HURCO®

MAX® CONTROL FOR TURNING CENTERS

Preliminary
NC Programming Manual

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

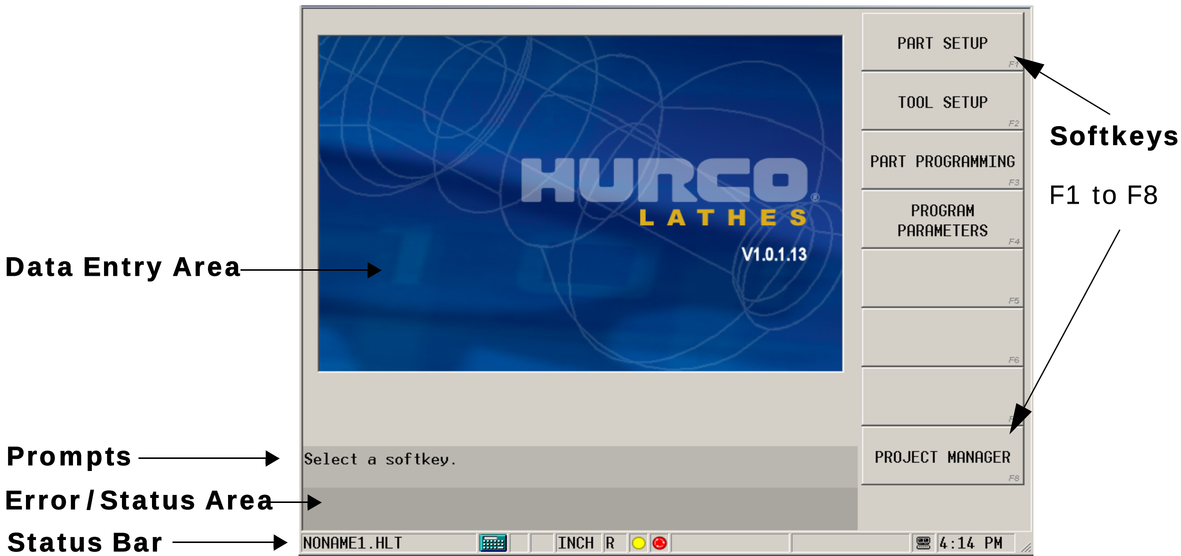
LIST OF FIGURES

LIST OF TABLES

USING THIS MANUAL

Sample Screens

Sample screens in this manual were taken from a Max Control for Lathe Console. All screens are subject to change. The screens on your system may vary slightly. The sample screen here illustrates softkeys and includes the software version.



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 *User Preferences,* on page 2 - 6 for information about making this selection. Softkeys may change upon field entries or other softkey selection.

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

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 can occur anytime the status or error is true. They are not based on the field with the blinking cursor. These messages provide machine information to the operator and error messages may also stop and/or prevent machine opeations. An example of a status message is “Way Lube is Low.”

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). When Inch is displayed, you can select it in the status bar to change to Millimeters, and vice versa.
- Programming mode (R for Radius; D for Diameter). When Radius is displayed, you can select it in the status bar to change to Diameter, and vice versa.
- A yellow icon indicating when the feed hold is on.
- A red icon indicating that the Emergency Stop button has been pressed.

Refer to *Control Panel Function Groups,* on page 1 - 4 for information about console buttons and keys, in addition to other information about using softkeys and the pop-up text entry window.

Using the Touch Screen

The Max Control for Lathes has a touch screen for entering programming data. 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 to make selections.

Printing

To print part or all of this manual from the CD, select **File / Print** to print this document. 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.

Understanding Icons

This manual may contain the following icons:

Caution/Warning

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

Hints and Tricks

Useful suggestions that show creative uses of the Lathe Max 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.

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

NC PROGRAMMING

This manual describes the programming codes supported by the Max Control for Turning Centers software and how to use them in a part program. Descriptions and examples show how to use G codes, M codes, and T codes.

Definitions

The following terms are used in this manual.

- **Block**—a line of code in a part program. Several blocks of code make up an entire part program. To continue a block onto the next line, put a **** (back slash) at the end of the line that is being continued.
- **Modal—**a programmed command that the control remembers until it is canceled by another programmed command from the same modal group. For example, programming G20 Inch mode and G90 Absolute mode in the first block of a part program tells the control to remain in inch mode until G21 Millimeter mode is programmed, and to remain in Absolute mode until G91 Incremental mode is programmed. A modal command stays in effect until another command from the same group is programmed.
- **Machine envelope**—the boundaries established by the axis overtravel switches. If an axis moves onto one of these switches, an emergency stop occurs.
- **Reference position (or Machine home)**—the position where all axes position at the end of a calibration cycle. The coordinates of this position establish the coordinates from which all machine coordinates are measured.

Command Overview

All part programs consist of a combination of the following items:

- **M Code**—Performs miscellaneous functions such as turning on the spindle, turning on coolant, specifying a program stop, or an end of program.
- **S Code**—Sets the spindle speed.
- **F Code**—Sets the modal feed rate for cutting moves.
- **T Code**—Identifies the active tool and activates the offsets for the tool.
- **E Code**—Specifies a fixture offset number to move part zero to a position that is convenient for the machine operator.
- **G Code**—G codes have two basic functions:
  - specify a modal condition (example: G20 establishes Inch mode, G21 establishes Millimeter mode, G90 establishes Absolute mode and G91 establishes Incremental mode).
  - specify the type of tool motion (example: G00 programs a rapid move, G01 programs a linear feed move, G02 and G03 program circular moves).
- **X, Z, I, K**—Coordinates for programming geometric information needed to determine the endpoint of a motion command.
- **N Numbers**—Each block may contain a nine-digit integer N Number. N Numbers are not required; however, programmers can use them as sequence numbers to order the part program. It is not required for the numbers to appear in order within the part program.

- **Comment Statements**—Comment statements provide information about the part program.You can insert comment statements at the end of any part program block by enclosing the comment within parentheses. You may make an entire block of code a comment statement by enclosing it within parentheses.
- **Loop (or Repeat) commands**—Data enclosed in parentheses is normally treated as a comment. The only exception to this is the Loop (or Repeat) command. The Loop command, which must be enclosed in parentheses, causes a specified part of the part program to repeat a specific number of times.
- **Block Delete**—Block Delete codes specify blocks of code that are skipped when the Block Delete control feature is enabled. The specified block, or portion of a block, begins with a / (forward slash).

  The Block Delete control is activated from the Operator Control window.

Off-line Part Program Formats

This section describes how an NC post processor part program code should be configured on an off-line CAM system. The two items that are important to a post processor are: the order in which the control processes a block of code and the data format for each code.

Block Code Processing

A block of code can contain 0 to 10 G codes and 0 to 10 M codes. The table below shows the order in which the control will process a block of part program code. Except for block delete codes, which are processed at the point they appear in a block, the order in which each entry appears within a block of code does not affect the order of processing.

Description	Code
N Numbers	N123400000
Comment Statements	(this is a message)
Block Delete Codes	/, /0, /4
Job Syncing M Codes	M100 - M199
Specific G Codes	G201, G202, G40-42, G07-08, G20-21, G90-91, G59, G53, G58, G92, G93-95, G96-97
Feedrate Override	M48, M49
Beginning of Block M Codes	M07, M08
Tool and Fixture Commands	T code, E code
After Tool Change, Before Motion M Codes	M03, M04
Spindle Speed Command	S code
Dwell Command	G04
Feedrate or Dwell	F code
Motion Commands	G00, G01, G02, G03, G33
Exact Stop	G09
Spindle Stop	M05
End of Block System M Codes	M00, M01, M02, M30

Table 1. Order of Block Code Processing

Unused Data

Any unrecognized code will stop execution and display an error message.

Formats for Part Programs

Block data formats provide numerical values for the commands in a part program.

Distance and Feedrate Formats

Distance block data formats describe distances for axes and feedrates. This variable determines the range of numbers you can use when writing a program for a particular machine tool. The table below shows the different ranges for shifting distance formats.

Blocks	Optional[a] (inch)	Default (inch)	Optional (mm)	Default (mm)
Geometric Entries X Z I K R (range)	(±84.50000)	3.4 (±845.0000)	4.4 (±2146.3000)	5.3 (±21463.000)
Feedrate (feed/min) and (feed/rev) F (range)	2.5 (00.00001 - 84.50000)	3.4 (000.0001 - 845.0000)	4.4 (000.0001 - 2146.3000)	5.3 (0000.001 - 21463.000)
Thread Lead (distance/rev) K(range)	1.7 (0.0000001 - 1.000000)	2.6 (00.000001 - 10.000000)	2.6 (00.000001 - 24.000000)	3.5 (000.00001 - 254.00000)

a. This option shows the decimal point moved 1 place to the left

Table 2. Most Common Block Data Formats

⇨ **Format**: n.m means n number of places to left of decimal point and m number of places to right of decimal point.

For example, for inches, 3.4 is 3 places to the left of the decimal point and 4 places to the right. Metric would be 4.3 or 4 places to the elft of the decimal point and 3 places to the right.

Other Block Data Formats

Other block data formats are for programming G and M codes, rate of component motion or dwell, sequence numbers, tool codes, and fixture offsets.

Entry	Description	Format (range)
G	G Code	5 (0-65535)
M	M (Miscellaneous) Code Job Syncing M Codes	5 (0-65535) 3 (100-199)
S	Spindle Speed (rpm) and (css)	9 (0 - 2 x 10^9) (no practical limit)
T	Tool Code + Offset T00 cancels tool offsets	0 (0-999999999)
E	Fixture Offset Code E00 cancels any fixture offsets	4 (0-9999)
N	N Numbers	9 (0-999999999)
F	G93 inverse mode G94 FPM Mode G95 FPR Mode	4.3 inverse minutes (0.001 - 9999.999) See previous table See previous table
F	Dwell in G93 inverse mode Dwell in G94 fpm mode Dwell in G95 fpr mode	3.4 seconds (0.0001 - 999.9999) 3.4 seconds (0.0001 - 999.9999) 3.4 revs (0.0001 - 999.9999)

Table 3. Other Block Data Formats

⇨ **Format**: n means number of digits. n.m means n number of places to left of decimal point and m number of places to right of decimal point. For example, 3.4 is 3 places to the left of the decimal point and 4 places to the right.

Creating NC Part Programs

The code below shows what a typical NC part program looks like when you enter it using a text editor.

```
N20 G90 G40 G94 T0000         (absolute mode, CRC off, reset tool
                              offset)

N30 G00 X2.5 T0101            (rapid to X, tool 1 offset 1)

N40 Z3                        (rapid to Z)

N60 X.5                       (rapid to X)

N70 G96 G07 S350 M03          (CSS, radius programming,
                              X axis at .5in.)
                              (350 SFM and clockwise)

N80 G01 G95 F.01 X0           (face, IPR 0.01)

N120 G00 Z4                   (rapid to Z)

N130 X.5 Z3                   (rapid to XZ)

N160 G01 X1 Z2                (taper)

N170 X1.7                     (face)

N180 Z0.3                     (turn)

N190 X1.9                     (face)

N200 Z0                       (turn)

N230 G00 X2.5                 (rapid to X)

N240 Z3                       (rapid to Z)

N900 M30                      (end of program, rewind)
```

Coordinate System

The coordinate system for a typical back turret lathe is shown below.

The most important part of the illustration is the outer most rectangle, which shows boundaries defined by the position of the axes over travel limit switches. A high limit and low limit switch establishes the maximum travel for each axis. The term **machine envelope** refers to the boundaries established by these switches. If you move an axis onto one of these switches, an emergency stop occurs.

The **reference position** shows the point where the axes position at the end of a **reference zero** cycle (calibration). The axes may reference anywhere within the machine's envelope.

The dashed lines in the following drawing show the location of the machine's **software travel limits**. These limits are measured from the machine's reference position. Attempting to move an axis past a software travel limit stops all axes motion, but does not generate an emergency stop.

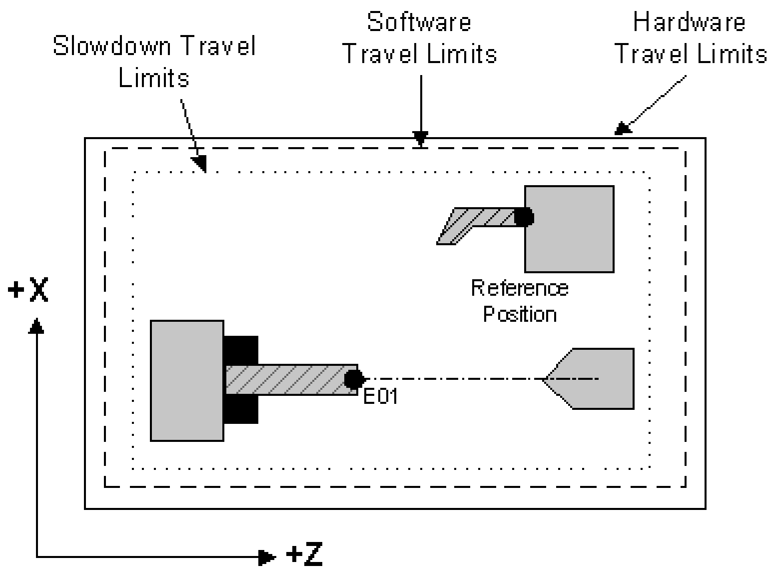


Figure 1. Coordinate System for Typical Back Turret Lathe

Slowdown limits located inside of the software travel limits may also be defined. Moving an axis within its slowdown limit restricts the move rate to a certain limit.

The point labeled **E01** (recorded in the fixture offset table) is the part zero location. You can move the fixture offset anywhere within the machine envelope. You may store up to 9999 different fixture offset positions and activate the one that you need to use.

G07/G08 Radius and Diameter Programming

You can program X axis dimensions using the radius or diameter of the part. Program **G07** when you wish to program radius dimensions; program **G08** when you wish to program diameter dimensions.

The table below provides information about how the modal setting will affect your part programs.

Programmed Value	G07 Modal	G08 Modal
Endpoint of linear move	X use radius	X use diameter
Endpoint of circular move	X use radius	X use diameter
Center and radius of circular move	IKR use radius	IKR use radius
X axis infeed distances	infeed distance use radius	infeed distance use radius

Table 4. Effects of Using G07 and G08

Absolute and Incremental Dimensions

You can use **absolute (G90)** or **incremental (G91)** dimensions directly from the blueprint. Most part programs should begin with a **G90** Absolute command that moves the tool to a fixed position on the machine. Once the start point is defined, you can study the part blueprint and determine whether to use **G90** Absolute or **G91** Incremental dimensions.

Absolute Dimensions

Absolute dimensions are measured from part zero. The figure shows the path that the tool center will follow; the actual part will be smaller by the radius of the tool.

Do not run this part on your machine; it is only intended to demonstrate the use of absolute dimensions.

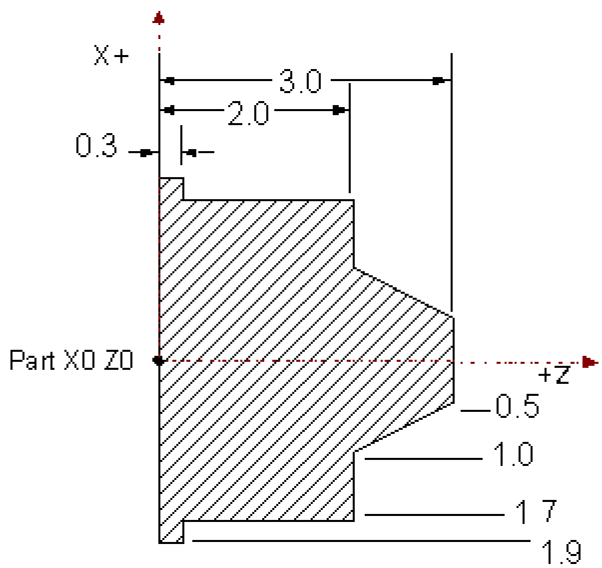


Figure 2. Absolute Dimensions

Select absolute dimension programming by entering **G90** in the part program. This setting remains active until you program **G91** to select incremental dimension programming.

Example—G90 Absolute Dimensions

```
N10                       (msg, lathe doc sample, abs G42, tool radius 0.1)

N20 G90 G40 G94 T0000     (absolute mode, CRC off, offset 0)

N30 G07 G00 X2.5 T0101    (radius programming,)
                          (rapid to X, tool 1 offset 1)

N40 Z3                    (rapid to Z)

N60 X.5                   (rapid to X)

N70 G96 S350 M03          (CSS, X axis at .5in,)
                          (350 SFM and clockwise)

N80 G01 G95 F.01 X0       (face, IPR 0.01)

N120 G00 Z4               (rapid to Z)

N125 G42                  (CRC right, skip if block del on)

N130 X.5 Z3               (rapid to XZ)

N160 G01 X1 Z2            (taper)

N170 X1.7                 (face)

N180 Z0.3                 (turn)

N190 X1.9                 (face)

N200 Z0                   (turn)

N230 G00 G40 X2.5         (CRC off, rapid to X, skip if)
                          (block del on)

N240 Z3                   (rapid to Z)

N900 M30                  (end of program, rewind)
```

Incremental Dimensions

Incremental dimensions specify the distance that the tool must move during each block (i.e. the distance from the start of the move to the end of the move).

Do not run this part on your machine; it is only intended to demonstrate the use of absolute dimensions.

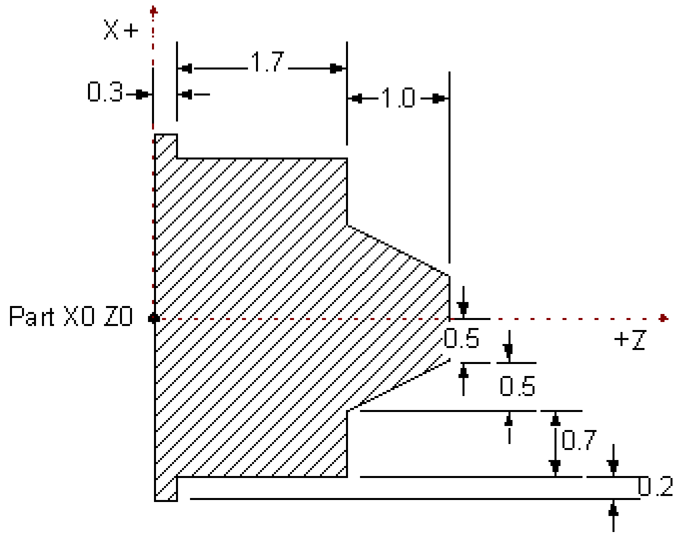


Figure 3. Incremental Dimensions

Enter **G91** in the part program to select incremental dimension programming. This setting remains active until you program **G90** to select absolute dimension programming.

Example: G91 Incremental Dimensions

```
N10                         (msg, lathe doc sample inc G42 tool rad 0.1)

N20 G90 G40 G94 T0000       (absolute mode, CRC off, offset 0)

N30 G07 G00 X2.5 T0101      (radius programming,)
                            (rapid to X, tool 1 offset 1)

N40 Z3                      (rapid to Z)

N60 G91 X-2                 (incremental mode, rapid to X)

N70 G96 S350 M03            (CSS, X axis at .5in,)
                            (spindle speed 350 and clockwise)

N80 G01 G95 F.01 X-.5       (face, RPM 0.01)

N120 G00 Z1                 (rapid to Z)

N125 G42                    (CRC right, skip if block del on)

N130 X.5 Z-1                (rapid to XZ)

N160 G01 X.5 Z-1            (taper)

N170 X.7                    (face)

N180 Z-1.7                  (turn)

N190 X.2                    (face)

N200 Z-.3                   (turn)

/N230 G00 G40 X.6           (CRC off, rapid to X, skip if)
                            (block del on)

N240 Z3                     (rapid to Z)


N900 M30                    (end of program, rewind)
```

G Codes

G Codes initiate axis motion, plane changes, and feedrate changes. G Codes are defined in this section.

G00 - Linear Motion at Rapid (default)

G00 sets rapid linear motion as the machine's modal condition. Any block that executes while **G00** is modal moves the axes to rapid to the programmed endpoint. The rate of each axis is controlled to ensure that all programmed axes reach their respective endpoints simultaneously.

G00 is cancelled by any G code in modal group 1.

G01 - Linear Motion at Feed

G01 sets feed linear motion as the machine's modal condition. Any block that executes while **G01** is modal moves the programmed axes directly to their programmed endpoint. The feedrate of each axis is controlled to ensure that all axes reach their endpoints simultaneously.

Example:

Assume G94 is active, if the X axis is programmed at a feedrate of 10 inches per minute (G01 X2 F10), the X axis feedrate will be 10 inches per minute. If both X and Z axes are programmed to move an equal distance at 10 inches per minute (G01 X2 Z2 F10), the X and Z axes will each move at 7.071 inches per minute to ensure that the feedrate along the cutting path is 10 inches per minute.

G01 is cancelled by any G code in modal group 1.

G02/G03 - Clockwise/Counterclockwise Circular Motion at Feed

G02 sets clockwise circular interpolation as the modal condition. **G03** sets counterclockwise circular interpolation as the modal condition. Either program the arc's endpoint and center coordinates, or program the arc's endpoint and radius.

Program Arc with Endpoint and Center Coordinates

This method requires you to define the arc's endpoint and center coordinates. Dimensions for the arc's endpoint (X, Z) may be defined with absolute or incremental dimensions, depending on the modal G90/G91 condition. All arc center dimensions (I, K) must be defined using signed incremental dimensions (the distance from the arc's start point to the arc's center).

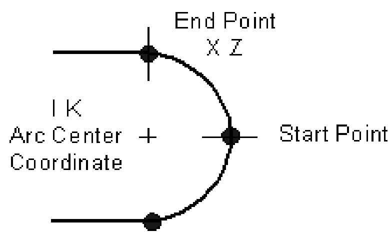


Figure 4. Arc with Endpoint and Center Coordinates

If **G07** Radius Programming is active, the arc's endpoint (X) and center (I) coordinates must be programmed as radius values.

If **G08** Diameter Programming is active the arc endpoint (X) must be programmed as a diameter value, but the arc center (I) must be programmed as a radius value.

Example—G02/G03 Arcs

The following example part program shows arcs programmed with X, Z, I, and K.

```
N1 G90 G20 G40
N2 G00 G07 X.0 Z.1 (G07 = radius programming)
N3 G01 Z0 F10
N4 X.2
N5 G03 X.8 Z-.6 I0 K-.6
N6 G01 G91 Z-1.0
N7 G90 G08 G02 X1.9512 Z-2.0243 R.6 (G08=diameter programming)
N8 G01 X2.2 Z-2.1487
N9 Z-2.5
N10 G00 X2.5
N11 Z.1
N12 M30
```

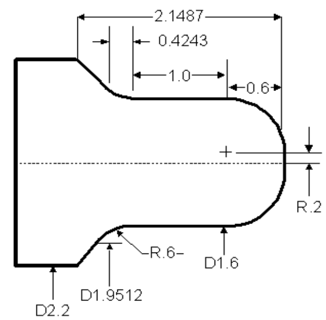


Figure 5. G02/G03 Arcs Example

Program Arc with Endpoint and Radius

The second method of programming a **G02** or **G03** block is to define the arc's endpoint and radius. There are two possible arcs when you program X, Z, and R. The sign of the R radius value determines whether you want the arc that spans less than 180° (by programming R as a positive value), or the arc that spans more than 180° (by programming R as a negative value).

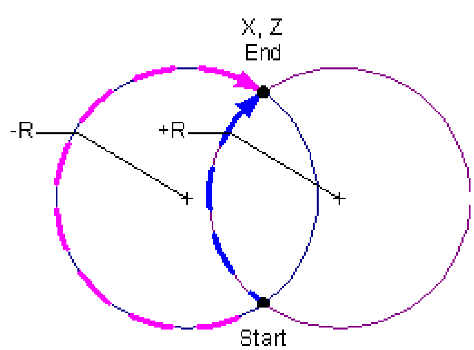


Figure 6. Arc with Endpoint and Radius

G04 - Dwell

G04 programs a pause in part program execution. F is the only entry allowed in a **G04** block.

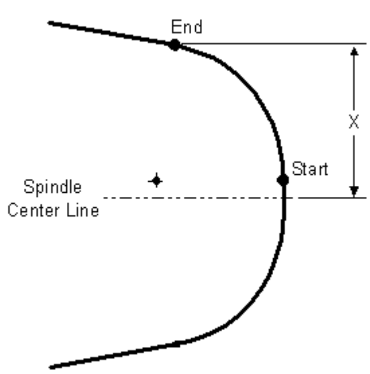
Active G Code	F in the G04 Block Programs
G93 Inverse Time Feed or **G94** Feed Per Minute	The number of seconds to dwell. For example, the block **G04** F2.1 pauses for 2.1 seconds before it executes the next block of code.
G95 Feed Per Revolution	The number of spindle revolutions to dwell. For example, the block **G04** F10 pauses for 10 spindle revolutions before it executes the next block of code.

G07/G08 - Radius Programming/Diameter Programming

G07, Radius programming, establishes a modal setting that interprets all cross-slide dimensions in radius values. **G08**, Diameter programming, specifies that all cross-slide positions are interpreted in diameter values. The arc center dimension (I) stays the same for both commands, but the cross-slide dimension (X) changes.The program may switch between **G07** Radius and **G08** Diameter at any time.

G07 Radius Programming

G08 Diameter Programming



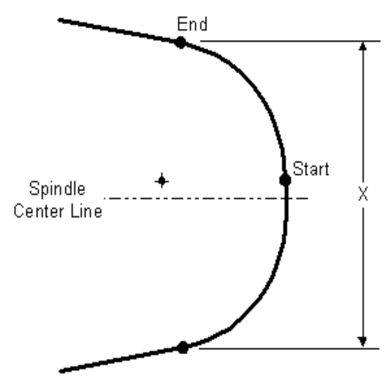


Figure 7. G07 Radius Programming

Figure 8. G03 Diameter Programming

G09 - Exact Stop

G09 moves the tool to its programmed endpoint before going to the next block in the part program. Since **G09** is non-modal, you must program it in every block that requires the tool to come to a complete stop.

G20 - Inch Mode

This code sets inch mode. If your part is dimensioned in inches, this command should be programmed in the first block of each part program to ensure that your part program dimensions are interpreted in the correct units.

G21 - Millimeter Mode

This code sets millimeter mode. If your part is dimensioned in millimeters, this command should be programmed in the first block of each part program to ensure that your part program dimensions are interpreted in the correct units.

Unit Display Mode

G20 and **G21** do not effect the Unit Display Mode as viewed in the position or offset screens. They interpret the correct units from the part program.

G33 - Threading

This code allows you to program a threading cycle. To program a threading move,

1. Program the thread's endpoint (using Z only, X only, or a combination of X and Z for a taper thread).
2. Program the lead of the thread using either of these codes:
    - **K** (the lead along the Z axis)
    - **I** (the lead along the X axis).

⇨ If you are programming a tapered thread (both X and Z axes move) you must program the lead (I or K) for the axis that has the longest move.

You can also start a thread at a given angle. The thread start angle is specified by a **P** value in a **G33** block, such as:

```
G0 X3.5 Z4          (Position for thread)
G33 Z1 K.2 P180     (Cut thread)
G1 X3.6             (Pullout of thread)
```

The angle value may range from -360.0000 to 360.0000. A zero angle starts the thread at the index mark of the spindle encoder, and is the default. The P value is not modal and must appear in the same block as the **G33**.

G38 - Turret Probe

There are four cycles within Turret Probing. Use the S parameter to differentiate between the turret probe cycles.

- S0 Datum X or Z
- S1 Set/Adjust Fixture Offset (X, Z)
- S2 Measure Diameter (ID, OD:X) or Part Length (Z)
- S3 Adjust Tool Offsets (X, Z)
- All turret probe cycles are defined by a common G-code, **G38**, to specify a turret probe cycle.
- The X and Z positions passed to any turret probe cycle must be in **part** coordinates to get the desired results.

Turret Probe Calibration - S0

- The turret probe datum cycle defines the datum offsets to be used for X and Z when measuring part surfaces in other turret probe cycles for most accurate results.
- The turret probe datum cycle allows you to specify which axis and direction (-X, -Z, +X, and/or +Z) to datum. In the other turret probe cycles, axis measurements are only permitted for those axes/directions that have been successfully calibrated.
- For the turret probe datum cycle, program **(S0)**.
- One G38 macro call per axis and direction must be programmed to datum the probe. The surface touch point for the probe must be known and specified in the macro. X and Z must be pre-positioned before the datum macro call. Only the "probe align axis" moves during the cycle. When the cycle is finished, this "probe align axis" will be back at its macro start point.

Format of Sample Recorded Datum Cycle Data

hh:mm:ss Probe Datum: Probe #14; -X Offset 0.0827 at X2.5247, Z7.5000

Parameters

Letter Parameter	Definition
S	**0** - Specifies Turret probe calibration cycle

Must select one of the following: X, Z	
X	Known datum contact point in part program coordinates for X
Z	Known datum contact point in part program coordinates for Z

Must be specified as the default or within the G38 block: A,F,T,V	
A	Approach distance from expected contact point at which feed rate changes from programmed rate **F** to probe rate **V**
F	Feedrate from approach point until contact
T	Tolerance about expected contact point
V	Feedrate to approach point and away from contact point

Optional	
P	Probe Number

Figure 9. Turret Probe Calibration—S0 Parameters

⇨ To datum in X, **(X)** is programmed. The current X position determines which direction the X datum is to occur.

To datum in Z, **(Z)** is programmed. The current Z position determines which direction the Z datum is to occur.

If Probe Number is neither defined in the tune file nor programmed in the macro call, a value of P-1 is recorded.

Macro Operation

X-Datum

- X feeds to known contact point ± approach position.
- X is commanded to feed to known contact point ± specified tolerance. Fault is flagged with the macro aborted unless probe contact is detected between **(X - Tol)** and **(X +Tol)**. Datum done in +X direction or datum done in -X direction is set to (X – actual contact position for X) when valid contact detected.
- X feeds back to its start position.

Z-Datum

- Z feeds to known contact point ± approach position.
- Z commanded to feed to known contact point ± specified tolerance. Fault is flagged with the macro aborted unless probe contact is detected between (**Z - Tol**) and (**Z + Tol**). Datum done in +Z direction or datum done in -Z direction is set to (Z – actual contact position for Z) when valid contact detected.
- Z feeds back to its start position.

Datum in -X and -Z example:

```
H00 E00
G00 G40 X10 Z5
G38 S0 X5 A.5 T.125 V20 F10
G0 Z10
G38 S0 Z7.6 A.5 T.125 V20 F10
```

In the above example:

If default values had been specified were appropriate for **A, T, V, F** for this datum cycle, the first G38 call block above could have been shortened to:

```
G38 S0 X5
```

- After contact is made during the X-datum at Z5, X returns to its start position (X10; F20).
- After contact is made during the Z-datum at X10, Z returns to its start position (Z10; F20).
- Start point when part program continues after the macro call is at X10, Z10 for this example.

Set/Adjust Fixture Offset (X,Z) - S1

- This cycle incrementally adjusts or absolutely sets the fixture offset for the specified axis and E-code based on the measured surface of a part for that axis.
- The **absolute** fixture offset value stored will be equal to the measured surface for the given axis in **part coordinates** altered by the adjusted offset specified for the axis in the macro call block, if any.
- The **incremental** adjustment to the specified axis fixture offset table value equals the difference between the expected and actual contact points on the surface of a part for the designated axis.
- The programmer must pre-position the "other axis" to the probe align position. This "other" axis is not moved within the macro.
- The **X** or **Z** expected contact point is programmed in the macro call block. The corresponding "axis offset" parameter (**I** or **K**) **may** also be programmed if fixture offset entry is absolute. This "axis offset" parameter is the incremental distance from the contact point to the program zero point for the axis. For example, if you want to call the probe contact position 5 inches in program coordinates, program an I or K value of -5 inches.
- To measure in X, program **XExp (X)** and, optionally, **Xadj (I)**. **Xadj (I)** may only be used with absolute fixture offset entries.
- To measure in Z, program **ZExp (Z)** and, optionally, **Zadj (K)**. **Zadj (K)** may only be used with absolute fixture offset entries.
- If the cycle is aborted due to premature or no probe contact, the probing axis is fed back to its start position at the approach feed rate **(FRAp, V)**.

Format of fixture offsets cycle recorded data

hh:mm:ss Probe Fixture X: E19 new: 3.2629 old: 3.2600

⇨ The expected contact point (X, XExp) and fixture offset adjust number (**I**, XAdj) for X must always be programmed in the same units (radius or diameter) as is modally active (G7, G8) in the control at the time the macro is called.

Parameters

Letter Parameter	Definition
S	**1** - Find part edge to adjust fixture offset

Must select one of the following: X, Z	
X	X expected contact point in part program coordinates
Z	Z expected contact point in part program coordinates
Required	
E	E-code for which designated axis entry is to be set or altered

Must be specified in either tune file as the default or within the G38 block: A,F,T,V	
A	Approach distance from expected contact point at which feed rate changes from programmed rate **F** to probe rate **V**
F	Feedrate from approach point until contact
T	Tolerance about expected contact point
V	Feedrate to approach point and away from contact point

Optional	
I	X fixture offset adjust number if absolute entry
K	Z fixture offset adjust number if absolute entry
Q	Incremental fixture offsets if 1

Figure 10. Fixture Offset (X,Z) - S1 Parameters

Macro Operation

- The designated axis is programmed to move to the approach point (expected contact position ± approach distance) at the approach feed rate.
- This axis is programmed to feed at the probe feed rate to the expected contact point ± tolerance band **(Tol, T)**.
- **Absolute entries**: When valid contact is made, the actual contact point in part coordinates plus the given axis adjust value, if any, is stored in the fixture offset table for the specified E-code value and axis letter.
- **Incremental entries**: When valid contact is made, the measured contact point – expected contact point is added to or subtracted from the specified axis fixture offset table value for the specified E-code value and axis letter.
- When the cycle is finished, the measuring axis is back at its macro start position.
- The altered fixture offset does not become effective until the designated select code value is reactivated.

Sample blocks

```
T01 E00 H01
G00 X--- Z---
G38 S1 E3 Z0 K-1 A.5 T.125 V20 F10
```

Measure Diameter or Part Length - S2

- This cycle measures the outer or inner diameter (X) or length (Z) of a part and record the results.
- The software determines the align direction from the macro start position and expected contact point.
- You must pre-position the "other axis" to the probe align position. This other axis is not moved within the macro.
- The expected contact point for the measuring axis (X, Z) is programmed in the macro call block.
- If the cycle is aborted due to premature or no probe contact, the probing axis is fed back to its start position at the approach feed rate.

Format of "measure diameter or part length" cycle recorded data

hh:mm:ss Probe Measure X: machine: 3.2629 part: 3.2600

Program Parameters

Letter Parameter	Definition
S	2 - Measure diameter (OD, ID: X) or length (Z) and record the results

Must select one of the following: X, Z	
X	X expected contact point in part program coordinates
Z	Z expected contact point in part program coordinates

Must be specified as the default or within the G38 block: A,F,T,V	
A	Approach distance from expected contact point at which feed rate changes from programmed rate **F** to probe rate **V**
F	Feedrate from approach point until contact
T	Tolerance about expected contact point
V	Feedrate to approach point and away from contact point

Figure 11. Measure Diameter or Part Length - S2 Parameters

⇨ The macro automatically determines whether an OD or ID (X) is to be measured based on the direction of the expected contact point from the macro start point for X. You must make sure that X and Z have been properly positioned before the macro is called.

The expected contact point for X must always be programmed in the same units (radius or diameter) as is modally active **(G7, G8)** in the control at the time the macro is called.

Macro Operation

- The designated axis is programmed to move to the approach point (expected contact position ± approach distance) at the approach feed rate**)**.
- This axis is programmed to feed at the probe feed rate to the expected contact point ± tolerance band **(Tol, T)**.
- When contact is made, the specified axis contact point in both machine and part coordinates is recorded.
- When the cycle is finished, the given axis is returned to its macro start position.

Sample blocks

```
T01 E00 H01
G00 X--- Z---
G38 S2 X0 A.5 T.125 V20 F10
```

Adjust Tool Offsets (X, Z) - S3

- This cycle measures the outer or inner diameter (X) or length (Z) of a part, adjusts the specified tool offset table value, and records the results.
- The software determines the align direction from the macro start position and expected contact point.
- You must pre-position the other axis to the probe align position. This other axis is not moved within the macro.
- The expected contact point for the measuring axis (X, Z) is programmed in the macro call block.
- If the cycle is aborted due to premature or no probe contact, the probing axis is fed back to its start position at the approach feed rate.

Format of adjust tool offset cycle recorded data

hh:mm:ss Probe Offset Z: T13 new: 0.9468 old: 0.9437

Parameters

Letter Parameter	Definition
S	**3** - specifies adjust tool offset (X, Z)

Must select one of the following: X, Z	
X	X expected contact point in part program coordinates
Z	Z expected contact point in part program coordinates

Required	
H	H-code for which the designated axis entry is to be altered

Must be specified as the default or within the G38 block: A,F,T,V	
A	Approach distance from expected contact point at which feed rate changes from programmed rate **F** to probe rate **V**
F	Feedrate from approach point until contact
T	Tolerance about expected contact point
V	Feedrate to approach point and away from contact point

Figure 12. Adjust Tool Offsets (X, Z) - S3 Parameters

⇨ The macro automatically determines whether an OD or ID (X) is to be measured based on the direction of the expected contact point from the macro start point for X. You must make sure that X and Z have been properly positioned before the macro is called. The expected contact point for X must always be programmed in the same units (radius or diameter) as is modally active **(G7, G8)** in the control at the time the macro is called.

Macro Operation

This cycle operation is identical to **(S2)**, but at the end of the cycle, this cycle incrementally adjusts the tool offset table entry (by the difference between the expected and measured contact points). Also, the recorded message formats of the two cycles differ as noted above.

Sample blocks

```
T01 H00 E00
G00 X--- Y---
G38 S3 X1.5 H8 A.5 T.125 V20 F10
```

G40 - Cutter Radius Compensation Off (default)

This code turns off cutter compensation. When **G40** is active, the tool center will move along the path defined in the part program.

G41 - Cutter Radius Compensation Left

This code turns cutter compensation on to the left of the programmed part profile. When **G41** is active, the tool path is offset by the radius value stored in the tool table. The offset will be to the left side of the part profile.

G42 - Cutter Radius Compensation Right

This code turns cutter compensation on to the right of the programmed part profile. When **G42** is active, the tool path is offset by the radius value stored in the tool table. The offset will be to the right side of the part profile.

⇨ **G41** and **G42** are similar to those used for milling a better approach may be to use Tool Orientations within the tool offset file. These are activated by using the tool offset call.

Cutter Radius Compensation Activation

The tool path varies depending on whether the cutting path is inside or outside the part. The table below shows the most common use of the variables. The following Table shows the possible actions that can be performed by the machine tool.

Variable Sets	Result
Activate and Exit Normal on	The tool feeds to a point that is perpendicular to the programmed start and end points.
Activate and Exit Normal off	The start and end points for the next tool motion are the current location of the tool.
Circular Joining on	Tool applies circular joining around outside corners.
Circular Joining off	Tool follows a series of linear blocks around outside corners.

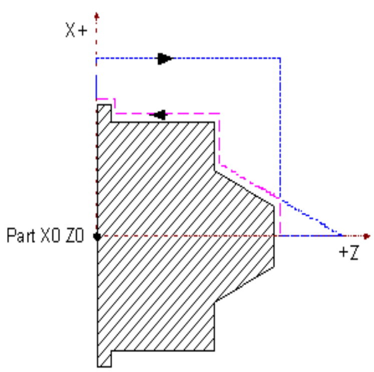
Figure 13. Activate and Exit Actions; Circular Joining Actions

Examples

The following examples use the part program below. One example turns cutter compensation on, and the other example turns cutter compensation off. Figure 14. Cutter Compensation Right Turned Off (G40), on page 30 shows the tool path with cutter compensation turned off. Figure 15. Cutter Compensation Turned On (G42), on page 30 shows the tool path with **G42** (Cutter Compensation Right) turned on.

```
N10                       (msg, lathe doc sample, abs G42, tool rad 0.1)

N20 G90 G40 G94 T0000     (absolute mode, CRC off, offset 0)

N30 G07 G00 X2.5 T0101    (radius programming,
                          (rapid to X, tool 1 offset 1)

N40 Z3                    (rapid to Z)

N60 X.5                   (rapid to X)

N70 G96 R.5 S350 M03      (CSS, axis at .5in.,)
                          (350 SFM and clockwise)

N80 G01 G95 F.01 X0       (face, IPR 0.01)

N120 G00 Z4               (rapid to Z)

N125 G42                  (Cutter Comp Right)

N130 X.5 Z3               (rapid to XZ)

N160 G01 X1 Z2            (taper)

N170 X1.7                 (face)

N180 Z0.3                 (turn)

N190 X1.9                 (face)

N200 Z0                   (turn)

N230 G00 G40 X2.5         (CRC off, rapid to X)

N240 Z3                   (rapid to Z)

N900 M30                  (end of program, rewind)
```

Cutter Compensation Right Turned Off (G40



Cutter Compensation Turned On (G42

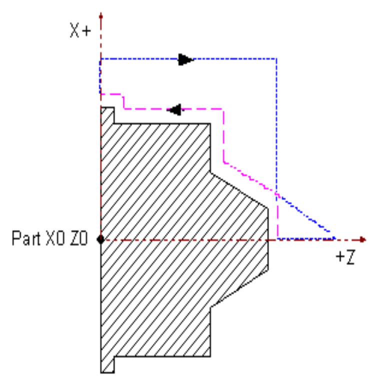


Figure 14. Cutter Compensation Right Turned Off (G40)

Figure 15. Cutter Compensation Turned On (G42)

G53 - Program Machine Coordinates

This code allows you to program the axes to a position that you define using machine coordinates. This code causes the part program to ignore the following offsets for one block:

- active work coordinate offsets (G92)
- active tool offset
- active fixture offset

Before you program **G53** in a block, the program must meet the following conditions:

- Cutter radius compensation (CRC) must be off.
- Absolute mode (G90) must be active.
- Either rapid (G00) or linear (G01) mode must be active.

G59 - Cancel Work Coordinate Offsets

This code cancels any active offset set by **G92** Work Coordinate Offsets.

G73 - Peck Drill with Chip Break Drill Cycle

Machine Integrator Configurable

G73 sets chip break drilling as the modal drill cycle. The tool motion for the chip break drill cycle is shown in Figure 16. Tool Motion for G73 Chip Break Drill Cycle. This cycle causes the tool to feed by the amount programmed in Q, then rapid back an incremental amount A. Next the tool will rapid to a distance slightly above the last infeed depth, then feed another Q distance into the part. This sequence repeats until the tool reaches the programmed Z depth.

Tool motion for **G73** chip break drill cycle - Tool returns to R plane between each peck:

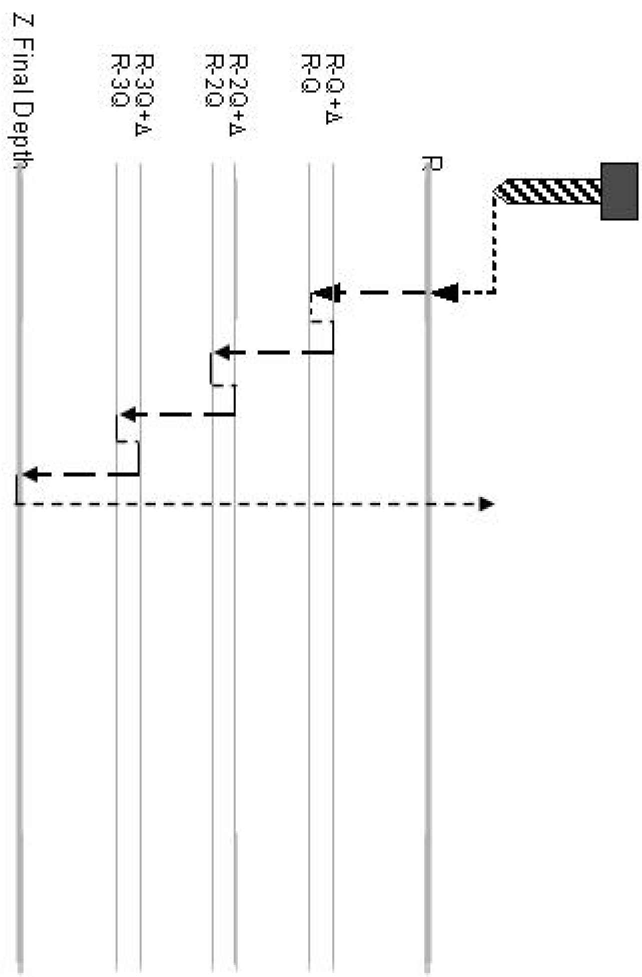


Figure 16. Tool Motion for G73 Chip Break Drill Cycle

Example:

```
G90
G73 X2 Y2 Z-1.2 R.1 Q.5 G99
```

G74 - Left Hand Tapping

G74 executes the same as G84 Right Hand Tapping except the spindle directions are reversed.

G78 - Threading Cycle

This feature provides a multi-pass Threading Cycle for OD and ID threads. This cycle supports taper threads, face threads, lead in and lead out angles, multiple spring passes, and various methods of specifying depth of cut.

To make the part program easier to read, you may program this cycle using single letter entries (e.g. A30, Z3.1) or multi-letter entries (anglIn30, ZEnd3.1). The second method is easier to read, while the first method is typically output by CAM systems.

⇨ Prior to executing this cycle, the spindle must be running at speed in G97 mode and the tool must be positioned so it can rapid directly to the thread's X and Z clearance position.

Turn Vs. Face Threading

The type of threading is determined by the **M** (Face) parameter.

Parameter Value	Description
M=0	Indicates a Turn Thread.
M=1	Indicates a Face Thread.
Default Value:	M=0

Table 5. Face Parameter

Turn Threading - Inner Diameter Vs. Outer Diameter

The type of thread cut (Inner or Outer Diameter) is defined by the signs of the **X** and **J** parameters. The **X** parameter (X, minR, minD) specifies the radius or diameter at the start of the final thread pass. The **J** parameter specifies the total depth of the thread.

- if (X, minR, minD) and (J) are of the same sign (both positive or both negative), an OD (outer diameter or external) thread cut will be implemented.
- If (X, minR, minD) and (J) are of the opposite sign (one positive and one negative), an ID (inner diameter or internal) thread cut will be implemented.

⇨ If the sign of **X+J** is not the same as the sign of **X**, the macro is aborted with an appropriate fault message displayed. The X-axis is not allowed to cross the spindle centerline when cycling a threading macro.

Required Parameters for Turn Threads

The following entries are required to define a turn thread.

J	Total depth of cut from the major radius to the minor radius.
X	Radius at the start of the final thread cut at the end of the infeed motion. Effective final pass radius = X or minR.[a] Diameter at the start of the final thread cut at the end of the infeed motion. Effective final pass radius = minD/2.[b]
K	Z position where thread starts.
F	Pitch.Inches/millimeters per revolution.
F-	Lead. Threads per inch. (F-8 is same as F.125)

a. The interpretation of this parameter is not affected by the Radius/Diameter Mode state (G7 or G8).
b. The interpretation of this parameter is not affected by the Radius/Diameter Mode state (G7 or G8).

Table 6. Turn Thread Parameters

You must program the following parameters to specify the length of the thread.

T	Length of thread between the infeed move and pullout move.
Z	The Z end position at X Clearance (T + Z = K)
H	The Z end position at the major radius

Table 7. Thread Length Parameters

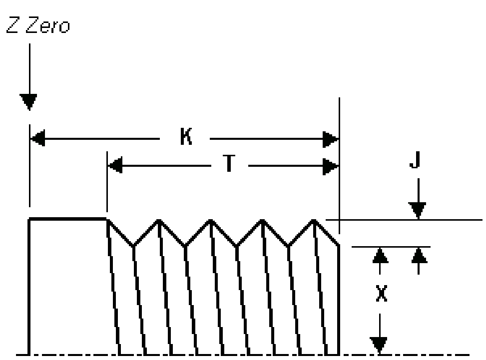


Figure 17. Thread Parameters Example

Clearance, Chase In and Chase Out Parameters

These parameters have default values and may not need to be programmed.

U	X return clearance. Measured from the major diameter. Default 0.1" or 2.54 mm.
W	Z clearance. Measured from the start of thread. Default 0.1" or 2.54 mm.
A	Lead in Angle. This is the infeed (chase in) angle with respect to the X axis. Default 0°
C	Lead out Angle. Pullout (chase out) angle with respect to X axis. Default 0°.

Table 8. Clearance and Chase Parameters

OD Thread with Offsets U and W Example

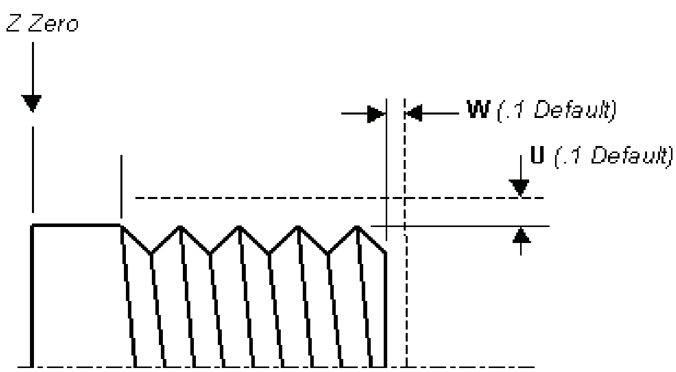


Figure 18. OD Thread with Offsets U and W Example

OD Thread with Chase in (A) and Chase out (C) Example

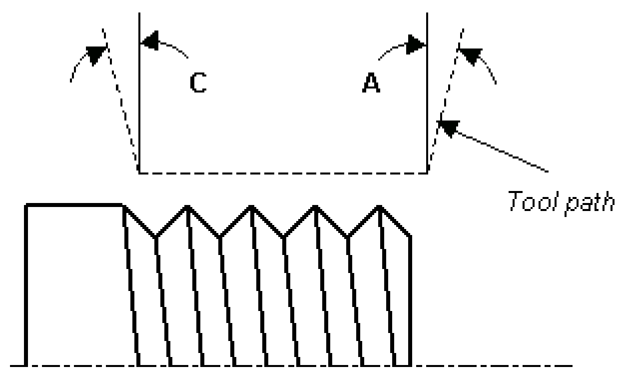


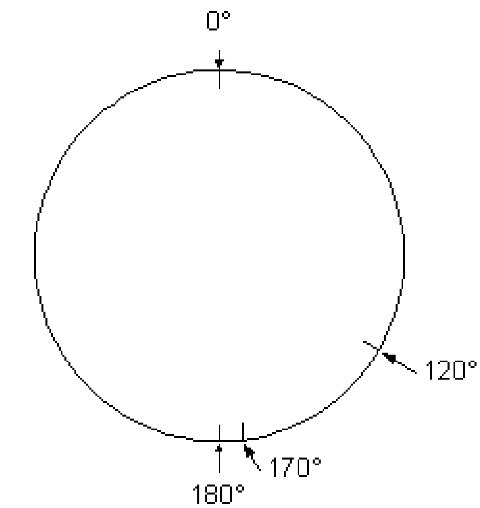
Figure 19. OD Thread with Chase in (A) and Chase out (C) Example

Multiple Start Parameters

These parameters allow you to define multiple starting points at different angles from 0 - 359.999° around the thread. This parameter is optional.

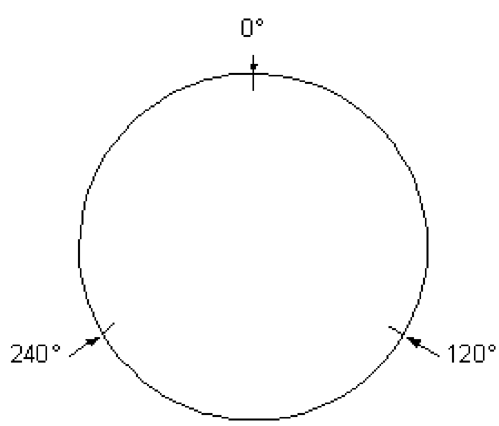
Q	Thread cut start angle. Default 0°.
+Q	Multiple Start number. Delta angle per thread start equals 360/Q.

Table 9. Multiple Start Parameters



G78 Q-120 + other parameters
G78 Q-170 + other parameters
G78 Q-180 + other parameters

Figure 20. OD with multiple starts at 120°, 170°, and 180° Example



G78 Q3 + other parameters

Figure 21. OD with 3 Evenly Spaced Threads Example

Tapered Thread

For a tapered thread, you must program **B** to specify the taper angle with respect to the Z axis for turn thread, or X axis for face thread.

B	Taper angle relative to the Z axis. -45° <= B <= 45°

Table 10.Tapered Thread Parameter

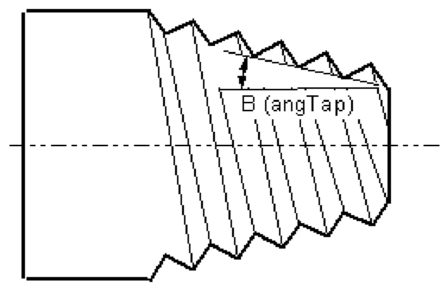


Figure 22. Tapered Thread Example

Cutting Parameters

The following cutting parameters affect how the thread will be cut. You can specify the start and final rough depth, number of finish passes, or the number of spring passes.

D	Start rough depth. **Required.**
D	Final rough depth. If E is not programmed, the value defaults to D unless V is programmed.
L	Depth of cut per finish pass. If L is not programmed, then there are no finish passes. If P is nonzero and L equals zero, then the number of spring passes equals P + R.
P	Number of finish passes implemented before starting spring passes. If P is not programmed and L is nonzero, set L to 1. Otherwise, no finish passes will occur.
R	Number of spring passes at 0 depth per pass.
V	When set to 1, the depth per pass results in constant volume removal per pass. When programming constant volume, do not program a final depth parameter (E) in the same block.

Table 11.Thread Cutting Parameters

No Finish Pass Programmed—G78 X1.05 J.45 D.1 E.05

Only the start rough depth (D) and final rough depth (E) are programmed. The tool moves progressively deeper from D to E through a number of control-calculated passes.

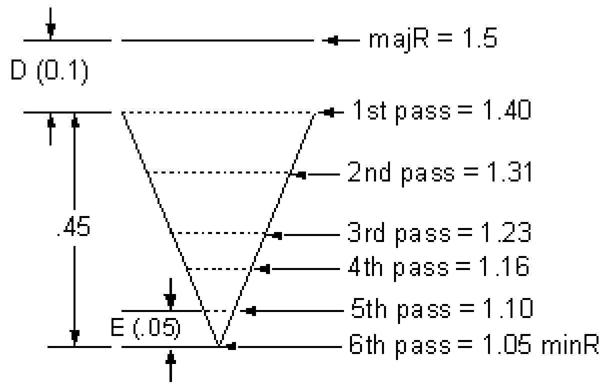


Figure 23. No Programmed Finish Pass Example

Three Finished Passes Programmed—G78 X1.05 J.45 D.1 E.05 L.01 P3

Three finish passes (L) are programmed along with the D and E values. The tool moves progressively deeper from D to E through a number of control-calculated passes, then moves in for three consecutive finishing passes.

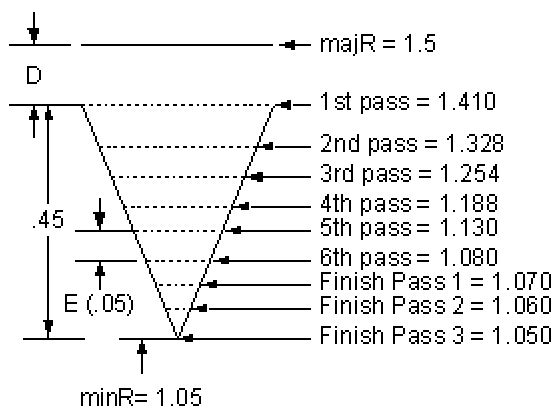


Figure 24. Three Programmed Finished Passes Example

Sample Thread Part Programs

The following sample code shows four types of threads:

Straight OD Thread, on page 40

Straight OD Thread with Lead In/Lead Out Angles, on page 41

Tapered OD Thread, on page 42

Straight ID Thread, on page 43

⚠ Do not run these sample part programs on your machine, they are intended only to demonstrate the types of threads. The drawings are for example only and are not to scale.

Straight OD Thread

This example shows a straight thread cut with:

- Variable depth per pass
- Two finish passes of .02" per finish pass
- Three spring passes

```
(Thread cut example - straight thread)
G70 G90 M03 S120
G0 Z4.1
G78 X.9 J.2 Z1.2 K2 F.15 D.04 E.03 P2 L.02 R3
M30
```

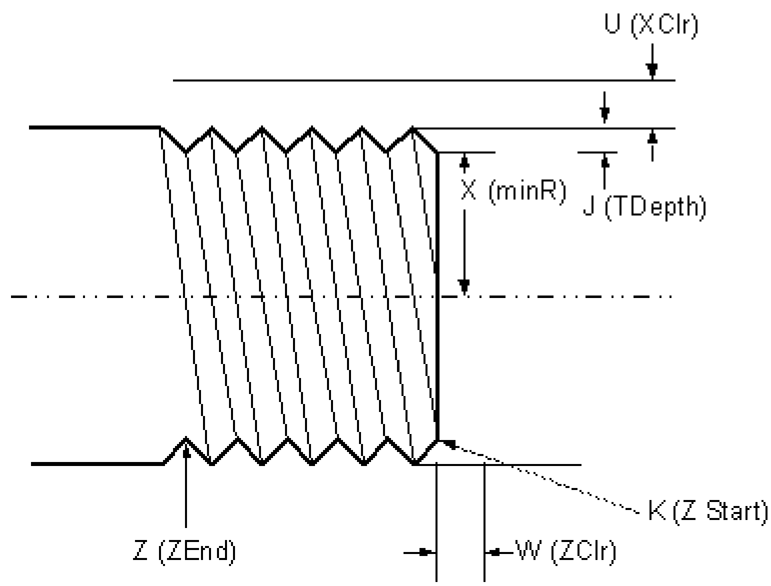


Figure 25. Straight OD Thread Example

Straight OD Thread with Lead In/Lead Out Angles

This example shows a straight thread cut with:

- Lead In angle (A) of 30°
- Lead Out angle (C) of 5°

```
(Thread cut example - straight thread with lead in/lead out angles)
G70 G90 M03 S120
G0 X4.1 Z2.5
G78 X.9 J.6 Z1.2 K2 A30 C5 F.15 D.04 E.03 P2 L.02
M30
```

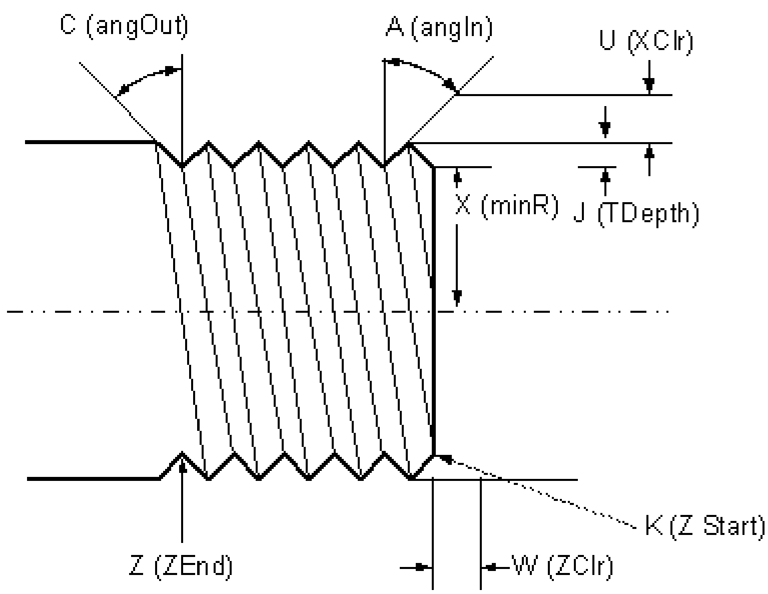


Figure 26. Straight OD Thread with Lead In/Lead Out Angles

Tapered OD Thread

This example shows a tapered thread cut with a Taper angle (B) of 14.7°.

```
(Thread cut example - tapered thread)
G70 G90 M03 S120
G0Z4.1
G78 X.9 J.6 B14.7 Z1.2 K4 F.375 D.1 E.08 P2 L.02 R3
M30
```

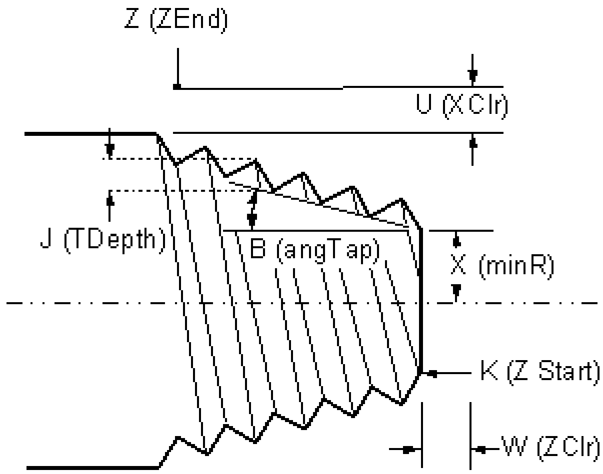


Figure 27. Tapered OD Thread Example

Straight ID Thread

This example shows a straight thread cut with:

- Variable depth per pass
- Two finish passes of .02" per finish pass

```
(Thread cut example - straight ID thread)
G70 G90 M03 S120
G0 Z4.1
G78 X.9 J-.2 Z1.2 K2 F.15 D.04 E.03 P2 L.02
M30
```

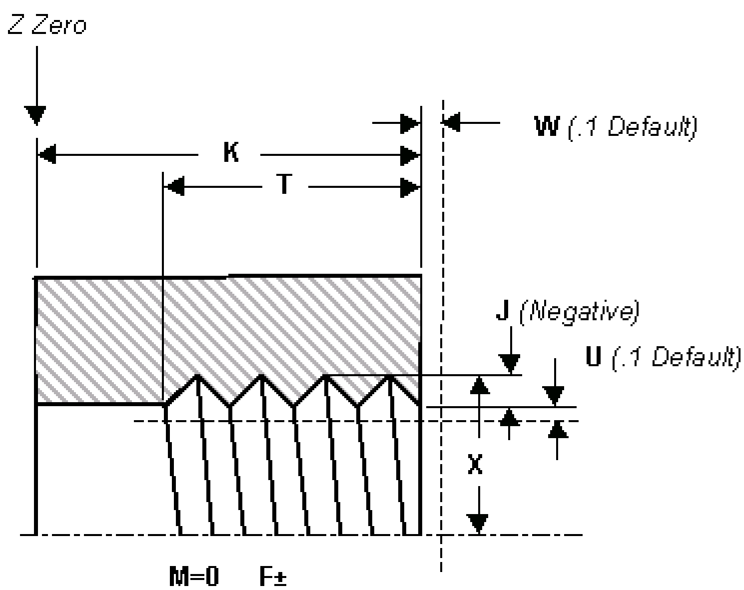


Figure 28. Straight ID Thread Example

Threading Equivalents

The following equivalents exist between the turn and face thread macro arguments.

Thread Axis	Turn Thread	Face Thread
Thread Start	K	I
Thread Normal End	Z	X
Optional Thread End	H	S
Optional Thread End #2	Thread Start + T	Thread Start + Td

Depth Axis	Turn Thread	Face Thread
Last Pass Depth	X	Z
Initial Depth	Last Depth + J	Last Depth + J

Table 12.Threading Equivlents between Turn and Face Thread Macros

1. The thread start, last pass depth, and total depth arguments (K, X, and J for turn thread; I, Z, and J for face thread) must be programmed for each thread cycle.
2. Exactly one of the three thread arguments (Z, H, or T for turn thread, X, S, or T for face thread) must be programmed for the given type of thread cycle (turn or face).

Face Threading

The following entries are required to define a face thread.

M	M1 is required to activate the face thread. No M-value or M=0 activates the turn thread. The M default value is 0.
J	Total depth of cut (End depth = Z, Start depth = Z+J.)
Z	Z-Position at the start of the final thread pass at the end of the infeed motion.
I	X-position where thread starts.
F	Pitch. Inches/millimeters per revolution.
F-	Lead. Threads per inch. (F-8 is same as F.125)

You must program the following to specify the length of the thread.

T	Length of thread between the infeed move and pullout move.
X	The X end position at Z Clearance (Z +J + W).
S	The X end position at Z start depth (Z+J).

Table 13.Face Threading Parameters

G80 - Cancel Drill Cycle

This code cancels any active drilling cycle. When **G80** is active X and Z motion commands move the tool directly to the programmed endpoint without performing a drilling cycle. In addition to turning off the drilling cycle, **G80** activates Linear Motion at Rapid and clears the modal drilling parameters such as R, Z, Q, and P.

If the drill cycle depth is not programmed after **G80**, then the drill cycle depth defaults to the R plane.

G00 cannot be programmed in the same block.

G81 - Drill Cycle

G81 sets drilling as the modal drill cycle. The tool motion for the drill cycle is shown below.

1. Position to XY
2. Rapid to reference plane
3. Feed to depth
4. Rapid to reference plane

Example:

```
N010 G90
N020 G81 X2 Y2 Z-1.2 R.1 F10 G99
```

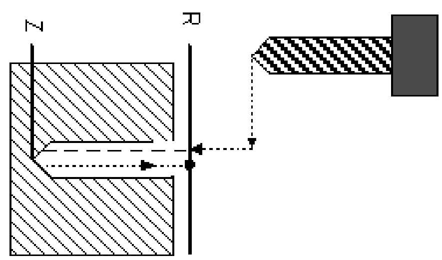


Figure 29. G81 Drill Cycle

G82 - Drill Cycle with Dwell

G82 sets drilling with dwell as the modal drill cycle. The tool motion for this cycle is the same as the drill cycle except a dwell for P seconds occurs at the bottom of the hole.

1. Position to XZ.
2. Rapid to reference plane.
3. Feed to depth.
4. Dwell for P seconds or revolutions (depending on FPM or FPR mode).
5. Rapid to reference plane.

Example:

```
G90
G82 X2 Z-1.2 R.1 F10 P2.2 G99
```

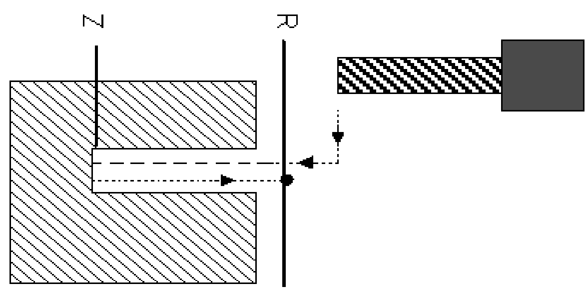


Figure 30. G82 Drill Cycle with Dwell

G83 - Peck Drill Cycle

G83 sets peck drilling as the modal drill cycle. This cycle causes the tool to feed by the amount programmed in Q, then rapid back to the R plane. Next the tool rapids to a distance slightly above the last infeed depth, then feed another Q distance into the part. This sequence repeats until the tool reaches the programmed Z depth.

Tool motion for **G83** peck drill cycle - Tool returns to R plane between each peck:

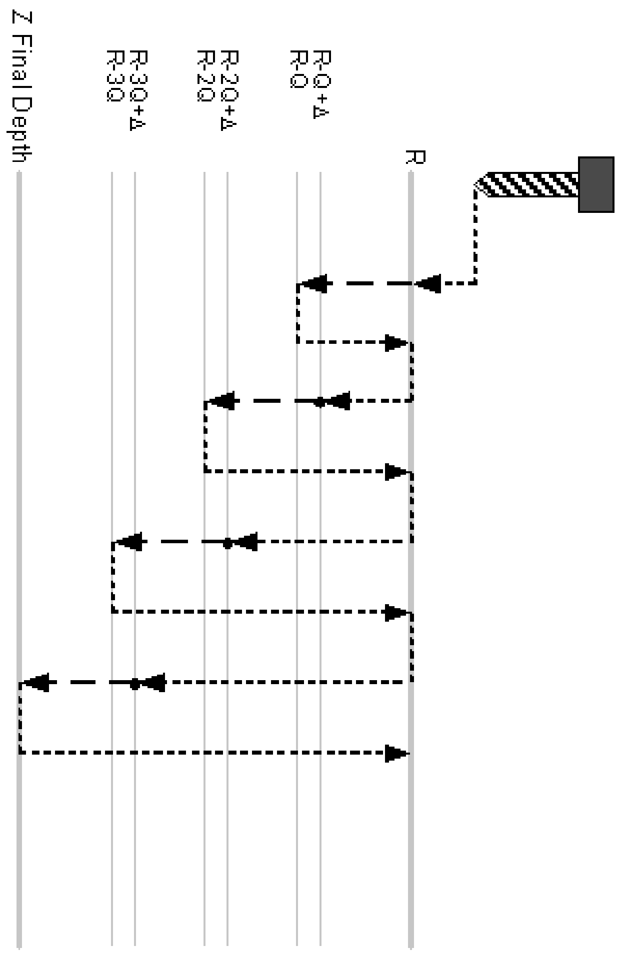


Figure 31. G83 Peck Drill Cycle

Example:

```
G90
G83 X2 Z-1.2 R.1 Q.5 G99
```

G84 - Right Hand Tapping and Float Tapping

G84 sets right-hand tapping as the modal drill cycle. The tool motion for this cycle is described below. The feedrate and spindle override controls to 100% are automatically set while this cycle is active.

⚠ There are two ways to stop tool motion when the tap is in the bore. One is to press emergency stop; the other is to use retract. The control will not respond to a motion-stop command until the tool returns to the reference plane.

The drawing below shows right-hand tapping tool motion:

1. Tool rapids to R plane.
2. Tool feeds to the programmed depth.
3. Spindle reverses direction.
4. Control dwells for P seconds if G94 is active or for P revolutions if G95 is active.
5. Tool feeds to R plane.
6. Spindle reverses direction

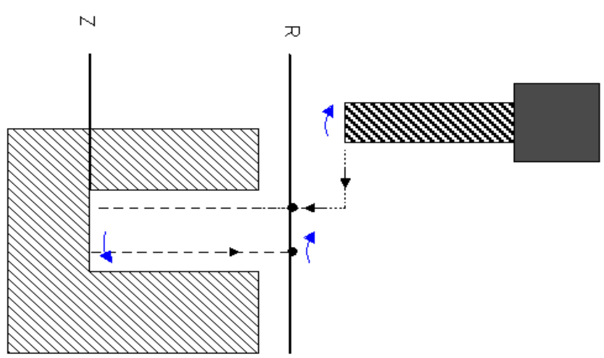


Figure 32. G84 Right Hand Tapping and Float Tapping

Deep Hole Tapping

Standard Tapping, using a floating tap holder, has added capability for deep hole tapping. Tapping may now specify in Q the depth for each in-feed. If Q is zero, the in-feed is to final depth. If Q is greater than zero, the in-feed for each pass is determined by Q. When Q is greater than zero, the return position is either back to the R plane, or specified by the tune variable. The return is always to the R plane when tune variable is zero.

G90 - Absolute Programming (default)

G90 establishes a modal setting that tells the control to interpret all XZ endpoint dimensions in absolute coordinates. **G90** Absolute Programming specifies that all tool endpoint positions are measured from the current part zero position.

Most part programs should begin with a **G90** command to establish part zero. A program may switch between **G90** Absolute mode and **G91** Incremental mode at any time.

G91 - Incremental Programming

G91 establishes a modal setting that tells the control to interpret all XZ endpoint dimensions in incremental coordinates. **G91** Incremental Programming specifies that all tool endpoint positions are measured from the tool's position at the start of the motion. A program may switch between **G90** Absolute and **G91** Incremental at any time within a part program.

G92 - Work Coordinate Offsets or Spindle Max Speed

G92 lets you establish the part program coordinates at the current position without generating any tool motion.

⇨ Work coordinate offsets are cancelled using G59.

G92 can also set a maximum spindle speed for constant surface speed mode.

Example

The code **G92** S5000 will prevent the spindle from exceeding 5000 rpm.

G93 - Inverse Time Feed

G93 puts the control into inverse time mode. When **G93** is modal, the F value represents 1/(minutes of cut).

Any dwell command that is programmed while **G93** is active will be interpreted as the number of seconds to dwell (the same as **G94** feed per minute mode).

When programming short times for long distances, remember that an axis will not move faster than its maximum rate.

To determine the F value, use the formula:

Inverse Time F value = (Feedrate)/(Distance of Move)

Example 1

To determine the F value for a four-minute cut, use this equation:

F = 1/(4 minutes) = 0.25

and program a block with F = 0.25:

```
N100 G93 X2 F.25
```

Example 2

To determine the F value for a 20-second cut, use this equation:

F = 1(20 seconds) = 1/.333333

and program a block with F = 3:

G94 - Feed per Minute (default)

G94 puts the control into feed per minute mode. When **G94** is modal, all F feedrate values are interpreted in inch/minute or millimeter/minute units. Any dwell command that is programmed while **G94** is active will be interpreted as number of seconds to dwell.

G95 - Feed per Revolution

G95 puts the control into feed per revolution mode. When **G95** is modal, all F feedrate values are interpreted in inch/revolution or millimeter/revolution units. Any dwell command that is programmed while **G95** is active will be interpreted as number of spindle revolutions to dwell.

G96 - Constant Surface Speed (CSS)

G96 sets feet per minute or meters per minute as the modal spindle command mode. When G96 is programmed, the spindle speed will increase as the tool tip moves toward the spindle centerline and decrease as the tool moves away from the spindle centerline. When **G96** is modal, the programmed S value is read as a feet per minute or meters per minute command. If the application uses multiple spindles, CSS is applied to all spindles in the same job stream.

To specify a radius of the tool tip that corresponds to the surface speed given in S, program the R value in the same block as G96.

- If you do not program an R value or the R value is 0, the control uses the X-axis position as the radius.
- If an R value has been previously programmed, and you program a **G96** without an R value, the previously programmed R value will be used.

Active Programming Mode	R Value
G07 Radius Programming	distance from the tool tip to the spindle centerline (radius)
G08 Diameter Programming	twice the distance from the tool tip to the spindle centerline (diameter)

Table 14.R Values and G96 Programming

G97 - Direct Spindle Speed (default)

G97 sets direct RPM as the modal spindle command mode for the job stream in which it is programmed. When **G97** is modal, the programmed S value is revolutions per minute. If the application uses multiple spindles, the control sets direct RPM for all spindles in the same job stream.

When using multiple spindles, the S code (commanded speed) applies to all spindles in a job stream unless it is used with a spindle designator. Using the spindle designator with an S code specifies an RPM for a specific spindle in the current job stream.

G98 - Drill Cycle Initial Level Return (default)

When **G98** is modal, all G81-G89 drill cycles command the tool to return to the Z axis coordinate where the tool was located.

If the initial Z coordinate is changed for subsequent cycles, the Z coordinate for the first cycle after G80 is used as the reference plane.

For example, in the following two blocks, the tool will rapid to Z10 in the second block even though Z20 (init plane) is specified:

```
N010 G0 Z10
N020 G81 Z5 F60        (feed to Z5, rapid to Z10)
N030 G0 Z20
N040 G81 Z15           (rapid to Z10, feed up to Z15, rapid to Z10)
N050 G80
```

To return to the init plane, **G98** requires a **G80** Cancel Drill Cycle to change to the init plane. For example:

```
N010 G0 Z10
N020 G81 Z5 F60        (feed to Z5, rapid to Z10)
N030 G80 Z20
N040 G81 Z15           (rapid to Z10, feed up to Z15, rapid to Z20)
N050 G80
```

G99 - Drill Cycle R Plane Return

When **G99** is modal, all G81-G89 drill cycles command the tool to return to the modal R Reference plane at the end of the drill cycle.

General Drill Cycle Information

G codes in the G81-G89 range perform some form of a drilling, tapping, or boring cycle. The drilling will occur along the Z axis. The tool motion that occurs at the end of each cycle depends on the modal status of **G98/G99**.

When **G99** is modal, the tool will return to the R Reference plane at the end of the block. When **G98** is modal, the tool will return to the Z axis coordinate where the tool was when the cycle was initiated.

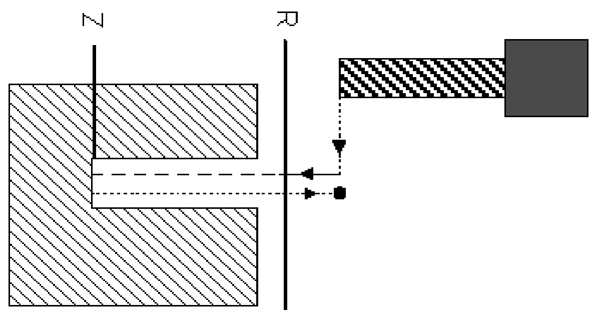


Figure 33. General Drill Cycle Tool Motion

Above - Drill Cycle Motion with G98

Tool returns to Z axis coordinate where tool was when cycle was initiated.

If the initial Z coordinate is changed for subsequent cycles, the Z coordinate for the first cycle after **G80** is used as the reference plane.

To return to the init plane, **G98** requires a **G80** Cancel Drill Cycle to change to the init plane.

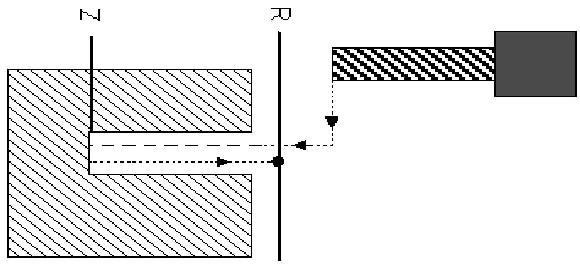


Figure 34. Drill Cycle Motion with G98

Above - Drill Cycle Motion with G99

Tool returns to R reference plane.

M Codes

The following M codes are available on most systems.

To help ensure that your CNC performs commands as expected, review the final command definitions with the machine integrator.

M codes belong to one of four execution categories. The execution categories are listed below.

- Execute before tool change or motion - example: M08.
- Execute after T and S codes, but before motion - examples: M03, M04.
- Execute after tool change or motion - example: M09.
- Execute after tool change or motion (system M codes) - examples: M00, M05, M30.

Understanding each M code's execution category is important when the M code is programmed in a block that also contains a T code, S code, E code, and XZ axes motion.

M00 - Program Stop

M00 causes:

- part program execution to stop.
- the spindle to turn off.
- the parser look-ahead program to finish processing the blocks in the parser look-ahead buffer and stop looking ahead for more blocks.

The operator must press the **START CYCLE** button to resume program execution and restart the look-ahead program.

M01 - Optional Stop

M01 performs the same functions as M00, but can be toggled on or off through the Opt Stop key on the control.

- **Off**—CNC ignores **M01**.
- **On (locked)**—CNC acts on **M01** and keeps it active.
- **On (turn off)**—CNC acts on **M01** then toggles it off.

M02 - End of Program (no rewind)

M02 stops program execution and leaves the part program pointer on the block that follows the **M02** command. **M02** commands the spindle to turn off. **M02** executes at the end of the block (after all motion occurs).

M03 - Spindle Clockwise

M03 turns the spindle on in the clockwise direction. **M03** will execute after T, S, and E codes execute, but before motion programmed in the block occurs.

An S code specifying the spindle speed must be programmed before the spindle will turn on. The advanced RPM Look Ahead feature that reduces part cycle time by ramping the spindle up to speed before the feed move begins improves the cycle time because the control does not dwell while the spindle ramps up to the programmed speed. When this feature is active, it is possible to create a part program containing an **M03** and spindle speed command that does not turn on the spindle.

M04 - Spindle Counterclockwise

M04 is the same as **M03** except it commands the spindle to rotate in the counterclockwise direction.

M05 - Spindle Off

M05 tells the control to turn the spindle off. **M05** executes at the end of the block after all motion occurs.

M07 - Secondary Coolant On

M07 commands the secondary coolant pump to turn on. Typically the second coolant pump is used to provide mist coolant. This command is cancelled by an **M09** command. **M07** is usually configured to execute at the beginning of the block before any motion occurs.

M08 - Primary Coolant On

M08 commands the primary coolant pump to turn on. Typically, the primary coolant pump is used to provide flood coolant. This command is cancelled by an **M09** command. **M08** is usually configured to execute at the beginning of the block before any motion occurs.

M09 - Coolant Off

M09 command tells the control to turn the primary and secondary coolant pumps off. **M09** is usually configured to execute at the end of the block after all motion occurs.

M30 - End of Program (rewind)

M30 stops program execution and moves the part program pointer to the top of the part program. **M30** commands the spindle to turn off. **M30** is configured to execute at the end of the block after all motion occurs.

M48 - Use feedrate override

M48 uses the value specified by the Feedrate Override (FPM %).

M49 - Ignore feedrate override

M49 ignores the value specified by the Feedrate Override (FPR %) and uses the programmed feedrate.

M200 - Block Delete Synchronization

M200 allows you to turn on block delete while the part program is executing. Using this code ensures that blocks marked with block delete codes are skipped after block delete has been turned on, without requiring you to stop and restart the program.

Block Delete Code

Block Delete will skip specified blocks of code when turned on.

Programming Block Delete

To skip a block of code, place a slash at the front of the block. Multiple block delete levels are allowed, but most controls only use one level, as shown in the example below.

```
N10                        (msg, lathe doc sample, abs G42, tool rad
                           0.1)
N20 G90 G40 G94 T0000      (absolute mode, CRC off, offset 0)
N30 G00 X2.5 T0101         (rapid to X, tool 1 offset 1)
N40 Z3                     (rapid to Z)
N60 X.5                    (rapid to X)
N70 G96 G07 R2 S350 M03    (CSS, radius programming,)
                           (part radius 2in.,)
                           (350 SFM and clockwise)
N80 G01 G95 F.01 X0        (face, IPR 0.01)
N120 G00 Z4                (rapid to Z)
/N125 G42                  (CRC right, skip if block del on)
.
.
.
/N230 G00 G40 X2.5         (CRC off, rapid to X, skip if)
                           (block del on)
N240 Z3                    (rapid to Z)
N900 M30                   (end of program, rewind)
```

To program different levels, place a slash and a single digit at the front of a block. The levels range from 0 to 9. When a level number is specified, all blocks containing the same level are skipped.

```
/0N7 X3.0                  (skip block if level 0 specified)
/1N7 X3.0                  (skip block if level 1 specified)
/8/9N7 X3.0                (skip block if level 8 or 9 specified)
```

Activating Block Delete

The parser look-ahead program processes blocks before they appear on the screen. If you change block delete mode while a part program is running, the control may have already processed blocks marked for deletion. The control will not go back and reprocess these blocks.

To ensure that block delete is applied to the marked blocks, there are three M code options for synchronizing program execution with the parser look-ahead program.

⇨ When programming with **M00**, **M01**, or **M200**, program the codes so they can be activated when the tool is off the part. Using these codes when in a cutting mode may cause the tool to leave dwell marks.

M Code	Software Process	Required Operator Response	What Happens Next
M200 Block Delete Synchronization	• Waits for parser look-ahead buffer to empty. • Prevents parser look-ahead program from looking further ahead until after it processes the M200 code.	None	Program execution **continues** and block deletes are recognized.
M00 Program Stop	• Stops part program execution • Turns off the spindle • Waits for the parser look-ahead buffer to empty. • Prevents the parser look-ahead program from looking further ahead until program execution is re-started	Press Cycle Start	Program execution **re-starts** and block deletes are recognized.
M01 >Optional Stop **OptStop** On	• Stops part program execution • Turns off the spindle • Waits for the parser look-ahead buffer to empty. • Prevents the parser look-ahead program from looking further ahead until program execution is re-started.	Press Cycle Start	Program execution **re-starts** and block deletes are recognized.
M01 >Optional Stop **OptStop** Off	• Waits for parser look-ahead buffer to empty. • Prevents parser look-ahead program from looking further ahead until after it processes the M01 code.	None	Program execution **continues** and block deletes are recognized.

Table 15.M COde Options for Look-Ahead Program Execution

Tool Change and Active Tool Offsets

This section describes how to program a tool change and activate the tool's offsets. Basic information about the tool change sequence is provided below.

Tool Change Sequence

A typical tool change sequence is shown below.

```
T0303
M03 S1200
```

On most machines, the T code will cause the turret to index into position. It is the responsibility of the part programmer or machine operator to ensure that the turret is in a safe position to perform an index.

On some machines the part programmer must precede this sequence with the commands required to move the axes to the machine's tool change position.

The format for a T code is:

T mmmmmmmnn

where:

mmmmmmm is the tool number

nn is the tool offset number

The last two digits identify the tool table offsets to use for the current tool. The preceding digits in the T code identify the physical tool that must be indexed into position. If your machine does not have an automatic turret indexer, or if your machine does not have an automatic tool changer, an M code is usually programmed and the T code or tool name will be listed in part program comments to inform you which tool must be inserted into the tool holder.

A **T00** cancels any tool offsets.

Tool Motion Following a T Code

Tool offsets are programmed with tool offset T codes. The coordinates for each tool offset are stored in the Tool Offset table. All dimensions in the tool offset table represent dimensions measured from machine coordinate zero to the desired tool offset position. A tool offset value is provided for both axes.

Execution of a tool offset T code does not cause tool motion unless axis motion is programmed in the block that contains the T code, or until motion is programmed in a following block. This applies to E codes as well.

⚠ To avoid incompatibilities between part programs created on different Hurco controls, you should make it a part programming practice to program all axes in the block that programs a tool offset, or in the first motion block after a tool change.

The way tool motion following a tool offset will occur is selected with one of the options listed in the following table. This selection also applies to the manner in which fixture offset E codes are implemented. The drawings that follow show how the tool will move depending on which option has been selected.

Option	Motion
Tool Offset: Immediate Activation	All axes will move to a position relative to the new tool offset as soon as any axis is programmed.
Tool Offset: Deferred Action	Only the axes that are programmed will move to a position relative to the new tool offset.

Table 16.Tool Offset Options

Tool Offset Behaviors

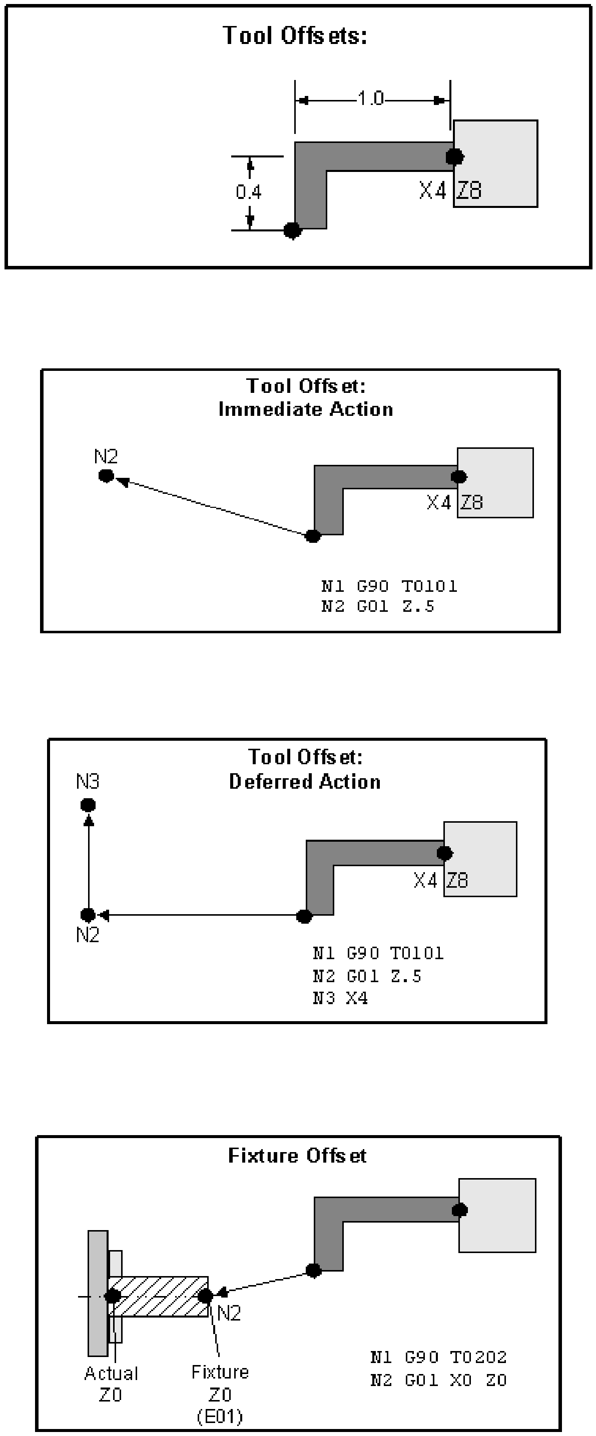


Figure 35. Tool Offset Behaviors

E Codes

Offsets from machine zero are programmed with Fixture Offset **E** codes. The coordinates for each fixture offset are stored in the Fixture Offset table. All dimensions in the Fixture Offset table represent dimensions measured from a machine coordinate to the desired fixture offset position. **E00** cancels any fixture offsets. A fixture offset value is provided for each axis that is installed on your machine.

Execution of a Fixture Offset **E** code does not cause tool motion unless axis motion is programmed in the block that contains the **E** code, or until motion is programmed in a following block.

Fixture Offset E Code

There are three options for positioning Z0 at a convenient position along the Z axis. The option used depends on your machine-tool configuration:

- All machines can use **G92** to establish the part program coordinates at the current position without generating any tool motion.
- Depending on how the variables have been set:
    - Fixture Offset Tables are functional.
    - You cannot directly enter data into the Fixture Offset table. You can enter an Z axis offset value into Fixture Offset #1 (E1). Then the offset is automatically activated whenever a nonzero Tool offset is active. The Z axis offset is removed when the Tool offset is cancelled.

Fixture Offset Immediate Activation

When this option is selected, both axes move to a position relative to the new fixture offset as soon as any axis is programmed.

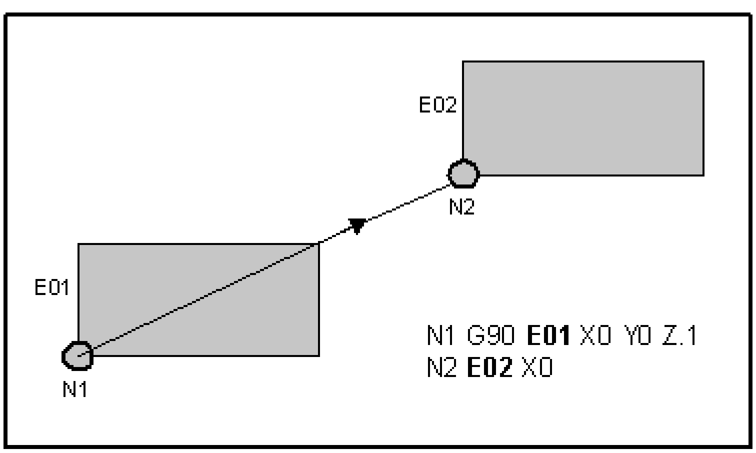


Figure 36. Fixture Offset: Immediate Activation

Fixture Offset Deferred Activation

When this option has been enabled, the programmed axis moves to a position relative to the new fixture offset.

To avoid incompatibilities between part programs created on different Hurco controls, you should make it a part programming practice to always program all axes in the block that programs an **E** code.

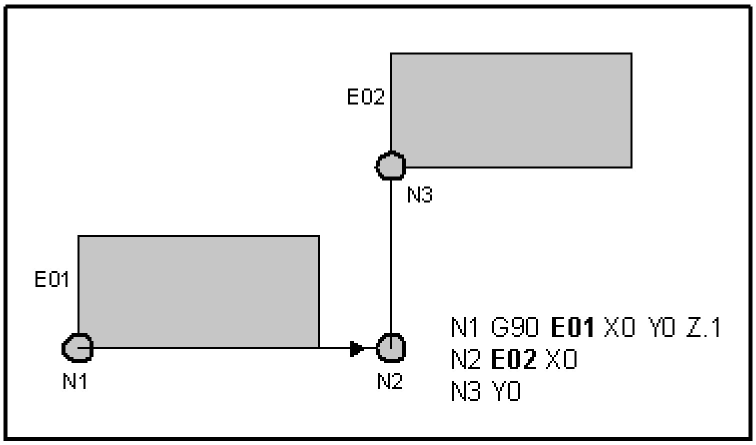


Figure 37. Fixture Offset: Deferred Activation

INDEX

U

W

Z