SwanSoft NANJING SWANSOFT

SWAN NC SIMULATION SOFTWARE

SINUMERIK SYSTEM
INSTRACTION OF OPERATION AND PROGRAMMING

Nanjing Swan Software Technology Co.,Ltd.
Version 05/2007

CONTENTS

CHAPTER 1 SUMMARY OF SWAN NC SIMULATION SOFTWARE

1.1 BRIEF INTOUCTION OF THE SOFTWARE

Based on factories' manufacturing and colleges' teaching experience, Nanjing Swan Software Technology Co., Ltd developed the following software: FANUC, SIMUMERIK, MITSUBISHI, GSK, HNK, KND, DASEN, and simulation software. Through which, we can attain the aim of enabling students to have the experience of practical manipulation on a largely-reduced cost.

Swan series NC simulation software can be furthere devided in 8 major types, 28systems and 62 controlling surfaces. Equipped with FANUC, SIMUMERIK, MITSUBISHI, GSK, HNK, KND, DASEN software, swan NC simulation software can help students to learn operation of NC milling tool, lathe and machining center of each system. Meanwhlie CAM NC program can be programmed or read in by manual.By internet teaching,teachers can have the first-hand information of their students'current manipulating condition .

1.2 FUNCTION OF THE SOFTWARE

1.2.1 CONTROLER

1. The screen configrations can be realized and all the functions are the same with CNC machine used in the industrial system.
2. Interprets NC codes and edits cutting feed commands of machine real-timely.
3. Operation panels are similar with the real NC machine can be provided.
4. Single brick operation，automatic operation,editing pattern,dry running,and so on.
5. Rate of travel adjusting, change over switch of unit millimeter pulse.

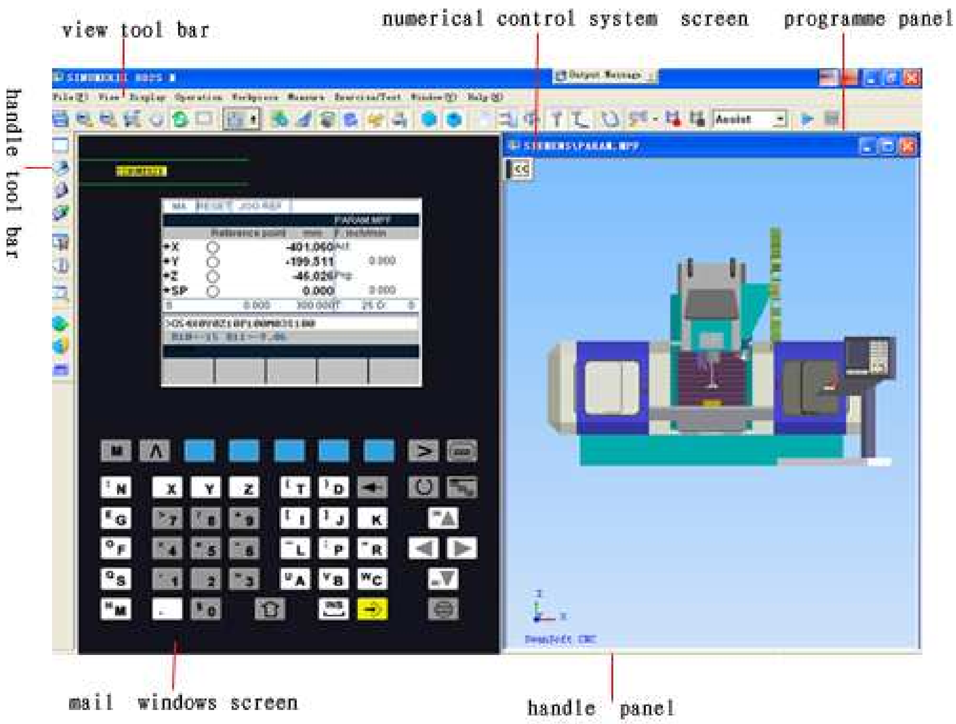


Fig.1.2-1 siemens 802s/cM(milling machine)

（1）Choose the blank function key at the left tool frame

（2）Choose reference mandril.

（3）Choose ordinance of reference mandril and thickness of spacer gauge.

（4）Preset workpiecedirectly and confirm that according to special hint on the bottom-left of window.

（5）Coordinate Z workpiece nullpoint = current coordinate Z – length of reference mandril – thickness of spacer gauge.

（6）Put the output:Z、Y、X axes workpiece nullpoint into G54～G59.

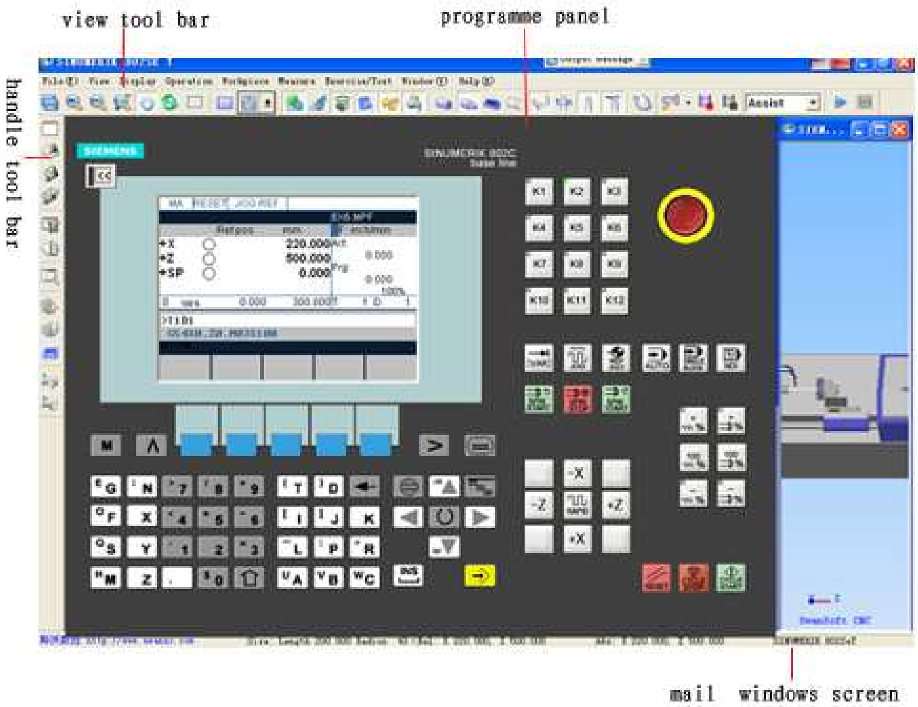


Fig.1.2-2 siemens 802se T

Fig.1.2-3 siemens 801

1.2.2 FUNCTON INTRODUCTION

★ The first domestic NC simulation software which can be downloaded and updated automatically for free.

★ Vivid 3DM NC machine and operation panels.

★ Support ISO-1056 preparatory function code（G code）、secondary function code（M code） and other operation codes.

★ Support system self-defining code and canned cycle.

★ Callin CAD/CAM postposition tailor file such as UG、PRO-E、Mastercam directly for simulation to processing.

★ Windows macro record and playback.

★ AVI files record and playback.

★ Placement and mounting of workpiece.

★ toochange mechanical hand、square-tool rest、all direction- tool rest.

★ rectifying tool by benchmark、rectifying tool by test cutting .

★ Components cutting，with processing coolant、processing sound、scrap iron and so on.

★ Tools such as edge detector、spacer gauge、micrometer、caliber rule.

★ Adopt data base management tools and performance parameter library.

★ There are many kinds of tools.

★ Support custom-defined tool function.

★ 3DM measurement function of processed model.

★ Measurement of components roughness based on cutting parameter of tools.

CHAPTER 2 OPERATIONS OF SWANSC NC SIMULATION SOFTWARE

2.1 STARTUP INTERFACE OF THE SOFTWARE

2.1.1 STARTUP INTERFACE OF PROBATIONAL VERSION

Fig. 2.1-1

（1）Choose PROBATIONAL VERSION in the left document frame.

（2）Click the left window to choose NC system needed.

（3）You can also select Super Demo if needed.

（4）Click Try It to login system interface after choose one system.

2.1.2 STARTUP INTERFACE OF NETWORK VERSION

Fig. 2.1-2

（1）Choose NETWORK in the left document frame.

（2）Choose the name of system needed in the top bar-frame at right.

（3）Choose your custom name and input password in the below tow frames.

（4）Choose between Remember Me and Remember My Password.

（5）Input the IP address of server.

（6）Click Sign in to login system interface.

（7）Startup SSCNCSRV.exe to login the main interface of SERVER,as the following Fig. show:

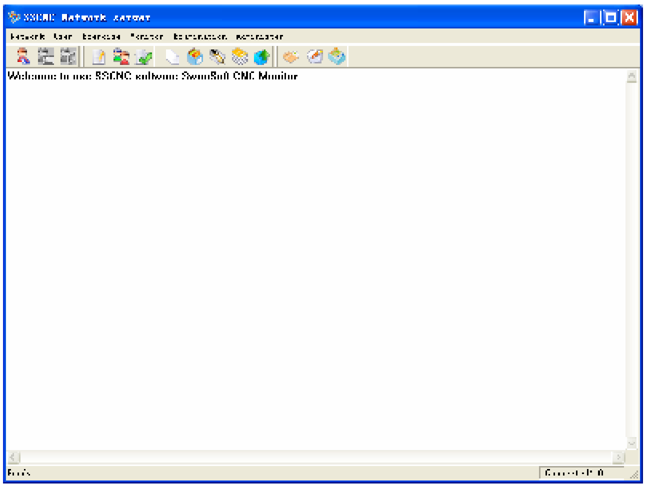


Fig. 2.1-3

（8）After click the icon“CUSTOM STATUS” in toolbar，it will show all the custom status,as the following graph show:

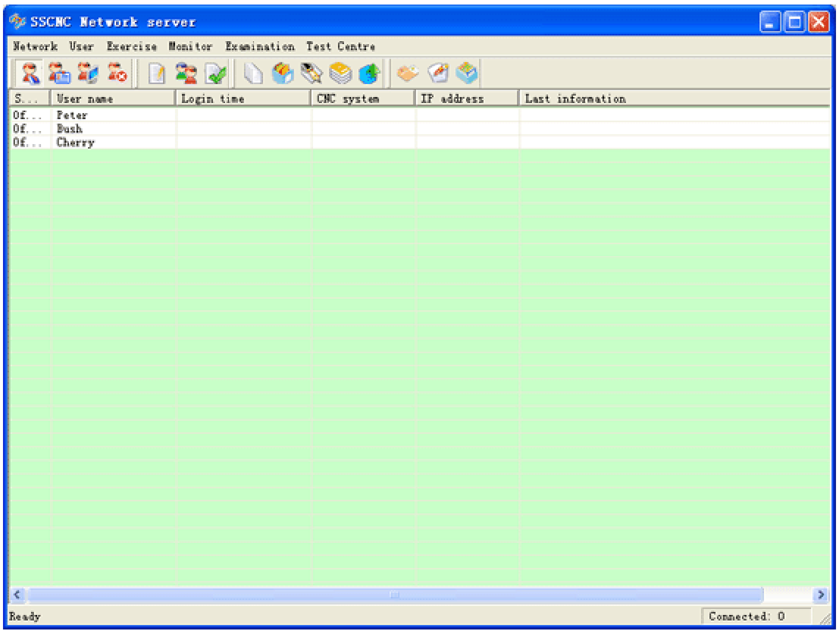


Fig. 2.1-4

（9）Choose a custom in Custom Statue List,and then click the icon "SET TEACHER’S COMPUTER" to set it Teacher’s Computer.

（10）After click the icon "CUSTOM MANAGEMENT", a dialog box " CUSTOM MANAGEMENT " will pop-up,as the following graph show:

Add custom name and its authority in the dialog box one by one or by batch.

a. In one by one pattern, input custom name ,name, secret code and code confirmation,and also you can set necessary authority then clik SAVE.

b. In batch pattern, input start numbering and number of customs, and also you can set necessary authority then clik SAVE.

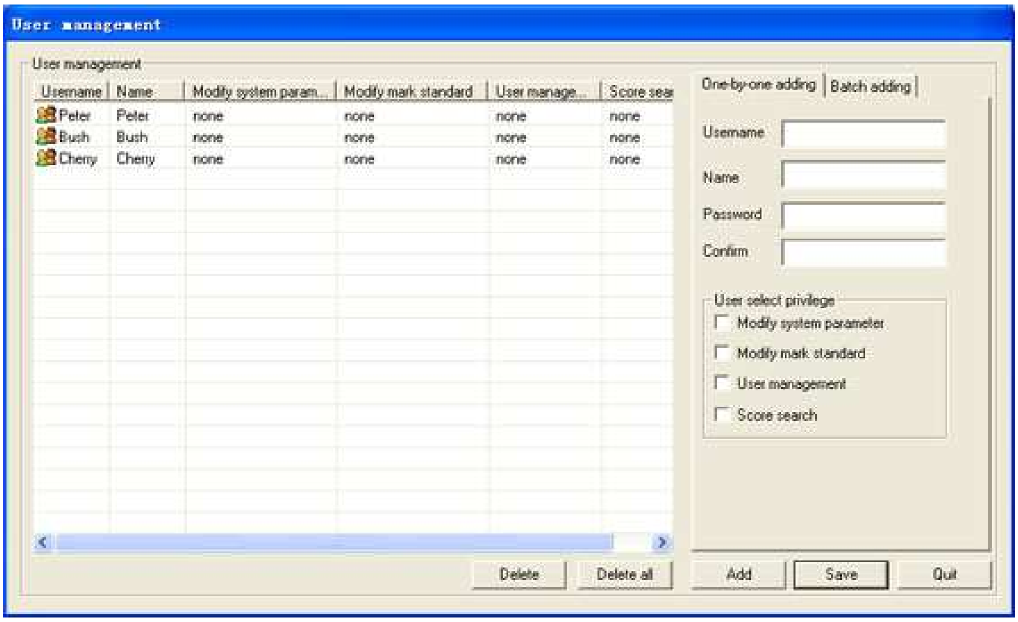


Fig. 2.1-5

2.1.3 SINGLE MACHINE VERSION STARTUP INTERFACE

Fig. 2.1-6

（1）Choose SINGLE MACHINE VERSION in the left document frame.

（2）Choose the name of system needed in the right bar-frame.

（3）Select one option between PC Encryption and Softdog Encryption.

（4）Click Run to login system interface.

2.2 SETUP OF TOOLBAR AND MENU

All the commands can be executed from the left toolbar in the window.System will show the name of its function when cursor points each button,and meanwhile the tip help of the function will be showed in the bottom statue bar.

Brief introduction of toolbar：

Setup new NC file

Open saved file(such NC file)

Save file(such as NC file)

Save as

Machine parametar

Cutter library management

Pattern of workpiece display

Choose size of workblank and coordinate of workpiece

Open/close machine door

Scrap iron display

Screen arrange：change screen arrange function by fixed sequence

Whole screen zoom up

Whole screen zoom down

Screen zoom up,zoom down

Screen translation

Screen revolve

X-Z plane selection

Y-Z plane selection

Y-X Plane selection

Machine encloser swich

Workpiece measurement

voice controler

Coordinate display

Jacket water display

Workblank display

Component display

Clarity display

ACT display

Display tools spacing number

Cutter display

Cutter path

Online help

REC parameter setup

REC start

REC stop

teaching start/stop

2.3 FILE MANAGEMENT MENU

Program file（*.NC）、tool file（*.ct）and workblank file（*.wp）callin and save and relevant function,such as the function used to open or save data file where NC code editing process is put.

Open： Open respective dialog box to choose the code file needed to disply the NC code in window. Process step into auto way automatically after whole code is loaded; Schedule of code is showed on the bottom of screen.

New ： Delete NC code being edited and loaded.If code is alternated system will register that whether to save the code.

Save ： Save the code edited on the screen.If execute this command to new loaded existing file nothing will be changed and system will ask for a new file name in despite of whether the file is loaded just now.

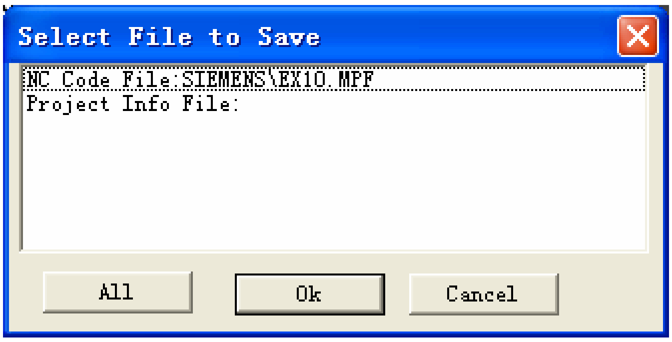


Fig.2.3-1

Save as

Save a file with a new file name known to the existing name.

Load project file

Save all the relevant data files(wp； nc； ct) into a engineering file (extension name：*.pj)，called project file. This function is used to load saved file in new condition..

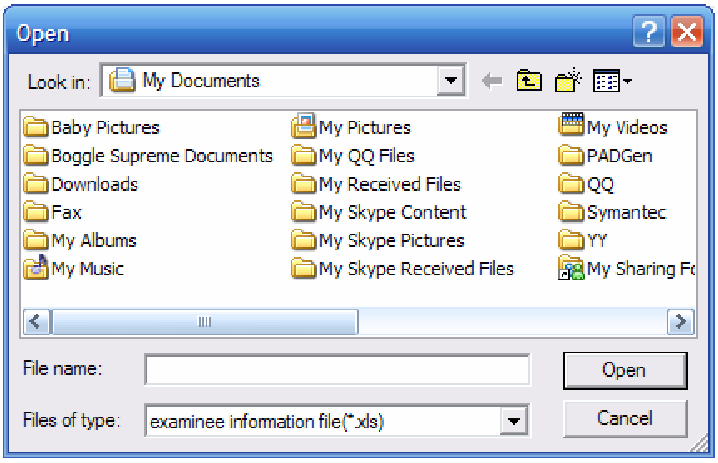


Fig.2.3-2

Project file save

This function save all the handled data into file.The blamx block on screen can be modified.

2.3.1 MACHINE PARAMETER

a. Machine parameter setup：

Drag dieblock of diago box“Parameter Setup”to choose appropriate toochange rate.

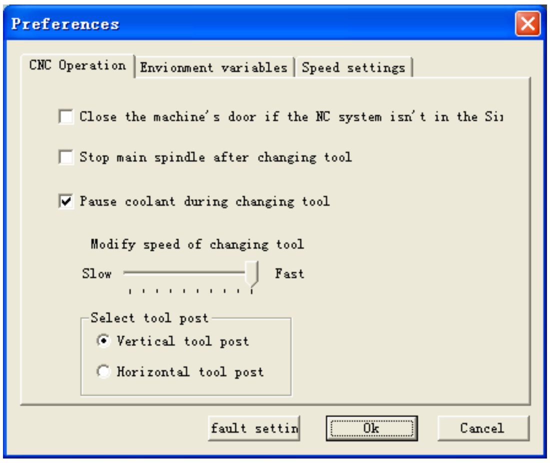


Fig.2.3-3

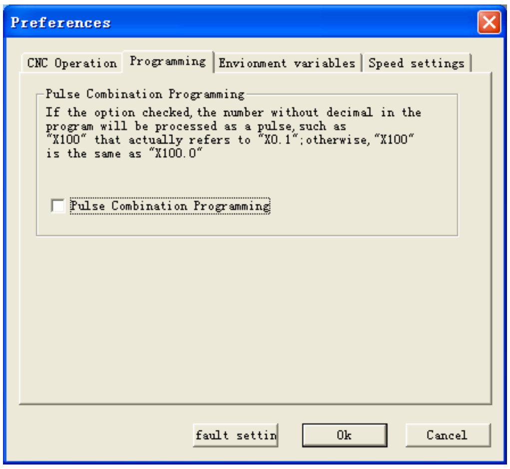


Fig.2.3-4

Click“Color Choose”to change background color of machine.

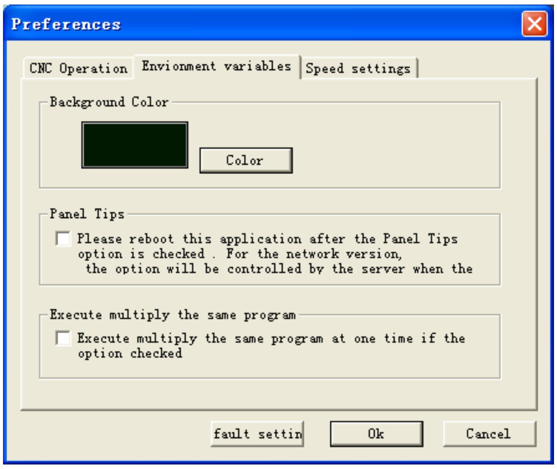


Fig.2.3-5

Adjust“Processing Drawing Display Acceleration”and“Display Precision”to gain appropriate speed of service of simulation software.

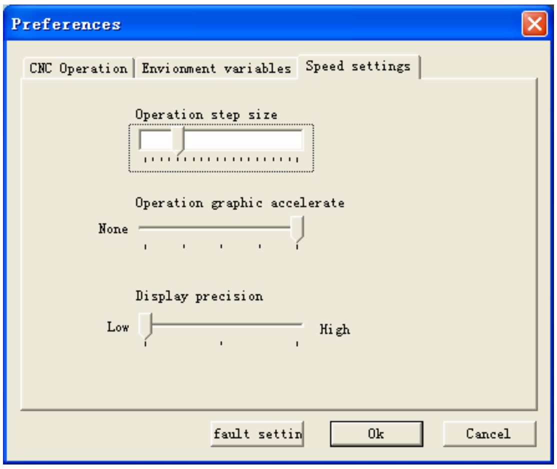


Fig.2.3-6

b.Display color：

Click “Confirm” after choose feeding route and color of machineing.

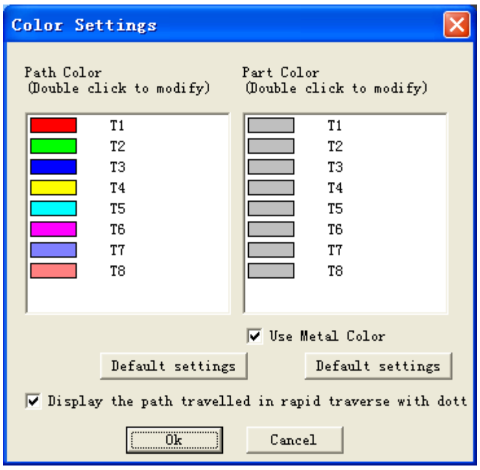


Fig.2.3-7

2.3.2 CUTTER MANAGEMENT

a. Milling machine

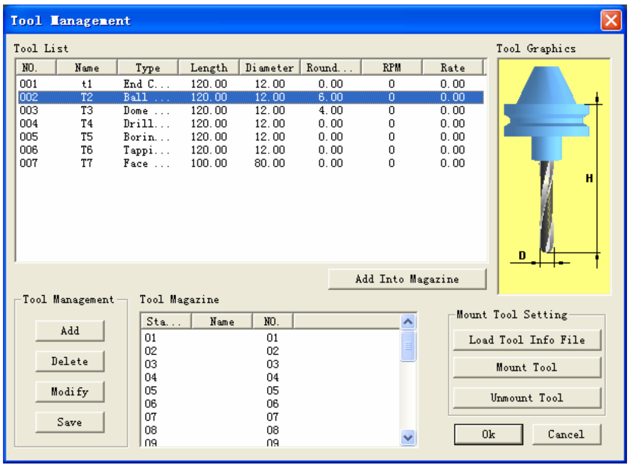


Fig.2.3-8

Add

(1).Input the number of tool

(2).Input the name of tool

(3). End-milling tools、buttonhead tools、dome-end tools、aiguilles、boring tools can be choosed.

(4). Diameter 、length of tool hoder 、rotation rate 、cutting feeding rate can be defined.

(5).Click“Confirm”to add them to tool management library.

Add tool to chief axes

(1).Choose the tool needed in the tool data-base, such as tool “01”.

(2).Press mouse left key and hode it, then pull it to machine library.

(3).Add to top rest, then click “confirm”.

b.lathe

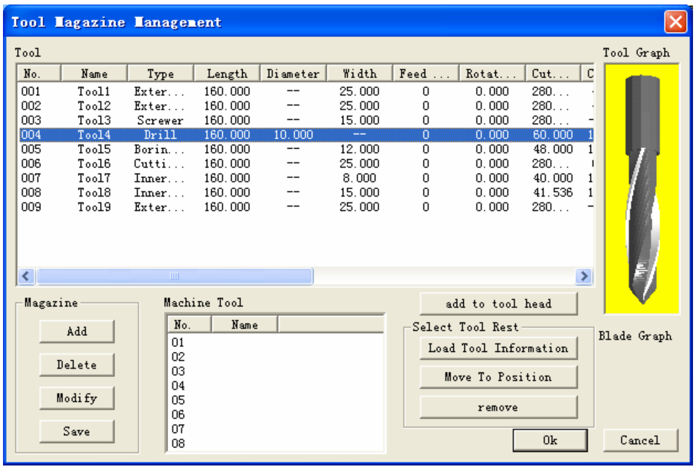


Fig. 2.3-9

add

(1). Input the number of tool.

(2). Input the name of tool.

(3). billmpse tool、cutting off tool、internal tool、aiguille、boring tool、screw tap、screwthread tool、internal screwthread tool、internal circle tool can be choosed.

(4).Many kinds of cutting blade、side length of cutting blade、thicknesscan be defined.

(5). Click"Confirm"to add them to tool management library.

Internal circle tool adding：

(1)Click"add"，popup diago box"add tool"，as the fowing graph show：

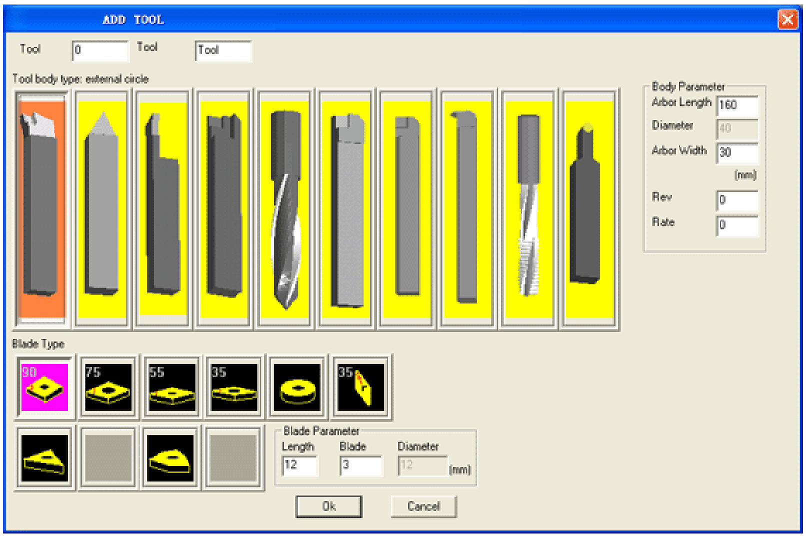


Fig. 2.3-10

(2)Choose bull-nose tool in diago box"add tool"，then popup "tool"， as the fowing graph show：

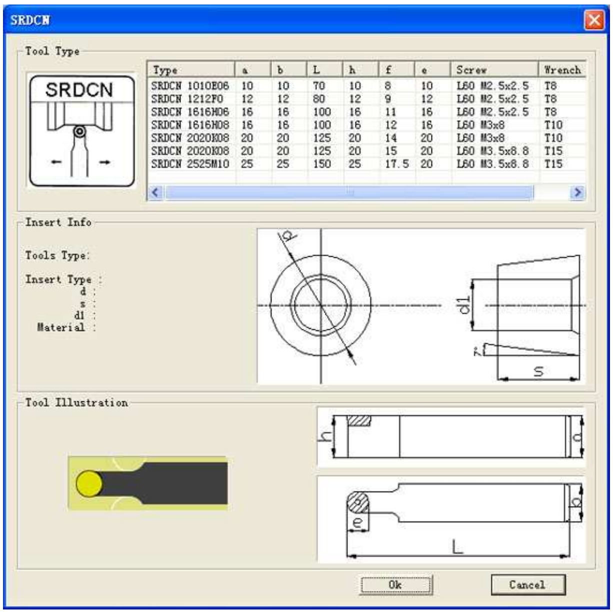


Fig. 2.3-11

(3)Choose the tool needed in diago “tool” and click “confirm”, then reverse back to “add tool”to input the number of tool and the name of tool.

Add tool to chief axes

(1) .Choose the tool needed in the tool data-base, such as tool “01”.

(2). Press mouse left key and hode it, then pull it to machine library.

(3). Add to top rest, then click “confirm”.

2.3.3 WORKPIECE PARAMETER AND ACCESSORY

a. milling machine

Size of workblank、coordinate of workpiece

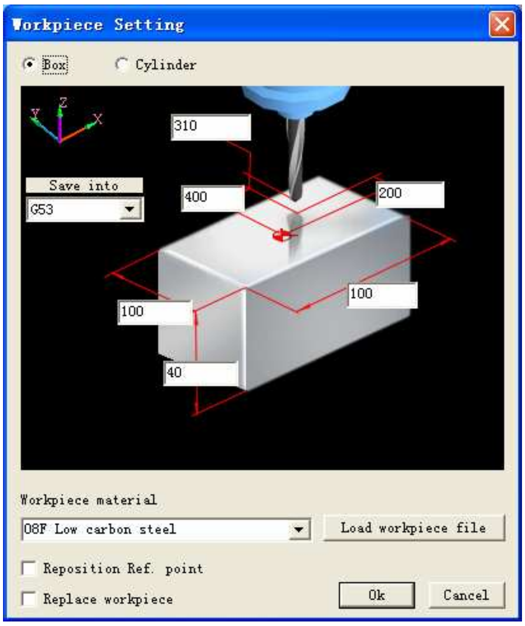


Fig. 2.3-12

(1)Define the length ,width and highness of workblank and its material.

(2)Define orgin of workpiece X、Y、Z.

(3)select changing machining orgin、changing workpiece.

b.Lathe

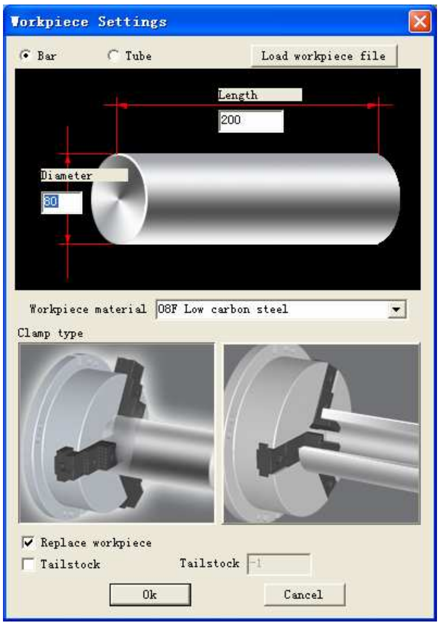


Fig. 2.3-13

(1)Define workblank type, length, diameter and its material.

(2)Define fixture.

(3)Choose tailstock.

Choose workholding fixture

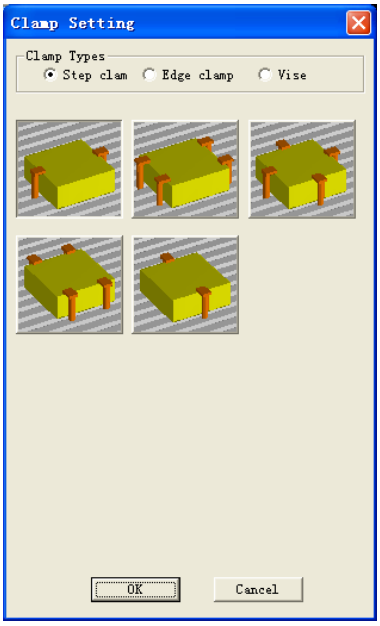


Fig. 2.3-14

Workpiece placement

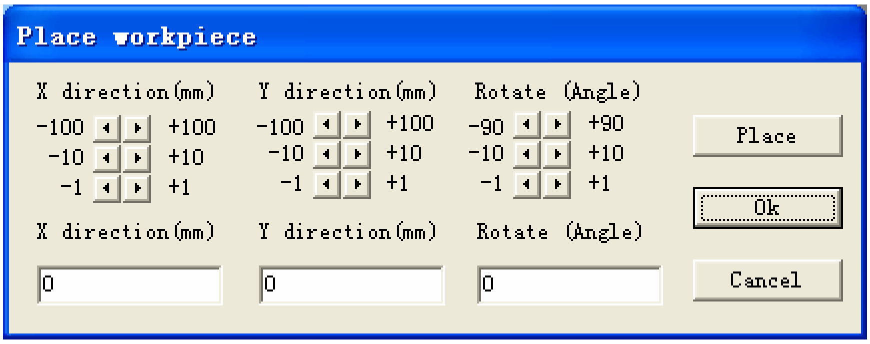


Fig. 2.3-15

(1)Choose the placement of direction X.

(2)Choose the placement of direction Y.

(3)Choose the placement of angle.

(4)Press“Place”and“Confirm”.

Edge detector measures null point of workpiece, so choose the edge detector needed in model list.

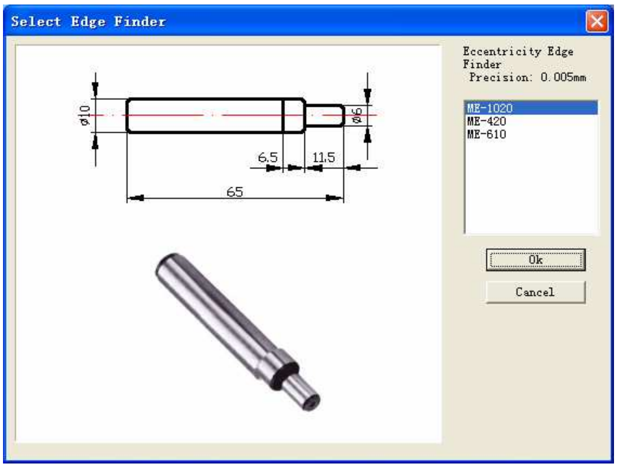


Fig. 2.3-16

Coolant pipe adjusting

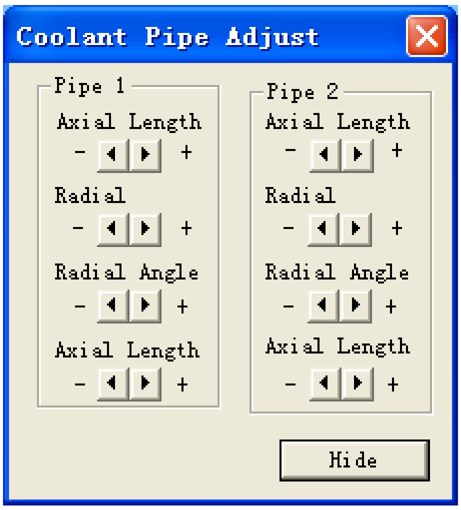


Fig. 2.3-17

2.3.4 RAPID SIMULATIVE MACHINING

(1)Programme by EDIT.

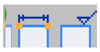
(2)Choose tool. 。

(3)Choose workblank and workpiece null point.

(4)Placement mode AUTO.

(5)Press the key to rapid simulative machining without machining.

2.3.5 WORKPIECE MEASUREMENT



Three modes of measurement

(1)Feature point.

(2)Feature line.

(3)Distribution of roughness.

You can use Up, Down, Left and Right on keyboard to measure size, also you can input value into diago box..

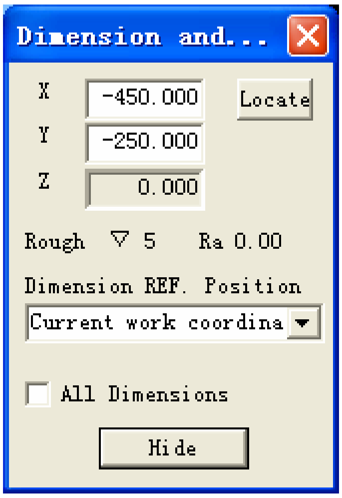


Fig. 2.3-18

2.3.6 REC PARAMETER SETUP

Three modes of REC area selection,setup as

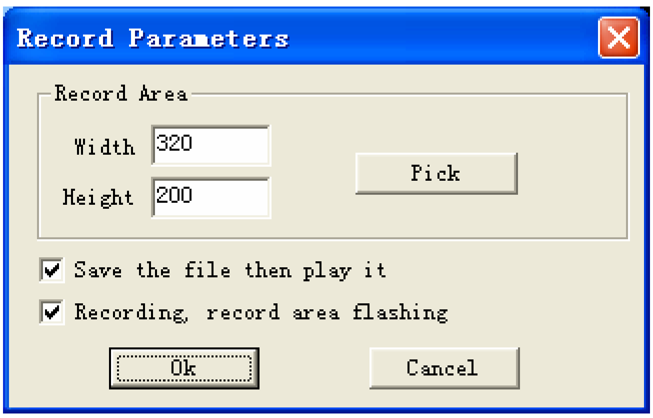


Fig. 2.3-19

2.3.7 WARING MESSAGE

Output current message files | Output all message files
Last day message | Next day message
Delete current message files | Parameter setup

When click “Parameter setup” , window “Info window parameter”will be appearance.

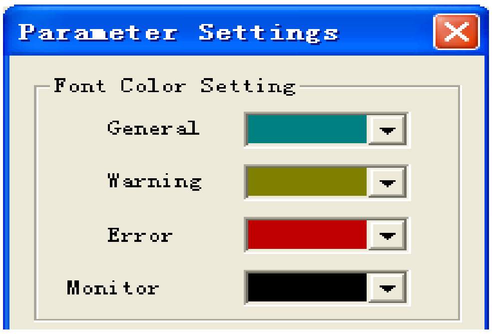


Fig. 2.3-20 Font color setup

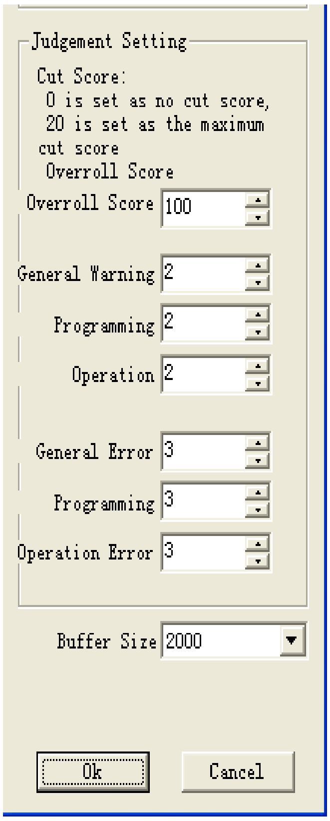


Fig. 2.3-21 Gradeing standard

1. VULGAR WARINGS

Return to reference point!

Backoff measuring piercing point bar of spindle(for milling machine only)!

Program protection is locked out, and it’s unable to edit!

Program protection is locked out, and it’s unable to delete program!

Modality is not booked！Please book first!

Input format: X*** or Y*** or Z*** (FANUC measurement)!

Cutter parameter is incorrect!

There is a tool hasing this tool number, please input new tool number!

No tool hasing this tool number in top rest!

Please backoff measuring piercing point bar before auto-toochange!

Please choose the mode Auto、Edit or DNC before open file!

The file is over the Max size,so it is unable to place workpiece!

2. PROGRAMMING WARING

Search program，no O****!

Program protection is locked out, and it's unable to edit new program number!

3. MACHINE PPERATION WARING

Electric source is not opened or intense electricity is unavailable!

Spindle startup should be in JOG、HND、INC or WHEEL mode!

Please close machine door!

Startup NCSTART，then switch to AUTO、MDI、TEACHING or DNC mode!

4. VULGAR ERRORS

Please backoff spindle measurement piercing point bar before startup NCSTART

X direction overshoot

Y direction overshoot

Z direction overshoot

5. PROGRAMMING ERRORS

General G code and cyclic program are something the matter!

No O*** in program direction!

Cutter number is on-unit!

Radius compensation register number D is on-unit!

Length compensation register number H is on-unit!

Modality O*** is not booked! It can't be deleted!

Vice program number is inexistence in subprogram call!

Vice program number is error in subprogram call!

It is lack of value F in G code!

There is no straightaway leadingin in tool compensation!

There is no straightaway eduction in tool compensation!

6. MACHINE OPERATION ERRORS

Cutter comes up against workbench!

Measuring piercing point bar comes up against workbench!

End face comes up against workpiece!

Cutter comes up against holding fixture!

Spindle is not stared,tool collision!

Measuring piercing point bar comes up against tool!

Cutter collision! Please replace small type measuring piercing point bar or raise spindle!

Teacher sends examination questions to student, and he or she can grade it which student finish and send to teacher by Swan simulation network server. Also teacher can control the machine operation panel of student and tips of error message.

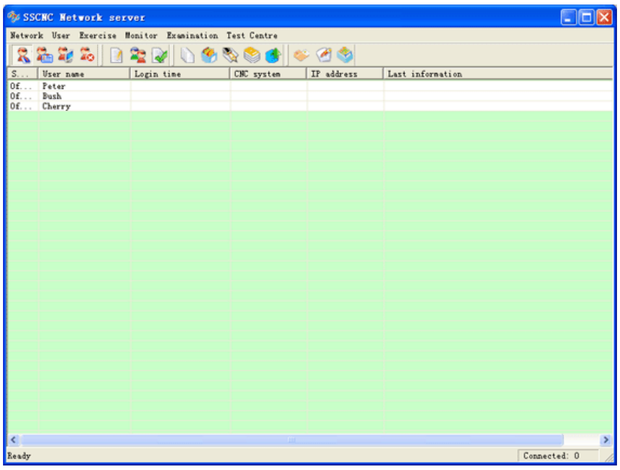


Fig. 2.3-22 Network management

CHAPTER 3 SINUMERIK 802S/c OPERATION

3.1 SINUMERIK 802S/c MACHINE PANEL OPERATION

Machine operation panel is on the bottom-right of window, as the followinggraph show. The panel composed with Choosing botton, Program Running Control Switch and so on is used to control the running status of machine.

Fig 3.1-1 802S/c(milling machine)panel

Fig 3.1-2 802S/c(lathe)panel

AUTO ：

Auto-machining mode

JOG ：

Manual mode, Move mesa or tool manually and continuously.

SINGL :

SPINSTAR :

Spin start.

SPINSTAR :

SPINSTP :

Spin stop.

RESET :

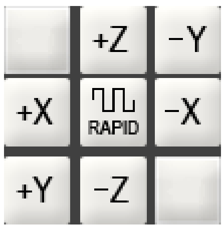
reset.

CYCLESTAR :

Program running startup.

CYCLESTOP :

Program running stop.



MANUAL MOVING MACHINE PANEL BUTTON

FEED-RATE(F) ADJUSTING KNOB

3.2 Operation button

3.2.1 EYSTOKE INTRODUCTION

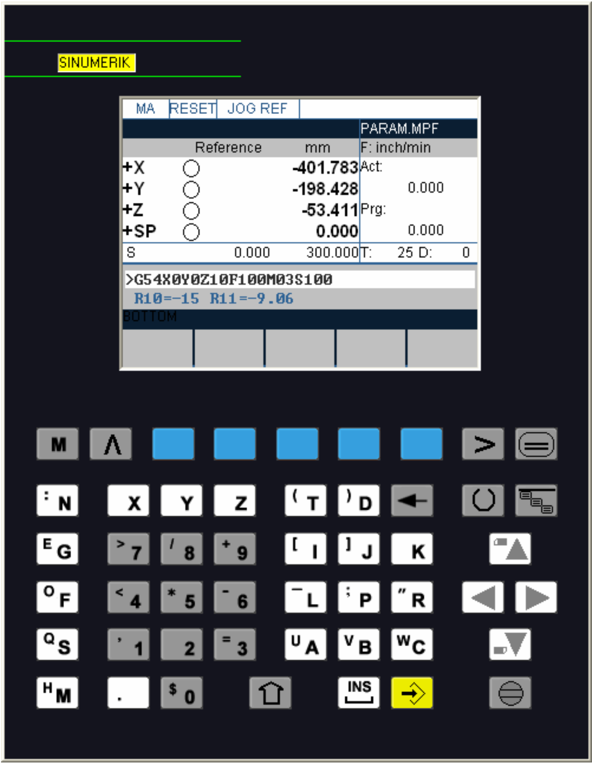
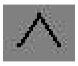
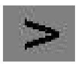


Fig 3.2-1

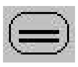
machining show



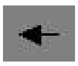
back



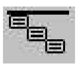
menu extend key



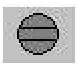
area conversion key



delete



Uprightness menu key

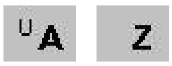


call the police key

select key

enter

Spacebar



letter key

3.2.2 MANUAL OPERATION OF VIRTUAL NC MACHINE

Start

Operating sequence

Connect CNC and machine power then start system and the default window is return the reference point in JOG mode.

Return the reference point (“Machine” operation area**)**

Operating sequence

- “Return the reference point” only can use in mode ZEN.
- Start return the reference point function with press the reference point button in machine control panel
- In return the reference point’s window the reference point’s state of selected axis will be shown.

3.3 NC SYSTEM OPERATION

3.3.1 Parameter Mode

1) Creating a new tool

Operating sequence

- Press “New” softkey to create a new tool.
- Pressing this softkey opens the input window and an overview of the tool numbers assigned.
- Enter the new T number (maximal only three digits) and specify the tool type.
- Press OK to confirm your entry; the Tool Compensation Data window is opened.

2) Tool compensation data

The tool compensation data are divided into length and radius compensation data.

Operating sequence

- Enter the offsets by positioning the cursor on the input field to be modified,
- entering value(s)
- and confirming your entry by pressing Input or a cursor selection.

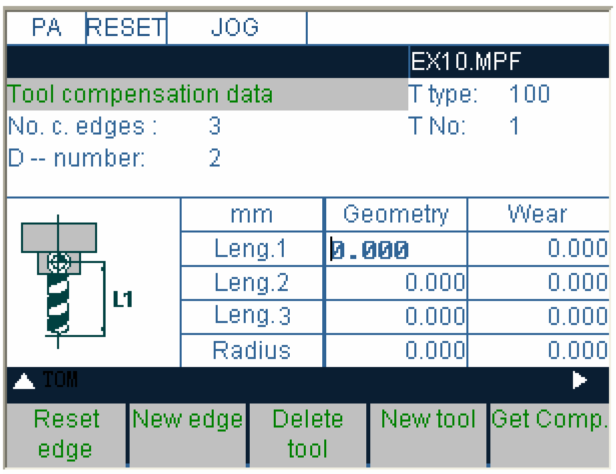


Fig 3.3-1

3) Determining the tool offsets

Operating sequence

- Select the softkey Get Comp. The window Compensation values opens.
- Enter offset if the tool edge cannot approach the zero point Gxx. If you work without zero offset, select G500 and enter offset.
- When the softkey Calculate is pressed, The determined compensation value is stored.

4) Entering/modifying the zero offset ("Parameter" operating area)

Functionality

The actual-value memory and thus also the actual-value display are referred to the machine zero after the reference-point approach. The workpiece machining program, however, refers to the workpiece zero. This offset must be entered as the zero offset.

Operating sequences

- Use the Parameter and Zero Offset softkeys to select the zero offset. An overview of settable zero offsets appears on the screen
- Position the cursor bar on the input field to be altered,
- enter value(s).
- The next zero offset overview is displayed by Page down. G56 and G57 are now displayed.
- Return to next-higher menu level, without saving the zero offset values.

Fig 3.3-2

Determining the zero offset

Prerequisite

You have selected the window with the corresponding zero offset (e.g. G54) and the axis for which you want to determine the offset.

Operating sequences

- Press Parameter softkey
- Softkey can be used to select the zero offsets G54 to G57. The selected zero offset is displayed on the selected softkey.

- Selects the next axis.
- Pressing the Calculate softkey calculates the zero offset.
- Press the OK softkey to quit the window.

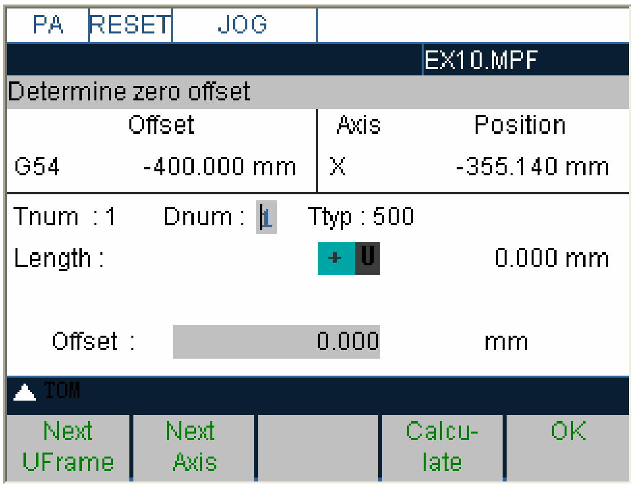


Fig 3.3-3

R parameters ("Parameters" operating area)

Functionality

All R parameters (arithmetic parameters) that exist in the control system are displayed on the R Parameters main screen as a list. These can be modified if necessary.

Operating sequence

- Use the Parameter and R Parameter softkeys
- to position the cursor on the input field that you want to edit.
- Enter value(s).
- Press Input or use the cursor keys to confirm.

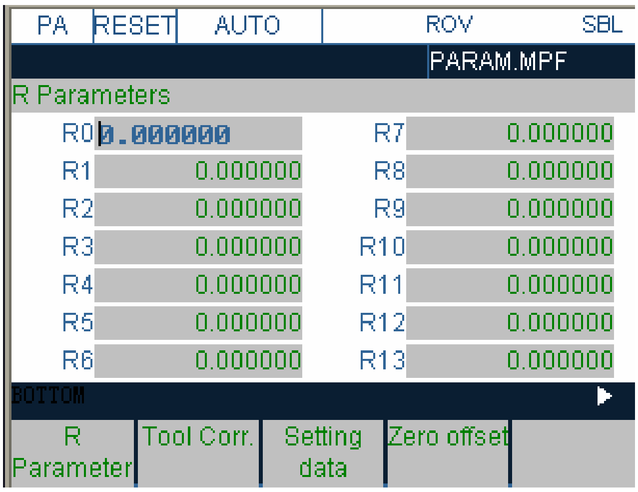


Fig 3.3-4

Programming the setting data ("Parameters" operating area)

Functionality

Use the setting data to define the settings for the operating states. These can also be modified if necessary.

Operating sequences

- Use the Parameter and Setting Data softkeys to select Setting Data.
- The Setting Data softkey branches to another menu level in which various control options can be set.

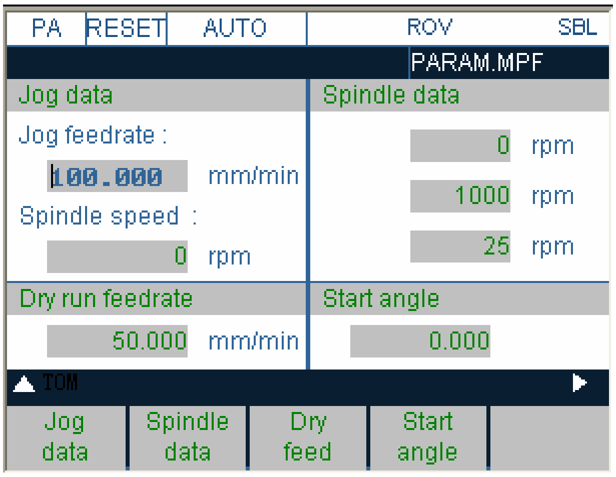


Fig 3.3-5

- Use the paging keys to position the cursor on the desired line within the display areas.
- Enter the new value in the input fields.
- Use Input or the cursor keys to confirm.

Softkeys

JOG data

This function can be used to change the following settings:

Jog feed

Feed value in Jog mode

If the feed value is zero, the control system uses the value stored in the machine data.

Spindle

Spindle speed

Direction of rotation of the spindle

Spindle data

Minimum / Maximum

Limits for the spindle speed set in the Max. (G26)/Min. (G25) fields must be within the limit values specified in the machine data. Programmed (LIMS) Programmable upper speed limitation (LIMS) at constant cutting speed (G96).

Dry feed

Dry-run feedrate for dry-run operation (DRY) The feedrate you enter here is used in the program execution instead of the programmed feed during the Automatic mode when the Dry-Run Feedrate is active

Start angle

Start angle for thread cutting (SF)

A start angle representing the starting position for the spindle is displayed for thread cutting operations. It is possible to cut a multiple thread by altering the angle and repeating the thread cutting operation.

3.3.2 Manually Operated Mode

"JOG" Mode ("Machine" operation area)

Functionality

In Jog mode, you can

- traverse the axes and
- set the traversing speed by means of the override switch, etc.

Operating sequences

- Use the Jog key on the machine control panel area to select the Jog mode.
- Press the appropriate key for the X or Z axis to traverse the desired axis. As long as the direction key is pressed and hold down, the axes traverse continuously at the speed stored in the setting data. If this setting is zero, the value stored in the machine data is used.
- If you press the Rapid Traverse Overlay key at the same time, the selected axis is traversed at rapid traverse speed as long as both keys are pressed down.
- In the Incremental Feed operating mode, you can use the same operating sequence to traverse the axis by settable increments. The set increment is displayed in the display area. Jog must be pressed again to cancel the Incremental Feed.
- The Jog main screen displays position, feed and spindle values, including the feedrate override and spindle override, gear stage status as well as the current tool.

Fig 3.3-6

MDA Mode (Manual Data Input) ("Machine" operating area)

Functionality

You can create and execute a part program block in the MDA mode. Contours that require several blocks (e.g. roundings, chamfers) cannot be executed/programmed.

Operating sequences

- Use the MDA key in the machine control panel area to select the MDA mode.
- Enter a block using the control keyboard.
- The entered block is executed by pressing NC START. The block cannot be executed while machining is taking place.

3.3.3 Automatic Mode

Selecting/starting a part program ("Machine" operating area)

Functionality

The control system and the machine must be set up before the program is started. Please note the safety instructions provided by the machine manufacturer.

Operating sequence

- Use the Automatic key to select the Automatic mode.
- An overview of all programs stored in the control system is displayed.
- Position the cursor bar on the desired program.
- Use the Select softkey to select the program for execution. The selected program name appears in the Program Name screen line.

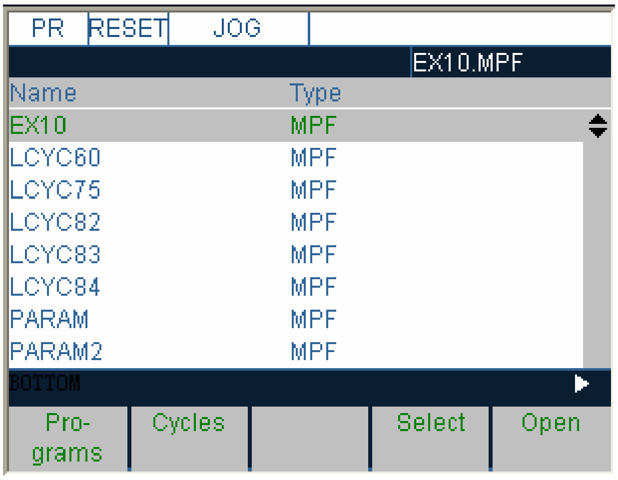


Fig 3.3-7

Automatic Mode

Functionality

In Automatic mode, part programs can be executed fully automatically, i.e. this is the operating mode for standard processing of part programs.

Operating sequence

- Use the Automatic key to select the Automatic mode.

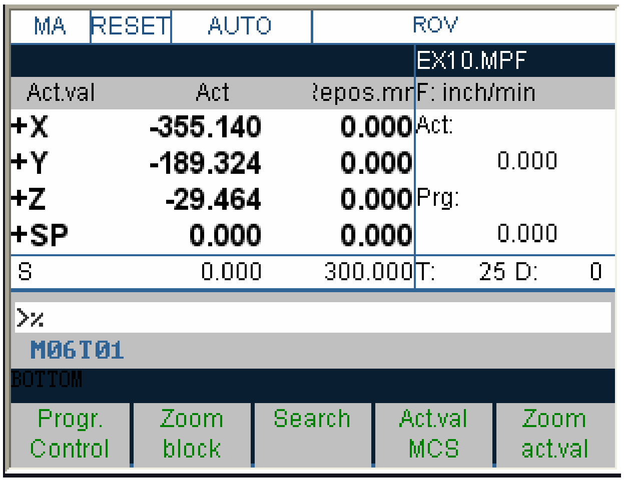


Fig 3.3-8

- An overview of all programs stored in the control system is displayed.

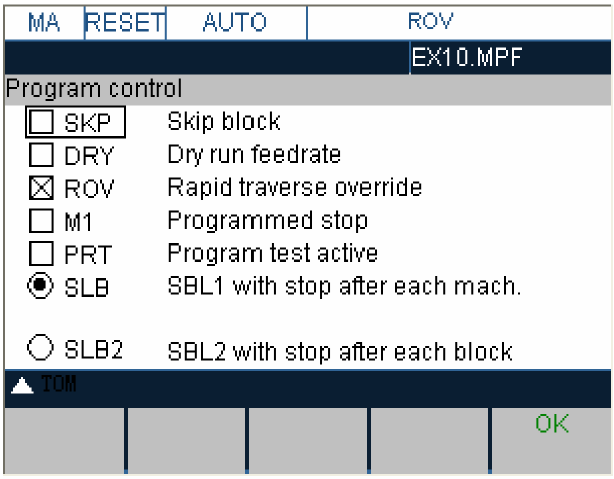


Fig 3.3-9

- Press select/switch key, select program control method
  Select area switch key, return to main menu
- Press program key
- Select program to machined
- Press select key, call the machining program
- Press open key to edit program
- Press single cycle key, select single cycle machining

Fig 3.3-10

Block search ("Machine" operating area)

Operating sequence

- Precondition: The desired program has already been selected,and the control system is in the reset state.
- The block search function can be used to advance the program up to the desired point in the part program. The search target is set by positioning the cursor directly on the desired block in the part program.

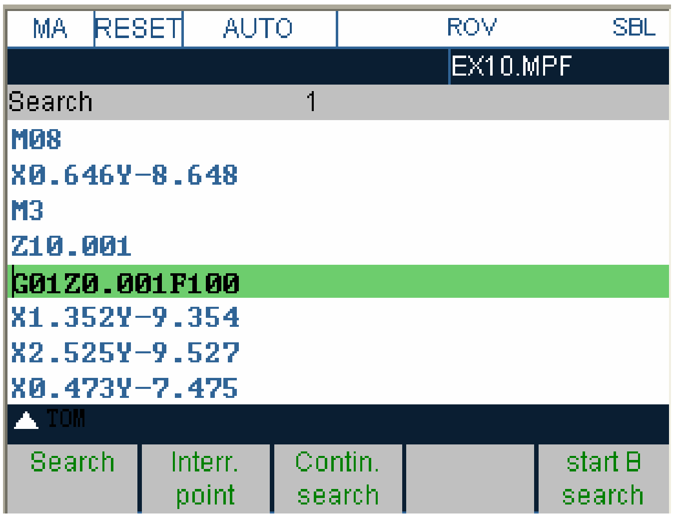


Fig 3.3-11

Start B search

This function starts program advance and closes the Search window.

Result of the search The desired block is displayed in the Current Block window.

3.3.4 Program Mode

Entering a new program ("Program" operating area)

Functionality

This Section describes how to create a new file for a part program. A window appears in which you are prompted to enter program name and type.

Operating sequences

- You have selected the Program operating area. The Program Overview window showing the programs already stored in the CNC is displayed on the screen.
- Press the New softkey. A dialog window appears in which you enter the new main program or subroutine program name. The extension .MPF for main programs is automatically entered. The extension .SPF for subroutines must be entered with the program name.
- Enter the new name.
- Complete your input by selecting the OK softkey. The new part program file is generated and is now ready for editing.
- The creation of the program can be interrupted by RECALL; the window is then closed.

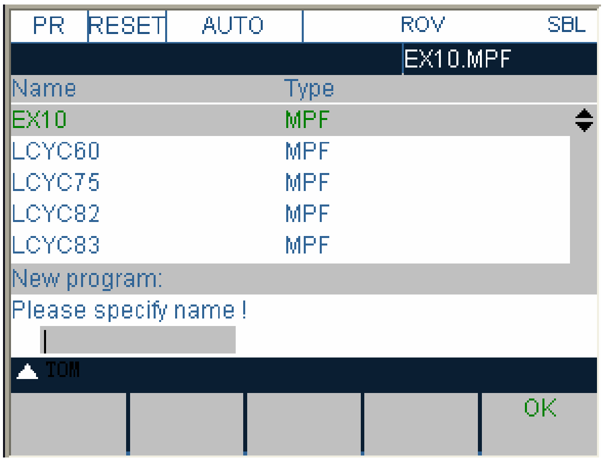


Fig 3.3-12

Editing a part program ("Program" operating area)

Functionality

Part programs or sections of a part program can only be edited if not being executed.

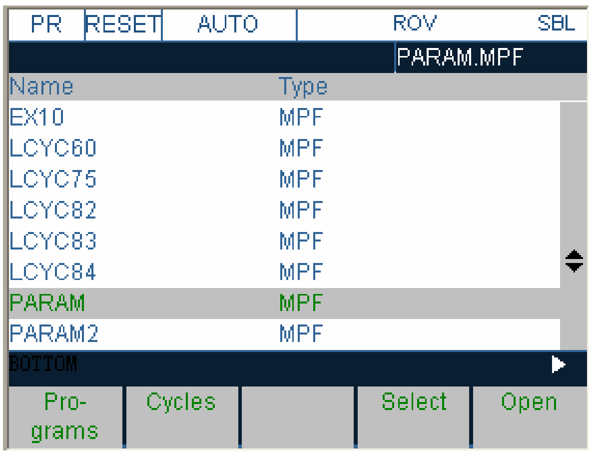


Fig 3.3-13

Operating sequence

- You are in the main menu and have selected the Programs operating area. The program overview appears automatically.
- Use the paging keys to select the program you wish to edit.
- Pressing the open softkey calls the editor for the selected program and pulls down the editor window. The file can now be edited. All changes are stored immediately.
- This function stores the changes in the file system and automatically closes the file.

CHAPTER 4 SINUMERIK 802D OPERATION

4.1 SINUMERIK 802D MACHINE PANEL OPERATION

Machine operation panel is on the bottom-right of window, as the followinggraph show. The panel composed with Choosing botton, Program Running Control Switch and so on is used to control the running status of machine.

Fig 4.1-1 802D (milling machine)panel

Fig 4.1-2 802D (lathe)panel

Fig4.1-3 802D (lathe)panel

MDA :

edit

AUTO ：

Auto-machining mode

JOG ：

Manual mode, Move mesa or tool manually and continuously

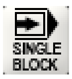
REFPOT :

return reference point.

VAR :

increment select.

SINGL :



single step.

PINSTP :

principal axis stop.

RESET :

diaplasis key.

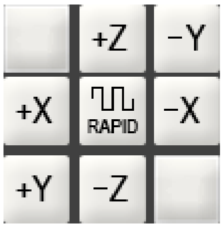
CYCLESTAR :

Program running startup.

CYCLESTOP :

Program running stop

aspect key :



(SIEMENS 802D milling machine)

speediness multiplicator

Urgency stop.

Speed accommodate.

working select

4.2 Operation button

4.2.1 EYSTOKE INTRODUCTION

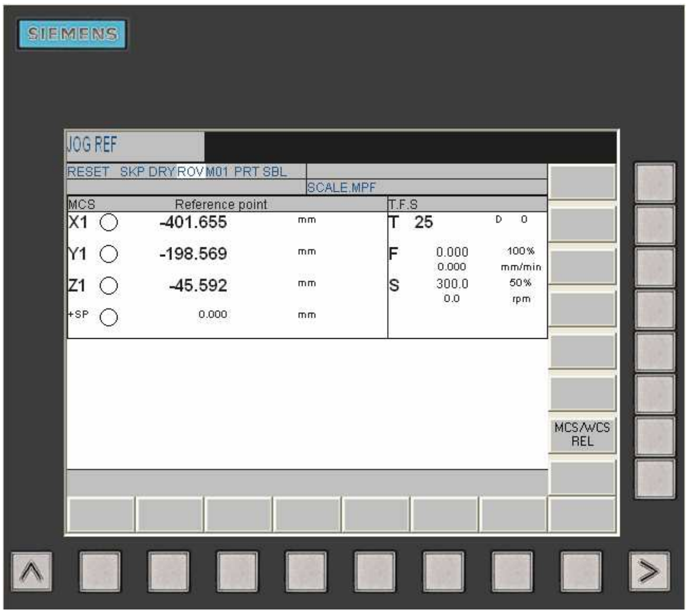
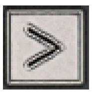


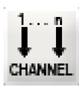
Fig4.2-1

Fig4.2－2



menu enlarge key

call the police key



alleyway conversion key

communication key 

control key

4.2.2 MANUAL OPERATION OF VIRTUAL NC MACHINE

Start

Operating sequence

Connect CNC and machine power then start system and the default window is return the reference point in JOG mode.

Return the reference point (“Machine” operation area**)**

Operating sequence

- “Return the reference point” only can use in mode ZEN.
- Start return the reference point function with press the reference point button in machine control panel
- In return the reference point’s window the reference point’s state of selected axis will be shown.

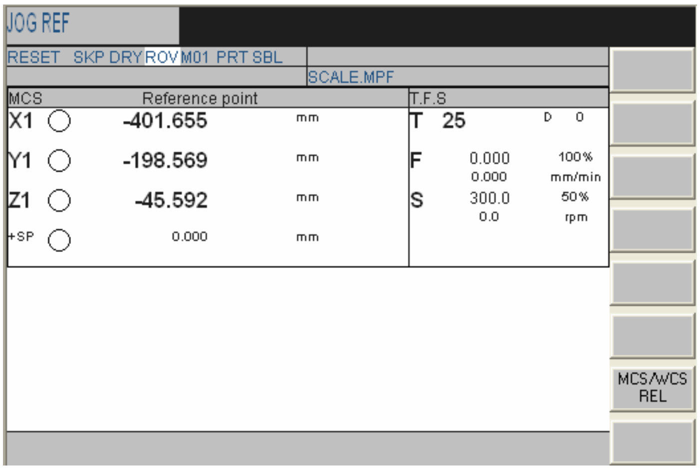


Fig4.2－3

4.3 NC SYSTEM OPERATION

4.3.1 Parameter Mode

1) Creating a new tool

Operating sequence

- Press “New” softkey to create a new tool.
- Pressing this softkey opens the input window and an overview of the tool numbers assigned.
- Enter the new T number (maximal only three digits) and specify the tool type.
- Press OK to confirm your entry; the Tool Compensation Data window is opened.

2) Tool compensation data

The tool compensation data are divided into length and radius compensation data.

Operating sequence

- Enter the offsets by positioning the cursor on the input field to be modified,

- entering value(s)
- and confirming your entry by pressing Input or a cursor selection.

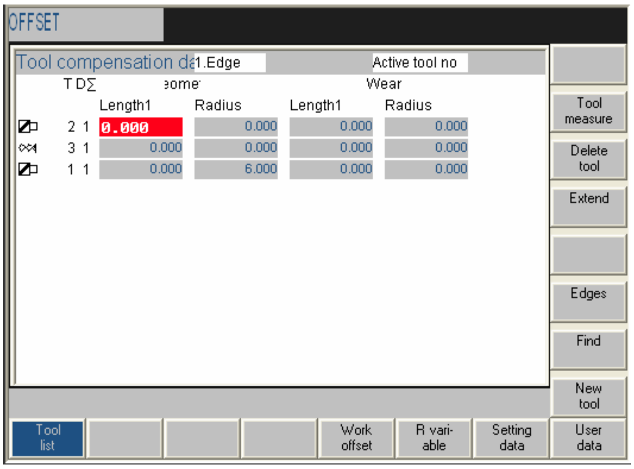


Fig 4.3-1

3) Determining the tool offsets

Operating sequence

- Select the softkey Get Comp. The window Compensation values opens.
- Enter offset if the tool edge cannot approach the zero point Gxx. If you work without zero offset, select G500 and enter offset.
- When the softkey Calculate is pressed, The determined compensation value is stored.

4) Entering/modifying the zero offset ("Parameter" operating area)

Functionality

The actual-value memory and thus also the actual-value display are referred to the machine zero after the reference-point approach. The workpiece machining program, however, refers to the workpiece zero. This offset must be entered as the zero offset.

Operating sequences

- Use the Parameter and Zero Offset softkeys to select the zero offset. An overview of settable zero offsets appears on the screen
- Position the cursor bar on the input field to be altered,
- enter value(s).
- The next zero offset overview is displayed by Page down. G56 and G57 are now displayed.
- Return to next-higher menu level, without saving the zero offset values.

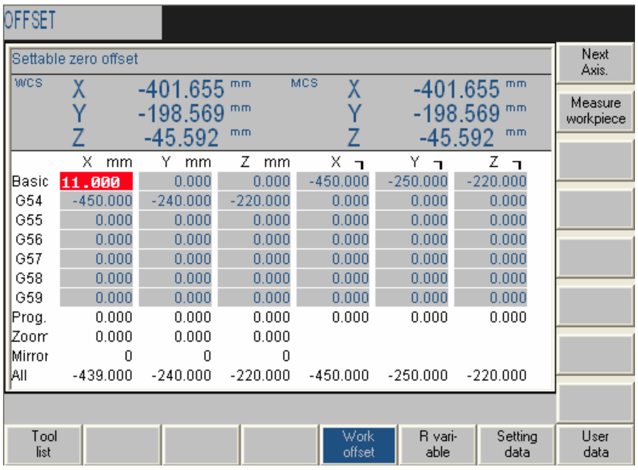


Fig 4.3-2

Determining the zero offset

Prerequisite

You have selected the window with the corresponding zero offset (e.g. G54) and the axis for which you want to determine the offset.

Operating sequences

- Press Parameter softkey
- Softkey can be used to select the zero offsets G54 to G57. The selected zero offset is displayed on the selected softkey.
- Selects the next axis.
- Pressing the Calculate softkey calculates the zero offset.
- Press the OK softkey to quit the window.

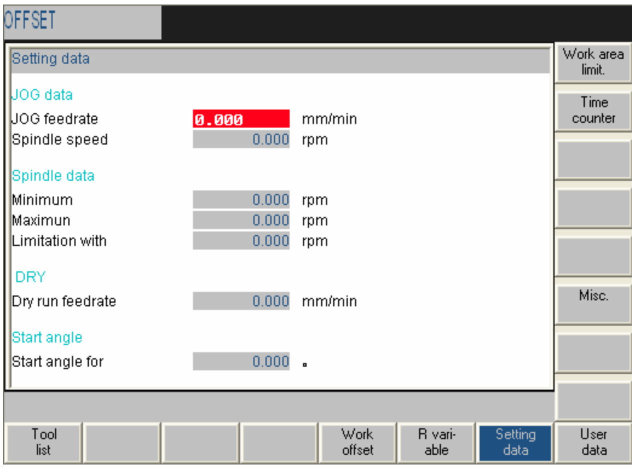


Fig 4.3-3

JOG data

This function can be used to change the following settings:

Jog feed

Feed value in Jog mode

If the feed value is zero, the control system uses the value stored in the machine data.

Spindle

Spindle speed

Direction of rotation of the spindle

Spindle data

Minimum / Maximum

Limits for the spindle speed set in the Max. (G26)/Min. (G25) fields must be within the limit values specified in the machine data. Programmed (LIMS) Programmable upper speed limitation (LIMS) at constant cutting speed (G96).

Dry feed

Dry-run feedrate for dry-run operation (DRY) The feedrate you enter here is used in the program execution instead of the programmed feed during the Automatic mode when the Dry-Run Feedrate is active

Start angle

Start angle for thread cutting (SF)

A start angle representing the starting position for the spindle is displayed for thread cutting operations. It is possible to cut a multiple thread by altering the angle and repeating the thread cutting operation.

4.3.2 Manually Operated Mode

“JOG” Mode (“Machine” operation area)

Functionality

In Jog mode, you can

- traverse the axes and
- set the traversing speed by means of the override switch, etc.

Operating sequences

- Use the Jog key on the machine control panel area to select the Jog mode.
- Press the appropriate key for the X or Z axis to traverse the desired axis. As long as the direction key is pressed and hold down, the axes traverse continuously at the speed stored in the setting data. If this setting is zero, the value stored in the machine data is used.
- If you press the Rapid Traverse Overlay key at the same time, the selected axis is traversed at rapid traverse speed as long as both keys are pressed down.
- In the Incremental Feed operating mode, you can use the same operating sequence to traverse the axis by settable increments. The set increment is displayed in the display area. Jog must be pressed again to cancel the Incremental Feed.
- The Jog main screen displays position, feed and spindle values, including the feedrate override and spindle override, gear stage status as well as the current tool.

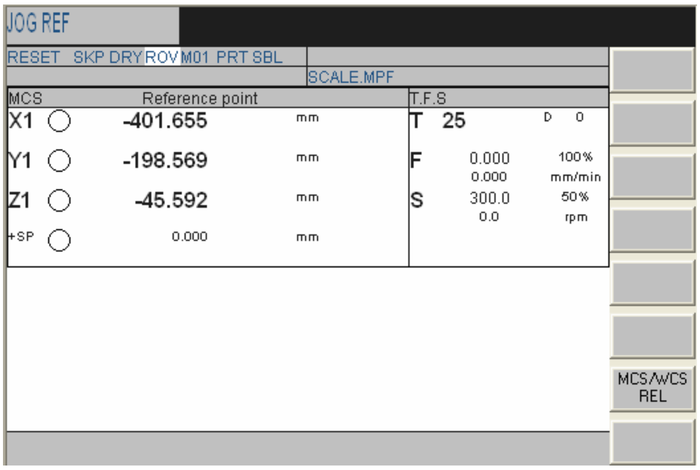


Fig 4.3-4

MDA Mode (Manual Data Input) ("Machine" operating area)

Functionality

You can create and execute a part program block in the MDA mode. Contours that require several blocks (e.g. roundings, chamfers) cannot be executed/programmed.

Operating sequences

- Use the MDA key in the machine control panel area to select the MDA mode.
- Enter a block using the control keyboard.
- The entered block is executed by pressing NC START. The block cannot be executed while machining is taking place.

4.3.3 Automatic Mode

Selecting/starting a part program ("Machine" operating area)

Functionality

The control system and the machine must be set up before the program is started. Please note the safety instructions provided by the machine manufacturer.

Operating sequence

- Use the Automatic key to select the Automatic mode.
- An overview of all programs stored in the control system is displayed.
- Position the cursor bar on the desired program.
- Use the Select softkey to select the program for execution. The selected program name appears in the Program Name screen line.

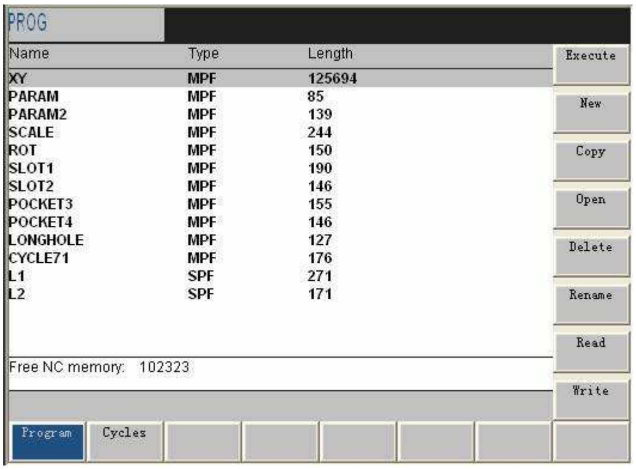


Fig 4.3-5

Automatic Mode

Functionality

In Automatic mode, part programs can be executed fully automatically, i.e. this is the operating mode for standard processing of part programs.

Operating sequence

- Use the Automatic key to select the Automatic mode.

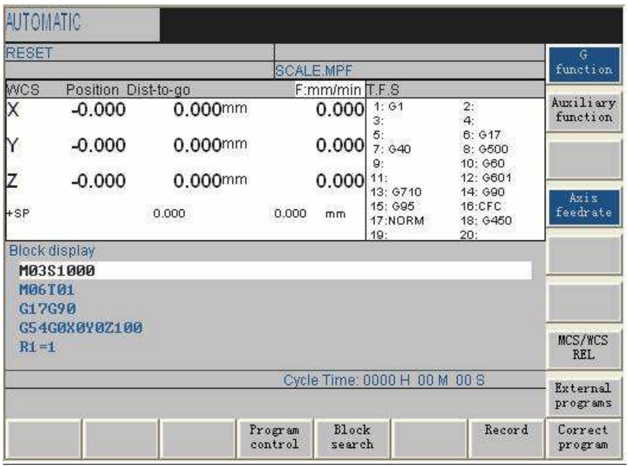


Fig 4.3-6

- An overview of all programs stored in the control system is displayed.

Fig 4.3-7

- Press select/switch key, select program control method

  Select area switch key, return to main menu
- Press program key
- Select program to machined
- Press select key, call the machining program
- Press open key to edit program
- Press single cycle key, select single cycle machining

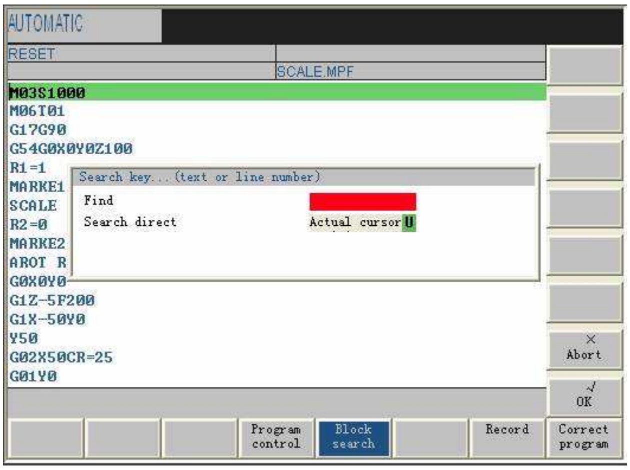


Fig 4.3-8

Block search (“Machine” operating area)

Operating sequence

- Precondition: The desired program has already been selected,and the control system is in the reset state.
- The block search function can be used to advance the program up to the desired point in the part program. The search target is set by positioning the cursor directly on the desired block in the part program.

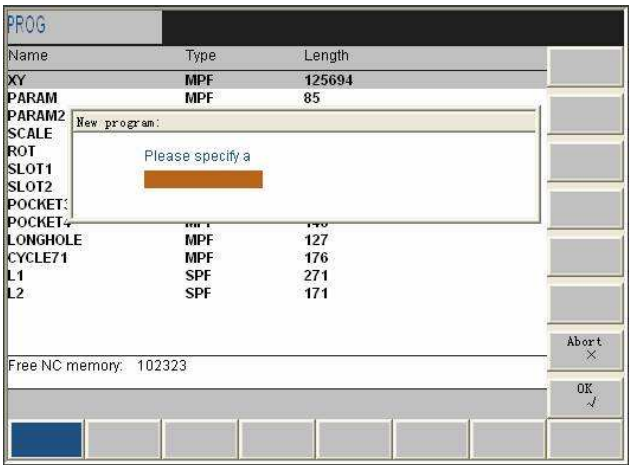


Fig 4.3-9

Start B search

This function starts program advance and closes the Search window.

Result of the search The desired block is displayed in the Current Block window.

4.3.4 Program Mode

Entering a new program ("Program" operating area)

Functionality

This Section describes how to create a new file for a part program. A window appears in which you are prompted to enter program name and type.

Operating sequences

- You have selected the Program operating area. The Program Overview window showing the programs already stored in the CNC is displayed on the screen.
- Press the New softkey. A dialog window appears in which you enter the new main program or subroutine program name. The extension .MPF for main programs is automatically entered. The extension .SPF for subroutines must be entered with the program name.
- Enter the new name.
- Complete your input by selecting the OK softkey. The new part program file is generated and is now ready for editing.
- The creation of the program can be interrupted by RECALL; the window is then closed.

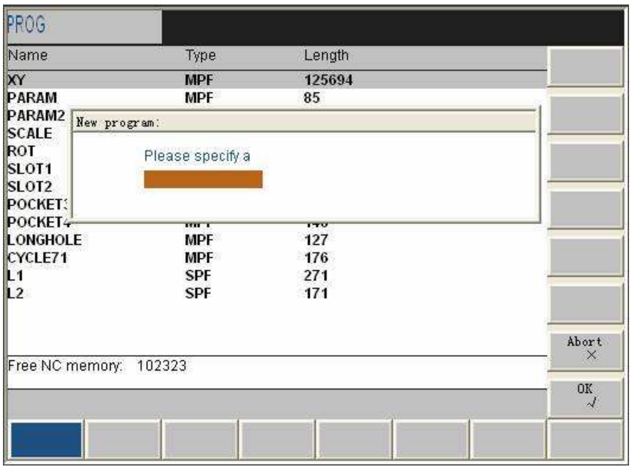


Fig 4.3-10

Editing a part program ("Program" operating area)

Functionality

Part programs or sections of a part program can only be edited if not being executed.

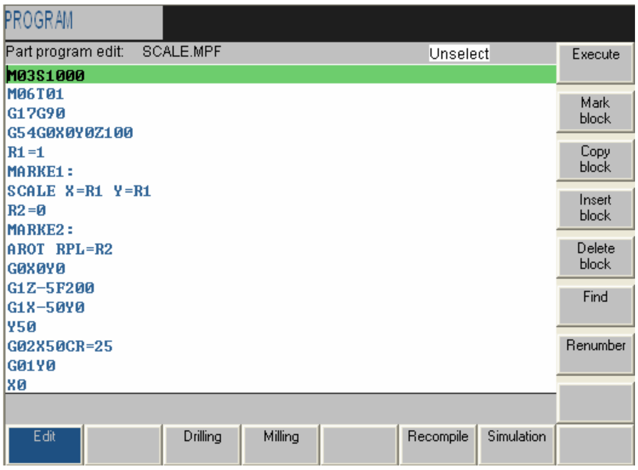


Fig 4.3-11

Operating sequence

- You are in the main menu and have selected the Programs operating area. The program overview appears automatically.
- Use the paging keys to select the program you wish to edit.
- Pressing the open softkey calls the editor for the selected program and pulls down the editor window. The file can now be edited. All changes are stored immediately.
- This function stores the changes in the file system and automatically closes the file.

CHAPTER 5SINUMERIK 810/840 OPERATION

5.1 SINUMERIK 810/840D MACHINE PANEL OPERATION

Machine operation panel is on the bottom-right of window, as the followinggraph show. The panel composed with Choosing botton, Program Running Control Switch and so on is used to control the running status of machine.

Fig5.1－1 810D milling machine panel

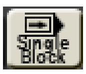
Fig5.1－2 810D lathe panel

automatism machining

Manual mode

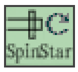
Return to reference point

VAR INCREMENT



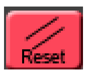
SINGLE BLOCK

SPINDLE START RIGHT Clockwise direction



SPINDLE START RIGHT Clockwise direction

SPINDLE STOP



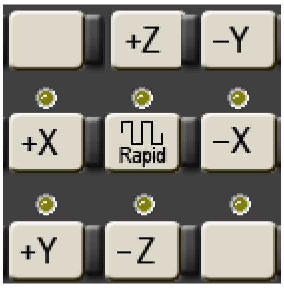
RESET

CYCLE START

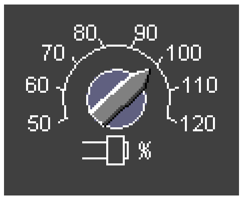
(SINUMERIK 810D lathe)

CYCLE STOP


Emergency stop knob

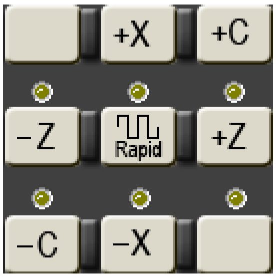


SELECT AXIS



Spindle speed adjusting knob

(SINUMERIK 810D milling machine)



SELECT AXIS

FEEDREAT(F) TUNE BUTTON

5.2 Operation button

5.2.1 EYSTOKE INTRODUCTION

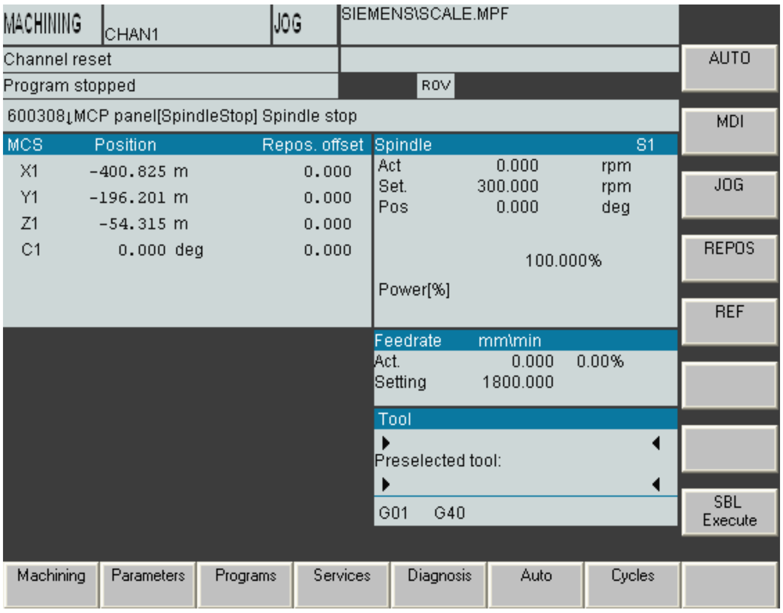


Fig5.2－1

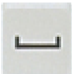
Number/letter key

 back

 ETC key

help

Shift key

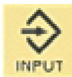
 Replace key

delete key

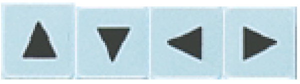
Delete key

insert key

ALT key



enter key

 Cursor DOWN

number key

letter key

select key

5.2.2 MANUAL OPERATION OF VIRTUAL NC MACHINE

Start

Operating sequence

Connect CNC and machine power then start system and the default window is return the reference point in JOG mode.

Return the reference point (“Machine” operation area**)**

Operating sequence

- “Return the reference point” only can use in mode ZEN.
- Start return the reference point function with press the reference point button in machine control panel

- In return the reference point's window the reference point's state of selected axis will be shown.

5.3 NC SYSTEM OPERATION

5.3.1 Manually Operated Mode

"JOG" Mode ("Machine" operation area)

Functionality

In Jog mode, you can

- traverse the axes and
- set the traversing speed by means of the override switch, etc.

Operating sequences

- Use the Jog key on the machine control panel area to select the Jog mode.
- Press the appropriate key for the X or Z axis to traverse the desired axis. As long as the direction key is pressed and hold down, the axes traverse continuously at the speed stored in the setting data. If this setting is zero, the value stored in the machine data is used.
- If you press the Rapid Traverse Overlay key at the same time, the selected axis is traversed at rapid traverse speed as long as both keys are pressed down.
- In the Incremental Feed operating mode, you can use the same operating sequence to traverse the axis by settable increments. The set increment is displayed in the display area. Jog must be pressed again to cancel the Incremental Feed.
- The Jog main screen displays position, feed and spindle values, including the feedrate override and spindle override, gear stage status as well as the current tool.

5.3.2 Parameter Mode

1) Creating a new tool

Operating sequence

- Press "New" softkey to create a new tool.
- Pressing this softkey opens the input window and an overview of the tool numbers assigned.
- Enter the new T number (maximal only three digits) and specify the tool type.
- Press OK to confirm your entry; the Tool Compensation Data window is opened.

2) Tool compensation data

The tool compensation data are divided into length and radius compensation data.

Operating sequence

- Enter the offsets by positioning the cursor on the input field to be modified,
- entering value(s)
- and confirming your entry by pressing Input or a cursor selection.

3) Determining the tool offsets

Operating sequence

- Select the softkey Get Comp. The window Compensation values opens.

- Enter offset if the tool edge cannot approach the zero point Gxx. If you work without zero offset, select G500 and enter offset.
- When the softkey Calculate is pressed, The determined compensation value is stored.

4) Entering/modifying the zero offset (“Parameter” operating area)

Functionality

The actual-value memory and thus also the actual-value display are referred to the machine zero after the reference-point approach. The workpiece machining program, however, refers to the workpiece zero. This offset must be entered as the zero offset.

Operating sequences

- Use the Parameter and Zero Offset softkeys to select the zero offset. An overview of settable zero offsets appears on the screen
- Position the cursor bar on the input field to be altered,
- enter value(s).
- The next zero offset overview is displayed by Page down. G56 and G57 are now displayed.
- Return to next-higher menu level, without saving the zero offset values.

Determining the zero offset

Prerequisite

You have selected the window with the corresponding zero offset (e.g. G54) and the axis for which you want to determine the offset.

Operating sequences

- Press Parameter softkey
- Softkey can be used to select the zero offsets G54 to G57. The selected zero offset is displayed on the selected softkey.
- Selects the next axis.
- Pressing the Calculate softkey calculates the zero offset.
- Press the OK softkey to quit the window.

R parameters (“Parameters” operating area)

Functionality

All R parameters (arithmetic parameters) that exist in the control system are displayed on the R Parameters main screen as a list. These can be modified if necessary.

Operating sequence

- Use the Parameter and R Parameter softkeys
- to position the cursor on the input field that you want to edit.
- Enter value(s).
- Press Input or use the cursor keys to confirm.

Programming the setting data (“Parameters” operating area)

Functionality

Use the setting data to define the settings for the operating states. These can also be modified if necessary.

Operating sequences

- Use the Parameter and Setting Data softkeys to select Setting Data.
- The Setting Data softkey branches to another menu level in which various control options can be set.
- Use the paging keys to position the cursor on the desired line within the display areas.
- Enter the new value in the input fields.
- Use Input or the cursor keys to confirm.

5.3.3 Automatic Mode

Selecting/starting a part program ("Machine" operating area)

Functionality

The control system and the machine must be set up before the program is started. Please note the safety instructions provided by the machine manufacturer.

Operating sequence

- Use the Automatic key to select the Automatic mode.
- An overview of all programs stored in the control system is displayed.
- Position the cursor bar on the desired program.
- Use the Select softkey to select the program for execution. The selected program name appears in the Program Name screen line.

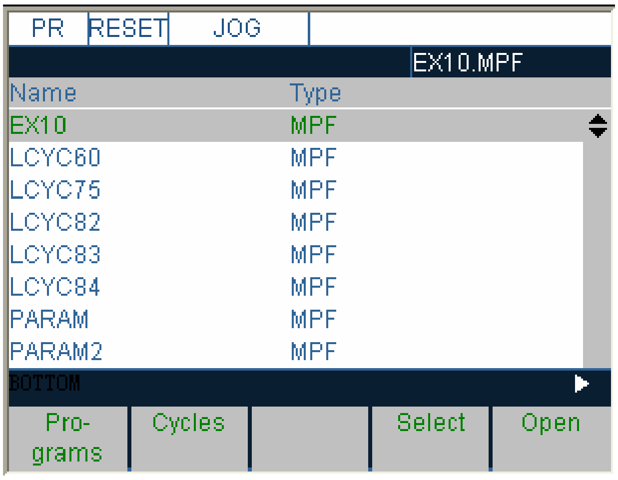


Fig 3.3-7

Automatic Mode

Functionality

In Automatic mode, part programs can be executed fully automatically, i.e. this is the operating mode for standard processing of part programs.

Operating sequence

- Use the Automatic key to select the Automatic mode.

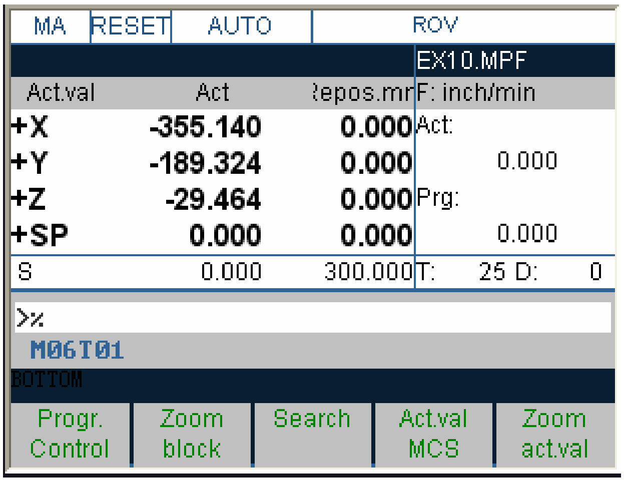


Fig 3.3-8

- An overview of all programs stored in the control system is displayed.
- Press select/switch key, select program control method
  Select area switch key, return to main menu
- Press program key
- Select program to machined
- Press select key, call the machining program
- Press open key to edit program
- Press single cycle key, select single cycle machining

Block search ("Machine" operating area)

Operating sequence

- Precondition: The desired program has already been selected,and the control system is in the reset state.
- The block search function can be used to advance the program up to the desired point in the part program. The search target is set by positioning the cursor directly on the desired block in the part program.

Start B search

This function starts program advance and closes the Search window.

Result of the search The desired block is displayed in the Current Block window.

CHAPTER 6 SINUMERIK 801 OPERATION

6.1 SINUMERIK 801 MACHINE PANEL OPERATION

Machine operation panel is on the bottom-right of window, as the followinggraph show. The panel composed with Choosing botton, Program Running Control Switch and so on is used to control the running status of machine.

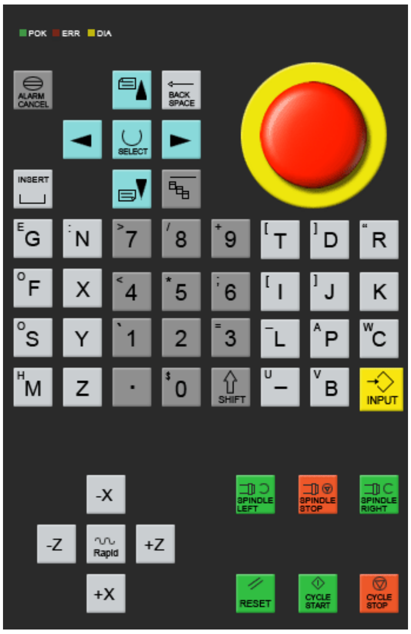


Fig6.1-1

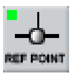
AUTO ：

AUTOMATIC.

JOG ：

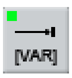
JOG

REFPOT :

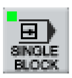


REFERENCE POINT

VAR :



INCREMENT



SINGLE BLOCK

SPINSTAR :

SPINDLE START RIGHT Clockwise direction

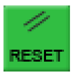
SPINSTAR :

主轴反转.

SPINSTP :

SPINDLE STOP

RESET :



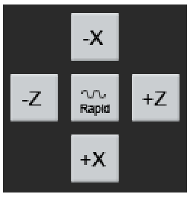
RESET

CYCLESTAR :

CYCLE START

CYCLESTOP :

CYCLE STOP

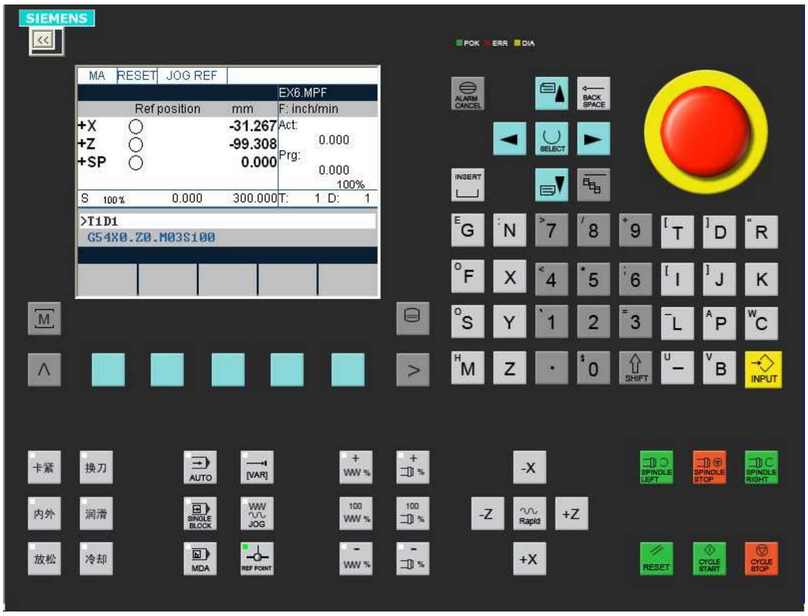


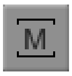
SELECT AXIS  SPINDLE SPEED TUNE BUTTON

6.2 Operation button

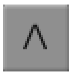
6.2.1 EYSTOKE INTRODUCTION

Machine operation panel is on the bottom-right of window, as the followinggraph show. The panel composed with Choosing botton, Program Running Control Switch and so on is used to control the running status of machine.



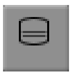


Machine area key

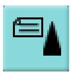


Recall key

ETC key

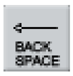


Area switchover key

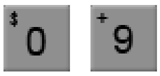


Cursor UP

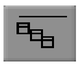
Cursor LEFT



Delete key



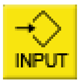
Numerical keys



Vertical menu

Acknowledge alarm

select key

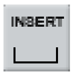


enter key

shift key

Cursor down

Cursor right



insert key

6.2.2 MANUAL OPERATION OF VIRTUAL NC MACHINE

Start

Operating sequence

Connect CNC and machine power then start system and the default window is return the reference point in JOG mode.

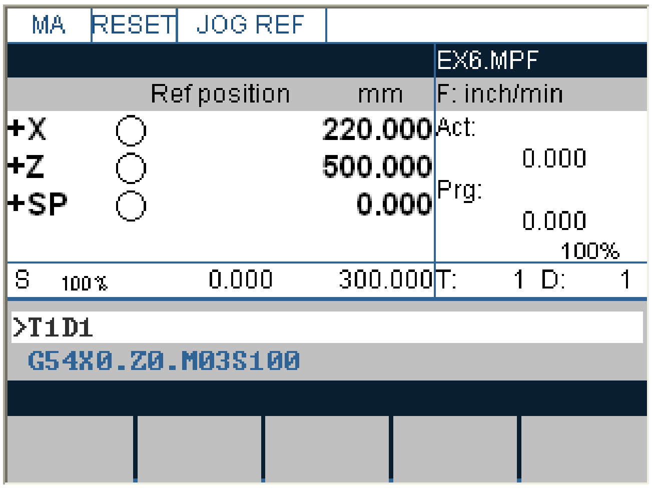


Fig6.2—1

Return the reference point (“Machine” operation area**)**

Operating sequence

- “Return the reference point” only can use in mode ZEN.
- Start return the reference point function with press the reference point button in machine control panel
- In return the reference point’s window the reference point’s state of selected axis will be shown.

6.3 NC SYSTEM OPERATION

6.3.1 Manually Operated Mode

“JOG” Mode (“Machine” operation area)

Functionality

In Jog mode, you can

- traverse the axes and
- set the traversing speed by means of the override switch, etc.

Operating sequences

- Use the Jog key on the machine control panel area to select the Jog mode.
- Press the appropriate key for the X or Z axis to traverse the desired axis. As long as the direction key is pressed and hold down, the axes traverse continuously at the speed stored in the setting data. If this setting is zero, the value stored in the machine data is used.
- If you press the Rapid Traverse Overlay key at the same time, the selected axis is traversed at rapid traverse speed as long as both keys are pressed down.
- In the Incremental Feed operating mode, you can use the same operating sequence to traverse the axis by settable increments. The set increment is displayed in the display area. Jog must be pressed again to cancel the Incremental Feed.
- The Jog main screen displays position, feed and spindle values, including the feedrate override and spindle override, gear stage status as well as the current tool.

MDA Mode (Manual Data Input) (“Machine” operating area)

Functionality

You can create and execute a part program block in the MDA mode. Contours that require several blocks (e.g. roundings, chamfers) cannot be executed/programmed.

Operating sequences

- Use the MDA key in the machine control panel area to select the MDA mode.
- Enter a block using the control keyboard.
- The entered block is executed by pressing NC START. The block cannot be executed while machining is taking place.

6.3.2 Parameter Mode

1) Creating a new tool

Operating sequence

- Press “New” softkey to create a new tool.
- Pressing this softkey opens the input window and an overview of the tool numbers assigned.
- Enter the new T number (maximal only three digits) and specify the tool type.
- Press OK to confirm your entry; the Tool Compensation Data window is opened.

2) Tool compensation data

The tool compensation data are divided into length and radius compensation data.

Operating sequence

- Enter the offsets by positioning the cursor on the input field to be modified,
- entering value(s)
- and confirming your entry by pressing Input or a cursor selection.

3) Determining the tool offsets

Operating sequence

- Select the softkey Get Comp. The window Compensation values opens.
- Enter offset if the tool edge cannot approach the zero point Gxx. If you work without zero offset, select G500 and enter offset.
- When the softkey Calculate is pressed, The determined compensation value is stored.

4) Entering/modifying the zero offset ("Parameter" operating area)

Functionality

The actual-value memory and thus also the actual-value display are referred to the machine zero after the reference-point approach. The workpiece machining program, however, refers to the workpiece zero. This offset must be entered as the zero offset.

Operating sequences

- Use the Parameter and Zero Offset softkeys to select the zero offset. An overview of settable zero offsets appears on the screen
- Position the cursor bar on the input field to be altered,
- enter value(s).
- The next zero offset overview is displayed by Page down. G56 and G57 are now displayed.
- Return to next-higher menu level, without saving the zero offset values.

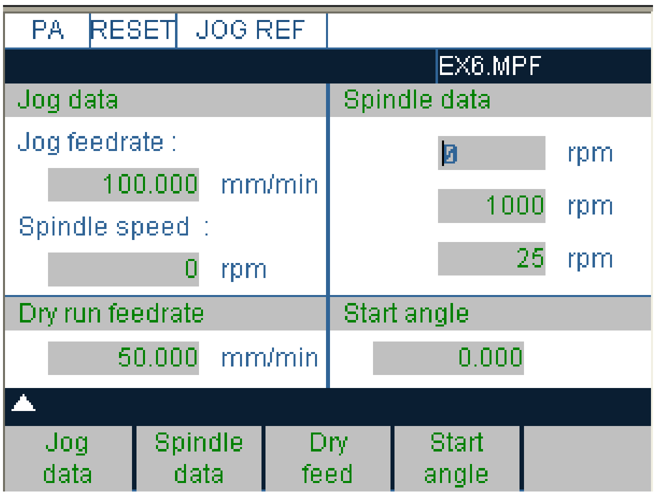


Fig 6.3-1

Determining the zero offset

Prerequisite

You have selected the window with the corresponding zero offset (e.g. G54) and the axis for which you want to determine the offset.

Operating sequences

- Press Parameter softkey
- Softkey can be used to select the zero offsets G54 to G57. The selected zero offset is

displayed on the selected softkey.

- Selects the next axis.
- Pressing the Calculate softkey calculates the zero offset.
- Press the OK softkey to quit the window.

R parameters ("Parameters" operating area)

Functionality

All R parameters (arithmetic parameters) that exist in the control system are displayed on the R Parameters main screen as a list. These can be modified if necessary.

Operating sequence

- Use the Parameter and R Parameter softkeys
- to position the cursor on the input field that you want to edit.
- Enter value(s).
- Press Input or use the cursor keys to confirm.

Programming the setting data ("Parameters" operating area)

Functionality

Use the setting data to define the settings for the operating states. These can also be modified if necessary.

Operating sequences

- Use the Parameter and Setting Data softkeys to select Setting Data.
- The Setting Data softkey branches to another menu level in which various control options can be set.

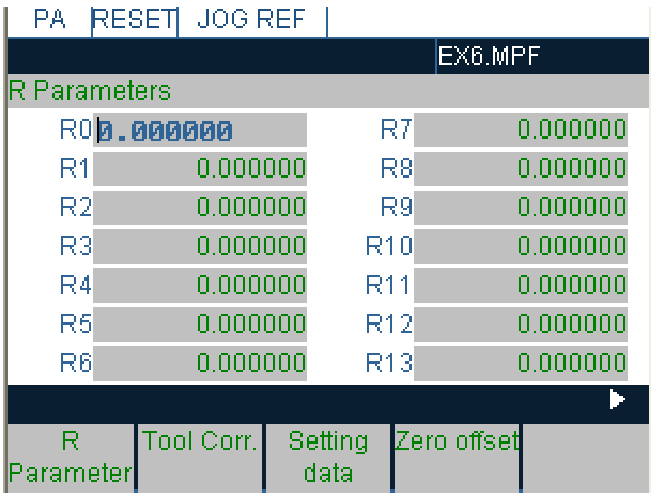


Fig 6.3-2

- Use the paging keys to position the cursor on the desired line within the display areas.
- Enter the new value in the input fields.
- Use Input or the cursor keys to confirm.

Softkeys

JOG data

This function can be used to change the following settings:

Jog feed

Feed value in Jog mode

If the feed value is zero, the control system uses the value stored in the machine data.

Spindle

Spindle speed

Direction of rotation of the spindle

Spindle data

Minimum / Maximum

Limits for the spindle speed set in the Max. (G26)/Min. (G25) fields must be within the limit values specified in the machine data. Programmed (LIMS) Programmable upper speed limitation (LIMS) at constant cutting speed (G96).

Dry feed

Dry-run feedrate for dry-run operation (DRY) The feedrate you enter here is used in the program execution instead of the programmed feed during the Automatic mode when the Dry-Run Feedrate is active

Start angle

Start angle for thread cutting (SF)

A start angle representing the starting position for the spindle is displayed for thread cutting operations. It is possible to cut a multiple thread by altering the angle and repeating the thread cutting operation.

6.3.3 Automatic Mode

Selecting/starting a part program ("Machine" operating area)

Functionality

The control system and the machine must be set up before the program is started. Please note the safety instructions provided by the machine manufacturer.

Operating sequence

- Use the Automatic key to select the Automatic mode.
- An overview of all programs stored in the control system is displayed.
- Position the cursor bar on the desired program.
- Use the Select softkey to select the program for execution. The selected program name appears in the Program Name screen line.

Automatic Mode

Functionality

In Automatic mode, part programs can be executed fully automatically, i.e. this is the operating mode for standard processing of part programs.

Operating sequence

- Use the Automatic key to select the Automatic mode.

- An overview of all programs stored in the control system is displayed.
- Press select/switch key, select program control method
  Select area switch key, return to main menu
- Press program key
- Select program to machined
- Press select key, call the machining program
- Press open key to edit program
- Press single cycle key, select single cycle machining

Block search (“Machine” operating area)

Operating sequence

- Precondition: The desired program has already been selected,and the control system is in the reset state.
- The block search function can be used to advance the program up to the desired point in the part program. The search target is set by positioning the cursor directly on the desired block in the part program.

Start B search

This function starts program advance and closes the Search window.

Result of the search The desired block is displayed in the Current Block window.

6.3.4 Program Mode

Entering a new program (“Program” operating area)

Functionality

This Section describes how to create a new file for a part program. A window appears in which you are prompted to enter program name and type.

Operating sequences

- You have selected the Program operating area. The Program Overview window showing the programs already stored in the CNC is displayed on the screen.
- Press the New softkey. A dialog window appears in which you enter the new main program or subroutine program name. The extension .MPF for main programs is automatically entered. The extension .SPF for subroutines must be entered with the program name.
- Enter the new name.
- Complete your input by selecting the OK softkey. The new part program file is generated and is now ready for editing.
- The creation of the program can be interrupted by RECALL; the window is then closed.

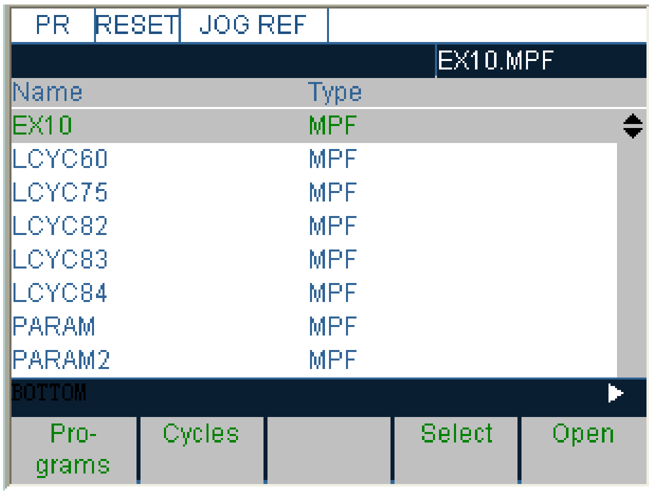


Fig 6.3-3

Editing a part program ("Program" operating area)

Functionality

Part programs or sections of a part program can only be edited if not being executed.

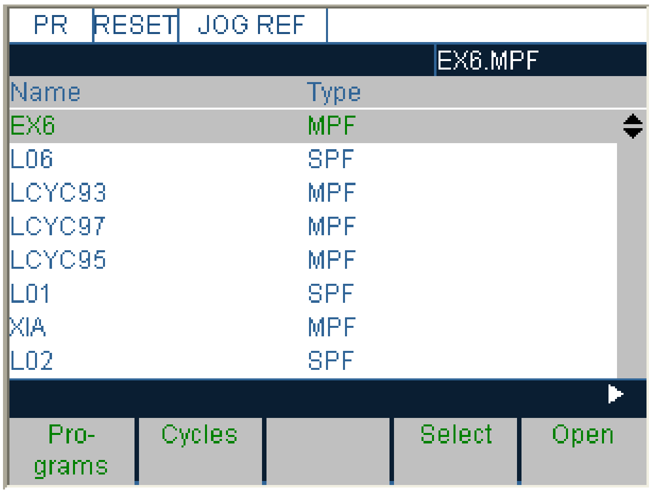


Fig 6.3-4

Operating sequence

- You are in the main menu and have selected the Programs operating area. The program overview appears automatically.
- Use the paging keys to select the program you wish to edit.
- Pressing the open softkey calls the editor for the selected program and pulls down the editor window. The file can now be edited. All changes are stored immediately.
- This function stores the changes in the file system and automatically closes the file.

CHAPTER 7 SINUMERIK 802Se OPERATION

7.1 SINUMERIK 802Se MACHINE PANEL OPERATION

Machine operation panel is on the bottom-right of window, as the followinggraph show. The panel composed with Choosing botton, Program Running Control Switch and so on is used to control the running status of machine.

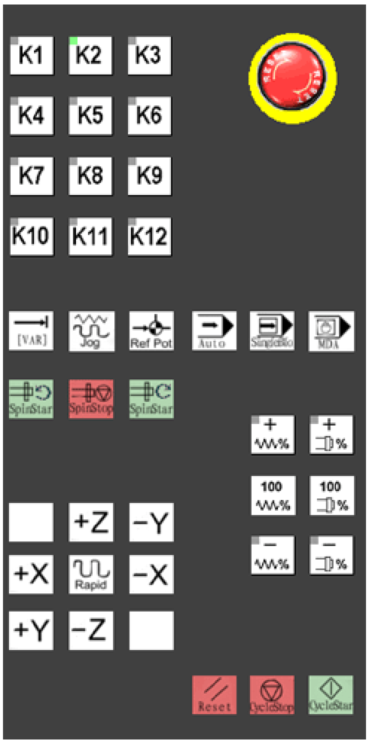


Fig7.1－1802Semilling machinepanel

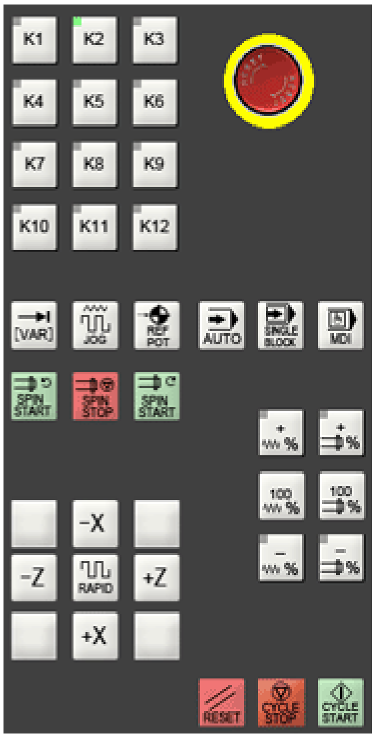


Fig7.1－2 802Selathepanel

AUTO ：

 AUTOMATIC

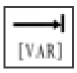
JOG ：

 JOG

REFPOT :

 REFERENCE POINT

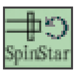
VAR :

 INCREMENT.

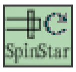
SINGL :

 SINGLE BLOCK.

SPINSTAR :

 SPINDLE START RIGHT Clockwise direction.

SPINSTAR :



SPINDLE START RIGHT Clockwise direction

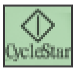
SPINSTP :

SPINDLE STOP

RESET :

RESET.

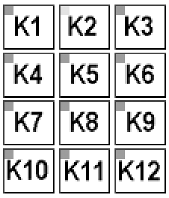
CYCLESTAR :



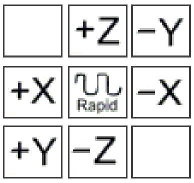
CYCLE START.

CYCLESTOP :

CYCLE STOP.



user-defined key



SELECT AXIS.


EMERGENCY STOP.

7.2 Operation button

Machine operation panel is on the bottom-right of window, as the followinggraph show. The panel composed with Choosing botton, Program Running Control Switch and so on is used to control the running status of machine.

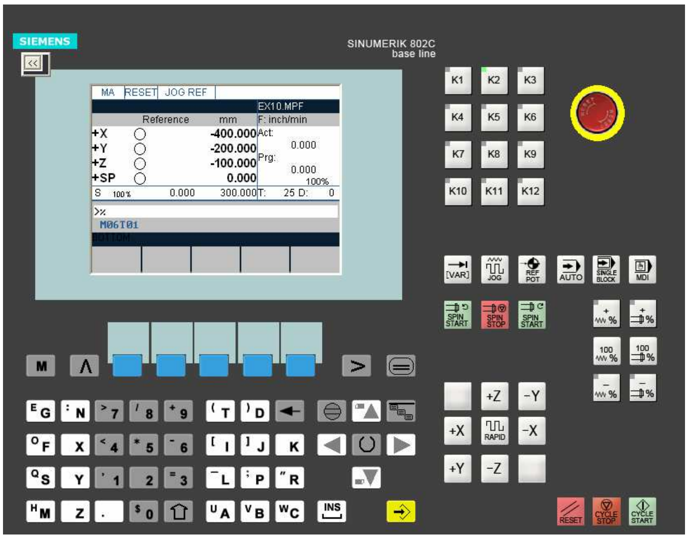


Fig7.2—1

7.2.1 EYSTOKE INTRODUCTION

Machine area key.

Recall key. ETC key Area switchover key

Cursor UP Cursor LEFT Delete key(backspace)

Numerical keys Vertical menu

Acknowledge alarm Selection key/toggle key

ENTER/input key Shift key

Cursor DOWN Cursor RIGHT

SPACE(INSERT) Alphanumeric keys

7.3 NC SYSTEM OPERATION

Start

Operating sequence

Connect CNC and machine power then start system and the default window is return the reference point in JOG mode.

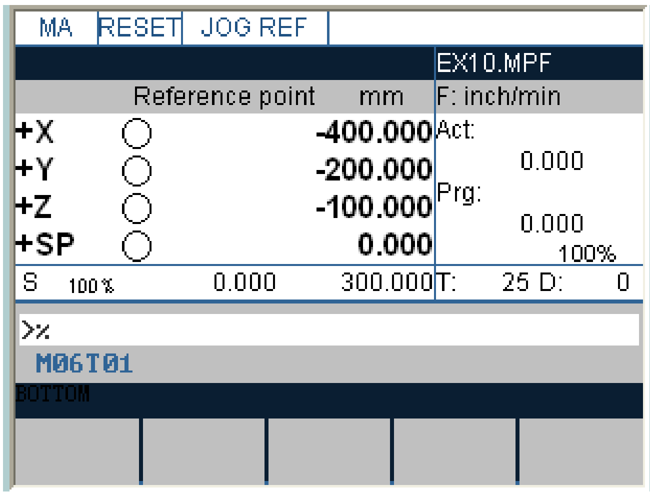


Fig7.3—1

Return the reference point (“Machine” operation area**)**

Operating sequence

- “Return the reference point” only can use in mode ZEN.
- Start return the reference point function with press the reference point button in machine control panel
- In return the reference point’s window the reference point’s state of selected axis will be

shown.

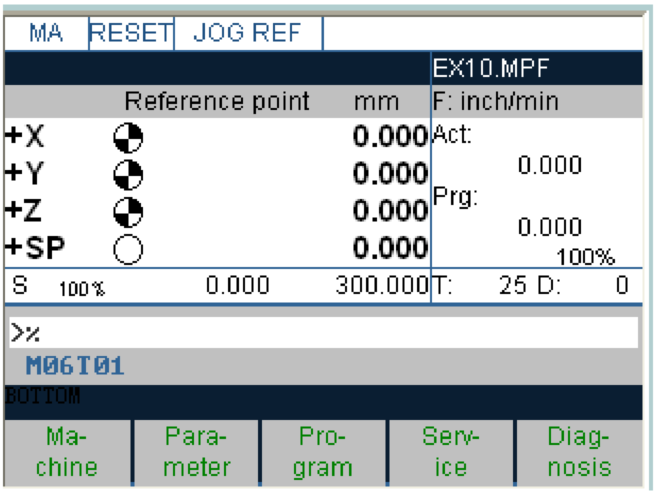


Fig7.3—2

7.3.1 Parameter Mode

1) Creating a new tool

Operating sequence

- Press “New” softkey to create a new tool.
- Pressing this softkey opens the input window and an overview of the tool numbers assigned.
- Enter the new T number (maximal only three digits) and specify the tool type.
- Press OK to confirm your entry; the Tool Compensation Data window is opened.

2) Tool compensation data

The tool compensation data are divided into length and radius compensation data.

Operating sequence

- Enter the offsets by positioning the cursor on the input field to be modified,
- entering value(s)
- and confirming your entry by pressing Input or a cursor selection.

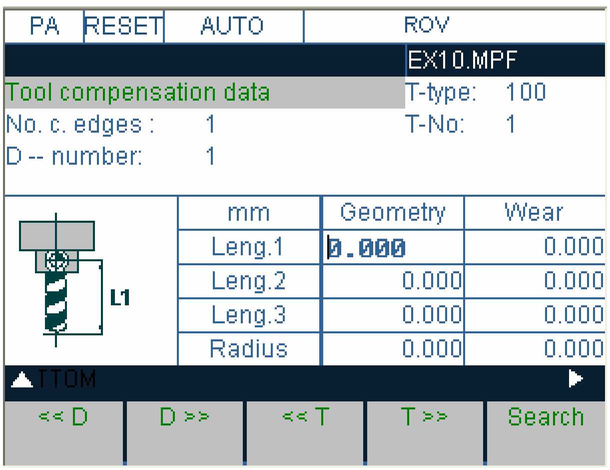


Fig 7.3-3

3) Determining the tool offsets

Operating sequence

- Select the softkey Get Comp. The window Compensation values opens.
- Enter offset if the tool edge cannot approach the zero point Gxx. If you work without zero offset, select G500 and enter offset.
- When the softkey Calculate is pressed, The determined compensation value is stored.

4) Entering/modifying the zero offset ("Parameter" operating area)

Functionality

The actual-value memory and thus also the actual-value display are referred to the machine zero after the reference-point approach. The workpiece machining program, however, refers to the workpiece zero. This offset must be entered as the zero offset.

Operating sequences

- Use the Parameter and Zero Offset softkeys to select the zero offset. An overview of settable zero offsets appears on the screen
- Position the cursor bar on the input field to be altered,
- enter value(s).
- The next zero offset overview is displayed by Page down. G56 and G57 are now displayed.
- Return to next-higher menu level, without saving the zero offset values.

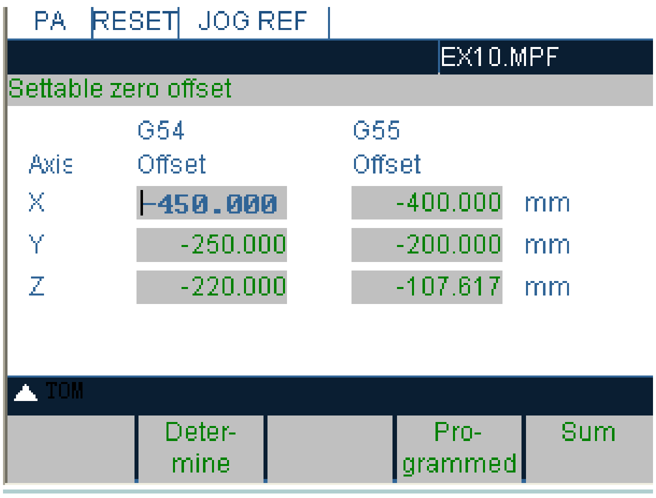


Fig 7.3-4

Determining the zero offset

Prerequisite

You have selected the window with the corresponding zero offset (e.g. G54) and the axis for which you want to determine the offset.

Operating sequences

- Press Parameter softkey
- Softkey can be used to select the zero offsets G54 to G57. The selected zero offset is

displayed on the selected softkey.

- Selects the next axis.
- Pressing the Calculate softkey calculates the zero offset.
- Press the OK softkey to quit the window.

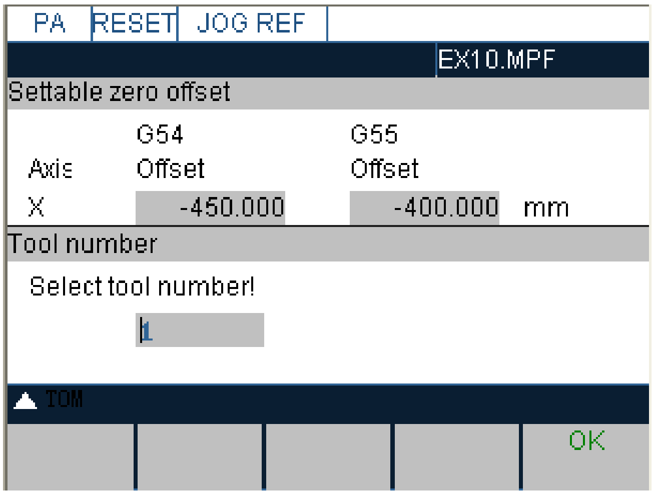


Fig 7.3-5

R parameters ("Parameters" operating area)

Functionality

All R parameters (arithmetic parameters) that exist in the control system are displayed on the R Parameters main screen as a list. These can be modified if necessary.

Operating sequence

- Use the Parameter and R Parameter softkeys
- to position the cursor on the input field that you want to edit.
- Enter value(s).
- Press Input or use the cursor keys to confirm.

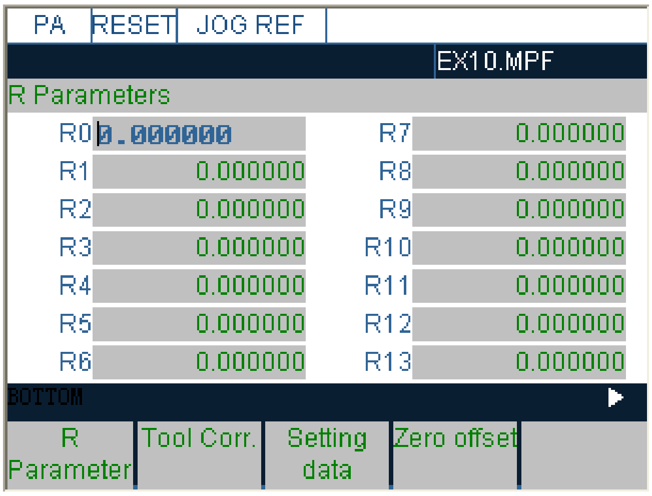


Fig 7.3-6

Programming the setting data ("Parameters" operating area)

Functionality

Use the setting data to define the settings for the operating states. These can also be modified if necessary.

Operating sequences

- Use the Parameter and Setting Data softkeys to select Setting Data.
- The Setting Data softkey branches to another menu level in which various control options can be set.

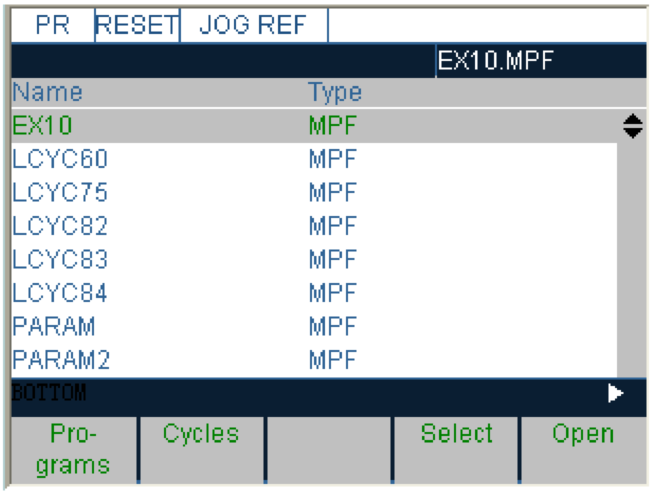


Fig 7.3-7

- Use the paging keys to position the cursor on the desired line within the display areas.
- Enter the new value in the input fields.
- Use Input or the cursor keys to confirm.

Softkeys

JOG data

This function can be used to change the following settings:

Jog feed

Feed value in Jog mode

If the feed value is zero, the control system uses the value stored in the machine data.

Spindle

Spindle speed

Direction of rotation of the spindle

Spindle data

Minimum / Maximum

Limits for the spindle speed set in the Max. (G26)/Min. (G25) fields must be within the limit values specified in the machine data. Programmed (LIMS) Programmable upper speed limitation (LIMS) at constant cutting speed (G96).

Dry feed

Dry-run feedrate for dry-run operation (DRY) The feedrate you enter here is used in the program execution instead of the programmed feed during the Automatic mode when the Dry-Run Feedrate is active

Start angle

Start angle for thread cutting (SF)

A start angle representing the starting position for the spindle is displayed for thread cutting operations. It is possible to cut a multiple thread by altering the angle and repeating the thread cutting operation.

7.3.2 Manually Operated Mode

"JOG" Mode ("Machine" operation area)

Functionality

In Jog mode, you can

- traverse the axes and
- set the traversing speed by means of the override switch, etc.

Operating sequences

- Use the Jog key on the machine control panel area to select the Jog mode.
- Press the appropriate key for the X or Z axis to traverse the desired axis. As long as the direction key is pressed and hold down, the axes traverse continuously at the speed stored in the setting data. If this setting is zero, the value stored in the machine data is used.
- If you press the Rapid Traverse Overlay key at the same time, the selected axis is traversed at rapid traverse speed as long as both keys are pressed down.
- In the Incremental Feed operating mode, you can use the same operating sequence to traverse the axis by settable increments. The set increment is displayed in the display area. Jog must be pressed again to cancel the Incremental Feed.
- The Jog main screen displays position, feed and spindle values, including the feedrate override and spindle override, gear stage status as well as the current tool.

Fig 7.3-8

MDA Mode (Manual Data Input) ("Machine" operating area)

Functionality

You can create and execute a part program block in the MDA mode. Contours that require several blocks (e.g. roundings, chamfers) cannot be executed/programmed.

Operating sequences

- Use the MDA key in the machine control panel area to select the MDA mode.
- Enter a block using the control keyboard.
- The entered block is executed by pressing NC START. The block cannot be executed while machining is taking place.

7.3.3 Automatic Mode

Selecting/starting a part program ("Machine" operating area)

Functionality

The control system and the machine must be set up before the program is started. Please note the safety instructions provided by the machine manufacturer.

Operating sequence

- Use the Automatic key to select the Automatic mode.
- An overview of all programs stored in the control system is displayed.
- Position the cursor bar on the desired program.
- Use the Select softkey to select the program for execution. The selected program name appears in the Program Name screen line.

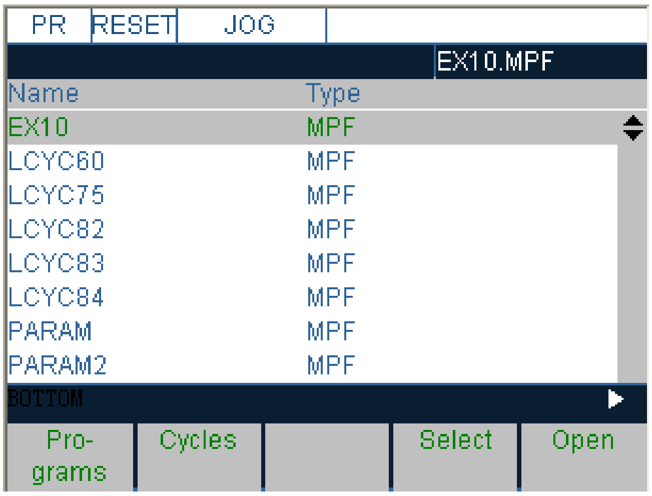


Fig 7.3-9

Automatic Mode

Functionality

In Automatic mode, part programs can be executed fully automatically, i.e. this is the operating mode for standard processing of part programs.

Operating sequence

- Use the Automatic key to select the Automatic mode.

Fig 7.3-10

- An overview of all programs stored in the control system is displayed.
- Press select/switch key, select program control method

  Select area switch key, return to main menu
- Press program key
- Select program to machined
- Press select key, call the machining program
- Press open key to edit program
- Press single cycle key, select single cycle machining

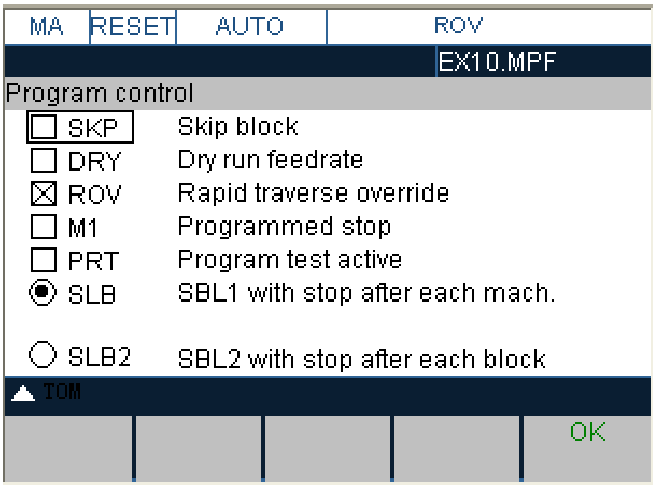


Fig 7.3-11

Block search ("Machine" operating area)

Operating sequence

- Precondition: The desired program has already been selected,and the control system is in the reset state.
- The block search function can be used to advance the program up to the desired point in the part program. The search target is set by positioning the cursor directly on the desired

block in the part program.

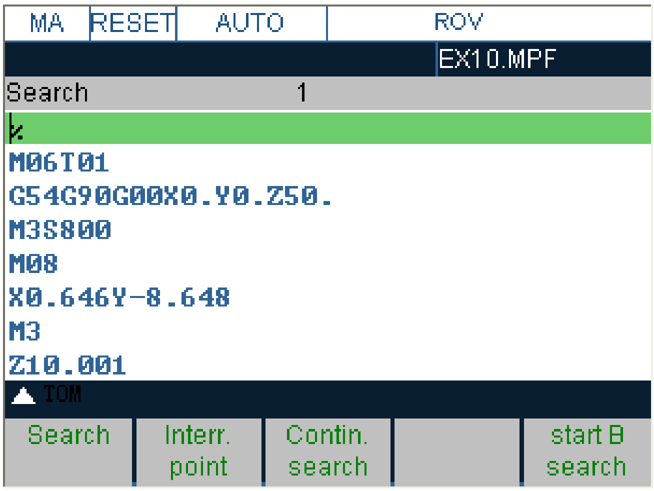


Fig 7.3-12

Start B search

This function starts program advance and closes the Search window.

Result of the search The desired block is displayed in the Current Block window.

7.3.4 Program Mode

Entering a new program ("Program" operating area)

Functionality

This Section describes how to create a new file for a part program. A window appears in which you are prompted to enter program name and type.

Operating sequences

- You have selected the Program operating area. The Program Overview window showing the programs already stored in the CNC is displayed on the screen.
- Press the New softkey. A dialog window appears in which you enter the new main program or subroutine program name. The extension .MPF for main programs is automatically entered. The extension .SPF for subroutines must be entered with the program name.
- Enter the new name.
- Complete your input by selecting the OK softkey. The new part program file is generated and is now ready for editing.
- The creation of the program can be interrupted by RECALL; the window is then closed.

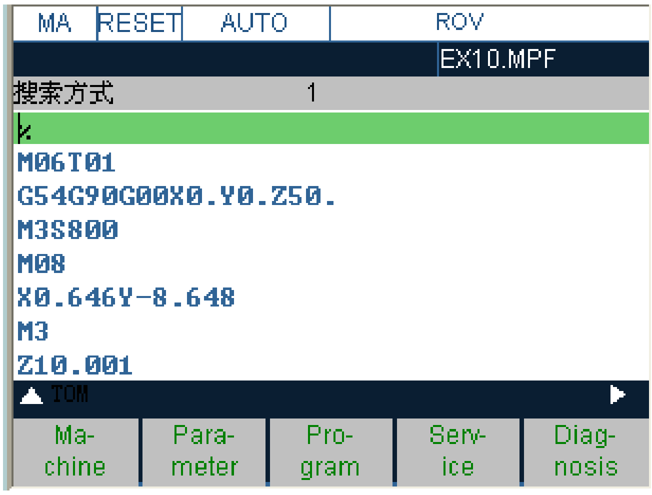


Fig 7.3-13

Editing a part program ("Program" operating area)

Functionality

Part programs or sections of a part program can only be edited if not being executed.

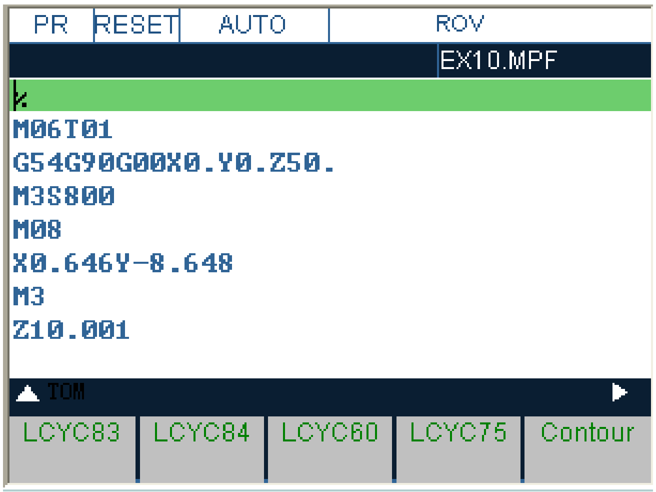


Fig 7.3-14

Operating sequence

- You are in the main menu and have selected the Programs operating area. The program overview appears automatically.
- Use the paging keys to select the program you wish to edit.
- Pressing the open softkey calls the editor for the selected program and pulls down the editor window. The file can now be edited. All changes are stored immediately.
- This function stores the changes in the file system and automatically closes the file.

CHAPTER 8 SINUMERIK 802D programme

8.1 Position

Plane selection: G17 to G19

Functionality

To assign, for example, **tool radius and tool length compensations**, a plane with two axes is selected from the three axes X, Y and Z. In this plane, you can activate a tool radius compensation.

For drill and cutter, the length compensation (length 1) is assigned to the axis standing vertically on the selected plane (see Section 8.6 ”Tool and tool offsets”). It is also possible to use a 3-dimensional length compensation for special cases.

Another influence of plane selection is described with the appropriate functions (e.g. Section 8.5 ”Rounding, chamfer”).

The individual planes are also used to define the **direction of rotation of the circle for the circular interpolation** CW or CCW. In the plane in which the circle is traversed, the abscissa and the ordinate are designed and thus also the direction of rotation of the circle.

Circles can also be traversed in a plane other than that of the currently active G17 to G19 plane (see Chapter 8.3 ”Axis Movements”).

The following plane and axis assignments are possible:

Table 8-2 Plane and axis assignments

G function	Plane (abscissa/ordinate)	vertical axis on plane (length compensation axis when drilling/milling)
G17	X / Y	Z
G18	Z / X	Y
G19	Y / Z	X

Fig8.1－1

Absolute / incremental dimensioning: G90, G91, AC, IC

Functionality

With the instructions G90/G91, the written positional data X, Y, Z, ... are evaluated as a coordinate point (G90) or as an axis position to traverse to (G91). G90/G91 applies to all axes.

Irrespective of G90/G91, certain positional data can be specified for certain blocks in absolute/ incremental dimensions using AC/IC.

These instructions do **not determine the path** by which the end points are reached; this is provided by a G group (G0, G1, G2 and G3... see Chapter 8.3 ”Axis Movements”).

Programming

G90 ; Absolute dimensioning

G91 ; Incremental dimensioning

X=AC(...) ; Absolute dimensioning for a certain axis (here: X axis), non-modal

X=IC(...) ; Absolute dimensioning for a certain axis (here: X axis), non-modal

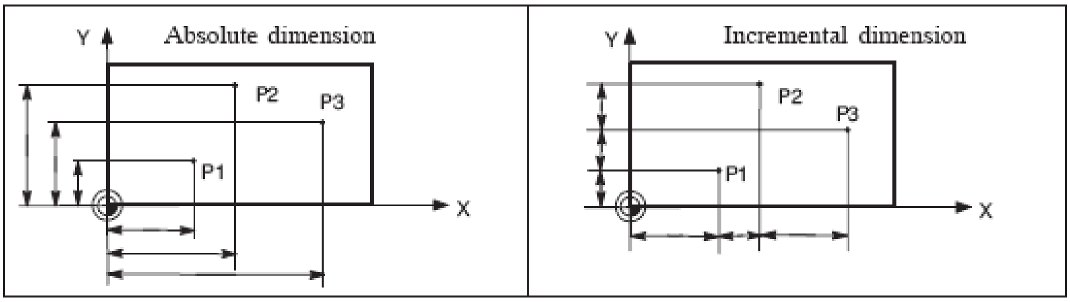


Fig8.1－2

Absolute dimensioning G90

With absolute dimensioning, the dimensioning data refers to the **zero of the coordinate system currently active** (workpiece or current workpiece coordinate system or machine coordinate system). This is dependent on which offsets are currently active: programmable, settable, or no offsets.

Upon program start, G90 is active for **all axes** and remains active until it is deselected in a subsequent block by G91 (incremental dimensioning data) (modally active).

Incremental dimensioning G91

With incremental dimensioning, the numerical value of the path information corresponds to the **axis path to be traversed**. The leading sign indicates the **traversing direction**.

G91 applies to all axes and can be deselected in a subsequent block by G90 (absolute dimensioning).

Specification with =AC(...), =IC(...)

After the end point coordinate, write an equality sign. The value must be specified in round brackets.

Absolute dimensions are also possible for circle center points using =AC(...). Otherwise, the reference point for the circle center is the circle starting point.

Programming example

N10 G90 X20 Z90 ; Absolute dimensioning

N20 X75 Z=IC(–32) ; X dimensioning continues to be absolute, Z incremental dimension

...

N180 G91 X40 Z20 ; Switching to incremental dimensioning

N190 X–12 Z=AC(17) ; X – continues to be incremental dimensioning, Z – absolute

Dimensions in metric units and inches: G71, G70, G710, G700

Functionality

If workpiece dimensions that deviate from the base system settings of the control are present (inch or mm), the dimensions can be entered directly in the program. The required conversion into the base system is performed by the control system.

Programming

G70 ; Inch dimension input

G71 ; Metric dimension data input

G700 ; Inch dimension data input; also for feedrate F

G710 ; Metric dimension data input; also for feedrate F

Programming example

N10 G70 X10 Z30 ; Inch dimension input

N20 X40 Z50 ; G70 continues to be active

...

N80 G71 X19 Z17.3 ; Metric dimesioning from here

Information

Depending on the **default setting** you have chosen, the control system interprets all geometric values as either metric **or** inch dimensions. Tool offsets and settable work offsets including their display are also to be understood as geometrical values; this also applies to the feedrate F in mm/min or inch/min. The default setting can be set via machine data.

All examples listed in this Manual are based on a **metric default setting**.

G70 or G71 evaluates all geometric parameters that directly refer to the **workpiece**, either as inches or metric units, for example:

_ Positional data X, Y, Z, ... for G0,G1,G2,G3,G33, CIP, CT

_ Interpolation parameters I, J, K (also thread pitch)

_ Circle radius CR

_ **Programmable** work offset (TRANS, ATRANS)

_ Polar radius RP

All remaining geometric parameters that are not direct workpiece parameters, such as feedrates, tool offsets, and **settable** work offsets, are not affected by **G70/G71**.

G700/G710 however, also affects the feedrate F (inch/min, inch/rev. or mm/min, mm/rev.).

Polar coordinates, pole definition: G110, G111, G112

Functionality

In addition to the common specification in Cartesian coordinates (X, Y, Z), the points of a workpiece can also be specified using polar coordinates.

Polar coordinates are also helpful if a workpiece or a part of it is dimensioned from a central point (pole) with specification of the radius and the angle.

Plane

The polar coordinates refer to the plane activated with G17 to G19.

In addition, the 3rd axis standing vertically on this plane can be specified. When doing so, spatial specifications can be programmed as cylinder coordinates.

Polar radius RP=...

The polar radius specifies the distance of the point to the pole. It is stored and must only be written in blocks in which it changes, after changing the pole or when switching the plane.

Polar angle AP=...

The angle is always referred to the horizontal axis (abscissa) of the plane (for example, with G17: X axis). Positive or negative angle specifications are possible.

The polar angle remains stored and must only be written in blocks in which it changes, after changing the pole or when switching the plane.

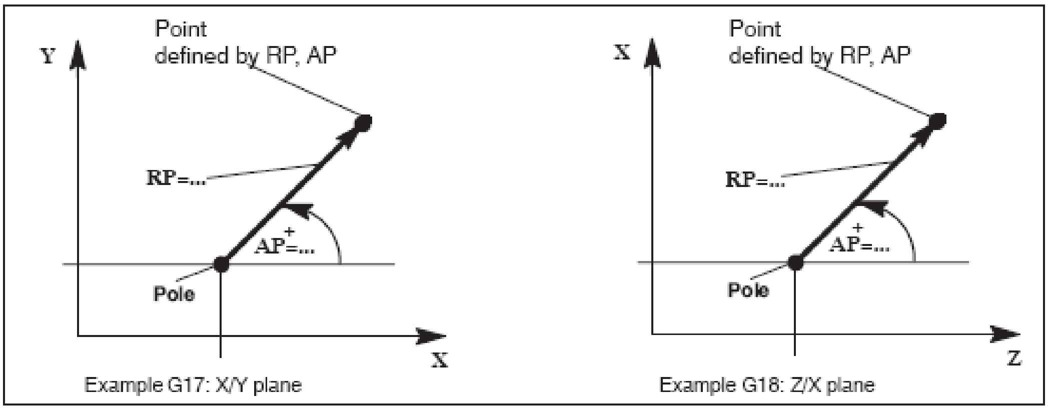


Fig8.1－3

Pole definition, programming

G110 ; Pole specification, relative to the last programmed set position

(in the plane, e.g. G17: X/Y)

G111 ; Pole specification, relative to the origin of the current

workpiece coordinate system (in the plane, e.g. G17: X/Y)

G112 ; Pole specification, relative to the last valid pole; preserve plane

Notes

_ Pole definitions can also be performed using polar coordinates. This makes sense if a pole already exists.

_ If no pole is defined, the origin of the current workpiece coordinate system will act as the pole.

Programming example

N10 G17 ; X/Y plane

N20 G111 X17 Y36 ; Pole coordinates in current workpiece coordinate system

...

N80 G112 AP=45 RP=27.8 ; New pole, relative to the last pole as a polar coordinate

N90 ... AP=12.5 RP=47.679 ; Polar coordinate

N100 ... AP=26.3 RP=7.344 Z4 ; Polar coordinate and Z axis (= cylinder coordinate)

8.2 G Commands

8.2.1 Fundamental Principles of NC Programming

Program names

Each program has its own program name. When creating a program, the program name can be freely selected, observing the following rules:

_ The first two characters must be letters;

_ Use only letters, digits or underscore.

_ Do not use delimiters (see Section ”Character set”).

_ The decimal point must only be used for separation of the file extension.

_ Do not use more than 30 characters.

Example: **FRAME52**

Program structure

Structure and contents

The NC program consists of a sequence of **blocks** (see Table 8-1).

Each block represents a machining step.

Instructions are written in the blocks in the form of **words**.

The last block in the execution sequence contains a special word for the **end of program: M2**.

Table 8-1 NC program structure

Block Word Word Word ... ; Comment

Block N10 G0 X20 ... ; 1. Block

Block N20 G2 Z37 ... ; 2. Block

Block N30 G91 ... ... ; ...

Block N40 ... ...

Block N50 M2 ; End of program

Word structure and address

Functionality/structure

A word is a block element and mainly constitutes a control command. The word consists of

_ **address character**: generally a letter

_ and a **numerical value**: a sequence of digits which with certain addresses can be added by a sign put in front of the address, and a decimal point.

A positive sign (+) can be omitted.

Word

Address Value

Example: G1

Word

Address Value

X –20.1

Word

Address Value

F300

Explanation: Traverse with

Linear

interpolation

Path or lilmit

position for the

X axis: –20.1 mm

Feedrate:

300 mm/min

Figure 8-1 Word structure (example)

Several address characters

A word can also contain several address letters. In this case, however, the numerical value must be assigned via the intermediate character **”=”**.

Example: **CR=5.23**

Additionally, it is also possible to call G functions using a symbolic name (see also Section ”List of instructions”).

Example: **SCALE** ; Enable scaling factor

Extended address

With the addresses

R Arithmetic parameters

H H function

I, J, K Interpolation parameters/intermediate point

the address is extended by 1 to 4 digits to obtain a higher number of addresses. In this case, the value must be assigned using an equality sign ”=” (see also Section ”List of instructions”).

Example: **R10=6.234 H5=12.1 I1=32.67**

Block structure

Functionality

A block should contain all data required to execute a machining step.

Generally, a block consists of several **words** and is always completed with the **end-of-block character** ”**LF**” (Line Feed). This character is automatically generated when pressing the line feed key or the **Input** key.

/N... Word1 Word2 ... Wordn ;Comment LF

End-of-block

character

only if required

is written at the end,

delimited from the

remaining part of the block

by **” ;** ”

Space Space Space Space

Block instructions

Block number – stands in front of instructions;

only if necessary; instead of ”N”, in main blocks,

the following character is used (**” : ”** Colon (:)

Block skip;

only if necessary; stands in the beginning

(BLANK)

Total number of characters in a block: **512** characters

Figure 8-2 Block structure diagram

Word order

If a block contains several instructions, the following order is recommended:

N... G... X... Y... Z... F... S... T... D... M... H...

Note regarding block numbers

First select the block numbers in steps of 5 or 10. Thus, you can later insert

Note regarding block numbers

First select the block numbers in steps of 5 or 10. Thus, you can later insert blocks and nevertheless observe the ascending order of block numbers.

Block skip

Blocks of a program, which are to be executed not with each program run, can be **marked** by a slash **/** in front of the block number. The block skip operation itself is activated either via **operation** (Program control: ”SKP”) or via the PLC (signal). It is also possible to skip a whole program section by skipping several blocks using the ” **/** ”.

If block skip is active during the program execution, all blocks marked with ” **/** ” are skipped. All instructions contained in the blocks concerned will not be considered. The program is continued with the next block without marking.

Comment, remark

The instructions in the blocks of a program can be explained using comments (remarks). A comment is started with the character ” ; ” and ends with the end–of–block chracter. Comments are displayed in the current block display, together with the remaining contents of the block.

Messages

Messages are programmed in a separate block. A message is displayed in a special field and remains active until a block with a new message is executed or until the end of the program is reached. Max. **65** characters of a text message can be displayed.

A message without message text will delete any previous message.

MSG (”THIS IS THE MESSAGE TEXT”)

Programming example

```
N10 ;G&S company, order no. 12A71
N20 ;Pump part 17, drawing no.: 123 677
N30 ;Program created by H. Adam, Dept. TV 4
N40 MSG(”BLANK ROUGHING”)
:50 G17 G54 G94 F470 S20 D2 M3 ;Main block
N60 G0 G90 X100 Y200
N70 G1 Y185.6
N80 X112
/N90 X118 Y180 ;Block can be skipped
N100 X118 Y120
N110 G0 G90 X200
N120 M2 ;End of program
```

Character set

The following characters are used for programming; they are interpreted in accordance with the relevant definitions.

Letters, digits

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

0, 1, 2, 3, 4, 5, 6, 7, 8, 9

No distinction is made between upper and lower case letters.

Printable special characters

(Round left bracket ” Inverted commas

) Round right bracket _ Underscore (belonging to letter)

[Square left bracket . Decimal point

] Square right bracket , Comma, delimiter

< Less than ; Start of comment

> Greater than % Reserved; do not use

: Main block, end of label & Reserved; do not use

= Assignment; subset of euqality ’ Reserved; do not use

/ Division; block skip $ System-internal variable identifier

* Multiplication ? Reserved; do not use

+ Addition; plus sign ! Reserved; do not use

– Subtraction; minus sign

Non–printable special characters

LF Line Feed (end-of-block character)

Blank Delimiter between words; blank

Tabulator Reserved; do not use

Overview of the instructions

Address	Meaning	Value assignment	Information	Programming
D	Tool offset number	0 ... 9, only integer, no sign	Contains offset data for a certain tool T... ; D0–> offset values= 0, max. 9 D numbers per tool	D...
F	Feedrate	0.001 ... 99 999.999	Path velocity of a tool/workpiece; unit: mm/min or mm/revolution depending on G94 or G95	F...
F	Dwell time in block with G4	0.001 ... 99 999.999	Dwell time in seconds	G4 F... ;separate block
G	G function (preparatory function)	Only integer, specified values	The G functions are divided into G groups. Only one G function of a group can be programmed in a block. A G function can be either modal (until it is canceled by another function of the same group) or only effective for the block in which it is programmed non-modal. **G group:**	G... or symbolic name, e.g.: CIP
G0	Linear interpolation at rapid traverse rate		1: Motion commands	G0 X... Y... Z... ; Cartesian using polar coordinates: G0 AP=... RP=... or with additional axis: G0 AP=... RP=... Z... ; e.g.: with G17, Z axis
G1 *	Linear interpolation at feedrate		(type of interpolation)	G1 X... Y... Z... F... With polar coordinates: G1 AP=... RP=... F... or with additional axis: G1 AP=... RP=... Z... F... ; e.g.: with G17, Z axis
G2	Circular interpolation CW (in conjunction with a 3rd axis and TURN=... also helix interpolation –> see also TURN)		modally effective	G2 X... Y... I... J... F... ; Center and end points G2 X... Y... CR=... F... ; Radius and end point G2 AR=... I... J... F... ; Aperture angle and center point G2 AR=... X... Y... F... ; Aperture angle and end point in polar coordinates: G2 AP=... RP=... F... or with additional axis: G2 AP=... RP=... Z... F... ; e.g.: with G17, Z axis
G3	Circular interpolation CCW (in conjunction with a 3rd axis and TURN=... also helix interpolation –> see also TURN)			G3 .... ;otherwise, as with G2
CIP	Circular interpolation via intermediate point			CIP X... Y... Z... I1=... J1=... K1=... F...
CT	Circular interpolation; tangential transition			N10 ... N20 CT X... Y... F... ;Circle, tangential transition to the previous path segment
G33	Thread cutting, tapping with constant lead			S... M... ;spindle speed, direction G33 Z... K... ;Tapping **with** compensation chuck, e.g. in the Z axis
G331	Thread interpolation			N10 SPOS=... ;Position-controlled spindle N20 G331 Z... K... S... ;Tapping **without** compensation chuck, e.g. in the Z axis ;RH or LH thread is specified via the arithmetic sign of the lead (e.g. K+): + : as with M3 – : as with M4
G332	Thread interpolation – retraction			G332 Z... K... ;rigid tapping (**without** compensation chuck, e.g. along the Z axis, **retraction motion** ; Sign of the lead as with G331
G4	Dwell time		2: Special motions non-modal	G4 F... ;separate block, F: Time in seconds or G4 S.... ;separate block, S: in spindle revolutions
G63	Tapping with compensation chuck			G63 Z... F... S... M...
G74	Reference point approach			G74 X1=0 Y1=0 Z1=0 ;separate block (machine axis identifier!)
G75	Fixed-point approach			G75 X1=0 Y1=0 Z1=0 ;separate block (machine axis identifier!)
G147	Smooth approach and retraction along a straight line			G147 G41 DISR=... DISCL=... FAD=... F... X... Y... Z...
G148	Smooth approach and retraction along a straight line			G148 G40 DISR=... DISCL=... FAD=... F... X... Y... Z...
G247	Smooth approach and retraction with a quarter			G247 G41 DISR=... DISCL=... FAD=... F... X... Y... Z...
G248	Smooth approach and retraction with a quarter			G248 G40 DISR=... DISCL=... FAD=... F... X... Y... Z...
G347	Smooth approach and retraction with a semicircle			G347 G41 DISR=... DISCL=... FAD=... F... X... Y... Z...
G348	Smooth approach and retraction with a semicircle			G348 G40 DISR=... DISCL=... FAD=... F... X... Y... Z...
TRANS	Programmable offset		3: Write memory	TRANS X... Y... Z... ;separate block
ROT	programmable rotation		non-modal	ROT RPL=... ;rotation in the current plane G17 ... G19, separate block
SCALE	Programmable scaling factor			SCALE X... Y... Z... ;scaling factor in the direction of the specified axis, separate block

MIRROR	Programmable mirroring		MIRROR X0	;Coordinate axis whose direction is changed; separate block
ATRANS	Additive programmable offset		ATRANS X... Y... Z...	;Separate block
AROT	Additive programmable rotation		AROT RPL=...	;Add. rotation in the current plane G17 ... G19, separate block
ASCALE	Additive programmable scaling factor		ASCALE X... Y... Z...	; Scaling factor in the direction of the specified axis, separate block
AMIRROR	additive programmable mirroring		AMIRROR X0	;Coordinate axis whose direction is changed; separate block
G25	Lower spindle speed limitation or lower working area limitation		G25 S... G25 X... Y ... Z...	;Separate block ;Separate block
G26	Upper spindle speed limitation or upper working area limitation		G26 S... G26 X... Y ... Z...	;Separate block ;Separate block
G110	Pole specification, relative to the last programmed set position		G110 X... Y... G110 RP=... AP=...	;Pole specification, Cartesian, e.g.: With G17 ;pole specification, polar separate block
G111	Pole specification, relative to the origin of the current workpiece coordinate system		G111 X... Y... G111 RP=... AP=...	;Pole specification, Cartesian, e.g.: With G17 ;pole specification, polar separate block
G112	Pole specification, relative to the POLElast valid		G112 X... Y... G112 RP=... AP=...	;Pole specification, Cartesian, e.g.: With G17 ;pole specification, polar separate block
G17 *	X/Y plane	6: Plane selection	G17	;Vertical axis on this
G18	Z/X plane	modally effective		plane is tool length
G19	Y/Z plane			offset axis
G40 *	Tool radius compensation OFF	7: Tool radius compensation		
G41	Tool radius compensation left of the contour	modally effective		
G42	Tool radius compensation right of the contour			

G500 *	Settable work offset OFF	8: Settable work offset	
G54	1st settable work offset	modally effective	
G55	2nd settable work offset		
G56	3rd settable work offset		
G57	4th settable work offset		
G58	5th settable work offset		
G59	6th settable work offset		
G53	Non-modal skipping of the settable work offset	9: Skipping of the settable work offset non-modal	
G153	Non-modal skipping of the settable work offset including base frame		
G60 *	Exact stop	10: Approach behavior	
G64	Continuous-path control mode	modally effective	
G9	Non-modal exact stop	11: Non-modal exact stop non-modal	
G601 *	Exact stop window, fine, with G60, G9	12: Exact stop window	
G602	Exact stop window, coarse, with G60, G9	modally effective	
G70	Inch dimension input	13: Inch / metr.dimension input	
G71 *	Metric dimension data input	modally effective	
G700	Inch dimension data input; also for feedrate F		
G710	Metric dimension data input; also for feedrate F		
G90 *	Absolute dimension data input	14: Absolute / incremental dimension	
G91	Incremental dimension data input	modally effective	
G94 *	Feed F in mm/min	15: Feedrate / spindle	
G95	Feedrate F in mm/spindle revolutions	modally effective	
CFC *	Feedrate with circle ON	16: Feedrate override	
CFTCP	Feedrate override OFF	modally effective	
G450 *	Transition circle	18: Behavior at corners when working with tool radius compensation	
G451	Point of intersection	modally effective	
BRISK *	Jerking path acceleration	21: Acceleration profile	
SOFT	Jerk-limited path acceleration	modally effective	

Address	Meaning	Value Assignment	Information	Programming
H H0= through H9999=	H function	± 0.0000001 ... 9999 9999 (8 decimals) or with specification of an exponent: ± (10^{-300} ... 10^{+300})	Value transfer to the PLC; meaning defined by the machine manufacturer	H0=... H9999=... e. g.: H7=23.456
I	Interpolation parameters	±0.001 ... 99 999.999 Thread: ±0.001 ... 2000.000	Belongs to the X axis; meaning dependent on G2,G3 –> circle center or G33, G331, G332 –> thread lead	See G2, G3, G33, G331 and G332
J	Interpolation parameters	±0.001 ... 99 999.999 Thread: ±0.001 ... 2000.000	Belongs to the Y axis; otherwise, as with I	See G2, G3, G33, G331 and G332
K	Interpolation parameters	±0.001 ... 99 999.999 Thread: ±0.001 ... 2000.000	Belongs to the Z axis; otherwise, as with I	See G2, G3, G33, G331 and G332
I1=	Intermediate point for circular interpolation	±0.001 ... 99 999.999	Belongs to the X axis; specification for circular interpolation with CIP	See CIP
J1=	Intermediate point for circular interpolation	±0.001 ... 99 999.999	Belongs to the Y axis; specification for circular interpolation with CIP	See CIP
K1=	Intermediate point for circular interpolation	±0.001 ... 99 999.999	Belongs to the Z axis; specification for circular interpolation with CIP	See CIP
L	Subroutine; name and call	7 decimals; integer only, no sign	It is also possible to use L1 ...L9999999, Instead of a free name; thus, the subroutine will be called in a separate block. Please observe: L0001 is not always equal to L1. The name "LL6" is reserved for the tool change subroutine.	L781 ;separate block
M	Miscellaneous function	0 ... 99 integer only, no sign	For example, for initiating switching actions, such as "Coolant ON"; max. 5 M functions per block	M...
M0	Programmed stop		The machining is stopped at the end of a block containing M0; to continue, press NC START.	
M1	Optional stop		As with M0, but the stop is only performed if a special signal (Program control: "M01") is present.	
M2	End of program		Can be found in the last block of the processing sequence	
M30	–		Reserved; do not use	
M17	–		Reserved; do not use	
M3	Spindle CW rotation			
M4	Spindle CCW rotation			

Address	Meaning	Value Assignment	Information	Programming
POT( )	Square			R12=POT(R13)
ABS( )	Amount			R8=ABS(R9)
TRUNC( )	Integer portion			R10=TRUNC(R11)
LN( )	Natural logarithm			R12=LN(R9)
EXP( )	Exponential function			R13=EXP(R1)
RET	End of subroutine		Used instead of M2 – to maintain the continuous-path control mode	RET ;separate block
S	Spindle speed	0.001 ... 99 999.999	Unit of measurement of the spindle r.p.m.	S...
S	Dwell time in block with G4	0.001 ... 99 999.999	Dwell time in spindle revolutions	G4 S... ;separate block
T	Tool number	1 ... 32 000 integer only, no sign	The tool change can be performed either directly using the T command or only with M6. This can be set in the machine data.	T...
X	Axis	±0.001 ... 99 999.999	G command	X...
Y	Axis	±0.001 ... 99 999.999	G command	Y...
Z	Axis	±0.001 ... 99 999.999	G command	Z...
AC	Absolute coordinate	–	The dimension can be specified for the end or center point of a certain axis, irrespective of G91.	N10 G91 X10 Z=AC(20) ;X – incremental dimension, Z – absolute
ACC[*axis*]	Percentage path acceleration override	1 ... 200, integer	Acceleration override for an axis or spindle; specified as a percentage	N10 ACC[X]=80 ;for the X axis: 80% N20 ACC[S]=50 ;for the spindle: 50%
ACP	Absolute coordinate; approach position in the positive direction (for rotary axis, spindle)	–	It is also possible to specify the dimensions for the end point of a rotary axis with ACP(...) irrespective of G90/G91; also applies to spindle positioning	N10 A=ACP(45.3) ;Approach absolute position of the A axis in the positive direction N20 SPOS=ACP(33.1) ;Position spindle
ACN	Absolute coordinate; approach position in the negative direction (for rotary axis, spindle)	–	It is also possible to specify the dimensions for the end point of a rotary axis with ACN(...) irrespective of G90/G91; also applies to spindle positioning	N10 A=ACN(45.3) ;Approach absolute position of the A axis in the negative direction N20 SPOS=ACP(33.1) ;Position spindle
ANG	Angle for the specification of a straight line for the contour definition	±0.00001 ... 359.99999	Specified in degrees; one possibility of specifying a straight line when using G0 or G1 if only one end-point coordinate of the plane is known or if the complete end point is known with contour ranging over several blocks	N10 G1 G17 X... Y.... N11 X... **ANG=...** or contour over several blocks: N10 G1 G17 X... Y.... N11 **ANG=...** N12 X... Y... **ANG=...**
AP	Polar angle	0 ... ±359.99999	Specified in degrees, traversing in polar coordinates, definition of the pole; in addition: RP – polar radius	see G0, G1, G2, G3 G110, G111, G112

Address	Meaning	Value Assignment	Information	Programming
SLOT2	Milling a circumferential slot			N10 SLOT2(...) ;separate block
POCKET3	Square pocket			N10 POCKET3(...) ;separate block
POCKET4	Circular pocket			N10 POCKET4(...) ;separate block
CYCLE71	Face milling			N10 CYCLE71(...) ;separate block
CYCLE72	Contour milling			N10 CYCLE72(...) ;separate block
LONG-HOLE	Long hole			N10 LONGHOLE(...) ;separate block
DC	Absolute coordinate; approach position directly (for rotary axis, spindle)	–	It is also possible to specify the dimensions for the end point of a rotary axis with DC(...) irrespective of G90/G91; also applies to spindle positioning	N10 A=DC(45.3) ;Approach absolute position of the A axis directly N20 SPOS=DC(33.1) ; Position spindle
DEF	Definition instruction		Defining a local user variable of the type BOOL, CHAR, INT, REAL, STRING[n], directly at the beginning of the program	DEF INT VARI1=24, VARI2 ; 2 variables of the type INT ; the name is defined by the user DEF STRING[12] VARS3="HELLO" ;max. 12 characters
DISCL	Approach / retraction distance of the infeed movement to the machining plane (SAR)	–	Safety clearance for switching the speed for the infeed movement; please observe: G340, G341	See with G147, G148 , G247, G248 , G347, G348
DISR	Approach/retraction distance or approach/retraction radius (SAR)	–	G147/G148: Distance of the cutter edge from the starting or end point of the contour G247, G347/G248, G348: Radius of the tool center point path	See with G147, G148 , G247, G248 , G347, G348
FAD	Infeed speed (SAR)	–	The speed acts after reaching the safety clearance during infeed. Please observe: G340, G341	See with G147, G148 , G247, G248 , G347, G348
FRC	Non-modal feedrate for chamfer/rounding	0, >0	In case FRC=0: Feedrate Fwill act	For the unit, see F and G94, G95; for chamfer/rounding, see CHF, CHR, RND
FRCM	Modal feedrate for chamfer/rounding	0, >0	In case FRCM=0: Feedrate Fwill act	For the unit, see F and G94, G95; for rounding/modal rounding, see RND, RNDM
FXS [*axis*]	Travel to fixed stop	=1: Selection =0: Deselection	*Axis:* Use the machine identifier	N20 G1 X10 Z25 FXS[Z1]=1 FXST[Z1]=12.3 FXSW[Z1]=2 F...
FXST [*axis*]	Clamping torque, travel to fixed stop	> 0.0 ... 100.0	in %, max. 100% from the max. torque of the drive, *axis:* Use the machine identifier	N30 FXST[Z1]=12.3
FXSW [*axis*]	Monitoring window, travel to fixed stop	> 0.0	Unit of measurement mm or degrees, axis-specific, *axis:* Use the machine identifier	N40 FXSW[Z1]=2.4
GOTOB	GoBack instruction	–	A GoTo operation is performed to a block marked by a label; the jump destination is in the direction of the pro-	N10 LABEL1: ... ...

Address	Meaning	Value Assignment	Information	Programming
OFFN	Groove width with TRACYL, otherwise specification of stock allowance	–	Only effective with the tool radius compensation G41, G42 active	N10 OFFN=12.4
RND	Rounding	0.010 ... 99 999.999	Inserts a rounding with the specified radius value tangentially between two contour blocks; special feedrate FRC= ... possible	N10 X... Y.... **RND=4.5** N11 X... Y...
RNDM	Modal rounding	0.010 ... 99 999.999 0	– Inserts roundings with the specified radius value tangentially at the following contour corners; special feedrate possible: FRCM= ... – Modal rounding OFF	N10 X... Y.... **RNDM=.7.3** ;modal rounding ON N11 X... Y... .... N100 **RNDM=.0** ;modal rounding OFF
RP	Polar radius	0.001 ... 99 999.999	Traversing in polar coordinates, definition of the pole; in addition: AP – polar angle	see G0, G1, G2; G3 G110, G111, G112
RPL	Angle of rotation with ROT, AROT	±0.00001 ... 359.9999	Specification in degrees; angle for a programmable rotation in the current plane G17 to G19	see ROT, AROT
SET(, , ,) REP()	Set values for the variable fields		SET: Various values, from the specified element up to: according to the number of values REP: the same value, from the specified element up to the end of the field	DEF REAL VAR2[12]=REP(4.5) ; all elements value 4.5 N10 R10=SET(1.1,2.3,4.4) ; R10=1.1, R11=2.3, R4=4.4
SF	Thread starting point when using G33	0.001 ... 359.999	Specified in degrees; the thread commencement point with G33 is offset by the specified value (not relevant for tapping)	See G33
SPI(n)	converts the spindle number n into axis identifier		n= 1 or n= 2 axis identifier: e.g. "SP1" or "C"	
SPOS	Spindle position	0.0000 ... 359.9999 If specified incrementally (IC): ±0.001 ... 99 999.999	specified in degrees; the spindle stops at the specified position (to achieve this, the spindle must provide the appropriate technical prerequisites: position control)	N10 SPOS=.... N10 SPOS=ACP(...) N10 SPOS=ACN(...) N10 SPOS=IC(...) N10 SPOS=DC(...)
STOPFIFO	Stops the fast machining step	–	Special function; filling of the buffer memory until STARTFIFO, "Buffer memory full" or "End of program" is detected.	STOPFIFO ;separate block, start of filling N10 X... N20 X...
START-FIFO	Start of fast machining step	–	Special function; the buffer memory is filled at the same time.	N30 X... STARTFIFO ;separate block, end of filling
STOPRE	Preprocessing stop	–	Special function; the next block is only decoded if the block before STOPRE is completed.	STOPRE ;separate block
TANG(Fo, Le1,Le2 ...)	Tangential control, definition	–	Fo: Name of the following axis) Le1: Name of master axis 1 Le2: Name of master axis 2 Further parameters optional This function is only available for the SINUMERIK 802Dsl pro.	TANG(C,X,Y) ; separate block TANG(C,X,Y,1"W","P") ; Max. number of parameters

Address	Meaning	Value Assignment	Information	Programming
TANGON (Fo,...)	Activate tangential control	–	Fo: Name of following axis (rotary axis) This function is only available for the SINUMERIK 802Dsl pro.	TANGON(C) ; Separate block TANGON(C,angle,dist,angletol) ; Max. number of parameters
TANGOF (Fo)	Deactivate tangential control	–	Fo: Name of following axis (rotary axis) This function is only available for the SINUMERIK 802Dsl pro.	TANGOF(C) ; separate block
TANGDEL (Fo)	Tangential control, delete definition	–	Fo: Name of following axis (rotary axis) This function is only available for the SINUMERIK 802Dsl pro.	TANGDEL(C) ; separate block
TLIFT(Fo)	Tangential control, insert intermediate block	–	Fo: Name of following axis (rotary axis) This function is only available for the SINUMERIK 802Dsl pro.	TLIFT(C) ; separate block
TRACYL(d)	Milling of the face end	d: 1.000 ... 99 999.999	Kinematic transformation	TRACYL(20.4) ; separate block ; Cylinder diameter: 20.4 mm TRACYL(20.4,1) ; also possible
TRAFOOF	Deactivate TRACYL	–	Disables all kinematic transformations	TRAFOOF ; separate block
TURN	Number of additional circle passes with helix interpolation	0 ... 999	in conjunction with circular interpolation G2/G3 in a plane G17 to G19 and infeed motion of the axis standing vertically on the plane	N10 G0 G17 X20 Y5 Z3 N20 G1 Z–5 F50 N30 G3 X20 Y5 Z–20 I0 J7.5 **TURN=2** ; in total, 3 full circles

8.2.2 Positional data

➢ Linear interpolation with rapid traverse: G00

Functionality

The rapid traverse movement G0 is used for rapid positioning of the tool, but not for **direct workpiece machining**.

All the axes can be traversed simultaneously – on a straight path.

For each axis, the maximum speed (rapid traverse) is defined in machine data. If only one axis traverses, it uses its rapid traverse. If two or three axes are traversed simultaneously, the path velocity (e.g. the resulting velocity at the tool tip) must be selected such that the **maximum possible path velocity** with consideration of all axes involved results.

A programmed feedrate (F word) has no meaning for G0. G2/G3 remains active until canceled by another instruction from this G group (G0, G1, G3, ...).

G0 X... Y... Z... ; Cartesian coordinates

G0 AP=... RP=... ; Polar coordinates

G0 AP=... RP=... Z... ; Cylinder coordinates (3-dimensional)

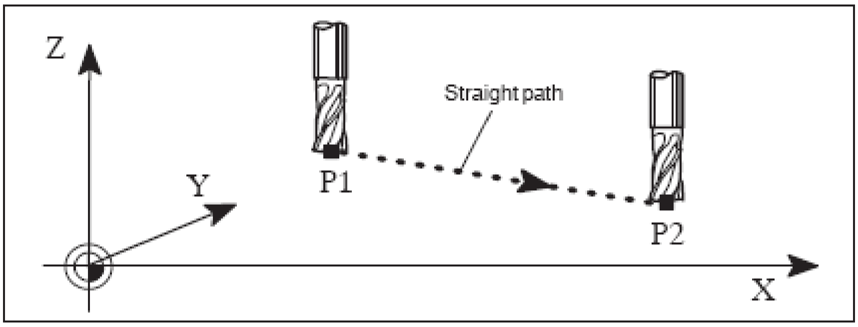


Fig 8.2-1

Programming

G0 X... Y... Z... ; Cartesian coordinates

G0 AP=... RP=... ; Polar coordinates

G0 AP=... RP=... Z... ; Cylinder coordinates (3-dimensional)

Note: Another option for linear programming is available with the angle specification

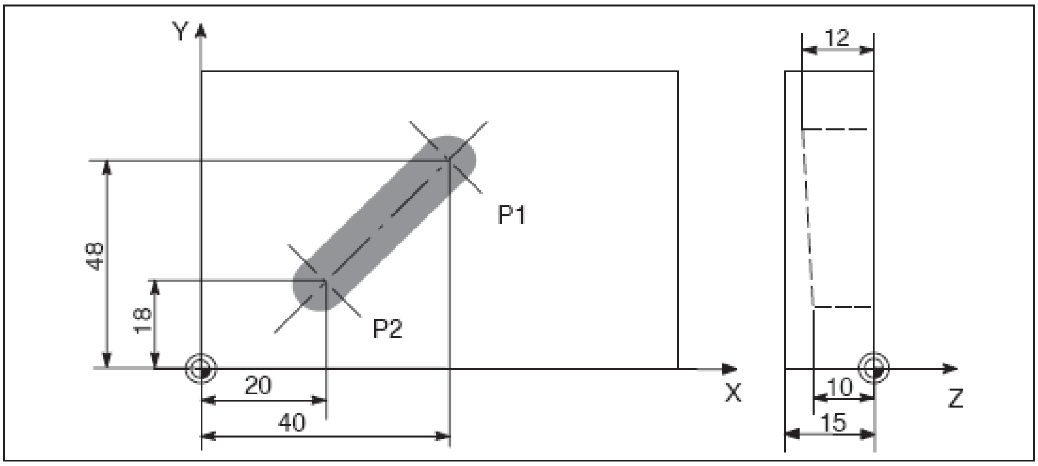


Fig 8.2-2

Programming example

N10 G0 X100 Y150 Z65 ; Cartesian coordinate

...

N50 G0 RP=16.78 AP=45 ; Polar coordinate

Information

Another group of G functions exists for movement to the position (see Section 8.3.16 ”Exact stop / continuous-path control mode: G60, G64”).

For G60 exact stop, a window with various precision values can be selected with another G group. For exact stop, an alternative instruction with non-modal effectiveness exists: G9. You should consider these options for adaptation to your positioning tasks.

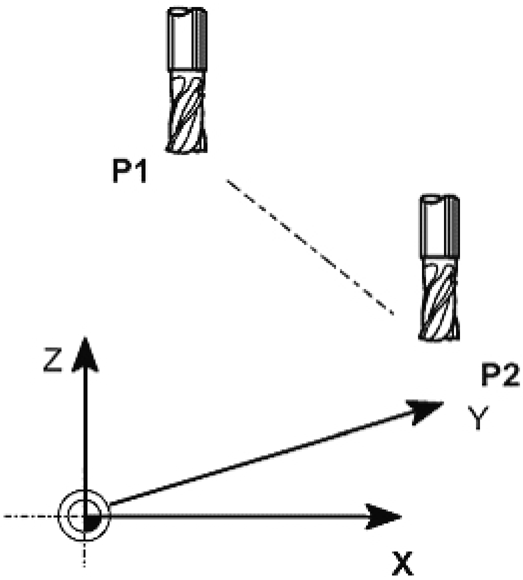


Fig 8.2-3

Linear interpolation with feedrate: G1

Functionality

The tool moves from the starting point to the end point along a straight path. The **path velocity** is determined by the programmed **F word**.

All axes can be traversed simultaneously.

G2/G3 remains active until canceled by another instruction from this G group (G0, G2, G3, ...).

Programming

G1 X... Y... Z... F... ; Cartesian coordinates

G1 AP=... RP=... F... ; Polar coordinates

G1 AP=... RP=... Z... F... ; Cylinder coordinates (3-dimensional)

Note: Another option for linear programming is available with the angle specification ANG=... (see Section 8.5.2 ”Blueprint programming”).

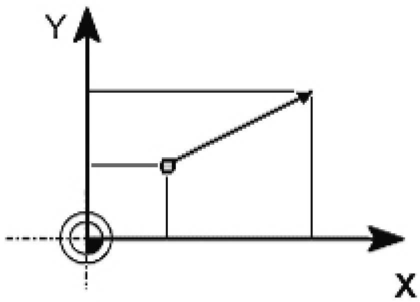


Fig 8.2-4

Programming example

N05 G0 G90 X40 Y48 Z2 S500 M3 ; Tools traverses at rapid traverse to P1,

3 axes simultaneously,

spindle speed = 500 r.p.m., CW rotation

N10 G1 Z–12 F100 ; Infeed to Z–12, feedrate 100 mm/min

N15 X20 Y18 Z–10 ; Tool traverses along a straight line in the space

to P2

N20 G0 Z100 ; Traversing at rapid traverse

N25 X-20 Y80

N30 M2 ; End of program

To machine a workpiece, spindle speed S ... and direction M3/M4 are required (see Section ”Spindle movement”).

Circular interpolation: G2, G3

Functionality

The tool moves from the starting point to the end point along a circular path. The direction is determined by the G function:

G2 ; CW

G3 ; CCW

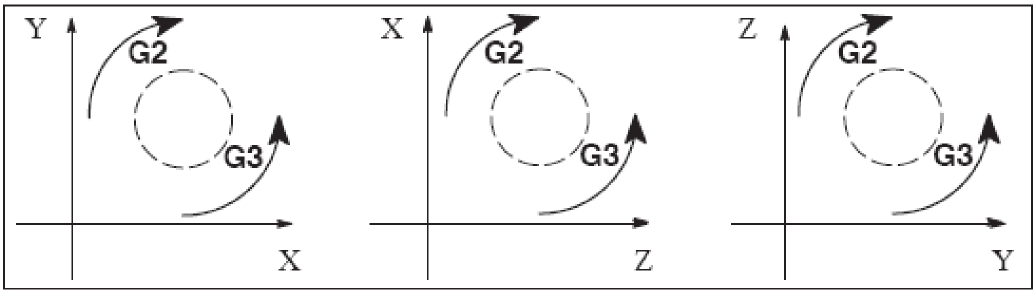


Fig 8.2-5

The description of the desired circle can be given in various ways:

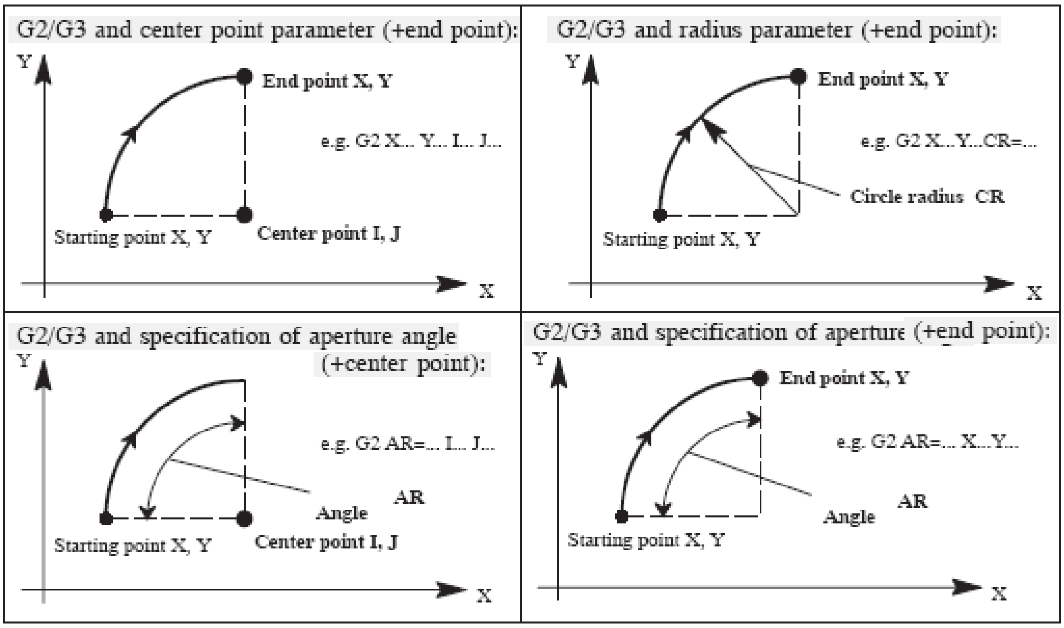


Fig 8.2-6

Programming

G2/G3 X... Y... I... J... ; Center and end points

G2/G3 CR=... X... Y... ; Circle radius and end point

G2/G3 AR=... I... J... ; Aperture angle and center point

G2/G3 AR=... X... Y... ; Aperture angle and end point

G2/G3 AP=... RP =... ; Polar coordinates, circle around the pole

Note

Further possibilities for circle programming result from:

CT – circle with tangential connection and

CIP – circle via intermediate point (see next sections).

Input tolerances for the circle

Circles are only accepted by the control system with a certain dimensional tolerance.

The circle radius at the starting and end points are compared here. If the difference is within the tolerance, the center point is exactly set internally. Otherwise, an alarm message is issued.

The tolerance value can be set via machine data (see ”Start-up Guide” 802Dsl).

Information

Full circles in a block are only possible if the center point and the end point are specified. For circles with radius specification, the arithmetic sign of CR =... is used to select the correct circle. It is possible to program 2 circles with the same starting and end points, as well as with the same radius and the same direction. The negative sign in front of CR =–... determines the circle whose circle segment is greater than a semi-circle; otherwise, the circle with the circle segment is less than or equal to the semi-circle and determined as follows:

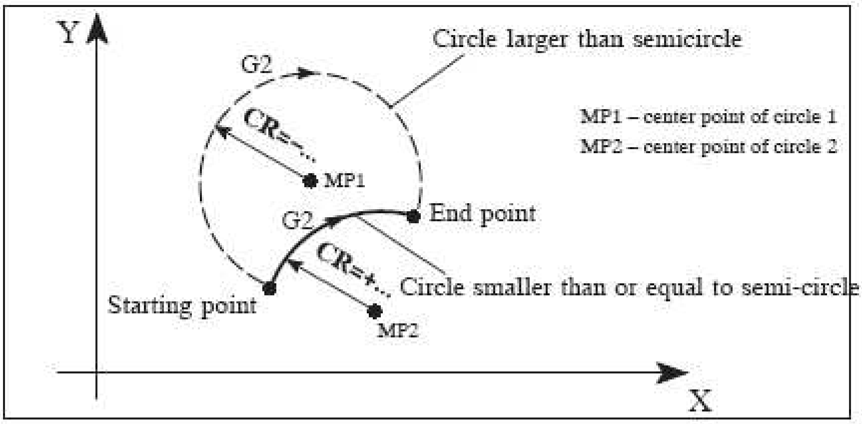


Fig 8.2-7

Programming example: Definition of center point and end point

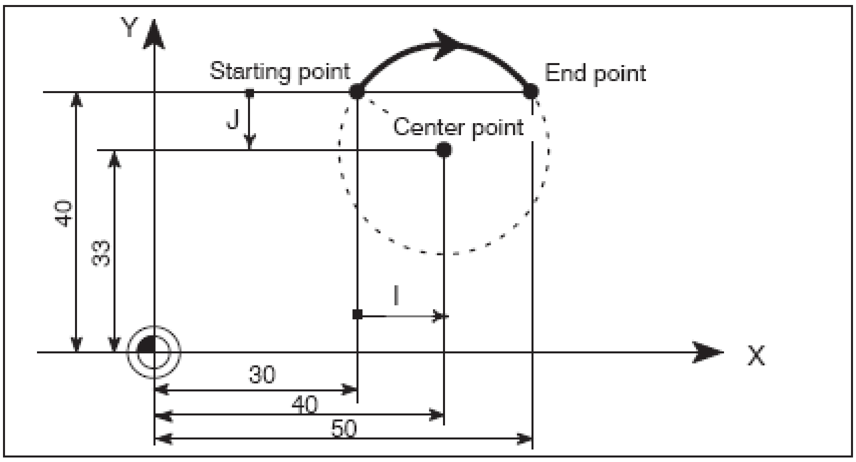


Fig 8.2-8

N5 G90 X30 Y40 ; Circle starting point for N10

N10 G2 X50 Y40 I10 J-7 ; End point and center point

Note: Center point values refer to the circle starting point!

Programming example: End point and radius specification

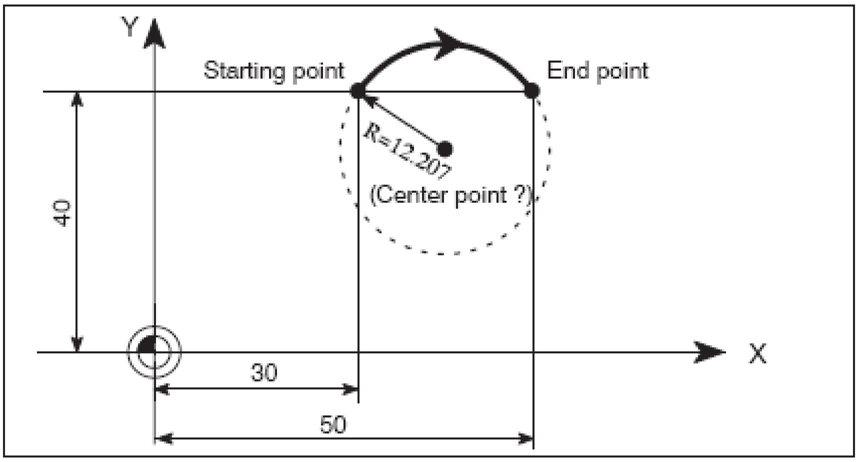


Fig 8.2-9

N5 G90 X30 Y40 ; Circle starting point for N10

N10 G2 X50 Y40 CR=12.207 ; End point and radius

Note: With a negative leading sign for the value with CR=–..., a circular segment larger than a semi-circle is selected.

Helix interpolation: G2/G3, TURN

Functionality

With helix interpolation, two movements are overlaid:

– circular movement in plane G17 or G18 or G19

– linear movement of the axis standing vertically on this plane.

The number of additional full–circle passes is programmed with TURN=. These are added to the actual circle programming.

The helix interpolation can preferably be used for the milling of threads or of lubricating grooves in cylinders.

Programming

G2/G3 X... Y... I... J... TURN =... ; Center and end points

G2/G3 CR = ... X... Y... TURN =... ; Circle radius and end point

G2/G3 AR = ... I... J... TURN =... ; Aperture angle and center point

G2/G3 AR = ... X... Y... TURN =... ; Aperture angle and end point

G2/G3 AP =... RP =... TURN =... ; Polar coordinates, circle around the pole

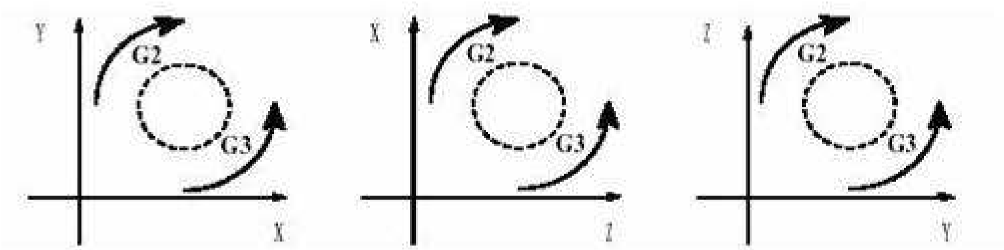


Fig 8.2-10

Programming example

N10 G17 ; X/Y plane, Z standing vertically on it

N20 ... Z ...

N30 G1 X0 Y50 F300 ; Approach starting point

N40 G3 X0 Y0 Z33 I0 J–25 TURN= 3 ; Helix

...

Thread cutting with constant lead: G33

Functionality

This requires a spindle with position measuring system.

The function G33 can be used to machine threads with constant lead of the following type:

If an appropriate tool is used, tapping with compensating chuck is possible.

The compensating chuck compensates the resulting path differences to a certain limited degree.

The drilling depth is specified by specifying one of the axes X, Y or Z; the spindle lead is specified via the relevant I, J or K.

G33 remains active until canceled by another instruction from this G group (G0, G1, G2, G3, ...).

Right-hand or left-hand threads

Right-hand or left-hand threads are set with the rotation direction of the spindle (M3 right (CW), M4 left (CCW) – see Section 8.4 ”Spindle movement”). To this end, the speed must be programmed under the address S or an appropriate speed must be set.

Remark:

A complete cycle of tapping with compensating chuck is provided by the standard cycle CYCLE840.

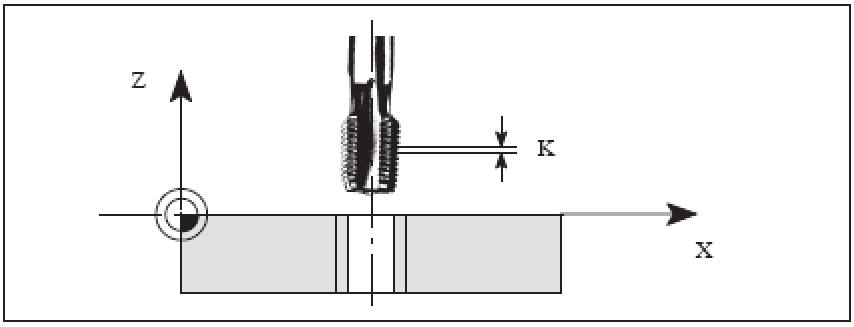


Fig 8.2-11

Programming example

metric thread 5,

pitch as per table: 0.8 mm/rev., tap hole already premachined:

N10 G54 G0 G90 X10 Y10 Z5 S600 M3 ; Approach starting point, spindle rotation CW

N20 G33 Z-25 K0.8 ; Tapping, end point –25 mm

N40 Z5 K0.8 M4 ; Retraction, spindle rotation CCW

N50 G0 X... Y... Z...

Axis velocity

With G33 threads, the velocity of the axis for the thread lengths is determined on the basis of the spindle speed and the thread pitch. The **feedrate F is not relevant**. It is, however, stored. However, the maximum axis velocity (rapid traverse) defined in the machine data can not be exceeded. This will result in an alarm.

Information

Important

_ The spindle speed override switch should remain unchanged for thread machining.

_ The feedrate override switch has no meaning in this block.

Tapping with compensating chuck: G63

Functionality

G63 can be used for tapping with compensating chuck. The programmed feedrate F must match with the spindle speed S (programmed under the address ”S” or specified speed) and with the thread pitch of the drill:

F [mm/min] = S [r.p.m.] x thread pitch [mm/rev.]

The compensating chuck compensates the resulting path differences to a certain limited degree.

The drill is retracted using G63, too, but with the spindle rotating in the opposite direction M3 –<–> M4.

G63 is non-modal. In the block after G63, the previous G command of the ”Interpolation type” group (G0, G1,G2, ...) is active again.

Right-hand or left-hand threads

Right-hand or left-hand threads are set with the rotation direction of the spindle (M3 right (CW), M4 left (CCW) – see Section 8.4 ”Spindle movement”).

Remark:

The standard cycle CYCLE840 provides a complete tapping cycle with compensating chuck (but with G33 and the relevant prerequisites).

Programming example

metric thread 5,

pitch as per table: 0.8 mm/rev., tap hole already premachined:

N10 G54 G0 G90 X10 Y10 Z5 S600 M3 ; Approach starting point, spindle rotation CW

N20 G63 Z-25 F480 ; Tapping, end point –25 mm

N40 G63 Z5 M4 ; Retraction, spindle rotation CCW

N50 X... Y... Z...

Fixed point approach: G75

Functionality

By using G75, a fixed point on the machine, e.g. tool change point, can be approached.

The position is stored permanently in the machine data for all axes. No offset is effective. The speed of each axis is its rapid traverse.

G75 requires a separate block and is non-modal. The machine axis identifier must be programmed!

In the block after G75, the previous G command of the ”Interpolation type” group (G0, G1,G2, ...) is active again.

Programming example

N10 G75 X1 = 0 Y1 = 0 Z1 = 0

Remark: The programmed position values for X1, Y1 (any value, here = 0) are ignored, but must still be written.

Reference point approach: G74

Functionality

The reference point can be approached in the NC program with G74. The direction and speed of each axis are stored in machine data.

G74 requires a separate block and is non-modal. The machine axis identifier must be programmed!

In the block after G74, the previous G command of the ”Interpolation type” group (G0, G1,G2, ...) is active again.

Programming example

N10 G74 X1 = 0 Y1 = 0 Z1 = 0

Remark: The programmed position values for X1, Y1 (any value, here = 0) are ignored, but must still be written.

Unit of measure for F with G94, G95

The dimension unit for the F word is determined by G functions:

_ G94 **F** as the feedrate in**mm/min**

_ G95 **F** as the feedrate in **mm/rev.** of the spindle

(only meaningful when the spindle is running)

Remark**:**

This unit of measure applies to metric dimensions. According to Section ”Metric and inch dimensioning”, settings with inch dimensioning are also possible.

Programming example

N10 G94 F310 ; Feedrate in mm/min

...

N110 S200 M3 ; Spindle rotation

N120 G95 F15.5 ; Feedrate in mm/rev.

Remark: Write a new F word if you change G94 – G95.

Exact stop / continuous-path control mode: G9, G60, G64

Functionality

G functions are provided for optimum adaptation to different requirements to set the traversing behavior at the block borders and for block advancing. Example: For example, you would like to quickly position with the axes or you would like to machine path contours over multiple blocks.

Programming

G60 ; Exact stop – modal

G64 ; Continuous-path control mode

G9 ; Exact stop – non-modal

G601 ; Exact stop window fine

G602 ; Exact stop window coarse

Exact stop G60, G9

If the exact stop function (G60 or G9) is active, the velocity for reaching the exact end position at the end of a block is decelerated to zero.

Another modal G group can be used here to set when the traversing movement of this block is considered ended and the next block is started.

_ G601 ; Exact stop window fine

Block advance takes place when all axes have reached the ”Exact stop window fine” (value in the machine data).

_ G602 ; Exact stop window coarse

Block advance takes place when all axes have reached the ”Exact stop window coarse”

The selection of the exact stop window has a significant influence on the total time if many positioning operations are executed. Fine adjustments require more time.

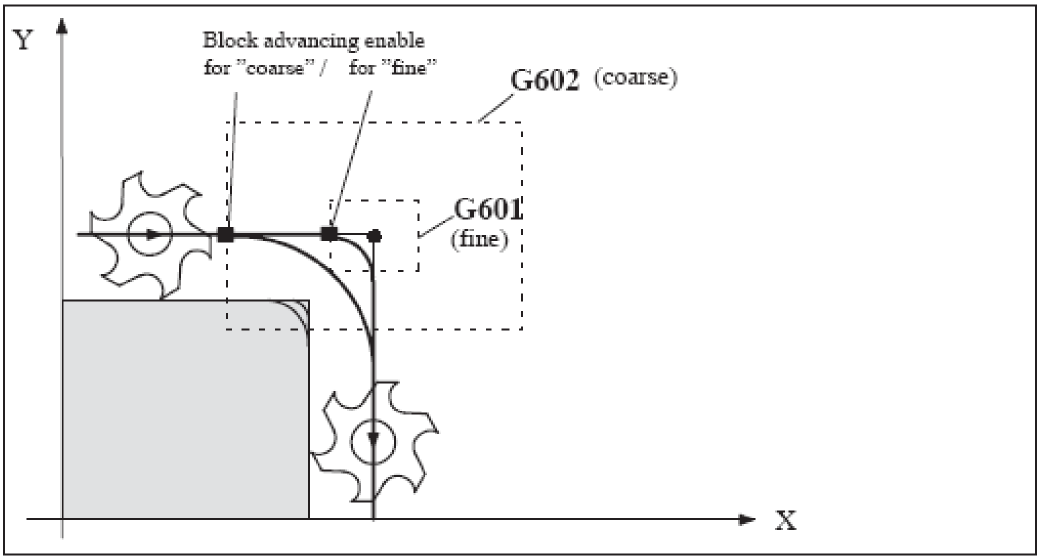


Fig 8.2-12

Programming example

N5 G602 ; Exact stop window coarse

N10 G0 G60 X... ; Exact stop modal

N20 X... Y... ; G60 remains active

...

N50 G1 G601 ... ; Exact stop window fine

N80 G64 X... ; Switching to continuous-path control mode

...

N100 G0 G9 X... ; Exact stop is only effective for this block

N111 ... ; Continuous-path control mode again

Remark: The G9 command only generates exact stop for the block in which it is programmed; G60, however, is effective until it is canceled by G64.

Continuous-path control mode G64

The objective of the continuous-path control mode is to avoid deceleration at the block boundaries and to switch **to the next block** with **a path velocity as constant as possible** (in the case of tangential transitions). The function works with **look-ahead velocity control** over several blocks.

For non-tangential transitions (corners), the velocity can reduced rapidly enough so that the axes are subject to a relatively high velocity change over a short time. This may lead to a significant jerk (acceleration change). The size of the jerk can be limited by activating the SOFT function.

Programming example

N10 G64 G1 X... F... ; Continuous-path control mode

N20 Y.. ; Continuous-path control mode continues to be active

...

N180 G60 ... ; switching to exact stop

Look-ahead velocity control

In the continuous-path control mode with G64, the control system automatically determines the velocity control for several NC block in advance. This enables acceleration and deceleration across multiple blocks with approximately tangential transitions. For paths that consist of short travels in the NC blocks, higher velocities can be achieved than without look ahead.

Spindle speed limitation: G25, G26

Functionality

In the program, you can limit the limit values that would otherwise apply for a controlled spindle by writing G25 or G26 and the spindle address S with the speed limit value.

This overwrites the values entered in the setting data at the same time.

G25 and G26 each require a separate block. A previously programmed speed S is maintained.

Programming

G25 S... ; Programmable lower spindle speed limitation

G26 S... ; Upper speed limitation

Information

The outmost limits of the spindle speed are set in machine data. Appropriate inputs via the operator panel can activate various setting data for further limiting.

Programming example

N10 G25 S12 ; Lower spindle limit speed: 12 r.p.m.

N20 G26 S700 ; Upper spindle limit speed : 700 r.p.m.

Note

G25/G26 are used in conjunction with axis addresses for a working area limitation (see Section ”Working area limitation”).

Selecting the tool radius compensation: G41, G42

Functionality

The control system is working with tool radius compensation in the selected plane G17 to G19.

A tool with a corresponding D number must be active. The tool radius compensation is activated by G41/G42. The control system automatically calculates the required equidistant tool paths for the programmed contour for the respective current tool radius.

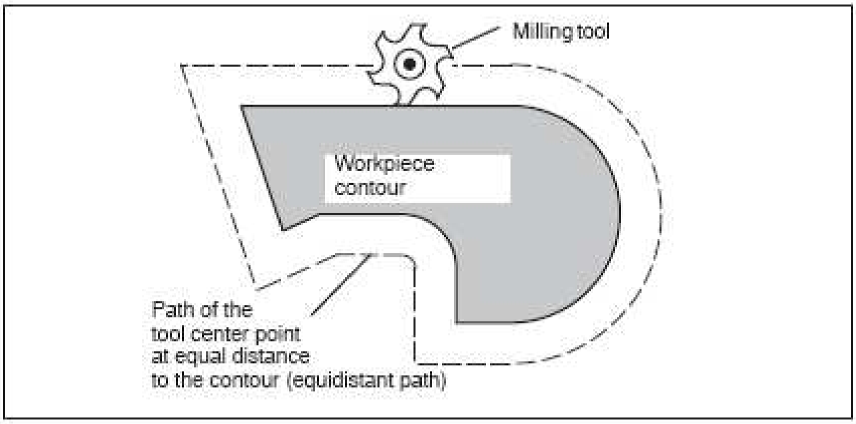


Fig 8.2-13

Programming

G41 X... Y... ; Tool radius compensation left of the contour

G42 X... Y... ; Tool radius compensation right of the contour

Remark: The selection can only be made for linear interpolation (G0, G1).

Program both axes of the plane (e.g. with G17: X, Y). If you only specify one axis, the second axis is automatically completed with the last programmed value.

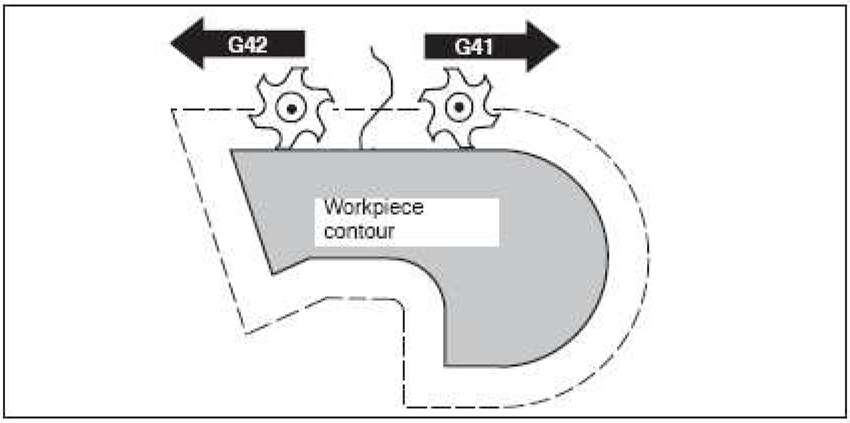


Fig 8.2-14

Tool radius compensation OFF: G40

Functionality

The compensation mode (G41/G42) is deselected with G40. G40 is also the activation position at the beginning of the program.

The tool ends the **block in front of G40** in the normal position (compensation vector vertically to the tangent at the end point);

If G40 is active, the reference point is the tool center point. Subsequently, when deselected, the tool tip approaches the programmed point.

Always select the end point of the G40 block such that collision-free traversing is guaranteed!

Programming

G40 X... Y... ; Tool radius compensation OFF

Remark: The compensation mode can only be deselected with linear interpolation (G0, G1).

8.3 Overview of cycles

Cycles are generally applicable technology subroutines that can be used to carry out a specific machining process, such as drilling of a thread (tapping) or milling of a pocket. These cycles are adapted to individual tasks by parameter assignment.

Drilling cycle, drilling pattern cycles and milling cycles

The following standard cycles can be carried out using the SINUMERIK 802D control system:

_ Drilling cycles

CYCLE81 Drilling, centering

CYCLE82 Drilling, counterboring

CYCLE83 Deep hole drilling

CYCLE84 Rigid tapping

CYCLE84 Tapping with compensating chuck

CYCLE85 Reaming 1 (boring out 1)

CYCLE86 Boring (boring out 2)

CYCLE87 Drilling with stop 1 (boring out 3)

CYCLE87 Drilling with stop 2 (boring out 4)

CYCLE85 Reaming 2 (boring out 5)

With SINUMERIK 840D, the boring cycles CYCLE85 ... CYCLE89 are called boring 1 ... boring 5, but are nevertheless identical in their function.

_ Drill pattern cycles

HOLES1 Row of holes

HOLES2 Circle of holes

_ Milling cycles

CYCLE71 Face milling

CYCLE72 Contour milling

CYCLE76 Rectangular spigot milling

CYCLE77 Circular spigot milling

LONGHOLE Long hole

SLOT1 Milling pattern 'Slots on a circle'

SLOT2 Milling pattern "Circular slots"

POCKET3 Rechtecktasche fräsen (mit beliebigem Fräser)

POCKET4 Milling of rectangular pocket (using any milling cutter)

CYCLE90 Thread milling

The cycles are supplied with the tool box. They are loaded via the RS232 interface into the part program memory during the start-up of the control system.

Auxiliary cycle subroutines

The cycle package includes the following auxiliary subroutines:

_ cyclesm.spf

_ steigung.spf and

_ meldung.spf

These must always be loaded in the control.

Drilling, centering – CYCLE81

Programming

CYCLE81(RTP, RFP, SDIS, DP, DPR)

RTP real Retraction plane (absolute)

RFP real Reference plane (absolute)

SDIS real Safety clearance (enter without sign)

DP real Final drilling depth (absolute)

DPR real Final drilling depth relative to the reference plane (enter without sign)

Function

The tool drills at the programmed spindle speed and feedrate to the entered final drilling depth.

Drilling, counterboring – CYCLE82

Programming

CYCLE82(RTP, RFP, SDIS, DP, DPR, DTB)

Parameters

Table 9-4 Parameters for CYCLE82

RTP real Retraction plane (absolute)

RFP real Reference plane (absolute)

SDIS real Safety clearance (enter without sign)

DP real Final drilling depth (absolute)

DPR real Final drilling depth relative to the reference plane (enter without sign)

DTB real Dwell time at final drilling depth (chip breaking)

Function

The tool drills at the programmed spindle speed and feedrate to the entered final drilling depth. A dwell time can be allowed to elapse when the final drilling depth has been reached.

Sequence

Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.

The cycle creates the following sequence of motions:

_ Approach of the reference plane brought forward by the safety clearance by using G0

_ Traversing to the final drilling depth with the feedrate (G1) programmed prior to the cycle call

_ Dwell time at final drilling depth

_ Retraction to the retraction plane with G0

Explanation of the parameters

For the parameters RTP, RFP, SDIS, DP, DPR, see CYCLE81

Deep hole drilling – CYCLE83

Programming

CYCLE83(RTP, RFP, SDIS, DP, DPR, FDEP, FDPR, DAM, DTB, DTS, FRF, VARI)

Parameters

Table 9-5 Parameters for CYCLE83

RTP real Retraction plane (absolute)

RFP real Reference plane (absolute)

SDIS real Safety clearance (enter without sign)

DP real Final drilling depth (absolute)

DPR real Final drilling depth relative to the reference plane (enter without

sign)

FDEP real First drilling depth (absolute)

FDPR real First drilling depth relative to the reference plane (enter without

sign)

DAM real Amount of degression (enter without sign)

DTB real Dwell time at final drilling depth (chip breaking)

DTS real Dwell time at starting point and for swarf removal

FRF real Feedrate factor for the first drilling depth (enter without sign)

Range of values: 0.001 ... 1

VARI int Machining type:

Chip breaking = 0

Swarf removal = 1

Function

The tool drills at the programmed spindle speed and feedrate to the entered final drilling depth.

Deep hole drilling is performed with a depth infeed of a maximum definable depth executed several times, increasing gradually until the final drilling depth is reached.

The drill can either be retracted to the reference plane + safety clearance after every infeed depth for swarf removal or retracted in each case by 1 mm for chip breaking.

Sequence

Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.

Function

The tool drills at the programmed spindle speed and feedrate to the entered final drilling depth.

Deep hole drilling is performed with a depth infeed of a maximum definable depth executed several times, increasing gradually until the final drilling depth is reached.

The drill can either be retracted to the reference plane + safety clearance after every infeed depth for swarf removal or retracted in each case by 1 mm for chip breaking.

Sequence

Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.

Rigid tapping – CYCLE84

Programming

CYCLE84 (RTP, RFP, SDIS, DP, DPR, DTB, SDAC, MPIT, PIT, POSS, SST, SST1)

Parameters

Table 9-6 Parameters for CYCLE84

RTP real Retraction plane (absolute)

RFP real Reference plane (absolute)

SDIS real Safety clearance (enter without sign)

DP real Final drilling depth (absolute)

DPR real Final drilling depth relative to the reference plane (enter without sign)

DTB real Dwell time at thread depth (chip breaking)

SDAC int Direction of rotation after end of cycle

Values: 3, 4 or 5 (for M3, M4 or M5)

MPIT real Pitch as a thread size (signed):

Range of values 3 (for M3) ... 48 (for M48); the sign determines the direction of rotation in the thread

PIT real Pitch as a value (signed)

Value range: 0.001 ... 2000.000 mm); the sign determines the direction of rotation in the thread

POSS real Spindle position for oriented spindle stop in the cycle (in degrees)

SST real Speed for tapping

SST1 real Speed for retraction

Function

The tool drills at the programmed spindle speed and feedrate to the entered final thread depth.

CYCLE84 can be used to perform rigid tapping operations. For tapping with compensating chuck, a separate cycle CYCLE840 is provided.

Sequence

Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.

The cycle creates the following sequence of motions:

_ Approach of the reference plane brought forward by the safety clearance by using G0

_ Oriented spindle stop (value in the parameter POSS) and switching the spindle to axis mode

_ Tapping to final drilling depth and speed SST

_ Dwell time at thread depth (parameter DTB)

_ Retraction to the reference plane brought forward by the safety clearance, speed SST1 and direction reversal

_ Retraction to the retraction plane with G0; spindle mode is reinitiated by reprogramming the spindle speed active before the cycle was called and the direction of rotation programmed under SDAC

Explanation of the parameters

For the parameters RTP, RFP, SDIS, DP, DPR, see CYCLE81

Tapping with compensating chuck – CYCLE840

Programming

CYCLE840 (RTP, RFP, SDIS, DP, DPR, DTB, SDR, SDAC, ENC, MPIT, PIT)

Parameters

RTP real Retraction plane (absolute)

RFP real Reference plane (absolute)

SDIS real Safety clearance (enter without sign)

DP real Final drilling depth (absolute)

DPR real Final drilling depth relative to the reference plane (enter without sign)

DTB real Dwell time at thread depth (chip breaking)

SDR int Direction of rotation for retraction

Values: 0 (automatic reversal of direction of rotation)

3 or 4 (for M3 or M4)

SDAC int Direction of rotation after end of cycle

Values: 3, 4 or 5 (for M3, M4 or M5)

ENC int Tapping with/without encoder

Values: 0 = with encoder

1 = without encoder

MPIT real Pitch as a thread size (signed):

Range of values 3 (for M3) ... 48 (for M60)

PIT real Pitch as a value (signed)

Value range: 0.001 ... 2,000.000 mm

Function

The tool drills at the programmed spindle speed and feedrate to the entered final thread depth.

Use this cycle to perform tapping with compensating chuck

_ without encoder and

_ with encoder.

Sequence of operations: Tapping with compensating chuck without encoder

Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.

Reaming 1 (boring 1) – CYCLE85

Programming

CYCLE85(RTP, RFP, SDIS, DP, DPR, DTB, FFR, RFF)

Parameters

Table 9-8 Parameters for CYCLE85

RTP real Retraction plane (absolute)

RFP real Reference plane (absolute)

SDIS real Safety clearance (enter without sign)

DP real Final drilling depth (absolute)

DPR real Final drilling depth relative to the reference plane (enter without sign)

DTB real Dwell time at final drilling depth (chip breaking)

FFR real Feedrate

RFF real Retraction feedrate

Function

The tool drills at the programmed spindle speed and feedrate to the entered final drilling depth.

The inward and outward movement is performed at the feedrate assigned to FFR and RFF respectively.

Sequence

Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.

The cycle creates the following sequence of motions:

_ Approach of the reference plane brought forward by the safety clearance by using G0

_ Traversing to the final drilling depth with G1 and at the feedrate programmed under the parameter FFR

_ Dwell time at final drilling depth

_ Retraction to the reference plane brought forward by the safety clearance with G1 and the retraction feedrate defined under the parameter RFF

_ Retraction to the retraction plane with G0

Boring (boring 2) – CYCLE86

Programming

CYCLE86 (RTP, RFP, SDIS, DP, DPR, DTB, SDIR, RPA, RPO, RPAP, POSS)

Parameters

Table 9-9 Parameters for CYCLE86

RTP real Retraction plane (absolute)

RFP real Reference plane (absolute)

SDIS real Safety clearance (enter without sign)

DP real Final drilling depth (absolute)

DPR real Final drilling depth relative to the reference plane (enter without sign)

DTB real Dwell time at final drilling depth (chip breaking)

SDIR int Direction of rotation

Values: 3 (for M3)

4 (for M4)

RPA real Retraction path along the 1st axis of the plane (incremental, enter with sign)

RPO real Retraction path along the 2nd axis of the plane (incremental, enter with sign)

RPAP real Retraction path along the boring axis (incremental, enter with sign)

POSS real Spindle position for oriented spindle stop in the cycle (in degrees)

Function

The cycle supports the boring of holes with a boring bar.

The tool drills at the programmed spindle speed and feedrate to the entered final drilling depth.

With boring 2, oriented spindle stop is activated once the drilling depth has been reached.

Then, the programmed retraction positions are approached in rapid traverse and, from there, the retraction plane.

Sequence

Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.

The cycle creates the following sequence of motions:

_ Approach of the reference plane brought forward by the safety clearance by using G0

_ Traversing to final drilling depth with G1 and the feedrate programmed prior to the cycle call

_ Dwell time to final drilling depth

_ Oriented spindle stop at the spindle position programmed under POSS

_ Traverse retraction path in up to three axes with G0

_ Retraction in the boring axis to the reference plane brought forward by the safety clearance by using G0

_ Retraction to the retraction plane with G0 (initial drilling position in both axes of the plane)

Boring with Stop 1 (boring 3) – CYCLE87

Programming

CYCLE87 (RTP, RFP, SDIS, DP, DPR, SDIR)

Parameters

RTP real Retraction plane (absolute)

RFP real Reference plane (absolute)

SDIS real Safety clearance (enter without sign)

DP real Final drilling depth (absolute)

DPR real Final drilling depth relative to the reference plane (enter without sign)

SDIR int Direction of rotation

Values: 3 (for M3)

4 (for M4)

Function

The tool drills at the programmed spindle speed and feedrate to the entered final drilling depth.

During boring 3, a spindle stop without orientation M5 is generated after reaching the final drilling depth, followed by a programmed stop M0. Pressing the NC START key continues the retraction movement at rapid traverse until the retraction plane is reached.

Sequence

Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.

Drilling with stop 2 (boring 4) – CYCLE88

Programming

CYCLE88 (RTP, RFP, SDIS, DP, DPR, DTB, SDIR)

Parameters

RTP real Retraction plane (absolute)

RFP real Reference plane (absolute)

SDIS real Safety clearance (enter without sign)

DP real Final drilling depth (absolute)

DPR real Final drilling depth relative to the reference plane (enter without sign)

DTB real Dwell time at final drilling depth (chip breaking)

SDIR int Direction of rotation

Values: 3 (for M3)

4 (for M4)

Function

The tool drills at the programmed spindle speed and feedrate to the entered final drilling depth. When boring with stop, a spindle stop without orientation M5 and a programmed stop are generated when the final drilling depth is reached. Pressing the NC START key continues the retraction movement at rapid traverse until the retraction plane is reached.

Sequence

Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.

The cycle creates the following sequence of motions:

_ Approach of the reference plane brought forward by the safety clearance by using G0

_ Traversing to final drilling depth with G1 and the feedrate programmed prior to the cycle call

_ Dwell time at final drilling depth

_ Spindle and program stop with M5 M0. After program stop, press the NC START key.

_ Retraction to the retraction plane with G0

Reaming 2 (boring 5) – CYCLE89

Programming

CYCLE89 (RTP, RFP, SDIS, DP, DPR, DTB)

Parameters

RTP real Retraction plane (absolute)

RFP real Reference plane (absolute)

SDIS real Safety clearance (enter without sign)

DP real Final drilling depth (absolute)

DPR real Final drilling depth relative to the reference plane (enter without sign)

DTB real Dwell time at final drilling depth (chip breaking)

Function

The tool drills at the programmed spindle speed and feedrate to the entered final drilling depth. When the filan drilling depth is reached, the programmed dwell time is active.

Sequence

Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.

The cycle creates the following sequence of motions:

_ Approach of the reference plane brought forward by the safety clearance by using G0

_ Traversing to final drilling depth with G1 and the feedrate programmed prior to the cycle call

_ Dwell time to final drilling depth

_ Retraction up to the reference plane brought forward by the safety clearance using G1 and the same feedrate value

_ Retraction to the retraction plane with G0

Row of holes – HOLES1

Programming

HOLES1 (SPCA, SPCO, STA1, FDIS, DBH, NUM)

Parameters

SPCA real 1. axis of the plane (abscissa) of a reference point on the straight line (absolute)

SPCO real 2. axis of the plane (ordinate) of this reference point (absolute)

STA1 real Angle to the 1st axis of the plane (abscissa)

Value range: –180<STA1<=180 degrees

FDIS real Distance from the first hole to the reference point (enter without

sign)

DBH real Distance between the holes (enter without sign)

NUM int Number of holes

Function

This cycle can be used to produce a row of holes, i.e. a number of holes arranged along a straight line, or a grid of holes. The type of hole is determined by the drilling hole cycle that has already been called modally.

Sequence

To avoid unnecessary travel, the cycle calculates whether the row of holes is machined starting from the first hole or the last hole from the actual position of the plane axes and the geometry of the row of holes. The drilling positions are then approached one after the other at rapid traverse.

Circle of holes – HOLES2

Programming

HOLES2 (CPA, CPO, RAD, STA1, INDA, NUM)

Parameters

CPA real Center point of circle of holes (absolute), 1st axis of the plane

CPO real Center point of circle of holes (absolute), 2nd axis of the plane

RAD real Radius of circle of holes (enter without sign)

STA1 real Starting angle

Value range: –180<STA1<=180 degrees

INDA real Incrementing angle

NUM int Number of holes

Function

Use this circle to machine a circle of holes. The machining plane must be defined before the cycle is called.

The type of hole is determined by the drilling hole cycle that has already been called modally.

Figure 9-30

Face milling – CYCLE71

Programming

CYCLE71 (_RTP, _RFP, _SDIS, _DP, _PA, _PO, _LENG, _WID, _STA,_MID, _MIDA, _FDP, _FALD, _FFP1, _VARI, _FDP1)

Parameters

_RTP real Retraction plane (absolute)

_RFP real Reference plane (absolute)

_SDIS real Safety clearance (to be added to the reference plane; enter without sign)

_DP real Depth (absolute)

_PA real Starting point (absolute), 1st axis of the plane

_PO real Starting point (absolute), 2nd axis of the plane

_LENG real Rectangle length along the 1st axis, incremental.

The corner from which the dimension starts results from the sign.

_WID real Rectangle length along the 2nd axis, incremental.

The corner from which the dimension starts results from the sign.

_STA real Angle between the longitudinal axis of the rectangle and the

1st axis of the plane (abscissa, enter without sign);

Range of values: 0 ≤_STA 180

_MID real Maximum infeed depth (enter without sign)

_MIDA real Maximum infeed width during solid machining in the plane as

a value (enter without sign)

_FDP real Retraction travel in the finishing direction (incremental,

enter without sign)

_FALD real Finishing dimension in the depth (incremental, enter without sign)

_FFP1 real Feedrate for surface machining

_VARI integer Machining type (enter without sign)

UNITS DIGIT

Values: 1 Roughing

2 Finishing

TENS DIGIT:

Values: 1 Parallel to the 1st axis of the plane, unidirectional

2 Parallel to the 2nd axis of the plane, unidirectional

3 Parallel to the 1st axis of the plane,

changing direction

4 Parallel to the 2nd axis of the plane,

changing direction

_FDP1 real Overrun travel in the direction of the plane infeed (incremental,

enter without sign)

Contour milling – CYCLE72

Programming

CYCLE72 (_KNAME, _RTP, _RFP, _SDIS, _DP, _MID, _FAL, _FALD, _FFP1, _FFD, _VARI, _RL, _AS1, _LP1, _FF3, _AS2, _LP2)

Parameters

_KNAME string Name of contour subroutine

_RTP real Retraction plane (absolute)

_RFP real Reference plane (absolute)

_SDIS real Safety clearance (to be added to the reference plane; enter without sign)

_DP real Depth (absolute)

_MID real Maximum infeed depth (incremental; enter without sign)

_FAL real Finishing allowance at the edge contour (enter without sign)

_FALD real Finishing allowance at the base (incremental, enter without sign)

_FFP1 real Feedrate for surface machining

_FFD real Feedrate for depth infeed (enter without sign)

_VARI integer Machining type (enter without sign)

UNITS DIGIT

Values: 1 Roughing

2 Finishing

TENS DIGIT:

Values: 0 Intermediate travel with G0

1 Intermediate travel with G1

HUNDREDS DIGIT

Values: 0...Retraction at end of contour to _RTP

1...Retraction at end of contour to _RFP + _SDIS

2 Retraction by _SDIS at end of contour

3 No retraction at end of contour

_RL integer Traveling around the contour either centrally, to the right or to the left (with G40, G41 or G42; enter without sign)

Values: 40...G40 (approach and retraction, straight line only)

41...G41

42...G42

Rectangular spigot milling – CYCLE76

Programming

CYCLE76 (_RTP, _RFP, _SDIS, _DP, _DPR, _LENG, _WID, _CRAD, _PA, _PO, _STA, _MID, _FAL, _FALD, _FFP1, _FFD, _CDIR, _VARI, _AP1, _AP2)

Parameters

_RTP real Retraction plane (absolute)

_RFP real Reference plane (absolute)

_SDIS real Safety clearance (enter without sign)

_DP real Final drilling depth (absolute)

_DPR real Final drilling depth relative to the reference plane (enter without sign)

_LENG real Spigot length (enter without sign)

_WID real Spigot length (enter without sign)

_CARD real Spigot corner radius (enter without sign)

_PA real Reference point of spigot, abscissa (absolute)

_PO real Reference point of spigot, ordinate (absolute)

_STA real Angle between longitudinal axis and 1st axis of plane

_MID real Maximum depth infeed (incremental; enter without sign)

_FAL real Final machining allowance at the margin contour (incremental)

_FALD real Finishing allowance at the base (incremental, enter without sign)

_FFP1 real Feedrate at the contour

_FFD real Feedrate for depth infeed

_CDIR integer Milling direction (enter without sign)

Values: 0 Synchronous milling

1 Conventional milling

2 With G2 (independent of spindle direction)

3 With G3

_VARI integer Machining type

Values: 1 Roughing up to finishing allowance

2 Finishing (allowance X/Y/Z=0)

_AP1 real Length of blank spigot

Function

Use this cycle to machine rectangular spigots in the machining plane. For finishing, a face cutter is required. The depth infeed is always carried out in the position upstream of the semicircle style approach to the contour.

_PA, _PO (reference point)

Use the parameters _PA and _PO to define the reference point of the spigot along the abscissa and the ordinate.

This is the spigot center point.

_STA (angle)

_STA specifies the angle between the 1st axis of the plane (abscissa) and the longitudinal axis of the spigot.

_CDIR (milling direction)

Use this parameter to specify the machining direction for the spigot.

By using the parameter _CDIR, the milling direction

_ can be programmed directly with ”2 for G2” and ”3 for G3” or,

_ alternatively, ”Synchronous milling” or ”Conventional milling”.

can be programmed for the transformation declared. synchronized operation or reverse rotation are determined internally in the cycle via the direction of rotation of the spindle activated prior to calling the cycle.

Synchronous milling Conventional milling

M3 →G3 M3 →G2

M4 →G2 M4 →G3

_VARI (machining type)

Use the parameter _VARI to define the machining type.

Possible values are:

_ 1 = roughing

_ 2 = finishing

_AP1, _AP2 (blank dimensions)

When machining the spigot, it is possible to take into account blank dimensions (e.g. when machining precast parts).

The blank dimensions for length and width (_AP1 and _AP2) are programmed without sign and are placed by the cycle symmetrically around the pocket center point via calculation.

The internally calculated radius of the approach semicircle depends on this dimension.

Circular spigot milling – CYCLE77

Programming

CYCLE77 (_RTP, _RFP, _SDIS, _DP, _DPR, _PRAD, _PA, _PO, _MID, _FAL, _FALD, _FFP1, _FFD, _CDIR, _VARI, _AP1)

Parameters

The following input parameters are always required:

Table 9-18 Parameters for CYCLE77

_RTP real Retraction plane (absolute)

_RFP real Reference plane (absolute)

_SDIS real Safety clearance (enter without sign)

_DP real Depth (absolute)

_DPR real Depth relative to the reference plane (enter without sign)

_PRAD real Spigot diameter (enter without sign)

_PA real Center point of spigot, abscissa (absolute)

_PO real Center point of spigot, ordinate (absolute)

_MID real Maximum depth infeed (incremental; enter without sign)

_FAL real Final machining allowance at the margin contour (incremental)

_FALD real Finishing allowance at the base (incremental, enter without sign)

_FFP1 real Feedrate at the contour

_FFD real Feedrate for depth infeed (or spatial infeed)

_CDIR integer Milling direction (enter without sign)

Values: 0 Synchronous milling

1 Conventional milling

2 With G2 (independent of spindle direction)

3 With G3

_VARI integer Machining type

Values: 1 Roughing up to finishing allowance

2 Finishing (allowance X/Y/Z=0)

_AP1 real Length of blank spigot

Function

Use this cycle to machine circular spigots in the machining plane. For finishing, a face cutter is required. The depth infeed is always carried out in the position upstream of the semicircle style approach to the contour.

Figure 9-48

Slots on a circle – LONGHOLE

Programming

LONGHOLE (RTP, RFP, SDIS, DP, DPR, NUM, LENG, CPA, CPO, RAD, STA1, INDA, FFD, FFP1, MID)

Parameters

RTP real Retraction plane (absolute)

RFP real Reference plane (absolute)

SDIS real Safety clearance (enter without sign)

DP real Slot depth (absolute)

DPR real Slot depth relative to the reference plane (enter without sign)

NUM integer Number of slots

LENG real Slot length (enter without sign)

CPA real Center point of circle of holes (absolute), 1st axis of the plane

CPO real Center point of circle of holes (absolute), 2nd axis of the plane

RAD real Radius of the circle (enter without sign)

STA1 real Starting angle

INDA real Incrementing angle

FFD real Feedrate for depth infeed

FFP1 real Feedrate for surface machining

MID real Maximum infeed depth for one infeed (enter without sign)

Function

Use this cycle to machine elongated holes arranged on a circle. The longitudinal axis of the slots is aligned radially.

Contrary to the slot, the width of the long hole is determined by the tool diameter.

Internally in the cycle, an optimum traversing path of the tool is determined, ruling out unnecessary idle passes. If several depth infeeds are required to machine an slot, the infeed is carried out alternately at the end points. The path to be traversed along the longitudinal axis of the slot will change its direction after each infeed. The cycle will search for the shortest path when changing to the next slot.

Slots on a circle – SLOT1

Programming

SLOT1(RTP, RFP, SDIS, DP, DPR, NUM, LENG, WID, CPA, CPO, RAD, STA1, INDA, FFD, FFP1, MID, CDIR, FAL, VARI, MIDF, FFP2, SSF)

Parameters

RTP real Retraction plane (absolute)

RFP real Reference plane (absolute)

SDIS real Safety clearance (enter without sign)

DP real Slot depth (absolute)

DPR real Slot depth relative to the reference plane (enter without sign)

NUM integer Number of slots

LENG real Slot length (enter without sign)

WID real Slot width (enter without sign)

CPA real Center point of circle of holes (absolute), 1st axis of the plane

CPO real Center point of circle of holes (absolute), 2nd axis of the plane

RAD real Radius of the circle (enter without sign)

STA1 real Starting angle

INDA real Incrementing angle

FFD real Feedrate for depth infeed

FFP1 real Feedrate for surface machining

MID real Maximum infeed depth for one infeed (enter without sign)

CDIR integer Mill direction for machining the slot

Values: 2 (for G2)

3 (for G3)

FAL real Finishing allowance at the slot edge (enter without sign)

VARI integer Machining type

Values: 0=complete machining

1=roughing

2=finishing

MIDF real Maximum infeed depth for finishing

FFP2 real Feedrate for finishing

SSF real Speed when finishing

Note

The cycle requires a milling cutter with an ”end tooth cutting across center” (DIN844).

Function

The cycle SLOT1 is a combined roughing-finishing cycle.

Use this cycle to machine slots arranged on a circle. The longitudinal axis of the slots is aligned radially. Unlike the slot, a value is defined for the slot width.

Function

The cycle SLOT1 is a combined roughing-finishing cycle.

Use this cycle to machine slots arranged on a circle. The longitudinal axis of the slots is aligned radially. Unlike the slot, a value is defined for the slot width.

Circumferential slot – SLOT2

Programming

SLOT2(RTP, RFP, SDIS, DP, DPR, NUM, AFSL, WID, CPA, CPO, RAD, STA1, INDA, FFD, FFP1, MID, CDIR, FAL, VARI, MIDF, FFP2, SSF)

Parameters

RTP real Retraction plane (absolute)

RFP real Reference plane (absolute)

SDIS real Safety clearance (enter without sign)

DP real Slot depth (absolute)

DPR real Slot depth relative to the reference plane (enter without sign)

NUM integer Number of slots

AFSL real Angle for the slot length (enter without sign)

WID real Circumferential slot width (enter without sign)

CPA real Center point of circle of holes (absolute), 1st axis of the plane

CPO real Center point of circle of holes (absolute), 2nd axis of the plane

RAD real Radius of the circle (enter without sign)

STA1 real Starting angle

INDA real Incrementing angle

FFD real Feedrate for depth infeed

FFP1 real Feedrate for surface machining

MID real Maximum infeed depth for one infeed (enter without sign)

CDIR integer Mill direction for machining the circumferential slot

Values: 2 (for G2)

3 (for G3)

FAL real Finishing allowance at the slot edge (enter without sign)

VARI integer Machining type

Values: 0 = complete machining

1 = roughing

2 = finishing

MIDF real Maximum infeed depth for finishing

Milling a rectangular pocket – POCKET3

Programming

POCKET3(_RTP, _RFP, _SDIS, _DP, _LENG, _WID, _CRAD, _PA, _PO, _STA, _MID, _FAL, _FALD, _FFP1, _FFD, _CDIR, _VARI, _MIDA, _AP1, _AP2, _AD, _RAD1, _DP1)

Parameters

_RTP real Retraction plane (absolute)

_RFP real Reference plane (absolute)

_SDIS real Safety clearance (enter without sign)

_DP real Pocket depth (absolute)

_LENG real Pocket length, for dimensioning from the corner with sign

_WID real Pocket width, for dimensioning from the corner with sign

_CRAD real Pocket corner radius (enter without sign)

_PA real Reference point for the pocket (absolute), 1st axis of the plane

_PO real Reference point for the pocket (absolute), 2nd axis of the plane

_STA real Angle between the pocket longitudinal axis and the first axis of the plane (enter without sign);

Value range: 0 . ≤_STA . 180

_MID real Maximum infeed depth (enter without sign)

_FAL real Finishing allowance at the pocket edge (enter without sign)

_FALD real Finishing allowance at the base (enter without sign)

_FFP1 real Feedrate for surface machining

_FFD real Feedrate for depth infeed

_CDIR integer Milling direction: (enter without sign)

Values: 0 Synchronous milling (according to the spindle direction)

1 Conventional milling

2 With G2 (independent of spindle direction)

3 With G3

_VARI integer Machining type

UNITS DIGIT

Values: 1 Roughing

2 Finishing

TENS DIGIT:

Values: 0 Perpendicular to the pocket center with G0

1 Perpendicular to the pocket center with G1

2 Along a helix

3 Perpediculation along a pocket longitudinal axis

The other parameters can be selected as options. Specify the plunge-cut strategy and the overlap for solid machining (to be entered without sign):

Function

The cycle can be used for roughing and finishing. For finishing, a face cutter is required. The depth infeed will always start at the pocket center point and be performed vertically from there; thus it is practical to predrill at this position.

_ The milling direction can be determined either using a G command (G2/G3) or from the spindle direction as synchronous or opposed milling.

_ For solid machining, the maximum infeed width in the plane can be programmed.

_ Finishing allowance also for the pocket base

_ There are three different insertion strategies:

– vertically to the pocket center

– along a helical path around the pocket center

– oscillating at the pocket central axis

_ Shorter approach paths in the plane for finishing

_ Consideration of a blank contour in the plane and a blank dimension at the base (optimum machining of preformed pockets possible).

Milling a circular pocket – POCKET4

Programming

POCKET4 (_RTP, _RFP, _SDIS, _DP, _PRAD, _PA, _PO, _MID, _FAL, _FALD, _FFP1, _FFD, _CDIR, _VARI, _MIDA, _AP1, _AD, _RAD1, _DP1)

Parameters

_RTP real Retraction plane (absolute)

_RFP real Reference plane (absolute)

_SDIS real Safety clearance (to be added to the reference plane; enter without sign)

_DP real Pocket depth (absolute)

_PRAD real Pocket radius

_PA real Starting point (absolute), 1st axis of the plane

_PO real Starting point (absolute), 2nd axis of the plane

_MID real Maximum infeed depth (enter without sign)

_FAL real Finishing allowance at the pocket edge (enter without sign)

_FALD real Finishing allowance at the base (enter without sign)

_FFP1 real Feedrate for surface machining

_FFD real Feedrate for depth infeed

_CDIR integer Milling direction: (enter without sign)

Values: 0 Synchronous milling (according to the spindle direction)

1 Conventional milling

2 With G2 (independent of spindle direction)

3 With G3

_VARI integer Machining type

UNITS DIGIT

Values: 1 Roughing

2 Finishing

TENS DIGIT:

Values: 0 Perpendicular to the pocket center with G0

1 Perpendicular to the pocket center with G1

2 Along a helix

The other parameters can be selected as options. Specify the plunge-cut strategy and the

overlap for solid machining (to be entered without sign):

_MIDA real Maximum infeed width as a value in solid machining in the plane

_AP1 real Pocket radius blank dimension

_AD real Blank pocket depth dimension from reference plane

_RAD1 real Radius of the helical path during insertion (referred to the tool center point path)

_DP1 real Insertion depth per 360 . revolution on insertion along helical path

Function

Use this cycle to machine circular pockets in the machining plane. For finishing, a face cutter is required.

The depth infeed will always start at the pocket center point and be performed vertically from there; thus it is practical to predrill at this position.

_ The milling direction can be determined either using a G command (G2/G3) or from the spindle direction as synchronous or opposed milling.

_ For solid machining, the maximum infeed width in the plane can be programmed.

_ Finishing allowance also for the pocket base

_ Two different insertion strategies:

– vertically to the pocket center

– along a helical path around the pocket center

_ Shorter approach paths in the plane for finishing

_ Consideration of a blank contour in the plane and a blank dimension at the base (optimum machining of preformed pockets possible).

_ _MIDA is recalculated during edge machining.

Thread milling – CYCLE90

Programming

CYCLE90 (RTP, RFP, SDIS, DP, DPR, DIATH, KDIAM, PIT, FFR, CDIR, TYPTH, CPA, CPO)

Parameters

RTP real Retraction plane (absolute)

RFP real Reference plane (absolute)

SDIS real Safety clearance (enter without sign)

DP real Final drilling depth (absolute)

DPR real Final drilling depth relative to the reference plane (enter without sign)

DIATH real Nominal diameter, outer diameter of the thread

KDIAM real Core diameter, internal diameter of the thread

Function

By using the cycle CYCLE90, you can produce internal or external threads. The path when milling threads is based on a helix interpolation. All three geometry axes of the current plane, which you will define before calling the cycle, are involved in this motion.

Sequence when producing an external thread

Position reached prior to cycle start:

The starting position is any position from which the starting position at the outside diameter of the thread at the height of the retraction plane can be reached without collision.

This start position for thread milling with G2 lies between the positive abscissa and the positive ordinate in the current level (i.e., in the 1st quadrant of the coordinate system). For thread milling with G3, the start position lies between the positive abscissa and the negative ordinate (i.e., in the 4th quadrant of the coordinate system).

8.4 Arithmetic Parameters R

Functionality

The arithmetic parameters are used if an NC program is not only to be valid for values assigned once, or if you must calculate values. The required values can be set or calculated by the control system during program execution.

The arithmetic parameter values can also be set by operator inputs. If values have been assigned to the arithmetic parameters, they can be assigned to other variable-setting NC addresses in the program.

Programming

R0 = ... bis R299 = ... ; Assign values to the R parameters

R[R0] = ... ; Indirect programming: Assign a value to the R parameter whose number can be found, e.g. in R0

X = R0 ; Assign arithmetic parameters to the NC addresses, e.g. for the X axis

Value assignment

You can assign values in the following range to the R parameters:

_(0.000 0001 ... 9999 9999)

(8 decimal places, arithmetic sign and decimal point)

The decimal point can be omitted for integer values. A plus sign can always be omitted.

Example:

R0 = 3.5678 R1 = –37.3 R2 = 2 R3 = –7 R4 = –45678.123

Use the **exponential notation** to assign an extended range of numbers:

_ (10–300 ... 10+300).

The value of the exponent is written after the **EX** characters; maximum total number of characters: 10 (including leading signs and decimal point)

Range of values for EX: –300 to +300

Example:

R0 = –0.1EX–5 ; Meaning: R0 = –0.000 001

R1 = 1.874EX8 ; Meaning: R1 = 187 400 000

Remark: There can be several assignments in one block incl. assignments of arithmetic expressions.

8.5 Local User Data

Local User Data (LUD)

Functionality

The operator/programmer (user) can define his/her own variable in the program from various data types (LUD = Local User Data). These variables are only available in the program in which they were defined. The definition takes place immediately at the start of the program and can also be associated with a value assignment at the same time. Otherwise the starting value is zero.

The name of a variable can be defined by the programmer. The naming is subject to the following rules:

_ A maximum of 32 characters can be used.

_ It is imperative to use letters for the first two characters; the remaining characters can be either letters, underscore or digits.

_ Do not use a name already used in the control system (NC addresses, keywords, names of programs, subroutines, etc.).

Programming / data types

DEF BOOL varname1 ; ”Bool” type, values: TRUE (= 1), FALSE (= 0)

DEF CHAR varname2 ; ”Char” type, 1 character in the ASCII code: ”a”, ”b”, ...

; Numerical code value: 0 ... 255

DEF INT varname3 ; Integer type, integer values, 32–bit value range:

; –2 147 483 648 ... +2 147 483 648 (decimal)

DEF REAL varname4 ; ”Real” type, natural number (as witrh R parameter),

; Value range: _(0.000 0001 ... 9999 9999)

; (8 decimal places, arithmetic sign and decimal point) or

; exponential notation: _ (10–300 ... 10+300)

DEF STRING[*string length*] varname41 ; STRING type, [*string length*]: Maximum number of characters

Each data type requires its own program line. However, several variables of the same type can be defined in one line.

Example:

DEF INT PVAR1, PVAR2, PVAR3 = 12, PVAR4 ; 4 variables of the INT type

Example for STRING type with assignment:

DEF STRING[12] PVAR = ”Hello” ; Define PVAR variable with maximum

string length 12 and character

sequence

Hello

Fields

In addition to the individual variables, one or two-dimensional fields of variables of these data types can also be defined:

DEF INT PVAR5[n] ; Single-dimensionsal field of INT type, n: integer

DEF INT PVAR6[n,m] ; Two-dimensional field of the INT type, n, m: integer

Example:

DEF INT PVAR7[3] ; Field with 3 elements of the INT type

Within the program, the individual field elements can be reached via the field index and can be treated like individual variables. The field index runs from 0 to a small number of the elements.

Example:

N10 PVAR7[2] = 24 ; The third field element (with index 2) is assigned the value 24.

Value assignment for field with SET instruction:

N20 PVAR5[2] = SET(1,2,3) ; Starting with the 3rd field element, different values are assigned.

Value assignment for field with REP instruction:

N20 PVAR7[4] = REP(2) ; Starting from the field element [4], all values are assigned the same value, here 2.

Jump destination for program jumps

Functionality

A **label** or a **block number** serve to mark blocks as jump destinations for program jumps. Program jumps can be used to branch to the program sequence.

Labels can be freely selected, but must contain a minimum of 2 and a maximum of 8 letters or numbers, and the **first two characters** must be **letters** or underscores.

Labels that are in the block that serves as the jump destination are **ended by a colon**.

They are always at the start of a block. If a block number is also present, the label is located **after the block number**.

Labels must be unique within a program.

Programming example

N10 LABEL1: G1 X20 ; LABEL1 is the label, jump destination

...

TR789: G0 X10 Z20 ; TR789 is the label, jump destination

– No block number existing

N100 ... ; A block number can be a jump destination.

CHAPTER 9 SINUMERIK 802S/c programme

9.1 Position

Absolute/incremental dimensions:G90/G91

1. **Functionality**

When instruction G90 or G91 is active, the specified position information X, Z is interpreted as a coordinate point (G90) or as an axis path to be traversed (G91). G90/G91 applies to all axes. These instructions do not determine the actual path on which the end points are reached. This is done by a G group

2. Programming

G90 absolute **dimension**

G91 **Incremental dimension**

X=AC（…）X axis programming in according to absolute **dimension**

X=IC（…）X axis programming in according to **Incremental dimension**

Absolute dimension G90:

When absolute dimensioning is selected, the dimension data refer to the zero point of the currently active coordinate system (workpiece coordinate system, current workpiece coordinate system or machine coordinate system). Which of the systems is active depends on which offsets are currently effective, i.e. programmable, settable or none at all.

G90 is active for all axes on program start and remains so until it is deactivated by G91 (incremental dimensioning selection) in a subsequent block (modal command).

Incremental dimension G91:

When incremental dimensioning is selected, the numerical value in the poison information corresponds to the path to be traversed by an axis. The traversing direction is determined by the sign.

G91 applies to all axes and can be deactivated by G90 (absolute dimensioning) in a later block.

3. example for G90 and G91 programming

```
N10 G90 X20 Z90          ;Absolute dimensioning
N20 X75 Z-32             ;Absolute dimensioning still active
…
N180 G91 X40 Z20         ; Switchover to incremental dimensioning
N190 X-12 Z17            ;Incremental dimensioning still active
```

Radius/diameter dimensions: G22/G23

1. **Functionality**

When parts are machined on turning machines, it is normal practice to program the position data for the X axis (facing axis) as a diameter dimension.

The specified value is interpreted as a diameter for this axis only by the control. It is possible to switch over to radius dimension in the program if necessary.

2. **Programming**

G22 Radius dimension

G23 Diameter dimension

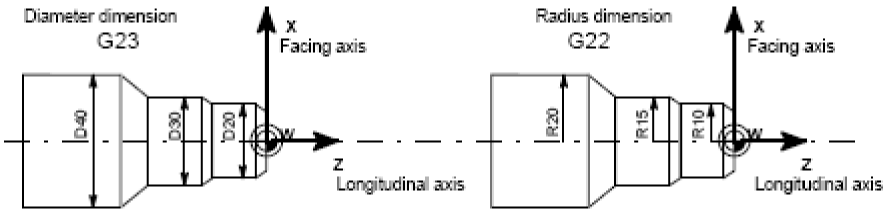


Fig 9.1-1

Information

When G22 or G23 is active, the specified end point for the X axis is interpreted as a radius or diameter dimension.

The actual value is displayed correspondingly in the workpiece coordinate system. A programmable offset with G158 X... is always interpreted as a radius dimension. See the following section for a description of this function.

3. **Programming example**

N10 G23 X44 Z30 ;Diameter for X axis

N20 X48 Z25 ;G23 still active

N30 Z10

...

N110 G22 X22 Z30 ;Changeover to radius dimension for X axis from here

N120 X24 Z25

N130 Z10

...

Programmable zero offset:G158

1. **Functionality**

Use the programmable zero offset for frequently repeated shapes/arrangements in different positions on a workpiece or when you simply wish to choose a new reference point for the dimension data. The programmable offset produces the current workpiece coordinate system. The newly programmed dimension data then refer to this system. The offset can be applied in all axes. A separate block is always required for the G158 instruction.

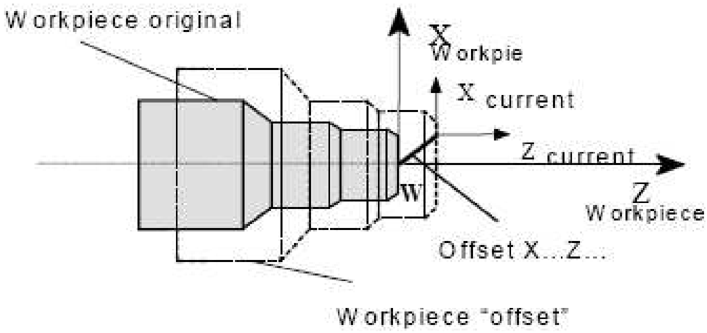


Fig 9.1-2

2. **Offset G158**

A zero offset can be programmed for all axes with instruction G158. A newly entered G158 instruction replaces any previous programmable offset instruction.

3. **Delete offset**

If the instruction G158 without axes is inserted in a block , then any active programmable offset will be deleted.

4. **Programming Example**

N10 ...

N20 G158 X3 Z5 ;Programmable offset

N30 L10 ;Subroutine call, contains the geometry to be offset

...

N70 G158 ;Offset deleted

...

Workpiece clamping - settable zero offset: G54 to G57, G500, G53

1. **Functionality**

The settable zero offset specifies the position of the workpiece zero point on the machine (offset between workpiece zero and machine zero). This offset is calculated when the workpiece is clamped on the machine and must be entered by the operator in the data field provided. The value is activated by the program through selection from four possible groups: G54 to G57.

2. **Programming**

G54 ;1st settable zero offset

G55 ;2nd settable zero offset

G56 ;3rd settable zero offset

G57 ;4th settable zero offset

G500 ;Settable zero offset OFF modal

G53 ;Settable zero offset OFF non-modal, also suppresses programmable offset

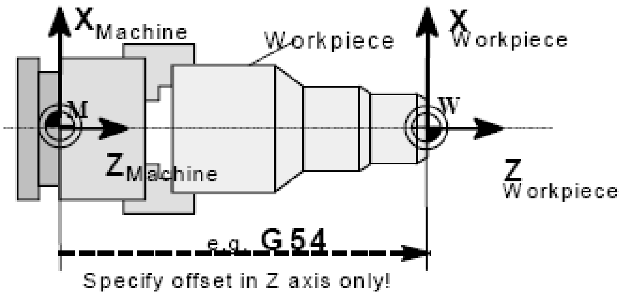


Fig 9.1-3

3. **Programming Example**

N10 G54 ... ;Call first settable zero offset

N20 X... Z... ;Machine workpiece

...

N90 G500 G0 X... ;Deactivate settable zero offset

9.2 G Commands

9.2.1 Linear interpolation at rapid traverse:

Functionality

The rapid traverse motion G0 is used to position the workpiece rapidly, but not to machine the workpiece directly. All axes can be traversed simultaneously resulting in a linear path.

The maximum speed (rapid traverse) for each axis is set in the machine data. If only one axis is moving, it traverses at its own rapid traverse setting. If two axes are traversed simultaneously, then the path speed (resultant speed) is selected so as to obtain the maximum possible path speed based on the settings for both axes.

A programmed feed (F word) is irrelevant for G0. G0 remains effective until it is canceled by another instruction from the same group (G1, G2, G3,...).

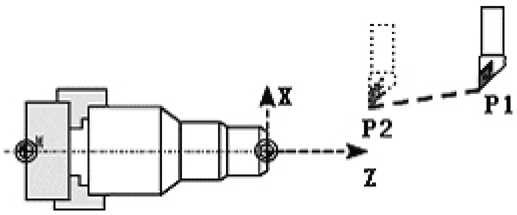


Fig 9.2-1

Programming example

N10 G0 X100 Y150 Z65 ; Cartesian coordinate

N50 G0 RP=16.78 AP=45 ; Polar coordinate

Information

Another group of G functions exists for movement to the position .For G60 exact stop, a window with various precision values can be selected with another G group. For exact stop, an alternative instruction with non-modal effectiveness exists: G9.

You should consider these options for adaptation to your positioning tasks.

9.2.2 Positional data

G01 Linear interpolation

Functionality

The tool moves from the start point to the end point along a straight path. The path speed is defined by the programmed F word. All axes can be traversed simultaneously.

G1 remains effective until it is canceled by another instruction from the same G group (G0, G2, G3, ...).

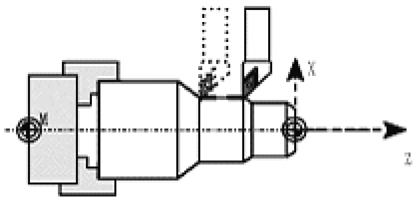


Fig 9.2-2

Programming example

N05 G54 G0 G90 X40 Z200 S500 M3 ;tool is moving at rapid traverse, spindle speed = 500 rpm, CW rotation

N10 G1 Z120 F0.15 ;Linear interpolation with feed 0.15 mm/rev

N15 X45 Z105

N20 Z80

N25 G0 X100 ;Traverse clear at rapid traverse

N30 M2 ;End of program

G02/G03 Circular interpolation

1.**Functionality**

The tool moves from the start point to the end point on a circular path. The direction is determined by the G function:

G2 - in clockwise direction

G3 - in counterclockwise direction

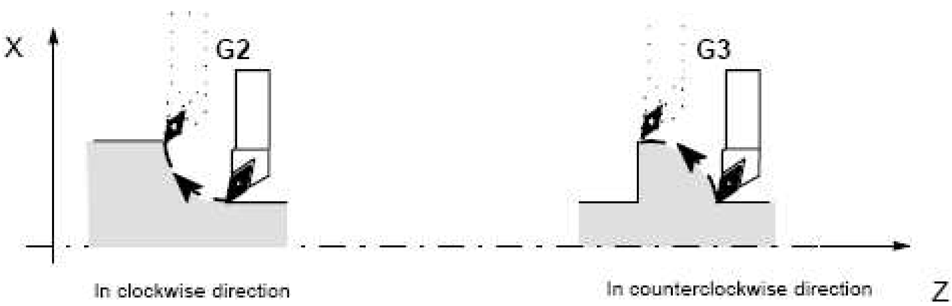


Fig 9.2-3

G2/G3 remain effective until they are canceled by another instruction from the same G group (G0, G1, ...).

Note： The required cycle can be described in different ways:

_ Center point and end point

_ Circle radius and end point

_ Center point and aperture angle

2**. Programming**

G2/G3 X... Y... I... J... ; Center and end points

G2/G3 CR=... X... Y... ; Circle radius and end point

G2/G3 AR=... I... J... ; Aperture angle and center point

G2/G3 AR=... X... Y... ; Aperture angle and end point

G2/G3 AP=... RP =... ; Polar coordinates, circle around the pole

Further possibilities for circle programming result from:

CT – circle with tangential connection and

CIP – circle via intermediate point (see next sections).

3. **Programming example**

Center point and end point specification:

N5 G90 Z30 X40 ;Circle start point for N10

N10 G2 Z50 X40 K10 I-7 ;End point and center point

End point and radius specification

N5 G90 X30 Y40 ; Circle starting point for N10

N10 G2 X50 Y40 CR=12.207 ; End point and radius

Note: With a negative leading sign for the value with CR=–..., a circular segment larger than a semi-circle is selected.

End point and aperture angle:

N5 G90 Z30 X40 ;Circle start point for N10

N10 G2 Z50 X40 AR=105 ;End point and aperture angle

Center point and aperture angle:

N5 G90 Z30 X40 ;Circle start point for N10

N10 G2 K10 I-7 AR=105 ;Center point and aperture angle

G05 Circular interpolation via intermediate point

1. **Functionality**

If you know three contour points around the circle instead of center point or
radius or aperture angle, you should preferably use the G5 function.

The direction of the circle in this case is determined by the position of the intermediate point (between start and end positions).

G5 remains effective until it is canceled by another instruction from the same G group (G0, G1, G2, ...).

Note: The dimension setting G90 or G91 applies to both the end point **and** intermediate point!

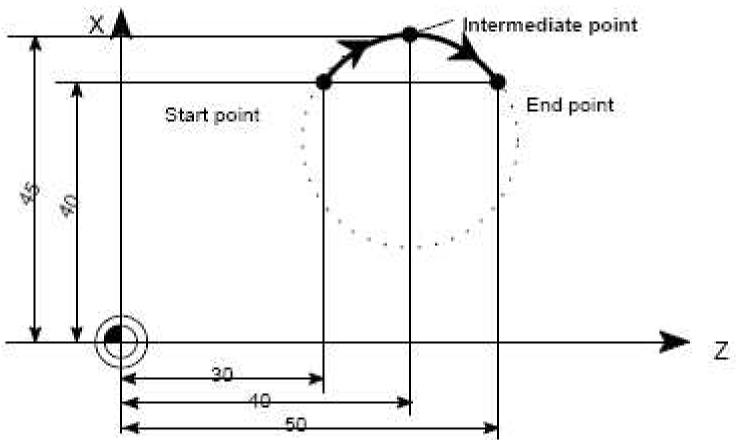


Fig 9.2-4

2. **Programming example**

N5 G90 Z30 X40 ;Circle start point for N10

N10 G5 Z50 X40 KZ=40 IX=45 ;End and intermediate points (XI must be programmed as a radius dimension)

G33 Thread cutting with constant lead:

1. **Functionality**

Function G33 can be used to cut the following types of threads with constant lead:

- z Thread on cylindrical bodies
- z Thread on tapered bodies
- z External/internal threads
- z Single-start/multiple-start threads
- z Multi-block threads (thread “chaining”)

G group (G0, G1, G2,G3,...).

2. **Prerequisite** This requires a spindle with position measuring system

G33 remains effective until it is canceled by another instruction from the same

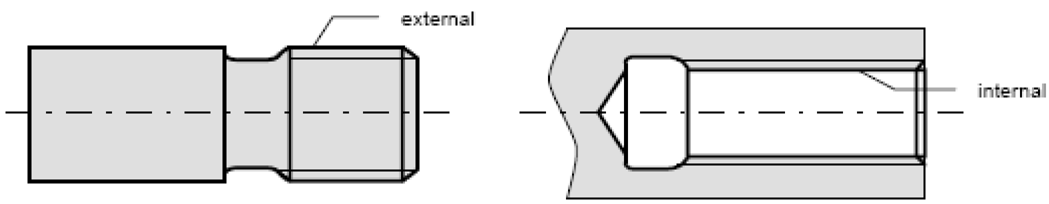


Fig 9.2-5

3. **Right-hand or left-hand threads**

The direction of the thread, i.e. right-hand or left-hand, is determined by the setting for the direction of rotation of the spindle (M3 - clockwise rotation, M4 - counterclockwise rotation). To this aim,the speed setting must be programmed under address S, or a speed must be set.

Note: The approach and run-out paths must be taken into account with respect to the thread length.

In the case of tapered threads (2 axes must be specified), the lead address I or K of the axis with the longer path (greater thread length) must be used. A second lead is not specified.

4. **Start-point offset** SF

A start-point offset of the spindle is required for machining multiple-start threads or threads in offset cuts. The start-point offset is programmed under address SF in the thread block with G33 (absolute position).

If a start point is not included in the block, the value from the setting data is activated.

Note: Any value programmed for SF= is always entered in the setting data as well.

5. **Programming example**

Cylindrical thread, two-start, start-point offset 180 degrees, thread length (including approach and run-out) 100 mm, thread lead 4 mm/rev.

RH thread, cylinder premachined:

```
N10 G54 G0 G90 X50 Z0 S500 M3      ;Approach start point, CW spindle rotation
N20 G33 Z-100 K4 SF=0 ;Lead        :4 mm/rev.
N30 G0 X54
N40 Z0
N50 X50
N60 G33 Z-100 K4 SF=180             ;2nd start, 180 degrees offset
N70 G0 X54 ...
```

G75 Fixed point approach

1. **Functionality**

By using G75, a fixed point on the machine, e.g. tool change point, can be approached. The position is stored permanently in the machine data for all axes. No offset is effective. The speed of each axis is its rapid traverse.

G75 requires a separate block and is non-modal. The machine axis identifier must be programmed!

In the block after G75, the previous G command of the ”Interpolation type” group (G0, G1,G2, ...) is active again.

2. **Programming example**

N10 G75 X0 Z0

Remark: The programmed position values for X, Z (any value, here = 0) are ignored, but must still be written.

G74 Reference point approach

1. **Functionality**

The reference point can be approached in the NC program with G74. The direction and speed of each axis are stored in machine data.

G74 requires a separate block and is non-modal. The machine axis identifier must be programmed!

In the block after G74, the previous G command of the ”Interpolation type” group (G0, G1,G2, ...)

is active again.

2. **Programming example**

N10 G74 X0 Z0

Remark: The programmed position values for X, Z (any value, here = 0) are ignored, but must still be written.

G9/G60/G64 Exact stop / continuous-path control mode

1. **Functionality**

G functions are provided for optimum adaptation to different requirements to set the traversing behavior at the block borders and for block advancing. Example: For example, you would like to quickly position with the axes or you would like to machine path contours over multiple blocks.

2. **Programming**

G60 ;Exact stop – modal

G64 ;Continuous-path control mode

G9 ; Exact stop – non-modal

G601 ; Exact stop window fine

G602 ; Exact stop window coarse

3. **exact stop fine G60,G9**

If the exact stop function (G60 or G9) is active, the velocity for reaching the exact end position at the end of a block is decelerated to zero.

Another modal G group can be used here to set when the traversing movement of this block is considered ended and the next block is started.

* G601 ; Exact stop window fine

Block advance takes place when all axes have reached the ”Exact stop window fine” (value in the machine data).

* G602 ; Exact stop window coarse

Block advance takes place when all axes have reached the ”Exact stop window coarse” (value in the machine data).

e selection of the exact stop window has a significant influence on the total time if many positioning operations are executed. Fine adjustments require more time.

4. **Programming example**

N5 G602 ; Exact stop window coarse

N10 G0 G60 X... ; Exact stop modal

N20 X... Y... ; G60 remains active

...

N50 G1 G601 .. . ; Exact stop window fine

N80 G64 X.. . ; Switching to continuous-path control mode

...

N100 G0 G9 X.. . ; Exact stop is only effective for this block

N111 .. ; Continuous-path control mode again

Remark: The G9 command only generates exact stop for the block in which it is programmed; G60, however, is effective until it is canceled by G64.

5. **Continuous-path control mode G64**

The objective of the continuous-path control mode is to avoid deceleration at the block boundaries and to switch **to the next block** with **a path velocity as constant as possible** (in the case of tangential transitions). The function works with **look-ahead velocity control** over several blocks. For non-tangential transitions (corners), the velocity can reduced rapidly enough so that the axes are subject to a relatively high velocity change over a short time. This may lead to a significant jerk (acceleration change). The size of the jerk can be limited by activating the SOFT function.

6. **Programming example**

N10 G64 G1 X... F... ; Continuous-path control mode

N20 Y. . ; Continuous-path control mode continues to be active

...

N180 G60 ... ; switching to exact stop

G4 Dwell **Time**

1. **Functionality**

Between two NC blocks, you can interrupt the machining for a defined time by inserting a **separate block** with G4,The words with F... or S... are only used in this block for the specified time. Any previously programmed feedrate F or a spindle speed S remain valid.

2. Programming

G4 F... ; Dwell time in s

G4 S... ; Dwell time in spindle revolutions

3. **Programming example**

N5 G1 F200 Z-50 S300 M3 ; Feedrate F, spindle speed S

N10 G4 F2.5 ; Dwell time 2.5 s

N20 Z70

N30 G4 S30 ; Dwell for 30 spindle revolutions; corresponds to S=300 r.p.m., and100 % speed override: t=0.1 min

N40 X... ; Feedrate and spindle speed continue to be effective

Remark

G4 S.. is only possible if a controlled spindle is available (if the speed specifications are also programmed via S...).

F Feedrate

1. **Functionality**

The feed F is the **path velocity** and represents the value of the geometric sum of the velocity components of all axes involved. The individual axis velocities therefore result from the portion of the axis path in the overall distance to be traversed.

The feedrate F is effective for the interpolation types G1, G2, G3, and G5 and is retained until a new F word is written.

2. Programming

F...

Remark**:**

For **integer values**, the decimal point is not required, e.g. F300.

Unit of measure for F with G94, G95

The dimension unit for the F word is determined by G functions:

_ G94 **F** as the feedrate in**mm/min**

_ G95 **F** as the feedrate in **mm/rev.** of the spindle (only meaningful when the spindle is running)

Remark**:**

This unit of measure applies to metric dimensions. According to Section ”Metric and inch dimensioning”, settings with inch dimensioning are also possible.

3. **Programming example**

N10 G94 F310 ; Feedrate in mm/min

...

N110 S200 M3 ; Spindle rotation

N120 G95 F15.5 ; Feedrate in mm/rev.

Remark: Write a new F word if you change G94 – G95.

S spindle speed/direction of rotation

1. **Functionality**

The spindle speed is programmed in r.p.m. under the address S provided that the machine possesses a controlled spindle.

The direction of rotation and the start or end of the movement are specified via M commands (also see Section 8.7 ”Miscellaneous function M”).

M3 ; Spindle CW rotation

M4 ; Spindle CCW rotation

M5 ; Spindle stop

Remark: For integer S values, the decimal point can be omitted, e.g. S270

Information If you write M3 or M4 in a **block with axis movements**, the M commands become active **before** the axis movements.

2. **Programming example**

N10 G1 X70 Z20 F300 S270 M3 ; Spindle accelerates CW to 270 r.p.m.**before** traversing of the X, Z axes...

N80 S450 .. . ; Speed change ...

N170 G0 Z180 M5 ; Z motion, spindle stops

G25/G26 main spindle speed limitation

1. **Functionality**

In the program, you can limit the limit values that would otherwise apply for a controlled spindle by writing G25 or G26 and the spindle address S with the speed limit value.This overwrites the values entered in the setting data at the same time.

G25 and G26 each require a separate block. A previously programmed speed S is maintained.

2. Programming

G25 S…limits the main spindle lower speed value

G26 S… limits the main spindle upper speed value.

1. **Information：** The outmost limits of the spindle speed are set in machine data. Appropriate inputs via the operator panel can activate various setting data for further limiting.

2. **Programming example**

N10 G25 S12 ; Lower spindle limit speed: 12 r.p.m.

N20 G26 S700 ; Upper spindle limit speed : 700 r.p.m.

SPOS Spindle positioning

1. **Functionality**

Prerequisite: The spindle must be technically designed for position control. With the function SPOS = you can position the spindle in a specific **angular position**. The spindle is held in the position by position control.

The **speed** of the positioning procedure is defined in machine data. With SPOS = **value** from the M3/M4 movement, the respective **direction of rotation** is maintained until the end of the positioning. When positioning from standstill, the position is approached via the shortest path. The direction results from the respective starting and end position.

Exception: First movement of the spindle, i.e. if the measuring system is not yet synchronized. In this case, the direction is specified in machine data.

Other movement specifications for the spindle are possible with SPOS = ACP (...), SPOS = ACN (...), ... as for rotary axes (see Section ”4th axis”). The spindle movement takes place parallel to any other axis movements in the same block. This block is ended when both movements are finished.

2. **Programming**

SPOS = ... ; Absolute position: 0 ... <360 degrees

3. **Programming example**

N10 SPOS = 14.3 ; Spindle position 14.3 degrees

...

N80 G0 X89 Z300 SPOS = 25.6 ; Positioning of the spindle with axis movements ; The block is only completed if all movements are performed.

N81 X200 Z300 ; The N81 block will only start if the spindle position from N80; is reached.

T Tool

1. **Functionality**

You select a tool by programming the T word. A machine data defines whether the T word represents a tool change or merely a preselection.

- z Tool change (tool call) is implemented directly by T word (e.g. normal practice for tool revolver on turning machines) or
- z the tool is changed through additional instruction M6 after preselection by T word (see also Section "Miscellaneous Functions M").

Please note:

If a certain tool has been activated, this will remain stored as the active tool even across the program end and after POWER ON of the control system. If you change a tool manually, then enter the change into the control system also manually to make sure that the control system detects the right tool. For example, you can start a block with a new T word in the MDA mode.

2. **Programming**

T... ;Tool number: 1 ... 32 000

Note A maximum of 15 tools can be stored in the control at a time.

3. **Programming example**

; Tool change without M6**:**

N10 T1 ; Tool 1

...

N70 T588 ; Tool 588

; Tool change with M6:

N10 T14 ... ; Preselect tool 14

...

N15 M6 ; Perform tool change; thereafter, T14 is active

D Tool offset number

1. **Functionality**

You can assign between 1 and 9 data fields with various tool offset blocks (for several tool edges) to each specific tool. If a special edge is required, it can be programmed by means of D plus a corresponding number.

D1 is the automatic default if no D word is programmed. When D0 is programmed, then the offsets for the tool are not active.

Note: A maximum of 30 data fields with tool offset blocks can be stored in the control at a time.

2. **Programming**

D... ;Tool offset number: 1 ... 9

D0 : No offsets active

Information： Tool length compensations take immediate effect when the tool is active. The values of D1 are applied if no D number has been programmed. The tool length is compensated when the first programmed traversal of the relevant length compensation axis is executed.

A tool radius compensation must also be activated by means of G41/G42.

3. **Programming example**

Tool change **without M6 command** (only with T)**:**

N5 G17 ; Determines the axis assignment for compensations

N10 T1 ; Tool 1 is activated with the appropriate D1

N11 G0 Z... ; With G17, Z is the length compensation axis, the length offset compensation ;is overlaid here

N50 T4 D2 ; Load tool 4, D2 from T4 active

...

N70 G0 Z... D1 ; D1 for tool 4 active; only cutting edge changed Tool change using the **M6** **;** command:

N5 G17 ; Determines the axis assignment for compensations

N10 T1 ; Tool preselection

...

N15 M6 ; Tool change, T1 is active with the appropriate D1

N16 G0 Z... ; With G17, Z is the length compensation axis, the length offset compensation ;is overlaid here

...

N20 G0 Z... D2 ; D2 for tool 1 is active; with G17, Z is the length compensation axis,the ;difference of the length compensation D1–>D2 is overlaid here

N50 T4 ; Preselection of tool T4; please observe:T1 with D2 is still active !

...

N55 D3 M6 ; Tool change, T4 with the appropriate D3 is active

G41/G42 Selection of tool radius compensation

1. **Functionality**

compensation (tool nose radius compensation) is activated by G41/G42. The control then automatically calculates the necessary tool paths equidistant from the programmed contour for the current tool radius.

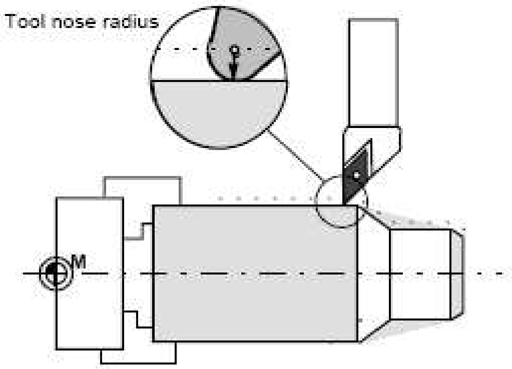


Fig 9.2-6

2. **Programming**

G41 X... Z... ;Tool radius compensation to left of contour

G42 X... Z... ;Tool radius compensation to right of contour

Note: You may only select the function for linear interpolation (G0, G1).

Program both axes. If you only specify one axis, then the last programmed value is automatically set for the second axis.

3. **Programming**

N10 T...

N20 G17 D2 F300 ; Offset no. 2, feedrate 300 mm/min

N25 X... Y. .. ; P0 – starting point

N30 G1 G42 X... Y... ; Selection right of the contour, P1

N31 X... Y.. . ; Starting contour, circle or straight line

After the selection, it is also possible to execute blocks that contain infeed motions or M outputs:

N20 G1 G41 X... Y... ; Selection left of the contour

N21 Z... ; Infeed motion

N22 X... Y... ; Starting contour, circle or straight line

G40 Tool radius compensation OFF

1. **Functionality**

The compensation mode (G41/G42) is deselected with G40. G40 is also the activation position at the beginning of the program.

The tool ends the **block in front of G40** in the normal position (compensation vector vertically to the tangent at the end point);

If G40 is active, the reference point is the tool center point. Subsequently, when deselected, the tool tip approaches the programmed point.

Always select the end point of the G40 block such that collision-free traversing is guaranteed!

2. **Programming**

G40 X... Y... ; Tool radius compensation OFF

Remark: The compensation mode can only be deselected with linear interpolation (G0, G1).

3. **Programming example**

N100 X... Y...   ; Last block on the contour, circle or straight line, P1

N110 G40 G1 X... Y..   ; Deactivate tool radius compensation, P2

Subroutine

Programming example

Main：LF10.MPF

```
G54 T1 D0 G90 G00 X60 Z10
S800 M03
G01 X70 Z8 F0.1
X-2
G0 X70
L10 P3              ; Call subroutine L10.SPF 3 times
G0Z50
M05
M02
```

subroutine：L10.SPF

```
M03S600       ;     subroutine directory
G01 G91 X-25 F0.1
X6 Z-3
Z-23.5
X15 Z-20.5
G02 X0 Z-71.62 CR=55
G03 X0 Z-51.59 CR=44
G01 Z-6.37
X14
X6 Z-3
Z-12
X10
X-32 Z194
G90
M02                 ;return
```

9.3 CYCLES

Cycles are process-related subroutines that support general implementation of specific machining processes such as, for example, drilling, stock removal or thread cutting. The cycles are adapted to the specific problem in hand by means of supply parameters.

Standard cycles for turning and milling applications are provided in the system.

Standard cycles for turning

1. **Overview of cycles**

LCYC82 Drilling, spot facing

LCYC83 Deep hole drilling

LCYC840 Tapping with compensating chuck

LCYC84 Tapping without compensating chuck

LCYC85 Boring_1

2. **Defining parameters**

The arithmetic parameters from R100 to R149 are used as supply parameters for the cycles. Before a cycle is called, values must be assigned to its transfer parameters. Any parameters not needed must be loaded with zero. The values of these transfer parameters are unchanged after the cycle has been executed.

3. **Arithmetic parameters**

The cycles use the parameters R250 to R299 as internal arithmetic parameters. These are deleted when calling the cycles.

4. **Call and return** conditions

The drilling cycles are programmed independently of the particular axis names. The drilling position must be approached prior to calling the cycle in the higher-level program. The required values for feed, spindle speed and direction of rotation of the spindle must be programmed in the part program, if there are no supply parameters in the drilling cycle.

G0 G90 G40 are always effective at the end of a cycle.

5. **Recompilation** of cycles

The cycle can only be recompiled if the set of parameters stands immediantely before the cycle call. The parameters may not be separated by NC statements or comments.

6. Plane **definition**

All drilling and milling cycles assume that the current workpiece coordinate system in which machining is to be performed is defined by selecting a plane G17, G18 or G19 and activating a programmed frame (zero offset, rotation).

The drilling axis is always the 3rd axis of this system. Prior to the call, a tool with tool offset of this plane must be active. This remains active even after the cycle has been completed.

LCYC82 Drilling, spot facing

1. **Function**

The tool drills with the spindle speed and feedrate programmed down to the entered final depth. When the final drilling depth is reached, a dwell time can be programmed. The drill is retracted from the drill hole at rapid traverse rate.

2. **Call**

LCYC82

3. **Precondition**

The spindle speed and the direction of rotation, as well as the feed of the drilling axis must be defined in the higher-level program.

The drilling position must be approached before calling the cycle in the higherlevel program.

The required tool with tool offset must be selected before calling the cycle.

4. **Parameters**

Parameter	Meaning, Value Range
R101	Retract plane (absolute)
R102	Safety clearance
R103	Reference plane (absolute)
R104	Final drilling depth (absolute)
R105	Dwell time in seconds

Information：

R101 The retract plane determines the position of the drilling axis at the end of the cycle.

R102 The safety clearance acts on the reference plane, i.e. the reference plane is shifted forward by an amount corresponding to the safety clearance.

The direction in which the safety clearance acts is automatically determined by the cycle.

R103 The starting point of the drill hole shown in the drawing is programmed under the reference plane parameter.

R104 The drilling depth is always programmed as an absolute value with refer to workpiece zero.

R105 The dwell time at drilling depth (chip breakage) is programmed in seconds under R105.

5. **Motional sequence**

Position reached prior to beginning of cycle:last position in the higher-level program (drilling position)

The cycle produces the following motional sequence:

1) Approach reference plane shifted forward by an amount corresponding to the safety clearance using G0.

2) Traverse to final drilling depth with G1 and the feedrate programmed in the higher-level program.

3) Execute dwell time to final drilling depth.

4) Retract to retract plane with G0.

5. Example

```
N10 G0 G17 G90 F500 T2 D1 S500 M4    ; Define technology values
N20 X24 Y15                          ; Approach drilling position
N30 R101=110 R102=4 R103=102 R104=75 ; Supply parameters
N35 R105=2                           ; Supply parameters
N40 LCYC82                           ; Call cycle
N50 M2                               ; End of program
```

CYCLE83 Deep hole drilling

1. **Function**

The deep-hole drilling cycle produces center holes down to the final drilling depth by repeated, step-by-step deep infeed whose maximum amount can be parameterized. The drill can be retracted either to the reference plane for swarf removal after each infeed depth or by 1 mm in each case for chip breakage.

2. **Call**

LCYC83

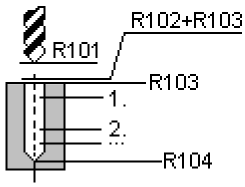
3. **Precondition**

The spindle speed and the direction of rotation must be defined in the higherlevel program.

The drilling position must be approached before calling the cycle in the higherlevel program.

Before calling the cycle, a tool offset for the drill must be selected.

4. **Parameters**



Parameter	Meaning, Value Range
R101	Retract plane (absolute)
R102	Safety clearance, enter without sign
R103	Reference plane (absolute)
R104	Final drilling depth (absolute)
R105	Dwell time to drilling depth (chip breakage)
R107	Feed for drilling
R108	Feed for first drilling depth
R109	Dwell time at starting point and for swarf removal
R110	First drilling depth(absolute)
R111	Absolute degression, enter without sign
R127	Machining type:Chip breakage = 0Swarf removal = 1

Note：

Information

R101 The retract plane determines the position of the drilling axis at the end of the cycle. The cycle is programmed on the assumption that the retract plane positioned in front of the reference plane, i.e. its distance to the final depth is greater.

R102 The safety clearance acts on the reference plane, i.e. the reference plane is shifted forward by an amount corresponding to the safety clearance.

The direction in which the safety clearance acts is automatically determined by the cycle.

R103 The starting point of the drill hole shown in the drawing is programmed under the reference plane parameter.

R104 The drilling depth is always programmed as an absolute value regardless of how G90/91 is set prior to cycle call.

R105 The dwell time at drilling depth (chip breakage) is programmed in seconds under R105.
R107, R108 The feed for the first drilling stroke (under R108) and for all subsequent drilling strokes (under R107) are programmed via these parameters.
R109 A dwell time at the starting point can be programmed in seconds under parameter R109. The dwell time at the starting point is executed only for the “with swarf removal” variant.
R110 Parameter R110 determines the depth of the first drilling stroke.
R111 Parameter R111 for the degression value determines the amount by which the current drilling depth is reduced with subsequent drilling strokes. The second drilling depth corresponds to the stroke of the first drilling depth minus the absolute degression value provided that this value is greater than the programmed absolute degression value. Otherwise, the second drilling depth also corresponds to the absolute degression value.
The next drilling strokes correspond to the absolute degression value provided that the remaining degression depth is still greater than twice the absolute degression value. The remainder is then distributed evenly between the last two drilling strokes.
If the value for the first drilling depth is in opposition to the total drilling depth, the error message 61107 “First drilling depth incorrectly defined” is displayed, and the cycle is not executed.
R127 Value 0: The drill travels 1 mm clear for chip breakage after it has reached each drilling depth. Value 1: The drill travels to the reference plane, which is shifted forward by an amount corresponding to the safety clearance for swarf removal after each drilling depth.
5. **Motional sequence**
Position reached prior to beginning of cycle:
last position in the higher-level program (drilling position)
The cycle produces the following motional sequence:
1) Approach reference plane shifted forward by an amount corresponding to the safety clearance using G0.
2) Traverse to first drilling depth with G1; the feedrate results from the feedrate programmed prior to cycle call after it has been computed with the setting in parameter R109 (feedrate factor).
Execute dwell time at drilling depth (parameter R105).
With chip breakage selected: Retract by 1 mm from the current drilling depth with G1 for chip breakage.
With swarf removal selected:
Retract for swarf removal to reference plane shifted forward by an amount corresponding to the safety clearance with G0 for swarf removal, executing the dwell time at starting point (parameter R106), approach last drilling depth minus clearance distance calculated in the cycle using G0,
3) Traverse to next drilling depth with G1 and the programmed feed; this motional sequence is continued as long as the final drilling depth is reached.
4) Retract to retract plane with G0.
5. Example

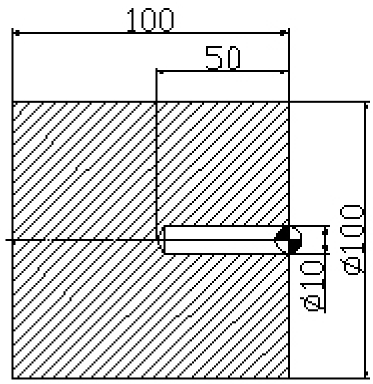


Fig 9.2-8

N10 T1D1 ;Define tool offset

N20 G0 X120 Z50

N30 M3 S500

N40 M8

N50 X0 Z50

N60 R101=50.000 R102=2.000 ; Define values

N70 R103=0.000 R104=-50.000

N80 R105=0.000 R107=200.000

N90 R108=100.000 R109=0.000

N100 R110=-5.000 R111=2.000

N110 R127=1.000

N120 LCYC83 ; call of cycle

N130 G0 X200 Z200

N140 M5 M9

N150 M2

LCYC840 Tapping with compensating chuck

1. Function

The tool drills with the programmed spindle speed and direction of rotation down to the entered thread depth. The feed of the drilling axis results from the spindle speed. This cycle can be used for tapping with compensating chuck and spindle actual-value encoder. The direction of rotation is automatically reversed in the cycle. The retract can be carried out at a separate speed.

2. Call LCYC84

3. Precondition

This cycle can only be used with a speed-controlled spindle with position encoder. The cycle does not check whether the actual-value encoder for the spindle really exists.

The spindle speed and the direction of rotation must be defined in the higherlevel program. The drilling position must be approached before calling the cycle in the higherlevel program.

The required tool with tool offset must be selected before calling the cycle.

4. Parameters declare

Parameter	**Meaning, Value Range**
R101	Retract plane (absolute)
R102	Safety clearance
R103	Reference plane (absolute)
R104	Final drilling depth (absolute)
R106	Thread lead as value.value range: 0.001 .... 2000.000 mm
R126	Direction of rotation of spindle for tapping.Value range: 3 (for M3), 4 (for M4)

Information:

R101-R104 See LCYC84

R106 Thread lead as value

R126 The tapping block is executed with the direction of rotation of spindle programmed under R126. The direction of rotation is automatically reversed in the cycle.

5. **Motional sequence**

Position reached prior to beginning of cycle:

last position in the higher-level program (drilling position)

The cycle produces the following motional sequence:

1. Approach reference plane shifted forward by an amount corresponding to the safety clearance using G0
2. Tapping down to final drilling depth with G33
3. Retract to reference plane shifted forward by an amount corresponding to the safety clearance with G33
4. Retract to retract plane with G0

5. Example

This program is used for tapping on the position X0; the Z axis is the drilling axis. The parameter for the direction of rotation R126 must be parameterized. A compensating chuck must be used for machining. The spindle speed is defined in the higher-level program.

```
N10 G0 G17 G90 S300 M3 D1 T1    ; Define technology values
N20 X35 Y35 Z60                 ; Approach drilling position
G17
N30 R101=60 R102=2 R103=56 R104=15 ; Parameter assignment
N40 R106=0.5 R126=3             ; Parameter assignment
N40 LCYC840                     ; Cycle call
N50 M2                          ; End of program
```

LCYC85 Boring

1. **Function**

The tool drills with the spindle speed and feedrate programmed down to the entered final drilling depth. When the final drilling depth is reached, a dwell time can be programmed. The approach and retract movements are carried out with the feedrates programmed under the respective parameters.

2. **Call**

LCYC85

3. **Precondition**

The spindle speed and the direction of rotation must be defined in the higherlevel program.

The drilling position must be approached before calling the cycle in the higherlevel program.

Before calling the cycle, the respective tool with tool offset must be selected.

4. **Parameters**

Parameter	Meaning, Value Range
R101	Retract plane (absolute)
R102	Safety clearance
R103	Reference plane (absolute)
R104	Final drilling depth (absolute)
R105	Dwell time at drilling depth in seconds
R107	Feed for drilling
R108	Feed when retracting from drill hole

Information：

Parameters R101 - R105 see LCYC82

R107 The feed value defined here acts for drilling.

R108 The feed value entered under R108 acts for retracting from the drill hole.

5. **Motional sequence**

Position reached prior to beginning of cycle: last position in the higher-level program (drilling position)

The cycle produces the following motional sequence:

1) Approach reference plane shifted forward by an amount corresponding to the safety clearance using G0

2)Traverse to final drilling depth with G1 and the feed programmed under parameter R106.

3)Execute dwell time at final drilling depth.

4) Retract to reference plane shifted forward by an amount corresponding to the safety clearance with G1 and the retract feed programmed under R108.

6. **Example**

The cycle LCYC85 is called in Z70 and X50 in the ZX plane. The Y axis is the drilling axis. No dwell time is programmed. The workpiece upper edge is at Y=102.

```
N10 G0 G90 G18 F1000 S500 M3 T1 D1      ; Define technology values
N20 Z70 X50 Y105                         ; Approach drilling position
N30 R101=105 R102=2 R103=102 R104=77 ; Define parameters
N35 R105=0 R107=200 R108=400             ; Define parameters
```

N40 LCYC85 ; Call drilling cycle

N50 M2 ; End of program

LCYC93 Recess cycle

1. Function

The recess cycle is designed to produce symmetrical recesses for longitudinal and face machining on cylindrical contour elements. The cycle is suitable for machining internal and external recesses.

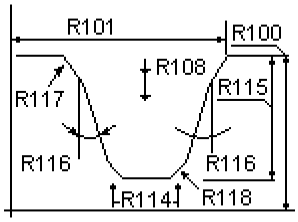
2. Call

LCYC93

3. **Precondition**

The recess cycle can only be called if G23 (diameter programming) is active. The tool offset of the tool whose tool nose width has been programmed with R107 must be activated before the recess cycle is called. The zero position of the tool nose faces machine zero.

4. **Parameters**



Parameter	Meaning, Value Range
R100	Starting point in facing axis
R101	Starting point in longitudinal axis
R105	Machining method, Value range 1 ... 8
R106	Finishing allowance, without sign
R107	Tool nose width, without sign
R108	Infeed depth , without sign
R114	Recess width, without sign
R115	Recess width, without sign
R116	Flank angle, without sign,between 0 <= R116 < = 89.999 degrees
R117	Chamfer on rim of recess
R118	Chamfer on recess base
R119	Dwell time on recess base

Information

R100 The recess diameter in X is specified in parameter R100

R101 R101 determines the point at which the recess starts in the Z axis.

R105 R105 defines the recess variant:

Value	Longitudinal/Facing	External/Internal	Starting Point Position
1	L	A	Left
2	P	A	Left

3	L	I	Left
4	P	I	Left
5	L	A	Right
6	P	A	Right
7	L	I	Right
8	P	I	Right

If the parameter is set to any other value, the cycle is aborted with the alarm 61002 “Machining type incorrectly programmed”.

R106 Parameter R106 determines the finishing allowance for roughing of the recess.

R107 Parameter R107 determines the tool nose width of the recessing tool. This value must correspond to the width of the tool actually used. If the tool nose of the active tool is wider, the contour of the programmed recess will be violated. Such violations are not monitored by the cycle. If the programmed tool nose width is wider than the recess width at the base, the cycle is aborted with the alarm G1602 “Tool width incorrectly defined”.

R108 By programming an infeed depth in R108, it is possible to divide the axisparallel recessing process into several infeed depths. After each infeed, the tool is retracted by 1 mm for chip breakage.

R114 The recess width programmed in parameter R114 is measured on the base.The chamfers are not included in the measurement.

R115 Parameter R115 determines the depth of the recess.

R116 The value of parameter R116 determines the angle of the flanks of the recess. When it is set to “0”, a recess with axis-parallel flanks (i.e. rectangular form) is machined.

R117 R117 defines the chamfers on the recess rim.

R118 R118 defines the chamfers on the recess base. If the values programmed for chamfers do not produce a meaningful recess contour, then the cycle is aborted with the alarm 61603 “Recess form incorrectly defined”.

R119 The dwell time on the recess base to be entered in R119 must be selected such that at least one spindle revolution can take place during the dwell period. It is programmed to comply with an F word (in seconds).

5. **Motional Sequence**

Position reached prior to beginning of the cycle:

- Any position from which each recess can be approached without risk of collision.

The cycle produces the following motional sequence:

- Approach with G0 starting point cacluated internally in the cycle.
- Execute depth infeeds:

  Roughing in parallel axes down to base, taking finishing allowance into account. Tool travels clear for chip breakage after each infeed.
- Execute width infeeds:

Width infeeds are executed perpendicular to the depth infeed with G0, the roughing process for machining the depth is repeated.

The infeeds both for depth and width are distributed evenly with the highest possible value.

- z Rough the flanks. Infeed along the recess width is executed in several steps if necessary.
- z Finish-machine the whole contour, starting at both rims and working towards center of recess base, at the feedrate programmed before the cycle call.

6. Example

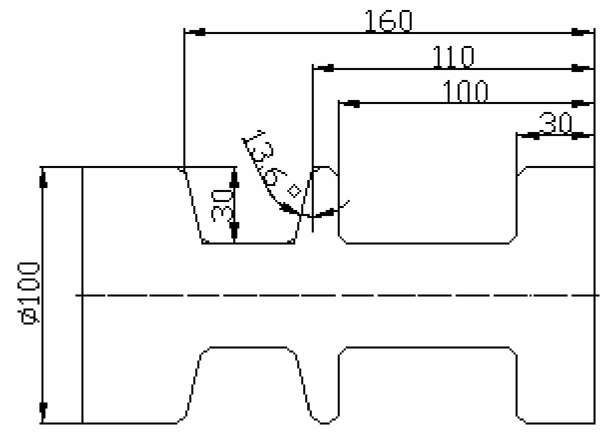


Fig 9.3-1

G55 G0 X0 Z0 M3 S1000 T01 D01

G0 X100

Z-50

R100=100 R101=-100 R105=1

R106=0 R107=3 R108=5

R114=70 R115=30 R116=0

R117=5 R118=5 R119=1

LCYC93

G0 X120

Z-50

R100=100 R101=-110 R105=5

R106=0 R107=3 R108=5

R114=50 R115=30 R116=13.6

R117=5 R118=5 R119=0.5

LCYC93

T01D00

M05

M2

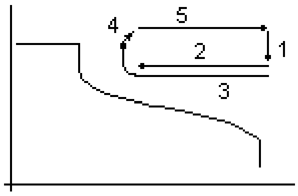
LCYC95 Stock removal cycle

1. Function

This cycle can machine a contour, which is programmed in a subroutine, in a longitudinal or face machining process, externally or internally, through axisparallel stock removal.
The technology (roughing/finishing/complete machining) can be selected. The cycle can be called from any chosen collision-free position. A tool offset must have been activated in the program with the cycle call.

2. Call

LCYC95



3. **Precondition**

- The cycle requires an active G23 (diameter programming).
- The file SGUD.DEF, which is supplied on the cycles diskette, must be available in the control system.
- The stock removal cycle can be called to the 3rd program level.

4. **Parameters**

Parameter	Meaning, Value Range
R105	Machining type：value range 1 ... 12
R106	Finishing allowance, without sign
R108	Infeed depth, without sign
R109	Infeed angle for roughing, it should be zero at face machining
R110	Contour clearance distance for roughing
R111	Feedrate for roughing
R112	Feedrate for finishing

Information

R105 The machining types:

- longitudinal/facing
- internal/external
- roughing/finishing/complete machining

are defined by the parameter determining the type of machining.When longitudinal machining is selected, the infeed always takes place in the facing axis, and vice versa.

Value	Longitudinal/Facing(P)	External/Internal(A/I)	Roughing/Finishing/Complete Machining
1	L	A	Roughing
2	P	A	Roughing
3	L	I	Roughing

4	P	I	Roughing
5	L	A	Finishing
6	P	A	Finishing
7	L	I	Finishing
8	P	I	Finishing
9	L	A	Complete
10	P	A	Complete
11	L	I	Complete
12	P	I	Complete

If any other value is programmed for the parameter, the cycle is aborted and the following alarm output 61002 “Machining type incorrectly programmed”.

R106 A finishing allowance can be programmed in parameter R106.

The workpiece is always rough-machined down to this finishing allowance. In this case, the residual corner produced in the course of each axis-parallel roughing process is immediately cut away in parallel with the contour at the same time. If no finishing allowance is programmed, the workpiece is roughmachined right down to the final contour.

R108 The maximum possible infeed depth for the roughing process is entered under parameter R108. However, the cycle itself calculates the current infeed depth that is applied in rough-machining operations.

R109 The infeed motion for roughing can be executed at an angle which can be programmed in parameter R109. In the face machining process a slanting immerse is not possible, R109 must be programmed to ZERO.

R110 Parameter R110 specifies the distance by which the tool is lifted from the contour in both axes after each roughing operation so that it can be retracted by G0.

R111 The feedrate programmed under R111 applies to all paths on which stock is removed during roughing operations. If finishing is the only machining type selected, then this parameter has no meaning at all.

R112 The feedrate programmed under R112 is applied for finishing operations. If roughing is the only machining type selected, then this parameter has no meaning at all.

Contour definition

The contour to be machined by stock removal is programmed in a subroutine. The name of the subroutine is transferred to the cycle via the _CNAME variable. The contour may consist of straight lines and circle segments; radii and chamfers can be inserted. The programmed circle sections can be quarter circles as a maximum.

Undercuts may not be contained in the contour. If an undercut element is detected, the cycle is aborted, and the alarm 61605 “Contour incorrectly defined” is output.

The contour must always be programmed in the direction that is traversed when finishing according to the selected machining direction.

Roughing

- Approach cycle starting point (calculated internally) with G0 in both axes simultaneously.
- Perform depth infeed with the angle programmed under R109 to the next roughing depth.
- Approach roughing cut point in parallel axes with G1 and at a federate programmed in R111.
- Travel in parallel with contour along contour + finishing allowance up to the last roughing cut point with G1/G2/G3 and at feedrate R111.
- Lift in each axis by the clearance (in mm) programmed in R110 and retract with G0.
- Repeat this sequence until the final roughing depth is reached.

Finishing

- Approach the cycle starting point in individual axes with G0
- Approach the contour starting point in both axes simultaneously with G0.
- Finish-machine along the contour with G1/G2/G3 and at the federate programmed in R112.
- Retract to cycle starting point in both axes with G0.

When finishing is selected, the tool radius compensation is automatically activated internally in the cycle.

Starting point

The cycle automatically calculates the point at which machining must start. The starting point is always approached in both axes simultaneously for roughing and in individual axes for finishing. In this case, the infeed axis approaches the starting point first.

When complete machining is selected, the tool does not return to the internally calculated starting point after the last roughing cut

5. Example

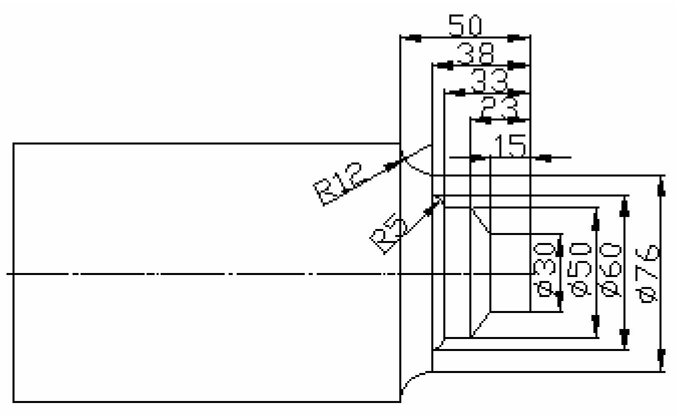


Fig 9.3-2

Main：LC95.MPF

G500 S500 M3 F0.4 T01 D01 ; setting workpiece

Z2 X142 M8

_CNAME="L01"

R105=1 R106=1.2 R108=5 R109=7

R110=1.5 R111=0.4 R112=0.25

LCYC95 ; call lcyc95

T02D01

R105=5 R106=0

LCYC95

G0 G90 X120

Z120 M9

M2

Subroutine：L01.SPF:

G0 X30 Z2

G01 Z-15 F0.3

X50 Z-23

Z-33

G03 X60 Z-38 CR=5

G01 X76

G02 X88 Z-50 CR=12

M02

LCYC97 Thread cutting

1. Function

The thread cutting cycle is suitable for cutting external and internal, single-start or multiple-start threads on cylindrical and tapered bodies in the facing or longitudinal axis. Depth infeed is an automatic function.

Whether a right-hand or left-hand thread is produced is determined by the direction of rotation of the spindle, which must be programmed before calling the cycle. Feed and spindle override are not effective in the traversing blocks containing thread cutting operations.

2. Call

LCYC97

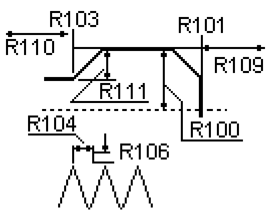


Fig 9.3-3

3. **Parameters**

Parameter	Meaning, Value Range
R100	Diameter of thread at starting point
R101	Thread starting point in longitudinal axis

R102	Diameter at end point
R103	Thread end point in longitudinal axis
R104	Thread lead as value, without sign
R105	Definition of thread cutting method:Value range: 1, 2
R106	Finishing allowance, without sign
R109	Approach path, without sign
R110	Run-out path, without sign
R111	Thread depth, without sign
R112	Starting point offset, without sign
R113	Number of rough cuts, without sign
R114	Number of threads, without sign

Information

R100, R101 These parameters define the thread starting point in X and Z.

R102, R103 The thread end point is programmed under R102 and R103. In the case of cylindrical threads, one of these parameters has the same value as R100 or R101.

R104 The thread lead is an axis-parallel value and is specified without sign.

R105 Parameter R105 defines whether the thread is machined internally or externally.

R105 = 1: External thread

R105 = 2: Internal thread

If the parameter is set to any other value, the cycle is aborted with the alarm 61002 “Machining type incorrectly programmed”.

R106 The programmed finishing allowance is subtracted from the specified thread depth. The remainder is divided into rough cuts.

The finishing allowance is removed in one cut after roughing.

R109, R110 Parameters R109 and R110 specifiy the internally calculated thread approach and run-out paths. The cycle shifts the programmed starting point forward by the approach distance. The run-out path extends the length of the thread beyond the programmed end point.

R111 Parameter R111 defines the total depth of the thread.

R112 An angle value can be programmed in this parameter. This value defines the point at which the first thread cut starts on the circumference of the turned part, i.e. it is a starting point offset.

Possible values for this parameter are between 0.0001 ... + 359.9999 degrees.

If no starting point offset is specified, the first thread automatically starts at the zero-degree marking.

R113 Parameter R113 determines the number of roughing cuts for thread cutting operations. The cycle independently calculates the individual, current infeed depths as a function of the settings in R105 and R111.

R114 This parameter specifies the number of threads. These are arranged symmetrically around the circumference of the turned part.

4 **Motional sequence**

Position reached prior to beginning of cycle:

- Any position from which the programmed thread starting point + approach path can be approached without risk of collision.

The cycle produces the following motional sequence:

- Approach starting point at the beginning of the approach path (calculated internally in the cycle) to cut first thread with G0.
- Infeed for rough cutting according to the infeed method defined under R105.
- Repeat thread cuts according to the programmed number of rough cuts.
- Remove the finishing allowance with G33.
- Repeat the whole sequence for every further thread.

5.Example

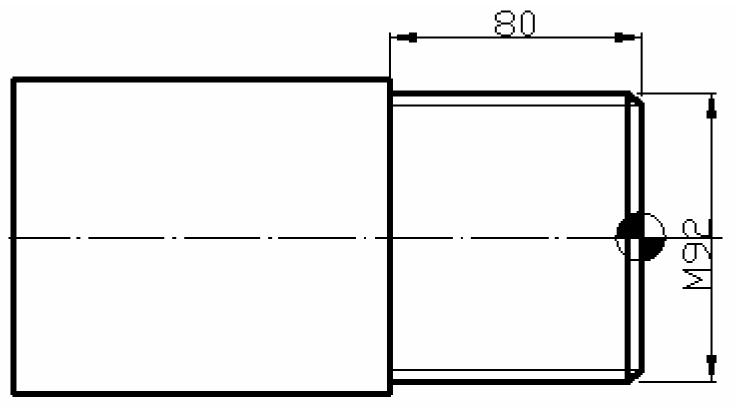


Fig 9.3-4

G55 G00 X0 Z0 M03 S1000 ;setting workpiece

T01 D01

G00 X100

Z50

R100=96 R101=0 R102=100 R103=-100

R104=2 R105=1 R106=0.5

R109=15 R110=35 R111=15

R112=0 R113=7 R114=1

LCYC97 ; call cycle

M05

M2

9.4 Arithmetic parameters R

1. Functionality

If you want an NC program in which you can vary the values to be processed, or if you simply needed to compute arithmetic values, then you can use R (arithmetic) parameters. The control system will calculate or set the values you need when the program is executed. An alternative method is to input the arithmetic parameter values directly. If the R parameters already have value settings, then they can be assigned in the program to other NC addresses that have variable values.

2.Programming

R0=...

to

R249=...

(to R299=..., if there are no machining cycles)

3. Explanation

250 arithmetic parameters with the following classification are available:

R0 ... R99 - for free assignment

R100 ... R249 - transfer parameters for machining cycles.

R250 ... R299 - internal arithmetic parameters for machining cycles.

If you do not intend to use machining cycles (see Section NO TAG “Machining Cycles”), then this range of arithmetic parameters is also available for your use.

4. Value assignment

Example：

R0=3.5678 R1=-37.3 R2=2 R3=-7 R4=-45678.1234

You can assign an extended numerical range using exponential notation:（10^{-300}...10^{+300}）.

The value of the exponent is typed after the characters EX. Maximum number of characters: 10 (including sign and decimal point).

Value range of EX: -300 to +300.

Example：

R0=-0.1EX-5 ;Meaning: R0 = -0,000 001

R1=1.874EX8 ;Meaning: R1 = 187 400 000

Note: Several assignments (including arithmetic expressions) can be programmed in one block.

5. Assignment to other addresses

You can obtain a flexible NC program by assigning arithmetic parameters or arithmetic expressions with R parameters to other NC addresses. Values, arithmetic expressions or R parameters can be assigned to any NC address with the exception of addresses N, G and L.

When making assignments of this kind, type the character “=” after the address character.

Assignments with a negative sign are also permitted.

If you wish to make assignments to axis addresses (traversal instructions), then you must do so in a separate program block.

Example：

N10 G0 X=R2 ;Assignment to X axis

6. Arithmetic operations functions

Operators/arithmetic functions must be programmed using the normal mathe / matical notation.

Processing priorities are set by means of round brackets. Otherwise the “multiplication/division before addition/subtraction” rule applies. Degrees are specified for trigonometric functions.

9.5 Program jumps

9.5.1 label --- Jump destination for program jumps

1. Functionality

1) A **label** or a **block number** serve to mark blocks as jump destinations for program jumps. Program jumps can be used to branch to the program sequence.

2) Labels can be freely selected, but must contain a minimum of 2 and a maximum of 8 letters or numbers, and the **first two characters** must be **letters** or underscores.

3) Labels that are in the block that serves as the jump destination are **ended by a colon**. They are always at the start of a block. If a block number is also present, the label is located **after the block number**.

4) Labels must be unique within a program.

2. Programming example

N10 LABEL1: G1 X20 ; LABEL1 is the label, jump destination

...

TR789: G0 X10 Z20 ; TR789 is the label, jump destination

– No block number existing

N100 .. . ; A block number can be a jump destination.

9.5.2 Unconditional program jumps

1. Functionality

NC programs process their blocks in the sequence in which they were arranged when they were written.

The processing sequence can be changed by introducing program jumps.

The jump destination can be a block with a **label** or with a **block number**. This block must be located within the program.

The unconditional jump instruction requires a separate block

2. Programming

GOTOF Lable ; GoTo operation

GOTOB Lable ;GoBack operation

AWL	Note
GOTOF	; GoTo operation (in the direction of the last block of the program)
GOTOB	;GoBack operation (in the direction of the first block of the program)
Lable	; Selected string for the label (jump label) or for the block number

9.5.3Conditional program jumps

1. Functionality

Jump conditions are formulated after the **IF instruction**. If the jump condition (**value not zero**) is satisfied, the jump takes place.

The jump destination can be a block with a **label** or with a **block number**. This block must be located within the program.

Conditional jump instructions require a separate block. Several conditional jump instructions can be located in the same block.

By using conditional program jumps, you can also considerably shorten the program, if necessary.

2.Programming

IF *condition* GOTOF *label* ; GoTo operation (forward jum)

IF *condition* GOTOB *label* ; GoBack operation (reverse jump)

AWL	**Meaning**
GOTOF	Jump direction forward (in the direction of the last block of the program)
GOTOB	Jump direction reverse (in the direction of the first block of the program)
Lable	Selected string for the label (jump label) or for the block number
IF	Introduction of the jump condition
Condition	R parameter, arithmetic expression for formulating the condition

3. Comparison operations

Operators	**Meaning**
==	Equal to
<>	Not equal to
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to

The comparison operations support formulating of a jump condition. Arithmetic expressions can also be compared.

The result of comparison operations is ”satisfied” or ”not satisfied.” ”Not satisfied” sets the value to zero.

4. Programming example for comparison operators

R1>1 ;R1 greater than 1

1 < R1 ;1 less than R1

R1<R2+R3 ;R1 less than R2 plus R3

R6>=SIN(R7*R7) ;R6 greater than or equal to SIN(R7)^2

9.5.4 Programming example

Task

Approaching points on a circle segment:

Given: Starting angle: 30 . in R1

Circle radius: 32 mm in R2

Spacing between the positions: 10 . in R3

Number of points: 11 in R4

Position of the circle center in Z: 50 mm in R5

Position of the circle center in X: 20 mm in R6

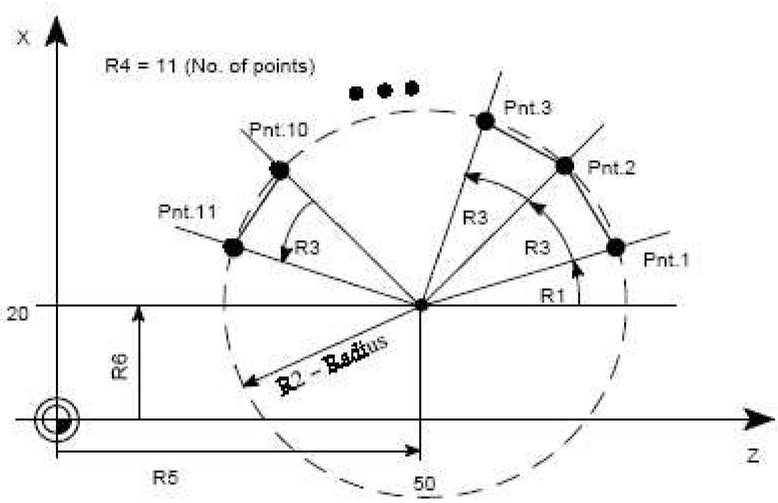


Fig 9.3－5

Programming example

N10 R1=30 R2=32 R3=10 R4=11 R5=50 R6=20 ; Assignment of the starting values

N10 MA1: G0 Z=R2 *COS (R1)+R5 X=R2*SIN(R1)+R6

; Calculation and assignment to axis addresses

N30 R1=R1+R3 R4= R4–1

N40 IF R4 > 0 GOTOB MA1

N50 M2

Explanation

In block N10, the starting conditions are assigned to the corresponding arithmetic parameters. The calculation of the coordinates in X and Z and the processing takes place in N20.

In block N30, R1 is incremented by the clearance angle R3, and R4 is decremented by 1. If R4 > 0, N20 is executed again; otherwise, N50 with end of program.

9.6 Subroutine

1. Application

Basically, there is no difference between a main program and a subroutine.
Frequently recurring machining sequences are stored in subroutines, e.g certain contour shapes. These subroutines are called at the appropriate locations in the main program and then executed.

One form of subroutine is the **machining cycle**. Machining cycles contain universally valid machining scenarios (e.g.: drilling, tapping, groove milling, etc.). By assigning values via included transfer parameters, you can adapt the subroutine to your specific application.

2. Structure

The structure of a subroutine is identical to that of a main program .Like main programs, subroutines contain **M2 – end of program** in the last block of the program sequence. This means a return to the program level where the subroutine was called from.

3. End of program

The end instruction **RET** can also be used instead of the M2 program end in the subroutine.

RET requires a separate block.

The RET instruction is used when G64 continuous-path mode is not to be interrupted by a return. With M2, G64 is interrupted and exact stop is initiated.

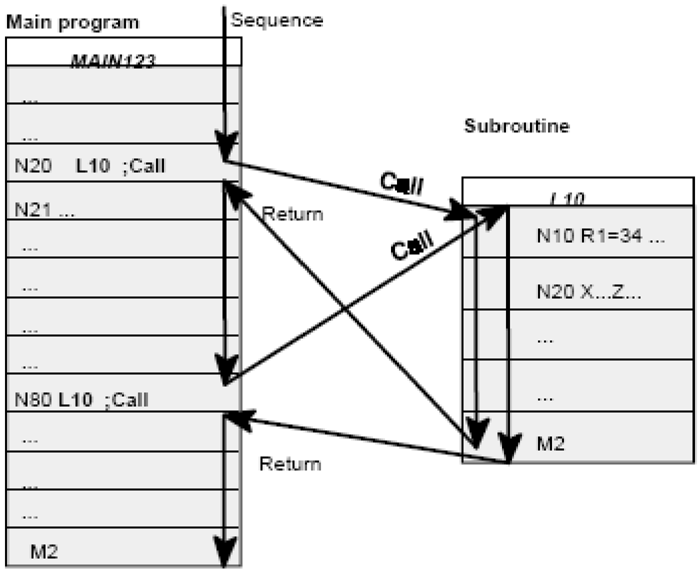


Fig 9.3－6

4. Subroutine name

The subprogram is given a unique name allowing it to be selected from several subroutines. When you create the program, the program name may be freely selected provided the following conventions are observed:

_ The first two characters must be letters

_ The others may be letters, digits or underscore

_ Maximum of 8 characters in total

_ No dashes (see Section “Character set”)

The same rules apply as for main program names.

5. Subroutine call

Subroutines are called in a program (main or subprogram) with their names. To do this, a separate block is required.

Example

N10 L785 ;Call of subroutine L785

N20 WELLE7 ;Call of subroutine WELLE7

6. Program repetition P...

If a subroutine is to be executed several times in succession, write the number of times it is to be executed in the block of the call after the subroutine name under the **address P**. A maximum of **9,999 cycles** are possible (P1 ... P9999).

Example

N10 L785 P3 ; Call of subroutine L785 , 3 passes

7. Nesting depth

It is not only possible to call subroutines in main programs, but also in other subroutines. There is a total of 4 program levels (including the main program level) available for programming this type of nested call.

Note: If you are working with machining cycles, please remember that these also need one of the four program levels.

8. Information

Modal G functions can be changed in the subroutine, e.g. G90 –> G91. When returning to the calling program, ensure that all modal functions are set the way you need them to be.

Please make sure that the values of your arithmetic parameters used in upper program levels are not inadvertently changed in lower program levels.

When working with SIEMENS cycles, up to 4 program levels are needed.

CHAPTER 10 SINUMERIK 810/840 programme

10.1 Position

Plane selection: G17 to G19

Functionality

To assign, for example, **tool radius and tool length compensations**, a plane with two axes is selected from the three axes X, Y and Z. In this plane, you can activate a tool radius compensation.

For drill and cutter, the length compensation (length 1) is assigned to the axis standing vertically on the selected plane (see Section 8.6 ”Tool and tool offsets”). It is also possible to use a 3-dimensional length compensation for special cases.

Another influence of plane selection is described with the appropriate functions (e.g. Section 8.5 ”Rounding, chamfer”).

The individual planes are also used to define the **direction of rotation of the circle for the circular interpolation** CW or CCW. In the plane in which the circle is traversed, the abscissa and the ordinate are designed and thus also the direction of rotation of the circle.

Circles can also be traversed in a plane other than that of the currently active G17 to G19 plane (see Chapter 8.3 ”Axis Movements”).

The following plane and axis assignments are possible:

Table 8-2 Plane and axis assignments

G function	Plane (abscissa/ordinate)	vertical axis on plane (length compensation axis when drilling/milling)
G17	X / Y	Z
G18	Z / X	Y
G19	Y / Z	X

Fig 10.1－1

Absolute / incremental dimensioning: G90, G91, AC, IC

Functionality

With the instructions G90/G91, the written positional data X, Y, Z, ... are evaluated as a coordinate point (G90) or as an axis position to traverse to (G91). G90/G91 applies to all axes.

Irrespective of G90/G91, certain positional data can be specified for certain blocks in absolute/ incremental dimensions using AC/IC.

These instructions do **not determine the path** by which the end points are reached; this is provided by a G group (G0, G1, G2 and G3... see Chapter 8.3 ”Axis Movements”).

Programming

G90 ; Absolute dimensioning

G91 ; Incremental dimensioning

X=AC(...) ; Absolute dimensioning for a certain axis (here: X axis), non-modal

X=IC(...) ; Absolute dimensioning for a certain axis (here: X axis), non-modal

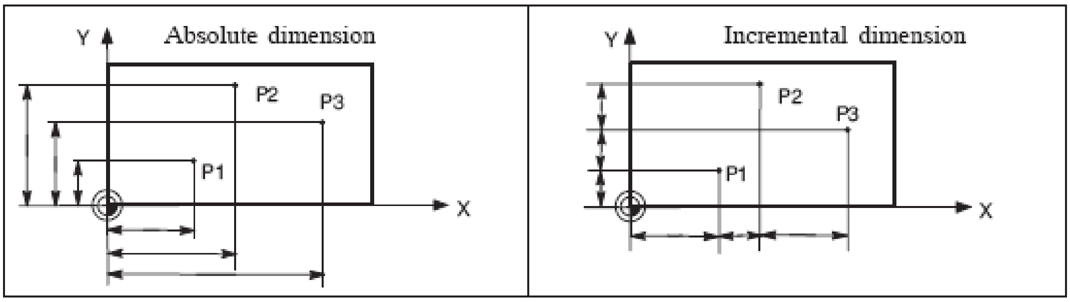


Fig 10.2－2

Absolute dimensioning G90

With absolute dimensioning, the dimensioning data refers to the **zero of the coordinate system currently active** (workpiece or current workpiece coordinate system or machine coordinate system). This is dependent on which offsets are currently active: programmable, settable, or no offsets.

Upon program start, G90 is active for **all axes** and remains active until it is deselected in a subsequent block by G91 (incremental dimensioning data) (modally active).

Incremental dimensioning G91

With incremental dimensioning, the numerical value of the path information corresponds to the **axis path to be traversed**. The leading sign indicates the **traversing direction**.

G91 applies to all axes and can be deselected in a subsequent block by G90 (absolute dimensioning).

Specification with =AC(...), =IC(...)

After the end point coordinate, write an equality sign. The value must be specified in round brackets.

Absolute dimensions are also possible for circle center points using =AC(...). Otherwise, the reference point for the circle center is the circle starting point.

Programming example

N10 G90 X20 Z90 ; Absolute dimensioning

N20 X75 Z=IC(–32) ; X dimensioning continues to be absolute, Z incremental dimension

...

N180 G91 X40 Z20 ; Switching to incremental dimensioning

N190 X–12 Z=AC(17) ; X – continues to be incremental dimensioning, Z – absolute

Dimensions in metric units and inches: G71, G70, G710, G700

Functionality

If workpiece dimensions that deviate from the base system settings of the control are present (inch or mm), the dimensions can be entered directly in the program. The required conversion into the base system is performed by the control system.

Programming

G70 ; Inch dimension input

G71 ; Metric dimension data input

G700 ; Inch dimension data input; also for feedrate F

G710 ; Metric dimension data input; also for feedrate F

Programming example

N10 G70 X10 Z30 ; Inch dimension input

N20 X40 Z50 ; G70 continues to be active

...

N80 G71 X19 Z17.3 ; Metric dimesioning from here

Information

Depending on the **default setting** you have chosen, the control system interprets all geometric values as either metric **or** inch dimensions. Tool offsets and settable work offsets including their display are also to be understood as geometrical values; this also applies to the feedrate F in mm/min or inch/min. The default setting can be set via machine data.

All examples listed in this Manual are based on a **metric default setting**.

G70 or G71 evaluates all geometric parameters that directly refer to the **workpiece**, either as inches or metric units, for example:

_ Positional data X, Y, Z, ... for G0,G1,G2,G3,G33, CIP, CT

_ Interpolation parameters I, J, K (also thread pitch)

_ Circle radius CR

_ **Programmable** work offset (TRANS, ATRANS)

_ Polar radius RP

All remaining geometric parameters that are not direct workpiece parameters, such as feedrates, tool offsets, and **settable** work offsets, are not affected by **G70/G71**.

G700/G710 however, also affects the feedrate F (inch/min, inch/rev. or mm/min, mm/rev.).

Polar coordinates, pole definition: G110, G111, G112

Functionality

In addition to the common specification in Cartesian coordinates (X, Y, Z), the points of a workpiece can also be specified using polar coordinates.

Polar coordinates are also helpful if a workpiece or a part of it is dimensioned from a central point (pole) with specification of the radius and the angle.

Plane

The polar coordinates refer to the plane activated with G17 to G19.

In addition, the 3rd axis standing vertically on this plane can be specified. When doing so, spatial specifications can be programmed as cylinder coordinates.

Polar radius RP=...

The polar radius specifies the distance of the point to the pole. It is stored and must only be written in blocks in which it changes, after changing the pole or when switching the plane.

Polar angle AP=...

The angle is always referred to the horizontal axis (abscissa) of the plane (for example, with G17: X axis). Positive or negative angle specifications are possible.

The polar angle remains stored and must only be written in blocks in which it changes, after changing the pole or when switching the plane.

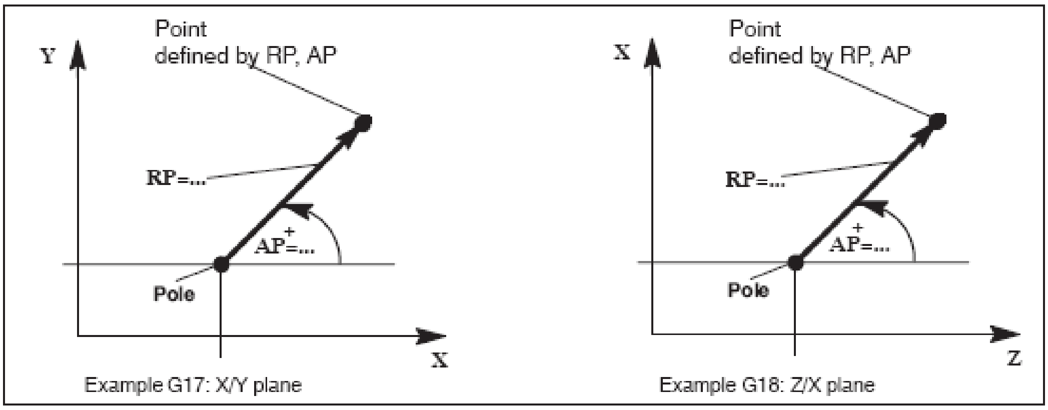


Fig 10.2—3

Pole definition, programming

G110 ; Pole specification, relative to the last programmed set position

(in the plane, e.g. G17: X/Y)

G111 ; Pole specification, relative to the origin of the current

workpiece coordinate system (in the plane, e.g. G17: X/Y)

G112 ; Pole specification, relative to the last valid pole; preserve plane

Notes

_ Pole definitions can also be performed using polar coordinates. This makes sense if a pole already exists.

_ If no pole is defined, the origin of the current workpiece coordinate system will act as the pole.

Programming example

N10 G17 ; X/Y plane

N20 G111 X17 Y36 ; Pole coordinates in current workpiece coordinate system

...

N80 G112 AP=45 RP=27.8 ; New pole, relative to the last pole as a polar coordinate

N90 ... AP=12.5 RP=47.679 ; Polar coordinate

N100 ... AP=26.3 RP=7.344 Z4 ; Polar coordinate and Z axis (= cylinder coordinate)

10.2 G Commands

10.2.1 Fundamental Principles of NC Programming

Program names

Each program has its own program name. When creating a program, the program name can be freely selected, observing the following rules:

_ The first two characters must be letters;

_ Use only letters, digits or underscore.

_ Do not use delimiters (see Section ”Character set”).

_ The decimal point must only be used for separation of the file extension.

_ Do not use more than 30 characters.

Example: **FRAME52**

Program structure

Structure and contents

The NC program consists of a sequence of **blocks** (see Table 8-1).

Each block represents a machining step.

Instructions are written in the blocks in the form of **words**.

The last block in the execution sequence contains a special word for the **end of program: M2**.

Table 8-1 NC program structure

Block Word Word Word ... ; Comment

Block N10 G0 X20 ... ; 1. Block

Block N20 G2 Z37 ... ; 2. Block

Block N30 G91 ... ... ; ...

Block N40 ... ...

Block N50 M2 ; End of program

Word structure and address

Functionality/structure

A word is a block element and mainly constitutes a control command. The word consists of

_ **address character**: generally a letter

_ and a **numerical value**: a sequence of digits which with certain addresses can be added by a sign put in front of the address, and a decimal point.

A positive sign (+) can be omitted.

Word

Address Value

Example: G1

Word

Address Value

X –20.1

Word

Address Value

F300

Explanation: Traverse with

Linear

interpolation

Path or lilmit

position for the

X axis: –20.1 mm

Feedrate:

300 mm/min

Figure 8-1 Word structure (example)

Several address characters

A word can also contain several address letters. In this case, however, the numerical value must be assigned via the intermediate character **”=”**.

Example: **CR=5.23**

Additionally, it is also possible to call G functions using a symbolic name (see also Section ”List of instructions”).

Example: **SCALE** ; Enable scaling factor

Extended address

With the addresses

R Arithmetic parameters

H H function

I, J, K Interpolation parameters/intermediate point

the address is extended by 1 to 4 digits to obtain a higher number of addresses. In this case, the value must be assigned using an equality sign ”=” (see also Section ”List of instructions”).

Example: **R10=6.234 H5=12.1 I1=32.67**

Block structure

Functionality

A block should contain all data required to execute a machining step.

Generally, a block consists of several **words** and is always completed with the **end-of-block character** ”**LF**” (Line Feed). This character is automatically generated when pressing the line feed key or the **Input** key.

/N... Word1 Word2 ... Wordn ;Comment LF

End-of-block

character

only if required

is written at the end,

delimited from the
remaining part of the block
by **” ;** ”
Space Space Space Space
Block instructions
Block number – stands in front of instructions;
only if necessary; instead of ”N”, in main blocks,
the following character is used (**” : ”** Colon (:)
Block skip;
only if necessary; stands in the beginning
(BLANK)
Total number of characters in a block: **512** characters
Figure 8-2 Block structure diagram

Word order

If a block contains several instructions, the following order is recommended:

N... G... X... Y... Z... F... S... T... D... M... H...

Note regarding block numbers

First select the block numbers in steps of 5 or 10. Thus, you can later insert

Note regarding block numbers

First select the block numbers in steps of 5 or 10. Thus, you can later insert blocks and nevertheless observe the ascending order of block numbers.

Block skip

Blocks of a program, which are to be executed not with each program run, can be **marked** by a slash **/** in front of the block number. The block skip operation itself is activated either via **operation** (Program control: ”SKP”) or via the PLC (signal). It is also possible to skip a whole program section by skipping several blocks using the ” **/** ”.
If block skip is active during the program execution, all blocks marked with ” **/** ” are skipped. All instructions contained in the blocks concerned will not be considered. The program is continued with the next block without marking.

Comment, remark

The instructions in the blocks of a program can be explained using comments (remarks). A comment is started with the character ” ; ” and ends with the end–of–block chracter. Comments are displayed in the current block display, together with the remaining contents of the block.

Messages

Messages are programmed in a separate block. A message is displayed in a special field and remains active until a block with a new message is executed or until the end of the program is reached. Max. **65** characters of a text message can be displayed.

A message without message text will delete any previous message.

MSG (”THIS IS THE MESSAGE TEXT”)

Programming example

N10 ;G&S company, order no. 12A71

N20 ;Pump part 17, drawing no.: 123 677

N30 ;Program created by H. Adam, Dept. TV 4

N40 MSG(”BLANK ROUGHING”)

:50 G17 G54 G94 F470 S20 D2 M3 ;Main block

N60 G0 G90 X100 Y200

N70 G1 Y185.6

N80 X112

/N90 X118 Y180 ;Block can be skipped

N100 X118 Y120

N110 G0 G90 X200

N120 M2 ;End of program

Character set

The following characters are used for programming; they are interpreted in accordance with the relevant definitions.

Letters, digits

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

0, 1, 2, 3, 4, 5, 6, 7, 8, 9

No distinction is made between upper and lower case letters.

Printable special characters

(Round left bracket ” Inverted commas

) Round right bracket _ Underscore (belonging to letter)

[Square left bracket . Decimal point

] Square right bracket , Comma, delimiter

< Less than ; Start of comment

> Greater than % Reserved; do not use

: Main block, end of label & Reserved; do not use

= Assignment; subset of euqality ’ Reserved; do not use

/ Division; block skip $ System-internal variable identifier

* Multiplication ? Reserved; do not use

+ Addition; plus sign ! Reserved; do not use

– Subtraction; minus sign

Non–printable special characters

LF Line Feed (end-of-block character)

Blank Delimiter between words; blank

Tabulator Reserved; do not use

Overview of the instructions

Address	Meaning	Value assignment	Information	Programming
D	Tool offset number	0 ... 9, only integer, no sign	Contains offset data for a certain tool T... ; D0–> offset values= 0, max. 9 D numbers per tool	D...
F	Feedrate	0.001 ... 99 999.999	Path velocity of a tool/workpiece; unit: mm/min or mm/revolution depending on G94 or G95	F...
F	Dwell time in block with G4	0.001 ... 99 999.999	Dwell time in seconds	G4 F... ;separate block
G	G function (preparatory function)	Only integer, specified values	The G functions are divided into G groups. Only one G function of a group can be programmed in a block. A G function can be either modal (until it is canceled by another function of the same group) or only effective for the block in which it is programmed non-modal. **G group:**	G... or symbolic name, e.g.: CIP
G0	Linear interpolation at rapid traverse rate		1: Motion commands	G0 X... Y... Z... ; Cartesian using polar coordinates: G0 AP=... RP=... or with additional axis: G0 AP=... RP=... Z... ; e.g.: with G17, Z axis
G1 *	Linear interpolation at feedrate		(type of interpolation)	G1 X... Y... Z... F... With polar coordinates: G1 AP=... RP=... F... or with additional axis: G1 AP=... RP=... Z... F... ; e.g.: with G17, Z axis
G2	Circular interpolation CW (in conjunction with a 3rd axis and TURN=... also helix interpolation –> see also TURN)		modally effective	G2 X... Y... I... J... F... ; Center and end points G2 X... Y... CR=... F... ; Radius and end point G2 AR=... I... J... F... ; Aperture angle and center point G2 AR=... X... Y... F... ; Aperture angle and end point in polar coordinates: G2 AP=... RP=... F... or with additional axis: G2 AP=... RP=... Z... F... ; e.g.: with G17, Z axis
G3	Circular interpolation CCW (in conjunction with a 3rd axis and TURN=... also helix interpolation –> see also TURN)			G3 .... ;otherwise, as with G2

CIP	Circular interpolation via intermediate point		CIP X... Y... Z... I1=... J1=... K1=... F...
CT	Circular interpolation; tangential transition		N10 ... N20 CT X... Y... F... ;Circle, tangential transition to the previous path segment
G33	Thread cutting, tapping with constant lead		S... M... ;spindle speed, direction G33 Z... K... ;Tapping **with** compensation chuck, e.g. in the Z axis
G331	Thread interpolation		N10 SPOS=... ;Position-controlled spindle N20 G331 Z... K... S... ;Tapping **without** compensation chuck, e.g. in the Z axis ;RH or LH thread is specified via the arithmetic sign of the lead (e.g. K+): + : as with M3 – : as with M4
G332	Thread interpolation – retraction		G332 Z... K... ;rigid tapping (**without** compensation chuck, e.g. along the Z axis, **retraction motion** ; Sign of the lead as with G331
G4	Dwell time	2: Special motions non-modal	G4 F... ;separate block, F: Time in seconds or G4 S.... ;separate block, S: in spindle revolutions
G63	Tapping with compensation chuck		G63 Z... F... S... M...
G74	Reference point approach		G74 X1=0 Y1=0 Z1=0 ;separate block (machine axis identifier!)
G75	Fixed-point approach		G75 X1=0 Y1=0 Z1=0 ;separate block (machine axis identifier!)
G147	Smooth approach and retraction along a straight line		G147 G41 DISR=... DISCL=... FAD=... F... X... Y... Z...
G148	Smooth approach and retraction along a straight line		G148 G40 DISR=... DISCL=... FAD=... F... X... Y... Z...
G247	Smooth approach and retraction with a quarter		G247 G41 DISR=... DISCL=... FAD=... F... X... Y... Z...
G248	Smooth approach and retraction with a quarter		G248 G40 DISR=... DISCL=... FAD=... F... X... Y... Z...
G347	Smooth approach and retraction with a semicircle		G347 G41 DISR=... DISCL=... FAD=... F... X... Y... Z...
G348	Smooth approach and retraction with a semicircle		G348 G40 DISR=... DISCL=... FAD=... F... X... Y... Z...
TRANS	Programmable offset	3: Write memory	TRANS X... Y... Z... ;separate block
ROT	programmable rotation	non-modal	ROT RPL=... ;rotation in the current plane G17 ... G19, separate block
SCALE	Programmable scaling factor		SCALE X... Y... Z... ;scaling factor in the direction of the specified axis, separate block

MIRROR	Programmable mirroring		MIRROR X0	;Coordinate axis whose direction is changed; separate block
ATRANS	Additive programmable offset		ATRANS X... Y... Z...	;Separate block
AROT	Additive programmable rotation		AROT RPL=...	;Add. rotation in the current plane G17 ... G19, separate block
ASCALE	Additive programmable scaling factor		ASCALE X... Y... Z...	; Scaling factor in the direction of the specified axis, separate block
AMIRROR	additive programmable mirroring		AMIRROR X0	;Coordinate axis whose direction is changed; separate block
G25	Lower spindle speed limitation or lower working area limitation		G25 S... G25 X... Y ... Z...	;Separate block ;Separate block
G26	Upper spindle speed limitation or upper working area limitation		G26 S... G26 X... Y ... Z...	;Separate block ;Separate block
G110	Pole specification, relative to the last programmed set position		G110 X... Y... G110 RP=... AP=...	;Pole specification, Cartesian, e.g.: With G17 ;pole specification, polar separate block
G111	Pole specification, relative to the origin of the current workpiece coordinate system		G111 X... Y... G111 RP=... AP=...	;Pole specification, Cartesian, e.g.: With G17 ;pole specification, polar separate block
G112	Pole specification, relative to the POLElast valid		G112 X... Y... G112 RP=... AP=...	;Pole specification, Cartesian, e.g.: With G17 ;pole specification, polar separate block
G17 *	X/Y plane	6: Plane selection	G17	;Vertical axis on this
G18	Z/X plane	modally effective		plane is tool length
G19	Y/Z plane			offset axis
G40 *	Tool radius compensation OFF	7: Tool radius compensation modally effective		
G41	Tool radius compensation left of the contour			
G42	Tool radius compensation right of the contour			

G500 *	Settable work offset OFF	8: Settable work offset modally effective	
G54	1st settable work offset		
G55	2nd settable work offset		
G56	3rd settable work offset		
G57	4th settable work offset		
G58	5th settable work offset		
G59	6th settable work offset		
G53	Non-modal skipping of the settable work offset	9: Skipping of the settable work offset non-modal	
G153	Non-modal skipping of the settable work offset including base frame		
G60 *	Exact stop	10: Approach behavior modally effective	
G64	Continuous-path control mode		
G9	Non-modal exact stop	11: Non-modal exact stop non-modal	
G601 *	Exact stop window, fine, with G60, G9	12: Exact stop window modally effective	
G602	Exact stop window, coarse, with G60, G9		
G70	Inch dimension input	13: Inch / metr.dimension input modally effective	
G71 *	Metric dimension data input		
G700	Inch dimension data input; also for feedrate F		
G710	Metric dimension data input; also for feedrate F		
G90 *	Absolute dimension data input	14: Absolute / incremental dimension modally effective	
G91	Incremental dimension data input		
G94 *	Feed F in mm/min	15: Feedrate / spindle modally effective	
G95	Feedrate F in mm/spindle revolutions		
CFC *	Feedrate with circle ON	16: Feedrate override modally effective	
CFTCP	Feedrate override OFF		
G450 *	Transition circle	18: Behavior at corners when working with tool radius compensation modally effective	
G451	Point of intersection		
BRISK *	Jerking path acceleration	21: Acceleration profile modally effective	
SOFT	Jerk-limited path acceleration		

Address	Meaning	Value Assignment	Information	Programming
H H0= through H9999=	H function	± 0.0000001 ... 9999 9999 (8 decimals) or with specification of an exponent: ± (10^{-300} ... 10^{+300})	Value transfer to the PLC; meaning defined by the machine manufacturer	H0=... H9999=... e. g.: H7=23.456
I	Interpolation parameters	±0.001 ... 99 999.999 Thread: ±0.001 ... 2000.000	Belongs to the X axis; meaning dependent on G2,G3 –> circle center or G33, G331, G332 –> thread lead	See G2, G3, G33, G331 and G332
J	Interpolation parameters	±0.001 ... 99 999.999 Thread: ±0.001 ... 2000.000	Belongs to the Y axis; otherwise, as with I	See G2, G3, G33, G331 and G332
K	Interpolation parameters	±0.001 ... 99 999.999 Thread: ±0.001 ... 2000.000	Belongs to the Z axis; otherwise, as with I	See G2, G3, G33, G331 and G332
I1=	Intermediate point for circular interpolation	±0.001 ... 99 999.999	Belongs to the X axis; specification for circular interpolation with CIP	See CIP
J1=	Intermediate point for circular interpolation	±0.001 ... 99 999.999	Belongs to the Y axis; specification for circular interpolation with CIP	See CIP
K1=	Intermediate point for circular interpolation	±0.001 ... 99 999.999	Belongs to the Z axis; specification for circular interpolation with CIP	See CIP
L	Subroutine; name and call	7 decimals; integer only, no sign	It is also possible to use L1 ...L9999999, Instead of a free name; thus, the subroutine will be called in a separate block. Please observe: L0001 is not always equal to L1. The name "LL6" is reserved for the tool change subroutine.	L781 ;separate block
M	Miscellaneous function	0 ... 99 integer only, no sign	For example, for initiating switching actions, such as "Coolant ON"; max. 5 M functions per block	M...
M0	Programmed stop		The machining is stopped at the end of a block containing M0; to continue, press NC START.	
M1	Optional stop		As with M0, but the stop is only performed if a special signal (Program control: "M01") is present.	
M2	End of program		Can be found in the last block of the processing sequence	
M30	–		Reserved; do not use	
M17	–		Reserved; do not use	
M3	Spindle CW rotation			
M4	Spindle CCW rotation			

Address	Meaning	Value Assignment	Information	Programming
POT( )	Square			R12=POT(R13)
ABS( )	Amount			R8=ABS(R9)
TRUNC( )	Integer portion			R10=TRUNC(R11)
LN( )	Natural logarithm			R12=LN(R9)
EXP( )	Exponential function			R13=EXP(R1)
RET	End of subroutine		Used instead of M2 – to maintain the continuous-path control mode	RET ;separate block
S	Spindle speed	0.001 ... 99 999.999	Unit of measurement of the spindle r.p.m.	S...
S	Dwell time in block with G4	0.001 ... 99 999.999	Dwell time in spindle revolutions	G4 S... ;separate block
T	Tool number	1 ... 32 000 integer only, no sign	The tool change can be performed either directly using the T command or only with M6. This can be set in the machine data.	T...
X	Axis	±0.001 ... 99 999.999	G command	X...
Y	Axis	±0.001 ... 99 999.999	G command	Y...
Z	Axis	±0.001 ... 99 999.999	G command	Z...
AC	Absolute coordinate	–	The dimension can be specified for the end or center point of a certain axis, irrespective of G91.	N10 G91 X10 Z=AC(20) ;X – incremental dimension, Z – absolute
ACC[*axis*]	Percentage path acceleration override	1 ... 200, integer	Acceleration override for an axis or spindle; specified as a percentage	N10 ACC[X]=80 ;for the X axis: 80% N20 ACC[S]=50 ;for the spindle: 50%
ACP	Absolute coordinate; approach position in the positive direction (for rotary axis, spindle)	–	It is also possible to specify the dimensions for the end point of a rotary axis with ACP(...) irrespective of G90/G91; also applies to spindle positioning	N10 A=ACP(45.3) ;Approach absolute position of the A axis in the positive direction N20 SPOS=ACP(33.1) ;Position spindle
ACN	Absolute coordinate; approach position in the negative direction (for rotary axis, spindle)	–	It is also possible to specify the dimensions for the end point of a rotary axis with ACN(...) irrespective of G90/G91; also applies to spindle positioning	N10 A=ACN(45.3) ;Approach absolute position of the A axis in the negative direction N20 SPOS=ACP(33.1) ;Position spindle
ANG	Angle for the specification of a straight line for the contour definition	±0.00001 ... 359.99999	Specified in degrees; one possibility of specifying a straight line when using G0 or G1 if only one end-point coordinate of the plane is known or if the complete end point is known with contour ranging over several blocks	N10 G1 G17 X... Y.... N11 X... **ANG=...** or contour over several blocks: N10 G1 G17 X... Y.... N11 **ANG=...** N12 X... Y... **ANG=...**
AP	Polar angle	0 ... ±359.99999	Specified in degrees, traversing in polar coordinates, definition of the pole; in addition: RP – polar radius	see G0, G1, G2, G3 G110, G111, G112

Address	Meaning	Value Assignment	Information	Programming
SLOT2	Milling a circumferential slot			N10 SLOT2(...) ;separate block
POCKET3	Square pocket			N10 POCKET3(...) ;separate block
POCKET4	Circular pocket			N10 POCKET4(...) ;separate block
CYCLE71	Face milling			N10 CYCLE71(...) ;separate block
CYCLE72	Contour milling			N10 CYCLE72(...) ;separate block
LONG-HOLE	Long hole			N10 LONGHOLE(...) ;separate block
DC	Absolute coordinate; approach position directly (for rotary axis, spindle)	–	It is also possible to specify the dimensions for the end point of a rotary axis with DC(...) irrespective of G90/G91; also applies to spindle positioning	N10 A=DC(45.3) ;Approach absolute position of the A axis directly N20 SPOS=DC(33.1) ; Position spindle
DEF	Definition instruction		Defining a local user variable of the type BOOL, CHAR, INT, REAL, STRING[n], directly at the beginning of the program	DEF INT VARI1=24, VARI2 ; 2 variables of the type INT ; the name is defined by the user DEF STRING[12] VARS3="HELLO" ;max. 12 characters
DISCL	Approach / retraction distance of the infeed movement to the machining plane (SAR)	–	Safety clearance for switching the speed for the infeed movement; please observe: G340, G341	See with G147, G148 , G247, G248 , G347, G348
DISR	Approach/retraction distance or approach/retraction radius (SAR)	–	G147/G148: Distance of the cutter edge from the starting or end point of the contour G247, G347/G248, G348: Radius of the tool center point path	See with G147, G148 , G247, G248 , G347, G348
FAD	Infeed speed (SAR)	–	The speed acts after reaching the safety clearance during infeed. Please observe: G340, G341	See with G147, G148 , G247, G248 , G347, G348
FRC	Non-modal feedrate for chamfer/rounding	0, >0	In case FRC=0: Feedrate Fwill act	For the unit, see F and G94, G95; for chamfer/rounding, see CHF, CHR, RND
FRCM	Modal feedrate for chamfer/rounding	0, >0	In case FRCM=0: Feedrate Fwill act	For the unit, see F and G94, G95; for rounding/modal rounding, see RND, RNDM
FXS [*axis*]	Travel to fixed stop	=1: Selection =0: Deselection	*Axis:* Use the machine identifier	N20 G1 X10 Z25 FXS[Z1]=1 FXST[Z1]=12.3 FXSW[Z1]=2 F...
FXST [*axis*]	Clamping torque, travel to fixed stop	> 0.0 ... 100.0	in %, max. 100% from the max. torque of the drive, *axis:* Use the machine identifier	N30 FXST[Z1]=12.3
FXSW [*axis*]	Monitoring window, travel to fixed stop	> 0.0	Unit of measurement mm or degrees, axis-specific, *axis:* Use the machine identifier	N40 FXSW[Z1]=2.4
GOTOB	GoBack instruction	–	A GoTo operation is performed to a block marked by a label; the jump destination is in the direction of the pro-	N10 LABEL1: ... ...

Address	Meaning	Value Assignment	Information	Programming
OFFN	Groove width with TRACYL, otherwise specification of stock allowance	–	Only effective with the tool radius compensation G41, G42 active	N10 OFFN=12.4
RND	Rounding	0.010 ... 99 999.999	Inserts a rounding with the specified radius value tangentially between two contour blocks; special feedrate FRC= ... possible	N10 X... Y.... **RND=4.5** N11 X... Y...
RNDM	Modal rounding	0.010 ... 99 999.999 0	– Inserts roundings with the specified radius value tangentially at the following contour corners; special feedrate possible: FRCM= ... – Modal rounding OFF	N10 X... Y.... **RNDM=.7.3** ;modal rounding ON N11 X... Y... .... N100 **RNDM=.0** ;modal rounding OFF
RP	Polar radius	0.001 ... 99 999.999	Traversing in polar coordinates, definition of the pole; in addition: AP – polar angle	see G0, G1, G2; G3 G110, G111, G112
RPL	Angle of rotation with ROT, AROT	±0.00001 ... 359.9999	Specification in degrees; angle for a programmable rotation in the current plane G17 to G19	see ROT, AROT
SET(, , ,) REP()	Set values for the variable fields		SET: Various values, from the specified element up to: according to the number of values REP: the same value, from the specified element up to the end of the field	DEF REAL VAR2[12]=REP(4.5) ; all elements value 4.5 N10 R10=SET(1.1,2.3,4.4) ; R10=1.1, R11=2.3, R4=4.4
SF	Thread starting point when using G33	0.001 ... 359.999	Specified in degrees; the thread commencement point with G33 is offset by the specified value (not relevant for tapping)	See G33
SPI(n)	converts the spindle number n into axis identifier		n= 1 or n= 2 axis identifier: e.g. "SP1" or "C"	
SPOS	Spindle position	0.0000 ... 359.9999 If specified incrementally (IC): ±0.001 ... 99 999.999	specified in degrees; the spindle stops at the specified position (to achieve this, the spindle must provide the appropriate technical prerequisites: position control)	N10 SPOS=.... N10 SPOS=ACP(...) N10 SPOS=ACN(...) N10 SPOS=IC(...) N10 SPOS=DC(...)
STOPFIFO	Stops the fast machining step	–	Special function; filling of the buffer memory until STARTFIFO, "Buffer memory full" or "End of program" is detected.	STOPFIFO ;separate block, start of filling N10 X... N20 X...
START-FIFO	Start of fast machining step	–	Special function; the buffer memory is filled at the same time.	N30 X... STARTFIFO ;separate block, end of filling
STOPRE	Preprocessing stop	–	Special function; the next block is only decoded if the block before STOPRE is completed.	STOPRE ;separate block
TANG(Fo, Le1,Le2....)	Tangential control, definition	–	Fo: Name of the following axis) Le1: Name of master axis 1 Le2: Name of master axis 2 Further parameters optional This function is only available for the SINUMERIK 802Dsl pro.	TANG(C,X,Y) ; separate block TANG(C,X,Y,1"W","P") ; Max. number of parameters

Address	Meaning	Value Assignment	Information	Programming
TANGON (Fo,...)	Activate tangential control	–	Fo: Name of following axis (rotary axis) This function is only available for the SINUMERIK 802Dsl pro.	TANGON(C) ; Separate block TANGON(C,angle,dist,angletol) ; Max. number of parameters
TANGOF (Fo)	Deactivate tangential control	–	Fo: Name of following axis (rotary axis) This function is only available for the SINUMERIK 802Dsl pro.	TANGOF(C) ; separate block
TANGDEL (Fo)	Tangential control, delete definition	–	Fo: Name of following axis (rotary axis) This function is only available for the SINUMERIK 802Dsl pro.	TANGDEL(C) ; separate block
TLIFT(Fo)	Tangential control, insert intermediate block	–	Fo: Name of following axis (rotary axis) This function is only available for the SINUMERIK 802Dsl pro.	TLIFT(C) ; separate block
TRACYL(d)	Milling of the face end	d: 1.000 ... 99 999.999	Kinematic transformation	TRACYL(20.4) ; separate block ; Cylinder diameter: 20.4 mm TRACYL(20.4,1) ; also possible
TRAFOOF	Deactivate TRACYL	–	Disables all kinematic transformations	TRAFOOF ; separate block
TURN	Number of additional circle passes with helix interpolation	0 ... 999	in conjunction with circular interpolation G2/G3 in a plane G17 to G19 and infeed motion of the axis standing vertically on the plane	N10 G0 G17 X20 Y5 Z3 N20 G1 Z–5 F50 N30 G3 X20 Y5 Z–20 I0 J7.5 **TURN=2** **;** in total, 3 full circles

10.2.2 Positional data

➢ Linear interpolation with rapid traverse: G00

Functionality

The rapid traverse movement G0 is used for rapid positioning of the tool, but not for **direct workpiece machining**.

All the axes can be traversed simultaneously – on a straight path.

For each axis, the maximum speed (rapid traverse) is defined in machine data. If only one axis traverses, it uses its rapid traverse. If two or three axes are traversed simultaneously, the path velocity (e.g. the resulting velocity at the tool tip) must be selected such that the **maximum possible path velocity** with consideration of all axes involved results.

A programmed feedrate (F word) has no meaning for G0. G2/G3 remains active until canceled by another instruction from this G group (G0, G1, G3, ...).

G0 X... Y... Z... ; Cartesian coordinates

G0 AP=... RP=... ; Polar coordinates

G0 AP=... RP=... Z... ; Cylinder coordinates (3-dimensional)

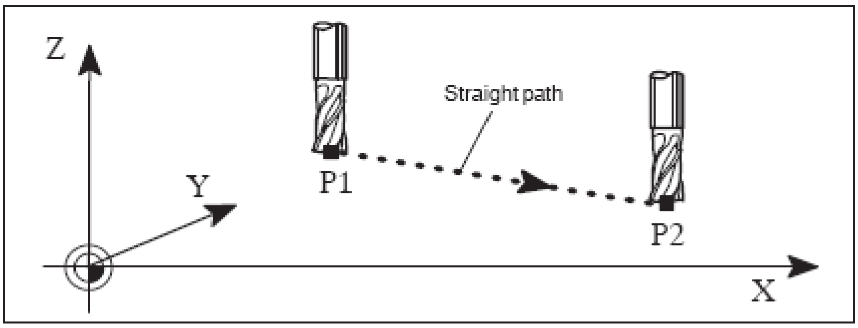


Fig 10.2－4

Programming

G0 X... Y... Z... ; Cartesian coordinates

G0 AP=... RP=... ; Polar coordinates

G0 AP=... RP=... Z... ; Cylinder coordinates (3-dimensional)

Note: Another option for linear programming is available with the angle specification

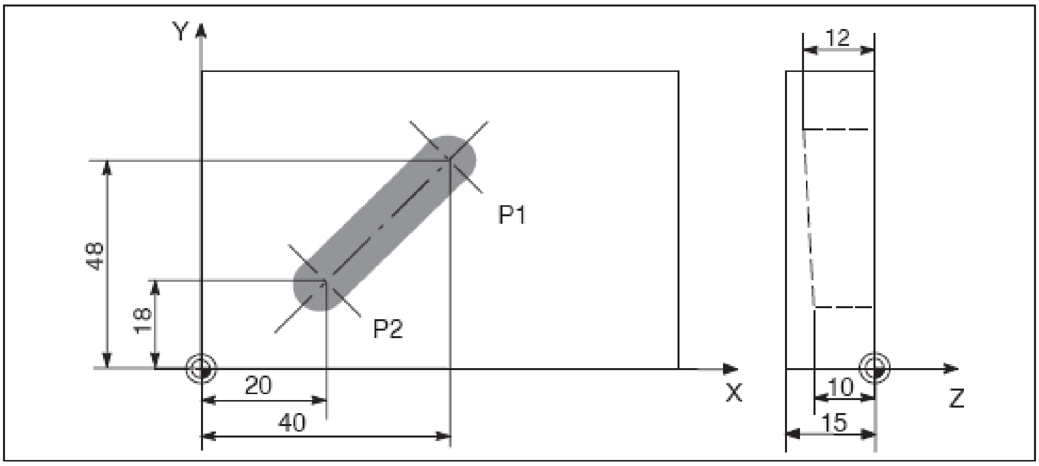


Fig 10.2－5

Programming example

N10 G0 X100 Y150 Z65 ; Cartesian coordinate

...

N50 G0 RP=16.78 AP=45 ; Polar coordinate

Information

Another group of G functions exists for movement to the position (see Section 8.3.16 ”Exact stop / continuous-path control mode: G60, G64”).

For G60 exact stop, a window with various precision values can be selected with another G group. For exact stop, an alternative instruction with non-modal effectiveness exists: G9. You should consider these options for adaptation to your positioning tasks.

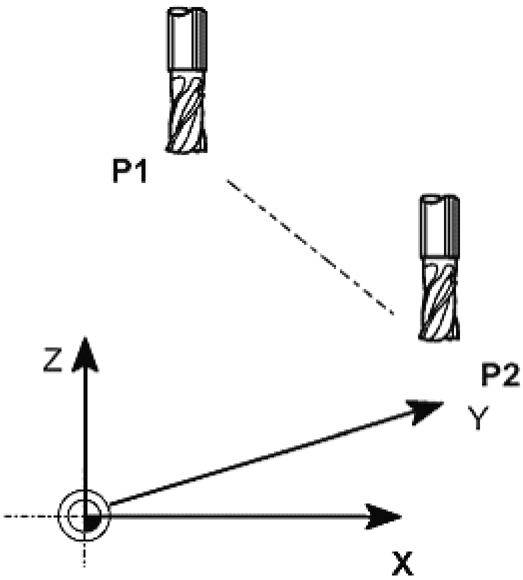


Fig 10.2-6

Linear interpolation with feedrate: G1

Functionality

The tool moves from the starting point to the end point along a straight path. The **path velocity** is determined by the programmed **F word**.

All axes can be traversed simultaneously.

G2/G3 remains active until canceled by another instruction from this G group (G0, G2, G3, ...).

Programming

G1 X... Y... Z... F... ; Cartesian coordinates

G1 AP=... RP=... F... ; Polar coordinates

G1 AP=... RP=... Z... F... ; Cylinder coordinates (3-dimensional)

Note: Another option for linear programming is available with the angle specification ANG=... (see Section 8.5.2 ”Blueprint programming”).

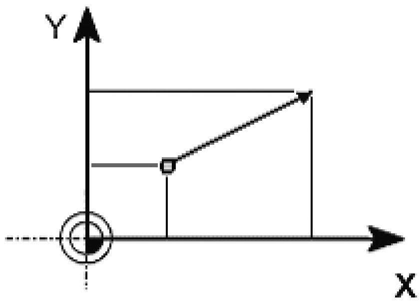


Fig10.2-7

Programming example

N05 G0 G90 X40 Y48 Z2 S500 M3 ; Tools traverses at rapid traverse to P1,

3 axes simultaneously,

spindle speed = 500 r.p.m., CW rotation

N10 G1 Z–12 F100 ; Infeed to Z–12, feedrate 100 mm/min

N15 X20 Y18 Z–10 ; Tool traverses along a straight line in the space

to P2

N20 G0 Z100 ; Traversing at rapid traverse

N25 X-20 Y80

N30 M2 ; End of program

To machine a workpiece, spindle speed S ... and direction M3/M4 are required (see Section ”Spindle movement”).

Circular interpolation: G2, G3

Functionality

The tool moves from the starting point to the end point along a circular path. The direction is determined by the G function:

G2 ; CW

G3 ; CCW

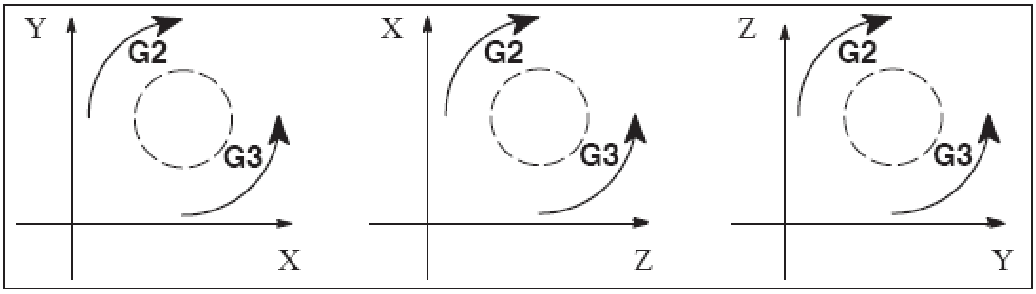


Fig10.2-8

The description of the desired circle can be given in various ways:

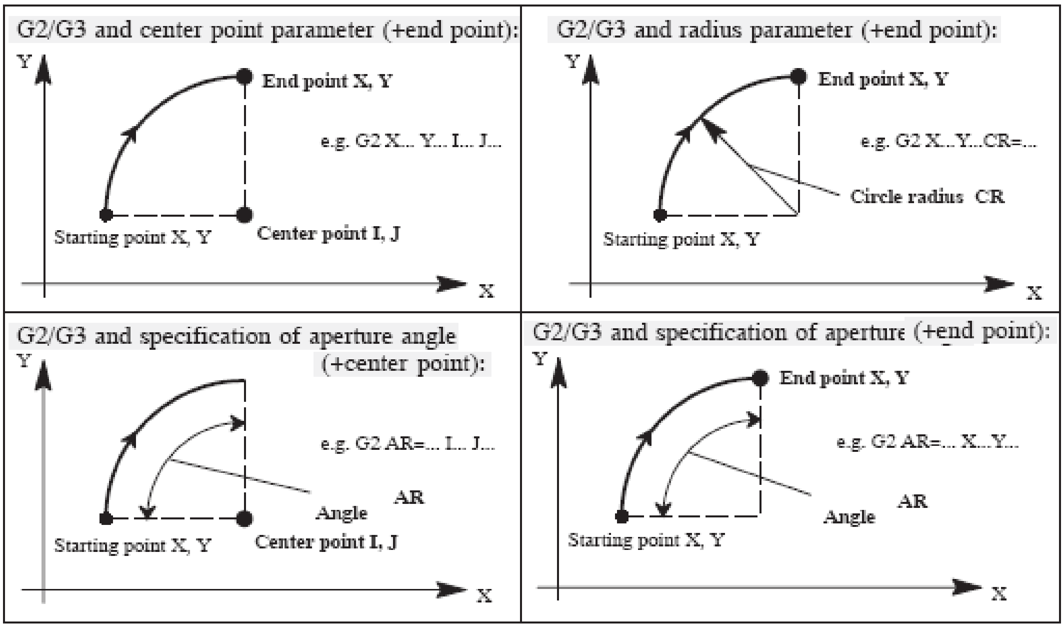


Fig10.2-9

Programming

G2/G3 X... Y... I... J... ; Center and end points

G2/G3 CR=... X... Y... ; Circle radius and end point

G2/G3 AR=... I... J... ; Aperture angle and center point

G2/G3 AR=... X... Y... ; Aperture angle and end point

G2/G3 AP=... RP =... ; Polar coordinates, circle around the pole

Note

Further possibilities for circle programming result from:

CT – circle with tangential connection and

CIP – circle via intermediate point (see next sections).

Input tolerances for the circle

Circles are only accepted by the control system with a certain dimensional tolerance.

The circle radius at the starting and end points are compared here. If the difference is within the tolerance, the center point is exactly set internally. Otherwise, an alarm message is issued.

The tolerance value can be set via machine data (see ”Start-up Guide” 802Dsl).

Information

Full circles in a block are only possible if the center point and the end point are specified. For circles with radius specification, the arithmetic sign of CR =... is used to select the correct circle. It is possible to program 2 circles with the same starting and end points, as well as with the same radius and the same direction. The negative sign in front of CR =–... determines the circle whose circle segment is greater than a semi-circle; otherwise, the circle with the circle segment is less than or equal to the semi-circle and determined as follows:

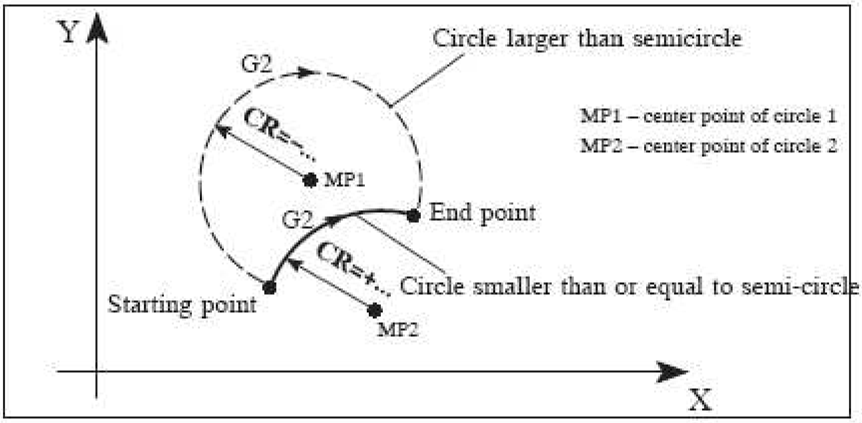


Fig10.2-10

Programming example: Definition of center point and end point

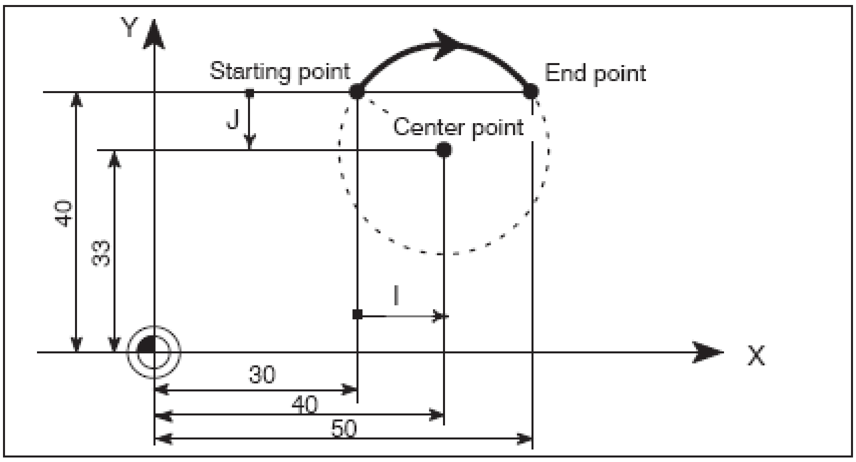


Fig10.2-11

N5 G90 X30 Y40 ; Circle starting point for N10

N10 G2 X50 Y40 I10 J-7 ; End point and center point

Note: Center point values refer to the circle starting point!

Programming example: End point and radius specification

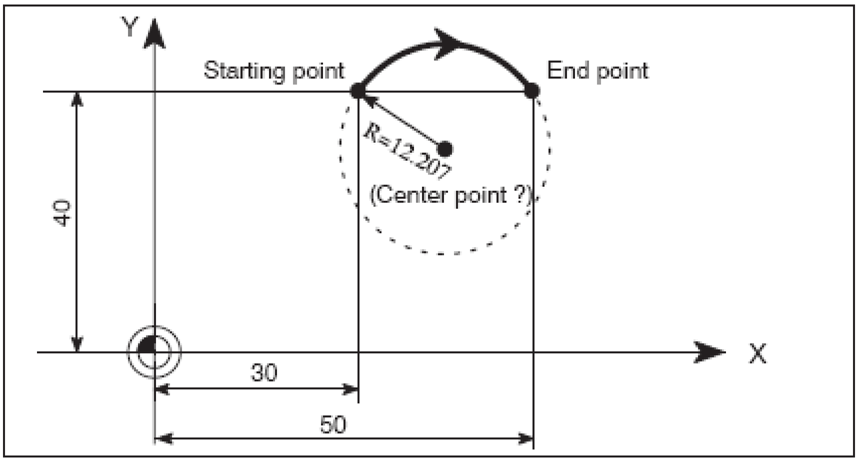


Fig10.2-12

N5 G90 X30 Y40 ; Circle starting point for N10

N10 G2 X50 Y40 CR=12.207 ; End point and radius

Note: With a negative leading sign for the value with CR=–..., a circular segment larger than a semi-circle is selected.

Helix interpolation: G2/G3, TURN

Functionality

With helix interpolation, two movements are overlaid:

– circular movement in plane G17 or G18 or G19

– linear movement of the axis standing vertically on this plane.

The number of additional full–circle passes is programmed with TURN=. These are added to the actual circle programming.

The helix interpolation can preferably be used for the milling of threads or of lubricating grooves in cylinders.

Programming

G2/G3 X... Y... I... J... TURN =... ; Center and end points

G2/G3 CR = ... X... Y... TURN =... ; Circle radius and end point

G2/G3 AR = ... I... J... TURN =... ; Aperture angle and center point

G2/G3 AR = ... X... Y... TURN =... ; Aperture angle and end point

G2/G3 AP =... RP =... TURN =... ; Polar coordinates, circle around the pole

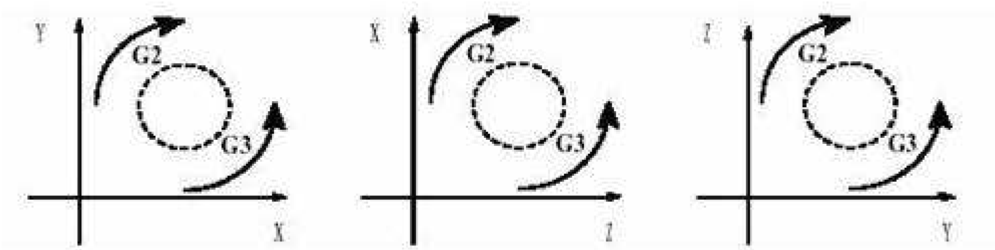


Fig10.2-13

Programming example

N10 G17 ; X/Y plane, Z standing vertically on it

N20 ... Z ...

N30 G1 X0 Y50 F300 ; Approach starting point

N40 G3 X0 Y0 Z33 I0 J–25 TURN= 3 ; Helix

...

Thread cutting with constant lead: G33

Functionality

This requires a spindle with position measuring system.

The function G33 can be used to machine threads with constant lead of the following type:

If an appropriate tool is used, tapping with compensating chuck is possible.

The compensating chuck compensates the resulting path differences to a certain limited degree.

The drilling depth is specified by specifying one of the axes X, Y or Z; the spindle lead is specified via the relevant I, J or K.

G33 remains active until canceled by another instruction from this G group (G0, G1, G2, G3, ...).

Right-hand or left-hand threads

Right-hand or left-hand threads are set with the rotation direction of the spindle (M3 right (CW), M4 left (CCW) – see Section 8.4 ”Spindle movement”). To this end, the speed must be programmed under the address S or an appropriate speed must be set.

Remark:

A complete cycle of tapping with compensating chuck is provided by the standard cycle CYCLE840.

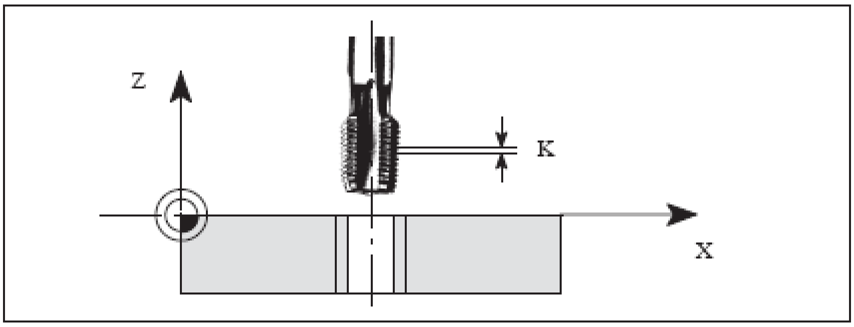


Fig10.2-14

Programming example

metric thread 5,

pitch as per table: 0.8 mm/rev., tap hole already premachined:

N10 G54 G0 G90 X10 Y10 Z5 S600 M3 ; Approach starting point, spindle rotation CW

N20 G33 Z-25 K0.8 ; Tapping, end point –25 mm

N40 Z5 K0.8 M4 ; Retraction, spindle rotation CCW

N50 G0 X... Y... Z...

Axis velocity

With G33 threads, the velocity of the axis for the thread lengths is determined on the basis of the spindle speed and the thread pitch. The **feedrate F is not relevant**. It is, however, stored. However, the maximum axis velocity (rapid traverse) defined in the machine data can not be exceeded. This will result in an alarm.

Information

Important

_ The spindle speed override switch should remain unchanged for thread machining.

_ The feedrate override switch has no meaning in this block.

Tapping with compensating chuck: G63

Functionality

G63 can be used for tapping with compensating chuck. The programmed feedrate F must match with the spindle speed S (programmed under the address ”S” or specified speed) and with the thread pitch of the drill:

F [mm/min] = S [r.p.m.] x thread pitch [mm/rev.]

The compensating chuck compensates the resulting path differences to a certain limited degree.

The drill is retracted using G63, too, but with the spindle rotating in the opposite direction M3 –<–> M4.

G63 is non-modal. In the block after G63, the previous G command of the ”Interpolation type” group (G0, G1,G2, ...) is active again.

Right-hand or left-hand threads

Right-hand or left-hand threads are set with the rotation direction of the spindle (M3 right (CW), M4 left (CCW) – see Section 8.4 ”Spindle movement”).

Remark:

The standard cycle CYCLE840 provides a complete tapping cycle with compensating chuck (but with G33 and the relevant prerequisites).

Programming example

metric thread 5,

pitch as per table: 0.8 mm/rev., tap hole already premachined:

N10 G54 G0 G90 X10 Y10 Z5 S600 M3 ; Approach starting point, spindle rotation CW

N20 G63 Z-25 F480 ; Tapping, end point –25 mm

N40 G63 Z5 M4 ; Retraction, spindle rotation CCW

N50 X... Y... Z...

Fixed point approach: G75

Functionality

By using G75, a fixed point on the machine, e.g. tool change point, can be approached.

The position is stored permanently in the machine data for all axes. No offset is effective. The speed of each axis is its rapid traverse.

G75 requires a separate block and is non-modal. The machine axis identifier must be programmed!

In the block after G75, the previous G command of the ”Interpolation type” group (G0, G1,G2, ...) is active again.

Programming example

N10 G75 X1 = 0 Y1 = 0 Z1 = 0

Remark: The programmed position values for X1, Y1 (any value, here = 0) are ignored, but must still be written.

Reference point approach: G74

Functionality

The reference point can be approached in the NC program with G74. The direction and speed of each axis are stored in machine data.

G74 requires a separate block and is non-modal. The machine axis identifier must be programmed!

In the block after G74, the previous G command of the ”Interpolation type” group (G0, G1,G2, ...) is active again.

Programming example

N10 G74 X1 = 0 Y1 = 0 Z1 = 0

Remark: The programmed position values for X1, Y1 (any value, here = 0) are ignored, but must still be written.

Unit of measure for F with G94, G95

The dimension unit for the F word is determined by G functions:

_ G94 **F** as the feedrate in**mm/min**

_ G95 **F** as the feedrate in **mm/rev.** of the spindle

(only meaningful when the spindle is running)

Remark**:**

This unit of measure applies to metric dimensions. According to Section ”Metric and inch dimensioning”, settings with inch dimensioning are also possible.

Programming example

N10 G94 F310 ; Feedrate in mm/min

...

N110 S200 M3 ; Spindle rotation

N120 G95 F15.5 ; Feedrate in mm/rev.

Remark: Write a new F word if you change G94 – G95.

Exact stop / continuous-path control mode: G9, G60, G64

Functionality

G functions are provided for optimum adaptation to different requirements to set the traversing behavior at the block borders and for block advancing. Example: For example, you would like to quickly position with the axes or you would like to machine path contours over multiple blocks.

Programming

G60 ; Exact stop – modal

G64 ; Continuous-path control mode

G9 ; Exact stop – non-modal

G601 ; Exact stop window fine

G602 ; Exact stop window coarse

Exact stop G60, G9

If the exact stop function (G60 or G9) is active, the velocity for reaching the exact end position at the end of a block is decelerated to zero.

Another modal G group can be used here to set when the traversing movement of this block is considered ended and the next block is started.

_ G601 ; Exact stop window fine

Block advance takes place when all axes have reached the ”Exact stop window fine” (value in the machine data).

_ G602 ; Exact stop window coarse

Block advance takes place when all axes have reached the ”Exact stop window coarse”

The selection of the exact stop window has a significant influence on the total time if many positioning operations are executed. Fine adjustments require more time.

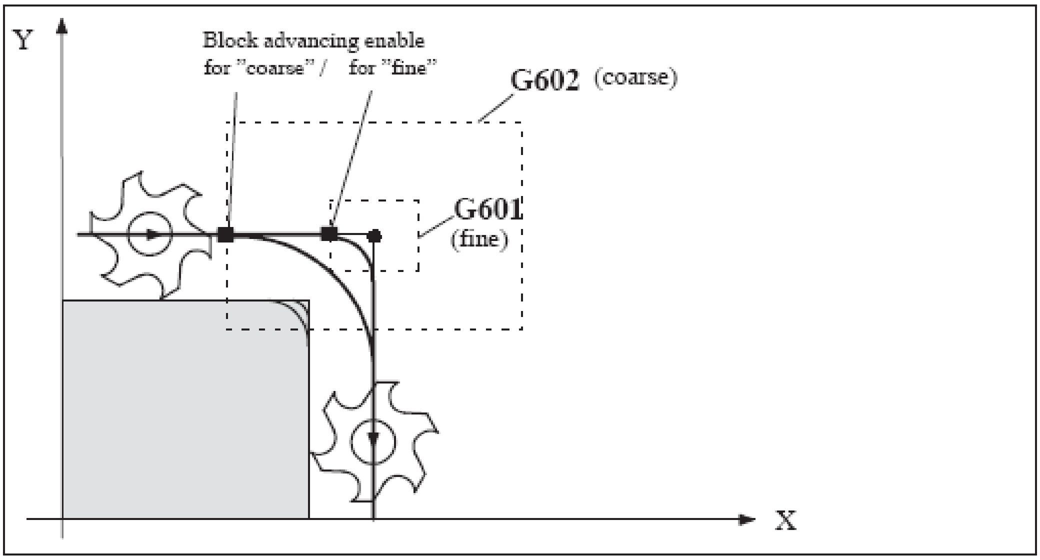


Fig10.2-15

Programming example

N5 G602 ; Exact stop window coarse

N10 G0 G60 X... ; Exact stop modal

N20 X... Y... ; G60 remains active

...

N50 G1 G601 ... ; Exact stop window fine

N80 G64 X... ; Switching to continuous-path control mode

...

N100 G0 G9 X... ; Exact stop is only effective for this block

N111 ... ; Continuous-path control mode again

Remark: The G9 command only generates exact stop for the block in which it is programmed; G60, however, is effective until it is canceled by G64.

Continuous-path control mode G64

The objective of the continuous-path control mode is to avoid deceleration at the block boundaries and to switch **to the next block** with **a path velocity as constant as possible** (in the case of tangential transitions). The function works with **look-ahead velocity control** over several blocks.

For non-tangential transitions (corners), the velocity can reduced rapidly enough so that the axes are subject to a relatively high velocity change over a short time. This may lead to a significant jerk (acceleration change). The size of the jerk can be limited by activating the SOFT function.

Programming example

N10 G64 G1 X... F... ; Continuous-path control mode

N20 Y.. ; Continuous-path control mode continues to be active

...

N180 G60 ... ; switching to exact stop

Look-ahead velocity control

In the continuous-path control mode with G64, the control system automatically determines the velocity control for several NC block in advance. This enables acceleration and deceleration across multiple blocks with approximately tangential transitions. For paths that consist of short travels in the NC blocks, higher velocities can be achieved than without look ahead.

Spindle speed limitation: G25, G26

Functionality

In the program, you can limit the limit values that would otherwise apply for a controlled spindle by writing G25 or G26 and the spindle address S with the speed limit value.

This overwrites the values entered in the setting data at the same time.

G25 and G26 each require a separate block. A previously programmed speed S is maintained.

Programming

G25 S... ; Programmable lower spindle speed limitation

G26 S... ; Upper speed limitation

Information

The outmost limits of the spindle speed are set in machine data. Appropriate inputs via the operator panel can activate various setting data for further limiting.

Programming example

N10 G25 S12 ; Lower spindle limit speed: 12 r.p.m.

N20 G26 S700 ; Upper spindle limit speed : 700 r.p.m.

Note

G25/G26 are used in conjunction with axis addresses for a working area limitation (see Section ”Working area limitation”).

Selecting the tool radius compensation: G41, G42

Functionality

The control system is working with tool radius compensation in the selected plane G17 to G19.

A tool with a corresponding D number must be active. The tool radius compensation is activated by G41/G42. The control system automatically calculates the required equidistant tool paths for the programmed contour for the respective current tool radius.

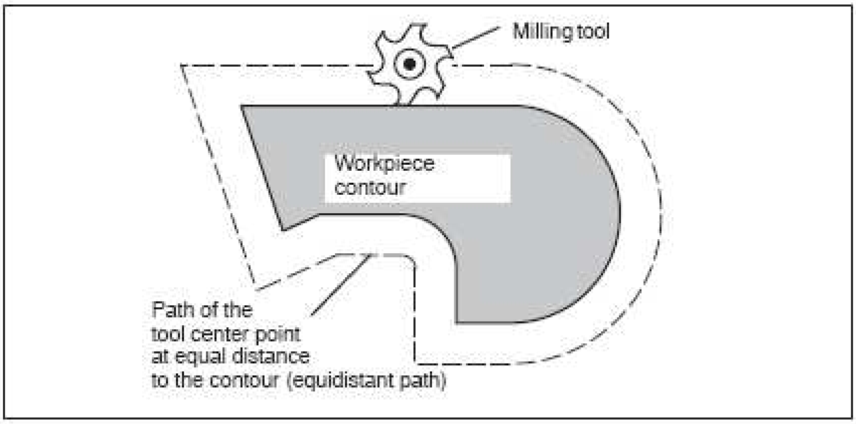


Fig10.2-16

Programming

G41 X... Y... ; Tool radius compensation left of the contour

G42 X... Y... ; Tool radius compensation right of the contour

Remark: The selection can only be made for linear interpolation (G0, G1).

Program both axes of the plane (e.g. with G17: X, Y). If you only specify one axis, the second axis is automatically completed with the last programmed value.

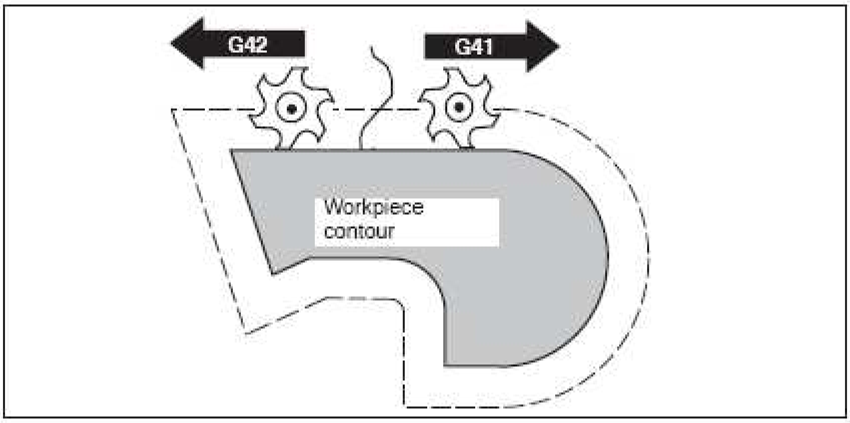


Fig10.2-17

Tool radius compensation OFF: G40

Functionality

The compensation mode (G41/G42) is deselected with G40. G40 is also the activation position at the beginning of the program.

The tool ends the **block in front of G40** in the normal position (compensation vector vertically to the tangent at the end point);

If G40 is active, the reference point is the tool center point. Subsequently, when deselected, the tool tip approaches the programmed point.

Always select the end point of the G40 block such that collision-free traversing is guaranteed!

Programming

G40 X... Y... ; Tool radius compensation OFF

Remark: The compensation mode can only be deselected with linear interpolation (G0, G1).

10.3 Overview of cycles

Cycles are generally applicable technology subroutines that can be used to carry out a specific machining process, such as drilling of a thread (tapping) or milling of a pocket. These cycles are adapted to individual tasks by parameter assignment.

Drilling cycle, drilling pattern cycles and milling cycles

The following standard cycles can be carried out using the SINUMERIK 802D control system:

_ Drilling cycles

CYCLE81 Drilling, centering

CYCLE82 Drilling, counterboring

CYCLE83 Deep hole drilling

CYCLE84 Rigid tapping

CYCLE84 Tapping with compensating chuck

CYCLE85 Reaming 1 (boring out 1)

CYCLE86 Boring (boring out 2)

CYCLE87 Drilling with stop 1 (boring out 3)

CYCLE87 Drilling with stop 2 (boring out 4)

CYCLE85 Reaming 2 (boring out 5)

With SINUMERIK 840D, the boring cycles CYCLE85 ... CYCLE89 are called boring 1 ... boring 5, but are nevertheless identical in their function.

_ Drill pattern cycles

HOLES1 Row of holes

HOLES2 Circle of holes

_ Milling cycles

CYCLE71 Face milling

CYCLE72 Contour milling

CYCLE76 Rectangular spigot milling

CYCLE77 Circular spigot milling

LONGHOLE Long hole

SLOT1 Milling pattern ’Slots on a circle’

SLOT2 Milling pattern ”Circular slots”

POCKET3 Rechtecktasche fräsen (mit beliebigem Fräser)

POCKET4 Milling of rectangular pocket (using any milling cutter)

CYCLE90 Thread milling

The cycles are supplied with the tool box. They are loaded via the RS232 interface into the part program memory during the start-up of the control system.

Auxiliary cycle subroutines

The cycle package includes the following auxiliary subroutines:

_ cyclesm.spf

_ steigung.spf and

_ meldung.spf

These must always be loaded in the control.

Drilling, centering – CYCLE81

Programming

CYCLE81(RTP, RFP, SDIS, DP, DPR)

RTP real Retraction plane (absolute)

RFP real Reference plane (absolute)

SDIS real Safety clearance (enter without sign)

DP real Final drilling depth (absolute)

DPR real Final drilling depth relative to the reference plane (enter without sign)

Function

The tool drills at the programmed spindle speed and feedrate to the entered final drilling depth.

Drilling, counterboring – CYCLE82

Programming

CYCLE82(RTP, RFP, SDIS, DP, DPR, DTB)

Parameters

Table 9-4 Parameters for CYCLE82

RTP real Retraction plane (absolute)

RFP real Reference plane (absolute)

SDIS real Safety clearance (enter without sign)

DP real Final drilling depth (absolute)

DPR real Final drilling depth relative to the reference plane (enter without sign)

DTB real Dwell time at final drilling depth (chip breaking)

Function

The tool drills at the programmed spindle speed and feedrate to the entered final drilling depth. A dwell time can be allowed to elapse when the final drilling depth has been reached.

Sequence

Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.

The cycle creates the following sequence of motions:

_ Approach of the reference plane brought forward by the safety clearance by using G0

_ Traversing to the final drilling depth with the feedrate (G1) programmed prior to the cycle call

_ Dwell time at final drilling depth

_ Retraction to the retraction plane with G0

Explanation of the parameters

For the parameters RTP, RFP, SDIS, DP, DPR, see CYCLE81

Deep hole drilling – CYCLE83

Programming

CYCLE83(RTP, RFP, SDIS, DP, DPR, FDEP, FDPR, DAM, DTB, DTS, FRF, VARI)

Parameters

Table 9-5 Parameters for CYCLE83

RTP real Retraction plane (absolute)

RFP real Reference plane (absolute)

SDIS real Safety clearance (enter without sign)

DP real Final drilling depth (absolute)

DPR real Final drilling depth relative to the reference plane (enter without

sign)

FDEP real First drilling depth (absolute)

FDPR real First drilling depth relative to the reference plane (enter without sign)

DAM real Amount of degression (enter without sign)

DTB real Dwell time at final drilling depth (chip breaking)

DTS real Dwell time at starting point and for swarf removal

FRF real Feedrate factor for the first drilling depth (enter without sign)

Range of values: 0.001 ... 1

VARI int Machining type:

Chip breaking = 0

Swarf removal = 1

Function

The tool drills at the programmed spindle speed and feedrate to the entered final drilling depth.

Deep hole drilling is performed with a depth infeed of a maximum definable depth executed several times, increasing gradually until the final drilling depth is reached.

The drill can either be retracted to the reference plane + safety clearance after every infeed depth for swarf removal or retracted in each case by 1 mm for chip breaking.

Sequence

Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.

Function

The tool drills at the programmed spindle speed and feedrate to the entered final drilling depth.

Deep hole drilling is performed with a depth infeed of a maximum definable depth executed several times, increasing gradually until the final drilling depth is reached.

The drill can either be retracted to the reference plane + safety clearance after every infeed depth for swarf removal or retracted in each case by 1 mm for chip breaking.

Sequence

Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.

Rigid tapping – CYCLE84

Programming

CYCLE84 (RTP, RFP, SDIS, DP, DPR, DTB, SDAC, MPIT, PIT, POSS, SST, SST1)

Parameters

Table 9-6 Parameters for CYCLE84

RTP real Retraction plane (absolute)

RFP real Reference plane (absolute)

SDIS real Safety clearance (enter without sign)

DP real Final drilling depth (absolute)

DPR real Final drilling depth relative to the reference plane (enter without sign)

DTB real Dwell time at thread depth (chip breaking)

SDAC int Direction of rotation after end of cycle

Values: 3, 4 or 5 (for M3, M4 or M5)

MPIT real Pitch as a thread size (signed):

Range of values 3 (for M3) ... 48 (for M48); the sign determines the direction of rotation in the thread

PIT real Pitch as a value (signed)

Value range: 0.001 ... 2000.000 mm); the sign determines the direction of rotation in the thread

POSS real Spindle position for oriented spindle stop in the cycle (in degrees)

SST real Speed for tapping

SST1 real Speed for retraction

Function

The tool drills at the programmed spindle speed and feedrate to the entered final thread depth.

CYCLE84 can be used to perform rigid tapping operations. For tapping with compensating chuck, a separate cycle CYCLE840 is provided.

Sequence

Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.

The cycle creates the following sequence of motions:

_ Approach of the reference plane brought forward by the safety clearance by using G0

_ Oriented spindle stop (value in the parameter POSS) and switching the spindle to axis mode

_ Tapping to final drilling depth and speed SST

_ Dwell time at thread depth (parameter DTB)

_ Retraction to the reference plane brought forward by the safety clearance, speed SST1 and direction reversal

_ Retraction to the retraction plane with G0; spindle mode is reinitiated by reprogramming the spindle speed active before the cycle was called and the direction of rotation programmed under SDAC

Explanation of the parameters

For the parameters RTP, RFP, SDIS, DP, DPR, see CYCLE81

Tapping with compensating chuck – CYCLE840

Programming

CYCLE840 (RTP, RFP, SDIS, DP, DPR, DTB, SDR, SDAC, ENC, MPIT, PIT)

Parameters

RTP real Retraction plane (absolute)

RFP real Reference plane (absolute)

SDIS real Safety clearance (enter without sign)

DP real Final drilling depth (absolute)

DPR real Final drilling depth relative to the reference plane (enter without sign)

DTB real Dwell time at thread depth (chip breaking)

SDR int Direction of rotation for retraction

Values: 0 (automatic reversal of direction of rotation)

3 or 4 (for M3 or M4)

SDAC int Direction of rotation after end of cycle

Values: 3, 4 or 5 (for M3, M4 or M5)

ENC int Tapping with/without encoder

Values: 0 = with encoder

1 = without encoder

MPIT real Pitch as a thread size (signed):

Range of values 3 (for M3) ... 48 (for M60)

PIT real Pitch as a value (signed)

Value range: 0.001 ... 2,000.000 mm

Function

The tool drills at the programmed spindle speed and feedrate to the entered final thread depth.

Use this cycle to perform tapping with compensating chuck

_ without encoder and

_ with encoder.

Sequence of operations: Tapping with compensating chuck without encoder

Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.

Reaming 1 (boring 1) – CYCLE85

Programming

CYCLE85(RTP, RFP, SDIS, DP, DPR, DTB, FFR, RFF)

Parameters

Table 9-8 Parameters for CYCLE85

RTP real Retraction plane (absolute)

RFP real Reference plane (absolute)

SDIS real Safety clearance (enter without sign)

DP real Final drilling depth (absolute)

DPR real Final drilling depth relative to the reference plane (enter without sign)

DTB real Dwell time at final drilling depth (chip breaking)

FFR real Feedrate

RFF real Retraction feedrate

Function

The tool drills at the programmed spindle speed and feedrate to the entered final drilling depth.

The inward and outward movement is performed at the feedrate assigned to FFR and RFF respectively.

Sequence

Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.

The cycle creates the following sequence of motions:

_ Approach of the reference plane brought forward by the safety clearance by using G0

_ Traversing to the final drilling depth with G1 and at the feedrate programmed under the parameter FFR

_ Dwell time at final drilling depth

_ Retraction to the reference plane brought forward by the safety clearance with G1 and the retraction feedrate defined under the parameter RFF

_ Retraction to the retraction plane with G0

Boring (boring 2) – CYCLE86

Programming

CYCLE86 (RTP, RFP, SDIS, DP, DPR, DTB, SDIR, RPA, RPO, RPAP, POSS)

Parameters

Table 9-9 Parameters for CYCLE86

RTP real Retraction plane (absolute)

RFP real Reference plane (absolute)

SDIS real Safety clearance (enter without sign)

DP real Final drilling depth (absolute)

DPR real Final drilling depth relative to the reference plane (enter without sign)

DTB real Dwell time at final drilling depth (chip breaking)

SDIR int Direction of rotation

Values: 3 (for M3)

4 (for M4)

RPA real Retraction path along the 1st axis of the plane (incremental, enter with sign)

RPO real Retraction path along the 2nd axis of the plane (incremental, enter with sign)

RPAP real Retraction path along the boring axis (incremental, enter with sign)

POSS real Spindle position for oriented spindle stop in the cycle (in degrees)

Function

The cycle supports the boring of holes with a boring bar.

The tool drills at the programmed spindle speed and feedrate to the entered final drilling depth.

With boring 2, oriented spindle stop is activated once the drilling depth has been reached.

Then, the programmed retraction positions are approached in rapid traverse and, from there, the retraction plane.

Sequence

Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.

The cycle creates the following sequence of motions:

_ Approach of the reference plane brought forward by the safety clearance by using G0

_ Traversing to final drilling depth with G1 and the feedrate programmed prior to the cycle call

_ Dwell time to final drilling depth

_ Oriented spindle stop at the spindle position programmed under POSS

_ Traverse retraction path in up to three axes with G0

_ Retraction in the boring axis to the reference plane brought forward by the safety clearance by using G0

_ Retraction to the retraction plane with G0 (initial drilling position in both axes of the plane)

Boring with Stop 1 (boring 3) – CYCLE87

Programming

CYCLE87 (RTP, RFP, SDIS, DP, DPR, SDIR)

Parameters

RTP real Retraction plane (absolute)

RFP real Reference plane (absolute)

SDIS real Safety clearance (enter without sign)

DP real Final drilling depth (absolute)

DPR real Final drilling depth relative to the reference plane (enter without sign)

SDIR int Direction of rotation

Values: 3 (for M3)

4 (for M4)

Function

The tool drills at the programmed spindle speed and feedrate to the entered final drilling depth.

During boring 3, a spindle stop without orientation M5 is generated after reaching the final drilling depth, followed by a programmed stop M0. Pressing the NC START key continues the retraction movement at rapid traverse until the retraction plane is reached.

Sequence

Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.

Drilling with stop 2 (boring 4) – CYCLE88

Programming

CYCLE88 (RTP, RFP, SDIS, DP, DPR, DTB, SDIR)

Parameters

RTP real Retraction plane (absolute)

RFP real Reference plane (absolute)

SDIS real Safety clearance (enter without sign)

DP real Final drilling depth (absolute)

DPR real Final drilling depth relative to the reference plane (enter without sign)

DTB real Dwell time at final drilling depth (chip breaking)

SDIR int Direction of rotation

Values: 3 (for M3)

4 (for M4)

Function

The tool drills at the programmed spindle speed and feedrate to the entered final drilling depth. When boring with stop, a spindle stop without orientation M5 and a programmed stop are generated when the final drilling depth is reached. Pressing the NC START key continues the retraction movement at rapid traverse until the retraction plane is reached.

Sequence

Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.

The cycle creates the following sequence of motions:

_ Approach of the reference plane brought forward by the safety clearance by using G0

_ Traversing to final drilling depth with G1 and the feedrate programmed prior to the cycle call

_ Dwell time at final drilling depth

_ Spindle and program stop with M5 M0. After program stop, press the NC START key.

_ Retraction to the retraction plane with G0

Reaming 2 (boring 5) – CYCLE89

Programming

CYCLE89 (RTP, RFP, SDIS, DP, DPR, DTB)

Parameters

RTP real Retraction plane (absolute)

RFP real Reference plane (absolute)

SDIS real Safety clearance (enter without sign)

DP real Final drilling depth (absolute)

DPR real Final drilling depth relative to the reference plane (enter without sign)

DTB real Dwell time at final drilling depth (chip breaking)

Function

The tool drills at the programmed spindle speed and feedrate to the entered final drilling depth. When the filan drilling depth is reached, the programmed dwell time is active.

Sequence

Position reached prior to cycle start:

The drilling position is the position in the two axes of the selected plane.

The cycle creates the following sequence of motions:

_ Approach of the reference plane brought forward by the safety clearance by using G0

_ Traversing to final drilling depth with G1 and the feedrate programmed prior to the cycle call

_ Dwell time to final drilling depth

_ Retraction up to the reference plane brought forward by the safety clearance using G1 and the same feedrate value

_ Retraction to the retraction plane with G0

Row of holes – HOLES1

Programming

HOLES1 (SPCA, SPCO, STA1, FDIS, DBH, NUM)

Parameters

SPCA real 1. axis of the plane (abscissa) of a reference point on the straight line (absolute)

SPCO real 2. axis of the plane (ordinate) of this reference point (absolute)

STA1 real Angle to the 1st axis of the plane (abscissa)

Value range: –180<STA1<=180 degrees

FDIS real Distance from the first hole to the reference point (enter without

sign)

DBH real Distance between the holes (enter without sign)

NUM int Number of holes

Function

This cycle can be used to produce a row of holes, i.e. a number of holes arranged along a straight line, or a grid of holes. The type of hole is determined by the drilling hole cycle that has already been called modally.

Sequence

To avoid unnecessary travel, the cycle calculates whether the row of holes is machined starting from the first hole or the last hole from the actual position of the plane axes and the geometry of the row of holes. The drilling positions are then approached one after the other at rapid traverse.

Circle of holes – HOLES2

Programming

HOLES2 (CPA, CPO, RAD, STA1, INDA, NUM)

Parameters

CPA real Center point of circle of holes (absolute), 1st axis of the plane

CPO real Center point of circle of holes (absolute), 2nd axis of the plane

RAD real Radius of circle of holes (enter without sign)

STA1 real Starting angle

Value range: –180<STA1<=180 degrees

INDA real Incrementing angle

NUM int Number of holes

Function

Use this circle to machine a circle of holes. The machining plane must be defined before the cycle is called.

The type of hole is determined by the drilling hole cycle that has already been called modally.

Figure 9-30

Face milling – CYCLE71

Programming

CYCLE71 (_RTP, _RFP, _SDIS, _DP, _PA, _PO, _LENG, _WID, _STA,_MID, _MIDA, _FDP, _FALD, _FFP1, _VARI, _FDP1)

Parameters

_RTP real Retraction plane (absolute)

_RFP real Reference plane (absolute)

_SDIS real Safety clearance (to be added to the reference plane; enter without sign)

_DP real Depth (absolute)

_PA real Starting point (absolute), 1st axis of the plane

_PO real Starting point (absolute), 2nd axis of the plane

_LENG real Rectangle length along the 1st axis, incremental.

The corner from which the dimension starts results from the sign.

_WID real Rectangle length along the 2nd axis, incremental.

The corner from which the dimension starts results from the sign.

_STA real Angle between the longitudinal axis of the rectangle and the

1st axis of the plane (abscissa, enter without sign);

Range of values: 0 ≤ _STA 180

_MID real Maximum infeed depth (enter without sign)

_MIDA real Maximum infeed width during solid machining in the plane as

a value (enter without sign)

_FDP real Retraction travel in the finishing direction (incremental,

enter without sign)

_FALD real Finishing dimension in the depth (incremental, enter without sign)

_FFP1 real Feedrate for surface machining

_VARI integer Machining type (enter without sign)

UNITS DIGIT

Values: 1 Roughing

2 Finishing

TENS DIGIT:

Values: 1 Parallel to the 1st axis of the plane, unidirectional

2 Parallel to the 2nd axis of the plane, unidirectional

3 Parallel to the 1st axis of the plane,

changing direction

4 Parallel to the 2nd axis of the plane,

changing direction

_FDP1 real Overrun travel in the direction of the plane infeed (incremental,

enter without sign)

Contour milling – CYCLE72

Programming

CYCLE72 (_KNAME, _RTP, _RFP, _SDIS, _DP, _MID, _FAL, _FALD, _FFP1, _FFD, _VARI, _RL, _AS1, _LP1, _FF3, _AS2, _LP2)

Parameters

_KNAME string Name of contour subroutine

_RTP real Retraction plane (absolute)

_RFP real Reference plane (absolute)

_SDIS real Safety clearance (to be added to the reference plane; enter without sign)

_DP real Depth (absolute)

_MID real Maximum infeed depth (incremental; enter without sign)

_FAL real Finishing allowance at the edge contour (enter without sign)

_FALD real Finishing allowance at the base (incremental, enter without sign)

_FFP1 real Feedrate for surface machining

_FFD real Feedrate for depth infeed (enter without sign)

_VARI integer Machining type (enter without sign)

UNITS DIGIT

Values: 1 Roughing

2 Finishing

TENS DIGIT:

Values: 0 Intermediate travel with G0

1 Intermediate travel with G1

HUNDREDS DIGIT

Values: 0...Retraction at end of contour to _RTP

1...Retraction at end of contour to _RFP + _SDIS

2 Retraction by _SDIS at end of contour

3 No retraction at end of contour

_RL integer Traveling around the contour either centrally, to the right or to the left (with G40, G41 or G42; enter without sign)

Values: 40...G40 (approach and retraction, straight line only)

41...G41

42...G42

Rectangular spigot milling – CYCLE76

Programming

CYCLE76 (_RTP, _RFP, _SDIS, _DP, _DPR, _LENG, _WID, _CRAD, _PA, _PO, _STA, _MID, _FAL, _FALD, _FFP1, _FFD, _CDIR, _VARI, _AP1, _AP2)

Parameters

_RTP real Retraction plane (absolute)

_RFP real Reference plane (absolute)

_SDIS real Safety clearance (enter without sign)

_DP real Final drilling depth (absolute)

_DPR real Final drilling depth relative to the reference plane (enter without sign)

_LENG real Spigot length (enter without sign)

_WID real Spigot length (enter without sign)

_CARD real Spigot corner radius (enter without sign)

_PA real Reference point of spigot, abscissa (absolute)

_PO real Reference point of spigot, ordinate (absolute)

_STA real Angle between longitudinal axis and 1st axis of plane

_MID real Maximum depth infeed (incremental; enter without sign)

_FAL real Final machining allowance at the margin contour (incremental)

_FALD real Finishing allowance at the base (incremental, enter without sign)

_FFP1 real Feedrate at the contour

_FFD real Feedrate for depth infeed

_CDIR integer Milling direction (enter without sign)

Values: 0 Synchronous milling

1 Conventional milling

2 With G2 (independent of spindle direction)

3 With G3

_VARI integer Machining type

Values: 1 Roughing up to finishing allowance

2 Finishing (allowance X/Y/Z=0)

_AP1 real Length of blank spigot

Function

Use this cycle to machine rectangular spigots in the machining plane. For finishing, a face cutter is required. The depth infeed is always carried out in the position upstream of the semicircle style approach to the contour.

_PA, _PO (reference point)

Use the parameters _PA and _PO to define the reference point of the spigot along the abscissa and the ordinate.

This is the spigot center point.

_STA (angle)

_STA specifies the angle between the 1st axis of the plane (abscissa) and the longitudinal axis of the spigot.

_CDIR (milling direction)

Use this parameter to specify the machining direction for the spigot.

By using the parameter _CDIR, the milling direction

_ can be programmed directly with ”2 for G2” and ”3 for G3” or,

_ alternatively, ”Synchronous milling” or ”Conventional milling”.

can be programmed for the transformation declared. synchronized operation or reverse rotation are determined internally in the cycle via the direction of rotation of the spindle activated prior to calling the cycle.

Synchronous milling Conventional milling

M3 →G3 M3 →G2

M4 →G2 M4 →G3

_VARI (machining type)

Use the parameter _VARI to define the machining type.

Possible values are:

_ 1 = roughing

_ 2 = finishing

_AP1, _AP2 (blank dimensions)

When machining the spigot, it is possible to take into account blank dimensions (e.g. when machining precast parts).

The blank dimensions for length and width (_AP1 and _AP2) are programmed without sign and are placed by the cycle symmetrically around the pocket center point via calculation.

The internally calculated radius of the approach semicircle depends on this dimension.

Circular spigot milling – CYCLE77

Programming

CYCLE77 (_RTP, _RFP, _SDIS, _DP, _DPR, _PRAD, _PA, _PO, _MID, _FAL, _FALD, _FFP1, _FFD, _CDIR, _VARI, _AP1)

Parameters

The following input parameters are always required:

Table 9-18 Parameters for CYCLE77

_RTP real Retraction plane (absolute)

_RFP real Reference plane (absolute)

_SDIS real Safety clearance (enter without sign)

_DP real Depth (absolute)

_DPR real Depth relative to the reference plane (enter without sign)

_PRAD real Spigot diameter (enter without sign)

_PA real Center point of spigot, abscissa (absolute)

_PO real Center point of spigot, ordinate (absolute)

_MID real Maximum depth infeed (incremental; enter without sign)

_FAL real Final machining allowance at the margin contour (incremental)

_FALD real Finishing allowance at the base (incremental, enter without sign)

_FFP1 real Feedrate at the contour

_FFD real Feedrate for depth infeed (or spatial infeed)

_CDIR integer Milling direction (enter without sign)

Values: 0 Synchronous milling

1 Conventional milling

2 With G2 (independent of spindle direction)

3 With G3

_VARI integer Machining type

Values: 1 Roughing up to finishing allowance

2 Finishing (allowance X/Y/Z=0)

_AP1 real Length of blank spigot

Function

Use this cycle to machine circular spigots in the machining plane. For finishing, a face cutter is required. The depth infeed is always carried out in the position upstream of the semicircle style approach to the contour.

Figure 9-48

Slots on a circle – LONGHOLE

Programming

LONGHOLE (RTP, RFP, SDIS, DP, DPR, NUM, LENG, CPA, CPO, RAD, STA1, INDA, FFD, FFP1, MID)

Parameters

RTP real Retraction plane (absolute)

RFP real Reference plane (absolute)

SDIS real Safety clearance (enter without sign)

DP real Slot depth (absolute)

DPR real Slot depth relative to the reference plane (enter without sign)

NUM integer Number of slots

LENG real Slot length (enter without sign)

CPA real Center point of circle of holes (absolute), 1st axis of the plane

CPO real Center point of circle of holes (absolute), 2nd axis of the plane

RAD real Radius of the circle (enter without sign)

STA1 real Starting angle

INDA real Incrementing angle

FFD real Feedrate for depth infeed

FFP1 real Feedrate for surface machining

MID real Maximum infeed depth for one infeed (enter without sign)

Function

Use this cycle to machine elongated holes arranged on a circle. The longitudinal axis of the slots is aligned radially.

Contrary to the slot, the width of the long hole is determined by the tool diameter.

Internally in the cycle, an optimum traversing path of the tool is determined, ruling out unnecessary idle passes. If several depth infeeds are required to machine an slot, the infeed is carried out alternately at the end points. The path to be traversed along the longitudinal axis of the slot will change its direction after each infeed. The cycle will search for the shortest path when changing to the next slot.

Slots on a circle – SLOT1

Programming

SLOT1(RTP, RFP, SDIS, DP, DPR, NUM, LENG, WID, CPA, CPO, RAD, STA1, INDA, FFD, FFP1, MID, CDIR, FAL, VARI, MIDF, FFP2, SSF)

Parameters

RTP real Retraction plane (absolute)

RFP real Reference plane (absolute)

SDIS real Safety clearance (enter without sign)

DP real Slot depth (absolute)

DPR real Slot depth relative to the reference plane (enter without sign)

NUM integer Number of slots

LENG real Slot length (enter without sign)

WID real Slot width (enter without sign)

CPA real Center point of circle of holes (absolute), 1st axis of the plane

CPO real Center point of circle of holes (absolute), 2nd axis of the plane

RAD real Radius of the circle (enter without sign)

STA1 real Starting angle

INDA real Incrementing angle

FFD real Feedrate for depth infeed

FFP1 real Feedrate for surface machining

MID real Maximum infeed depth for one infeed (enter without sign)

CDIR integer Mill direction for machining the slot

Values: 2 (for G2)

3 (for G3)

FAL real Finishing allowance at the slot edge (enter without sign)

VARI integer Machining type

Values: 0=complete machining

1=roughing

2=finishing

MIDF real Maximum infeed depth for finishing

FFP2 real Feedrate for finishing

SSF real Speed when finishing

Note

The cycle requires a milling cutter with an ”end tooth cutting across center” (DIN844).

Function

The cycle SLOT1 is a combined roughing-finishing cycle.

Use this cycle to machine slots arranged on a circle. The longitudinal axis of the slots is aligned radially. Unlike the slot, a value is defined for the slot width.

Function

The cycle SLOT1 is a combined roughing-finishing cycle.

Use this cycle to machine slots arranged on a circle. The longitudinal axis of the slots is aligned radially. Unlike the slot, a value is defined for the slot width.

Circumferential slot – SLOT2

Programming

SLOT2(RTP, RFP, SDIS, DP, DPR, NUM, AFSL, WID, CPA, CPO, RAD, STA1, INDA, FFD, FFP1, MID, CDIR, FAL, VARI, MIDF, FFP2, SSF)

Parameters

RTP real Retraction plane (absolute)

RFP real Reference plane (absolute)

SDIS real Safety clearance (enter without sign)

DP real Slot depth (absolute)

DPR real Slot depth relative to the reference plane (enter without sign)

NUM integer Number of slots

AFSL real Angle for the slot length (enter without sign)

WID real Circumferential slot width (enter without sign)

CPA real Center point of circle of holes (absolute), 1st axis of the plane

CPO real Center point of circle of holes (absolute), 2nd axis of the plane

RAD real Radius of the circle (enter without sign)

STA1 real Starting angle

INDA real Incrementing angle

FFD real Feedrate for depth infeed

FFP1 real Feedrate for surface machining

MID real Maximum infeed depth for one infeed (enter without sign)

CDIR integer Mill direction for machining the circumferential slot

Values: 2 (for G2)

3 (for G3)

FAL real Finishing allowance at the slot edge (enter without sign)

VARI integer Machining type

Values: 0 = complete machining

1 = roughing

2 = finishing

MIDF real Maximum infeed depth for finishing

Milling a rectangular pocket – POCKET3

Programming

POCKET3(_RTP, _RFP, _SDIS, _DP, _LENG, _WID, _CRAD, _PA, _PO, _STA, _MID, _FAL, _FALD, _FFP1, _FFD, _CDIR, _VARI, _MIDA, _AP1, _AP2, _AD, _RAD1, _DP1)

Parameters

_RTP real Retraction plane (absolute)

_RFP real Reference plane (absolute)

_SDIS real Safety clearance (enter without sign)

_DP real Pocket depth (absolute)

_LENG real Pocket length, for dimensioning from the corner with sign

_WID real Pocket width, for dimensioning from the corner with sign

_CRAD real Pocket corner radius (enter without sign)

_PA real Reference point for the pocket (absolute), 1st axis of the plane

_PO real Reference point for the pocket (absolute), 2nd axis of the plane

_STA real Angle between the pocket longitudinal axis and the first axis of the plane (enter without sign);

Value range: 0 ≤_STA 180

_MID real Maximum infeed depth (enter without sign)

_FAL real Finishing allowance at the pocket edge (enter without sign)

_FALD real Finishing allowance at the base (enter without sign)

_FFP1 real Feedrate for surface machining

_FFD real Feedrate for depth infeed

_CDIR integer Milling direction: (enter without sign)

Values: 0 Synchronous milling (according to the spindle direction)

1 Conventional milling

2 With G2 (independent of spindle direction)

3 With G3

_VARI integer Machining type

UNITS DIGIT

Values: 1 Roughing

2 Finishing

TENS DIGIT:

Values: 0 Perpendicular to the pocket center with G0

1 Perpendicular to the pocket center with G1

2 Along a helix

3 Perpediculation along a pocket longitudinal axis

The other parameters can be selected as options. Specify the plunge-cut strategy and the overlap for solid machining (to be entered without sign):

Function

The cycle can be used for roughing and finishing. For finishing, a face cutter is required.

The depth infeed will always start at the pocket center point and be performed vertically from there; thus it is practical to predrill at this position.

_ The milling direction can be determined either using a G command (G2/G3) or from the spindle direction as synchronous or opposed milling.

_ For solid machining, the maximum infeed width in the plane can be programmed.

_ Finishing allowance also for the pocket base

_ There are three different insertion strategies:

– vertically to the pocket center

– along a helical path around the pocket center

– oscillating at the pocket central axis

_ Shorter approach paths in the plane for finishing

_ Consideration of a blank contour in the plane and a blank dimension at the base (optimum machining of preformed pockets possible).

Milling a circular pocket – POCKET4

Programming

POCKET4 (_RTP, _RFP, _SDIS, _DP, _PRAD, _PA, _PO, _MID, _FAL, _FALD, _FFP1, _FFD, _CDIR, _VARI, _MIDA, _AP1, _AD, _RAD1, _DP1)

Parameters

_RTP real Retraction plane (absolute)

_RFP real Reference plane (absolute)

_SDIS real Safety clearance (to be added to the reference plane; enter without sign)

_DP real Pocket depth (absolute)

_PRAD real Pocket radius

_PA real Starting point (absolute), 1st axis of the plane

_PO real Starting point (absolute), 2nd axis of the plane

_MID real Maximum infeed depth (enter without sign)

_FAL real Finishing allowance at the pocket edge (enter without sign)

_FALD real Finishing allowance at the base (enter without sign)

_FFP1 real Feedrate for surface machining

_FFD real Feedrate for depth infeed

_CDIR integer Milling direction: (enter without sign)

Values: 0 Synchronous milling (according to the spindle direction)

1 Conventional milling

2 With G2 (independent of spindle direction)

3 With G3

_VARI integer Machining type

UNITS DIGIT

Values: 1 Roughing

2 Finishing

TENS DIGIT:

Values: 0 Perpendicular to the pocket center with G0

1 Perpendicular to the pocket center with G1

2 Along a helix

The other parameters can be selected as options. Specify the plunge-cut strategy

and the

overlap for solid machining (to be entered without sign):

_MIDA real Maximum infeed width as a value in solid machining in the plane

_AP1 real Pocket radius blank dimension

_AD real Blank pocket depth dimension from reference plane

_RAD1 real Radius of the helical path during insertion (referred to the tool

center point path)

_DP1 real Insertion depth per 360 . revolution on insertion along helical path

Function

Use this cycle to machine circular pockets in the machining plane. For finishing, a face cutter is required.

The depth infeed will always start at the pocket center point and be performed vertically from there; thus it is practical to predrill at this position.

_ The milling direction can be determined either using a G command (G2/G3) or from the spindle direction as synchronous or opposed milling.

_ For solid machining, the maximum infeed width in the plane can be programmed.

_ Finishing allowance also for the pocket base

_ Two different insertion strategies:

– vertically to the pocket center

– along a helical path around the pocket center

_ Shorter approach paths in the plane for finishing

_ Consideration of a blank contour in the plane and a blank dimension at the base (optimum machining of preformed pockets possible).

_ _MIDA is recalculated during edge machining.

Thread milling – CYCLE90

Programming

CYCLE90 (RTP, RFP, SDIS, DP, DPR, DIATH, KDIAM, PIT, FFR, CDIR, TYPTH, CPA, CPO)

Parameters

RTP real Retraction plane (absolute)

RFP real Reference plane (absolute)

SDIS real Safety clearance (enter without sign)

DP real Final drilling depth (absolute)

DPR real Final drilling depth relative to the reference plane (enter without sign)

DIATH real Nominal diameter, outer diameter of the thread

KDIAM real Core diameter, internal diameter of the thread

Function

By using the cycle CYCLE90, you can produce internal or external threads. The path when milling threads is based on a helix interpolation. All three geometry axes of the current plane, which you will define before calling the cycle, are involved in this motion.

Sequence when producing an external thread

Position reached prior to cycle start:

The starting position is any position from which the starting position at the outside diameter of the thread at the height of the retraction plane can be reached without collision.

This start position for thread milling with G2 lies between the positive abscissa and the positive ordinate in the current level (i.e., in the 1st quadrant of the coordinate system). For thread milling with G3, the start position lies between the positive abscissa and the negative ordinate (i.e., in the 4th quadrant of the coordinate system).

10.4 Arithmetic Parameters R

Functionality

The arithmetic parameters are used if an NC program is not only to be valid for values assigned once, or if you must calculate values. The required values can be set or calculated by the control system during program execution.

The arithmetic parameter values can also be set by operator inputs. If values have been assigned to the arithmetic parameters, they can be assigned to other variable-setting NC addresses in the program.

Programming

R0 = ... bis R299 = ... ; Assign values to the R parameters

R[R0] = ... ; Indirect programming: Assign a value to the R parameter whose number can be found, e.g. in R0

X = R0 ; Assign arithmetic parameters to the NC addresses, e.g. for the X axis

Value assignment

You can assign values in the following range to the R parameters:

_(0.000 0001 ... 9999 9999)

(8 decimal places, arithmetic sign and decimal point)

The decimal point can be omitted for integer values. A plus sign can always be omitted.

Example:

R0 = 3.5678 R1 = –37.3 R2 = 2 R3 = –7 R4 = –45678.123

Use the **exponential notation** to assign an extended range of numbers:

_ (10–300 ... 10+300).

The value of the exponent is written after the **EX** characters; maximum total number of characters: 10 (including leading signs and decimal point)

Range of values for EX: –300 to +300

Example:

R0 = –0.1EX–5 ; Meaning: R0 = –0.000 001

R1 = 1.874EX8 ; Meaning: R1 = 187 400 000

Remark: There can be several assignments in one block incl. assignments of arithmetic expressions.

10.5 Local User Data

Functionality

The operator/programmer (user) can define his/her own variable in the program from various data types (LUD = Local User Data). These variables are only available in the program in which they were defined. The definition takes place immediately at the start of the program and can also be associated with a value assignment at the same time. Otherwise the starting value is zero.

The name of a variable can be defined by the programmer. The naming is subject to the following rules:

_ A maximum of 32 characters can be used.

_ It is imperative to use letters for the first two characters; the remaining characters can be either letters, underscore or digits.

_ Do not use a name already used in the control system (NC addresses, keywords, names of programs, subroutines, etc.).

Programming / data types

DEF BOOL varname1 ; ”Bool” type, values: TRUE (= 1), FALSE (= 0)

DEF CHAR varname2 ; ”Char” type, 1 character in the ASCII code: ”a”, ”b”, ...

; Numerical code value: 0 ... 255

DEF INT varname3 ; Integer type, integer values, 32–bit value range:

; –2 147 483 648 ... +2 147 483 648 (decimal)

DEF REAL varname4 ; ”Real” type, natural number (as witrh R parameter),

; Value range: _(0.000 0001 ... 9999 9999)

; (8 decimal places, arithmetic sign and decimal point) or

; exponential notation: _ (10–300 ... 10+300)

DEF STRING[*string length*] varname41 ; STRING type, [*string length*]: Maximum number of characters

Each data type requires its own program line. However, several variables of the same type can be defined in one line.

Example:

DEF INT PVAR1, PVAR2, PVAR3 = 12, PVAR4 ; 4 variables of the INT type

Example for STRING type with assignment:

DEF STRING[12] PVAR = ”Hello” ; Define PVAR variable with maximum

string length 12 and character

sequence

Hello

Fields

In addition to the individual variables, one or two-dimensional fields of variables of these data types can also be defined:

DEF INT PVAR5[n] ; Single-dimensionsal field of INT type, n: integer

DEF INT PVAR6[n,m] ; Two-dimensional field of the INT type, n, m: integer

Example:

DEF INT PVAR7[3] ; Field with 3 elements of the INT type

Within the program, the individual field elements can be reached via the field index and can be treated like individual variables. The field index runs from 0 to a small number of the elements.

Example:

N10 PVAR7[2] = 24 ; The third field element (with index 2) is assigned the value 24.

Value assignment for field with SET instruction:

N20 PVAR5[2] = SET(1,2,3) ; Starting with the 3rd field element, different values are assigned.

Value assignment for field with REP instruction:

N20 PVAR7[4] = REP(2) ; Starting from the field element [4], all values are assigned the same value, here 2.

Jump destination for program jumps

Functionality

A **label** or a **block number** serve to mark blocks as jump destinations for program jumps. Program jumps can be used to branch to the program sequence.

Labels can be freely selected, but must contain a minimum of 2 and a maximum of 8 letters or numbers, and the **first two characters** must be **letters** or underscores.

Labels that are in the block that serves as the jump destination are **ended by a colon**.

They are always at the start of a block. If a block number is also present, the label is located **after the block number**.

Labels must be unique within a program.

Programming example

N10 LABEL1: G1 X20 ; LABEL1 is the label, jump destination

...

TR789: G0 X10 Z20 ; TR789 is the label, jump destination

– No block number existing

N100 ... ; A block number can be a jump destination.

CHAPTER 11 SINUMERIK 802Se programme

11.1 Position

Absolute/incremental dimensions:G90/G91

1. **Functionality**

When instruction G90 or G91 is active, the specified position information X, Z is interpreted as a coordinate point (G90) or as an axis path to be traversed (G91). G90/G91 applies to all axes. These instructions do not determine the actual path on which the end points are reached. This is done by a G group

2. Programming

G90 absolute **dimension**

G91 **Incremental dimension**

X=AC（…）X axis programming in according to absolute **dimension**

X=IC（…）X axis programming in according to **Incremental dimension**

Absolute dimension G90:

When absolute dimensioning is selected, the dimension data refer to the zero point of the currently active coordinate system (workpiece coordinate system, current workpiece coordinate system or machine coordinate system). Which of the systems is active depends on which offsets are currently effective, i.e. programmable, settable or none at all.

G90 is active for all axes on program start and remains so until it is deactivated by G91 (incremental dimensioning selection) in a subsequent block (modal command).

Incremental dimension G91:

When incremental dimensioning is selected, the numerical value in the poison information corresponds to the path to be traversed by an axis. The traversing direction is determined by the sign.

G91 applies to all axes and can be deactivated by G90 (absolute dimensioning) in a later block.

3. example for G90 and G91 programming

```
N10 G90 X20 Z90          ;Absolute dimensioning
N20 X75 Z-32             ;Absolute dimensioning still active
…
N180 G91 X40 Z20         ; Switchover to incremental dimensioning
N190 X-12 Z17            ;Incremental dimensioning still active
```

Radius/diameter dimensions: G22/G23

1. **Functionality**

When parts are machined on turning machines, it is normal practice to program the position data for the X axis (facing axis) as a diameter dimension.

The specified value is interpreted as a diameter for this axis only by the control. It is possible to switch over to radius dimension in the program if necessary.

2. **Programming**

G22 Radius dimension G23 Diameter dimension

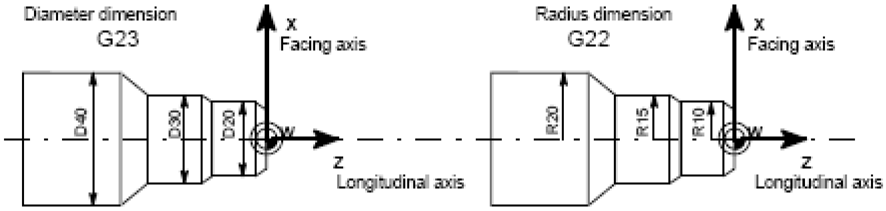


Fig 11.1-1

Information

When G22 or G23 is active, the specified end point for the X axis is interpreted as a radius or diameter dimension.

The actual value is displayed correspondingly in the workpiece coordinate system. A programmable offset with G158 X... is always interpreted as a radius dimension. See the following section for a description of this function.

3. **Programming example**

N10 G23 X44 Z30 ;Diameter for X axis

N20 X48 Z25 ;G23 still active

N30 Z10

...

N110 G22 X22 Z30 ;Changeover to radius dimension for X axis from here

N120 X24 Z25

N130 Z10

...

Programmable zero offset:G158

1. **Functionality**

Use the programmable zero offset for frequently repeated shapes/arrangements in different positions on a workpiece or when you simply wish to choose a new reference point for the dimension data. The programmable offset produces the current workpiece coordinate system. The newly programmed dimension data then refer to this system. The offset can be applied in all axes. A separate block is always required for the G158 instruction.

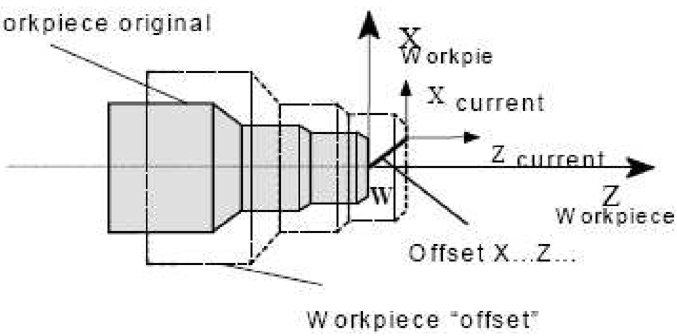


Fig 11.1-2

2. **Offset G158**

A zero offset can be programmed for all axes with instruction G158. A newly entered G158 instruction replaces any previous programmable offset instruction.

3. **Delete offset**

If the instruction G158 without axes is inserted in a block , then any active programmable offset will be deleted.

4. **Programming Example**

N10 ...

N20 G158 X3 Z5 ;Programmable offset

N30 L10 ;Subroutine call, contains the geometry to be offset

...

N70 G158 ;Offset deleted

...

Workpiece clamping - settable zero offset: G54 to G57, G500, G53

1. **Functionality**

The settable zero offset specifies the position of the workpiece zero point on the machine (offset between workpiece zero and machine zero). This offset is calculated when the workpiece is clamped on the machine and must be entered by the operator in the data field provided. The value is activated by the program through selection from four possible groups: G54 to G57.

2. **Programming**

G54 ;1st settable zero offset

G55 ;2nd settable zero offset

G56 ;3rd settable zero offset

G57 ;4th settable zero offset

G500 ;Settable zero offset OFF modal

G53 ;Settable zero offset OFF non-modal, also suppresses programmable offset

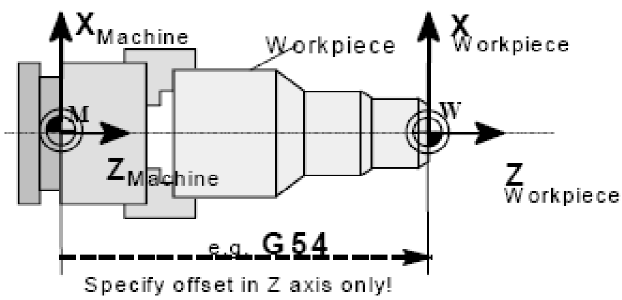


Fig 11.1-3

3. **Programming Example**

N10 G54 ... ;Call first settable zero offset

N20 X... Z... ;Machine workpiece

...

N90 G500 G0 X... ;Deactivate settable zero offset

11.2 G Commands

11.2.1 Linear interpolation at rapid traverse:

Functionality

The rapid traverse motion G0 is used to position the workpiece rapidly, but not to machine the workpiece directly. All axes can be traversed simultaneously resulting in a linear path.

The maximum speed (rapid traverse) for each axis is set in the machine data. If only one axis is moving, it traverses at its own rapid traverse setting. If two axes are traversed simultaneously, then the path speed (resultant speed) is selected so as to obtain the maximum possible path speed based on the settings for both axes.

A programmed feed (F word) is irrelevant for G0. G0 remains effective until it is canceled by another instruction from the same group (G1, G2, G3,...).

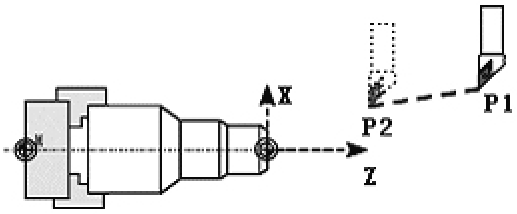


Fig 11.2-1

Programming example

N10 G0 X100 Y150 Z65 ; Cartesian coordinate

N50 G0 RP=16.78 AP=45 ; Polar coordinate

Information

Another group of G functions exists for movement to the position .For G60 exact stop, a window with various precision values can be selected with another G group. For exact stop, an alternative instruction with non-modal effectiveness exists: G9.

You should consider these options for adaptation to your positioning tasks.

11.2.2 Positional data

G01 Linear interpolation

Functionality

The tool moves from the start point to the end point along a straight path. The path speed is defined by the programmed F word. All axes can be traversed simultaneously.

G1 remains effective until it is canceled by another instruction from the same G group (G0, G2, G3, ...).

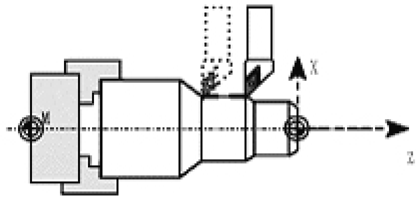


Fig 11.2-2

Programming example

N05 G54 G0 G90 X40 Z200 S500 M3 ;tool is moving at rapid traverse, spindle speed = 500 rpm, CW rotation

N10 G1 Z120 F0.15 ;Linear interpolation with feed 0.15 mm/rev

N15 X45 Z105

N20 Z80

N25 G0 X100 ;Traverse clear at rapid traverse

N30 M2 ;End of program

G02/G03 Circular interpolation

1.**Functionality**

The tool moves from the start point to the end point on a circular path. The direction is determined by the G function:

G2 - in clockwise direction

G3 - in counterclockwise direction

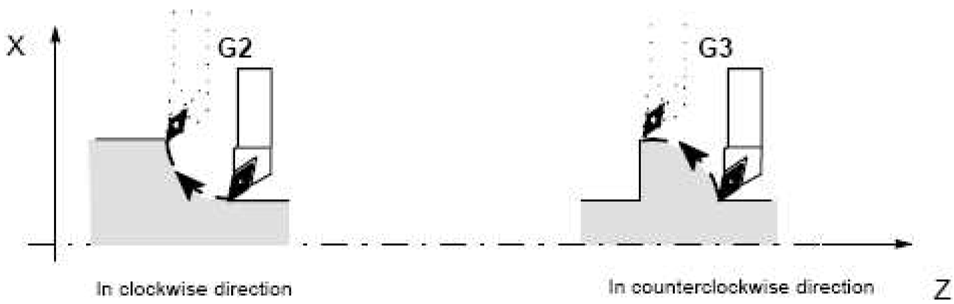


Fig 11.2-3

G2/G3 remain effective until they are canceled by another instruction from the same G group (G0, G1, ...).

Note： The required cycle can be described in different ways:

_ Center point and end point

_ Circle radius and end point

_ Center point and aperture angle

2. **Programming**

G2/G3 X... Y... I... J... ; Center and end points

G2/G3 CR=... X... Y... ; Circle radius and end point

G2/G3 AR=... I... J... ; Aperture angle and center point

G2/G3 AR=... X... Y... ; Aperture angle and end point

G2/G3 AP=... RP =... ; Polar coordinates, circle around the pole

Further possibilities for circle programming result from:

CT – circle with tangential connection and

CIP – circle via intermediate point (see next sections).

3. **Programming example**

Center point and end point specification:

N5 G90 Z30 X40 ;Circle start point for N10

N10 G2 Z50 X40 K10 I-7 ;End point and center point

End point and radius specification

N5 G90 X30 Y40 ; Circle starting point for N10

N10 G2 X50 Y40 CR=12.207 ; End point and radius

Note: With a negative leading sign for the value with CR=–..., a circular segment larger than a semi-circle is selected.

End point and aperture angle:

N5 G90 Z30 X40 ;Circle start point for N10

N10 G2 Z50 X40 AR=105 ;End point and aperture angle

Center point and aperture angle:

N5 G90 Z30 X40 ;Circle start point for N10

N10 G2 K10 I-7 AR=105 ;Center point and aperture angle

G05 Circular interpolation via intermediate point

1. **Functionality**

If you know three contour points around the circle instead of center point or radius or aperture angle, you should preferably use the G5 function.

The direction of the circle in this case is determined by the position of the intermediate point (between start and end positions).

G5 remains effective until it is canceled by another instruction from the same G group (G0, G1, G2, ...).

Note: The dimension setting G90 or G91 applies to both the end point **and** intermediate point!

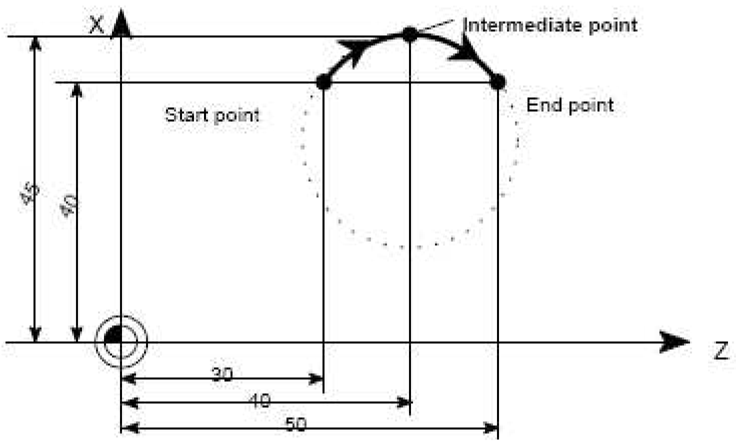


Fig 11.2-4

2. **Programming example**

N5 G90 Z30 X40 ;Circle start point for N10

N10 G5 Z50 X40 KZ=40 IX=45 ;End and intermediate points (XI must be programmed as a radius dimension)

G33 Thread cutting with constant lead:

1. **Functionality**

Function G33 can be used to cut the following types of threads with constant lead:

- z Thread on cylindrical bodies
- z Thread on tapered bodies
- z External/internal threads
- z Single-start/multiple-start threads
- z Multi-block threads (thread "chaining")

G group (G0, G1, G2,G3,...).

2. **Prerequisite** This requires a spindle with position measuring system

G33 remains effective until it is canceled by another instruction from the same

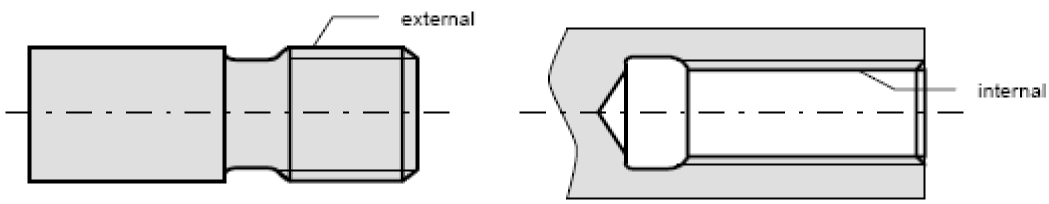


Fig 11.2-5

3. **Right-hand or left-hand threads**

The direction of the thread, i.e. right-hand or left-hand, is determined by the setting for the direction of rotation of the spindle (M3 - clockwise rotation, M4 - counterclockwise rotation). To this aim,the speed setting must be programmed under address S, or a speed must be set.

Note: The approach and run-out paths must be taken into account with respect to the thread length.

In the case of tapered threads (2 axes must be specified), the lead address I or K of the axis with the longer path (greater thread length) must be used. A second lead is not specified.

4. **Start-point offset** SF

A start-point offset of the spindle is required for machining multiple-start threads or threads in offset cuts. The start-point offset is programmed under address SF in the thread block with G33 (absolute position).

If a start point is not included in the block, the value from the setting data is activated.

Note: Any value programmed for SF= is always entered in the setting data as well.

5. **Programming example**

Cylindrical thread, two-start, start-point offset 180 degrees, thread length (including approach and run-out) 100 mm, thread lead 4 mm/rev.

RH thread, cylinder premachined:

```
N10 G54 G0 G90 X50 Z0 S500 M3     ;Approach start point, CW spindle rotation
N20 G33 Z-100 K4 SF=0 ;Lead          :4 mm/rev.
N30 G0 X54
N40 Z0
N50 X50
N60 G33 Z-100 K4 SF=180              ;2nd start, 180 degrees offset
N70 G0 X54 ...
```

G75 Fixed point approach

1. **Functionality**

By using G75, a fixed point on the machine, e.g. tool change point, can be approached. The position is stored permanently in the machine data for all axes. No offset is effective. The speed of each axis is its rapid traverse.

G75 requires a separate block and is non-modal. The machine axis identifier must be programmed!

In the block after G75, the previous G command of the ”Interpolation type” group (G0, G1,G2, ...) is active again.

2. **Programming example**

```
N10 G75 X0 Z0
```

Remark: The programmed position values for X, Z (any value, here = 0) are ignored, but must still be written.

G74 Reference point approach

1. **Functionality**

The reference point can be approached in the NC program with G74. The direction and speed of each axis are stored in machine data.

G74 requires a separate block and is non-modal. The machine axis identifier must be programmed!

In the block after G74, the previous G command of the ”Interpolation type” group (G0, G1,G2, ...)

is active again.

2. **Programming example**

N10 G74 X0 Z0

Remark: The programmed position values for X, Z (any value, here = 0) are ignored, but must still be written.

G9/G60/G64 Exact stop / continuous-path control mode

1. **Functionality**

G functions are provided for optimum adaptation to different requirements to set the traversing behavior at the block borders and for block advancing. Example: For example, you would like to quickly position with the axes or you would like to machine path contours over multiple blocks.

2. **Programming**

G60 ;Exact stop – modal

G64 ;Continuous-path control mode

G9 ; Exact stop – non-modal

G601 ; Exact stop window fine

G602 ; Exact stop window coarse

3. **exact stop fine G60,G9**

If the exact stop function (G60 or G9) is active, the velocity for reaching the exact end position at the end of a block is decelerated to zero.

Another modal G group can be used here to set when the traversing movement of this block is considered ended and the next block is started.

* G601 ; Exact stop window fine

Block advance takes place when all axes have reached the ”Exact stop window fine” (value in the machine data).

* G602 ; Exact stop window coarse

Block advance takes place when all axes have reached the ”Exact stop window coarse” (value in the machine data).

e selection of the exact stop window has a significant influence on the total time if many positioning operations are executed. Fine adjustments require more time.

4. **Programming example**

```
N5 G602 ; Exact stop window coarse
N10 G0 G60 X...    ; Exact stop modal
N20 X... Y...      ; G60 remains active
...
N50 G1 G601 .. . ; Exact stop window fine
N80 G64 X..      . ; Switching to continuous-path control mode
...
```

N100 G0 G9 X.. . ; Exact stop is only effective for this block

N111 .. ; Continuous-path control mode again

Remark: The G9 command only generates exact stop for the block in which it is programmed; G60, however, is effective until it is canceled by G64.

5. **Continuous-path control mode G64**

The objective of the continuous-path control mode is to avoid deceleration at the block boundaries and to switch **to the next block** with **a path velocity as constant as possible** (in the case of tangential transitions). The function works with **look-ahead velocity control** over several blocks. For non-tangential transitions (corners), the velocity can reduced rapidly enough so that the axes are subject to a relatively high velocity change over a short time. This may lead to a significant jerk (acceleration change). The size of the jerk can be limited by activating the SOFT function.

6. **Programming example**

N10 G64 G1 X... F... ; Continuous-path control mode

N20 Y. . ; Continuous-path control mode continues to be active

...

N180 G60 ... ; switching to exact stop

G4 Dwell **Time**

1. **Functionality**

Between two NC blocks, you can interrupt the machining for a defined time by inserting a **separate block** with G4,The words with F... or S... are only used in this block for the specified time. Any previously programmed feedrate F or a spindle speed S remain valid.

2. Programming

G4 F... ; Dwell time in s

G4 S... ; Dwell time in spindle revolutions

3. **Programming example**

N5 G1 F200 Z-50 S300 M3 ; Feedrate F, spindle speed S

N10 G4 F2.5 ; Dwell time 2.5 s

N20 Z70

N30 G4 S30 ; Dwell for 30 spindle revolutions; corresponds to S=300 r.p.m., and100 % speed override: t=0.1 min

N40 X... ; Feedrate and spindle speed continue to be effective

Remark

G4 S.. is only possible if a controlled spindle is available (if the speed specifications are also programmed via S...).

F Feedrate

1. **Functionality**

The feed F is the **path velocity** and represents the value of the geometric sum of the velocity components of all axes involved. The individual axis velocities therefore result from the portion of the axis path in the overall distance to be traversed.

The feedrate F is effective for the interpolation types G1, G2, G3, and G5 and is retained until a new F word is written.

2. Programming

F...

Remark**:**

For **integer values**, the decimal point is not required, e.g. F300.

Unit of measure for F with G94, G95

The dimension unit for the F word is determined by G functions:

_ G94 **F** as the feedrate in**mm/min**

_ G95 **F** as the feedrate in **mm/rev.** of the spindle (only meaningful when the spindle is running)

Remark**:**

This unit of measure applies to metric dimensions. According to Section ”Metric and inch dimensioning”, settings with inch dimensioning are also possible.

3. **Programming example**

N10 G94 F310 ; Feedrate in mm/min

...

N110 S200 M3 ; Spindle rotation

N120 G95 F15.5 ; Feedrate in mm/rev.

Remark: Write a new F word if you change G94 – G95.

S spindle speed/direction of rotation

1. **Functionality**

The spindle speed is programmed in r.p.m. under the address S provided that the machine possesses a controlled spindle.

The direction of rotation and the start or end of the movement are specified via M commands (also see Section 8.7 ”Miscellaneous function M”).

M3 ; Spindle CW rotation

M4 ; Spindle CCW rotation

M5 ; Spindle stop

Remark: For integer S values, the decimal point can be omitted, e.g. S270

Information If you write M3 or M4 in a **block with axis movements**, the M commands become active **before** the axis movements.

2. **Programming example**

N10 G1 X70 Z20 F300 S270 M3 ; Spindle accelerates CW to 270 r.p.m.**before** traversing of the X, Z axes...

N80 S450 .. . ; Speed change ...

N170 G0 Z180 M5 ; Z motion, spindle stops

G25/G26 main spindle speed limitation

1. **Functionality**

In the program, you can limit the limit values that would otherwise apply for a controlled spindle by writing G25 or G26 and the spindle address S with the speed limit value.This overwrites the values entered in the setting data at the same time.

G25 and G26 each require a separate block. A previously programmed speed S is maintained.

2. Programming

G25 S…limits the main spindle lower speed value

G26 S… limits the main spindle upper speed value.

1. **Information：** The outmost limits of the spindle speed are set in machine data. Appropriate inputs via the operator panel can activate various setting data for further limiting.

2. **Programming example**

N10 G25 S12 ; Lower spindle limit speed: 12 r.p.m.

N20 G26 S700 ; Upper spindle limit speed : 700 r.p.m.

SPOS Spindle positioning

1. **Functionality**

Prerequisite: The spindle must be technically designed for position control. With the function SPOS = you can position the spindle in a specific **angular position**. The spindle is held in the position by position control.

The **speed** of the positioning procedure is defined in machine data. With SPOS = **value** from the M3/M4 movement, the respective **direction of rotation** is maintained until the end of the positioning. When positioning from standstill, the position is approached via the shortest path. The direction results from the respective starting and end position.

Exception: First movement of the spindle, i.e. if the measuring system is not yet synchronized. In this case, the direction is specified in machine data.

Other movement specifications for the spindle are possible with SPOS = ACP (...), SPOS = ACN (...), ... as for rotary axes (see Section ”4th axis”). The spindle movement takes place parallel to any other axis movements in the same block. This block is ended when both movements are finished.

2. **Programming**

SPOS = ... ; Absolute position: 0 ... <360 degrees

3. **Programming example**

N10 SPOS = 14.3 ; Spindle position 14.3 degrees

...

N80 G0 X89 Z300 SPOS = 25.6 ; Positioning of the spindle with axis movements ; The block is only completed if all movements are performed.

N81 X200 Z300 ; The N81 block will only start if the spindle position from N80; is reached.

T Tool

1. **Functionality**

You select a tool by programming the T word. A machine data defines whether the T word represents a tool change or merely a preselection.

- z Tool change (tool call) is implemented directly by T word (e.g. normal practice for tool revolver on turning machines) or
- z the tool is changed through additional instruction M6 after preselection by T word (see also Section “Miscellaneous Functions M”).

Please note:

If a certain tool has been activated, this will remain stored as the active tool even across the program end and after POWER ON of the control system. If you change a tool manually, then enter the change into the control system also manually to make sure that the control system detects the right tool. For example, you can start a block with a new T word in the MDA mode.

2. **Programming**

T... ;Tool number: 1 ... 32 000

Note A maximum of 15 tools can be stored in the control at a time.

3. **Programming example**

; Tool change without M6**:**

N10 T1 ; Tool 1

...

N70 T588 ; Tool 588

; Tool change with M6:

N10 T14 ... ; Preselect tool 14

...

N15 M6 ; Perform tool change; thereafter, T14 is active

D Tool offset number

1. **Functionality**

You can assign between 1 and 9 data fields with various tool offset blocks (for several tool edges) to each specific tool. If a special edge is required, it can be programmed by means of D plus a corresponding number.

D1 is the automatic default if no D word is programmed. When D0 is programmed, then the offsets for the tool are not active.

Note: A maximum of 30 data fields with tool offset blocks can be stored in the control at a time.

2. **Programming**

D... ;Tool offset number: 1 ... 9

D0 : No offsets active

Information： Tool length compensations take immediate effect when the tool is active. The values of D1 are applied if no D number has been programmed. The tool length is compensated when the first programmed traversal of the relevant length compensation axis is executed.

A tool radius compensation must also be activated by means of G41/G42.

3. **Programming example**

Tool change **without M6 command** (only with T)**:**

N5 G17 ; Determines the axis assignment for compensations

N10 T1 ; Tool 1 is activated with the appropriate D1

N11 G0 Z... ; With G17, Z is the length compensation axis, the length offset compensation ;is overlaid here

N50 T4 D2 ; Load tool 4, D2 from T4 active

...

N70 G0 Z... D1 ; D1 for tool 4 active; only cutting edge changed Tool change using the **M6** **;** command:

N5 G17 ; Determines the axis assignment for compensations

N10 T1 ; Tool preselection

...

N15 M6 ; Tool change, T1 is active with the appropriate D1

N16 G0 Z... ; With G17, Z is the length compensation axis, the length offset compensation ;is overlaid here

...

N20 G0 Z... D2 ; D2 for tool 1 is active; with G17, Z is the length compensation axis,the ;difference of the length compensation D1–>D2 is overlaid here

N50 T4 ; Preselection of tool T4; please observe:T1 with D2 is still active !

...

N55 D3 M6 ; Tool change, T4 with the appropriate D3 is active

G41/G42 Selection of tool radius compensation

1. **Functionality**

compensation (tool nose radius compensation) is activated by G41/G42. The control then automatically calculates the necessary tool paths equidistant from the programmed contour for the current tool radius.

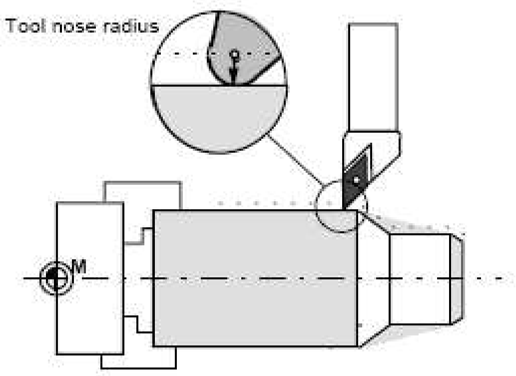


Fig 11.2-6

2. **Programming**

G41 X... Z... ;Tool radius compensation to left of contour

G42 X... Z... ;Tool radius compensation to right of contour

Note: You may only select the function for linear interpolation (G0, G1).

Program both axes. If you only specify one axis, then the last programmed value is automatically set for the second axis.

3. **Programming**

N10 T...

N20 G17 D2 F300 ; Offset no. 2, feedrate 300 mm/min

N25 X... Y. .. ; P0 – starting point

N30 G1 G42 X... Y... ; Selection right of the contour, P1

N31 X... Y.. . ; Starting contour, circle or straight line

After the selection, it is also possible to execute blocks that contain infeed motions or M outputs:

N20 G1 G41 X... Y... ; Selection left of the contour

N21 Z... ; Infeed motion

N22 X... Y... ; Starting contour, circle or straight line

G40 Tool radius compensation OFF

1. **Functionality**

The compensation mode (G41/G42) is deselected with G40. G40 is also the activation position at the beginning of the program.

The tool ends the **block in front of G40** in the normal position (compensation vector vertically to the tangent at the end point);

If G40 is active, the reference point is the tool center point. Subsequently, when deselected, the tool tip approaches the programmed point.

Always select the end point of the G40 block such that collision-free traversing is guaranteed!

2. Programming

G40 X... Y... ; Tool radius compensation OFF

Remark: The compensation mode can only be deselected with linear interpolation (G0, G1).

3. **Programming example**

N100 X... Y... ; Last block on the contour, circle or straight line, P1

N110 G40 G1 X... Y.. ; Deactivate tool radius compensation, P2

Subroutine

Programming example

Main：LF10.MPF

G54 T1 D0 G90 G00 X60 Z10

S800 M03

G01 X70 Z8 F0.1

X-2

G0 X70

L10 P3 ; Call subroutine L10.SPF 3 times

G0Z50

M05

M02

subroutine：L10.SPF

M03S600 ; subroutine directory

G01 G91 X-25 F0.1

X6 Z-3

Z-23.5

X15 Z-20.5

G02 X0 Z-71.62 CR=55

G03 X0 Z-51.59 CR=44

G01 Z-6.37

X14

X6 Z-3

Z-12

X10

X-32 Z194

G90

M02 ;return

11.3 CYCLES

Cycles are process-related subroutines that support general implementation of specific machining processes such as, for example, drilling, stock removal or thread cutting. The cycles are adapted to the specific problem in hand by means of supply parameters.

Standard cycles for turning and milling applications are provided in the system.

Standard cycles for turning

1. **Overview of cycles**

LCYC82 Drilling, spot facing

LCYC83 Deep hole drilling

LCYC840 Tapping with compensating chuck

LCYC84 Tapping without compensating chuck

LCYC85 Boring_1

2. **Defining parameters**

The arithmetic parameters from R100 to R149 are used as supply parameters for the cycles. Before a cycle is called, values must be assigned to its transfer parameters. Any parameters not needed must be loaded with zero. The values of these transfer parameters are unchanged after the cycle has been executed.

3. **Arithmetic parameters**

The cycles use the parameters R250 to R299 as internal arithmetic parameters. These are deleted when calling the cycles.

4. **Call and return** conditions

The drilling cycles are programmed independently of the particular axis names. The drilling position must be approached prior to calling the cycle in the higher-level program. The required values for feed, spindle speed and direction of rotation of the spindle must be programmed in the part program, if there are no supply parameters in the drilling cycle.

G0 G90 G40 are always effective at the end of a cycle.

5. **Recompilation** of cycles

The cycle can only be recompiled if the set of parameters stands immediantely before the cycle call. The parameters may not be separated by NC statements or comments.

6. Plane **definition**

All drilling and milling cycles assume that the current workpiece coordinate system in which machining is to be performed is defined by selecting a plane G17, G18 or G19 and activating a programmed frame (zero offset, rotation).

The drilling axis is always the 3rd axis of this system. Prior to the call, a tool with tool offset of this plane must be active. This remains active even after the cycle has been completed.

LCYC82 Drilling, spot facing

1. **Function**

The tool drills with the spindle speed and feedrate programmed down to the entered final depth. When the final drilling depth is reached, a dwell time can be programmed. The drill is retracted from the drill hole at rapid traverse rate.

2. **Call**

LCYC82

3. **Precondition**

The spindle speed and the direction of rotation, as well as the feed of the drilling axis must be defined in the higher-level program.

The drilling position must be approached before calling the cycle in the higherlevel program.

The required tool with tool offset must be selected before calling the cycle.

4. **Parameters**

Parameter	Meaning, Value Range
R101	Retract plane (absolute)
R102	Safety clearance
R103	Reference plane (absolute)
R104	Final drilling depth (absolute)
R105	Dwell time in seconds

Information:

R101 The retract plane determines the position of the drilling axis at the end of the cycle.

R102 The safety clearance acts on the reference plane, i.e. the reference plane is shifted forward by an amount corresponding to the safety clearance.

The direction in which the safety clearance acts is automatically determined by the cycle.

R103 The starting point of the drill hole shown in the drawing is programmed under the reference plane parameter.

R104 The drilling depth is always programmed as an absolute value with refer to workpiece zero.

R105 The dwell time at drilling depth (chip breakage) is programmed in seconds under R105.

5. **Motional sequence**

Position reached prior to beginning of cycle:last position in the higher-level program (drilling position)

The cycle produces the following motional sequence:

1) Approach reference plane shifted forward by an amount corresponding to the safety clearance using G0.

2) Traverse to final drilling depth with G1 and the feedrate programmed in the higher-level program.

3) Execute dwell time to final drilling depth.

4) Retract to retract plane with G0.

5. Example

```
N10 G0 G17 G90 F500 T2 D1 S500 M4    ; Define technology values
N20 X24 Y15                          ; Approach drilling position
N30 R101=110 R102=4 R103=102 R104=75 ; Supply parameters
N35 R105=2                           ; Supply parameters
N40 LCYC82                           ; Call cycle
N50 M2                               ; End of program
```

CYCLE83 Deep hole drilling

1. Function

The deep-hole drilling cycle produces center holes down to the final drilling depth by repeated, step-by-step deep infeed whose maximum amount can be parameterized. The drill can be retracted either to the reference plane for swarf removal after each infeed depth or by 1 mm in each case for chip breakage.

2. Call

LCYC83

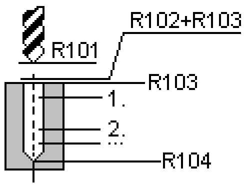
3. **Precondition**

The spindle speed and the direction of rotation must be defined in the higherlevel program.

The drilling position must be approached before calling the cycle in the higherlevel program.

Before calling the cycle, a tool offset for the drill must be selected.

4. Parameters



Parameter	Meaning, Value Range
R101	Retract plane (absolute)
R102	Safety clearance, enter without sign
R103	Reference plane (absolute)
R104	Final drilling depth (absolute)
R105	Dwell time to drilling depth (chip breakage)
R107	Feed for drilling
R108	Feed for first drilling depth
R109	Dwell time at starting point and for swarf removal
R110	First drilling depth(absolute)
R111	Absolute degression, enter without sign
R127	Machining type:Chip breakage = 0Swarf removal = 1

Note：

Information

R101 The retract plane determines the position of the drilling axis at the end of the cycle. The cycle is programmed on the assumption that the retract plane positioned in front of the reference plane, i.e. its distance to the final depth is greater.

R102 The safety clearance acts on the reference plane, i.e. the reference plane is shifted forward by an amount corresponding to the safety clearance.

The direction in which the safety clearance acts is automatically determined by the cycle.

R103 The starting point of the drill hole shown in the drawing is programmed under the reference plane parameter.

R104 The drilling depth is always programmed as an absolute value regardless of how G90/91 is set prior to cycle call.

R105 The dwell time at drilling depth (chip breakage) is programmed in seconds under R105.

R107, R108 The feed for the first drilling stroke (under R108) and for all subsequent drilling strokes (under R107) are programmed via these parameters.

R109 A dwell time at the starting point can be programmed in seconds under parameter R109. The dwell time at the starting point is executed only for the “with swarf removal” variant.

R110 Parameter R110 determines the depth of the first drilling stroke.

R111 Parameter R111 for the degression value determines the amount by which the current drilling depth is reduced with subsequent drilling strokes. The second drilling depth corresponds to the stroke of the first drilling depth minus the absolute degression value provided that this value is greater than the programmed absolute degression value. Otherwise, the second drilling depth also corresponds to the absolute degression value.

The next drilling strokes correspond to the absolute degression value provided that the remaining degression depth is still greater than twice the absolute degression value. The remainder is then distributed evenly between the last two drilling strokes.

If the value for the first drilling depth is in opposition to the total drilling depth, the error message 61107 “First drilling depth incorrectly defined” is displayed, and the cycle is not executed.

R127 Value 0: The drill travels 1 mm clear for chip breakage after it has reached each drilling depth. Value 1: The drill travels to the reference plane, which is shifted forward by an amount corresponding to the safety clearance for swarf removal after each drilling depth.

5. **Motional sequence**

Position reached prior to beginning of cycle:

last position in the higher-level program (drilling position)

The cycle produces the following motional sequence:

1) Approach reference plane shifted forward by an amount corresponding to the safety clearance using G0.

2) Traverse to first drilling depth with G1; the feedrate results from the feedrate programmed prior to cycle call after it has been computed with the setting in parameter R109 (feedrate factor).

Execute dwell time at drilling depth (parameter R105).

With chip breakage selected: Retract by 1 mm from the current drilling depth with G1 for chip breakage.

With swarf removal selected:

Retract for swarf removal to reference plane shifted forward by an amount corresponding to the safety clearance with G0 for swarf removal, executing the dwell time at starting point (parameter R106), approach last drilling depth minus clearance distance calculated in the cycle using G0,

3) Traverse to next drilling depth with G1 and the programmed feed; this motional sequence is continued as long as the final drilling depth is reached.

4) Retract to retract plane with G0.

5. Example

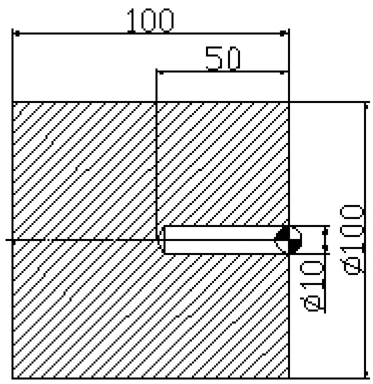


Fig 11.2-8

N10 T1D1 ;Define tool offset

N20 G0 X120 Z50

N30 M3 S500

N40 M8

N50 X0 Z50

N60 R101=50.000 R102=2.000 ; Define values

N70 R103=0.000 R104=-50.000

N80 R105=0.000 R107=200.000

N90 R108=100.000 R109=0.000

N100 R110=-5.000 R111=2.000

N110 R127=1.000

N120 LCYC83 ; call of cycle

N130 G0 X200 Z200

N140 M5 M9

N150 M2

LCYC840 Tapping with compensating chuck

1. Function

The tool drills with the programmed spindle speed and direction of rotation down to the entered thread depth. The feed of the drilling axis results from the spindle speed. This cycle can be used for tapping with compensating chuck and spindle actual-value encoder. The direction of rotation is automatically reversed in the cycle. The retract can be carried out at a separate speed.

2. Call LCYC84

3. Precondition

This cycle can only be used with a speed-controlled spindle with position encoder. The cycle does not check whether the actual-value encoder for the spindle really exists.

The spindle speed and the direction of rotation must be defined in the higherlevel program. The drilling position must be approached before calling the cycle in the higherlevel program.

The required tool with tool offset must be selected before calling the cycle.

4. Parameters declare

Parameter	**Meaning, Value Range**
R101	Retract plane (absolute)
R102	Safety clearance
R103	Reference plane (absolute)
R104	Final drilling depth (absolute)
R106	Thread lead as value.value range: 0.001 .... 2000.000 mm
R126	Direction of rotation of spindle for tapping.Value range: 3 (for M3), 4 (for M4)

Information：

R101-R104 See LCYC84

R106 Thread lead as value

R126 The tapping block is executed with the direction of rotation of spindle programmed under R126. The direction of rotation is automatically reversed in the cycle.

5. **Motional sequence**

Position reached prior to beginning of cycle:

last position in the higher-level program (drilling position)

The cycle produces the following motional sequence:

1. Approach reference plane shifted forward by an amount corresponding to the safety clearance using G0
2. Tapping down to final drilling depth with G33
3. Retract to reference plane shifted forward by an amount corresponding to the safety clearance with G33
4. Retract to retract plane with G0
5. Example

This program is used for tapping on the position X0; the Z axis is the drilling axis. The parameter for the direction of rotation R126 must be parameterized. A compensating chuck must be used for machining. The spindle speed is defined in the higher-level program.

```
N10 G0 G17 G90 S300 M3 D1 T1    ; Define technology values
N20 X35 Y35 Z60                 ; Approach drilling position
G17
N30 R101=60 R102=2 R103=56 R104=15 ; Parameter assignment
N40 R106=0.5 R126=3             ; Parameter assignment
N40 LCYC840                     ; Cycle call
N50 M2                          ; End of program
```

LCYC85 Boring

1. Function

The tool drills with the spindle speed and feedrate programmed down to the entered final drilling depth. When the final drilling depth is reached, a dwell time can be programmed. The approach and retract movements are carried out with the feedrates programmed under the respective parameters.

2. Call

LCYC85

3. **Precondition**

The spindle speed and the direction of rotation must be defined in the higherlevel program.

The drilling position must be approached before calling the cycle in the higherlevel program.

Before calling the cycle, the respective tool with tool offset must be selected.

4. **Parameters**

Parameter	Meaning, Value Range
R101	Retract plane (absolute)
R102	Safety clearance
R103	Reference plane (absolute)
R104	Final drilling depth (absolute)
R105	Dwell time at drilling depth in seconds
R107	Feed for drilling
R108	Feed when retracting from drill hole

Information：

Parameters R101 - R105 see LCYC82

R107 The feed value defined here acts for drilling.

R108 The feed value entered under R108 acts for retracting from the drill hole.

5. **Motional sequence**

Position reached prior to beginning of cycle: last position in the higher-level program (drilling position)

The cycle produces the following motional sequence:

1) Approach reference plane shifted forward by an amount corresponding to the safety clearance using G0

2)Traverse to final drilling depth with G1 and the feed programmed under parameter R106.

3)Execute dwell time at final drilling depth.

4) Retract to reference plane shifted forward by an amount corresponding to the safety clearance with G1 and the retract feed programmed under R108.

6. Example

The cycle LCYC85 is called in Z70 and X50 in the ZX plane. The Y axis is the drilling axis. No dwell time is programmed. The workpiece upper edge is at Y=102.

```
N10 G0 G90 G18 F1000 S500 M3 T1 D1      ; Define technology values
N20 Z70 X50 Y105                        ; Approach drilling position
N30 R101=105 R102=2 R103=102 R104=77 ; Define parameters
N35 R105=0 R107=200 R108=400            ; Define parameters
N40 LCYC85                              ; Call drilling cycle
```

N50 M2 ; End of program

LCYC93 Recess cycle

1. Function

The recess cycle is designed to produce symmetrical recesses for longitudinal and face machining on cylindrical contour elements. The cycle is suitable for machining internal and external recesses.

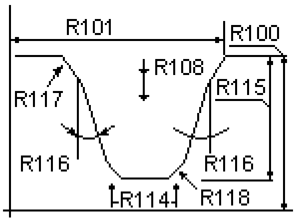
2. Call

LCYC93

3. **Precondition**

The recess cycle can only be called if G23 (diameter programming) is active. The tool offset of the tool whose tool nose width has been programmed with R107 must be activated before the recess cycle is called. The zero position of the tool nose faces machine zero.

4. **Parameters**



Parameter	Meaning, Value Range
R100	Starting point in facing axis
R101	Starting point in longitudinal axis
R105	Machining method, Value range 1 ... 8
R106	Finishing allowance, without sign
R107	Tool nose width, without sign
R108	Infeed depth , without sign
R114	Recess width, without sign
R115	Recess width, without sign
R116	Flank angle, without sign,between 0 <= R116 < = 89.999 degrees
R117	Chamfer on rim of recess
R118	Chamfer on recess base
R119	Dwell time on recess base

Information

R100 The recess diameter in X is specified in parameter R100

R101 R101 determines the point at which the recess starts in the Z axis.

R105 R105 defines the recess variant:

Value	Longitudinal/Facing	External/Internal	Starting Point Position
1	L	A	Left
2	P	A	Left

3	L	I	Left
4	P	I	Left
5	L	A	Right
6	P	A	Right
7	L	I	Right
8	P	I	Right

If the parameter is set to any other value, the cycle is aborted with the alarm 61002 “Machining type incorrectly programmed”.

R106 Parameter R106 determines the finishing allowance for roughing of the recess.

R107 Parameter R107 determines the tool nose width of the recessing tool. This value must correspond to the width of the tool actually used. If the tool nose of the active tool is wider, the contour of the programmed recess will be violated. Such violations are not monitored by the cycle. If the programmed tool nose width is wider than the recess width at the base, the cycle is aborted with the alarm G1602 “Tool width incorrectly defined”.

R108 By programming an infeed depth in R108, it is possible to divide the axisparallel recessing process into several infeed depths. After each infeed, the tool is retracted by 1 mm for chip breakage.

R114 The recess width programmed in parameter R114 is measured on the base.The chamfers are not included in the measurement.

R115 Parameter R115 determines the depth of the recess.

R116 The value of parameter R116 determines the angle of the flanks of the recess. When it is set to “0”, a recess with axis-parallel flanks (i.e. rectangular form) is machined.

R117 R117 defines the chamfers on the recess rim.

R118 R118 defines the chamfers on the recess base. If the values programmed for chamfers do not produce a meaningful recess contour, then the cycle is aborted with the alarm 61603 “Recess form incorrectly defined”.

R119 The dwell time on the recess base to be entered in R119 must be selected such that at least one spindle revolution can take place during the dwell period. It is programmed to comply with an F word (in seconds).

5. **Motional Sequence**

Position reached prior to beginning of the cycle:

- Any position from which each recess can be approached without risk of collision.

The cycle produces the following motional sequence:

- Approach with G0 starting point cacluated internally in the cycle.
- Execute depth infeeds:

  Roughing in parallel axes down to base, taking finishing allowance into account. Tool travels clear for chip breakage after each infeed.
- Execute width infeeds:

Width infeeds are executed perpendicular to the depth infeed with G0, the roughing process for machining the depth is repeated.

The infeeds both for depth and width are distributed evenly with the highest possible value.

- z Rough the flanks. Infeed along the recess width is executed in several steps if necessary.
- z Finish-machine the whole contour, starting at both rims and working towards center of recess base, at the feedrate programmed before the cycle call.

6. Example

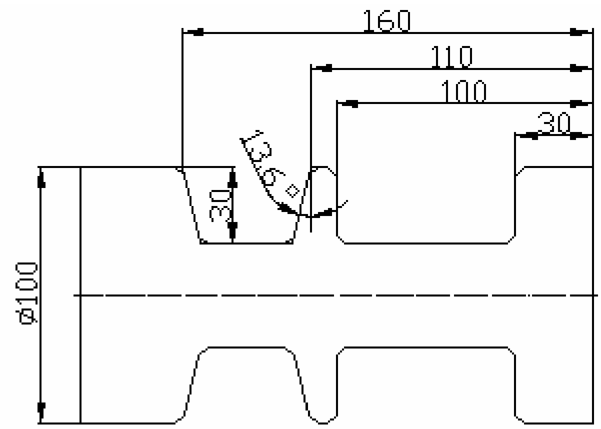


Fig 11.2-9

G55 G0 X0 Z0 M3 S1000 T01 D01

G0 X100

Z-50

R100=100 R101=-100 R105=1

R106=0 R107=3 R108=5

R114=70 R115=30 R116=0

R117=5 R118=5 R119=1

LCYC93

G0 X120

Z-50

R100=100 R101=-110 R105=5

R106=0 R107=3 R108=5

R114=50 R115=30 R116=13.6

R117=5 R118=5 R119=0.5

LCYC93

T01D00

M05

M2

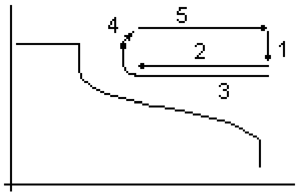
LCYC95 Stock removal cycle

1. Function

This cycle can machine a contour, which is programmed in a subroutine, in a longitudinal or face machining process, externally or internally, through axisparallel stock removal.
The technology (roughing/finishing/complete machining) can be selected. The cycle can be called from any chosen collision-free position. A tool offset must have been activated in the program with the cycle call.

2. Call

LCYC95



3. **Precondition**

- The cycle requires an active G23 (diameter programming).
- The file SGUD.DEF, which is supplied on the cycles diskette, must be available in the control system.
- The stock removal cycle can be called to the 3rd program level.

4. **Parameters**

Parameter	**Meaning, Value Range**
R105	Machining type：value range 1 ... 12
R106	Finishing allowance, without sign
R108	Infeed depth, without sign
R109	Infeed angle for roughing, it should be zero at face machining
R110	Contour clearance distance for roughing
R111	Feedrate for roughing
R112	Feedrate for finishing

Information

R105 The machining types:

- longitudinal/facing
- internal/external
- roughing/finishing/complete machining

are defined by the parameter determining the type of machining.When longitudinal machining is selected, the infeed always takes place in the facing axis, and vice versa.

Value	**Longitudinal/Facing(P)**	**External/Internal(A/I)**	**Roughing/Finishing/Complete Machining**
1	L	A	Roughing
2	P	A	Roughing
3	L	I	Roughing

4	P	I	Roughing
5	L	A	Finishing
6	P	A	Finishing
7	L	I	Finishing
8	P	I	Finishing
9	L	A	Complete
10	P	A	Complete
11	L	I	Complete
12	P	I	Complete

If any other value is programmed for the parameter, the cycle is aborted and the following alarm output 61002 “Machining type incorrectly programmed”.

R106 A finishing allowance can be programmed in parameter R106.

The workpiece is always rough-machined down to this finishing allowance. In this case, the residual corner produced in the course of each axis-parallel roughing process is immediately cut away in parallel with the contour at the same time. If no finishing allowance is programmed, the workpiece is roughmachined right down to the final contour.

R108 The maximum possible infeed depth for the roughing process is entered under parameter R108. However, the cycle itself calculates the current infeed depth that is applied in rough-machining operations.

R109 The infeed motion for roughing can be executed at an angle which can be programmed in parameter R109. In the face machining process a slanting immerse is not possible, R109 must be programmed to ZERO.

R110 Parameter R110 specifies the distance by which the tool is lifted from the contour in both axes after each roughing operation so that it can be retracted by G0.

R111 The feedrate programmed under R111 applies to all paths on which stock is removed during roughing operations. If finishing is the only machining type selected, then this parameter has no meaning at all.

R112 The feedrate programmed under R112 is applied for finishing operations. If roughing is the only machining type selected, then this parameter has no meaning at all.

Contour definition

The contour to be machined by stock removal is programmed in a subroutine. The name of the subroutine is transferred to the cycle via the _CNAME variable. The contour may consist of straight lines and circle segments; radii and chamfers can be inserted. The programmed circle sections can be quarter circles as a maximum.

Undercuts may not be contained in the contour. If an undercut element is detected, the cycle is aborted, and the alarm 61605 “Contour incorrectly defined” is output.

The contour must always be programmed in the direction that is traversed when finishing according to the selected machining direction.

Roughing

- Approach cycle starting point (calculated internally) with G0 in both axes simultaneously.
- Perform depth infeed with the angle programmed under R109 to the next roughing depth.
- Approach roughing cut point in parallel axes with G1 and at a federate programmed in R111.
- Travel in parallel with contour along contour + finishing allowance up to the last roughing cut point with G1/G2/G3 and at feedrate R111.
- Lift in each axis by the clearance (in mm) programmed in R110 and retract with G0.
- Repeat this sequence until the final roughing depth is reached.

Finishing

- Approach the cycle starting point in individual axes with G0
- Approach the contour starting point in both axes simultaneously with G0.
- Finish-machine along the contour with G1/G2/G3 and at the federate programmed in R112.
- Retract to cycle starting point in both axes with G0.

When finishing is selected, the tool radius compensation is automatically activated internally in the cycle.

Starting point

The cycle automatically calculates the point at which machining must start. The starting point is always approached in both axes simultaneously for roughing and in individual axes for finishing. In this case, the infeed axis approaches the starting point first.

When complete machining is selected, the tool does not return to the internally calculated starting point after the last roughing cut

5. Example

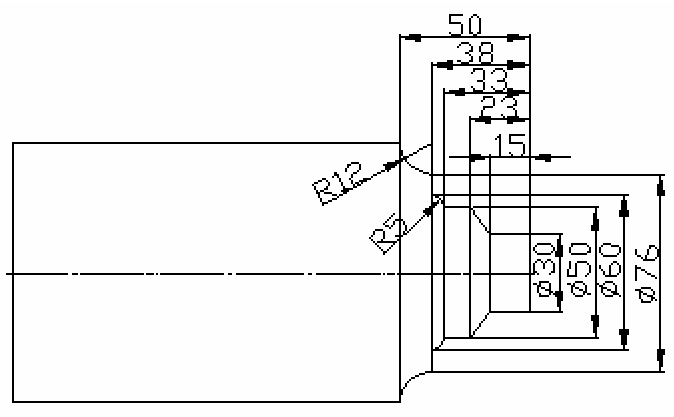


Fig 11.2-10

Main：LC95.MPF

G500 S500 M3 F0.4 T01 D01 ; setting workpiece

Z2 X142 M8

_CNAME="L01"

R105=1 R106=1.2 R108=5 R109=7

R110=1.5 R111=0.4 R112=0.25

LCYC95 ; call lcyc95

T02D01

R105=5 R106=0

LCYC95

G0 G90 X120

Z120 M9

M2

Subroutine：L01.SPF:

G0 X30 Z2

G01 Z-15 F0.3

X50 Z-23

Z-33

G03 X60 Z-38 CR=5

G01 X76

G02 X88 Z-50 CR=12

M02

LCYC97 Thread cutting

1. Function

The thread cutting cycle is suitable for cutting external and internal, single-start or multiple-start threads on cylindrical and tapered bodies in the facing or longitudinal axis. Depth infeed is an automatic function.

Whether a right-hand or left-hand thread is produced is determined by the direction of rotation of the spindle, which must be programmed before calling the cycle. Feed and spindle override are not effective in the traversing blocks containing thread cutting operations.

2. Call

LCYC97

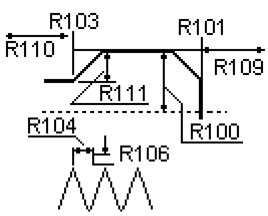


Fig 11.2-11

3. **Parameters**

Parameter	Meaning, Value Range
R100	Diameter of thread at starting point
R101	Thread starting point in longitudinal axis

R102	Diameter at end point
R103	Thread end point in longitudinal axis
R104	Thread lead as value, without sign
R105	Definition of thread cutting method:Value range: 1, 2
R106	Finishing allowance, without sign
R109	Approach path, without sign
R110	Run-out path, without sign
R111	Thread depth, without sign
R112	Starting point offset, without sign
R113	Number of rough cuts, without sign
R114	Number of threads, without sign

Information

R100, R101 These parameters define the thread starting point in X and Z.

R102, R103 The thread end point is programmed under R102 and R103. In the case of cylindrical threads, one of these parameters has the same value as R100 or R101.

R104 The thread lead is an axis-parallel value and is specified without sign.

R105 Parameter R105 defines whether the thread is machined internally or externally.

R105 = 1: External thread

R105 = 2: Internal thread

If the parameter is set to any other value, the cycle is aborted with the alarm 61002 “Machining type incorrectly programmed”.

R106 The programmed finishing allowance is subtracted from the specified thread depth. The remainder is divided into rough cuts.

The finishing allowance is removed in one cut after roughing.

R109, R110 Parameters R109 and R110 specifiy the internally calculated thread approach and run-out paths. The cycle shifts the programmed starting point forward by the approach distance. The run-out path extends the length of the thread beyond the programmed end point.

R111 Parameter R111 defines the total depth of the thread.

R112 An angle value can be programmed in this parameter. This value defines the point at which the first thread cut starts on the circumference of the turned part, i.e. it is a starting point offset.

Possible values for this parameter are between 0.0001 ... + 359.9999 degrees.

If no starting point offset is specified, the first thread automatically starts at the zero-degree marking.

R113 Parameter R113 determines the number of roughing cuts for thread cutting operations. The cycle independently calculates the individual, current infeed depths as a function of the settings in R105 and R111.

R114 This parameter specifies the number of threads. These are arranged symmetrically around the circumference of the turned part.

4 **Motional sequence**

Position reached prior to beginning of cycle:

- Any position from which the programmed thread starting point + approach path can be approached without risk of collision.

The cycle produces the following motional sequence:

- Approach starting point at the beginning of the approach path (calculated internally in the cycle) to cut first thread with G0.
- Infeed for rough cutting according to the infeed method defined under R105.
- Repeat thread cuts according to the programmed number of rough cuts.
- Remove the finishing allowance with G33.
- Repeat the whole sequence for every further thread.

5.Example

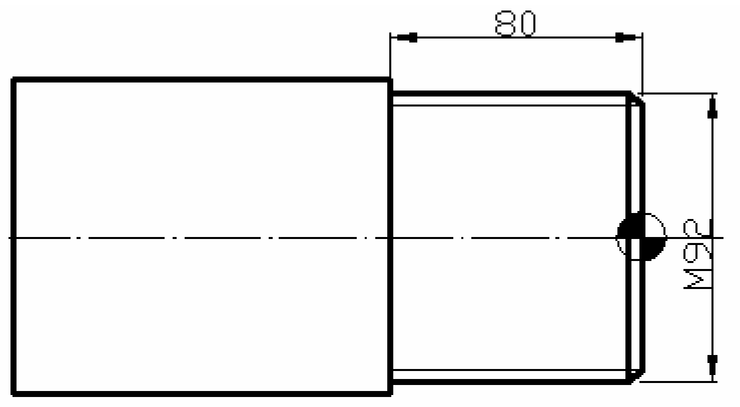


Fig 11.2-12

G55 G00 X0 Z0 M03 S1000 ;setting workpiece

T01 D01

G00 X100

Z50

R100=96 R101=0 R102=100 R103=-100

R104=2 R105=1 R106=0.5

R109=15 R110=35 R111=15

R112=0 R113=7 R114=1

LCYC97 ; call cycle

M05

M2

9.4 Arithmetic parameters R

1. Functionality

If you want an NC program in which you can vary the values to be processed, or if you simply needed to compute arithmetic values, then you can use R (arithmetic) parameters. The control system will calculate or set the values you need when the program is executed. An alternative method is to input the arithmetic parameter values directly. If the R parameters already have value settings, then they can be assigned in the program to other NC addresses that have variable values.

2.Programming

R0=...

to

R249=...

(to R299=..., if there are no machining cycles)

3. Explanation

250 arithmetic parameters with the following classification are available:

R0 ... R99 - for free assignment

R100 ... R249 - transfer parameters for machining cycles.

R250 ... R299 - internal arithmetic parameters for machining cycles.

If you do not intend to use machining cycles (see Section NO TAG “Machining Cycles”), then this range of arithmetic parameters is also available for your use.

4. Value assignment

Example：

R0=3.5678 R1=-37.3 R2=2 R3=-7 R4=-45678.1234

You can assign an extended numerical range using exponential notation:（10^{-300}...10^{+300}）.

The value of the exponent is typed after the characters EX. Maximum number of characters: 10 (including sign and decimal point).

Value range of EX: -300 to +300.

Example：

R0=-0.1EX-5 ;Meaning: R0 = -0,000 001

R1=1.874EX8 ;Meaning: R1 = 187 400 000

Note: Several assignments (including arithmetic expressions) can be programmed in one block.

5. Assignment to other addresses

You can obtain a flexible NC program by assigning arithmetic parameters or arithmetic expressions with R parameters to other NC addresses. Values, arithmetic expressions or R parameters can be assigned to any NC address with the exception of addresses N, G and L.

When making assignments of this kind, type the character “=” after the address character.

Assignments with a negative sign are also permitted.

If you wish to make assignments to axis addresses (traversal instructions), then you must do so in a separate program block.

Example：

N10 G0 X=R2 ;Assignment to X axis

6. Arithmetic operations functions

Operators/arithmetic functions must be programmed using the normal mathe / matical notation.

Processing priorities are set by means of round brackets. Otherwise the “multiplication/division before addition/subtraction” rule applies. Degrees are specified for trigonometric functions.

9.5 Program jumps

9.5.1 **label ---** Jump destination for program jumps

1. Functionality

1) A **label** or a **block number** serve to mark blocks as jump destinations for program jumps. Program jumps can be used to branch to the program sequence.

2) Labels can be freely selected, but must contain a minimum of 2 and a maximum of 8 letters or numbers, and the **first two characters** must be **letters** or underscores.

3) Labels that are in the block that serves as the jump destination are **ended by a colon**. They are always at the start of a block. If a block number is also present, the label is located **after the block number**.

4) Labels must be unique within a program.

2. Programming example

N10 LABEL1: G1 X20 ; LABEL1 is the label, jump destination

...

TR789: G0 X10 Z20 ; TR789 is the label, jump destination

– No block number existing

N100 .. . ; A block number can be a jump destination.

9.5.2 Unconditional program jumps

1. Functionality

NC programs process their blocks in the sequence in which they were arranged when they were written.

The processing sequence can be changed by introducing program jumps.

The jump destination can be a block with a **label** or with a **block number**. This block must be located within the program.

The unconditional jump instruction requires a separate block

2. Programming

GOTOF Lable ; GoTo operation

GOTOB Lable ;GoBack operation

AWL	Note
GOTOF	; GoTo operation (in the direction of the last block of the program)
GOTOB	;GoBack operation (in the direction of the first block of the program)
Lable	; Selected string for the label (jump label) or for the block number

9.5.3Conditional program jumps

1. Functionality

Jump conditions are formulated after the **IF instruction**. If the jump condition (**value not zero**) is satisfied, the jump takes place.

The jump destination can be a block with a **label** or with a **block number**. This block must be located within the program.

Conditional jump instructions require a separate block. Several conditional jump instructions can be located in the same block.

By using conditional program jumps, you can also considerably shorten the program, if necessary.

2.Programming

IF *condition* GOTOF *label* ; GoTo operation (forward jum)

IF *condition* GOTOB *label* ; GoBack operation (reverse jump)

AWL	**Meaning**
GOTOF	Jump direction forward (in the direction of the last block of the program)
GOTOB	Jump direction reverse (in the direction of the first block of the program)
Lable	Selected string for the label (jump label) or for the block number
IF	Introduction of the jump condition
Condition	R parameter, arithmetic expression for formulating the condition

3. Comparison operations

Operators	**Meaning**
==	Equal to
<>	Not equal to
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to

The comparison operations support formulating of a jump condition. Arithmetic expressions can also be compared.

The result of comparison operations is ”satisfied” or ”not satisfied.” ”Not satisfied” sets the value to zero.

4. Programming example for comparison operators

R1>1 ;R1 greater than 1

1 < R1 ;1 less than R1

R1<R2+R3 ;R1 less than R2 plus R3

R6>=SIN(R7*R7) ;R6 greater than or equal to SIN(R7)^2

9.5.4 Programming example

Task

Approaching points on a circle segment:

Given: Starting angle: 30 . in R1

Circle radius: 32 mm in R2

Spacing between the positions: 10 . in R3

Number of points: 11 in R4

Position of the circle center in Z: 50 mm in R5

Position of the circle center in X: 20 mm in R6□

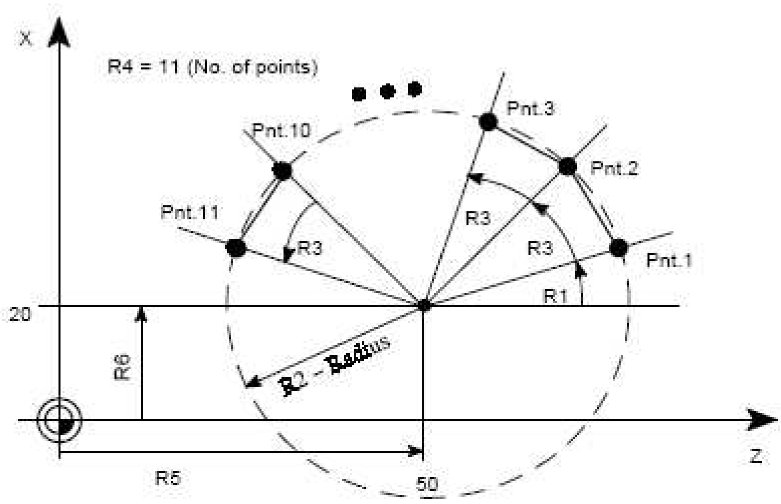


Fig 11.2－13

Programming example

N10 R1=30 R2=32 R3=10 R4=11 R5=50 R6=20 ; Assignment of the starting values

N10 MA1: G0 Z=R2 *COS (R1)+R5 X=R2*SIN(R1)+R6

; Calculation and assignment to axis addresses

N30 R1=R1+R3 R4= R4–1

N40 IF R4 > 0 GOTOB MA1

N50 M2

Explanation

In block N10, the starting conditions are assigned to the corresponding arithmetic parameters. The calculation of the coordinates in X and Z and the processing takes place in N20.

In block N30, R1 is incremented by the clearance angle R3, and R4 is decremented by 1. If R4 > 0, N20 is executed again; otherwise, N50 with end of program.

9.6 Subroutine

1. Application

Basically, there is no difference between a main program and a subroutine.

Frequently recurring machining sequences are stored in subroutines, e.g certain contour shapes. These subroutines are called at the appropriate locations in the main program and then executed.

One form of subroutine is the **machining cycle**. Machining cycles contain universally valid machining scenarios (e.g.: drilling, tapping, groove milling, etc.). By assigning values via included transfer parameters, you can adapt the subroutine to your specific application.

2. Structure

The structure of a subroutine is identical to that of a main program .Like main programs, subroutines contain **M2 – end of program** in the last block of the program sequence. This means a return to the program level where the subroutine was called from.

3. End of program

The end instruction **RET** can also be used instead of the M2 program end in the subroutine.

RET requires a separate block.

The RET instruction is used when G64 continuous-path mode is not to be interrupted by a return. With M2, G64 is interrupted and exact stop is initiated.

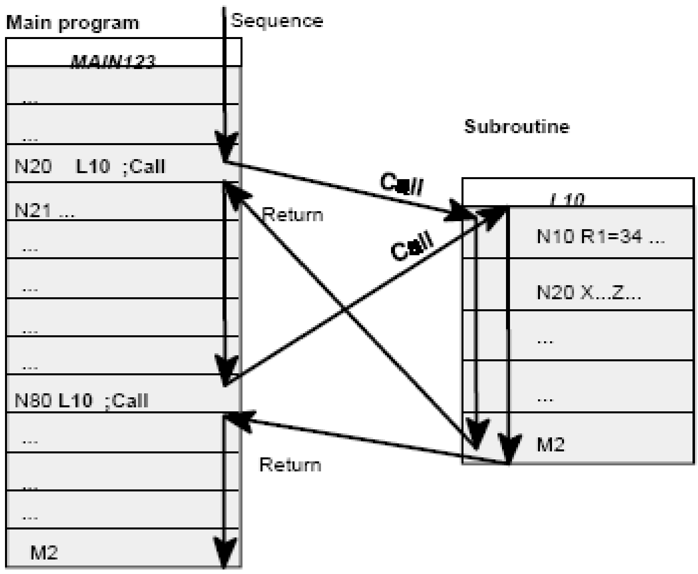


Fig 11.2－14

4. Subroutine name

The subprogram is given a unique name allowing it to be selected from several subroutines. When you create the program, the program name may be freely selected provided the following conventions are observed:

_ The first two characters must be letters

_ The others may be letters, digits or underscore

_ Maximum of 8 characters in total

_ No dashes (see Section “Character set”)

The same rules apply as for main program names.

5. Subroutine call

Subroutines are called in a program (main or subprogram) with their names. To do this, a separate block is required.

Example

N10 L785 ;Call of subroutine L785

N20 WELLE7 ;Call of subroutine WELLE7

6. Program repetition P...

If a subroutine is to be executed several times in succession, write the number of times it is to be executed in the block of the call after the subroutine name under the **address P**. A maximum of **9,999 cycles** are possible (P1 ... P9999).

Example

N10 L785 P3 ; Call of subroutine L785 , 3 passes

7. Nesting depth

It is not only possible to call subroutines in main programs, but also in other subroutines. There is a total of 4 program levels (including the main program level) available for programming this type of nested call.

Note: If you are working with machining cycles, please remember that these also need one of the four program levels.

8. Information

Modal G functions can be changed in the subroutine, e.g. G90 –> G91. When returning to the calling program, ensure that all modal functions are set the way you need them to be.

Please make sure that the values of your arithmetic parameters used in upper program levels are not inadvertently changed in lower program levels.

When working with SIEMENS cycles, up to 4 program levels are needed.

Address: Junlin Guoji Building, 5 Guangzhou Road,Suite A 1306, Nanjing, Jiangsu
210008 CHINA
Phone : 086-025-51860015
Fax : 086-025-51860015
Http : www.swansc.com
E-mail: sales@swansc.com
Najing Swan Software Technology Company