WinMax Mill
NC Programming

Dual Screen and Max Consoles
for Hurco Machining Centers

HURCO

Hurco Companies, Inc.
One Technology Way
P.O. Box 68180
Indianapolis, IN 46268-0180
Tel (317) 293-5309 (products)
(317) 298-2635 (service)
Fax (317) 328-2812 (service)

For Hurco subsidiary contact information, go to Hurco's Web site:
www.hurco.com

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

USING THIS MANUAL

This documentation uses several conventions to explain the safety features and emphasize key concepts. These conventions are described in this section.

Additional information is available on the machine's Documentation CD.

Sample Screens

Sample screens in this documentation were taken from a WinMax Mill single-screen control. All screens are subject to change. The screens on your system may vary slightly. The sample screen here illustrates softkeys and includes a software version.

Softkeys

Softkeys are located on the side of the screen. You can set the softkeys to appear on either the right or left side of the screen. Refer to the *Getting Started with WinMax Mill* for information about making this selection. Softkeys may change upon field entries or other softkey selection. References to softkeys in the documentation appear with the softkey's corresponding F-key. For example, the Part Setup softkey from the Input screen above is referenced as the PART SETUP *F1* softkey.

Screen Areas

The screens are divided into the following areas, in addition to the row of softkeys:

Data Entry

The data entry area is located on the opposite side of the screen from the softkeys.

Available softkeys may change even when the text and data entry area does not.

Fields in the data entry area display or receive information. Refer to *Using the Touchscreen,* on page xv for information on entering information in fields.

Prompts and Error/Status Area

The bottom portion of the screen is reserved for prompts, program status and error messages.

Prompts provide help on data entry selections based on the field with the blinking cursor.

Errors and status messages occur anytime the status or error occurs. They are not based on the field with the blinking cursor. These messages provide machine information to the operator.

Error messages may also stop and/or prevent machine operation until the cause of the error is corrected.

Status Bar

The status bar contains

- The name of the open, selected program.
- A calculator icon—select the icon to display a working, on-screen calculator.
- Units of measure (Inch or Millimeters)—select the units of measure in the status bar to toggle between Inch and Metric.
- Programming mode (R for Radius; D for Diameter)—select the programming mode in the status bar to toggle between Radius and Diameter.
- A yellow icon—indicates the feed hold is on when visible.
- A red icon—indicates the Emergency Stop button has been pressed when visible.
- A keyboard icon—select the icon to display a working on-screen keyboard.
- The current time.

When viewed on a single-screen console, all icons appear in the same status bar; when viewed on a dual-screen console, the program name and calculator icon appear on the left screen status bar, and the unit of measure, keyboard icon and time appear on the right screen status bar.

Console Buttons and Keys

References to console buttons and keys appear in bold text throughout the documentation. For example, the Start Cycle button appears as the **Start Cycle** button and the Manual key appears as the **Manual** console key in text.

Refer to the *Getting Started with Your WinMax Mill* for information about console buttons and keys, in addition to other information about using softkeys and the pop-up text entry window.

Using the Touchscreen

The console has a touchscreen for entering programming data. To make a selection, tap the screen on a softkey, field, or drop-down list using the stylus attached to the side of the console or another suitable pointing device.

Printing

To print part or all of this manual from the CD, select **File / Print**. Be sure to review the **Print Range** selections and make the appropriate choice for pages. Select **Properties / Paper / Quality** and adjust the **Tray Selection / Paper Source** if necessary.

Printing to a Post Script printer provides the best results.

Icons

This manual may contain the following icons:

Caution/Warning

The operator may be injured and the machine severely damaged if the described procedure is not followed.

Hints and Tricks

Useful suggestions that show creative uses of the WinMax features.

Important

Ensures proper operation of the machine and control.

Troubleshooting

?

Steps that can be taken to solve potential problems.

Where can we go from here?

Lists several possible options the operator can take.

Table of Contents

To assist with onscreen viewing, this icon is located on the cover page. Click the icon to access the Table of Contents (TOC).

You can also access many of the same TOC entries from the Adobe Reader bookmarks located on the left side of the PDF page.

USING AND PRINTING THE HELP

Hurco provides documentation for using WinMax software on a control or desktop in two formats: on-screen Help and PDF. The information contained in both formats is identical.

On-screen Help contains information about the current screen. If Help is not available for a screen, a Welcome screen appears with access to the Table of Contents, Index, or Search functions.

- To view the on-screen Help directly on a Hurco control, select either the Help console button or the F console key followed by the 1 key (F1).
- To view the on-screen Help on the desktop software, select either the Help icon in the menu bar or the F1 key on your keyboard.

PDF files are available on the hard drive. These files can be copied from the hard drive to a USB memory device and transferred to a PC for viewing and printing.

Using the On-screen Help

On-screen Help provides information about using WinMax. The Help is context-sensitive to the screen level. Press the console Help button to display the Help topic for the current screen. The following list describes Help functions:

- Buttons in the upper left-hand corner of the Help screen are used to move through Help topics and print screens.
  - Use the **Hide** button to hide the navigation pane.
  - Use the **Back** button to return to the previous Help screen.
  - Use the **Print** button to print the current dispalyed Help topic, if a printer is attached and configured. See *Printing the Help* for more information about printing.
- Use the arrow buttons to move between pages within a Help topic and to move through topics.
- Use the **Contents** tab for a list of information sorted by subject:
  1. Select the “+” to expand the topic and view sub-topics.
  2. Select the topic to display it.
- Use the **Index** tab to show the Help index:
  1. Quickly scroll to an index topic by typing the topic in the box at the top of the index.
  2. Select a topic and the Display button to view the topic.
- Use the **Search** tab to search the Help for a word or phrase:
  1. Type the search word(s) into the text box at the top of the pane.

2. Select the List Topics button. A list of topics that contain the search word(s) is displayed.
3. Select a topic and the Display button to view that topic.

- Use the **Favorites** tab to save Help topics for quick access:
  1. Select the Add button at the bottom of the pane to add the current topic.
  2. Select a topic from the Favorites list, and select the Display button to view it.
     - Select a topic from the Favorites list, and select the Remove button to remove it from the list.

Printing the Help

The WinMax On-screen Help is also provided in PDF format for easy printing. The information contained in the PDF files is identical to the on-screen Help. The PDF files may be copied to a floppy disk or USB memory device to be transferred to a PC for printing. Here are the steps to access the PDF files:

1. From the Input screen, select the PROGRAM MANAGER *F8* softkey.
2. Select the DISK OPERATIONS *F7* softkey.
3. In the left-hand pane, navigate through the folders:
   - For WinMax Mill on a machine, the path is D:\Hurco\Winmax Mill\hlp.
   - For WinMax Desktop on a PC, the path is C:\Program Files\Winmax Mill\hlp.

   The PDF files will appear in the right-hand pane.

   ⇨ The SHOW ALL FILE TYPES field in User Interface Settings must be set to YES (default is NO) in order to see the PDF files in the directory. Access the SHOW ALL FILE TYPES field in Auxiliary Mode, Utilities/ User Preferences/ User Interface Settings.
4. Highlight the PDF file(s) in the right-hand pane, and select the COPY *F2* softkey.
5. Ensure that your media is loaded (either a floppy disk in the disk drive or a USB memory device in the USB port), and navigate to the proper location in the left-hand pane of the DISK OPERATIONS screen (either the floppy drive A: or the USB port E:). Highlight the desired location.
6. Place the cursor in the right-hand pane and select the PASTE *F3* softkey to paste the PDF file(s) to the desired location.

You may now remove your media and load the PDF file(s) onto a PC for printing.

OVERVIEW

This documentation describes the use of NC (Numerical Control) Part Programming, which includes the BNC (Basic Numerical Control) and the ISNC (Industry Standard Numerical Control) Editor portion of the CNC software as it is used on the machine tool console. This section explains the following:

NC Part Programming Principles

NC part programming adheres to the ANSI/EIA *RS–274–D* standard terminology with extensions for BNC and ISNC dialects. In addition, the NC programming facilities were designed to use as much of the WinMax Conversational system as possible. As a result, most of the screens are the same in both the NC and the conversational systems. This allows a smooth transition between the two.

The primary difference between conversational and NC programming is the program editors. The NC is programming uses standard G and M codes; whereas, conversational programming uses plain English or another supported programming language.

⇨ The CNC software can read NC files from the serial port directly into dynamic memory or run NC files that are partially loaded into dynamic memory. NC files can be serially loaded to the hard disk.

NC part programs can be created using the CNC on the machine tool or off-line CNC programming software running on a personal computer. NC programs cannot be converted to conversational programs, nor can NC programs be converted automatically to any other NC format.

NC Part Program Components

NC programs are a series of characters and words that form program blocks. These program blocks tell the machine tool how and where to move. The operator needs to understand the basic program structure and the types of codes in order to create, edit, and run a program successfully. These components make up NC code:

Program Start

All NC programs begin with a "%" (percent) character. When a *percent character* is received, the control starts to accept, check, and load blocks into its memory. If you are creating a new part program at the control, the percent character is automatically inserted at the beginning of the program.

Sequence Number

A sequence number serves as a block label; it has no other significance within the part program except being required with GOTOs in the NCPP option and the M99 jump command. Sequence numbers are often used to mark the beginning of milling sequences so you can restart at a given sequence number or recall specific operations within the program.

When programming on an off-line system, sequence numbers should be used sparingly. Sequence numbers (N words) are optional in the NC Editor, and they are useful in programs sent over the RS-232 link. However, the absence of sequence numbers permits faster processing (loading, syntax checking, and parsing) of the part program and can result in improved part program execution. In addition, omission of these numbers increases the amount of the program that can fit into memory.

⇨ If you request renumbering of part program sequence numbers, any sequence numbers in GOTO statements will not be updated. You must then press the (F1) Yes softkey before re-sequencing will take place. To cancel the renumbering, press the (F8) No softkey. In general, you will not want to renumber part programs that use GOTO statements.

Address Characters

An address character is the first character of a word in a program block. The Ignore Command signals the system to ignore the remainder of the block. The Comment Command characters are used to delimit comments. The following is a list of the address characters recognized by this system:

/	Ignore Command
( )	Comment Command
:	Subprogram Number (NCPP Option)
A	Rotary Dimension Around X-axis
B	Rotary Dimension Around Y-axis
D	Tool Diameter Offset
F	Feedrate
G	Preparatory Functions
H	Index into the tool length offset table
I	X-axis Arc Center/Offset, X scale factor, Canned Cycle Bore Shift
J	Y-axis Arc Center/Offset, Y scale factor, Canned Cycle Bore Shift
K	Z-axis Arc Center/Offset, Z scale factor, Canned Cycle Repeat
L	Tool Length Offset, Data Set Mode
M	Miscellaneous Functions
N	Sequence Number
O	Subprogram Number (NCPP option)
P	Subprogram Number, Dwell Time, Scaling Factor
Q	Canned Cycle Bore Shift, Peck Depth
R	Rotation Angle, Return Level, Circular Interpolation Radius
S	Spindle Speed Function
T	Tool Select
X	Primary X Motion Dimension, Dwell Time
Y	Primary Y Motion Dimension
Z	Primary Z Motion Dimension

Special Characters

Special characters are ASCII characters within a file which have special meaning to the system and cannot be edited. The following special characters are recognized by the NC software:

- **%**—**Beginning/End of tape**—signals the system that all of the following characters are part of the program. The system automatically adds this character to the beginning of a new program. You can also include the % character to signal the End of Tape.
- **E. of tape**— **(EOT)** (optional for BNC and ISNC)—signals the NC system that no more legal program characters follow. This character is optional to provide compatibility with existing programs that include EOT characters at the end.
- **[CR]**—**Carriage Return**—signals the End of a Program Block.
- **[CRLF]**—**Carriage Return/Line Feed Pair**—signals the End of a Program Block (identical to [CR]).

⇨ [CRLF] is not shown when the program is viewed in the NC Editor.

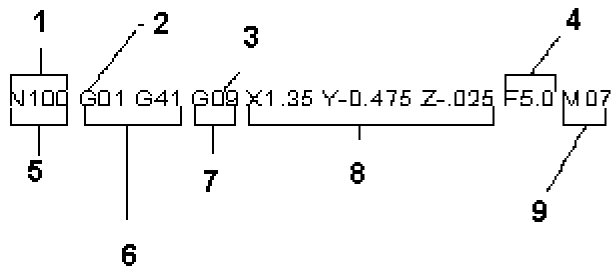
Words

A word is a group of alphanumeric characters. The first character is an address character—a letter such as M or G. The address character is followed by a signed or unsigned numeric value. Some sample NC words are “X-.03” and “G00.” One word or groups of words form a program block.

Block

A block is a group of words terminated by the end-of-block character: a carriage return [CR] or a carriage return/line feed pair [CRLF]. Each block within a part program must be terminated with either a [CR] or a [CRLF].

The following illustration shows a typical NC block and its components:



1	Word
2	Address character
3	Numeric character
4	Feedrate word
5	Sequence number
6	Modal preparatory functions
7	One-shot preparatory function
8	Dimension words
9	Miscellaneous function

Figure 1–1. Typical NC Block

Default M and G Codes

Upon power up, control reset, initial entry into the NC Editor, or after erasing a program, the system presets these M codes as defaults:

M05 Spindle Off

M09 Both Coolant Systems Off

The system also presets certain G codes as the default active codes. The default G codes are highlighted in the G Code Table in the “Preparatory Functions-G Codes” section.

The system uses the units specified when the NC Editor is selected, not the G codes, for graphics display and running the part program.

Navigation

To move the cursor from a block to the beginning of the next block, press the down arrow (↓). Use the right/advance arrow (→) and the left/back arrow (←) to move the cursor within a block. Use the Enter key to move the cursor between words and blocks.

To move to the beginning of the current block, press the *Home* key or the up arrow (↑). If the cursor is already at the beginning of the block, pressing the up arrow moves the cursor to the beginning of the last word in the previous block.

To move from a word to the beginning of the next word, press the Enter key. If the cursor is at the end of the current block when the Enter key is pressed, the editor automatically presents the next legal address character.

To move from one character to the next, press the right arrow. If the cursor is at the end of the current block, the cursor wraps around to the beginning of the block.

To move from one character to the preceding character, press the left arrow. If the cursor is at the start of the current block, it wraps around to the end of the current block.

Delete characters or words from a block using these methods:

- To delete numeric data, position the cursor on the number and press the Delete key.
- To erase the entire word, position the cursor on the address character and then press the left arrow or the Delete key. The entire word is removed since numeric data is not allowed in an NC program without an address character to introduce it.

NC Editor

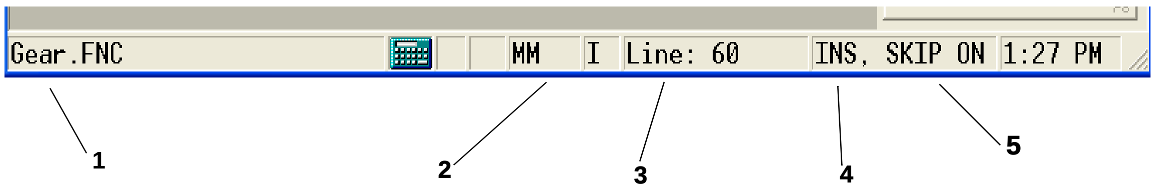
NC Editor provides a wide range of tools to review, create, and modify NC part programs. The editor has the following features:

- Several ways of selecting the code, such as by dragging the stylus across the screen, and by using keyboard keys (if available).
- Copying/Pasting/Cutting of a block of selected code, including to applications outside WinMax.
- Undo/Redo functionality.
- Unlimited number of tags.
- Real-time syntax check. Incorrect syntax is indicated by showing the incorrect text in red. Comments are shown in green:

```
N65[#1]]
N70X#100 Y#101 (Rapid to first tooth corner of tooth)
N75GE)
N80F#17 (MILL FEED)
N85(Calculate the end point of the other tooth corner)
```

- Real-time indicators of the meanings of the G codes. Place the cursor on the G code and the definition is displayed in the prompt area of the screen.
- Simplified and friendlier access to common editing tasks, such as jumping and searching operations.
- Keyboard shortcuts:
  - Ctrl + Home—jump to the beginning of program
  - Ctrl + End—jump to the end of program
  - Ctrl + C—copy selected text to the clipboard
  - Ctrl + X—cut the selected text from screen and copy to clipboard
  - Ctrl + V—paste the copied or cut text
  - Ctrl + Z—undo the last change
  - Ctrl + Y—redo the last undo

The status bar provides real-time updates on the status on the program being edited. The information shown in the status bar is as follows:



1	Part Program Name
2	Coding Standard
3	Current Line Number
4	Insert/Overstrike Mode
5	Block Skip On/Off

Figure 1–2. Status Bar in NC Editor

The Part Program name is shown first in the status bar. Coding standard can either be Industry Standard (I) or Hurco NC (H). Current Line number indicates the current location of the cursor in the part program. Insert mode is indicated as INS, while overstrike mode is indicated as OVR.

Refer to *NC Editor Menus, on page 1 - 10* for information about the NC Editor menus.

Starting a New NC Program

To begin NC part programming, press the Input key. Refer to *Getting Started with WinMax Mill*, Program Manager section, for information about saving, opening, and loading programs.

The NC file extension is set in User Preferences. Refer to *Getting Started with WinMax Mill*.

These steps help determine the most efficient tool movement and basic program structure to save time during programming:

1. Determine the tool path on the print and label the points where the path direction changes.
2. Make a chart showing the coordinates of each point identified in the previous step.
3. Identify the spindle movements that will be necessary during cutting.

NC Programming Rules

Here are some basic rules to follow when creating NC part programs:

- The axis letter always precedes the numeric information.
- In most cases an integer works the same as a decimal or real number. In the following cases an integer is scaled by the appropriate scaling factor to maintain compatibility with existing NC programs:

  Feedrate: F (BNC only)

  Rotation: R (ISNC Only)

  Dwell: P, X (Both BNC and ISNC)

  Scaling: P (ISNC only)

  ⇨ If an integer is below the acceptable range after scaling, a “Below Minimum Value” error message occurs.

- All axis dimensions are considered to be positive unless a minus sign is entered. When describing *axis motion*, the codes for the program block must contain the following information in order to move properly:
  - Axis identification (e.g., X, Y, Z).
  - Direction the axis will move (+ or -).
  - Distance the axis will move (e.g., 4.0).
  - Enter the speed preceded by the F address character to program a *feedrate* in a block.
  - Include a Z parameter in the NC part program to permit the system to draw the part on the graphics screen. An absolute Z command must occur after a tool change before making another move command.

NC Editor Menus

Hurco's NC system provides many levels of program editing, as well as editing tools, to simplify the task.

The NC Editor contains these top level menus:

Basic Programming Menu

While entering NC codes to create blocks, you may wish to insert new blocks, delete blocks, or display different sections of the part program on the screen.

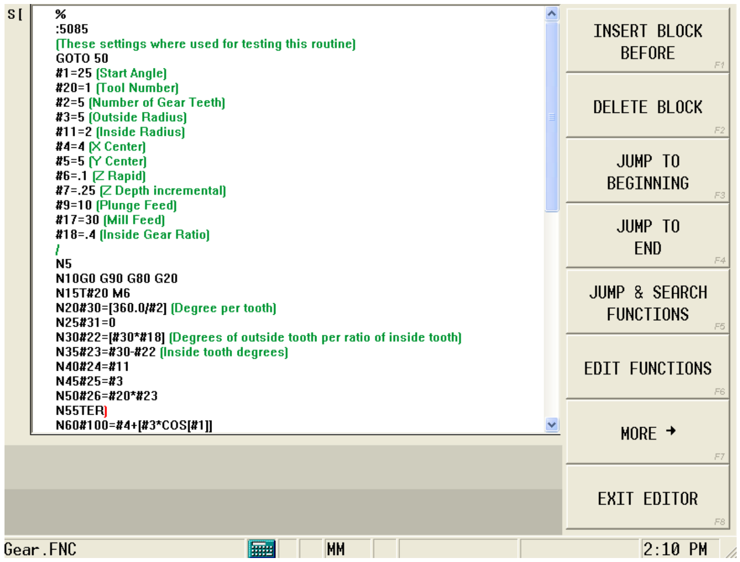


Figure 1–3. NC Basic Programming Menu

The Basic Programming softkeys provide these functions:

- **Insert Block Before—**Inserts a blank line before the block where the cursor is located. This permits addition of a new block of data. This softkey will be disabled if text can't be inserted at the current cursor location.
- **Delete Block—**Removes the block where the cursor is positioned. This softkey will be disabled if the block can't be deleted.
- **Jump to Beginning—**Moves the cursor to the beginning of the first program block in memory. If a keyboard is available, Ctrl + Home combination will result in the same action.
- **Jump to End—**Moves the cursor to the beginning of the last program block in memory. If the keyboard is available, Ctrl + End combination will result in the same action.
- **Jump and Search Functions—**Invokes Jump and Search Functions menu.
- **Edit Functions—**Invokes Edit Functions menu
- **More ->—**Displays the next menu.
- **Exit Editor—**Exits the NC Editor and moves to the Input screen. Select Part Programming to return to the NC Editor screen.

Jump and Search Functions Menu

The Jump and Search provides the flexibility to locate hard-to-find items in program memory using the block or sequence number or searching for specific address characters, numeric parameters, or words.

Figure 1–4. NC Jump and Search Functions Menu

When the Jump & Search Functions softkey is selected from the Basic Programming menu, these softkeys appear:

- **Jump to Block Number—**enter a block number in the popup box. The cursor is positioned on the specified block. The number entered refers to the position that block has in NC memory.

  ⇨ You can also use the touchscreen to select the Line number in the Editor status bar to open the Jump to Block popup box.

- **Jump to Sequence Number—**enter a sequence number (N code) in the popup bo.The cursor is positioned on that block.
- **Search for Text—**activates the Search submenu.
- **Jump to Beginning—**moves the cursor to the beginning of the first program block in memory. If a keyboard is available, Ctrl + Home combination will result in the same action.
- **Jump To End—**moves the cursor to the beginning of the last program block in memory. If the keyboard is available, Ctrl + End combination will result in the same action.
- **Jump To Previous Syntax Error—**tries to find the previous syntax error, and if successful, takes the user to that line.
- **Jump To Next Syntax Error—**tries to find the next syntax error, and if successful, takes the user to that line.
- **Exit—**returns to the previous menu.

Search Submenu

The Search submenu allows the user to search for specific text in the part program.

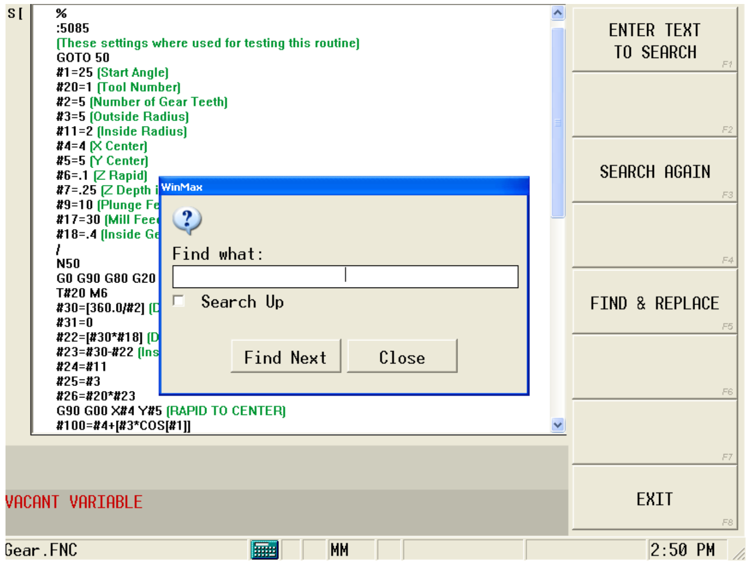


Figure 1–5. NC Search Submenu

The menu contains the following softkeys:

- **Enter Text To Search**—opens a popup box where search text is entered. Type search term and select the Find Next button. The NC editor finds the next occurrence of the entered text following the current cursor position. If found, the text is highlighted. Selecting the Search Up checkbox will search for the search term prior to the current cursor position.
- **Search Again—**repeats the last search operation without prompting the user.
- **Find and Replace**—opens a popup box where search text is entered and replacement term is specified.
  - The **Replace Next** button finds the next instance of the search term and replaces it with the new term. To find and replace the next instance you must select the Replace Next button again.
  - The **Replace All** button finds and replaces all instances of the search term.
- **Exit**—invokes the Jump and Search menu.

Edit Functions Menu

The Edit menu provides advanced editing functionality to the user.

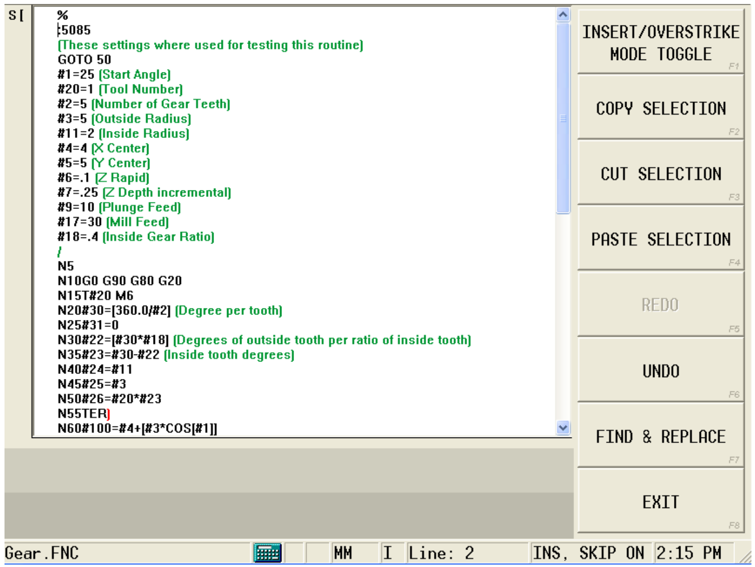


Figure 1–6. NC Edit Functions Menu

When the Edit softkey is selected from the Basic Programming menu, these softkeys appear:

- **Insert/Overstrike Mode Toggle—**switches the data entry style between insert and overwrite. Currently active mode is shown on the status bar of the window - “INS” indicates Insert and “OVR” indicates Overwrite.

  ⇨ You can also use the touchscreen to toggle the Insert/Overstrike Mode in the Editor status bar.

- **Copy Selection—**this softkey copies the selected text to the Windows' clipboard memory. If the keyboard is available, this action can be achieved by pressing Ctrl + C combination. Text can be selected by dragging the stylus (up/down/right/left) across the editing area of the screen, or by holding down the keyboard Shift key and using the up/down/left/right arrow keys to select text.
- **Cut Selection—**copies the selected text to the Windows clipboard. In addition, the selected text is deleted from the part program. If the keyboard is available, Ctrl + X combination will cut the selected text and place it in the Windows clipboard.
- **Paste Selection—**text previously copied to the Windows' clipboard in inserted at the current cursor position. If the keyboard is available, this action can be achieved by pressing Ctrl + V keys.
- **Redo—**each click redoes the editing operation(s) previously undone by the Undo softkey. If the keyboard is available, Ctrl + Y combination will result in the same action.
- **Undo—**each click on undoes the previous editing operations starting from the latest to the earliest. The depth of the undo buffer is 100. If more than one text modification was performed in a single step (e.g. “Replace All”), all those modifications will be undone in one step as well. If the keyboard is available, Ctrl + Z combination will result in the same action.

- **Find & Replace—**activates a popup box where search text and replacement text are entered. NC editor finds occurrences of the search text and replaces it with the replacement text.
- **Exit—**loads the Basic Programming menu.

Renumbering and Tagging Menu

Renumbering and tagging menu offers access to renumbering and tagging functionality. This menu can be accessed by pressing More-> from the Basic programming menu.

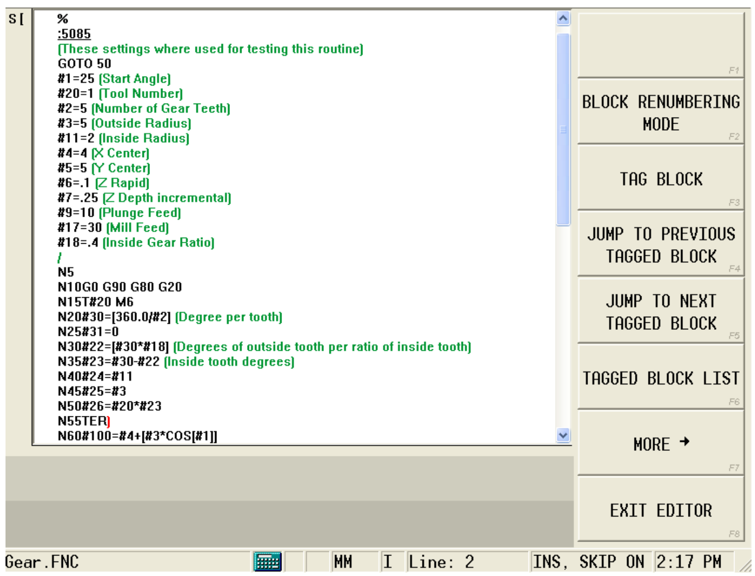


Figure 1–7. NC Renumbering and Tagging Menu

The menu contains the following softkeys:

- **Block Renumbering mode—**invokes Block Renumbering Mode submenu.
- **Tag Block—**tags a block by underlining it. There is no limit on the amount of tagged blocks you can have.
- **Jump to Previous Tagged Block—**NC editor tries to find the closest tagged block before the current cursor's position. If such block is found, the cursor is placed in the beginning of the block.
- **Jump to Next Tagged Block—**NC editor tries to find the closest tagged block after the current cursor's position. If such block is found, the cursor is placed in the beginning of the block.
- **Tagged Block List—**invokes Tagged Blocks List screen, which allows the user to review the tagged block and manage the tags.
- **More->—**invokes program Execution menu.
- **Exit Editor—**this softkey operates the same as in the Basic Programming menu.

Block Renumbering Submenu

The Block Renumbering mode submenu allows the user to search for specific text in the part program.

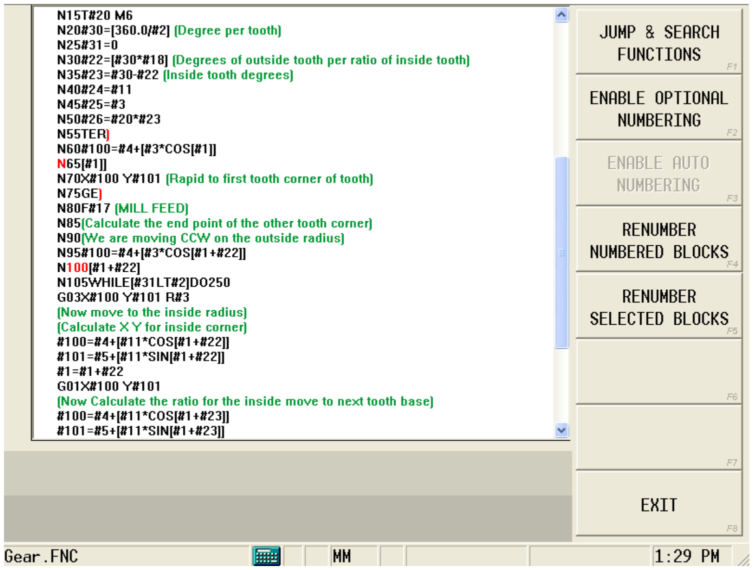


Figure 1–8. NC Block Renumbering Mode Submenu

The menu contains the following softkeys:

- **Jump & Search Functions—**opens the Jump and Search Functions menu.
- **Enable Optional Numbering**—enables user-assigned block numbering mode.
- **Enable Auto Numbering**—enables automatic block numbering mode.
- **Renumber Numbered Blocks—**enables block renumbering. The renumbering interval is entered into the popup box and the blocks are automatically renumbered.
- **Renumber Selected Blocks**—enables renumbering of selected blocks. First select the blocks to be renumbered by dragging the stylus (up/down/right/ left) across the editing area of the screen, or by holding down the keyboard Shift key and using the up/down/left/right arrow keys to select text. Then select the softkey and specify the renumbering interval. The selected blocks are renumbered.
- **Exit—**invokes the Renumbering and Tagging menu.

Program Execution Menu

Program Execution menu allows selection of the part program to be processed for graphics and for execution. This menu can be accessed by twice pressing More-> (first from the Basic programming menu and then from the Renumbering and Tagging menu.)

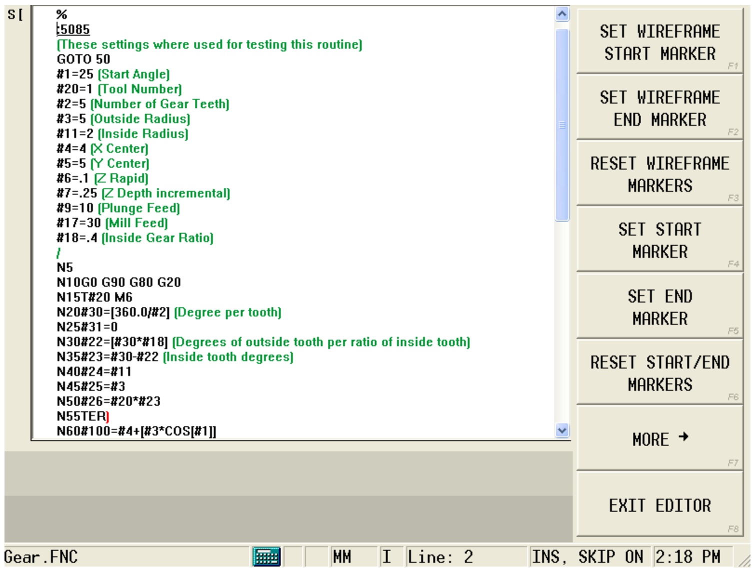


Figure 1–9. NC Program Execution Menu

The menu contains the following softkeys:

- **Set Wireframe Start Marker—**marks the current block as the beginning point for the graphics display by inserting a left bracket ([) to the left of the block.
- **Set Wireframe End Marker—**marks the current block as the ending point for the graphic display by inserting a right bracket (]) to the left of the block.

  ⇨ If the Wireframe Start and Wireframe End markers overlap, the pound (#) sign is displayed to the left of the line.

- **Reset Wireframe Markers—**returns start and end wireframe markers to their default locations. The defaults are at the beginning and end of the program.
- **Set Start marker—**indicates the block that the system should use to start program execution when running and verifying the program. A letter S is inserted to the left of the block.
- **Set End marker—**indicates the block that system should use to end program verification and execution. A letter E is inserted to the left of the block.

  ⇨ If the Start and End markers overlap, the dollar ($) sign is displayed to the left of the line.

- **Reset Start/End Markers—**restores start and end markers to their defaults. The defaults are at the beginning and end of the program.

- **More->—**invokes NC Editor Settings menu.
- **Exit Editor—**this softkey operates the same as in the Basic Programming menu.

NC Editor Settings Menu

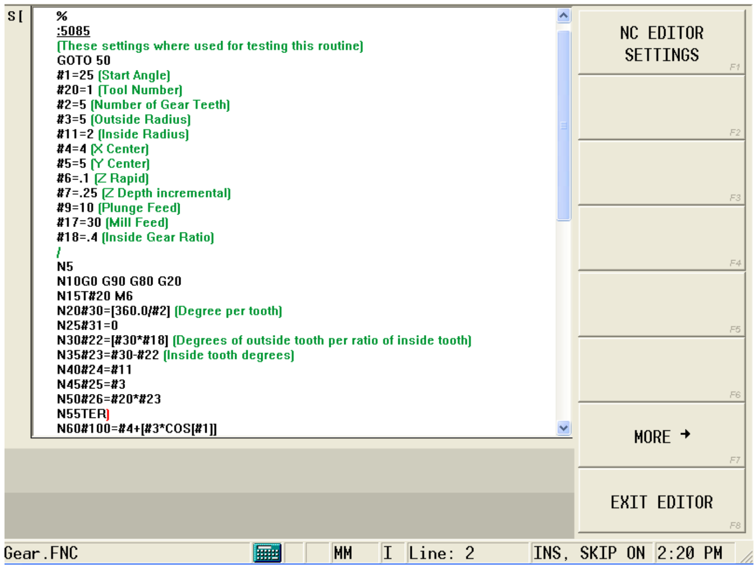


Figure 1–10. NC Editor Settings menu

- **NC Editor Settings—**invokes NC Editor Settings screen, which allows the user to modify editor's behavior. Currently, Block Skip Enable is the only option available for the user.
  - Block Skip Enable—skips the NC codes which follow the Ignore (/) character.
- **More ->—**invokes Basic Programming menu.
- **Exit Editor—**this softkey operates the same as in the Basic Programming menu.

NC Parameters

The **NC Parameters** softkey accesses these functions:

NC Configuration Parameters (Screen 1)

Use this screen to change general NC part program parameters.

The NC Configuration Parameters fields are defined as follows:

- **Least Dwell Units—**specifies the dwell units when using an integer to specify dwell. This field can be set to 0.001 or 0.0001.
- **Least Scaling Factor—**specifies the units of the scaling factor when an integer is used with the scaling command. This field can be set to 0.001 or 0.0001.
- **Disable X / Y / Z Scaling—**disables any scaling performed in the X, Y, or Z axis, respectively.
- **Reference Point X / Y / Z—**specifies the reference point for the G28 command. The WinMax software allows you to select the tool change positions (X, Y, Z) or maximum travel limit (X, Y).
- **Tool Length Tolerance—**determines value used for tool probing.
- **NC Optional Program Stop**—specifies if program should stop on M01 command.

NC Configuration Parameters (Screen 2)

- **M6 Initiates Tool Change—**initiates a tool change at the M06 code when set to Yes. If set to No, tool changes are initiated by a T code.
- **Default Tool Number—**indicates which tool will be used at the beginning of the program.
- **Allow Vacant Variables—**allows variable values to be left blank when Yes.
- **Assume Feedrate .1 Increment—**assumes a feedrate increment of .1 when Yes is selected.
- **Default Cutter Comp Lookahead**—identifies the number of segments in a contour that are checked to determine if the contour crosses itself and if the tool will fit into the contour. The default is 1. The range is from 1 to 9.

Use the default value of 1 look ahead block and test the program using Graphics. If the contour crosses itself when set to 1 block, reset this value to a higher number and retest using the graphics screen. The smaller the number of look ahead blocks, the more efficient your program will run, saving time and control memory.

- **Enable Lead In Calculations**—enables software to calculate cutter compensation lead in path.
- **Enable Lead Out Calculations**—enables software to calculate cutter compensation lead out path.

NC M and G Code Program Numbers

Use this screen to change M code and G code program numbers for an NC part program using the NCPP option.

The NC M and G Code Program Numbers fields are defined as follows:

- **Enable User M/G Codes—**enables user customization of M codes and/or G codes to perform specialized tasks. User defined M and G codes define a custom code which performs a specialized task, replace an existing G or M code, or provide compatibility between different NC dialects from various machine tool control manufacturers.
- **Enable User S/B/T Codes—**enables user customization of S codes, B codes, or T codes to perform specialized tasks. User defined S, B, or T codes replace spindle and tool functions with custom subprograms.
- **M-Code—**allows programming of customized M codes. Up to 13 user defined M codes can be programmed from M01 through M255 (except M02, M98, and M99). Negative numbers cannot be entered in the column for user defined M codes. Enable programmed M codes using the Enable User M Codes function.
- **G-Code—**allows programming of customized G codes. Up to 10 user defined G codes can be programmed from G01 through G255 (except G65, G66, and G67). If a negative number is entered for a user defined G code, the subprogram becomes modal. Enable programmed G codes using the Enable User G-Codes function.

NC Variables

Use this screen to define Global and System NC variable codes and subprograms for an NC part program using the NCPP option. Programs with variables can be reused. All variables must begin with the "#" character followed by a valid, writeable register number and an equal sign. The following example sets the variable value (#500) to 100:

500 = 100

Some variables are read only when an operator attempts to write to the variable.

There are four types of variables that can be used in NC programming:

- **Global—**can be used by all programs. Assign a global variable before it is used in an equation or expression, or the variable will be considered vacant, generating an error unless the Allow Vacant Variables field is set to Yes. Use the Global 100-199 and Global 500 - 999 softkeys to enter global variables on the NC Variables screen.
- **System—**provide information about the state of the system such as X, Y, and Z external work compensation, miscellaneous system parameters, modal information, position information, and G code group status. Use the **Tool Offset 2001-2200**, **Work Offset 2500-2999**, **Misc 3000-3014**, **Modal 4001-4320**, and **Position 5001-5083** softkeys to enter system variables on the NC Variables screen.
- **Local—**are valid only within the current program. These variables are only available in Macro Mode B and range from #1 to #33. Enter local variables in the NC Editor screen. Refer to *NC Editor, on page 1 - 8*.
    - Assign a value to the local variable before it is used in an equation or expression, or the variable will be considered vacant, generating an error unless the Allow Vacant Variables field is set to Yes.
    - When the subprogram call is not made using an M98, local variables are nested, meaning that when a subprogram call is made, a new set of local variables is received and the old set is stored. After leaving the subprogram, these local variables are destroyed and the previous set is restored.
    - Passing parameters to subprograms automatically initializes local variables when subprogram calls other than M98 are made.
- **Arguments—**available only Macro Mode A. Arguments are used to pass parameters to subprograms. Parameters are the addresses which follow G65, G66, and M98 codes. Enter arguments in the NC Editor screen. Refer to *NC Editor, on page 1 - 8*.

  Refer to the Macro Mode A G Code Group Status table.

Use the softkeys to select the type of NC Variable to appear on the NC Variables screen:

- Global 100-199
- Global 500 - 999
- Tool Len Offset 2001-2200
- Work Offset 2500-3000

The **More ➔** softkey accesses these softkey choices:

- Misc 3000-3021
- Position 5061-5083
- Tool Dia Offsets 12001-12200

The **More ➔** softkey returns to the first softkey menu described above.

The **Toggle Units** softkey toggles the dimensional variables (Tool Offset, Work Offset, Position) between inch and metric.

Macro Mode A Subprogram Variables

In this table, the values for the NC parameters are stored in addresses #8004 to #8026 for Macro Mode A subprogram calls.

The status for each variable is stored in address #8104 to #8126.

The status for the variables is non-zero (>1) if an argument is specified in the subprogram call, and zero otherwise.

Macro Mode A Subprogram Variables

NC Parameter	Value Address	Type	Status Address	R/WW/WWW
I	#8004	ARG	#8104	R
J	#8005	ARG	#8105	R
K	#8006	ARG	#8106	R
F	#8009	ARG	#8109	R
G	#8010	ARG	#8110	R
H	#8011	ARG	#8111	R
M	#8013	ARG	#8113	R
N	#8014	ARG	#8114	R
P	#8016	ARG	#8116	R
Q	#8017	ARG	#8117	R
R	#8018	ARG	#8118	R
S	#8019	ARG	#8119	R
T	#8020	ARG	#8120	R
X	#8024	ARG	#8124	R
Y	#8025	ARG	#8125	R
Z	#8026	ARG	#8126	R

Table 1–1. Macro Mode A Subprogram Variables

Macro Mode A G Code Group Status

In this table, the value for each G Code Group is stored in addresses #8030 to #8046 for Macro Mode A subprogram calls G65, G66, and user defined G codes and M codes.

The status for each G Code Group is stored in addresses #8130 to #8146.

The status is non-zero if an argument is specified in the subprogram call, and empty otherwise.

Macro Mode A G Code Group Status

G Code	Value Address	Type	Status Address	R / W
00	#8030	ARG	#8130	R
01	#8031	ARG	#8131	R
02	#8032	ARG	#8132	R
03	#8033	ARG	#8133	R
05	#8035	ARG	#8135	R
06	#8036	ARG	#8136	R
07	#8037	ARG	#8137	R
08	#8038	ARG	#8138	R
09	#8039	ARG	#8139	R
10	#8040	ARG	#8140	R
11	#8041	ARG	#8141	R
15	#8045	ARG	#8145	R
16	#8046	ARG	#8146	R

Table 1–2. Macro mode A G Code Group Status

NC Probing Part Setup

It is possible to perform a part probe setup using calls to a set of predefined subprograms. A subprogram is a group of commands stored under one name. These probing subprogram calls mimic the probe part setup conversational data block.

The first five subprogram calls (P1000 through P5000) are used to set internal reference locations that perform the probing function. The sixth subprogram, P6000, performs the probing operation.

In addition, an NC program utilizing G31 commands can be used to perform a probing part setup.

⇨ Probing is used for setting up part zero for G54 only.

Here is an example of the NC codes for probing part setup:

G165 P1000 X0.0 Y0.0

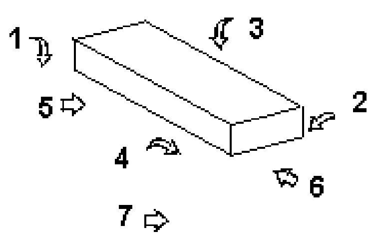
G165 P2000 X1.0 Y8.0 Z-10.0

G165 P3000 X4.0 Y1.0 Z-10.0

G165 P4000 X4.0 Y2.0 Z-5.0

G165 P5000 X1.0 Y2.0 Z-10.0

G165 P6000 X+1 Y+1 A1



1	P2000 XYZ X Start Location
2	P3000 XYZ Y Start Location
3	P4000 XYZ Z Start Location
4	P5000 XYZ Skew Start Location
5	X Probe Direction
6	Y Probe Direction
7	Skew Axis: X

The previous example illustrates these subprograms:

- P1000 is used to set the X and Y reference locations.
- P2000 is used to set the X Start Location.
- P3000 is used to set the Y Start Location.
- P4000 is used to set the Z Start Location.
- P5000 is used to set the Skew Start Location.
- P6000 is used to set the X and Y direction (+1.0 means positive -1.0 means negative) and the Skew axis (A = 1 for X axis, A = 2 for Y axis, any other value or no A parameter indicates no skew axis).

⇨ The P1000 to P5000 subprograms must be used prior to P6000. Once a P6000 is used, the internal reference locations are reset to zero after the probing operation is performed. To retry the part setup, the P1000 to P5000 subprograms must be reset before P6000.

PREPARATORY FUNCTIONS - G CODES

This section defines G codes and their functions. This information is often needed when using an off-line CAM or CAD/CAM system to create NC part programs

G Code Groups

The G codes are grouped by functions.

Group	Function	Group	Function
00	One-Shot	10	Return from Canned Cycles
01	Interpolation	11	Scaling
02	Plane Selection	12	Macro/Subprogram
03	Dimension	14	Coordinate System Selection
05	Feed	15	Precision Cornering
06	Measurement	16	Rotation
07	Cutter Compensation	17	Polar Coordinates
08	Tool Length Compensation	18	Mirroring
09	Canned Cycles	19	Program Parameters for Surface Finish/Data Smoothing

Table 2–1. G Code Groups

⇨

The system displays the number 010 as an alarm if an invalid G code (one not listed in the following table) is entered.

More than one G code can be specified in the same block. If more than one is from the same group, the last G code entered is active.

Specifying a group 01 (Interpolation) G code in a canned cycle automatically enters the G80 condition (Canned Cycle Cancel). Conversely, a group 01 G code is not affected by the canned cycle G codes.

G Code Table

The following table lists the G codes, identifies the defaults (in the shaded areas), lists Modal (M) or Non-modal (N) types, identifies groups, and describes the G codes' functions.

Some G codes are strictly BNC or strictly ISNC, and are identified as such in this manual. Otherwise, the G codes apply to either dialect.

G Code	Type	Group	Function
G00	M	01	Positioning (Rapid Traverse)
G01	M		Linear Interpolation (Cutting Feed)
G02	M		Circular Interpolation/Helical CW
G02.4	M		3D Circular Interpolation CW
G03	M		Circular Interpolation/Helical CCW
G03.4	M		3D Circular Interpolation CCW
G04	N	00	Dwell, Exact Stop
G05.1	M	19	Surface Finish Parameters
G05.2	M		Data Smoothing
G05.3	M		Surface Finish Quality
G07.2	M	00	Cylindrical Rotary Wrap On
G07.3	M	00	Cylindrical Rotary Wrap Off
G08.1	M	00	ASR Command Buffer On
G08.2	M	00	ASR Command Buffer Off
G09	N	00	Decelerate Axis to Zero
G10	N		Data Setting
G11	N		Data Setting Mode Cancel
G15	M	17	Polar Coordinates Cancel
G16	M		Polar Coordinates
G17	M	02	XY Plane Selection
G18	M		ZX Plane Selection
G19	M		YZ Plane Selection
ISNC G20	M	06	Input in Inch
ISNC G21	M		Input in mm
G28	N	00	Return to Reference Point
G29	N		Return from Reference Point
G31	N		Skip Function

G Code	Type	Group	Function (Continued)
G40	M	07	Cutter Compensation Cancel
G41	M		Cutter Compensation Left
G41.2	M		3D Tool Geometry Compensation
G42	M		Cutter Compensation Right
G43	M	08	Tool Length Compensation + Direction
G43.4	M		5-Axis Linear Interpolation
G44	M		Tool Length Compensation - Direction
G45	N	00	Tool Offset Increase
G46	N		Tool Offset Decrease
G47	N		Tool Offset Double Increase
G48	N		Tool Offset Double Decrease
G49	M	08	Tool Length Offset Compensation Cancel
G50	M	11	Scaling Cancel
G51	M		Scaling
G50.1	M	18	Mirroring Cancel
G51.1	M		Mirroring
G52	N	00	Local Coordinate System Setting
G53	N		Machine Coordinate System Selection
G54	M	14	Work Coordinate System 1 Selection
G54.1	M		Aux Work Coordinate Systems
G55	M		Work Coordinate System 2 Selection
G56	M		Work Coordinate System 3 Selection
G57	M		Work Coordinate System 4 Selection
G58	M		Work Coordinate System 5 Selection
G59	M		Work Coordinate System 6 Selection
G61	M	15	Decelerates to Zero–Precision Cornering
G64	M		Cancels Precision Cornering
G65	N	12	Macro Command, Subprogram Call
G66	M		Modal Subprogram Call
G67	M		Modal Subprogram Call Cancel
G68	M	16	Coordinate Rotation

G Code	Type	Group	Function (Continued)
G68.2	M		Global Rotation NC Transform Plane
G68.3	M		Local Rotation NC Transform Plane
G69	M		Coordinate System Rotation Cancel
BNC G70	M	06	Input in Inch
BNC G71	M		Input in mm
G73	M	09	Peck Drilling Cycle
ISNC G74	M		Left-handed Tapping Cycle
ISNC G74 with M29	M		Rigid Tapping
BNC G74	M	01	Single-quadrant Circular Interpolation
BNC G75	M		Multi-quadrant Circular Interpolation
G76	M	09	Bore Orient Cycle
G80	M		Canned Cycle Cancel
G81	M		Drilling Cycle, Spot Boring
G82	M		Drilling Cycle, Counter Boring
G83	M		Peck Drilling Cycle
G84	M		Tapping Cycle
ISNC G84.2	M		Rigid Tapping Cycle
ISNC G84.3	M		Rigid Tapping Cycle
ISNC G84 with M29	M		Rigid Tapping Cycle
G85	M		Boring Cycle
BNC G86	M		Bore Orient Cycle
ISNC G86	M		Bore Rapid Out Cycle
BNC G87	M		Chip Breaker Cycle
ISNC G87	M		Back Boring Cycle
BNC G88	M		Rigid Tapping Cycle
ISNC G88	M		Boring Cycle Manual Feed Out, Dwell
G89	M		Boring Cycle Bore and Dwell
G90	M	03	Absolute Command
G91	M		Incremental Command
G92	N	00	Programming of Absolute Zero Point

G Code	Type	Group	Function (Continued)
G93	M	05	Inverse Time
G94	M		Feed per Minute
G94.1	M	05	Rotary Tangential Velocity Control
G98	M	10	Return to Initial Point in Canned Cycle
G99	M		Return to R Point in Canned Cycle

Table 2–2. G Codes in order of Codes

The following table lists the G codes by group, identifies the defaults (in the shaded areas), lists Modal (M) or Non-modal (N) types, and describes the G codes' functions.

Group	G Codes	Type	Function
00	G04	N	Dwell, Exact Stop
	G07	M	Cylindrical Rotary Wrap
	G08	M	ASR Command Buffering
	G09	N	Decelerate Axis to Zero
	G10	N	Data Setting
	G11	N	Data Setting Mode Cancel
	G28	N	Return to Reference Point
	G29	N	Return from Reference Point
	G31	N	Skip Function
	G45	N	Tool Offset Increase
	G46	N	Tool Offset Decrease
	G47	N	Tool Offset Double Increase
	G48	N	Tool Offset Double Decrease
	G92	N	Programming of Absolute Zero Point
01	G00	M	Positioning (Rapid Traverse)
	G01	M	Linear Interpolation (Cutting Speed)
	G02	M	Circular Interpolation/Helical CW
	G02.4	M	3D Circular Interpolation CW
	G03	M	Circular Interpolation/Helical CCW
	G03.4	M	3D Circular Interpolation/Helical CCW
	BNC G74	M	Single-quadrant Circular Interpolation
	BNC G75	M	Multi-quadrant Circular Interpolation
02	G17	M	XY Plane Selection
	G18	M	ZX Plane Selection
	G19	M	YZ Plane Selection

Group	G Codes	Type	Function (Continued)
03	G90	M	Absolute Command
	G91	M	Incremental Command
05	G93	M	Inverse Time
	G94	M	Feed per Minute
	G94.1	M	Rotary Tangential Velocity Control
06	BNC G70	M	Input in Inch
	BNC G71	M	Input in mm
07	G40	M	Cutter Compensation Cancel
	G41	M	Cutter Compensation Left
	G41.2	M	3D Tool Geometry Compensation
	G42	M	Cutter Compensation Right
08	G43	M	Total Length Compensation + Direction
	G43.4	M	5-Axis Linear Interpolation
	G44	M	Total Length Compensation – Direction
	G49	M	Tool Length Offset Compensation Cancel

09	G73	M	Peck Drilling Cycle
	ISNC G74	M	Left-handed Tapping Cycle
	ISNC G74 with M29	M	Rapid Tapping
	G76	M	Bore Orient Cycle
	G80	M	Canned Cycle Cancel
	G81	M	Drilling Cycle, Spot Boring
	G82	M	Drilling Cycle, Counter Boring
	G83	M	Peck Drilling Cycle
	G84	M	Tapping Cycle
	ISNC G84.2	M	Rigid Tapping Cycle
	ISNC G84.3	M	Rigid Tapping Cycle
	ISNC G84 with M29	M	Rigid Tapping Cycle
	G85	M	Boring Cycle
	BNC G86	M	Bore Orient Cycle
	ISNC G86	M	Bore Rapid Out Cycle
	BNC G87	M	Chip Breaker Cycle
	ISNC G87	M	Back Boring Cycle
	BNC G88	M	Rigid Tapping Cycle
	ISNC G88	M	Boring Cycle Manual Feed Out, Dwell
	G89	M	Boring Cycle, Bore and Dwell

Group	G Codes	Type	Function (Continued)
10	G98	M	Return to Initial Point in Canned Cycle
	G99	M	Return to R Point in Canned Cycle
11	G50	M	Scaling Cancel
	G51	M	Scaling
12	G65	N	Macro Command, Subprogram Call
	G66	M	Modal Subprogram Call
	G67	M	Modal Subprogram Call Cancel
14	G54	M	Work Coordinate System 1 Selection
	G55	M	Work Coordinate System 2 Selection
	G56	M	Work Coordinate System 3 Selection
	G57	M	Work Coordinate System 41 Selection
	G58	M	Work Coordinate System 5 Selection
	G59	M	Work Coordinate System 6 Selection
15	G61	M	Decelerates to Zero-Precision Cornering
	G64	M	Cancels Precision Cornering
16	G68	M	Coordinate Rotation
	G68.2	M	Global Rotation NC Transform Plane
	G68.3	M	Global Rotation NC Transform Plane
	G69	M	Coordinate System Rotation Cancel
17	G15	M	Polar Coordinate Cancel
	G16	M	Polar Coordinates
19	G05.1	M	Surface Finish Parameters
	G05.2	M	Data Smoothing
	G05.3	M	Surface Finish Quality

Table 2–3. G Codes in order of Groups

Rapid Traverse (G00)

Rapid Traverse mode (G00) is the default and moves the axes to a specified location at the rapid feedrate programmed in the Program Parameters screen. Up to five axes (X, Y, Z, A, B) of coordinated rapid motion can be specified while in this mode.

Set either a linear or non-linear tool path on the NC Parameters screen. The linear tool path is the default. ISNC and BNC have different linear tool path modes. In the ISNC linear mode the tool motion is in all three axes (X, Y, Z) simultaneously. In the BNC linear mode the motion is divided into separate X, Y, and Z moves. The motion in the XY plane is a straight line.

The ISNC and BNC non-linear modes are the same. In the non-linear tool path mode, the XY plane motion is broken down into a 45° move and a straight line move parallel to either the X or Y axis. The determination of whether the 45° move or the straight line move is made depends first on the distances from the current position to the end position along the X and Y axes.

If it is desired that the tool move to a position which is compensated, G41 or G42 needs to be specified along with the offset before any axis coordinates are given. The rapid traverse rate is set on the General Parameters screen.

The G00 mode is canceled by using the G01, G02, G03, or canned cycle (G73, G76, G81–G89) commands.

G17, G18, or G19 determine plane of offset.

G90 specifies absolute dimensioning and G91 specifies incremental dimensioning.

Another work coordinate system can be selected by using commands G54 through G59.

Format

The format of the rapid traverse command is as follows:

G00 X_____Y_____Z_____A_____B_____

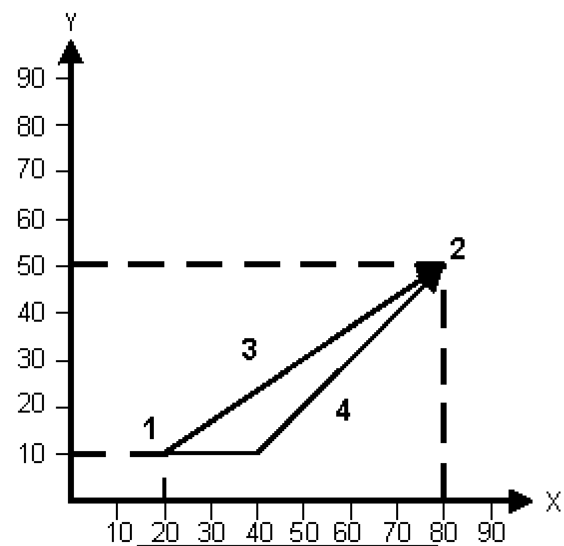
Example

If one axis of movement is specified in a G00 block, that axis moves at the rapid traverse feedrate. When two axes of movement are specified in a G00 block, the rapid traverse feedrate is assigned to the longest vector component. The resulting feed that appears on the screen may actually exceed the rapid traverse feedrate parameter.

If a block containing a G00 word also contains a Z word that causes the Z-axis to move away from the part, the Z-axis moves first. The other specified axes then move in linear or non-linear mode at the rapid feedrate to their specified end points. If Z is to move toward the part, all axes except Z move in linear or non-linear mode at rapid feedrate to their specified end points; then Z moves down to its end point. If no Z is programmed, all axes move at rapid feedrate coordinated to the specified end point. G00 is a member of the tool positioning code group and is *canceled* by G01, G02, G03, and G81–G89.

This code is used for positioning only and should never be used for cutting material.

The diagram below shows the two different rapid traverse modes:



1	Start Point
2	End Point
3	Linear
4	Non-Linear

Figure 2–1. G00 Axis Movement

Linear Interpolation (G01)

The Linear Interpolation code (G01) moves the axes to a specified location at the programmed feedrate. Up to five axes (X, Y, Z, A, B) of coordinated motion can be specified while in this mode. The programmed feedrate can be changed by adding an F word to any NC block while in this mode. X, Y, Z, A, B, and F dimensions need to be supplied only if they change.

G01 is a member of the tool positioning code group and is *canceled* by G00, G02, G03 and the canned cycle (G73, G76, G81–G89) commands.

G90 specifies absolute dimensioning and G91 specifies incremental dimensioning.

G41 or G42 may be optionally selected if a cutter offset is desired.

This code is used when the tool is in contact with the work piece to cut a line parallel to an axis or at an angle to an axis.

Format

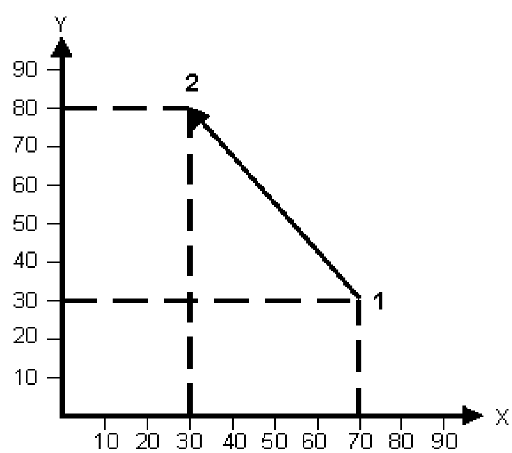
The format of the linear interpolation command is as follows:

G01 X_____ Y_____ Z_____ A_____ B_____ F_____

F specifies the associated feedrate along the tool path. If rotary axis parameters (A and B) are used, the feedrate units are in degrees/minute.

Example

The diagram below illustrates the linear interpolation axis movement:



1	Start Point
2	End Point

Figure 2–2. G01 Axis Movement

Circular and Helical Interpolation (G02 and G03)

These two codes are members of the tool positioning code group. The Clockwise Circular or Helical Interpolation code (G02) causes the axes to generate an arc or helix in a clockwise direction. The previous block's end point defines the start point of the arc.

⇨ If the first segment in a contour is an arc with a radius smaller than the radius of the tool, the +-control will generate an error message indicating that you need to use a tool with a smaller radius or program a larger radius.

The Counterclockwise Circular or Helical Interpolation code (G03) causes the axes to generate an arc or helix in a counterclockwise direction. The previous block's end point defines the start point of the arc.

⇨ Calculate the linear feedrate to verify that it does not exceed various limit values.

Both G02 and G03 codes are *canceled* by G00, G01, the canned cycle (G73, G76, G81–G89) commands, or by each other.

The programmed feedrate can be changed by adding an F word to any NC block when this code is active.

G17, G18, or G19 specify plane of interpolation.

G41 or G42 may be optionally selected if a cutter offset is desired.

G40 is used to *cancel cutter offset*.

G02 or G03 cannot be used in a start up block in offset mode.

(X,Y) for G17, (X,Z) for G18, and (Y,Z) for G19 specify the end location on the selected plane.

⇨ R or the incremental coordinates ((I,J) or (I,K) or (J,K)) specify the arc center location. The R is modal and stays in effect until another R value is specified or (I,J) is used. With the R (radius) parameter, you specify the radius. You do not need to calculate the center point.

- A positive R produces an arc of less than or equal to 180°.
- A negative R produces an arc of greater than or equal to 180°.
- The R takes precedence over an I, J, or K in the same block.

For BNC, I, J, K, and R are modal for G02 and G03. The I, J, and K center point location is incremental from start point in G91 mode and absolute coordinates in G90 mode.

For ISNC, when G02 or G03 are specified, the I, J, and K are reset to 0.0. They remain modal until another G02 or G03 is encountered. R is not reset to 0.0. For ISNC, the I, J, and K are incremental from the start point in both G90 and G91 mode.

You can specify an R value for arcs when the arc is in the G17 XY plane or the G19 YZ plane.

Arcs use the right-hand coordinate system for all planes, except when using Basic NC for the G18 XZ plane. Arcs use a left-hand coordinate system when using the Basic NC for the G18 XZ plane.

⇨ When interpreting an arc in ISNC, a helix will be the result if a valid center point was established in the previous block and only a Z value is given.

F specifies the feedrate in degrees/minute along the arc in the circular plane.

Format

The formats of the Circular Interpolation commands are as follows:

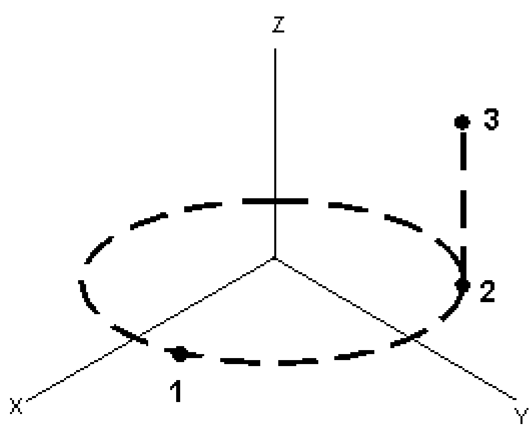
Circular interpolation (Z = 0)
Helical interpolation (Z ≠ 0)

G02/G03 (for G17) X_____ Y_____ {R_____ or [I_____ and J_____]}Z_____ F_____

G02/G03 (for G18) X_____ Z_____ {R_____ or [I_____ and K_____]}Z_____ F_____

G02/G03 (for G19) Y_____ Z_____ {R_____ or [J_____ and K_____]}Z_____ F_____

This diagram illustrates circular and helical interpolation:



1	Start Point
2	End Point (Circular)
3	End Point (Helical)

Figure 2–3. Circular and Helical Interpolation

G02 Example

Consider the following section of an NC program using a G02 code in absolute mode using R to specify the modal turn radius:

NC Part Program 1 Inch G02.FNC

```
%
G00 G90
M25
T1 M06
Z5.05
X2.0 Y0.0
S2000 M03
Z0.05
G01 Z-0.5 F10.
G01 X2.0 Y0.0
G01 X0.5
G02 X0.0 Y0.5 R0.5      ⇐ R needs to be specified only once.
G01 Y2.5
G02 X0.5 Y3.0
G01 X3.5
G02 X4.0 Y2.5
G01 Y0.5
G02 X3.5 Y0.0
G01 X2.0
M25
M05
M02
```

BNC G03 Example

Consider the following section of a BNC program using a G03 code in absolute mode:

NC Part Program 1 Inch G03ABS.HNC

```
%
N110 Z0 G91
N115 T01 M06
N116 X0. Y0. Z0.
N120 F40 S1000 M3
N130 G00 X3. Y4.
N140 G01 X3. Y2. F10
N150 G03 X4. Y1.5858 I4. J3.
N160 X7.4142 Y5. J5.
N170 G01 Y7.
N220 M02
E
```

This is an example of the same geometry using the incremental mode:

NC Part Program 1 Inch G03INC.HNC

```
%
N110 Z0
N115 T01 M06
N116 X0. Y0. Z0.
N120 F40 S1000 M3
N130 G00 X3. Y4.
N140 G01 Y-2. F10
N150 G03 X1. Y-.4142 I1. J1.
N160 X3.4142  Y3. I0. J3.4142
N170 G01 Y2.
N220 M02
E
```

A and B words are not allowed in circular interpolation mode. The programmed feedrate can be changed by adding an F word to any block while in this mode.

X, Y, and Z define the end point of the arc and I, J, and K define the center point of the arc. I represents the X center point; J represents the Y center point; and K represents the Z center point. The X, Y, Z, and F words do not need to be programmed when you are initially setting the circular interpolation mode if they have not changed from the previous block.

For BNC, the I, J, and K dimensions must be specified when initially setting the circular interpolation mode (when a G02 or G03 is in the block) to establish a center point.

For ISNC, I, J, or K are set to 0.0 if they are not initially specified.

Once the circular interpolation mode is set, the X, Y, Z, I, J, K, and F dimensions need to be supplied only if they change. A block with missing dimensions uses the last specified locations.

⇨ A circle or circular helix may be programmed by either using the same end and start point, or by not programming the end points. Ensure that the specified end point is mathematically on the arc.

If the programmed end point is not on the arc or helix, an end point is calculated using the start point, center point, and programmed end point. The start point and center point determine the radius of the arc and thus the distance of the calculated end point from the center point. The center point and programmed end point determine the line on which the calculated end point results.

⇨ Arcs in this system are approximations that are comprised of small line segments or arc chords.

The chord error of arcs and helices may be controlled through the *chord error* parameter in the Program Parameters screen. The default chord error is 0.0001 inches (.003 mm). This creates very smooth arcs, but may limit the maximum feedrate for the arc or helix. Larger chord errors allow higher feedrate arcs or helices, but may be less accurate.

3D Circular Interpolation (G02.4 and G03.4)

The 3D Circular Interpolation (G02.4 and G03.4) codes are part of the tool positioning code group. These codes require two lines of NC code:

- The first line contains a set of X, Y, and Z values which represent the Intermediate Point.
- The second line contains a set of X, Y, and Z values which represent the End Point.

The Radius, Direction (CW or CCW), and Center Point are calculated based on the current location, the Intermediate Point, and the End Point. G02.4 and G03.4 can be used interchangeably to represent the same arc. The actual direction is calculated by the software.

⇨ Both G02.4 and G03.4 codes are canceled by G00, G01, the canned cycle commands (G73, G76, G81-G89), or by each other.

The programmed feedrate can be changed by adding an F word to any NC block when this code is active.

G17, G18, G19 are irrelevant for these G codes.

G41 and G42 may not be used with these G codes.

Example

Below is a program example using G03.4:

```
%
T1 M6 S500 M3
G0 X0 Y0 Z6
G1 X0 Y0.0 F5.
Z0
G3.4 X5.0 Y2.5 Z1.0           (Intermediate Point)
X10.0 Y0.0 Z0.0               (End Point)
G0Z6
M2
E
```

Dwell Mode (G04)

The Dwell Mode code (G04) causes the machine to delay the shift to the next block in the program by the amount specified by parameter P or X for a specified amount of time. When an integer is used with the G04 command, the value is multiplied by 0.01 for BNC and.001 or .0001 for ISNC depending on the Least Dwell Units programmed on the NC Parameters screen.

The BNC format for the dwell time is as follows:

Real Number:.3 second = G04 X.3 or G04 P.3

Integer Number:.3 second = G04 X30 or G04 P30

This is the ISNC format for the dwell time programmed with a real number:

Real Number:.3 second = G04 X.3 or G04 P.3

When .001 is programmed for the Least Dwell Units field on the NC Parameters screen, the ISNC format for the dwell time programmed with an integer is this:

Integer Number: .3 second = G04 X300 or G04 P300

When .0001 is programmed for the Least Dwell Units field on the NC Parameters screen, the ISNC format for the dwell time programmed with an integer is this:

Integer Number:.3 second = G04 X3000 or G04 P3000

The Dwell Mode code is only active in the programmed block, but the dwell time is modal and it affects most of the canned cycles.

⇨ A decimal after an integer indicates whole seconds. For example, **G04 P5** = .005 second dwell, while **G04 P5.** = 5 second dwell.

Format

G04 P_____ or X_____

The P or X parameters specify seconds. The range of values is 0.001–9999.999 seconds.

Surface Finish (G05.1)

The code determines the type of finish quality, Pn. When n = 1, the Surface Finish Quality is Precision. When n = 2 the Surface Finish Quality is Standard. When n = 3 the Surface Finish Quality is Performance. The parameter Qm sets the chord segment for the finish, where m is the acceptable error value.

Data Smoothing (G05.2)

This code has two components.

Pn enables or disables the process. When n = 0, NC Block Smoothing is Off. When n = 1, NC Block Smoothing is On.

Qm sets the Tool Path Tolerance. The deviation from the tool path the system will tolerate, m, must be between 0 to 0.0005 inches, inclusive (0 to 0.012 mm).

Surface Finish Quality (G05.3)

The G05.3 command is used with the Select Surface option.

G05.3 P_, where the P value is 1.0 to 100.0. P1 gives a smoother surface but requires a longer cutting time. P100.0 cuts down time to cut the part, but gives a rougher surface finish.

Cylindrical Rotary Wrap On (G07.2) (preliminary)

The G07.2 code wraps X-, Y-, and Z-axis commanded motion to a cylinder for 4- and 5-axis machines.

There are two methods for setting the cylinder location and orientation: Rotary Axis Method and Vector Method. Using default parameters for both styles, the X-direction is wrapped around the cylinder, the Y-direction points along the cylinder axis, and the Z-direction always maps to the cylinder radius.

The optional H_ parameter is the rotation in degrees about the Z-direction for the 3D feature that will be wrapped (prior to being wrapped). 0 degrees is the default if no H parameter is specified, this will align the Y-direction to the cylinder axis and the X-direction to be wrapped around the cylinder.

Format

G07.2 requires parameters to identify the location and the orientation of the cylinder:

Rotary Axis Method

The Rotary Axis Method wraps XYZ to the cylinder.

G07.2 X_Y_Z_ C_ D_P_L_Q_

- X_Y_Z_ is cylinder coordinate system location with respect to the current coordinate system. If not specified, the control sets them to zero.
- C_ indicates cylinder axis is along C-axis and the # indicates the mapping radius. The radius cannot be zero. If the radius is negative, it will wrap to the inside of the cylinder along the negative radial direction (180 degrees rotation). A_ or B_ can be specified instead of C to define the rotary axis the cylinder is parallel to.
- D_ (optional) Indicates inside cylinder wrap. When specified, it will invert the wrap at the same angle (no 180 degree rotation as when a negative radius is specified).
- P_ (optional) Indicates lead/lag distance (tangential to positive rotation direction at contact point on cylinder).
- L = (optional) Incremental distance from first position after G07.2 to the Retract Plane (same as for G08.2). If not specified, control uses L=infinity.
- Q = (optional) Incremental distance from first position after G07.2 to the Check Plane (same as for G08.2). If not specified, control uses Q=0 (Check Plane passes through the Target Point).

Vector Method

The Vector Method uses vectors (IJK and UVW) to define the zero-angle direction and the cylinder axis direction.

G07.2 X_Y_Z_I_J_K_U_V_W_R_D_P_L_Q_

- X_Y_Z_ is the cylinder coordinate system location with respect to the current coordinate system. If not specified, the control sets them to zero.

- I_J_K_ is the cylinder Zero Rotation direction with respect to the current coordinate system.
- U_V_W_ is the cylinder axis direction with respect to the current coordinate system.
- R is the mapping radius. The radius cannot be zero. If the radius is negative, it will wrap to the inside of the cylinder along the negative radial direction (180 degrees rotation).
- D_ (optional) indicates inside cylinder wrap. When specified, it will invert the wrap at the same angle (no 180 degree rotation as when a negative radius is specified).
- P_ (optional) Indicates lead/lag distance (tangential to positive rotation direction at contact point on cylinder).
- L = (optional) Incremental distance from first position after G07.2 to the Retract Plane (same as for G08.2). If not specified, control uses L=infinity.
- Q = (optional) Incremental distance from first position after G07.2 to the Check Plane (same as for G08.2). If not specified, control uses Q=0 (Check Plane passes through the Target Point).

Cylindrical Rotary Wrap Off (G07.3) (preliminary)

This command cancels the G07.2 command.

Automatic Safe Repositioning Command Buffer On (G08.1) (preliminary)

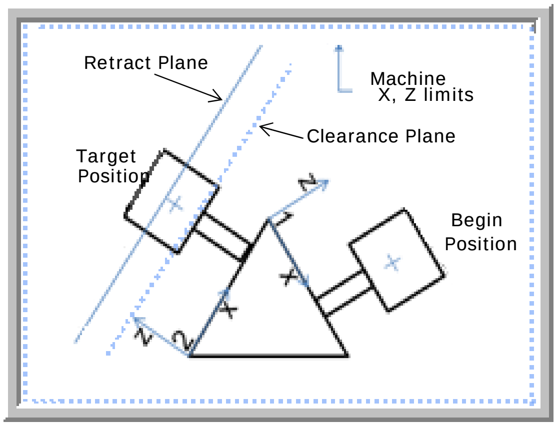
The G08.1 code turns on the Automatic Safe Repositioning (ASR) Command Buffering. This feature allows NC programs to automatically retract, reorient, and plunge from anywhere in the machine to any point in the program using a series of moves computed automatically that do not violate machine limits. When G08.1 is called, the control will stop outputting machine motion and commands. Internally the control will continue to process the program and buffer up all machine commands. Each subsequent block is processed and the Target Position is updated with each new programmed move.

Example

The following example outlines the basics of Automatic Safe Repositioning (ASR). This example is for an SR machine; the objective is to safely move the tool from one Transform Plane to another, both at different locations and orientations.

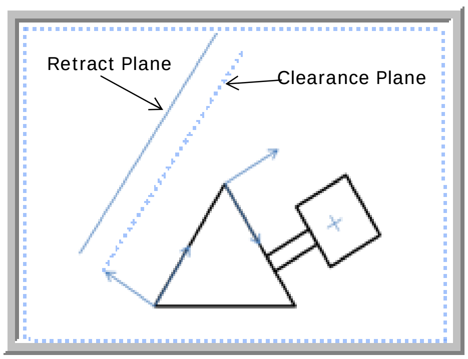
⇨ While the control will try to move the tool tip to the retract plane, the tool tip must at least reach the clearance plane, or the control will terminate program execution and generate an error message.

The following graphic shows the tool in the begin and target positions, as well as the retract plane, clearance plane, and machine limits:

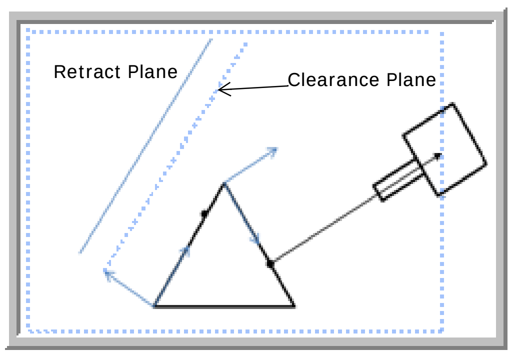


The tool starts in the begin position (1) and retracts along the tool vector (direction) to the machine limits, M140 (2):

1

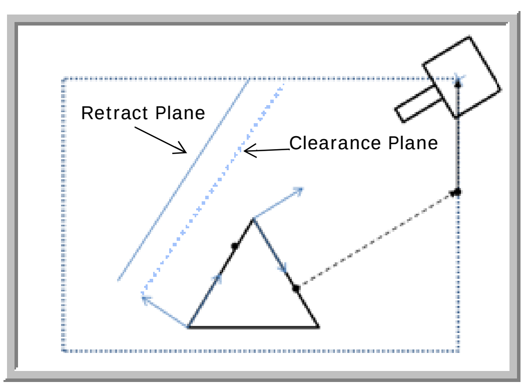


2

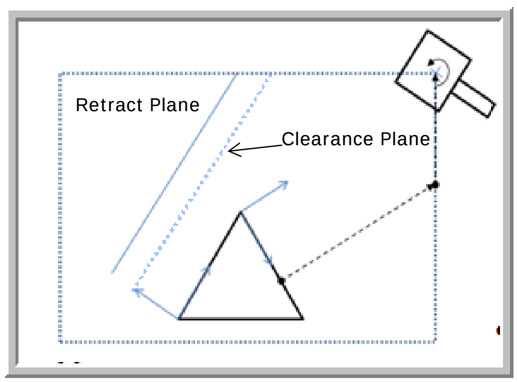


The tool retracts to the vertical (Z) machine limit (3) and then re-orients to the target position orientation (4):

3

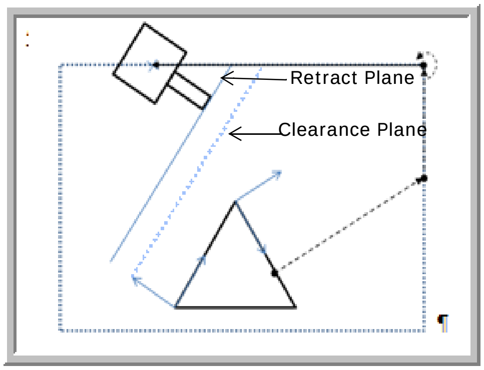


4

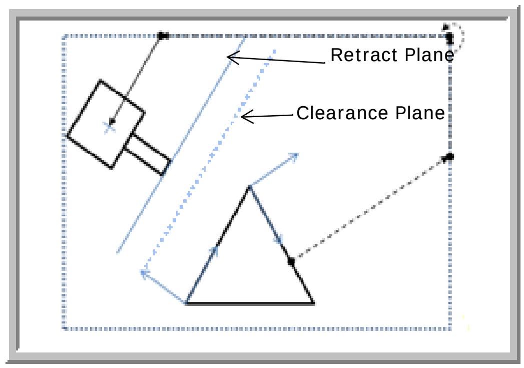


The tool tip moves to the retract plane (5) and then moves to a point above the plunge position in XY of the retract plane (6):

5



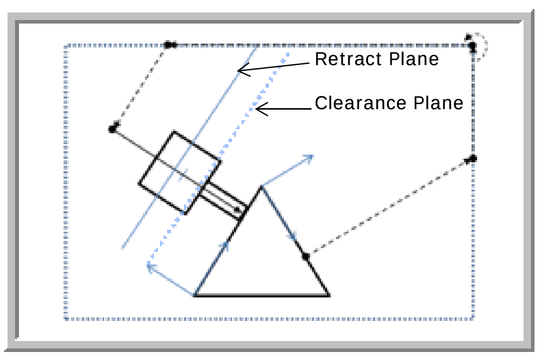
6



⇨ If the tool tip is not at or above the clearance plane, program execution is stopped and an error message is generated.

The tool plunges to the target position along the plunge direction (7):

7



Automatic Safe Repositioning Command Buffer Off (G08.2) (preliminary)

The G08.2 code turns off ASR Command Buffering. When the G08.2 command is called, the buffer state is turned off and the control outputs any buffered up machine commands (for example, preparatory functions, tool change, and so on).

⇨ The control will not output redundant commands after the ASR Command Buffer state is turned off. For example, if a rotary clamp was on before G08.1 and the clamp was toggled off and on several times, then finally set to off, G08.2 will only output one unclamp command.

The target position is the last machine position computed while the buffering was on (prior to G08.2 call). The control will perform the ASR move sequence to restore that position.

Format

G08.2 has several optional parameters, listed below.

Retract Plane location

- No Parameter {DEFAULT}. Retract Plane is infinite distance from Target Point along Retract Direction (sets L = infinity).
- L_ = Retract Plane is located an incremental distance from the Target Point along the Retract Vector (direction).
- X_Y_Z_ = Retract Plane passes through this point defined with respect to the current coordinate system. This parameter is not permitted if G08.2 is called while G07.2 is active (cylindrical wrap).
- L_ and {X_Y_Z_} cannot be combined in the same block.

Retract Direction

- No Parameter {DEFAULT}. Retract direction is along Target Position's tool vector (direction).
- A_B_C_ = Retract Direction vector with respect to the current coordinate system.

Clearance Plane

- No Parameter {DEFAULT}. Check Plane passes through Target Position (sets Q = 0).
- Q_ = Clearance Plane is located an incremental distance from the Target Point along the inverse of the Plunge Direction.
- I_J_K_= Clearance Plane passes through this point defined with respect to the current coordinate system. This parameter is not permitted if G08.2 is called while G07.2 is active (cylindrical wrap).
- Q_ and {I_J_K_} cannot be combined in the same block.

Plunge Direction

- No Parameter {DEFAULT} Plunge Direction is along Target Position's tool vector (direction).
- U_V_W_ = Plunge Direction vector with respect to the current coordinate system.

Retract and Clearance Plane Normal

- No Parameter {DEFAULT} Forces Plane Normal to be inverse of Plunge Direction.
- R = 1. Plane Normal is set along the current coordinate system's Z-direction.

  ⇨ Note that the clearance plane and retract plane always share normal directions.

Calling the G08.2 Buffer Off command with no parameters sets all the following defaults:

- Retract Direction is along the current Tool Vector (direction) in the current coordinate system.
- Plunge Direction is the inverse of the current Tool Vector (direction).
- Plane Normal is along the inverse of the Plunge Direction vector (along Tool Vector).
- L=infinity (Retract Plane is an infinite distance from the Target Point along the Plane Normal).
- Q=0. (Clearance Plane is passes through the Target Point).

G08.1, G08.2 Programming Example

`%`	Program Start
`N1 G08.1`	Buffer ON
`N2 T1 M6`	Move tool #1 into spindle
`N3 G68.2X50. Y-50. Z0.I0.J0.K-1.U0.V1.W0.`	Call Transform Plane coordinate system: X-direction vector = {0,0,-1} Y-direction vector = {0,1,0}
`N4 S3000M3M8`	Spindle speed 3000 rpm, Spindle rotate clockwise on, coolant on
`N5 G0X0.Y0.Z5.I0.J0.K1.`	Orient tool vector to {0,0,1} direction and rapid move to {0,0,5} with respect to Transform Plane

`N6 G08.2 X0.Y0.Z100.A0.B0.C1.\`
`I0.J0.K50.U0.V0.W-1.\`
`R1.`
	Buffer Off command (see Explanation 1 below)

`N7 G1Z-5.F500.`	Plunge at slow feedrate

`N8 G1X100.F1000`
`N9 Y100.`
`N10 X0.`
`N11 Y0.`
	Mill a 100 mm x100 mm square on RHS of block at feedrate of 1000 mm/min.

`N12 M5M9`	Spindle off, Coolant off
`N13 G69`	Transform Plane Off
`N14 G08.1`	Buffer ON
`N15 G68.2X-50. Y-50. Z-100. I0.J0.K1.U0.V1.W0.`	X-direction vector = {0,0,1} Y-direction vector = {0,1,0}
`N16 G0X0.Y0.Z5. I0.J0.K1.`	Orient tool vector to {0,0,1} direction and rapid move to {0,0,5} with respect to Transform Plane

N17 M3M8	Spindle rotate clockwise on (at 3000 rpm), coolant on
N18 G08.2 L95. Q45.	Buffer Off command (see Explanation 2 below)
N19 G1Z-5.F500.	Plunge at slow feedrate
N20 G1X100.F1000 N21 Y100. N22 X0. N23 Y0.	Mill a 100 mm x 100 mm square on RHS of block at feedrate of 1000 mm/min.
N24 M5M9	Spindle off, Coolant off
N25 G0M140	Retract tool from cut to machine limits
N26 G69	Transform Plane Off
N27 M02	Program Stop
N28 E	Program End

Explanation 1:

```
G08.2 X0.Y0.Z100.A0.B0.C1.\
      I0.J0.K50.U0.V0.W-1.\
      R1.
```

This block fully specifies the parameter for the Buffer Off command as follows:

- X0.Y0.Z100. = Point in the Retract Plane with respect to the current coordinate system, which happens to be a Transform Plane.
- A0.B0.C1. = Retract Direction vector with respect to the current coordinate system.
- I0.J0.K50. = Point in the Check Plane with respect to the current coordinate system.
- U0.V0.W-1. = Plunge Direction vector with respect to the current coordinate system.
- R1 = Sets Plane Normal vector to be along the current coordinate system's Z-direction.

Explanation 2:

G08.2 L95. Q45.

This block specifies the following using default parameters:

- Retract Direction is along the current Tool Vector in the current coordinate system {0,0,1}.
- Plunge Direction is the inverse of the current Tool Vector {0,0,-1}.
- Plane Normal is along the inverse of the Plunge Direction vector {0,0,1}.
- L95. = Specifies the incremental distance (95mm) from the Target Point

{0,0,5} along the Plane Normal {0,0,1} that the Retract Plane passes through. The computed point is {0,0,100}.

- Q45 = Specifies the incremental distance (45mm) from the Target Point {0,0,5} along the Plane Normal {0,0,1} that the Check Plane passes through. The computed point is {0,0,50}.

Precision Cornering (G09)

The Precision Cornering code (G09) decelerates the axes to zero velocity at the end of the block in which it is programmed. After stopping, the axes accelerate to the programmed feedrate in the next block. This causes a sharp corner to be cut regardless of the programmed feedrate. The G09 code is not part of a G code group, so it affects only the axis movement of the block in which it is specified.

Setting Work Coordinate Systems with G10

This code is used for setting tool offsets, entering tool wear data, and changing work coordinate systems. The G11 command cancels the data setting mode.

⚠ G10 commands can not be performed by graphics at the same time that the program is running.

Setting External Work Zero Offsets (G10 with L2)

One of six work coordinate systems can be changed as shown below where P is used to select the external work zero point offset value (P parameter = 0), or one of the work coordinate systems (P parameter = 1 to 6), and X, Y, Z, A, B is the work zero point offset value of each axis. G90 specifies absolute dimensioning and G91 specifies incremental dimensioning.

Format

The command format for setting external work zero offsets is as follows:

G10 L2 P____ X____Y____Z____ A ____ B ____

Setting Tool Offsets with G10

This code is used for setting tool length and radius offsets. The G11 command cancels the data setting mode.

Initializing Tool Length Offsets (G10 with P, R)

G10 is used with the P and R parameters. P is the offset number 01 through 200, and R is the offset amount which may be absolute or incremental depending on G90 or G91.

Format

This is the command format for initializing tool length offsets:

G10 P____ R____

Initializing Tool Offsets (G10 with T, H, D)

G10 is used with the T, H and D parameters. The T parameter is the offset number 01 through 200, H is the tool length offset and D is the tool radius offset.

Format

This is the command format for initializing tool offsets:

G10 T___ H___ D____

Assigning Tool Offsets (G10 with L3)

The following example shows how to assign tool offsets. T is the tool number, H is the tool length offset number, and D is the cutter compensation (offset) number.

Format

The command format for assigning tool offsets is as follows:

G10 L3 T_____H_____D_____

Example

The following tool offset initialization example shows how to set up a program to assign offsets to the tools.

G10L3

T0001 H_____ D_____ (for tool 1)

T000n H_____ D_____(for tool n)

G11 (to cancel)

⇨ The tool number does not require leading zeroes.

Polar Coordinates Command (G16)

This command allows coordinates in the current block to be input in polar coordinates (radius and angle). The first coordinate in the currently selected plane is the radius coordinate in mm, and the second coordinate in the plane is the angle in degrees. For the XY plane, the X value represents radius and the Y value represents the angle.

G16 is canceled by G15 (Polar Coordinates Cancel).

Format

The command format for Polar Coordinates is as follows:

G16 X____ Y_____ Z_____

Example

Select Metric mode for the following sample program using the Polar Coordinates command:

NC Part Program 1 Metric PIE.FNC

```
%
T1 M06
M03 G00 G90 X0 Y0 Z0 S1800
G01 Z-.25 F20.
G01 G16 X50. Y60.
G03 X50. Y120. R50.
G15
G01 X0 Y0
M02
```

Plane Selection

The three codes in the plane selection group and their relationships to each other are illustrated below:

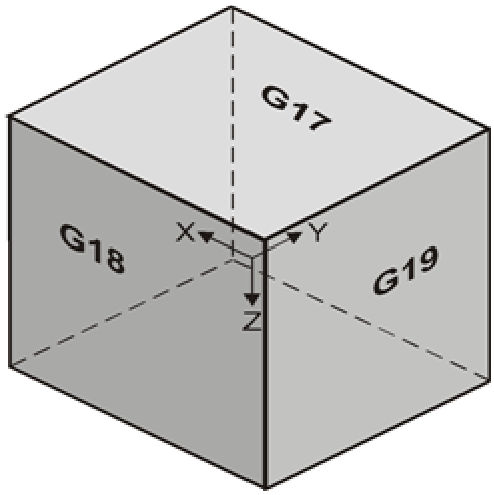


Figure 2–4. Plane Selection Group Codes

XY Plane Selection (G17)

The XY Plane Selection code (G17) is the default and sets the plane for circular interpolation modes G02 and G03 to the XY plane. The X, Y, Z, I, and J words are valid in circular interpolation blocks; K words are invalid. If a Z word is programmed in the circular interpolation block, a helix is generated in the XY plane. The direction of an arc or helix in the XY plane can be determined by looking at the XY plane with positive X to the right and positive Y going up. The XY plane is a right-handed coordinate system (thumb points to positive Z, and fingers wrap in counterclockwise direction).

In G17, the arc end point is defined by the X and Y words in the block. The arc center point is defined by the I and J words in the block.

G17 is canceled by G18 and G19.

Format

The format of the XY plane selection command is as follows:

G17 X____ Y____

Example

The diagram below illustrates XY plane selection:

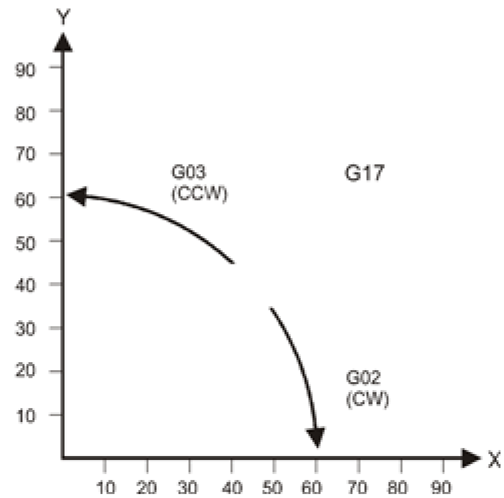


Figure 2–5. XY Plane Selection (G17)

XZ Plane Selection (G18)

The XZ Plane Selection code (G18) sets the plane for the circular interpolation codes G02 and G03 to the XZ plane. The X, Y, Z, I, and K words are valid in circular interpolation blocks; J words are invalid. If a Y word is programmed in the circular interpolation block, a helix is generated in the XZ plane. The direction of an arc or helix in the XZ plane can be determined by looking at the XZ plane with positive X to the right and positive Z going up.

Basic NC and ISNC handle the XZ plane differently. For Basic NC, the XZ plane is a left-handed coordinate system (thumb points to positive Y, and fingers wrap in clockwise direction). For ISNC, the XZ plane is a right-handed coordinate system (thumb points to positive Y, and fingers wrap in a counterclockwise direction).

In G18, the arc end point is defined by the X and Z words in the block. The arc center point is defined by the I and K words in the block.

G18 is canceled by G17 and G19.

Format

The format of the XZ plane selection command is as follows:

G18 Z____ X____

Example

The diagrams below illustrate XZ plane selection in Basic NC and in ISNC:

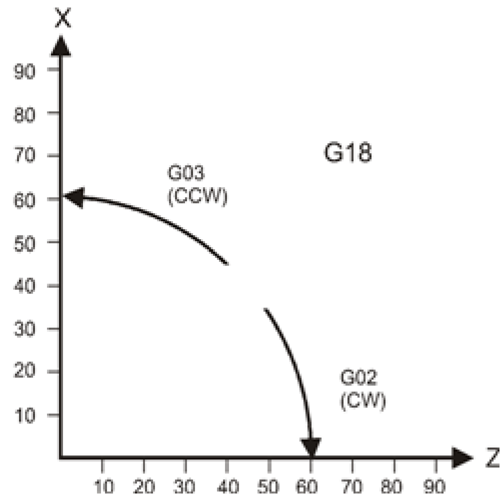


Figure 2–6. Basic NC XZ Plane Selection (G18)

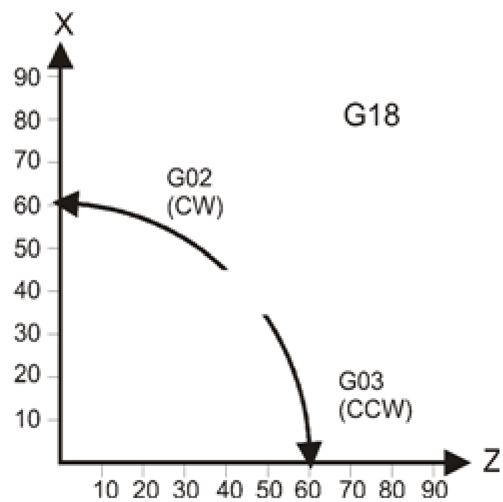


Figure 2–7. ISNC NC XZ Plane Selection (G18)

YZ Plane Selection (G19)

The YZ Plane Selection code (G19) sets the plane for circular interpolation codes G02 and G03 to the YZ plane. The X, Y, Z, J, and K words are valid in circular interpolation blocks; I words are invalid. If an X word is programmed in the circular interpolation block, a helix is generated in the YZ plane. The direction of an arc or helix in the YZ plane can be determined by looking at the YZ plane with positive Y to the right and positive Z going up. The YZ plane is a right-handed coordinate system (thumb points to positive X, and fingers wrap in counterclockwise direction).

In G19, the arc end point is defined by the Y and Z words in the block. The arc center point is defined by the J and K words in the block.

G19 is canceled by G17 and G18.

Format

The format of the YZ plane selection command is as follows:

G19 Y____ Z____

Example

The diagram below illustrates YZ plane selection:

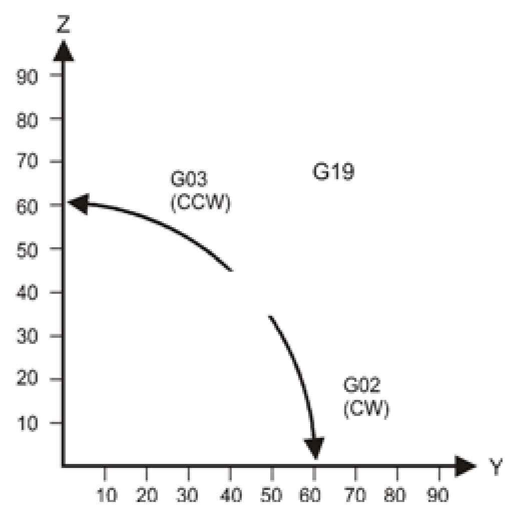


Figure 2–8. YZ Plane Selection (G19)

Units of Measure ISNC G20, G21

Before setting the coordinate system at the beginning of the program, the units of measure must be specified in an independent block. A part program may switch between English and Metric modes as long as the format of the dimensions is correct for the chosen mode.

The Imperial Units of Measure code (ISNC G20) signals the system that the dimensions are in inches.

ISNC G20 is canceled by G21.

The Metric Units of Measure code (ISNC G21) signals the system that the dimensions are metric units.

ISNC G21 is canceled by G20.

Format

These are the command formats for the inch/metric conversion commands:

ISNC:

G20: Inch command

G21: Metric command

⇨ The ISNC G20 and G21 codes do not affect the units of measure used in the graphics and machine status display screens. The displays are controlled by the units selected when entering NC editing.

Automatic Return To and From Reference Point (G28 and G29)

Any point within the machine coordinate system can be selected as the reference point. The return to reference point is often used to move the part forward so you can remove chips from the part and inspect the part. Select the reference point on the NC Parameters screen.

The Automatic Return To Reference Point command (G28) specifies an automatic return to the reference point for the designated axes. An intermediate point can be specified with the X____Y____Z____ parameters. If no intermediate point coordinates are specified, the system uses the previous intermediate point coordinates. If no intermediate point coordinates are specified during the current program execution, the machine returns directly to the reference point.

The Automatic Return From Reference Point command (G29) specifies an automatic return from the reference point through the intermediate point, if specified by a previous G28, and to the end point designated by the X, Y, and Z parameters. If no intermediate point coordinates were specified during the current program execution, the machine will return directly from the reference point to the specified end point.

G28 Format

The format for the automatic return to reference point command is:

G28 X____ or Y____ or Z____

These parameters specify the absolute or incremental location of the intermediate point in coordinates relative to the current coordinate system. The G28 command is only performed for the axes which follow the G28. For example, if an X value follows the G28, the machine moves to the X reference point, not the Y or Z reference point.

As another example, a typical method to home the Z axis using incremental mode is shown below. The combination of G91 and a value of zero for the Z axis causes the Z axis to move directly to the home (reference) point without moving through an intermediate point.

```
G91 G28 Z0
G90
.
.
.
```

G29 Format

The format for the Automatic Return to Reference Point command is:

G29 X____ Y____ Z____

When the G29 command is given, the system returns to the most recently used working coordinate system. These parameters specify the absolute or incremental location of the end point in coordinates relative to the current coordinate system in effect when the G28 command was processed.

Example

This sample program uses G28 and G29 to return the spindle to and from the reference point. Set Part Zero to X12 Y9 before running the program.

NC Part Program 1 Inch
PLAIN_28.FNC

```
%
G00 X0 Y0 Z0
G28 X-7 Y-8
G29 X3 Y-4
M02
```

Skip (Probing) Function (G31)

The Skip Function command is used to perform probing within an NC program, allowing you to specify a target position. The machine axis will stop when the target position is reached, or if the probe comes in contact with another surface. The Skip Function command can be programmed on a PC, but it can only be run properly on the control. On a PC, the command works similarly to the Linear Interpolation (Cutting Feed) (G01) command. The Skip Function command is a one-shot command and is only effective in the current block.

Two-touch and single-touch probing is supported. These modes are selected with the M41 and M42 codes. Two-touch probing is the default probing mode.

When performing two-touch probing (M42 specified), the probe moves in the specified direction until it touches the part, backs up away from the part, and then moves forward again at a feedrate of F/10. When the probe touches the part again, the trigger point is stored in variable #5061 (X axis), #5062 (Y axis), or #5063 (Z axis) with the NCPP option. The NCPP option allows you to create macro subroutines and use conditional statements and math functions. (Refer to *NC Productivity Package Option, on page 6 - 1* for more information about the variables and subprograms.)

When performing one-touch probing (M41 specified), the probe does not back up after the first touch.

Values may be written to tool offset variables so they can be viewed after running the program on the Tool Offset screen. If the system does not have the NCPP option, the values need to be recorded manually. A Programmed Stop (M00) command can follow the G31 block to stop the machine so you can record the machine's location.

The values which are stored are referenced to the current coordinate system (working, local, or machine). If the probe does not touch the part before the end of the movement, the end coordinate value is stored in #5061, #5062, or #5063.

The current positions of the XY and Z axes can be retrieved using the #5041, #5042, and #5043 registers. The values can then be stored into part setup using the G10 code. For example, to set the X value for work offset G54, use the following G10 command:
G10 L2 P1 X[#5041]

Format

The format for the Skip (Probing) Function is as follows:

G31 X____ Y____ Z____ and/or F____

⇨ This command cannot be performed with cutter compensation (G41, G42, G43 {with G18 or G19}, G45, G46, G47, and G48).

Example

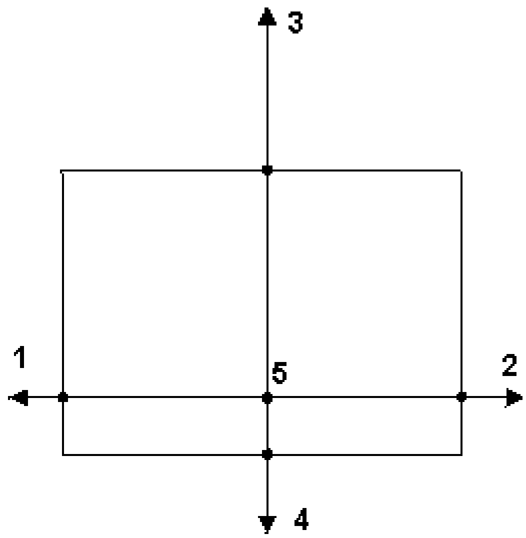
The following program example finds the center point within a box when run on the control.

NC Part Program 1 Inch
G31_TEXT.FNC

```
%
(GO TO INITIAL PART ZERO)
G01 X0 Y0 F15.
G31 X7 F15.
#2001 = #5061
G01 X0 Y0 F25.
G31 X-7 F15.
#2002 = #5061
#2003 = [#2002+#2001]/Z
G01X#2005Y#2006F25.
N100 M00
G31 Y5 F15.
#2004 = #5062
G01 X#2003 Y0F25.
G31 Y-5 F15.
#2005 = #5062
#2006 = [#2004+#2005]/Z
(THE SPINDLE NOW MOVES TO THE CENTER OF THE BOX)
G01 X#2003 Y#2006
```

Parallel sides are assumed to be aligned with the X and Y axes. Additional programming steps will be required to determine the angle between the sides and the X and Y axes (skew angle) if the sides are not aligned. The initial part zero is set somewhere within the box.

The probe moves in the +X and then the -X direction to determine the center point between the sides in the X axis.



1	X Left
2	X Right
3	Y Top
4	Y Bottom
5	Initial Part Zero

Figure 2–9. ISNC Skip (Probing) Function

Tool Offsets (G40–G49)

Tool Offsets include Cutter Compensation and the Tool Length Offsets and Tool Radius Offsets. Cutter Compensation G codes G40–G42 are used to control tool movement. Access the Tool Offsets from the Tool Setup screen. Select the Tool Offsets (F4) softkey. From the Tool Offsets screen, select either Tool Length (F3) or Tool Radius (F3). The softkey toggles between these selections.

- The Tool Length Offset Table contains the tool length offset (G43, G44).
- The Tool Radius Offset Table holds signed values for cutter compensation (G40–G42) and diameter compensation (G43–G48).

The measurement units for the offsets in the Tool Offset Table depend on the programmed units. If -9.5 is entered for tool offset 15, that tool offset is -9.5 inches (or -9.5 mm, depending on the unit of measurement).

The **Toggle Units softkey** on the NC Tool Offsets screen changes the units of measurement between inch and metric.

Use either the **Page Up** and **Page Down** softkeys, the PAGE UP and PAGE DOWN keys, or the scroll bar to scroll through all of the Tool Offset fields. The keyboard up and down arrows move through the fields displayed on the screen.

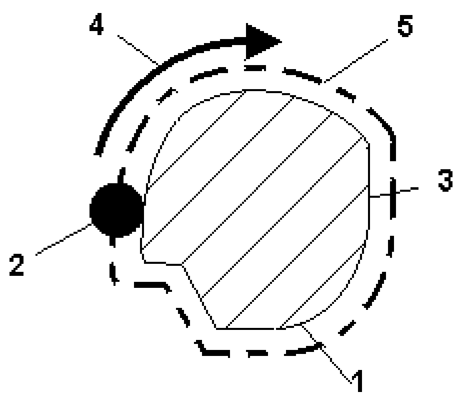
Cutter Compensation (G40–G42)

Cutter compensation may be used for two purposes. First, it may be used when the dimensions in the program and the part surface are the same. The system calculates the proper tool path by using the part surface and the tool diameter information.

Second, cutter compensation corrects the difference between the diameters of the tool specified and the tool actually used to cut the part. This situation often occurs when the program originates from an off-line device. Note that the coordinates of those programs are usually tool center line data.

⇨ If the first segment in a contour is an arc with a radius smaller than the radius of the tool, the control will generate an error message indicating that you need to use a tool with a smaller radius or program a larger radius.

Cutter compensation is based on the direction of travel of the tool. To determine which type of cutter compensation to use, look at the part as if you are moving around the part always keeping the tool ahead of you. Then it becomes obvious whether the tool needs to be on the right or the left of the programmed line or the boundary of the part as shown in the illustration below.



1	Part
2	Cutter
3	Programmed Part Outline
4	Direction of Cutter Travel
5	Compensated Cutter Path

Figure 2–10. Cutter Compensation

Cutter Compensation – ISNC and Basic NC Programming Differences

You may program cutter compensation using ISNC or Basic NC. There are differences between the two. For example, programming cutter compensation using ISNC requires that you use D codes. This is not the case with Basic NC. Other differences include how ISNC and Basic NC interpret D code values.

Tool Radius Offset

ISNC

To program cutter compensation using ISNC, you must use a D code. The D code specifies an index into the Tool Offset Table or an actual offset value. For example, in the command G41 D5, the index value is an actual offset value of 5.

An error will be generated if there is a decimal point.

⇨ You may access the Tool Radius Offset Table from the Tool Length Offset screen. The Tool Radius Offset page contains 200 registers for storing radius offsets.

Basic NC

To program cutter compensation using Basic NC, you may choose whether or not to use a D code. If you do not use a D code, Basic NC will use the value in the Diameter Comp field from the Tool Setup screen.

If you use a D code, Basic NC interprets the D code value based on whether you are calling out a tool change or commanding a G41 or G42.

If you use a D code when calling out a tool change, enter the actual tool diameter. Basic NC divides this value by two to calculate the tool diameter offset.

If you are programming a G41 or G42 code, Basic NC interprets the D code based on whether the D code value contains a decimal point:

- Contains a decimal point – Basic NC interprets the D value as the tool diameter offset
- Does not contain a decimal point – Basic reads the D value as an index value for the Tool Offset table

Tool Length Offset

You may use a G43 code to program tool length offsets. This is true for ISNC and Basic NC. Use an H code to specify an index into the Tool Offset Table. For example, in the command G43 H01, the index value is “01”.

The value in the Tool Offset Table is a negative value that represents the distance from the Z home position to the top of the part with the tool tip touching the top of the part.

Another way to program a tool length offset would be to use the Zero Calibration field in the Tool Setup screen, and not use the G43 H code. This is recommended, especially if you are using the Tool Probing software.

Cutter Compensation Off (G40)

The Cutter Compensation Off code (G40) is the default. It cancels cutter compensation by erasing all the data in the system’s cutter compensation look-ahead buffers and moving to the current uncompensated endpoint at the programmed feedrate.

G00 or G01 must be selected in order for this command to cancel the offset compensation. Each axis moves straight (G01) or at rapid traverse (G00) from the point of the old vector at the start point toward the end point. The machine should be in G40 mode before the end of a program. Otherwise, when the program ends in the offset mode, positioning cannot be made to the terminal point of the program, and the tool position will be separated from the terminal position by the vector value.

Format

The command format for Cutter Compensation Off is as follows:

G40 X____ Y____

If the parameters are omitted, the tool moves the old vector amount in the opposite direction which effectively cancels the offset.

It is possible to switch from left to right cutter compensation (and vice versa) without first canceling cutter compensation.

Cutter Compensation Left (G41)

The Cutter Compensation Left code (G41) switches on cutter compensation. It establishes a new tool path left and parallel to the programmed path. The distance between the new tool path and the programmed path is equal to the cutter compensation value for the programmed tool.

G41 is canceled by G40.

The offset executes only in the G17 offset plane. In simultaneous three-axis control, the tool path projected on the offset plane is compensated.

G00, G01, G02, or G03 must be specified.

Format

The command format for cutter compensation left is as follows:

G41 X _____ Y_____ D_____

If the offset number for cutter compensation is D00, the system will not go into G41 mode.

3D Tool Geometry Compensation (G41.2)

⇨ 3D Tool Geometry Compensation (G41.2) is functional only with the VMX42 SR machine.

3D Tool Geometry Compensation (G41.2) allows the specification of the Surface Contact Point, Surface Normal Vector, and Tool Vector for ball nose, flat end, and bull nose endmills. See *3D Tool Geometry Compensation (G41.2), on page 2 - 47* discussion in Tool Centerpoint Management (M128) for more information.

Cutter Compensation Right (G42)

The Cutter Compensation Right code (G42) switches on cutter compensation and establishes a new tool path right of and parallel to the programmed path. The distance between the new tool path and the programmed path is equal to the cutter compensation value for the programmed tool.

G42 is canceled by G40.

This command is an offset method similar to G41 except that the offset is to the right of the programmed path looking in the direction in which the tool is advancing. The offset is performed only in the G17 offset plane. Only the coordinate values of an axis in the offset plane are affected by the offset. In simultaneous three-axis control, the tool path projected on the offset plane is compensated.

G00, G01, G02, or G03 must be specified.

Format

The command format for cutter compensation right is as follows:

G42 X _____ Y_____ D_____

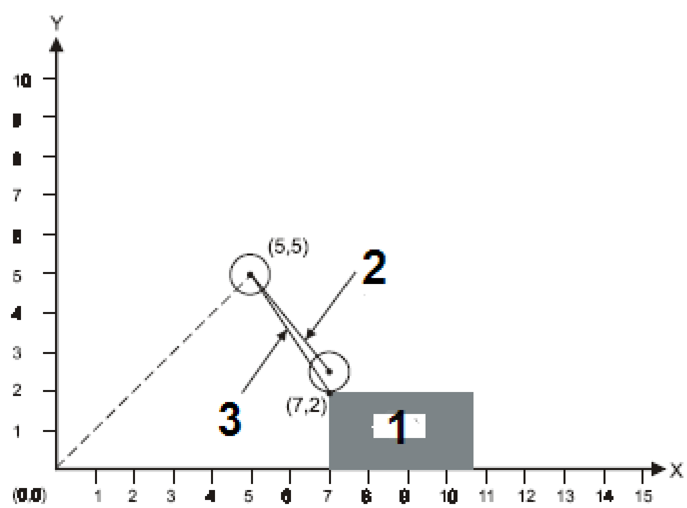
If the offset number is D00, the system will not go into G42 mode.

Cutter Compensation Programming

Follow these steps to use cutter compensation:

1. Enter the part surface description according to the final dimensions of the part.
2. Enter the full cutter diameter as a positive number in the Diameter Compensation field in Tool Setup, or supply a D word when changing tools (Basic NC only).
3. Activate cutter compensation in the desired direction (left or right of part surface with respect to tool path direction).
4. Supply an entry move from somewhere outside the part to the start point of the part surface, i.e., somewhere outside of the compensated path. The part surface appears as a blue line on the graphics display.
5. Following all of the blocks to be compensated, provide an *exit move* to somewhere outside the compensated path and turn off cutter compensation (G40). When a G40 is programmed, cutter compensation extracts any remaining information from its look-ahead buffer and moves to the last programmed end point. The tool moves from the compensated end point of the previous move to the end point of the exit move.
6. Be certain that the exit move is outside the compensated path. Otherwise, turning off cutter compensation may cut into the part surface. To check the exit move, use graphics to verify the tool path movements.

The tool moves from the start point of the entry move and ends at the compensated start point for the part surface as shown in the graph below:



1	Part
2	Actual Tool Path
3	Programmed Move

Figure 2–11. Cutter Compensated Tool Movement

In the previous illustration, the value in the Diameter Compensation field is 1.00, and these codes were used to control tool movement:

- G00 G40 X5. Y5.
- G41 X7. Y2.

Z movements may be used for the entry and exit moves. For example, turn on cutter compensation when moving to a Z Start plane before plunging. Turn off cutter compensation after retracting the tool from the part.

Turn off cutter compensation using a G40 code before ending a program or all programmed blocks may not be cut.

Refer to *NC Parameters, on page 1 - 19* for information about the DEFAULT CUTTER COMP LOOK AHEAD field in the NC Parameters screen.

Tool Length Offset (G43, G44, G49)

Tool offsets, G43 and G44 tool length compensation codes, are used to compensate tool length without altering the NC program. G43 is for positive tool length compensation. G44 is for negative tool length compensation. Either the G49 command, or an H00 command, immediately cancels the offset.

The tool offset specified by a G43 or G44 overrides the tool length offset from the Zero Calibration field on the Tool Setup screen. The Zero Calibration field on the Tool Setup screen is always treated as a negative Z offset. For example, if a value of 3.0 is put in the Zero Calibration field, a Z offset value of -3.0 is stored. If the command G43 H1 is then used where the value -2.2 is stored in the H1 offset register, WinMax uses a tool offset of –2.2. The table below illustrates tool offsets:

Tool Setup Screen Zero Calibration	Tool Length Offset Mode	H1	Total Tool Offset
3.0	G43	-2.2	-2.2
3.0	G44	-2.2	+2.2
3.0	G49	-2.2	-3.0
3.0	G49	+2.2	-3.0
3.0	G43	+2.2	+2.2
3.0	G44	+2.2	-2.2

Table 2–4. Tool Offsets

The values in the Tool Setup screens always remain in the units selected when going into the NC Editor.

For Basic NC (BNC)

An H address may specify an index into the tool length offset table without specifying a G43 or G44. In such a case, the value in the tool length offset table is used as the tool offset. This is equivalent to the Zero Calibration field on the Tool Setup screen.

For BNC and ISNC, if the system is in the G43 and G44 mode already, an H code can be used by itself to replace the existing tool length already in effect.

If no G43 or G44 is programmed and an H offset is not specified, tool lengths are taken from the Tool Setup screen.

An offset in the X axis can be specified with the G19.

An offset in the Y axis can be specified with the G18.

If a G17 is provided, or none of the plane selection commands (G17, G18, and G19) is present, specify an offset in the Z axis.

⇨ The G17, G18, and G19 used in this block are only used to specify the axis of the tool offset and will not affect the specified plane.

An offset in the X or Y axis cannot be specified when cutter compensation is active or commands G45–G48 are being used.

Commands G45–48 support existing X or Y axis tool offset programs; however, to save time, automatic Cutter Compensation (G40–G42) should be used instead.

Commanding an H00 cancels an offset.

Either G43 or G44 is in effect until a G49 or H00 is used.

Format

The H is the Offset Code with a range of H00 to H200. G17 is optional when a Z axis offset is desired.

[G17 _____] or [G18 _____] or [G19 _____] H_____

The following four examples illustrate tool length offset H codes with the G43 and G44 codes. Tool 1 had a value of 5.0 for the Zero Calibration field on the Tool Setup screen. The Tool Length Offset 1 value is –6.0 and the Tool Length Offset 2 value is –7.5.

Example 1

T01 M06

With these offsets, the calibrated tool length will be 5.0. That length is taken from the Tool Setup screen)

Example 2

T01 M06

H02

At this point, the calibrated tool length will still be 5.0 because no G43 or G44 has been entered.

Example 3

T01 L4.0 M06

G43 H01

At this point, the calibrated tool length is 6.0 because the L value was superseded by the offset value of 6.0.

Example 4

T01 M06 - sets the tool length to be 5.0

H02 - tool length remains 5.0

G43 - tool length remains 5.0

H02 - sets the tool length to be 7.5

5-Axis Linear Interpolation (G43.4)

⇨ 5-Axis Linear Interpolation (G43.4) is functional only with the VMX42 SR machine.

The CNC offers two rotary interpolation modes of the tool relative to the Workpiece Coordinate System. Refer to *Getting Started with WinMax Mill*.

- Linearly interpolate the Tool Vector with respect to the Workpiece Coordinate System between tool positions. Refer to *Getting Started with WinMax Mill*.

  Ensure the tool start point is relatively close to the C-axis centerline location immediately before activating the G43.4 command.

  ⇨ When executing a Tilt Axis Preference command, the machine may need to rotate the part up to 180° around the machine singularity point when moving to the contouring start point and tool orientation. Refer to *Tilt Axis Preference (M200), on page 5 - 20*.

  If the tool start point is too far away from the C-axis centerline, an “Axis out of limits” error will occur.

- Linearly interpolate the programmed rotary angle between tool positions.

Format

⇨ The 0 and 1 parameters for G43.4 are optional. If you do not choose a parameter, the control will automatically choose:

- 0 when *Tool Vector Input, on page 5 - 15* is used.
- 1 when *Axes Angle Input, on page 5 - 14* is used.

G43.4 Q{0, 1} will turn on the 5-axis workpiece-relative linear interpolation.

G43.4 Q0 linearly interpolates the Tool Vector and Tool Tip between NC points with respect to the Workpiece Coordinate System. Refer to *Getting Started with WinMax Mill*.

⇨ Although G43.4 Q0 will interpolate the Tool Tip and Tool Vector relative to the Workpiece Coordinate System, the CAM software must generate properly toleranced tool paths to machine smooth surfaces.

G43.4 Q1 linearly interpolates the Tool Tip between tool positions with respect to the Workpiece Coordinate System and the rotary angles in the Machine Coordinate System. Refer to *Getting Started with WinMax Mill*.

Tool Radius Offset (G45–G48)

ISNC and Basic NC – the tool used for cutter compensation must be smaller than or equal to the arc that you have programmed. If the tool radius is greater than or equal to the arc radius, the compensated tool path will sweep in the opposite direction of the programmed arc.

Tool position offset commands increase or decrease the amount of axis movement. Offset values within the following ranges can be selected for the tool radius offset commands:

	mm input	inch input
Offset Value	0~±999.999 mm	0~±99.999 in.
	0~± 999.999°	0~±999.999°

G45–48 support existing X or Y axis tool offset programs. However, to save time, use the automatic Cutter Compensation (G40-G42) instead.

Tool Radius Offset Increase (G45)

This command increases the specified block’s tool radius offset amount by the value stored in the offset value memory.

Tool Radius Offset Decrease (G46)

This command decreases the specified block’s movement amount by the value stored in the offset value memory.

Tool Radius Offset Double Increase (G47)

This command increases the specified block’s movement amount by twice the value stored in the offset value memory.

Tool Radius Offset Double Decrease (G48)

This command decreases the specified block’s movement amount by twice the value stored in the offset value memory.

Format

The command format for the tool position offsets is as follows:

GXX X_____Y_____Z_____A_____B____ D____

GXX is an optional Interpolation (Group 1) move command, and D is the offset command. The number which follows D is an index into the tool offsets table. The offset value is modal and needs to be specified only once. The offset is applied to all axes specified in the parameters.

Example

Set tool offset 1 to the desired offset before running the following program using the Tool Radius Offset commands (G45 through G48):

Industry-Standard NC Part Program 1 Inch
G45_G48.FNC

```
%
N10 G10 P1 R0.5
N20 G00
N30 G90
N40 M25
N50 T1 M06
N60 Z5.0 X0. Y0.
N70 S2000 M03
N80 Z0.05
N90 G00 Z-0.5 F10.
N200(INNER OUTLINE WITHOUT USING OFFSETS)
N210 G91 X4 Y4
N220 G01 X3
N230 Y1.5
N240 X4
N250 G45 Y-1.5
N260 X3
N130 G03 X1 Y1 I0. J1
N140 G01 Y4
N150 X0
N160 G02 X-2 Y2 I0. J2
N170 G01 Y0
N180 X-3
N190 Y-2.5
N200 X-3
N210 Y2.5
N220 X-3
N230 G03 X-1 Y-1 I0. J-1
N240 G01 Y-2
N250 X1
N260 Y-4
N265 Z5.05
N270 G00 X-4 Y-4
N275 G90 X0 Y0
N280(OUTER OUTLINE USING G45, G46, G47, AND G48)
```

NC Part Program 2 Inch
G45_G48.FNC

```
N290 G91 G46 X4 Y4 D1
N300 G47 G01 X3 F20.
N305 Y1.5
N307 G48 X4
N308 Y-1.5
N309 G45 X3
N310 G45 G03 X1 Y1 I0. J1
N320 G45 G01 Y4
N330 G46 X0
N340 G46 G02 X-2 Y2 I0. J2
N350 G45 G01 Y0.
N360 G47 X-3
N370 Y-2.5
N380 G48 X-3
N390 Y2.5
N400 G45 X-3
N410 G45 G03 X-1 Y-1 I0. J-1
N420 G45 G01 Y-2
N430 X1
N440 Y-4
N450 G00 G46 X-4 Y-4
N460 G00 Z5.05
N470 M25
N480 M05
N490 M02
```

Scaling (G50 and G51)

The G51 code is used to scale subsequent move commands by a programmable scale factor and must be in an independent block.

Scaling is not applicable to the following movement in case of canned Z axis movements. If scaling results are rounded and units less than 5 are ignored, the move amount may become zero and may affect cutter movement. Whether the scaling function is effective or not, it can be set by a parameter for each axis. The scaling function always becomes effective for the circular radius command R in the G51 mode, regardless of these parameters.

One or more axes' scaling can be disabled on the NC Parameters screen. The methods for specifying the scaling center point and the scaling factor are different with BNC and ISNC.

For BNC, X, Y, and Z are the scaling center points in absolute coordinates. The I, J, and K codes specify the scale factor for the X, Y, and Z axes. If only I is specified, all axes will be scaled by that factor. Scaling G51 codes may not be nested.

For ISNC, these two methods can be used to specify scaling parameters:

Method 1: X, Y, or Z present. X, Y, Z define the scaling center point. If I, J, or K are present, they define the scaling factors. If they are absent and P is present, the P value defines the scaling factor for all three axes. If P is an integer (no decimal point) the value is multiplied by the least scaling factor parameter on the NC Parameters screen; otherwise, the exact P value is used.

Method 2: X, Y, or Z absent. I, J, K define the scaling center point. P provides the scaling factor if provided for all three axes. If P is an integer (no decimal point) the value is multiplied by the least scaling factor parameter on the NC Parameters screen; otherwise, the exact P value is used.

Format

The format of the BNC scaling code is as follows:

G51 X___ Y___ Z___ I___ J___ K___

The format of the ISNC scaling code is as follows:

Method 1: G51 X___ Y___ Z___ (I____ J_____ K____ or P____)

Method 2: G51 I____ J_____ K____ P____

⇨

The smallest unit of scaling is either 0.001 or 0.00001 when an integer P value is provided. The Least Scaling Factor field on the Configuration Setup screen is used for setting the smallest unit of scaling.
If scaling factors are not specified, the default scaling factor 1.0 is used.
If the scaling center point is not specified, the G51 command point is used for the scaling center.
Scaling can be enabled/disabled for a particular axis on the Configuration Setup screen.

Example

Here is a BNC sample using the scaling codes:

Using G91 — G00 X20. Y20.

G51 X40. Y40. I.5

G01 X40.

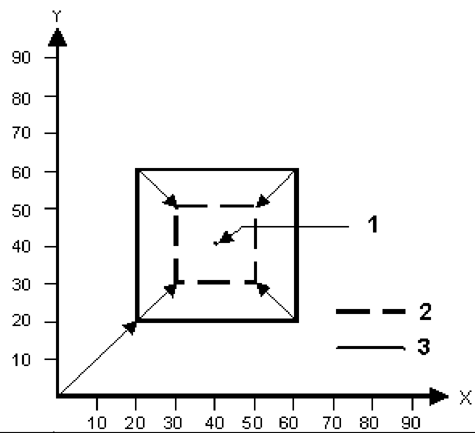
Y40.

X-40.

Y-40.

G50

The following diagram illustrates the previous code sample:



1	Centerpoint
2	Scaled
3	Original

Figure 2–12. G51 Scaling Code

The Scaling (G51) command must always be canceled with a Cancel Scaling (G50) command.

Mirror Image (G50.1 and G51.1)

Mirroring (G51.1) and Mirroring Cancel (G50.1) commands are used when the shape of the workpiece is symmetric to an axis. The whole part can be prepared by programming a subprogram and using programmable mirror imaging. Ordinary mirror image comes after the programmed mirror image. The first movement command must be absolute when in this mode. The following actions occur when the mirror image is applied to only one axis composing a plane:

- Circular Command: CW and CCW are reversed.
- Cutter Compensation: Right and Left Offset are reversed.
- Coordinate Rotation: Rotation angle becomes reversed.

Format

The formats of the mirroring codes are as follows:

G51.1 X___ or Y___ or Z___

G50.1 [X0] or [Y0] or [Z0]

Specifying a G50.1 with no X, Y, or Z parameter cancels the mirroring code in the X, Y, and Z axes. The coordinates about which the tool path will be mirrored are in absolute values. The mirroring codes create the following special conditions:

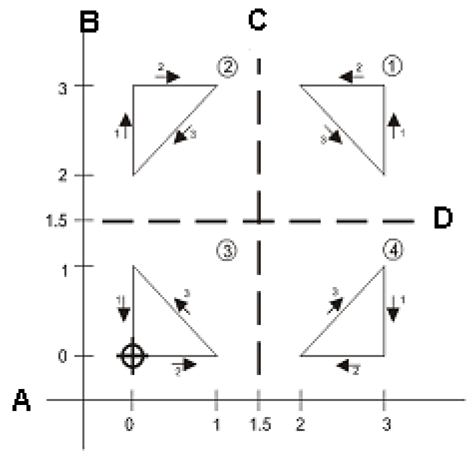
- For circular commands CW and CCW are reversed.
- Cutter compensation for Right and Left are reversed.
- Mirroring G51.1 codes may not be nested.

G51.1 is used to mirror a tool path about the X, Y, or Z axis while G50.1 is used to cancel mirroring for the X, Y, or Z axis.

⇨ This mode is canceled by G50.1. The first movement command after a G50.1 command must be an absolute command. This mode must not be specified in the G68 or G50 mode.

Example

In the illustration below, part #1 (in the upper right corner) is mirrored three times into part #2, #3 and #4. Note that the direction of the tool path (shown as directional arrows numbered 1, 2, and 3) on each part changes with each mirroring operation:



A	X Axis
B	Y Axis
C	X Mirror Axis
D	Y Mirror Axis

Figure 2–13. BNC G50.1 and G51.1 Mirroring Codes

The following program example mirrors the part as shown in the example from the previous page:

NC Part Program 1 inch MIRROR.FNC

```
%
N10 (FIG 7-94 MIRRORING EXAMPLE)
N42 (   )
N44( MAIN PROGRAM )
N46 (   )
N50 M98 P8888
N60 (2-PART#1 MIRRORING IN X)
N70 G51.1 X1.5
N80 M98 P8888
N90 (3-MIRRORING CONTINUED IN Y)
N100 G51.1 Y1.5
N110 M98 P8888
N120 (CANCEL INITIAL X & Y MIRROR)
N130 G50.1
N140 (4-PART#1 MIRRORED IN Y)
N150 G51.1 Y1.5
N160 M98 P8888
N170 M02
N172 (END OF MAIN PROGRAM)
N180 (   )
N190 (SUB-PROGRAM 8888)
N200 (1-PART#1 UPPER RIGHT)
N210 (TRIANGLE 3,2 3,3 2,3)
N215 (   )
N220 O8888
N230 G00 G90 T01 M06
N240 X0 Y0 Z.05 M03 S800
N250 G00 X3 Y2 Z0
N260 G1 Y3 F50
N270 X2
N280 X3 Y2
N290 M99
E
```

Local Coordinate System Setting (G52)

While programming in a work coordinate system, it is sometimes more convenient to use a local coordinate system within the current work coordinate system. The zero point of each local coordinate system is equal to the X, Y, Z, position of the current work coordinate system.

To cancel the local coordinate system, the zero point of the local coordinate system should be matched with the zero point of the work coordinate system by using the G52 X0 Y0 Z0 command. The local coordinate system can also be canceled by switching to another work coordinate system (including the current work coordinate system) or to the machine coordinate system (G53).

The G52 command is most useful in conjunction with subprogram commands. A subprogram could be used to cut a part and the local coordinate system can be shifted to cut a number of similar parts.

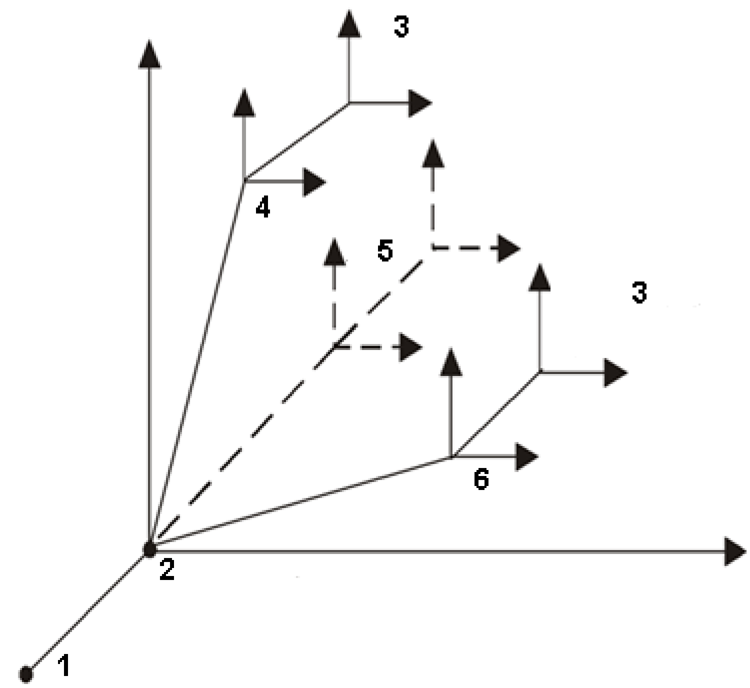
Format

The format of the local coordinate system command is as follows:

G52 X_____Y_____Z_____

Example

This illustration shows setting a local coordinate system using the G52 command:



1	Machine Zero Point
2	Shift Offsets
3	Local Coordinate System
4	G54
5	G55 through G58
6	G59

Figure 2–14. Setting Local Coordinate System Using G52

The following is a sample program, which uses G52 to set local coordinates:

NC Part Program **1** **Inch**
LOC_COOR.FNC

```
%
N10 G00 G90
N40 M25
N45 X0 Y0
N50 T1 M06
N60 Z5.
N90 S2000 M03
N100 Z0.05
N110 M98 P2121
(USE LOCAL COORD SYSTEM)
N240 G52 X-1.5 Y-1.5
N320 G65 P2121
N380 G52 X1.5 Y-1.5
N390 M98 P2121
N430 G52 X0 Y-3
N440 M98 P2121
N430 Z5.
N1170 M25
N1190 M05
N1200 M02
O2121
N500 X1
N510 Y1
N520 X0
N530 Y0
M99
```

Machine Coordinates (G53)

This Machine Coordinates (G53) command moves the tool to the X, Y, Z, A, B machine coordinate position at rapid traverse. This command is only effective in the block in which it is specified and in Absolute mode (G90). The system reverts to the last commanded work coordinate system.

If a local coordinate (G52) is used before a machine coordinate (G53) is commanded, the local coordinate is canceled when the system goes back to the last commanded coordinate system. Reinstate the local coordinate system with another G52.

Format

The format of the machine coordinates command is as follows:

G53 X____Y____Z____ A ____ B ____

Example

Before running this sample program, set the shift offsets to X0 Y0 Z0 and set part zero to X2.0 Y3.0 and Z1.0.

```
%
G00 G90
M25
X0 Y0
T1 M06
Z5.
S2000 M03
Z0.05
G01 X1 F30.
Y1
X0
Y0
(USE MACHINE COORD SYSTEM)
G01 G53 X0
G53 X1 F30.
G53 Y1
G53 X0
G53 Y0
G53 G00 Z5.
M25 M05
M02
E
```

⇨ When running a program on the control, do not use negative shift offsets with G28 or G53. An error message will occur since the negative machine positions cannot be implemented.

Multiple Work Coordinate Systems (G54–G59)

These modal commands select the work coordinate systems 1–6. The work coordinate systems are affected by the work offsets, the shift offset, and the G92 (Set Part Zero) command. Coordinate system 1 is the same as the part setup and it is the default coordinate system. Coordinate systems 1–6 are established by manually entering work offset values for G55–G59 on the Work Offset screen or with the G10 command.

Use the G10 command to set tool offsets, enter tool wear data, and change work coordinate systems, and use the G92 command to set part zero. All six work coordinate systems can be moved an equal distance and direction by using the G92 command.

Format

The format of the multiple work coordinates command is as follows:

G54 (Select work coordinate system 1)

G55 (Select work coordinate system 2)

G56 (Select work coordinate system 3)

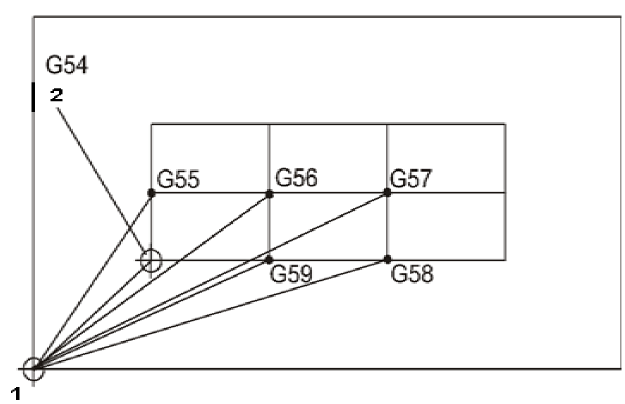
G57 (Select work coordinate system 4)

G58 (Select work coordinate system 5)

G59 (Select work coordinate system 6)

Example

When in the NC mode, the Part Setup screen has the Work Offsets (F1) softkey to display up to six work coordinates (G54–G59) and a set of shift offset values. As shown below, these codes are used to set multiple part zeroes for multiple parts fixtured to the table and milled consecutively using the same part program.



1	Machine Zero
2	Part Zero

Figure 2–15. Work Offset G Codes for Multiple Parts

The coordinates defining G54 are the part zero coordinates for the original part defined on the Part Setup screen. Set the X, Y, and Z values for the G54 to G59 codes. These work offsets are stored in memory, but not with the part program.

The G54 work offsets are the same registers as those in the Part Setup screen for Part Zero X, Y, and offset Z. Editing G54 work offsets for multiple coordinate systems updates the part setup for X, Y, and Z on the Part Setup screen.

Aux Work Coordinate Systems (G54.1)

There are 93 additional X, Y, Z, and optional Rotary A and B work offsets available in NC programming. To access any of these offsets call G code **G54.1 Pn**, where **n** is 1 thru 93. For example, to change to auxiliary work offset 46, call **G54.1 P46** in the NC program.

To update work offset values, use data setting G code **G10 L20 Pn** to set the Auxiliary work offsets values. For example, to update work offset 46 value call **G10 L20 P46 X12.5 Y3.0 Z-0.5**

Precision Cornering On (G61) and Off (G64)

Precision cornering allows non-tangent blocks to be milled with precise corners, regardless of programmed feedrate.

⇨ Precision cornering works differently on machines that have the UltiPro II option installed. Please use the tables below to determine how precision cornering will operate on your machine.

The NC Precision Cornering codes work in the following manner in standard Hurco machines. If you have the UltiPro II option installed, refer to the Precision Cornering with UltiPro II Option table.

Code	Action
G61	Causes the axes to decelerate to zero velocity at the end of a block, if the blocks are not tangent. Tangency is defined as an angle of 5° or less between two consecutive blocks. If the angle is greater than 5°, the system stops and then accelerates to the programmed feedrate in the next block.
G64 (default)	Causes the axes to decelerate to zero velocity at the end of a block, if the blocks are not tangent. Tangency is defined as an angle of 44° or less between two consecutive blocks. If the angle is greater than 44°, the system stops and then accelerates to the programmed feedrate in the next block. The first line is marked as a stop when complete.

Table 2–5. Standard Precision Cornering

The NC Precision Cornering codes work in the following manner in Hurco machines that have the UltiPro II option installed. If you do not have the UltiPro II option installed, please refer to the Standard Precision Cornering table.

Code	Action
G61	Causes the axes to decelerate to zero velocity at the end of a block, if the blocks are not tangent. Tangency is defined as an angle of 44° or less between two consecutive blocks. If the angle is greater than 44°, the system stops and then accelerates to the programmed feedrate in the next block. The first line is marked as a stop when complete.
G64 (default)	Causes the axes to traverse all blocks at a constant feedrate and blends for constant surface finish with no regard to tangency. The first line is not marked as a stop when complete.

Table 2–6. Precision Cornering with UltiPro II Option

Special Program Support

Rotation (G68 and G69)

The Coordinate Rotation (G68) command turns on coordinate system rotation, and the Coordinate Rotation Cancel (G69) command turns off coordinate system rotation.

Format

The G68 code uses this format to command rotation:

G68 (XY___ or XZ ___ or YZ___) R___

When the G17 plane is used, X and Y addresses are used in the format to describe the center point. When G18 is used, X and Z describe the center point. If the plane is defined using G19, the Y and Z addresses define the center point.

R specifies the angle of rotation. A positive R value indicates a CCW direction, and a negative R value indicates a CW direction. When the coordinate values of rotation center are omitted, the current position is used as the center point.

The range of R depends on whether BNC or ISNC is selected and whether an integer or decimal value is specified. Here are the R ranges for each NC type:

BNC: R has a range of -360.0 to +360.0, whether an integer or real number is used.

ISNC: Units of R have a value of 0.001° when R is an integer.

R has a range of $-360{,}000 \leq R \leq 360{,}000$ when R is an integer value.

R has a range of -360.0 to +360.0 when R is a real number.

Rotation is canceled with a G69. Do not use G17, G18, or G19 while in the G68 mode. Use G69 to disable the G68 mode, change the plane, and then go back to the G68 mode.

⇨ G68 codes may not be nested.

Example

This program uses the rotation codes:

ISNC Part Program 1 Inch G68.FNC

```
%
(USING REAL NUMBER WITH G68)
T1 M06
G68 X0 Y0 R45
Z5.05
G01 Z-0.5 F10.
G91 X1.0
Y2.0
X-1.0
Y-2.0
(CANCEL ROTATION)
G69
(USING INTEGER NUMBER WITH G68)
G68 X0 Y0 R45000
X1.0
Y2.0
X-1.0
Y-2.0
(CANCEL ROTATION)
G69
M05
M02
```

Global Rotation NC Transform Plane (G68.2) and Local Rotation NC Transform Plane (G68.3)

Global Rotation NC Transform Plane (G68.2) and Local Rotation NC Transform Plane (G68.3) are functional only with the VMX42 SR machine.

The NC Transform Plane block is not a motion block; it does not execute motion.

G68.2

G68.2 specifies global rotations for the A, B, and C angles in the NC Transform Plane block. G69 cancels G68.2.

The rotation sequence will be in the order of A, followed by B, followed by C, where all rotations are around the X, Y, and Z axes of the Workpiece Coordinate System. Refer to *Getting Started with WinMax Mill*.

G68.3

G68.3 specifies local rotations for the A, B, and C angles in the NC Transform Plane block. G69 cancels G68.3.

The rotation sequence will be in the order of A, followed by B, followed by C, where all rotations are around the X, Y, and Z axes of the rotating Transform Plane Coordinate System.

- Rotation of the A angle will be around the workpiece's X-axis.
- Rotation of the B angle will be around the Y-axis of the rotating Transform Plane Coordinate System that has been rotated by the A-angle.
- Rotation of the C angle will be around the Z-axis of the Transform Plane Coordinate System that has been rotated by the A-angle and B-angle rotations.

Format

The G68.2 code uses this format to command rotation of the NC Transform Plane:

G68.2 X_Y_Z_A_B_C_

The G68.3 code uses this format to command rotation of the NC Transform Plane:

G68.3 X_Y_Z_A_B_C_

All subsequent tool positions (X, Y, Z positions, I, J, K Tool Vectors, and U, V, W Surface Normal Vectors) are specified with respect to NC Transform Plane, except program rotary angles (i.e., B and C angles).

Rotary moves are permitted while NC Transform Plane is active.

NC Transform Planes can be stacked and the coordinates are relative to the last stacked coordinate system.

NC Hole cycles are permitted only if the Tool Vector is aligned with the Z-direction of the Transform Plane.

Coordinate System Rotation Cancel (G69)

Coordinate System Rotation Cancel (G69) cancels the following rotation commands:

- Coordinate Rotation (G68)
- Global Rotation NC Transform Plane (G68.2)
- Local Rotation NC Transform Plane (G68.3)

Units of Measure (BNC G70, G71)

Before setting the coordinate system at the beginning of the program, the units of measure must be specified in an independent block. A part program may switch between English and Metric modes as long as the format of the dimensions is correct for the chosen mode.

The Imperial Units of Measure code (BNC G70 signals the system that the dimensions are in inches.

BNC G70 is canceled by G71.

The Metric Units of Measure code (BNC G71signals the system that the dimensions are metric units.

BNC G71 is canceled by G70.

Format

These are the command formats for the inch/metric conversion commands:

BNC:

G70: Inch command

G71: Metric command

⇨ The BNC G70 and G71 codes do not affect the units of measure used in the graphics and machine status display screens. The displays are controlled by the units selected when entering NC editing.

Peck Drilling (G73)

For Peck Drilling, the spindle moves down in incremental steps and retracts to a position set on the Holes Parameter screen. After each peck, the drill is retracted by the Peck Clearance Plane value set on the General Parameters screen. These screens are described earlier in this manual.

Spindle positioning is performed on the XY plane and hole machining is performed on the Z axis. These parameters are stored as modal values; therefore, if a parameter value does not change for subsequent drilling commands, those commands do not have to contain the parameter.

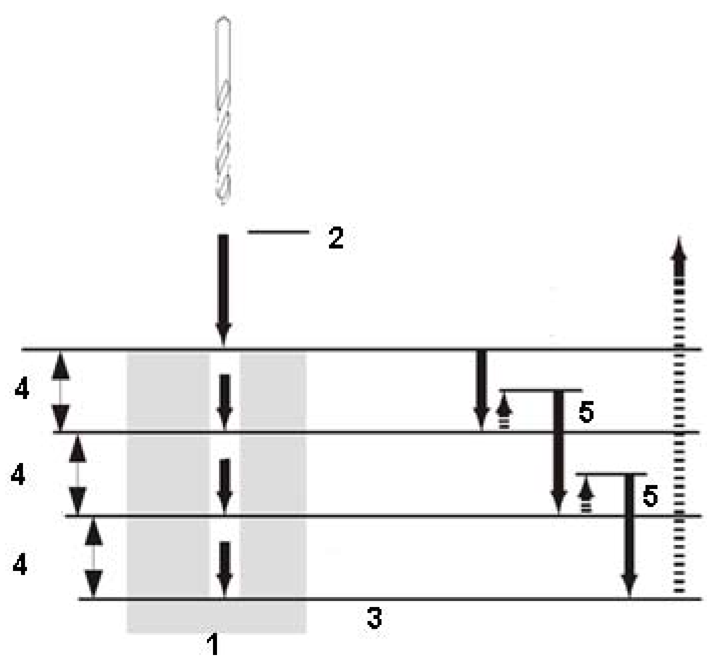
Format

The command format for the Peck Drilling canned cycle is as follows:

G73 X____, Y____, Z____, R____, Q____, F____, [K____ or L____]

Example

The diagram below illustrates tool movement for the G73 command:



1	Part
2	Z Start (Basic); Return Point (Industry Standard)
3	Z Bottom
4	Peck Depth
5	NC Parameter Peck Clearance Distance

Figure 2–16. Tool Movement for the Peck Drilling Cycle (G73)

Left-Handed Tapping Cycle (ISNC G74)

During the Left-Handed Tapping Cycle the spindle rotates CCW to the bottom of the hole. Then the spindle stops, an optional dwell is performed, the spindle rotates CW, and left-handed tapping is performed.

The positioning for this cycle is performed on the XY plane and hole machining is performed on the Z axis. During left-handed tapping, the feedrate override is ignored and the cycle does not stop until the end of the return operation, even if a feed hold is applied.

If a Start Spindle Clockwise (M3) code is in effect, the spindle direction will be reversed prior to executing a G74 cycle. Rigid Tapping is performed when an Enable Rigid Tapping (ISNC M29) code is used in a block previous to the G74 block.

Format

The command format for the Left-Handed Tapping cycle is as follows:

G74 X____, Y____, Z____, R____, P____, F____, [Q____,]
[K____, or L____].

Z is the distance from the R Point (in Return to R Point in Canned Cycle [G99] mode) to the Z Bottom or the distance from the Initial Point (in Return to Initial Point in Canned Cycle [G98] mode) to the Z Bottom.

Q is the optional peck depth. If Q equals 0.0, pecking is not performed. Q used for G74 with M29 applies only to rigid tapping.

Single-Quadrant Circular Interpolation (BNC G74)

The Single-Quadrant Circular Interpolation Mode (G74) causes the system to interpolate arcs and helices in a single quadrant only. The arc or helix must remain within the quadrant in which it started (the arc or helix cannot be larger than 90°). Since the arc cannot cross quadrants, the center point is determined by looking toward the center of the arc from the start point. I, J, and K are unsigned incremental distances from the arc start point to the center of the arc.

G74 is canceled by G75.

Multi-Quadrant Circular Interpolation (BNC G75)

The Multi-Quadrant Circular Interpolation Mode (G75) is the default and causes the system to interpolate an arc or helix across all quadrants. The arc or helix can start and end in any quadrant. An arc or helix may be up to 360° in this mode. The center point data can be represented in two different ways based on the current machine dimension mode (G90 and G91).

G75 is canceled by G74.

Bore Orient (G76)

The Bore Orient cycle provides a feed-in, stop-feed, orient spindle, move tool away from part surface, rapid-out, and spindle restart sequence suitable for boring operations when the tool needs to be moved away from the part surface before retracting out of the hole. If the default Bore Orient Retract vector is not suitable, I and J words may be used to specify a new retract position.

A value needs to be entered in the Bore Orient Retract field on the Holes Parameters screen (described earlier in this manual). That value specifies the distance the X and Y axes travel to retract the tool from the part surface during the Bore Orient cycle.

A spindle oriented stop is performed at the bottom of the hole and the spindle retracts after shifting in the direction opposite to the cutter direction. High precision and efficient boring is performed without scratching the workpiece surface.

⇨ The Bore Orient G86 mode continues to be supported to provide compatibility with existing BNC programs.

The bore orient cycle moves the axes in this manner:

1. The spindle should already be switched on.
2. The spindle positions the tool at the rapid speed to the XY location, if necessary.
3. The spindle moves down at the specified feedrate to the Z value.
4. The spindle stops and orients.
5. The spindle moves from the XY location to the IJ position or to the Bore Orient Retract distance.
6. The system rapidly moves Z to the initial Z location.

⇨ This cycle applies only to machines that have an electronic or mechanical orient feature (refer to the machine tool owner's manual).

Format

The format of the Bore Orient cycle is as follows:

G76 X____, Y____, Z____, [I_____, J_____, or Q____] R____, P____, F____, [K____, or L____]

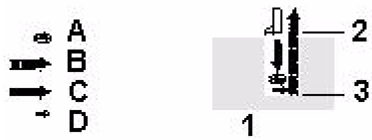
⇨ I and J may also be used instead of Q to specify an incremental bore shift value and direction. If Q is used, the Q value must be a positive number; otherwise, an error message will occur.

For BNC, Q and I, J are optional to maintain compatibility with older programs. A default Z value of 1.0 will be used for BNC if a Q word is not contained in the same block with the G86 command. Q is not modal for BNC.

⚠ The Q value is modal. Since Q is used as the cut-in value for G73 and G83, use care when specifying Q.

Example

The diagram below illustrates tool movement for the Bore Orient cycle:



1	Part
2	Z Start (Basic); Return Point (Industry Standard)
3	Z Bottom

A	Stop and Orient Spindle
B	Repid Feed
C	Feedrate
D	Move Tool

Figure 2–17. Tool Movement for the Bore Orient Cycle (G76)

Canned Cycle Cancel (G80)

Canned Cycle Cancel is a machine default mode and cancels all canned cycles. When a cycle is canceled using a G80, program execution returns to the One-Shot (G00, G01, G02, or G03) mode that was in effect before the canned cycle was executed. Use either G00, G01, G02, or G03 to cancel a canned cycle.

The G80 cycle also cancels the R and Z Points. That is, R = 0 and Z = 0 for the incremental command. Other drilling data are also canceled.

Drill, Spot Boring (G81)

The Drill, Spot Boring cycle is a feed-in, rapid-out sequence. The axes move in this manner with the spindle switched On:

1. Ensure that the initial Z location is above Z bottom and above any obstructions.
2. The tool is positioned at the Initial Z location and moves at the rapid speed to XY if it is in the block.
3. The spindle drills down to Z Bottom at the specified feedrate.
4. The spindle moves up to Z Start at the rapid speed.

Format

The command format for Drill cycle is as follows:

G81 X____, Y____, Z____, R____, F____, [K____, or L____]

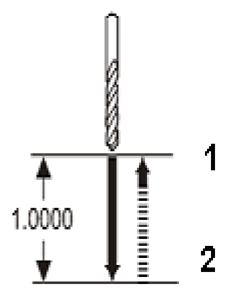
Example

This is a sample BNC Drilling cycle:

G81 Z1.0000 (inches) G90 or G91

Here is a sample ISNC Drilling cycle and a tool movement diagram:

G81 Z-1.0000 (inches) in G91 mode



1	Z Start (Basic); Return Point (Industry Standard)
2	Z Bottom

Figure 2–18. Tool Movement for the Spot Boring Cycle (G81)

Drill with Dwell, Counter Boring (G82)

The Drill with Dwell, Counter Boring cycle provides a feed-in, dwell, and rapid-out sequence.

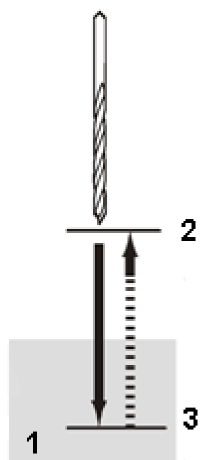
Format

The command format for the Drill with Dwell cycle, or Counter Boring, is as follows:

G82 X____, Y____, Z____, R____, P____, F____, [K____, or L____]

Example

This diagram illustrates tool movement for the Counter Boring cycle:



1	Part
2	Z Start (Basic); Return Point (Industry Standard)
3	Z Bottom Dwell Point

Figure 2–19. Tool Movement for the Counter Boring Cycle (G82)

Deep Hole Drilling (G83)

The Deep Hole Drilling cycle provides a sequence of feed-in and rapid-out movements until the specified hole depth is reached.

For BNC, each feed-in moves the distance of the peck depth. The tool will rapid back to the Z Start position.

For ISNC the tool will rapid out to the Return point.

Next the tool will rapid down until it reaches the starting point for the next peck (for either BNC or ISNC). The starting point is an incremental distance above the last peck, defined on the Holes Parameter screen as the Peck Clearance Distance.

BNC has three Z values: Z1, Z2, and Z3. They may be programmed in this canned cycle and are unsigned incremental distances. There is a rapid traverse back to R at the end of each pecking cycle and then the tool feed begins above where the tool stopped during the last pecking cycle.

Z1 is the total depth for the hole.

Z2 is the depth of the first peck.

Z3 is the depth for each of the remaining pecks.

Z2 and Z3 must be smaller than Z1.

⇨ If Z2 and Z3 are not programmed, this canned cycle functions like G81.

If Z3 is not programmed, Z2 is the depth for each peck. The last peck for the hole is the programmed peck depth or the remaining distance from the last peck to the bottom of the hole, whichever is smaller.

If Z1, Z2, and Z3 do not change between G83 blocks, they need not be reprogrammed. Use the Precision Cornering codes (G61 and G64) to control the Z axis deceleration between pecks.

Format

The command formats for the Deep Hole Drilling cycle are as follows:

BNC: G83 X____,Y____, Z____, [Z____,] [Z____,] R____,
F____,[K____, or L____]

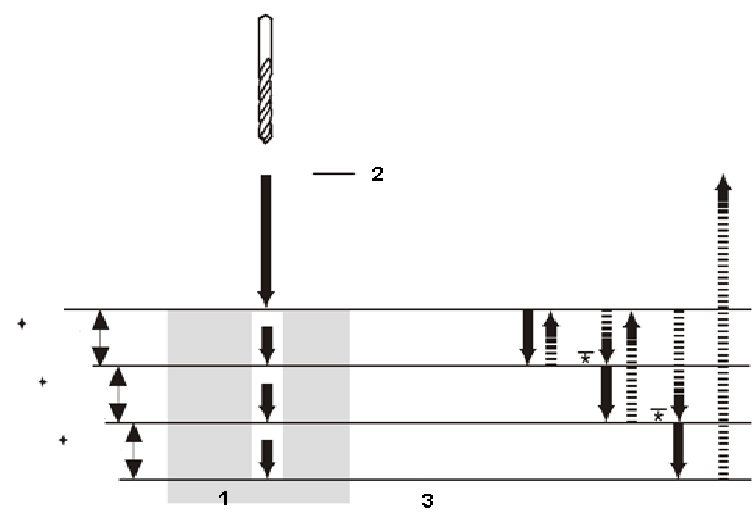
> For BNC, the first Z is the distance from Z Start to Z Bottom. The second Z is the first cut-in depth. The optional third Z is the depth of the remaining pecks. The Zs are always positive. All of the peck depths will be the same if the third Z is left out.
>
> For BNC, R is always positive and is an incremental distance from the initial point to point R.

ISNC: G83 X____,Y____, Z____, R____, Q____, F____,
[K____, or L____]

> ISNC has one Z parameter which represents the location of Z Bottom.

Example

The diagram below illustrates tool movement for the G83 code:



1	Part
2	Z Start (Basic); Return Point (Industry Standard)
3	Z Bottom
+	Up to 3 Peck Depths Can be Programmed
*****	NC Parameter Peck Clearance Distance

Figure 2–20. Tool Movement for the Deep Hole Drilling Cycle (G83)

Tapping (G84)

The Tapping cycle provides a tap sequence. The current feedrate (F) and spindle speed (S) are used. The spindle accelerates to the defined speed and the Z axis plunges at the defined feedrate. At the bottom of the hole, the spindle and Z axis decelerate in coordination to a stop. They then reverse directions and accelerate in coordination to the programmed feed and speed. Once back to the original Z level, the spindle shuts off and reverses back to the original direction in preparation for the next operation.

For BNC, G84 is used for right- and left-handed tapping. Start Spindle Clockwise (M3) or Start Spindle Counterclockwise (M4) commands determine whether right- or left-handed tapping is performed.

For ISNC, G84 performs right-handed tapping only. A Start Spindle Counterclockwise (M4) command causes the system to reverse the spindle direction at the start of the cycle to ensure that right-handed tapping is performed.

Use the following formula to calculate the correct feed and speed for the tap cycle:

Feedrate:

Feed in inches or mm per minute = Spindle RPM / threads per inches or mm

Spindle RPM:

Spindle RPM = Feed in inches (mm) per minute × threads per inch (mm)

When an M3/M4 command is detected in a program and the current tool in the spindle is defined as a tapping tool in tool setup, the system looks 10 blocks ahead for another tap cycle, a G01/G02/G03 code, or a canned cycle other than a tap. If any cutting move (G01, G02, G03, or any canned cycle other than a tap) is found within 10 moves or 10 rapid moves are found, the spindle is turned on as usual. If a G84 is found and all moves from the M3/M4 are rapid moves, the spindle is not turned on, and the rapid moves will be executed with the spindle off.

⇨ The spindle rotates clockwise to the bottom of the hole. At the bottom of the hole, the spindle is reversed and rotates counterclockwise and tapping is performed. During the tapping, the feedrate override is ignored and the cycle does not stop until the end of the return operation, even if a feed hold is applied.

For ISNC, a Rigid Tap Enable (M29) command initiates rigid tapping instead of regular tapping. Rigid Tap is disabled with a G00, G01, G02, G03, or G80 command. The programmed feedrate can be overridden for rigid tapping.

Format

The command format for the Tapping cycle is as follows:

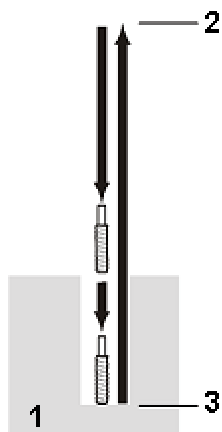
G84X____, Y____, Z ____, R ____, P____, F ____, [Q____,] [K____, or L____]

P is used only with ISNC for the Tapping cycle. P specifies a dwell period at the bottom of the hole and after leaving the hole.

Q, the optional peck depth, is only used with ISNC for the Tapping cycle. If Q equals 0.0, pecking is not performed. M29 is required with Peck for Rigid Tapping.

Example

The diagram below illustrates tool movement for the Tapping cycle (G84):



1	Part
2	Z Start (Basic); Return Point (Industry Standard)
3	Z Bottom Spindle Stop; Dwell Spindle Reverse

Figure 2–21. Tool Movement for the Tapping Cycle (G84)

Boring (G85)

The Boring cycle provides a feed-in and feed-out sequence suitable for boring.

The boring cycle moves the axes in this manner:

1. The spindle should already be switched on using an M3 code.
2. The tool is positioned over the hole location.
3. At the G85, the spindle feeds to Z Bottom as specified.
4. At Z Bottom, the spindle feeds to the Z Start position.

⇨ It is possible to have an XY position move with the G85 code.

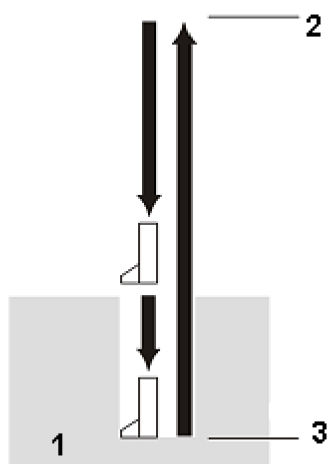
Format

The command format of the Boring cycle is as follows:

G85 X____, Y____, Z____, R____, F____, [K____, or L____]

Example

The diagram below illustrates tool movement for the Boring cycle (G85):



1	Part
2	Z Start (Basic); Return Point (Industry Standard)
3	Z Bottom

Figure 2–22. Tool Movement for the Boring Cycle (G85)

Bore Rapid Out Cycle (ISNC G86)

The ISNC Bore Rapid Out canned cycle is a feed-in, rapid-out sequence. The spindle stops at the bottom of the hole and is retracted at the rapid traverse rate.

The Bore Rapid Out canned cycle moves the axes in this manner with the spindle switched on:

1. The tool is positioned at the Initial Z location and moves at the rapid speed to XY if it is in the block.
2. The spindle bores down to Z Bottom at the specified feedrate.
3. The spindle turns off.
4. The spindle moves up to Z Start at the rapid speed.
5. The spindle turns on.

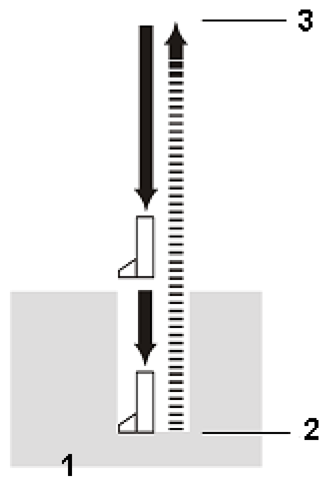
Format

The command format for the Bore Rapid Out cycle is as follows:

G86 X____, Y____, Z____, R____, F____, [K____, or L____]

Example

This diagram illustrates tool movement for the Bore Rapid Out cycle:



1	Part
2	Z Start (Basic); Return Point (Industry Standard)
3	Z Bottom Spindle Stop

Figure 2–23. Tool Movement for the Bore Rapid Out Cycle (G86)

Chip Breaker (BNC G87)

The Chip Breaker cycle provides drilling with a dwell every 0.050" (1.27 mm) to break off the chip. The dwell time is automatically calculated so the spindle revolves two times to break the chip. After the dwell, the system feeds another 0.050" (1.27 mm) and again breaks the chip until the bottom of the hole is reached. This cycle breaks the chip without retracting the tool entirely from the hole as with the Deep Hole Drilling cycle (G83). Use the Precision Cornering codes (G61 and G64) to control the Z axis deceleration between dwells.

The Chip Breaker cycle moves the axes in this manner with the spindle switched on:

1. The tool is positioned at the rapid speed to XY if necessary.
2. The spindle moves down 0.05" at the feedrate.
3. The spindle dwells at that location for two rotations.
4. The spindle moves down another 0.05" at the feedrate.
5. This is repeated until the Z depth is reached.
6. The spindle moves at the rapid speed to the initial Z location.

Format

The format of the Chip Breaker cycle is as follows:

G87 X___ Y___ Z___ F____, [K____, or L____]

Back Boring (ISNC G87)

The Back Boring cycle provides a boring sequence in the positive Z direction. Boring is performed from the specified R level to the Z level. Positioning is performed on the XY plane and hole machining is performed on the Z axis.

Format

The command format for the back boring cycle is as follows:

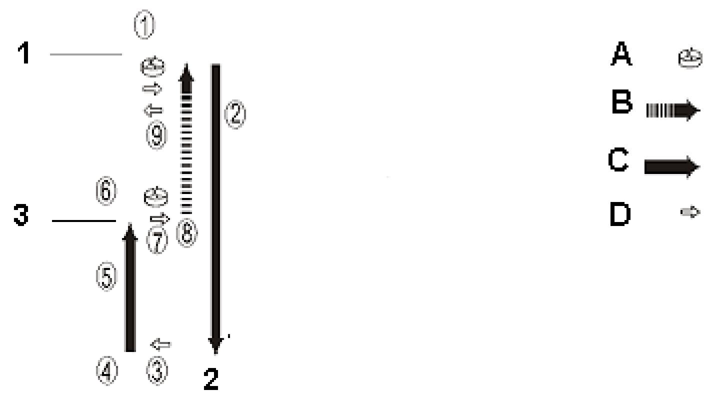
G87 X____, Y____, Z____, R____, Q____, I_____, J_____, P____,
F____, [K____, or L____]

R is used to specify the depth to which the bore moves before shifting over Q or IJ and moving up to the Z level.

⇨ Q is used to store an incremental bore shift value. I and J may also be used instead of Q to specify an incremental bore shift value. I and J can be used to specify a distance and direction. Q can only specify distance; the direction is pre-defined by machine parameters.

ISNC G87 Example

The drawing below illustrates tool movement for the Back Boring cycle (ISNC G87):



1	Z Start (Basic); Return Point (Industry Standard)
2	R Point
3	Z Bottom Dwell Point

1	Spindle Stop, Orient, and Move	A	Stop and Orient Spindle
2	Feed Down to R Point	B	Repid Feed
3	Spindle Stop, Spindle Move	C	Feedrate
4	Spindle Start	D	Move Tool
5	Feed Up to Z Bottom		
6	Dwell Point		
7	Spindle Stop, Orient, and Move		
8	Rapid Move		
9	Spindle Move		

Figure 2–24. Tool Movement for the Back Boring Cycle (ISNC G87)

Rigid Tapping (BNC G88; ISNC G84.2; ISNC G84.3)

Rigid tapping allows the same hole to be tapped repeatedly with precision. The rigid tapping feature increases accuracy by synchronizing the rotation of the spindle with the feed of the Z axis. ISNC G84.2 is used for right-handed tapping, and ISNC G84.3 is used for left-handed tapping. M29 is required for Rigid Tapping.

The format of the rigid tapping cycle is as follows:

M29

G88X____, Y____, Z____, Z____, R____, F____, P____
[K____, or L____]

or

M29

G84.2X____, Y____, Z____, Z____, R____, F____, P____
[K____, or L____]

or

M29

G84.3X____, Y____, Z____, Z____, R____, F____, P____
[K____, or L____]

⇨ The second Z parameter defines the peck depth.

Canned Boring with Manual Feed Out and Dwell (ISNC G88)

With this canned cycle, a dwell is performed at the bottom of the hole and the system goes into Interrupt mode. The spindle can then be retracted manually using the jog controls. When the desired manual position is reached, follow these steps:

1. Press the console Auto button (in Machine Mode group).
2. The Start button starts flashing and the “Press Start Button” message displays.
3. Press the Start button.
4. The program finishes the canned cycle and then continues with the rest of the program.

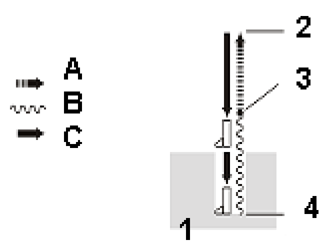
Format

The command format for the Boring With Manual Feed Out and Dwell canned cycle is as follows:

G88 X____, Y____, Z____, R____, I_____, J_____, P____, F____, [K____, or L____]

Example

The drawing below illustrates tool movement for the Canned Boring with Manual Feed Out and Dwell cycle (ISNC G88):



1	Part
2	Z Start (Basic); Return Point (Industry Standard)
3	Z Bottom Spindle Stop after Dwell
4	Resume Program Execution
A	Rapid Feed
B	Manual Retract
C	Feedrate

Figure 2–25. Tool Movement for ISNC G88 Cycle

Bore with Dwell (G89)

The Bore with Dwell cycle provides a feed-in, dwell, and feed-out sequence.

The Bore with Dwell cycle moves the axes in this manner with the spindle switched on:

1. The tool positions at the rapid speed to XY position, if necessary.
2. The spindle moves down at the feedrate to Z Bottom.
3. The spindle stays at the Z Bottom position for the specified dwell time.
4. The spindle moves Z up to Z Start at the rapid speed.

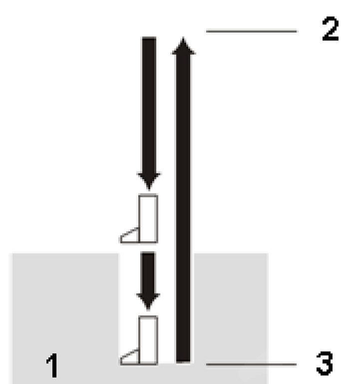
Format

The command format for the Bore with Dwell cycle is as follows:

G89 X____, Y____, Z____, R____, P____, F____, [K____, or L____]

Example

The drawing below illustrates tool movement for the Bore with Dwell cycle (G89):



1	Part
2	Z Start (Basic); Return Point (Industry Standard)
3	Z Bottom Dwell Point

Figure 2–26. Tool Movement for the Bore with Dwell Cycle (G89)

Absolute and Incremental (G90, G91)

The Absolute Machining Mode (G90) is the default and signals the system that the programmed dimensions are relative to part zero. Once programmed, this default stays in effect until canceled with a G91.

The Incremental Machining Mode (G91) signals the system that all programmed dimensions are incremental distances from the position in the previous block. Once programmed, this mode stays in effect until canceled with a G90.

If Absolute Machining Mode (G90) is activated, the center points I, J, and K are absolute Cartesian (rectangular) coordinates from part zero.

If Incremental Machining Mode (G91) is activated, the center points I, J, and K are signed incremental distances from the arc start point.

Format

This is the command format for each position command:

Absolute command:

G90 X_____Y_____Z_____

Incremental command:

G91 X_____Y_____Z_____

Example

A machine is resting at the programmed part zero location, and the following blocks are executed in inches:

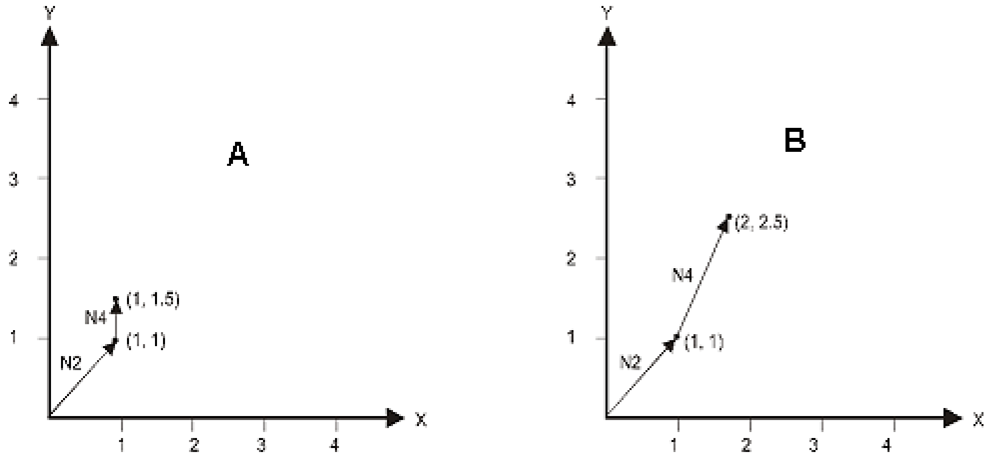
N2 G01 X1.0 Y1.0 F10.0

N4 X1.0 Y1.5

If the system is in Absolute Machining mode (G90), the N2 block causes the axes to travel at a 45° angle to the 1.0" position in X and 1.0" in Y. As a result of the N4 block, the machine remains at the 1.0" position in X and Y moves to the 1.5" position.

If the system is in Incremental Machining mode (G91), the N2 block causes the axes to travel at a 45° angle to the 1.0" position in X and the 1.0" position in Y—just as before. But, as a result of the N4 block, X continues to move 1.0" to the 2.0" position; Y moves 1.5" to the 2.5" position.

The diagram below illustrates absolute and incremental axis moves.



A	Absolute
B	Incremental

Figure 2–27. Differences Between Absolute and Incremental

Coordinate System Setting

This section explains the commands used for these coordinate system settings: part zero, machine coordinates, multiple work coordinates, local coordinates, polar coordinates, and automatic return to and from reference point.

Part Zero Setting (G92)

This command establishes the work coordinate system so that a certain point of the tool, for example the tool tip, becomes X, Y, Z, A, B in the established work coordinate system. The distance shifted with this command is added to all subsequent work coordinate system zero point offset values; all work coordinate systems move by the same distance. The G92 command can be used in any work coordinate system (G54–G59).

⇨ Cancel Scaling (G50) must be active before selecting G92.

A G92 command makes the dimensions included in the block the new part relative position for the current machine location. The new part zero location is calculated from the current location of the axes and the dimensions included in the G92 block.

The part zero location is only altered for dimensions programmed in the G92 block. This makes it possible to alter the part zero locations for certain axes without affecting the others.

G92 is invalid while *cutter compensation* is on.

Format

This is the format of the setting part zero command:

G92 X_____Y_____Z_____A_____B_____

Example

Set Part Zero (G92) establishes new part relative coordinates at the current axis positions. For example, if the machine is positioned at part relative X2.0 and Y2.0, the block G92 X0.0 Y0.0 would make the current X and Y axis part relative positions equal 0.0. The machine axes will not move, but the status screen changes to reflect the new part zero reference point(s). Any programmed coordinates after the G92 block are referenced to the new part zero location(s).

Use the G92 code for repeating parts of a program at another location. The following is a sample of the codes used in incremental mode. Refer to the diagram below for an illustration of these codes.

NC Part Program 1 Inch
PARTZERO.FNC

```
%
N10 G0 X20. Y20.
N12 X40.0
N14 Y40.0
N16 X20.0
N18 Y20.0
N20 X70.
N22 G92 X0. ⇐ Set new part zero
N24 X20.0
N26 Y40.0
N28 X0.0
N30 Y20.0
M02
```

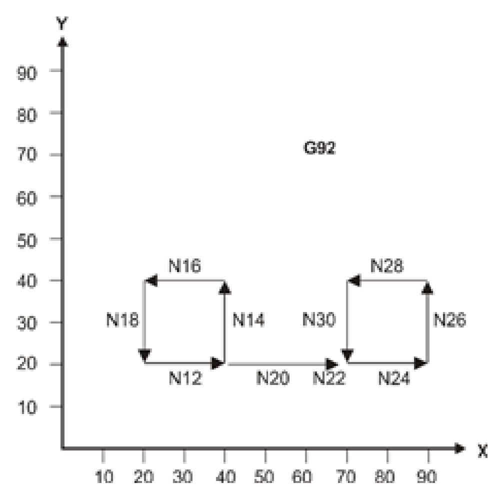


Figure 2–28. Set Part Zero (G92)

Feed Functions

The Feedrate (F words) value establishes the non-rapid move feedrate. It remains active for all non-rapid moves until another Feedrate code is entered.

For Inch units in Basic NC, the actual feedrate is dependent on the usage of a decimal point. If no decimal point, the actual feedrate is one-tenth of the programmed feedrate (F30 equates to 3 inches per minute). If a decimal point is specified then the actual feedrate will be the programmed feedrate (F30.0 is 30 inches per minute). For metric units in Basic NC, the actual feedrate is the same as the programmed feedrate regardless of a decimal point (F75 and F75.0 are both 75 millimeters per minute).

The Feedrate code is active before the other commands in the program block are executed. G94, Feed per Minute Feedrate, is the default setting unless otherwise specified.

Inverse Time Feedrate (G93) and Feed Per Minute Feedrate (G94)

The default setting for Feedrate is G94 for Feed per Minute Feedrate, either inches per minute or millimeters per minute. G93 cancels G94 and G94 cancels G93.

Inverse Time (G93) can be specified to change the feedrate as a function of time and distance. If the time is unchanged but the distance changes then the actual feedrate will change proportionally. The format for Inverse Time is F6.3 (maximum of six digits before the decimal point and maximum of three digits after the decimal point) and the units are minutes. Feedrates of up to 999999.999 can be programmed using G93. The time is computed by dividing one by the Inverse Time programmed. The actual feedrate is the distance divided by the time.

The feedrate must be specified for every move.

Example

G93 G1 X5.0 F10.0

Y7.0 F10.0

Time is 1/10.0min = 0.1min

Actual Feedrate for first line is 5.0in/0.1min = 50ipm.

Actual Feedrate for second line is 7.0in/0.1min = 70ipm.

Format

G93 X____ Y____ Z____ A____ C____ F____ (activate Inverse Time)

X ____Y____ Z____ A____ C____ F____

...

G94 X____ Y____ Z____ F____ (cancel Inverse Time and enable Feed Per Minute feedrate)

Rotary Tangential Velocity Control (G94.1) (preliminary)

The Rotary Tangential Velocity Control (G94.1) command maintains the tool tip feedrate regardless of tool or part linear or rotary motion. This command works identically for 4- and 5-axis simultaneous motion, transitions between 4- and 5-axis simultaneous motion, linear to rotary moves, combined rotary to linear moves, or rotary-only moves.

G94.1 computes feedrate changes in real time as the tool approaches and moves away from a centerline of rotation, even if this happens during the execution of a single NC block. For example, on a 4-axis machine with Rotary A configuration, if the tool tip starts away from the rotary centerline and the NC block commands only the Y-axis to move such that the tool tip moves over and past the A centerline of rotation while rotating the A-axis, the machine will start with slower axes movement, then speed up to higher A- and Y-axes feedrates when the tool tip is just above the centerline of rotation. It will then slow down as it moves away from the centerline. This happens because the radial distance of the tool tip is shortest when the tool is just above the centerline, so the axes will have to move more quickly to maintain the feedrate.

⇨ Absolute Tool Length mode is the recommended tool calibration mode for use with the G94.1 command. If Z calibration mode is used, the Part Setup Offset Z, Z Table Offset, Tool Z calibration, and Rotary and Tilt axes centerline locations must be set. See “Tool Calibration Modes” in *Getting Started with WinMax Mill* for more information.

⇨ G94.1 is not supported in M128 or Transform Plane modes.

Example

```
G21

G94.1

G1A90F1000
```

When executed on a 4 axis machine with Rotary A configuration, this example will result in a 90 degree arc cut on the cylinder circumference. The move will maintain a tool tip velocity with respect to the rotating part of 1000mm/min.

Canned Cycle Descriptions

Canned cycle descriptions, formats, and examples follow.

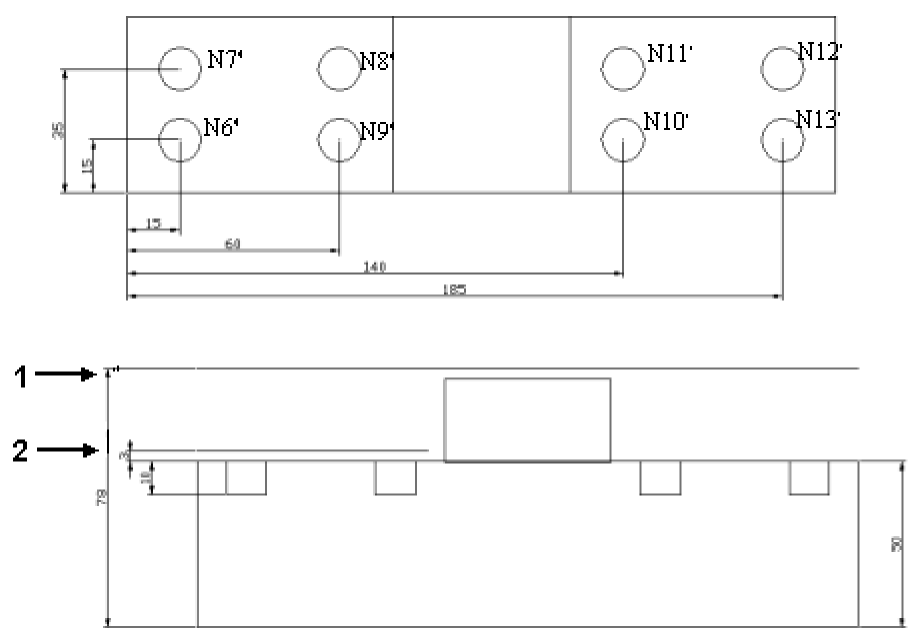
Return to Initial Point in Canned Cycles (G98)

Position the Z axis to the initial level. The initial level is the last position of the Z axis before the canned cycle is started. The Z axis rapids or feeds to the Z Retract Clearance level, based on the canned cycle being performed. Z Start in the canned cycle description is then equal to the initial point.

Format

G98 (no parameters follow)

Example



1	Initial Level
2	Retract Level

Figure 2–29. Return to R Plane Example

Sample NC Part Program Using G98

Below is a program example using G98:

```
%
N1 G90 G80 G40 G21
N2 T1M6
N3 G43 H1 S3000 M3
N4 Z78.0 M8
N5 F1270.0
N6 G99 G81 X15 Y15 R28.0 Z-10.0
N7 Y35.0
N8 X60
N9 G98 Y15.0
N10 G99 X140.0
N11 Y35.0
N12 X185.0
N13 G98 Y15
N14 G80
N15 G0 G91 M28 Z0 M5 M9
E
```

Return to R Level in Canned Cycles (G99)

The Return to R Level in Canned Cycles command positions the Z axis to a return (R) level. The Z axis rapids or feeds to the return level between locations during canned cycles. Z Start in the canned cycle descriptions is then equal to the Return Point. Even when the canned cycle is performed in G99 mode, the initial level remains unchanged.

For BNC, specify an R with the G99.

For ISNC, the modal value of R is used.

Format

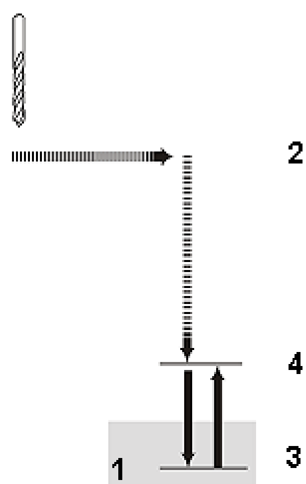
The format of this code is as follows:

G99 R___

> For BNC, the R parameter is an incremental distance from the initial Z level. Use this code to reduce the returned distance between locations during canned cycles.
>
> For ISNC, the R parameter is an absolute Z level in G90 mode and an incremental negative Z distance in G91 mode.

Example

The drawing below illustrates tool movement for the Return to R Level in Canned cycles (G99) command:



1	Part
2	Initial Point
3	Z Bottom
4	Return Point (Industry Standard); Z Start (Basic)

Figure 2–30. Tool Movement for the G99 Cycle

Canned Cycles

Canned cycles use a one-block G code to provide drilling, boring, and tapping operations. Using one G code instead of several helps simplify writing NC programs. Various parameters are used in common with all or most of the canned cycles. For instance, Z is used to specify the canned cycle's depth, P is used to specify dwell time, and F is used to specify the feedrate. For BNC, if there is no spindle speed and direction specified in the program, these values are retrieved from the tool page.

The table below contains canned cycles, G codes, and spindle operation while moving in the negative Z direction, being at Z Bottom, and moving in the positive Z direction.

Operation	G Codes		Spindle Operation		
Canned Cycle	BNC	ISNC	In -Z Direction	At Z Bottom	In +Z Direction
Peck Drilling	G73	G73	Peck Feed	None	Rapid Traverse
Left Handed Tapping	G84 with M04	G74	Feed	Spindle Stop, Dwell, Spindle CW	Feed
Bore Orient	G76 G86	G76	Feed	Oriented Spindle Stop	Rapid Traverse
Canned Cycle Cancel	G80	G80	None	None	None
Drill, Spot Boring	G81	G81	Feed	None	Rapid Traverse
Drill with Dwell, Counter Boring	G82	G82	Feed	Dwell	Rapid Traverse
Deep Hole Drilling	G83	G83	Peck Feed	None	Rapid Traverse
Tapping	G84 with M03	G84	Feed	Spindle Stop, Dwell, ISNC Spindle CCW or BNC Spindle CW	Feed
Boring	G85	G85	Feed	None	Feed

Operation	G Codes		Spindle Operation		
Canned Cycle	BNC	ISNC	In -Z Direction	At Z Bottom	In +Z Direction
Bore Orient Cycle	G86	-			
Bore Rapid Out	-	G86	Feed	Spindle Stop	Rapid Traverse
Back Boring	–	G87	Feed	Spindle Stop, Spindle Move, Spindle Start	Rapid Traverse
Chip Breaker	G87	–	Peck Feed with Dwell	None	Rapid Traverse
Boring with Manual Feed Out	–	G88	Feed	Dwell	Manual Move, Rapid Traverse
Rigid Tapping	G88	G74 with M29; G 84 with M29; 84.2; or 84.3	Feed	Spindle Stop, Dwell, Spindle Reverse	Feed
Bore with Dwell	G89	G89	Feed	Dwell	Rapid Traverse

Table 2–7. Canned Cycles, G Codes and Z Spindle Operations

These canned cycles are different for BNC than ISNC:

BNC-Specific Canned Cycles	ISNC-Specific Canned Cycles
G84 with M04 Left-Handed Tapping	G74—Left-Handed Tapping
G84 with M03 Tapping	G84—Tapping
G86—Bore Orient Cycle	G86—Bore Rapid Out
G87—Chip Breaker	G87—Back Boring
G88—Rigid Tapping	G74 and G84 with M29— Rigid Tapping
	G88—Boring with manual Feed Out

Table 2–8. BNC and ISNC Specific Canned Cycles

Canned Cycle Parameters

These parameters are used for programming the various canned cycles. They determine the spindle movement. In the pages that follow, the canned cycles are described and the parameters for each one are identified.

Parameter	Description
F	Feedrate
I	Signed, incremental distance from start point to center of spindle shift position (X axis).
J	Signed, incremental distance from start point to center of spindle shift position (Y axis).
K	Number of repeats for a series of operations in a specified block. Range = 1 through 6; Default = 1. If K = 0, drilling data is stored and no drilling is performed. The incremental distance and direction between canned cycles is determined by the previous block's position from the first canned cycle's position. K and L parameters function the same.
L	Number of repeats for a series of operations in a specified block. Range = 1 through 6; Default = 1. If L = 0 drilling data is stored and no drilling is performed. The incremental distance and direction between canned cycles is determined by the previous block's position from the first canned cycle's position. K and L parameters function the same.
P	Dwell time at the bottom of the hole.
Q	Incremental peck depth value or spindle shift distance.
R	BNC: Incremental, positive distance from the Initial Point to Point R. Only used in G99 mode for BNC. ISNC: Represents absolute Z level at which machining begins in either G98 or G99. Must be specified for all ISNC canned cycles.
X	X axis hole position data.
Y	Y axis hole position data.
Z	Defines Z Bottom location. BNC:Always a positive value. In G98 mode:incremental distance down from initial point. In G99 mode:incremental distance down from the R level. ISNC:In G90 mode: absolute Z level. In G91 mode: negative incremental value measured from the R level.

Table 2–9. Canned Cycle Parameters

Depth (Z Parameter)

Z is used to specify the canned cycle's depth. All canned cycles require a Z word. Z Start is the Z level where the negative Z (-Z) axis movement begins. This dimension is the same as the Return to Initial Point in Canned Cycle (G98) and the Return to R Point in Canned Cycle (G99) codes. The Z Bottom parameter is the point of maximum Z down (except for ISNC G88) and the dimension where the -Z axis movement ends.

A rapid move at the Z Start level is automatically used to move from one canned cycle block to another. Make sure the current Z Start level is high enough to clear all fixtures and obstacles.

Note the differences in the definitions for BNC and ISNC Z parameters in the previous table.

- For BNC, the current Z level should be established before invoking the canned cycle (via G00 or G01). Once a Z distance is established, it does not need to be reprogrammed until the canned cycle mode is canceled or changed.
- For ISNC, the Z word represents a negative or positive absolute Z drilling level in G90 mode which must be below the current Z level, or an incremental negative distance from the current R level in G91 mode.

Dwell (P Parameter)

Many of the canned cycles have dwell capability. The scaling factors used with the canned cycle dwell *parameter* P are the same as Dwell, Exact Stop (G04). The length of dwell time is modal and can be specified using one of these methods:

- G04 with a P or X value
- P value with a canned cycle command
- Dwell parameters on the Holes Parameters screen

⇨ Taps use the Bore Dwell parameter.

If you use the default dwell parameters on the Holes Parameters screen, G04 P0.0 or a P0.0 is required with the canned cycle command to cancel any previously commanded dwell time.

Feedrate (F Parameter)

The current feedrate is used for feed moves and may be reprogrammed in any canned cycle block by including an F word. The feedrate parameter applies only to the Z direction during canned cycles.

For BNC files, if no decimal point is included, the system automatically divides the feedrate by 10.

For ISNC files, if no decimal point is included and the Assume Feedrate .1 Increment field on the NC Parameters—Configuration Parameters screen is set to Yes, the system automatically divides the feedrate by 10.

Canceling or Replacing Canned Cycles

All canned cycles are *canceled* by G00, G01, G02, G03, (the One-Shot Group 00 G codes) or G80 (Canned Cycle Cancel).

Current canned cycles can be *replaced* with another canned cycle without first canceling the canned cycle.

If a G00, G01, G02, G03, or G80 occurs in the same block with a canned cycle command (for example G00 G85), the G00 is ignored and the canned cycle command (G85 in this case) is executed. If a G00, G01, G02, or G03 command follows a canned cycle command, the X, Y, Z parameters are used to perform the interpolation or rapid positioning, and the remaining canned cycle parameters in the block are ignored.

All canned cycle data are modal. When a canned cycle is canceled using G00 or G80, the R point, canned cycle repetition value K, and the Q (cut-in, bore shift) are canceled.

Except for tap cycles, canned cycles do not activate the spindle. The program must have a Start Spindle Clockwise (M03) or Start Spindle Counterclockwise (M04) to turn on the spindle prior to executing a canned cycle. For tap cycles, both the spindle speed and direction are retrieved from the tool library if not specified in the program. If a spindle speed is not provided with the M3 or M4, the spindle speed from the tool library is used.

Canned cycles, which turn off the spindle during the cycle, automatically restore the spindle to the original speed and direction before completing the cycle. If a canned cycle requires a certain spindle direction and the opposite spindle direction is currently in effect, the system reverses the spindle direction automatically.

SPINDLE SPEED - S CODES

The Spindle Speed code (S) specifies the spindle rotation speed. The spindle does not rotate until a Start Spindle Clockwise (M03) or a Start Spindle Counterclockwise (M04) is programmed. The software retrieves the spindle speed from the tool library if an S code is not provided.

If the S is present with an M03 or an M04 in the same program block, it is active before the other codes in the program block are executed. If an S is not specified prior to the first M03 or M04, the speed specified in the Tool Setup data is used. As soon as an S appears in the program, its value is used for the M03s and M04s that follow until a newS value is encountered.

For ISNC, if the spindle has already been turned on, the S code is sufficient for changing spindle speed. If the spindle is already turned on and an S code occurs either in a tool change block or in a block following a tool change block, the spindle ramps up to the new spindle speed after the tool change.

TOOL FUNCTIONS

These codes control tool selection: T, L, and D. The L and D codes are for BNC only. To activate these codes, an M06 code must be contained in the same block. To activate the L and D words, an M06 must be used with a T word. The NC Parameters screen contains two fields for controlling tool changes: the Default Tool Number and the M6 Initiates Tool Change.

D Codes

The Tool Diameter Offset codes (D values) are used in ISNC and BNC programs and cause the specified dimension to be loaded into the tool diameter register.

Otherwise, for BNC only, the Diameter value in the appropriate Tool Setup data is used. This dimension is used for cutter compensation, again, only for BNC.

Negative values are not permitted.

L Codes (BNC)

The Tool Length Offset (L) codes cause the specified dimension to be loaded into the tool offset register. Otherwise, the Zero Calibration value in the appropriate Tool Setup data is used.

Negative values are not permitted.

T Codes

The Tool Select (T) codes specify the tool number. The value is composed of up to two digits. Placing the T word in a block does NOT cause a tool change to occur.

If the M6 Initiates Tool Change field is set to Yes, the M06 code must be used to initiate the tool change.

If the M6 Initiates Tool Change field is set to Yes, and a program has a T Code without an M6 Code, the machine will pre-fetch the tool. When this occurs, the tool changer moves the tool carousel so the next tool is ready, but does not complete the change until it encounters the M6.

MISCELLANEOUS FUNCTIONS - M CODES

Miscellaneous Functions (M codes) cause machine-related action (e.g., coolant control and tool changes). Each Miscellaneous Function is explained below. Multiple M codes can be used within an NC block.

M Code Table

M Code	Definition
M00	Cancels the spindle and coolant functions; stops part program execution
M01	Program stop often used when the operator wants to refixture the part
M02	Marks the end of the program; stops the spindle, coolant, and axes feed
M03	Starts clockwise rotation of the spindle
M04	Starts counterclockwise rotation of the spindle
M05	Switches the spindle off
M06	Requests an automatic tool change
M07	Switches on secondary coolant systems
M08	Switches on primary coolant system
M09	Switches off both the primary and secondary coolant
M10	Switches on both the primary and secondary coolant
M12	Clamp Rotary C Axis
M13	Unclamp Rotary C Axis
M20	Advances the indexer one position
M21	Initiates lubrication
M25	Retracts the Z axis to the home position (tool change height)
M26	Select Part Probe Signal
M27	Select Tool Probe Signal
ISNC M29	Enables rigid tapping
M30	Program End
M31	Resets the rotary axis encoder position
M32	Clamps the rotary A axis
M33	Unclamps the rotary A axis
M34	Clamps the rotary B axis
M35	Unclamps the rotary B axis
M36	Switches off the servos
M38	Reads and places the state of the laser OK signal
M39	Reads and places the state of the laser static signal
M40	Reads and places the state of the laser dynamic signal

Table 5–1. M Codes

M Code Table

M Code	Definition
M41	Deactivates two-touch probing when using the G31 command
M42	Enables automatic two-touch probing with the G31 command. If the part probe touches during a G31 move, the probe will automatically back up and then attempt a second touch at a reduced feedrate.
M43	Increases the barrier air.
M44	Reduces barrier air.
M45	Opens the shutter.
M46	Closes the shutter.
M47	Turns the laser emitter on.
M48	Turns the laser emitter off.
M49	Turns the laser receiver on.
M50	Turns the laser receiver off.
M52	Enables auxiliary output 1.
M53	Enables auxiliary output 2.
M54	Enables auxiliary output 3.
M55	Enables auxiliary output 4.
M56	Rotates the pallet changer for a non-confirmation pallet change.
M57	Rotates the pallet changer to pallet 1.
M58	Rotates the pallet changer to pallet 2.
M59	Turns chip conveyor forward mode on.
M60	Turns chip conveyor reverse mode on.
M61	Stops the chip conveyor.
M62	Disables auxiliary output 1.
M63	Disables auxiliary output 2.
M64	Disables auxiliary output 3.
M65	Disables auxiliary output 4.
M68	Enables washdown coolant system.
M69	Disables washdown coolant system.
M76	Normal A Axis operation (default).
M77	Reverses A Axis operation.
M78	Normal B Axis operation (default).
M79	Reverses B Axis operation.
M80	C Axis is right-handed (default).
M81	C Axis is left-handed.
M98	Subprogram call.
M99	Jump; Return from subprogram.

Table 5–2. M Codes

M Code	Definition
M126	Shortest Rotary Angle Path Traverse
M127	Cancels Shortest Rotary Angle Path Traverse (M126)
M128	Tool Center Point Management
M129	Cancels Tool Center Point Management (M128)
M140	Retract Along Tool Vector
M200	Tilt Axis Preference

Table 5–3. M Codes

Program Functions

The Program Functions (M00, M01, and M02) stop the execution of the part programs.

Program Stop (M00)

The Program Stop (M00) cancels the spindle and coolant functions and terminates further program execution after completion of other commands in the same program block. When the program is stopped, existing modal information remains unchanged as in single block operation. The Start Cycle button on the control flashes and this prompt message appears:

Cycle complete; press start to continue.

Pressing the Start Cycle button resumes the spindle and coolant operation and continues the program execution.

This M code should not be set simultaneously with other M codes. M00 is executed following execution of the rest of the address words on the block. Here is an example using the M00 code:

```
N10 G01 X2. Y1. F10. M00
```

In this example, the machine moves to the X2/Y1 location before it shuts down.

⇨ Program blocks should be included that retract the tool to a safe position before a block containing an M00 is programmed. If these program blocks are not included, the spindle stops while cutting the part.

Planned Stop (M01)

The Planned Stop Code (M01) pauses the program and shuts off the spindle. M01 is ignored unless previously validated in the parameter page.

- Include a data block to retract the tool to a safe position before a block containing an M01 is programmed. If the retract tool data block is not included, the spindle will stop while cutting the part.
- If you want to open the CE Safety enclosure doors after M01 executes, press the Machine Mode Interrupt console key, then press the Start Cycle button. The enclosure doors can be opened and the axes jogged. To continue with the program, close the enclosure doors and press the Start Cycle button.

⇨ You can also pause the program and shut off the spindle by selecting the NC Optional Stop On/Off softkey on the Auto Run screen or setting the NC Optional Program Stop parameter on the NC Configuration Parameters screen. Refer to Auto Mode Monitoring in *Getting Started with WinMax Mill* and *NC Parameters, on page 1 - 19* for more information.

End of Program (M02)

The End Of Program code (M02) indicates the end of the main program (the completion of the part), and is necessary for the registration of CNC commands from tape to memory. M02 stops the spindle, the coolant, and the axis feed after completing all of the commands in the program. M02 is active after the block is executed.

⇨ The M02 does NOT stop the NC program loader if the program is loading from a serial link. An E character must be transmitted to signal the loader that the entire program has been sent to the remote device.

⇨ This M code should not be set simultaneously with other M codes unless it is the last M code in the block.

Start Spindle Clockwise (M03)

The Start Spindle Clockwise code starts a clockwise spindle rotation (as viewed from the headstock). The spindle reaches the programmed speed before X, Y, and Z (also A and B if present) axis feed starts. If the spindle speed has not been defined, the Tool Setup screen's spindle speed is used.

M03 is active before the other commands in the block are executed.

Start Spindle Counterclockwise (M04)

The Start Spindle Counterclockwise code starts spindle rotation in a counterclockwise direction (as viewed from the headstock). The spindle reaches the programmed speed before X, Y, Z (A or B) feed starts. If the spindle speed has not been defined, the Tool Setup screen's spindle speed is used.

M04 is active before the other commands in the block are executed.

Spindle Off (M05)

The Spindle Off code is the default and causes the spindle to stop in a normal manner. If the machine is equipped with a brake, it is applied. The coolant is also turned Off.

M05 is active after the other commands in the block are executed.

M6 Initiates Tool Change

Use this field on the NC Parameters screen to indicate whether tool changes are initiated with the M6 or with the T code. Set this field to No and the M6 is ignored and tool changes are initiated whenever a T code is found in the program (not when T is used for user-defined subprogram or subprogram parameter).

If this field is set to Yes, the M6 is required for tool changes.

If this field is set to Yes and a T code is used without the M6, the machine will "pre-fetch" the tool. When this occurs, the tool changer moves the tool carousel so the next tool is ready, but does not complete the change until it encounters the M6.

Change Tool (M06)

The Change Tool code requests that the machine perform a tool change. These tool changes should be performed in rapid traverse mode. The following sequence occurs if an automatic tool changer is present and in the Auto Tool Change mode:

1. The Z axis retracts to tool change position.
2. The machine moves the X and Y axes to the Tool Change position if the tool change position parameter is set to Yes.
3. The spindle orients and stops.
4. The "old" tool is returned to the tool changer.
5. The "new" tool is placed in the spindle.
6. New tool offsets from the Tool Offset screen are loaded into the appropriate registers. The Tool Length Offsets from G43 and G44 remain in effect.
7. The program continues.

⇨ The M06 is optional if the M6 Initiates Tool Change field on the NC Parameters screen is set to Yes; otherwise, tool changes are performed with the T code.

This sequence occurs for manual tool changes:

1. Z axis retracts to its tool change position.
2. The machine moves the X and Y axes to the Tool Change position if the tool change position parameter is set to Yes.
3. The spindle stops and orients.
4. The screen prompts for a tool change.
5. Change the tool and press the Start Cycle button on the control to allow the program to continue.
6. New tool offsets are loaded into the appropriate registers.
7. The program continues.

⇨ The first Z dimension after a tool change must be absolute. Any Z dimension programmed in a tool change block is ignored.

Secondary Coolant On (M07)

The Secondary Coolant On code switches on the mist coolant, if available. M07 is active before the other commands in the block are executed.

Primary Coolant On (M08)

The Primary Coolant On code switches on the flood coolant, if available. M08 is active before the other commands in the block are executed.

Both Coolant Systems Off (M09)

The Coolant Off code is the default and switches off the coolant if it has been activated by Secondary Coolant On (M07) or Primary Coolant On (M08). M09 is active after the other commands in the block are executed.

Both Coolant Systems On (M10)

The Both Coolant Systems On code switches on the coolant if it has been activated by Both Coolant Systems Off (M09).

Clamp C-axis (M12)

The Clamp C axis code clamps the C axis. For C axis moves after M12, the C axis is automatically unclamped for the move and clamped again after the move is complete.

M12 is active before the other commands in the block are executed and is canceled by an Unclamp C axis (M13) command.

Unclamp C-axis (M13)

The Unclamp C axis code unclamps the C axis until an M12 is programmed.

M13 is active before the other commands in the block are executed and is canceled by a Clamp C axis (M12) command.

Oriented Spindle Stop (M19)

The Oriented Spindle Stop code causes the spindle to stop in the oriented position. A brake, if available, will be applied. The coolant is also turned off. This function only applies to machines which have an orient feature. On machines without the orient feature, this function works like the Spindle Off (M05) command.

M19 is active after the other commands in the block are executed.

Pulse Indexer One Increment (M20)

The Pulse Indexer One Increment code advances the *indexer* one position. A reply signal is sent back from the indexer to indicate when it is in position. When the signal is received, the program continues. For multiple indexes, separate M20 blocks must be programmed. (Refer to the indexer's manual and the *Hurco Maintenance Manual* for information on attaching an indexer to the machine.)

M20 is active after the other commands in the block are executed.

Z Axis to Home Position (M25) - Basic NC Programming only

The Z Axis to Home Position code retracts the Z axis to the home position (tool change height) at the rapid traverse rate selected in the Program Parameters screen. The first Z value after an M25 must be absolute.

M25 is active before the other commands in the block are executed.

Select Part Probe Signal (M26)

When G31 is used to invoke probe motion, the machine will move to the specified destination until either the destination is reached or a probe deflection occurs. The Select Part Probe Signal (M26) alerts the G31 move to detect a part probe deflection.

Select Tool Probe Signal (M27)

When G31 is used to invoke probe motion, the machine will move to the specified destination until either the destination is reached or a probe deflection occurs. The Select Tool Probe Signal (M27) alerts the G31 move to detect a tool probe deflection.

Enable Rigid Tapping (ISNC M29)

When Enable Rigid Tapping (M29) is used before a Left-Handed Tapping Cycle (ISNC G74) or Taping Cycle (G84) command, rigid tapping is performed. M29 stays in effect until a One-Shot (G00, G01, G02, G03) code or Canned Cycle Cancel (G80) command is used.

Program End (M30)

The Program End code (M30) indicates the end of the main program (the completion of the part). M30 stops the spindle, the coolant, and the axis feed after completing all of the commands in the program. M30 is active after the block is executed.

Rotary Encoder Reset (M31)

⇨ Rotary Encoder Reset (M31) is functional only with VMX42 SR and VTXU machines.

M31 will reset the rotary-axis (C-axis for the VMX42 SR or VTXU machine) encoder position to lie between -180° and +180°, if it has wound up or down multiple revolutions.

An M31 command can be called in both Conversational and NC part programs. If M31 is executed during contouring operations in a part program, the move prior to the M31 command will come to an exact stop before M31 is executed.

Clamp A-axis (M32)

The Clamp A-axis code clamps the A axis. For A axis moves after M32, the A axis is automatically unclamped for the move and clamped again after the move is complete.

M32 is active before the other commands in the block are executed and is canceled by an Unclamp A axis (M33) command.

Unclamp A-axis (M33)

The Unclamp A axis code unclamps the A axis until an M32 is programmed.

M33 is active before the other commands in the block are executed and is canceled by a Clamp A axis (M32) command.

Clamp B-axis (M34)

The Clamp B axis code clamps the B axis. For B axis moves after M34, the B axis is automatically unclamped for the move and clamped again after the move is complete.

M34 is active before the other commands in the block are executed and is canceled by an Unclamp B axis (M35) command.

Unclamp B-axis (M35)

The Unclamp B axis code unclamps the B axis until an M34 is programmed.

M35 is active before the other commands in the block are executed and is canceled by a Clamp B axis (M34) command.

Servo Off Code (M36)

The servos may be turned off using the Servo Off (M36) command.

Control power to the machine will be turned off. The control will still be powered on. This is similar to an emergency stop.

Laser Input Update (M38-M40)

These codes read the state of the three laser inputs (M38: OK signal;
M39: static signal; and M40: dynamic signal).

Single-Touch Probing (M41)

For a G31 probing move, perform one touch.

Double-Touch Probing (M42)

For a G31 probing move, perform two touches. This is the default mode.

Barrier Air Control (M43 and M44)

Barrier air is used to prevent chips and debris from getting into the laser emitter and receiver. M43 causes the air flow at the probe to increase; M44 reduces the airflow. During operation of the probe, the barrier air should be increased whenever the probe shutter is open. It should remain at the high flow rate except during the actual tool measurement. When the shutter is closed, the flow rate may be reduced.

Shutter Probe Control (M45 and M46)

A pneumatic shutter protects the probe. During a measurement, the barrier air should be increased and the shutter opened. After the probe cycle is completed, the shutter should be closed and the barrier air reduced. M45 causes a brief puff of air that helps clear chips and debris from the probe. M46 closes the shutter.

Laser Emitter On/Off Control (M47 and M48)

M47 turns the laser emitter on. M48 turns the laser off. It is recommended to turn the laser emitter off when not in use.

Laser Receiver On/Off (M49 and M50)

M47 turns the laser receiver on. M48 turns the laser receiver off. It is recommended to turn the laser receiver off when not in use.

Enable Auxiliary Output 1 through 4 (M52 – M55)

M52 through M55 are used to individually enable auxiliary equipment or a unique machine function from within a part program. Enter performance time for the machine-specific M code in the M Code Table. When M52 through M55 are active, the corresponding auxiliary equipment or machine function is turned on, and any performance time is added to estimated run time.

M52 enables Auxiliary Output 1, M53 enables Auxiliary Output 2, M54 enables Auxiliary Output 3, M55 enables Auxiliary Output 4.

Nonconfirmation Pallet Change (M56 – M58)

M56 rotates the pallet changer without regard to position or pallet setup confirmation. M57 rotates the pallet changer to pallet 1. M58 rotates the pallet changer to pallet 2.

The Z-axis automatically moves to zero when a M56, M57 or M58 command is executed.

Chip Conveyor Fwd/Reverse/Stop (M59, M60, M61)

M59 enables chip conveyor forward mode. M60 enables chip conveyor reverse mode. M61 stops the chip conveyor motion.

Disable Auxiliary Output 1 through 4 (M62 – M65)

M62 through M65 turn off auxiliary equipment or machine functions enabled with M codes M52 through M55.

M62 disables Auxiliary Output 1 (M52), M63 disables Auxiliary Output 2 (M53), M64 disables Auxiliary Output 3 (M54), and M64 disables Auxiliary Output 4 (M55).

Washdown Coolant System (M68, M69)

M68 enables washdown coolant system. M69 disables washdown coolant system.

Right Handed C Axis (M80)

When this M code is active and a command is given to the C axis to go in a positive direction, the axis will rotate counter clockwise.

Left Handed C Axis (M81)

When is M code is active and a command is given to the C axis to go in a negative direction, the axis will rotate clockwise.

Subprogram Call (M98)

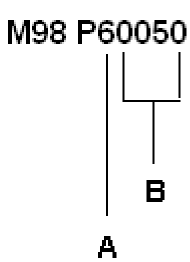
One way of specifying the number of iterations for a subprogram to perform is with M98 subprogram calls.

When making M98 subprogram calls, the P parameter is used to specify iterations as well as the subprogram number. Up to four digits can be used to specify iterations for a maximum of 9999 iterations. Leading zeros are not required when specifying iterations; however, leading zeros are required with a subprogram number that is less than 1000.

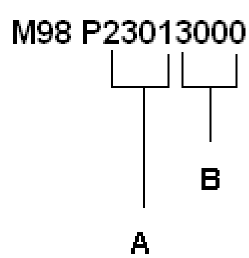
In Example 1 below, M98 P60050 must be used instead of M98 P650 to run program 50 with 6 iterations because the subprogram number (50) is less than 1000.

In Example 2, the M98 P23013000 subprogram example, the four digits to the left (2301) specify the number of iterations, and the four digits to the right (3000) specify the subprogram number.

Example 1



Example 2



A	Number of Items
B	Subprogram Number

As other examples, M98 P1 runs program 1 with no iterations, and M98 P100001 runs program 1 ten times.

Jump; Return from Subprogram (M99)

Each subprogram ends with an *M99* Jump statement.

Shortest Rotary Angle Path Traverse (M126) and Shortest Rotary Angle Path Traverse Cancel (M127)

⇨ Shortest Rotary Angle Path Traverse (M126) and Shortest Rotary Angle Path Traverse Cancel (M127) are functional only with the VMX42 SR machine.

M126 activates Shortest Rotary Angle Path Traverse. The control will move the rotary-axis through the shortest angular distance to the commanded position. M127 cancels the Shortest Rotary Angle Path Traverse.

Example

The table below illustrates machine movement with the M126 command.

Initial Position	Commanded Position	Angular Distance Traverse
350°	20°	+30°
20°	350°	-30°

Table 5–4. Machine Movement with M126 Command

The table below illustrates machine movement with the Shortest Rotary Angle Path Traverse Cancel (M127) command. Standard machine movement for the VMX42 SR is the same as machine movement with the Shortest Rotary Angle Path Traverse cancelled, except when *Tool Vector Input, on page 5 - 15* (G00 or G01) or *3D Tool Geometry Compensation (G41.2), on page 5 - 16* are active.

Initial Position	Commanded Position	Angular Distance Traverse
350°	20°	-330°
20°	350°	+330°

Table 5–5. Machine Movement with M127 Command

Tool Center Point Management (M128) and Tool Center Point Management Cancel (M129)

The Tool Center Point Management feature allows programming of 5-axis tool positions in the Workpiece Coordinate System, independent of the Part Setup location in the machine. *Refer to Getting Started with WinMax Mill*.

M128 activates Tool Center Point Management and M129 cancels Tool Center Point Management.

> G00, G01, G02, and G03 moves are supported in M128 mode.
>
> NC Hole cycles are permitted with M128 activated, only if the Tool Vector is vertical.
>
> Both G93 (Inverse Time) and G94 (UPM) are supported when M128 is active.

Three input modes are available with M128:

- *Axes Angle Input, on page 5 - 14* (G00 or G01): Tool Bottom Centerpoint (X_Y_Z_) and Axes Angle Input (B_C_)
- *Tool Vector Input, on page 5 - 15* (G00 or G01): Tool Bottom Centerpoint (X_Y_Z_) and Tool Vector (I_J_K_)
- *3D Tool Geometry Compensation (G41.2), on page 5 - 16*: Surface Contact Point (X_Y_Z_) and Tool Vector (I_J_K_) and Surface Normal Vector at contact point (U_V_W_)

Axes Angle Input

Allows the Operator to specify the Tool Bottom Centerpoint with respect to the Workpiece Coordinate System, and the rotary and tilt axes relative to the Unrotated Coordinate System. *Refer to Getting Started with WinMax Mill*.

Format

G01 X_Y_Z_B_C_

{X, Y, Z} is the Tool Bottom Centerpoint.

{B, C} are the rotary and tilt angles relative to Part Setup. B and C are modal.

Tool Vector Input

Tool Vector Input is functional only with the VMX42 SR machine.

⇨ Tool Vector Input is only available if M128 or NC Transform Plane is active.

A vector is a direction in 3D space that is defined using values for the X, Y, and Z direction components, as shown in the figure below. Since vectors describe a direction, their base is always at the Coordinate System origin from which they point outward. Tool Vector is the vector that describes the orientation of the tool axis; the direction from the tool tip pointing up through the spindle and away from the workpiece.

⇨ The Tool Vector can be specified with up to six decimal places. It is highly recommended that the full precision be used. Field values will normally lie in the range of -1.000 000 to +1.000 000. If the magnitude of the vector does not equal 1, the vector will be normalized by the CNC.

The magnitude of a vector is determined by using the following equation: $magnitude = \sqrt{I^2 + J^2 + K^2}$

The drawing below is a plot of a vector with its , Â I, J, and K components that correspond to the X, Y, and Z directions.

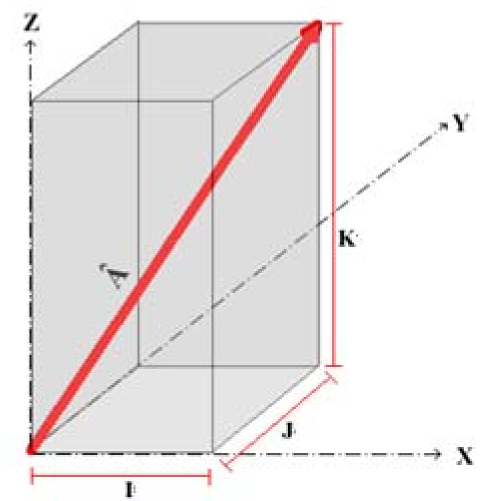


Figure 5–1. Plot of a Vector

When the NC program contains Tool Vectors in the tool position blocks with M128 or NC Transform Plane active, the CNC will compute the appropriate rotary and tilt axes (B-axis and C-axis) positions. The Tool Tip location in the part program specifies where on the workpiece the tool should be positioned. The CNC computes the X, Y, and Z machine axes positions to move the rotated tool tip to the specified point on the rotated workpiece.

⇨ The Tool Tip and Tool Vector are specified with respect to the Workpiece Coordinate System defined in the CAM software or the NC Transform Plane. The CNC will automatically compute the machine axes positions using the Tool Length and the Part Setup information. *Refer to Getting Started with WinMax Mill.*

Format

G01 X_Y_Z_I_J_K_

{X, Y, Z} is the Tool Bottom Centerpoint. X, Y, and Z are modal.

{I, J, K} is the Tool Vector. I, J, and K are non-modal.

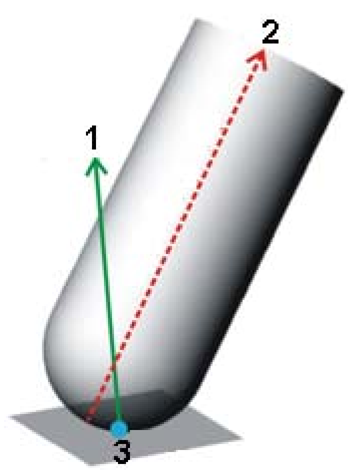
⇨ Both G93 (Inverse Time) and G94 (UPM) are supported with Tool Vector Input.

3D Tool Geometry Compensation (G41.2)

⇨ 3D Tool Geometry Compensation (G41.2) is functional only on the VMX42 SR machine.

M128 must be active when using G41.2.

G41.2 allows specification of the Surface Contact Point, the Surface Normal Vector, and the Tool Vector. The CNC will compute the tool position automatically for ball nose, flat end, and bull nose endmills. The tool will be positioned to tangentially touch the specified Surface Contact Point. The following figure shows the Surface Normal Vector, Tool Vector and Surface Contact Point for a tool.



1	Surface Normal Vector
2	Tool Vector
3	Surface Contact Point

Figure 5–2. Tool Components for 3D Tool Geometry Compensation

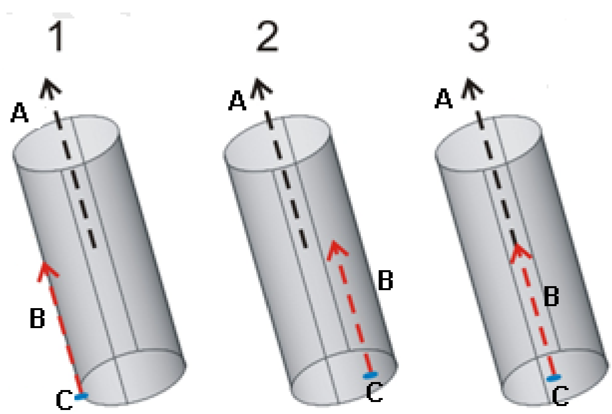
Although ball, flat, and bull nose endmills can be positioned interchangeably with a G41.2 command, there is no guarantee that the selected tool dimensions and geometry will not cause gouging of the part.

It is the responsibility of the Operator to ensure that the tool path is gouge-free for the selected tool.

The Surface Contact Point, Surface Normal Vector, and Tool Vector are specified with respect to the Workpiece Coordinate System defined in the CAD/CAM software or part drawing. The CNC will automatically compute the machine axes coordinates using the tool dimensions and Part Setup information. Refer to *Getting Started with WinMax Mill*.

Tool Vector and Surface Normal Vector can be specified with up to six decimal places. It is highly recommended that the full precision be used. Field values will normally lie in the range of -1.000 000 to +1.000 000. If the magnitude of the vector does not equal 1, the vector will be normalized by the CNC.

For flat and corner radius endmills, there are infinite solutions for the tool position when Tool Vector and Surface Normal Vector point in the same direction and any point on the bottom face of the tool can touch the Surface Contact point (shown by #1 and #2 in the following figure). When this condition exists, the CNC will place the Tool Bottom Center on the Surface Contact Point, as shown by #3 in the figure below.



A	Tool Vector
B	Surface Normal Vector
C	Surface Contact Point

Figure 5–3. 3D Tool Geometry Compensation Infinite Solution Examples

- G41.2 requires the radius and corner radius to compute tool positions. Refer to the following figure for tool geometry.
- G41.2 D_R_ specifies the Tool Radius (D_) and the Corner Radius (R_) for both the ISNC and Hurco Basic NC dialects. The values in D_ and R_ are indexes for the Tool Radius Offset Table and Tool Corner Radius Offset Table, respectively.

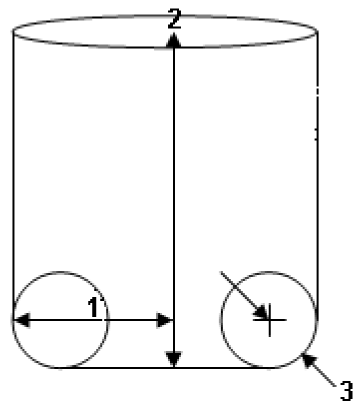


Figure 5–4. Tool Geometry

Format

G41.2 uses G01 for contouring moves.

G40 cancels G41.2

G01 X_Y_Z_I_J_K_U_V_W_, where all coordinates are specified in the Workpiece Coordinate System. Refer to *Getting Started with WinMax Mill*.

{X, Y, Z} is the part's Surface Contact Point and is modal.

{U, V, W} is the Surface Normal Vector of the part contact point and is non-modal.

{I, J, K} is the Tool Axis Vector and is non-modal.

3D Tool Geometry Compensation (G41.2) is well-suited for using with a ball nose endmill. The potential for surface gouging exists when using the G41.2 command with flat and corner radius endmills.

Retract Along Tool Vector (M140)

⇨ Retract Along Tool Vector (M140) is functional only on the VMX42 SR machine.

M140 allows the Operator to move the tool along the current Tool Vector for a specified distance or to retract to machine limits. In the drawing below, positive Retract Along Tool Vector is in the direction of the arrow (which will move the tool away from the part).

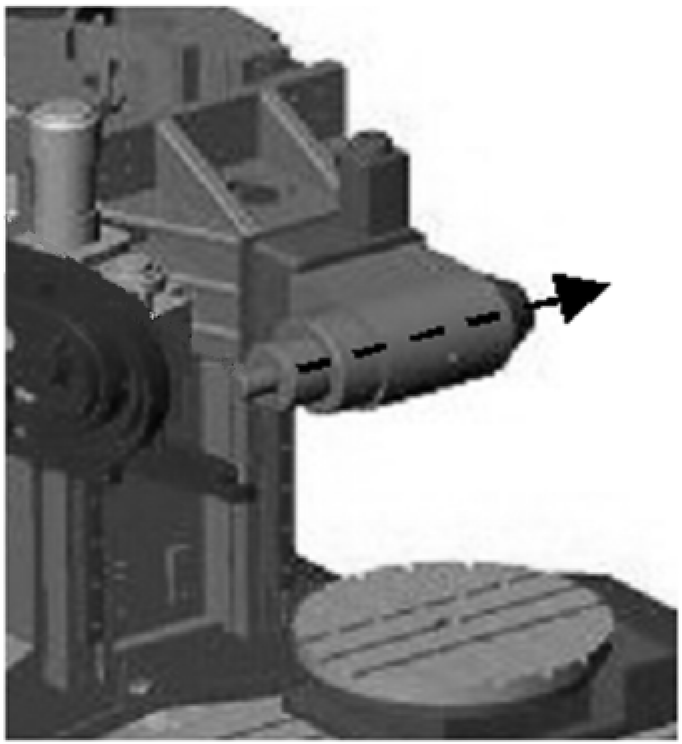


Figure 5–5. Positive Retract Along Tool Vector

Format

M140 is non-modal and is only active for the current block.

M140 [L_] , where L_ is the incremental distance the tool will move from its current position along the current Tool Vector direction.

- A positive L_ value will move the tool in the direction pointing from the cutting tool bottom up through the spindle (i.e., moves the tool away from the part). Typically, a positive L_ value will be programmed.
- A negative L_ value will move the tool in the opposite direction.
- When M140 is used without the L_ parameter, the tool will retract along the positive direction of the current Tool Vector to the machine limits (i.e., move the tool tip away from the workpiece).

Tilt Axis Preference (M200)

> ⇨ Tilt Axis Preference (M200) is functional only on the VMX42 SR machine.

To improve the work volume when the spindle is horizontal, the C-axis table on the VMX42 SR is installed at the corner of the machine's base table. Although this configuration provides a large work volume for negative B-axis angles, it restricts the work volume for positive B-axis angles.

However, this configuration does not impose a limitation of the parts that can be cut on a VMX42 SR machine. Parts that require positive B-axis angles can be cut using a negative B-axis angle along with a rotation of 180° of the C-axis table.

> ⇨ Since positive B-axis angles have a restricted work volume for 3+2-axis machining, the Operator may request that the post processor set the B-axis limits from -90° to 0° to prevent using positive B-axis angles.

Tilt Axis Preference is only applied when all of these conditions exist:

- Tool Center Point Management (M128) command is active.
- Tool Bottom Centerpoint with *Tool Vector Input, on page 5 - 15* are used.
- G43.4 command is used with Tool Tip and Tool Vector interpolation active.

When the three conditions above exist, the machine will force the B-axis to remain in the specified Tilt Axis Preference region, if the tool path is within machine limits.

If the B-axis is requested to move to the opposite side of the Tilt Axis Preference region, the CNC will interpolate the Tool Tip and Tool Vector up to the machine singularity point (B-axis at 0°). The machine will then rotate around the singularity point (i.e., the CNC will rotate the C-axis and interpolate the X and Y axes, while keeping the Tool Tip at a constant location relative to the workpiece, within the control's tolerance parameters), followed by interpolating the B-axis and Tool Tip to their final positions with the B axis on the Tilt Axis Preference side.

> ⇨ Moving the B-axis to the opposite side of the Tilt Axis Preference region (described above) may leave a dwell mark on the workpiece.

Format

M200 P0—Neutral Tilt Axis Preference. This setting is used to turn off the current Tilt Axis Preference and the program will execute the Shortest Angular Traverse.

M200 P1—Positive Tilt Axis Preference. The B-axis is kept between 0° and +90°.

M200 P2—Negative Tilt Axis Preference; the B-axis is kept between -90° and 0°.
M200 P2 is the default setting for an NC program.

NC PRODUCTIVITY PACKAGE OPTION

The NC Productivity Package (NCPP) option provides features that enhance productivity and aid in producing smaller, more powerful, and easier to maintain NC programs. NCPP features include variables, subprogram calls, macros, user-defined codes, mathematical equations and address expressions. The NCPP option requires the presence of the ISNC option.

⇨ NC files that are larger than dynamic RAM memory can be serially loaded to the hard disk. The CNC can run NC files that do not entirely fit into dynamic RAM memory.

Macro Modes

The CNC software provides compatibility between different NC dialects from various machine tool control manufacturers. The software calls NC macros (Macro Mode A or Macro Mode B) to be compatible with existing NC macros.

Older NC macros use the Macro A method of calling subprograms. The main difference between the two macro modes is Macro Mode A does not provide for local (general purpose) variables within a subprogram. Also, Macro Mode B provides the potential to embed more NC computer programming. The table below identifies each macro mode's variables and the functions for which the variables are used. (Refer to the "Variables" section for more information about local variables.)

Functions	Subprogram Variables	
	Macro Mode A	**Macro Mode B**
Local Variables	None	#1-33
Tool Offsets	#1-#99	#2001-#2200
User defined M Codes	9001-9003	9001-9003; 9020-9029
Indirect Variable Referencing	#9100	#[#100]
Pass Subprogram Parameters	#8004-#8026; #8104-#8126	#1-#33
G Code Status	#8030-#8046; #8130-#8146	

Table 6–1. Subprogram Variables

To enable the appropriate macro mode, press the NC Parameters (F3) softkey on the Program Parameters screen. The NC Parameters— Configuration Parameters screen appears with the cursor in the upper left-hand corner at the default Macro Mode B Yes field.

Enable Macro Mode B by selecting the Yes (F2) softkey; enable Macro Mode A by selecting the No (F1) softkey.

- Macro Mode A contains 3 program numbers (9001–9003).
- Macro Mode B contains 13 program numbers (9001–9003 and 9020–9029).

Refer to *User Defined Codes, on page 6 - 35* for more information about user defined codes.

The user can assign G or M codes in the appropriate column on the NC Parameters—M and G Code Program Numbers screen for each program number.

Variables

Variables are used to create programs that can be easily modified. Programs with variables can be reused for various applications. All variables must begin with the "#" character followed by a valid, "writeable" register number and an equal sign.

The example that follows sets the variable value (#500) to 110:

500 = 110.

There are four types of variables that can be used in NC programming: global, system, local, and arguments. Arguments and local variables are only available in Macro Mode A. Some variables are read only and an error is generated when an attempt is made to write to the variable.

Global Variables

Global variables are general purpose variables that can be used by all programs. Assign a value to the global variable before it is used in an equation or expression, or the variable will be considered vacant. An error message is generated when the system attempts to read a vacant variable.

If the value of a global variable is changed in a program, all other programs can reference that variable with the new value.

Global variables range between #100 to #199 and #500 to #999.

System Variables

System variables are predefined variables that provide information about the state of the system such as X, Y, Z, external work compensation, miscellaneous system parameters, modal information, position information, and G code group status.

For instance, the coordinates of a probe touch are saved to variables #5061, #5062, and #5063 when using the G31 command. These variables contain information about the probe's location when the probe touch occurs.

Macro Mode A Local Variables

Local variables are general purpose variables that are only valid within the current program. They are only available in Macro Mode A and range from #1 through #33. Assign a value to the local variable before using it in an equation or expression, or it will be considered vacant. An error message is generated when the system attempts to read a vacant variable.

These variables are nested, meaning that when a subprogram call is made, a new set of local variables is received and the old set is stored. After leaving the subprogram, these local variables are destroyed and the previous set is restored.

Passing parameters to subprograms automatically initializes local variables when subprogram calls other than M98 are made. Refer to *Passing Single Dedicated*

Parameters to Subprograms, on page 6 - 36 for more information.

Macro Mode A Arguments

Parameters are the addresses which follow G65, G66, and M98. Arguments include the parameter's G group status, and they are used to pass parameters to subprograms. In the table below, the subprogram numbers listed in the Value column contain the code variable or G group modal status, and the subprogram numbers in the Status column contain the status of corresponding values. Notice that these arguments are read only.

Macro Mode A Parameters

In the table below, the parameters' values (I, J, K,.....Z) are stored in addresses #8004 to #8026 for Macro Mode A subprogram calls. The status for each variable is stored in addresses #8104 to #8126. The status for the variables is non-zero (≥1) if an argument is specified in the subprogram call, and zero otherwise.

Macro Mode A Subprogram Parameters				
	Value of		**Status of**	
I	#8004	ARG	#8104	R
J	#8005	ARG	#8105	R
K	#8006	ARG	#8106	R
F	#8009	ARG	#8109	R
G	#8010	ARG	#8110	R
H	#8011	ARG	#8111	R
M	#8013	ARG	#8113	R
N	#8014	ARG	#8114	R
P	#8016	ARG	#8116	R
Q	#8017	ARG	#8117	R
R	#8018	ARG	#8118	R
S	#8019	ARG	#8119	R
T	#8020	ARG	#8120	R
X	#8024	ARG	#8124	R
Y	#8025	ARG	#8125	R
Z	#8026	ARG	#8126	R

Table 6–2. Macro Mode A Subprogram Parameters

Macro Mode A G Code Groups

The value for each G Code Group is stored in addresses #8030 to #8046 for Macro Mode A subprogram calls G65, G66, and user defined G and M Codes. The status is stored in addresses #8130 to #8146. The status is non-zero if an argument is specified in the subprogram call, and empty otherwise.

Macro Mode A G Code Group Status				
	Value of		Status of	
00	#8030	ARG	#8130	R
01	#8031	ARG	#8131	R
02	#8032	ARG	#8132	R
03	#8033	ARG	#8133	R
05	#8035	ARG	#8135	R
06	#8036	ARG	#8136	R
07	#8037	ARG	#8137	R
08	#8038	ARG	#8138	R
09	#8039	ARG	#8139	R
10	#8040	ARG	#8140	R
11	#8041	ARG	#8141	R
15	#8045	ARG	#8145	R
16	#8046	ARG	#8146	R

Table 6–3. Macro Mode A G Code Group Status

Read/Write Restrictions

Read only variables are fixed values. You can change write only variables. Some variables within NCPP are read only (R), some are write only (W), and others are read/write (R/W). Most variables can be used to store either real variables or 32 bit binary values, and the software performs the appropriate conversions when the variables are used within equations. The types of variables are identified as follows: Argument (A), Global (G), Local (L), and System (S).

This table lists the NCPP variable types and read/write restrictions.

Variable Number	Type	Restriction	Variable Number	Type	Restriction
#1 to #33	L	R/W	#4309	S	R
#1 to #99	S	R/W	#4311	S	R
#100 to #199	G	R/W	#4313	S	R
#500 to #999	G	R/W	#4314	S	R
#2000	S	R	#4315	S	R
#2001 to #2200	S	R/W	#4319	S	R
#2500	S	R/W	#4320	S	R
#2501 to #2506	S	R/W	#5001 to #5004	S	R
#2600	S	R/W	#5021 to #5023	S	R
#2601 to #2606	S	R/W	#5041 to #5043	S	R
#2700	S	R/W	#5061 to #5063	S	R/W
#2701 to #2706	S	R/W	#5081 to #5083	S	R
#3000	S	R/W	#8004	A	R/W
#3004	S	R/W	#8005	A	R/W
#3005	S	R	#8006	A	R/W
#4001 to #4021	S	R	#8009	A	R/W
#4022	S	R	#8010	A	R/W
#4102	S	R	#8011	A	R/W
#4107	S	R	#8013	A	R/W
#4109	S	R	#8014	A	R/W
#4111	S	R	#8016	A	R/W
#4113	S	R	#8017	A	R/W
#4114	S	R	#8018	A	R/W
#4115	S	R	#8019	A	R/W

Variable Number	Type	Restriction	Variable Number	Type	Restriction
#4119	S	R	#8020	A	R/W
#4120	S	R	#8024	A	R/W
#4201 to #4221	S	R	#8025	A	R/W
#4222	S	R	#8026	A	R/W
#4302	S	R	#8030	A	R
#4307	S	R	#8031	A	R
#8032	A	R	#8117	A	R
#8033	A	R	#8118	A	R
#8035	A	R	#8119	A	R
#8036	A	R	#8120	A	R
#8037	A	R	#8124	A	R
#8038	A	R	#8125	A	R
#8039	A	R	#8126	A	R
#8040	A	R	#8130	A	R
#8041	A	R	#8131	A	R
#8045	A	R	#8132	A	R
#8046	A	R	#8133	A	R
#8104	A	R	#8136	A	R
#8105	A	R	#8137	A	R
#8106	A	R	#8138	A	R
#8109	A	R	#8139	A	R
#8110	A	R	#8140	A	R
#8111	A	R	#8141	A	R
#8113	A	R	#8145	A	R
#8114	A	R	#8146	A	R
#8116	A	R			

Table 6–4. NCPP Variable Types and Read Write Restrictions

Addresses with Variables

NC blocks contain addresses with specific numbers. Variables can be used in place of numbers for addresses in the NC blocks, making the program generic. The example below uses variables in the block's address instead of the numbers they represent:

Number	Variable
0.00	#110
-10.00	#115
1.00	#120
0.25	#121
12.00	#122

Address with Variables

G#110 X[#122+.3] Y-[#115/5.] Z[#120 + #121]

Address with Numbers

The same address would be written as follows if numbers were used instead of variables:

G0.00 X[12.00 + .3] Y-[10.00/5] Z[1.00 + 0.25]

–Or–

G0.00 X12.3 Y2 Z1.25

Alarm 3000 Messages

Variable #3000 writes an Alarm 3000 error message to the screen. The following is an example of this type of error message:

#3000 = 140 (ARGUMENT MISSING)

The right-hand side of the equation must begin with a number in the range of 0 to 200 followed by a left parenthesis, a string which is limited to 26 characters, and a right parenthesis. This number is added to 500 and stored to variable #3000. The message "ARGUMENT MISSING" is displayed on the screen.

Vacant Variables

A variable is considered vacant if a local or global variable has not been assigned a value before it is used in an equation or expression. An error message occurs with vacant variables.

A variable can be tested to determine if it is vacant by comparing it with the null variable #0. The variable #0 is called the "null variable" because it cannot be used to store a value and is only used to perform vacant variable tests.

For example, the following IF conditional statement is true if variable #510 is vacant and false if the variable is not vacant. (Refer to the “IF Statements” section of this chapter for information about IF statements.)

```
IF[#510 EQ #0] GOTO 100
```

⇨ The function NE (not equal) can also be used with vacant variables.

It is best to avoid using vacant variables in equations. However, when it is necessary to use them to maintain compatibility with existing programs, vacant variables can be used in some circumstances without receiving an error message.

The following table shows what happens when vacant variables are used in equations versus setting variables to zero. This table shows the difference between using vacant variables and setting variables to 0 in equations:

Comparison of Vacant Variables and Setting Variables to Zero (0)			
Function	**Examples**	**Null / Vacant Variable (#10 = <vacant>)**	**Variable Set to 0 (#10 = 0)**
Assignment	#20 = #10	Error Message	#20 = 0
Multiplication	#20 = #10 * 3	#20 = 0	#20 = 0
	#20 = #10 * #10	Error Message	#20 = 0
	#20 = #10 * #0	Error Message	Error Message
	#20 = #0 * 3	Error Message	-
	#20 = #0 * #0	Error Message	-
Addition	#20 = #10 + 3	#20 = 0	#20 = 0
	#20 = #10 + #10	Error Message	#20 = 0
	#20 = #10 + #0	Error Message	Error Message
	#20 = #0 + 3	Error Message	-
	#20 = #0 + #0	Error Message	-
EQ (equal)	#10 EQ #0	True	False
	#10 EQ 0	Error Message	True
NE (not equal)	#10 NE #0	False	True
	#10 NE 0	Error Message	False
GE (greater than or equal to)	#10 GE #0	True	False
	#10 GE 0	Error Message	True
GT (greater than)	#10 GT #0	Error Message	Error Message
	#10 GT 0	Error Message	False
Other Functions	-	Error Message	Depends on Function

Table 6–5. Comparison of Vacant Variables and Setting Variables to Zero (0)

Variable Expressions

Instead of using a number after an NC parameter, a variable expression (or math expression) can be used.

- The “[”and the “]” characters serve as delimiters in the expressions.
- A negative sign entered before the left bracket ([) indicates that the expression is negative (i.e. X-[[#110+3.4] + 4.5]).

Expression Symbols and Keywords

Various keywords and symbols can be used in the expressions. At least two letters of the keyword are required: RO, ROU, ROUN, and ROUND perform the same function. The software checks spelling. RUON is not a valid abbreviation for ROUND, but ROUN is acceptable.

Symbol	Description	Example
+	Addition	#500 = #600 + 2.3
-	Subtraction	#500 = #600 - 2.3
/	Division	#500 = #600 / 2.3
*	Multiplication	#500 = #600 * 2.3
^	Power (i.e. 2^2, 2 to the 2nd power, or 4)	#500 = 4.5 ^ 2 #500 will be set to 20.25.

Table 6–6.   NC Expression Symbols

The keywords are described and examples are provided in the following table:

Operation Keyword	Description	Example
ABS	Absolute Value	#500 = ABS [-#550]
ACOS	Arc or Inverse Cosine function	#500 = ACOS [#540]
AND	Logical AND	#500 = #600 AND 48
ASIN	Arc or Inverse Sine function	#500 = ASIN [#540]
ATAN	Arc Tangent (degrees)	#500 = ATAN [.34]
BCD	Convert Binary to BCD Format	#500 = BCD [#600]
BIN	Convert BCD to Binary Format	#500 = BIN [#600]
COS	Cosine (degrees)	#500 = COS [45.3]
DEGREES	Converts radians to degrees	#500 = DEGREES [5.437] #500 will be set to 311.52 degrees.
EQ	Equal	#500 = #510 EQ 3.4 #500 will be set to 0 if false; 1 if true.
EXP	Exponential function	#500 = EXP [3.67] #500 will be set to 39.252.
FIX	Discards fractions less than 1	#500 = FIX [45.2375] #500 will be set to 45
FUP	Adds 1 for fractions less than 1	#500 = FUP [45.2375] #500 will be set to 46
GE	Greater Than Or Equal To	#500 = #510 GE 3.4 #500 will be set to 0 if false; 1 if true.
GT	Greater Than	#500 = #510 GT 3.4 #500 will be set to 0 if false; 1 if true.
HSIN	Hyperbolic Sine function	#500 = HSIN[#540]
HCOS	Hyperbolic Cosine function	#500 = HCOS [#540]
INVERSE	Binary Inverse function	#500 = [7 AND [INV[3]]] #500 will be set to 4.0.
LE	Less Than or Equal To	#500 = #510 LE 3.4 #500 will be set to 0 if false; 1 if true.
LN	Natural Logarithmic function	#500 = LN [24.89] #500 will be set to 3.2144.

Operation Keyword	Description	Example
LOG	Logarithmic function	#500 = LOG [345.89] #500 will be set to 2.5389.
LT	Less Than	#500 = #510 LT 3.4 #500 will be set to 0 if false; 1 if true.
MOD	Modulus operator	#500 = 19 MOD 6.7 Will return a value of 5.6
NE	Not Equal	#500 = #510 NE 3.4 #500 will be set to 0 if false; 1 if true.
OR	Logical OR	#500 = 41 OR 4
RADIANS	Converts degrees to radians	#500 = RADIANS [270.34] #500 will be set to 4.718 radians.
ROUND	Rounds off	#500 = ROUND [34.56 result is 35]
SIN	Sine (degrees)	#500 = SIN [#610]
SQRT	Square Root	#500 = SQRT [#540]
TAN	Tangent (degrees)	#500 = TAN [32.4]
XOR	Logical Exclusive OR	#500 = #560 XOR 34

Table 6–7. NC Expression Keywords

The software automatically converts real numbers to hexadecimal format before performing logical operations. The Operation Keyword “AND” does not function on real numbers. As shown below, the #500 value is truncated to 32 and the #550 value is truncated to 48. When the “AND” function is performed, the truncated numbers are stored in variable #560.

- #500 = 32.456
- #550 = 48.98
- #560 = [#500 AND #550]

These examples are valid variable expressions:

- G01 X#140 Y [#500 + 2.] Z[#550 * [SIN [#130 + 23.5]]]
- G02 Z [2.3 / [SIN 43]] Y[2 ^ 3] G20 M25
- X [ROUN[3.45 * COS[#520]]]
- R [SQRT[[#510 ^ 2] + [#511 ^ 2]]]
- G01 X-#510 Y-[#520 + 4.5] Z4

Operation Priorities

The interpreter gives operations within the expression a certain priority in order to determine how the expression is evaluated. This is a listing of the priorities:

Priority	Operation	
Highest	Functions	
Second	Symbols	Power (^)
Third		Multiplication (*) Division (/)
Lowest		Addition (+) Subtraction (-)

Table 6–8. Numerical Operations Priorities

Even though the interpreter assumes this priority, in order to make the NC program more understandable and more maintainable, use brackets to divide the expressions. For example, G01 X[34.5+23.4 / 32] should be rewritten as G01 X[34.5 + [23.4/32]]. Using spacing within an expression can also make the expression more readable. Decimal points and leading or trailing zeros are not required with the numbers.

Indirect Variables

Variables can be referenced indirectly by using multiple levels of pound signs (#) and brackets ([and]).

#100 = 600⇐ #100 is equal to 600.

#600 = 4.5 ⇐ #600 is equal to 4.5.

#[#100] = 4.5⇐ #[#100] is equal to #600; #600 equals 4.5.

Macro Mode A variables are referenced indirectly by using a “9” as the first number:

In #9500, #9 is the address of the value at 500, which is the same as using #[#500] in Macro Mode A or Macro Mode B.

Saving Variable Values To a File on the Control

When running the program on the CNC, if an error occurs during the program run, the variable values are not saved. The variable values are saved if the program runs successfully.

Variable Example

This program illustrates the use of #0 in an IF statement to determine if an argument is passed to subprogram 3100. There are two IF statements in sequence numbers 100 and 200 in the subprogram which test to verify that the calling program (0100) had passed parameters I and J which correspond to #4 and #7 in subprogram 3100, respectively. If either variable #4 or #7 is vacant, an Alarm 3000 error message is written to the screen. (Refer to the “Program Control Statements” section for more information about IF statements.)

ISNC Part Program 1 Inch
TRU_CRC.FNC

```
%
O0100                                   ⇐ Calling Program—0100—Start
T01 M06
S1500 M03
G00 G90 X5.0 Y5.0
G43 Z.1 H01
M08
G01 Z-.5 F5.0
G65 P3100 I.5 D2 F15.0
G00 Z.1 M09
G91 G28 Z0 M05
M30                                     ⇐ End of Program—0100
:3100(True CIRCLE TYPE 1)               ⇐ Subprogram—3100—Start
#27 = #4001
#28 = #4003
#29 = #4107
N100 IF[#4EQ#0] GOTO 1000               ⇐ Vacant Variable Check
N200 IF[#7EQ#0] GOTO 1000
#1 = ABS [#4]-ABS [#[2000+#7]]
IF [#1LE0] GOTO 2
#20 = #1/2
#21 = ROUND [#20*1000]
#22 = #21/1000
#2 = #1-#22
#3 = #1-#2
IF [#23EQ#0] GOTO 10
G01 G91 X-#2 Y-#3 F#9
G17 G02 X-#3 Y#3 J#3
```

```
I#1
X#3 Y#3 I#3
G01 X#2 Y-#3 F[#9*3]
GOTO 5
N10 G01 G91 X-#2 Y#3 F#9
G17 G03 X-#3 Y-#3 J-#3
I#1 J0
X#3 Y-#3 I#3
G01 X#2 Y#3 F[#9*3]
GOTO 5
N1000 #3000 = 100(ARGUMENT MISSING)  ⇐ Alarm Message
N5 G#27 G#28 D#29
M99                                  ⇐ M99 is end of Subprogram 3100
```

View the part using the Draw console key to verify that the part is programmed correctly.

Program Control Statements

Program control statements are NC blocks which direct the flow of the NC program or subprogram. The following section describes using the different NCPP option's program control statements.

Program control statements use keywords: GOTO, IF, WHILE, and DO. At least two letters of the keyword are required. For example, WH, WHI, WHIL, and WHILE all perform the same function. Some program control statements are only effective within the current program or subprogram, and other program control statements cause program execution to go to subprograms. The software can only locate sequence numbers that are in memory.

The following program control statements are effective only within the *current program* being executed:

- WHILE [conditional expression] DO#
- DO#
- IF [conditional expression] GOTO [expression or #]
- GOTO [expression or #]
- END#
- M99 or M99 P____

These program control statements cause program execution to *call subprograms*:

- M98 P____
- G65 P____ L____ [Optional Argument List]
- G66 P____ L____ [Optional Argument List]
- User defined G code followed by [Optional Argument List]
- User defined M code followed by [Optional Argument List]
- User defined B, S, and T codes followed by optional parameter

Variables can be *referenced indirectly* to initialize a large block of variables, for example:

- #100 = 500
- WHILE [#100 LT 1000] DO 250
- #[#100] = 1.5
- #100 = #100+1
- END 250

The alternative to indirectly referencing variables is to have a program line for each variable as shown below:

- #500 = 1.5
- #501 = 1.5
- ...
- #999 = 1.5

In this case, 500 program lines would be required to perform what five program lines accomplished in the first example.

GOTO Statements

GOTO statements jump the program to a specific number in the program. Any valid address expression can be used in place of a sequence number after the GOTO. Fractions are truncated. For example, GOTO 3.45 and GOTO 3 work the same. The program cannot locate sequence numbers that are not in memory. If the search reaches the end of the program without finding the sequence number, the software generates an error message.

Positive GOTO Statement

If the resultant value is positive, the software searches for the sequence number from the point of the GOTO to the end of the program. Then it proceeds to the beginning of the program and searches for the sequence number until reaching the starting point (GOTO statement).

Negative GOTO Statement

If the resultant value of the expression is negative, the search begins at the beginning of the program.

IF Statements

IF statements contain a conditional expression and a GOTO statement. The expression which follows the GOTO must result in a valid sequence number; otherwise, an error message is generated. The program cannot locate sequence numbers that are not in memory. The following line illustrates an IF statement's components:

- IF [conditional expression] GOTO [expression or #]
- If the conditional expression has a value of 1, it is true, and the GOTO is performed.
- If the conditional expression has a value of 0, it is false, and the next NC block is executed.
- If the conditional expression has a value other than 0 or 1, it is invalid.

These are examples of conditional expressions used in IF statements:

```
IF[[[#100 LT 2.3] OR [#320LE7.34]] AND [#400LT3.4]] GOTO#340
IF[#150 EQ 2] GOTO 10
IF[#750 GT 2.34] GOTO [[#550+23]/40]
```

WHILE Loops

WHILE loops contain a conditional expression and a DO statement. This is a sample WHILE loop:

- WHILE [conditional expression] DO number
- NC block
- NC block
- NC block
- END number

The blocks between the WHILE statement and the END statement are repeated as long as the conditional expression is true. The following are other details about WHILE loops:

- A WHILE loop must have a matching END statement within the same program.
- The DO must match the number following END and must be an integer in the range of 1 to 255.
- The program cannot locate sequence numbers that are not in memory.
- No other NC commands can be contained on the same lines as the WHILE or END statements.
- If the WHILE conditional expression is false, the program continues execution with the NC block which follows the END statement.
- DO loops operate the same as WHILE loops with a conditional expression which is always true.
- The DO statement can also be used by itself without the WHILE conditional statement.

⇨ To exit an infinite WHILE loop while the program is being drawn, press the console Draw key.

DO Loops

DO loops operate the same as WHILE loops with a conditional expression which is always true. The DO statement can also be used by itself without the WHILE conditional statement. The following are some additional details about DO loops:

- DO loops must contain a matching END statement within the same program.
- The numbers following DO and END must match and must be an integer in the range of 1 to 255.
- The program cannot locate sequence numbers not in memory.
- No other NC commands can be contained on the same lines as the DO or END statements.

The following is a sample DO loop:

- DO number
- NC block
- NC block
- NC block
- END number

The blocks between the DO statement and the END statement are repeated continuously in an infinite loop unless one of the following events occurs:

- The program exits the loop with a GOTO or M99 P ____ jump statement.
- The program execution is terminated with an M02 or M30.
- The right mouse button is pressed. The right mouse button acts as a graphics reset.

⇨ To exit an infinite DO loop while the program is being drawn, press the console Draw key.

Stop Program Execution

The M02 (End of Program) and M30 (End Program) program control statements stop program execution. The following examples of program control statements are used correctly:

Nested WHILE Loops	Branch Outside WHILE Loop	Subprogram Call from Inside WHILE Loop	Reuse of DO-END Pairing Number
WHILE[...] DO 100 NC blocks **WHILE[...] DO 200** NC blocks **WHILE[...] DO 250** NC blocks **END 250** NC blocks **END 200** NC blocks **END 100**	**WHILE[...] DO 200** NC blocks **GOTO 3535** NC blocks **END 200** NC blocks **N3535**	**WHILE[...] DO 150** NC blocks **M98 P3000** NC blocks **END 150** NC blocks **WHILE[...] DO 200** NC blocks **G65 P3000** NC blocks **END 200** NC blocks	**WHILE[...] DO 100** NC blocks **END 100** NC blocks **WHILE[...] DO 100** NC blocks **END 100** NC blocks **WHILE[...] DO 100** NC blocks **END 100**

Table 6–9. Correct Program Control Statement Examples

These examples show **incorrect** use of program control statements:

Incorrectly Nested WHILE Loops	Branch Into a WHILE Loop	Improper Reuse of DO-END Pairing Number
WHILE[...] DO 100 NC blocks **WHILE[...] DO 200** NC blocks **WHILE[...] DO 250** NC blocks **END 100** NC blocks **END 200** NC blocks **END 250**	**GOTO 3535** *NC blocks* **WHILE[...] DO 200** NC blocks **N3535** NC blocks **END 200**	**WHILE[...] DO 100** NC blocks **WHILE[...] DO 100** NC blocks **END 100** NC blocks **END 100** NC blocks

Table 6–10. Incorrect Program Control Statement Examples

Subprograms

Subprograms are stand-alone NC programs that can be called from another NC program. Subprograms begin with the letter “O” or the “:” character followed by a four-digit number that identifies the subprogram. Each subprogram ends with an *M99* statement. The only limitation for the number of NC files and subprograms the software can load is the amount of available dynamic RAM memory.

The following is a sample subprogram:

```
N10     O7162          ⇐ begins with “O” followed by 4-digit number
N20     G00 G90
N30     M25
N40     X0 Y0
N50     T1 M06
N60     Z5.
N70     S2000 M03
N80     Z0.05
N90     M99            ⇐ ends with M99
```

Subprograms can be nested 15 levels deep. In general, different types of subprogram calls can be used in various combinations. There are some restrictions in the use of modal subprograms and user defined G, M, B, S, and T subprogram calls, however, which will be described in more detail later.

Programs cannot call themselves as subprograms because the repetition exhausts the 15 levels of subprogram nesting. For the same reason, a user defined code cannot be used in a program which is associated with the same user defined code. For example, a G65 P5000 command is illegal within the program 5000.

G65 Subprogram Call

The G65 subprogram command has the following form:

G65 P____ L_____ [followed by optional arguments]

The P represents the subprogram number and the L represents the number of iterations that the subprogram must perform. These two methods of argument passing can be used together:

Arguments

In a G65 subprogram call, the local variables in the calling program are not copied to the local variables in the called subprogram. Arguments which follow the G65 command are copied to the local variables in the subprogram as illustrated in the following command:

G65 P5080 A0.0 B8 C2.3 S6 T2 H81 I9 J3.5 K0 Z-1 R.1

The value which follows A is copied to the local variable #1 in the subprogram. The table on the following page shows the relationships between the subprogram arguments and the local variables in the subprograms.

Multiple Arguments

Multiple I, J, and K arguments can also be used as subprogram arguments. For example, if three I arguments are used in the subprogram call, the first I maps to the #4 variable, the second I maps to the #7 variable, and the third I maps to the #10 variable. The following subprogram call is legitimate:

G65 P2000 A2.3 B3.2 I2.0 J3. K5.4 I3. I5. J2. I6. W3. U3

Only numbers may be used as arguments in a G65 subprogram call; no variables or expressions can be used. If multiple iterations of the subprogram are to be performed, the local variables will be initialized to the same argument values.

Passing Argument Lists to Subprograms in Macro Mode B

There are several methods for passing arguments and parameters to subprograms. The G65 and G66 subprogram calls allow an argument list to be provided after the G65 and G66, respectively. The user defined M Code and the user defined G Code allow an argument list to be provided after the user defined Code. The argument list consists of various letters followed by values. The values are then stored as local variables within the subprogram.

The table below lists the correspondence between the arguments and the local variables in Macro Mode B. The argument list is optional. Any arguments which are not included in the list are given vacant status.

Macro Mode B			
Local Variables	**Subprogram Arguments**	**Local Variable**	**Subprogram Arguments**
#1	Argument A	#18	Argument R or K5
#2	Argument B	#19	Argument S or I6
#3	Argument C	#20	Argument T or J6
#4	Argument I or I1	#21	Argument U or K6
#5	Argument J or J1	#22	Argument V or I7
#6	Argument K or K1	#23	Argument W or J7
#7	Argument D or I2	#24	Argument X or K7
#8	Argument E or J2	#25	Argument Y or I8
#9	Argument F or K2	#26	Argument Z or J8
#10	Argument I3	#27	Argument K8
#11	Argument H or J3	#28	Argument I9
#12	Argument K3	#29	Argument J9
#13	Argument M or I4	#30	Argument K9
#14	Argument J4	#31	Argument I10
#15	Argument K4	#32	Argument J10
#16	Argument I5	#33	Argument K10
#17	Argument Q or J5		

Table 6–11. Macro Mode B Local Variables and Subprogram Arguments

Layering of Local Variables within Subprogram Calls

M98 subprogram calls use local variables differently from other subprogram calls since the called subprogram does not get a new set of local variables. Changes made to the local variables within the current subprogram will be retained when the calling program is re-instated.

Changes can be made to the local variables within the current subprogram, but when program execution returns to the calling program, the values of the local variables of the calling program are reinstated. The local variables in the subprogram can be changed, however, without affecting the local variables in the calling program. With other subprogram calls, unless an argument list is passed to the subprogram, the local variables are given vacant status.

Specifying Subprogram Iterations

The number of iterations for a subprogram to perform are specified with G65, G66, and M98 subprogram calls.

Using G65 and G66

When making G65 and G66 subprogram calls, the L parameter is used to specify iterations. The maximum number of iterations which can be specified with the G65 and G66 subprogram calls is 999.

Using M98

When making M98 subprogram calls, the P parameter is used to specify iterations as well as the subprogram number. Up to four digits can be used to specify iterations for a maximum of 9999 iterations. Leading zeros are not required when specifying iterations; however, leading zeros are required with a subprogram number that is less than 1000.

G65 Subprogram Example

In the following example a line of holes will be drilled along a line. The type of canned cycle can be determined along with the distance between the holes in both the X and Y axes. The type of canned cycle and various canned cycle parameters can also be set.

ISNC Part Program 1 Inch
BOLT_LN.FNC

```
O4000
T1 M06
M03 G00 G90 X0 Y0 Z0 S1800
/
(B REPRESENTS THE NUMBER OF BOLT HOLES)
(H REPRESENTS THE DESIRED CANNED CYCLE)
(X,Y REPRESENT THE INCREMENTAL DISTANCE BETWEEN HOLES)
(Z REPRESENTS THE HOLE DEPTH)
(R REPRESENTS THE R PLANE LEVEL)
/
#500=99 (RETURN TO R LEVEL)
G65 P5070 X.5 Y.75 B10 H81 Z-1 Q0. R.1 F20.
M30
(************************************************)
(    BOLT HOLE LINE PATTERN  - SUBPROGRAM 5070    )
(************************************************)
O5070
(#2 IS THE NUMBER OF HOLES)
(#11 IS THE CANNED CYCLE NUMBER)
(#26 IS THE HOLE DEPTH)
(#500 IS 99 FOR RETURN TO R LEVEL MODE OR 98 FOR RETURN TO INITIAL POINT)
(#5003 IS THE Z COORDINATE BEFORE THE CANNED CYCLE IS PERFORMED)
/
WHILE [#2GT0] DO250
G91 G#500 G#11 Z#26 Q#17 R[#5003-#18] F#9
G00 X#24 Y#25
#2 = #2-1
END250
M99
```

View the part using the Draw console key to verify that the part is programmed correctly.

Macro Instruction (G65)

G65 Macro instructions are G65 commands which are used to perform mathematical, trigonometric, or program control functions instead of subprogram calls. These commands are intended to support existing programs which use this program format.

The value in the H parameter defines the operation being performed. In all instructions except the GOTO commands H80 through H86, a variable number follows the P parameter. The operation's result is stored in that variable number. In the following command the value stored in variable #100 is added to the number 1 and the resultant value is stored in variable #115.

```
G65 H02 P#115 Q#100 R1
```

For the GOTO commands, the value which follows the P is a positive or negative integer. If the number is negative, the software begins searching for the sequence number at the beginning of the file and continues to search for the sequence number until reaching the end of the file. If the number is positive, the search for the sequence numbers begins with the block after the GOTO command and continues until reaching the end of the file. The software then searches from the beginning of the file until reaching the GOTO command block.

The values which follow Q and R are general purpose parameters which are used in mathematical, logical, or GOTO operations. The specific operations are listed in the following table.

Format

The following is the G65 Macro Instruction format:

G65 H ____, P #a , Q #b , R #c .

The table below lists the Descriptions and Instruction Functions for the H codes used in the G65 macro instructions:

H Code	Description	Instruction Function
H01	Definition, Substitution	#a = #b
H02	Addition	#a = #b + #c
H03	Subtraction	#a = #b - #c
H04	Product	#a = #b * #c
H05	Division	#a = #b / #c
H11	Logical Sum	#a = #b .OR. #c
H12	Logical Product	#a = #b .AND. #c
H13	Exclusive OR	#a = #b .XOR. #c
H21	Square Root	#a = √#b
H22	Absolute Value	#a = |#b|
H23	Remainder	#a = #b - trunc (#b/#c) * #c trunc: discard fractions less than 1.
H24	Conversion from BCD to Binary	#a = BIN(#b)
H25	Conversion from binary to BCD	#a = BCD(#b)
H26	Combined Multiplication/Division	#a = (#a * #b) / #c
H27	Combined Square Root 1	#a = √(#b2 + #c2)
H28	Combined Square Root 2	#a = √(#b2 - #c2)
H31	Sine	#a = #b * SIN(#c)
H32	Cosine	#a = #b * COS (#c)
H33	Tangent	#a = #b * TAN(#c)
H34	Arc tangent	#a = TAN(#b/#c)
H80	Unconditional Divergence (GOTO)	GOTO a
H81	If Statement, Equals	IF #b = #c, GOTO a
H82	If Statement, Not Equal	IF #b ≠ #c, GOTO a
H83	If Statement, Greater Than	IF #b > #c, GOTO a
H84	If Statement, Less Than	IF #b < #c, GOTO a
H85	If Statement, Greater Than/Equals	IF #b >= #c, GOTO a
H86	If Statement, Less Than/Equals	IF #b <= #c, GO TO a

Table 6–12. H Code Descriptions and Instruction Functions for G65 Macro Instructions

For H80 through H86, if “a” has a negative value, the software performs a GOTO but begins looking for the sequence number at the beginning of the program. No variables can be used for the P parameter for H80 through H86.

The G65 Macro Instructions are intended to support existing Macro A programs. Use equations and regular GOTO statements in place of these instructions when developing new programs.

For example use #100 = 4.56 OR #110 instead of G65 #11 P#100 Q4.56 R#110.

And use IF [#150 EQ #160] GOTO 100 instead of G65 H81 P100 Q#150 R#160.

These commands can be used in either Macro A or B mode.

Example

The following example shows how to use G65 macro instructions in a Bolt Hole Circle subprogram:

ISNC Part Program 1 Inch G65INST.FNC

```
%
(#600 IS BOLT HOLE CIRCLE X COORD)
(#601 IS BOLT HOLE CIRCLE Y COORD)
(#602 IS BOLT HOLE CIRCLE RADIUS)
(#603 IS STARTING ANGLE)
(#604 IS NUMBER OF BOLT HOLES)
#600 = 0
#601 = 0
#602 = 2.
#603 = 30.
#604 = 12.
T1 M6
G00 X0 Y0 Z0.05
M98 P3030
G00 X0 Y0 Z0.05
M02
O3030
(#110 IS BOLT HOLE COUNTER)
(#112 IS ANGLE OF CURRENT HOLE)
(#113 IS X COORD OF CURRENT HOLE)
(#114 IS Y COORD OF CURRENT HOLE)
N10 G65 H01 P#110 Q0
G65 H22 P#111 Q#604
N20 G65 H04 P#112 Q#110 R360
G65 H05 P#112 Q#112 R#604
```

ISNC Part Program 2 Inch

G65INST.FNC

```
G65 H02 P#112 Q#603 R#112
G65 H32 P#113 Q#602 R#112
G65 H02 P#113 Q#600 R#113
G65 H31 P#114 Q#602 R#112
G65 H02 X#114 Q#601 R#114
G90 H00 X#113 Y#114
G81 Z-1. F20.
G80
G65 H02 P#110 Q#110 R1
G65 H84 P-20 Q#110 R#111
M99
```

View the part using the Draw console key to verify that the part is programmed correctly

Modal Subprograms

Modal subprograms are executed every time a motion is performed (i.e., after a Move command). Use them for performing repetitive tasks at different locations. The repetitive tasks can be put inside a modal subprogram. A subprogram call can be made to a program which contains X, Y, and Z locations and will be executed at each of these locations.

A modal subprogram will not be modal within another modal subprogram. If the modal subprogram contains Move commands, the modal subprogram will not be performed after Move commands within the modal subprogram. This allows Move commands to be contained within modal subprograms.

These methods allow the subprogram call to be modal:

- A Modal Subprogram Call (G66) Command
- A Modal user defined G code

Modal Subprogram Call (G66)

In a G66 Modal subprogram call, the subprogram is repeatedly executed after each Move command until the Modal Subprogram Call Cancel (G67) command is performed.

Modal User Defined G Code

The user defined G code is made modal by entering a negative number in the G code column on the Change NC Parameters screen. Only one user defined G code can be designated as a modal subprogram. The first one in the list is treated as modal if more than one negative number is entered in the G code column. The remaining negative G codes are treated as regular user defined G codes.

When the modal user defined G code is encountered in the NC program, the subprogram becomes modal until a G67 is used. Only one modal subprogram can be in effect at any given time; an error message occurs if a modal subprogram is first initiated with a G66 command and the modal user defined G code is then attempted.

Modal Subprogram Cancel (G67)

The G67 command is used to cancel modal subprograms initiated with either the G66 or with a modal user defined G code.

Modal Subprogram Call (G66) Example

The following program draws a series of squares and rectangles:

NC Part Program 1 Inch G66.FNC

```
%
(EXAMPLE OF MODAL SUBPROGRAM CALL G66)
(P6010 IS USED AS MODAL SUBPROGRAM)
(THE VALUES AFTER I AND J ARE PASSED TO)
(THE SUBPROGRAM. THE SUBPROGRAM IS ONLY)
(EXECUTED AFTER BLOCKS WITH MOVE COMMANDS.)
X0 Y0 Z0
X5 G66 P6010 I1. J1.5
Y-3
X-5
Y0
(MODAL SUBPROGRAM IS NOW CANCELED WITH G67)
G67
Y3
(THE MODAL SUBPROGRAM IS STARTED AGAIN WITH)
(NEW PARAMETERS.)
X0 G66 P6010 I3.J1.
Y0
Y-2
M02
:6010
(THIS SUBPROGRAM CREATES A SIMPLE BOX SHAPE.)
G91
X#4
Y#5
X-#4
Y-#5
G90
M99
```

View the part using the Draw console key to verify that the part is programmed correctly.

User Defined Codes

M, G, S, B, and T codes can be customized to perform specialized tasks. Enable these codes on the NC Parameters Configuration Parameters screen by placing the cursor at the code's field and selecting the Yes (F2) softkey.

User defined G and M codes define a custom code which performs a specialized task, replaces an existing G or M code, or provides compatibility between different NC dialects from various machine tool control manufacturers. For instance, if a manufacturer uses G codes for canned cycles, User defined G codes can re-map the canned cycles. This allows a BNC subprogram to be used in ISNC mode.

The user defined B, S, and T subprograms provide additional user defined subprograms. User defined S and T subprograms replace the spindle and tool functions with custom subprograms.

M Codes

Up to 13 user defined M codes can be programmed from M01 through M255 (except M02, M30, M98, and M99). Enable the user defined M codes by selecting Yes in the Enable User M Code field on the NC Parameters screen. The user defined M codes can be assigned to subprograms 9020 through 9029 and 9001 through 9003.

There are no modal user defined M codes; therefore, negative numbers cannot be entered in the column for user defined M codes on the Change NC Parameters screen.

G Codes

G1 through G255 (except G65, G66, and G67) can be programmed for user defined G codes. Enable the user defined G codes by selecting Yes in the Enable User G Code field on the NC Parameters screen. If a negative value is entered for one of the user defined G codes, the subprogram becomes modal. The subprogram is executed after every Move command once the modal G code is invoked.

The modal G code, like the G66 code, is canceled with a G67 command. Use programs 9010 through 9019 only for the user defined G codes.

S, B, and T Codes

Enable the user defined S, B, or T codes by selecting Yes in the Enable User S, B, or T Code field on the Change NC Parameters screen. The software executes the appropriate subprogram when it encounters an S, B, or T code in an NC program.

If a user defined T subprogram call is made, Tool Function commands T____ contained within program 9000 will be treated as normal Tool Function commands. If a number follows the T, this value is stored in variable #149. Including a number after the S, B, or T is optional.

The variable numbers and subprogram numbers are *fixed* for these subprogram calls:

User Defined B	Variable #146	Program #9028
User Defined S	Variable #147	Program #9029
user Defined T	Variable #149	Program #9000

Table 6–13. Fixed Variable and Subprogram Numbers

Passing Single Dedicated Parameters to Subprograms

User defined subprogram calls' conditions are listed in the table below. If a user defined M, G, S, B, or T subprogram call is not allowed, it is treated as a normal M, G, S, B, or T code. There are no restrictions for G65, G66, and M98 subprogram calls provided that the subprogram has been loaded in memory.

For user defined S, B, and T subprograms, a single parameter is passed to the subprogram. These parameters are optional and they are stored at specific variable locations. The value of the passed parameter can be retrieved by accessing the specific variable which corresponds to the parameter. For example, variable #149 is used for the T subprogram parameter, variable #147 is used for the S subprogram parameter, and #146 is used for the B subprogram parameter. The table below lists the conditions for using modal and user defined subprograms:

Types of User Defined Subprograms	**Conditions Under Which User Defined Subprograms Can Be Utilized**								
	No Sub-prgram Call	**M98**	**G65**	**G66**	**User M Code**	**User G Code**	**User B Code**	**User S Code**	**User T Code**
User G Code	X	X	X	X		X	X	X	
User M Code	X	X	X	X			X	X	
User T Code	X	X	X	X					
User B Code	X	X	X	X					
User S Code	X	X	X	X					

Table 6–14. Conditions Under Which User Defined Subprograms Can Be Utilized

While executing a user defined S Code subprogram, user defined G and M Codes can be used, but user defined S, B, and T Codes cannot. There are several different ways of performing subprogram calls. The information in the following table illustrates the different methods for making subprogram calls. The format of the optional argument list is the same for all the different methods of subprogram calls.

Types of Subprogram Calls	Subprogram Capabilities			
	Modal Capability	Can Specify Iterations	Optional Argument List	Single Predefined Parameters
G65		X	X	
G66	X	X	X	
M98		X		
User Defined G Codes	Optional		X	
User Defined M Codes			X	
T Code				X
B Code				X
S Code				X

Table 6–15. Subprogram Capabilities

This table shows which program numbers are assigned to the different macro calls and their variables:

Program #	Macro Call	Variables	Note
9000	T-Subprogram		Parameter @ #149
9001	M-Macro Mode A	#8004-#8026 are R/W, #8030-#8046 are R	Status #8104-#8146 Tool Offsets #1-#99
9002	M-Macro Mode A	#8004-#8026 are R/W, #8030-#8046 are R	Status #8104-#8146 Tool Offsets #1-#99
9003	M-Macro Mode A	#8004-#8026 are R/W, #8030-#8046 are R	Status #8104-#8146 Tool Offsets #1-#99
9010	G-Code		
9011	G-Code		
9012	G-Code		
9013	G-Code		
9014	G-Code		
9015	G-Code		
9016	G-Code		
9017	G-Code		
9018	G-Code		
9019	G-Code		
9020	M-Macro Mode B	#1-#33	
9021	M-Macro Mode B	#1-#33	
9022	M-Macro Mode B	#1-#33	
9023	M-Macro Mode B	#1-#33	
9024	M-Macro Mode B	#1-#33	
9025	M-Macro Mode B	#1-#33	
9026	M-Macro Mode B	#1-#33	
9027	M-Macro Mode B	#1-#33	
9028	M-Macro Mode B	#1-#33	B-Parameter @ #146
9029	M-Macro Mode B	#1-#33	S-Parameter @ #147

Table 6–16. Program Numbers and Their Assigned Macro Calls and Variables

User Defined G Code Example

The example below shows how the BNC G86 Bore Orient cycle can be re-mapped to the ISNC G76 cycle. Up to 10 user defined G codes can be defined on the Change NC Parameters screen.

To re-map BNC G86 to ISNC G76 follow these steps:

1. Enable the user defined G codes.
2. Set 9010 to 86 on the Change NC Parameters screen.
3. Load the 9010: program.

```
NC Part Program                    1                              Inch
G86_TRAN.FNC
G99 G90 G00 X0.0 Y0.0 Z1.0
G86 X2.0 Y3.0 Z3.0 I1.0 J0.0 R1.0 F100.    ⇐ User defined G86 called
O9010                                      ⇐ Subprogram—9010— Start
IF [#4003 EQ 91] GOTO 100                     user defined G86)
(TRANSLATION FOR ABSOLUTE MODE)
G76 X#24 Y#25 Z[#5003-#26] R[#5003-#18] I#4 J#5 F#9
GOTO 200
(TRANSLATION FOR INCREMENTAL MODE)
N100 G76 X#24 Y#25 Z[-#26] R[-#18] I#4 J#5 F#9
N200
M99                                        ⇐ End of Subprogram—9010
```

User Defined G and M Code Example

Follow these steps before running the sample user defined G and M code program:

1. Press the console Input key to display the Input screen.
2. Press the Program Parameters (F4) softkey.
3. Select the NC Parameters (F3) softkey. The Change NC Parameters screen appears.
4. Enable the User M Codes and G Codes fields by placing the cursor at each field and selecting the Yes (F2) softkey.
5. Enter 77 into the 9020 field.
6. Enter -96 into the 9010 field.

The following program re-maps a BNC G86 Bore Orient cycle to the ISNC G76 cycle:

NC Part Program 1 Inch
USER_MAC.FNC

```
%
(GOTO NC PARAMETERS PAGE)
(FIRST ENABLE USER DEFINED G AND M CODES)
(SET USER DEFINED 9010 TO -96)
(SET USER DEFINED 9020 TO 77)
X0 Y0 Z0     ⇐ Main Program Start
G96                ⇐ G96 Call
X5
Y-3
X-5
Y0
G67
Y3
X0
M77                ⇐ M77 Call
Y4
M77
M02                ⇐ Main Program End
:9010              ⇐ Subprogram—9010— Start (user defined G96)
G91
X1
Y1
X-1
Y-1
G90
M99                ⇐ End of Subprogram—9010
:9020              ⇐ Subprogram—9020— Start (user defined M77)
G91
X.75
Y1.5
X-.75
Y-1.5
M99                ⇐ End of Subprogram—9010
E
```

View the part using the Draw console key to verify that the part is programmed correctly.

User Defined S, B, and T Code Example

Follow these steps before running the sample user defined S, B, and T code program:

1. Press the console Input key to display the Input screen.
2. Press the Program Parameters (F4) softkey.
3. Select the NC Parameters (F3) softkey. The NC Parameters screen appears.
4. Enable the User S Codes, B Codes, and T Codes fields by placing the cursor at each field and selecting the Yes (F2) softkey.

NC Part Program 1 Inch
BST.FNC

```
%
(GOTO NC PARAMETERS PAGE)
(FIRST ENABLE B, S, AND T CODES)
X0 Y0 Z0
X5 B1.4 Y2  ⇐ user defined B Call
Y-3 S2.
X-3 T1.5 Y-1.5⇐ user defined T Call
X-5 Y0 B.75
Y3 S1.8     ⇐ user defined S Call
X0 T2.
Y0
M02
:9000(USER DEFINED T MACRO)
G91
X#149
X-[#149/2] Y#149
X-[#149/2] Y-#149
G90
M99
:9028(USER DEFINED B MACRO)
G91
X#146
Y#146
X-#146
Y-#146
G90
M99
```

NC Part Program 2 Inch
BST.FNC

```
:9029(USER DEFINED S MACRO)
G91
X#147
X[#147/2] Y#147
X-#147
X-[#147/2] Y-#147
G90
M99
```

View the part using the Draw console key to verify that the part is programmed correctly.

NCPP Variable Summary

In the tables below, the Type column indicates the type of variable: Argument (A), System (S), Common (C), and Local (L). The R/W column indicates whether the variable is Read or Read/Write.

Variable Number	Type	R/W	Local Variables for Macro Mode B (Note 3)
#1	A	R/W	Address A (Note 4)
#2	A	R/W	Address B (Note 4)
#3	A	R/W	Address C (Note 4)
#4	A	R/W	Address I (Note 1)or I1 (Note 2)
#5	A	R/W	Address J (Note 1) or J1 (Note 2)
#6	A	R/W	Address K (Note 1) or K1(Note 2)
#7	A	R/W	Address D (Note 1) or I2 (Note 2)
#8	A	R/W	Address E(Note 1) or J2 (Note 2)
#9	A	R/W	Address F (Note 1) or K2 (Note 2)
#10	A	R/W	Address I3 (Note 2)
#11	A	R/W	Address H (Note 1) or J3 (Note 2)
#12	A	R/W	Address K3 (Note 2)
#13	A	R/W	Address M (Note 1)or I4 (Note 2)
#14	A	R/W	Address J4 (Note 2)
#15	A	R/W	Address K4 (Note 2)
#16	A	R/W	Address I5 (Note 2)
#17	A	R/W	Address Q (Note 1) or J5 (Note 2)
#18	A	R/W	Address R (Note 1) or K5 (Note 2)
#19	A	R/W	Address S (Note 1) or I6 (Note 2)
#20	A	R/W	Address T (Note 1) or J6 (Note 2)
#21	A	R/W	Address U (Note 1) or K6 (Note 2)
#22	A	R/W	Address V (Note 1) or I7 (Note 2)
#23	A	R/W	Address W (Note 1) or J7 (Note 2)
#24	A	R/W	Address X (Note 1) or K7 (Note 2)
#25	A	R/W	Address Y (Note 1) or I8 (Note 2)
#26	A	R/W	Address Z (Note 1) or J8 (Note 2)
#27	A	R/W	Address K8(Note 2)
#28	A	R/W	Address I9 (Note 2)
#29	A	R/W	Address J9 (Note 2)
#30	A	R/W	Address K9(Note 2)
#31	A	R/W	Address I10 (Note 2)
#32	A	R/W	Address J10 (Note 2)
#33	A	R/W	Address K2 (Note 2)

Table 6–17. NCPP Local Argument Variables (#1 - #33) for Macro Mode B

1. Valid for argument assignment method 1 where multiple sets of (I,J,K) are not used.
2. Valid for argument assignment method 2 where multiple sets of (I,J,K) are used.
3. Local variables are used to pass arguments to a macro. If a local variable without a transferred argument is vacant in its initial status, it can be used freely in the macro.
4. Valid for argument assignment method 1 and 2

Custom Macro Mode A			
Variable Number	**Type**	**R/W**	**Tool Offset Amounts**
#1 to #99	S	R/W	Tool offset amounts for custom macro mode A

Table 6–18. Tool Offset Variable Numbers for Macro Mode A (#1 - #99)

Variable Number #100 to #199 #500 to #999	Type	R/W	Common Variables
#100 to #199	C	R/W	Use these variables to store binary numbers as well as real numbers. All programs and subprograms can read and write to them. Variables #146, #147, and #149 also store the values which follow the B, S, and T code when B, S, and T subprogram calls are performed.
#500 to #999	C	R/W	Use these variables to store binary numbers as well as real numbers. All programs and subprograms can read and write to them.

Table 6–19. NCPP Common Variables (#100 - #199 and #500 - #599)

Variable Number #2000 to #2200	Type	R/W	Tool Offset/Wear Number Variables Macro Mode B
#2000	S	R	Tool Length Offset Number 00. (Always 0.)
#2001 to #2200	S	R/W	Tool Length Offset Number 1-200 for H.
#12001 to #12200	S	R/W	Tool Radius Offset Number 1-200 for D.

Table 6–20. Tool Offset/Wear Number Variables for Macro Mode B (#2000 - #2200)

Variable Number #2500 to #2706	Type	R/W	(X,Y,Z) External Work Compensation and Work Coordinate System 1 (G54) to 6 (G59) Used to Read/Write Zero Point Offset Values
#2500	S	R/W	X External Work Compensation. Compensation is applied to the Work Coordinate System 1 to 6 X value.
#2501 to #2506	S	R/W	X For Work Coordinate System 1 to 6
#2600	S	R/W	Y External Work Compensation. Compensation is applied to the Work Coordinate System 1 to 6 Y value.
#2601 to #2606	S	R/W	Y For Work Coordinate System 1 to 6
#2700	S	R/W	Z External Work Compensation. Compensation is applied to the Work Coordinate System 1 to 6 Z value.
#2701 to #2706	S	R/W	Z For Work Coordinate System 1 to 6
#2800 to #2906	S	R/W	A and B For External Work Coordinates 1 to 6

Table 6–21. External Work Compensation and Work Coordinate System 1 Variables (#2500 - #2706)

Variable Number #3000 to #3005	Type	R/W	Miscellaneous System Parameters
#3004	S	R/W	If #3004 = 0 to 7, feed hold, feedrate override (F.O.), or exact stop check (E.S.C.) will be enabled or disabled. Currently not implemented. #3004 Feed Hold F.O. E.S.C. 0 0 0 0 1 X 0 0 2 0 X 0 3 X X 0 4 0 0 X 5 X 0 X 6 0 X X 7 X X X Key: 0 = Effective, X= Suppressed Feed Hold and Exact Stop are currently not implemented.
#3005	S	R	Provides the condition of mirror image of each axis at that time. **Bit 0:** X axis (set to 1 if X axis mirroring is being used) **Bit 1:** Y axis (set to 1 if Y axis mirroring is being used) **Bit 4:** inch/mm status flag inch input (set to 1) mm input (set to 0) **Bit 6:** absolute/incremental flag absolute (set to 1) incremental (set to 0)
#3020	S	R	Indicates whether the probe touched during a G31 move. Equals 0 if the probe does not touch. Equals 1 if the probe does touch.

Table 6–22. Miscellaneous System Parameters Variables (#3000 - #3005)

For Ultimax PC/PC Plus #3102 is always 1; #3103-#3129 are always 0.0. For Ultimax 4 (CNC) #3110-#3115, #3120-#3125; #3129 are always 0.0.

Tool Probe			
Variable Number	**Type**	**R/W**	**Description**
#3101	S		Current Tool Number
#3102	S	R	Tool Probe present
#3103	S	R	Tool Probe X location
#3104	S	R	Tool Probe Y location
#3110	S	R	Tool Probe X Plus Offset location
#3111	S	R	Tool Probe X Negative Offset location
#3112	S	R	Tool Probe Y Plus Offset location
#3113	S	R	Tool Probe Y Negative Offset location
#3114	S	R	Tool Probe Z Plus Offset location
#3115	S	R	Tool Probe Z Negative Offset location
#3116	S	R	Tool Probe Tool Length tolerance

Table 6–23. Tool Probe Variables (#3101 - #3116)

Part Probe			
Variable Number	**Type**	**R/W**	**Description**
#3120	S	R	Part Probe X Plus Offset location
#3121	S	R	Part Probe X Negative Offset location
#3122	S	R	Part Probe Y Plus Offset location
#3123	S	R	Part Probe Y Negative Offset location
#3124	S	R	Part Probe Z Plus Offset location
#3125	S	R	Part Probe Z Negative Offset location
#3126	S	R	Part Probe Z (reserved; not supported)
#3127	S	R	Part Probe Safety Min Z (reserved; not supported)
#3128	S	R	Tool Probe safety Min Z (reserved; not supported)
#3129	S	R	Tool Probe Z (reserved; not supported)

Table 6–24. Part Probe Variables (#3120 - #3129)

Tool Variables			
Variable Number	**Type**	**R/W**	**Description**
#3201-#3300	S	R	Tool Type
#3301-#3400	S	R	Tool Diameter
#3401-#3500	S	R	Reserved
#3501-#3600	S	R	Tool Probe Offset X
#3601-#3700	S	R	Tool probe Offset Y
#3701-#3800	S	R/W	Probe Calibration
#3801-#3900	S	R/W	Tool Calibration

Table 6–25. Tool Variables (#3201 - #3900)

⇨ There are 100 variables each reserved for tool type, tool diameter, tool calibration, probe calibration, tool probe offset X, and tool probe offset Y regardless of whether or not the machine can handle that many tools. If the program tries to access a variable for a tool that does not exist, an error is generated. The variables for tool type (#3201 – #3300) have these values:

Undefined	-10
Drill	-11
Tap	-12
Bore	-13
Mill	-14
Face Mill	-15
Ball End	-16
Back Spot Face	-17
Probe	-18
Gun Drill	-19

Variable Number #4001 to #4120	Type	R / W	Modal Information from Previous Block
#4001 to #4021	S	R	G Code Groups 1 to 21
#4022	S	R	G Code Group 22
#4102	S	R	B Code
#4107	S	R	D Code
#4109	S	R	F Code
#4111	S	R	H Code
#4113	S	R	M Code
#4114	S	R	Sequence Number of previous block
#4115	S	R	Program Number of previous block
#4119	S	R	S Code
#4120	S	R	T Code

Table 6–26. Modal Information from Previous Block Variables (#4001 - #4120)

Variable Number #4201 to #4210	Type	R / W	Modal Information for Current Block
#4201 to #4221	S	R	G Code Groups 1 to 21
#4222	S	R	G Code Group 22
#4302	S	R	B Code
#4307	S	R	D Code
#4309	S	R	F Code
#4311	S	R	H Code
#4313	S	R	M Code
#4314	S	R	Sequence Number of current block
#4315	S	R	Program Number of current block
#4319	S	R	S Code
#4320	S	R	T Code

Table 6–27. Modal Information for Current Block Variables (#4201 - #4320)

Variable Number #5001 to #5083	Type	R/W	Position Information
#5001 to #5004	S	R	X, Y, Z, and A axis block end part coordinate respectively. Coordinates are referenced to the current coordinate system.
#5021 to #5023	S	R	X, Y, and Z axis machine coordinate position respectively. Coordinates are referenced to the current working coordinate system. Based on real time measured position. These variable cannot be used while Cutter Compensation is active.
#5041 to #5043	S	R	X, Y, and Z axis work coordinate position respectively. Coordinates are referenced to the current working coordinate system. Based on real time measured position. These variable cannot be used while Cutter Compensation is active.
#5061 to #5063	S	R/W	X, Y, and Z axis skip signal position respectively. Coordinates are referenced to the current coordinate system (machine, local, or working). Based on real time measured position.
#5081 to #5083	S	R	X, Y, and Z axis tool offset respectively.

Table 6–28. Position Information Variables (#5001 - #5083)

Arguments in the following table are variables used only in Macro A subprograms that pass parameters to subprograms. They are used to support existing Macro A subprograms. When a Macro A subprogram is called, the variables #8004 to #8026 are initialized with the address values in the calling program. Variables #8104 to #8126 are set to 1 if the address value is valid, and they are set to 0 if the address value is invalid.

In general, variables #8004 to #8026 are initialized after a subprogram call is made. These variables are not kept up to date. They are only valid immediately after a subprogram call. Variables #8104 to #8126 are set to 1 during a subprogram call and reset to 0 when the software returns from the subprogram.

Likewise, variables #8030 to #8046 are initialized to the G group modal status when a Macro A subprogram is called. Variables #8130 to #8146 are then set to 1 if the corresponding parameter is passed.

Variable Number				Macro Mode A
Value	Status	Type	R/W	Subprogram Parameters
#8004	#8104	A	R	I
#8005	#8105	A	R	J
#8006	#8106	A	R	K
#8009	#8109	A	R	F
#8010	#8110	A	R	G
#8011	#8111	A	R	H
#8013	#8113	A	R	M
#8014	#8114	A	R	N
#8016	#8116	A	R	P
#8017	#8117	A	R	Q
#8018	#8118	A	R	R
#8019	#8119	A	R	S
#8020	#8120	A	R	T
#8024	#8124	A	R	X
#8025	#8125	A	R	Y
#8026	#8126	A	R	Z

Table 6–29. Macro Mode A Subprogram Parameters Variables (#8004 - #8026, #8104 - #8126)

⇨

The G code groups' values are stored in addresses #8004 to #8026 for Macro Mode A subprogram calls G65, G66, and user defined M and G codes. The G code groups' status is stored in addresses #8114 to #8126. The software sets the value to 1 if an argument is specified in the subprogram call.

#8104 is non-zero if an argument is specified during the macro call and zero if no argument is specified.

#8004 has a valid value if #8104 is non-zero and may be undefined if #8104 is zero.

Variable Number				Macro Mode A
Value	**Status**	**Type**	**R / W**	**G Code Group Status**
#8030	#8130	S	R	Group 00 G codes
#8031	#8131	S	R	Group 01 G codes: G00, G01, G02, G03
#8032	#8132	S	R	Group 02 G codes: G17, G18, G19
#8033	#8133	S	R	Group 03 G codes: G90, G91
#8035	#8135	S	R	Group 05 G codes: G94
#8036	#8136	S	R	Group 06 G codes: G20, G21
#8037	#8137	S	R	Group 07 G codes: G40, G41, G42
#8038	#8138	S	R	Group 08 G codes: G43, G44, G49
#8039	#8139	S	R	Group 09 G codes: G73, G74, G76, G80-G89
#8040	#8140	S	R	Group 10 G codes: G98, G99
#8041	#8141	S	R	Group 11 G codes: G66, G67
#8045	#8145	S	R	Group 15 G codes: G61, G62, G63, G64
#8046	#8146	S	R	Group 16 G codes: G68, G69

Table 6–30. Macro Mode A G Code Group Status Variables (#8030 - #8046, #8130 - #8146)

⇨ The G code groups' status is stored in addresses #8030 to #8046 for Macro Mode A subprogram calls G65, G66, and user defined M and G codes.

The status is stored in addresses #8130 to #8146 and is non-zero if an argument is specified in the subprogram call. It is zero if no argument is specified in the subprogram call.

Programming Examples

This section contains NC programming examples. The first one uses basic BNC programming features while the second example uses the NCPP programming features. The third example illustrates Polar Coordinates in a subprogram.

NC Part Program Example

Following the simple drawing below, of a part program, is a sample NC part program that may be used to test the BNC programming features.

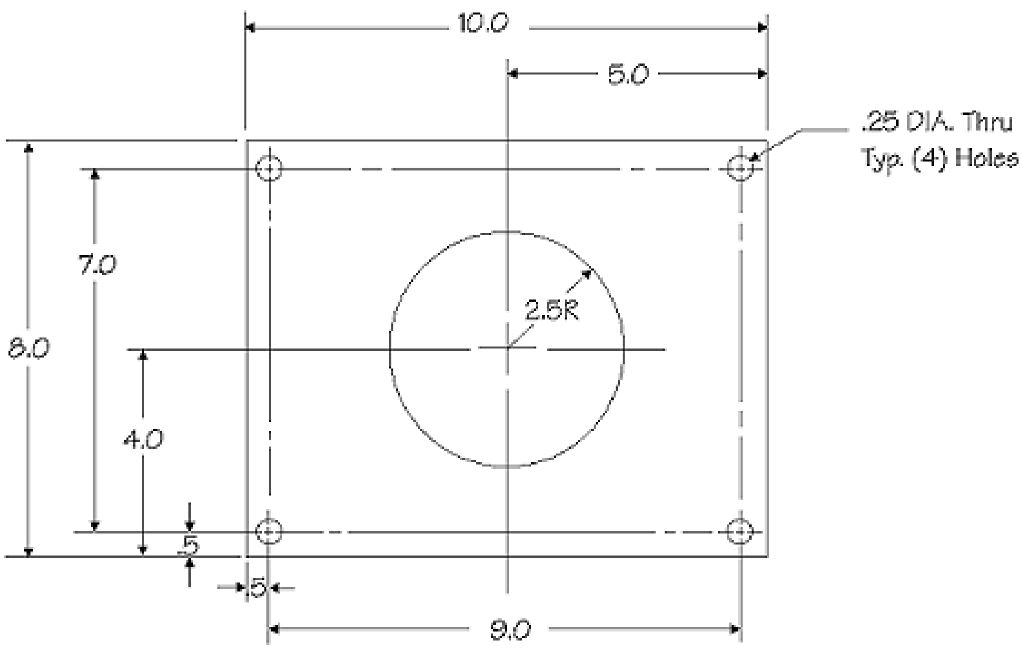


Figure 6–1. Sample NC Part Program Drawing

Here is one way the part shown on the previous page may be programmed using the NC system:

```
BNC Part Program                  1                                Inch
PART.HNC
%
N10 G0 G90 x0. Y0. S500 T1 M6
N12 X0.5 Y0.5 Z0.5 M3
N15 G81 X0.5 Y0.5 Z0.75 F5.
N20 X0.5 Y7.5
N25 X9.5 Y7.5
N30 X9.5 Y0.5
N35 G0 X0. Y0. S1000 T2 M6
N40 G0 X5. Y6.5 Z0.5
N50 G0 G42 X5. Y6.5 M3
N55 G1 Z-0.25 F5.
N65 G2 X5. Y1.5 I5. J4. F10.
N70 X5. Y6.5 I5. J4.
N71 G0 Z0.5
N72 G0 G40 X5. Y6.5
N75 G0 Y10.5 Z3. M2
E
```

NCPP Example—Bolt Hole Circle

The Bolt Hole Circle program uses subprograms to produce five different Bolt Hole patterns, as shown below, and specifies which canned cycle to use, how many holes to skip, and on which hole to begin the skip.

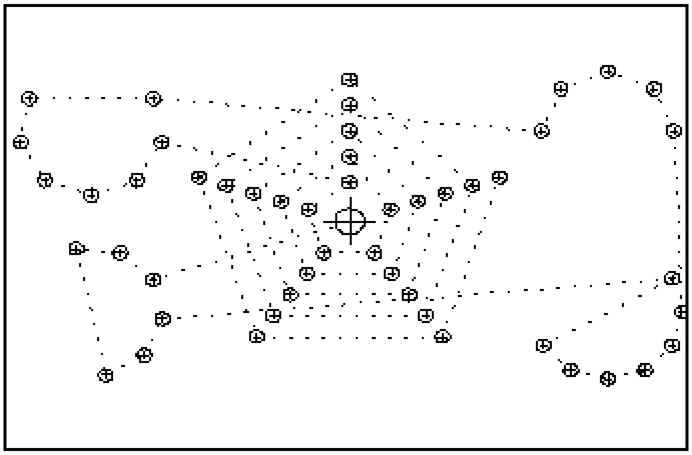


Figure 6–2. Bolt Hole Circle Example Drawing

ISNC Part Program 1 Inch

BOLT_ABS.FNC

```
%
O4000
T1 M06
M03 G00 G90 X0 Y0 Z0 S1800
#500 = 99
G65 P5080 A30.0 B10 C2.5 S4 T3 H81 I-9 J-3.5 K0 Z-1 R-.7
G65 P5080 A30.0 B12 C2.6 S2 T4 H81 I9  J-3.5 K0 Z-1 R-.2
G65 P5080 A0.0  B8  C2.3 S6 T2 H81 I9  J3.5  K0 Z-1 R-.4
G65 P5080 A30.0 B9  C2.5 S2 T1 H81 I-9 J3.5 K0  Z-1 R-.1
#1 = 0
WHILE [#1LT5] DO 100
#500 = 98
G65 P5080 A90.0 B5 C[1.5+#1] S0 T0 H81 I0 J0 K.5 Z-2 R-.7
#1 = #1+1
N1000 END 100
M02
```

ISNC Part Program 2 Inch
BOLT_ABS.FNC

```
O5080
(#1 IS THE START ANGLE)
(#2 IS THE NUMBER OF HOLES)
(#3 IS THE RADIUS)
(#4 IS THE BOLT CIRCLE CENTER PT X COORD)
(#5 IS THE BOLT CIRCLE CENTER PT Y COORD)
(#6 IS THE BOLT CIRCLE CENTER PT Z COORD)
(#18 IS THE RETURN LEVEL)
(#19 IS THE HOLE TO SKIP)
(#11 IS THE CANNED CYCLE NUMBER)
(#26 IS THE HOLE DEPTH)
#30 = [360.0/#2]
#31 = 0
#32 = 0
#33 = 0
WHILE [#31LT#2] DO 250
#7 = [#1+[#31*#30]]
IF [[#19-1]EQ#31] GOTO 200
IF[#32EQ1] GOTO 200
#33 = 1
G00 Z#6
G#500 G#11 Z#26 X[#4+[#3*COS[#7]]] Y[#5+[#3*SIN[#7]]] R[#18] F20.
N200 #31 = #31+1
IF [#33EQ1] GOTO 300
IF [#20EQ0] GOTO 300
#20 = #20-1
#32 = 1
GOTO 310
N300 #32 = 0
N310 #33 = 0
N400 END 250
M99
```

NCPP Example—Gear Pattern

The program below uses Polar Coordinates in a subprogram to generate a Gear pattern:

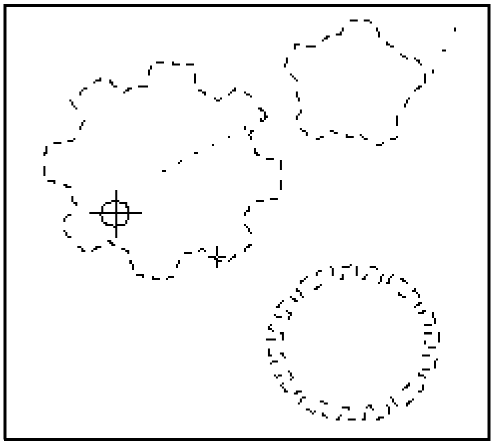


Figure 6–3. Display of Gear Pattern Example

NC Part Program 1 Inch
GEAR_ABS.HNC

```
%
M03 G00 G21 G90 X0 Y0 Z0 S1800
(VARIABLE #4006 - INCHES/METRIC)
IF [#4006 EQ 20] GOTO 10
IF [#4006 EQ 21] GOTO 15
N10 #850 = 25.4
GOTO 20
N15 #850 = 1.0
N20
G65 P5085 A30.0 B8 C2.5 S0 H2. I1 J1 K1 R.45 T.2
G65 P5085 A0.0 B5  C1.5 S0 H1.2 I5 J3   K1 R.3 T.2
G65 P5085 A15. B20 C1.8 S0 H1.5 I5 J-3. K1 R.6 T.1
M02
/
```

NC Part Program **2** **Inch**
GEAR_ABS.HNC

```
O5085
(#1 IS THE START ANGLE)
(#2 IS THE NUMBER OF GEAR TEETH)
(#3 IS THE OUTSIDE RADIUS)
(#11 IS THE INSIDE RADIUS)
(#4 IS THE GEAR CENTER PT X COORD)
(#5 IS THE GEAR CENTER PT Y COORD)
(#6 IS THE GEAR CENTER PT Z COORD)
(#19 IS THE TOOTH TO SKIP)
(#18 IS THE TOOTH RATIO)
/
#30 = [360.0/#2]
#31 = 0
#22 = [#30*#18]
#23 = #30-#22
#24 = #11*#850
#25 = #3*#850
#26 = #20*#23
G52 X#4 Y#5 Z#6
G90 G00 G16 X#25 Y#1
G01 Z-.25 F20.
WHILE [#31LT#2] DO 250
#1 = [#1+[#22]]
G03 G16 X#25 Y[#1] R#3
G01 X#24 Y[#1+#26]
#1 = [#1+[#23]]
G03 X#24 Y[#1-#26] R#11
G01 X#25 Y[#1]
G15
N200 #31 = #31+1
N400 END 250
M99
E
```

RECORD OF CHANGES

704-0116-310 rB, February 2010

Revised by: H.Arle

Approved by: D.Skrzypczak, February 2010

Changes
• Added G07.2, G07.3, G08.1, G08.2, G94.1. • Updated to reflect software changes.

704-0116-310 rA, December 2009

Revised by: H.Arle

Approved by: D.Skrzypczak, Dec 2009

Changes
• Updated to reflect software changes.

INDEX

Numerics

A

B

C

G

H

I

J

K

L

Q

R

S

T

U