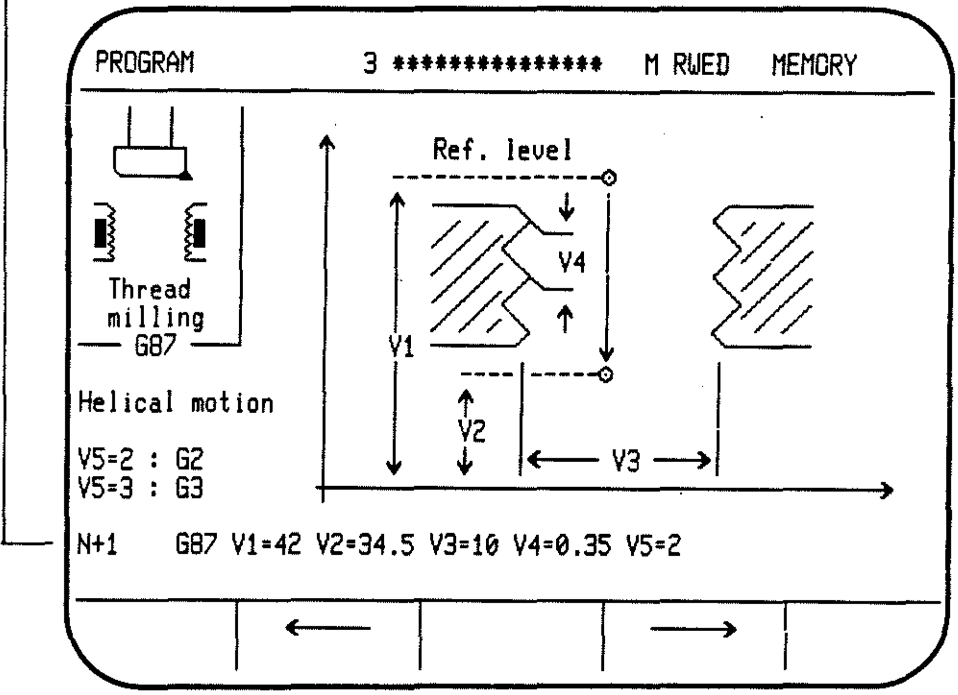
4 tool positioned to centre of bore

5 retract in rapid to V1



Example

```
:
N10   F500  S250  M3        T1212                               conditions
N11   G87  V1=42  V2=34.5   V3=10  V4=0.35  V5=2                call-up for
                                                                cycle G87 and
                                                                definition of
                                                                variables
N12   X44   Y24
N13   X32   Y26
:
N19   G80                                                       cancellation
                                                                of cycle
N20
```



PROGRAM   3 ****************   M RWED   MEMORY

N+1   G87 V1=42 V2=34.5 V3=10 V4=0.35 V5=2

DIMENSIONING

G90 A ABSOLUTE DIMENSIONS
G91 INCREMENTAL DIMENSIONS

Definition

Positions on workpiece contours can be defined with:

G90 absolute dimensions, i.e. all dimensional values relate to the active program zero point

or

G91 incremental dimensions, i.e. all dimensional values relate to the respective previous positions. It is advisable to use G91 for contours which need to be machined repeatedly in different places.

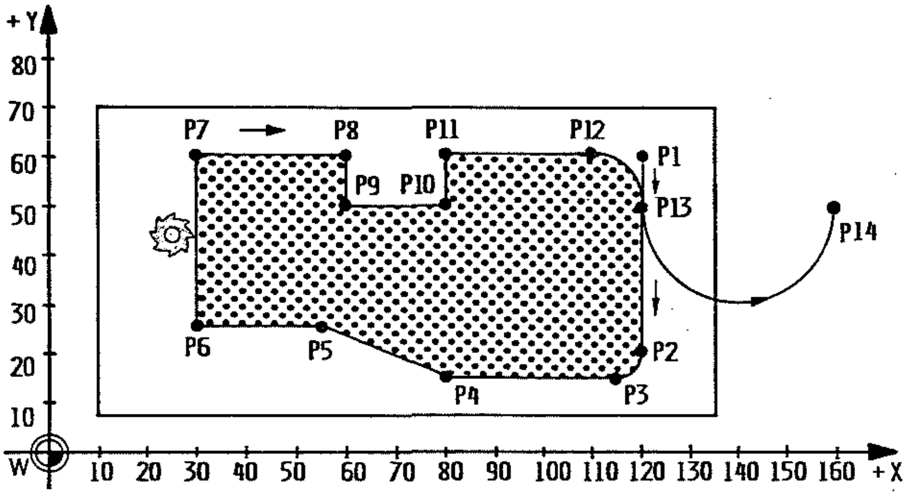
Operation

G90/91are modal and exclude one another.

They can be programmed with or without axis information.

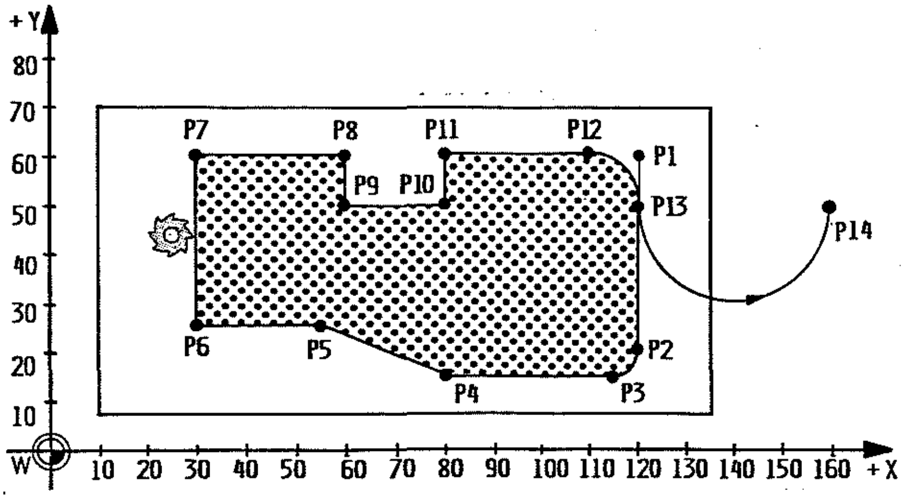
The axis displays are not influenced by these functions.
When G92 is cancelled G90 becomes active.

Example G90



```
program                        points in
                               drawing
N1    G90
N2    G0    X120    Y60     P1
N3    G1    Y20     F300    P2
N4    G5    X115    Y15     P3
N5    G1    X80             P4
N6    X55   Y25             P5
N7    X30                   P6
N8    Y60                   P7
N9    X60                   P8
N10   Y50                   P9
N11   X80                   P10
N12   Y60                   P11
N13   X110                  P12
N14   G5    X120    Y50     P13
N15   G5    X160            P14
N16   M2
```

Example G91 all pieces of axis information relate to the coordinates of point P1



```
N1    G90   X120    Y60     P1
N2    G91
N3    G1    Y-40    F300    P2
N4    G5    X-5     Y-5     P3
N5    G1    X-35            P4
N6    X-25  Y10             P5
N7    X-25                  P6
N8    Y35                   P7
N9    X30                   P8
N10   Y-10                  P9
N11   20                    P10
N12   Y10                   P11
N13   X30                   P12
N14   G5    X10     Y-10    P13
N15   X40                   P14
N16   G90
N17   M2
```

SETTING POSITION STORES G92

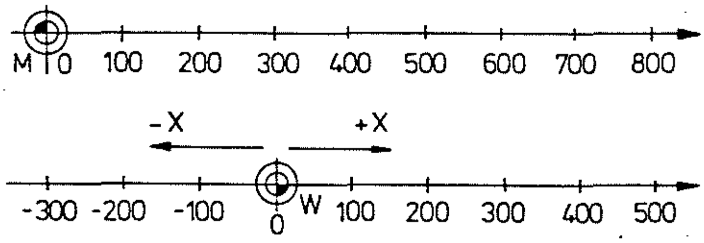
Definition
G92 X Y Z E

G92 is used to assign a new value to the position at which the axis stands, and to display this value. There is no axis movement involved.

G92

By programming G92 without axis values the machine coordinates are reactivated.

Example



machine coordinates without/before G92 (X Y Z E)

program coordinates after G92 (X Y Z E); programming:
G92 X-300

G92 S

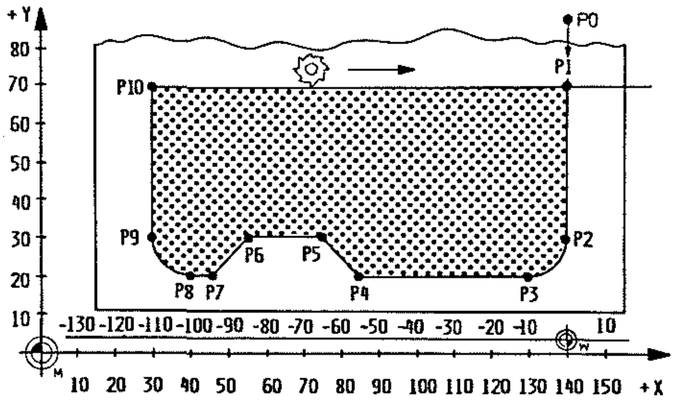
Setting of upper spindle speed limit.

Operation

Values can be set for up to 4 axes.
G92 can be used in MDI or in automatic.
G92 is active only in the block in which it is programmed.
To cancel G92 no other functions must be programmed in the same block as G92.
Any values within the input range can be used.
The travel limits determined by the hardware and software limit switches are not affected.

Programming	**G92 X.. Y.. Z.. E..**	assign axis values
	G92	reactivate machine coordinates
	G92 S..	limit spindle speed

Example



```
program                         points in
N1   G90  F200                  drawing
N2   G1   X140  Y70             P1
N3   G92  X0
N4   G1   Y30                   P2
N5   G5   X-10  Y20             P3
N6   G1   X-55                  P4
N7   X-65 Y30                   P5
N8   X-85                       P6
N9   X-95 Y20                   P7
N10  X-100                      P8
N11  G5   X-110 Y30             P9
N12  G1   Y70                   P10
N13  X0
N14  G92
N15  M30
```

Application
G92 X to E

G92 X to E is used to adapt a program compiled with dimensions from a drawing for a particular clamping position.

Comments/
Restrictions

G92 should **not** be used in conjunction with the following:

- active tool radius compensation **G41/42**
  active tool length compensations adress T

- active field limitation **G25/26**
  (G92 resets any field limitation)

In addition the following points must be taken into consideration:

- G92 does not take any active zero shift into account;
  when G92 is cancelled **G54 - G59** are reset as well.

- If while **G92 S..** is effective a spindle speed in excess of the limit is programmed elsewhere in the program the set max. value will be output with automatic gear range selection.

- When **G92 is cancelled** (G92 without axis or spindle values) G1 is automatically activated.

- G92 (without axis addresses) sets **G27**, cancelling any field limitation.

- When a **field limitation** is set with **G25, G26** any zero shift with G92 X.. Y.. is ignored.

-When G92 is cancelled **G90** is activated.

FEEDRATES

G94 A feedrate direct
G93 time programming

G94

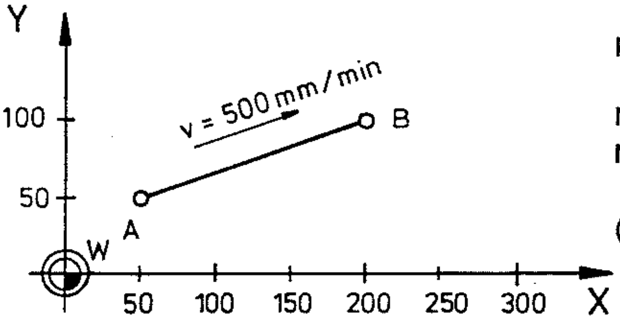
Direct specification of feedrate F in mm/min.
This type of feedrate output is acitve on switch-on.
Axis movement is possible with the spindle at standstill.

G94
Programming

G94 F... = feedrate in mm/min

With or without axis information. Feedrate F programmed with G94 is modal.

Example



program

```
N1   G0    X50   Y50          A
N2   G1    X200  Y100  F500   B
```

(G94 active on switch-on)

G93

Specification of the execution time for a path section.

The corresponding feedrate is calculated by the control.

Possible applications

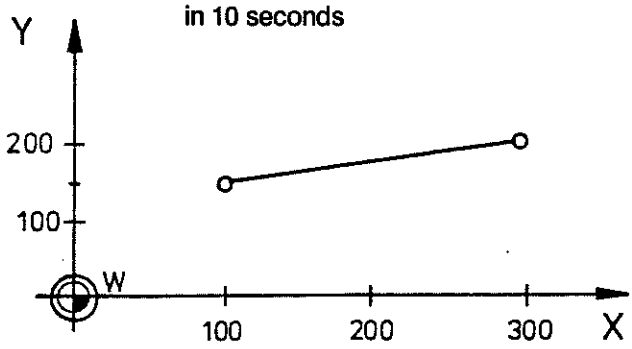
Complex movements involving more than one axis.
Simultaneous movement of linear axes and rotary axes.
Polygon contours for which block preparation time is of importance.

G93
Programming

G93 F... F = execution time in seconds

With or without axis information. The execution time F programmed with G93 only applies for the block in which it is programmed.

Example


execution of path section
in 10 seconds

program

```
N1   G0    X100  Y150
N2   G93   F10
N3   G1    X300  Y200
N4   G94
N5   M30
```

Operation

G93/G94/G95 exclude one another.

FEEDRATE MM/REV G95

Definition

Programmed feedrates relate to the speed of the main spindle.
The axis movements are derived from the actual spindle speed
and are therefore synchronised with the spindle.
The spindle speed determines the axis feedrate.

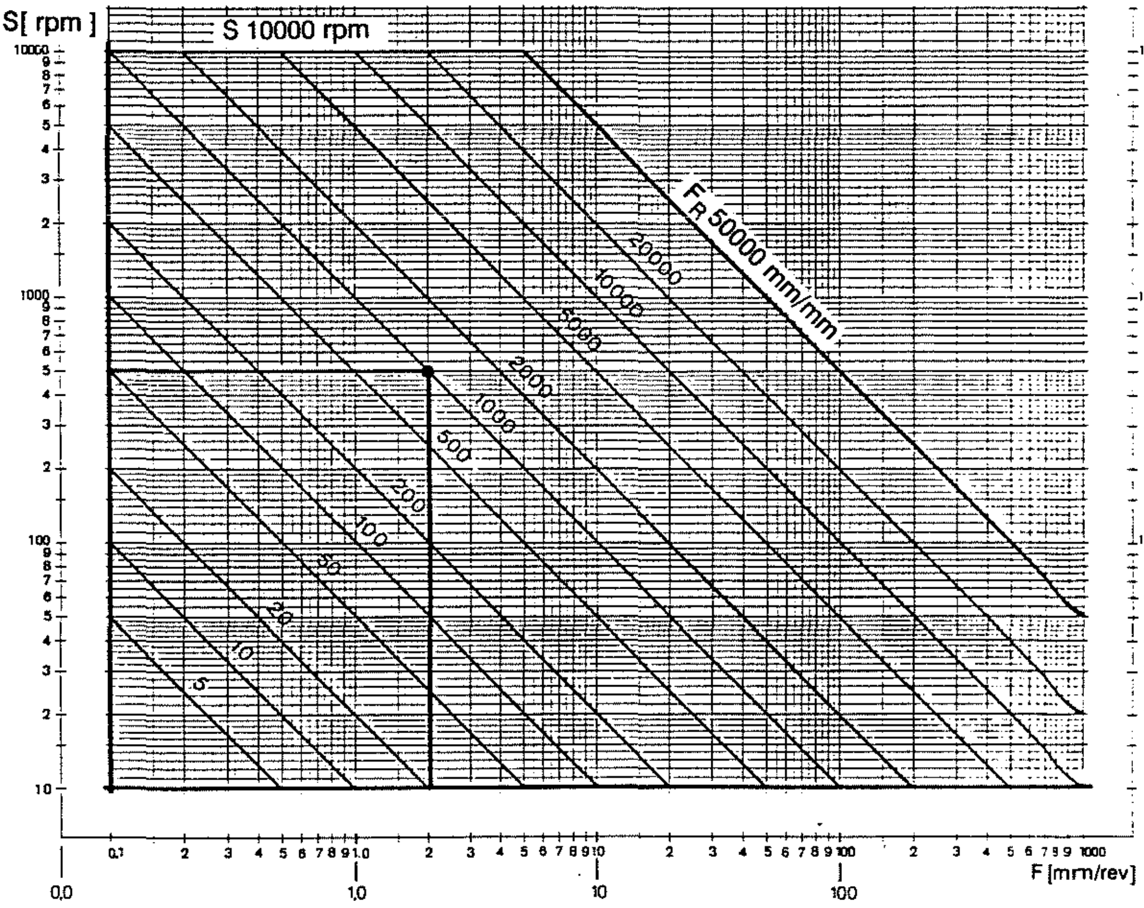
Programming Example

G95 F 1.67 Feedrate 1.67 mm/rev applies.
Feedrates of linear axes = mm/rev.
Feedrate of rotary axis = degrees/rev.

Resultant Feedrate F_R

F_R = feedrate in mm/min

F = feedrate in mm/rev.
S = spindle speed



Example: F = 2 mm/rev. and S = 500 rpm produce a resultant feedrate of 1000 mm/min.

Axis movements in feed only if the main spindle is running!

Note:

- Since the feedrate is derived from the actual spindle speed no axis movements are possible when there is a fault in the main spindle servo loop.
- Spindle speed output in BCD is not permitted.
- With G95 active no S-word is output via the BCD bus.

AUTOM. CALCULATION OF CUTTING SPEED IN M/MIN G96

Definition The control calculates the spindle speed from the data for cutting speed and tool radius stored in the technology store of the programmed tool.

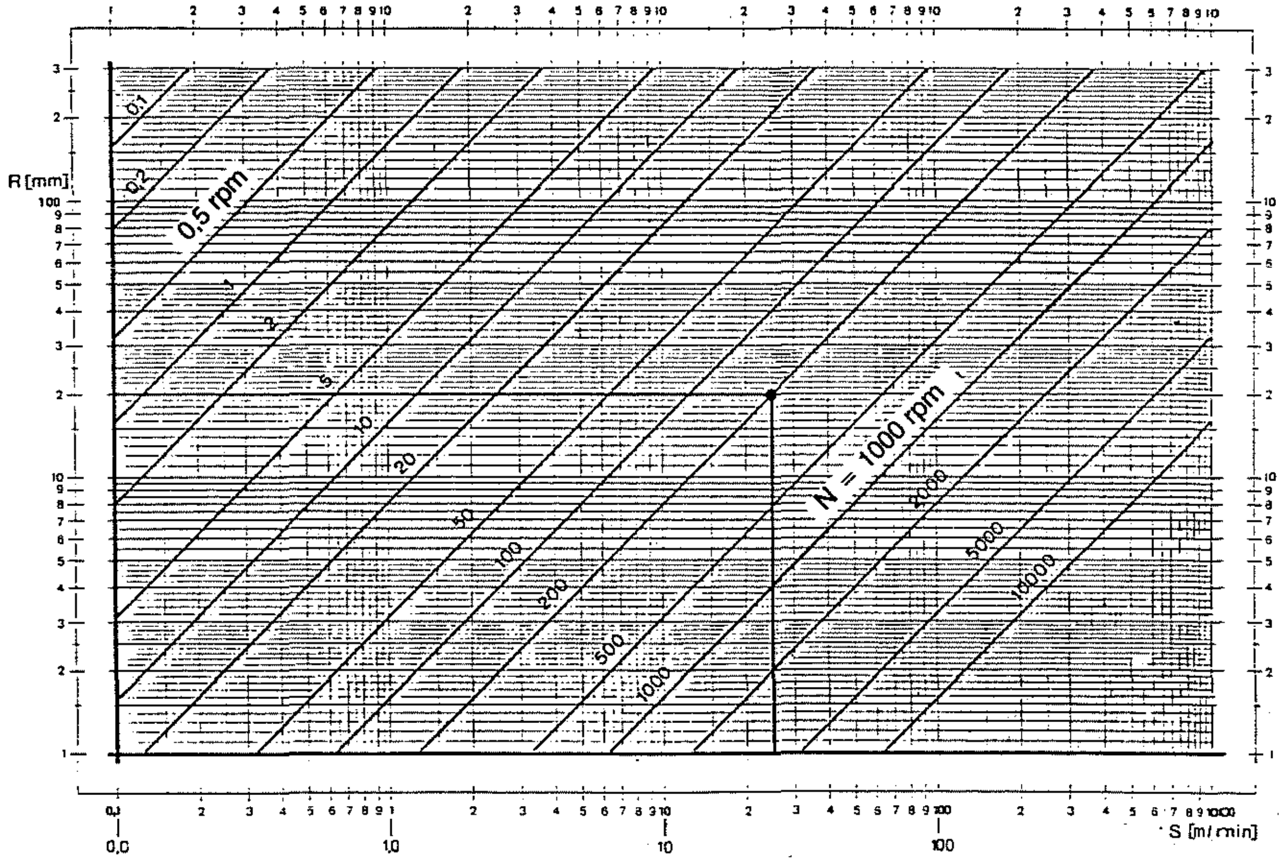
Operation With M3 or M4 active, the spindle speed is activated after the programming of a T-word. G96 and G97 cancel one another.

Programming On its own or together with other instructions.

Spindle speed $n = \frac{S}{2 . \pi . R}$ formula for calculating the spindle speed

R = tool radius (in tool table)

S = cutting speed "



Example: A cutter radius of 20 mm plus a required cutting speed of approx. 25 m/min will result in the output of 200 rpm for the spindle speed.

Usage Before the call-up fo G96 a starting speed can be selected (with G97). G96 must be cancelled before M5 (G97).

Note It is important for the calculations that the tool radius is defined in mm in the tool table.
The cutting speed is specified in mm/min.
The tool wear compensation (DR-value in tool table) is not taken into account in the spindle speed calculation.

SPINDLE SPEED DIRECT G97 A

Definition The speed of the main spindle is determined with S directly in rpm.
The spindle speed does not influence any axis movements.
G97 is active on switch-on.

Programming

G97	no change in the spindle speed
G97 S...	new spindle speed S... is activated

Overriding the programmed spindle speed:

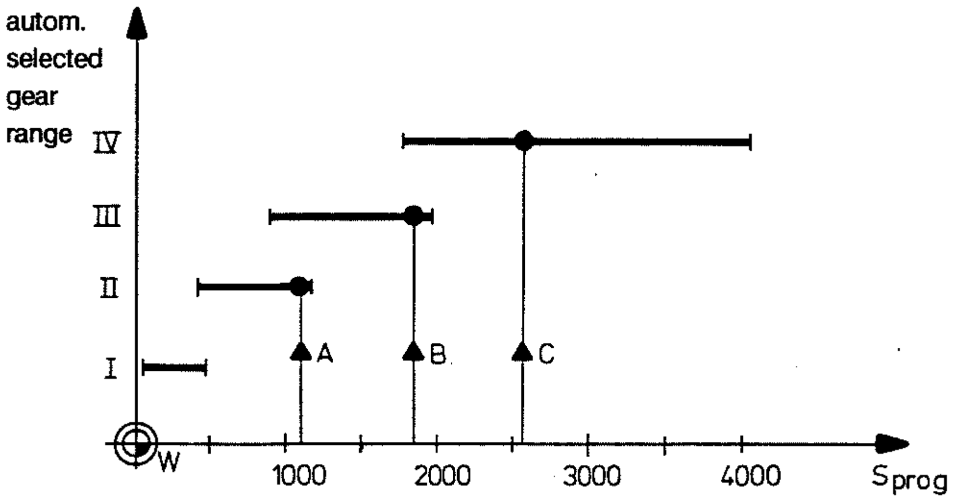
G66	overriding is possible via the potentiometer on the manual panel
G63	the override potentiometer is not effective

Effect of gear range selection:

M40	automatic recognition and output
M41-44	direct programming of the gear range

see also	SPINDLE SPEEDS	chapter 3	ADRESS S
	GEAR RANGES		
	M-FUNCTIONS		

M40 Automatic gear range selection and speed ranges for the individual gear ranges:



Selected gear ranges when different speeds are programmed:

A: gear range II
B: gear range III
C: gear range IV

With speeds at which two gear ranges overlap the lower gear range (higher motor speed and higher torque) will be output.

SUBPROGRAM END G99

Definition

G99 designates the end of a subprogram.
G99 is the instruction to jump back within the program from which the call-up was made to the position at which the subprogram was called up.
The next program block will then be executed.

Programming

G99 without any other instructions.

Example

```
                 beginning of main program 5
N1 .
N2 .
N3   G22  P15    call-up of subprogram 15
.
.
N20  G22  P12    call-up of subprogram 12
.
.
N37  G22  P20    call-up of subprogram 20
N38 .
N39 .
N40  M2          main program end

                 (there must be M2/M30 between the
                  main program and the associated subprograms!)

N41  $15         beginning of subprogram 15
N78 ...
N79  G99         end of subprogram 15

N80  $12         beginning of subprogram 12
N115...
N116 G99         end of subprogram 12

N117 $20         beginning of subprogram 20
N207...
N208 G99         end of subprogram 20
```

Note

The program from which the call-up is made can be a main program, a subprogram or a cycle.
Maximum nesting depth is 10 (see under G21, G22).

THREE-DIGIT G-CODES

G800 to G869

Definition

The control operates with 3-digit G-codes.
The functional content of these codes must be defined by the machine tool builder or the user himself.

Programming

The machining sequence is programmed as a cycle.
Both the standard instructions as well as the parametric functions can be used to program these cycles.

Application examples:

Machine specific operations such as
delivery and removal of workpieces
measuring, spot checks
tool inspection
punching/nibbling cycles
control of auxiliary machinery

Simplification of programming by the use of cycles for

the firm's own particular methods
for the machining of bores,
of standard parts, of part families,
for calculations,
for the adaptation of the CC 100 to
special machines.

Cycle Numbers and Call-up

Programming	Call-up
cycle	G-function
1	G 801
.	.
.	.
69	G 869

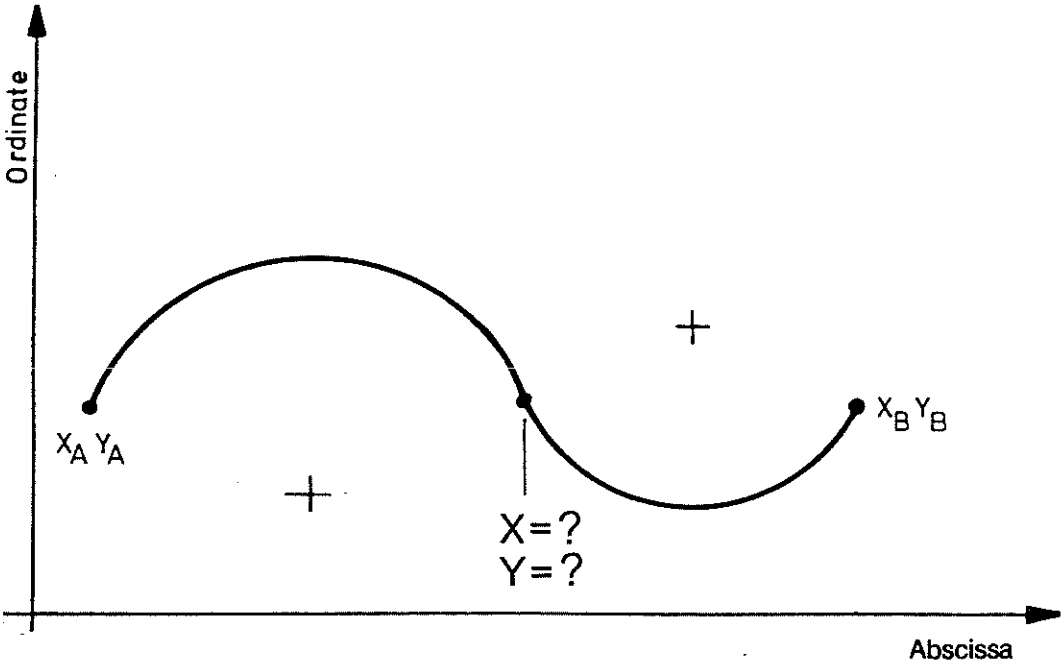
Example

A machining cycle written under cycle 45 is called up by G845.

CONTOUR CYCLES

G890 to G898

These 3-digit G-codes calculate positions which might not be provided on the drawing in all 3 main planes.



The control automatically makes the correct allocation of entered abscissa and ordinate values to the relevant axes, dependent on the plane selection.

Axis Allocation

	G17	G18	G19
abscissa A	X	Z	Y
ordinate 0	Y	X	Z

Execution

Cycles G890 and G891 are pure calculating cycles. The results obtained by calling them up can then be used in the course of the part program.

Cycles G892 to G898 process the values by executing the contour.

Call-up of Contour Cycles in a Program

Operating Sequence

Main mode EDIT

TOOLS	ZERO SHIFTS	VARIABLES	PROGRAMS	CYCLES

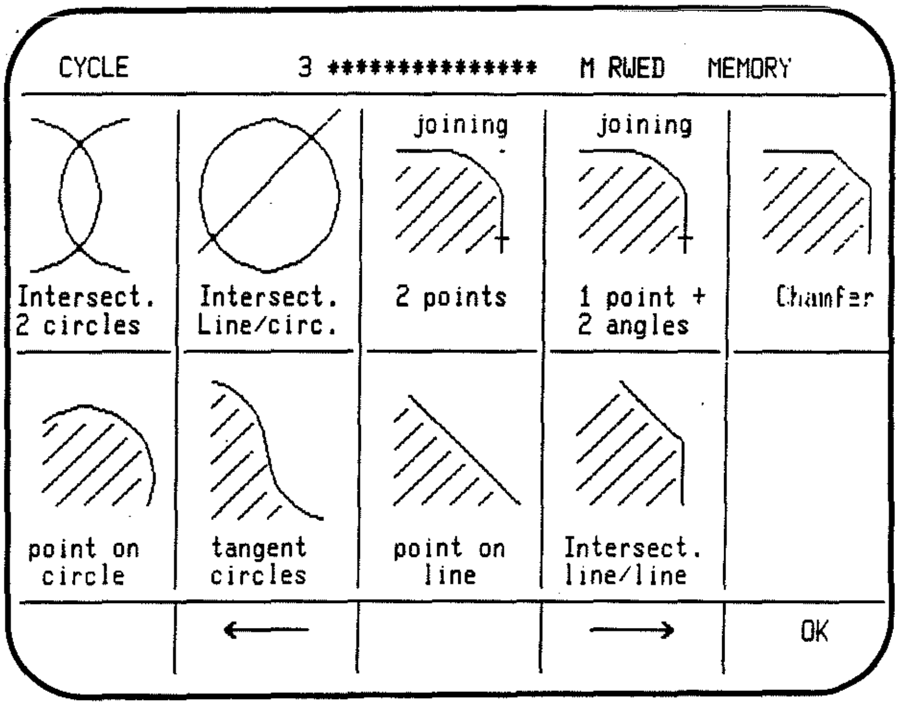
—> program call, e. g. [9] [ENTER]

COMMAND	NEXT PAGE	EDIT	LOAD	SAVE

SEARCH GRAPHIC	↱	SCROLL	↱	MODIFY INSERT

			BORING CYCLES	CONTOUR CYCLES

⇓

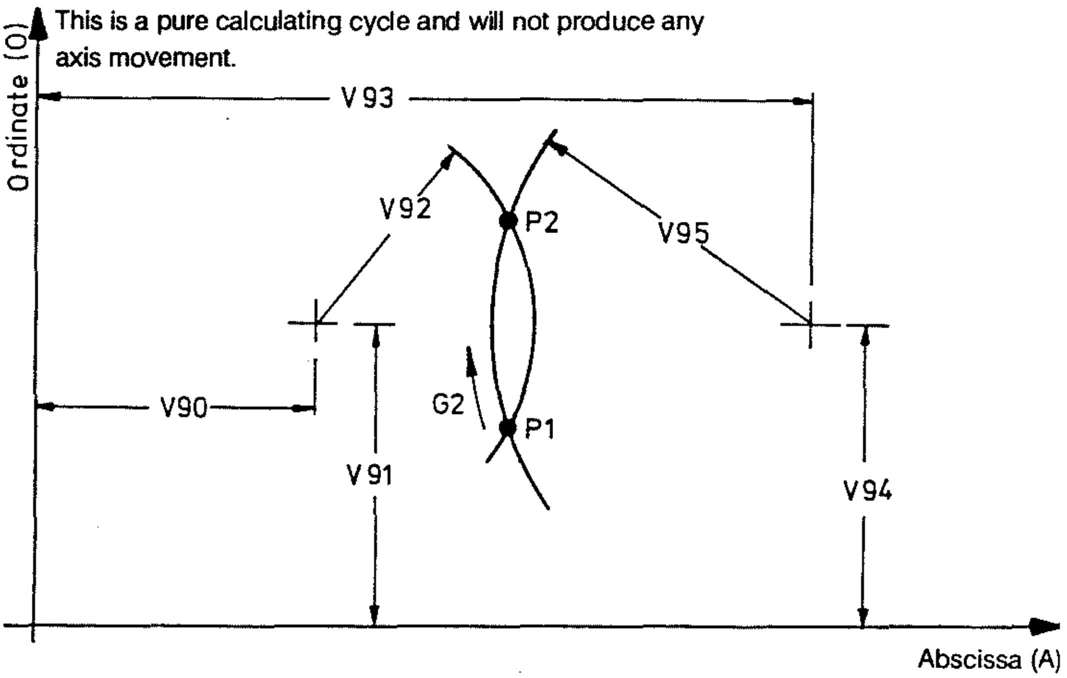


The required cycle is to be selected with the [←] or the [→] key and confirmed with [ok] .

Once the variables have been defined and the cycle input confirmed with [ENTER] the contour cycle will be stored in the program.

INTERSECTION CIRCLE/CIRCLE G890

Definition This is a pure calculating cycle and will not produce any axis movement.



Input

G890		G17	G18	G19
V90 = A	1st centre	X	Z	Y
V91 = 0		Y	X	Z
V92 = 1st	radius			
V93 = A	2nd centre	X	Z	Y
V94 = 0		Y	X	Z
V95 = 2nd	radius			

Results After the call-up the contents of the variables are as follows:

intersection P1	V90 = abscissa	V91 = ordinate
intersection P2	V93 = abscissa	V94 = ordinate

Position of the intersections, looking from the first to the second centre of the circle:

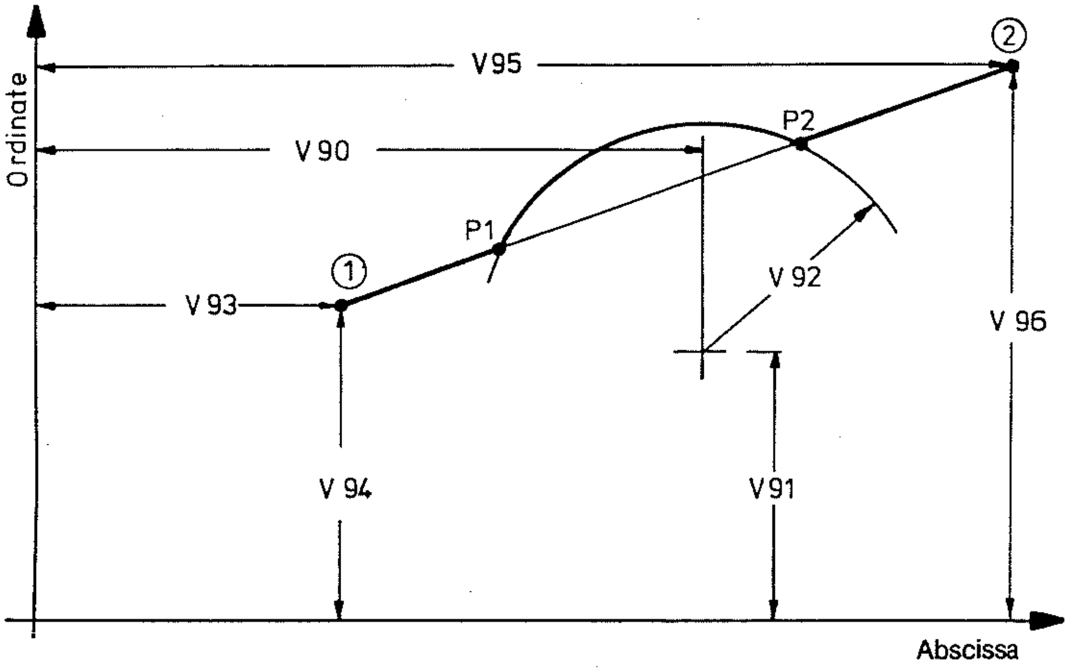
P1 lies to the right of the connecting line
P2 lies to the left of the connecting line

Example The calculated points P1/P2 could be used as follows:

```
N1   G0    X60   Y85
N2   G890 V90=75  V91=90  V92=5  V93=82  V94=100  V95=8.5 cycle call-up
N3   G1    F750
N4   X=V90       Y=V91                          allocation of variables
N5   G2
N6   X=V93       Y=V94     R=V95                allocation of variables
N7   M30
```

INTERSECTION LINE/CIRCLE G891

Definition This is a pure calculating cycle and it will not produce any axis movement.



Input

G891		G17	G18	G19
V90 = A	centre of	X	Z	Y
V91 = 0	circle	Y	X	Z
V92 =	radius			
V93 = A	for point	X	Z	Y
V94 = 0	①	Y	X	Z
V95 = A		X	Z	Y
V96 = 0	②	Y	X	Z

Results After the call-up the calculated values will be stored in variables as follows:

intersection P1	V90 = abscissa	V91 = ordinate
intersection P2	V93 = abscissa	V94 = ordinate

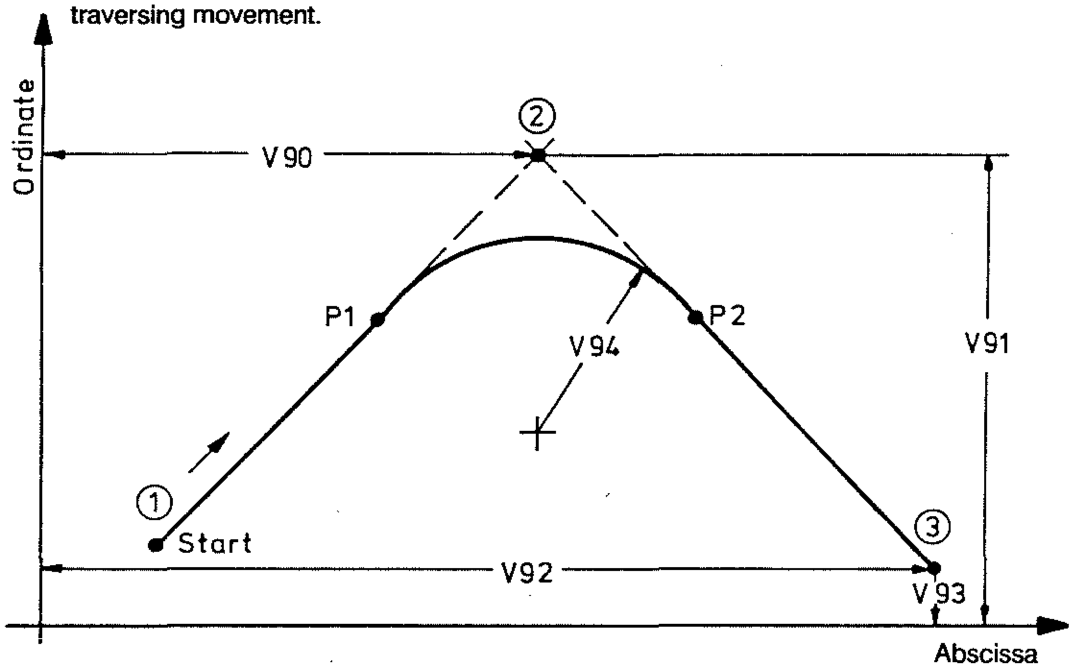
Application The calculated points P1/P2 can be used as follows:

```
N1    G0    X0    Y0
N2    G891 V90=40 V91=30 V92=15 V93=15         cycle call-up
      V94=30 V95=100 V96=45
N3    G1    F750
N4    X=V90      Y=V91                          allocation of variables
N5    G2
N6    X=V93     Y=V94     R=V92                 allocation of variables
N7    M30
```

ROUNDING CORNERS (3 POINTS) G892

Definition

Calculation of positions to round corners with 3 known points. This cycle calculates the positions and then initiates the traversing movement.



Input

G892

① starting point = last programmed position

		G17	G18	G19
② intersection	V90 = A	X	Z	Y
	V91 = 0	Y	X	Z
③ end point	V92 = A	X	Z	Y
	V93 = 0	Y	X	Z

V94 = radius
V95 = 0 = machining up to P2
1 = machining up to end point ③

End Point of Machining

The end point of the machining is determined by the input for V95. When several contour cycles are linked together the previous contour cycle must not be machined up to the end point.

Results

1st transition point P1 V96 = abscissa V97 = ordinate
2nd transition point P2 V98 = abscissa V99 = ordinate

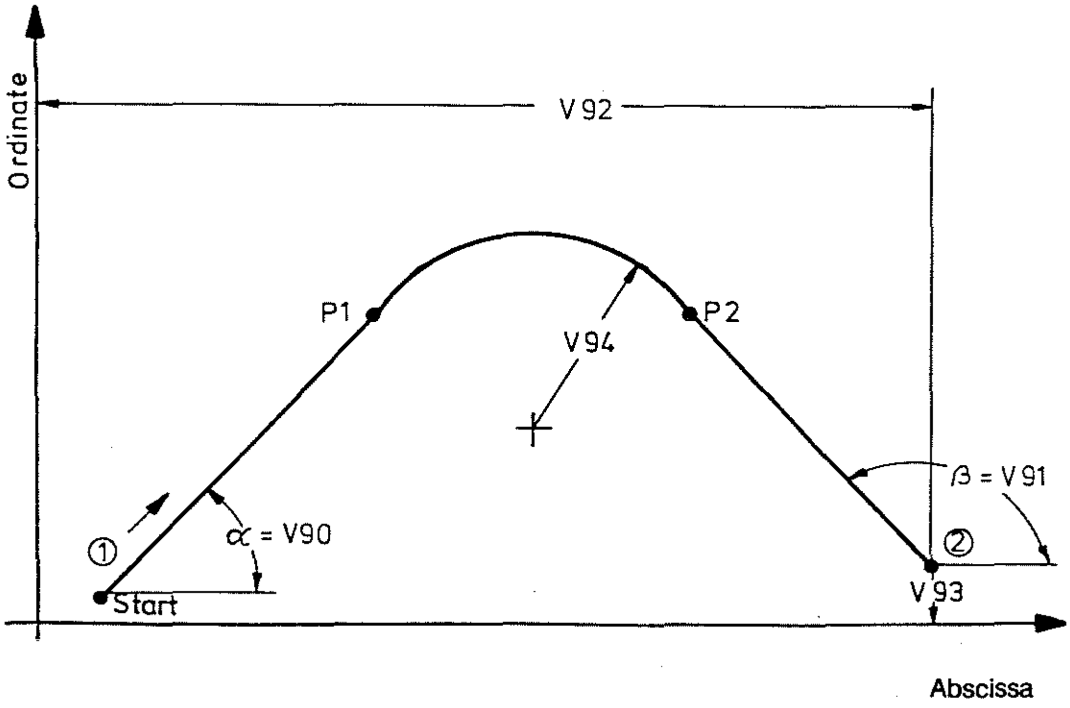
Example

```
N1   G1   X20   Y20   F100                     starting position
N2   G892 V90=55 V91=60 V92=100 V93=30 V94=20 V95=1 cycle call-up
                                                + variable allocation
N3   X110   Y65
N4   M30
```

ROUNDING CORNERS (2 ANGLES) G893

Definition

Calculation of positions to round corners with known angle values.
The cycle calculates the positions and initiates the traversing movement.



Input

G893

starting point① = last programmed position

angles V90 = alpha V91 = beta
- input range -180° to + 180°
- sign determines direction of rotation
- the abscissa is the reference axis

end point ②

	G17	G18	G19
V92 = A	X	Z	Y
V93 = 0	Y	X	Z

V94 = radius for arc
V95: 0 machining up to point P2
1 machining up to end point

End Point of Machining

The end point of the machining is determined by the input for V95. When several contour cycles are linked together, the previous contour cycle must not be machined up to the end point.

Programming

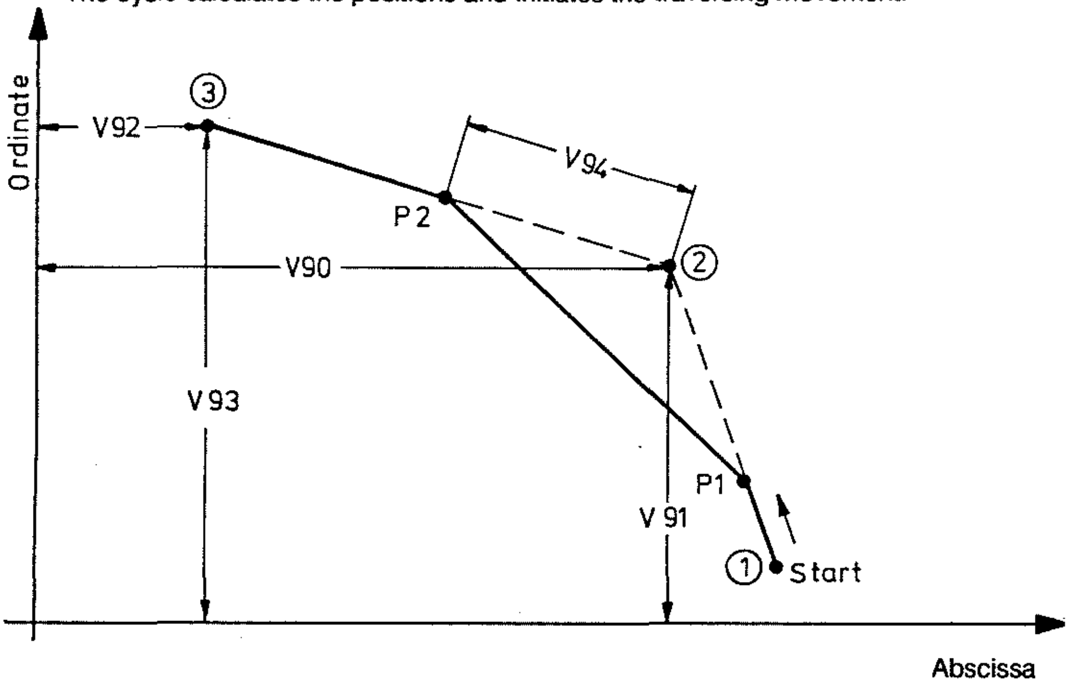
G893 V90 = . . . V91 = . . . V92 = . . . V93 = . . . V94 = . . . V95 = . . .

Results

1st transition point P1	V96 = abscissa	V97 = ordinate
2nd transition point P2	V98 = abscissa	V99 = ordinate

CHAMFERING G894

Definition Calculation of positions to apply chamfers to straight contour elements.
The cycle calculates the positions and initiates the traversing movement.



Input

G894
starting point 1 = last programmed position

			G17	G18	G19
intersection	2				
		V90 = A	X	Z	Y
		V91 = 0	Y	X	Z
end point	3	V92 = A	X	Z	Y
		V93 = 0	Y	X	Z

V94 = length of chamfer
V95: 0 = machining up to point P2
1 = machining up to end point

End Point of Machining The end point of the machining is determined by the input for V95. When linking several contour cycles the previous contour cycle must not be machined up to the end point.

Results

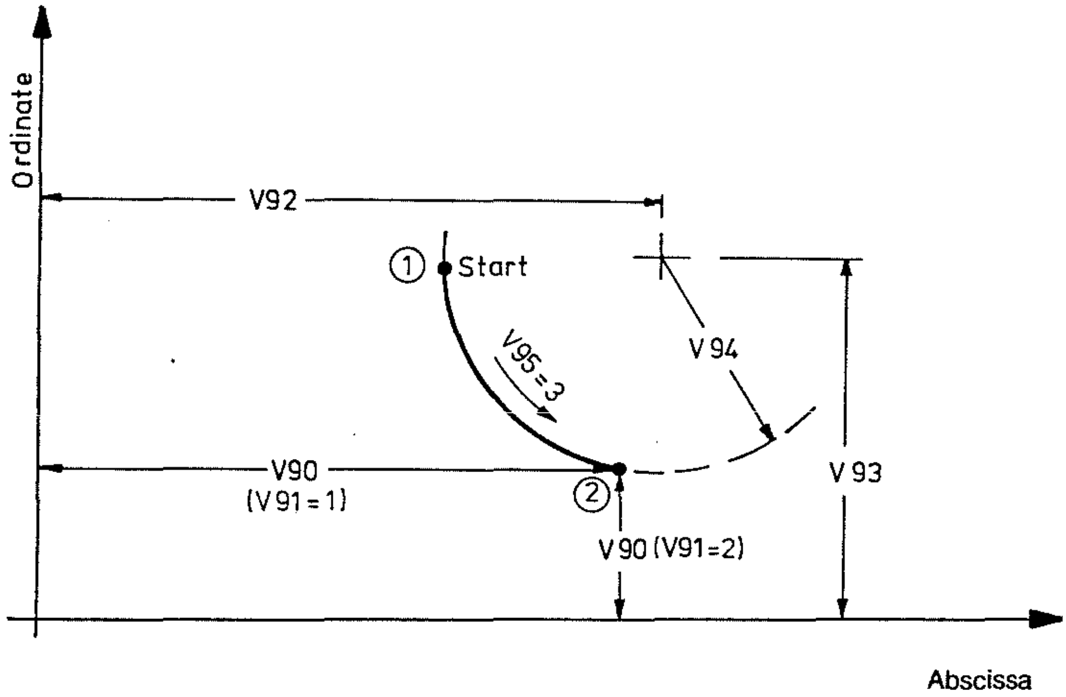
1st transition point P1	V96 = abscissa	V97 = ordinate
2nd transition point P2	V98 = abscissa	V99 = ordinate

Example

```
N1   G1   X80   Y5   F1000                        starting position
N2   X50                                          last contour point before call-up
N3   G894 V90=50 V91=40 V92=10 V93=45 V94=10 V95=1
                                                  cycle call-up and variable allocation
N4   M30
```

CALCULATION OF THE END POINT OF AN ARC G895

Definition Calculation of the end point of an arc, of which only one coordinate is known. The cycle calculates the position and initiates the traversing movement if COND. SBP CALL-UP is high.



Input G895

starting point ① = last programmed position .

end point ② V90 = A or 0-value
V91 = 1: V90 represents abscissa
V91 = 2: V90 represents ordinate

centre of circle V92 = A
③ V93 = 0
V94 = ±radius (+) larger/equal 180°
- smaller 180°
V95 = direction of rotation = 2 : G2
= 3 : G3

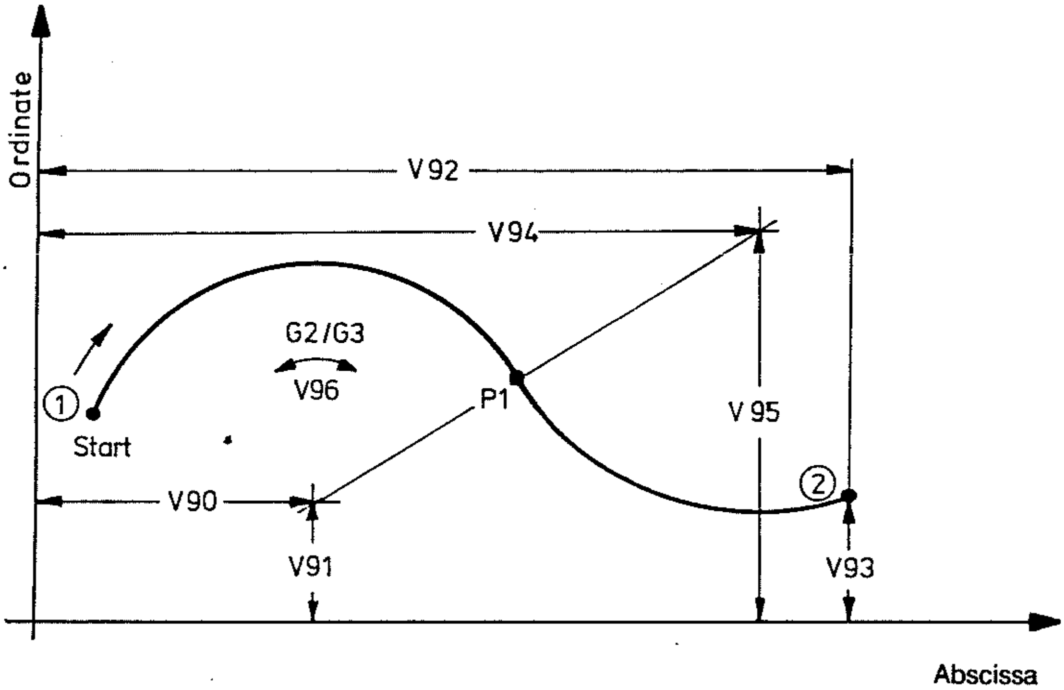
Programming G895 V90 = . . . V91 = . . . V92 = . . . V93 = . . . V94 = . . . V95 = . . .

Results end point ② V 90 = abscissa V91 = ordinate

The missing coordinate value of the end point is calculated.

TRANSITION POINT ARC/ARC tangential G896

Definition The control calculates the transition point of two consecutive arcs with tangential transition and a reversal of the direction of rotation. The cycle calculates the positions and initiates the traversing movement.



Input G896

starting point ① = last programmed position

1st centre of circle V90 = A V91 = 0
2nd centre of circle V94 = A V95 = 0

end point ② V92 = A V93 = 0

direction of ratation V96 = 2 corresponds to G2/G3
V96 = 3 corresponds to G3/G2

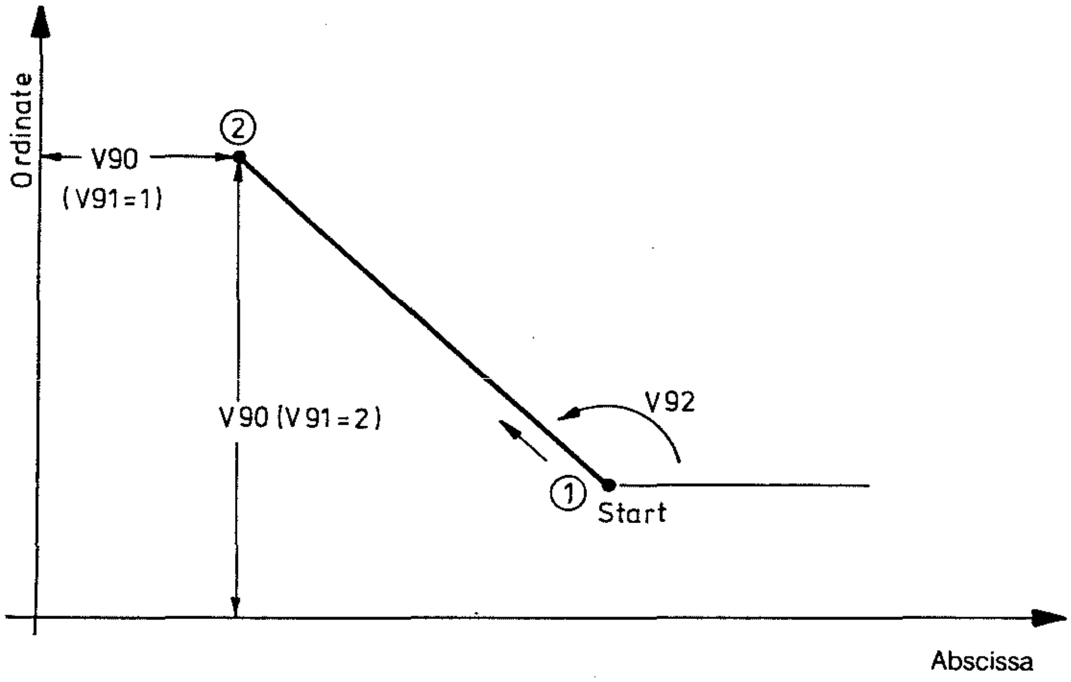
Programming G896 V90= . . . V91= . . . V92=. . . V93=. . . V94=. . . V95=. . . V96. . .

Results Transition point P1 V97 = abscissa V98 = ordinate

CALCULATION OF THE END POINT OF A STRAIGHT LINE G897

Definition Calculation of the end point of a straight line, of which only one coordinate is known.

The cycle calculates the positions and initiates the traversing movement.



Input **G897**

starting point ① = last programmed position

angle V92 = ± 180

end point ② V90 = A or 0-value
V91 = 1: V90 represents abscissa value
V91 = 2: V90 represents ordinate value

Programming G897 V90=... V91=... V92=...

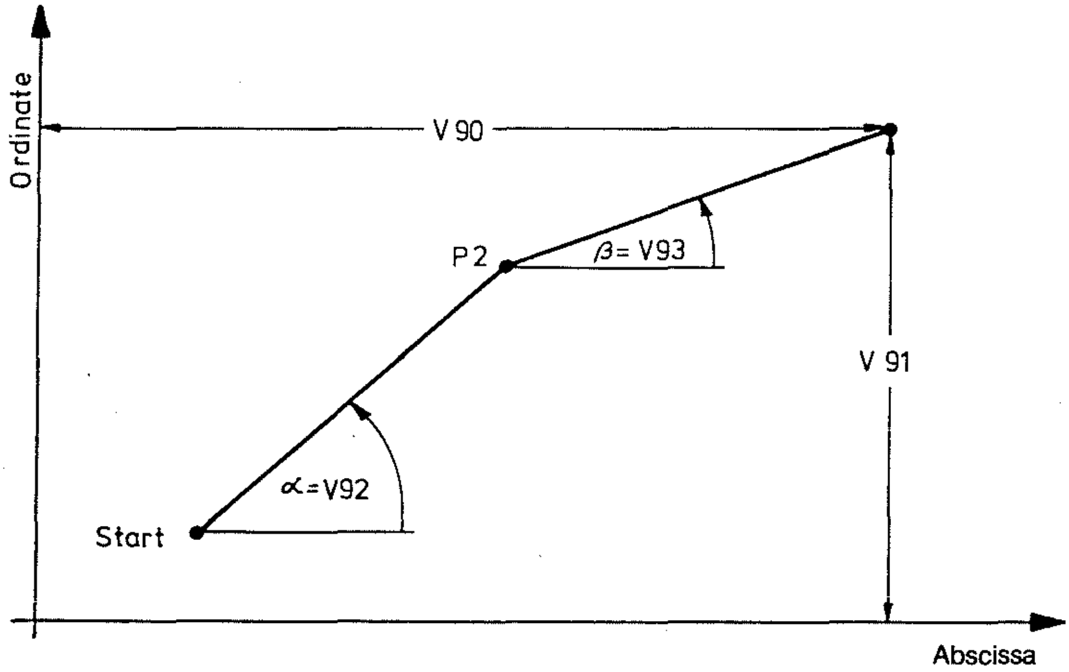
Results The unknown coordinate of the end point is calculated, after which the contents of the variables will be as follows:

V90 = abscissa value
V91 = ordinate value

INTERSECTION OF TWO STRAIGHT LINES G898

Definition Calculation of the intersection of two straight lines from the entered angle values.

The cycle calculates the positions and initiates the traversing movement.



G898

Input

starting point	= last programmed position	
end point	V90 = abscissa	V91 = ordinate
angles	V92 = alpha	V93 beta

- input range -180° to + 180°
- sign determines direction of rotation
- the abscissa is the reference axis

V94 = 0: machining up to P2
= 1: machining up to end point

End Point of Machining The end point of the machining is determined by the input for V94. When several contour cycles are linked together the previous contour cycle must not be machined up to the end point.

Programming G898 V90=. . . V91=. . . V92=. . . V93=. . . V94=. . .

Results The position of the intermediate point P2 is calculated and the axes drive to this position; the values are stored in the following variables:

V95 = abscissa value
V96 = ordinate value

SURVEY OF FIRMLY ALLOCATED CYCLES

	Function	Programmed under cycle, main mode	Call-up via
User cycles	freely programmable	1 - 69	G8nn
MTB cycles	priority routine	74	interface signal fast input on SERVO card
	MTB cycle	75 "	M22
	MTB cycle	76 "	M21
	MTB cycle	77 "	M6
	allocation of functions for keys F1 to F10 of customer keypad	78 "	customer keys
	referencing cycle	79 "	soft key selection

Cycles 1 - 69 are available for use by the enduser, unless predetermined by the MTB. These cycles can be used to program recurring machining tasks.
A cycle with the number nn is called up with G8nn. Input variables can be written together with the 3-digit G-code, for instance:

G824 V1 =. . . V10 =. . . V55 =. . . (call-up for cycle 24)

Cycles 70 - 73 are routines which are used internally by the control and which have fixed functions. They are not available for use by the enduser.

4. PARAMETRIC FUNCTIONS

V 15 = ATG VX

Range The following functions are available:

load instructions for numerical values, 125 variables V1 to V99, VA to VZ, basic arithmetic functions, trigonometric functions, copy instructions, logic operations, branching, access to NC data.

The user can write his own cycles with parametric functions.
CPC = Customer parametric Cycle
A CPC represents the solution of a problem in principle.
Values such as spindle speed, dimensions, tool no. etc. are kept variable.

Once the parametric program has been produced the only actions necessary for the execution are to load values for the variables and call up the program.

Applications Production of customer's own cycles for:

automatic measuring cycles with calibration of the probe,
measuring of the workpiece, and automatic tool wear compensation

production counters, random sample counters

scale factors for similar parts,

variable programs of all types

Programming During **panel input** the CPC key is pressed before the input of a computing function. This automatically activates the secondary function (inscribed at the top) of the dual function keys.

During **external programming** the mnemonic codes used by the control when printing out parametric instructions must be used to write the program.

Example: load variable 5 with the content of variable 2 + value 10

To store	Q V	BLT 5	CPC	= E	C V	BRA 2	CPC	+ G	COR 1	BGT 0	Enter
To execute	Q V	BLT 5	CPC	= E	C V	BRA 2	CPC	+ G	COR 1	BGT 0	START

External programming N12 V5 = V2 + 10 (**Note** Only whole numbers are accepted)

One program line can contain several computing functions. They will be executed in the same sequence in which they were written.

Example: V17 = V2 * V3 V25 = SIN V17 V26 = COS V17

Note
- The programming in each line must be either all conventional or all parametric.
- Parametric functions must always be programmed without space characters, e.g. ATG VX, in order to avoid syntax errors.

Program Planning

Before starting to produce programs it is advisable to do some general program planning. This should take the following points into consideration:

- Is a program to be used completely independently?
- Or is the program to be used in conjunction with other program modules? If so, with which ones?
- Is the program to be produced as a main program, a subprogram or a cycle?
- Which other programs must/can be stored in the memory at the same time?
- Which variables will be used?

Aims

Simplification of the continuing program administration.

- Rationalized program production
- Problem-free combination of programs
- Multiple use of program modules

FORMS

The following forms help with program planning:

- Memory Allocation
- General Progam Planning
- Variables
- Program Description

Memory Allocation

This form shows **which programs, cycles, subprograms etc. are stored in the control together.**

Program Planing

This form shows at a glance **which variables are used by which program,** and which are still available to be used.

Variables

This form can be used when **testing** programs, by tracking the meaning and the contents of the variables.

Program Description

This is an aid for the **program user,** and it should consist of at least a top sheet with

- a sketch of inputs/possibly the sequence
- required storage capacity, short functional description

MEMORY ALLOCATION

	name	no.	function, sequence	required storage capacity
program				
assigned subprograms (local)				

	name	no.	function, sequence	required storange capacity
program				
assigned subprograms (local)				

	name	no.	function, sequence	required storage capacity
cycle (global)				

PROGRAM PLANNING

PROGRAM / CYCLE

SBP

function

Cycle

function

VARIABLES V

1 2 3 4 5 6 7 8 9

1 2 3 4 5 6 7 8 9

1 2 3 4 5 6 7 8 9

1 2 3 4 5 6 7 8 9

VARIABLES (global)

L = loaded value
C = constant
Ca = calculated value (temporary)

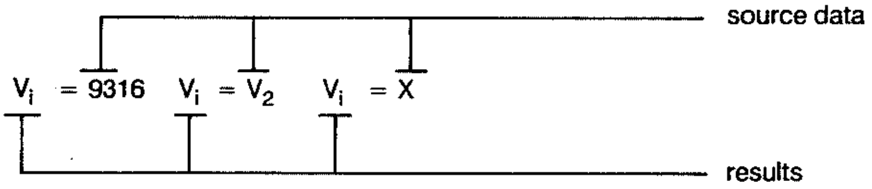
function	V	value	function	V	value
	0			0	
	1			1	
	2			2	
	3			3	
	4			4	
	5			5	
	6			6	
	7			7	
	8			8	
	9			9	
	0			0	
	1			1	
	2			2	
	3			3	
	4			4	
	5			5	
	6			6	
	7			7	
	8			8	
	9			9	
	0			0	
	1			1	
	2			2	
	3			3	
	4			4	
	5			5	
	6			6	
	7			7	
	8			8	
	9			9	
	0			0	
	1			1	
	2			2	
	3			3	
	4			4	
	5			5	
	6			6	
	7			7	
	8			8	
	9			9	

LOAD FUNCTIONS

Load V_i directly with numerical value	**V1 = 9316**
Load V_i with content of a variable (copy)	**V1 = V2** or **V1 = V2 + V15** **V1 = V2 - 4**
Load V_i with content of an NC address	**V1 = X**

Definition The variables to the left of the equal sign are loaded from the sources written on the right.

Programming



i from 1 to 99 and from A to Z

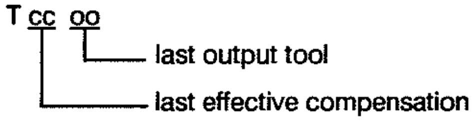
Several of these functions can be written into the same line.

Example N1 V12=1.6 V3=V5 V4=Z

Execution When N1 is carried out the programmed variables are loaded one after the other.

The sequence in which the variables are written determines the order of execution.

NC address values which can be loaded:

address	loaded value corresponds to:
X Y Z E A D R I J K	absolute positions in the active type of dimension in the machine coordinates or relating to the zero point set with G92
T	T is loaded with 4-digits 
F	feedrate in the active type of dimension as defined by G94 / 95 / 96
S	spindle speed or cutting speed as defined by G96 / 97

ARITHMETIC FUNCTIONS

Addition	V1 = V2+V3	
	V1 = V2+157	*)
Subtraction	V1 = V2-V3	
	V1 = V2-157	*)
Multiplication	V1 = V2*V3	
	V1 = V2*157	*)
Division	V1 = V2/V3	
	V1 = V2/157	*)
Square root	V1 = SQR V2	**)

Definition Arithmetic functions, using the contents of variables or direct numerical values.

Programming VN = VM + VP   V1 = VN * 12 *)   VJ = SQRV1 **)

Example Finding the square roots of a quadratic equation in a program line.

$$X_{1/2} = -\frac{P}{2} \pm \sqrt{\left(\frac{P}{2}\right)^2 - q}$$

with P = V1
q = V2

V4 = V1 / 2   V3 = V4 * V4   V3 = V3 - V2

V3 = SQR V3   V5 = V4 * -1   VX = V5 - V3   VY = V5 + V3

VX and VY will contain the solutions after the execution of the program line.

Sequence The sequence in which the functions are written determines the sequence in which they are executed.

Note *) For arithmetic and trigonometric functions the numerical values can be entered directly with max. 3-digit, positive integer numbers or max. 3-digit negative integer numbers.

**) CPC computing functions (SQR, COS, SIN, ATG) should be programmed without any space characters only with variables; numerical values are not permitted.

INCREMENT / DECREMENT

Increment value INC V1

Decrement value DEC V1

Definition

The content of a variable is incremented or decremented by 1.
Any digits after the decimal points are deleted.

Programming

INC VN DEC VM

Example

```
N1  V1 = 12  V4 = 1.7  V5 = -1.3
.
.
.
N13  INC  V1  INC  V4  DEC  V5
```

After the execution of N13 the contents of the variables are as follows:
V1 = 13 V4 = 2 V5 = -2

Integer Number

A real number can be converted into the corresponding integer number by performing the INCREMENT and then the DECREMENT function.

REGISTERING TIME TIM V1

Definition

The time elapsed since the start of the program is loaded into the variable (seconds).

Programming

TIM VN
N from 1 to 99 and from A to Z.

Example

```
N1  $5
N2  G91
N3  G1  X1  F250
N4  TIM  V1                   registering time for G1-function
                              storing value in V1,
N5  V2 = 50                   time limit 50 seconds;
N6  V3 = V2 - V1  BGT  P5     checking condition and branching;
                              the program sequence is not
                              completed until the time limit
                              of 50 seconds is reached; other-
                              wise a jump is made into SBP 5.
N   M2
```

TRIGONOMETRIC FUNCTIONS

Sine	**VN = SIN V1**
Cosine	**VN = COS V1**
Arc tangent	**VN = ATG V1**

Definitions The sine or cosine value of an angle (in degrees) is formed (SIN/COS).

The corresponding angle (in degrees) is formed from the tangent (ATG).

Programming VN = SIN V1 V0 = COS V2 VP = ATG V3

Example N1 V10 = 30 VX = COS V10 VY = SIN V10

.

.

.

.

Operation The sine or cosine of any angle can be formed.
Angle values are to be entered via variables.
The direct input of numerical values is not permitted.

TOOLS

LOAD TOOL STORE COR = V1 R = V2 L = V3 **)

Definitions

The tool store is loaded.

Programming

```
N1    VN = 15
N2    COR = VN  R = VP  DR = VR  L = VQ  S = VS
```

VN from 1 to max. 48.

After the execution of N2 tool 15 will be loaded with the data from VP to VS.

COPY TOOL DATA COR = V1 V2 = RR V3 = L **)

Definition

Variables are copied from the tool store.

Programming

```
COR = VN  VP = R  VR = DR  VQ = L  VS = S
```

VN from 1 to max. 48.

Operation

Values are only copied, i.e. the tool data do not affect the machined path.

Example

```
N1    V12 = 15  V13 = 15.0  V 14 = 75
N2    COR = V12 R = V13  L = V14
```

After the execution tool 15 is loaded with

R = 15.0 L = 75.0

Example

```
N1    V4 = 25
N2    COR = V4  V1 = R  V2 = L  V3 = DR *)
```

After the execution of N2 the contents of the variables will be as follows:

V1 = radius V2 = length V3 = tool wear of tool 25.

Note

*) The input of the tool wear (DR) depends on the radius (R); limit: 10% of radius.
The DR value/ modification is entered as an incremental value.

) The COR instruction should be programmed in **a single line together with the variables.

ZERO SHIFTS

Load zero shift G54 to G59	**TRF = V1 X = V2 Y = V3**
Copy zero shift G54 to G59	**TRF = G54 V1 = X V2 = Y**
Copy active zero shift G92	**TRF = G92 V1 = X V2 = Y**
Copy active pole (polar coordinates)	**TRF = G20 V1 = X V2 = Y**
Copy active scaling factor switching G36	**TRF = G36 V1 = X V2 = Z**

Definitions

The zero shift table is loaded or values are copied from the zero shift table. The values of the zero shifts and the values of the pole position are copied. The values of the active scaling factor are copied into V1 for the active plane, and into V2 for a possible change in scaling factor in the third axis.

Programming

Load zero shift G54 to G59

```
N2  TRF = VN  X = VP  Y = VQ  Z = VR  E = VS
.
```

Copy zero shift G54 to G59

```
N2  TRF = G54  VP = X  VQ = Y  VR = Z  VS = E
.
```

Copy pole (the coordinates relating to the active G20 zero point of the active pole are copied)

```
N2  TRF = G20  V1 = X  V2 = Y  V3 = Z  V4 = E
.
```

Copy zero shift G92 (current difference between commanded position and machine position)

```
N2  TRF = G92  V1 = X  V2 = Y  V3 = Z  V4 = E
.
```

Example

```
N1  V1 = 54  V2 = 100  V3 = 200  V4 = 150  V5 = 70
N2  TRF = V1  X = V2  Y = V3  Z = V4  E = V5
N3  M2
```

After the execution of N2 the zero shift corresponding to G54 is defined as follows:

X workpiece zero point at coordinate 100
Y workpiece zero point at coordinate 200
Z workpiece zero point at coordinate 150
E workpiece zero point at coordinate 70

UNCONDITIONAL BRANCHING

Jump into subprogram	**indirect jump**	**BSR V1**
	direct jump	**BSR P5**
Jump to label	**indirect jump**	**BRA V1**
	direct jump	**BRA P5**

Definitions

Depending on the results of calulations a freely selectable subprogram can be called up, or a jump can be performed. Both instructions can also be carried out without condition.

The jump target can therefore either be
P5: direct jump address label 5 ($5)
or subprogram 5 ($5)
V1: indirect jump address
label no. = content of V1 or
subprogram no. = content of V1

Programming a jump indirectly

```
N1  VN = M          allocation of value to variable
N2  BSR  VN         call-up of subprogram no. M
.                   (content of VN)
.
```

Programming a jump directly

```
N2  BSR  P5         call-up of subprogram 5
.
.
```

Indirect jump to label

```
N2  BRA  V1
.
.
```

Note on multiple branching

Several branching directions BSR can be programmed in one NC block. The first subprogram call-up the conditions for which are fulfilled will be carried out. Subsequently the next block number will be executed.

CONDITIONAL BRANCHING

In addition to being dependent on signals program branching can be tied to the following conditions:

- mathematical comparisons
- modal effect of various G/M-functions
- whether or not mirror image is active

SETTING CONDITION REGISTER

TST V1

The basis of all types of branching described in the following text is the status of the

CONDITION REGISTER (CR).

After mathematical operations or after "TST" the control will load the result into the internal condition register with the values of the variables.

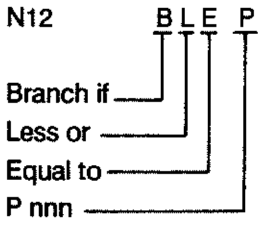
TST must be used before the branching, if the variable on which the branching is to depend is not yet in the CR.

Programming **N10 TST VN**

Branching operations are only carried out correctly if the result from the preceding operation contains the conditions for the particular branching.

If, for instance, a multiplication is carried out in line 5 and no further instruction follows, which would set the condition registers, the result of this multiplication would still take effect in block 12 of the example on the next page.

General format for programming conditional jumps:



branch if
less or
equal to 0 to
label nnn

If the jump condition is not fulfilled, the subsequent block will be executed.

CONDITIONAL BRANCHING / CONDITION REGISTER (CR)

Automatic Loading

The condition register is loaded automatically by operations such as the basic arithmetic functions.
After the operation it will indicate what the result of the computation is compared to zero:

EQ = equal zero
NE = not equal zero
GT = greater than zero
LT = less than zero
LE = less/equal zero
GE = greater/equal zero

Loading via TST

Not all operations load the condition register automatically.
Example: A value is copied into V15 from the tool table.

If a branching is to depend on the value contained in V15 after the copying the condition register must be set with TST 15 before the decision is defined.

Programming

```
N12  BLE  P27                  jump to label 27, if condition
.                              "BLE" is fulfilled; otherwise
.                              continue at block 13
N20  $27                       jump target
or
N10  V12 = V11-V10             calculation of required jump address
N11  TST  V15                  set condition register
N12  BLE  V12                  jump to address 28 (content of V12) if
.                              condition regarding V15 is fulfilled
.
N31  $28                       jump target
```

CONDITIONAL BRANCHING AFTER MATHEMATICAL COMPARISON

The jump address can be defined by one of two means:

- **indicated,** as content of **a variable V** or
- **directly,** by specifying a **label with P.**

Conditional branching does not automatically set the condition register.

BEQ Branch if EQual to zero

BEQ V5
BEQ P1

All digits before and after the decimal point must be 0

BNE Branch if Not Equal to zero

BNE V5
BNE P1

The jump condition is fulfilled if at least one digit before or after the decimal point is not equal to zero.

BGT Branch if Greater Than zero

BGT V5
BGT P1

The condition is fulfilled if the result is a positive number of at least one increment.

BLT Branch if Less Than zero

BLT V5
BLT P1

The condition is fulfilled if the result is a negative number of at least one increment.

BGE Branch if Greater than or Equal to zero

BGE V5
BGE P1

The condition is fufilled if the result is = 0 or positive.

BLE Branch if Less than or Equal to zero

BLE V5
BLE P1

The condition is fulfilled if the result is = 0 or negative.

Note: If several jump instructions are programmed in one block the user must check the corresponding jump addresses.

Example **JUMP AFTER COMPARISON WITH A VARIABLE VALUE**

The X-axis is to traverse to the value calculated for V7.

Condition The traversing movement is to be carried out if the value in V7 exceeds 10 (content of V5).

If the value is greater the program is to be abandoned by making a jump onto the program end.

```
N1    G1    X100        F500
:
N12   V5 = 10                       load value for comparison
N13   V7 = V3 / V2                  calculate V7 from
                                    variable sources
N14   V4 = V7 - V5                  determine whether jump
                                    condition is fulfilled
N15   BGT P2                        define jump condition and
                                    jump target
N16   X = V7                        carry out movement in X,
                                    since content of V7 does
                                    not exceed 10
.
.
.
      $2                            corresponding jump target
N20   M2
```

Note If the jump condition in block 15 is defined as "**BGT**", the movement will be carried out for V7 values of up to 10.000.

A V7 content of 10.001 will produce program stop.

The jump condition "**BGE**" in block 15 would produce a program stop for a content of 10.000 and above.

The jump condition "**BEQ**" in block 15 would only produce a program stop if V7 was exactly 10.000.

BRANCHING CONDITION: NC INSTRUCTION

Definition

Branching can be made dependent on the active state of certain modal conditions.

The tests described below will set the condition register (CR) = 0, if the relevant condition is fulfilled.

After the test branching can take place, dependent on the status of the condition register.

G-FUNCTIONS

Test whether a particular G-function is active as a modal function **TST G1**

When testing for G1 the CR = 0 if G1 is active; the CR = 1 if G1 is not active.

Programming

N12 TST Gn

Range of G-functions (n) for which the test can be carried out:

G0, 1, 2, 3, 17, 18, 19, 39, 53-59, 62, 65, 66, 90, 93, 94, 95, 97

Example

```
N10  TST  G17  BEQ  P1
N11  TST  G18  BEQ  P2
N12  TST  G19  BEQ  P3
.
N19   $1
.
.
N30   G99
```

check whether working plane G17 is active; if not, jump to label $1

M-FUNCTIONS

Test whether a particular M-function is active as a modal function **TST M41**

Programming

N12 TST M n

Admissible range for n: 3, 4, 5, 19, 41 - 44

MIRROR IMAGE

Test whether mirror image function is active for one or several axes **TST QX**

INCH/METRIC

The whether measuring system is defined as inch or metric **TST QM**

Programming

N12 TST Qn
Admissible for n: X, Y, Z, E, M, M = metric

AXIS INFORMATION X = V1

Definition NC addresses are loaded from variables.

Operation During subsequent program execution the control will carry out the instructions as with DIN programming.

Programming

```
N10  X = V1  Y = V15
.
.
```

Admissible DIN addresses	
axis traverse	X Y Z E I J K A D R = VN
output M-functions/aux. functions T must be loaded with 4 digits, possibly with internal effect	M S T = V1
determine feedrate	F = V1
select/set G-functions 2 or 3-digit address	G = V1

Example

```
N1    F500
N2    V10=50  V11 = 27  V12 = 3  V13 = 15
N3    V1 = 60
N4    V2 = V12/V1
V5    V3 = V2 + V11
N6    V2 = V13/V1
N7    V2 = V2/V1
N8    V3 = V2 + V3
N9    A = V3  D = V10
N10   M2
```

Execution Axes traversing to a position which is defined by the following values:

	value
radius V10	50
angle V11 degrees	27
V12 minutes	3
V13 seconds	15

Axes X and Y traverse to the following positions:
X = 44.529 Y = 22.742

POSITIONING POS

(Traverse axes with external command)

Linear axes can be traversed with an external command during the execution of a program. The POS (axis) function must be written into the part program at the appropriate place. The interface signal DRIVES ON goes "low" for the particular axis(es) (servo loop is open). The current position is displayed. An external command can then be applied. Once the interface signal DRIVES ON is switched back on the servo loop closes. The active part program will then be resumed.

Example

```
N1  G1  X50  Y20  Z10  F500       all servo lops are closed;
                                  traverse to axis positions
N2  X100
N3  M55                           interface signal DRIVES ON for
                                  Y and Z = "low" (via M-function, for instance);
N4  POS  Y                        Y and Z axis are traversed with
N5  POS  Z                        external command (servo loop open):
                                  interface signal DRIVES ON for
                                  Y and Z = "high" (servo loop closed);
N6  X10  Y10  Z15                 traverse to axis positions
.
.
.
.
N15  M2
```

Note: Each POS function only applies to one axis; if several axes are involved the POS functions must be programmed in separate blocks.

The POS function can only be applied for the E-axis if it is defined as a linear axis.

STV - FUNCTION (SET VARIABLE FUNCTION)

Definition

During the course of a program execution it is possible to enter values into the variables table by MDI, via the serial interface or via the PLC. The updated variable values become active within the current program through programming of the STV function.

The STV function stops the program execution and interrupts the block preparation process. The updated value from the variable table is transferred into the working store.
With the interface signal "STV" the program sequence is resumed.
The new variable value will be processed by the program.

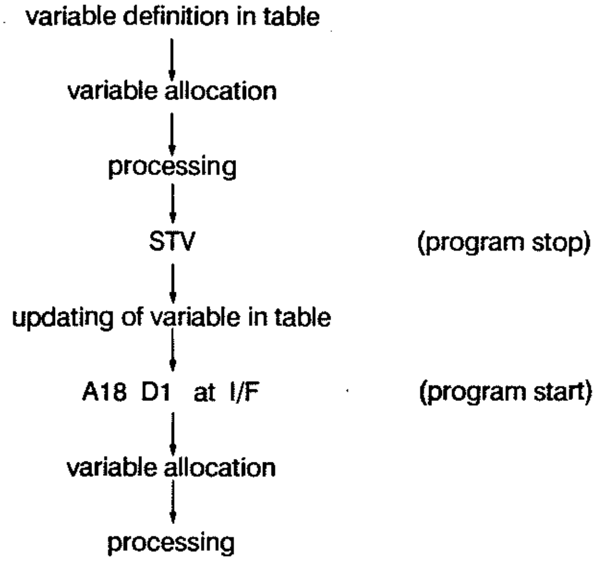
Operation

- STV is not modal.

- The complete variable table is updated with STV.

Programming

The STV function must be programmed immediately prior to the variable which is to be changed.



Example

Variables for a fixed machining cycle are loaded into the NC by the PLC via the STV function.

N15 V70 = 50 V75 = 115 . . .	loading current variables via MDI directly into the variable table or in program by means of load instruction
N19 G0 X = V70 Z = V75	traversing movement
N20 F500 S250 M3	machining parameters
N21 G81 V1 = 35 V2 = 109	1st boring operation
N22 STV →	NC to interface **A18 Data 0** (part program stops)
←	interface to NC: data transfer Axx Dxx V70 Axx Dxx V75 Axx Dxx V76 . . .
	A18 Data 1 - end of data transfer; program continues
N23 X = V70 Z = V75 . .	next boring position defined by STV; 2nd boring operation

Output BCD output bus: **A18 Data 0**

Input BCD input bus: **A18 Data 1**

Note The function is applicable to the complete variable table (V1 to V99 and VA to VZ).

CPC SAMPLE PROGRAMS: 1. Ellipse

Task

Path calculation for an ellipse (centre of ellipse = coordinates 0/0).
The ratio between the to radii is to be 0.4. The program is stored as cycle 65. It is called up with G865. The ratio should be definable by one single variable.

Sequence

N1	jump addrress (label 1)
N2,3	calculation of X-coordinate
N4-6	calculation of Y-coordinate
N7	positioning to X/Y coordinates
N8,9	feed-in in Z (lst positioning only)
N10	increment angle until final value is reached

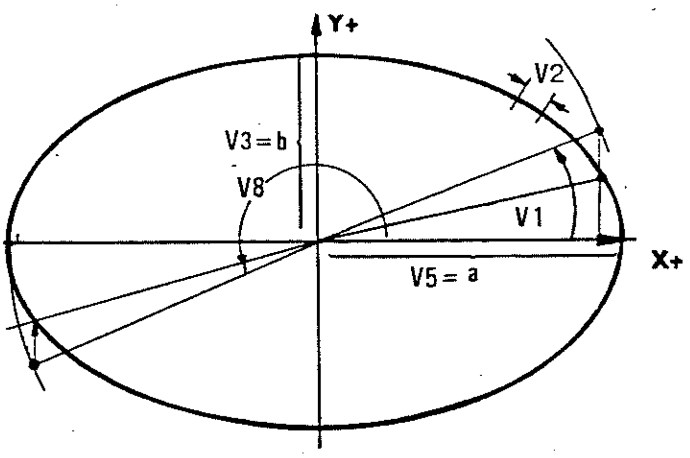
Used Parameters

V1	starting angle alpha	0
V2	incrementing angle in alpha	2
V3	radius b	10
V4	value for condition	
V5	radius a (larger radius)	25
V8	final angle	360
V6	cosine - -> X-component	
V7	sine - -> Y-component	
V10	milling depth in Z	-0.5

Advantages

The resulting program is considerably shorter than a conventional program, which would describe an ellipse as a contour made up of at least 10 arcs. It is also fully flexible with regard to the used radii and the ratio between them (b/a).

Programming



Cycle 65

```
N1   $1
N2   V6 = COS V1
N3   V6 = V6 * V5
N4   V7 = SIN  V1
N5   V7 = V7 * V5
N6   V7 = V7 * V3
N7   X = V6  Y = V7
N8   G1
N9   Z = V10
N10  V1 = V1 + V2
N11  V4 = V8 - V1
N12  BGE P1
N13  G0
N14  M2
```

Call-up and Example

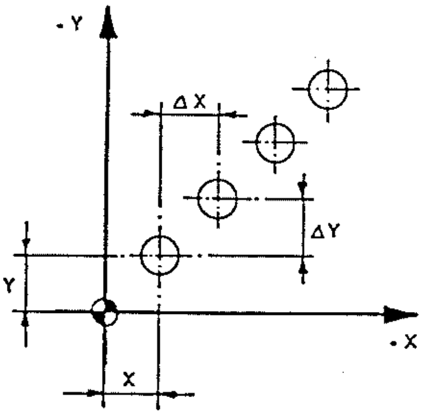
```
N1   G0    Z20
N2   G865 V1=1  V2 = 8  V3 = 10  V5 = 30  V8 = 359  V10 = 12
N3   Z20
N4   M30
```

Note:

Careful selection of the V2 value (incremental angle) makes it possible to achieve an optimum combination of accuracy and speed. Angle values relate to the circle with radius a. The corresponding Y-coordinate is modified by radius b (V3)!
The program will work in a counter-clockwise direction.

CPC SAMPLE PROGRAMS: 2. Row of Holes

Definition of the variables



X = V90

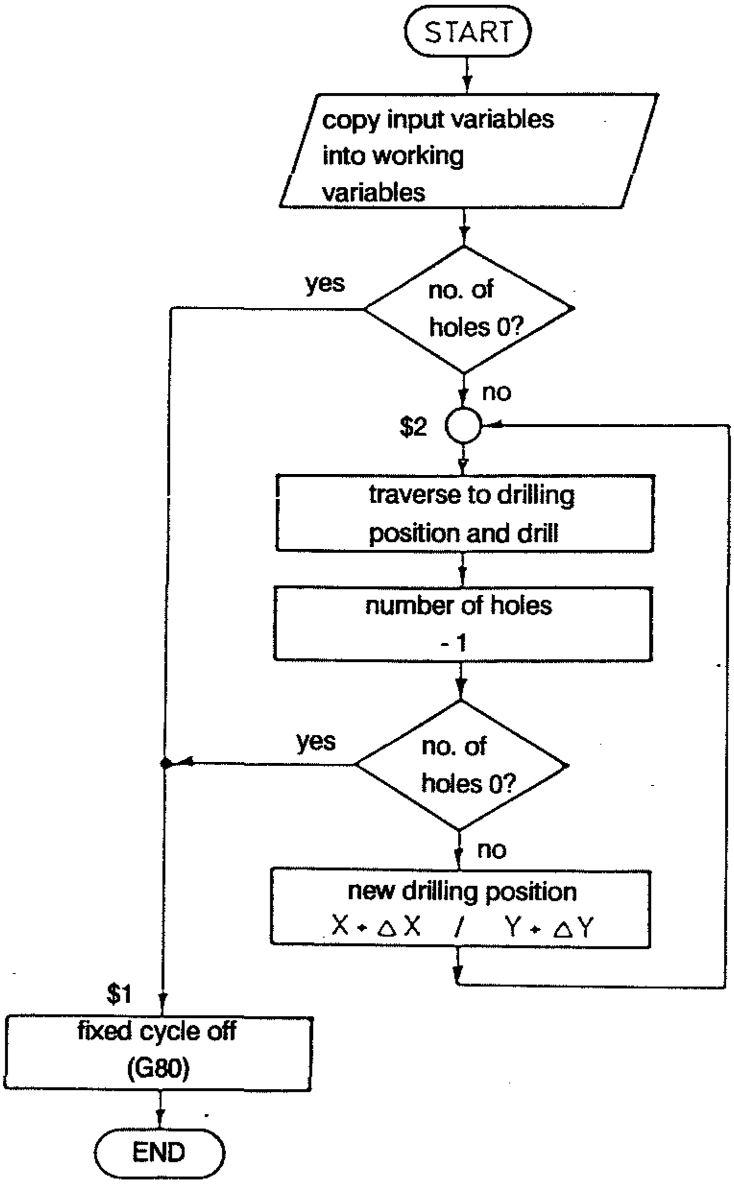
Y = V91

ΔX = V92

ΔY = V93

number of holes = V94

Sequence



Program construction
(solution)

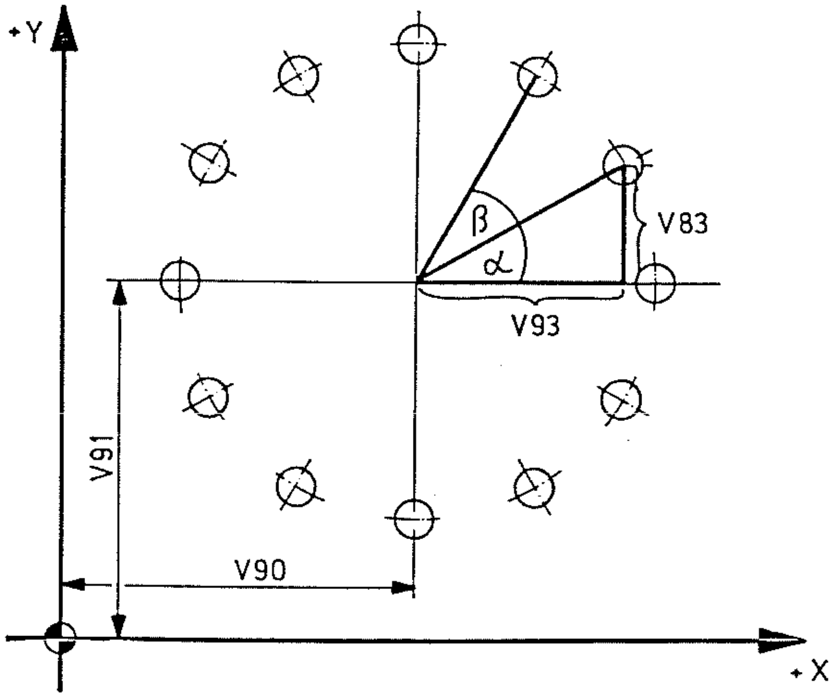
```
N1   V40 = V90
N2   V41 = V91
N3   V44 = V94

N4   BEQ P1
N5   F500  S250  M3
N6   G81  V1=20  V2=0
N7   $2

N8   G0
N9   X = V40 Y = V41

N10  DEC V44

N11  BEQ P1

N12  V40 = V40 + V92
N13  V41 = V41 + V93
N14  BRA P2
N15  $1
N16  G80

N17  M2
```

CPC SAMPLE PROGRAMS: **3. Bolt Hole Circle**

The following requirements need to be provided for:

- variable X/Y position
- variable number of holes
-variable angle related hole distribution

Definition of the Variables

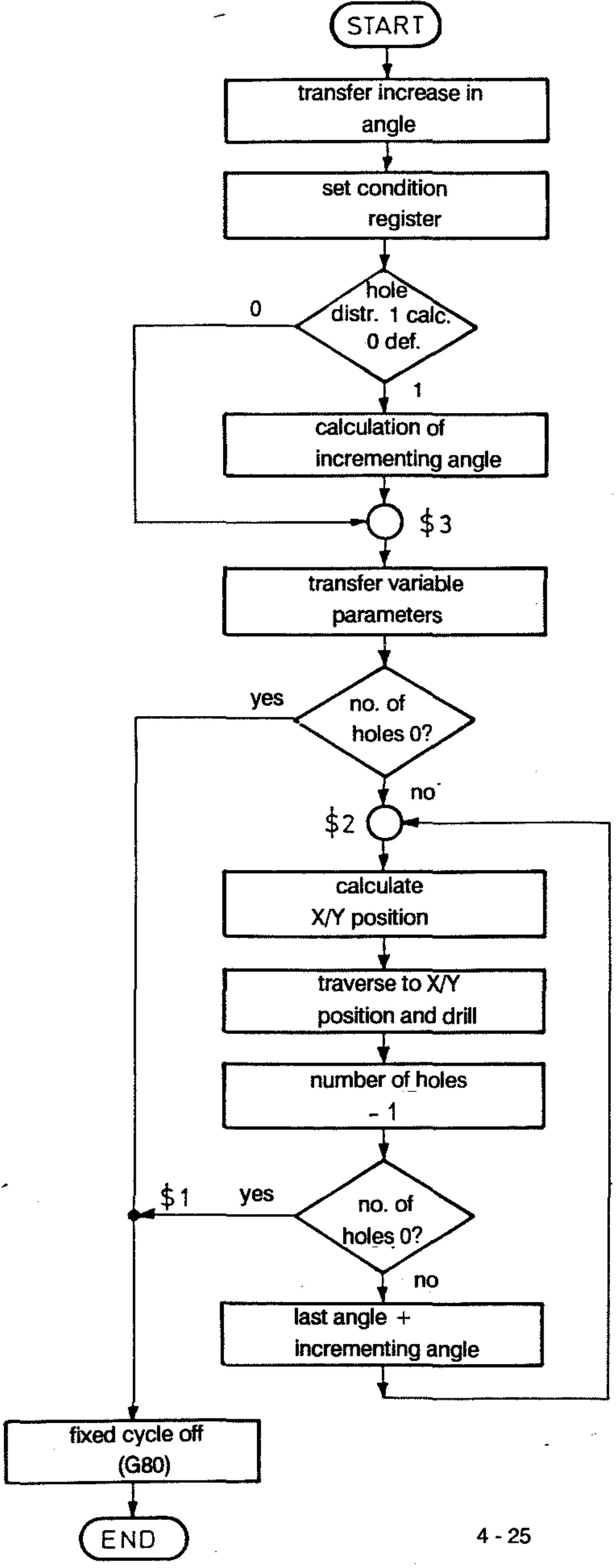


V33 calculated angle value
V35 calculated number of holes
V80 sine value for Y
V81 cosine value for X
V82 bolt hole circle radius
V83 Y-position
V84 X-position
V85 calculated increase in angle

V90 position of centre of circle in X
V91 position of centre of circle in Y
V92 bolt hole circle diameter
V93 starting angle
V94 incrementing angle
V95 number of holes
V96 hole distribution 1 = calculated angle
0 = defined angle (V94)

V97 angle for hole distribution

Sequence



Program Construction

(solution)

```
N1    F750
N2    V85 = V94
N3    TST V96
N4    BEQ P3
N5    V85 = V97/V95
N6    $3
N7    V33 = V93
N8    V35 = V95
N9    BEQ P1
N10   V82 = V92/2
N11   $2
N12   V80 = SINV33  V81 = COSV33
N13   V83 = V82 x V80  V84 = V82 x V81
N14   V84 = V84 + V90  V83 = V83 + V91
N15   X = V84   Y = V83
N16   V35 = V35-1
N17   BEQ P1
N18   V33 = V33 + V85
N19   BRA P2
N20   $1
N21   G80
N22   M2
```

5. TECHNOLOGY

PROGRAMMING

INTERNAL PROCESSING OF TOOL TECHNOLOGY DATA

When the relevant machining functions are called up the control automatically provides tool compensation according to the tool data in the technology store:

Tool Geometry

G40 to G42	tool radius
T	tool length

The compensations for tool length and tool radius and their cancellation are programmed with separate instructions. Once called up the compensations remain active as modal functions. The relevant compensation group must be defined.

Tool radius compensation can be further defined by

G68/69 behaviour at outside corners.

All compensation data can be input via the keyboard after selection of TOOLS by soft key.

Parametric functions can be used to make allocations to tool compensation table data, and compensation data can be copied and applied.

Feedrate

The programmed feedrate (F-word) is interpreted in different ways:

G94/95	feedrate in mm per minute or per revolution
G96/97	cutting speed / spindle speed

The feedrate applies as follows:

with G64	along the programmed contour (cutting point path)
with G65	along the tool centre path

Cutting Speed

With G97 the control forms the spindle speed directly from the active S-word.
The programmer determines the cutting speed by programming the appropriate spindle speed.

With G96 the control calculates and outputs the required spindle speed depending on the S-address (cutting speed), as defined in the technology store, and the used tool radius

Gear Ranges

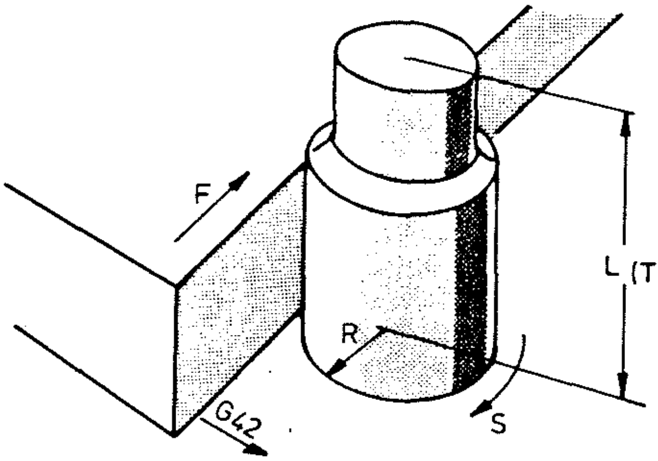
M41-44	direct selection in the program
M40	selection made automatically by the control at the beginning of the block

TOOL COMPENSATION

Definition

The control can convert a part-related program into a tool path.

When a tool compensation is programmed the control will automatically take into account the following tool-related characteristics, which are stored in the technology store:



Tools

L	length	mm
R	radius	mm
DR	radius wear	mm
S	cutting speed	m/sec
	number of compensation groups	max. 48

Without Tool Compensation

The control can carry out a program without any modification if the machine and the required machining do not require any adjustment. The block processing time is short.
Geometry, spindle speed, output signals, and feedrate take effect as programmed.

External Tool Compensation

This also applies to programs through which the cutter centre path is described by external calculations.

Any demands regarding values, which are to be determined indirectly, such as constant cutting speed, usage of the optimum spindle speed, must be realized through specific values for M and S for the particular program run.

See also chapter 1 INTERFACES for the transmission conditions.

Compensation Call-up

The tool length compensation is called up via T.
The radius compensation is called up with G41/42.

Cancelling the Tool Compensation

Both tool length and radius compensation are cancelled with T00.
G40 cancels the radius compensation alone.

TOOL LENGTH COMPENSATION ADDRESS T

The tool length is taken into account when the T-word is called up.
The effect of T is restricted to the tool length compensation.
Tool length compensation can be used in all machining modes.

General Format

```
T  XX XX
   |  |______ tool number, physical, for output
   |_________ compensation group for internal computations
```

Allocation

Compensation group and output tool number can be freely combined in the call-up for T.

Examples

T can be programmed with 2 or 4 digits.

T 00	tool length compensation and path compensation are cancelled; no output
T 12	compensation group 12 is selected; no output of number
T .. 02	tool number 2 is output; tool length compensation remains unchanged
T 0812	compensation group 8 is selected; tool number 12 is output
T 1212	compensation group 12 is selected; the same number is output

Effect

The first two digits behind the T (Txx) always effect the tool compensation call-up.

The 3rd and 4 th digits specify the tool number and are output at the interface, if they are programmed. The tool length L, which is stored in the tool table, is incorporated according to the sign into the values for the axis, in which the tool length compensation applies.

The compensation value takes effect
- immediately for the axis display
- for the path once the relevant axis is programmed.

Examples

T + Z programmed separately			T and Z programmed together			
N2	T08	corrected display for Z-axis	N2	T08	Z50	immediate phasing in of the compensation in Z-axis movement + corrected axis display
N3	Z50	phasing in of tool length compensation				

Note

When a tool number is programmed with 4 digits the last two are displayed in automatic mode to show the active tool number.

TOOL LENGTH COMPENSATION ADDRESS T

Call-up

The tool length compensation is phased in and out during a movement in a linear mode. The feed-in axis is to be programmed on its own.

Allocation

Plane		tool length is compensated for in
G17	(X/Y)	Z
G18	(Z/X)	Y
G19	(Y/Z)	X

There are basically two situations in which the tool length compensation is used:

Programming without consideration of the tool length.

In this instance the effective length of the tool needs to be stored in the tool table.

The compensation value corresponds to the distance between spindle nose and the tip of the tool.

Example:
complete
tool length

G1 Z-50 F100 T08
Content of tool length 8 = 100
The Z-axis will position to -50 + 100 = 50

Programming with reference to a zero tool

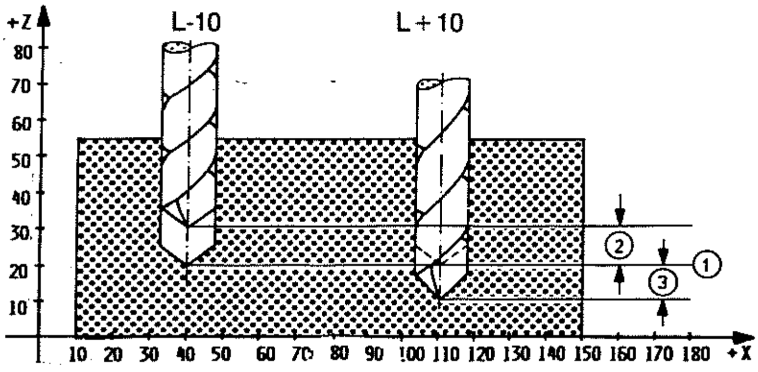
When using this original tool, tool compensation value L = 0 is applied. If a new tool is any shorter or longer, the difference $L_{act.} - L_{orig.}$ is entered into the compensation store.

Example:
difference
in tool
lengths

G1 Z50 F100 T08
Tool length taken into account by the program: = 100 mm.
Actual length of tool 8 = 90 mm.
Tool length compensation in Z-axis.
Plane G17 (X/Y).
Z will position to 40.

drill too short
compensation
L-10

drill too long
compensation
L+10



1 = programmed drilling depth
2 = "-" compensation
3 = "+" compensation

TOOL RADIUS COMPENSATION G40 / 41 / 42

Definition

The radius compensation converts the contour related part program into a cutter centre path (equidistant).
The equidistant runs parallel to the programmed contour at a distance which corresponds to the active cutter radius.
The side at which the equidistant runs with respect to the programmed path is determined with G41/G42.

Treatment of Corners

The control calculates

- intersections at inside corners and
- auxiliary arcs at outside corners (G68) or also
- intersections at outside corners (G69)
  Whether G68 or G69 is active on switch-on is determined by M-parameter.

Feedrates F

Feedrate values modal and, when relating to the machining of the part contour, apply

- to the **cutting point** (G64) or
- to the **cutter centre path** (G65)

Cutting Speed

The cutting speed can be determined indirectly by

- the determination of a fixed spindle speed for a given tool radius (G97 + S-word).

Alternatively, automatic and direct definition is possible via

- G96 with the S-word in the technology store.

Cutter Radius

R positive

R is stored in the tool table and represents the cutter radius relevant for the program execution.

R = 0

R can be set to 0 if, for instance due to extreme speed requirements, the part is programmed by describing the tool centre path.
The program is then executed without any path compensation.

Tool Wear DR

Additive, small compensation for the nominal tool radius, which, for instance, takes into account the regrinding of the tool.

If DR is programmed without sign this corresponds to an increase in the effective tool radius.

Detailed description of the functioning of the TOOL COMPENSATION in the relevant chapter.

STARTING POINT, BEGINNING OF CONTOUR

Starting Point

In many cases it is not possible to drive directly onto the contour from the tool change point; usually it is necessary to position to an intermediary position (starting point).

The choice of a suitable starting point helps to avoid damage to the contour. The compensations are phased in during the movement onto this point.

If possible the starting point should allow a tangential approach to the contour, but at least it should be positioned so that there will be no reversal of the direction of any axis at the first contour point (free-cutting).

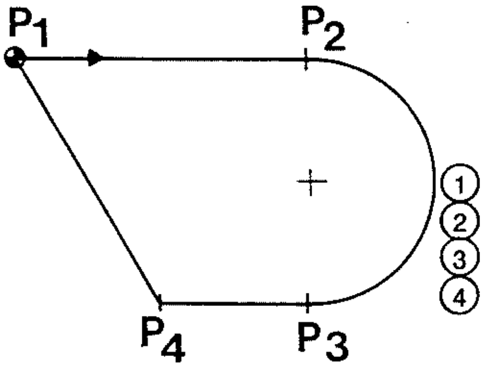
Beginning of Contour

A linear workpiece edge should be chosen, otherwise an intermediary linear movement (of at least 3 increments) must be made.

Compensation Call-up

Compensation call-up must be made while in a linear mode (G0, G1, G61). The block following directly after a call-up (G40, G41, G42) should also be linear.

Sample Contour without Compensation Call-up



```
N1   G1    Z10    F100
N2   X0    Y0
N3   X20   F200
N4   G2    Y-20   R10
N5   G1    X10
N6   X0    Y0
N7   M2
```

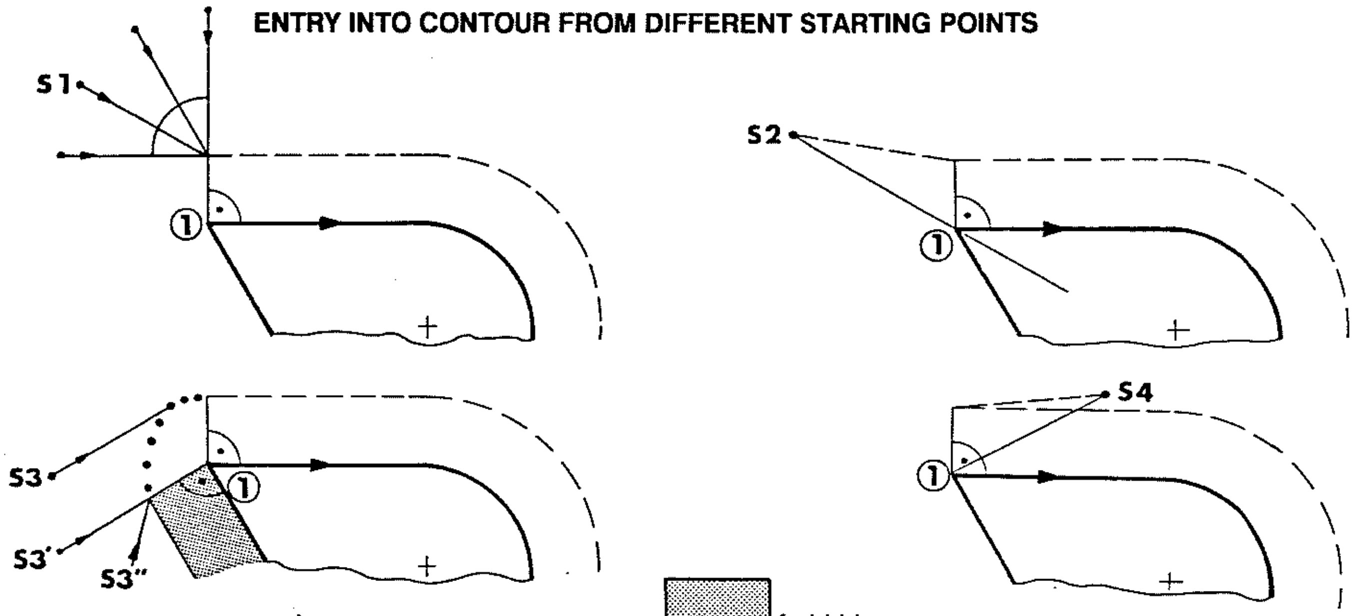
Call-up of a compensation with positioning of the axis (es) in which the compensation is active:

Example: positioning in Z for call-up of T (XY plane)
position in XY for G41, G42 (XY plane)

Phasing in the Radius Compensation

When a radius compensation is called up the control phases in the relevant value in a linear traversing movement. The equidistant starts vertically above the beginning of the first path section for which the compensation is to apply.

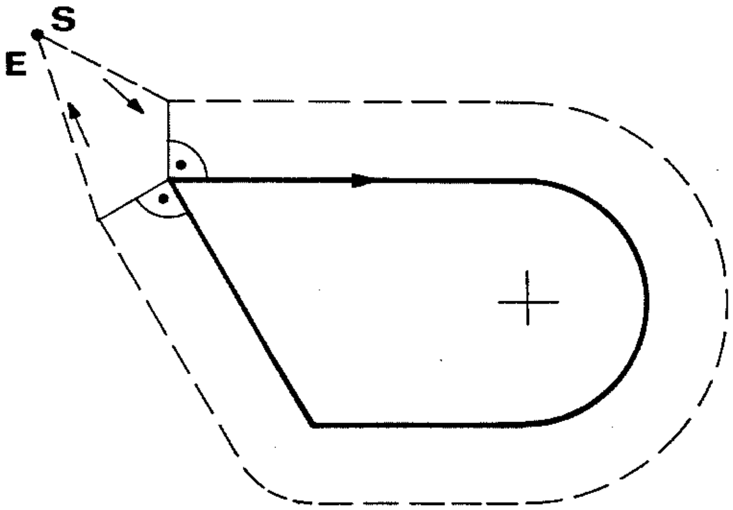
ENTRY INTO CONTOUR FROM DIFFERENT STARTING POINTS



————	contour
————	programmed path
- - - - -	cutter centre path
▨	forbidden area
●●●●	auxiliary movement generated by the control

S1 - S4	The compensation value is phased in from the starting point to P1 in a linear movement. The contour is fully machined at all points and there is no damage to the contour.
S1	Cleanest contour entry through tangential approach movement.
S2	Good contour entry; starting point can also be used as end point.
S3	Lowest possible starting point without collision, considering contour section ④- - >①.
S4	Free-cutting at ① due to a change in direction!

Example with S2 as starting and end point incl. tool compensation

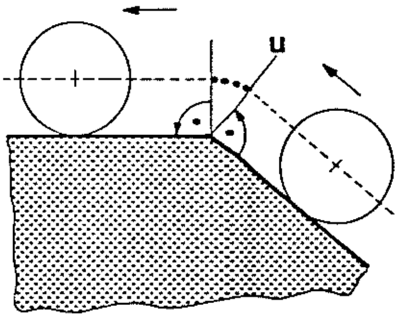


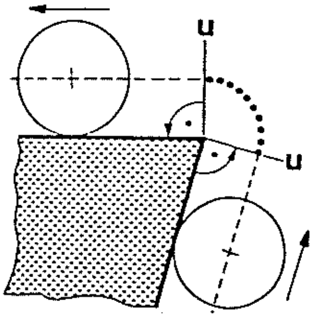
```
N1   G0    Z-10   T01
N2   G1    X-20   Y20   F200
N3   G41   X0     Y0    S500  M3
N4   X20
N5   G2    Y-20   R10
N6   G1    X10
N7   X0    Y0
N8   G40   X-20   Y20
N9   Z300  T00
N10  M2
```

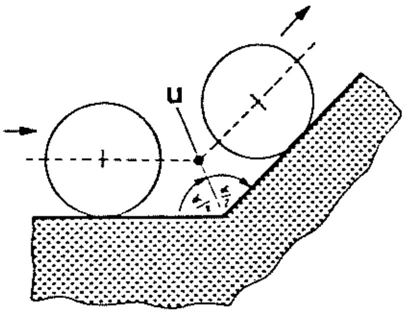
CONTOUR TRANSITIONS WITH G68 (AUXILIARY ARC)

The following examples show how the tool compensation works on corners, by the generation of auxiliary arcs (outside corners) and the calculation of the angle bisector (inside corners).

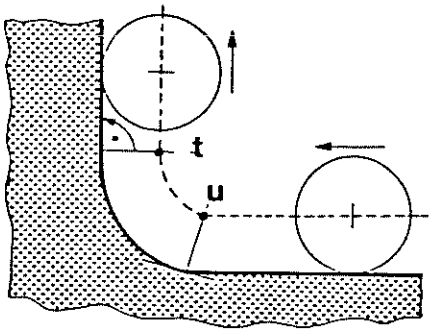
Transitions between linear path sections

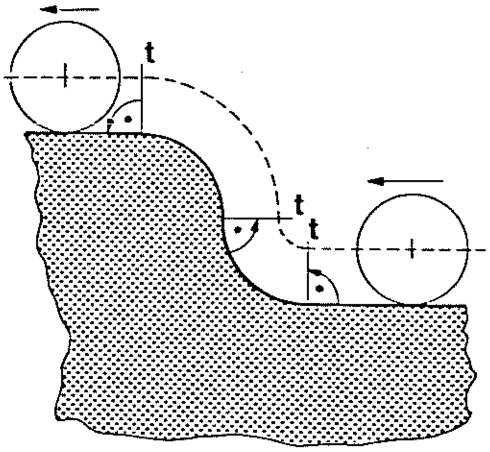


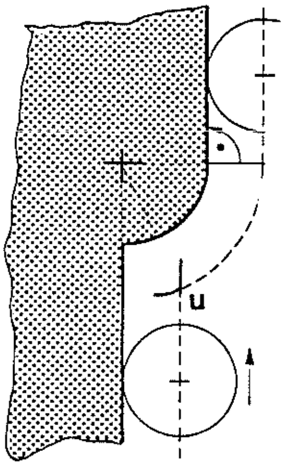




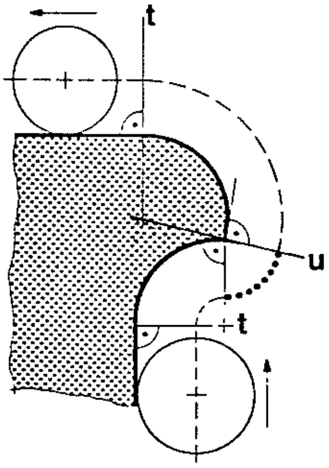
Transitions between circular path sections

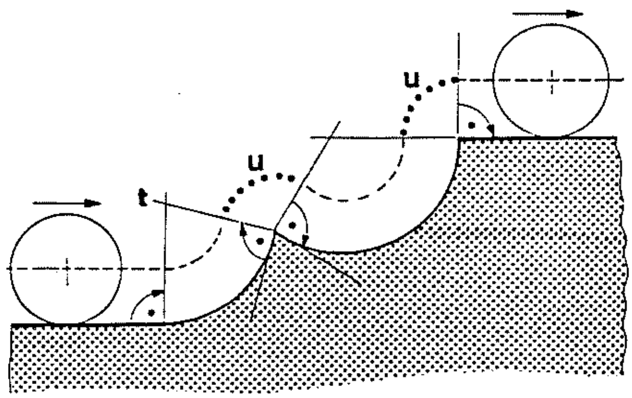






Discontinuous transitions





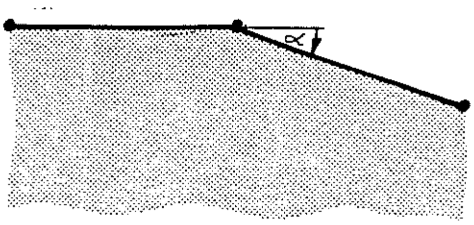
t = tangential
u = discontinuous

.... arc generated automatically by the control

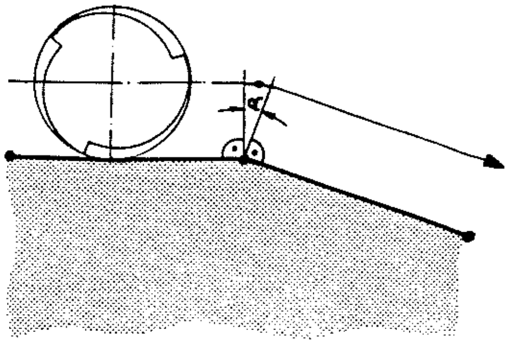
CONTOUR TRANSITIONS WITH G69 (INTERSECTION)

Programming

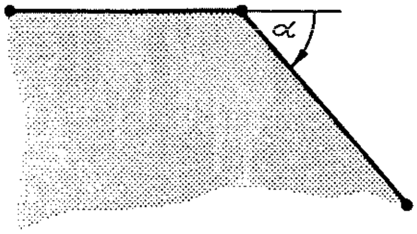
angular variation 0 to 90°

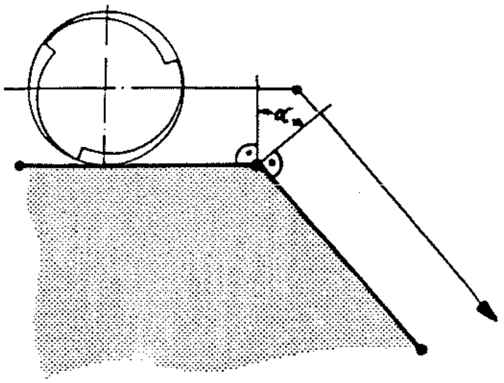


Execution

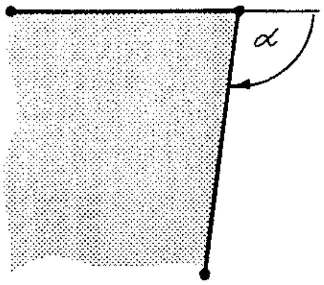


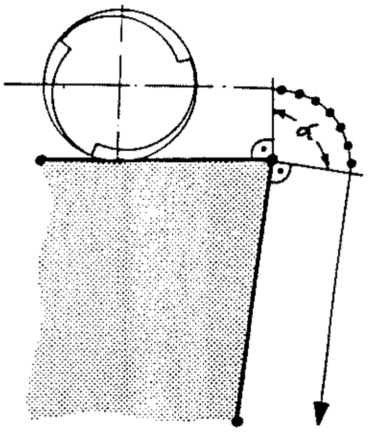
angular variation 0 to 90°





angular variation 90 to 180°





... movement generated automatically by the control

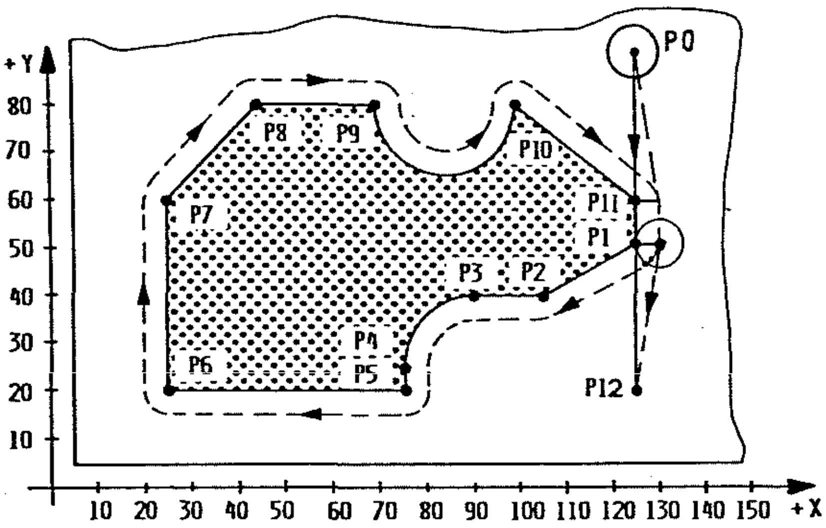
EXAMPLES

G41 on outside contour

Contour programming with tool radius compensation to the left of the workpiece and phasing out of the compensation at the end of the machining.

The tool radius compensation value was stored in the technology table as the R-value. (in this example for T1 : R = 2.5 DR = 0.05 L = 250.0 S = 25.0)

______ programmed path (workpiece contour)
------- corrected path (cutter centre path)

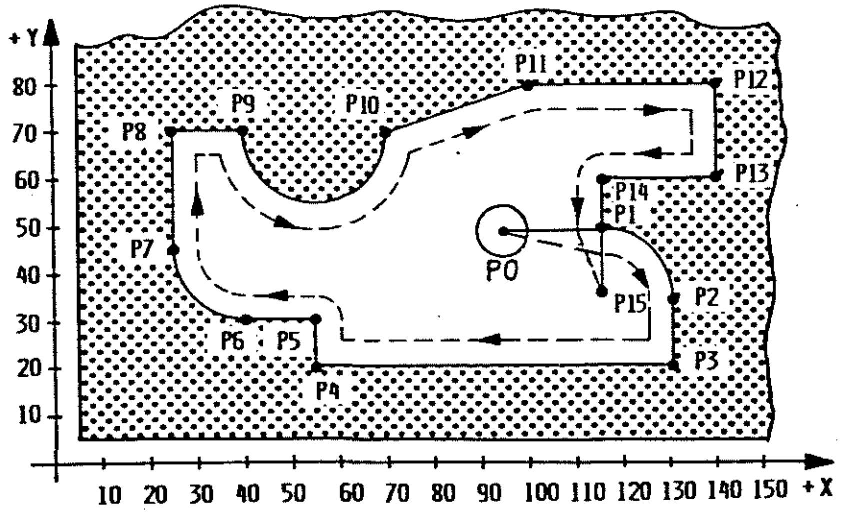


```
N1    G0    X125  Y90
N2    G1    F200
N3    G41   X125  Y60  T01
N4    Y50
N5    X105  Y40
N6    X90
N7    G5    X75   Y25
N8    G1    Y 20
N9    X25
N10   Y60
N11   X45   Y80
N12   X70
N13   G3    X100  R15
N14   G1    X125  Y60
N15   Y50
N16   G40   Y20
N17   M2
```

G42 on inside contour

Contour programming with cutter path compensation to the right of the workpiece and phasing out of the compensation at the end of the machining. (in this example for T3 : R = 3.25 DR = 0.06 L = 175.0 S = 17.5)

The tool radius compensation value was stored in the technology table as the R-value.



```
N1    G0    X95   Y50
N2    G42   T03   F300
N3    G1    X115  Y50
N4    G5    X130  Y35
N5    G1    Y20
N6    X55
N7    Y30
N8    G1    Y20
N9    G5    X25   Y45
N10   G1    Y70
N11   X40
N12   G3    X70   R15
N13   G1    X100  Y80
N14   X140
N15   Y60
N16   X115
N17   Y50
N18   G40   Y35
N19   M2
```

END POINT, CANCELLING THE COMPENSATION

End Point

The return from the contour to the tool change point is usually not made directly, but via an intermediary position (end point).

The choice of a suitable end point helps to avoid damage to the contour, and the tool length compensation can be phased out between the end point and tool change point.

The end point should, if possible, allow a tangential exit from the contour with active radius compensation.
It should be positioned so that there will be no free-cutting due to a change in direction when driving away from the contour.

End of Contour

The last section of the contour should be linear. Otherwise a short linear positioning movement (of at least 3 increments) must be inserted past the end of the contour.

Cancelling the Compensation

The cancellation must be made while in a linear mode (G0, G1, G61). In cases where the tool radius is relatively large in comparison to the contour radius the block following immediately after the cancellation (G40) must also describe a linear movement.

With regard to the choice of the end points the same applies as for the choice of the starting point, in principle (see contour entry).
The optimum exit movement is the direct extension of the last contour section (in analogy to starting point 1).
Starting and end point are different in this case.
A joint starting and end point (such as S2) is also possible.

Referencing is not possible until tool radius compensation has been cancelled

Cancelling Compensation for Inside Contours

Even when working within a restricted space the radius compensation must be cancelled in conjunction with a positioning movement, which must at least equal the tool radius.

To keep the required space to a minimum one of two methods should be used:
- continue in the direct extension of the last movement, or
- move to a position which lies on the same side on which the radius compensation was active, i.e. the right side with G42.

The recommended programming sequence is as follows (G17/G41 active):

- last contour machining (for instance with G2)
- tangential exit from the contour in G1 (program X/Y only)
- retract Z-axis with G1 (program Z on its own)
- G40 with X/Y movement as an extension of the last movement (program only X/Y)
- T00 with Z-movement (program Z on its own)
- program end

SPECIAL CASES - TOOL COMPENSATION

CHANGE OF COMPENSATION

There should preferably be no compensation values active when selecting a new tool.

Any active compensation can only be changed for a new block within the contour description. The interpolation mode in the block in which the change is programmed and in the following block must be linear.

The new compensation value will not be activated until a positioning instruction is carried out in the axis(es) which the compensation applies to.

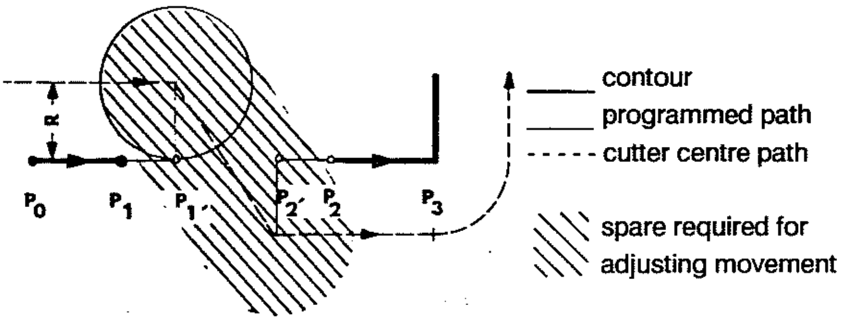
Example: G41 X5 Y7 T02 (XY plane)

The new compensation value is phased in gradually to become fully effective at the end of the first block in which the relevant axes carry out a movement.

SWITCHING BETWEEN G41 and G42

Switching from G41 to G42 and vice versa should preferably take place without radius compensation being active.

If radius compensation is active switching between G41 and G42 is only possible during linear interpolation.
The control will generate an adjusting movement which must be taken into account during the programming!



For instance: Section P0 - -> P1 can be extended to P1', and similarly P2 - -> P3 can be started at P2', in order to achieve a smooth change-over movement.

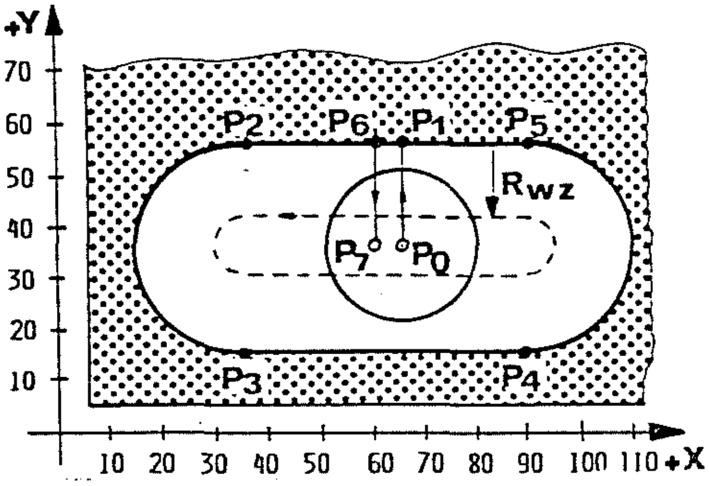
In some cases it might be necessary to cancel compensations via G40, program intermediary positions, and make a new compensation call-up with G41/G42. The minimum length of path sections with which a compensation can be called up or cancelled is 3 increments.

Example 1 **Direct entry into and exit from contour**

Tool radius compensation is used (G17 active).

The contour entry and exit movements overlap.

Contents of tool table for T03 : R = 3.25 DR = 0.06 L = 175.0 S = 1.75



Program

```
N1  G0  X0  Y0  Z0
N2  G1  X65 Y35 F200
N3  G41 Y55 T03
N4  X35
N5  G3  Y15 R20.0
N6  G1  X90
N7  G3  Y55 R20.0
N8  G1  X60
N9  G40 Y35
N10 M2
```

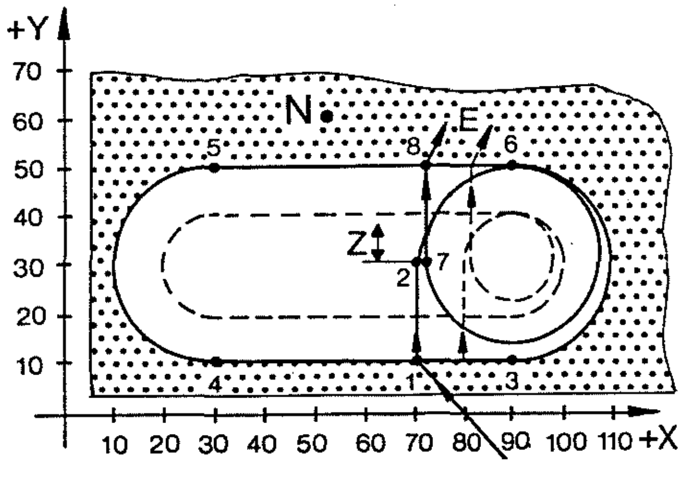
Example 2 **Tangential entry into and exit from contour**

Tool compensations for length and radius (G17 active).
The max. possible tool diameter corresponds to the width of the keyway - 1 inc.

The compensations are phased in and out above the workpiece, which makes this procedure suitable for very limited spaces.

When activating or cancelling a compensation only the axis(es) involved in the radius compensation should be moved.

In the program below the tool table contains the following for T10 : R = 8.0 DR = 0 L = 0 S = 0



Program

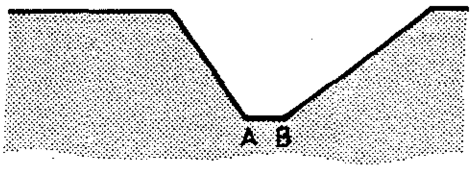
```
N1  G0  Z5    M3   T10
N2  G42 X70   Y10
N3  G1  Y30   F200
N4  Z-2 F50
N5  G2  X90   Y50  R-20 F250
N6  G2  X90   Y10  I90  J30
N7  G1  X30
N8  G2  X30   Y50  I30  J30
N9  G1  X90
N10 G5  X90   Y11
N11 G1  Z5    F2000      M5
N12 Y50
N13 G40 X81   Y60
N14 Z100
N15 M30
```

To cancel the compensation (G40) a movement is required from 7 to 8 in Y positive, or in Y and X positive direction.
Recommended exit via end point such as E, E', E" etc.; exit via end point such as N not recommended. Contour might be disturbed.

SUPPRESSION OF CONTOUR ELEMENTS

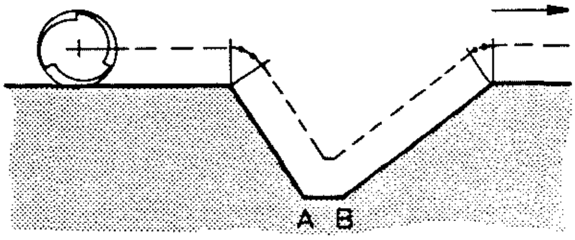
Not all programmed contour elements can be machined because of the radius of the used tool.

Programmed contour

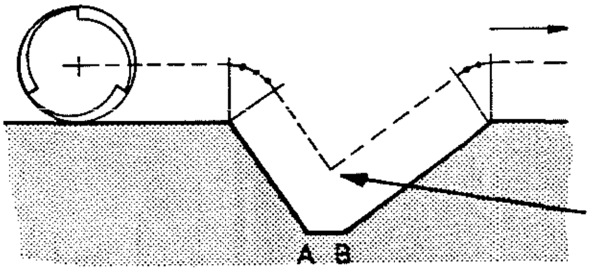


Execution with G68

1. All contour elements are machined.



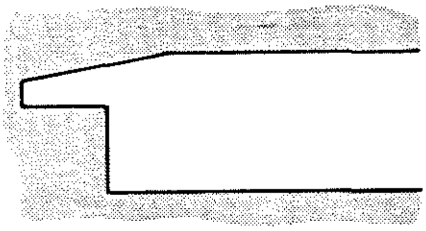
2. One element is suppressed, since tool radius is larger than contour element.



Note: If more than one contour element can not be machined due to the geometrical data the control will interrupt the machining and output an error message.

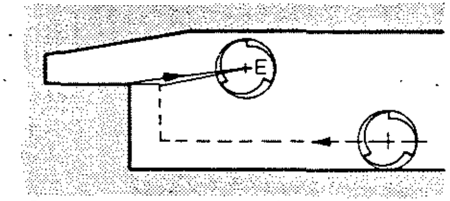
Cancelling Compensation on Inside Corners

Programmed contour

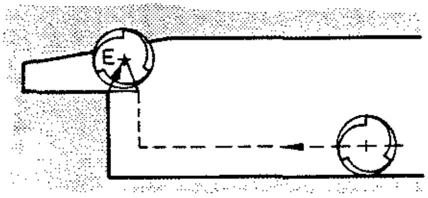


Cancelling compensation with different end points

1. All programmed contour elements are machined correctly.



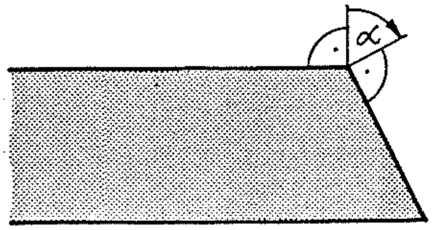
2. One contour element is damaged.



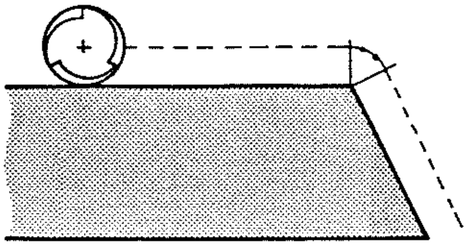
E = end point programmed in conjunction with G40 for the cancellation of the radius compensation

OUTSIDE CORNERS

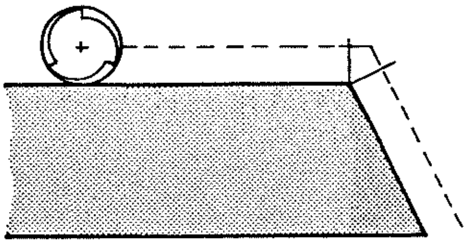
angular variation alpha
between 0 and 90°



execution with G68

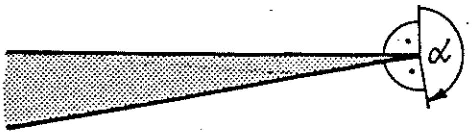


execution with G69

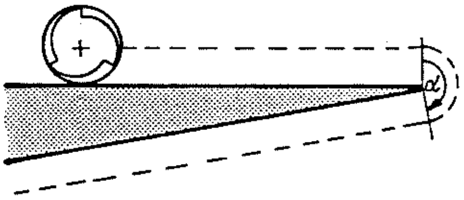


angle alpha larger than 90°
and smaller than 180°

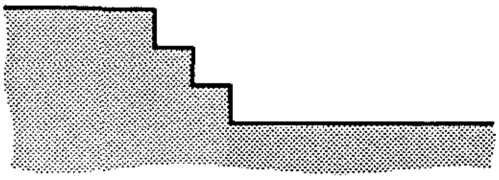
programmed contour



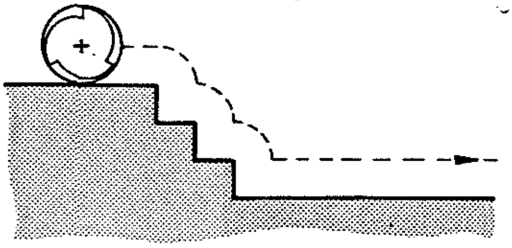
identical execution with G68/69



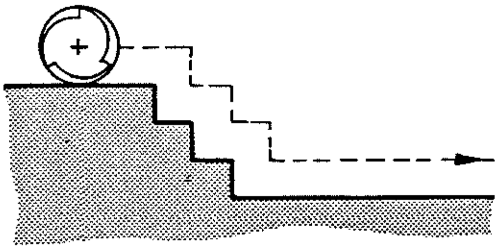
<u>steps</u>
tool radius larger than
contour radius



execution with G68



execution with G69



6. APPENDIX

PROGRAMMING KEY

G-CODES

Code		Functions	Group
0	XYZE	Positioning in rapid, with	a
1	XYZE	Linear interpolation at programmed feedrate	a
2	xxR	Circular interpolation, clockwise, 2 axes	a
3	xxR	Circular interpolation, counter-clockwise, 2 axes	a
4	F	Dwell in seconds	
5	xxx	Circular interpolation with tangential entry, 2 axes	a
6	XYZE	Linear interpolation in rapid with extended IN POS range	a
17		Plane selection X/Y	b
18		Plane selection Z/X	b
19		Plane selection Y/Z	b
20	XY	Setting pole for polar coordinates	
21	P	Subprogram call-up depending on I/F signal	
22	PL	Subprogram call-up, unconditional	
23	PL	Jump to program label depending on I/F signal	c
24	P	Jump to program label, unconditional	c
25	XYZE	Field limitation, setting minimum values	d
26	XYZE	Field limitation, setting maximum values	d
27	XYZE	Cancelling field limitation	d
36	-	Scale factor switching	e
38	xx	Switch on programmable mirroring	e
39	xx	Switch off programmable mirroring	e
40		Cancelling tool radius compensation	c
41	xx	Tool radius compensation to the left of the path	c
42	xx	Tool radius compensation to the right of the path	c
53		Cancel zero shift	f
54 to		Switch on zero shift	f
59	XYZE		
61	XYZE	'In Position' function on	g
62	XYZE	'In Position' function off	g
63		Feedrate and spindle speed set to 100%	h
64		Feedrate applies to contour on circular contours	i
65		Feedrate applies for tool centre path	i
66		Feedrate/spindle speed can be modified via pot.	h
68		Auxiliary arc on outside corners	j
69		Intersection on outside corners	j
74		Referencing	
75		Measuring probe	
80		Cancel fixed cycles G81 to G89	k
81	V	Drilling, centering	k
82	V	Boring with dwell	k
83	V	Deep hole drilling with positioning movements in rapid	k
84	V	Tapping with dwell	k
85	V	Boring with dwell/oriented spindle stop	k
86	V	Reaming	k
87	V	Thread milling	k
90	XYZE	Input in absolute dimensions	l
91	XYZE	Input in incremental dimensions	l
92	XYZE	Setting position stores	c
	S	Setting top limit for spindle speed	d
93	S	Time programming	
94	F	Feedrate direct in mm/min	m
95	F	Feedrate in mm/rev	m
96	S	Automatic calculation of cutting speed	n
97	S	Direct spindle speed programming	n
99		Subprogram end	
800to		Customer cycles: call-up via G-functions	
869		with corresponding numbers	

Group identifications a to n: Functions of the same group exclude one another.

G-CODES 3-digit

Code	Function	Group
	Contour Cycles	
890 V	intersection circle/circle	0
981 V	intersection line/circle	0
892 V	rounding corners (3 points)	0
893 V	rounding corners (2 angles)	0
894 V	chamfering	0
895 V	calculation of end point of arc	0
896 V	transition point arc/arc tangential	0
897 V	end point of straight line	0
898 V	intersection line/line	0

Machine specific G-codes (cycles)

Code Call-up	Function

M-CODES	**system specific functions**
Code	Internal effect
M0	program stop after execution of the block
M2	main program end, cycle end
M3 / M13	main spindle on CW / coolant on
M4 / M14	main spindle on CCW / coolant on
M5	main spindle stop / coolant off
M6	call-up of the automatic tool change cycle (cycle 77)
M19	orientation of main spindle to fixed position
M19(S. .)	orientation of main spindle to programmable position (degrees)
M21	call-up of MTB cycle 76
M22	call-up of MTB cycle 75
M30	program end with return to beginning (continuation with Cycle Start)
M40	automatic gear range selection
M41-44	selection of fixed gear range 1 to 4
M98	SINGLE BLOCK command is not accepted
M99	SINGLE BLOCK command is possible, i.e. the effect of M 98 is cancelled

MACHINE SPECIFIC M-FUNCTIONS

Code	Function

Parametric Functions

Instruction	Function	CR set	Time
V1 = n	load a numerical value	X	
$X = V_n, m = V_n$ m = XYZEIJKADGFRST	execution instruction		
$V_n = X, V_n = p$ p = XYZEIJKADFRST	transfer active data		
V1 = V2 + V3 (V1 = V1 + 10)	addition	X	
V1 = V2 - V3 (V1 = V2 - 12)	subtraction	X	
V1 = V2 * V3 (V1 = V2 * 10)	multiplication	X	
V1 = V2/V3 (V1 = V2/2)	division	X	
V1 = V2	copy	X	
V1 = SQR V2	square root	X	
INC V1	increment value, delete digits after decimal point	X	
DEC V1	decrement value, delete digits after decimal point	X	
V1 = SIN V2 (degrees)	sine (360° ≤ V2 ≤ 360°)	X	
V1 = COS V2 (degrees)	cosine (-360° ≤ V2 ≤ 360°)	X	
V1 (degrees) = ATG V2	arc tangent	X	
BSR V1 (BSR P5)	jump to subprogram (label 5) with no. V1		
BRA V1 (BRA P5)	jump to label no. V1 (label 5)		
BEQ V1 (BEQ P5)	jump to label no. V1, (label 5) **if CR = 0**		
BNE V1 (BNE P5)	jump to label no. V1, (label 5) **if CR=0**		
BGT V1 (BGT P5)	jump to label no. V1, (label 5) **if CR>0**		
BLT V1 (BLT P5)	jump to label no. V1, (label 5) **if CR<0**		
BGE V1 (BGE P5)	jump to label no. V1, (label 5) **if CR≥0**		
BLE V1 (BLE P5)	jump to label no. V1, (label 5) **if CR≤0**		

Note: CR = condition register; time = execution time in ms

Instruction	Function	CR set	Time
COR = V1 R = V 2 L = V3 DR = V4 S = V5 (COR = T1)	load tool no. V1 with values		
COR = T10 V1 = R V2 = L . . (COR = T10)	copy values from tool no. 10		
TRF = V1 X = V2 Y = V3 Z = V4 E = V5 (TRF = G54)	load zero shift no. V1 with values		
TRF = G54 V1 = X V2 = Y V3 = Z V4 = E (TRF = G54)	copy values from the G54 table		
TRF = G20 V1 = X V2 = Y. .	copy active pole		
TST V1	compare V1 with 0. set CR accordingly	X	
TST G1, TST G_n	CR = 0 if G01 active	X	
n=0-3,17-19,36,39,53-59,62,63, 65,66,90,93,94,95,97	CR = 0 if G_n active	X	
TST M41, TST M_n	CR = 0 if M41 active	X	
n = 3, 4, 5, 13, 41 - 44	CR = 0 if M_n active	X	
TST QX, TST Q_n	CR = 0 if X-axis mirrored	X	
n = X,Y,Z,E	CR = 0 if $_n$-axis mirrored	X	
TST QM	CR = 0 if metric dimensions	X	
TIM V1	record time from program start in seconds		
POS X (Y, Z, E)	axes traverse with external command		
STV	updating variables		

Axis Information

Format: +/- 7 digits, for instance 1.234567 or 123456.7

X -	X-axis	(mm/inch)
Y -	Y-axis	(mm/inch)
Z -	Z-axis	(mm/inch)
E -	E-axis	(mm/inch/degrees)
I -	centre of circle (X-direction)	(mm/inch)
J -	centre of circle (Y-direction)	(mm/inch)
K -	centre of circle (Z-direction)	(mm/inch)
R -	radius	(mm/inch)
D -	vector length (polar coord.)	(mm/inch)
A -	angle (polar coord.)	(degrees)

M-functions / Auxiliary Functions	**M** (0. .99)	M-function (M0, 2,3,4,5,6,13,14,19,21,22,40, 41,42,43,44,98,99 have a predetermined internal effect)
	T xx xx	(0. .99) tool number (output as location number) (0. .48) compensation group (activates tool length compensation)
	F (0.001 . . 120000)	feedrate (mm/min) or (mm/rev) time (sec)
	S (0. .9999)	spindle speed (rpm)
Subprograms and Jumps	**$** (0. . . 99)	jump address or beginning of subprogram
	P (0. . .99)	SBP number / label number (used in call-up)
	L (0. . .99)	number of SBP repetitions (used in call-up)
Special Characters	(. . . . .)	texts and comments
	N (1. . .9999)	block number
	V (1. . .99 and A. . .Z)	CPC variables
Control Characters	**STX** -	Start of Text (beginning of a data block such as a part program)
	ETX -	End of Text (end of a data block, such as a tool table)
	EOT -	End of Transmission (end of the transmission of one or several data blocks)
	CR LF -	Record Separator (separates two records, such as 2 NC blocks).

ASCII - Set of Characters

ASCII Character	Parity Bit	7-Bit Code	ASCII Character	Parity Bit	7-Bit Code	Meaning of the Character
A	0	1000 001	NUL	0	0000 000	0
B	0	1000 010	SOH	1	0000 001	start of header
C	1	1000 011	STX	1	0000 010	start of text
D	0	1000 100	ETX	0	0000 011	end of text
E	1	1000 101	EOT	1	0000 100	end of transmission
F	1	1000 110	ENQ	0	0000 101	enquiry
G	0	1000 111	ACK	0	0000 110	positive acknowledgement
H	0	1001 000	BEL	1	0000 111	bell
I	1	1001 001	BS	1	0001 000	back space
J	1	1001 010	HT	0	0001 001	horizontal tabulator
K	0	1001 011	LF	0	0001 010	line feed
L	1	1001 100	VT	1	0001 011	vertical tabulator
M	0	1001 101	FF	0	0001 100	form feed
N	0	1001 110	CR	1	0001 101	carriage return
O	1	1001 111	SO	1	0001 110	shift out
P	0	1010 000	SI	0	0001 111	shift in
Q	1	1010 001	DLE	1	0010 000	data link escape
R	1	1010 010	DC1	0	0010 001	DC on
S	0	1010 011	DC2	0	0010 010	control 2
T	1	1010 100	DC3	1	0010 011	DC off
U	0	1010 101	DC4	0	0010 100	control 4
V	0	1010 110	NAK	1	0010 101	negative acknowledge
W	1	1010 111	SYN	1	0010 110	synchro
X	1	1011 000	ETB	0	0010 111	end of transmission block
Y	0	1011 001	CAN	0	0011 000	cancel
Z	0	1011 010	EM	1	0011 001	end of medium (paper)
			SUB	1	0011 010	substitute
0	0	0110 000	ESC	0	0011 011	escape (code switching)
1	1	0110 001	FS	1	0011 100	file separator
2	1	0110 010	GS	0	0011 101	group separator
3	0	0110 011	BS	0	0011 110	block separator
4	1	0110 100	US	1	0011 111	unit separator
5	0	0110 101	SP	1	0100 000	space
6	0	0110 110	!	0	0100 001	
7	1	0110 111	"	0	0100 010	
8	1	0111 000	#	1	0100 011	
9	0	0111 001	$	0	0100 100	
			%	1	0100 101	
a	1	1100 001	&	1	0100 110	
b	1	1100 010	'	0	0100 111	
c	0	1100 011	`		1100 000	
d	1	1100 100	(	0	0101 000	
e	0	1100 101	)	1	0101 001	
f	0	1100 110	*	1	0101 010	
g	1	1100 111	+	0	0101 011	
h	1	1101 000	,	1	0101 100	
i	0	1101 001	-	0	0101 101	
j	0	1101 010	.	0	0101 110	
k	1	1101 011	/	1	0101 111	
l	0	1101 100	:	0	0111 010	
m	1	1101 101	;	1	0111 011	
n	1	1101 110	<	0	0111 100	
o	0	1101 111	=	1	0111 101	
p	1	1110 000	>	1	0111 110	
q	0	1110 001	?	0	0111 111	
r	0	1110 010	@	1	1000 000	
s	1	1110 011	[	1	1011 011	
t	0	1110 100	\	0	1011 100	
u	1	1110 101	]	1	1011 101	
v	1	1110 110	^	1	1011 110	
w	0	1110 111	-	0	1011 111	
x	0	1111 000	{	0	1111 011	
y	1	1111 001	|	1	1111 100	
z	1	1111 010	}	0	1111 101	
			~	0	1111 110	
			DEL	1	1111 111	

OUTPUT OF ERROR MESSAGES

Definition

The CC 100 M will transmit errors recognized internally to the interface controller. The error messages are output in coded form, one digit to indicate the error message group (0-2) and two further digits to indicate the error number (01 - 88).

Example

0 3 9 E axis must be programmed alone

error number 39 - Text: E axis must be programmed alone

error message group 0

Soft key operation for error display

1. **EDIT**	Incorrect program blocks are automatically displayed with error numbers and descriptions.
2. **MACHINE**	Incorrect entries in MDI are displayed automatically with error numbers and descriptions.
3. **AUTOMATIC**	Incorrect program blocks, which are not recognized until RUN operation, cause program stop and a general error signal. To obtain information about the type of error you need to switch into INFO mode; there the error number and the description will be displayed.

Error message group 0: **A14 D0**

Data	Meaning
1	syntax error
2	syntax error
3	
4	system error H-Size overflow
5	system error N-H-Size overflow
6	system error L-H-Size overflow
7	system error R-Size overflow
8	system error D-Size overflow
9	
10	repetition (L) without subroutine call
11	cutter comp. programmed without tool
12	this G code must be alone in block
13	this G/M code is not allowed with TEACH IN or MDI
14	max. 3 axes or A, D allowed
15	R or I, J, K not allowed
16	max. 2 axes out X, Y, Z allowed
17	TIM, COR, or TRF must be alone in block
18	max. 4 axes with value allowed
19	max. 4 without value allowed
20	max. 2 axes out of X, Y, Z, E or A, D allowed
21	enter Dwell time (F)
22	unadmissible G number
23	enter S without sign.
24	value too large
25	with D, F, or R, zero not allowed
26	repetition of address not allowed
27	max. 2 coordinates out of I, J, K allowed
28	no radius programmed with polar coordinates
29	max. 2 axes with polar coordinates (A, D)
30	max. 3 axes R or I, J, K
31	enter jump target (P)
32	jump target (P) allowed with G21/22/23/24
33	G code required with P or L
34	axis without value not allowed
35	only integer value
36	this M code must be alone in block
37	test not allowed
38	unadmissible tool number
39	E axis must be programmed alone
40	input range 1 to 127
41	with G 96, S value not allowed
42	with G 92, S value not allowed
43	max. 4 axes or A, D allowed
44	axis value not allowed
45	max. 2 digits with $,P,L or M

Error message group 0: **A14 D0**

Data	Meaning
46	only 2 or 4 digits with T
47	too many digits
48	max. 1 axis with value allowed
49	max. 4 digits with S
50	sign. not allowed
51	input range 0.001 to 5
52	enter value
53	Y(es) or N(o) required
54	
55	input range 0 to 999
56	input range 1 to 720
57	input range 0 to 4
58	input range 0 to 20000
59	input range 0 to 50000
60	input range 0 to 90000
61	input range 1 to 1000
62	input range -9999 to 9999
63	input range 0 to 100
64	input range 0 to 3
65	input range 0 to 359.999
66	input range 0 to 5
67	input range 1 to 100
68	E not allowed
69	F not allowed with G0
70	only X, Y, Z allowed
71	only P, L allowed
72	only X, Y, Z, E allowed
73	only X, Y, Z, E or M, T allowed
74	only X, Y, Z, E or F, S, M allowed
75	only X,Y, Z, E or S allowed
76	input range -100 to 100
77	DR value = -10 % to +10 % of R (1 mm or 0.05 i max)
78	input range 1 to 50000
79	with TEACH IN or MDI P, L not allowed
80	M 19 must be programmed alone or with S
81	incorrect input of variables
82	incorrect variable number
83	input range12 to 48
84	input range 256 to 32767
85	M 06 must be programmed alone or with tool number
86	input range -10000 to 1 or to 10000
87	input range 0 to 9999
88	address modification must be alone in block
89	message has to start with "("

Error message group 1: **A14 D1**

Data	Meaning
1	no previous movement before G5
2	full circle programming not allowed
3	radius value null or missing
4	negative root
5	G code not allowed in automatic mode
6	bad polar radius programmed
7	G95 and M5 or S value = 0
8	
9	G5 not allowed following G0
10	
11	incorrect circle defintion
12	centre coordinates incorrect
13	programmed radius was rounded
14	tool radius too large (1)
15	tool radius too large (2)
16	tool radius too large (3)
17	tool radius too large (5)
18	no intersection possible parallel lines
19	no intersection possible line / circle
20	no intersecition possible circle / circle
21	tool radius too large (4)
22	the circles are not tangent
23	M 30 or M 2 required
24	jump target not found
25	max. 10 subroutine levels
26	cycle does not exist
27	G99 and no subroutine acitve
28	M2 or M30 seen with cutter comp. active
29	G code not allowed with cutter comp. active
30	Highest spindle speed exeeded
31	1. gear range defined incorrectly
32	M3 or M4 missing
33	gear range unadmissible
34	G99 with subroutine or M2 with cycle
35	no feed programmed with G75/94/95/93
36	
37	
38	
39	input missing
40	rotary axis with circular interpolation
41	incorrect position programmed with E axis
42	cycle end is M2
43	subroutine end is G99
44	preset not allowed with active zero shift

The control will display the messages in clear text.

Error message group 1: **A14 D1**

Data	Meaning
45	cycle unadmissible with cutter comp.
46	unadmissible value for G code
47	G code unadmissible with mirror function
48	G code unadmissible with cutter comp.
49	
50	
51	
52	one movement missing for cutter comp.
53	
54	block modified or not executed due to cutter comp.
55	max. 2 axes out of X, Y, Z, E or A, D allowed
56	V95 must be 0 or 1
57	transfer not possible
58	reentry not allowed with G84
59	probe not triggered
60	unadmissible jump target
61	double definition of axis (polar)
62	max. 1 axis with G2/3/5 and polar programming
63	max. 3 axes with G2/3/5
64	G0/1/5 and radius or I, J, K not allowed
65	G2/3 with radius and I, J, K not allowed
66	no new cutter comp. with G2/3/5
67	G21/23 with cutter comp. not allowed
68	cutter comp. not allowed without tool number
69	G40/41/42 not allowed with G2/3/5
70	no G2/3/5 following a zero shift
71	
72	G96 not allowed with S value
73	spind. speed calcul. not possible, tool radius = 0
74	G92 not allowed with G41/42/T
75	G code not allowed with cut. or length comp.
76	new plane not allowed with cutter comp.
77	V95 must equal 2 or 3
78	division by zero
79	coordinates do not comply with active plane
80	no tool active
81	unadmissible tool number
82	unadmissible G number
83	V91 must equal 1 or 2
84	this zero shift is already active
85	DR value = -10 % to +10 % of R (1 mm or 0.05 i max.)
86	spindle orientation not possible
87	calculation not possible
88	angle range -180 to +180 deg.

The control will display the messages in clear text.

Error message group 2: **A14 D2**

Data	Meaning
1	99 programs exist
2	memory full
3	memory too small for jump target table
4	check sum error
5	undefined jump target
6	parity memory
7	duplication of jump target
8	69 CYCLES exist
9	memory too small to copy
10	file protected
11	
12	unadmissible file
13	file already exists
14	device not ready
15	parity error
16	incorrect data format
17	incorrect baud rate
18	timeout period expired
19	no corresponding file type
20	TEACH IN
21	movement not allowed with E, 2 blocks created
22	memory error, switch off
23	reference cycle does not exist
24	interruption, abort with clear block
25	inch / metric selection incorrect
26	no corresponding cycle
27	undefined key
28	reference not allowed with length comp.
29	movement not allowed, 2 blocks created
30	
31	warning sent by PLC
32	too many characters for one block
33	bad value for tool table size
34	size of memory changed, memory cleared
35	no machine reference, send axes to reference
36	
37	
38	
39	circle calculation not possible
40	limit

Data	Meaning
41	emergency stop
42	servo error
43	measuring system: marker missing
44	measuring system: not connected
45	measuring system: pulse is lost
46	measuring system: no feedback
47	bad axes parameters
48	gearbox not OK
49	interpolator stop error
50	axis error
51	code:
52	T (s)
53	/mn
54	/rev
55	conflict between hardware and software, NC stopped