Typ3 osa

DIN Programming Instructions
Part 1 and Part 2

1070 073 738-109 (00.05) GB

© 1993 - 2000
by Robert Bosch GmbH, Erbach / Germany
All rights reserved, including applications for protective rights.
Reproduction or distribution by any means subject to our prior written permission.

Discretionary charge DM 30.00
## Contents

### Part 1:

<table>
<thead>
<tr>
<th>1</th>
<th>Safety Instructions</th>
<th>1–1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Proper use</td>
<td>1–1</td>
</tr>
<tr>
<td>1.2</td>
<td>Qualified personnel</td>
<td>1–2</td>
</tr>
<tr>
<td>1.3</td>
<td>Safety markings on components</td>
<td>1–3</td>
</tr>
<tr>
<td>1.4</td>
<td>Safety instructions in this manual</td>
<td>1–4</td>
</tr>
<tr>
<td>1.5</td>
<td>Safety instructions for the described product</td>
<td>1–5</td>
</tr>
<tr>
<td>1.6</td>
<td>Documentation, software release and trademarks</td>
<td>1–7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2</th>
<th>Basics</th>
<th>2–1</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Function and structure of an NC program</td>
<td>2–1</td>
</tr>
<tr>
<td>2.1.1</td>
<td>Programming principles</td>
<td>2–1</td>
</tr>
<tr>
<td>2.1.2</td>
<td>Program design elements</td>
<td>2–5</td>
</tr>
<tr>
<td>2.1.3</td>
<td>Subprograms</td>
<td>2–7</td>
</tr>
<tr>
<td>2.1.4</td>
<td>Jump destinations and jump instructions</td>
<td>2–10</td>
</tr>
<tr>
<td>2.1.5</td>
<td>Typ3 osa standard programming formats</td>
<td>2–13</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3</th>
<th>Axes and coordinate systems</th>
<th>3–1</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Physical and logical axes</td>
<td>3–1</td>
</tr>
<tr>
<td>3.2</td>
<td>Axis and working range coordinates</td>
<td>3–2</td>
</tr>
<tr>
<td>3.3</td>
<td>Axis and coordinate designations</td>
<td>3–4</td>
</tr>
<tr>
<td>3.4</td>
<td>Coordinate systems</td>
<td>3–6</td>
</tr>
<tr>
<td>3.4.1</td>
<td>ACS axes coordinate system</td>
<td>3–6</td>
</tr>
<tr>
<td>3.4.2</td>
<td>MCS machine coordinate system</td>
<td>3–6</td>
</tr>
<tr>
<td>3.4.3</td>
<td>WCS workpiece coordinate system</td>
<td>3–8</td>
</tr>
<tr>
<td>3.4.4</td>
<td>PCS program coordinate system</td>
<td>3–8</td>
</tr>
<tr>
<td>3.4.5</td>
<td>TCS tool coordinate system</td>
<td>3–9</td>
</tr>
<tr>
<td>3.5</td>
<td>NC functions for defining coordinate systems</td>
<td>3–10</td>
</tr>
<tr>
<td>3.5.1</td>
<td>Axis transformation: Axis coordinates and machine coordinates</td>
<td>3–10</td>
</tr>
<tr>
<td>3.5.2</td>
<td>Coordinate transformation: Machine coordinates, workpiece coordinates and program coordinates</td>
<td>3–10</td>
</tr>
<tr>
<td>3.5.3</td>
<td>Functions for manipulating the programmed contour</td>
<td>3–14</td>
</tr>
<tr>
<td>3.5.4</td>
<td>Functions for tool compensation</td>
<td>3–15</td>
</tr>
<tr>
<td>3.5.5</td>
<td>Tool movements (interpolation types)</td>
<td>3–20</td>
</tr>
</tbody>
</table>

| 4 | G instructions | 4–1 |

See also the table of all G instructions and functions in the Annex.
Part 2:

### 5 Spindles

- Individual spindles, spindle groups and channels ........................................ 5–1
- Assigning individual spindles to spindle groups ........................................ 5–2
- Reserving spindles and spindle groups for specific channels ...................... 5–4
- Functions for individual spindles and spindle groups (M functions) in speed mode .................................................. 5–7
  - Spindle functions ........................................................................ 5–7
  - Gear functions ........................................................................ 5–9
  - Specifying the spindle speed ........................................................... 5–11
  - Activating spindles via machine functions ........................................... 5–12
  - Activating spindles via the interface ................................................. 5–13
  - G functions involving spindle programming ....................................... 5–14
  - Special spindle functions ................................................................. 5–15
  - Spindle functions in position mode .................................................... 5–15
  - Spindle referencing ....................................................................... 5–15
  - Switching to position-controlled spindle operation ................................ 5–15
  - Spindle operation in positional synchronism ........................................ 5–16
  - Configuring slave spindles ................................................................. 5–17
  - Defining groups of coupled spindles ................................................... 5–18
  - Programming while coupling is active ............................................... 5–20
  - Spindle coupling process .................................................................. 5–21
  - Test mode with spindle coupling active ............................................. 5–22
  - Effects of spindle-specific interface signals on spindle couplings ......... 5–22
  - Effects of drive-specific messages on spindle couplings .................... 5–23
  - Notes for commissioning (spindle position mode) .............................. 5–24

### 6 Auxiliary and special functions

- F address (feedrate) ........................................................................ 6–2
- FA address (feedrate of asynchronous axes, auxiliary axes) ..................... 6–3
- M functions ..................................................................................... 6–4
- Subprogram calls ............................................................................ 6–4
- Stop processing M00, M01, M02/M30 ................................................. 6–5
- Tool change M6 .............................................................................. 6–6
- T address (tool selection) .................................................................. 6–7

### 7 Tool compensation (tool correction)

- Length compensation ........................................................................ 7–1
- Radius compensation ......................................................................... 7–2
- Selection and deselection of the cutter path compensation .................... 7–3
- Selection of cutter path compensation ................................................ 7–3
- Deselection of the cutter path compensation ....................................... 7–4
- Examples ......................................................................................... 7–5

### 8 Leadscrew error compensation (LSEC)

- Determination of the compensation values of an axis ......................... 8–1
- Programming the compensation values ............................................... 8–3
- Activating the compensation values .................................................... 8–5
<table>
<thead>
<tr>
<th>Contents VII</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
</tr>
<tr>
<td>9.1</td>
</tr>
<tr>
<td>9.2</td>
</tr>
<tr>
<td>9.3</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>10.1</td>
</tr>
<tr>
<td>10.2</td>
</tr>
<tr>
<td>10.3</td>
</tr>
<tr>
<td>A</td>
</tr>
<tr>
<td>A.1</td>
</tr>
<tr>
<td>A.2</td>
</tr>
<tr>
<td>A.3</td>
</tr>
<tr>
<td>A.4</td>
</tr>
</tbody>
</table>
1 Safety Instructions

Please read this manual before programming the Typ3 osa or modifying existing programs. Store this documentation in a place to which all users have access at any time.

1.1 Intended use

This manual contains all information required for the proper use of the control unit. For reasons of clarity, however, it cannot contain each and every detail about each and all combinations of functions. Likewise, as the control unit is usually part of a larger installation or system, it is impossible to consider each and any aspect of integration or operation.

The Typ3 osa is used to

- activate feed drives, spindles and auxiliary axes of a machine tool via SERCOS interface for the purpose of guiding a processing tool along a programmed path to process a workpiece (CNC). Furthermore, a PLC is required with appropriate I/O components which – in communication with the actual CNC – controls the machine processing cycles holistically and acts as a technical safety monitor.
- program contours and the processing technology (path feedrate, spindle speed, tool change) of a workpiece.

Any other application is deemed improper use!

The product described

- has been developed, manufactured, tested and documented in compliance with the safety standards. This product poses no danger to persons or property if it is used in accordance with the handling stipulations and safety notes prescribed for its configuration, mounting, and proper operation.
- complies with the requirements of
  - the EMC Directives (89/336/EEC, 93/68/EEC and 93/44/EEC)
  - the Low-Voltage Directive (73/23/EEC) the harmonized standards EN 50081-2 and EN 50082-2
  - the harmonized standards EN 50081-2 and EN 50082-2
- is designed for operation in industrial environments (emission class A), i.e.
  - no direct connection to public low-voltage power supply,
  - connection to the medium- or high-voltage system via a transformer.

In residential environments, in trade and commerce as well as small enterprises class A equipment may only be used if it does not inadmissibly interfere with other equipment.

This is a class A device which may cause radio interference in residential environments. In this case, the operator may be required to take suitable countermeasures and to bear the cost of the same.

The faultless, safe functioning of the product requires proper transport, storage, erection and installation as well as careful operation.
1.2 Qualified personnel

The requirements as to qualified personnel depend on the qualification pro-
files described by ZVEI (central association of the electrical industry) and
VDMA (association of German machine and plant builders) in:
Weiterbildung in der Automatisierungstechnik
edited by: ZVEI and VDMA
MaschinenbauVerlag
Postfach 71 08 64
D-60498 Frankfurt.

The present manual is designed for

- NC programming personnel and NC project engineers.

These persons need special knowledge of
- the operation, syntax and instruction set of the DIN programming lan-
guage.

Programming, start and operation as well as the modification of program
parameters is reserved to properly trained personnel! This personnel must
be able to judge potential hazards arising from programming, program
changes and in general from the mechanical, electrical, or electronic equip-
ment.

Interventions in the hardware and software of our products, unless de-
scribed otherwise in this manual, are reserved to our specialized personnel.

Tampering with the hardware or software, ignoring warning signs attached to
the components, or non-compliance with the warning notes given in this
manual can result in serious bodily injury or property damage.

Only electrotechnicians as recognized under IEV 826-09-01 (modified) who
are familiar with the contents of this manual may install and service the prod-
ucts described.

Such personnel are
- those who, being well trained and experienced in their field and familiar
  with the relevant norms, are able to analyze the jobs being carried out
  and recognize any hazards which may have arisen.
- those who have acquired the same amount of expert knowledge through
  years of experience that would normally be acquired through formal tech-
  nical training.

Please note our comprehensive range of training courses.
Our training center will be pleased to provide you with further information,
telephone: +49 (0) 6062 78-258.
1.3 Safety markings on products

- Warning of dangerous electrical voltage!
- Warning of danger caused by batteries!
- Components sensitive to electrostatic discharge!
- Warning of hazardous light emissions (optical fiber cable emissions)
- Disconnect from mains before opening!
- Pin for connecting PE conductor only!
- Connection of shield conductor only
1.4 Safety instructions in this manual

---

**DANGEROUS ELECTRICAL VOLTAGE**
This symbol is used to warn of a dangerous electrical voltage. The failure to observe the instructions in this manual in whole or in part may result in personal injuries.

---

**DANGER**
This symbol will be used if the failure to observe the instructions in this manual in whole or in part may result in personal injuries.

---

**CAUTION**
This symbol will be used if the failure to observe the instructions in this manual in whole or in part may result in damages to equipment or data files.

---

This symbol will be used to draw the user’s attention to special circumstances.

★ This symbol will be used if user activities are required.
1.5 Safety instructions concerning the product described

DANGER
Danger of life through inadequate EMERGENCY-STOP devices!
EMERGENCY-STOP devices must be active and within reach in all system modes. Releasing an EMERGENCY-STOP device must not result in an uncontrolled restart of the system!
First check the EMERGENCY-STOP circuit, then switch the system on!

DANGER
Incorrect or undesired axis movement!
First, new programs should be tested carefully without axis movement! For this purpose, the control offers the possibility of inhibiting axis movements and/or auxiliary function outputs by appropriate softkeys in the “automatic” group operating mode.

DANGER
Incorrect or undesired control unit response!
Bosch accepts no liability for damage resulting from the CNC parameter settings, the execution of an NC program, an individual NC block or the manual movement of axes!
Furthermore, Bosch accepts no liability for consequential damages which could have been avoided by programming the PLC appropriately!

DANGER
Retrofits or modifications may adversely affect the safety of the products described!
The consequences may include severe injuries, damage to equipment, or environmental hazards. Possible retrofits or modifications to the system using third-party equipment therefore have to be approved by Bosch.

DANGEROUS ELECTRICAL VOLTAGE
Unless described otherwise, maintenance works must be performed on inactive systems! The system must be protected against unauthorized or accidental reclosing.
Measuring or test activities on the live system are reserved to qualified electrical personnel!
DANGER
Tool or axis movements!
Feed and spindle motors generate very powerful mechanical forces and can accelerate very quickly due to their high dynamics.
- Always stay outside the danger area of the machine when it is running!
- Do not ever deactivate the safety-relevant functions of the unit!
- Report any malfunction of the unit to your servicing and repairs department immediately!

CAUTION
Only spare / replacement parts approved by Bosch may be used!

CAUTION
Danger to the module!
All ESD protection measures must be observed when using the module! Prevent electrostatic discharges!

The following protective measures must be observed for modules and components sensitive to electrostatic discharge (ESD):
- The personnel responsible for storage, transport, and handling must have been trained for ESD protection.
- ESD-sensitive components must be stored and transported in their prescribed protective packaging.
- ESD-sensitive components may only be handled at special ESD-workplaces.
- Personnel, working surfaces, as well as all equipment and tools which get in contact with ESD-sensitive components must have the same potential (e.g., by grounding).
- Wear an approved grounding bracelet. The grounding bracelet must be connected with the working surface through a cable with an integrated resistor of 1 MΩ.
- ESD-sensitive components must by no means get in contact with chargeable objects, including most plastic materials.
- When ESD-sensitive components are installed in or removed from equipment, the equipment must be de-energized.
1.6 Documentation, software release and trademarks

Documentation

The present manual provides information on

- the operation, syntax and instruction set of the DIN programming language.

<table>
<thead>
<tr>
<th>Overview of available documentation</th>
<th>Part no.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Part no.</td>
<td>German</td>
</tr>
<tr>
<td>Interface conditions for project engineering and maintenance</td>
<td>1070 073 704</td>
</tr>
<tr>
<td>Operating instructions Standard operator interface</td>
<td>1070 073 726</td>
</tr>
<tr>
<td>Operating instructions – Diagnostics Tools</td>
<td>1070 073 779</td>
</tr>
<tr>
<td>DIN programming instructions for programming to DIN 66025</td>
<td>1070 073 725</td>
</tr>
<tr>
<td>CPL programming instructions</td>
<td>1070 073 727</td>
</tr>
<tr>
<td>ICL700 system description, Program structure of the integrated PLC</td>
<td>1070 073 706</td>
</tr>
<tr>
<td>ICL700 project planning manual, software interfaces and CNC interface signals of the integrated PLC</td>
<td>1070 073 728</td>
</tr>
<tr>
<td>MACODA Operation and configuration of the machine parameters</td>
<td>1070 073 705</td>
</tr>
<tr>
<td>Tool Management – Parameterization</td>
<td>1070 073 782</td>
</tr>
<tr>
<td>Software PLC Development environment for Windows NT</td>
<td>1070 073 783</td>
</tr>
<tr>
<td>Measuring cycles for touch-trigger switching probes</td>
<td>1070 073 788</td>
</tr>
<tr>
<td>Universal Milling Cycles</td>
<td>–</td>
</tr>
<tr>
<td>Software installation</td>
<td>1070 073 796</td>
</tr>
</tbody>
</table>

Special keys or combinations of keys are represented by pointed brackets:

- Special keys: e.g. <enter>, <pgup>, <del>
- Key combinations (pressed simultaneously): e.g. <ctrl> + <pgup>

Release


This manual refers to the following product version:
Software: V5.1.x

The current release number of the individual software modules can be viewed by selecting the "Control-Diagnostics" softkey in the "Diagnostics" group operating mode.

The software version of Windows95 or WindowsNT may be displayed as follows:
1. click with right mouse key on the "My Computer" icon on your desktop
2. select menu item "Properties".
Trademarks

All trademarks of software installed on Bosch products upon delivery are the property of the respective manufacturer.

Upon delivery, all installed software is copyright-protected. The software may only be reproduced with the approval of Bosch or in accordance with the license agreement of the respective manufacturer.

MS-DOS® and Windows™ are registered trademarks of Microsoft Corp.

PROFIBUS® is a registered trademark of PROFIBUS Nutzerorganisation e. V.

SERCOS interface® is a registered trademark of Interessengemeinschaft SERCOS interface e.V.
2 Basics

2.1 Function and structure of an NC program

The NC program serves to provide the control unit with all information required for machining on the machine.

The structure of an NC program is variable. Only the guidelines are summarized in DIN 66025*). In this publication you can find the rules according to which the programming blocks are to be formed in the NC program.

* The contents of DIN 66025, ‘Program structure for numerically controlled machines’ (Parts 1 and 2) corresponds to the ISO/DIS 6983 and ISO/DP 6983 international standards, ‘Numerical control of machines’.

The Typ3 osa offers two different ways of programming:
- DIN 66025 programming
- CPL programming

The present manual covers programming according to DIN 66025. Machine-dependent cycles (MTB cycles) are not considered.

All NC programs (part programs) are maintained in the “File System” of the Typ3 osa.

For the structure as well as for detailed explanations of the File System and the File Protection (access rights), please refer to the section “Directories” of the Typ3 osa operating instructions. The operating instructions will also give you information about the new creation and editing of part programs.

2.1.1 Programming principles

Workpiece contours are divided into straight lines and circular arcs. The control unit is able to execute the respective movements required for each of these geometrically ‘simple’ contour elements in one machining step – a program block. As a prerequisite, all the machining steps must be determined in the correct sequence and with all necessary marginal conditions within the NC program.

The NC program consists of individual program blocks. These contain preparatory functions, positional data, auxiliary and special functions. These blocks are used to enter details about the position, the technology and the program flow.

The program end is indicated by "M02" or "M30" at the beginning of the last program line (for details, please refer to section 6.3.2).

The memory available (e.g. for NC programs) depends on the control memory option (osa master L/XL).
Example: Procedure for machining

- Breaking down the machining process into logical (and possibly recurrent) sections
- Breaking down the contour into “simple” consecutive contour elements
- Creation of the program (incl. subprograms, if any), program input into the CNC
- Program start
- CNC controls the machining of the workpiece

Program blocks

The control unit executes the program blocks one by one.

Each program block consists of a set of program words which, in turn, are made up of an address letter and a string of digits.

Example: A program block comprising 10 program words

N. G. {optional parameters {=} <value>} X. Y. Z. F. S. T. M.

<table>
<thead>
<tr>
<th>block number</th>
<th>instruction</th>
<th>equals sign, optional parameter contents</th>
<th>path commands</th>
<th>tool number</th>
<th>M function</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>option instruction</td>
<td></td>
<td></td>
<td>spindle speed</td>
<td></td>
</tr>
</tbody>
</table>

Program words

Each program word of a block may consist of an address letter and a number (e.g. G00, X−23.450, Y40, M03, S250).

Example: Program word

X−2407.0458

<table>
<thead>
<tr>
<th>decimal value</th>
<th>integer value</th>
<th>sign</th>
<th>address</th>
</tr>
</thead>
</table>

- Leading zeros do not have to be programmed
- Non-integers are written with a decimal point; trailing zeros may be omitted (e.g. “X100.500” corresponds to “X100.5”).
- The CNC processes variable block lengths. The number of words per block may vary.
- Words containing positional data determine the tool path. These may also include a sign (+/−). If no sign is programmed, the positive value is assumed. In the case of a negative value, you must program a minus sign.
Some G-functions include **optional** program words. These are set between **braces**, which are to be omitted for programming.

**Example:**

Syntax definition:  
G631 {SYM<s>} {ANG<a>}

Programming:  
G631 SYM2 ANG10

Setting the SYM<s> parameter as shown above between braces {...} is optional. If this parameter is omitted in the programmed instruction, a preset value (s) from the MACODA is automatically assigned to the SYM parameter.

**Example:**

Syntax definition:  
G612 <axis name i><timeaxis name i>

Programming:  
G612 X10

The "axis name i" between the <...> angle brackets denotes the physical i-axis, e.g.

- x-axis=1<sup>st</sup> physical axis
- y-axis=2<sup>nd</sup> physical axis

"Time<axis name i>" is 10 ms and refers only to the x-axis.

Program words put between **square** brackets represent various parameters of one and the same category or are to be assigned specific values. If required, these program words must be set correspondingly when programming.

**Example:** Parameters for modal subprograms

Syntax definition:  
G81 [<parameter 1>,<parameter 2>, 
{<parameter 3>},{<parameter 4>}]

Programming:  
G81 [Z,R1,P,R2]

**Modal effect**  
Most words have a modal effect. This means that their effect remains in force until you program the same word with a different value, or until you switch off the word's function.

**Example:**

As soon as you have programmed G1 (linear interpolation at feed) in a program block, the control unit will approach all subsequently specified positions at feed without the necessity of having to program G1 again. G1 remains effective until you program a different interpolation type (e.g. “G2”: circular interpolation or “G0”: linear interpolation in rapid feed).

**Nonmodal effect**  
Words of this type are only effective within the block in which they were programmed.
In terms of their effect, program words act either as instructions or special functions.

For instance, the CNC must be told in which manner and to which position a tool is supposed to travel. This positional data is communicated to the CNC via the G address (manner of travelling) and the X, Y, Z, C addresses etc. (where to travel).

**G address**

G addresses are used to program the type of traversing movement (e.g. rapid feed, linear or circular interpolation etc.); this is also the reason why reference to ‘preparatory functions’ is made.

All preparatory functions are ‘sorted’ in **groups**. Preparatory functions from different groups do not interfere with each other. However, because preparatory functions contained within one and the same group act modally, not more than one G instruction from any one group may be used per program block.

From section 4 on, you will find a list of all preparatory functions which are recognized by the CNC. It also describes their respective groups.

**Addresses X, Y, Z, C etc.**

You use these addresses to determine the axis that is to travel to a specific position or over a specific distance.

The address for identifying an axis (axis name) can be set using the 100100001 configuration parameter. For this reason, axis names may also end in a numeral (e.g. "X1", "X2", "B1", "PALLET1" etc.). If this is the case, the "=" sign or a blank must be programmed between the axis name and any subsequent positional/path information!

**Example:**

G1 B1=90 or G1 B1 90

Please ask your system administrator for the axis names which are programmed in your specific control unit.

**Example:**

N ... G60 X10 Y10 B1 35 1traversing movement X,Y: feed axes B1: auxiliary axis

Within the NC program, instructions may be supplemented by **special functions**. Some examples of important address letters representing special functions:

- **F** Feedrate
- **S** Spindle speed
- **M** M functions (e.g. gear-stage selection, direction of spindle rotation)
- **T** T word (tool selection)

For details, please refer to section 6, “Auxiliary and special functions”.
Example: Positional data with special functions

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>G01</td>
<td>path command instruction</td>
</tr>
<tr>
<td>X40</td>
<td>special functions</td>
</tr>
<tr>
<td>Y50</td>
<td>: path information</td>
</tr>
<tr>
<td>F250</td>
<td>Traverse at feedrate to X40, Y55 at programmed F value (feedrate) and S value (speed) with clockwise rotating spindle and make tool T05 available in tool magazine.</td>
</tr>
<tr>
<td>S500</td>
<td>T05</td>
</tr>
<tr>
<td>M03</td>
<td></td>
</tr>
</tbody>
</table>

2.1.2 Program design elements

Block numbers

You may identify each NC block by a block number. This improves the program’s readability. DIN block numbers are always on the left at the beginning of a program line and consist of the “N” address letter and a number following directly behind (example: “N10 ...”).

You should program the block numbers in ascending order and with an increment width of 10 (N10 ...; N20 ...; N30 ... etc.). This way, you can insert additional program lines between two blocks in the case of program changes without impairing the readability of the program.

If you wish to use branching instructions or jump markers containing block numbers as parameters, you must identify the target blocks with block numbers. Likewise, block numbers are required in subprograms and cycles.

Comments

Comments are used to provide program parts with explanations or to document them. Well-commented programs facilitate and accelerate subsequent familiarization for other programmers, e.g. if the program needs to be changed. However, each comment character will increase the program file by 1 byte.

Comment text is usually put between parentheses, e.g. “(...)” or preceded by a semicolon “;” set before the comment text. The Typ3 osa will ignore text between the brackets.

Example: Comment text

N50 (Pocket machining)

or

N50 ; Pocket machining

Notes

If you program notes, they are used to display text on the CNC screen during the execution of the program. You can use such notes to inform the operating staff about the current status of the program, or to give them instructions for action.
There are two kinds of notes:

- **Channel-specific notes**
  
  **Syntax variants:** (MSG ...), (*MSG ...), (MSG, ...), (*MSG, ...), MSG (...)
  
  They are displayed in the MSG window of the “Automatic” operation mode of the call channel. Additionally, these notes are displayed under “Messages” in the info dialog. They are deleted when the program is cancelled or the control is reset.

- **Cross-channel notes**

  **Syntax variants:** (GMSG ...), (GMSG, ...)
  
  These notes are displayed under the channel-independent notes in the info dialog. They are deleted when the control is reset.

A programmed note may have up to 80 characters.

For an instruction to take action, you would program, e.g., an "M0" in the very same line or in the following one. This ensures that the execution of the program will only be resumed after the note has been acknowledged by "NC Start".

**Example:** Note text

N60  (MSG Measure workpiece!)
N70  M0

**Running of the program**

In the absence of instructions relating to the program flow, the program blocks will be processed one by one. However, you have the following options of influencing the running of the program:

- Subprogram calls (please refer to sections 2.1.3 and 6.3.1)
- Repeat instructions (please refer to the CPL manual)
- Jump instructions (please refer to the CPL manual)
- ‘Skip block’ instruction

**Skip block**

You can mark program blocks in such a manner that the control unit will simply ignore these blocks if the input signal "I4.4 Skip block" is active. To do so, you program the "/" sign at the beginning of the program line.

**Example:**

I4.4 is active: N30 block will be ignored
I4.4 is not active: N30 block will be processed.

**Channel designation**

A program may contain a channel designation.

Syntax: $<channel number>

If a program containing a channel designation is started on another channel, a runtime error will occur.

**Example:**

N10  $2  The following program can be executed on channel 2 only.
N20  G.. X.. Y..  Program instructions on channel 2
N30  $1  The following program can be executed on channel 1 only.
N40  G.. X.. Y..  Program instructions on channel 1
...
### 2.1.3 Subprograms

If you need to repeat a specific processing operation within a program, it is recommended that you write this program part in the form of a subprogram and call it whenever it is needed.

This will save programming code and memory space. In addition, your programs will become clearer and easier to maintain.

#### Subprogram call with P address

You call subprograms via the P address in the form of "P<SP name> DIN".

**Explanation:**

- `<SP name>`: The name of the subprogram.
- `DIN`: Optional parameter. Prevents the subprogram from being linked. You should state this parameter only if the entire SP consists of DIN blocks and does not call any other SP's. If this is not the case (e.g. if the SP contains CPL blocks), an error message will appear while the program is running. For further information, please refer to the "CALL command" of the "CPL programming instructions".

Traversing movements which are programmed in one line will be executed prior to the subprogram call.

(e.g. "N40 PTest1 X10 Y10 Z0").

The subprogram is executed unconditionally.

A subprogram can call further subprograms (nesting):

**Example:** Subprogram call

\[ \begin{align*}
\text{N...} \\
\text{N40 PBohrbild1} & \quad \text{"Bohrbild1" (drilling graph 1) is called and executed once.} \\
\text{N50...} & \quad \text{Subsequently, the calling program will be further executed by the N50 block.} \\
\text{N...} \\
\end{align*} \]

**Example:** Nesting of subprograms

```
<table>
<thead>
<tr>
<th></th>
<th>P1</th>
<th>N1</th>
<th>P5</th>
<th>N1</th>
<th>P2</th>
<th>N1</th>
<th>P7</th>
<th>N1</th>
<th>P8</th>
</tr>
</thead>
<tbody>
<tr>
<td>N9</td>
<td>N10</td>
<td>N11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

P1: Main program
P5, P2, P7, P8: Subprograms
Nesting is possible to a depth of 9 (incl. main program), i.e. with complete nesting, the main program can open a maximum of 8 subprograms.

Subprograms can also be called via G and M addresses (please refer to section 2.1.4, G addresses, or to section 6.3.1, M addresses).

Subprogram call without a P address

Subprograms can also be called directly without a P address. In this case, it is sufficient to state the name of the subprogram.

The statement

<SP name> stands for the subprogram name.

In both cases, the syntax is the same:

N40 XSP  subprogram call without a P address
and
N40 PXSP  subprogram call with a P address

Example: Subprogram call

N...
N40 XSP  Subprogram "XSP" is called and executed.
N50...  Subsequently, the calling program will be further executed by the N50 block.
N...

Any confusion with normal syntax must be avoided!

N30  G1 F100  Linear interpolation G1 is called
N40  X0SP  \(\rightarrow\) results in a syntax error
X0 is interpreted as the '0' coordinate in the X-axis to which the X-axis is to traverse, i.e. a subprogram called "X0SP" will not even be recognized by the interpreter.

Always give your subprograms unmistakable names to avoid misinterpretations by the control interpreter.

Subprogram call via nonmodal G instructions

Besides using various M-functions (see page 6–4) or the P address (see page 2–7), subprograms can also be called via 16 nonmodal G instructions.

You can use the MACODA to determine both the G instructions and the programs to be called via these G instructions. With each call, the respective subprogram is executed once.
The assignment of G instructions to program names is specific for each application and can be programmed in machine parameters 309000001 and 309000002. Please ask your system administrator which G instructions have been programmed specifically for your machine to call subprograms.

Programming

Each block may contain no more than one subprogram call via a P, G or M instruction. If an address letter (e.g. G or M) occurs repeatedly in a block, the address calling the subprogram must be programmed at the end of the line.

Example: Subprogram call via Gxx
N... G0 X20 Y30 Z50 Gxx...

In addition to these 16 nonmodal G instructions, another 16 modal G instructions can be programmed to call subprograms in machine parameters 309000005, 309000006 and 309000007. The Typ3 osa will continue to execute the subprograms thus called in every program block until the modal action is canceled by an explicit command to this effect. This feature is very useful for drilling cycle applications because you just need to traverse to a new drill position. After traversing to the new position, the hole is then drilled automatically by the subprogram.
2.1.4 Jump destinations and jump instructions

As a rule, main program and subprogram blocks and cycles are executed in the same order as they were programmed.

The order of execution can be changed by program jumps.

The following instructions are available:

- **Jump destinations (LABELS)**: Stating jump destinations with user-defined names.
- **Jump destination (G23, G24)**: Jump destinations dependent on an interface signal with a block number stated.
- **Jump instructions (GOTOF and GOTOB)**: Allow branching from any point in the program to a jump destination. Program execution continues immediately upon arrival at the jump destination.

### Jump destination

**Destination labelling (LABEL)** within a program:
User-defined branches in a program can be programmed by defining jump destinations.

- Label names with a minimum of 2 and a maximum of 32 characters (letters, digits, underline) are assigned.
- The first two characters must be letters or underlines.
- The label name must always be followed by a colon.
- Labels are always written at the beginning of an NC block, directly after the block number.
- Jump destinations are addressed by means of jump instructions (GOTOF and GOTOB).

### Jump instruction

**Jump instruction (GOTOF)** with a **forward** jump destination (towards the end of the program):
- must be programmed in a separate block.
- is programmed in combination with a LABEL.

**Jump instruction (GOTOB)** with a **backward** jump destination (towards the beginning of the program):
- must be programmed in a separate block.
- is programmed in combination with a LABEL.

**Examples**: Label, GOTOF, GOTOB

N100 GOTOF TO_PART2 Jump forward to jump destination
N110.. “TO PART2”
N120..
N130 TO_PART1: Definition of jump destination “TO PART1”
N140..
N150 GOTOB TO_PART1 Jump forward to jump destination “TO PART1”
N160..
N170 TO_PART2: Definition of jump destination “TO PART2”
The jump destination (G24) is a block number and is executed unconditionally. The jump destination is defined as an L-address with a block number.

If an "unconditional jump" is programmed incorrectly, an endless loop may occur.

G24 L <block number>
with
<brick number> = 15 digits, with an optional ".".

Please note for G24 L...:
- Jumps must never be programmed together with any other instructions in the same block.
- The syntax in the statement of the L-address must be identical with the jump destination (N-word) (also in the case of preceding zeros).

Example:
N020  G1 X200 Y300 F500
...
N500 G24  L20         Wrong!
N500 G24  L020         Correct!

- You can only jump to DIN blocks. No CPL blocks may be used for the L-address.

Conditional jump

The jump destination (G23) is conditioned by the status of the interface signal "CONDITIONAL JUMP". The interface signal is scanned while the G23 block is being prepared.

Any interface signals between block preparation mode and block execution mode will be ignored!
Unless this can be ensured, block preparation must be interrupted by programming a WAIT instruction.

The jump destination is defined as an L-address with a block number.

G23 L <block number>
with
<brick number> = 15 digits, with an optional ".".

Please note for G23 L...:
- Jumps must never be programmed together with any other instructions in the same block.
- The syntax in the statement of the L-address must be identical with the jump destination (N-word) (also in the case of preceding zeros).
- Only DIN blocks can be jumped to. Blocks written in CPL may not be used as an L-address.
Example:

N68  X–250  Y20  Jump destination
...
N100 X100 Y200 Z50
N101 X0  Y0  Z10
102 WAIT  Waiting for an IF signal, block preparation interrupted.
N103 G23 L68  Jump to N68 is executed if an IF condition is fulfilled.
N104 X200 Y–300 ...

In CPL block 102, the programmed WAIT instruction ensures that any signal changes are recognized by the NC immediately before N103 processing.
2.1.5 Typ3 osa standard programming formats

The standard format applies to metric input in terms of “mm” and a measuring-system resolution of 0.0001 mm.

<table>
<thead>
<tr>
<th>Addresses</th>
<th>Preparatory functions</th>
<th>Format</th>
<th>Meaning</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>variable, e.g.</td>
<td></td>
<td>real</td>
<td>Positional data:</td>
<td></td>
</tr>
<tr>
<td>X,Y,Z,C</td>
<td>e.g.: G1, G2</td>
<td></td>
<td>cartesian axis position</td>
<td>mm or degree</td>
</tr>
<tr>
<td>X(p1,p2,p3,p4)</td>
<td>e.g.: G581</td>
<td>real</td>
<td>positional data incl. parameter list</td>
<td>mm or degree</td>
</tr>
<tr>
<td>X = AC(50)</td>
<td>e.g.: AC(..)</td>
<td>real</td>
<td></td>
<td></td>
</tr>
<tr>
<td>I,J,K,R</td>
<td>G2, G3</td>
<td>real</td>
<td>circle parameter</td>
<td>mm</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>circle radius</td>
<td>mm</td>
</tr>
<tr>
<td>D</td>
<td>G41/G42</td>
<td>int</td>
<td>Technological information</td>
<td>comp. no.</td>
</tr>
<tr>
<td>F</td>
<td>G94</td>
<td>real</td>
<td>cutter-radius compensation</td>
<td>mm/ min</td>
</tr>
<tr>
<td>F</td>
<td>G4</td>
<td>real</td>
<td>feedrate (sync. axis)</td>
<td>sec</td>
</tr>
<tr>
<td>FA</td>
<td></td>
<td>real</td>
<td>dwell time (&quot;)</td>
<td>mm/ min</td>
</tr>
<tr>
<td>H</td>
<td></td>
<td>int</td>
<td>feedrate (auxil. axis)</td>
<td>comp. no.</td>
</tr>
<tr>
<td>S</td>
<td></td>
<td>real</td>
<td>tool-length compensation</td>
<td>rpm</td>
</tr>
<tr>
<td>T</td>
<td></td>
<td>real</td>
<td>spindle speed</td>
<td>tool no.</td>
</tr>
<tr>
<td>N (block no.)</td>
<td>int</td>
<td></td>
<td>N1, N2, N3 etc. block address</td>
<td></td>
</tr>
<tr>
<td>P,K,V</td>
<td>str</td>
<td></td>
<td>program, compensation, zero-shift address</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>int</td>
<td></td>
<td>G function</td>
<td></td>
</tr>
<tr>
<td>M</td>
<td>int</td>
<td></td>
<td>special machine function</td>
<td></td>
</tr>
</tbody>
</table>

Meaning of the values in the “Format” column:
- int: Digit string consisting of 9 digits max., without decimal point
- real: Digit string. consisting of 15 digits max., with decimal point
- str: Character string

Auxiliary functions (e.g. F, FA, S, etc.) can be bit- or bcd-coded (see Section 6).
Notes:
3 Axes and coordinate systems

3.1 Physical and logical axes

The aggregate of all drives of a machine is made up of axes and spindles. The term of axes refers in this context to the entirety of the drives controlling the nominal positional values. Spindles, in contrast, use a speed interface. The term of spindles/C axes refers to spindles which can be selected both in spindle operation (speed interface) as well as in C-axis operation (positional interface). The specific type of drive is determined in the MACODA parameter 100100001, "Drive function type".

The terminology of the Typ3 osa makes a distinction between physical and logical axes.

Physical axes

Physical axes are also called system axes. Unless its display is suppressed by machine parameter settings, each physical axis corresponds to a real SERCOS axis. Its axis/drive index (1...n) is unique throughout the system and corresponds to the index of the respective individual drive or axis parameters in the MACODA (function area 10, machine). The "physical axis designation" MACODA parameter 100300001 assigns a unique, system-wide name to each physical axis.

The "Channel assignment" MACODA parameter 100300002 can be used to accurately assign each physical axis to a machining channel. On one channel, all the axes which are in one and the same machining-technological relationship are grouped. The aggregate of all axes of one channel is referred to as an axis group. Considering that different axis groups are assigned to different channels, it is possible for several axis groups to perform various machining operations independently of each other and in parallel. In this context, the channel can only select the axis group which is assigned to it.

Physical axes which are not assigned to a specific channel are referred to as asynchronous axes or auxiliary axes. These can be selected from any channel, but are in no interpolary relationship to other axes. Auxiliary axes may, for instance, be used to drive tool or pallet changers.

Logical axes

The axes of an axis group are referred to as the logical axes of the channel. Since there is an interpolary relationship between the axes of a specific channel, they are also referred to as synchronous axes. On one channel, all the axes assigned to it are numbered on the basis of the logical axis index. The logical axis index results from the order of the physical axis indices, i.e. on one channel the axis with the lowest physical axis index will receive the logical axis index of 1. Consequently, the axis with the highest physical axis index will receive the highest logical axis index.

When a new axis is transferred to a channel, the logical axis indices, including the physical axis index of the new axis, are resorted in accordance with the new order of physical axis indices of all existing axes.

MACODA parameter 701000010, "Logical axis designation", allows to assign a logical axis name to every logical axis on a channel. Every logical axis name must be unique on the respective channel. An axis can be selected both under its logical and under its physical axis name. In the case of identical names, the logical name always takes precedence over the physical name!
Logical axes can be selected from no other channel than from the one they have been assigned to via MACODA or a G-function (please refer to ‘axis group’ above).

The assignment of axes to channels as set in the MACODA can be changed using the “Axis transfer” function (G510–G516) in the part program.

**Example of an axis configuration:**
The logical axis names for each channel are taken from MACODA parameter 701000010!

<table>
<thead>
<tr>
<th>physical axis/drive index = MACODA parameter index</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>physical axis name (1003 0000 1)</td>
<td>X1</td>
<td>Y1</td>
<td>X2</td>
<td>Y2</td>
<td>–</td>
<td>Z1</td>
<td>Z2</td>
<td>E</td>
</tr>
<tr>
<td>channel assignment (1003 0000 2)</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>–</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>drive function type (1001 0000 1)</td>
<td>axis</td>
<td>axis</td>
<td>axis</td>
<td>axis</td>
<td>spindle</td>
<td>axis</td>
<td>axis</td>
<td>axis</td>
</tr>
</tbody>
</table>

| logical axis index channel 1 (701000010)          | 1 | 2 | – | – | – | 3 | – | – |
| logical axis name channel 1                        | X | Y | – | – | – | Z | – | – |

| logical axis index channel 2 (701000010)          | – | – | 1 | 2 | – | – | 3 | – |
| logical axis name channel 2                        | – | – | X | Y | – | – | Z | – |

<table>
<thead>
<tr>
<th>axis type</th>
<th>sync.</th>
<th>sync.</th>
<th>sync.</th>
<th>sync.</th>
<th>sync.</th>
<th>sync.</th>
<th>sync.</th>
<th>async.</th>
</tr>
</thead>
<tbody>
<tr>
<td>axis</td>
<td>axis</td>
<td>axis</td>
<td>axis</td>
<td>axis</td>
<td>axis</td>
<td>axis</td>
<td>axis</td>
<td>axis</td>
</tr>
</tbody>
</table>

### 3.2 Axis and working range coordinates

The basic function of a program-controlled machine consists in interpolating the predefined – frequently Cartesian – movement requirements \((p_1, p_2, ... p_n)\) and transforming them into physical axis values \((q_1, q_2, ... q_n)\) in such a manner that the resulting relative movement between tool and workpiece meets the specified requirements.

In this context, the resulting transformation chain may show very different degrees of complexity depending on the machine kinematics and the active coordinate transformations.

In the simplest case, the \(p_i\) positions correspond exactly to the \(q_i\) axis values and are shifted by an offset: \(p_i = q_i + \text{offset}\).

In the most complex case, there are transformational relationships for all coordinate axes with regard to machine axes. This may be the case even with a simple machine tool, for instance, if the “inclined plane” function is used to define a program coordinate system having an arbitrary orientation within the working area of the machine.

It should be explained at this point that the term of Cartesian coordinate system as used herein mainly refers to so-called generalized Cartesian coordinates, since there is a total of 6 degrees of geometric freedom \((p_1,...p_6)\), namely 3 translatory degrees of freedom in addition to the 3 rotary degrees.
In the case of a machine tool, one has to distinguish between 3 typical types (of position settings):

- **1st case:** $P = (q_1, q_2, q_3, ..., q_n)$
  There is no transformational relationship between the machine and the axis coordinates. Programming consists in the direct programming of axis coordinates.

- **2nd case:** $P = (p_1, p_2, p_3, q_4, ..., q_n)$
  There is only a transformational relationship as regards "position-creating" coordinates. These are programmed by entering the working range coordinates. For the other coordinates, programming consists again in the direct programming of the physical axes (axis coordinates).

- **3rd case:** $P = (p_1, p_2, p_3, p_4, p_5, p_6)$
  There is a transformational relationship between machine and axis coordinates for all coordinates. Programming thus becomes independent of the actual machine kinematics. All coordinates are programmed by entering the working range coordinates.

In order to emphasize the separation between these three types of coordinates in programming more clearly, it may actually make sense to use different blocks of logical axis addresses on one channel, depending on whether axis coordinates or working range coordinates are referred to. This may be necessary simply because the number of existing programming coordinates does not necessarily correspond to the number of the physical axes in each case.

**Example: 5-axis milling machine**

- **1st case:** Direct programming of the physical axis values:
  $P = (q_1, q_2, q_3, q_4, q_5) = (X_A, Y_A, Z_A, A_A, B_A)$

- **2nd case:** Programming of working range coordinates with any orientation to define the position of the workpiece (e.g. by activating an inclined plane), and direct programming of the physical axis values to define the orientation of the workpiece with regard to the tool:
  $P = (p_1, p_2, p_3, q_4, q_5) = (X_R, Y_R, Z_R, A_R, B_R)$

- **3rd case:** Programming of working range coordinates with any orientation to define position and orientation:
  $P = (p_1, p_2, p_3, p_4, p_5, p_6) = (X_R, Y_R, Z_R, A_A, B_A)$

In the 3rd case, the corresponding axis values $(q_1, q_2, q_3, q_4, q_5) = (X_A, Y_A, Z_A, A_A, B_A)$ are calculated within the control unit by machine-specific axis transformation.
3.3 Axis and coordinate designations

DIN 66217 contains the definition of the arrangement and designation of the coordinate axes of the machine coordinate system:
The system used is a right-handed rectangular coordinate system with the linear X, Y and Z axes, each of which being assigned the rotary A, B and C axes.

Machine coordinate system

If the machine is equipped with a minimum of one spindle, the Z axis is parallel with the main spindle, otherwise it is perpendicular to the clamping surface.

Positive axis direction corresponds to the direction of the workpiece relative to the tool.

The X axis is parallel with and horizontal to the workpiece clamping surface. In analogy to the above coordinate system, the arrangement of the Y axis results from X and Z.

The X, Y and Z axes are referred to as main axes.

Further, independently controlled axes arranged in parallel with the main axis are referred to as U, V, W. Rotational movements carried out by machine components around the main axis (e.g. rotary axis) are referred to as A, B and C.

The sense of rotation is mathematically positive if one looks from one coordinate vertex in the direction of the coordinate origin and the sense of rotation is counter-clockwise (positive sense of rotation).
If further linear axes exist, which are parallel with the above X, Y, Z main axes, they are referred to as U, V and W, and if there are even more axes of this description, they are called P, Q and R.

Non-parallel linear axes can be designated freely as U, V, W, P, Q and R.

In the case of machines with various positioning or parallel-motion options, these movements can be indexed using a letter and a number (e.g. X1). The index has to be an integer and greater than zero. The main movements can be performed with or without an index. This way, it is possible to have indexed and non-indexed movements on one and the same machine.

If there are further rotary movements in addition to the rotary A, B and C main movements, they are called D or E (address letter must be available). Indexing can be used for rotary axes, too.

Certain address letters (e.g. G, M and F) cannot be used for movements.
3.4 Coordinate systems

In order to be able to run machining programs without changes at any point and with different tool dimensions, several coordinate systems are defined, some of which are permanently stored and machine-independent, while most are freely definable and may vary with the machine employed. The transition from one coordinate system to another is done using so-called coordinate transformation.

3.4.1 ACS axes coordinate system

The entirety of all the axes assigned to a specific channel forms the so-called axes coordinate system (ACS). If you specify a movement in the direction of a coordinate of the axes coordinate system, this movement can always be realized by traversing one real (physical) axis.

3.4.2 MCS machine coordinate system

Considering that the axes coordinate system depends on the kinematics of the machine, hence on the machine type, it is only of minor significance to the specification of movements for workpiece machining by part programs. Therefore, a so-called machine coordinate system (MCS) is assigned to each channel, which is mainly Cartesian and thus independent of the machine kinematics. The machine coordinate systems may be defined identically for various axis groups (channels).

Zero-point designation and symbol: M Machine zero point

The relationship between the machine coordinate system and the related axes coordinate system is represented by the so-called axis transformation which is also referred to as reverse transformation.
Examples of the relationship between the machine coordinate system and the related axes coordinate system:

- **R reference point**

  The reference point is required to establish the dimensional reference between a machine axis equipped with an incremental measuring system and the control unit. In the case of axes of this type, the reference point must always be approached upon power-up of the machine or after a power failure, respectively.

  This is not necessary in the case of axes with an absolute measuring system.

  In most cases, the reference point of an axis is located within its marginal area and can be approached automatically. The reference positions are defined axis by axis in the axes coordinate system. The reference point R represents the position within the machine coordinate system which results if all the axes are in their reference position.
3.4.3 WCS workpiece coordinate system

Zero-point designation and symbol: \( W \) workpiece zero point

The workpiece coordinate system (WCS) is a coordinate system which is freely set off and rotated in relation to the machine coordinate system.

Only with the help of the workpiece coordinate system is it possible to run a part program without changes at any position within the working area of the machine.

It is possible to define several different \( W_i \) workpiece coordinate systems, which are all related to each other and act additively. This enables a part program to be built up in analogy to the dimensioning of a constructional drawing. Whereas, for instance, a tool coordinate system describes a \( W_1 \) local zero point as a function of the absolute workpiece zero point, another tool coordinate system represents the relationship between the absolute \( W_0 \) workpiece zero point and the \( M \) machine zero point. Additional workpiece coordinate systems may establish the relationship between 2 local workpiece zero points (\( W_0 \leftrightarrow W_1, W_1 \leftrightarrow W_2 \)).

3.4.4 PCS program coordinate system

Zero-point designation and symbol: \( P \) program zero point

The workpiece coordinate system with the highest index (\( W_i \) with \( i=\text{max} \)) is referred to as program coordinate system (PCS).

The “P” program coordinate system is the point which all programmed coordinate values of the program refer to. According to the definition of the workpiece coordinate system, the program coordinate system may be freely set off and rotated with regard to any \( W_i \) workpiece zero point.

Relationship between the machine, workpiece and program coordinate systems
3.4.5 TCS tool coordinate system

All the previously explained coordinate systems describe the relationship between the workpiece and its contour and the machine zero point. In contrast, the tool coordinate system (TCS) establishes the position and orientation of the tool with regard to the machine coordinate system.

In this context, the rotation of the tool in relation to the machine coordinates is of special interest.

In general terms, the tool orientation can be described:

- in absolute terms, in relation to the machine coordinate system, or
- in relative terms, in relation to any local coordinate system (e.g. \( W_i \)), or
- in relation to a coordinate system which is carried along with the contour.

- Zero-point designation and symbol: \( T \) tool reference point
- The dimensions of the tools, for instance in the case of a 3-axes machine, are related to a fixed tool setting point (e.g. tool clamping point, tool zero point).
3.5 NC functions for defining coordinate systems

3.5.1 Axis transformation: Axis coordinates and machine coordinates

Considering that the kinematic structures of machines are very different, axis transformation in the Typ3 osa control unit is done according to customer specification.

3.5.2 Coordinate transformation: Machine coordinates, workpiece coordinates and program coordinates

Machine coordinates

Absolute position data normally refer to machine coordinates and thus to the machine datum M.

For practical purposes, all dimensions and paths to be traveled as stated in the part program refer to program zero P or to the workpiece datum W. This way you “unlink” or keep your part programs detached from your absolute machine coordinates. Owing to the programmed shifts, you can execute every program at any point within the machine working range without having to change any dimensional data in the program.

Every shift or rotation has a cumulative effect, i.e. the data is added to the machine coordinates. Both positive and negative shifts or rotational values are permitted. The directions of the coordinates in the workpiece coordinate system are designated using the logical (physical) axis addresses.

If no shifts have been programmed, all part program values are interpreted as being machine coordinates!

Workpiece coordinates

- The following instructions are available in the Typ3 osa for workpiece zero shifts:
  - G53, G54 .. G59 Zero shift (ZS)
  - G153, G154 .. G159 “1st additive ZS”
  - G253, G254 .. G259 “2nd additive ZS”
  - G160, G260, G360, G167 “External zero shift”

- The position of a workpiece can be corrected by shifting the workpiece zero (X/Y, X/Z, Y/Z planes) including a rotation of the workpiece coordinate system (X/Y plane) with the following instructions:
  - G138, G139 “Workpiece position compensation (correction)”

- In order to shift the workpiece zero and/or rotate the workpiece coordinate system on various planes (on the X/Y, X/Z and Y/Z planes), use the following instructions:
  - G353, G354..G359 ”Inclined plane”
  - G453, G454..G459 “1st additive inclined plane” (in planning)
  - G553, G554..G559 “2nd additive plane” (in planning)

Program coordinates

The last coordinate system in the series of workpiece coordinate systems is also referred to as program coordinate system (see also section 3.4.4).

- When shifting the program zero, you can shift the position of the program coordinate system in relation to the workpiece zero coordinate system by using the following instructions:
  - G169, G168 “Program coordinate shift”
  - G269, G268 “Additive program coordinate shift”
Example: Effects of the individual functions

1: ZS (e.g. G54)
2: external ZS (G160; via PLC)
3: 1st additive ZS (e.g. G154)
4: 2nd additive ZS (e.g. G254)
5: workpiece position compensation WPC (G138; e.g. shift incl. rotation)
6: program coordinate shift G168
7: additive program coordinate shift G268
8: workpiece position compensation WPC (G138; e.g. shift incl. rotation)
9: inclined plane (e.g. G354)
10: program coordinate shift G168
11: additive program coordinate shift G268

You should position the workpiece zero point in such a manner that the dimensions of the production drawing can be easily translated into coordinates or, resp., imported.
Notes on coordinate systems

- In order to shift the workpiece zero, you can either use the "Inclined plane" function or any of the other functions available for shifting zero points (see above: G53...G360). However, calling the "Inclined plane" function simultaneously with any other zero shift function is not permitted.

- Within the Typ3 osa, free three-dimensional positioning and orientation of the program coordinate system, P, in relation to the workpiece coordinate system, W, will not be available with the two additive groups for the "Inclined plane" function (G453...G459 and G553...G559) until a future software update.

- If the program coordinate system and the machine coordinate system have the same orientation, all zero shift functions described in this section may be used.

Therefore, any reference in the following sections of this manual concerning a program coordinate system or a program zero also applies to workpiece coordinate systems and workpiece zeros.

Activating shift values

Depending on the various G-functions, the values of the various kinds of shifts are activated in different ways:

- "Zero shift tables": "ZS", "1st additive ZS" and "2nd additive ZS"

  You use the zero-shift tables to store the "distance" between the "M" machine zero point and the "P" program zero point or "W" workpiece zero point for each axis (On the recording of this "distance", please see the following figure.). If the corresponding ZS is then activated, the control unit will automatically add the stored "distance" to each programmed absolute axis position.

  Zero shift tables in the form of ASCII files are available in the file system of the Typ3 osa. They are activated for each channel using G22.

  For details about editing such tables, please refer to the Typ3 osa operating instructions manual.


- "External zero shift"

  The shift values are set for each axis by the PLC. For further details regarding programming, please refer to G160, G167, G260 and G360.
• "Workpiece position compensation (WPC)":
The shift values for individual logical axes and the rotation of the X/Y plane are programmed directly in the path command of the G138 G instruction. The shift values can be recorded by means of an edge sensor, e.g. For further details, please refer to "G138".

• "Inclined plane":
The shift values for individual logical axes and the unrestricted three-dimensional rotation are stored in an ASCII file with any name, which is addressed using the ID syntax. The values are activated by G22. Input of three-dimensional orientation is possible via programming systems, e.g. For further details, please refer to "G353–G359".

• "Shifting the program start":
The shift values for individual logical axes (on the X/Y, X/Z, Y/Z planes) are to be programmed directly in the path command of G instruction G168 and/or G268. For further details, please refer to "G168 and G268".

Example: Zero shift recording

First, the required zero shift values are determined when setting up the machine (see next fig.). The shift values are independent of the tool lengths.

- In the X and Y direction using edge or centering callipers
- In the Z direction using a stop gauge, dial gauge with tripod, a scribing block or a barrel gauge.

Subsequently, you enter the values so determined in the zero offset table activated.
3.5.3 Functions for manipulating the programmed contour

Within the programming system, the Typ3 osa is able to manipulate a programmed contour as follows:

- **shift** (G60 – programmed contour shift) and/or
- **mirror, scale and rotate** the contour around an axis parallel to the coordinate axes of the program coordinate system (G37, G38).

Please note: Using the G37, G38 and G60 instructions, you cannot influence the program coordinate system itself, but only change the position, orientation and scaling of the programmed contour in relation to the program coordinate system.

In contrast to the other zero shifts, a G60 shift will also be influenced (e.g. scaled: see fig. below) by G37 and G38. For further details, please refer to the corresponding functions.
3.5.4 Functions for tool compensation

Tools

Within the NC part program, the tools are addressed by the **T address letter** and a number (example: “T9”; the tool with number 9).

A complete tool consists of:

- the tool
- the tool chuck

When machining a workpiece, the tool tip or cutting edge is supposed to be controlled as accurately as possible on the desired machining path along the workpiece.

Considering the differences in the shapes and **dimensions of tools**, the exact dimensions of all the tools being used must be determined and entered in the control unit once, prior to machining. Only then can the control unit calculate a course of path resulting in the programmed workpiece contour, irrespective of the tool being used.

When a tool is clamped in the spindle and the corresponding “tool compensation” is activated, the CNC can take the relevant tool data into account automatically.

For further details about the required indispensable data, please refer to the next section.

Tool compensations D and H

When executing the program, the control unit has to take the **actual dimensions** of the currently active machining tool into account.

The tool compensation data is activated by the following **address letters**:

- H: length
- D: radius
The data of tool length compensation and tool radius compensation is evaluated by the control as follows:

**Length compensation**

**Method 1:** Length compensation, referring to the “SPN” spindle nose.

- The effective length of a tool (distance tool tip – spindle nose) corresponds to the compensation value. Only positive values are used.

![Diagram showing method 1](image)

- Input in the length compensation sheet of the tool compensation table:

  For tool:
  - T01: $H_1 = 70.8320$
  - T02: $H_2 = 81.7120$
  - T03: $H_3 = 100.0030$

**Method 2:** Length compensation, referring to a zero tool.

- You define a tool of the tool set to be the zero tool. In this case the tool tip of the zero tool represents the point of reference for setting up (WSN). The zero tool (in the example **T02**) is assigned the length-compensation value “0”.

  In the case of longer tools the length-compensation value corresponds to the distance between WSN and tool tip with a positive sign.

  In the case of shorter tools the length-compensation value corresponds to the distance between WSN and tool tip with a negative sign.
• Input in the length compensation sheet of the tool compensation table:

For tool:
T01: \( H1 = -20.813 \)
T02: \( H2 = 0 \)
T03: \( H3 = 25.821 \)

For further details, please refer to the "Tool compensation" section under "Length compensation".

Radius compensation

In order to obtain the programmed workpiece contour, the Typ3 osa guides the cutter on a "path parallel to the programmed contour" (equidistant). The distance between the programmed contour and this equidistant is in each machining position equal to the actual cutter radius. This path is also referred to as "cutter center path".

• The compensation value is equal to the cutter radius.

• Entry on the radius-compensation sheet of the geometry-compensation table:

e.g. for tool:
T01: \( D1 = 14 \) (with a cutter diameter of 28 mm)
T02: \( D2 = 22 \) (with a cutter diameter of 44 mm)

For further details, please refer to the G 40, G41 and G42 G instructions in the "Tool compensation" section under "Radius compensation".

Direction of compensation

In the case of the Typ3 osa, tool compensation is carried out along the axes of the tool coordinate system or the program coordinate system! Most machines do not allow setting the orientation with reference to the machine coordinate system (e.g. three-axis machining). Therefore, length compensation on those machines is done only along the axes of the machine coordinate system (H compensation, e.g., in the direction of the \( Z_m \) axis).

Activating internal compensation values

For a radius or length compensation to be taken into account, a tool compensation table containing the compensation values must first have been activated. The tool compensation tables are stored as ASCII files with any name in the file system of the Typ3 osa. These files are activated with G22K{<path>}{<name>} for each channel.

"Editing compensation tables" is described in the Typ3 osa Operating Manual.

External tool compensation

The tool compensation data (e.g. multiple compensation for combination tools) are entered via the PLC and activated with G145 – G845 (see also section 4.57):

• \( R_{ext} \): radius compensation
• \( H_{ext} \): length compensation

This compensation data has an additive effect on any other length or radius compensation that may be active.
General tool compensation

General tool compensation is activated with **G147 – G847**. Tool compensation data may consist of compensation sets of various length compensations to carry out a three-dimensional tool offset, e.g., or of parallel length compensations for operations with several different drilling tools (please refer also to section 4.58):

- **L1, L2 and/or L3** : length compensation values and/or offset values
- **R** : radius compensation
- **SL** : cutting edge position

Compensation data has an additive effect and/or is independent of any other compensation values, or it may overwrite other active length or radius compensations, as the case may be.
Overview:
Methods of tool compensation

Apart from the H and D standard compensations, there are other methods of tool compensation. They are activated in different ways, depending on the respective G functions:

<table>
<thead>
<tr>
<th>Compensation method</th>
<th>Save under</th>
<th>Activate via</th>
<th>Programming (Example)</th>
<th>Notes</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard length compensation radius compensation</td>
<td>Length: H1 – H48 Radius: D1 – D48</td>
<td>G22 K{&lt;path&gt;}:&lt;file name&gt; and H.. D.. (G41, G42)</td>
<td>G22 KGeo1 .. .. G1 Z.. H.. G41 X.. Y..D..</td>
<td>Length and radius compensation values are activated using H or D addresses in the compensation table K&lt;file name&gt;, which is activated with G22. For deactivation, use H0, D0.</td>
<td>The effect of a radius compensation always comes to bear on the working plane selected in the program coordinate system.</td>
</tr>
<tr>
<td>External tool compensation (1st external tool compensation)</td>
<td>Length: Hext. Radius: Rext.</td>
<td>G145 – G845</td>
<td>G145</td>
<td>For each PLC, up to 8 external length and radius paired values can be activated. For deactivation, use G146.</td>
<td>Additive on – standard length and radius compensation</td>
</tr>
<tr>
<td>General tool compensation (2nd external tool compensation)</td>
<td>Length: L1, L2 and/or L3 Radius: R Cutting edge position: SL</td>
<td>G147 – G847</td>
<td>G147</td>
<td>For each PLC, up to 8 external length and radius compensation data blocks can be activated. For deactivation use G148.</td>
<td>Independent of – standard length compensation – external length compensation Additive on – standard radius compensation External radius compensations will be overwritten!</td>
</tr>
<tr>
<td>Drill axis compensation of – 1st compensation group – 2nd compensation group</td>
<td>Length: H or Hext. L1, L2 and/or L3</td>
<td>G78</td>
<td>G78 X22</td>
<td>The drill axis (the axis on which the length compensation takes effect) is switched using G78.</td>
<td>Length compensation values refer to the axes to which the compensation group relates.</td>
</tr>
</tbody>
</table>

For information on how to program and activate tool compensations, see the section on "Tool compensation".
3.5.5 Tool movements (interpolation types)

**Linear interpolation, straight-line interpolation**

The tool can approach each point within the three-dimensional working area of the machine **on a straight line**.

First of all, the CNC will calculate the required tool path on the basis of the programmed coordinates.

Subsequently, it controls the positioning speed of all axes involved in such a manner that the programmed **path velocity** (rapid feed, feedrate) is never exceeded and all axes reach the target position simultaneously.

For further details, please refer to the G00 and G01 G functions.

**Circular interpolation**

The tool can traverse along a **circular path** (located on the current working plane). First of all, the CNC will calculate the required tool path on the basis of the programmed coordinates (e.g. end and center coordinates of a full circle or of circle segments).

Subsequently, it controls the positioning speed of all axes involved in such a manner that the programmed **path velocity** (rapid feed, feedrate) is never exceeded and all axes reach the target position simultaneously.

For further details, please refer to the G02, G03 and G05 G functions.

**Helical interpolation**

Same as for circular interpolation. In addition, the axis standing perpendicularly on the working plane is carried along linearly.

The Typ3 osa controls the positioning speed of all axes involved in such a manner that the programmed **path velocity** (rapid feed, feedrate) is never exceeded and all axes reach the target position simultaneously.

> Axes that can be moved using the above interpolation types are referred to as "machining axes" or "synchronous axes".
> The standard machining axes include, for instance, X, Y or Z axes.
> Axes used as auxiliary axes (positioning of the tool magazine, e.g.) are called "asynchronous axes".
> Axes of this type cannot be traversed in an interpolary manner. Within the part program, in terms of syntax, they are addressed like machining axis, but from the programmed block on they traverse in rapid feed to the target position.
> If you wish to influence the speed of asynchronous axes, you can use the "FA" auxiliary-function address. For further details about the "FA" address, please refer to page 6–3.
4  G instructions

A table of all G functions of the Typ3 osa is shown in the annex.

4.1  Linear interpolation at rapid travel  G00

Effect

The programmed position is interpolated and approached at max. possible speed on a straight line.

At least one axis travels at max. speed or acceleration. The speed of the other axes is controlled in such a manner that they reach the target point at the same time.

- The speed can be influenced using the potentiometer.
- With active G0, the G0 ACTIVE channel IF signal is output.
- With active G0, the system decelerates to V=0 after each block.

The determination whether G0 is to be active with or without “exact positioning” is made using G161/G162.

If decelerating to V=0 after each block is not desired, you must use the G200 function instead of G0.

Programming

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

Please note for G0:

- Programmable with or without axis addresses.
- No feedrate value has to be programmed. The max. axis speed (100500002) is defined in MACODA.
- The feedrate of G0 rapid travel can be limited to the value set in MACODA parameter 703000110 by means of the channel-related interface signal “Limited Rapid Traverse Velocity” (NC–I1.7).
- Acts modally until a new movement type is selected.
- G0 deletes G1, G2, G3, G5, G10–G13, G73, G200.

Example: rapid-travel programming

X100  Y100  Starting position

G0  X500  Y300  Programmed target position
4.2 Linear interpolation at rapid feed without decelerating to V=0  G200

Effect
If G0 is programmed, the system will – irrespective of G161/G162 – decelerate to V=0 at the block end. If this is not desired, use G200 instead.

This means that interpolation can continue without braking beyond the block limits. However, the following preconditions apply:
- G61 is not active and
- G163 is not active.

If G61 is actually active, the control unit will, despite G200, decelerate to V=0 after each block.

If G163 is active, the behavior depends on the respectively set "exact positioning mode" (please refer to G164 to G166).

The effect of G200 corresponds to "G1 Fmax".

Programming
G200  X...  Y...  Z....

Please note for G200:
- Programmable with or without axis addresses.
- No feedrate value has to be programmed. The max. axis speed (100500002) is defined in MACODA.
- The feedrate of G200 rapid travel can be limited to the value set in MACODA parameter 703000110 by means of the channel-related interface signal "Limited Rapid Traverse Velocity" (NC–I1.7).
- Acts modally until a new movement type is selected.
- G200 deletes the types of movement G0, G1, G2, G3, G5, G10–G13, G73.
4.3 Linear interpolation at feedrate

**Effect**

The programmed point is approached on a straight line at the acting feedrate (F word).

The movement is coordinated in such a manner that all axes involved arrive at the programmed end point simultaneously. At the end of the traversing movement, the control unit will perform a complete downslope down to speed V=0, unless a G8 is active.

The programmed feedrate-value (F) acts as path feed; this means in the case of more than one moving axis that the portion of each individual axis is smaller than F.

The feedrate can be limited by machine parameters (as regards the axis or path).

The speed can be influenced using the feedrate potentiometer.

Use G61/G62 to determine whether G1 is to be active with or without "exact positioning".

**Programming**

G1 X... Y... Z... F....

Please note for G1:

- G1 can be programmed with or without positional data.
- G1 has to be programmed with F word if no feedrate is yet active.
- The programmed feedrate will remain active until it is overwritten by a new one.
- G1 deletes G0, G2, G3, G5, G10–G13, G73 and G200.

**Example:** Linear programming

<table>
<thead>
<tr>
<th>X100</th>
<th>Y100</th>
<th>Starting position</th>
</tr>
</thead>
<tbody>
<tr>
<td>G0</td>
<td>X500</td>
<td>Y300 F100</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Programmed target position</td>
</tr>
</tbody>
</table>

![Diagram showing linear interpolation with X and Y axes, starting position at (100, 100) and target position at (500, 300)]
4.4 Circular interpolation / helical interpolation G02, G03

Effect
The programmed end point is approached on a circular path at the active feedrate (F word).

The determination whether G2/G3 is to be active with or without “exact positioning” is made using G61/G62.

The movement is coordinated in such a manner that all axes involved arrive at the programmed end point simultaneously. This applies also if an axis outside the circular plane is programmed within the block. In this case the Typ3 osa will interpolate this axis linearly together with the other axes. A helical movement will result (helical interpolation).

G2, G3 acts modally and deletes the G functions of the same group or is deleted by them.

The machine traverses at the programmed feedrate and in a circular motion in the selected plane:
• G2 clockwise
• G3 counter-clockwise.

A feedrate value has to be active. By G20, “Plane selection 2 out of 6 axes”, it is possible to execute circles with two freely definable synchronous axes.

For programming, you can chose between
• radius programming and
• center-point programming.

Another option is G05 (circular interpolation with tangential entry).

Depending on the type of programming, various parameters must be programmed in the G02/G03 block. Please refer to the following sections.

4.4.1 Radius programming

Using the current position as starting point, you define a circular movement with the programmed radius leading through the programmed end point.

The end point may be programmed in absolute or incremental terms. The radius will always act as an incremental value.

On the basis of the starting point, end point and radius, the Typ3 osa at first calculates the circle center point. This results in two intersections, located to the left and to the right of the starting/end point distance:
The applicable center point of the two is determined by the sign of the radius value:

- **Positive** radius value: center point left
- **Negative** radius value: center point right

The direction of rotation of the arc is already specified by G2 or G3.

As the diagrams show, the radius must be at least half as large as the distance between the starting and end point, since otherwise no intersection point can be created.

If the radius is exactly half as large as the starting to end point path, this special case results in only one intersection point. This is only possible with a semi-circle. The sign of the radius value is then freely definable.

**Radius programming cannot be used to create full circles.** The smallest possible arc depends on the set machine parameters of the control unit (approx. 10 increments IN POS range).

**Example:**

```
N... G17 G3 X... Y... R±.. F... S... M...
```

**Explanation:**

- **G17**: Selection of the circular path in the X/Y plane
- **G3**: Circle in counter-clockwise sense
- **X,Y**: End point of the circle
- **R**: Circle radius
4.4.2 Center-point programming

Using the current position as starting point, you define a circular movement with the programmed center point leading through the programmed end point.

CAUTION
The compensation threshold in the machine parameter may cause a deviation of center-point coordinates from those programmed!

The CNC will at first use the starting point/center point" and "center point/end point" distances to calculate the required circle radius. If differences occur during this process which are smaller than the machine parameter 705000020 (e.g. caused by inaccurate programming of the center point), the Typ3 osa will automatically correct the center-point coordinates if necessary (the compensation threshold can be set via machine parameter 705000010).

If the differences are greater than that, an error message will be generated.

The axes involved in circular interpolation have the I, J and K "interpolation parameters" assigned.

These define the incremental distance between the A circle starting point and the M circle center point in parallel with the axes. Their sign results in analogy to the vector direction from A to M.

The assignment between the interpolation parameters and the corresponding axes is defined in the "axis classification" machine parameter 701000030.
For details about the interpolation parameters specified for your control unit, please contact your system administrator.

The standard assignment of the interpolation parameters is as follows:

\[
\begin{align*}
  I &= M(X) - A(X) \\
  J &= M(Y) - A(Y) \\
  K &= M(Z) - A(Z)
\end{align*}
\]

I, J, K as interpolation parameters
X, Y, Z axis proportion of relevant coordinate
M for circle center point,
A for circle starting point.
Examples:

Programming

N... G90 G17 G2 X350 Y250 I200 J-50 F... S... M...

Quarter circle as a quadrant

Feature:
One of the interpolation parameters is always zero and is not necessary for programming. In the example, I is left out.

Semicircle made up of two quadrants

Feature:
The coordinates of the starting and end points are the same in one axis. The axis proportion is not required as a target input. The interpolation parameter belonging to this axis is zero and can also be left out. Y and J are left out in the example.
**Full circle**

N... G17 G2 I... F... S... M...

**Features:**

The coordinates of the starting and end points are identical. Both axis proportions may be left out as preset target points. If the starting and end points are situated exactly on a quadrant boundary, one of the interpolation parameters is zero and therefore does not need to be programmed either. X, Y and J are left out in the example.

- If an interpolation parameter is programmed which does not correspond to the selected plane, the control unit will report the "Programmed interpolation parameter outside the selected plane" runtime error.

  **Example:** N... G17 G2 X5 I9 K7   K is invalid

- If interpolation parameters and circle radius are programmed within one and the same block, only the radius will be used.
4.5 Helical-N-Interpolation

Effect

For a helical-N-movement, the travel to the programmed position is executed by the axes providing the working plane travel to this position along a circular arc while all other axes are moved along linearly with the effect that all axes arrive simultaneously. A maximum of 6 synchronous axes can be moved linearly, with permissible axis movement types being linear, endless, or rotary axis.

The function “Helical-N-Interpolation” is a generalised version of the previous “Helical Interpolation”, which is still available (see “Special case” below). With helical interpolation, only one linear feed axis can be moved along and this axis must be configured as a normal axis, i.e. vertically to the selected working plane (MACODA parameter 701000030, axis classified from version 108 up).

Those axes travelling along a circular arc are clearly defined by the selected working plane (G17, G18, G19, G20). The maximum circular movement that can be programmed in one traversing block is a full circle.

The function “Helical-N-Interpolation” allows the programming of “helical movements coupled with a change in orientation”.

Generally, the programmed feedrate applies to all the axes traversing in a block. Those axes that are moved along linearly, however, are controlled by the specific MACODA parameters set for functions G594 and G595, defining the feed.

Any circular or helical movement can also be programmed as an equivalent helical-n-movement.

As a modal function, the helical-n-interpolation remains active until it is deselected by respective programming or by another active function causing a movement.

All standard compensations like zero offset, tool length, workpiece position, or cutter compensation are also effective for helical-n path segments. Helical-n path segments can be programmed also for working in inclined planes.

Programming

G202: circular movement, turning right, clockwise
G203: circular movement, turning left, counter-clockwise

Both radius programming (R) and center-point programming (I, J, K) is possible:

- The sign of the radius determines whether the resulting center of the circle is located left (+) or right (−) of the line between the starting point and the end point.
- The size of the radius must be at least half the distance between the starting and the end point. If the radius is smaller than this value and if the balance lies within the tolerance window defined in MACODA parameter 705000030, the radius will be corrected automatically to half the above distance.
- If center-point programming is used, the coordinates of the center point refer to the starting point of the circular movement (center-point coordinates are incremental).
- If the starting point and the end point in the circle plane are identical, the control will generate automatically a full circle if center-point programming is used.
MACODA parameters 705000010 and 705000020 allow to configure the required programming accuracy.

If a center-point coordinate is programmed as lying outside the working plane, the control will display a runtime error.

Because the helical-n-interpolation is a modal function causing a movement, the active G-code is shown on the MMI display of the respective modal function activated.

The behavior upon resetting the control or switching on/off is determined by the init strings configured in the MACODA for control start-up or following control reset.

This function belongs to group 2.

Because the number of axes per channel is limited to a maximum of 8, coupled linear motion is limited to a maximum of 6 synchronous axes.

Special case

Helical interpolation

If a third axis is programmed in addition to circular interpolation of two axes, this third axis will traverse linearly. The result is a helical movement (see also the figure below).

The tool-path compensation acts within the circular-path plane which can be freely selected via plane selection (G17...). The F feedrate corresponds to the real path speed.

Example:
Circular interpolation with the X and Y axes, linear interpolation with the Z axis:
N... G91 G17 G3 X... Y... Z... I... J... F... S... M...

Features:
The coordinates of the starting and end point are identical with regard to the X and Y coordinate. The K interpolation parameter is inapplicable since the starting point is within the X/Y plane.

Application: e.g. thread cutting
4.6 Dwell time

Effect  
G4 is programmed with an F word for the dwelling duration without positional data. It acts only in the programmed block.

Only auxiliary and additional functions are permissible in the dwell time block.

The dwell time is started when the dwell time block has been completely processed by the CNC. The program is stopped for the dwell duration. All movements are continued up to the target point (tool change,lag etc.). The spindle, however, is not stopped.

Programming  
N... G4 F... (dwell time in seconds)

If you wish a dwell time for subsequent blocks again, you must program G04 plus F word in each of these blocks again.

4.7 Circular interpolation / helical interpolation with tangential entry

Effect  
The control unit uses G5 to calculate a tangential circle entry. Only a transition involving no reversal of direction is referred to as 'tangential'.

The first entry tangent determines all following contour elements with G5 if several G5 movements take place consecutively.

The size and position of the arc formed are calculated by the control unit in accordance with the following:

Programming  
G5  X...  Y...

No radius is programmed.

Restrictions:

- Programming G5 in manual data input or as 1st block within the program is impossible, since no tangent can be calculated there.

- A block containing a traversing movement has to be programmed ahead of G5.

- The plane must not be switched over directly ahead of or during an active G5.

CAUTION  
When helical interpolation is used, machining marks may occur at the block transition!

The tangential transition refers only to the circle plane! The spatial tangent (with helical interpolation) may jump at the block transition!
Influence of tangent

\[ T_n = \text{tangent} \quad A = \text{start of circle segment} \]
\[ M_n = \text{center point} \quad E = \text{end of circle segment} \]
4.8 Acceleration programming

Effect The upper limits of the max. axis acceleration defined in MACODA (please refer to MACODA parameter 101000001) can be reduced temporarily within the part program via G6.

DANGER
Incorrect axis addressing may cause inadvertent axis movements that may pose a hazard to the machine and personnel.

This programming refers directly to a real physical axis. A logical axis, addressed, for instance, by a coordinate transformation (e.g. inclined plane) with the same axis address will lead to incorrect axis values. This might result in damage to the workpiece and/or the machine. There might even be danger to persons.

Programming

<table>
<thead>
<tr>
<th>G06 with axis information:</th>
<th>Supersedes the max. MACODA axis acceleration values defined in parameter 101000001 with the programmed values. Depending on the currently used measuring units (G71/G70), the Typ3 osa will interpret the programmed values as &quot;1000 inch/s²&quot; or &quot;m/s²&quot;. You should program G6 preferably in a separate block.</th>
</tr>
</thead>
<tbody>
<tr>
<td>G06 without axis information:</td>
<td>see G206.</td>
</tr>
<tr>
<td>G07</td>
<td>The max. acceleration values defined in parameter 101000001 apply again to all axes. G7 may be programmed together with traversing information.</td>
</tr>
<tr>
<td>G206</td>
<td>Storing of the currently valid max. acceleration values of all axes in an internal memory. During program selection, this memory is initialized with the values defined in parameter 101000001. Programming G6 without axis information will reactivate all the acceleration values stored in this memory.</td>
</tr>
</tbody>
</table>

Example 1:

G6 X2 Y2   max. acceleration of axes X and Y is 2m/s², each.
Example 2:
Starting situation: The value of 8.0 m/s² is preassigned to axes X through Z in parameter 101000001.

G6 X1.0 Z2.1  max. acceleration for X axis: 1.0 m/s²
...          max. acceleration for Y axis: 8.0 m/s²
             max. acceleration for Z axis: 2.1 m/s²
G206         storing all current acceleration values
...          reactivated the values defined in MACODA parameter 101000001
G6 Y5        max. acceleration for X axis: 8.0 m/s²
...          max. acceleration for Y axis: 5.0 m/s²
             max. acceleration for Z axis: 8.0 m/s²
G6           max. acceleration for X axis: 1.0 m/s²
...          max. acceleration for Y axis: 8.0 m/s²
             max. acceleration for Z axis: 2.1 m/s²
4.9 Path slope

Effect

Using the “path slope” function, the control unit attempts during contour machining to generate a speed as constant as possible within the magnitude of the programmed feedrate.

Without “path slope”, the control unit performs a complete up and down slope (speed ramp) at the start and end of a traversing block.

With “path slope”, this slope will – except for the beginning and end of machining – only take place to the extent required for going around a corner. In this process the Typ3 osa takes into account the value for the max. axis step change.

Please note that the two time axes in the above illustration have different scalings.

With active G8, the P8 point will be approached within a shorter time than with active G9.

Decelerating to V=0 is performed after each G0 block!

After a G200 block, decelerating to V=0 is only performed if

- G61 or
- G163 are active!

Programming

G08: Path slope on
G09: Path slope off

The function has a modal effect. Path slope acts only on the machining axes.

Example: G08, path slope ON

N... G8          (path slope ON)
N... G0 X100 Y50  (at rapid travel to P1)
N... G1 X150 F5000 (continue at feedrate)
N...

When programming auxiliary functions while the path slope function is active, please make sure that the travel paths programmed are long enough for the amount of time required for interpolating the NC block is greater than the time required for executing the auxiliary function, including acknowledgement. (Basically, the time required for executing an NC block is defined by the travel path programmed and the feedrate.)
4.10 Path slope taking the whole deceleration slope into account  

Effect

The function "Path slope G08" is designed to calculate always a speed profile as smooth as possible for several blocks in a row. To avoid the effects of potential variations in the look ahead, the speed at the end of a block is reduced only to the extent required so as not to exceed the maximum step change permissible for the next corner or to ensure proper traversing at reduced speed in the next block.

If the deceleration distance available in the last block is taken into account within the available look-ahead range, this would have the effect that due to the limited look-ahead range the speed would have to be reduced more than technically or physically required.

Normally, transitions from block to block or transition angles in the paths of programmed contours are more or less smooth. In a few special cases, however, when the contour ends abruptly after the last block taken into account, excessive acceleration values may occur because the path of any such last block is not long enough for deceleration.

Using G108, traversing in these special cases – which can usually also be solved by block splitting – can be adjusted so as to prevent excessive acceleration. The look-ahead range for computing the required deceleration distance can be set with MACODA parameter 705000310, which in turn is limited by parameter 706000110 or by the PREPNUM function.

Programming

N... G0.... G108 Activate G108

Example:

G8
...
N998 X1000
N999 G0 X5000 G10 8 At the end of N999, the speed is reduced so as to obtain the same speed curve as with "Solution 2"
...
N1000 X5001
M30

Function G108 is not suitable for contour machining, in particular not for machining involving high block cycle time requirements.
In the following example, G8 and G108 are compared to demonstrate the programming of G108:

**Example:**

G8

...  
N998  X1000  
N999  G0  X5000  
N1000  X5001  Depending on the machine parameter, excessive acceleration may occur.

M30

**Solution 1:**

Because the path in block N1000 is very short, rapid travel does not make any sense! Therefore, feedrate (G1) is used in block N1000.

G8

...  
N998  X1000  
N999  G0  X5000  
N1000  G1  X5001  Traversing at G1.

M30

**Solution 2:**

Because there is neither a directional nor a modal change from block N999 to block N1000, N999 and N1000 can be combined.

G8

...  
N998  X1000  
N999  G0  X5000  
N1000  G1  X5001  Blocks N1000 and N999 are combined.

M30
4.11 Block transition without deceleration

Effect
The “Path slope” function computes the velocity at the transition from one NC block to the next, taking into account the programmed feed, the maximum step change, and the available deceleration distance. To limit contour deviations occurring at real corners, the maximum step change must not be set at too high a value in the MACODA. On the other hand, if the maximum step change is set too low, this will result in undesirable deceleration at minor knees in the contour (quasi-continuous transitions).

The function “Block transition without deceleration” allows to limit the impact of the maximum step change on wide transition angles. By proper parameterisation, the behavior can be defined so as to take major knees in the contour into account in detail, whereas quasi-continuous contour transitions are rounded and thus smoothed due to the higher machining speed.

Programming
G228 \{K<transition angle>\}

Explanation:
K K address
transition angle transition angle = 0° to 50°

If G228 is programmed without stating the K address, the basic transition angle setting made in MACODA parameter 703000310 will be activated.

Please note for G228:

- G228 is not modal as such, but it acts modally.
- After a control reset, the respective init string setting is activated. If there is no G228 entered there, the previously activated setting remains active.

When configuring MACODA parameter 703000310, the desired traversing speed must be taken into account because at a high speed and a wide transition angle it is theoretically possible that a servo error occurs.
4.12 Point-to-point movement using SHAPE

Effect

The shape function is used to share out jumps in the course of path acceleration between several interpolation cycles. This enables "sin²-shaped" path-speed behavior patterns, i.e. jerk-free speed changes, to be made.

Path-speed behavior patterns

Compared to G9 (unchanged acceleration), the interpolation time ("with shape") per point-by-point movement extends by order*interpolation cycle.

Using the LIN and SIN parameters, the characteristics of the acceleration transition and thus a jerk-free path speed can be set.

**LIN <number>**

Number of interpolation cycles (settings: 2 – 41 cycles) between which an occurring path-acceleration jump is to be shared out. The acceleration increase or decrease is linear.

**SIN <numerical value>**

Activation of firmly set acceleration-behavior patterns. The acceleration increase or decrease is sin²-shaped.
The following **firmly set** \( \sin^2 \)-shaped acceleration patterns are preset within the system:

- **SIN 0**: SHAPE is cancelled (=G9)
- **SIN 3**: 3 interpolation cycles in the ratio of 25% – 50% – 25%
- **SIN 4**: 4 interpolation cycles in the ratio of 12.5% – 37.5% – 37.5% – 12.5%
- **SIN 5**: subdivided into 5 interpolation cycles
- **SIN 10**: subdivided into 10 interpolation cycles
- **SIN 15**: subdivided into 15 interpolation cycles
- **SIN 20**: subdivided into 20 interpolation cycles
- **SIN 40**: subdivided into 40 interpolation cycles

The SIN parameter has priority over the LIN parameter.

*Linear and \( \sin^2 \)-shaped acceleration transitions*
Programming

G408: Default setting corresponds to G408 LIN 2
     (2 interpolation cycles)
G408 SIN 3 LIN 5: Acceleration jump only with SIN 3
     (3 interpolation cycles with firmly set acceleration
     pattern)
G408 LIN 5: Acceleration jump with LIN 5
     (5 interpolation cycles)
G408 LIN 2: corresponds to default setting
     (2 interpolation cycles)

Invalid programming:

G408 SIN 5  SIN valid, therefore:    G408 SIN 5
G408 LIN 41 LIN invalid (value too   G408 LIN 2
     high), therefore:
G408 SIN 3 LIN 5  SIN valid, therefore:    G408 SIN 3
G408 SIN 7 LIN 5  SIN invalid, therefore:    G408 LIN 5
G408 SIN 7 LIN 41 SIN and LIN invalid,    G408 LIN 2
     (value too high), therefore:

The SIN parameter supersedes the LIN parameter.

The default setting will be selected if invalid values were programmed for LIN
or SIN.

Please note for G408:

- G408 acts modally (belongs to the G8, G9, G108, G608 group)
4.13 Axis-by axis programmable SHAPE

Effect
Using the shape function, which can be programmed axis by axis, you can define for each synchronous axis a maximum permitted jerk that must not be exceeded in traversing.

Each axis is programmed to have one special shape order of its own. Internally, the control unit forms a resultant shape order based on all the axes involved for path interpolation purposes.

The shape order determines the sharing out of the path acceleration of one individual axis between a programmed number of interpolation steps (also ref. to G408).

For this purpose, the programmed shape order determines the max. shape order which can act for the respective axis.

If more than one axis is involved in the interpolation, the resulting shape order acting along the path is computed.

Programming
G608 <axis i><shape order i> ... <axis n><shape order n>

Explanation:
axis i logical i\(^{th}\) axis
shape order i the programmed max. shape order of the i\(^{th}\) axis (= number of interpolation cycles (21 max.) among which an occurring path acceleration jump of the i\(^{th}\) axis is to be shared out).

Example:
N.. G608 X4 Y6 Z10
shape order (X axis) = 4
shape order (Y axis) = 6
shape order (Z axis) = 10

Please note for G608:
- The G608, G8, G9, G408 functions act modally and cancel each other mutually.
- The shape order must only take on integer values (1<= shape order<=21)
- Unprogrammed axes will be preallocated with default values (also ref. to MACODA parameter 100300008).
- If G608 is programmed alone, all the axes are preallocated with default values (also ref. to MACODA parameter 100300008).
- An active G608 function will always lead to a block-transition speed = 0 and is therefore only suitable for positioning movements.
- In power-up condition, G09 is activated.
Resulting shape order

The resulting path shape order $S_p$ is the maximum of the effective axis shape order $S_{\text{eff}}^i$ of all the axes involved in the interpolation.

$$S_p = \max \{ S_{\text{eff}}^1, \ldots, S_{\text{eff}}^n \}$$

The effective axis shape orders $S_{\text{eff}}^i$ are computed from the programmed shape orders using the formula:

$$S_{\text{eff}}^i = S_p \frac{a_{\text{eff}}^i}{a_{\text{max}}^i}$$

Explanation:

$S_p^i$ Axis shape order programmed with G608

$a_{\text{eff}}^i$ Effective axis acceleration in the current NC block. With linear interpolation, this depends on the current path segment of the axis. With other types of interpolation (e.g. circular, helical), it is the axis acceleration usually programmed with G06.

If the functions “inclined plane” or “axis coupling” are used, the effective axis acceleration is generally decreased once again as compared to the G06 value!

$a_{\text{max}}^i$ Maximum axis acceleration (MACODA parameter).

Caution: G06 does not change this value!

Interrelationship between shape order and jerk

With axis shape orders $S_{\text{eff}}^i$, a maximum jerk $j_{\text{max}}^i$ (derivative from acceleration after time) is defined as a limit not to be exceeded in any movement.

This jerk is defined by:

$$j_{\text{max}}^i = \frac{a_{\text{max}}^i}{S_p^i T_{\text{ipo}}}$$

$T_{\text{ipo}}$ is the cycle time of the interpolator.

Example:

Let us assume the X axis has a maximum acceleration (MACODA parameter) of 10 m/s$^2$, the programmed axis shape order is 5, and the interpolation cycle time is 4 ms. Applying the above formula, the maximum jerk defined for the X axis is 500 m/s$^3$. 
4.14 Polar-coordinate programming

Effect

Within the polar-coordinate system, and in contrast to Cartesian coordinate systems, you specify points by defining the radius and angle starting from a freely selectable pole.

The pole corresponds to the origin of the polar coordinate system and can be defined via G20 within all admissible planes.

Example:
G20 X100 Z100 Pole lies on the Z/X plane at the Cartesian coordinates X = 100 Z = 100.

- If the pole is not programmed, the Typ3 osa will always use the coordinate origin as pole.
- The position of a point is described by the radius axis (one of the two axis forming the plane), the radius value and the angle. This angle relates to the programmed radius axis (see the example below). The syntax "A" of this angle may be declared differently in MACODA parameter 80050001.
- A positive axis direction of the radius axis always corresponds to the angle value of 0 degrees. All angular information refers to the positive axis direction.

Example 1
N150  G20  Z25  X10 Pole determination
N160  G10  Z20  A70 Determination of the radius axis incl. radius value and angle)

Example 2
N150  G20  Z30  X20 Pole determination
N160  G10  X20  A70 Determination of the radius axis incl. radius value and angle)

You select the desired type of interpolation via the corresponding G instruction.

Explanation:
G10  Polar-coordinate programming at rapid travel (corresponds to G0)
G11  Polar-coordinate programming at feedrate (corresponds to G1)
G12  Polar-coordinate programming with circular clockwise interpolation (corresponds to G2, without helical)
G13  Polar-coordinate programming with counter-clockwise circular interpolation (corresponds to G3, without helical)

The G0, G1, G2, G3, G5 and G10–G13 functions act modally and cancel each other mutually.

Unless explicitly switched over, the plane programmed via G20 during pole determination will also remain active after the cancellation of polar-coordinate programming.
4.15 Loop gain (KV) programming

Effect

The function enables the program-controlled change of the KV values of individual axes.

This can be used for short-term increases of the rigidity of axes (e.g. for milling a bore). The KV values for the programmed axes, defined as machine parameters, are irrelevant during active KV programming.

The following applies:

\[ KV = \frac{V}{S} \]

where:

- \( V \) = feedrate
- \( S \) = lag

Decelerating to \( V=0 \) is performed ahead of each block containing a KV switch, since the KV value in the drive should only be switched over at standstill.

KV switching as such is always carried out directly ahead of the next traversing movement.

---

DANGER

Incorrect axis addressing may cause inadvertent axis movements that may pose a hazard to the machine and personnel.

This programming refers directly to a real physical axis. A logical axis, addressed, for instance, by a coordinate transformation (e.g. inclined plane) with the same axis address will lead to incorrect axis values. This might result in damage to the workpiece and/or the machine. There might even be danger to persons.

---

Programming

<table>
<thead>
<tr>
<th>Programming</th>
<th>G14:</th>
<th>KV programming ON</th>
</tr>
</thead>
<tbody>
<tr>
<td>G15:</td>
<td>KV programming OFF</td>
<td></td>
</tr>
</tbody>
</table>

Example:

- G14 X1.20 Y1.20 Z1.20 the KV value of "1.2" is defaulted for the X, Y and Z axes
- G14 Z1.4 KV value of "1.4" defaulted for the Z axis
- G15 X200 Y300 Z-150 The KV parameters (S-0-0104) defined in the SERCOS file apply again.

The max. programmable KV value is "655.35".

G15 may also be programmed without axis information.
4.16 No plane

Effect

Function G16, "No plane", is to be selected for the following applications:

- If with the Axis transfer function (section 4.70) a principal or a secondary axis on the previously active plane is removed from a channel, the control unit will automatically deactivate the plane currently selected and activate function G16, "No plane". In this case, circular or helical interpolation cannot be performed on this channel before a valid plane is selected.

- If there is no plane function (G17, G18, G19, G20) entered for an active channel after power-up (MACODA parameters 706000010 and 706000020), function G16, "No plane", will be activated for the respective channel by implication. In this case, no entry will appear on the channel-specific display of active functions.

- Some applications, types of machines, or processing functions do not require the definition of a plane because, e.g., no circular or helical interpolation is necessary (e.g., for channels with only one machining axis assigned). In this case, axis classification (MACODA parameter 701000030) is also irrelevant and you can just enter 999 – of no functional relevance – for the classification of each axis.

Programming

No plane:

- G16 deactivate plane selection

Please note for G16:

- Function G16 is modal and forms a group together with G17...G20. These functions mutually cancel each other.

- After M30/control reset, the plane as defined in the MACODA for power-up condition is automatically reactivated.
4.17 Plane selection:
X/Y plane G17
Z/X plane G18
Y/Z plane G19

Effect

Used to define the working plane within the workpiece or program coordinate system. The effects of G2, G3, G5 as well as the polar-coordinate programming and the tool compensations are linked to this function.

Immediately after the activation of an axis, the Typ3 osa places the pole for polar-coordinate programming into the origin of the plane’s coordinates. If an angle value is still active, it will be set to "0".

Within a Cartesian coordinate system, the three X, Y and Z axes form three different basic planes. These planes are characterized in that the respective third axis, as feed axis, is standing perpendicularly on these planes.

The following illustration shows the principle:

Considering that the Typ3 osa enables any names to be assigned to the individual axes, the machine can also be configured using axis descriptions other than X, Y and Z. In this case the individual planes are assigned according to their "axis classification" (please refer to machine parameter 701000030). The following assignment scheme applies here:

<table>
<thead>
<tr>
<th>Classification of the main axis</th>
<th>Classification of the secondary axis</th>
</tr>
</thead>
<tbody>
<tr>
<td>G17</td>
<td>1</td>
</tr>
<tr>
<td>G18</td>
<td>3</td>
</tr>
<tr>
<td>G19</td>
<td>2</td>
</tr>
</tbody>
</table>

If no axis meeting the above classification is defined for a selected plane, the "Selected plane cannot be configured" runtime error will be displayed.
The axis classification also determines the **addresses of the interpolation parameters** for circular and helical interpolation in the case of center-point interpolation:

<table>
<thead>
<tr>
<th>Axis classification</th>
<th>Address of the interpolation parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>I</td>
</tr>
<tr>
<td>2</td>
<td>J</td>
</tr>
<tr>
<td>3</td>
<td>K</td>
</tr>
</tbody>
</table>

If more than 2 machining axes (= feed axes) are defined within the system, the **infeed axis** is determined according to the following assignment scheme:

<table>
<thead>
<tr>
<th>Plane</th>
<th>Classification of the infeed axis</th>
</tr>
</thead>
<tbody>
<tr>
<td>G17</td>
<td>3</td>
</tr>
<tr>
<td>G18</td>
<td>2</td>
</tr>
<tr>
<td>G19</td>
<td>1</td>
</tr>
</tbody>
</table>

If no axis meeting the stated axis classification has been defined, the system will always select as feed axis the first axis within the system which does not have the axis classification of the main and secondary axis.

**Example:**

<table>
<thead>
<tr>
<th>System axis index *)</th>
<th>Axis address</th>
<th>Axis classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Y</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>B</td>
<td>200</td>
</tr>
<tr>
<td>2</td>
<td>C</td>
<td>300</td>
</tr>
<tr>
<td>3</td>
<td>X</td>
<td>1</td>
</tr>
</tbody>
</table>

*) corresponds to the order of the machine parameters

In the case of a G17 plane, the B axis would be the feed axis, since it is the axis with the lowest system axis index (except for the axis having the 1 and 2 classification) and since, in addition, an axis with a classification of 3 does not exist.

From a functional perspective, the definition of a pole via G20 is equivalent to the selection of a plane.

**Impact on tool compensation with standard axis classification being used:**

<table>
<thead>
<tr>
<th>G instruction</th>
<th>Tool-radius compensation</th>
<th>Tool length compensation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Circular interpolation</td>
<td>Infeed axis with standard bore (drill) cycles</td>
</tr>
<tr>
<td>G17</td>
<td>X/Y plane</td>
<td>Z axis</td>
</tr>
<tr>
<td>G18</td>
<td>Z/X plane</td>
<td>Y axis</td>
</tr>
<tr>
<td>G19</td>
<td>Y/Z plane</td>
<td>X axis</td>
</tr>
</tbody>
</table>

**Example:**

N ... G19 ... (Selection of the Y/Z plane)

- G16, G17, G18, G19 and G20 are modal and cancel each other mutually (for G16, "No plane", please refer to section 4.16).
- After M30, the plane defined as power-up condition in the machine parameters will automatically become active.
- No plane change must be programmed with active cutter-path compensation (G41 or G42).
4.18 Plane selection 2 out of 6 axes
Pole programming for polar-coordinate programming

Effect

G20 allows the free selection of the circular interpolation and cutter-radius compensation plane. In addition, G20 is used to determine the pole for polar-coordinate programming (please refer to G10–G13).

CAUTION
Incorrect programming may lead to confusion and thus cause damage to the machine.

Whereas the X, Y and Z axes represent the 3 main axes of the current workpiece coordinate system, all the other axis addresses (e.g. the "A" rotary axis) always designate real physical axes.

Programming

In the G20 block you indicate the axes of the desired plane. The control unit will interpret programmed axis values (e.g. X100 Y40) as pole coordinates.

N... G20 X0 Y0 (Selection of the X/Y plane as interpolation plane. The pole for polar-coordinate programming is set to X=0 and Y=0)
N... G20 Y100 Z200 (Selection of the Y/Z plane as interpolation plane. The pole for polar-coordinate programming is set to Y=100 and Z=200)

- G20 may only be programmed together with two axis addresses. If G20 is programmed without, with one or with more than two axis addresses, an error message is output and processing is stopped at the end of the previous block.
- G20 is latching. It will delete the G17, G18 and G19 functions. After M30 the plane defined as power-up condition in the machine parameters will automatically become active.
- After M30 the plane defined as power-up condition in the machine parameters will automatically become active.
- The axes of the selected plane will automatically have the cutter-path compensation assigned.
- G20 may not be programmed during active cutter path compensation; i.e. an active cutter path compensation must be deactivated with G40 before a new plane is selected.
- With active helical interpolation, the programmed axis which does not lie within the circular interpolation plane will be driven along linearly.

Circular interpolation is only possible in the axis of the selected planes. As is described in section 4.4.2, center-point programming requires the specification of interpolation parameters. The assignment of interpolation parameters and corresponding axis is defined in the "axis classification" machine parameter 701000030.

Determination of the interpolation parameters:
The table to the right shows the correlation between the axis classification and the required interpolation parameter. If both axes of the plane are within one and the same column, the parameters at the right margin apply, otherwise those at the bottom margin.
Function G21, "Axis classification" (see also G17, G18, G19, plane selection, in section 4.17), defines the **functional significance** of an axis on a machining channel.

**Axis classifications** define the following:

- The axes defining the G17, G18 and G19 planes and which of these axes are the principal, the secondary and the feed axes,
- which of the two axes programmed in G20 is the principal axis and which one is the secondary axis, and
- which of the interpolation parameters I, J, and K is assigned to the respective principal axis and the secondary axis for circular and helical interpolation.

You can specify the axis classification of all logical axes on each channel by using MACODA parameter 701000030.

**Transferring an axis** may impact the functional relevance of logical axes. Therefore, the following definitions apply:

- If an axis not included in the power-up condition of a channel is transferred to this channel, this axis is assigned the "neutral axis classification" 999 (no functional significance) for the time being.
- Classifications of axes in the part program can be changed using function G21.
  - This function allows to include an axis not pertaining to a channel in its start-up condition with the effect that the axis thus included can be used to co-define a plane and taken into account for circular or helical interpolation.
  - When programming axis classification G21, please note that no axis classification of functional significance (1, 2, 3, 10, 20, 30, 100, 200, 300) may be assigned more than **once** on any one channel. By contrast, axis classification '999' – no functional significance – may be assigned any number of times on any one channel.
- If an axis pertaining to a channel in its power-up condition is first transferred to another channel and then transferred back to its original channel, this axis is reassigned its original axis classification in the power-up condition.
- If an axis is transferred from a group of axes on the currently active plane, i.e. this axis is either the principal or secondary axis of the selected working plane, this selected plane is not available any more because one of its defining elements is missing.
  - The control unit will then implicitly deactivate the selected plane and instead activate the **G16** function, "No plane".

**Example:**

N100 G17 X0 Y0 Z0

...  

N200 G512(Y)

Y is removed from the group of axes. Implicit switch to G16. Circular interpolation is rendered impossible.

N210 G511(YA)

Axis YA is included in the group of axes and assigned a neutral classification.

N220 G21 YA2

YA is assigned axis classification 2.

N230 G17

Switch to X/YA plane.

N240 G2 X.. YA..

Circular interpolation is possible again.
Programming

**Programming of axis classifications:**
G21 (<LANi> <axis classification>,..,<LANn> <axis classification>)

Explanation:
LAN designation of logical axis/axes
Axis classification axis classification value that may be entered

Permitted values:
1, 2, 3, 10, 20, 30, 100, 200, 300, 999.
Please note that – with the exception of ‘999’ – no axis classification may be assigned more than once on each channel!

Example:
G21 X1 Y2 X3 B200 In the part program, the default axis classification is as follows:
X=1, Y=2, Z=3, B=200

---

**4.20 Table activation**

**Effect**

Use G22 to activate:
- zero-shift tables
- compensation tables
- tables for the “inclined plane” function

These tables are stored as ASCII files in the file system of the Typ3 osa. The number of tables is limited by the storage capacity of this file system.

**Programming**

N... G22 V{<path>}<file name> activation of zero-shift table
N... G22 K{<path>}<file name> activation of a compensation table
N... G22 ID{<path>}<file name> activation of a compensation table “inclined plane”

**Explanation:**

<file name> freely defined file name
<path> optional statement of path (directory) where the file is stored
Examples:

- **G22 V/mnt/npvtab1.npv**
  Activates the zero-shift table "npvtab1.npv" in the mounted directory "/mnt".

- **G22 Kgeotab2.geo**
  Activates the geometry compensation table "geotab2.geo". The file will be searched along the the "/database" directory. If the table you search is available there, it will be activated. Otherwise, the search will continue on the search path for subprograms and the first table with the name geotab2.geo that is found will be activated.

- **G22 Vnpvtab3.npv Kgeotab3.geo**
  Tables "npvtab3.npv" and "geotab3.geo" will be searched in the "/database" directory and – if not found there – on the search path for subprograms and then activated. Within one and the same block, several tables can be activated.

Please note for G22 and zero shift tables:

- Table columns are assigned to the axes on a channel via the axis names entered on the table. These may be names of both logical and physical axes, with logical axis names taking precedence over physical axis names.

  You can select the option "Strict assignment" for the table in the table editor (or when writing the table in CPL).

  If the option "Strict assignment" is available for a zero-shift table and if this table is activated using **G22**, the system will check whether the current axis configuration of the respective channel matches the table entries. If this is not the case, an error message is displayed and program execution is canceled.

  If the option "Strict assignment" has not been activated, any discrepancies between the current axis configuration and the table columns will not give rise to an error message. This allows to activate tables containing shift values of just some of the axes. Also, this allows to use tables containing additional columns with shift values of axes to be transferred to the respective channel at a later point in time.

**Example 1:** G22 V npvtab1.npv (ZS table on channel 1)

There are 3 axes assigned to channel 1.

strict assignment: 3 channel axes <-> 3 table axes

**Example 2:** G22 V npvtab2.npv (ZS table on channel 2)

There are 4 axes assigned to channel 2

no strict assignment: 3 channel axes <-> 2 table axes

---
No error message displayed because the "Strict assignment" option has **not** been activated.

**Example 3:** G22 V npvtab1.npv (ZS table on channel 2)

There are 4 axes assigned to channel 2.

"Strict assignment" option has been activated.

---
An error message is displayed because the "Strict assignment" option has been activated.
npvtab1.npv with "Strict assignment" activated

<table>
<thead>
<tr>
<th></th>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>G54</td>
<td>54.01</td>
<td>54.02</td>
<td>54.03</td>
</tr>
<tr>
<td>G55</td>
<td>55.01</td>
<td>55.02</td>
<td>55.03</td>
</tr>
<tr>
<td>G56</td>
<td>56.01</td>
<td>56.02</td>
<td>56.03</td>
</tr>
<tr>
<td>G57</td>
<td>57.01</td>
<td>57.02</td>
<td>57.03</td>
</tr>
<tr>
<td>G58</td>
<td>58.01</td>
<td>58.02</td>
<td>58.03</td>
</tr>
<tr>
<td>G59</td>
<td>59.01</td>
<td>59.02</td>
<td>59.03</td>
</tr>
<tr>
<td>G154</td>
<td>154.01</td>
<td>154.02</td>
<td>154.03</td>
</tr>
<tr>
<td>G155</td>
<td>155.01</td>
<td>155.02</td>
<td>155.03</td>
</tr>
<tr>
<td>G156</td>
<td>156.01</td>
<td>156.02</td>
<td>156.03</td>
</tr>
<tr>
<td>G157</td>
<td>157.01</td>
<td>157.02</td>
<td>157.03</td>
</tr>
<tr>
<td>G158</td>
<td>158.01</td>
<td>158.02</td>
<td>158.03</td>
</tr>
<tr>
<td>G159</td>
<td>159.01</td>
<td>159.02</td>
<td>159.03</td>
</tr>
<tr>
<td>G254</td>
<td>254.01</td>
<td>254.02</td>
<td>254.03</td>
</tr>
<tr>
<td>G255</td>
<td>255.01</td>
<td>255.02</td>
<td>255.03</td>
</tr>
<tr>
<td>G256</td>
<td>256.01</td>
<td>256.02</td>
<td>256.03</td>
</tr>
<tr>
<td>G257</td>
<td>257.01</td>
<td>257.02</td>
<td>257.03</td>
</tr>
<tr>
<td>G258</td>
<td>258.01</td>
<td>258.02</td>
<td>258.03</td>
</tr>
<tr>
<td>G259</td>
<td>259.01</td>
<td>259.02</td>
<td>259.03</td>
</tr>
</tbody>
</table>

npvtab2.npv with "Strict assignment" deactivated

<table>
<thead>
<tr>
<th></th>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>G54</td>
<td>54.01</td>
<td>54.02</td>
</tr>
<tr>
<td>G55</td>
<td>55.01</td>
<td>55.02</td>
</tr>
<tr>
<td>G56</td>
<td>56.01</td>
<td>56.02</td>
</tr>
<tr>
<td>G57</td>
<td>57.01</td>
<td>57.02</td>
</tr>
<tr>
<td>G58</td>
<td>58.01</td>
<td>58.02</td>
</tr>
<tr>
<td>G59</td>
<td>59.01</td>
<td>59.02</td>
</tr>
<tr>
<td>G154</td>
<td>154.01</td>
<td>154.02</td>
</tr>
<tr>
<td>G155</td>
<td>155.01</td>
<td>155.02</td>
</tr>
<tr>
<td>G156</td>
<td>156.01</td>
<td>156.02</td>
</tr>
<tr>
<td>G157</td>
<td>157.01</td>
<td>157.02</td>
</tr>
<tr>
<td>G158</td>
<td>158.01</td>
<td>158.02</td>
</tr>
<tr>
<td>G159</td>
<td>159.01</td>
<td>159.02</td>
</tr>
<tr>
<td>G254</td>
<td>254.01</td>
<td>254.02</td>
</tr>
<tr>
<td>G255</td>
<td>255.01</td>
<td>255.02</td>
</tr>
<tr>
<td>G256</td>
<td>256.01</td>
<td>256.02</td>
</tr>
<tr>
<td>G257</td>
<td>257.01</td>
<td>257.02</td>
</tr>
<tr>
<td>G258</td>
<td>258.01</td>
<td>258.02</td>
</tr>
<tr>
<td>G259</td>
<td>259.01</td>
<td>259.02</td>
</tr>
</tbody>
</table>

Compatibility with previous software versions:

- In previous software versions, tables were managed in the database and table numbers were used for programming instead of file names. In order to ensure compatibility with previous programs, notation of table numbers is still supported.

- ZS tables in previous software versions do not contain axis names. If there is a channel identification contained on a table (software version 4.3x), this channel identification must be identical with the channel on which function G22 is executed. Otherwise, an error message will be generated and program execution will be canceled. If these channels are identical, the channel identification serves to assign the table columns to the channel axes.

- If the ZS table does not contain any channel identification, the logical axis index in the initial configuration of the channel on which function G22 is executed determines the assignment of table columns to the axes.

- If you use tables from previous versions, please make sure that the correct unit is entered in MACODA parameter 902000001.

Please refer to the Typ3 osa Operating Manual on how to create or edit tables!
4.21 Jump destinations:
- Unconditional jump (block number) G24
- Conditional jump (interface signal) G23
- Jump backward GOTOB
- Jump forward GOTOF

Generally, the blocks of main programs, subprograms and cycles are executed in the same sequence in which they have been programmed.

The processing sequence can be changed by program jumps. There are various jump destination types available for this purpose.

See the explanation, section 2.1.4, p. 2–10 ff.
4.22  Tapping without compensation chuck

Effect
The Typ3 osa synchronizes linear interpolation of the drill axis with the spindle switched to C axis operation. This eliminates the need for a compensation chuck which would otherwise be required for taking up the speed difference between the drill axis and the spindle.

The “G32 ACTIVE” interface signal will be output for the duration of the tapping process. During this time only the feed potentiometer is active.

Programming
G32  <drill axis><infeed per cut> {F<feedrate>} M<3|4>  
S<speed>|H<thread pitch>

In addition to the infeed per cut, the following must be entered for programming a G32 block:

- spindle speed (S) or the thread pitch (H) and
- the sense of rotation (M3/M4)

M and S act only within the programmed G32 block.

The Typ3 osa uses the active path feed (F word) if no other value is stated in the G32 block.

The thread pitch results from the ratio between the path feed and the speed, unless the thread pitch (H) is programmed.

Example:
N10  G0  X20  Y15  Z10  F1000  S5000  Positioning the axes
N20  G32  Z–20  F1000  M3  S1000  Drilling (Z drill axis)
N30  G32  Z5  F1000  M4  S1000  Retraction (Z drill axis)

In the case of direct programming of the H thread pitch, the pitch, if smaller than 1, is to be programmed as follows:

- H.5 instead of H0.5 or
- H 0.5 instead of H0.5

Example:
N10  G0  X30  Y5  Z0  F1500  Positioning the axes
N20  G32  Z–20  M3  H.75  Positioning the axes
N30  G32  Z0  M4  H.75  Retraction (Z drill axis)

CAUTION
Possible damage to workpieces!

Drilling and retraction must always be programmed with identical thread pitch (F/S)!

- G32 acts block by block.
- Neither M19, nor M5 are required ahead of G32. Switch-over to C axis operation is done automatically. Prior to starting, the Typ3 osa internally waits for “INPOS” of all axes involved. In case an axis drifts out of its INPOS range, G32 will not be started (for INPOS range, please refer to machine parameters).
- The drilling and retraction blocks must be programmed directly one after another, otherwise the “Retraction block not programmed” runtime error will appear.
- After the retraction block, the spindle will return to spindle operation.

Please refer to section 4.23 for tapping using several spindles, G532, and to section 4.74 for suppressing axes for calculating the feedrate, G594.
4.23 Activation of tapping without compensation chuck for several spindles

**Effect**
You can tap threads in parallel without compensation chuck using up to 8 spindles. Programming and effect correspond to G32. However, you use G532 to determine the spindles to which G32 is to refer.

If you do not program any G532, G32 will always refer to the 1st spindle.

**Programming**

- **G532 CAX<i> ..{CAX<n>}** tapping (G32) using the i-th spindle(s)
- **G532 GRP<j>** tapping (G32) using (a) spindle(s) from the j-th spindle group
- **G532 GRP<j> .. CAX<i> ..{CAX<n>}** tapping (G32) using (a) spindle(s) from the j-th spindle group and additionally the i-th spindle(s)

**Explanation:**
- **CAX** spindle axis
- **i = 1 .. max. 8 (n)** number of the spindle
- **GRP** spindle group
- **j = 1 .. max. 4** number of the spindle group

**Example:**

- **G532 CAX1** tapping (G32) using the 1st spindle
- **G532 CAX2 CAX4 CAX7** tapping (G32) using the 1st, 4th and 7th spindle
- **G532 GRP2** tapping (G32) using (a) spindle(s) from spindle group 2
- **G532 GRP3 CAX4** tapping (G32) using (a) spindle(s) from spindle group 3 and the additional 4th spindle

**Please note for G532:**

- With G532 you cannot activate more than one spindle group.
- CAX1 through CAX8 can be combined in any way and in addition to any spindle group.

**Note:** The information will remain active until a new G532 is programmed (i.e. even after CONTROL RESET!). After a control reset, this function can be entered in the MACODA start-up string.
4.24 Retraction from tapped hole

Effect

If function "Tapping without compensation chuck" (G32) was cancelled (by "control reset" or due to a voltage drop) while the screw tap was still in operation, G9321 or G9322 can be used to retract the tap from the tapped hole. Programming can be done by manual input or within a part program (cycle).

Programming

G9321 Switches the spindle(s) running again in spindle speed mode after "control reset" (control start-up) to Position mode.

G9322 F Initiates the retraction motion proper (with F value).

G9321 must be programmed before retracting!

2 situations must be distinguished when applying the function "Retraction from tapped hole":

- retraction after "control reset"
- retraction after control start-up (power failure)

Retraction after "control reset" (automatic retraction)

Each time, a tapping process is started, the following data is stored:

- starting position of the 1st G32 block
- thread pitch
- direction of rotation
- spindles involved

This data is deleted as soon as the tapping process is finished. If tapping is cancelled by "control reset", the stored data can still be accessed. The spindles are switched over to position mode with G9321. If G9322 is programmed next, the tap moves out of the thread and on to the stored starting position. Only the desired feedrate (F-value) must be programmed together with G9322.

Retraction after power failure (manual retraction)

If the control is first switched off and then on again while the tapping function is active, the thread data stored internally is lost and the parameters must be programmed explicitly together with G9322:

G91 G9322 S<spindle speed> F<feedrate> M3/M4 <drill axis> <incremental path>

Preconditions for the use of G9321/G9322:

- The C axes involved must be defined as endless rotary axes. This concerns MACODA parameter 1001 0000 4 = 2 and the SERCOS position data weighting (S-0-0076=Ob1xxxxxxx)
- In SERCOS secondary operation mode 1 (S-0-0033), the bit for drive-controlled change of operation mode must not be set: S-0-0033=Ob000001011 or S-0-0033=Ob000001100.
Application of subprograms for retraction from tapping:

To make the application of the function "Retraction from tapping" easier, it is recommended that you write one subprogram (cycle), each, for automatic and manual retraction. With MACODA parameters 3090 0000 1 and 3090 0000 2, these subprograms can then be assigned to any G-code available.

Example: automatic retraction cycle

AutoTR[feedrate]

1
2 If P1=NUL THEN
3 (MSG, ** P1 NO FEEDRATE PROGRAMMED **) 
4 M0
5 GOTO N3
6 ELSE
7 GFEED% = SD (1,7,2)
8 FFEED% = SD (5,1,2)
9 G9321
10 G94 G9322 F(P1)
11 G[GFEED%]
12 IF GFEED% = 94 THEN
13 F[FFEED%]
14 ENDIF
15 ENDIF
M30

Example: manual retraction cycle

ManTR[drill axis number, path, master pitch, feedrate]

1
2 If P1=NUL THEN
3 (MSG, ** AXIS NOT PROGRAMMED **) 
4 M0
5 GOTO N3
6 ENDIF
7 If P2=NUL THEN
8 (MSG, ** P2 PATH NOT PROGRAMMED **) 
9 M0
10 GOTO N8
11 ENDIF
12 If P3=NUL THEN
13 (MSG, ** P3 THREAD PITCH NOT PROGRAMMED **) 
14 M0
15 GOTO N13
16 ENDIF
17 If P4=NUL THEN
18 (MSG, ** P4 FEEDRATE NOT PROGRAMMED **) 
19 M0
20 GOTO N18
21 ENDIF
22 BAXIS%=ROUND(P1)
23 IF P3>0 THEN
24 MCODE1%=4
25 ELSE
26 MCODE1%=3
27   ENDIF
28   PITCH=ABS(P3)
29   GABS_INC%=SD(1,4,2)
30   GFEED% = SD (1,7,2)
31   FFEED% = SD (5,1,2)
N32   G9321
M[MCODE1%]
N34   G[GABS_INC%] G[GFEED%]
35   IF GFEED% = 94 THEN
36   F[FFEED%]
37   ENDIF
M30

For the cycles to run, some configurations need to be set in the MACODA.

**MACODA configuration:**

Parameter: 3090 0000 1

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>9032</td>
</tr>
<tr>
<td>1</td>
<td>9132</td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

Parameter: 3090 0000 2

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>AutoTR</td>
</tr>
<tr>
<td>1</td>
<td>ManTR</td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

**Example of function call:**

G9032[1000] automatic retraction at a feedrate of F1000mm/min

G9132[3,-100,0.5,500] manual retraction at a pitch of 0.5, a retraction path of –100 mm, F500, with the 3rd logical axis of the channel as the drill axis
### Rounding of corners

**Effect**

The "corner rounding" function inserts tangential transition arcs between 2 linear blocks (G34, G134) and between circular and helical blocks (G134 only) in the principal plane. On the one side, this leads to a minor modification of the programmed contour at these corners, but on the other side, continual speed and acceleration patterns are achieved during interpolation.

The max. admissible deviation between the modified and the programmed contour can be preset in the MACODA parameter 705000110 and also changed in the part program.

**Programming**

- **G34:** Switch on "corner rounding" with max. admissible deviations
- **G35:** Switch off "corner rounding"
- **E address:** You use the E word to program the "max. admissible deviation" (in μm) between the modified and the programmed contour. Fractional parts are allowed. Programming "E" is only possible if G34 is active.
- **G36:** Deletes a max. admissible deviation programmed via E word. The value activated in the MACODA parameter 705000110 becomes active again.
- **G134:** Switch "corner rounding" on with specification of the rounding radius.
- **R address:** You use an R word to program the radius of the transition arc. Fractional parts are allowed. Programming "R" requires that G134 is active.
Please note for G134:
- the radius is programmed together with G134 in one block,
- the radius acts modally,
- in the case of helical blocks, only the components of the circular plane for rounding are taken into account.

With active G34, the control unit will not execute "corner rounding" if:
- at least one of the two neighbouring blocks is no linear block.

With active G34, G134 the control unit will not execute "corner rounding" if:
- at least one of the two neighbouring blocks has a path portion outside the selected principal plane, or
- at least one of the two neighbouring blocks has a traversing path which is smaller than the path set in the MACODA parameter 705000120 (2 to 90 μm, default value: 2 μm), or
- no "quasi-continual" block transition according to the definition in the MACODA parameter 705000130 is present. In 705000130 it is possible to state an angle between 0 and 45 degrees (default value: 1). If the angle between the two neighbouring programmed linear blocks is greater than the value in 705000130, the Typ3 osa does not assume the presence of a quasi-continual block transition.
4.26 Mirroring, scaling, rotating

**Effect**

**Mirroring:**
The control unit mirrors a programmed contour for machining. You do not need to change the programmed contour for this purpose.

**Scaling:**
The control unit scales a programmed contour up or down for machining. You do not need to change the programmed contour for this purpose.

**Rotating:**
The control unit rotates a programmed contour for machining. You do not need to change the programmed contour for this purpose.

The named functions can also be combined.

**Programming**
You can influence the mirroring, scaling and rotating functions jointly via the G37, G38 and G39 functions:

- **G37:** Determination of the mirror or rotation point
- **G38:** Activate the mirroring, scaling or rotating function
- **G39:** Deactivate the mirroring, scaling or rotating function

---

**CAUTION**
The functions act only in the automatic, single-block and manual-input modes. The traversing direction during jogging will not change with active G38.

For detailed explanations, please refer to the sections below.
4.26.1 Mirroring

G37, G38, G39

Effect

The control unit will machine a programmed contour or, for instance, a bore-hole pattern in the form of a mirror image. The “scaling” and “rotating” functions can be used simultaneously with “mirroring”.

Programming

G37: Pole definition (special case). This is used to determine the position of the “mirror point”. This position has to be input as an absolute pair of coordinates referring to the program zero point.

G37 is not required if

- mirroring is supposed to be referring to the program zero point
- rotating is supposed to be referring to the program zero point (please refer to “Rotating”).

The G37 function:

- acts modally. The pole values remain effective until G39 or G37 (together with other pole values) is programmed.
- acts only in combination with G38
- does not cause any axis traversing
- may be programmed together with other positional data in the same block; auxiliary functions are permitted
- is not influenced by the factors programmed in G38 (in the case of the “scaling” function) and by the angle of rotation (in the case of the “rotating” function).

Example:

```
N... G37 X100 Y–200
```

G38: Mirroring on.

You switch ”mirroring” on by programming axis addresses (e.g. X) with the value of “–1” in the same block as G38. By doing so you instruct the control unit to multiply all subsequently programmed path commands of the corresponding axis (e.g. X100) internally by the value “–1”.

This means that the ”mirroring” function is exclusively realized by the minus sign. If you specify a value other than “1”, you will furthermore change the size of the mirrored contour (please refer to “Scaling”).

Mirroring will become effective together with the next traversing information.

The G38 function:

- acts modally. It remains effective until G39 is programmed.
- must always be written into the same block with the axes to be mirrored
- may be programmed with other preparatory functions in the same block
- can be programmed with auxiliary functions
- takes account of interpolation parameters in the case of circular interpolation
- influences the programmable contour shift G60
- does not affect the zero shifts G54–G259, G92 (set actual value) or cutter-radius and tool-length compensation.

Example:

All subsequently programmed axis commands regarding the X and Y axis will be multiplied by the control unit by the value of "–1".

Switch mirroring on (together with the next traversing movement)

G39: Switch mirroring, scaling, rotation off.
Any subsequently programmed axis values will no longer be multiplied by the value of "–1" within the control unit.
The approached axis positions are retained until they are reprogrammed.

The G39 function:
- acts modally
- deletes all mirroring axes
- deletes G37 and G38 and sets the pole coordinates to the value of "0"
- can be written into a block with preparatory functions, traverse information and auxiliary functions.

Mirroring examples:
The control unit scales the programmed contour up or down for machining. This can be used in part programs for programming contours using one fixed size (standard size). Then, prior to calling such a part program (e.g. as a subprogram), you use scaling factors for each axis to determine the scale of the programmed contour. In this way it is, for instance, easy to compensate for the contraction of the workpieces in the manufacture of the forms for cast and forged parts.

The “scaling” function can be used together with “mirroring” and “rotating”.

- Scaling does not influence feed programming or the active feedrate.
- M2/M30 in a subprogram does not switch scaling off.

In principle, different scaling factors can be stated for each axis. However, if you wish to use circular interpolation (or helical interpolation) with active scaling, the scaling factors must be the same for all axes involved! Otherwise an error message will be generated.

Scaling factors will also change the I, J, K interpolation parameters as well as the amount of the R address (for radius programming).

### Effect

- G0, G1, G2, G3, G5, G10, G11, G12, G13, G73, G200 Scaling acts in combination with the programmed axis information.
- G20 Scaling acts in combination with the programmed axis information.
- G37 Pole values will not be scaled.
- G40, G41, G42, G43, G44 Scaling is active; compensation values will not be scaled.
- G54–G59, G154–G159, G254–G259 Zero-shift values will not be scaled.
- G60 The programmable contour-shift values will be scaled.
- G70, G71 Scaling acts independently of the active unit of measurement.
- G74, G76 Scaling is not active.
- G90, G91, G190, G191 Scaling acts with absolute- and incremental data input.
- G92 Offset will not be scaled.
If the starting position of the workpiece contour is to be uninfluenced by a scale-down or scale-up of the programmed contour, you should select the starting point of the workpiece contour to be the program zero point, too.

G38: Scaling on.
You switch "scaling" on by programming axis addresses (e.g. X) with a positive factor in the same block as G38. By doing so you instruct the control unit to multiply all subsequently programmed path commands of the corresponding axis (e.g. X10) internally by this value.

If you specify a value other than "1", you will change the size of the contour:
Factor > 1: the contour will be scaled up.
Factor < 1: the contour will be scaled down.
If you program the factor with a negative sign, you switch on the "mirroring" function in addition.

The G38 function:
- acts modally. It remains active until G39 is programmed.
- has always to be written together with the axes to be scaled within one block
- does not cause any axis traversing
- may be written with other preparatory functions in the same block
- may be programmed with auxiliary functions.

Example:

N...
G38 X3 Y0.5

All subsequently programmed X coordinates will be multiplied by "3", Y coordinates will be multiplied by "0.5"

Scaling on (with next traversing movement)

G39: Mirroring, scaling, rotating off.
Subsequently programmed axis values are no longer affected.
Approached axis positions are retained until re-programmed.
The G39 function:
- acts modally
- deletes all mirror axes
- deletes G38 and sets the internal scaling factors to the value of "1"
- can be written into a block with preparatory functions, traverse information and auxiliary functions.

**Scaling examples:**

- **Effect of:** G38 X2 Y2  
  (program zero point=workpiece zero point)

- **Effect of:** G38 X2  
  (program zero point=workpiece zero point)

- **Effect of:** G38 Y2  
  (program zero point=workpiece zero point)

- **Effect of:** G38 X0.5 Y0.5  
  (program zero point=workpiece zero point)

- **Effect of:** G38 X0.5  
  (program zero point=workpiece zero point)

- **Effect of:** G38 Y0.5  
  (program zero point=workpiece zero point)
4.26.3 Rotating

**Effect**

The control unit rotates a programmed contour in the active plane (please refer to G17, G18, G19 or G20).

This means that you must program only once such recurrent programming steps as are rotated around a specific angle.

In addition you do not have to convert the dimensions of angled workpieces to the machine coordinates; you simply take them over directly from a production drawing and specify the corresponding angle of rotation. The Typ3 osa will do the rest.

"Scaling" and "mirroring" can be used in conjunction with the "rotating" function.

**Programming**

G37: Pole definition.

This determines the position of the "center of rotation". This position has to be input as an absolute pair of coordinates referring to the program zero point.

G37 is **not** required if

- rotating is supposed to be referring to the program zero point
- mirroring is supposed to be referring to the program zero point (please refer to "Mirroring").

The G37 function:

- acts modally. The pole values remain effective until G39 or G37 (with other pole values) is programmed.
- is only effective in conjunction with G38
- does not cause any axis traversing
- may be programmed together with other preparatory functions within the same block; auxiliary functions are allowed
- is not influenced by scaling factors programmed in G38 (in the case of the "scaling" function) or their sign (in the case of the "mirroring" function).

**Example:**

N...G17 G37 X200 Y100

pole coordinates

pole-definition call

plane selection (may be omitted if rotation is supposed to be performed in the active plane)
G38: Rotating on.
Program the "R" address with the desired angle of rotation in the same block as G38.

- Positive values: Counter-clockwise rotation
- Negative values: Clockwise rotation

By doing so you instruct the control unit to rotate all subsequently programmed coordinates of the corresponding plane around the center of rotation (please refer to G37). The rotation becomes active with the next traversing information. A programmed contour shift (G60) will be included in the calculation of the coordinate rotation.

The G38 function:

- acts modally. It remains effective until G39 is programmed.
- must always be programmed with angle of rotation R in the same block
- may be programmed together with other preparatory functions in the same block
- can be programmed with auxiliary functions.

Example:

N... G38 R+30

Angle of rotation
Rotating on

<table>
<thead>
<tr>
<th>+Y</th>
<th>Y'</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>X'</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>+X</td>
<td></td>
</tr>
</tbody>
</table>

Point of rotation

Angle of rotation R: +R: mathematically positive
- R: mathematically negative
G39: Switch mirroring, scaling, rotating off.

Any subsequently programmed coordinates will no longer be rotated.
Approached axis positions are retained until re-programmed.

The G39 function:

- acts modally
- deletes G37 and sets the angle of rotation and the coordinates of the center of rotation to the value of "0"
- can be written into a block with preparatory functions, traverse information and auxiliary functions.

Rotation examples:

<table>
<thead>
<tr>
<th>Effect of:</th>
<th>G38 R45</th>
<th>Effect of:</th>
<th>G38 R–45</th>
</tr>
</thead>
<tbody>
<tr>
<td>(no G37 programmed = center of rotation corresponds to program zero point)</td>
<td>[Diagram]</td>
<td>(no G37 programmed = center of rotation corresponds to program zero point)</td>
<td>[Diagram]</td>
</tr>
</tbody>
</table>

- 1: definition of the center of rotation (X10;Y13)
- 2: rotating on

Effect of: G37 X10 Y13

- G38 R45

Effect of: G37 X10 Y13

- G38 R–45
4.26.4 Combining mirroring, scaling and rotating

If rotating and mirroring or scaling are programmed simultaneously, rotating will be executed first, followed by mirroring or scaling.

**Example:** Rotating + Mirroring + Scaling

N... G37 X100 Y–200 determination of center of rotation and mirror point

N... G38 X–3 Y–2 R115 counter-clockwise rotation by 115 degrees;
mirroring produced by the minus sign, and multiplication of the X and Y coordinates by "3" or "2"

N... G39 (switch all off)

4.26.5 Relationship between G37/G38 and G60 or G54..G259

Within the program coordinate system, G37/G38 is influenced by G60:

**Example:** G60

N5... P1: current position

N10 G60 P2: G60 shift of P1

N20 G38 X2 Y2 Scaling ON

N30 G1 X10 Y10 P3: scaled position of P2

As a principle, zero shifts (e.g. G54..G259) will shift the entire program coordinate system with regard to the machine coordinate system. Therefore they do not cause any change to the operations within the program coordinate system, triggered, for instance, by G37/G38 or G60:
Example: G54
N10  G54
N20  G37  X10  Y10
N30  G60  X10
N40  G38  R90
N50  G1  X10  Y10

Example: G55
N110 G55
N120 G37  X10  Y10
N130 G60  X10
N140 G38  R90
N150 G1  X10  Y10

Comment
Call-up function
P1: center of rotation on X10 Y10
P2: G60 shift of P1
P3: coordinate rotation of P2
4.27  **Cutter path compensation**  

**G40, G41, G42**

*Effect*

The cutter path compensation function causes the tool to move along an equidistant path parallel to the programmed path during execution of a partspecific program. (Equidistant path = path with right-angled, constant distance to the programmed contour.) The distance between the equidistant and the programmed path depends on the value of cutter-path compensation.

The following illustration shows the principle:

![Cutter path compensation illustration](image)

Along a contour element and in the case of tangential contour transitions, the equidistant – and thus the cutter path – is uniquely defined by the programmed contour.

![Tangential transition illustration](image)

This is not the case at **unsteady contour transitions in the case of outer corners**. In this case the control unit has to calculate a path independently in order to combine the equidistants of the contour elements involved. To do so, you can instruct the control unit via G68 or G69 how to treat such transitions (please refer to page 4–63):
At unsteady contour transitions in the case of inner corners the control unit will use the intersection of the equidistants to determine the required path. With some contour patterns (e.g. indentations) this principle may lead to contour damage. To avoid damage to the contour, the Typ3 osa offers the “collision monitoring” function (please refer to page 4–142).

![Diagram](image)

\[ t = \text{tangential transition} \]
\[ u = \text{unsteady transition} \]

**Programming**

**G40:** Cutter-path compensation off. G40 is the power-up state. In the same block as G40, it is furthermore possible to program a linear traversing movement for axes lying in the active plane. In this case the control unit will deactivate the compensation on the way to the target position. If no traversing movement is programmed, the compensation will be deactivated on the spot – perpendicular to the last traversing block.

An active tool-length compensation will not be influenced by G40.

---

**CAUTION**

Incorrect programming of the G40 function may cause damage to the workpiece and the machine!

With active G2, G3 or G5, G40 may only be programmed without any traversing movement.

---

**Cutter-path compensation G41**

**G41:** Switching cutter-path compensation on to the left of the workpiece (seen in the direction of machining), referred to positive compensation values. To do so, you assign the required cutter-radius compensation via the D address or T address and programming of M06 (tool change).

In the same block as G41, it is furthermore possible to program a linear traversing movement for axes lying in the active plane. If a traversing movement was programmed, the compensation will be activated on the way to the target position and will stand perpendicularly on it. If no traversing movement is programmed, the compensation will be activated while the tool is in position – perpendicular to the last traversing block.
Example: Cutter-path compensation G41

N60 G41 X... Y... Z... D...

or

N60 T123 M06
N65 G41 X... Y... Z...

---

CAUTION
Incorrect programming of the G41 function may cause damage to the workpiece and the machine!
With active G2, G3 or G5, G41 may only be programmed without any traversing movement.

The following is not allowed with active G41 or G42:
- Plane change-over (G17 to G20)
- Inch/metric change-over (G70, G71)
- G32, G74, G75, G76, G92
- Programming the WAIT CPL command

---

Cutter-path compensation G42
G42: Switching cutter-path compensation on to the right of the workpiece. Apart from that, identical to G41.

Example: Cutter-path compensation G42

N60 G42 X... Y... Z... D...

or

N60 T123 M06
N65 G42 X... Y... Z...

💡 For additional information on activating and deactivating cutter-path compensation, please refer to section 7, "Tool compensation".
4.28 Zero shifts (ZS)

<table>
<thead>
<tr>
<th>ZS OFF</th>
<th>G53</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZS ON</td>
<td>G54-G59</td>
</tr>
<tr>
<td>1st additive ZS OFF</td>
<td>G153</td>
</tr>
<tr>
<td>1st additive ZS ON</td>
<td>G154-G159</td>
</tr>
<tr>
<td>2nd additive ZS OFF</td>
<td>G253</td>
</tr>
<tr>
<td>2nd additive ZS ON</td>
<td>G254-G259</td>
</tr>
</tbody>
</table>

Effect

Using the ZS, it is possible to shift the program zero point to any point in reference to the machine coordinate system.

The corresponding distance values are stored in the zero-shift tables. Each zero-shift table contains a maximum of 3 groups with 6 zero shifts, each (G54..G59; G154..G159; G254..G259).

For details about editing the ZS tables, please refer to the operating instructions.

To activate a ZS, you first select the desired ZS table (please refer to G22). Subsequently, you simply program the corresponding G instruction. If the G instruction is programmed without additional positional data, this will not result in a traversing movement.

Zero shifts from different groups always act additively to each other (e.g. G54 + G156 + G259).

Zero shifts within the individual groups overwrite each other.

Programming

Zero shift

N... G22 V1 (activate ZS table V1)
N... G54 (ZS activated; no traversing movement)
(or)
N... G54 X...Y...Z... (shift applies already to the position programmed here)
N...
N... G154 X...Y...Z... (1st add. ZS activated; with traversing movement)
N...
N... G254 X...Y...Z... (2nd add. ZS activated; with traversing movement)
N...
N... G253 (only 2nd add. ZS off)
N...
N... G53 (all still active ZS off)

Please note for G53, G54...G59:
- G54 to G59 act modally and cancel each other mutually.
- They are switched off by G53.
- G53 switches the 1st and 2nd ZS off, too.
- G53 will not influence a "programmed contour shift" (G60).

Please note for G153, G154...G159:
- G154 to G159 act modally and cancel each other mutually.
- They are switched off by G153 and G53.
Please note for G253, G254...G259:

- G254 to G259 act modally and cancel each other mutually.
- They are switched off by G253 and G53.

Please note for G354...G359:

- G354 to G359 (inclined plane) switch off G54..G59, G154..G159 and G254..G259.

Example:
ZS principle (in the example, the program and the workpiece zero point are identical)

The ZS function is not permitted in combination with the "inclined plane" function.
4.29 Programmed contour shift

Effect

G60 does not shift the program coordinate system with regard to the machine coordinate system, but only the contour within the program coordinate system (please refer to section 4.63).

From this program block on, all programmed coordinates will be shifted correspondingly. If no coordinate rotation is active (please refer to G38, "rotating"), the programmed shift values act directly additively with other compensations.

G60 does not cause any traversing movement.

CAUTION
Incorrect programming may cause damage to the workpiece and the machine!

G60 is influenced by G38 (mirroring, scaling, rotating), i.e. the coordinates of the new zero point which were programmed in the G60 block will also be mirrored, scaled or rotated!

Programming

G60: Programmed contour shift on
G67: Programmed contour shift off

Example: Programmed contour shift

<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N10</td>
<td>G60</td>
<td>X10 Y10 Z50</td>
<td>new program zero point in X10 Y10 Z50; no axis traversing in this block</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>axis traversing taking the shift values into account</td>
</tr>
<tr>
<td>N100</td>
<td>G1</td>
<td>X... Y... Z...</td>
<td>axis traversing without taking the G60 values into account</td>
</tr>
<tr>
<td></td>
<td>G67</td>
<td>X... Y... Z...</td>
<td>or</td>
</tr>
<tr>
<td>N210</td>
<td>G60</td>
<td>X20 Y20</td>
<td>reset of G60</td>
</tr>
</tbody>
</table>

When reprogramming G60, any axis not reprogrammed in the process will retain its previously active contour shift values.

Example: programmed contour shift (repeat programming of G60)

<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N10</td>
<td>G60</td>
<td>X10 Y10 Z50</td>
<td>new program start at X10 Y10 Z50; no traversing of axes in this block</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>traversing of axes with shift values taken into account</td>
</tr>
<tr>
<td>N100</td>
<td>G60</td>
<td>X20 Y20</td>
<td>new program start at X20 Y20 Z50 (Z shift is retained as programmed previously in G60!); no traversing of axes in this block</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>traversing of axes with shift values taken into account</td>
</tr>
<tr>
<td>N120</td>
<td>G1</td>
<td>X... Y... Z...</td>
<td>reset of G60</td>
</tr>
</tbody>
</table>
4.30 Exact positioning

**Effect**
During the control of a tool movement, a timely offset between the set and the actual values of the individual axes occurs owing to the dynamics of the machine. This 'lag effect' causes a following distance error during machining, the size of which depends on the feedrate speed and the KV factor (axis dynamics). In the case of unsteady contour transitions (corners) this following distance error becomes noticeable in the form of a “slurring” of the corner.

This effect can be avoided using G61. Via the G164 to G166 functions, three different exact positioning options can be set.

G61 acts only on movements at feedrate (for exact positioning at rapid travel, please refer to G161/G162).

The action of G163 is overriding and affects movements both at feedrate as well as at rapid (G0, G200). G163 supersedes G161/G162.

**Programming**
G61: Exact positioning at feedrate on.
G62: Exact positioning off.
G163: Switch exact positioning on at feedrate and at rapid travel.

Please note for G61, G62 and G163:
- G61, G62 and G163 act modally. M2/M30 sets the power-up condition.
- G61, G62 or G163 must be programmed at the latest within the block to which the respective function applies.

Example: Programming of G61/G62
N10 G61 no movement; exact positioning ON
N11 G1 Y200 interpolation with exact positioning
(or)
N10 G62 interpolation without exact positioning
N11 G1 Y200
N50 G61 X200 interpolation with exact positioning as early as in this block
4.31 Feedrate 100%  

Effect You use program control to influence the function of the feedrate potentiometer (for feedrate and rapid travel). Both functions are of effect in the “manual data input” and “automatic” operating modes.

Considering that the G63 function also sets the "G63" output signal, the spindle potentiometer can be influenced as well. To do so, you must link the "G63" output signal to the "spindle override 100%" input signal.

Programming

- **G63**: feed 100% on. Deactivates the feedrate potentiometer. The feedrate is set to 100% of the programmed value irrespective of the position of the feedrate potentiometer.
- **G66**: feed 100% off. Activates the feedrate potentiometer. The feedrate depends on the position of the feedrate potentiometer.

Please note for G63 and G66:
- Both functions are modal and cancel each other mutually.
- G63 and G66 can also be programmed together with other preparatory functions.
- M2/M30 sets the power-up condition.

Example: Programmed “feed 100% ON”
N ... G63 G1 X120.675 Y34.896 Z–34.765 F200 S1000 M04
4.32 Feed compensation:
Cutter contact point
Cutter center point

Effect
You define whether the Typ3 osa is supposed to keep the programmed feed constant
• either at the cutter contact point (cutter intersection path)
• or at the cutter center-point path.

Programming
G64: The control unit keeps constant the feedrate $F_B$ along the intersection path. This calculation is only possible when $G41/G42$ are active for arcs G2/03/05.

G65: The control unit keeps the $F_M$ feedrate constant along the cutter center-point path.

G64 should be used only for finish milling because the feedrate speed can increase considerably along circular contours.

Please note for G64 and G65:
• G64 and G65 act modally and cancel each other mutually.
• The power-up state can be set via machine parameters.
The effective feedrate in the case of a **compensating circle** in combination with cutter-path compensation depends on the point where an F word is programmed:

**Example 1:**
N10  G64 X100  F100  
N20  Y100  F200  
The compensating circle is traversed at F100

**Example 2:**
N10  G64 X100  F100  
N20  F200  
N30  Y100  
The compensating circle is traversed at F200

**Example 3:**
N10  G64 X100  F100  
N20  Z50  
N30  Y100  F200  
The compensating circle is traversed at F100
4.33 Contour transitions:  
Arc  
Intersection

Effect

With active cutter-path compensation (G41, G42), the control unit calculates a transition between two contour elements. In the case of outside angles, this transition can be in the form of an automatically generated arc (G68) or an intersection of the equidistants (G69).

G68: Arc

The path gap is closed by a tangential arc with the radius $r$:

![Diagram of G68 arc transition](image)

G69: Intersection

The control unit tries to close the path gap by way of a determination of the intersection of the two equidistants.

1st case: An intersection exists.

Depending on the "$A$" distance between the "KE" contour corner and the "S" intersection, the control unit will proceed as follows:

![Diagram of G69 intersection transition](image)

For $A \leq \sqrt{2} \times r$ the two equidistants will be extended into the intersection:

In the case of greater distances the control unit will cut off the peak at a distance of $A = \sqrt{2} \times r$ and close the gap by a straight line:
2nd case: An intersection does not exist.

This case can occur in the case of an unsteady contour transition between a straight line and an arc or between 2 arcs. In such a case, as with G68, the path gap will be closed by a tangential arc with the radius of \( r \):
4.34 Inch programming

Effect
Positional and feed data after G70 must be entered in inches. All effective metric values and zero shifts are automatically converted to inches. G70 is modal and deletes the G71 function.

Programming
Programming G70

N... G70 from here on all positional and feed information is interpreted in terms of inch
N... X... Y... Z... all path and feed data to be entered in inch units

4.35 Metric programming

Effect
Positional and feed data after G71 must be entered in metric measures. All effective inch values and zero shifts are automatically converted to metric values. G71 is modal and deletes the G70 function.

Programming
Programming G71

N... G71 from here on all positional and feed information is interpreted in terms of metric units
N... X... Y... Z... all path and feed data to be entered in metric units

4.36 Linear interpolation with exact positioning

Effect
In contrast to G1, a G73 block is always executed with exact positioning – irrespective of G61/G62. The exact positioning option is globally determined via the G164 to G166 functions.

Programming
G73 X... Y... Z... F....

Please note for G73:
- G73 may be programmed with or without positional data.
- G73 has to be programmed together with an F word before any feed is active.
- The programmed feedrate remains effective until overwritten by a new one.
- G73 deletes G0, G1, G2, G3, G5, G10–G13 and G200.
4.37 Approach reference point

**Effect**

The axes – programmed in the same block as G74 – traverse simultaneously and at rapid to the reference point(s).

With G74 neither reference point cams nor markers are taken into account. G74 is purely a positional procedure (i.e. applies to axes with distance-coded encoders) for the absolute axis positions.

---

**CAUTION**

Possibly active compensations will remain unconsidered in the course of this positioning process!

---

- G74 is effective only on a block-by-block basis.
- When the reference point is approached with G74 the axis actual values are not set or reset, i.e. the programmed shift values are not affected.
- G74 is deleted when the machine axes programmed with G74 in the block have reached the reference point.
- Any offsets still active, ZS etc. are not taken account of in the G74 block for the programmed axes.
- G74 does not set the interface signal 'RAPID'.

**Programming**

G74 is programmed in a separate block with the axes to be traversed. The axis addresses must be programmed together with a numerical value (e.g. “X0”, “Y0”, “Z0”). The numerical value does not affect the reference point position; it is used only to complete the word.

Auxiliary and additional functions may be programmed within the same block.

**Example: Programming G74**

N100 G74 X0 Y0 Z0 The X, Y and Z axes approach the reference point positions simultaneously.
### 4.38 Traverse to reference point

<table>
<thead>
<tr>
<th>Effect</th>
<th>The function &quot;Traverse to reference point&quot; via G374 allows to initiate axis referencing in the part program.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>In a G374 block, the reference point can be approached by one or several axes. For this purpose, the SERCOS command &quot;Drive-controlled referencing&quot; (S-0-0148) is given within the NC for each of the programmed axes. The respective response of the individual axes is determined under the SERCOS ID No. of reference point parameter S-0-0147 in the respective SERCOS configuration file.</td>
</tr>
<tr>
<td>Programming</td>
<td>G374 &lt;axis address1&gt; &lt;integer&gt;...&lt;axis address n&gt; &lt;integer&gt;</td>
</tr>
<tr>
<td>Example:</td>
<td>N..</td>
</tr>
<tr>
<td></td>
<td>N.. G374 X1 Y1 Z1 referencing of axes X, Y and Z</td>
</tr>
<tr>
<td></td>
<td>N..</td>
</tr>
<tr>
<td>Please note for G347:</td>
<td></td>
</tr>
<tr>
<td>• For axis with absolute values measured, you can define in reference point parameter S-0-0147, whether these axes shall reference to the set reference point or not.</td>
<td></td>
</tr>
<tr>
<td>• If several axes have been programmed in one block, these axes will approach their respective reference points simultaneously, but independently of each other (no continuous-path operation!).</td>
<td></td>
</tr>
<tr>
<td>• With G374, you can program both synchronous and asynchronous axes. Block processing is suspended until all axes have traversed to their respective reference points (implicit WAIT).</td>
<td></td>
</tr>
<tr>
<td>• For matters of compatibility with the Typ1 osa control, a numeric value must be entered behind the respective axis address which, however, will not be evaluated. (The approach to the reference point is configured statically via ID No. S-0-0147.)</td>
<td></td>
</tr>
<tr>
<td>• There is no difference between the axes traversing to their reference points initiated this way and the operation mode &quot;Manual&quot;:—Activate traversing to reference point—.</td>
<td></td>
</tr>
</tbody>
</table>
**Effect**  
The control unit drives the measuring axis at feedrate in the direction of the position programmed via G75 and checks in this process whether the probe was triggered.  
The axes for which the measuring probe function G75 is to be activated are entered in MACODA parameter 100300012.

As soon as the edge defined with machine parameters is recognised (probe comes into contact with the measuring surface; the edge evaluation can be set via MACODA parameter 100300011: all axes involved must have identical edge), the control unit performs the following:
- the actual position is stored
- initiates an axis stop with Down Slope function
- deletes the difference
- deletes G75 (effective block-by-block)

**This function should only be used in combination with a CPL program (e.g. with measuring cycles).**

### Programming

Programming G75

N100 G75 X400

**Please note for G75:**
- G75 must be programmed together with at least one axis position. This value represents the maximum search depth to which the probe must have switched at the latest.
- No auxiliary functions may be programmed in the G75 block. G75 can be programmed with other preparatory functions as far as required.

**You do not have to program WAIT in the part program.**

### Evaluation of G75

Continuation of program, evaluation of axis information, safety monitoring, generation of error messages etc. must be realised in the CPL program.

**Example:** Measuring program

N100 G75 Y250 F500  
Approach the contour to be measured (at F500)

110 IF SD(9)=0 THEN  
Query whether probe was displaced

120 YPOS=PPOS(2)  
Store the position in the 'YPOS' variable upon switching of the 2nd axis (Y axis)

N130 (MSG, PROBE DISPLACED)  
140 GOTO N180

150 ENDIF

N160 (MSG, PROBE NOT DISPLACED)  
N170 M0  
Program stop

N180 ...
4.40 On-the-fly measurement  

**Effect**

With the “On-the-fly measurement” NC functionality it is possible to use the SERCOS measuring-probe function for measurements without cancelling the programmed movement. Possible collisions are avoided by a suitable measuring probe (contactless or suitable mechanical design).

The function acts as follows:

The probe logic of the drive has to be initialized via G175 first (one time after each SERCOS phase startup). Subsequently, the initialized drive (= physical axis) is ready to recognize and store measuring (actual) values.

G275 is newly programmed for each measurement.

The measuring probe traverses on a programmed path to its end point. Via the probe signal, generated by a mechanical positioning or a contactless system, the actual value of the predefined axis is recorded and stored in the drive (probe logic in the drive). **In contrast to G75, a successful measurement will neither cause a cancellation nor a ramping-down (decelerating to standstill) of the movement.**

**Example:**

Path movement on X and Y direction: actual-value recording of the 1st physical axis = X axis; for further details, please refer to MACODA parameter 100300001

```
N... G1 G275 MpiAxis 1 X100 Y100
```

**Initialization:**

```
G175 MpiAxis <i>
```

Explanation:

**G175:** **Initialization** of the probe logic in the SERCOS drive. It is called once following the SERCOS startup via G175.

**MpiAxis** probe axis parameter

<i> index of the physical axis the actual values of which are to be taken
Starting the measuring cycle:
G275 MpiAxis <i> <axis 1> ..<axis n>

Explanation:
G275: starting the measuring cycle proper
MpiAxis probe axis parameter
i index of the physical axis the actual values of which are
to be taken with G175/G275
axis 1 ... n probe positioning axes

Please note for G175 and G275:
- G175 and G275 act block by block.
- G175 and G275 act in parallel with the active interpolation.
- G175 and G275 can be programmed together with all interpolation types.
- If no measurement is made at G275 during interpolation, the system will
  wait at the block end until the measurement is completed (to be ended by
  program cancellation).
- The measuring position can be inquired via PPOS (CPL).
- PPOS (CPL) will supply for linear modulo axes the measuring value re-
  ferred to the respectively valid zero point.
- The information to know whether a measurement was carried out can be
  inquired via SD(9) (CPL).
- The block used to further process the probe information should be pre-
  ceded by a WAIT (CPL) to avoid that further blocks are edited (50 or more
  blocks in the Typ3 osa).
- As an alternative to WAIT, the number of edited blocks can be determined
  in advance using the PREPNUM function.

For further details about the CPL commands, please refer to the CPL manual
available on this topic.

Since both functions involve communication via the SERCOS service
channel, you must ensure that the programmed synchronous travers-
ning paths are sufficiently long so that SERCOS communication can be
completed within the course of interpolation (at present several
100 msec). If communication was not complete, or if no measurement
was performed, the speed will sharply drop at the end of the block and
the control unit will wait for the cancellation of SERCOS communica-
tion.
WAIT may be waived if, for instance, ”creeping errors” are to be cor-
rected and the main focus is not on the latest measuring results.
4.41 Switching NC blocks via high-speed signal

Function G575 allows early jumping to the next block via the high-speed inputs (HS signal) of the DC/IO.

The traversing blocks in a program are programmed beyond the actual contour. As a result, the end position of an NC block switched by an HS signal is never actually reached. Instead, the motion is measured "on-line", i.e. continuously, and the block end is reported by an HS signal. The NC block currently being executed is cancelled and the next one is executed.

This function is linked with NC blocks with linear motion.

A linear motion (G0, G1, G10, G11, G73, G200) can be terminated early via a high-speed input signal by programming function G575, which acts block by block.

With G575, there are two options for changing blocks early:
- on-the-fly change of blocks: early block change without changing the programmed end points but with changed geometry,
- change of blocks with abortion: early block change with cancellation of the remaining distance to go.

4.41.1 On-the-fly change of blocks – G575 HSx=y

Whenever the voltage level y is reached at HS input x, an early change is made to the next block, which must also contain a linear motion. Blocks are changed without checking the maximum velocity step change.

With the function "On-the-fly change of blocks", the change from one block to the next is made at the current velocity.

Please note the exceptions described for operation mode "Automatic" and the special features in operation modes "Program Block" and "Manual Data Input".

Programming

G575 HS<x>=<y>

Explanation:

<table>
<thead>
<tr>
<th>x</th>
<th>1..8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>0..1</td>
</tr>
<tr>
<td>HS1–HS8</td>
<td>designate the HS input at the DC/IO module.</td>
</tr>
<tr>
<td>HSx=0 or HSx=1</td>
<td>designates the voltage level required for changing blocks:</td>
</tr>
<tr>
<td>HSx=0</td>
<td>(for 0V)</td>
</tr>
<tr>
<td>HSx=1</td>
<td>(for 1–24V)</td>
</tr>
</tbody>
</table>
Example: On-the-fly change of blocks – G575 HSx=y

Depending on external events, up to 3 different feedrates are to be used for traversing.

The end points of the programmed blocks must be different because the respective next block (from the point of view of the part program) must contain a path to be traversed.

```
N20 G0 X0 Y0
N30 G575 G1 X70 Y7 F1000 HS1=1 X axis traverses at F1000 until HS input No. 1 is “High” or X70 Y7 is reached.
N40 G575 X90 Y9 F900 HS1=0 X axis traverses at F900 until HS input No. 1 is “Low” or X90 Y9 is reached.
N50 X100 Y10 F800 The remaining distance to go to X100 Y10 is traversed at F800.
```

DANGER
The function “On-the-fly change of blocks” only changes the contour, not the programmed end point of an NC block.

See the following example:
Example: Changes in contour caused by "On-the-fly change of blocks"

```
N05  G1   F100 X0 Y5   end point: X0 Y5
N10  G575 HS1=1 G90  X50   end point: X50 Y5
N20  G575 HS1=0 G91  Y10   end point: X50 Y15 (abs. Y5+incr. Y10)
N30  G91  Y5   end point: X0 Y5 (abs. Y15+incr. Y5)
```

Operation mode "Automatic"

Early changes of blocks are usually made without deceleration to zero axis velocity.

Exceptions:
- Change of blocks at 0 velocity due to a knee in the contour: The change from one block to the next is made at zero velocity if there is a knee > 90 degrees in the contour between the block to be cancelled and the next block.
- Change of blocks at 0 velocity due to G function:
  - The block marked with G575 must end at 0 velocity (G0, G10, G73).
  - Exact positioning function is generally activated (e.g. G61, G161, G163).
  - The next block must begin with 0 velocity due to additional information programmed (e.g. G14, G15, G114, G115).
  - Block-by-block interpolation with path shape is active (G408, G608).

End point information of axes not programmed in the next block is always taken from the cancelled block.

Operation modes "Single block" and "Single step"

In these operation modes, just one block of a part program at a time is executed. Because this precludes an on-the-fly change of blocks, the velocity is reduced to 0 when the external event for a block change occurs.

The end points of axes not programmed in the next block are taken from the cancelled blocks. This allows testing the "behavior in the Automatic operation mode" by approximations also in the operation modes "Single block" and "Single step".
Operation mode "Program block"

In this operation mode, each block of a part program is executed as if it were a complete part program. Because this precludes any next blocks, the velocity is reduced to 0 when the external event for a block change occurs.

---

**DANGER**
The remaining distance to go at the end of the block is cancelled. Any next NC block with incremental programming will result in traversing to an incorrect end point.

---

Operation mode "Manual data input"

In this operation mode, the velocity is reduced to 0 when the external event for a change of blocks occurs. The same applies when a part program is executed with manual data input.

---

**DANGER**
The remaining distance to go at the end of the block is cancelled. Any next NC block with incremental programming will result in traversing to an incorrect end point.

---

4.41.2 Change of blocks with cancellation – G575 HSx=y HSSTOP=z

If the level y is reached at the HS input x, the velocity in the current block is reduced to 0 (HSSTOP=0) or a velocity step change is made (HSSTOP=-1) and the remaining distance to go is cancelled (same behavior as with G75).

The end point of any NC block and thus also the starting point of the next NC block are determined by the occurrence of an external event.

**Programming**

G575 HS<x>=<y> HSSTOP=<z>

**Explanation:**

- x: 1..8
- y: 0..1
- z: -1 or 0

**HS1–HS8** designate the HS input at the DC/IO module.

**HSx=0 or HSx=1** designate the voltage level required for a change of blocks:
- HSx=0 (for 0V)
- HSx=1 (for 24V)

**HSSTOP=-1** designates the cancellation with a velocity step change to 0.

**HSSTOP=0** designates the cancellation with a velocity down-slope to 0.
Example: Change of blocks with cancellation

| N05 | G1 F1000 X0 Y5 | end point: X0 Y5 |
| N10 | **G575** HS1=1 HSSTOP=0 **G90** X50 | end point: external event at X ≤ 50 |
| N20 | G575 HS1=0 HSSTOP1=0 **G91** Y10 | end point: external event at Y ≤ 10 |
| N30 | **G91** Y5 | end point: incremental by Y5 from the last occurrence of external event (Y ≤ 10) |

In blocks N20 and N30, the Y axis traverses by increments. The end point of the Y axis results from the current position of the Y axis upon cancellation of the previous block to which the programmed incremental step is added.

With the function "Change of blocks with cancellation", any NC block ends at 0 velocity (even if the programmed event does not occur). Additionally, the remaining distance to go in the aborted block is cancelled.

Because the starting point of the next NC block is unknown after a cancellation, this has to be a linear block.

Example:
Traversing to a dead stop (e.g. pressure- or torque-driven)

| N20 | G1 F1000 X1=0 X2=0 |
| N30 | **G575** F10 X1=10 HS1=1 HSSTOP=–1 |
| N40 | **G575** F200 X2=100 HS2=1 HSSTOP=0 |
| N50 | G0 X1=0 X2=0 |
4.42 Traverse to machine-oriented absolute axis position

Effect

G76 is used to traverse to a machine-oriented absolute axis position (absolute position referring to the machine coordinate system) e.g. to change tools, check tools for damage, run measuring cycles, change pallets, etc.

G76 acts only block by block. Prior to any traversing movement, the control unit deselects various compensations and cancels some active functions block by block:

- length compensations (Hxx)
- radius compensations (G41, G42)
- zero shifts (G54 . . . G259)
- inclined plane (G352, G354 . . . G359)
- incremental data input (G91)
- set actual value (G92)
- mirror function (G38)
- workpiece position compensation (G138)

Subsequently, the programmed axes travel simultaneously at active speed (at feedrate or at rapid travel!) to the programmed machine position.

If deleted by G76, the following functions or compensations, respectively, will become active again in the next block:

- length compensations (Hxx)
- zero shifts
- incremental data input (G91)
- set actual value (G92)
- mirror function (G38)

Programming

G76 is programmed together with positional data

Please note for G76:

- G76 can be written together with other preparatory functions (e.g. G93, G94, G95, G0, G1)
- G76 can be written together with the F word
- G76 is effective in conjunction with G93, G94, G95 and the F word
4.43 Drill axis switching

Effect
Drilling can be performed independently of machine kinematics using any machining axes.

With the G78.. function, Drill axis switching, you can activate the drill axes and their geometry compensation data assigned in the MACODA to the pertinent logical axes either in the part program or in the manual data input mode.

Length compensations of drill axes are organized by 2 compensation groups as follows:

1st compensation group refers to axes to which
- H compensations or
- H_{ext}/L3 compensations of the “External tool compensation” functions (G145–G845) apply.

2nd compensation group refers to axes to which
- L1, L2, L3 compensations of the “General tool compensation” functions (G147–G847) apply.

Programming
G78: Activation of drill axis switching

G78 axis name i <compensation_i> .. {axis name n <compensation_n>}

Explanation:
axis names i..n i.. n\textsuperscript{th} logical axis address

\[
\text{compensation}_i \pm 1 \text{ or } \pm 13 \quad 1\text{st compensation group (H, H_{ext}) of the } i\text{th axis} \\
\pm 21 \quad 2\text{nd compensation group, 1st length compensation (L1_{ext}) of the } i\text{th axis} \\
\pm 22 \quad 2\text{nd compensation group, 2nd length compensation (L2_{ext}) of the } i\text{th axis} \\
\pm 23 \quad 2\text{nd compensation group, 3rd length compensation (L3_{ext}) of the } i\text{th axis}
\]

The + or – sign before the compensation defines the direction of the compensation. This ensures that the proper direction is taken into account in the computation of length compensations. When double-digit compensation values are entered, the 1st digit stands for the compensation group and the 2nd digit for the compensation index.

Compensations relating to several axes can be switched simultaneously in a single G78 block.

G79: Deactivate drill axis switching

G79 {<CG> i}

Explanation:

CG: With G79, you have the option of resetting either all compensation groups to the default setting as defined in MACODA parameter 705000420, or just one selected compensation group (CG i, i=1, 2).
**Default settings** in MACODA parameter 705000420:

- **H**: 3rd logical axis address (usually the Z axis)
- **L1**: no axis assigned
- **L2**: no axis assigned
- **L3**: no axis assigned

Please note for G78 and G79:

- G78 and G79 act modally.
- The axis address under G78 refers to the workpiece coordinates.
- G79 may be programmed with other preparatory functions, traversing information or auxiliary functions.

**Examples:**

- **G78 X–1**: drill axis switching to X axis, negative direction of length compensation
- **G78 Y1**: drill axis switching to Y axis, positive direction of length compensation
- **G78 YA21 YB22**: drill axis switching to YA and YB axes, positive direction of L1 and L2 length compensations
- **G79**: drill axis switching OFF
  The 3rd axis becomes the active axis again (e.g. the Z axis). YA and YB are no longer assigned to any drill axes (default setting).
The boring cycles are programmed in CPL. The assignment between the individual CPL program names and the G functions to be used as well as the number of the transfer parameters are entered as a standard in the MACODA parameter blocks 309000005 to 309000007 and must not be changed.

For further details about the parameter blocks, please refer to the MACODA manual.

The FEED HOLD interface signal also acts in boring cycles. If required, it has to be blocked by appropriate PLC programming for the duration of a cycle.

For further information, please refer to the "ICL700 Project Planning Manual".

Effect

If you program a boring cycle (G81 to G86 or G184) including the required boring parameters within a program block, the control unit will automatically perform the currently active boring cycle upon reaching the programmed position in the same or in all following blocks, if a machining axis has been programmed there.

General movement sequence of a boring cycle:

1. Positioning in the active plane at rapid
2. Infeed to the programmed R1 reference plane at rapid
3. Infeed to the programmed Z drilling depth at the active feed
4. Waiting over a programmed P dwell time (the dwell time serves to compensate for a possibly required deceleration or acceleration to command speed of the spindle in the reversing point)
5. Retraction movement at feed or rapid to the programmed R1 reference plane
6. Optional approach of the R2 reference plane at rapid

All boring cycles can be performed in positive (Z > R1) or negative (Z < R1) boring direction.

The active boring axis can be switched over using G78 (please refer to page 4–77).
The optionally programmed R2 reference plane may also be located underneath the R1 reference plane. Please note, however, that the distance between the two reference planes is always traversed at rapid. The tool should therefore not be in contact with the workpiece within this range.

Boring-cycle overview:

<table>
<thead>
<tr>
<th>Preparatory function</th>
<th>Machining cycle</th>
<th>Infeed to drilling depth at</th>
<th>Action when drilling depth is reached</th>
<th>Retraction movement to reference plane 1</th>
<th>Application examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>G80</td>
<td>no</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>delete boring cycles</td>
</tr>
<tr>
<td>G81</td>
<td>yes drilling 1</td>
<td>feed</td>
<td>spindle turning (dwell time)</td>
<td>rapid</td>
<td>drilling, boring</td>
</tr>
<tr>
<td>G82</td>
<td>yes drilling 2</td>
<td>feed</td>
<td>spindle turning (dwell time)</td>
<td>feed</td>
<td>facing, centering</td>
</tr>
<tr>
<td>G83</td>
<td>yes deep-hole drilling</td>
<td>stepwise feed</td>
<td>spindle turning (dwell time)</td>
<td>rapid</td>
<td>deep-hole drilling</td>
</tr>
<tr>
<td>G84</td>
<td>yes tapping 1</td>
<td>working feedrate</td>
<td>spindle reversion (dwell time)</td>
<td>feed</td>
<td>tapping with comp. chck</td>
</tr>
<tr>
<td>G85</td>
<td>yes boring 1</td>
<td>feed</td>
<td>spindle hold (dwell time)</td>
<td>rapid</td>
<td>boring 1</td>
</tr>
<tr>
<td>G86</td>
<td>yes boring 2</td>
<td>feed</td>
<td>spindle hold (dwell time)</td>
<td>feed</td>
<td>boring 2</td>
</tr>
<tr>
<td>G184</td>
<td>yes tapping 2</td>
<td>working feedrate</td>
<td>spindle reversion</td>
<td>feed</td>
<td>exact tapping comp. chck</td>
</tr>
</tbody>
</table>

Programming

You specify parameters directly following the corresponding G function (except for G80). The number of parameters depends on the selected boring cycle. The order of the individual parameters is not arbitrary.

All the parameters must be programmed between the "[" and "]" signs and separated from each other by commas.

You can program parameters either as numerical values or as variable names. If you are using variable names, the variables must have valid values at the latest at the time the block is being processed.

Overview of the parameters used:

G80: N... G80 Switch off active boring cycle
G81: N... X... Y... G81 [Z,R1,P,R2] G81 on
G82: N... X... Y... G82 [Z,R1,P,R2] G82 on
G83: N... X... Y... G83 [Z,R1,K,k,P,R2] G83 on
G84: N... X... Y... G84 [Z,R1,P,R2] G84 on
G85: N... X... Y... G85 [Z,R1,P,R2] G85 on
G86: N... X... Y... G86 [Z,R1,P,R2] G86 on
G184: N... X... Y... G184 [Z,R1,P,R2,GS,U1,U2] G184 on
Some parameters may, but do not have to be specified.

The following applies:

- The parameter value as such may be omitted, but the neighbouring commas must be written.
  Example: G81 [Z,R1,,R2] (P missing)

- Commas ahead of the “]” sign are omitted.
  Examples: G81 [Z,R1] (P and R2 missing) or G81 [Z,R1,P] (R2 missing)

For details about the meaning of the individual parameters and the movement patterns of each boring cycles, please refer to the sections below. The Z, R1, P and R2 parameters, however, were already introduced to you in the figure on page 4–79.

The following applies in general:

- “Control reset” always cancels active boring cycles. “M02” or “M30” cancels active boring cycles only if the value “G80” is also entered in MP 706000020. If you wish to switch an active boring cycle off, you should at best program G80.

- No expression between brackets must be programmed in a G80 block.

- In the case of more than one program word within one and the same line the parameters must be programmed directly following the boring-cycle call:
  Correct: N10 G55 G81 [...]  Wrong: N10 G81 G55 [...]  

- The desired boring position within the positioning plane has to be in the cycle-calling block or after it and may also contain a positioning of the boring axis in infeed or retraction direction. The last position of the boring axis prior to programming a boring cycle, however, has to be on the positioning plane.

- G80 has to be programmed ahead of a change of the boring cycles.

- The entire cycle has to be called for a change of active parameter values.

- Boring cycles may be used both with active absolute data input (G90) as well as with active incremental data input (G91). Please note, however, that the transition parameters will be interpreted differently:

![Diagram showing G90 (absolute) and G91 (incremental) with positioning plane, reference plane 1, drilling depth, feed, rapid, and programmed values for R and Z.]
### 4.44.1 Boring cycle

**G81**

**Application**
Centering and simple drilling operation, facing, boring.

**Effect**
Upon reaching the Z drilling depth – depending on the programming – a dwell time becomes active. Subsequently the retraction movement will be performed at rapid.

**Programming**
N100 X... Y... G81 [Z, R1, P; R2]
Z, R1 must be programmed
P, R2 may be programmed

---

### 4.44.2 Boring cycle with retraction movement

**G82**

Just as G81. The retraction movement to R1, however, is at feedrate.

**Programming**
N100 X... Y... G82 [Z, R1, P; R2]
Z, R1 must be programmed
P, R2 may be programmed
4.44.3 Deep-hole drilling cycle

**Application**
Deep-hole drilling with complete removal of the drilling chips.

**Effect**
After each arrival at the programmed K infeed depth per cut, one retraction movement at rapid to the reference R1 plane will be performed.

Renewed infeed to the programmed k distance (speed-change point) will also be performed at rapid. Subsequently, the Typ3 osa will switch back to feedrate.

Stepwise infeed with corresponding retraction to the reference plane will be repeated until the programmed Z total drilling depth is reached.

---

**Programming**

```
N100  X... Y... G83  [Z, R1, K, k, P, R2]
```

Z, R1, K, k must be programmed

P, R2 may be programmed

The K infeed depth has to be programmed without sign in incremental dimensions, irrespective of the drilling direction.

If the Z max. drilling depth is exceeded owing to erroneous programming of the K infeed depth, the control unit will first interrupt the boring cycle via M0 and display the "K DRILLING DEPTH TOO LARGE" error message.

After a new start, the boring cycle is cancelled (M30).
4.44.4 Tapping with compensation chuck

**Application**

Tapping (left and right) with compensation chuck.

Precondition: One internal spindle has to be used as drilling axis. External spindles are not allowed.

**Effect**

Tool infeed is done at programmed M3 cw rotating spindle or M4 ccw rotating spindle (right- or left-handed thread).

As soon as the Z drilling depth (thread depth) has been reached, the sense of rotation is reversed, and the P dwell time (if programmed) starts to run.

Subsequently, the retraction movement to the reference plane is performed at feedrate. As soon as it has been reached, the reversal of the sense of rotation is cancelled again.

**Programming**

N100 X... Y... G84 [Z, R1, P, R2]

Z, R1 must be programmed

P, R2 may be programmed

---

**CAUTION**

Possible damage to tools or workpieces!

During the cycle, any active single-block processing will not be suppressed! This means that the spindle will keep on running after a positioning process within the cycle. This may lead to damage to the tool and the workpiece. You should therefore ensure that the control unit executes the cycle in the AUTOMATIC mode only!
4.44.5 Tapping without compensation chuck

Application

Exact tapping without compensation chuck (left and right).

Precondition: controlled spindle; G32 exact tapping without compensation chuck.

Effect

Tool infeed is calculated internally on the basis of the product of "speed x thread pitch". You select the sense of rotation (right- or left-handed thread) via the sign of the GS parameter (thread pitch).

As soon as the Z drilling depth (thread depth) is reached, the sense of rotation is reversed. Subsequently, the retraction movement to the reference plane is performed at feedrate. The sense of rotation of the spindle remains effective until you program a new boring cycle.

Programming

N100 X... Y... G184 [Z,R1,P,R2,GS,U1,U2,RP*] right-handed thread

N100 X... Y... G184 [Z,R1,P,R2,–GS,U1,U2,RP*] left-handed thread

Z, R1, GS, U1 must be programmed

R2, U2, RP may be programmed

*) Optionally, it is also possible to program the RP parameter. RP determines the orientation position of the spindle.

For matters of compatibility, P may be assigned a value. However, P dwell times programmed are not evaluated any more!

The syntax entered would read as follows:

N100 X... Y... G184 [Z,R1,,R2,–GS,U1,U2,RP*] (example: left-handed thread)
**4.44.6  Boring**

**Application**
Boring

**Effect**
Upon reaching the Z drilling depth, the spindle stops. Depending on programming, a dwell time will start to run. Subsequently, retraction at rapid will take place.

**Programming**
N100  X...  Y...  G85  [Z, R1, P, R2]
Z, R1 must be programmed
P, R2 may be programmed

---

**4.44.7  Boring with retraction movement**

**G86**

Like G85. However, retraction to R1 is at feedrate.

**Programming**
N100  X...  Y...  G86  [Z, R1, P, R2]
Z, R1 must be programmed
P, R2 may be programmed
4.44.8 Programming examples

Example 1: General programming.

```
N90 G1 M3S1050 F400 cycle call without positioning
N91 G81 [-1000,-800] drilling starts from this block
N92 X600 Y800
N95 X500 Y700 G81 [-1000,-800] Cycle call with positioning. From this position, drilling is already being performed
N96 X600 Y800 retraction to R2 plane; no dwell time
N100 X800 Y700 G81 [-1000,-800,,-600] retraction to R1 plane only; no dwell time
N110 X0 Y0 G81 [-1000,-800] retraction to R1 plane only; dwell time 1s
N111 X-100 Y-500
N150 X-400 Y200 G81 [-1000,-800,1] retraction to R1 plane only; dwell time 1s
N151 X200 Y300
```

Example 2: Programming the boring-cycle parameters via CPL variables.

```
N5 X200 Y400 M3
10 Z=1000 Definition of the CPL variables
20 R1=800
30 P=2
40 R2=900
N50 X... Y... G84 [Z,R1,P,R2]
```

Example 3: Calling the boring cycle in the main program. The positions to be approached have been programmed in a subprogram.

```
N05 X100 Y100 Z200
N10 G91 G81 [100,10]
N20 P1000 subprogram
N30 G80
N40 G90
N50 X500 Y100 Z200
N10 G91 G81 [100,10]
N20 P1000
N30 G80
N40 G90
```

Example 4: drill axis switching; X axis with positive compensation

```
N10 G78 X1
N20 G1 M3S1050 F400
N30 G81 [-1000,-800]
N40 Y500 Z700
N50 G80
N60 G79
```
Workpiece drawings may be dimensioned in absolute or relative (incremental) dimensions.

The Typ3 osa may be set in both formats. The two variants of absolute programming differ as to the treatment of endless axes (modulo axes) correspondingly configured via the machine parameter 100300005.

**G90:** The control unit will interpret dimensions as **absolute values** referring to the active zero point. Endless axes, entered as "can be switched over via G90/G189" in the MP 100300005, will traverse with sign logic (also ref. to section 4.59).

**G91:** The control unit will interpret dimensions as **incremental values** referring to the position last approached (relative or incremental dimension).

**G189:** The control unit will interpret dimensions as **absolute values** referring to the active zero point. Endless axes, entered as "can be switched over via G90/G189" in the MP 100300005, will traverse with "shortest-path" logic (also ref. to section 4.59).

The following diagram illustrates the difference between G90 and G91:

Please note for G90, G91, and G189:

- All functions act modally, cancel each other mutually and are included in one group with G190, G191.
- While G90 is active, the length compensation is added to or subtracted from – depending on the sign before the correction value – any new displacement entered for the spindle axis when the H word is called with the next position data for the spindle axis. While G91 is active, the compensation value is taken into account only in the computation of the first displacement.
Programming

G90, G91

N10 G90 all subsequent dimensions will be interpreted as absolute values referring to the active zero point.

N20 X100 Y100 current machine position: X100 Y100

N30 G91 all subsequent dimensions will be interpreted as relative values referring to the position last approached.

N40 X50 Y10 current machine position: X150 Y110

Via G190 and G191 it is possible to identify dimensions word by word as absolute or relative dimensions, respectively. For further details on the positioning types of endless axes, also refer to G150/G151.

Local absolute / incremental data input

With G90/G189 or G91 active, using the AC and IC address attributes, it is possible to program individual axes block by block in absolute or incremental terms.

- **AC(...):** the programmed axis value is to be treated as absolute.
- **IC(...):** the programmed axis value is to be treated as incremental.

Programming

<logical axis address> = <address attribute>(<value>)

X = AC (50) Irrespective of the presetting via G90/G189 or G91, the X axis will travel to the absolute position 50 (referred to the current coordinate system).

Within one block it is possible to program different attributes for different axes.

**Example:**

G91 X=AC(50) Y50

X: X travels to the absolute position X=50
Y: Y travels by increments of 50 mm (to Y=60)

Local incremental data input using the IC address attribute in the context of function G76, "Traverse to machine-oriented absolute axis position” is not permitted because it would lead to a runtime error.
4.46 Set actual value

Effect

The action of G92 differs depending on the programmed axis information:

Programming G92 without axis information:
The current actual value of all axes is set to machine coordinates without taking into account compensations and zero shifts.

Programming G92 with axis information:
The current actual value of an axis is set to the programmed value.

CAUTION
This option must not be used as long as a ZS is active. If necessary, you must program the G53 G instruction ahead of G92.

No axis movement will occur with G92. The new position values will be displayed.

Programming

N... G92 X0 Y0
N... G92
N... G92
N... G92

The current actual values of the X and Y axes are set to “0” (point-of-reference shift). The actual values of the Z axis remain unchanged.

Set all machine parameters to machine coordinates cancel point-of-reference shift).

Please note for G92:
• G92 acts block by block.
• Other functions may be programmed jointly with G92 in the same block provided these functions do not require an axis address.
• Whether a G92 shift is to be deleted after a control reset or whether this shift is to be retained can be entered for each channel in MACODA parameter 705000510. If the preset option remains unchanged, the shift values will be deleted upon control reset.

Example:
4.47 Time programming

Effect
The control unit will interpret subsequent F words as machining time for a programmed linear (G1) or circular (G2, G3, G5) distance.

The same applies in the case of polar coordinate programming.

Programming

Example: Time programming

N10 G93 G1 X300 Z400 A50 B120 F60  The programmed linear interpolation lasts 60 seconds.

Please note for G93:

• acts modally

• remains stored internally in the case of a switch-over to G94 or G95 and becomes active again if G93 is selected again

• The power-up condition can be set in machine parameters 303000001 and 303000002.

Upon "power off, "control reset" or "reset", the machining time as set in machine parameters 706000020 and 706000010 will become active (default value = F0)!
In addition, please keep taking machine parameter 800400001 into account!

The Typ3 osa will calculate automatically the required feed on the basis of the path length of the block and the programmed machining time.
This feed, however, can be restricted according to the programmed path and the max. values of the axes involved!
**Effect**

The control unit will interpret subsequent F words as feedrate in mm/min (with active G71).

**You can adapt the unit of measure to your specific requirements in machine parameter 704000010!**

The limit of the F word depends on the max. value of the feedrate of the axis parameters involved.

**Programming**

**Example:** Feedrate programming in mm/min

N10  G1 G94 X200 Z300 F200 programmed feedrate 200 mm/min  
N11  G4 F40 dwell time 40 seconds  
N12  X300 Z400 the 200 mm/min feedrate is active again

**Please note for G94:**

- acts modally  
- **remains internally stored** in the case of a switch-over to G93 or G95 and becomes active again if G94 is selected again  
- The power-up condition can be determined in machine parameters 706000020 and 706000010.

**CAUTION**

Upon "power off", "control reset" or reset", the feedrate as set in machine parameters 706000020 and 706000010 will become active (default value = F0)!
4.49 Incremental speed programming with acceleration adaptation

Effect

Using the G194 function it is possible to increase or decrease the active feedrate in increments. Within a block in which G194 was programmed the acceleration will be adapted in such a manner that the resulting speed will only be reached at the end of the block. This can be used to automatically achieve a very soft acceleration behavior. Existing limits regarding acceleration or deceleration values will be monitored (if necessary, the final speeds will only be reached in the next block).

Additionally, you can change the spindle speed by increments for a specific path. Within the block where this function is programmed, the spindle speed is adjusted linearly along the path so that the desired spindle speed is reached at the end of the block.

Any number of blocks containing G194 may be programmed. The programmed speed change will always refer to the existing speed of the preceding block.

Programming

N.. G194 F100 X.. Y.. Z.. The path speed will increase within the block by 100 mm/min.
N.. G194 F–50 X.. Y.. Z.. The path speed will decrease within the block by 50 mm/min.
N.. G194 S1=100 X.. Y.. Z.. The specified speed of the 1st spindle is increased within the block by 100 rev./min.
N.. G194 F100 S2=150 X.. Y.. Z.. Within the block, the path speed is increased by 100 mm/min and the specified speed of the 2nd spindle by 150 rev./min.

Please note for G194:

- The function is not modal. However, the resulting feedrate will have a modal effect on the subsequent blocks.
- The calculated acceleration acts only block by block.
- The unit of incremental feedrate corresponds to the F word programmed via G94.

If the program is cancelled within a G194 block, deceleration at the calculated acceleration will occur.
4.50  Feedrate programming in mm/rev  G95

Effect
The control unit will interpret all subsequent F words as feedrate in mm/rev.

You can adapt the unit to your specific requirements in machine parameter 704000020!

The main spindle must be activated before the first traversing movement is to be performed with G95. The main spindle is defined via machine parameter 702000010. With the main spindle running, the following axis data is interpolated for the feedrate in mm/rev in accordance with the F word programmed.

The limit of the F word depends on the max. feedrate values set in the parameters of the axes involved.

DANGER
Changing the feedrate may pose a risk to the machine and personnel!

Upon “power off”, control reset” or ”reset”, the feedrate as set in machine parameters 706000020 and 706000010 will become active (default value = F0)!

Considering that the current feedrate is derived from the spindle speed, the active feedrate will be influenced by both the spindle as well as by the feedrate potentiometer!

Programming

Example: Feedrate programming in mm/rev with dwell time

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N9</td>
<td>S2000 M4</td>
</tr>
<tr>
<td>N10</td>
<td>G1 G95 X200 Z300 F0.2</td>
</tr>
<tr>
<td>N11</td>
<td>G104 F4</td>
</tr>
<tr>
<td>N12</td>
<td>X300 Z400</td>
</tr>
</tbody>
</table>

Please note for G95:

- acts modally
- remains internally stored in the case of a switch-over to G93 or G94 and becomes active again if G95 is selected again
- The power-up condition can be set in machine parameters 706000010 and 706000020.
4.51 Direct speed programming

Constant cutting speed

Effect

An S word programmed with G97 causes a constant speed, irrespective of the position of the 1st axis.

An S word programmed with G196 causes a constant cutting speed. For this purpose, the Typ3 osa changes the speed of the axis entered in the machine parameter 701000110, “Reference axis for constant cutting speed”.

The spindle potentiometer also acts with active G196.

You can set a lower or upper limit of G196 and G97 using G192 and G292 (speed limitation).

A gear-range change has to be selected ahead of G196. If automatic gear-range recognition is active with G196 programming, the active gear range will remain active both during the entire machining at G196 as well as in the case of a change between G196 and G97.

Programming

G97: Determining the speed.

In the same block as G97, you program the spindle speed via the S word. If G97 precedes G196 in the beginning section of the program, or if a specific speed is supposed to be reached, the S word must be written.

The S word may be omitted if a change from G196 to G97 is made. In this case the Typ3 osa will take over the currently active speed.

G196: Determining the cutting speed.

In the same block as G196, you program the cutting speed via the S word in mm/min. If the cutting speed is supposed to be changed in the further course of the program, one just has to program the S word (without G196).

The S word may also be omitted if a change is made from G196 to G97 and back again. In this case the initially programmed with G196 cutting speed will apply again.

To determine the tool contact point, the tool length will be subtracted internally from the current value of the 1st axis. The Typ3 osa will take the tool length from the 1st additive ZS group (G154–G159).

Please note the following for G97 and G196:

- G97 and G196 act modally and cancel each other mutually.
**Example:** Behavior of the function with 2 spindles configured

G97  S... -and-  G196  S...

- **G196  S1=...**
  - Only the 1st spindle runs at constant cutting speed.
  - The 2nd spindle has speed programming.
- **G196  S2=...**
  - Only the 2nd spindle runs at constant cutting speed.
  - The 1st spindle has speed programming.
- **G196  S1=...  S2=...**
  - The 1st and 2nd spindle run at constant cutting speed.
- **G97 [S1=...  S2=...]**
  - Both spindles run under speed programming (the speed is calculated internally, unless programmed).
  - The last setting active under G196 is valid again.
4.52 Zeroseeting of modulo axis (linear endless axis)  G105

Effect
Using the G105 "modulo axis zeroseeting" function, the point of reference (program zero point) of a **linear endless axis** can be determined. As soon as the modulo value is reached, the actual value of the linear endless axis is automatically set to zero. This modulo calculation prevents an overflow of the axis values and enables the axis to travel at "endless".

G105 determines the program zero point. Using this point the control unit calculates the distance from the zero point of the command-value system. The resulting offset is internally added to all subsequent values.

Modulo value
The modulo value should be as long as possible (e.g. 20 m) in order to have a large programming range available. The take-over of the modulo value into the drive is done **as early as** at SERCOS startup via the S-0-0103 ID no. Changing the modulo value requires another SERCOS startup.

Traversing range
The control unit will **not permit any** programming of positions greater than the modulo value.

A linear endless axis can also travel backwards. Negative input is possible as long as the amount is smaller than the modulo value.

If an endless axis traverses with a negative value (e.g. X=–17), the end point will automatically be transformed into a positive X=3 end position as soon as it is reached.

Example: Linear endless axis with modulo value = 20 m

![Diagram](image)

Traversing in negative direction from X=17 to X=–17 (subsequently X=3) without setting G105!

The measurement-probe function (G175/G275) can be used for linear endless axes if the programmed positions have a positive sign. Backward travelling using the probe (programming of negative positions) will not provide unique values.
G105 is used to set the **program zero point** for all internal linear endless axes configured in the MACODA parameter 100300004.

G105 X..<br>Setting the program zero point and programming a traversing movement (e.g. X..) which refers already to the new zero point. One or more axes can be traversed.

G105 LinModAxis<physical axis index><br>G105 is used to set the **program zero point** only for the linear endless axis configured in the MACODA parameter 100300004 the **physical axis index** of which (1..n) is set.

G105 LinModAxis<physical axis index> X..<br>G105 is used to set the **program zero point** only for the linear endless axis configured in the MACODA parameter 100300004 the **physical axis index** of which (1..n) is set. One or more axes (e.g. X) can be traversed.

**Example:** Programming the linear endless axis

- N..G105 Setting the program zero point of all linear endless axes.
- N..G105 X200 Setting the program zero point of all linear endless axes, and traversing the X axis to position 200 following zerosetting.
- N..G105 LinModAxis1 Setting the program zero point of the linear endless axis with the physical axis index of 1.
- N..G105 LinModAxis1 X–200 Setting the program zero point of the linear endless axis with the physical axis index of 1, and traversing the X axis to –200.

**Please note for G105:**

- The workpiece position is always calculated in terms of modulo: 0 ≤ X < Xmod
- If the modulo range is exceeded, Xmod on the actual value display will jump to 0 or from 0 to Xmod. The setpoint value jumps only from 0 to Xmod because Xmod cannot be exceeded in the other direction.
- The stored axis offset is deleted upon control reset, i.e. the program zero point coincides with the axis zero point.
- The program value always indicates the position last programmed.

**For linear endless axes, MACODA parameter 100300004 must be set to 4.**
4.53 Consideration of the existing braking distance with active path slope

Effect
The "Consideration of the existing braking distance with active path slope" function looks ahead at the respectively following block and lowers the final speed of the current block of block preparation to such an extent that a speed of $V = 0$ can be reached at the end of the following block.

Programming
- G112: Deactivate braking-distance consideration.
- G113: Activate braking-distance consideration.

Please note for G112 and G113:
- Programming G112 requires an active G8. The G112 and G113 functions act modally and deactivate each other mutually.

In the case of short blocks, the restricted lookahead may cause a feedrate reduction although this may not be necessary from a geometry perspective.
### 4.54 Feed forward control

**Effect**

Contour errors are for the most part caused by system dependent lag. The current lag depends on the feedrate in the steady state of the axis, and in the acceleration phase on the acceleration as well.

The feed forward control will correct interpolator command values of the CNC in such a manner that the lag is reduced. This enables a more accurate contour to be achieved.

**The feed forward control function is integrated in the drive software according to the manufacturer’s specifications and is only activated by the Typ3 osa via SERCOS. For a detailed description of the function, please refer to the drive documentation.**

To know whether the drives being used are supporting feed forward control, please ask your system administrator. In the affirmative the function has to be released for the corresponding axes via MACODA parameter 100300009.

Upon activation, the drive will switch to “secondary mode 1” (ID no. S-0-0033; bit 3 set = position control excluding following error).

**Programming**

- **G114:** Activate feed forward control.
  - G114 programmed without axis addresses; all axes will be switched to secondary mode 1 if machine parameter 100300009 is set for these axes.
  - G114 X0 Y0 The programmed axes will be switched back to the main operating mode.

- **G115:** Deactivate feed forward control.
  - G115 programmed without axis addresses; all axes will be switched back to the main operating mode.
  - G115 X0 Y0 The programmed axes will be switched back to the main operating mode.

**The feed forward control parameters (e.g. P-0-0500 or P-0-0510 for Bosch drives) can be defined using function G900.**

**Example:** Programming the feed forward control

```
N10 G114 activate feed forward control of all axes
N20 F1000 S500
N30 G1 X1800 Y800
.
N160 X1500 Y1500
N170 G2 I50
N180 G114 Z0 deactivate feed forward control for Z
.
N210 G115 deactivate feed forward control of all axes
M30
```
DANGER
This programming might result in damage to the workpiece and/or the machine! There might even be danger to persons.

This programming refers directly to a real physical axis. A logical axis, addressed by a coordinate transformation (e.g. inclined plane) with the same axis address, will lead to incorrect axis values.
4.55  Tangential tool guidance ON
Tangential tool guidance OFF

Effect

Tangential tool guidance allows to approach a path on a selected plane with a tool axis at a specified offset angle with the path. The tool axis is in 0° position if it is set tangentially (offset angle 0°) to the principal axis traversing in a positive direction.

In the case of a circular path, the offset angle with the tangent to the circular path is calculated in accordance with the interpolator clock pulse. Consequently, the tool axis keeps turning by the respective offset angle calculated with each interpolator clock pulse.

In the course of the execution of all blocks or block segments, the tool axis reaches its full tangential angle at the starting point of the path (unlike in the case of the function "Tangential tool orientation", see section 4.76).

Depending on the “Adaptation angle” parameter, an intermediate block is automatically inserted between two NC blocks:

- If the angle at the contour knee is wider than the adaptation angle programmed, a tool rotation block (intermediate block) is automatically inserted, which rotates the tool axis towards the new starting tangent.
- If the angle at the contour knee is smaller than the adaptation angle programmed, the tool axis jumps to its new position at the beginning of the block.

Axis rotation with tangential tool guidance

Example:
adaptation angle: IA= 90°
Programming

G131: Tangential tool guidance ON

G131 \{TAX\{=\}<axis>\} \{SYM\{=\}<s>\} \{ANG\{=\}<a>\}
\{IA\{=\}<adapt. angle>\} \{PLC\{=\}<p>\}

Explanation:

TAX TAX (tool axis) is used for programming the axis which is to approach the path.

axis Designation of the axis to which the function "Tangential tool guidance" is to apply. You may enter either the logical axis name, or the physical axis name, or the logical axis number. Also CPL terms are permitted.

SYM SYM is used to enter the symmetry value of the tool (number of tool edges). A tool with the symmetry value s returns to its original position after a rotation of $360^\circ /s$.

s Symmetry value: any integer except 0.

s=1: The tool is asymmetric having 1 tool edge. The tool edge is run along the contour with the offset angle taken into account.

s>1: The tool is symmetric having several, equally spaced tool edges. If there is a knee in the contour, the tool is rotated just enough for the nearest tool edge to be positioned at the offset angle with the contour. "s" is dependent on the kind of tool being used, i.e. with a rectangular tool, s = 2, with a square tool, s = 4, etc.

s<0: a negative symmetry has the effect that the tool is not rotated if a reversal of direction of motion (180° knee) occurs, irrespective of the offset angle. In every other respect tool operation is the same as in the case of a positive symmetry.

ANG ANG is used for programming the offset angle.

a Offset angle \([-180^\circ .. 180^\circ]\): The offset angle indicates the angular offset between the path and the tool.

IA IA is used to program the adaptation angle (adapt. angle)

adapt. angle Adaptation angle \([0^\circ .. 180^\circ]\): The adaptation angle specifies from how many degrees upwards of a contour knee angle an intermediate block is inserted to rotate the tool axis. If the angle at the contour knee is smaller than the limit thus specified, no intermediate block is inserted to rotate the tool axis. Instead, at the start of the next block, the tool jumps to its new position.

PLC PLC is used to switch NC–PLC communication on and off while an intermediate block is being executed.

p p=0: NC–PLC communication is switched off while an intermediate block is being executed. The NC executes the rotation block unconditionally.

p=1: Execution of a rotation block is controlled via NC–PLC communication.
The programmed parameters may be omitted. In this case, they will be initialized by the following MACODA parameters:

- 705000210: number of the tool axis (TAX)
- 705000220: symmetry value (SYM)
- 705000230: adaptation angle (IA)
- 705000240: offset angle (ANG)
- 705000260: NC–PLC communication (PLC)

**Programming**

G130: Tangential tool guidance OFF

Please note the following for G130, G131:

- The functions "Tangential tool guidance" and "Tangential tool orientation" (G630, G631 or TTON/TTOFF) must **never** be active **simultaneously**.
- G131 does not produce a traversing motion after power-up.
- G131 must never be programmed together with an axis motion in the same block (error message!).
- Approach motions of tool axes programmed with G131 are executed only together with the next traversing movement to be carried out. Depending on the adaptation angle,
  - a rotation block is executed first, or
  - the tool jumps to its new position at the beginning of the next block.
- If no offset angle is entered when programming G131, please note the following:
  - the current rotary axis angle is taken to be the offset angle, or
  - the angle preset in MACODA parameter 705000250 is applied as the offset angle.

In MACODA parameter 705000250, you may select one of the above options.

**Syntax examples:**

- G131 Approach movements of all axes are executed with the SYM, ANG and IC initialization values of the MACODA.
- G131 TAX=C SYM1 ANG90 IA20 PLC0 Programming using logical axis names
- G131 TAX3 SYM=1 ANG90 IA20 PLC1 Programming using logical axis numbers
- G131 TAX[NAME$] SYM1 ANG=90 IA20 Programming using CPL variable
NC–PLC interface signals

The PLC is able to control the execution of the intermediate block if NC–PLC communication is active (PLC=1).

Channel output signal NC→PLC:

- **NCO 4.0, "G131, Tool rotation"**
  A signal is sent to the PLC indicating that the current angle between two blocks is wider than the "adaptation angle" (IA). The NC does not execute the intermediate block before it receives an acknowledgement from the PLC. The signal is not reset before the intermediate block is executed.

Channel input signal PLC→NC:

- **NCI 3.2, "G131, "Tool rotation release":1**
  The PLC signals the release of the execution of the intermediate block to the NC. After the execution of the intermediate block, the NC will not continue to execute any of the following blocks before the signal is reset.
4.56 Workpiece position compensation

**Effect**

The 'workpiece position compensation' function unlinks coordinates of part programs \( P \) totally from the machine coordinate system \( M \). This function acts on the **first 3** logical axes on the respective channel.

In contrast to the zero shift functions (please refer to G53–G59, G60), also the 1\(^{\text{st}}\) and 2\(^{\text{nd}}\) axes (standard axis addresses: \( X \) and \( Y \)) can be rotated here, i.e. they can be rotated around the 3\(^{\text{rd}}\) axis (standard axis address: \( Z \)). This allows you to adapt the coordinate system to any workpiece position.

In the course of the execution of a part program, all the programmed traversing movements will then be referring to the "new" – offset and rotated – coordinate system.

Workpiece position compensation is only possible in the valid working area of the machine and acts additively to active zero shifts!

The following diagram illustrates the principle:

![Diagram showing workpiece position compensation]

**Programming**

G138: Switch on workpiece position compensation.

At the beginning of the part program you program – in the same block as G138 – the

- **displacement** of the \( W \) workpiece zero point in the \( X \), \( Y \) and \( Z \) direction together with the corresponding axis address, as well as the
- **angle of rotation** of the 1\(^{\text{st}}\) and 2\(^{\text{nd}}\) axis (standard axis addresses: \( X \) and \( Y \)) as an R address (value range: \(-360^\circ < \text{angle of rotation} < 360^\circ\)).

All programming values must be **absolute machine coordinates**.

G139: Switch off workpiece position compensation.

**Please note for G138 and G139:**

- G138/G139 act modally and cancel each other mutually.
- If workpiece position compensation is active, G37, G38, G54 – G59, G154 – G159, G254 – G259, G60, G160 – G360, G168, G268, G145 – G845, G147 – G847 as well as tool-length compensation Hx will be taken into account.
- You can use the "Inclined plane" function, G352, G354..G359, to program "Workpiece position compensation". "Inclined plane" acts additively on the workpiece position compensation.
- G38, "Scaling", has no impact on the parameters of the inclined plane function.
The axis addresses under G138 refer to machine coordinates.

G138/G139 must never be programmed in combination with a traversing motion.

Programming G138/G139 will interrupt the look-ahead function. Therefore, G138/G139 must never be programmed while the cutter compensation function, G41/G42, is active. If required, workpiece position must be programmed before activating the cutter compensation function.

The current workpiece position is taken into account in the display of workpiece coordinates.

**Example:** Calling workpiece position compensation

```plaintext
N ... G90 G17 F1000 S250  ...  
N ... G138 X50 Y300 Z10 R1.23  Set workpiece zero point to machine coordinates X50 Y300 Z10 and rotate X/Y plane counter-clockwise by 1.23 degrees.  
N ...  
N ...  
N ... G139  Switch off workpiece position compensation.
```
External tool compensation

Effect
Activation of one out of 8 external compensation pairs for radius and length compensation. For this purpose, the respective compensation values must be imported from the PLC (application, e.g., for multiple compensation in the case of combination tools).

The effective radius or length compensation value is then equal to the total of any active geometry compensation table values plus the activated external tool compensation pair.

If the PLC changes the currently active external compensation values in the course of the execution of a part program, this change will only become active in the block being under preparation as the next block. Under certain circumstances this can mean that even more blocks are due to execution without this change.

In order to avoid this effect you must program the "WAIT" CPL command directly after the block causing the PLC to hand over the new compensation values. By doing so you hold the block preparation of the Typ3 osa until all program blocks ahead of "WAIT" have been executed.

Subsequently, in the program block following "WAIT", the new compensation values will already be active (please refer to Example 2).

Programming

G145..G845 External tool compensation on.
G146 External tool compensation off

Please note for G145 ... G845 and G146:

- G145...G845 / G146 act modally and cancel each other mutually.
- G145...G845 / G146 may be written together with other preparatory functions, axis information and auxiliary functions within the same block.
- G145...G845 / G146 will not cause any traversing movement if programmed alone in a block.

Example 1:
N ... G0  X0  Y0  Z0  
N ... H0 table length compensation OFF
N ... G146 external tool compensation OFF
N ... G1 table length compensation 1 ON
N ... X10  Y10  Z10 traversing movement with table length compensation 1
N ... G145 external tool compensation G145 ON
N ... X20  Y20  Z20 traversing movement with table length compensation 1 plus external tool compensation G145
N ... G345 external tool compensation G145 OFF and G345 ON
N ... X30  Y30  Z30 traversing movement with table length compensation 1 plus external tool compensation G345
N ... H0 table length compensation OFF
N ... X40  Y40  Z40 traversing movement with external tool compensation G345
N ... G146 external tool compensation OFF
N ... X0  Y0  Z0 traversing movement without compensation
Example 2:

N ... G145  external tool compensation G145 ON
N ... M... The M function will cause the PLC to perform the following cycle:
  1. Transfer new compensation values
  2. Send acknowledgement to CNC (the CNC will interpret this acknowledgement as "Block or subprogram with 'M' as having been executed").

10 WAIT Block preparation will be held until "M" is executed.

Length compensation will be in the direction of the selected drilling axis (please refer to G78/G79).

The active tool compensation is displayed at the "External tool compensation bit 0 ... bit 3" channel interface (please refer to Typ3 osa ICL700 Project Planning Manual).
4.58  General tool compensation  

**Effect**

General tool compensation is available as the 2nd external tool compensation function (2nd compensation group). This function may be used with drilling, milling, turning and anglehead tools.

General tool compensation may be activated in addition to and independent of the external tool compensation function (G145, G146, G245–G845).

The compensation data is stored in a compensation data set that may include the following parameters as a maximum:

- **L1, L2 and/or L3** (length compensation values and/or shift values)
- **R** (radius value)
- **SL** (tool edge position)

You can program the selection of one compensation data set from a total of 8.

**L1, L2 and L3 length compensation parameters and/or shift parameters:**

With a total of 3 shift values, L1, L2 and L3, you can perform both constant three-dimensional tool shifts and parallel length compensations of 3 different tools as a maximum.

**Example 1:** Three-dimensional tool shift

L1, L2 and L3 shift values are assigned to the respective axes via MACODA parameter 705000420 (see also G78, G79 in sect. 4.43). The control unit will check internally whether or not the assignments are correct.

<table>
<thead>
<tr>
<th>Shift parameters</th>
<th>Assignment to logical axis names</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1</td>
<td>X</td>
</tr>
<tr>
<td>L2</td>
<td>Y</td>
</tr>
<tr>
<td>L3</td>
<td>Z</td>
</tr>
</tbody>
</table>

![Diagram of tool compensation](image)
Example 2: Parallel length compensations of up to 3 different tools

Assignments, which will be internally checked for correctness, are to be entered in MACODA parameter 705000420 as follows:

<table>
<thead>
<tr>
<th>Shift parameter</th>
<th>Assignment to logical axis names</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1</td>
<td>Z1 (drill axis 1)</td>
</tr>
<tr>
<td>L2</td>
<td>Z2 (drill axis 2)</td>
</tr>
<tr>
<td>L3</td>
<td>Z3 (drill axis 3)</td>
</tr>
</tbody>
</table>

These axes must be different from the drill axis on which the external tool compensation function (G145 ...) or the length compensation function H act (external tool compensations and length compensations always act additively on one and the same axis).

Parameter R for radius compensation:

If the general tool compensation function is activated together with the external tool compensation function (G145 ... G845) in the same block, the radius set in the general tool compensation function will be applied invariably. (see Example 1 below).

While the cutter compensation function G41/G42 is active, the radius values set in general tool compensation (G147 ff.) and those set in external tool compensation (G145 ff.) automatically cancel each other, i.e. only the value that was activated last takes effect (see Example 2 below). The radius value that was activated last always acts additively on any D word (radius compensation) that may have been programmed.

The required compensation values must be specified by the PLC. Any changes in the currently active compensation values made by the PLC in the course of the execution of a part program will take effect only in the next block to be prepared. Therefore, it may happen that a number of blocks are executed before a change in a compensation value takes effect.

To prevent this, you must enter the CPL command "WAIT" directly after the block that causes the PLC to transfer the new compensation values. Block preparation by the Typ3 osa will be suspended until all program blocks before the "WAIT" command have been executed. In the first NC block after the "WAIT" command, the new compensation values will be taken into account.
Example 1:
N10 G147 G145 X.. Y.. The radius value set in the general tool compensation function will take effect!

Example 2:
N10 G147 The radius value set in the general tool compensation function is effective!
N20 G145 X.. Y.. The radius value set in the external tool compensation function will take effect!

SL parameter, tool edge position:
The SL parameter, tool edge position, determines the orientation of the tool edge relative to the Z axis.

Orientation of SL, tool edge position, relative to the Z axis:

Tool edge position 9 is used if the programmed contour describes the tool nose center.

Evaluation of tool edge positions not available at present!
Programming

G147..G847  general tool compensation ON
G148     general tool compensation OFF

Please note for G147 ... G847 and G148:

- G147 .. G847 / G148 act modally and cancel each other mutually.
- G147 .. G847 / G148 may be entered together with other preparatory functions, axis data or auxiliary functions in the same block.
- G147 .. G847 / G148 do not result in any traversing motion if programmed alone in an individual block.
- If the general tool compensation function and the external tool compensation function (G145 ... G845) are activated jointly in one and the same NC block, it is always the general tool compensation radius that takes effect. If the radius of the external tool compensation is to be taken into account, this function must be programmed separately in the next NC block.
- When active, the general tool compensation is displayed at the channel interface “General tool compensation”, bit 0 ... bit 3 (please refer to the Typ3 osa Project planning manual, ICL 700).
4.59 Changing the positioning type for endless axes

Local setting of the positioning type for endless axes

Effect

The Typ3 osa allows the positioning type for endless axes (type: rotary or endless) to be configured in a very flexible manner (please refer to G90/G189, G151/G150 and MACODA parameters 100300005, 100300050).

The following table shows various setting and switchover options for the positioning type of an endless axis:

<table>
<thead>
<tr>
<th>Presetting in MACODA parameter 100300005 (the way in which the endless axis translates the programmed value into a movement)</th>
<th>Presetting in MACODA parameter 100300005 (Switchover with G151: 1= yes 0 = no)</th>
<th>Switchover via G90/G189</th>
<th>Switchover via G150/G151 e.g. B axis&lt;..&gt;</th>
<th>Block-by-block switchover via DC(..), ACP(..), ACN(..) (applies only to linear interpolation G0, G1)</th>
<th>Axis traverses using positioning type:</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>no special logic</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>shortest path</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>sign logic</td>
</tr>
<tr>
<td>3</td>
<td>0 or 1</td>
<td>G90</td>
<td></td>
<td></td>
<td>sign logic</td>
</tr>
<tr>
<td>3</td>
<td>0 or 1</td>
<td>G189</td>
<td></td>
<td></td>
<td>shortest path</td>
</tr>
<tr>
<td>unequal to 3</td>
<td>1</td>
<td>G150: switchover according to presetting in 100300005</td>
<td></td>
<td></td>
<td>— no special logic or — shortest path or — sign logic</td>
</tr>
<tr>
<td>unequal to 3</td>
<td>1</td>
<td>G151 B0</td>
<td></td>
<td></td>
<td>no special logic</td>
</tr>
<tr>
<td>unequal to 3</td>
<td>1</td>
<td>G151 B1</td>
<td></td>
<td></td>
<td>shortest path</td>
</tr>
<tr>
<td>unequal to 3</td>
<td>1</td>
<td>G151 B2</td>
<td></td>
<td></td>
<td>sign logic</td>
</tr>
<tr>
<td>0, 1, 2 or 3</td>
<td>0 or 1</td>
<td>DC(..)</td>
<td></td>
<td></td>
<td>shortest path</td>
</tr>
<tr>
<td>0, 1, 2 or 3</td>
<td>0 or 1</td>
<td>ACP(..)</td>
<td></td>
<td></td>
<td>sign logic (in positive direction)</td>
</tr>
<tr>
<td>0, 1, 2 or 3</td>
<td>0 or 1</td>
<td>ACN(..)</td>
<td></td>
<td></td>
<td>sign logic (in negative direction)</td>
</tr>
</tbody>
</table>
Switching over the positioning type for endless axes

Programming

G151: Change position type.
You specify within the same block the axis address of the axis the positioning type of which is to be switched over. In addition to the axis address the desired positioning type is defined by the 0, 1 and 2 numerals:

0: no logic. The axis will subsequently always traverse without positioning logic to the respectively last programmed position.
1: shortest path. The axis will always use the shortest path for traversing to the respectively programmed position (traversing-movement is always smaller than 180 degrees).
2: sign logic: The programmed sign determines the sense of rotation of the axis, the numeric value defines the position.

G150: The positioning type is switched back to the state programmed in MP 100300005.

Example:

G151 B0 axis B: no logic.
G151 A1 C2 axis B: logic according to MP 100300005
axis A: shortest path
axis C: sign
G150 axes A, B and C according to MP 100300005

Please note for G150 and G151:

• G150/G151 act modally and cancel each other mutually.

Local setting of the positioning type for endless axes

With local setting of the positioning type for endless axes you have the possibility of determining or switching over the positioning type of an endless axis block by block and regardless of the set MACODA parameters and active (modal) NC functions.

Programming

DC(...): the programmed position is approached on the shortest path.
ACP(...): the programmed position is approached in mathematically positive direction.
ACN(...): the programmed position is approached in mathematically negative direction.

Note: mathematically positive direction = counter-clockwise sense of rotation seen from a coordinate axis in the direction of the coordinates origin.
Programming

<physical axis address> = <address attribute>(<value>)

B = ACP (258) Irrespective of the presetting by G150/G151, the B axis will traverse in mathematically positive direction to the position 258 degrees.

Please note for the ACP function:

- The address attributes only act block by block.
- For different endless axes it is possible to program different attributes within one block.
- The evaluation of the address attributes is only performed for endless axis. They are ignored for other axis-movement types.
- The positioning type of endless axes does only apply to G00, G01 linear interpolations (quasi-positioning mode). For other interpolation types, interpolation will include endless axes in analogy to a rotary axis.
- Only the amount of the axis value will be evaluated (negative sign will be ignored).
- The positioning type of endless axes will only be evaluated with absolute programming (G90).

DANGER
This programming of the ACP function might result in damage to the workpiece and/or the machine! There might even be danger to persons!

This programming refers directly to a real physical axis. A logical axis, addressed by a coordinate transformation (e.g. inclined plane) with the same axis address, will lead to incorrect axis values.
4.60 External zero shift

Effect

You can perform one out of max. 3 external zero shifts for each applied machining axis (= synchronous axis).

For this purpose the PLC has to default the corresponding values.

The **effective** zero shift will then correspond to the **total of**

- possibly active zero-shift values of the **ZS tables**.
- a possibly active **workpiece position compensation** and
- the activated **external zero shift**.

If the PLC changes the currently active external shift values in the course of the execution of a part program, this change will only become active in the **block being under preparation** as the **next** block. Under certain circumstances this can mean that even more blocks are due to execution without this change.

In order to avoid this effect you must program the "WAIT" CPL command directly after the block causing the PLC to hand over the new shift values. By doing so you hold the block preparation of the Typ3 osa until all program blocks ahead of "WAIT" have been executed.

Subsequently, in the program block following "WAIT", the new values will already be active.

Programming

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>G160</td>
<td>External zero shift no. 1 on.</td>
</tr>
<tr>
<td>G260</td>
<td>External zero shift no. 2 on.</td>
</tr>
<tr>
<td>G360</td>
<td>External zero shift no. 3 on.</td>
</tr>
<tr>
<td>G167</td>
<td>External zero shift off.</td>
</tr>
</tbody>
</table>

**Please note for G160, G260, G360 and G167:**

- G160, G260, G360, G167 act modally and cancel each other mutually.
- G160, G260, G360, G167 will not cause any traversing movement if programmed alone in a block.
- The compensation values under G160, G260, G360 refer to **machine or axis-coordinate values**.
- The active "external zero shift" will be displayed at the **External zero shift bit 0..bit 1** channel interface (also ref. to Typ3 osa ICL700 Project Planning Manual).
- G160, G260, G360 is **not allowed** in combination with the G352, G354..G359 "inclined plane" function.
4.61 Exact positioning at rapid travel  

Effect: During the control of a tool movement, an offset between the command and the actual values of the individual axes occurs owing to the dynamics of the machine.

In the case of positioning movements this effect has to be avoided if an accurate position is to be reached prior to the start of machining.

Using G161 you activate the "exact positioning" function especially for movements at rapid (for movement at feedrate, please refer to G61/62). The G164 to G166 G functions can be used to set 3 different exact positioning options.

Please note that G161/G162 are superseded with active G163!

Programming:

G161: Exact positioning at rapid travel on.
G162: Exact positioning at rapid travel off (only if G163 is not active).

Please note for G161 and G162:
- G161 and G162 act modally. M2/M30 sets the power-up state.
- G161 or G162 has to be programmed at the latest in the block to which the respective function is supposed to apply.

Example: Programming G161/G162

<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N10</td>
<td>G161</td>
<td>no movement; exact positioning ON</td>
</tr>
<tr>
<td>N11</td>
<td>G0 Y200</td>
<td>rapid travel with exact positioning (or)</td>
</tr>
<tr>
<td>N10</td>
<td>G162</td>
<td>rapid travel without exact positioning</td>
</tr>
<tr>
<td>N11</td>
<td>G0 Y200</td>
<td></td>
</tr>
<tr>
<td>N50</td>
<td>G161 X200</td>
<td>rapid travel with exact positioning as early as in the present block</td>
</tr>
</tbody>
</table>
4.62 Exact positioning mode

Effect

Using G164 to G166 you first determine the behavior of the “exact positioning” function. Subsequently, you activate “exact positioning” via the G functions:

- G61 (for movements at feedrate)
- G161 (for movements at rapid travel)
- G163 (for movements at feedrate and rapid travel).

Programming

G164: At the block end, the Typ3 osa reduces the path speed to V=0. It checks via the SERCOS interface whether the "positioning window fine" (SERCOS ID no.: S-0-0057) has been reached for all axes involved. For this purpose, the ID no. S-0-0336 is assigned to the real-time bit 2 S-0-0307. Only when this positioning window has been reached for all axes involved will the traversing movement of the next block be executed.

G165: At the block end, the Typ3 osa reduces the path speed to V=0. Subsequently, it checks via the SERCOS interface whether the "positioning window rough" (SERCOS ID no.: S-0-0261) has been reached for all axes involved. For this purpose, the ID no. S-0-0341 is assigned to the real-time bit 2 S-0-0307. Only when this positioning window has been reached for all axes involved will the traversing movement of the next block be executed.

G166: At the block end, the Typ3 osa reduces the path speed to V=0. Subsequently, the traversing movement of the next block is executed without a positioning-window check being performed.

Please note for G164, G165 and G166:

- G164, G165 and G166 act modally. M2/M30 sets the power-up state.
- As long as the 'positioning window rough' (G165) is selected, this is indicated at the Inpos range 2 activated channel interface (also ref. to Typ3 osa ICL700 Project Planning Manual).

The "positioning window fine" and "positioning window rough" parameters can be determined in the SERCOS files for Phase 3. For further details about the SERCOS files, please refer to the "Configuration parameters and MACODA parameter description" manual under "SERCOS initialization".
The following table shows the exact positioning behavior as a function of the different interpolation types:

- **G1, G2** linear and circular interpolation
- **G73** linear interpolation with exact positioning
- **G0** rapid travel with exact positioning (with deceleration to V=0)
- **G200** rapid travel without exact positioning (without deceleration to V=0)

as a function of the modal functions:

- **G61** exact positioning at feedrate
- **G62** exact positioning at feedrate off
- **G161** exact positioning at rapid travel
- **G162** exact positioning at rapid travel off (only if G163 is not active)
- **G163** exact positioning at feedrate and rapid travel
- **G164** positioning window, fine (V=0)
- **G165** positioning window, rough (V=0)
- **G166** without positioning window (V=0)

### InPos table:

<table>
<thead>
<tr>
<th>InPos</th>
<th>61</th>
<th>61</th>
<th>61</th>
<th>61</th>
<th>61</th>
<th>62</th>
<th>62</th>
<th>62</th>
<th>62</th>
<th>163</th>
<th>163</th>
<th>163</th>
<th>163</th>
<th>163</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rapid travel, InPos</td>
<td>161</td>
<td>161</td>
<td>161</td>
<td>161</td>
<td>162</td>
<td>162</td>
<td>162</td>
<td>162</td>
<td>161</td>
<td>161</td>
<td>161</td>
<td>162</td>
<td>162</td>
<td>161</td>
</tr>
<tr>
<td>InPos-window mode</td>
<td>164</td>
<td>165</td>
<td>166</td>
<td>164</td>
<td>165</td>
<td>166</td>
<td>164</td>
<td>165</td>
<td>166</td>
<td>164</td>
<td>165</td>
<td>166</td>
<td>164</td>
<td>165</td>
</tr>
<tr>
<td>G1, G2</td>
<td>164</td>
<td>165</td>
<td>164</td>
<td>165</td>
<td>166</td>
<td>165</td>
<td>166</td>
<td>164</td>
<td>165</td>
<td>166</td>
<td>164</td>
<td>165</td>
<td>166</td>
<td>164</td>
</tr>
<tr>
<td>G73</td>
<td>164</td>
<td>165</td>
<td>164</td>
<td>165</td>
<td>166</td>
<td>164</td>
<td>165</td>
<td>166</td>
<td>164</td>
<td>165</td>
<td>166</td>
<td>164</td>
<td>165</td>
<td>166</td>
</tr>
<tr>
<td>G0</td>
<td>164</td>
<td>165</td>
<td>166</td>
<td>166</td>
<td>166</td>
<td>164</td>
<td>165</td>
<td>166</td>
<td>166</td>
<td>164</td>
<td>165</td>
<td>166</td>
<td>166</td>
<td>166</td>
</tr>
<tr>
<td>G200</td>
<td>166</td>
<td>166</td>
<td>166</td>
<td>166</td>
<td>166</td>
<td>166</td>
<td>166</td>
<td>166</td>
<td>166</td>
<td>166</td>
<td>166</td>
<td>166</td>
<td>166</td>
<td>166</td>
</tr>
</tbody>
</table>

**Examples:** Using the table

The following exact positioning defaults are activated:

- **InPos** 62: G62 (deactivate exact positioning function at normal feedrate) is selected
- **rapid travel InPos** 162: G162 (deactivate exact positioning function at rapid feedrate) is selected
- **InPos window mode** 165: G165 (positioning window, rough (V=0)) is selected

The above InPos settings result in the behavior at block transition for the following active G functions:

- **G1,G2** - - - : No ramping-down (exception: max. axis step change)
- **G73** 165: reaching the "InPos window rough" is being waited for
- **G0** 166: ramping-down to speed V=0
- **G200** - - - : no ramping-down (exception: max. axis step change)

Despite G162 being active, ramping-down will be performed with G0. With regard to G0, G162 has the same effect as the combination of G161, G166.
4.63 Program coordinate shift
Additive program coordinate shift

Effect

All programmed coordinates of the feed axes (synchronous axes on a channel) of a part program or an MDI block refer to the program coordinate system (PCS or P). Therefore, a program zero point can be shifted relative to a freely defined workpiece zero (WCS or W).

Shifting the program coordinate zero point allows the execution of part programs without any changes to be made and at any position within the working range of the machine.

Additive shifting of the program coordinates allows to describe several successive coordinate systems and thus to design a part program that is equivalent to the dimensioning of a design drawing.

If one program coordinate shift is already active while a new one is being programmed, any axes for which no new values are entered will retain their previous shift values. This is the same behavior as with the "Programmed contour shift" function, G60.

The program coordinate shift function may be used in the context of all NC functions defining coordinate system transformations and in particular together with zero shifts (G54 ... G259) or the inclined plane function (G352 ... G359).

However, the program coordinated shift function must always be deactivated when an inclined plane function is activated or deactivated.

Difference between this function and "Programmed contour shift, G60/G67":

The functionality of "Program coordinate shift" and "Programmed contour shift", G60, is the same with the exception of their behavior in combination with function G38, "Mirroring, scaling, rotating":

Whereas shift values programmed with G60 are impacted by G38 (shift values are also scaled, mirrored and rotated), G38 has no effect on any shifts made with the program coordinate shift function. Unlike G60, the program coordinate shift function has the effect of shifting coordinate systems.
**Example:** G168 versus G60, both times in combination with G38 (scaling)

The following shifts result with the scaling factor=2:

<table>
<thead>
<tr>
<th>Axis</th>
<th>Scaling factor</th>
<th>Shift using G168 and G38</th>
<th>Shift using G60 and G38</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>2</td>
<td>△X=1 unit</td>
<td>△X=2 units</td>
</tr>
<tr>
<td>Y</td>
<td>2</td>
<td>△Y=1 unit</td>
<td>△Y=2 units</td>
</tr>
</tbody>
</table>

**Example:**

Programmed contour shift

<table>
<thead>
<tr>
<th>Block</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>N10</td>
<td>G168 X10 Y10 Z50</td>
<td>Setting program zero at X10, Y10, Z50 of the current workpiece coordinate system. There is no traversing motion included in this block.</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N100</td>
<td>G1 X... Y... Z...</td>
<td>Programmed positions refer to the program coordinate system defined above.</td>
</tr>
<tr>
<td>N110</td>
<td>G268 X20 Y10</td>
<td>Now, the program coordinate system is set at X30, Y20, Z50 relative to the workpiece coordinate system. There is no traversing motion included in this block.</td>
</tr>
<tr>
<td>N200</td>
<td>G169</td>
<td>The program coordinate system entered previously is deleted. Now, the program coordinate system is identical with the workpiece coordinate system.</td>
</tr>
</tbody>
</table>
4.64 Torque reduction

Effect

The positive edge of the TORQUE REDUCTION axis interface signal can be used to set the max. torque of an axis to a value defined in the machine parameter 1003000010. The TORQUE LIMIT output signal indicates that the reduced torque has become effective.

G177 offers you the possibility of overwriting the value as preset via machine parameter on an axis-per axis basis within the value range from 0 to 500 (0 to 50% of the standstill torque) using program control. The value of the machine parameter itself is not changed.

Programming

G177 X5 With the next positive edge of the “torque reduction” signal the max. torque for the X axis in the drive will be limited to 0.5% of the standstill torque.

G177 Y7 With the next positive edge of the “torque reduction” signal the max. torque for the Y axis in the drive will be limited to 0.7% of the standstill torque.

G177 With the respectively next positive edge of the “torque reduction” signal the max. torque for the individual axes will again be limited to the value determined in parameter 100300010.

When entering an invalid value the control unit will limit the input value to the admissible range (0 to 500) and output a warning.

In the case of a negative edge of TORQUE REDUCTION the max. torque will be set back to the value which was active in the drive after the last SERCOS phase startup. If this value is to be changed, it has to be entered in the corresponding SERCOS file under the SERCOS ID no. S-0-0092, and subsequently a phase startup has to be performed. For further details about the SERCOS files, please refer to the “Configuration parameters and MACODA parameter description” manual under “SERCOS initialization”.

DANGER

Programming the G177 function might result in damage to the workpiece and/or the machine! There might even be danger to persons!

This programming refers directly to a real physical axis. A logical axis, addressed by a coordinate transformation (e.g. inclined plane) with the same axis address, will lead to incorrect axis values.
4.65 Absolute with relative dimension

Effect

With G90 or G91, you can determine in a global manner whether the Typ3 osa is to interpret a dimensional statement in the part program in terms of an absolute or a relative dimension.

With active G190 or G191 this behavior can be modified word by word.

Programming

G190: You instruct the control unit to interpret all dimensional statements in the part program as absolute dimensions (referred to a zero point).

In contrast to G90, however, you can switch over to relative dimension on a word-by-word basis. The programmed end point will then refer to the last approached axis position.

To do so, you program the "I" address parameter (I = incremental) in addition to the axis address (e.g. "X", "Y" etc.). This means that the complete axis designation will then consist of the "XI", "YI", "ZI" etc. character string plus one numeric value.

Examples: "XI200", "ZI40".

G191: You instruct the control unit to interpret all dimensional statements in the part program as relative dimensions (referred to the last approached axis position).

In contrast to G91, however, you can switch over to absolute dimension on a word-by-word basis. The programmed end point will then refer to the currently active zero point.

To do so, you program the "A" address parameter (A = absolute) in addition to the axis address (e.g. "X","Y" etc.). This means that the complete axis designation will then consist of the "XA", "YA", "ZA" etc. character string plus one numeric value.

Examples: "XA200", "Z(A40".

Please note for G190 and G191:

- G190 and G191 act modally, cancel each other mutually and cancel G90, G91 and G189.

Example: Absolute with relative dimension -G190-;

N 20  G190 ... From the next block on, relative dimension is possible
N 30  G1  X100 Y150 F1000 S150 axes traverse to X100,Y150
N 40  XI50 YI50 axes traverse to X150,Y200
N 50  X200 Y250 axes traverse to X200,Y250
N 60  X210 YI250 axes traverse to X210,Y500

Example: Relative with absolute dimension -G191-;

N 20  G191 ... from the next block on, absolute dimension is possible: current absolute axis position X10,Y10
N 30  G1  X100 Y50 F1000 S150 axes traverse to X110,Y60
N 40  XA250 YA200 axes traverse to X250,Y200
N 50  X200 Y250 axes traverse to X450,Y450
N 60  XA100 Y50 axes traverse to X100,Y500

Word by word programming of relative dimension with "XI", "YI", "ZI", etc. is not permitted in combination with function G76, "Traverse to machine-oriented absolute axis position" and would cause a runtime error.
4.66 Speed limitation  

**G192, G292**

**Effect**

To make sure that the spindle speed does not rise or fall excessively (clamping pressures, centrifugal forces on chuck, etc.), the upper or lower limit of the admissible speed range can be programmed in the part program.

A speed change – even if initiated by the spindle potentiometer – will not be performed unless it is within the absolute limits specified.

The limit values apply to all speed ranges, however, they are effective only if they are within the speed range limits.

**Programming**

G192: Determine lower limit of admissible speed.

The spindle minimum speed is programmed in the same block as G192 with the S word.

G292: Determine upper limit of admissible speed.

The spindle maximum speed is programmed in the same block as G292 with the S word.

A limit value can be cancelled by programming the following in the same block as the corresponding G instruction

- no S word, or
- an S word with a value \( \leq 0 \) (e.g. "S0", "S–1").

**Please note for G191 and G292:**

- G192 and G292 can be replaced by programming new limit values.
- The speed limits influence the direct speed programmed in G97 and the constant cutting speed in G196.
- The S values programmed with a speed limit do not influence the speeds in connection with M3/M4 programming. They remain stored and effective until M2/M30, "Control reset", "Reset" or one of the cancelling functions is activated.

**Example:** Programming a speed limit

N...
N100 X... Y... G192 S1500  minimum speed: 1500 rpm
N101 X... Y... G292 S2500  maximum speed: 2500 rpm
N... X... Y... G292 S–1  cancel maximum speed
N... X... Y... G192 S0  cancel minimum speed
4.67 Oscillating axis

Effect
Using the "oscillating axis" function, an oscillating movement can be performed with any synchronous axis while linear interpolation is carried out for the other synchronous axes of the channel (e.g. flat grinding).

Any synchronous axis can be defined as oscillating axis:
- linear axis
- rotary axis
- C axis

The parameters of the oscillating movement (initialization) are set by G350 and saved modally.

This initialization must be programmed prior to the actual oscillating movement:
- selection of the oscillating axis
- starting and end position as reversing points of the oscillating movement
- frequency or speed of oscillating movement

The oscillating axis is implemented as modal function G301, "oscillation with linear movement". The transition of the oscillating movement between 2 consecutive oscillation blocks is steady (also in terms of speed).

Programming

1. Initialization: Oscillating axis

G350 OscAxis<physical axis index> URP<axis position>
    LRP<axis position> F<speed> OF<oscillation frequency>
    R<reversing range>

Explanation:
OscAxis selection of the oscillating axis (physical axis index)
URP upper reversing point of the oscillating axis (mm)
LRP lower reversing point of the oscillating axis (mm)
F speed of the oscillating axis (mm/min)
- alternatively to F –
OF oscillation frequency (Hz in 1/sec) – alternatively to F –
R reversing range (not yet available, will be implemented in a later option)
Example: Oscillating axis

G350 OscAxis4 URP200 LRP100 OF5 Oscillating axis is the 4th physical axis (e.g. U axis)
G301 X100 Y10 F20 Time 200

2. Programming the oscillating movement: Oscillating axis

G301 X<axis position> Y<axis position> F<speed> Time<duration>

Explanation:
X synchronous axis, interpolating linearly with Y
Y synchronous axis, interpolating linearly with X
F path feed of axes (X, Y)
Time duration (ms) of oscillating movement for blocks without traversing movement

- If the axis address of the oscillating axis is programmed with a traversing path, an error message will be generated. None of the explicitly programmed axes may be defined as an oscillating axis.

- The programmed time (Time) only refers to the block in which "Time" was programmed. The oscillating movement takes at least as long as the programmed time. A synchronous traversing movement programmed in the same block whose execution takes more time than the programmed oscillating time, will let the oscillating movement take longer. If oscillating is still active in the next block, although no time has been programmed, the execution time of this block will be solely determined by the synchronous axis movement.

The display is in workpiece coordinates. If a zero point shift is selected for the oscillating axis, the application of the shift will be postponed as long as the oscillating axis is active. However, it will be applied to the workpiece position display.

The last position prior to the beginning of the oscillating movement will be displayed as end position. This position also acts as starting position of the oscillating movement.

The distance to go is defined as the difference between the end point and the current machine position setpoint. It oscillates between 0 and the distance between the reversing points.

Please note for G301 and G350:
- G301 is a modal function (groups G1, G2, ...)
- G350 is not modal (therefore not in display)
- G350 sets the parameters modally, i.e. the old parameters can only be overwritten by programming G350 with suitable parameters again.
- When oscillating has been activated, the oscillating axis first traverses to the reversing point which can be reached within the shortest distance starting from its current position.
- Oscillating will remain active until a new modal movement function (e.g. G1, G2, ...) is programmed.
- For as long as oscillating is active, the oscillating movement will be steady and can be differentiated across block limits.
- When oscillating is turned off, the oscillating axis will return to the reversing point from where it started.
- After Control Reset, the oscillating movement will not be cancelled before the next reversing point (speed = 0) has been reached. The modal function will be cancelled in accordance with the default status.
- Programming of NC functions (loop gain (KV) programming, feed forward control, etc.) which act on the physical address of the oscillating axis should be avoided within the machining section because sudden speed losses may be the consequence.
- Programming the address of the oscillating axis in connection with a function internal to the NC (e.g. G60 oscillating axis address<value>), this function will be activated, however, it will not take effect on the oscillating axis for as long as oscillating is active. The compensation (e.g. G60) will not be selected before the oscillating movement has stopped.
- The program value equals zero during the oscillating movement.
- If a zero offset is activated while G301 is being programmed, first make a query of the current data of the oscillating axis using CPL function "FXC" because this data must be taken into account when programming the "URP" and the "LRP" (G350).

**Example:**

1  A=FXC(4)
N2  G350 OSCAxis4 URP[200+A] LRP [100+A] OF5
N3  G301 X100 Y10 F20 Time200

**Restriction:**
Function G301, "Oscillating axis", is not permitted in combination with functions G61 or G163, "In-position".

While G301 is active, the following functions must not be programmed (because otherwise interpolation would be aborted abruptly which may result in a servo error):
G4, G14/G15, G32, G75, G114/G115, G374, G590/G591, G900

**Auxiliary functions** may be programmed together with G301 only if the time required for interpolating the NC block is more than the time required for the execution of the auxiliary function, including acknowledgement. Basically, the time required for executing an NC block is determined by the path and the feedrate programmed for this block as well as by the duration of the oscillating movement, "Time". Any G301 blocks for which no duration of the oscillating movement nor a traversing motion has been entered are executed within one interpolation cycle.

To ensure that the oscillating movement is properly terminated, WAIT needs to be programmed before any M0/M1. If the execution of the program is continued upon a cycle start command, also the oscillating movement will be resumed.

**Example:**

N50  G301 ...
    ...
N60  WAIT N70 M0
N80  ...
4.68  Precision programming

Effect

The precision programming function automatically reduces the feedrate for contour transitions or circular contour segments (circles, helical, helicalN) to ensure compliance with accuracy specifications (please refer to the figure below).

For this purpose, a feedrate value is calculated based on a control system model (closed position control loop in steady-state condition, with feed forward control taken into account). This feedrate value will ensure that the programmed tolerance does not fall below the required precision at the block transition.

In contrast to the "in-position" function, the feedrate is not reduced to 0 at a contour transition with the "precision programming" function. With the "in-position" function, the actual following errors of all axes of the respective channel are checked after deceleration to 0 speed.

The precision tolerance range is set by selecting one of 2 different parameters:

- **Deviation from contour** \( \varepsilon \) : Maximum permissible deviation for contour transitions or maximum permissible deviation from radius for circular arcs.
- **Overtravel** \( \delta \) : Maximum overtravel (distance from corners) not to be exceeded at the actual transition point of a contour transition.

Programming

G328 Activation of precision programming with the value of deviation from contour \( \varepsilon \) as preset in MACODA parameter 705000510.

G328 EPS<deviation from contour> Activation of precision programming

G328 DIST<distance from corners> Activation of precision programming

G329 Deactivation of precision programming

Explanation:

**EPS**

**Contour transition**: A deviation from contour at a block transition is the minimum deviation of the actual contour at the transition from the programmed position.

**Circular arc**: A deviation from radius is the difference between the programmed radius and the resulting actual radius, which is dependent on path velocity.

**Deviation from contour**

Distance \( \varepsilon \) to be entered in mm or inch (depending on the unit set with G70 or G71).

**DIST**

The overtravel is the distance between the position where the actual contour first deviates from the programmed contour before a contour transition and the block position programmed.

When programming circles with DIST, the \( \varepsilon \) value set in MACODA parameter 705000510 is taken into account.

**Deviation at corners**

Distance \( \delta \) to be entered in mm or inch (depending on the unit set with G70 or G71).

To avoid the need for entering the deviation from contour \( \varepsilon \) explicitly each time, a default value may be entered in MACODA parameter 705000510.
Please note for G328 and G329:

- Identical dynamics must be set for all axes. If drives other than Bosch drives are used, active feed forward control functions will be taken into account only in terms of quality (following error reduced by 50%). The configuration of operation without any following errors in SERCOS varies with the manufacturer.
- A path slope function, G8 or G108, must be active. Otherwise, the speed is decelerated to 0 at every contour transition.
- In-position (G61, G163) must be deactivated to prevent deceleration to 0 at every contour transition.
**Effect**

The “inclined plane” function defines the position of a workpiece coordinate system (WCS) positioned and oriented spatially in relation to the machine coordinate system (MCS). The coordinates programmed in the part program refer to the workpiece or program coordinate system. The interpolated program coordinates are automatically converted into machine coordinates for every interpolation step in the control unit.

Since there are 3 degrees of freedom for orientation, every orientation can be represented by 3 consecutive basic rotations.

The basic rotations underlying the “inclined plane” function are shown on the following figures:

- Machine coordinate system with zero point
- Rotation of the coordinate system by the axis $Z_M$ and the angle $F=\phi$
- Rotation of the coordinate system by the axis $Z''_W (=Z'_W)$ and the angle $Y=\psi$
- “Inclined plane” offset relative to the machine coordinate system

Coordinate system offset by the distance $DX$, $DY$, $DZ$ and rotation by the angles $\phi$, $\theta$ and $\psi$ relative to the machine coordinate system

$M =$ machine coordinate system

$W =$ workpiece coordinate system

(or program coordinate system)
Since the inclined plane can be offset and rotated in space relative to the MCS, it is necessary to define the precise location and the orientation of the program or workpiece zero point.

The zero point of the workpiece or program coordinate system of the "inclined plane" relative to the machine coordinate system

- can be entered directly using G352

Programming

G352 X<X offset> Y<Y offset> Z<Z offset> PHI<1st Eulerian angle> THETA<2nd Eulerian angle> PSI<3rd Eulerian angle>

Explanation:
X Offset value in X direction relative to MCS zero point
Y Offset value in Y direction relative to MCS zero point
Z Offset value in Z direction relative to MCS zero point
PHI Angle of rotation by Z axis relative to MCS (syntax: PHI, Phi, phi)
THETA Angle of rotation by the new Y axis (relative to the position of the coordinate system after rotation by PHI) (syntax: THETA, Theta, theta, the)
PSI Angle of rotation by the new Z axis (relative to the position of the coordinate system after rotation by THETA) (syntax: PSI, Psi, psi)

- or called indirectly
  - with G354..G359 (internal call of a table containing all position and orientation parameters). The table has the format of an ASCII file: ID<filename>. G22 will activate the ID table.
  - With G353, the "inclined plane" can be deactivated.

Example: G354..G359

N... G22 IDTab1 activate Inclined plane table Tab1
N... G354 "inclined plane” active; no traversing movement
(or)
N... G354 X...Y...Z... offset and angle of rotation already apply to position programmed in this block
N... G353 cancel active inclined plane

Please note for G352, G353, G354..G359:

- The "inclined plane" function is retained after Control Reset provided that no appropriate function is included in the default (power-up) status.
- Functions G352, G353, G354..G359 act modally and deactivate each other mutually.
- Activating the "inclined plane" function automatically deselects the following zero point shifts, i.e. this function must not be programmed while the "inclined plane" function is active:
  - G54..G59
  - G154..G159
  - G254..G259
  - G160, G260, G360
- G352..G359 must not be programmed together with a traversing motion.
• Programming G352...G359 will interrupt the look-ahead function. Therefore, G352...G359 must **not** be programmed while the "cutter compensation" function is active. Consequently, the inclined plane must be selected before the "inclined plane" function is activated.

• If workpiece coordinates are displayed, they refer to the "inclined plane".

• The "inclined plane" function acts additively on function G138, "Workpiece position compensation".

• When the "inclined plane" function is active, functions G37, G38, G60, G168, G268, G145 – G845, G147 – G847 and Hx are also taken into account.

• The "inclined plane" function always refers to the **first three "logical axes"** of a channel.

• Within an active "inclined plane" function, a plane can be selected with G17, G18, G19, G20. Its coordinates refer to the coordinate system of the "inclined plane".

  The following coordinates form the respective plane:
  
  - G17: Xprog Yprog
  - G18: Zprog Xprog
  - G19: Yprog Zprog,

  provided that the corresponding axis classifications have been set in MACODA.

• A contour offset programmed with G60 (with active "inclined plane" function) refers to the coordinate system of the "inclined plane". Programmed axis addresses specify the coordinate directions relative to the "inclined plane".

• Axis addresses acting **directly** on the axes are **not affected** by the "inclined plane" (e.g. G14 X2: In this case, the loop gain (Kv) value acts on the X axis of the machine coordinate system).

• The axis display is in machine and/or workpiece coordinates.

---

The structure of the ID<...> table for the "inclined plane" is described in section "ASCII files (tables)".
4.70 Axis transfer

Overview

The following axis transfer functions are available:

- **G510**: Integrating axes with an error message if axes are not available for integration
- **G511**: Integrating axes with WAIT until axes are released
- **G512**: Removing an axis from an axis group
- **G513**: Accepting the axis configuration from MACODA
- **G515**: Assigning "Logical axis name"
- **G516**: Removing "Logical axis name"
- **G21**: Programming axis classification (see sect. 4.19)
- **G16**: No plane (see sect. 4.16)

Parameters

The following syntax applies to all of the following functions:

- **PAN** physical axis name
- **PAI** physical axis index
- **LAN** logical axis name
- **PANi | PAi | LANi**  
  optional:
  - ith physical axis name, or
  - ith physical axis index, or
  - ith logical axis name
- i, n number of axes applied (i=1..n; currently available: nmax.=8)

Functions

The **axis transfer** function serves to define the conditions of

- transferring axes to channels or axis groups
- removing axes from axis groups, or
- transferring axes from one channel to another.

Please note that transferring axes from or to a channel will impact the machining function of the logical axes (see also sect. 4.19).

Integrating axes in axis groups

Effect

Function G510 is used to integrate one or several axes in an axis group (i.e. the axis is transferred to a channel).

At the time the block is being prepared, **one** of the following conditions must be fulfilled or the transfer cannot be performed:

- The axis must be asynchronous, i.e. it is **not** assigned to **any other channel**, and it must be at rest when it is to be transferred. Otherwise, an error message is displayed.
- If the axis is assigned to another channel, this channel must **not** be active.

If the channel to which the axis was originally assigned is activated (e.g. by program selection) after the axis is transferred to another channel, an error message relating to the original channel is displayed. The axis involved must not be assigned to more than **one active channel**!
If none of the above conditions is fulfilled, a runtime error will occur.

In contrast to G510, there is a WAIT included in function G511 until the axis (or all axes) has/have ceased to be busy.

When transferring an axis to another channel, entry of a logical axis name by which the axis is to be addressed on the new channel is optional.

**Example:**

The following figure shows how the physical axis YP is integrated in an axis group. Subsequently, this axis is no longer available for being addressed as the asynchronous axis YP on another channel. On the channel where it has been integrated, the synchronous axis can be programmed optionally under its logical or its physical axis name. The channel axis display will also show the Y axis.

Channel axes Z and B are assigned new logical axis indices.

An axis that is integrated in a channel is assigned axis classification 999 unless the axis is part of the default configuration of this channel. If the axis is part of the default configuration of this channel, the axis is assigned its default classification from the MACODA.
Removing an axis from an axis group

Effect

Function G512 is used to remove one or several axes from an axis group (i.e. the axis is released).

Thus, a previously synchronous channel axis turns into an asynchronous axis once it is released.

Block preparation is not interrupted by this action.

The following figure shows the release of channel axis Y. Following its release, the axis can be programmed as an asynchronous axis YP on all channels. Channel axes Z and B are assigned new logical axis numbers. On the channel axis display, the Y axis is not shown any more. Programming the logical axis Y will trigger an error message.
Transferring axes in axis groups

When transferring axes from one channel to another, two different cases are to be distinguished:

- The channel of the "source axis group" from where the axis is transferred is not active:

  If the channel of the source axis group is not active, any of its axes may be integrated anytime in another axis group, i.e., you might say the axis concerned is lent to the other channel. Once it is released from that other channel, it is automatically reintegrated in the source axis group from where it was transferred.

  For instance, following a system reset, this type of axis transfer allows you to start a user-written reset program on one channel that is designed to borrow all system axes from other channels, too, and to reset them to their respective initial state.

  If an axis of the source channel is lent to another channel and you try to select a program or "Manual data input", an error message will be displayed.

- The channel of the source axis group from where the axis is transferred is active:

  When an axis is transferred from an active channel (axis group 1), the axis is removed from its axis group in a first step, and in a second step it is integrated in another channel (axis group 2).

  To ensure that the axis transfer can be executed without any waiting time occurring in block preparation, the axis must have first been removed from axis group 1. This is the case if at the time the axis is integrated in axis group 2, the respective NC block in axis group 1 is already active (i.e., the axis has first been removed from there with G512).

Example:
The X axis of channel 1 is transferred to channel 2 (see fig. below).
On channel 1, preparation has advanced to block N1310 and block N1220 is active. Thus, the release of the X axis is concluded.
At this point in time, block N2110 is active on channel 2. Preparation has advanced to block N2220 and is going to take over physical axis XP (previously the X axis on channel 1).
Axis XP is assigned the name ZA in the process. Because this name is already known on channel 2, no waiting time will occur in the transfer process.

Channel 1

<table>
<thead>
<tr>
<th>N1100 ...</th>
<th>N1110 X0 Y0 Z0</th>
</tr>
</thead>
<tbody>
<tr>
<td>: machine with axes X, Y, Z</td>
<td></td>
</tr>
<tr>
<td>..</td>
<td></td>
</tr>
<tr>
<td>: remove X axis from axis group (channel 1)</td>
<td></td>
</tr>
<tr>
<td>N1210</td>
<td>G512(X)</td>
</tr>
<tr>
<td>N1220 Y0 Z0</td>
<td></td>
</tr>
<tr>
<td>N1310 Y100 Z100</td>
<td></td>
</tr>
<tr>
<td>..</td>
<td></td>
</tr>
</tbody>
</table>

NC block preparation on channel 1
active block, release of X concluded

Channel 2

| N2100 ... |
| N2110 XA0 YA0 ZA0 |
| : machine with axes XA, YA, ZA |
| ..         |                |
| : remove ZA axis from axis group (channel 2) |
| N2210 G512 (ZA) |
| : integrate axis XP and assign the name ZA to it |
| N2220 G510(XP, ZA) |
| N2230 XA0 YA0 ZA0 |
| : machine with axes XA, YA, ZA |
| N2310 XA100 YA100 ZA100 |
| ..       |                |

NC block preparation on channel 2
active block

ZA (PAN: ZP):  ➔ asynchronous
axis transfer

Effect

Synchronous: LAN: ZA PAN: XP
Asynchronous:
Axis transfer programming: G510, G511, G512, G513

Axis designations are programmed by pairs. A parameter list may include several such pairs.

Integration of axis with an error message displayed if axes are not available:
G510 (<PAN | PAI>,{<LANi>},..,<PAN n | PAIn>,{<LANn>})

Explanation:
PAN | PAI Defines the axis/axes to be integrated in the receiving channel.
LAN Programming is optional and defines the "logical name" by which the axis to be integrated is to be addressed on the receiving channel. This logical name must have been predefined in one of channel-specific MACODA parameters 701000010, "Logical axis designation", or 701000020, "Optional axis designation". If you choose not to assign a logical name, enter two commas.

Example:
G510 (YP,,ZP,Z) Physical axes YP and ZP are integrated in the receiving channel. ZP may be addressed with its logical name Z, while YP can be addressed with YP only. If either one of these axes has not been released, a runtime error will occur.

Integration of axes including a WAIT until axes have been released:
G511 (<PANi | PAi>,{<LANi}>,..,<PAN n | PAIn>,{<LANn}>)

Explanation:
PAN | PAI same as G510
LAN same as G510

Example:
G511 (YP,ZP,Z) same as G510, except that block preparation will wait until YP and ZP have been released.

Removal of an axis from an axis group:
G512 (<PANi | PAi | LANi>,..,<PAN n | PAIn | LANn>)

Explanation:
PAN | PAI | LAN Defines the axis/axes to be removed.

Example:
G512 (XP,2,Z) Physical axis XP, the physical axis assigned the index 2, and logical axis Z are removed from this axis group.

Accepting the axis configuration from MACODA
G513

Please note for G513:
If an axis that is to be transferred has not been released yet, this will cause a runtime error. Therefore, G513 should be entered in a suitable position in the init string. If G513 is entered after the key word #Reset: in the init string, G513 is executed only with a reset and, if entered after #SysRes: only with an overall reset of the system.
Assigning a logical axis name

With function G515, you can assign a new logical axis name on a channel. Again, the same conditions apply as in the case of an axis transfer, i.e. the "new logical axis name" must have been predefined in one of the two channel-specific MACODA parameters, either in 701000010, "Logical axis designation", or in 701000020, "Optional axis designation".

Using G516, the assignment of the name can be reversed.

Axis designations are programmed by pairs. A parameter list may include several such pairs.

**Programming**

**Assignment of a "Logical axis name":**

G515 (<PAN1 | PAI1 | LAN1>,<LAN1>,...,<PANn | PAIn | LANn1>,<LANn2>)

**Explanation:**

PAN | PAI | LAN1 Defines the axis/axes that are to be assigned a new "logical axis name" (LAN2).
LAN2 Specifies the "new" logical axis names.

Axis designations are programmed by pairs. A parameter list may include several such pairs.

**Example:**

G515 (YP,X,3,Y,B,Z) Physical axis YP is assigned the logical axis name X, the 3rd physical axis is assigned the logical name Y, and logical axis B is assigned the logical name Z. Programming B will generate a runtime error.

**Removing a "Logical axis name":**

G516 (<PANi | PAIi | LANi>,...,<PANn | PAIn | LANn1>)

**Explanation:**

PAN | PAI | LAN1 Defines the axis/axes to be removed from the receiving channel.

**Example:**

G516 (YP,3,Z) The logical names of physical axis YP, the 3rd physical axis and logical axis Z are removed from the receiving channel.
4.71 Drive-controlled interpolation  

**Effect**

Using the "Drive-controlled interpolation" function, a synchronous axis can be switched to drive-controlled operation and controlled in parallel to synchronous operation.

In the case of drive-controlled interpolation, a synchronous axis can be given position data under its physical axis address through any channel although it continues to be permanently assigned to one channel.

At a given moment of time, an axis can only process position data received from one channel. If another channel wants to access the axis, and if a drive-controlled interpolation is already being performed for this axis by another channel, an error message will be output. The position data of the requesting channel will not become active for this axis.

If a channel specifies new position data although the last data specified by the same channel has not been completely interpolated yet, the old data will be replaced by the new one.

**Programming**

N.. G522 X1 Z1  
X and Z are switched to drive-controlled interpolation. (Only possible on the channel to which axes X and Z have been assigned.)

N.. G521  
All drive-controlled axes of the channel are returned to NC operation.

N.. G520 X100  
Drive-controlled interpolation of the X axis shall be performed for position X100 (can be programmed from any other channel only if the axis is not occupied and has been switched over accordingly). G520 cannot be programmed from the channel to which the axis is assigned.

N.. G523 X1000  
The positioning speed of the drive-controlled X axis is to be 1000 mm/min (unit specified in MACODA parameter 70400001). This can be programmed from any channel and at any time. The positioning speed will be entered under SERCOS ID no. S-0-0259.

N.. G524 X3  
The acceleration of the drive-controlled X axis is to be 3m/sec. This can be programmed from any channel. The acceleration will be entered under SERCOS ID No. S-0-0260.
Please note for G521:

- During drive-controlled interpolation, the NC’s internal override function for the axis in question is not active. However, the PLC can directly specify the Feedrate Override (SERCOS ID no. S-0-0108) through the SERCOS service channel in the same way as for referencing.
- The drive responds to the Drive Hold control signal, i.e. drive-controlled interpolation is halted by Feed Hold for the channel to which the axis has been assigned in MACODA parameter 100300002. The axis will continue to run after NC Start. Control Reset for this channel will cancel drive-controlled interpolation and resume NC-controlled position control (corresponding to G522).
- Axis reset will not have any effect on any synchronous axis.
- Only the channel to which the axis is assigned can switch over the drive mode. The “drive-controlled interpolation” drive mode is output at the axis interface of the axis in question.
- Auxiliary modes 2 and 3 of the SERCOS drives must be assigned appropriate parameters for “drive-controlled interpolation”:
  - If main operation mode was active previously, the mode is switched to secondary mode 2.
  - If secondary mode 1 was active previously, the mode is switched to secondary mode 3.
  - Bit 4 (interpolation in the drive) must be set in the respective SERCOS configuration parameters for secondary modes 2 and 3.
- Functions G521 and G522 are active modally and deselect each other mutually.
- Functions G520, G523 and G524 are active block-by-block.

**CAUTION**

When programming functions G523 or G524 of release 4.x.x, incorrect data may sometimes be written to the drive!

If this happens, use G900 for entering positioning speed and acceleration data.
4.72  **Collision monitoring ON**
G543
**Collision monitoring OFF**
G544
**Look-ahead for collision monitoring**
G500

**Effect**

In connection with G41 and G42, these functions are designed to monitor the contour offset by the cutter path compensation for potential collisions.

A collision occurs if the look-ahead function of the collision monitoring function detects a point of intersection or contact of two path segments on the offset contour path computed by the cutter path compensation function. If an intersection is detected, the offset path always turns off into a loop. In particular, any point where the contour intersects with itself results in a collision. Only the two coordinates (axes) of the active working plane are taken into account for detection of any contour loops. Any changes that may be caused by the infeed per cut are ignored because the control has no information of how deep the tool plunges into the workpiece.

The effective look-ahead range for collision monitoring can be adjusted (default=2 blocks).

If the machining tool radius does not permit machining of individual contour elements, the control unit will try to modify the related path so as to avoid damages to the contour.

![Diagram of programmed contour without and with collision monitoring](image)

Although full circles always constitute a programmed loop in itself on the programmed contour, they are excluded from collision monitoring provided they are retained as full circles if a compensation value is taken into account.

![Diagram of no and with collision](image)
Generally, it is impossible for the control to recognise whether a detected collision may have been programmed on purpose or not. Therefore, the control functions can be adjusted to individual machining segments.

For this purpose, the Typ3 osa offers the following options:

**G543:** Activate collision monitoring. With collision monitoring activated, you can state whether you want collisions to be indicated by a runtime error or a warning message or not.

**G544:** Deactivate collision monitoring.

**G500:** With G500, you can set the look-ahead range for collision monitoring, either generally by changing the default setting, or just locally, for a specific segment.

The Typ3 osa performs collision monitoring even if the value of the active D word of the tool radius compensation is set to “0” (e.g. D1=0).

---

### Programming

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>G543 CollErr 0</td>
<td>Activate collision monitoring; if an anticipated collision is detected, neither runtime errors nor warning messages are displayed.</td>
</tr>
<tr>
<td>G543 CollErr 1</td>
<td>Activate collision monitoring; if a collision is detected, a runtime error is displayed and machining is suspended.</td>
</tr>
<tr>
<td>G543 CollErr 2</td>
<td>Activate collision monitoring; if a collision is detected, a warning message is displayed but machining continues.</td>
</tr>
<tr>
<td>G543</td>
<td>Activate collision monitoring; behavior in terms of messages displayed remains unchanged. Unless the response in the event of a collision has been programmed previously (or entered in the init string), G543 = G543 CollErr 0!</td>
</tr>
<tr>
<td>G544</td>
<td>Deactivate collision monitoring.</td>
</tr>
</tbody>
</table>

ℹ️ Optionally, the CollErr variable may also be spelt COLLERR.

### Programming

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>G500 VS 3</td>
<td>The look-ahead range for collision monitoring is set locally to 3 blocks. When G41 or G42 are programmed the next time, the set default value becomes effective again.</td>
</tr>
</tbody>
</table>

ℹ️ In release V4.3.8.1 or later, “G500 VS n” may be programmed in one block with G41 or G42. The look-ahead range can be set to up to 5 blocks.
G500 DVS 1: With the DVS variable, you can change the preset value of the look-ahead range. The look-ahead range thus programmed becomes effective the next time G41 or G42 is programmed. The preset value can be overridden locally with "G500 VS n". "G500 DVS n" must be programmed in a block before G41 or G42. It is recommended that you select a value for presetting between 1 and 5 blocks.

G500: The preset value of the look-ahead range is reset to the default value of 2 blocks. The effect of G500 is the same as if you program "G500 DVS 2".

Please note for G543, G544 and G500:

- You can select the power-up condition (monitoring response following control reset) – Monitoring ON/OFF – by setting MACODA parameters 70600020 and 70600010 for a specific channel.

- The response in terms of messages displayed in the event of a collision (G543, CollErr) programmed last remains effective until it is reprogrammed or a new default is set via the init string at control reset. If the response in terms of messages displayed shall always be the same, you can program this simply by making the respective entry in the init string. In the part programs, you may only have to program G543 and G544.

- The DVS default value programmed last remains effective until a new presetting or just G500 is programmed or activated via the init string at control reset. If you want to apply the preset look-ahead range (i.e. a range other than 2 blocks) all the time, you just need to program "G500 DVS n" in the init string. With G500 DVS 1, e.g., you can set the look-ahead range for collision monitoring to 1 block (like with CC220 or Typ1 osa).

- If the look-ahead function finds a traversing block for which the collision monitoring function is deactivated (G544), collision monitoring, which was activated by a previous block, is terminated in this traversing block. When collision monitoring is activated again (G543), a new look-ahead process is started for collision monitoring. In order to suspend the collision monitoring function temporarily, it is not enough just to program G544 in two subsequent blocks and then to program G543 in the next block. In addition, a traversing motion must be programmed between G544 and G543. This traversing motion may be programmed together with G544 in one and the same block.
Collision monitoring with reverse compensation direction

In order to move backwards along the contour while the cutter path compensation function is active, a reversal in compensation direction (G41 becomes G42, or G42 becomes G41) must be programmed. In this case, the collision monitoring function will not signal a collision of the forward motion with the subsequent backward motion.

To achieve this, the look-ahead function for collision monitoring is canceled in the block in which the compensation direction is reversed. Subsequently, a new look-ahead is started in the block reversing the compensation direction.

Example:

N10 G41 G500 DVS10 H1 Traversing forward with cutter path compensation on the left
N20 X10
N30 X20
N40 X30
N50 G42 Reversing motion with cutter path compensation on the right. Although the look-ahead function is set to 10 blocks ahead, collision monitoring is canceled and restarted in block N50.
N60 X20
N70 X10
N80 X0
N90 G40
M30
### Axis coupling

**Effect**

The axis coupling function establishes a fixed relationship between the positions of a **master axis** and a **slave axis**.

**Group of coupled axes:**

Master axes and slave axes are linked to form a **group of coupled axes**. Every group of coupled axes consists of just one master axis and up to **seven** slave axes. All axes belonging to a group of coupled axes must be on one and the **same channel**. A channel may be assigned more than one group of coupled axes.

The following overview shows the structure of parallel axes in terms of group of coupled axes and channel assignment:

- **Group of coupled axes:** several groups of coupled axes per channel;
- **Spindle:** spindle must not belong to a group of coupled axes!

- **Channel:** several channels per NC
- **Group of coupled axes:** several groups of coupled axes per channel;
- **Spindle:** spindle must not belong to a group of coupled axes!
There are some special groups of coupled axes with specific names, e.g.:

- **parallel axes** (layout of machine tables in parallel)
- **electronically controlled gear box** (axes moving at certain ratios)

**Coupling characteristics:**
As a rule, a group of coupled axes can be defined by its coupling characteristics.
The coupling characteristics describe the position of the slave axis relative to the position of the master axis. **Linear** coupling characteristics are distinguished from **freely defined coupling** relationships of the axis positions relative to each other.

**Linear coupling:** The relationship between the position of the master axis, $p_m$, and the position of the slave axis, $p_s$, can be linear:

$$p_s = p_m \times k + o$$  \hspace{1cm} \text{(formula 1)}

- $k=1 \rightarrow$ parallel axes
- $k \neq 1 \rightarrow$ electronically controlled gears

**Freely defined coupling:** Function $f(pm)$ is stored as a function table (coupling table) in the file system of the Typ3 osa.

$$p_s = f(p_m - p_m^o) \times k + o$$  \hspace{1cm} \text{(formula 2)}

A group of coupled axes consists of a master axis and one or more slave axes: **Each** slave axis has its own specific coupling relationship with the master axis as defined by (1) or (2).

The second members of equations (1) and (2) represent reference values in the form of parameters that are dependent on the position of the master axis and that are used in every interpolation cycle as a position input for the slave axes.
4.73.1 Types of axes

The following types of axes may be used both as master and slave axes:
- synchronous axes
- axes that can be switched from synchronous to asynchronous (if they belong to a group of axes, they must be switched to synchronous)
- modulo axes

The following types of axes must not be used as master or slave axes:
- asynchronous axes
- Hirth axes

Restrictions applying to modulo axes:
- Linear coupling characteristics:
  If the master axis is a modulo axis (linear or endless: see MACODA parameter 100300004), the slave axis must be a modulo axis, too. The following restriction applies in respect of the modulo value (drive parameter):

\[
m_m \cdot k \mod m_s = 0
\]

modulo value, slave axis
modulo value, master axis

- Coupling via coupling table:
  Modulo master axes can be coupled with non-modulo slave axes via the coupling table. The restrictions in respect of the coupling table (see sect. 4.73.6, parameter #20) apply to master axes (modulo axes). Slave axes are not subject to any restrictions.

4.73.2 Forming a group of axes

Programming

G581: Forming a group of axes with linear coupling

G581<master name>0 <slave name 1>({<o_s1>,<k_s1>}) ... {<slave name n>({<o_sn>,<k_sn>})}

Explanation:
master name logical axis address of the master axis
slave name i logical axis address of the i\textsuperscript{th} slave axis
\(o_{si}\) shift of the i\textsuperscript{th} slave axis
\(k_{si}\) coupling factor of the i\textsuperscript{th} slave axis
\(i = 1 .. \text{max. 7 (n)}\) max. number of slave axes per channel and group of coupled axes

Example:
G581 Z0 A(4,2) B(2,1) Z= master axis,
A and B = slave axes
G581: Forming a group of axes with **freely defined** coupling (via coupling table)

\[ G581 <\text{master name}>0 <\text{slave name }i> \{<o_{si}>,<k_{si}>,<p^{0}>,<f_{si}>\} \]
\[ ...<\text{slave name }n>\{<o_{sn}>,<k_{sn}>,<p^{0}>,<f_{sn}>\} \}\]

**Explanation:**
- **master name** logical axis address of the master axis
- **slave name i** logical axis address of the **ith** slave axis
- **o_{si}** shift of the **ith** slave axis
- **k_{si}** coupling factor of the **ith** slave axis
- **p^{0}** shift of the master axis
- **f_{si}** name of the coupling table for the **ith** slave axis
- \( i = 1 .. \text{max. 7} \text{ (n)} \) max. number of slave axes per channel and group of coupled axes

**Example:**

\[ G581 \text{ X0 A(4,2) B(–3,0.5,0,"Ftab_B")} \]

- coupling table for coupling slave axis B and master axis X

For any parameters not stated in the syntax, **default values** are set:

\[ k_{S} = 1; o_{S} = 0; p^{0} = 0 \]

There is no default available for the coupling table.

Therefore, the following terms are available for the coupling syntax (slave axis B):

- B() linear coupling with o = 0, k = 1
- B(2) linear coupling with o = 2, k = 1
- B(–1) linear coupling with o = 0, k = –1
- B(...,Tab) coupling via table with o = 0, k = 1, p^{0} = 0, f = Tab
- B(...,4.5,Tab) coupling via table with o = 0, k = 1, p^{0} = 4.5, f = Tab
- B(–3.2,...,4.5,Tab) coupling via table with o = –3.2, k = 1, p^{0} = 4.5, f = Tab

The following **actions** are carried out with G581 <master name>0 ...:

- Deletion of existing group of axes <master name>
- Slave axes 1 through n stated in the syntax are deleted in other groups of axes
- Formation of the <master name> group of axes with slave axes 1 through n.

**CAUTION**

Forming a group of axes with NC block "G581 <master name>0 ..." will cause a traversing motion of all slave axes programmed in this block.

Each slave axis will traverse to its specific coupling position (reference value) defined by the position of the master axis and the coupling characteristics.

The syntax used previously for axis coupling with G functions

- G590 MASTER= formation of a group of axes
- G591 deletion of all groups of axes

is still supported.
4.73.3 Expanding an existing group of axes

An existing group of axes may be expanded by one or several slave axes, or existing axis couplings may be changed:

Programming

G581: Expanding an existing group of axes with linear or freely defined coupling (via the coupling table)

G581 <master name> i <slave name i>({<o_i>,<k_i>,<p_i>,<f_i> })
...{<slave name n>({<o_n>,<k_n>,<p_n>,<f_n> })}

The coupling syntax for slave axes is the same as shown in sect. 4.73.2.

With G581 <master name>1..., you can expand the group of axes defined by the master name by slave axes 1 through n. You may also use this syntax to change the coupling characteristics of an existing slave axis in this group.

CAUTION

Expanding a group of axes with NC block "G581 <master name>1 ..." will cause a traversing motion of all slave axes programmed in this block.

Each slave axis will traverse to its specific coupling position (reference value) defined by the position of the master axis and the coupling characteristics.

The following actions are carried out with G581 <master name>1 ...

- Slave axes 1 through n stated in the syntax are deleted in other groups of axes
- Expansion of the <master name> group of axes by slave axes 1 through n.

4.73.4 Reducing an existing group of axes

One or several slave axes can be removed from an existing group of axes.

Programming

G581: Reducing an existing group of axes with linear or freely defined coupling

G581 <master name> - i <slave name i>()...<slave name n>()

The group of axes defined by <master name> is reduced by slave axes "slave name i" through "slave name n" (with i = 1...n).

You do not have to enter the coupling characteristics in the syntax for the slave axes.

Example:

G581 Z–1 A() B() Slave axes A and B are removed from the Z group of axes.

G581 Z–1 If no slave axes are programmed, the whole group of axes is disbanded.
4.73.5 Disbanding all groups of axes

Programming

G580: With G580, all existing groups of axes are disbanded.

Apart from G580, disbanding of all groups of axes is possible by repeated programming of "G581 <master name>–1". In the latter case, however, block preparation of axis coupling will remain active. G581 will remain the active G function displayed on the operator interface.

Example:

G581 Z0 A(4,2) B(2,1) Z group of axes
G581 Y0 C(3,1) Y group of axes
...
G580 The whole group of axes is disbanded.

or:

G581 Z–1 Disband Z group of axes
G581 Y–1 Disband Y group of axes.

4.73.6 Coupling table

A coupling table contains the coupling characteristics in the form of a coupling function between slave and master axes. The coupling relationship is described in the form of a coupling function:

\[
p_s = f(p_m - p_m^o) + k + o
\]

- \(p_s\): computed position of the slave axis
- \(f\): coupling function (in the form of a coupling table)
- \(p_m\): shift of the master axis
- \(p_m^o\): computed position of the slave axis

The user defines the coupling function \(f(p_m)\) in the form of a table with pairs of interpolation points, \((p_{mi}, f_i)\) \((i=1,...,n)\).

Based on these pairs of interpolation points, the NC interpolator computes the values of the function between the interpolation points and thus the position of the slave axis.

The following approximations may be selected for the computation of the positions between the interpolation points:

- **linear** – as a line between two interpolation points, or
- **cubic spline** – as a spline curve between two interpolation points with the previous and the following interpolations points taken into account (refer also to sect. 4.73.7).
Spline approximation is to be preferred whenever a **curved** shape is desired and no data on the exact shape is available. Cubic spline approximation allows a curve shape with smooth transitions at the interpolation points.

**Structure of the coupling table**

The coupling table is structured as follows:

- **#1**: type of interpolation
  - 1 = linear
  - 3 = cubic spline
- **#11**: unit of $p_{mi}$ values
  - –3 mm, –2 cm, –1 dm, 0 m,
  - 1 inch, 2 degree, 3 rad
- **#12**: unit of $f_i$ values
  - –3 mm, –2 cm, –1 dm, 0 m,
  - 1 inch, 2 degree, 3 rad
- **#20**: periodic
  - 0 = non-periodic
  - 1 = periodic
- **#100** $<p_{m1}><f_1>$ 1st pair of interpolation points
- **#100** $<p_{m2}><f_2>$ 2nd pair of interpolation points
- ... $n$th pair of interpolation points
- ... $n$th pair of interpolation points

Any tables addressed by the NC program will be searched on the subprogram path (compiled table: e.g. /usr/lnk/cam.fct.s).
Coupling table parameters:

#1 defines the type of interpolation to be used between the interpolation points. The types available are linear interpolation (value = 1) or cubic spline interpolation (value = 3). The default value is 1, i.e. linear interpolation.

#11 defines the unit of the \( p_m \) values. If you specify units of length (values -3 through +1) here, the table may be used for linear master axes only. Accordingly, if you specify angular units (values 2 and 3), the table may be used for rotary master axes only. The default value is -3 (mm).

#12 defines the unit of the \( f \) values (the same units as with #11). In tables to be used for linear or rotary slave axes, no other values than -3 through +1 or 2 and 3 may be specified. The default value is -3 (mm).

#20 defines whether the coupling function is to be periodic or non-periodic. If the coupling function is to be periodic (value = 1), the last \( p_m \) value defines the period. If the position of the master axis exceeds the period, function value \( f(p_m) \) is determined by means of a modulo calculation for \( p_m \) to fall within the periodic interval. For non-periodic coupling functions, modulo calculation is deactivated. The default value is 0, i.e. non-periodic.

Please note the following rules:

- #20 = 0, **non-periodic:**
  The limit switch range of the master axis is restricted to the periodic interval \([p_{m1}, p_{mn}]\). No modulo axes (linear modulo axes or endless axes) may be used as master axes.

- #20 = 1, **periodic:**
  The \( p_m \) values must begin with 0, i.e. \( p_{m1} = 0 \). The last \( p_m \) value defines the period. The \( f \) values of the first and the last pairs of interpolation points must be identical, i.e. \( f(p_{m1}) = f(p_{mn}) \). If the master axis is a modulo axis, the \( AxModVal \) mod cycle must be 0, with \( AxModVal \) being the axis-specific modulo value (drive parameter).

#100 defines a pair of interpolation points. You may enter any number of interpolation points in the table. The \( p_m \) values must be entered in ascending order.

; The comment indicator to be used in the table is the semicolon.
Example:
A camshaft with two cams of identical shape and with 180° rotational offset from each other is to move two tappets.

Mechanical design of the camshaft:

Interpolation points of the cam coupling table:

The shape of the two cams is described by eight pairs of interpolation points (see coupling table “cam.fct”)

Coupling table:
Various angular positions $\alpha_i$ of the camshaft are assigned cam radius values $r_i$. This results in a coupling table “cam.fct” with the following contents:

```
#1 3 ; cubic spline approximation
#11 2 ; unit of the $p_m$ values is degree
#20 1 ; periodic coupling function
#100 0.0 30.0 ; 1st pair of interpolation points ($\alpha_1$, r)
#100 30.0 28.0 ; 2nd pair of interpolation points ($\alpha_2$, r)
#100 90.0 24.0
#100 135.0 22.0
#100 180.0 20.0
#100 225.0 22.0
#100 270.0 24.0
#100 330.0 28.0
#100 360.0 30.0 ; last pair of interpolation points
```

Syntax: A blank or a TAB is to be inserted between the values.
Establishing the coupling relationship:
The two tappets with lengths $l_1$ and $l_2$ are represented by two linear axes $Z_1$ and $Z_2$. The camshaft proper is assumed to be an endless axis, called $A$. The zero point of the linear axes is assumed to be located in the center of the shaft.

**Note:** There is no need to have the camshaft physically represented by an axis. Because the Typ3 osa does not support virtual axes, axis $A$ must be entered in MACODA. However, it may be suppressed in the SERCOS interface ring.

Thus, the coupling relationships within the group of axes are as follows:

The following applies to position $p_s$ of axis $Z_1$:

$$p_s = f(p_{mt} + 90) + l_1$$

The master shift in this case is $-90^\circ$ because, with camshaft $A$ in an angular position, slave axis $Z_1$ is to be in this position (see fig. 1 and fig 2).

The following applies to position $p_s$ of axis $Z_2$:

$$p_s = f(p_{mt} - 90) + l_2$$

The master shift in this case is $+90^\circ$ because the two cams show a rotational offset from each other of $180^\circ$.

The axis coupling is generated in the NC program with the following syntax:

```
G581 A0 Z1(<l1>,,–90.0,，“cam.fct”) Z2(<l2>,,90.0,，“cam.fct”)
```

When this coupling syntax is interpreted during block preparation, the corresponding spline table `/<link directory>/cam.fct.s` is created (see next section).

### 4.73.7 Creating a spline table file

**Effect**

The spline table is a **separate** table. The spline table is created on the basis of the pairs of interpolation points contained in the coupling table

- whenever the "coupling table name" appears in a syntax unless this spline table already exists.

The NC block interpolator accesses an image of the spline table and – on the basis of this table – calculates the function values between the various interpolation points and thus the position of the slave axis.

Spline tables are created during block preparation when the coupling syntax is being interpreted. Spline tables are stored in the link table directory. Spline tables have the same names as coupling tables, with the exception of their extension ".s".

**Example:** coupling table: `curve.fct` —> spline table: `curve.fct.s`

A spline table is created only if it does not already exist or if its time stamp is older than the corresponding coupling table.

**Note:** The default link table directory is `/usr/lnk`. However, it may also be freely defined using MACODA parameter 30800004.
Creating a spline table file while the program is running:

1st step:
Creating coupling table(s) "fsi" ... "fsn":

```
#1  3 ; cubic spline approximation
#11 2 ; unit of the p_m values is degree
#20 1 ; periodic coupling function
#100 0.0 30.0 ; 1st pair of interpolation points (α_1, r_1)
#100 30.0 28.0 ; 2nd pair of interpolation points (α_2, r_2)
#100 90.0 24.0
```

2nd step:
Creating a group of axes with freely defined coupling (via coupling table):

```
G581<master name>0 <slave name i>({<o_si>,<k_si>,<p0>,<f_si>})
...{<slave name n>({<o_sn>,<k_sn>,<p0>,<f_sn>})}
```

3rd step:
Start the program

4th step (automatic):
With the program running and while G581 is interpreted, the NC creates automatically one or several spline table files on the basis of the coupling table(s) "fsi" ... "fsn".

Programming

G582: Creating a spline table via G582

```
G582  STAB(<"coupling table name">,{<1|0>})
```

Explanation:

- **coupling table name**
  - The coupling table is searched on the sub-program search path and the corresponding spline table is created in the link table directory.

- **1|0**
  - If "0" is entered here, the spline table is created only if it does not already exist or if the existing spline table is older than the coupling table. If you enter "1" instead, a spline table is created under any circumstances. The default value is "0".

With G582, you can create a spline table even without an existing group of axes (e.g. in Manual Data Input mode).
Example:
G582 STAB("curve.fct") creates the spline table, if required
/\<link directory>/\<name>.s
G582 STAB("curve.fct",0)
G582 STAB("curve.fct",1) creates the spline table irrespective of the
time stamp and whether the spline table already exists or not/\<link direc-
tory>/\<name>.s

Please note for G580 and G581:

- All axes within a group of coupled axes must be on the same channel and
  may be assigned to this group of coupled axes only.
- A maximum of 7 slave axes is permissible for each group of coupled
  axes.
- You may define and program additional groups of coupled axes. Slave
  axes are automatically integrated in a new group of axes or, respectively,
  deleted from an existing group of coupled axes.
- No slave axis may be the master axis of any third axis at the same time
  (multi-stage coupling is not permitted!).
- **Referencing:** Deactivate coupling prior to referencing to facilitate refer-
  encing of each axis individually.
- Existing axis couplings are retained after the end of a program.
- **Program value display:** The slave axis display depends on the values of
  the master axis, taking the coupling relationship into account.
  The display of the slave axes can be suppressed by MACODA settings.
- **Limit switches:** When a group of axes is coupled, the permitted travers-
  ing range of the master axis may be reduced if a slave axis has a smaller
  traversing range than the master axis, or if the shift \( \delta \) or the coupling
  factor \( k_s \) (electronic gearbox) are set accordingly. In this case, the traversing
  range of the master axis will be given **new limit switch values.**
- **Limit switch suppression:** Limit switches can be suppressed both for
  master axes and for slave axes with the result that limit switches have no
  effect on master and/or slave axes.
- **Axis dynamics:** When coupling is active, the dynamics of the weakest
  axis and/or the velocity relationships resulting from the coupling relation-
  ships between master and slave axes (e.g. \( V_m=25 \), \( V_s=100 \), with a cou-
  pling factor of 1:4) will determine the dynamics of the whole group. The
  maximum admissible speed of the group equals the lowest maximum ad-
  missible speed of all axes within the group of coupled axes. The same
  applies to the maximum admissible acceleration.
- **Axis inhibit/Test mode:** All axes within a group of coupled axes must be
  in the same mode. Axis inhibit of individual axes within a group of coupled
  axes is not permitted. Activating a couple in test mode is only permitted if
  test mode is switched on with active coupling (since the slave axis might
  not be in coupling position when test mode is switched off, a setpoint jump
  may occur which may cause a drive error!). Therefore, the PLC must en-
  sure that the interface signals act on **all** axes within a group of coupled
  axes. Axes coupled in test mode must be uncoupled before test mode is
  switched off.
- Programming a traversing motion of slave axes is **not** permitted.
### G594: Suppressing axes for feedrate computing and separate programming of rotary axis feedrate

#### Effect

Function G594 is designed to exclude individual axes from the feedrate computation by setting the respective machine parameter. The feedrate programmed in the part program refers exclusively to the axes contributing to the feed function. Those axes not taken into account for computing the feedrate follow synchronously.

MACODA parameter 10030002 0 allows to determine statically for each synchronous system axis whether it is to be included in feedrate computing or not when G594 is active. Once both axes contributing to the feed function and axes not contributing to it have been programmed in an NC block, the programmed feedrate refers only to those axes contributing to the feedrate. All other axes (not contributing to the feedrate) follow synchronously in the interpolation process. This means that all axes start and finish traversing at the same time. Also all acceleration and deceleration processes are locked synchronously.

Just like the axes contributing to the feed function, also those axes not contributing to it may cause a reduction of the programmed feedrate block by block if otherwise an axis limit is exceeded.

This function may be used to suppress all rotary/endless axes in the feedrate computation. The programmed F value is then interpreted just as a tool feed. With G594 activated, each feedrate value programmed (F word) is copied to 2 internal velocity values (feedrate1 and feedrate2). While inch/metric switching (G70/G71) applies to feedrate1 (for axes contributing to the feed function), this is not the case with feedrate2. If a programmed NC block does not contain any other axes that those excluded from the feedrate computation, those axes are traversed at feedrate2. As in the example below, the feedrate programmed last in [degrees/min] is interpreted.

Another example of an application of this function are machine kinematics with parallel axes that are not coupled:

**Example:** 2 parallel Z axes

If you suppress the display of the feedrate of one of these parallel axes, with parallel programming, only one of these Z axes is taken into account for computing the feedrate, i.e. the effective feedrate remains constant, no matter whether one or two Z axes are involved in a parallel motion.

If the function “Tapping without compensation chuck” is used with axes for which the display of the feedrate is suppressed, the feedrate display of the C axes (spindles) involved in the tapping must also be suppressed in MACODA when G594 is active in order to obtain the feedrate programmed in the part program.
Example: G594

The following configuration conditions must be set in MACODA:

<table>
<thead>
<tr>
<th>Axis movement type (MACODA 10030000 4)</th>
<th>axis not contributing to feed function (MACODA 100030002 0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>axis 1 linear</td>
<td>no</td>
</tr>
<tr>
<td>axis 2 linear</td>
<td>no</td>
</tr>
<tr>
<td>axis 3 linear</td>
<td>no</td>
</tr>
<tr>
<td>axis 4 rotary</td>
<td>yes</td>
</tr>
<tr>
<td>axis 5 rotary</td>
<td>yes</td>
</tr>
</tbody>
</table>

Programming example with the settings made in parameter 100030002 0:

N1  G70 G1 G594 F100
F100: feedrate1=100 inch/min=2540mm/min;
    feedrate2=100 degrees/min
N2  X200
   X axis moves at 100 inch/min
N3  Y200 B200
   Y axis moves at 100 inch/min
   B axis follows
N4  C200
   C axis moves at 100 degrees/min

Please note in general for programming with G594:

<table>
<thead>
<tr>
<th>programmed axes</th>
<th>axes contributing to feedrate computation, having the following feedrate:</th>
<th>axes not contributing to feedrate computation, having the following feedrate:</th>
</tr>
</thead>
<tbody>
<tr>
<td>axes contributing to feedrate computation</td>
<td>feedrate1</td>
<td>--/--</td>
</tr>
<tr>
<td>both contributing and non-contributing axes</td>
<td>feedrate1</td>
<td>follow</td>
</tr>
<tr>
<td>axes not contributing to feedrate computation</td>
<td>--/--</td>
<td>feedrate2</td>
</tr>
</tbody>
</table>

Separate feedrate programming (Omega) for axes not contributing to feedrate computation (e.g. rotary/endless axes) with G594 activated

With G594 activated, axes not contributing to the feedrate computation are interpolated at feedrate2 unless an axis contributing to the feedrate computation has been programmed. Feedrate2 is then equal to feedrate1. Both internal feedrate values are computed from the feedrate programmed (F word).

Additionally, there is the option of programming feedrate1 and feedrate2 independently of each other. With the "Omega" parameter, you can program feedrate2 directly. Omega has no impact on feedrate1 (F value). Once programmed, Omega continues to act modally until a new Omega is programmed. With any Omega programmed other than 0, none of the F values programmed have an impact on feedrate2. Only when you program "Omega 0" will the following F values have an impact on feedrate2 again.
Programming

Example: G594 and Omega

N1 G70 G1 G594 F100 Omega1000
F100: feedrate1= 100 inch/min
= 2540 mm/min;
Omega 1000: feedrate2
=1000 degrees/min

N2 X200
X axis moves at 100 inch/min

N3 Y200 B200
Y axis moves at 100 inch/min
B axis follows

N4 C200
C axis moves at
1000 degrees/min

N5 Omega 0
Omega is cancelled:
feedrate2=0

N6 F50
F50: feedrate1= 50 inch/min
=1270 mm/min;
feedrate2= 50 degrees/min

N7 B200
B axis moves at 50 degrees/min

If a programmed NC block does not contain any axes other than those excluded from feedrate computation, those axes are traversed at the resulting feedrate2.

G595:

Effect

With G595, all programmed synchronous axes of a channel are included in the feedrate computation. Parameter 704000110 allows to adjust the feedrate ratio of linear and rotary axes to typically European or US programming patterns.

These patterns are different insofar as the control, depending on whether European or US measuring units are used, applies metric units or inches to define degrees.

Therefore, localised presetting of the measuring system to be applied in operation is essential because the internal computations performed by the control will then be based on the selected measuring system.

You must select both for G70 (programming based in inches) and G71 (programming based on the metric system) whether 1 degree is to be interpreted as

- 1 mm or
- 1 inch

in the feedrate computation.
### European setting

For controls to be used in accordance with **European** programming patterns, the following must be entered in parameter 704000110 for G70/G71:

<table>
<thead>
<tr>
<th>Definition</th>
<th>Parameter</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>000000 (G70) inch programming:</td>
<td>1</td>
<td>1 degree = 1 mm = 0.03937 inch (or 25.4 degree = 1 inch)</td>
</tr>
<tr>
<td>000001 (G71) metric programming:</td>
<td>1</td>
<td>1 degree = 1 mm</td>
</tr>
</tbody>
</table>

Both with metric and with inch programming, 1 degree is always interpreted **internally** as 1 mm.

- If the rotary axis alone has been programmed with G71, the feedrate is interpreted as being stated in degrees/min (F100 = 100 mm/min or 100 degrees/min).
- If the rotary axis alone has been programmed with G70, the rotary axis will traverse too fast by the factor 25.4 (feedrate value is interpreted as being stated in degrees/min: e.g. F100 = 100 inches/min = 2540 mm/min or 2540 degrees/min).
- If the rotary axis and the linear axis traverse jointly, both are **assigned the same priority** with G71.
  
  **Example:** G71 is preset.
  
  The linear axis and the rotary axis have the same distance to traverse of 100 mm or 100 degrees. Therefore, according to the internal computation, both axes traverse at an axis feedrate of 0.7071 x the programmed tool path feedrate, each, if both distances to be traversed have the same length.

- If with G70 (inch setting) the linear axes traverse by 100 inches and the rotary axis by 100 degrees, the linear axis traverses at almost 100% of the programmed tool path feedrate because the traversed distance of the linear axis is longer by the inch-to-metric translation factor of 25.4 than the distance to be traversed by the rotary axis (see Example: G594).

### US setting

For controls to be used in accordance with **US** programming patterns, the following must be entered in parameter 704000110 for G70/G71:

<table>
<thead>
<tr>
<th>Definition</th>
<th>Parameter</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>000000 (G70) inch programming:</td>
<td>2</td>
<td>1 degree = 1 inch</td>
</tr>
<tr>
<td>000001 (G71) metric programming:</td>
<td>2</td>
<td>1 degree = 1 inch = 25.4mm (0.03937 degree = 1 mm)</td>
</tr>
</tbody>
</table>

Both with metric and with inch programming, 1 degree is always interpreted **internally** as 1 inch.

- If the rotary axis alone has been programmed with G70, the feedrate is interpreted as being stated in degrees/min (F100 = 100 mm/min or 100 degrees/min).
- If the rotary axis alone has been programmed with G71, the rotary axis will traverse too slowly by the factor 25.4 (feedrate value is interpreted as being stated in degrees/min: e.g. F100 = 100 mm/min = 3.937 inches/min or 3.937 degrees/min).
- If the rotary axis and the linear axis traverse jointly, both are **assigned the same priority** with G70.
  
  **Example:** G70 is preset.
The linear axis and the rotary axis have the same distance to traverse of 100 mm or 100 degrees. Therefore, according to the internal computation, both axes traverse at an axis feedrate of 0.7071 x the programmed tool path feedrate, each, if both distances to be traversed have the same length.

- If with G71 (metric setting) the linear axes traverse by 100 mm and the rotary axis by 100 degrees, the rotary axis traverses at almost 100% of the programmed tool path feedrate because the traversed distance of the rotary axis is longer by the inch-to-metric translation factor of 25.4 than the distance to be traversed by the linear axes (see Example: G594).

### Universal setting

For controls to be used in accordance with both US and European programming patterns, the following must be entered in parameter 704000110 for G70/G71:

<table>
<thead>
<tr>
<th>Definition</th>
<th>Parameter</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>000000 (G70)</td>
<td>2</td>
<td>1 degree = 1 inch</td>
</tr>
<tr>
<td>inch programming:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>000001 (G71)</td>
<td>1</td>
<td>1 degree = 1 mm</td>
</tr>
<tr>
<td>metric programming:</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

With metric programming, 1 degree is interpreted internally as exactly 1 mm, with inch programming, 1 degree is always interpreted internally as 1 inch.

- Both with G70 and G71, the feedrate is interpreted as being stated in degrees/min if only the rotary axis has been programmed.
- If the rotary axis and the linear axis traverse jointly, they are assigned the same priority, both with G70 and G71.

**Example:** G70 is preset.
Both axes have the same distance to traverse: 100 inches or 100 degrees. Therefore, according to the internal computation, both axes traverse at an axis feedrate of 0.7071 x the programmed tool path feedrate, each, if both distances to be traversed have the same length.

**Example:** G71 is preset.
Both axes have the same distance to traverse: 100 inches or 100 degrees. Therefore, according to the internal computation, both axes traverse at an axis feedrate of 0.7071 x the programmed tool path feedrate, each, if both distances to be traversed have the same length.

Please note for G594 and G595:

- Functions G594 and G595 act modally and deselect each other mutually.
- If neither G594 nor G595 is selected, the effect is the same as with G595 with the MACODA parameter settings 000000:1 and 000001:1 (European setting!)
- G594 and G595 may be programmed together with other path conditions, traversing information and auxiliary functions in one block.
4.75 Stroke release time (default values) G610
Stroke release time (inpos window reached) G611
Stroke release time (interpolation end point reached) G612

Effect

The time when the stroke is to be released in a punching or nibbling process (see sect. 4.77) can be influenced with G610 through G612. This is a way to activate the stroke release early (which is the rule), e.g.

- to save time by releasing the stroke while the positioning axes are still moving.

or to release a stroke not before the axes have come to a safe standstill, e.g.:

- to achieve increased positioning accuracy (for axes with poor dynamic characteristics)
- for punching with holding-down appliances.

The stroke release time required for the above purposes can be programmed as a waiting time which starts to run as soon as the respective axis reaches a specific zero point in time (time referenced point).

Two different zero points in time (time referenced points) may be used:

- zero point in time = time when all axes have reached the inpos window
  Only when all axes involved in the traversing motion have reached their respective inpos windows and the programmed waiting time has elapsed will the stroke be released.

- zero point in time = time when the interpolation has reached the end point
  The time when the NC interpolation has reached the end point of the traversing motion is taken to be the zero point in time. In this case, you can program both positive times (delayed stroke release) and negative times (early stroke release).

![Stroke release with G612](inpos window reached)

- direction of axis motion
- inpos window
- stroke release without G612
- punching die stroke
- stroke release with G612
- t0 (time referenced point)
- release time
- t1 (release point in time)

![Stroke release with G611](interpolation has reached the end point)

- direction of axis motion
- interpolation has reached the end point
- stroke release without G611
- punching die stroke
- direction of stroke
- stroke position
- stroke release with G611
- t0 (time referenced point)
- t1 (release point in time)
- release time
What you need to program is the time referenced point for each axis. With every punching block, the "weakest" axis of all axes involved in terms of release time and time referenced point determines the stroke release.

**Programming**

**G611**: Stroke release time (interpolation has reached the end point)

G611 <axis name i><time axis name i>...<axis name n><time axis name n>

**Explanation:**
- **axis name i**: Name of the i\(^{th}\) logical axis to which a stroke release time is to apply with "interpolation has reached the end point" as the time referenced point.
- **time axis name i**: Waiting time (in ms) of the i\(^{th}\) axis referenced to the interpolation reaching the end point where a stroke is to be released.

**Condition**: time axis name i \(\geq 0\)

\[i = 1 \ldots n\]

**Programming**

**G612**: Stroke release time (inpos reached)

G612 <axis name i><time axis name i>...<axis name n><time axis name n>

**Explanation:**
- **axis name i**: Name of the i\(^{th}\) logical axis to which a stroke release time is to apply with "inpos window reached" as the time referenced point.
- **time axis name i**: Waiting time (in ms) of the i\(^{th}\) axis referenced to the time the inpos window is reached, where a stroke is to be released.

**Condition**: time axis name i \(\geq 0\)

\[i = 1 \ldots n\]

**Programming**

G610 Setting the MACODA stroke release times (parameters 800100010, 800100020, 800100021) for all axes (default values of times and time referenced points).

**Please note for G610, G611 and G612:**
- The times programmed are not subject to any restrictions. Therefore, you can program an early stroke release by setting a negative value.
- All time values entered are brought up to match the SERCOS cycle time slots. For each traversing motion, the stroke release behavior is determined by the "weakest" axis. First the maximum reference and then the maximum time is computed for all axes involved in the motion.

Please note for the maximum reference:
"Inpos window reached" is greater than "Interpolation has reached the end point".
Example:
- Axis configuration: X, Y, C
- SERCOS cycle time: 3 ms

Initializing the reference axis:

<table>
<thead>
<tr>
<th>Code</th>
<th>Time referenced point</th>
<th>Programmed release time</th>
<th>Release time with SERCOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>N10 G612 Y10 C2</td>
<td>&quot;inpos reached&quot;</td>
<td>Y axis: 10 ms; C axis: 2 ms</td>
<td>Y axis: 12 ms; C axis: 3 ms</td>
</tr>
<tr>
<td>N20 G611 X–10 ..</td>
<td>&quot;Ipo end point reached&quot;</td>
<td>X axis: –10 ms</td>
<td>X axis: –9 ms</td>
</tr>
</tbody>
</table>

During machining:

<table>
<thead>
<tr>
<th>Code</th>
<th>Stroke release</th>
<th>Time referenced point and release time</th>
</tr>
</thead>
<tbody>
<tr>
<td>N30 G1 Y20 C10</td>
<td>Y determines the stroke release behavior (maximum time!).</td>
<td>time referenced point = &quot;inpos reached&quot;; release time = 12 ms</td>
</tr>
<tr>
<td>N40 X20 C20</td>
<td>C determines the stroke release behavior (maximum reference!).</td>
<td>time referenced point = &quot;inpos reached&quot;; release time = 3 ms</td>
</tr>
<tr>
<td>N50 X30</td>
<td>X determines the stroke release behavior (only X axis traversing!)</td>
<td>time referenced point = &quot;Ipo end point reached&quot;; release time = –9 ms</td>
</tr>
<tr>
<td>N60 G610</td>
<td>All axes are assigned MACODA values</td>
<td>All axes are assigned MACODA values</td>
</tr>
</tbody>
</table>
4.76 Tangential tool orientation ON
Tangential tool orientation OFF
Tangential tool orientation ON
Tangential tool orientation OFF

Effect

Function G631, “Tangential tool orientation” is designed to set a punching
die tangentially to the path with every stroke (possible only when G661,
punching, or G662, nibbling, is active).

In all blocks or block segments, the orientation axis of the tool reaches its full
tangential angle normally at the end point of the path (unlike with “Tangen-
tial tool guidance”, see sect. 4.55). The orientation axis moves synchronously
with the linear axes arriving with them at the end point simultaneously.

However, if an additional stroke is released (see para 1.2, “Nibbling”) at the
beginning of the first path segment, a block inserted specifically for this pur-
pose turns the orientation axis to the tangential angle of the first block seg-
ment. This is to ensure correct tool orientation with every punching stroke.

Together with every orientation motion, the path search logic “shortest path”
is active. If symmetry values greater than 1 occur, the nearest equivalent
angle will be approached.

Programming

G631:  Tangential tool orientation ON

G631 {SYM<s>} {ANG<a>} or
TTON {SYM<s>} {ANG<a>}  (syntax alternative)

Explanation:

SYM  SYM defines the symmetry value s. A tool with the symmetry
value s returns to its original position after completing a 360°/s
rotation.

s  Any positive integer may be entered for the symmetry value s.
It depends on the type of tool used, i.e. a rectangular tool has
the symmetry value s = 2, a square tool the symmetry value
s = 4, etc.

Default: If no entry is made for SYM, the default value for s=1.

ANG  ANG is used to enter the offset angle a with the path tangent.

a  The offset angle may be within the half-open range
(a > −180° through a ≤ 180°).

Default: If no entry is made for ANG, the default value for a=0.

Programming

G630:  Tangential tool orientation OFF

G630 or TTOFF  (syntax alternative)

Please note for G630, G631 or TTON/TTOFF:

- G631 and TTON/TTOFF can be applied only in combination with punch-
ing, G661, or nibbling, G662 (see also sect. 4.77).
- Although function G631 may be programmed while G660 is active, tan-
gential axis setting will be executed only for punching/nibbling blocks.
- The number of the orientation axis is defined by MACODA parameter
705000210, i.e. it is not programmable.
- For circular blocks with block splitting, the tangent to the circular contour is taken as the basis of orientation with each stroke, while the traversing motion from one stroke to the next is linear.
- Functions "Tangential tool orientation" and "Tangential tool guidance", G130, G131, must never be active simultaneously.

**Example:**

```
N10 G660 X0Y0C0
N10 G631 Tang. tool orientation ON, symmetry 1, offset angle 0
N20 G1 G91 X10 Y10 G660 is still active → no orientation of the C axis
N30 G661 X10 Y10 C end point = 45°
N40 Y–10 C end point = −90°
      After the modulo computation, C is set to 270°
N50 G660 Punching OFF
N60 G662 LEN=30 Nibbling ON, with 30 mm path segments
N70 G2 X114.6
     I57.3J0 Semicircle with 180 mm length of arc:
       – C rotates from 270° to 90°; stroke at the starting point
       – 1st block segment, C from 90° to 60°
       – 2nd block segment, C from 60° to 30°
       – 3rd block segment, C from 30° to 0°
       – 4th block segment, C from 0° to −30° (= 330°)
       – 5th block segment, C from 330° to 300°
       – 6th block segment, C from 300° to 270°
       (see also fig. below)
N80 G630 Tangential tool orientation OFF
N90 X200 NUM=2 C remains at 270°
```

- C axis rotation with tangential tool orientation

![Diagram of C axis rotation with tangential tool orientation](image-url)
### 4.77 Punching/Nibbling OFF

Punching ON  
Nibbling ON

<table>
<thead>
<tr>
<th>Effect</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>When G661 or G662 is active, a punching stroke is released at the end of every path segment. The subsequent traversing motion is not started before the stroke is finished.</td>
<td></td>
</tr>
</tbody>
</table>

Basically, punching and nibbling have the same functionality. The differ in the following respects:

- **Stroke release**
  - For punching (G661), strokes are always released at the end of the paths or path segments.
  - For nibbling (G662), there are two cases where an additional stroke is released at the beginning of the first path segment:
    - if the previous block does not contain any traversing motion, or
    - if G660 was active in the previous block.

- **Block splitting**
  - Block splitting need not be active for punching. In this case, every NC block is concluded at its end with a stroke (individual stroke operation).
  - For nibbling, it is mandatory that you program block splitting — either by entering a modal LEN value or a local NUM value. Individual stroke operation is not possible for nibbling.

Block splitting means that the length of a path on the selected plane is split in block segments of equal size. Any additional axes programmed outside this plane will reach their respective end point not before the last block segment (see Example 3).

- **No** stroke is released in the following cases:
  - No axis coordinate on the selected plane has been programmed (e.g. N..C60). In this case, execution continues without a stroke being released.
  - Stroke release is suppressed by the PLC. Execution is suspended at the point of the stroke until the stroke is released by the PLC.

- **The time when a stroke is to be released can be influenced via G611 and G612 (see sect. 4.75).**

### Programming

| G661 | Punching ON  
Together with G661, path splitting may be programmed by entering LEN and NUM values. |
|------|-------------|
| G662 | Nibbling ON  
Together with G662, path splitting must be programmed by entering LEN and NUM values. |
| G660 | Punching/Nibbling OFF |

- **LEN=<value>**  
  "value" = length of the path segment of an NC block

- **NUM=<value>**  
  "value" = number of path segments of an NC block
Please note for G661, G661 and G662:

These three NC functions form a modal group, i.e. at any given point in time, no more than one of these three NC functions is active.

Punching or nibbling functions can be activated only if the desired function is set in MACODA parameter 800100010 and if the path shape (G408 or G608) in the NC program is active.

Please note for LEN:

- Defines the lengths of the path segments of each NC block.
- May be any positive number. When LEN=0, block splitting is deactivated.
- You do not have to enter the exact divisor of the programmed length of path. Internally, the NC computes an effective LEN value ≤ the LEN value entered with the effect that the effective path segments are equal divisions of the programmed length of path.
- The unit for programming is the same as for the axis coordinates.
- Applies both to linear and to circular blocks. In the latter case, the LEN value refers to the length of the arc. However, the traversing motion from one stroke to the next is always linear.
- May be overwritten block-by-block by NUM (see below).
- Can be programmed while G660 is active. However, block splitting begins only after punching or nibbling have been activated.
- Acts modally, i.e. the segmentation of the path applies to all subsequent NC blocks as long as G661 or G662 is active.

Please note for NUM:

- Defines the number of path segments of an NC block.
- May be any positive integer. NUM=0 is not permitted. NUM=1 does not produce any block splitting.
- Acts locally, i.e. only in the NC block where it is programmed.
- Overwrites block-by-block any path segmentations programmed with an LEN value.
- May be programmed only while either punching or nibbling is active.
**Example:** Punching and nibbling

**Punching:**

N10 C10 G661

N20 C60
N30 X0
N40 LEN=12
N50 X110
N60 Y30 NUM=3
N70 Y90
N80 X50 Y50 G660

**Punching** is activated. No stroke is released because X and Y are not programmed.

No stroke

Stroke because X is programmed (no traversing motion)

Path segmentation, 12 mm

Block is split in 10 path segments of 11 mm, ea.

Strokes are executed in positions X11, X22, ..., X99, X110

LEN=12 is overwritten block by block. Strokes at Y10, Y20, Y30

Modal LEN active again

Strokes at Y42, Y54, Y66, Y78, Y90

Punching OFF. No stroke is released.

**Nibbling:**

N01 X0 Y0 C0 G90 G1
N30 C10 G662
N40 X0
N50 X110
N60 Y30 NUM=3
N70 Y90
N80 X50 Y50 G660

**Nibbling** is activated. No stroke because X and Y are not programmed.

No stroke because X and Y are not programmed.

Stroke because X is programmed. No traversing motion.

Block is split in 10 path segments of 11 mm, ea.

Additional stroke at X0 because there is no traversing motion in N40.

Further strokes at X11, X22, ..., X99, X110

LEN=12 is overwritten block by block. Strokes at Y10, Y20, Y30

Modal LEN is active again.

Strokes at Y42, Y54, Y66, Y78, Y90

Nibbling OFF. No stroke is released.
Example: Block splitting

N10 G1 X0 Y0 C0 G660  
Length of path segments: 15 mm.  
No block splitting because G660 is active.

N20 X100 Y100 LEN=15  
Punching is activated. Path length of 141.42 mm is split in 10 block segments.  
Stroke positions at (110,110,18), (120,120,36),..., (200,200,180)

N30 X200 Y200 C180 G661  
Path length of 90 mm is split in 6 block segments.  
Stroke positions at (200,215,185), (200,230,190),..., (200,290,210)

N40 Y290 C210  
Punching off.  
Subsequently, no more block splitting.

N50 G660
4.77.1 Stroke release

By setting the high-speed output 0 on the DCIO (Typ3 power supply unit) (HSO = 1), the NC instructs the punching control to release a stroke. The punching control acknowledges the job after releasing the stroke by resetting the high-speed input 0 on the DCIO (HSI = 0). Subsequently, the NC resets the HSO.

There is no traversing motion while the stroke is being executed (HSI = 0). The NC starts the next traversing block when HSI = 1.

4.77.2 Interface signals used in the punching process

It may sometimes be necessary to release individual strokes (if metal sheets get jammed, working position is not properly aligned yet, e.g.). The logic required for this can be applied by means of the PLC.

Effect The PLC can release a stroke by sending an instruction to this effect to the NC via the NC–PLC interface.

Bit signals used between the NC, the punching control and the PLC are as follows:
All signals are available on the "general interface".

**NC outputs:**
- NC O5.0  "stroke planned" With this signal, the NC indicates to the PLC that a stroke is to be released.
- NC O5.1  "stroke not running" The high-speed input HSI–0 is relayed by the punch-HS logic to the PLC.

**NC inputs:**
- NC I1.0  "stroke inhibit" This allows the PLC to prevent HSO–0 setting.
- NC I1.1  "stroke reserved" This allows the PLC to reserve the high-speed output HSO–0 for a stroke release directly by the PLC.
- NC I1.2  "stroke ON" The PLC instructs the NC to release a stroke.

The following signal conditions must be fulfilled at the input of the Ipo-HS logic for a stroke to be released (HSO–0 = 1):

<table>
<thead>
<tr>
<th>Stroke to be released by the NC:</th>
<th>Stroke to be released by the PLC:</th>
</tr>
</thead>
<tbody>
<tr>
<td>NC O5.0: &quot;stroke planned&quot;=1</td>
<td>NC I1.0: &quot;stroke inhibit&quot;=0</td>
</tr>
<tr>
<td>NC I1.0: &quot;stroke inhibit&quot;=0</td>
<td>NC I1.1: &quot;stroke reserved&quot;=1</td>
</tr>
<tr>
<td>NC I1.1: &quot;stroke reserved&quot;=0</td>
<td>NC I1.2: &quot;stroke ON&quot;=1</td>
</tr>
</tbody>
</table>
4.78 Programming SERCOS ID numbers while in a part program

**Effect**

Using the **G900** function, SERCOS ID numbers can be directly programmed while in a part program.

Both
- manufacturer-independent ID numbers (S-0-..) and
- manufacturer-specific ID numbers (P-0-..)

may be programmed provided that the respective number is permitted in SERCOS phase 4.

**Programmung**

- **G900** S-0-... X... Y... Z... manufacturer-independent ID numbers
- **G900** P-0-... X... Y... T... manufacturer-specific ID numbers

**Examples:**

N... G900 P-0-0500 X100 Y100 Z99
Writing the feed forward control parameters (Bosch)

N... G900 P-0-0501 X50 Y47 Z22

**DANGER**

Improper changes in SERCOS parameters may cause damage to workpieces and/or the machine and pose a hazard to personnel due to unexpected machine reactions.

The G900 function must not be programmed in machining sections because sudden speed reductions (downslope) may occur.
Do not program ID numbers before axes have come to a standstill!

Improper changes in SERCOS parameters may lead to conditions that can only be corrected by a control restart or a drive reset.

Incorrect programming of a SERCOS ID number will produce a runtime error and may have the following causes:
- an invalid ID number
- an ID number cannot be programmed in SERCOS phase 4

or is write-protected:
- The permitted limit values of the parameter have been exceeded.
- An attempt was made to program a non-SERCOS axis.
- Several ID numbers were programmed at the same time.

The parameters set remain valid until they are overwritten, e.g. by a run-up phase.

**Read-only access is possible via the CPL command SCS.**

When writing ID numbers with G900, please note the specific weighting of parameters determined by the drive manufacturer. You can display this weighting in the group mode "Diagnostics" on the SERCOS monitor!
The "Control area" function monitors traversing motions of machine axes as to whether or not their positions are within the specified rectangular, two-dimensional areas with paraxial boundaries. In the event of a violation of a specified area, this function generates a runtime error.

Up to 10 areas can be set in MACODA (NC function parameters/control areas 80020001 – 80020033) and may be defined as:

- **dead ranges** which must not be crossed over or touched upon by a traversing motion. No starting or end points must be located in a dead range. The area of a dead range extends up to and includes its boundaries!
- **working ranges** the boundaries of which must not be exceed by any traversing motion. Starting and end points must be within the working range. The area of a working range extends up to and includes its boundaries!

Validity of areas:
- Every area used is to be assigned two system axes in MACODA. These axes define the plane on which the control area is located.
- Numbering of system axes starts with 1 for the 1st system axis! The designation of these axes is **system**-specific as opposed to channel-specific and, therefore, unique within the whole system!
- Every area is unique throughout the system. On every channel, an area can be monitored only if the system axes of both monitoring dimensions are located on the very same channel.
- The axes of an area that is activated must be on the channel because otherwise a runtime error will occur.
- In the case of an **axis transfer** it must be ensured that every area containing one of the transferred axes is **deactivated** on the channel from which the axes were transferred.
- The transfer of an axis from an active area to another channel will cause a runtime error.
- If the axes of an area are transferred to another channel, this area is assigned the default values of the new channel as preset in MACODA and the area is deactivated! The values set on the original channel are not transferred in the process. (This applies also when the axes are retransferred to their original channel!)
- When you activate the jogging mode, the values last activated apply to every area.

Default status (start-up):
- The control area data is input as default values from MACODA. The control areas are not active.

Control reset:
- The data and status of the control areas are retained unless they are expressly modified by a syntax in the init string.
Monitoring

All data required to monitor an area can be set in MACODA with the effect that all you have to do is activate the respective area in the part program or the cycle.

The areas are monitored in terms of their absolute machine positions.

**Monitoring in automatic mode:**
- In automatic mode, active areas are monitored only if their system axis dimensions on the channel are located on the selected plane.
- Both linear and circular traversing motions are monitored. Blocks not containing any traversing motion are not monitored.

**Monitoring in jog mode:**
- In jog mode, all active areas are monitored. Monitoring is based on the physical axes.
- Only one of the axes defining a dead area may be jogged at a time. If both axes are jogged, an error will occur.
- If an axis is about to touch upon an area boundary, it stops and a warning is displayed.
- There is no monitoring of handwheel operation.
- While the monitoring function is active, use of the "Inclined plane" function is not permitted if this would involve axes defining the control area.

Programming

**Programming control areas:**
A control area is programmed and takes effect on one channel only.

Every area can be programmed individually with the following parameters:
- Position
- Expansion
- Type
- Activation of the area programmed
- Deactivation of the area programmed

Dimensions must be entered in accordance with the unit of measurement valid in each case.
Modifying area i and programming "activation" or "deactivation" at the same time:

N.. AREADEF (i,k,type, position1, position2, expansion1, expansion2)

Explanation:

<table>
<thead>
<tr>
<th>i</th>
<th>i=1..10</th>
</tr>
</thead>
</table>
| k   | 1: modification of the ith area, including "activation"
     | 0: modification of the ith area, including "deactivation" (without activation) |
| type| 0: not defined
     | 1: dead range
     | 2: working range |
| position1 | Determination of the center position of the 1st area dimension relating to the number of the 1st system axis set in MACODA parameter 800200001 to define the 1st dimension of the ith area. |
| position2 | Determination of the center position of the 2nd area dimension relating to the number of the 2nd system axis set in MACODA parameter 800200002 to define the 2nd dimension of the ith area. |
| expansion1 | Determination of the expansion of the 1st area dimension relating to the number of the 1st system axis set in MACODA parameter 800200001 to define the 1st dimension of the ith area. |
| expansion2 | Determination of the expansion of the 2nd area dimension relating to the number of the 2nd system axis set in MACODA parameter 800200002 to define the 2nd dimension of the ith area. |

Please note the following for AREADEF:

- The syntax can be applied to a specific area only. If you want to modify more than one area, you must call up the syntax repeatedly.
- **No entries** are required for parameters that are to remain unchanged (provided that there are further parameters following), or these parameters may be **omitted** altogether (if at the very end). Please refer also to the example below.
- If no entries are made for the type, position1, position2, expansion1, or expansion2 parameters, the respective values are retained.
- If a parameter has never been configured before, the value preset in MACODA is applied.
Example:
N10 AREADEF(4,0,100,200) Type of range, expansion 1 and expansion 2 remain unchanged (MACODA or previous settings are applied).

Activating or deactivating control areas:
Control areas can be activated or deactivated by means of the syntax used. However, the position, expansion and type of the respective range must be known before.

Programming
Activating or deactivating the control of the ith area or of all areas on a channel:
N.. AREVALID(i , k)

Explanation:
\[ \begin{align*}
\text{i} & \quad i = 1..10: \text{selecting the ith area} \\
\text{l} & \quad l = -1: \text{selecting all areas on a channel for monitoring} \\
\text{k} & \quad k = 1: \text{activating the monitoring function} \\
\text{k} & \quad k = 0: \text{deactivating the monitoring function}
\end{align*} \]
Settings in MACODA:

There is a box with 10 entries for each of the following area control parameters. This allows you to define 10 areas:

800200001  System axis number of the 1st axis defining the 1st dimension of the area (e.g., 1 for the 1st system axis ...).

800200002  System axis number of the 2nd axis defining the 2nd dimension of the area.

800200011  1st dimension center point of the area [mm], relative to the axis as defined by parameter 800200001.

800200012  2nd dimension center point of the area [mm], relative to the axis as defined by parameter 800200002.

800200021  1st dimension expansion of the area [mm], relative to the axis as defined by parameter 800200001.

800200022  2nd dimension expansion of the area [mm], relative to the axis as defined by parameter 800200002.

800200031  Type of control area:
            0: not defined
            1: dead range
            2: working range

800200032  Area can be modified by reprogramming:
            0: no
            1: yes
4.80 Limitation of the maximum number of prepared blocks

**Effect**
Using the PREPNUM function, the maximum number of blocks prepared by the block preparation function can be limited.

If more blocks have been prepared than specified by the PREPNUM function when PREPNUM is programmed, preparation of additional blocks will be stopped until the number of blocks prepared is identical with the number specified by the PREPNUM function.

For example, the PREPNUM function can be used to control further processing of runtime measuring results in the part program.

**Programming**

<table>
<thead>
<tr>
<th>PREPNUM</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Block preparation is limited to 5.</td>
</tr>
<tr>
<td>0</td>
<td>Block preparation will use the maximum number of blocks prepared available within the NC.</td>
</tr>
</tbody>
</table>

Please note for the PREPNUM function:

- If the number of blocks programmed exceeds the maximum number of prepared blocks available in the NC, block preparation will have the same effect as if PREPNUM 0 had been programmed. The maximum number of prepared blocks available is defined in MACODA parameter 706000110.
5  Spindles

Spindles may be operated
- as individual spindles
- in spindle groups.

For operation, spindles are implicitly
- assigned to channels
- transferred from one channel to another.

A spindle may be operated
- in speed mode
- in position mode, or
- in synchronism with other spindles.

Spindles can be programmed
- in the part program
- by manual data input, or
- via machine functions,

and can be
- set via the interface.

Spindles are programmed using
- M functions
- S functions
- G functions
- special functions for "position mode".

5.1  Individual spindles, spindle groups and channels

Individual spindles and spindle groups are not assigned to a specific channel in the Typ3 osa until a channel makes an implicit "reservation" for the spindles required.

Each individual spindle or spindle group has the following standard functions:
- Turn right, with/without coolant
- Turn left, with/without coolant
- Stop
- Positioning (spindle orientation)
- Automatic gear selection
- Manual gear selection
- Spindle speed programming
- Tapping without compensation chuck (G32)

The functionality of spindle groups is not to be confounded with a group of two or more coupled spindles with position control, which are operated in positional synchronism (see sect. 5.4).

The following contains explanations of spindle groups, channel reservation and all functions that may occur in the context of spindle programming.
5.1.1 Assigning individual spindles to spindle groups

Effect

Each of the 8 spindles that can be configured in the Typ3 osa as a maximum can be assigned to one of a maximum of 4 configurable spindle groups. The same motion setting applies to all spindles of a spindle group. This is also referred to as parallel spindle.

Programming (see sect. 5.2, “Spindle functions”) a whole spindle group requires less parametric programming than entering the parameters of each individual spindle assigned to a group. When programming a spindle group function, the auxiliary functions of both the spindle group function and of the functions of the individual spindles assigned to this group are displayed on the interface, provided that the corresponding bit-coded auxiliary functions have been programmed in MACODA.

Each spindle assigned to a spindle group can be activated both via spindle group functions and via individual spindle functions. If an NC block contains concurrent instructions for spindle groups and for individual spindles, the control unit sends an error message.

Modal assignment of spindles to spindle groups:

Spindle groups are specified by programming them in the part program / via manual data input for each channel individually, i.e. one and the same spindle may be assigned to different spindle groups on different channels. In order to modify a spindle group, the spindles to be newly assigned to this spindle group must be specified, e.g.

Programming

SPG1(1,2,3) means that from now on spindles 1, 2 and 3 are assigned to spindle group 1 on this channel.

(SP = spindle)

At the time a spindle group is being programmed, the spindles required to form this group may still be included in some other groups. However, every spindle may be assigned to any spindle group at any time. When a spindle function is programmed for a spindle group, prior to its activation the respective spindles are checked as to whether they are released by the so-called spindle data management and are thus available for being assigned. This means that if a spindle was previously activated by another channel, this spindle must now be switched to speed mode (no C axis mode) and be in STOP state (M5).
The spindle assignment entered in MACODA for a spindle group can be reset to its original configuration on a channel by programming as follows:

**SPGn(0)**  

\( n = \text{spindle group index 1 ... 4} \)

A spindle group can be disbanded on a channel by entering the value "–1":

**SPGn(–1)**  

No spindles are assigned to the respective spindle group any more on this channel.

**Examples:**

N... SPG1(1,2,3)  

Spindle group 1 consists of spindles 1, 2 and 3.

N... M19  

Spindles 1, 2 and 3 traverse to their respective reference point.

N... SPG2(2,4,5)  

(→ SPG1(1,3))  

Spindle group 2 consists of spindles 2, 4 and 5

N... M19  

Spindles 1 and 3 traverse to their respective reference point.

**Restoring spindle group default settings:**

**SPGALL(0)**  

By entering SPGALL(0), the MACODA default setting for all spindle groups on a channel can be restored.

The default setting of the allocation of individual spindles to spindle groups can be entered under block No. 104000002 in the MACODA. This default settings may be overwritten by proglramming a spindle allocation in the part program / by manual data input.

**Local override of spindle assignments to spindle groups (not yet available):**

By entering an additional parameter, you can set an optional reference to a list with instructions to which spindles a programmed spindle group function is to be applied. This list acts block by block and only on the programmed spindle group function. The local spindle group must always be entered before the spindle function in the respective block. No more than one local spindle group may be programmed in one NC block.

**SPL(2,6) M19**  

Spindles 2 and 6 traverse to their respective reference point.
5.1.2 Reserving spindles and spindle groups for specific channels

<table>
<thead>
<tr>
<th>Individual spindles</th>
<th>Spindle groups</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 2 3 4 5 6 7 8</td>
<td>1 2 3 4</td>
</tr>
<tr>
<td></td>
<td>spindle1</td>
</tr>
<tr>
<td></td>
<td>spindle2</td>
</tr>
<tr>
<td></td>
<td>channels</td>
</tr>
</tbody>
</table>

Channel 1: spindle 5
Channel 2: spindle 6
Channel 3: –
Channel 4: spindle group 1
Channel 5–n: –

Spindle 6 reserved for channel 4

Effect

As a rule, no spindle (spindle group) is permanently assigned to any specific channel. Therefore, all spindles (spindle groups) can be activated from any channel.

Whenever you enter a traversing motion for a spindle on a channel via the part program or manual data input, the spindle concerned is reserved implicitly for that respective channel. This applies irrespectively of whether the input of the traversing motion is made through an individual spindle function or a spindle group function.

Access to reserved spindles from another channel is blocked, i.e. any attempt to select them from another channel will result in a runtime error (exception: conditional spindle release, see below, Programming with SADM).

Until a reserved spindle is released, it can be activated only on the channel (authorized channel) where the reservation originated. Therefore, a spindle is not released and thus does not become available to any other channel until it is stopped on its authorized channel.

channel 1 (authorized channel)
N...
N... G32 Z–20 F500 M3...
: reservation of spindle 1
... NC block preparation on channel 1

channel 2
N...
N... G32 Z–5 F1000 M3...
: reservation of spindle 1 inhibited because it is being accessed by channel 1!
... NC block preparation on channel 2 only after block preparation on channel 1
Making a spindle reservation on a channel:

The following spindle functions produce an implicit spindle reservation for the calling channel:

- M3, M13
- M4, M14
- M19
- G32 (tapping without compensation chuck)
- G196 (constant cutting speed)

Any spindle functions available may be programmed on the channel from where the spindle was reserved:

- spindle speed programming S, S1 – S8, SSPG1 – SSPG4
- M3, M13, M4, M14, M5, M19
- M40, M41–M44, M48
- G32
- G192, G292
- G196

An attempt to activate a reserved spindle from any other than the authorized channel will produce runtime error 2001: “Spindle is used by another channel!”

The following functions are subject to this restriction:

- spindle speed programming S, S1 – S8, SSPG1 – SSPG4
- M3/M13, M4/M14, M19, M5
- M40, M41–M44, M48
- G32
- G192, G292
- G196

If called from the init string of an unauthorized channel, the following functions are suppressed and, therefore, no runtime error will occur:

- M5
- M40, M41–M44, M48
- G196, G97
Releasing a spindle reserved for a channel:

**Programming**

A reserved spindle can be released on the authorized channel:
- by programming M5, or
- at the end of G32 if the spindle was reserved using G32.

**Termination of a part program with M30 or a control reset on the authorized channel has the following effects:**
- deselection of M40 (if active), current gear range remains selected,
- deselection of G196, spindle speed programming is activated, and
- functions entered in the init string (MACODA block 706000020) are activated.

The functions below will produce the following effects:

<table>
<thead>
<tr>
<th>Programming</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>M5</td>
<td>Spindle is stopped if M3/M13, M4/M14 were active or if the spindle was positioned with M19. Subsequently, the spindle is released.</td>
</tr>
<tr>
<td>M40</td>
<td>Selection/Repeat selection of the “Automatic gear selection” function</td>
</tr>
<tr>
<td>M41–M48</td>
<td>Automatic gear selection may have already been deselected (see above). Manual gear selection in the init string does not produce a gear switch.</td>
</tr>
</tbody>
</table>

Transferring a reserved spindle to another channel:

In exceptional cases, it may be necessary to activate a spindle reserved with M3, M4 or M19 from an adjacent channel (part program or manual data input).

In those cases, NC function SADM makes it possible for the authorized channel currently “owning” the spindle to transfer the spindle reservation to another channel.

**Conditional spindle release:**

<table>
<thead>
<tr>
<th>Programming</th>
<th>Effect</th>
</tr>
</thead>
</table>
| SADM Si=0 ... Sn=0 | A channel currently holding a reservation for one or more spindles, which was made by programming M3, M4 or M19, is caused by this command to grant any other channel the right to access the spindle:  
  - With the SADM command, the spindle can now be accessed by any other channel.  
  - The spindle can be activated via machine functions anytime. |

**Activating a conditionally released spindle from another channel:**

<table>
<thead>
<tr>
<th>Programming</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>SADM Si=1 ... Sn=1</td>
<td>This command allows you to activate a conditionally released spindle from another channel via a part program or by manual data input.</td>
</tr>
</tbody>
</table>

i index of the i\textsuperscript{th} spindle (i=1..n)  
n number of available spindles (currently: n\textsubscript{max}=8)
5.2 Functions for individual spindles and spindle groups (M functions) in speed mode

5.2.1 Spindle functions

Effect

Spindle functions can be programmed in the part program or by manual data input for individual spindles or for spindles assigned to a spindle group. Each of the 8 spindles may be assigned optionally to one of 4 spindle groups. Any number of individual spindles and/or spindle groups may be programmed in one NC block.

Syntax

The syntax for every spindle function of every spindle/spindle group is determined in MACODA. Apart from common M functions, the various functions may be assigned freely defined names with up to 8 digits.

- **No distinction is made between individual spindles and spindle groups in the following description of the various spindle functions because they show the same behavior. (Basically, programming a spindle group just means less programming effort.)**

- **The M functions described below are suggestions of MACODA settings. For better transparency, the previous (fixed) M codes are used as default parameters.**

Declaration applying to the documentation below:

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Assignment</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1st spindle group</td>
<td>M3</td>
</tr>
<tr>
<td>2</td>
<td>1st spindle</td>
<td>M103</td>
</tr>
<tr>
<td>3</td>
<td>2nd spindle</td>
<td>M203</td>
</tr>
</tbody>
</table>

**Programming**

Spindle, Turn right (clockwise):

- M3: The spindle(s) start(s) rotating clockwise (viewed when facing the spindle working range).
- M103: The spindle speed can be set via the pertinent S address. The spindle speed may be programmed together with M3 in one and the same block.
- M203: The function remains **modally active** until it is canceled by another command for the spindle(s) concerned.
  This means that after a gear change, e.g., the type of motion is restored that was active previously.
Programming

Spindle, Turn right (clockwise) and cooling ON:

M13 Same as with M3, M103, M203, plus activation of cooling.
M113
M213

Programming

Spindle, Turn left (counterclockwise):

M4 The spindle starts rotating counterclockwise (viewed when facing the spindle working range). Otherwise the same as M204 "Spindle, Turn right".
M104
M204

Programming

Spindle, Turn left (counterclockwise) and cooling ON:

M14 Same as with M4, M114, M214, plus activation of cooling.
M114
M214

Programming

Spindle, Stop

M5 The spindle is stopped. This command remains active until it is canceled by another spindle command.
M105
M205

Programming

Programmable spindle orientation:

M19 The spindle positions itself at a specific angle.
M119 This function may be executed while the spindle is at a standstill or rotating. When at a standstill, the spindle will orient itself along the shortest possible path. When rotating, the spindle will maintain its last direction of rotation.
M219

When this function is activated, the drive – running at a speed below the spindle positioning speed (cf. SERCOS parameter S-0-0222) – switches automatically to internal drive operation mode "Position control". As soon as another spindle command is activated, drive operation mode "Position control" is deactivated (M3, M4, M5). This function may be programmed alone or together with other M or G instructions. However, there must not be any other function programmed for one and the same spindle that would have to run concurrently (e.g. M3, M4, M5).
The spindle orientation function may be programmed in a block with or without the corresponding S word:

- Programming without the S word:
  The spindle positions itself relative to its reference point (see SERCOS interface).

- Programming with the S word (= positioning angle in degrees)
  - The spindle positions itself at the angle specified by the S word, which is relative (i.e. additive) to its reference point. Angles programmed outside the interval \([0^\circ \leq \text{positioning angle} < 360^\circ]\) will be translated by the control unit to match the permitted interval. This has the effect that the spindle will never have to traverse for more than one rotation.
  - If the spindle is already in the correct position, no motion is executed.
Examples: programmable spindle orientation

N... M19 The spindles of the 1st spindle group position themselves relative to their respective reference points.
N... M119 The 1st spindle positions itself relative to its reference point.
N... M219 The 2nd spindle positions itself relative to its reference point.
N... M19 S1= 180° The spindles of the 1st spindle group position themselves at 180°.
N... M119 S1= –180° The 1st spindle positions itself at 180°.
N... M219 S2=370° The 2nd spindle positions itself at 10°.
N... PTEST10 M119 The 1st spindle positions itself relative to its reference point. Subsequently, subprogram “TEST10” is executed.

5.2.2 Gear functions

Effect
The total speed range available on a machine is divided by switched gears into several smaller speed ranges, for which various gear ranges may be defined.

The number of gear ranges (max. 4), their speed limits (min./max speed) and other specific spindle parameters are defined in group 1040 of MACODA.

Prior to changing gears in a program, the display of a message to notify the operating staff to this effect may be advisable.

Gear selection functions have no impact whatsoever on analog spindles.

Programming
Automatic gear selection

M40 By means of the M function, gear selection is programmed once at the beginning of the part program.
M140 On the basis of the programmed speed, the control unit selects the appropriate gear from the maximum of 4 gear ranges that may be programmed in MACODA.
M240 Please note:
- In the event of overlapping speed ranges of the various gears, the control unit will always select the lower gear (with the higher motor speed).
- Programming “0” for the speed will have the effect that no gear changes are carried out.

In MACODA blocks 706000010 and 706000020, you can configure the automatic gear selection function to be the default status.
Examples:
N... M40  Automatic gear selection for 1st spindle group
N... M140 Automatic gear selection for 1st spindle
N... M240 Automatic gear selection for 2nd spindle

Programming

Manual gear selection

If desired, the gear range (1–4) for every spindle/spindle group may be entered manually in the part program. In this case, the control unit deselects the automatic gear selection function.

If a speed outside the speed range of a gear is entered in the case of manual gear selection, the Typ3 osa will display the minimum or maximum speed for the respective gear.

M41–M44  Manual gear selection is programmed in the part program, if required.
M141–M144
M241–M244

Examples:
N... M42  Manual selection of 2nd gear for 1st spindle group
N... M141 Manual selection of 1st gear for 1st spindle
N... M244 Manual selection of 4th gear for 2nd spindle

Programming

Neutral gear

M48  Changes the gear of the spindle/spindle group to neutral.
M148
M248
5.2.3 Specifying the spindle speed

Effect

The spindle speed refers to individual spindles, or – in the presence of several spindles – to all spindles of a spindle group.

Programming

Specifying the spindle speed:

\[ S_{i_1}, \ldots, S_{i_n},\ S_{SPG_{j_1}}, \ldots, S_{SPG_{m}}, \ S \]

Explanation:

- \( S_{i_1} \): Speed specified for the \( i^{th} \) spindle.
- \( i \): Index of the \( i^{th} \) spindle (\( i=1..n \)).
- \( n \): Number of available spindles (currently: \( n_{\text{max.}}=8 \)).

- \( S_{SPG_{j_1}} \): Speed specified for the \( j^{th} \) spindle group.
- \( j \): Index of the \( j^{th} \) spindle group (\( j=1..m \)).
- \( m \): Number of available spindle groups (currently: \( m_{\text{max.}}=4 \)).

- \( S \): Abridged programming format for the
  - speed of the spindle group to which the 1st spindle is assigned by default
  - speed of the 1st spindle: \( S \equiv S_{1} \) provided that
    the 1st spindle in MACODA block 104000002 (parameter 1:=0) is not assigned to a spindle group.

Please note for spindle speed specifications:

- The speed values programmed are interpreted by default as rotations per minute.
- When G196 is active, the programmed speed is interpreted as the cutting speed in mm per minute.
- A programmed speed value can be modified with the spindle-specific override. An override setting of 100% is equivalent to the programmed speed value.
- The control unit will always reduce outputs of the speed setpoint to comply with the limits entered in MACODA. Please note that these limits depend on the selected gear.
- You can set an additional speed limit by programming G192 or G292.
- The set speed value is applied until it is overwritten by a new "S word" (acting modally).
- In test mode (general inhibit, please refer to the operating manual), there is no speed output to spindles.
- Programmed speed values of auxiliary spindles are output only in the form of auxiliary functions (e.g. 32 bit aux.)
Examples:
N.. G97
N.. G.. X.. Y.. Z.. F.. SSPG1=1000
N.. G.. X.. Y.. Z.. F.. S1500

Speed programming active.
The spindles of the 1st spindle group are to rotate at 1000 rpm.
1st spindle speed: 2000 rpm.
3rd spindle speed: 2000 rpm.
The 1st spindle or the spindle group to which the 1st spindle is assigned by default is to rotate at 1500 rpm.

CAUTION
Incorrect programming may cause machine damage!

In combination with the spindle positioning function (M19, ...), the control unit will interpret the S word not as speed but instead as the positioning angle! The meaning of the S word (spindle speed/cutting speed) is defined by G97/G196!

5.2.4 Activating spindles via machine functions

Spindle functions can be selected if one of the following conditions is fulfilled:

- The spindle was stopped with M5.
- The spindle was started by a part program, which has been stopped by a feed-hold command (spindle is reserved for this channel).
  In this case, the spindle must not be reserved by G32 (tapping without compensation chuck).
- The spindle was started via a machine function.
- The spindle was started in setting-up mode (spindle manual or spindle jog).

Spindle selection via machine functions will not result in spindle reservation.

The following spindle functions can be selected via machine functions:

- Speed S or SSPG
- Spindle functions M3/M13, M4/M14, M5 or M19
- Gear ranges M41–M44 or M48

CAUTION
If a spindle in a part program that has been stopped by a feed-hold command is switched to constant cutting speed (G196), the speed programmed for this spindle under machine functions will be interpreted as being the cutting speed.

When machine functions are deselected, the spindle functions entered in the init string (MACODA block 706000020, channel 0) are executed:

- The spindle is stopped by M5 if the spindle was activated via machine functions with M3/M13, M4/M14 or M19.
- M41–M44, M48 are skipped. No gear change is executed.
CAUTION
Because no spindle reservation is made if a spindle is selected via machine functions, the respective spindle may be activated anytime by a part program, an external NC block selection, or by jogging.

5.2.5 Activating spindles via the interface

Spindles can be activated via the interface (spindle manual or spindle jog) if one of the following conditions is fulfilled:

- The spindle was stopped with M5.
- The spindle was started by a part program, which has been stopped by a feed-hold command (spindle is reserved for this channel). In this case, the spindle must not be reserved by G32 (tapping without compensation chuck). Additionally, the spindle must not be switched to constant cutting speed (G196) in the part program.
- The spindle was started in setting-up mode (spindle manual or spindle jog).

The following spindle functions may be selected via the respective, spindle-specific interface:

- JogM3, JogM4

Spindle selection via the interface will not result in spindle reservation. No spindle-specific action is released when manual mode is deactivated.

CAUTION
Because no spindle reservation is made if a spindle is selected via the interface, the respective spindle may be activated anytime by a part program, machine function or an external NC block selection.
5.3 G functions involving spindle programming

The following G functions involve spindles:

- G32 Tapping without compensation chuck
- G95 Feedrate programming in mm/rev.
- G97 Direct speed programming
- G196 Constant cutting speed
- G192 Speed limitation, minimum speed
- G292 Speed limitation, maximum speed

For detailed descriptions, please refer to section 4, "G instructions".
5.4 Special spindle functions

5.4.1 Spindle functions in position mode

Effect
In normal operation, spindles are always operated in spindle speed mode. In special cases (e.g. “Spindle operation in positional synchronism”), spindles must be activated in position mode (position control).

The following functions are available for position mode:
- Spindle, Turn right (see sect. 5.2.1)
- Spindle, Turn left (see sect. 5.2.1)
- Spindle, Stop (see sect. 5.2.1)
- Spindle orientation (see sect. 5.4.2)
- Spindle operation in positional synchronism (see sect. 5.4.4 and 5.4.2)

5.4.2 Spindle referencing

When position mode is active, the spindle reference point must be known for the following functions:
- Spindle orientation
- Spindle operation in positional synchronism

The reference point is determined by means of the spindle positioning function (M19) while spindle speed mode is active.

Programming

Spindle referencing

M<19> Si ...Sn

Spindles are referenced with the spindle positioning function (M19 or application-specific M function) while spindle speed mode is active.

5.4.3 Switching to position-controlled spindle operation

Effect
For spindle operation with spindle position control, the spindle must be switched from speed mode to position mode.

Programming

Manual drive interface switching: SDOM
SDOM Si=0|1 ... Sn=0|1
or
SpDriveOpMode Si=0|1 ... Sn=0|1

Explanation:
- Si Switches the drive interface of the i-th spindle(s) to:
  - Si=0: speed mode, or
  - Si=1: position mode
- n Number of available spindles (currently: nmax.=8)
- i Index of the i-th spindle (i=1..n)

Function SDOM should be called only after the spindle was stopped with M5 because otherwise a spindle-stop command is triggered internally.
5.4.4 Spindle operation in positional synchronism

Spindle operation in positional synchronism is required mainly for lathes (e.g. for clamping a workpiece/tool onto 2 spindles facing each other, for transferring workpieces, etc.). These processes require several spindles running simultaneously and in positional synchronism (spindle coupling).

**Definition of spindle coupling**

Up to 4 different groups of coupled spindles can be activated simultaneously by the Typ3 osa:

- A group of coupled spindles consists of one leadscrew (master) and up to seven slave spindles.
- The slaves can be coupled with the master at any offset angle between 0° and 359.9999° (coupling distance).
- When coupling is active, every slave spindle can be turned by up to ±3600° (absolute) relative to its coupling distance.
- Slave spindles can be added or removed while coupling is active.

The following **marginal conditions** must be fulfilled for spindle coupling:

- The spindle drives must be configured as endless rotary axes (modulo axes) (see SERCOS files in sect. 5.4.12).
- All spindles of a group of coupled spindles must have a common speed range.
- All spindles of a group of coupled spindles must have similar dynamics.
- All spindles of a group of coupled spindles must be equipped with an encoder system which, when M19 (Spindle positioning) is used in speed mode, determines simultaneously the reference points of
  - the spindle (spindle speed interface)
  - the "C axes" (position interface).

**Restrictions applying to spindle operation in positional synchronism:**

- Maximum spindle speed of a group of coupled spindles depends on the NC cycle time (MACODA parameter 903000001): $S_{\text{max}}[\text{min}^{-1}] = 14,400 / \text{NC cycle time}[\text{msec}]$

Example: NC cycle time = 4 min⁻¹

$S_{\text{max}} = 14400/4 = 3600 \text{ min}^{-1}$
5.4.5 Configuring slave spindles

**Coupling distance: SCD**

**Effect**
The coupling distance defines the positional difference in setpoint values between the master spindle and its slave spindle(s) from the time coupling takes effect.

**Programming**

**Setting the coupling distance for one or several slave spindles:**

SCD  Si=<distance i>... Sn=<distance n>

or

SpCoupleDistance  Si=<distance i>... Sn=<distance n>

**Explanation:**

Si  

i\textsuperscript{th} slave spindle(s)

distance i  

positional difference in setpoint values of the i\textsuperscript{th} slave spindle(s) from the time coupling takes effect

n  

number of available spindles (currently: n\text{max.}=8)

i  

index of the i\textsuperscript{th} spindle (i=1..n)

range of values (distance):  

–359.9999° .. + 359.9999°

default: 0°

validity: spindle-specific date

**Synchronous mode window: SCSW**

**Effect**

At the start of spindle synchronization, the NC waits until the deviation of the actual position values from the setpoint values of the respective slave spindles lies within the interval defined by the window [–vale,+value]. When synchronous spindle mode is active, this window is monitored. If an error occurs, the IF signal "positional synchronism 1" on the spindle-specific output interface is reset.

**Programming**

**Defining the synchronous mode window:**

SCSW  Si=<window i>... Sn=<window n>

or

SpCoupleSyncWindow  Si=<window i>... Sn=<window n>

**Explanation:**

Si  

i\textsuperscript{th} slave spindle(s)

window i  

specification of the synchronous mode window for the i\textsuperscript{th} slave spindle(s)

n  

number of available spindles (currently: n\text{max.}=8)

i  

index of the i\textsuperscript{th} spindle (i=1..n)

range of values (window):  

0° .. 20°

default: 1°

validity: spindle-specific date
**Synchronous mode error window: SCEW**

**Effect**

When synchronous mode is active, this window is monitored. If an error occurs, the IF signal "positional synchronism 2" on the spindle-specific output interface is reset.

**Programming**

**Defining the synchronous mode error window:**

SCEW $S_i=<\text{window } i>... \text{ Sn}=<\text{window n}>$

or

SpCoupleSyncErrorWindow $S_i=<\text{window } i>... \text{ Sn}=<\text{window n}>$

**Explanation:**

- $S_i$ $i$th slave spindle(s)
- window $i$ specification of the synchronous mode error window of the $i$th slave spindle(s)
- $n$ number of available spindles (currently: $n_{\text{max.}}=8$)
- $i$ index of the $i$th spindle ($i=1..n$)

- range of values (window): $0^\circ \ldots 359.9999^\circ$
- default: $10^\circ$
- validity: spindle-specific date

---

**5.4.6 Defining groups of coupled spindles**

**Creating, modifying or disbanding groups of coupled spindles: SCC**

**Effect**

With the SCC command, you can define, modify (add or remove slave spindles) or delete groups of coupled spindles.

When groups of coupled spindles are defined or new slave spindles are added to them, they are subsequently switched automatically to position mode, if required.

Conversely, when slave spindles are removed or a group of coupled spindles is disbanded, the spindles are automatically switched back to speed mode if this mode was active prior to coupling.

Please note for the following **functions**:

- CP, couple group of coupled spindles
- group $j$ of number of the $j$th group of coupled spindles: 1 ... 4 coupled spindles
- MA, master $i$th slave spindle(s)
- $S<\text{number } i>$ number physical spindle index of the master spindle: $i=1 ... n$
- number $i .. n$ physical spindle index of the $i$th slave spindle
- $n$ number of available spindles (currently: $n_{\text{max.}}=8$)
- $i$ index of the $i$th spindle ($i=1..n$)
- $j$ $j = 1...4$
Defining a group of coupled spindles:
SCC CP=<group j of coupled spindles> MA=<number> S<number i>=1 ... S<number n>=1
or
SpCoupleConfig Couple=<group j of coupled spindles> Master=<number> S<number i>=1 ... S<number n>=1

Adding slave spindles to or removing slave spindles from a group of coupled spindles:
SCC CP=<group j of coupled spindles> S<number i>=0|1 ... S<number n>=0|1
or
SpCoupleConfig Couple=<group of coupled spindles> S<number i>=0|1 ... S<number n>=0|1

Explanation:
S<number i> i\textsuperscript{th} slave spindle(s):
S<number i>=0: remove spindle from group
S<number i>=1: add spindle to group

When modifying a group of coupled spindles, there is no need to program the number of the master spindle because the group of coupled spindles is already clearly identified by its number.

Disbanding a group of coupled spindles:
SCC CP=<group j of coupled spindles> MA=0
or
SpCoupleConfig Couple=<group j of coupled spindles> Master= 0

Waiting for synchronous mode: SCWAIT

Effect
The part program waits until the programmed group of coupled spindles is successfully created, reconfigured or disbanded. The effect of this function is equivalent to that of a conditional WAIT.

Waiting for synchronous mode:
SCWAIT CP=<group j of coupled spindles>
or
SpCoupleWaitSync Couple=<group j of coupled spindles>
5.4.7 Programming while coupling is active

Entering an angular offset while coupling is active: SCPO

Effect: When coupling is active, the stated angular offset is approached. This angle acts additively on the existing coupling offset. Thus, you can twist the master and slave spindles with reference to each other while coupling is active; the absolute offset angle (coupling distance SCD + angular offset SCPO) between master and slave spindles can be redefined anytime.

The torsion may be carried out while spindle rotation is active.

While a torsion is active, the "positional synchronism 1" signal is reset on the spindle-specific output interface.

Programming

Entering an angular offset while coupling is active:

SCPO S<number i>=<offset i> ... S<number n>=<offset n> {POSVEL<spindle speed>}
or
SpCouplePosOffset S<number i>=<offset i> ... S<number n>=<offset n>{POSVEL<spindle speed>}

Explanation:

S<number i> \( i \) th slave spindle(s)

offset i ... offset n Torsion angle of the \( i \) th slave spindle(s). The torsion angle is entered as an absolute value: \( \pm 360° \)

spindle speed Speed ratio between master and slave spindle at which the stated offset is activated. This parameter is optional and acts modally. The default value is the respective standard spindle speed specified via SERCOS ident number S-0-0222.

n Number of available spindles (currently: \( n_{\text{max}}=8 \))

i index of the \( i \) th spindle (\( i=1..n \))

Waiting for angular offset: SCPOWAIT

Effect: The part program is stopped until the angular offset programmed with SCPO is completed. The effect of this function is equivalent to a conditional WAIT.

Programming

Waiting for angular offset:

SCPOWAIT CP<group j of coupled spindles> or
SpCouplePosOffsetWait Couple=<group j of coupled spindles>

Explanation:

group j of coupled spindles number of the \( i \) th group of coupled spindles: 1 ... 4
5.4.8 Spindle coupling process

1. Creating a spindle coupling

The following conditions must be fulfilled for creating a spindle coupling:

- The coupling parameters (distance, synchronous mode window, ...) have been configured.
- The reference points of the spindles involved have been determined with M19 in speed mode.
- Position mode is activated for all spindles involved:
  Failing position mode activation for a spindle involved, the spindle concerned is stopped, switched to position mode, and restarted.

Following programming of SpCoupleConfig (SCC):

- The limits applying to the group of coupled spindles (spindle speed, acceleration) are determined and relayed to the future master spindle.
- The “coupling number” is output on the interfaces of all spindles involved (master and slave(s)).
- "Spindle is master" is output on the interface of the master spindle.

Varying with the state of motion of the spindles involved, various sequences of motions may occur in the coupling process:

<table>
<thead>
<tr>
<th>Creating a coupling</th>
<th>Sequence of motions in the coupling process</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Master and slave spindles are at rest (M5 or M19):</td>
<td>Slave spindle approaches its coupling point on the shortest possible path. Upon reaching the synchronous mode window, &quot;Positional synchronism 1&quot; and &quot;Positional synchronism 2&quot; are output on the interface.</td>
<td>The PLC must authorize a slave spindle motion. This authorization can be generated by evaluating the IF signals &quot;Coupling number&quot; + &quot;Spindle command&quot;.</td>
</tr>
<tr>
<td>Master spindle at rest (M5 or M19), slave spindle rotating (M3 or M4):</td>
<td>Slave spindle approaches its coupling point directly. Upon reaching the synchronous mode window, &quot;Positional synchronism 1&quot; and &quot;Positional synchronism 2&quot; are output on the interface.</td>
<td></td>
</tr>
<tr>
<td>Master spindle rotating (M3 or M4), slave spindle at rest (M5 or M19)</td>
<td>Slave spindle speed is accelerated to match the speed of the master spindle. When their speeds match, the slave spindle approaches its coupling point (SpCoupleDistance) on the shortest path. Upon reaching the synchronous mode window, &quot;Positional synchronism 1&quot; and &quot;Positional synchronism 2&quot; are output on the interface.</td>
<td>The PLC must authorize a slave spindle motion. This authorization can be generated by evaluating the IF signals &quot;Coupling number&quot; + &quot;Spindle command&quot;.</td>
</tr>
<tr>
<td>Master and slave spindles rotating (M3 or M4):</td>
<td>Slave spindle speed is accelerated or slowed down to match the speed of the master spindle. When their speeds match, the slave spindle approaches its coupling point on the shortest path. Upon reaching the synchronous mode window, &quot;Positional synchronism 1&quot; and &quot;Positional synchronism 2&quot; are output on the interface.</td>
<td></td>
</tr>
</tbody>
</table>
2. Coupling is active

The slave spindles follow the master spindle. If the limits of the programmed synchronous mode window and/or the synchronous mode error window are exceed, this is signaled by the NC by

- resetting the IF signal "Positional synchronism 1" (synchronous mode window),
- resetting the IF signal "Positional synchronism 2" (synchronous mode error window).

While an offset angle is activated with SpCouplePos Offset (SCPO), the NC resets the IF signal "Positional synchronism 1".

3. Uncoupling

When a slave spindle is uncoupled, it takes over the active motion functions (spindle speed and direction of rotation) of the master spindle.

On the interface, the following signals are reset:

- Coupling number,
- Spindle is master,
- Positional synchronism 1, and
- Positional synchronism 2.

If the "Spindle orientation" function is active when spindles are uncoupled, the slave spindles are switched to M5 (spindle stop).

5.4.9 Test mode with spindle coupling active

DANGER
By switching to test mode all groups of coupled spindles are stopped and uncoupled.

5.4.10 Effects of spindle-specific interface signals on spindle couplings

IF signal "Drive OFF"

The IF signal "Drive OFF" causes the NC to slow down the group of coupled spindles to a stop. The state of motion of the master spindle is set to spindle stop (M5). When the group of coupled spindles has come to a standstill, the "Drive OFF" signal of the spindle concerned is relayed to the drive.

This will produce the following effects:

- The NC blocks any further programming of the group of coupled spindles.
- The IF signals "Positional synchronism 1" and "Positional synchronism 2" of all slave spindles are reset.
- The IF signal "Coupling error" is set for the master spindle.
- The state of motion of all spindles involved is set to spindle stop (M5).
IF signal "Drive inhibit"

When the IF signal "Drive inhibit" is set, the NC must relay this signal to the drive. This will immediately break the control loop of the drive!

Therefore, the NC cannot intervene actively, instead, it can only respond:
- The NC slows down the remaining group of coupled spindles to a stop.
- The NC blocks any further programming of the group of coupled spindles.
- The IF signals "Positional synchronism 1" and "Positional synchronism 2" of all slave spindles are reset.
- The IF signal "Coupling error" is set for the master spindle.
- The state of motion of all spindles involved is set to spindle stop (M5).

Fault conditions caused by IF signals "Drive OFF" and "Drive inhibit" can be overcome only by a master spindle control reset (IF signal) or an overall control reset (PLC or operator input).

5.4.11 Effects of drive-specific messages on spindle couplings

Resetting "Drive under control":

The NC responds to this signal by
- slowing down the remaining group of coupled spindles to a stop.
- The NC blocks any further programming of the group of coupled spindles.
- The IF signals "Positional synchronism 1" and "Positional synchronism 2" of all slave spindles are reset.
- The IF signal "Coupling error" is set for the master spindle.
- The state of motion of all spindles involved is set to spindle stop (M5).

This fault condition can be overcome only by a master spindle control reset (IF signal) or an overall control reset (PLC or operator input).

Diagnostics class 1 error

A "Diagnostics class 1 error" will immediately break the control loop of the drive. The NC will respond as follows:
- Slowing down the remaining group of couple spindles to a stop.
- The NC blocks any further programming of the group of coupled spindles.
- The IF signals "Positional synchronism 1" and "Positional synchronism 2" of all slave spindles are reset.
- The IF signal "Coupling error" is set for the master spindle.
- The state of motion of all spindles involved is set to spindle stop (M5).

This fault condition (reset of "Drive under control" or diagnostics class 1 error) can be overcome only by a master spindle control reset (IF signal) or an overall control reset (PLC or operator input).
5.4.12 Notes for commissioning (spindle position mode)

MACODA parameters:
You must configure the following MACODA parameters when commissioning the system:
- MACODA 104000051, position mode: 1st acceleration in rad/s²
- MACODA 104000052, position mode: 2nd acceleration in rad/s²
- MACODA 104000053, position mode: speed for switching to 2nd acceleration in rpm

SERCOS parameters:
You must configure the following SERCOS parameters when commissioning "spindle position mode" and "C axis operation":
- S-0-0016 Drive telegram: S-0-0047 (position setpoint) must be included.
- S-0-0024 MDT telegram: S-0-0051 or S-0-0053 (position actual value 1 or 2) must be included.
- S-0-0032 Main operating mode: Spindle speed mode must be configured.
- S-0-0034 Secondary mode 2: Spindle position mode must be configured.
- S-0-0076 Type of weighting for position data: Rotary position prioritization must be configured in modulo format.
- S-x-0103 Modulo value: 360 [degrees]
- S-x-0104 Loop gain factor of position controller (KV): All spindles of a group must have the same values.

The following ident number should be relayed cyclically with the signal status word (S-0-0144) configured via S-0-0026:
- S-0-0403 Position actual values status
6 Auxiliary and special functions

In addition to path information, auxiliary and special functions are required to provide technological information.

Auxiliary functions are sent to the PLC. The transfer sequence is defined as follows:

- **bit-coded auxiliary functions**
  They are collected in the sequence in which they have been programmed and sent in packages of 13 auxiliary functions or upon receipt of the last one. An acknowledgement is compulsory only with the last package.

- **bcd-coded auxiliary functions**
  They are sent individually in the sequence in which they have been programmed. An exception are auxiliary functions that act internally (e.g. "S"). They are sent last.

- **combined programming**
  Bit-coded auxiliary functions are sent after each 13th auxiliary functions or, resp. after the last one. With the exception of auxiliary functions acting internally ("S"), bcd-coded auxiliary functions programmed previously are sent previously. Auxiliary functions acting internally are sent last.

---

**CAUTION**

The functions described below may have different effects on your machine!

The configuration options of the Typ3 osa provide an opportunity for machine tool manufacturers to implement many auxiliary and special functions that may differ from one manufacturer to the next.

Therefore, the documentation provided by the respective machine tool manufacturer always takes priority. If you are not sure whether the functions described herein actually apply to your machine, contact your system administrator!
6.1 F address (feedrate)

**Effect**

F addresses are used to determine the *feedrate of the tool* during machining.

However, Typ3 osa may interpret F addresses in different ways.

Depending on the G instruction currently active, programmed F words will act as:

- interpolation time in seconds for G1, G2, G3 and G5 (see G93, page 4–91) or
- feedrate in mm/min (see G94, page 4–92) or
- feedrate in mm/rev (see G95, page 4–94).

**DANGER**

Failure to observe the feedrate preset in the machine parameters may pose a hazard to the machine and personnel!

"Power off", "Control reset", or "Reset" will activate the F word set in machine parameters 706000020 or 706000010 (default value = F0)!

This parameter also contains the information whether G93, G94 or G95 will be active after the events mentioned above (default value = G94)!

---

**Example:** Time programming with G93

N10 G93 G1 X300 Z400 A50 B120 F60  

The programmed linear interpolation will last 60 seconds.

**Example:** Feedrate programming in mm/min with G94

N10  G1 G94 X200 Z300 F200 programed feedrate of 200 mm/min
N11  G4 F40 40 seconds dwell time
N12  X300 Z400 the 200 mm/min feedrate is active again

**Example:** Feedrate programming in mm/rev. with G95

N9  S2000 M4 spindle speed of 2000 rpm, ccw run
N10  G1 G95 X200 Z300 F0.2 programmed feedrate of 0.2 mm/rev.
N... the 0.2 mm/rev feedrate is active again
N12  X300 Z400

You may also program a dwell time using an F word in connection with G04 (cf. G04, page 4–11).
6.2 FA address (feedrate of asynchronous axes, auxiliary axes)

| Effect | Normally, asynchronous axes are traversed in rapid mode. If this behaviour is not desired under certain circumstances, you may use the FA address to influence the traversing speed of all asynchronous axes programmed in the same block. |

| CAUTION | Incorrect programming may cause damage to the machine! |
| This feedrate will only be active in the block it is programmed. |
| Furthermore, it will only be effective for the asynchronous axes programmed in the same block as the FA word! |
| Programming asynchronous axes without the FA word in a subsequent block will let the axes traverse in rapid mode again. |

| Programmierung | Example: Programming the feedrate of asynchronous axes in mm/min |
| N10 G1 G94 X200 Z300 F200 | programmed feedrate of synchronous axes of 200 mm/min |
| N11 UA400 VA140 FA250 | the asynchronous axes UA and VA are traversed to the programmed positions at 250 mm/min |
| N12 UA0 WA10 | The asynchronous axes UA and WA traverse to the programmed positions in rapid mode |

S address (spindle speed)

| Effect | See "Specifying the spindle speed", sect. 5.2.3. |
6.3 M functions

M functions (which are sometimes also referred to as M codes) consist of the address letter M and a key number. A leading "0" in the key number need not be written in the program.

Example: M function M03 (spindle ON-clockwise)

\[
\begin{align*}
\text{N ... G01 X200 Y145 Z–67.678 F250 S1000 T16 M03}
\end{align*}
\]

M functions can be used to form separate program blocks, or can be combined with other words (G, S, F, T) in one block.

The standard M functions of Typ3 osa are shown in the “Annex” under the “Overview of M functions”.

If 2 mutually exclusive internally effective M functions are programmed in the same block, the last M function programmed will be active.

This refers to M functions within the following groups:
- M03–M05, M13–M14, M19, M103–M105, M113–M114, M119
- M203–M205, M213–M214, M219
- M40, M41–M44, M140, M141–M144, M48, M148
- M240, M241–M244, M248

6.3.1 Subprogram calls

In addition to different G addresses and the P address (cf. page 2–7), subprograms can also be called by 8 non-modal M functions.

You may define the actual M functions as well as the programs called by these M functions in MACODA.

The subprogram called will be executed once.

The assignment of the M function to a program name is machine tool manufacturer specific and can be defined in machine parameters 30900003 and 30900004. Please contact your systems administrator for the M functions defined as subprogram calls for your machine.

Programmierung

As a rule, only one subprogram call by P, G or M may be included in one block.

In the event of several identical address letters in one block (e.g. G or M), the address calling the subprogram must be programmed at the end of the line.

Example: Calling a subprogram with M6

\[
\begin{align*}
\text{N500 M3 S500 M6 \quad \text{Correct!}} \\
\text{N500 M6 M3 S500 \quad \text{Wrong! (will produce runtime error)}}
\end{align*}
\]

If both a traversing motion and a subprogram call are programmed in one block, the subprogram is called after completion of the traversing motion.
6.3.2 Stop processing

### Program stop M00

**Effect**
The NC program is interrupted and the machine movements are stopped when the block has been executed.

Program execution is restarted by "NC-Start". Current statuses will not be changed.

**Programmierung**
You may program "Program stop" together with other NC functions. When all programmed functions have been executed, "Program stop" will become effective.

### Conditional program stop M01

**Effect**
The NC program will stop if the "optional stop" interface signal is additionally present.

Program execution is restarted by "NC-Start". Current statuses will not be changed.

**Programmierung**
You may program "Conditional program stop" together with other NC functions. When all programmed functions have been executed, "Conditional program stop" will become effective.

### End of main program M02, M30

**Effect**
M2 or M30 will end a program.

When ending a main program, the "end of program" interface signal will be set and "program running" will be cancelled in both cases.

The control unit then activates all statuses defined in machine parameter 706000020, "Default status", in MACODA.

Both functions can also be used in subprograms. In this case, program execution will jump back to the calling program. Modal statuses that have been changed in the subprogram will not be reset!

**Programmierung**
You may program M02 or M30 as the only instruction in a program block.

---

**CAUTION**
Undefined default statuses may cause damage to the machine!

Conditions or functions which have not been defined in machine parameter 706000020, "Default status", in MACODA will be cancelled with M02 or M30 only if functions from the same group are contained in MP 706000020.

For more information on "groups", please refer to "Instructions and special functions", sect. 2.1.1.

To see which G functions belong to which group, please refer to "Overview of G instructions" in the annex.

The M functions which mutually influence each other are described in the annex section "Overview of M functions".
Both a CONTROL RESET and M30 (end of main program) activate the functions entered for the default status (MACODA parameter 706000020). If any functions were active previously with a different function for its modal group entered in parameter 706000020, these previously active functions are automatically deselected by the default setting in the init string. Modal functions of a group from which no function is entered in the default status remain active!

Group assignment: A.2

Spindle commands M03–M219
- 1st spindle group: M03, M04, M05, M13, M14, M19
- 1st spindle: M103, M104, M113, M114, M105, M119
- 2nd spindle: M203, M204, M213, M214, M205, M219

See “Spindle functions”, sect. 5.2.1.

Speed ranges M40–M244
- 1st spindle group: M40, M41-M44
- 1st spindle: M140, M141-M144
- 2nd spindle: M240, M241-M244

See “Gear functions”, sect. 5.2.2.

Disengage speed range M48–M248
- 1st spindle group: M48
- 1st spindle: M148
- 2nd spindle: M248

See "Gear functions", sect. 5.2.2.

6.3.3 Tool change M6

The M6 function initiates a tool change.

It calls a subprogram with the name freely defined in MACODA, machine parameters 309000003 and 309000004.

For tool change, please also refer to section 6.4.
6.4 T address (tool selection)

**Effect**

With this function, you request the tool to be used in the next machining process.

The related tool number identifies the tool. It is also used for storing and calling the tool dimensions during part program execution.

Typ3 osa can output the tool number in BCD or binary format to an automatic tool changer to initiate the magazine search run.

For machine tools with manual tool change, the programmed T word is used as a job instruction for the operating personnel or for checking coincidences between the required tool and the tool available in the spindle.

The actual tool change is initiated by M06.

The appropriate signal settings and the maximum word length for programming the tool number are defined in MACODA.

Please refer to your machine tool builder’s manual for the structure and length of the tool number.

**Programmierung**

Depends on tool management system.

**Example:** Programmed tool selection

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N100</td>
<td>T123 M06 Select tool 123. Then initiate tool change with M06.</td>
</tr>
<tr>
<td>N110</td>
<td>G0 X100 Y200 Start machining with tool 123.</td>
</tr>
<tr>
<td>N120</td>
<td>G1 X150 Y230</td>
</tr>
<tr>
<td>N...</td>
<td></td>
</tr>
<tr>
<td>N500</td>
<td>T234 M06 Select tool 234. Then initiate tool change with M06.</td>
</tr>
</tbody>
</table>
Notes:
7 Tool compensation (tool correction)

The tool compensation function enables you to neglect the dimensions of a tool when programming a workpiece contour. The type3 osa automatically "compensates" the required contour and path depth (e.g. for drilling) on the basis of the length and radius of the tool.

Preconditions:
- The length and radius of the tool in use must be listed in compensation tables stored in the Typ3 osa file system,
- the appropriate compensation table must be active,
- depending on the type of compensation required (length and/or radius compensation), the H word (tool length) and/or the D word (tool radius) must be selected, and
- for radius compensation, the cutter path compensation function (G41 or G42) must be activated.

7.1 Length compensation

Effect

Length compensation becomes active upon the next path information of the spindle axis (standard: Z axis) following the call of the H word.

Positive and negative compensation values may become active.

In the following figure, for example, the length of the drills deviates from the length of the zero tool (cf. section 3.5.4). This difference can be balanced by performing a tool length compensation. For this purpose, the signs for tool length compensation must be entered as follows for a clockwise coordinate system:

Drill too short: compensation value Z –10
Drill too long: compensation value Z +10

The compensation acts modally and can only be cleared by calling a new length compensation or by programming “H0”.

By itself, the H word initiates no traversing movement.

If G90 is active, the length compensation is added to or subtracted from every subsequent programmed path value of the spindle axis, depending on the sign of the compensation value. If G91 is active, the compensation value will only be applied to the first path value.
Initiate the appropriate compensation table prior to the first axis movement of the tool and select the proper length compensation by programming the H word (tool length).

The H word may be programmed on its own or together with other information in one block.

**Example:**

N99  G22  K10
N100 G1  X... Y... Z... H... F...

Call of the compensation number (value stored in compensation table)
Application of the value to the Z axis

In order to avoid collisions when traversing over the workpiece and the part clamping device you should first traverse to an intermediate position above the workpiece, starting from the tool change point, to start length compensation.

### 7.2 Radius compensation

**Effect**

Radius compensation is activated by calling the D word in connection with G41/G42.

Select G17/G18/G19/G20 to determine the plane in which radius compensation will be active.

The D word acts modally. The old value is cleared by programming a new D word which in turn activates a new value.

---

**DANGER**

Activation of compensation values may pose a hazard to the machine and personnel!

**Under certain circumstances, the compensation may be activated while the tool is in position! The same may happen when the compensation value is deactivated. For more information, please refer to sect. 7.3.**

The radius compensation values are interpreted as the difference to a cutter radius assumed when the part program was programmed. The value will be given the unit of measure of the main axis at the plane active when compensation was activated in block preparation.

**Programming**

Activate the appropriate compensation table prior to the first axis movement of the tool and select the proper radius compensation by programming the D word (tool radius).

The D word may be programmed on its own or together with other information in one block.

**Example:**

N99  G22  K10
N100 G1  G41 X... Y... F... D...

Call of the compensation number (value stored in the compensation table)
Call the cutter path compensation to the left-hand side of the workpiece
7.3 Selection and deselection of the cutter path compensation

In many cases, it is not possible to approach the contour directly from the tool change point. In most cases, machining is started from an intermediate position (starting point; cf. examples 1 and 2: "S").

The selection of a suitable starting point helps avoid damage to the contour.

The starting point should facilitate a tangential approach to the contour and must be positioned in such a way that the direction of an axis does not change (relief cutting) at the first contour point.

7.3.1 Selection of cutter path compensation

If the control unit is in a circular mode (G2, G3, G5), no traversing movement may be programmed in the G41 or G42 block.

If no traversing movement is programmed with G41/G42, the control unit will select cutter path compensation in position – vertically to the next traversing block (cf. example 1).

If a linear traversing movement, e.g. towards the starting point of the contour, is programmed with G41/G42, the control unit will activate cutter path compensation on its way to this point (cf. example 2).

Example 1: G41 without traversing movement

Example 2: G41 with linear traversing movement for activating the compensation
7.3.2 Deselection of the cutter path compensation

As a rule, the tool does not directly move from the contour to the tool change point but rather via an intermediate position (end point).

The selection of a suitable end point helps avoid damage to the contour. Furthermore, the compensation can be deactivated on the way towards the end point.

The end point should facilitate a tangential contour departure when the radius compensation function is active.

The end point should be positioned in such a way that relief cutting does not take place due to a change in direction when the tool moves away from the contour.

If a linear traversing movement towards the end point is programmed with G40, the control unit will deactivate the cutter path compensation in a linear manner on its way there.

If no traversing movement is programmed with G40, the control unit will deselect the cutter path compensation in position – vertically to the last traversing block.

---

CAUTION
Damage to the contour possible with inside contours!

If the control unit is in a circular mode (G2, G3, G5), no traversing movement may be programmed in the G40 block.

If no traversing movement is programmed in the G40 block, the control unit will deactivate the cutter path compensation in position – vertically to the last traversing block! This may cause damage to the contour.

Therefore, before deactivating the cutter path compensation, you should move out of the inside contour (e.g. in Z direction), or program a suitable end point.
The following procedure is recommended (G41 active):

- last contour machining procedure (e.g. G2 active)
- tangential departure from contour with G1 (e.g. only X and Y programmed)
- relief cutting on Z... with G1 (e.g. program Z only)
- program G40 with X/Y movement in the extension of the last movement
- Z movement (e.g. program Z only)
- end of program

### 7.3.3 Examples

**G41 with outside contour**

Program contour with cutter path compensation left of the workpiece and deselect compensation at the end of machining.

The respective geometry compensation table and the radius compensation value are already active.
G42 with inside contour

Program contour with cutter path compensation right of the workpiece and deselect compensation at the end of machining.

The respective compensation table and the radius compensation value are already active.

```
G42 F300
N2 G42 F300
N3 G1 X115 Y50
N4 G5 X130 Y35
N5 G1 Y20
N6 X55
N7 Y30
N8 X40
N9 G5 X25 Y45
N10 G1 Y70
N11 X40
N12 G3 X70 R15
N13 G1 X100 Y80
N14 X140
N15 Y60
N16 X115
N17 Y50
N18 G40 Y35
N19 M2
```

---

G42 with inside contour

[Diagram showing programmed path (workpiece contour) and compensated path (cutter center point path).]

---

G42 with inside contour
8 Leadscrew error compensation (LSEC)

For machine axes equipped with a direct position encoder (e.g. linear scale), LSEC is not required.

Indirect measuring systems (attached to the motor or the feed screw) do not yield a feedback of the actual axis position but only information on the encoder speed (incremental encoder) or the position of the encoder within its own measuring range (absolute encoder).

However, the actual axis position depends on the precision of the mechanics which convert the motor’s rotary movement into the axis movement.

With the help of the leadscrew error compensation function, the Typ3 osa can balance mechanical inaccuracies in this range for linear axes. For this purpose, compensation values have to be determined during commissioning of a machine and saved in ASCII files in compliance with a particular scheme.

8.1 Determination of the compensation values of an axis

Determination of compensation values in three steps:

1. First divide the traversing range to be compensated into a number of sections of equal size (recommended value: approx. 10–20 mm). By this grid you determine the positions within the traversing range where compensation values have to be determined.

The maximum number of compensation values, including CCOMP (cross compensation), is 12,000.
Step size: Distance between two consecutive measuring positions.

Value range: 0.1 μm to 214748364.7 μm.

For axis positions between two consecutive measuring positions, the control unit automatically calculates a compensation value from the two adjacent compensation values by means of linear interpolation.

Starting point: Lowest coordinate value for which a compensation value is determined.

Value range: ±214748364.7 μm

If the subsequent measuring process is not started at this position, all measured compensation values will have to be sorted accordingly before they can be entered in the LSEC file.

If the starting point is not identical with the beginning of the complete traversing range, the Typ3 osa applies the compensation values of the starting point (P0 and N0) to range “A” (cf. figure).

End point: Highest coordinate value for which a compensation value is determined.

It may be in the axis position range of ±214748364.7 μm, however, it must be greater than the “starting point”.

If the end point is not identical with the end of the complete traversing range, the Typ3 osa applies the compensation values of the end point (Pn and Nn) to range “B” (cf. figure).

2. Move the axis – after referencing, if necessary – to the beginning of the axis traversing range.

Now determine the compensation value \( P_i \) (\( i \)=absolute number of measuring position: 0 to n; cf. figure) for each measuring position in positive traversing direction with the help of an external measuring device (e.g. laser). The compensation values may be in the range of ±3276.7 μm.

The following applies:

Compensation value in μm: position setpoint (NC) – actual position (machine)

All compensation values must be determined in the unit of measure “micrometer” with a maximum resolution of 0.1 μm at present.

3. After having determined the last compensation value in positive traversing direction, move the axis to the end of the axis traversing range. Now determine the compensation value \( N_i \) (\( i \)=absolute number of measuring position: n to 0; cf. figure) for each measuring position in negative traversing direction.
8.2 Programming the compensation values

Axes are numbered consecutively by the Typ3 osa system, starting with 1. All compensation values for an axis must be entered in an ASCII file named "lsecc<axis number>.tab".

Examples:

- lsecc01.tab (compensation values for axis 1)
- lsecc02.tab (compensation values for axis 2)
- ...
- lsecc08.tab (compensation values for axis 8)
- ...
- lsecc16.tab (compensation values for axis 16)

All compensation values for one axis (only for axes 1 to 8) can also be entered in the ASCII files in the old format of the filename "lsec<axis number>.tab".

Examples:

- lsec0.tab (compensation values for axis 1)
- lsec1.tab (compensation values for axis 2)
- ...
- lsec7.tab (compensation values for axis 8)

LSEC files must have the following structure:

```
<starting point> <step size> <comment>
<P0> <N0> <comment>
<P1> <N1> <comment>
<P2> <N2> <comment>
<P3> <N3> <comment>
<P4> <N4> <comment>
...
P<n> <Nn> <comment>
```

At least one blank must be programmed between the individual items in one line. Finish every line by hitting Enter.

- `<starting point>` and `<step size>` For more information, please refer to sect. 8.1.
- `<comment>` Optional string which is not evaluated by the control unit when the file is read. For example, you may enter a line number (starting with "0") and/or the axis position as a comment. The string must not exceed the current line because 2 compensation values are always expected at the beginning of a new line.
Compensation value for positive traversing direction (in μm with a maximum resolution of 0.1 μm at present). Compensation values may be in the range of ±3276.7 μm.

The following applies: Compensation value in μm

Compensation value = Position setpoint (NC) – Actual position (machine)

The following value is valid in the axis position:

<starting point> + i* <step size>

Compensation value for negative traversing direction (in μm with a maximum resolution of 0.1 μm at present). Compensation values may be in the range of ±3276.7 μm.

The following applies: Compensation value in μm

Compensation value = Position setpoint (NC) – Actual position (machine)

The following value is valid in the axis position:

<starting point> + i* <step size>

Programming example

The traversing range of the X axis (0 to 90 mm) was subdivided into a grid with a step size of 10000 (10 mm). The absolute axis position of 20000 μm (20 mm) was selected as starting point for LSEC compensation.

Seven axis positions were measured for each the positive and negative traversing direction. The compensation values of the starting point apply to axis positions below 20mm. The compensation values of the end point apply to axis positions greater than 80mm.

The X axis is defined as axis 1 in the system. Therefore, the LSEC data for this axis must be entered in file "lsecc01.tab".

Contents of file "lsecc01.tab":

20000.0  10000.0  (starting point 20mm; step size 10mm; 7 measurements)
2.0   -6.0  (position 20mm)
2.0   -4.0  (position 30mm)
4.0    2.0  (position 40mm)
6.0    2.0  (position 50mm)
2.0    6.0  (position 60mm)
-4.0    0.0  (position 70mm)
-4.0  -6.0  (position 80mm)
8.3 Activating the compensation values

The following action must be taken to activate compensation values:

- Copy or load all necessary LSEC files into the "root directory ("/"") or the "User FEPROM" ("/usrfep") subdirectory (for directory structure, refer to Typ3 osa operating instructions).

CAUTION
Danger of damage to the machine or the workpiece!
Do not initiate a control reset during processing.

- Initiate a control reset.

If no LSEC file is available for a particular axis, or if an existing file contains values that are not allowed, leadscrew error compensation is not activated for this axis.

If an LSEC file contains values that are not allowed, the control unit sends one of the following messages:

LSEC: Error when reading line <n>, file <m>!

or

LSEC: One of the numbers in line <n>, file <m> is too high!

or

LSEC: Maximum number of LSEC values in line <n>, file <m> exceeded!

Explanation:

<n> line number (0,1,2,3...)

<m> name of incorrect LSEC file:

- lsec0.tab – lsec7.tab (old names for LSEC files)
- lsecc01.tab – lsec08.tab (new names for LSEC files)
9  Cross compensation (CCOMP)

The cross compensation function is used to compensate geometric deviations of control paths or slide systems due to axis sagging. Every axis to be compensated can be influenced by a maximum of two other axes (reference axes).

**Difference between CCOMP and LSEC**

Leadscrew error compensation (LSEC) is designed to compensate mechanical inaccuracies of the drive spindle or linearity errors of a machine’s measuring system. With this function, the compensation values of an axis to be compensated are applied to that same axis depending on the machine position.

**Cross compensation function**

In addition to the errors to be compensated with LSEC, other inaccuracies may cause position errors arising from interactions between one axis and other (orthogonal) axes. In this case, the actual position of one axis influences the actual positions of other orthogonal axes, e.g. due to machine bed sagging. The cross compensation function (CCOMP) compensates these geometric deviations (statical location-related deviations) of an axis as a function of the actual position of one or two axes.

The values calculated by cross compensation (CCOMP) act in addition to the LSEC values of the axis in question.
Example:

System: X, Y, Z axes
CCOMP: cross compensation acts on X and Y axes as a function of the Z axis

Compensation values: Calculation for X and Y axes

- \( X_{\text{CORR}i} = LSEC(X_i) + \text{CCOMP}_X(Z_i) \)
- \( Y_{\text{CORR}i} = LSEC(Y_i) + \text{CCOMP}_Y(Z_i) \)

The correlations between the X axis and the Z axis (\( \text{CCOMP}_X(Z) \)) and between the Y axis and the Z axis (\( \text{CCOMP}_Y(Z) \)) are different functions.

For each axis, one or two CCOMP compensation values can be applied in addition to the LSEC compensation. Each CCOMP compensation is saved as an error matrix in a CCOMP file (i.e. max. 2 error matrix files for each axis to be compensated).

For cross compensation, the machine manufacturer will create an error matrix for the positive and negative traversing direction for each correlation.

Measurement of cross compensation values

Example: \( \text{CORR} = \text{position setpoint (NC)} - \text{actual position (machine)} \)

\[
\begin{align*}
P_0(Y_0) &= Z_0 \text{ (NC)} - Z_0 \text{ (machine)} = 0 \mu\text{m} \\
P_1(Y_1) &= Z_1 \text{ (NC)} - Z_1 \text{ (machine)} = 30 \mu\text{m} \\
P_2(Y_2) &= Z_2 \text{ (NC)} - Z_2 \text{ (machine)} = 50 \mu\text{m}
\end{align*}
\]
### 9.1 Determination of the compensation values of an axis

1. First divide the **traversing range to be compensated** into a number of sections of equal size. By this grid you determine the positions within the traversing range where compensation values have to be determined.

> Please note that a maximum of 12,000 compensation values is permitted for all axes and for a combination of LSEC and CCOMP.

---

**Step size:** Distance between two consecutive measuring positions.

**Value range:** $0.1 \, \mu m$ to $214748364.7 \, \mu m$.

For axis positions between two consecutive measuring positions, the control unit automatically calculates a compensation value from the two adjacent compensation values by means of linear interpolation.

**Starting point:** Lowest coordinate value for which a compensation value is determined.

**Value range:** $\pm 214748364.7 \, \mu m$

If the subsequent measuring process is not started at this position, all measured compensation values will have to be sorted accordingly before they can be entered in the CCOMP file.

If the starting point is not identical with the beginning of the complete traversing range, the Typ3 osa applies the compensation values of the starting point ($P_0$ and $N_0$) to range “A” (cf. figure).

For reference axes of the type “ENDLESS” or “ROTARY” (Macoda 100100004 = 2 or 3), the most suitable “starting point” is 0.0 because the position of the reference axis is calculated as modulo 360 degrees before the table is accessed.

**End point:** Highest coordinate value for which a compensation value is determined.

It may be in the axis position range of $\pm 214748364.7 \, \mu m$, however, it must be greater than the "starting point".

If the end point is not identical with the end of the complete traversing range, the Typ3 osa applies the compensation values of the end point ($P_n$ and $N_n$) to range “B” (cf. figure).
2. Move the axis – after referencing, if necessary – to the beginning of the axis traversing range.

Now determine the compensation value \( P_i \) \((i=\text{absolute number of measuring position: 0 to n; cf. figure})\) for each measuring position in positive traversing direction with the help of an external measuring device (e.g. laser). The compensation values may be in the range of \( \pm3276.7 \, \mu\text{m} \).

The following applies:
Compensation value \( P_i \) in \( \mu\text{m} \) = position setpoint (NC) – actual position (machine)

This is the compensation value for the positive traversing direction of the reference axis in the \((\text{starting point} + i\text{'step size})\).

All compensation values must be determined in the unit of measure “micrometer” with a maximum resolution of 0.1 \( \mu\text{m} \) at present.

3. After having determined the last compensation value in positive traversing direction, move the axis to the end of the axis traversing range.

Now determine the compensation value \( N_i \) \((i=\text{absolute number of measuring position: n to 0; cf. figure under 1.)}\) for each measuring position in negative traversing direction. The compensation values may be in the range of \( \pm3276.7 \, \mu\text{m} \).

The following applies:
Compensation value \( N_i \) in \( \mu\text{m} \) = position setpoint (NC) – actual position (machine)

This is the compensation value for the negative traversing direction of the reference axis in the \((\text{starting point} + i\text{'step size})\).

All compensation values must be determined in the unit of measure “micrometer” with a maximum resolution of 0.1 \( \mu\text{m} \) at present.
9.2 Programming the compensation values

The assignment of the axes (axis to be compensated, reference axes) is made by means of the file names.

The file name must be entered in the following format:

cnc##r$$tab

where

cn  (compensation)

## compensated axis ## =
axis number between 01 and 16

r$$ reference axis $$ =
axis number between 01 and 16

Axis numbers $$ and ## must be different.

As for LSEC, the CCOMP compensation values are saved to a file in a previously defined grid. All grid points must be defined.

The grid points of LSEC and CCOMP need not be identical.

A CCOMP file contains the compensation values for one axis pair.

Examples:

cnc03r01.tab  (compensation values for axis 03 as a function of axis 01)
cnc03r02.tab  (compensation values for axis 03 as a function of axis 02)

CCOMP files must have the following structure:

<starting point> <step size> <comment>
<P0> <N0> <comment>
<P1> <N1> <comment>
<P2> <N2> <comment>
<P3> <N3> <comment>
<P4> <N4> <comment>
;
<Pn> <Nn> <comment>

At least one blank must be programmed between the individual items in one line. Finish every line by pressing Enter.

<starting point> and <step size>  Refer to section 9.1 for a description.

<comment> Optional string which is not evaluated by the control unit when the file is read. For example, you may enter a line number (starting with "0") and/or the axis position as a comment.

The comment must not exceed the current line because 2 compensation values are always expected at the beginning of a new line.
Cross compensation (CCOMP)

**<Pᵢ>** Compensation value for **positive** traversing direction (in μm with a maximum resolution of 0.1 μm at present). Compensation values may be in the range of ±3276.7 μm.

The following applies: Compensation value in μm

\[
\text{Compensation value} = \text{Position setpoint (NC)} - \text{Actual position (machine)}
\]

The following value is valid in the axis position:

\[
\text{<starting point> + i * <step size>}
\]

**<Nᵢ>** Compensation value for **negative** traversing direction (in μm with a maximum resolution of 0.1 μm at present). Compensation values may be in the range of ±3276.7 μm.

The following applies: Compensation value in μm

\[
\text{Compensation value} = \text{Position setpoint (NC)} - \text{Actual position (machine)}
\]

The following value is valid in the axis position:

\[
\text{<starting point> + i * <step size>}
\]

**Programming example:**

The traversing range of the Y axis (0 to 700 mm) was subdivided into a grid with a step size of 100000 (100 mm). The absolute axis position of 20000 μm (20 mm) was selected as starting point for CCOMP compensation.

Seven axis positions were measured for each the positive and negative traversing direction. The compensation values of the starting point apply to axis positions below 20mm. The compensation values of the end point apply to axis positions greater than 620mm.

The Y axis is the reference axis and has been defined as axis 02 within the system. The Z axis is the compensated axis and has been defined as axis 03 within the system. Therefore, the CCOMP data for this axis must be entered in file "cnc03r02.TAB".

### Contents of file "cnc03r02.tab": (compensates axis 3, dependent on axis 2)

20000.0 100000.0 (starting point 20mm; step size 100mm; 7 measurements)

<table>
<thead>
<tr>
<th>Position</th>
<th>Compensation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>4.0</td>
<td>2.0</td>
</tr>
<tr>
<td>6.0</td>
<td>6.0</td>
</tr>
<tr>
<td>6.5</td>
<td>6.5</td>
</tr>
<tr>
<td>6.0</td>
<td>6.0</td>
</tr>
<tr>
<td>4.0</td>
<td>2.0</td>
</tr>
<tr>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>
9.3 Activating the compensation values

The following action must be taken to activate compensation values:

- Copy or load all necessary CCOMP files into the "root directory" ("/") or the "User FEPROM" ("/usrfep") subdirectory (for directory structure, refer to Typ3 osa operating instructions).

**CAUTION**
Danger of damage to the machine or the workpiece.
Do not initiate a control reset during processing.

- Initiate a control reset.

If no CCOMP file is available for a particular axis, or if an existing file contains values that are not allowed, cross compensation is not activated for this axis.

If a CCOMP file contains values that are not allowed, the control unit sends one of the following messages:

- CCOMP: Error when reading line <n>, file <m>
- or
- CCOMP: One of the numbers in line <n>, file <m> is too high!
- or
- CCOMP: Maximum number of CCOMP values in line <n>, file <m> exceeded!

Explanation:

- <n> line number (0,1,2,3...)
- <m> name of incorrect CCOMP file:
  - cnc01r02.tab – cnc08r07.tab

If more than two cross compensation tables are available for the very same axis to be compensated, only the first two files will be accepted.

**I** Please make sure that the number of cross compensation tables for one axis never exceeds 2.
Notes:
10 ASCII files for tables

Zero shift data, compensation data for tool compensation, and data for the "Inclined plane" function is maintained in tables stored as files in the Typ3 osa file system. The number of these tables is limited only by the available storage capacity.

Knowledge of the structure of the internal table format is required only if tables are to be created externally, i.e. not in the control unit.

CAUTION
These tables must never be edited with a text editor because this involves the risk that the table format is changed, which would render these tables unusable.

10.1 Format of zero shift tables

Effect
Zero shift tables contain the following data:
- shift values for
  - G54..G59 zero shifts,
  - 1st additive zero shift, G154..G159, and
  - 2nd additive zero shift, G254..G259,
- indication of the logical or physical axis name (per column),
- indication whether axis is linear or rotary (per column),
- unit of measurement of shift values, and
- indication whether the "Strict assignment" option is selected.

Structure
Zero shift tables are structured with
- 2 lines containing general data and
- 18 lines for shift values.

1st line
The 1st line of a zero shift table contains information on strict assignments, if any, the unit of measurement, and whether rotary or linear axes are involved. This information is encoded with 5 decimal numbers separated by blanks or tabs:

[<5000 + number of axes> <unit of measurement> <strict assignment> <0> <rotary axes>]

Explanation:
- <5000 + number of axes> Number of axes to which a shift is to be applied +5000
- <unit of measurement> unit of measurement applying to the table I.A.W MACODA parameter 902000010
- <strict assignment> 0 --> No, 1 --> Yes
- <0> fixed 0 (reserved)
- <mask with rotary axes> ASCII equivalent of mask with rotary axes
The 2nd line of a zero shift table contains the logical or physical axis names, with logical axis names taking precedence over physical axis names. Names may be entered beginning with the 3rd column up to and including the 16th column as a maximum.

<name1> <..> <namen>

Explanation:

<namen> logical or physical name of the nth axis (n= 1..16)

Table columns are assigned to the axes of a channel by the axis names entered in the table. These may be names of both logical and physical axes, with logical axis names taking precedence over physical axis names.

From the 3rd through the 20th line, all lines of the zero shift table have the same structure:

<G..><>:<value1> <..> <valuen>

Explanation:

<G..> name of G function (G54 to G259)

<:> colon

<valuen> shift value of the nth axis relating to the respective G function

Both positive and negative compensation values may be entered for the respective G functions.

### Structure of the ASCII file for zero shift tables:

<table>
<thead>
<tr>
<th>line 1</th>
<th>line 2</th>
<th>line 3 - 20</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;5000+number of axes&gt; &lt;unit of measurement&gt; &lt;strict assignment&gt; &lt;0&gt; &lt;rotary axes&gt;</td>
<td>&lt;name1&gt; &lt;name2&gt; &lt;name3&gt; ... &lt;name n&gt;</td>
<td>&lt;value n&gt;</td>
</tr>
<tr>
<td>G54 :</td>
<td>&lt;value 1&gt; &lt;value 2&gt; &lt;value 3&gt; ... &lt;value n&gt;</td>
<td>G54 : &lt;value 1&gt; &lt;value 2&gt; &lt;value 3&gt; ... &lt;value n&gt;</td>
</tr>
<tr>
<td>G55 :</td>
<td>&lt;value 1&gt; &lt;value 2&gt; &lt;value 3&gt; ... &lt;value n&gt;</td>
<td>G55 : &lt;value 1&gt; &lt;value 2&gt; &lt;value 3&gt; ... &lt;value n&gt;</td>
</tr>
<tr>
<td>G56 :</td>
<td>&lt;value 1&gt; &lt;value 2&gt; &lt;value 3&gt; ... &lt;value n&gt;</td>
<td>G56 : &lt;value 1&gt; &lt;value 2&gt; &lt;value 3&gt; ... &lt;value n&gt;</td>
</tr>
<tr>
<td>G57 :</td>
<td>&lt;value 1&gt; &lt;value 2&gt; &lt;value 3&gt; ... &lt;value n&gt;</td>
<td>G57 : &lt;value 1&gt; &lt;value 2&gt; &lt;value 3&gt; ... &lt;value n&gt;</td>
</tr>
<tr>
<td>G58 :</td>
<td>&lt;value 1&gt; &lt;value 2&gt; &lt;value 3&gt; ... &lt;value n&gt;</td>
<td>G58 : &lt;value 1&gt; &lt;value 2&gt; &lt;value 3&gt; ... &lt;value n&gt;</td>
</tr>
<tr>
<td>G59 :</td>
<td>&lt;value 1&gt; &lt;value 2&gt; &lt;value 3&gt; ... &lt;value n&gt;</td>
<td>G59 : &lt;value 1&gt; &lt;value 2&gt; &lt;value 3&gt; ... &lt;value n&gt;</td>
</tr>
<tr>
<td>G154 :</td>
<td>&lt;value 1&gt; &lt;value 2&gt; &lt;value 3&gt; ... &lt;value n&gt;</td>
<td>G154 : &lt;value 1&gt; &lt;value 2&gt; &lt;value 3&gt; ... &lt;value n&gt;</td>
</tr>
<tr>
<td>G155 :</td>
<td>&lt;value 1&gt; &lt;value 2&gt; &lt;value 3&gt; ... &lt;value n&gt;</td>
<td>G155 : &lt;value 1&gt; &lt;value 2&gt; &lt;value 3&gt; ... &lt;value n&gt;</td>
</tr>
<tr>
<td>G156 :</td>
<td>&lt;value 1&gt; &lt;value 2&gt; &lt;value 3&gt; ... &lt;value n&gt;</td>
<td>G156 : &lt;value 1&gt; &lt;value 2&gt; &lt;value 3&gt; ... &lt;value n&gt;</td>
</tr>
<tr>
<td>G157 :</td>
<td>&lt;value 1&gt; &lt;value 2&gt; &lt;value 3&gt; ... &lt;value n&gt;</td>
<td>G157 : &lt;value 1&gt; &lt;value 2&gt; &lt;value 3&gt; ... &lt;value n&gt;</td>
</tr>
<tr>
<td>G158 :</td>
<td>&lt;value 1&gt; &lt;value 2&gt; &lt;value 3&gt; ... &lt;value n&gt;</td>
<td>G158 : &lt;value 1&gt; &lt;value 2&gt; &lt;value 3&gt; ... &lt;value n&gt;</td>
</tr>
<tr>
<td>G159 :</td>
<td>&lt;value 1&gt; &lt;value 2&gt; &lt;value 3&gt; ... &lt;value n&gt;</td>
<td>G159 : &lt;value 1&gt; &lt;value 2&gt; &lt;value 3&gt; ... &lt;value n&gt;</td>
</tr>
<tr>
<td>G254 :</td>
<td>&lt;value 1&gt; &lt;value 2&gt; &lt;value 3&gt; ... &lt;value n&gt;</td>
<td>G254 : &lt;value 1&gt; &lt;value 2&gt; &lt;value 3&gt; ... &lt;value n&gt;</td>
</tr>
<tr>
<td>G255 :</td>
<td>&lt;value 1&gt; &lt;value 2&gt; &lt;value 3&gt; ... &lt;value n&gt;</td>
<td>G255 : &lt;value 1&gt; &lt;value 2&gt; &lt;value 3&gt; ... &lt;value n&gt;</td>
</tr>
<tr>
<td>G256 :</td>
<td>&lt;value 1&gt; &lt;value 2&gt; &lt;value 3&gt; ... &lt;value n&gt;</td>
<td>G256 : &lt;value 1&gt; &lt;value 2&gt; &lt;value 3&gt; ... &lt;value n&gt;</td>
</tr>
<tr>
<td>G257 :</td>
<td>&lt;value 1&gt; &lt;value 2&gt; &lt;value 3&gt; ... &lt;value n&gt;</td>
<td>G257 : &lt;value 1&gt; &lt;value 2&gt; &lt;value 3&gt; ... &lt;value n&gt;</td>
</tr>
<tr>
<td>G258 :</td>
<td>&lt;value 1&gt; &lt;value 2&gt; &lt;value 3&gt; ... &lt;value n&gt;</td>
<td>G258 : &lt;value 1&gt; &lt;value 2&gt; &lt;value 3&gt; ... &lt;value n&gt;</td>
</tr>
<tr>
<td>G259 :</td>
<td>&lt;value 1&gt; &lt;value 2&gt; &lt;value 3&gt; ... &lt;value n&gt;</td>
<td>G259 : &lt;value 1&gt; &lt;value 2&gt; &lt;value 3&gt; ... &lt;value n&gt;</td>
</tr>
</tbody>
</table>
Example: Zero shifts G54 and G259 for axes 1–5

<table>
<thead>
<tr>
<th>[5005 0 0 0 24]</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>G54</td>
<td>54.01</td>
<td>54.02</td>
<td>54.03</td>
<td>54.04</td>
<td>54.05</td>
</tr>
<tr>
<td>G55</td>
<td>55.01</td>
<td>55.02</td>
<td>55.03</td>
<td>55.04</td>
<td>55.05</td>
</tr>
<tr>
<td>G56</td>
<td>56.01</td>
<td>56.02</td>
<td>56.03</td>
<td>56.04</td>
<td>56.05</td>
</tr>
<tr>
<td>G57</td>
<td>57.01</td>
<td>57.02</td>
<td>57.03</td>
<td>57.04</td>
<td>57.05</td>
</tr>
<tr>
<td>G58</td>
<td>58.01</td>
<td>58.02</td>
<td>58.03</td>
<td>58.04</td>
<td>58.05</td>
</tr>
<tr>
<td>G59</td>
<td>59.01</td>
<td>59.02</td>
<td>59.03</td>
<td>59.04</td>
<td>59.05</td>
</tr>
<tr>
<td>G154</td>
<td>154.01</td>
<td>154.02</td>
<td>154.03</td>
<td>154.04</td>
<td>154.05</td>
</tr>
<tr>
<td>G155</td>
<td>155.01</td>
<td>155.02</td>
<td>155.03</td>
<td>155.04</td>
<td>155.05</td>
</tr>
<tr>
<td>G156</td>
<td>156.01</td>
<td>156.02</td>
<td>156.03</td>
<td>156.04</td>
<td>156.05</td>
</tr>
<tr>
<td>G157</td>
<td>157.01</td>
<td>157.02</td>
<td>157.03</td>
<td>157.04</td>
<td>157.05</td>
</tr>
<tr>
<td>G158</td>
<td>158.01</td>
<td>158.02</td>
<td>158.03</td>
<td>158.04</td>
<td>158.05</td>
</tr>
<tr>
<td>G159</td>
<td>159.01</td>
<td>159.02</td>
<td>159.03</td>
<td>159.04</td>
<td>159.05</td>
</tr>
<tr>
<td>G254</td>
<td>254.01</td>
<td>254.02</td>
<td>254.03</td>
<td>254.04</td>
<td>254.05</td>
</tr>
<tr>
<td>G255</td>
<td>255.01</td>
<td>255.02</td>
<td>255.03</td>
<td>255.04</td>
<td>255.05</td>
</tr>
<tr>
<td>G256</td>
<td>256.01</td>
<td>256.02</td>
<td>256.03</td>
<td>256.04</td>
<td>256.05</td>
</tr>
<tr>
<td>G257</td>
<td>257.01</td>
<td>257.02</td>
<td>257.03</td>
<td>257.04</td>
<td>257.05</td>
</tr>
<tr>
<td>G258</td>
<td>258.01</td>
<td>258.02</td>
<td>258.03</td>
<td>258.04</td>
<td>258.05</td>
</tr>
<tr>
<td>G259</td>
<td>259.01</td>
<td>259.02</td>
<td>259.03</td>
<td>259.04</td>
<td>259.05</td>
</tr>
</tbody>
</table>

The table must always be complete. If an axis requires no zero shift, enter the value zero (e.g.: 0.000). The given table format with a sufficient number of blanks in between the columns must be maintained. Otherwise, an error message will be output.
10.2 Format of compensation tables

Effect

Compensation tables contain the following data:
- tool radius compensation D
- tool length compensation H

Structure

Compensation tables are structured with
- 1 line containing general data
- 48 lines for radius compensation values
- 48 lines for length compensation values

1st line

The 1st line of a compensation table indicates the unit of measurement. This information is encoded with 2 decimal numbers separated by blanks or tabs:

[<5000> <unit>]

Explanation:

<5000> 5000 (fixed)
[unit of measurement] unit of measurement applying to the table
I.A.W. MACODA parameter 902000010

2nd through 49th line

The next 48 lines (lines 2 through 49) of the compensation table all have the same structure:

<D1X..>=<value>

Explanation:

<D1X..> designation of 1st through 48th radius compensation (D1X – D48X)
[value] 1st through 48th radius compensation value

50th through 97th line

The next 48 lines (line 50 through 97) of the compensation table all have the same structure:

<H1Z..>=<value>

Explanation:

<H1Z..> designation of 1st through 48th length compensation (H1Z – H48Z)
[value] 1st through 48th length compensation values relating to the respective length compensation

Both positive and negative compensation values may be entered.

Structure of the ASCII file for compensation tables

<table>
<thead>
<tr>
<th>line 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>[&lt;5000&gt; &lt;unit of measurement&gt;]</td>
</tr>
<tr>
<td>D1X  =  &lt;value&gt;</td>
</tr>
<tr>
<td>D2X  =  &lt;value&gt;</td>
</tr>
<tr>
<td>D3X  =  &lt;value&gt;</td>
</tr>
<tr>
<td>..   ..   ..</td>
</tr>
<tr>
<td>D48X =  &lt;value&gt;</td>
</tr>
<tr>
<td>H1Z  =  &lt;value&gt;</td>
</tr>
<tr>
<td>H2Z  =  &lt;value&gt;</td>
</tr>
<tr>
<td>H3Z  =  &lt;value&gt;</td>
</tr>
<tr>
<td>..   ..   ..</td>
</tr>
<tr>
<td>H48Z =  &lt;value&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>line 2 – 49</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>line 50 – 97</th>
</tr>
</thead>
</table>
Example: Compensations

<table>
<thead>
<tr>
<th>[5000 0]</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>D1X</td>
<td>&lt;0.001&gt;</td>
</tr>
<tr>
<td>D2X</td>
<td>&lt;0.002&gt;</td>
</tr>
<tr>
<td>D3X</td>
<td>&lt;0.003&gt;</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>D48X</td>
<td>&lt;0.048&gt;</td>
</tr>
<tr>
<td>H1Z</td>
<td>&lt;0.101&gt;</td>
</tr>
<tr>
<td>H2Z</td>
<td>&lt;-0.102&gt;</td>
</tr>
<tr>
<td>H3Z</td>
<td>&lt;0.003&gt;</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>H48Z</td>
<td>&lt;-0.048&gt;</td>
</tr>
</tbody>
</table>

10.3 Structure of the ASCII file for shift/orientation values of the inclined plane

There is no table editor available for editing the tables of the "Inclined plane" function. Please use a standard text editor.

Effect

The ASCII file for tables of the "Inclined plane" function contains the following data:

- zero shift compensation values for the inclined plane
- orientation compensation values for the inclined plane

Structure

The compensation table is structured with

- 6 lines for zero shift and orientation compensation values.

1st through 6th line

These 6 lines (line 1 through 6) of the compensation table all have the same structure:

\(<G..><:><value1> <value2> <value3> <value4> <value5> <value6>\)

Explanation:

\(<G..>\) designation of G function (G354 to G359)
\(<:\) colon
\(<value1>\) to \(<value3>\) zero shift values for the \(n^{th}\) axis (\(n=1..3\))
\(<value4>\) to \(<value6>\) orientation angles (Eulerian angles) for the \(n^{th}\) axis (\(n = 1..3\)):
- value 4: \(\phi\) for 1st axis
- value 5: \(\theta\) for 2nd axis
- value 6: \(\psi\) for 3rd axis

Both positive and negative compensation values may be entered. The unit of measurement applying to the table depends on the unit as set in MACODA.
Structure of the ASCII file for inclined plane tables:

<table>
<thead>
<tr>
<th>G354</th>
<th>G355</th>
<th>G356</th>
<th>G357</th>
<th>G358</th>
<th>G359</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;value1&gt;</td>
<td>&lt;value2&gt;</td>
<td>&lt;value3&gt;</td>
<td>&lt;value4&gt;</td>
<td>&lt;value5&gt;</td>
<td>&lt;value6&gt;</td>
</tr>
<tr>
<td>&lt;value1&gt;</td>
<td>&lt;value2&gt;</td>
<td>&lt;value3&gt;</td>
<td>&lt;value4&gt;</td>
<td>&lt;value5&gt;</td>
<td>&lt;value6&gt;</td>
</tr>
<tr>
<td>&lt;value1&gt;</td>
<td>&lt;value2&gt;</td>
<td>&lt;value3&gt;</td>
<td>&lt;value4&gt;</td>
<td>&lt;value5&gt;</td>
<td>&lt;value6&gt;</td>
</tr>
<tr>
<td>&lt;value1&gt;</td>
<td>&lt;value2&gt;</td>
<td>&lt;value3&gt;</td>
<td>&lt;value4&gt;</td>
<td>&lt;value5&gt;</td>
<td>&lt;value6&gt;</td>
</tr>
<tr>
<td>&lt;value1&gt;</td>
<td>&lt;value2&gt;</td>
<td>&lt;value3&gt;</td>
<td>&lt;value4&gt;</td>
<td>&lt;value5&gt;</td>
<td>&lt;value6&gt;</td>
</tr>
</tbody>
</table>

The columns must be separated from each other by blanks or tabs.

Example:

Zero shifts and orientation angles G354 and G355 for axes 1–3

G354 : 10.001 11.059 –22.050 25.000 –10.000 1.500
G355 : 100.345 0.000 30.444 25.000 10.000 0.000

A shift or orientation angle for an axis = 0 must be entered as a value (e.g.: 0.000). This only applies if the values could not be clearly assigned to the axis addresses without the value 0.

For more information, please refer to sections "Basics" (Coordinate systems) and "G instructions" (Inclined plane).
## A.1 Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C:</td>
<td>Drive name, in this case drive C (hard disk drive)</td>
</tr>
<tr>
<td>ESD</td>
<td>Electro-Static Discharge Abbreviation for all terms relating to electro-static discharge, e.g. ESD protection, ESD hazards, etc.</td>
</tr>
<tr>
<td>Fx</td>
<td>Function key with number x</td>
</tr>
<tr>
<td>GOM</td>
<td>Group Operating Mode</td>
</tr>
<tr>
<td>HP</td>
<td>Main Program (‘Hauptprogramm’)</td>
</tr>
<tr>
<td>LSEC</td>
<td>Lead Screw Error Compensation</td>
</tr>
<tr>
<td>MDI</td>
<td>&quot;Manual Data Input&quot; mode</td>
</tr>
<tr>
<td>MP</td>
<td>Machine Parameter</td>
</tr>
<tr>
<td>MSD</td>
<td>Machine Status Display</td>
</tr>
<tr>
<td>MTB</td>
<td>Machine Tool Builder</td>
</tr>
<tr>
<td>NC, CNC</td>
<td>Control Unit</td>
</tr>
<tr>
<td>OI</td>
<td>Operator Interface</td>
</tr>
<tr>
<td>OM</td>
<td>Operating Mode</td>
</tr>
<tr>
<td>PE</td>
<td>Protective Earth</td>
</tr>
<tr>
<td>PLC</td>
<td>Programmable Logic Controller</td>
</tr>
<tr>
<td>SK</td>
<td>Softkey</td>
</tr>
<tr>
<td>SP</td>
<td>Subprogram</td>
</tr>
</tbody>
</table>
# A.2 Overview of G instructions

<table>
<thead>
<tr>
<th>Instruction</th>
<th>as of version</th>
<th>Description</th>
<th>Group ¹</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>G00</td>
<td></td>
<td>Linear interpolation at rapid travel (see G200)</td>
<td>2</td>
<td>4–1</td>
</tr>
<tr>
<td>G01</td>
<td></td>
<td>Linear interpolation at feedrate (see G73)</td>
<td>2</td>
<td>4–3</td>
</tr>
<tr>
<td>G02</td>
<td></td>
<td>Circular interpolation / helical interpolation clockwise</td>
<td>2</td>
<td>4–4</td>
</tr>
<tr>
<td>G03</td>
<td></td>
<td>Circular interpolation / helical interpolation counter-clockwise</td>
<td>2</td>
<td>4–4</td>
</tr>
<tr>
<td>G04</td>
<td></td>
<td>Dwell time</td>
<td>0</td>
<td>4–11</td>
</tr>
<tr>
<td>G05</td>
<td></td>
<td>Circular interpolation / helical interpolation with tangential entry</td>
<td>2</td>
<td>4–11</td>
</tr>
<tr>
<td>G06</td>
<td></td>
<td>Acceleration programming ON (see G206)</td>
<td>11</td>
<td>4–13</td>
</tr>
<tr>
<td>G07</td>
<td></td>
<td>Acceleration programming OFF (see G206)</td>
<td>11</td>
<td>4–13</td>
</tr>
<tr>
<td>G08</td>
<td></td>
<td>Path slope ON</td>
<td>3</td>
<td>4–15</td>
</tr>
<tr>
<td>G09</td>
<td></td>
<td>Path slope OFF</td>
<td>3</td>
<td>4–15</td>
</tr>
<tr>
<td>G10</td>
<td></td>
<td>Polar-coordinate programming (like G0)</td>
<td>2</td>
<td>4–24</td>
</tr>
<tr>
<td>G11</td>
<td></td>
<td>Polar-coordinate programming (like G1)</td>
<td>2</td>
<td>4–24</td>
</tr>
<tr>
<td>G12</td>
<td></td>
<td>Polar-coordinate programming (like G2)</td>
<td>2</td>
<td>4–24</td>
</tr>
<tr>
<td>G13</td>
<td></td>
<td>Polar-coordinate programming (like G3)</td>
<td>2</td>
<td>4–24</td>
</tr>
<tr>
<td>G14</td>
<td></td>
<td>Loop gain (KV) programming ON</td>
<td>9</td>
<td>4–25</td>
</tr>
<tr>
<td>G15</td>
<td></td>
<td>Loop gain (KV) programming OFF</td>
<td>9</td>
<td>4–25</td>
</tr>
<tr>
<td>G16</td>
<td>5.1</td>
<td>No plane</td>
<td>5</td>
<td>4–26</td>
</tr>
<tr>
<td>G17</td>
<td></td>
<td>X/Y plane selection</td>
<td>5</td>
<td>4–27</td>
</tr>
<tr>
<td>G18</td>
<td></td>
<td>Z/X plane selection</td>
<td>5</td>
<td>4–27</td>
</tr>
<tr>
<td>G19</td>
<td></td>
<td>Y/Z plane selection</td>
<td>5</td>
<td>4–27</td>
</tr>
<tr>
<td>G20</td>
<td></td>
<td>Plane selection 2 out of 6 axes and Pole programming for polar-coordinate programming</td>
<td>5</td>
<td>4–29</td>
</tr>
<tr>
<td>G21</td>
<td>5.1</td>
<td>Programming of axis classifications</td>
<td>0</td>
<td>4–30</td>
</tr>
<tr>
<td>G22</td>
<td></td>
<td>Table activation</td>
<td>0</td>
<td>4–31</td>
</tr>
<tr>
<td>G23</td>
<td></td>
<td>Conditional jump</td>
<td>0</td>
<td>2–10</td>
</tr>
<tr>
<td>G24</td>
<td></td>
<td>Unconditional jump</td>
<td>0</td>
<td>2–10</td>
</tr>
<tr>
<td>G32</td>
<td></td>
<td>Tapping without compensation chuck (see G532)</td>
<td>0</td>
<td>4–35</td>
</tr>
<tr>
<td>G34</td>
<td></td>
<td>Rounding of corners (with max. admissible deviation) ON (see G36)</td>
<td>12</td>
<td>4–40</td>
</tr>
<tr>
<td>G35</td>
<td></td>
<td>Rounding of corners OFF (see G36)</td>
<td>12</td>
<td>4–40</td>
</tr>
<tr>
<td>G36</td>
<td></td>
<td>Delete max. admissible deviation programmed for G34</td>
<td>0</td>
<td>4–40</td>
</tr>
<tr>
<td>G37</td>
<td></td>
<td>Determination of mirror or rotating point</td>
<td>22</td>
<td>4–42</td>
</tr>
<tr>
<td>G38</td>
<td></td>
<td>Mirroring, scaling, rotating ON</td>
<td>22</td>
<td>4–42</td>
</tr>
<tr>
<td>G39</td>
<td></td>
<td>Mirroring, scaling, rotating OFF</td>
<td>22</td>
<td>4–42</td>
</tr>
<tr>
<td>G40</td>
<td></td>
<td>Cutter path compensation OFF</td>
<td>41</td>
<td>4–53</td>
</tr>
<tr>
<td>G41</td>
<td></td>
<td>Cutter path compensation to the left of the workpiece ON</td>
<td>41</td>
<td>4–53</td>
</tr>
<tr>
<td>G42</td>
<td></td>
<td>Cutter path compensation to the right of the workpiece ON</td>
<td>47</td>
<td>4–53</td>
</tr>
<tr>
<td>G53</td>
<td></td>
<td>Zero shift (ZS) OFF (see G153; G253)</td>
<td>17</td>
<td>4–56</td>
</tr>
<tr>
<td>G54..G59</td>
<td></td>
<td>Zero shift (ZS) ON (see G154; G254)</td>
<td>17</td>
<td>4–56</td>
</tr>
<tr>
<td>G60</td>
<td></td>
<td>Programmed contour shift ON (see G67)</td>
<td>20</td>
<td>4–58</td>
</tr>
<tr>
<td>G61</td>
<td></td>
<td>Exact positioning ON (see G163 and G164..166)</td>
<td>13</td>
<td>4–59</td>
</tr>
<tr>
<td>G62</td>
<td></td>
<td>Exact positioning OFF (see G163)</td>
<td>13</td>
<td>4–59</td>
</tr>
<tr>
<td>G63</td>
<td></td>
<td>Feedrate 100% ON (see G66)</td>
<td>7</td>
<td>4–60</td>
</tr>
<tr>
<td>G64</td>
<td></td>
<td>Feed compensation: Cutter contact point</td>
<td>42</td>
<td>4–61</td>
</tr>
<tr>
<td>G65</td>
<td></td>
<td>Feed compensation: Cutter center point</td>
<td>42</td>
<td>4–61</td>
</tr>
<tr>
<td>Instruction</td>
<td>as of version</td>
<td>Description</td>
<td>Group</td>
<td>Page</td>
</tr>
<tr>
<td>-------------</td>
<td>---------------</td>
<td>-------------</td>
<td>-------</td>
<td>------</td>
</tr>
<tr>
<td>G66</td>
<td></td>
<td>Feedrate 100% OFF (see G63)</td>
<td>7</td>
<td>4–60</td>
</tr>
<tr>
<td>G67</td>
<td></td>
<td>Programmed contour shift OFF (see G60)</td>
<td>20</td>
<td>4–58</td>
</tr>
<tr>
<td>G68</td>
<td></td>
<td>Outside angle compensation: Arc</td>
<td>43</td>
<td>4–63</td>
</tr>
<tr>
<td>G69</td>
<td></td>
<td>Outside angle compens.: Intersection of the equidistants</td>
<td>43</td>
<td>4–63</td>
</tr>
<tr>
<td>G70</td>
<td></td>
<td>Inch programming</td>
<td>8</td>
<td>4–65</td>
</tr>
<tr>
<td>G71</td>
<td></td>
<td>Metric programming</td>
<td>8</td>
<td>4–65</td>
</tr>
<tr>
<td>G73</td>
<td></td>
<td>Linear interpolation at feedrate with exact positioning (see G1)</td>
<td>2</td>
<td>4–65</td>
</tr>
<tr>
<td>G74</td>
<td></td>
<td>Approach reference point</td>
<td>0</td>
<td>4–66</td>
</tr>
<tr>
<td>G75</td>
<td></td>
<td>Probe input</td>
<td>0</td>
<td>4–68</td>
</tr>
<tr>
<td>G76</td>
<td></td>
<td>Traverse to machine-oriented absolute axis position</td>
<td>0</td>
<td>4–76</td>
</tr>
<tr>
<td>G78</td>
<td></td>
<td>Drill-axis switching ON</td>
<td>36</td>
<td>4–77</td>
</tr>
<tr>
<td>G79</td>
<td></td>
<td>Drill-axis switching OFF</td>
<td>36</td>
<td>4–77</td>
</tr>
<tr>
<td>G80</td>
<td></td>
<td>Deactivate boring cycles G81-G86 and G184</td>
<td>1</td>
<td>4–79</td>
</tr>
<tr>
<td>G81</td>
<td></td>
<td>Boring cycle: Drilling; retraction at rapid</td>
<td>1</td>
<td>4–79</td>
</tr>
<tr>
<td>G82</td>
<td></td>
<td>Boring cycle: Drilling; retraction at feedrate</td>
<td>1</td>
<td>4–79</td>
</tr>
<tr>
<td>G83</td>
<td></td>
<td>Boring cycle: Deep-hole drilling; retraction at rapid</td>
<td>1</td>
<td>4–79</td>
</tr>
<tr>
<td>G84</td>
<td></td>
<td>Boring cycle: Tapping with compensation chuck</td>
<td>1</td>
<td>4–79</td>
</tr>
<tr>
<td>G85</td>
<td></td>
<td>Boring cycle: Boring; retraction at rapid</td>
<td>1</td>
<td>4–79</td>
</tr>
<tr>
<td>G86</td>
<td></td>
<td>Boring cycle: Boring; retraction at feedrate</td>
<td>1</td>
<td>4–79</td>
</tr>
<tr>
<td>G90</td>
<td></td>
<td>Absolute data input 1 (see G189, G190)</td>
<td>4</td>
<td>4–88</td>
</tr>
<tr>
<td>G91</td>
<td></td>
<td>Incremental data input (see G191)</td>
<td>4</td>
<td>4–88</td>
</tr>
<tr>
<td>G92</td>
<td></td>
<td>Set actual value</td>
<td>0</td>
<td>4–90</td>
</tr>
<tr>
<td>G93</td>
<td></td>
<td>Time programming</td>
<td>6</td>
<td>4–91</td>
</tr>
<tr>
<td>G94</td>
<td></td>
<td>Feedrate programming in mm/min</td>
<td>6</td>
<td>4–92</td>
</tr>
<tr>
<td>G95</td>
<td></td>
<td>Feedrate programming in mm/rev</td>
<td>6</td>
<td>4–94</td>
</tr>
<tr>
<td>G97</td>
<td></td>
<td>Direct speed programming (see G196)</td>
<td>35</td>
<td>4–95</td>
</tr>
<tr>
<td>G105</td>
<td></td>
<td>Zerosetting of modulo axis (linear endless axis)</td>
<td>0</td>
<td>4–97</td>
</tr>
<tr>
<td>G108</td>
<td></td>
<td>Path slope taking the whole deceleration slope into account</td>
<td>3</td>
<td>4–16</td>
</tr>
<tr>
<td>G112</td>
<td></td>
<td>Consideration of the existing braking distance with active path slope OFF</td>
<td>38</td>
<td>4–99</td>
</tr>
<tr>
<td>G113</td>
<td></td>
<td>Consideration of the existing braking distance with active path slope ON</td>
<td>38</td>
<td>4–99</td>
</tr>
<tr>
<td>G114</td>
<td></td>
<td>Feed forward control ON</td>
<td>10</td>
<td>4–100</td>
</tr>
<tr>
<td>G115</td>
<td></td>
<td>Feed forward control OFF</td>
<td>10</td>
<td>4–100</td>
</tr>
<tr>
<td>G130</td>
<td>5.1</td>
<td>Tangential tool guidance OFF</td>
<td>45</td>
<td>4–102</td>
</tr>
<tr>
<td>G131</td>
<td>5.1</td>
<td>Tangential tool guidance ON</td>
<td>45</td>
<td>4–102</td>
</tr>
<tr>
<td>G134</td>
<td></td>
<td>Rounding of corners (with specification of rounding radius) ON</td>
<td>12</td>
<td>4–40</td>
</tr>
<tr>
<td>G138</td>
<td></td>
<td>Workpiece position compensation ON</td>
<td>23</td>
<td>4–106</td>
</tr>
<tr>
<td>G139</td>
<td></td>
<td>Workpiece position compensation OFF</td>
<td>23</td>
<td>4–106</td>
</tr>
<tr>
<td>G145..G845</td>
<td></td>
<td>External tool compensation ON (1st to 8th)</td>
<td>25</td>
<td>4–108</td>
</tr>
<tr>
<td>G146</td>
<td></td>
<td>External tool compensation OFF</td>
<td>25</td>
<td>4–108</td>
</tr>
<tr>
<td>G147..G847</td>
<td>4.4.1</td>
<td>General tool compensation ON (1st to 8th)</td>
<td>52</td>
<td>4–110</td>
</tr>
<tr>
<td>G148</td>
<td>4.4.1</td>
<td>General tool compensation OFF</td>
<td>52</td>
<td>4–110</td>
</tr>
<tr>
<td>G150</td>
<td></td>
<td>Changing the positioning type for endless axes (MP)</td>
<td>27</td>
<td>4–114</td>
</tr>
<tr>
<td>G151</td>
<td></td>
<td>Changing the positioning type for endless axes ON</td>
<td>27</td>
<td>4–114</td>
</tr>
<tr>
<td>G153</td>
<td></td>
<td>1st additive ZS OFF</td>
<td>18</td>
<td>4–56</td>
</tr>
<tr>
<td>G154..G159</td>
<td></td>
<td>1st additive ZS ON</td>
<td>18</td>
<td>4–56</td>
</tr>
<tr>
<td>Instruction</td>
<td>As of version</td>
<td>Description</td>
<td>Group</td>
<td>Page</td>
</tr>
<tr>
<td>-------------</td>
<td>--------------</td>
<td>-------------</td>
<td>-------</td>
<td>------</td>
</tr>
<tr>
<td>G160..G360</td>
<td></td>
<td>External zero shift ON (1st to 3rd) (see G167)</td>
<td>24</td>
<td>4–117</td>
</tr>
<tr>
<td>G161</td>
<td></td>
<td>Exact positioning at rapid travel ON</td>
<td>14</td>
<td>4–118</td>
</tr>
<tr>
<td>G162</td>
<td></td>
<td>Exact positioning at rapid travel OFF</td>
<td>14</td>
<td>4–118</td>
</tr>
<tr>
<td>G163</td>
<td></td>
<td>Exact positioning ON at feedrate and rapid travel (see G61/62 and G161/162)</td>
<td>13</td>
<td>4–59</td>
</tr>
<tr>
<td>G164</td>
<td></td>
<td>Exact positioning mode: v=0 and check for “positioning window”</td>
<td>15</td>
<td>4–119</td>
</tr>
<tr>
<td>G165</td>
<td></td>
<td>Exact positioning mode: v=0 and check for “positioning window rough”</td>
<td>15</td>
<td>4–119</td>
</tr>
<tr>
<td>G166</td>
<td></td>
<td>Exact positioning mode: v=0 without check for “positioning window”</td>
<td>15</td>
<td>4–119</td>
</tr>
<tr>
<td>G167</td>
<td></td>
<td>External zero shift OFF (see G160)</td>
<td>24</td>
<td>4–117</td>
</tr>
<tr>
<td>G168</td>
<td>5.1</td>
<td>Program coordinate ON</td>
<td>46</td>
<td>4–121</td>
</tr>
<tr>
<td>G169</td>
<td>5.1</td>
<td>Program coordinate OFF</td>
<td>46</td>
<td>4–121</td>
</tr>
<tr>
<td>G268</td>
<td>5.1</td>
<td>Additive program coordinate shift ON</td>
<td>47</td>
<td>4–121</td>
</tr>
<tr>
<td>G269</td>
<td>5.1</td>
<td>Additive program coordinate shift OFF</td>
<td>47</td>
<td>4–121</td>
</tr>
<tr>
<td>G175</td>
<td></td>
<td>On-the-fly measurement (initialization) (see G275)</td>
<td>0</td>
<td>4–69</td>
</tr>
<tr>
<td>G177</td>
<td></td>
<td>Torque reduction</td>
<td>0</td>
<td>4–123</td>
</tr>
<tr>
<td>G184</td>
<td></td>
<td>Boring cycle: Tapping without compensation chuck</td>
<td>1</td>
<td>4–79</td>
</tr>
<tr>
<td>G189</td>
<td></td>
<td>Absolute data input 2 (see G90, G190)</td>
<td>4</td>
<td>4–88</td>
</tr>
<tr>
<td>G190</td>
<td></td>
<td>Absolute with relative dimension (XI)</td>
<td>4</td>
<td>4–124</td>
</tr>
<tr>
<td>G191</td>
<td></td>
<td>Relative with absolute dimension (XA)</td>
<td>4</td>
<td>4–124</td>
</tr>
<tr>
<td>G192</td>
<td></td>
<td>Speed limitation minimum speed (see G292)</td>
<td>29</td>
<td>4–125</td>
</tr>
<tr>
<td>G194</td>
<td></td>
<td>Incremental speed programming with acceleration adaptation</td>
<td>0</td>
<td>4–93</td>
</tr>
<tr>
<td>G196</td>
<td></td>
<td>Constant cutting speed (see G97)</td>
<td>35</td>
<td>4–95</td>
</tr>
<tr>
<td>G200</td>
<td></td>
<td>Linear interpolation at rapid feed without decelerating to v=0 (see G0)</td>
<td>2</td>
<td>4–2</td>
</tr>
<tr>
<td>G202</td>
<td></td>
<td>Circular movement, turning right, clockwise</td>
<td>2</td>
<td>4–9</td>
</tr>
<tr>
<td>G203</td>
<td></td>
<td>Circular movement, turning left, counter-clockwise</td>
<td>2</td>
<td>4–9</td>
</tr>
<tr>
<td>G206</td>
<td></td>
<td>Storing of currently valid axis acceleration (see G6)</td>
<td>0</td>
<td>4–13</td>
</tr>
<tr>
<td>G228</td>
<td></td>
<td>Block transition without deceleration</td>
<td>0</td>
<td>4–18</td>
</tr>
<tr>
<td>G253</td>
<td></td>
<td>2nd additive ZS OFF</td>
<td>19</td>
<td>4–56</td>
</tr>
<tr>
<td>G254..G259</td>
<td></td>
<td>2nd additive ZS table</td>
<td>19</td>
<td>4–56</td>
</tr>
<tr>
<td>G275</td>
<td></td>
<td>On-the-fly measurement (see G175)</td>
<td>0</td>
<td>4–69</td>
</tr>
<tr>
<td>G292</td>
<td></td>
<td>Speed limitation maximum speed (see G192)</td>
<td>30</td>
<td>4–125</td>
</tr>
<tr>
<td>G301</td>
<td></td>
<td>Oscillating axis (see G350)</td>
<td>2</td>
<td>4–126</td>
</tr>
<tr>
<td>G328</td>
<td></td>
<td>Precision programming ON</td>
<td>2</td>
<td>4–129</td>
</tr>
<tr>
<td>G329</td>
<td></td>
<td>Precision programming OFF</td>
<td>2</td>
<td>4–129</td>
</tr>
<tr>
<td>G350</td>
<td></td>
<td>Oscillating axis (initialization) (see G301)</td>
<td>0</td>
<td>4–126</td>
</tr>
<tr>
<td>G352</td>
<td></td>
<td>Inclined plane (direct programming)</td>
<td>26</td>
<td>4–131</td>
</tr>
<tr>
<td>G353</td>
<td></td>
<td>Inclined plane OFF</td>
<td>26</td>
<td>4–131</td>
</tr>
<tr>
<td>G354..G359</td>
<td></td>
<td>Call of an inclined plane table</td>
<td>26</td>
<td>4–131</td>
</tr>
<tr>
<td>G374</td>
<td></td>
<td>Traverse to reference point</td>
<td>0</td>
<td>4–67</td>
</tr>
<tr>
<td>G408</td>
<td></td>
<td>Point-to-point movement using SHAPE</td>
<td>3</td>
<td>4–19</td>
</tr>
<tr>
<td>G500</td>
<td></td>
<td>Look-ahead for collision monitoring (3 blocks)</td>
<td>0</td>
<td>4–142</td>
</tr>
<tr>
<td>G500</td>
<td></td>
<td>Look-ahead for collision monitoring (5 blocks) programmable in connection with G41, G42 in one block</td>
<td>0</td>
<td>4–142</td>
</tr>
<tr>
<td>Instruction</td>
<td>as of version</td>
<td>Description</td>
<td>Group¹</td>
<td>Page</td>
</tr>
<tr>
<td>-------------</td>
<td>--------------</td>
<td>------------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------</td>
</tr>
<tr>
<td>G510</td>
<td>5.1</td>
<td>Integrating axes in axis groups with opt. error message</td>
<td>0</td>
<td>4–134</td>
</tr>
<tr>
<td>G511</td>
<td>5.1</td>
<td>Integrating axes in axis groups with WAIT</td>
<td>0</td>
<td>4–134</td>
</tr>
<tr>
<td>G512</td>
<td>5.1</td>
<td>Removing an axis from an axis group</td>
<td>0</td>
<td>4–134</td>
</tr>
<tr>
<td>G513</td>
<td>5.1</td>
<td>Accepting the axis configuration from MACODA</td>
<td>0</td>
<td>4–134</td>
</tr>
<tr>
<td>G515</td>
<td>5.1</td>
<td>Assigning &quot;Logical axis name&quot;</td>
<td>0</td>
<td>4–134</td>
</tr>
<tr>
<td>G516</td>
<td>5.1</td>
<td>Removing &quot;Logical axis name&quot;</td>
<td>0</td>
<td>4–134</td>
</tr>
<tr>
<td>G520</td>
<td></td>
<td>Drive-controlled interpolation</td>
<td>0</td>
<td>4–140</td>
</tr>
<tr>
<td>G521/G522</td>
<td></td>
<td>Drive-controlled interpolation</td>
<td>37</td>
<td>4–140</td>
</tr>
<tr>
<td>G523/G524</td>
<td></td>
<td>Drive-controlled interpolation</td>
<td>0</td>
<td>4–140</td>
</tr>
<tr>
<td>G532</td>
<td></td>
<td>Activation of tapping without compensation chuck for several spindles (see G32)</td>
<td>0</td>
<td>4–36</td>
</tr>
<tr>
<td>G543</td>
<td></td>
<td>Collision monitoring ON</td>
<td>44</td>
<td>4–142</td>
</tr>
<tr>
<td>G544</td>
<td></td>
<td>Collision monitoring OFF</td>
<td>44</td>
<td>4–142</td>
</tr>
<tr>
<td>G575</td>
<td></td>
<td>Switching NC blocks via high-speed signal</td>
<td>0</td>
<td>4–71</td>
</tr>
<tr>
<td>G581, G580</td>
<td>5.1</td>
<td>Axis coupling</td>
<td>48</td>
<td>4–146</td>
</tr>
<tr>
<td>G580</td>
<td>5.1</td>
<td>Disbanding a group of axes</td>
<td>48</td>
<td>4–151</td>
</tr>
<tr>
<td>G581</td>
<td>5.1</td>
<td>Forming a group of axes</td>
<td>48</td>
<td>4–148</td>
</tr>
<tr>
<td>G582</td>
<td>5.1</td>
<td>Creating a spline table file</td>
<td>0</td>
<td>4–155</td>
</tr>
<tr>
<td>G594</td>
<td></td>
<td>Suppressing axes for feedrate computing and separate programming of rotary axis feedrate</td>
<td>53</td>
<td>4–158</td>
</tr>
<tr>
<td>G595</td>
<td></td>
<td>Taking all axes of a channel into account for feedrate computing</td>
<td>53</td>
<td>4–158</td>
</tr>
<tr>
<td>G608</td>
<td>5.1</td>
<td>Axis-by-axis programmable SHAPE</td>
<td>3</td>
<td>4–22</td>
</tr>
<tr>
<td>G610</td>
<td>5.1</td>
<td>Stroke release time (default values)</td>
<td>49</td>
<td>4–163</td>
</tr>
<tr>
<td>G611</td>
<td>5.1</td>
<td>Stroke release time (inpos window reached)</td>
<td>49</td>
<td>4–163</td>
</tr>
<tr>
<td>G612</td>
<td>5.1</td>
<td>Stroke release time (interpolation end point reached)</td>
<td>49</td>
<td>4–163</td>
</tr>
<tr>
<td>G630</td>
<td>5.1</td>
<td>Tangential tool orientation OFF</td>
<td>50</td>
<td>4–166</td>
</tr>
<tr>
<td>G631</td>
<td>5.1</td>
<td>Tangential tool orientation ON</td>
<td>50</td>
<td>4–166</td>
</tr>
<tr>
<td>G660</td>
<td>5.1</td>
<td>Punching/Nibbling OFF</td>
<td>51</td>
<td>4–168</td>
</tr>
<tr>
<td>G661</td>
<td>5.1</td>
<td>Punching ON</td>
<td>51</td>
<td>4–168</td>
</tr>
<tr>
<td>G661</td>
<td>5.1</td>
<td>Nibbling ON</td>
<td>51</td>
<td>4–168</td>
</tr>
<tr>
<td>G900</td>
<td></td>
<td>Programming SERCOS ID numbers while in a part program</td>
<td>0</td>
<td>4–174</td>
</tr>
<tr>
<td>G9321</td>
<td></td>
<td>Retraction from tapped hole (switching of spindle to position mode)</td>
<td>0</td>
<td>4–37</td>
</tr>
<tr>
<td>G9322</td>
<td></td>
<td>Retraction from tapped hole</td>
<td>0</td>
<td>4–37</td>
</tr>
<tr>
<td>Instruction</td>
<td>Description</td>
<td>Group 1)</td>
<td>Page</td>
<td></td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
<td>----------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>AREADEF 5.1</td>
<td>Area definition</td>
<td>0</td>
<td>4–175</td>
<td></td>
</tr>
<tr>
<td>AREAVAILD 5.1</td>
<td>Activating/Deactivating the control area</td>
<td>0</td>
<td>4–175</td>
<td></td>
</tr>
<tr>
<td>GOTOBO</td>
<td>Jump instruction with beginning of the program as the destination</td>
<td></td>
<td>2–10</td>
<td></td>
</tr>
<tr>
<td>GOTOLO</td>
<td>Jump instruction with end of the program as the destination</td>
<td></td>
<td>2–10</td>
<td></td>
</tr>
<tr>
<td>LABEL</td>
<td>Destination labelling within a program</td>
<td></td>
<td>2–10</td>
<td></td>
</tr>
<tr>
<td>PREPNUM</td>
<td>Limitation of the maximum number of prepared blocks</td>
<td>0</td>
<td>4–180</td>
<td></td>
</tr>
<tr>
<td>TTOFF</td>
<td>Tangential tool orientation OFF</td>
<td>none</td>
<td>4–166</td>
<td></td>
</tr>
<tr>
<td>TTON</td>
<td>Tangential tool orientation ON</td>
<td>none</td>
<td>4–166</td>
<td></td>
</tr>
<tr>
<td>AC(...)</td>
<td>Local absolute data input (see G90/G91/G189)</td>
<td></td>
<td>4–88</td>
<td></td>
</tr>
<tr>
<td>IC(...)</td>
<td>Local incremental data input (see G90/G91/G189)</td>
<td></td>
<td>4–88</td>
<td></td>
</tr>
<tr>
<td>DC(..)</td>
<td>Local setting of the positioning type for endless axes ACP(..), ACN(..)</td>
<td></td>
<td>4–114</td>
<td></td>
</tr>
</tbody>
</table>

1) For explanations on “Groups”, see paragraph “Instructions and special functions”, section 2.1.1.

G instructions of group "0" are no "modal" functions. Therefore, they will not deselect each other mutually!

By selecting a new G function, the modal effect of a previously active G function with the same group number is deselected and replaced by the modal effect of the new G function.
## A.3 Overview of M functions and spindle functions

<table>
<thead>
<tr>
<th>Function</th>
<th>as of version</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>M00</td>
<td></td>
<td>Program stop; restart execution after “NC-Start”</td>
<td>6–5</td>
</tr>
<tr>
<td>M01</td>
<td></td>
<td>Conditional program stop; restart execution after “NC-Start”</td>
<td>6–5</td>
</tr>
<tr>
<td>M02</td>
<td></td>
<td>End of program; for main and subprograms</td>
<td>6–5</td>
</tr>
<tr>
<td>M03</td>
<td></td>
<td>Spindle ON, Turn right (1st spindle/spindle group)</td>
<td>5–7</td>
</tr>
<tr>
<td>M04</td>
<td></td>
<td>Spindle ON, Turn left (1st spindle/spindle group)</td>
<td>5–7</td>
</tr>
<tr>
<td>M05</td>
<td></td>
<td>Spindle, Stop (1st spindle/spindle group)</td>
<td>5–7</td>
</tr>
<tr>
<td>M06</td>
<td></td>
<td>Tool change</td>
<td>6–6</td>
</tr>
<tr>
<td>M13</td>
<td></td>
<td>Spindle, Turn right and cooling ON (1st spindle/spindle group)</td>
<td>5–7</td>
</tr>
<tr>
<td>M14</td>
<td></td>
<td>Spindle, Turn left and cooling ON (1st spindle/spindle group)</td>
<td>5–7</td>
</tr>
<tr>
<td>M19</td>
<td></td>
<td>Spindle “orientation” (1st spindle/spindle group)</td>
<td>5–7</td>
</tr>
<tr>
<td>M30</td>
<td></td>
<td>like M02</td>
<td>6–5</td>
</tr>
<tr>
<td>M40</td>
<td></td>
<td>Automatic gear selection (1st spindle/spindle group)</td>
<td>5–9</td>
</tr>
<tr>
<td>M41–M44</td>
<td></td>
<td>Manual gear selection (1st spindle/spindle group)</td>
<td>5–9</td>
</tr>
<tr>
<td>M48</td>
<td></td>
<td>Neutral gear (1st spindle/spindle group)</td>
<td>5–9</td>
</tr>
<tr>
<td>M103</td>
<td></td>
<td>Spindle ON, Turn right (like M03)</td>
<td>5–7</td>
</tr>
<tr>
<td>M104</td>
<td></td>
<td>Spindle ON, Turn left (like M04)</td>
<td>5–7</td>
</tr>
<tr>
<td>M105</td>
<td></td>
<td>Spindle, Stop (like M05)</td>
<td>5–7</td>
</tr>
<tr>
<td>M113</td>
<td></td>
<td>Spindle, Turn right and cooling ON (1st spindle/spindle group)</td>
<td>5–7</td>
</tr>
<tr>
<td>M114</td>
<td></td>
<td>Spindle, Turn left and cooling ON (1st spindle/spindle group)</td>
<td>5–7</td>
</tr>
<tr>
<td>M119</td>
<td></td>
<td>Spindle “orientation” (like M19)</td>
<td>5–7</td>
</tr>
<tr>
<td>M140</td>
<td></td>
<td>like M40</td>
<td>5–9</td>
</tr>
<tr>
<td>M141–M144</td>
<td></td>
<td>like M41–M44</td>
<td>5–9</td>
</tr>
<tr>
<td>M148</td>
<td></td>
<td>Neutral gear (1st spindle/spindle group)</td>
<td>5–9</td>
</tr>
<tr>
<td>M203</td>
<td></td>
<td>Spindle ON, Turn right (2nd spindle/spindle group)</td>
<td>5–7</td>
</tr>
<tr>
<td>M204</td>
<td></td>
<td>Spindle ON, Turn left (2nd spindle/spindle group)</td>
<td>5–7</td>
</tr>
<tr>
<td>M205</td>
<td></td>
<td>Spindle, Stop (2nd spindle/spindle group)</td>
<td>5–7</td>
</tr>
<tr>
<td>M213</td>
<td></td>
<td>Spindle, Turn right and cooling ON (2nd spindle/spindle group)</td>
<td>5–7</td>
</tr>
<tr>
<td>M214</td>
<td></td>
<td>Spindle, Turn left and cooling ON (2nd spindle/spindle group)</td>
<td>5–7</td>
</tr>
<tr>
<td>M219</td>
<td></td>
<td>Spindle “orientation” (2nd spindle/spindle group)</td>
<td>5–7</td>
</tr>
<tr>
<td>M240</td>
<td></td>
<td>Automatic gear selection (2nd spindle/spindle group)</td>
<td>5–9</td>
</tr>
<tr>
<td>M241–M244</td>
<td></td>
<td>Manual gear selection (2nd spindle/spindle group)</td>
<td>5–9</td>
</tr>
<tr>
<td>M248</td>
<td></td>
<td>Neutral gear (2nd spindle/spindle group)</td>
<td>5–9</td>
</tr>
<tr>
<td>S..</td>
<td></td>
<td>Specifying the spindle speed for individual spindles or for all spindles of a spindle group</td>
<td>5–11</td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
<td>Page</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>SADM</td>
<td>Transferring a spindle to another channel</td>
<td>5–4</td>
<td></td>
</tr>
<tr>
<td>SCC</td>
<td>Creating, modifying or disbanding groups of coupled spindles</td>
<td>5–18</td>
<td></td>
</tr>
<tr>
<td>SCD</td>
<td>Setting the coupling distance for one or several slave spindles</td>
<td>5–17</td>
<td></td>
</tr>
<tr>
<td>SCSW</td>
<td>Defining the synchronous mode window</td>
<td>5–17</td>
<td></td>
</tr>
<tr>
<td>SCEW</td>
<td>Defining the synchronous mode error window</td>
<td>5–17</td>
<td></td>
</tr>
<tr>
<td>SCPO</td>
<td>Entering an angular offset while coupling is active</td>
<td>5–20</td>
<td></td>
</tr>
<tr>
<td>SCPWAIT</td>
<td>Waiting for angular offset</td>
<td>5–20</td>
<td></td>
</tr>
<tr>
<td>SCWAIT</td>
<td>Waiting for synchronous mode</td>
<td>5–18</td>
<td></td>
</tr>
<tr>
<td>SDOM</td>
<td>Switching to position-controlled spindle operation</td>
<td>5–15</td>
<td></td>
</tr>
<tr>
<td>SPGALL(0)</td>
<td>Restoring spindle group default settings</td>
<td>5–2</td>
<td></td>
</tr>
<tr>
<td>SPG</td>
<td>Modal assignment of spindles to spindle groups</td>
<td>5–2</td>
<td></td>
</tr>
<tr>
<td>SPL</td>
<td>Local override of spindle assignments to spindle groups</td>
<td>5–2</td>
<td></td>
</tr>
<tr>
<td>SSPG..</td>
<td>Specifying the spindle speed for a spindle group</td>
<td>5–11</td>
<td></td>
</tr>
</tbody>
</table>
A.4 Index

A
Absolute data input (G90/G189), 4–88
Absolute data input AC(...), local, 4–88
Absolute with relative dimension (G190), 4–124
Acceleration programming, 4–13
Access rights of files, 2–1
Additional functions, 6–4
Additive program coordinate shift (G268, G269), 4–121
Address, 2–2
Approach reference point G74, 4–66
ASCII file
for tables, 10–1
glometry compensation file, 10–4
inclined plane compensation file, 10–5
zero shift file, 10–1
Asynchronous axes, 3–1
feedrate, 6–3
Auxiliary and special functions, 6–1
Auxiliary axes, 3–1
feedrate, 6–3
Axes, 3–1
asynchronous, 3–20
released, 4–138
synchronous, 3–20
Axes coordinate system, ACS, 3–6
Axes coupling (freely defined), Forming a group of axes, 4–149
Axes coupling (linear), Forming a group of axes, 4–148, 4–150, 4–156
Axis classification, 4–30
Programming, 4–30
Axis coupling (G581, G580), 4–146
Axis designations, 3–4
Axis group, 3–1
Axis index
logical, 3–1
physical, 3–1
Axis transfer, 4–138
(G510..G513, G515, G516), 4–134
active area, 4–175
Control area, 4–175
Axis transformation, 3–10
Axis–by–axis programmable SHAPE, (G608), 4–22

B
Block number, N instruction, 2–5
Boring cycles (G80 to G86 and G184), 4–79
Braking distance consideration, Path slope, 4–99

C
CCOMP (cross compensation), 9–1
Center–point programming, 4–6
Changing the positioning type for endless axes, 4–114
Channel, 3–1
Channel designation, Comments, 2–6
Circular interpolation, 3–20
Circular interpolation G02/G03, 4–4
Circular interpolation G05, 4–11
Collision monitoring, 4–142
Look–ahead, 4–142
Comments, 2–5
Commissioning, 5–24
Compensation tables, 10–4
Conditional jump G23, 4–34
Conditional program stop M01, 6–5
Constant cutting speed (G196), 4–95
Contour transitions G68/G69, 4–63
Control area, 4–175
Coordinate designations, 3–4
Coordinate system, program, 3–8
Coordinate systems, 3–6
Coordinate transformation, 3–10
Coordinates, position values, 2–4
Coupling (active)
Angular offset, 5–20
Programming, 5–20
Coupling characteristics, 4–147
Coupling distance, 5–17
Definition, 5–18
Coupling process, Sequence of motions, 5–21
Coupling table, 4–151
Cross compensation, 9–1
Cutter path compensation, 4–53
selection / deselection of the compensation, 7–3
Cutting speed, constant (G196), 4–95

D
D address, 7–2
Deceleration, none, with block transition, 4–18
Deviation from contour e, 4–129
Dimensions G70/G71
inch, 4–65
metric, 4–65
Direct speed programming (G97), 4–95
Documentation, 1–7
Drill axis switching (G78/G79), 4–77
Drive–controlled interpolation, (G520..G524), 4–140
Dwell time, in spindle revolutions, 4–94
Dwell time G04, 4–11

E
EMC Directive, 1–1
EMERGENCY–STOP devices, 1–5
End of main program M02/M30, 6–5
Endless axes, changing the positioning type, 4–114
local, 4–114
Endless axis programming, 4–114
ESD
Electrostatic discharge, 1–6
grounding, 1–6
workplace, 1–6
ESD–sensitive components, 1–6
European setting, 4–161
Exact positioning always ON (G163), 4–59
Exact positioning at rapid travel (G161/G162), 4–118
Exact positioning G61/G62, 4–59
Appendix A–10

Exact positioning: Mode, (G164,G165,G166), 4–119
External tool compensation, 3–17
External tool compensation (G145...G845, G146), 4–108
External zero shift (G160,260,360/G167), 4–117

F
F address, 6–2
FA address, 6–3
Feed 100% G63/G66, 4–60
Feed compensation G64/G65, 4–61
Feed forward control, 4–100
Feedrate
asynchronous axes, 6–3
synchronous axes, 6–2
Feedrate computation (Omega), for axes not contributing to feedrate computation, 4–159
Feedrate computing, Programming of rotary/endoless axis, 4–158
Feedrate programming in mm/min, 4–92
Feedrate programming in mm/rev, 4–94

G
Gear functions, 5–9
Gear ranges, 5–9
Neutral, 5–10
Gear selection
  Automatic, 5–9
  Manual, 5–10
General tool compensation, 3–18
General tool compensation (G147...G847, G148), 4–110
Grounding bracelet, 1–6
Group of axes
  Disbanding, 4–151
  Expanding, 4–150
  Forming a, 4–148
  Reducing, 4–150
Group of coupled spindles
  Adding slave spindles, 5–19
  Disbanding, 5–19
  Removing slave spindles, 5–19

H
H address, 7–1
Helical interpolation, 3–20
Helical interpolation G05, 4–4, 4–11
Helical–N–Interpolation, G202/G203, 4–9

I
Inch–metric programming
  European setting, 4–161
  Universal setting, 4–162
  US setting, 4–161
Inclined plane (G352, G353, G354...G359), 4–131
Incremental data input, 4–88
Incremental data input IC(...), local, 4–88
Incremental speed programming with acceleration adaptation, (G194), 4–93

J
Individual spindles, 5–2
  Assigning to spindle groups, 5–2
Instruction, 2–4
Intended use, 1–1
Interpolation types, 3–20
Jump destinations and jump instructions, 2–10

K
KV programming, 4–25

L
Leadscrew error compensation, 8–1
Length compensation, 3–16, 3–19, 7–1
Limitation of prepared blocks, maximum number (PREP-NUM), 4–180
LIN (G408), 4–19
Linear coupling, 4–147
Linear interpolation, 3–20
  at feedrate (G01), 4–3
  at feedrate with exact positioning (G73), 4–65
  at rapid travel (G00), 4–1
  at rapid with exact positioning (G161/G162), 4–118
Local setting of the positioning type for endless axes, 4–114
Logical axes, 3–1
Loop gain (KV) programming, 4–25
Low–Voltage Directive, 1–1
LSEC, 8–1

M
M addresses (M instructions), 6–4
M0 (Program stop), 6–5
M1 (Conditional program stop), 6–5
M2, M30 (End of main program), 6–5
Machine coordinate system, MCS, 3–6
Machine functions, Spindle, 5–12
Machine zero point, 3–6
Machining axes, 3–20
Machining time, time programming, 4–91
Maintenance of part programs, 2–1
Measuring activities, 1–5
Measuring probe (G75), 4–68
Mirroring of a contour, 4–43
Modal, 2–3
Modules sensitive to electrostatic discharge. See ESD–sensitive components
  Modulo axis, Zerosetting, 4–97

N
NC program, 2–1
Nibbling ON (G662), 4–168
No plane, 4–26
Nonmodal, 2–3

O
On–the–fly measurement, (G175,G275), 4–69
Operation in positional synchronism, Restrictions, 5–16
Oscillating axis (G301, G350), 4–126
Overtravel, 4–129
P
Parallel spindle, 5–2
Path compensation, 4–53
Path slope, 4–15
  taking the whole deceleration slope into account, 4–16
Physical and logical axes, 3–1
Plane selection G17/G18/G19, 4–27
Plane selection G20, 2 out of 6 axes, 4–29
Point–to–point movement using SHAPE, (G408), 4–19
Polar–coordinate programming, 4–24
Precision programming (G328/G329), 4–129
Preparatory functions, G instruction, G functions, 2–4
PREPNUM, maximum number of prepared blocks, 4–180
Program, running of, 2–6
Program block, NC block, 2–2
Program coordinate shift (G168/G169), 4–121
Program coordinate shift (G268, G269), additive , 4–121
Program coordinate system, PCS, 3–8
Program design, 2–5
Program end, 2–1
Program stop M0, 6–5
Program word, 2–2
Program zero point, 3–8
Programmed contour, manipulating, 3–14
Programmed contour shift (G60/G67), 4–58
Programming format, standard format, 2–13
Programming notes, Comments, 2–5
Programming of rotary/endless axis, Scaling, 4–158
Punching, Interface signals, 4–172
Punching ON (G661), 4–168
Punching/Nibbling OFF (G660), 4–168
Q
Qualified personnel, 1–2
R
Radius compensation, 3–17, 3–19, 7–2
Radius programming, 4–4
Rapid feed without decelerating to V=0, (G200), 4–2
Reference point, 3–7
Relative with absolute dimension (G191), 4–124
Release, 1–7
Replacement parts, 1–6
Reservation, Releasing a spindle, 5–6
Retraction from tapped hole, 4–37
Rotary axis programming, 4–114
Rotating of a contour, 4–48
Rounding of corners, 4–40
S
S address, 5–11
SADM, Transferring a spindle to another channel, 5–6
Safety instructions, 1–4
Safety markings, 1–3
Scaling of a contour, 4–45
SERCOS ID numbers, programming while in a part program, 4–174
Set actual value, 4–90
Shape (axis–by–axis), (G608), 4–22
Shape (Interpolation), (G408), 4–19
SIN (G408), 4–19
Skip block, 2–6
Slave spindles, 5–17
  Configuration , 5–17
Slope, 4–15
Spare parts, 1–6
Special function, 2–4
Special functions, 6–1
  Auxiliary and special functions, 2–4
Speed, 5–11
Speed limitation, (G192, G292), 4–125
Speed mode, 5–7
Speed programming, direct (G97), 4–95
Spindle coupling, 5–16, 5–22
  Effects of drive–specific messages, 5–23
  Process, 5–21
Spindle functions, 5–7
  In position mode, 5–15
Spindle groups, 5–2
Spindle mode, Commissioning, 5–24
Spindle orientation, Programmable, 5–8
Spindle programming, G functions, 5–14
Spindle speed, 5–11
Spindles, 3–1, 5–1
  Activating via the interface, 5–13
  Machine functions, 5–12
  Operation in positional synchronism, 5–16
  Position–controlled, 5–15
  Referencing, 5–15
  Reserving for specific channels, 5–4
  Stop, 5–8
  Switching, 5–15
  Turn left, 5–8
  Turn left with cooling ON, 5–8
  Turn right, 5–7
  Turn right with cooling ON, 5–8
Spline table, 4–155
  Creating, 4–155
  Spline table file, 4–155
Standard functions, Spindles, 5–1
Straight–line interpolation, 3–20
Stroke release, 4–172
Stroke release time
  Default values, 4–163
  Inpos window, 4–163
  Interpolation end point, 4–163
Subprogram calls
  by M function, 6–4
  with P address, 2–7
Switching NC blocks via high–speed signal, G575, 4–71
Switching the drill axis (G78/G79), 4–77
Synchronous axes, 3–1
Synchronous mode, Waiting for, 5–19
Synchronous mode error window, 5–18
Synchronous mode window, 5–17
T
T address, 6–7
Table activation, 4–31
Tangential tool guidance OFF (G130), 4–102
Tangential tool guidance ON (G131), 4–102
Tangential tool orientation OFF (G630), 4–166
Tangential tool orientation OFF (TTOFF), 4–166
Tangential tool orientation ON (G631), 4–166
Tangential tool orientation ON (TTON), 4–166
Tapped hole, retraction from, 4–37
Tapping of threads (G32), 4–35
Tapping of threads (G532), 4–36
Tapping without compensation chuck (G32), 4–35
  automatic retraction, 4–37
  manual retraction, 4–37
  Retraction, 4–37
Tapping without compensation chuck (G532), 4–36
Thread pitch, 4–35
Time programming, 4–91
Tool change M06, 6–6
Tool clamping point, 3–9
Tool compensation, functions, 3–15
Tool compensation (G145…G845, G146), external, 4–108
Tool compensation (G147…G847, G148), general, 4–110
Tool compensations, 3–15
Tool coordinate system, TCS, 3–9
Tool movements, 3–20
Tool reference points, 3–9
Tool selection, 6–7
Tool zero point, 3–9
Tools, 3–15
Torque reduction (G177), 4–123
Trademarks, 1–8
Transferring a spindle to another channel, SADM, 5–6
Traverse to reference point (really) G374, in the part program, 4–67
Traversing to a machine–oriented absolute axis position, G76, 4–76

U
Unconditional jump G24, 4–34
Universal setting, 4–162
US setting, 4–161

W
Workpiece coordinate system, WCS, 3–8
Workpiece position compensation, 4–106
Workpiece zero point, 3–8

Z
Zero shift, 3–10, 3–12
Zero shift (G160,260,360/G167), external, 4–117
Zero shift (ZS), 4–56
Zero setting of modulo axis, 4–97
Bosch Automation Technology

Australia
Robert Bosch (Australia) Pty. Ltd.
Head Office
Cnr. Centre - McNaughton Roads
P.O. Box 66
AUS-3168 Clayton, Victoria
Fax (03) 95 41 77 03

Great Britain
Robert Bosch Limited
Automation Technology Division
Meridian South Meridian Business Park
GB-LE3 2WY Braunstone
Leicestershire
Fax (01 16) 28-9 28 78

Canada
Robert Bosch Corporation
Automation Technology Division
6811 Century Avenue
CAN-Mississauga, Ontario L5N 1R1
Fax (905) 5 42-42 81

USA
Robert Bosch Corporation
Automation Technology Division
Fluid Power Products
7505 Durand Avenue
USA-Racine, Wisconsin 53406
Fax (414) 5 54-81 03

Robert Bosch Corporation
Automation Technology Division
Factory Automation Products
816 East Third Street
USA-Buchanan, MI 49107
Fax (616) 6 95-53 63

Robert Bosch Corporation
Automation Technology Division
Industrial Electronic Products
40 Darling Drive
USA-Avon, CT 0 60 01-42 17
Fax (860) 4 09-70 80

We reserve the right to make technical alterations

Your concessionary

Robert Bosch GmbH
Geschäftsbereich
Automationstechnik
Antriebs- und Steuerungstechnik
Postfach 11 62
D-64701 Erbach
Fax +49 (0) 60 62 78-4 28