BWO Control System 900

CNC Programming 900

1. Introduction

1.1 Hardware construction

The control system 900 consists of two components:

- chassis with slots for the plug-in modules

- operating panel in the housing or with planner front for the cabinet installation

1.1.1 Modules

Environment conditions

Operating temperature	0 to +50 °c
Storage temperature	-10 to +60 °c
Dampness	10 to 90 %, non-condensing

Chassis with 1, 2, 4 and 8 slots for the accommodation of the modules. The individual chassis can by plugging together to be line up.

Back-up battery	PB	Buffering the RAM Memory (+3,6V)
Bus termination	BA	Termination of the internal bus system

Power packs	NG 4	+5V / 4A	and ±15V / 0,2A	Chassis with 1 slot
	NG 8	+5V / 8A	and ±15V / 0,5A	Chassis with 1 slot
	NG 16	+5V / 16A	and ±15V / 1A	Chassis with 2 slots
	NG 24	+5V / 24A	and ±15V / 1,5A	Chassis with 2 slots

Central processing units

CNC 32Bit / 64Bit ETH	for all CNC and PLC Functions, user memories 1,5MB / 3MB for NC programs and parameters, flag memory 60kB.
CNC 32Bit / 64Bit	for all CNC Functions as well as alternatively for integrated / all PLC functions, user memories 1,5MB / 3MB for NC programs and parameters, flag memory 60kB.
PLC 16k / 64k	High performance RISC processor for all PLC Functions as well as NC functions with POS modules, Flag memory 60kB, Program memory for 16k / 64k of instruction.

1.1.1 Modules (continuation)

Axis modules

Axis module	AAZ1	passively, for 1 analog triggered axis, incremental
Axis module	AAZ2	passively, for 2 analog triggered axes, incremental
Axis module	AAZ4	passively, for 4 analog triggered axes, incremental
Axis module	AZA2	passively, for 2 analog triggered axes, absolutely
Axis module	AZA4	passively, for 4 analog triggered axes, absolutely
Axis module	ASM	passively, for 3 stepping motors
Axis module	SERC	aktiv, for 8 digitally triggered axes with SERCOS LWL Interface

Digital I/O modules max. ones 1024 in and outputs

Input module	EK	32 inputs
In / output module	AEK	16 inputs, 16 outputs (0.5A)
Output module	AK	32 outputs (0,5A)
Output module	AK2	16 outputs (2A)
Relay module	AKR	16 outputs

Analog I/O moduls

Digital-analog converter	DAW2	two-way
Digital-analog converter	DAW4	fourfold
Analog-digital converter	ADW4	fourfold

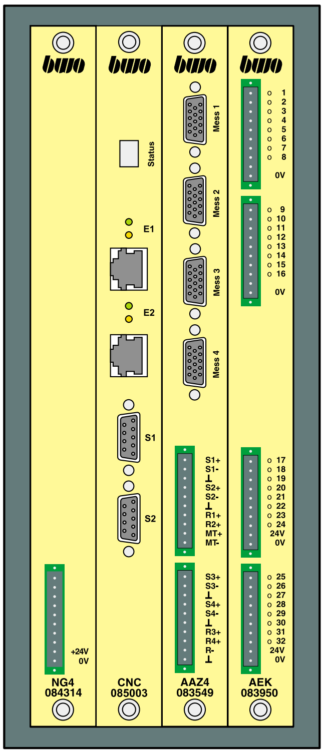
Communication modules

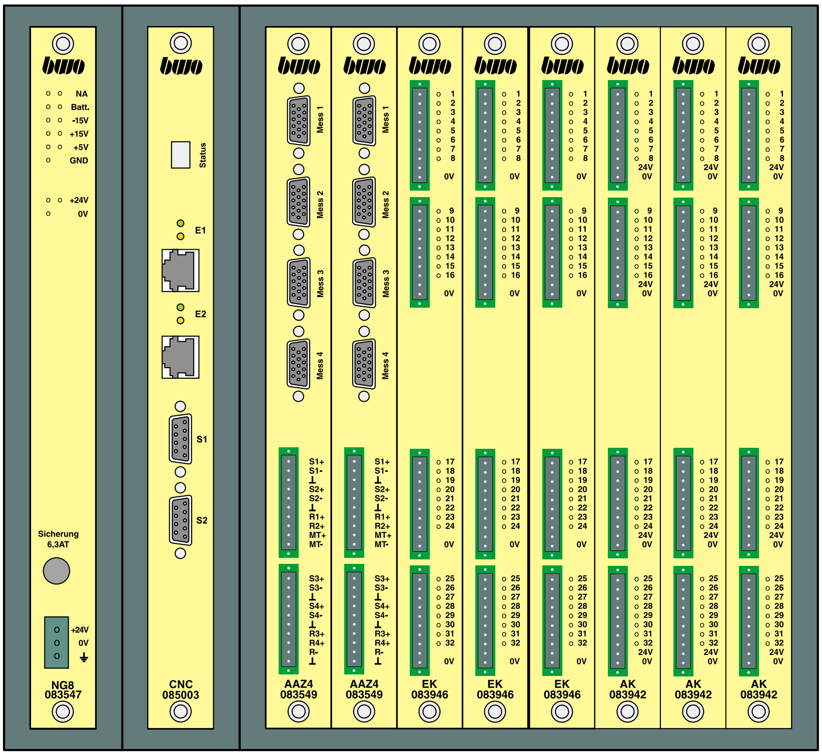
Logs	COM	programming processor module for special applications
Ethernet connection	ETH	connection of the system PLC 900 to the Ethernet
Connection of peripheral	AS I	peripheral connection of peripheral device
CAN bus connection	CAN	connection to the CAN bus for drives and I/O

Addition modules

Koppelmodul	KOP	to couple of two groups of chassis (PLC CPU)

1.1.2 Examples of CNC controllers





Example of a CNC controller of 4 axes

Chassis with:

4 slots for Power pack NG4,

Central processing unit CNC (32bit),

Axis module AAZ and
AEK module.

1 Back-up battery
(left at the chassis beside the power pack)

1 Bus termination
(on the right attached at the chassis)

Example of a CNC Controller of medium performance of 8 axes

Chassis with:

1 slot for Power pack NG8,

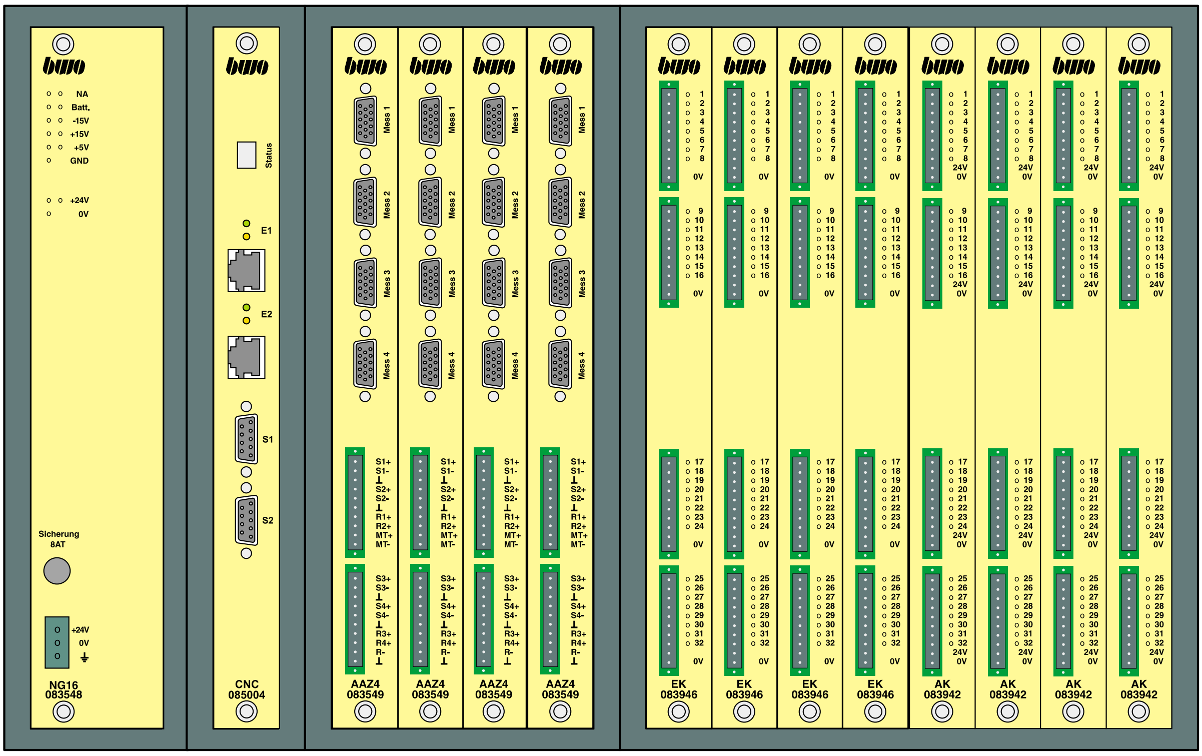
1 slot for central processing unit CNC (32bit),

8 slots for Axis modules AAZ and
I/O modules.

1 Back-up battery
(left at the chassis beside the power pack)

1 Bus termination
(on the right attached at the chassis)

1.1.2 Examples of CNC controllers (continuation)



Example of a CNC Controller of high performance of 16 axes

Chassis with:

2 slots for Power pack NG16,

2 slots for Central processing unit CNC (64bit),

4 slots for Axis modules AAZ,

8 slots for I/O modules.

1 Back-up battery (left attached at the chassis beside the power pack)

1 Bus termination (on the right attached at the chassis)

1.1.3 Operating panels

Operating panel CNC910

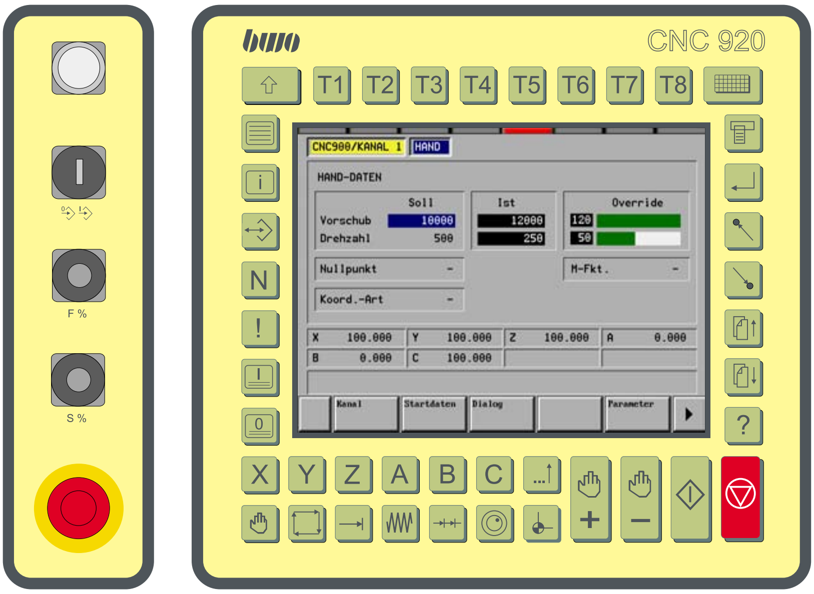
- LCD-Bildschirm TFT	6.5 “
- Resolution	640 x 480
- Colours of 4096	256
- Touch screen with resolution	1024 x1024
- Run-time memory DRAM	16MB
- Flash disk memory	8MB

- 42 function keys, of it 15 freely shapable

- PLC Keys with display on the LCD Display

- connections	1 Ethernet RJ45 1 serial interface
- machine operating panel (integrated)	1 emergency stop 1 key-operated switsch 2 potentiometers

Operating panel CNC920



- LCD-Bildschirm TFT	10.4“
- Resolution	640 x 480
- Colours of 4096	256
- Touch screen	1024 x1024
- Run-time memory DRAM	16MB
- Flash disk memory	8MB

- 42 function keys,
  of it 15 freely shapable

- PLC keys with display
  on the LCD display

- Connections	1 Ethernet RJ45 1 serial interface
- Machine operating panel (separately)	1 emergency stop 1 key-operated switsch 2 potentiometers 1 illuminated push button

1.1.3 Operating panels

Operating panels RC910

for the mobile application

in the versions (from above):
Standard,
with handwheel,
with joystick

- LCD LCD-Bildschirm TFT	6.5 “
- Resolution	640 x 480
- Colours of 4096	256
- Touch screen with resolution	1024 x1024
- Run-time memory DRAM	16MB
- Flash disk memory	8MB

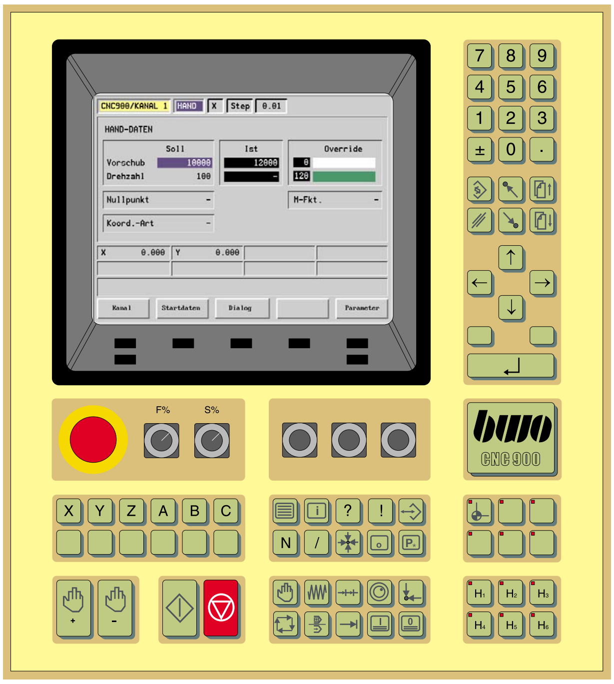
- 42 function keys, of it 15 freely shapable

- PLC Keys with display on the LCD Display

- connections	1 Ethernet RJ45 1 serial interface
- machine operating panel (integrated)	1 emergency stop 1 key-operated switsch 2 potentiometers

1.1.3 Operating panels

Operating panel CNC 900



Wide Color LCD TFT,
VGA diagram (640 x 480),

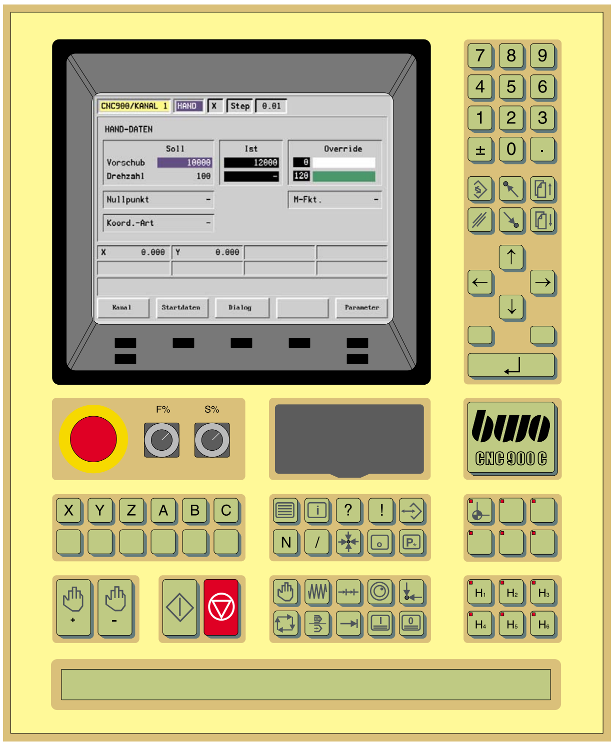
512kB EEPROM for user specific
masks and 256k flash drive,

Integrated function keys, cursor keys,
numeric key block,

Separate keys for axis selection,
keys for machine functions,
Peripheral device interface,
1 serial interface,

emergency-stop key,
2 potentiometers and key switches,
2 control elements (optional)

Operating panel CNC 900C



Wide Color LCD TFT,
VGA diagram (640 x 480),

CPU Pentium compatibly, ≥200MHz,
RAM Memory ≥16MB,
Hard disk ≥ 3,2GB,
Floppy disk drive 3.5" 1.44MB,
ASCII foil keyboard (optional),

Integrated function keys, cursor keys,
numeric key block,
Separate keys for axis selection,
keys for machine functions,

Peripheral device interface,
1 parallel and 2 serial interfaces,
2 free slots,
Emergency-stop key and
2 potentiometers

1.2 Software construction

1.2.1 Overall view

The required software consists of several components:

- PLC900 for the programming of the adaptation program
  (manual for the machine manufacturer)

- PROMA to the programming of the graphic control surface and display
  (manual for the machine manufacturer)

- CNC900, the CNC Core

- WINBV for operating system administration with CNC910 / CNC920 / RC910 / CNC ETH (10.1)

- BV.EXE for operating system administration with CNC 900 / CNC 900 C (10.2)

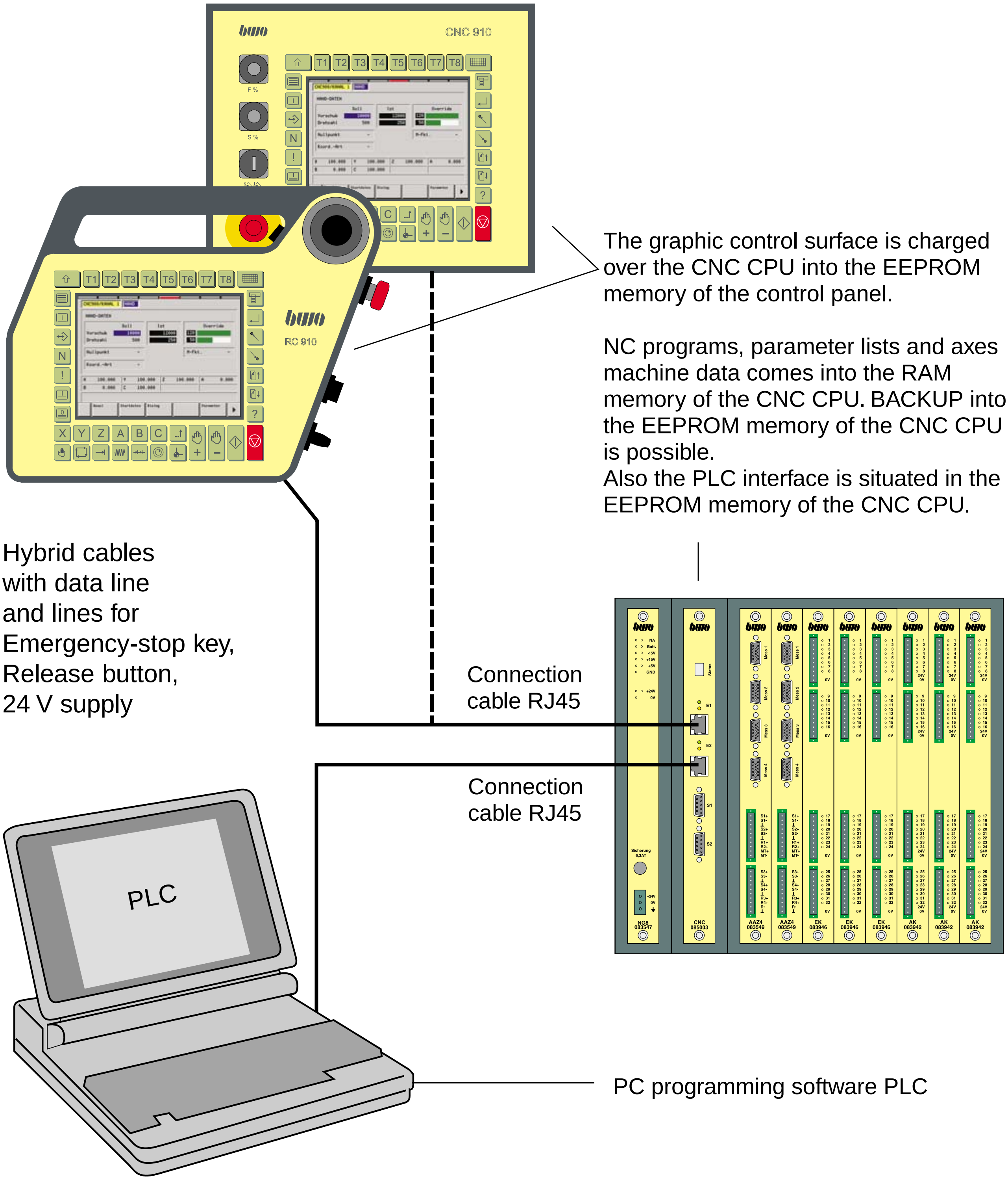
- DLL for operating system extension

- NCARC for the archiving of the NC programs (10.3)

in this manual **operation and NC programming** for the CNC user

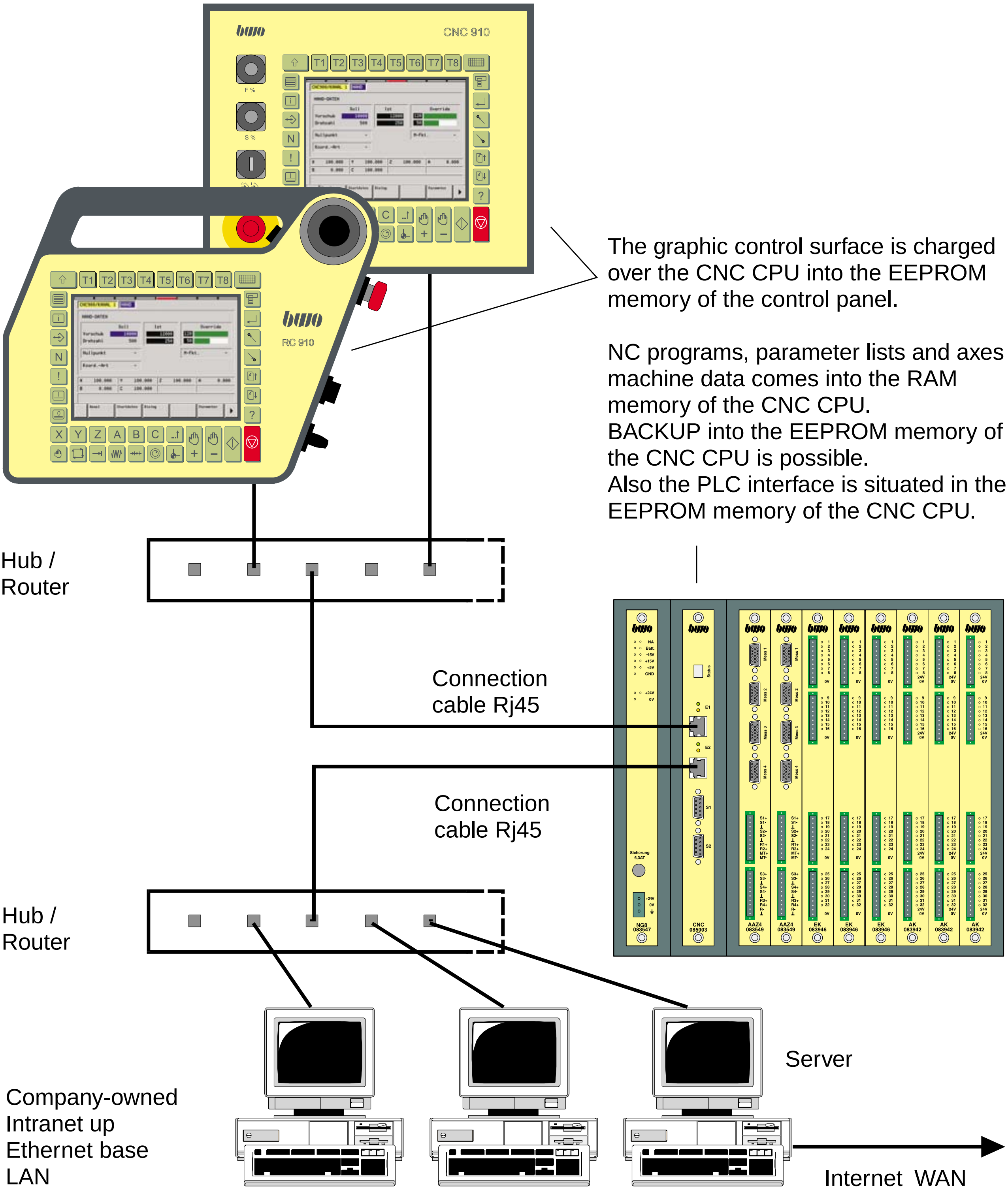
1.2.2 Software input with operating panels CNC910 / CNC920 / RC910

Configuration with PC



1.2.2 Software input with operating panels CNC910 / CNC920 / RC910

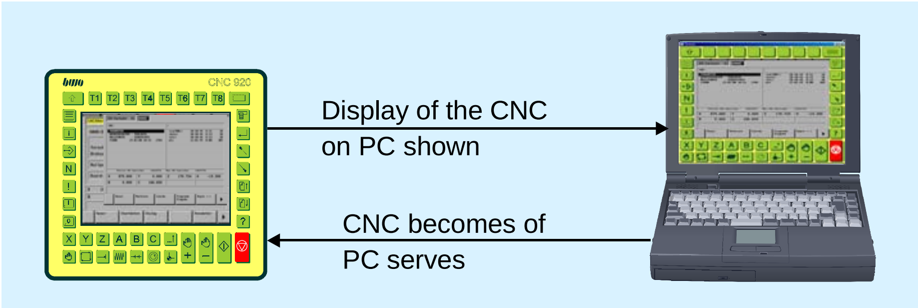
Configuration with networks



1.2.3 Software for data tansfer between CNC Operating panel and PC

WINBV

Software on a PC to controlling of the CNC.



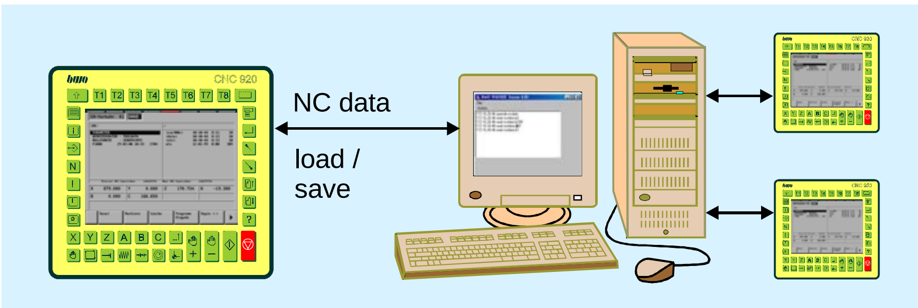
The display of the CNC is illustrated on a PC. The control can be served then from the PC.

Thus can be implemented:
• Diagnosis locally
• On-line remote diagnostics
• Operating system care
• Data adminstration
• Data protection
• NC archiving

Details in addition in this manual paragraph 10.1.

SERVER

Software on a PC for data transfer with the CNC.

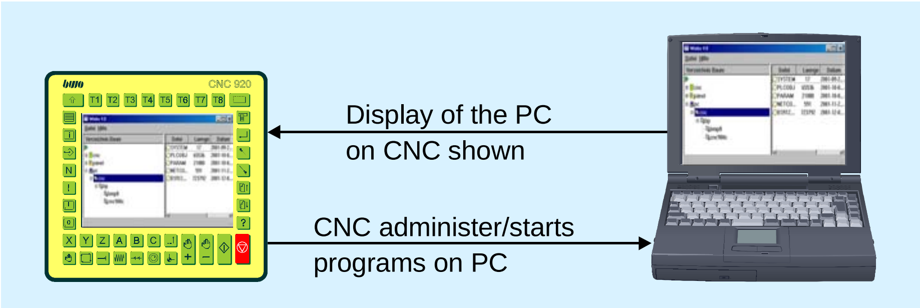


By the CNC can be accessed the drive assemblies of a server. Several controls can be connected at a server and load NC data from the server or save on the server.

1.2.3 Software for data tansfer between CNC Operating panel and PC

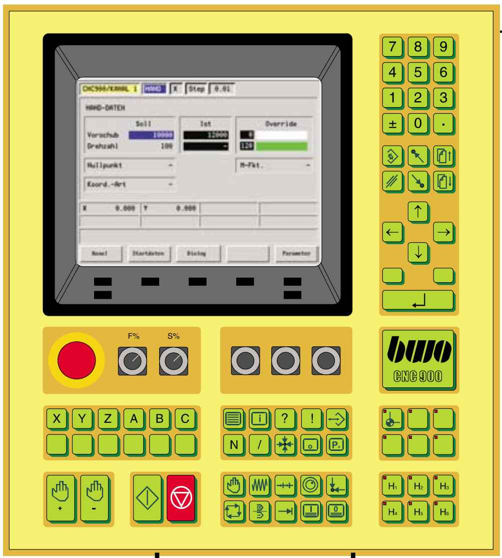
WINTERM

Software on the CNC Operating panel to the control PC



On the CNC Operating panel the display PC is illustrated.
From the CNC Operating panel programs on the PC can be started and managed.

1.2.4 Software input with operating panel CNC 900



The graphic control surface is charged with PROMA into the EEPROM memory.

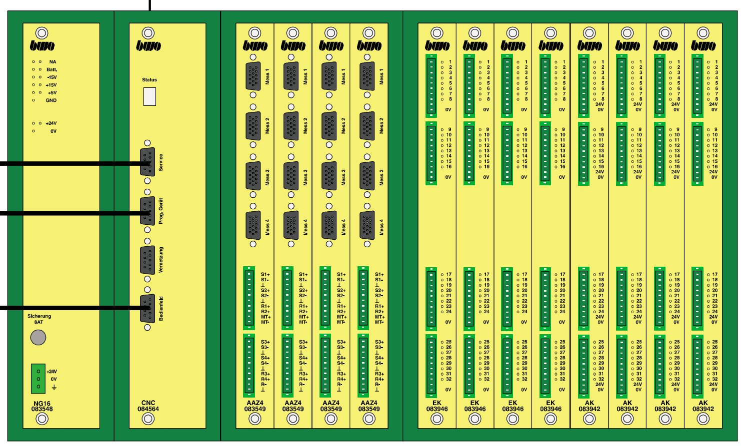
NC programs, parameter lists and axis machine data are charged into the RAM memory of the CPU. Backup into the EEPROM memory of the CPU is possible.

Update of the CNC system software is possible with the operating system management program BV.EXE. For this a transducer set (P-LINK and cable connection) is necessary.

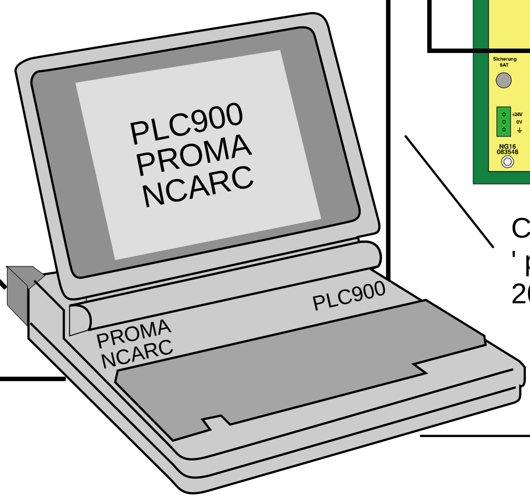
Also the PLC interface program is charged with PLC900 into the EEPROM memory of the CNC CPU.

Cable connection ' peripheral ' V24 interface

Cable connection ' operating panel '



Transducer set P-LINK 900 and cable connections ' service '



Cable connections ' prog. device ' 20mA interface

PC programming and data protection software

PLC900	PLC program
PROMA	Graphic control surface
NCARC	NC archive program

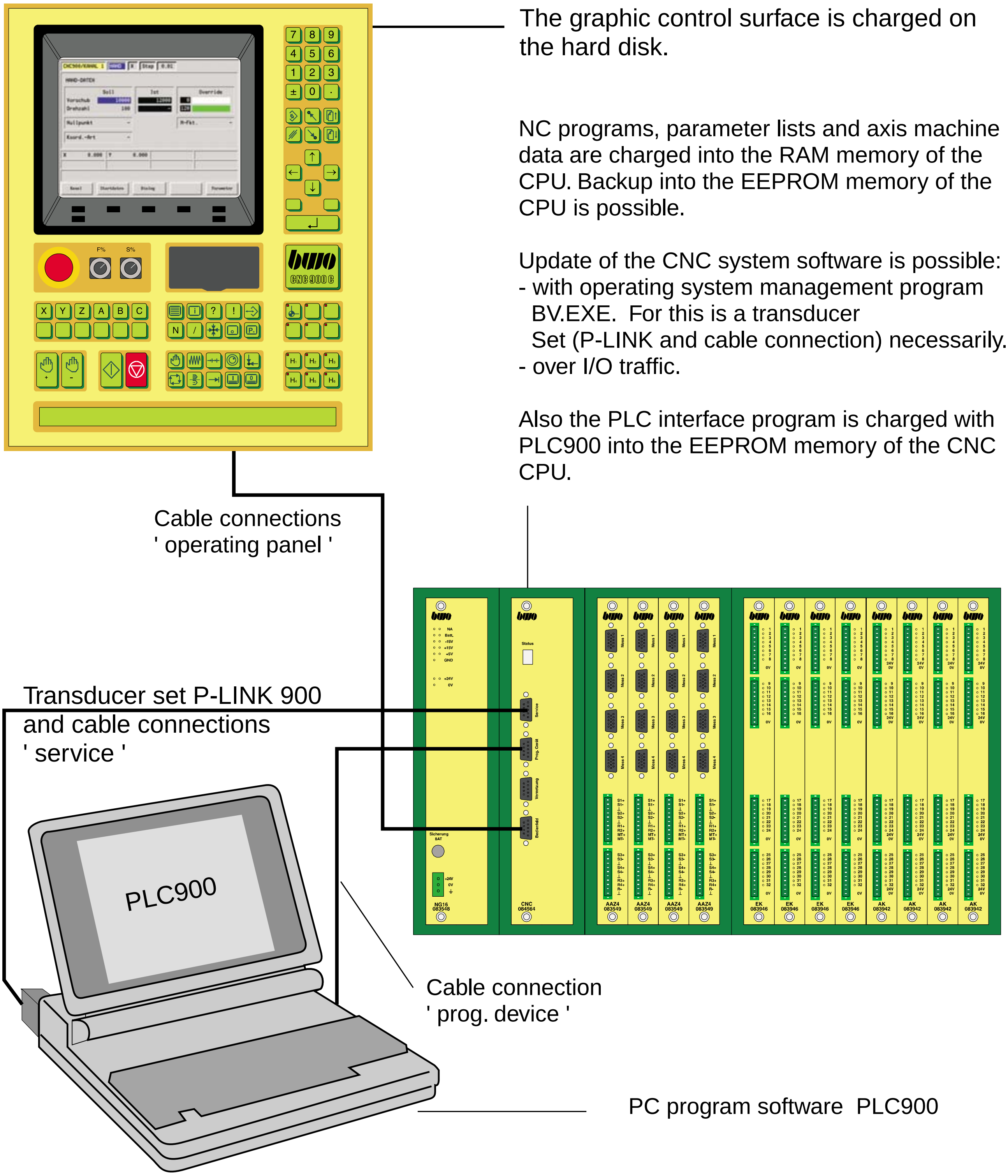
1.2.5 Software input with operating panel CNC 900 C

The graphic control surface is charged on the hard disk.

NC programs, parameter lists and axis machine data are charged into the RAM memory of the CPU. Backup into the EEPROM memory of the CPU is possible.

Update of the CNC system software is possible:
- with operating system management program BV.EXE. For this is a transducer Set (P-LINK and cable connection) necessarily.
- over I/O traffic.

Also the PLC interface program is charged with PLC900 into the EEPROM memory of the CNC CPU.



1.2.6 Software programs

Bootstrap loader (Boot software or Booter)

The bootstrap loader for the CNC900 is on the boat PROM (purchase number 084076) of the CNC module. The CNC module is delivered with the topical status of the bootstrap loader.

Operating system often commodity consisting of the files

- ' BS900 '	BWO standard operating system for CNC900
- ' DLL '	BWO library software for customized operating system extensions.

The operating system software for the CNC900 is on the FLASH EPROM of the CNC module. The CNC module is delivered with the topical status of the operating system often commodity.

NC program

In the buffered SRAM memory of the CNC module the following data are stored:

- NC programs
- parameter P
- machine data q or P

These data can be read in or output with I/O traffic over the operating panel (V.24 peripheral device interface). In addition the archiving program „ NCARC „ can be used.

Operating system management program BV.EXE

With BV.EXE valid operating system and library files can be burned in the Flash EPROM of the CNC module. Furthermore BV.EXE offers different auxiliary functions for the administration and archiving of operating systems, libraries as well as NC programs and CNC parameters.

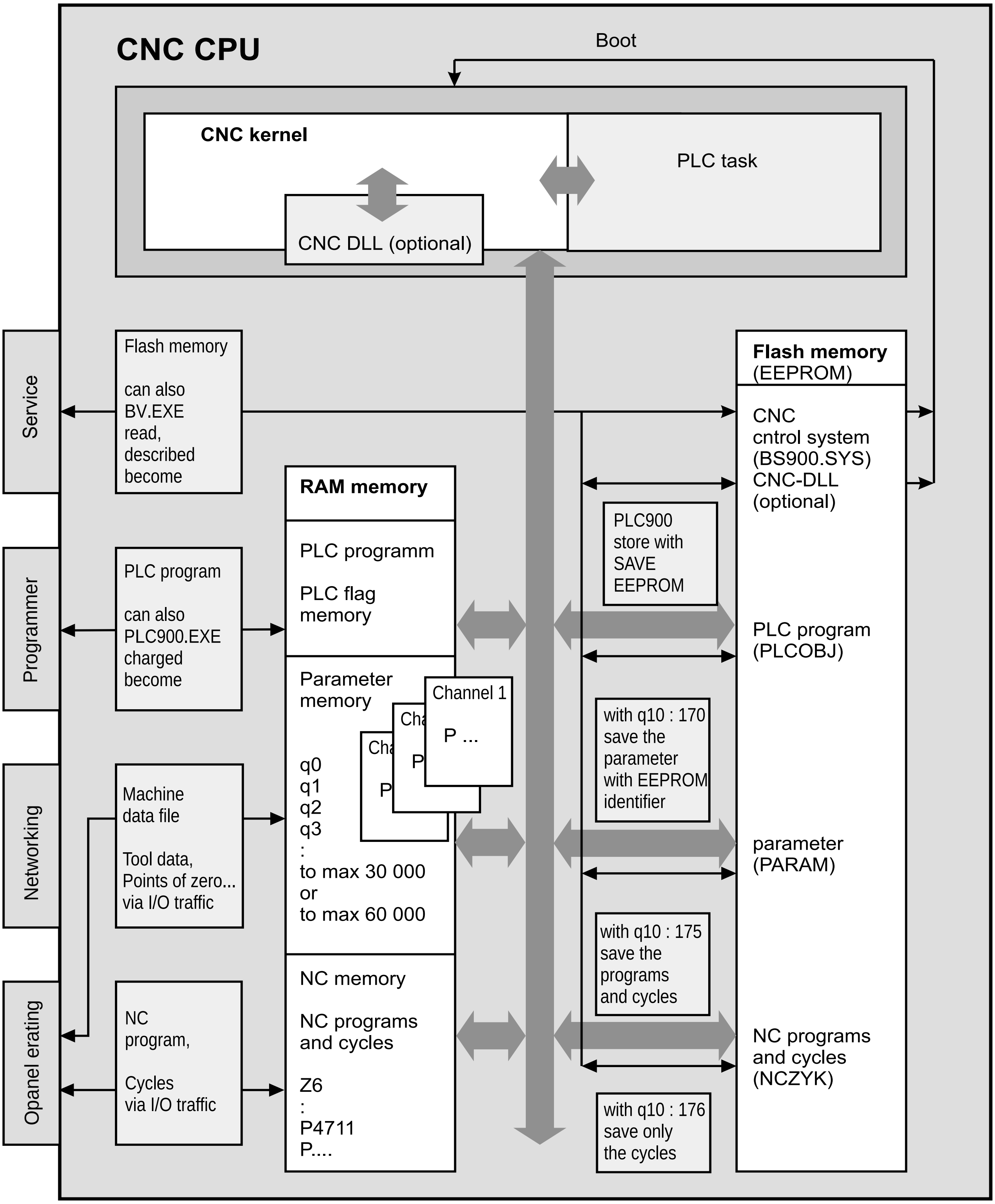
Programming software PLC900

With the programming software PLC900 can be created on a PC a machine-specific interface program.

Programming software PROMA

With the programming software PROMA can be created on an IBM-compatible PC (with VGA 640 x 480) a machine-specific graphic control surface. A control surface consists of several frames, which are summarized in a project.

1.2.6 Software programs (continuation)



1.3 Software - components / - version check / - update

Note: in the following text xxx the 3 digit version number always means

1.3.1 operating panels

Software components	**CNC900**	**CNC900C** BWO standard Installation in C:\CNC900C
Operating system	**CNC900.RCE**	**CNC900X.EXE** or **CNC900X.DLL** in connection with one machine-specific application
User surface / screen	**SHOW_E**	**SHOW_E**
		further one required files in C:\CNC900C : - RTM.EXE subroutine for CNC900X.xxx - DPMI16BI.OVL " - PROCEX.DLL " - CNC900X.CFG configuration file for CNC900X.xxx - PLC_EDIT.EXE text editor

Software version check

CNC900.RCE , **CNC900X.EXE** , **CNC900X.DLL** :
Dive with current control surface into the diagnostic display (soft keys F6 + F7 press at the same time, return with F1).
Here becomes with ' operating panel software: CNC900X.EXE/DLL: ' the time stamp of the operating panel software displayed.

SHOW_E :
Dive with current control surface into the diagnostic display (soft keys F6 + F7 press at the same time, return with F1).
Here becomes with ' status SHOW_E: ' Project name and time stamp displayed or
during switching on routine is displayed the SHOW_E project name and time stamp in the line with ' test operating panel '.

Software update

CNC900	bringing in **CNC900.RCE** or **SHOW_E** with PROMA.EXE or the operating system management program BV.EXE
CNC900C	copying of the system files into the work directory (BWO standard installation in C:\CNC900C)

1.3.2 CNC CPU

Software components

CNC CPU type :	083671 before index C (without FLASH file system)	083671 from index C (with FLASH file system)	084564
CNC-Betriebssystem	**BS900.xxx**	**BS902.xxx**	**BS910.xxx**
	All 3 files are packed together in **BSxxx.ZIP**.		
CNC-DLL (optionally and customized)	**DLL**	**DLL3264.xxx**	**DLL6464.xxx**
		Both files are together packed in **DLLxxx.ZIP**.	

Information to the FLASH file system (not with 083671 before index C): (see q10:501)

Contents of the FLASH file system with a CNC CPU 084564 look about as follows:
(I/O traffic picture)

——— FLASH-DIR ——————————————

SYSTEM -> BS910.SYS	01.01.1999
BS910.SYS	01.01.1999
PARAM	01.01.1999
Optionally :	
NCZYK	01.01.1999
PLCOBJ	01.01.1999
DLL64 -> DLL910.SYS	01.01.1999
DLL910.SYS	01.01.1999
PLC_BIG	01.01.1999
	date displays: Day, at that the file in the FLASH memory one stored (independently of the time stamp of the read in file).

1.3.2 CNC CPU (continuation)

Software components

SYSTEM ->	LINK on CNC operating system This entry is created automatically with the download of an operating system, and points to the CNC operating system.
BS910.SYS or BS910.SY1	CNC operating system (fixed term) Each CNC operating system is stored under this term. Note: With the download of operating systems via I/O traffic alternates extention between SYS and SY1.
PARAM	stored machine data (this file is created with q10:170)
NCZYK	stored NC programs / cycles (this file is created with q10:175/176)
PLCOBJ	PLC file only, if no external PLC is available
DLL64 ->	LINK on CNC-DLL this entry is created automatically with the download of an operating system, and points to the CNC DLL If this entry does not exist, then also no CNC DLL is merged.
DLL910.SYS or DLL910.SY1	CNC DLL (fixed term) Each CNC DLL is stored under this term Note: With the download of operating systems via EA traffic alternates extention between SYS and SY1.
PLC_BIG	Option for PLC program If PLC_BIG is available, becomes the PLC program memory size of 16k statements on 64k statements quadruples. (only with internal PLC in connection with PLC900X version 4.x)

1.3.2 CNC CPU (continuation)

Software version check

CNC operating system **: BS9...xxx,**
During switching on routine that becomes CNC operating system –
Status with date in the line with ‘ test BS ‘ displayed.
or
With current control surface in the parameter editor ab q12

q12	: version
q13	: test character
q14 ... q19	: time stamp

CNC DLL **: DLL...xxx,**
During switching on routine that becomes CNC DLL system –
Term with date in the line under ‘ test BS ‘ displayed.
or
With current control surface in the parameter editor starting from q22

q22	: version
q23	: test character
q24 ... q29	: time stamp

1.3.2 CNC CPU (continuation)

<u>Software update</u>

There are 2 possibilities, in order to execute an update of an operating system status / CNC DLL.

- with the operating system administration program BV.EXE

CNC operating system (see also documentation BV.EXE)

With the function ' FLASH ' - ' WRITING ' is selected **the file** BSxxx.ZIP and loaded into the FLASH Memory of the CNC CPU.
(BV looks itself up in the **BSxxx.ZIP** the suitable file to the available hardware. Naturally also **BSxxx.ZIP** would know beforehand unpacked to be become and then the suitable file **BS900.xxx, @@@BS902.xxx** or **BS910.xxx** into the FLASH memory be loaded.)
When storing in the FLASH the file name is changed! (e.g. with CNC_CPU 084564 after **Bs910.sys**.
The LINK **SYSTEM** is set on the charged file.

CNC DLL

The flow is the same as with the CNC BS update.
DLLxxx.ZIP is charged or, depending upon hardware, **DLL3264.xxx** or **DLL6464.xxx** .
When storing in the FLASH the file name is changed! (e.g. with CNC_CPU 084564 after **DLL910.SYS** .
The LINK **DLL64** is set on the loaded file.

1.3.2 CNC CPU (continuation)

Software update

- via I/O traffic

CNC operating system

(Only with CNC CPU 083671 starting from index C and 084564 in connection with the operating panel CNC900C and at least one CNC operating system version 090 of 21.10.99.)
In the EA traffic picture the suitable CNC operating system is marked and knocked against with softkey F5 the copying process. (Same process as NC file copy of hard disk into the NC memory.)
The loaded file is checked now for correctness and written into the FLASH memory. All of this is correct, after demand of the LINK is set, in order to use the loaded operating system with the next system start.
The file name, with that is stored the CNC operating system in the FLASH memory is
CNC_CPU 083671 starting from index C : **BS902.SYS**
CNC_CPU 084564 : **BS910.SYS**
The file extention alternates between SYS and SY1 ; i.e. the current LINK is set on **BS910.SYS**, becomes from safety reasons a file: **BS910.SY1**.

CNC DLL

The flow is the same as with the CNC BS update.
The file name, with that is stored the CNC DLL in the FLASH memory is
CNC_CPU 083671 starting from index C : **DLL902.SYS**
CNC_CPU 084564 : **DLL910.SYS**
The file extention alternates between SYS and SY1; i.e. the current LINK is set on **DLL910.SYS**, becomes from safety reasons a file: **DLL910.SY1.**
With q11:510 and q10:510 the current CNC DLL can be reset.

Notes

The time stamp in the FLASH file system refers to the time of the downloads!

The loaded operating system may not be packed (.ZIP)!

With q10:501 the flash contents can be checked. I.e. in the I/O traffic picture of the FLASH contents are displayed now instead of NC memory contents.

1.3.3 PLC CPU

Software components

PLC program
SPS programm name : customized

Software version check

In switching on routine the picture becomes with ‘ test PLC ‘ term, test character and version the PLC of program displayed.
or
Version check with PLC900X.EXE (see DOKU PLC900X)

Software update

with PLC900X.EXE (see DOKU PLC900X)

1.4 Notes for the line-up

Memory resets (see the also following pages)

With the line-up and with unclear conditions in the FLASH memory the following steps should be executed:

- Passing through of the switching on check with	test parameter:	No Restore
	test NC memory:	No Restore

Reset parameter and NC memory without store back of machine data and NC programs from the FLASH memory.

- Loading of the machine data (channel descriptors, axis definitions, axis data) over I/O traffic.

- Machine and axis configuration check.

- Memory of the characterized machine data into the FLASH memory of the CPU (see q10).

- System switch off and restart.

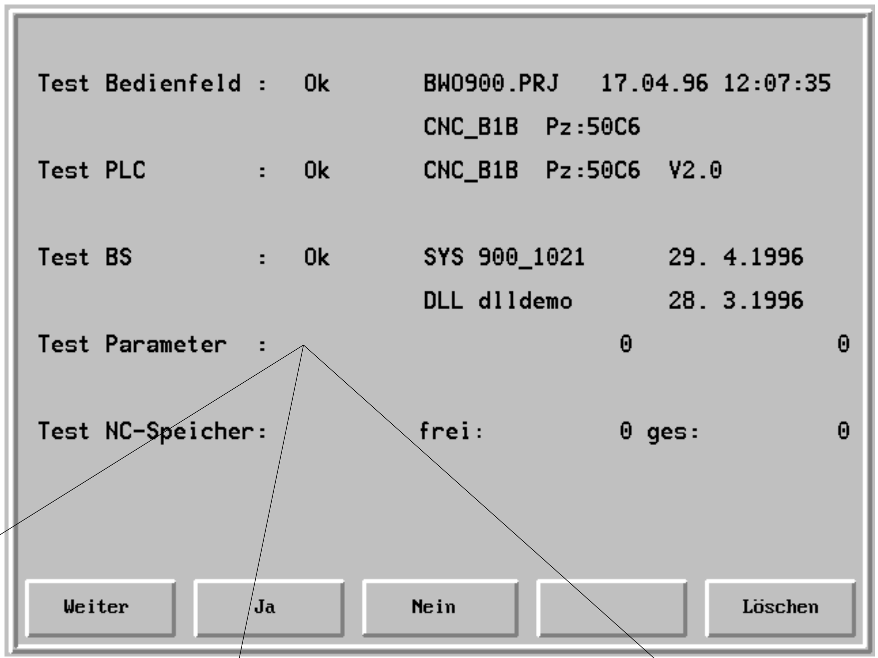
- In the switching on checks in the test parameter ‘ resetting ‘ ‘ input Yes ‘
  and in the test NC memory ‘ resetting ‘ ‘ input Yes ‘.

1.4 Notes for the line-up (continuation)

Test parameter

Possible internal messages with the passing through of the switching on checks with the displays of ' ok ' or ' defect ':

- ' far ones '
- ' resetting ' with Restore
- ' resetting ' without Restore



Display **Ok**

Key of ' far ones ' (F1) press

Effect:
- The parameter values remain received.

- Machine data becomes from THE FLASH memory not loaded.

- Channels and channel descriptor are defined in accordance with parameter value.

Display **Ok / Defect**

Key (reset) press
Reset??

Key ' Yes ' (F2) press

Effect:
- The total parameter memory is cleared.

- Parameter basic adjustment one executes.

- Machine data becomes from the FLASH memory loaded (Restore) (q10).

- Channels and channel descriptor are defined in accordance with parameter value.

display **Ok / Defect**

Key (reset) press
Reset??

Key (picture off) press
NO Restore??
Key ' Yes ' (F2) press

Effect:
- The total parameter memory is cleared.

- Parameter basic adjustment one executes.

- Machine data becomes from the FLASH Memory not loaded (No Restore).

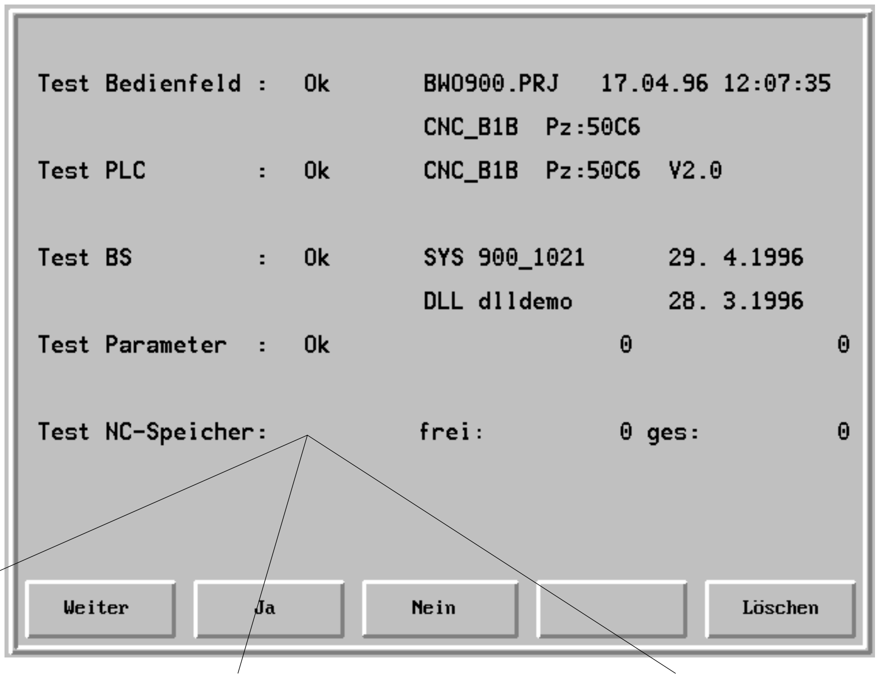
- channels and channel descriptor are defined in accordance with parameter reason in position.

1.4 Notes for the line-up (continuation)

Test NC memory

Possible internal messages with the passing through of the switching on checks with the displays of ‘ ok ‘ or ‘ defect ‘:

- ‘ far ones ‘
- ‘ resetting ‘ with Restore
- ‘ resetting ‘ without Restore



display **Ok**

Key of ‘ far ones ‘ (F1) press

Effect:

- the NC programs remain received.

display **Ok / Defect**

Key (reset) press
Reset??

Key ‘ Yes ‘ (F2) press

Effect:

- the total NC memory one resets.
- NC programs become from the FLASH memory into the NC RAM memory loaded (Restore) (see also q10).

display **Ok / Defect**

Key (reset) press
Reset??

Key (picture off) press
NO Restore??

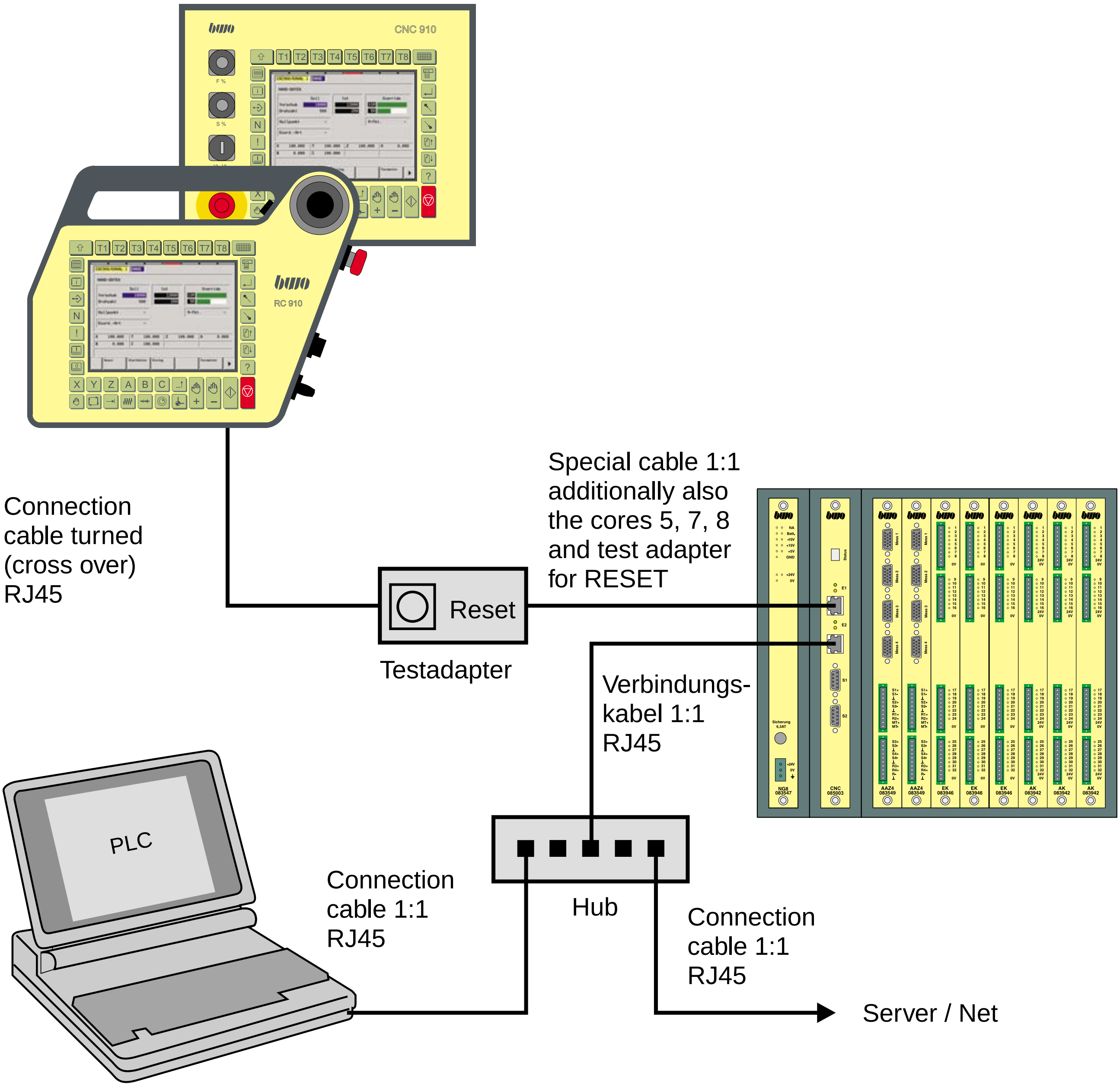
Key ‘ Yes ‘ (F2) press

Effect:

- the total NC memory one resets.
- NC programs become from the FLASH Memory <u>not</u> into the NC RAM memory loaded (NO Restore), (see q10).

NC memory remains empty.

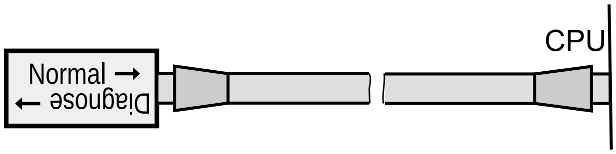
1.4 Notes for the line-up (continuation)



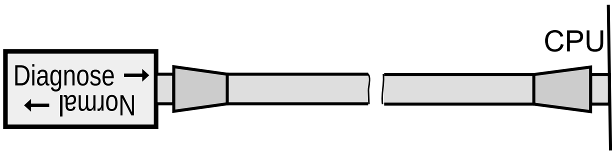
1.4 Notes for the line-up (continuation)

Function of the test adapter

Actual the IP address of the CPU admits:
Test adapters toward ' normal → ' put
to the loading of the operating system.



Actual the IP address of the CPU unknown:
Test adapters toward ' diagnosis → ' put.
Over the RESET key the CPU receives those
default IP address 172.16. 20.180



1.5 CNC functions

Version	**CNC 32Bit ETH**		**CNC 64Bit ETH**	
	Standard	**Export**	**Standard**	**Export**
Item No	085003	085008	085004	086004
Clock frequency	240MHz		240MHz	
Flag memory	60kB		60kB	
NC memory	1,5MB		3MB	
Memory instructions	16ki		64ki	
Parameter	30 000		60 000	
Channels	4		8	
Axis	16		32	
Linear nterpolation in axis	16	4	32	4
Circular interpolation in axis	3 +13	2+1	3 + 29	2+1
Screw interpolation in axis	2 +14	-	2 + 30	-
Spline interpolation	•	-	•	-
Polynom interpolation	•	-	•	-

• Function is only possible by standard design

- Function is not possible by export version

1.5 CNC functions (continuation)

CPU dependent functions	CNCE (32Bit)	CNC (32Bit)	CNC (64Bit)
Clock frequency	133 MHz	133 MHz	133 MHz
Flag memory	60 kB	60 kB	60 kB
NC memory	1.5 MB	1.5 MB	3.0 MB
Parameter	30,000	30,000	60,000
Block cycle time	< 6 ms	< 6 ms	< 1.5 ms
Mounting gel clock	< 3 ms	< 3 ms	250 µs
Channels	2	4	8
Axes	8	16	32
Linear interpolation in	4	16	32
Circular interpolation in	2 + 1	3 + 13	3 + 29
Screw interpolation in	-	2 + 14	2 + 30
Spline interpolation	-	•	•
Polynomial interpolation	-	•	•

1.5 CNC functions (continuation)

- Several spindles
- Tangential axis
- Axes couple, reflect and exchange
- Restarting after abort
- Feed, corners, circle and outline dynamics
- Electronic gears
- Handwheel
- Digital and analog drives
- Polar coordinates system
- Polar transformation
- Robot transformation ** / tool coordinates / workpiece coordinates
- Axes and graphic simulation
- Coordinates turn, reflect and shift
- Measuring cycles and processing cycles
- Interpolation plane selection
- Tool radius path correction
- Automatic selection of linear and circular interpolation
- Zero points / zero point shift
- Outline path short programming
- Parameter calculation
- Diagnostic functions

** with max. 4 axes by export version

1.5 CNC functions (continuation)

Diagnostic function status

The 7 section display ‘ status ‘ shows the hardware status of the CPU on.

Display	Function
Segments circle	everything in order, everything runs ‘ approximately ‘.
out	CPU defectively, no voltage, display defectively.
0	CPU in the monitor operation.
8.	hardware RESET.
1 - 9	hardware test after that boats. if status remains 1 - 9, if the hardware test was not ok - > CPU defectively.
b	writing in the flash, do not switch off.
E flashes	error while the loading of the operating system.
E1	fatal error, please at BWO turn.
E2	fatal error, please at BWO turn.
F	operating system is charged from the flash.
F0	hardware error. Module or network failed.
F1	back-up battery defectively.
F2	voltage ±15V defectively.
F3	back-up battery and voltage ±15V defectively.
F4	CPU fan defectively.

2. Operating

2. Operating

2.1 Operating control functions

2.1.1 Operating panels CNC 910 / CNC 920 / RC 910

The operating panels are conceived for the application at CNC controllers. The fully diagramable color LC display and foil keyses with mechanical pressure point facilitate the operation.

Standard design CNC910

Dimensions 277mm x 227mm (width x height)

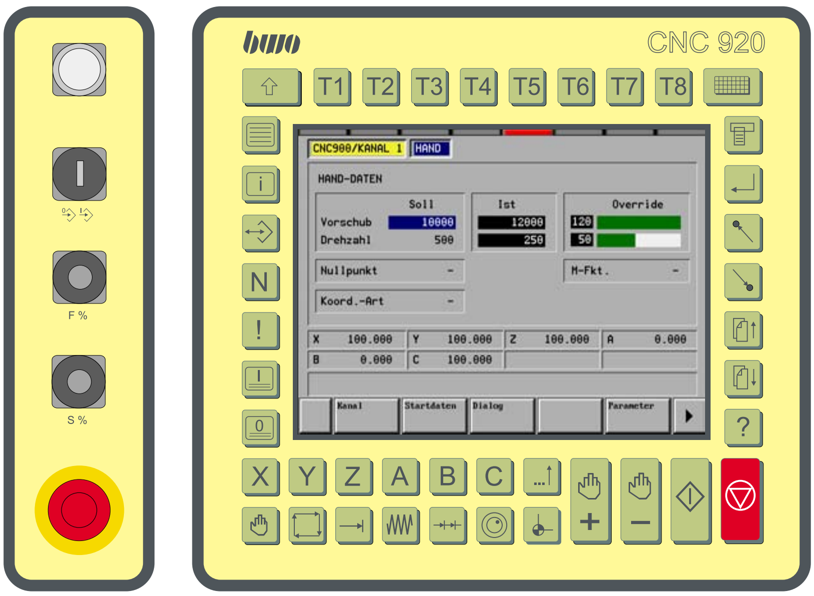
Machine operating panel integrates

Standard design CNC920

Dimensions 328mm x 310mm
(width x height)

Machine operating panel
separately

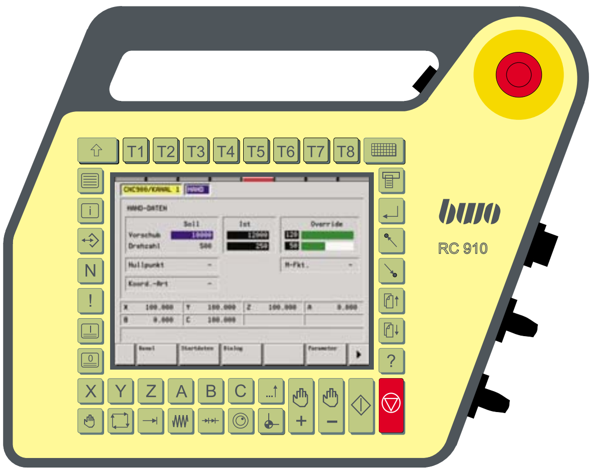
Dimensions 80mm x 310mm
(width x height)



2.1.1 Operating panels CNC 910 / CNC 920 / RC 910 (continued)

Operating panels RC 910
The operating panels have an ergonomic and manualfair form with a fully diagramable color LC display and foil keyses with mechanical pressure point, which facilitate the operation.

Dimensions 310mm x 240mm (width x height)



Standard version RC910
are suitable for the application as mobile operating panels at CNC controllers or as additional manual operating console for a firmly built in operating panel.

Execution with handwheel RC910
are suitable for the application as mobile operating panels at CNC controllers or as additional manual operating console for a firmly built in operating panel.

Execution with joystick RC910
is appropriate for the application at robotic controls. In this execution robot and manual axes can be moved three-dimensional in each case.

2.1.1 Operating panels CNC 910 / CNC 920 / RC 910 (continued)

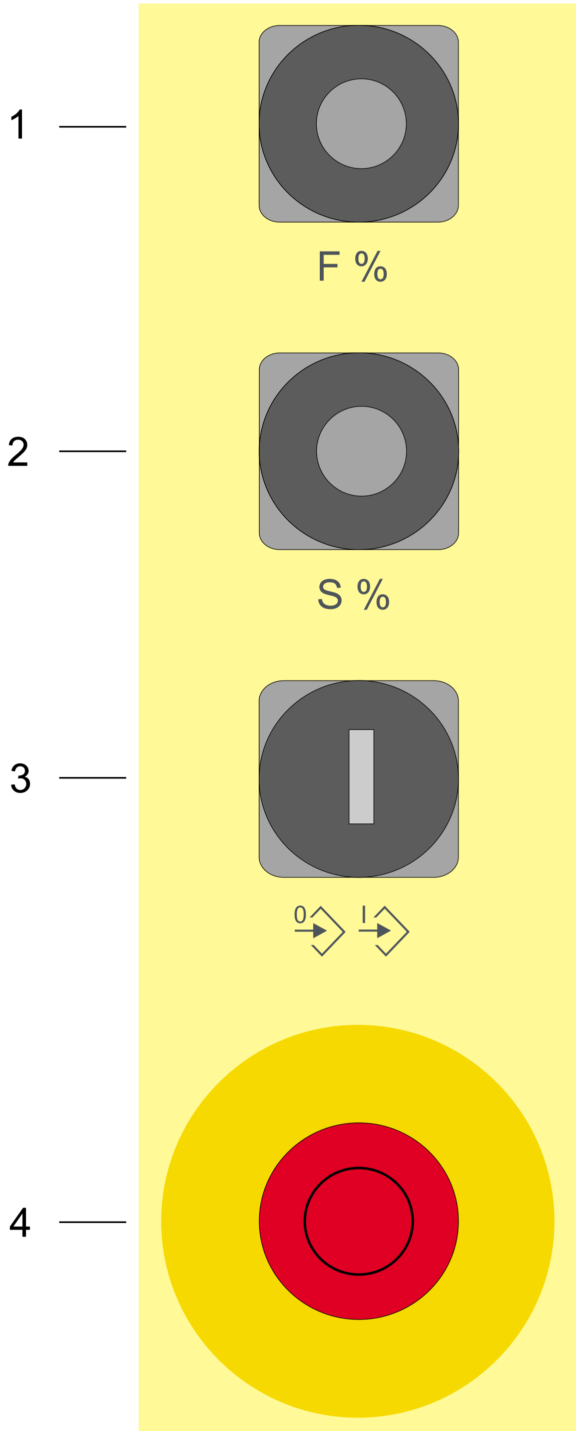
Machine operating panel CNC 910

1 Feed override

2 Spindle override

3 Code switch
Program block-free

4 Emergency-stop key



Note:

Code switch 1 opener (21/22) freely

Emergency stop 2 openers (11/12 and 21/22) freely

2.1.1 Operating panels CNC 910 / CNC 920 / RC 910 (continued)

Separate machine operating panel CNC920

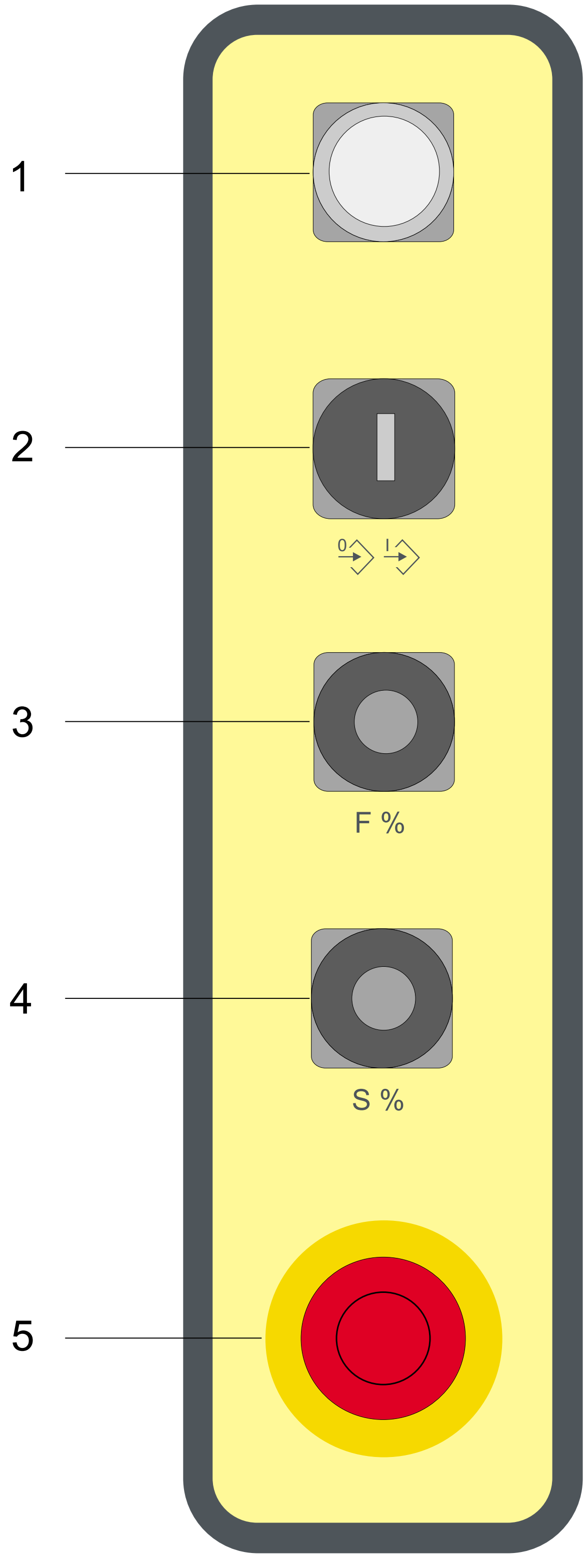
1 Illuminated push button

2 Code switch
Program block-free

3 Feed override

4 Spindle override

5 Emergency-stop key



Note:

Code switch 1 opener (21/22) freely

Emergency stop 2 openers (11/12 and 21/22) freely

2.1.1 Operating panels CNC 910 / CNC 920 / RC 910 (continued)

Machine operating panel RC910

1 Handwheel or joystick,
during standard version emergency stop

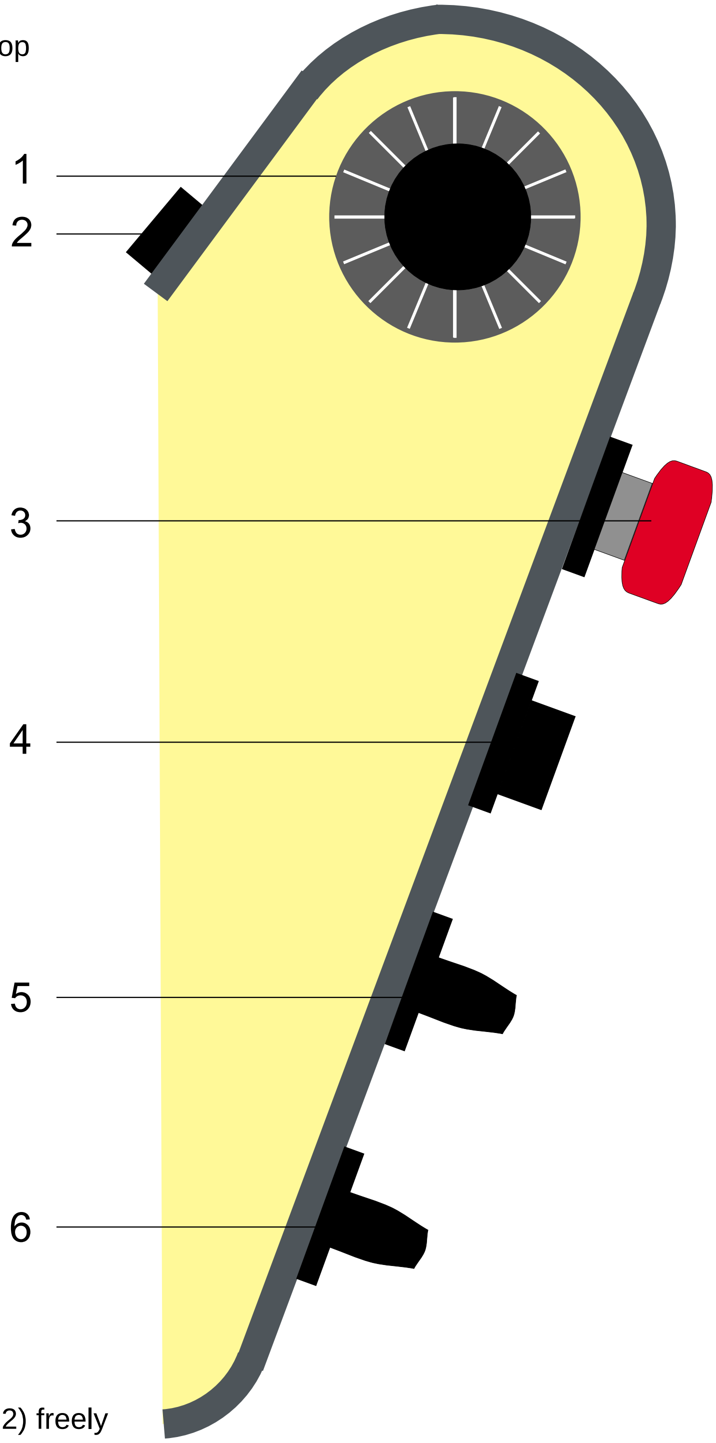
2 Release button

3 Emergency stop with
handwheel and joystick

4 Code switch
Program block-free

5 Feed override

6 Spindle override
(not with joystick)



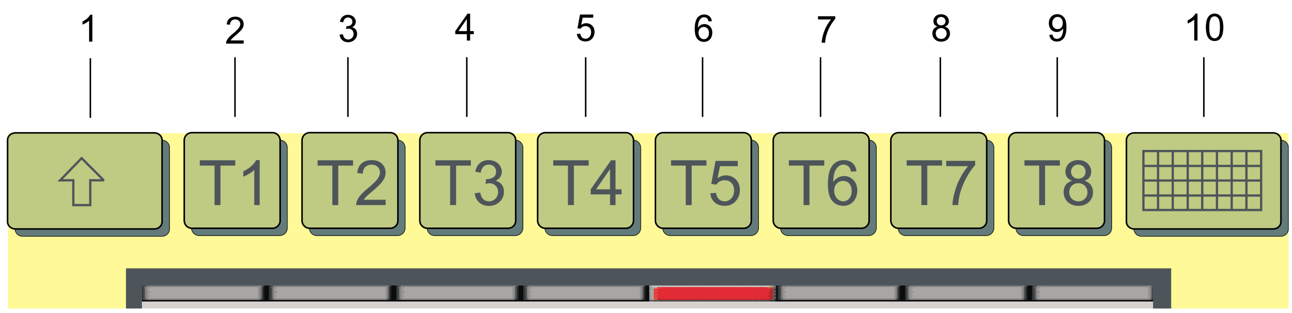
Note:

Code switch 1 opener (21/22) freely

Emergency stop 2 openers (11/12 and 21/22) freely

2.1.1 Operating panels CNC 910 / CNC 920 / RC 910 (continued)

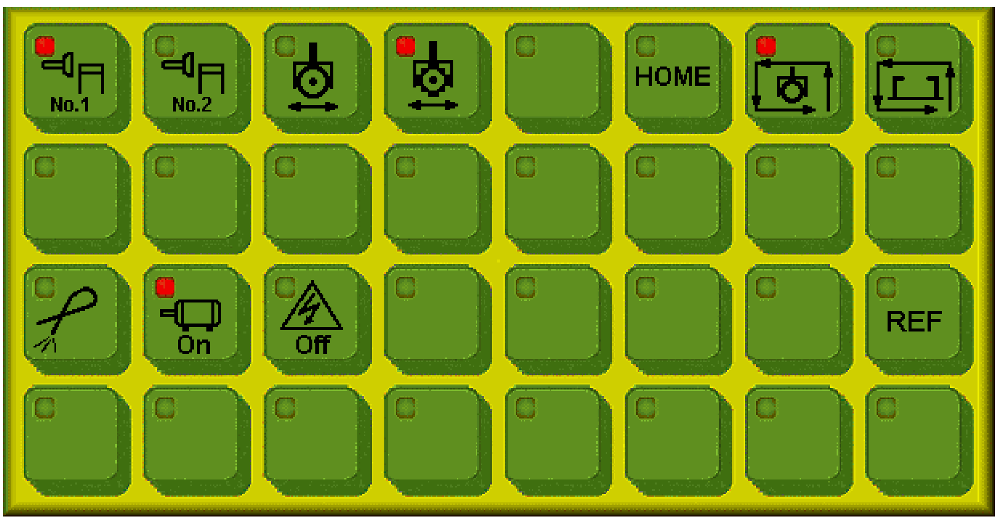
Control keys for PLC functions



1 Shift
In connection with further keys special functions can be released, e.g. can with editor on / off the touch screen activated or deactivated become (to the cleaning of the screen).

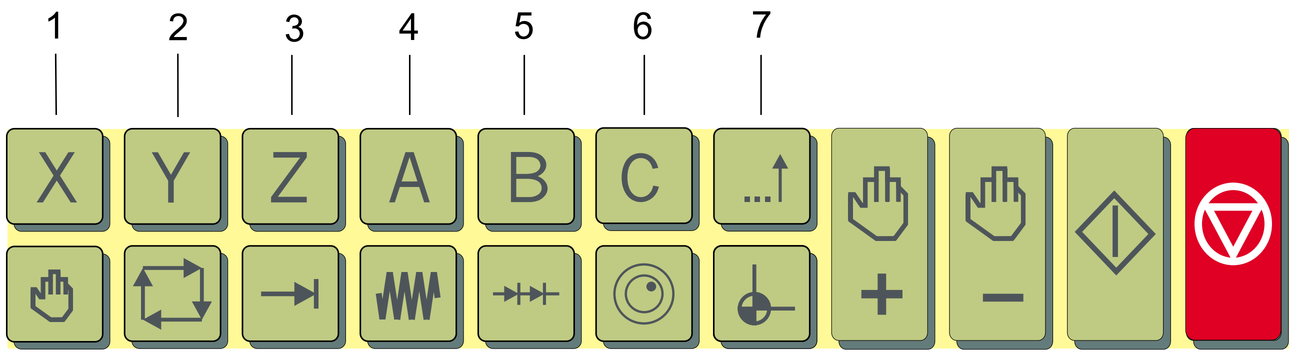
2 to 9 8 PLC keys with LED on the display can by the user as desired be marked (with label strip e.g. T1 to T8).

10 additional 32 PLC keys with LED on the Touch screen The PLC additive keys can be designated by the user as desired, e.g.:



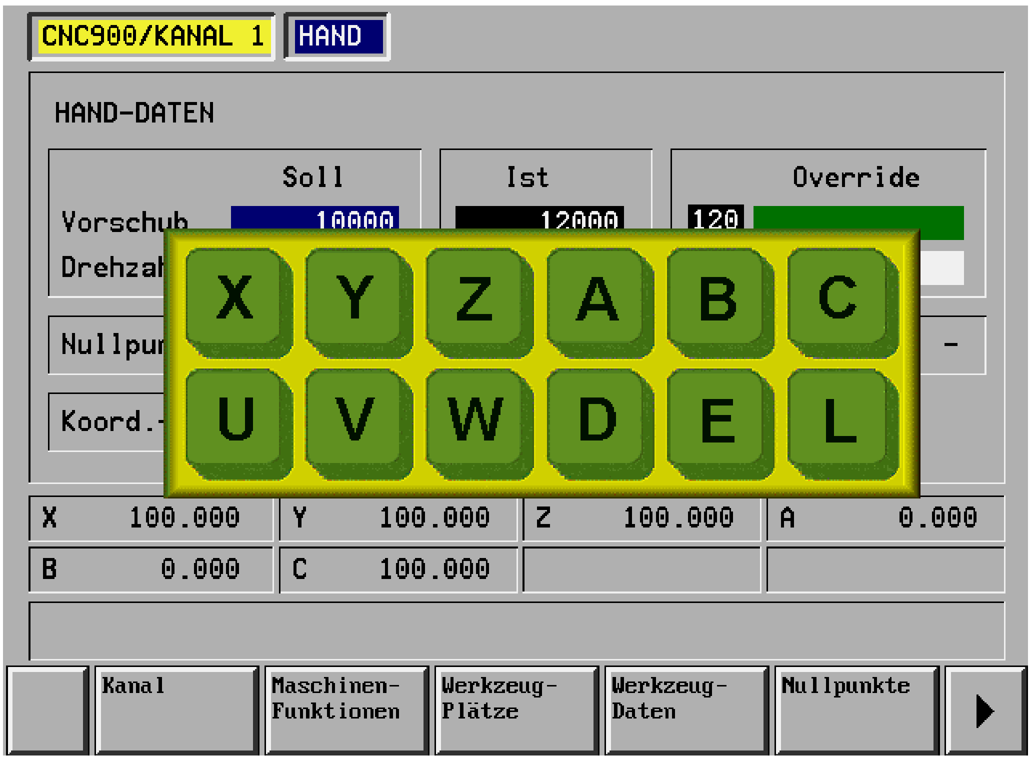
2.1.1 Operating panels CNC 910 / CNC 920 / RC 910 (continued)

Control keys for axis selection



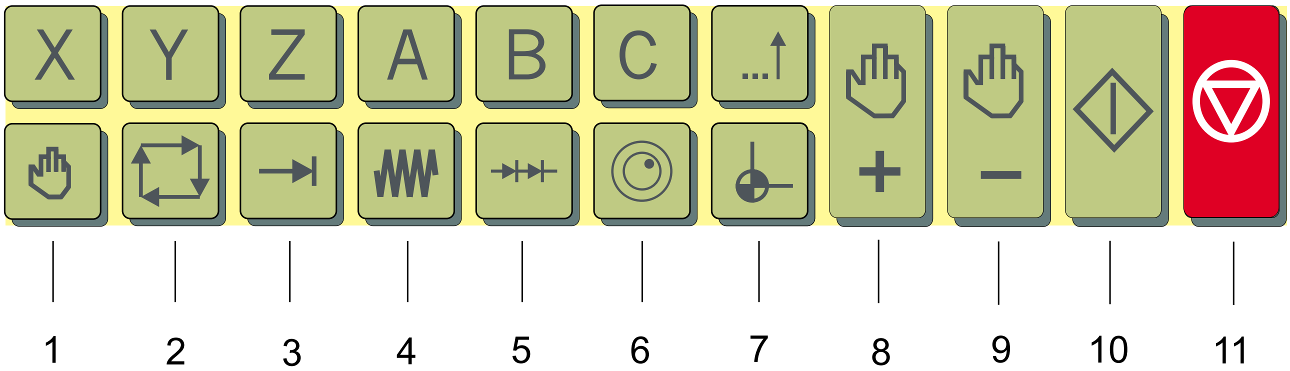
1 bis 6 6 axis keys
Can by the user as desired be marked
(with label strip e.g. X, Y, Z, A, B, C).

7 12 axis keys on the Touch screen,
Can by the user as desired be marked
e.g.:



2.1.1 Operating panels CNC 910 / CNC 920 / RC 910 (continued)

Control keys for modes of operation

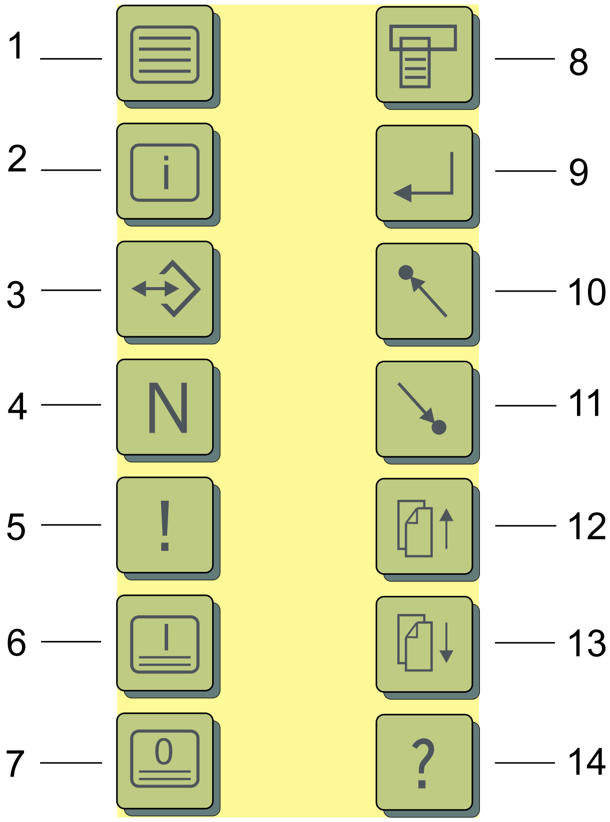


1	Manual mode	
2	Switching in automatic mode:	With the 1. Pressures becomes the sequential block adjusted. Afterwards becomes when each pressing key between sequential block and single block back and forth switched.
3	Switching in positioning mode	
4	Continuous traversed in the manual mode	
5	Step by step traversed in the manual mode, incrementations freely definably	
6	With handwheel traversed	
7	Automatic home position and travel of point of reference	
8 und 9	Push buttons for traversing the selected axis with manual mode	
10 und 11	start and stop The automatic program sequence is started or stopped - in the positioning mode: The NC block in the indication area is processed. - in the automatic mode: The selected NC program is worked on.	

2.1.1 Operating panels CNC 910 / CNC 920 / RC 910 (continued)

Control keys for program input and data communication

1	Key function freely shapable, e.g. graphic simulator.
2	Key function freely shapable, e.g. cycle or info. pictures.
3	I/O circulation Selection of the I/O menu to the data communication
4	Block editor
5	Diagnosis Information about conditions, interfaces, parameters.
6	Editor on
7	Editor off
8	Menu selection
9	Enter
10 und 11	Beginning / end
12 und 13	Picture up / down
14	Additional information Information about lining up NC program.



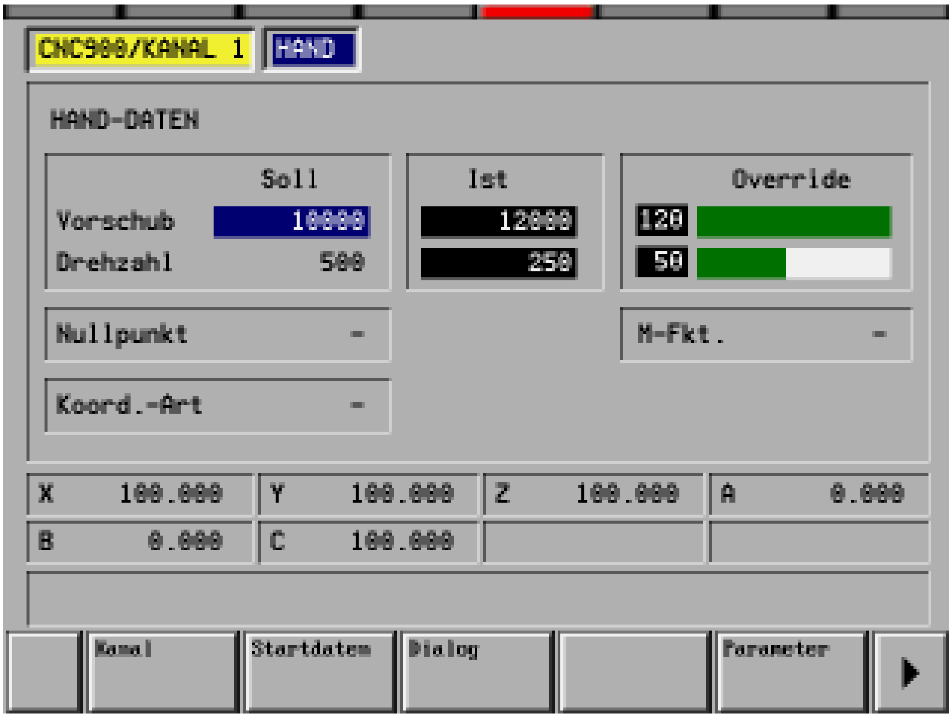
2.1.1 Operating panels CNC 910 / CNC 920 / RC 910 (continued)

Touch screen keys for manual mode and program input

Opening display
Selection menu

Menu screen 1

Selection keys on the touch screen



Menu screen 2

Switching between the menu screens with key  (roll function)

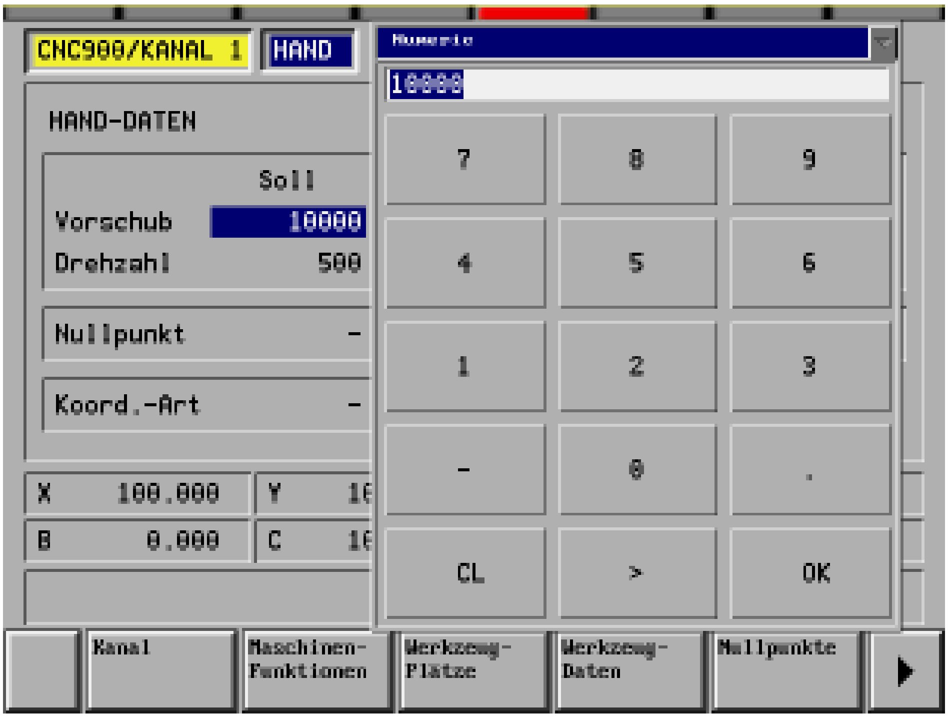
2.1.1 Operating panels CNC 910 / CNC 920 / RC 910 (continued)

Touch screen keys for input with numeric keyboard

The cursor press and shift on a wished input field .

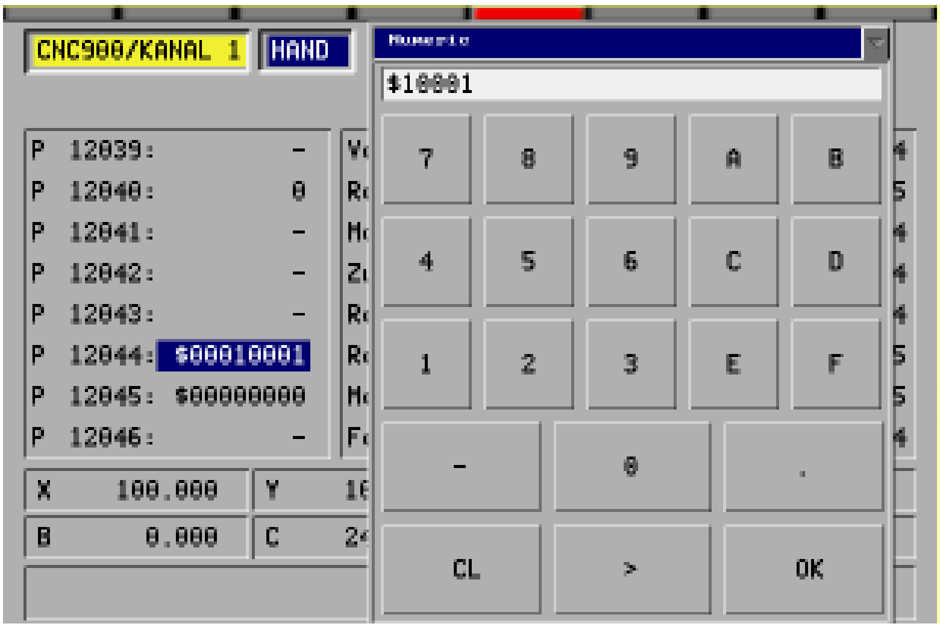
The numeric keyboard will during contact an input field faded in.

Input break off:
With the finger press on not with input fields occupied place on that display.



Requires the input field hexadecimal input, will the hexa-decimal keyboard faded in.

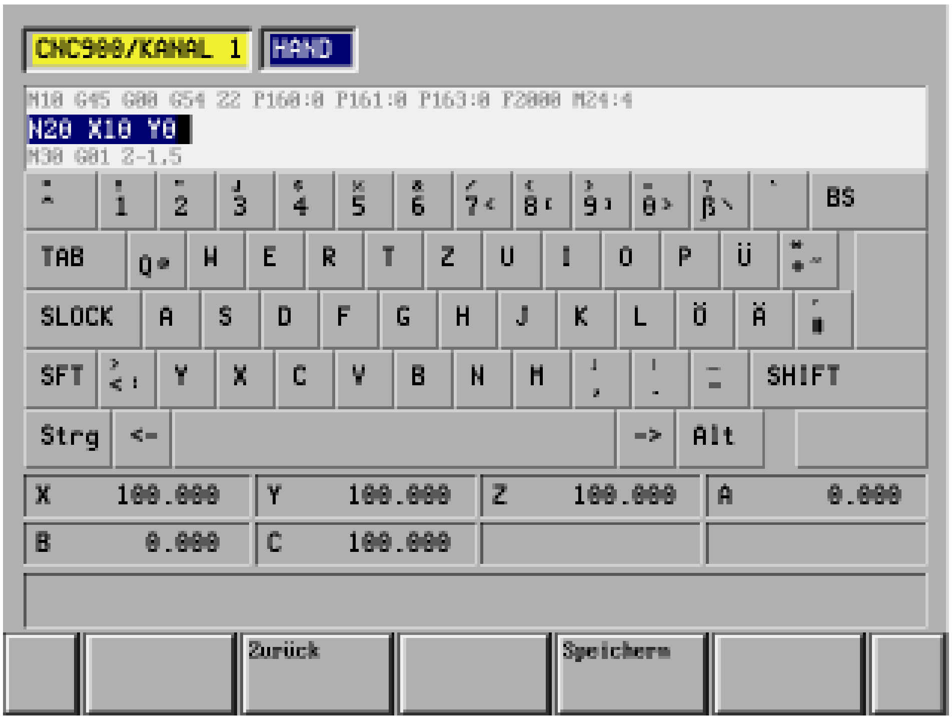
With 2 x pressures on ' - ' indication becomes between decimally and hexadecimal keyboard around switched.



2.1.1 Operating panels CNC 910 / CNC 920 / RC 910 (continued)

Touch screen keys for input with ASCII keyboard

With the program input will a ASCII keyboard faded in.



Special functions with WINTERM

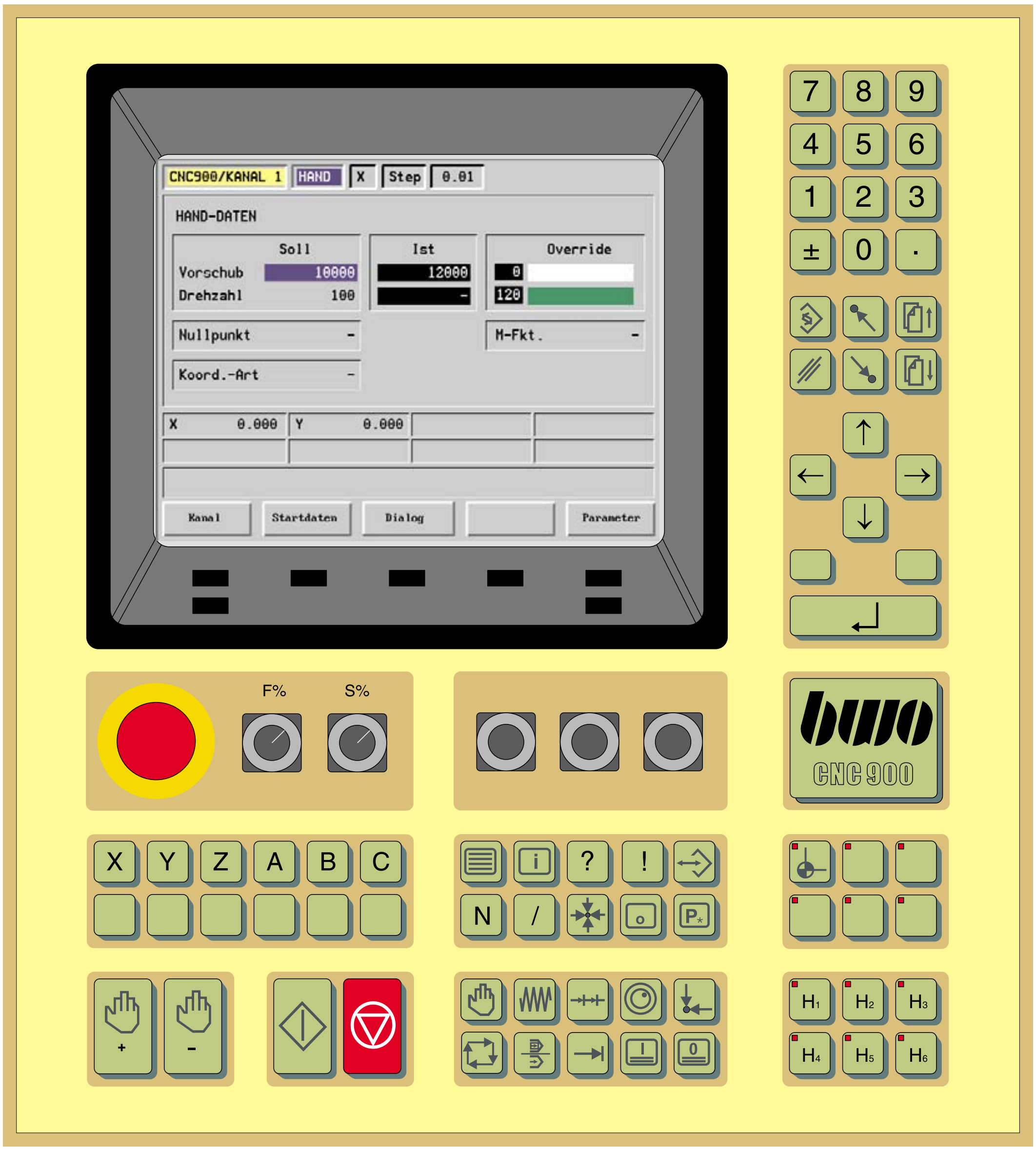
The ASCII keyboard can be switched by hand: on / off 

The contrast of the announcement of the ASCII keyboard can be adjusted.
It can be struggled so far that the keyboard appears only as background picture.

The key functions remain.
However one can read now also the text, which is under the keyboard.

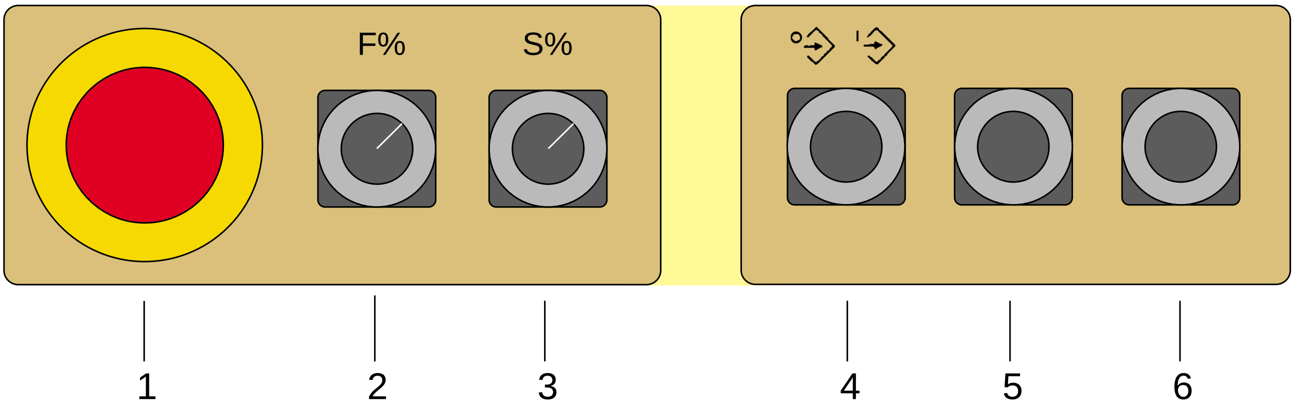
Contrast of the announcement: more brightly  more darkly 

2.1.2 Operating panels CNC 900 / CNC 900C



2.1.2 Operating panels CNC 900 / CNC 900C (continued)

Operating controls for control functions CNC900

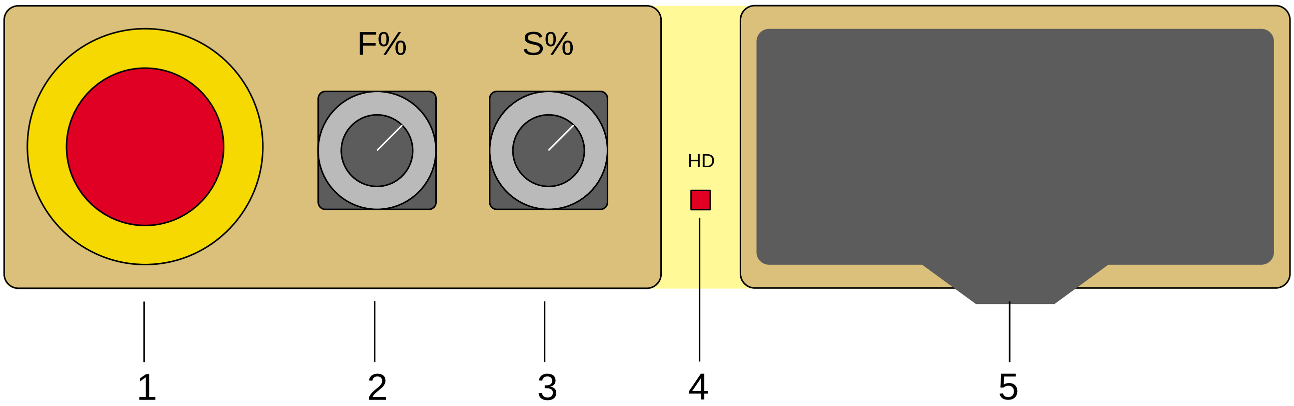


1 Emergency-stop key
During the pressing of the key all movements of the machine are interrupted (analysis in interface control).

2 to 6 User or machine-specific allocation with CNC900C.
for example feed and spindle Override or programming block / freely.

2.1.2 Operating panels CNC 900 / CNC 900C (continued)

Operating controls for control functions CNC900C



1 Emergency-stop key
During the pressing of the key all movements of the machine are interrupted (analysis in interface control).

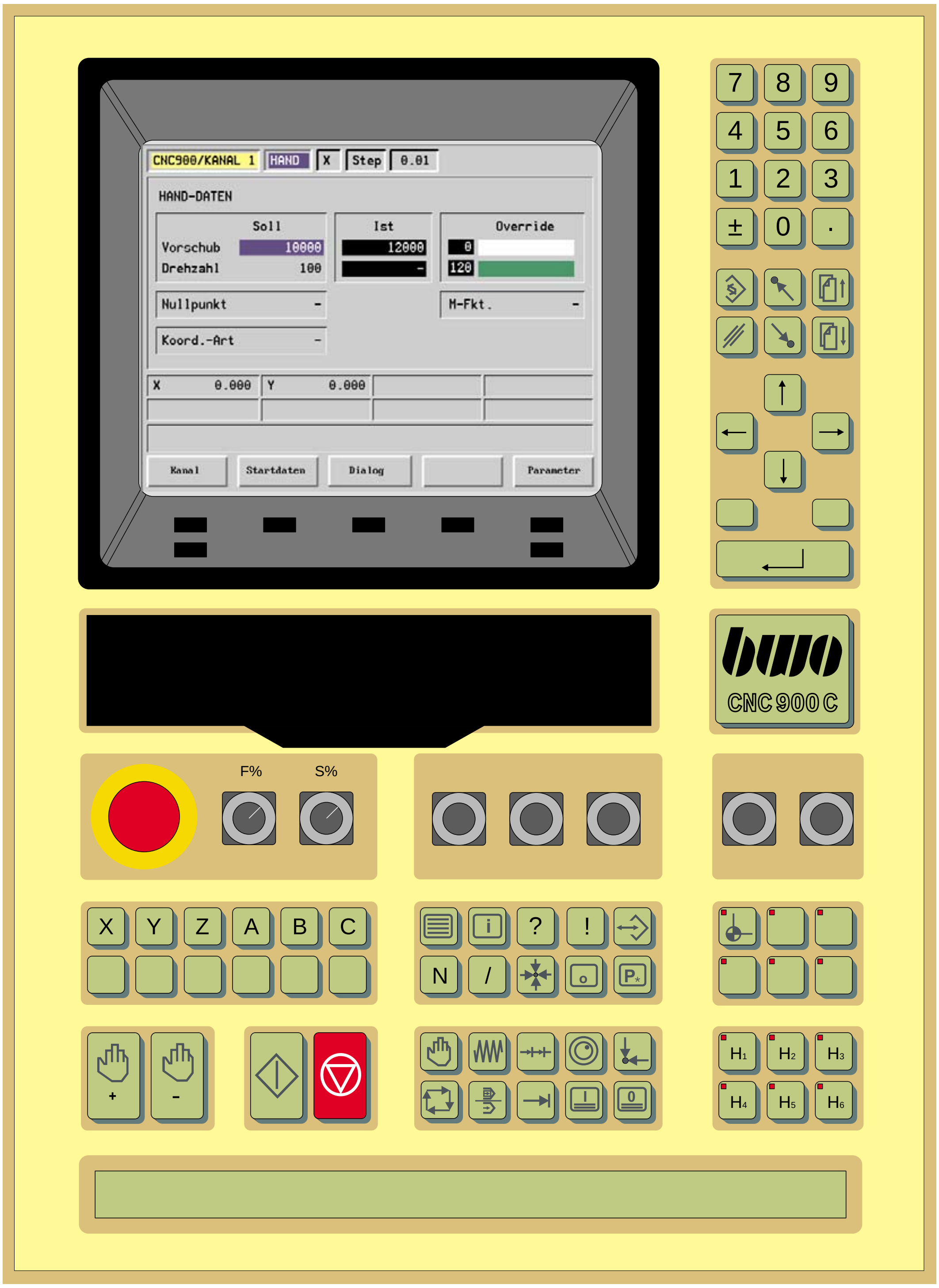
2 to 3 User or machine-specific allocation with CNC900C.
for example feed and spindle Override or programming block / freely.

4 Light emitting diode hard disk drive

5 Dikettenlaufwerk 3.5 “

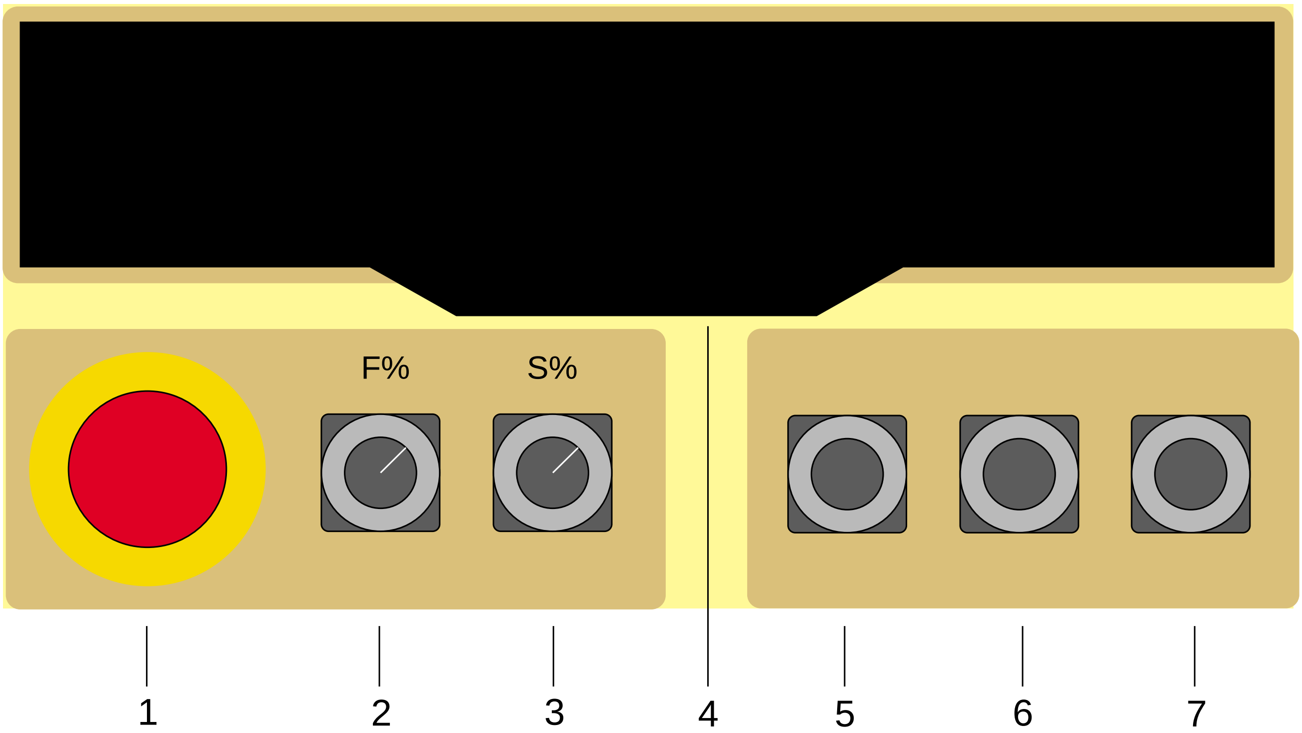
2.1.2 Operating panels CNC 900 / CNC 900C (continued)

Operating controls for control functions CNC900C with Windows operating system



2.1.2 Operating panels CNC 900 / CNC 900C (continued)

Operating controls for control functions CNC900C with Windows operating system



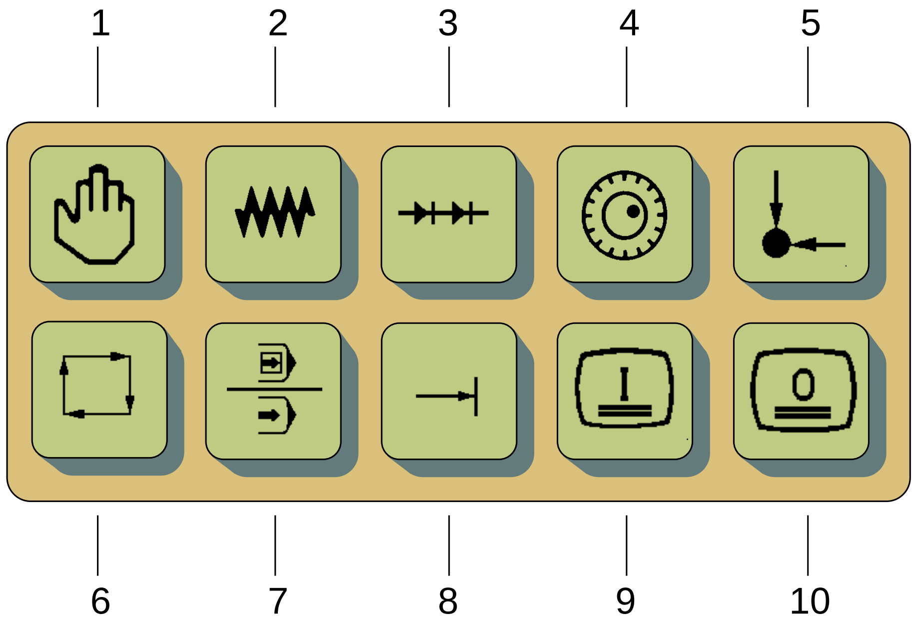
1 Emergency-stop key
During the pressing of the key all movements of the machine are interrupted (analysis in interface control).

2 to 3 User specific or machine-specific allocation with CNC900C.
5 to 7 For example feed and spindle Override or programming block / freely.

4 Floppy disk drive 3.5 “ and CD ROM drive

2.1.2 Operating panels CNC 900 / CNC 900C (continued)

Operating controls for operating modes



1 Manual mode. The selected axes can be moved with the manual keys.

2 Continuous traverse during manual mode.

3 Step by step traverse during manual mode. steps can be defined arbitrarily.

4 Traverse during manual mode with handwheel.

5 Approach home position in manual mode.

6 Switch on automatic mode (sequential block).

7 Changeover single/sequential block during automatic mode.

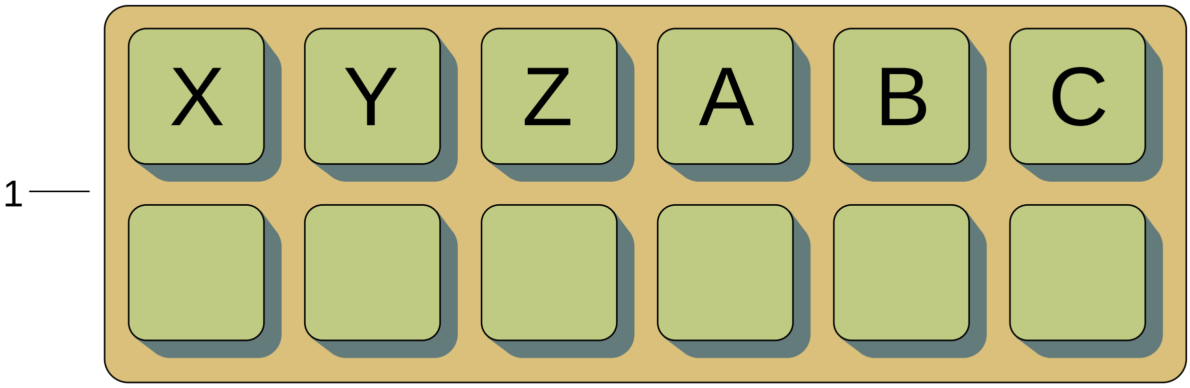
8 Switch on positioning.

9 Editor on

10 Editor off

2.1.2 Operating panels CNC 900 / CNC 900C (continued)

Operating controls for axes selection

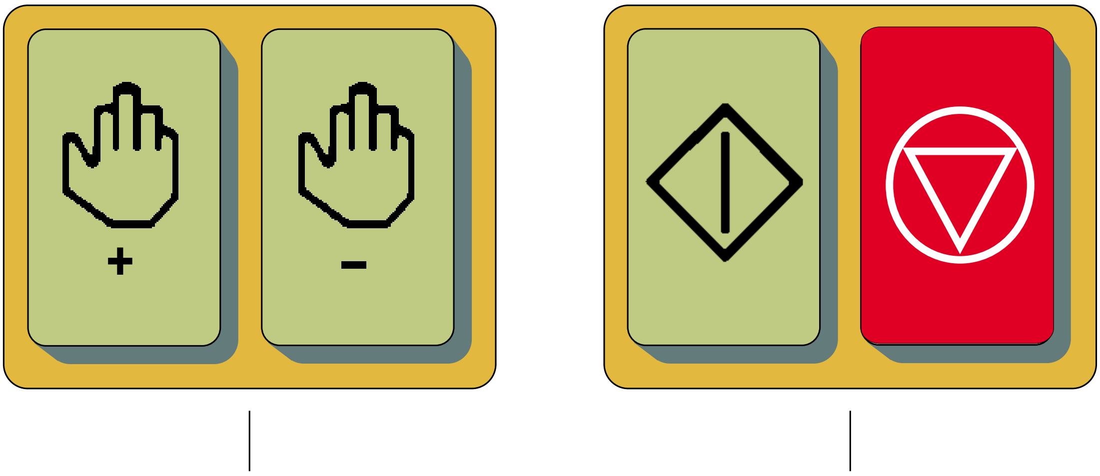


1 Axes keys

Selection: during manual mode for traversing the axes
during automatic mode for displaying actual values when the NC program is input.

2.1.2 Operating panels CNC 900 / CNC 900C (continued)

Operating controls for manual and automatic functions



1 2

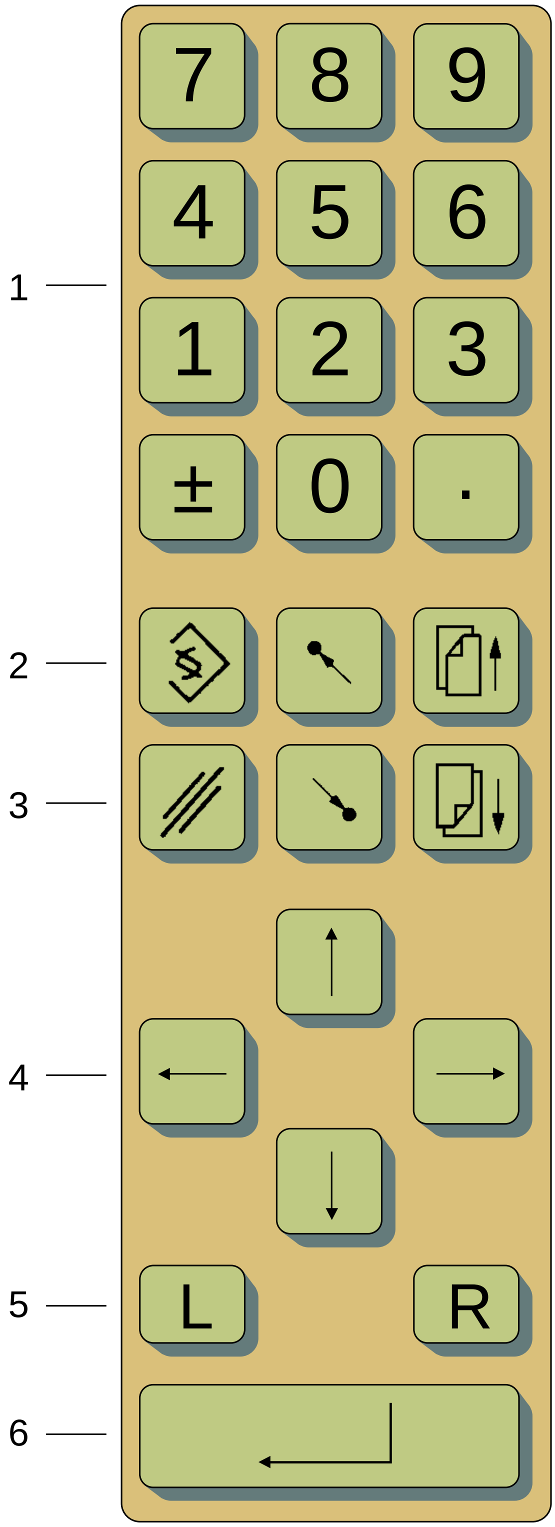
1 Manual keys
During manual mode for traversing the selected axis

2 Start and Stop
The automatic program sequence is started or shut down

- in positioning mode: the displayed NC block is worked.
- in automatic mode: the selected NC program is worked.

2.1.2 Operating panels CNC 900 / CNC 900C (continued)

Operating controls for program inputting



1 Numerical keyboard 0 to 9.
Prefix +/- and decimal point

2 Keys (seen from left side)
'Change' 'Pos 1' 'page up'

3 Keys (seen from left side)
'Clear' 'End' 'page down'

4 Cursor keys

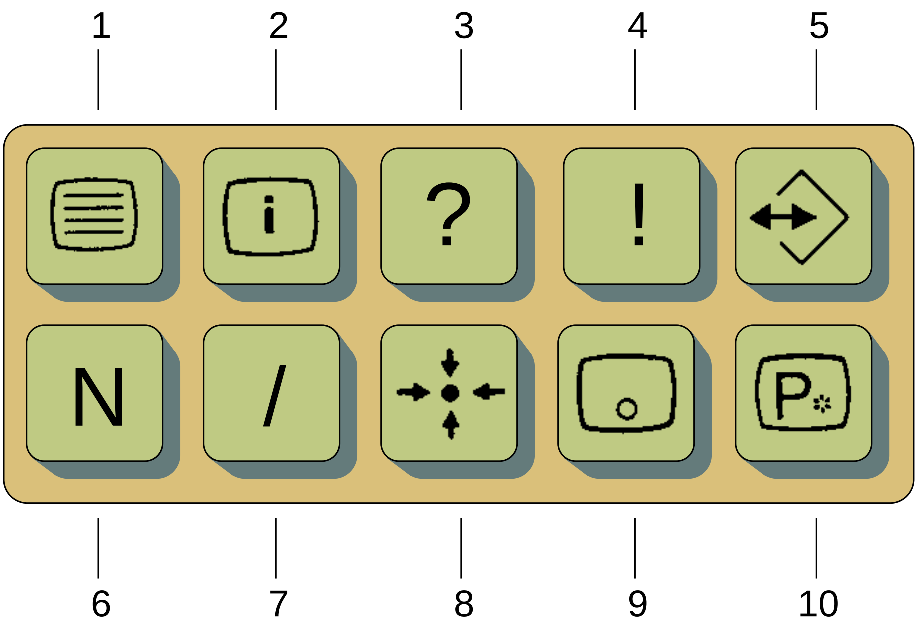
5 Enter
Selected enter mode is activated and
program data stored.

Note to adjusting the display brightness

With F6 (within the display frame down on the right) and
' picture up ' display becomes continuously brighter.
' picture down ' display becomes continuously darker.

2.1.2 Operating panels CNC 900 / CNC 900C (continued)

Operating controls for display and data transfer

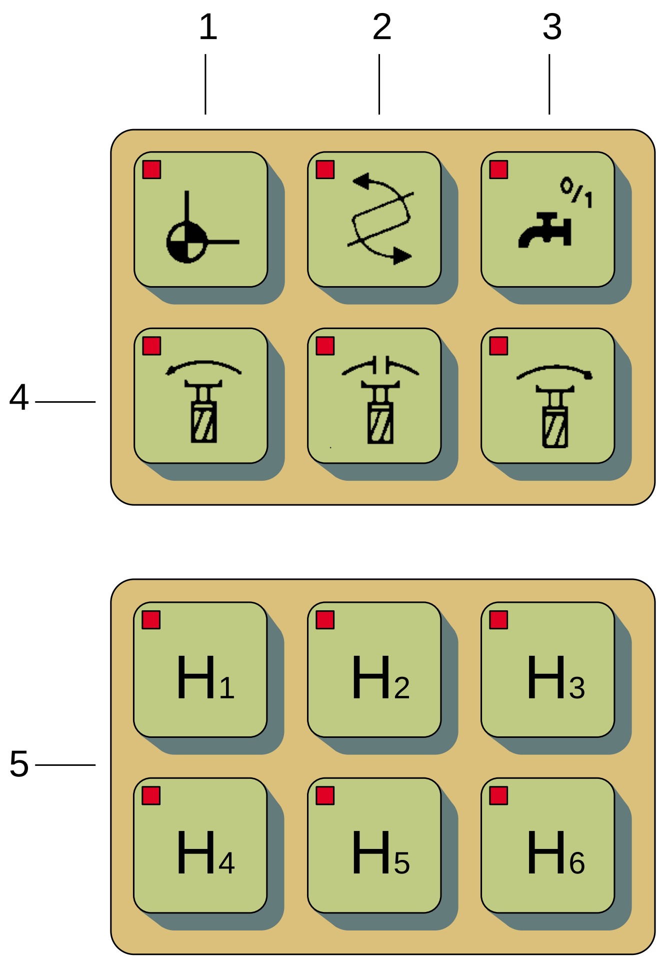


1	Graphics <-> Normal	Changeover between graphics and normal.
2	Cycle patterns	Display menu of all cycles provided.
3	Supplementary information	Informations on current NC program.
4	Diagnosis	Informations on status. interfaces. parameters.
5	I/O	Selection of I/O menu for data transfer.
6	Block input	Cursor jumps to block number.
7	Skip block	during inputting of block and Start menu.
8	Target-value = actual-value	displayed actual-values are accepted as the programmed values.
9	Clear screen	When inputting the block. the old block is cleared.
10	Parameter calculation	Parameter calculation can be carried out when the block is being input.

2.1.2 Operating panels CNC 900 / CNC 900C (continued)

Operating controls for machine functions

The machine functions are machine-specific and are determined in the interface.
A keyset is already provided on the operating panel for these functions so that extra keyboards are not necessary. The occupation is only an example.



1 Automatic referencing

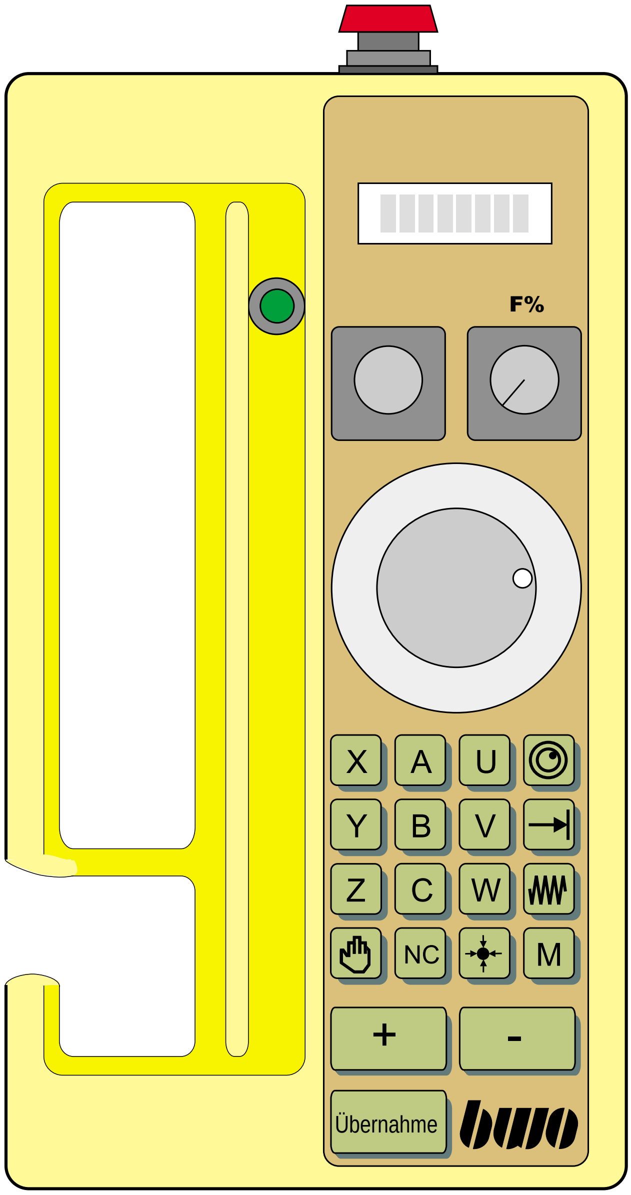
2 Tool change

3 Coolant on/off

4 Spindle left-handed rotation / stop / right-handed rotation

5 Help functions

2.1.3 Teachpanel



Note!

Off safety reasons may the manual operating console (TP) only with switched off machine to the main operating console (CNC900) to be connected or from it separated.

2.1.3 Teachpanel (continued)

Operating controls for control functions

1 Emergency-stop switch

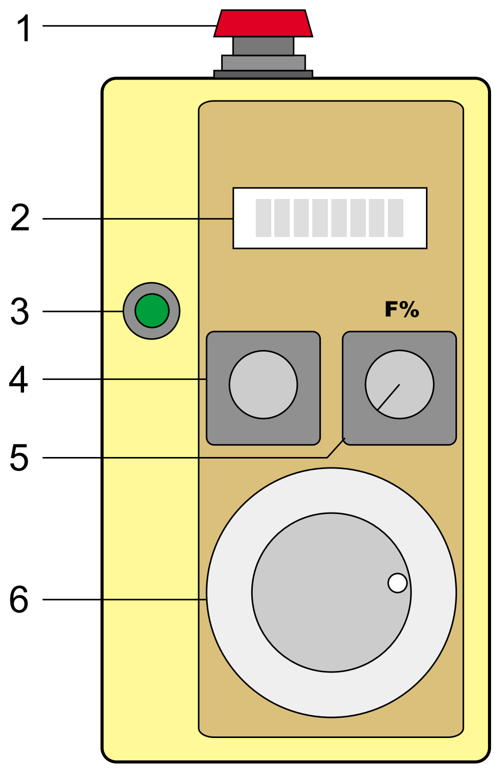
2 Display field

3 Release button

4 Key-operated switch

5 Feed override

6 Handwheel



2.1.3 Teachpanel (continued)

Operating controls for axes selection and for operating modes

1 CNC keys X, Y, Z, A, B, C, U, V, W

2 Manual operating mode

3 NC editor

4 + / -

5 Enter

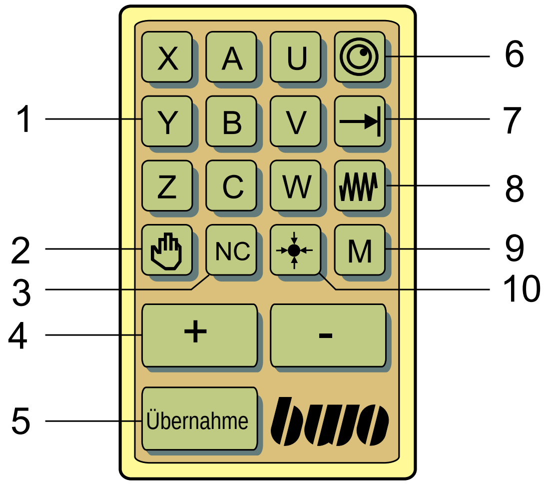
6 Handwheel operating mode

7 Step by step operating mode

8 Continuous operating mode

9 PLC keys, G functions

10 Target-value = actual-value



2.1.3 Teachpanel (continued)

Functions

Emergency-stop switch

With the application of this switch, all machine movements are stopped immediately (evaluation in the interface control).

Key-operated switch

The teachpanel is switched on and off with the key-operated switch.
The machine operating panel is also activated/deactivated simultaneously by this action.

Attention:
The teachpanel has to be switched on before you enter the machine area of risk and before switching off the teachpanel, you have to leave this area.

Release button

For safety reasons, the release button circuit is required to be connected in such way that the axes can only be moved as long as the button is depressed.
When it is released, all axis movements should come to an immediate standstill.

Feed override

The programmed feedrate (manual data in P8755) can be steplessly adjusted with the feedrate potentiometer.

Display

```
TP active
```

The display field consists of a single line of eight characters.

When switching on the teachpanel, the display ‘TP active’ appears.

2.1.3 Teachpanel (continued)

Axis selection X - W

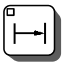
When pressing an axis key, the corresponding LED is switched on. The actual value appears in the display. Non existing axes can not be selected. Only one axis can be selected at a time.

When switching on the teachpanel, no axis is selected.

Continuous operating mode 

The operating mode ‘continuous’ is selected with this key. When pressing the key, the LED is switched on.

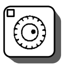
The selected axes can be moved with the key + or -.

Step by step operating mode 

The operating mode ‘step’ is selected with this key. When pressing the key, the LED is switched on. The actual step width appears in the display (e.g. 0,01; 0,1; 1; 10).
In the list of parameter P8763 to P8769 you can find the selectable step width.

The step width can be switched over by pressing the key again.

The selected axes can be moved with the key + or -.

Handwheel operating mode 

When pressing the key handwheel, the corresponding LED is switched on. The actual handwheel factor HF (e.g. 1, 10, 100) appears in the display.
In the list of parameter P11407 to P11409 you can find the selectable handwheel factor.

The handwheel speed factor can be switched over by pressing the key again.

2.1.3 Teachpanel (continued)

Manual operating mode 

This key starts the manual operating mode. The LED is switched on when pressing the key.
When the key is pressed, the actual feed appears in the display.
Further operating modes are not acceptable on the teachpanel.

Menue M

This key activates a menue for the simulation of PLC keys 1-12.

When pressing the key again, the G-functions G1, G2, G3, G123 appear which can be selected with the keys + / - and they can be input in the block editor with enter.

Target = Actual 

The actual position of the axes set in parameter P8747 are overtaken to the block editor with target = actual.
The block is stored with the key enter.

Keys + / -

The selected axis moves in positive / negative direction with the selectedmode (continuous / step / handwheel).

With these keys

- you can scroll the block number while the NC editor (display ‘n’ + block number) is active,

- you can move the axes in manual operating mode or active block editor
  (display ‘N’ + block number)

- you can scroll PLC keys and G-functions when the menue is activated (key M).

2.1.3 Teachpanel (continued)

NC editor NC

The NC editor is called up with the key NC.
The program which is activated by pressing NC is specified in the parameter P11145 and the block number is specified in the parameter P11146.

In the display of the main operating panel, the actual program with ‘n’+block numbers appears.
A block can be selected in the program with the keys + / -.

The block editor with the selected block is called up with the key enter.
In the display of the main operating panel, the actual block with ‘N’+block number appears now. The block can be edited now. E.g. the actual positions can be overtaken at the teachpanel with target = actual. Further modifications of the NC block are possible parallel to this at the main operating panel.

The NC block is stored by pressing the key enter again.
At modifications within a program, the display ‘n’+block number appears again in the main operating panel. If a block was edited at the program end, ‘N’+block number appears in the display of the main operating panel. A new empty block is inserted with a new block number, that consists of the last block number +10.

Enter

Function in the NC program:	The selected NC block is stored in the block editor with the key ‘enter’.
Menue function:	The selected PLC key is transmitted to the PLC with the key enter or the G-function standing in the display is written into the block.

2.2 Power on tests

The display field is activated after swichting on the control. The control starts a self-test. The position of the key-operated switch 1 (P11000) determines the course.

2.2.1 Automatic power on tests

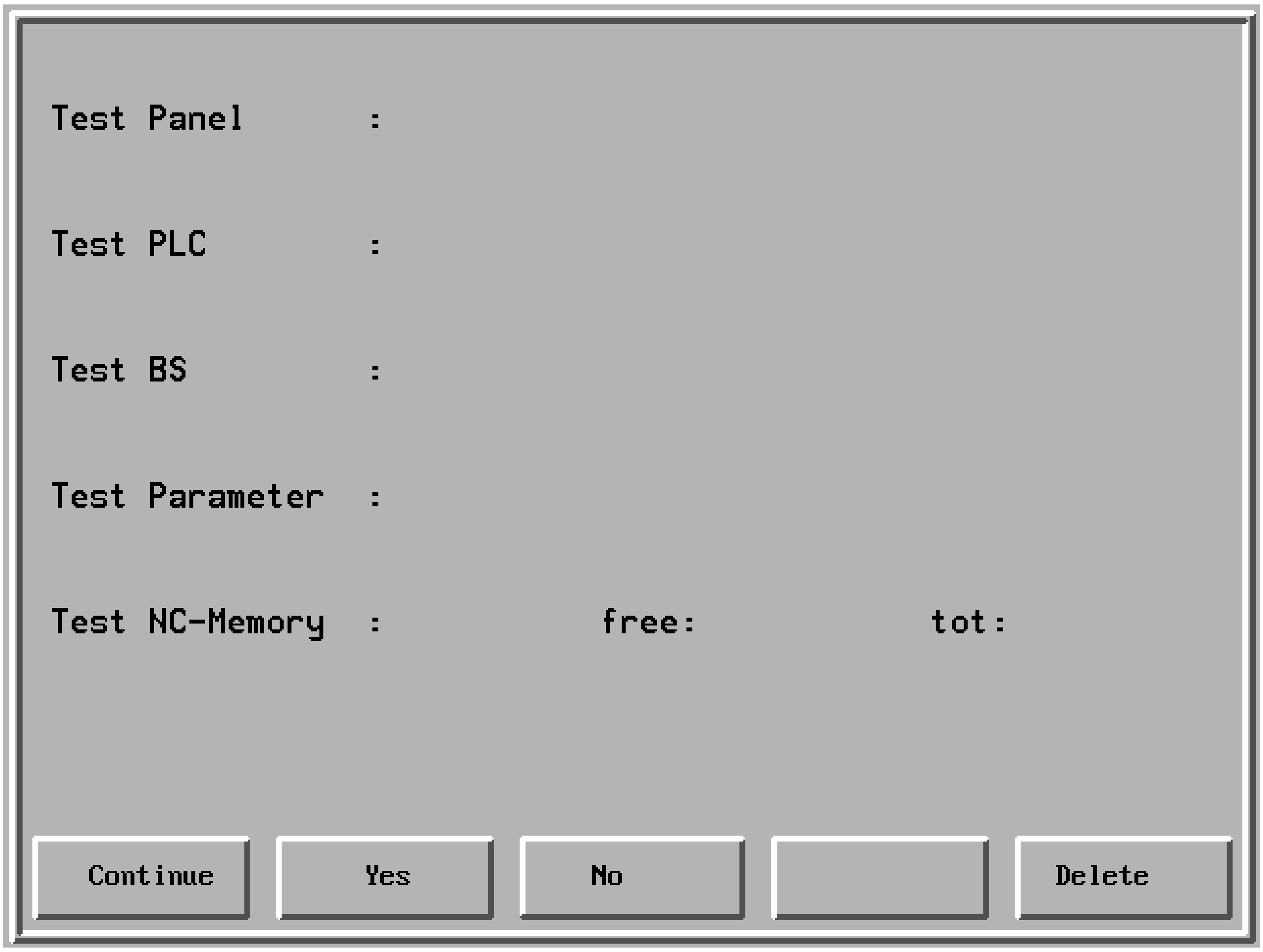
If the key-operated switch is closed (position 1), the power on tests are running automatically, if no errors appear (e.g. def. parameter or def. NC memory).

Closed switch	Position 1	Automatic power on test
Open switch	Position 0	Power on test has to be activated with pressing a key.

2.2.2 Power on test with confirmation

Test operating panel

First, the operating panel is tested (takes some seconds).

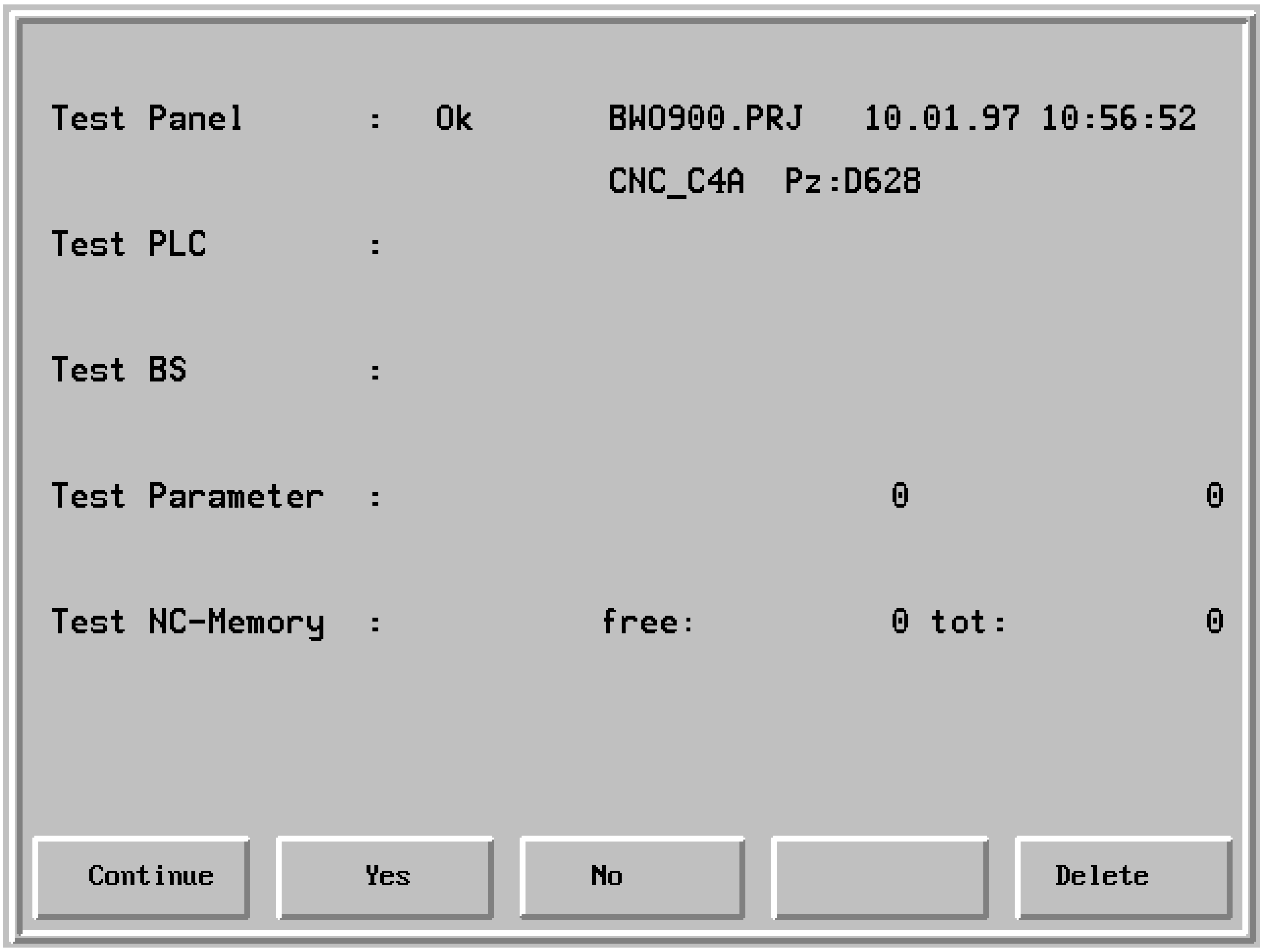


Here the function keys “Continue” and “Yes” and “No” are used combined with the key (Delete).

2.2.2 Power on test with confirmation (continued)

Test operating panel

After a successful test, the following display appears:



Meaning:

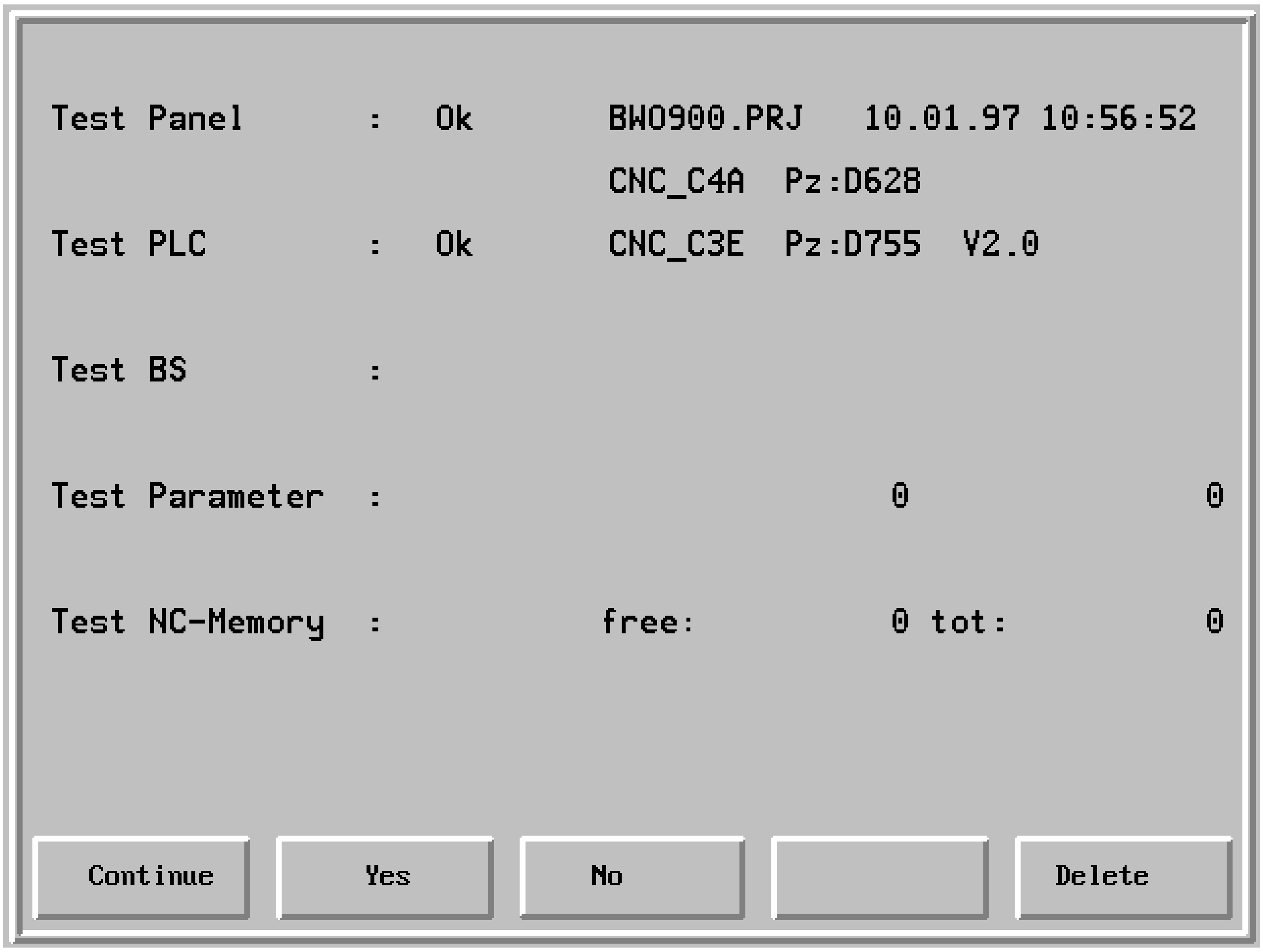
Line 1 Project name of the PROMA surface
with date and time of making the surface (SHOW_E).

Line 2 PLC program, from which the symbols are taken, with check sum.

2.2.2 Power on test with confirmation (continued)

Test PLC program

After a successful test, the following display appears:



Meaning:

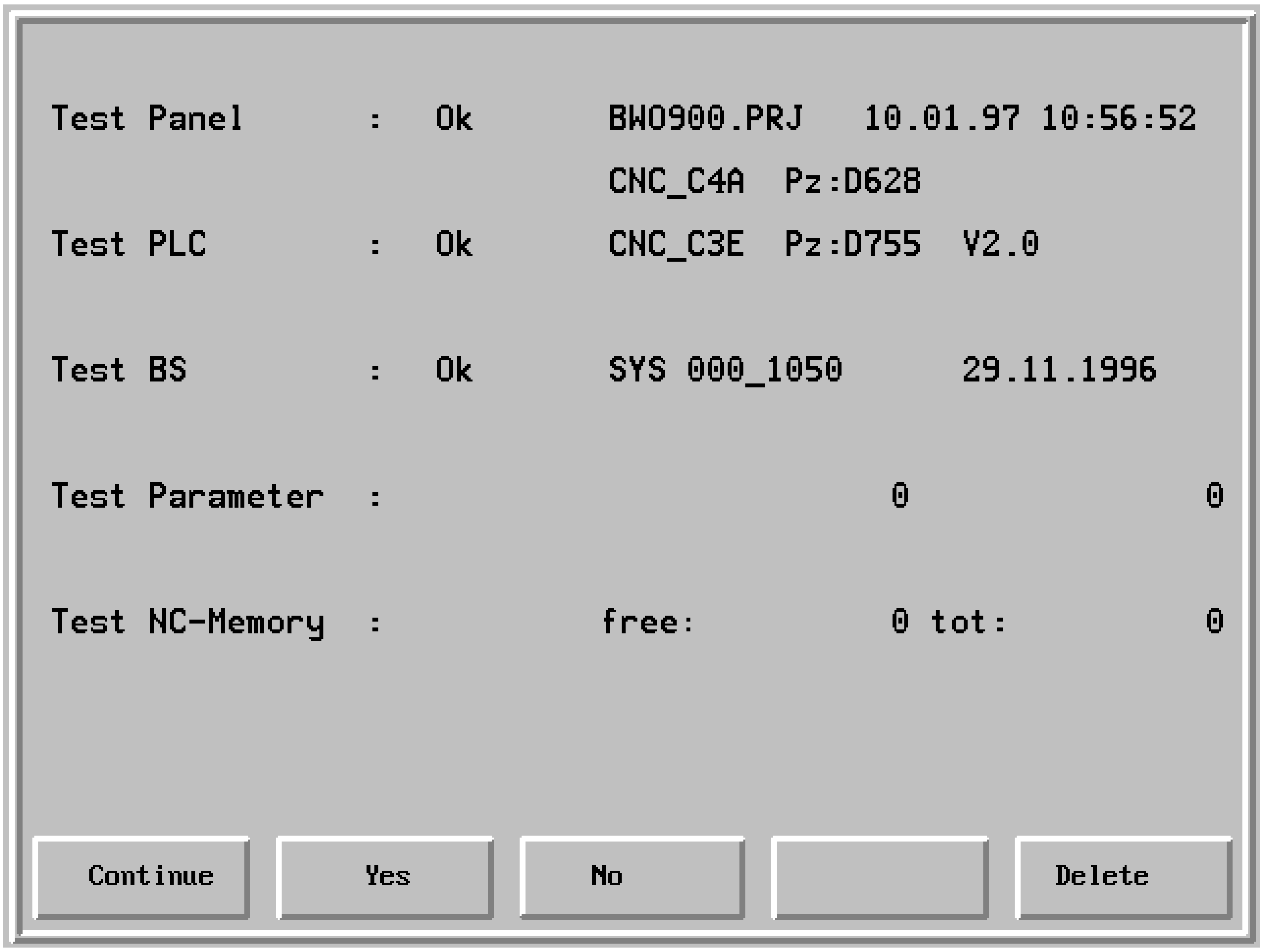
Line 3 Actual PLC program with check sum and number of version

Note: Program name and check sum of the PLC program (operating panel) from which the symbols are taken, and of the actual PLC program should be the same (line 2 and 3).
If the names or check sums are different, it has to be checked if the symbols used in the surface have the same position as they do in the actual NC program.

2.2.2 Power on test with confirmation (continued)

Test operating system

After a successful test, the following display appears:



Meaning:

Line 4 CNC standard operating system with version and date

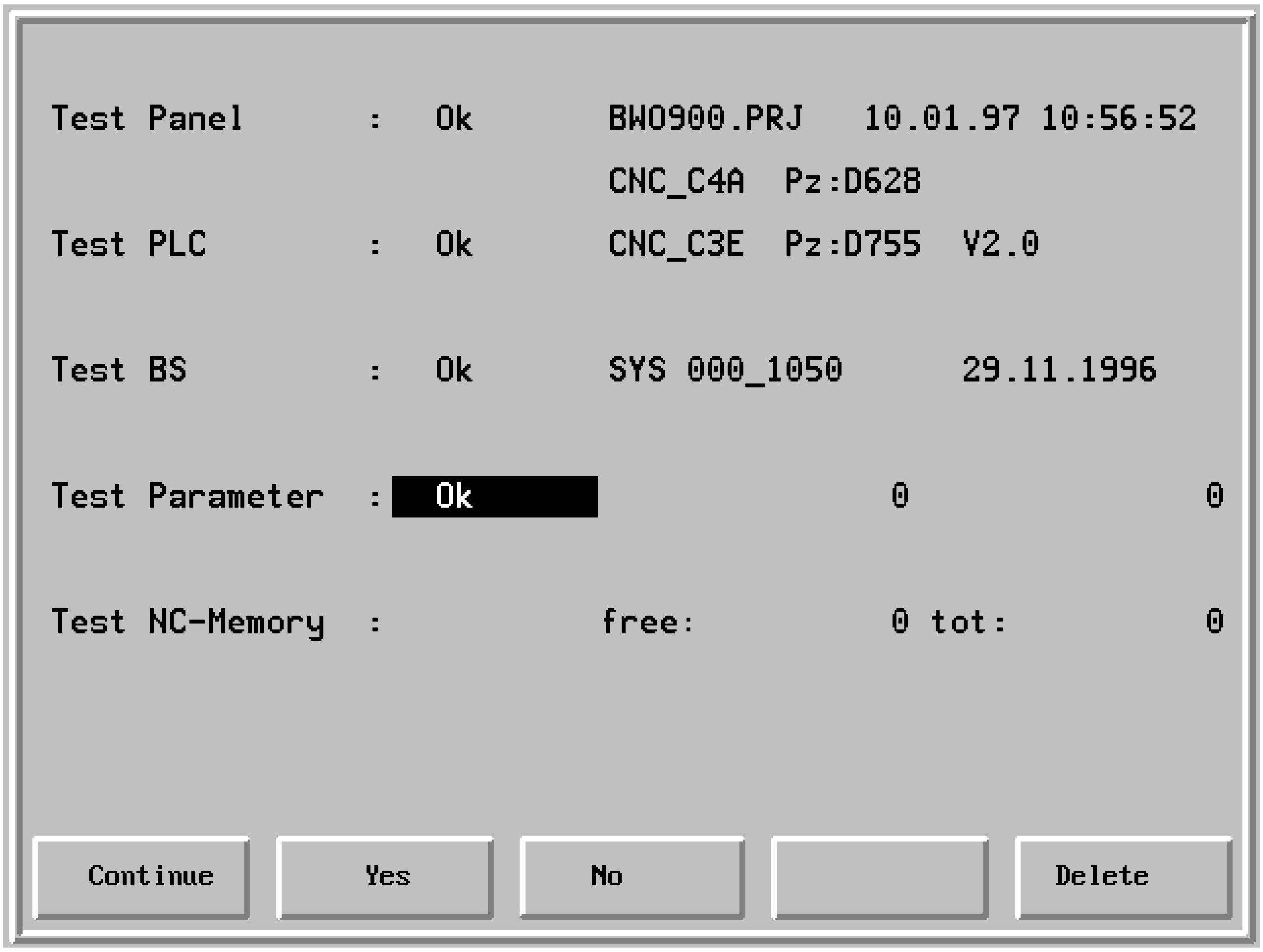
Line 5 Option: DLL-software for customer-specific operating system enlargements with name and date

2.2.2 Power on test with confirmation (continued)

Test parameter

After pressing the function key “Continue”, the parameter memory is checked.

The following display appears after a successful test:



Meaning:

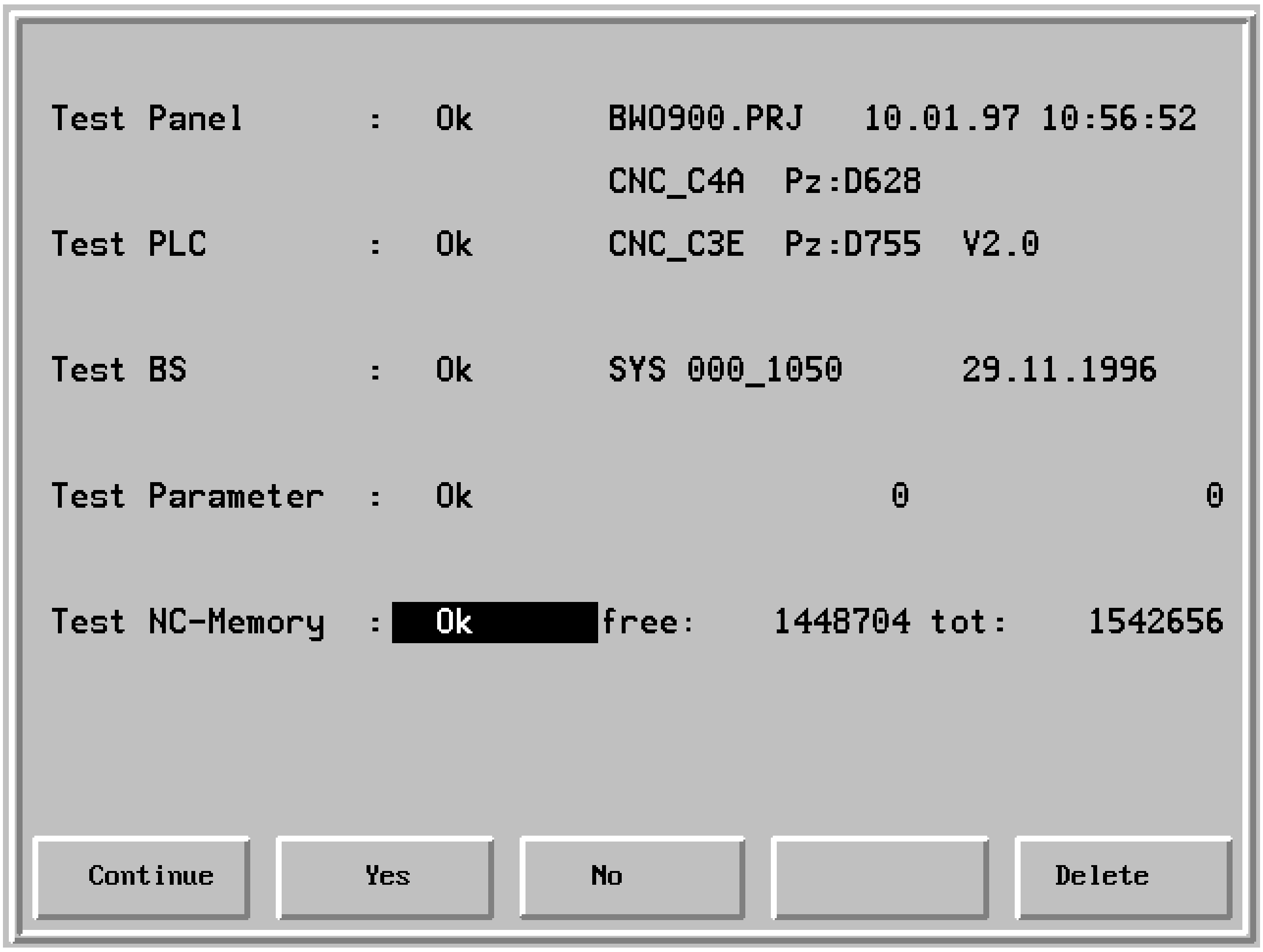
Line 6 The parameter memory is all right.

2.2.2 Power on test with confirmation (continued)

Test NC memory

After pressing the function key “Continue”, the NC memory is checked.

The following display appears after a successful test:



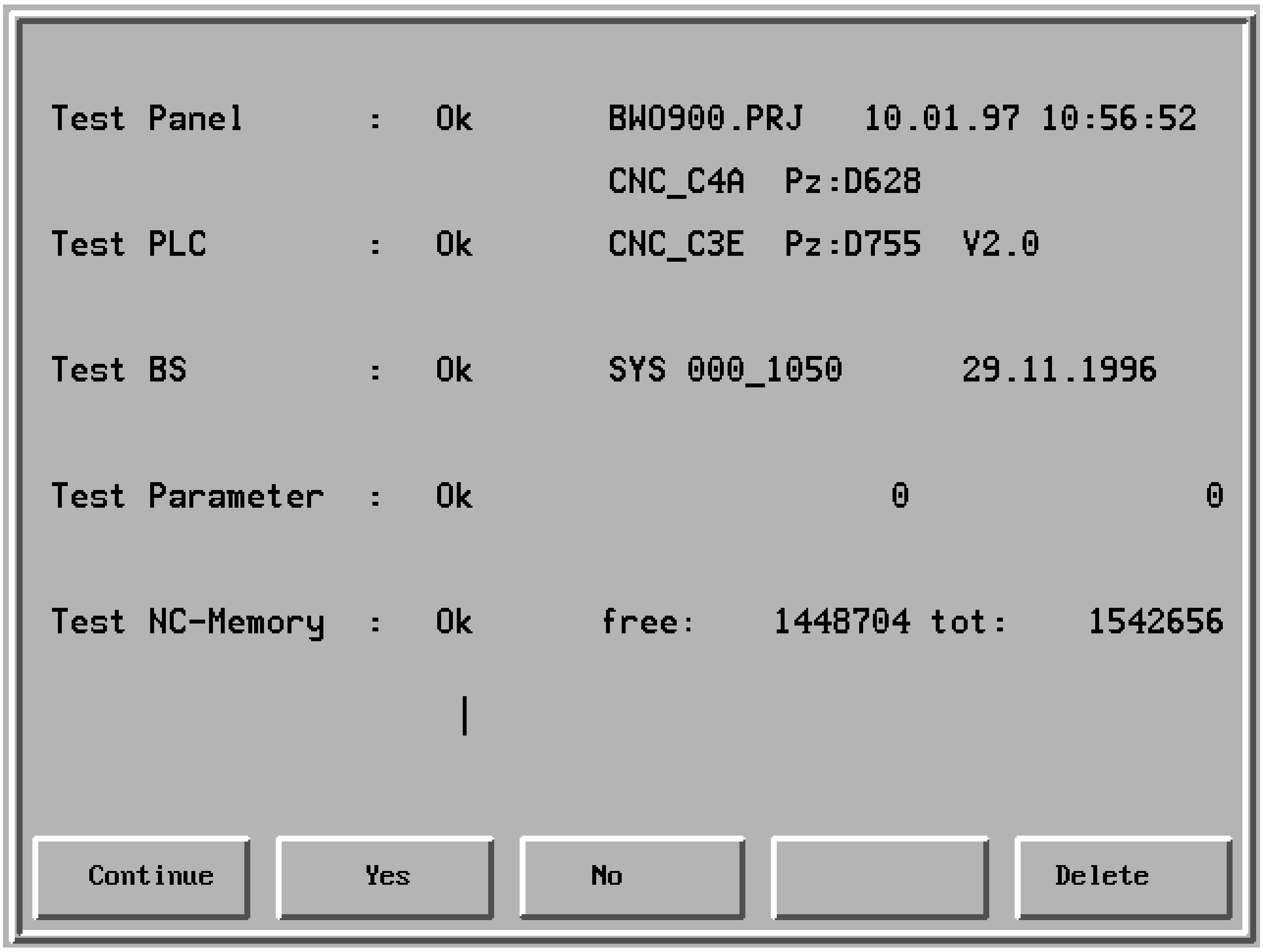
Meaning:

Line 7 NC memory is all right
Indicating free memory and complete memory

2.2.2 Power on test with confirmation (continued)

Finish power on test

After pressing the function key “Continue” the power on test is finished and the system is loaded (takes some seconds).
The following display appears after a successful test:

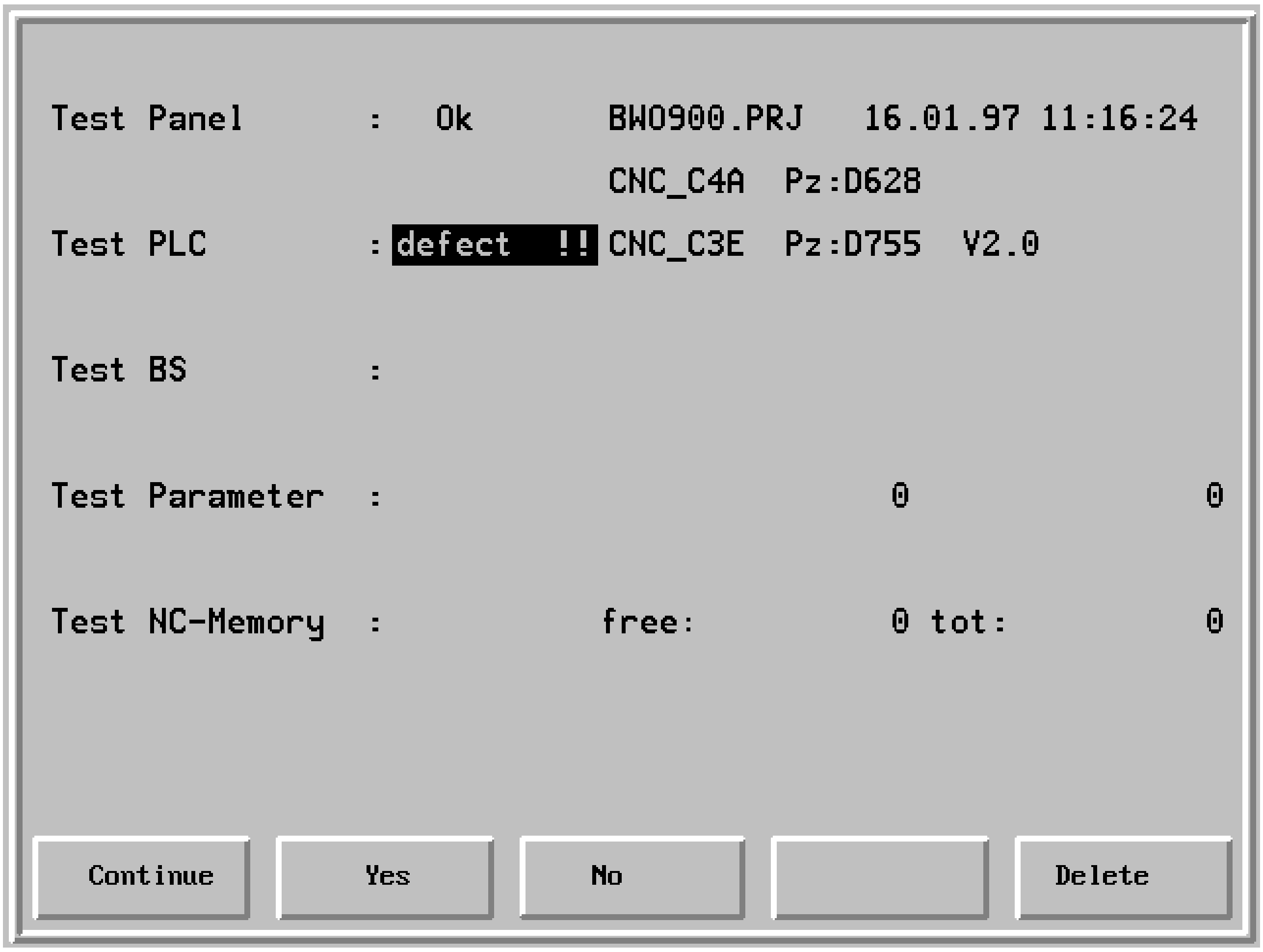


Then, the company sign appears.

2.2.3 Power on test with error

PLC test

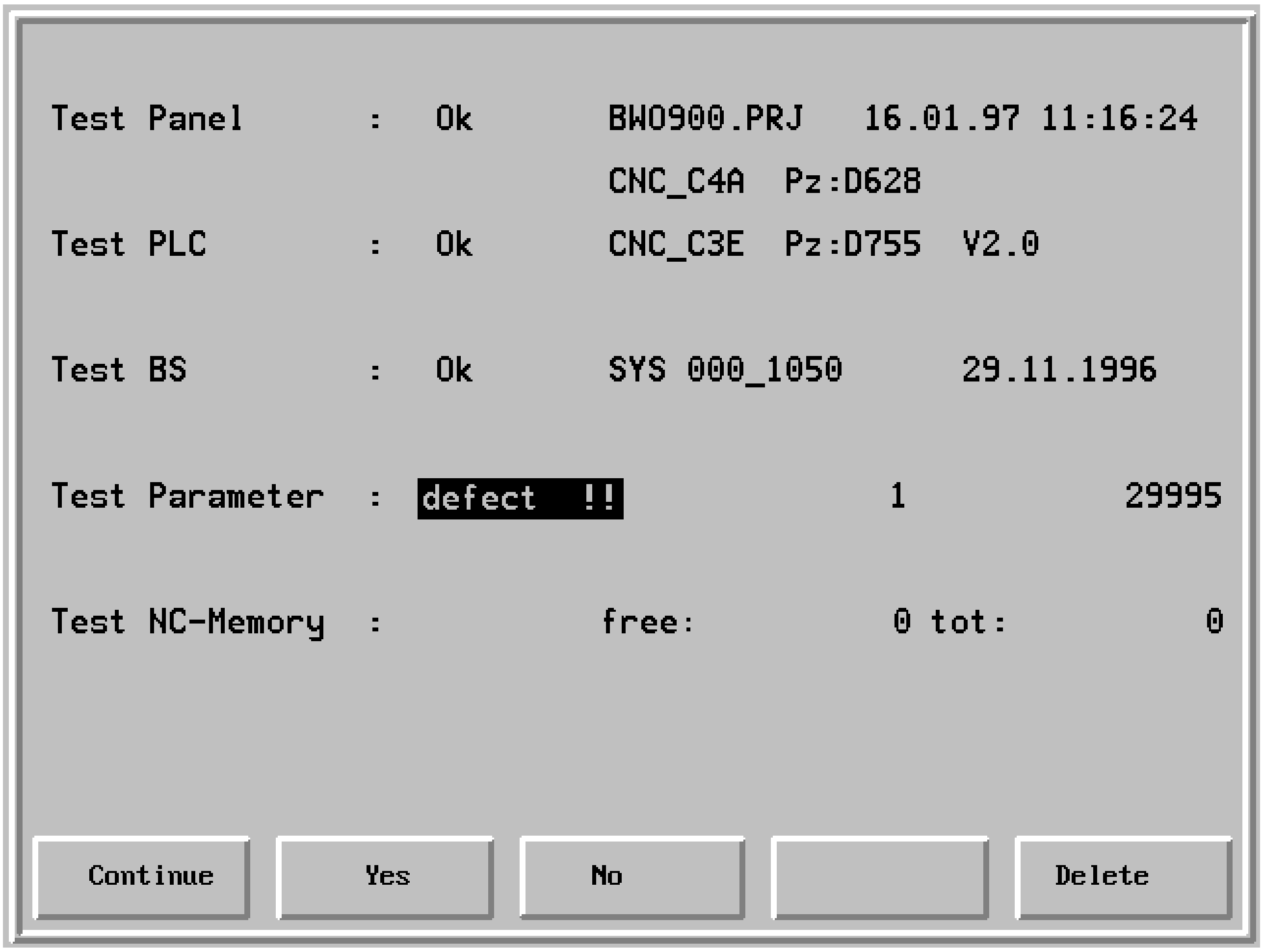
If the following display appears, the PLC program is defective. The power on test is stoped.
In this case, the PLC program has to be loaded again and has to be stored in the EEPROM.



2.2.3 Power on test with error (continued)

Parameter test

The following display appears if the parameter memory is defective. The power on test is stoped.



Meaning:

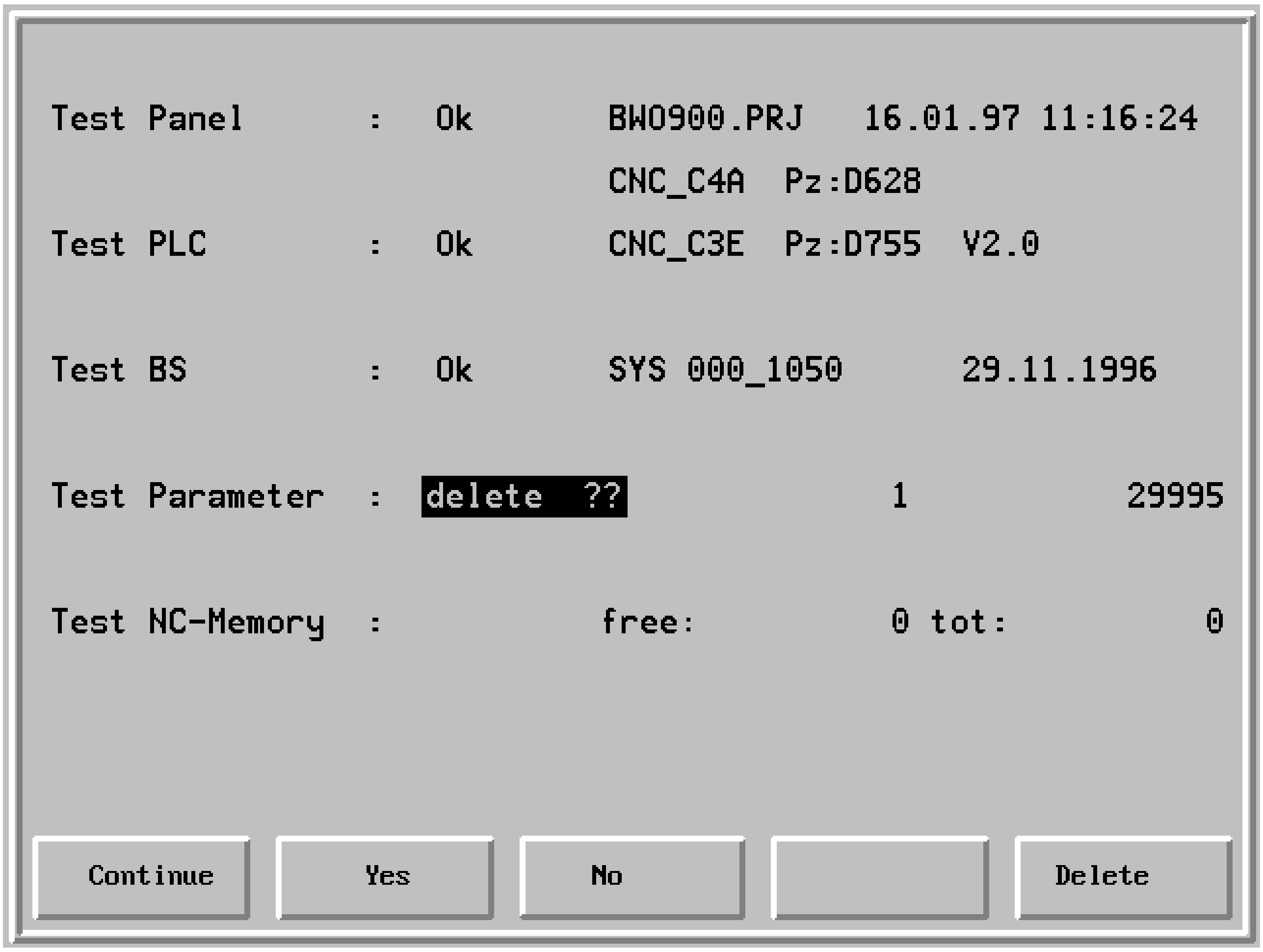
Line 6 Display of the first defective parameter (q) and the number of defective parameters.

In exceptional cases it can be practical, despite defective parameters the switching on check to continue. This actual with key possible.

2.2.3 Power on test with error (continued)

Parameter test

The power on test is only continued if the key (Delete) is pressed. The following display appears.



Delete ??

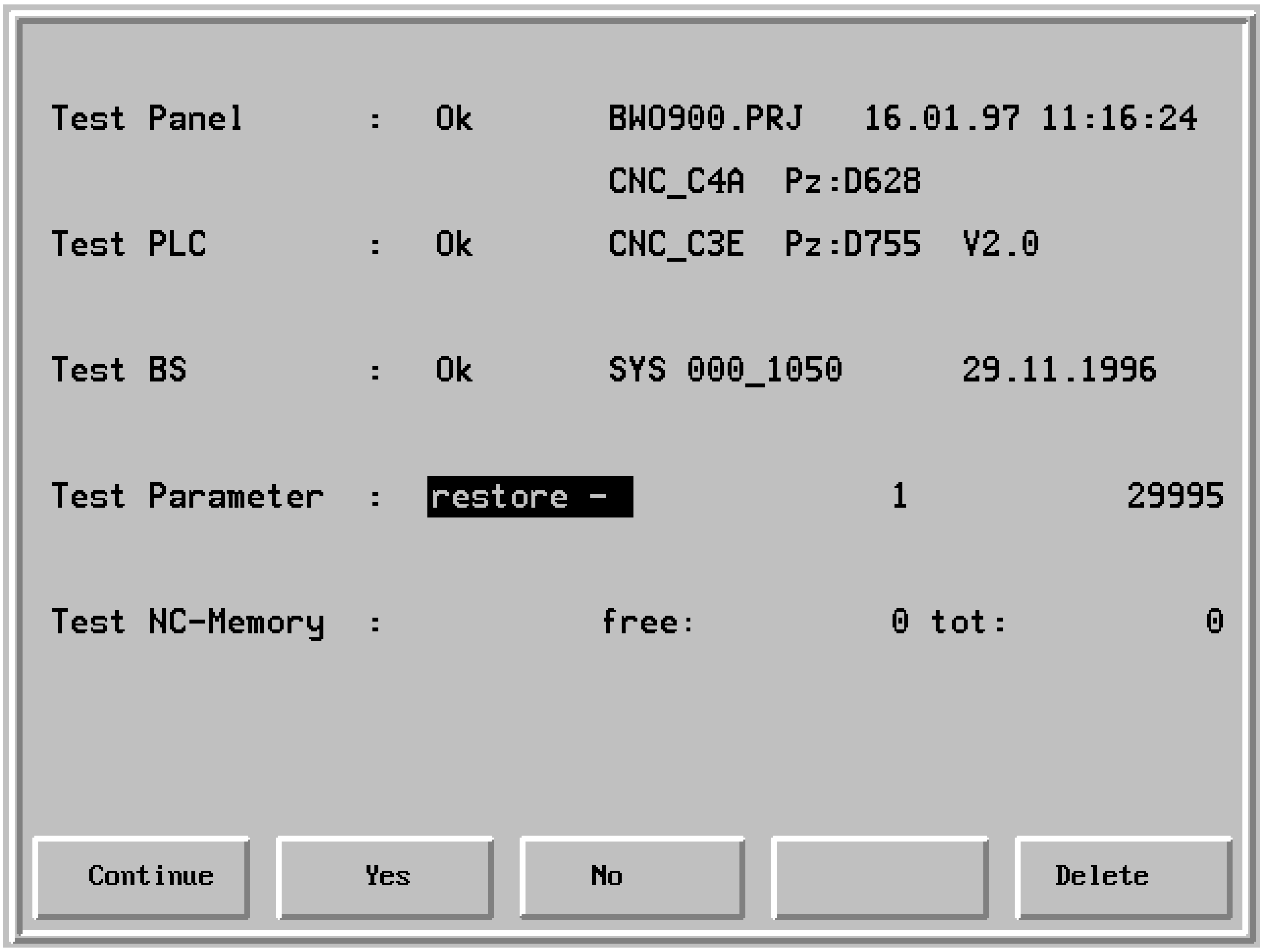
Key “Yes” Delete the complete parameter memory

Key “No” The parameter memory is not deleted.
Go on to test NC memory with pressing the key “Continue”

2.2.3 Power on test with error (continued)

Parameter test

After Delete ? and pressing the key “Yes” the following functions are loaded
- BWO presettings
- customer-specific data from the EEPROM.

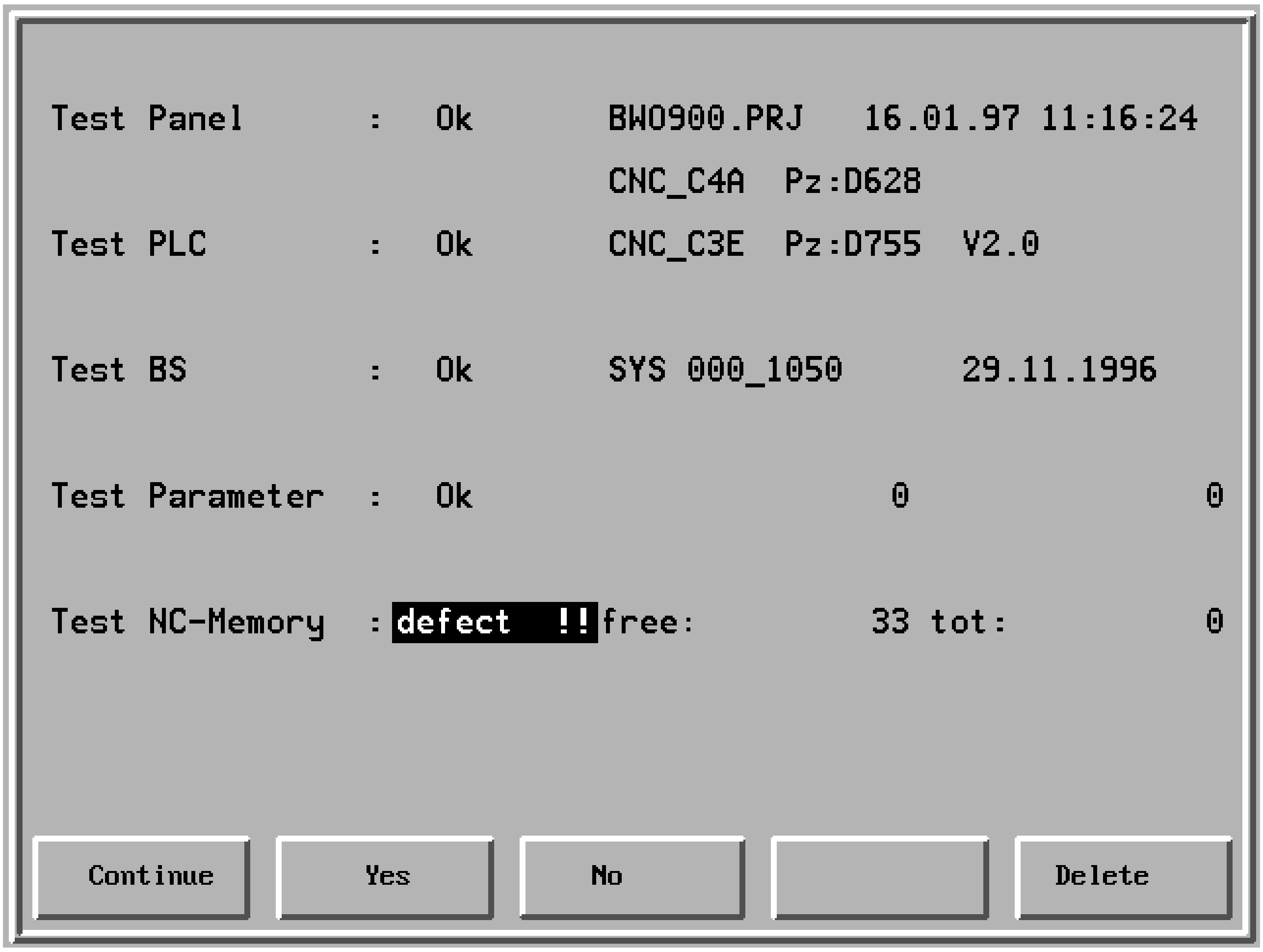


Go on to test NC memory with pressing the key “Continue”

2.2.3 Power on test with error (continued)

Test NC memory

The following display appears if the NC memory is defective. The power on test is stoped.



Meaning:

Line 7 Error code (33) and additional information (0)

In exceptional cases it can be practical, despite defective NC Speichers the switching on check to continue. This actual with key ↘ possible.

2.2.3 Power on test with error (continued)

Test NC memory

Meaning of the error codes

Code	Meaning
02	Program not found
05	End of program or Write-/ Read indicator > program size
06	No free memory
09	Faulty check-sum
10	Error when reading from NC memory
11	Error when writing on NC memory
15	No program input , e.g. key-operated switch locking
16	Program should be opened for writing, but is already opened
25	Key-operated switch interlocking
30	Defective directory chain
31	Defective program chain
32	Defective list of free blocks
33	Defective check-sum of system data
35	Blocks in program system are double-chained

2.2.3 Power on test with error (continued)

Test NC memory

The power on test is only continued if the key (Delete) is pressed. The following display appears.

Delete ??

Key “Yes” Delete complete NC memory

Power on test is finished with pressing the key “Continue” and the company sign appears.

2.2.4 Company sign

2.3 Reference points

Reference points are machine-specific mechanical fixed points.

After the power on tests, you have to approach with each axis the corresponding reference point, if the machine works in the incremental measuring system.
The reference point is stored and the actual value display is set with the correct value.
In the first line of the display field, the axes of which the reference point is not stored, are shown.

Because the traverse directions for taking reference points are depend on the machine, the following explanation can only be an example.

Note: The reference point must not be identical with the machine zero point.

2.3.1 Approaching reference points automatically

The reference points can be approached automatically, if a corresponding program is stored in the PLC. The key  is reserved for this program.

2.3.2 Manual approaching of reference points

Requirements

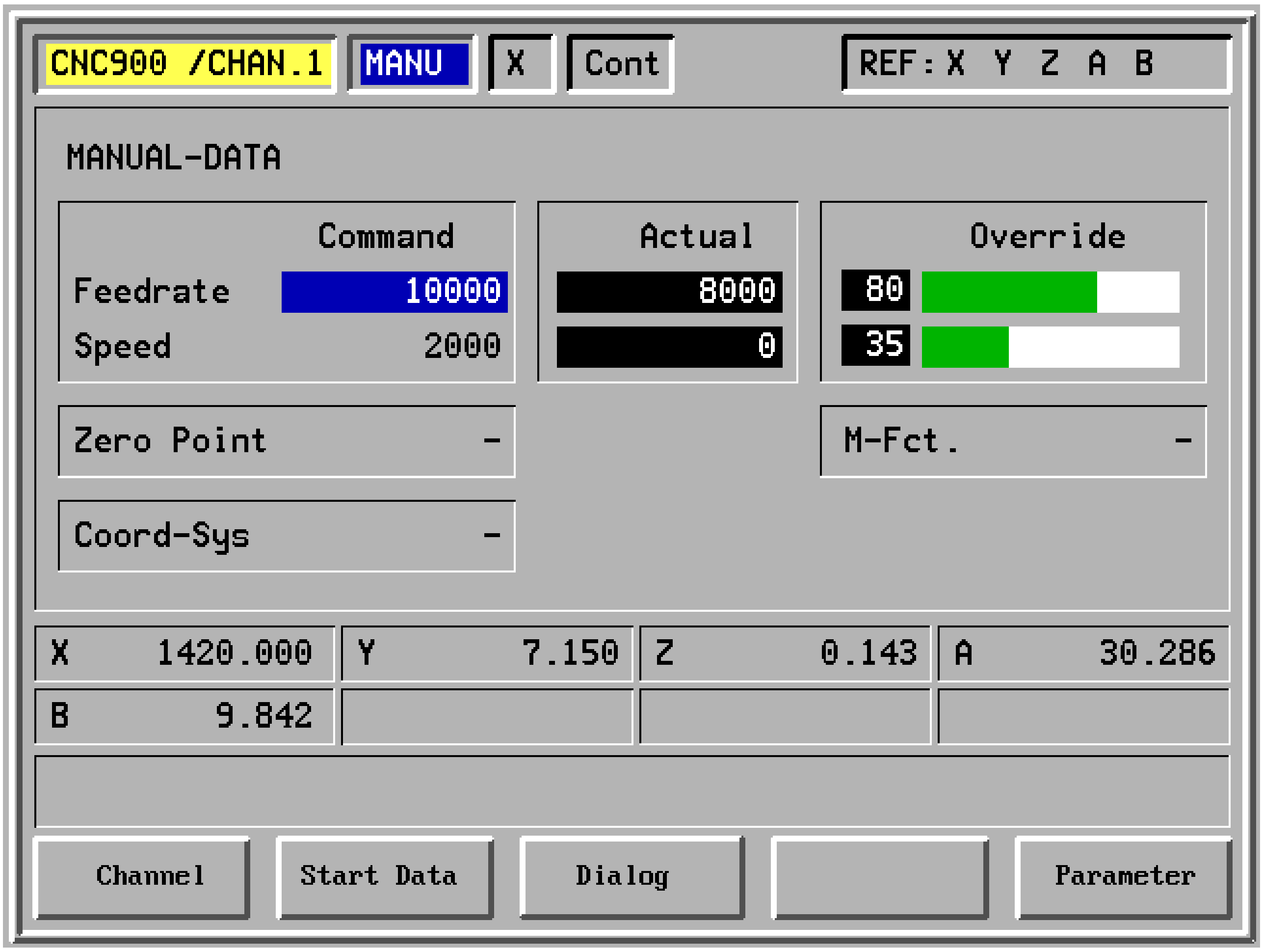
- Power on tests are ready.
- The control is switched on (key “Control on” is pressed)
- The operating mode “Manual” is set.
- The feed for manual mode is given, feed override > 0.
- The drive mode is set (e.g. “continually”).

Approach reference point of an axis, e.g. X-axis

Press key “X”

Display: MANUAL X Cont Ref: X Y Z A B

Press key “Manual” until max. end position of the X-axis is reached.



2.3.2 Manual approaching of reference points (continued)

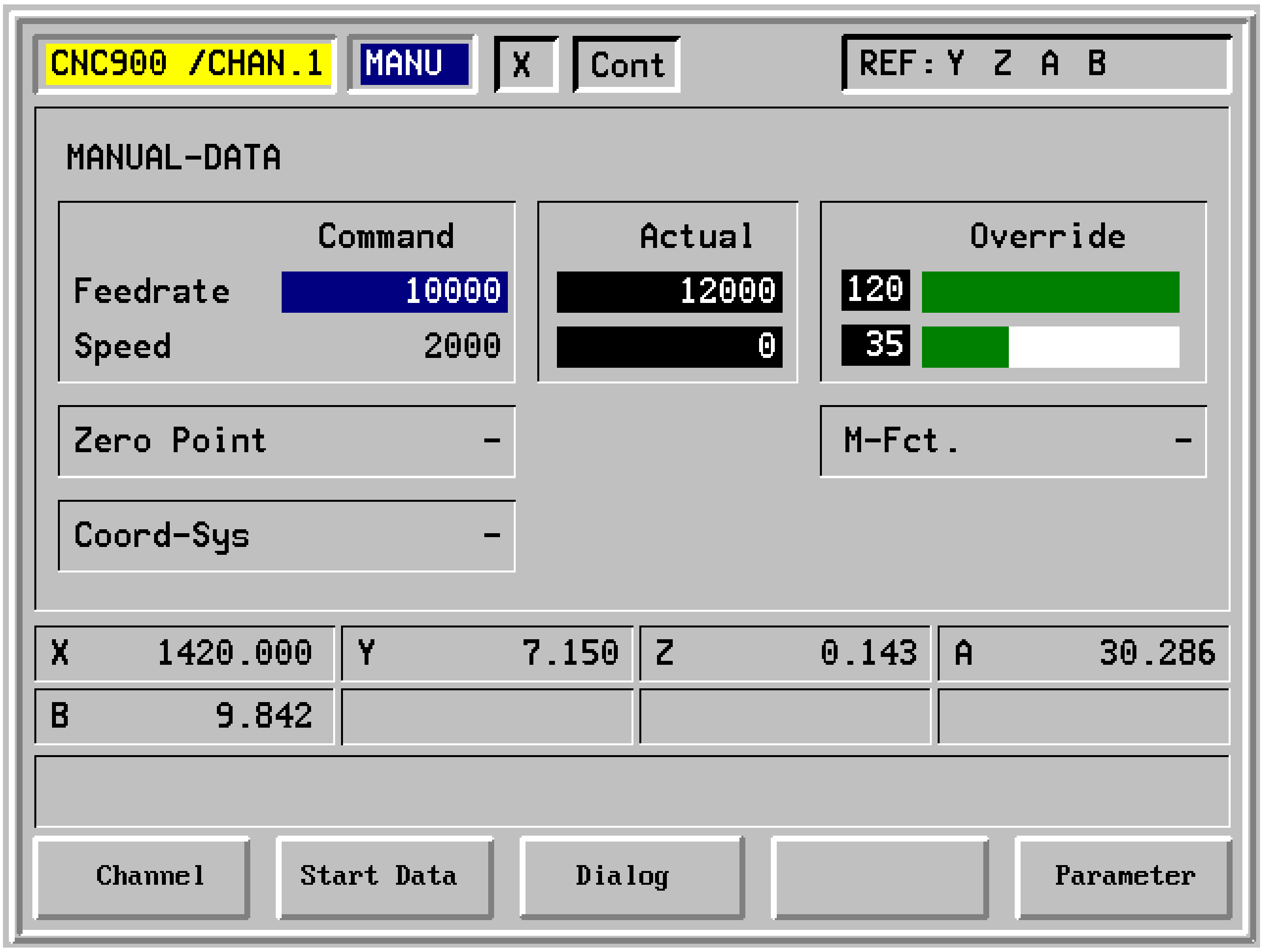
Approach reference point of an axis, e.g. X-axis

Then press “Manual” until the X disappears from the display.

Display: Manual X Cont REF: Y Z A B

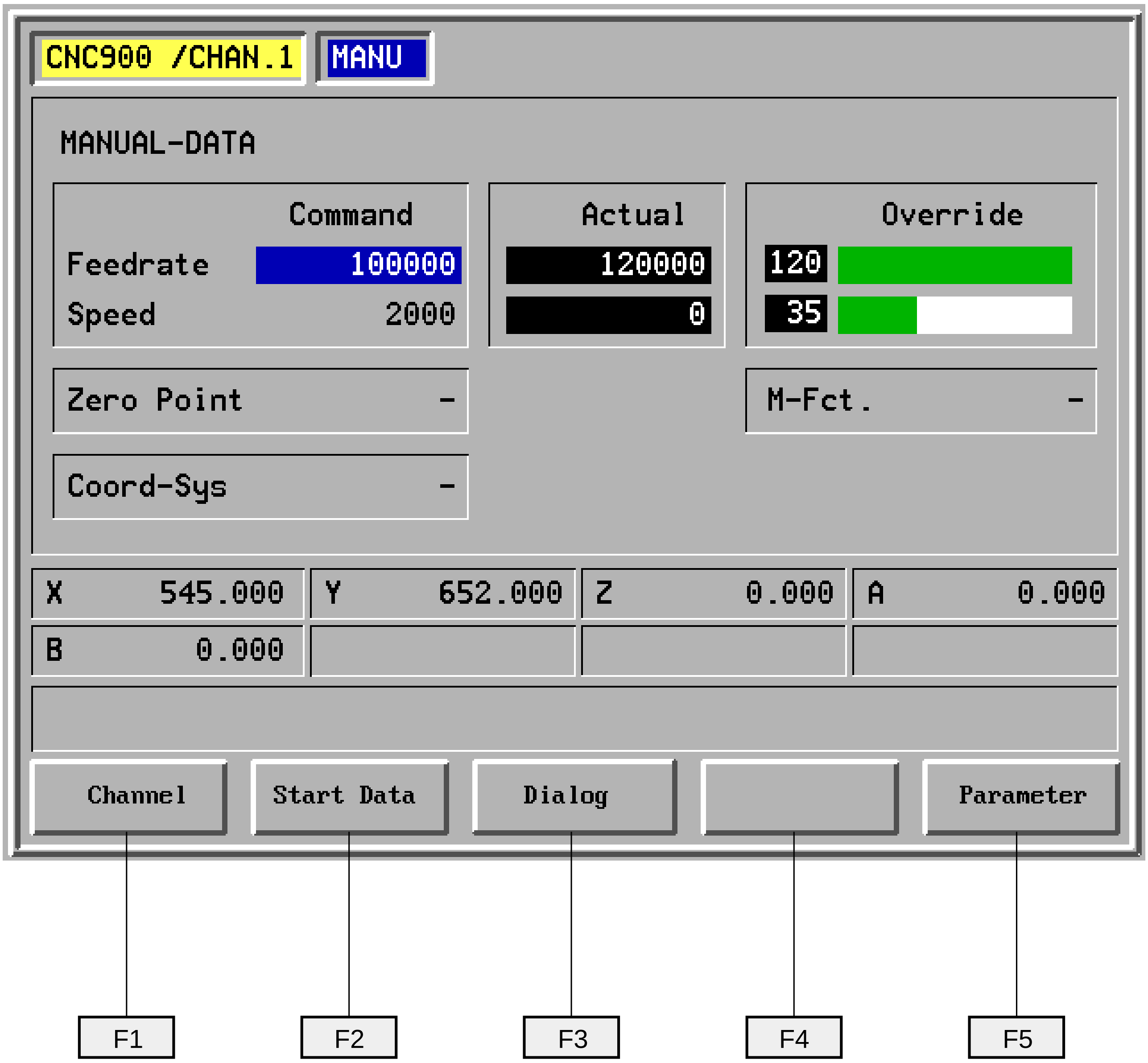
Now the reference point for the X-axis is stored.

The same procedure is valid for the other axes.



2.4 Screen keys and screen displays

2.4.1 Position of the function keys in the screen frame



F7 **back** to the previous picture

roll function **continue** F6

2.4.2 Setting the screen brightness

The brightness of the screen can be set continuously.

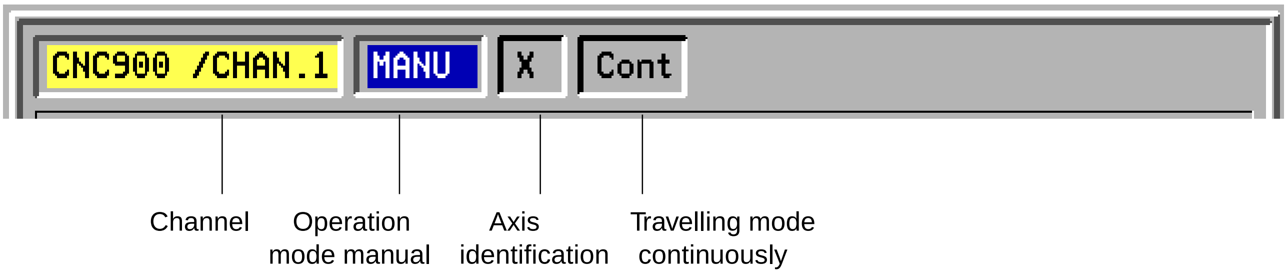
When pressing the function key “**F6**” and

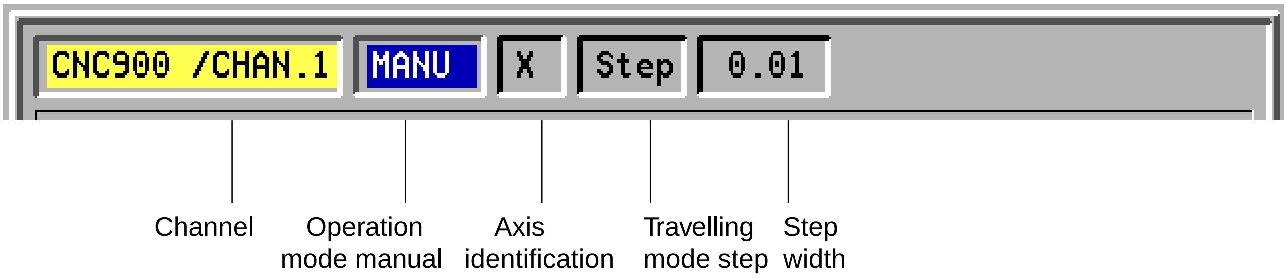
 (Page up) the creen becomes brighter,

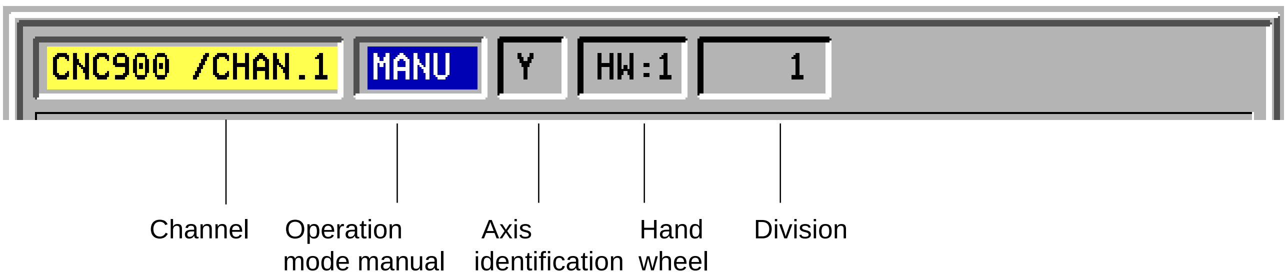
 (Page down) the screen becomes darker.

2.4.3 Screen display in the headline

Displays in manual mode

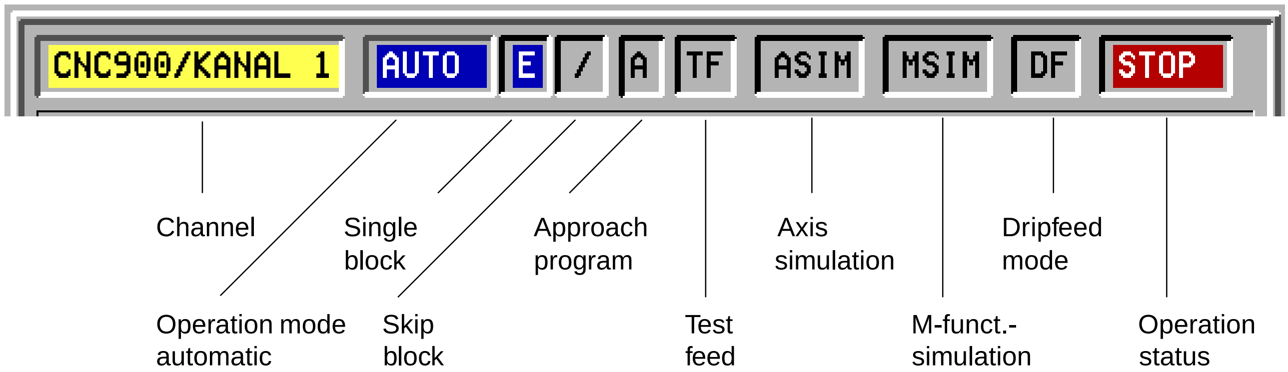






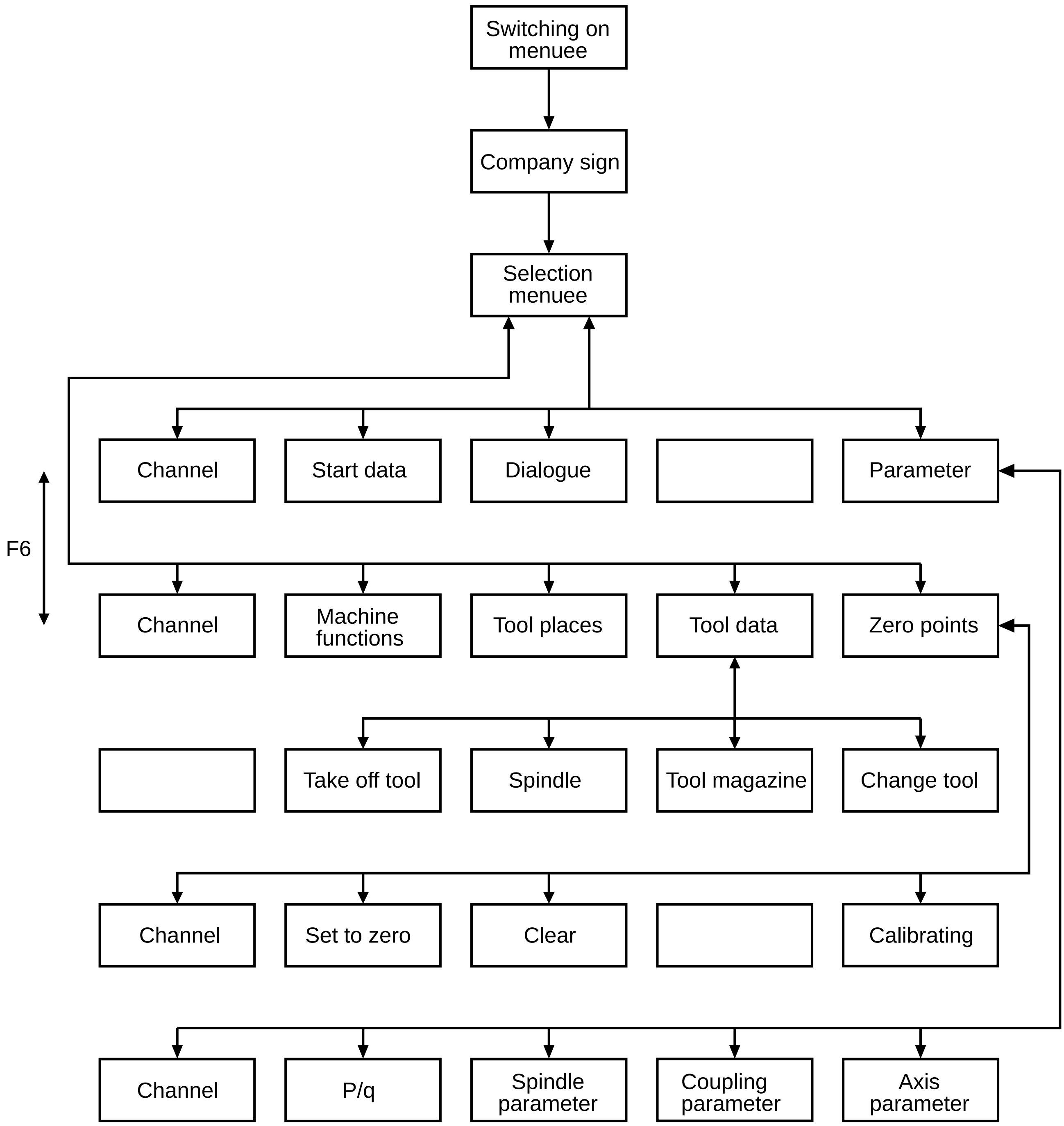
2.4.3 Screen display in the headline (continued)

Displays in automatic mode



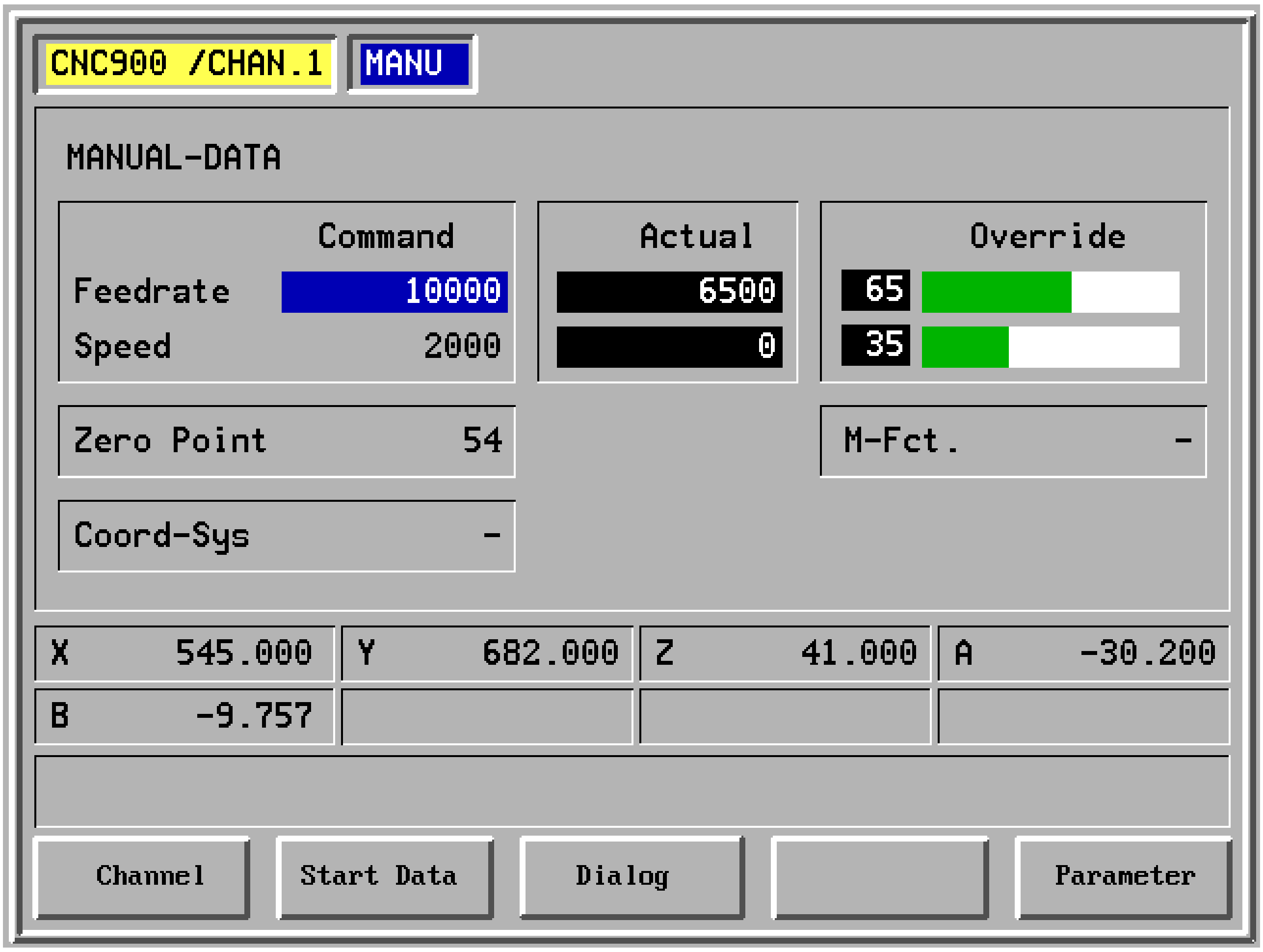
2.5 Selection menue

Menue tree



2.5 Selection menue (continued)

All functions can be selected with the selection menue with the function keys F 1 to F7.
The menue consists of two function pictures. **Selection picture 1:**



Selection of

F1 Channel

F2 Start data

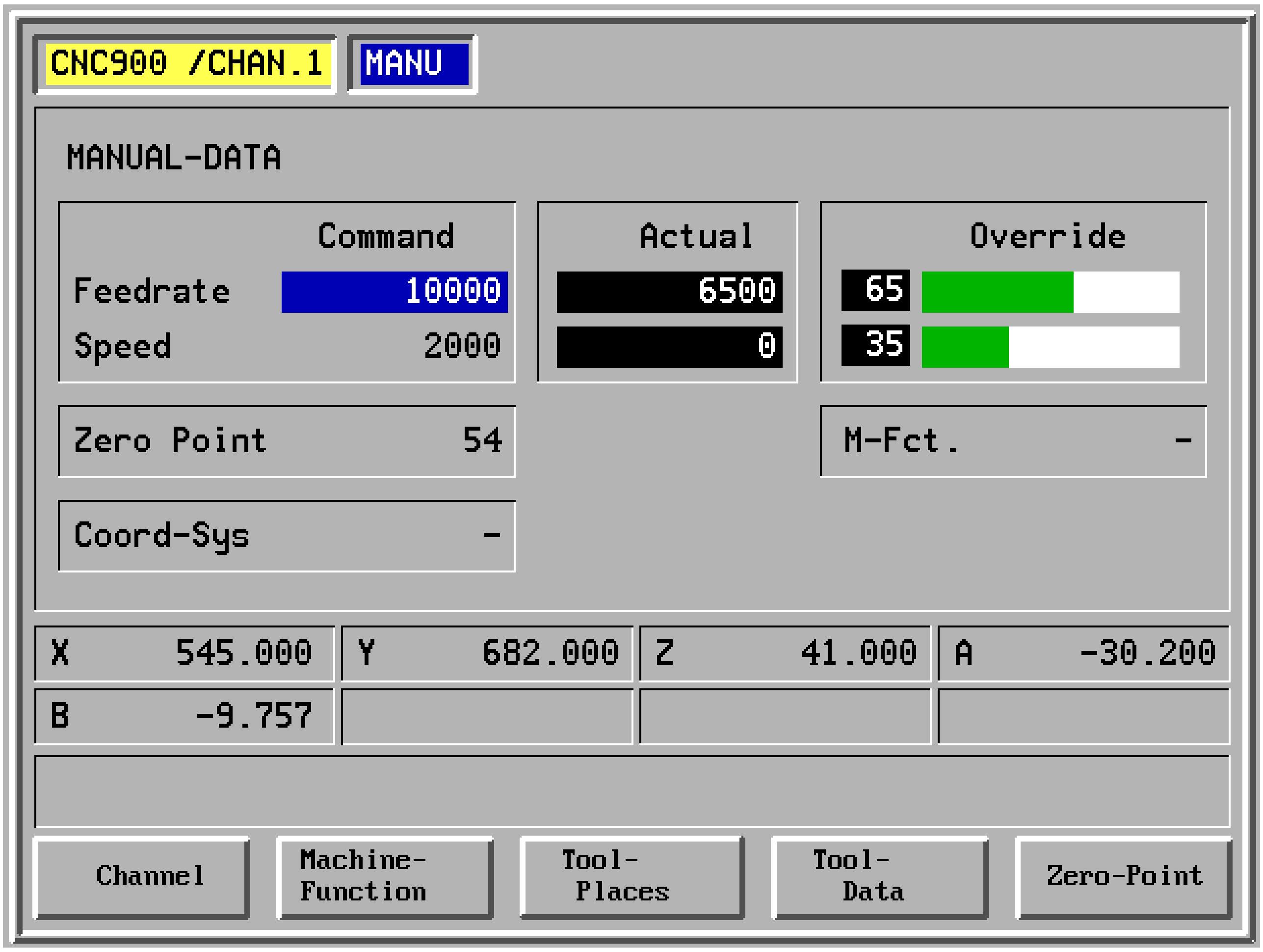
F3 Dialogue

F4 -

F5 Parameter

2.5 Selection menue (continued)

Selection picture 2:
Switching from selection picture 1 <—> selection picture 2 with function key F6 (roll function)



Selection of

F1 Channel

F2 Machine function

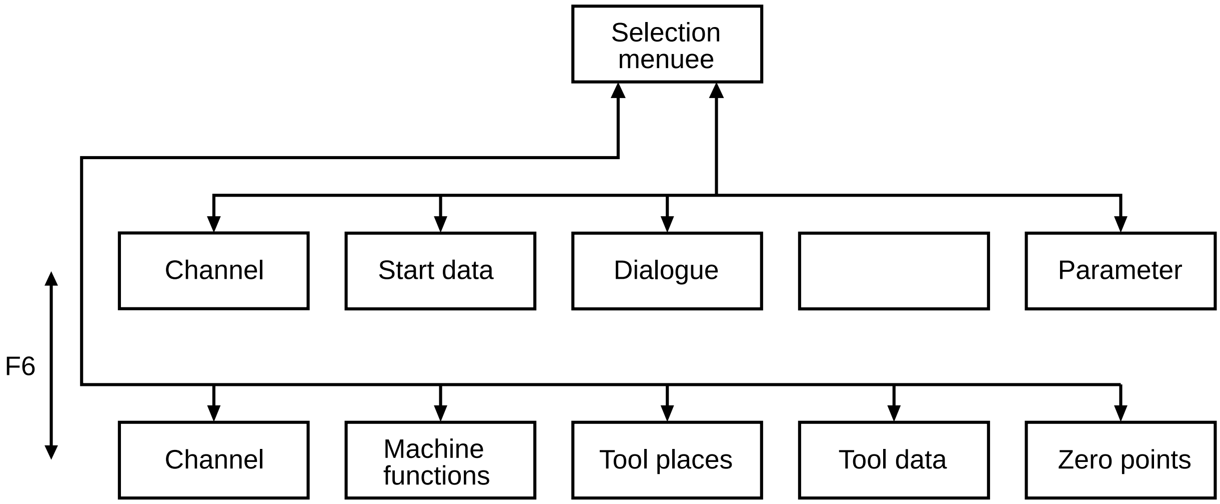
F3 Tool places

F4 Tool data

F5 Zero points

2.5 Selection menue (continued)

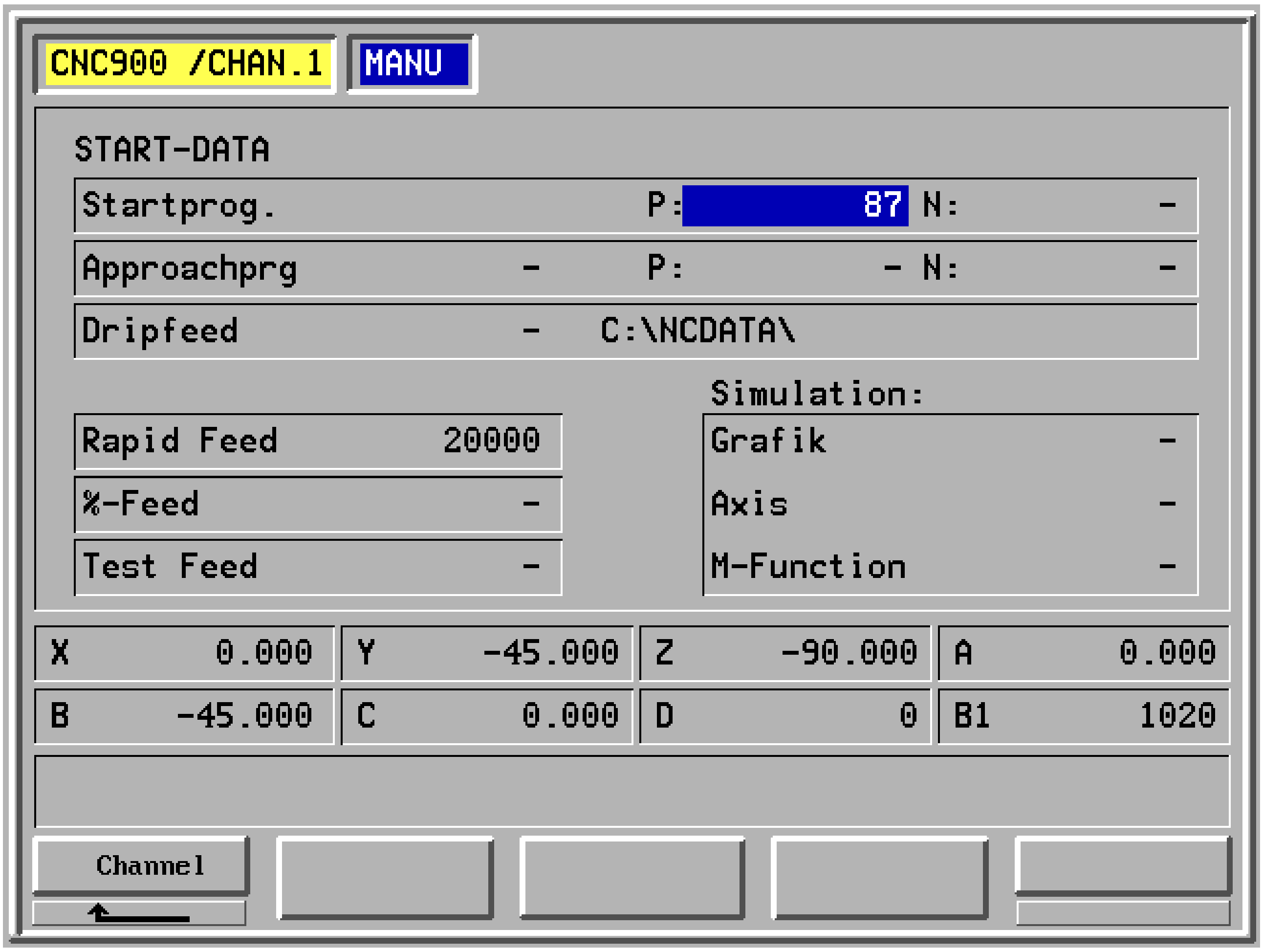
Menue tree



Functions	Section
Start data	2.5.1
Dialogue	2.5.2
Parameter	2.5.3
Machine functions	2.5.4
Tool data	2.5.5
Zero points	2.5.6

2.5.1 Start data

The following menue enables setting the start data.



F1 Channel

F2 -

F3 -

F4 -

F5 -

2.5.1 Start data (continued)

Block

Input block number of the starting block. If the value is 0 or deleted, the NC program is started with the 1st block.

Rapid traverse

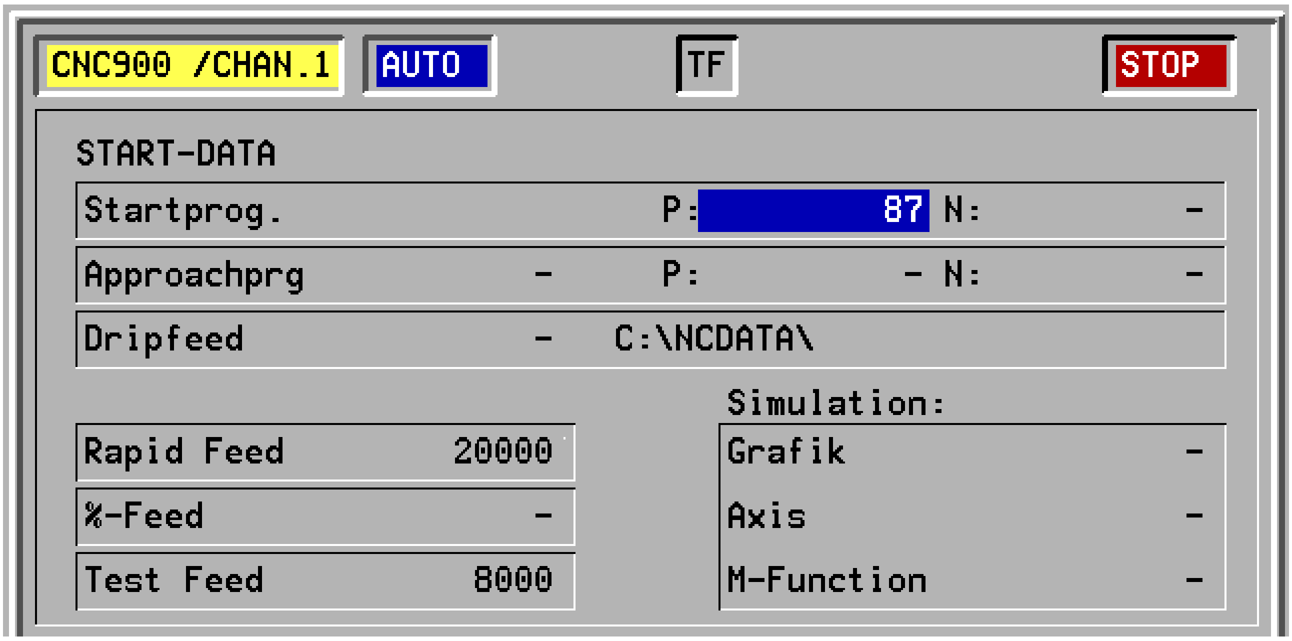
The desired rapid traverse can be input. If the value is 0 or deleted, the manual feed is active.

% Feed

The programmed feeds in the NC program are modified with the indicated percentage.

Test feed

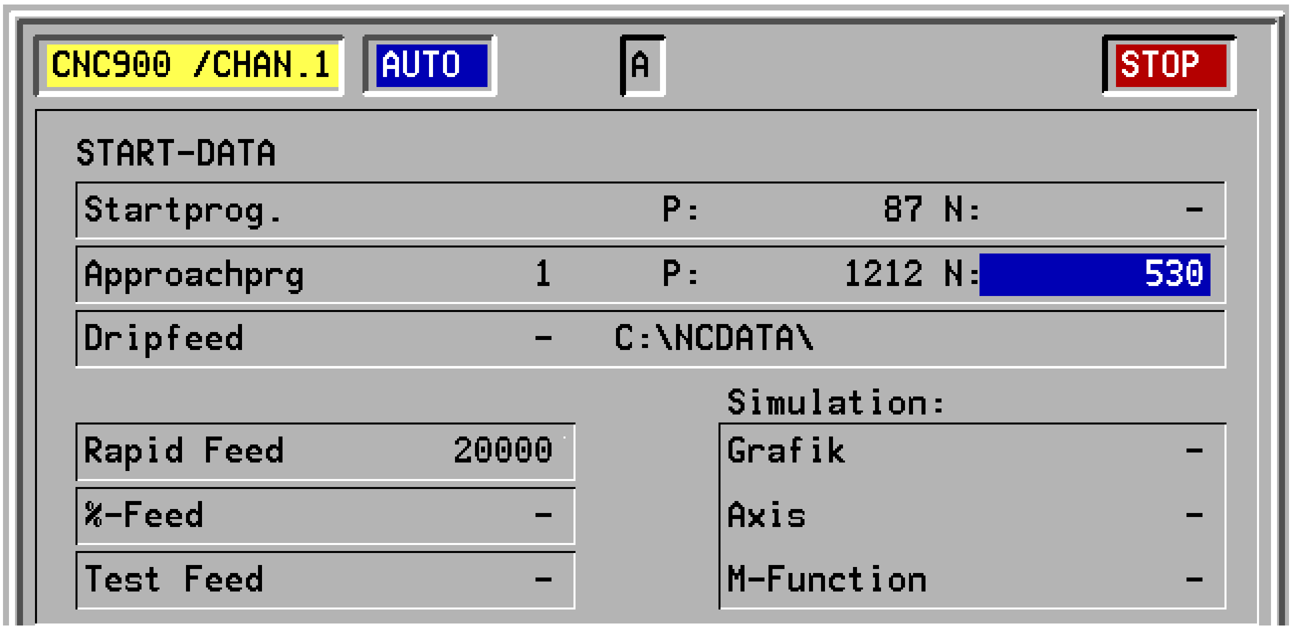
If a test feed is input the programmed feeds in the NC program are ineffective.
The test feed is active. If the value for the test feed is 0 or deleted, the test feed is not active.



2.5.1 Start data (continued)

Reapproach program

In the case of the abort of a current NC program (HAND abort, message) by the system the actual NC program position in the parameters P8802, P8803 and P8807 is stored. The NC program position is displayed in the start data menu. On properly terminated NC program resets this information.



After an NC abort if the restarting mode is switched on (P8804=1)und the NC program started, then the NC program up to the restarting point is simulated, i.e.. no axis movements result.

In the restarting block the simulation is switched off. The restarting position is started on direct path. With achieving the restarting position P8804=0 is settinged.

One re+starts on the initial position of the aborted block.

If the NC program in a process cycle is aborted, then becomes to cycle start started.
When restarting m-functions and m-cycles are treated according to m-Funktionsdefinition (P11050..., P8250...).

2.5.1 Start data (continued)

Dripfeed mode

The NC programs that should be processed are not in the NC memory of the control but in external data carriers.

The operating panel **CNC 900 C** has the NC programs on the hard disk or on a data server.

With the operating panel **CNC 900**, the NC programs have to be read in via I/O (BWO-I/O-socket, NCARC.EXE).

NC programs for dripfeed have to be linear, i.e. block skips and sub-program call-ups are not allowed.

Activating in the start-data menue:

Dripfeed	1	Dripfeed on
Dripfeed	0	Dripfeed off

Dripfeed mode with operating panel CNC 900 C

- In the start-data menue:
  switch on dripfeed mode,
  input program number,
  check DOS-path-name,
  path-name is indicated in the start-data menue (see also CNC900X.CFG).

- Change of operating mode after AUTOMATIC.
  Now the data transmission to the dripfeed-buffer is started.
- NC start.

Dripfeed mode with operating panel CNC 900

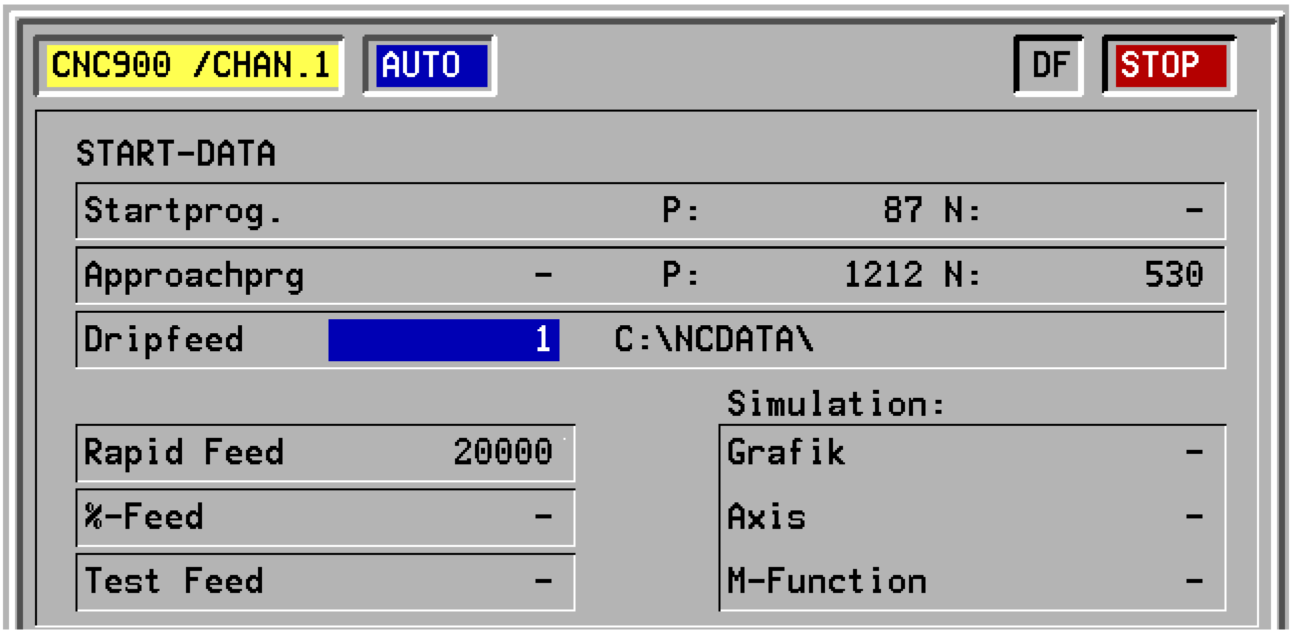
- In the start data menue:
  switch on dripfeed mode

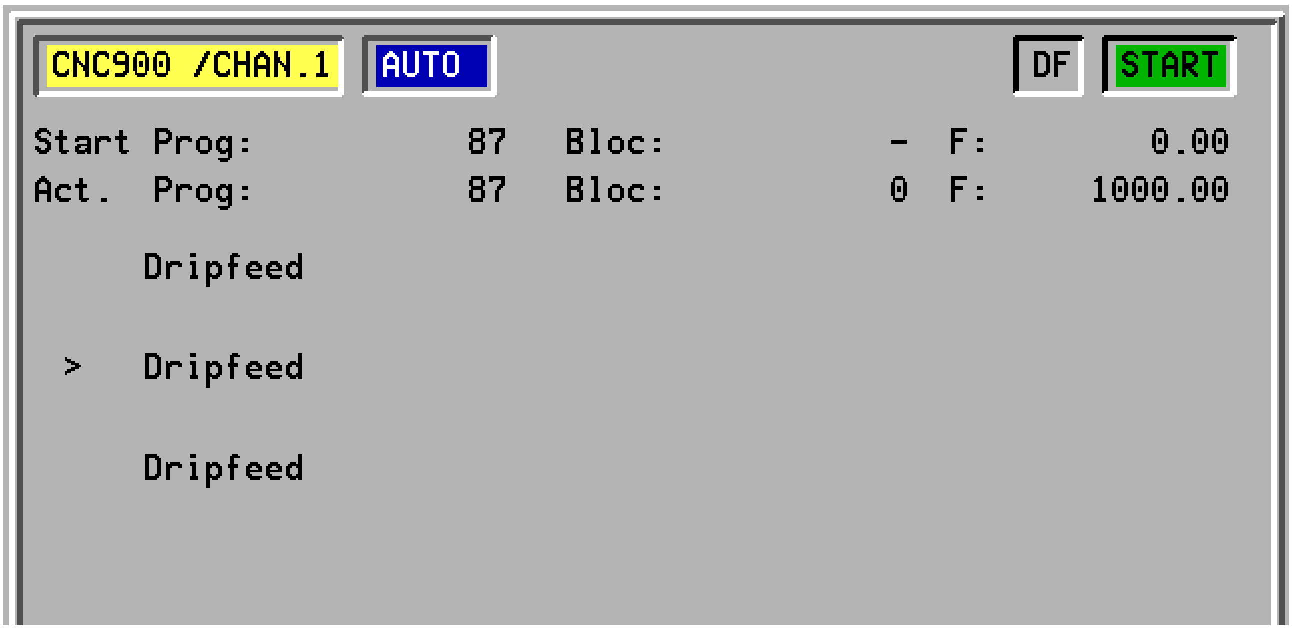
- Check I/O parameters
- Connect external data carrier
- Change of operating mode after AUTOMATIC,
  due to this an I/O-input-start is made automatically.
  The CNC is now waiting on a data transfer.
- Start data transfer at the external data carrier,
- NC start

2.5.1 Start data (continued)

Dripfeed mode

with operating panel CNC 900 C





2.5.1 Start data (continued)

Simulation

Graphic

The graphic simulation can be used for controlling optically the program run. Thereby e.g. the axes and the transmission of the M-functions can be switched off for a test run. The The graphic simulation is started in the start menue with ‘1’ and switched off with ‘0’ or ‘deleted’.

With the key you can switch from graphic to normal picture. If the graphic simulation is not switched on in the start menue, the message M4402: ‘Graphic simulation not active’ appears if you press the key .

Axes

If the simulation of the axes is activated (1), the axes do not move during program run. But the axes movements can be seen on the screen. ‘ASIM’ is written in the headline. The function can be switched off with ‘0’ or ‘deleted’.

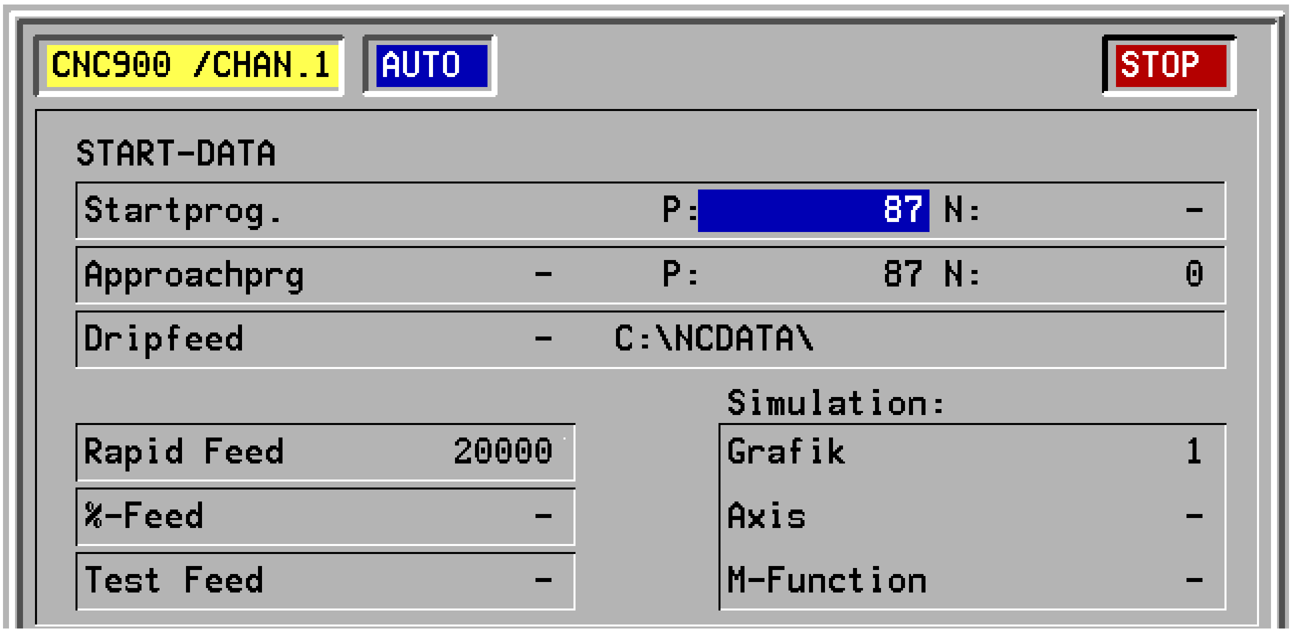
M-functions

If the simulation of the M-functions is activated (1), the M-functions are not transmitted to the PLC. ‘MSIM’ is written in the headline. The function can be switched off with ‘0’ or ‘deleted’.

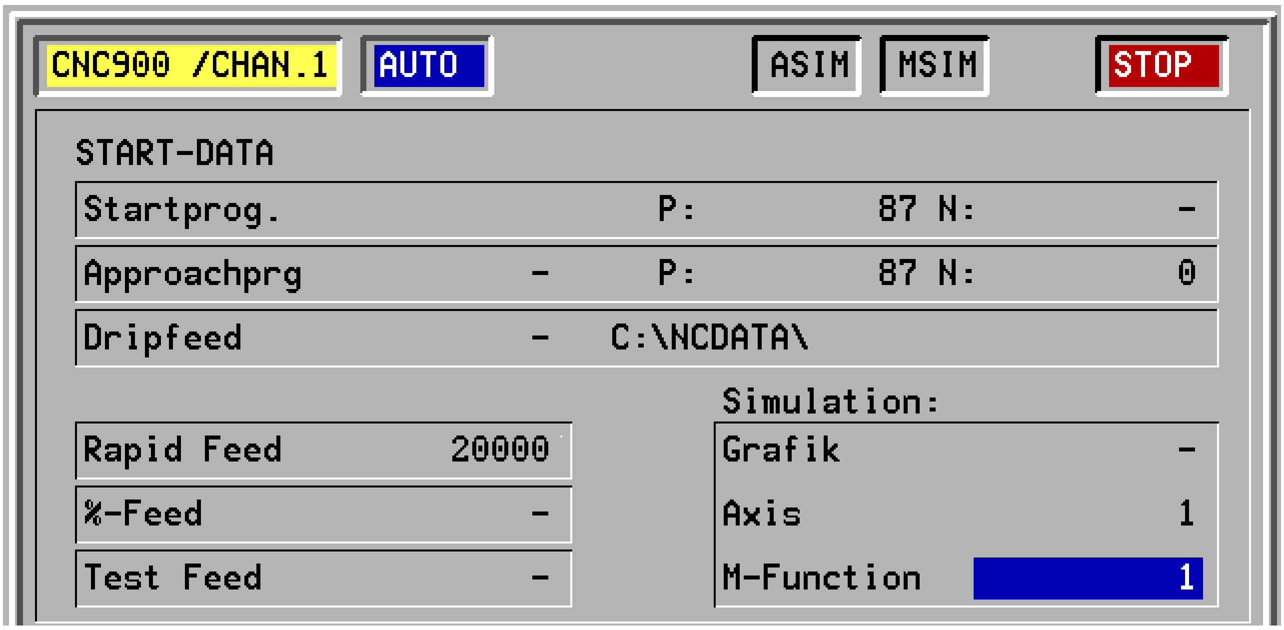
2.5.1 Start data (continued)

Simulation

Graphic on



Axes and M-functions on



2.5.1 Start data (continued)

Graphic simulation

Data input (F2)		Input with numerical keys
X, Y and Z		X, Y, and Z origin
Size		Zoom
View	0	XY-coordinates (G17)
	1	XZ-coordinates (G18)
	2	YZ-coordinates (G19)
	3	3D
A-angle		turning around X-axis
B-angle		turning around Z-axis

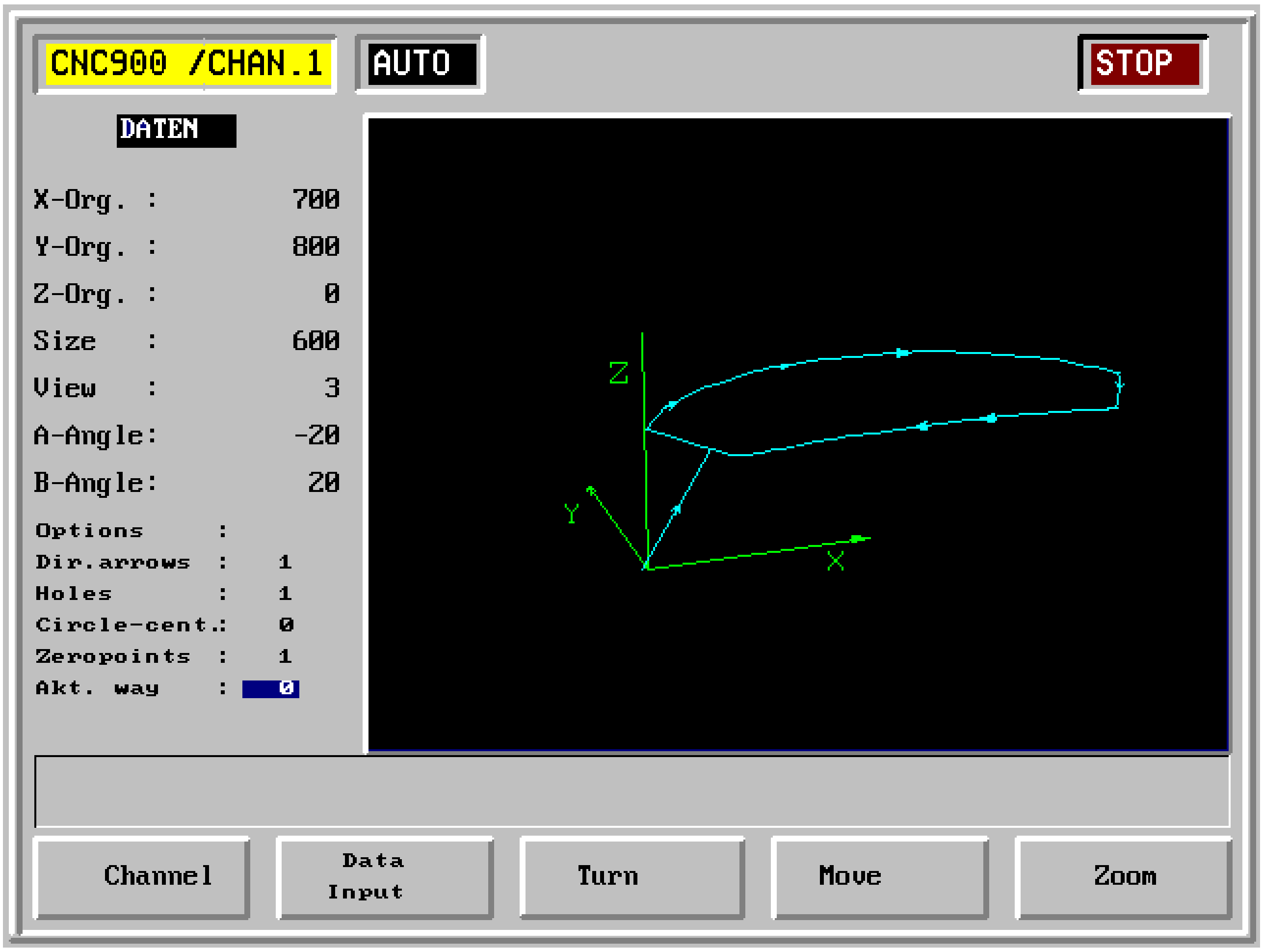
Options

Direcion arrows 1 on, 0 off
Holes 1 on, 0 off
Circle centres 1 on, 0 off
Zero points 1 on, 0 off
Actual ways 1 on, 0 off

2.5.1 Graphic simulation

Graphic simulation

Display of the target way (turned around X and Z axis)

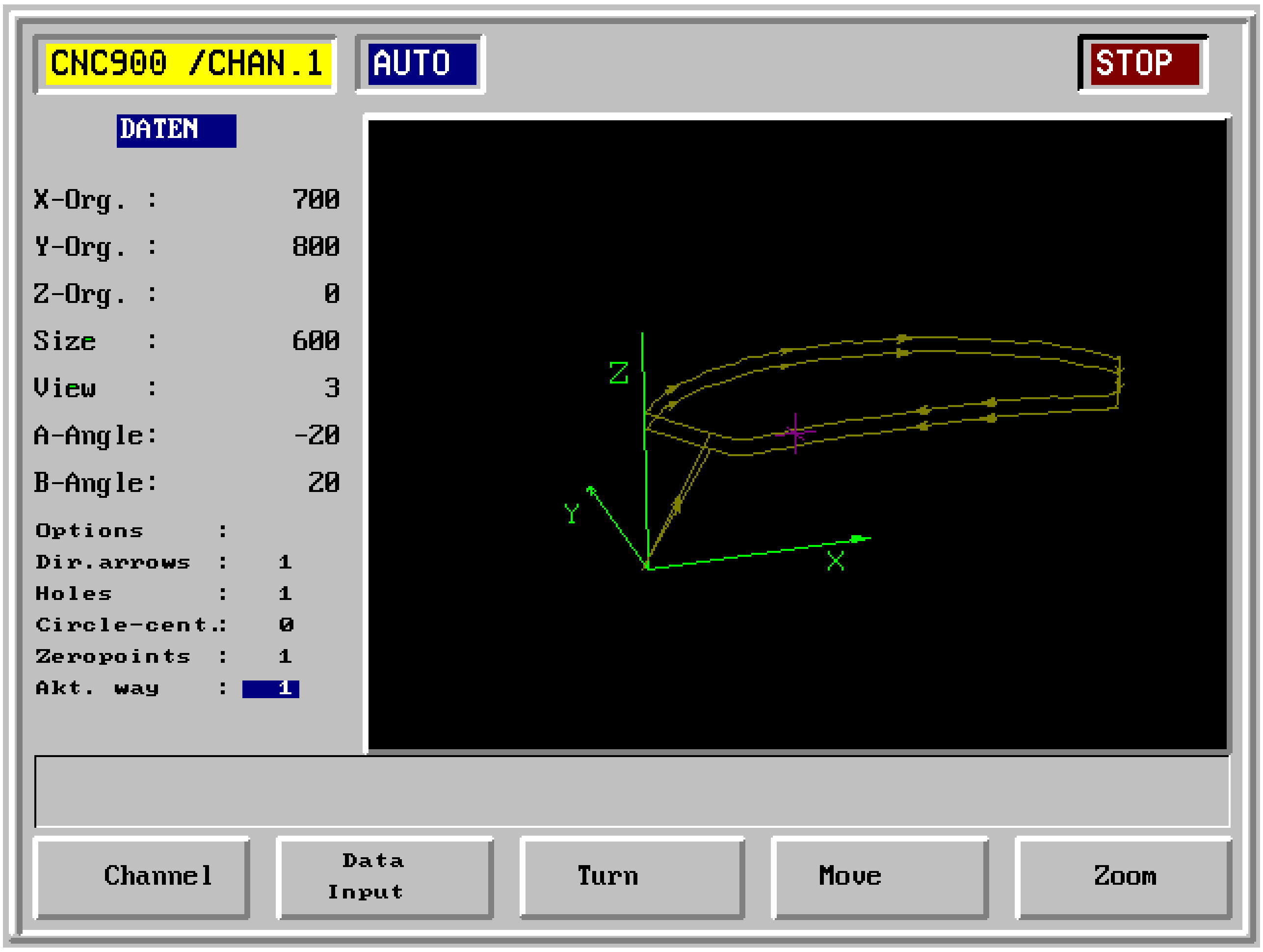


F1	Channel	
F2	Data input	Input with numerical keys
F3	Turning	Turning around the X-, Z-axis, input with cursor
F4	Move	X, Y and Z origin, input with cursor
F5	Zoom	Size, input with cursor

2.5.1 Start data (continued)

Graphic simulation

Display of the target and actual way (turned around X and Z axis)



F1 Channel

F2 Data input Input with numerical keys

F3 Turning Turning around the X-, Z-axis, input with cursor

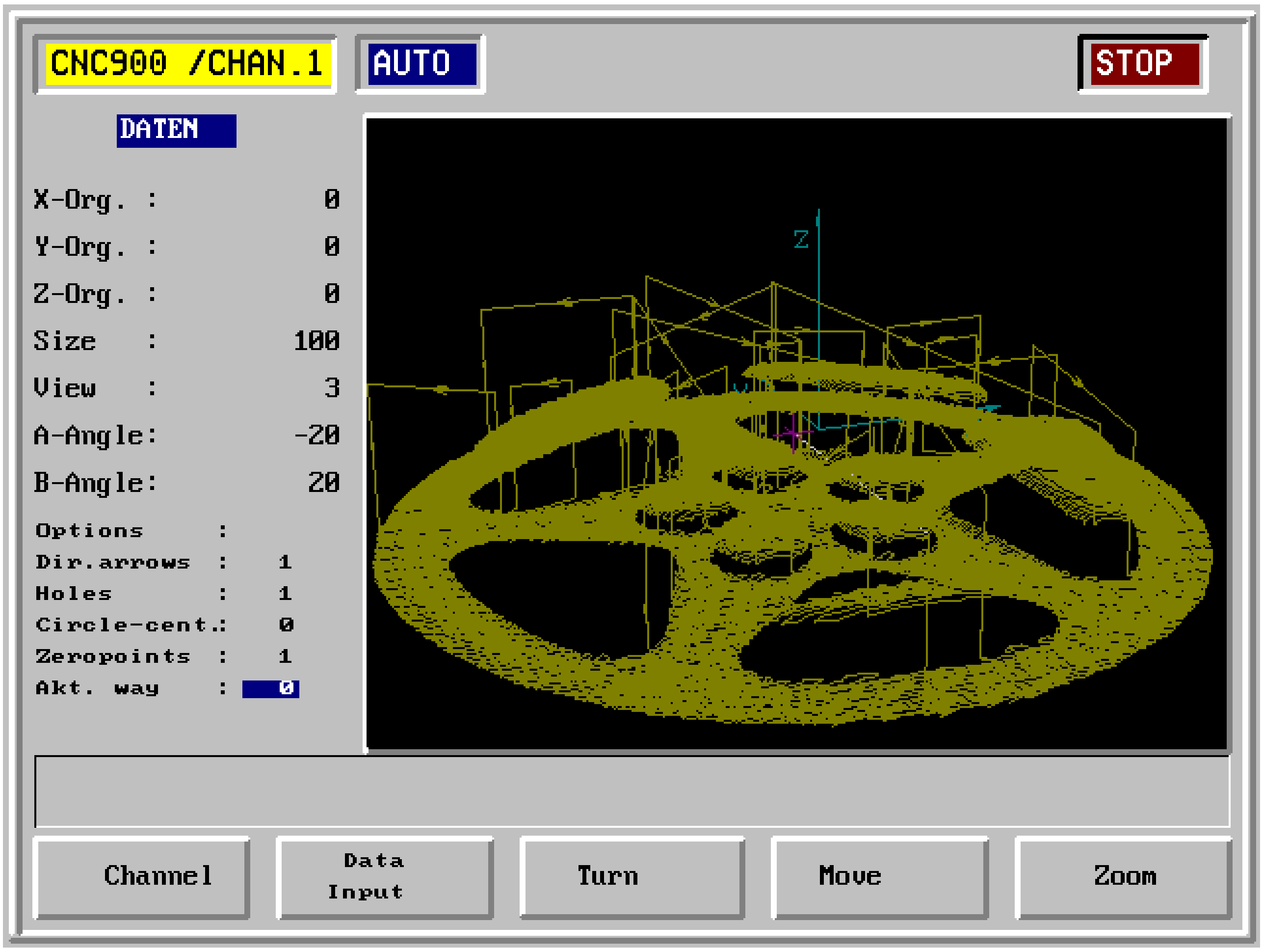
F4 Move X, Y and Z origin, input with cursor

F5 Zoom Size, input with cursor

2.5.1 Start data (continued)

Graphic simulation

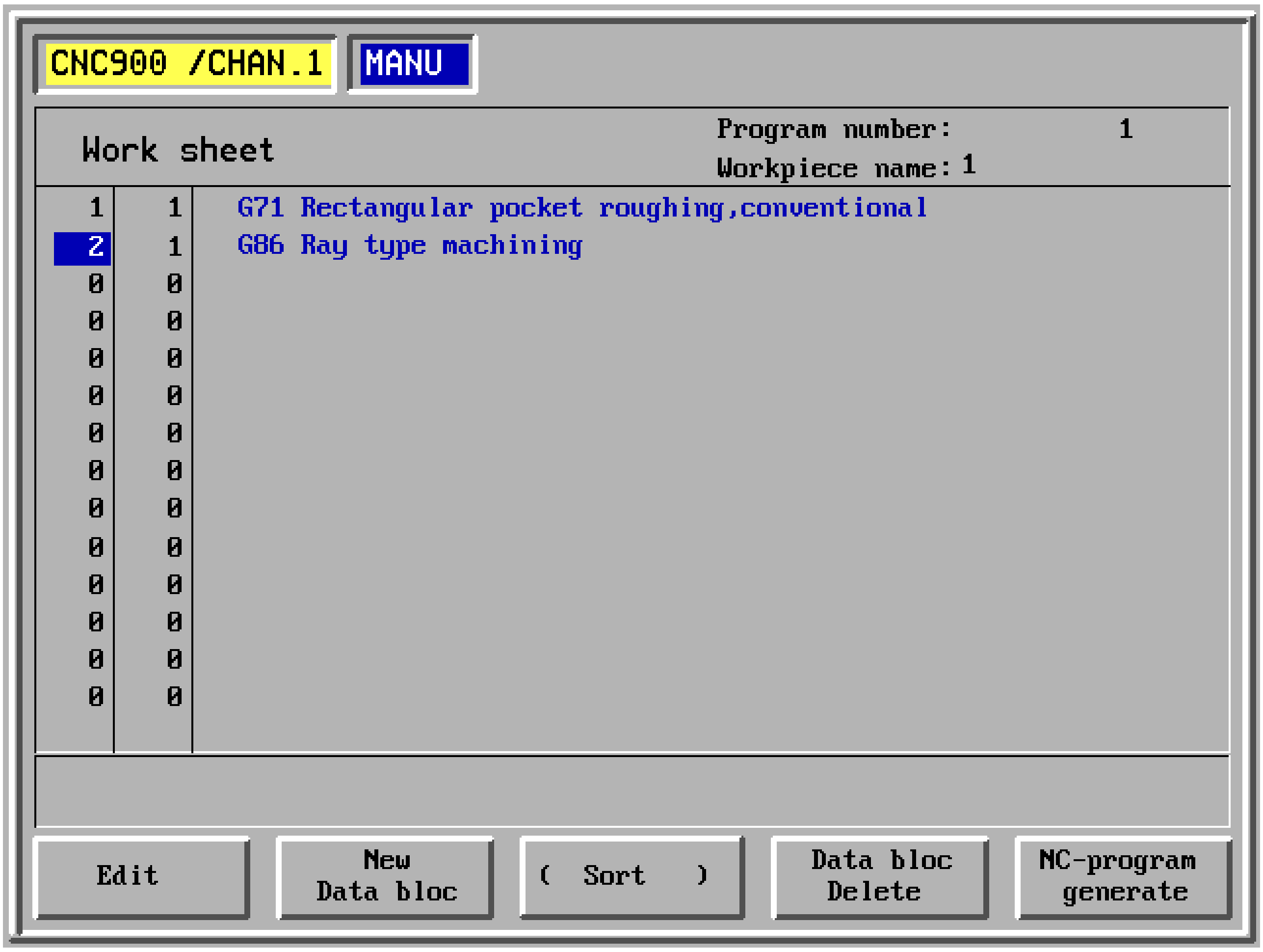
Example of a complex workpiece



F1	Channel	
F2	Data input	Input with numerical keys
F3	Turning	Turning around the X-, Z-axis, input with cursor
F4	Move	X, Y and Z origin, input with cursor
F5	Zoom	Size, input with cursor

2.5.2 Dialogue

Here you get the possibility to switch on a customer-specific dialogue,
e.g. **programming with work sheet**.



F1 Edit

F2 New data block

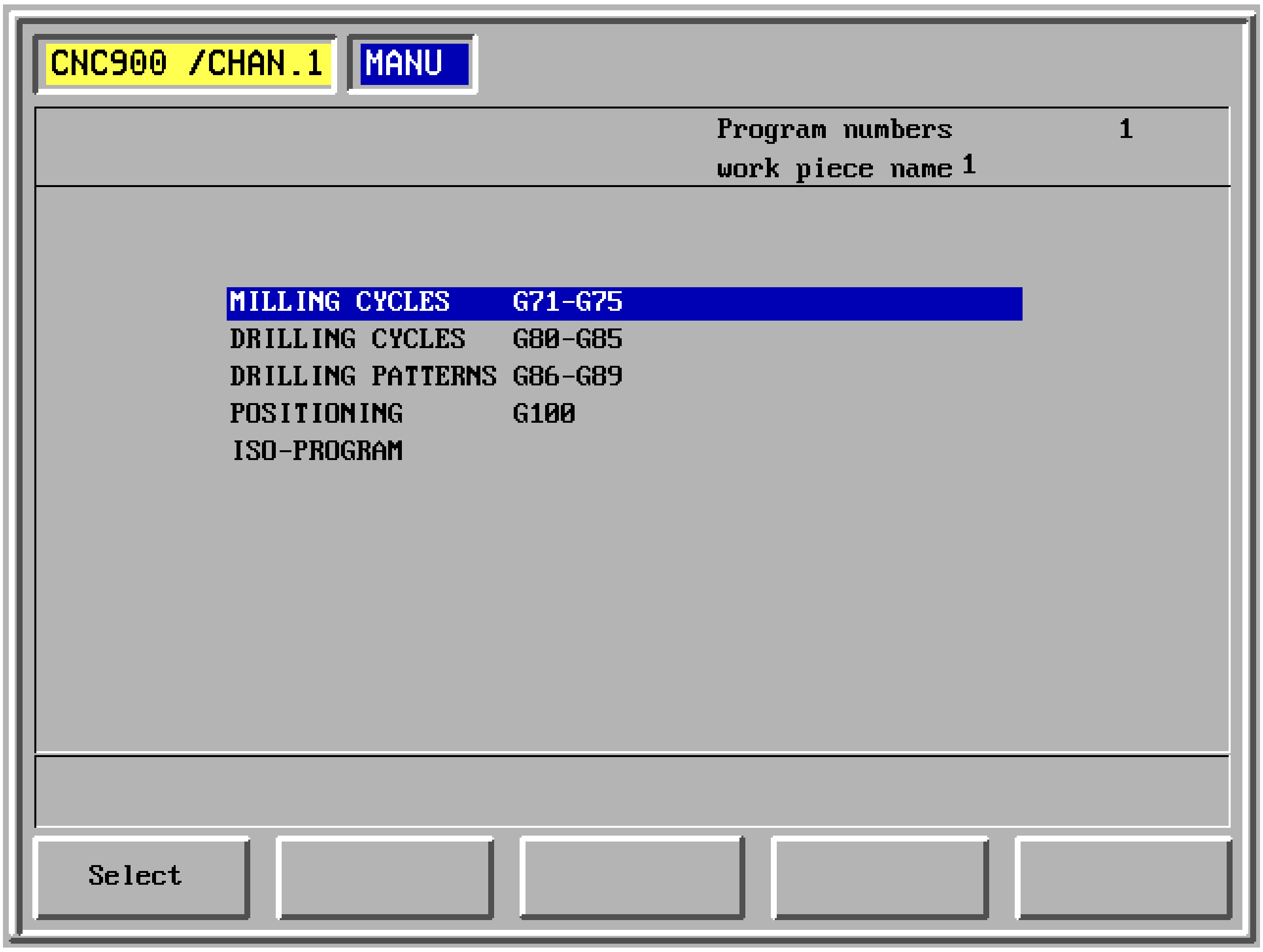
F3 Sort

F4 Delete data block

F5 Generate NC program

2.5.2 Dialogue (continued)

Programming with work-sheet - Selecting the cycle mode



F1 Selecting cycle mode

F2 -

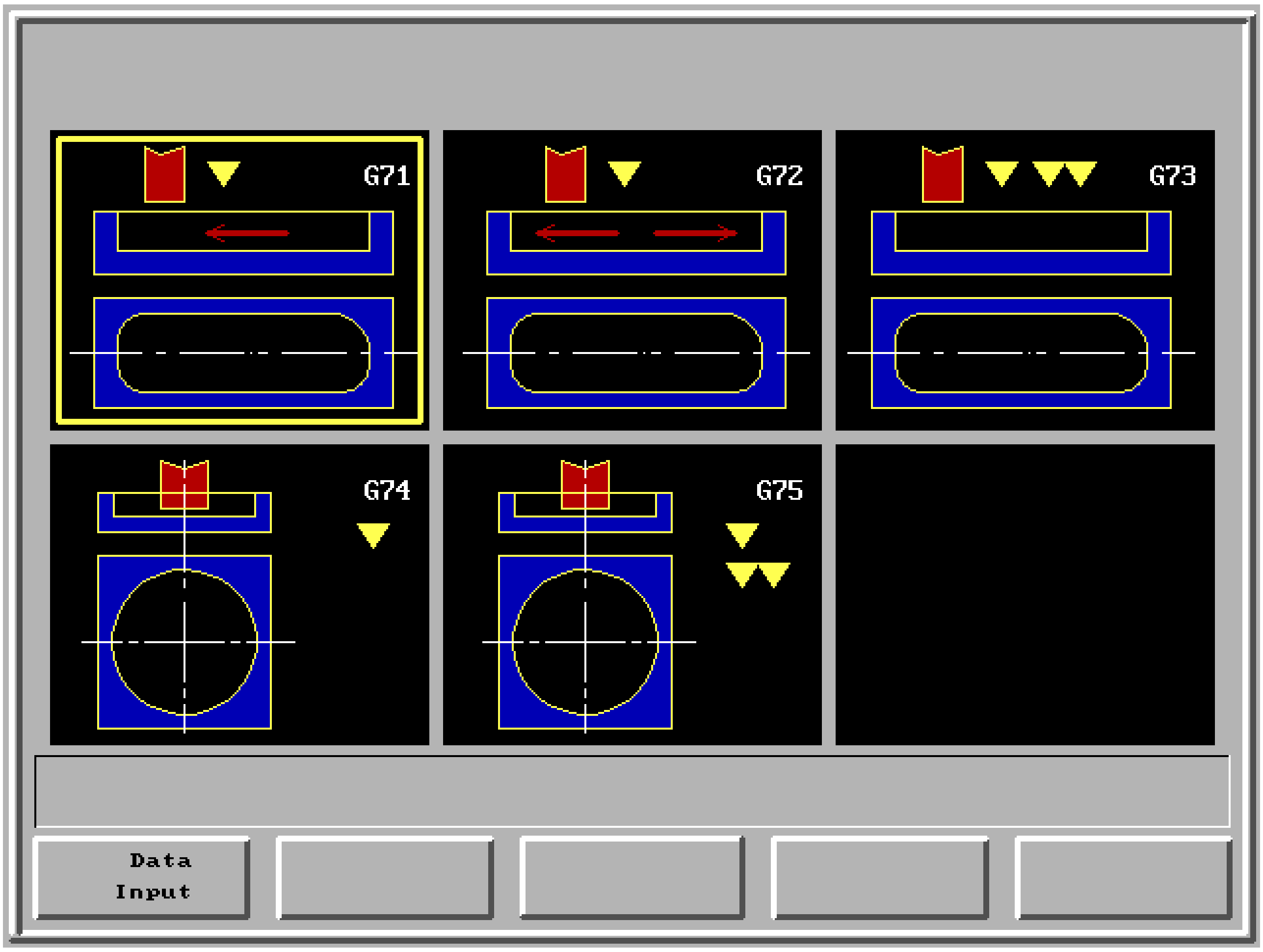
F3 -

F4 -

F5 -

2.5.2 Dialogue (continued)

Programming with work sheet - Selecting the cycle



F1 Data input

F2 -

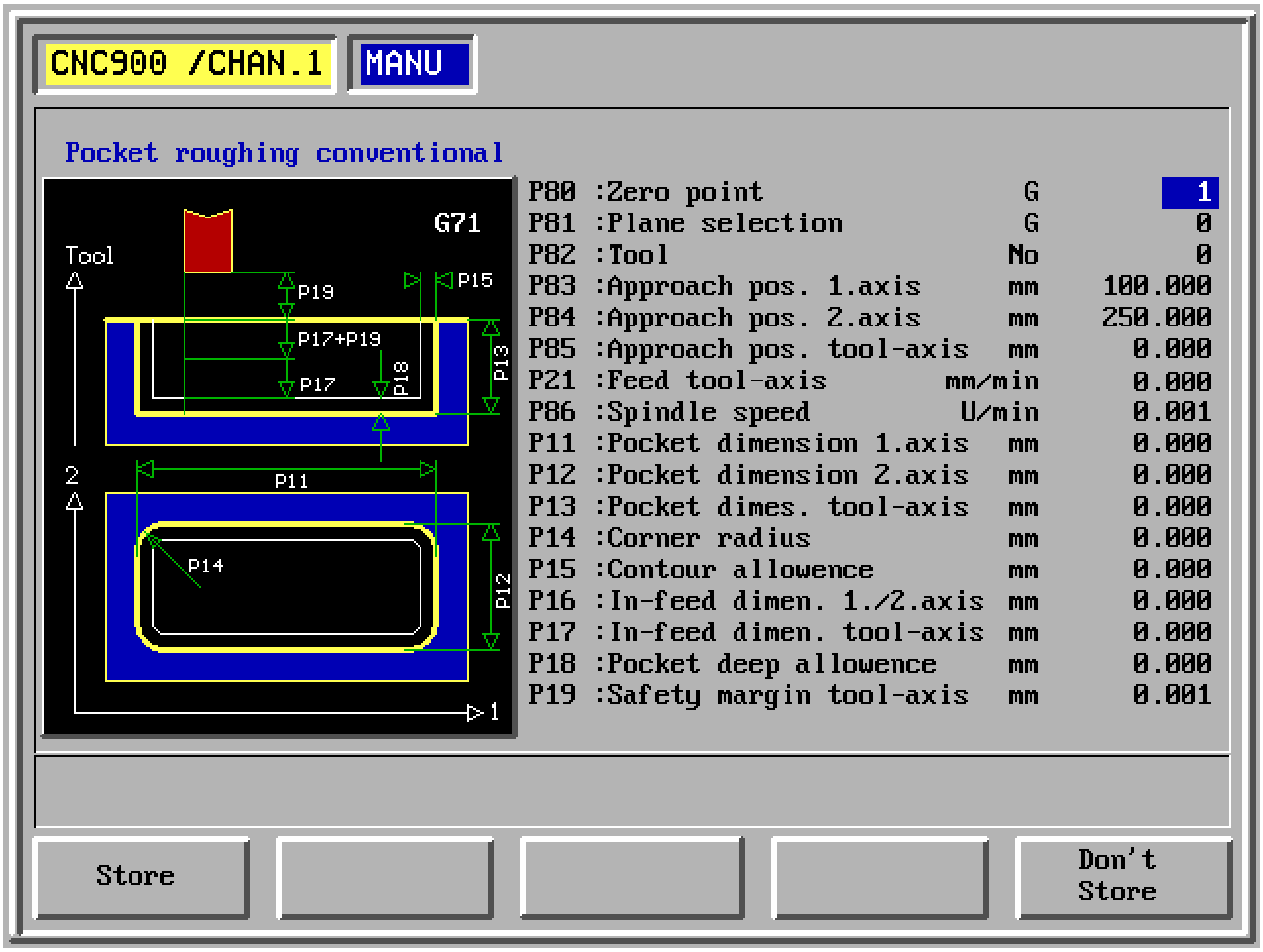
F3 -

F4 -

F5 -

2.5.2 Dialogue (continued)

Programming with work sheet - Selecting the cycle



F1 Store

F2 -

F3 -

F4 -

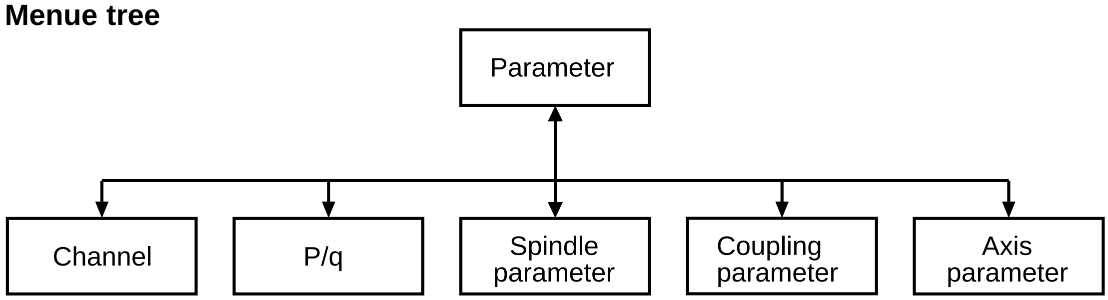
F5 Do not store

2.5.3 Parameter

The parameter menue consists of:

- Channel changeover
- Parameter changeover P/q (Channel - / System parameter)
- Spindle parameter
- Coupling parameter
- Axis parameter

Menue tree



System parameter q

0... 99	General system configuration
100... 999	Definition channel descriptor#
1000... 1999	System settings, system overlapping data
2000... 9999	Axis data
10000...	Channel parameter

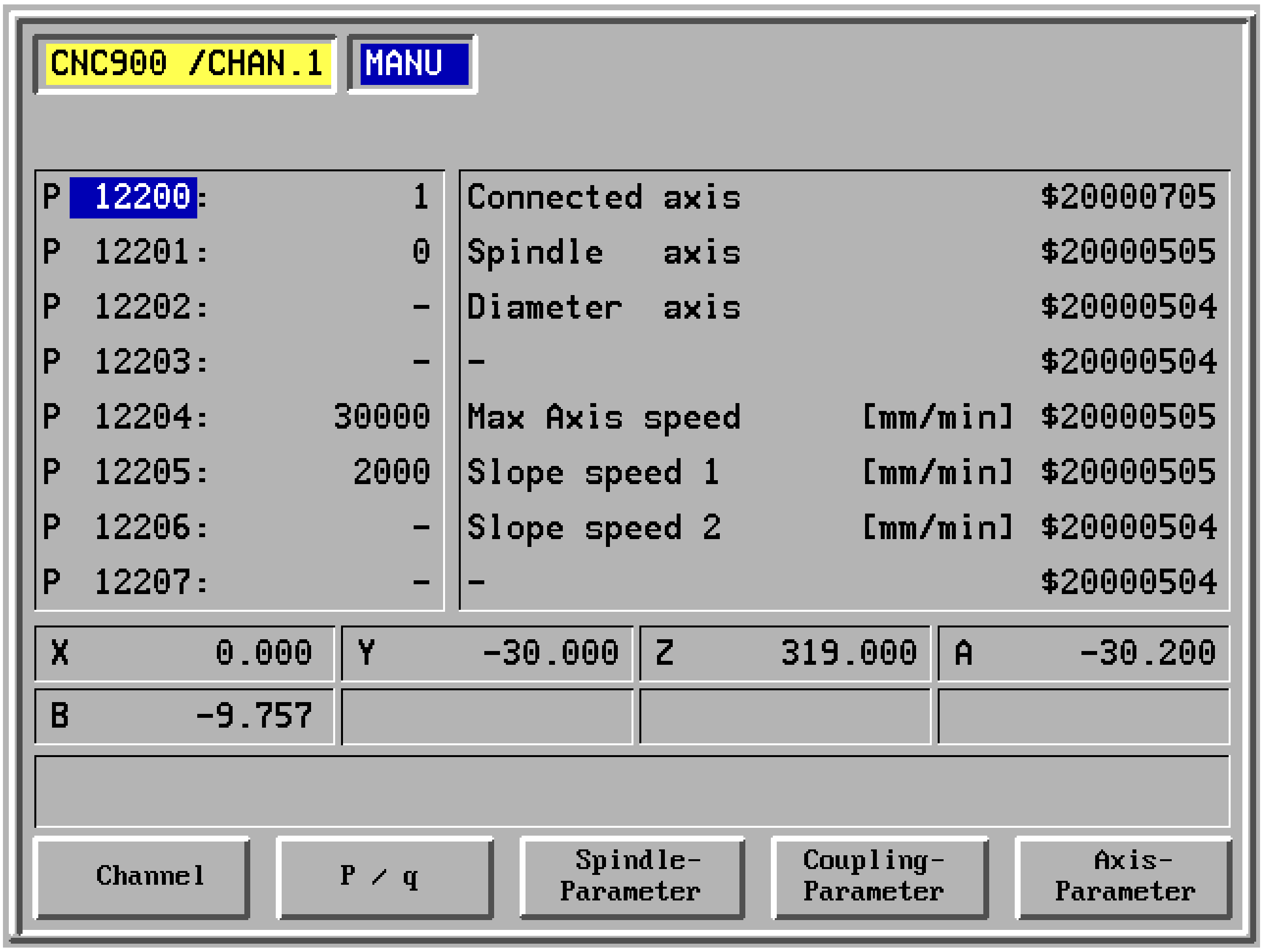
Channel parameter P

0... 6999	User block 1
0... 499	Reserved for BWO standard cycles
0... 299	Cycle-area for cycle interfaces
300... 399	Area reserved for cycles, area for static, modal data
400... 499	Area reserved for cycles, area for temporary data
500... 6999	Area free for the user
7000... 9999	Fix defined channel parameters
11000...11999	System overlapping data, common area of all channels
12000...18399	Axis data
20000...29999	User block 2
30000...39999	User block 3

2.5.3 Parameter

Channel parameter P:

Input addresses, displaying parameters, input values and store.

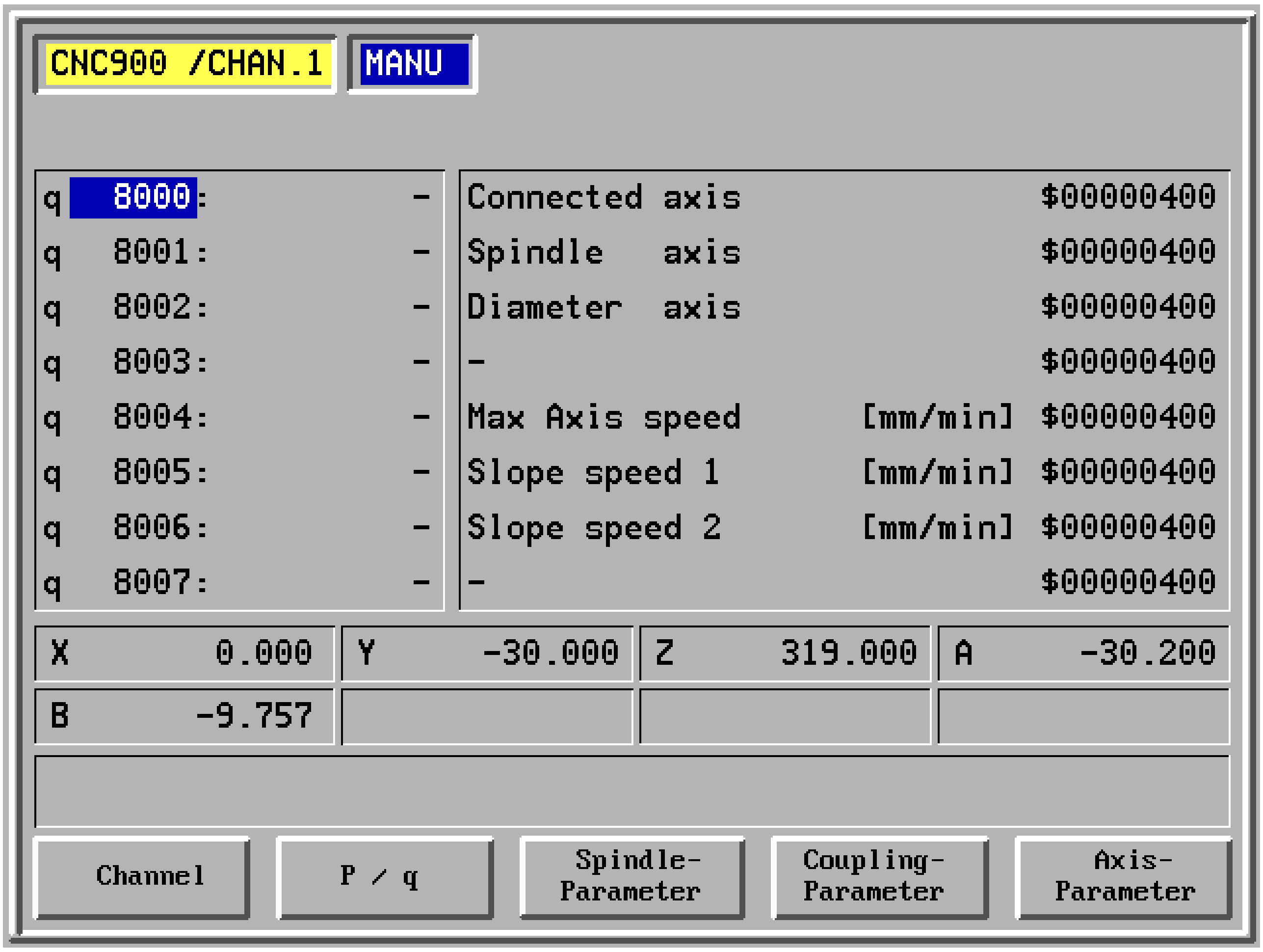


F1	Channel	
F2	P/q	Switching between channel (P) and system parameters (q)
F3	Spindle parameter	
F4	Coupling parameter	
F5	Axis parameter	

2.5.3 Parameter (continued)

System parameter q:

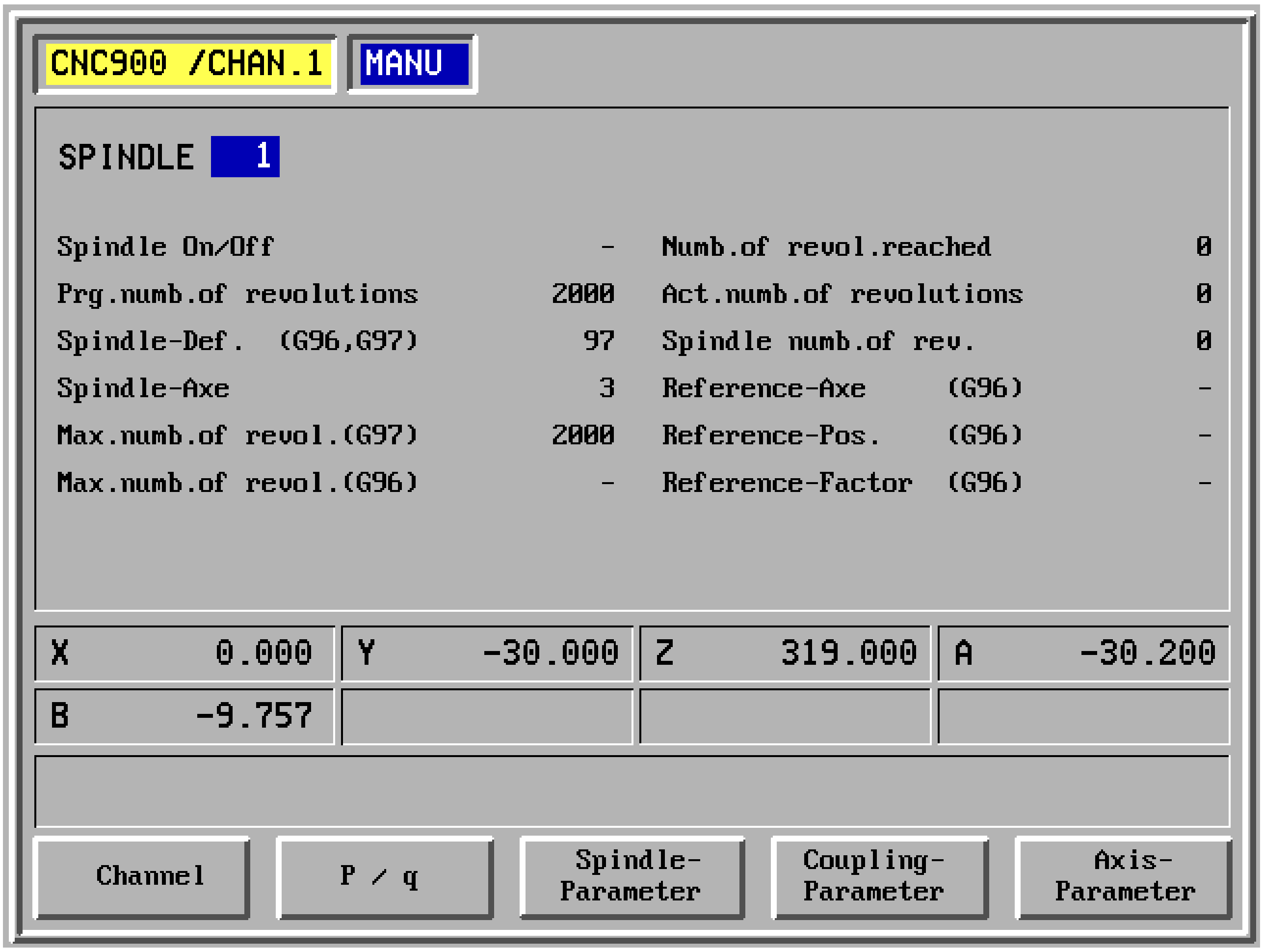
Input addresses, displaying parameters, input values and store.



F1	Channel	
F2	P/q	Switching between channel (P) and system parameters (q)
F3	Spindle parameter	
F4	Coupling parameter	
F5	Axis parameter	

2.5.3 Spindle parameter

Inputing spindle data is possible in this menue.



F1 Channel

F2 P/q Switching between channel (P) and system parameters (q)

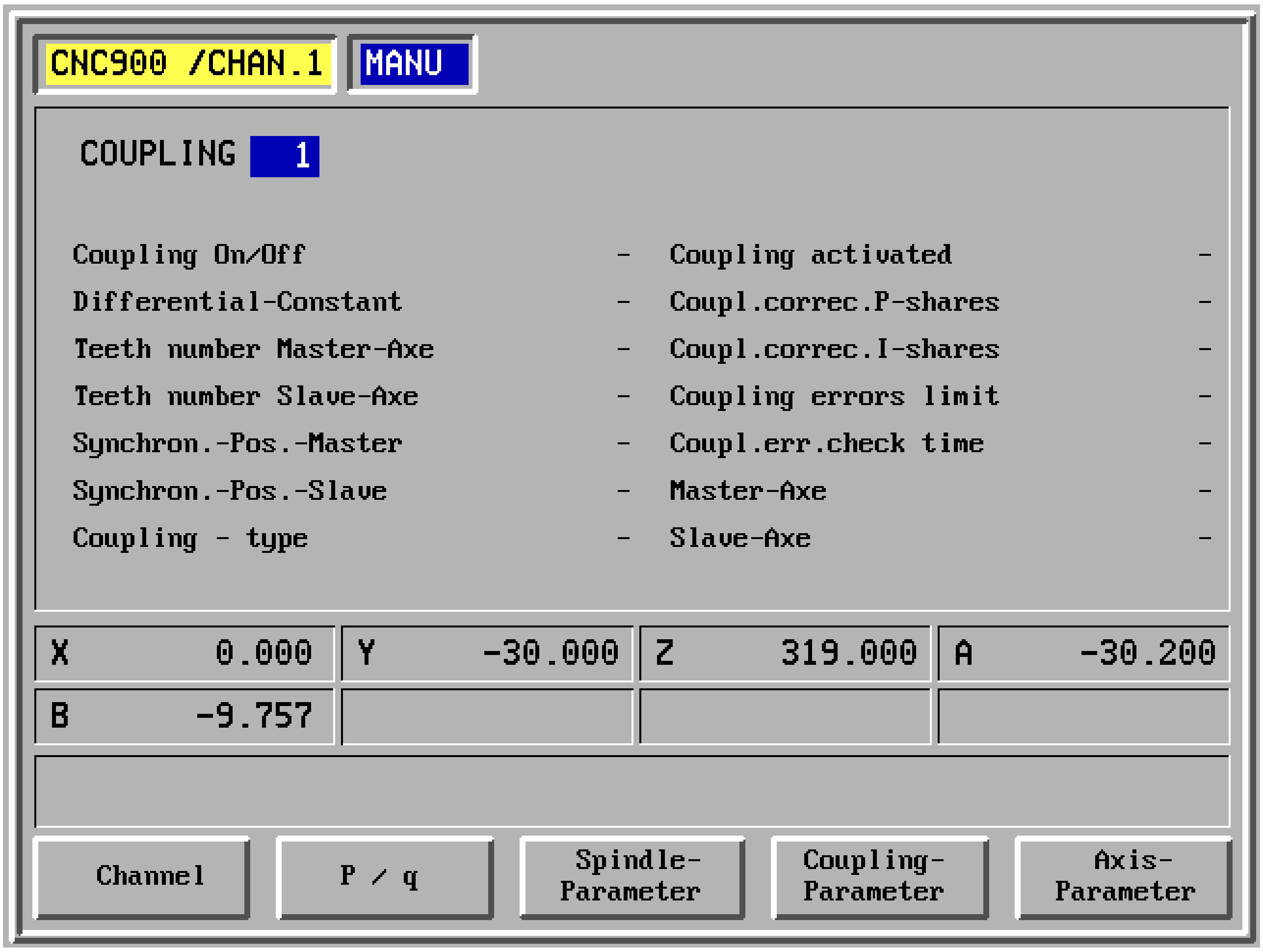
F3 Spindle parameter

F4 Coupling parameter

F5 Axis parameter

2.5.3 Coupling parameter

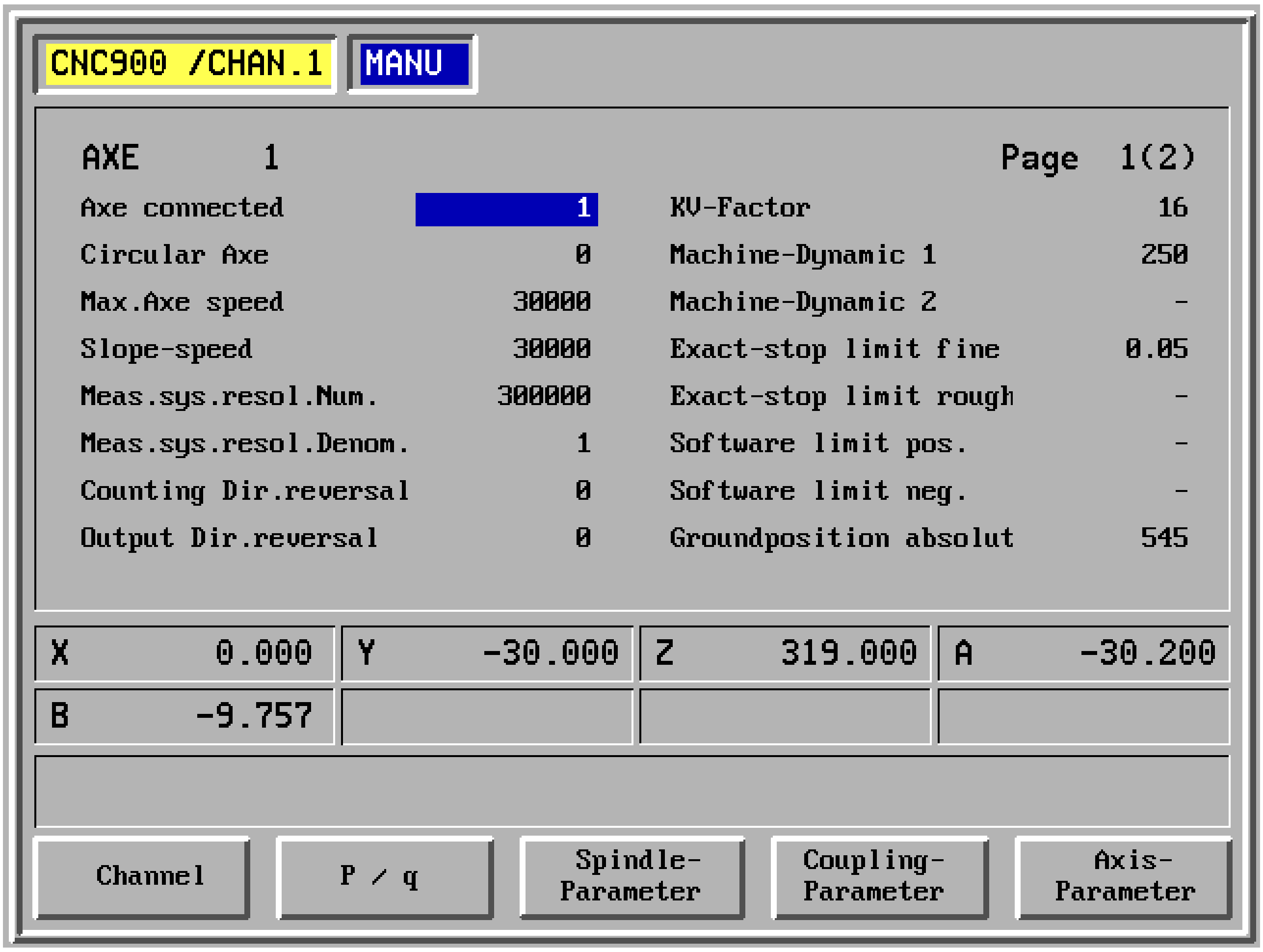
Inputing coupling data is possible in this menue.



F1	Channel	
F2	P/q	Switching between channel (P) and system parameters (q)
F3	Spindle parameter	
F4	Coupling parameter	
F5	Axis parameter	

2.5.3 Axis parameter

Inputing axis data is possible in this menue. Page 1



F1 Channel

F2 P/q Switching between channel (P) and system parameters (q)

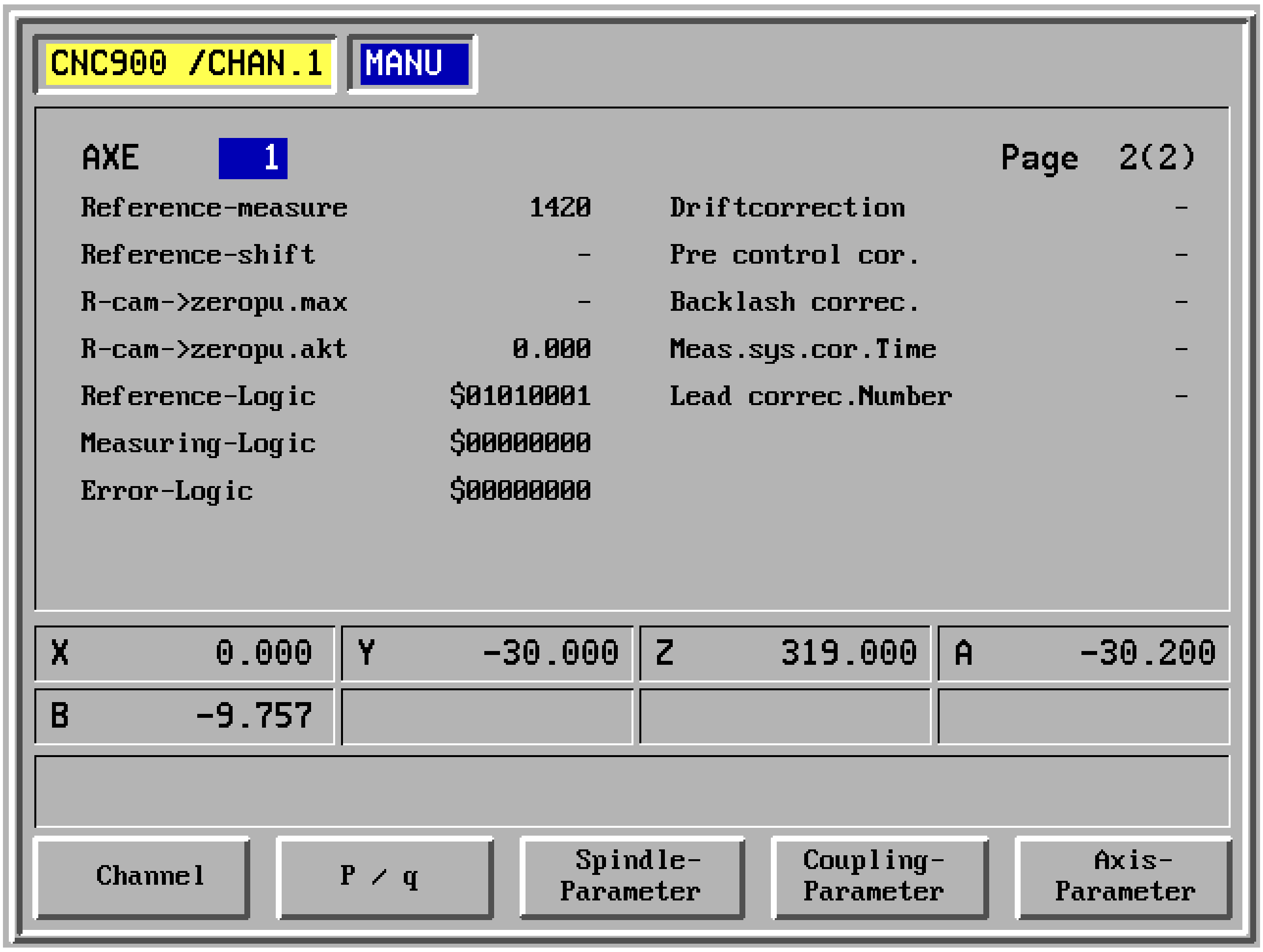
F3 Spindle parameter

F4 Coupling parameter

F5 Axis parameter

2.5.3 Axis parameter (continued)

Inputing axis data is possible in this menue. Page 2

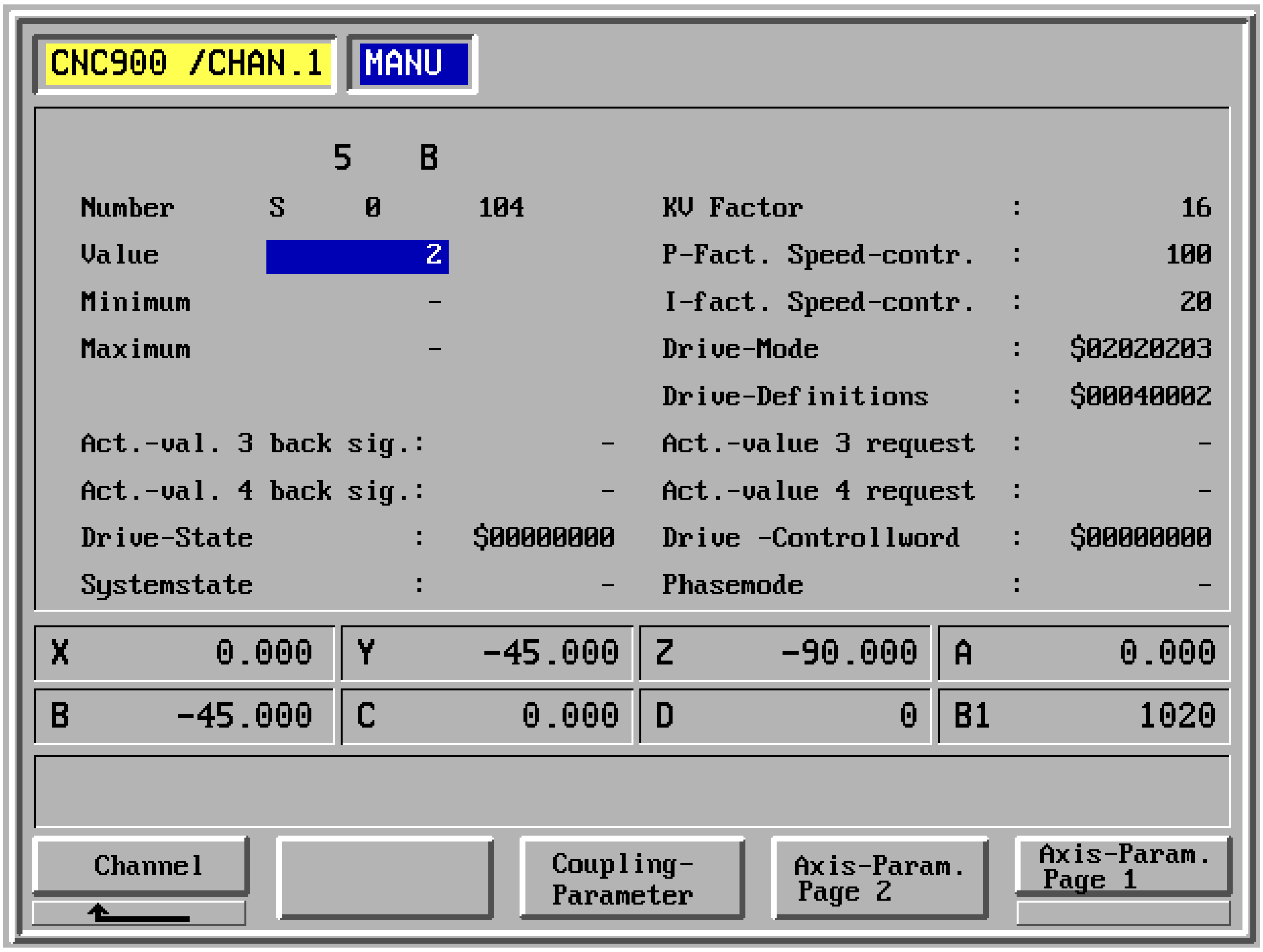


F1	Channel	
F2	P/q	Switching between channel (P) and system parameters (q)
F3	Spindle parameter	
F4	Coupling parameter	
F5	Axis parameter	

2.5.3 Axis parameter (continued)

Drive parameter

This menu permits the Input of the drive data.



F1 Channel

F2 -

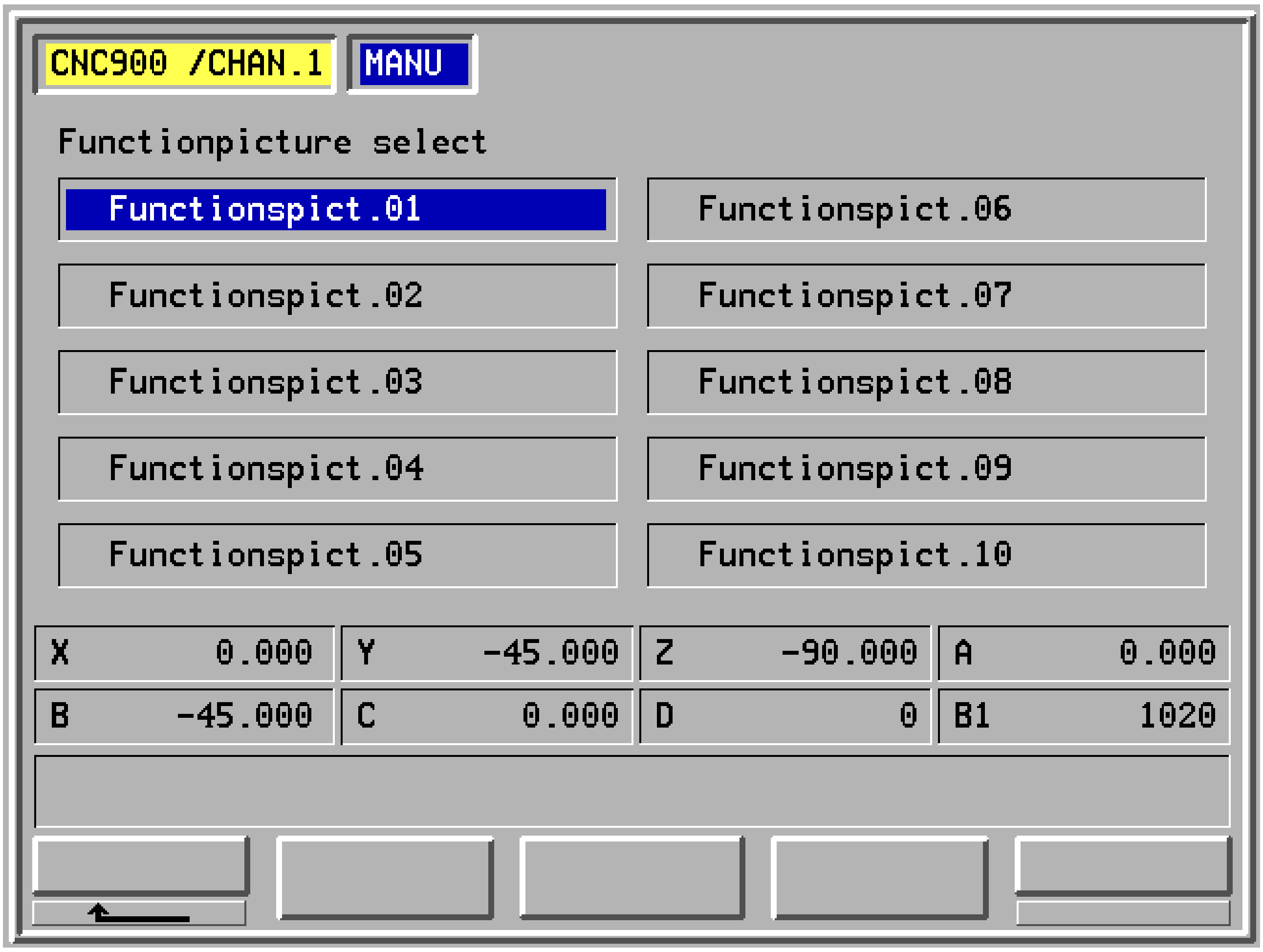
F3 Drive parameter

F4 Axis parameter page 2

F5 Axis parameter page 1

2.5.4 Machine functions

The actual function picture appears.



F1 -

F2 -

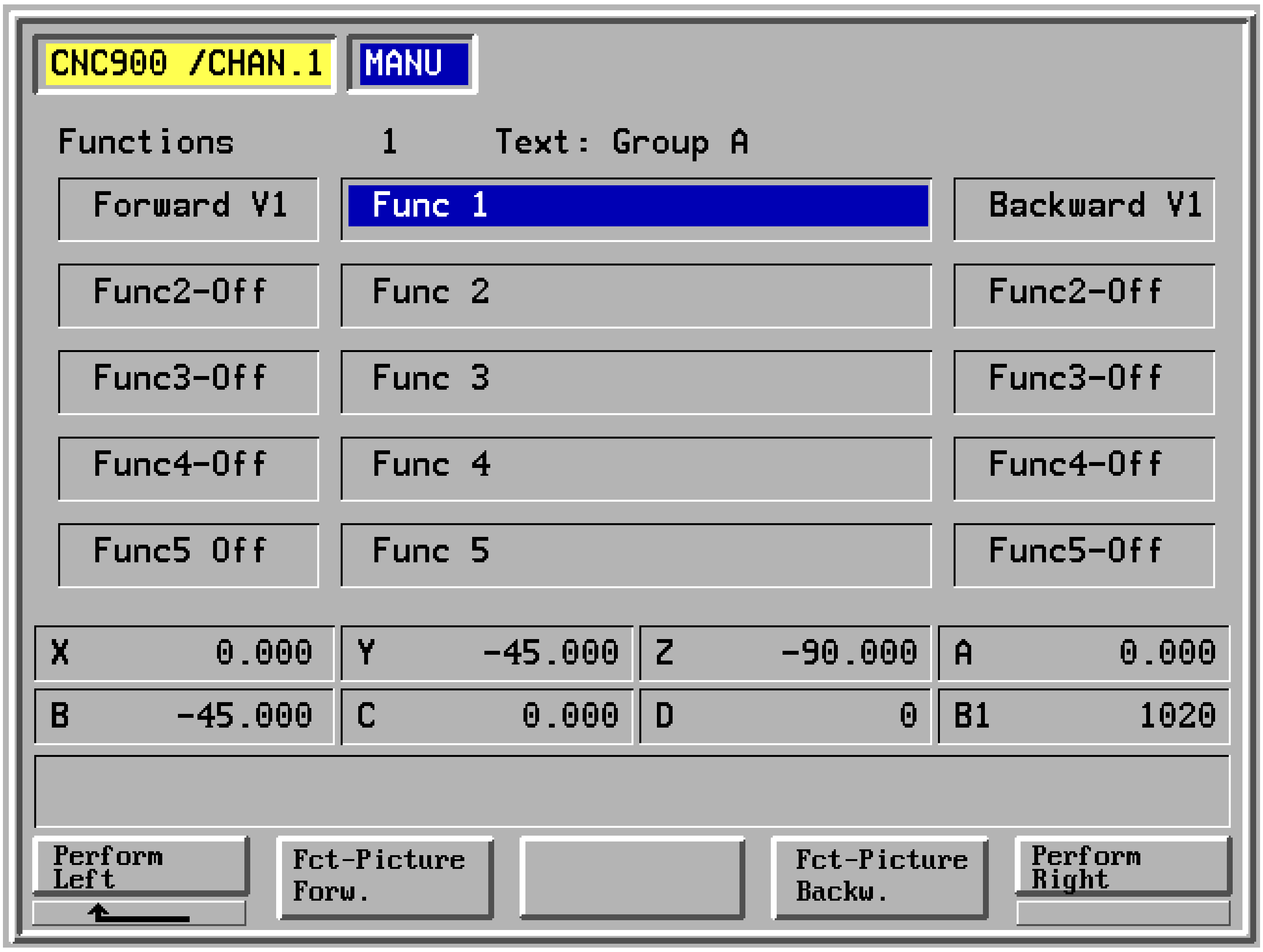
F3 -

F4 -

F5 -

2.5.4 Machine functions (continued)

The actual function picture appears.



F1 Performance left

F2 Function picture forward

F3 -

F4 Function picture backward

F5 Performance right

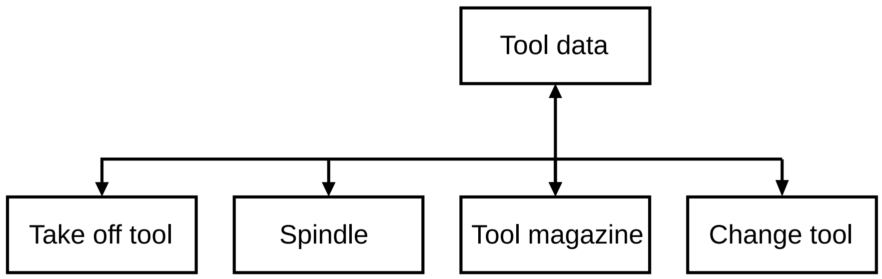
2.5.5 Tool data

The actual tool data can be input here.

The tool data menue consists of

- Lay down tool
- Spindle
- Magazine
- Tool change

Menue tree

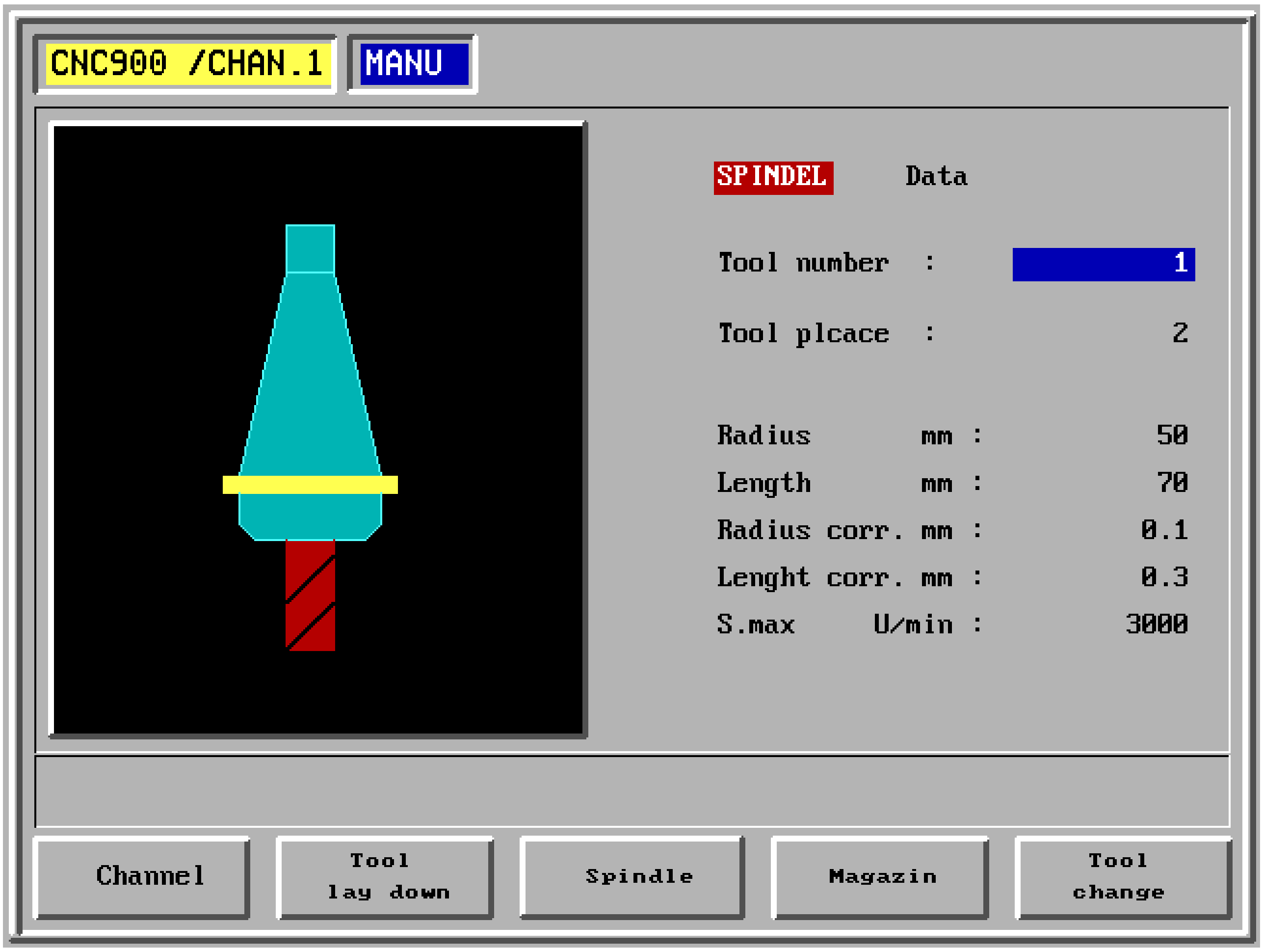


The tool dates (length, radius...) can be called up with 10-digit tool numbers in the program run.

Inputs are possible in manual mode via the operating panel or external data carriers without indicating a block number. If ‘delete store’ is input, the tool data are deleted during power on test.

2.5.5 Spindle

Tool in the spindle.



F1 Channel

F2 Lay down tool

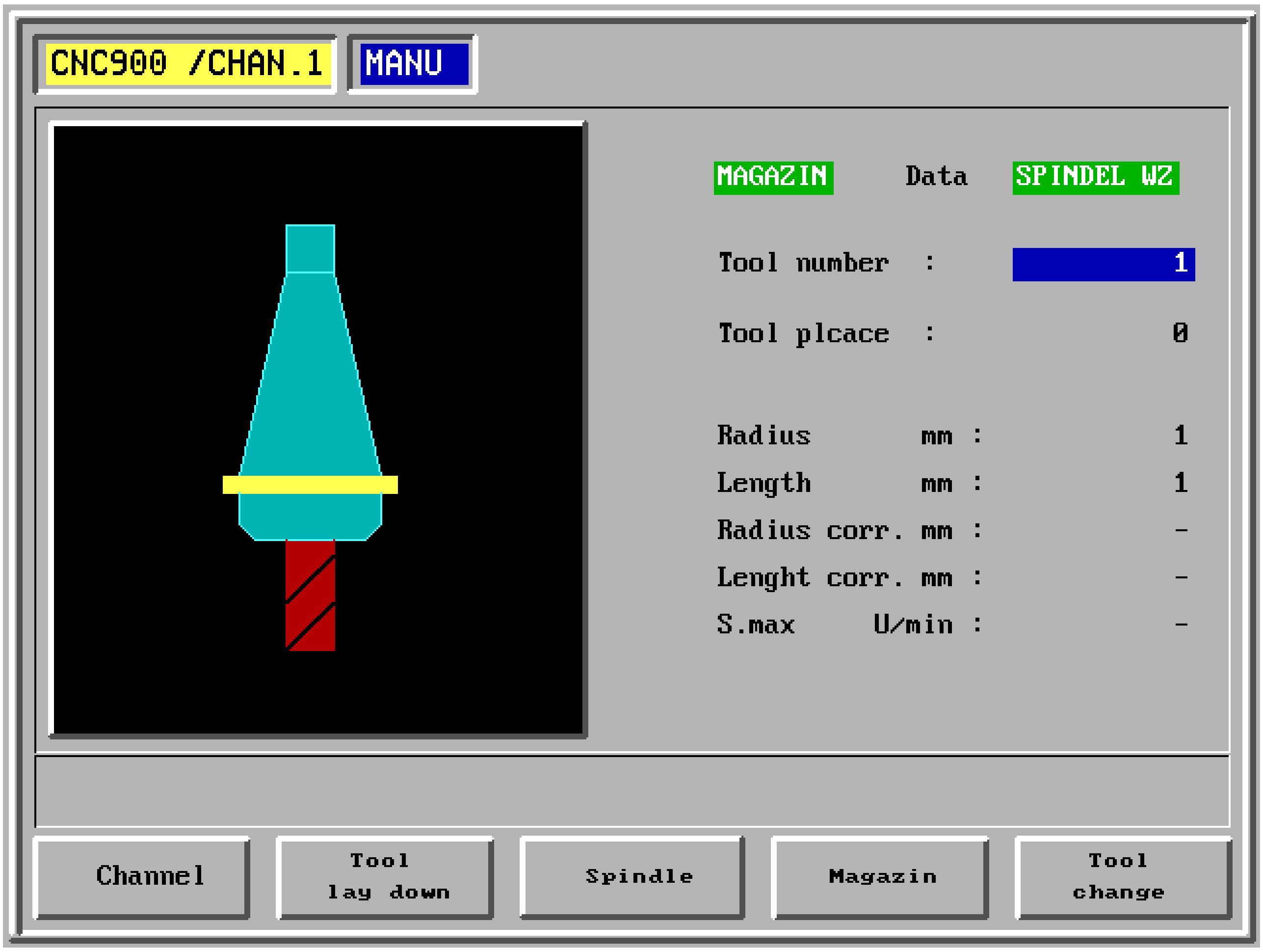
F3 Spindle

F4 Magazine

F5 Tool change

2.5.5 Magazine

Tool in the magazine.



F1 Channel

F2 Lay down tool

F3 Spindle

F4 Magazine

F5 Tool change

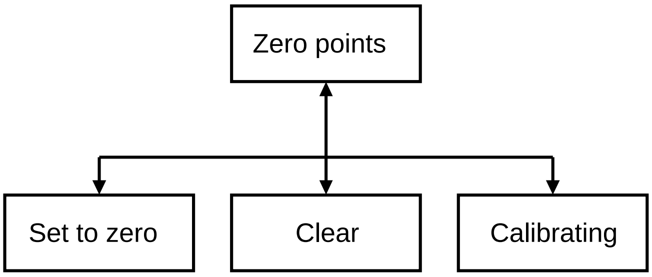
2.5.6 Zero points

The actual zero point data can be input here.

The zero point data menue consists of

- Set to zero
- Delete
- Calibrate

Menue tree



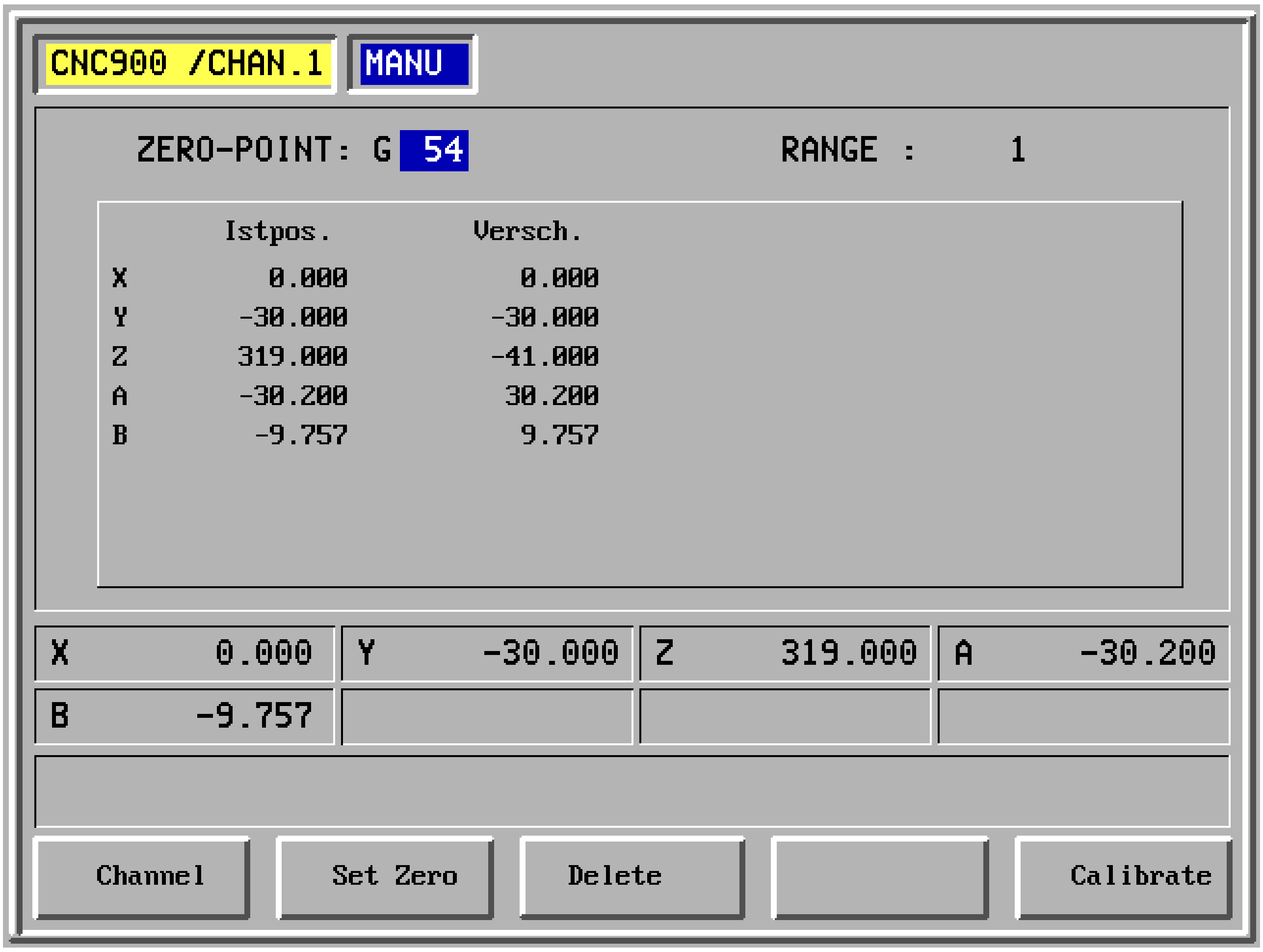
For each axis, a maximum of 6 zero points can be set with G54 to G59 and can be called up during program run.

Zero point shift is made ineffective with G53. The program then refers to the machine zero point.

Inputs are possible in manual mode via the operating panel or external data carriers without indicating a block number. When deleting the memory during power on test, the zero point data are alos deleted.
Zero point data can be loacated mathematically or they can be approached (edge scanner, clock gauge).

2.5.6 Zero points (continued)

The actual zero point is displayed. The values can be input and stored.



F1 Channel

F2 Set to zero

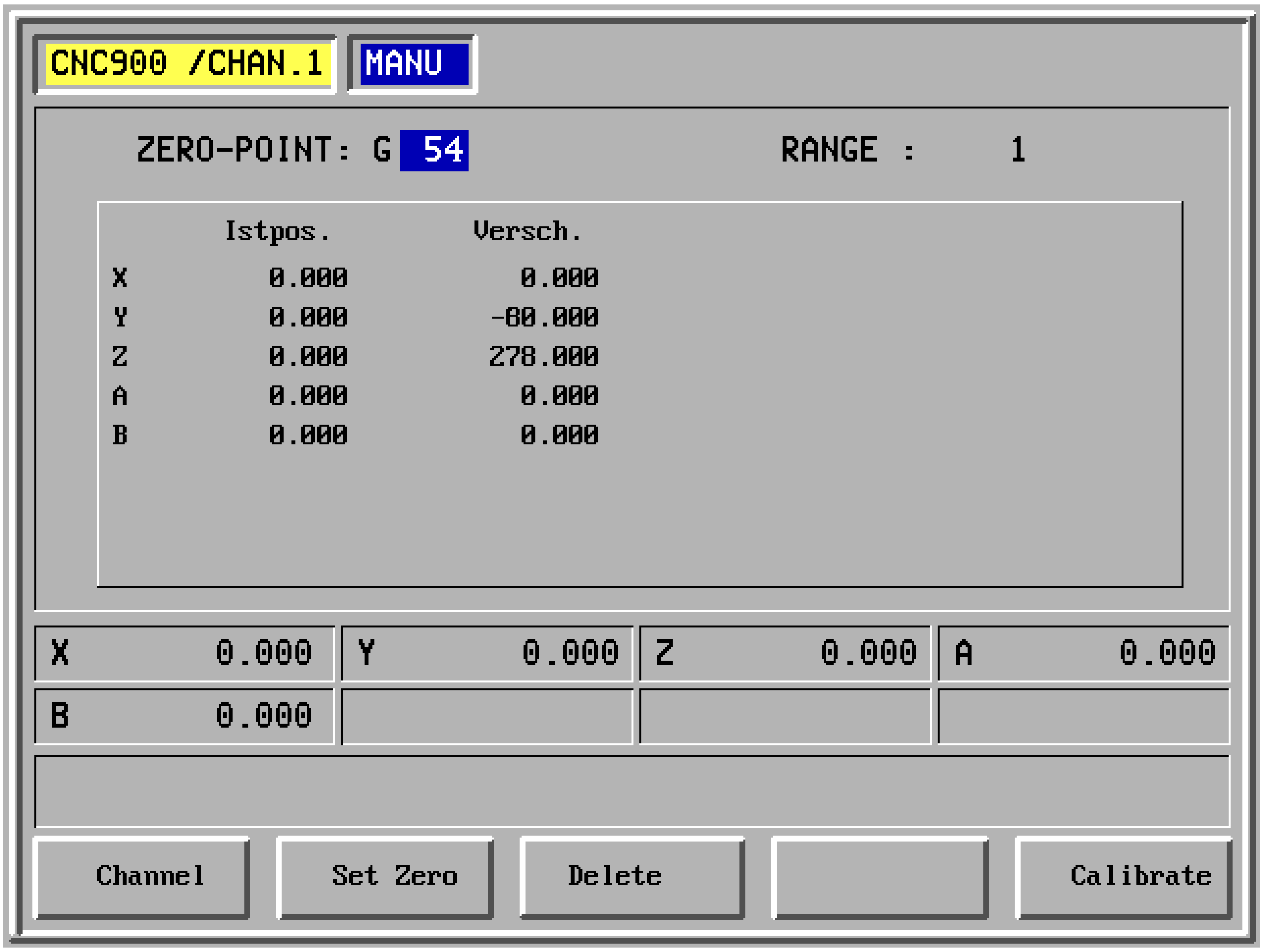
F3 Delete

F4 -

F5 Calibrate

2.5.6 Zero points (continued)

The actual values can be set to zero.



F1 Channel

F2 Set to zero

F3 Delete

F4 -

F5 Calibrate

2.6 Operating modes

The following operating modes can be set:
- MANUAL
- AUTOMATIC Sequential block
  Single block
  Positioning

2.6.1 Manual mode

Manual mode is switched on with key .

For traversing the axes, the axis name (X, Y, Z, ...) and travelling mode (continuously or step by step) and the feed rate have to be input first.

By pressing the keys (Manual+) or (Manual-) the selected axis is traversed in positive or negative direction.

The length of the actual tool is calculated when traversing the tool axis.

Travelling mode

During (continuously) travelling mode, the axis is moving as long until the manual key is pressed.
When traversing with fix length, the complete length is traversed by pressing a manual key once (the step width can be selected by pressing the key (step) again and again).
During this time, the keys for Manual+/-, axis name and travelling mode remain ineffective.

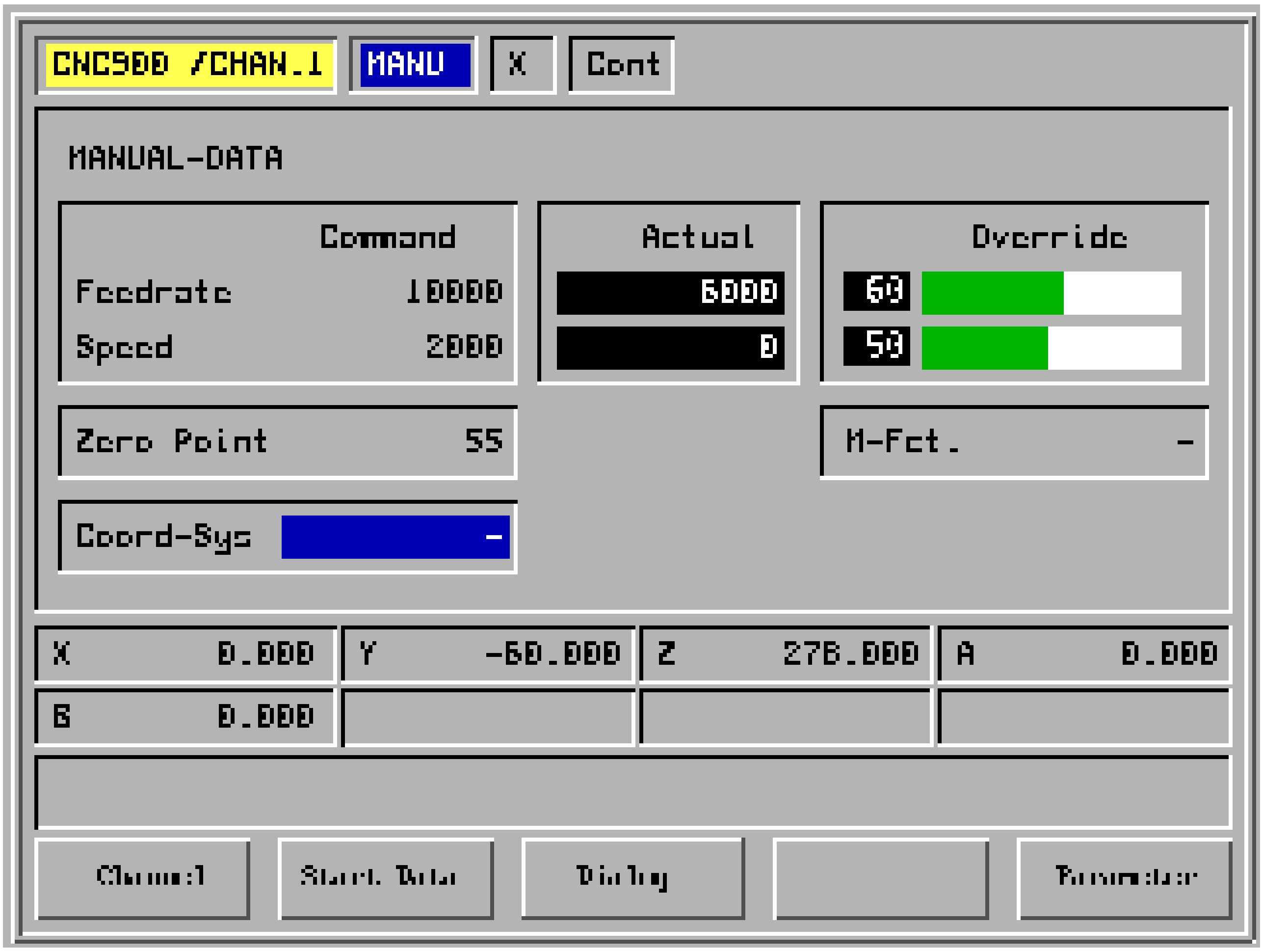
When pressing the key (Stop), the process can be interrupted. If a manual key is pressed again, the axis moves from the new position on with the selected length.

Command = Actual
Desired positions can be approached in the program and they can be overtaken into the command value display with key (Command=Actual) and can be written into the memory with (Input).

2.6.1 Manual mode (continued)

Traverse **continuously in manual mode**.

Switch on manual mode with key . Select continuously with key .



F1 Channel

F2 Start data

F3 Dialogue

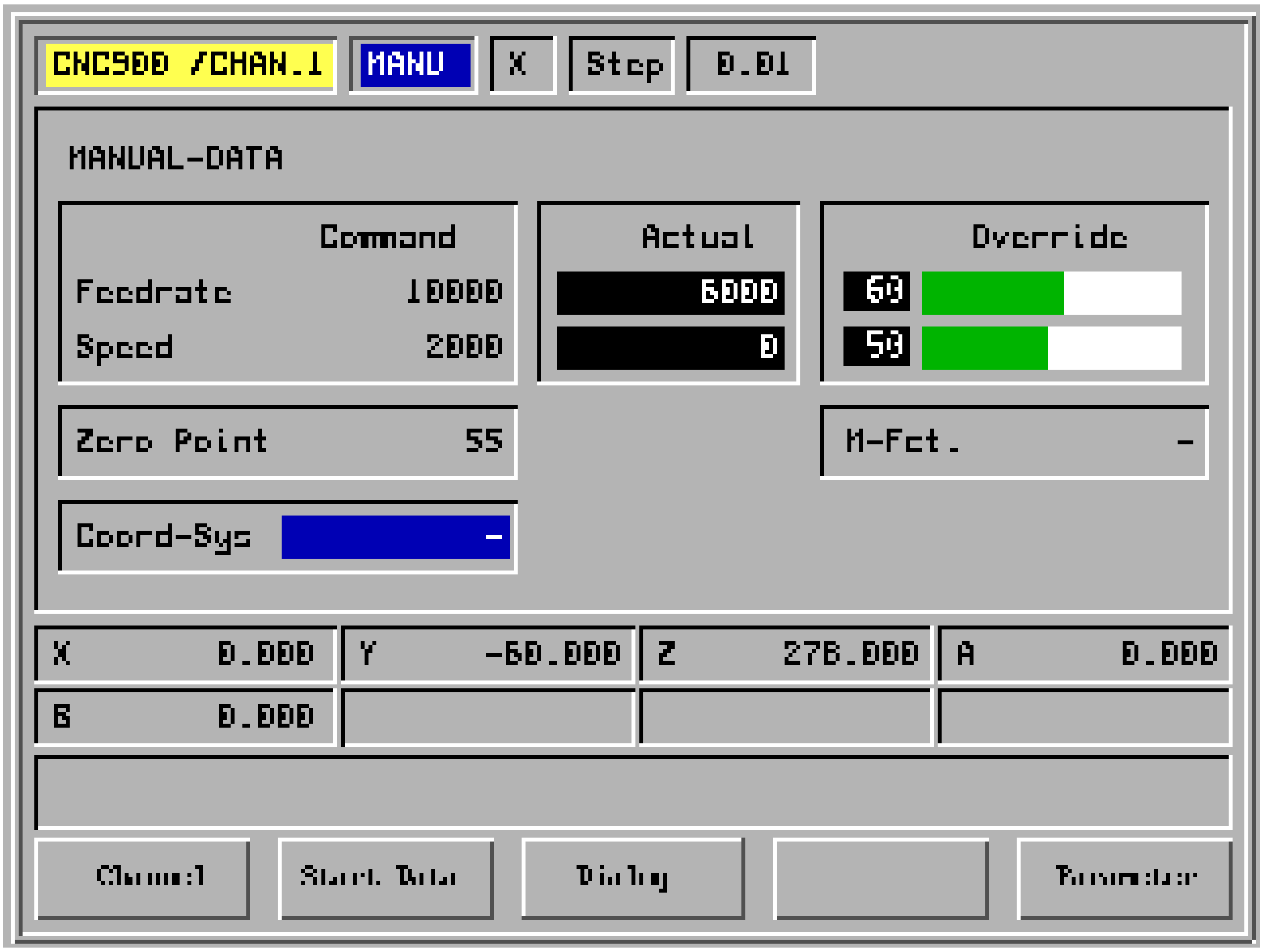
F4 -

F5 Parameter

2.6.1 Manual mode (continued)

Traverse **step by step in manual mode**.

Switch on manual mode with key . Select step-by-step with key .



F1 Channel

F2 Start data

F3 Dialogue

F4 -

F5 Parameter

2.6.2 Automatic mode

Automatic mode is switched on with the keys:

Automatic sequential block

Automatic single block

Positioning

Start/Stop

The selected program is started with key (Start). It can be interrupted with key (Stop)

and continued with key (Start) at any time.

The keys for machine functions are dependant of the PLC. The speed can be controlled with the override.

Skip block

When making a program, the blocks which do have a slash placed in front of them, are skiped in automatic mode if the key / (skip block) is switched on.

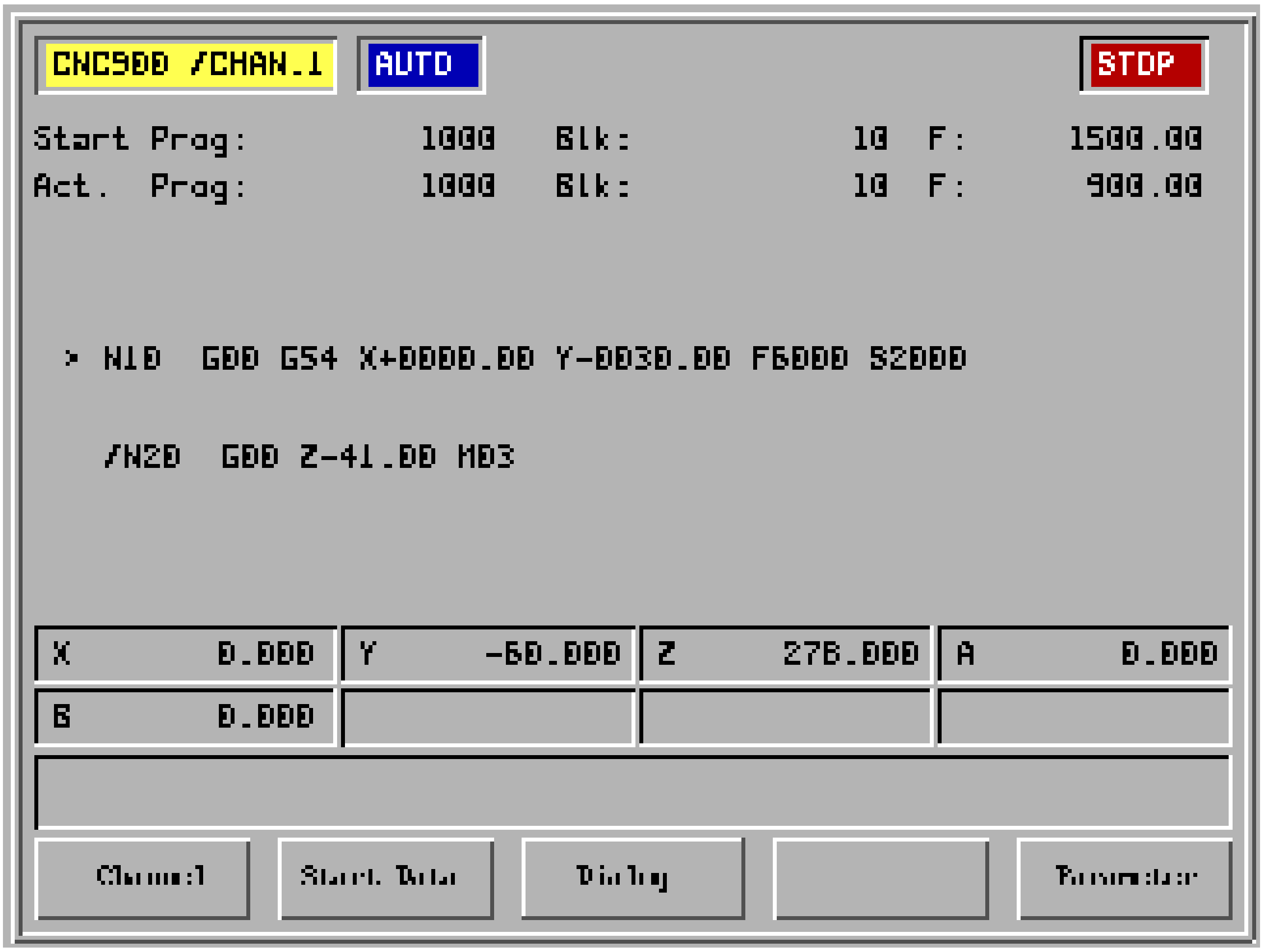
Switch on: Press key /
The key is only effective before program start.
The display of the operating mode is completed by the symbol '/'.

Switch off: Press key / again.

2.6.2 Automatic mode (continued)

In **automatic sequential block**, all blocks of a program are worked one after the other.

Switch on sequential block with key .



F1 Channel

F2 Start data

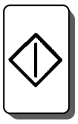
F3 Dialogue

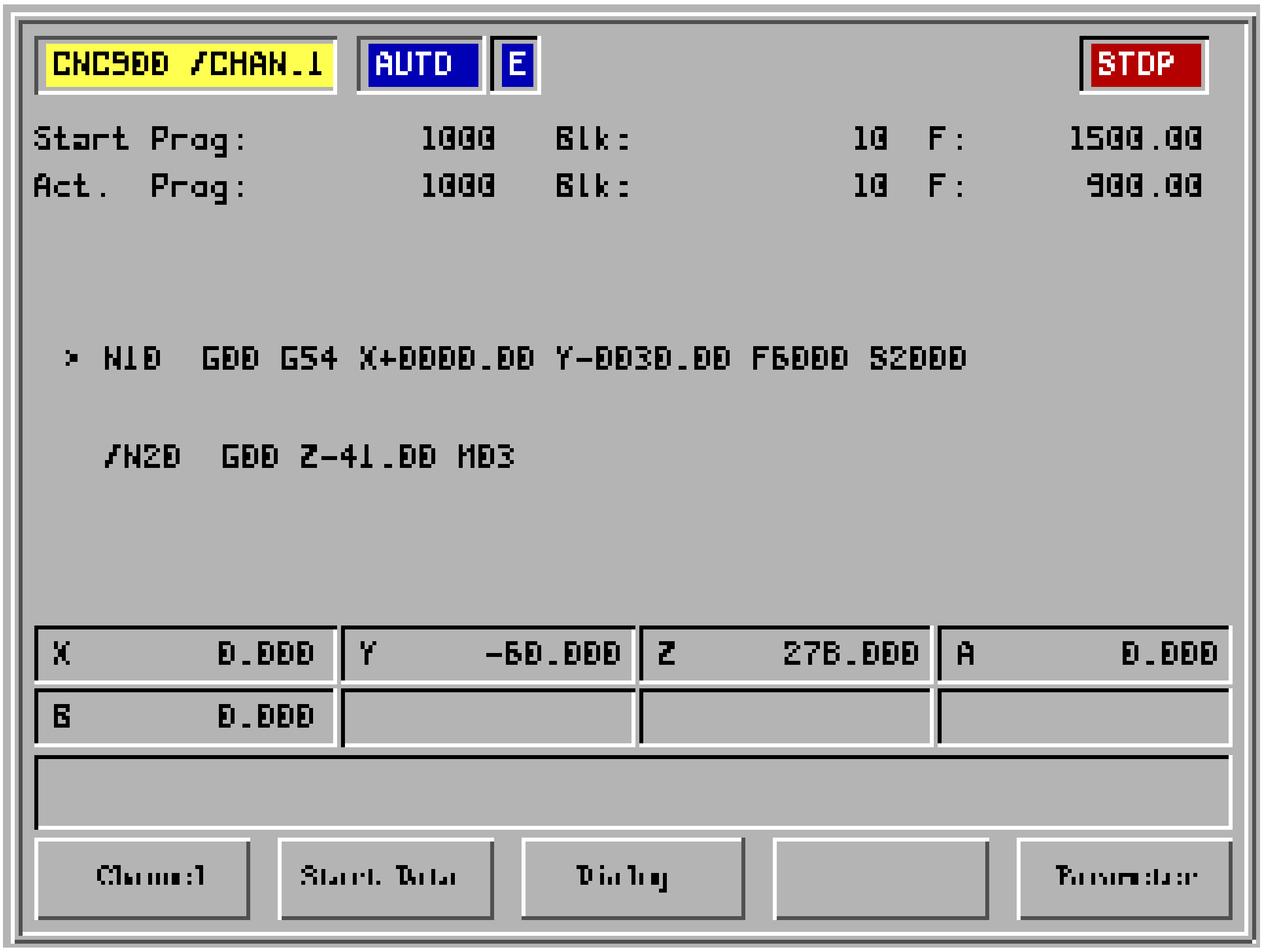
F4 -

F5 Parameter

2.6.2 Automatic mode (continued)

In automatic single block, only one block is worked after the start. Start the next block with key

. Switch on automatic single block with key .



F1 Channel

F2 Start data

F3 Dialogue

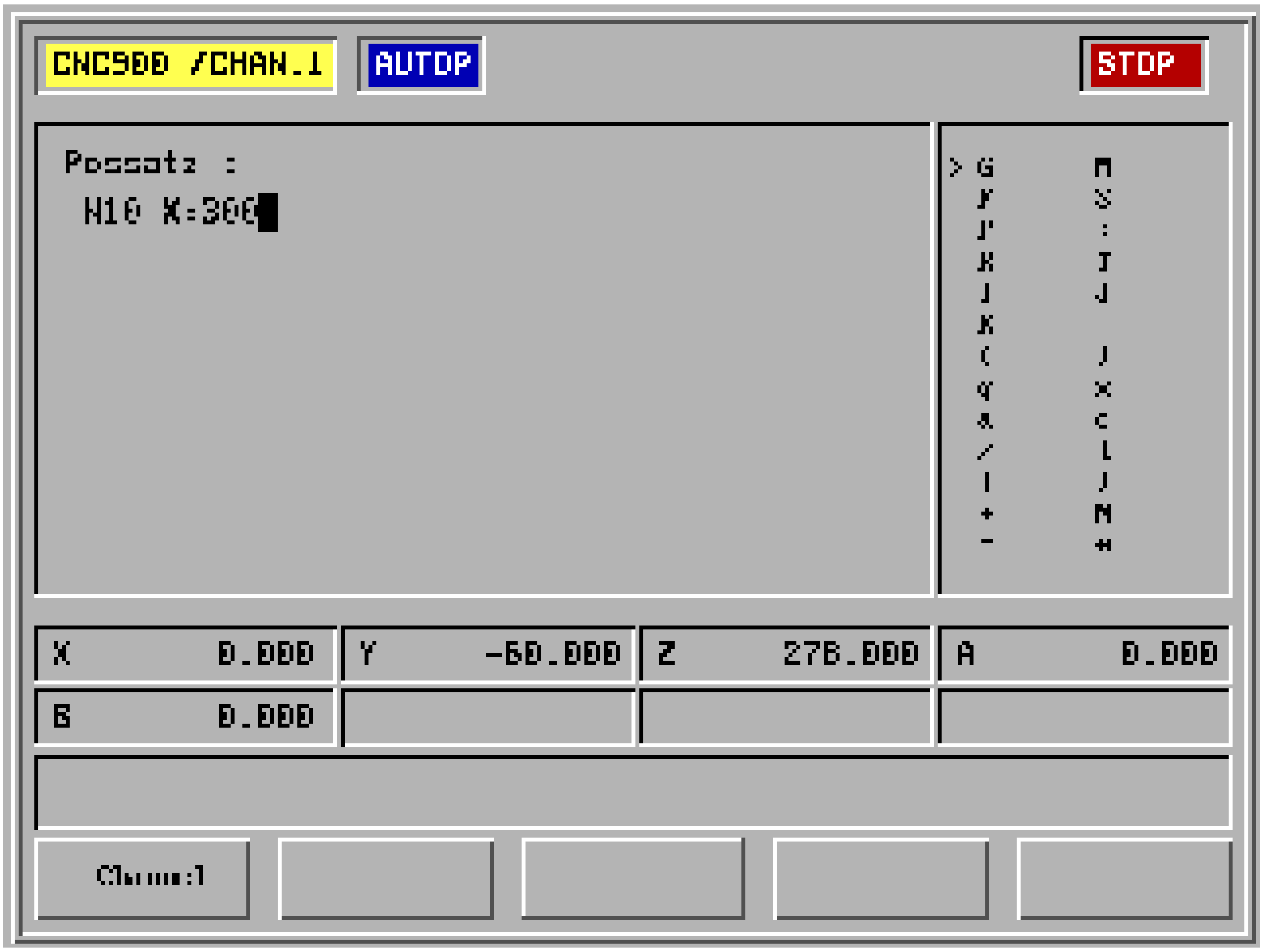
F4 -

F5 Parameter

2.6.2 Automatic mode (continued)

With positioning, a complete NC block or parts of it can be input via the numerical keys and worked without storing.

Switch on positioning with key ⊣.



F1 Channel

F2 -

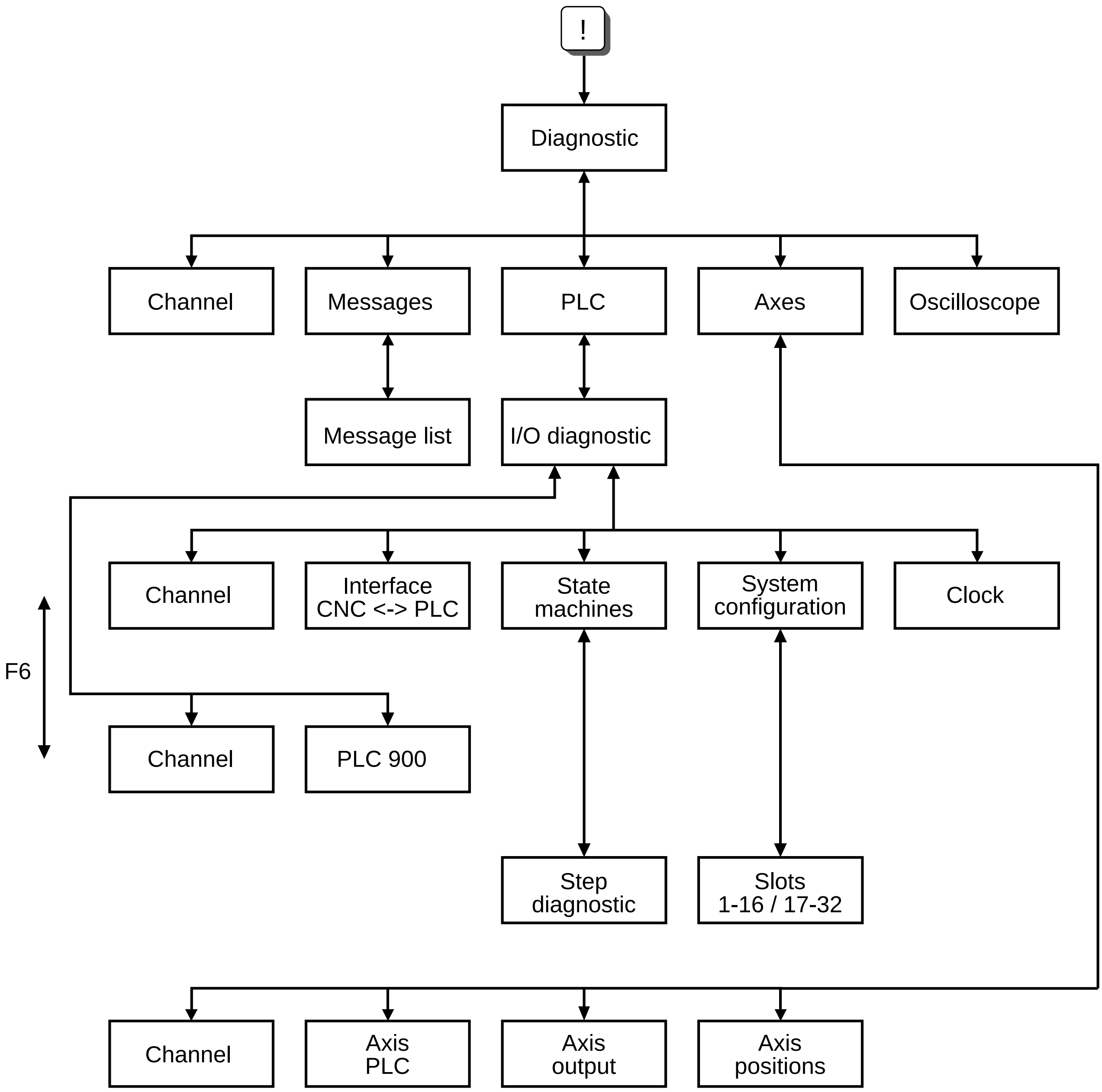
F3 -

F4 -

F5 -

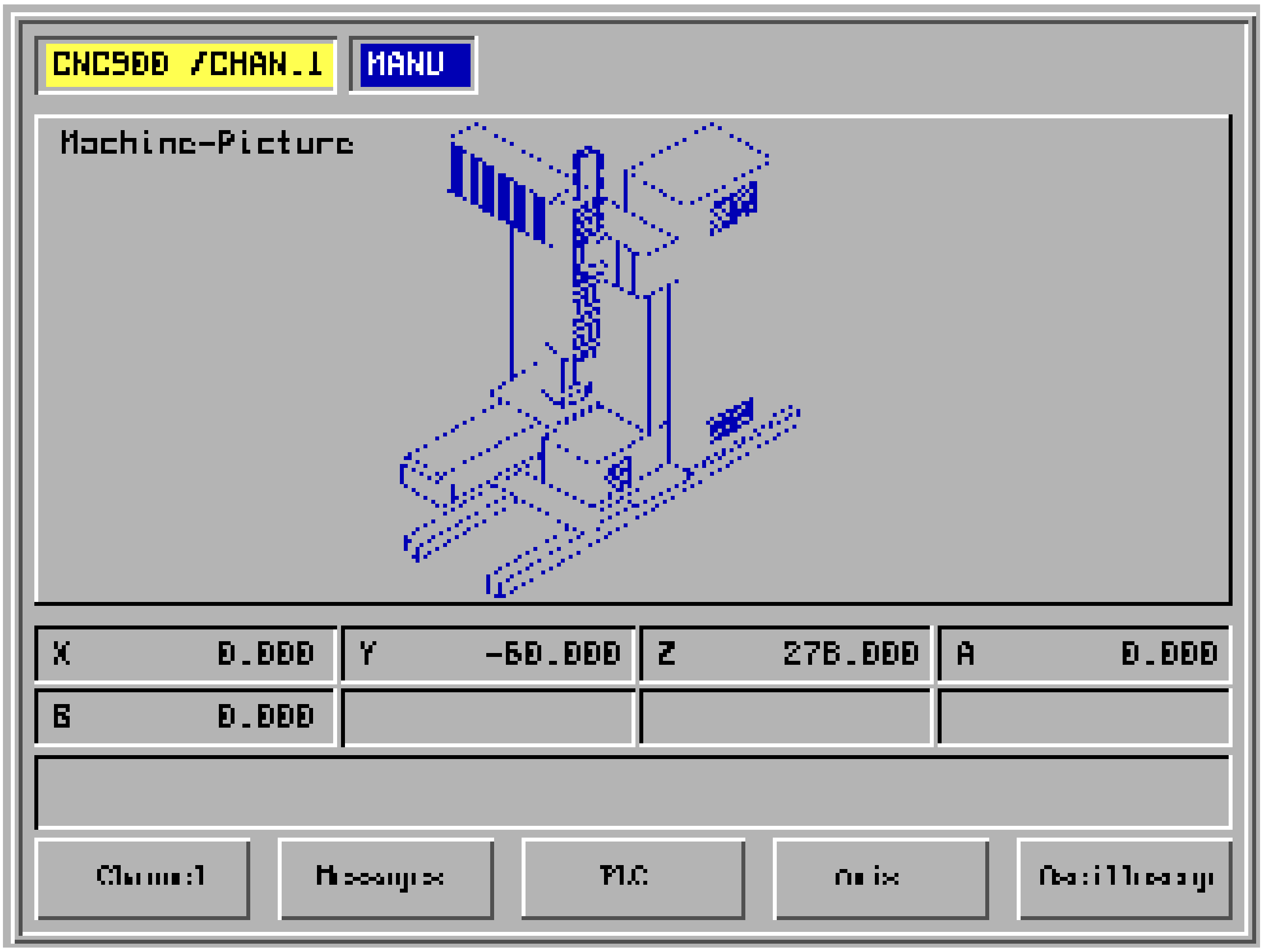
2.7 Diagnostic

Menue tree



2.7 Diagnostic

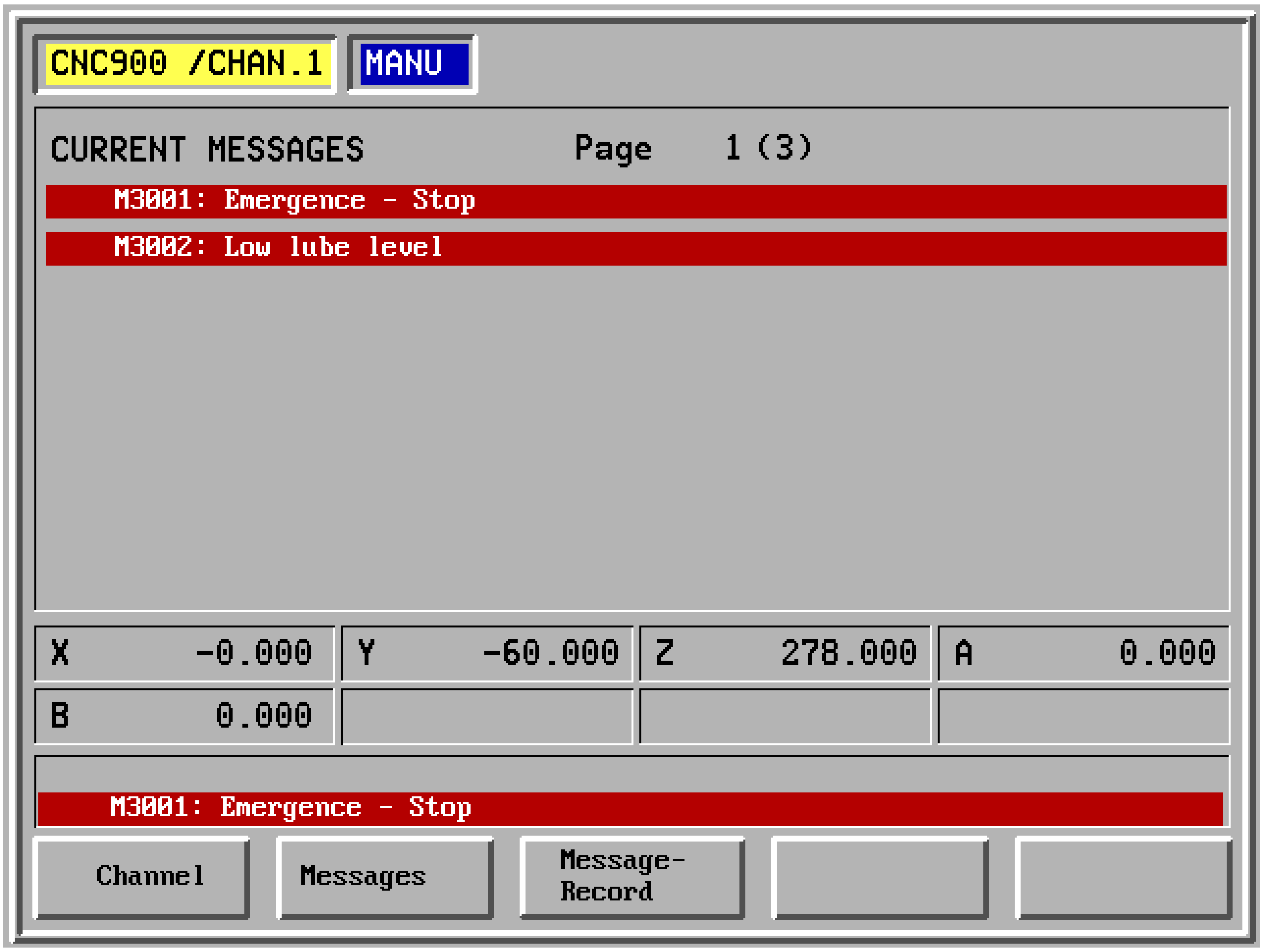
Diagnostic is switched on with the key [!]. The machine picture appears, e.g.



F1	Channel	
F2	Messages	Display messages
F3	PLC	PLC diagnostic
F4	Axes	Axis drive diagnostic
F5	Oscilloscope	

2.7.1 Messages

Current messages



F1 Channel

F2 Messages

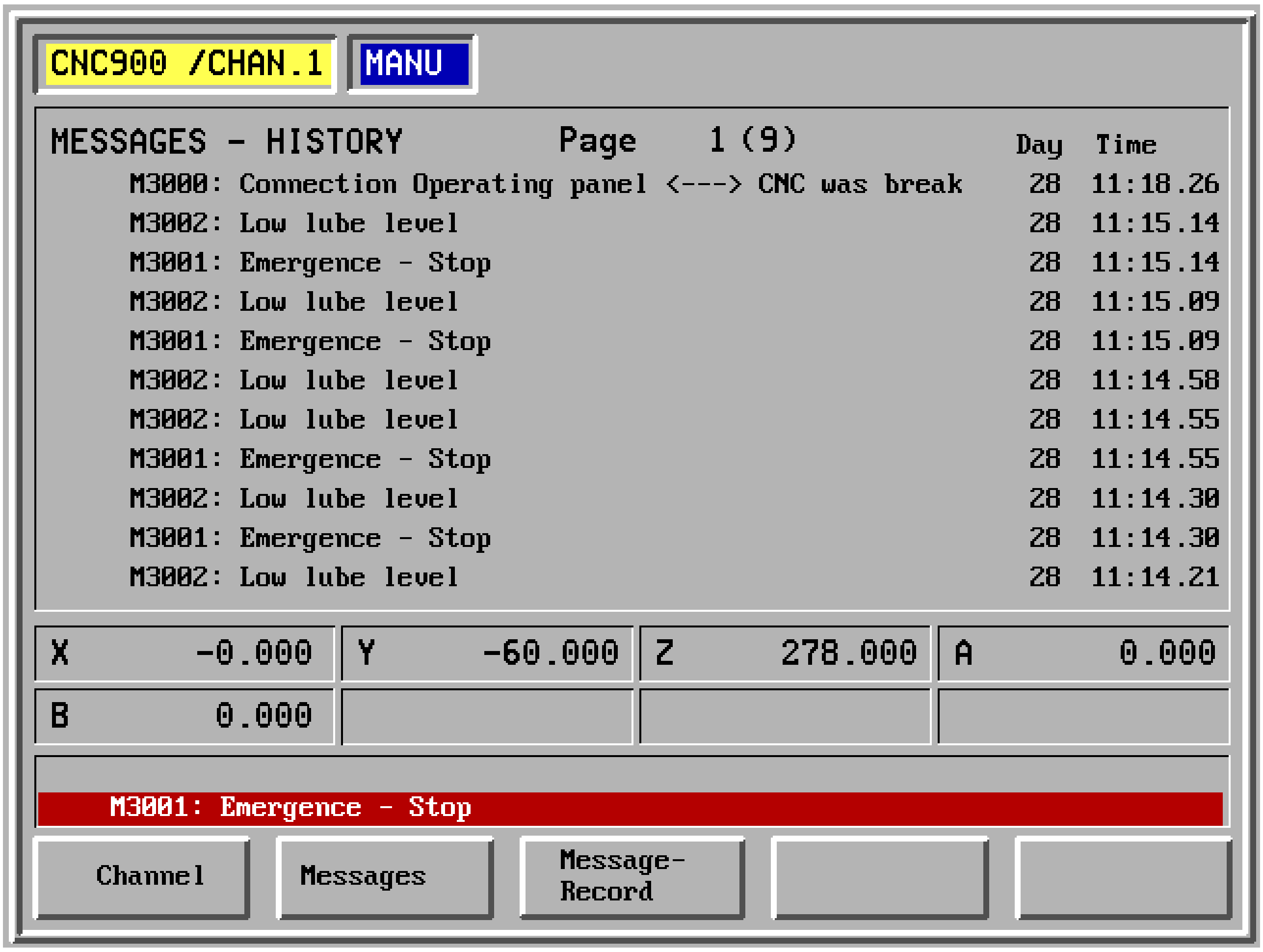
F3 Message record Display of previous messages

F4 Error information

F5 -

2.7.1 Messages

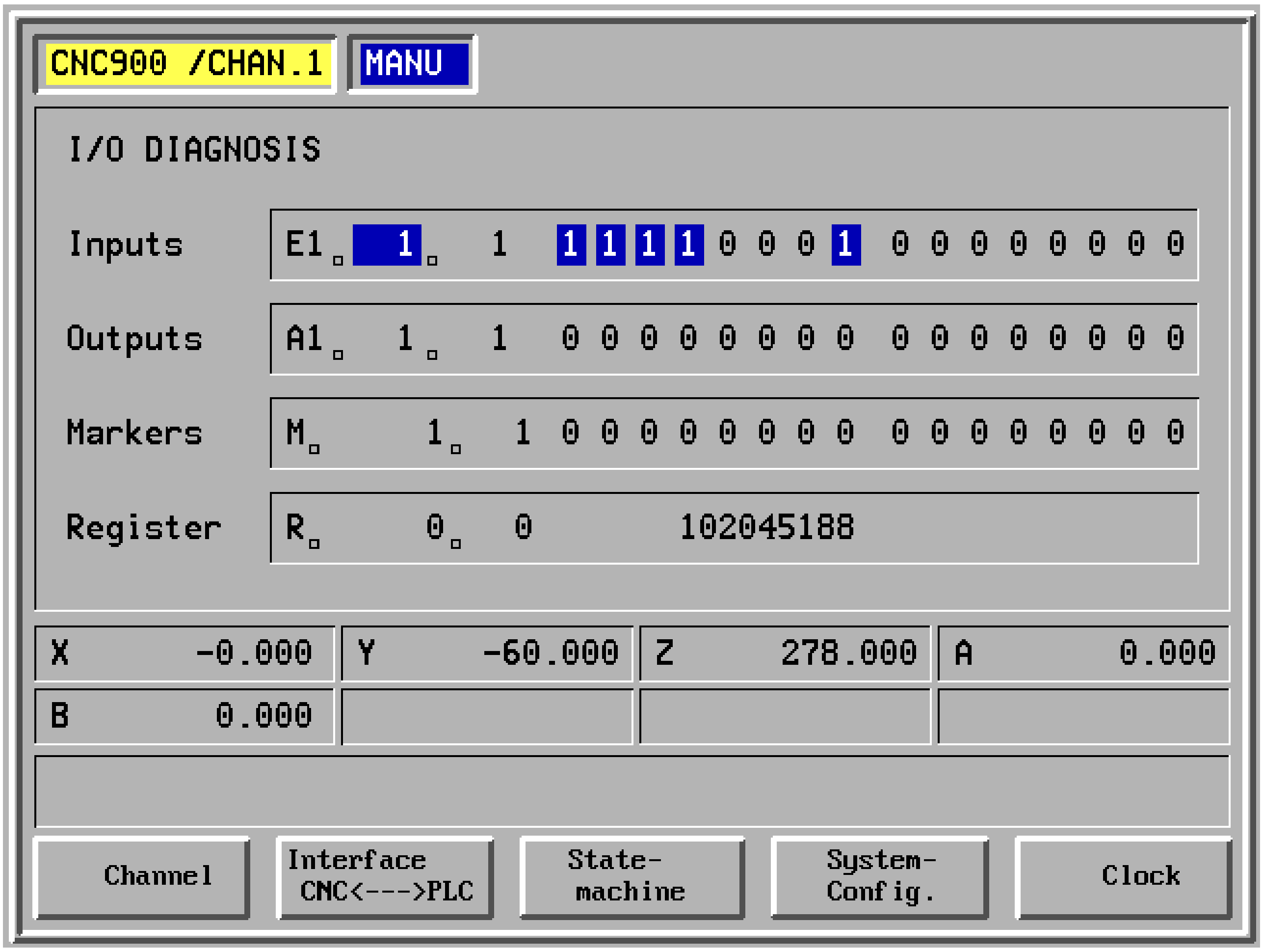
Current messages



F1	Channel	
F2	Messages	
F3	Message record	Display of previous messages
F4	-	
F5	-	

2.7.2 PLC

I/O diagnostic



F1 Channel

F2 PLC900

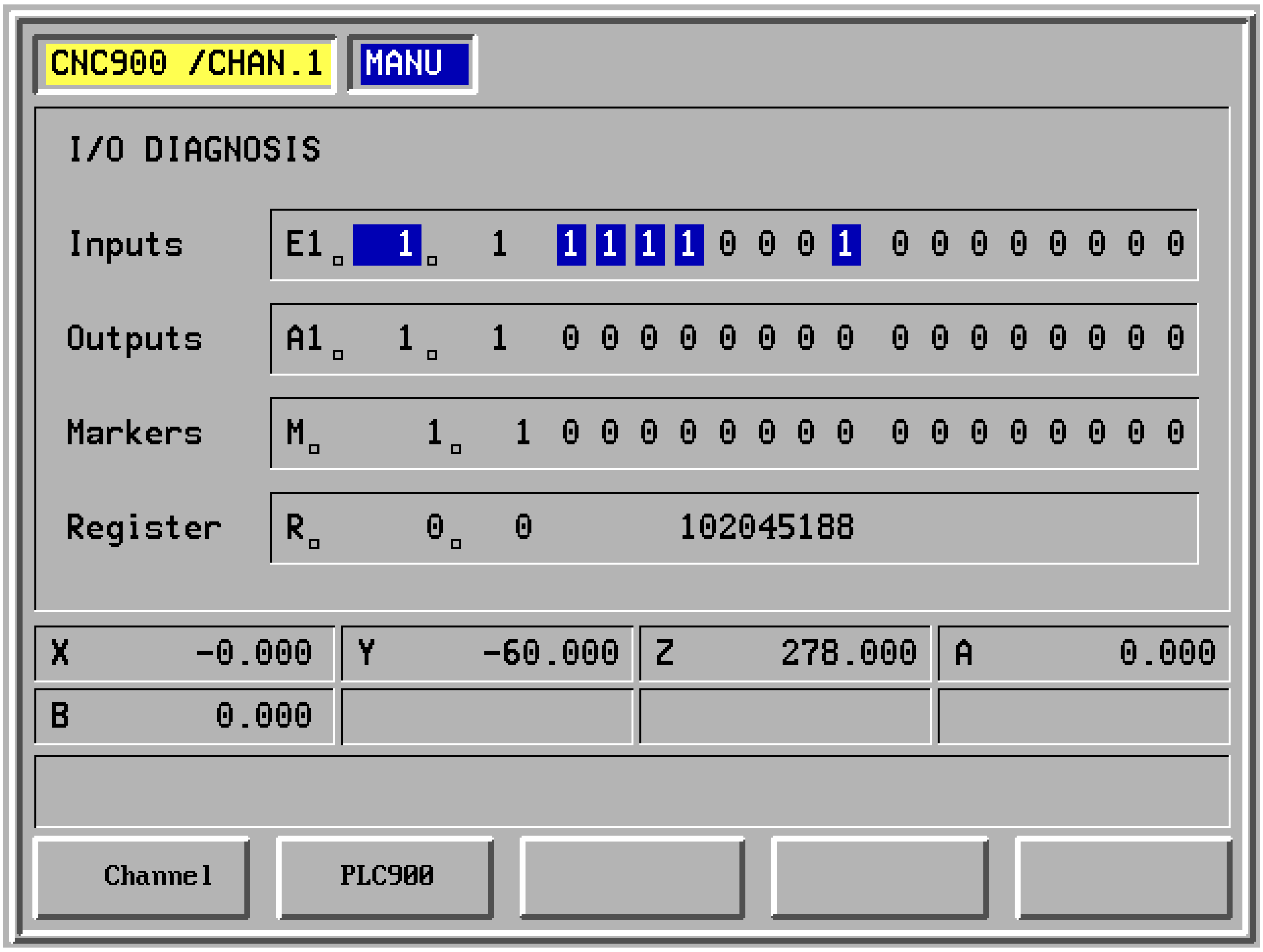
F3 Interface CNC <—> PLC

F4 -

F5 -

2.7.2 PLC (continued)

I/O diagnostic



F1 Channel

F2 -

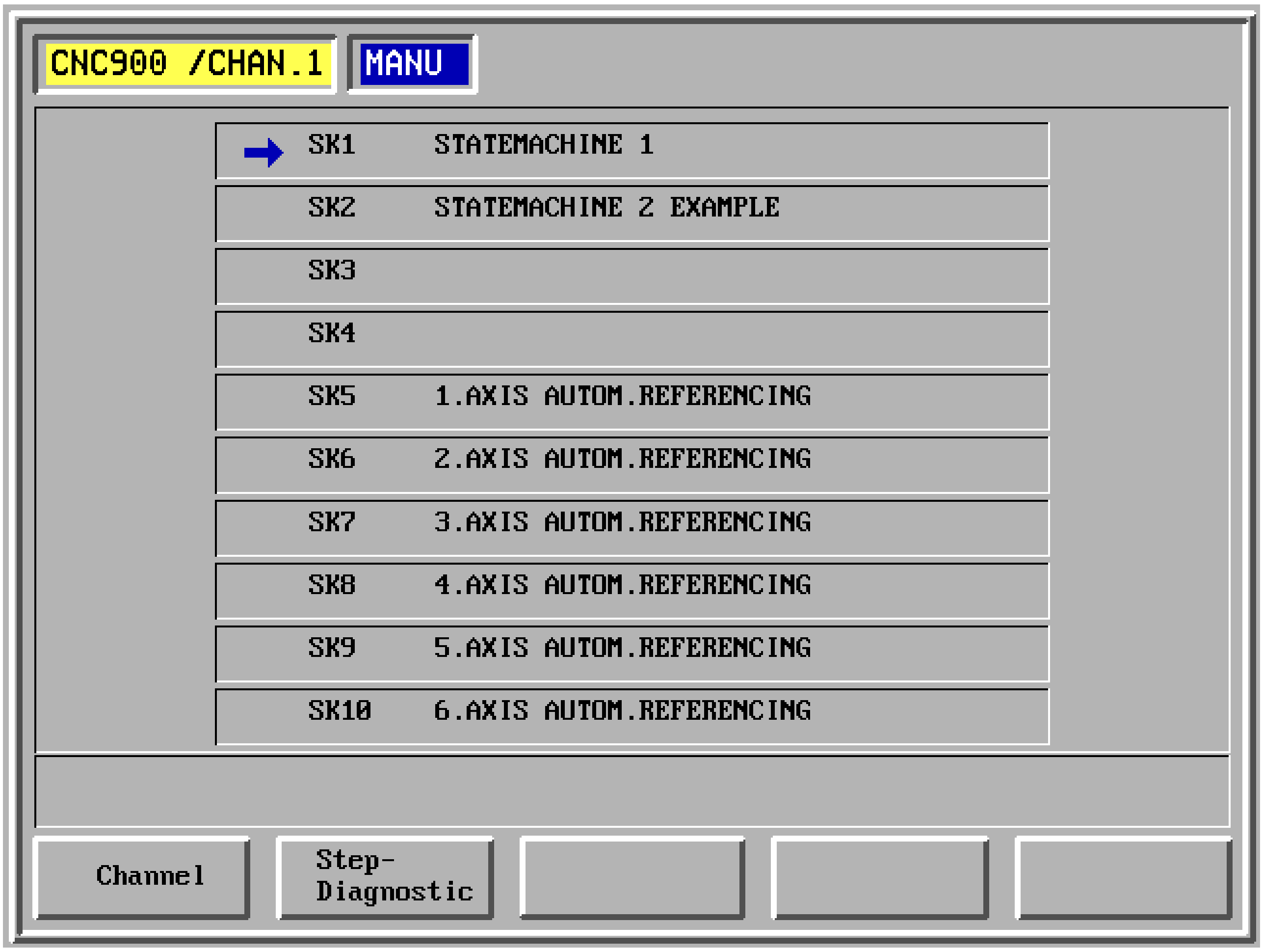
F3 State machine

F4 Sytem configuration

F5 Clock

2.7.2 PLC (continued)

State machine diagnostic



F1 Channel

F2 State machine diagnostic

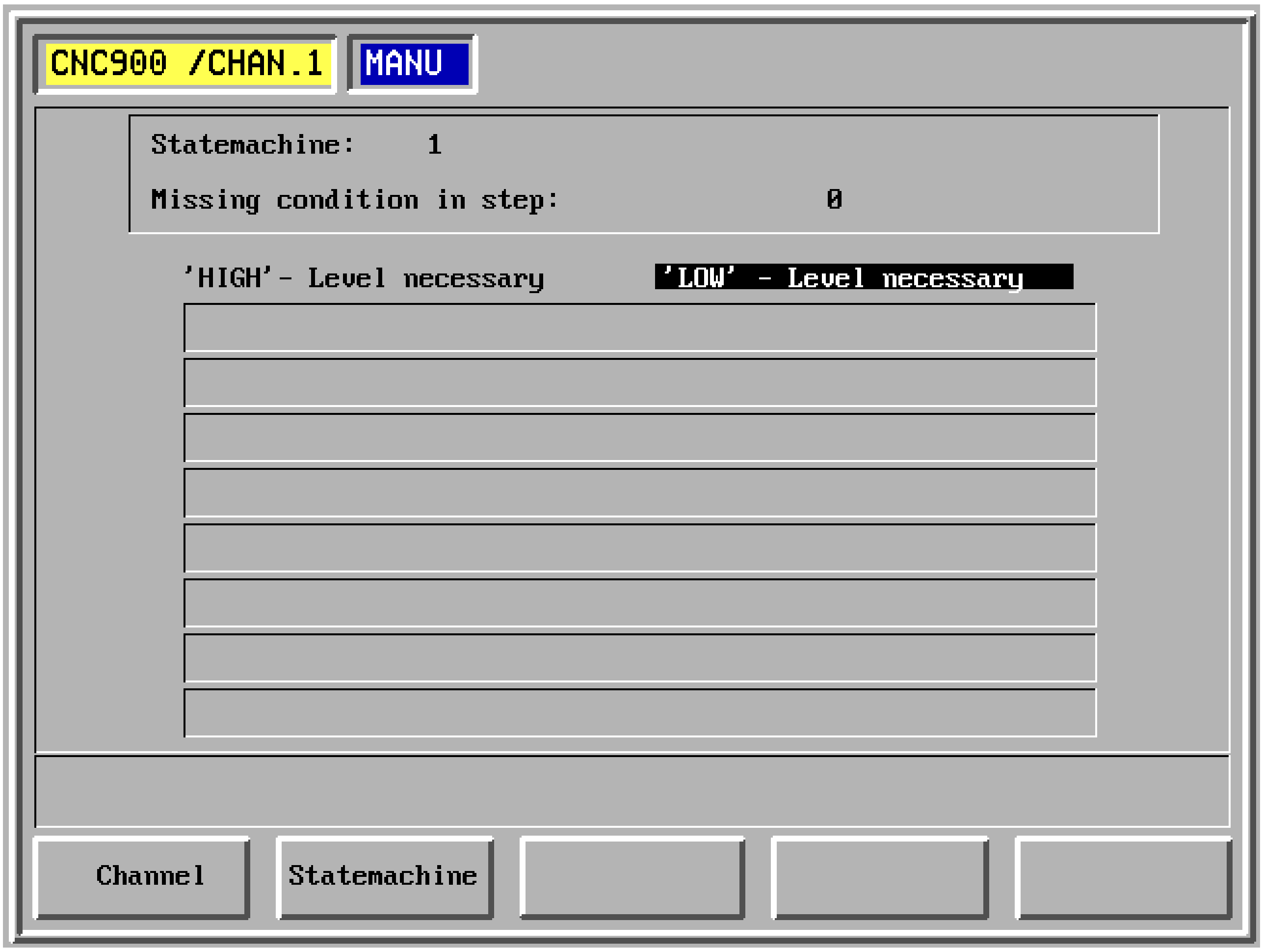
F3 -

F4 -

F5 -

2.7.2 PLC (continued)

State machine diagnostic



F1 Channel

F2 State machine

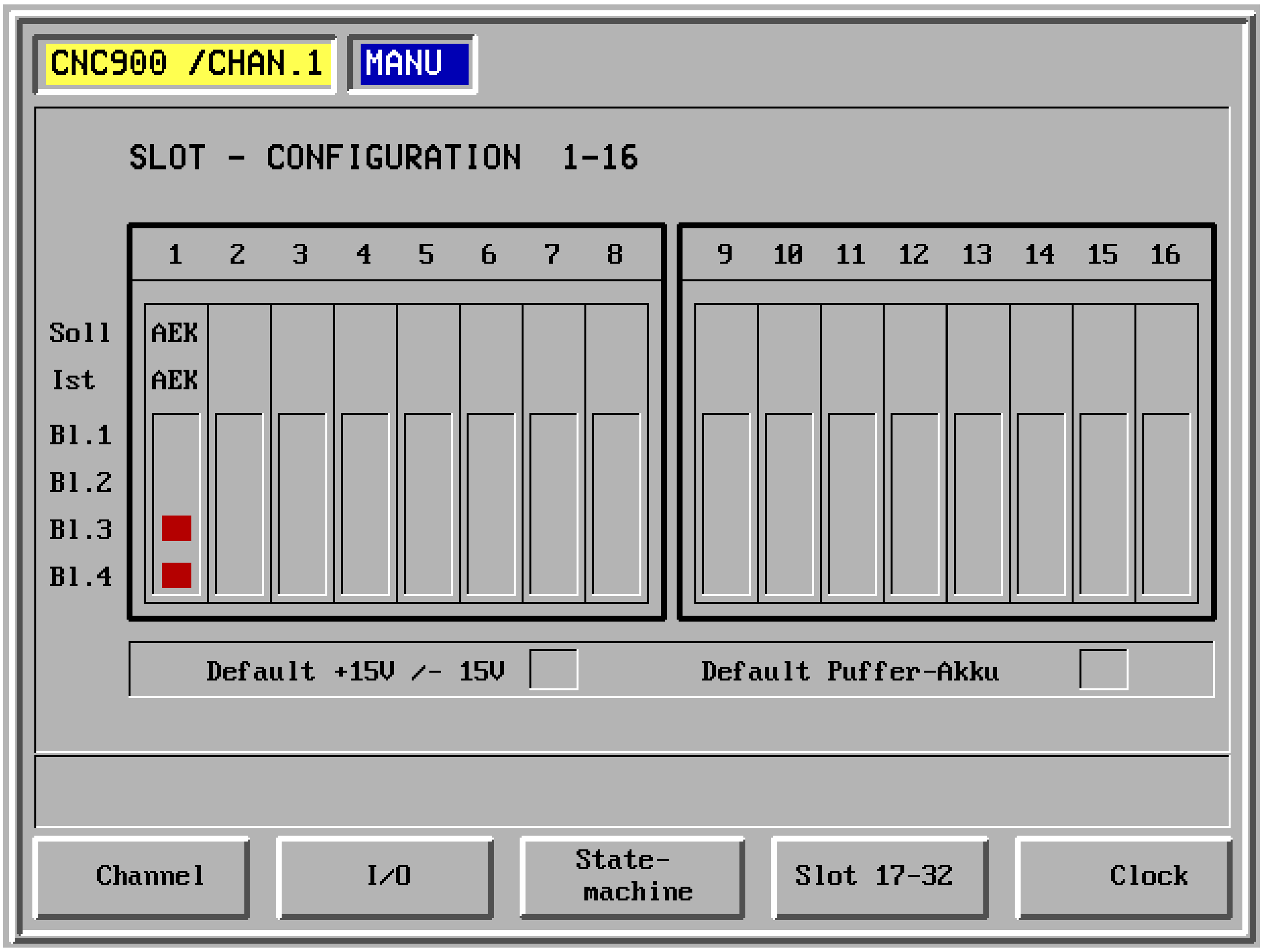
F3 Page forwards

F4 Page back

F5 -

2.7.2 PLC (continued)

System configuration



F1 Channel

F2 -

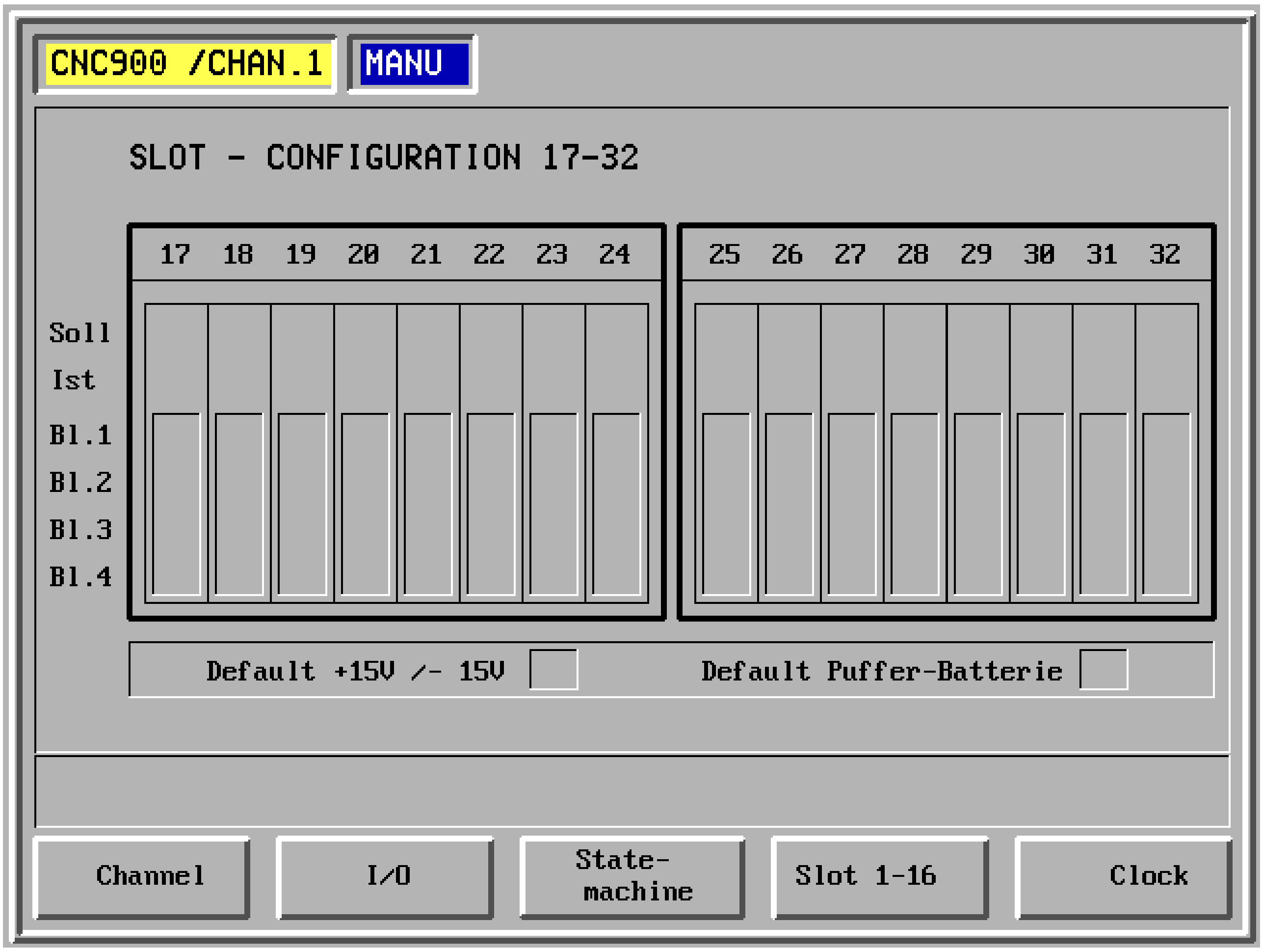
F3 State machine

F4 System configuration, slots 17-32

F5 Clock

2.7.2 PLC (continued)

System configuration



F1 Channel

F2 -

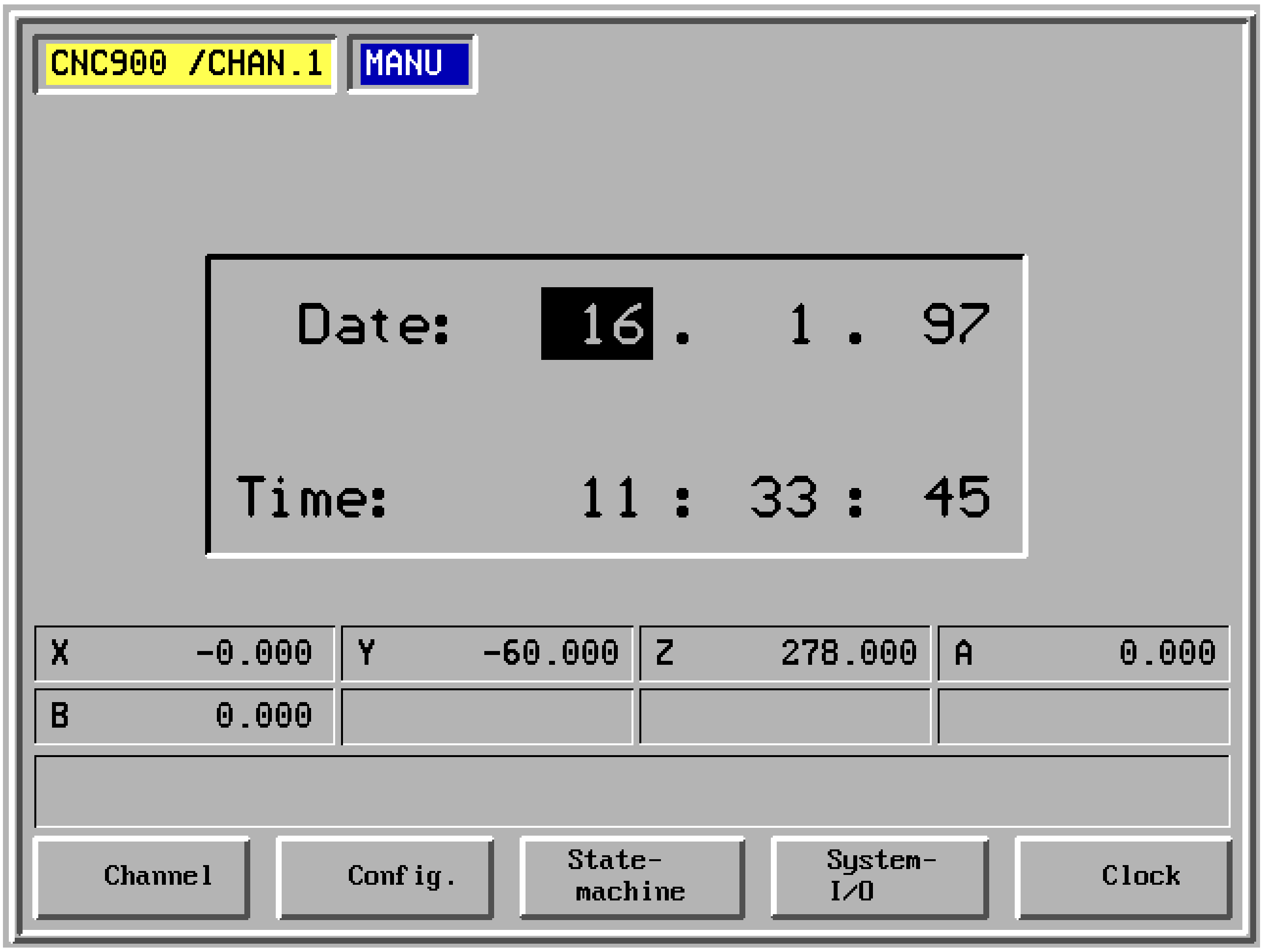
F3 State machine

F4 System configuration, slots 1-16

F5 Clock

2.7.2 PLC (continued)

Clock



F1 Channel

F2 I/O

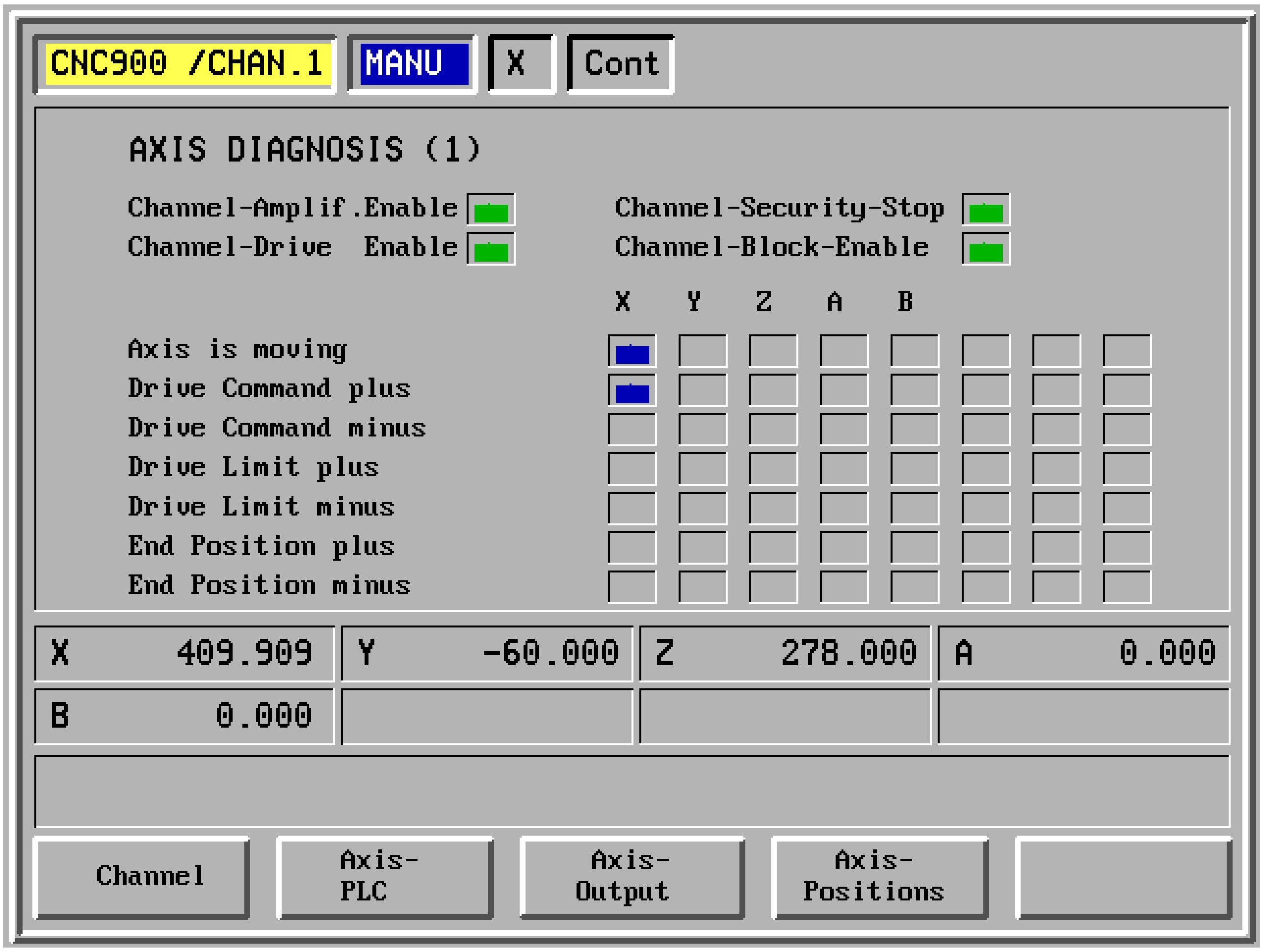
F3 State machine

F4 System configuration

F5 Clock

2.7.3 Axes

Axis diagnostic 1



F1 Channel

F2 Axis PLC

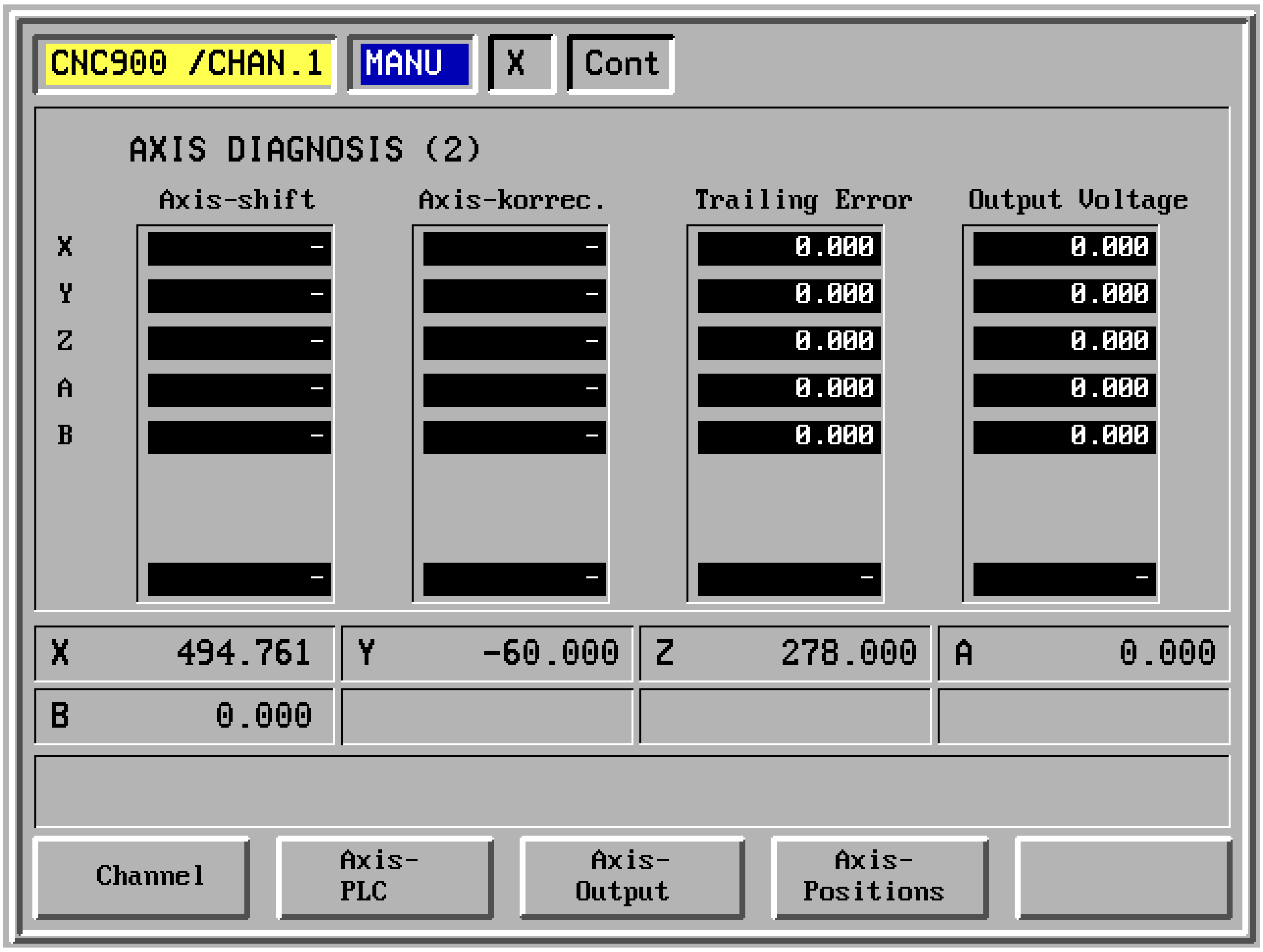
F3 Axis output

F4 Axis positions

F5 -

2.7.3 Axes (continued)

Axis diagnostic 2



F1 Channel

F2 Axis PLC

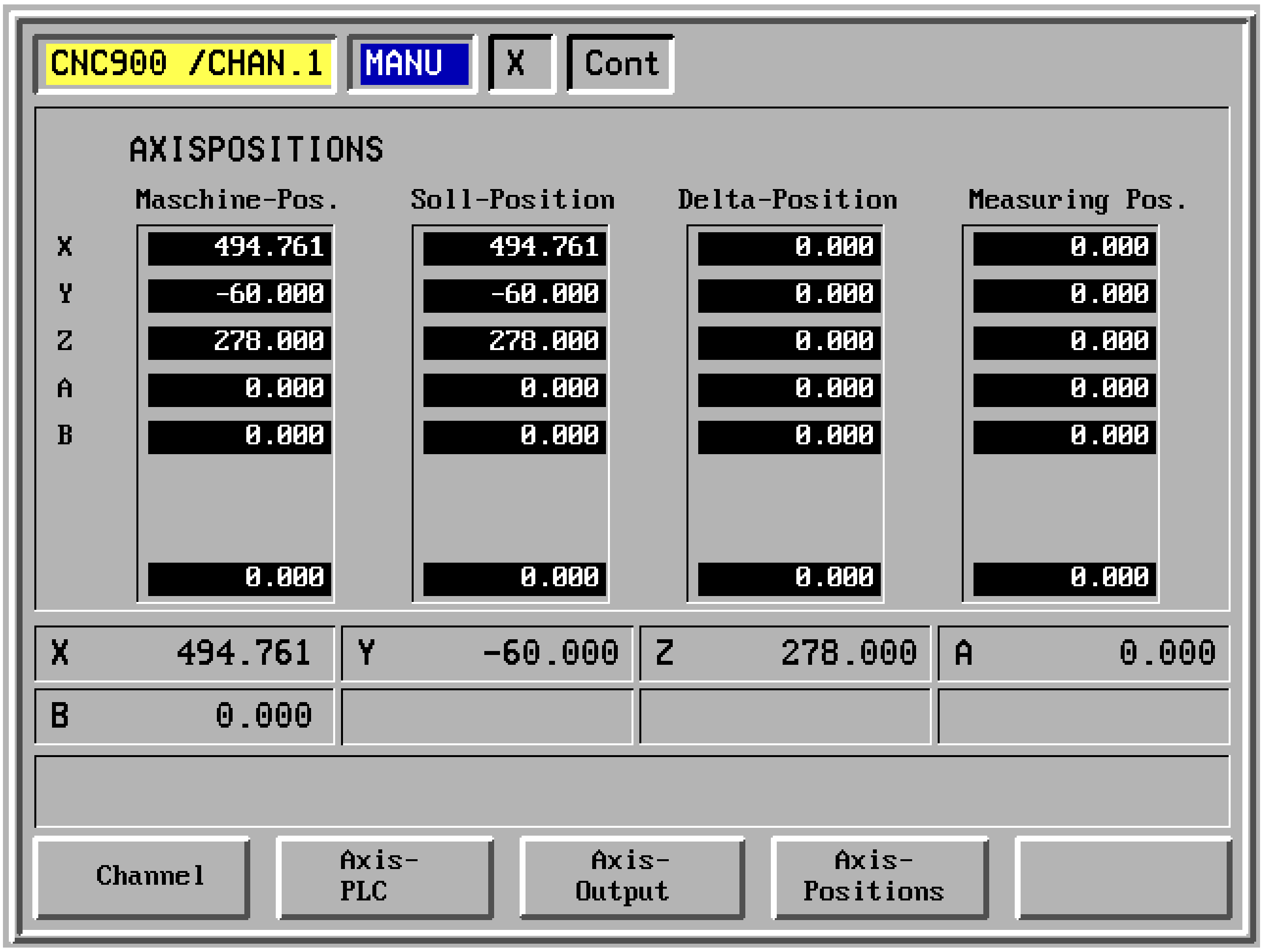
F3 Axis output

F4 Axis positions

F5 -

2.7.3 Axes (continued)

Axis positions



F1 Channel

F2 Axis PLC

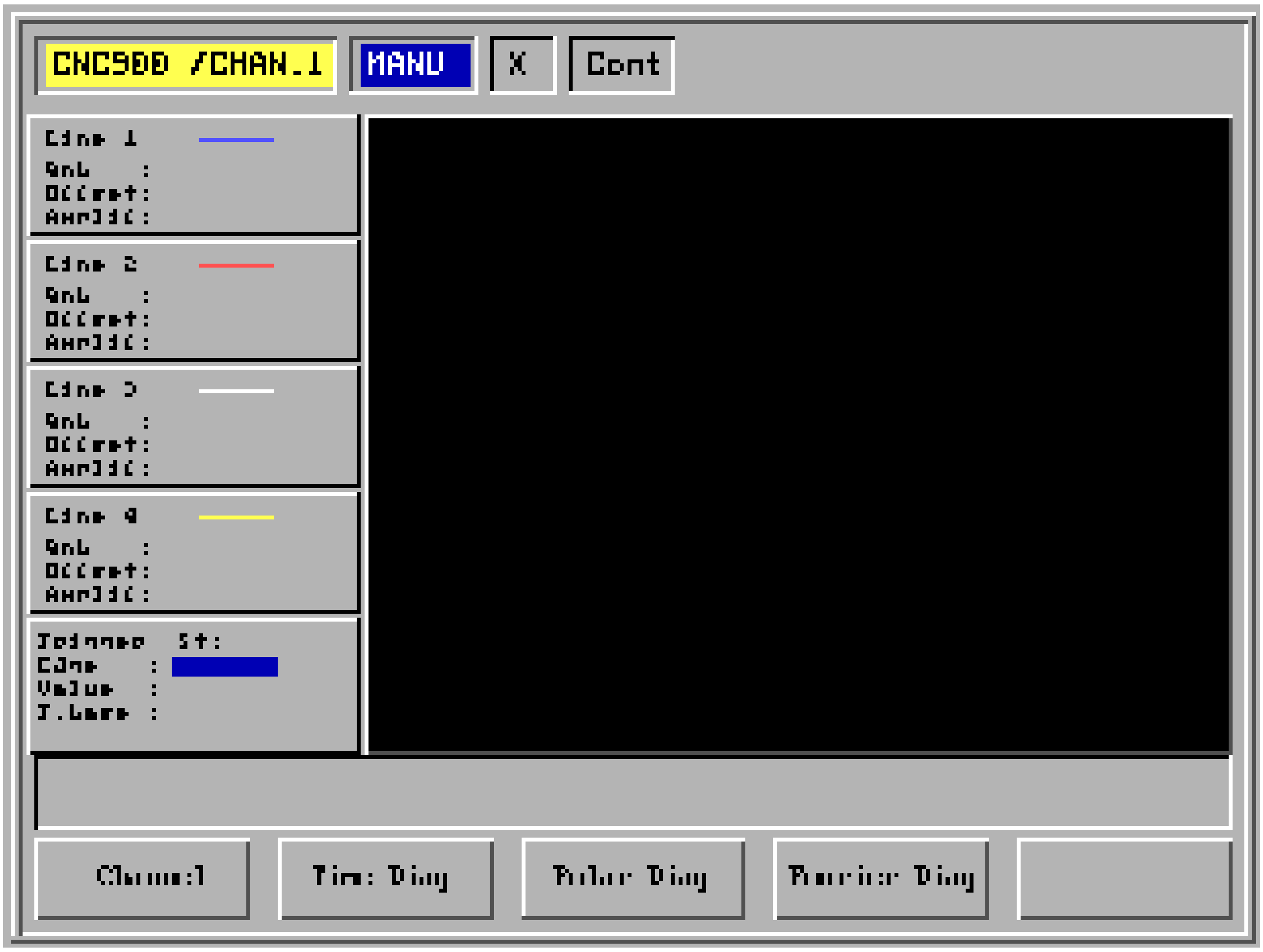
F3 Axis output

F4 Axis positions

F5 -

2.7.4 Oscilloscope

Four-channel oscilloscope with time-, polar- and Fourier diagnostic for evaluating the mechanical settings and for recognising defective mechanical parts.



F1 Channel

F2 Start time diagnostic

F3 Start polar diagnostic

F4 Start Fourier diagnostic

F5 -

2.7.4 Oscilloscope (continued)

Qnr Parameter number (connection)

Number	Meaning	
q2150	Command position	[mm, degree]
q2152	Actual position	[mm, degree]
q2160	Lag distance	[mm, degree]
q2161	Coupling correction	[mm, degree]
q2168	Actual difference (corresponds to speed)	[mm, degree]
q2169	Output voltage of positioning control	[V]

Offset Vertical offset

Amplif Amplification, vertical resolution, units per division

Trigger St Trigger line number

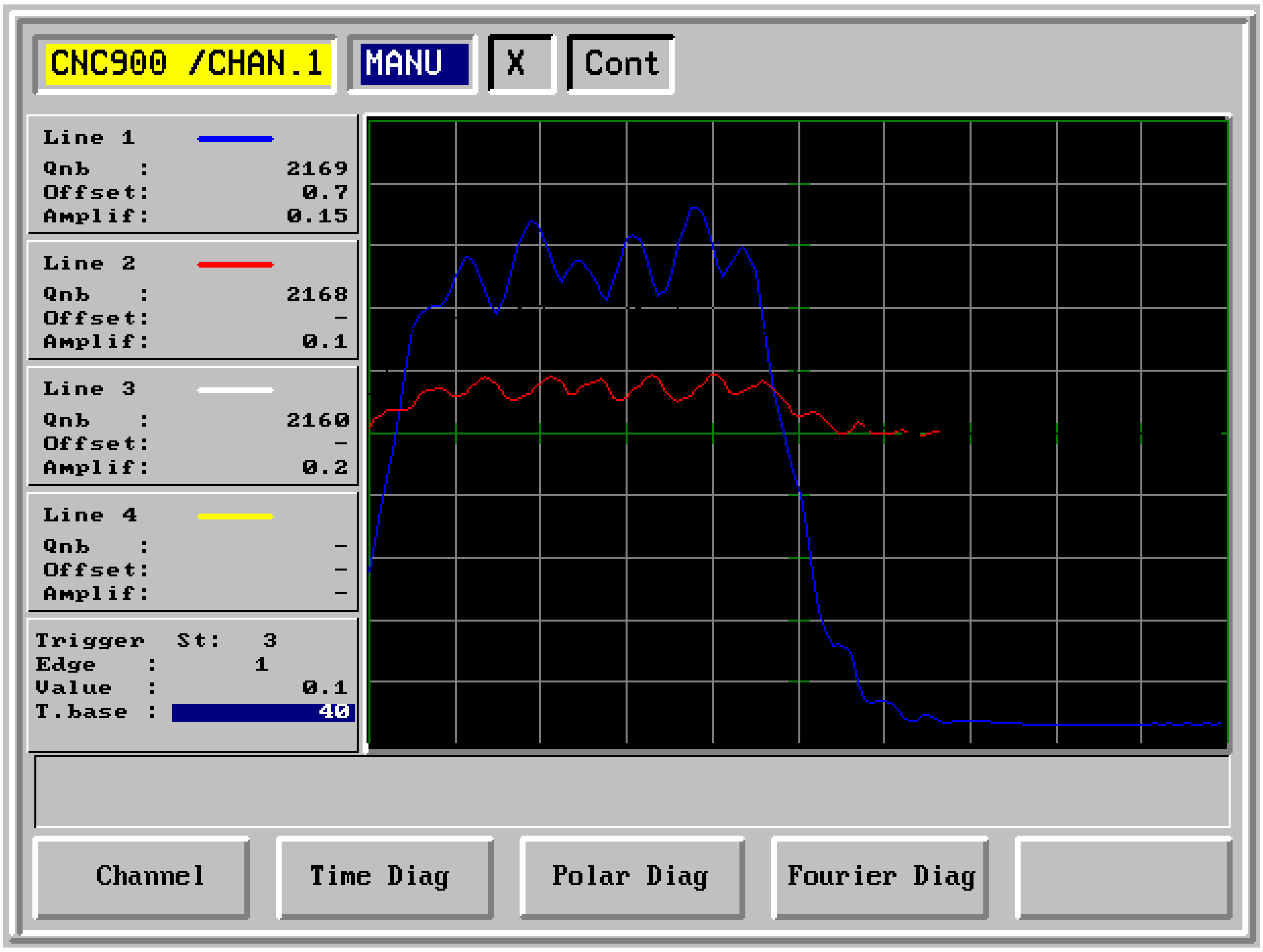
Edge
 1 trigger at rising edge
-1 trigger at declining edge
 0 trigger immediately

Value Value at which it is triggered.

TB/Freq Time basis / frequency, horizontal resolution, units per division [ms, Hz]

2.7.4 Oscilloscope (continued)

Example: Three-channel time diagnostic

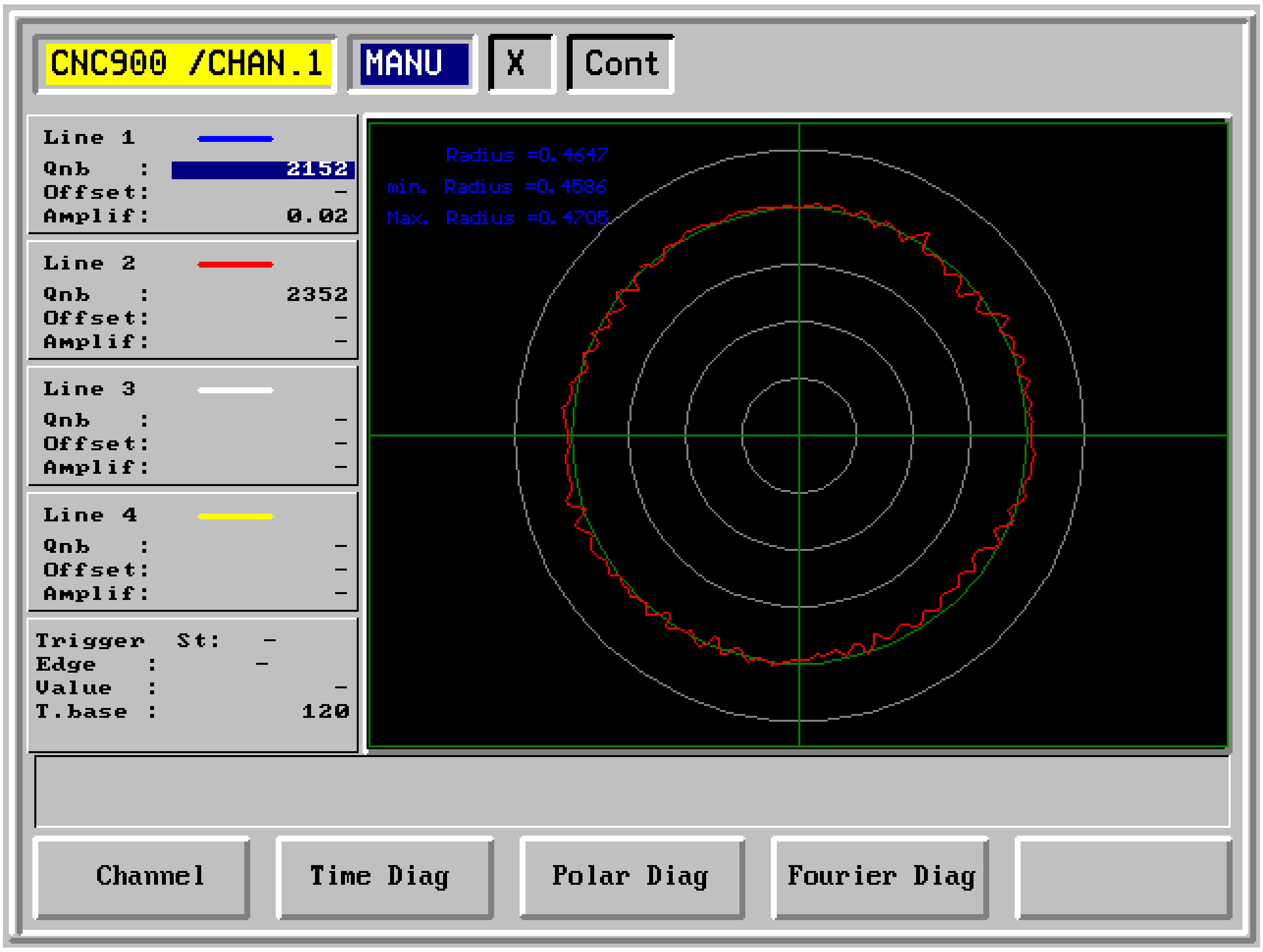


Line 1	Qnr	2169	output voltage of positioning control in V
	Offset	0,7	vertical offset
	Amplif.	0,15	units per division
Line 2	Qnr		actual difference in mm/degree
	Amplif.	0,1	units per division
Line 3	Qnr	2160	lag distance in mm/degree
	Amplif.	0,2	units per division

Trigger	Line number	3	
Edge		1	trigger at rising edge
Value		0,1	trigger at this value
Time basis / Frequency		40	units per division

2.7.4 Oscilloscope (continued)

Example: polar diagnostic, circle test

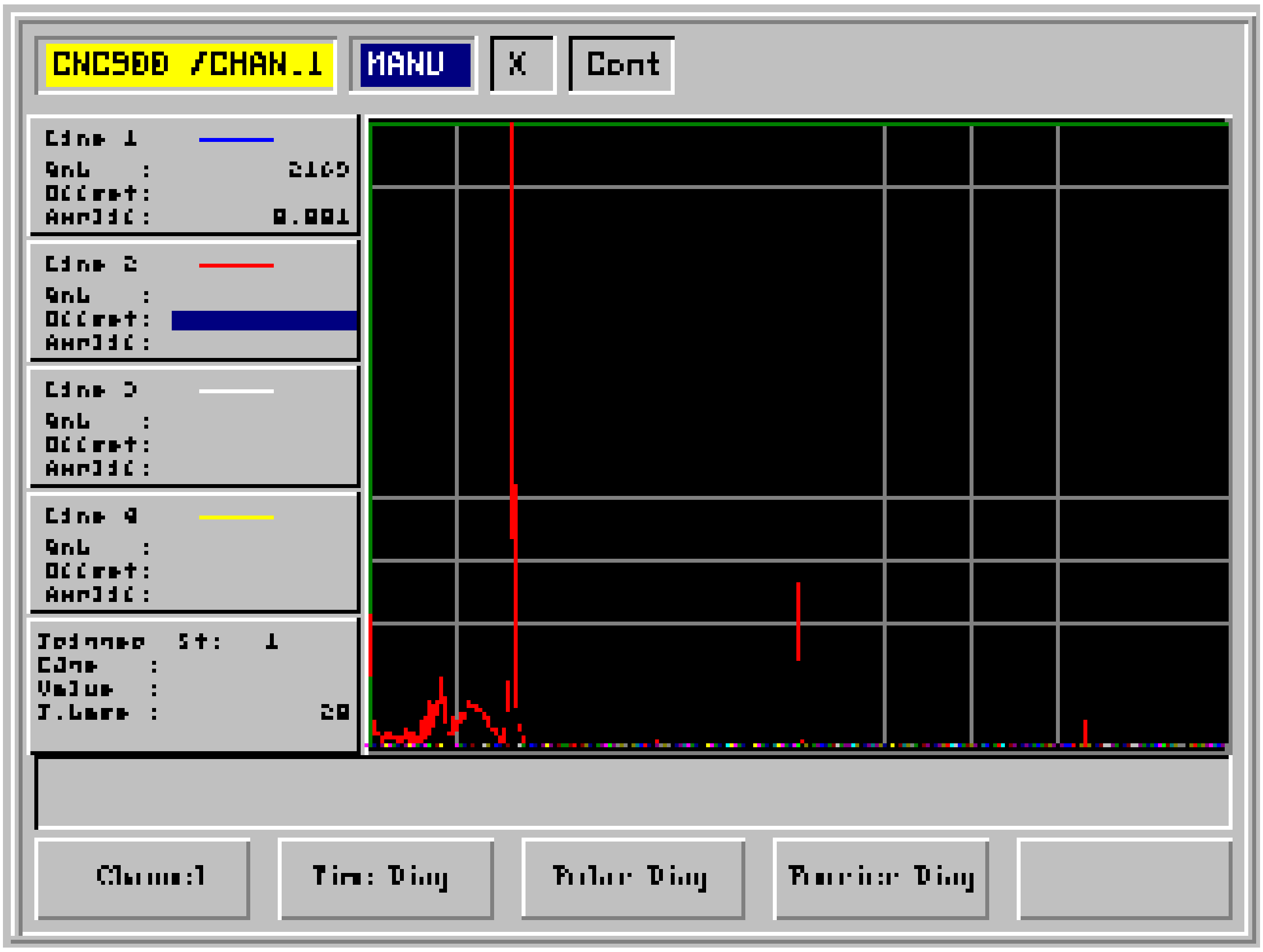


Line 1	Qnr	2152	actual position 1st axis in mm/degree
	Amplification	0,02	units per division
Line 2	Qnr	2352	actual position 2nd axis in mm/degree
Time basis / Frequency		120	units per division chose time/frequency basis so that at least one full circle is run.

2.7.4 Oscilloscope (continued)

Example: Fourier diagnostic, frequency spectrum

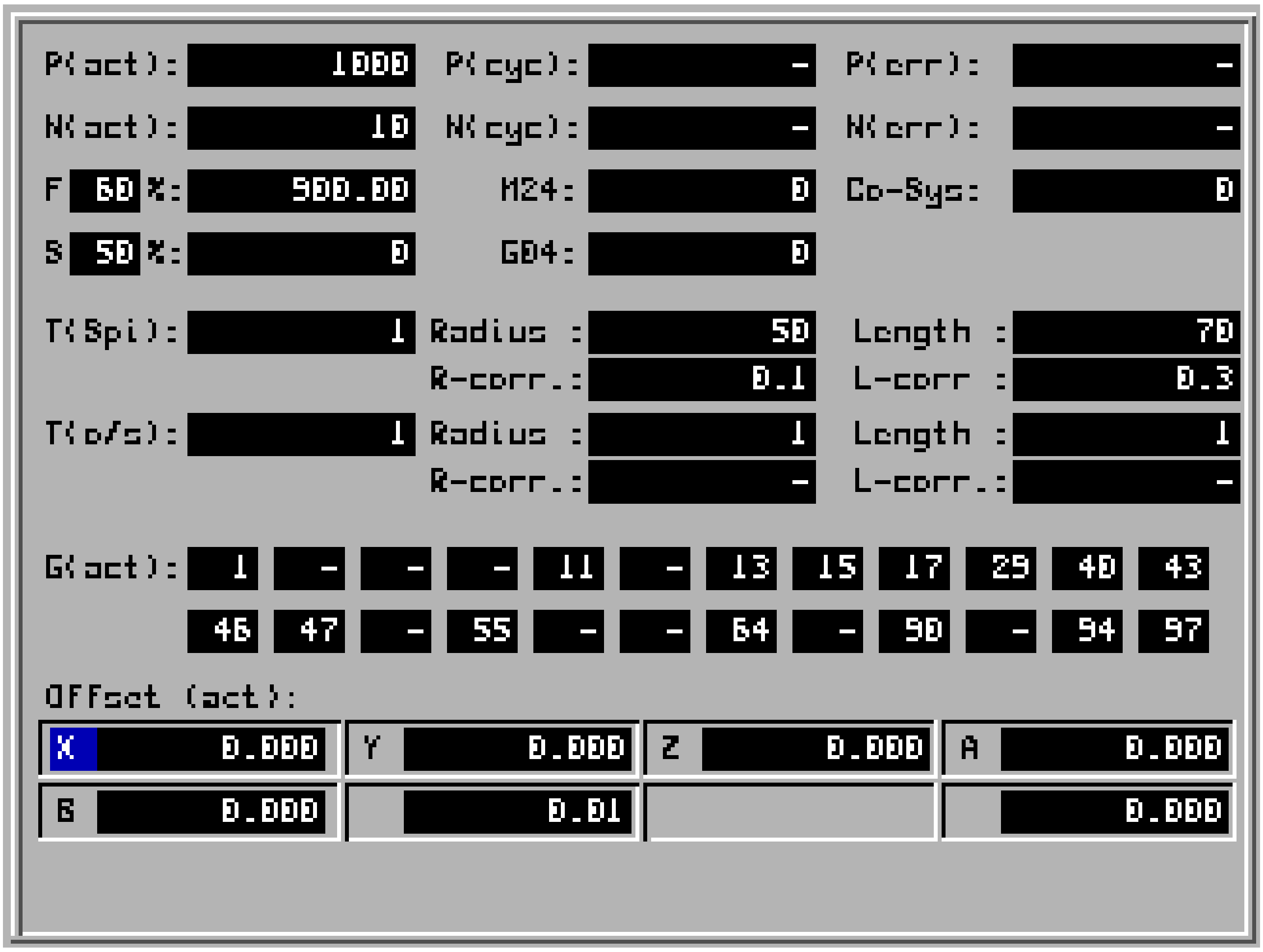
Frequency spectrum with oscillations at 33Hz and harmonics at 99Hz and 165HZ



Line 1	Qnr	2169	output voltage of positioning control in V
	Amplification	0,01	units per division vertically
Trigger	Line number	1	
Time basis / frequency		20	units per division horizontally in Hz

2.7.5 Additional informations

Additional online informations can be called up during operating with the key ?.

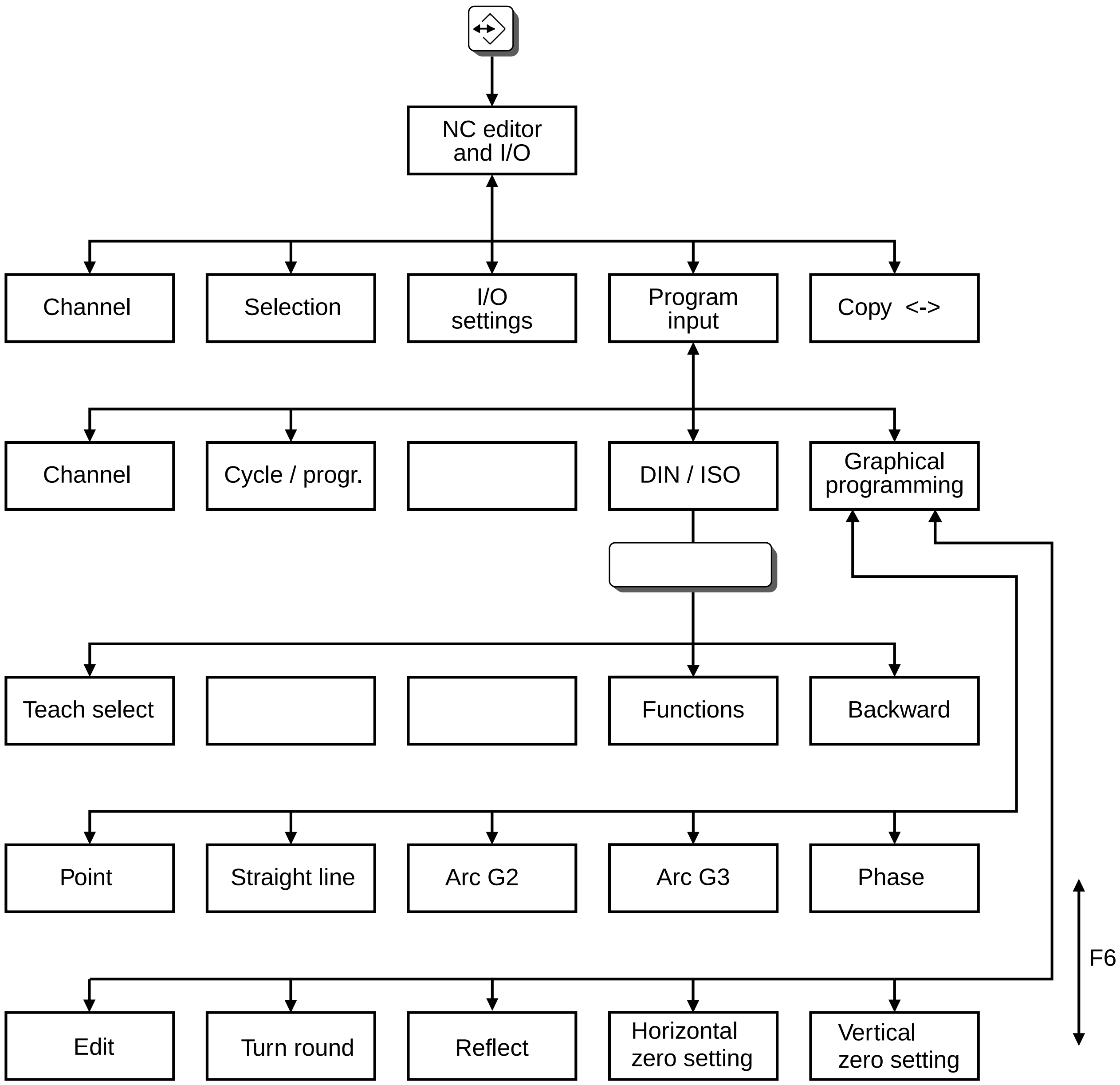


Note: With negative program numbers it concerns a cycle.

2.8 NC editor and I/O

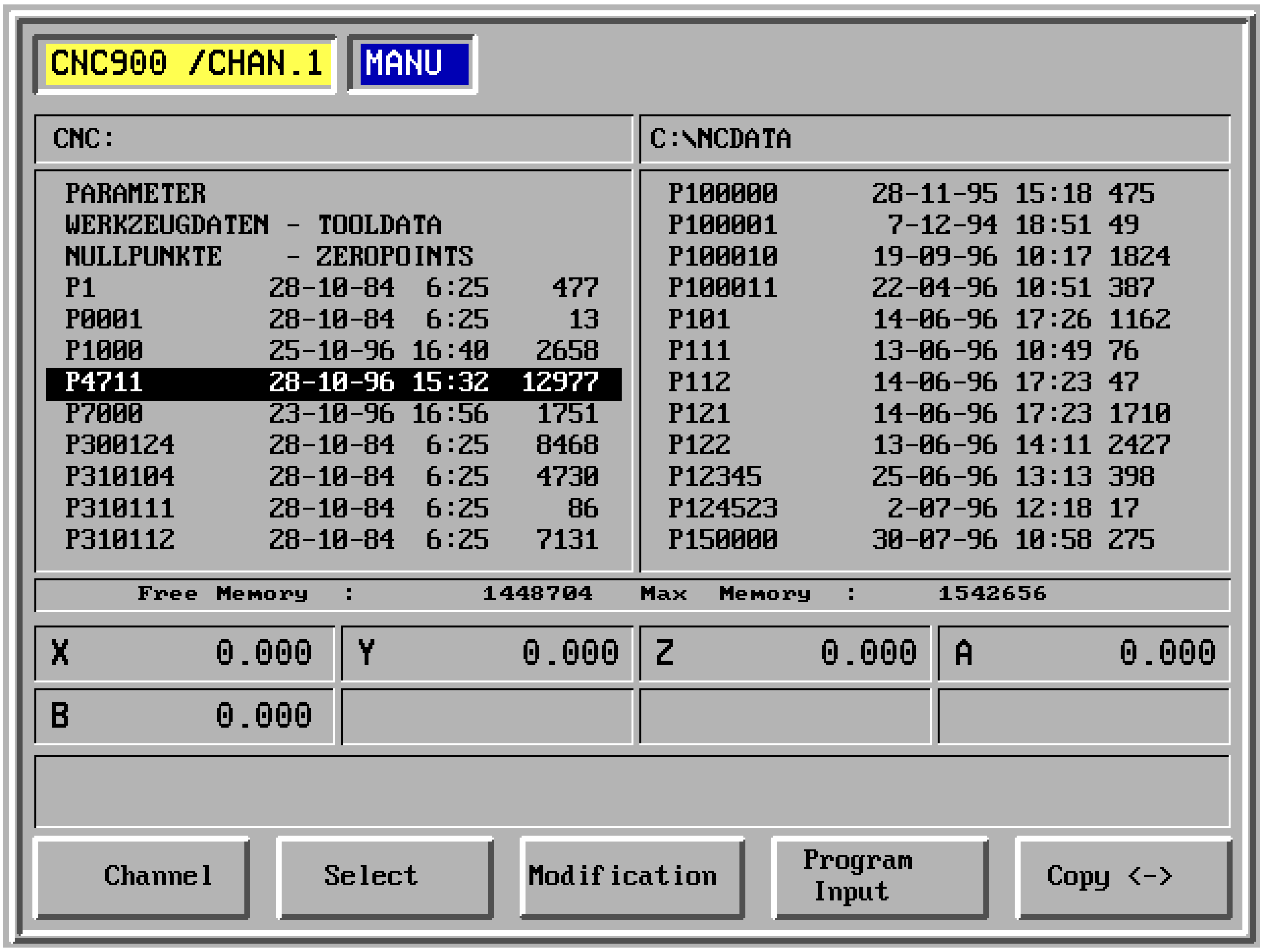
Writing, changing and storing NC programs;
Storing parameter, tool data and zero points.

Menue tree



2.8 NC editor and I/O (continued)

When pressing the key the menue for NC programming and I/O appears.



F1 Channel

F2 Selection

F3 I/O settings

F4 Program input

F5 Copy <->

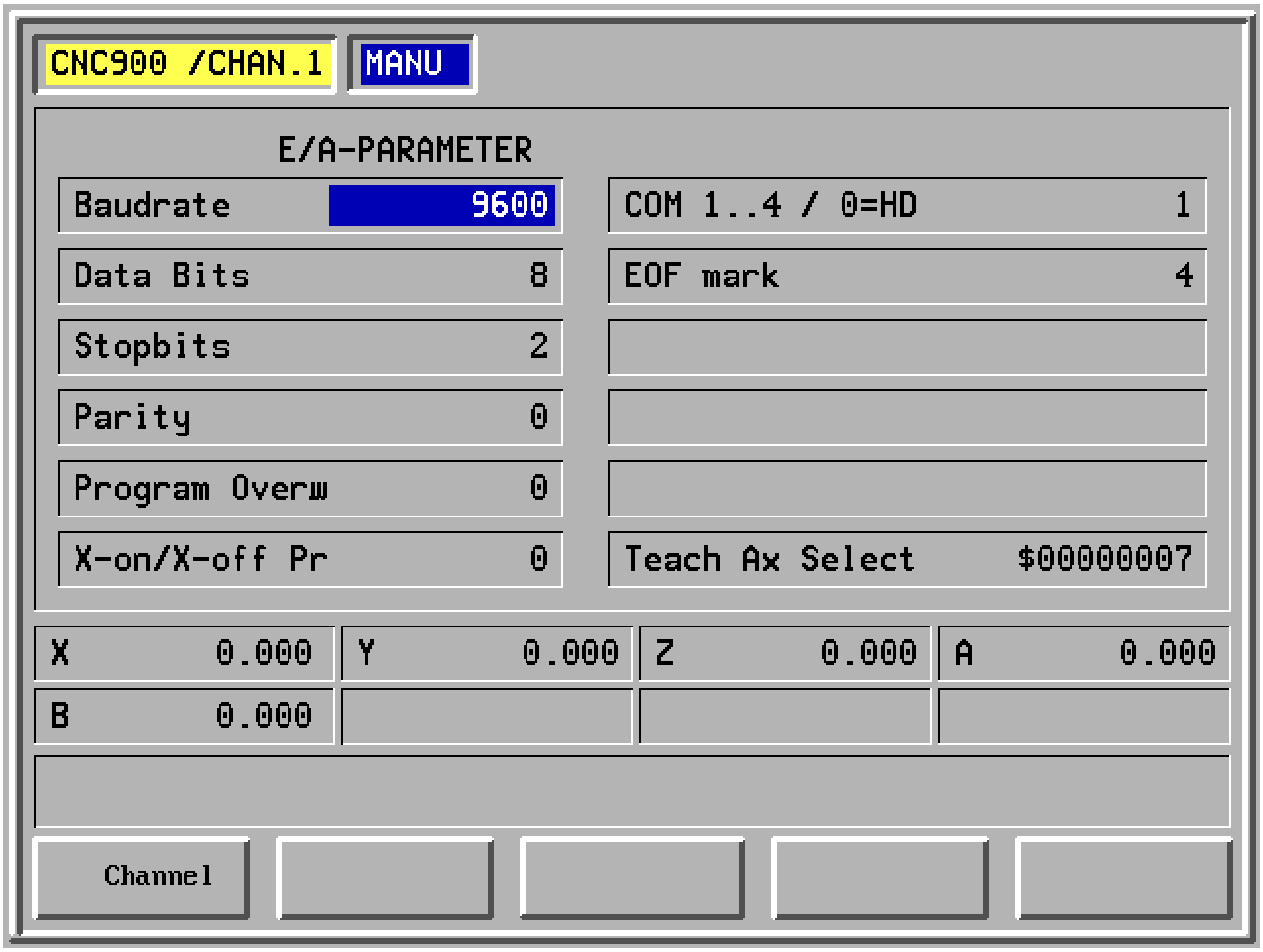
2.8.1 Selection

Selection of storing modes

on the left side	on the right side
CNC:	CNC:
C:\NCDATA	C:\NCDATA
	A:\
	Serial I/O

2.8.2 I/O parameters

This menue can be used for inputing I/O parameters.



F1 Channel

F2-

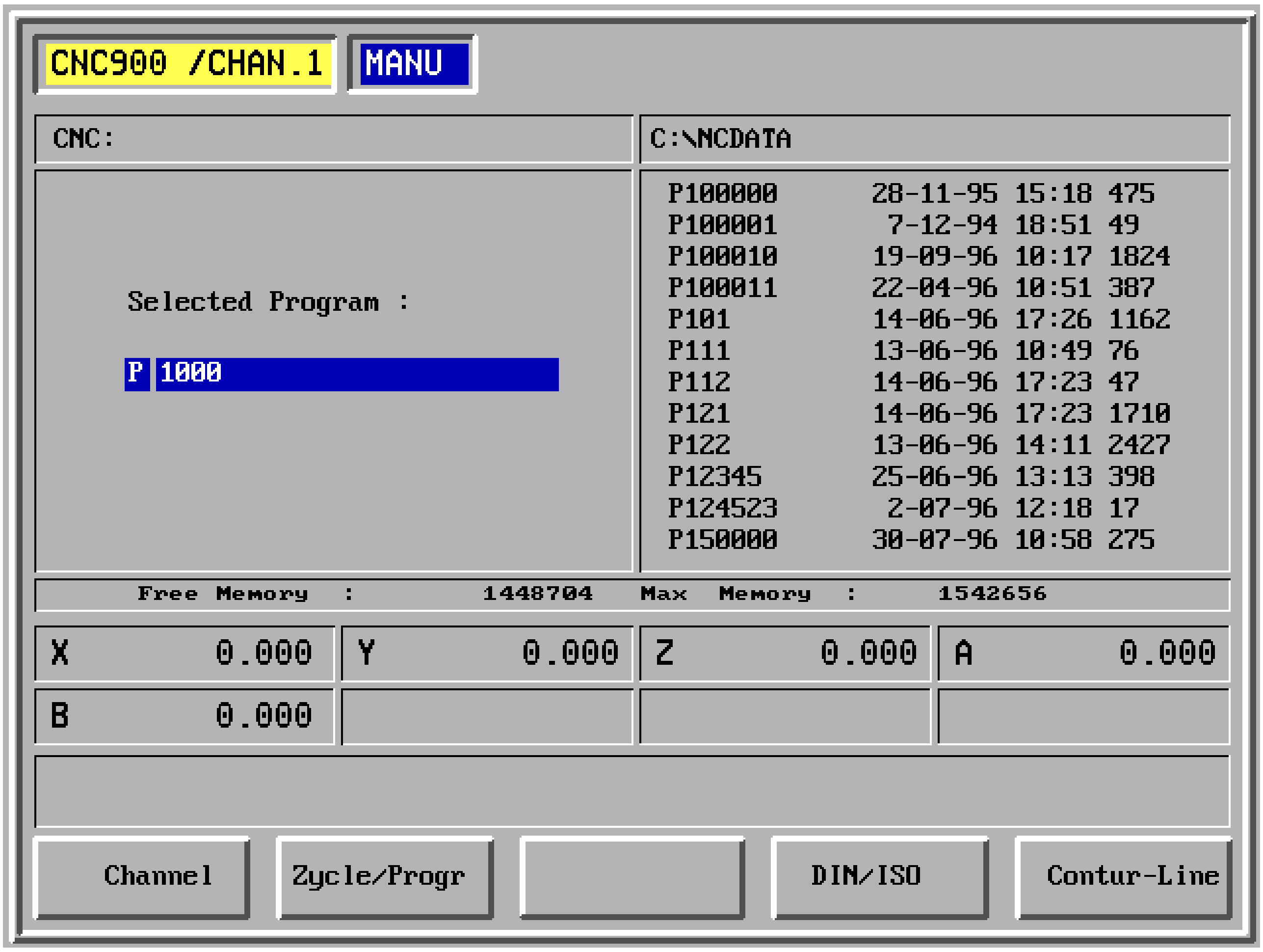
F3-

F4-

F5-

2.8.3 Program input

Program selection appears with pressing key F4 (in screen frame). With the cursor keys an existing program can be chosen or the number of a new program can be input with the numerical keyboard.



F1 Channel

F2 Cycle /program

F3 -

F4 Input a program according to DIN / ISO

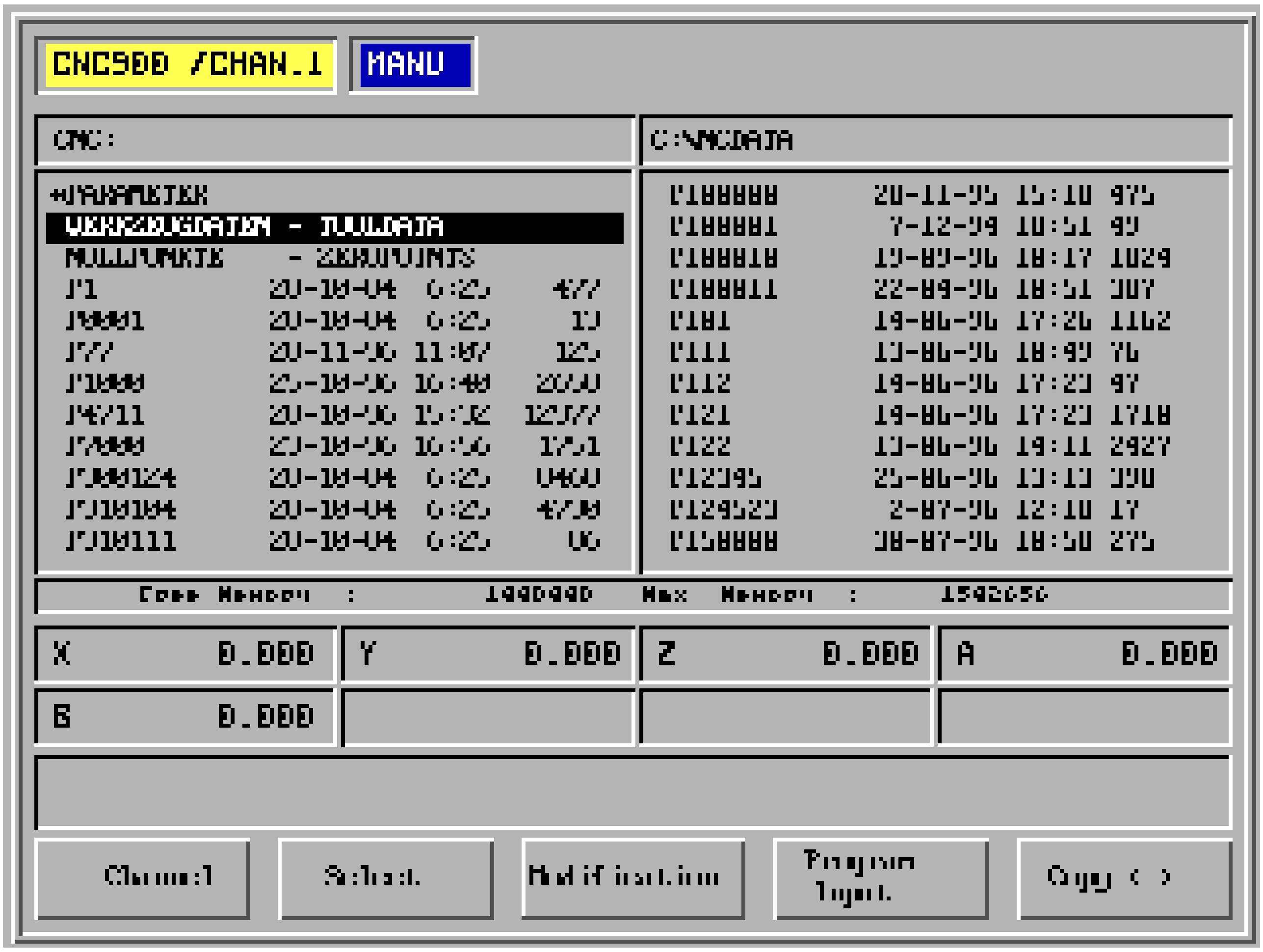
F5 Input a program with outline

2.8.3 Program input (continued)

Program input	paragraph
according to DIN / ISO	2.9
also Teach in	2.10
with graphic support (outline path)	2.11

2.8.4 Copying

Parameters, tool data, zero points and NC programs can be selected with the cursor keys for copying. The selected data are activated with the key and marked with an *.



F1 Channel

F2 Selection

F3 I/O settings

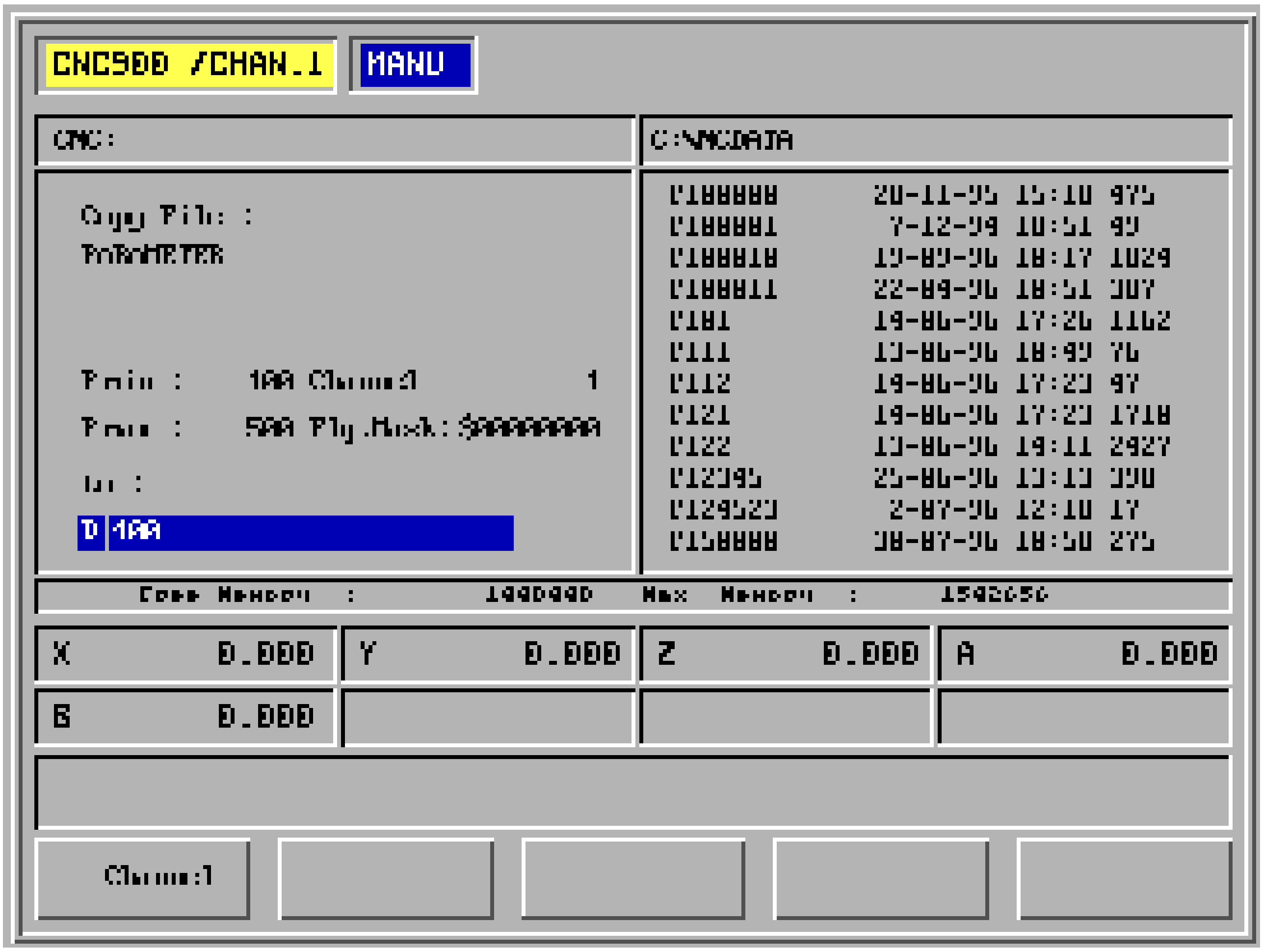
F4 Program input

F5 Copy <->

2.8.4 Copying (continued)

Parameter

After selecting with the cursor keys and after activating with parameters can be copied with F5 from CNC: to C:\NCDATA or A:\ . Input Pmin and Pmax, identification letter D.



F1 Channel

F2 -

F3 -

F4 -

F5 Copy <->

2.8.4 Copying (continued)

Parameter

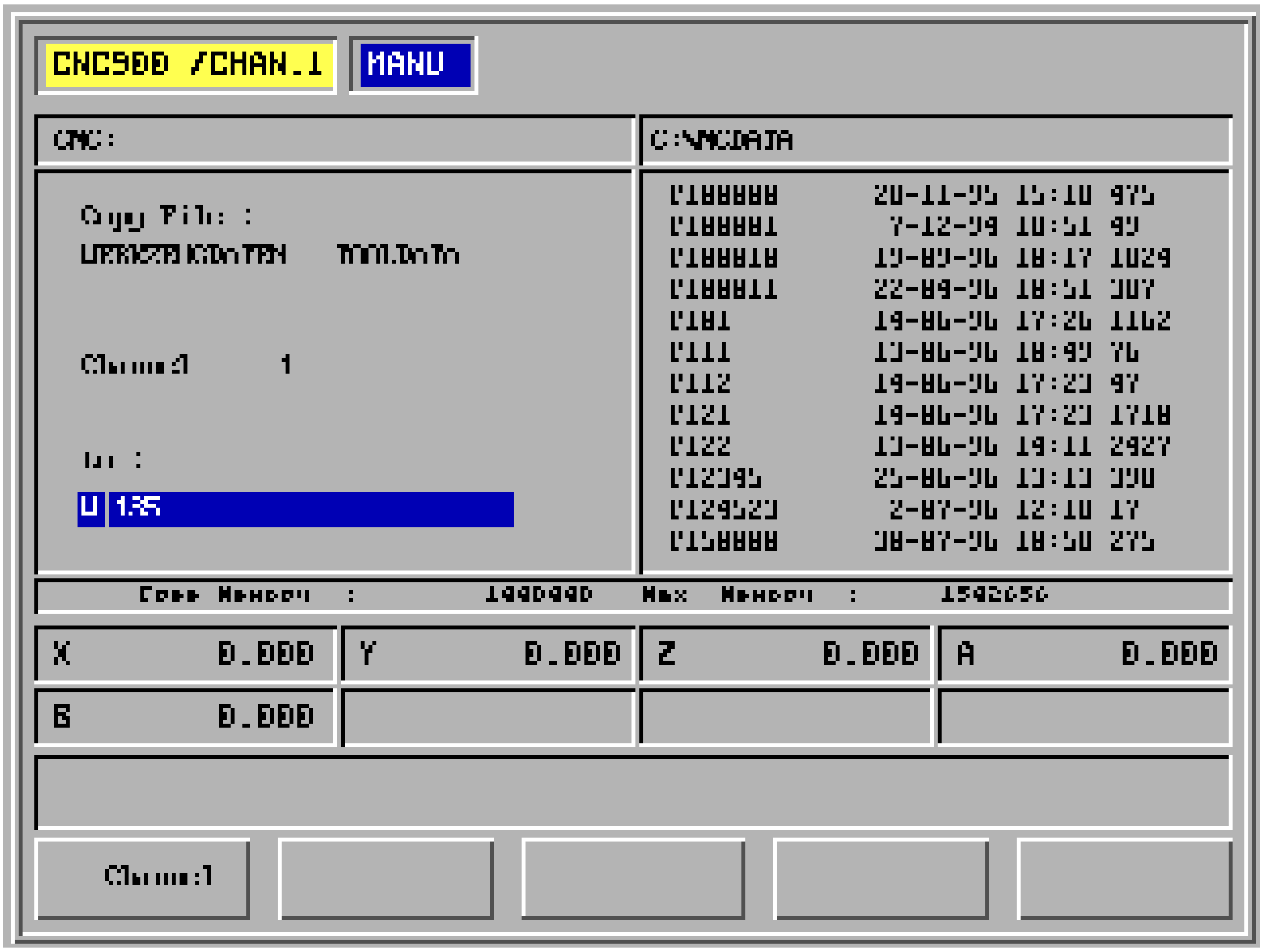
Meaning of the input fields

Pmin:	first parameter of output	
Pmax:	last parameter of output	
Channel	0	Output of q-parameters
	1 to 8	Output of P-parameters of the corresponding channel
Flag mask	Status flag mask	
	0	Output of all parameters according to Pmin - Pmax
	<>0	Output of parameters in the range of Pmin - Pmax, at which the bits are set in the parameter status according to the flag mask. Herewith an output of all parameters in which the EEPROM bit is set, is possible.
to:	DOS file name	

2.8.4 Copying (continued)

Tool data

After selecting with the cursor keys and after activating with tool data can be copied with F5 from CNC: to C:\NCDATA or A:\ . Identification letter W.



F1 Channel

F2 -

F3 -

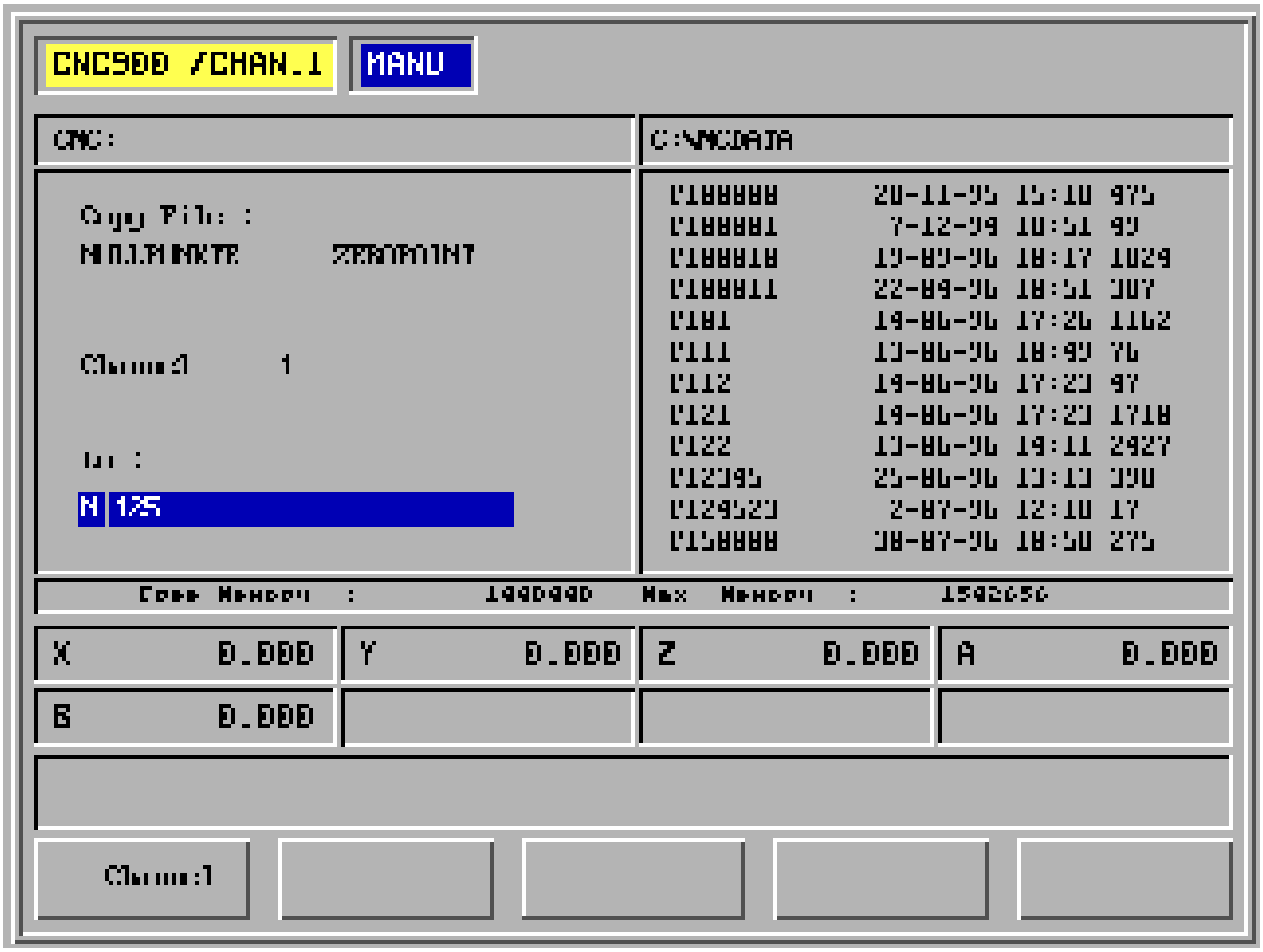
F4 -

F5 Copy <->

2.8.4 Copying (continued)

Zero points

After selecting with the cursor keys and after activating with zero points can be copied with F5 from CNC: to C:\NCDATA or A:\ . Identification letter N.



F1 Channel

F2 -

F3 -

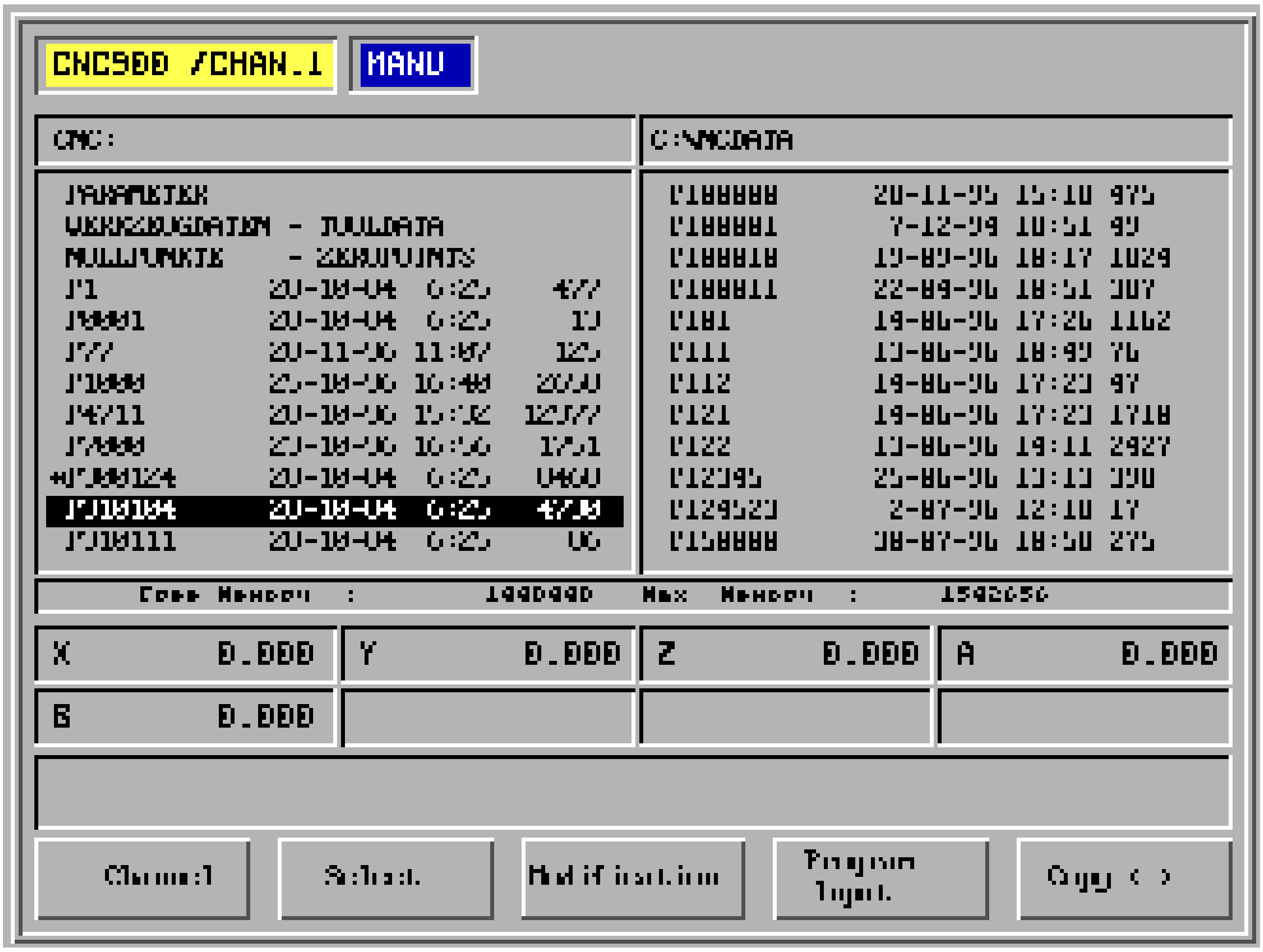
F4 -

F5 Copy <->

2.8.4 Copying (continued)

NC programs

After selecting with the cursor keys and after activating with , the selected NC program is marked with *.



F1 Channel

F2 Selection

F3 I/O settings

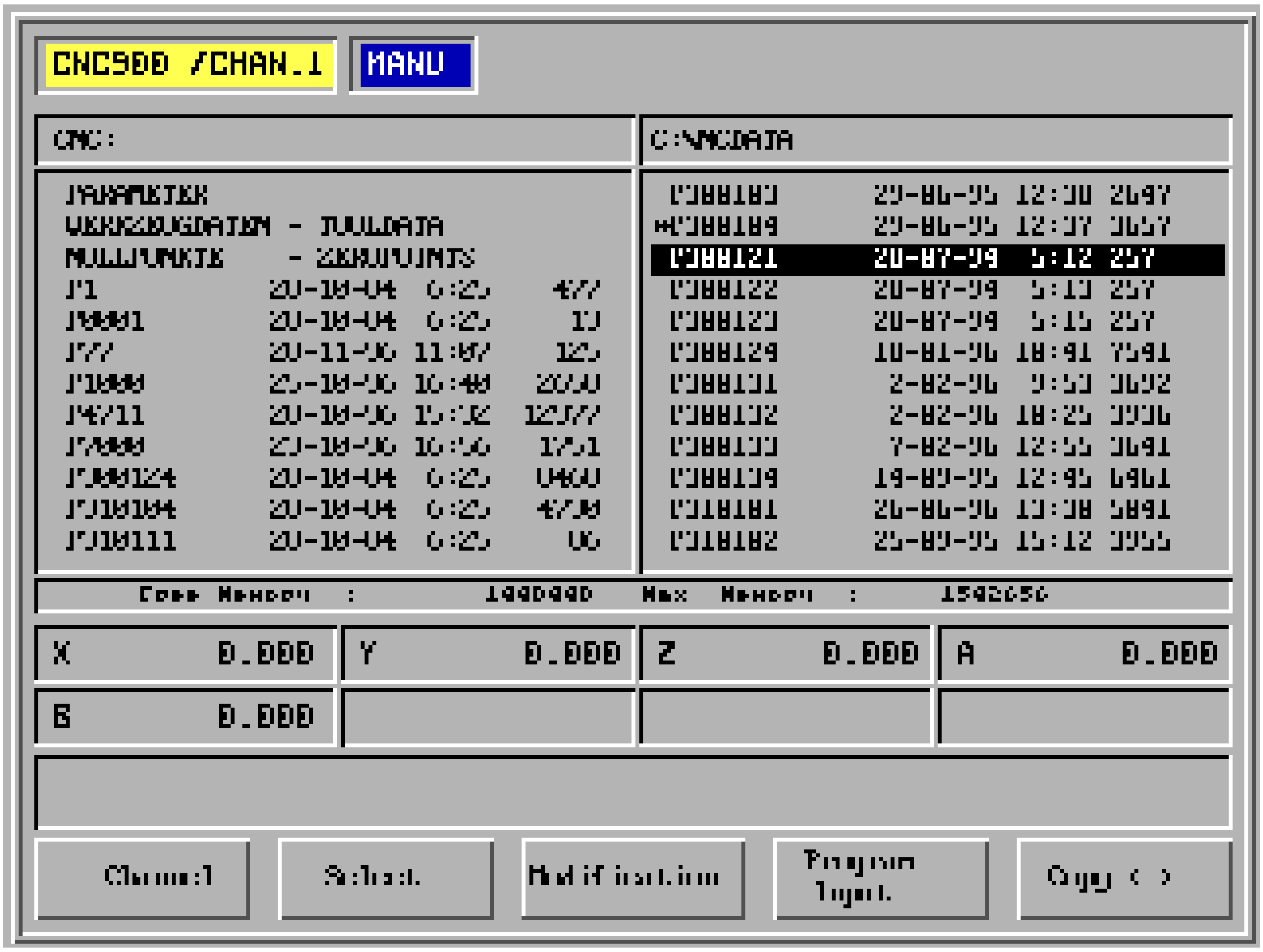
F4 Program input

F5 Copy <->

2.8.4 Copying (continued)

NC programs

After selecting with the cursor keys and after activating with tool daNC programs can be copied with F5 from CNC: to C:\NCDATA or A:\ . Identification letter P.



F1 Channel

F2 Selection

F3 I/O settings

F4 Program input

F5 Copy <->

2.8.4 Copying (continued)

NC programs

All programs mark

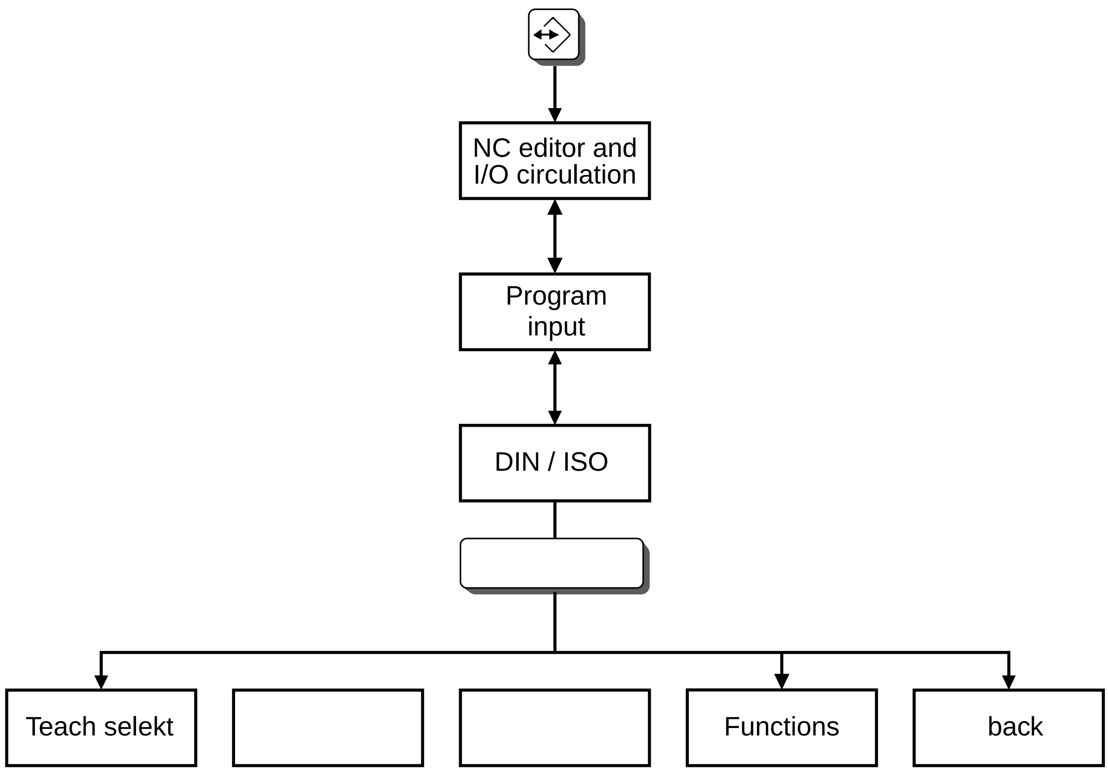
F6 and inserting branch press.

All programs off NC memory copy
(store total NC memory in a file)

Programs mark and Funktionstate F5 (copy) press.

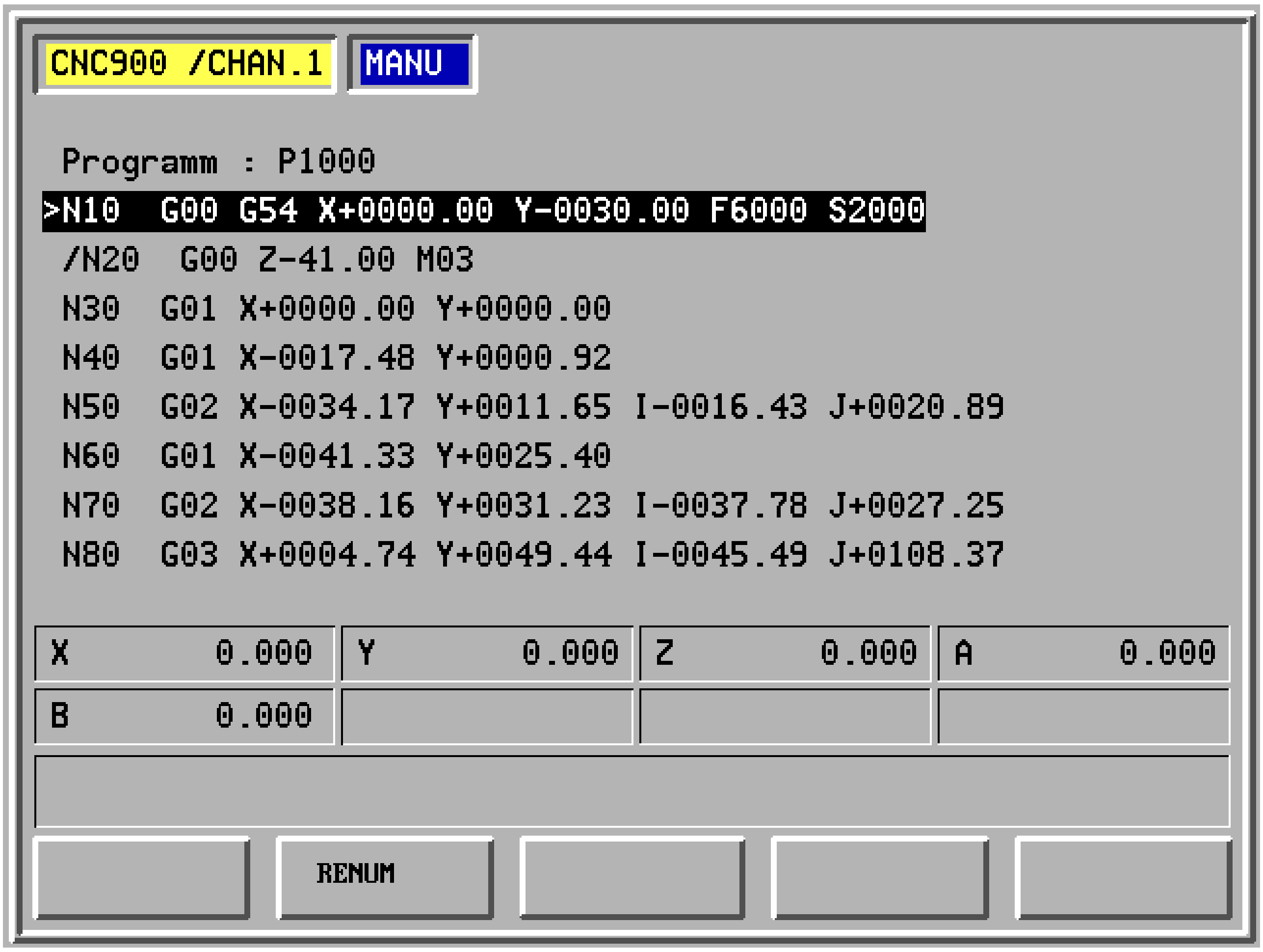
2.9 Input a program according to DIN / ISO

Menue tree



2.9 Input a program according to DIN / ISO (continued)

After inputing the program number and after pressing the key DIN / ISO the program appears with the first blocks in the display, if a program is existing with the indicated number. If not, only the program number and >N10 appears.



F1 -

F2 Renumber

F3 -

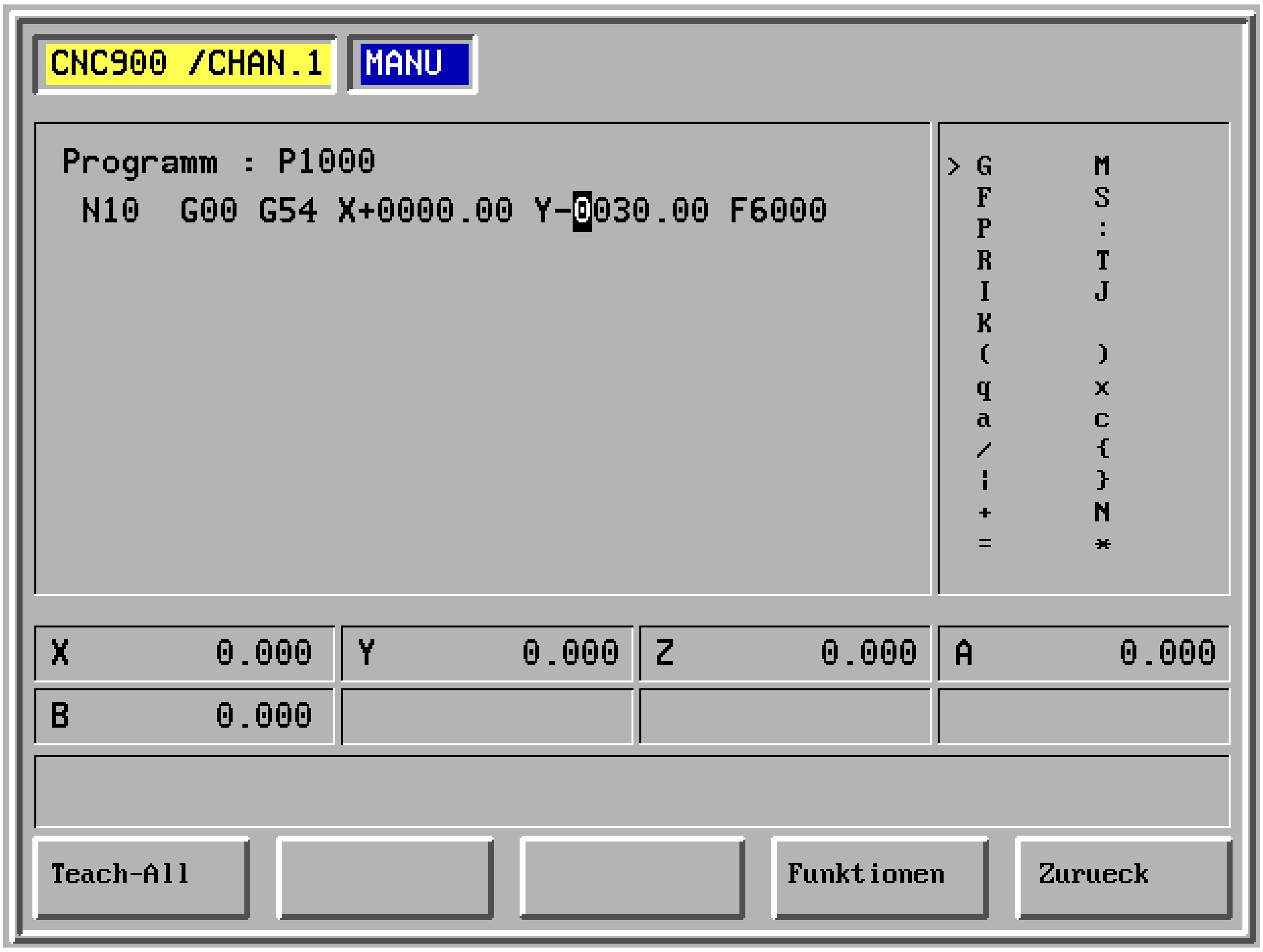
f4 -

F5 -

2.9 Input a program according to DIN / ISO (continued)

Changing or inputing blocks

When pressing ↵ the selected block and the function appear.



F1 Teach-Select

F2 -

F3 -

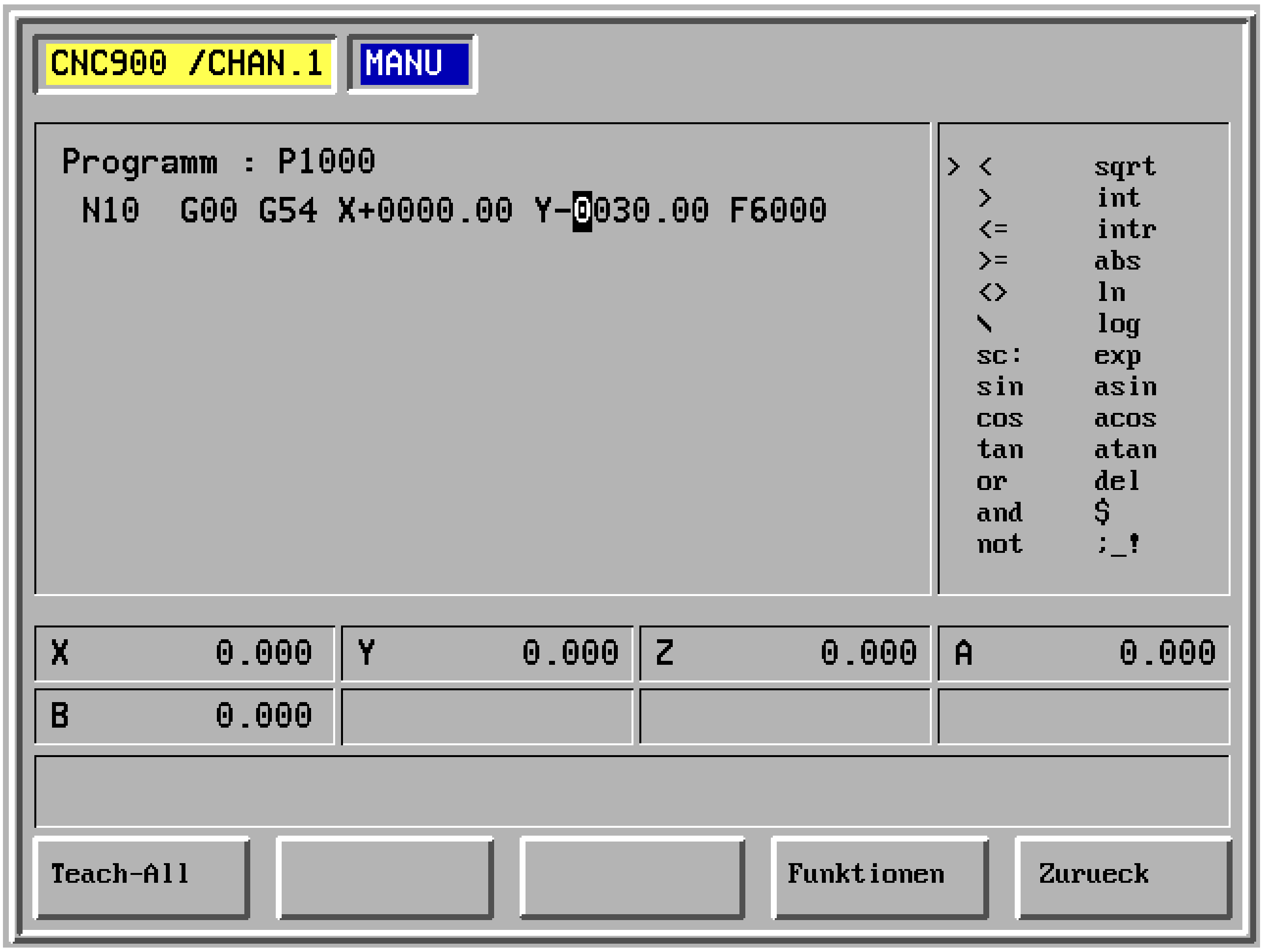
F4 Functions Switching on further programming functions

F5 Back

2.9 Input a program according to DIN / ISO (continued)

Functions

When pressing the functions are activated. Selection with cursor keys.



F1 Teach-Select

F2 -

F3 -

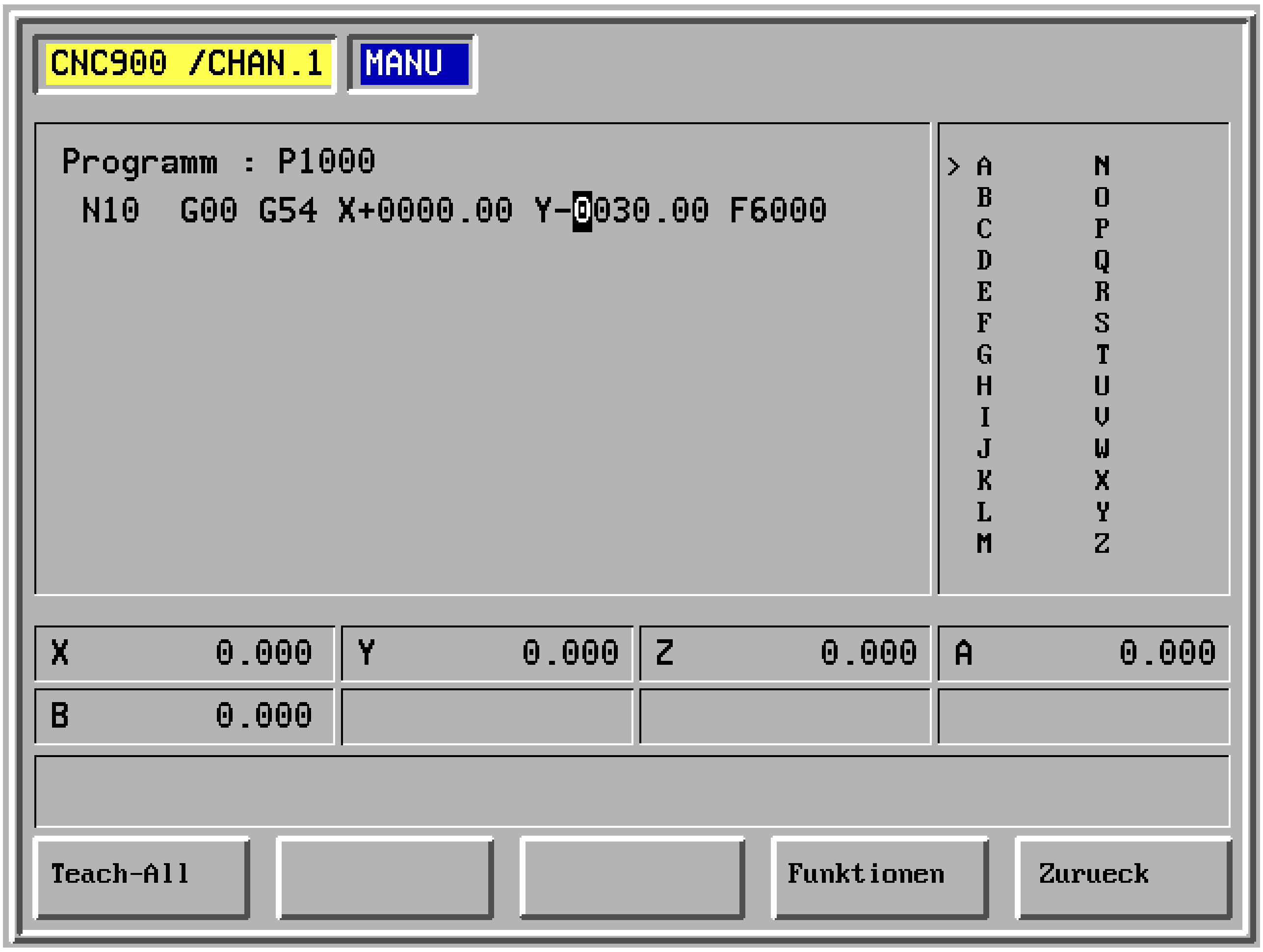
F4 Functions Switching on further programming functions

F5 Back

2.9 Input a program according to DIN / ISO (continued)

Functions

When pressing XXX another function siede appears.



F1 Teach-Select

F2 -

F3 -

F4 Functions Switching on further programming functions

F5 Back

2.10 Preparing a NC program in Teach mode

2.10.1 Setting zero points

The parameter P11804 (tool carrier - length) must be loaded with the right values and the tool dimension must be active.

Drive axes in the desired zero point position.

Call up function „**Command=Actual**“ and store with **‘Enter’**.
Herewith the current actual values of the axes are transmitted to the zero point memory.

In this way different zero pointes can be set.

If these zero point shifts should be active in manual operation mode, the parameter P8758 has to be loaded with the desired value (G54 to G59).

In automatic mode, the call up is made with the functions G54...G59.

Remark:
The zero point shifts are only effective in the tool coordinate system (G48) or in the workpiece coordinate system (G49).

2.10.2 Enter a program with “Teaching”

It is advisable, to mark the workpiece with all known or determining „Teach-points“.
This facilitates later a fast discovering of the individual NC blocks, to insert in the program certain data and/or functions.

A further help would be, if the stored „Teach-points” would be written in a list with the corresponding block number, e.g. Point 5 = block no. 80.

The stored zero point shift, on which the NC program refers, can be activated with parameter P382.

Select mode of operation „Positioning“.
Approach zero point position with a positioning block,
e..g. N10 G0 G55 X0 Y0 Z0 A0 B15

The mode of coordinate, in which the command / actual data were stored (P8751), must be inserted in the block over the corresponding G - function (G48, G49).

Select manual operation mode:
Enter and store program number and the corresponding functions and technological data in the designated NC blocks,

e.g.	N10	T1	M16					
	N20	G55	G49	FOR. ..	S	X	Y..	Z.. ... ..

With „continuous drive“ or „step drive“ the desired position is approached with all axes.
If all axes are in their anticipated position, the position is stored with the function ‘Command=Actual’ and ‘Enter and transmitted to the indicated block.

e.g.	N30	X	Y..	Z..	A..	C.. ... ..

The next Teach - points are started and stored likewise.

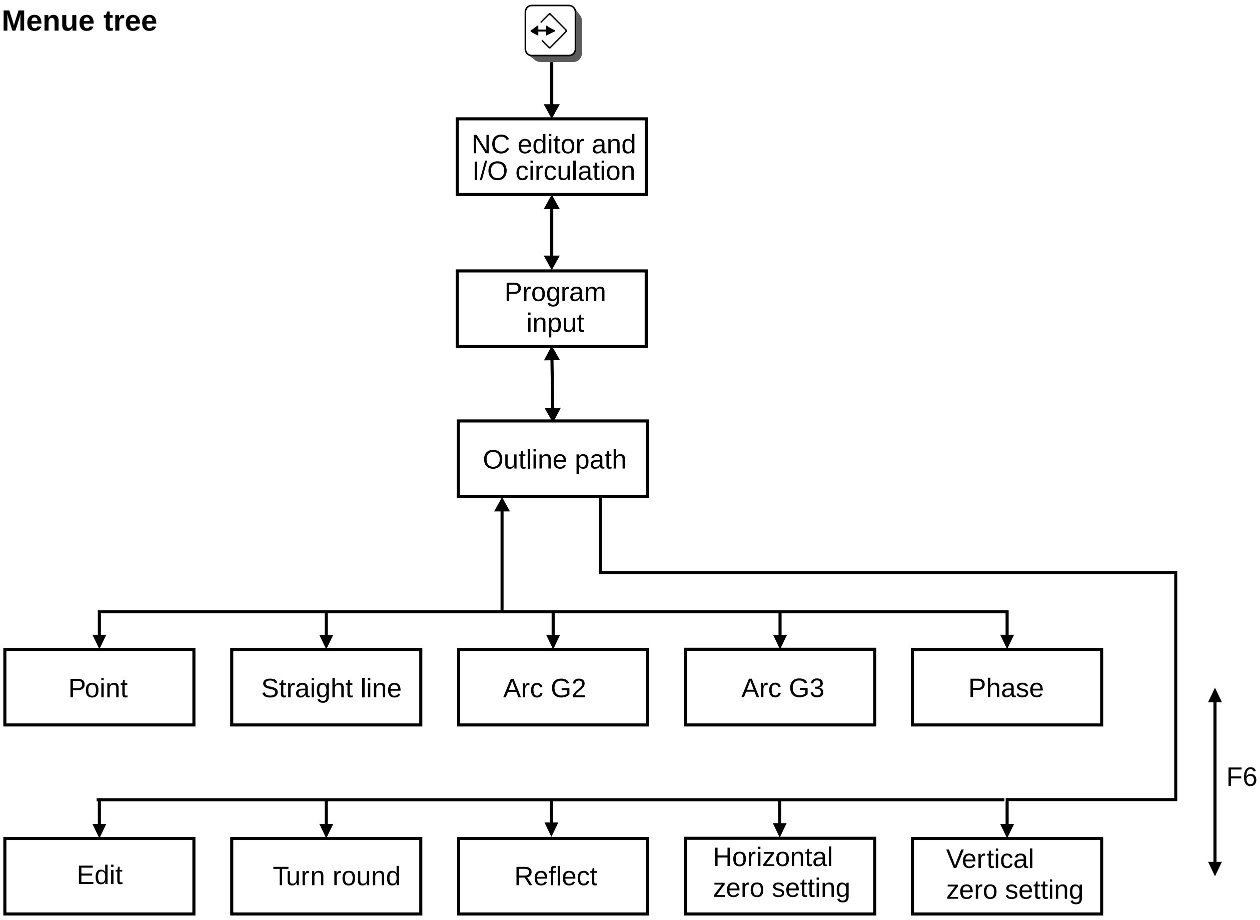
Subsequently the program is optimizeed by inserting feedrate, number of revolutions etc.

After reviewing the program and after a test run, the program is finished.

2.11 Program input with graphic support (outline path)

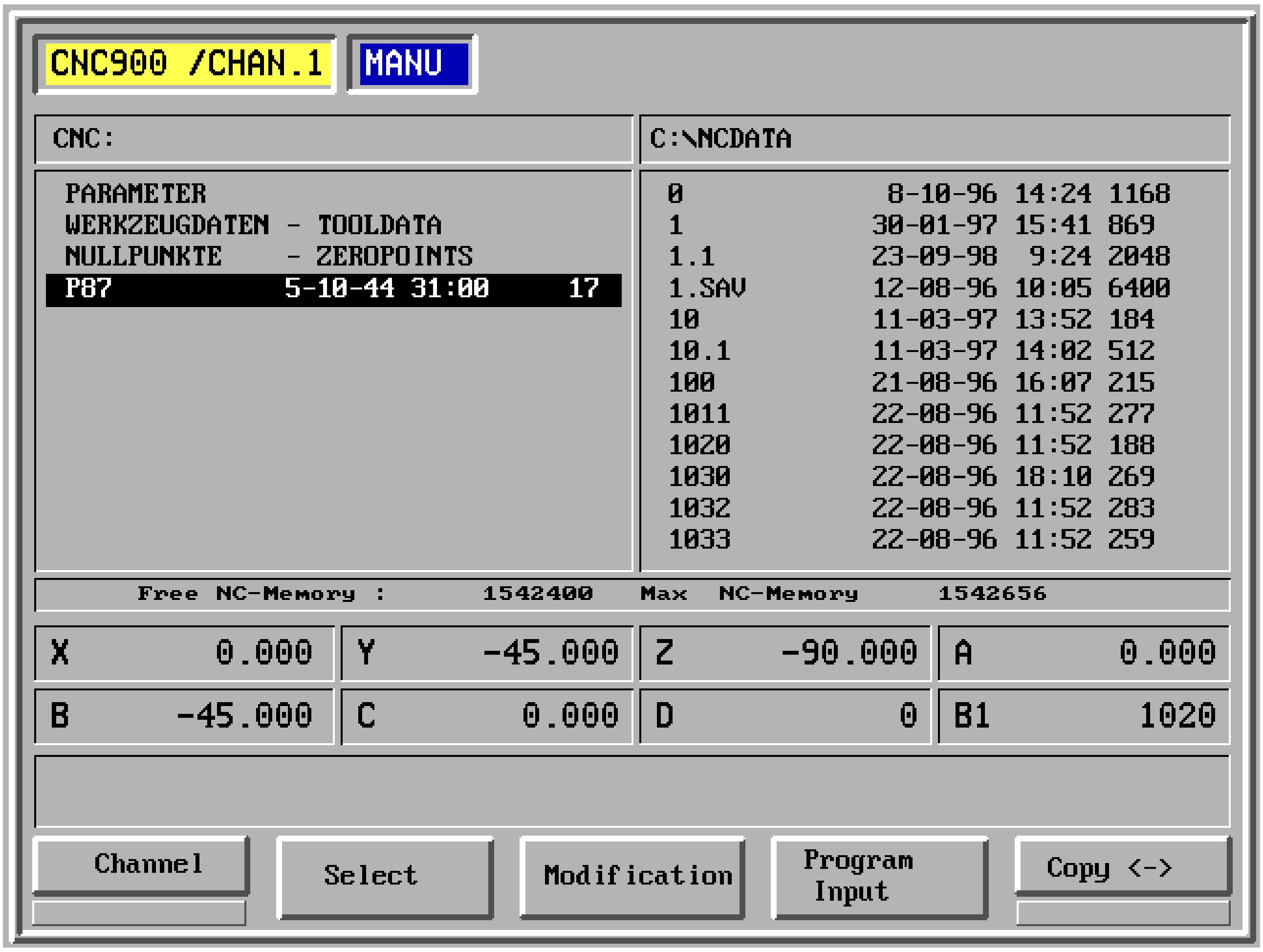
2.11.1 General

Menue tree



2.11.1 General (continued)

With pushing from key the menu for NC programming seems.



F1 Channel

F2 Selection

F3 Adjustments

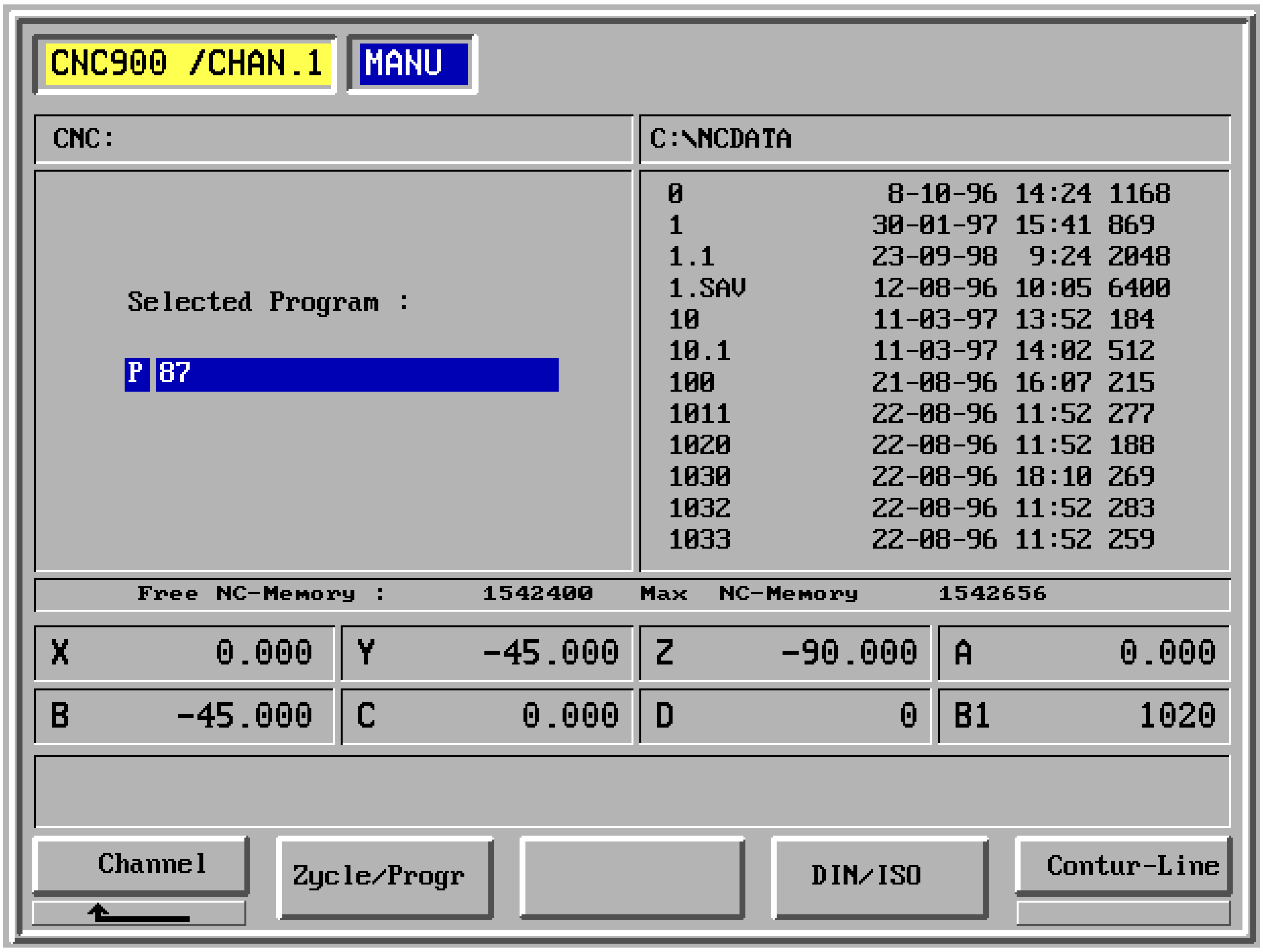
F4 Program input

F5 Copy <->

2.11.1 General (continued)

Program input

With pushing of key F4 (within the display frame) seems the program selection. The paragraph of a new program can be input with the numerical keyboard.



F1 Channel

F2 Cycle/program

F3 -

F4 Program input according to DIN / ISO

F5 Program input with outline path

2.11.1 General (continued)

With the cursor keys an existing program can be chosen or the number of a new program can be input with the numerical keyboard. After pressing the key F5, the picture with the coordinates appears.

With the keys (page up) and (page down), the picture can be enlarged or reduced (zoom function) for a better view.

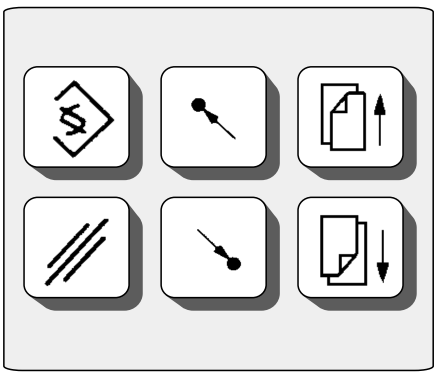
New outlines can be inserted continuously whereby a blue point is indicating the inserting place.

With the keys (Pos1) and (End) NC blocks can be moved forward and backward, they do then appear in red. Those moved NC blocks can be inserted or deleted with the keys (Change) or (Delete).

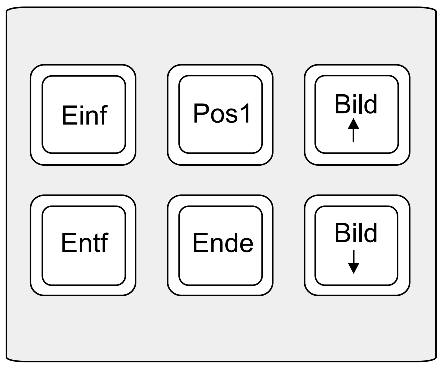
The block numbers are automatically numbered continuously (increased).

Max. 100 blocks can be programmed in a program with GPE.

Programming is finished and the program is stored with F7.



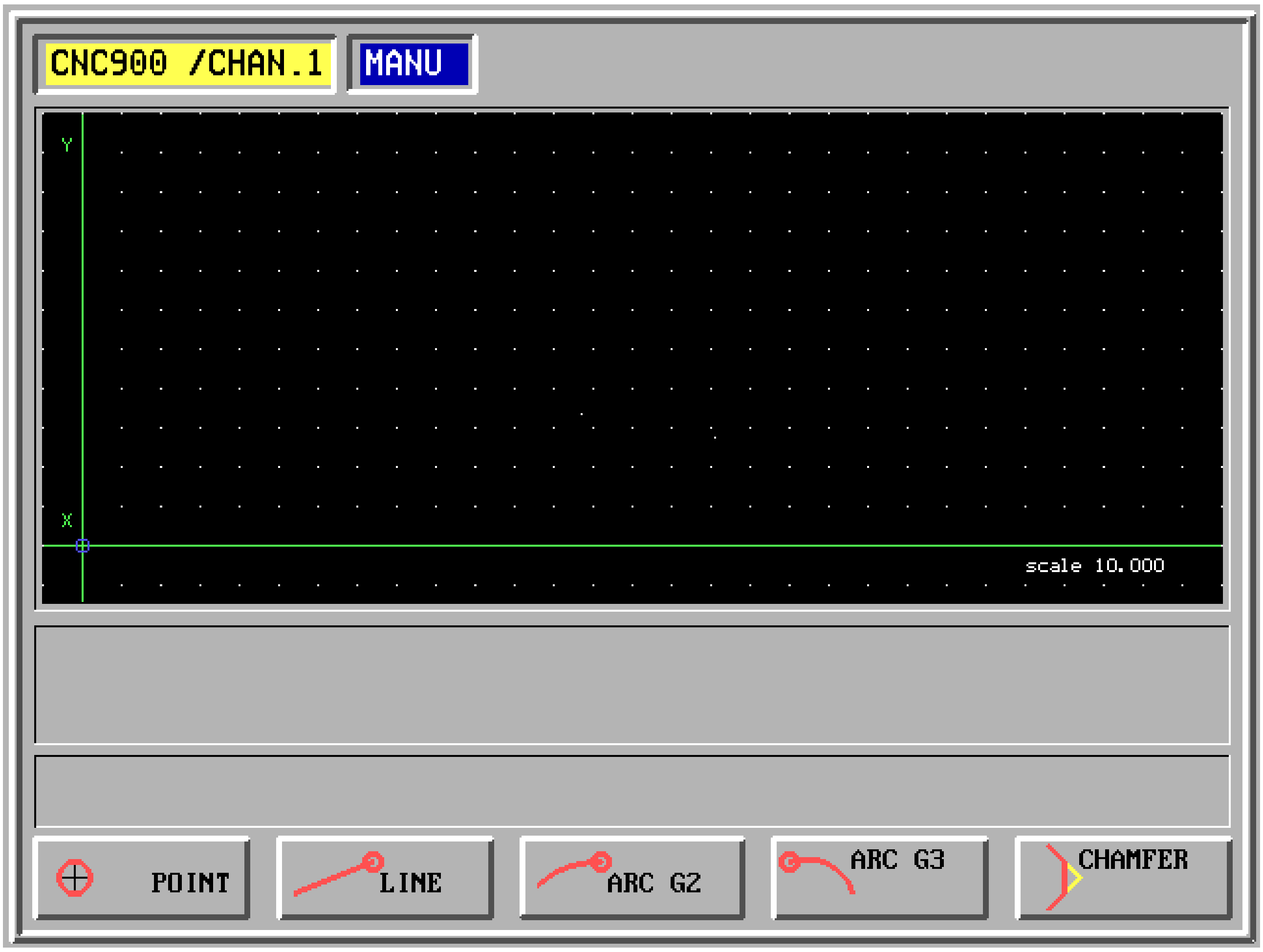
Key field operating panel



and suitable keys on the PC

2.11.1 General (continued)

Basic menu 1 (switch with F6)



F1 Insert point

F2 Insert straight line

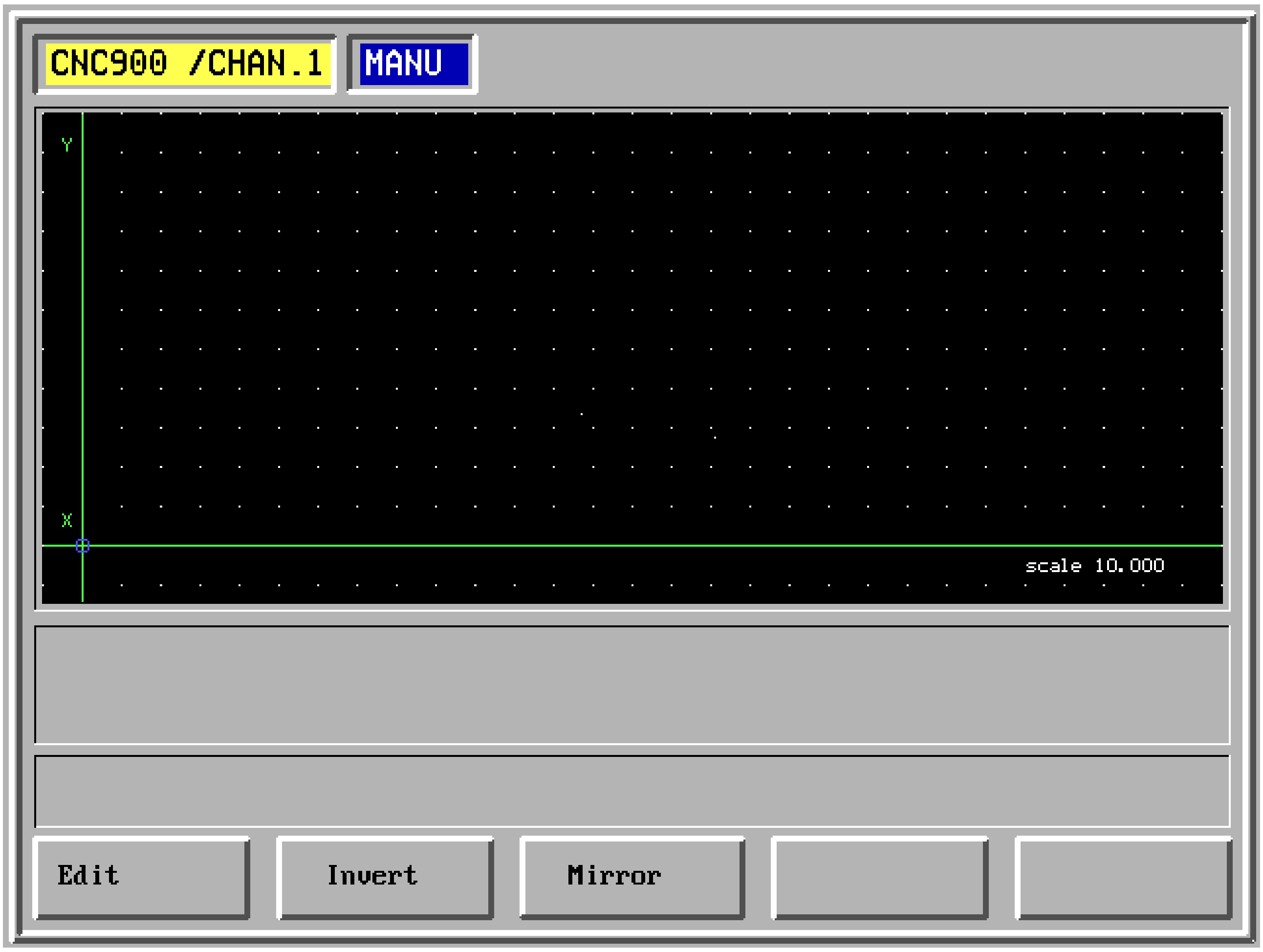
F3 Arc G2 (clockwise)

F4 Arc G3 (counter-clockwise)

F5 Insert phase and rounding

2.11.1 General (continued)

Basic menu 2 (switch with F6)



F1 Program edit with full function range (F, G, M etc.),
if a ASCII keyboard available actual

F2 Processing direction turn around

F5 Outline reflect

F4 Horizontal zero setting
F5 Vertically zero setting

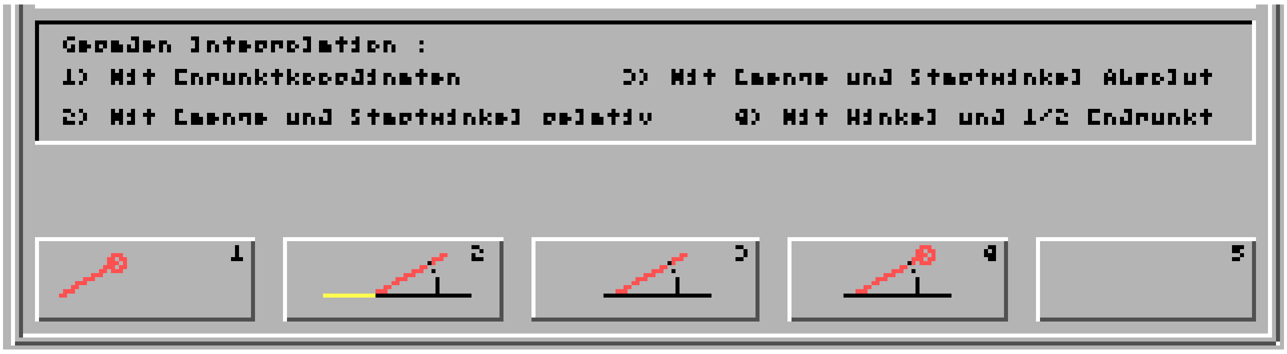
2.11.2 Inserting a point

With the numerical keyboard the coordinates of one point can be input and inserted with key .

The input point is started with rapid traverse (G00).

2.11.3 Insert a straight line

With the numerical keyboard the coordinates of a straight line can be input and inserted with key . One sets always at the blue point.



Selection of the different types of the straight line generation:

F1 Linear interpolation with Input of the terminator point coordinates

F2 Linear interpolation with Input of length and start angle relative

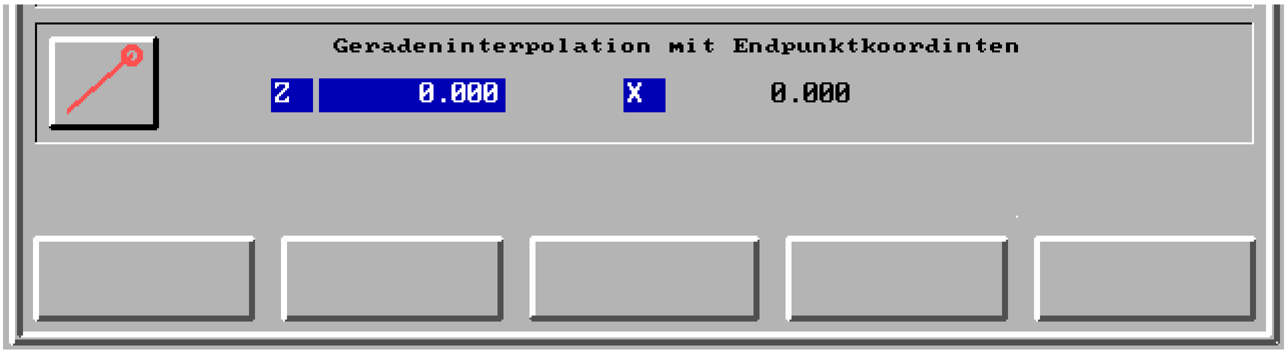
F3 Linear interpolation with Input of length and start angle absolutely

F4 Linear interpolation with Input of angle and 1/2 terminator point,
i.e. that only one of the coordinates must be input.
The coordinate input last is taken over.

2.11.3 Insert a straight line (continued)

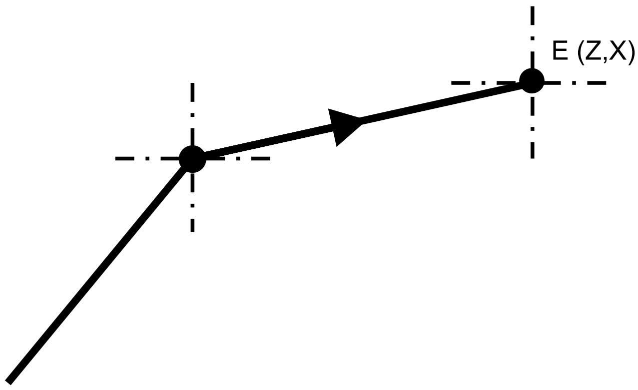
linear interpolation with Input of the terminator point coordinates

The terminator point coordinates input from the start point to a linear (G1) is inserted.



Linear interpolation with the terminator point coordinates (E) Z and X

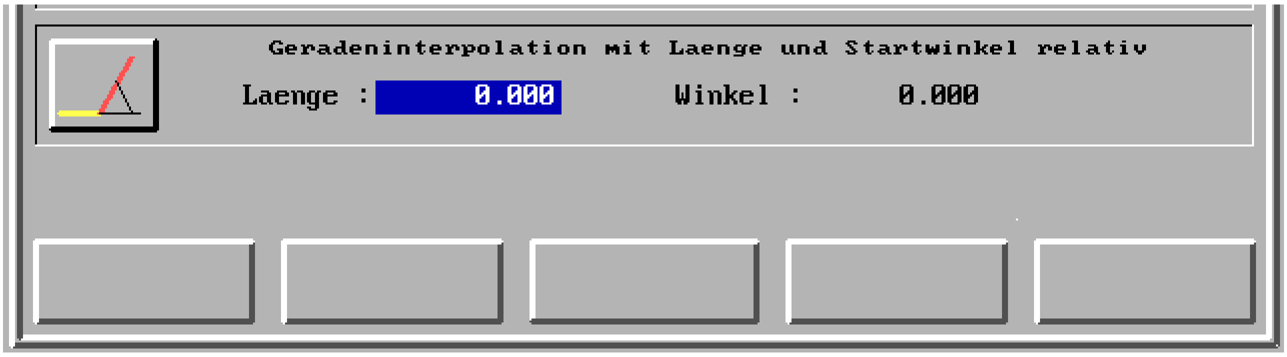
Example



2.11.3 Insert a straight line (continued)

linear interpolation with Input of length and start angle relative

A linear with length and start angle is relatively inserted by the start point.

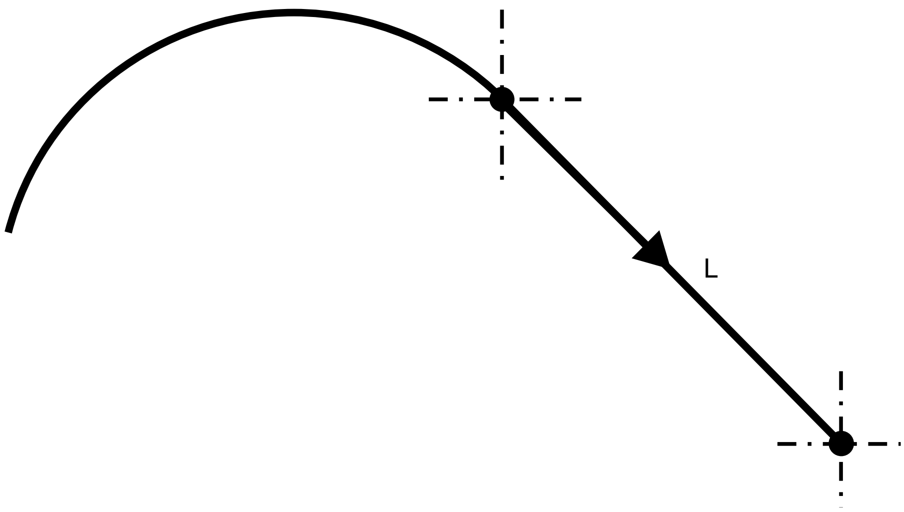


Linear interpolation
with length (l) and
Start angle (SW) relative
to the preceding block

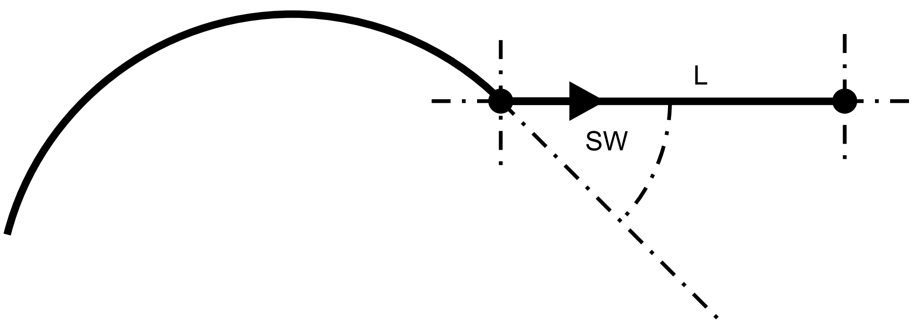
Examples:

Angle relative 0°

i.e. tangential
to the preceding block



Angle relative 45°
to the preceding block



2.11.3 Insert a straight line (continued)

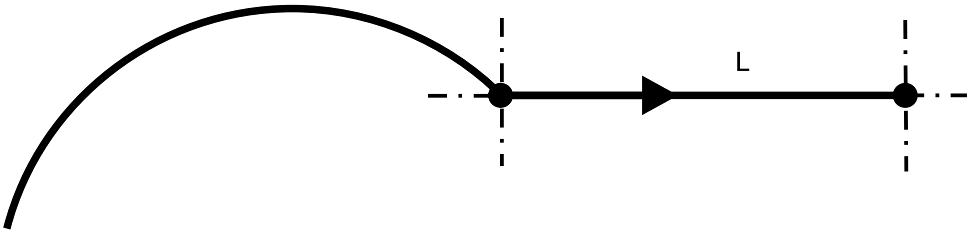
linear interpolation with Input of length and start angle absolutely

A linear with length and start angle is absolutely inserted by the start point.

Linear interpolation
with length (l) and final angle (EW) absolutely

Example:

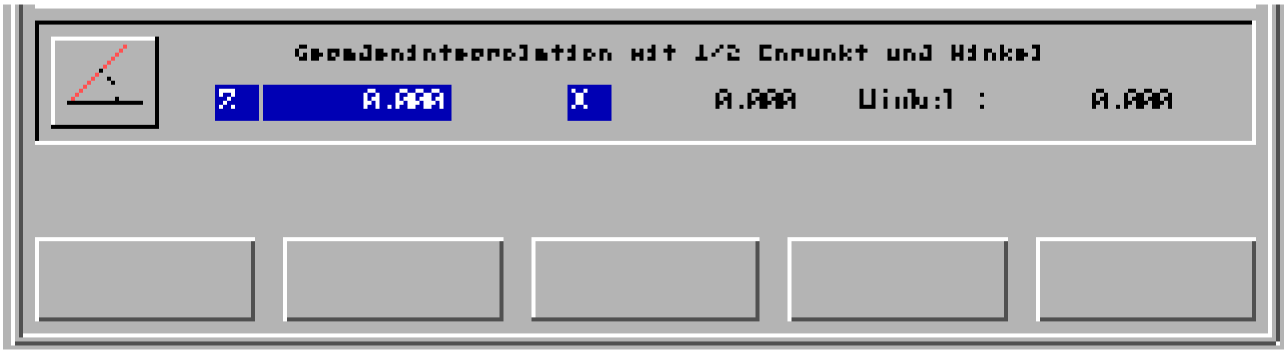
Angle absolutely 0°



2.11.3 Insert a straight line (continued)

linear interpolation with Input of final angle absolutely and 1/2 terminator point

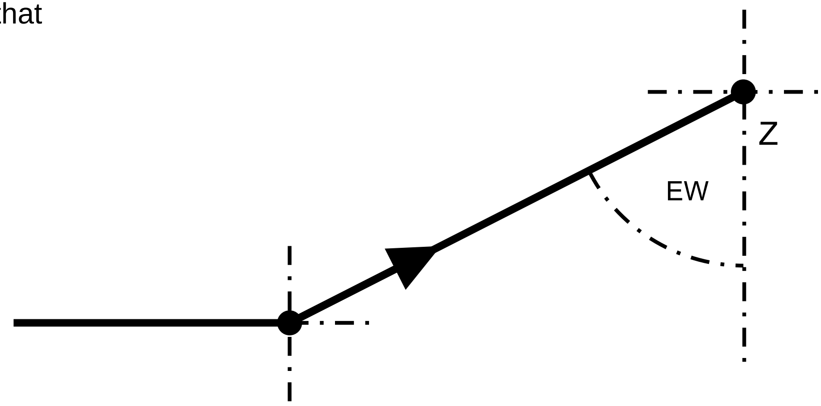
The terminator point coordinates with 1/2 terminator point and final angle, input from the start point to, absolutely a linear is inserted.



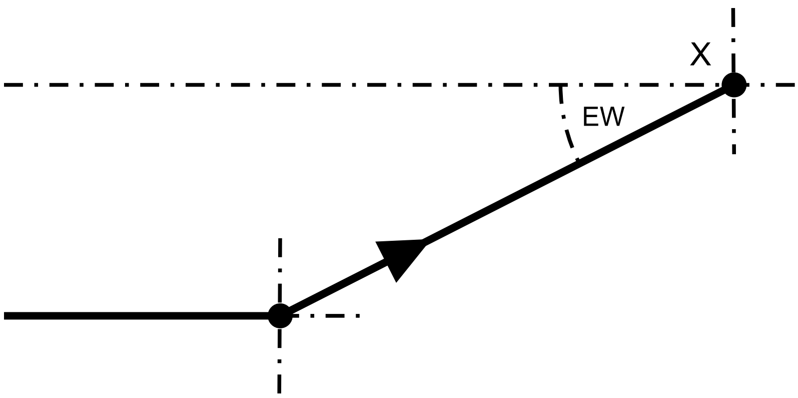
Linear interpolation
with 1/2 terminator point and final angle (EW) absolutely
The indicated angle actual of the final angles
the programmed straight lines with that
indicated direction in Z or X.

Examples:

Final angle absolutely 60° to Z



Final angle absolutely 30° to X



2.11.4 Inserting an arc

With the numerical keyboard the coordinates of an arc can be input and inserted with key .
One sets always at the blue point.

F1 Circular interpolation with terminator point coordinates and radius

F2 Circular interpolation with terminator point and centre point

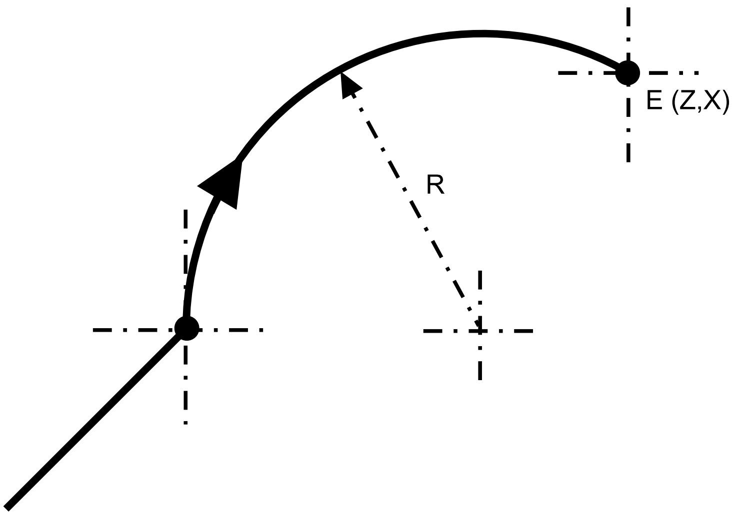
F3 Circular interpolation with radius and final angle

F4 Circular interpolation with radius and 1/2 terminator point
i.e. that only one of the coordinates must be input.
The coordinate input last is taken over.

2.11.4 Inserting an arc (continued)

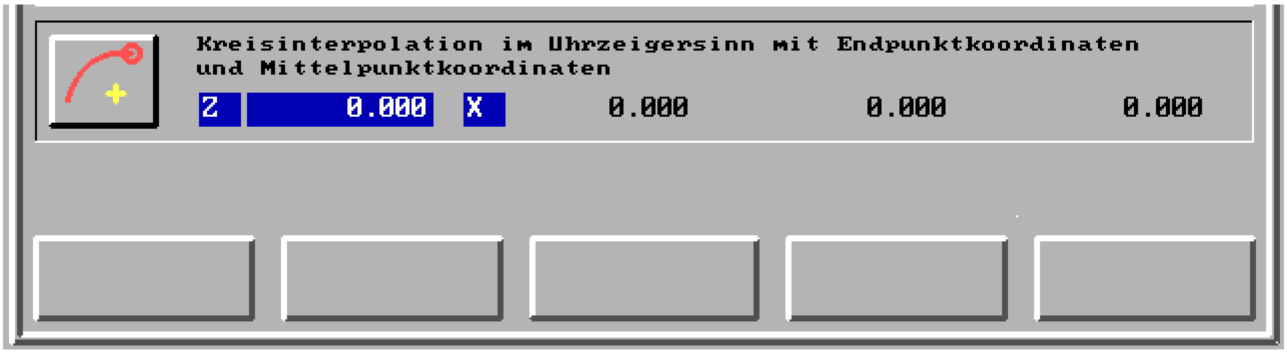
Circular interpolation with terminator point coordinates and radius

Circular interpolation in the clockwise direction
with terminator point coordinates (E) Z and X and radius (r)

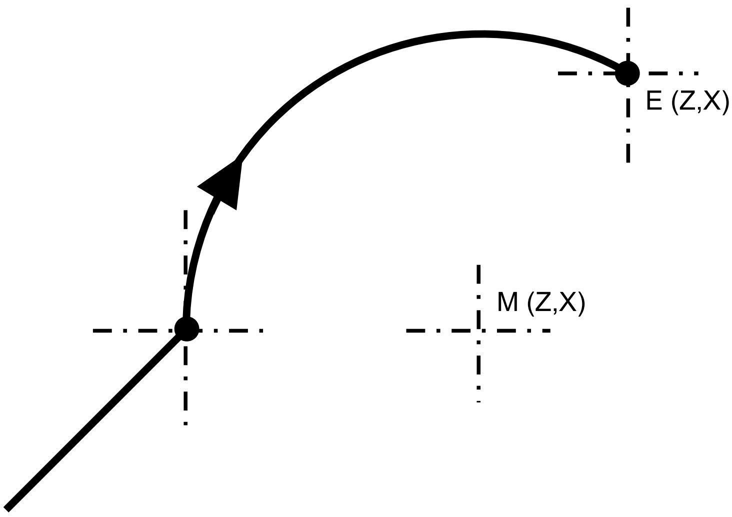


2.11.4 Inserting an arc (continued)

Circular interpolation with terminator point and centre point

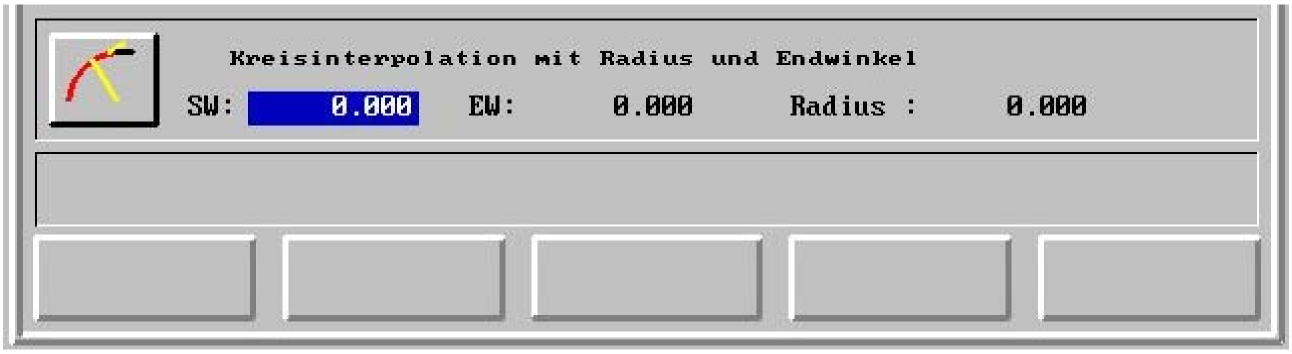


Circular interpolation in the clockwise direction
with terminator point coordinates (E) Z and X
and Mittelpunkkoordinaten (m) Z and X

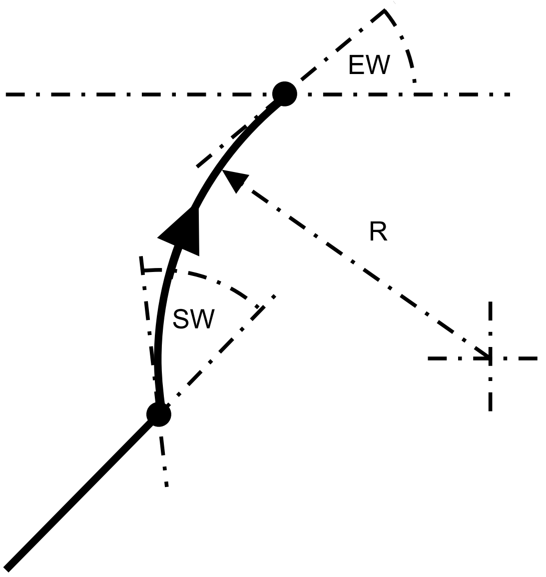


2.11.4 Inserting an arc (continued)

Circular interpolation with radius and final angle

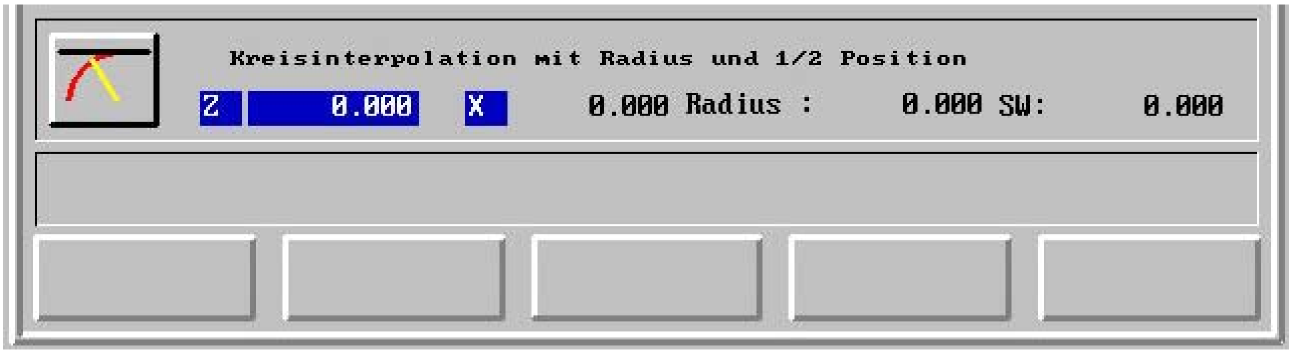


Circular interpolation in the clockwise direction
with radius (r) and start angle (SW) and final angle (EW)

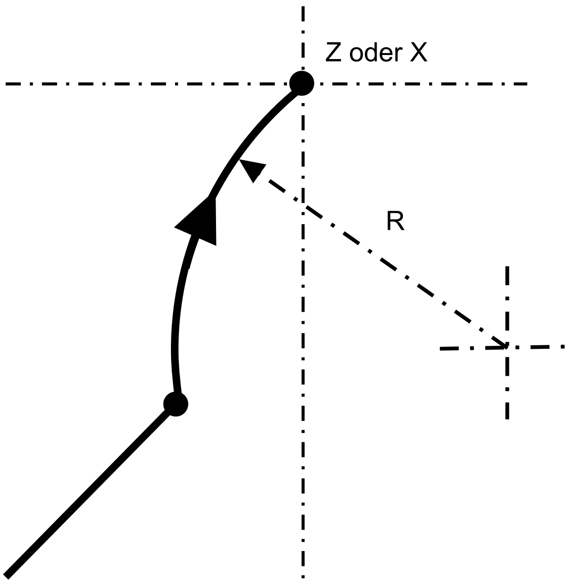


2.11.4 Inserting an arc (continued)

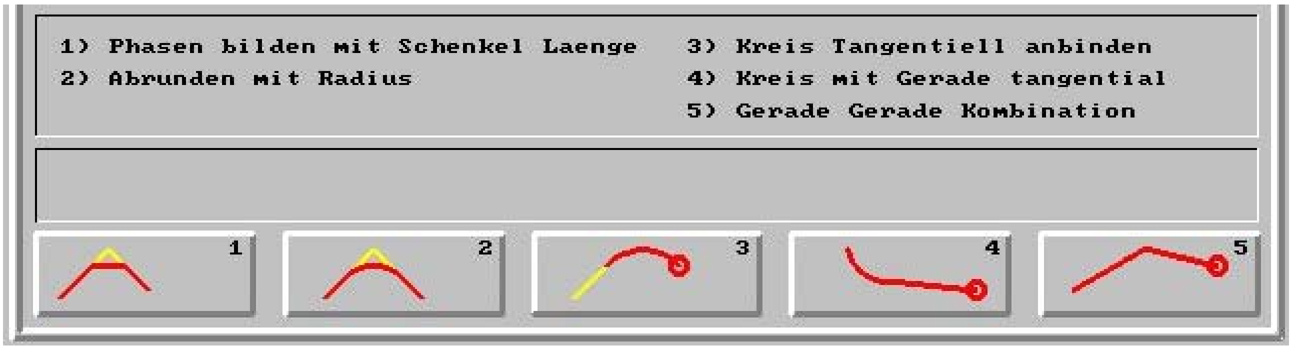
Circular interpolation with radius and 1/2 terminator point



Circular interpolation in the clockwise direction
with radius (r) and 1/2 final position (Z or X)



2.11.5 Inserting a chamfers or roundness



F1	Chamfers form with leg length
F2	Round off with radius
F3	Circle tangential tie up
F4	Circle with linear tangential
F5	Combination linear with linear

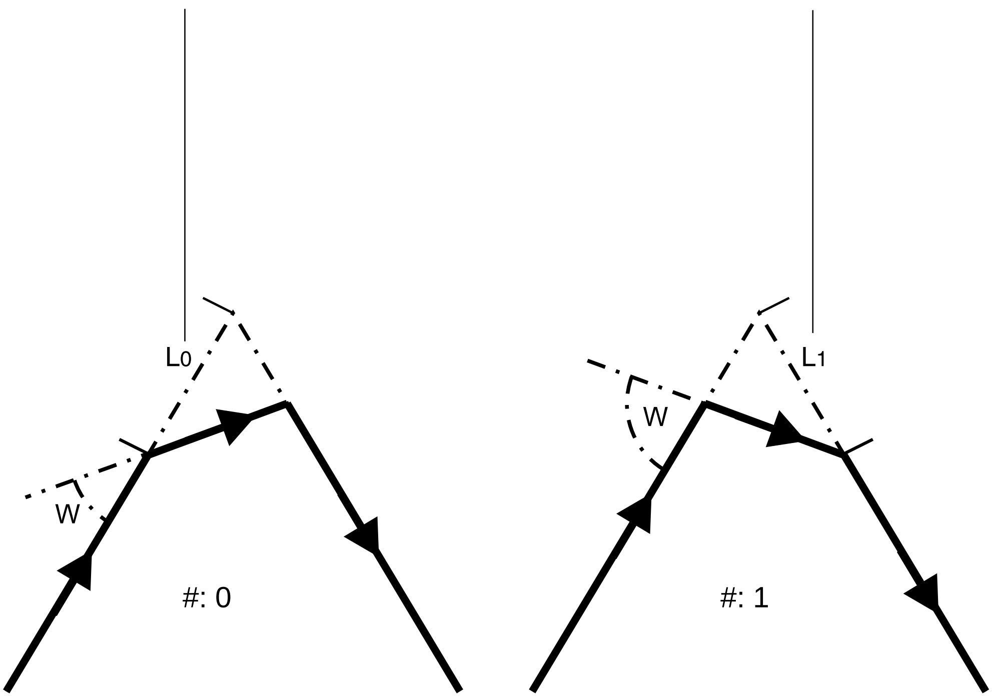
2.11.5 Inserting a chamfers or roundness (continued)

chamfers form with leg length

Chamfers at an edge with leg length and angle (w)

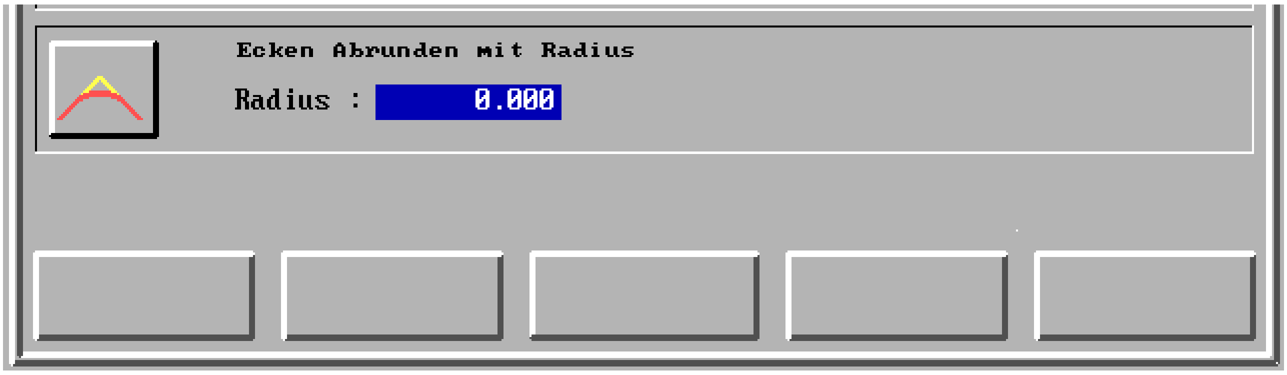
Selection of the length #: 0 1. Length #: 1 2. Length

Examples:



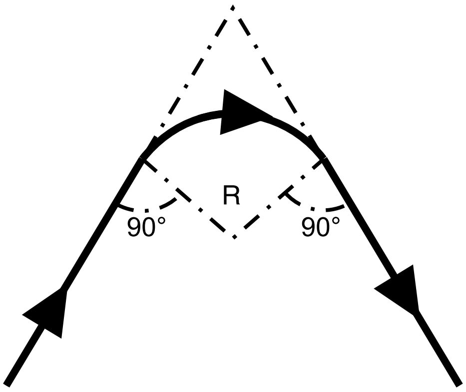
2.11.5 Inserting a chamfers or roundness (continued)

Rounding off with radius



Corner-round off with radius (r)

Example:



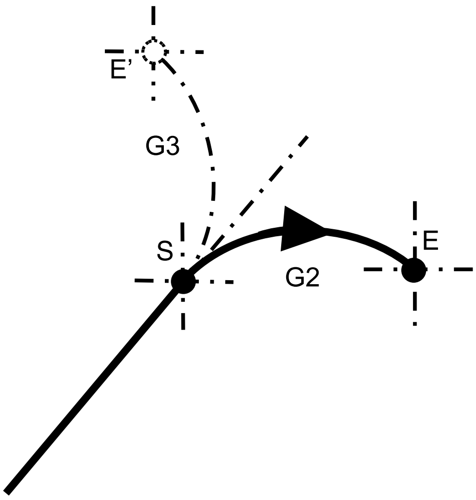
2.11.5 Inserting a chamfers or roundness (continued)

Circle tangential tie up

Circular interpolation with corner point and tagentieller binding

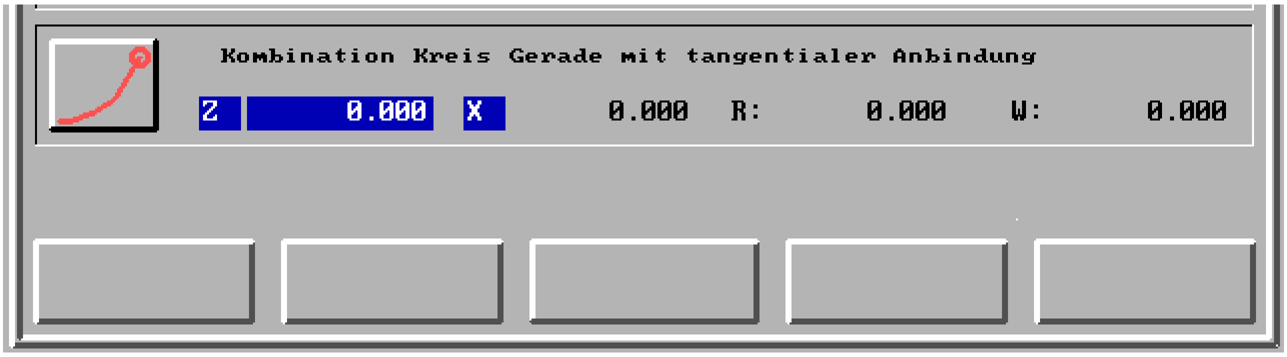
G2 or G3 become automatically
according to the position of the terminator point (E, E ‘) selected

Example:



2.11.5 Inserting a chamfers or roundness (continued)

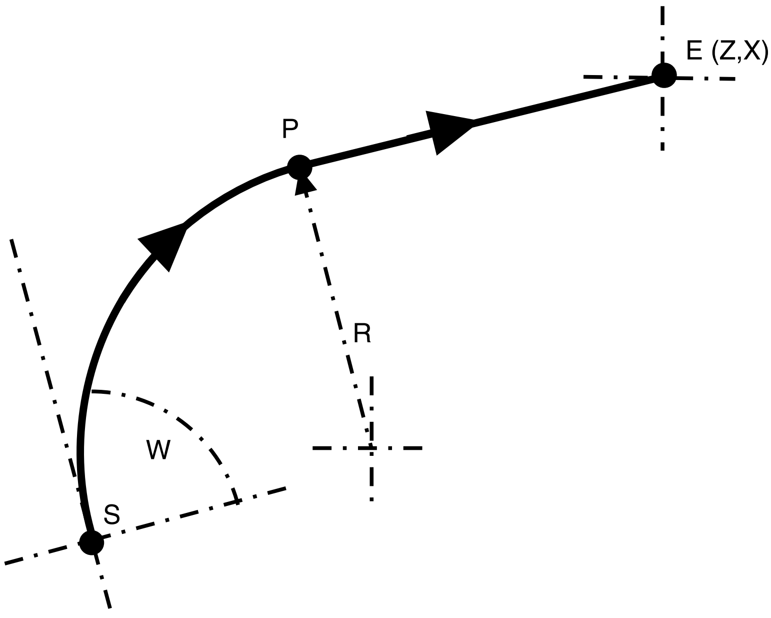
Circle with linear tangential



Combination circle - linear with tangential binding

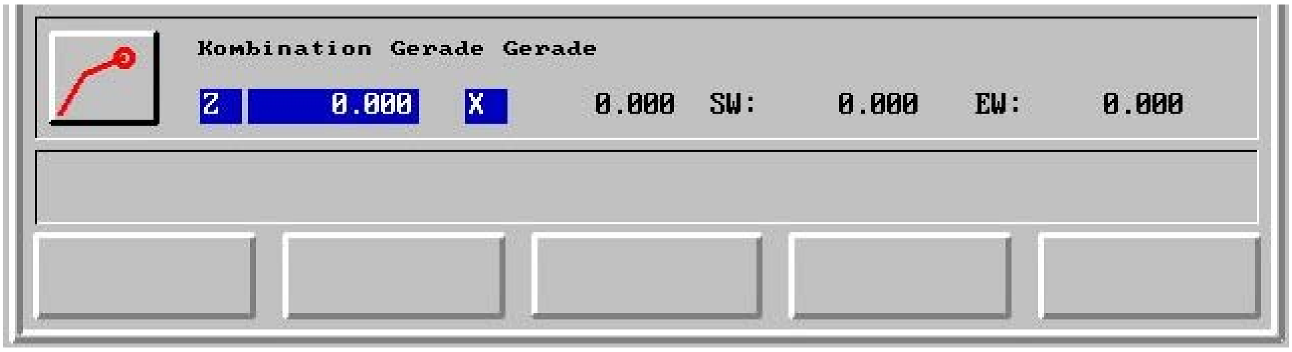
The position of the transition point (p) actual unknown.

Example:



2.11.5 Inserting a chamfers or roundness (continued)

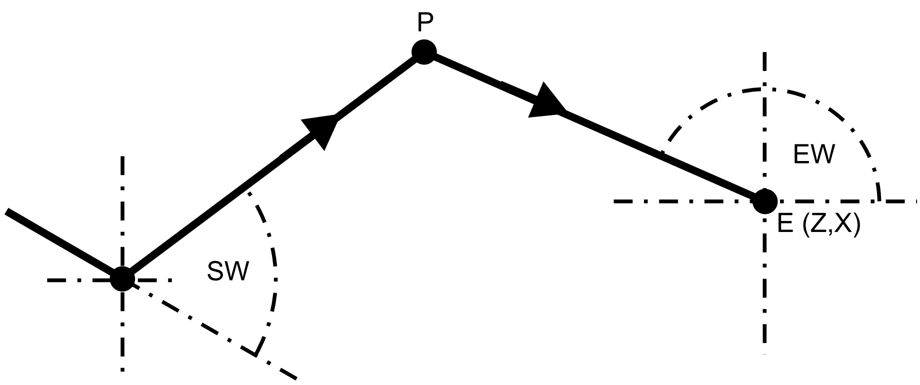
Combination linear - linear



Combination linear - linear
with start angle, final angle and
Terminator point with the coordinates Z and X
The start angle actual relative
to the preceding block.

The position of the transition point (p)
actual unknown.

Example:



2.11.6 Editing

If the processing direction determined, still the functions (F, T, G, M) know actual are inserted into the program.

F1 Program edit with full function range (F, T, G, M, P etc.), if a ASCII keyboard available actual

F2 Processing direction turn around

F3 Mirrors (in both axes)

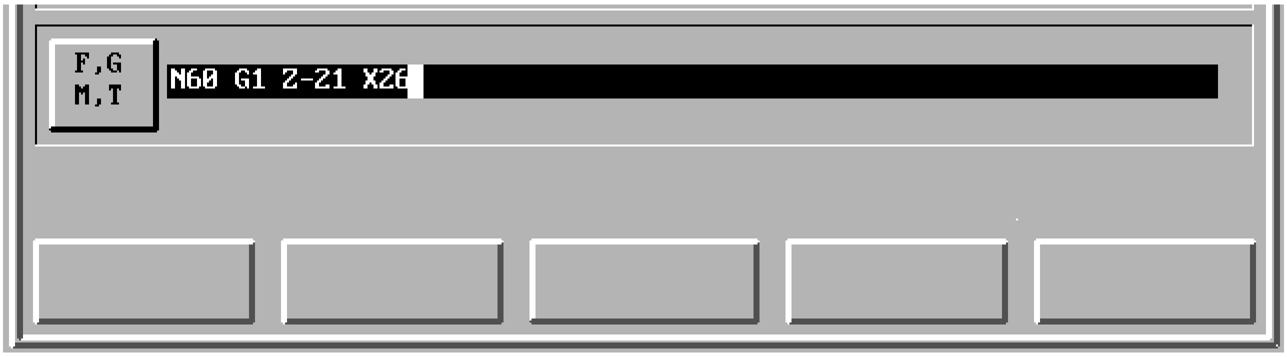
F4 Horizontal zeros (settings the white point on zero)

F5 Vertically zeros (settings the white point on zero)

2.11.6 Editing (continued)

Program edit with full function range

The program can with full function range (F, T, G, M, P etc..) are edited,
if a ASCII keyboard available actual



3. Program configuration

3. Program configuration

3.1 Program

Each program begins with a program number with a maximum of 7 digits (DOS - PC) and/or 9 digits. The program consists of individual blocks.

3.2 Block

Each block is identified by a block number with a maximum of 10digits. A block is made up of various words, which contain all instructions for an operation. The block length is variable (maximum 200 signs).
Blocks are identified in the program with ascending block numbers. This determines the sequence when the program is running and means that identical block numbers are not acceptable.

3.3 Word

A word consists of an address and a data section. The word length is variable. All words which can be contained in a block are included in the following table.

3.4 Mathematicals expression

A number can be also replaced by a mathematical expression in round brackets.

For example: ((110 + P20) / 3)

3.5 Block functions

Word	Address	No. of words per block	Data digits	Dimensional unit	Effect	Description in section
Block number	N	1	10 #		block by block	
Feed	F	1	x ~	mm/min	modal	3.
Spindle speed	S	8	x ~	1/min	modal	3.
Traverse conditions	G	8 *	10 #		modal/bl.-by-bl.	4.
Circle centre point	I / J / K	1	x ~		modal	4.
Circle radius	R	1	x ~		modal	4.
Cycle	G	8 *	10 #		block-by-block	5.
Additonal functions	M	8	3 #		modal/bl.-by-bl.	6.
Tool	T	1	10 #		modal	7.
Parameters	P / q	x	x		modal/bl.-by-bl.	8.

Key: ~ Floating point
Integer

* Traverse condition and cycle together 8 per block

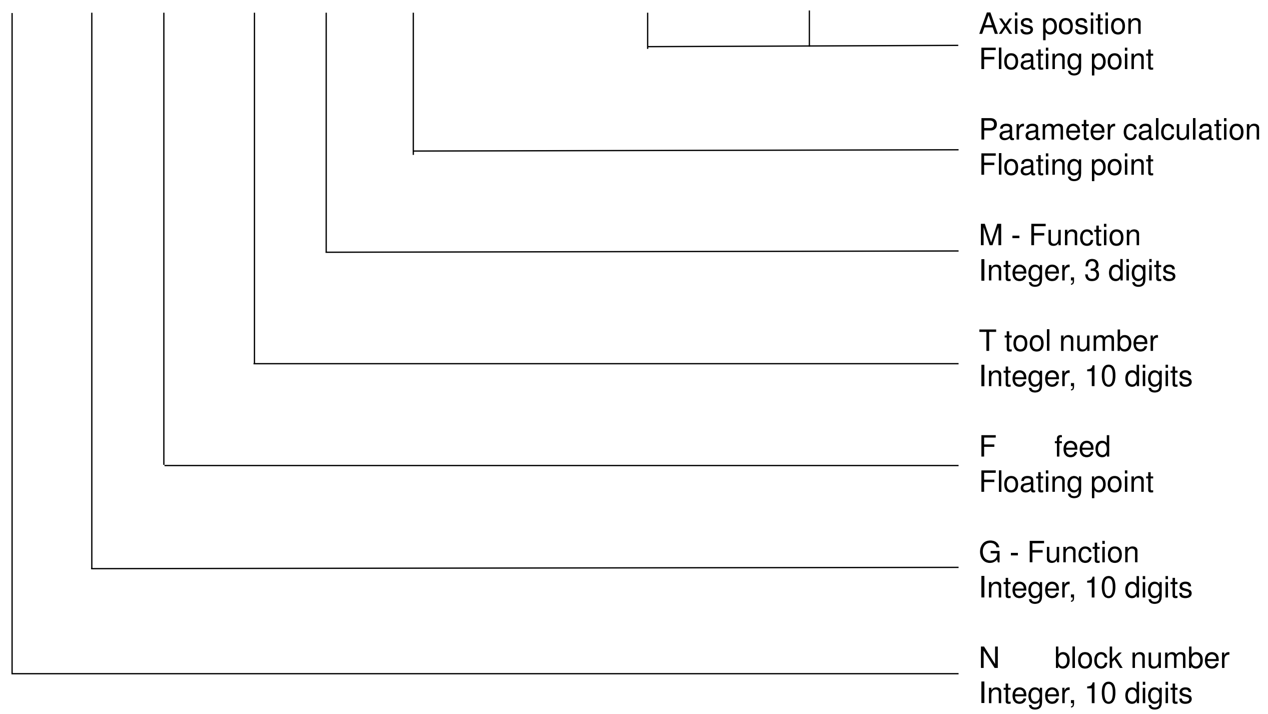
All other letters can be used for axis terms.

3.6 Syntax

Fundamental construction of a NC block:

Example :

N10 G01 F100 T03 M100 P30:(P20+35)*3 X200.0000 Y400.550 { comment }



Block length = 200 signs

3.6 Syntax (continued)

General functions

/	Block skip
\	Chain blocks, i.e. several blocks are joined to one NC block.
()	Bracket functions, mathematical expressions or comment
{}	Bracket comment
:	Allocation
$	Signal for hexadecimal numbers **Hexadezimale expressions must be completed with the separating sign(;)!** e.g.: N10 G01 F1000 P500 : $1AF; X: P500

Axes

Axes can be marked:

- with a letter	X, Y, Z, U, V, W . . . X100, Z33
- with a letter and index 1 to 8	X1, X2, X3 . . . X2:100, Z1:33 . . .

Clear parameter content

P500: - - clears the content of P500

3.6 syntax (continued)

Comparison operators

		Example :
=	Equal	P500=110.50
		(Skip to block 50, if the content of P500 = 110)
>	Larger	
<	Smaller	
<>	Unequal	
>= =>	larger equal	
<= >=	smaller equal	

If the skip - condition is fulfilled, it is skipped to the indicated block number.

Calculation operators

		Example .:
*	Multiplication	P500:P200*5
+	Addition	P500:P200+P201
-	Subtraction	P500:P200-1
/ %	Division	P500:P200 / 2
| mod	Modulus	P500:P200|10
		The trigonometrical functions use degree !
sin cos	tan	P500:sin(90)
asin acos	atan	P500:acos(P10)
or and	not	Bit operations
		P500:P500 or $100;
sqr	Square root	
int	Integer value	
intr	Rounding on integer	
abs	Amount	
ln	Logarithm with basis e	
lay	Logarithm with basis 10	
exp	Exponent	
del	test if cleared	P500:del(510)
		Feedback value : 0 parameter not cleared
		1 parameter cleared

Functions are always written in lower-case letters!

3.6 syntax (continued)

Definition of a tool radius t

If no tool administration is existing in the system, a temporary correciton radius can be defined with identification ‘t’. With this correction radius, the subsequently activated radius correction (G41/42) is working.

Example:

```
N10..
N20 t:0.5
N30 G1 X100 Y100 G42
N40..
```

When programming ‘t’
- the tool radius is written and
- the tool radius correction is deleted
  in the active data block (P8150 ...)

The quadrant of the ‘t’-correction is always 0!

3.6 Syntax (continued)

Syntax of symbolic variables

Symbolic variables always do start with the sign '_' (e.g.: _abs, _test5, ...), they may have a maximum length of 30 signs. Capitel and small letters are allowed, but the meaning of _karl may not be the same as _Karl! (case-sensitive).

Binding a symbol to a parameter

_wegx ::= 500;

According to this definition, _wegx is in place of P500, i.e. both expressions are equivalent according to the above mentioned example (_wegx <—> P500).

P500 : 10
_wegx:=10

Indexing is allowed at parameter variables.
Example

_wegx(0) := 10; (d.h. P500:10)
_wegx(1) := 11; (d.h. P501:11)

Internal variables

_wegxy := 500;

If a value is assigned to a variable, which is not 'bound' to a parameter, this variable is allocated as internal variable. That means that values can be stored without using a parameter.
Variables that are not initialized have the value 0.
Internal variable do only exist as long as the NC program is active.
Internal variables can not be displayed directly (e.g. at a program test).

Example

n10 _test1::=12 (Binding to P12)
N20 _test2:=10 (internal variable)
N30 G00 X:_test1 Y:_test2
N40 ...

3.6 Syntax (continued)

Text output of NC programs

Arbitrary texts from NC programs can be displayed in the message line. When switching back to MANUAL, texts that are eventually standing in line, are deleted.

Delete syntax messages

N10 !
N10 M33 P1:23 !

The identification '!' may also be programmed with other NC block elements in the same NC block. However, '!' has to be the last sign of the block!

Displaying messages with predefined colours

N10 !	this is a message text	white letters on blue bottom
N20 !0,	this is a message text	black letters on grey bottom
N30 !1,	this is a message text	white letters on blue bottom

Displaying messages with colour selection that can be defined freely.

N40 !S8E, this is a message text (yellow on black)

N40 !code, this is a message text code = HF + VF

HF	(background colour)	VF	(Forefront colour)
80	black	0	black
90	blue	1	blue
A0	green	2	green
B0	turquoise	3	turquoise
C0	red	4	red
D0	magenta	5	magenta
E0	brown	6	brown
F0	light grey	7	light grey
		8	dark grey
		9	light blue
		A	light green
		B	light turquoise
		C	light red
		D	light magenta
		E	yellow
		F	white

3.6 Syntax (continued)

Working sequence of the block interpreter

1. Parameter calculations, Parameter allocations
   are executed in the sequence programmed in the NC block.

2. Parameter skips
   are executed in the sequence programmed in the NC block.

3. M - Function - Skips
   are executed in the sequence programmed in the NC block.

4. Sub-program call M28

The sequence of the block elements when dispatching at PLC (real time)

1. Block number

2. Parameter (real time parameter)

3. S-value

4. T-value

5. M-function before traverses / after traverses

3.6 Syntax (continued)

Enlarged syntax

The NC interpreter offers with System-Calls (sc) further possibilities, to shift the interpreter -mode or to trigger functions.

Syntax sc: n

Function numbers

0 Activating of interpreter-mode 0, standard mode.
- Is always preset at NC program start.

- Each block produces a block end. When switching on the interpreter to the next NC block a block change command results, with which the NC block informations are transmitted to the following modules.

1 Activating of interpreter-mode 1, supervision mode.
- In mode 1 the block change command is suppressed. When switching the interpreters to the next NC block no block change command results.
Because the analyzed NC block elements are only transmitted to the following modules with block change, they remain now for the time being in the block interpreter.
(This does not count for the additonal functions.)

- When switching back to mode 0 a block change command is given.

2 Activating the interpreter mode 2
- supervision mode at M26

- as mode 1, however, when switching the mode, the system waits until it gets a feedback from the previous NC block. E.g. if you want to make some calculations or supervising loops while an axis is moving, the calculation and/or test loop will start at the beginning of this movement (and the pipeline of the NC control is previously deleted).

- A block change command is sent when switching back to mode 0.

3.6 Syntax (continued)

Example of a supervision loop

While N10 is processed, supervision functions can be perceived in the loop (N30 . .. N50).

N10	G01 F100 X1000	
N20	sc :1	changeover in mode 1
N30	P500>P501.140	skip, if supervision loop should be left.
N40	. ..	
N50	M23.30	skip to the beginning of the loop
N140	. ..	
N150	sc :0	end of the supervision loop
N200	G00 X200	

100 Triggering of a block change command
- In interpreter mode 1 a block change can be forced herewith.

3.7 Feed

The feed with the address letter F is programmed in mm or inch according to the set unit of measurement.

G94, G95 and G99 determine the feed modification.

G94	Feed in mm/min
G95	Feed in mm/r
G99	Block end feedrate in mm/min

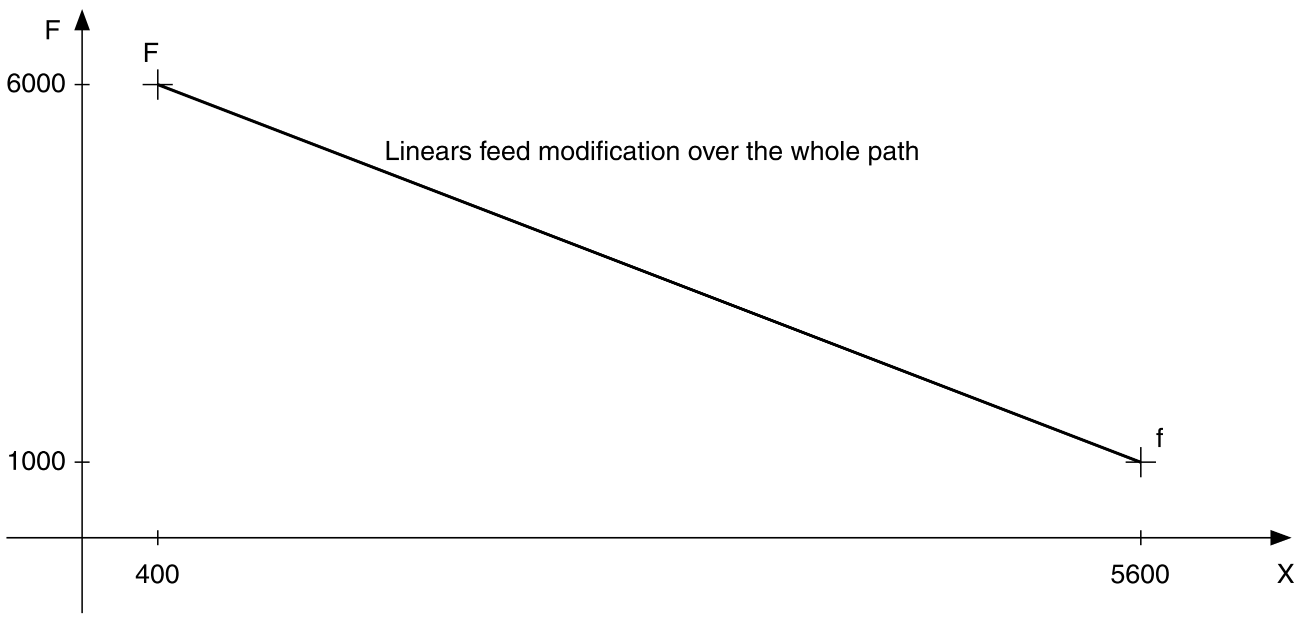
A programmed feed is effective modally and can only by overwritten with another feed.

Through positioning in rapid traverse (G00) the previously stored effective feed is not cleared, but becomes effective again with G01, G02 and G03.

With the feed-override-potentiometer the programmed feed can be changed in the area of 0 to 120%.

F	path feed
f	final feed, i.e. path feed, which are achieved at the end of block.

Example:	N50	X400			
	N60	G1	F6000	f1000	X5600



3.8 Spindle speed

The spindle speed is programmed with the address letter S. With index 1 to 8 several spindles can be programmed.

Example: S1000

S1:1000
:
S8:8000

G96 and G97 determine the speed modifications.

G96 constant cutting speed in mm/min
G97 number of revolutions in 1/min

The spindle speed is effective modally and can only be overwritten through another spindle speed.

With the spindle-override-potentiometer the programmed spindle speed can be changed in the area of 0 to 120%.

3.9 C-NC Syntax

3.9.1 Programming Example

```
P1000
        // Program - example
        int    idx=0;
        double step_x=1.234;
        P100:1;
        G0 X0 Y0 Z0
N100 P100:P100+1;
        if (P100>6) M23.200 //======>N200
        if (P100>5)
        {
        G0 X123;
        }
        else
        {
        G0 X99;
        }
        G1 F1000 X100 M543;
        for (idx=0; idx<5; idx++)
        {
        G91 F123 X:step_x;
        }
        G1 G90 F1000 X123.5 Y82.6
        G0      Z150
        G1 F5000 X99.5  Y43.123
        M23.100
N200 M30
#
```

3.9 C-NC Syntax

3.9.2 Commissioning

3.9.2.1 System Requirements

Necessary Software- Versions:	CNC : BS-Version : 160 / 220 or higher Panel: PPCIMAGE/QT900 : Version 160 or higher current show_e

- nc_line_mode must be enabled	(System-wide setting)
- C- Interpreter must be enabled	(channel-specific)

3.9.2.2 Activation nc_line_mode

In order to execute NC programs in C interpreter mode, the system must be operated in nc_line_mode.
If nc_line_mode is activated, NC blocks do not necessarily start with block numbers; NC Editor is line oriented.

Activation of nc_line_mode:
Entry in the file netconf in the CNC:
'nc_line_mode'

nc_line_mode- Status, see q41
q41 : Panel - System-Status-Infos :
q41 is written by the power-on routine by the system.
Content:
Byte 1 :
Bit 0 : 0 : nc_line_mode is not enabled

NC blocks must necessarily begin with 'N' or '/ N'.
NC-Editor is block - oriented.
When switching from nc_line_mode enabled, to nc_line_mode disabled the NC -Memory has to be deleted.

1 : nc_line_mode is enabled

NC blocks do not have to necessarily start with 'N' or '/ N'.
Jump Labels still have to start with 'N' or '/ N'.
When switching from nc_line_mode disabled to nc_line_mode enabled, it is not necessary, delete the NC-Memory.

3.9 C-NC Syntax

3.9.2.2 Activation nc_line_mode (continued)

IO-INPUT :
- NC Program Input: Blank lines are not filtered.
- Phrases starting with ';' be filtered as before!

NC Editor is line oriented. (Display the line number)
NC block numbers now act as entry point for NC block jumps.
Jump-marks must be in ascending order!!

e.g.: .

```
G01 F1000 X100
M23.100    --> Jump to Mark N100
.
N100 X150
G04.1
X100
.
```

Function: NC Renumber is blocked at nc_line_mode enabled.
During the teaching of NC blocks, no block number is inserted.

In AUTOMATIC Picture NC blocks are displayed with line numbers.

The following parameters now contain line numbers:

P8695	Line number at RWL
P8696	Line number at the measuring position recording
P8803	Line number at NC program abort
P8901	current line number NC program real-time
P8905	current line number cycle real-time
P8907	Line number at NC program abort
P8911	Line number of the incorrect block
P8914	current line number in subroutine call
P9301	current line number NC program pre-analysis
P9305	current line number Cycle pre-analysis
P11146	Line number of the last edited NC program
P11152	not used
P11153	not used
P11154	not used
P11184	current line number at IO In-/Output

3.9 C-NC Syntax

3.9.2.2 Activation nc_line_mode (continued)

The following parameters contain As before block numbers:
P8801 NC Start block number
P8821 Emergency program 1 block number
P8823 Emergency program 2 block number
P8825 Emergency program 3 block number
P8827 Emergency program 4 block number

3.9.2.3 Activation of C-interpreter

P8805 : Selection NC Interpreter
0, -- : NC program is executed with standard NC Interpreter.
Automatic mode display: AUTO
1 : NC program is executed with C-NC Interpreter.
--> C- Syntax
Automatic mode display: AUTOC

--> See the menu: Start data C interpreter

3.9 C-NC Syntax

3.9.3.1 NC Elements

Definition :
NC elements are :
- Axis identifiers X,Y,... x,c..
- axis CMD a(pos)
- N, M, G, F, f, S, R, I, J, K, T, t, tr, tq, tl, P.., p.., q..
- sc.. sc1
- !
- internal Vars (starting with '_') Example: _ncvar
- NC Functions / Functional

- NC elements are prohibited as a variable name

- A separator must be programmed between NC elements!
Separators are:
';' (Semikolon)
' ' (Blank)
operators

Example:
F1000 x100 c45
P500:12;P501:13;
P100:P123 mod 5

Are NC items in the block, the NC Items will be executet with block end, i.e., NC-elements will be send to 'real time'.

-'N' will only accepted as entry point, if programmed at the 1st digit of the line!

3.9.3.2 Block Commands

{ : C- Beginning of Block
} : C- End of Block

e.g.:
```
{
int abcde;
.
.
}
```

M2 / M30 may NOT be programmed out of C blocks!

NC block jumps may NOT be made out of C blocks!
(--> see message M1375)

3.9 C-NC Syntax

3.9.3.2 Block Commands (continued)

NC block jumps are:
- M23 jumps M23.xx,
- defined jump-M functions Mxxx.xxx
- parameter jumps P100=P101.xxx
  _test=_test1.xxx
- M24/M25
- M29

e.g. 1: correctly

```
N100.
.

{            // C- block start
int abcde=0;
.
.
}            // C- end of Block
G01 F1000 X100
G54
M23.100      // Block jump is allowed
.
.
```

e.g. 2: !!false!!

```
N100.
.
{            // C- block start
int abcde=0;
.
.
M23.100      // Block jump INADMISSIBLE --> M1375
             // --> NC block jumps may not be made out of C blocks !!
}            // C- end of Block
G01 F1000 X100
G54
.
.
```

If the C-block start and C-block end is in the same line, this block is filtered out as a comment.

e.g.:

```
if (test>10)  { I'm a comment}
{
P500:test;
}
```

3.9 C-NC Syntax

3.9.3.3 Comment

// Comment from / / to end of line
e.g.:
G0 X100 // Traverse the X-axis

/* Comment-start
*/ End comment
e.g.:
G0 X100
/* text
text
text */
G0 Y100

For single-line comment '{' and '}' may also be used. (NC-Compatible)

e.g.:
if (test>10) { I'm a comment }
{
P500:test;
}

3.9.3.4 Constants

M_PI (3.14....)
Example:
myint=M_PI;

3.9.3.5 Definitions

#define defname defconst

e.g.:
#define axis1 1

Definitions apply globally.
! The '#' is not at the beginning of line!

3.9 C-NC Syntax

3.9.3.6 Includes

#include <fname>

e.g.:
#include <myinclude> File 'myinclude' will be included

Character '#' must not be at beginning of line!
Function declarations in the main program apply globally across all programs until the program ends.
Function declarations in a subroutine are local to this subroutine.

3.9.3.7 Data types and scope of application.

char	:	1 Byte	signed	-128	..	127
unsigned char	:	1 Byte	unsigned	0	..	255
byte	:	1 Byte	unsigned	0	..	255
short	:	2 Byte	signed	-32768	..	32767
unsigned short	:	2 Byte	unsigned	0	..	65535
int	:	4 Byte	signed	-2147483648	..	2147483647
unsigned int	:	4 Byte	unsigned	0	..	4294967295
long	:	4 Byte	signed	-2147483648	..	2147483647
unsigned long	:	4 Byte	unsigned	0	..	4294967295
float	:	Floating-point single precision				
double	:	Double precision floating point value				
string	:	text; ONLY one-dimensional C string, ie NULL Terminated				

e.g.:
```
teststr="abc";
teststr[0] : 'a'
teststr[1] : 'b'
teststr[2] : 'c'
teststr[3] :  0
```

e.g.:
Simple data types declaration:
```
double  dwert1;
double  dwert1, dwert2, dwert3;
double  dwert1=1.0;
int     a=1;
int     a,b=1,c;
char    mych=65;
char    mych='A';
```

Declaration of arrays:
```
double  dwert[20];       // One-dimensional array
int     xxxxx[3,2];      // Two-Dimensional array
```

3.9 C-NC Syntax

3.9.3.7 Data types and scope of application (continued))

Deklaration von Strings:
```
string myname;
string myname="hugo  ";
```

Scope of declarations:

- Declarations in the main program (not defined in a block) are global across all programs until the program ends.
  (Also applies to declarations of include files which are includiert from the main program.)
- Declarations in a subroutine apply only locally in this subroutine.
- Declarations in functions that apply to the function end.
- Declarations in a block, valid until end of block.

e.g.:
```
{
	int test1;
.
}
```
- Declaration within a block, displaces a declaration outside of the block until the block end.

e.g.:
```
int test1=12;
{
	int test1=14;
	P1:test1;
}
P2:test1;	// Resultat --> P1 : 14
		// P2 : 12
```
- Declaration in a subprogram, displaces a global declaration until end of the subroutine.

3.9.3.8 Hex code

$ $ generally valid.
0x   0x ONLY allowed in a C syntax element,
that is, hex code '0x 'can NOT be used with NC-elements.
!! 'x' is the POLAR radius identifier. !!

e.g.:
```
correctly:	P1:$ff;
		i=0xff;
false:		P1:0xff;
		X:0xff;
```

3.9 C-NC Syntax

3.9.3.9 Assignment Operator

=

C-Element	'='	value

Assignment operator for C-elements '=' Example: myvar =1.234;
dwert=12.123;

!! **Difference to the NC syntax** !!
Please note :
The previous NC syntax remains unchanged!

NC-Element	':'	value

Assignment operator for NC elements ':' Example: P500:3
_myvar:3

NC-Element	'='	value	'.'	jump destination

Comparison operator for NC elements '=' Example: P500=P501.123
_myvar=P123.100

3.9.3.10 Arithmetic Operators

*	:	multiplication		
/	:	division		
%	:	division (Deviating from the standard C: In C is %: modulo)		
+	:	addition		
-	:	subtraction		
**	:	potentiate	tst=10**3	-->1000
mod	:	modulo	tst=23.2 mod 3	-->2.2

3.9.3.11 Comparison Operators

==
!=
>=
<=

3.9 C-NC Syntax

3.9.3.12 Logical Operators

!	not	logically
||	Or operator	logically
&&	And operator	logically

3.9.3.13 Bitwise Operators

&	And operator Bit Operation	e.g.: aa=bb&cc;
|	Or operator Bit Operation	e.g.: aa=bb|cc;
^	Exclusive Or Bit Operation	e.g.: test=test^$ff;
<<	Shift operator to the left	e.g.: test=test<<2;
>>	Shift operator to the right	e.g.: test=test>>2;
~	BitComplement	e.g.: test=~test;

3.9.3.14 Combination Operators

++	increment	e.g.: aa++;	--> aa=aa+1;
--	decrementing	e.g.: aa--;	--> aa=aa-1;
	Take note: P500: -- delete Parameter		
+=	addition	e.g.: aa+=5;	--> aa=aa+5;
-=	subtraction	e.g.: aa-=5;	--> aa=aa-5;
*=	multiplication	e.g.: aa*=5;	--> aa=aa*5;
/=	division	e.g.: aa/=5;	--> aa=aa/5;
>>=	Shift to the right	e.g.: aa>>=2;	--> aa=aa>>2;
<<=	Shift to the left	e.g.: aa<<=2;	--> aa=aa<<2;
&=	And operator Bit Operation	e.g.: aa&=$ff;	--> aa=aa&$ff;
^=	Exclusive or Bit Operation	e.g.: aa^=$ff;	--> aa=aa^$ff;
|=	Or operator Bit Operation	e.g.: aa|=$ff;	--> aa=aa|$ff;

3.9.3.15 General functions / system functions

Default : Arguments in DEGREE

Function Name+ '(' + Function arguments.. + ')'

asin()
acos()
atan()
sin()
cos()
tan()

3.9 C-NC Syntax

3.9.3.15 General Functions / System Functions (continued)

deg()	Conversion rad --> deg	; tst=deg(3.14);
rad()	Conversion deg --> rad	; tst=rad(180);

log2() Binary logarithm
e.g.: tst=log2(16); --> tst=4
For system version <230 : result: integer
From System Version 230 : result: double

ln()
exp()
log()

abs()	absolute Value	tst=abs(-1.2) -->	1.2
int()	integer Value	tst=int(1.667) -->	1
intr()	Rounding to integer	tst=intr(1.667) -->	2
sqr()	Square root	tst=sqr(4)	
rnd()	Random value/random number	tst=rnd(3.4) -->	Random number in the interval:-3.4 ... 3.4

msleep(n) n : [msec] Time for execution of the command

Return value of msleep: System timer [msec] at the command call

n : -1 : no yield of msleep,
n : 0 : 1 x yield
n : >0 : yield, until n reached

printf() Formatted output to the upper message line (such as '!')
printf(control, arg1, arg2..)

!!
Format elements and arguments must match the number and type!
!!

Format elements :
%d : Decimal Representation
%nd: Decimal Representation n: minimum Field width
%x : Hexadecimal Representation
%nx: Hexadecimal Representation n: minimum Field width
%f : The argument must be float or double, and is represented by 6 decimal places
%n.mf: The argument must be float or double, and is represented by 6 decimal places
n: minimum Field width
m: Number of digits right from the decimal point
(it m==0 --> 6 decimal places)
%c : The argument is a single character
%s : Argument is a string

3.9 C-NC Syntax

3.9.3.15 General Functions / System Functions (continued)

e.g.:

```
int i=5;
double dd=0.123;
char ch=$51;
string str="Hello World";
printf("output -->'%d' '%f' '%c' '%s'",i,dd,ch,str);
Output:
output -->'5' '0.123000' 'Q' 'Hello World'
printf("output -->'%3d' '%6.2f'",i,dd);
Autput:
output -->'  5' '  0.12'
```

Choice of colors for the message line:

default Color:	Foreground color:	white
	Background color :	blue

If the output string starts with "$xx,", "$xx," is for color selection and won´t be displayed.

Color definition:
$xx = HF + VF

HF (Background Color) =	80	black
	90	blue
	A0	green
	B0	turquoise
	C0	red
	D0	magenta
	E0	brown
	F0	light gray

VF (Vordergrund- Farbe) =	0	black
	1	blue
	2	green
	3	turquoise
	4	red
	5	magenta
	6	brown
	7	light gray
	8	dark grey
	9	light blue
	A	light green
	B	light turquoise
	C	light red
	D	light magenta
	E	yellow
	F	white

3.9 C-NC Syntax

3.9.3.15 General Functions / System Functions (continued)

unlink(file_name) delete File

file_name : "_ncram/P1000" : File in the NC memory
"//server/ncdata/P1234" : File from the file server
"//control_panel/ncdata/P1234" : File in the control panel

Return : 0 : Function executed error-free
!=0 : Error code

e.g. : int err;
err=unlink("_ncram/P1000");

3.9.3.16 String Manipulation

= copy String
e.g .: string2 = string1;

+ Combine strings
e.g.: string3= string1+ " abc " + string2;

==
!= Compare string
e.g .: if (string1 == string2)...
if (string1 != string2)...

strlen() Determine the string length
string mystring="abcd";
len=strlen(mystring); // --> len : 4

mid(str,idx,len) str : source string
idx : Index in the source string [0..n]
len : string length

Returns a string, starting at index idx with length len.
Return a null string if the string is empty or index idx outside source string.

e.g.: stringx="Hello World";
stringy=mid(stringx,1,4); // stringy=="ello"

3.9 C-NC Syntax

3.9.3.16 String Manipulation (continued)

idx=find(str,suchstr)

```
str         : source string
searchstr   : searchstr index in the source string [0..n]
Return
idx         : -1   , if searchstr not found in str
            :  0..n , if searchstr found in str

e.g.:  stringx  ="bananas";
       subs     = find(stringx,"an");    // subs == 1
```

stringx[idx] Indexed String Access [0..n]

```
e.g.:  chx = string1[2];
       string1[1]='A';
```

chr(n) Conversion ASCII code --> string

```
e.g.:  mystring=chr(65);        // mystring == "A"
```

itoa Integer Conversion to string

```
e.g.:  string mystr;
       int    myint=15;
       mystr=itoa(myint);       // --> mystr : "15"
```

hex(val) Integer Conversion according to string with base 16 (hexadecimal)

```
e.g.:  value=35243;
       stringx=hex(value);      // stringx == "89ab"
```

ftoa Conversion float (double) to Ascii (String)

```
e.g.:  string mystr;
       double mydoub=15.1234;
       mystr=ftoa(mydoub);      // --> mystr : "15.1234"
```

atof() Conversion String to Float (double)

```
e.g.:  string mystr="15.1234";
       mydoub=atof(mystr);      //mydoub == 15.1234

       On error: return value = 0.0
```

3.9 C-NC Syntax

3.9.3.17 f- CMDs

fchksum(pn); Programchecksum of a file in the NC memory

pn	:	Programnumber
Return	:	Filechecksum
e.g.: P1	:	fchksum(123);

fexist(pn); FileCheck in the NC memory

pn	:	Programnumber
Return	:	0 : Program in the NC memory not available 1 : Program available
e.g.: P1	:	fexist(123);

fnccopy(src_pn, dst_pn) NC-Copy in the NC memory

src_pn	:	Source program number
dst_pn	:	Target program number
Return	:	0 : Function executed error-free >0 : Error code 1250 .. 1299
e.g.: P1	:	fnccopy(1,123);

filecopy(src_name, dst_name, mode) FileCopy to/from NC_memory

src_name	:	FileName source
dst_name	:	FileName target
mode	:	CopyMode

0: In File --> NCRAM: Filename target = dst_name, regardless of the programname in source file
In NCRAM --> File: Programname IM targetfile = File Name targetFile

Filenamen : "_ncram/P1000"	:	File in NC_memory
"//server/ncdata/P1234"	:	File from the file server
"//control_panel/ncdata/P1234"	:	File in the control panel
"//cnc5001/_ncram/P1234"		

Return : 0 : Function executed error-free
>0 : Error code 1200 .. 1299

e.g.: P1:filecopy("//server/ncdata/P1000","_ncram/P1000", 0);

funlink(file_name) delete File

file_name : "_ncram/P1000"	:	File in NC_memory
"//server/ncdata/P1234"	:	File from the file server
"//control_panel/ncdata/P1234"	:	File in the control panel

Return : 0 : Function executed error-free
>0 : Error code 1200 .. 1299

e.g.: P1:funlink("_ncram/P1000");

3.9 C-NC Syntax

3.9.3.18 if/else- Authorizing

The if-else statement is used in making decisions.
The else part is optional.
Syntax :

```
if (Expression)
        Statement1;
else
        Statement2;
```

If the expression has a value different from 0, so Statement1 is executed.
If the expression value is 0, statement2 is executed if an else part is present.
Assignments in the expression are not allowed! Example: if (iwert = abc) ...

e.g.:

```
if (iwert > 100)
{
G0 X100;
}
else
{
G0 X200;
}
```

or

```
if (P[2,500] == 1)
{
  do_something();
}
```

Caution:

For single-line comment '{' and '}' may also be used. (NC-Compatible)
e.g.:

```
if (test>10)  { I'm a comment }
{
  P500:test;
}
```

3.9 C-NC Syntax

3.9.3.19 switch- Authorizing

The switch statement is a special kind of selection among several alternatives.
If this is the case, then the branch is accordingly.

Syntax:
```
switch (Expression)
{
case konst1 :
        Statement1;
break;
case konst2 :
        Statement2;
break;
.
.
.

default :
        Statement3;
break;
}
```

Expression is evaluated and gives a result. The result must be an integer.
The result is compared with all the case constants.
Is there a case constant matches the result, the execution of the program in this case label is continued until the break statement.
With the break statement the switch will exit.
If there is no match, the process continues with the default label.
default is optional.
Assignments in the expression are not allowed! Example: switch (test1 = abc) ...

e.g.:
```
switch (test1)
{
case 1 :
    G0 X100;
    test1=2;
  break;
case 2 :
    G0 Y100;
    test1=100;
  break;
default :
    test1=1;
  break;
}
```

3.9 C-NC Syntax

3.9.3.20 for- Loop

for(Expression1; Expression2; Expression3)
Statement

Expression1 is used to initialize the loop.
Expression2 defines a test that is evaluated before each execution of statement.
If Expression2 has the value 0, the loop is finished.
The term expression3 is performed at the end of each repetition.
Each of the three expressions may be missing.

e.g.:

```
for(i=0; i<10; i++)
{   // Loop describes    P500: 0
    //                   P500: 1
    //..
    //                   P509: 9
    P(500+i):i;
}
```

3.9.3.21 while- Loop

while(Expression)
Statement

Statement are repeated as long as the value of the expression is not 0.
Expression is evaluated BEFORE each execution of statement.
Assignments in the expression are not allowed! Example: while (i = abc) ...

e.g.:

```
int i=0;
while(i<10)
{   // Loop describes    P500: 0
    //                   P500:1
    //..
    //                   P509:9
    P(500+i):i;
    i++;
}
```

3.9 C-NC Syntax

3.9.3.22 do-while- Loop

```
do
        Statement
while(Expression)
```

Statement are repeated as long as the value of the expression is not 0.
Expression is evaluated AFTER any execution of statement.
Assignments in the expression are not allowed! Example: while (i = abc) ...

e.g.:

```
int i=0;
do
    {   // Loop describes    P500:0
        //                   P500:1
        //..
        //                   P509:9
        P(500+i):i;
        i++;
    }
while(i<10);
```

3.9.3.23 return- Instruction

In a function, the return statement ensures that the execution of the program resumes after the function call.

e.g.:

```
void funktion1( int xyz)
{
.
.
return;          Exit the function
.
.
}

int funktion2( int xyz)
{
.
.
return 123;      Exit from the function, and Define 123 return value.
.
.
}
```

3.9 C-NC Syntax

3.9.3.24 break- Instruction

Break provides for the discontinuation of nearest do, for, switch, or while statement, a function of which the break statement occurs.
Program execution continues with the statement following the terminated do, for, switch, or while statement.

3.9.3.25 continue- Instruction

Continue must be in a do, for, or while statement.
Continue ensures that the execution of the program will continue from the point at which a decision on the repetition of a do, for, or while statement will be made.

3.9.3.26 Defining and Calling Functions

Definition:

```
function_name(Parameter...)
{
Statement
}
```

call:

```
function_name(Parameter...)
```

When calling the function, function parameters are passed by value.
(Call by Value)
that is, after the function call, the function parameters are the same, although they have been changed within the function.

e.g.:

```
void do_something(void)
{
// do_something
}
.
.
do_something();   // call
```

3.9 C-NC Syntax

3.9.3.26 Defining and Calling Functions (continued)

e.g.:

```
void writepara10(int value)
{       // value is written to P10
P10:value;
}
.
.
writepara10(1234);
```

e.g.:

```
double get_alfa(double value)
{
double result;
result=atan(value);
return result;
}
.
.
P11:get_alfa(P10);
```

3.9 C-NC Syntax

3.9.3.27 Integration of classes

Declaration:
class ClassVarName:Classname;

Call:
ClassVarName.functionName(Parameter)

ClassVarName : any name
Classname : File name of the class, without extension '.class'

e.g.:

```
//Declaration
class varx:myclass;
.


//Call :
P11:varx.get_alfa(P10)
.
.

File : myclass.class
//myclass.class
double x,y;

//Constructor
void myclass()
{
   x=0.0;
   y=0.0;
}

double get_alfa(double value)

{
   double result;
   result=atan(value);
   return result;
}
//Filend myclass.class
```

3.9 C-NC Syntax

3.9.3.40 System Kommandos

syscal(syscall_cmd, ...)

Read by CNC SYSTEM STRINGS

string syscall(__CMD_READ_CNC_STRING__, str_num)

str_num : stringnumber / Ident	0x0000xxxx	free customer string number
	from 0x0001xxxx	BWO systemstring - number
	0xffffffff	BWO systemstring - number

Return : CNC-String

e.g.:

```
string s=syscall(__CMD_READ_CNC_STRING__, 5);
```

Writing of CNC SYSTEM STRINGS

int syscall(__CMD_WRITE_CNC_STRING__, str_num, str)

str_num : stringnumber / Ident	0x0000xxxx	free customer string number
	from 0x0001xxxx	BWO systemstring - number
	0xffffffff	BWO systemstring - number

str : String- content
Return : Result : 0 : OK
1 : Failed to write the string

e.g.:

```
res=syscall(__CMD_WRITE_CNC_STRING__, 5, "Hello World");
```

3.9 C-NC Syntax

3.9.4 Compatibility with the NC interpreter

If C-NC interpreter activated :

'(' ..')' should NOT used as a framing comment!
'{' and '}' are allowed for single-line comment.
Better use: / * .. */ or '/ /'.

Since version 162/222 from 14.09.2010 '(' .. ')' allowed as comment framing.

'|' was at NC interpreter the mod function (fixed decimal point)
'|' at C-NC: or-bitoperator

Required separators at operators:

```
so far:            C-NC-Interpr.:
P1:P2mod3   -->  P1:P2 mod 3
P1:P2orP3   -->  P1:P2 or P3
P1:P2andP3  -->  P1:P2 and P3
```

necessary brackets for function calls:

```
so far:            C-NC-Interpr.:
P1:notP2    -->    P1:not(P2)
                  |
P1:sinP2    -->    P1:sin(P2)
P1:cosP2    -->    P1:cos(P2)
P1:tanP2    -->    P1:tan(P2)
P1:asinP2   -->    P1:asin(P2)
P1:acosP2   -->    P1:acos(P2)
P1:atanP2   -->    P1:atan(P2)

P1:sqrP2    -->    P1:sqr(P2)
P1:intP2    -->    P1:int(P2)
P1:intrP2   -->    P1:intr(P2)
P1:absP2    -->    P1:abs(P2)
P1:lnP2     -->    P1:ln (P2)
P1:logP2    -->    P1:log(P2)
P1:expP2    -->    P1:exp(P2)
etc..
```

3.9 C-NC Syntax

3.9.5 Error messages from the C interpreter

M1350 C Interpreter enabled but not active nc_line_mode (see q41)
M1351 Syntax error: C Error NO Heaphen
M1352 Syntax error: 'while' not programmed
M1353 Syntax error: missing Semicolon
M1354 Syntax error: missing Brackets
M1355 Syntax error: invalid Type
M1356 Syntax error: Variable not found
M1357 Syntax error: Function not found
M1358 Syntax error: error in Char-constants
M1360 Syntax error: missing Comma
M1361 Syntax error: invalid Operator
M1362 Syntax error: invalid Parameter
M1363 Syntax error: missing Point
M1364 Syntax error: Jump Destination can not be found
M1365 Syntax error: missing Colon
M1366 Syntax error: Class not found
M1367 Syntax error: unexpected EOF

M1370 Syntax error: Prototype incorrect
M1371 Syntax error: File not found
M1372 Syntax error: Unsigned not possible
M1373 Syntax Error: Invalid variable access
M1374 Syntax error: Symbol programmed twice
M1375 Syntax Error: NC jump in C block not allowed

3.9.6 Marginal Comments

II
- 'N' will mark only accepted as entry point, if it is programmed to first place!
- C-NC interpreter is case-sensitive.

3.10 Input / output (I/O)

DOS data format

Structure of a NC program file

Blank line	(CR, LF)

Identification P/Z with program number (program number with max. 9 digits

NC block beginning with N or /N
...
...
...
NC block

Program end sign	(#)
EOF-sign	(default : character 04)
Blank line	(CR, LF)

Example:
File name : P123456

```
P123456
N10 G0 X0 Y0 Z0
N20 F100 G1 X100
N30 M30
#
```

3.10 Input / Output (continued)

Structure of a parameter file

Blank line	(CR, LF)	
Identification D	(at identification D:	parameter status is not overtaken exception: If mantissa programs,
	(at identification D+:	parameter status is overtaken Exception: If mantissa programs, however in the parameter status the loading bit (byte 1, bit 1) is not set, than the bit ‘ parameter loaded ‘ is set! example: D+ K1 P1: 123 S:$32000100 in this case the status becomes to S:$32000101!

q parameter number : parameter content [S: parameter status] *   * [ ] optional

...
...
...

Program end sign	(#)
EOF-sign	(default : character 04)
Blank line	(CR, LF)

3.10 Input / Output (continued)

Structure of a parameter file

e.g.:

Filename : D123

```
D
q     0:  —             S:$00000000
q     1:          8     S:$00000001
q     2:          2
q     3:      30000
q     4:  —
q     5:  —             S:$00000000
#
```

or

```
D
K1:P  10:         1     S:$00000009
K1:P  11:       100     S:$0000000D
K1:P  12:       200     S:$00000001
K1:P  13:         5
K1:P  14:  —
K1:P  15:         2
#
```

Extensions

starting from version 080 :

With identifier D+ knows the parameter status with the function ‘ SET ‘, or with which old parameter status with the function ‘ OR ‘ is set.

Example: Parameter status with function ‘ SET ‘:
K1 P1: 123 S:$32000101

Parameter status with function ‘ OR ‘:
K1 P1: 123 S|$32000101

4. Traverse functions

4. Traverse functions

4. Traverse functions

The traverse functions, togehter with the traverse information, establish the geometric part of the program. They consist of the address letter G and a 10-digit number. One block may contain 8 traverse functions.

If the traverse functions and the appropriate traverse information are programmed in different blocks, the traverse functions in the program should always precede the traverse information. Traverse functions become effective before the programmed traverse information.

The following table contains the traverse functions made available by the control.

Traverse functions within a group overwrite each other mutually (in addition G92 is overwritten by G53 to G59).

The traverse functions preset at the start of a program are identified by an *.

The traverse functions in the program are displayed when the ‘?’ key is pressed (except for the traverse functions which act block-by-block).

4. Traverse functions (continued)

Traverse functions		Effect
G00 G01* G02 / G03 G123	Positioning at rapid traverse Linear interpolation Circular interpolation, CW/CCW Automatic selection of linear and circular interpolation	modal modal modal modal
G04	Dwell	block-by-block
G05 / G06 / G07	Driving direction of round axes	block-by-block
G08 / G09	Precision stop, on/off	block-by-block
G10 / G11*	Polar coordinate programming, on	modal
G12	Contour path, rapid programming	block-by-block
G13*/ G14	Tangential axis off/on	modal
G15*/ G16	Polar transformation off/on	modal
G17 / G18 / G19	Plane selection XY / XZ / YZ	modal
G28 / G29*	Switching on/off precision stop	modal
G40*/ G41 / G42	Milling cutter path correction offset, clearing / left / right	modal
G43*/ G44	Axis correction off/on	modal
G45 G46*	Turning on/off	modal
GG47/G48/G49	Robot transformation off / Tool / workpiece coordinates	modal
G147	Transformation off	modal
G50/G51/G52	Spline interpolation	modal
G53* G153 G54...G59	Machine zero point Zero point shift off Zero points	modal modal modal
G60*/ G61 / G62	Mirror imaging, off	modal

4. Traverse functions (continued)

Traverse functions		Effect
G63 / G64*	Switching on feed rate 100%	modal
G66	Switching off all offsets	block by block
G90*	Absolute dimension input	modal
G91	Incremental dimension input	modal
G92	Zero point shift	modal
G94	Feed in mm/min	modal
G95	Feed in mm/r	modal
G96	Constant cutting speed	modal
G97	Number of revolution in 1/min	modal

4.1 G00 positioning at rapid traverse rate

Positioning at rapid traverse rate is called up by G00.

All axes can be traversed simultaneously as long as the machine tool is designed for this. Absolute and incremental dimension input are both possible.

G00 acts modally and can be overwritten by G01, G02, G03, G50, G51 and G52.

When traversing at rapid rate the programmed point is homed into via the shortest route.
The axis with the longest programmed traverse distance is traversed at rapid rate and determines the positioning time. The speed of the other axes is chosen by the control system such that they reach their end point simultaneously with the fastest axis.
The feed rate override potentiometer is also operative during positioning at rapid rate.

Any feed rate stored in front of G00 again becomes operative after a rapid traverse through G01, G02, G03, G50, G51 and G52.

Example: positioning at rapid rate

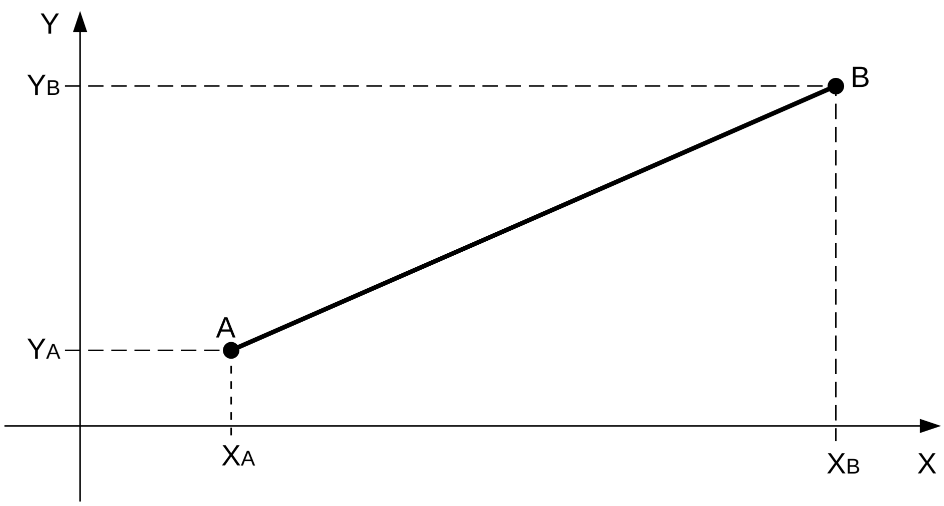


Figure 4-1

4.2 G01 Linear interpolation

The programmed target position is homed into by G01 along a straight line. The programmed feed rate is the contouring rate.

G01 acts modally and can be overwritten by G00, G02 and G03. G01 is automatically preset at start of the program.

Example: linear interpolation

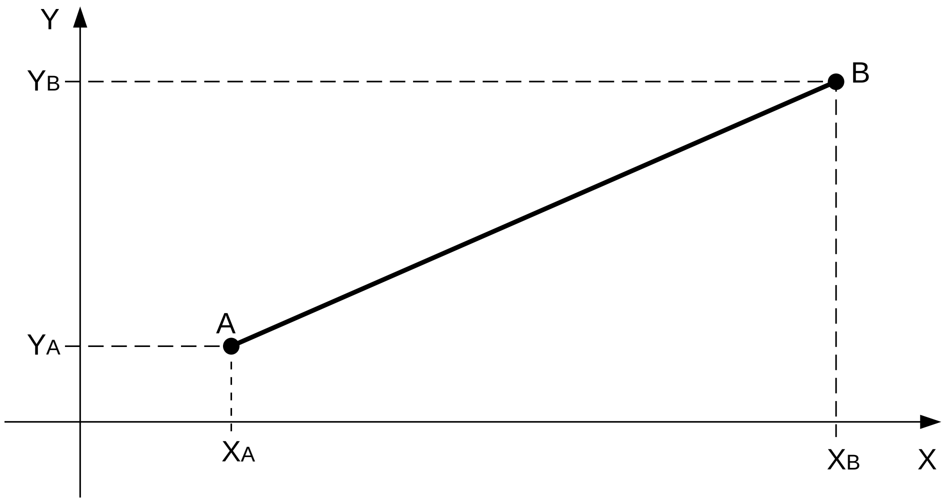


Figure 4-2

4.3 G02 / G03 Circular interpolation

When inputting G02 and G03, the programmed target position is homed into along a circle segment having a centre point determined by the interpolation parameters I, J and K and/or having a radius determined by R.

G02 signifies clockwise circular interpolation and
G03 signifies counter-clockwise circular interpolation

The G02 / G03 functions act modally and overwrite each other mutually and can be cleared by G00, G01, G50, G51 and G52.

Circle centre point

The circle limit point and the interpolation parameters can be input simultaneously using either absolute or incremental dimensions. For complete circle programming the limit point is equal to the starting point.

Interpolation parameter address letter	Circle centre point distance in direction
I	X axis
J	Y axis
K	Z axis

The speed at which the circle is started is proportional to the stored feed rate which can be adjusted between 0 and 120% by the feed rate override potentiometer.

4.3 G02 / G03 Circular interpolation (continued)

The interpolation parameters can be input in such a way that the deviation at A is less than or equal to 10 increments. If A is greater than 10 increments the program is still not shut down. In all cases the control system recalculates the circle centre point where R = (R1 + R2)/2.

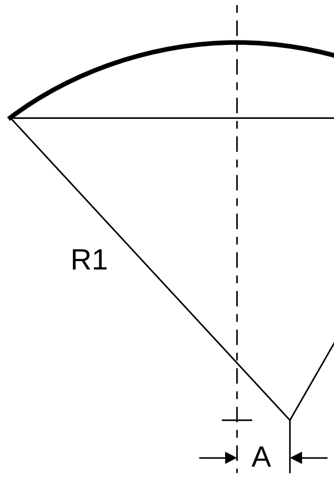


Figure 4-3

Three-dimensional circular interpolation

A programmed circle is three-dimensional driven off, if in a block 3 axes (X, Y, Z) and 3 circle center points (I, J, K) are indicated.

With programmed
G02 is executed a long arc,
G03 is executed a short arc.

The area levels G17, G18 and G19 are here without meaning.

4.3 G02 / G03 Circular interpolation (continued)

The main planes for the circular interpolation and tool correction offsets are selected by G17, G18 and G19 (see also 4.12 Plane selection).

Main plane	Parameters for circle centre point
XY	IJ
ZX	KI
YZ	JK

Example: clockwise circular interpolation

End point and circle centre point are programmed in absolute dimensions (G90).

A = Starting point
B = End point

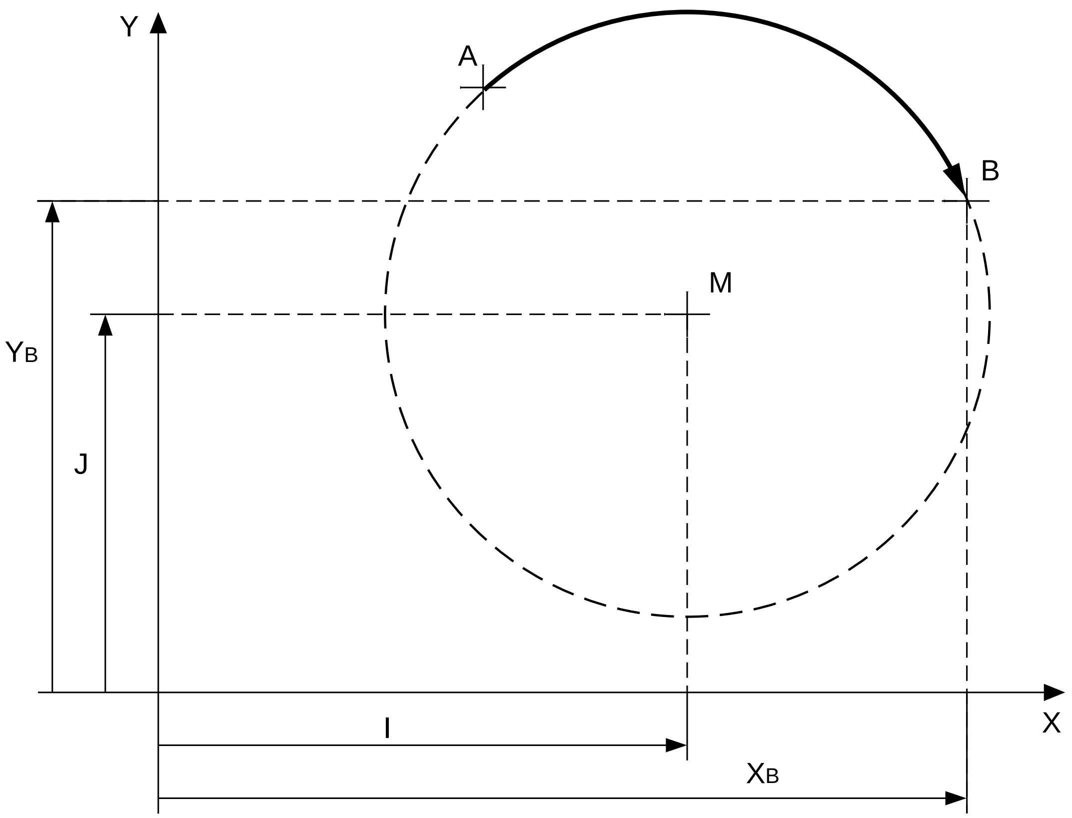


Figure 4-4

4.3 G02 / G03 Circular interpolation (continued)

The circle end point can be input in absolute or incremental dimensions. Full circle programming is not permitted.

The smaller and larger angular paths are described by positive and negative radius information respectively.

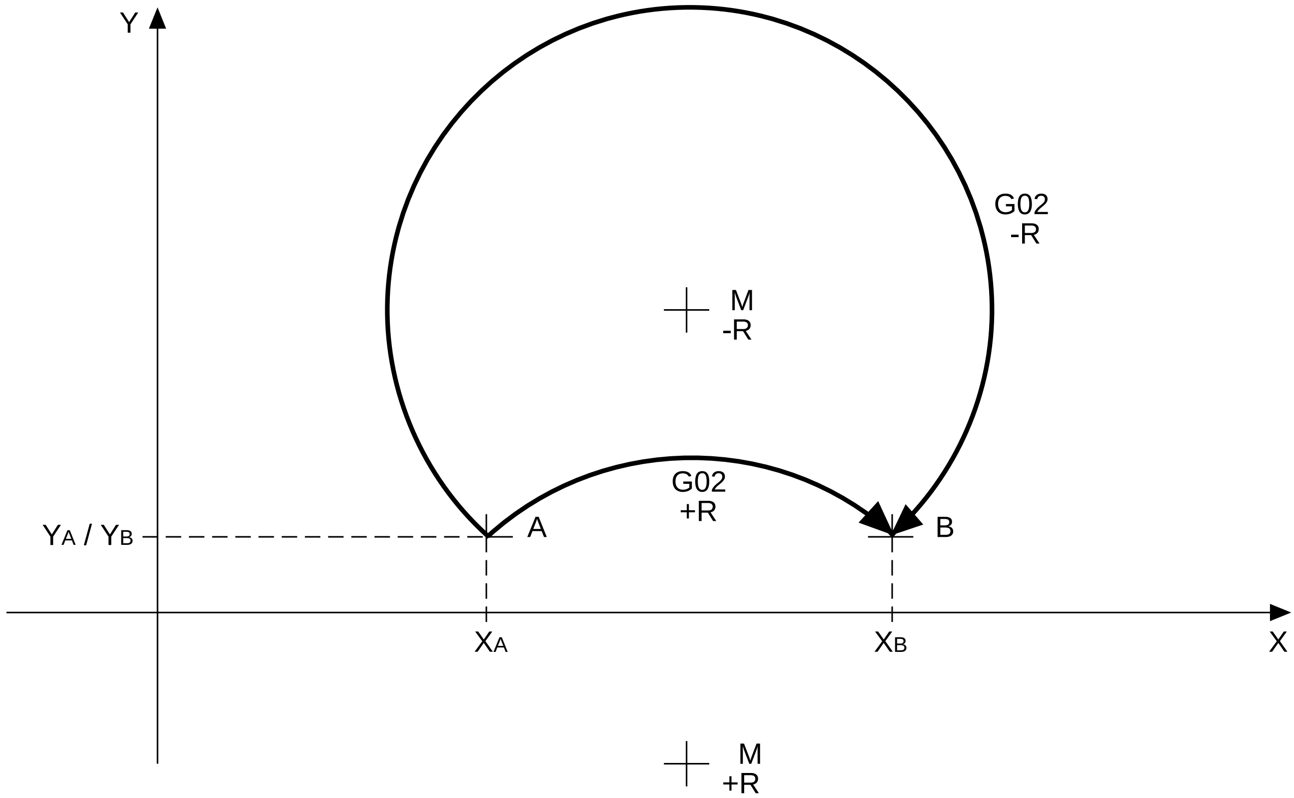


Figure 4-5

Radius R must be greater than or equal to the semi-chord AB/2 in order to give a centre point. If the R being input is less than the semi-chord, message 1407 is initiated and the control system selects an R = AB/2 without the program being shut down.

4.4 Automatic selection of linear and circular interpolation

G123 selects automatically after analysis of the position of three points G01 / G02 / G03:

1st point	position
2nd point	programmed position in the actual block
3rd point	programmed position in the next block

Example G123 -> G01

N10	G01	X20	Y50	position
N20	G123	X70	Y45	position in actual block
N30		X130	Y40	position in next block

Example G123 -> G02

N10	G01	X20	Y50	position
N20	G123	X70	Y71,5	position in actual block
N30		X130	Y63	position in next block

Example G123 -> G03

N10	G01	X20	Y50	position
N20	G123	X70	Y18	position in actual block
N30		X130	Y18	position in next block

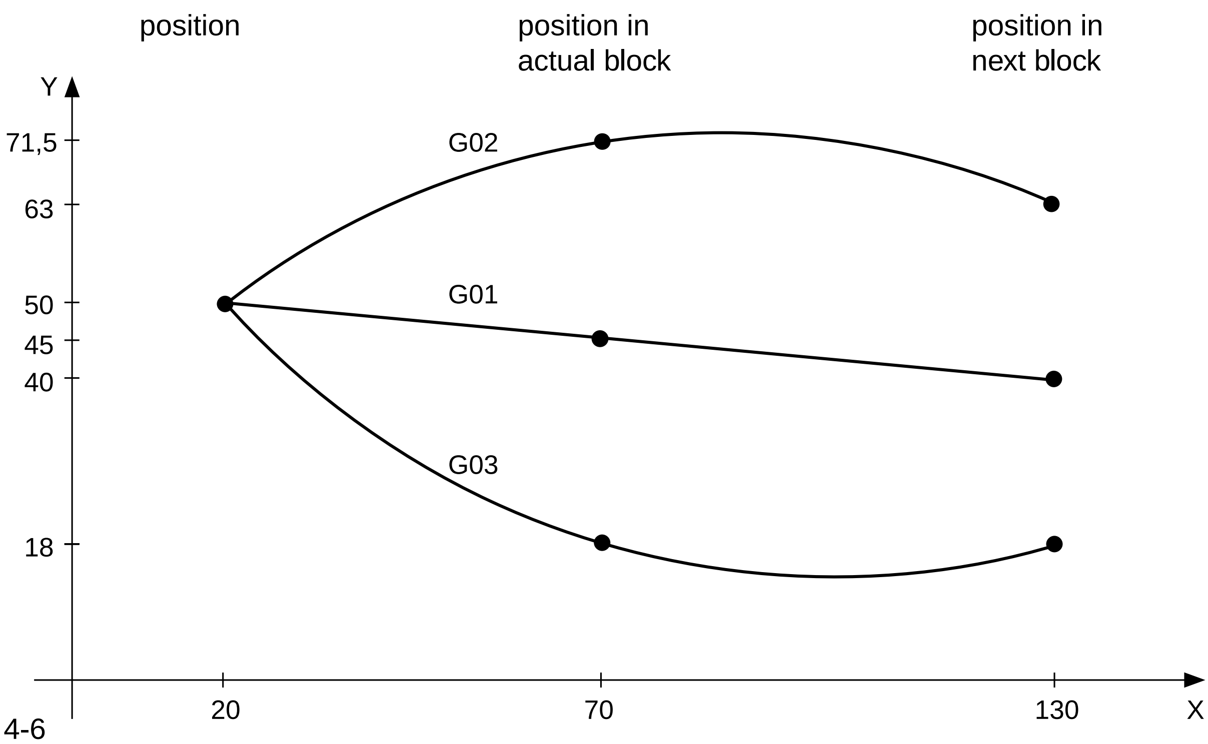


Figure 4-6

4.4 Automatic selection of linear and circular interpolation

Program example

```
P2050
N2                                                        { G123 free outline }
N10   G49 G56 G0 F3000   X:0        Y:0       Z:2
N20                      X:21.1     Y:9.300   Z:-1        { point 1 }
N30   G123 G1            X:19.6     Y:54.3                { point 2    selection G02 }
N40                      X:45.0     Y:61.0                { point 3 }
N50   G123               X:63.0     Y:46.7                { point 4    selection G01 }
N60                      X:82.9     Y:30.5                { point 5 }
N70   G123               X:81.2     Y:18.8                { point 6    selection G03 }
N80                      X:108.8    Y:21.6                { point 7 }
N90   G123               X:101.0    Y:33.0                { point 8    selection G02 }
N100                     X:110.3    Y:64.4                { point 9 }
N110  G123               X:108.2    Y:40.6                { point 10   selection G03 }
N120                     X:131.8    Y:25.5                { point 11 }
N130  G123               X:135.7    Y:13.8                { point 12   selection G01 }
N140                     X:138.447  Y:4.4                 { point 13 }
N150                                          Z:2
N160  M30
```

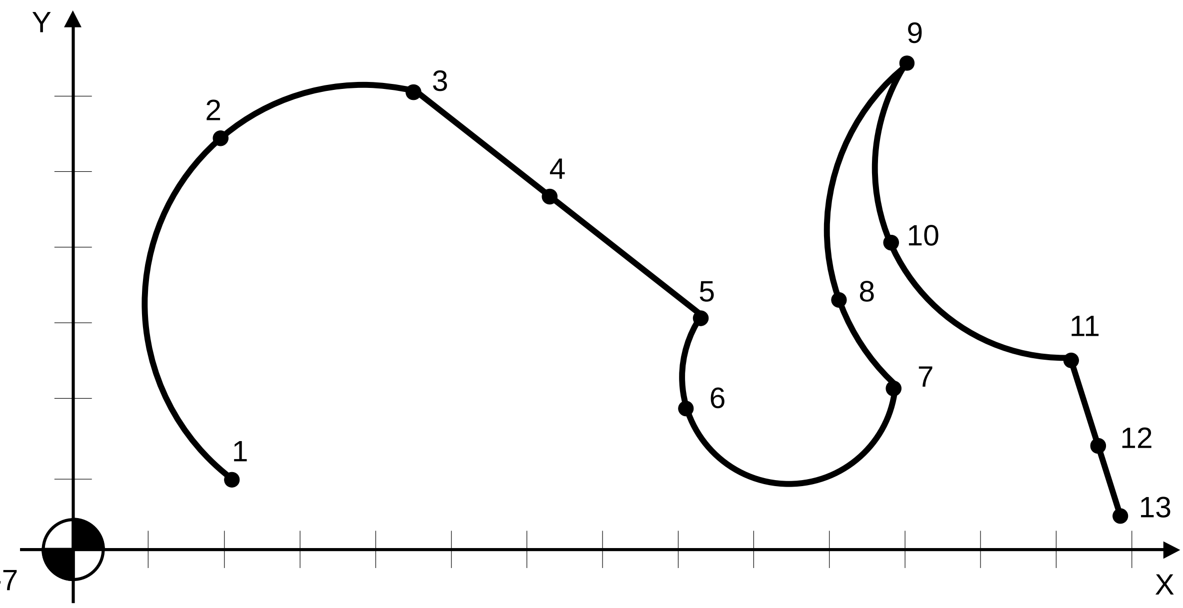


Figure 4-7

4.5 G04 Dwell

A dwell is called up by G04 and can be programmed between 0 and 999.999 s. The dwell must be input in the correct format, i.e. leading and trailing zeros and the decimal point should be input.

Example: G04.001.50 (1.5 s dwell)

The G04 function is only operative in the block in which it was written. G04 is operative after the traverse information and before the M functions subsequently active.

4.6 G05 / G06 / G07 driving direction of round axes

G5 driving direction positively, against clockwise direction.
G6 driving direction negatively, in the clockwise direction.
G7 driving direction neg. / pos., selection shortest path to the target position.

Example: G5 driving direction positively against clockwise direction

NC-Program				Start	Target	Driving direction	Drive	Display
N10	G5	G90	C:90	C:0	90	pos.	90	90
N20	G5	G90	C:270	C:90	270	pos.	180	270
N30	G5	G90	C:35	C:270	35	pos.	125	35
N40	G5	G90	C:10	C:35	10	pos.	335	10
N50	G5	G91	C:30	C:10	40	pos.	30	40
N60	G5	G90	C:150	C:40	150	pos.	110	150
N70	G5	G91	C:-150	C:150	0	neg.	150	0

Example: G6 driving direction negatively in the clockwise direction

NC-Program				Start	Target	Driving direction	Drive	Display
N10	G6	G90	C:90	C:0	90	neg.	270	90
N20	G6	G90	C:270	C:90	270	neg.	180	270
N30	G6	G90	C:35	C:270	35	neg.	235	35
N40	G6	G90	C:10	C:35	10	neg.	25	10
N50	G6	G91	C:30	C:10	40	pos.	30	40
N60	G6	G90	C:150	C:40	150	neg.	250	150
N70	G6	G91	C:-150	C:150	0	neg.	150	0

Example: G7 driving direction negatively / positively selection shortest path to the target position

NC-Program				Start	Target	Driving direction	Drive	Display
N10	G7	G90	C:90	C:0	90	pos.	90	90
N20	G7	G90	C:270	C:90	270	pos.	180	270
N30	G7	G90	C:35	C:270	35	pos.	125	35
N40	G7	G90	C:10	C:35	10	neg.	25	10
N50	G7	G91	C:30	C:10	40	pos.	30	40
N60	G7	G90	C:150	C:40	150	pos.	110	150
N70	G7	G91	C:-150	C:150	0	neg.	150	0

Note: With incremental the technique the functions G5/G6/G7 are ineffective.
The driving direction certainly by the sign.

4.7 G08 / G09 Precision stop block-by-block

G08 precision stop, block-by-block, On
G08 precision stop, block-by-block, Off

G08 initiates a precision stop at the end of a block.

The function overwrites a programmed G29 block by block. The precision stop is automatically set block by block with G00 and G81 to G85.

G09 overwrites a precision stop programmed with G28 block by block

4.8 G10 / G11 Polar coordinate system

G10 Polar coordinate system on
G11 Polar coordinate system off

The function G10 is used to activate the programming of target points in polar coordinates.
The function G11 deactivates this function. Both functions act modally.

The coordinates programmed in the block with G10 define the pole, but not the travel of the axes (this is only valid for the two coordinates of the interpolation plane defined with G17...G19). If the coordinates are not programmed, the existing pole is kept.

At the program end or if the program is interrupted, the programmed pole as well as the polar radius (x) and polar angle (c) are cleared. Polar radius (x) and polar angle (c) act modally.

The polar coordinate system plane corresponds to the interpolation plane defined with G17...G19.
The polar radius is always interpreted as a positive value!

Polar radius and polar angle can be corrected with G44.

4.8 G10 / G11 Polar coordinates (continued)

Definition of the poles in different interpolation planes

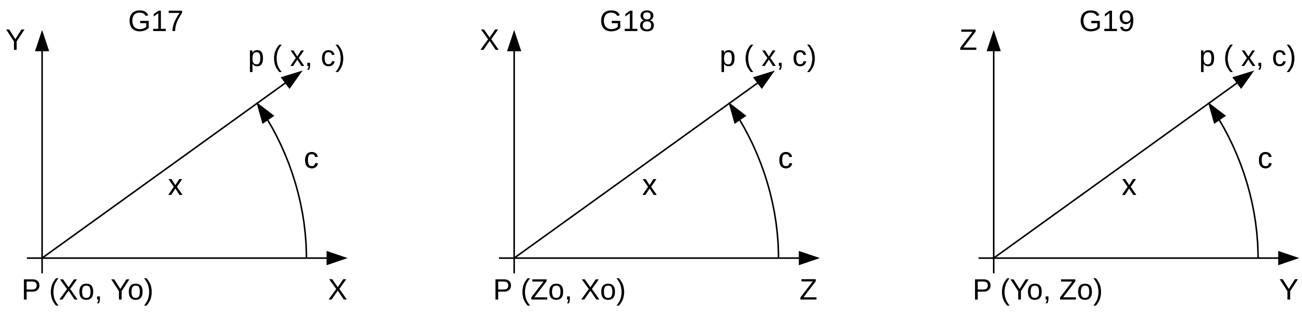


Figure 4-8

Parameters used for polar coordinate system

G10	Activating the polar coordinate input position with radius vector x and polar angle c	
G11	Deactivating the polar coordinates	
G17	XY plane	pole P() is in plane XY, the X axis is the polar axis
G18	ZX plane	pole P() is in plane ZX, the Z axis is the polar axis
G19	YZ plane	pole P() is in plane YZ, the Y axis is the polar axis

q150 Polar coordinate system: identification for radius vector (e.g. 'x')
Input of the ASCII-code Default: 'x'

q151 Polar coordinate system: identification for polar angle (e.g. 'c')
Input of the ASCII-code Default: 'c'

4.8 G10 / G11 Polar coordinates (continued)

Program example: Coordinates in X / Y

```
P656
N05   G55  G49   G00   F3000        T1      M16
N10                          X:0           Y:0             Z:2           { point 1 }
N20               G01                                      Z:-2
N30                          X:90          Y:0                           { point 2 }
N40                          X:60          Y:25                          { point 3 }
N50                          X:90          Y:60                          { point 4 }
N60               G03        X:30          Y:85            R:34          { point 5 }
N70               G02        X:45          Y:55            R:18          { point 6 }
N80               G01        X:0           Y:0                           { point 1 }
N90                                                        Z:2
N100       M30
```

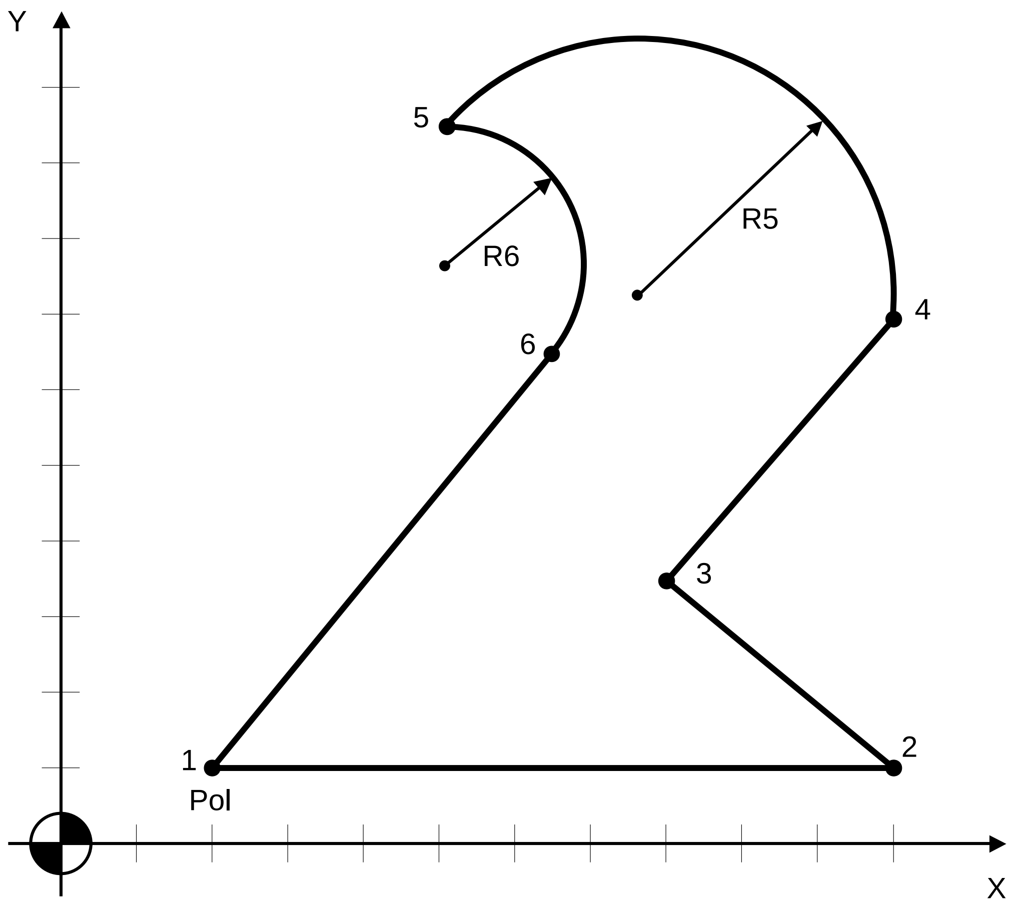


Figure 4-9

4.8 G10 / G11 Polar coordinates (continued)

Program example: Point of pole is the null point X:0 / Y:0

```
P657
N05   G55 G49  G00  F3000     T1      M16
N10                      X:0       Y:0       Z:2       { point 1 }
N20            G01                           Z:-2
N30       G10            X:0       Y:0                 { point 1  pole set }
N40                      x:90      c:0                 { point 2 }
N50                      x:65      c:22.619            { point 3 }
N60                      x:108.166 c:33.690            { point 4 }
N70            G03       x:90.138  c:70.559  R:34      { point 5 }
N80            G02       x:71.063  c:50.710  R:18      { point 6 }
N90            G01       x:0       c:0                 { point 1 }
N100      G11                                Z:2
N110      M30
```

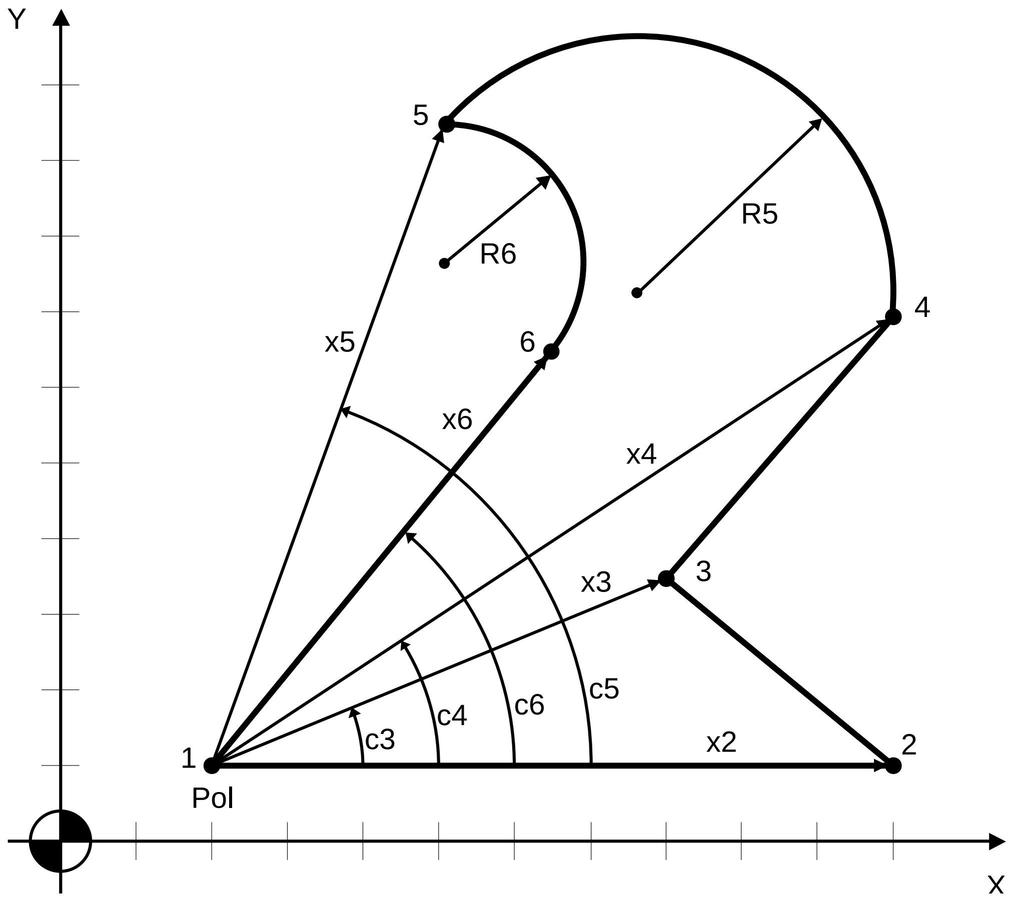


Figure 4-10

4.8 G10 / G11 Polar coordinates (continued)

Program example: Point of pole are thr points 2, 3 und 6

```
P659
N05   G55 G49  G00  F3000       T1     M16
N10                      X:0         Y:0          Z:2           { point 1 }
N20              G01                              Z:-2
N30                      X:0         Y:0                        { point 2 }
N40         G10          X:90        Y:0                        { point 2 pole set }
N50                      x:39.05     c:140.194                  { point 3 }
N60                      X:60        Y:25                       { point 3 pole set }
N70                      x:46.09     c:49:398                   { point 4 }
N80         G11  G03     X:30        Y:85         R:34          { point 5 pole reset }
N90              G02     X:45        Y:55         R:18          { point 6 }
N100        G10          X:45        Y:55                       { point 6 pole set }
N110             G01     x:71.062    c:230.710                  { point 1 }
N120        G11                                   Z:2           { point 1 pole reset }
N130        M30
```

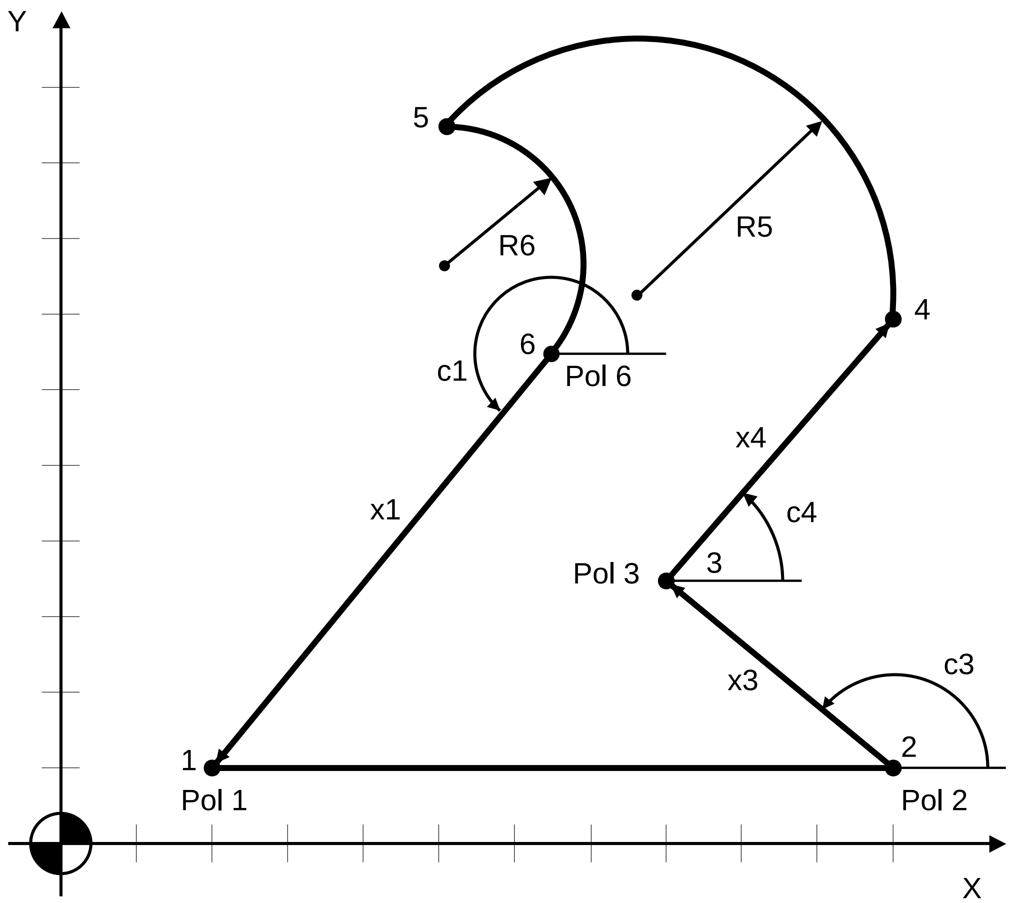


Figure 4-11

4.9 G12 Rapid graphical programming environment

Contour sections can be linked by of straight lines and circles.

This function acts block-by-block.

Parameter	Leg	Radius
	P70	P71

Rapid programming is only carried out with blocks in which the distance to be traversed is unequal to zero (differences in distances are unequal to zero for straight lines, radius is not equal to zero for circles).
Blocks with distances equal to zero are carried out at the start of the chamfer or circle segment.

Linear programming

A chamfer can be inserted at the intersection of two straight lines.
The length of the chamfer is given by P70 and the intersection point SP is programmed in the selected interpolation plane.

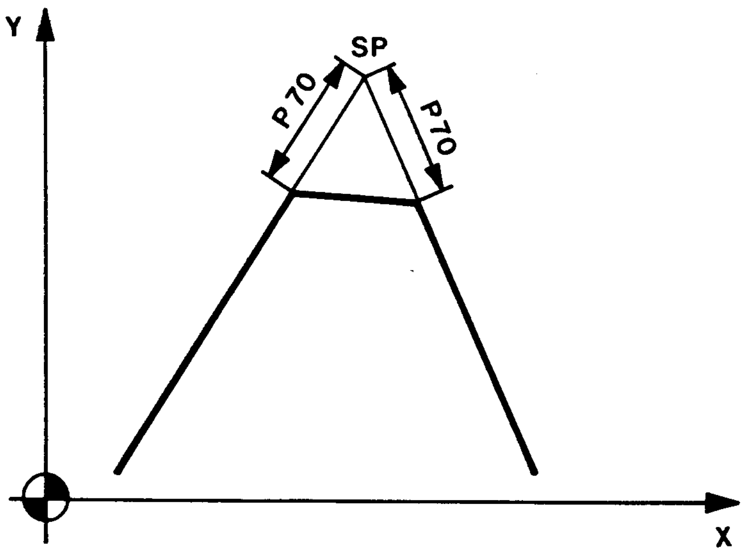


Figure 4-12

4.9 G12 Rapid graphical programming environment (continued)

Radius programming

A circle segment can be inserted at the intersection point of
- two straight lines,
- one straight line and one circle,
- two circles.

The radius of the circle is given by P71 and the intersection point SP is programmed in the selected interpolation plane.

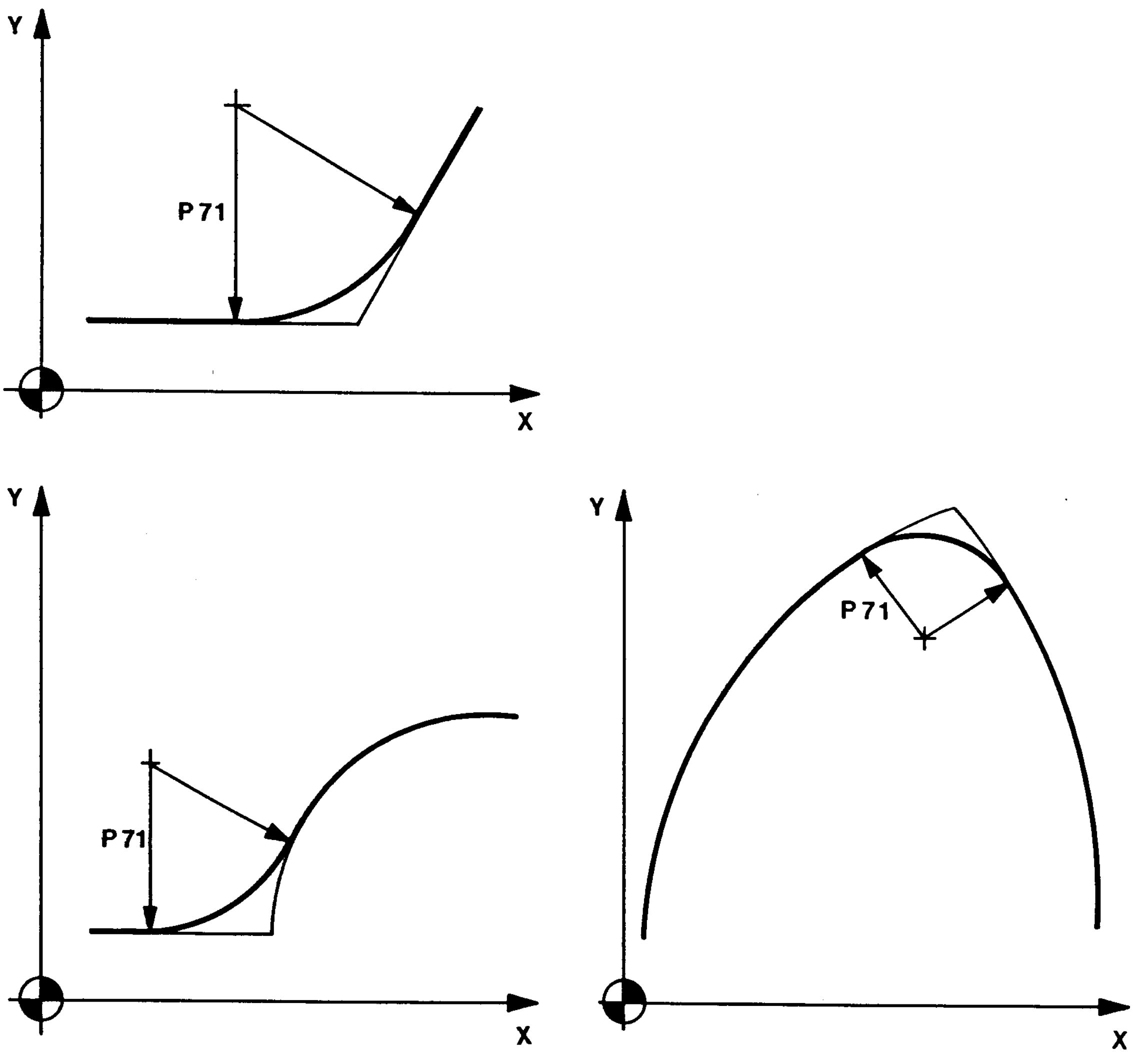


Figure 4-13

4.9 G12 Rapid graphical programming environment (continued)

Program example: Outline path short programming with radius and leg

```
P391
N10     G55  G29  G48  G00   F2000  T1   M16  X0      Y0    Z2
N20          G01                                            Z:-1
N30     G12            P71:8                  X:40    Y:0         { point 1  radius }
N40     G12  G02  R65  P71:28                 X62.5   Y:91        { point 2  radius }
N50     G12  G02  R65  P71:15                 X95     Y:35        { point 3  leg }
N60     G12  G01       P71:8                  X120    Y:35        { point 4  radius }
N70     G12  G02       P71:12       I95  J35  X95     Y:10        { point 5  radius }
N80     G12  G03       P71:10       I70  J10  X45     Y:10        { point 6  radius }
N90          G01                              X34.5   Y:10        { point 7  radius }
N100                   G00                                  Z:2
N110    M30
```

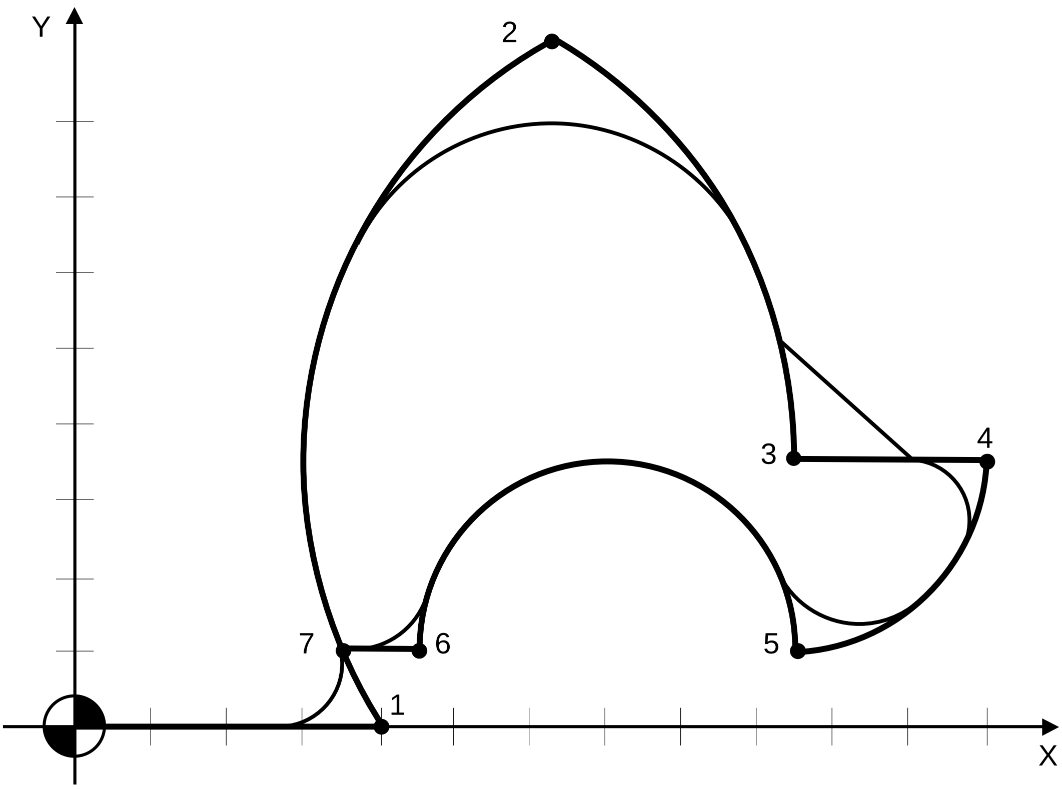


Figure 4-14

4.10 G13 / G14 tangential axis

G13 tangential axis off
G14 tangential axis on

G13 switches the tangential axis off.

G14 switches the tangential axis on.

With this function it is possible to control a round axis so that it is always in a certain position (tangential) to the path of the main axes.
For example for band saws, glass cutting, moist collectors.

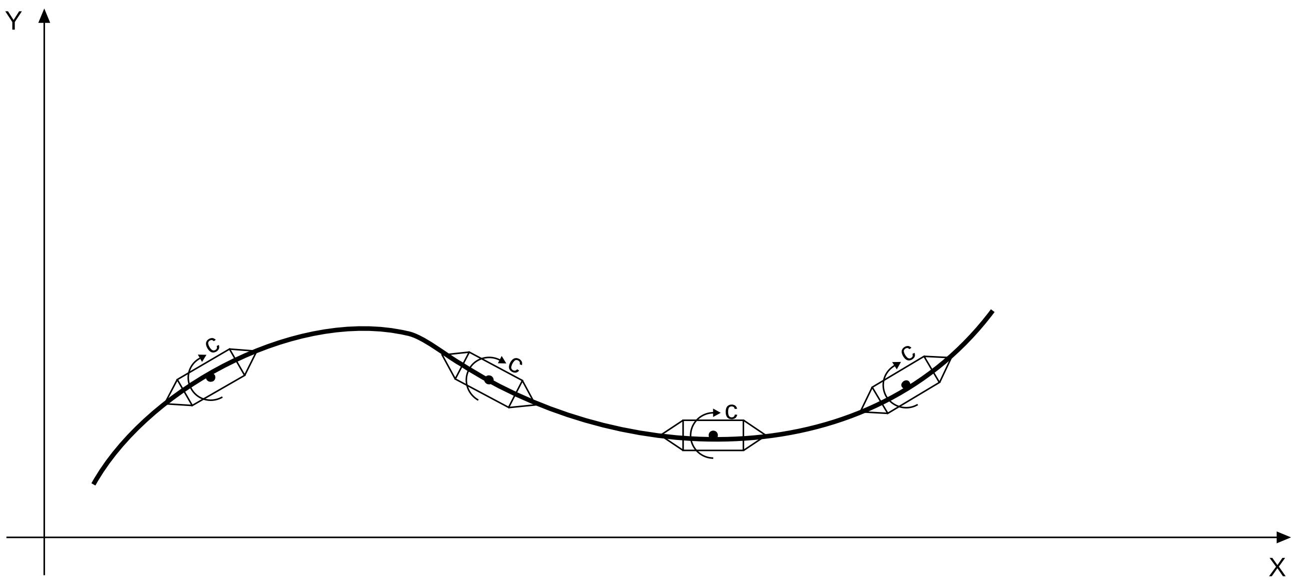


Figure 4-15

4.11 G15 / G16 Polar transformation

G15 Polar transformation off
G16 Polar transformation on

G15 switches the polar transformation off.

G16 switches the polar transformation on.

The polar transformation is applied for example for programming of uneven parts on a grinding machine or lathe with polar coordinate system. Usually the workpiece can be described more easily in the cartesian coordinate system. With the polar transformation G16 the control converts the cartesian coordinates (X/Y/Z) into polar coordinates (radius vector x and polar angle c). This method is very advantageous, because it requires less time for contour featuring.

Definition of the polar coordinate system

With the polar coordinates each point of the coordinates are determined by the two values radius vector ‘x’ and polar angle ‘c’.

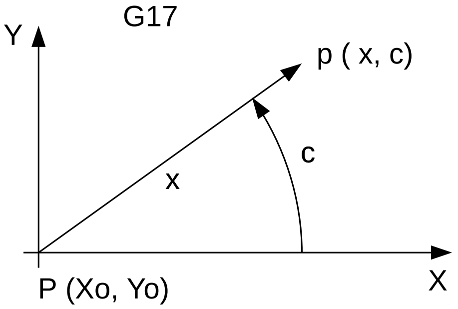


Figure 4-16 Polar coordinates with G 17 (XY - plane)

The radius vector x is the distance of point p(x,c) to the origin P(Xo,Yo).
The polar angle c is between the X-axis and the connecting straight line from the origin P(Xo,Yo) to point p(x,c).
The X - axis is marked as polar axis and the origin P(Xo,Yo) as pole. The polar angle is positive, if it is measured from the pole axis X counter-clockwise direction. The angle is negative in clockwise direction.

4.11 G15 / G16 Polar transformation (continued)

Utilized parameter for polar coordinate system

G15	Selecting the polar coordinate transformation
G16	Call up of the polar coordinate transformation, input position with X, Y

G17	XY plane	pole P() lies in XY Plane, X axis is polar axis
G18	ZX Plane	pole P() lies in ZX Plane, Z axis is polar axis
G19	YZ Plane	pole P() lies in YZ Plane, Y axis is polar axis

P11820 Polar coordinate axis (physical axis number)
e.g.
Byte 3, 2, 1: the first 2 axes (x, y) 00 02 01 Hex
Byte 4 : change of direction (Bit 2, 1)

P11821	Radius vector offset (ro) for polar coordinate transformation	[mm]
P11822	Polar angle offset (wo) for polar coordinate transformation	[degrees]
P11823	Pole offset (po) for polar coordinate transformation	[mm]

4.11 G15 / G16 Polar transformation (continued)

Example: Polar transformation G16 for square contour

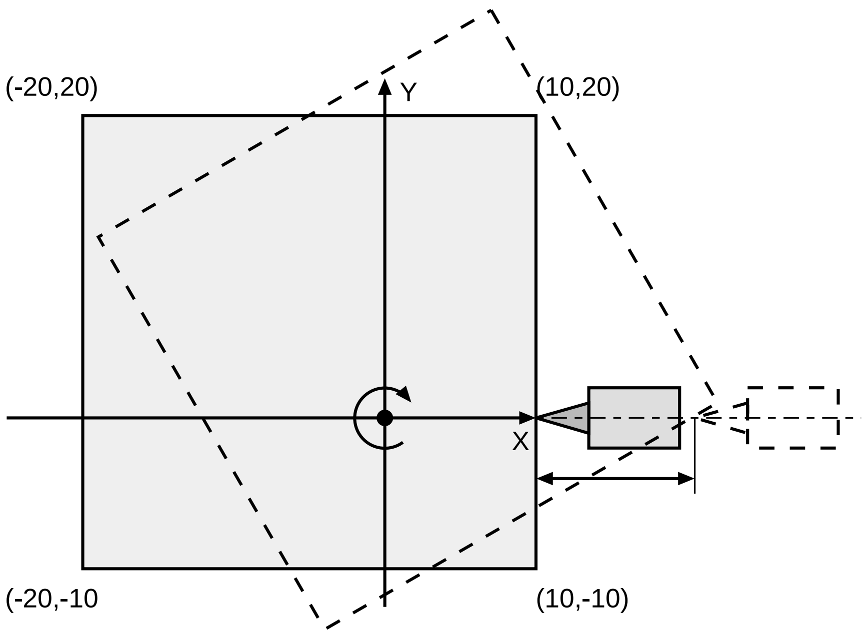


Figure 4-17 Square contour

```
N10  G17 G1 F1000   ;   Definition of coordinate plane
. ..
N150 G16 X10 Y0     ;   Call up polar transformation and X,Y
                    ;   are transformed accord. equation 2 into polar coorinates
                    ;   here: Xo = Yo = 0;
N160  X10 Y 20      ;   Position X= 10 and Y= 20 are transformed into
                    ;   polar coordinates. Approach to transformed position
N170  X - 20 Y 20   ;   X= -20 and Y= 20 are transformed into polar coordinates.
                    ;   Approach to transformed position
N180 X - 20 Y-10    ;   X= -20 and Y= -10 are transformed into polar coordinates.
                    ;   Approach to transformed position
N190 X10 Y-10       ;   X= 10 and Y= -10  are transformed into polar coordinates.
                    ;   Approach to transformed position
N200 X10 Y 0 G15    ;   X= 10 and Y= 0 are transformed into polar coordinates.
                    ;   Approach to transformed position
                    ;   Switch off  polar transformation.
N220 . . .
N230 M30            ;   Program end
```

4.11 G15 / G16 Polar transformation (continued)

Example: Polar transformation G16 for cam contour

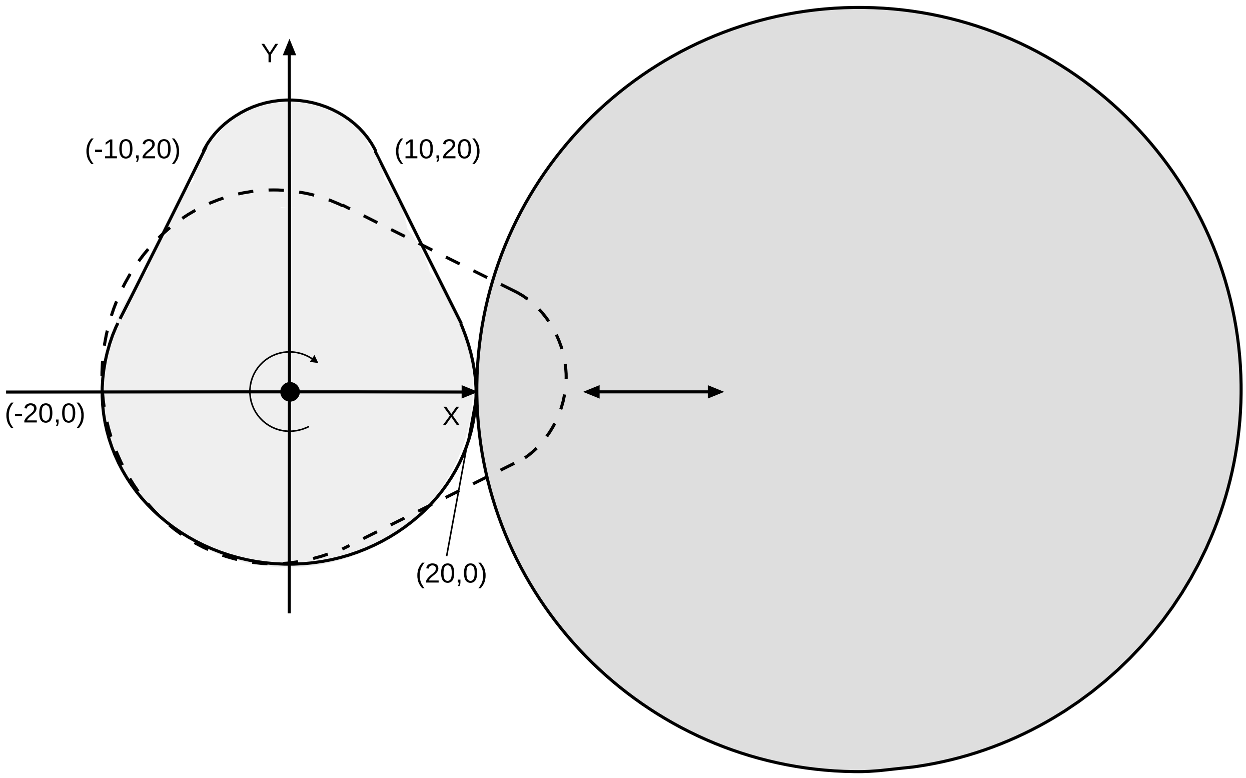


Figure 4-14 Cam contour

N10 G17 G1 F1000	;	Definition coordinate plane
. ..		
N150 G16 X20 Y0	;	Call up polar transformation and X,Y
	;	are transformed accord. equation 2 into polar coorinates
	;	here: Xo = Yo = 0;
N160 X10 Y 20	;	Position X= 10 and Y= 20 are transformed into
	;	polar coordinates. Approach to transformed position
N170 G2 X -10 Y 20 R11	;	X= -10 and Y= 20 are transformed into
	;	polar coordinates. Approach to G2 until transformed pos. N180
G1 X -20 Y0	;	X= -20 and Y= 0 are transformed into polar coordinates
	;	Approach to transformed position
N190 G2 X20 Y0 R22	;	X=20 and Y= 0 are transformed into polar coordinates
	;	Approach to G2 until transformed position
N200 G15	;	Switch off polar transformation.
N220 . . .		
N230 M30	;	Program end

4.12 G17 / G18 / G19 Plane selection

The main planes for circle interpolation and tool correction offsets are selected using G17, G18 and G19.

Traverse functions	Main plane	Parameters for circle centre point
G17	XY	IJ
G18	ZX	KI
G19	YZ	JK

These functions act modally and overwrite each other mutually. The plane preset is controlled by the PLC parameter „WZ“ tool axis.
The tool correction offset takes place in the given interpolation plane. The correction offset should be cancelled using G40 before changing the interpolation plane.

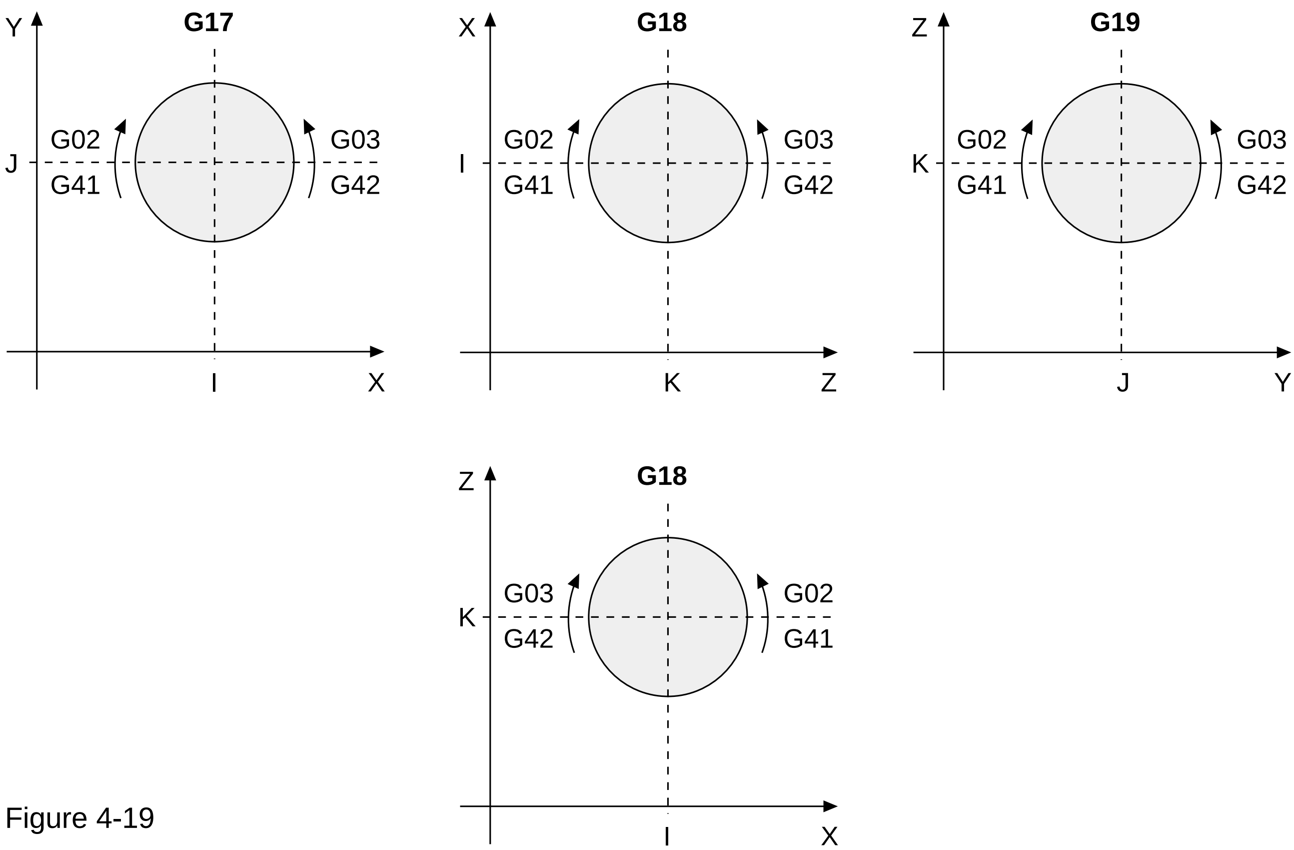


Figure 4-19

4.12 G17 / G18 / G19 Plane selection (continued)

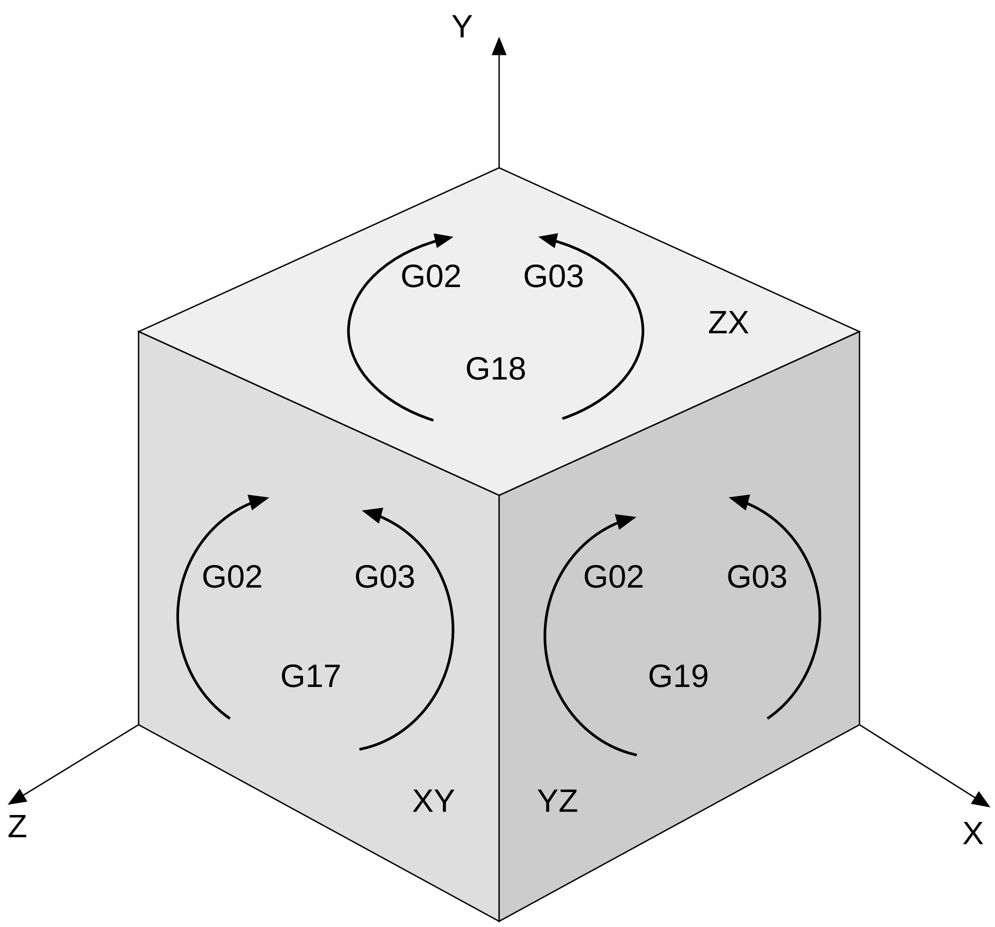


Figure 4-20

4.13 G28 / G29 Precision stop modal

G28 Switching on modal precision stop

G28 initiates a precision stop at the end of a block. The tool traverses to the end point programmed in the block and stops.
Subsequently it traverses to the end point programmed in the next block and stops again.

The function acts modally and can be cleared by G29.

G29 Switching off modal precision stop

G29 switches off a precision stop programmed with G28.

The function acts modally and can be cleared by G28.

G29 is automatically set when the program starts.

4.14 G40 / G41 / G42 Milling cutter path correction offsets

G40 Clearing milling cutter path correction offsets

The milling cutter path correction offset programmed using G41 or G42 is cleared using G40.

G40 acts modally and can be cleared by G41 and G42. G40 is automatically set when the program starts.

G41 / G42 Actuating milling cutter path correction offset

A milling cutter path correction offset can be switched on using G41 and G42.

G41 effects a correction to the left of the programmed path.
G42 effects a correction to the right of the programmed path.

The reference direction is in the feed direction in each case.

The functions act modally, overwrite each other mutually and can be cleared by G40.

The feed rate is proportional to the milling cutter centre point path when the tool correction offset is switched on.

Further details on this subject appear in the section „Tool correction offset“.

4.15 G43 / G44 Axis correction

With G43 / G44 an axis correction can be programmed.

G43 axis correction off
G44 axis correction on

The programmed traverse informations in the block with G44 determine the axis correction, whereby no traverse is driven.
These corrections remain modally.

At program end or program abort these corrections are cleared, or switched off with G43.
A programmed G44 correction is a component, which is added on all further traverse information.

With G44 polar angles and polar radius can also be corrected.

Input: e.g.:

N100 G44 X10 Y20 Z30 axes do not drive !

N120 G01 F1000 X100 Y100 Z100 axes drive to X: 110, Y: 120; Z: 130

N130 G43

4.16 G45 / G46 Turning of coordinates

The „Turn“ function initiates the turning of a coordinate pair about a given angle E.
The traverse function G45 switches turning on, G46 switches the function off. G45 and G46 act modally.

G45 / G46 Geometric definition turning

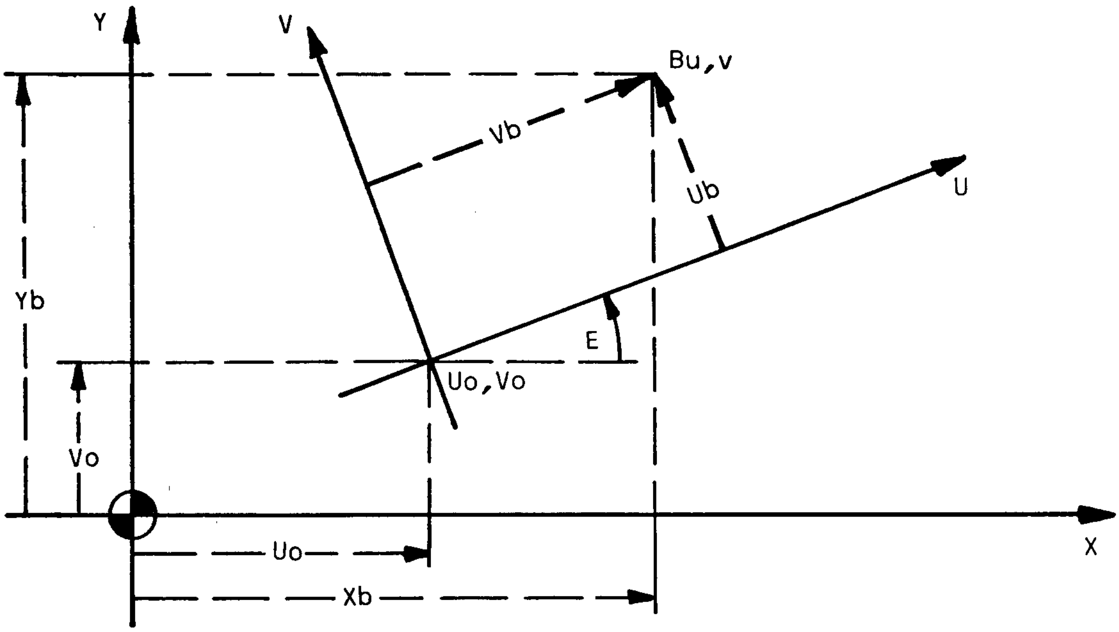


Figure 4-21

X, Y Machine coordinate system
U, V Coordinate axes of the turned coordinate system

E Angle about which the machine system X axis is to be turned in the direction of the machine system Y axis in order to retain the UV coordinate system orientation.
Angle E is positive if measured counter-clockwise to the X axis and negative in the clockwise direction.

Uo, Vo Original UV coordinate system coordinates, referred to the machine XY coordinate system zero point.
Ub, Vb Point B coordinates in UV coordinate system.
Xb, Yb Point B coordinates in XY coordinate system, referred to the effective shift (NP and G92).

Converting the UV system coordinates into those of the XY system
Xb = (U.cosE - V.sinE) + Uo
Yb = (U.sinE + V.cosE) + Vo

4.16 G45 / G46 Turning of coordinates (continued)

Programming the coordinate origins U0, V0 (referred to the machine system zero point) and the turning angle E take place via parameters.

Parameter	Characteristic	Significance
P160	Uo	UV system original coordinate
P161	Vo	UV system original coordinate
P163	E	Turning angle

The parameters act modally.

The parameters P160, P161 and the reference axis for the turning angle are dependant on the plane set.

Plane set	P160	P161	Reference axis for turning angle
G17 (XY)	Xo	Yo	X axis
G18 (ZX)	Zo	Xo	Z axis
G19 (YZ)	Yo	Zo	Y axis

Shifts with additive effect

In addition to the adjustment values P160, P161, P163, shifts can be programmed which act additively with respect to the direction.

Parameter	Characteristic	Significance
P165	U	Shift in U
P166	V	Shift in V
P168	E	Turning angle incrementation

When one of these parameters is called up, it's value is added to the former value of the corresponding shift. The value obtained in this way acts modally. The original parameters are retained.

Cancellation of the additive shifts is by re-calling up the adjustment values P160, P161 or P163 (or by switching G45 off with G46).
The additive shifts are plane-dependent, as are the adjustment values.

4.16 G45 / G46 Turning of coordinates (continued)

Mirror imaging of the turned coordinate system

The mirror functions G61 and G62 are referred to the UV coordinate system:
G61 mirror images the U axis
G62 mirror images the V axis

The mirror imaged coordinates are expressed in the machine system XY in the following manner:
Xb = - (U.cosE - V.sinE) + Uo
Yb = - (U.sinE + V.cosE) +Vo

The effect on G92

The shifts programmed using G92 are carried out.

The effect on G66 on the turning function

G66 initiates the block-by-block switching off (G46) of turning G45.

Turning the circle centre point coordinates I, J, K

Turning the circle centre point coordinates I, J, K takes place according to the turning of coordinates as described above.

4.16 G45 / G46 Turning of coordinates (continued)

Example 1 Repeated machining with turning

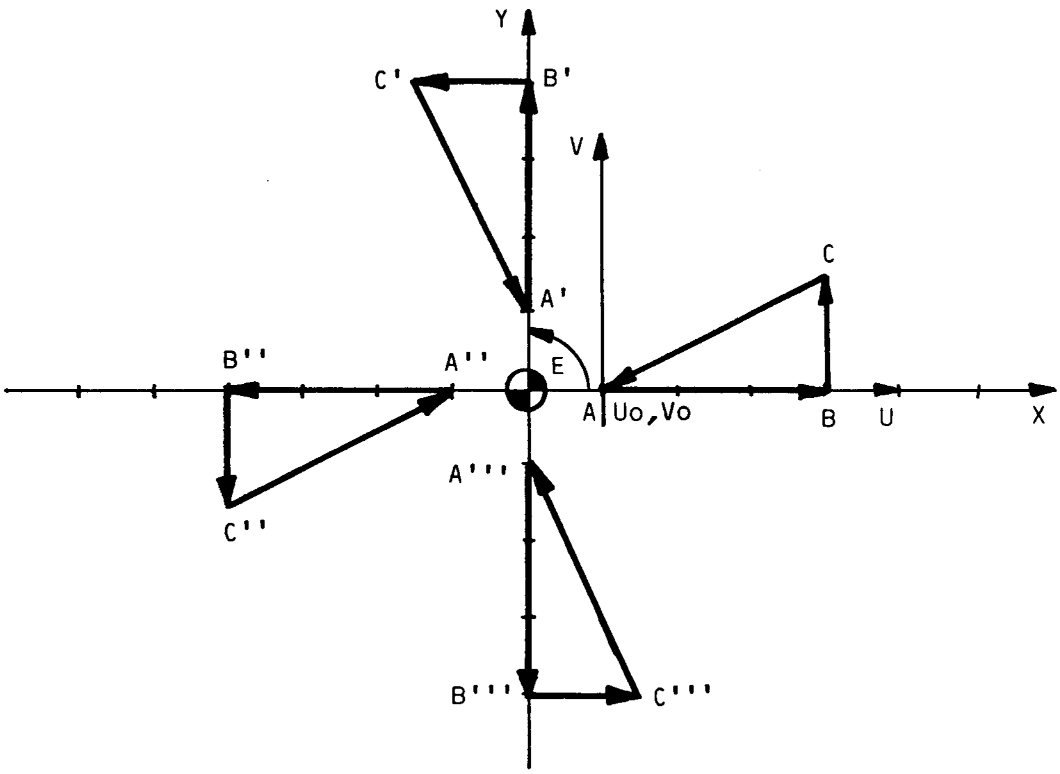


Figure 4-22

```
P1    N10   G45   G00   G54   Z2     P160:0      P161:0      P163:0      F2000       M24.4
N20   G00   X10   Y0
N30   G01   Z-1,5
N40   X40
N50   Y15
N60   X10   Y0
N70   Z2    P168:90     M25
N80   M30
```

The triangular shape ABC is to be machined four times with appropriate orientation of the +X, +Y, -X, -Y coordinate axes.

The machining sequence on the triangle is:
1st. Traverse A to B
2nd. Traverse B to C
3rd. Traverse C to A

4.16 G45 / G46 Turning of coordinates (continued)

Program run, example 1

Block 10 Switching on the coordinate turning function (G45)

Input set-up data U0=10 (P160) Vo=0 (P161) E=0 (P163)
Uo, Vo coincide with point A
Approach to Z safety dimension, Loop input

Block 20 Positioning to point A = Uo, Vo

Block 30 Tool axis in-feed to machining depth

Block 40 Machining along line AB

Block 50 Machining along line BC

Block 60 Machining along line CA

Block 70 Traversing tool axis away for positioning, incrementing the turning angle to +90 deg., skip back to Block 20

Block 20 Positioning to point „A“

Block 30 In-feed to machining depth
.
.
Block 70

Block 20 Point „A“ Machining, x-orientated
.
.
Block 70

Block 20 Point „A“ Machining, y-orientated
.
.
Block 70

Block 80 End of machining operations

The final machine position is at point „A“.

4.16 G45 / G46 Turning of coordinates (continued)

Example 2 Repeated machining with shift

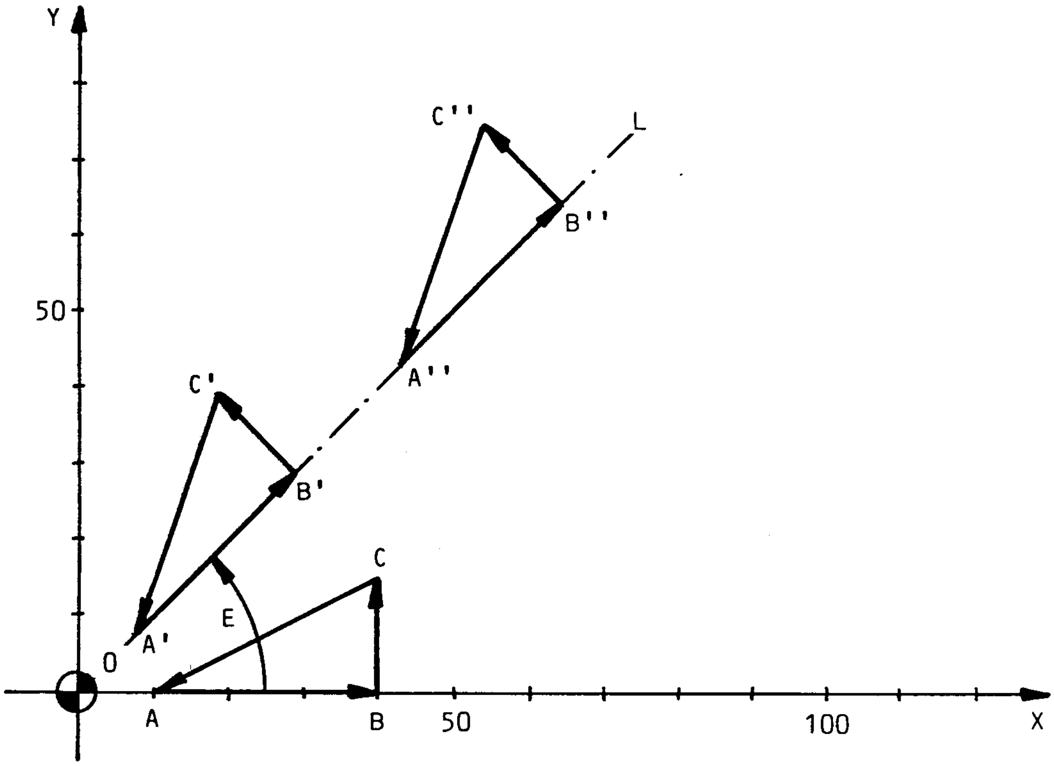


Figure 4-23

```
P2  N10  G45  G0   Z0     P160:0    P161:0    P163:45   F2000    M24.2
    N20  G00  X10  Y0
    N30  G01  Z50
    N40  X40
    N50  Y15
    N60  X10  Y0
    N70  Z70  P165:55   M25
    N80  M30
```

The basic triangular shape ABC is to be machined twice along line OL. Line OL makes an angle E to the axis. The distance between the two triangles A', B', C' and A", B", C" is 55mm (distance A'-A").

The machining sequence on the triangle is:
1st. Traverse A to B
2nd. Traverse B to C
3rd. Traverse C to A

4.16 G45 / G46 Turning of coordinates (continued)

Program run, example 2

Block 10	Switching the coordinate turning function on (G45)
	Set-up data input Uo=10 (P160) Vo=0 (P161) E=45 (P163)
	Approach to Z safety margin, Loop input
Block 20	Positioning to point A'
Block 30	Tool axis in-feed to machining depth
Block 40	Machining along line A' - B'
Block 50	Machining along line B' - C'
Block 60	Machining along line C' - A'
Block 70	tool axis retraction for positioning, shifting the Uo-value P165:55, skip back to Block 20
.	
.	
.	
.	
Block 20	Positioning to point A“
Block 30	Tool axis in-feed to machining depth
Block 40	Machining along line A“ - B“
Block 50	Machining along line B“ - C“
Block 60	Machining along line C“ - A“
Block 70	Tool axis retraction
Block 80	End of program

4.16 G45 / G46 Turning of coordinates (continued)

General data for the examples 3, 4 and 5

Program rectangle figure process

```
P452
N10  G0  F3000  Z1.5
N20  G1  G91  X0  Y0  Z-2.5
N30  X30
N40  Y20
N50  X-30
N60  Y-20
N70  Z2.5
N80  G90  M02
#
```

Parameter P8859

the parameter P8859 influences the order of rotation and shift.

P8859:0	with G92, G147, G48 and G49	first turninged, then shifted
P8859:1	with G92, G48 and G49, NP:A>0	first turninged, then shifted
P8859:2	with G92, G147, G48 and G49	first shifted, then turninged
P8859:3	with G92, G48 and G49, NP:A>0	first shifted, then turninged

4.16 G45 / G46 Turning of coordinates (continued)

Example 3 rectangle figure turn with different brackets, absolutely and incremental

```
P451
N10        T1  M16  G0
N20        G55  G49  G60  G0  F3000  X0  Y0  Z2   M28.452        {W 1}
N30   G45  P160:50  P161:10  P163:30  X0  Y0  Z2   M28.452        {W 2 turning}
N40        P165:50  P166:10  P168:30  X0  Y0  Z2   M28.452        {W 3 incremental turning}
N50   G46  G90                        X0  Y0  Z5
N60        M30
#
```

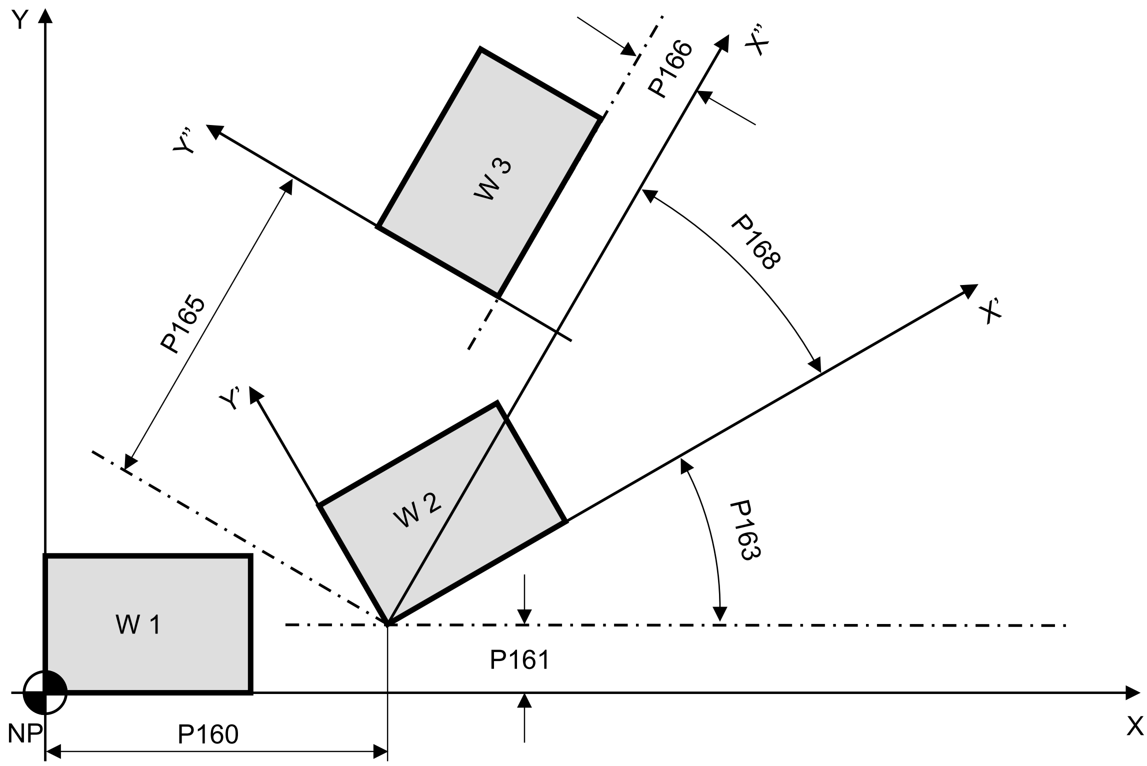


Fig. 4-24

4.16 G45 / G46 Turning of coordinates (continued)

Example 4 rectangle figure shift and turn

P8859:2 first shift by G92 and then turn by G45

```
P451
N10       T1 M16 G0
N20       G55  G147  G0  F3000 X0   Y0  Z2        M28.452    {W 1}
N30  G45  P160:50  P161:10  P163:30  X0   Y0  Z2        M28.452    {W 2 turning}
N40       P165:50  P166:10  P168:30  X0   Y0  Z2        M28.452    {W 3 incremental turning}
N50  G92                             X60 Y0                        {shift}
N60  G45  P160:50  P161:10  P163:30  X10 Y5  Z2 M28.452    {W 2' turning}
N70       P165:50  P166:10  P168:30  X0   Y0  Z2 M28.452    {W 3' incremental turning}
N80  G46  G90                        X0   Y0  Z5
N90       M30
#
```

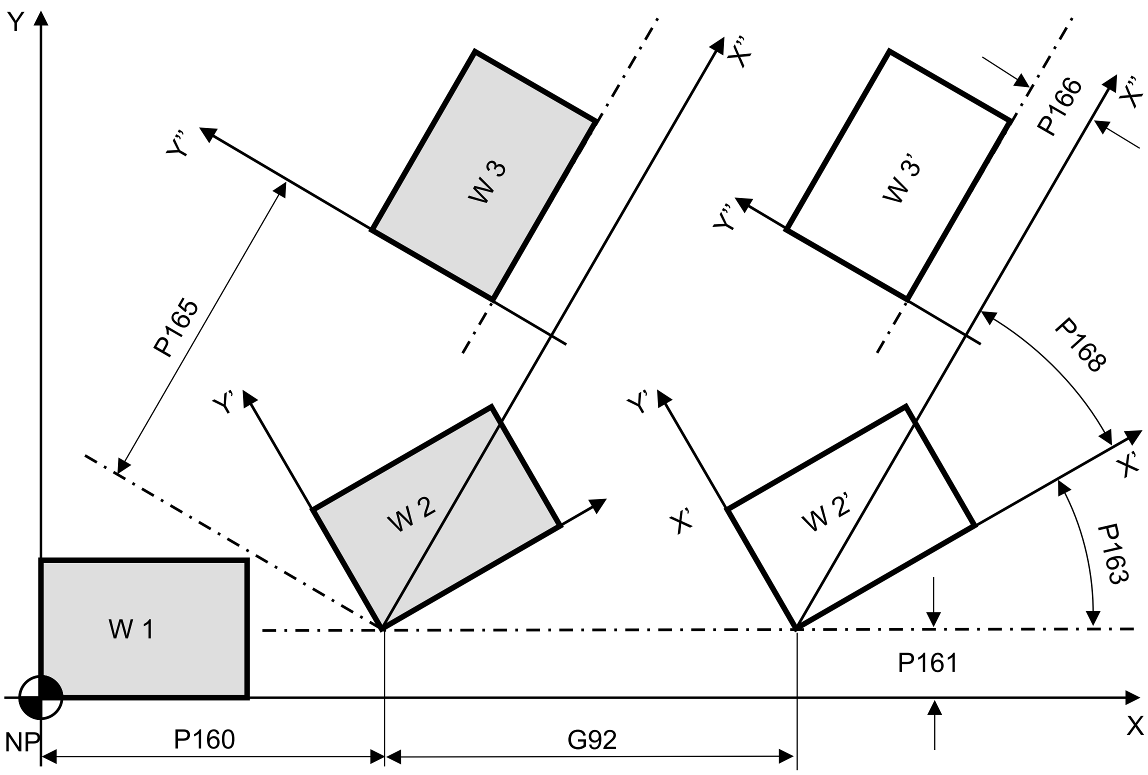


Fig. 4-25

4.16 G45 / G46 Turning of coordinates (continued)

Example 5 rectangle figure turn with different brackets and shift by G92

P8859:0 first turn by G45 and then shift by G92

```
P451
N10        T1 M16 G0
N20        G55  G147  G0  F3000 X0   Y0  Z2          M28.452    {W 1}
N30  G45   P160:50  P161:10  P163:30  X0   Y0  Z2    M28.452    {W 2 turning}
N40        P165:50  P166:10  P168:30  X0   Y0  Z2    M28.452    {W 3 incremental turning}
N58  G92                              X60  Y0                   {shift}
N60  G45   P160:50  P161:10  P163:30  X10  Y5  Z2    M28.452    {W 2'' turning}
N70        P165:50  P166:10  P168:30  X0   Y0  Z2    M28.452    {W 3'' incremental turning}
N80  G46   G90                        X0   Y0  Z5
N90        M30
#
```

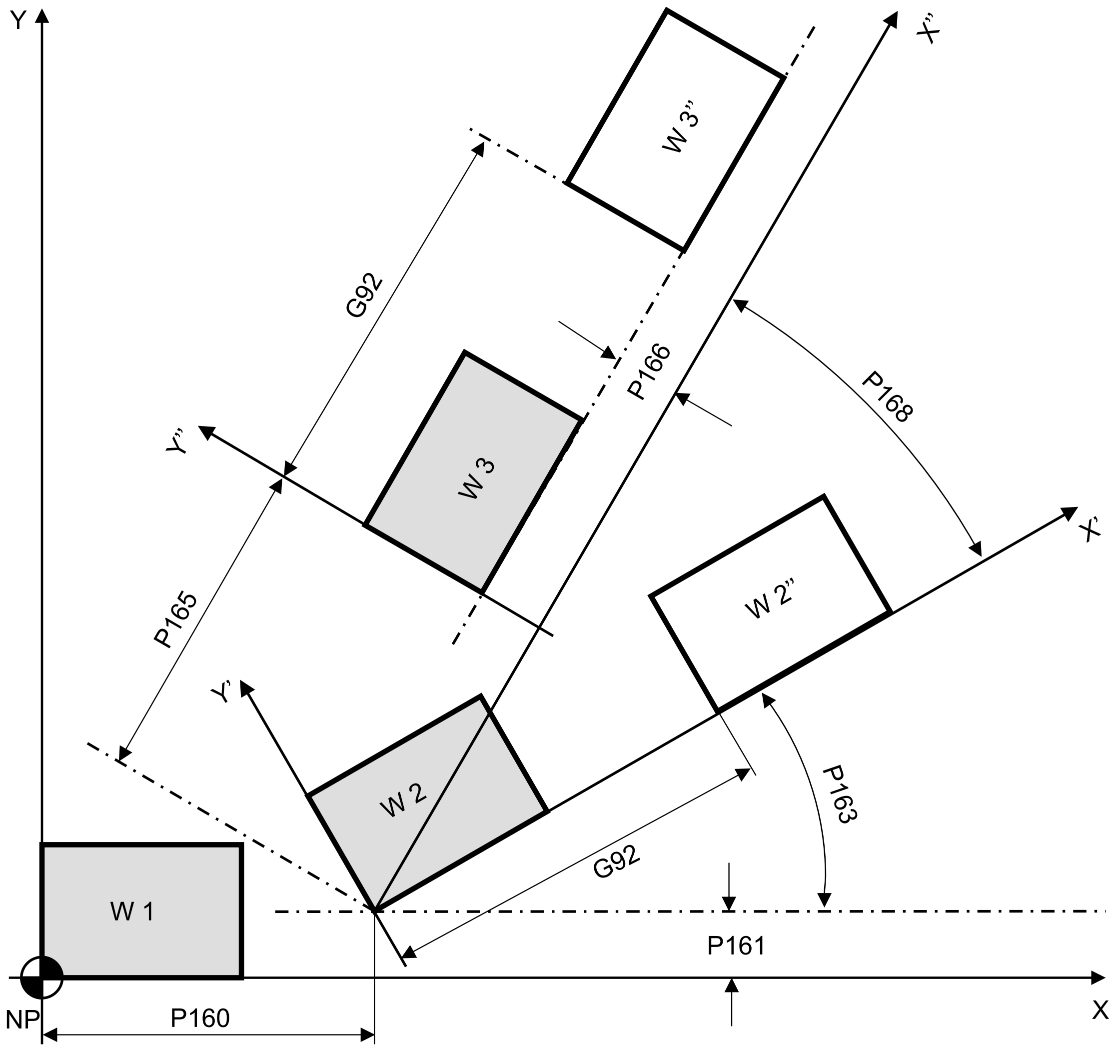


Fig. 4-26

4.17 G47 / G48 / G49 / 147 Robot transformation

It is advantageous for different applications, if the user can select the movements in different mode of coordinates when installing the machine (e.g. machine tool or handling system). With the BWO-ROBOT-system three modes of coordinates can be chosen.

Machine coordinates G47

The general and most frequent case is, that the movement is refered to a machine coordinate system, Thereby the coordinate axes and an origin of coordinates are determined. The cartesian coordinates are used.

Transformation off by G147

G147 turns out the transformation. The transformation is preset at startup.

The zero points G54 to G59, the tool length and tool length offsets are active.

The coordinate type zero (0) or deleted (--) corresponds to G147 (in manual mode)

Tool coordinates G48

If the movement is refered to the tool or a grip, infeed movements can be executed very simple for machining - or assembling tasks. In this case, the coordinates are oriented at the tool peak.

Workpiece coordinates G49

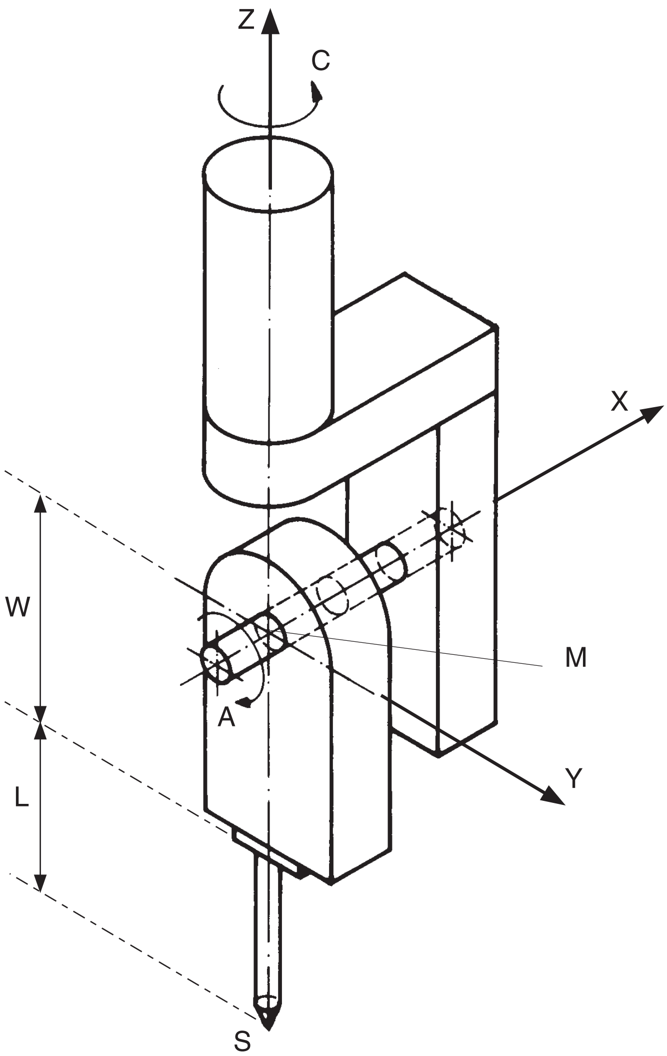
Programming is facilitated for many machining functions, if the movement is refered to a coordinate system oriented at the workpiece.

Switching between the above named modes of coordinates (G47 / G48 / G49) is an essential help for the user at programming with Teach-in, because e.g. for establishing a NC-program the contours of a workpiece only have to be touched and stored (teached). If equal workpieces are processed in different positions, the same NC program can be used through a single transformation of the coordinates.

4.17 G47 / G48 / G49 / 147 Robot transformation (continued)

The coordinate system demands the following axis configuration:

The A - axis turns around an axis parallel to the X - axis.
The C - axis turns around an axis parallel to the Z - axis.
The spindle axis stands vertically on the A - axis.
The intersection of all three rotation axes is the center point M.
The positive rotating direction of C is seen from positive Z-direction in counter-clockwise direction.
The positive rotating direction of A is seen from positive X - direction in counter-clockwise direction.
The reference point offsets are to be determined, so that in position A = C = 0 the tool system is axis parrallel to the basic coordinate system.



A. C	Angle
M	Center point
W	Tool loader dimension
L	Tool length
S	Tool peak

Figure 4-27 Axis configuration

4.17 G47 / G48 / G49 / 147 Robot transformation (continued)

Machine coordinate system (G47)

At the machine coordinate system the coordinates refer to the machine origin.
Thereby the machine axes are defined in cartesian coordinates. The axis movements orientate to the axis coordinates.

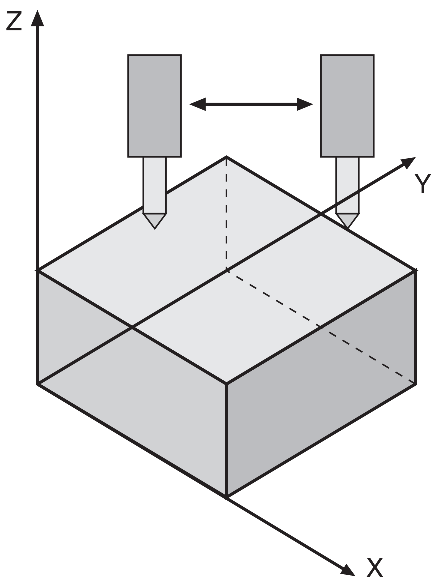


Figure 4-28 Axis movements in the machine coordinate system

In G47, the zero points G54 to G59, the tool length and tool length correction not be charged.

4.17 G47 / G48 / G49 / 147 Robot transformation (continued)

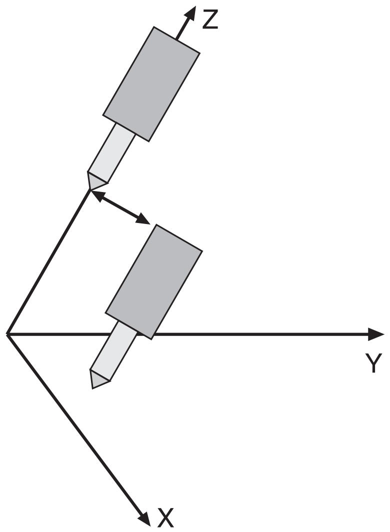
Tool cordinate system (G48)

At the tool coordinate system the coordinates refer to the tool. The cartesian coordinate system is only used with a firm unit of length (e.g. mm).

If the tool system is arbitrarily shifted and rotated through the NC program, the system must fulfill the following conditions.

The coordinate source is the point, around which the tool is turned, if its orientation is changed. This point is named tool peak.

At rotating tools the Z-axis is the rotation-axis and indicates from the tool peak into the direction of the chuck.



- The Robot displacement parameters P11802 to P11807,
- the zero points G54 to G59 and
- the tool length and tool length correction

are active.

Figure 4-29 Axes movements in too coordinate system

The speed of the tool peak is interpreted as actual working feed.

When activating the tool dimensions the control shifts the tool coordinate system in the Z-direction for an amount corresponding to the tool length.

If the X -, Y - or Z- traverse direction is selected, the X -, Y - or Z - axis traverse according to the position of the rotating angle of the C-axis and the position of the tumbler angle of the A-axis, so that the tool stands always vertically on the X- Y- plane.

4.17 G47 / G48 / G49 / 147 Robot transformation (continued)

Workpiece coordinate system (G49)

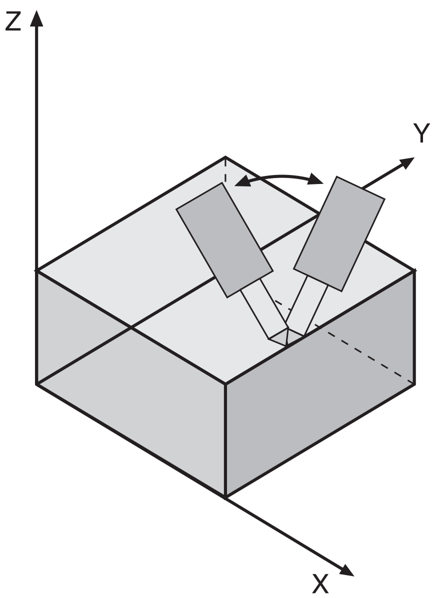
At the workpiece coordinate system the coordinates refer to the workpiece.
Only the cartesian coordinate system with a firm unit of length (e.g. mm) is used. The advantage of the workpiece system is, that it can be shifted and turned arbitrarily with the NC program.

For each point a coordinate vector can be allocated in reference to the workpiece system, for example:

- If for the linear axis the X -, Y - or Z- traverse direction is selected, the X -, Y - or Z - movement is parallel to the axes coordinates (independently of the position of the rotating angle of the C-axis and the tumbler angle of the A-axis).

- If only axis C is traversed, X and Y are relocated so that the tool peak is always at the same point. The X - and Y - axis describe a circle around this point. The orbit proceeds vertically under the center point.

- If axis A is traversed, additionally the Z - axis is moved in plus - or minus direction according to the tumbler angle position of A.



- The Robot displacement parameters P11802 to P11807,

- the zero points G54 to G59 and

- the tool length and tool length correction

are active.

Figure 4-30 Axes movements in workpiece coordinate system

Appearing coordinate transformation when switching over between G47, G48, G49 determines the position and orientation of the tool relative to the workpiece reversable definitely.

4.17 G47 / G48 / G49 / 147 Robot transformation (continued)

Offset functions

Sometimes it is necessary, that at the construction of the machine the point of balance does not correspond to the center point M. In this case, a function for compensating this deviation is necessary.

The system offers three offset functions. By using this function, the mechanical offset is corrected automatically.

Offset in X - direction

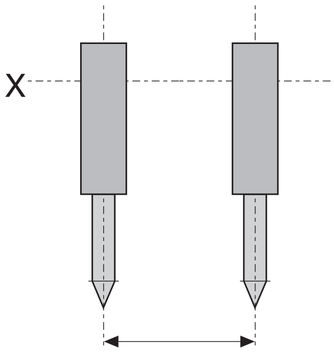


Figure 4-31 Offset in X - direction

Paramter P11802 is offset value [mm].

Offset in Y - direction

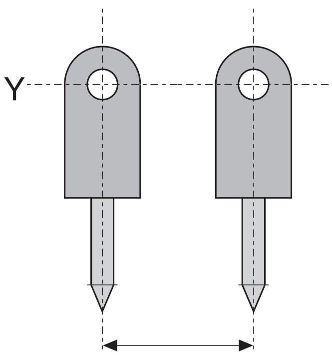


Figure 4-32 Offset in Y - direction

Paramter P11803 is offset value [mm].

4.17 G47 / G48 / G49 / 147 Robot transformation (continued)

Offset in Z - direction

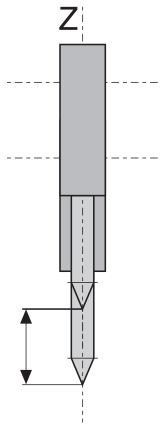


Figure 4-33 Offset in Z - direction

Paramter P11804 is offset value [mm].

4.17 G47 / G48 / G49 / 147 Robot transformation (continued)

Tool carrier with oblique angled axes

The configuration of the tool carrier with oblique angled axes (in zero position) is shown in figure 4.24. The A - axis turns around an axis, which does not stand vertically on the rotation axis Z. That means, that the tilt is not parallel to the X - axis. The paramter P11806 is the tilt angle.

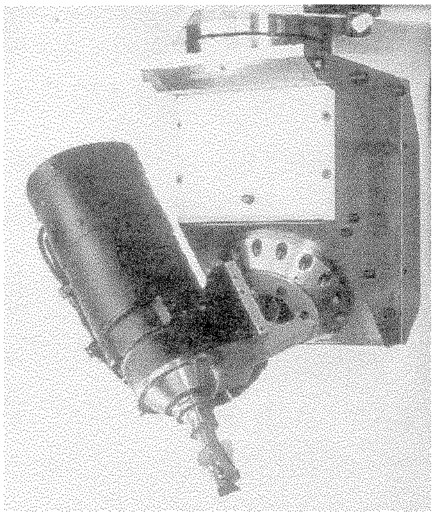


Figure 4-34 Tool carrier with oblique angled axes

Value areas and handling

The parameter values used are limited in the following areas.

For the parameters P11802, P11804 and P11806 positive or negative values are possible.
The amount of P11806 must be smaller than 180 degrees.
If P11806 is = 0, the presetting is 90 degrees, namely ξ = 90[degree]
If ξ = 90[degree], $\alpha - \alpha 12 = 0$, $\beta - \beta 12 = 0$ and $\theta = 0$

The amount (β) is dependent on the areas of ξ.

4.17 G47 / G48 / G49 / 147 Robot transformation (continued)

P8759 Coordinate mode in Manual

0	Transformation off
15	Polar transformation off
16	Polar transformation on
47	Robot transformaton off
48	Tool coordinate system
49	Workpiece coordinate system

P11800 Robot linear axes (physical axis number)
e.g. the first 3 axes (X, Y, Z)
Byte 3,2,1 03 02 01 Hex
Byte 4 reverse (Bit 3, 2, 1)

P11801 Robot rotation axes (physical axis number)
e.g. the 4th and 5th axis (A, C)
Byte 3, 2, 1 00 05 04 Hex
Byte 4 reverse (Bit 3, 2, 1)

P11802	Offset X, rotation axis - tool axis	[mm]
P11803	Offset Y, rotation axis - tilt	[mm]
P11804	Offset Z, length of tool carrier	[mm]
P11805	Angle between tilt - tool axis	[degree]
P11806	Angle between rotation axis - tilt	[degree]
P11807	Angle where the tool shows downwards	[degree]

4.17 G47 / G48 / G49 / 147 Robot transformation (continued)

Zero points

The Robot system is given through a block of coordinates, which indicates its position in reference to the machine system. The zero point system G54 to G59 is analogously valid in the Robot system. The zero point indicates the position in reference to the machine system, i.e. the position of the tool is in reference to the workpiece. It can be filed in a zero point memory and is activated by calling up G54 to G59.

The functions G54 to G59 overwrite mutually. Besides they overwrite a shift effective through G92.

Zero point shift

The Robot system can be shifted and turned again in the NC program. The position of the shifted system relative to the old one is given through a block of coordinates. Calling up results through the zero point shift G92. This position of G92 is the reference to the actual zero point coordinates. In manual mode this zero point shift is not disposable.

The position of the tool in the machine system is given through 3 blocks of coordinates

- Zero point coordinates

- Zero point shift

- Coordinates of the position

4.17 G47 / G48 / G49 / 147 Robot transformation (continued)

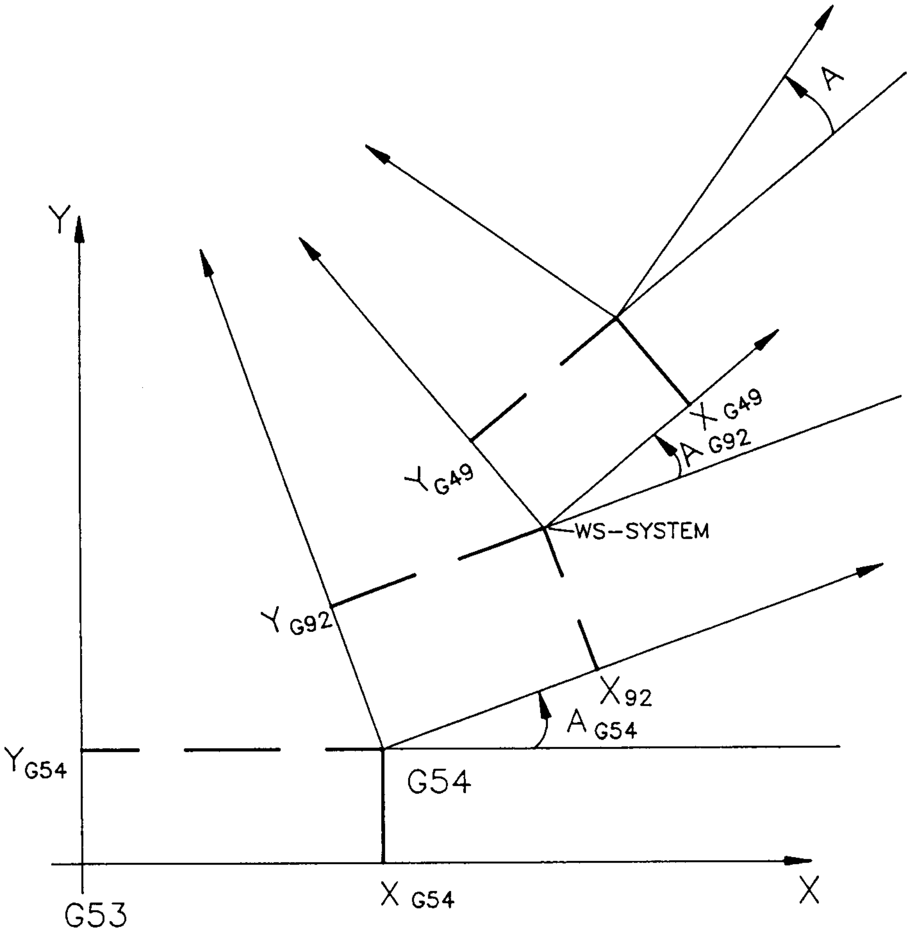


Figure 4-35 Zero point shift

4.17 G47 / G48 / G49 / 147 Robot transformation (continued)

Denomination of the 3 axes, e.g. tilt B, linear axes X and Z.

With 3-axes-tool machines, G49 is controlling the axes so that when moving the tilt B, the axes X and Z are always so that the tool peak is always at the same position.

Configuration and definition for the 3-axes-coordinate system

The 3-axes-coordinate system demands the following axis layout:

- X and Z axis are the basic coordinates.

- The B axis is rotating around an axis that is positioned vertically to the plane XZ.

- The positive rotating direction of B is seen from positive X direction counter-clockwise.

- The reference measures are termined so that the tool system is axis-parallell to the basic coordinates in position B=0.

- The offset between the tool peak and the rotating point is determined by the parameters P11802 (offset radius) and P11803 (offset angle).

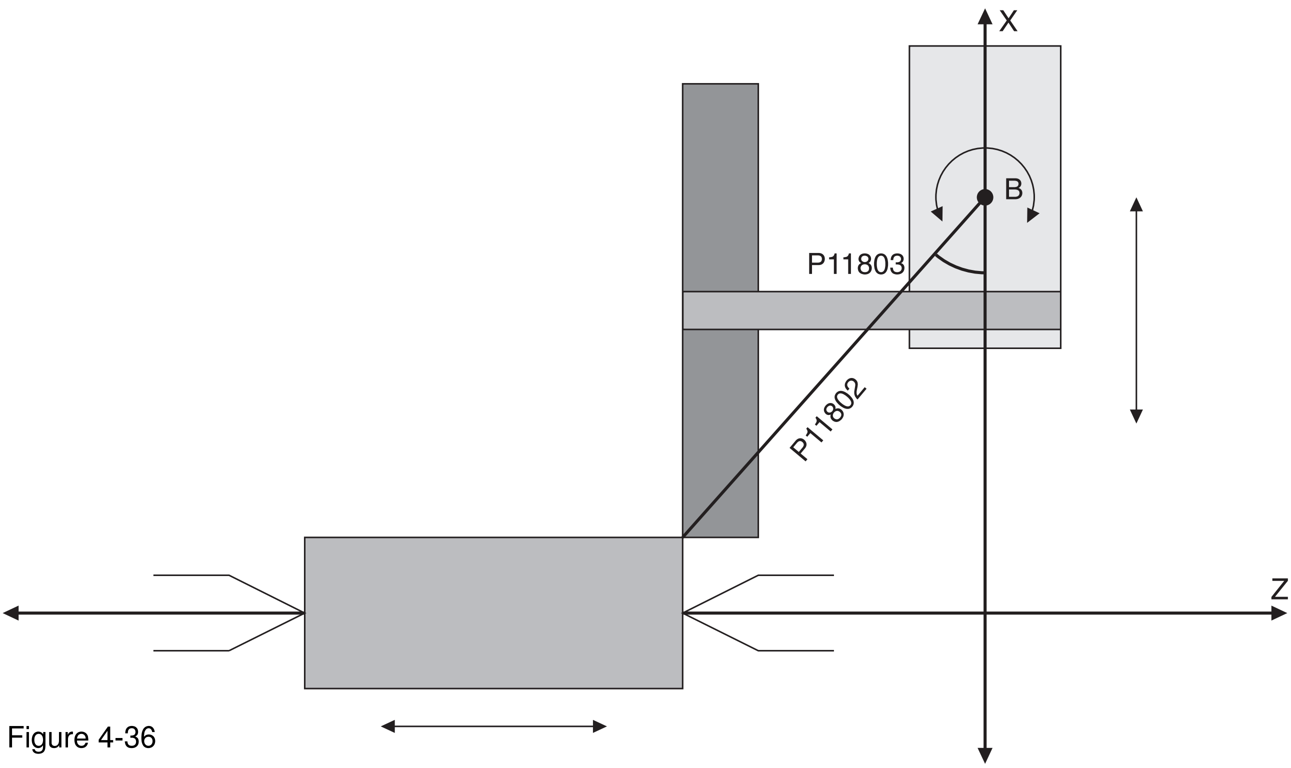


Figure 4-36

4.17 G47 / G48 / G49 / 147 Robot transformation (continued)

Used parameters

P11800 Linear axes(physical axis number)

e.g. the first 2 axes (X,Z)
Byte 3, 2, 1 00 02 01 Hex
Byte 4 reverse (Bit 2, 1)

P11801 Rotation axes (physical axis number)

e.g. the 3rd axis (B)
Byte 3, 2, 1 00 00 03 Hex
Byte 4 reverse (Bit 1)

P11802 Offset radius, tilt - tool peak [mm]

P11803 Offset angle, tilt - tool peak [degree]

4.18 G50 / G51 / G52 Spline interpolation

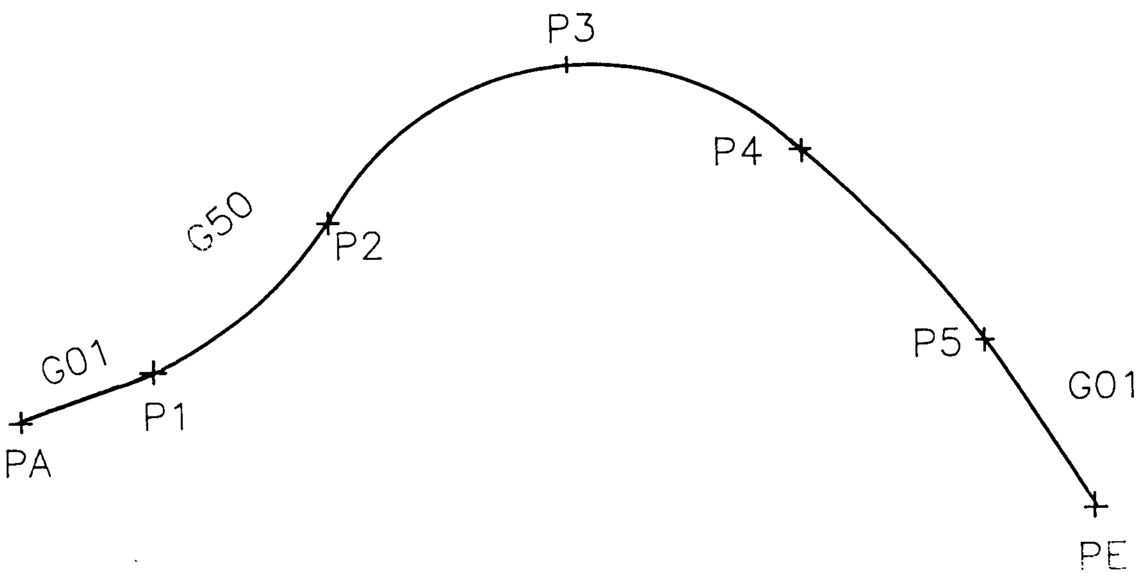
G50 Spline interpolation
G51 Spline interpolation with feed adaptation at the contour
G52 Polynom interpolation

G50 / G51 Spline interpolation

A sequence of points can be joined tangentially with a smooth curve trace with the spline interpolation. The points can be entered both as NC program as well as in the manual operation mode through „Teaching“ with command=target.

The spline curve is started through a traverse, which is run with G00 or G01. The start can also be with G02 or G03, if the starting position is in a main plane (G17, G18, G19).

The spline interpolation is activated in the subsequent block through G50 / G51. There results a tangential transition, whereby the starting tangent of the spline curve is determined through the starting segment.



PA Starting point
PE End point
P1...P5 Spline points

Figure 4-37

4.18 G50 / G51 / G52 Spline interpolation (continued)

In the corresponding way the spline curve is finished through programming the @@Ausleitungs-segment with G00, G01, G02 or G03.
The traverse conditions G00, G01, G02, G03 and G50 overwrite themselves mutually.

If the starting segment or the finishing segment are not programmed, the NC program stands still, because the spline interpolation can not be started or finished duly. In this case, the key 'Manaul' has to be pressed and the error has to be cleared.

G52 Polynom interpolation

The path a is described with the formula $\vec{P} = \vec{V}_3\ t^3 + \vec{V}_2\ t^2 + \vec{V}_1\ t^1 + \vec{V}_0$,

whereby the variable t can have a value between 0 and 1.

The vectors V_3, V_2, V_1 are programmed in components in the following form:

X : [V_{3X}, V_{2X}, V_{1X}]   Y : [V_{3Y}, V_{2Y}, V_{1Y}]   Z : [V_{3Z}, V_{2Z}, V_{1Z}]

V_0 is not programmed, since it is the location of the machine at the beginning of the block.

Starting point $\vec{P}_S = \vec{V}_0$

Final point $\vec{P}_E = \vec{V}_3 + \vec{V}_2 + \vec{V}_1 + \vec{V}_0$

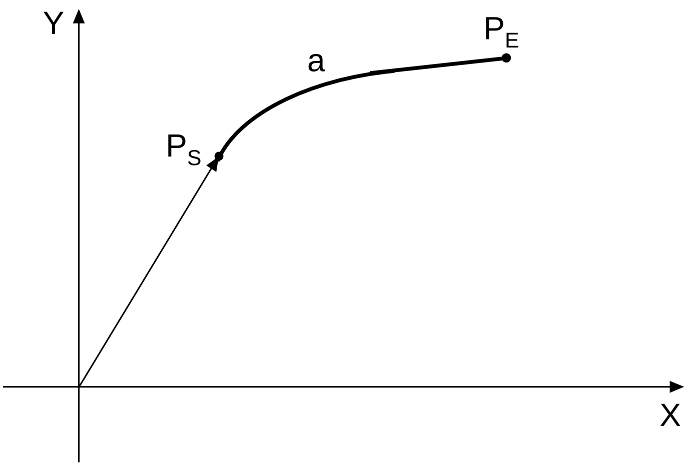
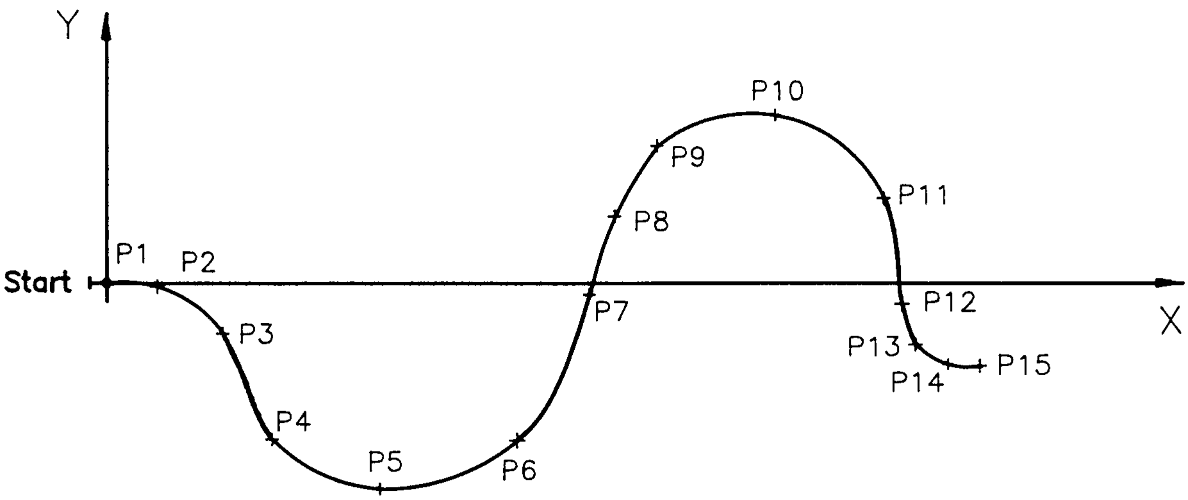


Figure 4-38

4.18 G50 / G51 / G52 Programming examples

Spline contour

N5	G54	G49	G00	F2000	X - 8	Y0	Z5		Starting point
N10	G01				X0	Y0	Z0	point 1	
N20	G50				X6,8	Y-0,5		point 2	Spline on
N30					X15,5	Y-6,7		point 3	
N40					X22	Y-20,6		point 4	
N50					X36,3	Y-27,1		point 5	
N60					X54,3	Y-20,7		point 6	
N70					X63,8	Y-1,5		point 7	
N80					X67,124	Y8,749		point 8	
N90					X72,7	Y18		point 9	
N100					X88,1	Y22		point 10	
N110					X102,5	Y11,1		point 11	
N120					X104,9	Y-2,8		point 12	
N130					X106,7	Y-8,1		point 13	
N140					X111	Y-10,7		point 14	
N150	G01				X115,2	Y-10,9		point 15	Spline off
N160	M30						Z5		



Point 1 Zero point
Point 2 Switching on spline interpolation
Points 3 - 14 Spline points
Point 15 Switching off of spline interpolation

Figure 4-39

4.19 G53 Machine zero point

With the input of G53 with G54 to G59 as well as with G92 programmed zero point shifts become ineffective. The program refers then to the machine zero point.

The function is effective modal and can be overwritten by G54 to G59.
With the program start G53 is automatically adjusted.

G153 switches the zero shift off G53 as well as G54 to G59.

4.20 G 54 to G59 Zero points

With G54 to G59 points of zero can be called within the program.

The functions are effective modal and overwrite themselves opposite-acting. Additionally one is reset by G92 programmed shift.

The points of zero are called in the program with G54 to G59 in the null point memory. The size of the zero shift is determined by the stored values in the null point table.

Example: programming zero points

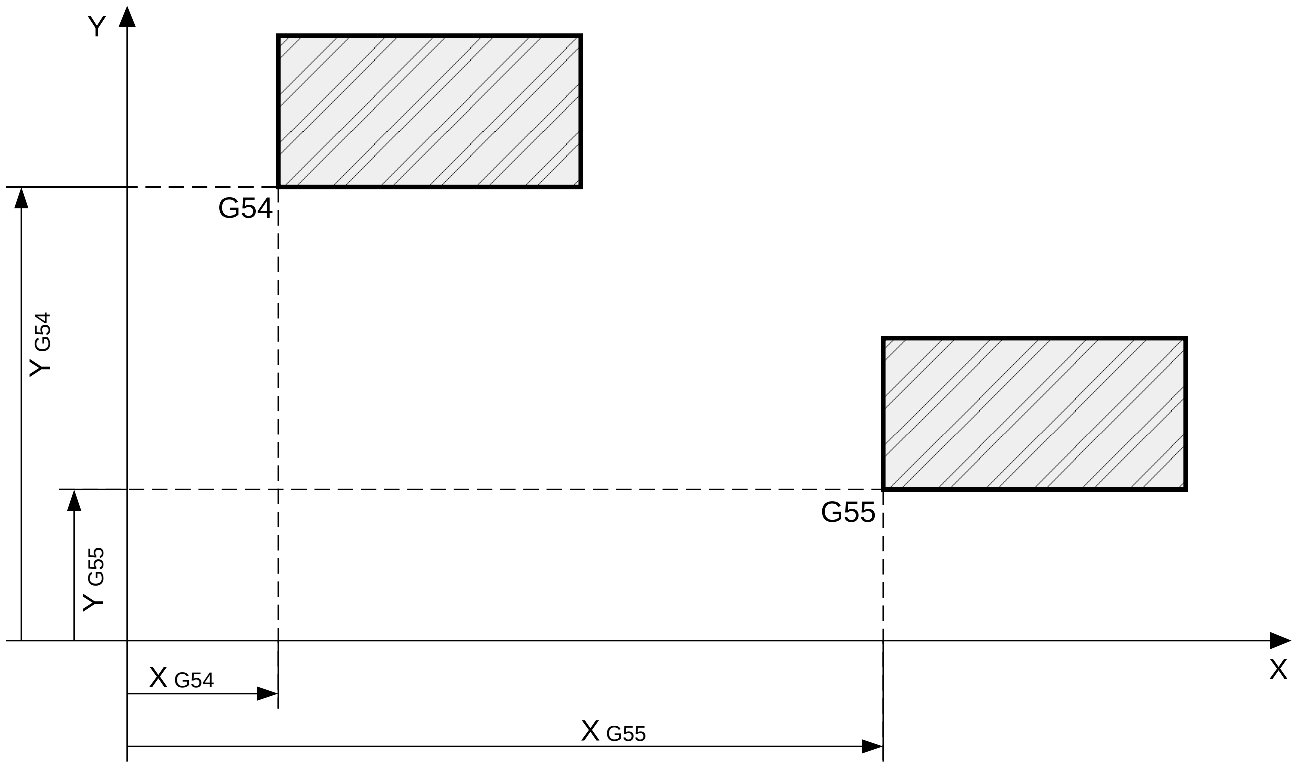


Figure 4-40

4.21 G60 / G61 / G62 Mirror imaging of coordinates

The mirror imaging function inverts the sign of programmed coordinates.
The G61 function inverts the sign of the 1st main axis program values.
The G62 function inverts the sign of the 2nd main axis program values.

Set plane	G61	G62
G17 (XY)	X becomes -X	Y becomes -Y
G18 (ZX)	Z becomes -Z	X becomes -X
G19 (YZ)	Y becomes -Y	Z becomes -Z

G60 cancels the mirror imaging functions G61 and G62 out. G60, G61 and G62 act modally.

Geometric definition

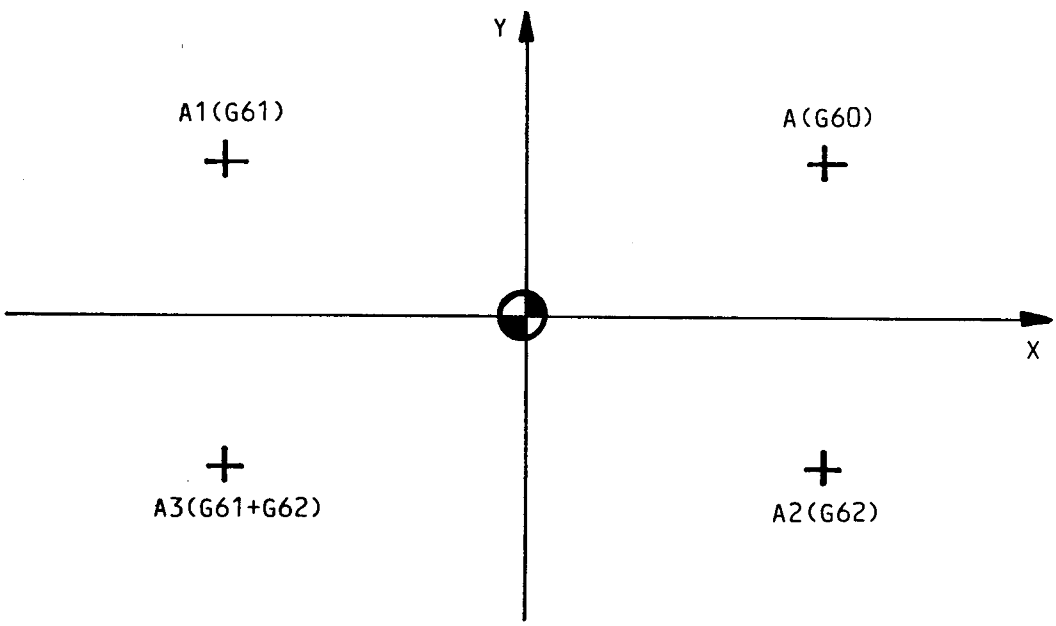


Figure 4-33

The programmed coordinates for point A are Xa and Ya.

When G61 is switched on point A becomes point A1 with -Xa and Ya.
When G62 is switched on point A becomes point A2 with Xa and -Ya.
When G61 and G62 are switched on point A becomes point A3 with -Xa and -Ya.

4.21 G60 / G61 / G62 Mirror imaging of coordinates (continued)

Circle centre point coordinates I, J, K

The circle centre point coordinates are mirror imaged according to the plane and functions set.

Zero points G53 to G59

Zero points G53 to G59 are not mirror-imaged.

Shift G92

The values of shift G92 are mirror imaged according to the appropriate plane and functions set.

Circular interpolation G02 and G03

When circular interpolation is switched on and G61 and G62 are active, G02 becomes G03 and G03 becomes G02. The direction of rotation remains unchanged when G61 and G62 are switched on.

Tool correction offset G41 and G42

When the tool correction offset is switched on and G61 and G62 are active, G41 becomes G42 and G42 becomes G41. The selected correction offset remains unchanged when G61 and G62 are switched on.

Effect of G66

G66 effects the block by block switching off of the mirror-imaging functions G61 and G62.

Incremental dimension programming G91

Using G91, the target point in the system that has not been mirror-imaged is firstly determined and is then mirror-imaged according to the appropriate plane and mirror-imaging functions set.

4.22 G63 / G64 „Feed rate 100%“

G63 Override 100% switch on

With G63 the Override value is set firmly to 100%, i.e. the override is not active any longer. NC program processing runs with the programmed feed.

The function is modal effectively and can by G64 again be switched off.

G64 Override 100% switch off

G64 switches G63 out

The function is modal effectively and can by G63 be overwritten.

With the program start G64 is preset.

4.23 G66 Switching off all correction offsets

When programming G66, the machine zero point is taken as the reference point. All dimension inputs then refer to the machine zero point.

The G66 function is effective only in the block in which it was written.

In this particular block the zero points set by G54 to G59 and shift G92 are inoperative, as are any tool length and tool radius correction offsets.
They remain stored, however, and are effective in the next block again.

4.24 G90 Absolute dimension input

When G90 is being programmed absolute dimension input is chosen, i.e. all dimensions are referred to the program zero point (programmable using G54 to G59 and G92).

G90 operates modally and can be overwritten by G91. G90 is automatically set when the control system is switched on.

Example: Absolute dimension input

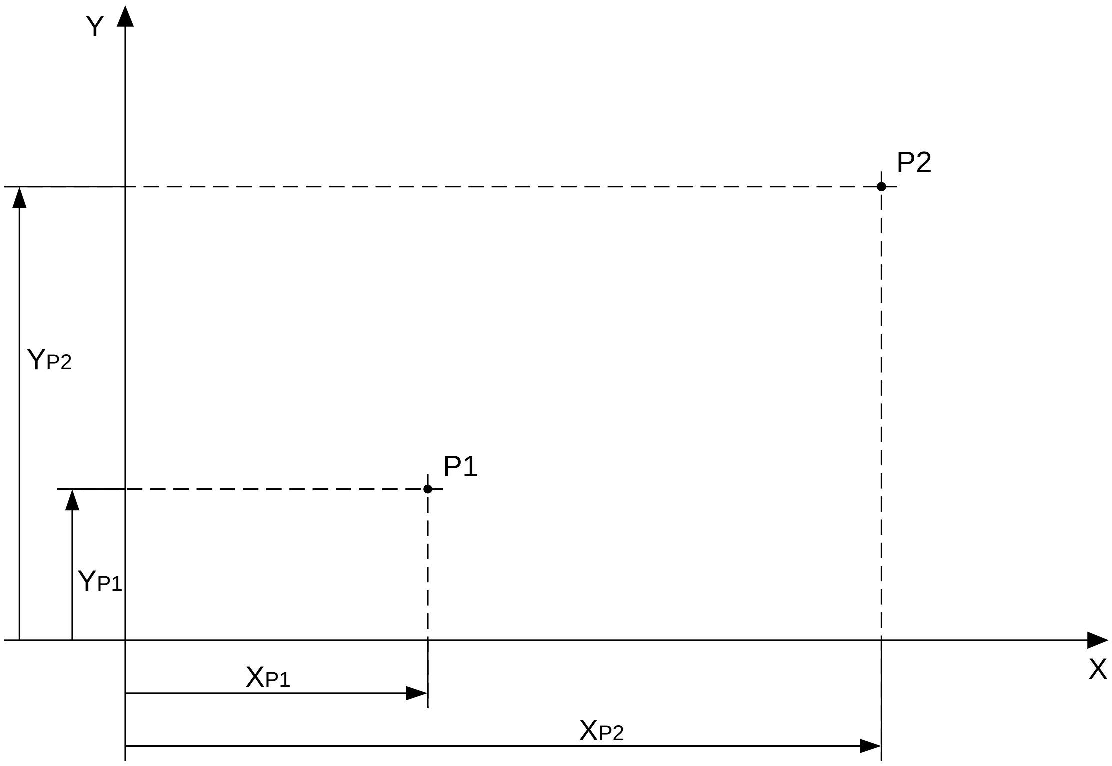


Figure 4-34

4.25 G91 Incremental dimension input

With the programming of G91 the chain measure input is selected. It always refers to the location of the axes started last.

G91 is modal effectively and can by G90 be overwritten.

Example: Incremental dimension input

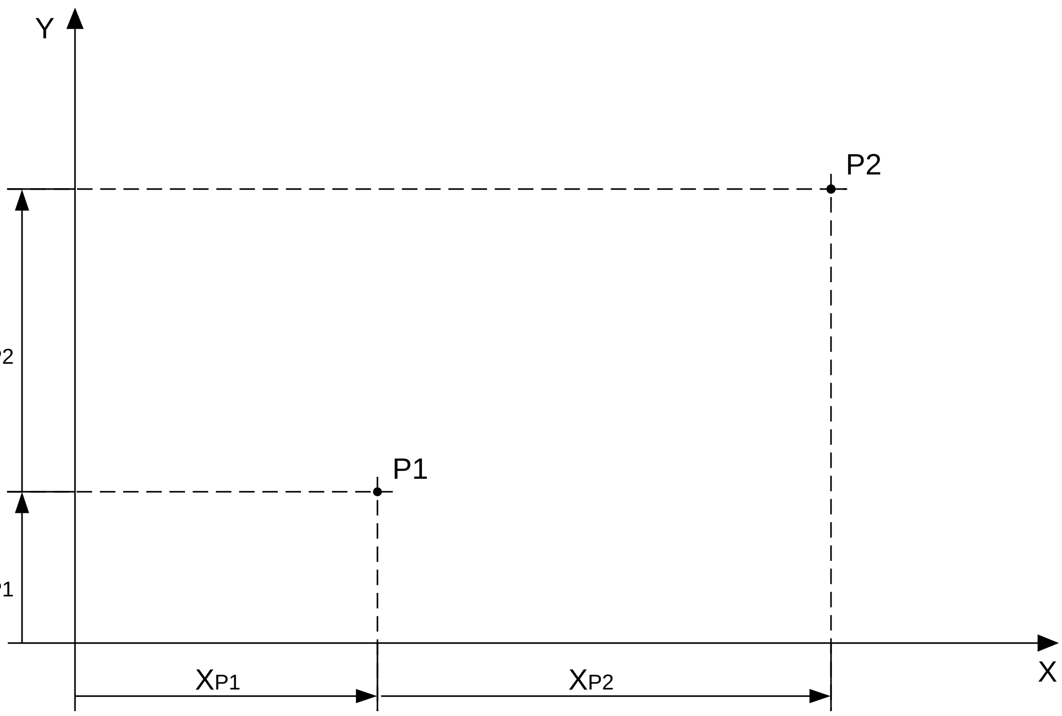


Figure 4-35

4.26 G92 Zero point shift

Zero point shifs programmed by G92 are a fixed part of the NC program and therefore independent of the clamping of the workpiece. In contrast, zero points are programmed outside the NC program by G54 to G59 and activated in the NC program.

Traverse information in the block with G92 determines the zero point shift, but no traverse is activated.

Input: G92 X... Y... Z...

The zero point is shifted by the programmed amounts.

The zero point shift programmed by G92 operates modally and is always referred absolutely to the machine zero point (G53) or to a set zero point (G54 to G59).
G92 can be cleared by G53 to G59 or overwritten by a new G92.

Example: Zero point shift

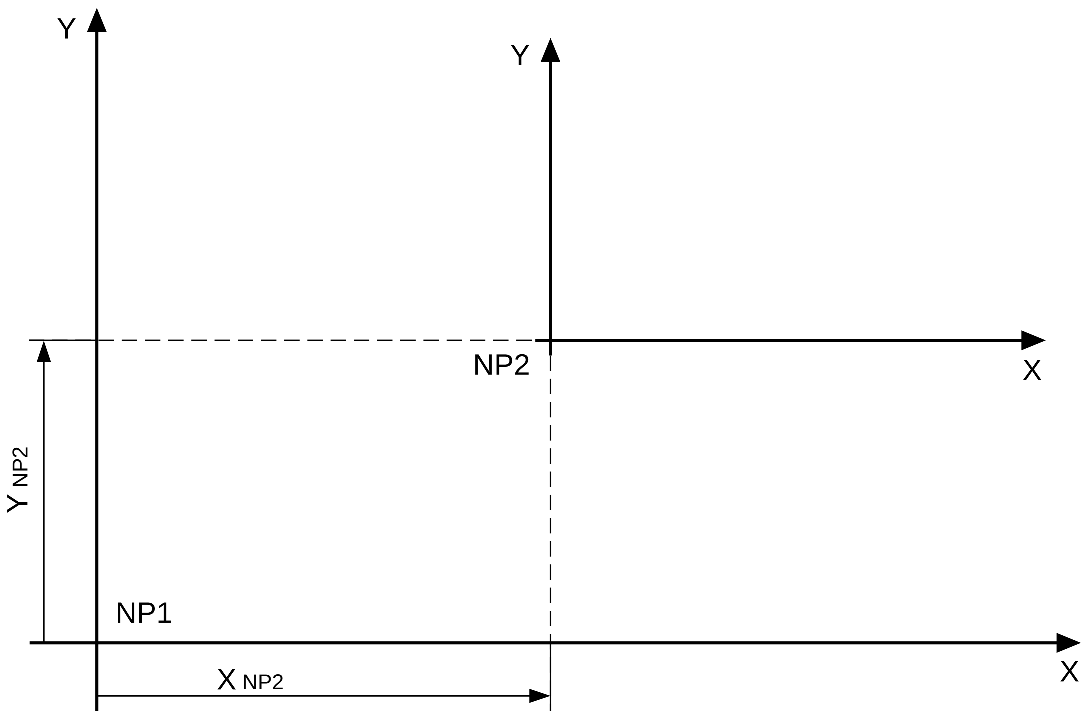


Figure 4-36

4.27 G94 / G95 Feed modification

The functions determine the feed modifications:

G94 Feed in mm/min

G95 Feed in mm/r (revolution)

4.28 G96 / G97 Number of revolution modification

The functions define the speed modifications:

G96 **constant cutting velocity in mm/min**

G97 **speed in U/min**

In order to activate G96 / G97, the following parameters must be set:

P11640 spindle affects axis (physical axis)
(example round axis C corresponds the 6. Axis Input P11640:6)

P11641 datum axis for G96 (physical axis)
(example procedure axis Y corresponds the 2. Axis Input P11641:2)
If P11641 = 0 applies, then the reference position in P11642.

P11642 reference position for G96 [mm]
(e.g.: disk diameters)

P11643 reference factor for G96 [mm/min]
0 or — mm/min
1000 m/min

P11644 max. speed for G96

P11645 max. speed for G97

P11646 reference factor for G97 [U/min]
0 or — U/min
1 degree/min

P11647 speed definition G96 / G97
Presetting 0 = G97
96 = G96

4.28 G96 / G97 Number of revolution modification (continuation)

In the display selection menu the inputs under „ spindle parameters „ can be input.
Spindle axis: 6
Datum axis: 2

Example NC Progr.:
N10 G0 Y:50
N20 G96 G1 F50 S100 Y:0
The speed of the round axis increases the more,
the more near processing toward the position Y:0 comes.

N10 G0 Y:0
N20 G96 G1 F50 S100 Y:50
The speed of the round axis decreases itself the more,
the more near processing toward the position Y:50 comes.

4.29 G170 / G171 inch system

G171 enables metric processing in the NC program with switched on inch system.

G170 switches G171 off

5. Cycles

5.1 G69 measuring cycles

General

The measuring cycle is a NC program with the term Z69. This cycle must like all other cycles into the NC memory be loaded. The measuring cycle is called with G69.

Measuring cycles use the parameters P140 to P160 for programming. These parameters contain two types of information:

- Data, e.g.. P141 safety margin

- Parameter numbers, which indicate, where the data are stored,
  e.g.. P149: 500 (P149 is a pointer on another parameter).

Measuring cycles use the parameters P200 to P299 for internal calculations.
Parts it safely that only free parameters are used for the addressing of data fields.
Measuring cycles activate G40 (tool radius path correction out).

5.1 G69 measuring cycles (continued)

Measuring preparation

- Connecting it the sensor with the measuring input on the AAZ module (15pol. HD Sub Socket). If the measuring axes are distributed on several modules, all interconnect you Measuring inputs together (MT+ / MT-).

- Checking it the measuring logic for each measuring axis (P12045: $xxxx1010...).

- Checking it the tool data of the used sensor. G69 requires tool length and tool radius of the sensor.

5.1 G69 measuring cycles (continued)

Compilation of the cycle parameters

Parameter		meaning
P140		cycle selection
P141	[mm]	safety margin
P143	[mm/min]	measuring speed
P144	[n]	number of measurements at the same point (> 0)
P147		pointer on the measuring tolerance data
(P147)+4	[mm]	confidence range
P148		measuring axis in plane system (1... 3)
P149		pointer on the data of point of trigger
(P149)+0	[mm]	XN, point of trigger in negative direction 1. axis
(P149)+1	[mm]	XP, point of trigger in positive direction 1. axis
(P149)+2	[mm]	YN, point of trigger in negative direction 2. axis
(P149)+3	[mm]	YP, point of trigger in positive direction 2. axis
(P149)+4	[mm]	ZN, point of trigger in negative direction 3. axis
(P149)+5	[mm]	ZP, point of trigger in positive direction 3. axis
P150		pointer on the calibration drilling data
(P150)+0	[mm]	position of the calibration drilling focal point 1. axis (I)
(P150)+1	[mm]	position of the calibration drilling focal point 2. axis (J)
(P150)+2	[mm]	position of the calibration drilling focal point 3. axis (K)
(P150)+3	[mm]	calibration drilling diameter
P152	[mm]	expected drilling diameter
P154		pointer on the measuring data
(P154)+0	[mm]	measuring position 1. axis
(P154)+1	[mm]	measuring position 2. axis
(P154)+2	[mm]	measuring position 3. axis

5.1 G69 measuring cycles (continued)

Cycle	Parameter	Calibrate of the sensor in drilling	Calibrate of the sensor at area	Determining of centre a drilling	Determining of positions at area
Cycle selection	P140	1	2	60	40
Safety margin	P141	x	x	x	x
Measuring speed	P143	x	x	x	x
Number of measurem.	P144	x	x	x	x
Measuring tolerance	P147	x	x	x	x
Confidence range	(P147)+4	x	x	x	x
Measuring axis in plane	P148				x
Data of point of trigger	P149	x	x	x	x
Point of trigger XN	(P149)+0	o		x	x
Point of trigger XP	(P149)+1	o		x	x
Point of trigger YN	(P149)+2	o		x	x
Point of trigger YP	(P149)+3	o		x	x
Point of trigger ZN	(P149)+4				
Point of trigger ZP	(P149)+5		o		x
Calibration drilling data	P150	x			
Centre position 1.axis I	(P150)+0	x**	x*		
Centre position 1.axis J	(P150)+1	x**	x*		
Centre position 1.axis K	(P150)+2	x**	x*		
Drilling diameter	(P150)+3	x			
Expected diameter	P152			x	
Output data of the pos.	P154			x	x
Measuring pos. 1. axis	(P154)+0			o**	o*
Measuring pos. 2. axis	(P154)+1			o**	o*
Measuring pos. 3. axis	(P154)+2			o**	o*

x: Parameter must be programmed
o: cycle edition

* : one of these
**: two of these

5.1 G69 measuring cycles (continued)

Message	Meaning
8130	uppersize
8131	undersize
8132	permissible measure difference exceeded
8133	confidence range exceeded
8134	reference drilling smaller than sensor
8135	boring axis is measuring axis
8136	sensor cannot be initiated
8137	P140 incorrectly programs
8138	P141 incorrectly programs
8139	P142 incorrectly programs
8140	P143 incorrectly programs
8141	P144 incorrectly programs
8142	P145/147/149/150 incorrectly programs
8143	empirical value incorrectly programs
8144	P146 incorrectly programs
8145	measuring tolerance 1 incorrectly programs
8146	measuring tolerance 2 incorrectly programs
8147	measuring tolerance 3 incorrectly programs
8148	measuring tolerance 4 incorrectly programs
8149	measuring tolerance 5 incorrectly programs
8150	P148 incorrectly programs
8151	point of trigger XN incorrectly programs
8152	point of trigger XP incorrectly programs
8153	point of trigger YN incorrectly programs
8154	point of trigger YP incorrectly programs
8155	point of trigger ZN incorrectly programs
8156	point of trigger ZP incorrectly programs

5.1 G69 measuring cycles (continued)

Message	Meaning
8157	point of drilling I incorrectly programs
8158	point of drilling J incorrectly programs
8159	point of drilling K incorrectly programs
8160	point of drilling R incorrectly programs
8161	P151 incorrectly programs
8162	P152 incorrectly programs
8163	P153 incorrectly programs
8164	tool not called
8165	G53 not actively
8166	G54 to G59 not actively
8167	sensor actively outside of the safety area
8168	P154 incorrectly programs

5.1 G69 measuring cycles (continued)

Calibration of the sensor in drilling G69 P140:1

The cycle requires a calibration drilling and a sensor. Measuring is executed with the two axes of the selected interpolation plane.

The sensor must be pre-positioned first on the drilling centre point in the selected interpolation plane and the sensor ball within the drilling.

If for example G17 is defined, the centre point of the calibration drilling must in (P150)+0 and (P150)+1 and the diameter in (P150)+3 to be defined.

The cycle positions first the 1. and 2. axis in the interpolation plane to the programmed centre point of the drilling. Now the cycle moves the 1. axis in positive direction to the drilling edge. The axis starts with the feed of the NC program and reduces the feed to F:p143 at the distance of P141 (safety margin) before the expected drilling edge. If a measuring signal before the safety margin occurs, a message is output (M8167).

From the safety margin the cycle moves the axis to the expected drilling edge plus confidence interval. If there is still no signal, a message (M8136) appears.

If there is a signal of the sensor, immediately the axis movement stops and the cycle calculates the point of trigger of the sensor and writes it in (P149)+n (axis movement positively: Point of trigger negatively).

If P144 is > 1, the cycle returns to the safety margin. From here the cycle starts a new measuring movement to the same drilling edge. Afterwards the cycle looks the other side up of the drilling. Become subsequently, the movements with the 2. Axis executed.

If the cycle is finished, then is the data field (P149)+n with the points of trigger of the sensor described. Following measuring cycles can operate with these data. The position of the moved axes now is in the center of the drilling.

Parts it safely,
- that the tool radius of the sensor is correct (P8160).
- to operate in the null point, in which the parameter data (P150) apply.

5.1 G69 measuring cycles (continued)

Example: The calibration of the sensor in drilling G69 P140:1

```
N20     T1   M16
N30     G17  G53  G0   X600.000  Y300.000  Z280.000  A0  B0
N40     G1   Z270      F3000

N100                                                 {Cycle G69 / 1 initialize }
N110    P140:1                                       {Cycle type}
N120    P141:3                                       {Safety margin }
N130    P143:100                                     {Measuring speed }
N140    P144:2                                       {Number of the measurements }
N150    P147:500                                     {Pointer on tolerance data }
N160    P504:5                                       {Confidence range }
N170    P149:510                                     {Pointer on trigger data }
N180    P150:520                                     {Pointer on measuring data }
N182    P520:624.96    P521:324.3P523:54             {I, J, D }
N230    G69                                          {Cycle call }

N240    G0   Z280      M30
```

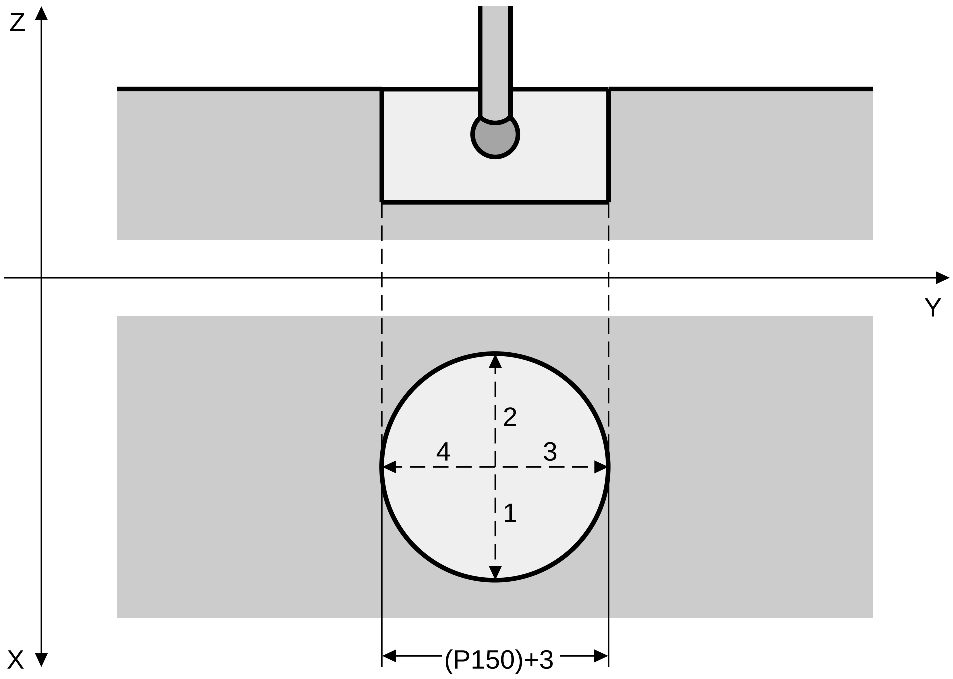


Figure 5-1

5.1 G69 measuring cycles (continued)

Calibration of the sensor at area G69 P140:2

The cycle requires a calibrationing area and a sensor. Measuring becomes with the 3. axis of the selected interpolation plane executed (definition of G17...).
The sensor must be pre-positioned on the calibrationing area.

If for example G17 is defined, the position on the calibrationing area must in (P150)+2 to be defined.

The cycle moves now 3. axis toward the calibrationing area. The axis starts with the feed of the NC program and reduces the feed to F:p143 at the distance of P141 (safety margin) before the expected area. If a measuring signal before the safety margin occurs, a message is output (M8167).

From the safety margin the cycle moves the axis to the expected area plus confidence interval. If there is still no signal, a message (M8136) appears.

If there is a signal of the sensor, immediately the axis movement stops and the cycle calculates the point of trigger of the sensor and writes it in (P149)+n (axis movement positively: Point of trigger negatively).

If P144 is > 1, the cycle turns back to the safety margin. From here the cycle starts a new measuring movement to the same area.

If the cycle is finished, then is the data field (P149)+n with the points of trigger of the sensor described. Following measuring cycles can operate with these data. The position of the moved axis now is in the safety margin.

Parts it safely,
- that the tool radius of the sensor is correct (P8160, P8162).
- to operate in the null point, in which the parameter data (P150) apply.

5.1 G69 measuring cycles (continued)

Example: The calibration of the sensor in drilling G69 P140:2

```
N20     T1   M16
N30     G17  G53  G0   X600.000  Y300.000  Z280.000  A0   B0
N40     G1   Z270      F3000

N100                                      {Cycle G69 / 2 initialize }
N110    P140:2                            {Cycle type}
N120    P141:3                            {Safety margin }
N130    P143:100                          {Measuring speed }
N140    P144:2                            {Number of the measurements }
N150    P147:500                          {Pointer on tolerance data }
N160    P504:5                            {Confidence range }
N170    P149:510                          {Pointer on trigger data }
N180    P150:520                          {Pointer on measuring data }
N182    P522:260.20                       {K }
N230    G69                               {Cycle call }

N240    G0   Z280      M30
```

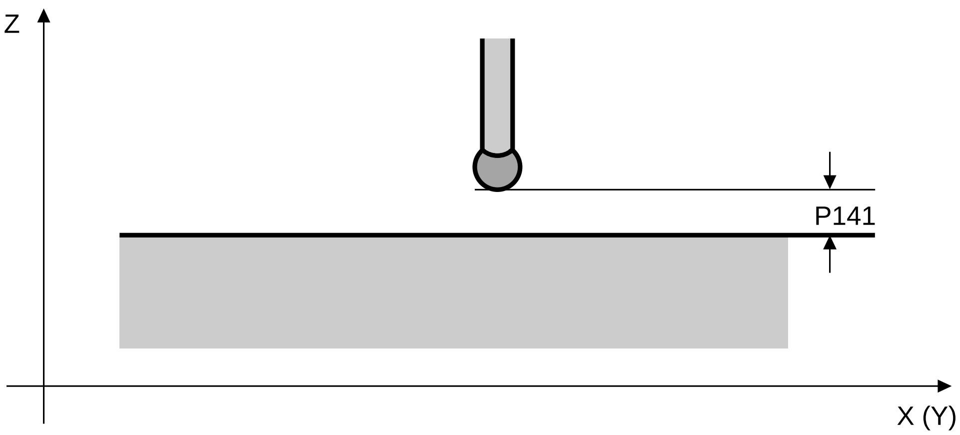


Figure 5-2

5.1 G69 measuring cycles (continued)

Determining the center of a drilling G69 P140:60

The measurement is executed with both axes of the selected interpolation plane (see G17...). The sensor should be at the beginning approximately on the drilling centre point in the selected interpolation plane and the sensor ball within the drilling pre-positioned.

The cycle moves the 1. axis in positive direction to the drilling edge. The axis starts with the feed of the NC program and reduces the feed to F:p143 at the distance
(diameters / 2 - safety margin) = (P152 / 2 - P141).

If a measuring signal before this point occurs, a message is output (M8167).

From here the cycle moves the axis to the expected drilling edge plus confidence interval. If there is still no signal, a message (M8136) appears.

If there is a signal of the sensor, immediately the axis movement stops and the cycle calculates the edge position.

If P144 is > 1, the cycle turns back to the safety margin. From here the cycle starts a new measuring movement to the same drilling edge. Afterwards the cycle looks the other side up of the drilling. Now become the movements with the 2. axis made.

If the cycle is finished, then is the data field (P154)+n with the drilling centre point described. The position of the moved axes now is in the center of the drilling.

Parts it safely that the tool radius of the sensor is correct (P8161, P8163).

5.1 G69 measuring cycles (continued)

Example: The calibration of the sensor in drilling G69 P140:60

```
N20     T1   M16
N30     G17  G53  G0   X600.000  Y300.000  Z280.000  A0   B0
N40     G1   Z270      F3000

N100                                         {Cycle G69 / 60 initialize }
N110    P140:60                              {Cycle type}
N120    P141:3                               {Safety margin }
N130    P143:100                             {Measuring speed }
N140    P144:2                               {Number of the measurements }
N150    P147:500                             {Pointer on tolerance data }
N160    P504:5                               {Confidence range }
N170    P149:510                             {Pointer on trigger data }
N180    P150:520                             {Pointer on measuring data }
N182    P154:530                             {Pointer on output }
N230    G69                                  {Cycle call }

N240    G0   Z280      M30
```

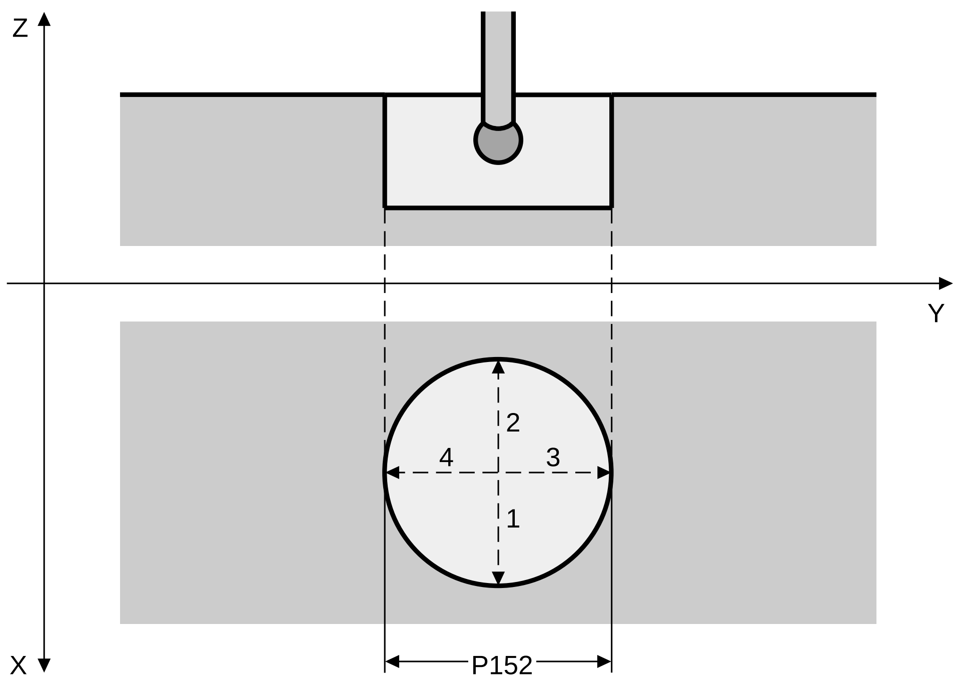


Figure 5-3

5.1 G69 measuring cycles (continued)

determining an individual item in the area G69 P140:40

The sensor must be pre-positioned at the beginning over the contact surface.
The measurement is executed with the axis defined in P148.

The cycle starts into negative direction with measuring speed (P143)
If there is still no signal of the sensor, the axes in the position (initial position - confidence interval (P147)+4) stop and a message (M8136) appears.

If there is a signal of the sensor, immediately the axis movement stops and the cycle calculates the position on the area and writes the data in (P154)+n.

Example G17	with P148:3	writes after cycle	P(154)+2
	with P148:2	writes after cycle	P(154)+1
	with P148:1	writes after cycle	P(154)+0

If P144 is > 1, the cycle turns back to the safety margin. From here the cycle starts a new measuring movement to the same area.

The position of the moved axes is in the safety margin.

Parts it safely,
- that the tool length of the sensor is correct (P8161, P8163), if 3. axis is selected
- that the tool radius of the sensor is correct (P8160, P8162), if 1. or 2. axis is selected

5.1 G69 measuring cycles (continued)

Example: The calibration of the sensor in drilling G69 P140:40

```
N20     T1   M16
N30     G17  G53  G0   X600.000  Y300.000  Z280.000  A0   B0
N40     G1   Z270      F3000

N100                                                  {Cycle G69 / 40 initialize }
N110    P140:40                                       {Cycle type}
N120    P141:3                                        {Safety margin }
N130    P143:100                                      {Measuring speed }
N140    P144:2                                        {Number of the measurements }
N150    P147:500                                      {Pointer on tolerance data }
N160    P504:5                                        {Confidence range }
N170    P144:3                                        {Measuring axes }
N180    P149:510                                      {Pointer on trigger data }
N182    P150:520                                      {Pointer on measuring data }
N230    G69                                           {Cycle call }

N240    G0   Z280      M30
```

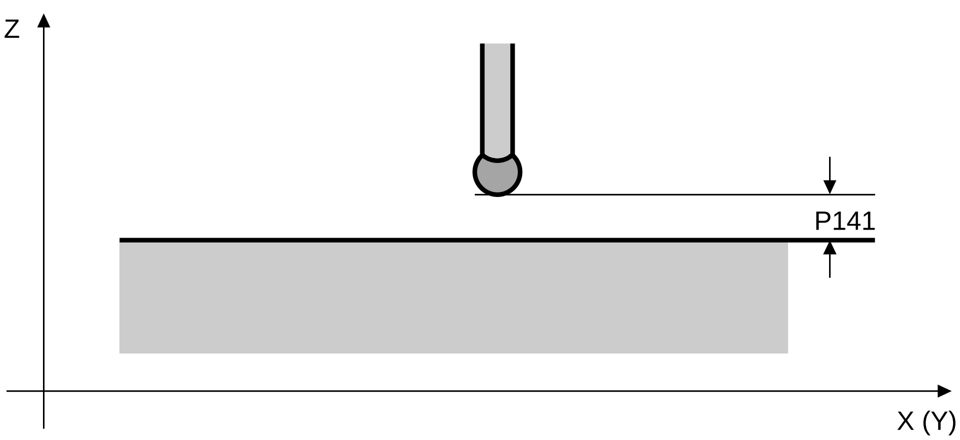


Figure 5-4

5.2 G71 / G72 / G73 / G74 / G75 Milling cycles

List of canned milling cycles

G71 Rectangular pocket roughing, conventional
G72 Rectungular pocket roughing, conventional and climb
G73 Rectangular pocket roughing and finishing
G74 Circular pocket roughing
G75 Circular pocket roughing and finishing

The millinging cycles are block by block effective.
The parameter inputs remain against it modal effectively.

In order to avoid over regulation, all parameters are to be reset, with a programming type not to be used.

Messages

The numbers for cycle messages start with 8xxx.
A list can be found in section messages

5.2.1 G71 Rectangular pocket roughing, conventional

Starting point definition

Before starting any machining cycles, the milling cutter should be above the workpiece by the safety margin and at the centre of the pocket.

Whether the starting point S1 or S2 is homed into depends on the sign of the programmed in-feed P16.
Positive sign: starting point S1
Negative sign: starting point S2

The milling direction is established by the CNC and always in the direction of the longer pocket side. If the pocket sides are the same, milling takes place in the X direction.

Parameter input

All parameters except for P16 should have positive sign when inputting.

Parameter	Significance
P11	Pocket dimension in X direction
P12	Pocket dimension in Y direction
P13	Pocket depth in Z direction
P14	Corner radius
P15	Contour allowance in X and Y directions
P16	In-feed dimension in X or Y directions (prefix + or -)
P17	In-feed dimension in Z direction
P18	Pocket depth allowance in Z direction
P19	Safety margin in Z direction
P21	Feed rate in Z direction (when plunging into material)

During cycle processing a possibly programmed correction becomes G41 / G42 switched off, since the tool radius correction in the cycle is considered.

Feed in z-direction (only when immersing into the material) is programmable with P21.
Is P21 programmed, does not become immersing into the material with under F programmed feed executed.

5.2.1 G71 Rectangular pocket roughing, conventional (continued)

Machining sequence

- Positioning in the pocket center and in the safety margin P19
- Home in on starting point S1 or S2 (depending on prefix of P16)
- In-feed in Z direction at feed rate P21 (if programmed)
- Conventional milling to pocket dimension less allowance P15 in X and Y directions
- Retraction of milling cutter by 2 mm in Z direction
- Retraction in X or Y direction
- In-feed in Z direction
- In-feed in X or Y direction
- In-feed repetition until pocket dimension less allowance P15 is reached
- Milling over burrs
- Retraction in Z direction to safety margin
- Retraction to starting position in XY plane
- Repetition of in-feeds until pocket depth less allowance P18 is reached
- Retraction to starting position (centre of pocket)

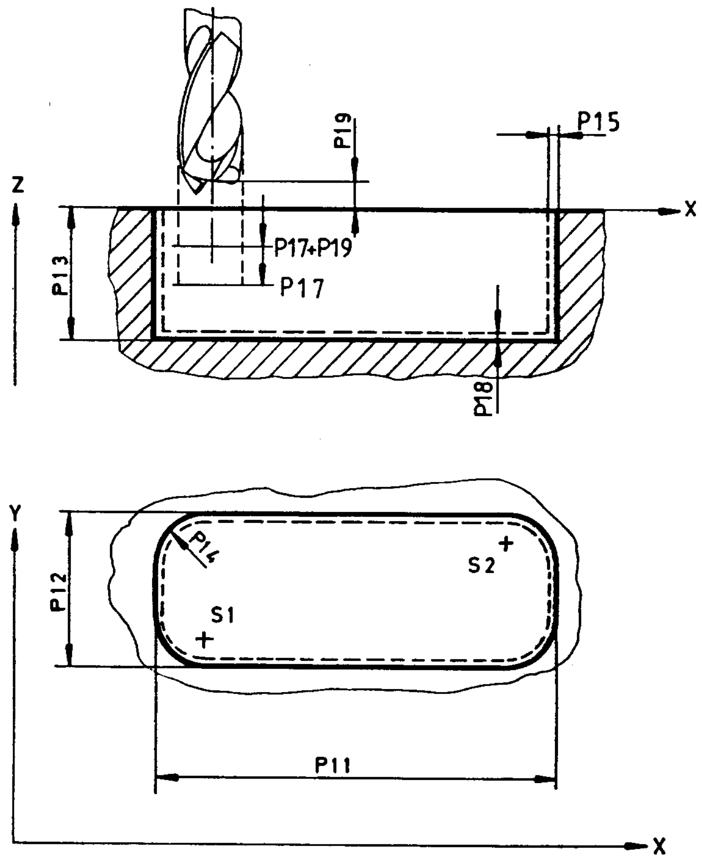


Figure 5-5

5.2.1 G71 Rectangular pocket roughing, conventional (continued)

G 71 Example 1: Pocket dimensions X=71mm, Y=41mm, Z=5mm; Tool-radius=3mm

N10	G00	G54	G90	F2500	S900	T02	M06	M03	M07
N20	G71	X55,5	Y40,5	Z2	P11:71	P12:41	P13:5		
	P14:--	P15:0	P16:4	P17:3	P19:2	P21:100			M30

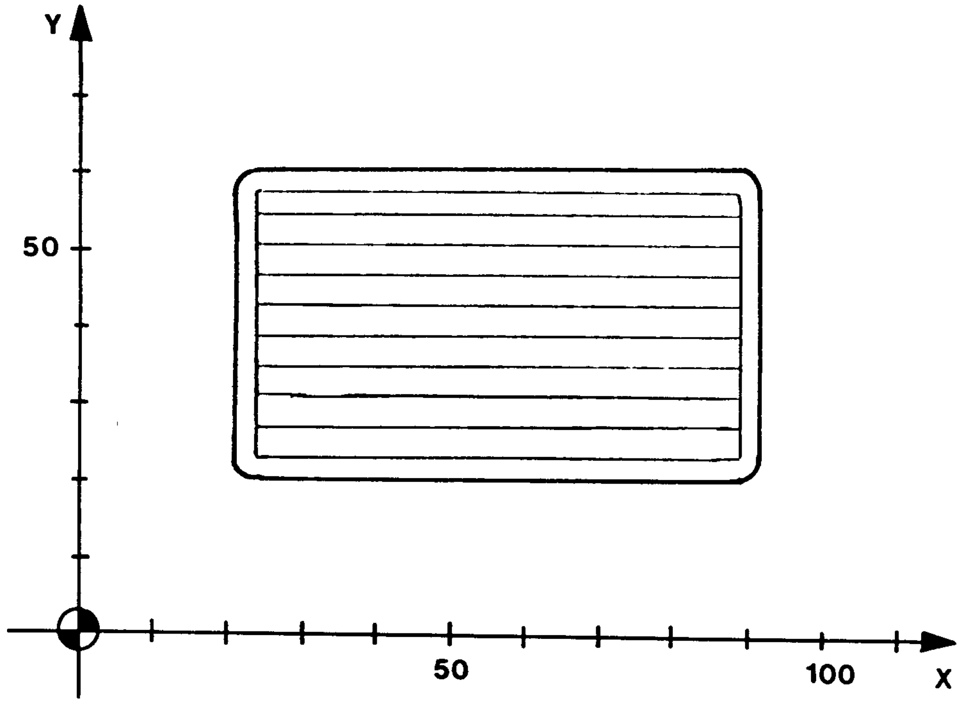


Figure 5-6

Parameters:

P11, P12, P13	Pocket dimensions in X, Y, Z direction
P16, P17	In-feed in Y, Z direction
P19	Safety margin in Z direction
P21	Feed rate in Z direction

Run: Block 10 Zero point, absolute dimension, feed rate, spindle, tool
Block 20 Cycle call-up, positioning on centre of pocket X, Y, Z, pocket dimensions P11, P12, P13, in-feed dimensions P16, P17, safety margin P19

Remarks: Since the corner radius P14 is not programmed, it becomes with the programmed tool radius equated.

5.2.1 G71 Rectangular pocket roughing, conventional (continued)

G71 Example 2: Pocket dimensions X=46mm, Y=81mm, Z=30mm;
Tool-radius=3mm, corner radius=13mm

N10	G00	G54	G90	F1000	S750	T02 M06	M03 M07
N20	G71	X43	Y60,5	Z1	P11:46	P12:81	P13:30
	P14:13	P16:-4	P17:7	P19:1	P21:400	M30	

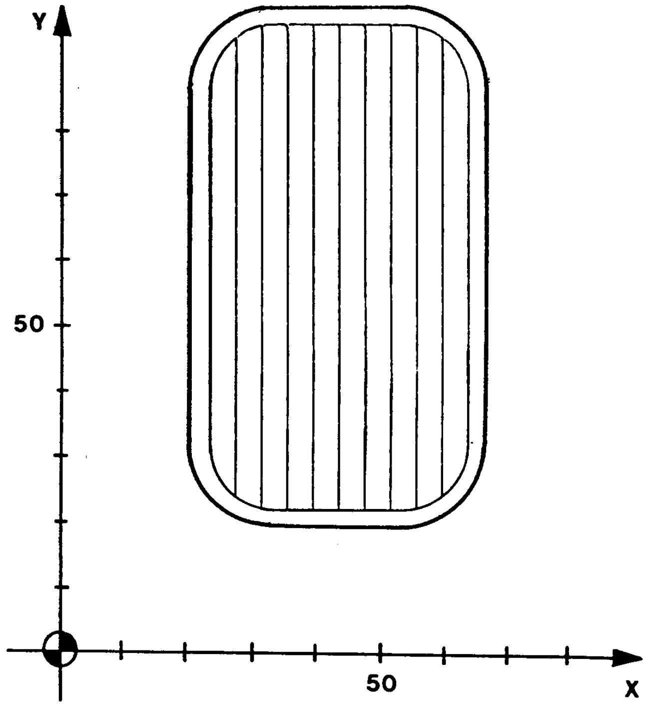


Figure 5-7

Parameters:

P11, P12, P13	Pocket dimensions in X, Y, Z direction
P14	Corner radius
P16, P17	In-feed in X, Z direction
P19	Safety margin in Z direction
P21	Feed rate in Z direction

Run:	Block 10	Zero point, absolute dimension, feed rate, spindle, tool
	Block 20	Cycle call-up, positioning on centre of pocket X, Y, Z; pocket dimensions P11, P12, P13, in-feed dimensions P16, P17, corner radius P14, safety margin P19

Remarks: If P14 = 0 is or smaller than the tool radius, then becomes after that start clearing the program execution stopped and message 8001 indicated.

5.2.2 G72 Rectangular pocket roughing, climb and conventional

Starting point definition

Before the start of the machining cycles the milling cutter should be above the workpiece by the safety margin and in the middle of the pocket.

Whether the starting point S1 or S2 is homed into depends on the sign of the programmed in-feed P16.
Sign, positive: starting point S1
Sign, negative: starting point S2

The determination of the milling direction takes place in the CNC and always in the direction of the longer pocket side. If the pocket sides are the same, milling takes place along the X direction.

Parameter input

All parameters except for P16 should have a positive sign during inputting.

Parameter	Significance
P11	Pocket dimension in X direction
P12	Pocket dimension in Y direction
P13	Pocket depth in Z direction
P14	Corner radius
P15	Contour allowance in X and Y directions
P16	In-feed dimension in X or Y directions (sign +/-)
P17	In-feed dimension in Z direction
P18	Pocket depth allowance in Z direction
P19	Safety margin in Z direction
P21	Feed rate in Z direction (when plunging into material)

5.2.2 G72 Rectangular pocket roughing, climb and conventional (continued)

Machining sequence

- Positioning in the pocket center and in the safety margin P19
- Home in on starting point S1 or S2 (each according to sign of P16)
- In-feed in Z direction at feed rate P21 (if programmed)
- Conventional milling to pocket dimension less the allowance P15 in X and Y direction
- In-feed in X or Y direction
- Milling in conventional direction
- In-feed in X or Y direction
- In-feed repetition until pocket dimension less allowance P15 is reached
- Milling over burrs along edge
- Retraction in Z direction to safety margin
- Retraction to starting position in XY plane
- In-feed in Z direction to previous dimension
- Milling over burrs along other edge
- Retraction in Z direction to safety margin
- Retraction to starting position in XY plane
- Repetition of in-feeds until pocket depth less allowance P18 is reached
- Retraction to starting position (centre of pocket)

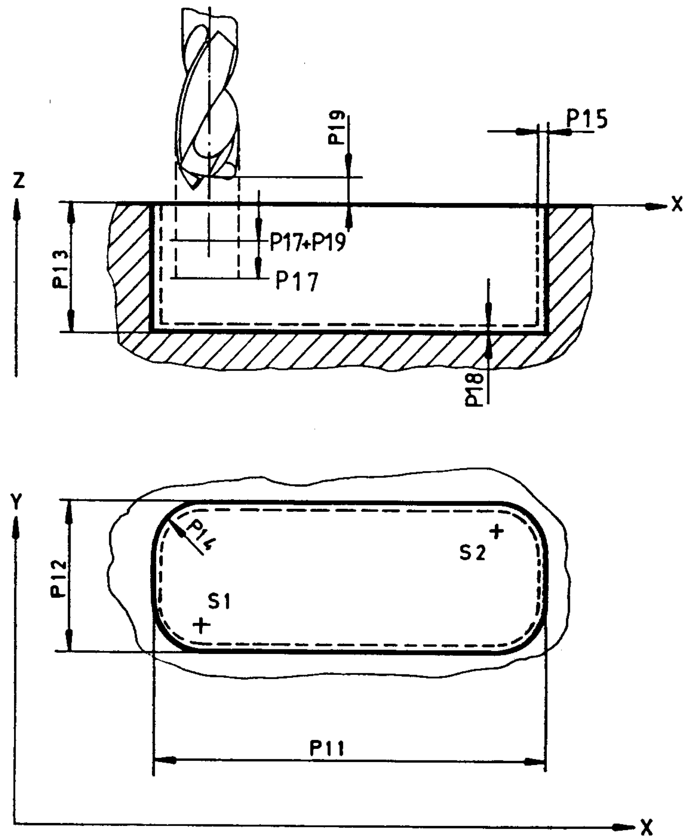


Figure 5-8

5.2.2 G72 Rectangular pocket roughing, climb and conventional milling (cont.)

G72 Example 1: Pocket dimensions X=66mm, Y=31mm, Z=5mm; Tool-radius=3mm

N10	G00	G54	G90	F2000	S850	T02	M06	M03	M07	
N20	G72	X53	Y35,5	Z1	P11:66	P12:31	P13:5	P14:--		P15:0
	P16:5,1		P17:3	P19:1	M30					

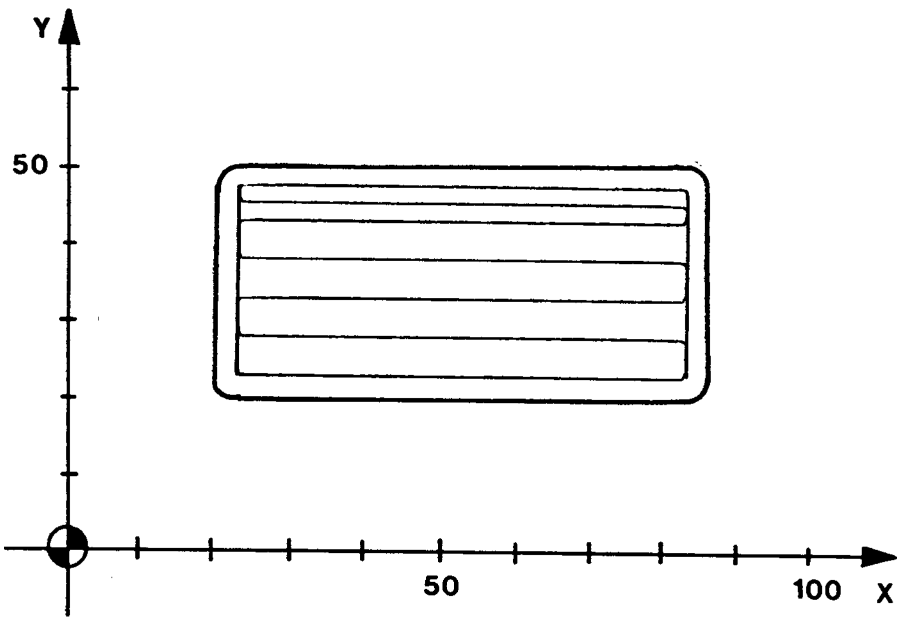


Figure 5-9

Parameters:

P11, P12, P13	Pocket dimensions in X, Y, Z direction
P16, P17	In-feed in Y, Z direction
P19	Safety margin in Z direction

Run:	Block 10	Zero point, absolute dimension, feed rate, spindle, tool
	Block 20	Cycle call-up, positioning on centre of pocket X, Y, Z; pocket dimensions P11, P12, P13, in-feed dimensions P16, P17, safety margin P19

Remarks: As the corner radius P14 is not programmed it is identified along with the programmed tool radius.

5.2.2 G72 Rectangular pocket roughing, climb and conventional milling (cont.)

G72 Example 2: Pocket dimensions X=44mm, Y=69mm, Z=3mm; Tool radius=3mm, corner radius=15mm

N10	G00	G54	G90	F2000	S950	T02	M06	M03	M07
N20	G72	X42	Y54,5	Z1	P11:44	P12:69	P13:3	P14:15	P15=0
	P16:5,5		P17:3	P19:1	P21:100	M30			

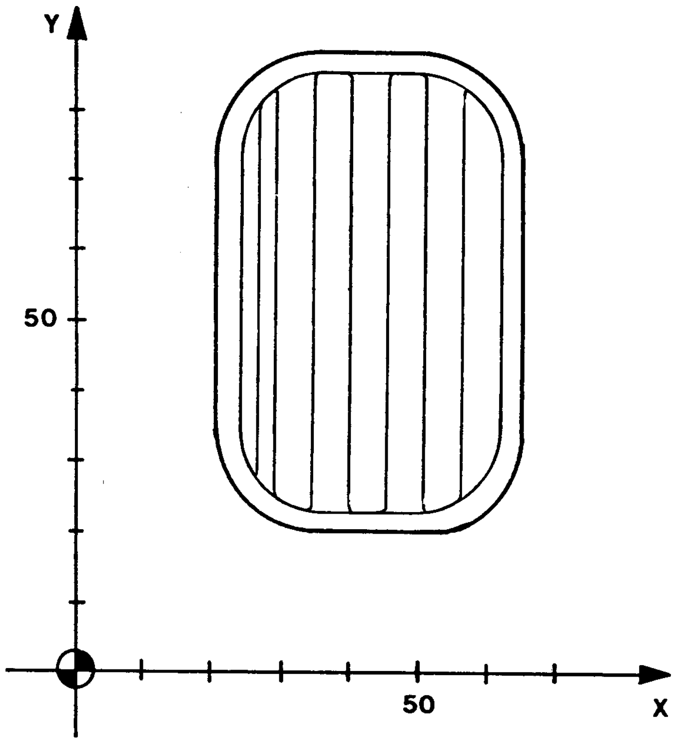


Figure 5-110

Parameters:

P11, P12, P13	Pocket dimensions in X, Y, Z direction
P14	Corner radius
P16, P17	In-feed in X, Z direction
P19	Safety margin in Z direction
P21	Feed rate in Z direction

Run:	Block 10	Zero point, absolute dimension, feed rate, spindle, tool
	Block 20	Cycle call-up, positioning on centre of pocket X, Y, Z; pocket dimensions P11, P12, P13, in-feed dimensions P16, P17, corner radius P14, safety margin P19

Remarks: If P14=0 or is smaller than the tool radius the program run is shut down after the start and fault feed-back 1301 (geometry fault) is displayed.

5.2.3 G73 Rectangular pocket roughing and finishing

Starting point definition

Before the start of the machining cycles, the milling cutter should be above the workpiece at the safety margin and in the middle of the pocket.

Whether the starting point S1 or S2 is homed into depends on the sign of the programmed in-feed P16.
Sign, positive: starting point S1
Sign, negative: starting point S2

The milling direction is established in the CNC and is always in the direction of the longer pocket side. If the pocket sides are the same, milling takes place along the X direction.

Parameter input

All parameters except for P16 should have a positive sign when inputting.

Parameter	Significance
P11	Pocket dimension in X direction
P12	Pocket dimension in Y direction
P13	Pocket depth in Z direction
P14	Corner radius
P15	Contour allowance in X and Y direction
P16	In-feed dimension in X or Y direction (sign +/-)
P17	In-feed dimension in Z direction
P18	Pocket depth allowance in Z direction
P19	Safety margin in Z direction
P20	In-feed dimension when finishing, to remove allowance P15; In-feed direction as for P16
P21	Feed rate in Z direction (when plunging into material)

5.2.3 G73 Rectangular pocket roughing and finishing (continued)

Machining

- Positioning in the pocket center and in the safety margin P19
- Home in on starting point S1 or S2 (each according to sign of P16)
- In-feed in Z direction at feed rate P21 (if programmed)
- Milling pocket contour (less allowance P15) to starting point
- In-feed repetition until pocket depth P13 less allowance P18 is reached
- Retraction to starting plane
- In-feed in X or Y direction
- In-feed in Z direction at feed rate P21 (if programmed)
- Conventional and climb milling of remaining material
- Retraction in Z direction and to starting position and in-feed
- In-feed repetition until pocket dimension less allowance P18 is reached.
- Retraction in Z direction to starting plane
- In-feed for finishing
- In-feed in Z direction at 0.5 of the programmed feed rate F to pocket depth less P18
- Finishing contour to starting point at 0.5 of the programmed feed rate
- In-feed repetition until finished contour is reached
- Retraction to starting position at programmed feed rate

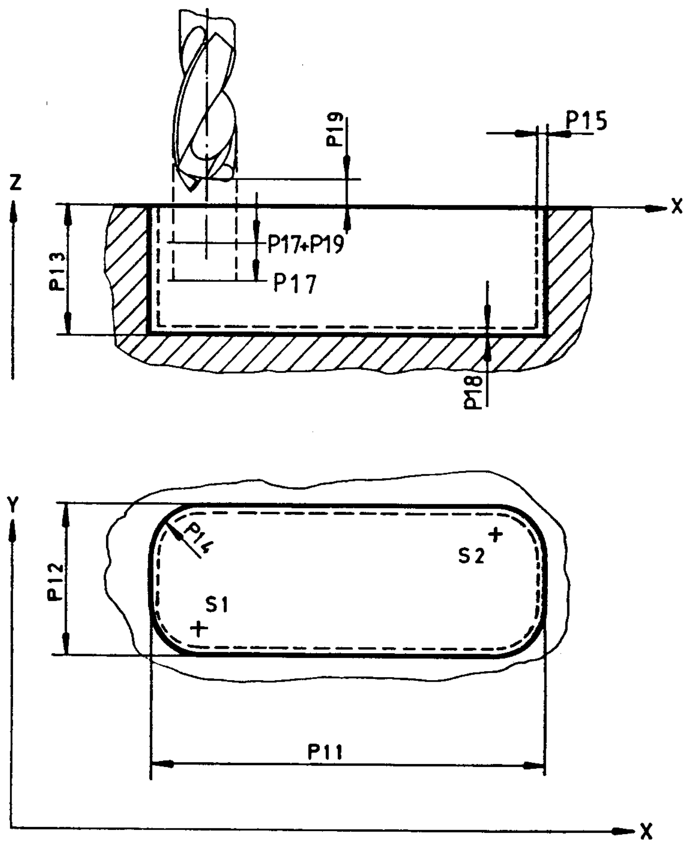


Figure 5-11

5.2.3 G73 Rectangular pocket roughing and finishing (continued)

G73 Example 1: Pocket dimensions X=64mm, Y=35mm, Z=5mm; Tool radius=3mm

N10 G00 G54 G90 F1500 S950 T02 M06 M03 M07

N20 G73 X52,5 Y37,5 Z1 P11:64 P12:35 P13:5 P14=0
P15:1 P16:4,5 P17:2 P19:1 P20:1 M30

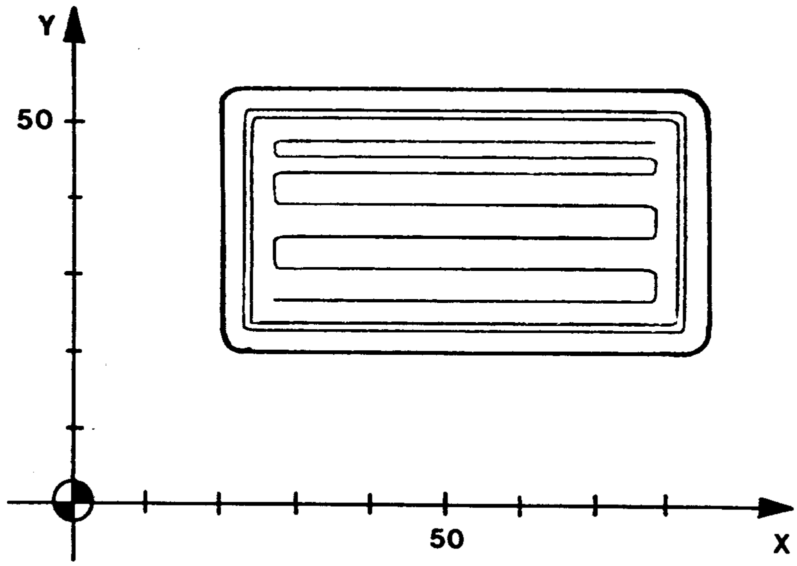


Figure 5-12

Parameters:

P11, P12, P13	Pocket dimensions in X, Y, Z direction
P15	Allowance in X and Y direction
P16, P17	In-feed in Y, Z direction
P19	Safety margin in Z direction
P20	Allowance during finishing

Run:

Block 10	Zero point, absolute dimension, feed rate, spindle, tool
Block 20	Cycle call-up, positioning on centre of pocket X, Y, Z; pocket dimensions P11, P12, P13, in-feed dimensions P16, P17, allowance P15, safety margin P19, in-feed dimension when finishing P20

Remarks: As the corner radius P14 is not programmed it is made equal to the programmed tool radius.

5.2.3 G73 Rectangular pocket roughing and finishing (continued)

G73 Example 2: Pocket dimensions X=65mm, Y=35mm, Z=5mm; Tool radius=3mm, corner radius=11mm

N10	G00	G54	G90	F1500	S950	T02	M06	M03	M07
N20	G73	X52,5	Y37,5	Z1	P11:65	P12:35	P13:5	P14:11	
	P15:1,5		P16:4,2	P17:2	P19:1	P20:1	M30		

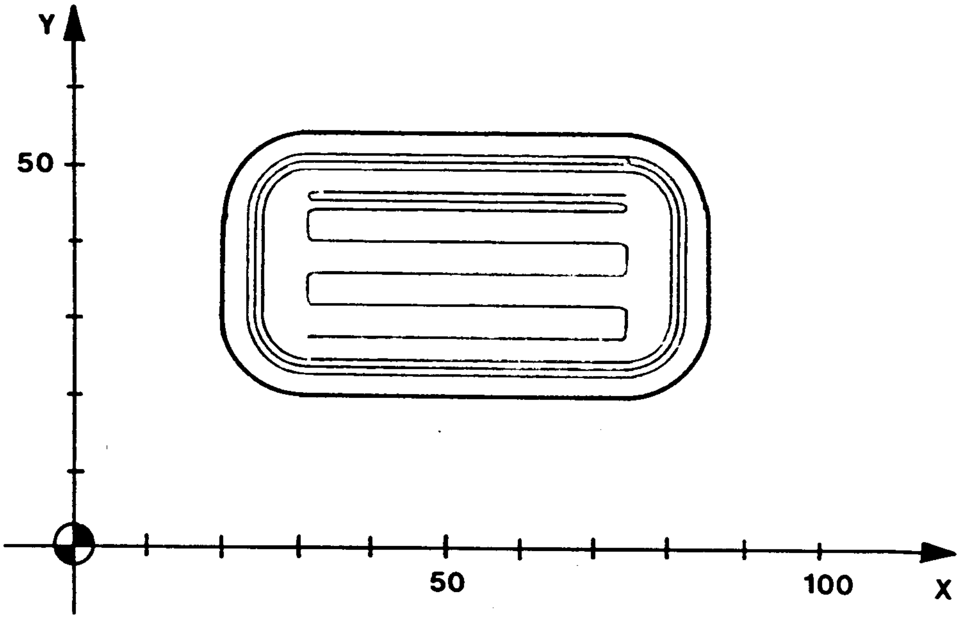


Figure 5-13

Parameters:

P11, P12, P13	Pocket dimensions in X, Y, Z direction
P14	Corner radius
P15	Allowance in X and Y direction
P16, P17	In-feed in Y, Z direction
P19	Safety margin in Z direction
P20	Allowance during finishing

Run: Block 10 Zero point, absolute dimension, feed rate, spindle, tool

Block 20 Cycle call-up, positioning on centre of pocket X, Y, Z; pocket dimensions P11, P12, P13, in-feed dimensions P16, P17, corner radius P14, safety margin P19, allowance P15, infeed dimension during finishing P20

Remarks: If P14=0 or is smaller than the tool radius the program run is shut down after the start and message 1301 (geometry fault) is displayed.

5.2.4 G74 circular pocket roughing

Starting point definition

Before the start of the machining cycles the milling cutter should be above the workpiece at the safety margin and at the centre of the pocket.

Parameter input

All parameters except for P16 should have a positive sign when inputting. The sign of P16 determines the direction of milling, i.e.:
Milling direction G02, the sign of P16 is positive
Milling direction G03, the sign of P16 is negative

Parameter	Significance
P11	Internal radius (if core drilled)
P12	Depth of pocket
P14	Pocket radius
P15	Allowance at circumference
P16	In-feed dimension in X direction
P17	In-feed dimension in Z direction
P18	Pocket depth allowance in Z direction
P19	Safety margin
P21	Feed rate in Z direction (when plunging into material)

When specifying the internal radius P11, the first in-feed in the X direction is corrected by the appropriate amount: 1st in-feed = P11 + P16.

5.2.4 G74 circular pocket roughing (continued)

Machining

- Positioning in the pocket center and in the safety margin P19
- In-feed in Z direction at feed rate P21 (if programmed)
- In-feed in X direction
- Spiral milling
- Milling circular pocket to finished size less allowance P15
- Retraction along the semicircle with R=2.5mm max.
- Retraction in Z direction by 2mm
- Retraction to centre of pocket
- In-feed repetition until pocket dimension less allowance P18 in Z direction is reached
- Retraction to starting position

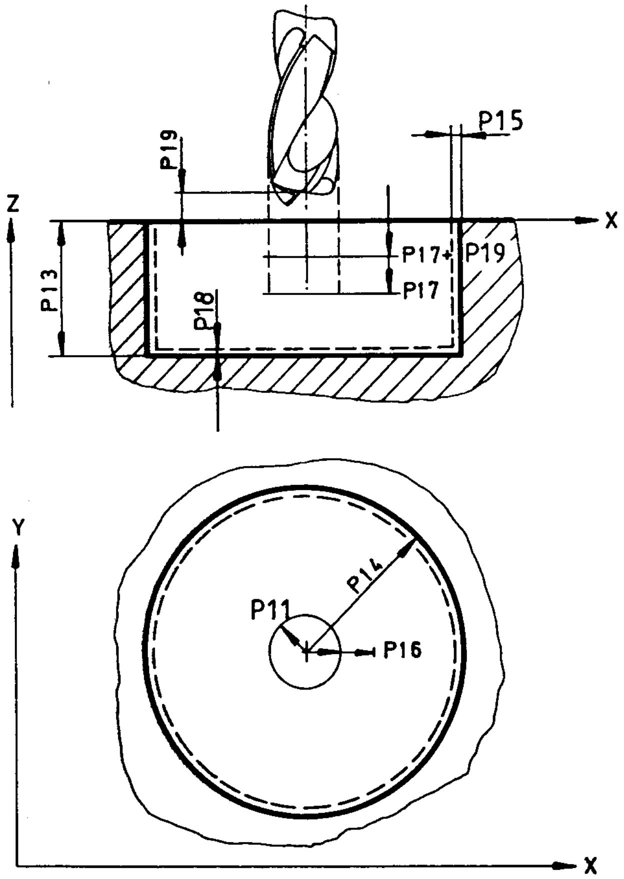


Figure 5-14

5.2.4 G74 Circular pocket roughing (continued)

G74 Example 1: Pocket radius R=33mm, pocket depth Z=27mm; Tool radius=3mm

N10	G00	G54	G90	F1500	S950	T02	M06	M03	M07	
N20	G74	X53	Y53	Z1	P13:27	P14:33	P16:5,1	P17:5	P19:1	M30

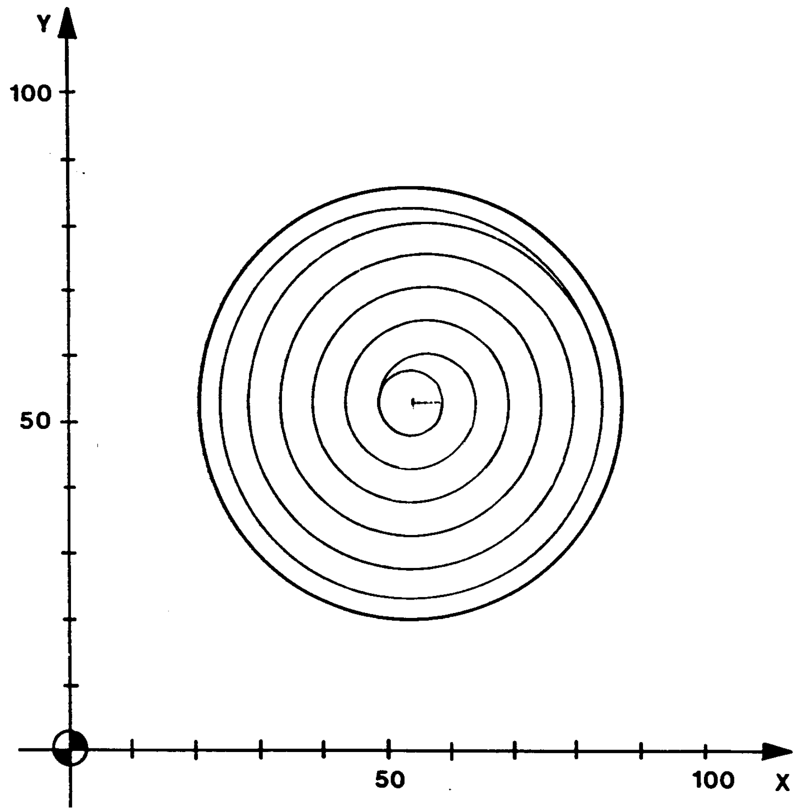


Figure 5-15

Parameters:

P13	Pocket depth in Z direction
P16, P17	In-feed in X, Z direction
P14	Pocket radius
P19	Safety margin in Z direction

Run:	Block 10	Zero point, absolute dimension, feed rate, spindle, tool
	Block 20	Cycle call-up, positioning on centre of pocket X, Y, Z, pocket depth P13, in-feed dimensions P16, P17, pocket radius P14, safety margin P19

5.2.4 G74 circular pocket roughing (continued)

G74 Example 2: Pocket radius R=30mm, pocket depth Z=31mm; Tool radius=3mm, core bore drilling at 20mm dia.

N10 G00 G54 G90 F1500 S950 T02 M06 M03 M07

N20 G74 X50 Y50 Z1 P11:10 P13:31 P14:30 P16:4 P17:8
P19:1 M30

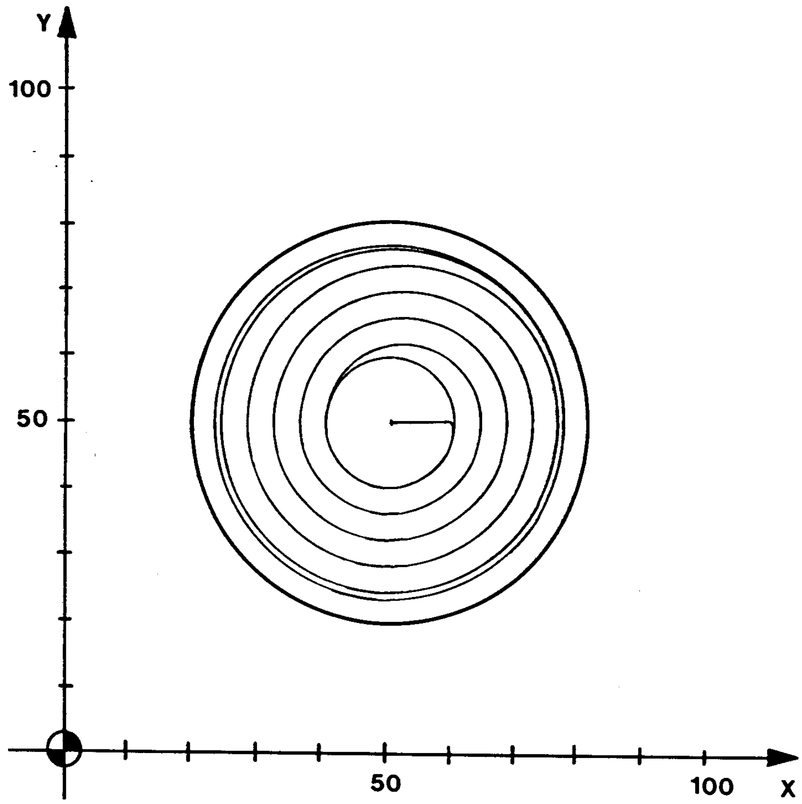


Figure 5-16

Parameters:
P11 Internal radius (if core drilled)
P13 Pocket depth in Z direction
P16, P17 In-feed in X, Z direction
P14 Pocket radius
P19 Safety margin in Z direction

Run: Block 10 Zero point, absolute dimension, feed rate, spindle, tool
Block 20 Cycle call-up, positioning on centre of pocket X, Y, Z, pocket depth P13, in-feed dimensions P16, P17, pocket radius P14, safety margin P19

5.2.5 G75 circular pocket roughing and finishing

Starting point definition

Before the start of the machining cycles the milling cutter should be above the workpiece at the safety margin and at the centre of the pocket.

Parameter input

All parameters except for P16 should have a positive sign when inputting. The sign of P16 determines the direction of milling, i.e.:
Milling direction G02, the sign of P16 is positive
Milling direction G03, the sign of P16 is negative

Parameter	Significance
P11	Internal radius (if drilled)
P13	Depth of pocket
P14	Pocket radius
P15	Allowance at circumference
P16	In-feed dimension in X direction
P17	In-feed dimension in Z direction
P18	Pocket depth allowance in Z direction
P19	Safety margin
P20	In-feed dimension when finishing, to remove P15, in-feed direction as with P16
P21	Feed rate in Z direction (when plunging into material)

With specification of the internal radius P11 the first feed in x-direction becomes around the suitable amount corrects: 1. Feed = P11 + P16.

5.2.5 G75 circular pocket roughing and finishing (continued)

Machining

- Positioning in the pocket center and in the safety margin P19
- In-feed in Z direction at feed rate P21 (if programmed)
- In-feed in X direction
- Spiral milling
- Milling circular pocket to finished size less allowance P15
- Retraction along the semicircle with R=2.5mm max.
- Retraction in Z direction
- Retraction to centre of pocket
- In-feed repetition until pocket dimension less allowance P18 in Z direction is reached
- Retraction along the semicircle with R=2.5mm max.
- In-feed at half feed rate for finishing along the semicircle
- Milling along the circle at half of the programmed feed rate
- Retraction along the semicircle
- In-feed repetition until finished dimension is reached
- Retraction along the semicircle
- Retraction to starting position at programmed feed rate

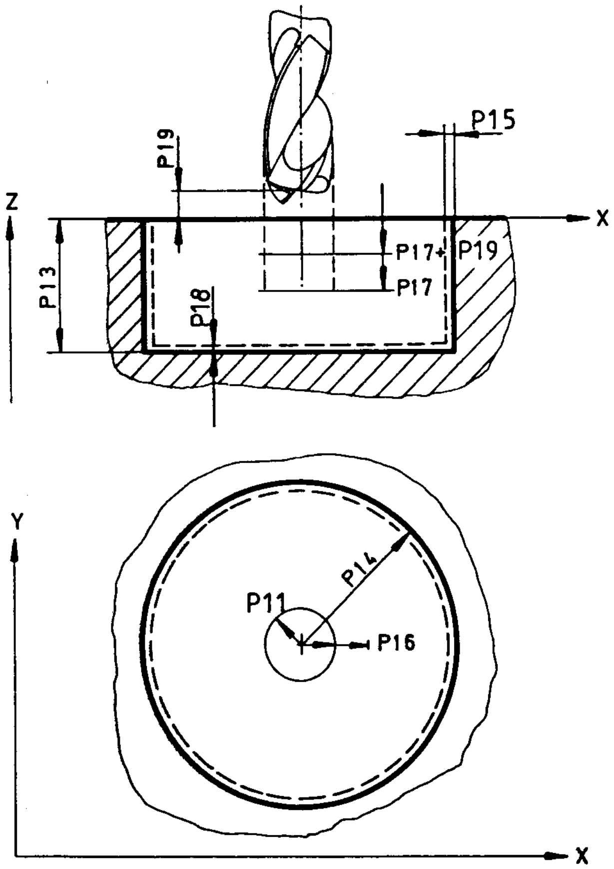


Figure 5-17

5.2.5 G75 circular pocket roughing and finishing (continued)

G75 Example 1: Pocket radius R=28mm, pocket depth Z=35mm; Tool radius=3mm

```
N10  G00  G54  G90  F1500   S950    T02     M06     M03      M07

N20  G75  X48  Y48  Z1      P13:35  P14:28  P15:2   P16:4,1  P17:8
     P19:1     P20:2    M30
```

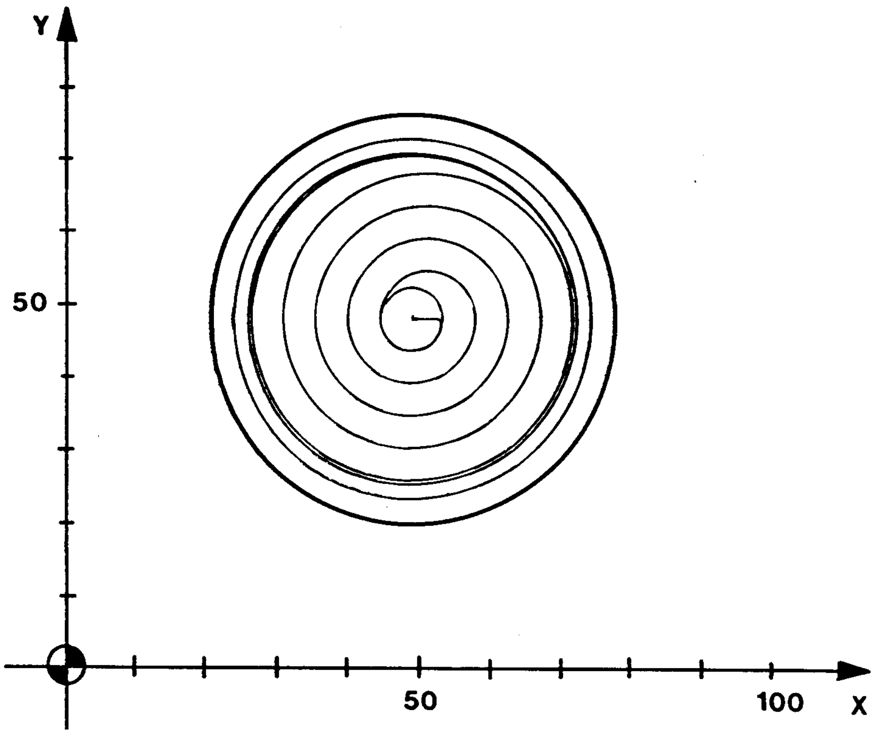


Figure 5-18

Parameters:

P13	Pocket depth in Z direction
P14	Pocket radius
P15	Allowance at circumference
P16, P17	In-feed in X, Z direction
P19	Safety margin in Z direction
P20	In-feed dimension during finishing

Run:

Run:	Block 10	Zero point, absolute dimension, feed rate, spindle, tool
	Block 20	Cycle call-up, positioning on centre of pocket X, Y, Z, pocket depth P13, in-feed dimensions P16, P17, pocket radius P14, safety margin P19, in-feed dimension during finishing P20

5.3 G81 / G83 / G84 / G85 Drilling cycles

The drill cycles are block by block effective.
The parameter inputs remain against it modal effectively.

In order to avoid over regulation, all parameters are to be reset, which are not used with a programming type.

List of drilling cycles

Drilling cycle according to DIN 66025		Programming in CNC900
G81	Drilling with rapid return	G81
G82	Drilling with rapid return and	G81, G04. Free cutting time free-cutting
G83	Deep hole drilling	G83
G84	Tapping	G84, G04. Spindle turnaround time
G85	Drilling with return at feed rate	G85

D

5.3 G81 / G83 / G84 / G85 Drilling cycles

Programming of characteristics

for the programming of the drill cycles only parameters can be used.

Parameter	Meaning		
P30	Free cutting time	Turn back time of the spindle after achieving the drilling depth	
P31	Compensating chuck	G84 P31 : 3 = thread cutting without compensating chuck right P31 : 4 = thread cutting without compensating chuck left P31 : 0 or - = G84 standard cycle (with compensating chuck)	
P32	Drilling feed		
P32	Thread pitch	bei G84	
P33	Drilling depth		absolute
P33	Thread depth	bei G84	absolute
P34	Anticipation plane	Safety margin of the processing upper edge	absolute
P35	Retreat plane	To run around of obstacles in the setting axis	absolute
P36	No. of strokes	G83, constant setting depth = (P33 - P34) / P36	
P36	Setting depth	G83, degressive setting depth	
P37	1st Stroke	G83, degressive setting depth	
P38	Safety margin	G83, presetting = 1mm	incremental
P39	Gradual decrease	G83, degressive setting depth P39 : 1 = on P39 : 0 or - = off	

Drilling depth P33, anticipation plane P34 and retreat plane P35 are dependant of the planes set over G17, G18 or G19. In the following examples, G17 is activated with the axis names X (1st axis), Y (2nd axis) and Z (3rd axis).

5.3 G81 / G83 / G84 / G85 Drilling cycles (continued)

Parameters for drilling cycles

Drilling cycle	Parameter									
	Free cutting time	Drilling feed	Drilling depth	Antici-pation plane	Retreat plane	Number of strokes	1st stroke	Safety margin	Gradual dec-rease	Dri-ling speed
	P30	P32	P33	P43	P35	P36	P37	P38	P39	S
G81	*	*	*	*	*	-	-	-	-	*
G83 constant	*	*	*	*	*	*	-	*	-	*
G83 degressiv	*	*	*	*	*	*	*	*	*	*
G84	*	*	*	*	*	-	-	-	-	*
G85	*	*	*	*	*	-	-	-	-	*

When calling up a drilling cycle, a direction of spindle rotation and spindle speed should be active. If these two values have already been set when the cycle is called up they are taken as the preset values for the drilling cycles.

Programming of a safety margin (P38) is optional. If this characteristic is not input a safety margin of 1 mm is preset.

When calling up a drilling cycle, the fields marked * should have been programmed in one way or another.

Messages

The numbers for cycle messages start with 8xxx.
A list can be found in section messages

Activation conditions

For a drilling cycle to be carried out, at least one of the X, Y or Z coordinates should be programmed per main block. This also applies to parameter P33 for the drilling depth Z.

5.3 G81 / G83 / G84 / G85 Drilling cycles (continued)

Positioning speed

The pre-positioning of the axes takes place at rapid traverse. When using polar coordinates programming with interpolation mode G02 or G03 set, positioning is at the programmed feed rate. It is therefore good policy to specify the feed rate with parameter P32.

Effect of G91 incremental dimensions

Using incremental dimensions, the positioning axes (XY with G17, ZX with G18, YZ with G19) are traversed in incremental mode. The specified coordinates are always taken as absolute for the characteristics (Z, R, K).

Effect of G66

A programmed G66 is effective throughout the complete drilling cycle.

Effect of G63/G64

The function, when set, is retained throughout the complete drilling cycle. The tapping cycle G84 automatically sets G63 for period of the in-feed traverse.

Effect of other additional functions

Other additonal functions are effective in a block with travel information (cycle is executed) before or after the travels that are defined in the cycle.

5.3.1 G81 Drilling with return at rapid travers

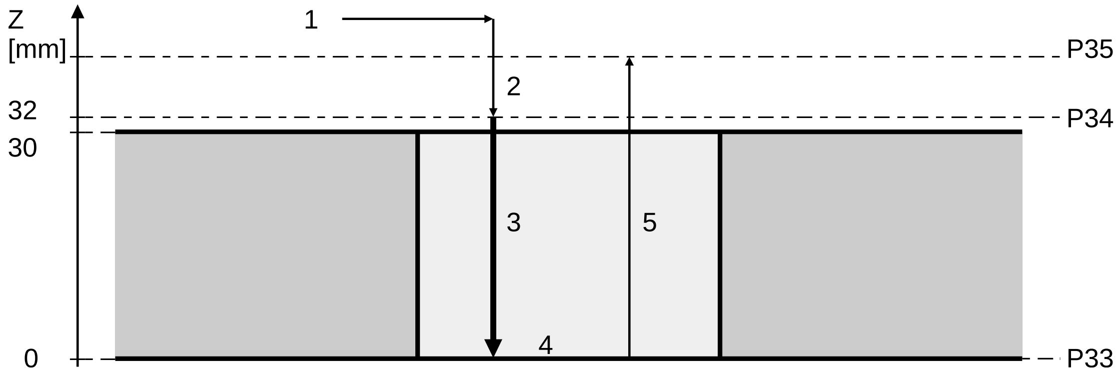


Figure 5-19

Meaning of the parameters

P30	Free cutting time	Turn back time of the spindle after achieving the drilling depth
P32	Drilling feed	
P33	Drilling depth	
P34	Anticipation plane	Safety margin of the processing upper edge
P35	Retreat plane	To go around of obstacles in the setting axis

5.3.1 G81 Drilling with return at rapid travers (continued)

Example

N20	G56	G00	X0	Y0	Z100	S1200	F1500	M03	T01	M16
N30	**G81**	Z80	P30:1		P32:800	**P33:0**	**P34:32**	P35:60		
N40	M05	...								

Sequence plan

1 Switch on spindle (M03, M04) and position X and Y axis in rapid traverse.

2 After reaching the X and Y position, position the Z axis to the anticipation plane (P34) in rapid traverse.

3 After reaching the anticipation plane the Z axis is traversed at working feed rate (P32) to the drilling depth (P33).

4 After reaching the drilling depth any possibly programmed free cutting time (P30) is awaited.

5 After it the Z-axis in rapid traverse moves on the retreat plane (P35) and afterwards the spindle is switched off (M05).

5.3.2 G83 Deep-hole drilling with constant setting depth

If the parameter P39:0 or is not programmed, the programmed drilling depth (P33) by repeated constant setting depth of the active drilling depth is achieved.

The active drilling depth is (P33 - P34) * n / P36 , however n = 1, 2, .. (settin depth = stroke) is.

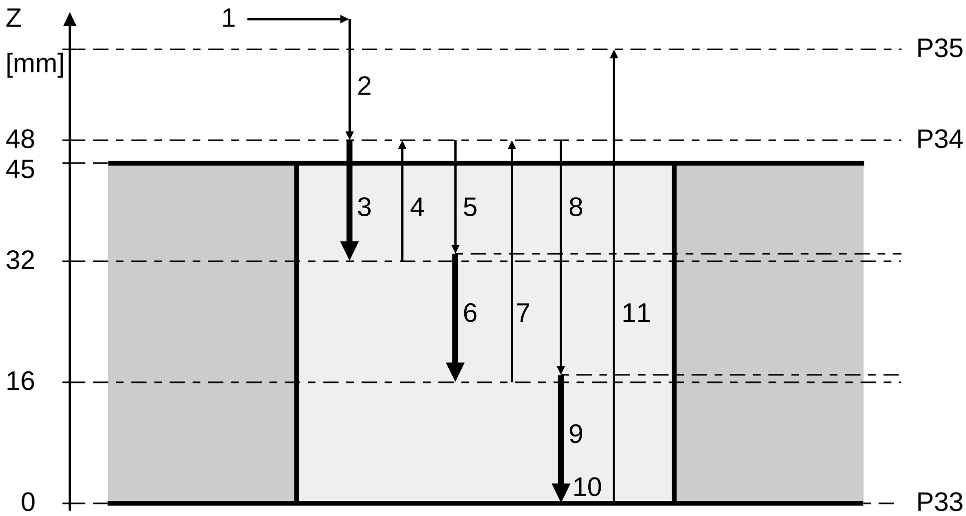


Figure 5-20

Meaning of the parameters

P30	Free cutting time	Turn back time of the spindle after achieving the drilling depth
P32	Drilling feed	
P33	Drilling depth	
P34	Anticipation plane	Safety margin of the processing upper edge
P35	Retreat plane	To go around of obstacles in the setting axis
P36	No. of strokes	constant setting depth = (P33 - P34) / P36
P38	Safety margin	presetting = 1mm
P39	Gradual decrease	0 or - = off

5.3.2 G83 Deep-hole drilling with constant setting depth (continued)

Example

N20 G56 G00 X0 Y0 Z100 S1200 F1500 M03 T01 M16

N30 **G83** Z80 P30:1 P32:800 **P33:0** **P34:48** P35:60 **P36:3** P38:1 P39:0

N40 M05 ...

Sequence plan

1 Switch spindle on (M03, M04) and X and Y axis in rapid traverse position.

2 After achieving the X and Y position the Z-axis runs in rapid traverse on those anticipation plane (P34).

3 After achieving the anticipation plane (P34) the Z axis with working feed moves (P32) to the 1. drilling depth (P34 + ((P33 - P34) * 1 / P36)).

4 After achieving the 1. drilling path the Z axis drives back in rapid traverse up to the anticipation plane.

5 From the anticipation plane the Z axis drives afterwards in rapid traverse up to the 1. drilling depth plus safety margin (P34 + ((P33 - P34) * 1 / P36)) + P38).

6 The Z axis induces itself with working feed (P32) to 2. drilling depth (P34 + ((P33 - P34) * 2 / P36)).

7 After achieving the 2. drilling depth the Z axis drives back in rapid traverse up to the anticipation plane.

8 From the anticipation plane the Z-axis drives afterwards in rapid traverse up to the 2. drilling depth plus safety margin (P34 + ((P33 - P34) * 2 / P36)) + P38).

9 The Z axis induces itself with working feed (P32) to the 3. drilling depth etc. to P33.

10 With the achieving of the drilling depth (P33) a programmed free cutting time (P30) becomes been waiting.

11 Afterwards the Z axis in rapid traverse moves back on the retreat plane (P35) and afterwards the spindle is switched off (M05).

5.3.3 G83 Deep-hole drilling with degressive setting depth

If the parameter P39:1 is programmed (the input value is not determining), the programmed drilling depth (P33) by repeated, reduced setting depth (gradual decrease) of the active drilling depth is achieved.

The active drilling depth is P37-(n-1)*P36, whereby n=1, 2,.. is (nth setting depth = stroke).
If P37-(n-1)*P36 ≤ P36, the active drilling depth = P36 is set.

Before each setting depth checked, whether two setting depths (drilling depth calculated + P36 (gradual decrease measure)) still feasible are. If not, then the new drilling depth calculated by remainder path / 2 and executed the final two setting depths with this value.

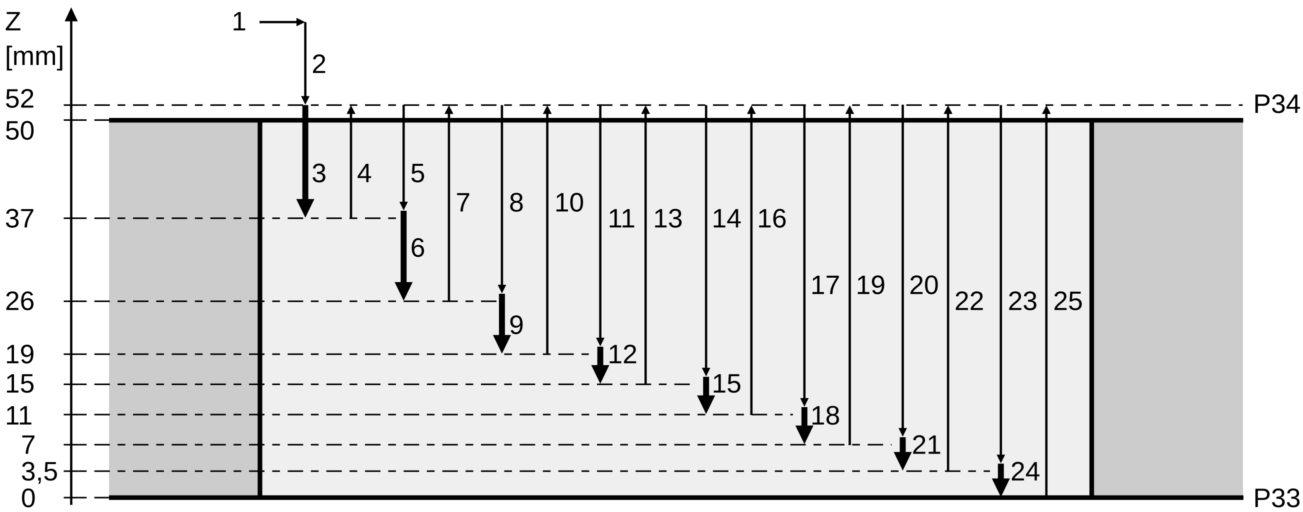


Figure 5-21

Meaning of the parameters

P32	drilling feed	
P33	drilling depth	
P34	anticipation plane	safety margin of the processing upper edge
P35	retreat plane	to the run around of obstacles in the setting axis
P36	setting depth	degressive setting depth
P37	1. stroke	degressive setting depth
P38	safety margin	presetting = 1mm
P39	gradual decrease	degressive setting depth 1 = on

5.3.3 G83 Deep-hole drilling with degressive setting depth (continued)

Example

N20 G56 G00 X0 Y0 Z100 S1200 F1500 M03 T01 M16

N30 **G83** Z80 P32:800 **P33:0** **P34:52** **P36:4** **P37:15** P38:1 **P39:1**

N40 M05 ...

Sequence plan

1 Switch spindle on (M03, M04) and X and Y axis in rapid traverse position.
2 After achieving the X and Y position the Z axis runs on the anticipation plane (P34).

3 After achieving the anticipation plane the Z axis with working feed induces itself to 1. drilling depth (P37) on the Z position: P34 - P37 = 52 - 15 = 37.
4 After achieving 1.drilling depth returns the Z axis in rapid traverse to the anticipation plane.

5 Of the anticipation plane in rapid traverse up to the 1. drilling depth plus safety margin (P38).
6 With working feed to 2. drilling depth on the Z position: 37 - 11 = 26 (11 = 15 - 4).
7 After achieving the 2. drilling depth to the anticipation plane back.

8 Of the anticipation plane in rapid traverse up to 2. drilling depth plus safety margin (P38).
9 With working feed to the 3. drilling depth on the Z position: 26 - 7 = 19 (7 = 11 - 4).
10 After achieving the 3. drilling depth to the anticipation plane back.

11 Of the anticipation plane in rapid traverse up to the 3. drilling depth plus safety margin (P38).
12 With working feed to 4. drilling depth on the Z position: 19 - 4 = 15 (P36:4).
13 After achieving the 4. drilling depth to the anticipation plane back.

14 Of the anticipation plane in rapid traverse up to 4. drilling depth plus safety margin (P38).
15 With working feed to 5. drilling depth on the Z position: 15 - 4 = 11 (P36:4).
16 After achieving the 5. drilling depth to the anticipation plane back.

17 Of the anticipation plane in rapid traverse up to 5. drilling depth plus safety margin (P38).
18 With working feed to 6. drilling depth on the z-position: 11 - 4 = 7 (P36:4).
19 After achieving the 6. drilling depth to the anticipation plane back.

20 Of the anticipation plane in rapid traverse up to 6. drilling depth plus safety margin (P38).
21 With working feed to 7. drilling depth on the Z position: 7 / 2 = 3,5.
22 After achieving the 7. drilling depth to the anticipation plane back.

23 Of the anticipation plane in rapid traverse up to 7. drilling depth plus safety margin (P38).
24 With working feed to 8. drilling depth on the Z position: 3.5 - 3.5 = 0.
25 After achieving 8. drilling depth returns the Z axis in rapid traverse to the anticipation plane.

5.3.4 G84 Thread cutting

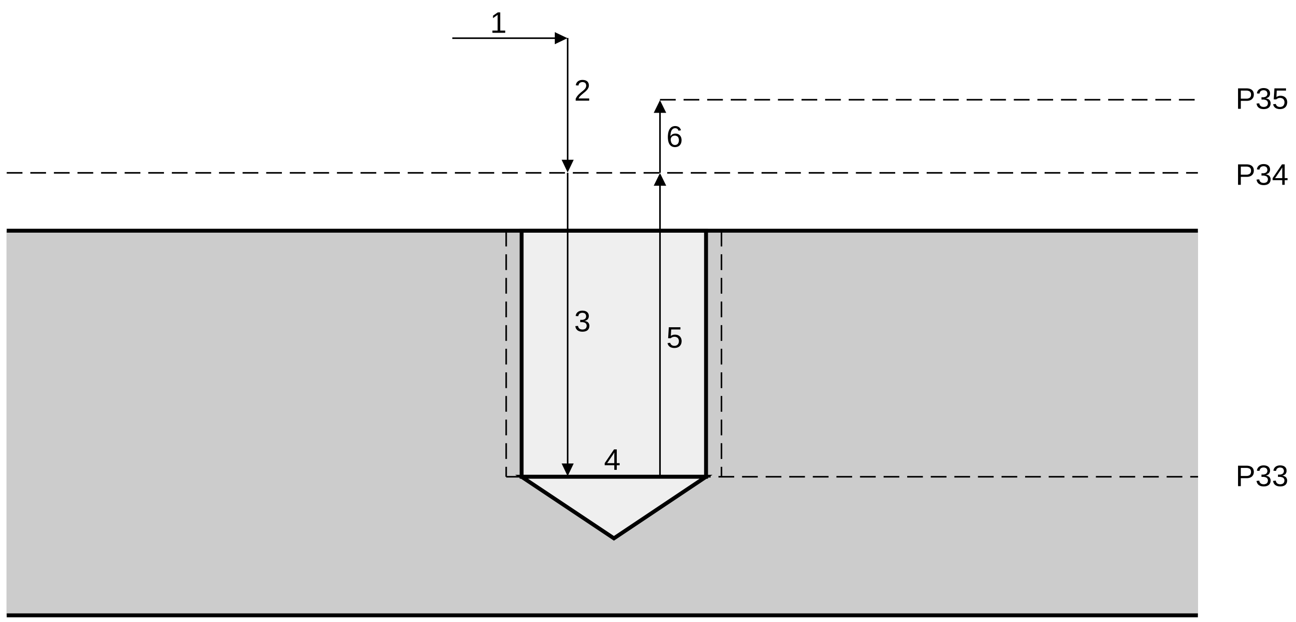


Figure 5-22

Meaning of the parameters

P31	balance fodder	P31: 1	= thread cutting without balance fodder
		P31: 0 or -	= G84 standard cycle (with balance fodder)
P32	thread pitch		
P33	thread depth		
P34	anticipation plane	safety margin of the processing upper edge	
P35	retreat plane	to running around obstacles in the setting axis	

5.3.4 G84 Thread cutting (continued)

Example

N10	...									
N20	G56	G00	X0	Y0	Z100	S1200	F1500	M03	T01	M16
N30	**G84**	Z80	**P32:10**		**P33:20**	**P34:52**	P35:60			
N40	M05	...								

Sequence plan

1 Switch spindle on (M03, M04) and X and Y axis in rapid traverse position.

2 After achieving the X and Y positions the Z axis runs on the anticipation plane.

3 After achieving the anticipation plane the Z axis is joined with the spindle axis and the Z axis moves to the drilling depth (P33).

4 With the achieving of the drilling depth the spindle rotation is turned around and the spindle reversal time (P30) recalled.

5 At process of the retention time the Z axis returns to the anticipation plane (P34).

6 With the achieving of the anticipation plane the Z-axis in rapid traverse becomes on the retreat plane (P35) driven and afterwards the spindle switched off (M05).

5.3.5 G85 Drilling with retraction at feed rate

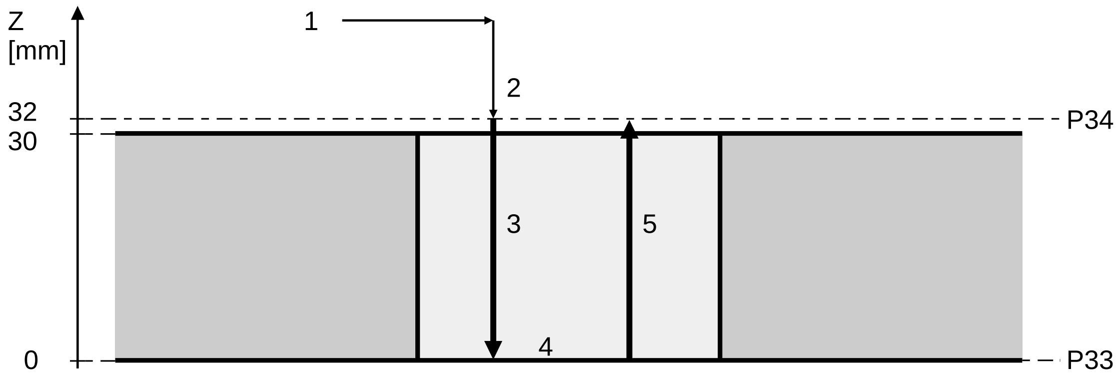


Figure 5-23

Meaning of the parameters

P30	free cutting time	
P32	drilling feed	
P33	drilling depth	
P34	anticipation plane	safety margin of the processing upper edge
P35	retreat plane	to running around obstacles in the setting axis

5.3.5 G85 Drilling with retraction at feed rate (continued)

Example

N10	...									
N20	G56	G00	X0	Y0	Z100	S1200	F1500	M03	T01	M16
N30	**G85**	Z80	P30:1		P32:800	**P33:0**	**P34:32**			
N40	M05	...								

Sequence plan

1 Switch spindle on (M03, M04) and X and Y axis in rapid traverse position.

2 After achieving the X and Y positions the Z axis runs in rapid traverse on those anticipation plane (P34).

3 After achieving the anticipation plane the Z axis with working feed becomes the drilling depth (P33) moves.

4 With the achieving of the drilling depth a programmed free cutting time (P30) is waited for.

5 Afterwards retreat in the working feed to anticipation plane and becomes following spindle switched off (M05).

5.4 G86 / G87 / G88 / G89 cycle patterns

The cycle patterns serve for the repeated version of prozess cycles (boring and milling cycles). Over the programming of certain parameters the points of positioning are described, in which the prozess cycles are to be executed.

The parameters cover the definition of a target as well as the number of points of positioning. For definition of the target and the points of positioning different programming types can be used.

The partitioning of the cycle patterns takes place on the basis of geometrical criteria:

G86 vector type processing
G87 parallelogram type processing
G88 grid type processing
G89 circle type processing

The cycle patterns are block by block effective.

The parameter inputs remain against it modal effectively.

In order to avoid over regulation, all parameters are to be reset, which are not used with a programming type.

A compilation of the parameters is listed on the next side.

5.4 G86 / G87 / G88 / G89 cycle patterns (continued)

Parameter for G86 vector type processing

P100	coordinate of the 1. axis (X)
P101	coordinate of the 2. axis (Y)
P103	vector bracket related to the1. axis (X)
P104	vector length
P105	vector splitting
P106	number of points of positioning

Parameter for G87 periphery processing of a parallelogram

Vector 1	Vector2	
P100	P110	coordinate of the 1. axis (X)
P101	P111	coordinate of the 2. axis (Y)
P103	P113	vector bracket related to the 1. axis (X)
P104	P114	vector length
P105	P115	vector splitting
P106	P116	number of points of positioning

Parameter for G88 grid processing

Vector 1	Vector 2	
P100	P110	coordinate of the 1. Axis @@@(x)
P101	P111	coordinate of the 2. Axis @@@(y)
P103	P113	vector bracket related to the @@@1.Achse (x)
P104	P114	vector length
P105	P115	vector splitting
P106	P116	number of points of positioning

Parameter for G89 circle processing

P120	coordinate of the 1. axis (X) of the pitch diameter focal point
P121	coordinate of the 2. axis (Y) of the pitch diameter focal point
P122	circle diameter
P123	start bracket
P124	travel bracket
P125	vector part
P126	number of points of positioning

5.4 G86 / G87 / G88 G89 cycle patterns (continued)

Messages

Messages to the cycles have numbers, which start with 8xxx.
A listing is to be found in the paragraph ‘ 9,9 messages of cycles ‘.

Messages appear, if

- the target with a programming type by programming of additional parameters multiple is defined and no agreement is achieved.

- the target is not sufficiently determined.

In these cases the program is interrupted.

Therefore all parameters are to be reset, which are not used with a programming type for the definition of the target.

5.4 G86 / G87 / G88 G89 cycle patterns (continued)

Call of prozess cycles with cycle patterns

Together with a cycle pattern call a process cycle (boring or milling cycle) can be activated, which is then called after each positioning step in the cycle pattern.

Effectiveness of M functions

Pre path M functions are executed in the starting point A, after path M functions in the target B.

M functions, which are effective within cycles (e.g. M03, M05,..), become for each cycle call pre path or after path actively.

Start the points of positioning

Starting the respective points of positioning effected with G00. Contains the theorem a cycle call, in each point of positioning the cycle is executed. Otherwise an accurate stop (G08) is activated

5.4.1 Vector type processing

Start point and target point with G86

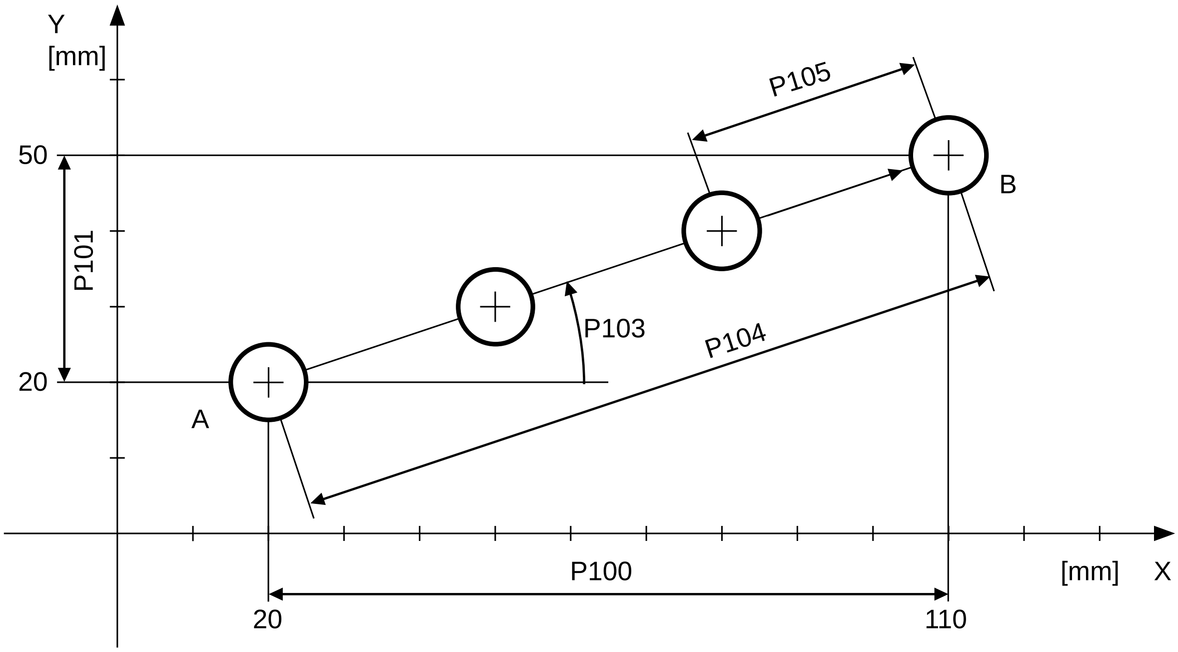


Figure 5-24A = Start point, B = Target point

Meaning of the parameters

P100 AB coordinate of the 1. axis (X)
P101 AB coordinate of the 2. axis (Y)
P103 vector bracket related to the 1. axis (X)
P104 vector length
P105 vector splitting
P106 number of positioning points

5.4.1 Vector type processing (continued)

Example 1: Programming with AB coordinates P100 and P101 and number of points of positioning

N10	G00	**X20**	**Y20**	Z100	S500	M03	T01	M16
N20	**P100:90**	**P101:30**	P103:–	P104:–	P105:–	**P106:4**		
N30	P32:800	P33:0	P34:32					
N40	**G86**	G81						

Example 2: Programming with vector brackets P103 and vector length P104 and number of points of positioning

N10	G00	**X20**	**Y20**	Z100	S500	M03	T01	M16
N20	P100:–	P101:–	**P103:20**	**P104:96**	P105:–	**P106:4**		
N30	P32:800	P33:0	P34:32					
N40	**G86**	G81						

Example 3: Programming with vector brackets P103 and vector splitting P105 and number of points of positioning

N10	G00	**X20**	**Y20**	Z100	S500	M03	T01	M16
N20	P100:–	P101:–	**P103:20**	P104:–	**P105:32**	**P106:4**		
N30	P32:800	P33:0	P34:32					
N40	**G86**	G81						

5.4.1 Vector type processing (continued)

Process

N10	programming of the start point with X and Y
N20	programming of the target point with the parameters P100 and P101 or P103 and P104 or P103 and P105
N30	programming of the parameters of the process cycle
N40	calls of the cycle pattern and the process cycle

Sequence of the operation: position first and then process.

Note: not used parameters are to be reset.

5.4.1 Vector type processing (continued)

Programming the start point A

The start point is established by programming the X, Y.. coordinates. If these coordinates have not been programmed the machine stop location coordinates are used as the starting coordinates.

Programming the target point B

Programming mode 1

The target point B is established by programming the X (AB) P100 and Y (AB) P101 coordinates.

Bx = Ax + X (AB)
By = Ay + Y (AB)

Programming mode 2

The target point B is established by programming the angle P103 and the length of vector P104.

X (AB) = P104 * cosP103
Y (AB) = P104 * sinP103

Programming mode 3

The target point B is established by programming the angle P103, the vector part P105 and the number of positioning points P106.

For P106 equal to or greater than 2 the following apllies:

X (AB) = P105 * (P106-1) * cosP103
Y (AB) = P105 * (P106-1) * sinP103

For P106=0 and P106=1:

X (AB) = P105 * P106 * cosP103
Y (AB) = P105 * P106 * sinP103

5.4.1 Vector type processing (continued)

Data analysis

Angel P103

≥ 0°: angle in positive mathematical sense (left-handed rotation) referred on the positive vector of the 1. axis (X)

< 0°: angle in negative mathematical sense (right-handed rotation) referred on the positive vector of the 1. axis (X)

≥ 360°: reduction of the angle on smaller 360°

Influence of the planes that are switched on

The sizes X and Y and the angle P103 are plane-oriented.

Plane	Axis	P103 referred on the positive vector of the axis
G17 (XY)	X Y	X
G18 (ZX)	Z X	Z
G19 (YZ)	Y Z	Y

Vector length P104 and vector part P105

Negative values are changed into positive values without indication.

5.4.1 Vector type processing (continued)

Number of positioning points P106

Negative values are changed into positive values.
Non-integer values are round down to the next smaller integer value.

Number of positioning points P106=0
The target point B (from A to B) is approached directly; eventually activated cycles are not executed.

Number of positioning points P106=1
The target point B (from A to B) is approached directly and an eventually activated cycle is executed in the target point.

Number of positioning points P106 and vector part P105

Further identifications for the programming modes 1 and 2 could be the number of positioning points P106 or the vector part P105.

When indicating the vector part P105, the number of positioning points is determined:

P106 = (AB / P105) + 1

After this, the vector part is calculated:

P105' = AB / (P106 - 1)

If the calculated number N is an integer number:

P105' = P105

If the calculated number N is not an integer number:

P105' unequal to P105

The positioning points is executed with the value T'.

5.4.2 G87 Parallelogram type processing

Start and target point with G87

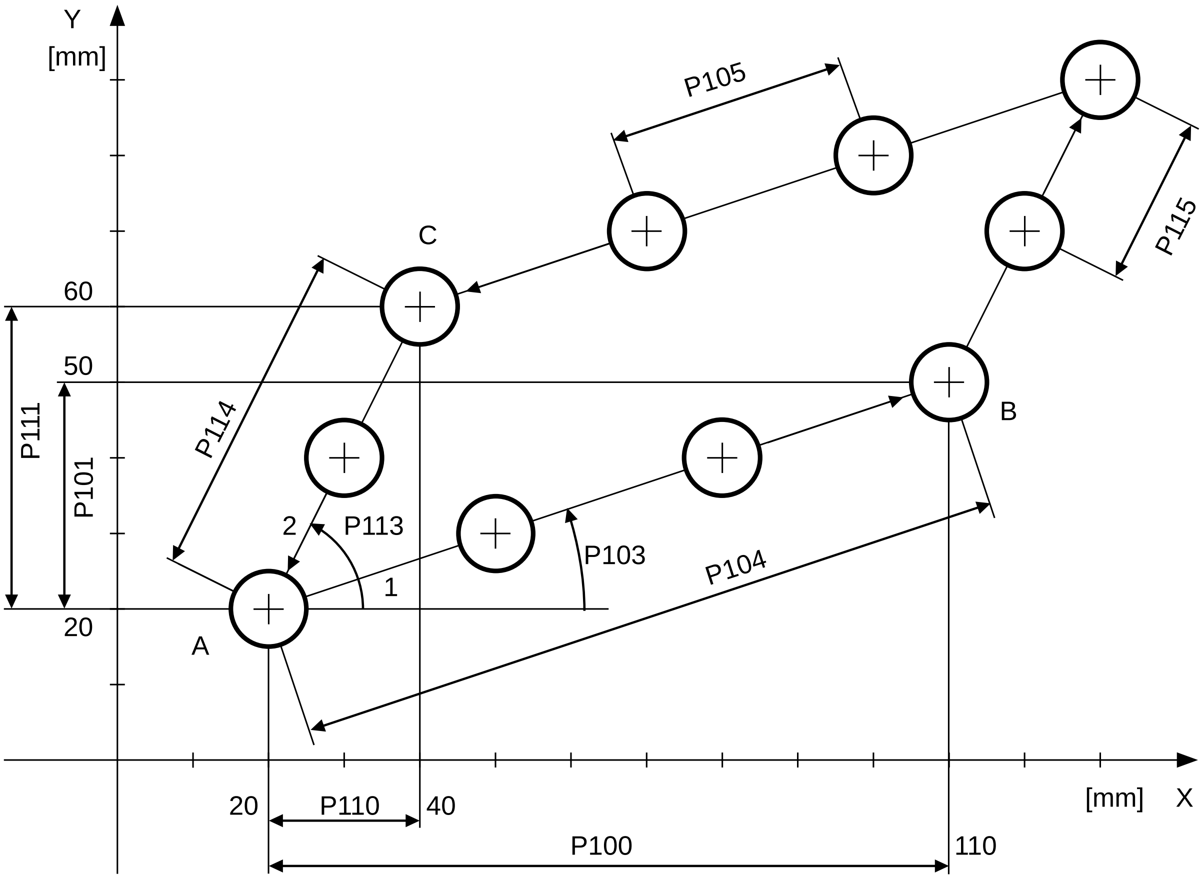


Figure 5-25

Meaning of the parameters

Vector 1	Vector 2	
P100	P110	coordinate of the 1. axis (X)
P101	P111	coordinate of the 2. axis (Y)
P103	P113	ector bracket related to the 1. axis (X)
P104	P114	vector length
P105	P115	vector part
P106	P116	number of positioning points

5.4.2 G87 Parallelogram type processing (continued)

Example 1: **Programming with AB coordinates P100, P101 and P110, P111 and number of positioning points**

N10	G00	**X20**	**Y20**	Z100	S500	M03	T01	M16
N20	**P100:90**	**P101:30**	**P110:20**	**P111:40**	P103:–	P113:–	P104:–	
	P114:–	P105:–	P115:–	**P106:4**	**P116:3**			
N30	P32:800	P33:0	P34:32					
N40	**G87**	G81						

Example 2: **Programming with vector brackets P103, P113 and vector length P104, P114 and number of positioning points**

N10	G00	**X20**	**Y20**	Z100	S500	M03	T01	M16
N20	P100:–	P101:–	P110:–	P111:–	**P103:20**	**P113:60**	**P104:96**	
	P114:46	P105:–	P115:–	**P106:4**	**P116:3**			
N30	P32:800	P33:0	P34:32					
N40	**G87**	G81						

Example 3: **Programming with vector brackets P103, P113 and vector splitting P105, P115 and number of positioning points**

N10	G00	**X20**	**Y20**	Z100	S500	M03	T01	M16
N20	P100:–	P101:–	P110:–	P111:–	**P103:20**	**P113:60**	P104:–	
	P114:–	**P105:32**	**P115:23**	**P106:4**	**P116:3**			
N30	P32:800	P33:0	P34:32					
N40	**G87**	G81						

5.4.2 G87 Parallelogram type processing (continued)

Process

N10 programming of the start point with X and Y

N20 programming of the target point with the parameters
P100, P110 and P101, P111 or P103, P113 and P104, P114 or P103, P113 and P105, P115

N30 programming of the parameters of the process cycle

N40 calls of the cycle pattern and the process cycle

Sequence of the operation:	first position and then process.
Sequence of processing:	for and return trip always first vector 1 and then vector 2

Note: not used parameters are to be reset

5.4.2 G87 Parallelogram type processing (continued)

Programming of the start point A

The start point A is determined by programming the coordinates X, Y..
If the coordinates are not programmed, the coordinates of the machine location are used as start coordinates.

Programming of the target points B and C

Programming mode 1

The target points B and C are determined by programming the coordinates X (AB) P100 / Y (AB) P101 and X (AC) P110 / Y (AC) P111.

Target point B	Bx = Ax + X(AB) By = Ay + Y(AB)	Target point C	Cx = Ax + X(AC) Cy = Ay + Y(AC)

Programming mode 2

The target points B and C are determined by programming the angles P103 / P113 and vector length P104 / P114.

Target point B	X(AB) = P104 * cosP103 Y(AB) = P104 * sinP103	Target point C	X(AC) = P114 * cosP113 Y(AC) = P114 * sinP113

Programming mode 3

The target points B and C are determined by programming the angles P103 / P113, the vector parts P105 / P115 and number of positioning points P106 / P116.

For P106 / P116 >= 2:

Point B	X(AB) = P105 * (P106-1) * cosP103 Y(AB) = P105 * (P106-1) * sinP103	Point C	X(AC) = P115 * (P116-1) * cosP113 Y(AC) = P115 * (P116-1) * sinP113

For P106 / P116 = 0 and P106 / P116 = 1 :

Point B	X(AB) = P105 * P106 * cosP103 Y(AB) = P105 * P106 * sinP103	Point C	X(AC) = P115 * P116 * cosP113 Y(AC) = P115 * P116 * sinP113

5.4.2 G87 Parallelogram type processing (continued)

Data evaluation

Angle P103, P113

≥ 0°:	angle in positive mathematical sense (left-handed rotation) referred on the positive vector of the x-axis
< 0°:	angle in negative mathematical sense (right-handed rotation) referred on the positive vector of the x-axis
≥ 360°:	reduction of the angle on smaller 360°

Influence of the plane that is switched on:

The sizes X and Y and the angle P103, P113 are plane-oriented.

Plane	Axis	P103, P113 referred on the positive vector of the axis
G17 (XY)	X Y	X
G18 (ZX)	Z X	Z
G19 (YZ)	Y Z	Y

Vector length P104, P114 and vector part P105, P115

Negative values are changed into positive values without indication.

5.4.2 G87 Parallelogram type processing (continued)

Number of positioning points P106, P116

Negative values are changed into positive values.
Non-integer values are round down to the next smaller integer value.

Number of positioning points P106, P116 = 0
The target point B / C (from A to B / C) are approached directly; eventually activated cycles are not executed.

Number of positioning points P106, P116 = 1
The target points B / C (from A to B / C) are approached directly and an eventually activated cycle is executed in the target point.

Number of positioning points P106, P116 and vector part P105, P115

Further identifications for the programming modes 1 and 2 could be the number of drilling holes P106, P116 or the vector part P105, P115.

When indicating the vector part P105, P115, the number of drilling holes is determined:

P106, P116 = (AB / P105, P115) + 1

After this, the vector part is calculated:

P105', P115' = AB / (P106, P116 - 1)

If the calculated number P106, P116 is an integer number:

P105', P115' = P105, P115

If the calculated number P106, P116 is not an integer number:

P105', P115' unequal to P105, P115

The drilling pattern is executed with the value P105', P115'.

5.4.3 G88 Grid type processing

Start and target point at G88

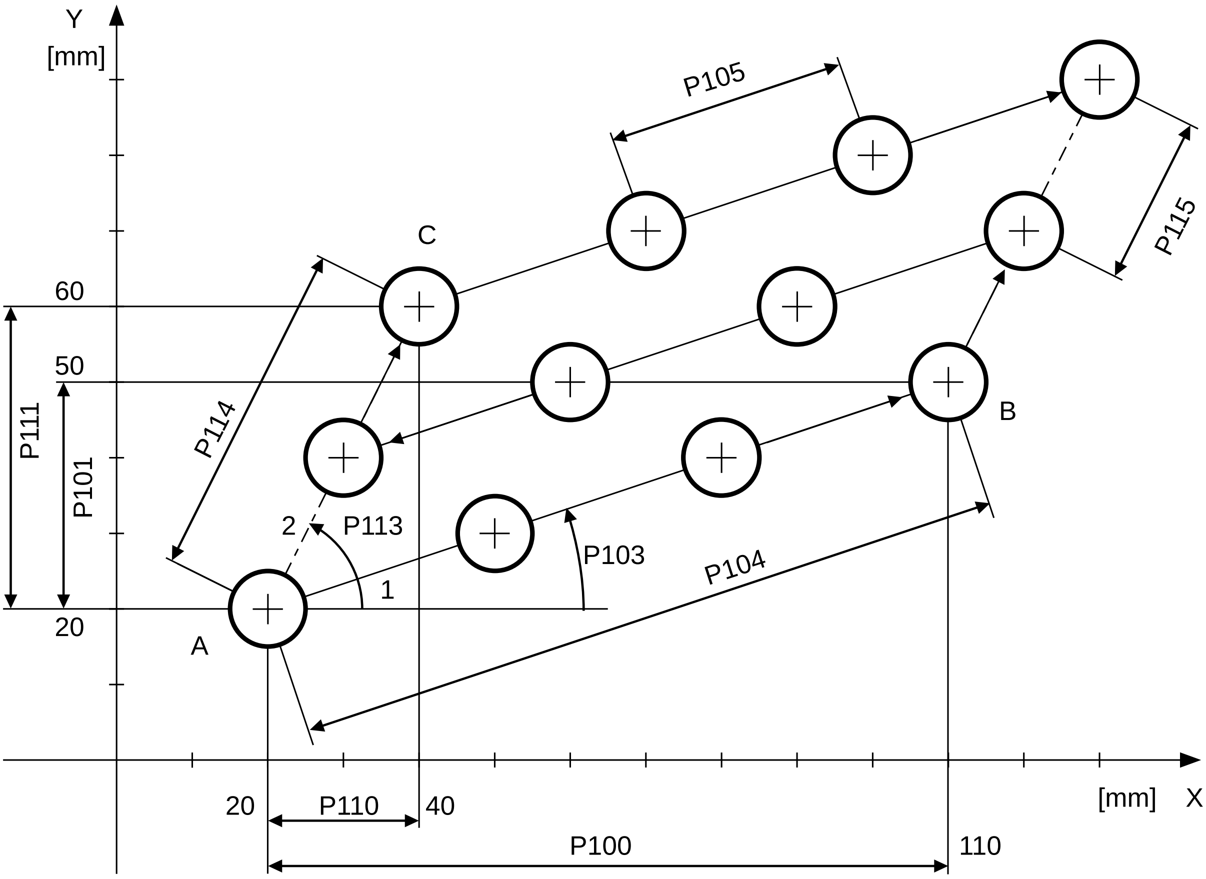


Figure 5-26

Meaning of the parameters

Vector 1	Vector 2	
P100	P110	coordinate of the 1. axis (X)
P101	P111	coordinate of the 2. axis (Y)
P103	P113	ector bracket related to the 1. axis (X)
P104	P114	vector length
P105	P115	vector part
P106	P116	number of positioning points

5.4.3 G88 Grid type processing (continued)

Example 1: **Programming with AB coordinates P100, P110 und P101, P111 and number of positioning points**

N10	G00	**X20**	**Y20**	Z100	S500	M03	T01	M16
N20	**P100:90**	**P101:30**	**P110:20**	**P111:40**	P103:–	P113:–	P104:–	
	P114:–	P105:–	P115:–	**P106:4**	**P116:3**			
N30	P32:800	P33:0	P34:32					
N40	**G88**	G81						

Example 2: **Programming with vector brackets P103, P113 and vector length P104, P114 and number of positioning points**

N10	G00	**X20**	**Y20**	Z100	S500	M03	T01	M16
N20	P100:–	P101:–	P110:–	P111:–	**P103:20**	**P113:60**	**P104:96**	
	P114:46	P105:–	P115:–	**P106:4**	**P116:3**			
N30	P32:800	P33:0	P34:32					
N40	**G88**	G81						

Example 3: **Programming with vector brackets P103, P113 and vector splitting P105, P115 and number of positioning points**

N10	G00	**X20**	**Y20**	Z100	S500	M03	T01	M16
N20	P100:–	P101:–	P110:–	P111:–	**P103:20**	**P113:60**	P104:–	
	P114:–	**P105:32**	**P115:23**	**P106:4**	**P116:3**			
N30	P32:800	P33:0	P34:32					
N40	**G88**	G81						

5.4.3 G88 Grid type processing (continued)

Process

N10 programming of the start point with X and Y

N20 programming of the target point with the parameters
P100, P110 and P101, P111 or P103, P113 and P104, P114 or P103, P113 and P105, P115

N30 programming of the parameters of the process cycle

N40 calls of the cycle pattern and the process cycle

Sequence of the operation: first position and then process.

Sequence of processing: first with journey there first vector 1 completely
and afterwards the first position on vector 2,

then with return trip vector 1 completely
and afterwards the next position on vector 2,

then again with journey there vector 1 completely etc.
until the final position on vector 2 achieves
and the processing of vector 1 is executed

thereafter the initial position is started.

Note: not used parameters are to be reset

5.4.3 G88 Grid type processing (continued)

Programming the start point A

The start point A is determined by programming the coordinates X, Y...
If the coordinates are not programmed, the coordinates of the machine location are used as starting coordinates.

Programming of the target points B and C

Programming mode 1

The target points B and C are determined by programming the coordinates X (AB) P100, Y (AB) P101 and X (AC) P110, Y (AC) P111.

Target point B	Bx = Ax + X(AB) By = Ay + Y(AB)	Target point C	Cx = Ax + X(AC) Cy = Ay + Y(AC)

Programming mode 2

The target points B and C are determined by programming the angles P103, P113 and vector length P104, P114.

Target point B	X(AB) = P104 * cosP103 Y(AB) = P104 * sinP103	Target point C	X(AC) = P114 * cosP113 Y(AC) = P114 * sinP113

Programming mode 3

The target points B and C are determined by programming the angles E1 / E2, the vector parts T1 / T2 and number of drilling holes N1 / N2.

For N >= 2:

Point B	X(AB) = P105 * (P106-1) * cosP103 Y(AB) = P105 * (P106-1) * sinP103	Point C	X(AC) = P115 * (P116-1) * cosP113 Y(AC) = P115 * (P116-1) * sinP113

For P106, P116 = 0 and P106, P116 = 1:

Point B	X(AB) = P105 * P106 * cosP103 Y(AB) = P105 * P106 * sinP103	Point C	X(AC) = P115 * P116 * cosP113 Y(AC) = P115 * P116 * sinP113

5.4.3 G88 Grid type processing (continued)

Data evaluation

Angle P103, P113

≥ 0°: angle in positive mathematical sense (left-handed rotation) referred on the positive vector of the x-axis

< 0°: angle in negative mathematical sense (right-handed rotation) referred on the positive vector of the x-axis

≥ 360°: reduction of the angle on smaller 360°

Influence of the plane that is switched on:

The sizes X and Y and the angle P103, P113 are plane-oriented.

Plane	Axis	P103, P113 referred on the positive vector of the axis
G17 (XY)	X Y	X
G18 (ZX)	Z X	Z
G19 (YZ)	Y Z	Y

Vector length P104, P114 and vector part P105, P115

Negative values are changed into positive values without indication.

5.4.3 G88 Grid type processing (continued)

Number of drilling holes P106, P116

Negative values are changed into positive values.
Non-integer values are round down to the next smaller integer value.

Number of drilling holes P106, P116 = 0
The target points B / C (from A to B / C) are approached directly; eventually activated cycles are not executed.

Number of drilling holes P106, P116 = 1
The target points B / C (from A to B / C) are approached directly and an eventually activated cycle is executed in the target point.

Number of drilling holes P106, P116 and vector part P105, P115

Further identifications for the programming modes 1 and 2 could be the number of drilling holes P106, P116 or the vector part P105, P115.

When indicating the vector part P105, P115, the number of drilling holes is determined:

P106, P116 = (AB / P105, P115) + 1

After this, the vector part P105, P115 is calculated:

P105', P115' = AB / (P106, P116 - 1)

If the calculated number P106, P116 is an integer number:

P105', P115' = P105, P115

If the calculated number P106, P116 is not an integer number:

P105', P115' unequal to P105, P115

The drilling pattern is executed with the value P105', P115'.

5.4.4 G89 Circle type processing

Start and target point at G88

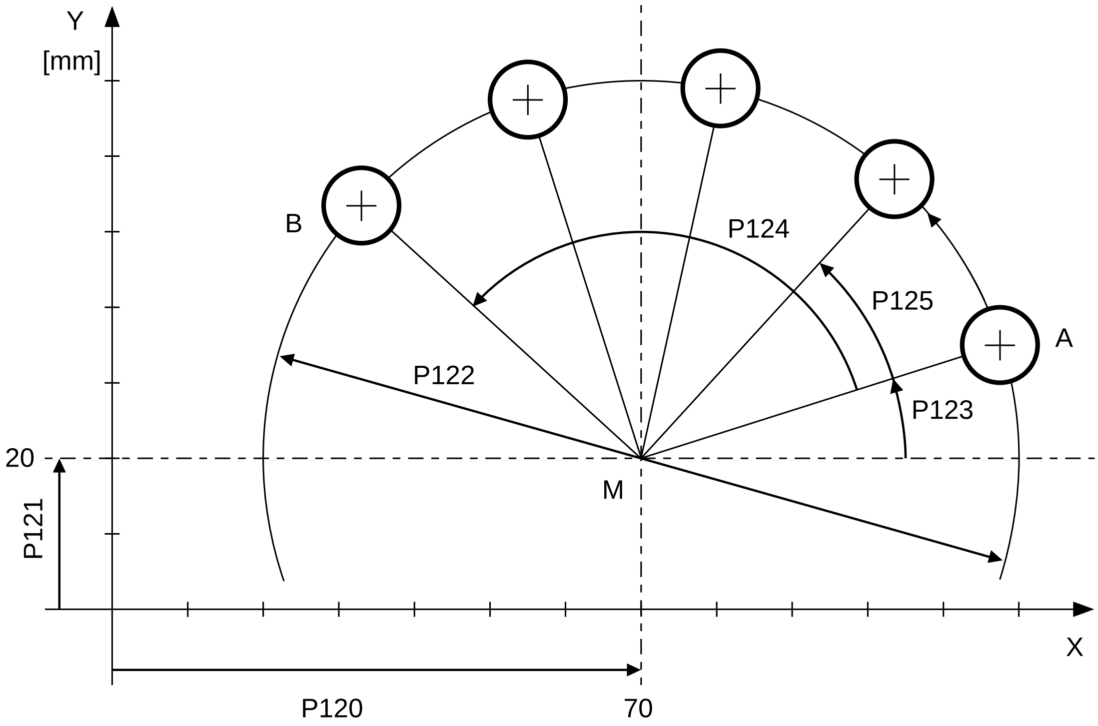


Figure 5-27

Meaning of the parameters

P120	Coordinate of the 1. axis (X) of the positioning points
P121	Coordinate of the 2. axis (Y) of the positioning points
P122	Pitch circle diameter
P123	Start angle
P124	Travel angle
P125	Vector part
P126	Number of positioning points

5.4.4 G89 Circle type processing (continued)

Example 1: **Programming with centre coordinates P120, P121, circle diameter P122, start brackets P123 and travel bracket P124 and number of positioning points**

N10	G00	Z100	S500				
	M03	T01	M16				
N20	**P120:70**	**P121:20**	**P122:100**	**P123:18**	**P124:120**	P125:--	**P126:5**
N30	P32:800	P33:0	P34:32				
N40	**G89**	G81					

Example 2: **Programming with centre coordinates P120, P121, circle diameter P122, start brackets P123 and vector part P125 and number of positioning points**

N10	G00	Z100	S500				
	M03	T01	M16				
N20	**P120:70**	**P121:20**	**P122:100**	**P123:18**	P124:--	**P125:30**	P126:--
N30	P32:800	P33:0	P34:32				
N40	**G89**	G81					

5.4.4 G89 Circle type processing (continued)

process

N10 programming of the start point with P120, P121, P122 and P123

N20 programming of the target point with the parameters
P120, P121, P122, P123 and P124 or P120, P121, P122, P123 and P125

N30 programming of the parameters of the process cycle

N40 calls of the cycle pattern and the process cycle

Sequence of the operation: first position and then process.

Sequence of processing: dependent on the programmed type of interpolation G00, G01 or G02, G03 become the particulars positions straight linear or circular interpolates started.

Note: not used parameters are to be reset

5.4.4 G89 Circle type processing (continued)

Programming the start point A

The starting point A always has to be determined by programming identifications at G89,
i.e. the pitch circle centre point P120, P121, the pitch circle radius P122/2 and the starting angle P123.

Programming the target point B

Programming mode 1

The target point B is determined by programming the travel angle P124.
Bx = P120 + (P122/2) * cos(P123+P124)
By = P121 + (P122/2) * sin(P123+P124)

Programming mode 2

The target point is determined by programming the number of positioning points P126 and the vector part P125.
Bx = P120 + (P122/2) * cos(P123 + (P126-1) * P125)
By = P121 + (P122/2) * sin(P123 + (P126-1) * P125)

P124 = (P126-1) * P125

5.4.4 G89 Circle type processing (continued)

Data evaluation

Angle P123

≥ 0°: angle in positive mathematical sense (left-handed rotation) referred on the positive vector of the x-axis

< 0°: angle in negative mathematical sense (right-handed rotation) referred on the positive vector of the x-axis

≥ 360°: reduction of the angle on smaller 360°

Influence of the plane that is switched on:

The sizes X and Y and the angle P123 are plane-oriented.

Plane	Axis	P123 referred on the positive vector of the axis
G17 (XY)	X Y	X
G18 (ZX)	Z X	Z
G19 (YZ)	Y Z	Y

Vector part P125

Negative values are changed into positive values without indication.

5.4.4 G89 Circle type processing (continued)

Number of positioning points P126

Negative values are changed into positive values.
Non-integer values are round down to the next smaller integer value.

Number of positioning points P126 = 0
The target point B (from A to B) is approached directly; eventually activated cycles are not executed.

Number of positioning points P126 = 1
The target point B (from A to B) is approached directly and an eventually activated cycle is executed in the target point.

Number of positioning points P126 and vector part P125

Further identifications for the programming modes 1 and 2 could be the number of positioning points P126 or the vector part P125.

When indicating the vector part P125, the number of positioning points P126 is determined:

P126 = (P124 / P125) + 1

After this, the vector part is calculated:

P125' = P124 / (P126-1)

If the calculated number P126 is an integer number:

P125' = P125

If the calculated number N is not an integer number:

P125' unequal to P125

The drilling pattern is executed with the value P125'.

5.4.4 G89 Circle type processing (continued)

Starting the points of positioning (G89)

Starting of the points of positioning depends on the programmed type of interpolation.

G00 or G01 actively

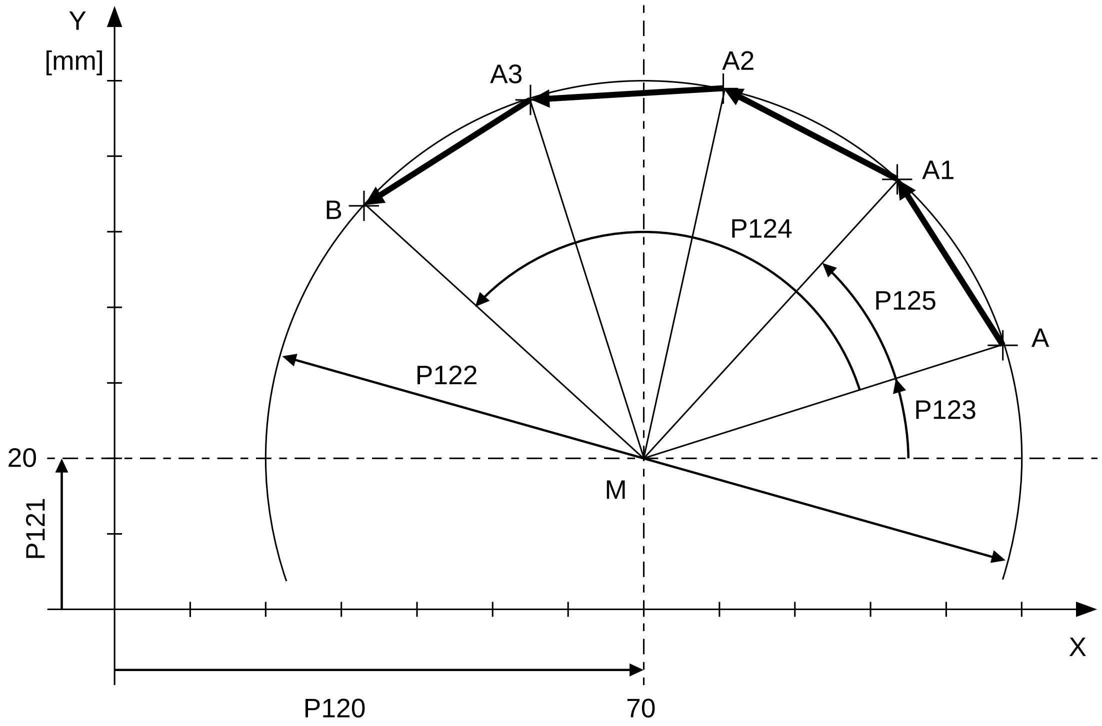


Figure 5-28

The positions (A1, A2, A3 and B) are started with G00.
The sign of the travel bracket P124 or, if this does not indicated, the sign of the vector part P125 determines, whether the circle is anti-clockwise rotation or is clockwise rotation processed:

Positive sign of the bracket:	anti-clockwise rotation,
Minus sign of the bracket:	clockwise rotation.

5.4.4 G89 Circle type processing (continued)

G02 or G03 actively

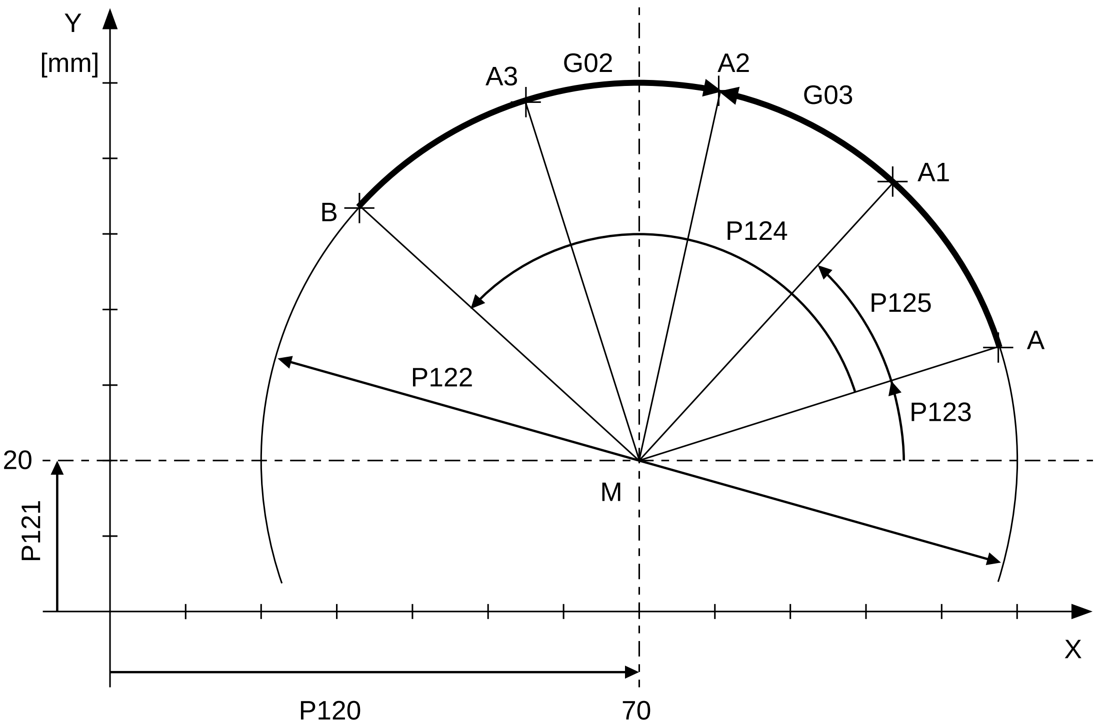


Figure 5-29

The positioning points are started with G02 or G03 and max. programmable feed speed.

Two cases are to be differentiated:

The travel bracket is positive (anti-clockwise rotating)

With G03 also the positioning direction is positive.
With G02 the positioning direction is negative (moving in opposite directions to the travel bracket).

The travel bracket is negative (clockwise rotatingly)

With G02 also the positioning direction is positive.
With G03 the positioning direction is negative (moving in opposite directions to the travel bracket).

5.5 Customer specific cycles

A cycle is a program that generates certain repeated sequences.

For user-specific cycles, these sequences are defined in a Z-program.

Example:

```
Z1001
N10  ...
.
.
N40  P500 : P500+1
.
.
N60  M30
```

The defined cycle can be started then in a machining program with a G-function.
The G-function must have the same number as the Z-program.

Example:

```
P2000

N10  ...
.
.
N60  P500 : 2    G1001
.
.
N90  M30
```

The G-function numbers can be selected arbitrarily except those numbers that are already reserved for other functions.

Input variables can be transmitted to the Z-program with parameters. The parameter numbers P500 to P4999 are free for the user. P0 to P499 is reserved for BWO-cycles.
If the BWO-cycles are not used, they are also free for the user.

Addtional functions to the operating system
Additional functions could be implemented with a DLL (Dynamic Link Library) that is written in programming language C.

6. Supplementary functions

6.1 Program controlling supplementary functions

M00 Measuring stop

M01 Synchronization

M02 Sub-routine skip back

M06 / M16 Tool functions

M23 Unconditional skip with statement of block number

M24 / M25 Program loops

M26 Supervision ‘clear remaining travel’

M28 Unconditional skip in sub-routine

M29 End of program with program repetition

M30 End of program

If parameter skips are programmed in the NC block, they are treated with priority!

6.1.1 M00 Measuring stop

Program run is shut down and spindle stop becomes effective after completion of the block in which M00 is programmed. Automatic mode is possible after Auto-Stop.

With pressing the key 'Start' the NC program continues and the spindle is switched on.

Example: X37,95 Y12,76 M00

After approaching the coordinates X and Y the program sequence is interrupted.

The function operates block by block.

M00 may only be programmed when the tool-radius-contour correction (G40) is switched off.

6.1.2 M01 Synchronization

The function M01 causes a synchronization of program analysis and real time machining.
M01 operates block by block and is executed after the path machining.

When processing a program the block analysis of the real time runs some blocks ahead.

Example:

```
N10  G01  F1000      X10  Y10
N20                  X20
N30                  X100
N40                  X50
N50  P100:20
```

In this NC program the axes traverse in the current block N10, while the block analysis is already executing the parameter instruction of block N50.

This running ahead of the block analysis can be stopped through programming M01. IF M01 is programmed in a block, the block analysis at the block end is waiting until this block is really worked.

6.1.3 M02 Sub-routine skip back

After processing the block in which M02 is programmed, a skip back into the calling programm is executed in a sub-routine in the calling program (after traverses, without stop).

6.1.4 M06 / M16 Tool functions

M06 Tool change

M16 Tool data call

see paragraph 7.1 Tool functions

6.1.5 M23 Unconditional skip

With M23 the program is continued at the indicated program - and block number.

The function operates block-by-block.

The skip is programmed with M23 and indicating program - and block number, e.g.

M23.110 skip to block 110 or

M23.10.110 skip in program 10 to block 110 .

6.1.6 M24 / M25 Program loop

M24 Program loop start

With loop programming program parts of the same kind can be repeated.

The program loop start is programmed with M24 and indicating the runs, e.g. M24.07 (7 runs of the loop). The loop number is to be found in P8840.

It has to be considered, that the block, in which the program loop stard is to be found (M24.. .), does not belong to the program loop.

The function M24 operates block-by-block.

M25 Program loop end

This function marks the end the program loop. After processing the block in which M25 is programmed, the program skips back to the loop start. If all runs are worked, the program is continued with the block following on M25.

The function operates block-by-block.

6.1.7 M26 Supervision ‘Clear remaining travel’

The supervision ‘clear remaining travel’ is active.

The function operates block-by-block and effective before axes movements.

From M26 on the interpolator supervises the signal ‘clear remaining travel’ from the PLC and executes the command. If the signal lines up already, the travel is cleared immediately.

6.1.8 M28 Unconditional skip in sub-routine

Program parts, which are repeated in a program, can be written as sub-routine. Also each arbitrary program can be called up as sub-routine, e.g.

M28.300 starts program 300 .

The control remembers the skip address and continues the calling program at the block following on M28, if the sub-routine is finished.

The sub-routine skip back can be programmed with M02.

A nesting of the sub-routines is possible four times.

The function M28 operates block-by-block.

6.1.9 M29 End of program and program repetition

After processing the block in which M29 is programmed, the program is finished, executed a skip at the beginning of the program and the program is restarted automatically.
The preset G - functions become effective again.

IF M29 stands at the end of a program, which was called up as sub-routine, a sub-routine skip back is executed.

The function operates block-by-block.

6.1.10 M30 End of program and skip to program start

After processing the block in which M30 is programmed, the program is finished and a skip to the beginning of the program is executed (after traverse, with stop).
The preset G - functions become effective again. Besides that Spindle Stop and Coolants off become active.

The function operates block-by-block.

6.2 Machine controlling supplementary functions

Area : 0 .. 999

There may be programmed 8 M-functions per NC block.

The M-functions can be defined as
- before traverse
- after traverse
- with stop
- with skip

(Skip-M-functions are always before traverse with stop, see q1050 .. 1099.)

At M-functions with skip, the skip target can exist either only of the block number or of program number and block number

e.g.:

M41.2.10	If M-function confirmed, skip to program 2 block number 10
M41.10	If M-function confirmed, skip to block number 10

If the M-function is confirmed, it is skiped to the indicated block number and/or in the indicated program with indicated block number.
At a skip impending ways are deleted and the NC block buffer is cleared.

With no confirmation the NC block interpreter switches over to the next NC block.

7. Tool

7.1 Tool functions

General to the tool data

- The tool length in the offsetting record is always taken into consideration in autooperation (according to G17 / G18 / G19 in 3. defined axis).

- In the case of NC program end or NC program abort the spindle tool data becomes theorem into the offsetting record copies.
  Thus tool offset compensations of the NC program (M16) become ineffective.
  When starting of the next NC program is immediately the length of the spindle tool actively (offsetting data).

- The tool length works as shift.
  The indicator position is the programmed position from the NC program.
  The tool length adds itself in the sum shift parameter (P12155).

- The who things radius works as correction.
  The indicator position is smaller or larger than the programmed position from that NC program.

7.1 Tool functions (continued)

T Selection of the group of tools in the NC program

The tool number T is identified and treated with in the analysis. In the tool data array a browsing is started after the suitable group of tools.
When finding the programmed group of tools this record is copied into the PLC tool data array (P8050) and transferred thus to the PLC.

A transfer to the PLC finds however only,

- the first T call in the NC program or
- a modification of the T number.

Special case
If a tool is called, which already is in the spindle (P8100), then the PLC tool data array is filled not from the tool data pool, but the spindle tool record is only copied into the PLC tool record.

Example N10 T5

T extension
For special customer adaptations the T function permits a covered subroutine reference. With each analyzed T, the NC memory is scanned for a Z5. If Z5 exists, then this cycle is covered called.
With the call of Z5 the PLC Interface (P8050...) with the new data is described.

7.1 Tool functions (continued)

Tool definitions t...

Existed in the system no tool tools, then can over the identifier, ‘t:’, ‘tr:’, ‘tq:’, ‘tl:’ tool data to be set, with which a radius correction (G41/42) activated afterwards operates. These tool definition can be combined at will in a NC block.
With all functions the tool offsetting record is described (P8150...); i.e. there is temporary tool datas, which are overwritten at a Tx M6 or Tx M1 or at the program end.

For activating everything of these tool tools is not necessary M16.

‘ t: ‘ - Tool radius correction

With the programming of ‘ t: ‘ in the offsetting record only the radius correction of the tool (P8162) is described and activated.
A G41/42 programmed afterwards uses the total of P8160 and P8162 as correction radius!
The quadrant and other tool datas remain unchanged!
This function is practical, in order to correct past tool radius.

Example:

```
N10  ...
N20  t:0.5                       {modification of the past tool radius }
N30  G1 X100 Y100 G42
N40  ...
```

‘ tr: ‘ - Tool radius

With the programming of ‘ tr: ‘ in the offsetting record the radius correction of the tool (P8162) is set to zero and the tool radius (P8160) with the programmed value is described.
The quadrant and other tool data remain unchanged!

Example:

```
N10  ...
N20  tr:50                       { setting the tool radius }
N30  G1 X100 Y100 G42
N40  ...
```

7.1 Tool functions (continued)

Tool definitions t...

‘ tl: ‘ - Tool length

With the programming of ‘ tl: ‘ in the offsetting record the length correction of the tool (P8163) is set to zero and the tool length (P8161) with the programmed value is described.
The quadrant and other tool datas remain unchanged!

Example:
```
N10   ...
N20   tl:80                        { setting the tool length }
N30   G1 X100 Y100
N40   ...
```

‘ tq: ‘ - Tool quadrant

With the programming of ‘ tq: ‘ in the offsetting record the tool quadrant (P8164) is set or modified.
Other tool datas remain unchanged!

Example:
```
N10   ...
N20   tr:5   tq:4                  {setting a tool radius with quadrant }
N30   G1 X100 Y100 G42
N40          ...
```

7.1 Tool functions (continued)

M06 Tool change

With M06 the tool change is started up. M06 with synchronisation and stop must be defined (see P11051 or P8351).

2 possibilities the tool change consist to execute:

- 1. Tool change in the cycle
- 2. Tool change completely in the PLC implements.

to 1. If the theorem analysis a M06 identifies, it is checked whether the cycle 6 (Z6) exists. There is Z6, then this sub-routine is called, in that the tool change one handles. The actual tool change may not then in the PLC any more with M06 to be executed. But other M function numbers must be used. At the program end of Z6 from the system synchronized and following will become the tool datas taken into consideration (M16).

to 2. If Z6 does not exist, then this means that the PLC with the reception of M06 that Tool change executes. After acknowledgement of M06 the tool datas become taken into consideration (M16).

7.1 Tool functions (continued)

Process of a tool change over PLC (M06 with stop and synchronisation defines):

PLC receives M06

- If a tool is in the spindle, old tools placing.
  This is closed, as P8047 with the tool place is described.

- Subsequently, the tool (T) is changed.
  (tool number and tool place were transferred in the tool data theorem).
  If the change is final, this is acknowledged with the M function-acknowledgement.

Beforehand P8045 must be however still described.

7.1 Tool functions (continued)

Course of a tool change with Z6

Z6 is called up as sub-routine with M06.

Example:	Z6		
	N10	P8047:1	store tool data
	N20	P8045:1	activate tool data
	N30	M30	

N10 P8047:1 Store tool data

The tool data theorem in the spindle tool data theorem (P8100) is retransferred into the tool data array. If the spindle tool datas are from T0, one does not retransfer.

N20 P8045:1 Activate tool data

The tool data theorem (P8050...) becomes into the spindle tool data theorem (P8100...) and the offsetting record (P8150...) copies.

N30 M30

With the program end of Z6 the tool data are activated (M16).

7.1 Tool functions

Example for course of a tool change with Z6

NC main program
N...
N...
N100 T3 M6 Call up tool change
N..
N...

When calling up Z6, the parameter area P8050..8099 is actualized with the new tool data of T3.

Z6

N10 _wzwxl_x_pos := 1000; _wzwxl_y_pos:= 300;	Definition of the position for laying down / picking up
N30 P8100=P8050.200	Check if tool-old = tool-new
N40 P8100=0.100	Check if a tool is in the spindle If not: do not store tool data
N50 G00 X:_wzwxl_x_pos Y:_wzwxl_y_pos M01	Approach position for lay down P8120:
N60 P8047:1	Store tool data (Trigger)
N100 P8050=0.200	Check if tool should be picked up. No tool picking up at T0.
N120 G00 X:_wzwxl_x_pos Y:_wzwxl_y_pos M01	Approach position for picking up
N140 P8045:1	Activate tool data
N200 M30	
#	

7.1 Tool functions (continued)

M16 Tool data call up

M16 with T in the NC block e.g.: N10 T1 M16

During pre-analysis time the corresponding tool data block from the tool data block array is loaded into the actual settlement data block with M16 according to the programmed T-number.
At T0 the settlement data block is cleared.

M16 without T in the NC block e.g.: N10 M16

The settlement data block is activated with M16.

The current spindle tool data block is not changed through M16. M16 can be transmitted to the PLC, if desired. This is not necessary for the function of the tool data call up.
The tool radius is activated with calling up G41 / G42 (tool radius path correciton left / right). The tool - length becomes active at the block end.

e.g.:	N100 T2 M16	tool data call up
	:	
	N120 X100 Y100 G42	call up tool radius path correction

7.2 Tool correction

The workpiece programming with tool correction enables the application of tools with different dimensions (example: regrinding of tools).
The dimensions of the tools are indicated with the corresponding T-address. The tool data are calculated by the control on the target path.

This enables the programming of graphically determined workpiece correction and results that the time-costly calculation of base for the cutter center point path is discontinued.

The correction method used for the BWO-control is composed of a combination between intersection - and blending radius correction.

The tool correction is only accomplished in blocks, in which the traverse to be proceeded is unequal to zero (travel differences unequal to zero at straights, radius unequal to zero at circles).
The blocks with travels equal to zero are executed at the intersection point or at the end of the inserted transition circle. The amount of one after another lying blocks with travels equal to zero is limited on 12.

Outer contour

Completion of the tool correction
at the beginning of the transition circle

Inner contour

completion of the tool correction at
the intersection point

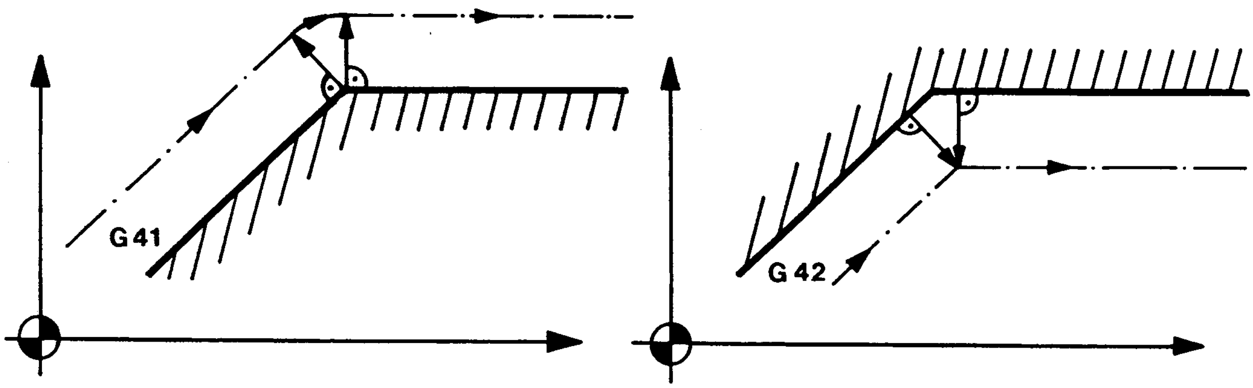


Figure 7-1 Completion of the tool correction

7.2 Tool correction (continued)

Tool dimensions

With M06 / M16 the stored tool dimensions are activated and the tool correction is accomplished.

The tool dimensions of the active tool data are calculated into the position display.
Thereby the position display of the tool axis is the same like the programmed target position, if the axis is driven in position.

The tool axis is determined by the active plane selection of the interpolation main plane. See also G17, G18, G19.

Tool radius

With the traverse conditions G40, G41 and G42 is determined, whether and how the tool radius correction has to be calculated.

Any correction is stopped with G40.
A preceded G41 or G42 is cancelled.

G41 means, that the tool is found on the left side of the programmed path (seen in feed direction).
G42 means, that the tool is found on the right side of the programmed path (seen in feed direction).

The tool radius can be entered positive or negative.
At a positive tool radius the programmed tool correction is calculated.
At a negative tool radius the programmed tool correction is changed: G41 becomes G42 and G42 becomes G41.

The tool radius correction is executed in the indicated interpolation plane. Before changing the interpolation plane correction must be cancelled with G40.

The tool correction can calculate different tool radius, which can be called up with different T-functions.

7.2 Tool correction (continued)

Position of the tool

The relative position (quadrant) between tool and workpiece is indicated in P8164.

Definition of the quadrants

P8164 = 1 to 8	quadrant 1 to 8
P8164 = 0 = 9	no quadrant correction SP = SM

SP	theoretical tool peak
SM	radius center point ot the tool

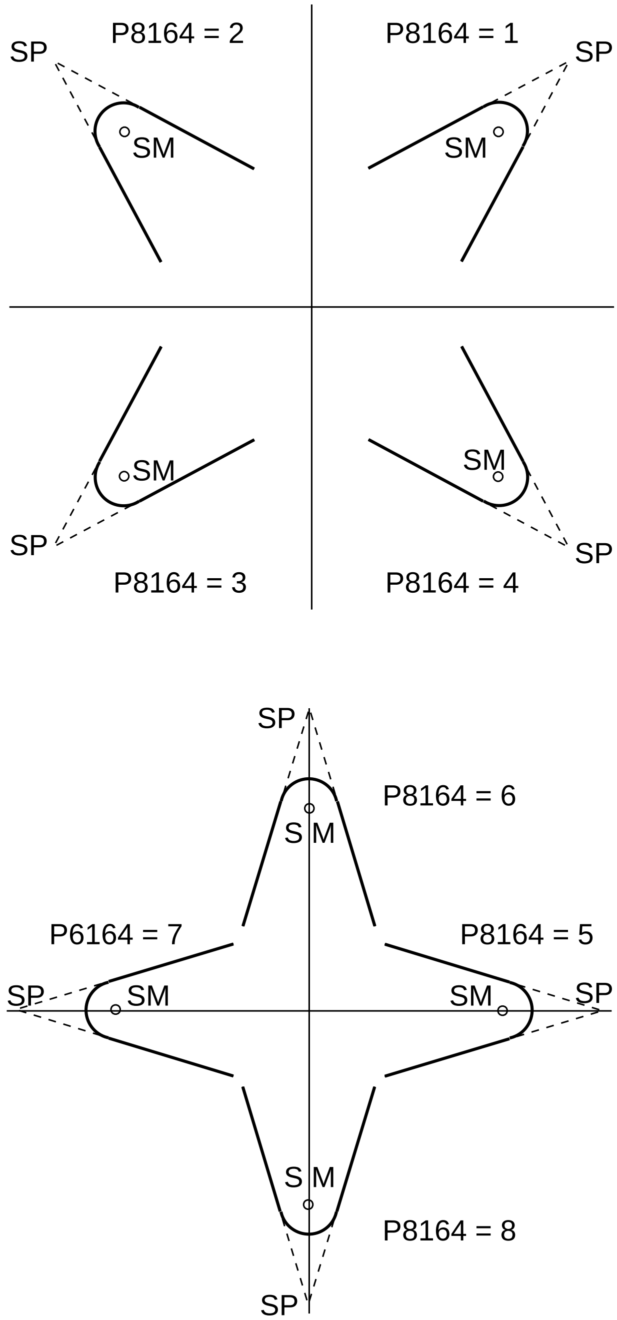


Figure 7-2

7.2 Tool correction (continued)

Special case: Switch off the correction with changing the plane

The switching off of the correction (G40) in a block with travel equal to zero followed by a block with changing plane (G17, G18, G19) leads to wrong positioning of the axes.

Example: N200 G40 Z100	switching off of the correction in plane G17 and travel equal to zero.
N210 G18 X20 Y50	position is approached wrong.
Solution: N200 G40 X40 Y100 Z100	switching off of the correction in plane G17 and travel unequal to zero.
N210 G18 X20 Y50	position is approached right.

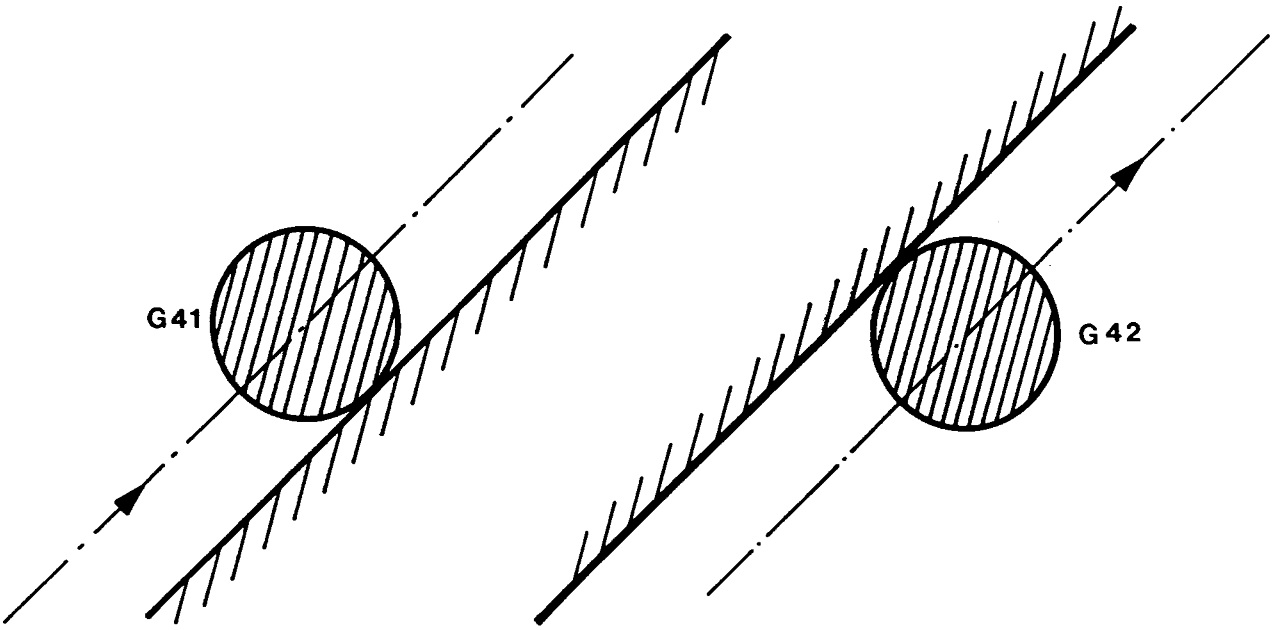


Figure 7-3 Definition of G41 and G42

7.3 Correction principle

If the correction is switched on, the path correction is always accomplished with the same principle:

The corrected path is displaced for the amount of the indicated radius value in reference to the programmed path. The thereby emerging path end points are determined by the cutting of the corrected path elements.

At the outer contour and at changing the correction a transition radius is inserted. At the inner contour the intersection is calculated.

The principle of the correction is explained in the following sketches:

Transition straight - straight

outer contour inner contour

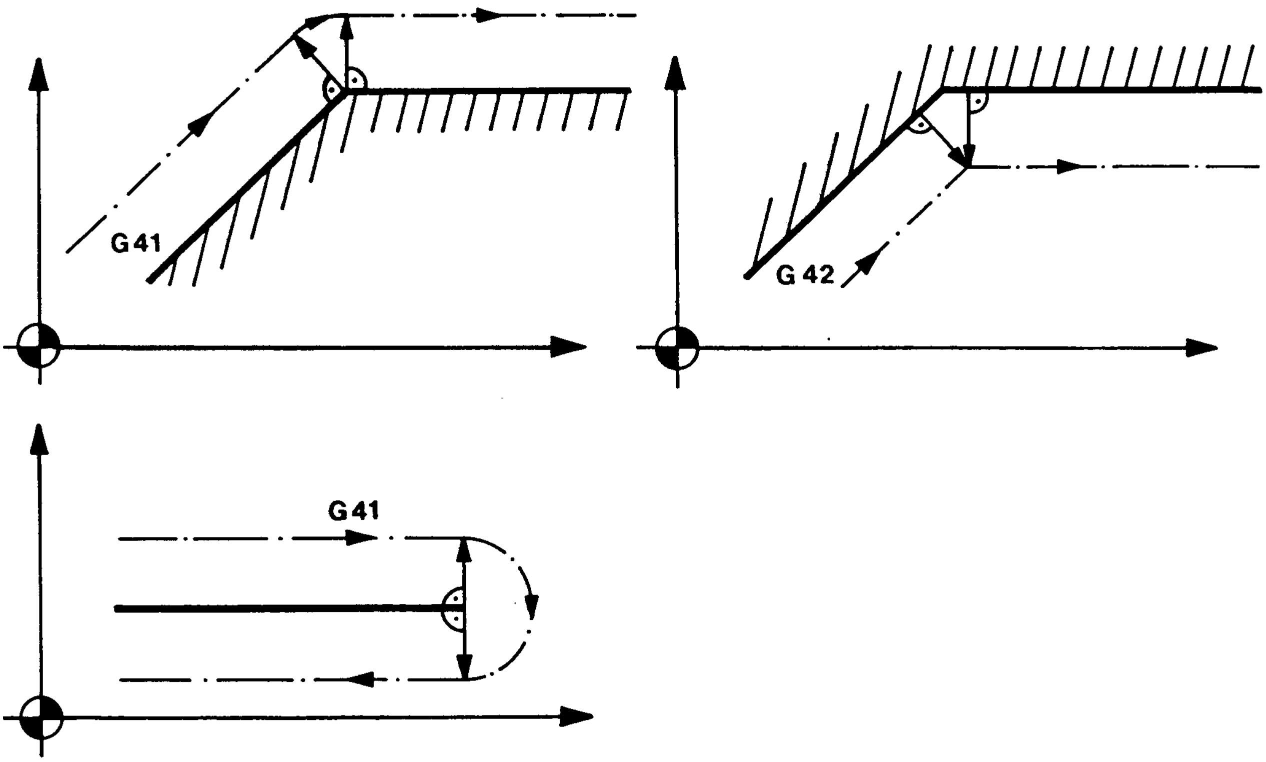
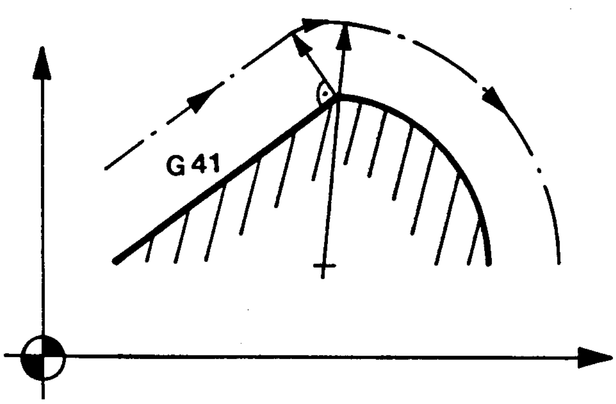


Figure 7-4

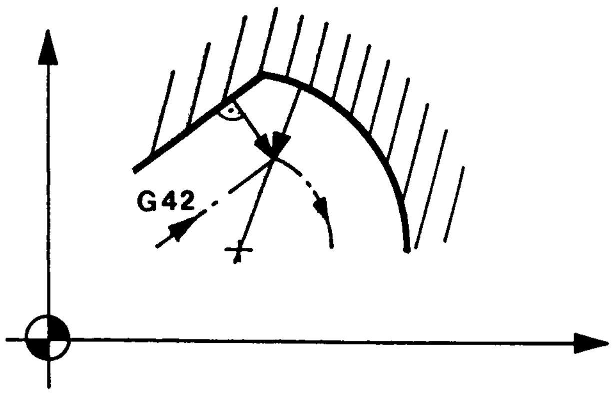
7.3 Correction principle (continued)

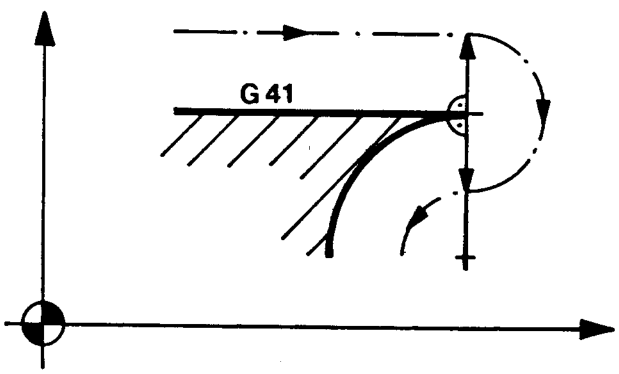
Transition straight - circle (also for circle - straight)

outer contour

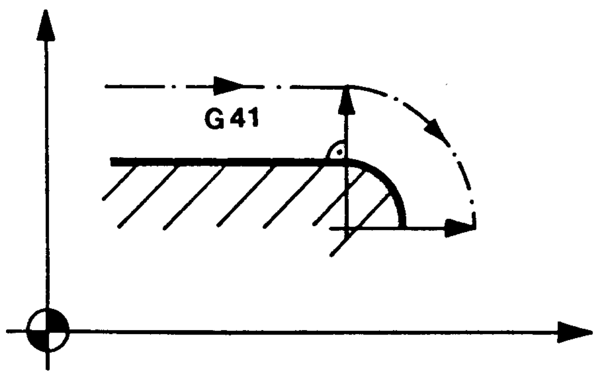


inner contour





Tangential transitions



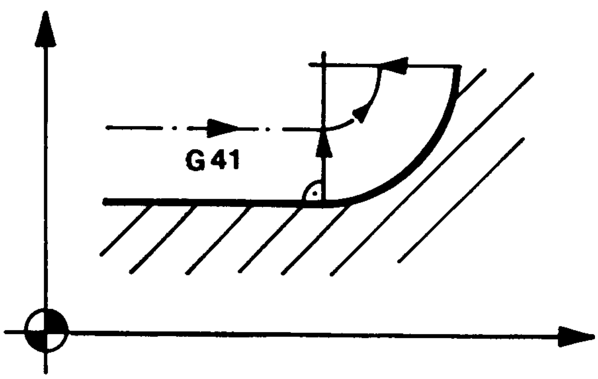
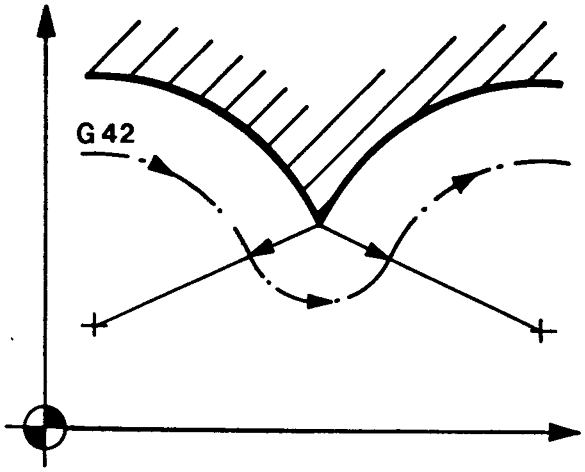


Figure 7-5

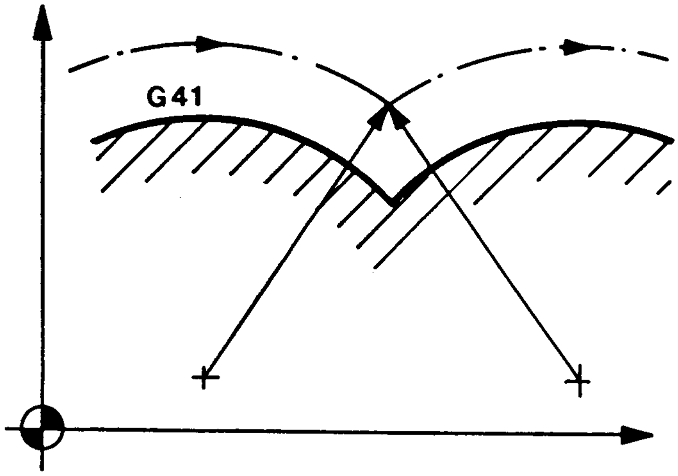
7.3 Correction principle (continued)

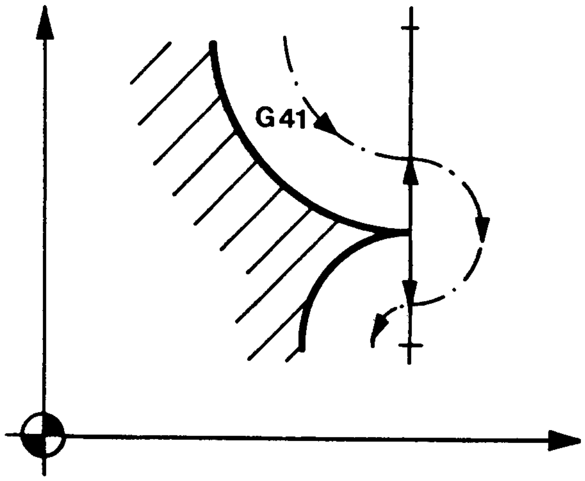
Transition circle - circle

outer contour



inner contour





Tangential transition

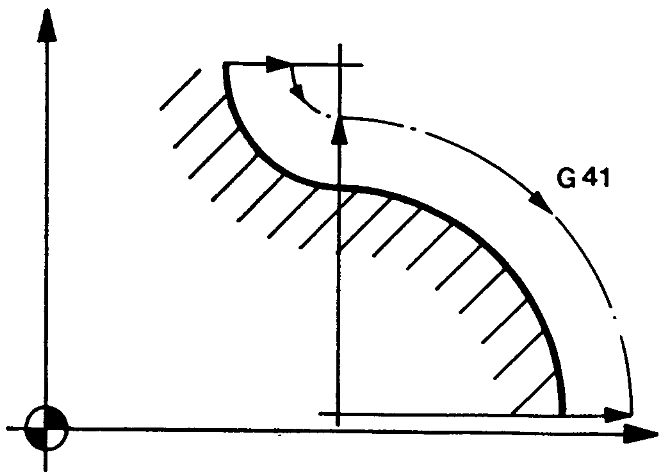


Figure 7-6

7.4 Change of the tool radius

When changing the tool radius, the intersections and transition circles are calculated first with the old radius.
The starting point (old radius) and the end point (new radius) of the corrected path do no more havethe same distance to the programmed path. This is valid for straights and circles.

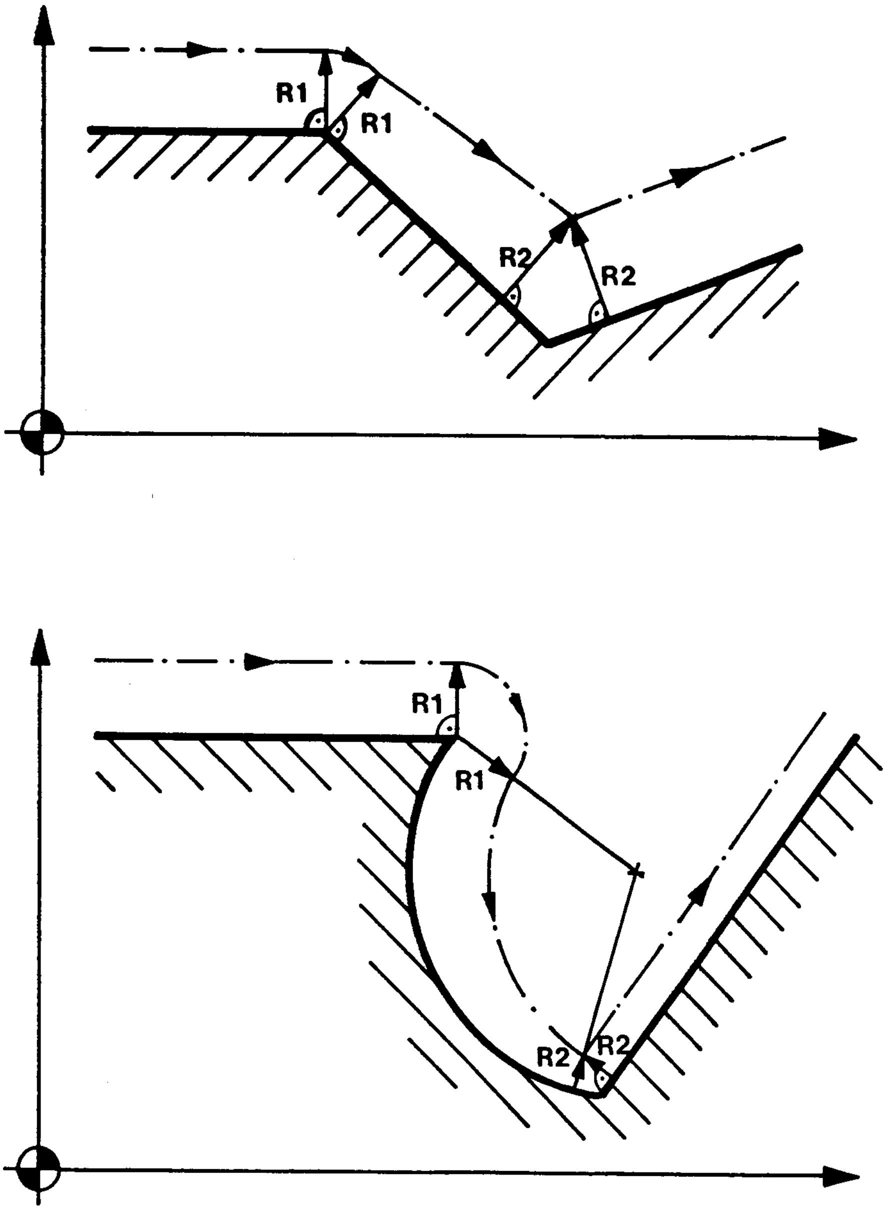


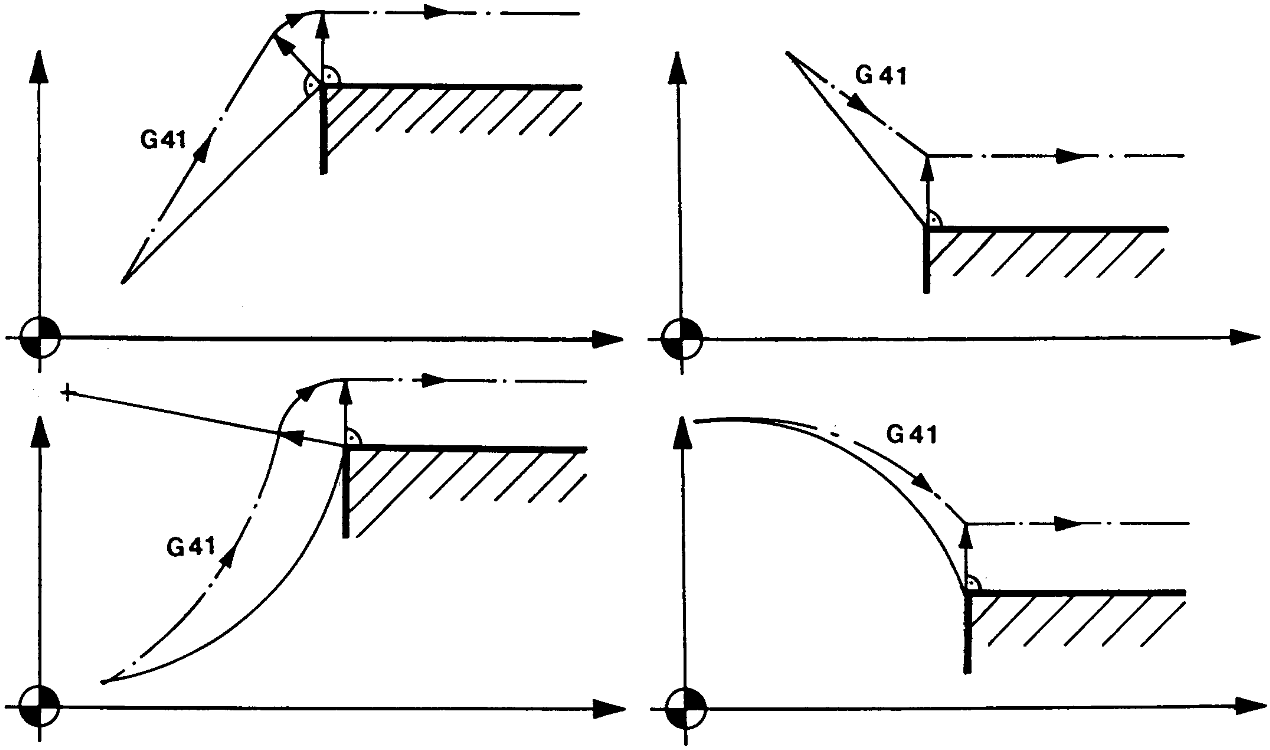
Figure 7-7

7.5 Switching on the correction

The programmed correction is clculated in each block.

outer contour

inner contour



approach tangentially

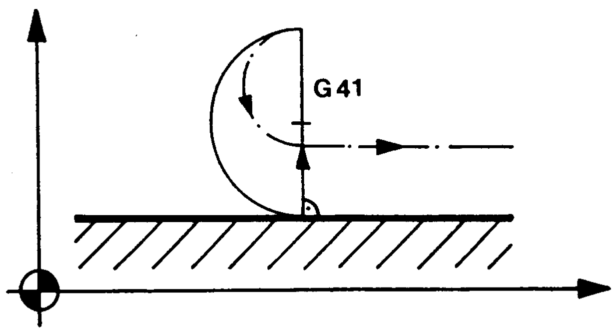
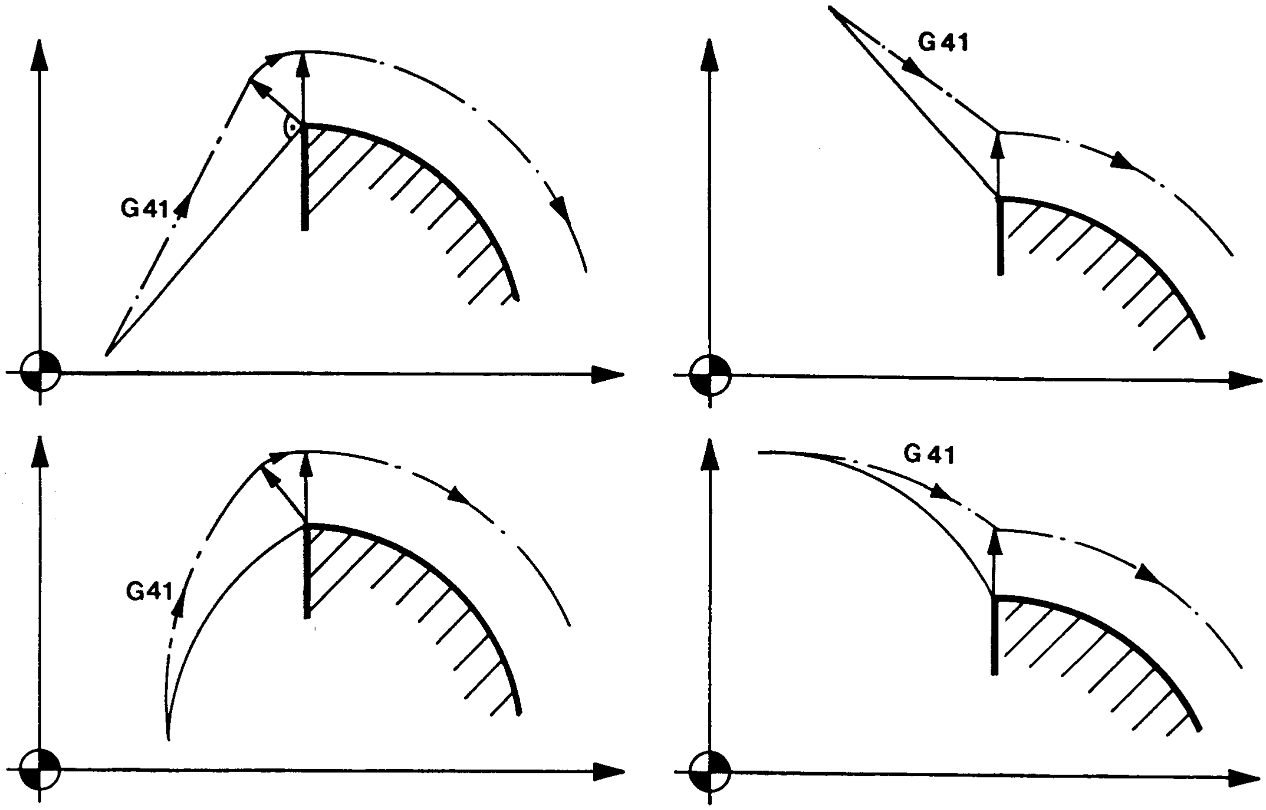


Figure 7-8

7.5 Switching on the correction (continued)

outer contour inner contour



approach tangentially

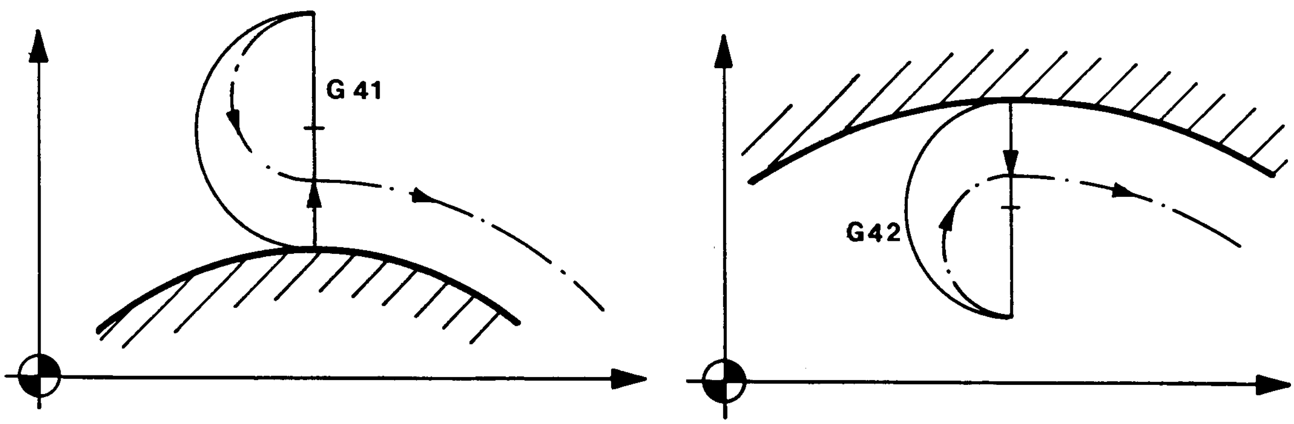


Figure 7-9

7.6 Switching off the correction

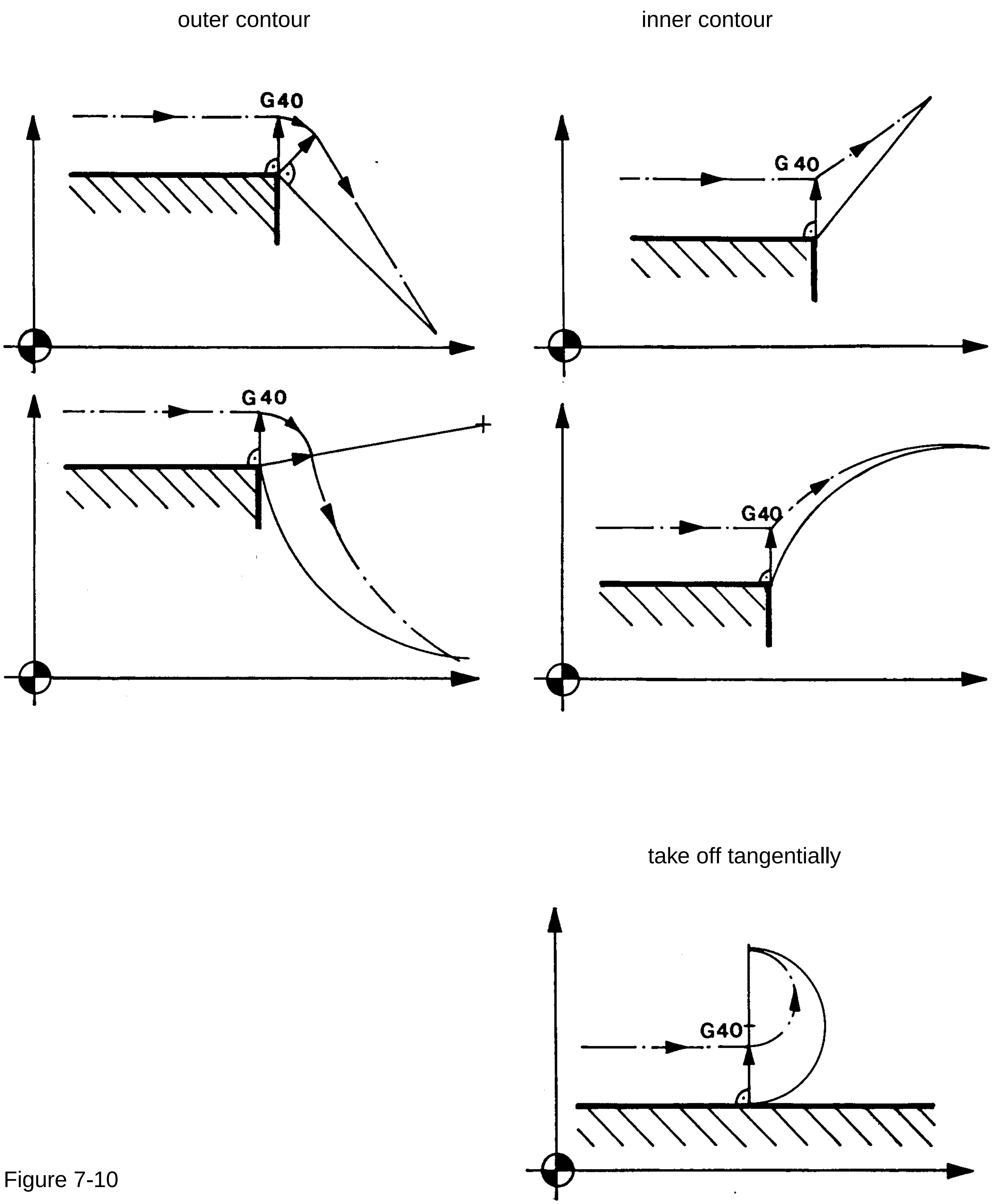
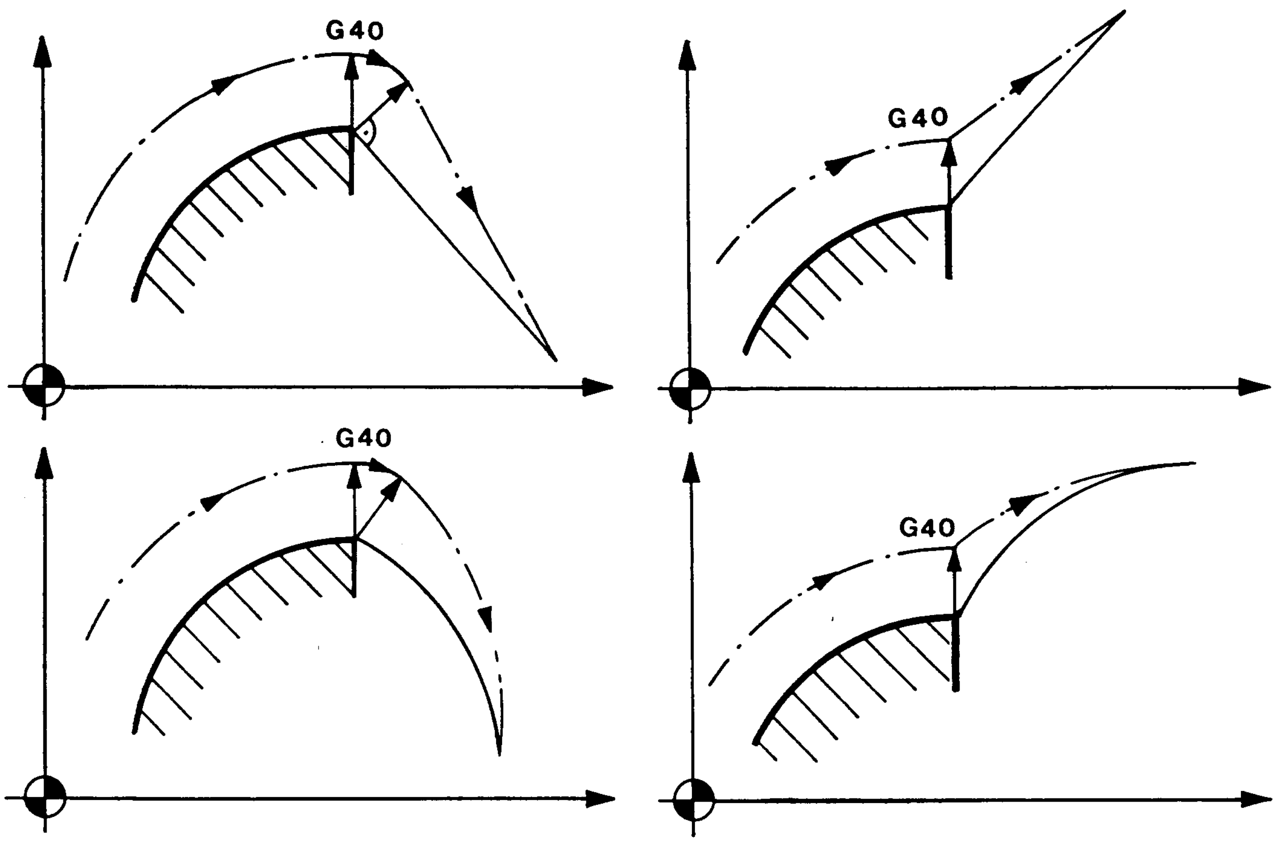


Figure 7-10

7.6 Switching off the correction (continued)

outer contour inner contour



take off tangentially

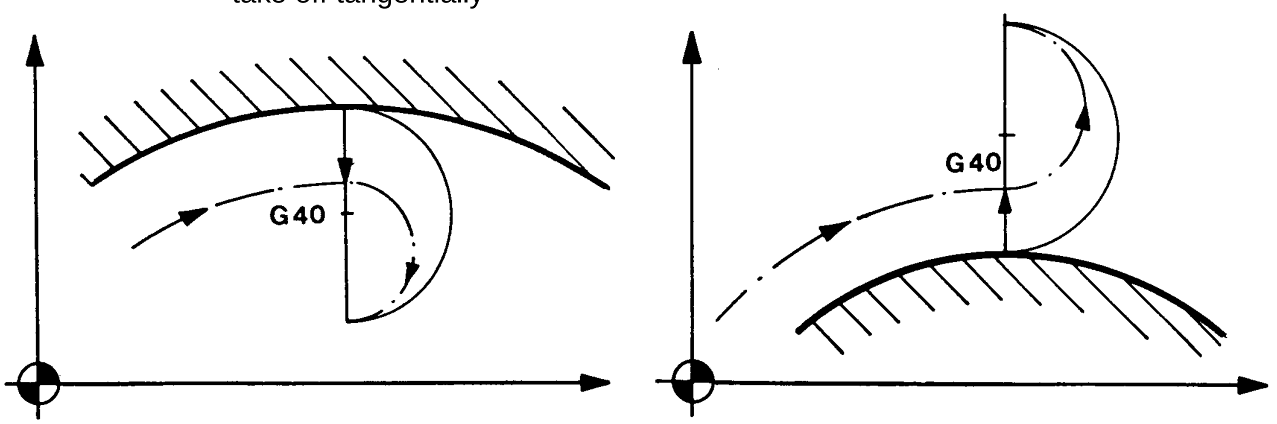


Figure 7-11

7.7 Special cases at inner contours

At inner contours undesirable contour errors can appear. In these cases the messages 1416 and 1420 are given out. The program is not interrupted.

The causes of these errors are:

1416 Too short travels in comparison with the dimension of the tool radius.
1420 Intersection not possible.

In the following figures the response of the control is illustrated for different cases.

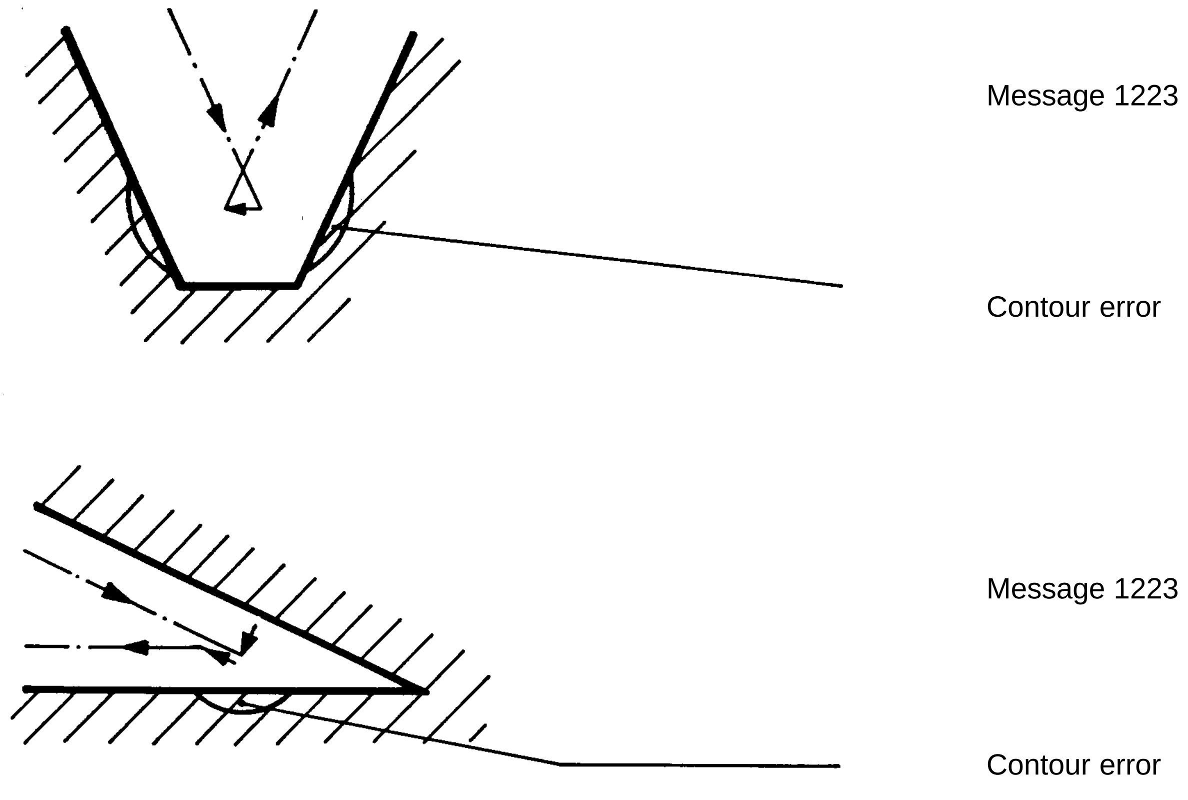
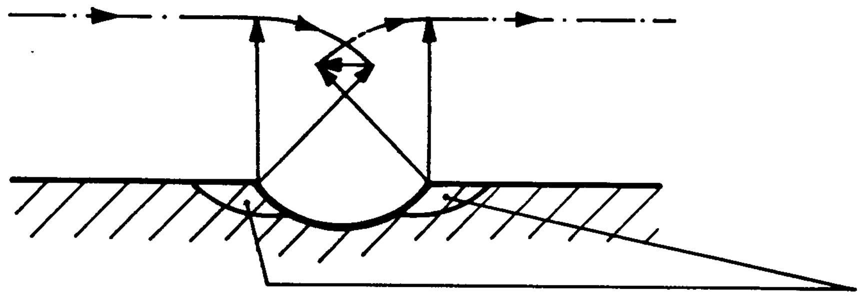


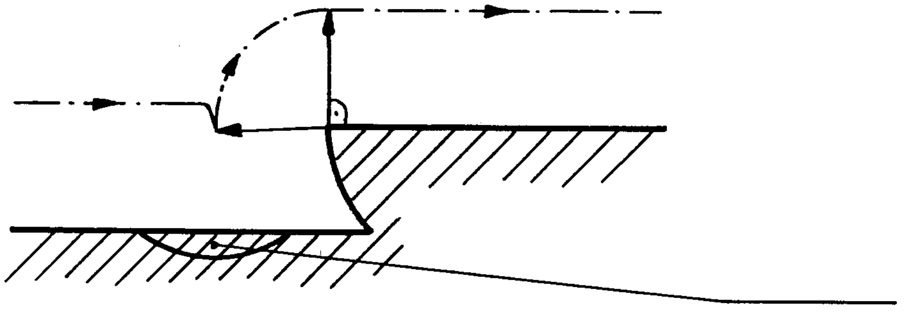
Figure 7-12

7.7 Special cases (continued)



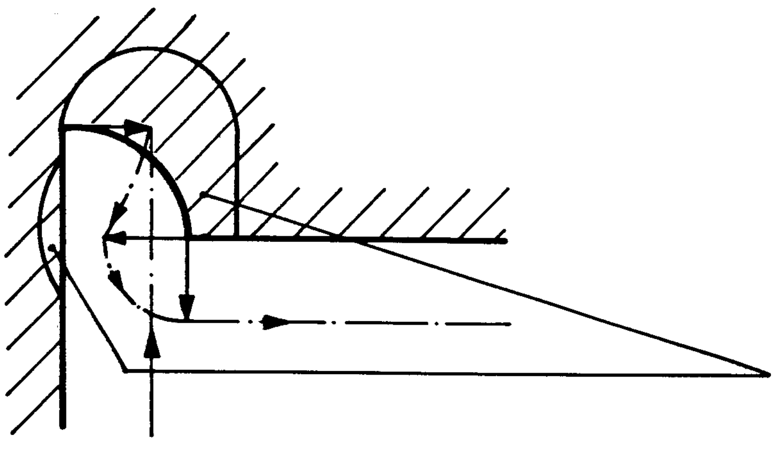
Message 1223

Contour error



Message 1223

Contour error



Message 1223

Contour error

Figure 7-13

8. Parameters

8. Parameters

8.1 General

The **System 900** puts 10000 to 50000 parameters at the users disposal, depending on the system size.
It is distinguished between two kinds of parameters:

System - Parameter q

Channel - Parameter P

System - Parameter q

This is a linear array, which is generally accessible, from q0 .. q (max. parameter).

The area of q is subdivided in function blocks:

0 ... 99	General system-configuration
100 ... 999	Definition channel-descriptor
1000 ... 1999	System-setups, system overlapping data
2000 ... 9999	Axis data
10000 ...	Channel - Parameter

Notes:

Input 0 or — means 0 or reset

Counting method byte 1...4

Counting method bit 0...7

Example Counting method byte

Parameter Nr.	Status indication	byte	Meaning
P1000	$ **30 02 04 01**		
	01	byte **1**	parameter loaded
	04	byte **2**	parameter is transferred to the operating panel
	02	byte **3**	parameter closed, if barrier 2 is settinged
	30	byte **4**	hexadecimal format / parameter in the EEPROM store

8.1 General (continued)

Channel - Parameter P

When programming a channel - parameter, a system parameter is always accessed in dependence on the channel - descriptor.
That means channel - parameters are actually system - parameters, whereby the channel - descriptor defines, which channel - parameter accesses on which system - parameter.

The assignment between channel - parameter and system - parameter is not linear.
This assignment is defined in the channel - descriptor.

Channel - Parameter are virtual parameters, which show on a system - parameter. Therefore, the channel-parameter-number can be bigger than the maximum number of the parameters.

The area of P is subdivided into function - blocks:

0 ... 6999	User block 1 Free user parameters, actual number definably in q103 standard adjustment: 5000 parameters. The user block 1 actual always on channel-specific area, i.e.: P0..P6999 von Kanal 1 and P0..P6999 von Kanal 2 are different parameters. User block 1 subdivided in:
0 ... 499	Reserved for BWO standard - cycles
0 ... 299	Cycles - transmission - area for cycles - interface
300 ... 399	Area reserved for cycles Area for filing statical, modal data
400 ... 499	Area reserved for cycles Temporary data area, Calculation fields, Scratch etc.
500 ... 6999	Free area for the user
7000 ... 9999	Fixed defined channel - parameter
11000 ... 11999	System - setup, system overlapping data, common area of all channels
12000 ... 18399	Axis data
20000 ... 29999	User block 2, e.g. Zero point datas
30000 ... 39999	User block 3, e.g.. Tool datas

d

8.1 General (continued)

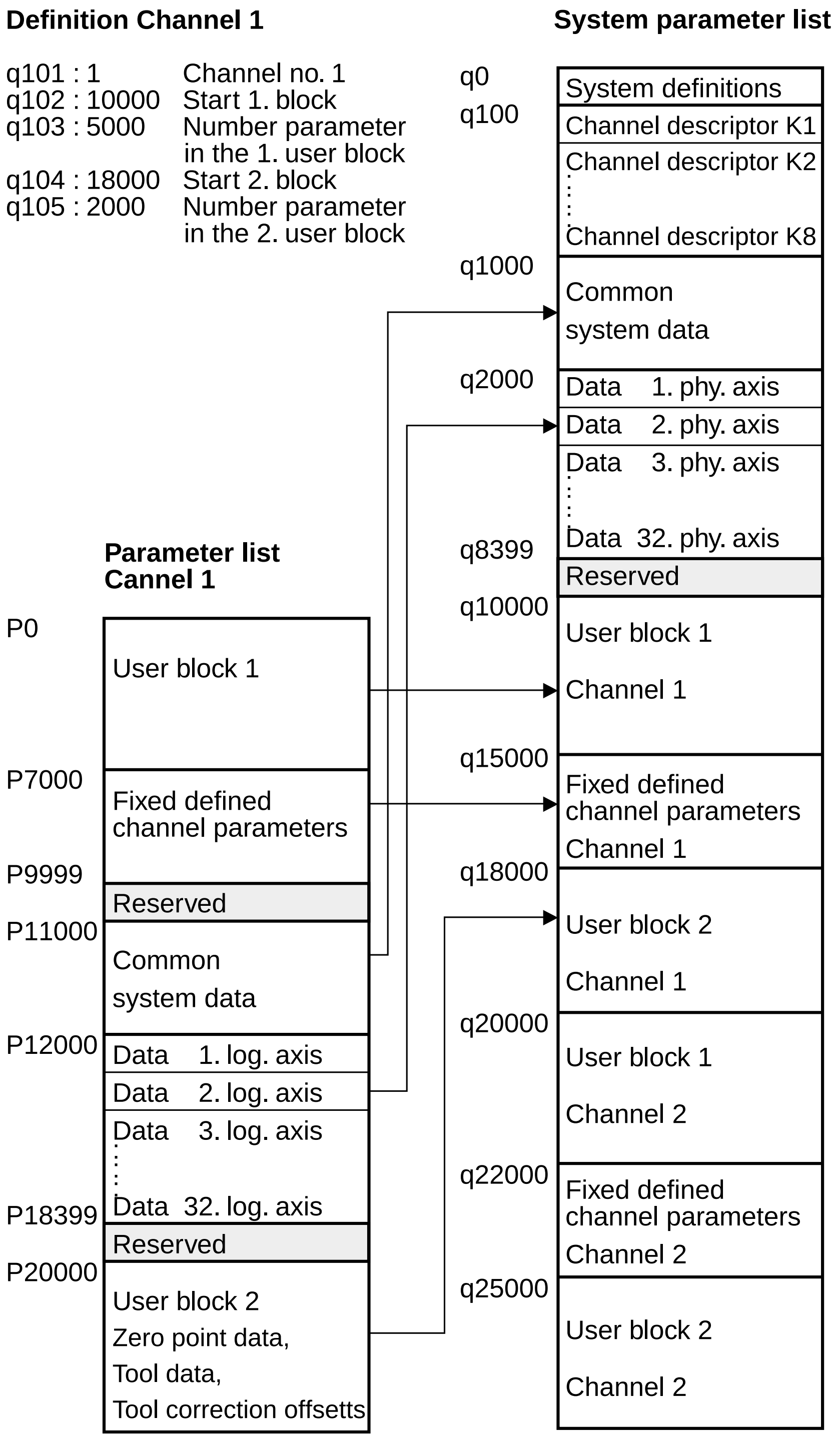
Each physical axis occupies a parameter block of 200 parameters.

In the system has physical axis	the area	in the channel has logical axis	the area
1.	q2000 ... q2199,	1.	P12000 ... P12199,
2.	q2200 ... q2399,	2.	P12200 ... P12399,
3.	q2400 ... q2599,	3.	P12400 ... P12599,
4.	q2600 ... q2799,	4.	P12600 ... P12799,
5.	q2800 ... q2999,	5.	P12800 ... P12999,
6.	q3000 ... q3199,	6.	P13000 ... P13199,
7.	q3200 ... q3399,	7.	P13200 ... P13399,
8.	q3400 ... q3599,	8.	P13400 ... P13599,
9.	q3600 ... q3799,	9.	P13600 ... P13799,
10.	q3800 ... q3999,	10.	P13800 ... P13999,
11.	q4000 ... q4199,	11.	P14000 ... P14199,
12.	q4200 ... q4399,	12.	P14200 ... P14399,
13.	q4400 ... q4599,	13.	P14400 ... P14599,
14.	q4600 ... q4799,	14.	P14600 ... P14799,
15.	q4800 ... q4999,	15.	P14800 ... P14999,
16.	q5000 ... q5199,	16.	P15000 ... P15199,
17.	q5200 ... q5399,	17.	P15200 ... P15399,
18.	q5400 ... q5599,	18.	P15400 ... P15599,
19.	q5600 ... q5799,	19.	P15600 ... P15799,
20.	q5800 ... q5999,	20.	P15800 ... P15999,
21.	q6000 ... q6199,	21.	P16000 ... P16199,
22.	q6200 ... q6399,	22.	P16200 ... P16399,
23.	q6400 ... q6599,	23.	P16400 ... P16599,
24.	q6600 ... q6799,	24.	P16600 ... P16799,
25.	q6800 ... q6999,	25.	P16800 ... P16999,
26.	q7000 ... q7199,	26.	P17000 ... P17199,
27.	q7200 ... q7399,	27.	P17200 ... P17399,
28.	q7400 ... q7599,	28.	P17400 ... P17599,
29.	q7600 ... q7799,	29.	P17600 ... P17799,
30.	q7800 ... q7999,	30.	P17800 ... P17999,
31.	q8000 ... q8199,	31.	P18000 ... P18199,
32.	q8200 ... q8399	32.	P18200 ... P18399

In the channel descriptor physical axes are assigned to the channel axes (q110 ... q141, Byte 3).

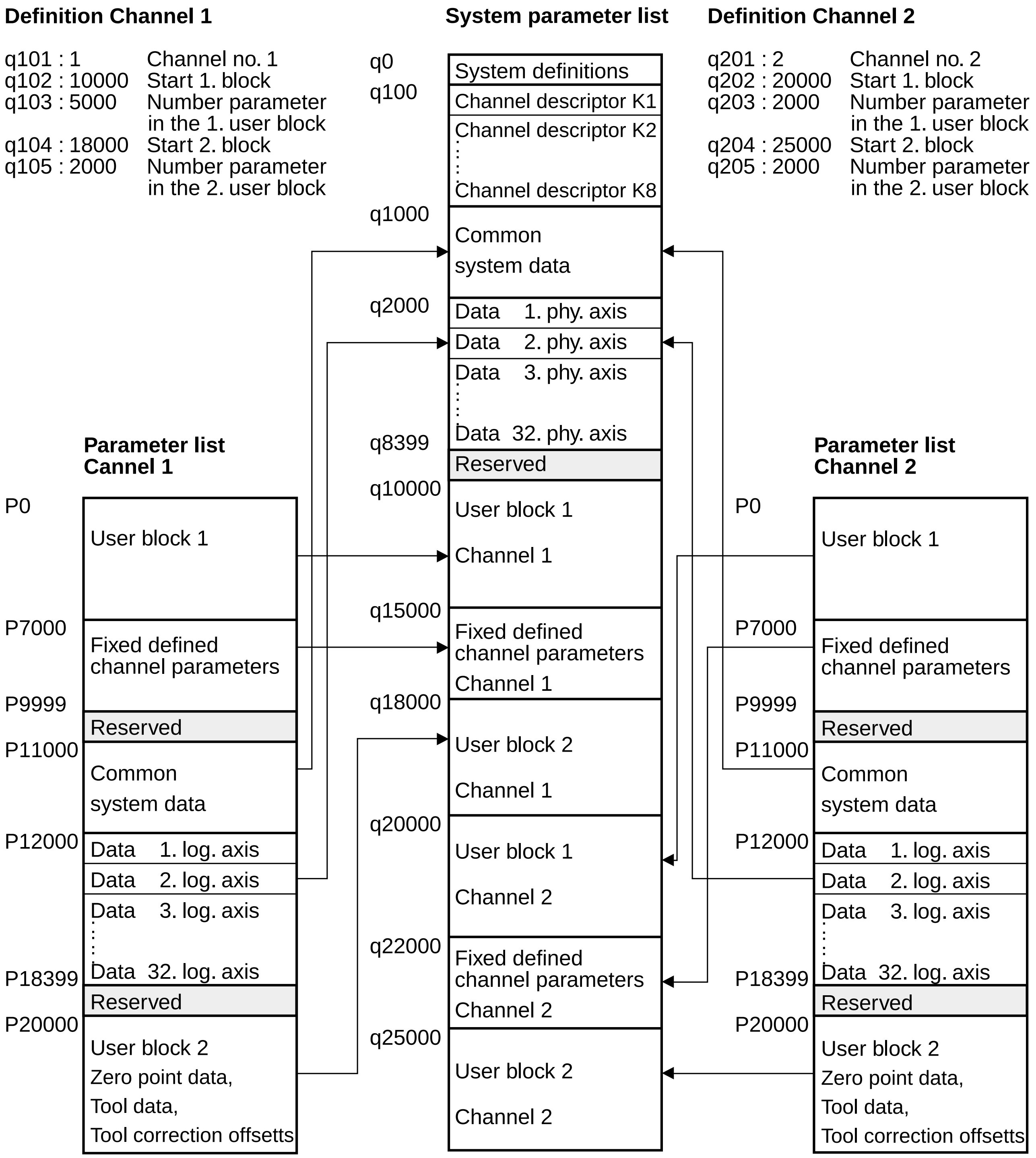
8.1 General (continued)

Configuration example for system with one channel (q2:1)



8.1 General (continued)

Configuration example for system with two channels (q2:2)



8.1 General (continued)

General to the Decimal / Hexadecimal and Binary format

display and input in the controller Representation in **Decimal / Hexa** format		allocation of the Byte's and Bit's Representation in **Binary** format (half byte format)	
0	$0	0000	0000
1	$1	0001	0001
2	$2	0010	0010
3	$3	0011	0011
4	$4	0100	0100
5	$5	0101	0101
6	$6	0110	0110
7	$7	0111	0111
8	$8	1000	1000
9	$9	1001	1001
10	$A	1010	1010
11	$B	1011	1011
12	$C	1100	1100
13	$D	1101	1101
14	$E	1110	1110
15	$F	1111	1111

input in the **Hexa** format

with key ± clocks to $ in display appears.

Now the digits know A, B, C, D, E, F over the following keys to be input :

A key ' modifying '
B key ' pos 1 '
C key ' picture up '
D key ' resetting '
E key ' end '
F key ' picture down '

With the operate of the input key ↵ those become original functions modifying, pos 1 etc. again adjusted.

Bytes	Byte 4	Byte 3	Byte 2	Byte 1
Bit number in the Byte	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
Bit number in 4 Byte	31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0

To meaning see display and input with parameter status, M functions etc.

8.1 General (continued)

Parameter status

Each parameter possesses 4 status bytes, in which additional information is contained. The parameter status is displayed in the menu parameter editor.

Example:
Parameter NR. Status indication Meaning

P1000 $30 02 04 01

byte 1 parameter loaded
byte 2 parameter is transferred to the operating panel
byte 3 parameter closed, if barrier 2 is settinged
byte 4 hexadecimal format / parameter in the EEPROM store

Status byte 1	bit 0	0	Parameter is reset.
		1	Parameter is loaded.
	bit 1		-
	bit 2	1	Parameter is loaded to the real time. This identifier works only in the AUTOMATIK operation when processing a NC program. Real time parameters do not become with an interpreting of the NC block (advance) described, but only if this NC block is actually processed. Real time parameters become from the Interpoator before the pre path M functional module described.
	bit 3	1	Parameter with synchronisation. This identifier works only in the AUTOMATIK operation when processing a NC program. Becomes from a NC program if parameters with this identifier described, becomes at the NC end of record synchronizes. (the advance of the theorem interpreter disassembled).
	bit 4		Inch bit.
		0	Parameter is not influenced by input system.
		1	After switching the system of units of inch -> metric or metric -> inch becomes this parameter in each case into that different one system of units converted (see also P11308).
	bit 5 - 6		-
	bit 7	1	FAST parameter. Parameter is transferred over fast SMMS channel. (system internal information).

8.1 General (continued)

Parameter status

Status byte 2	bit 0	1	Parameter is transferred when describing to the Interpolator.
	bit 1	1	Parameter is transferred when describing to the PLC. All parameters with settinged PLC bit become of the CNC transfer automatically after each switching on routine to the PLC.
	bit 2	1	Parameter is transferred when describing to the operating panel. With each run of the switching on routine this bit becomes with all parameters reset.
	bit 3	1	Identifier procedure parameter. (SMMS chain is passed through when describing).
	bit 4	1	Parameter becomes when describing to further CNC stations transfer (NET_GROUP) With (q59=0) this bit becomes with each run of the switching on routine with all parameters reset. With (q59=1) the bit is not changed.
	bit 5		-
	bit 6	1	Parameter is transferred when each describing to the operating panel. (speed Par function)
	bit 7	0 1	Parameter sample identifier not settinged Parameter sample identifier settinged, i.e. if P11271..=4 becomes, describing this parameter in the sample buffer logs.

8.1 General (continued)

Parameter status

Status byte 3

With activated parameter barrier
- is the parameter status nevertheless to be described
- can the PLC nevertheless the parameter value change
- see also P8511

Bit 0	1	Parameters closed, if parameter barrier 1 is settinged
Bit 1	1	Parameters closed, if parameter barrier 2 is settinged
Bit 2	1	Parameters closed, if parameter barrier 3 is settinged
Bit 3	1	Parameters closed, if parameter barrier 4 is settinged
Bit 4 - 7		-

Status byte 4

Bit 0 - 3		Format for display: Number of post-decimal positions
Bit 4	0	Decimal notation
	1	Hex notation
Bit 5	0	Do not save parameter to EEPROM
	1	Save parameter to EEPROM (see also status Byte4/Bit6)
Bit	6	Additional info for status Byte4/Bit5 (Save parameter to EEPROM)
	0	Parameter is stored in EEPROM completely (that is: the mantissa parameters and status parameters are the EEPROM is written and stored back.)
	1	ONLY parameter status is stored in EEPROM, The mantissa parameter is NOT stored. Case of a restore, the mantissa is not changed! The bit "Loaded parameter" is not changed!
Bit	7 --	

8.2 System Initializing

If a parameter in the area of q0 . .. q9 is changed, the control must be reset with ‘Clear parameter’.

q1 Number of axes
Is written at the end of the starting routine by the system. The number of axes is calculated according to the definitions in the channel descriptors (q110..).
q1 is transmitted to the PLC.

q2 Number of valid channel - descriptors.
Corresponds to the number of channels started when initializing.

q3 Number of q-parameters, minimum 30000.
q3 is described at the end of the switching on routine by the system!

q4 Number of in the system involved slave CPUs (see the File:NETCONF)

q5 Pointer on system data enlargement

q7 CPU DRAM Memory Size [MByte]

q8 CPU clock
One describes by the system

q9 Version of the operating system of the operating panel
Parameter is written from the operating panel after the starting routine.
Enable new operating panel commands
0
1 new actual position display

8.2 System - Initializing (continued)

q10 FLASH Functions / special functions

So long a FLASH function actively is, should not the system not switched off become! For processing a q10-Funktion itself the machine must in operating mode HAND to find.

0 Back message: Function terminates
1 Reset the debugger

7 Direktory chain of the NC Speichers is again created
(see also P11150 byte 2)

10 BWO system function
11 BWO system function

30 Delete the NC program was defined by q11 in the NC memory.
q10 : 0 : re-registration: executed function

q11 : error code:	0	OK
	12xx	error

50 FileScan in NC memory
conformity of all program names in NC saving is tested.
programs with ‘ ‘ or / \:*?” > < | are reset.

51 FileScan in FLASH memory
conformity of all files in CNC FLASH Saving is tested.
files with ‘ ‘ or / \:*?” > < | are reset.

8.2 System - Initializing (continued)

q10 FLASH Functions / special functions

99 Activate channel descriptors (function 99 is effective only with EA import)

Application:
With this function defined channel descriptors (q100.., q200.., q300..) become in accordance with q2 (number of valid channel descriptors) activates.
Channels are however not started!
This function is helpful while the loading of the machine data, even if here channel parameters must be described by channels, which not yet actively are.

Example:

```
q      2 : x
:
q  100 : ...
q  200 : ...
q  300 : ...
:
q 1000 : ...
q 2000 : ...
:
q   10 : 99    Activate channel descriptors
:
K1:P8250 : ...
:
K2:P8250 : ...
:
K3:P8250 : ...
:
```

8.2 System - Initializing (continued)

q10 Functions:

120 Reset all parameters in the FLASH

125 Reset all NC programs in the FLASH

170 Store all marked parameters in the FLASH memory
(max. 14000 parameters).

Being supposed the parameter areas of several channels to be stored, the following is to be observed:
Channel descriptors that channels which can be stored must available / defined to be, i.e., q2 (number of channels) was changed, to start then the CNC is before memory again. Only with the passing through of the switching on routine become in the CNC, in accordance with q2, channels started.

In the machine data file with q10:99 if the channel descriptors was activated, so the parameters can to be stored immediately.

175 Store all visible NC programs / cycles from that NC memory in the FLASH
(see also P11149)
(max. 262kBytes).
In the starting routine after the ' NC memory ' all will reset in the FLASH stored programs into the NC memory written back.

176 Store all cycles of the NC memory in the FLASH
In the starting routine after the ' NC memory ' all will reset in the FLASH stored programs into the NC memory written back.

177 Store all cycles and *. CLASS files of the NC memory into Flash
File : NCZYK
After switching on the NC memory is deleted, then all stored in FLASH programs are written in the NC memory.

185 Function as CMD 175, however, the data is stored (NCZYK) coded.
186 Function as CMD 176, however, the data is stored (NCZYK) coded.
187 Function as CMD 177, however, the data is stored (NCZYK) coded.

190 Restore q0..q999 from the FLASH memory.

200 Restore all parameters (apart from q0..q999) from the FLASH memory.

205 Restore all NC programs / cycles from the FLASH.
(File NCZYK is read)

8.2 System - Initializing (continued)

500 With the I/O picture in the CNC direktory of the NC memory one displays.

501 With the I/O picture in the CNC Direktory FLASH Contents are displayed.

510 The CNC DLL Left is removed
Caution: With the next start of the system, those is to CNC DLL not
more for the order.
Function is executed only if in q11:510 one enters before.

520 The switch ' NO_DMA ' is removed
Function is executed only if in q11:520 one enters before.

521 The switch ' NO_DMA ' is settinged
Function is executed only if in q11:521 one enters before.

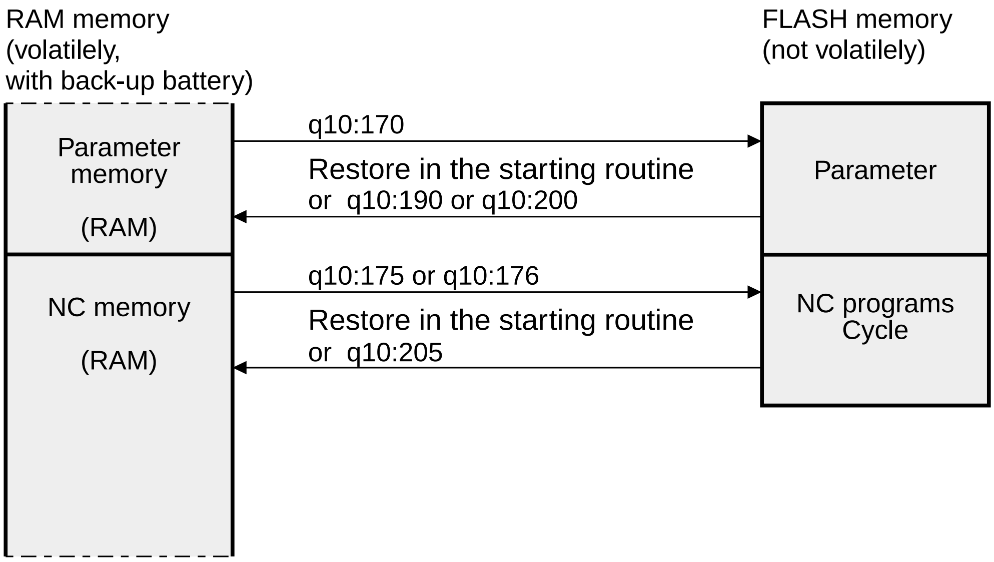
530 The switch ' PLC_BIG ' is removed
Function is executed only if in q11:530 one enters before.

531 The switch ' PLC_BIG ' is settinged
Function is executed only if in q11:531 one enters before.

540 The switch / Document ECC is removed.
Old parameters certification mark is formed for all parameters.
Function is only executed if the 540 was entered in q11.
The machine must be in the Emergency.
Outputs may fall under certain circumstances.
see also q86 / M181

541 The switch / Document ECC is removed.
ECC parameters certification mark is formed for all parameters.
Function is only executed if the 540 was entered in q11.
The machine must be in the Emergency.
Outputs may fall under certain circumstances.
see also q86 / M1812

8.2 System - Initializing (continued)



8.2 System - Initializing (continued)

q12	Operating system - version
q13	Operating system - check sum
q14	Operating system - time stamp day
q15	Operating system - time stamp month
q16	Operating system - time stamp year
q17	Operating system - time stamp hour
q18	Operating system - time stamp minute
q19	Operating system - time stamp second

q22	DLL version
q23	DLL check sum
q24	DLL time stamp day
q25	DLL time stamp month
q26	DLL time stamp year
q27	DLL time stamp hour
q28	DLL time stamp minute
q29	DLL time stamp second

q32	PLC version	(planned)
q33	PLC check sum	(planned)
q34	PLC time stamp day	(planned)
q35	PLC time stamp month	(planned)
q36	PLC time stamp year	(planned)
q37	PLC time stamp hour	(planned)
q38	PLC time stamp minute (planned)	
q39	PLC time stamp second (planned)	
q40	BF Systemkennung	(panel ID)

8.2 System - Initializing (continued)

q41 Panel - System-bit information

CNC: from version 150/210 and panel version 150
q41 is set by the system after the power-on routine.
Byte 1 :
Bit 0 0 : nc_line_mode not active
1 : nc_line_mode active
(Switch 'nc_line_mode' in NETCONF set)

nc_line_mode not enabled
NC blocks must necessarily corresponding 'N' or '/ N' begin.
NC Editor is NC block-oriented.

When switching from nc_line_mode by NOT
must nc_line_mode the NC memory can not be erased.
NC programs with sentences without number can not be properly read / processed!

nc_line_mode activated
(From the CNC version 150 and the panel version 150 is implemented)
The nc_line_mode is activated with the entry: 'nc_line_mode' in the file:
NETCONF der CNC
NC blocks do not have to necessarily start with 'N' or '/N' now.
Valid entry point marks still have to but start with 'N' or '/ N'..

When switching from NOT nc_line_mode after nc_line_mode,
the NC memory must not be deleted.

IO-INPUT :
- When reading of NC programs blank lines are not filtered.
- Sentences starting with ';' are still be filtered!

NC Editor is line oriented.
(Display the line number)
NC block numbers now act as a jump marks for NC block jumps.
Entry point marks must be in ascending order!

e.g.: .

```
G01 F1000 X100
M23.100 —> Jump by Brand N100
.
N100 X150
G04.1
X100
.
```

8.2 System - Initializing (continued)

Function: NC Renumber is blocked at nc_line_mode enabled.
During the teaching of NC blocks no block number is inserted.

In AUTOMATIC Picture NC blocks are displayed with line numbers.

The following parameters now contain line numbers:

P8695	Line number at RWL
P8696	Line number at the measuring position recording
P8803	Line number in the NC program abort
P8901	current line number NC program real-time
P8905	current line number cycle real-time
P8907	Line number in the NC program abort
P8911	Line number of the faulty block
P8914	current line number in subroutine call
P9301	current line number NC program pre-analysis
P9305	current line number pre-analysis cycle
P11146	Line number of the last edited NC program
P11152	not used
P11153	not used
P11154	not used
P11184	current line number at EA In-/Output

The following parameters contain According STILL sentence numbers:

P8801	NC Start record number
P8821	Emergency program 1 set number
P8823	Emergency program 2 set number
P8825	Emergency program 3 set number
P8827	Emergency program 4 set number

Bit 1 : -
Bit 2 : -
Bit 3 : -
.
Byte 2 :
Byte 3 :
Byte 4 :

8.2 System - Initializing (continued)

q42	OP version	(planned)
q43	OP check sum	(planned)
q44	OP time stamp day	(planned)
q45	OP time stamp month	(planned)
q46	OP time stamp year	(planned)
q47	OP time stamp hour	(planned)
q48	OP time stamp minute	(planned)
q49	OP time stamp second	(planned)

8.2 System - Initializing (continued)

ESR (starting routine) information

q50	Info	1	Parameters are deleted in ESR.
		0	Parameters are not deleted in ESR.
			Parameter shows information of the latest ESR.
q51	Info	1	Parameters are deleted in ESR.
			Parameter is waiting and has to be deleted by the user.
q52	Info	1	NC program memory is deleted in ESR.
		0	NC program memory is not deleted in ESR.
			Parameter shows information of the latest ESR.
q53	Info	1	NC program memory is deleted in ESR.
			Parameter is waiting and has to be deleted by the user.

q55 IP address of the CNC CCU (only ETH)
Example: CNC CCU IP address = 172.16.30.78
Parameter value: $$AC101EÊ
$$AC -> 172
$10 -> 16
$1e -> 30
$$AC -> 78

q59 0 or — NET_GROUP bits in the parameter status become in each switching on routine reset.
1 NET_GROUP bits in the parameter status become in each switching on routine does not change.

q60 Password 1 (planned)
:
q68 Password 9 (planned)

q70 Priority PLC (planned)
q71 Priority channel 1 (planned)
:
q78 Priority channel 8 (planned)

q80 Parameter number (q) the defective parameter
q81 Parameter value 1 of the parameter defined in q80
:
q84 Parameter value 4 of the parameter defined in q80

8.2 System - Initializing (continued)

q85 Number of corrected parameters with ECC algorithm

q86 Parameter-marked type : 0 no ECC
1 ECC turned on

q87 Exception reason in Q87 : (see also P8505)
Byte 1 :
Bit 0 Inexact
Bit 1 Underflow
Bit 2 Overflow
Bit 3 Divide by Zero
Bit 4 Invalid Operation
Bit 5 Unimplemented Operation
Bit 6 :
Bit 7 :

Byte 2 :
Bit 1 : NotAlligned-memory access

q88 System address in Q88 (see also P8506)

q89 Task-Id in q89 (see also P8507)

$100	PLC-Task
$101	PLC-CMD-Task
$102	PLC-Background-Task
$103	PLC-Server-Task
$200	Proma-Task
$201	Proma-CMD
$300	Network-Task
$301	Network-Master
$400	CMD-STR-Task
$500	SMMS-Task
$600	Oskar-Task
$700	position control
$80x	AUTOMATIK-Task
$900	real time-Task
$a00	Monitor-Task
$b00	Proma-Transport-Task
$c00	CPU2-TX-Task
$c01	CPU2-RX-Task
$d00	Follower-Task

8.2 System - Initializing (continued)

Internal system parameter to the PLC

q97 Fatal error

0 or — No error

< > 0 Error number

a cause for fatal error (M1800)

q97 is set by the PLC

q98 Error message information for the display

q98 and q99 are set again at each channel changeover

Byte 1 Channel 1

Bit 0: System message
Bit 1: High priority message
Bit 2: Low priority message
Bit 3: Display message
Bit 4: PLC high priority message
Bit 5: PLC low priority message
Bit 6: free
Bit 7: free

Byte 2 Channel 2

Byte 3 Channel 3

Byte 4 Channel 4

q99 Error message informations

Byte 1 Channel 5

8.3 Channel - descriptor

The channel - descriptors are in the area of q100 . . . 899, that means:

Channel 1 q100 . . . q199
Channel 2 q200 . . . q299
Channel 3 q300 . . . q399
Channel 4 q400 . . . q499
Channel 5 q500 . . . q599
Channel 6 q600 . . . q699
Channel 7 q700 . . . q799
Channel 8 q800 . . . q899

If a parameter in the area of q100 . . . q899 is changed, the control must be restarted.

8.3 Channel - descriptor (continued)

Parameters for channel - descriptor 1

q100 Channel parameter Relokator
0 or — The fixed channel parameter area (P7000... P9999) adds itself seamless to the 1. User block (P0... Pxxxx) on.
> 0 The fixed channel parameter area (P7000.. P9999) starts off the defined q parameter.

q101 Channel - No. 1 channel 1 fix defined

q102 Parameter - relocator
Here is defined, from which q on the channel starts, e.g. q102: 10000,
i.e. P0 shows on q10000

q103 Amount of channel - parameters in the 1st user block P0 . . . P6999
Here is defined, how many parameters actually should be reserved.
Input: 1 . . . 7000
The fixed defined channel-parameters succeed to the 1st user block (3000 parameters).

q104 Parameter - relocator for 2nd user block P20000 . . .
Here is defined, from which q on this block starts, e.g. q104: 15000,
i.e. P20000 shows on q15000
The area for the fixed defined channel-parametes is between the 1st. and 2nd user block (3000 parameter).

q105 Amount of the channel - parameters in the 2nd. user block P20000 . . . P29999
Here is defined, how many parameters actually should be reserved.

Input:	0 or —	no user block installs
	1... 10000	number of available parameters

q106 Parameter - relocator for 3rd user block P30000 . . .
Here is defined, from which q on this block starts, e.g. q106: 10000
i.e. P30000 shows on q10000

q107 Amount of channel - parameters in the 3rd. user block P20000 . . . P39999
Here is defined how many parameters actually should be reserved.

Input:	0 or —	no user block installs
	1... 10000	number of available parameters

8.3 Channel - descriptor (continued)

Parameters for channel - descriptor 1

q110 Axis name and axis assignment
Here is defined, with which name the axis is programed and which physical axis is addressed.
The Control determines with the system initialising the number of the axes available in the channel using this data.

1st. log. axis assignment -> to physical axis!
byte 1 Axis name (ASCII)

ASCII	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Hex	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51	52	53	54	55	56	57	58	59	5A
ASCII	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
Hex	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F	70	71	72	73	74	75	76	77	78	79	7A

Not certified axis identifiers are deposited with raster.

byte 2 Axis identifier extended 1...8
e.g.: X1, X2 etc.
note: If byte 1 and byte 2 = 0,
there is no actual position display at the operating panel.
If byte 2 = 0, consists the axis identifier only of one character (byte 1).

byte 3 Physical axis number 1...32
Defines, which physical axis is to be addressed.
To observe: Input in the hexadecimal system
e.g.: axis number 32 —> input: $20

byte 4 Key allocation
Defines, with which key in the operating panel axis block this axis one selects.
0 Axis is not selectable
1...79 Key number
To observe: Input in the hexadecimal system
e.g.: key number 66 —> input: $42

with RC910:
extended axis selection keys 1..12. Axis code 1.. 12
13..32. Axis code 113..132
graphic —> keyboard RC910 extended axis keys

8.3 Channel - descriptor (continued)

Parameters for channel descriptor 1

q110 Axis identifier and axis allocation (continuation)

Examples:

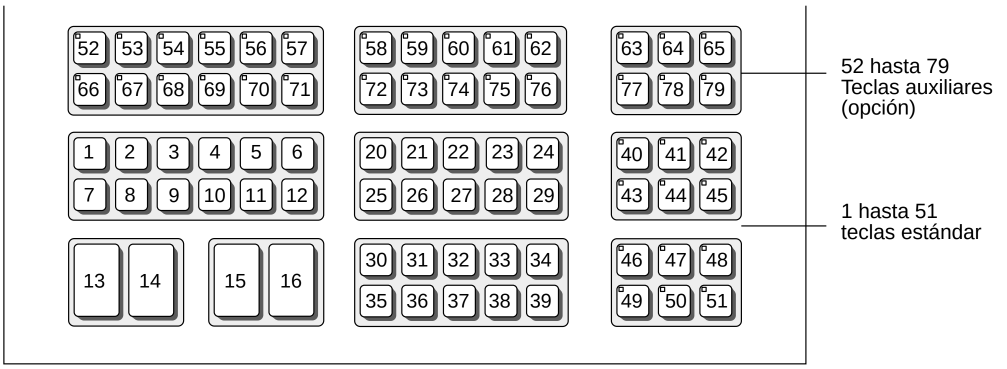
q 1 1 0 : $ 0 1 0 1 0 0 5 8

axis identifier	58 -> X
axis identifier extension	00 -> keine
physical axis	01 -> 1
axis selectable over key	01 -> 1

q 1 1 0 : $ 4 2 1 f 0 3 5 9

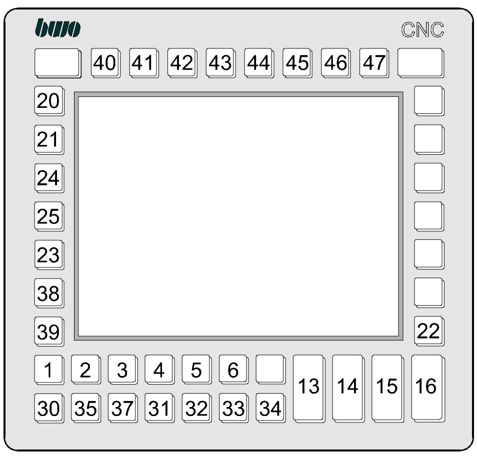
axis identifier	59 -> Y
axis identifier extension	03 -> 3 (Y3)
physical axis	1 f -> 31
axis selectable over key	42 -> 66

Key numbers in CNC 900 and CNC 900C



Key numbers in

CNC 902 / CNC 903
CNC 904 / CNC 905
CNC 910 / CNC 920
CNC 930 / RC 910



8.3 Channel - descriptor (continued)

Parameters for channel - descriptor 1

q111	2nd. log. Axis
q112	3rd. log. Axis
:	
q141	32nd.log. Axis

q142
:
q149

Spindel assignments
Assignment of the in the NC - program programmed spindle data to the corresponding spindle blocks. Example: q143 : 4
When programming S2 in the NC - program, the number of revolutions in the 4th. spindle block is changed. Input: 1...8

q142	Spindle block reference for S
q143	Spindle block reference for S2
q144	Spindle block reference for S3
q145	Spindle block reference for S4
q146	Spindle block reference for S5
q147	Spindle block reference for S6
q148	Spindle block reference for S7
q149	Spindle block reference for S8

q150	Polar co-ordinate system : Name for radius (e.g. 'x')	Input the ASCII - code
q151	Polar co-ordinate system : Name for angles (e.g. 'c')	Input the ASCII - code

8.3 Channel - descriptor (continued)

Parameters for channel - descriptor 1

q152 Definition of the coordinate system for G17
byte 1 1. logical axis number abscissa (horizontal axis)
01 centre point identifier I
byte 2 2. logical axis number ordinate (vertically axis)
02 centre point identifier J
byte 3 3. logical axis number
03 vertically axis on the interpolation plane.
default $030201

q153 Definition coordinate system for G18
byte 1 1. logical axis number abscissa (horizontal axis)
03 centre point identifier K
byte 2 2. logical axis number ordinate (vertically axis)
01 centre point identifier I
byte 3 3. logical axis number
02 vertically axis on the interpolation plane.
default $020103

q154 Definition coordinate system for G19
byte 1 1. logical axis number abscissa (horizontal axis)
02 centre point identifier J
byte 2 2. logical axis number ordinate (vertically axis)
03 centre point identifier K
byte 3 3. logical axis number
01 vertically axis on the interpolation plane.
default $010302

8.3 Channel - descriptor (continued)

Parameters for channel - descriptor 1

q157 CNC error message Acknowledgement
0 or — The acknowledgement of pending CNC Error messages takes place with mode change after HAND (—> when describing P8683).
Additional is possible it to acknowledge over P8508 CNC Error messages.
1 The acknowledgement of pending CNC Error messages takes place only with describing P8508.
With the mode change after HAND NO CNC becomes error messages acknowledges.

q158 M function definition block from system or channel
0 or — M0..M199 are defined in the system block (i.e. P11050..P11099)
M200..M999 are defined in the channel block (i.e. P8300..P8499)
1 M0..M999 are defined in the channel block (i.e. P8250..P8499)

q159 Default of the number of post-decimal positions of the position parameters
After ‘ parameter = resetting ‘ (switching on routine) the following parameters become (in accordance with contents of q159 adjusted in the FLASH Memory) to post-decimal positions:
- axis displays P12150...12169....
- null point record
- tool data theorem (starting from P10)
0 or — 3 post-decimal positions (standard adjustment)

q166 Definition of the robotic data set (1..2)
contnent: 0 , deleted,
1 Robot-data set 1 (P11800..P11809) is used
2 Robot-data set 2 (P11810..P11819) is used

In q166 = 0 or deleted : Robot-data set 1 (P11800..P11809) is used

q167 Definition of Polar data set (1..2)
Content: 0, deleted,
1 Polar Record 1 (P11820.. P11824) is used
2 Polar Record 2 (P11820.. P11824) is used
In q167 = 0 or deleted: Polar Record 1 is used

8.3 Channel - descriptor (continued)

Parameters for channel - descriptor 1

q170 At the beginning of the zero point datas of blocks 1. Parameter
Defined, at which parameter the zero point data array starts

q171 Number zero point blocks work areas
Standard adjustment 1 (see P8555)

q172 Number of additional groups per zero point blocks
Defined, from many individual shifts itself the total shift together settings.
Standard adjustment 0

q173 Number of items (axes) per group
Number of entries per zero point. The zero point array always becomes for 7 points of zero created. Number of used parameters:

Number zero point blocks	(q171 of work areas)
* (number of groups +1)	(q172 + 1)
* number of items per group	(q173 axes)
* 7	(number of points of zero with q174 = 0, -)

Number of used parameters for zero point array

q174 Number of usable, programmable points of zero
0 or — (preset)
7 available points of zero G53..G59 (G153 always programmable)

1 G154..G159 additionally programmable
altogether 13 points of zero available G53..G59, G154..G159
2 G254..G259 additionally programmable
altogether 19 points of zero available G53..G59, G154..G159, G254..G25
:
9 G954..G959 additionally programmable
altogether 61 points of zero available G53..G59, G154..G159.... G954..G959

The zero point array is increased at additional points of zero:
Number of used parameters:

Number zero point-corrodes	(q171 of work areas)
* (number of groups + 1)	(q172+1)
* number of items per group	(q173 axes)
* number of points of zero	(q174*6 + 7)

Number of used parameters for null point array

8.3 Channel - descriptor (continued)

Zero point records q170... q174

Example: Zero point box with 2 work areas

q170	20000	at the beginning of the zero point record
q171	2	number of zero point blocks (work areas)
q172	2	number of additional groups per zero point block (shift)
q173	3	number of items (axes) per group
q174	-	number of usable, programmable zero points: G53... G59 = 7

Work area			sum shift	shift 1	shift 2
Work area 1	G53	1. Achse	P20000	P20021	P20042
		2. Achse	P20001	P20022	P20043
		3. Achse	P20002	P20023	P20044
	G54	1. Achse	P20003	P20024	P20045
		2. Achse	P20004	P20025	P20046
		3. Achse	P20005	P20026	P20047
	G55	1. Achse	P20006	P20027	P20048
		2. Achse	P20007	P20028	P20049
		3. Achse	P20008	P20029	P20050
	G56	1. Achse	P20009	P20030	P20051
		2. Achse	P20010	P20031	P20052
		3. Achse	P20011	P20032	P20053
	G57	1. Achse	P20012	P20033	P20054
		2. Achse	P20013	P20034	P20055
		3. Achse	P20014	P20035	P20056
	G58	1. Achse	P20015	P20036	P20057
		2. Achse	P20016	P20037	P20058
		3. Achse	P20017	P20038	P20059
	G59	1. Achse	P20018	P20039	P20060
		2. Achse	P20019	P20040	P20061
		3. Achse	P20020	P20041	P20062
Work area 2	G53	1. Achse	P20063	P20084	P20105
		2. Achse	P20064	P20085	P20106
		3. Achse	P20065	P20086	P20107
	G54	1. Achse	P20066	P20087	P20108
		2. Achse	P20067	P20088	P20109
		3. Achse	P20068	P20089	P20110
	G55	1. Achse	P20069	P20090	P20111
		2. Achse	P20070	P20091	P20112
		3. Achse	P20071	P20092	P20113
	G56	1. Achse	P20072	P20093	P20114
		2. Achse	P20073	P20094	P20115
		3. Achse	P20074	P20095	P20116
	G57	1. Achse	P20075	P20096	P20117
		2. Achse	P20076	P20097	P20118
		3. Achse	P20077	P20098	P20119
	G58	1. Achse	P20078	P20099	P20120
		2. Achse	P20079	P20100	P20121
		3. Achse	P20080	P20101	P20122
	G59	1. Achse	P20081	P20102	P20123
		2. Achse	P20082	P20103	P20124
		3. Achse	P20083	P20104	P20125

8.3 Channel - descriptor (continued)

Zero point records q170... q174

Example: Zero point box with extended zero points

q170	20000	at the beginning of the zero point record
q171	1	number of zero point blocks (work areas)
q172	3	number of additional groups per zero point block (shift)
q173	4	number of items (axes) per group
q174	1	number of usable, programmable zero points: G53 ... G59 and G154 ... G159 = 13

Work area 1			sum shift	shift 1	shift 2	shift 3
	G53	1. Achse	P20000	P20052	P20104	P20156
		2. Achse	P20001	P20053	P20105	P20157
		3. Achse	P20002	P20054	P20106	P20158
		4. Achse	P20003	P20055	P20107	P20159
	G54	1. Achse	P20004	P20056	P20108	P20160
		2. Achse	P20005	P20057	P20109	P20161
		3. Achse	P20006	P20058	P20110	P20162
		4. Achse	P20007	P20059	P20111	P20163
	G55	1. Achse	P20008	P20060	P20112	P20164
		2. Achse	P20009	P20061	P20113	P20165
		3. Achse	P20010	P20062	P20114	P20166
		4. Achse	P20011	P20063	P20115	P20167
	G56	1. Achse	P20012	P20064	P20116	P20168
		2. Achse	P20013	P20065	P20117	P20169
		3. Achse	P20014	P20066	P20118	P20170
		4. Achse	P20015	P20067	P20119	P20171
	G57	1. Achse	P20016	P20068	P20120	P20172
		2. Achse	P20017	P20069	P20121	P20173
		3. Achse	P20018	P20070	P20122	P20174
		4. Achse	P20019	P20071	P20123	P20175
	G58	1. Achse	P20020	P20072	P20124	P20176
		2. Achse	P20021	P20073	P20125	P20177
		3. Achse	P20022	P20074	P20126	P20178
		4. Achse	P20023	P20075	P20127	P20179
	G59	1. Achse	P20024	P20076	P20128	P20180
		2. Achse	P20025	P20077	P20129	P20181
		3. Achse	P20026	P20078	P20130	P20182
		4. Achse	P20027	P20079	P20131	P20183
	G154	1. Achse	P20028	P20080	P20132	P20184
		2. Achse	P20029	P20081	P20133	P20185
		3. Achse	P20030	P20082	P20134	P20186
		4. Achse	P20031	P20083	P20135	P20187
	G155	1. Achse	P20032	P20084	P20136	P20188
		2. Achse	P20033	P20085	P20137	P20189
		3. Achse	P20034	P20086	P20138	P20190
		4. Achse	P20035	P20087	P20139	P20191
	G156	1. Achse	P20036	P20088	P20140	P20192
		2. Achse	P20037	P20089	P20141	P20193
		3. Achse	P20038	P20090	P20142	P20194
		4. Achse	P20039	P20091	P20143	P20195
	G157	1. Achse	P20040	P20092	P20144	P20196
		2. Achse	P20041	P20093	P20145	P20197
		3. Achse	P20042	P20094	P20146	P20198
		4. Achse	P20043	P20095	P20147	P20199
	G158	1. Achse	P20044	P20096	P20148	P20200
		2. Achse	P20045	P20097	P20149	P20201
		3. Achse	P20046	P20098	P20150	P20202
		4. Achse	P20047	P20099	P20151	P20203
	G159	1. Achse	P20048	P20100	P20152	P20204
		2. Achse	P20049	P20101	P20153	P20205
		3. Achse	P20050	P20102	P20154	P20206
		4. Achse	P20051	P20103	P20155	P20207

8.3 Channel - descriptor (continued)

Parameters for channel - descriptor 1

q175 Beginning of the tool data blocks 1. parameter
Defines, from which parameter on the tool data array starts.

q176 Number of tool data blocks
Number of elements of the tool data arrays.

q177 Number of Sub datas of blocks
--, 0, 1 it exists to only 1 main data of blocks
> 1 number of Sub datas of blocks

Lengthens a Sub data of blocks: (q178 - 8) / q177

q178 Number of parameters of a tool kit
Total length of an item in the tool data array.
(inclusive all Sub data)

q179 Number of tool places available in the tool magazine

8.3 Channel - descriptor (continued)

Example of tool data pool

Without Sub datas of blocks

q175	21000
q176	99
q177	0
q178	20

Tool datas T1

P21000	tool group number T1	
P21001	tool number	(current numbering, starting with 1)
P21002	tool workstation in the magazine	
P21003	tool Sub data of blocks	(selected)
P21004	type of tool	

P21006 service life command
P21007 service life actual

P21008 service life status

0 or —	service life of OK ONE
1	service life run

P21009 type of service life
P21010 radius of the tool
P21011 length of the tool
P21012 radius correction
P21013 length correction
P21014 type of tool (active quadrant 0..9)
P21015 customer data etc..
:
P21019

8.3 Channel - descriptor (continued)

Example of tool data pool

Tool datas T2

P21020 tool group number T2
P21021 tool number
P21022 tool workstation in the magazine
P21023 tool Sub data of blocks
P21024 type of tool

P21026 service life command
P21027 service life actual

P21028 service life status
0 or — service life of OK ONE
1 service life run

P21029 type of service life
P21030 radius of the tool
P21030 radius of the tool
P21031 length of the tool
P21032 radius correction
P21033 length correction
P21034 type of tool (active quadrant 0..9)
P21035 customer data etc..

P21039

Tool datas T3

P21040 Group of tools of number T3

8.3 Channel - descriptor (continued)

Example of tool data pool

with 3 Sub datas of blocks

q175	21000	at the beginning of the tool datas
q176	20	number of tool data blocks
q177	3	number of Sub data blocks
q178	38	size of a data of blocks

Service life commandist is situated in the simultaneous data area!
Being this required its, thus that each Sub data block receives its own service life, do not have the service life data in customers specific area to be held.

Tool datas T1

* - Simultaneous data area

P21000	* tool group number T1	
P21001	* tool number	(current numbering, starting with 1)
P21002	* tool workstation in the magazine	
P21003	* tool Sub data of blocks (selected)	
P21004	* type of tool	
P21006	* service life command tool-spreading!!	
P21007	* service life actual tool-spreading!!	
P21008	service life status	
	0 or —	service life of OK ONE
	1	service life run
P21009	type of service life	
P21010	radius of the tool	
P21011	length of the tool	
P21012	radius correction	
P21013	length correction	
P21014	type of tool	(active quadrant 0..9)
P21015	customer data etc..	

8.3 Channel - descriptor (continued)

Example of tool data pool

SUB data of blocks 2 / T1

P21018	service life status	
	0 or —	service life of OK ONE
	1	service life run

P21019	reserves
P21020	radius of the tool
P21021	length of the tool
P21022	radius correction
P21023	length correction
P21024	type of tool (active quadrant 0..9)
P21025	customer data etc..

SUB data of blocks 3 / T1

P21028	service life status	
	0 or —	service life of OK ONE
	1	service life run

P21029	reserves
P21030	radius of the tool
P21031	length of the tool
P21032	radius correction
P21033	length correction
P21034	type of tool (active quadrant 0..9)
P21035	customer data etc..

8.3 Channel - descriptor (continued)

Example of tool data pool

Tool datas T2

P21038	tool group number T2
P21039	tool number (e.g.: current numbering)
P21040	tool workstation in the magazine
P21041	tool Sub data of blocks (selected)
P21042	reserves
P21043	reserves
P21044	reserves
P21045	reserves

SUB data of blocks 1 / T2

P21046	service life status	
	0 or —	service life of OK ONE
	1	service life run
P21047	reserves	
P21048	radius of the tool	
P21049	length of the tool	
P21050	radius correction	
P21051	length correction	
P21052	type of tool (active quadrant 0..9)	
P21053	customer data etc..	

SUB data of blocks 2 / T2

P21056	service life status	
	0 or —	service life of OK ONE
	1	service life run)
P21057	reserves	
P21058	radius of the tool	
P21059	length of the tool	
P21060	radius correction	
P21061	length correction	
P21062	type of tool (active quadrant 0..9)	
P21063	customer data etc..	

8.3 Channel - descriptor (continued)

Example of tool data pool

SUB data of blocks 3 / T2

P21066	service life status	
	0 or —	service life of OK ONE
	1	service life run
P21067	reserves	
P21068	radius of the tool	
P21069	length of the tool	
P21070	radius correction	
P21071	length correction	
P21072	type of tool	(active quadrant 0,,9)
P21073	customer data etc..	

Tool datas T3

P21076 tool group number T3

8.3 Channel - descriptor (continued)

Configuration - Example

Example:	Amount of channels	: 1	
	Amount of axes	: 3	(X, Y, Z1)

q2	1	channels	
q101	1	channel number	
q102	10000	parameter - relocator user block 1	(q number)
q103	5000	number of channel parameters in the 1st. block	
q104	18000	parameter - relocator user block 2	(q number)
q105	2000	number of channel parameters in the 2. block	
q110	$01010058	1st. logical axis = 1. physical axis, axis name ‘X’, key 1	
q111	$12020059	2nd. logical axis = 2. physical axis, axis name ‘Y’, key 18	
q112	$0303005A	3rd. logical axis = 3. physical axis, axis name ‘Z’, key 3	
q170	20000	beginning of the zero point data block	
q171	1	number of zero point blocks	
q172	0	number of groups per zero point block	
q173	32	number of elements per group	
q175	21000	beginning of the tool data block	
q176	50	number of tool data blocks	
q177	1	number of sub data blocks	
q178	20	number of elements per data block	

Parameters for this example are lying for this example as follows:

	q0 ... q99	general system configuration
	q100 ... q999	definition channel descriptors
P11000 ... P11999	q1000 ... q1999	system overlapping data
P12000 ... P13599	q2000 ... q3599	axis data
P0 ... P4999	q10000 ... q14999	customer parameter block 1
P7000 ... P9999	q15000 ... q17999	fix defined channel parameters
P20000 ... P21999	q18000 ... q19999	for zero pointes, tool data

8.4 Modes of operation - groups

Operating mode network 1

The following parameters contain bit information for suitable channel selection.

q903 Operating mode switching in the network
Bit information: bit 0 channel 1
bit 1 channel 2
:
e.g.: $00000003: Channel 1 & channel 2
(P8683 in the defined channels is simultaneous settinged)

With the describing of q903 operating mode switching in the network becomes changes immediately.

q904 Stop in the network (see q903)
Stop in the network works only for STOP key
i.e. if a channel over STOP key is stopped, become the remaining channels in network also stopped.
Stop in the network does not work with stop by messages, measuring stops, Program end etc.
With the describing of q904 stop in the network is changed immediately.

q905 Abort in the network (see q903)
With the describing of q905 abort in the network is changed immediately.

q908 Message display in the network
(messages of the CNC)

Byte 1	Messages system and channel-specific messages of the PLC
Byte 2	Messages High Prio.
Byte 3	Messages Low Prio.
Byte 4	Messages display and channel-specific messages of the PLC

All bytes contain the following
Bit information bit 0 channel 1
bit 1 channel 2
:
bit 7 channel 8

Example q908 $00030303
i.e.. Messages that priority system / High Prio. / Low Prio.
in both channels (K1 + K2) are displayed.
The display messages are displayed only in the selected channel (byte 4 = 0).

8.4 Modes of operation - groups (continued)

Operating mode network 1

q909 PLC message display in the network
(messages of the PLC)

Byte 1	PLC Messages High Prio.
Byte 2	PLC Messages Low Prio.

see further q908

q910 Operating mode network 2
:
q919

q920 Operating mode network 3
:
q929

q930 Operating mode network 4
:
q939

8.5 Standard channel parameters

Milling cycles
(only if the milling cycles G71 / G72 / G73 / G74 / G75 are used, otherwise freely available)

P11	Pocket dimension in x-direction
P12	Pocket dimension in y-direction
P13	Pocket depth in z-direction
P14	Corner radius
P15	Allowance on outline in x and y-direction
P16	Setting measure in x or y-direction (sign + / -)
P17	Setting measure in z-direction
P18	Allowance on pocket depth in z-direction
P19	Safety margin in z-direction
P21	Feed in z-direction (when immersing into the material)

8.5 Standard channel parameters (continued)

Drill cycles
(only if the drill cycles G81 / G83 / G84 / G85 are used, otherwise freely available)

P30	Free cutting time	
P31	Balance fodder	G84
	3 thread cutting without balance fodder on the right	
	4 thread cutting without balance fodder links	
	0 or -- G84 standard cycle (with balance fodder)	
P32	Drill feed	
P32	Thread pitch	G84
P33	Drilling depth	
P33	Hread depth	G84
P34	Anticipation plane	Safety margin of the processing upper edge
P35	Retreat plane	To running around obstacles in the setting axis
P36	Stroke rate	G83, constant setting depth = (P33 - P34) / P36
P36	Setting depth	G83, degressive setting depth
P37	1. Stroke	G83, degressive setting depth
P38	Safety margin	G83, default = 1mm
P39	Gradual decrease	G83, degressive setting depth
	1 on	
	0 or -- off	

8.5 Standard channel parameters (continued)

Cycle patterns
(only if the cycle patterns G86 / G87 / G88 / G89 are used, otherwise freely available)

G86 vector type processing

P100	Coordinate of the 1. Axis (x)
P101	Coordinate of the 2. Axis (y)
P103	Vector bracket related to the 1.Achse (x)
P104	Vector length
P105	Vector splitting
P106	Number of points of positioning

G87 scope handling of a parallelogram

Vector 1	Vector 2	
P100	P110	Coordinate of the 1. Axis (x)
P101	P111	Coordinate of the 2. Axis (y)
P103	P113	Vector bracket related to the 1.Achse (x)
P104	P114	Vector length
P105	P115	Vector splitting
P106	P116	Number of points of positioning

G88 grid handling

Vector 1	Vector 2	
P100	P110	Coordinate of the 1. Axis (x)
P101	P111	Coordinate of the 2. Axis (y)
P103	P113	Vector bracket related to the 1.Achse (x)
P104	P114	Vector length
P105	P115	Vector splitting
P106	P116	Number of points of positioning

G89 circle handling

P120	Coordinate of the 1. Axis (x) of the pitch diameter centre point
P121	Coordinate of the 2. Axis (y) of the pitch diameter centre point
P122	Circle diameter
P123	Start bracket
P124	Travel bracket
P125	Vector part
P126	Number of points of positioning

8.5 Standard channel parameters (continued)

Measuring cycle
(only if the measuring cycle G69 are used, otherwise freely available)

P140	Cycle selection	
P141	Safety margin	[mm]
P143	Measuring rate	[mm/min]
P144	Number of measurements at the same point (> 0)	[n]
P147	Pointer on the measuring tolerance data	
(P147)+4	Confidence interval	[mm]
P148	Measuring axis in plane system (1...3)	
P149	Pointer on the data of point of trigger	
(P149)+0	XN, point of trigger in negative direction 1. Axis	[mm]
(P149)+1	XP, point of trigger in positive direction 1. Axis	[mm]
(P149)+2	YN, point of trigger in negative direction 2. Axis	[mm]
(P149)+3	YP, point of trigger in positive direction 2. Axis	[mm]
(P149)+4	ZN, point of trigger in negative direction 3. Axis	[mm]
(P149)+5	ZP, point of trigger in positive direction 3. Axis	[mm]
P150	Pointer on the calibrationing drilling data	
(P150)+0	Position of the calibrationing drilling centre point 1. Axis (i)	[mm]
(P150)+1	Position of the calibrationing drilling centre point 2. Axis (j)	[mm]
(P150)+2	Position of the calibrationing drilling centre point 3. Axis (k)	[mm]
(P150)+3	Calibrationing drilling diameter	[mm]
P152	Expected drilling diameter	[mm]
P154	Pointer on the measuring data	
(P154)+0	Measuring position 1. Axis	[mm]
(P154)+1	Measuring position 2. Axis	[mm]
(P154)+2	Measuring position 3. Axis	[mm]

8.5 Standard channel parameters (continued)

Channel parameters

P70	Graphical programming environment with phase Input leg length	(G12)	[mm]
P71	Graphical programming environment with radius Input radius	(G12)	[mm]
P160	U_0 Origin axis 1 of the system UV (Rotating 45) Parameter is activated in the block with G45.		[mm]
P161	V_0 Origin axis 1 of the system UV (Rotating 45) Parameter is activated in the block with G45.		[mm]
P163	E Rotating angle Content is activated in the block with G45. Conversion of the coordinates of the UV-system into coordinates of the XY-system Xb = (U*cosE - V*sinE) + U0 Yb = (U*sinE + V*cosE) + V0 see also G45/G56		[degree]
P165	U additive shift axis 1 (rotation G45 planned)		[mm]
P166	V additive shift axis 2 (rotation G45 planned)		[mm]
P168	E incrementing of the rotating angle (planned)		[degree]

8.5 Standard channel parameters (continued)

System internally parameter

P7000 Parameter area is used by the system
.
.
P7999

	Offset
- Measuring position	0 ... 31
- Display positions	32 ... 63
- Program positions	64 ... 95
- Delta positions	96 ... 127
- Display shifts	128 ... 159

Zoom

P7900 Zoom shot function
(P7900 with each system start is reset)

0	OFF	
1	zoom shot ON	affects machine coordinates, indicator position is not influenced
2	zoom shot ON	affects programmed coordinates

P7901 zoom shot factor axis 1
P7902 zoom shot factor axis 2
:
P7932 zoom shot factor axis 32

8.5 Standard channel parameters (continued)

ZSM control parameter

P7950 ZSM command

0 in the case of current measurement, the measurement is aborted.
1 start ZSM measurement
-> P7960 to 1 is settinged
-> if P7960 = 10 measurement terminates error free.

Internal operational sequence with the measuring start:
With the ZSM command ' start ZSM measurement ' becomes with the suitable Measuring axis (P7951) the ZSM Messaufzeichnung activates.
In accordance with the defined measuring logic of this measuring axis (P12045, byte 4), becomes with the switch edge defined there the ZSM recording started. The first edge defines a gap/tooth change.

Example: P12045: $0 xx xx xx switch edge positively effectively
i.e.. The measured value recording starts with a positive switch edge at the sensor input (0/24V-Sprung).
A positive edge is detected as gap/tooth change —> beginning of the tooth.
A negative edge is detected as tooth/gap change —> beginning of the gap.

P7951 Axis number
Contents log. Axis number (1... 32)

P7952 Number of teeth which can be measured (max. 250)
With (P7952 <= 3) the BWO ZSM Korrekturalgorithmus is not used.
In this case the measured value results over the arithmetic means.
With (P7952 = 1) P7962 and P7963 are reset after the measuring analysis

8.5 Standard channel parameters (continued)

ZSM control parameter

P7953 ZSM mode

0 or — Measuring analysis of the CNC System (default)
CNC System leads measured value recording and measuring analysis off.
With success P7960:10 is settinged

Measuring recording
P7960: 5

Measuring analysis
in the CNC
P7960: 10

1 Measuring analysis only externally
CNC System leads measured value recording off.
With success P7960:8 is settinged
external measuring analysis (CNC-DLL, NC program.) becomes with P7960=8 triggered and ZSM status describes P7960:10 with success.

Measuring recording
P7960: 8

external
Measuring analysis
P7960: 10

2 Measuring analysis internally and externally
CNC System executes measured value recording and measured value analysis. With error free operational sequence P7960:8 is settinged.
external measuring analysis (CNC-DLL, NC program.) becomes with P7960=8 triggered and ZSM status describes P7960:10 with success.

Measuring recording
P7960: 5

Measuring analysis
in the CNC
P7960: 8

external
Measuring analysis
P7960: 10

8.5 Standard channel parameters (continued)

ZSM Statusparameter

P7954 Correction window

[%]

0...100
Here the ZSM Korrekturalgorithmus can be influenced,
100 ZSM Korrekturalgorithmus ineffectivy.
0 ZSM Korrekturalgorithmus fully effectively
if 0 or -- 80

P7955 Pointer on ZSM measuring data array
Contents:
0 or -
> 0 Measuring data after the measured value recording additionally stored in parameters. In the Messdatem array there are no modulo jumps.
< 0 Measuring data after the measured value recording additionally stored in parameters. In the Messdatem array there are modulo jumps with round axes.

Example

P7955	100
P100	Number of following measured values
P101	1. Measured value
P102	2. Measured value
:	

8.5 Standard channel parameters (continued)

ZSM Statusparameter

P7960 ZSM status
Parameter may not be described.

1 measured value recording runs
5 measuring analysis runs
10 measurement terminates, measured value error free valid (P7950 reaches 0)
> 10 measurement incorrectly or aborted
20 measurement aborted
21 no edges were detected (message of the Interpolator)
22 division = 0 determines
23 fewer than 1 tooth per rotation
24 measuring recording not correctly, missing measured values in the data array
25 measuring recording not correctly, measured values not continuously up/down-rising
26 no measured values available
30 incorrect Input, P7951 / P7952 check
> 100 customized messages during external measuring analysis

P7961 Measured value 1 [°]
Tooth center
Area 0... measured division
Parameter is reset with start ZSM measurement

P7962 Measured value 2 [°]
Gap center
Area: P7962 > P7961
Parameter with start ZSM measurement is reset
With number of teeth = 1 (P7952 = 1) P7962 remains reset

P7963 Measured value 3 [°]
measured division
Parameter with start ZSM measurement is reset
With number of teeth = 1 (P7952 = 1) P7963 remains reset

P7964 Measured value 4 [number of teeth]
Number of teeth on 360Grad
Parameter with start ZSM measurement is reset

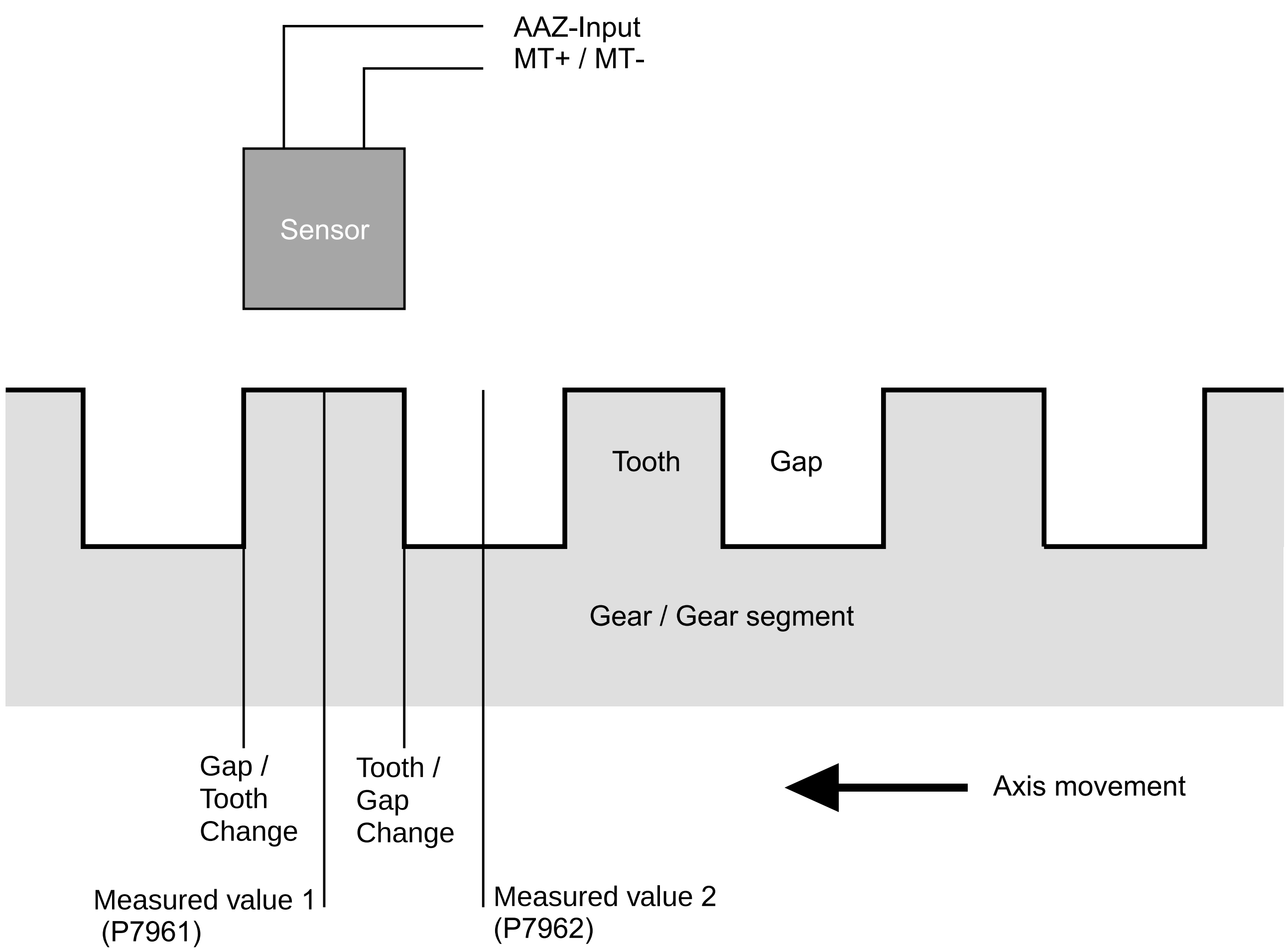
P7965 Measured value 5 [°]
measuring away
Distance: At the beginning of first tooth to end final tooth
Worth actual signed!
Parameter with start ZSM measurement is reset

8.5 Standard channel parameters (continued)

ZSM Statusparameter

Example of a measuring flow in the NC program

N100 P7950:1	{ start measurement }
N110 M1	{ loop re-entry point
N120 P7960<10.110	{ control room, to measurement terminates or aborted }
N130 P7960=10.200	{ branch, if measurement of OK ONE }
N140...	{ error handling, measurement aborted or errors }
:	
N200...:P7961	{ measured value process }
:	



8.5 Standard channel parameters (continued)

Channel parameters

P8000 Axis shifts
:
P8031

P8000	Shift axis 1
P8001	Shift axis 2
P8002	Shift axis 3
P8003	Shift axis 4
:	
P8031	Shift axis 32

8.5 Standard channel parameters (continued)

Tool - Data - Interface

P8040 T read in locking
0 or -- T read in locking is open
1 T read in locking is set
When a locking is set, it is prevented that the realtime data block (P8200...) is written by the system.

If the read in locking is set and 'T' is programmed in the NC block, the NC program run is stoped as long as the T read in locking is set.

T read in locking is reset:
- at a NC program start
- when copying the real time data block into the spindle data block (P8046 : 2).

P8041 Tool change BLOCKING in MANUAL mode
content : 0,—: Allows tool change via P8043
1 : Tool change via P8043 NOT possible

P8042 Mapping Function for tool interface
Byte 1 : P8050..P8099 from channel n
Byte 2 : P8100..P8149 from channel n
Byte 3 : P8150..P8199 from channel n
Byte 4 : P8200..P8249 from channel n
The mapping is active immediately in writing of the parameter.

P8043 Tool change in manual mode
(See release P8041)
Tool data block into the PLC tool Data of blocks (P8050... P8099) and in real time data of blocks (P8200...) charge.
see also 8044

Example: P8043:2
In the tool data array the tool group number 2 is looked up.
Actual the search successfully, becomes the suitable data of blocks after P8050... and P8200... copies.
With T0 one zeroes.
If the data of blocks is not found, appears a message.
Subsequently, into the autooperation is changed and Z6 is called.
at the end of Z6 to the manual mode one shifts back.
If Z6 does not become available actual, instead of the program call (Z6) M6 to the PLC transmitted.

With active tool change DLL command 1 is transmitted to the DLL, before Z6 on one calls.

8.5 Standard channel parameters (continued)

Tool - Data - Interface

P8044 Tool Sub data of blocks number with tool change in the HAND operation (P8043)
P8044 must be described before P8043!

Contents: -, 0,	1	Main data block / 1. Sub data block
	2	2. Sub data block
	3	3....

P8045 Tool datas take up

Byte 1	CMD
Byte 2	CMD Extention

CMD 1
Thus the PLC tool Data of blocks (P8050..) becomes into the spindle tool data of blocks (P8100..) and the offsetting data of blocks (P8150..) copy.
Contents of CMD Extention actual all the same.

CMD 2
Thus the real time data of blocks (P8200..) becomes into the spindle tool data block (P8100..) copied.
Additionally t-read in-barrier is reset. (P8040:0)
With CMD Extention: Channel number of the real time of data of blocks.

0 or —	Umkopieren within the current channel
1..8	Umkopieren channel-spreading Real time data of blocks 1..8. Channel (CMD Extention) becomes in spindle tool data block (P8100..) of the current channel copies. the current channel is in the operating mode HAND becomes additional the spindle tool data block (P8100..) into the offsetting data of blocks (P8150..) copied. The tool data set array is set to zero when the corresponding tool location tool.

CMD 3
Thus the spindle tool data block (P8100..) of the active channel becomes into the spindle tool data block (P8100..) in CMD Extention defined Channel umkopiert.

CMD 10
Thus the PLC tool Data of blocks (P8050..) becomes into the Spindelwerkzeug-data of blocks (P8100..) copies.

8.5 Standard channel parameters (continued)

Tool - Data - Interface

P8046 Tool datas copy over

1 Thereby the spindle tools record (P8100..) becomes in offsetting record (P8150) copies.

2 Thereby the PLC becomes tool record (P8050...) into that offsetting record (P8150) copies.

P8047 Tool datas placings

1 Thereby the spindle tool data block (P8100..) becomes in the suitable tool data block array back-stored and afterwards the actual spindle tool data block reset.

If P8100 = 0 or -- (no tools in the spindle) this procedure becomes leave immediately.
With active tool change DLL only command 2 is transmitted to the DLL.

P8048 Tool group number (t-paragraph from NC program)
(parameter to the real time is described)
The parameter is described only if ' T ' is programmed in the NC block

P8049 Info NC block contains tool-controlling m-function

0	no tool-controlling m-function
6	M06 in the block
16	M16 in the block

P8049 is described in the advance, if T is programmed in the NC block
P8049 is updated, before the parameter block P8050... is described.

8.5 Standard channel parameters (continued)

Tool - Data - Interface

P8050 .. P8099

current PLC tool data (Interface to PLC)
Here are the tool data that has been transferred to the PLC.
This tool data are loaded in the flow, with the tool data according
to the programmed in the NC program T.
(see also P8049)

special case:
If the tool group number (T number) is equal to the tool group number of
the spindle data, the PLC tool data is not loaded from the tool data array,
but the spindle-record in the PLC-Tool record copied.

8.5 Standard channel parameters (continued)

Tool - Data - Interface

Organization of the tool datas

The type of tool system (Offset=4) in the tool data block determines, how tool length work:

Type of tool system (Offset=4)
-, 0.. 99 (milling machine)
500,,599 (centre lathe)

System type Milling machine
Total of tool length (P8161) and werkzeuglaengen-Korrektur (P8163) work in the axis, which is vertically on the current processing level.

Example
Standard axis allocation
1. 2. 3.
G17 X Y Z length works in Z-axis
G18 Z X Y length works in Y axis
G19 Y Z X length works in x axis
— processing level

Tool data block organization
Offset
0 tool group number (t-NR)
1 tool number
2 tool workstation in the magazine
3 tool Sub data of blocks
4 type of tool system (0,,99 = milling machine)
5 reserves
6 service life command
7 service life actual
8 service life status (0 or — service life of OK , 1 service life run)
9 type of service life
10 radius of the tool
11 length of the tool
12 radius correction
13 length correction
14 type of tool (active quadrant 0,,9)
15 customer data etc.. <— the customized area starts here
..
49

8.5 Standard channel parameters (continued)

Tool - Data - Interface

Type of system centre lathe
Total of tool length 1 (P8161) and tool length 1 correction (P8163)
===> works in the 2. even axis,
Total of tool length 2 (P8165) and tool length 2 correction (P8166)
===> works in the 1. even axis,

Example
Standard axis allocation
1. 2. 3.
G17 X Y Z length 1 works in Y-axis, length 2 works in X-axis

G18 Z X Y length 1 works in X-axis, length 2 works in Z-axis

G19 Y Z X length 1 works in Z-axis, length 2 works in Y-axis
— processing level

Tool data block organization
Offset
0 tool group number (t NR)
1 tool number
2 tool workstation in the magazine
3 tool Sub data of blocks
4 type of tool system (500.. 599 = centre lathe)
5 reserves
6 service life command
7 service life actual
8 service life status (0 or — service life of OK ONE, 1 service life run)
9 service life type
10 radius of the tool
11 length 1 of the tool
12 radius correction
13 length 1 correction
14 type of tool (active quadrant 0,,9)
15 length 2 of the tool
16 length 2 correction
17 customer data etc.. <— the customized area starts here
..
49

8.5 Standard channel parameters (continued)

Tool - Data - Interface

P8050	tool group number (t-NR) P8050 is not transferred during the restarting simulation (P8686=1) to the PLC
P8051	tool number, current numbering, starting with 1 When describing the PLC tool Data of blocks becomes of the system P8051 with serial-number of the data of blocks in the tool data array described. With the placing of the data of blocks (P8047) this serial-number becomes addressing in the tool data array uses.
P8052	tool workstation in the magazine
P8053	tool Sub data of blocks --, 0, 1 main data of blocks / 1. Sub data block 2 2. Sub data block....
P8054	type of tool system
P8055	reserves
P8056	service life command
P8057	service life actual
P8058	service life status 0 or — service life of OK ONE 1 service life run
P8059	service life type
P8060	radius of the tool
P8061	length of the tool
P8062	radius correction
P8063	length correction
P8064	type of tool (active quadrant 0... 9)
P8065	user data <- - here the user specific area starts
:	
P8099	

8.5 Standard channel parameters (continued)

Tool - Data - Interface

Actual spindle tool data block

P8100 Actual spindle tool data block.
: The data block refers to the actual tool in the spindle.
P8149 This data block is loaded from the tool data array and when laying back the tool stored again into the array.

P8100	Tool group number	(T - number)
P8101	Tool number	(e.g. continuous numbering)
P8102	Tool place in the magazine	
P8103	Tool sub data block	(selected)
P8104	type of tool system	
P8105	reserves	
P8106	service life command	
P8107	service life actual	

P8108 service life status
0 or — service life of OK ONE
1 service life run

P8109 service life type

P8110 Tool radius
P8111 Tool length
P8112 Radius correction offset
P8113 Length correction offset
P8114 Tool type (active quadrant 0...9)
P8115 Customer data <— Here the customer specific area begins
:
P8149

8.5 Standard channel parameters (continued)

Tool - Data - Interface

Actual data block

P8150	Actual data block.
:	This is the data block, which is calculated into the NC program.
P8199	From this data block, it can not be stored back to the data array.

P8150	Tool group number	(T - number)
P8151	Tool number	(e.g. continuous numbering)
P8152	Tool place in the magazine	
P8153	Tool sub data block	(selected)
P8154	type of tool system	
P8155	reserves	
P8156	service life command	
P8157	service life actual	

P8158 service life status

0 or —	service life of OK ONE
1	service life run

P8159 service life type

P8160	Tool radius
P8161	Tool length
P8162	Radius correction offset
P8163	Length correction offset
P8164	Tool type (active quadrant 0...9)
P8165	Customer data <— Here the customer specific area begins
:	
P8199	

8.5 Standard channel parameters (continued)

Tool - Data - Interface

Real time data block

This is the data block that is loaded from the system to the real time:
- T programmed in the NC block
- T M6 programmed in the NC block (is not written at T M16)

Special case:
If the tool group number (T number) is the same like the tool group number of a spindle data block, the PLC tool data are not loaded from the tool data array, but the spindle data block is copied into the PLC tool data block.
See also T read in locking P8040

With external tool administration, the real time data block is loaded with calling up command 3.

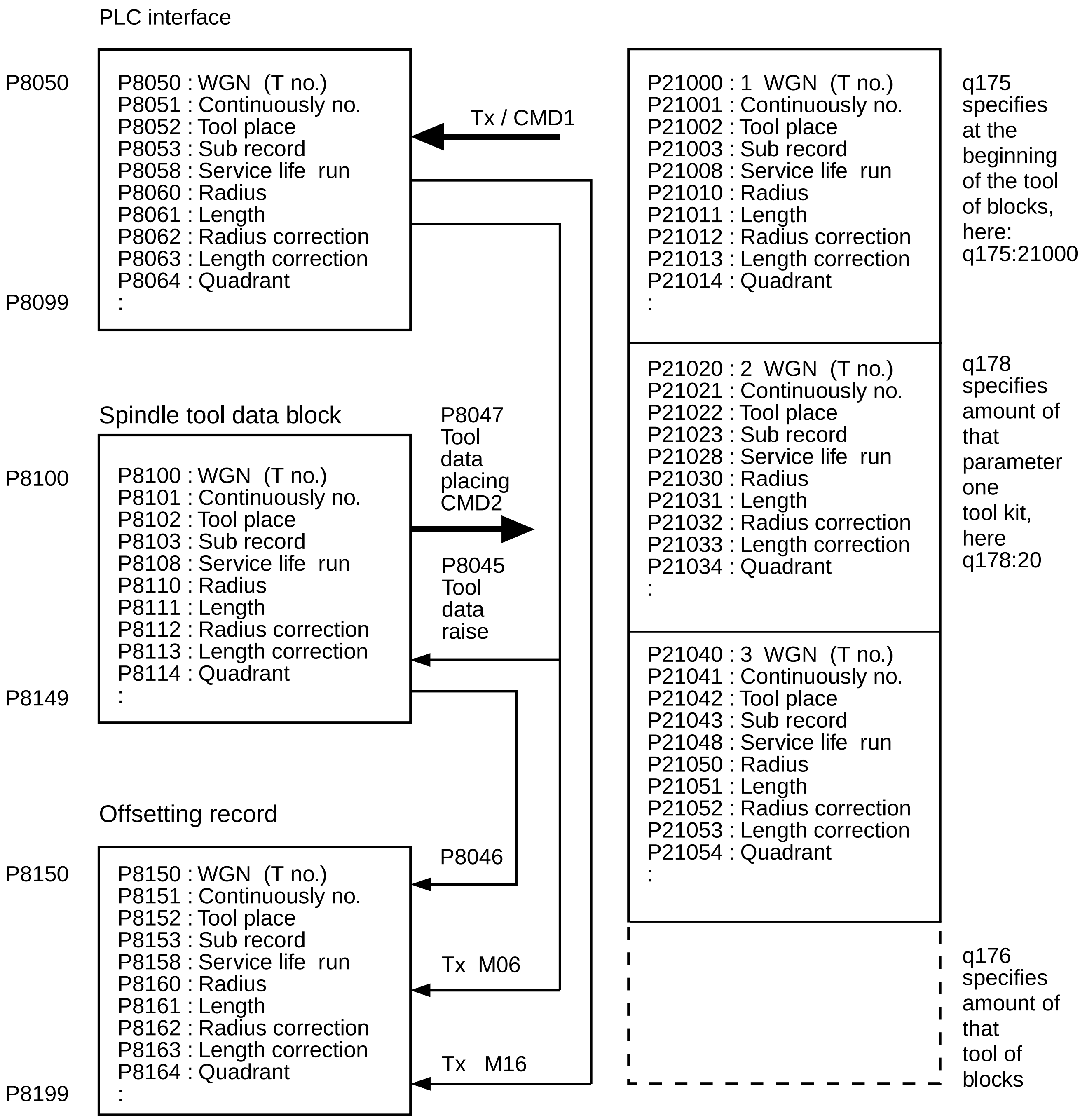
P8200	Tool group number	(T number)
P8201	Tool number	(e.g. continuous numbering)
P8202	Tool place in the magazine	
P8203	Tool sub data block	(selected)
P8204	Type of tool system	
P8205	eserves	
P8206	Service life command	
P8207	Service life actual	
P8208	Tool life status 0 or -- tool life o.k., 1 tool life over	
P8209	Service life type	
P8210	Tool radius	
P8211	Length of the tool	
P8212	Radius correction	
P8213	Length correction	
P8214	Tool type	(acitve quadrant 0..9)
P8215	Customer specific area	
.		
.		
P8249		

8.5 Standard channel parameters (continued)

Tool data blocks in the overview

Tool data interface
specifies firmly

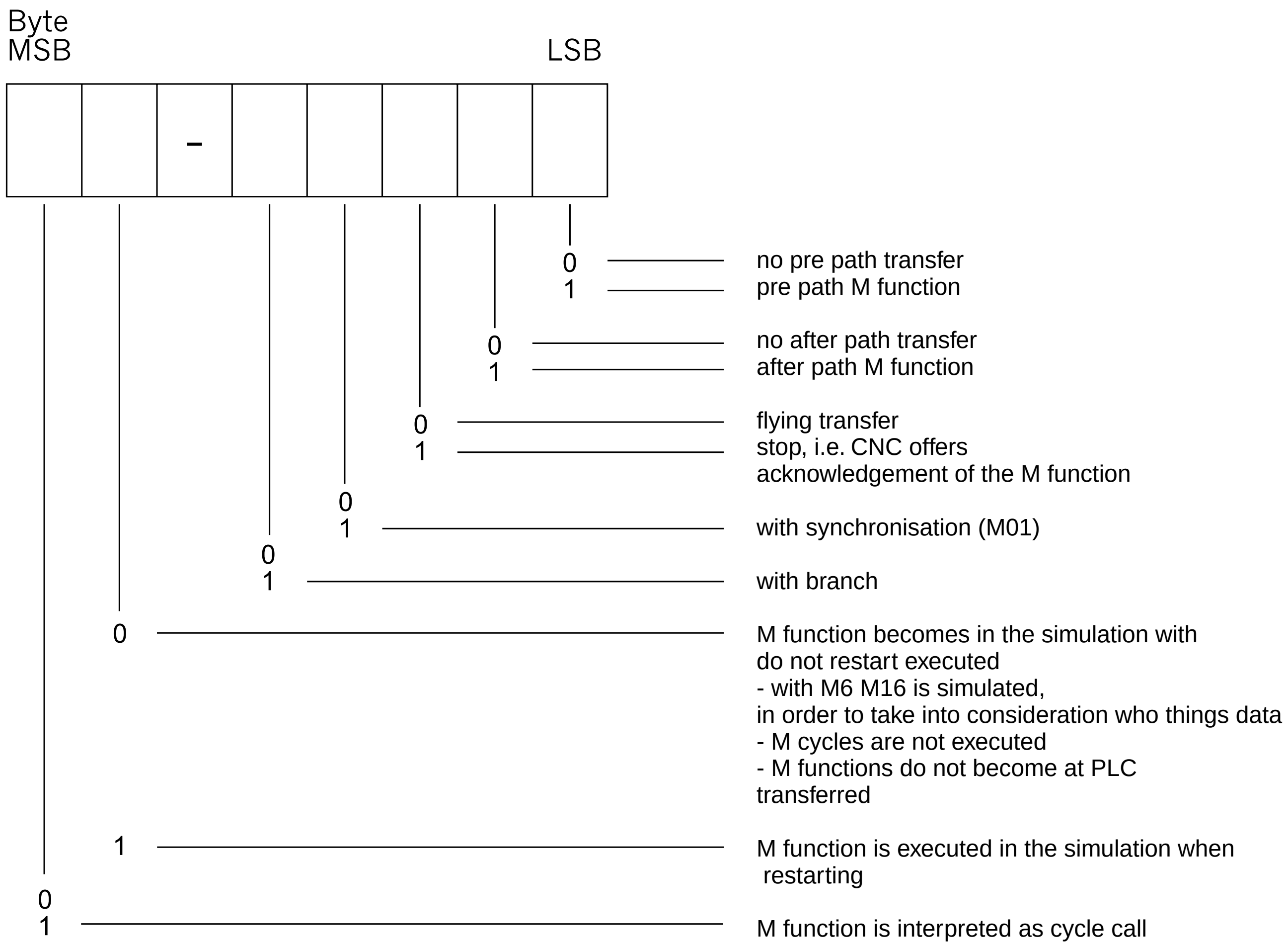
Tool data array
freely definably



8.5 Standard channel parameters (continued)

Channel specific definition of the M functions

Information for a M - function



P8250 Channel-specific definition of the M functions
see table starting from P11050, see also q158
Here 1000 channel-specific M functions can be specified.
Note: A modification in P8250.. becomes with a operating modes changes
by HAND after AUTO effectively.

8.5 Standard channel parameters (continued)

Definition of the M functions

P8250	M	03	02	01	00
P8251	M	07	06	05	04
P8252	M	11	10	09	08
P8253	M	15	14	13	12
P8254	M	19	18	17	16
P8255	M	23	22	21	20
P8256	M	27	26	25	24
P8257	M	31	30	29	28
P8258	M	35	34	33	32
P8259	M	39	38	37	36
P8260	M	43	42	41	40
P8261	M	47	46	45	44
P8262	M	51	50	49	48
P8263	M	55	54	53	52
P8264	M	59	58	57	56
P8265	M	63	62	61	60
P8266	M	67	66	65	64
P8267	M	71	70	69	68
P8268	M	75	74	73	72
P8269	M	79	78	77	76
P8270	M	83	82	81	80
P8271	M	87	86	85	84
P8272	M	91	90	89	88
P8273	M	95	94	93	92
P8274	M	99	98	97	96
P8275	M	103	102	101	100
P8276	M	107	106	105	104
P8277	M	111	110	109	108
P8278	M	115	114	113	112
P8279	M	119	118	117	116
P8280	M	123	122	121	120
P8281	M	127	126	125	124
P8282	M	131	130	129	128
P8283	M	135	134	133	132
P8284	M	139	138	137	136
P8285	M	143	142	141	140
P8286	M	147	146	145	144
P8287	M	151	150	149	148
P8288	M	155	154	153	152
P8289	M	159	158	157	156
P8290	M	163	162	161	160
P8291	M	167	166	165	164
P8292	M	171	170	169	168
P8293	M	175	174	173	172
P8294	M	179	178	177	176
P8295	M	183	182	181	180
P8296	M	187	186	185	184
P8297	M	191	190	189	188
P8298	M	195	194	193	192
P8299	M	199	198	197	196

P8300	M	203	202	201	200
P8301	M	207	206	205	204
P8302	M	211	210	209	208
P8303	M	215	214	213	212
P8304	M	219	218	217	216
P8305	M	223	222	221	220
P8306	M	227	226	225	224
P8307	M	231	230	229	228
P8308	M	235	234	233	232
P8309	M	239	238	237	236
P8310	M	243	242	241	240
P8311	M	247	246	245	244
P8312	M	251	250	249	248
P8313	M	255	254	253	252
P8314	M	259	258	257	256
P8315	M	263	262	261	260
P8316	M	267	266	265	264
P8317	M	271	270	269	268
P8318	M	275	274	273	272
P8319	M	279	278	277	276
P8320	M	283	282	281	280
P8321	M	287	286	285	284
P8322	M	291	290	289	288
P8323	M	295	294	293	292
P8324	M	299	298	297	296
P8325	M	303	302	301	300
P8326	M	307	306	305	304
P8327	M	311	310	309	308
P8328	M	315	314	313	312
P8329	M	319	318	317	316
P8330	M	323	322	321	320
P8331	M	327	326	325	324
P8332	M	331	330	329	328
P8333	M	335	334	333	332
P8334	M	339	338	337	336
P8335	M	343	342	341	340
P8336	M	347	346	345	344
P8337	M	351	350	349	348
P8338	M	355	354	353	352
P8339	M	359	358	357	356
P8340	M	363	362	361	360
P8341	M	367	366	365	364
P8342	M	371	370	369	368
P8343	M	375	374	373	372
P8344	M	379	378	377	376
P8345	M	383	382	381	380
P8346	M	387	386	385	384
P8347	M	391	390	389	388
P8348	M	395	394	393	392
P8349	M	399	398	397	396

8.5 Standard channel parameters (continued)

Definition of the M functions

P8350	M	403	402	401	400
P8351	M	407	406	405	404
P8352	M	411	410	409	408
P8353	M	415	414	413	412
P8354	M	419	418	417	416
P8355	M	423	422	421	420
P8356	M	427	426	425	424
P8357	M	431	430	429	428
P8358	M	435	434	433	432
P8359	M	439	438	437	436
P8360	M	443	442	441	440
P8361	M	447	446	445	444
P8362	M	451	450	449	448
P8363	M	455	454	453	452
P8364	M	459	458	457	456
P8365	M	463	462	461	460
P8366	M	467	466	465	464
P8367	M	471	470	469	468
P8368	M	475	474	473	472
P8369	M	479	478	477	476
P8370	M	483	482	481	480
P8371	M	487	486	485	484
P8372	M	491	490	489	488
P8373	M	495	494	493	492
P8374	M	499	498	497	496
P8375	M	503	502	501	500
P8376	M	507	506	505	504
P8377	M	511	510	509	508
P8378	M	515	514	513	512
P8379	M	519	518	517	516
P8380	M	523	522	521	520
P8381	M	527	526	525	524
P8382	M	531	530	529	528
P8383	M	535	534	533	532
P8384	M	539	538	537	536
P8385	M	543	542	541	540
P8386	M	547	546	545	544
P8387	M	551	550	549	548
P8388	M	555	554	553	552
P8389	M	559	558	557	556
P8390	M	563	562	561	560
P8391	M	567	566	565	564
P8392	M	571	570	569	568
P8393	M	575	574	573	572
P8394	M	579	578	577	576
P8395	M	583	582	581	580
P8396	M	587	586	585	584
P8397	M	591	590	589	588
P8398	M	595	594	593	592
P8399	M	599	598	597	596

P8400	M	603	602	601	600
P8401	M	607	606	605	604
P8402	M	611	610	609	608
P8403	M	615	614	613	612
P8404	M	619	618	617	616
P8405	M	623	622	621	620
P8406	M	627	626	625	624
P8407	M	631	630	629	628
P8408	M	635	634	633	632
P8409	M	639	638	637	636
P8410	M	643	642	641	640
P8411	M	647	646	645	644
P8412	M	651	650	649	648
P8413	M	655	654	653	652
P8414	M	659	658	657	656
P8415	M	663	662	661	660
P8416	M	667	666	665	664
P8417	M	671	670	669	668
P8418	M	675	674	673	672
P8419	M	679	678	677	676
P8420	M	683	682	681	680
P8421	M	687	686	685	684
P8422	M	691	690	689	688
P8423	M	695	694	693	692
P8424	M	699	698	697	696
P8425	M	703	702	701	700
P8426	M	707	706	705	704
P8427	M	711	710	709	708
P8428	M	715	714	713	712
P8429	M	719	718	717	716
P8430	M	723	722	721	720
P8431	M	727	726	725	724
P8432	M	731	730	729	728
P8433	M	735	734	733	732
P8434	M	739	738	737	736
P8435	M	743	742	741	747
P8436	M	747	746	745	744
P8437	M	751	750	749	748
P8438	M	755	754	753	752
P8439	M	759	758	757	756
P8440	M	763	762	761	760
P8441	M	767	766	765	764
P8442	M	771	770	769	768
P8443	M	775	774	773	772
P8444	M	779	778	777	776
P8445	M	783	782	781	780
P8446	M	787	786	785	784
P8447	M	791	790	789	788
P8448	M	795	794	793	792
P8449	M	799	798	797	796

8.5 Standard channel parameters (continued)

Definition of the M functions

P8450	M	803	802	801	800
P8451	M	807	806	805	804
P8452	M	811	810	809	808
P8453	M	815	814	813	812
P8454	M	819	818	817	816
P8455	M	823	822	821	820
P8456	M	827	826	825	824
P8457	M	831	830	829	828
P8458	M	835	834	833	832
P8459	M	839	838	837	836
P8460	M	843	842	841	840
P8461	M	847	846	845	844
P8462	M	851	850	849	848
P8463	M	855	854	853	852
P8464	M	859	858	857	856
P8465	M	863	862	861	860
P8466	M	867	866	865	864
P8467	M	871	870	869	868
P8468	M	875	874	873	872
P8469	M	879	878	877	876
P8470	M	883	882	881	880
P8471	M	887	886	885	884
P8472	M	891	890	889	888
P8473	M	895	894	893	892
P8474	M	899	898	897	896
P8475	M	903	902	901	900
P8476	M	907	906	905	904
P8477	M	911	910	909	908
P8478	M	915	914	913	912
P8479	M	919	918	917	916
P8480	M	923	922	921	920
P8481	M	927	926	925	924
P8482	M	931	930	929	928
P8483	M	935	934	933	932
P8484	M	939	938	937	936
P8485	M	943	942	941	940
P8486	M	947	946	945	944
P8487	M	951	950	949	948
P8488	M	955	954	953	952
P8489	M	959	958	957	956
P8490	M	963	962	961	960
P8491	M	967	966	965	964
P8492	M	971	970	969	968
P8493	M	975	974	973	972
P8494	M	979	978	977	976
P8495	M	983	982	981	980
P8496	M	987	986	985	984
P8497	M	991	990	989	988
P8498	M	995	994	993	992
P8499	M	999	998	997	996

8.5 Standard channel parameters (continued)

Messages

P8500	Message	system	causes abort of all channels
P8501	Message	high-priority	causes abort of the channel
P8502	Message	low-priority	causes stop of the channel
P8503	Message	only display	

P8504	Additional information of axis	
P8505	Additional information 1 for message	Info1 in the picture with key '?'
P8506	Additional information 2 for message	Info2 in the picture with key '?'
P8507	Additional information 3	Info3 in the picture with key '?'

P8508 Acknowledgement of pending CNC messages
(see also q157)

P8509 Actually displayed message (parameter of the PLC one describes)
Byte 1 Message paragraph
Byte 2 Message paragraph
Byte 3 Channel number (1... 8)
Byte 4 Axis paragraph (1... 32)
(Parameter is described by PLC)
When message numbers between 3000 ... 3999, the message is only entered in the sample buffer when P11263 was not described.
(see also P11263)

8.5 Standard channel parameters (continued)

Channel locking

P8510 Automatic transmission locking
The lockings that are defined here are activated in the AUTOMATIC mode.
Activating can be done with the mode of operation changeover (see also P11010 . . .)
Byte 0

Bit 0	Parameter locking 1	(see also parameter status byte 3)
Bit 1	Parameter locking 2	(see also parameter status byte 3)
Bit 2	Parameter locking 3	(see also parameter status byte 3)
Bit 3	Parameter locking 4	(see also parameter status byte 3)

Byte 3

Bit 0 NC-memory locking
Bit 1 Cycle-memory locking

P8511 Removal of the parameter barrier for NC program
For the parameter barriers defined here one becomes in the AUTO operation
Parameter allocation off the NC program out certified.

Byte 1	Bit 0	Parameter barrier 1	(see also parameter status byte 3)
	Bit 1	Parameter barrier 2	(see also parameter status byte 3)
	Bit 2	Parameter barrier 3	(see also parameter status byte 3)
	Bit 3	Parameter barrier 4	(see also parameter status byte 3)

8.5 Standard channel parameters (continued)

Channel informations

P8540 M function - interface diagnosis
Diagnostic parameter only described with active PLC< >CNC
P8549 interface diagnosis. (P11280 = 1)

Byte 1 M function paragraph
Byte 2 M function paragraph
Byte 3 M function definition

P8550 Channel number, is pre-allocated.
Contents 1 with channel 1
Contents 2 with channel 2
:
Contents 8 with channel 8

P8551 Number of axes of this channel, is obtained when the switching on routine.

P8552 Final axis in the channel (logical axis paragraph)
After the switching on routine one obtains.

P8553 Mapping test parameter
Mode of operation: P8553 is described with a p-paragraph, sees P8554.

P8554 Those obtains the Procedure called when describing the parameter P8553
q paragraph this parameter and this writes after P8554.

P8555 Actual work area, switches zero point blocks over
0 or -- Invalid work area
1 1. Record of the zero points actively
2 2. Record of the zero points actively
: :

P8556 Mask for suppressed shifts when activating a zero point
in the AUTOMATIK operation.
Contents: Bit information
Example: P8556 = 4 (0000 0100)
==> with the next zero point call becomes 3. Shift does not consider

P8559 Control rooms on end of block acknowledgement with M function with stop
0 or -- ok
1 Wait for acknowledgement (block release)
(CNC waits for the PLC Acknowledgement of a m-function with stop)

P8561 Dripfeed buffer level

8.5 Standard channel parameters (continued)

Channel informations

P8570 ... Actual reference point display

P8570	1st display field	Content:	1..32 logical axis number
:			0 no display
P8577	8th display field		

P8578 Number of max. reference point displays in the status line, if P8578 < axis number of the last channel axis, the reference point displays are showing the subsequent reference points.

P8679 All reference points taken
0 or -- Not all reference points taken
1 All reference points in the channel taken

8.5 Standard channel parameters (continued)

Channel informations

P8580 : P8587
Actual organization of the actual position display
This parameter area is set by the operating system.
(may not be written).

P8580
Number of the display field
Content: 1..32 logical axis number
0 no axis display

Byte 1	display field 1
Byte 2	display field 2
Byte 3	display field 3
Byte 4	display field 4

P8581
Number of the display field

Byte 1	display field 5
Byte 2	display field 6
Byte 3	display field 7
Byte 4	display field 8

:

P8587 Number of the display field 29..32

P8588 Maximum number of actual display fields

8.5 Standard channel parameters (continued)

Channel informations

P8590 Standard (customer specific) organization of the actual position display
: At each operating mode change, the here defined actual position display is activated.
P8597

P8590 Number of the display field
Content: 1..32 logical axis number
0 no axis display
Byte 1 display field 1
Byte 2 display field 2
Byte 3 display field 3
Byte 4 display field 4

Comment:
With the describing of P8590 the Istpositionsanzeige is again structured.
(trigger parameters)

P8591 Paragraph of the indication area
Byte 1 display field 5
Byte 2 display field 6
Byte 3 display field 7
Byte 4 display field 8

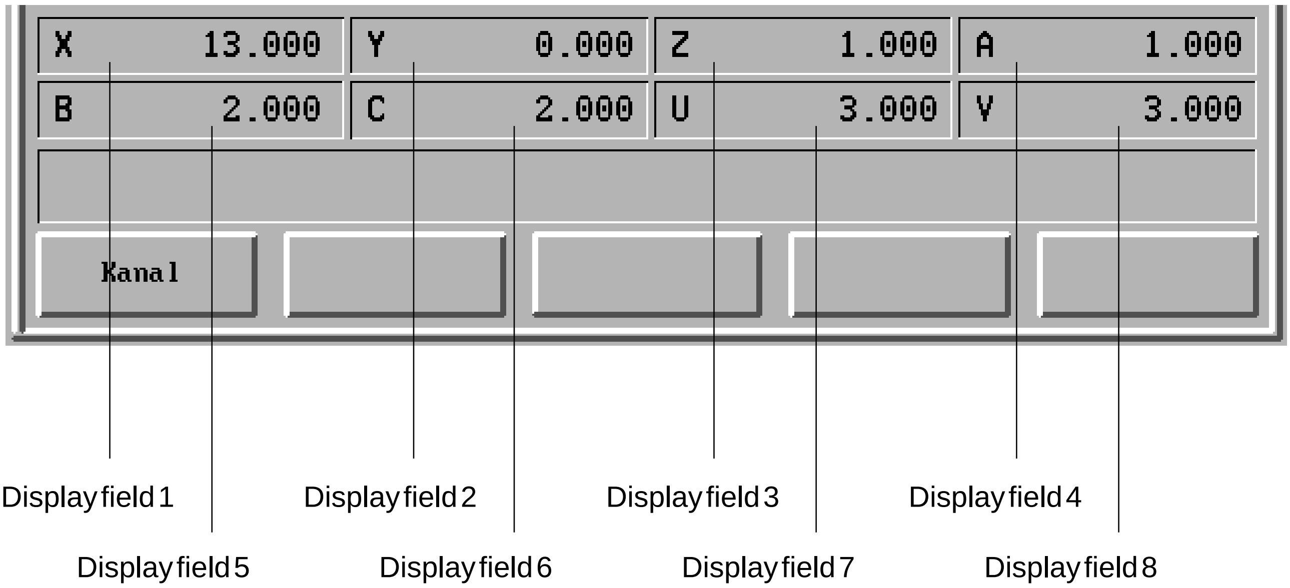
P8597 Paragraph of the indication area 29.. 32
Parameter with trigger function.
With each describing P8597 becomes into that
parameters P8590... activates p8597 specified Istpos display.

P8598 Display mode of the actual position display
0 or -- If the actual position display knows not all available axes at the same time
to raise, if this becomes on the final indication area with the axis selection
of an axis, which not at the moment to display comes, another axis
to displace.
1 Actual position display does not change.

8.5 Standard channel parameters (continued)

Channel informations

Example actual position display



8.5 Standard channel parameters (continued)

Presettings MANUAL (planned)

P8600 Presetting axis position display
:
P8639

8.5 Standard channel parameters (continued)

Presetting AUTO (planned)

P8640 1 NC block analysis transfer only, if cycle in the block programs actual (P9700.. 9789 described)
no positioning of the master record off
no version of I/J/K/R/T/F/G/M/S...
P9700..9789 initialized:
- NUMERICAL CONTROL program start
- return of the cycle to the main program

P8642 0 or — P9790.. P9799 is not described
<> 0 pointer on additional syntax list for NC blocks with cycle
additional syntax items, which in blocks with cycles for the order are.

Example
P8642:100 (pointer on P100)

P100: 72 (' H ') —> P9790: Contents of H
P101: 75 (' K ') —> P9791: Contents of K
P102: 79 (' O ') —> P9792: Contents of O
P103: 0 (end of the list)

with P8642 <> 0:
P9790..9799 initialized:
- NC program start
- return of the cycle to the main program

P8650 G-function-default AUTOMATIK operation
Parameter value: G-function code
Table is processed up to the ßt reset parameter or to P8659
Example: P8650: 28 / G28 settings
P8651: 54 / G54 settings
P8652: 18
P8653: -
P8659:

P8660
:
P8679

8.5 Standard channel parameters (continued)

Channel - Commands to PLC

P8681 Key Manual (to PLC) is written by the operating panel
0 Key released
1 Key pressed (PLC:THANDK1)
P8681 is written by the operating panel.

P8682 Key Start / Stop (to PLC) is written by the operating panel
0 Keys released
1 Key Stop pressed (PLC:TSTOPK1)
2 Key start pressed (PLC:TSTARK1)
When writing P8682, P8684 is written with AUTO Start /-Stop and answered to the PLC.

P8683 Mode of operation change (to PLC)
1 Manual (PLC:BAWHAK1)
Pending CNC messages become with mode change after hand reset.
2 Automatic sequential blovk (PLC:BAWAUK1)
3 Automatic single block (PLC:BAWAUK1)
8 Automatic positioning block (PLC:BAWAUK1)

8.5 Standard channel parameters (continued)

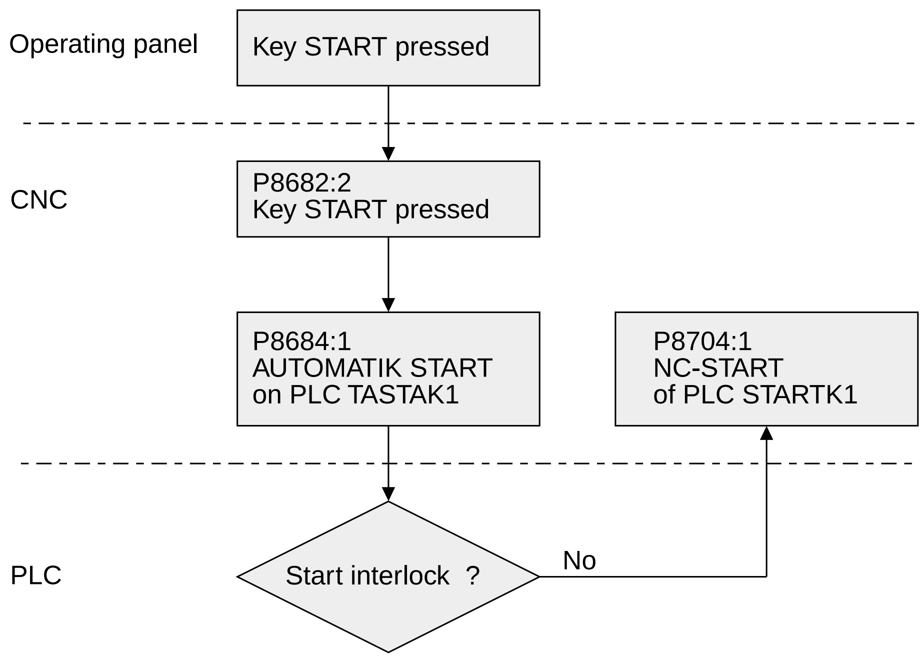
Channel - Commands to PLC

P8684 Automatic transmission Start/Stop (to PLC)

0	Stop	(PLC:TASTOK1)
1	Start	(PLC:TASTAK1)

Example: NC Program start
When writing P8682, P8684 is written with AUTO- Start/Stop and answered to the PLC. If there are no start interlocking in the PLC, the Start/Stop command is answered to P8704. With P8704 the NC program (P8800) is started.

Example: Flow of a NC program start



P8685 Automatic program active (to PLC)
The CNC indicates here the actual condition.
The parameters may not be written. .

0	Program not yet active or finished.	(PLC:ANAK_K1)
1	Program active Stop	(PLC:AASP_K1)
2	Program active Start	(PLC:AAST_K1)

8.5 Standard channel parameters (continued)

Channel - Commands to PLC

P8686 Restarting info. at PLC
- 0 No restarting, normal railway company.
- 1 Restarting without axis movement, NC program is in the browsing.
- 2 Restarting at outline (axis movement), starting block was found.
  Outline on direct path started with rapid traverse feed (P8816).
  With achieving the outline P8686:0

P8687 AUTO program runs
Parameter may not be described!
- 0 with program end or program abort
- 1 parameter is described uniquely with program start with 1

P8688 Acknowledgment of the current operating mode of the interpolator.
P8707 —(Interpolator)—> P8688

P8689 Actual type of coordinate (see also P8759)
0 or -- Transformation out
15, 16 Polar transformation
47, 48, 49 Robot transformation

P8690 M function code at PLC
Transmit a m-code at PLC in the manual operation.

P8695 Acknowledgement for remainder path resetting
P8695: 0, at P8715:0
P8695: -1, at the start of an NC block with M26
If an NC path over
- Hardware probe input
- P8715

aborted, P8695 is described by the current NC block number.

P8696 Acknowledgement for measuring position accommodation
P8696: 0, bei P8716:0
P8696: -1, at the start of a test drive P8717
When measuring position taken
- Hardware probe input
- P8716

P8696 is described by the current NC block number.
(see also P8895)

8.5 Standard channel parameters (continued)

Channel commands of the PLC

P8700 Enable regulator for channel (PLC:REOK_K1)
0 no regulator enabling, i.e. command = actual of the axis
1 enable regulator

P8701 Enable move of all axes in the channel (PLC:KAFR_K1)
0 no channel enabling, i.e. stopping the axes without dynamics
1 channel enabling

P8702 Security stop of all axes in the channel (PLC:SICHAK1)
1 Stop, i.e. stopping the axes with dynamics

P8703 Mode of operation (PLC:BA_K1)
1 MANUAL (PLC:HAND_K1)
2 AUTO sequential block (PLC:AUTF_K1)
4 AUTO single block (PLC:AUTE_K1)
8 AUTO positioning block (PLC:AUTP_K1)

P8704 AUTO Start / Stop (PLC:STARTK1)
Herewith a NC program is started (number is in P8800).
0 Stop
1 Start

P8705 NC program abort (PLC:ABBRUK1)
1 Abort

P8706 Enable block (PLC:SAFREK1)
Sentence by sentence control of a NC program
0 No block change
1 Block change

P8709 Reference points taken (PLC:REFALK1)
0 Reference points not yet taken
1 All reference points taken

8.5 Standard channel parameters (continued)

Channel command of PLC

P8710 Emergency programm call 1 (PLC:NPA1K1)
0 No call
1 Program call
Program - and block number according to P8820, P8821
Function:
When calling up an emergency program a running NC program is interrupted.
When activating a emergency program in manual mode, the operating mode AUTO sequential block is changed.
If operating mode change is locked, the message 1320 appears.
At the end of the emergency program the mode of operation AUTO sequential block remains. When starting the emergency program P8710 is loaded with 0.

P8711 Emergency program call 2 (PLC:NPA2K1)
0 No call
1 Program call
Program- and block number according to P8822, P8823
Function see P8710

P8714 Conditional block skip (M22) (PLC:BSP22K1)
0 no skip
1 skip after block number

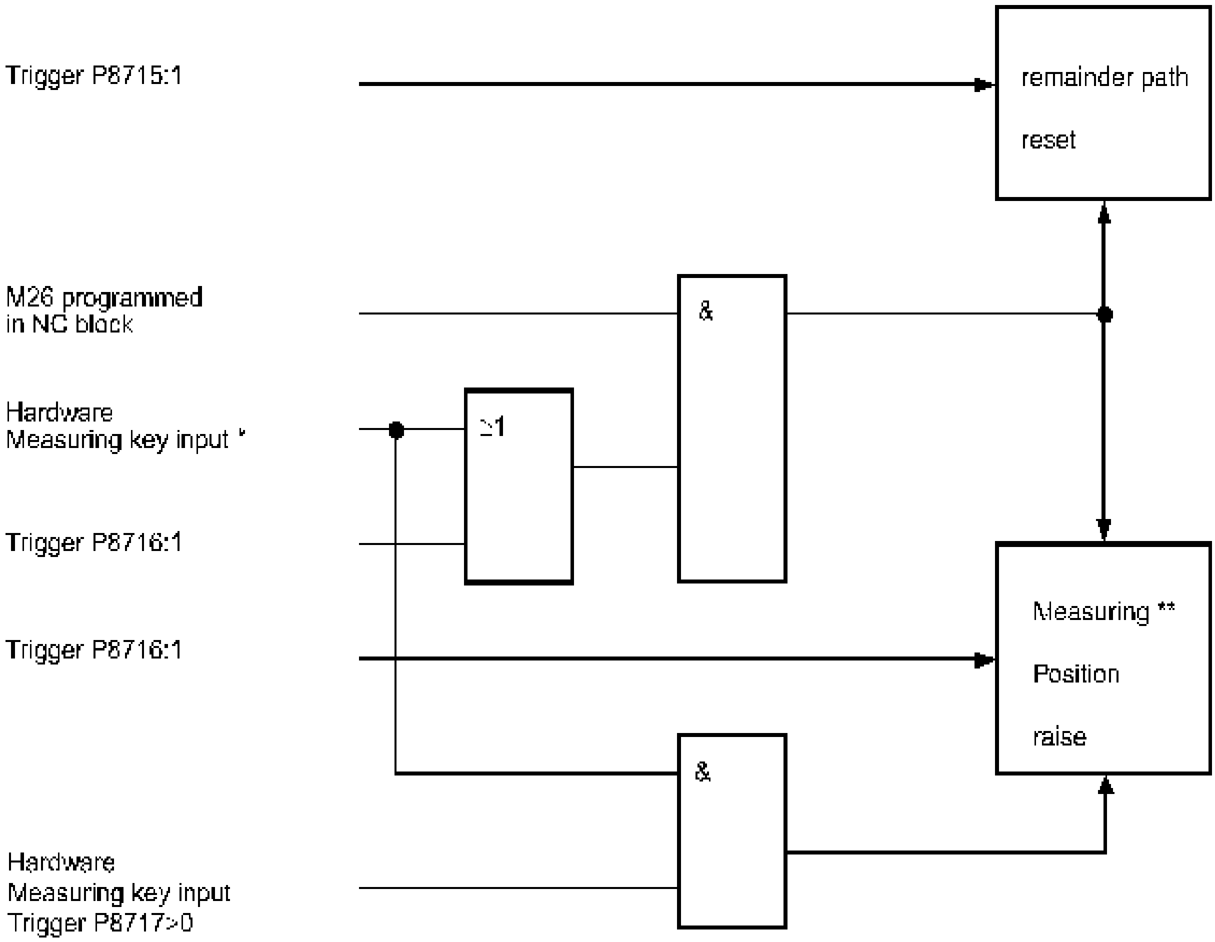
P8715 Trigger for clearing movement in progress (M26) (PLC:TRWLOK1)
0 set P8695:0
1 trigger (see P8695)

P8716 Trigger for software-measuring key (PLC:TSWMTK1)
0 set P8696:0
1 Trigger for recording measurement position of all axes in the channel
see: P8696, P8895

P8717 Measuring position release
1 Measuring position release over sensor input
unique release for measuring position accommodation
2 Measuring position release over sensor input
constant release for measuring position accommodation
(see P8696)

8.5 Standard channel parameters (continued)

Clearing residual traverse (movement in progress) Measuring position logics



note:

* see also	P12045	measuring logic
** see also	P12153	measuring position
	P12181	measuring position achieves
	P8855	measuring position type of display
	P8655	acknowledgement measuring position accommodation

8.5 Standard channel parameter (continued)

Teach-in

P8747 Axis selection at Teach-In
0 or -- All existing axis positions are overtaken in the NC block at key target=actual
<>0 Only axis positions according to selection are overtaken in the NC block
Example:
P8747: 0FH At axis 1 to 4, the position is overtaken in the NC block.

P8748 Radius with Teach in used tools.

Example: The NC program created in Teach in with tool radius R=10mm.
This NC program can be processed now with a drill by R=8mm,
if palpation tool P8748:10 is settinged.

The werkzeugradius-Korrektur actually which can be executed becomes by this Input influences.
Example:

Tool	T1:	Radius R=10mm
Palpation tool:		P8748:2,5 (mm)

N10 T1 M6
N20 G42
. —> radius correction operates with R=7.5mm
.

P8749 Mode of coordinate for Teach - In

Input	0	off
	48	tool coordinate mode
	49	work piece coordinate mode

8.5 Standard channel parameters (continued)

Manual operation

P8750	Traverse mode selection			
	1	Continuously	(relative)	(PLC:KONT_K1)
	2	Step	(relative)	(PLC:STEP_K1)
	3	Hand-wheel		(PLC:HARA_K1)
	4	Divide	(absolute/relative)	(PLC:TEIL_K1)
	5	Home	(absolute/relative)	(PLC:HOME_K1)

P8751 Axis selection (PLC:RAGAWK1)
Logical axis number 1...32

P8752	Manual move + / -		
	0	stop	
	1	+	(PLC:HAND+K1)
	2	-	(PLC:HAND-K1)

With the describing of this parameter the technique becomes the axis (P8751) started in the suitable procedure mode (P8750). Parameter becomes of the system with depressing the key hand + / - described.

P8753 Relative-travel in manual mode
(internally used parameter)

P8754 max. programmable manual feed [mm/min, degree/min]

P8755 Programmed manual feed rate [mm/min, degree/min]

P8756 Actual manual feed rate [mm/min, degree/min]

8.5 Standard channel parameters (continued)

Manual operation

P8757 Coordinate system in HAND

Input	--, 0, 17	G17
	18, 19, 117, 118, 119	Gxxx

The tool length always affects the vertically axis of the interpolation plane.
Parameter wid after each passing through of the switching on routine reset.

P8758 Zero pointes in Manual
- display of the actual zero point in manual operation.
- When writing the corresponding zero point is activated.

Input 53...59, 153... (reset 15)
Parameter wid after each passing through of the switching on routine reset.

P8759 Coordinate mode in Manual
0 transformation off
Polar transformation
Input 15 off
16 on
Robot transformation
Input 47 off
48 Tool-coordinates
49 Workpiece-coordinates
4849 Automatic switch between 48 and 49
Parameter wid after each passing through of the switching on routine reset.

8.5 Standard channel parameters (continued)

Step operation

P8760	Actual step width		[mm]
P8761	step width 1 h on step width		
P8762	Pointer on selected step		
P8763 : P8769	Step width table		
P8763	Step width 1 default	 10	
P8764	Step width 2 default	 1	
P8765	Step width 3 default	 0,1	
P8766	Step width 4 default	 0,01	
P8767	Step width 5 default	 0,001	
P8768	Step width 6 default	 0	
P8769	Step width 7 default	 0	

8.5 Standard channel parameters (continued)

Hand-wheel

P8770 Active hand-wheel
0 No hand-wheel switched on
1..8 Number of hand-wheel switched on

P8771 Hand-wheel 1
Bitinfo for axes, which may proceed with hand-wheel.
Bit 0...31, axis 1...31
Example: 8771: 03
The 1st and 2nd axis can only be selected with the hand-wheel.
:
:
P8778 Hand-wheel 8

P8779 Hand-wheel definition
Byte 1 0 Hand-wheels are activated individually with the hand-wheel key.
Hand-wheel multiplicator is modified individually.
During hand-wheel far one circuit axis selection is not changed.

$10 All hand-wheels are always simultaneous activated.
Hand-wheel multiplicator is modified individually.
During the hand-wheel far circuit the former axis selection
becomes this hand-wheel again activates.

8.5 Standard channel parameters (continued)

Indexing (planned)

P8780
:
P8789

8.5 Standard channel parameters (continued)

Home (planned)

P8790
:
P8799

8.5 Standard channel parameters (continued)

Start data

P8800 Program number of the program to be started (see P8704)

P8801 Block number
Start from this block.

P8802 Program number in the case of the NC program abort
Parameter is described by the system
see function: Restart at outline

P8803 Record number in the case of the NC program abort
Parameter is described by the system
see function: Restart at outline

P8804 Restarting at outline
0 or -- Restarting function not actively
1 Restarting function actively

Mode of operation restarting at outline:

In the case of the abort of a current NC program (HAND abort, message) becomes of system the actual NC program position in the parameters P8802, P8803 and P8807 stored. A properly terminated NC program resets these information.

After an NC abort if the restarting mode is switched on and the NC program started, then if the NC program up to the restarting point is simulated,
i.e.. no axis movements egeben themselves.

In the restarting block the simulation is switched off. The restarting position on direct path one starts. With achieving the restarting position becomes P8804=0 settinged.

One restarts on the initial position of the aborted block.
If the NC program in a process cycle is aborted, then becomes to cycle start started.

When restarting become accordingly m-functions, M6 and m-cycles
M function definition (P11050..., P8250...) treats.

When restarting become accordingly m-functions, M6 and m-cycles
M-Funktionsdefinition (P11050..., P8250...) treats.
During the restarting simulation P8050 (advance t NR.) becomes
to the PLC do not transfer.

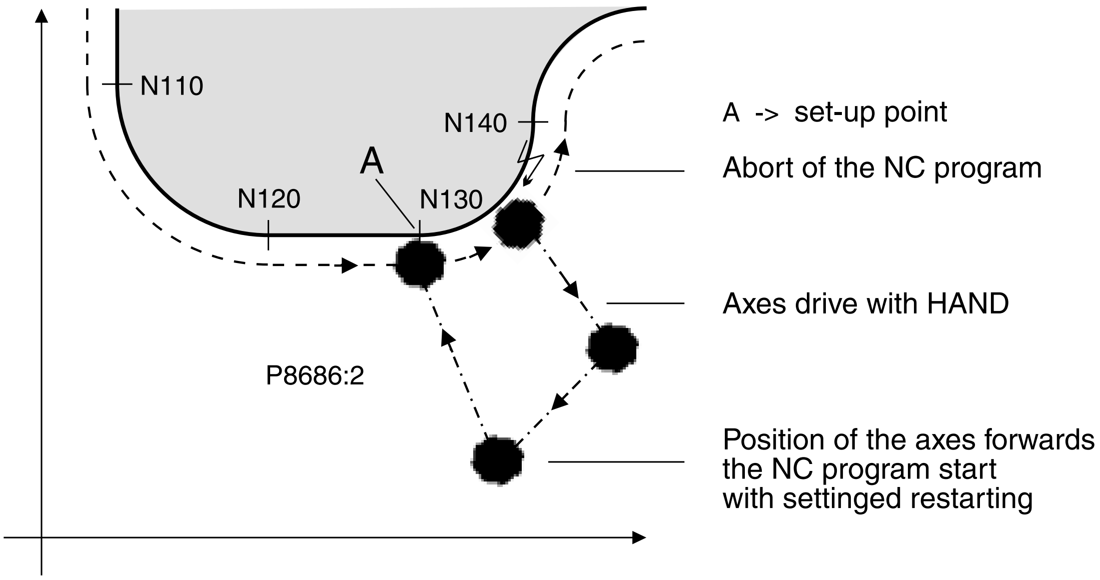
see also P8686

8.5 Standard channel parameters (continued)

Start data

P8805 Restarting adjustments

Example:



P8805 Selection NC interpreter

0, — NC program is executed with standard NC interpreter.

1 NC program is executed with enhanced NC interpreter.
—> Processing of C high-level language elements.

8.5 Standard channel parameters (continued)

Start data

P8806 Start data flag

Byte 1 bit 0 0 No block display during cycle handling
1 Block display runs during cycle handling

bit 1 1 Micro block display with DLL cycles (display of the advance)
in the 2. Message line. Display the block items, which to
NC interpreters are geschleusst.
(parameter allocations are carried out in the CNC DLL,
if not real time echtzeit-oder synchronisation parameters)
see also P8814

Bit 2 -
Bit 3 -
Bit 4 0 All NC blocks are stopped in AUTO SINGLE BLOCK mode
1 Only records with real-time information to be stopped in
AUTO SINGLE BLOCK.
e.g.: Blocks with parameter calculations can not be stopped
Blocks with traversing positions or M function... etc. be stopped
Bit 5 -
Bit 6 -
Bit 7 -

Byte 2 bit 0 0 Increasing automatic file barrier actively,
that is, of the NC interpreter processed NC programs and
cycles remain up to the program end (abort or M30)
with the editing barrier provide (max. 250 entries).
1 Increasing automatic file barrier not actively,
now the editing barrier is only so long active, as the NC
interpreter the NC program opened
(interpreter operates in the advance).
Bit 1 1 Message M1307 is suppressed (symbolic variables unknown)
Bit 2 0 Virtual Console releases of the NC program (Example: N200 message)
are displayed in the respective channel. (Default)
1 Virtual Console releases of the NC program (Example: N200 message)
are shown in the current display channel.
Bit 3 -
Bit 4 -
Bit 5 -
Bit 6 -
Bit 7 -

8.5 Standard channel parameters (continued)

Start data

Byte3 bit 0 0 Drill cycles do not pend to modal,

G80 does not have effect.
1 Drill cycle modality switched on,
with the unique program of drilling cycle (G81, 83,84,85) is in the following NC blocks, implicitly WITH way, the first called programmed drilling cycle.
This Modalitaet is switched off with G80.

Bit 1 0 —
1 Drilling programming via address letters activated;
that is, if G81/83/84/85 programmed in the block following address values are stored only on parameters.

drilling feed:	F	P32
drilling depth	X,Y,Z	P33 (depends on current plane G17/18/19)
advanced stop plain	R	P34
retraction plain	I,J,K	P35
Number of strokes	H	P36

plain	drilling depth	advanced stop plain	retraction plain
G17	Z	R	K
G18	Y	R	J
G19	X	R	I

Bit 2 :-
Bit 3 :-
Bit 4 :-
Bit 5 :-
Bit 6 :-
Bit 7 :-

Byte 4 bit 0 0 Initialization program number numerical
1 Initialization program number alphanumerically
Bit 1 :-
Bit 2 :-
Bit 3 :-
Bit 4 :-
Bit 5 :-
Bit 6 :-
Bit 7 :-

8.5 Standard channel parameters (continued)

Start data

P8807 Start data flag 2

Byte 1 bit 0 0 —
1 With the call of a tool in the NC block (T) Z5 becomes processed.
With the call of Z5 actual the PLC Interface (P8050..) with new data described.
Error message, if Z5 missing!

bit 1 0 —
1 With the call of a tool in the NC block (T) none become tool datas loaded. No access to the tool data array.
With a call of Z5 P9322/P9323 are loaded.

Bit 2 —-

Bit 3 —-

Bit 4 0 (Default) Loading the Dripfeed buffer, change of operating modes and for auto mode
1 Loading the Drip Feed buffer only at the start of the automatic program

Byte 2 :

Bit 0 0 Start of the NC interpreter with NC START in automatic mode
1 Start of the NC interpreter change of operating modes to AUTO.
The NC interpreter begins immediately by the mode change to AUTOMATIC with the interpretation of the NC blocks and prepares the NC data for the interpolator.
This is done until the first synchronization in the NC program, or until the buffer is filled to the interpolator.
(see on forerun P9300/P9301)

Will reference for a preliminary executed:
- Parameter calculations / allocations (not real-time assignments!)
- Program Controlling M functions M23, M28 ..
  if not defined with synchronization
- Cycle calls
With NC-Start will be:
- traverse paths
- M functions to the PLC
- Real-time parameter assignments
executed.

At the end of the program the NC interpreter is started up again immediately.

8.5 Standard channel parameters (continued)

Start data

In the mode change to AUTO :
P8685 : 1 (STOP program active)
P8687 : 1 (Program runs)
set to!!

Bit 1 -
Bit 2 -
Bit 3 -

Bit 4 -
Bit 5 -
Bit 6 -
Bit 7 -

Byte 3 —-

Byte 4 —-
Bit 0 1 ALL records for interpolator looped through (temporary!!)
Bit 1 1 Enable cache NC

8.5 Standard channel parameters (continued)

Start data

P8808 Driving mode AUTOMATIC
- 0 No traversing the axes in AUTOMATIC
- 1 Manual traversing of the axes in the AUTOMATIK operation approved

P8809 Graphics on/off
- 0 Graphics off
- 1 Graphics on

P8810 Block skip
- 0 Block skip not switched on
- 1 Block skip switched on

P8811 Dripfeed operation
- 0 No Dripfeed operation
- 1 With Dripfeed operation

P8812 Without axis movement
- 1 NC program sequence without axis movement

P8813 Without M - function
- 1 NC program sequence without M-function
  - M-functions at PLC are not transmitted
  - program-controlling M-functions work unchanged
  - branch M-functions do not branch

8.5 Standard channel parameters (continued)

Start data

P8814 Single block mode with / without stop in intermediate blocks
0 or -- each main block is stopped in the single block mode
1 after each block end is stopped in the single block mode
(at cycle blocks, insert block etc.

P8815 Test feed rate [mm/min]
0 or -- switched off
<>0 at G01, G02 and G03 this programmed feed is used.

P8816 Rapid traverse G00 [mm/min]
0 or -- Manual feed rate active
max. Axis speed (P12004)
the axes available in the channel positions.
< > 0 with prog. Rapid traverse is positioned with this speed.

P8817 Asynchronous feed rate [mm/min]
The programmed feed rate of the active NC block is replaced
until the block end by the feed rate of P8817

P8818 Percentage feed rate change [%]
The programmed feed rates in the NC program are changed with this
percentage value.

P8819 eroding feedrate

8.5 Standard channel parameters (continued)

Start data

P8820 Program number — Emergency program 1
P8821 Block number
Emergency program call up see P8710

P8822 Program number — Emergency program 2
P8823 Block number
Emergency program call up see P8711

P8824 Program number — (Emergency program 3 planned)
P8825 Block number
Emergency program call up see P8712

P8826 Program number — (Emergency program 4 planned)
P8827 Block number
Emergency program call up see P8713

P8828 M-function-modification
Byte 1 bit0 0 With M0 by the CNC System always on STOP is generated
1 M0 - measuring stop is controlled via PLC
M0 stops: PLC settings with M0 program STOP (P8704) forwards. The M-function-acknowledgement.
M0 does not stop: PLC does not setting program STOP before that M-function-acknowledgement

P8830 Zero point in AUTO
0 or -- NC program starts with G53
53..59,153.. NC program starts with the programmed zero point

P8836 Correction-reviewed at G 151/152
Percentage contour distance
content: 0.0 : prog. Points to go (G152 ineffective)
.
.
1.0 : Corrected points to go

8.5 Standard channel parameters (continued)

Program controlling parameters

P8840 Definition of the coordinate system for G117
(see also q152...)

Byte 1	1. logical axis paragraph abscissa (horizontal axis) centre point identifier I	
Byte 2	2. logical axis paragraph ordinate (vertically axis) abscissa and ordinate specify the level, in which the circular interpolation takes place. centre point identifier J	
Byte 3	3. logical axis paragraph definition of the vertically axis on the interpolation level	

The definition of the sense of circle turning a right-angled coordinate system always becomes assumed.
The direction of rotation in the various levels is as follows determined:
One looks against the direction of the axis, which is vertically on the level.
(axis No. in byte 3).
With G02 now in the clockwise direction one interpolates.
With G03 against the clockwise direction one interpolates.

Programming example:
N100 G117 P8840:$00010302
byte: 4 3 2 1

P8841 Definition of the coordinate system for G118 (see also q153..)
Default, if
0 or -- $020103

Byte 1	03 (centre point identifier	K)
Byte 2	01 (centre point identifier	I)
Byte 3	02	

P8842 Definition of the coordinate system for G119 (see also q154..)
Default, if
0 or -- $010302

Byte 1	02 (centre point identifier	J)
Byte 2	03 (centre point identifier	K)
Byte 3	01	

8.5 Standard channel parameters (continued)

Program controlling parameters

P8843 Max. one permitted radius error during centre point programming [mm]

0 or -- Default radius error 0.003mm
otherwise error message M1238

<>0 Permitted error of the radius during centre point programming, starting from that the error message M1238 is output.

P8844 Radius correction messages suppress

Byte 1

bit 0 0 M1154 release
1 M1154 is suppresses

bit 1 0 M1155 release
1 M1155 is suppresses

P8845 Inserting sets / intersections with werkzeugradius-Korrektur.

-- at outside edges sets are always inserted.

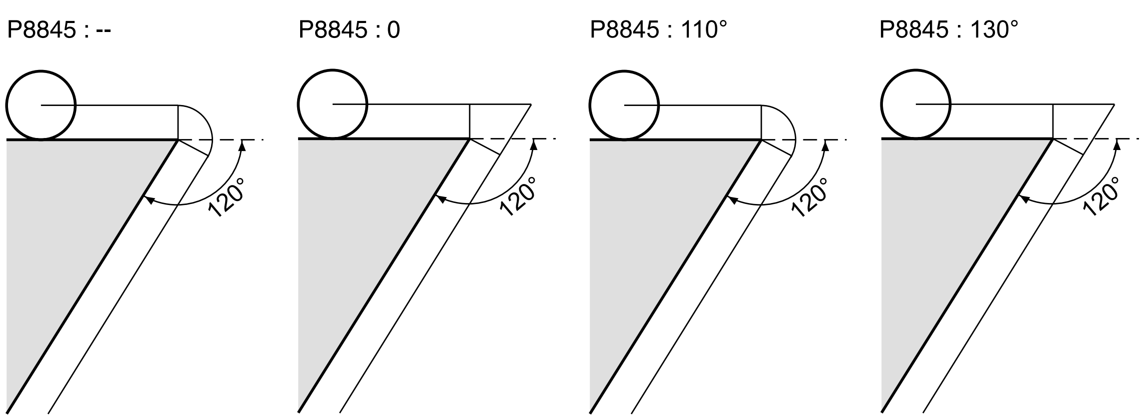
0 in the case of direction deviation over 120 degrees sets are inserted under 120 degrees intersections are calculated.

0...180 [degrees] in the case of direction deviations, which are larger than the input degree number, sets are inserted and in the case of direction deviations, which are smaller than the input degree number, intersections are calculated.

Example:

Direction deviation 120° and degree number of 110: Circle is inserted.

Direction deviation 120° and degree number of 130: Intersection is calculated.



8.5 Standard channel parameters (continued)

Program controlling parameters

P8846 Spline modification
0 or -- 3D-spline (standard)
<>0 5D-spline switched on
byte 1 logical axis paragraph of the 4. spline axis
byte 2 logical axis paragraph of the 5. spline axis

1., 2., 3. Spline axis are specified over G17, G18, G19.
Example: P8846:$0604 means
4. Spline axis is 4th logical axis
5. Spline axis is 6th logical axis

P8847 Spline modification
0, -- or 0,5 Standard spline
0,5...2 Spline modifies
i.e. decaying the Splines is influenced.
0,5 Flat spline
0,85 Spline decays more

Effectiveness becomes clear with large bracket steps.

8.5 Standards channel parameters (continued)

Program controlling parameters

Feed increase or feed degradation with convex or concave sections

P8848 Feed dynamics convex Input: [0-100% positively or negatively]
Feed is proportionally increased or lowered.

P8849 Feed dynamics concave Input: [0-100% positively or negatively]
Feed is proportionally increased or lowered.

When inserting values in P8848 and P8849 the following formula applies:

F calculated feed
F_p programmed feed in the NUMERICAL CONTROL program
R outline radius
WR radius of the tool

F = F_p * (1 +(P8848 or P8849)/100 * (P8850-R)/(P8850-WR))
with R > P8850 is F = F_p

P8850 Feed dynamics
0 or -- Feed dynamics switched off
1 With switched on werkzeugradiuskorrektur the feed becomes the actual path adapted, i.e. the programmed path feed applies to the edge contact tool.
> 1 Petition of the outline Referenzradiuses, starting from which no feed modification one executes. P8848 and P8849 actively.

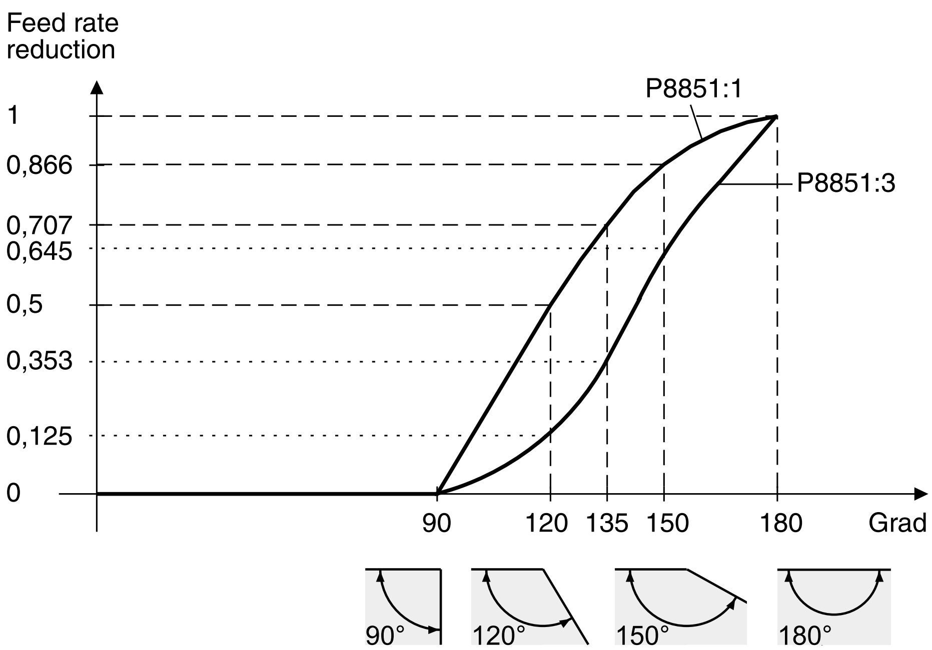
8.5 Standards channel parameters (continued)

Program controlling parameters

P8851 Corner dynamics [Input: 0...10]
0 or -- Standard corner dynamics actively
<>0 Corner dynamics
-1 Switch off

Function of the standard corner dynamics
When approaching a corner, the contour feedrate is reduced in dependancy of the corner angle, that means slow down before corners. Because of the corner angle, a reducing factor is effective (cos), which results in the contour feedrate before the corner when multiplying with the programmed feed rate.

Example:	Corner angle	Corner feed reduction	
		at P8851:1	at P8851:3
	≤90 degree	0	0
	120 degree	0,5	0,125
	135 degree	0,707	0,353
	150 degree	0,866	0,649
	180 degree	1	1



Function corner dynamics
At P8851>0, the standard reducing factor is raised to a power with the content of P8851. The larger P8851 is, the more it is slowed down before corners.

8.5 Standards channel parameters (continued)

Program controlling parameters

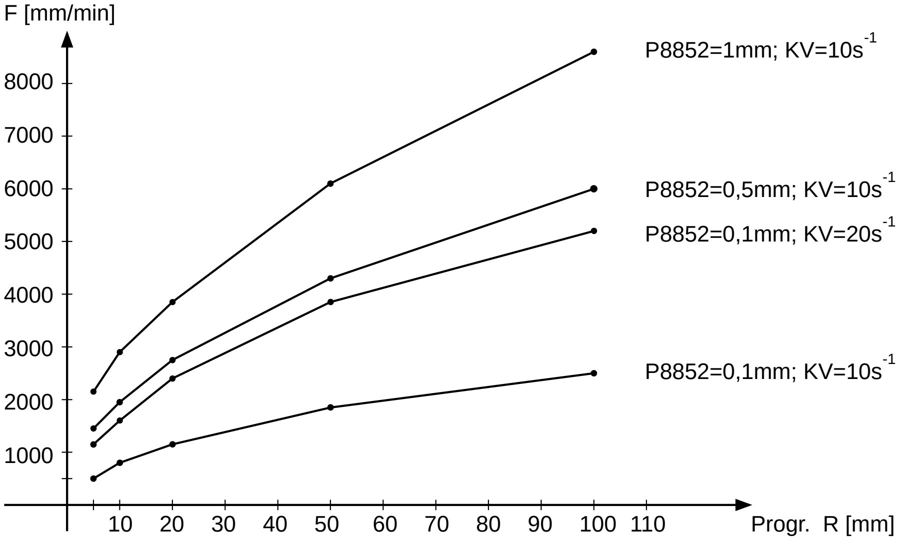
P8852 Circle dynamics [mm]
0 or -- Circle dynamics switched off
<>0 Circle dynamics active
input value permissible path deviation [mm]
function reduction of the feedrate at circular interpolation

At circular interpolations (G02/G03), an internal circle feed rate is calculated with the help of the permissible path deviation which is limiting the feed rate that is programmed in the NC program.

Internal circle feed rate is dependant of
- programmed radius
- KV factor
- P8852

Remark:
P8852 always has to be smaller than the programmed radius, if not error message M1237 appears

Example



8.5 Standards channel parameters (continued)

Program controlling parameters

P8853 Preset coordinate mode at NC program start
0 or -- No coordinate mode selected
49 NC program is starting with the indicated coordinate mode 49

P8854 Radius correction at G49
0 Radius correction in space, ONLY in spline contour (G50) or tangentiellem contour course
1 Radius compensation in the plane

P8855 Measuring position modification (P12153)
Byte 1:
0 Measurement position with respect to save the machine position
1 Measurement position with respect to save display position (with ramping)
Byte 2:
1 Measurement position save position with respect to machine (for M26 abort without ramp)
Byte 3:
1 Measurement position save position with respect to machine (for M26 wait to mess-active)

P8856 Internal parameter
(zero point setting CNC <—> Bediendeld)

P8857 Outline dynamics with path interpolation
0 or -- Outline dynamics switched off
<>0 Outline dynamics actively

Input value certified path deviation [mm]
Function reduction of the feed
drive path (G1, G2, G3, G50)
The programmed points of path are connected with sets.
Thus the calculated circle radii form the basis for one
possible path feed reduction. To calculation see P8852.

P8858 Definition of rapid traverse (G00)
Rapid traverse rate (see P8816)
0 or -- Rapid traverse always with exact stop,
i.e.. at the end of record feed = 0,
lag distance < exact stop
1 Rapid traverse always with feed stop
i.e.. at the end of record feed = 0,
for lag distance < exact stop is not waited
2 Rapid traverse no stop
end of record handling as with (G1, G2, G3, G50)

8.5 Standards channel parameters (continued)

Program controlling parameters

P8859 G92-Modifikation
Parameter must be settinged before the NC program start!
Byte 1
bit 0 0 G92-Verschiebung rotates with 1. Rotation axis
1 G92-Verschiebung no rotating with 1. Rotation axis
bit 1 0 G92-Verschiebung is turned by G45 (starting from version 090)
1 G92-Verschiebung is not driven by G45
bit 2 0 G92-Verschiebung does not work kummulativ
G92-Verschiebung resets pending G92-Verschiebung of all axes.
1 G92-Verschiebung works axis kummulativ
(effect only on programmed axis)
pending G92-Verschiebungen not programmed axes are preserved.
Byte 2 bit 0 0 I/J/K programming works in accordance with more up-to-date
G90 / G91 programming
1 I/J/K programming always works in G91

P8860 G95-Modifikation
Byte 1
$xx 0 G95 acquisition data concerning spindle data 1 (P11640)
1 - 8 G95 reference spindle from spindle data 1-8
Byte 2
$xx 1 - 8 G95 acquisition data bezueghlich coupling data 1-8
(* tooth number of masters / tooth number of Slave)

P8861 Feed modification for not interpolated axes
Only effectively if P8893 < > 0;
0 or -- Axis feed concerning absolute feed P12122
1 Equal axis feed programmed path feed in the NC program
2 Axis feed equal programmed path feed proportion everything
in the NC Progroramm more programmieter (not interpolated axes)
i.e.. Axis final positions are achieved at the same time.

P8865 Channel-specific definition of the Overrides for interpolation rate
0 or -- Default Override 1 (1-8)
Observe:
Override Definitions in the axis data (P12125) are superordinate!

8.5 Standards channel parameters (continued)

Program controlling parameters

P8866 Channel-specific definition of the Overrides for relative velocity

0 or -- Default Override 1 (1-8)

Observe:

Override Definitions in the axis data (P12126) are superordinate!

P8867 Channel-specific definition of the Overrides for absolute speed

0 or -- Default Override 1 (1-8)

Note: Override Definitions in the axis data (P12127) are superordinate!

P8870 Path length modification (describe from P8871)

Parameter contains bit information

P8870 0 P8871 is not described

Bit 0 1 P8871 kummuliert path components of the ßt logical axis

Bit 1 1 P8871 kummuliert path components of the 2nd logical axis

:

Bit 31 1 P8871 kummuliert path components of the 32nd logical axis

When each describing P8870, P8871 is reset

P8871 Cummulative path length (e.g. for service life monitoring)

P8880 Axis change, switch on, switch off

0 or -- Switched off

1 Axis change on

2 Axis replacement for AUTO -> Delete HAND switching (by interpolator)

If an axis change is switched on, the display position is exchanged.

Zero point shifts from G53 to G59 effect on the physical axes, they remain at the same physical axis at an axis change.

Zero point shift G92 and correction G44 effect on logical axes and are influenced therefore by the axis exchange.

8.5 Standard channel parameters (continued)

Program controlling parameters

P8881 Exchange the two physical axes
Byte 1: on axis
Byte 2: from axis
Input of the logical axis numbers

Example: P8881 : $0106
P8882 : $0601
i.e. axis change between axis 1 and axis 6

Programmed commands in the NC program
for 1st axis are transmitted to 6th axis,
for 6th axis are transmitted to 1st axis.

P8882 Axis change
P8883 Axis change
:
P8889

P8890 Tangential axis
0 No tangential axis available in the channel
1-32 logical axis number
The here indicated axis becomes tangential the programmed path of the NC program adjusted.

P8891 Axis specific block locking

Bit 0	1	1st logical axis with axis-specific block locking
Bit 1	1	2nd logical axis
:		
Bit 31	1	32nd logical axis

Function:
If the axis-specific block locking is selected, executing the NC program in the interpolator is stopped if axes should be moved which are selected with the axis-specific block locking.

8.5 Standard channel parameters (continued)

Program controlling parameters

P8892 Axis reflection
Parameter contains bit information

Bit 0	1	1. logical axis reflect
Bit 1	1	2. logical axis reflect
:		:
Bit 31	1	32. logical axis reflect

Mode of operation:
One reflects around the actual machine position P12150.
Is modified thus the debit position P12151 of +<->-;
and the indicator position P12154 of +<->-;
e.g.. P12154: 150,0 P12154 becomes: -150.0
With programmed positive drive, the axis moves around this amount in negative direction.

Parameter is reset with each mode change after HAND.

P8893 Traverse axis not interpolating
Parameter with bit information

Bit 0	1	1st logical axis with axis-specific block locking
Bit 1	1	2nd logical axis
:		
Bit 31	1	32nd logical axis

Mode of operation:
During the processing of a NC program in the Interpolator, the settinged axis becomes from the interpolation removes and parallel (as absolute position with feed P12122... or with progr. To path feed see P8861 driven).
At the NC end of record an exact stop requested waits the CNC to all axes (also the not interpolated) in position, or with not interpolated axes and programmed Vorposition P12113, this is achieved

Parameter is reset in each switching on check.

8.5 Standard channel parameters (continued)

Program controlling parameters

P8894 Axes reduct switch off
Parameter contains bit information
Bit 0 1 1. logical axis no Achsreduct
Bit 1 1 2. logical axis
:
Bit 31 1 32nd logical axis

Mode of operation:
During processing of a NC program the axes indicated here become with the calculation of the feed dynamics (P8851) does not consider, or with programmed path feed (see P8861) proceed.

Parameter is reset in each switching on routine.

P8895 Measurement position recording released
In P8895 == 0: all axes in the channel be released (default)
If P8895 > 0: Enable of individual axes (parameter with Bitinfos)
Bit 0 0 Not released first logical axis measurement position recording
Bit 0 1 Released first logical axis measurement position recording
Bit 1 Released second logical axis measurement position recording
.. ..
Bit 31 Released 32. logical axis measurement position recording

P8899 Bracket axis system modification (see P11830 etc..)
0 Path operation
(display as programmed, to axes drive with bracket proportions)

1 Axis operation
(drive display with bracket proportions, of axes as programmed)

8.5 Standard channel parameters (continued)

Real time - Information block analysis

P8900 Actual program numbers of the NC - program
P8901 Actual block number of the NCS - program

P8904 Actual cycle number
if cleared, no cycle active.
P8905 Actual block number in the cycle
if cleared, no cycle active.

P8906 Actual program number when NC program is aborted
- is loaded with the actual program number when NC program is aborted
- is written 0 at program end with M30

P8907 Actual block number when NC program is aborted
- is loaded with the actual program number when NC program is aborted
- is written 0 at program end with M30

P8910 Faulty program
- When the actual NC - program is aborted by the NC - interpreter (because of error message) the faulty program, which has led to the abort, is indicated here.

P8911 Faulty block
- When the actual NC - program is aborted by the NC - interpreter (because of error message) the faulty block, which has led to the abort, is indicated here.

P8912 Actual sub-program plane

P8913 Actual sub-program call up, program number
Here the actual program number is placed when switching to a sub-program.
P8914 Actual sub-program call up, block number
Here the actual block number is placed when switching to a sub-program.

P8918 I
P8919 J
P8920 K
P8921 R

P8925 S1 programmed in NC - block
:
P8932 S8 programmed in NC - block

8.5 Standard channel parameters (continued)

Real time - Information block analysis

P8934	Programmed loop counter	M24		
P8935	Actual loop counter	M24		
P8936	Programmed feedrate F			
P8937	Programmed block end feedrate			
P8938	Actual feed F (of the interpolator)			
P8940	Actual interpolation mode	G00 / G01 / G02 / G03	(to the PLC)	
	0	G00		(PLC:G00_K1)
	1	G01		(PLC:G01_K1)
	2	G02		(PLC:G02_K1)
	3	G03		(PLC:G03_K1)
P8941	Programmed dwell	G04		[s]
P8942	Actual dwell	G04		[s]
	When running a dwell (G04) programmed in the NC - program, the remaining dwell is indicated here.			
P8943	Actual rotating direction circular axis	G05 / G06 / G07		
P8945	Actual polar coordinate programmng	G10 / G11		
P8946	Actual contour programming	G12		
P8947	Actual tangential axis	G13 / G14		
P8948	Actual polar transformation	G15 / G16		
P8950	Actual interpolation plane	G17 / G18 / G19		
P8951	Axis 1			
P8952	Axis 2			
P8953	Axis 3			
P8958	Actual exact stop modal	G28 / G29		
P8964	Actual tool - radius - function	G40 / G41 / G42		
P8965	Actual axis correction	G43 / G44		
P8966	Actual rotating on / off	G45 / G46		
P8967	Actual co-ordinate system	G47 / G48 / G49		
P8968	Actual	G50 / G51 / G52		
P8969	Actual zero pointes	G53 . . . G59		
	Content: G53 . . G59	(actual zero point - shift see P12155. . .)		

8.5 Standard channel parameters (continued)

Real time - information block analysis

P8970	Actual reflecting on	G61
P8971	Actual reflecting on	G62
P8972	Actual feedrate 100%	G63/G64
P8973	Actual	G65
P8974	Actual corrections switched off	G66
P8975	Actual	G67/G68/G69
P8978	Actual absolute/chain dimension	G90/G91
P8979	Actual zero point-shift Content: 0 no G92-shift active 92 G92-shift active	G92
P8880	Actual feedrate definition actually	G93/G94/G95
P8981	Actual number of revolution mode spindle	G96/G97/G98
P8982	Actual end feedrate	G99

P9080 Tool group number (T-No.)

P9083 Tool Sub record (selected)

P9090 Radius of the tool
P9091 Length of the tool
P9092 Radius correction
P9093 Length correction
P9094 Type of tool (active quadrant 0...9)

Real time - data blocks

Parameters are pre-allocated with program start with 0

P9100	Actual G92 shift 1st. axis	[mm]
P9101	Actual G92 shift 2nd. axis	
:		
P9131	Actual G92 shift 32nd. axis	
:		
:		
P9199		

8.5 Standard channel parameters (continued)

Preanalysis-information block of the block interpreter

P9300	Actual program numbers of the block interpreter		
P9301	Actual block number of the bliock interpreter		
P9303	Sub-program - plane		
P9304	Actual cycle number of the block interpreter If cleared, no cycle in the block interpreter.		
P9305	Actual block number in the cycle of the block interpreter If cleared, no cycle in the block interpreter.		
P9318	I		
P9319	J		
P9320	K		
P9321	R		
P9322	T-number		
P9323	T-Sub-number		
P9325	S1		
:			
P9332	S8		
P9334	Loop depth M24		
P9336	Programmed feedrate F		
P9340	Type of interpolation programmed 0, 1, 2, 3, 50, 51, 52 Contents with program start	 1	G00 / G01 / G02 / G03 G50 / G51 / G52
P9341	Retention time programmed		G04
P9343	Direction of rotation programmed round axis		G05 / G06 / G07
P9344	Programmed exact stop		G08 / G09
P9345	Polar coordinates programming programmed 10, 11 Contents with program start	 11	G10 / G11
P9346	Outline path programming programmed		G12

8.5 Standard channel parameters (continued)

Preanalysis-information block of the block interpreter

P9350 Interpolation level programmed G17 / G18 / G19 / G117
17, 18, 19, 117
Contents with program start 17

P9351 Axis 1(1st principal axis) contents of logical record number
P9352 Axis 2(2nd principal axis) contents of logical record number
P9353 Axis 3(delivering axis) contents of logical record number

P9358 Programmed exact stop modal G28 / G29
28, 29
Contents with program start 29

P9364 Tool offset compensation programmed G40 / G41 / G42
40, 41, 42
Contents with program start 40

P9365 Programmed axis correction G43 / G44
43, 44
Contents with program start 43

P9366 Programming rotation in/out G45 / G46
45, 46
Contents with program start 46

P9367 Programmed coordinate system G47... G147
47, 48, 49, 147
Contents with program start 147

P9369 Programming actual zero point G53... G159
53, 54, 55, 56, 57, 58, 59, 153, 154, 155...
Contents with program start 53

P9370 Programming reflecting G60 / G61
60, 61
Contents with program start 60

P9371 Programming reflecting G60 / G62
60, 62
Contents with program start 60

8.5 Standard channel parameters (continued)

Preanalysis-information block of the block interpreter

P9372 Programmed feed 100% G63 G63 / G64
Feed override G64
Contents with program start 64

P9374 Programmed corrections switched off G66

P9378 Programmed absolute measure/chaining absolute G90 / G91
90, 91
Contents with program start 90

P9379 Zero shift programmed actively G92
92
Contents with program start 92

P9380 Feed definition programmed up-to-date G94, G95
Contents 94, 95
Contents with program start 94 (mm/min)

P9381 Type of speed programmed spindle G96, G97
Contents 96, 97
Contents with program start 97 (U/min)

8.5 Standard channel parameters (continued)

Preanalysis data blocks

P9500	Actual G92 shift 1st. axis	[mm]
P9501	Actual G92 shift 2nd. axis	
:		
P9531	Actual G92 shift 32nd. axis	
:		
:		
P9699		

8.5 Standard channel parameters (continued)

Preanalysis NC-Block - informations

P9700	Programmed position 1st. Axis	
:		
P9731	Programmed position 32nd. Axis	
P9732	I	
P9733	J	
P9734	K	
P9735	R	
P9736	T	
P9738	f	
P9739	F	
P9740	G-function 1	
:		
P9747	G-function 8	
P9748	G-function	enlargement field
P9750	M-function 1	
P9761	M-function 1	enlargement field 1
P9762	M-function 1	enlargement field 2
:		
P9771	M-function 8	
P9772	M-function 8	enlargement field 1
P9773	M-function 8	enlargement field 2
P9780	Speed 1	
:		
P9787	Speed 8	
P9790	Customized cycle interface	
:	see P8642	
P9799	Customized cycle interface	
P9800	Tool data - interface	
:	- Interface to the PLC	
:	- Actual spindle tool - data block	
:	- Actual active data block	
P9999		

8.6 System parameters

Key - switch - information

The following parameters are copies of the key-switches.

P11000 Key - switch 1 (Info)
0 : Switch open
1 : Switch closed
With the key - switch 1 the conduct of the starting routine is influenced.
P11000 : 0 —> ESR must be confirmed with pressing the key
P11000 : 1 —> ESR runs automatically, in so far as no errors appear.
(e.g.: parameters def., or NC - memory def.)

P11001 Key - switch 2
P11002 Key - switch 3
P11003 Key - switch 4

#P11006 Software - Sperre 3
#P11007 Software - Sperre 4

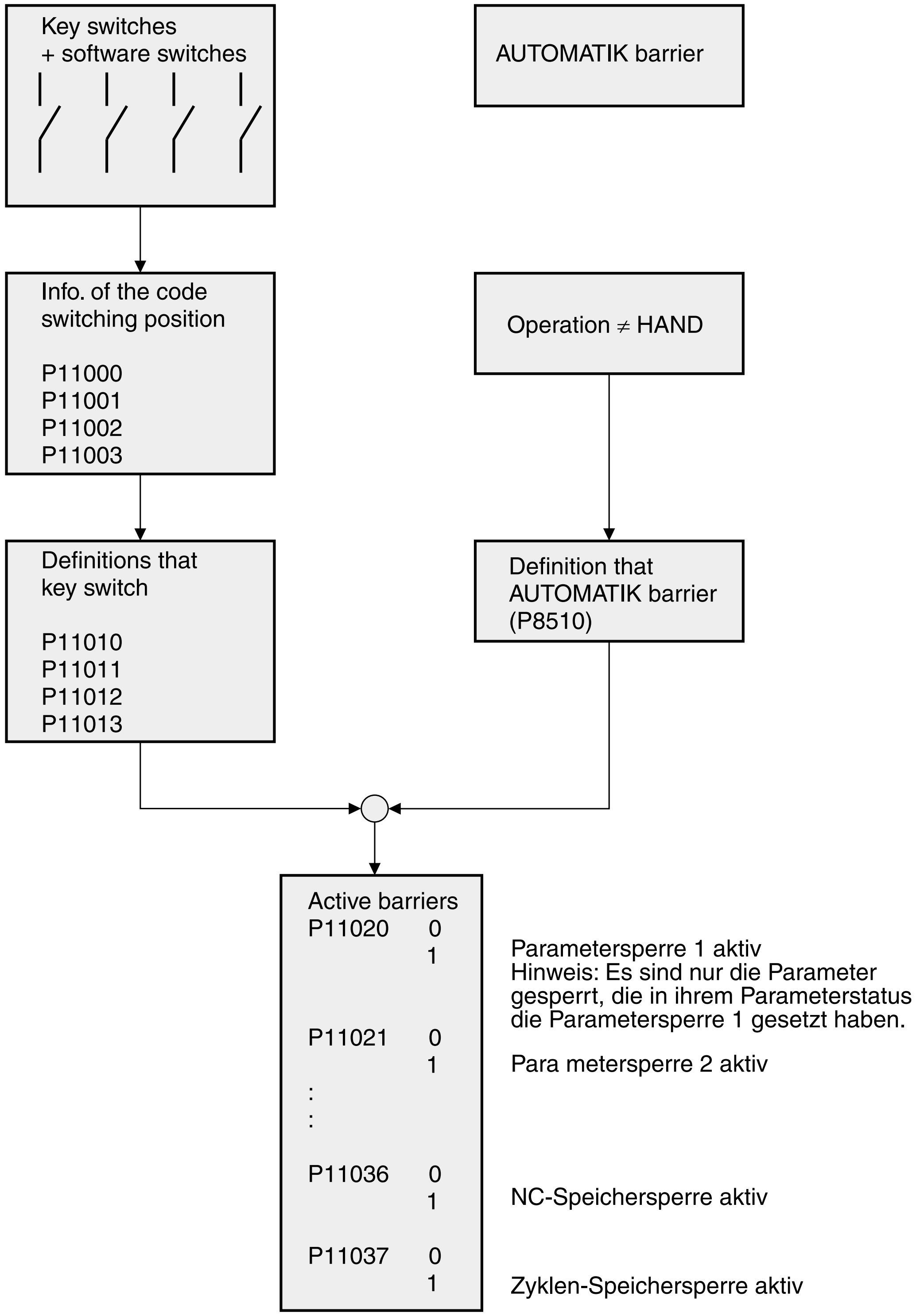
P11008 Software locking 1

0	not active	
1	active	lockings activated according to P11018.
-1	release	superordinate release of the barriers defined in P11018

P11009 Software barrier 2

8.6 System parameters (continued)

Overall view parameter and NC memory barriers



8.6 System parameters (continued)

Key - switch - definitions

Definition, which locking is selected with the switch.

P11010 Key-switch 1 (see also P8510 / P8511)
Observe
With activated parameter barrier
- the parameter status is to be described nevertheless,
- the PLC can change nevertheless the parameter value.

Byte 1
Bit 0 parameter locking 1 (see also parameter status byte 3)
Bit 1 parameter locking 2 (see also parameter status byte 3)
Bit 2 parameter locking 3 (see also parameter status byte 3)
Bit 3 parameter locking 4 (see also parameter status byte 3)

Byte 2
Bit 0 Custom lock 1
Bit 1 Custom lock 2
Bit 2 Custom lock 3
Bit 3 Custom lock 4

Byte 3
Bit 0 NC memory locking
Bit 1 Cycle memory locking

Byte 4
Editing lockings of the MMI's
Bit 0 editing locking parameter mantissa
Bit 1 editing locking parameter status
Bit 2 editing locking axis / spindle / coupling data
Bit 3 editing locking zero point data
Bit 4 editing locking tool data
Bit 5 editor OFF locked

P11011 Key - switch 2
P11012 Key - switch 3
P11013 Key - switch 4

P11018 Software lock 1
P11019 Software lock 2

8.6 System parameters (continued)

Active lockings

P11020	0	Locking not active
:	1	Locking active
P11049		

This area is updated with each modification of a key switch status and each mode change.

P11020 System-parameter-locking 1
P11021 System-parameter-locking 2
P11022 System-parameter-locking 3
P11023 System-parameter-locking 4

P11028 Customized. Block 1-Status
P11029 Customized. Block 2-Status
P11030 Customized. Block 3-Status
P11031 Customized. Block 4-Status

P11036 NC memory locking
P11037 Cycle memory locking

Editing lockings of the MMI's

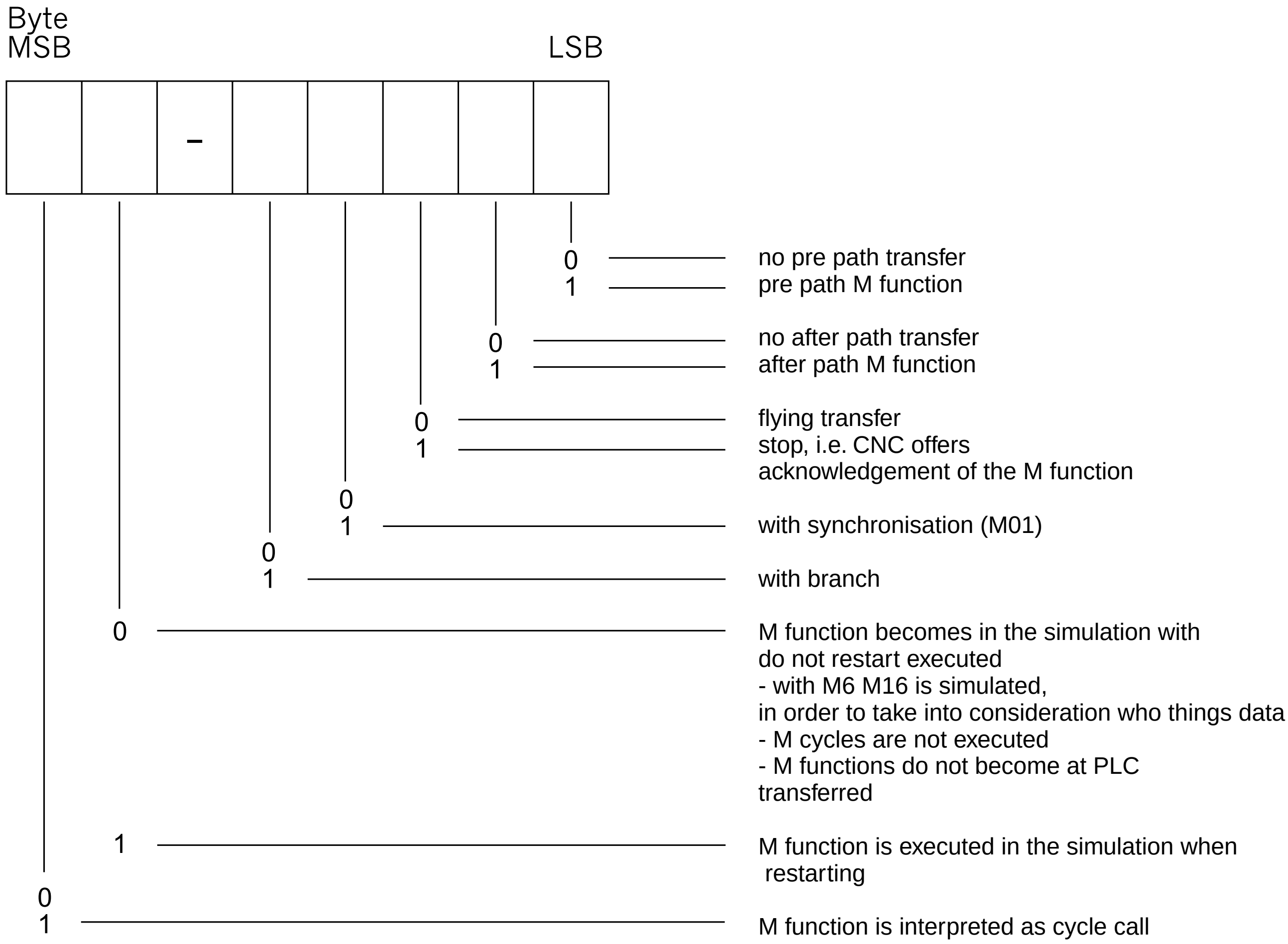
P11044 Editing locking parameter mantissa
P11045 Editing locking parameter status
P11046 Editing locking axis/spindle/coupling data
P11047 Editing locking zero point data
P11048 Editing locking tool data
P11049 Editor OFF locked

8.6 System parameters (continued)

Definitions of the M - functions

Maximum 200 system overlapping M - functions are programmable (M000 ... M199). If more than 200 M - functions are
required, those must be defined channel specifically (to M999).
4 M - functions are defined per parameter. Input in hexadecimal!

Info for a M - function :



M-functions without stop are transmitted ‘flying’ to the PLC according to definition before traverse, after traverse or both. The NC - program runs continuously.

8.6 System parameters (continued)

M-functions with stop are transmitted to the PLC according to definition before traverse, after traverse or both. Axes run directly before the transfer of the M-function approach their end position with exact-stop. The NC-program sequence is stopped until the M -function is confirmed. The program pre analysis continues working. The confirmations are controlled in the PLC-program with enable block (SAFREK1). Therefore the corresponding marker of a M-function must be connected with stop with enable block.

M-functions with skip work always before traverse with stop! The traverse can be cleared at any time through the single - confirmation of the programmed M-function. When clearing the delay traverse, the measuring - position - parameters are written. In this case the NC program branches to the programmed skip target.
If the skip - M - function is not confirmed, the programmed NC - block is processed completely and switched on to the next block. Marker of a M - function with skip in the PLC may not be connected with enable block (SAFREK1).

M - Function as cycle. If this bit is set in the M - function - definition, no M - function code is sent to the PLC, but a cycle with the corresponding number is called up, i.e. M234 starts Z234.
At this definition the other definitions of this M - function are not considered (before traverse, after traverse or with skip).
The programmed M - function must be programmed without extension (e.g.: M1234.56 is not allowed).
The cycle becomes active at the block end. The M - function number must be larger than 30!

Byte 2
MSB LSB

Example:

0	0	–	0	0	1	1	0

P11051	$	10	05	**06**	01
M - number	7	6	**5**	4	

i.e.:

M07	with skip	
M06	before traverse active,	with stop
M05	after traverse active,	with stop
M04	before traverse active,	without stop

Note: Modifications in P11050... becomes with a mode change by HAND after AUTO effectively.

8.6 System parameters (continued)

Definitions of the M - functions

P11050M	03	02	01	00
P11051M	07	06	05	04
P11052M	11	10	09	08
P11053M	15	14	13	12
P11054M	19	18	17	16
P11055M	23	22	21	20
P11056M	27	26	25	24
P11057M	31	30	29	28
P11058M	35	34	33	32
P11059M	39	38	37	36
P11060M	43	42	41	40
P11061M	47	46	45	44
P11062M	51	50	49	48
P11063M	55	54	53	52
P11064M	59	58	57	56
P11065M	63	62	61	60
P11066M	67	66	65	64
P11067M	71	70	69	68
P11068M	75	74	73	72
P11069M	79	78	77	76
P11070M	83	82	81	80
P11071M	87	86	85	84
P11072M	91	90	89	88
P11073M	95	94	93	92
P11074M	99	98	97	96
P11075M	103	102	101	100
P11076M	107	106	105	104
P11077M	111	110	109	108
P11078M	115	114	113	112
P11079M	119	118	117	116
P11080M	123	122	121	120
P11081M	127	126	125	124
P11082M	131	130	129	128
P11083M	135	134	133	132
P11084M	139	138	137	136
P11085M	143	142	141	140
P11086M	147	146	145	144
P11087M	151	150	149	148
P11088M	155	154	153	152
P11089M	159	158	157	156
P11090M	163	162	161	160
P11091M	167	166	165	164
P11092M	171	170	169	168
P11093M	175	174	173	172
P11094M	179	178	177	176
P11095M	183	182	181	180
P11096M	187	186	185	184
P11097M	191	190	189	188
P11098M	195	194	193	192
P11099M	199	198	197	196

8.6 System parameters (continued)

Internal oscilloscope

P11100 Amount of active tracks 1 to 4

P11101 Parameter address for track 1
P11102 Parameter address for track 2
P11103 Parameter address for track 3
P11104 Parameter address for track 4

P11110 Trigger track 1 to 4
P11111 Trigger level

P11112 Trigger slope
- 1 trigger on rising edge
- -1 trigger on dropping edge
- 0 trigger immediately

P11113 Scanning time [ms]

P11114 Oscilloscope Start/Stop
- 0 stop
- 1 start
- 2 automatic new start

P11115 Triggerpoint reached
- 0 not reached
- 1 reached

8.6 System parameters (continued)

Graphics parameter

P11120 Origin of the co-ordinate system screen X
P11121 Origin of the co-ordinate system screen Y
P11122 Origin of the co-ordinate system screen Z
P11123 Zoom - size (P11123 small _> drawn picture large)

P11124 Graphics display
- 0 X,Y - Plane
- 1 Z,X - Plane
- 2 Y,Z - Plane
- 3 3-D

P11125 X tilt
P11126 Z tilt
P11127 Forward / backward of the display

P11128 Display bits

Byte 1	Bit 0	free	
	Bit 1	free	
	Bit 2	free	
	Bit 3	1	display with actual path
	Bit 4	1	display with zero point
	Bit 5	1	display with circle centre points
	Bit 6	1	display with bores
	Bit 7	1	display with direction arrows

8.6 System parameters (continued)

System commands / informations

P11130 Selected language
0 or -- 1st language
1 2nd language
: :

P11131 Default language
If the language text accord. to P11130 is not available, the default language is used.
0 or -- 1st language
1 2nd language
: :

P11132 Teachpanel operating mode
0 or -- Teachpanel off
1 Teachpanel switched on

P11133 Teachpanel key pressed / unpressed
Byte 1 Key code (see list from P11200 on)
Byte 2 0 unpressed
1 pressed
Byte 3 Channel number 1..8

P11134 LED's teachpanel (planned)

P11135 Desired channel in the surface
For multi-panel operation:
Byte 1 desired channel panel 1
Byte 2 desired channel panel 2
Byte 3 desired channel panel 3
Byte 4 desired channel panel 4

P11136 Selected channel in the surface
Byte 1: channel selected panel 1
Byte 2: channel selected panel 2
Byte 3: channel selected panel 3
Byte 4: channel selected panel 4
In the descriptions of P11136 via the PLC individual bytes can be set without changing the other bytes!
e.g.: P11136 : $0203
describe P11136 via PLC with $0400
P11136 is to $0403; that is, Byte 1 is unchanged!
see also P11265

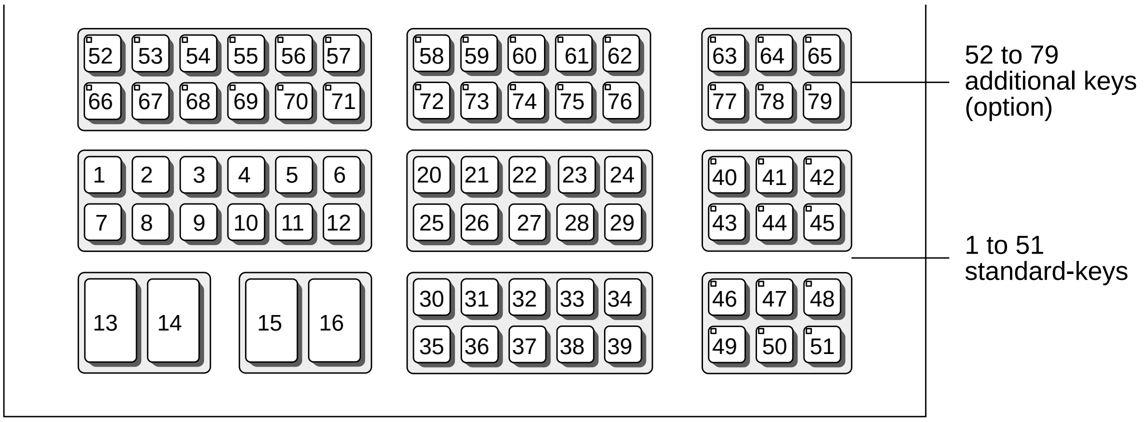
8.6 System parameters (continued)

System commands / information

P11137 Keyboard code pressed / unpressed (only CNC keys)
(parameter is transmitted to the PLC)

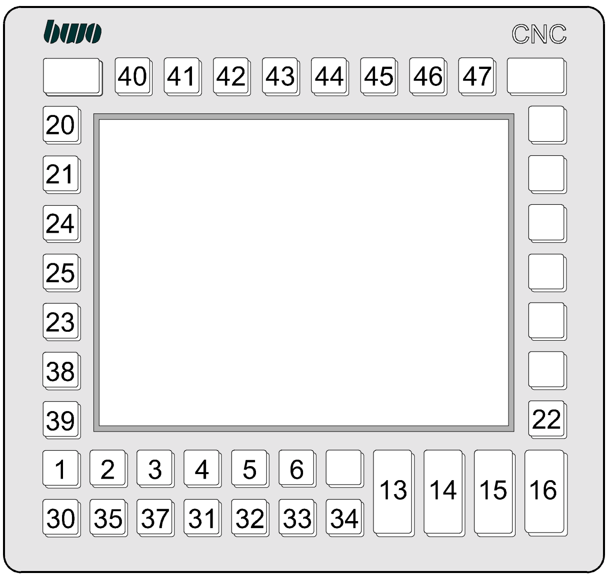
Byte 1	Key code		
Byte 2	0 unpressed		
	1 pressed		
Byte 3	Channel number 1..8		
Byte 4	Panel number	0	1. Panel (default)
		1	2. Panel

Key numbers in CNC 900 and CNC 900C



Key numbers in

CNC 902 / CNC 903
CNC 904 / CNC 905
CNC 910 / CNC 920
CNC 930 / RC 910



8.6 System parameters (continued)

System commands / information

P11138	LEDs 1..12 (bit information)			
	Byte 1	bit 0	0	LED key 40 out
			1	LED key 40 on
		bit 1		key 41
		bit 2		key 42
		bit 3		key 43
		bit 4		key 44
		bit 5		key 45
		bit 6		key 46
		bit 7		key 47
	Byte 2	bit 0		key 48
		bit 1		key 49
		bit 2		key 50
		bit 3		key 51
P11139	LEDs (bit information) extended			
	Byte 1	bit 0	0	LED key 52 out
			1	LED key 52 on
		bit 1		key 53
		bit 2		key 54
		bit 3		key 55
		bit 4		key 56
		bit 5		key 57
		bit 6		key 58
		bit 7		key 59
	Byte 2	bit 0		key 60
		bit 1		key 61
		bit 2		key 62
		bit 3		key 63

8.6 System parameters (continued)

System - memory / Informations

P11140 Free NC memory

P11141 Amount of NC programs

P11142 Number of NC programs

P11145 Number of latest edited NC - program

P11146 Number of latest edited NC - block

P11149 Display mode in the NC - directory
0 There appears no Z - programs in the directory-list
1 In the directory ALL programs are displayed.

P11150 Display mode 2 in the NUMERICAL CONTROL Direktory
Byte 1 bit 0 0 Display of the time stamp and the file size
1 Display of the 1. NC block
2 Displays the date (useful for NC file name length > 14)
3 Display the file size (useful for NC file name length > 14)

Byte 2 bit 0 0 Assortment of the NC Direktories after program numbers
1 Assortment of the NC Direktories alphanumerically
Note: In the case of modification of the assortment algorithm command the consisting Direktory chain to be again created! (q10: 7)

P11152 First block number when numerating a NC program
If not loaded, the first block number is 10.

P11153 Block step width when numerating a NC program
If not loaded, the step width is 10.

P11154 Record number of the NUMERICAL CONTROL of program, starting from which that number again one starts.
NC blocks are changed starting from this record number.
If not loaded: again number starting from the first record number.

8.6 System parameters (continued)

System - memory / Informations

P11158 Proma-menu Control
Byte 1:
Bit 1 0 (default)
When you call the PROMA diagnostics (press '!') Is returned to the last diagnostic image
1 When you call the PROMA diagnostics (press '!') That basic diagnostic image is always returned

P11159 OVERLAY insertion in operating panel surface
0 or -- Overlay operation reset
1.. Overlay operation display

System - Clock

P11160 Second

P11161 Minute

P11162 Hour

P11163 Day

P11164 Month

P11165 Year

P11166 Weekday 0 Sunday
1 Monday
2 Tuesday
3 Wednesday
4 Thursday
5 Friday
6 Saturday

P11169 100 ms timer parameter is incremented in stroke of 100 ms

8.6 System parameters (continued)

I/O circulation Interface 1

P11170 Interfaces Mode (planned, I/O etc.)

P11171 Interfaces Selector (planned, default = operating panel COMxx)
0 hard disk operating panel
1 operating panel COM1
2 operating panel COM2
3 operating panel COM3

P11172 Baudrate standard setting 9600 [Baud]
max 38400

P11173 Data bits standard setting 8

P11174 Stop bits standard setting 2
P11175 Parity check standard setting 0
0 no
1 odd
2 even

P11176 Handshake standard setting 0
0 RTS/CTS
1 Xon/Xoff

P11177 File end signal standard setting 04 (EOT)
Input ASCII sign code
Function At an output (IO-OUTPUT), the file end sign is put as last sign.
At an input (IO-INPUT) the I/O is finished with a received file end sign.

P11178 Controlling information
Bit 0 0 Programs can be overwritten
1 Programs cannot be overwritten over I/O inputs.
A message seems.
Bit 1 0
1 IO circulation program output 785 - compatibly
i.e.. Program number is provided with leading zeros
Bit 2 0
1 IO traffic: Parameter output parameters without text

8.6 System parameters (continued)

I/O circulation Interface 1

P11179 Block test character with I/O traffic
With P11179 the block test character is switched with I/O IN/COutputs.
Method of operation: With activated block test character the data link becomes
between operating console and external data terminal secured.
practical in connection with the archiving program NCARC.EXE.
The block test character is stored in no file, but
in an educated manner from the sender / recipient to the real
time in each case / checks.

0 or -- no block test character
I/O INPUT
The block test character of the data which can be read in is optional.
If a block test character is received, it is checked.
With not correct test character:
M1115 and abort of EA traffic.

I/O OUTPUT
no block test character with data output.

1 block test character activates
I/O INPUT
All read in data must with the block test character provided to be.
With missing test character:
M1114 and abort of EA traffic
With not correct test character:
M1115 and abort of EA traffic.

I/O OUTPUT
All data which can be output become with the block test character
provide.

8.6 System parameters (continued)

I/O circulation Interface 1

P11181 Parameter output mode
Parameter is reset after each knocked against parameter output!

0 or -- Parameter EA output from parameter memory
1 Parameter EA output from FLASH Memory
output: D+, i.e. status are also settinged when rereading in
5 Parameter EA output of all parameters from the parameter memory
output: D+, i.e. status are also settinged when rereading in

q0	...	q999	
q1000	...	q1999	
q2000	...	q...	(number of axes)
P0	...	P5xxx	(user block 1aller of channels)
P7000	...	P9999	(fixed channel parameters of all channels)
P20000	...	P2xxx	(user block 2 of all channels)

10 output of the SAMPLE buffer (see also P11270...)
15 output of the ZSM recordings (see P7950...)
50 output of the oscilloscope memory
output format: Fixed point (ulongs)
20 output of the SERCOS Data which can be protected
21 output of all SERCOS Data
22 output by SERCOS Einzelparameter
50 output of the oscilloscope Speichers
output format: Fixed point (ulongs)

P11183 Info. parameter
actual program number I/O input/output
or current channel number with parameter I/O input

P11184 Info. parameter
actual record number I/O input/output
or current parameter number with parameter I/O input

8.6 System parameters (continued)

I/O circulation Interface 2

P11185 Interfaces 2 (planned)
:
P11199

Remote diagnostics (only for ETH CNC)

P11200 ONLINE / OFFLINE Request
0 switch CNC OFFLINE
1 switch CNC ONLINE
P11200 is cleared at every system startup (as of version 144)

P11201 current ONLINE / OFFLINE - state of the CNC
0 CNC OFFLINE
1 CNC ONLINE

P11202 Entering the message number

P11203 ISERVICE IP - address
If P11203 == 0, IService IP address from the file:
NETCONF is used (from the Version144)

8.6 System parameters (continued)

CAN IO-node

P11209 CAN IO-node

IO-node 11 - 20 Optional

P11210	node 11
P11211	“
P11212	node 12
P11213	“
P11214	node 13
P11215	“
P11216	node 14
P11217	“
P11218	node 15
P11219	“
P11220	node 16
P11221	“
P11222	node 17
P11223	“
P11224	node 18
P11225	“
P11226	node 19
P11227	“
P11228	node 20
P11229	“

8.6 System parameters (continued)

CAN-IO-node

IO-node 01 - 10 Standard (available)

P11230	node 01
P11231	“
P11232	node 02
P11233	“
P11234	node 03
P11235	“
P11236	node 04
P11237	“
P11238	node 05
P11239	“
P11240	node 06
P11241	“
P11242	node 07
P11243	“
P11244	node 08
P11245	“
P11246	node 09
P11247	“
P11248	node 10
P11249	“

8.6 System parameters (continued)

CAN-IO-node

Definitions per node / Example: Node 01

```
P11230:   xx xx 00 01    Should be: specifications from CANCONV -> P11230
          ————           P11230/1 CNC ->PLC
           |  |  |
           |  | +->      configured node / PLC: C1_KON = H
           |  |
           |  +——-       Socket 1 or 2  /  -
           |
           +————         Slot No.  /  -

           note:
           Status bit (byte 2, bit 1) must be set
           Transmit parameters by writing to PLC

P11231:   00 01 01 01    Is: Read about SC130 -> PLC
          ————           P11231 states of PLC -> CNC transmitted.
           |  |  |
           |  |  +-      Present / Installs / PLC: C1_PRES = H
           |  |
           |  +——--      Status / Ready / PLC: C1_BER = H
           |
           +————---General Fault / PLC: C1_KSA = H
```

8.6 System parameters (continued)

I/O circulation **Information for diagnostic purposes**

P11250 Error messages, which are entered into the error message history of the PLC i.e. each detailed error message.
(parameter of the PLC is described)

Byte 1	Message paragraph
Byte 2	Message paragraph
Byte 3	Channel number (1... 8)
Byte 4	Axis paragraph (1... 32)

P11251
:
P11259 Actual pending error messages of the actually selected channel
(parameter of the PLC is described)

Byte 1	Message paragraph
Byte 2	Message paragraph
Byte 3	Channel number (1... 8)
Byte 4	Axis paragraph (1... 32)

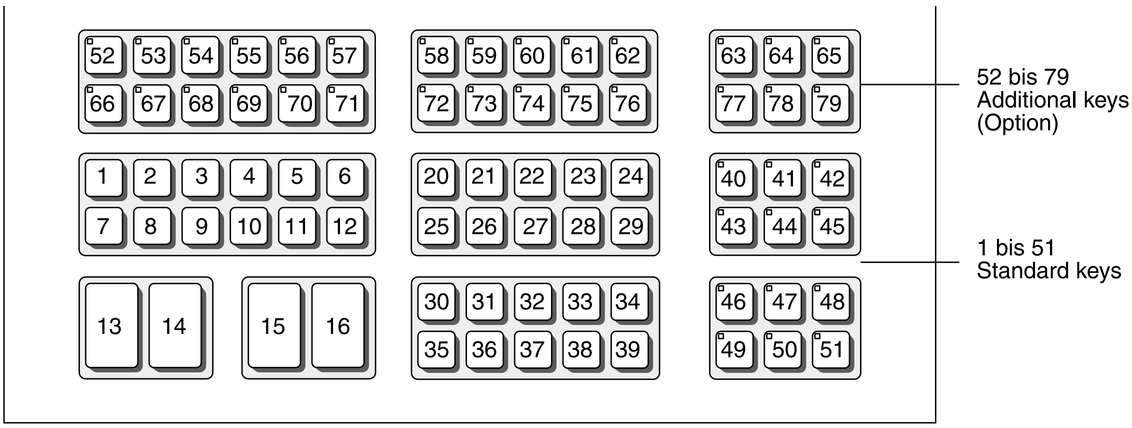
8.6 System parameters (continued)

Information for event logging for diagnostic purposes

P11260 Key information (parameter by the operating console one describes)
total palpation printouts in this parameter are logged

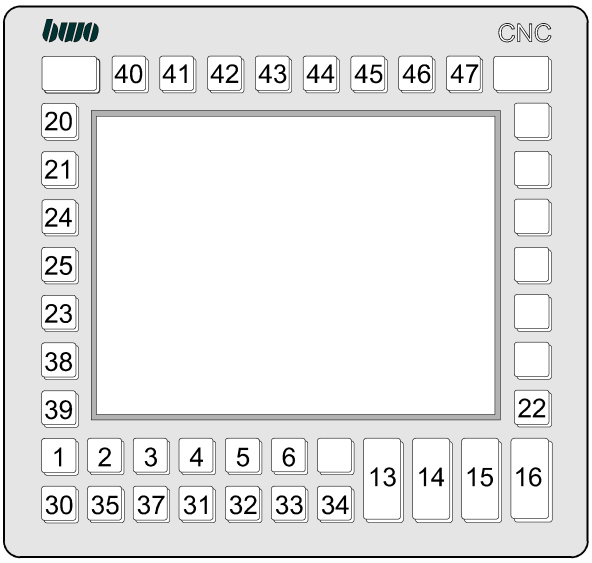
Byte 1	Key paragraph or	(CNC keys)
	ASCII character	(keyboard)
Byte 2 0	Key released	(only with CNC Keys)
1	Key pressed	
Byte 3 0		
Byte 4 0	Keyboard key	
1	Function key	
2	CNC Key	
3	Teachpanel key	

Key numbers in CNC 900 and CNC 900C



Key numbers in

CNC 902 / CNC 903
CNC 904 / CNC 905
CNC 910 / CNC 920
CNC 930 / RC 910



8.6 System parameters (continued)

Information for event logging for diagnostic purposes

P11261 Bildname1 CNC Surface (parameter by the operating console one describes)
Byte 1 CHAR 1
Byte 2 CHAR 2
Byte 3 CHAR 3
Byte 4 CHAR 4

P11262 Bildname2 CNC Surface (parameter by the operating console one describes)
Byte 1 CHAR 5
Byte 2 CHAR 6
Byte 3 CHAR 7
Byte 4 CHAR 8

P11263 Error Message History
PLC describes P11263 with PLC error messages (M3000. 3999)
Error messages are stored in the sample buffer
With the first describing P11263,
P8509 will NOT be recorded more in sample buffer!
Byte 1: message number
Byte 2: „
Byte 3: Channel number (0..8)
Byte 4: axis number (1..32)

P11264 —

P11265 activated panel in multi-panel operation
enabled means: -CNC axes may be moved
-NC programs may be started
-1 all panels active
0 Panel 1 activated (default)
1 Panel 2 activated

P11266 CNC station number
Displaying the definition of "CNCCON" in netconf file
0 no CNCCON-CMD in the netconf file defined; (Stand-alone CNC)
1 "CNCCON 0" in the file netconf defined ==> Station 1
2 "CNCCON 1" in the file netconf defined ==> Station 2

Parameter is described after each power-on routine by the system.

P11267 —
P11268 —
P11269 —

8.6 System parameters (continued)

Event logging for diagnostic purposes

P11270 Activation of the recording ring buffer
(in further Text als SAMPLE buffer marks)
Byte 1
0 Events are written into the SAMPLE buffer
1 SAMPLE buffer is not no more described
100 delete SAMPLE Buffer
Byte 2 :
Byte 3 :
Bit 0 : 0 : Off data logger
1 : Data-logger is switched on (only ETH CPU)
Sample events are sent to the host
(Host: see netconf 'host')
On the host, the program must: 'SERVER.EXE' be active.
Sample data will be in the file: server.log saved.

With data logger is activated ALWAYS will be:
- key codes
- Current screen name
- Logs error messages.
Furthermore, can
- the START / STOP of an NC program will be logged.
(See P11271 for K1; ..)
Bit 1 : 1 : In data-logger mode (only CPU ETH)
Additional recording of the virtual console
(Texts, which are output via the NC program with CMD!)

With active SAMPLE buffer (P11270:0) all CNC Key generally always becomes presses and all messages (CNC messages P8500.. P8503 and SPS messages (P8509 or over P11263) as well as P11279 with a zeitmarke stored.
Memory depth corresponds to 100 (083671 before index C) or 2000 (084564) Recordings.
The available recordings can over I/O circulation with parameter output to be read, if before the output the parameter output mode (P11181) = 10 one settings.

Starting from status 082 of 1.2.99
With a function call in the CNC DLL can this SAM polarizing ring buffer as section NC RAM to be specified.
The size of this buffer is freely definable. (number of Samples)
The available NC memory is reduced accordingly.
Contents of this static ring buffer remain after switching off that Controller preserve. (is with back-up battery like the NC memory)

8.6 System parameters (continued)

Event logging for diagnostic purposes

An output could look as follows:

SAM polarizing DATA

Output:
DATE: 1. 2,99 date / time-of-day of the output
Time: 02:12:57

hh:mm:ss, ms cmd
1 00:00:00,000 ======== system boot =============

2 02:08:39,445 PICTURE: INIT	picture change
3 02:08:42,455 PICTURE: A_INIT	picture change
33 02:11:55,650 C-KEY AX 1	CNC key axis selection 1
34 02:11:56,965 F-KEY F5	function key F5
35 02:11:58,525 KEY '1'	PC key '1'
42 02:12:00,655 F-KEY RETURN	function key RETURN
43 02:12:01,290 F-KEY cursor V	function key cursor
57 02:12:08,990 PARA CNC —> PLC K0 q1137: 65822	parameter transmitted by CNC at PLC transfer on modification
58 02:12:08,990 PAReq CNC —> PLC K1 q1137: 6	parameter transmitted by CNC at PLC parameter was called by PLC at CNC
60 02:12:08,995 PARA CNC <— PLC K1 P8703: 1	parameter transmitted by PLC at CNC
97 02:12:13,820 K0 message: M3498	message M3498
448 02:12:21,015 PROG: K1 P1000 N11160	processing of NC block
449 02:12:21,015 PROG: K1 P1000 N11170	
831 02:12:21,340 K1 M30 —> PLC, NR in the Satz:0,	without branch transfer of m functions at PLC
832 02:12:21,340 K1 END OF BLOCK CNC —> PLC, CNC WAITS for acknowledgement	
834 02:12:21,340 K1 CNC <— PLC SAFRE: 1	
934 02:13:22,540 P11279: 12345	entry over P11279

SAM-POLARIZE-END

8.6 System parameters (continued)

Event logging for diagnostic purposes

P11271 Activation of additional recordings into the SAM polarizing buffer for channel 1
0 or -- No additional recordings

Byte 1 bit 0 1 Parameter transfers between
CNC <—> PLC are recorded.
(q- parameter transfers are recorded when channel 1 is enabled)

bit 1 1 Parameter transfers between
CNC <—> operating panel are recorded.

bit 2 1 Parameter with parameter SAM polarizing identifier
are recorded. (see parameter status Byte2 Bit8)
(parameters from channel 1 become together with
q-parameters recorded)
bit 3 —

bit 4 1 M functions at PLC, end of block, end of record,
M function-acknowledgements and block release
are recorded.
bit 5 --
bit 6 --
bit 7 --

Byte 2 bit 0 1 beginning of a NC block with Programmnumme becomes
recorded.
(pre path acknowledgement in the real time)

bit 1 0 —
1 Recording of programmed NC block positions
(Recording in real time (W2) / machine positions to interpolator)
(Recording in Prerecord-feedback)

bit 2 —
bit 3 — Start / Stop of a CNC program is recorded
bit 4 1 beginning of a NC block with program number one records.
(recording of the preparatory program/record number, pre path)

bit 5 1 recording of the positions programmed in the NC block
(recording in the advance)
bit 6 --
bit 7 --

8.6 System parameters (continued)

Event logging for diagnostic purposes

Byte 3:
- Bit 0 —
- Bit 1 —
- Bit 2 —
- Bit 3 1 Recording of generator data communications over S2 at switch: generator_com
- Bit 4 —
- Bit 5 —
- Bit 6 —
- Bit 7 —

Byte 4:
- Bit 0 —
- Bit 1 —
- Bit 2 —
- Bit 3 —
- Bit 4 —
- Bit 5 —
- Bit 6 —
- Bit 7 —

P11272 Activation of additional recordings into the SAM polarizing buffer for channel 2
P11273 Activation of additional recordings into the SAM polarizing buffer for channel 3
P11274 Activation of additional recordings into the SAM polarizing buffer for channel 4
P11275 Activation of additional recordings into the SAM polarizing buffer for channel 5
P11276 Activation of additional recordings into the SAM polarizing buffer for channel 6
P11277 Activation of additional recordings into the SAM polarizing buffer for channel 7
P11278 Activation of additional recordings into the SAM polarizing buffer for channel 8

P11279 SAM-POLARIZE-entry
With the describing of P11279 an entry in the SAM polarizing buffer is made.

8.6 System parameters (continued)

System diagnostic

P11280 Activating the interface diagnostic
0 Interface diagnostic off
1 PLC <-> CNC interface diagnostic on

P11281...P11289 direction CNC —> PLC
P11290...P11299 direction CNC <— PLC

P11281	Event counter	CNC —> PLC	
P11282	Channel number	CNC —> PLC	
P11283	Parameter number	CNC —> PLC	
P11284	Parameter content	CNC —> PLC	
P11291	Event counter	CNC <— PLC	(q1160 is not indicated and not counted as event.)
P11292	Channel number	CNC <— PLC	
P11293	Parameter number	CNC <— PLC	
P11294	Parameter content	CNC <— PLC	

8.6 System parameters (continued)

System data for Interpolator

P11300 Interpolation nicety [INC/mm]
Specifies the internal path resolution at the Interpolator
0 preset: 10000
Input 100, 1000, 10000
In the case of a modification of this parameter the controller must again be started.

P11301 Position adjustment timer [ms]
In the case of a modification of this parameter the controller must again be started.

P11302 DSI timer [ms]
In the case of a modification of this parameter the controller must again be started.

P11305 Definition of the axis card off set
In the case of a modification of this parameter the controller must again be started.
This axis allocation can by means of P12003... for each axis separately and to be made axis-card-independently.
0 or -- physical axis numbering sequentially

Example:

1. Axis card		physical numbering
ASM	1	1
	2	2
	3	3
2. Axis card		
AAZ4	1	4
	2	5
	3	6
	4	7

1 per card slot 4 physical axes are addressed

Example:

1. axis card		physical numbering
ASM	1	1
	2	2
	3	3
2. Axis card		
AAZ4	1	5
	2	6
	3	7
	4	8

8.6 System parameters (continued)

System data for Interpolator

P11306 Definition for axis cards disable
In the case of a modification of this parameter the controller must again be started.
0 or -- physical axis numbering sequentially

Bit 0-31 No. of the respective slot
e.g.. 0x2 in slot 2 no card (AAZ, DAW, ADW) is detected

P11308 Input system

0 or -- Input system metrically
1 Input system inch, post-decimal positions around a position extends
2 Input system inch, post-decimal positions around two positions extends
3 Input system inch, post-decimal positions around three positions extends

Inch switching
With ' parameter reset ' in the switching on routine becomes
- in the zero point data array,
- all kanalgemappten Achsparametern, (P12153..12157...),
- the tool data interface, (P8110/11/12/13 and P8160/61/62/63),
- and the feed parameters in the channel
the inch bit in the parameter status settinged.
With the axis data of round axes no inch information is settinged. If one is linear axis without inch switching to be, then if the inch bit is to be removed in the q2x54.
After switching the system of units of INCH -> METRIC or METRIC -> INCH only these parameters become into respective the different one.
System of units converted, which the ZOLL bit in the parameter status settinged.

Step-by-step operation:
Incrementation table is preserved, however the actual incrementation becomes always over factor 10 reduces
(see also G170 / G171).

With activated ZOLL Input system, all channels become into this Input system brought.

P11309 System internal command parameter
Interpolator —> CNC
1 trigger for reference status

8.6 System parameters (continued)

Override (1 - 9)

The override allocation is defined depending on the traverse mode per axis (see P12125..).

General: From the operating panel the corresponding override value is written in P11311 (content: 0.. 255). P11311 is a procedure - parameter. In the started system sub-routine the override-schedule is accessed according to P11314 (pointer on table). The result is copied to P11310. P11310 is limited through P11312/P11313.

	Override 1		
P11310	Actual value	[%]	value from table
P11311	Actual value		from digital-to-analogue converter
P11312	Minimum limit	[%]	e.g. 0 %
P11313	Maximum limit	[%]	e.g. 120%
P11314	Pointer on override-table If pointer on override-schedule is cleared or =0, the override is active continuously. Pointer-number shows on a q-parameter.		

P11315	Override 2	
P11320	Override 3	
P11325	Override 4	(Teachpanel)
P11330	Override 5	(fixed value 0%)
P11335	Override 6	(fixed value 10%)
P11340	Override 7	(fixed value 50%)
P11345	Override 8	(fixed value 100%)

8.6 System parameters (continued)

Override (1 - 9)

P11370	Override-table	Amount: Preset 27 values
P11371	0%	
P11372	1%	
P11373	2%	
P11374	5%	
P11375	10%	
P11376	15%	
P11377	20%	
:		
P11397	120%	

8.6 System parameters (continued)

Hand-wheels (1 - 8)

Hand-wheel 1

P11400 Hand-wheel connected on axis (physical axis number)
contents: 1...32 physical axis number
-1 handwheel in the operating panel integrates (e.g.: RC 910)

P11401 Hand-wheel works on axis (physical axis number)

P11402 Activating the hand-whel
0 or -- Hand-wheel not active
1 Hand-wheel active

P11403 Hand-wheel resolution [INC/mm]
The pulses of the measuring system are quadrupled in the AAZ.
Example: resolution 1000, At 1000 INC 1mm is driven.
The sign determines the traverse direction.

P11404 Actual hand-wheel - multiplier
With help of the hand-wheel - multiplier the hand wheel- pulses are multiplied.
Example: P11404 = cleared or 0 or 1, 1000 INC -> 1mm distance
P11404 = 10 1000 INC -> 10mm distance

P11405 Switch to next multiplier
1 switch to next

P11406 Pointer on multiplier, parameters is used from the system.
P11407 Multiplier 1
P11408 Multiplier 2
P11409 Multiplier 3

P11410 Hand-wheel 2
P11420 Hand-wheel 3
P11430 Hand-wheel 4
P11440 Hand-wheel 5
P11450 Hand-wheel 6
P11460 Hand-wheel 7
P11470 Hand-wheel 8

8.6 System parameters (continued)

Couplings (1 - 8)

General: With active coupling the machine dynamics and the max. axis speed of the masters axis and slave axis are adapted to the admissible values

	Coupling 1	
P11480	Master - axis	(physical axis number)
P11481	Slave - axis	(physical axis number)

P11482 Coupling mode

Byte 1	$01	Machine coordinates coupling	
Byte 2	$01	Rigidity coupling	
Byte 3	$00	Type of status coupling	master <——> Slave
	$01	Differential coupling	master ——> Slave

P11483 Coupling modification
0 or -- Coupling correction on
1 Coupling correction off
2 Coupling correction off, however with coupling error control procedure actively

P11484	Coupling error limit	[mm, degree]
P11485	Coupling error supervision time	[ms]
P11486	Coupling correction P share	[1/s]
P11487	coupling correction I share	[ms]
P11490	Synchronization position of the Masteraxis	[mm, degree]
P11491	Synchronization position of the Slaveaxis	[mm, degree]
P11492	Transmission ratio - relationship masteraxis	
P11493	Transmission ratio - relationship slave axis	
P11494	Differential constant	

P11495 Coupling on / off
1 Coupling on (PLC:KOP1)
2 Coupling synchronization on (see P11490 or P11491) (PLC:KOPSYN1)

P11496 Coupling activated
1 Switched on coupling (PLC:KOPE1)
2 Coupling performed synchronization (PLC:KOPSYE1)

8.6 System parameters (continued)

Couplings (1 - 8)

P11500 Coupling 2
:
P11520 Coupling 3
:
P11540 Coupling 4
:
P11560 Coupling 5
:
P11580 Coupling 6
:
P11600 Coupling 7
:
P11620 Coupling 8
:

8.6 System parameters (continued)

Spindle (1 - 8)

	Spindle 1		
P11640	Spindle works on axis	(physical axis number)	
P11641	Reference axis for G96	(physical axis number)	
	If P11641 = 0, then reference position in P11642 is valid.		
P11642	Reference position for G96		[mm]
	e.g. grinding wheel diameter		
P11643	Reference factor for G96		[mm/min]
	0 or -- mm/min		
	1000 m/min		
P11644	Maximum speed for G96		[mm/min]
P11645	Maximum speed for G97		[rpm]
P11646	Reference factor for G97		[rpm]
	0 or -- rpm		
	1 degree/min		
P11647	Definition of speed	(G96, G97)	
	Presetting 0 = G97		
P11648	Programmed speed		[rpm or mm/min]
P11649	Actual speed		[rpm or mm/min]
P11650	Spindle in/out		
	0 or -- Spindle out		
	1 Positive direction of rotation		(PLC:SPIRE1)
	-1 Negative direction of rotation		(PLC:SPILI1)
	2 Positively spindle reset		(PLC:SPIRR1)
	-2 Negatively spindle reset		(PLC:SPIRL1)
P11651	reached rotation speed		
	Byte 1		
	$01 : reached rotation speed		(PLC:SPIDZE1)
	Byte 2		
	$01 : Achieved spindle alignment position		(PLC:SPIRPE1)
	Byte 3		
	$01 : spindle runs		(PLC:SPILFT1)
	Byte 4		
	$01 : Speed reached 0		(PLC:SPIDZN1

8.6 System parameters (continued)

Spindle (1 - 8)

P11652 Spindle resetting position

P11653 Speed monitoring tolerance limit [%]
0 or -- actual speed monitoring switched off
1...100 i.e. P11651=1 achieves the info. speed
settinged if programmed speed equal actual speed
and the deviation between actual speed (Messystem) and
smaller than the input value is appropriate for programmed speed.

P11654 Rotational speed reaches 0 monitoring tolerance limit (min-1)
—, 0 off monitoring
1.xx the info: rotation speed fallen below
P11651/Byte4 = $01 is set if the actual speed is smaller than the entry value.

P11660 Spindle 2
:
P11680 Spindle 3
:
P11700 Spindle 4
:
P11720 Spindle 5
:
P11740 Spindle 6
:
P11760 Spindle 7
:
P11780 Spindle 8
:

8.6 System parameters (continued)

Data for Robot-system 1

P11800 Linear axes (physical axis numbers)

At 5 axes transformation (Robot)
e.g. the first 3 axes (X, Y, Z)
Byte 3, 2, 1 03 02 01 Hex
Byte 4 direction change (Bit 3, 2, 1)

At 3 axes transformation
e.g. the first 2 axes (X, Y)
Byte 3, 2, 1 00 02 01 Hex
Byte 4 direction change (Bit 2, 1)

At 6 axes transformation
e.g. the first 3 axes (X, Y, Z)
Byte 3, 2, 1 03 02 01 Hex
Byte 4 direction change (Bit 3, 2, 1)

P11801 Robots rotation axes (physical axis numbers)

At 5 axes transformation (Robot)
e.g. the 4th and 5th axes (A, B)
Byte 3, 2, 1 00 05 04 Hex
Byte 4 direction change (Bit 3, 2, 1)

At 3 axes transformation
e.g. the 3rd axis (A)
Byte 3, 2, 1 00 00 03 Hex
Byte 4 direction change (Bit 1)

At & axes transformation (Robot)
e.g. the 4th, 5th and sixth axes (A, B, C)
Byte 3, 2, 1 06 05 04 Hex
Byte 4 direction change (Bit 3, 2, 1)

P11802 At 5 axes transformation (Robot)
Offset x, rotation axis - tool axis [mm]

At 3 axes transformation
Offset radius, tilt - tool peak [mm]

8.6 System parameters (continued)

Data for Robot-system 1

P11803	At 5 axes transformation (Robot) Offset y, rotation axis - tilt	[mm]
	At 3 axes transformation Offset angle, tilt - tool peak	[degree]
P11804	Offset z, tool carrier length	[mm]
P11805	Offset, tilt - tool axis	[mm]
P11806	Angle, rotation axis - tilt	[degree]
P11807	Angle, at which the tool is downwards	[degree]

8.6 System parameters (continued)

Data for Robot-system 2

P11810 Robot linear axes (physical axis paragraphs)
To contents see P11800

P11811 Robot rotation axes (physical axis paragraphs)
To contents see P11801

P11812 Offset x [mm]
To contents see P11802

P11813 Offset y [mm]
To contents see P11803

P11814 Offset z [mm]
To contents see P11804

P11815 Offset, drag axis - tool axis [mm]
To contents see P11805

P11816 Angle, axis of rotation - drag axis [degrees]
To contents see P11806

P11817 Angle, with that the tool downwards shows [degrees]
To contents see P11807

8.6 System parameters (continued)

Data for polar-system

P11820	Polar axis e.g. the first 2 axes (X, Y) Byte 3, 2, 1 00 02 01 Hex Byte 4 direction change (Bit 2, 1)	(physical axis number)	
P11821	Offset r0		[mm]
P11822	Offset w0		[mm]
P11823	Offset v0		[mm]

8.6 System parameters (continued)

Data for angle system

P11830	Angle axes e.g. the first 2 axes (X, Y) Byte 3, 2, 1 00 02 01 Hex	(physical axis paragraph)
P11831	Angle value	[degrees]

8.6 System parameters (continued)

Configuration parameters for robot systems

P11850 Mechanics

	$0100	6-Achsen-Knickarm even
	$0101	6-Achsen-Knickarm diagonally
	$0200	3-Achsen-Scara
	$0201	2-Achsen-Scara

P11852 Pendulum axis paragraph
P11853 Pendulum angle

Geometry parameter of the respective robot system

P11854	lever	1
P11855	lever	2
P11856	lever	3
P11857	hand	1
P11858	hand	2
P11859	hand	3

8.6 System parameters (continued)

Parameter for joystick

P11880 joystick definition
Byte 1 Calibrationing number
Byte 2 Slot number

-1 joystick logged on with CNC910 Panel
(to the activation start system again)

P11881 joystick direction definition
(with CNC900 and CNC910)
Byte 1 1 direction 1 invert
Byte 2 1 direction 2 invert
Byte 3 1 direction 3 invert

P11882 joystick axis selection
Byte 1 physical axis number direction 1
Byte 2 physical axis number direction 2
Byte 3 physical axis number direction 3

Current joystick feed is in the P12117.
Feed value = (joystick deflection * manual feed)

Parameter for welding seam tracing

P11890 Pointer on table

Parameter for Teachfunktion

P11891 Pointer on table

Parameter for external Robot simulation

P11892 0 Connection off
1 Connection with ROBOT actively

8.7 Axis parameters

Each physical axis occupies a parameter block of 200 parameters.

In the system has physical axis	the area	In the channel has logical axis	the area
1.	q2000 ... q2199,	1.	P12000 ... P12199,
2.	q2200 ... q2399,	2.	P12200 ... P12399,
3.	q2400 ... q2599,	3.	P12400 ... P12599,
4.	q2600 ... q2799,	4.	P12600 ... P12799,
5.	q2800 ... q2999,	5.	P12800 ... P12999,
6.	q3000 ... q3199,	6.	P13000 ... P13199,
7.	q3200 ... q3399,	7.	P13200 ... P13399,
8.	q3400 ... q3599,	8.	P13400 ... P13599,
9.	q3600 ... q3799,	9.	P13600 ... P13799,
10.	q3800 ... q3999,	10.	P13800 ... P13999,
11.	q4000 ... q4199,	11.	P14000 ... P14199,
12.	q4200 ... q4399,	12.	P14200 ... P14399,
13.	q4400 ... q4599,	13.	P14400 ... P14599,
14.	q4600 ... q4799,	14.	P14600 ... P14799,
15.	q4800 ... q4999,	15.	P14800 ... P14999,
16.	q5000 ... q5199,	16.	P15000 ... P15199,
17.	q5200 ... q5399,	17.	P15200 ... P15399,
18.	q5400 ... q5599,	18.	P15400 ... P15599,
19.	q5600 ... q5799,	19.	P15600 ... P15799,
20.	q5800 ... q5999,	20.	P15800 ... P15999,
21.	q6000 ... q6199,	21.	P16000 ... P16199,
22.	q6200 ... q6399,	22.	P16200 ... P16399,
23.	q6400 ... q6599,	23.	P16400 ... P16599,
24.	q6600 ... q6799,	24.	P16600 ... P16799,
25.	q6800 ... q6999,	25.	P16800 ... P16999,
26.	q7000 ... q7199,	26.	P17000 ... P17199,
27.	q7200 ... q7399,	27.	P17200 ... P17399,
28.	q7400 ... q7599,	28.	P17400 ... P17599,
29.	q7600 ... q7799,	29.	P17600 ... P17799,
30.	q7800 ... q7999,	30.	P17800 ... P17999,
31.	q8000 ... q8199,	31.	P18000 ... P18199,
32.	q8200 ... q8399	32.	P18200 ... P18399

In the channel descriptor physical axes are attached to the channel axes (q110 ... q141, Byte 3).

8.7 Axis parameters (continued)

Machine data for the 1st. axis

P12000 Connected axis

0 or —	axis not connected		
1	axis connected		
2	axis may not be driven, otherwise message seems M2020		
3	Axis controlled with measuring system and the reference point taken		
-1	axis simulation, Messystem not necessarily, Command value output according to actual axis rate, point of reference taken.		
-2	Axis is simulated Measuring system is not necessary Set value output with accordingly act. axis speed Taken reference point without hardware detection		

P12001 Round axis

0 or —	no round axis		
1	round axis		
2	round axis drives shortest path		
3	round axis drives pursuant to	G5	positively
		G6	negatively
		G7	shortest path

P12002 Diameter axis

0 or —	no diameter axis
1	diameter axis

P12003 Axis module address

0 or —	address recognition internally	
Byte 1		
	$xx	axis paragraph 1,,8 (axis on the axis module)
Byte 2		
	$xx	slot paragraph 1... 32
Byte 3		
	$xx	drive address with Sercos Axes 1... 255
	$00	drive address equal axis paragraph

8.7 Axis parameters (continued)

Machine data for the 1st. axis

P12004 Maximum axis speed [mm/min, degree/min]
with Sercos axes transfer at final drive is affected as:
S-0-0091 speed limiting value bipolar

P12005 Slope speed 1 [mm/min, degree/min]
0 or — Slope rate = 10 % max. axis speed
to Slope to activate see P12136

P12006 Slope speed 2 [mm/min, degree/min]
0 or — Slope rate = 20 % max. axis rate
to Slope to activate see P12136

P12008 Machine dynamics 1 [ms]
0 or — default = 250 msec

The dynamics determine the time in that the axis of 0 on max axis rate one accelerates or one brakes. Always works with manual operation and automatic (G0).
With Sercos axes the transfer to the final drive is affected as:
S-0-0138 acceleration bipolar,
S-0-0042 reference driving acceleration.

P12009 Machine dynamics 2 [ms]
0 or — default = dynamics 1
dynamics 2 > dynamics 1, then always work dynamics 2 with path driving (G1, G2, G3, G50) not with G0.

P12010 Dynamic definition
0 or — linear axis acceleration drive.
1 sine axis acceleration drive.

8.7 Axis parameters (continued)

Machine data for the 1st. axis

P12012 Measuring system - dissolution numerator [INC/mm, degree]
Input only integral !

P12013 Measuring system - dissolution denominators
The pulses of the measuring system are quadrupled in the AAZ.
Example: measuring system with 2500 INC/r, pitch of the spindle of 10mm/r
i.e. 1 revolution -> 2500INC * 4 = 10000 INC / 10mm
Input: P12012 1000 ; P12013 1
Input only integral !

P12014 Measuring system - counting reverse

Byte 1	$01	inverted
Byte 2	$xx	number of invalid data bits on the right hand side
Byte 3	$xx	number of invalid data bits on the left hand side (ref. 32 Bit)

Example of an absolute encoder

32 1

xxxxxxxx00000000000000000000xxxx

xxxxxxxx xxxx
8-bits to the left fade out 4-Bit on the right fade out

Input P12014: $080400

P20015 Measuring system - absolute encoder (AZA)

Byte 1	$xx	number of data bits that have to be transmitted 00H = 1. Bit ... 1FH = 32. Bit
Byte 2	$xx	Transmitting frequency 00H = 250khz, 01H = 330kHz, 02H = 500 kHz, 03H = 1mhz
Byte 3	$xx	Data mode 00H = binary code, 01H = Gray code
Byte 4:	$01	Actual position is settinged in ESR on old value from P12152 i.e.. Axis reported (indicator REF1=1)

8.7 Axis parameters (continued)

Machine data for the 1st. axis

P12016 Exact stop boundary finely [mm, degree]
0 or — default = square root (1 / measuring system resolution)

Example: resolution scrutineer 1000
denominator 1
Default = sqrt (1 / (1000 / 1)) = 0,0316 mm

P12017 Exact stop boundary roughly [mm, degree]
0 or — default = 8 x exact stop boundary finely

Mode of operation: An axis starts a programmed position with exact stop, i.e. programmed position is achieved:
- lag distance (Sollpos being) is smaller than exact stop boundary gross, in such a way the drift correction is activated.
- lag distance is small exact stop boundary finely, in such a way the signal driving instruction (P12188) is reset.

P12018 Downtime monitoring [ms]
0 or — default = 3000 ms

If this monitoring time is exceeded, error message (M2022) is affected.

P12020 KV factor [1/s]
0 or — default = 16 x 1/s

The KV factor is a valuation (p-proportion) the Lagereglung.
With Sercos Axes transfer at final drive is affected as:
S-0-0104 KV factor attitude control

P12021 P-reinforcement speed controlling device
0 or — default = speed regulation in the drive

With Sercos Axes transfer at final drive is affected as:
S-0-0100 p-proportion of speed controlling devices

P12022 I-reinforcement (reset time) [ms]
0 or — default = 10000ms (drift correction)

With Sercos axes transfer at final drive is affected as:
S-0-0101 i-proportion of speed controlling devices

8.7 Axis parameters (continued)

Machine data for the 1st. axis

P12024 Lag error limit [%]
0 or — default = 50% over debit lag distance

With the help of the axis speed and the KV's that can
Lag distance to be calculated.
Lag distance [mm] = feed [mm/min] / (60 x KV [1/s])

P12025 Lag error limit monitoring time [ms]
0 or — default = 50ms
-1 error limit monitoring switched off

If this lag is exceeded, error message (M2001) is affected

P12026 Admissible command value output [volts]
Around this monitoring to switch on P12026 must be > 0, and P12142 = 1.
With achieving this lag is affected error message (M2002)
and analog output = 0 settinged.

P12027 Maximum command value output [volts]
0 or — default = 9,5V

Those max. output voltage is achieved, if these
axis with max. axis speed drives.

P12028 Command value output direction reversal
1 inverts

P12029 Command value output axis exchange
0 command value output to this axis
1,,32 command value output to indicated axis
(caution: Duplicate of axes avoid)

8.7 Axis parameters (continued)

Machine data for the 1st. axis

P12030	Software limits positive; -- no software end position check With Sercos axes transfer at final drive takes place as: S-0-0049 position limiting value positively	[mm, degree]
P12031	Software limits negative; -- no software end position check With Sercos Axes transfer at final drive takes place as: S-0-0050 position limiting value negatively	[mm, degree]
P12032	Emergency - position absolute absolute measure, refered to machine zero point approach emergency position see P12131	[mm, degree]
P12033	Emergency - position relative relative measure, refered to machine-location approach emergeny position see P12131	[mm, degree]
P12034	Basic position absolute absolute measure, in reference to machines - zero point approach to basic position see P12131	[mm, degree]
P12035	Basic position relative relative measure, refered to machine-location approach basic position see P12131	[mm, degree]
P12036	Fixed position absolute measure, refered to machine zero point approach fixed position see P12132	[mm, degree]
P12037	Pendling position relative measure, refered to machine-location approach pendling position see P12132	[mm, degree]
P12038	Dwell for pendling position 1 0 or cleared Pendling position 1 with exact stop approach pendling position see P12132	[s]
P12039	Dwell for pendling position 2 0 or cleared pendling position 2 with exact stop approach pendling position see P12132	[s]

8.7 Axis parameters (continued)

Machine data for the 1st. axis

P12040 Reference point offset [mm, degree]
When taking the reference points (e.g. in the moment of T0), the actual position (P12151) is loaded with P12040. This means if this axis is approached, the axis is standing on it's reference point.

P12041 Modulo measure [degree]
0 or -- Presetting 360°

P12042 Acceptable distance (reference switch - zero pulse) [mm, degree]
when exceeding -> error message

P12043 Reference distance (distance coded measuring system) [mm, degree]

P12044 Reference driving logic

Byte 1
- $01 Release of the reference accommodation in positive driving direction (see also P12130, byte 1)

Byte 2
- $01 Release of the reference accommodation in negative driving direction (see also P12130, byte 2)

Byte 3

Bit0	0	with switch edge
	1	with switch and nullimpuls
Bit4	0	standard measuring system
	1	2. Measuring system

Byte 4

Bit0	0	switch positively switching
	1	switch negatively switching
Bit1	0	reference switch info. (P12180, byte 2) always actually
	1	reference switch info. (P12180, byte 2) becomes after point of reference taken no more does not update.

Conditions for the reference accommodation

P12044		Reference driving logic specify
P12079	Byte3	Sercos reference driving control parameter specify

P12130 starts reference accommodation,
Reference accommodation acknowledgement P12180,
so long reference travel actively,
- P12130 Byte 3 $01 / PLC:REPOF1 = 1 reference position release
is the drive to be driven as follows
- adjusted without restarting
P12140 Byte 2 $01 / PLC:NGOW1 = 1

8.7 Axis parameters (continued)

Machine data for the 1st. axis

P12045 Sensor logic
Byte 1
$01: Release of measuring position accommodation in positive driving direction
Byte 2
$01: Release of measuring position accommodation in negative driving direction

Byte 4

Bit0	0	switch edge positively effectively
	1	switch edge negatively effectively
Bit4	0	standard pressure foot
	1	2. Sensor

Conditions for the measuring position accommodation

P12045		Pressure foot logic specify
P12079	Byte4	Sercos sensor expensive parameter specify
P12082		Sercos sensor of 1 measuring pos parameter specify (S-0-0130 (pos edge), S-0-0131 (neg edge)
P12084		Sercos sensor of 2 measuring pos parameters specify (S-0-0132 (pos edge), S-0-0133 (neg edge)

P8716, P8717, P12131 start measuring accommodation with M26,
Measuring accommodation acknowledgement P8696, P12181
Measuring position P12152

P12046 Error logic
Byte 1
$01 Wire break T0, T1, T2, inverted signals missing
Byte 2
$01 UAS signal responded (interference, contamination)
Byte 3
$01 Measuring system frequency too largely
Byte 4
$01 Sercos messages actively
$02 Display drive status

P12047 Emergency stop logic
Bit 0 - 31 According to axis 1 - 32
A emergency stop position (P12032, P12033) is started,
the axes indicated in P12046 are stopped.

8.7 Axis parameters (continued)

Machine data for the 1st. axis

Drift correction

P12049 Drift correction (integral action time in P12022)
- 0 drift correction only in standstill
- 1 drift correction always active
- 2 drift correction switched off

8.7 Axis parameters (continued)

Machine data for the 1st. axis

P12050	Feed forward correction 0 or — Feed correction out Example: P12050:50% the actual lag distance is bisected P12050:100% the actual lag distance close zero i.e.. Axis drives error limits freely	[%]
P12051	Feed forward delay	[ms]

8.7 Axis parameters (continued)

Machine data for the 1st. axis

lot correction

P12052 Backlash correction [mm, degree]
0 or — Backlash correction out

Input value is bisected for both driving directions
Example: Input 0.042mm
- Korretur +0,021 with positive drive
- Korretur -0,021 with negative drive

P12053 Backlash correction time [ms]
0 or — Backlash correction in P12052 works as static value
> 0 Backlash correction in P12052 works as dynmischer value over the time after everyone, input in P12053 travel direction changes
< 0 Backlash correction in P12052 works as dynmischer value over the time after everyone, input in P12053 start of a movement

Correction dynamics

This dynamics cause that modifications of the axis correction (backlash -, measuring system -, upward gradient correction) with this admissible dynamics to be executed.

P12055 Correction dynamics [ms]
0 or — default (machine dynamics 1 / 2)

8.7 Axis parameters (continued)

Machine data for the 1st. axis

Measuring system correction with 2. Measuring system

P12056 Measuring system correction datum axis (2. Measuring system)
Byte 1
$xx 1 - 32 physical axis paragraph 2. Measuring system
2. Measuring system becomes by 1. Measuring system reports.
Debit 2. Measuring system reference system its,
so P12044 must: (byte 4, Bit4=1) to be settinged.
1. Measuring system becomes by 2. Measuring system reports.

Byte 2
$xx: 1 - 32 physical axis paragraph 2. Measuring system
2. Measuring system as exchange axis,
i.e.. Messsytemposition (all axis positions)
2. Measuring system become on the 1. Measuring system
rerouted and displayed.

Byte 3
$xx: 1 - 32 physical axis paragraph 2. Measuring system
2. Measuring system as scrutineer axis,
i.e.. Messsytem scrutineer 2. Measuring system
become on the 1. Measuring system rerouted.

Example of activation 2. Measuring system:
q 118: $00060000 (6th axis log on)
P13001 if round axis
P13012, P13013, P13014 Messystem resolution

P12057 Measuring system correction reset time [ms]
0 or — Correction off
> 0 Correction on

P12058 Measuring system correction admissible correction [mm]
0 or — Default 10mm

8.7 Axis parameters (continued)

Machine data for the 1st. axis

Measuring system correction with 2. Measuring system

P12059 Measuring system correction with internal correction table

0 or —	Function not actively	
> 0	Measuring system correction with correction table Input of the distance of the corrections	[mm]

Note: This function should be only activated, if those resolution 2. Measuring system more roughly is than the resolution too correcting axis.
(axis for swinging would be lively)

In order to structure the correction table, must after the activation the axis uniquely from the negative software end position to the positive software end position to be moved.

8.7 Axis parameters (continued)

Drive data for digital drives 1st. axis

Pitch correction

P12060 Pitch correction pointer on table
Parameter (q-NR.) where table starts.

P12061 Pitch correction number of corrections

0 or —	Correction out
> 0	Correction

P12062 Pitch correction table offset
Table access is permitted only with values of item between the end positions.
Thus the table offset must become so in against that this applies.
Offset = - (machine position (ßt measured value) / distance of the corrections)
Example: P12062: -6 = (P12150: 120,000) / P12063: 20,000)

P12063 Pitch correction distance of the corrections [mm]

P12064 Pitch correction datum axis (interaxis correction, cross correction)
Input physical axis paragraph

0 or —	Correction algorithm uses the axis position (q2x50) the personal axis, over in the correction table the suitable to obtain steigungsfehler.
1...32	Physical axis paragraph correction algorithm uses the axis position (q2x50) the datum axis, over in the correction table the suitable to obtain steigungsfehler.

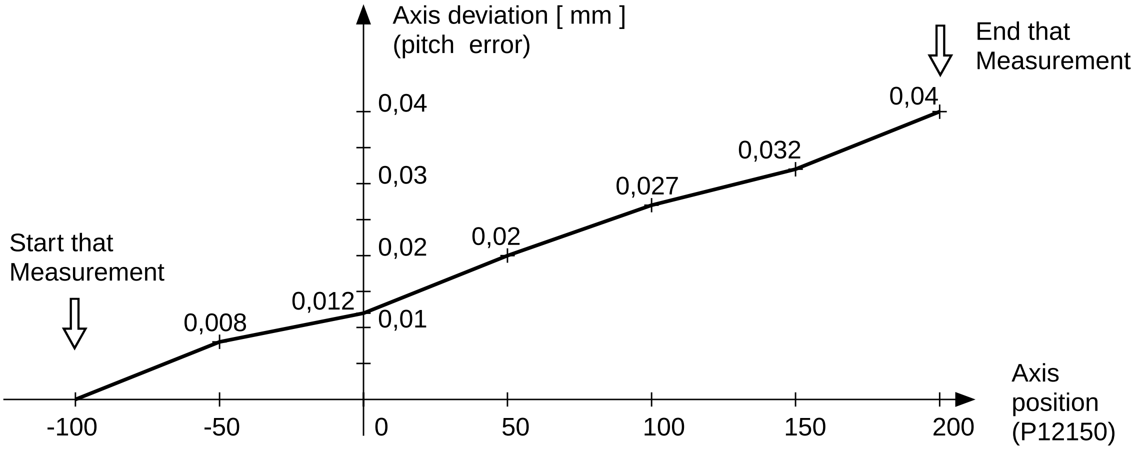
Notes to the upward gradient correction:
To obtain if the machine position of the axis becomes larger than in the correction table covered correction area, the correction algorithm ' interconnects ' start and end the correction table around the entspechenden correction value,
i.e.. Correction values repeat themselves.

Recurring corrections can so in a compact correction table to be stored.
Start and end of the correction table should contain thereby the same values, since it can lead otherwise within this area to striking the axis.

8.7 Axis parameters (continued)

Drive data for digital drives 1st. axis

Example of pitch correction



Allocation of the parameters

				table parameter	of values
Pointer on table	P12060	20000	>	q20000	0
Number of corrections	P12061	7		q20001	0,008
Table offset	P12062	2		q20002	0,012
Distance of the corrections	P12063	50 [mm]		q20003	0,02
				q20004	0,027
				q20005	0,032
				q20006	0,04

8.7 Axis parameters (continued)

Drive data for Sercos

P12070	Sercos parameter (“S”, “ p”)	*
P12071	Sercos parameter (record)	*
P12072	Sercos parameter (paragraph)	*
P12073	Sercos parameter (attribute)	*
P12074	Sercos parameter (value)	*
P12075	Sercos parameter (minimum) *	
P12076	Sercos parameter (maximum)	*

* In the menu: parameter / axis - parameters / drive - parameter contents of all Sercos (s) - product (p) - know parameters over input - boxes, output - boxes to be displayed and changed.

8.7 Axis parameters (continued)

Drive data for Sercos

P12078 Sercos drive operating modes (selection over P12144, byte 4)
Byte 1 S-0-0032 main operating mode
- $02 Speed control
- $03 Position adjustment error limit-afflicted with generator 1(Motor)
- $04 Position adjustment error limit-afflicted with generator 2(Ext.)
- $0b Position adjustment error limit-free with generator 1(Motor)
- $0c Position adjustment error limit-free with generator 2(Ext.)

Byte 2 S-0-0033 type of subsidiary establishment 1
- $02 Speed control
- $03 Position adjustment error limit-afflicted with generator 1(Motor)
- $04 Position adjustment error limit-afflicted with generator 2(Ext.)
- $0b Position adjustment error limit-free with generator 1(Motor)
- $0c Position adjustment error limit-free with generator 2(Ext.)

Byte 3 S-0-0034 type of subsidiary establishment 2
- $xx: (planned)

Byte 4 S-0-0035 type of subsidiary establishment 3
- $xx: (planned)

8.7 Axis parameters (continued)

Drive data for Sercos

P12079 Sercos drive definitions

Byte 1

Bit0	0	Position data in absolute absolute
	1	Position data in modulo format
Bit1	0	Speed data in m/min
	1	Speed data in U/mim

Byte 2

$xx		External measuring system (paragraph of the axis)

Byte 3 Sercos S-0-147 reference driving control parameter

Bit0	0	Clockwise rotation of the motor shaft
	1	Anti-clockwise turn of the motor shaft
Bit3	0	Reference take with motor generator
	1	Reference take with external generator
Bit5	0	Reference switch is analysed
	1	Reference switch is not analysed
Bit6	0	Reference label is analysed
	1	Reference label is not analysed
Bit7	0	Any position after reference take
	1	Point of reference after reference take

Byte 4 Sercos S-0-169 sensor control parameter

Bit0	0	Positive edge	sensor 1 is not analysed
	1	Positive edge	sensor 1 is analysed
Bit1	0	Negative edge	sensor 1 is not analysed
	1	Negative edge	sensor 1 is analysed
Bit2	0	Positive edge	sensor 2 is not analysed
	1	Positive edge	sensor 2 is analysed
Bit3	0	Negative edge	sensor 2 is not analysed
	1	Negative edge	sensor 2 is analysed
Bit4	0	Position actual value to operating mode (S-0-0051/S-0-0053) referred	
	1	Position actual value always to S-0-0051 referred	
Bit5	0	Release individual measuring for	sensor 1
	1	Release subsequent measurement for	sensor 1
Bit6	0	Release individual measuring for	sensor 2
	1	Release subsequent measurement for	sensor 2

8.7 Axis parameters (continued)

Drive data for Sercos

P12080 Sercos phase switching (acknowledgement in P12081)
- 0 Switching on phase
- 2 Parameter mode
- 4 Operating mode

P12081 Sercos system status (acknowledgement of P12080) (PLC:SZSA_01)
- $E001 Sercos phase 0
- $E002 Sercos phase 1
- $E003 Sercos phase 2
- $E004 Sercos phase 3
- $E005 Sercos phase 4
  Sercos drives are ready for use,
  other values show entsprechene intermediates phase on

P12082 Sercos cyclic actual value 3 request
Paragraph by the Sercos parameter
z.B.:130 (S-0-0130 measured value 1-postiv)

P12083 Sercos cyclic actual value 3 acknowledgement
Worth from Sercos parameter
e.g.: Worth to the s-paragraph (S-0-0130 measured value 1-positiv),
which over P12082 was called)

P12084 Sercos cyclic actual value 4 request
Paragraph by the Sercos parameter
z.B.:84 (S-0-0084 torque actual value)

P12085 Sercos cyclic actual value 4 acknowledgement
Worth from Sercos parameter
e.g.: Worth to the s-paragraph (S-0-0084 torque actual value)
which over P12084 was called)

P12089 Sercos speed standardisation
- 0 or — Standart Sercos priority
- > 0 Relation between motor priority and load priority

8.7 Axis parameters (continued)

Area monitoring axes

Area 1

P12100 Channel paragraph
Byte 1:
0 or — P12101 and P12102 = q-parameter
1-8 P12101 and P12102 = p-parameter

Byte 2:
$xx Number of the reference axis for which the monitoring area is to apply

Byte 3:
$01 CNC monitored over the area
(Stops with error message: reached full scale position)
$10 Value in P12103 affects in the + and- -> marker field K1B1E1
$20 Value in P12104 affects in the + and- -> marker field K1B1E1

Byte 4:
0 Control position of P12102 Content: Pointer to parameter
$01 Control position is a value

P12101 Reference position
Contents: Pointer on parameters
0 or — Reference position = internal machine position (q2150)
-2150 Reference position = internal machine position (q2150)
-2152 Reference position = internal machine position (q2152)
With the reference position (internal) are these positions immediately for the order, over parameters only time-delayed.

P12102 Control position
Contents: Pointer on parameters
0 or — Monitoring logic not actively
Note:
Monitoring logic is AFTER the reference point taken active!

P12103 Area 1 relative related to P12101
Acknowledgement achieves P12188 byte 2 = 1

P12104 Area 2 relative related to P12101
Acknowledgement achieves P12188 byte 3 = 1

8.7 Axis parameters (continued)

Area monitoring axes

Area 2

P12105 Channel paragraph
Byte 1
0 or — P12106 and P12107 = q-parameter
1-8 P12106 and P12107 = p-parameter

Byte 2:
$xx: Number of the reference axis for which the area of monitoring is to apply

Byte 3:
$01: CNC monitored over the area
(Stops with error message: reached full scale position)
$10: Value in P12108 affects in the + and- -> marker field K1B1E11
$20: Value in P12109 affects in the + and- -> marker field K1B1E1

Byte 4:
0 Control position of P12107 Content: Pointer to parameter
$01 Control position is a value

P12106 Reference position
Contents: Pointer on parameters
0 or — Reference position = internal machine position (q2150)
-2150 Reference position = internal machine position (q2150)
-2152 Reference position = internal machine position (q2152)
With the reference position (internal) are these positions immediately for the order, over parameters only time-delayed.

P12107 Control position
Contents: Pointer on parameters
0 or — Monitoring logic not actively
Note:
Monitoring logic is AFTER the reference point taken active!

P12108 Area 1 relative related to P12106
Acknowledgement achieves P12189 byte 2 = 1

P12109 Area 2 relative related to P12106
Acknowledgement achieves P12189 byte 3 = 1

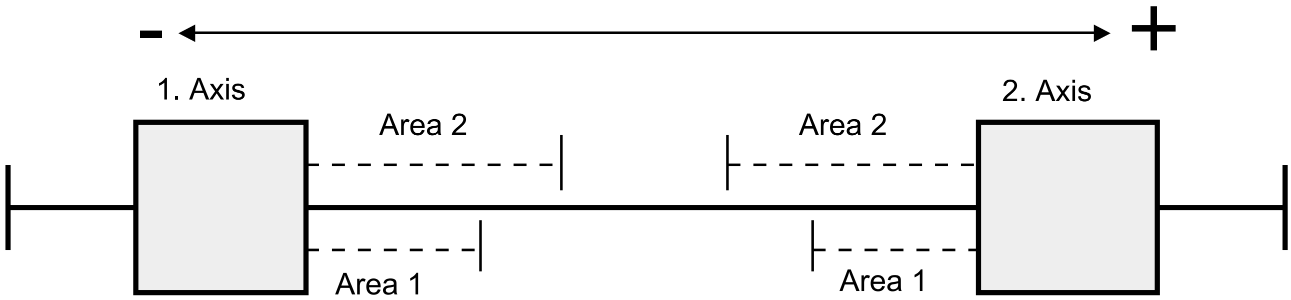
8.7 Axis parameters (continued)

Area monitoring axes

Area 2 relative related to P12106
Acknowledgement achieves P12189 byte 3 = 1

Example: Area monitoring 1. and 2. Axis

P12100	0	P12300	0
P12101	0	P12301	0
P12102	2350	P12302	2150
P12103	10	P12303	-10
P12104	20	P12304	-20



If axis 1 drives 2 from axis 2 into area, indicator K1BE1 is settinged
If axis drives 1 into area 1 from axis 2, indicator K1B1E1 is settinged
If machine pos. Axis 1 = machine pos. Axis 2 is settinged indicator K1B1

If axis drives 2 into area 2 from axis 1, indicator K1BE2 is settinged
If axis 2 drives 1 from axis 1 into area, indicator K1B1E2 is settinged
If machine pos. Axis 2 = machine pos. Axis 1 is settinged indicator K1B2

8.7 Axis parameters (continued)

Axis commands / functions

P12110 Axis position setting [mm, degree]
Display or debit position is settinged to input value.
If parameter is reset, then old value is displayed.

P12111 Relative relative start [mm, degree]
Axis starts when describing.

Feed in	P12121
Override in	P8866 or P12126
Acknowledgement achieves	P12191=1
If P12111 reset	P12191=0

An active relative positioning is aborted .

P12112 Absolute absolute start [mm, degree]
Axis starts when describing.

Feed in	P12122,
Override in	P8867 or P12127
Acknowledgement achieves	P12192=1
If P12112 reset	P12192=0

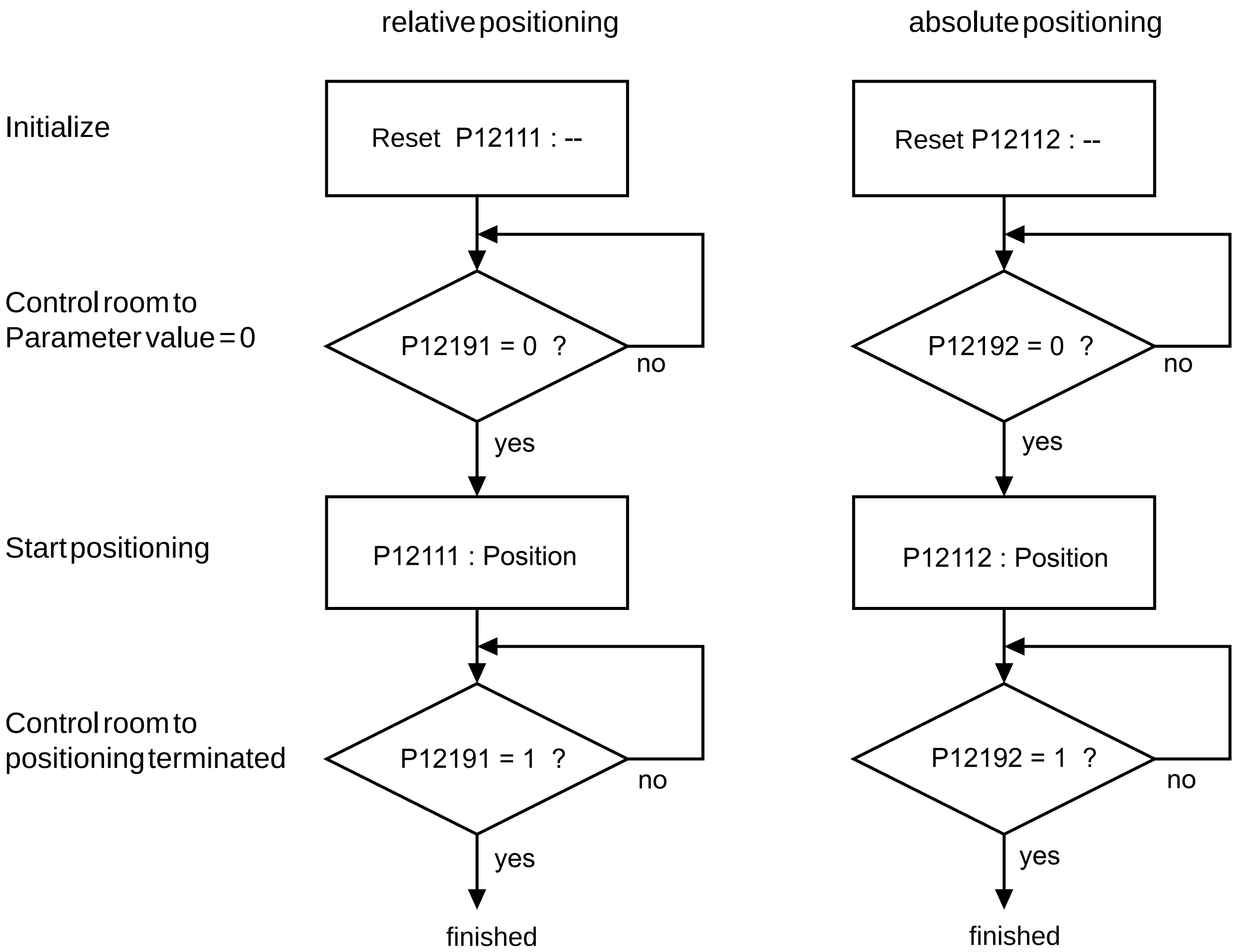
An active absolute positioning is aborted.

P12113 Absolute before position [mm, degree]
0 or — Before position check not actively
This before position can, with programmed not-interpolated axes (P8893)
in the NC program and active exact stop (G8, G28) for
premature step up of the active NC block to be used.
With achieving these before position
this is settinged in the parameter on zero,
i.e. it is active only for a monitoring.
Example:
N100 G28 F100 X:100
N110 P12113:20 X200
N120 Y100
i.e. with drive into block N110 and path difference to
Final position X200 < = P12113 is affected block changes after N120.

8.7 Axis parameters (continued)

Axis commands / functions

Example of a relative / an absolute absolute



8.7 Axis parameters (continued)

Axis commands / functions

P12114 Output skip
Axis is moving with the input (speed) without dynamic (skip function) in the selected direction. Axis position is updated.

P12115 Output via output cards (AK, AEK)
0 or — Not actively
-1 i.e. the four signals become into those PLC Interface (see P12195) written.

Byte 1
$xx 1-32 AK clip No. for this axis
Axis drives over outputs of a AK, AEK
per axis of 4 binary signals:
example:
Kl. 1 - drive positively
Kl. 2 - drive negatively
Kl. 3 - drive with max axis speed
Kl. 4 - drive with Slope speed
(settinged if indicator SLOP11 = 1)

Byte 2
$xx: 1-32 slot No. of the AK -, AEK card

8.7 Axis parameters (continued)

Axis commands / functions

P12119	Handwheel speed 0 or — max. Axis speed		[mm/min, degree/min]
P12120	Interpolation speed 0 or — max. Axis speed		[mm/min, degree/min]
P12121	Relative velocity 0 or — max. Axis speed		[mm/min, degree/min]
P12122	Absolute speed 0 or — max. Axis speed		[mm/min, degree/min]
P12123	Fixed position speed 0 or — max. Axis speed		[mm/min, degree/min]
P12124	Spindle speed 0 or — max. Axis speed		[degree/min]

8.7 Axis parameters (continued)

Axis commands / functions

P12125 Override with interpolation speed drive
0 or — Default see P8865
Otherwise Override paragraph 1-8

P12126 Override with relative speed drive
0 or — Default see P8866
Otherwise Override paragraph 1-8

P12127 Override with Absolu t speed drive
0 or — Default see P8867
otherwise Override paragraph 1-8

P12128 Override with fixedposition - speed drive
Default = 1 contents Override paragraph 1-8

P12129 Override at spindle speed drive
Default = 2 contents Override paragraph 1-8

8.7 Axis parameters (continued)

Parameter block of PLC (Marker interface)

P12130 Approach reference position

Byte 1

$00	Stop of the positive reference drive	(PLC: REPO+1)
$01	Start of the reference drive in positive direction. feed: Hand feed P8755 back message achieves P12180	

Byte 2

$00	Stop of the negative reference drive	(PLC: REPO-1)
$01	Start of the reference drive in negative direction. feed: Hand feed P8755 back message achieves P12180	

Byte 3

$01	Reference position release back message achieves P12180 (see also P12044) note: If with one already referenzierten axis the reference position release is again settinged, the reference point reset. the axis can again be referenziert.	(PLC: REPOF1)

Byte 4

$01	Reference position setting back message achieves P12180	(PLC: REPOS1)

8.7 Axis parameters (continued)

Parameter block of PLC (Marker interface)

P12131 Approach measuring position

Byte 1
$00 Stop of the positive measuring travel (PLC: MEPO+1)
$01 Start of the measuring drive in positive direction.
feed: Hand feed P8755
back message started P12181

Byte 2
$00 Stop of the negative measuring drive (PLC: MEPO-1)
$01 Start of the measuring drive in negative direction.
feed: Hand feed P8755
back message started P12181

Byte 3
$01 Measuring position release (PLC: MEPOF1)
back message settinged P12181

Byte 4
$01 Measuring position setting (PLC: MEPOS1)
back message settinged P12181

8.7 Axis parameters (continued)

Direction PLC —> interpolator

P12132 Approach emergency position, basic position

Byte 1 0 Stop for emergency position absolutely achieve (PLC: NOPOA1)
1 Start for emergency position absolutely achieve
Position in P12032
Feed: max. axes speed
Override 100%
All other axes in the channel go on axis stop.
Back message achieves in P12182
When starting to that emergency position drive becomes those message M2000 settinged.

Byte 2 0 Stop for emergency position relatively achieve (PLC: NOPOR1)
1 Start for emergency position relatively achieve
Position in P12033
Feed: max. axes speed
Override 100%
All other axes in the channel go on axis stop.
Back message achieves in P12182
When starting to that emergency position drive becomes those message M2000 settinged.

Byte 3 0 Stop for reason of position absolutely achieve (PLC: GRPOA1)
1 Start for reason of position absolutely achieve
Position in P12034
Feed: max. axes speed
Back message achieves in P12182
Override P12128

Byte 4 0 Stop for reason of position relatively achieve (PLC: GRPOR1)
1 Start for reason of position relatively achieve
Position in P12035
Feed: max. axes speed
Back message achieves in P12182
Override P12128

8.7 Axis parameters (continued)

Direction PLC —> interpolator

P12133 Approach fixed position, pendling position

Byte 1	0	Stop for approaching fixed position	(PLC: FEPO1)
	1	Start for approaching fixed position Position in P12036 Feed in P12123 Feedback reached in P12183 Override P12128	
Byte 2	0	Stop for approaching pendling position	(PLC: PEPO1)
	1	Start for approaching pendling position Position in P12037 Feed in P12123 Feedback reached in P12183 Override P12128	

P12134 Drive spindle

Byte 1	0	Stop for driving spindle positive	(PLC: SPE+1)
	1	Start for driving spindle positive Speed in P12124 Override P12129 Feedback reached in P12184	
Byte 2	0	Stop for driving spindle negative	(PLC: SPE-1)
	1	Start for driving spindle negative Speed in P12124 Override P12129 Feedback reached in P12184	

8.7 Axis parameters (continued)

Direction PLC —> interpolator

P12135 End position reached

Byte 1	1	Positive hardware end position reached	(PLC: EL+1)
Byte 2	1	Negative hardware end position reached	(PLC: EL-1)

P12136 Slope

Byte 1	1	Activate slope 1 actual axis speed <= slope speed 1 slope speed in P12005	(PLC: SLOP11)
Byte 2	1	Activate slope 2 actual axis speed <= slope speed 2 slope speed in P12006	(PLC: SLOP21)

P12137 Drive locking / drive stop

Byte 1	1	Drive locking for positive direction	(PLC: FASP+1)
Byte 2	1	Drive locking for negative direction	(PLC: FASP-1)
Byte 3	0	Axis stops with dynamics function (axis stop)	(PLC: FASOD1)
	1	Axis stops without dynamics function (axis locking) Dynamics in P12008, P12009	

8.7 Axis parameters (continued)

Direction PLC —> interpolator

P12138 Axis simulation

Byte 1 1 Simulation without output voltage (PLC: SIMMA1)
i.e. the axis moves controlled (P12139) and
the reference signal is simulated internally.

P12139 Axis controlled

Byte 1 1 Controlled with measuring system (PLC: GSMM1)
i.e. the real axis position is carried internally.
If controlled is cleared (positioning control on), this axis position is taken as actual position

Byte 2 1 Controlled without measuring system (PLC: GSOM1)

P12140 Axis updated (target = actual)

Byte 1 1 Updating with reapproaching the old position (PLC:NGMW1)

Byte 2 1 Updating without reapproaching (PLC: NGOW1)

Byte 3 1 Tracking Feedback with motion command (PLC: NGFB1)

P12141 Axis clamped

Byte 1 1 Clamping is switched on (PLC: GEKL1)
i.e. no drive locking for axis;
drift correction is switched off
Axes output = 0 volts

Byte 2 1 How byte 1,1 (PLC: GKAE1)
additionally been frozen display

8.7 Axis parameters (continued)

Direction PLC —> interpolator

P12142 Admissible output monitoring

0 or —	Monitoring switched off Around this monitoring to switch on must: P12026 > 0, and P12142 = 1 its. With achieving this boundary is affected error message (M2002) and analog output = 0 is settinged.

P12143 Relative positions separately announce

Byte 1 1	Relatively movements (P12111) are summed up in P12158. Display position is preserved. P12158 is reset when switching on on.
Byte 2 1	Handwheel movements are summed up in P12158

P12144 Sercos drive control word (PLC -> Interpolator -> Sercos Drive)

Byte 1

$00	drive off	(PLC:ATEIN1)
$01	drive on	

Byte 2

$00	no release	(PLC:ATFRG1)
$01	drive release	

Byte3

$00	drive stop	(PLC:ATSTR1)
$01	drive start	

Byte4

$00	selection head operating mode (byte specifies 1) in P12078, e.g.: Position of control with generator 1 drag errors freely	(PLC:ATBTR1)
$01	selection apart from operating mode 1 (byte specifies 2) in P12078, e.g.: Speed control	

8.7 Axis parameters (continued)

Axis feedbacks / informations

P12150	Machine position actual mechanical position of the axis	[mm, degree]
P12151	Target position position of the axis referred to the machine reference point G53	[mm, degree]
P12152	Actual position actual measuring position of the axes	[mm, degree]
P12153	Measuring position (display) stored display position at measuring signal (P8855) (Channel mapping) *	[mm, degree]
P12154	Display - position Display = target position - display shift (Channel mapping) *	[mm, degree]
P12155	Programmed position position programmed in the NC block (Channel mapping) *	[mm, degree]
P12156	Delta position Delta = programmed position - display position (Channel mapping) *	[mm, degree]
P12157	Display - shift Sum of all shifts (e.g. zero point, G92 shift, tool length) (Channel mapping) *	[mm, degree]

* Channel mapping parameters are channel specific,
i.e.. it does not exist a channel-spreading q-parameter.
During the switching on routine becomes however the parameter status q-parameter into the parameter status of the p-parameter copies; i.e. is in a parameter with channel mapping the parameter status to be redefined, then this must in suitable status of the q-parameter are affected.
Example: P12153 is to be provided with 4 post-decimal positions
—> setting parameter status of q2153 with $24xxxxxx
After restarting parameter status in P12153 corresponds the parameter status q2153

8.7 Axis parameters (continued)

Axis feedbacks / information

P12158	Position shift (e.g. handwheel-shift in Automatic)	[mm, degree]
P12159	Distance reference switch -> reference mark	[mm, degree]
P12160	Actual lag distance	[mm, degree]
P12161	Actual coupling correction	[mm, degree]
P12162	Actual drift correction	[mm, degree]
P12163	Actual backlash correction	[mm, degree]
P12164	Actual measuring system correction	[mm, degree]
P12165	Actual pitch correction	[mm, degree]

8.7 Axis parameters (continued)

Axis feedbacks / information

P12167	Command speed (regulator) Access with oscilloscope functions	[mm, degree]
P12168	Actual speed (measuring system) Access with oscilloscope functions	[mm, degree]
P12169	Actual output voltage of the position adjustment	[V]
P12170	Actual axis speed	[mm/min, degree/min]
P12171	Actual relative speed	[mm/min, degree/min]
P12172	Actual absolute speed	[mm/min, degree/min]
P12173	Actual fixed position speed	[mm/min, degree/min]
P12174	Actual spindle speed	[degree/min]

8.7 Axis parameters (continued)

Axis feedbacks / information

P12176 ADW module Address
Byte 1
$xx ADW input Paragraph 1... 4
(selection of the A/D transducer on the ADW Module)
Byte 2
$xx slot paragraph 1... 32

P12177 ADW Input value
the A/D of transducer addressed in P12176
Contents: [32767... -32768]

P12178 DAW module address
Byte 1
$xx DAW output Paragraph 1... 4
(selection of the D/A transducer on the DAW module)
Byte 2
$xx slot paragraph 1... 32

P12179 DAW output value [V]
the D/A of transducer addressed in P12178
Note:
If P12178 does not specify, the DAW value becomes up
the axis module of this axis output.

8.7 Axis parameters (continued)

Parameter block to the PLC (marker interface)

P12180	Reference position		
	Byte 1 1	Reference point taken	(PLC: REF1)
	Byte 2 1	Switch, active edge going	(PLC: REFS1)
P12181	Measuring position		
	Byte 1 1	measuring position taken	(PLC: MEGT1)
	Byte 2 1	switch, active edge going	(PLC: MES1)
P12182	Emergency position, basic position reached		
	Byte 1 1	emergency position reached	(PLC: NOPOE1)
	Byte 3 1	Basic position reached	(PLC: GRPOE1)
P12183	Fixed position, pendling position reached		
	Byte 1 1	fixed position reached	(PLC: FEPOE1)
	Byte 2 1	pendling position reached	(PLC: PEPOE1)
P12184	Speed reached		
	Byte 1 1	speed reached	(PLC: DREZE1)
P12185	Software end position		
	Byte 1 1	positive software end position	(PLC: SWEL+1)
	Byte 2 1	negative software end position	(PLC: SWEL-1)

8.7 Axis parameters (continued)

Parameter block to the PLC

P12186 Axis drives

Byte 1 (PLC: ACHFA1)

0 No axis specification
i.e.: Axis does not drive or axis builds the even lag distance off

1 Axis does not drive —> axis specification / axis feed to zero
i.e.: Axis specifications are available.

P12187 Driving instruction

0 No driving instruction pends
i.e.. - no verfahrweg for axis
- axis finds within exact stop finely (P12016)

Byte 1 (PLC: FB+1)

1 Driving instruction for positive driving direction
(driving instruction remains pending, if e.g. in the drive path the feed to zero becomes)

Byte 2 (PLC: FB-1)

1 Driving instruction for negative driving direction
(driving instruction remains pending, if e.g. in the drive path the feed to zero becomes)

Byte 3

1 Axis drive actual actively
i.e. the even position actual which can be started yet does not achieve.

P12188 Back message area monitoring axes area 1

Byte 1 1 Reference position (P12101) = control position (P12102) (PLC: K1E1)
Byte 2 1 Control position area 1 (P12103) achieves (PLC: K1B1E1)
Byte 3 1 Control position area 2 (P12104) achieves (PLC: K1BÈ1)
Note:
Signals are set AFTER the reference point taken!

P12189 Back message area monitoring axes area 2

Byte 1 1 Reference position (P12106) = control position (P12107) (PLC: KÈ1)
Byte 2 1 Control position area 1 (P12108) achieves (PLC: K2B1E1)
Byte 3 1 Control position area 2 (P12109) achieves (PLC: K2BÈ1)
Note:
Signals are set AFTER the reference point taken!

8.7 Axis parameters (continued)

Parameter block to the PLC

P12191	Relative position achieves			
	Byte 1 1	Relative position achieves i.e.: Axis actual around the error limit from the target position removes.		(PLC: REPOE1)
P12192	Absolute position achieves			
	Byte 1 1	Absolute position achieves i.e.: Axis actual around the error limit from the target position removes.		(PLC: ABPOE1)
P12194	Sercos drive status (Sercos drive -> Interpolator -> PLC)			
	Byte 2 1			
		$0000	Drive not yet ready for the performance power, there drive check not finally.	
		$0001	Drive ready for the performance power,	
		$0100	Drive actual ready for use and performance supply switched on, note: due to this back message are those to operate drive releases (P12144)	
		$0101	Drive actual ready for use, drive releases (P12144) are effective, drive output stage actual actively.	
P12195	AK / AEK - outputs Definition in P12115			
	Byte 1			
		$01	drive in positive direction	(PLC:POSFA1)
	Byte 2			
		$01	drive in negative direction	(PLC:NEGFA1)
	Byte 3			
		$01	drive with max. Axis speed	(PLC:SCHFA1)
	Byte 4			
		$01	drive with Slope speed	(PLC:LGSFA1)

8.7 Axis parameters (continued)

Parameter block to the PLC

P12197 Message number (PLC: MELNR1)
Message display for this axis
e.g. M2110 approach reference position

P12198 Channel number (PLC: KANNR1)
Axis belongs to this channel (1...8)

P12199 Logicals axis number (PLC: LANA1)
Axis has this logical number (1...32) in the channel

9. Messages

The following abbreviations are used for the effect of the messages:

SA system abort (all channels)
KA channel abort
KS channel stop
A display

9.1 Groups of messages

M1000 . .. M1999	Messages of the CNC, block interpreter, I/O
M1150... M1199	Error off tool radius correction
M1950... M1999	SYSTEM error with the call by DLL Functions
M2000 . .. M2999	Messages of the interpolator
M3000 . .. M3999	Messages of the PLC
M3000 . .. M3499	Global messages of the plant (i.e. channel overlapping)
M3000 . .. M3249	Messages are indicated with gray writing on red ground
M3250 . .. M3499	Messages are indicated with black writing on turquois ground
M3500 . .. M3999	Channel specific messages
M3500 . .. M3749	Messages are indicated with gray writing on red ground
M3750 . .. M3999	Messages are indicated with black writing on turquois ground
M4000 ... M4999	Messages of the operating panel
M4000 ... M4499	BWO messages
M4000 ... M4299	DOS messages
M4300 ... M4399	Messages of the system
M4400 ... M4499	Messages of the MMI
M4500 ... M4999	free for the user
M5000 . .. M7999	Reserved
M8000 . .. M8999	Messages of cycles
M8000 . .. M8499	Standard cycles
M8500 . .. M8999	Free for users
M9000... M9999	Error message off CNC DLL
M9000... M9099	Reserves for BWO
M9100... M9999	Freely for users

9.2 General CNC messages

Message	Effect	Meaning
1000	KS	Error when writing in Flash-memory P8505 Error code Info 1 P8506 Channel number Info2
1001	KS / KA	FLASH Function not executed
1002	KS	No NC program in the Flash-memory
1008	KA	Approach reference points
1009	A	Switching AUTO <—> POS not allowed Switching has to be done in MANUAL!
1050	KA	Tool group not found in the tool data memory
1053	KA	Service life of the group of tools run
1098	A	Error message when renumbering a NC program -> skip target is not available. The renumbering is not executed. Number of the NC block which has caused this message is written in P8505.
1099	A	Error message when renumbering a NC program -> indirect skip in NC memory available. Renumbering is not stopped, but this NC block is marked with the text ‘warning’.

9.3 Messages at input/output

Message	Effect	Meaning
1100	A	Error in the test character or in the character number with the datentransfer between operating panel and CNC, i.e. the operating panel obtain on divergent Test character / file size as the CNC.
1102	A	No abort confirmation from the operating panel Each I/O-abort-confirmation-telegram (CNC —> OP) has to be confirmed by the operating panel. If this is not the case, the message 1102 is.
1105	A	Error when opening a program in the NC memory
1106	A	NC program becomes overwritten
1107	A	NC program is already existing and will not be overwritten
1110	A	Wrong file identification i.e.: the identification at the file start is not correct ‘P’ for NC programs ‘D’ for parameters ‘B’ for zero points Line number in P8505
1111	A	Line read can not be interpreted Line number in P8505 Reasons e.g.: - NC block does not start with block number - parameter line is damaged - preceeding file has no end identification (#) - I/O parameter is not correct (Baudrate, parity ...)
1112	A	Invalid channel Reason: Read or write operations with channel parameters of which the channel does not exist.

9.3 Messages at input/output (continued)

Message	Effect	Meaning
1113	A	Reception string overflow with I/O traffic (input) (max. character length of a line: 400) Read in string actual longer than reception s string. A cause: - It tried on false file format to read in (EXE, being file...) - Errors in the input file
1114	A	Reception string does not contain a test character
1115	A	Reception string contains false test character
1120	KS	Memory space problem with the system down load (too little temporary memory (HEAP) available)
1121	KS	Download software does not fit the hardware version of the CNC Card, e.g.. to charge it tried on BS902.xxx into a CNC Hardware 084564.
1122	KS	Check total the loaded system software actual not correctly. - > file possibly damages - > problems while the loading.
1123	KS	Check total the system software actual burned in the FLASH not correctly. - > file possibly damages - > hardware problem

9.4 Messages with the tools radius correction

Message	Effect	Meaning
1151	KA	Tool radius too largely
1152	KA	Circle radius too small (message with stop)
1153	KA	Tool radius correction error with ON/OFF drive
1154	KA	G0 / G1 with tool radius correction not possibly
1155	KA	G2 / G3 with tool radius correction not possibly
1156	KA	With tool radius correction no intersection of the sets
1157	KA	Too many blocks without path
1158	KA	G12 not possible
1159	KA	No path difference

9.5 Messages of the NC interpreter

Message	Effect	Meaning
1200		Coprocessor is missing
1201	KA	Channel descriptor not valid
1202	KA	Parameter number too large
1203	KA	Division through zero
1207	KA	Too many axes programs. Message only in the export version A cause: It is tried, in a block more axes to interpolate as certified. Error correction: Program fewer axes in the incorrect block.
1208	KA	Too many parameters programmed in the block (max. 32)
1209	KA	Too many bracket levels (maximum 10)
1210	KA	Bracket is missing
1211	KA	Syntax error - in the automatic mode incorrect program/record number becomes in P8910 / P8911 displayed. - with the Renumbern record number in the P8505 (Info1) is displayed.
1212	KA	Negative root
1213	KA	Function can not be executed e.g. at - logarithm-calculation - tan (90) ...
1214	KA	No parameter operation
1215	KA	Syntax error
1216	KA	Unknown axis name
1217	KA	Error at M25 The NC program contains on loop end (M25) without loop start (M24).
1218	KA	Error at M24

9.5 Messages of the NC interpreter (continued)

Message	Effect	Meaning
1219	KA	Errors at sub-routine call up
1220	KA	Too many sub-routine call up
1221	KA	Parameters not loaded
1222	KA	Too many M - or G - or P - functions in a block programs
1223	KA	Error in the outline path In the case of NC program abort: A possible cause tool radius actual too largely, in order to process outline. Error correction Tool with smaller tool tools use. With NC program stop: Programmed outline radius actual smaller than the half chord. Radius is corrected. NC program can be started again.
1224	KA	Spindle not available
1225	KA	No tool group defined
1226	KA	Geometry error possible reason: - at G12, G123 next block with travel is missing - function G123 is programmed with M01
1227	KA	M - function number too large
1228	KA	M - function definition not correct - a M-function with skip target is in the NC program. M-Function is not defined as skip-M-function. - a M-function without skip target is in the NC program. M-Function is defined as skip-M-function.
1229	KA	M1 is not allowed in this block e.g. it may not be synchronized if tool radius correction is switched on.
1230	KA	No feed was programmed for G01/02/03
1231	KA	G02/G03 without path difference

9.5 Messages of the NC interpreter (continued)

Message	Effect	Meaning
1232	KA	Not allowed combination of G or M function programmed in the NC block.
1233	KA	Circular interpolation (G02/G03) programmed without R or I / J / K
1234	KA	Programmed outline radius actual smaller than the half chord, i.e. with this programmed radius can the programmed terminator point not to be achieved. NC program is stopped. Radius is corrected. NC program can be started again.
1235	KA	Spline not possible
1237	KA	Circle dynamics, Error when calculating feed Reason: Acceptable path deviation (P8852) is larger than the programmed radius. Erasing the error: Reduce path deviation (P8852)
1238	KA	Programmed centre point is not correct with programmed start and terminator point (see also P8843)
1240	KA	No valid operation range defined (P8555) Message appears e.g. at zero point call up
1248		Command / instruction not certified usually on instruction off a DLL Cycle, does not permit in such a way.
1249	KA	Invalid NC interpreter default Error correction: The following parameters check: Parameter P8830 Preset zero point Parameter P8853 Preset type of coordinate
1252	KA / A	Program not found
1255	KA / A	Block not found
1256	A	NC memory full

9.5 Messages of the NC interpreter (continued)

Message	Effect	Meaning
1257	KA	Memory error in drip-feed-buffer Over - or underflow
1258	KA	Memory error in drip-feed-buffer Faulty block synchronization when reading the buffer
1259	KA / A	NC block with defective check sum
1260	KA / A	Error when reading NC memory
1261	A	Error when writing in NC memory
1262	KA / A	File not defined
1263	KA / A	Program not found
1264	KA / A	Program not opened for reading
1265	A	Program write-locked Interlocking through key-operated switch etc..
1266	KA / A	NC program is locked Program is already opened for writing by an other user.
1267	KA / A	Too many programs opened
1268	KA / A	Circle buffer - overflow e.g. Drip-feed-buffer overflow
1269	KA / A	Applied function is not supported at ring - buffer (Drip-feed-operation)
1270	KA / A	Format error (The opened file is not a NC program)
1271	KA / A	Faulty NC block e.g. ‘N’ is missing before the block number
1272	KA / A	Error at access on the NC memory
1275	A	NC saving write protected
1280	KA / A	Directory chain of NC memory defective

9.5 Messages of the NC interpreter (continued)

Message	Effect	Meaning
1281	KA / A	File chain of NC program defective
1283	KA / A	System data for NC memory organization defective
1285	KA / A	Blocks in the NC memory are double-chained

1290 NC blocks are not sorted
(beginning with the small number and ending with the big one)

P8505	lines-number of the I/O files	Info 1
P8506	new block number	Info 2

9.5 Messages of the NC interpreter (continued)

Message	Effect	Meaning
1300	KA / A	Parameter not installed - cleared Parameter number in P8505 Info 1 Channel in P8506 Info 2
1301	KA / A	Parameter-check sum defective Difference to M1811: parameter is actually read
1302	KA / A	Parameter number larger than maximum parameter number
1304	KA / A	Parameter is write-locked Parameter number in P8505 Info 1 Channel in P8506 Info 2
1305	KA / A	Channel parameter not available. It was tried, to write a channel parameter, whose channel does not exist.
1306	A	AUTO file frame erre OVERFLOWS A cause: With einges chalteter increasing AUTOMATIC file locks is enough storage space not off (see also P8806).
1310	KA	Program number not loaded at emergency program call (P8820 etc.)
1320	KA	Mode of operation change locked e.g. during emergency program call it can not be switched to AUTO mode of operation
1321	KA	Control not in MANUAL operation mode
1322	KA	Control not in AUTO operation mode

9.6 Messages of the system

Message	Effect	Meaning
1800		Fatal error; System start again After sequence error - M1808 - M1820 - M1830 - M1840
1808	SA	Interface operating console < —> CNC overflow of the receive buffer CNC receive buffer is flooded. Possible causes: Override Inputs connected at the operating console not
1809	KA	Interface PLC —> CNC: Parameter write routine (SPARC) with Invalid channel number information: parameter number in P8505 channel in P8506
1810	KA	Interface CNC —> PLC: Data overflow During a parameter transfer to the PLC there was a data overflow. (floating decimal point format does not fit into the fixed point format, mantissa too largely) Information: parameter number in P8505 (info. 1), channel in P8506 (info. 2)
1811	KS	Parameter test character defectively If the channel 1 in the operating mode HAND is, becomes from the operating system in background constantly the total parameter area checks. Info.: Q-parameter number in P8505 (info. 1)
1815	KA	Real time buffer overflow - faulty real time buffer administration or - faulty feedback of the interpolator
1816	KS	Feed back buffer overflow - Feedbacks of the interpolator can not be processed any more.

9.6 Messages of the system (continued)

Message	Effect	Meaning		
1820	SA	Floatingpoint exception		
		Exception reason in P8505	Byte 1:	Bit 0 Inexact
				Bit 1 Underflow
				Bit 2 Overflow
				Bit 3 Divide by Zero
				Bit 4 Invalid operation
			Byte 2	Bit 8 not alligned memory access
		System address in	P8506	
		Task-Id in	P8507	
			$100	PLC Task
			$101	PLC CMD Task
			$102	PLC Background Task
			$103	PLC Server Task
			$200	Proma Task
			$201	Proma CMD
			$300	Network Task
			$301	Network Master
			$400	CMD STR Task
			$500	SMMS Task
			$600	Oskar Task
			$700	Position adjustment
			$800	AUTOMATIK Task
			$900	Real time Task
			$a00	Monitor Task
			$b00	Proma Transport Task
			$c00	CPU2 TX-Task
			$c01	CPU2 RX-Task
			$d00	Follower Task
1825		Network errors		
1826		Network errors		

9.6 Messages of the system (continued)

Message	Effect	Meaning
1830	SA	NA signal became intermittently active (NA: Power failure signal) After the message M1830 position adjustment may not be eingeschatet any longer! (interlock in the PLC) No more NC programs may be started! Axis positions cannot be correct possibly. To start after the occurance of of this error message actual the system again. (power OFF / on) A cause: Cure-early mains voltage failure Errors eliminate: Mains voltage check Power supply unit check
1840	SA	(F0) General hardware error. Module or network failed. (see also at the operating panel under DIAGNOSIS system configuration)
1841	SA	(F1) Battery voltage is missing Causes: - Battery defectively - contact problems of the battery plug
1842	SA	(F2) ±15V-Spannung is missing
1843	SA	(F3) Battery and ±15V-Spannung are missing
1844	SA	(F4) Processor fan defectively Caution: With defective fan overheating danger consists, there that processor on the CNC Module is no longer sufficiently cooled. an overheating of the CNC Processor leads to malfunctions or to complete failure of the system.

9.6 Messages of the system (continued)

Message	Effect	Meaning
1950	A	DLL Function or subfunction missing Off the CNC DLL a function in the CNC CORE is called, which does not exist. Information: Info1 / P8505: Function code Info2 / P8506: Subfunction paragraph
1951	A	Invalid channel with function call Off the CNC DLL a function in the CNC CORE is called, also invalidly / illegal channel number Information: Info1 / P8505: Function code Info2 / P8506: Subfunction paragraph
1952	A	Invalid axis paragraph with function call Off the CNC DLL a function in the CNC CORE is called, also invalidly / illegal axis paragraph Information: Info1 / P8505: Function code Info2 / P8506: Subfunction paragraph

9.7 Mssages of the interpolator

Message	Effect	Meaning
2001	SA	Lag distance too large
2002	SA	Admissible output voltage exceeded (see P12142..)
2004	SA	Measuring system signal is missing
2005	SA	Measuring system UAS is missing
2006	SA	Measuring system frequency too large
2007	SA	Measuring system correction too large
2010	SA	Drive errors

2020 Axis can not be moved

2022 Axis not in position

2023 Coupling difference too large

2024 Reference distance too large

2050 Block buffer overflow in the interpolator

2052 Coordinate mode not defined

Possible causes:
- Type of coordinate does not specify, i.e. e.g. with the Robot system is missing to Robot definition in P11800
- When switching on of a type of coordinate (G48 on, G49...) are yet all axes involved does not report.

Errors eliminate:
- Type of coordinate specify
- Axes report

9.7 Mssages of the interpolator (continued)

Message	Effect	Meaning
2101		Hardware end position +
2102		Hardware end position -
2103		Software end position +
2104		Software end position -
2110		Approach reference position
2112		Approach measuring position
2114		Approach emergency position
2116		Approach basic position

9.8 Messages of the PLC

Message	Effect	Meaning
3000		Connection operating panel <-> CNC was interrupted - Security stop of all axes is set
3501		Mode of operation change -> Automatic locked
3502		Mode of operation change -> Manual locked

9.9 Messages of the operating panel

Message	Effect	Meaning
		Range 4000 .. 4299 are DOS error messages
4002	A	File not found
4003	A	Path not found
4004	A	Maximum number of files already opened (FILES=xx)
4005	A	File access refused
4006	A	Invalid file handle not defined or
4012	A	Invalid file mode
4015	A	Drive number not allowed
4016	A	Actual directory can not be cleared
4100	A	Error when reading a file
4101	A	Error when storing a file
4102	A	File variable is not allocated to a file
4103	A	File is not opened
4104	A	File was not opened for reading operations
4105	A	File was not opened for storing operations
4106	A	Invalid numerical format

9.9 Messages of the operating panel (continued)

Message	Effect	Meaning
4150	A	Disc is not write-locked
4151	A	Peripheral device unknown / not connected
4152	A	Drive not ready
4153	A	Invalid DOS function
4154	A	Check sum error when reading from the floppy / hard disc
4155	A	Invalid disc parameter block
4156	A	Head-positioning error on the floppy / hard disk
4157	A	Sector format unknown
4158	A	Disc sector can not be localised
4160	A	Write error when accessing a peripheral device
4161	A	Read error when accessing a peripheral device
4200	A	Division with zero
4201	A	Range test: error
4202	A	Stack test: overflow
4203	A	No more space in the Heap-range
4204	A	Invalid pointer operation
4205	A	Floating point overflow
4206	A	Floating point underflow
4207	A	Floating point error
4211	A	Program too deep

9.9 Messages of the operating panel (continued)

Message	Effect	Meaning
		Area 4300... 4399 messages of the operating system operating panel
4300	A	More than 400 programs in the directory (CNC: NCDATA \...). it cannot do all to be displayed.
4301	A	Operating panel Input SIO parity error
4302	A	Operating panel Input SIO Overrun error
4303	A	Operating panel Input SIO Framing error
4310	A	More than 1600 programs in the directory (CNC: NCDATA \...). it cannot do all to be displayed. (only on PC operating panel with driver CNC900X.EXE starting from 09.06.97.)
4390	A	Connection operating panel <-> CNC actual interrupted. No response of the CNC within Timeout.
		Area 4400... 4499 messages of the control surface
4400	A	Control not in MANUAL
4401	A	Changeover to AUTO not made
4402	A	Graphics simulation not active

9.10 Cycle messages

Message	Meaning	
8001	Geometry errors in the cycle This message appears when the control finds out that the pocket-contour can not be processed with the programmed tool data. E.g. P14 smaller than tool radius or tool radius = 0, or tool radius is not programmed.	(G71, 72, 73, 74, 75)
8003	Corner radius too small	
8004	Corner radius too big	
8005	Invalid tool radius	
8006	Sequence error	
8007	Pocket radius < tool radius	
8008	Pre-bore tool radius > pocket radius	
8009	In-feed > pocket depth (P13) - allowance on pocket depth (P18)	
8010	Invalid inner radius	(P11)
8011	Invalid 1st. pocket dimension	(P11)
8012	Invalid 2nd. pocket dimension	(P12)
8013	Invalid pocket depth	(P13)
8014	Invalid radius	(P14)
8015	Invalid allowance	(P15)
8016	Invalid in-feed	(P16)
8017	Invalid in-feed	(P17)
8018	Invalid allowance on pocket depth	(P18)
8019	Invalid safety allowance	(P19)
8020	Invalid fine-infeed	(P20)

9.10 Cycle messages (continued)

Message	Meaning	
8021	Invalid plange in feed	(P21)
8032	Invalid feed	(P32)
8033	Invalid drilling depth	(P33)
8034	Invalid preliminary stop plane	(P34)
8035	Invalid retraction plane on pocket depth	(P35)
8036	Invalid stroking rate	(P36)
8037	Invalid parameter for 1st. stroke	(P37)
8038	Invalid safety allowance	(P38)

9.10 Cycle messages (continued)

Message	Meaning	
8100	Invalid X-coordinate X(AB)	(P100)
8101	Invalid Y-coordinate Y(AB)	(P101)
8103	Invalid angle (E1)	(P103)
8104	Invalid vector length (L1)	(P104)
8105	Invalid vector division (T1)	(P105)
8106	Invalid number of positionings (N1)	(P106)
8107	Vector division (T1) or number of positionings (N1) is missing	(P105) (P106)
8110	Invalid X-coordinate X(AC)	(P110)
8111	Invalid Y-coordinate Y(AC)	(P111)
8113	Invalid angle (E2)	(P113)
8114	Invalid vector length (L2)	(P114)
8115	Invalid vector division (T2)	(P115)
8116	Invalid number of bores (N2)	(P116)
8117	Vector division (T2) or number of positionings (N2) is missing	(P115) (P116)
8120	Invalid X-coordinate pitch circle centre point	(P120)
8121	Invalid Y-coordinate pitch circle centre point	(P121)
8122	Invalid pitch circle diameter (D)	(P122)
8123	Invalid starting angle (E)	(P123)
8124	Invalid travelling angle (L)	(P124)
8125	Invalid pitch angle (T)	(P125)
8126	Invalid number of bores (N)	(P126)

9.10 Cycle messages (continued)

Message	Meaning		
8300	No spindle speed programmed	(G81, G83, G84, G85)	
8301	No spindle direction of rotation	(G81, G83, G84, G85)	
8302	Spindle speed = 0	(G81, G83, G84, G85)	
8303	Spindle speed not reached	(G81, G83, G84, G85)	
8309	Oversized pitch dimension (T1) and number of positionings (N1)		(P105) (P106)
8310	Oversized vector division (T1) and final point B (Final point B is defined by X and Y coordinate (P101 and P102))		(P105)
8311	Oversized vector length (L1) and final point B (Final point B is defined by X and Y coordinate (P101 and P102))		(P104)
8313	Oversized vector length (L1) and vector part (T1)		(P104) (P105)
8315	Oversized vector division (T1) and number of positionings (N1)		(P105) (P106)
8316	Input number of positionings with 0		(P106)
8320	Oversized vector division (T2) and final point C (Final point C is defined by X and Y coordinate (P111 and P112))		(P115)
8321	Oversized vector length (L2) and final point C (Final point C is defined by X and Y coordinate (P111 and P112))		(P114)
8323	Oversized vector length (L2) and vector division (T2)		(P114) (P115)
8325	Oversized vector division (T2) and number of positionings (N2)		(P115) (P116)
8326	Input number of positionings with 0		(P116)

9.10 Cycle messages (continued)

Message	Meaning	
8332	Wrong proportioned travelling angle L (P124) and pitch angle T (P125)	
8333	Oversized pitch angle T and number of positionings N	(P125) (P126)
8334	Indicate pitch angle with 0	(P125)
8335	Indicate pitch angle with 1	(P125)
8336	Indicate number of bores N with 0	(P126)
8337	Indicate number of bores N with 1	(P126)

10. Program packages

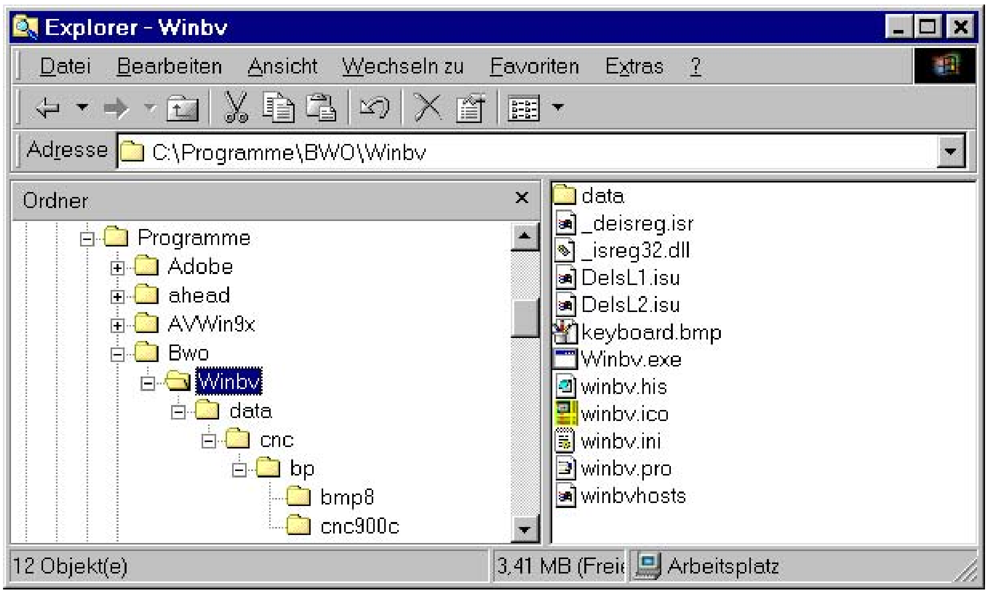
10.1 Operating system management program WINBV

for CNC CPU´s, which is programmed over the Ethernet interface.
- CNC CPU 085003 and 085008
- CNC CPU 085004 and 086004
- CNC CPU 800242 and 880242
- CNC CPU 800301 and 800801 ... and future CPU´s

WINBV is actual a tool for:
- System maintenance
- Data adminstration
- Data protection
- NC archiving
- On-line remote diagnostics

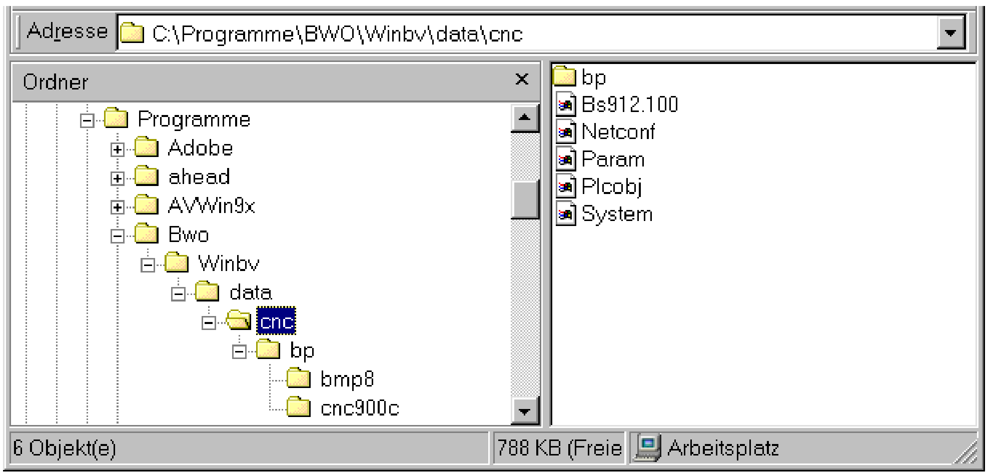
10.1.1 WINBV install

WINBV becomes over on by BWO supplied SETUP program installed. WINBV puts one linkage and the following directories on



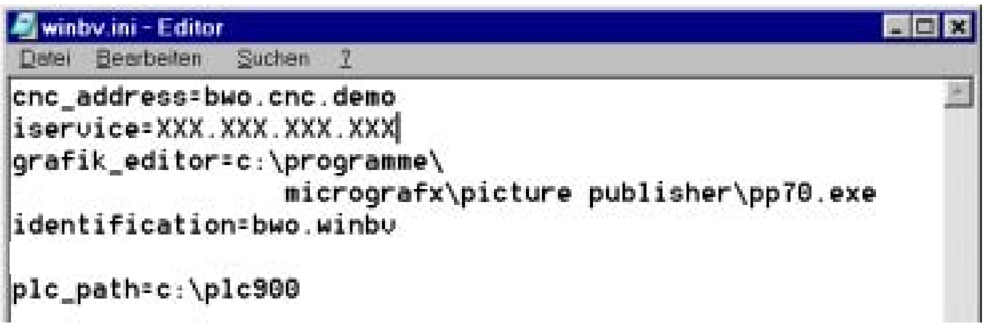
In the subdirectory CNC are already all directories and files for those standard CNC contained.

For further applications it is advisable, the directories and files after the same sample to create.



10.1.1 WINBV install (continuation)

WINBV.ini



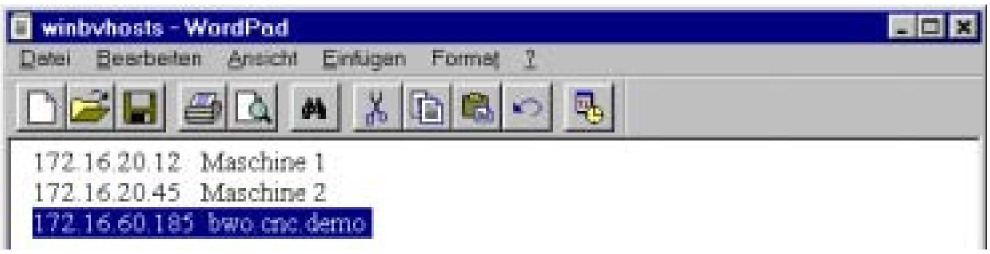

```
cnc_address=bwo.cnc.demo
iservice=XXX.XXX.XXX.XXX
grafik_editor=c:\programme\
                    micrografx\picture publisher\pp70.exe
identification=bwo.winbv

plc_path=c:\plc900
```

For the remote diagnostics into the winbv.ini must be the Iservice address.
As graphic editior **Micrografx\picture publisher** one recommends.

Language selection:	without specification	>	English
	deutsch	>	German

WINBVhosts



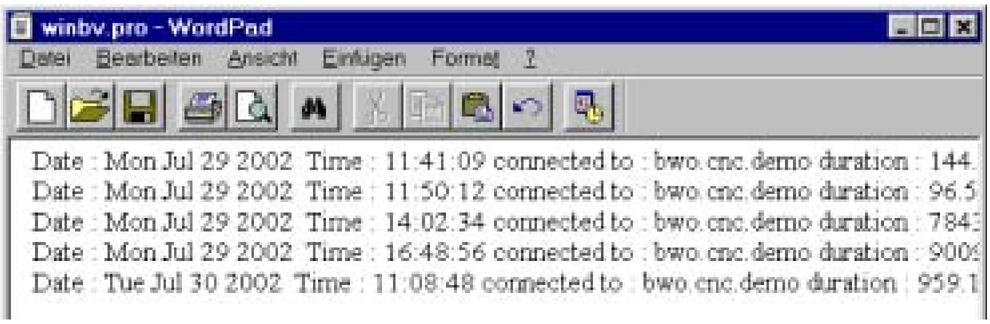

```
172.16.20.12  Maschine 1
172.16.20.45  Maschine 2
172.16.60.185  bwo.cnc.demo
```

Definition of symbolic addresses

WINBV.his


```
Maschine 1
Maschine 2
bwo.cnc.demo
```

WINBV.pro

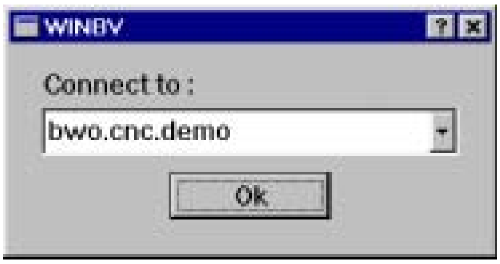



```
Date : Mon Jul 29 2002  Time : 11:41:09 connected to : bwo.cnc.demo duration : 144.
Date : Mon Jul 29 2002  Time : 11:50:12 connected to : bwo.cnc.demo duration : 96.5
Date : Mon Jul 29 2002  Time : 14:02:34 connected to : bwo.cnc.demo duration : 784
Date : Mon Jul 29 2002  Time : 16:48:56 connected to : bwo.cnc.demo duration : 900
Date : Tue Jul 30 2002  Time : 11:08:48 connected to : bwo.cnc.demo duration : 959.1
```

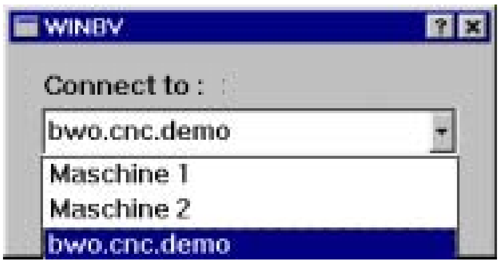
Log of the connections

10.1.2 WINBV start

Connection makes to
controller CNC CPU
with IP No. xxx.xxx.xxx.xxx
or with symbolic address
(determining with WINBVhosts)



Selection of CNC

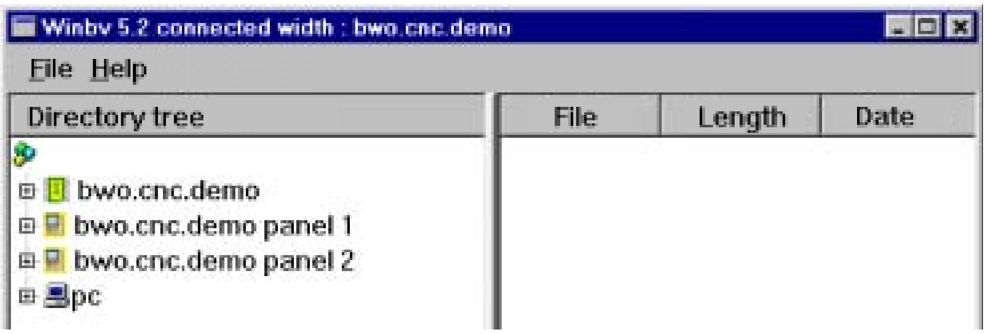


Connection makes to
operating panel
(only with started operating panel)

10.1.2 WINBV start (continuation)

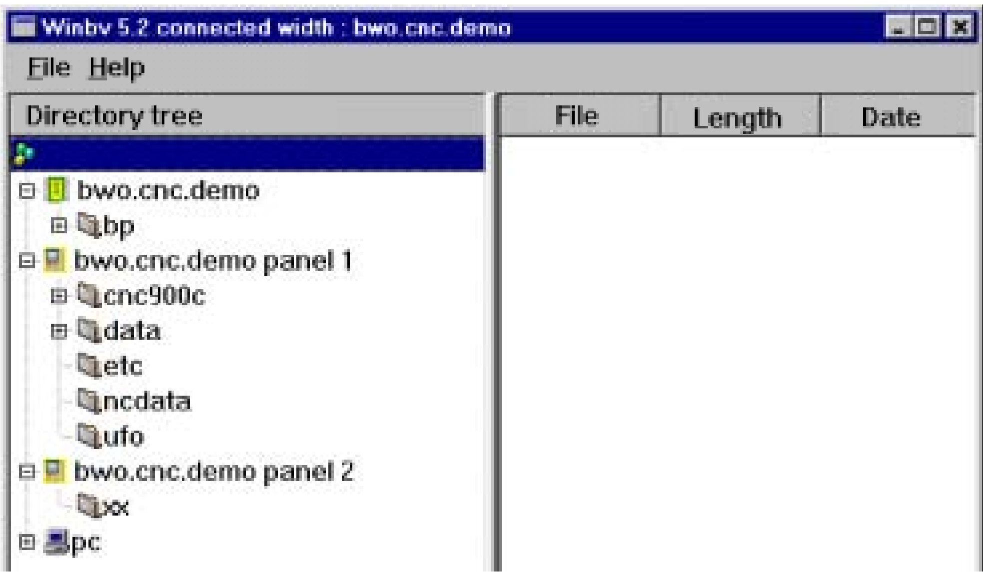
View after the start

Connections to the CNC CPU and to operating panel are maked



Total directory structure

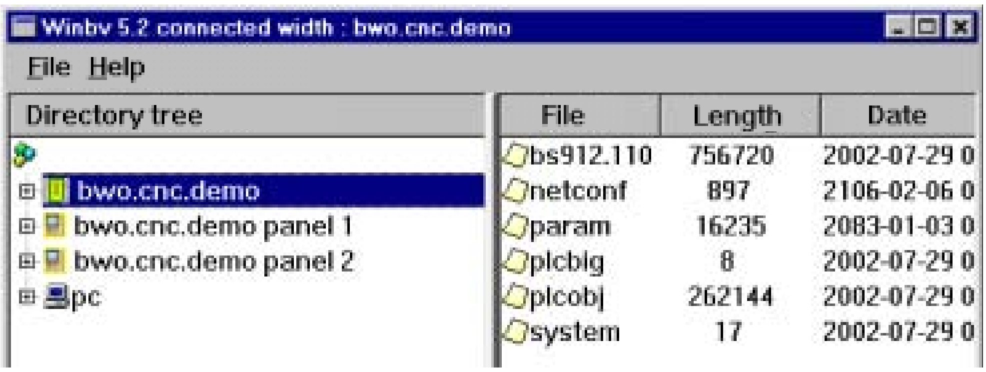
CNC CPU, operating panel 1 operating panel 2 as well as PC



10.1.3 Directories

Directory CNC

Files in general directory CNC



bs912.100 CNC operating system

netconf Ethernet configurations

```
# lines starting with # in the first column are a comment

# define  cnc network adres
local           172.16.60.185
netmask         255.255.0.0
#
# define  panel network adres
control_panel   172.16.60.186   255.255.0.0   000000000000
#
# define  file server adress
host            172.16.60.189
#
# define  route to file servers
route           172.16.60.189   255.255.0.0   0.0.0.0   1
#
Identification  mycompany.mycnc.mynum
#
gateway         xxx.xxx.xxx.xxx
#
iservice        xxx.xxx.xxx.xxx
#
mail1     to:        sub:        msg:
#
serial2  9600 8 e 1
```

param Installation related parameters / machine datas

10.1.3 Directory CNC (continuation)

plcobj	Installation related PLC programm
system	link on the active CNC operating system / link—>bs912.100

Notes to the Ethernet configuration

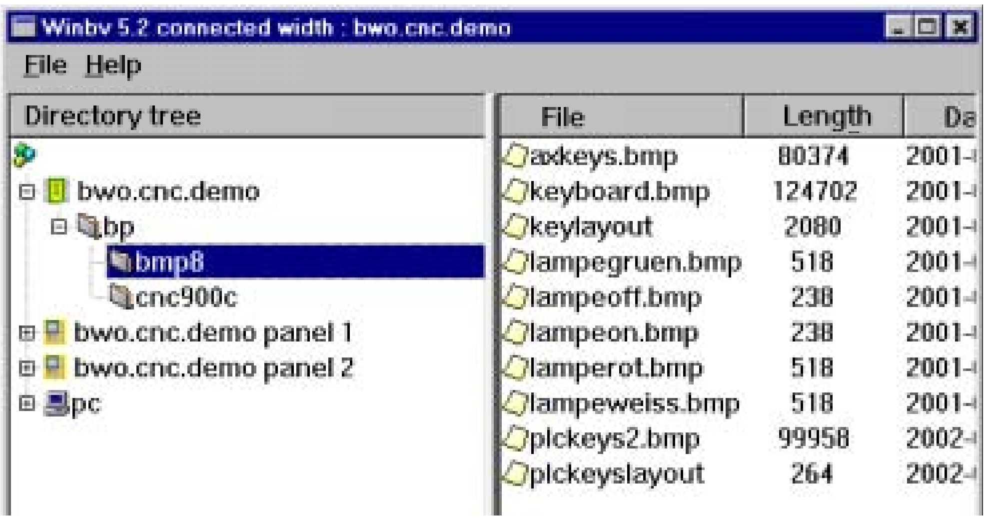
local	own CNC address
netmask	own network mask
control_panel	Control panel address, network address, Ethernet address. Ethernet address 0 means: each operating panel is accepted. Control_panel can be indicated up to 4 times.
host	file server address (must be equipped with SERVER.EXE).
route	commands for the integrated rooter PC address, gateway address, Ethernet connection number 0 (E1), 1 (E2)
identification	symbolic term of the controller The recommended sequence: machine producer. user. machine no.
gateway	Address of computer in the house network, which makes the acces to internet.
iservice	Address of the BWO central internet computer for on-line remote diagnostics (inquire address please from BWO).
mail1 : mail9	to: address@company.com sub: concern msg: messages In the concern and in the message contents of parameters can be transmitted. to it control characters are to 3 at the disposal : % $ # % the total parameters transmit $ only the lower 16 bits of the parameter transmit # the identification of the machine transmits Example:
mail1	to: myaddress@company.de sub: # msg: machine is with message $:8509 Clearing by an error message at the machine. Max. 256 characters.
serial1 / serial2	configuration of the serial interfaces 1 and 2 baudrate bits parity stopbit parity: e (even) / o (odd) / n (no)

10.1.3 Directory CNC (continuation)

Bit-map files in the directory CNC \ BP \ BMP8

BP actual the subdirectory for the operating panel.

All files off this directory are read with ‘ boot ‘ by the operating panel.



axkeys.bmp	layout for the axis selection keys
keyboard.bmp	key board for inserting on toach screen
keylayout	keyboard layout for BWO virtual keyboard
lampegruen.bmp	PLC keys T1 –T8 ‘ key actuates ‘ inserted green box
lamperot.bmp	PLC keys T1 –T8 ‘ key actuates ‘ inserted red box
lampeweiss.bmp	PLC keys T1 –T8 ‘ key does not actuate ‘ inserted box off
lampeon.bmp	PLC keys T1 –T40 ‘ key actuates ‘ inserted red square
lampeoff.bmp	PLC keys T9 –T40 ‘ key does not actuate ‘ inserted red square off
plckeys2.bmp	Layout for PLC keys T9 – T40

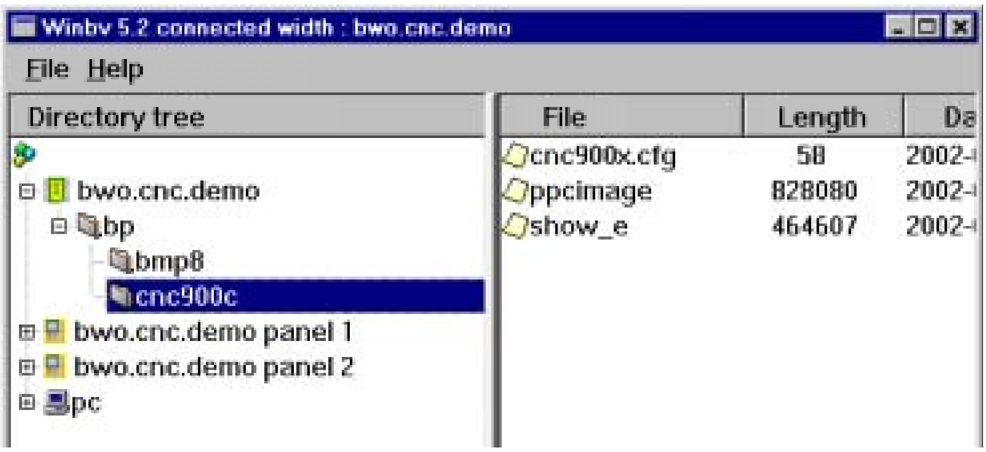
Note:
The files **axkeys.bmp** and **plckeys2.bmp** can be provided by the customer with installation related letters or layouts. For this BWO recommends the program ‘ Micrografx Picture Publisher ‘.

10.1.3 Directory CNC (continuation)

Files in the directory CNC \ BP \ CNC900C

BP actual the subdirectory for the operating panel.

All files off this directory are read with ‘ boot ‘ by the operating panel.



cnc900x.cfg Configuration file for releasing from directories with path specification (here NCDATA).

beep:off	‘bleeper‘ ON / OFF
DRIPFEED://server/ncdata	drippfeed mode
NCDATA1:/ncdata	flash disk (operating panel)
NCDATA2://server/ncdata	peripheral file server

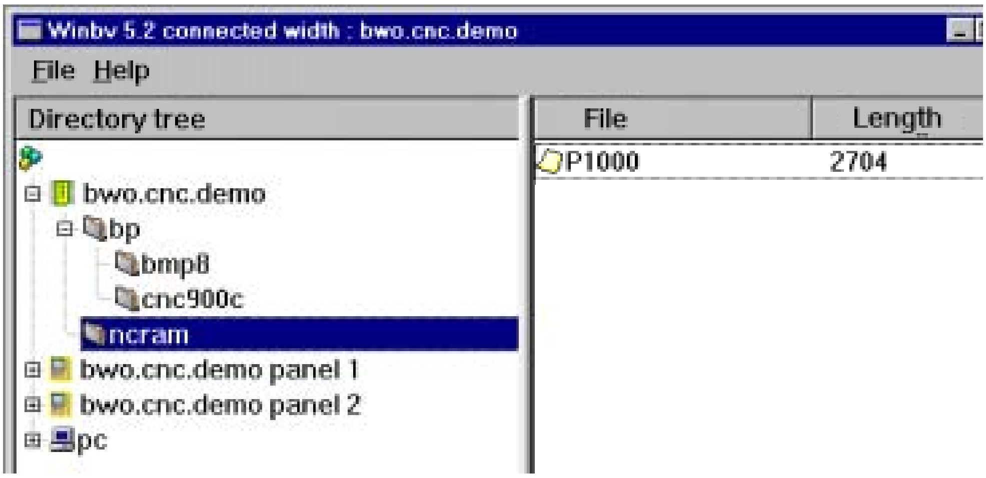
ppcimage operating system operating panel CNC910 / CNC920

show_e installation related user surface / pictures

10.1.3 Directory CNC (continuation)

Files in the directory CNC \ NCRAM

NC programs in the RAM memory of the CNC

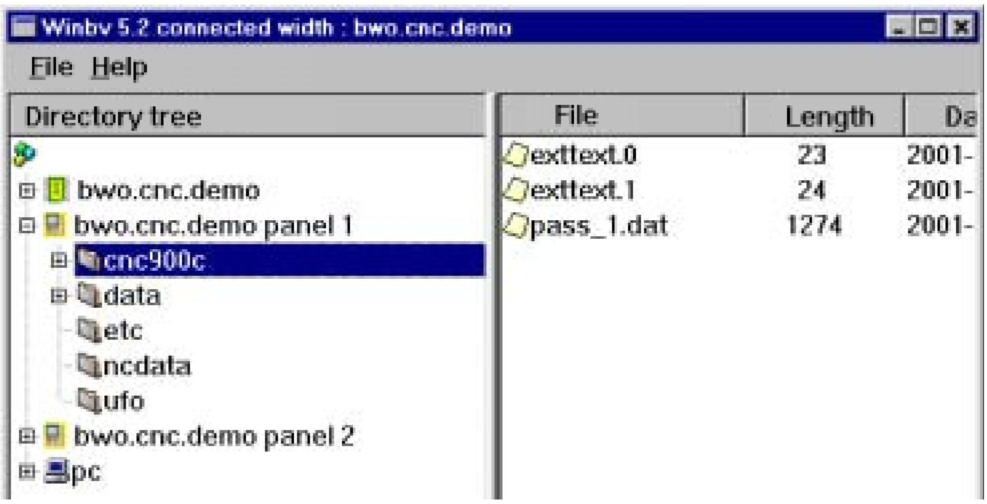


10.1.3 Directory CNC (continuation)

Directory PANEL

Files in the directory PANEL \ CNC900C

PANEL actual the general directory on the Flash disk of the operating panel



These are installation related files (secured on the flash disk)

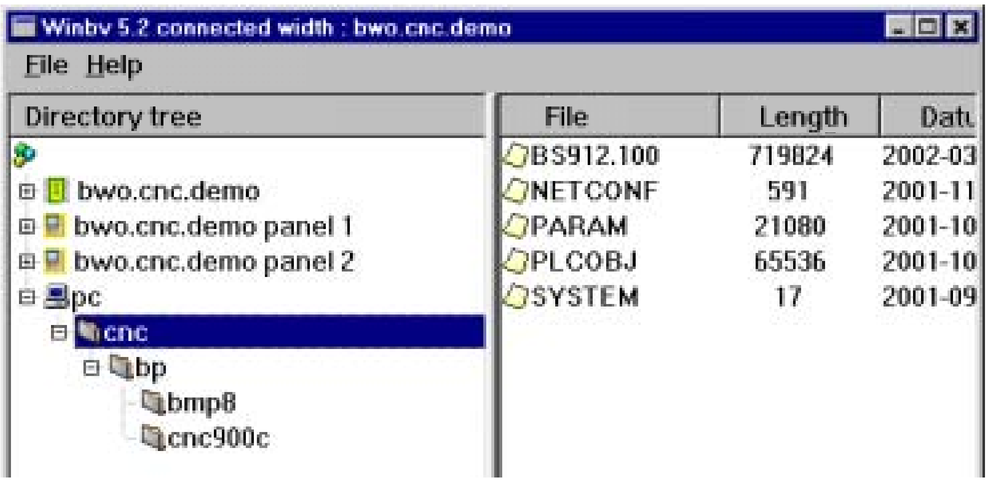
NC programs
zero points
tool datas
parameter (axis machine data)
parameter (SERCOS final drive data)

The directory PANEL on the flash disk of the operating panel can cover still further directories, which contain various configuration files as well as user-referred files or texts due to application.

10.1.3 Directory CNC (continuation)

Directory PC

Structure of the directory PC



The subdirectories can be selected and organized freely however are recommended it to structure the organization of the subdirectories after the following system.

For each application on subdirectory, which again all subdirectories for total application contains (CNC CPU, PANEL) .

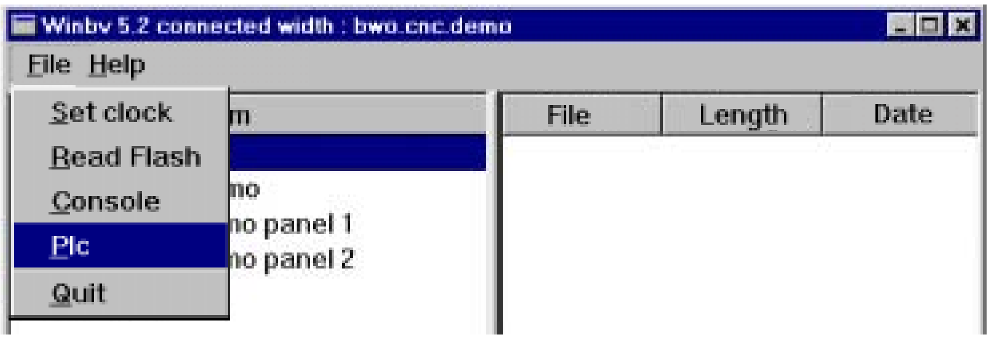
10.1.4 System functions

WINBV System functions

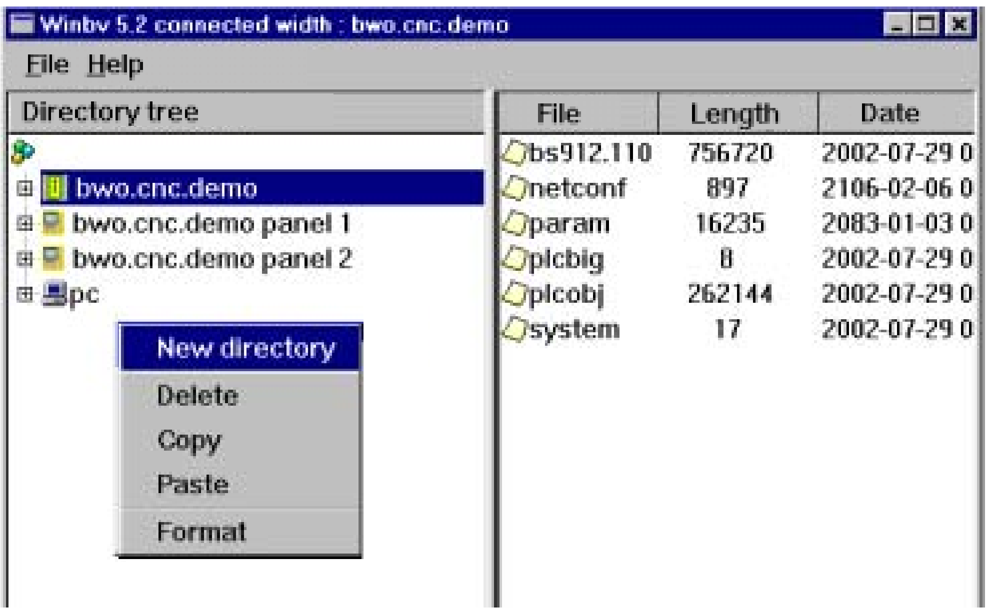
Directories and files can be created, copied, inserted, deleted as well as processed by comfortable functions.

Thus the possibility consists to create backup on the PC by each application on.

Menü File

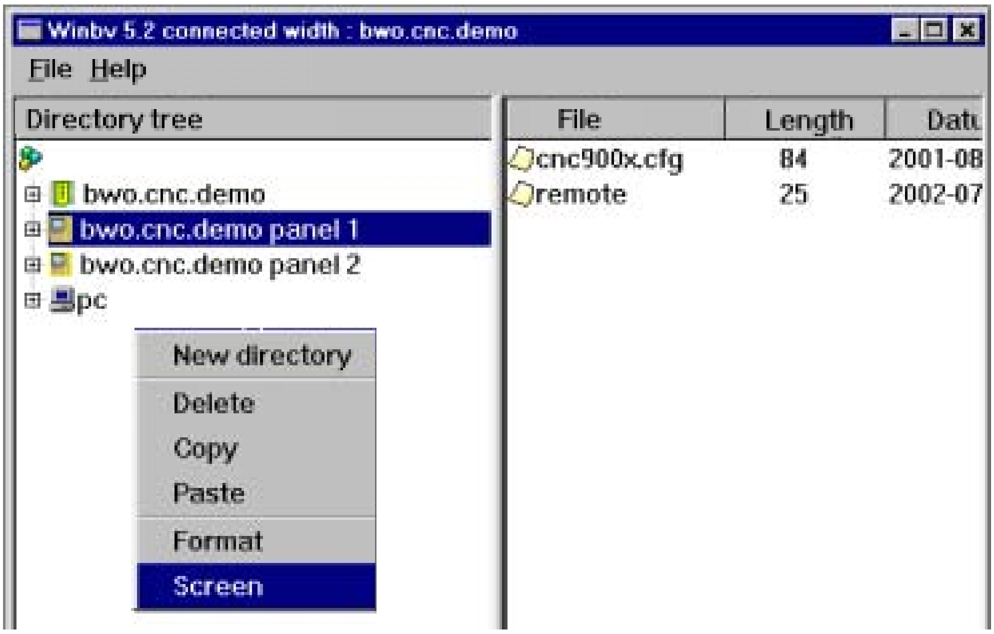


Functions with directory CNC
(right mouse button)

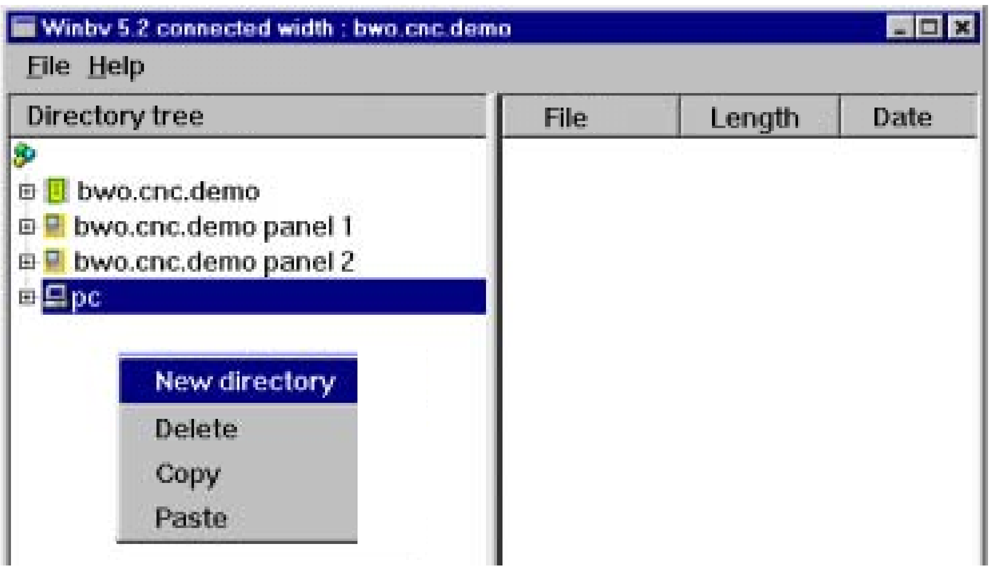


10.1.4 System functions (continuation)

Functions with directory panel
(right mouse button)



Functions with directory PC
(right mouse button)

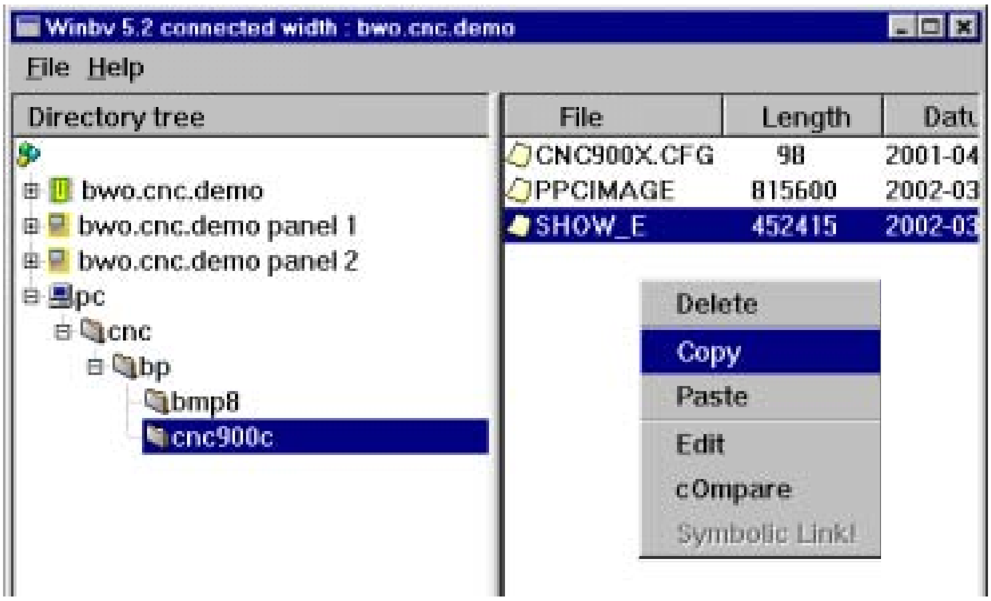


10.1.4 System functions (continuation)

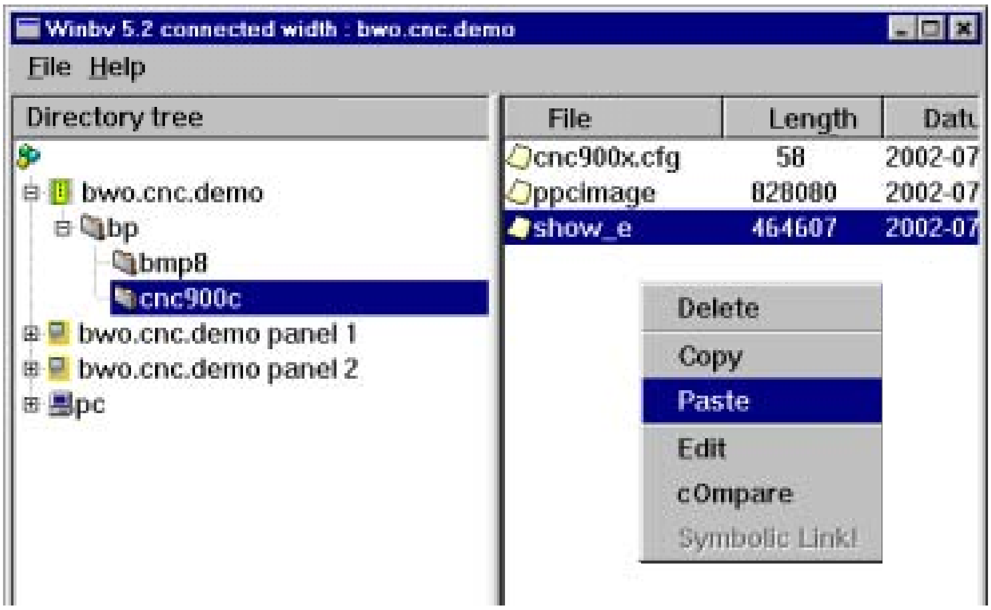
Functions with file
(right mouse button)

Example copying the file SHOW_E

from PC\CNC\BP\CNC900C



to CNC\BP\CNC900C



10.1.5 Remote diagnostics

Mode of operation of the BWO remote diagnostics with internal firm network

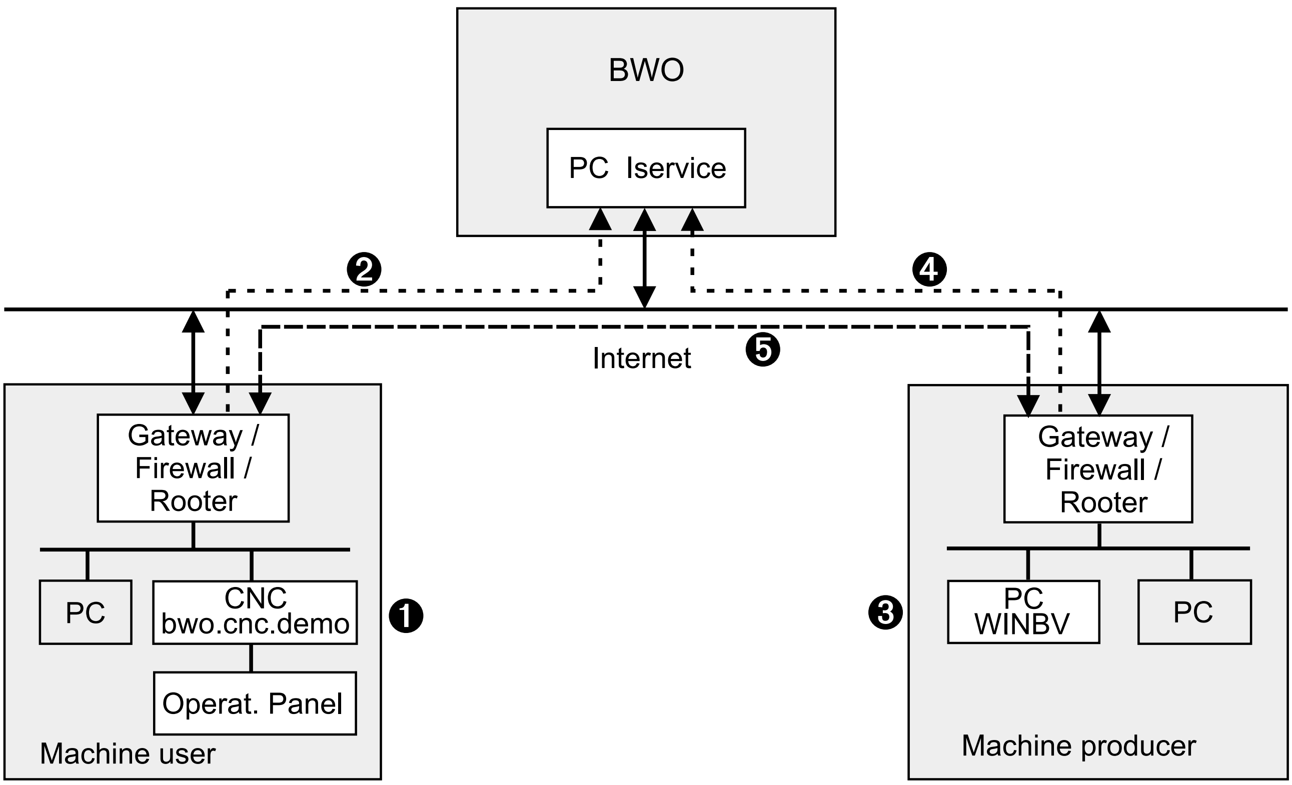
❶ CNC control of the machine user.

❷ The CNC control of the machine user goes with an incident (at the request of the operator) on-line and sends over internet a message with the own CNC address (e.g. bwo.cnc.demo) to the internet PC by BWO (Iservice address).

❸ Program WINBV on PC of the machine producer.

❹ The program WINBV asks the BWO internet PC whether the CNC of the machine user is on-line actual.

❺ The BWO internet PC links the CNC directly with programm WINBV. Now can of the machine producer remote diagnostics of the machine to be executed.
On the Firewall computer of the machine user the IP masquerading must for suitable devices de-energised. Bond BWO with questions please



10.1.5 Remote diagnostics (continuation)

Mode of operation of the BWO remote diagnostics with modem

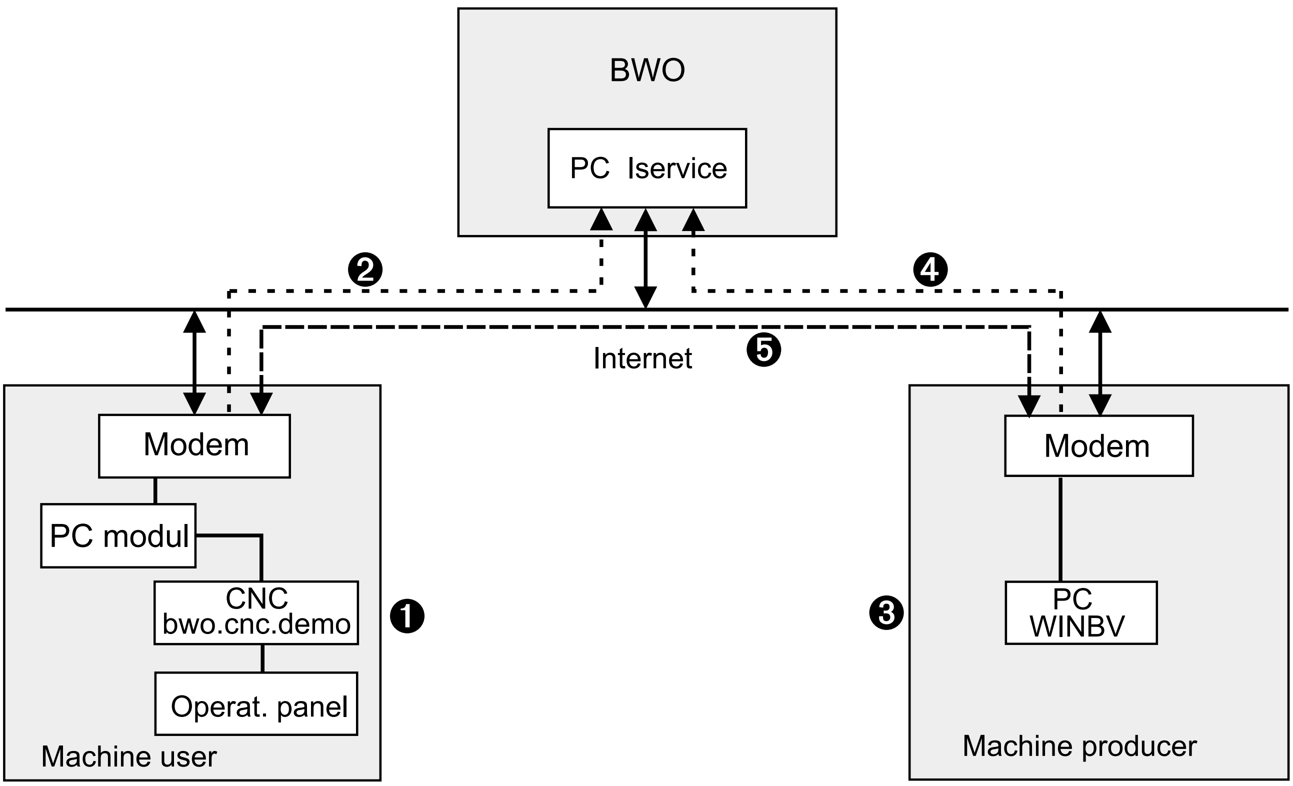
❶ CNC control of the machine user.

❷ The CNC control of the machine user goes with an incident (at the request of the operator) on-line and sends over internet a message with the own CNC address (e.g. bwo.cnc.demo) to the internet PC by BWO (Iservice address).

❸ Program WINBV on PC of the machine producer.

❹ The program WINBV asks the BWO internet PC whether the CNC of the machine user is on-line actual.

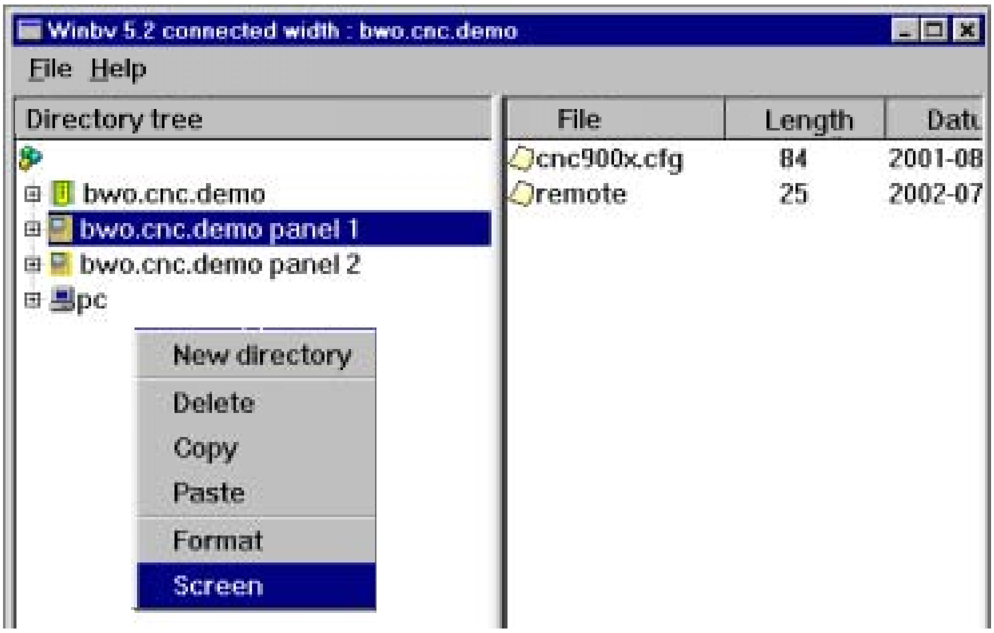
❺ The BWO internet PC links the CNC directly with programm WINBV. Now can of the machine producer remote diagnostics of the machine to be executed.



10.1.6 Display of the operating panel

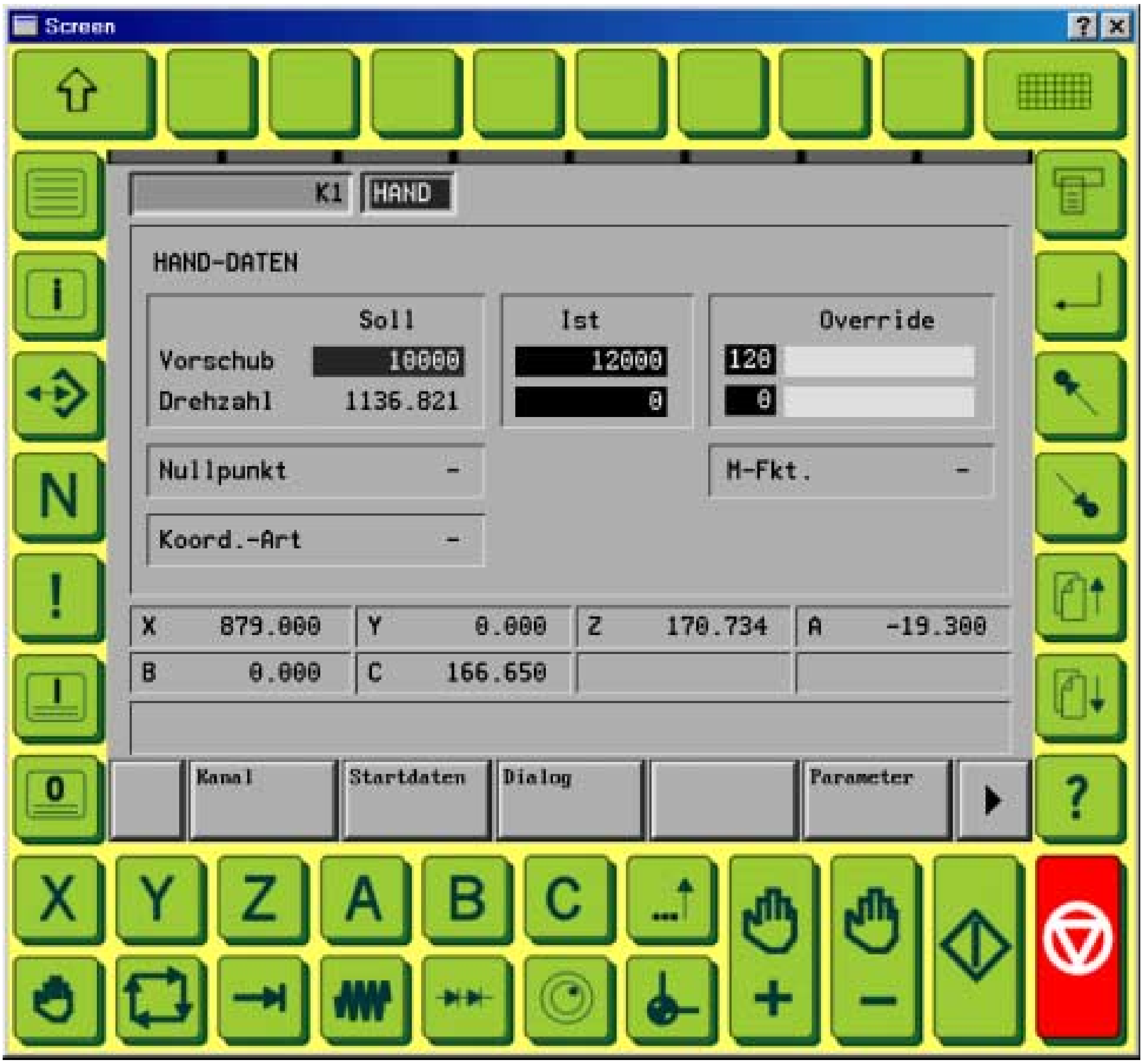
Switching on

Function Screen
(right mouse button)



10.1.6 Display of the operating panel (continuation)

The display is illustrated. The sceen contents actual black-and-white.

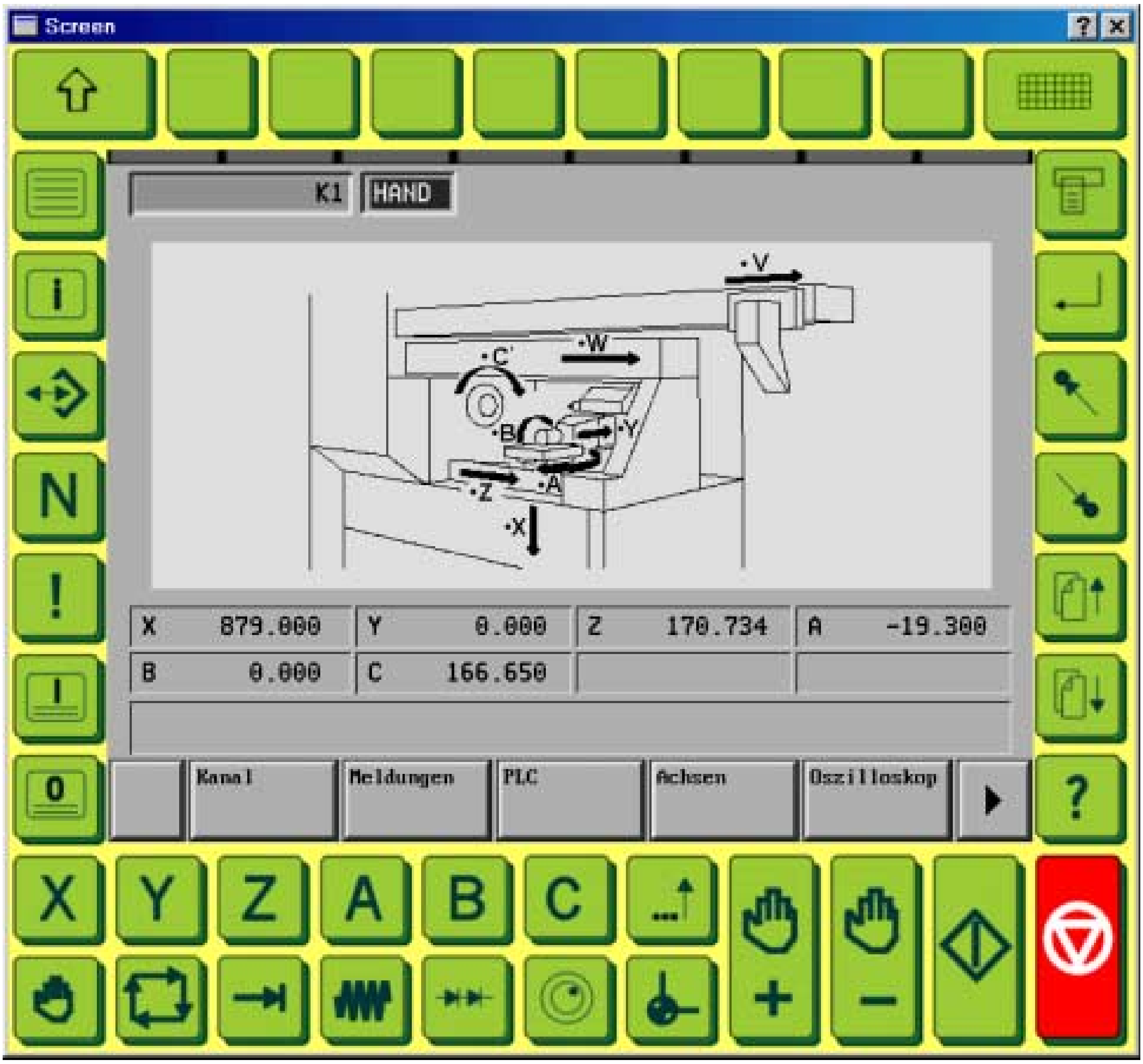


10.1.6 Display of the operating panel (continuation)

I/O circulation

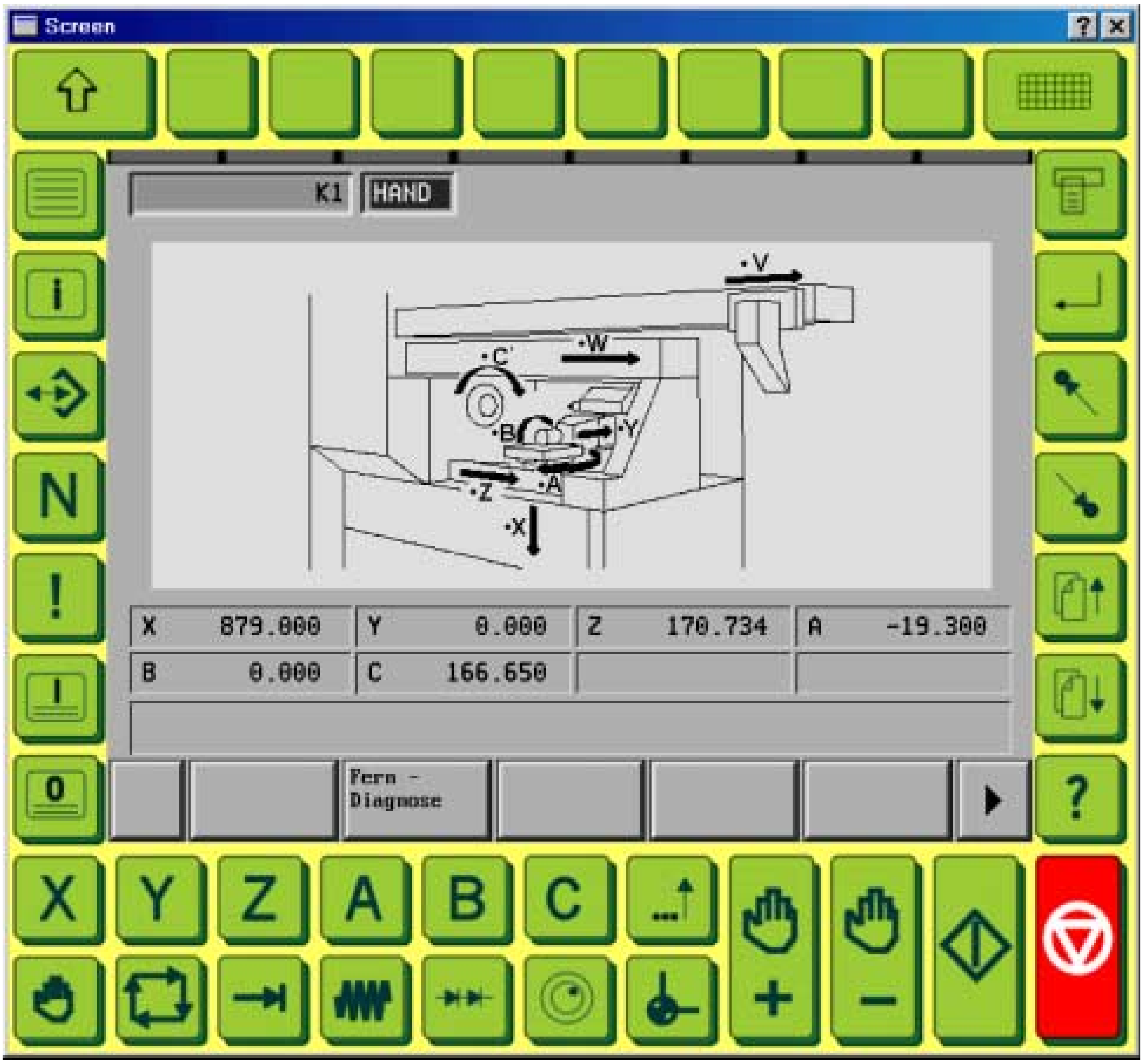
10.1.6 Display of the operating panel (continuation)

Diagnosis
Switching on with the key „!“



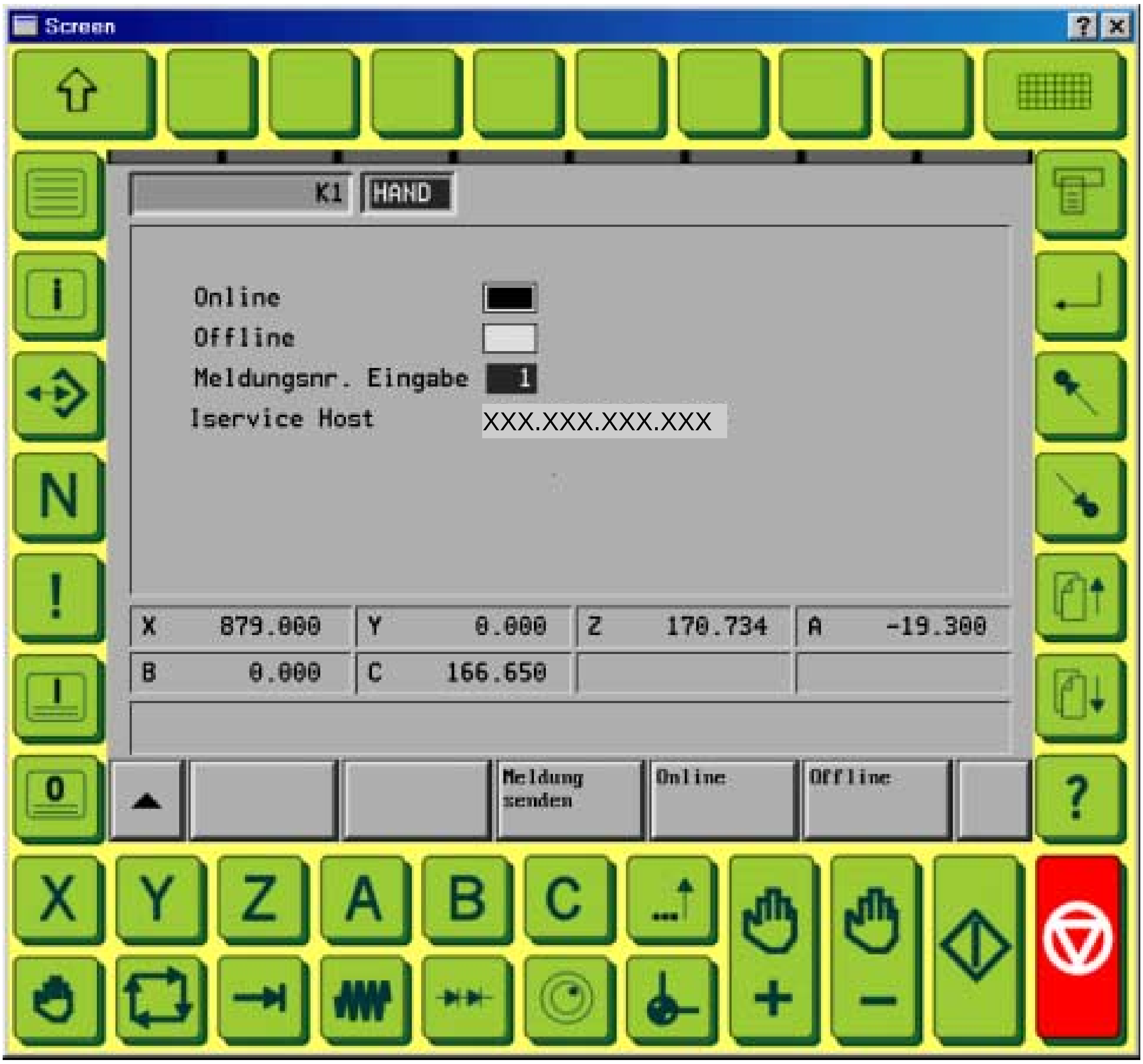
10.1.6 Display of the operating panel (continuation)

Remote diagnostics
Switching on with the keys „!“ and „ remote diagnostics „



10.1.6 Display of the operating panel (continuation)

Switching with the keys „ on-line “ / „ off-line “



10.2 Operating system administration

10.2.1 Boot program

The BOOT EPROM contains the boot program for the CNC module. The boot program initialize the hardware of the CNC module and start the actual operating system, which is in the FLASH EPROM of the CNC module. The user has the possibility of interrupting this flow with the transition from the initialization program to the operating system and of starting a utility routine, which operates the service interface. With the program BV.EXE can be managed then from the PC the operating system on the CNC module. Interrupting the flow takes place from the user through printing of the RESET key at the plug of the interface cable.

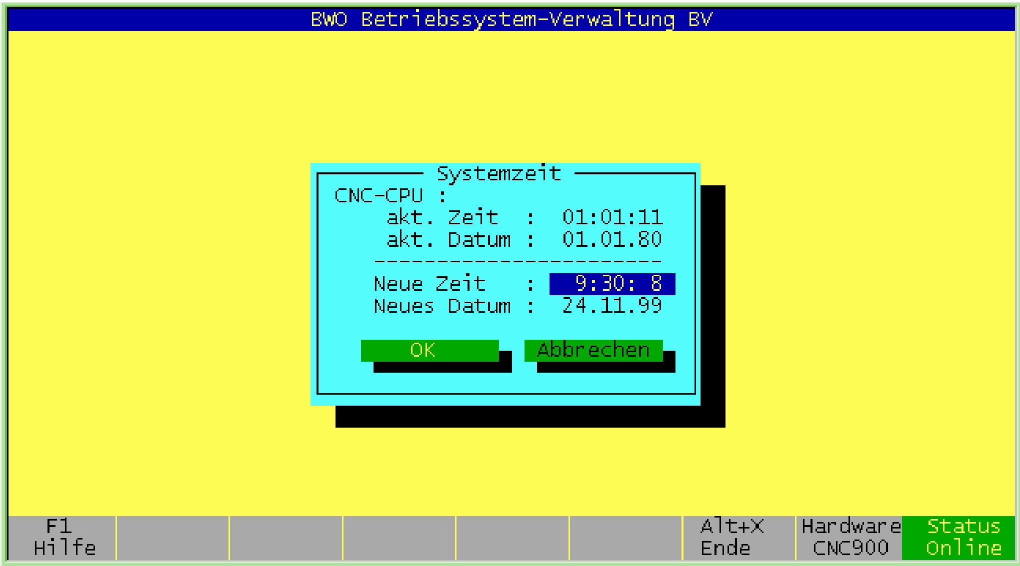
10.2.2 Operating system management program BV.EXE

for CNC CPU´s, which with the help of "P-link adapter 083767" are programmed.
- CNC CPU32B 083671 und 088671
- CNC CPU64B 084564

Menu after the call by BV.EXE

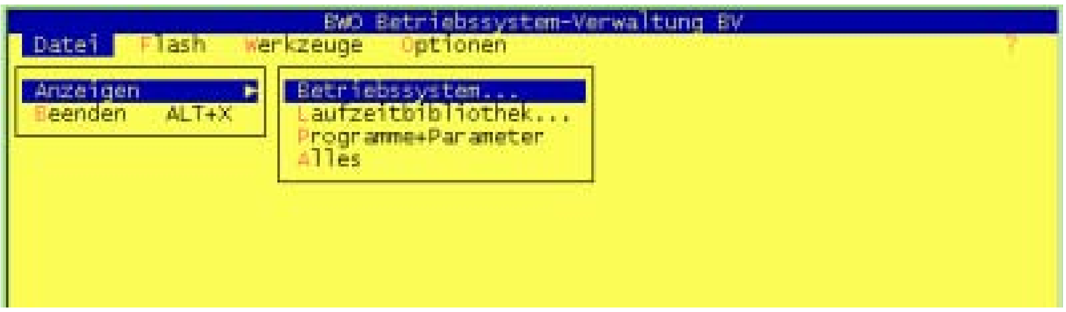
Now the link RESET key press and immediately after it through to printing of the RETURN key acknowledge.

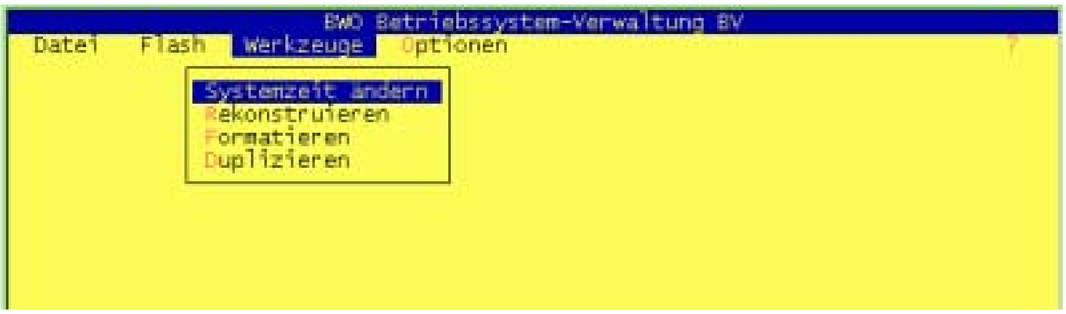
If the system time of the CNC CPU with the system time of the PC differs, the following menu comes to the display. Here now the system time of the CNC CPU can be set. One suggests the system time of the PC.



10.2.2 Operating system management program BV.EXE (continued)

Basic menus





10.2.2 Operating system management program BV.EXE (continued)

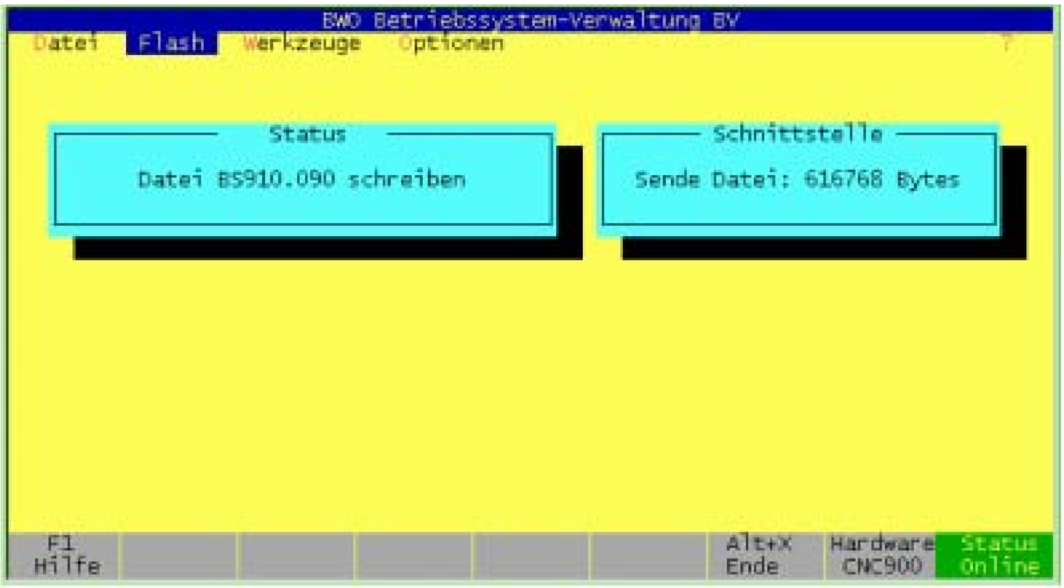
Example: **Operating system write / update**

With the selection ‘ Flash ‘ ‘ writing ‘ is opened the following selection menu.

In this menu the file which can be written is selected and acknowledged.

Subsequently, further menus follow

10.2.2 Operating system management program BV.EXE (continued)



The status indication of the CNC CPU displays again ‘ 0 ‘, is the recording procedure terminated.

10.2.2 Operating system management program BV.EXE (continued)

Example: Operating system read

10.3 Archiving program NCARC

With the help of the archiving program NCARC.EXE can be in or read NC data (NC programs - parameter tables - machine data) over a serial interface into the controller.

Connection V24:
PC COM1 or COM2 < ————> peripheral device interface CNC900

Start menu

NC - Archiving and administration program		BWO - Elektronik
===> E	Input of NC data	(from controller —> in PC)
===> A	Output of NC data	(from PC —> in controller)
===> B	NC data or error message	(display output)
===> P	I/O interfaces parameters	(modify / a checking)
===> Q	Program end	

10.3 Archiving program NCARC (continued)

Menu output of NC data

NC programs - parameter tables - machine data

NC - Archiving and administration program BWO - Elektronik

MENU = > A < = output of NC data (PC —> controller)

Program file name input: ===>????

Menu input of NC data

NC programs - parameter tables - machine data

NC - Archiving and administration program BWO - Elektronik

MENU = > E < = input of NC data (controller —> PC)

Program file name input: ===> ?????