CHAPTER 11
G Codes

G Code		Group	Description
G00	*	A	Rapid Positioning
G01		A	Linear Interpolation
G02		A	Circular or Helical Interpolation CW
G03		A	Circular or Helical Interpolation CCW
G04		B	Dwell
G10		B	Parameter Setting
G20	*	K	Select Inch Units
G21		K	Select Metric Units
G22		O	Work envelope on
G23	*	O	Work envelope off
G28		B	Return to Reference Point
G29		B	Return from Reference Point
G30		B	Return to Secondary Reference Point
G32		A	Constant Lead Thread Cutting
G40	*	D	Cutter Diameter Compensation Cancel
G41		D	Cutter Diameter Compensation Left
G42		D	Cutter Diameter Compensation Right
G50		B	Coordinate System Setting, Max. Spindle Speed Setting
G52		B	Offset Local Coordinate System
G53		B	Rapid Position in Machine Coordinates
G54		L	Select Work Coordinate System #1
G55		L	Select Work Coordinate System #2
G56		L	Select Work Coordinate System #3
G57		L	Select Work Coordinate System #4
G58		L	Select Work Coordinate System #5
G59		L	Select Work Coordinate System #6
G65		J	Call Macro
G70		B	Profile Finishing Cycle
G71		B	Profile Stock Removal in Turning
G72		B	Profile Stock Removal in Facing
G73		B	Profile Path Repeating Cycle
G74		B	End Face Peck Cutting Cycle
G75		B	Outer/Inner Diameter Peck Cutting Cycle
G76		B	Multi-Pass Threading Cycle
G80	*	B	Cancel Canned Cycle
G83		B	Deep Hole Drilling
G84		B	Tapping
G85		B	Boring Cycle
G90		A	Outer/Inner Diameter Cutting Cycle
G92		A	One-Pass Threading Cycle
G94		A	End Face Cutting Cycle
G96		H	Constant Surface Speed
G97	*	H	Constant Surface Speed Cancel
G98		I	Per Minute Feed
G99	*	I	Per Revolution Feed

NOTES:

●All the default G Codes have been marked with the symbol " * ".

●A given line of a program may contain more than one G code.

●If several G codes from one group are used in the same line, only the G code specified last will remain active.

●G codes from group B are of "one shot" type (active only in the line in which they are specified). All other G codes are modal (active until another G code of the same group is specified).

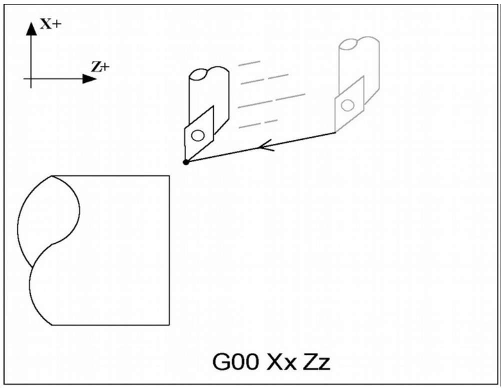
G00 - Rapid Positioning

G0 moves to the specified position at the maximum motor rate. The coordinates may be either absolute positions or incremental distances. G0 is modal and remains in effect until another positioning mode (G1, G2, G3 etc.) is commanded. G0 is the default-positioning mode.

Example:

```
   G0 X0.0 Z0.0
```

This command moves both X and Z to the absolute coordinate 0.0 at maximum feedrate.



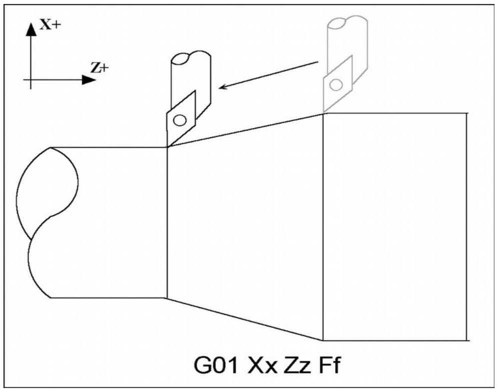
CAUTION The feedrate override knob has no effect on G0 moves unless rapid override is turned ON

G01 - Linear Interpolation

G1 moves to the specified position at the programmed feedrate. The coordinates may be either absolute positions or incremental distances. The movement will be along a straight line. G1 is modal and remains in effect until another positioning mode (G0, G2, G3 etc.) is commanded.

Example:

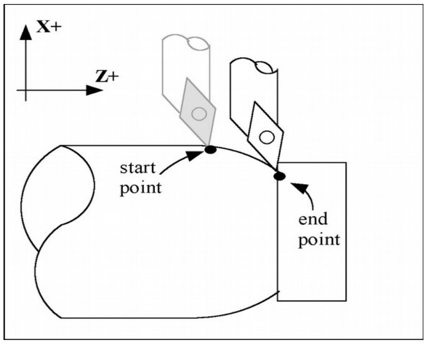
```
   G01 X2 Z4 F10
   G01 X6 Z3 F20
```



G02 & G03 - Circular Interpolation

G2 moves in a clockwise* circular motion, and G3 moves in a counterclockwise* circular motion. The X or Z position specified in the G2 or G3 command is the end position of the arc, and may be an absolute position (X, Z) or an incremental distance (U, W). G2 and G3 are modal and remain in effect until another positioning mode (G0, G1, etc.) is commanded.

Circular motion can be programmed in two different ways: specifying the final point and the radius of the arc, or specifying the final point and the parameters I and K (center point of the arc as incremental values from the start position).



*The terms clockwise and counterclockwise can be somewhat confusing because they are relative directions which change based on ones perspective. To help conceptualize the correct perspective, always program your part and set up your tools as though the machine were a horizontal lathe with the tool post mounted in the rear and the head stock to your left.

Rules of thumb:

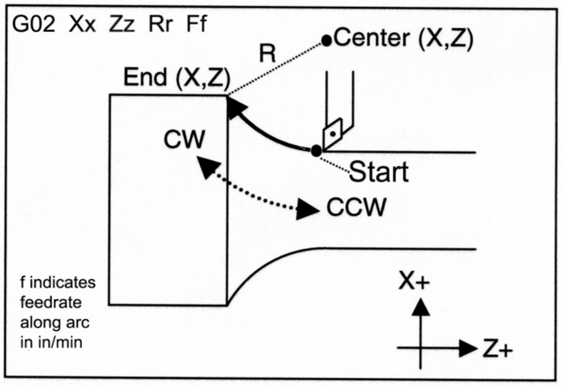
1. All Convex OD Arcs which move towards a more negative Z position should be programmed as CCW.
2. All Concave OD Arcs which move towards a more negative Z position should be programmed as CW.
3. All Convex OD Arcs which move towards a more positive Z position should be programmed as CW.
4. All Concave OD Arcs which move towards a more positive Z position should be programmed as CCW.
5. All Convex ID Arcs which move towards a more negative Z position should be programmed as CCW.
6. All Concave ID Arcs which move towards a more negative Z position should be programmed as CW.
7. All Convex ID Arcs which move towards a more positive Z position should be programmed as CW.
8. All Concave ID Arcs which move towards a more positive Z position should be programmed as CCW.

METHOD 1: USING FINAL POINT AND RADIUS

The commands G2 and G3 will have the following structure:

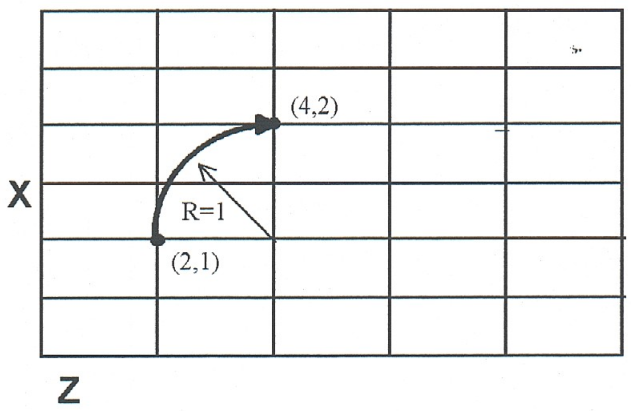
```
G2 Xx Zz Rr
G3 Xx Zz Rr
```

where *x* and *z* will be the X and Z coordinates of the final point of the arc, and *r* will be the radius.



Example:

```
G00 X2.0 Z1.0     ;rapid to start
                  ;position X2, Z1
G02 X4.0 Z2.0 R1  ;arc to X4 Z2 with
                  ;radius of 1
```



NOTE: A lathe is not usually used to cut an arc larger than 90 degrees. With the use of special tools, a lathe can cut a 180-degree arc. This is the maximum value a lathe can cut an arc. Make sure the radius chosen follows the cutting ability of the lathe.

METHOD 2: USING FINAL POINT AND PARAMETERS I & K

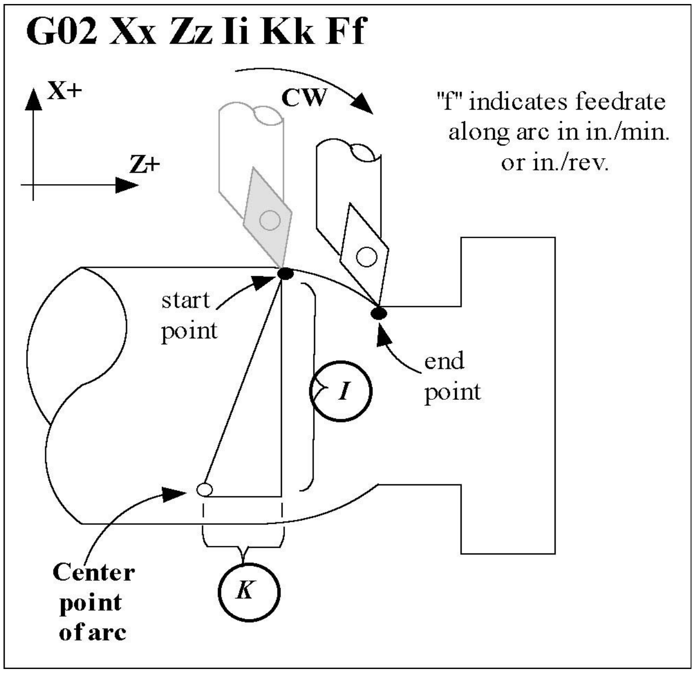
Another way to specify a circular operation is using the parameters I and K instead of the radius R. The parameters I and K are the **incremental** distances from the start point to the center of the arc.

I = X center (radius) - X start (radius)
K = Z center - Z start

● NOTE: X coordinates are diameter values, but I and R are always radius values.

Example:

```
   G00 X2.0 Z1.0     ;rapid to start
                     ;pos. X2, Z0
   G02 X4.0 Z2.0 K1  ;arc to X4 Z1
                     with radius 1
```



G04 - Dwell

G4 causes motion to stop for the specified time. The P parameter is used to specify the time in seconds to delay. G4 causes the block to decelerate to a full stop.

The minimum delay is 0.01 seconds and the maximum is 327.67 seconds. The dwell time is performed after all motion and M functions on the line. If the P parameter is not specified, X will be used instead. If neither P nor X is specified, the default dwell time of 0.01 seconds will be used.

Example:

```
   G0 X1 Z1  ; rapid to X1 Z1
   G4 P2.51  ; pause for 2.51 seconds
   G0 X2 Z2  ; rapid to X2 Z2
```

G10 - Parameter Setting

G10 allows you to set parameters for different program operations.

Examples:

```
G10 P5 Z-1.1      ; Sets tool #5 z offset to -1.1 in the Offset Library
G10 P5 X-1.3      ; Sets tool #5 x offset to -1.3 in the Offset Library
G10 P5 R.25       ; Sets tool #5 nose radius to .25 in the Offset Library
G10 P5 Q3         ; Sets tool #5 nose vector to 3 in the Offset Library
G10 P1073 R.05    ; Sets machine parameter 73 to 0.05
```

G20 - Select Inch Units

G20 selects inch units, affecting the interpretation of all subsequent dimensions and feedrates in the job file. G20 does not change the native machine units as set on the Control Configuration Menu.

G21 - Select Metric Units

G21 selects metric units, affecting the interpretation of all subsequent dimensions and feedrates in the job file. G21 does not change the native machine units, as set on the Control Configuration Menu.

G22/G23 – Work Envelope On/Off

G22 turns on programmable work envelope in machine coordinates. When the machine tries to move into the forbidden area an "axis work envelope exceeded" message is displayed letting you know which line of the program is at fault. The work envelope is set with the Z and X for the '+' limit and I and J for the '-' limit. G22 is modal and remains on until turned off by G23 or the end of the job. The limits entered in the Z, X and I, J parameters are stored in the WCS menu under <F3> Work Envel.(see chapter 5).

G28 - Return to Reference Point

G28 moves to the first reference point, by way of an intermediate point. The location of the reference point, in machine coordinates, may be set in the Work Coordinate System Configuration menu. The intermediate point is specified in the local coordinate system, and may be at the current location (resulting in a move directly to the reference point). If an intermediate point is specified, only those axes for which positions are specified will be moved. If no axes are specified, only the X and Z axes will be moved. The location of the intermediate point is stored for later use with G29.

Examples:

```
G28 W0      ; move Z axis directly to reference point
            ; (X doesn't move)

G28 U.5 W0  ; move X +.5, then move BOTH axes to
            ; reference point

G28 X2 Z.1  ; move both axes to (2,0.1), then to
            ; reference point

G28         ; move all axes to the reference point
            ; (no intermediate point)
```

The G28 position is of great importance because it specifies the Tool Check position and the usual Tool Change position. The G28 position is the machine coordinate position that the machine will move to when the <TOOL CHECK> button is pressed. Also, the G28 position is the usual position at which tool changes occur during a job run.

G29 - Return from Reference Point

G29 moves the X and Z axes to the intermediate point stored in a preceding G28 or G30 command. It may be used to return to the workpiece. If a position is specified, the machine will move to that position (in local coordinates) after reaching the intermediate point. G29 may only be specified after G28 or G30, though there may be intervening moves.

Examples:

```
G29         ; move X and Z axes back from reference point to
            ;intermediate point
G29 X1 Z2   ; move X and Z axes to intermediate point, then move to X1 Z2
```

G30 - Return to Secondary Reference Point

G30 moves to a specified return reference point, by way of an intermediate point. The P parameter may be used to specify one of the 4 available Return Reference Points: The intermediate point is specified in the local coordinate system, and may be at the current location (resulting in a move directly to the reference point). If an intermediate point is specified, only those axes for which positions are specified will be moved. If no axes are specified, all axes will be moved. The location of the intermediate point is stored for later use with G29.

The 4 available return reference points are defined in the Work Coordinate System Configuration menu. If you issue G30 without a P parameter, it functions exactly like G28, except that by default it uses the second reference return point.

The following table shows how to issue G-codes to utilize the 4 available Return Reference Points:

Return Reference Point	G-Code	Equivalent Alternate G-Code
Return #1	G28	G30 P1
Return #2	G30	G30 P2
Return #3	G30 P3	---
Return #4	G30 P4	---

Examples:

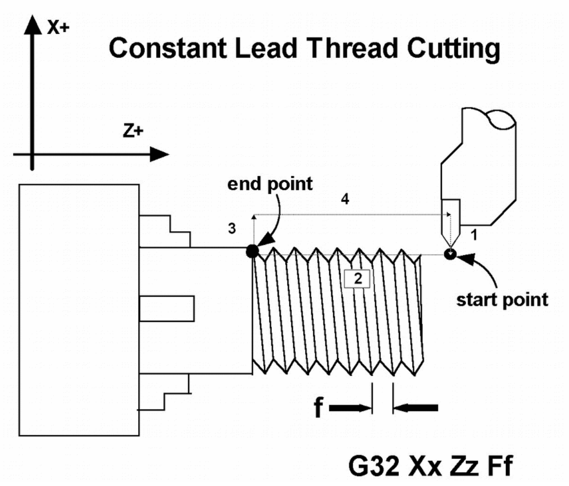
```
G30 Z0      ; move Z axis directly to second reference point
G30 P1      ; move all axes to first reference point
```

● NOTE: G30 P1 is equivalent to G28.

G32 - Constant Lead Thread Cutting

G32 sets the constant lead thread cutting mode. During this mode, both axes are locked to the spindle encoder count. Once the encoder outputs a 1 turn signal, thread cutting is started at a fixed point so that the tool path remains unchanged for repeated thread cutting. Thread cutting follows the same tool path in rough cutting through finish cutting.

● NOTE: When G32 is used, X and Z indicate the endpoint of the cut and F indicates the lead. Also, E can be used instead of F.



Example:

```
G00 X1.5 Z0.0     ; Step 1 - rapid move
G32 X1.5 Z-2.0 F0.125  ; Step 2 - straight
                  ; thread cut of 2 inches, lead of .125
                  ; or 8 threads per inch
G00 X1.7          ; Step 3 - Clear X-axis
G00 z0.0          ; Step 4 - Retract Z-axis
```

G40, G41, G42 –Cutter Diameter Compensation

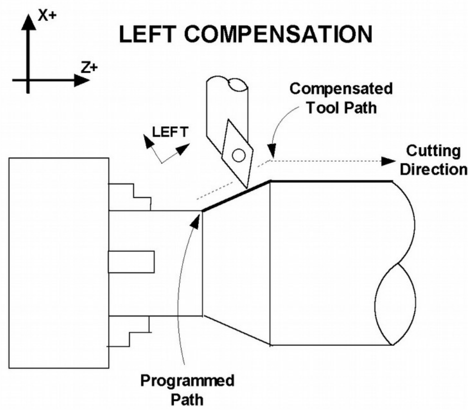
G41 and G42 in conjunction with the selected tool (T code) apply cutter compensation to the programmed tool path. Cutter compensation is required whenever an angle or radius is being cut. G41 offsets the tool selected with the T code the amount of its nose radius, to the left of the workpiece, relative to the direction of travel. G42 offsets the tool selected with the T code the amount of its nose radius, to the right of the workpiece, relative to the direction of travel. G40 cancels G41 and G42. ***Always program cutter compensation as though the machine were a horizontal lathe with the tool post mounted in the rear and the head stock to your left.***

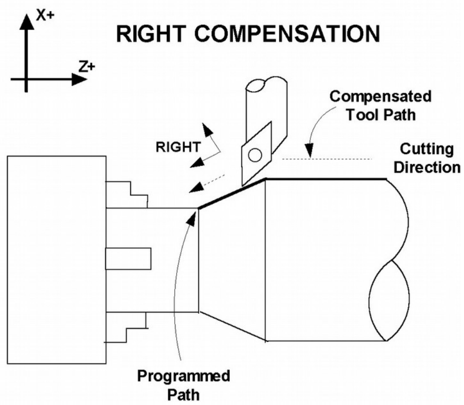
Rules of Thumb:

1. *All OD moves which move towards a more negative Z should use cutter comp right.*
2. *All OD moves which move towards a more positive Z should use cutter comp left.*
3. *All ID moves which move towards a more negative Z should use cutter comp left.*
4. *All ID moves which move towards a more positive Z should use cutter comp right.*

Example:

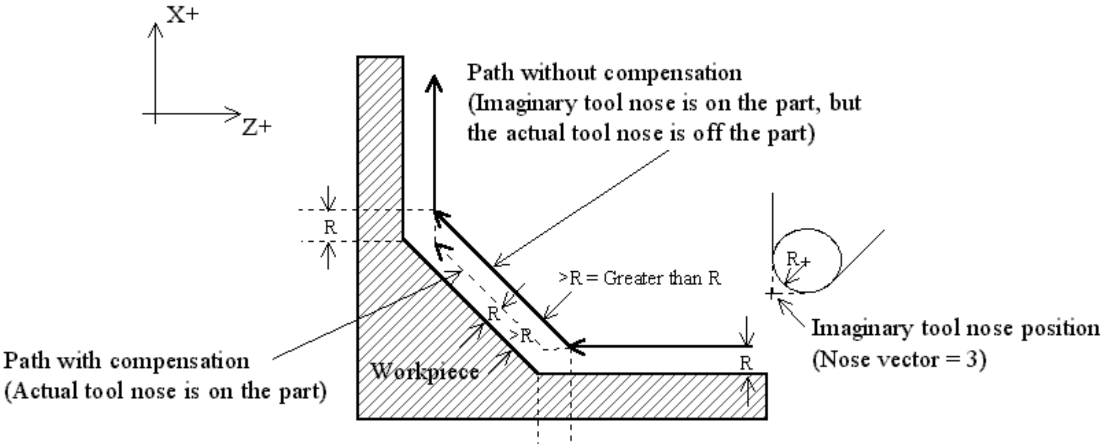
```
G41 T03     ; Tells the machine to compensate left the amount of the
            ; nose radius that corresponds to T03 in the Offset
            ; Library.
```



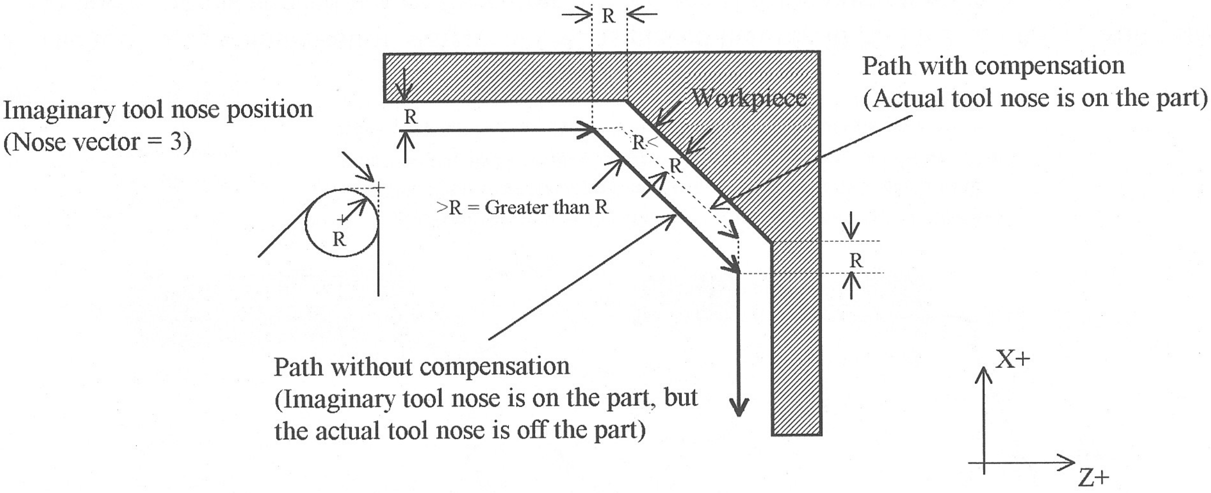


Imaginary Tool Nose

Tool nose compensation is necessary to prevent under-cutting (not cutting enough material) on diagonal lines and arcs. Tool nose compensation does not affect horizontal and vertical lines because in those cases the actual tool nose is at the same depth as the imaginary tool nose. When tool nose compensation is not used, it is the imaginary tool nose that moves to the programmed position and not the cutter. Cutter compensation adjusts for the difference in position by moving the actual tool nose to the programmed position.

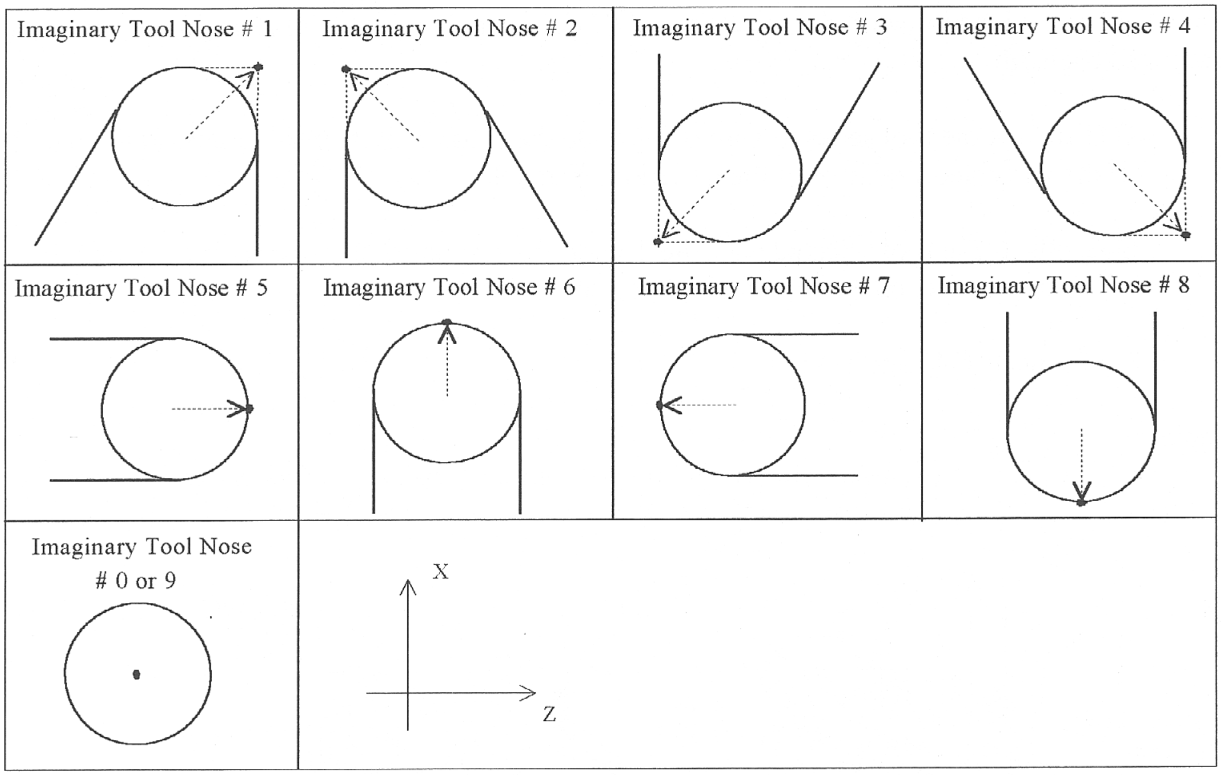

X+
Z+
Path without compensation
(Imaginary tool nose is on the part, but
the actual tool nose is off the part)
R
>R = Greater than R
R
>R
R
R+
Imaginary tool nose position
(Nose vector = 3)
Path with compensation
(Actual tool nose is on the part)
Workpiece

Example with tool located on back side of material.


R
Path with compensation
(Actual tool nose is on the part)
Imaginary tool nose position
(Nose vector = 3)
R
Workpiece
R<
R
>R = Greater than R
R
R
Path without compensation
(Imaginary tool nose is on the part, but
the actual tool nose is off the part)
X+
Z+

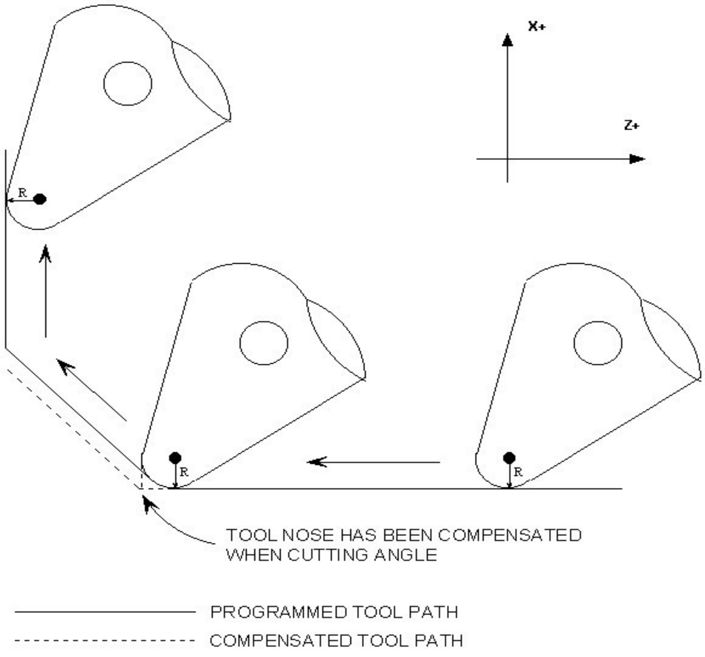
Example with tool located on front side of material.

The direction of the imaginary tool nose is related to the nose vector or direction of the tool during cutting (see Chapter 4). The following drawings show the possible imaginary tool nose directions. Imaginary Tool Nose directions (tool located in back of material):



The tool nose compensation function (G41 or G42) should be in effect before the tool reaches the cutting start point..

IMAGINARY TOOL NOSE #3



G50 -Coordinate System Setting OR Maximum Spindle Speed for CSS mode

G50 has two functions depending on the supplied parameters:

● With axis parameters, G50 sets the current absolute position to the coordinates specified OR
● With the S parameter, G50 sets the maximum spindle speed when using constant surface speed (see G96 and G97).

Examples:

```
G00 X5 Z-2  ; moves to the specified location
G50 X1 Z0   ; sets the current position to the absolute position
            ; specified.
G50 S2500   ; limit spindle to 2500 rpm in G96 mode, no matter how
            ; close X gets to 0.
```

G52 - Offset Local Coordinate System

G52 shifts the local coordinate system origin by a specified distance. Multiple G52 codes are not cumulative; subsequent shifts replace earlier ones. The G52 shift may therefore be canceled by specifying a shift of zero. If you are using multiple coordinate systems, the G52 shift amount will affect all coordinate systems.

Example:

```
G0 X0 Z0    ; move to origin
M98 P9100   ; call subprogram
G52 Z4      ; shift coordinate system 4 inches in Z
G0 X0 Z0    ; move to new origin
M98 P9100   ; call subprogram again with new coordinates
G52 Z0      ; restore unshifted coordinate system
```

G53 - Rapid Positioning in Machine Coordinates

G53 is a one-shot code that performs a rapid traverse using machine coordinates. It does not affect the current movement mode (G00-G03) or coordinate system (G54-G59).

Example:

```
G53 X15 Z0  ; move to 15,0 in machine coordinates
```

G54 - G59 - Select Work Coordinate System

G54 through G59 select among the six regular work coordinate systems (WCS #1 through WCS #6). After issuing the code, subsequent absolute positions will be interpreted in the new coordinate system.

Example:

```
G54 G00 X0 Z0     ; select first WCS, move to origin
G02 X1 Z-.5 R.5   ; cut something...
G55 G00 X1 Z1     ; select second WCS, move to X1,Z1
```

Using Extended Work Coordinate Systems: There are 12 additional work coordinate systems available as an extra-cost option. In a G-code program, these 12 additional work piece origins may be selected with “G54 P1” (WCS # 7) through “G54 P12” (WCS #18).

Regular WCS

WCS	G-Code
WCS #1	G54
WCS #2	G55
WCS #3	G56
WCS #4	G57
WCS #5	G58
WCS #6	G59

Extended Work Coordinate Systems (optional)

WCS	G-Code
WCS #7	G54 P1
WCS #8	G54 P2
WCS #9	G54 P3
WCS #10	G54 P4
WCS #11	G54 P5
WCS #12	G54 P6

WCS	G-Code
WCS #13	G54 P7
WCS #14	G54 P8
WCS #15	G54 P9
WCS #16	G54 P10
WCS #17	G54 P11
WCS #18	G54 P12

G65 - Call Macro

G65 calls a macro with user-specified values. A macro is a subprogram that executes a certain operation (e.g. linear cut, threading, etc.) with values assigned to variable parameters within the operation.

Calling methods:

```
G65 Pxxxx Lrrrr Arguments
```

or

```
G65 "program.cnc" Lrrrr Arguments
```

where *xxxx* is the macro number (referring to file O*xxxx*.cnc, 0000-9999 allowed, leading zeros required in filename, capital O, lower-case .cnc), *rrrr* is the repeat value, "program.cnc" is the name of the macro file, and *Arguments* is a list of variable identifiers and values. Arguments to macro calls are specified by using letters A-Z, excluding G, L, N, O, and P.

Macros are written just like normal programs. However, macro programs may access their arguments by using #A, #B, etc., or by using numbers: #1 for A, #2 for B, etc. (exceptions: #4-6 for I-K, #7-11 for D-H). Arguments I, J, and K can be used more than once in a macro call, with the first set of values stored as #4-6, the second as #7-9, etc., to a maximum of 10 sets.

Macros 9100 - 9999 may be embedded into a main program, using O91*xx* to designate the beginning of the macro and M99 to end it. The CNC software will read the macro and generate a file O91*xx*.cnc, but will not execute the macro. It will be executed when G65 is issued.

Example 1:

Main program:

```
G65 "TEST.cnc" A5 B3
```

Macro TEST.cnc:

```
G01 X#B Z-#A
```

This call will produce:

```
G01 X3 Z-5
```

Example 2:

Main program:

```
G65 "TEST2.cnc" I3 J-5 K0.1 I2 J-2 I0 J0
```

Macro TEST2.cnc:

```
G01 X#4 Z#5 F#6
G01 X#7 Z#8
G01 X#10 Z#11
```

This call will produce:

```
G01 X3 Z-5 F0.1
G01 X2 Z-2
G01 X0 Z0
```

G70 – Profile Finishing Cycle

The G70 finishing cycle is used for applying finish cut passes on a defined profile path by removing material purposely left behind by a previously run “roughing” cycle (G71, G72 or G73). Cutter compensation can be used with this cycle. The feedrate, spindle speed and tool can be different than those used on the “roughing” cycle.

Command Format:

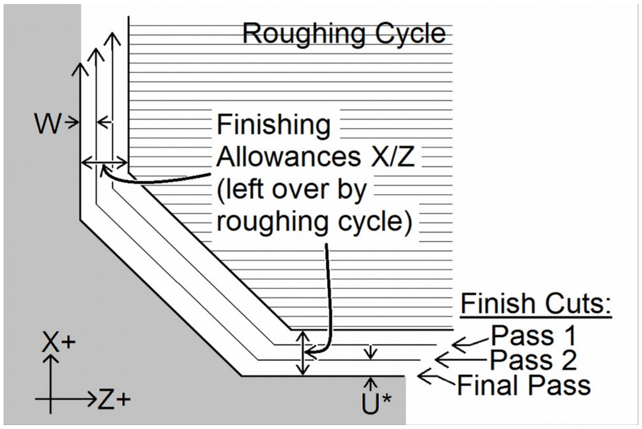
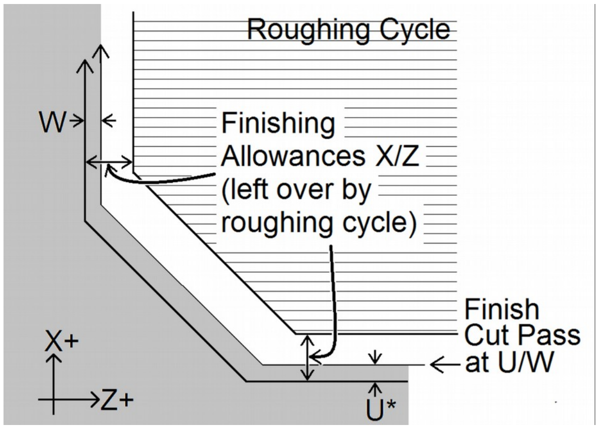
```
G70 P_ Q_ U_ W_
```

P = Starting Block number of finish profile path
Q = Ending Block number of finish profile path
U = Finishing Offset in X direction *
W = Finishing Offset in Z direction

Description of P and Q:
P and Q are required parameters.
P and Q (Starting/Ending Block numbers) do not need to match those used in the “roughing” cycle.
The P and Q parameters must refer to G-code lines occurring before the G70 finishing cycle and not after.

Description of U and W:
The U and W parameters are optional and default to U0 and W0 if not specified.
U and W axis direction signs are computed automatically by the control software.

The G70 cycle has 2 different behaviors depending on the setting of Parameter 274:

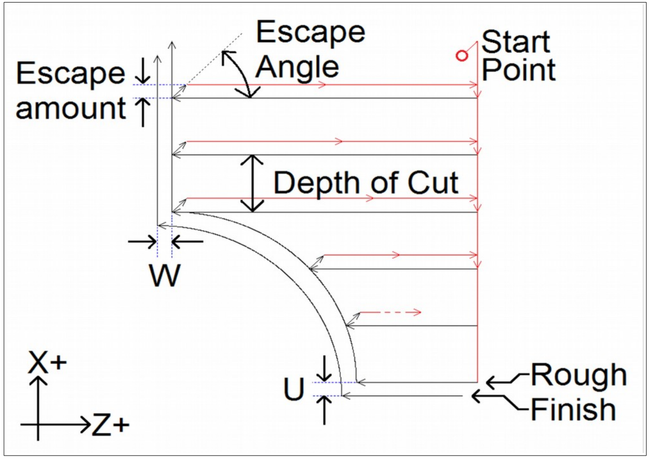
G70 Multi-pass Behavior Enabled (Parameter 247 = 0)	**G70 Single Pass Mode (Parameter 247 = 1)**
 Roughing Cycle W Finishing Allowances X/Z (left over by roughing cycle) Finish Cuts: Pass 1 Pass 2 Final Pass X+ Z+ U*	 Roughing Cycle W Finishing Allowances X/Z (left over by roughing cycle) Finish Cut Pass at U/W X+ Z+ U*
The number of finish cut passes is the larger of the number of times U or W can be divided into their respective X/Z Finish Allowances from a previously run G71/G72/G73 roughing cycle. The resultant finish cut passes generated will be evenly offset from each other. The final pass will be done on the defined profile path without any X/Z offset. However, if U and W are not specified (or is specified as U0 W0), then a single finish cut pass will be done on the defined profile path without any X/Z offset.	A single finish cut pass will be done at the specified cut level. U and W sets the X/Z cut level. A non-zero U or W will leave material behind. This behavior is industry standard.

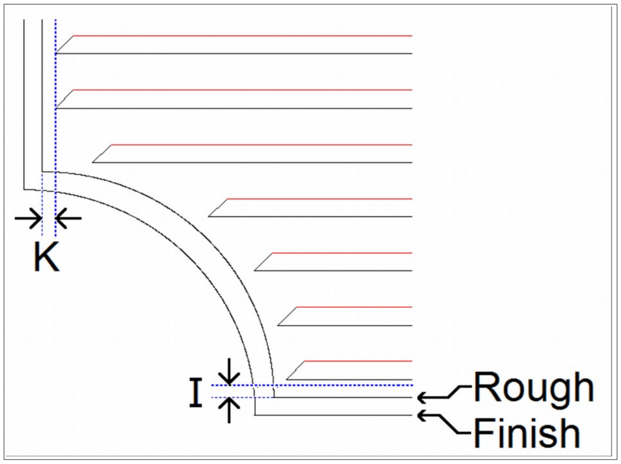
*The U value is subject to Radius/Diameter mode of the machine (as set by Parameter 55 – see Chapter 14).

If codes such as F, S, T and G4 P_ are encountered in the profile, they will be executed by the G70 finishing cycle, unlike a “roughing” cycle such as G71, G72 or G73 which ignore them.

G71 – Profile Stock Removal in Turning

The G71 cycle removes material by using a series of cutting passes parallel to the Z axis and moving away from the Start Point. Each successive cutting pass is done at a progressively deeper turning diameter in the X axis direction incrementing away from the Start Point. The whole cycle is limited to within the confines of the Start Point and a defined profile path (specified via P and Q). The Start Point is wherever the cutter is sitting when the G71 cycle is invoked with P and Q parameters. When all the Z-parallel cuts are done, a single rough cut pass is then run along the profile path. The cycle can perform both inner and outer diameter cleanouts and the profile path can be cutter compensated. The whole cycle can be optionally offset by U and W to leave material behind in preparation for the G70 Finishing Cycle. Also the The Z-parallel cuts themselves can be optionally offset by I and K to further separate them from the single rough cut pass. Note that the axis direction signs for U, W, I and K are computed automatically by the control software.





G71 has two forms:

2 Line Format	**1 Line Format**
Line 1: `G71 U_ R_` Line 2: `G71 P_ Q_ U_ W_ I_ K_ A_`	`G71 P_ Q_ D_ U_ W_ I_ K_ A_`

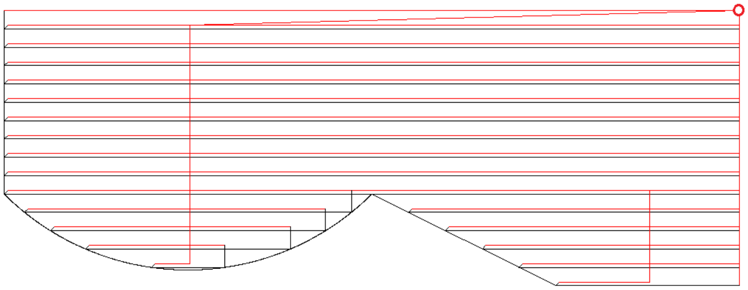
Line 1 U = Depth of Cut *
Line 1 R = Escape amount
Line 2 P = Starting Block number of profile path
Line 2 Q = Ending Block number of profile path
Line 2 U = Finishing Allowance in X direction *
Line 2 W = Finishing Allowance in Z direction
Line 2 I = Roughing Allowance in X direction
Line 2 K = Roughing Allowance in Z direction
Line 2 A = Escape Angle
	P = Starting Block number of profile path
	Q = Ending Block number of profile path
	D = Depth of Cut †
	U = Finishing Allowance in X direction *
	W = Finishing Allowance in Z direction
	I = Roughing Allowance in X direction
	K = Roughing Allowance in Z direction
	A = Escape Angle

* U values are always in radius regardless of Radius/Diameter mode of the machine (as set by Parameter 55 – see Chapter 14).
† D value is subject to the "implied floating point" re-interpretation rules of Parameters 245 and 246.
P and Q are required parameters.
U and W Finishing Allowances are optional and default to U0 and W0 if not specified.
I and K are optional and default to I0 and K0 if not specified.
The modal default for Depth of Cut is from Parameter 43.
The modal default for Escape Amount is from Parameter 44.
The modal default for Escape Angle is from Parameter 58.

The profile path needs to be defined after the G71 P_ Q_. The first and last G-code lines of the profile path must be tagged with the N block numbers so that P and Q can refer to them. If there are any G-code lines occurring after the G71 P_ Q_ and before the first G-code line of the profile path, those lines will not be run by this cycle, but they can be referenced later by a G70 P_ Q_ Finishing Cycle.

Note that any F, S, T and G4 P_ codes encountered in the profile path will be ignored by the G71 cycle.

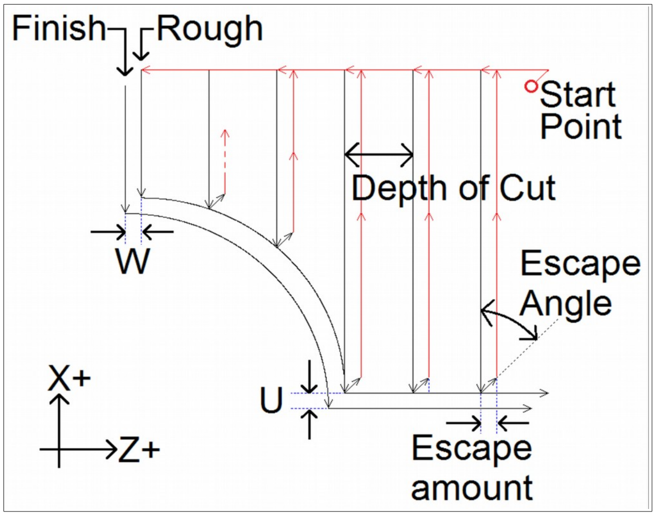
G71 Example:

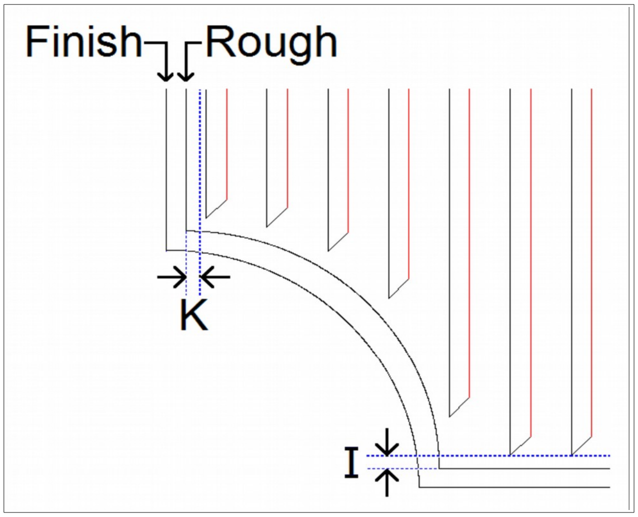


```
G0 X3 Z0                ; Rapid to Start Point
G71 U.05 R.01           ; Set Depth of Cut and Escape Amt
G71 P10 Q20 F.02 S800   ; Cleanout Profile at N10 to N20
N10 G0 X1.5             ;  PROFILE first block
    G1 X1.5 Z-.5        ;  PROFILE
    G1 X2 Z-1           ;  PROFILE
    G2 X2 Z-2 I.5 K-.5  ;  PROFILE
N20 G1 X3               ;  PROFILE last block
```

G72 – Profile Stock Removal in Facing

The G72 cycle removes material by using a series of cutting passes parallel to the X axis and moving away from the Start Point. Each successive cutting pass is done at an incrementally progressing Z face in the direction away from the Start Point. The whole cycle is limited to within the confines of the Start Point and a defined profile path (specified via P and Q). The Start Point is wherever the cutter is sitting when the G72 cycle is invoked with P and Q parameters. When all the X-parallel cuts are done, a single rough cut pass is then run along the profile path. The profile path can be cutter compensated. The whole cycle can be optionally offset by U and W to leave material behind in preparation for the G70 Finishing Cycle. Also the The X-parallel cuts themselves can be optionally offset by I and K to further separate them from the single rough cut pass. Note that the axis direction signs for U, W, I and K are computed automatically by the control software.





G72 has two forms:

2 Line Format	**1 Line Format**
Line 1: `G72 W_ R_` Line 2: `G72 P_ Q_ U_ W_ I_ K_ A_`	`G72 P_ Q_ D_ U_ W_ I_ K_ A_`

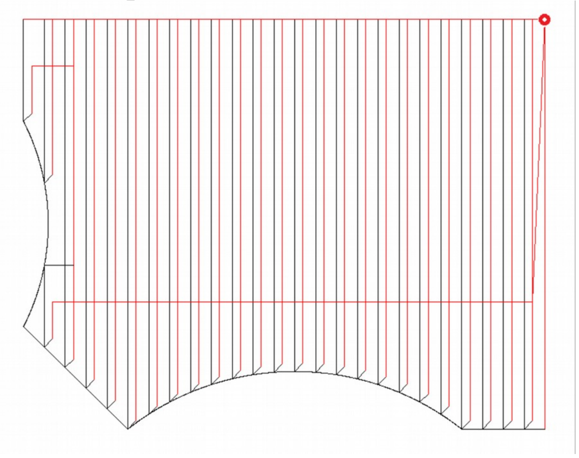
Line 1 W = Depth of Cut
Line 1 R = Escape amount
Line 2 P = Starting Block number of profile path
Line 2 Q = Ending Block number of profile path
Line 2 U = Finishing Allowance in X direction *
Line 2 W = Finishing Allowance in Z direction
Line 2 I = Roughing Allowance in X direction
Line 2 K = Roughing Allowance in Z direction
Line 2 A = Escape Angle
	P = Starting Block number of profile path
	Q = Ending Block number of profile path
	D = Depth of Cut †
	U = Finishing Allowance in X direction *
	W = Finishing Allowance in Z direction
	I = Roughing Allowance in X direction
	K = Roughing Allowance in Z direction
	A = Escape Angle

* U values are subject to Radius/Diameter mode of the machine (as set by Parameter 55 – see Chapter 14).
† D value is subject to the "implied floating point" re-interpretation rules of Parameters 245 and 246.
P and Q are required parameters.
U and W Finishing Allowances are optional and default to U0 and W0 if not specified.
I and K are optional and default to I0 and K0 if not specified.
The modal default for Depth of Cut is from Parameter 43.
The modal default for Escape Amount is from Parameter 44.
The modal default for Escape Angle is from Parameter 58.

The profile path needs to be defined after the G72 P_ Q_. The first and last G-code lines of the profile path must be tagged with the N block numbers so that P and Q can refer to them. If there are any G-code lines occurring after the G72 P_ Q_ and before the first G-code line of the profile path, those lines will not be run by this cycle, but they can be referenced later by a G70 P_ Q_ Finishing Cycle.

Note that any F, S, T and G4 P_ codes encountered in the profile path will be ignored by the G72 cycle, including the single rough cut pass at the end.

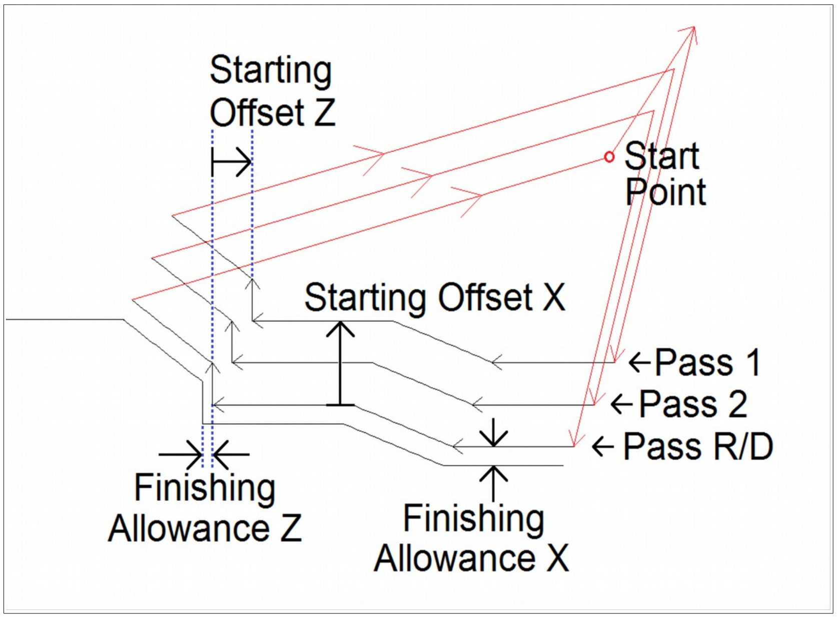
G72 Example:



```
G0   X3 Z0                      ; Rapid to Start Point
G72  W.05 R.02                  ; Setup Depth of Cut and Escape Amount
G72  P11 Q22 F.01 S600          ; Cleanout Profile at N11 to N22
N11  G0 X1                      ;  PROFILE first block
     G1 X1 Z-.2                 ;  PROFILE
     G3 X1 Z-1 I-.5 K-.4        ;  PROFILE
     G1 X1.5 Z-1.25             ;  PROFILE
     G3 X2.5 Z-1.25 I.25 K-.5   ;  PROFILE
N22  G1 X3 Z-1.25               ;  PROFILE last block
```

G73 – Profile Path Repeating Cycle

The G73 cycle removes material by repeating a defined profile path (specified via P and Q) progressively deeper into the material. The profile path can be cutter compensated. The amount of material to remove is determined by the Starting Offset Amounts (in X and Z direcitons). The whole cycle can be optionally offset by Finishing Allowances (in X and Z directions) to leave material behind in preparation for the G70 Finishing Cycle.



G73 has two forms:

2 Line Format	**1 Line Format**
Line 1: `G73 U_ W_ R_` Line 2: `G73 P_ Q_ U_ W_`	`G73 I_ K_ D_ P_ Q_ U_ W_`

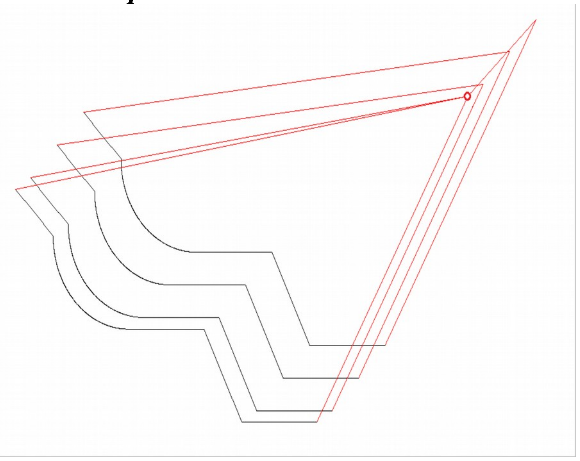
Line 1 U = Starting Offset Amount in X direction *†
Line 1 W = Starting Offset Amount in Z direction †
Line 1 R = Total number of Passes
Line 2 P = Starting Block number of profile path
Line 2 Q = Ending Block number of profile path
Line 2 U = Finishing Allowance in X direction *†
Line 2 W = Finishing Allowance in Z direction †
	I = Starting Offset Amount in X direction †
	K = Starting Offset Amount in Z direction †
	D = Total number of Passes
	P = Starting Block number of profile path
	Q = Ending Block number of profile path
	U = Finishing Allowance in X direction *†
	W = Finishing Allowance in Z direction †

* U values are always radius regardless of Radius/Diameter mode of the machine (as set by Parameter 55 – see Chapter 14).
† Note that axis direction signs for U, W, I and K are computed automatically by the control software.
P and Q are required parameters.
U and W Finishing Allowances are optional and default to U0 and W0 if not specified

The profile path needs to be defined after the G73 P_ Q_. The first and last G-code lines of the profile path must be tagged with the N block numbers so that P and Q can refer to them. If there are any G-code lines occurring after the G73 P_ Q_ and before the first G-code line of the profile path, those lines will not be run by this cycle, but they can be referenced later by a G70 P_ Q_ Finishing Cycle.

Note that any F, S, T and G4 P_ codes encountered in the profile path will be ignored by the G73 cycle.

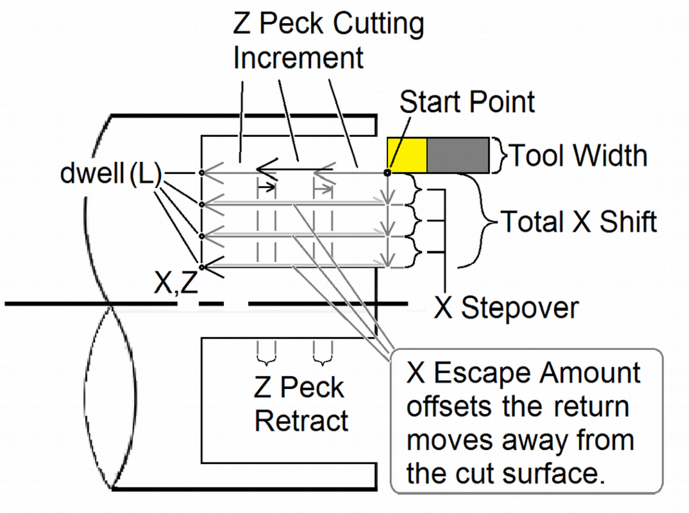
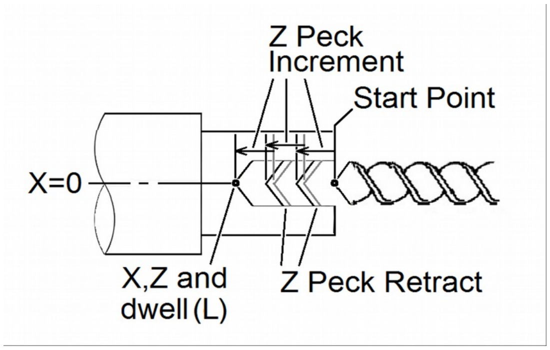
G73 Example:



```
G0 X2.2 Z1.6                       ; Rapid to Start Point
G73 U.14 W.14 R3                   ; REPEAT PROF R=3x, U/W=Starting Offset
G73 P88 Q99 U.05 W.04 F.03 S500 ; REPEAT PROF P-Q,  U/W=Finish Allowance
N88 G0 X.8 W-.40                   ;  PROFILE first block
    G1 W-.2                        ;  PROFILE
       X1.2 W-.1                   ;  PROFILE
       W-.2                        ;  PROFILE
    G2 X1.6 W-.2 R.2               ;  PROFILE
N99 G1 X1.8 W-.1                   ;  PROFILE last block
G70 P88 Q99                        ; FINISH CYCLE P-Q
```

G74 - End Face Peck Cutting Cycle

G74 is used for end face grooving and can also be used for end face peck drilling. The cycle's Z peck cutting action is used for chip breaking. When doing end face grooving, an optional X Shift and X Stepover can be specified to direct the cycle to run multiple shifted groove cut passes in order to produce a groove width that is wider than the tool width. The Start Point of the G74 cycle is wherever the cutter is sitting when the cycle is invoked with a Z_ parameter †.

For End Face Peck Cutting:	**For Peck Drilling:**
 Z Peck Cutting Increment Start Point Tool Width dwell (L) Total X Shift X,Z X Stepover Z Peck Retract X Escape Amount offsets the return moves away from the cut surface.	 Z Peck Increment Start Point X=0 X,Z and dwell (L) Z Peck Retract

G74 has two forms:

2 Line Format	**1 Line Format**
Line 1: `G74 R_` Line 2: `G74 X_ Z_ P_ Q_ R_ L_`	`G74 X_ Z_ I_ K_ D_ L_`

Line 1 R = Z Peck Retract Amount *
Line 2 X = Final X Shift Position †
Line 2 Z = Final Z depth †
Line 2 P = X Stepover ‡
Line 2 Q = Z Peck Cutting Increment ‡
Line 2 R = X Escape Amount for return move
Line 2 L = Dwell time at each Final Z depth
	
	X = Final X Shift Position †
	Z = Final Z depth †
	I = X Stepover
	K = Z Peck Cutting Increment
	D = X Escape Amount for return move ‡
	L = Dwell time at each Final Z depth

* The modal default for Peck Retract Amount is from Parameter 73. Note that in the 1 Line format, you cannot specify a Peck Retract Amount directly. If you are using the 1 Line format exclusively, make sure Parameter 73 is correctly set.
† Incremental X and Z positions are specified using U and W respectively. Note that if U is used it will be subject to Radius/Diameter mode of the machine (as set by Parameter 55 – see Chapter 14).
‡ P, Q and D values are subject to the "implied floating point" re-interpretation rules of Parameters 245 and 246.

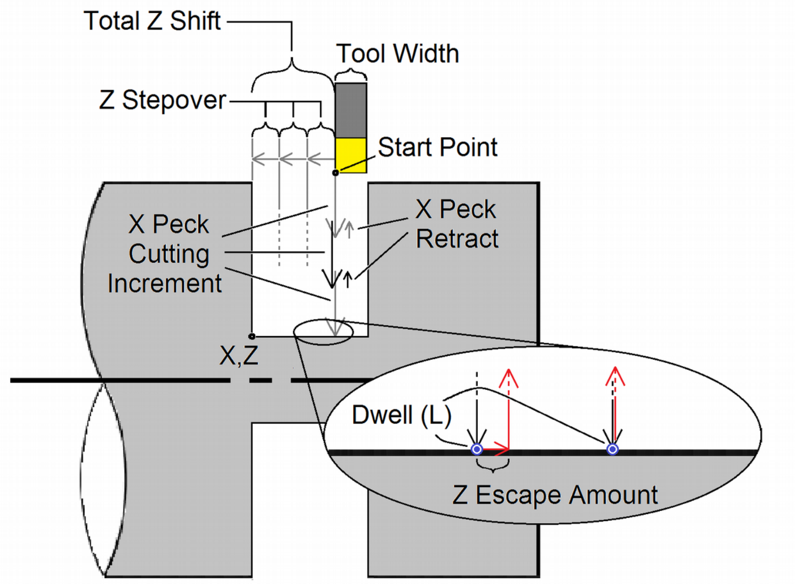
Final Z depth (Z or W) and the Z Peck Cutting Increment (Q or K) are the minimal required parameters; all the others are optional.
Final X Shift Position defaults to the cycle's X Start Point if it is not specified.
X Stepover (P or I) defaults to 0 if not specified.
X Escape Amount (R or D) defaults to 0 if not specified.
Dwell Time (L) should be left out entirely if no dwell is desired because L0 will still cause a (very brief) dwell.

When doing end face grooving, the total width of the resultant groove is the total of the Tool Width added to the Total X Shift. A return move will be done after each groove cut pass (and after the dwell if it is specified). This return move will be spaced away from the cut by the specified X Escape Amount except for the very first cut pass. To prevent the grooving tool from backing up against a wall, the very first groove cut will have no X escape amount.

For centerline peck drilling, the cycle's X Start Point should be 0, and both the Final X Shift Position and the X Stepover should be 0.

G75 - Outside/Inside Diameter Peck Cutting Cycle

G75 is used for outside/inside diameter grooving. The cycle's X peck cutting action is used for chip breaking. An optional Z Shift and Z Stepover can be specified to direct the cycle to run multiple shifted groove cut passes in order to produce a groove width that is wider than the tool width. The Start Point of the G75 cycle is wherever the cutter is sitting when the cycle is invoked with an X_ parameter †.



G75 has two forms:

2 Line Format	**1 Line Format**
Line 1: `G75 R_` Line 2: `G75 X_ Z_ P_ Q_ R_ L_`	`G75 X_ Z_ I_ K_ D_ L_`

Line 1 R = X Peck Retract Amount *
Line 2 X = Final X depth †
Line 2 Z = Final Z Shift Position †
Line 2 P = X Peck Cutting Increment ‡
Line 2 Q = Z Stepover ‡
Line 2 R = Z Escape Amount for return move
Line 2 L = Dwell time at each Final X depth
	X = Final X depth †
	Z = Final Z Shift Position †
	I = X Peck Cutting Increment
	K = Z Stepover
	D = Z Escape Amount for return move ‡
	L = Dwell time at each Final X depth

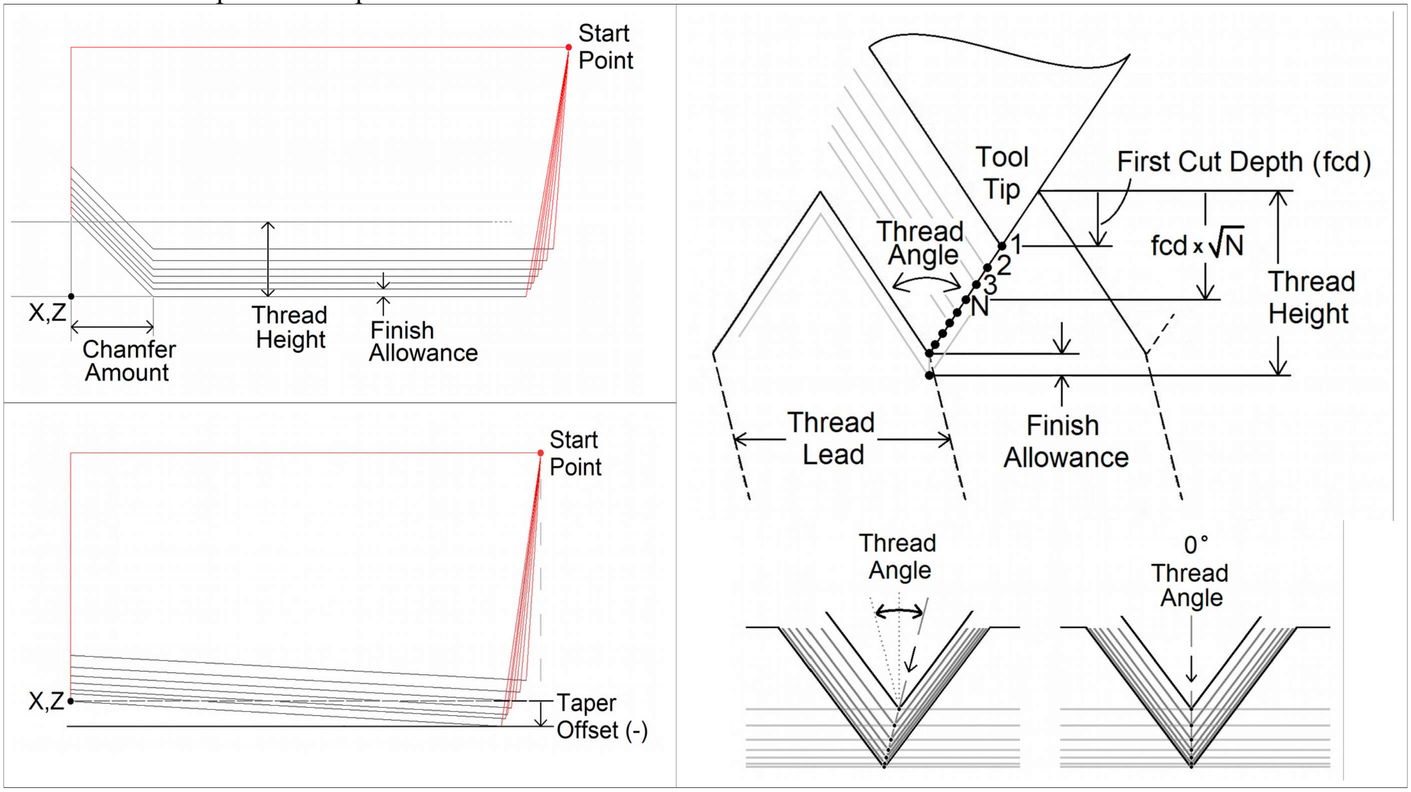
* The modal default for Peck Retract Amount is from Parameter 73. Note that in the 1 Line format, you cannot specify a Peck Retract Amount directly. If you are using the 1 Line format exclusively, make sure Parameter 73 is correctly set.
† Incremental X and Z positions are specified using U and W respectively. Note that X and U are subject to Radius/Diameter mode of the machine (as set by Parameter 55 – see Chapter 14).
‡ P, Q and D values are subject to the "implied floating point" re-interpretation rules of Parameters 245 and 246.

Final X depth (X or U) and the X Peck Cutting Increment (P or I) are the minimal required parameters; all the others are optional.
Final Z Shift Position defaults to the cycle's Z Start Point if it is not specified.
Z Stepover (Q or K) defaults to 0 if not specified.
Z Escape Amount (R or D) defaults to 0 if not specified.
Dwell Time (L) should be left out entirely if no dwell is desired because L0 will still cause a (very brief) dwell.

When this cycle is used for grooving, the total width of the resultant groove is the total of the Tool Width added to the Total Z Shift. A return move will be done after each groove cut pass (and after each optional dwell). This return move will be spaced away from the cut by the specified Z Escape Amount except for the very first cut pass (to prevent the grooving tool from backing up against a wall).

G76 - Multi-Pass Threading Cycle

In this cycle, threading is performed in successive incremental passes to a specified depth. The Start Point of the G76 cycle is wherever the cutter is sitting when the cycle is invoked with an X_ and Z_ parameter *. Optional chamfer and taper can be specified.



The first thread pass is done at the “First Cut Depth” and each successive incremental cut depth will be progressively less than the previous in order to account for greater chip load as the cuts get deeper. However, you can optionally specify a lower limit to this diminishing cut depth increment by specifying a “Minimum Limit Cut Depth”.

The Thread Angle dictates the chip load balance on each side of the thread cutter from one threading pass to the next. A Thread Angle of 0 means an even amount of chip load for each side of the thread cutter.

G76 has two forms.

2 Line Format	**1 Line Format**
Line 1: `G76 Pmmrraa Q_ R_` Line 2: `G76 X_ Z_ R_ P_ Q_ F_`	`G76 X_ Z_ K_ D_ A_ I_ F_`

Line 1 P mm_____ = Finish / Spring Pass Count
Line 1 P __rr__ = Chamfer Amount in units of $^1/_{10}$ Thread Lead
Line 1 P _____aa = Thread Angle
Line 1 Q = Minimum Limit Cut Depth †
Line 1 R = Finish Allowance
 	
Line 2 X = Thread Final Depth Position X *
Line 2 Z = Thread Final Position Z *
Line 2 R = Taper Offset (+/-)
Line 2 P = Thread Height †
Line 2 Q = First Cut Depth †
Line 2 F = Thread Lead ‡
	X = Thread Final Depth Position X *
	Z = Thread Final Position Z *
	K = Thread Height
	D = First Cut Depth †
	A = Thread Angle
	I = Taper Offset (+/-)
	F = Thread Lead ‡

* Incremental X and Z positions are specified using U and W respectively. Note that X and U are subject to Radius/Diameter mode of the machine (as set by Parameter 55 – see Chapter 14).
† P, Q and D length values are subject to "implied floating point" re-interpretation rules of Parameters 245 and 246.
‡ F is not strictly a parameter of this cycle as it is a modal value. And alternatively, the modal E value can also be used to specify Thread Lead.

The minimal required parameters for this cycle are Thread Final positions X depth and Z, First Cut Depth (Q or D), and Thread Height (P or K); all the others are optional and have a modal effect if specified. Whatever optional values are not specified, they are taken from the current modal state. In the case of the 1 Line format, some values simply cannot be specified using that format, so these unspecified values will come from the current modal values whatever they happen to be, which at the beginning of a job are initialized from Parameters 49-53.

The modal default for Finish / Spring Pass Count is from Parameter 50.
The modal default for Chamfer Amount is from Parameter 49 (specified in multiples of Thread Lead).
The modal default for Thread Angle is from Parameter 51.
The modal default for Minimum Limit Cut Depth is from Parameter 52.
The modal default for Finish Allowance is from Parameter 53.

G80 – Canned Cycle Cancel

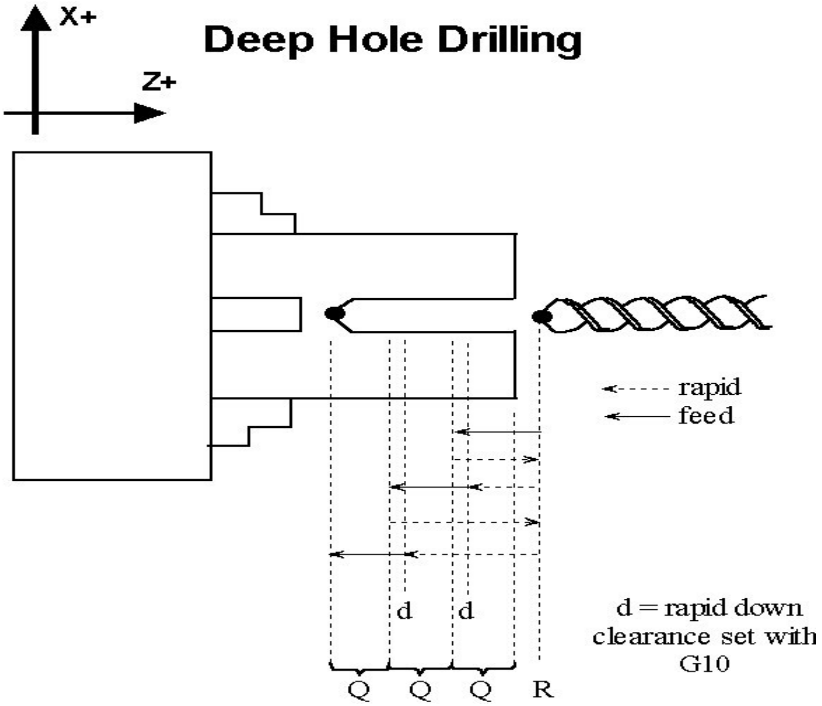
G80 is used to cancel a canned cycle once the operation has been performed.

G83 – Deep Hole Drilling

G83 is a deep hole drilling cycle. It periodically retracts the tool to the surface to clear accumulated chips, then returns to resume drilling where it left off. The retract and return are performed at a rapid rate. Because there may be chips in the bottom of the hole, the tool does not return all the way to the bottom at the rapid rate. It slows down to federate a short distance above the bottom. This clearance distance is selected by setting parameter 83 with G10 (see example below).

Example:

```
G10 P83 R.05                ; set clearance to .05"
G83 X0 R.1 Z-2 Q.5          ; drill 2" deep hole in 0.5" steps
```



G84 – Tap/Counter Tap

G84 performs both right-hand (Tap) and left-hand (Counter Tap) cycles. The tap direction, spindle speed, spindle direction, feedrate and/or thread pitch should be set before the G84 is invoked. The tapping direction is set by calling M29 with or without a P1 parameter (see M29 in Chapter 12). G84 defaults to right-hand Tap cycle if M29 P1 is not specified beforehand. The spindle should be started beforehand using M3 (Spindle CW) for the Tap cycle, or using M4 (Spindle CCW) for the Counter Tap cycle.

For a floating tap head, the combination of the modal feedrate and spindle speed implicitly determines the approximate thread lead or pitch. However, if Rigid Tapping is enabled, a Q may be used to explicitly set the thread lead or pitch. However, because Q is not modal in the case of Rigid Tapping, you must specify Q on every line at which Rigid Tapping is to occur.

The Tap/Counter Tap cycle might to cut a short distance beyond the programmed Z height as the spindle comes to a stop before reversing. When tapping blind holes, be sure to specify a Z height slightly above the bottom of the hole to prevent the tool from reaching bottom before the spindle stops. The exact distance you must allow will depend on your machine and the diameter and pitch of the tapping tool.

WARNING FEED HOLD is temporarily disabled during the tapping cycle, but it will be re-enabled at the end of the cycle.

NOTICE Pressing CYCLE CANCEL while the tap is in the hole will very probably break the tap or strip the threads in the tap hole. However, do so if it is an emergency.

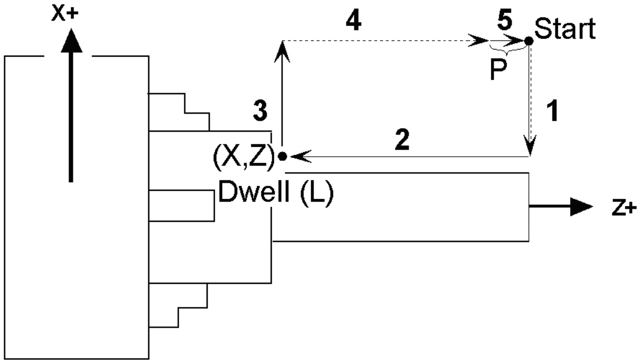
G85 – Boring Cycle

G85 is used to bore a hole so that a smooth finish may be acquired. The tool will feed into depth at the specified federate and retract back out at the same federate.

G90 - Outside/Inside Diameter Cutting Cycle

G90 sets the outer/inner diameter cutting cycle command. These diameters can be specified along straight cuts or diagonal/taper cuts.

Straight Cutting



The general form of the Straight Cutting Cycle is as follows:

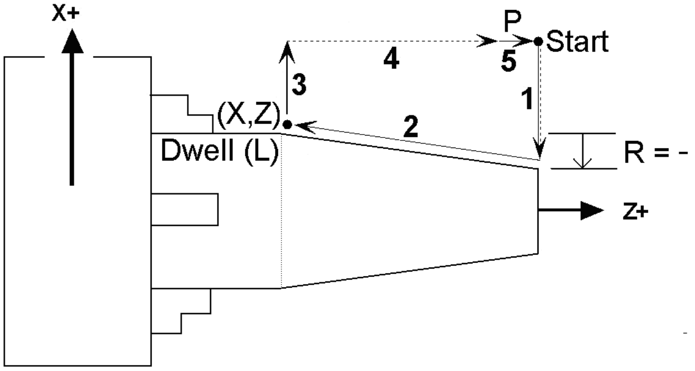
```
    G90 X_ Z_ P_ L_
```

In incremental programming form, the X and Z can be substituted with U and W. Note that X (or U) is affected by the radius/diameter programming mode (see parameter 55 in Chapter 14). The optional parameter P specifies the length of the return feed move (segment 5 in illustration above). This cycle behaves differently depending on whether a non-zero P is specified or not. If P does not exist or is 0, then segments 1 and 4 will be rapid moves, segments 2 and 3 will be feedrate moves, and segment 5 will be omitted. If P does exist and is non-zero, then segment 4 will be a rapid move and all the other segments will be feedrate moves. The optional parameter L specifies a dwell time between segments 2 and 3.

Example:

```
G00 X2.5 Z-1.0             ; rapid to start point
G90 X1.5 Z-4.0 F0.5 L1.5   ; G90 cycle with 1.5 sec dwell at X1.5 Z-4.0
```

Taper Cutting



The general form of the Taper Cutting Cycle is as follows:

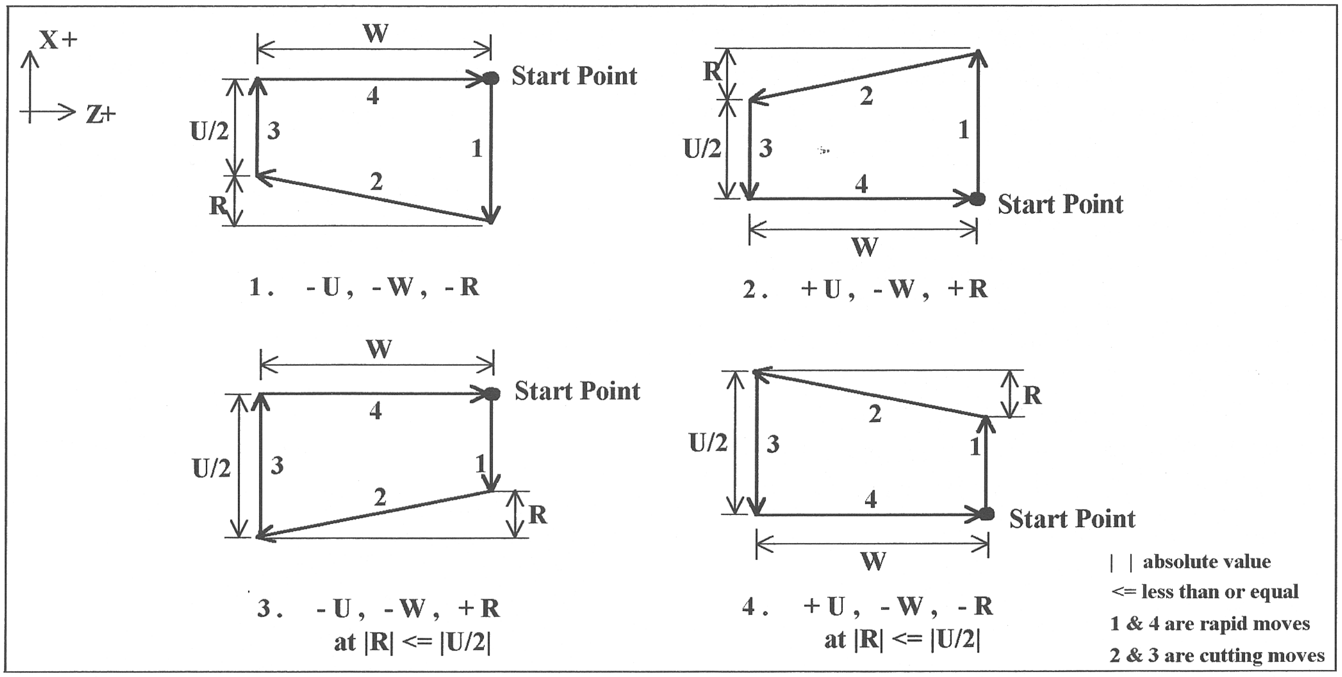
```
G90 X_ Z_ R_ P_ L_
```

This is actually the same as the Straight Cutting cycle (mentioned above) but with the addition of the R parameter. Parameters P and L are optional. Taper is determined by offsetting the point between segments 1 and 2 on the X coordinate by the incremental amount specified by the R parameter. Note that R is unaffected by the radius/diameter programming mode, but X is (see parameter 55 in Chapter 14). All the other parameters have the same meaning as those of the Straight Cutting cycle (mentioned above).

Example:

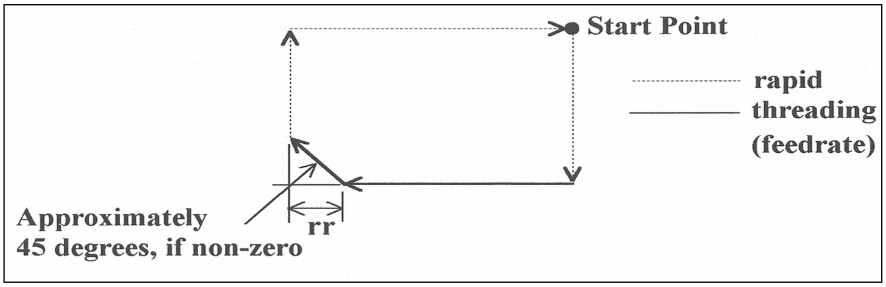
```
G00 X2.5 Z-1.0              ; rapid to start point
G90 X1.5 Z-4.0 R-0.25 F0.5  ; Tapered G90 cycle
```

The following table shows the relationship between the tool paths and the signs of U, W, and R during incremental programming when performing taper cutting.



G92 - Thread Cutting Cycle

G92 sets the thread cutting cycle command. This cycle can be specified for straight thread cutting or taper thread cutting. In incremental programming, the signs of U and W will depend on the direction of the tool path when approaching the workpiece. That is, if the cutter moves in the negative X direction, then the value of U will be negative.



G92 is similar to G32 in that X and Z indicate the endpoint of the cut and F indicates the thread lead and X & Z are slaved to the spindle. The chamfering amount, rr, which is selected by Parameter 49 (see Chapter 14), is a multiplier of the thread lead. That is, the chamfer distance is rr times the thread lead.

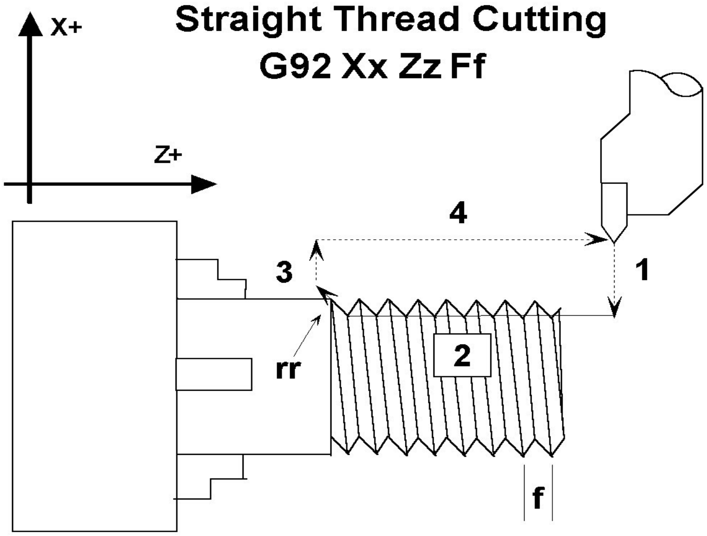
Straight Thread Cutting

In this cycle, the cutter moves to the diameter indicated by X and threads in a straight line to the depth or length indicated by Z.

In the example below, the cutter first rapids to the start point located at X2.5Z-1, then rapids down to X2 at the same Z, and then cuts with the specified lead to Z-3. At Z-3, the cutter pulls out of the part the amount of the chamfering distance, then rapids back up to X2.5 and returns to the start point.

Example:

```
G00 X2.5 Z-1.0
G92 X2.0 Z-3.0 F.1
```

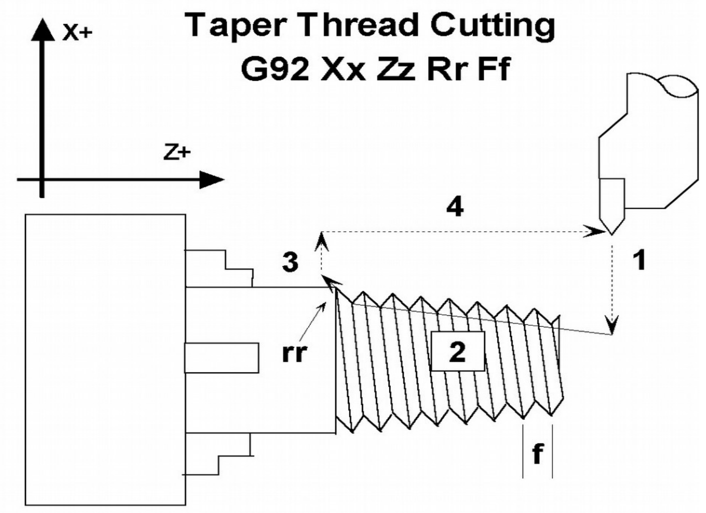


1, 3, and 4 are rapid moves. 2 is cutting move

Taper Thread Cutting

In this cycle, the cutter threads diagonally to the diameter and depth indicated by X and Z, respectively. The value of R will dictate the value of the starting diameters. A negative R will make the ending diameter equal to X and the starting diameter equal to X minus twice the absolute value of R. A positive R will make the ending diameter equal to X and the starting diameter equal to X plus twice the value of R.

In the example below, the cutter first rapids to the start point located at X3.5 Z-1, then rapids down to X2.5, the inner diameter, at the same Z, and then cuts with the specified lead to Z-3. At Z-3 the value of the outer diameter is 2.5 and the cutter pulls out of the part the amount of the chamfering distance, then rapids back up to X2.5 and returns to the start point.



1, 3, and 4 are rapid moves. 2 is cutting move.

Example:

```
G00 X3.5 Z-1.0
G92 X2.5 Z-3.0 R-0.25 F.1
```

Multiple thread leads

This is done by using the formula:

2nd – nth thread lead start point = previous thread lead start point + ((1/TPI) / # of leads)

Example:

We want to produce a triple lead thread with a thread lead of 10 threads per inch (TPI). The start point for the first thread lead is 0.1000 from the face of the material being threaded.

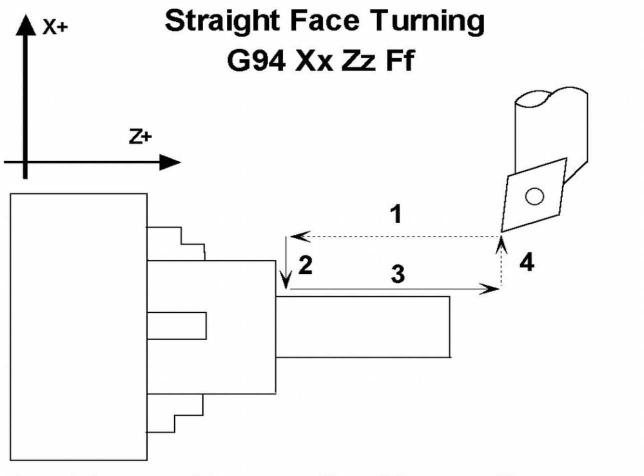
Thread lead # 1 start point = 0.1000.
Thread lead # 2 start point = 0.1000 + ((1/10)/3) = 0.1333.
Thread lead # 3 start point = 0.1333 + ((1/10)/3) = 0.1666.

G94 - End Face Turning

G94 sets the end face turning cycle command. This cycle can be specified for straight face turning or taper face turning. In incremental programming, the signs of U and W will depend on the direction of the tool path when approaching the work piece. That is, if the cutter moves in the negative X direction, then the value of U will be negative. The L parameter can be set to allow the part to rotate at least one full revolution, at the end X position, before the tool is moved back to the starting Z position.

Straight Face Turning

In this cycle, the cutter moves to the depth indicated by Z and then cuts to the diameter indicated by X. In the example below, the cutter first rapids to the start point located at X2Z-1, then rapids to Z-1.25 at the same X, and then cuts at the specified feedrate to X1. At X1, the cutter dwells for .5 secs, then moves back to Z-1 at the same feedrate and rapids back up to the start point.


Straight Face Turning
G94 Xx Zz Ff

1 and 4 are rapid moves. 2 and 3 are cutting moves.

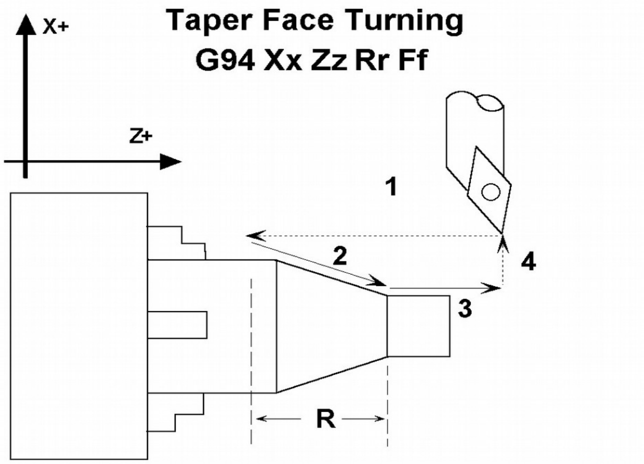
Example:

```
   G00 X2.0 Z-1.0
   G94 X1.0 Z-1.25 F0.1 L.5
```

Taper Face Turning

In this cycle, the cutter cuts diagonally to the diameter and depth indicated by X and Z, respectively. The value of R will dictate the approach of the cutter to the specified Z coordinate, that is, the value of R will determine how much the cutter will stop short (positive R) or pass (negative R) Z before cutting diagonally down to the specified diameter.

In the example below the value of R is negative, thus, the cutter first rapids to the start point located at X2Z-1, then rapids to Z-1.5 at the same X, and then cuts diagonally down to X1 at the specified feedrate. At X1, the value of Z is -1.25, then the cutter moves back to Z-1 at the same feedrate and rapids back up to the start point.

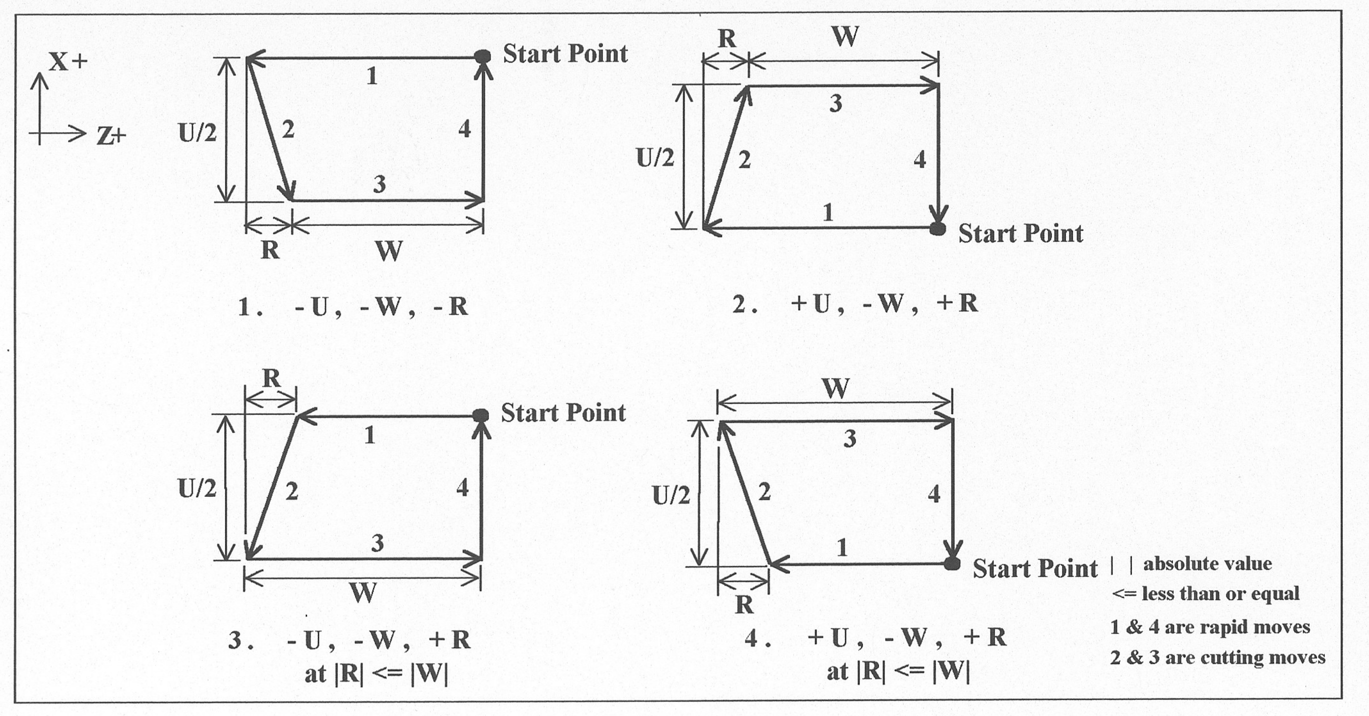

Taper Face Turning
G94 Xx Zz Rr Ff

1 and 4 are rapid moves. 2 and 3 are cutting move

Example:

```
   G00 X2.0 Z-1.0
   G94 X1.0 Z-1.25 R-0.25 F0.1
```

The following table shows the relationship between the tool paths and the signs of U, W, and R during incremental programming when performing taper face turning.



G96 & G97 - Constant Surface Speed Control & Cancel

G96 sets the mode for constant surface speed control in feet/min (sfm) or meters/min. S values are assumed as surface speed. When CSS is active, the spindle speed changes as the X position changes, to maintain a constant linear velocity at the tool tip. No matter how close X gets to X0, the spindle speed will not exceed the speed set with G50 or the machine's maximum spindle speed, whichever is less. G97 cancels the constant surface speed control.

```
G96 S800          ; sets constant surface speed to 800 feet/min
G01 X1 Z-3 F0.1
G97 S1200         ; cancels constant surface speed and sets
                  ; spindle speed to 1200 rpm
```

G98 - Feed per minute

G98 sets the cutting feedrate mode in units/minute. There are no associated parameters.

G99 - Feed per revolution

G99 sets the cutting feedrate mode in units/rev. There are no associated parameters.

Chapter 12
M functions

M-functions are used to perform specialized actions in CNC programs. Most of the T-series Control M-functions have default actions, but they can be customized with the use of macro files.

Certain restrictions apply to calling M functions:
● Only one M function per program line is permitted.
● M-functions are not allowed on the same line as a tool change (see T in Chapter 10).

Summary of M functions

M00 Stop For Operator	M94/M95 Output On/Off
M01 Optional Stop for Operator	M98 Call Subprogram
M02 Restart Program	M99 Return from Macro or Subprogram
M03 Spindle On Clockwise	M100 Wait for PLC bit (Open, Off, Reset)
M04 Spindle On Counterclockwise	M101 Wait for PLC bit (Closed, On, Set)
M05 Spindle Stop	M102 Restart Program
M07 Mist Coolant On	M103 Programmed Action Timer
M08 Flood Coolant On	M104 Cancel Programmed Action Timer
M09 Coolant Off	M105 Move Minus to Switch
M10 Clamp On	M106 Move Plus to Switch
M11 Clamp Off	M107 Output Tool Number
M13 (macro) Cutoff *	M108 Enable Override Controls
M16 (macro) Chuck ID selection *	M109 Disable Override Controls
M18 (macro) Chuck OD selection *	M115,M116,M125,M126 Protected Move Probing Functions
M19 (macro) Spindle Orient *	M120 Open data file (overwrite existing file)
M22 (macro) Extend part chute *	M121 Open data file (append to existing file)
M23 (macro) Retract part chute *	M122 Record position(s) in data file
M26 Set Axis Home	M123 Record value and/or comment in data file
M29 Set Tap Mode for G84	M124 Record machine position(s) in data file
M30 Restart Program	M127 Record Date and Time in a data file
M32 (macro) Tailstock Quill forward (out) *	M128 Move Axis by Encoder Counts
M33 (macro) Tailstock Quill retract (in) *	M129 Record Current Job file path to data file
M34 (macro) Part Catch forward *	M130 Run system command
M35 (macro) Part Catch retract *	M150 Set Spindle Position to 0 on Next Index Pulse
M41,M42,M43 (macro) Select Spindle Gear Range *	M151 Unwind C axis
M46 (macro) Door Open *	M200/M201 Stop for Operator, Prompt for Action
M47 (macro) Door Close *	M223 Write Formatted String to File
M50 C Axis Disable	M224 Prompt for Operator Input Using Formatted String
M51 C Axis Enable	M225 Display Formatted String for A Period of Time
M91 Move to Minus Home	M290 Digitize Profile
M92 Move to Plus Home	M300 Fast Synchronous I/O update
M93 Release/Restore Motor Power	M1000-M1015 Graphing Color for Feedrate movement

* M functions marked with “(macro)” actually have no standard default action, and could possibly be unimplemented and therefore unavailable on your machine. Also, their stated function is only standard on certain machines.

Macro M functions (custom M functions)

Macro M functions are M functions that have been customized with a macro file. The T-Series CNC M functions from 0 through 90 can be fully customized. No M-functions above 90 may be customized with macros. The default action listed will be performed unless that M-function has been customized.

To create a macro for an M-function, a file must be created in the C:\cnct directory. The file's name must be mfuncXX.mac where XX is the M-function number used to call the macro. M-functions 0-9 must use single digits in the filename (e.g. use mfunc3.mac, **not** mfunc03.mac). The contents of the file may be any valid M and G-codes.

The following is an example macro M-Function to turn on spindle with variable frequency drive and wait for "at speed" response.

```
  M94/1  ; request spindle start
  M101/50005  ; wait for up to speed signal
```

These lines would be placed in the file c:\cnct\mfunc3.mac. Each time the M-function is encountered in a program, the macro file will be processed line by line.

● NOTE: Nesting of macro M-functions is allowed, but, recursive calls are not. If a macro M-function does call itself, the default action of the function will be executed.

● NOTE: Unlike subprograms invoked with G65, macro arguments passed into a macro M function are passed in by reference. This means local changes to #1 through #33 or #A through #Z will change the their values upon return from the macro M function.

● NOTE: The M and G-codes within a macro M-function are not usually displayed on the screen as they are executed, and are all treated as one operation in block mode. If you wish to see or step through macro M-functions (e.g. for testing purposes), see Machine Parameter 10 in Chapter 14

● NOTE: The cnctch.mac file, which contains the G-code sequence for doing a customized tool change, is also considered to be an M-function Macro so that its behavior can be modified by Machine Parameter 10.

M00 - Stop For Operator

Motion stops and the operator is prompted to press the CYCLE START button to continue.

M01 - Optional Stop for Operator

M1 has no effect unless optional stops are turned on. When optional stops are on, M1 is identical to M0.

Default action:

```
  M100/75     ; if optional stops are turned on.
```

M02 - Restart Program

Restarts the program from the first line. The operator is prompted to press the CYCLE START button to continue.

M03 - Spindle On Clockwise

M3 requests the PLC to start the spindle in the clockwise direction.

Default action:

```
  M95/2
  M94/1
```

M04 - Spindle On Counterclockwise

M4 requests the PLC to start the spindle in the counterclockwise direction.

Default action:

```
  M95/1
  M94/2
```

M05 - Spindle Stop

M5 requests the PLC to stop the spindle.

Default action if the spindle had been spinning CW:
M95/2
M95/1

Default action if the spindle was OFF or was spinning CCW:
M95/1
M95/2

M07 - Mist Coolant On

M7 causes the PLC to start the mist coolant system.

Default action:

```
  M95/3
  M94/5
```

M08 - Flood Coolant On

M8 causes the PLC to start the flood coolant system.

Default action:

```
  M95/5
  M94/3
```

M09 - Coolant Off

M9 causes the PLC to stop the coolant system.

Default action:

```
M95/3/5
```

M10 - Clamp On

M10 causes the PLC to activate the clamp.

Default action:

```
M94/4
```

M11 - Clamp Off

M11 causes the PLC to release the clamp.

Default action:

```
M95/4
```

M19 – Spindle Orient (Macro)

M19 has no default action, therefore a custom M19 macro must be defined for this feature to work. If defined, the M19 macro sends a request to the PLC to rotate the spindle to its pre-set orient position.

M26 - Set Axis Home

M26 sets the machine home position for the specified axis to the current position (after the line's movement).

Example:

```
M92/X  ; home X axis to plus home switch
M26/X  ; set machine home for X-axis there
M91/Z  ; home Z-axis to minus home switch
M26/Z  ; set machine home for Z-axis there
```

M29- Set Tap Mode for G84

M29 sets the tap mode for G84; either right-hand or left-hand tapping. Right-hand tap mode is the initial default at job start-up. If Left-hand tap mode is required, M29 and P1 need to be specified on the same line.

Tap Mode	Command
CW (Right-hand)	M29
CCW (Left-hand)	M29 P1

M30 - Restart Program

Restarts the program from the first line. The operator is prompted to press the CYCLE START button to continue. This has the same functionality as M2.

M41, M42, M43 – Select Spindle Gear Range (Macros)

M41, M42, and M43 have no default actions, and therefore custom macros must be defined for these M codes in order to make this feature work. If defined, these macros notify the PLC of which spindle gear range is selected according to the following table:

Macro M function	Action
M41	Select Low Gear Range
M42	Select Medium-Low Gear Range
M43	Select High Gear Range

M50 – C Axis Disable

M50 is the command to disable the C axis and it is a locked software option. When the C axis is disabled, no axis label will be present on the screen and the encoder information for the C axis is ignored. In order for the M50 command to work, the 3rd or 4th axis label must be set to 'C' with the associated parameter (93 for 3rd axis and 94 for 4th axis) set for C axis operation. In practical applications, the default behavior for the M50 command is usually modified using a custom mfunc50.mac program.

Example mfunc50.mac:

```
  M95/9     ; Switch to speed mode
  M50       ; Perform the default actions for C axis disable
```

M51 – C Axis Enable (Optional)

M51 is the command to enable the C axis and it is also locked as a software option.. When C axis is enabled, the C axis label will be present on the DRO and encoder information for the C axis is used to determine the position of the C axis. In order for the M51 command to work, the 3rd or 4th axis label must be set to 'C' with the associated parameter (93 for 3rd axis and 94 for 4th axis) set for C axis operation. In practical applications, the default behavior for the M51 command is modified using a custom mfunc51.mac program to ensure that the spindle has stopped before the C axis is enabled.

Example mfunc51.mac:

```
  G97         ; Turn off CSS (constant surface speed)
  M3 S0       ; Turn off spindle
  M101/9      ; Wait for zero speed signal form inverter on INP9
  M94/9       ; Switch to torque mode
  M51         ; Perform the default actions for C axis enable
  M151        ; Unwind C-axis position
```

Note in the above examples for M50 and M51 where the M95/9 (turn off INP41) and M94/9 (turn on INP41) commands are used, it is assumed that the plc program, conditioned upon the state of INP41, has been modified to output the appropriate hardware signals required to switch between speed and torque mode.

M91 - Move to Minus Home

M91 moves to the minus home switch of the axis specified at the slow jog rate for that axis. After the minus home switch is reached, the tool is moved back until the home switch resets. Then the next encoder index pulse is reached.

Example:

```
  M91/Z       ; move the Z-axis to the minus home switch.
  G50 Z-10    ; sets Z minus home switch at -10
```

M92 - Move to Plus Home

M92 moves to the plus home switch of the axis specified at the slow jog rate for that axis. After the plus home switch is reached, the tool is moved back until the home switch resets. Then the next encoder index pulse is reached.

Example:

```
M92/X       ; moves the X-axis to the plus home switch.
G50 X+10    ; Sets X plus home switch at +10
```

M93 – Release/Restore Motor Power

M93 releases or restores motor power for the axis specified. If no axis is specified, then all axes are released.

Example:

To release motor power:

```
M93/X       ; releases the X axis.
M93         ; releases the motors on all axes.
```

Example:

To restore power:

```
M93/X P1    ; restore power to the X axis motor.
M93 P1      ; restore power to the motors on all axes.
```

● NOTE: Any axis freed within a CNC program should **not** be used in that program afterwards. Incorrect positioning may result.

M94/M95 - Output On/Off

There are 128 user definable system variable bits that can be used to communicate with the PLC. M94 and M95 are used to request those system variable bits to turn on or off respectively. Requests 1-128 are mapped to the PLC as system variables SV_M94_M95_1 through SV_M94_M95_128 as shown in the following table:

On	Off	PLC bit
M94/1	M95/1	SV_M94_M95_1
M94/2	M95/2	SV_M94_M95_2
M94/3	M95/3	SV_M94_M95_3
M94/4	M95/4	SV_M94_M95_4
. . .	. . .	. . .
M94/128	M95/128	SV_M94_M95_128

To use M94 and M95 to control a function external to the servo control, such as an indexer, the input request must be mapped to one of the PLC outputs in the PLC program. See M94/M95 function usage in the PLC section of the service manual.

Example:

```
M94/5/6  ; turns on SV_M94_M95_5 and SV_M94_M95_6.
```

* NOTE: M94 and M95 will cause prior motion to decelerate to a stop before the requested bits are turned on or off.

* NOTE: Requests 1-5, 15, and 16 are controlled by the default actions of M3, M4, M5, M6, M7, M8, M9, M10, M11, and M39. To override or disable a bit used in one of these M codes, define a custom M-function.

M98 - Call Subprogram

M98 calls a user-specified subprogram. A subprogram is a separate program that can be used to perform a certain operation (e.g. a drilling pattern, contour, etc.) many times throughout a main program.

Calling methods:

```
M98 Pxxxx Lrrrr
```

or

```
M98 "program.cnc" Lrrrr
```

where *xxxx* is the subprogram number (referring to file O*xxxx*.cnc, 0000-9999 allowed, leading 0's required in filename, capital O, lowercase .cnc), *rrrr* is the repeat value, and "program.cnc" is the name of the subprogram file.

Subprograms are written just like normal programs, with one exception: an M99 should be at the end of the subprogram. M99 transfers control back to the calling program.

Subprograms can call other subprograms (up to 20 nested levels of calling may be used), Macro M-functions, and Macros. Macro M-functions and Macros can similarly call subprograms.

Subprograms 9100-9999 can also be embedded into a main program, using O9xxx to designate the beginning of the subprogram and M99 to end it. CNC11 will read the subprogram and generate a file O9*xxx*.cnc. CNC11 will not execute the subprogram until encounters M98 P9*xxx*.

● NOTE: An embedded subprogram definition must be placed before any calls to the subprogram.

M99 - Return from Macro or Subprogram

M99 designates the end of a subprogram or macro and transfers control back to the calling program when executed. M99 may be specified on a line with other G-codes. M99 will be the last action executed on a line. If M99 is not specified in a subprogram file, M99 is assumed at the end of the file:

Example:

G1 X3 M99 ;Move to X3 then return to calling program.

If M99 is encountered in the main job file, it will be interpreted as the end of the job. If M99 is encountered in an M-function macro file, it will be interpreted as the end of any enclosing subprogram or macro or as the end of the job.

M100 - Wait for PLC bit (Open, Off, Reset)

M101 - Wait for PLC bit (Closed, On, Set)

The M100/M101 commands wait for a PLC bit to reach a state as indicated in the table below.

Number	PLC bit	M100	M101
50001 – 51312	INP1 – INP1312	open	closed
60001 – 61312	OUT1 – OUT1312	off	on
70001 – 71024	MEM1 – MEM1024	reset	set
90001 – 90064	T1 – T64 status bits	reset (not expired)	set (expired)
93001 – 93256	STG1 – STG256 status bits	reset (disabled)	set (enabled)
94001 – 94256	FSTG – FTSG256 status bits	reset (disabled)	set (enabled)
The number ranges 1-240 can be used to reference the first eighty INP, OUT, or MEM bits. It is recommended that existing CNC10 programs and macros be converted to the new ranges for use with CNC11.			
1 – 80	INP1 – INP80	open	closed
81 – 160	OUT1 – OUT80	off	on
161 – 240	MEM1 – MEM80	reset	set

Example:

```
M101/50001  ; wait for INP1 to close
M100/60002 ; wait for OUT2 to turn off
M101/70123 ; wait for MEM123 to be set (1)
```

NOTE: The numbers assigned to the PLC bits (except 1-240) are the same as those that can be used when referencing system variables in M- and G-code programs.

M102 - Restart Program

M102 performs any movement requested, and restarts the program from the first line. The operator is **NOT** prompted to press the CYCLE START button to continue.

M103 - Programmed Action Timer

M103 is used to set up the time limit for a timed operation. If the timer is canceled (usually by M104) before the specified time limit , the program will be canceled and the message "Programmed action timer expired" will be displayed. If another M103 is issued before the time limit expires, then this time limit is nullified and the new time limit will be set up as specified by the latest occurring M103. Note also that if M0 or M1 causes the program to stop momentarily and the “M0 jogging” feature is enabled, then the the timer will also be canceled without the need to issue M104.

Example:

Activate a device and wait for a response. If no response within 4.5 seconds, cancel the program.

```
M94/12 ; turn on relay
M103/4.5    ; start 4.5 second timer
M100/4 ; wait for input 4 to open
M104   ; input 4 opened, cancel timer
```

M104 - Cancel Programmed Action Timer

M104 stops the timer started by the last M103 executed.

M105 - Move Minus to Switch

M105 moves the requested axis in the minus direction at the current feedrate until the specified switch opens (if the given P parameter is positive), or until the scecified switch closes (if P parameter is negative).

Example:

```
M105/X P5 F30                ; move the X axis in minus direction at 30"/min until
                             ; the switch on INP5 opens
G92 X10                      ; Sets X position to 10
M105/Z P-6                   ; move the Z axis in minus direction until switch on INP6 closes
```

M106 - Move Plus to Switch

M106 moves the requested axis in the plus direction at the current feedrate until the specified switch opens (if the given P parameter is positive), or until the scecified switch closes (if P parameter is negative).

Example:

```
M106/Z P3 F30          ; move the Z axis in the plus direction at 30"/min, until
                       ; the switch on INP3 opens
G92 X10                ; Sets Z position to 10
M106/X P-3             ; move the X axis in the plus direction until the switch on INP3 closes
```

M107 - Output Tool Number

M107 sends the current tool number to the automatic tool changer, via the PLC. M107 does not set the tool changer strobe or look for an acknowledge from the changer.

Example:

```
M107        ; send request for tool to changer
M94/16      ; turn on tool changer strobe
M101/5      ; wait for acknowledge on input 5
M95/16      ; turn off strobe
M100/5      ; wait for acknowledge to be removed
```

M108 - Enable Override Controls

M108 re-enables the feedrate override and/or spindle speed override controls if they have previously been disabled with M109. A parameter of 1 indicates the feedrate override; a parameter of 2 indicates the spindle speed override.

Example:

```
M109/1/2    ; disable feedrate and spindle speed overrides
M108/1      ; re-enable feedrate override
M108/2      ; re-enable spindle speed override
```

M109 - Disable Override Controls

M109 disables the feedrate override and/or spindle speed override controls. M109 cannot be used in MDI mode.

Example:

```
M3 S500     ; start spindle clockwise, 500 rpm
M109/1/2    ; disable feedrate and spindle speed overrides
M108/1/2    ; re-enable overrides
```

M115/M116/M125/M126 – Protected Move Probing Functions

The protected move probing functions provide the capability to program customized probing routines. The structure for these commands is:

```
Mnnn /Axis pos Pp Ff
```

nnn is either 115, 116, 125, or 126.
Axis is a valid axis label, i.e., X, Z, etc.
pos is an optional position
p is a plc bit number, which can be negative.
f is a feedrate (in units per minute.)
L1 is an option for the M115/M116 commands that prevents an error if the probe does not detect a surface
Q1 is an option for M115/M116 that forces the DSP probe to move a "Recovery Distance" on retries
(Note: the Q1 option only applies for DSP Probes)

For M115 and M116 functions, the indicated *axis* will move to *pos* (if specified) until the corresponding plc bit *p* state is 1, unless *p* is negative, in which case movement is until the plc bit state is 0. A *p* value of 1 to 80 (or -1 to -80) specifies plc bits INP1-INP80, 81 to 160 (or -80 to -160) specifies plc bits OUT1-OUT80, and 161 to 240 (or -161 to -240) specifies plc bits MEM1-MEM80. Warnings are generated in the CNC11 message window for "*Missing P value*" and "*Invalid P value*."

If *pos* is not specified, M115 will move *axis* in the negative direction, and M116 will move *axis* in the positive direction. Note that is *pos* is specified, then if does not matter whether M115 or M116 is used.

If *pos* is not specified, the movement is bounded by the settings in the software travel limits. In the absence of software travel limits, movement is bounded by the maximum probing distance (Machine Parameter 16). In cases where *pos* is specified, it is still bounded by the software travel limits.

If the bounded position is reached before the awaited plc bit state is found, a "*Probe unable to detect surface*" error will be generated unless the *L1* option is specified.

For M125 and M126 protected move functions, the behavior is identical to that of the M115 and M116 commands, except in regards to the plc bit state. M125 and M126 will generate an "*Unexpected probe contact*" error message if the specified plc bit state is triggered, again stopping any running job.

In summary, the M115 and M116 commands are to be used when one expects contact to be made and M125 and M126 commands are to be used when one does not expect any contact to be made.

Example:

```
M115/X P-15 F20      ; Move X minus at 20ipm waiting for contact on INP15
M116/X P15 F5        ; Move X plus until no contact at 5 ipm
```

M120 - Open data file (overwrite existing file)

This M function will open the requested data file for writing. If no drive or directory is specified with the file name, then the file will be opened in the same directory as the CNC program. If the file cannot be successfully opened, then an error will be returned, ultimately terminating the job. If a data file is already open when M120 is called, that file will first be closed, then the new file opened.

Example:

```
M120 "probetst.dat"
```

M121 - Open data file (append to existing file)

This M function will open the requested file for writing at the end of the file. If no drive or directory is specified with the file name, then the file will be opened in the same directory as the CNC program. If the file does not already exist, it will be created. This is not an error. If the file cannot be successfully opened, then an error will be returned, ultimately terminating the job. If a data file is already open when M121 is called, that file will first be closed, then the new file opened.

Example:

```
M121 "c:\probetst.dat"
```

M122 - Record position(s) and optional comment in data file

This M function will write the current expected position value to the data file, in the usual format (i.e. axis label before number, 4 decimal places in inch mode, 3 decimal places in millimeter mode. Any comment that appeared on the line with M122 will be output after the position(s). With no axis arguments, M122 will write the positions of all installed axes. With axis arguments, it will write the positions only of the requested axes. Positions will be written in local (not machine) coordinates, in native machine units. If no data file has been opened with M120 or M121 before M122 is called, then M122 will return an error and terminate the job. The parameter L1 may be used to suppress the new line character normally outputted after the last position. Furthermore, the output of axis labels, comma separators, and spaces can be enabled or suppressed via machine parameter 72 (see Parameter 72 in Chapter 14).

Examples (M function and sample output):

```
M122              ->    X1.2345 Z-0.5678
M122 /Z  ; at 10 ipm   ->    Z-.4321 ; at 10 ipm
M122 /X/Z              ->    X-1.0000 Z0.8732
M122 /X L1             ->    X-1.5000
M122 /X                ->    X-1.5000 X-2.0000
```

M123 - Record value and/or comment in data file

This M function will write the specified parameter value (if any) to the data file, followed by any comment that appeared on the line with M123. If a P value is specified, M123 will output a numeric value (4 decimal places in inches, 3 in millimeters). If no P value is specified, then M123 outputs the comment only. If neither a P value nor a comment was specified, M123 does nothing. This is not an error. If no data file has been opened with M120 or M121 before M123 is called, then M123 will return an error and terminate the job. The parameter L1 may be used to suppress the new line character normally outputted after the last value. The R and Q parameters can be used to specify the field width and precision, respectively. Furthermore, the output of axis labels, comma separators, and spaces can be enabled or suppressed via machine parameter 72 (see Parameter 72 in Chapter 14).

Examples (M function and sample output):

```
      M123 P1.2345                     ->1.2345
      M123 P#A ; first macro argument  ->1.2345 first macro argument
M123 ; Probing X+ to surface      ->Probing X+ to surface
M123                              -><nothing>
M123 ;                            -><nothing>
M123 ;; my comment                ->; my comment
M123 Q0 P1.23                     ->1
M123 Q1 P1.23                     ->1.2
M123 R7 Q2 L1 P1.234
M123 R7 Q2    P98.765             ->    1.23   98.77
```

M124 - Record machine position(s) and optional comment in data file

Identical to M122 above except that the m124 reports machine position instead of a local WCS position.

M127 - Record Date and Time in a data file

This M function is used to write the date, time, and year to the specified data file called out by the M120 or M121. **NOTE: This function does not insert a semi-colon in front of the date.** If desired, use an M123 as shown in example below.

Examples (M function and sample output):

M123 ;;
M121 “testdata.dat”
M127

If you opened testdata.dat you would see: Day of week, Month, day, time, and year.
(i.e. ;Wed Aug 29 11:56:57 2007)

M128 – Move Axis by Encoder Counts

M128 moves the requested axis by L which specifies an encoder count position.

Example:

M128/X L-5000 ; move the X axis to machine position -5000 counts

M129 - Record Current Job file path to data file

This M function is used to write the current job's file path to the specified data file called out by the M120 or M121.

Example:

Run a job named job.cnc which contains the following 2 lines:

```
M121 “output.txt”
M129
```

If you opened the output.txt file you would see:

```
c:\cncm\ncfiles\job.cnc
```

M130 - Run system command

This allows shell commands to be called from a CNC program or MDI. M130 takes one string argument which contains the system command to execute.

For example:

```
M130 "mycommand.bat"
```

will run the batch file *mycommand.bat*.

Normally, the command will run asynchronously, meaning that the G-code program will not wait for the command to finish before continuing. However, if an L1 parameter is given, the command will prevent further G-code execution until a fault occurs (such as E-Stop).

M150 – Set Spindle Position to 0 on Next Index Pulse

M150 will cause the spindle encoder position to be reset to 0 upon the next encounter of the spindle encoder's index pulse. M150 will not generate spindle movement. As a matter of fact, the spindle needs be be commanded to move in order for M150 to work.

M151 – Unwind C axis

This M function will reset the C axis position to less than one revolution of the C axis (< 360 degrees).

Example (M51)

```
G97         ; Turn off CSS (constant surface speed)
M3 S0       ; Turn off spindle
M101/9      ; Wait for zero speed signal form inverter on INP9
M94/9       ; Switch to torque mode
M51         ; Perform the default actions for C axis enable
M151        ; Unwind C-axis position
```

Note in the above examples for M50 and M51 where the M95/9 (turn off INP41) and M94/9 (turn on INP41) commands are used, it is assumed that the plc program, conditioned upon the state of INP41, has been modified to output the appropriate hardware signals required to switch between speed and torque mode.

```
Warning: The spindle must be stopped before issuing the M151 or
unpredictable positions can result.
```

Formatted String Commands- M200, M223, M224, M225 & M290

The formatted string commands are provided to assist in custom screen and file I/O. A "*formatted-string*" is similar to the C programming language "printf" command, with various restrictions. The basic form of a *formatted-string* is a quoted string (comprised of a **single** line of up to 1024 characters) followed by a (possibly empty) list of user and/or system variable expressions. The variable expression is a '#' character followed by a number or bracketed expression. For example, given #100 = 88* (ASCII 'X'), #300 = "absolute", and #101 = 1.2345, this string:

"The %c* axis %s position is %f" #100 #300 #101   evaluates to "The X* axis absolute position is 1.234500"

The "%c"* is replaced by the ASCII character value of user variable #100, the "%s" is replaced by the string user variable #300, and the "%f" is replaced by the value of user variable #101.

Type specifiers

The 's', 'c', and 'f' are type specifiers, with 's' specifying a string user variable, 'f' specifying a floating point user variable, and 'c' specifying a single character substitution using the integer part of a floating point user variable. There should be one user variable expression for every '%' character in the quoted string. It is also possible to specify a field width by inserting a number between the '%' and the type specifier. Example:

%20s – specifies that the substituted string is displayed in a field 20 characters long, right justified and padded with spaces on the left. Use "%-20s" for left justification.

The 'f' type can specify a precision such as:

- "%.4f" - display number rounded at the fourth decimal place.
- "%9.4f" - as above but in a field width nine characters wide.
- "%+9.4f" - as above with a '+' output if variable is positive.
- "%.0f" - display number rounded to integer

Special characters

The quoted string may contain one or more "\n", each of which will be converted to a single newline character. Up to seven newlines can be specified in a single formatted string. However, a formatted string may not contain an embedded quote character '"' or other *printf*-style escape sequences such as '\t', '\\', or '\"'. If a quote character is desired, use a %c type specifier with a variable expression equal to 34.

User string variables #300-#399. These variables can be assigned a quoted string up to 80 characters in length and are retained until the CNC software is exited. For example,

#300 = "This is a text string of characters"

*The above method of representing an axis label should be used only when writing to an external file or for display in a message box. It is not valid if you are attempting to "build" a motion command in real-time from within the currently running g code program. If your intent is to use a variable to represent an axis label for a real-time command, you should instead use $ as the placeholder. The parser will replace a '$' character and the numerical expression following it with the ASCII character equivalent to the numerical expression, provided that it evaluates to the characters 'A' (65) through 'Z' (90). If the numerical expression is out-of-bounds, an "Invalid character" error occurs. Ex:

Given #100 = 88, #101 = 1, #102 = 90, #103 = 2 and #104 = 10

G1 $[#100][#101] $[#102][#103] F[#104] evaluates to G1 X1 Z2 F10

M200/M201 – Stop for Operator, Prompt for Action

M200 is used to pause the currently running job and prompt the operator for action. If M0_jogging is unlocked, or the control is in DEMO mode, jogging is enabled while waiting for the operator to respond. If this option has not been enabled, the behavior will default to that of a standard M0. (jogging disabled)

The syntax is: M200 *formatted-string [[user_var_expr] ...]*

Example:

M200 "Please jog the %c and %c axes to the desired X0, Z0 position\nPress Cycle Start to continue" #100 #101

M201 behaves exactly like M200 except that PLC bits SV_PROGRAM_RUNNING, SV_MDI_MODE, and SV_JOB_IN_PROGRESS are turned off while the prompt is displayed.

M223 – Write Formatted String to File

The M223 command writes a *formatted-string* to a file that was opened using the M120 or M121 commands. The syntax is:

M223 *formatted-string [[user_var_expr] ...]*

Example:

M223 "; The measured diameter of the pocket = %.4f\n" #100

M224 – Prompt for Operator Input Using Formatted String

The M224 command displays a *formatted-string and then accepts user input. The synta*x is:

M224 lvalue*_expr formatted-string* `[[user_var_expr] ...]`

Where *lvalue_expr* is a *user_var_expr that* evaluates to a user variable that can be written. If *lvalue_expr* is a string type (#300-#399) then the user input is assigned verbatim to the string. Otherwise, the user input is evaluated as any other "bracketed" numerical expression.

Example:

M224 #300 "Please enter the direction that you wish to probe in the %c axis: (+ or -)" #100

M225 – Display Formatted String for A Period of Time

The M225 command displays a *formatted-string* for a specified period of time. The syntax is:

M225 *time_expr formatted-string [user_var] ...*

where *time_expr* is a *user_var_expr* that evaluates to a floating point variable specifying the number of seconds to display the output, with a value of zero interpreted as indefinitely. The CYCLE_START key can be used to immediately continue running without waiting for the time to expire.

Example:

M225 #100 "Warning, %s is not selected\nPlease select %s and press Cycle Start to continue" #300 #300

M290 – Digitize Profile (Optional)

This performs a 2 axis digitize, probing along an axis while stepping over using a perpendicular travel axis. M290 expects that a file is already open with M120/121 (however, if not open, there will be no output).

The syntax is:

```
  M290  /a__  #vvv  /b__  #vvvv  "formatted-string"  Q__  R__  P__  L__
```

The first axis mentioned will be treated as the probing axis and the second will be treated as the travel axis.

Explanation of M290 Arguments:
/a +-nnn.nnn is the probing direction and max distance on axis "a"
/b +-nnn.nnn is the travel direction and max distance on axis "b" (perpendicular to probing axis)
#vvv and #vvvv (optional) are G-code variables that will be a receptacle of the very last probed position of the cycle.
"formatted-string" (optional) is the format of the output (if not mentioned, then there will be no output to file)
Q__ is the stepover along the travel direction (this is a positive quantity)
R__ (optional) is the retract/pullback amount upon interruption or completion.
P__ (optional) is the interruption PLC bit state which causes a graceful end to the cycle. (If not mentioned, then no PLC bit will checked for graceful interrruption.)
L__ (optional) is the output variable to which to store the interrupt status
(0=no interruption, 1=interrupted by PLC bit P__, 2=surface not found error).

M300 – Fast Synchronous I/O update

There are 32 user definable fast system integer variables that can be used to communicate with the PLC (similar to M94 and M95), but without causing motion to decelerate to a stop* (unlike M94 and M95). The syntax is:

M300 /*nn* /*vvv*

where *nn* is 1-32 and *vvv* is a 32-bit signed integer value. The parameter *nn* (1-32) maps to system variables SV_FSIO1 - SV_FSIO32. These commands work in conjunction with a PLC program that can read the SV_FSIOx and act upon them.

Example:

```
M300 /21 /-1234 ; set SV_FSIO21 to integer value -1234
```

* NOTE: Motion will be decelerated to a stop if Smoothing is turned on (P220 = 1).

M1000-M1015 – Graphing Color for Feedrate movement

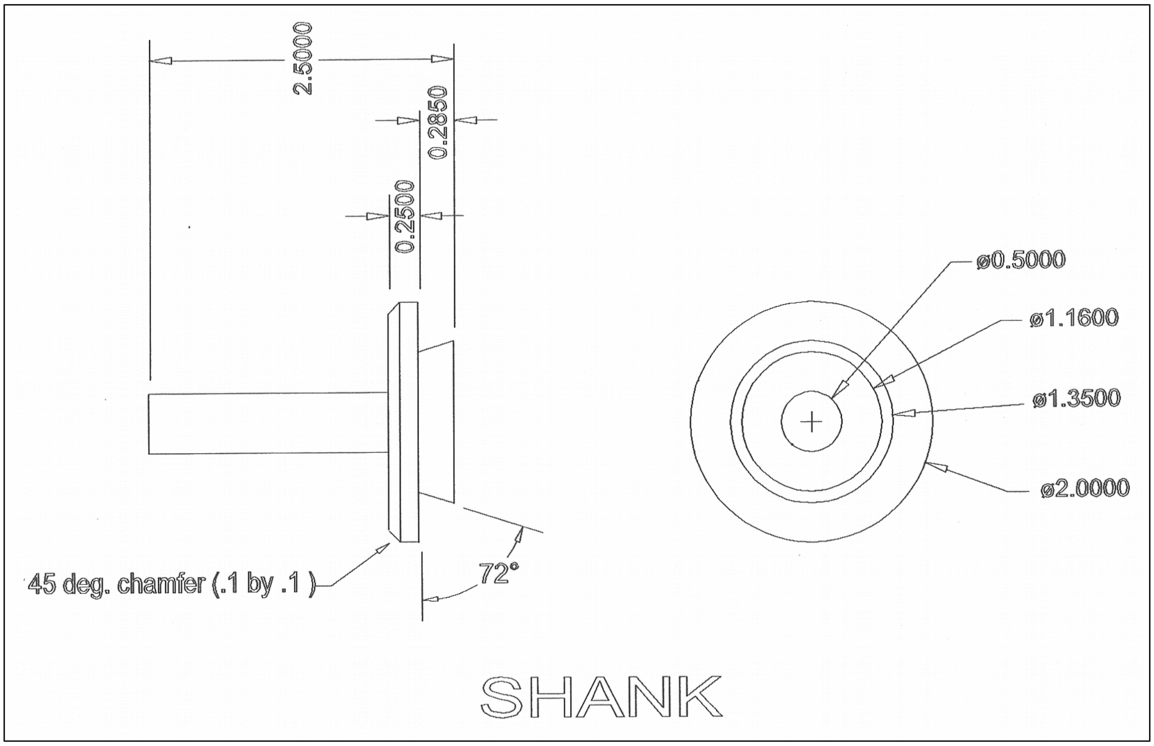
When a CNC program is graphed (F8 from the Main Screen), feedrate movements are normally plotted using the color yellow. This color setting can be changed to another color as stated in the chart below.

M Code	Feedrate Graphing Color
M1000	black
M1001	Navy blue
M1002	green
M1003	teal
M1004	orange
M1005	blue
M1006	lime
M1007	aqua

M Code	Feedrate Graphing Color
M1008	maroon
M1009	purple
M1010	olive
M1011	gray
M1012	red
M1013	fuschia
M1014	yellow
M1015	white

Changing this feedrate graphing color can be used as a method highlighting or hiding parts of a graphed CNC program, but will not affect the normal run of the program (when the CYCLE START button is pressed on the Main Screen). The limitations to using these M codes are as follows: These M codes cannot be placed on the same line as another M code, and also the rapid (G0) movement color cannot be changed.

Chapter 13
CNC Program Example



CNC Program

```
N010 G20
N015 G50 S3000
N020 G00 T0303
N025 G97 S1777 M03
N030 G00 X1.72 Z0.
N035 G96 S800
N040 X1.72
N045 G99 G01 Z-1.955 F.01
N050 X1.7901
N055 X2.02 Z-2.0699
N060 Z-2.215
N065 X2.04
N070 G00 Z0.
N075 X1.42
N080 G01 Z-1.955
N085 X1.74
N090 G00 Z0.
N095 X1.12
N100 G01 Z-1.955
N105 X1.44
N110 G00 Z0.
N115 X.82
N120 G01 Z-1.955
N125 X1.14
N130 G00 Z0.
N135 X.52
N140 G01 Z-1.955
N145 X.84
N150 G00 Z0.
N155 X.52
N160 G01 Z-1.955
N165 X.54
N170 G00 X2.1
N175 G97 S3000
N180 Z0.
N185 X.5
N190 G96 S1000
N195 G01 Z-1.965 F.003
N200 X1.7818
N205 X2. Z-2.0741
N210 Z-2.215
N215 G28 T0300
N220 M05
N225 M00
N230 G50 S3000
N235 G00 T0404
N240 G97 S1135 M03
N245 G00 X2.02 Z-2.228
N250 G96 S600
N255 X2.02
N260 G99 G01 X1.1932
N265 G00 X2.02
N270 Z-2.2392
N275 G01 X1.2005
```

```
N280 G00 X2.02
N285 Z-2.2503
N290 G01 X1.2078
N295 G00 X2.02
N300 Z-2.2615
N305 G01 X1.2151
N310 G00 X2.02
N315 Z-2.2727
N320 G01 X1.2224
N325 G00 X2.02
N330 Z-2.2838
N335 G01 X1.2297
N340 G00 X2.02
N345 Z-2.295
N350 G01 X1.237
N355 G00 X2.02
N360 Z-2.3062
N365 G01 X1.2444
N370 G00 X2.02
N375 Z-2.3173
N380 G01 X1.2517
N385 G00 X2.02
N390 Z-2.3285
N395 G01 X1.259
N400 G00 X2.02
N405 Z-2.3396
N410 G01 X1.2663
N415 G00 X2.02
N420 Z-2.3508
N425 G01 X1.2736
N430 G00 X2.02
N435 Z-2.362
N440 G01 X1.2809
N445 G00 X2.02
N450 Z-2.3731
N455 G01 X1.2882
N460 G00 X2.02
N465 Z-2.3843
N470 G01 X1.2956
N475 G00 X2.02
N480 Z-2.3955
N485 G01 X1.3029
N490 G00 X2.02
N495 Z-2.4066
N500 G01 X1.3102
N505 G00 X2.02
N510 Z-2.4178
N515 G01 X1.3175
N520 G00 X2.02
N525 Z-2.429
N530 G01 X1.3248
N535 G00 X2.02
N540 Z-2.4401
N545 G01 X1.3321
N550 G00 X2.02
N555 Z-2.4513
N560 G01 X1.3394
N565 G00 X2.02
N570 Z-2.4625
N575 G01 X1.3468
N580 G00 X2.02
N585 Z-2.4736
N590 G01 X1.3541
N595 G00 X2.02
N600 Z-2.4848
N605 G01 X1.3614
N610 G00 X2.02
N615 Z-2.496
N620 G01 X1.3687
N625 G00 X2.02
N630 G97 S1910
N635 Z-2.218
N640 X2
N645 G96 S1000
N650 G01 X1.1656
N655 X1.3497 Z-2.4991
N660 G28 T0400
N665 M05
N670 M30
```