



# DELTA TAU

## Power PMAC-NC 2016

Motion Commander Foundation  
© 2016 Greene & Morehead Engineering, Inc.



# Mill G-Code Manual

## Power PMAC-NC 2016



Delta Tau Data Systems, Inc.

May 27, 2016

O020-E-01

## **COPYRIGHT INFORMATION**

Software: © 2014 Delta Tau Data Systems, Inc. All rights reserved.

Software User Manual: © 2014 Delta Tau Data Systems, Inc. All rights reserved.

Motion Commander Foundation: © 2012-2014 Greene & Morehead Engineering, Inc. All rights reserved.

This document contains proprietary information of Delta Tau Data Systems, Inc. The information contained herein is not to be used by or disclosed to third parties without the express written permission of an officer of Delta Tau Data Systems, Inc.

## **TRADEMARK ACKNOWLEDGMENT**

Windows, Visual Studio and .NET Framework are registered trademarks of Microsoft Corporation. MTConnect is a registered trademark of the MTConnect Institute. Other brands, product names, company names, trademarks and service marks are the properties of their respective holders.

## **REVISION HISTORY**

<b>Version</b>	<b>Date</b>	<b>Description</b>
1.0	3/25/2016	Initial creation

# Power PMAC-NC 16™ - Mill G-Code Manual

---

## Contents

Introduction .....	6
NC Mill Basics .....	6
Tool Motion.....	6
Tool Movement Specification .....	6
Axis Move Specification .....	6
Feed Specification .....	6
Cutting Speed Specification .....	7
Tool Movement Considerations.....	7
Coordinate Systems .....	8
Machine Coordinates .....	8
Program Coordinates .....	8
Absolute Coordinate Positions.....	8
Incremental Coordinate Values .....	8
Reference Point.....	9
Machining Center G Code Library .....	9
G-Code Summary .....	9
G-Code Descriptions .....	11
G00 Rapid Traverse Positioning .....	11
G01 Linear Interpolation.....	11
G01.1 Spline Interpolation .....	11
G02 Circular Interpolation CW (Helical CW) .....	12
G03 Circular Interpolation CCW (Helical Interpolation CCW).....	12
G04 Dwell.....	13
G09 Exact Stop .....	14
G17/G18/G19 (XY/ZX/YZ) Plane Selection .....	14
G20/G21 Inch Mode/Metric Mode.....	14
G25 Spindle Detect Off .....	15
G26 Spindle Detect On.....	15
G28 Return to Reference Point.....	15
G29 Return from Reference Point .....	15
G40/G41/G42 Cutter Compensation .....	16

G43/G44/G49 Tool Length Compensation + /- and Cancel .....	17
G50/G51 Coordinate Scaling.....	17
G50.1/G51.1 Coordinate Mirroring .....	17
G52 Local Coordinate System Set .....	18
G53 Machine Coordinate Selection .....	18
G54-59 Work Coordinate System 1-6 Selection .....	19
G54.1 Pxx Auxiliary Work Coordinate System Selection.....	19
G61 Exact Stop Mode.....	19
G64 Cutting Mode.....	20
G68/G69 Coordinate System Rotation .....	20
G70 Bolt Hole Circle Pattern .....	20
G71 Arc Pattern.....	21
G72 Bolt Line Pattern .....	21
G73-89 Canned Cycles .....	22
G73 High Speed Peck Drilling Cycle-Short Retract.....	22
G74 Tapping Cycle – Float, Reverse .....	23
G76 Fine Boring Cycle .....	23
G80 Canned Cycle Cancel.....	24
G81 Drilling Cycle .....	24
G83 Deep Hole (Peck) Drilling Cycle .....	25
G84 Tapping Cycle – Float, Standard .....	26
G84.2 Tapping Cycle – Rigid, Standard .....	27
G84.3 Tapping Cycle – Rigid, Reverse .....	28
G85 Reaming, Boring Cycle .....	29
G86 Boring Cycle (Finishing cut) .....	29
G87 Back Boring Cycle.....	30
G89 Boring Cycle (Finishing Cut, Free Cutting) .....	31
G90 Absolute/G91 Incremental Mode.....	31
G90.1 Absolute Arc Radius/G91.1 Incremental Arc Radius .....	32
G92 Work Coordinate System Set.....	32
G93 Inverse Time Feed .....	33
G94 Feed Per Minute .....	33
G98/G99 Canned Cycle Return Point.....	33
M Code Library – CNC M-Codes.....	34

M00 Program Stop .....	34
M01 Optional Stop .....	34
M02 End of Program with Rewind .....	34
M03 Spindle Clockwise .....	35
M04 Spindle Counterclockwise .....	35
M05 Spindle Stop .....	35
M06 Tool Change .....	35
M07 Coolant Mist On .....	35
M08 Coolant Flood On .....	35
M09 Coolant Off .....	35
M19 Spindle Orient .....	36
M30 End of Program with Rewind .....	36
M98 Sub-Program Call .....	36
Description .....	36
P Code Specification .....	36
Comment Specification .....	37
Volatile vs. Non-Volatile Sub-Programs .....	37
M99 Return from Subroutine / Program Repeat .....	37
T-Codes .....	38
T-Code Format .....	38
Miscellaneous .....	38
Block Delete Character: / .....	38

# Power PMAC-NC™ - Software User Manual

## Introduction

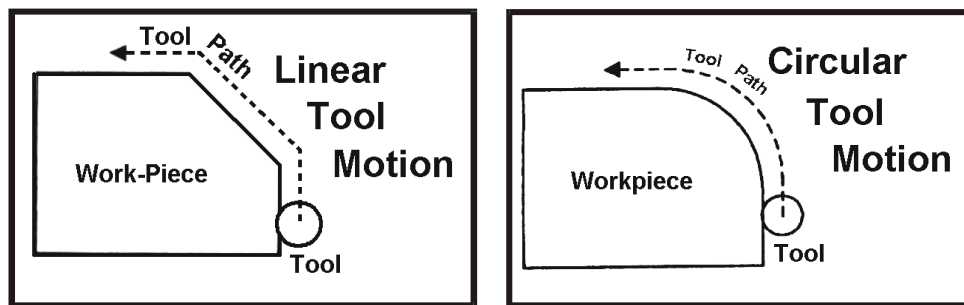
This reference manual describes the basic G-code set and instructions for the Power PMAC NC Software (Mill Version). The goal of this document is to provide descriptions of the RS-274 style G/M/T/D codes used in a milling environment.

The default G-codes delivered with the Power PMAC NC Software are designed to emulate a typical generic milling G-code set. Typically CAM systems will include a “Generic” machine post which in general will be compatible.

## NC Mill Basics

### Tool Motion

The tool moves through lines and arcs within the table boundaries as required to manufacture a part. In a working machine, the table is moved in relation to the rotating tool, so the actual table displacement will be the reverse of commanded tool motion.



### Tool Movement Specification

Program commands for NC machines are called the preparatory functions, also known as G codes. The function of moving the table along straight lines and arcs is called interpolation. Preparatory functions specify the type of interpolation used. The three basic interpolation preparatory functions are:

1. Table movement along straight line: **G01**
2. Table movement along circular arc: **G02 / G03**
3. Table movement along specified trajectory: **G01.1**

Reference to the axis position word executes motion. The PMAC controller coordinates the movement of the axis motors to execute the command. In this document, the generalized form of the axis position word, **X\_Y\_Z\_**, is used.

### Axis Move Specification

The last commanded position is the starting position of a move and the final position is the commanded position. The final position may be either an absolute position (a point referenced to program zero) or a relative move (signed incremental distance from the previous point). This is specified with axis move or position words, the axis address letter followed by a numeric literal:

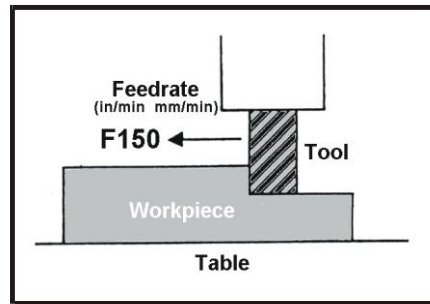
N100 X5.2Y0Z-.001 (length units in. or mm.)

### Feed Specification

Movement of the table at a specified speed for cutting a work piece is called the feedrate. Feedrates can be specified similarly with the feed word:

N100 F150.0 (length/time units in/min or mm/min or mm/sec)

Length units are within program control (see the G-code definitions in the next section). The machine builder sets the time units.



**Tool Feedrate Example**

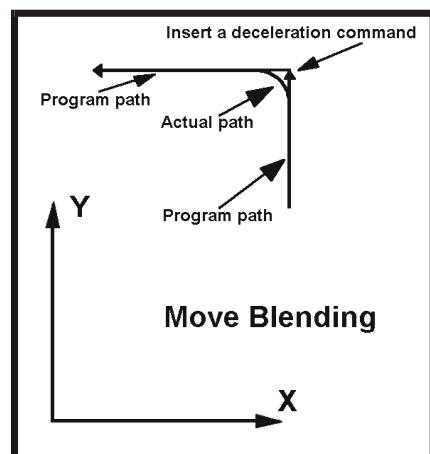
## Cutting Speed Specification

The relative rotational speed of the tool with respect to the work piece during a cut is called the spindle speed. As for the CNC, the spindle speed can be specified in rpm units, using the S address letter followed by the value:

N100 S250 (rpm units)

## Tool Movement Considerations

At multiple move (or block) boundaries, the CNC applies a coordinated ramp of the vector velocity into and out of the point without stopping. The result of this is called move blending. Because of blending, corners are not cut sharply. If sharp corners are required to be cut, Exact Stop or Dwell must be commanded in the block or set modally (see **G04**, **G09**, **G61**). This forces an in-position stop before starting the next move. In-position means that the feed motor is within a specified range about the commanded position.



**Move Blending Example**

## Coordinate Systems

There are two types of coordinate systems. One is fixed by the machine mechanics, and the other is a relative coordinate system specified by the NC program that coincides with the part drawing. The control is aware only of the fixed one. Therefore, to correctly cut the work piece as specified on the drawing, the two coordinate systems must be specified at machine startup. When a work piece is set on the table, these two coordinate systems are as follows:

- Coordinate system specified by the CNC: Machine Coordinates
- Coordinate system specified by the part: Program Coordinates

## Machine Coordinates

The machine zero point is a standard reference point on the machine. The machine coordinate system is established when the reference point return is first executed after the machine power is turned on or the homing cycle is executed. Once the machine coordinate system is established, it is not changed. A G-code program will not execute without the machine coordinate system being established first (i.e., all the machine axes must be homed before a G-code program can be executed). Machines with absolute feedback may not require homing.

## Program Coordinates

The Program coordinates are always within one of the Work coordinate systems, **G54** through **G59** (or supplemental **G54 Pxx**), and are either absolute positions or incremental values. A Work coordinate offset,  $W_{off}$ , defines the position within the Machine coordinate space. Within the Work coordinate system, a Local coordinate offset,  $L_{off}$ , may define a Local coordinate system. When no Work or Local offsets are in effect, or the work coordinates are zero, then the Machine and Program coordinates are the same. It is possible that the Machine zero position is not accessible by the tool.

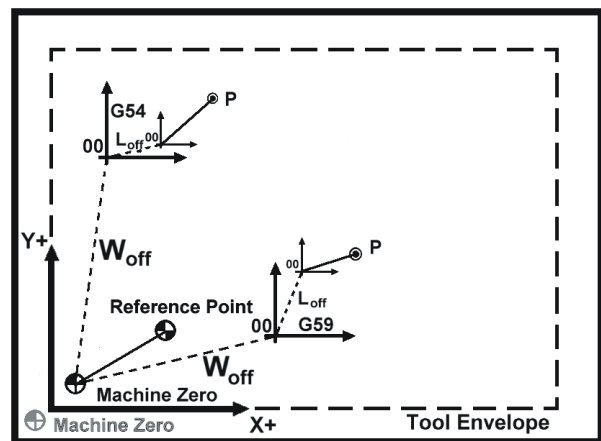


Table Envelope  
Coordinate and Reference Point Examples

## Absolute Coordinate Positions

The table moves to a point at the distance from zero point of the coordinate system (i.e. to the position of the coordinate values). Specify the table movement from point A to point B by using the coordinate values of point B.

## Incremental Coordinate Values

This specifies table moves relative to the current table position. A move from point A to point B will use the signed difference between the two points. The term Relative is also used.



## Reference Point

Aside from Machine zero, a machine tool may need to locate other fixed positions corresponding to attached hardware (i.e. a tool changer). This position is called the reference point (which may coincide with Machine zero). The tool can be moved to the reference point in two ways either manually or automatically.

In general, manual reference point return is performed first after the machine power is turned on. Usually this is the same as the homing function, since the reference point is at a fixed offset from the Machine zero position. In order to move the tool to the reference point for tool change thereafter, the function of automatic reference point return is used.

## Machining Center G Code Library

### G-Code Summary

Group 01	G00	Rapid Traverse
	G01	Linear Interpolation
	G01.1	Spline Interpolation
	G02	Circular/Helical Interpolation CW
	G03	Circular/Helical Interpolation CCW
Group 00	G04	Dwell
	G09	Exact Stop
Group 02	G17	XY Plane Select
	G18	ZX Plane Select
	G19	YZ Plane Select
Group 06	G20	Imperial Input Mode
	G21	Metric Input Mode
Group 00	G28	Return to Reference Point
Group 07	G40	Cutter Compensation Cancel
	G41	Cutter Compensation Left
	G42	Cutter Compensation Right
Group 08	G43	Tool Length Compensation Positive Direction +
	G44	Tool Length Compensation Negative Direction -
	G49	Tool Length Compensation Cancel
Group 11	G50	Scaling Cancel
	G51	Scaling
Group 22	G50.1	Mirror Image Cancel
	G51.1	Mirror Image
Group 00	G52	Local Coordinate System Set
	G53	Machine Coordinate System Set

Group 14	G54	Work Coordinate System Select 1
	G55	Work Coordinate System Select 2
	G56	Work Coordinate System Select 3
	G57	Work Coordinate System Select 4
	G58	Work Coordinate System Select 5
	G59	Work Coordinate System Select 6
	G54.1 Pxx	Auxiliary Work Coordinate System Select (P01 – P100)

Group 15	G61	Exact Stop Mode
	G64	Cutting Mode

Group 16	G68	Coordinate System Rotation
	G69	Coordinate System Rotation Cancel

Group 09	G70	Bolt Hole Circle Pattern
	G71	Bolt Hole Arc Pattern
	G72	Bolt Line Pattern
	G73	High Speed Peck Drilling Cycle
	G74	Tapping Cycle Float – Left Hand
	G76	Fine Boring Cycle
	G80	Canned Cycle Cancel
	G81	Spot Drilling Cycle
	G83	Peck Drilling Cycle
	G84	Tapping Cycle Float – Right Hand
	G84.2	Tapping Cycle Rigid – Right Hand
	G84.3	Tapping Cycle Rigid – Left Hand
	G85	Boring Cycle
	G86	Boring Cycle
	G87	Back Boring Cycle
	G89	Boring Cycle

Group 03	G90	Absolute Programming Mode
	G91	Incremental Programming Mode
	G90.1	Arc Radius Absolute Mode
	G91.1	Arc Radius Incremental Mode

Group 00	G92	Workpiece Coordinate System Set
----------	-----	---------------------------------

Group 05	G93	Inverse Time Mode
	G94	Feed Per Minute Mode

Group 10	G98	Canned Cycle Return Point – Return to Initial Point
	G99	Canned Cycle Return Point – Return to R Plane

## G-Code Descriptions

### G00 Rapid Traverse Positioning

This is used to position the tool from the current programmed point to the next programmed point at maximum traverse rate. **G00** is group 01 modal. It is canceled by other group 01 functions. The rapid move mode uses an approximated linear trajectory so that all axes complete in the same amount of time. The slowest axis determines the overall move time. Rapid moves are never blended with adjacent blocks.

**Syntax:** G00 X\_Y\_Z\_

**Example Code:**

```
N005 G49 G54 G20 G90 G40 G80
N010 S2500 M03
N015 G55
N020 G20 G90 G00 X0 Y0
```

### G01 Linear Interpolation

Linearly interpolates the position of the tool from the current point to the programmed point in the **G01** block. The speed of the tool is controlled by the modal feedrate word **F** and is the vector velocity of the tool path defined by:

$$F_x = F * \frac{L_x}{\sqrt{L_y^2 + L_x^2}}; F_y = F * \frac{L_y}{\sqrt{L_y^2 + L_x^2}}$$

Linear moves may blend with adjacent interpolative blocks. If the **G01** block contains a Dwell (**G04**) or an Exact Stop (**G09**), a controlled deceleration to a stop with in-position going true will inhibit blending with the next block. If the **G61** modal Exact Stop is active, no blending between linear blocks will occur until canceled (**G64** Cutting Mode). **G01** is group 01 modal. It is canceled by other group 01 functions.

**Syntax:** G01 X\_Y\_Z\_F\_

**Example Code:**

```
N030 X1.125 Y2.25
N040 G61 G01 Z-.02 F20
N050 G64 G03 X0.5 Y2.0 R0.375
```

### G01.1 Spline Interpolation

Spline interpolation mode puts the motion program in the cubic B-spline move mode and sets the spline move time(s), expressed in milliseconds using arguments A, B, and C. If only a single time is specified, all three sections will use this time. If the motion program is already in the cubic B-spline mode, the command can be used to change the time for the moves (this can be done without stopping). Subsequent move commands in the motion program will be processed according to the rules of this mode.

In the cubic B-spline mode, each programmed move consists of three sections. In the middle of a spline mode sequence, the sections for each programmed move overlap sections from the previous and following programmed moves, with the resulting profile derived from the superposition of these three moves. See the Power PMAC Software Reference Manual for more detailed information. **G01.1** is modal in-group 01. It is canceled by other group 01 functions.

**Syntax:** G01.1 A\_B\_C\_

**Example Code:**

```
N6 Z.1 H1 M8
N7 G01.1 A5.5
N8 X10 Y10
N9 X10.2236 Y10.2236
N10 X10.4729 Y10.4729
```

## G02 Circular Interpolation CW (Helical CW)

Circular interpolation uses the axis information contained in a block to move the tool in a clockwise arc of a circle, up to 360 degrees. The velocity at which the tool is moved is controlled by the feedrate word and is a vector tangent in the interpolation plane:

$$F_t = \sqrt{f_x^2 + f_y^2}.$$

All circles are defined and machined by programming three pieces of information to the PMAC. They are:

- Start Point of the arc
- End Point of the arc
- Arc Center of the arc or Arc Radius

The Start Point is defined prior to the **G02** line, usually by a **G01** or **G00** positioning move. The End Point is defined by the axis coordinates within the **G02** line. The Arc Center is defined by the **I J K** values (vector incremental from the start point) or the **R** value within the **G02** line. The full format for a **G02** line must reflect in which plane the arc is being cut. This is accomplished by use of a G code to define the interpolation plane and the letter addresses **I J K**.

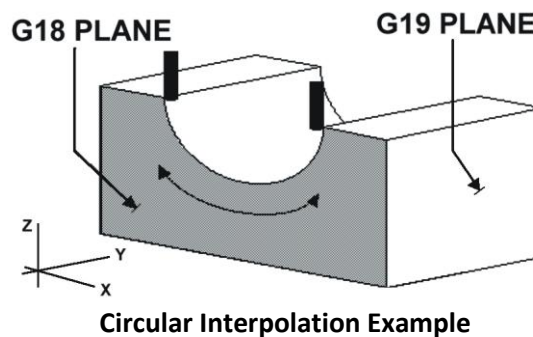
- G17 (XY - Plane) Letter address I for X Letter address J for Y
- G18 (XZ - Plane) Letter address I for X Letter address K for Z
- G19 (YZ - Plane) Letter address J for Y Letter address K for Z

The **I, J** and **K** vector incremental values are signed distances from where the tool starts cutting (Start Point) the arc to the Arc Center. For 90-degree corners or fillets, the **I J K** values can be determined easily. The **G17** (XY - Plane) is the default or power on condition. If another axis not specified in the circular interpolation is programmed, then helical cutting will be affected.

**Syntax:** [G17/G18/G19] G02 X\_Y\_Z\_I\_J\_K\_F\_  
[G17/G18/G19] G02 X\_Y\_Z\_R\_F\_

### Example Code:

```
N040 G73 G1 Z-.02 F20  
N050 G64 G02 X0.5 Y2.0 R0.375  
N060 G1 Y1.5625
```



## G03 Circular Interpolation CCW (Helical Interpolation CCW)

Circular interpolation uses the axis information contained in a block to move the tool in a counterclockwise arc of a circle, up to 360 degrees. The velocity at which the tool is moved is controlled by the feedrate word and is a vector tangent in the interpolation plane:

$$F_t = \sqrt{f_x^2 + f_y^2}.$$

All circles are defined and machined by programming three pieces of information to the PMAC. They are:

- Start Point of the arc
- End Point of the arc
- Arc Center of the arc

The Start Point is defined prior to the **G02** line, usually by a **G01** or **G00** positioning move. The End Point is defined by the axis coordinates within the **G02** line. The Arc Center is defined by the **I J K** values (vector incremental from the start point) or the **R** value within the **G02** line. The full format for a **G02** line must reflect in which plane the arc is being cut. This is accomplished by use of a G code to define the interpolation plane and the letter addresses **I J K**.

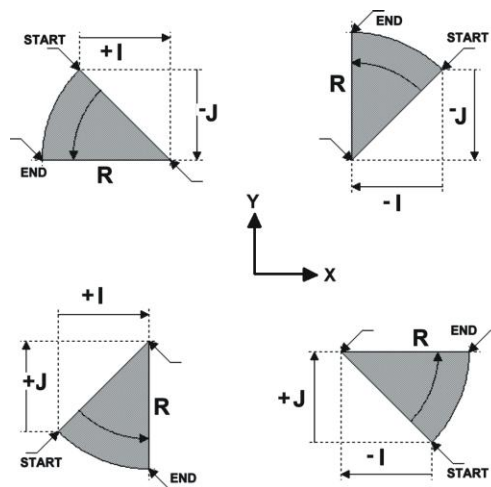
- G17 (XY - Plane) Letter address I for X Letter address J for Y
- G18 (XZ - Plane) Letter address I for X Letter address K for Z
- G19 (YZ - Plane) Letter address J for Y Letter address K for Z

The **I**, **J** and **K** vector incremental values are signed distances from where the tool starts cutting (Start Point) the arc to the Arc Center. For 90-degree corners or fillets, the **I J K** values can be determined easily. The **G17** (XY - Plane) is the default or power on condition. If another axis not specified in the circular interpolation is programmed, then helical cutting will be affected.

**Syntax:** [G17/G18/G19] G03 X\_Y\_I\_J\_F\_  
[G17/G18/G19] G03 X\_Y\_R\_F\_

#### Example Code:

```
N4 G0 G90 G17 S500 M3
N5 X0 Y1.0156
N6 Z.1 H1 M8
N7 G03 I1 J1 Y0 X2 F150.
```



**Circular Interpolation Example**

## G04 Dwell

When programmed in a block following some motion such as **G00**, **G01**, **G02** or **G03**, all axis motion will be stopped for the amount of time specified in the **F**, **P** or **X** word in seconds. Only axis motion is stopped; the spindle and machine functions under PLC control are unaffected. If no parameter is specified then a default value of 0 seconds dwell is executed.

**Syntax:** G04 X\_, G04 P\_, G04 F\_

**Example Code:**

```
N4 G0 G90 S500 M3
N5 X0 Y1.0156
N6 Z.1 H1 M8
N7 G04 X10
N8 G04 P0.055
```

## G09 Exact Stop

This forces a controlled deceleration to a stop, with in-position registration, at the end of the block. This is used to prevent move blending with the next block (i.e. sharp corners are cut). **G09** is not modal. It is valid for the current block only and is affected by issuing a dwell of zero time (see **G61** for modal Exact Stop).

**Syntax:** G09

**Example Code:**

```
N030 X1.125 Y2.25
N040 G09 G1 Z-.02 F20
N050 G3 X0.5 Y2.0 R0.375
```

## G17/G18/G19 (XY/ZX/YZ) Plane Selection

When cutting motion is for **X** and **Y** using circular interpolation, the **G17** plane must be in effect. The **G17** plane is a power on default, so normally is not programmed. When cutting motion is for **Z** and **X** using circular interpolation, the **G18** plane must be in effect. When cutting motion is for **Y** and **Z** using circular interpolation, the **G19** plane must be in effect.

**Syntax:** G17/G18/G19

**Example Code:**

```
N4 G0 G90 G17 S500 M3
N5 X0 Y1.0156
N6 Z.1 H1 M8
N7 G03 I1 J1 Y0 X2 F150
```

## G20/G21 Inch Mode/Metric Mode

Either inch or metric dimensional data may be selected by programming a **G20** (inch) or **G21** (metric) code. The **G20** or **G21** code must be programmed before setting the coordinate system at the beginning of the program. The inch/metric status is the same as that in effect before the power was turned off or the control was reset. Stored information, such as tool offset values, is automatically converted to the active measurement state when the **G20** or **G21** command is issued. All manual input data is assumed to be in the units of the current **G20** or **G21** mode, such as the values input on the tool offset page and the work offset page.

**Syntax:** G20, G21

**Example Code:**

```
N005 G49 G20 G90
N010 S2500 M03
N015 G55
```

## G25 Spindle Detect Off

**G25** turns Spindle Speed Detect OFF. When G25 is active the program will not wait for the spindle to achieve an *At Speed* state before moving forward. G25 is the default state of the control.

**Syntax:** G25

## G26 Spindle Detect On

**G26** turns Spindle Speed Detect ON. When G26 is active the program will wait for the spindle to achieve an *At Speed* state before moving forward.

**Syntax:** G26

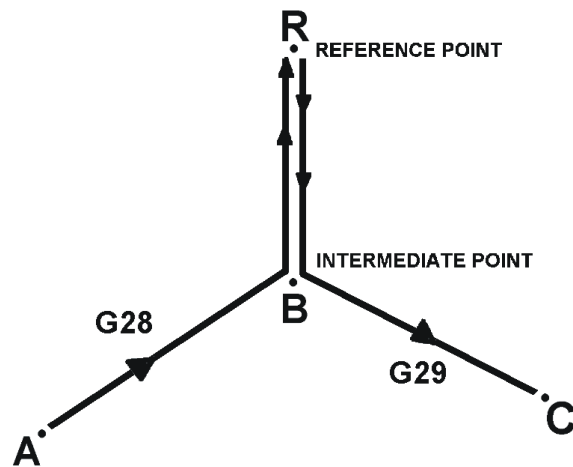
## G28 Return to Reference Point

**G28** positions the tool at rapid traverse to the optional intermediate point (ip) and then the reference point. The ip is saved for subsequent use by **G29**. If no intermediate point is specified the machine will move directly to the reference point.

**Syntax:** G28 (X\_\_Y\_\_Z\_\_)

### Example Code:

```
N4 G0 G90 S500 M3
N5 G28 X0 Y1.0156 Z.1
```



Return to Reference Point Example

## G29 Return from Reference Point

**G29** will command the tool to move to the ip position saved during the **G28** command executed immediately prior, and then to the positions specified in the **G29**.

**Syntax:** G29 (X\_\_Y\_\_Z\_\_)

### Example Code:

```
N4 G0 G90 S500 M3
N5 G29 X10.514 Y11.741 Z.3
```

## G40/G41/G42 Cutter Compensation

Cutter compensation allows programming of paths without knowing the diameter of the tool which will be used ahead of time. Cutter compensation allows the selection of the tool at the control and automatically moves the tool to the left or right of the programmed path to compensate for the diameter of the tool being called.

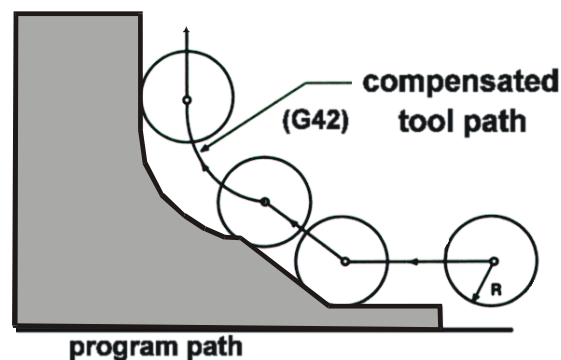
The control will offset the tool normal to the instantaneous surface tangent of the work piece with respect to the direction of tool motion in the compensation plane. This allows a programmer to compensate for cutters of different radial dimensions without the need for complex trigonometric code changes. Climb milling will use G41 to instate cutter radius compensation. Conventional milling will use G42 to instate cutter radius compensation. Of greatest concern is how to position the tool just prior to the start up of cutter radius compensation. The control will not engage compensation unless a move having a vector component in the compensation plane is commanded.

- **G40** - Cancel cutter radius compensation
- **G41** - Cutter compensation, tool on the left of the work piece (in the feed direction)
- **G42** - Cutter compensation, tool on the right of the work piece (in the feed direction)

When activating cutter compensation (**G41/G42**), care must be taken in selecting a clearance move in the compensation plane. On start up, the tool will move a vector distance equal to the offset value + the initial compensation in-plane move. The tool must be positioned so that as the compensation engages, the tool begins cutting normal to the surface. In addition, the center of the cutter must be at least the cutter radius away from the first surface to be machined. Cutter radius compensation is modal. Once cutter radius compensation is correctly engaged, it will remain in effect until it is canceled.

Make any zero component compensation plane axis moves before cutter compensation. Make an axis startup move, having a non-zero component in the compensation plane (G17/18/19), on or immediately after the G41 or G42 block. The compensation adjustment will be vectored with this move.

The programmer must consider this effect when moving out of the current plane, as during depth changes in pocket milling. Execute a move whose vector component in the compensation plane parallel to the last in-plane compensation move, but have opposite direction is interpolated with the intended out-of-plane axis move.



### Cutter Radius Compensation Example

When deactivating cutter compensation (**G40**), care must be taken in selecting a clearance move. If the move is omitted, the control will not cancel cutter radius compensation (and resulting axis motion) until a block with a non-zero move component in the compensation plane is executed. Do not cancel cutter compensation on any line that is still cutting the part. Cancel of cutter compensation may be a one- or two-axis move. When cutter compensation is active, the control applies a virtual cutter with zero diameter. The physical or actual diameter of the cutter is stored in the control by the operator on the page that contains the cutter tool lengths and diameters.



Normally the tool length and the tool diameter are assigned the same tool offset number. Cutter compensation takes the stored value for the diameter and calculates the cutter path offset from that value. Because of look-ahead, care must be taken that programmed moves do not violate the called-for compensation.

**Syntax:** G42 (D\_X\_Y\_F\_), G42 (D\_X\_Y\_F\_), G40 (X\_Y\_F\_)

**Example Code:**

```
G01 X-1.993 Y.487 Z0. F6.42
G41 D2 X-1.658 Y.118
G3 X-1.28 Y-.0443 I.369 J.337
```

## G43/G44/G49 Tool Length Compensation + /- and Cancel

Program zero is a point of reference for a part program, usually from a key location on the work piece. The position of the tool's center in **X** and **Y** does not change as the tool changes. In the Z-axis, this is not the case. If the length of the tool changes, so does the distance from the tip of each tool to the program zero point in **Z**. Note that each tool has a different distance from the tip of the tool to a surface on the part.

Tool length compensation lets the control call out the Z axis movements in a program as the tool changes, although, physical interference problems between the work piece and the tool must still be overcome by the programmer.

The programmer initializes tool length compensation in each tool's first Z-axis approach move to the work piece. This initialization command includes a **G43/G44** word and an **H** or **T** word to invoke the desired tool offset. It must also contain a Z-axis positioning move. Tool length compensation is modal. Once in effect it remains until cancelled or changed. **G49** cancels tool length compensation that is in effect.

**Syntax:** G43 Z\_\_, G44 Z\_\_, G49

**Example Code:**

```
N020 G20 G90 G0 X0 Y0
N025 G43 Z0.25 H1
N030 X1.125 Y2.25
```

## G50/G51 Coordinate Scaling

**G51** is used for coordinate scaling where **XYZ** is the center of scaling and **IJK** are the scaling magnitudes for each axis respectively. These parameters are meaningful only in absolute mode. A **P** can be used instead of the **IJK** if a uniform scaling is desired for all axes. When performing circular interpolation specified by a Radius, the maximum value of the scaling magnification for the appropriate plane is applied to the Radius component. For example, if the selected plane is the **X-Z** plane then the maximum magnification of **X-Z** is used to scale **R**. Likewise. If the selected plane is the **X-Y** plane, the maximum magnification of **Y** is used to scale **R**. When performing circular interpolation with **I J K** components, each component is magnified by its appropriate scale factor.

Coordinate scaling is canceled with **G50**.

**Syntax:** G51 I\_J\_K\_P\_X\_Y\_Z  
G50

### G50.1/G51.1 Coordinate Mirroring

**G51.1** is used for image mirroring where **XYZ** defines the axis to mirror about. The value of this parameter is meaningful only in absolute mode. It indicates the line about which mirroring occurs. In incremental mode, only the axis letter is meaningful and the actual value may be anything.

Mirroring is canceled with **G50.1**.

**Syntax:**        G51.1 X\_\_Y\_\_Z\_\_  
                  G50.1

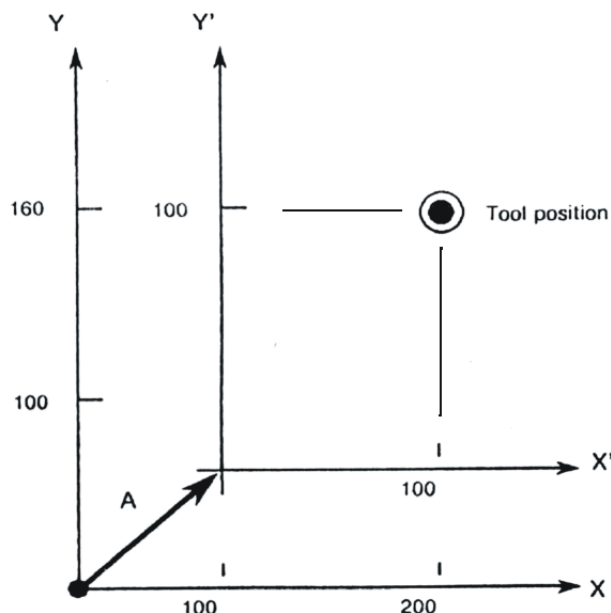
## G52 Local Coordinate System Set

While programming in a work coordinate system, it is sometimes more convenient to have a common coordinate system within all the work coordinate systems. This coordinate system is called a local coordinate system. The **G52** specifies the local coordinate system. The Local CS ( $x'y'$ ) is offset from the Work CS ( $xy$ ) by the vector (**A**) that makes the current tool point in the Local CS equal to the position word in the **G52** block (**G52 X100 Y100**). The local coordinate system can be changed by specifying the **G52** command with the zero point of a new local coordinate system in the work coordinate system. To cancel the local coordinate system, specify **G52 X0 Y0**.

**Syntax:**        G52 X\_\_Y\_\_Z\_\_  
                  G52 X0 Y0 (To Cancel)

### Example Code:

```
N4 G0 G90 S500 M3
N5 G52 X.0157 Y1.0156 Z0
```



Local Coordinate System Example

## G53 Machine Coordinate Selection

The machine zero point is a standard point on the machine. A coordinate system having the zero point at the machine zero point is called the machine coordinate system. The tool cannot always move to the machine zero point. The machine coordinate system is established when the reference point return is first executed after the power is on. Once the machine coordinate system is established, it is not changed by reset, change of work coordinate system (**G92**), local coordinate system setting (**G52**) or other operations unless the power is turned off. Occasionally it is necessary to move the axes to a specific position in relation to machine zero and ignore any tool and work offsets that are active. This is accomplished using **G53** for machine coordinate programming. Machine coordinates are always expressed as absolute coordinates. All **G92** codes and offsets are ignored. The tool is moved to the absolute Machine coordinates expressed in the **G53** block.

**Syntax:**        G53 X\_\_Y\_\_Z\_\_

**Example Code:**

```
N4 G53 X0 Y0 Z0
```

## G54-59 Work Coordinate System 1-6 Selection

There are six standard (**G54-G59**) work offset which can be used and selected at any time during the machining process.

- Work coordinate system 1 **G54**
- Work coordinate system 2 **G55**
- Work coordinate system 3 **G56**
- Work coordinate system 4 **G57**
- Work coordinate system 5 **G58**
- Work coordinate system 6 **G59**

The six coordinate systems are determined by setting distances (work zero offset values) in each axis from the machine zero point to their respective zero points. The offsets are saved in the *Work Offsets* tab of the Power PMAC NC16 program.

**Example Code:**

```
G55 G00 X20.0 Z100.0
X40.0 Z20.0
```

In the above example, positioning is made to positions (X=20.0, Z=100.0) and (X=40.0, Z=20.0) in work coordinate system 2. Where the tool is positioned on the machine depends on work zero point offset values.

Work coordinate systems 1 to 6 are established after reference point return (or homing) after the power is turned on. When the power is turned on the machine is in Machine Coordinates by default.

**Syntax:** G54-G59

## G54.1 Pxx Auxiliary Work Coordinate System Selection

There are 100 extra work coordinate systems available via the G54.1 Pxx command. These function identically as the six standard (**G54-G59**) work offsets.

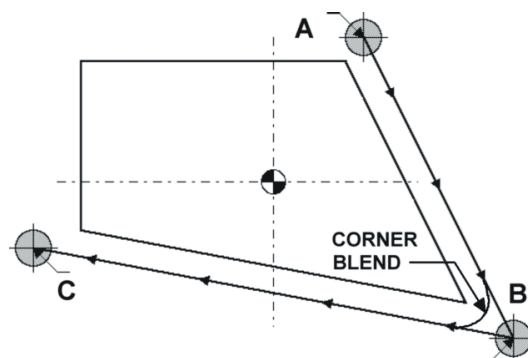
**Example Code:**

```
G54.1 P23 G00 X20.0 Z100.0
X40
```

## G61 Exact Stop Mode

**G61** causes a stop between block moves so that no corner rounding or blending between the moves is done (i.e. sharp corners are cut). When **G61** is commanded, deceleration is applied to the end point of the cutting block, and the in-position check is performed every block thereafter. The **G61** is valid until **G64** (cutting mode). Cutting mode (**G64**) is the startup default.

**Syntax:** G61



**Exact Stop Mode Example**

## G64 Cutting Mode

When **G64** is commanded, deceleration at the end point of each block is not performed thereafter, and cutting is blended to the next block. This command is valid until **G61** (exact stop mode) is commanded. However, in **G64** mode, feed rate is decelerated to zero and in-position check is performed in the following cases:

- Positioning mode (**G00**)
- Block with exact stop check (**G09**)
- Next block is a block without movement command

**Syntax:**            G64

## G68/G69 Coordinate System Rotation

A programmed shape can be rotated about a point. By using this function, it becomes possible, for example, to modify a program using a rotation command when a work piece has been placed with some angle rotated from the programmed position on the machine. Further, when there is a pattern comprising some identical shapes in the positions rotated from a shape, the time required for programming and the length of the program can be reduced by preparing a subprogram of the shape and calling it after rotation. The angle of rotation (+ is the CCW direction) is commanded with a signed angle value in decimal degrees using the **R** address in the **G68** block. The center of rotation is specified in the block with axis data **X Y Z**. After this command is specified, subsequent commands are rotated by the specified parameters. Command the angle of rotation (R) within the range of –360 to 360 degrees.

A rotation plane must be specified (**G17**, **G18**, **G19**) when **G68** is designated, though not required to be designated in the same block. **G68** may be designated in the same block with other commands. Tool offsets, such as cutter compensation, tool length compensation, or tool offset is performed after the coordinate system is rotated for the command program. The coordinate system rotation is cancelled by **G69**.

**Syntax:**            G68 X\_Y\_Z\_R\_  
                      G69

### Example Code:

```
N4 G17 G69 X1 Y1 R90
```

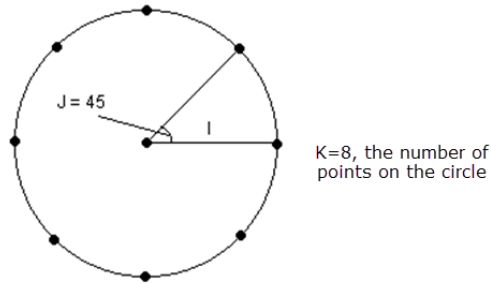
## G70 Bolt Hole Circle Pattern

When commanded, the tool will first drill a center hole, and then drill holes located at points equally distributed on the circle. This G-code must be preceded by a valid canned drilling cycle (i.e., G73 ~ G88). The canned cycle G-code must precede **G70** to establish the method of drilling for the pattern cycle. The X\_ and Y\_ parameters specified on the line containing the G73 ~ G88 determine where the center of the pattern will reside. The drilling canned cycle cannot reside on the same line as the drilling pattern cycle, **G70**.

**Syntax:** G70 I\_ J\_ K\_  
          I:        Radius of circle must be greater than 0.  
          J:        Angle formed by X-axis and vector from center of circle to start point.  
          K:        Number of points in the circle.

### Programming Example:

```
G83 X_ Y_ Z_ R_ L_  
G70 I3 J45 K8  
G80  
G84 X_ Y_ Z_ R_ L_ F_ P_ Q_  
G70 I3 J45 K8  
G80
```



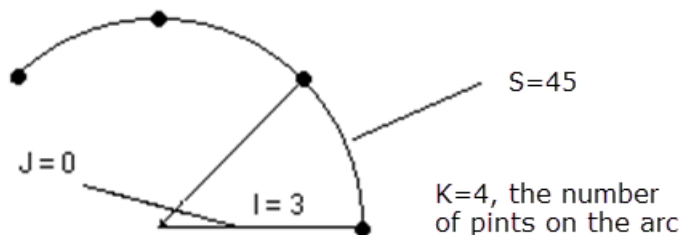
## G71 Arc Pattern

When commanded, the tool is located at points distributed equally on an arc. This G-Code must be preceded by a valid canned cycle (i.e. G81, G82, G83, G84, G85, G86, G87, G88). The canned cycle G-code must precede **G71** to establish the method of drilling for the pattern cycle. The X\_ and Y\_ parameters specified on the line containing G81- G88 determine where the center of the pattern will reside. The canned cycle G81 - G88 cannot reside on the same line as the pattern cycle **G71**.

**Syntax:** G71 I\_ J\_ K\_ S\_  
 I: Radius of arc must be greater than 0.  
 J: Angle formed by X-axis and vector from center of arc to start point.  
 K: Number of points in the arc  
 S: Angle between points on the arc

### Programming Example:

```
G83 X_ Y_ Z_ R_ L_  
G71 I3 J0 K8 S45  
G80
```



## G72 Bolt Line Pattern

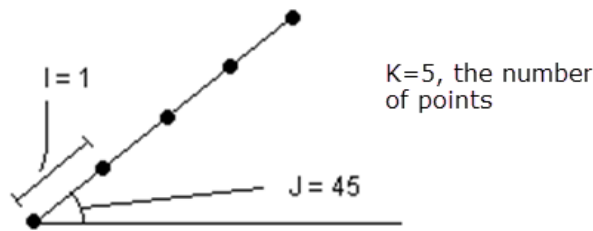
When commanded the tool is located at points distributed equally on a line. This G-Code must be preceded by a valid canned cycle. The canned cycle G-code must precede **G72** so as to establish the method of drilling for the pattern cycle. The X\_ and Y\_ parameters specified on the line containing G73-G88 determine where the start of the pattern will reside. The canned cycle G73-G88 cannot reside on the same line as the pattern cycle **G72**.

**Syntax:** G72 I\_ J\_ L\_  
 I: Distance between drill points, must be greater than 0.  
 J: Angle formed by X-axis and vector of line.  
 K: Number of points on the line.

### Programming Example:

```
G83 X_ Y_ Z_ R_ L_  
G72 I1 J45 K5  
G80
```

```
G84 X_ Y_ Z_ R_ L_ F_ P_ Q_
G72 I1 J45 K5
G80
```

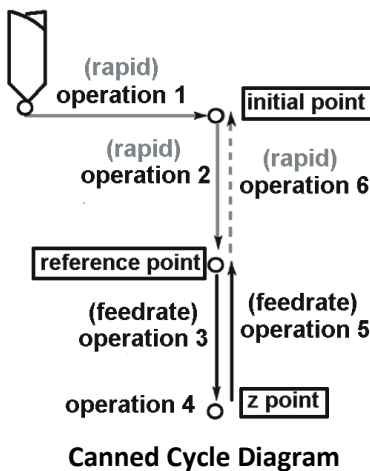


## G73-89 Canned Cycles

A canned cycle simplifies programming through the use of single G codes to specify machine operations normally requiring several blocks of NC code. The canned cycle consists of a sequence of five operations as shown here:

1. Position of axes
2. Rapid to initial point
3. Hole body machining
4. Hole bottom operations
5. Retract to reference point

A canned cycle has a positioning plane and a drilling axis. The positioning plane is the **G17** plane. The Z-axis is used as the drilling axis. Whether the tool is to be returned to the reference point or to the initial point is specified according to **G98** or **G99**. Use **G99** for the first drilling and **G98** for the last drilling. When the canned cycle is to be repeated by **L** in **G98** mode, the tool is returned to the initial level from the first time drilling. In the **G99** mode, the initial level does not change even when drilling is performed.



The drilling data can be specified following the **G** and a single block can be formed. This command permits the data to be stored in the control unit as a modal value. The machining data in a canned cycle is specified as shown below.

## G73 High Speed Peck Drilling Cycle-Short Retract

G73 is a high speed peck drilling cycle with a short retract specified by the D argument. Drilling commences from the specified R plane but only retracts by the retract amount until the hole depth is reached, which at that point retracts to the R plane. This cycle will occur on every line, which includes an X and Y move, until the mode is canceled with **G80** canned cycle cancel. As new arguments are passed during the canned cycle mode they will stay persistent for subsequent lines until the canned cycle mode is canceled or a new argument is passed.

**Syntax:** G73 X\_Y\_Z\_R\_Q\_S\_F\_L\_P\_D\_  
 X: Center location of hole along X  
 Y: Center location of hole along Y  
 Z: Depth to drill to  
 R: Clearance plane in Z  
 Q: Peck Depth  
 S: Spindle RPM  
 F: Cutting feedrate  
 L: Number of repeats  
 P: Dwell at bottom  
 D: Retract Distance

**Programming Example:**

```
G99 G73 X-3. Y-2.75 Z-1. R0.1 Q0.0125 S1500 F25. P0.1
X-2.75
X-2.5 P0.5
X-2.25
G80
```

## G74 Tapping Cycle – Float, Reverse

G74 is a float tapping cycle used for creating reverse or left hand threads. This cycle should be used with the appropriate float tap tool holder. When this cycle is commanded, the tool is located to the specified X and Y at rapid traverse rate, followed by a rapid traverse to the R-value. Linear movement is then performed at the programmed feedrate to the specified Z position. The spindle direction is then reversed and Z is fed linearly to the R-value. The return point in Z is the value of Z when the canned cycle is called if G98 mode is active. Otherwise, the return point in Z is the value of R specified on the G84 line if G99 mode is active. This cycle occurs on every line that includes an X and Y move until the mode is canceled with G80 canned cycle cancel. During this cycle, manual feedrate override is ignored. As new arguments are passed during the canned cycle mode they will stay persistent for subsequent lines until the canned cycle mode is canceled or a new argument is passed. The spindle RPM and Feedrate must be commanded appropriately in order to correctly use G74. The following formula must be followed below:

$$Feedrate = \frac{Spindle\ RPM}{Tap\ Threads\ per\ Inch}$$

**Syntax:** G74 X\_Y\_Z\_R\_F\_S\_  
 X: Center location of hole along X  
 Y: Center location of hole along Y  
 Z: Final depth for tapping  
 R: Clearance plane in Z  
 F: Cutting feedrate IPM (RPM \* (1 / number of threads per inch))  
 S: Spindle RPM

**Programming Example:**

```
G99 G74 X-4.3 Y3.21 Z-0.825 R0.1 S250 F8.928
X-4.8
X-5.3
G80
```

## G76 Fine Boring Cycle

The G76 Fine Boring Cycle uses a typical canned cycle motion but includes a spindle stop, orientation, and axis shift at the bottom of the hole in order to clear the sidewall on retract. This cycle will occur on every line, which includes an X

or Y move, until the mode is canceled with **G80** canned cycle cancel. As new arguments are passed during the canned cycle mode they will stay persistent for subsequent lines until the canned cycle mode is canceled or a new argument is passed.

**Syntax:** G76 X\_ Y\_ Z\_ R\_ Q\_ F\_ S\_  
X: Center location of hole along X  
Y: Center location of hole along Y  
Z: Final depth for boring  
R: Clearance plane in Z  
Q: Bore Shift amount in X  
F: Cutting feedrate  
S: Spindle RPM

**Programming Example:**

```
G99 G76 X-4.3 Y3.21 Z-0.825 R0.1 Q0.02 S1750 F7.5  
X-4.8  
X-5.3  
G80
```

## G80 Canned Cycle Cancel

G80 cancels any active canned cycles.

**Syntax:** G80

## G81 Drilling Cycle

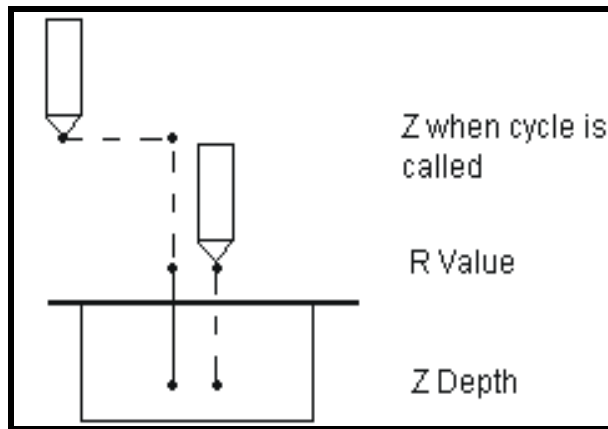
When this cycle is commanded, the tool is located to the specified X, Y at rapid traverse rate, followed by a rapid traverse to the R-value. Normal drilling is then performed at the specified feedrate to the specified Z position. The tool is then immediately retracted from the bottom of the hole at rapid traverse rate. The return point in Z is either the value of Z when the canned cycle is called if **G98** mode is active. Otherwise, the return point in Z is the value of **R** specified on the **G81** line if **G99** mode is active. This cycle will occur on every line, which includes an X or Y move, until the mode is canceled with **G80** canned cycle cancel. As new arguments are passed during the canned cycle mode they will stay persistent for subsequent lines until the canned cycle mode is canceled or a new argument is passed.

**Syntax:** G81 X\_ Y\_ Z\_ R\_ S\_ F\_ L\_  
X: Center location of hole along X  
Y: Center location of hole along Y  
Z: Depth to drill to  
R: Clearance plane in Z  
S: Spindle RPM  
F: Cutting feedrate  
L: Number of repeats

**Programming Example:**

```
G99 G81 X-3. Y-2.75 Z-0.05 R0.1 F250 L2  
X-2.75  
X-2.5 L4  
X-2.25  
G80
```

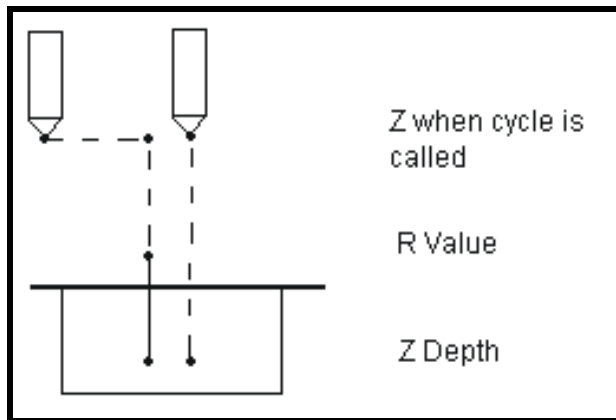




**Drilling Cycle with G99 Active**

**Programming Example:**

```
G98 G81 X-3. Y-2.75 Z-0.05 R0.1 F25.0 L2
X-2.75
X-2.5 L4
X-2.25
G80
```



**Drilling Cycle with G98 Active**

## G83 Deep Hole (Peck) Drilling Cycle

When this cycle is commanded, the tool is located to the specified X, Y at rapid traverse rate, followed by a rapid traverse to the R-value. Normal drilling is then performed at the specified feedrate to a depth of Q below the R-value. The tool is then retracted from the bottom of the hole at rapid traverse rate to the R-value.

The tool is then moved at rapid traverse rate to the height of the last drilling plus the **R** parameter. Normal drilling is then repeated to a depth of **Q** below the last hole. The tool is then once again retracted from the bottom of the hole at rapid traverse rate to the R-value.

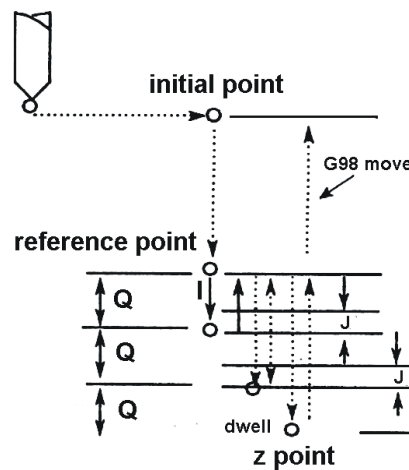
This pattern is repeated until the depth of the **Z** parameter is achieved. This cycle permits intermittent cutting feed to the bottom of the hole, to assist in removing chips from the hole.

The return point in Z is the value of **Z** when the canned cycle is called if **G98** mode is active. Otherwise, the return point in Z is the value of **R** specified on the **G83** line if **G99** mode is active. This cycle occurs on every line that includes an X or Y move until the mode is canceled with **G80** canned cycle cancel.

**Syntax:** G83 X\_Y\_Z\_R\_Q\_F\_L\_D\_P\_S\_  
X: Center location of hole along X  
Y: Center location of hole along Y  
Z: Depth to drill to  
R: Reference plane in Z  
Q: Peck depth  
F: Cutting feedrate  
L: Number of repeats  
D: Retract Distance  
P: Dwell at bottom of hole  
S: Spindle RPM

#### Programming Examples:

G98 G83 X-2 Y-1 Z-0.600 R0.1 Q0.150 D0.010 F25  
G80



**Deep Hole (Peck) Drilling Cycle Example**

### G84 Tapping Cycle – Float, Standard

G84 is a float tapping cycle used for creating standard or right hand threads. This cycle should be used with the appropriate float tap tool holder. When this cycle is commanded, the tool is located to the specified X and Y at rapid traverse rate, followed by a rapid traverse to the R-value. Linear movement is then performed at the programmed feedrate to the specified Z position. The spindle direction is then reversed and Z is fed linearly to the R-value. The return point in Z is the value of Z when the canned cycle is called if G98 mode is active. Otherwise, the return point in Z is the value of R specified on the G84 line if G99 mode is active. This cycle occurs on every line that includes an X and Y move until the mode is canceled with G80 canned cycle cancel. During this cycle, manual feedrate override is ignored. As new arguments are passed during the canned cycle mode they will stay persistent for subsequent lines until the canned cycle mode is canceled or a new argument is passed. The spindle RPM and Feedrate must be commanded appropriately in order to correctly use G74. The following formula must be followed below:

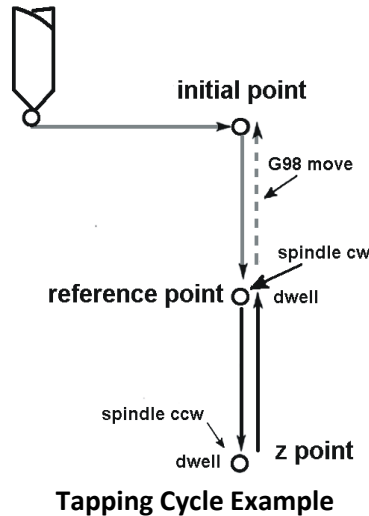
$$\text{Feedrate} = \frac{\text{Spindle RPM}}{\text{Tap Threads per Inch}}$$

**Syntax:** G84 X\_Y\_Z\_R\_F\_S\_  
X: Center location of hole along X  
Y: Center location of hole along Y  
Z: Final depth for tapping

R: Clearance plane in Z  
F: Cutting feedrate  
S: Spindle RPM

#### Programming Example:

```
G99 G84 X-4.3 Y3.21 Z-0.825 R0.1 S250 F8.928  
X-4.8  
X-5.3  
G80
```



## G84.2 Tapping Cycle – Rigid, Standard

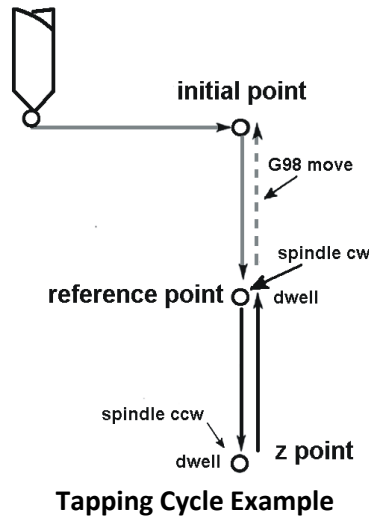
G84.2 is a rigid tapping cycle used for creating standard or right hand threads. This cycle should only be used with machines equipped with a spindle capable of tight closed loop control. When this cycle is commanded, the tool is located to the specified X and Y at rapid traverse rate, followed by a rapid traverse to the R-value. Coordinated linear movement between X and Z is then performed at the programmed feedrate to the specified Z position. The spindle direction is then reversed and Z is fed linearly to the R-value. The return point in Z is the value of Z when the canned cycle is called if **G98** mode is active. Otherwise, the return point in Z is the value of R specified on the **G84** line if **G99** mode is active. This cycle occurs on every line that includes an X and Y move until the mode is canceled with **G80** canned cycle cancel. During this cycle, manual feedrate override is ignored. As new arguments are passed during the canned cycle mode they will stay persistent for subsequent lines until the canned cycle mode is canceled or a new argument is passed.

**Syntax:** G84 X\_ Y\_ Z\_ R\_ F\_ S\_  
X: Center location of hole along X  
Y: Center location of hole along Y  
Z: Final depth for tapping  
R: Clearance plane in Z  
F: Cutting feedrate  
L: Number of Repeats  
P: Dwell at bottom of tapped hole

#### Programming Example:

```
G99 G84.2 X-4.3 Y3.21 Z-0.825 R0.1 S600 F30.0
```

X-4.8  
X-5.3  
G80



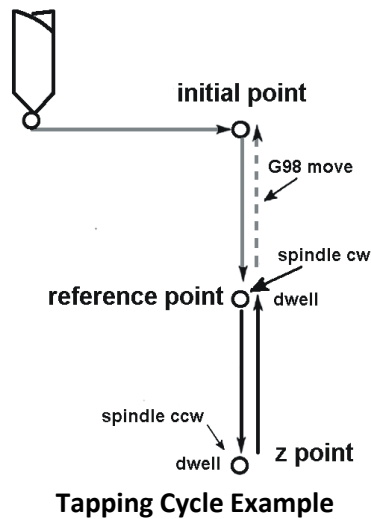
### G84.3 Tapping Cycle – Rigid, Reverse

G84.2 is a rigid tapping cycle used for creating reverse or left hand threads. This cycle should only be used with machines equipped with a spindle capable of tight closed loop control. When this cycle is commanded, the tool is located to the specified X and Y at rapid traverse rate, followed by a rapid traverse to the R-value. Coordinated linear movement between X and Z is then performed at the programmed feedrate to the specified Z position. The spindle direction is then reversed and Z is fed linearly to the R-value. The return point in Z is the value of Z when the canned cycle is called if **G98** mode is active. Otherwise, the return point in Z is the value of **R** specified on the **G84** line if **G99** mode is active. This cycle occurs on every line that includes an X and Y move until the mode is canceled with **G80** canned cycle cancel. During this cycle, manual feedrate override is ignored. As new arguments are passed during the canned cycle mode they will stay persistent for subsequent lines until the canned cycle mode is canceled or a new argument is passed.

**Syntax:** G84 X\_ Y\_ Z\_ R\_ F\_ S\_  
X: Center location of hole along X  
Y: Center location of hole along Y  
Z: Final depth for tapping  
R: Clearance plane in Z  
F: Cutting feedrate  
L: Number of Repeats  
P: Dwell at bottom of tapped hole

#### Programming Example:

```
G99 G84.3 X-4.3 Y3.21 Z-0.825 R0.1 S600 F30.0  
X-4.8  
X-5.3  
G80
```



## G85 Reaming, Boring Cycle

When this cycle is commanded, the tool is located to the specified X, Y at rapid traverse rate, followed by a rapid traverse to the R-value. Linear movement is then performed at the programmed feedrate to the specified Z position. Z is then fed linearly to the R-value. The return point in Z is the value of **Z** when the canned cycle is called if **G98** mode is active. Otherwise, the return point in Z is the value of **R** specified on the **G85** line if **G99** mode is active. This cycle occurs on every line that includes an X and Y move until the mode is canceled with **G80** canned cycle cancel. During this cycle, manual feedrate override is ignored. As new arguments are passed during the canned cycle mode they will stay persistent for subsequent lines until the canned cycle mode is canceled or a new argument is passed.

**Syntax:**        G85 X\_ Y\_ Z\_ R\_ F\_ L\_

X:    Center location of hole along X

Y:    Center location of hole along Y

Z:    Depth to drill to

R:    Reference plane in Z

F:    Cutting feedrate

L:    Number of repeats

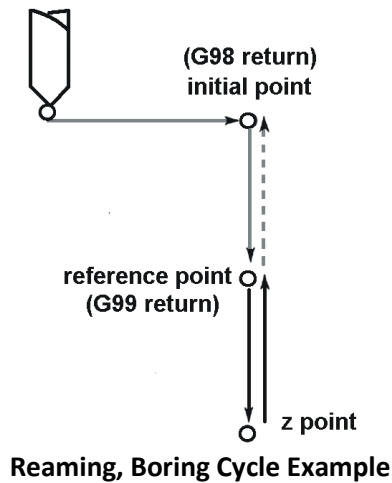
S:    Spindle RPM

### Programming Examples:

```
G99 G85 X-3.1 Y-2.75 Z-0.50 R0.1 F25.0
X-2.75
X-2.5
G80
```

## G86 Boring Cycle (Finishing cut)

When G86 is commanded, the tool is located to the specified X Y at rapid traverse rate, followed by a rapid traverse to the R-value. Linear movement is then performed at the programmed feedrate to the specified Z position. At this point, the spindle is stopped and a dwell of **P** seconds will occur. **Z** is then fed rapidly to the R-value. The return point in Z is either the value of **Z** when the canned cycle is called if **G98** mode is active. Otherwise, the return point in Z is the value of **R** specified on the **G85** line if **G99** mode is active. This cycle occurs on every line that includes an X and Y move until the mode is canceled with **G80** canned cycle cancel. During this cycle, manual feedrate override is ignored. As new arguments are passed during the canned cycle mode they will stay persistent for subsequent lines until the canned cycle mode is canceled or a new argument is passed.



**Syntax:** G86 X\_ Y\_ Z\_ R\_ F\_ P\_ L\_

X: Center location of hole along X

Y: Center location of hole along Y

Z: Depth to drill to

R: Reference plane in Z

F: Cutting feedrate

P: Dwell in seconds at the bottom of the cut

L: Number of repeats

#### Programming Examples:

```
G98
G86 X-3.1 Y-2.75 Z-0.50 P0.5 R0.1 F25.0
X-2.75
X-2.5
G80
```

### G87 Back Boring Cycle

When this cycle is commanded, the tool is located to the specified X Y at rapid traverse rate, the spindle is stopped and subsequently oriented, the X axis is shifted out by the Q value followed by a rapid traverse to the R-value. Then the X axis will shift in by the Q value and spindle rotation will commence followed by linear movement at the programmed feedrate to the specified Z position. At this point the spindle is stopped and oriented. The X axis will again shift out by the Q value and the Z-axis is returned to the R value in rapid mode where the X axis will shift in by the Q value and the spindle will restart.

**Syntax:** G87 X\_ Y\_ Z\_ R\_ F\_ L\_ Q\_ S\_

X: Center location of hole along X

Y: Center location of hole along Y

Z: Depth to drill to

R: Reference plane in Z

F: Cutting feedrate

L: Number of repeats

Q: Bore shift amount in X axis

S: Spindle RPM

#### Programming Examples:

```
G99
```

```
G87 X-3.2 Y-2.75 Z-1.005 R0.1 F25.0
X-2.75
X-2.5
G80
```

## G89 Boring Cycle (Finishing Cut, Free Cutting)

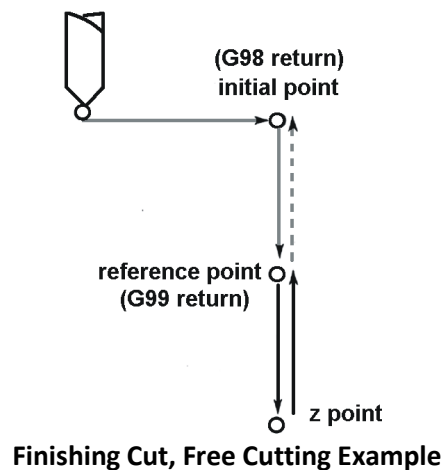
When this cycle is commanded, the tool is located to the specified X Y at rapid traverse rate, followed by a rapid traverse to the R-value. Linear movement is then performed at the programmed feedrate to the specified Z position. At this point a dwell of **P** seconds is performed. **Z** is then fed linearly to the R-value. The return point in Z is the value of **Z** when the canned cycle is called if **G98** mode is active. Otherwise, the return point in Z is the value of **R** specified on the **G89** line if **G99** mode is active. This cycle occurs on every line that includes an X and Y move until the mode is canceled with **G80** canned cycle cancel. During this cycle, manual feedrate override is ignored. As new arguments are passed during the canned cycle mode they will stay persistent for subsequent lines until the canned cycle mode is canceled or a new argument is passed.

**Syntax:** G89 X\_ Y\_ Z\_ R\_ F\_ P\_ L\_ S\_

- X: Center location of hole along X
- Y: Center location of hole along Y
- Z: Depth to drill to
- R: Reference plane in Z
- F: Cutting feedrate
- P: Dwell in seconds at the bottom of the cut
- L: Number of repeats
- S: Spindle RPM

### Programming Examples:

```
G99
G89 X-3.3 Y-2.75 Z-2.005 P.5 R0.1 F25.0
X-2.75
X-2.5
G80
```



## G90 Absolute/G91 Incremental Mode

Movement of the axes may be programmed either in absolute or incremental commands. The absolute mode is selected automatically when the power is turned on. In the absolute mode (**G90**), all axis word dimensions are referenced from a

single program zero point. The algebraic signs (+ or -) of absolute coordinates denote the position of the axis relative to program zero.

In the incremental mode (**G91**), the axis word dimensions are referenced from the current position. The input dimensions are the distance to be moved. The algebraic sign (+ or -) specifies the direction of travel.

**Syntax:**           G90 (Absolute mode)  
                  G91 (Incremental mode)

**Example Code:**

```
N020 G20 G90 G0 X0 Y0
N025 G43 Z0.25 H1
N030 X1.125 Y2.25
```

## G90.1 Absolute Arc Radius/G91.1 Incremental Arc Radius

The vector to the center point of a G02 or G03 arc move may be programmed in either incremental mode (from the starting point of the arc) or absolute mode (from the programming origin). Incremental mode is the default.

**Syntax:**           G90.1 (Absolute mode)  
                  G91.1 (Incremental mode)

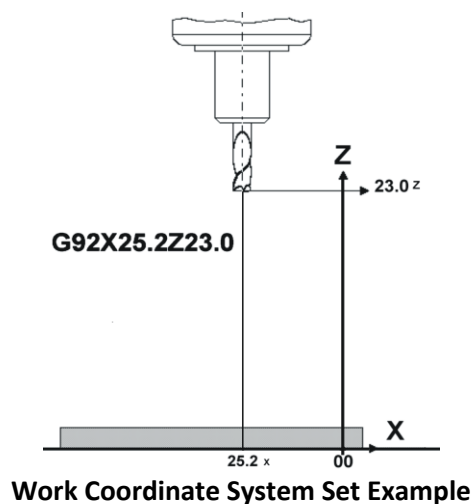
## G92 Work Coordinate System Set

This command establishes the work coordinate system so that a certain point of the tool (e.g. tool tip) becomes IP in the established work coordinate system. Any subsequent absolute commands use the position in this work coordinate system. Meet the programming start point with the tool tip and command **G92** at the start of program (**G92X25.2Z23.0**). When creating a new work coordinate system with the **G92** command, a certain point of the tool becomes a certain coordinate value; therefore, the new work coordinate system can be determined irrespective of the old work coordinate system. If the **G92** command issued to determine a start point for machining based on work pieces, a new coordinate system can be created even if there is an error in the old work coordinate system. If the relative relationships among the **G54** to **G59** work coordinate systems are correctly set at the beginning, all work coordinate systems become new coordinate systems as desired.

**Syntax:**           G92 X\_\_Y\_\_Z\_\_

**Example Code:**

```
N4 G53 X0 Y0 Z0
N5 G92 X0 Y1.0156
N6 Z.1 H1 M8
```





## G93 Inverse Time Feed

**G93** specifies inverse time mode: move is specified by 1 / **F** word minutes. In inverse time feed mode, an **F** word is interpreted to mean the move should be completed in [one divided by the **F** word] minutes. For example, if the **F** word is 2.0 the move should be completed in half a minute (thirty seconds). Customarily, this code is used in rotary or 5-axis programming.

- a. Solve for move time of 5 sec.

$$F = 60 \text{ sec} \div T_{\text{sec}}$$

$$F = 12$$

- b. Recode the block

G01 G93 A30 F12

**Syntax:** G93 F\_

## G94 Feed Per Minute

The **G94** preparatory function code specifies the feed rate in terms of vector per unit time. The **G94** preparatory function is modal and remains in effect until replaced. The mode is set to **G94** by power on, data reset and the **M30** code.

**Syntax:** G94

## G98/G99 Canned Cycle Return Point

Used in a canned cycle block to determine the return point. **G98**: Initial point. **G99**: clearance plane or reference point.

**G98** causes the tool to return to the point from which it was first called. **G99** causes the tool to return to the point specified by the **R** address.

**Syntax:** G98/G99

**Example Code:**

```
N4 X0Y0
N5 G98
N6 G81X1 Y1R0.1Z-3
.
.
.
N4 Z5
N5 G99
N6 G81X1Y1R0.1Z-3
```

## M Code Library – CNC M-Codes

Program Group	M00	Program Stop
	M01	Option Stop
	M02	End of Program w/Rewind
	M30	End of Program w/Rewind

Spindle Group	M03	Spindle CW
	M04	Spindle CCW
	M05	Spindle Stop
	M19	Spindle Orient

Tool Group	M06	Tool Change Request
------------	-----	---------------------

Coolant Group	M07	Coolant Mist ON
	M08	Coolant Flood ON
	M09	Coolant OFF

Sub-Program Group	M98	Sub-Program Call
	M99	Return From Sub-Program

### M00 Program Stop

Unconditional stop of part program at current block. Machine state does not change until restart or rewind.

**Example:**

```
X-1.25  
X-1.  
M00  
G01 X5.2 Y3.2 F21
```

### M01 Optional Stop

Same as M00, but conditional on Optional stop switch setting.

**Example:**

```
X-1.25  
X-1.  
M01  
G01 X5.2 Y3.2 F21
```

### M02 End of Program with Rewind

This resets the program buffer to the beginning of the program. In early generation CNC controls there was a differentiation between M30 and M02. In modern controls these commands are identical as no actual rewind or a tape is necessary.

**Example:**

```
. . .  
G0 G49 X0 Y0 Z0  
Z.5 M5 M9  
G90 G0 G49 M5 M9  
X0 Y0 Z0  
M02
```

## M03 Spindle Clockwise

Starts the spindle in the clock-wise direction (CW) using the current setting for speed.

### Example:

```
N30 G54 G0 X-3.7185 Y-.1649
N40 S5000 M03 T1
N50 G43 H1 Z.1
. . .
```

## M04 Spindle Counterclockwise

Starts the spindle in the counter-clockwise direction (CCW) using the current setting for speed.

## M05 Spindle Stop

Stops the spindle.

### Example:

```
G28 X0. Z0.
M05
G04 X2.
M02
```

## M06 Tool Change

M06 initiates a tool change request to the currently presiding T-code. The tool change sequence is machine dependent and will be unique for every machine.

### Example:

```
G0 G49 X0 Y0
T03 M06
M03 S100
M08
G00 X1.5 Y-1.5
```

## M07 Coolant Mist On

Engages the flood coolant pump.

### Example:

```
G43 Z0.5 H10
M07
```

## M08 Coolant Flood On

Engages the flood coolant pump.

### Example:

```
G43 Z0.5 H10
M08
```

## M09 Coolant Off

Disengages the both the flood and mist coolant pumps.

### Example:

```
X-4.1657 Y-5.4552
G02 X-4.2073 Y-5.4421 I-0.0056 J0.0547
G00 Z0.5 M05 M09
```

## M19 Spindle Orient

Initiates a spindle orientation.

### Example:

```
X-4.1657Y-5.4552
```

### M19

```
Z-2.01
```

## M30 End of Program with Rewind

This resets the program buffer to the beginning of the program. In early generation CNC controls there was a differentiation between M02 and M30. In modern controls these commands are identical as no actual rewind or a tape is necessary.

### Example:

```
. . .  
Z.5 M5 M9  
G90 G0 G49 M5 M9  
X0 Y0 Z0  
M30
```

## M98 Sub-Program Call

**Syntax :**        **M98** [P\_\_\_\_] [L\_\_\_\_]  
                  **M98** (C:\...) [L\_\_\_\_]

Where:

- L\_\_\_\_ - specifies the number of times to execute the program
- P\_\_\_\_ - specifies a program name of the form P\_\_\_\_.nc
- (C:\...) - specifies a program name as a path and filename in a comment.

## Description

The M98 code calls a sub-program from an executing program. When an M98 command is encountered in a currently executing program (the calling program), control is transferred to the specified program in the M98 block (the called program). The program control is transferred back to the calling program using an M99 at the end of the sub-program. The number of levels which can be called depends on the stack-offset which can be set in the .ini file. In the default configuration with a stack-offset of 256, Sub-Programs can be nested 30 deep. If the default stack-offset is reduced the maximum number of Sub-programs calls is 127.

There are two ways of specifying a called program in an M98 block:

1. P Code specification
2. Comment specification

## P Code Specification

The most used and standard method is by referencing a program number with a Pxxxx address code. The Pxxxx code must be within either the volatile (100-199 by default) or non-volatile (0-99 by default) program number range specified in the PowerPmacNC.ini setup file.

**Typical Syntax:** M98 P0102

### Example:

```
G90 G54 G00 X1.0 Y0 S2000 M03 T03  
G43 Z1.0 H02 M08  
G01 Z0.02 F30.0  
M98 P0101 L2
```

The actual filename of the program which will be called will be of the form Oxxxx.nc and should be formatted as such on the hard disk. The control will attempt to look for sub-programs in the sub-programs folder defined in the PowerPmacNC.ini file. The default sub-program folder is "C:\NC".

**Typical Path and Filename:** C:\NC\O0102.nc

### Comment Specification

Another way to specify a program is by specifying a full path and filename in a comment that is on the M98 line. This is a way to transfer control to routines that are not in the specified sub-program directory. All that is necessary is to place a valid file path name in the comment. If the path or filename is invalid, then an alarm is issued.

#### Typical Example:

```
G90
G01 F50 X10 Y10
M98 (C:\SubLater\BabyDogRoughCycle.nc)
X0 Y0
M30
```

Note: It is critical to note the intended program number must be present within the header of the sub-program being called or the control will issue an error. The example above references the BabyDogRoughCycle.nc file, but in order to accommodate the controls requirements we must include a program number reference within the defined volatile program number range as shown in the example below:

```
%
O0102 (Intended Sub-Program Number)
N100 G20
N102 G0 G17 G40 G49 G80 G90
N104 T249 M6
N106 G0 G90 G54 X3.5483 Y-0.1 A0.0 S4278 M3
N108 G43 H249 Z.369
N110 G1 Z.1699 F250
```

### Volatile vs. Non-Volatile Sub-Programs

The Power Pmac NC16 software supports two different kinds of sub-programs, volatile and non-volatile. Volatile sub-programs are called spontaneously from the part program and are not saved at the control. Non-volatile sub-programs are special sub-programs included by the machine builder and reside at the control level (i.e. do not disappear through a power cycle). Non-Volatile sub-programs must be carefully configured and downloaded. Refer to the Power PMAC software manual for detailed instructions on how to do this.

### M99 Return from Subroutine / Program Repeat

**M99** transfers program control back to a calling program if it resides in the sub-program. If M99 resides in the main program it will serve as a looping function if an L command is present. The action of **M99** is different depending on whether **M99** is encountered in a subroutine or in the main program.

**Syntax Sub-Program:** M99

**Syntax Main Program:** M99 Lxx

## T-Codes

### T-Code Format

**Tnn** Where **nn** specifies tool number from the Tools page in the NC display.

**Example:**

```
(TOOL 4 = .437 DRILL)
(TOOL 3 = 1/2-13 TAP)
G90 G80 G49 G40 G20 G17 G56
T04 M06
M03 S3000
M08
G00 X1.5 Y-1.5
```

## Miscellaneous

### Block Delete Character: /

Prevents execution of the block when Block Delete is on. Must be the first character in the block.