



**Haas Factory Outlet**

**A Division of Productivity Inc.**

***HFO MN***  
***Haas Mill Series***  
***Training Manual***  
***Haas G&M Code Programming***



Rev 8/2018



This Manual is the Property of Productivity Inc.  
It may not be reproduced or disseminated without the express written permission  
of  
Productivity Inc. The content must not be altered, nor have the Productivity name  
removed from the materials. This training manual is a guide for the operation of  
the Machine Tool.

The Operator is responsible for following Safety Procedures as outlined by their  
Instructor or the Manufacturers Specification. Downloading and/or other use of  
this manual does not certify the completion of the Training Course. This manual is  
for reference only.

For more information on Additional Training Opportunities or our Classroom  
Schedule,

Contact Productivity Inc.

☎ 763.476.8600 ☎ (800)328-3272 Toll Free

Visit us on the Web: [www.productivity.com](http://www.productivity.com)

To obtain permissions contact: ✉ [trainingmn@productivity.com](mailto:trainingmn@productivity.com)

Note: Some of the content, images and screen shots included in this manual are taken from  
Haas manuals, controllers and web information with permission from Haas Automation Inc.  
2800 Sturgis Road Oxnard CA 93030-8933

# ***G&M Code Mill Programming Training Manual***

## ***Table of Contents***

<b>INTRODUCTION .....</b>	<b>5</b>
<b>MACHINE HOME WITH WORK OFFSETS .....</b>	<b>7</b>
WORK COORDINATE SELECTION .....	8
<b>TOOL LENGTH COMPENSATION G43 .....</b>	<b>9</b>
<b>ABSOLUTE AND INCREMENTAL POSITIONING .....</b>	<b>10</b>
THE CARTESIAN COORDINATE SYSTEM .....	11
WORD ADDRESS PROGRAMMING .....	12
PROGRAMMING .....	14
<b>ALPHABET WORD ADDRESS ASSIGNMENTS .....</b>	<b>15</b>
<b>PREPARATORY FUNCTIONS (G CODES) .....</b>	<b>19</b>
<b>MACHINE FUNCTIONS (M CODES) .....</b>	<b>23</b>
<b>PROGRAM STRUCTURE AND FORMAT .....</b>	<b>27</b>
PROGRAM FORMAT .....	28
MACHINE DEFAULTS .....	29
PROGRAMMING WITH CODES .....	30
PROGRAM STRUCTURE .....	31
<b>LINEAR AND CIRCULAR TOOL PATHS .....</b>	<b>33</b>
LINEAR/CIRCULAR MOVEMENT – CREATING TOOL PATH .....	34
INTERPOLATION COMMANDS .....	35
CIRCULAR INTERPOLATION (G02 AND G03) COMMANDS .....	36
<b>CUTTER COMPENSATION (G41, G42) .....</b>	<b>43</b>
<b>FORMULAS – TAPPING, SPEEDS AND FEEDS .....</b>	<b>52</b>
<b>DRILLING, TAPPING, BORING CANNED CYCLES .....</b>	<b>53</b>
<b>CANNED CYCLES .....</b>	<b>54</b>
<b>LOOPING COMMAND CYCLES .....</b>	<b>72</b>
<b>BOLT HOLE PATTERNS .....</b>	<b>74</b>

<b>ADDITIONAL G CODES</b> .....	<b>80</b>
<b>MILLING CIRCLES WITH CUTTER COMP</b> .....	<b>81</b>
<b>THREAD MILLING</b> .....	<b>82</b>
<b>CIRCULAR POCKET MILLING USING G12 AND G13</b> .....	<b>83</b>
<b>CIRCULAR PLANE SELECTION</b> .....	<b>88</b>
INCH / METRIC SELECTION (G20, G21).....	89
<b>SETTING WORK, TOOL OFFSETS THROUGH THE PROGRAM (G10)</b> .....	<b>90</b>
<b>GENERAL PURPOSE POCKET MILLING (G150)</b> .....	<b>91</b>
<b>ENGRAVING (G47)</b> .....	<b>98</b>
<b>SUBROUTINES (SUBPROGRAMS)</b> .....	<b>103</b>
SUBROUTINES .....	104
<b>EXERCISES</b> .....	<b>106</b>
FINAL EXERCISES .....	110

# *Introduction*

Welcome to Productivity, Inc., your local Haas Factory Outlet (H.F.O.) for the Mill Programming Class. This class is intended to give a basic understanding of the programming of a Haas Machining Center.

After 1945 design of wings for the US Air Force were becoming extremely complex and hard to manufacture using conventional machine tools. MIT developed a machine that was able to control a cutting tool path with a series of straight lines defined by axial coordinates at prescribed feed rates. The first NC machine tool was introduced to the defense and aerospace industry by MIT in 1952. The contour of a constantly changing curvature could be described by a series of short lines determined by a series of coordinate in three axes.

The first machine tools were run with instructions or programs punched out on paper tape. The files of the early machine tools were often in the format which later became known as G-code. The reason for the name being that many of the lines of text began with the letter G.

In an NC machine, the tool is controlled by a code system that enables it to be operated with minimal supervision and with a great deal of repeatability. "CNC" (Computerized Numerical Control) is the same type of operating system, with the exception that a computer monitors the machine tool.

The same principles used in operating a manual machine are used in programming a NC or CNC Machine. The main difference is that instead of cranking handles to a position on a slide to a certain point, the dimension is stored in the memory of the machine control **once**. The control will then move the machine to these positions each time the program is run.

The operation of the VF-Series Vertical Machining Center requires that a part program be designed, written, and entered into the memory of the control. There are several options for getting these programs to the control. RS-232 (serial port with a computer), 3.5" Floppy Disk, Ethernet / Networking/ and USB are all viable ways to transmit and receive programs.

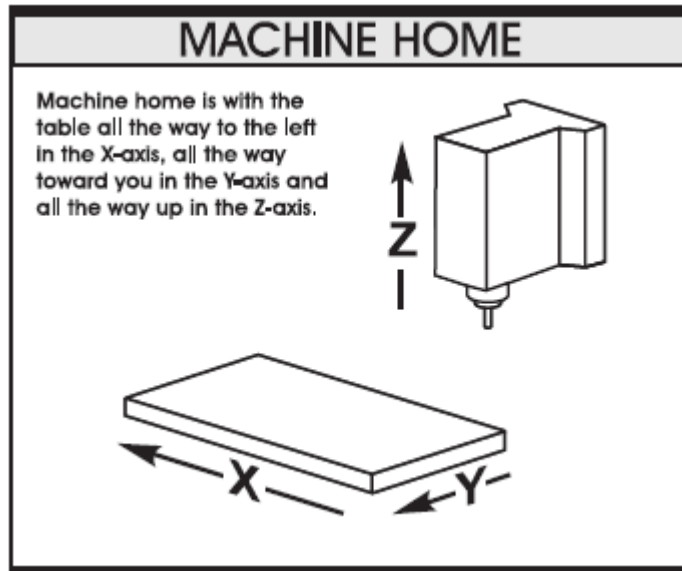
In order to operate and program a CNC controlled machine, a basic understanding of machining practices and a working knowledge of math are necessary. It is also important to become familiar with the control console and the placement of the keys, switches, displays, etc., that are pertinent to the operation of the machine.

This manual is intended to give a **basic** understanding of CNC programming and its applications. It is not intended as an in-depth study of all ranges of machine use, but as an overview of common and potential situations facing CNC programmers. Also use of the new Haas Control feature "Intuitive Programming System" or (IPS) will be demonstrated. It will produce G-Code programs for simple machine operations.

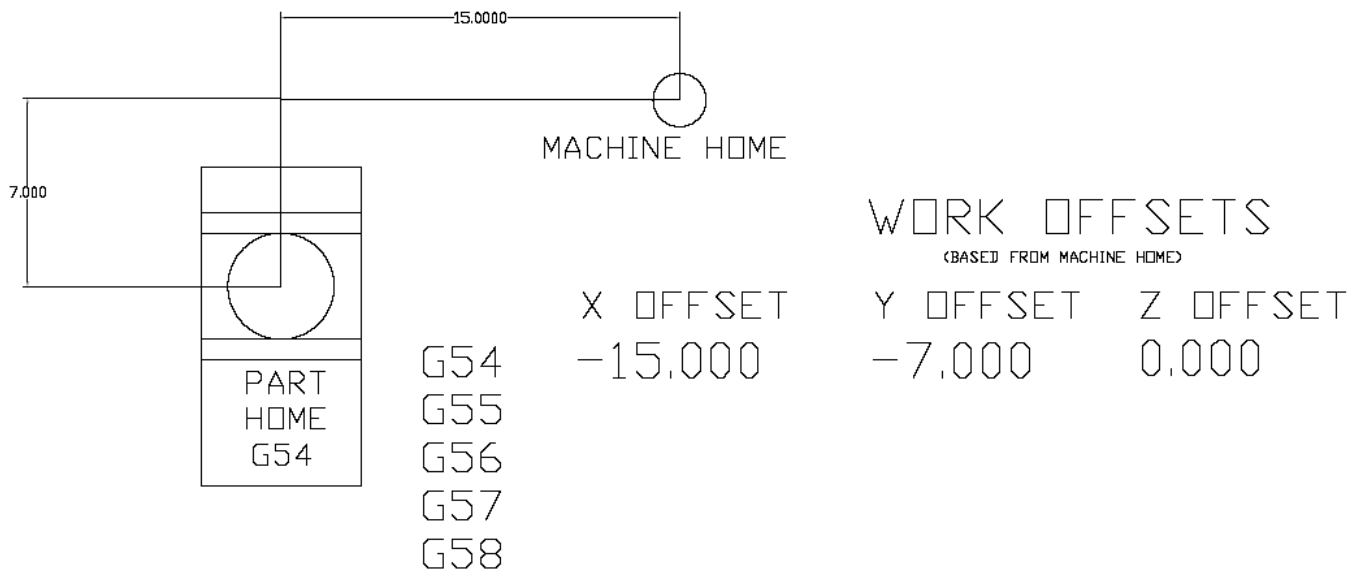


## Machine Home with Work Offsets

The principle of machine home may be seen when doing a reference return of all machine axes at machine start-up. A zero return (POWER UP/RESTART) is required when you power on machine, all three axes are moved to extreme positive locations until limit switches are reached. The reason the machine does this is to double check its position with the “Home” switches of the machine.



This is crucial to the operation and function of a CNC machine as all of our programs, fixturing, and tooling are based off of machine home.



Above: The relationship of machine home to “work home”, otherwise know as “work offset”

## ***Work Coordinate Selection***

### **What is a “Work Coordinate”?**

A work coordinate (otherwise known as a part offset) is how we tell the machine where our part (or parts) are located at in the travels of the machine. Under the Work Offsets page in the control, we hand wheel the machine to the X & Y “Zero” location for our part, and use the “Part Offset Measure” key under the Reset key to set the corresponding work offset from our program (G54, G55, G56, etc.....)

### **G54 – 59      Work Offsets #1 – 6**

These are the first G-Codes that were assigned to work Coordinates. This is how we tell the machine that we are working on Part #1, Part #2, etc.... thru Part #6. Originally no one thought we would need more than 6 part offsets, but thru time and the invention of new types of machines, more were needed.....

### **G110 – G129    Work Offsets #7 – 26 (Older Machines)**

### **G154 P1-P99    Work Offsets #7-106 (Newer Machines)**

These codes are the same as **G54** to **G59**; they add more places as X & Y zero. We now can set up to 99 additional “zeros” within the travels of our machine.

#### **MORE WORK COORDINATE SYSTEM SELECTION**

*Note: The G52 command works differently depending on the value of Setting 33. This setting selects the FANUC, HAAS, or YASNAC style of coordinates, which are listed below.*

### **G52 Global Work Coordinate Shift**

G52 will “shift” all work offsets that are set in the machine. In the Work Offsets page of the control, if we input a value of X +1.0000, ALL of the offsets will move one to the right by a value of 1.0000. This is most commonly used in casting and forging work where we have core movement.

### **G53 Positioning In Regards to Machine Home (Non Modal)**

G53 is used inside a program when we want to move the machine a certain distance and location from Machine Home. This is quite often used if we want to establish a safe tool change position because we have large parts or tools and need to clear the tool changer.

### **G92 Set Work Coordinate System**

G92 Can be used to set our work offsets while “on the fly” in our program. G92 was used back when machines only had one offset to choose from. We had to cut our first part, move the spindle over to the second part X&Y zero, and then call G92 X0Y0 in our program. Our work offset is now set around the second part. Using G54 – G129 is much faster, more tunable, and easier to use.

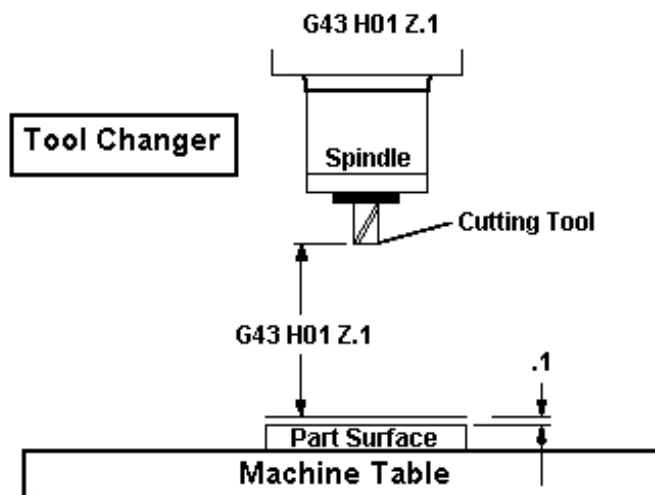


# Tool Length Compensation G43

## G43 Tool Length Compensation

G43 is the code we use to establish a tool length to the control. Upon setup, the operator will determine the tool length and input that dimension into the Tool Offset Memory for that tool. Each tool in the machine will have a defined tool length, and this will be presented to the control in the form of an "H" value. (H1 is equal to tool length offset #1, H2 = length offset #2, etc.....)

*The programmed code would be:*



## Canceling Tool Compensation (G49 or H00)

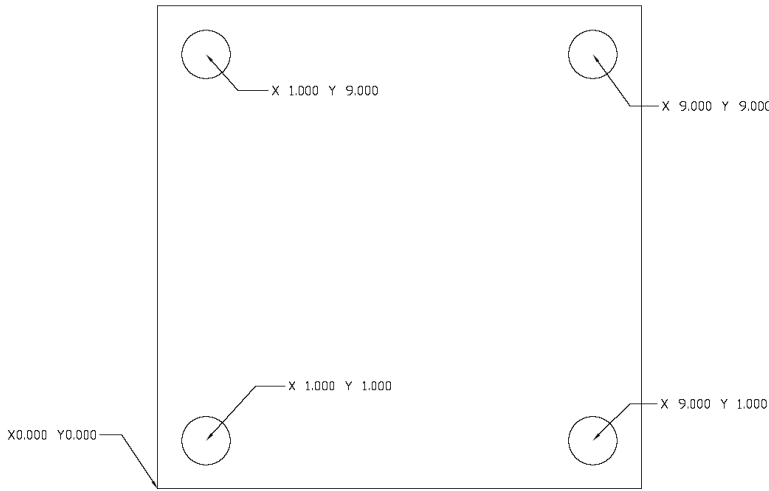
To cancel tool length compensation, we can either use the code of G49, G28 (Go to machine home position) or use an H value of H00. M30 (program end) or depressing the reset button will also cancel tool length compensation.

# Absolute and Incremental Positioning

There are two different systems used in positioning our machine. Both will “steer” the machine where we need it to go, both can net the same results. The reason we use more than one, is flexibility. Below we will talk about both, and they are the first two “G-Codes” that we are going to talk about.

## Absolute Positioning:

With absolute positioning, we tell the machine where to move based on a common point, called X0 Y0 and Z0. Every time we need to move to a certain position, the ending point of that move is in direct relationship to this “common point”



### G90 Absolute Positioning

Program to move the machine to these 4 hole locations when using G90 (Abs.)

```
X 1.0000 Y 1.0000  
X 9.0000 Y 1.0000  
X 9.0000 Y 9.0000  
X 1.0000 Y 9.0000
```

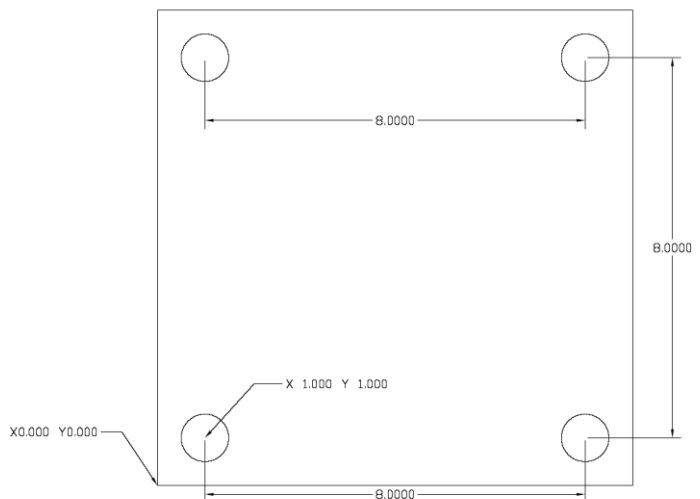
## Incremental Positioning:

With incremental positioning, we are telling the machine where to go in relationship to where it currently is at. Basically like a set of directions given from where the machine stopped last.

### G91 Incremental Positioning

Program to move the machine to the same 4 hole locations using G91 (Incr.)

```
X 1.0000 Y 1.0000  
X 8.0000  
Y 8.0000  
X -8.0000
```

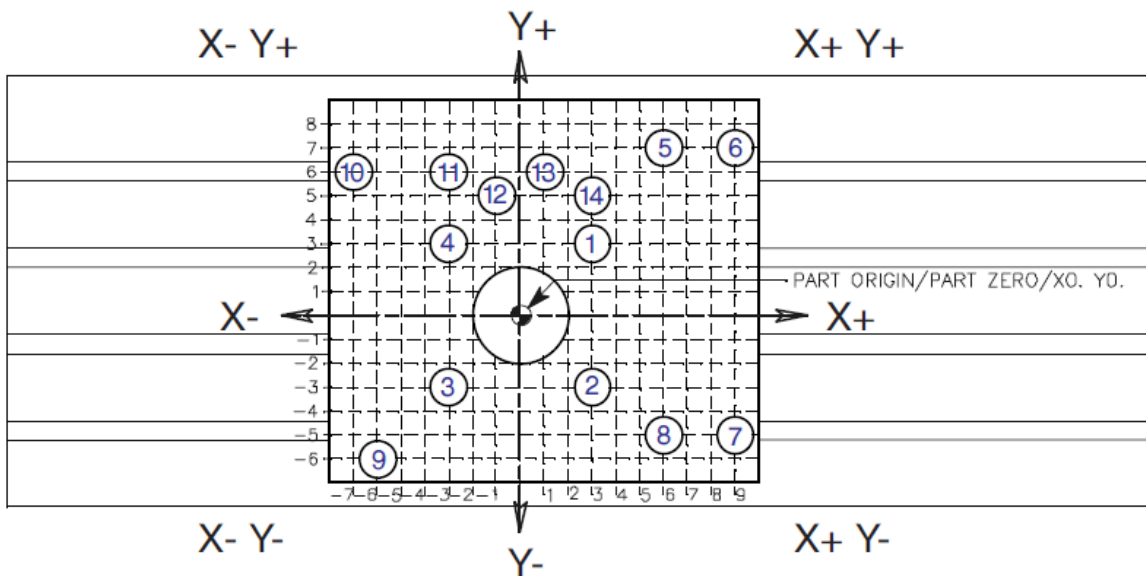


When do we decide which to use?

We switch between the two when it is more convenient. One example is look at the above 2 prints. Sometimes the print doesn't call out the hole-locations, but will give the distance between the holes.

# The Cartesian Coordinate System

## POSITIONING EXERCISE



What is the value in X and Y for each hole in absolute G90 positioning when each move is defined from a single fixed part zero point of an X0 Y0 origin point.

- PT1 = X \_\_\_\_\_ Y \_\_\_\_\_
- PT2 = X \_\_\_\_\_ Y \_\_\_\_\_
- PT3 = X \_\_\_\_\_ Y \_\_\_\_\_
- PT4 = X \_\_\_\_\_ Y \_\_\_\_\_
- PT5 = X \_\_\_\_\_ Y \_\_\_\_\_
- PT6 = X \_\_\_\_\_ Y \_\_\_\_\_
- PT7 = X \_\_\_\_\_ Y \_\_\_\_\_
- PT8 = X \_\_\_\_\_ Y \_\_\_\_\_

What is the value for each hole in INCREMENTAL G91 positioning when each move is defined from the previous position and the zero point shifts with the new position.

- From PT8 to PT9 = X \_\_\_\_\_ Y \_\_\_\_\_
- From PT9 to PT10 = X \_\_\_\_\_ Y \_\_\_\_\_
- From PT10 to PT11 = X \_\_\_\_\_ Y \_\_\_\_\_
- From PT11 to PT12 = X \_\_\_\_\_ Y \_\_\_\_\_
- From PT12 to PT13 = X \_\_\_\_\_ Y \_\_\_\_\_
- From PT13 to PT14 = X \_\_\_\_\_ Y \_\_\_\_\_



## ***Word Address Programming***

This unit will give a broad overview of word address programming; all alpha codes and their basic functions will be discussed.

### ***Objectives:***

Upon completion of this unit, the student will:

- 1) Understand the purpose or role of each alpha character involved in word address programming.
- 2) Understand the concept of Modal and Non-Modal commands.
- 3) Have a basic idea of the function of G and M codes.
- 4) Have an overall understanding of the basic theory of G and M code programming.

## ***Programming***

A CNC Mill program is defined as a set of instructions given to the machine control to move the positioning of the machine spindle, changes to the spindle RPM, and changes to the machine's other features (Tool Changes, Coolant System, Chip Control, etc.....)

Tool movements consist of rapid positioning commands, straight line movement of the tool at a controlled speed, and movement along an arc.

The machine has three (3) linear axes named **X**, **Y**, and **Z**. The **X**-axis moves the table left and right, the **Y** axis moves it to and from the operator, and the **Z** moves the milling head up and down. The machine zero position is where the tool is at the right corner of the mill table farthest away from the front doors. Motion in the X-axis will move the table to the right for negative numbers and to the left for positive numbers. Motion in the Y-axis will move the table away from the operator for negative numbers and toward the operator for positive numbers. Motion in the Z-axis will move the tool down for negative numbers and up for positive numbers.

The optional fourth, or rotary, axis can be programmed for both rapid positioning commands and for feed commands either by itself or in conjunction with the other axes.

In addition to the above, there may be up to five external axes that can be programmed for rapid or feed motions, but only one axis at a time.

To accomplish all of these functions, we use machining "G-Code" often referred to as "Fanuc" G-Code or ISO G-Code. This code is just a simple language. It is a simple language that consists of less than 300 words. As compared to English, Spanish, French, German, etc... which contain thousands of words, but are easily taught every day.

We are going to work with this language to train on Haas programming. In order to understand what a program is doing, we need to talk about several subjects:

Word Address Assignments (The ABCs of CNC)  
    Spindle Commands  
    Tool Change Commands  
Creating Tool Path with Linear and Circular Interpolation  
    G Code Definitions  
    M Code Definitions  
    Machine Defaults  
    Program Format  
    Canned Cycles and Hole Definition  
Canned Cycle Modifiers (Bolt Hole Circles, Bolt Arcs, Lines of Holes)  
    Cutter Compensation  
    Circular Pocket Milling  
    Helical Motion and Thread Milling  
    Circular Plane Selection  
    Subprograms and Subroutines  
    Haas Pocket Milling Cycle (G150)  
    Haas Text & Serial Number Engraving

## ***Alphabet Word Address Assignments***

Below is a list of Word Address Letters (otherwise known as the ABCs of CNC) for a Haas VMC:

### **A FOURTH AXIS ROTARY MOTION**

The letter **A** is used to specify motion for the optional fourth, **A**, axis. It specifies an angle in degrees for the rotary axis. We can assign a value of rotary motion between -8380.000 degrees, and 8380.000 degrees. Both positioning and simultaneous motion can be accomplished with a rotary axis. Normally the A axis is designated as rotation around the X axis.

### **B FIFTH AXIS ROTARY MOTION**

The letter **B** is used to specify motion for the optional fourth, **B** axis. It specifies an angle in degrees for the rotary axis. We can assign a value of rotary motion between -8380.000 degrees, and 8380.000 degrees. Both positioning and simultaneous motion can be accomplished with a rotary axis. Normally the B axis is designated as rotation around the Y axis.

### **C AUXILIARY EXTERNAL ROTARY AXIS**

The letter **C** is used to specify motion for the optional fourth, **C** axis. It specifies an angle in degrees for the rotary axis. We can assign a value of rotary motion between -8380.000 degrees, and 8380.000 degrees. This axis is an optional axis that is interfaced thru the control for positioning moves only. Normally the C axis is designated as rotation around the Z axis.

### **D TOOL DIAMETER SELECTION**

**D**'s are used to define a tool diameter offset from the Tool Offset Page. We can choose a D value from D01 – D200, which corresponds to the "Geometry" column in the Tool Offset Page. For example, D01="Tool Offset Value Number 1", D02="Offset Number 2", etc.....

### **E CONTOURING ACCURACY**

**E**'s are used in conjunction with the G187 code that is "Haas Specific". G187 is defined as Contouring Control with machines that have the high speed machining option. G187 is used to control the machine during high feed rates and control the acceleration / de-acceleration of the machine's axis. The range of values possible for the E code is 0.0001 to 0.25. Normally setting #85 is set to .005 and setting #191 is set to medium on machines with the high speed machining option.

**E** is also used in the G47 Engraving Canned Cycle to prescribe the infeed rate in in/min.

### **F FEED RATE**

**F**'s are used to define the speed of the movement of the spindle as it travels. Typically used while the tool is in the material, this is either defined as Inches Per Minute (IPM) or in Millimeters Per Minute (MMPM). It is the distance that the machine would move in one minute (Example, F10.0 = 10 Linear IPM of Speed)

## **G PREPARATORY FUNCTIONS (G CODES)**

G Codes establish Modes of Operation. When we define a G Code, think of it like we are flipping a rotary switch on a TV to another mode (channel). For example, G83 is the code for Deep Hole Peck Drilling. We tell the machine “G83” and then tell the machine where the holes are located since we are in Drilling Mode. When done, we take the machine out of Drilling Mode with a “G80” which means Canned Cycle Cancel. G codes are used to establish what “Mode” the machine is in.

## **H TOOL LENGTH OFFSET VALUE**

The H is used to tell the machine what tool length value to use from the Tool Offset page. If we define H01, we are telling the machine to use the value that is located under Tool Length #01. H02 = Tool Length Value #2

## **I CANNED CYCLE AND CIRCULAR OPTIONAL DATA**

The letter I is used two different ways. It can be used in canned cycles (Drilling Operations) and it is used in defining arcs, in that we tell the machine incrementally from the start point of an arc, where the center of the arc is. I is used to tell the machine how far away the center of the arc is in the X axis.

## **J CANNED CYCLE AND CIRCULAR OPTIONAL DATA**

The letter J is used two different ways. It can be used in canned cycles (Drilling Operations) and it is used in defining arcs, in that we tell the machine incrementally from the start point of an arc, where the center of the arc is. J is used to tell the machine how far away the center of the arc is in the Y axis.

## **K CANNED CYCLE AND CIRCULAR OPTIONAL DATA**

The letter K is used two different ways. It can be used in canned cycles (Drilling Operations) and it is used in defining arcs, in that we tell the machine incrementally from the start point of an arc, where the center of the arc is. K is used to tell the machine how far away the center of the arc is in the Z axis.

## **L LOOP COUNT FOR REPEATED CYCLES**

The L address character is used to specify a repetition count for some canned cycles and auxiliary functions. It is followed by an unsigned number between 0 and 32767.

## **M MISCELLANEOUS FUNCTIONS (M CODES)**

M codes are used to turn on and off functions specific to that of the machine. For example, M3 and M4 turn the spindle on, M5 turns the spindle off. M8 turns coolant on, M9 off. Think of it like M means “Machine Function”.

## **N NUMBER OF BLOCK**

The N address character is entirely optional. It can be used to identify or number each block of a program. It is followed by a number between 0 and 99999. The M97 functions may reference an N line number.



## **O PROGRAM NUMBER/NAME**

The O address character is used to identify a program. It is followed by a number between 0 and 99999. A program saved in memory always has an **Oxxxxx** identification in the first block; it cannot be deleted. Altering the O in the first block causes the program to be renamed. A program can only have one O address.

## **P DELAY TIME OR PROGRAM NUMBER**

P is another dual function letter in that it can be used to define a pause with a G04 code, or it is used with a M97 or M98 code to tell the machine to “jump” from its current place in a program to another place in the program (in the case of M97 P100 = Jump to line N100) or to another program entirely (with M98 P520 = Jump to program O520). A length of a pause can be defined two different ways, in Seconds (with a decimal point) or Milliseconds (without a decimal). G4 P.1 would mean wait .1 seconds, and G4 P100 (without a decimal) would mean wait 100 Milliseconds. Both .1 Seconds and 100 Milliseconds are the same amount of time.

## **Q CANNED CYCLE OPTIONAL DATA**

The letter Q is used in canned cycles, most often as the “Peck” distance in a drilling cycle.

## **R CANNED CYCLE AND CIRCULAR OPTIONAL DATA**

R is another dual role character. It can be used in canned (drilling) cycles to define the “Rapid Plane” (how far above the part to rapid the tool to), or it is used in defining an arc’s radius (replacing the I, J, and K method). Refer to the Line and Arc Interpolation Section of this manual for more detail.

## **S SPINDLE SPEED COMMAND**

S defines the spindle rpm. We can use a value anywhere between S0 – S99999. If we define a speed higher than the capacity of the machine’s spindle, it will max out the machine’s RPM and start cutting. For example, a standard VF spindle is 7,500 RPM. If we tell the machine S15000 M03 (turn on the spindle forward at 15,000 RPM) the machine will go to the 7,500 and start cutting. Be aware of your machine’s capabilities (4k, 7.5k, 10k, 12k, 15k or 30k) before programming speeds and feeds.

## **T TOOL SELECTION CODE**

A standard Haas VMC can be equipped with a 10, 20, 24, 30, 40, etc.... Tool Changers, but the control has the capability of saving in its memory up to 200 Tools. It is possible to STORE tool number 121 in a machine that only has a 24-tool Tool Changer. A T code tells the machine what tool we want to put in the spindle. \*NOTE\* on Haas machines with a side mount tool changer (Tools are stored randomly in the magazine) a “Tool Pre-Call” may be necessary to “Stage” the tool change.

## **X      LINEAR X-AXIS MOTION**

The **X** address character is used to specify motion for the X-axis. It specifies a position or distance along the X-axis. It is either in inches with four fractional positions or mm with three fractional positions. It is followed by a signed number in inches between -8380.000 and 8380.000 or between -83800.00 and 83800.00 for metric. If no decimal point is entered, the last digit is assumed to be 1/10000 inches or 1/1000 mm.

## **Y      LINEAR Y-AXIS MOTION**

The **Y** address character is used to specify motion for the Y-axis. It specifies a position or distance along the Y-axis. It is either in inches with four fractional positions or mm with three fractional positions. It is followed by a signed number in inches between -8380.000 and 8380.000 or between -83800.00 and 83800.00 for metric. If no decimal point is entered, the last digit is assumed to be 1/10000 inches or 1/1000 mm.

## **Z      LINEAR Z-AXIS MOTION**

The **Z** address character is used to specify motion for the Z-axis. It specifies a position or distance along the Z-axis. It is either in inches with four fractional positions or mm with three fractional positions. It is followed by a signed number in inches between -8380.000 and 8380.000 or between -83800.00 and 83800.00 for metric. If no decimal point is entered, the last digit is assumed to be 1/10000 inches or 1/1000 mm.

## ***Preparatory Functions (G Codes)***

The definition of “G” code is typically referred to as a “preparatory function”. They establish the mode of operation that the machine needs to be in to accomplish what the programmer intends. Imagine a rotary switch like that on an older TV; we are just turning the switch to different “modes”.

Before considering the meaning and the use of codes, it is helpful to lay down a few guidelines:

- 1) Codes come in **groups**. Each group of codes will have a specific group number. (Imagine each group of codes as a knob on a TV)
- 2) A “G” code from the same group can be replaced by another code in the same group. By doing this, the programmer establishes modes of operation. The universal rule here is that codes from the same group cannot be used more than once on the same line. (We cannot have a knob in two different positions)
- 3) There are **modal G codes**, which, once established, remain effective until replaced with another code from the same group (Like a light switch on a car, turn the switch on it stays on till it is turned off)
- 4) There are **non-modal G codes** which, once called, are effective only in the calling block and are immediately forgotten by the control (Like a horn in a car, it is only on momentarily)

The rules above govern the use of all codes for programming the Haas (and other) controls. The concept of grouping codes and the rules that apply will have to be remembered if we are to effectively program the machine tool. The following is a discussion of the codes most basic to the operation of the machine.

The following two pages display a summary of the G codes, A " \* " indicates the default within each group, if there is one:

Code:	Group:	Function:
G00	*01	Rapid Motion
G01	01	Linear Interpolation Motion
G02	01	CW Interpolation Motion
G03	01	CCW Interpolation Motion
G04	00	Dwell
G09	00	Exact Stop (non-modal)
G10	00	Programmable Offset Setting
G12	00	CW Circular Pock Milling (Yasnac)
G13	00	CCW Circular Pock Milling (Yasnac)
G17	*02	XY Plane Selection (circular interpolation)
G18	02	ZX Plane Selection (circular interpolation)
G19	02	YZ Plane Selection (circular interpolation)
G20	06	Inch Programming Selection
G21	06	Metric Programming Selection
G28	00	Return to Machine Zerothrough Reference Point
G29	00	Move to Location through G28 Reference- never used
G31	00	Skip Function (used in probing)
G35	00	Automatic Tool Diameter Measurement (probing)
G36	00	Automatic Work Offset Measurement (probing)
G37	00	Automatic Tool Length Measurement (probing)
G40	*07	Cutter Comp Cancel
G41	07	Cutter Compensation Left
G42	07	Cutter Compensation Right
G43	08	Tool Length Compensation
G44	08	Tool Length Compensation (never used)
G47	00	Engraving
G49	*08	G43/G44 Cancel
G50	11	G51 (scaling) Cancel
G51	11	Scaling (option)
G52	12	Select Work Coordinate System G52 (global work shift) (Yasnac)
G52	00	Set Local Coordinate System (Fanuc)
G52	00	Set Local Coordinate System (HAAS)
G53	00	Non-Modal Machine Coordinate Selection
G54	*12	Select Work Coordinate System 1
G55	12	Select Work Coordinate System 2
G56	12	Select Work Coordinate System 3
G57	12	Select Work Coordinate System 4
G58	12	Select Work Coordinate System 5
G59	12	Select Work Coordinate System 6
G60	00	Unidirectional Positioning (never used)
G61	13	Exact Stop Modal
G64	*13	G61 Cancel
G65	00	Macro Subroutine Call (used in conjunction with P value)
G68	16	Rotation (option, comes with probing)
G69	16	G68 Cancel

Code:	Group:	Function:
G70	00	Bolt Hole Circle (Yasnac)
G71	00	Bolt Hole Arc (Yasnac)
G72	00	Bolt Holes Along an Angle (Yasnac)
G73	09	High Speed Peck Drill Canned Cycle
G74	09	Reverse Tap Canned Cycle
G76	09	Fine Boring Canned Cycle
G77	09	Back Bore Canned Cycle
G80	*09	Canned Cycle Cancel
G81	09	Drill Canned Cycle
G82	09	Spot Drill Canned Cycle
G83	09	Peck Drill Canned Cycle (for deep holes)
G84	09	Tapping Canned Cycle
G85	09	Boring Canned Cycle
G86	09	Bore/Stop Canned Cycle
G87	09	Bore/Manual Retract Canned Cycle
G88	09	Bore/Dwell Canned Cycle
G89	09	Bore Canned Cycle
G90	*03	Absolute positioning
G91	03	Incremental positioning
G92	00	Set Work Coordinates - FANUC or HAAS
G92	00	Set Work Coordinates - YASNAC
G94	05	Feed per minute
G95	05	Feed per revolution
G98	*10	Initial Point Return
G99	10	R Plane Return
G100	00	Disable Mirror Image
G101	00	Enable Mirror Image (comes with probing)
G102	00	Programmable Output to RS-232
G103	00	Block Look ahead Limit (used in conjunction with P value)
G107	00	Cylindrical Mapping
G110-G129		Select Coordinate System 7 thru 26, group 12
G154 P1-P99		Select Coordinate System 7 thru 106 (Newer Machines) group 12
G136	00	Automatic Work Offset Center Measurement
G150	00	General Purpose Pocket Milling
G187	00	Accuracy Control for High Speed Machining



## ***Machine Functions (M Codes)***

### **Typical Haas M Codes:**

M Codes are used by the programmer to turn on and off certain functions of the machine. Think of M codes as codes that turn on and off different Machine Functions.

#### **M00 Stop Program**

The M00 code is used to stop a program. It also stops the spindle and turns off the coolant and stops interpretation look ahead processing. This is used to force the operator to interact with the machine (such as check a dimension, flip a part over, blow chips from a hole to tap, etc...)

#### **M01 Optional Program Stop**

M01 works much like M00, providing the **OPT STOP** mode is **ON**. If this mode is turned on, the machine will stop at M01, if it is turned off, it is ignored. Often used when the operator has discretion on stopping the machine or not (usually at a tool change)

#### **M03 Spindle Forward**

The M03 code will start the spindle moving in a clockwise direction at whatever speed was previously set. The machine will stop and wait for acceleration of the spindle to full speed prior to moving to ensure the spindle is ready to make a cut.

#### **M04 Spindle Reverse**

The M04 code will start the spindle moving in a counterclockwise direction at whatever speed was previously set.

#### **M05 Spindle Stop**

The M05 code is used to stop the spindle. The block is delayed until the spindle slows below 10 rpm.

#### **M06 Tool Change**

The M06 code is used to initiate a tool change. The previously selected tool (Tn) is put into the spindle. If the spindle was running, it will be stopped. No previous axis commands are required before the tool change unless there is a problem with tool/part/fixture clearance. The Z-axis will automatically move up to the machine zero position and the selected tool will be put into the spindle. The Z-axis is left at machine zero. The spindle will not be started again after the tool change but the Snnnn speed and gear will be unchanged. The Tnn must be in the same block or in a previous block. The coolant pump will be turned off during a tool change and a air purge will open to keep chips out of the spindle.

#### **M08 Coolant On**

The M08 code will turn on the coolant supply. Note that the M code is performed at the end of a block so that if a motion is commanded in the same block, the coolant is turned on after the motion. The low coolant status is only checked at the start of a program so that a low coolant condition will not stop a program which is already running.

**M09 Coolant Off**

The M09 code will turn off the coolant supply.

**M10 Engage 4<sup>th</sup> Axis Brake**

The M10 code is used to apply the optional brake to the 4<sup>th</sup> axis. It is only used when M11 is used to release the brake.

**M11 Release 4<sup>th</sup> Axis Brake**

The M11 code will “pre-release” the 4<sup>th</sup> axis brake. This is useful to prevent the delay otherwise occurring when a 4<sup>th</sup> axis is used with a brake and a motion is commanded in that axis. It is not required, but without a prior M11, there will be a delay in motion in order to release the air.

**M16 Tool Change**

The M16 code is used to initiate a tool change. In the present machine configuration, M16 works exactly like M06. (normally not used)

**M19 Orient Spindle**

The M19 code is used to orient the spindle to a fixed position. This command leaves the spindle in that position and locked. The next spindle motion command (Snnnn, M3, M4, M41, or M42) will unlock the spindle.

**M21-M28 Optional User M**

The M21 through M24 codes are optional for user interfaces. They will activate one of relays 25 through 28, wait for the M-fin signal, release the relay, and wait for the M-fin signal to cease. The RESET button will terminate any operation that is hung-up waiting for M-fin.

**M30 Program End and Rewind**

The M30 code is used to stop a program. It also stops the spindle and turns off the coolant. The program pointer will be reset to the first block of the program and stop. The parts counters displayed on the Current Commands display are also incremented. M30 will also cancel tool length offsets.

**M31 Chip Conveyor Forward**

M31 starts the chip conveyor motor in the forward direction.

**M32 Chip conveyor Backward**

M32 starts the chip conveyor motor in the reverse direction.

**M33 Chip Conveyor Stop**

M33 stops conveyor motion.



**M34 Increment Coolant Spigot Position Up**

M34 increments the current spigot position up one place. When the machine loads a tool, and it is equipped with the PCOOL option, it will associate a “Coolant Position” (0 is all the way up, 34 is almost straight down) with each tool and change the coolant nozzle position for each individual tool. M34 will pull the nozzle up one position from where it currently sits. If we need to go up three positions, we need to call M34; M34; M34. One can actually use the nozzle to blow chips out of a hole with a dwell and a series of M34 commands.

**M35 Decrement Coolant Spigot Position**

M35 increments the current spigot position down one place. When the machine loads a tool, and it is equipped with the PCOOL option, it will associate a “Coolant Position” (0 is all the way up, 30 is almost straight down) with each tool and change the coolant nozzle position for each individual tool. M35 will pull the nozzle down one position from where it currently sits.

**M39 Rotate Tool Turret**

The M39 is used to rotate the tool turret without performing a tool change. If you have a large tool in an Umbrella Style tool changer, and are worried about running it into the part while machining, we can index the turret to move the tool to the back of the sheet metal.

**M41 Low Gear Override**

If your machine has the optional high torque gear box option, M41 will force the machine into low gear. Usually the machine will decide what gear to use based on RPM (0-1000 RPM uses low gear, 1000RPM and higher uses high gear)

**M42 High Gear Override**

If your machine has the optional high torque gear box option, M42 will force the machine into high gear. Usually the machine will decide what gear to use based on RPM (0-1000 RPM uses low gear, 1000RPM and higher uses high gear). This is useful if don't need all that torque at a lower RPM.

**M51-M58 Optional User M On**

M51 through M54 codes are optional for user interfaces. They will activate one of relays 17 through 24 and leave it active. These are the same relays used for M21-M28. Use M61-M68 to turn these off. The RESET button will turn off all of these relays.

**M59 Output Relay Set (N)****M61-M68 Optional User M Off**

The M61 through M68 codes are optional for user interfaces. They will deactivate one of relays 17 through 24. These are the same relays used for M21-M28.

**M69 Output Relay Clear (N)****M76 Disable Displays**

This code is used to disable the updating of the screen displays during high speed machining. It is used to increase the speed of the central processing unit in the controls of the Haas machine tool.

**M77 Enable Displays**

This code is used to enable the updating of the screen displays at the end of high speed machining.

**M80 Automatic Door Open (option)**

**M81 Automatic Door Close (option)**

**M82 Tool Unclamp**

This code is used to release the tool from the spindle. It is not normally needed since tool change operations do this automatically and a manual TOOL RELEASE button is available to the operator. ***This M code is not recommended for use since the tool will be dropped from the spindle and may damage the tool, the machine, or your setup.***

**M83 Auto Air Jet On (option)**

**M84 Auto Air Jet Off (option)**

**M86 Tool Clamp**

This code will clamp a tool into the spindle. It is not normally needed since tool change operators do this automatically and a manual TOOL RELEASE button is available to the operator.

**M88 Thru the Tool Coolant On (option)**

This code turns on high pressure coolant thru the tool. Make sure you have a thru the tool retention knob and also a tool with holes for thru the tool capability.

**M89 Thru the Tool Coolant Off (option)**

**M95 Sleep Mode (Long Dwell)**

This code may be used to warm up the machine before operation. Format: M95(hh:mm)  
The lines following may call up the spindle warm-up program (M98 P2020)

**M97 Local Sub-Program Call**

M97 is a code that we use to jump from where we currently sit in our program and to “jump” to a different spot in the program. We follow the M97 with a P word that corresponds with a line number. M97 P100 = Jump to line N100

**M98 Sub Program Call**

M98 works like M97, but jumps us out of the program we are currently in, and to another program entirely. This is most often used when we have similar functions or features on different parts (such as pockets, bolt hole patterns, etc....). M98 P510 = Jump to Program O510.

**M99 Sub Program Return or Loop**

M99 is used when we need to return from our M97 or M98 program jump. It tells the machine “go back from where you came”. If we do not use a M97 or M98, but put in a M99, the machine will jump to the beginning of the program and re-run the program.

## ***Program Structure and Format***

The sequence of events necessary to successfully develop a CNC program will be discussed in detail in this unit.

### ***Objectives:***

Upon completion of this unit, the student will:

- 1) Be familiar with the proper sequence of events that should be followed when developing a CNC program.
- 2) Be able to define “Safety Line” and be familiar with the codes it contains.
- 3) Understand critical information that must be included before and after each tool.
- 4) Be capable of following an existing program and explain the sequence of events that it follows.
- 5) Understand the definition and purpose of “default” values.

## ***Program Format***

Program format, or style, is an important part of CNC machining. Each individual will have their own way to format their programs differently but in most cases, a programmer could not identify a program written by them 6 months after they wrote it. The point is that a programmer needs to be consistent and efficient by writing code in the way it is listed and in the order it appears in the program. For example: **X, Y, Z**, should be written in order of appearance. The machine will read **X, Y, or Z** in any order, but we want to be consistent. Write **X** first, **Y** second, **Z** third. This will help ensure confidence and trust between you the programmer and others who may use the program.

We should start any program with a “**safety line**” even though it may be redundant, we want to put our G Code Switches (Grouping of G Codes) in the position we intend on using them at the start of our program. G00 puts machine in a rapid mode. G17 selects the X-Y plane for circular interpolation. G40 cancels cutter compensation. It is a good idea to cancel Tool Length Comp (G49). G80 cancels any canned cycles. G90 puts machine in absolute. G98 instructs machine to return to initial start point in any canned drill cycles. This will FORCE the machine into what we want it to do. We do not want to assume the machine is already set the way we want it.

### **G00 G17 G40 G49 G80 G90 G98 (SAFETY LINE)**

The next line we want to return to machine zero in Z, the tool index position.

### **G28 G91 Z0 or (G00 G53 Z0)**

The next line we want to enter is a tool change. The correct tool may be in the spindle already, and we don't need to do a tool change to run our part, but the next time we start our program, that tool may not be in the spindle. Force the machine to put the tool we want in the spindle. Never assume it is already there.

### **T1 M06 (LOAD TOOL 1)**

The fourth line or block will contain a rapid command (G00), an absolute or incremental command (G90, G91), a work zero for **X** and **Y** (G54), a positioning **X** and **Y** coordinate, a spindle speed command (S---), and a spindle ON clockwise command (M03).

### **G00G90G54 X\_\_\_\_\_Y\_\_\_\_\_S\_\_\_\_\_M03 (RAPID, ABSOLUTE, OFFSET #1, SPINDLE ON)**

The next line will contain a "read tool length compensation" command (G43); a tool length offset number (H01), a Z-axis positioning move (Z.1), and an optional coolant ON command (M08).

### **G43 H01 Z1.0 M08 (TOOL LENGTH COMP OFFSET #1, GO TO Z1.0, TURN COOLANT ON)**

An example program's first five lines will look like this:

```
G00 G17 G40 G49 G80 G90 G98;  
G00 G53 Z0;  
T1 M06;  
G00 G90 G54 X0 Y0 S2500 M03;  
G43 H01 Z1.0 M08;
```

All the necessary code is listed above for each approach we will make towards the part. There are several variations on the above code that will work and net the same results, and it is up for you to decide what you like best, but once you do, stay with it and be consistent!

**QUESTION:** If G00, G90, and G54 are defaults, why do we list them in the second line of a program and for each different tool?

**ANSWER:** G00, G90, and G54 are listed for an operator/setup person's aid so he/she can determine if the machine will rapid position, if the machine is in fact in the absolute coordinate mode, and most important, the work zero. The work zero is always different between setups, and multiple work zeros are very common.

**QUESTION:** Can we combine the third and fourth lines excluding the M08 code? If so, why do we write the lines separately?

**ANSWER:** Yes. The four G codes G00, G90, G54, and G43 all belong to different groups. Remember, no two G codes of the same group can be listed on the same line.

The main reason for using two lines is SAFETY. Remember, only one line of information can be executed at a time. The **X** and **Y** coordinates will position first and then the tool length and the **Z** coordinate move will be executed. If combined, all three axis will move simultaneously, and any interfering clamps or fixtures can be struck and/or destroyed. Combining an **X**, **Y**, and **Z** move together will increase the chance of crashing the machine.

## ***Machine Defaults***

When we first power up the control, the machine loads its software and parameters, and the machine needs to go to some sort of "base" settings as far as our G-Codes are concerned. The switches have to be turned to some sort of position, and we call these the "defaults".

The control automatically reads these G codes when power is turned on:

G00	Rapid traverse
G17	X, Y Circular plane selection
G40	Cutter Compensation cancel
G49	Tool length compensation cancel
G54	Work coordinate zero #1 (1 of 109 available)
G64	Exact stop cancel
G80	Canned cycle cancel
G90	Absolute programming
G98	Initial point return

There is no default FEED RATE (F code), but once an F code is programmed, it will apply until another is entered or the machine is turned off.

The spindle is also set to a speed of 0 RPM when first turned on, so an S command is needed in order to turn on the spindle even in manual mode.

## ***Programming with Codes***

A program is just a set of written instructions given in the order they are to be performed.

Imagine if we could send the machine an “E-Mail” in plain English telling it what we wanted it to do. The program would look something like this:

```
STEP #1 = SELECT CUTTING TOOL
STEP #2 = TURN THE SPINDLE ON AT A CERTAIN RPM
STEP #3 = TURN THE COOLANT ON
STEP #4 = RAPID TO OUR START POSITION
STEP #5 = MAKE OUR CUTS AT A CERTAIN FEED RATE
STEP #6 = TURN OFF THE SPINDLE AND THE COOLANT
STEP #7 = RETURN TOOL TO HOME AND SELECT NEXT TOOL
```

...and so on. But our machine control only understands these messages when given in machine code.

*Why do we use G&M Code and not a different language (English for example)?*

CNC Machines were first built in the 1940's and 50's with the invention of computers. They were custom built, and several companies were trying to come to market with the first consumer available NC machine tools.

It was the early 60's before CNC machines became commercially available, and not only was the machine invented, so was the computer, and the software to run the computer. These first computers were limited and didn't have enough memory to remember an entire language such as English. So “Code” had to be invented, and was limited to words that had one letter and two numbers behind it (G 0 0 for Example).

Several inventors came out with machine and controls in the early days, and each one had to invent their own “code”. Soon there were several brands and several different kinds of code. This is when the ISO (International Standards Organization) said they were going to set a “Standard” code. They went with the code that was by far the most popular at the time, FANUC. This is why it is also referred to as “Fanuc G-Code”

The basics of this code remain the same to this day. Things get added from time to time, and it just adds new codes to the end of the G-Code list. Last code invented was G187 Machine Accuracy Control, and came out in the late 1990's. The first G-Codes were G00, G01, G02, and G03. They were established in the 50's. As a new need for a new machine “mode” is invented, new G-Codes will get added on to the list.

*Why do we stick with the code since technology has come so far?*

Standardization and Familiarity: It has been a great benefit in many ways. Operators and programmers trained on one machine can move to another brand, control, shop, etc... and with a little familiarization can get up to speed quickly. Companies can share programs.

## ***Program Structure***

In the CNC world, the term “**Block**” is used to define what we know as a line of code. The reason it is called a block is that the machine can read the entire line all at once, not from left to right as we are used to. A typical program consists of one or more blocks of commands.

After each block we need to end the block with a semi-colon (;). On the operator’s panel, the EOB key (End of Block) will generate this symbol. This is how the machine tells one line of code from the next.

The " / " symbol, sometimes called a forward slash, is used to define an optional block. A block that contains this symbol can be optionally deleted when the machine is in the block delete mode (BLKDEL) when running a program.

The machine does not read left to right as we do, again it takes a whole block in at one time. We can arrange our G-Codes, M-Codes, and Alphabetic Address Codes (X, Y, Z, etc.....) in any order we want. The rules are we can have one (G-Code) from each group, only one (M-Code), and alpha characters on each line. The machine will ALWAYS run the G-Codes and alpha characters first and the M-Code last.

### **PROGRAM START-UP LINES**

<b>G00 G17 G40 G49 G80 G90 G98</b>	(Safety Line)
G00	Rapid
G17	Selects X-Y Work Plane
G40	Cancels Cutter Comp
G49	Cancels Tool Length Offset
G80	Cancels Canned Cycle
G90	Absolute
G98	Initial Point Return for canned cycles
<b>G91 G28 Z0</b> <b>(G00 G53 Z0)</b>	G28 This preparatory function causes a return to machine zero of all specified axes. If an X, Y or Z are specified, only those axis move incrementally (G91) to machine home.
<b>T1 M06</b>	T1 Selects tool number one to be loaded into the spindle. M06 Activates the tool change sequence.
<b>G90 G54 G00 X__ Y__ S1200 M03</b>	G90 Activates control to be in <b>ABSOLUTE</b> . G54 Selects work coordinate system No. 1. G00 Preparatory function for a <b>Rapid Movement</b> . X__ Axis move to initial “ <b>X</b> ” position. Y__ Axis move to initial “ <b>Y</b> ” position. S1200 Informs the control that <b>1200</b> has been selected as the RPM for this tool M03 Turns the spindle “ <b>ON</b> ” in a clockwise direction at a speed of 1200 RPM.
<b>G43 H01 Z.1 M08</b>	G43 Activates the tool length stored in the “H” code offset number. H01 Informs control as to what the stored offset value is. Z.1 Informs the control to move from full spindle retract to this “Z” value and apply the tool length offset. M08 Turns the coolant “ <b>ON</b> .”

## CUTTING TOOL PATH LINES

### PROGRAM ENDING LINES

<b>G00 Z1. M09</b>	G00	Preparatory function for a <b>Rapid Movement</b> .
	Z1.	Retracts tool tip to 1.0 above "PART" or "PROGRAM ZERO" in preparation for full spindle retract.
	M09	Turns the coolant "off."
<b>G00 G91 G28 Z0 M5</b>	G28	This preparatory function causes a return to machine zero of all axes. If an X, Y and Z are specified, only the axis specified will be moved to machine zero. Machine must be in incremental mode (G91).
	G49	Cancels Tool Length Offset
	M5	Stop spindle rotation
<b>(G00 G90 G53 Z0)</b>	G53	Non-modal selects machine coordinates.
<b>G28 G91 Y0 (G53 Y0)</b>	G91	Preparatory function for INCREMENTAL mode, which is required for rapid move to the G28 reference point.
	Z0.	This will ensure a full retract, of Z axis in the "UP" direction to machine zero.
	Y0	Ensure "Y" axis is in the full out position for ease of loading and unloading.
<b>M30</b>	M30	Signals END OF PROGRAM AND REWIND to the control. The program will reset to the first block of the program and stop. It also stops the spindle, turns off the coolant and cancels tool length offsets. It also makes the control ready for the next cycle.

Now with our numbers we need to input, we are somewhat limited when we go past the decimal, and there are some unique ways of programming. In the early days, the controls didn't have enough memory or smarts to even allow for a decimal, so all values were entered in what is called an 8 character assignment. All numbers had to have 8 numbers, 4 in front of the decimal, 4 behind:

10.000 = 00100000  
1.0000 = 00010000  
.0010 = 00000010  
.0003 = 00000003

As controls got smarter, they eventually were able to integrate a decimal into the machines. But, the above programming method is still viable. With that said, when we are entering a whole unit (such as 1 inch) we need to input the decimal (1 inch should be 1.0). If we input the number one, with no decimal behind it, the machine will read it as .0001, and one tenth of one thousandth is a long way away from 1.0



## ***Linear and Circular Tool Paths***

Tool paths are the movements that we program once we have loaded our tool, turned our spindle on, and have approached the part. It is the motion that we generate with a particular tool to generate the shapes and sizes we need to complete our parts.

### ***Objectives:***

Upon completion of this unit, the student will:

- 1) Understand the major differences in G01 and G00.
- 2) Understand and apply the five criteria needed to produce an arc.
- 3) Understand the principles of programming and applications with and without cutter compensation and the advantages and disadvantages of each.
- 4) Be capable of producing a tool path program, containing lines and arcs, with and without cutter compensation.
- 5) Be capable of determining feeds and speeds given an SFM and cutting tool, and be capable of determining feed rate given an RPM and chip load.
- 6) Be able to integrate a tool path into a part program.
- 7) Understand the rules governing the use of cutter compensation.
- 8) Have a basic understanding of the concept of arc in/arc out and some of its applications.

## ***Linear/Circular Movement – Creating Tool Path***

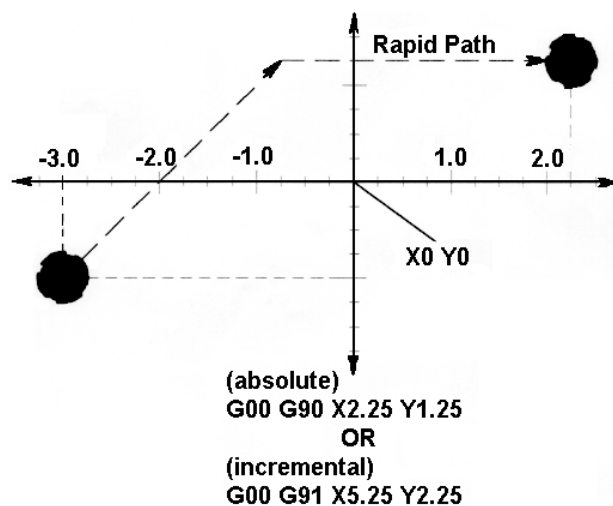
### ***Rapid Position Commands***

#### **G00 Rapid Motion Positioning**

X	Optional X-axis motion command
Y	Optional Y-axis motion command
Z	Optional Z-axis motion command
A	Optional A-axis motion command

This G code is for rapid traverse of the axes of our machine. The human mind thinks point to point, and as in the example below, we will move from X-3.0 Y-1.0 to X2.25 Y1.25 and we want to think the machine will move strait or “as the crow flies”. The code G00, means “MOVE AS FAST AS POSSIBLE”, which means move all axis at full speed. Since all of the axis will move at the same speed, the machine will make a 45 degree move till it achieves the position in one axis (Y in the example below) and finish the other till the point is achieved.

If we need to move in a strait line, the machine will only move in a strait line when told so, and that is what G01 is for.



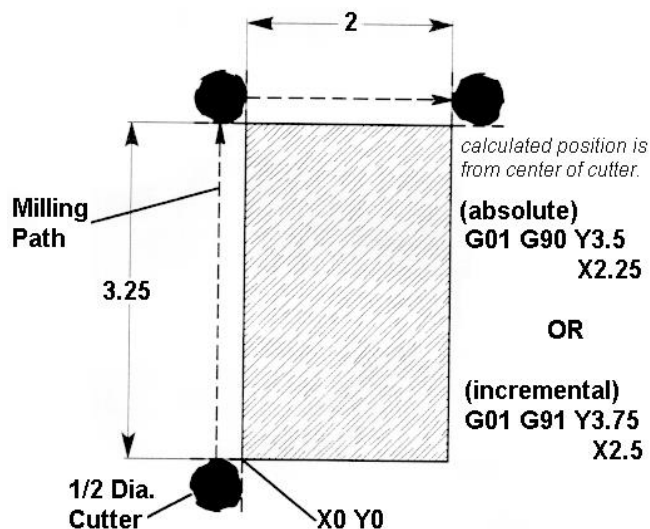
## Interpolation Commands

### G01 Linear Interpolation Commands

F	Feed rate in inches (mm) per minute
X	Optional X-axis motion command
Y	Optional Y-axis motion command
Z	Optional Z-axis motion command
A	Optional A-axis motion command

This G code provides for straight-line (linear) motion from point-to-point. Motion can occur in 1, 2 or 3 axes at a time. All our axes start and finish motion at the same time, including rotary axes.

The speed of all axes is controlled by a feed rate specified along with axis moves. Rotary axis feed rate is dependent on the rotary axis diameter setting (Setting 34) and will provide a controlled motion. The F command is modal and may be specified in a previous block. Only the axes specified are moved in either absolute (G90) or incremental (G91) modal commands, which change how values are interpreted.



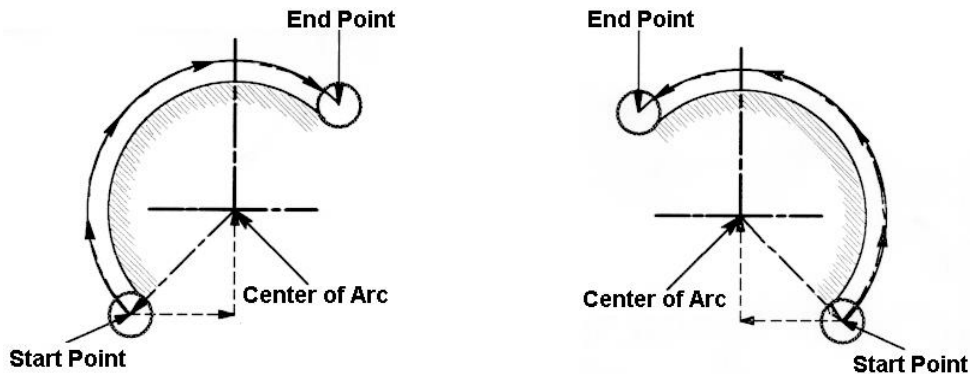
## Circular Interpolation (G02 and G03) Commands

### G02 CW Circular Interpolation Motion

F	Feed rate in inches (mm) per minute
I	Optional distance along X-axis to center of circle
J	Optional distance along Y-axis to center of circle
K	Optional distance along Z-axis to center of circle
R	Optional radius of circle
X	Optional X-axis motion command
Y	Optional Y-axis motion command
Z	Optional Z-axis motion command
A	Optional A-axis motion command

Circular interpolation commands are used to move a tool along a circular arc to the commanded end position.

**Five pieces of information are required for executing a circular interpolation command:**



- 1) Plane selection
- 2) Arc start position coordinates
- 3) Rotation direction
- 4) Arc end position coordinates
- 5) Arc center coordinates or arc radius.

**1 – Plane Selection** – The machine is capable of cutting an arc on different axes, basically meaning which two ball screws need to turn to achieve the desired arc. If we want a arc cut parallel to the top of the table of the machine (using the X & Y ball screws), we use a code called **G17 (X&Y Plane Selection)** to do so. NOTE\* G17 is a machine default, and is the most common setting. If we need to cut a arc along the left to right of the machine, down towards the table of the machine (using the X & Z ball screws) then we use the code G18. If we are going to cut an arc town towards the table along the front to rear of the machine (using the Y & Z) then we use G19.

**2 – Arc Start Position** – This is as simple as the position of the last move prior to the arc.

**3 - Arc Direction** – This is why we define G02 or G03, Clockwise vs. Counter – Clockwise from the start point. This is defined as the way the tool is looking at the arc, not necessarily the way you are looking at it on the machine.

4 – **Arc End Co-Ordinates** – This is where we want the arc to end. We define the end point of the arc in either G90 or G91, and we can use up to 3 axis to do so.

5 – **Center of Arc Location** – I, J, and K method or R method

**I,J,K** are used to define the distance from our known Start Point of our arc to the center of the arc. I = the incremental distance in X, J = the incremental distance in Y, and in the case of G18 and G19, K = the incremental distance in Z. All of these values tell the machine our arc “hinges” on a point so far from the arc start point.

R method enables the machine to find the location on it’s own. In our arc command line we define the end point, then tell the machine to maintain a certain radius between the points. The R method will work on arcs from 0 – 359.9999 degrees. If it is a full circle, we use I,J,K

Either system is valid. The R method arrived after machines were smart enough to calculate an arc.

### ***G02 Circular Interpolation Clockwise Command***

The circular interpolation contouring control uses the axis information contained in a block to move the tool in a **CLOCKWISE** arc of a circle up to 360 degrees.

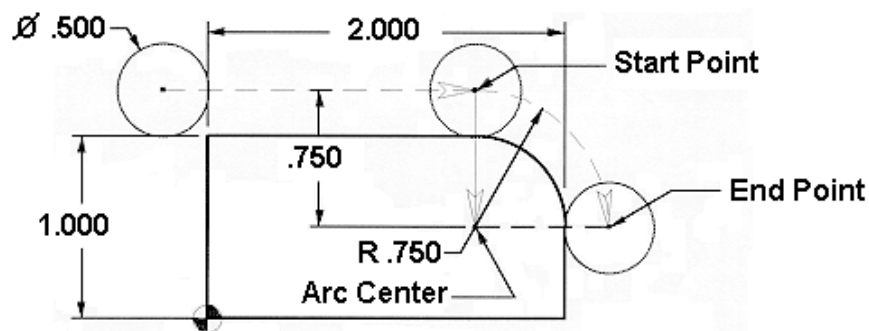
The velocity at which the tool is moved is controlled by the feedrate (F) command.

All circles are defined and machined by programming three pieces of information to the control:

- 1) **START POINT** of the arc
- 2) **END POINT** of the arc
- 3) **ARC CENTER** distance from the start point

The **START POINT** is defined prior to the G02 line, usually by a **G01** linear positioning move.

The **END POINT** is defined by the “X” and “Y” coordinates within the **G02** line when in the **G17 – XY PLANE**.



```
N4 G01 X-.25 F12  
N5 Y1.25  
N6 X1.500 (to start point)  
N7 G02 X2.250 Y.500 I0. J-.75 or (R.75)
```

The **ARC CENTER** is defined in the **G02** line by the “I” and “J” values, when in the **G17-XY PLANE**, or by a “R” value. Note the above will cut a .500 radius on the part.

To cut a **complete circle of 360 degrees**, program I, J, or K to define the center of the circle. You do not need to specify X, Y, or Z as an ending point. The following line will cut a **complete circle**:

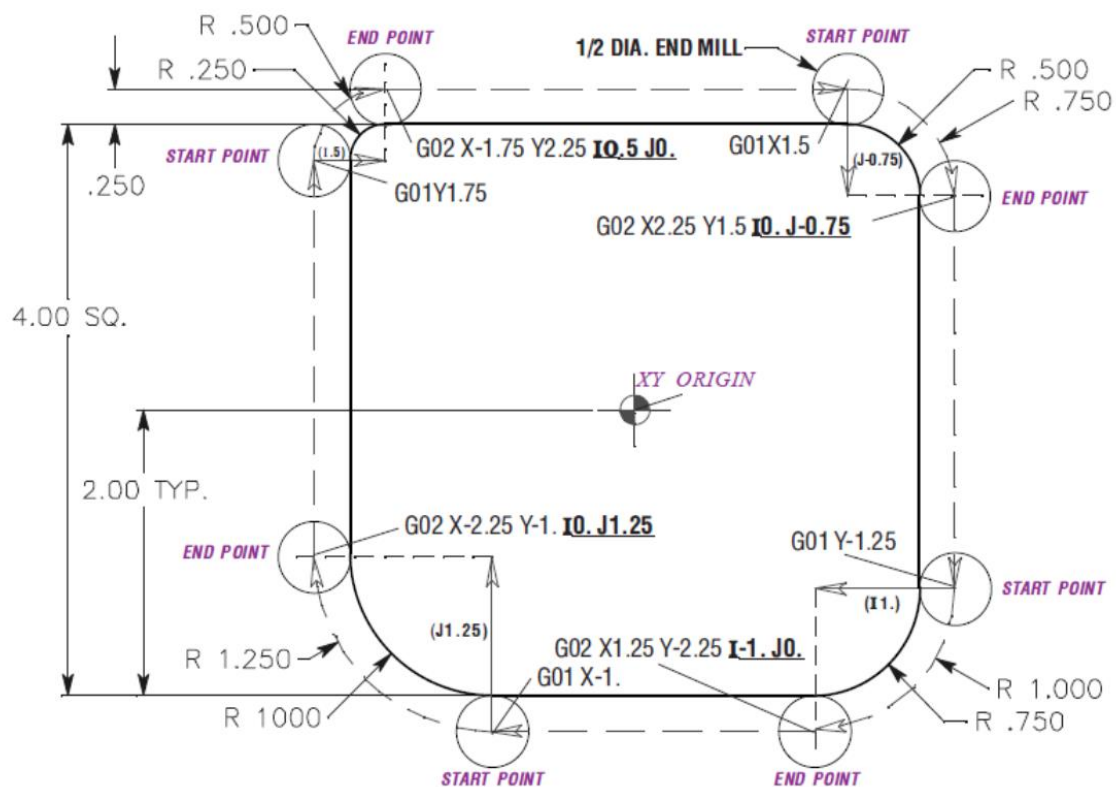
**G02 I-2.5** (this command will interpolate a 360 degrees, 5.0 diameter circle)

### ***G02 CW Circular Interpolation Using "I", and "J" for the X & Y Axis***

The "I", "J", and "K" values are INCREMENTAL distances from where the tool starts cutting the arc (START POINT) to the ARC CENTER.

When I, J or K are used to specify the center of the arc, R is not used. Only the I, J or K specific to the selected plane (IJ for G17, IK for G18, JK for G19) are allowed. If only one of the I, J K is specified, the others are assumed to be zero. The I, J or K is the distance from the starting point to the center of the circle. Using I, J, or K is the only way to cut a complete 360 degree arc; in this case, the starting point is the same as the ending point and no X, Y, or Z is needed. To cut a complete circle of 360 degrees (360°), you do not need to specify an ending point X, Y, or Z; just program I, J or K to define the center of the circle.

NOTE: This example of G02 circular moves is not using cutter compensation, and so the circular moves that are shown here are defined from the center of the cutter around the arc. For 90 degree corners or fillets the "I" and "J" values can be defined like you see below.



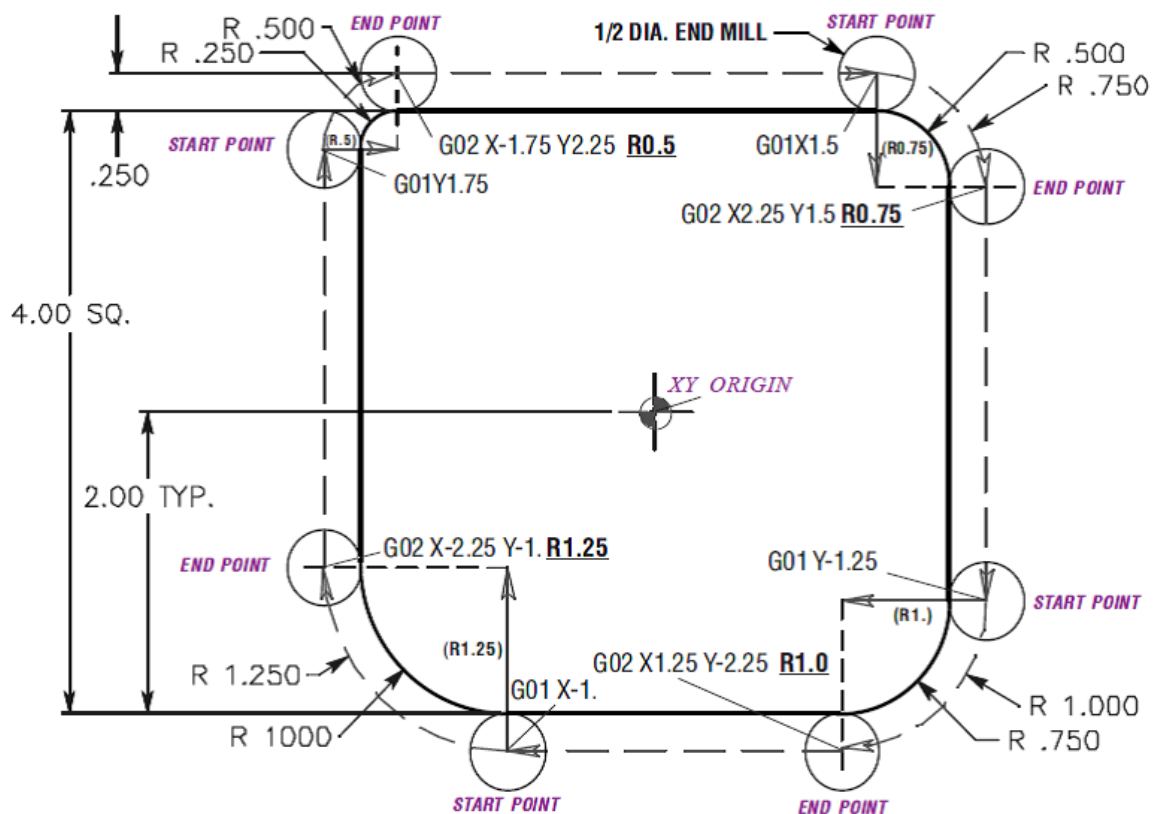
## G02 Circular Interpolation Clockwise Command Using "R"

The letter address "R" can be substituted for the letter addresses "I", "J", and/or "K" for various corners or fillets, which is more easily defined.

Just as "I", "J", and "K" are used to locate the ARC CENTER in relation to the START POINT, "R" does the same function with less input.

When R is used to specify the center of the circle, a complete 360 degree arc is not possible. When defining an arc less than 360 degrees using the R command, the X, Y, or Z needs to specify an endpoint different than the starting point. And R defines the distance from the starting point to the center of the circle. With a positive R, the control will generate a radius of 180 degrees or less; to generate a radius of over 180 degrees, you need to specify an R command with a negative value.

NOTE: This example of G02 circular moves is not using cutter compensation, and so the circular moves that are shown here are defined from the center of the cutter around arc. For 90 degree corners or fillets the "R" values can be defined like you see below:



## G03 Circular Interpolation Counterclockwise Command

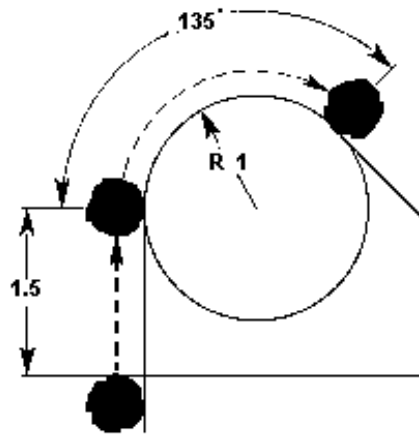
G03 Works exactly the same as G02, all that changes is the direction of the arc (clockwise vs. counterclockwise)

## Positive "R" vs. Negative "R"

The first R capable controls were able to use the R, but only up to 180 degrees. The reason being, is that for any two points in space, when a radius is defined, there are actually two paths that will use both points and that defined arc that are correct. One is less than 180 degrees, one is greater. We only could use the arcs less than 90 degrees. Eventually, we had enough computing power to tell the machine which one we want. Examples are below.

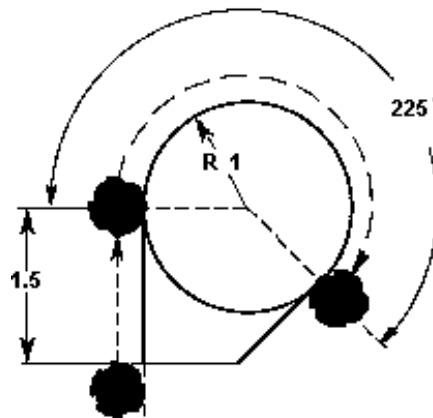
The following line will cut an arc less than 180 degrees using an R positive:

```
G90 G54 G00 X-.25 Y-.25  
G01 Y1.5 F12.  
G02 X1.884 Y2.384 R1.25
```



To generate an arc of more than 180 degrees using a negative R minus:

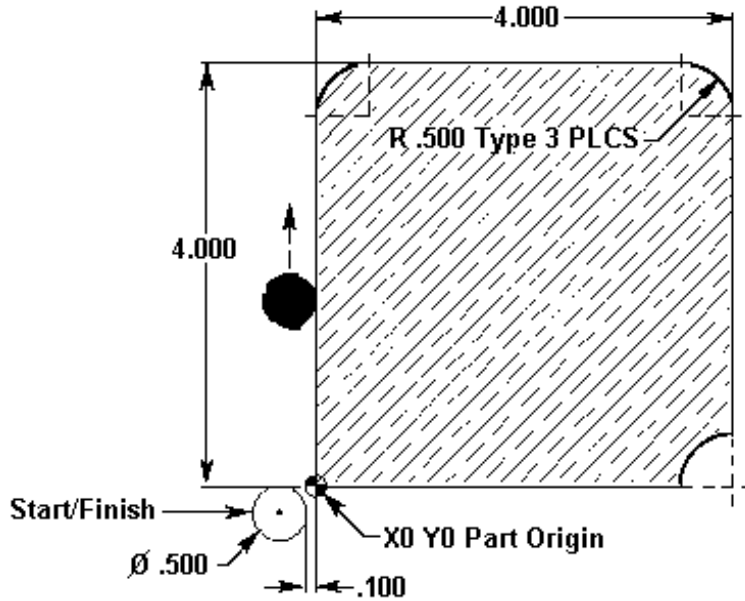
```
G90 G54 G00 X-.25 Y-.25  
G01 Y1.5 F12  
G02 X1.884 Y.616 R-1.25
```





**Interpolation Exercise**

Cutter will be a .500 diameter end mill. Start contour from the lower left corner of part and mill around outside of part .375 deep. To position around the part, mill around outside of part with tool by defining the very center of the cutter.



```

O0010 (Enter in this program as #10)
G00 G17 G40 G49 G80 G90 G98
G00 G53 Z0
T1 M06 (.500 diameter 4 flute. End Mill)
G90 G54 G00 X_____ Y_____ S1000 M03
G43 H01 Z.1 M08
G01 Z_____ F50.
      X_____ F10. (feed onto left side of part programming from center of tool)
      Y_____ (feed up to start point of upper left radius and continue on around to the end)
G0_ X_____ Y_____ I_____ J_____ or R_____
G0_ X_____
G0_ X_____ Y_____ I_____ J_____ or R_____
G0_ Y_____
      X_____
G0_ X_____ Y_____ I_____ J_____ or R_____
G0_ Y_____
      X_____
G00 Z1. M09
G00 G53 Z0 M05
G00 G53 Y0
M30
    
```



## ***Cutter Compensation (G41, G42)***

As you can see with the above examples, programming our contours can require a certain amount of thinking and math to work with our tool diameters. The math isn't hard, as long as we have sharp corners and even sized end mills and even sized parts. When we start running re-grind tooling, have special angels or arcs, or any other no-typical variables on our part, programming becomes more difficult.

As time progressed, the machine computer controls became sophisticated enough to self-compensate for our tool diameters. We could tell the machine we had a .500 roughing tool and finish with another tool that was .375 and still the same program. We just had to tell the machine our tool was of a different size.

We do this by using a feature called Cutter Compensation, or Cutter Comp. for short. With cutter comp we program the contour of our part, tell the machine what side of the profile to keep the tool on, and the tool diameter.

### ***Advantages of Cutter Compensation***

- 1) The mathematical computations for determining a tool path are greatly simplified.
- 2) The part, and not the tool center, is programmed. Therefore, the same program can be used for a variety of different cutter diameters.
- 3) The same program path can be used for roughing, as well as finishing cuts, by using different cutter offsets.
- 4) Inside, as well as outside, cuts can be performed.

### ***Some Restrictions with Cutter Compensation***

- 1) A cutter compensation command (G41 or G42) must be on the same block with an X and/or Y linear move when ramping on to the part using cutter compensation.
- 2) You can only use cutter compensation in the G17 XY plane.
- 3) Cancel of cutter compensation command (G40) must be on the same block with an X and/or Y linear move when ramping off the part using cutter compensation.

### ***When activating cutter compensation, use this type of procedure***

- 1) Position our tool at a X & Y location that is off of our part with enough room to allow us to move from un-compensated to fully compensated, at least  $\frac{1}{2}$  our tool diameter.
- 2) Bring the "Z" axis down without cutter compensation in effect.
- 3) From our position in (1) above, program a X & Y location that would put us exactly on the profile that we are trying to cut. Then include a G41 (LH) or a G42 (RH) compensation command to tell the machine what side of the profile to keep the tool on. Also include the D number of your tool. This move has to be as large or larger than the amount of offset (radius) that we are compensating for.

## ***When canceling cutter compensation, care must be taken to:***

- 1) Select a clearance point in “X” and/or “Y” axis off the part.
- 2) **DO NOT** cancel cutter compensation on any line that is still cutting the part.
- 3) Cancel of cutter compensation may be a one or two axis move.
- 4) Cancel of cutter compensation (G40) may need values entered for both “X” and “Y” axis. This may need to be done to ensure that both axes will position to the location you want or remain fixed and not move during the cancel (G40) process. This is a programming technique that may be a programmer’s preference.
- 5) Linear move equal to or greater than the amount being compensated for (radius of the tool).

### **The cutter compensation commands are:**

- 1) Cutter Comp Off (G40)
- 2) Cutter Comp Left (G41)
- 3) Cutter Comp Right (G42).

### **G40 Cutter Compensation Cancel**

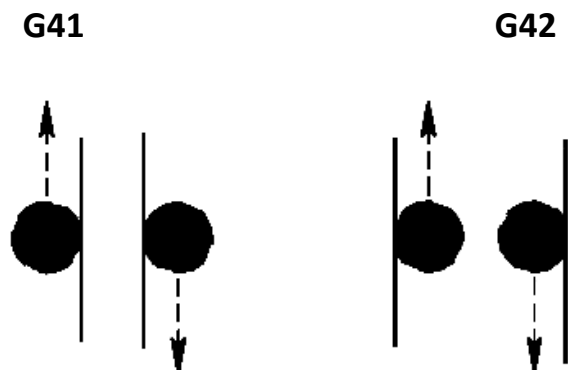
G40 will cancel the G41 or G42 cutter compensation commands. The tool will change from a compensated position to an uncompensated position. Programming D00 will also cancel cutter compensation.

### **G41 Cutter Compensation Left**

G41 will select cutter compensation **left**; that is, the tool is moved to the left of the programmed path to compensate for the radius of the tool. A Dnn must also be programmed to select the correct tool size from the Tool Offset Register (D01 = Diameter Offset #1, D2 = #2, etc..)

### **G42 Cutter Compensation Right**

G42 will select cutter compensation **right**; that is, the tool is moved to the right of the programmed path to compensate for the size of the tool. A Dnn must also be programmed to select the correct tool size from the Tool Offset Register (D01 = Diameter Offset #1, D2 = #2, etc..)



**Dnn Cutter Compensation Value** – The actual offset amount must be input in the specified tool offset display number. On the HAAS you have 200 tool offsets to use. Usually, you will have only one cutter offset for a tool, and it is best to use the same cutter offset number as is the tool number. The cutter GEOMETRY column, in the offset display, is to set the initial cutter offset value. It can be designated as either a DIAMETER value or a RADIUS value by selecting the one you would like to use with SETTING 40. The selection chosen will be listed at the top of the offset geometry column. The WEAR column on the right of the tool GEOMETRY column is for any adjustments needed to make the initial tool GEOMETRY offset. These values are added together by the control and used as one value.

Understanding cutter compensation can be simplified if one has a basic understanding of manual machining. There are two common types of cutting conditions associated with milling machines:

- 1) **CLIMB**
- 2) **CONVENTIONAL**

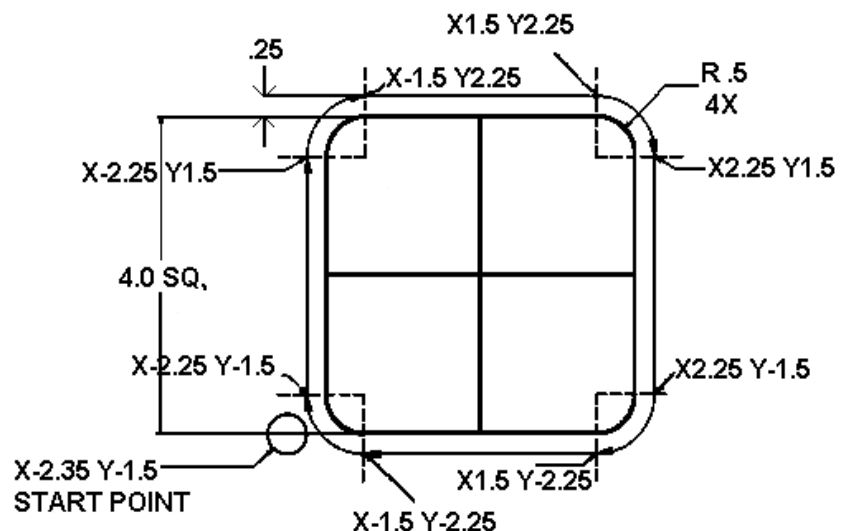
*Two common rules for these types of cuts are:*

If the programmed cutter path needs to mill **CLIMB** cutting and it is a standard right handed tool, it will then be programmed with **G41** cutter **LEFT** of the programmed path.

If the programmed cutter path needs to mill with **CONVENTIONAL** cutting and it is a standard right handed tool, it will then be programmed with the **G42** cutter **RIGHT** of the programmed path.

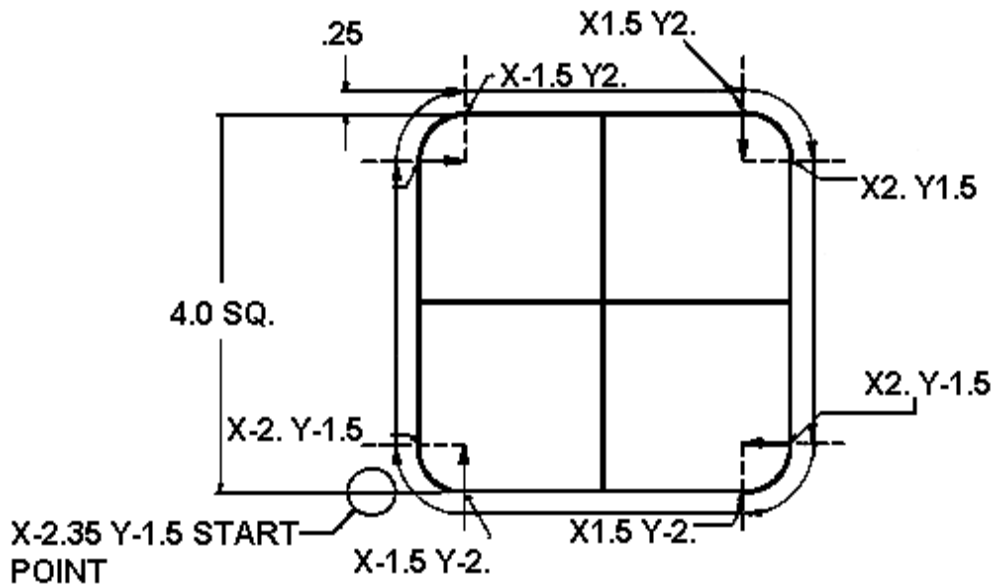
**Program without cutter compensation:**

```
G00 X-2.35 Y-2.
G01 Z-.45 F10
X-2.25
Y1.5
G02 X-1.5 Y2.25 R.75
G01 X1.5
G02 X2.25 Y1.5 R.75
G01 Y-1.5
G02 X1.5 Y-2.25 R.75
G01 X-1.5
G02 X-2.25 Y-1.5 R.75
G01 X-2.25
```

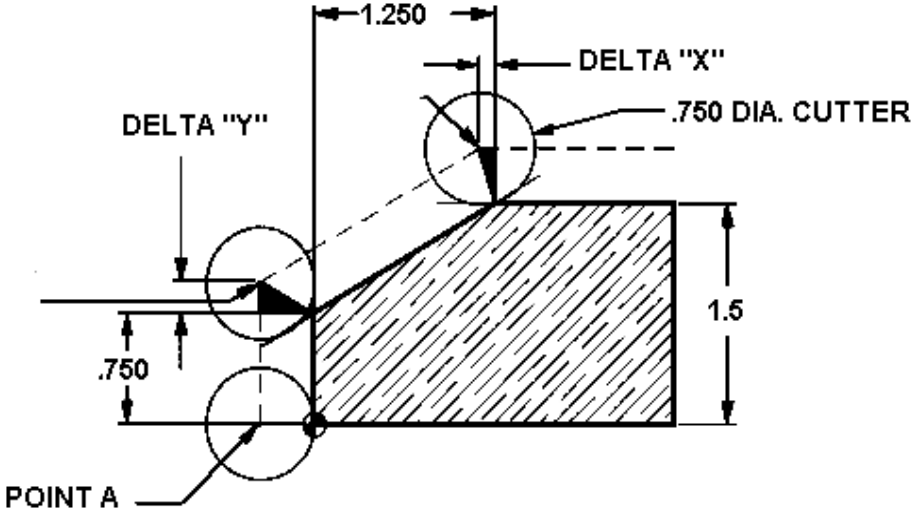


**Program with cutter compensation (Diameter value for D01 would be .500 entered into diameter offset register #1):**

```
G00 X-2.35 Y-2.0  
G01 Z-.45 F10  
G41 X-2. D01 (turn on CC with an X and/or Y move)  
Y1.5  
G02 X-1.5 Y2. R.5  
G01 X1.5  
G02 X2. Y1.5 R.5  
G01 Y-1.5  
G02 X1.5 Y-2. R.5  
G01 X-1.5  
G02 X-2. Y-1.5 R.5  
G40 G01 X-2.35 (turn off CC with an X and/or Y move off part)
```



When programming without cutter compensation to the center of the cutter, a problem occurs cutting angle geometry. The cutter center must be offset to the part geometry to maintain the cutter tangency. For example, the **Y-axis** move from Point **A** to Point **B** must have the **Delta "Y"** calculated dimension added to the .75 dimension. The **X-axis** move from Point **B** to Point **C** must have the **Delta "X"** calculated dimension subtracted from the 1.25 dimension.

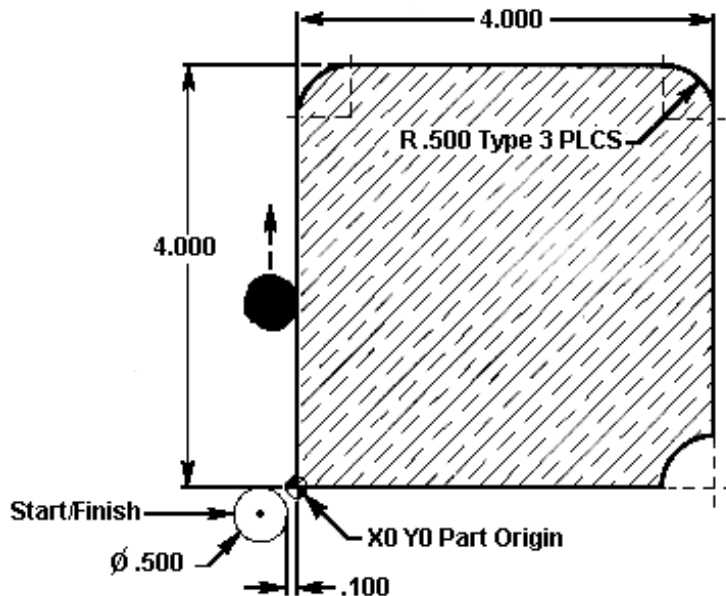






## Cutter Compensation Exercise #1

Cutter will be a .500 diameter end mill. Start contour from the lower left corner of part and mill around outside of part .375 deep. Mill around outside of part using cutter compensation for cutter offset and then define the actual geometry of the part to position tool around part.



Cutter will be a 1/2 diameter End Mill; Mill around outside contour of part .375 deep using Cutter Compensation.

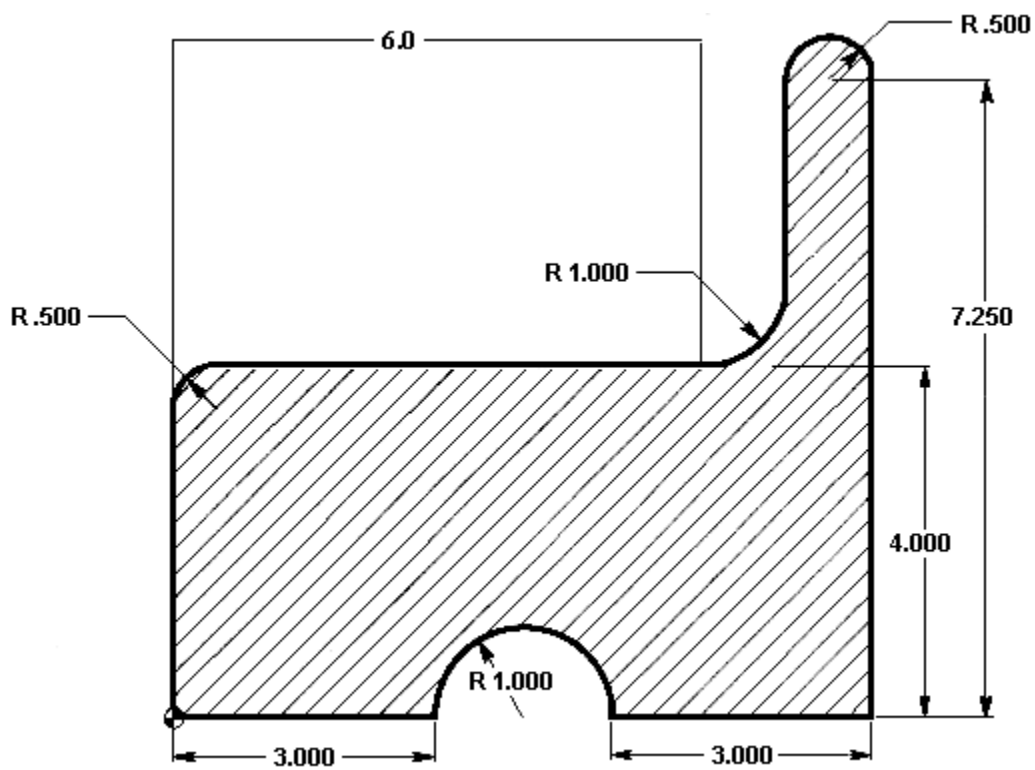
```

O0020
G00 G17 G40 G49 G80 G90 G98
G00 G53 Z0
T1 M06 (1/2 Diameter 4 flute. End Mill)
G90 G54 G00 X_____ Y_____ S1000 M03
G43 H01 Z.1 M08
G01 Z_____ F50.
G_ X_____ D_____ F10. (turn on cutter compensation)
  Y_____
G0_ X_____ Y_____ R_____
G0_ X_____
G0_ X_____ Y_____ R_____
G0_ Y_____
G0_ X_____ Y_____ R_____
G0_ X_____
G_ X_____ Y_____ (turn off cutter compensation)
G00 Z1. M09
G00 G53 Z0 M05
G00 G53 Y0
M30
  
```

## Cutter Compensation Exercise #2

To program this part profile, use the .5 diameter end mill starting at the lower left-hand corner. Use CUTTER RADIUS COMPENSATION. Climb cut the part profile, with one finish pass, at a depth of .450.

- TOOL NUMBER ONE** -  $\frac{1}{2}$  diameter 4 flute end mill
- surface speed (SFM) = 190 ft/min
  - chip load per flute = .005 inches/rev



## **00057 (CUTTER COMPENSATION EXERCISE)**

N1 G\_\_ G\_\_ G\_\_ G\_\_ G\_\_ G\_\_ G\_\_ G\_\_ (SAFETY LINE)

N2 G00 G53 Z0

N3 T01 M\_\_ (TOOL CHANGE)

N4 G\_\_ G\_\_ G\_\_ X-.35 Y-.25 M\_\_ S\_\_

N5 G\_\_ Z.\_\_\_\_ H\_\_ M\_\_

N6 G\_\_ Z-.\_\_\_\_ F\_\_

N7 G4\_ X\_\_\_\_ D\_\_

N8 Y\_\_\_\_

N9 G\_\_ X.\_\_\_\_ Y\_\_\_\_ R\_\_\_\_

N10 G\_\_ X\_\_\_\_

N11 G\_\_ X\_\_\_\_ Y\_\_\_\_ R\_\_\_\_

N12 G\_\_ Y\_\_\_\_

N13 G\_\_ X\_\_\_\_ Y\_\_\_\_ R\_\_\_\_

N14 G\_\_ Y\_\_\_\_

N15 X\_\_\_\_

N16 G\_\_ X\_\_\_\_ Y\_\_\_\_ R\_\_\_\_

N17 G\_\_ X-.\_\_\_\_

N18 G4\_ X-.\_\_\_\_ Y-.\_\_\_\_

N19 G\_\_ Z\_\_ M\_\_

N20 G\_\_ G\_\_ Z\_\_\_\_

N21 G\_\_ G\_\_ Y\_\_\_\_

N22 M\_\_

## ***Formulas – Tapping, Speeds and Feeds***

### ***Tapping***

STANDARD thread formula:

Feed rate in inches per minute = Revolutions per minute (RPM) divided by threads per inch (TPI)

$$\mathbf{F = RPM/TPI}$$

METRIC thread formula:

Feed rate in inches per minute = Pitch (P) multiplied by .03937 multiplied by RPM

$$\mathbf{F = (P \times .03937) \times RPM}$$

### ***Speeds and Feeds***

S.F.M. (Surface Feet per Minute):

SFM = .262 multiplied by the cutter diameter multiplied by the RPM

$$\mathbf{SFM = .262 \times \text{Cutter Diameter} \times RPM}$$

R.P.M. (Revolutions per Minute):

RPM = 3.82 multiplied by the recommended SFM divided by the cutter diameter

$$\mathbf{RPM = 3.82 \times SFM / \text{Cutter Diameter}}$$

Feed (Inch per Minute) for twist drills:

$$\mathbf{F(\text{inch/min}) = F(\text{inch /rev}) \times RPM}$$

Feed (Inch per Minute) for end mills:

Feed rate in inches per minute = Feed per tooth (Inch/rev) multiplied by the number of cutter teeth multiplied by the RPM =

$$\mathbf{F (\text{inch/min}) = (\text{Feed/tooth} \times n) \times RPM}$$

CUBIC INCH PER MINUTE:

Cubic inch per minute = Effective diameter of cut multiplied by the depth of cut multiplied by the inch per minute feed rate.

$$\mathbf{CIPM = (E \text{ Diameter} \times d) \times IPM}$$

## ***Drilling, Tapping, Boring Canned Cycles***

Early on in NC, if we needed to drill a deep hole and had to “peck” our drill to get it to depth, we had to program each peck with 3 lines of code. A feed move in, a rapid move out, & a rapid move back in to the hole.

If we had to make 20 pecks in a hole, it would take 60 lines of code. Again, with the acceleration of computer technology, we were able to take all of this and condense it to one line of code, this is referred to as a canned cycle.

### ***Objectives:***

Upon completion of this unit, the student will:

- 1) Be able to define “canned cycle.”
- 2) Be familiar with common canned cycles, their variables, the proper applications, and the correct deployment of them.
- 3) Understand the usage of the “L” variable to loop canned cycles.

## *Canned Cycles*

A canned cycle is used to simplify programming of a part. Canned cycles are defined for the most common Z-axis repetitive operations such as drilling, tapping, and boring. There are 13 canned cycles to choose from. Once selected, a canned cycle is active until canceled with G80.

There are five operations involved in every canned cycle:

- 1) positioning of X and Y axes (and optional A),
- 2) rapid traverse to R plane,
- 3) drilling, tapping and boring,
- 4) operation at bottom of hole,
- 5) retraction to R plane.

Once a Canned Cycle code is turned on, every time we move to a new X & Y location the machine will drill a hole. This will continue until we cancel the drilling with a G80 code. This is how canned cycles save time. If we had 100 holes in a part, we would save 6000 lines of code in the example above.

Canned cycles are presently limited to only operations in the Z-axis. That is, only the G17 X-Y plane is allowed. This means that the canned cycle will be executed in the **Z**-axis whenever a position is selected in the **X** or **Y** axes. The operation of a canned cycle will vary according to whether incremental (G91) or absolute (G90) is active.

G98 and G99 are modal commands that change the way canned cycles operate. When G98 is active, the Z-axis will return to the start position (initial plane) when it completes a single operation. When G99 is active, the Z-axis will be returned to the **R** point (plane) when the canned cycle completes a single hole. Then the machine will go to the next hole.

**The following is a summary of the canned cycles defined for the VF Series Mill:**

G Code	Z Drilling Direction	Operation at Bottom of Hole	Retraction Z Direction	Application
G73	Intermittent Feed	None	Rapid	High Speed Peck Drilling
G74	Feed	Spindle CW	Feed	Left Hand Tapping
G76	Feed then Stop	Orient Spindle	Rapid	Fine Boring
G77	Rapid to Bottom	Orient Spindle	Back Bore/Rapid	Back Boring
G81	Feed	None	Rapid	Spot Drilling
G82	Feed	Dwell	Rapid	Counter Boring
G83	Intermittent Feed	None	Rapid	Peck Drilling
G84	Feed	Spindle CCW	Feed	Tapping Cycle
G85	Feed	None	Feed	Boring Cycle
G86	Feed	Spindle Stop	Rapid	Boring Cycle
G87	Feed	Spindle Stop	Manual/Rapid	Back Boring
G88	Feed	Dwell, then Spindle Stop	Manual/Rapid	Boring Cycle
G89	Feed	Dwell	Feed	Boring Cycle

Incremental motion in a canned cycle is often useful as a loop (L) count, which can be used to repeat the operation with an incremental X or Y move between each cycle.

**NOTE:** If an L0 is in the canned cycle line, the cycle will not execute until the control reads the next X or Y location.

The G80 code is used to cancel a canned cycle. In addition to this, a G00 or G01 code will also cancel any active canned cycle.

Once a canned cycle is defined, the canned cycle is repeated at every X-Y position in sequential blocks. Some of the canned cycle numerical values can also be changed after the canned cycle is defined. The most important of these are the R plane value and the Z depth value.

If these are changed in a block with an X-Y, the change doesn't take effect until the next block. All subsequent blocks are performed with the new R or Z value.

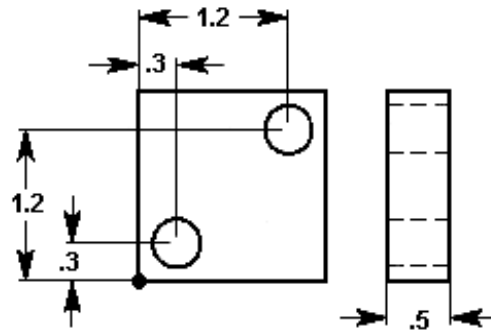
Changes to the G98/G99 selection can also be made after the canned cycle is active. If changed, the new G98/G99 value will be active in the next block and all subsequent blocks.

## G80 Canned Cycle Cancel

This **G** code is modal in that it deactivates all canned cycles **G73, G74, G76, G77, or G81-G89** until a new one is selected. Note that the use of **G00** or **G01** will also cancel a canned cycle.

## G81 Drill Canned Cycle – Straight Drilling without Peck

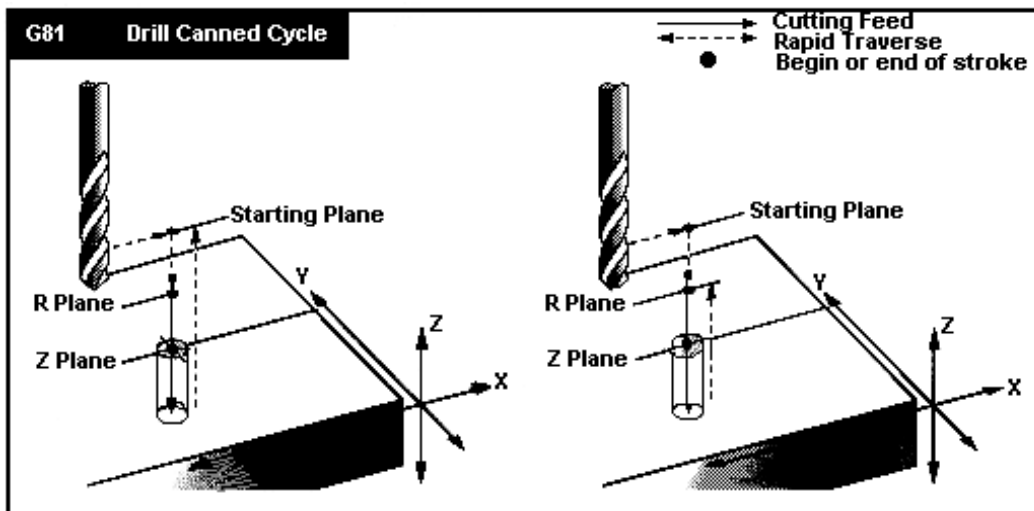
- F Feed Rate in inches (mm) per minute
- R Position of the R plane
- X Optional X-axis motion command
- Y Optional Y-axis motion command
- Z Position of bottom of hole



```

N1 T1 M06
N2 G90 G54 G00 X.3 Y.3
N3 S1200 M03
N4 G43 H01 Z1. M08
N5 G81 Z-.6 R.1 F10.
N6 X1.2 Y1.2
N7 G80 G00 Z1. M09
N8 G28 G91 Z0. M05
N9 M30
    
```

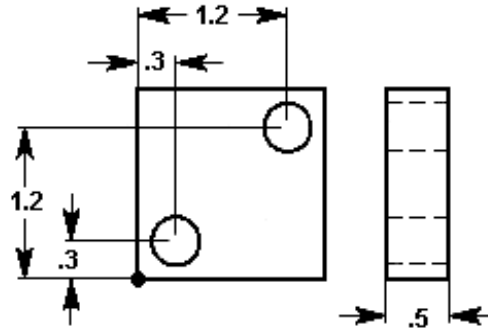
This **G** code is modal in that it activates the canned cycle until it is canceled or another canned cycle is selected. Once activated, every motion of **X** or **Y** will cause this canned cycle to be executed.





## G82 Spot Drill Canned Cycle – Spot Drilling / Dwell

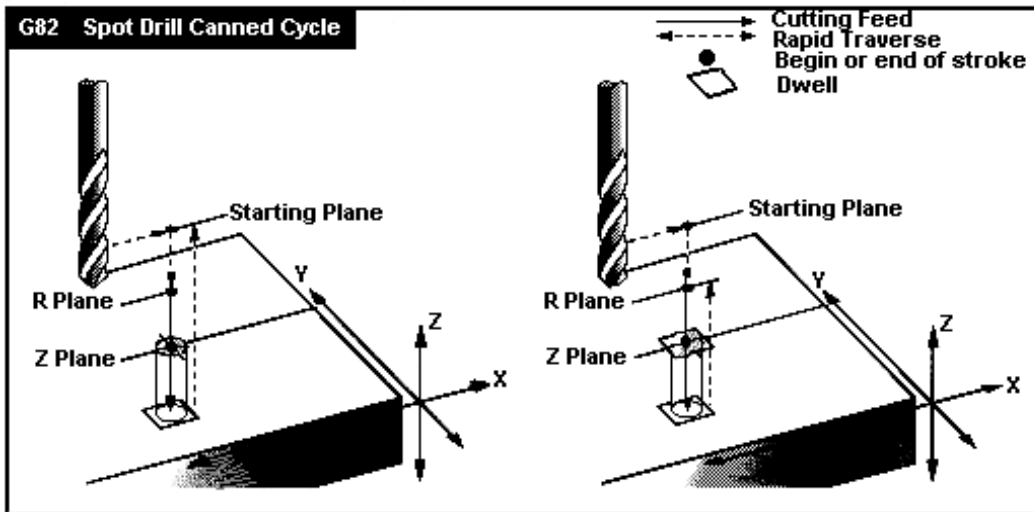
- F Feed Rate in inches (mm) per minute
- P The dwell time at the bottom of the hole in seconds
- R Position of the R plane
- X Optional X-axis motion command
- Y Optional Y-axis motion command
- Z Position of bottom of hole



```

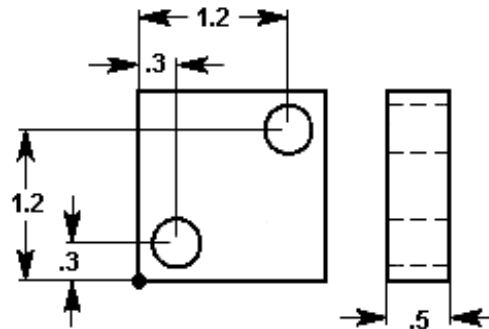
N1 T1 M06
N2 G90 G54 G00 X.3 Y.3
N3 S1200 M03
N4 G43 H01 Z1. M08
N5 G82 Z-.125 P1.5 R.1 F10.
N6 X1.2 Y1.2
N7 G80 G00 Z1. M09
N8 G91 G28 Z0. M05
N9 M30
    
```

This G code is modal in that it activates the canned cycle until it is canceled or another canned cycle is selected. Once activated, every motion of X or Y will cause this canned cycle to be executed.



## G83 Peck Drilling Canned Cycle – Deep Hole Peck

F	Feed Rate in inches (mm) per minute
I	Optional size of first cutting depth
J	Optional amount to reduce cutting depth each pass
K	Optional minimum depth of cut
Q	The cut-in value, always incremental
R	Position of the R plane
X	Optional X-axis motion command
Y	Optional Y-axis motion command
Z	Position of bottom of hole



```
N1 T1 M06
N2 G90 G54 G00 X.3 Y.3
N3 S1200 M03
N4 G43 H01 Z1. M08
N5 G83 Z-.6 Q.10 R.1 F10.
N6 X1.2 Y1.2
N7 G80 G00 Z1. M09
N8 G91 G28 Z0. M05
N9 M30
```

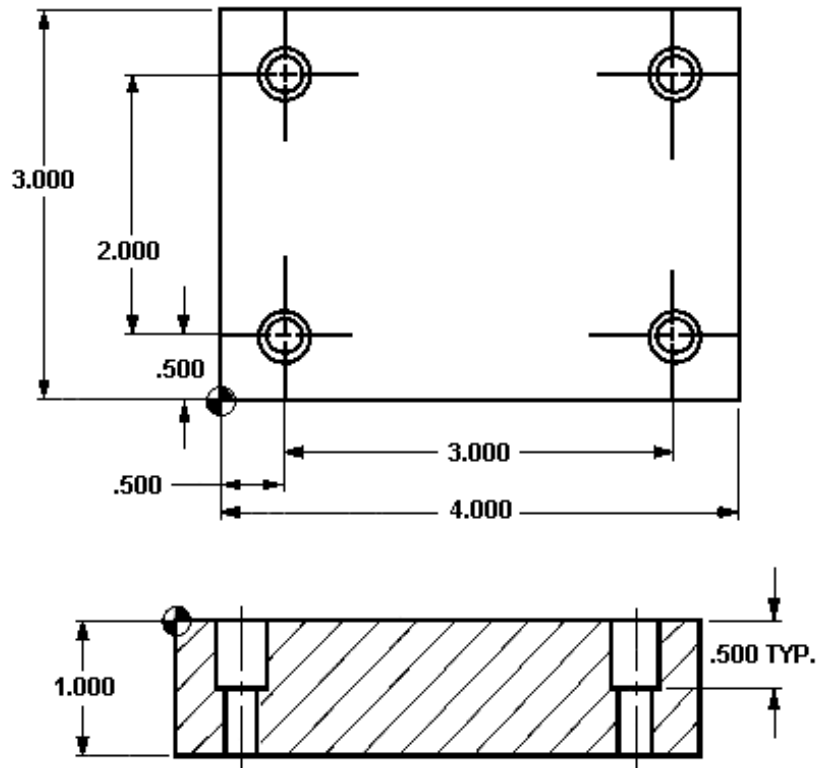
This G code is modal in that it activates the canned cycle until it is canceled or another canned cycle is selected. Once activated, every motion of X or Y will cause this canned cycle to be executed.

If I, J, and K are specified, a different operating mode is selected: 1) The first pass will cut in by I; and 2) each succeeding cut will be reduced by amount J with a minimum cutting depth of K.

If Q is specified, the machine will incrementally peck till full depth is achieved.



## Canned Cycle Exercise #1



### TOOL NUMBER ONE

- $\frac{1}{2}$  Diameter 90 Degree Spot Drill (use G81 cycle)
- surface speed (SFM) = 190 ft/min
- feed per. rev. = .0035 in/rev
- Drill to depth that will put .01 break on 3/8 counter bore

### TOOL NUMBER TWO

- $\frac{1}{4}$  Diameter Drill, standard 118 degree (use G83 cycle)
- surface speed (SFM) = 190 ft/min
- feed per. rev. = .006 in/rev
- Drill deep enough so flute goes .02 below bottom of part

### TOOL NUMBER THREE

- $\frac{3}{8}$  Diameter End Mill, 2 flute (use G82 cycle)
- surface speed (SFM) = 190 ft/min
- feed per. rev, per tooth. = .006 in/rev-tooth

**00067 (CANNED CYCLE EXERCISE)**

G\_\_G\_\_G\_\_G\_\_G\_\_G\_\_G\_\_G\_\_ (SAFETY LINE)

N1 (1/2 DIA. SPOT DRILL)

G\_\_G\_\_Z\_\_

T\_\_M\_\_

G\_\_G\_\_G\_\_X.\_\_\_Y.\_\_\_M\_\_S\_\_

G\_\_Z\_\_H\_\_M\_\_

G\_\_Z-.\_\_\_R\_\_F\_\_

Y\_\_

X\_\_

Y\_\_

G\_\_G\_\_Z\_\_M\_\_

G\_\_G\_\_Z\_\_

M\_\_

N2 (1/4 DIA. DRILL)

T\_\_M\_\_

G\_\_G\_\_G\_\_X\_\_Y\_\_M\_\_S\_\_

G\_\_Z\_\_H\_\_M\_\_

G\_\_Z-\_\_R\_\_F\_\_Q\_\_

Y\_\_

X\_\_

Y\_\_

G\_\_G\_\_Z\_\_M\_\_

G\_\_G\_\_Z\_\_

M\_\_

N3 (3/8 DIA. ENDMILL)

T\_\_M\_\_

G\_\_G\_\_G\_\_X\_\_Y\_\_M\_\_S\_\_

G\_\_Z\_\_H\_\_M\_\_

G\_\_Z-\_\_R\_\_F\_\_P\_\_

Y\_\_

X\_\_

Y\_\_

G\_\_G\_\_Z\_\_M\_\_

G\_\_G\_\_Z\_\_

G\_\_G\_\_Y\_\_

M\_\_

## G84 Tapping Canned Cycle – Tapping with RH Threads

## G74 Tapping Canned Cycle – Tapping with LH Threads

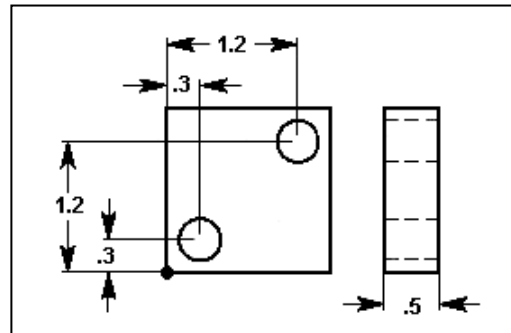
You do not need to turn on the spindle for a tap when using G84 because G84 will do it for you. G84 or G74 do however require a (S) spindle speed value described in the code.

F	Feed Rate in inches (mm) per minute
R	Position of the R plane
X	Optional X-axis motion command
Y	Optional Y-axis motion command
Z	Position of bottom of hole

$$Feed \left( \frac{in}{min} \right) = \frac{Speed \left( \frac{rev}{min} \right)}{TPI}$$

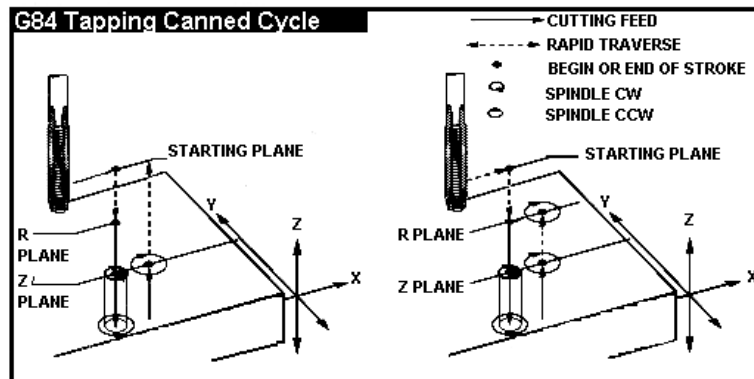
```

N1 T1 M06
N2 G90 G54 G00 X.3 Y.3
N3 S1200 (you do not need to turn spindle on)
N4 G43 H01 Z1. M08
N5 G84 Z-.85 R.1 F60.
N6 X1.2 Y1.2
N7 G80 G00 Z1. M09
N8 G91 G28 Z0. M05
N9 M30
    
```



This G code is modal in that it activates the canned cycle until it is canceled or another canned cycle is selected. Once activated, every motion of X or Y will cause this canned cycle to be executed. When rigid tapping is used, the ratio between the feed rate and spindle speed must be precisely the thread pitch being cut.

You do not need to start the spindle CW before this canned cycle. The control does this automatically.

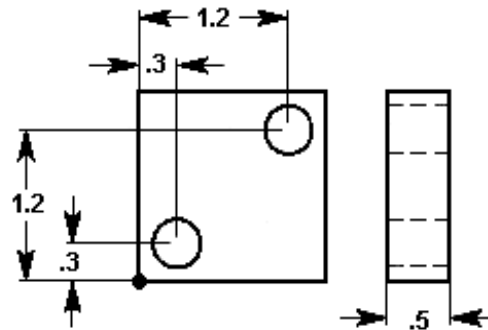


**Setting 130 Tap Retract Speed** (Allows retract to be two to four times faster)

**Setting 133 Repeat Rigid Tap** (Allows holes to be re-tapped)

## G85 Boring Canned Cycle - Feed Down and Feed Up

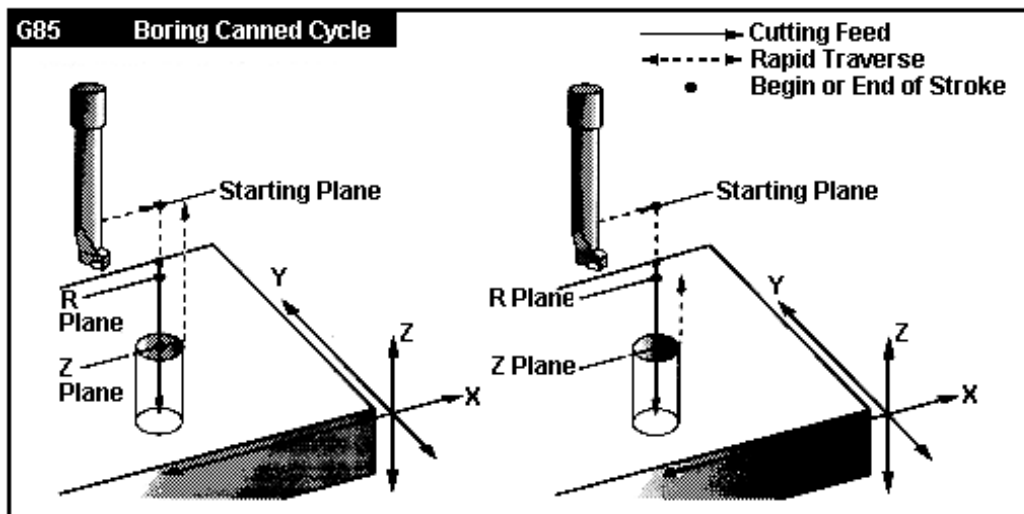
F Feed Rate in inches (mm) per minute  
 R Position of the R plane  
 X Optional X-axis motion command  
 Y Optional Y-axis motion command  
 Z Position of bottom of hole



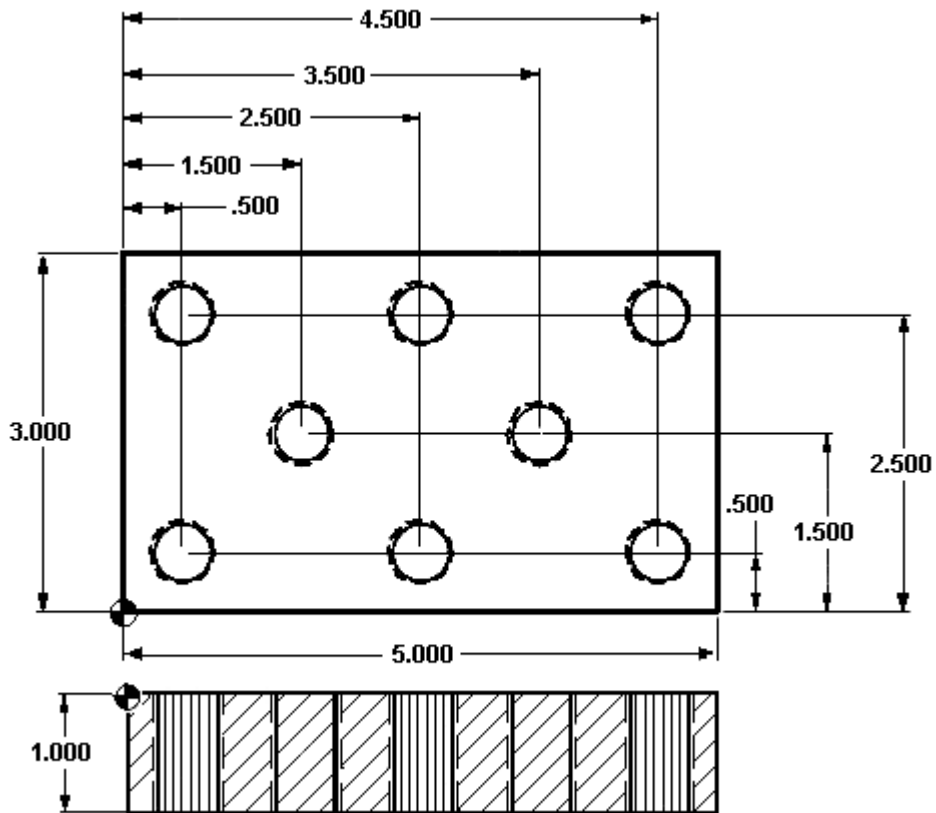
```

N1 T1 M06
N2 G90 G54 G00 X.3 Y.3
N3 S1200 M03
N4 G43 H01 Z1. M08
N5 G85 Z-.6 R.1 F10.
N6 X1.2 Y1.2
N7 G80 G00 Z1. M09
N8 G91 G28 Z0. M05
N9 M30
  
```

This G code is modal in that it activates the canned cycle until it is canceled or another canned cycle is selected. Once activated, every motion of X or Y will cause this canned cycle to be executed.



## Canned Cycle Exercise #2



### TOOL NUMBER ONE

- 5/8 Diameter 90 degree Spot Drill (use G81 cycle)
- surface speed (SFM) = 190 ft/min
- feed per. rev. = .0035 in/rev
- feed deep enough to leave .02 break on ½ tapped hole

### TOOL NUMBER TWO

- 27/64 Diameter Drill, 118° (use G73 canned cycle)
- surface speed (SFM) = 190 ft/min
- feed per. rev. = .006

### TOOL NUMBER THREE

- ½-13 Tap (use G84 cycle)
- 600 RPM



**00071 (CANNED CYCLE EXERCISE)**

N1 (5/8 DIA. SPOT DRILL)

G00 G17 G40 G49 G80 G90G98

G\_\_ G\_\_ Z\_\_

T\_\_ M\_\_

G\_\_ G\_\_ G\_\_ X.\_\_\_\_ Y.\_\_\_\_ M\_\_ S\_\_\_\_

G\_\_ Z\_\_ H\_\_ M\_\_

G\_\_ Z-\_\_\_\_ R\_\_ F\_\_

X\_\_\_\_

X\_\_\_\_

X\_\_\_\_ Y\_\_\_\_

X\_\_\_\_

X\_\_\_\_ Y\_\_\_\_

X\_\_\_\_

X\_\_\_\_

G\_\_ G\_\_ Z\_\_ M\_\_

G\_\_ G\_\_ Z\_\_

M\_\_

N2 (27/64 DIA. DRILL)

T\_\_ M\_\_

G\_\_ G\_\_ G\_\_ X\_\_\_\_ Y\_\_\_\_ M\_\_ S\_\_\_\_

G\_\_ Z\_\_ H\_\_ M\_\_

G\_\_ Z-\_\_\_\_ R\_\_ F\_\_ Q\_\_\_\_

X\_\_\_\_

X\_\_\_\_

X\_\_\_\_ Y\_\_\_\_

X\_\_\_\_

X\_\_\_\_ Y\_\_\_\_

X\_\_\_\_

X\_\_\_\_

G\_\_ G\_\_ Z\_\_ M\_\_

G\_\_ G\_\_ Z\_\_

M\_\_

N3 (1/2 - 13 TAP)

T\_\_\_ M\_\_\_

G\_\_\_ G\_\_\_ G\_\_\_ X\_\_\_ Y\_\_\_ S\_\_\_

G\_\_\_ Z\_\_\_ H\_\_\_ M\_\_\_

G\_\_\_ Z-\_\_\_ R\_\_\_ F\_\_\_

X\_\_\_

X\_\_\_

X\_\_\_ Y\_\_\_

X\_\_\_

X\_\_\_ Y\_\_\_

X\_\_\_

X\_\_\_

G\_\_\_ G\_\_\_ Z\_\_\_ M\_\_\_

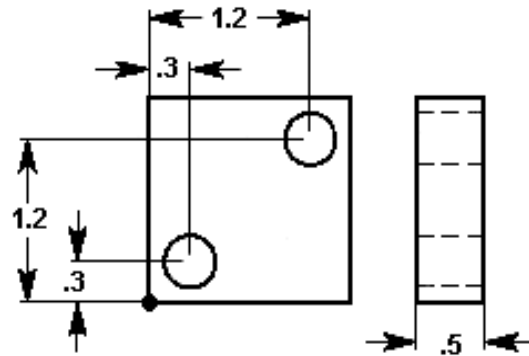
G\_\_\_ G\_\_\_ Z\_\_\_

G\_\_\_ G\_\_\_ Y\_\_\_

M\_\_\_

### G89 Bore Canned Cycle (feed down, dwell, jog out)

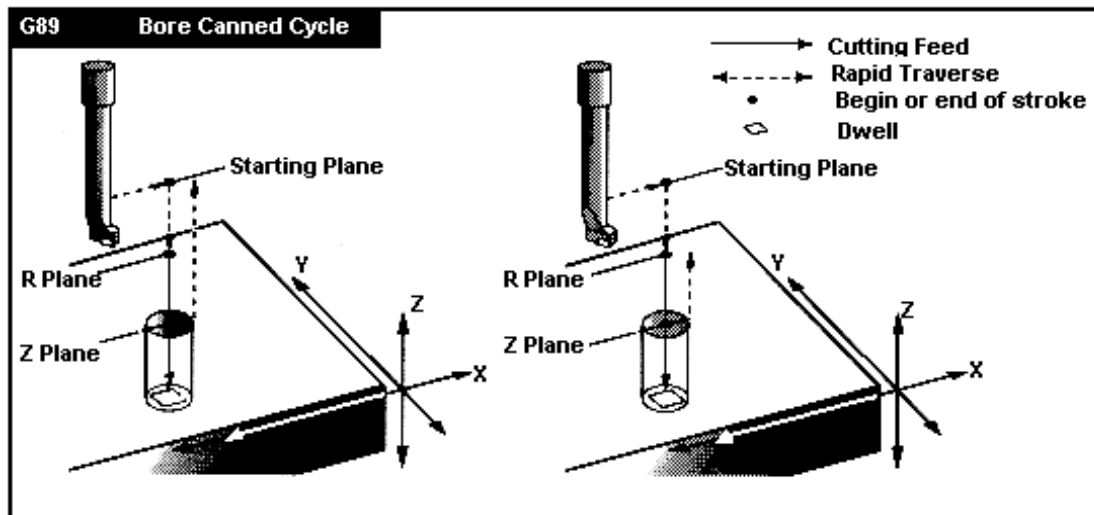
- F Feed Rate in inches (mm) per minute
- P The dwell time at the bottom of the hole
- R Position of the R plane
- X Optional X-axis motion command
- Y Optional Y-axis motion command
- Z Position of bottom of hole



```

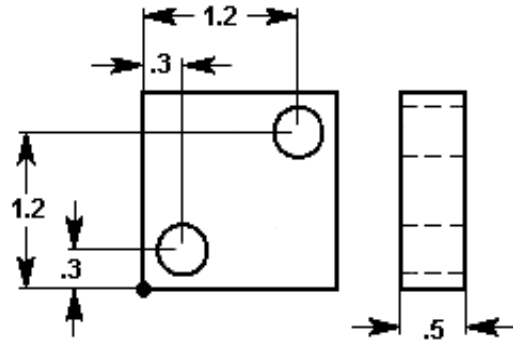
N1 T1 M06
N2 G90 G54 G00 X.3 Y.3
N3 S1200 M03
N4 G43 H01 Z1. M08
N5 G89 Z-.6 P1.0 R.1 F10.
N6 X1.2 Y1.2
N7 G80 G00 1Z M09
N8 G91 G28 Z0. M05
N9 M30
    
```

This G code is modal in that it activates the canned cycle until it is canceled or another canned cycle is selected. Once activated, every motion of X or Y will cause this canned cycle to be executed.



## G73 High Speed Peck Drilling Canned Cycle

F	Feed Rate in inches (mm) per minute
I	Optional size of first cutting depth
J	Optional amount to reduce cutting depth each pass
K	Optional minimum depth of cut
Q	The cut-in value, always incremental
R	Position of the R plane
X	Optional X-axis motion command
Y	Optional Y-axis motion command
Z	Position of bottom of hole



```
N1 T1 M06
N2 G90 G54 G00 X.3 Y.3
N3 S1200 M03
N4 G43 H01 Z1. M08
N5 G73 Z-.6 Q.25 R.1 F10.
N6 X1.2 Y1.2
N7 G80 G00 Z1. M09
N8 G91 G28 Z0. M05
N9 M30
```

This G code is modal in that it activates the canned cycle until it is canceled or another canned cycle is selected. Once activated, every motion of X or Y will cause this canned cycle to be executed. This cycle is a high speed peck cycle where a retract distance is set by **Setting 22**.

I, J, K, and Q are always positive numbers.

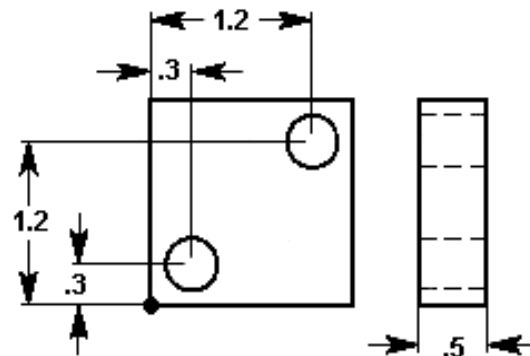
Setting 52 also changes the way G73 works when it returns to the R plane. Most programmers set the R plane well above the cut to ensure that the chip clear motion actually allows the chips to get out of the hole, but this causes a wasted motion when first drilling through this “empty” space. If Setting 52 is set to the distance required to clear chips, the R plane can be put much closer to the part being drilled. When the clear move to R occurs, the Z will be moved above R by this setting.

If I, J, and K are specified with G73, a different operating mode is selected. The first pass will cut in by I, each succeeding cut will be reduced by amount J with a minimum cutting depth of K.

If K and Q are both specified, a different operating mode is selected for this canned cycle. In this mode, the tool is returned to the R plane after a number of passes totals up to the K amount. This allows much faster drilling than G83, but still returns to the R plane occasionally to clear chips.

## G74 Reverse Tap Canned Cycle

- F Feed Rate in inches (mm) per minute
- R Position of the R plane
- X Optional X-axis motion command
- Y Optional Y-axis motion command
- Z Position of bottom of hole

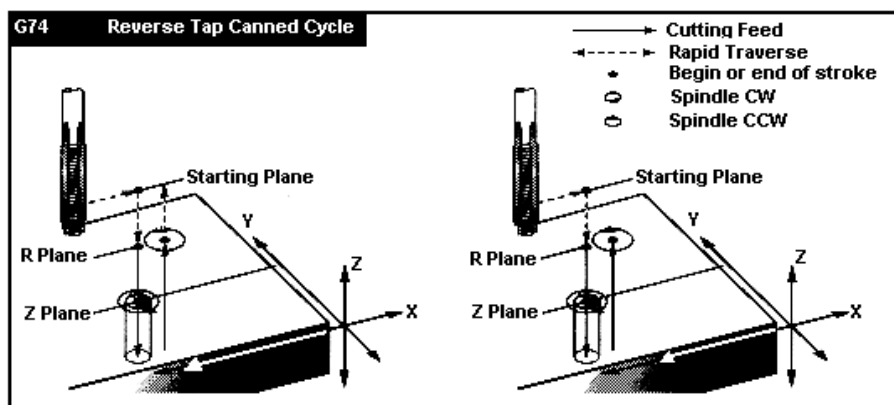


```

N1 T1 M06
N2 G90 G54 G00 X.3 Y.3
N3 S1200
N4 G43 H01 Z1. M08
N5 G74 Z-.6 R.1 F60.
N6 X1.2 Y1.2
N7 G80 G00 Z1. M09
N8 G91 G28 Z0. M05
N9 M30
    
```

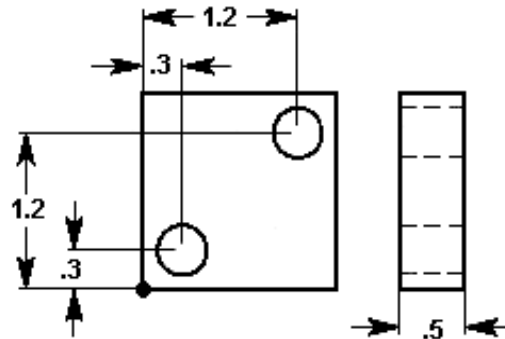
This G code is modal in that it activates the canned cycle until it is canceled or another canned cycle is selected. Once activated, every motion of X or Y will cause this canned cycle to be executed. Note that operation of this cycle is different if the rigid tapping option is installed and selected. When rigid tapping is used, the ratio between the feed rate and spindle speed must be precisely the thread pitch being cut.

You do not need to start the spindle CCW before this canned cycle. The control does this automatically.



**G76 Fine Boring Canned Cycle (feed down, spindle stop, orient spindle, step over, rapid out)**

- F Feed Rate in inches (mm) per minute
- I Optional shift value (in x-axis), if Q is not specified
- J Optional shift value (in y-axis), if Q is not specified.
- P The dwell time at the bottom of the hole
- Q The shift value, always incremental
- R Position of the R plane
- X Optional X-axis motion command
- Y Optional Y-axis motion command
- Z Position of bottom of hole

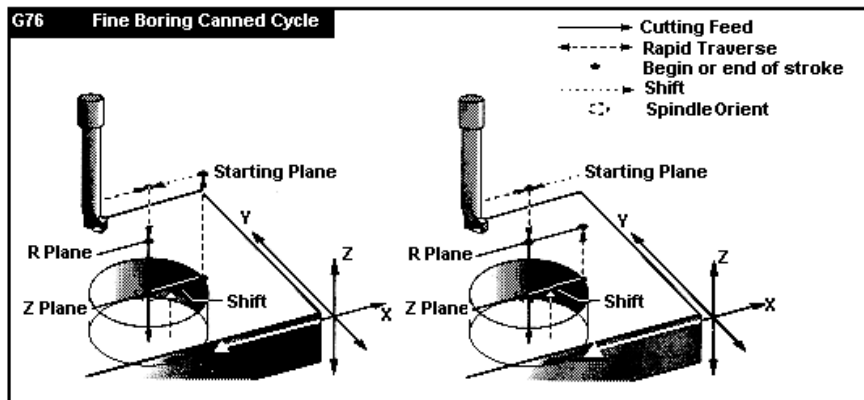


```

N1 T1 M06
N2 G90 G54 G00 X.3 Y.3
N3 S1200 M03
N4 G43 H01 Z1. M08
N5 G76 Z-.6 R.1 F10. Q.02
N6 X1.2 Y1.2
N7 G80 G00 Z1. M09
N8 G91 G28 Z0. M05
N9 M30
    
```

This cycle will shift the X and/or Y-axis prior to retracting in order to clear the tool while exiting the part. This shift direction is set by **Setting 27**.

The Q value distance will shift in the direction set by Setting 27. If Q is not specified, the optional I and J values are used to determine the shift direction and distance.



## ***CANNED CYCLE AUXILIARY FUNCTIONS***

### **G98 Canned Cycle Initial Point Return**

This G code is modal and changes the way canned cycles operate. With G98, the canned cycle will return to the initial starting point of the canned cycle when it completes a hole before moving to the next hole.

### **G99 Canned Cycle R Plane Return**

This G code is modal and changes the way canned cycles operate. With G99, the canned cycle will return to the R plane when the canned cycle completes before moving to the next hole.

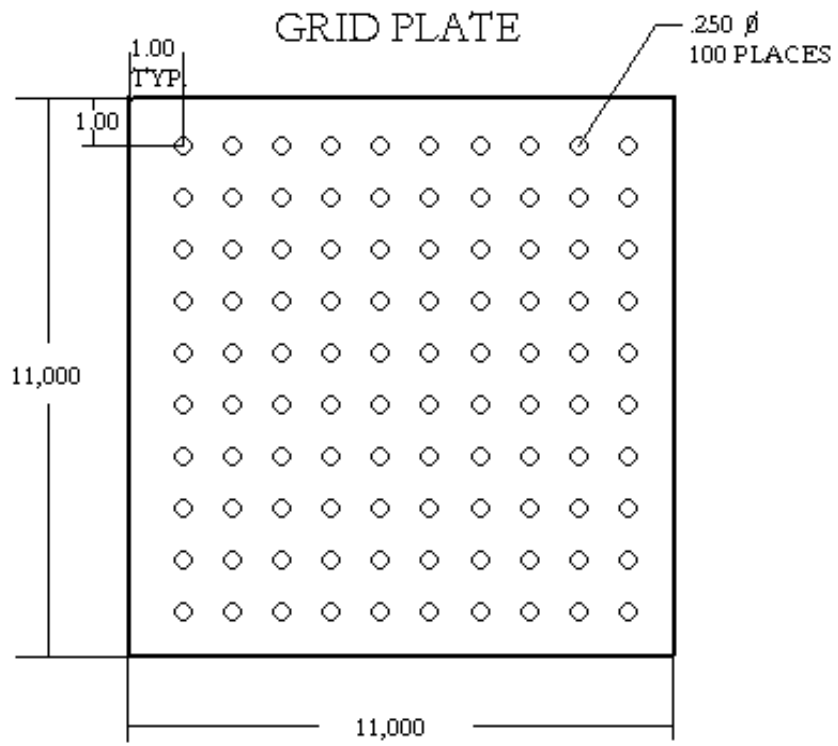
## *Looping Command Cycles*

Below is an example of a program using a drilling canned cycle that is incrementally looped. Compare the grid plate drawing on the following page to the program below.

```
%  
O3400                                (Drilling grid plate)  
T1 M06  
G00 G90 G54 X1.0 Y-1.0 S2500 M03  
G43 H01 Z.1 M08  
G81 Z-1.5 F15. R.1  
G91 X1.0 L9  
G90 Y-2.0                                (Or stay in G91 and repeat Y-1.0)  
G91 X-1.0 L9  
G90 Y-3.0  
G91 X1.0 L9  
G90 Y-4.0  
G91 X-1.0 L9  
G90 Y-5.0  
G91 X1.0 L9  
G90 Y-6.0  
G91 X-1.0 L9  
G90 Y-7.0  
G91 X1.0 L9  
G90 Y-8.0  
G91 X-1.0 L9  
G90 Y-9.0  
G91 X1.0 L9  
G90 Y-10.0  
G91 X-1.0 L9  
G00 G90 G80 Z1.0 M09  
G28 G91 Y0 Z0  
M30  
%
```

**NOTE:** The sequence of drilling used here is designed to save time and to follow the shortest path from hole-to-hole.





## ***Bolt Hole Patterns***

There are three G codes that provide patterns usually used for bolt holes. These are G70, G71, and G72. These G codes must be used with one of the canned cycles G73, G74, G76, G77, or G81-G89. The starting angle is from 0 to 360.0 degrees CCW from horizontal starting from Three o'clock.

### **G70 Bolt Hole Circle**

I	Radius (Minus Reverses Starting Position)
J	Starting angle (0 to 360.0 degrees CCW from horizontal)
L	Number of holes evenly spaced around the circle

The tool must be positioned at the center of the circle either in a previous block or in the G70 block. G70 belongs to Group zero and thus is non-modal. For a G70 to work correctly, a canned cycle should be active so that at each of the positions, some type of drill or tap cycle is performed.

### **G71 Bolt Hole Arc**

I	Radius
J	Starting angle (degrees CCW from horizontal)
K	Angular spacing of holes (+ or -)
L	Number of holes)

This G code is similar to G70 except that it is not limited at one complete circle. G71 belongs to Group zero and thus is non-modal. For a G71 to work correctly, a canned cycle should be active so that at each of the positions, some type of drill or tap cycle is performed.

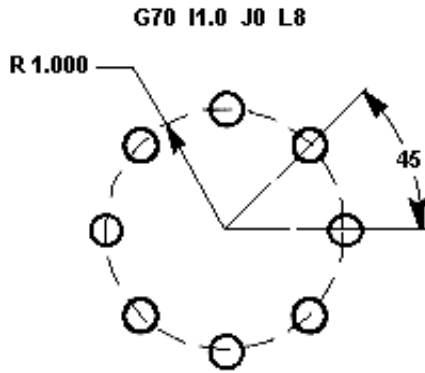
### **G72 Bolt Holes Along An Angle**

This G code drills L holes in a straight line at the specified angle. It operates similarly to G70 and G71. G72 belongs to Group zero and thus is non-modal. For a G72 to work correctly, a canned cycle should be active so that at each of the positions, some type of drill or tap cycle is performed.

Example of G70 Bolt Hole Circle program used in conjunction with G83 Peck Drill Cycle.

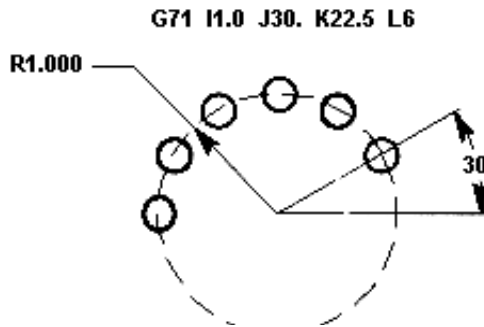
```
%  
O100  
T1 M6  
G0 G90 G54 X0 Y0 S1000 M3  
G43 H1 Z1.0 M8  
G83 Z-2.0 R.1 Q.2 F1.5 L0  
G70 I1.0 J0 L8  
G00 G80 Z1.0 M9  
G53 Z0  
G53 Y0  
M30  
%
```

# BOLT HOLE PATTERNS



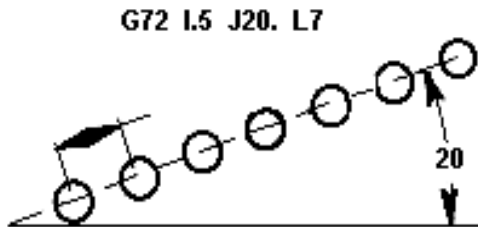
**G70**  
**BOLT HOLE CIRCLE**

I = Radius of the bolt circle  
J = Starting angle from the three o'clock position  
L = Number of holes (evenly spaced)



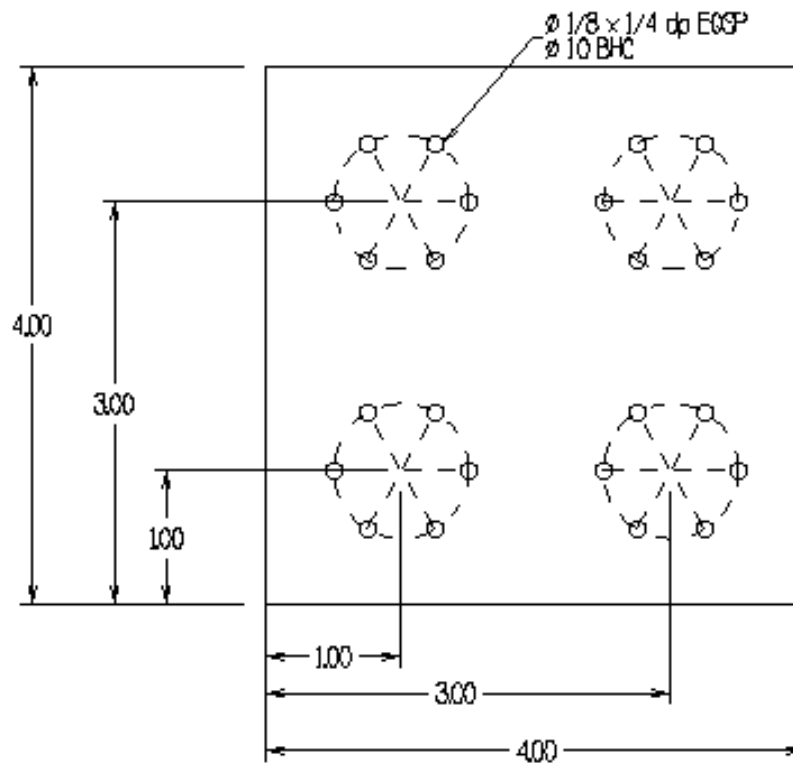
**G71**  
**BOLT HOLES ARC**

I = Radius of the bolt circle  
J = Starting angle from three o'clock position  
K = Angular spacing between holes  
L = Number of holes (evenly spaced)



**G72**  
**BOLT HOLES ALONG AN ANGLE**

I = Distance between holes  
J = Angle from three o'clock position  
L = Number of holes (evenly spaced)



**Four Bolt Hole Circle Exercise: Drill 1/8" holes 1/4" deep in .5 thick plate. Spot drill deep to leave .012 chamfer on 1/8" holes.**

T1 - 1/4" 90 DEG SPOT DRILL 80 FT/MIN .003 IN/REV  
 T2 - 1/8" CONVENTIONAL DRILL 80 FT/MIN ., .003 IN/REV

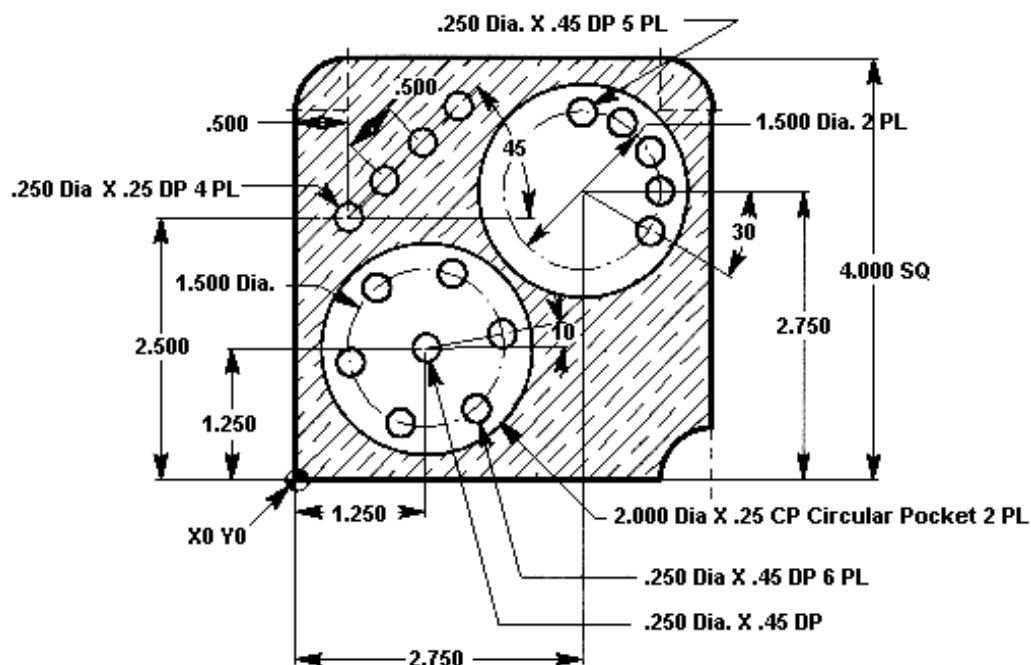
```

O03901 (1" Ø BHC)
N1 (1/2 DIAM SPOT DRILL)
G00 G17 G40 G49 G80 G90 G98
G00 G53 Z0
T1 M06 (1/4 90 DEG SPOT DRILL)
G00 G90 G54 X_____ Y_____ M03 S_____
G43 H01 Z1. M08
G__ Z-_____ R___ F___ L___
G__ I___ J___ L___
X___ Y___ L___
G__ I___ J___ L___
X___ Y___ L___
G__ I___ J___ L___
X___ Y___ L___
G__ I___ J___ L___
G00 G___ Z1. M09
G00 G53 Z0
M01
  
```

```
T2 M06 (1/8 DRILL)
G00 G90 G54 X_____ Y_____ M___ S_____
G43 H___ Z1. M08
G81 Z-_____ R_____ F___ L___
G___ I___ J___ L___
X___ Y___ L___
G___ I___ J___ L___
X___ Y___ L___
G___ I___ J___ L___
X___ Y___ L___
G___ I___ J___ L___
G00 G___ Z1. M09
M05
G00 G53 Z0
G00 G___ Y_____
M30
```



### Canned Cycle Exercise #3



Program with a 1/4 Diameter Drill, 118° at 210 SFM with feed of .005 in/rev. (Use the HAAS calculator to get the spindle speed and feed.) Define A rapid plane .050 up from the bottom of circular pocket for *Bolt Hole Circle* and *Bolt Hole Arc*. Rapid plane for *Bolt Holes at an Angle* changes back to .1 above Z zero top of part.

**NOTE:** If an L0 is in the canned cycle line, the cycle will not execute that command until the control reads the next X and/or Y location.

O0010 (put this in after tool #2, but right before the M30 on program O0010)

T03 M06 (T3 – 1/4 Diameter Carbide Stub Drill)

G\_\_\_ G\_\_\_ G\_\_\_ X\_\_\_ Y\_\_\_ S\_\_\_ M\_\_\_ (XY position to center of bolt circle)

G\_\_\_ H\_\_\_ Z\_\_\_ M\_\_\_

G\_\_\_ G\_\_\_ Z\_\_\_ R\_\_\_ F\_\_\_ (drilling canned cycle with *initial point return*)

G\_\_\_ I\_\_\_ J\_\_\_ L\_\_\_ (Bolt Hole Circle Command)

X\_\_\_ Y\_\_\_ L\_\_\_ (XY position center of bolt hole arc, but do not drill a hole here)

G\_\_\_ I\_\_\_ J\_\_\_ K\_\_\_ L\_\_\_

G\_\_\_ X\_\_\_ Y\_\_\_ I\_\_\_ J\_\_\_ L\_\_\_ R\_\_\_ Z\_\_\_ (Bolt hole at an angle and change the R plane)

G\_\_\_ G\_\_\_ Z\_\_\_ M\_\_\_

G\_\_\_ G\_\_\_ Y\_\_\_ Z\_\_\_

M\_\_\_

## ***Additional G Codes***

### **G04 Dwell**

**P code** The dwell time in seconds or milliseconds.

**G04** is used to cause a delay or dwell in the program. The block containing **G04** will delay for the time specified in the **P** code. When programmed on a line following some motion, such as G00, G01, G02 and G03, all motion will be stopped for the amount of time specified in the **P** command in seconds. If the **P** has no fraction part, the delay is in milliseconds (0.001 seconds); otherwise, the delay is in seconds.

The slide motion is stopped, but the spindle will continue to rotate at the requested RPM and the coolant will stay on.

### **G04 P\_\_\_\_\_**

Minimum value	– <b>P.001</b> of a second
Maximum value	– <b>P1000.</b> Seconds



## Milling Circles with Cutter Comp

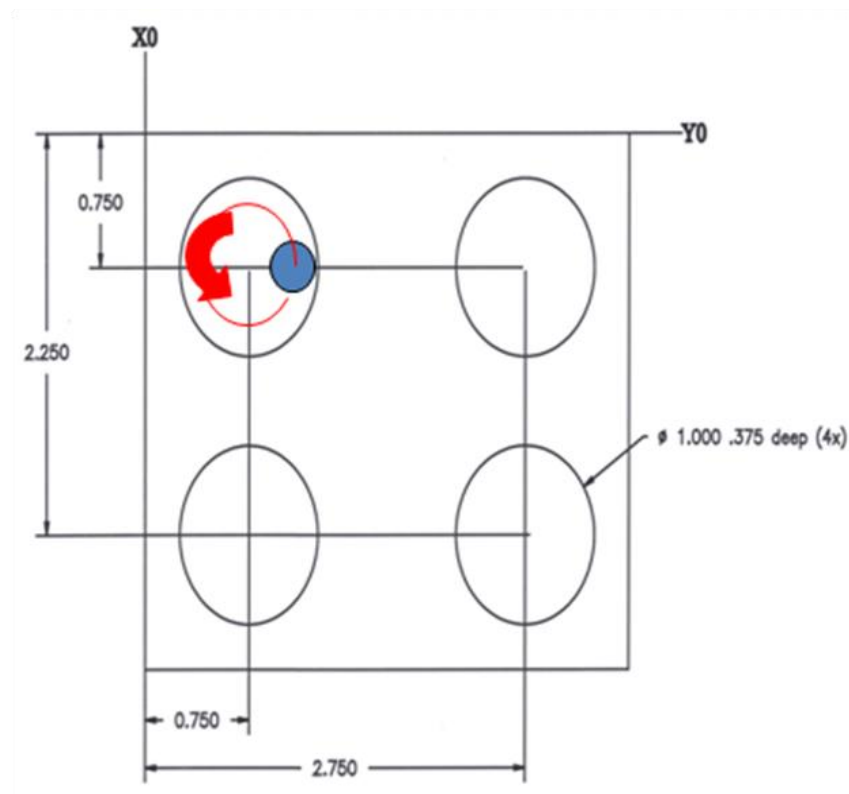
Cutter compensation may be turned on when milling circles. This provides control over the size of the circular pocket by making small changes in the wear of the diameter of the tool used.

Below G3 I-.500 J0 will create a counter clockwise 1.0 inch pocket after turning on the G41 cutter comp. Tool is placed at the center X and Y coordinates of the pocket to be milled.

Tool is taken to depth. Then G41 is turned on to go to the X coordinate of the edge of the finish pocket.

A counter clockwise arc is then created with G3 I-.500 J0 going along the outside of the pocket giving a 1.0 inch pocket. Then G40 is turned on return to the center of the pocket.

- O10000;
- G0 G90 G53 G49 Z0;
- T1 M6 (1/2" E-Mill D1=.250);
- G0 G90 G54 X.750 Y-.750 M3 S3500;
- G43 H1 Z1.0 M8;
- Z.1;
- G1 Z-.375 F5.;
- G41 X1.25 D1 F20.;
- G3 I-.500 J0;
- G1 G40 X.750 Y-.750;
- G0 Z1.0 M9;
- G0 G90 G53 G49 Z0 M5;
- M30;
- %



## ***Thread Milling***

Thread milling uses a standard G02 or G03 move to create the circular move in X-Y, then adds a Z move on the same block to create the thread pitch. This generates one turn of the thread; the multiple teeth of the cutter generate the rest.

Typical line of code: **N100 G02 I-1.0 Z-.05 F5.** (this will generate an 1-inch radius for a 20-pitch thread)

Thread Milling notes: Internal holes smaller than 3/8 inch may not be possible or practical. Always climb cut the cutter. Use a G03 to cut I.D. threads or a G02 to cut O.D. threads. An I.D. right hand thread will move up in the Z-axis by the amount of one thread pitch. An O.D.

right hand thread will move down in the Z-axis by the amount of one thread pitch. PITCH = 1/Threads per inch (Example - 1.0 divided by 8 TPI = .125)

### **Thread Milling Example:**

```
O02300 (Thread milling 1.5 x 8 TPI)
T1 M06 (0.5 DIA 2FLT. THREAD MILL)
G00 G90 G54 X0. Y0. S1910 M03      (X0. Y0. is at the center of the hole)
G43 H01 Z0.1 M08                  (Z0. is at the top of the part - using .5" thick material)
G00 Z-0.6
N1 G01 G41 D01 X0.125 F30.        (Turn on Cutter Comp)
N2 G03 X0.75 Y0. R0.3125 F11.5    (Arc on to the I.D. of bored hole)
N3 G03 I-0.75 Z-0.475            (One full revolution with Z moving up .125)
N4 G03 X0.125 Y0. R0.3125 F30.    (Move away from the new threads)
N5 G01 G40 X0. Y0.               (Cancel Cutter Comp)
G00 Z0.1 M09
G28 G91 Y0. Z0.
M30
```

This program will I.D. thread mill a 1.5 x 8 TPI hole using a .750 diameter x 1.0 thread hob.

To start, take the hole diameter (1.500). Subtract the cutter diameter .750 and

Then divide by 2.  $(1.500 - .75) / 2 = .375$

The result (.375) is the distance the cutter starts from the I.D. of the part. After the initial positioning, the next step of the program is to turn on cutter compensation and move to the I.D. of the circle. The next step is to program a complete circle (G02 or G03) with a Z-axis command of the amount of one full pitch of the thread (this is called "helical interpolation")

The last step is to move away from the I.D. of the circle and turn off cutter compensation

Cutter compensation cannot be turned off or on during an arc movement. A linear move must be made, either in the X or Y axis to move the tool to and from the diameter to cut. This move will be the maximum compensation amount that can be adjusted

## ***Circular Pocket Milling using G12 and G13***

The Haas control has included in its software a Yasnac style circular pocket milling program (G12 clockwise circular pocket.G13 Counterclockwise Pocket). These G codes imply the use of cutter compensation, i.e., a G41 or G42 is not required to be stated in the program line. However, a D -- offset number for cutter radius or diameter is required for the ability to adjust the circle diameter.

In this section, we will cover the G12 and G13 format, as well as the different ways these programs can be written for many various applications:

**SINGLE PASS:** Using **I** only.

**APPLICATIONS:** One-pass counter boring; rough and finish pocketing of smaller holes; I.D. keyway cutting; "O"-ring grooves.

**MULTIPLE PASS:** Using **I**, **K**, and **Q**.

**APPLICATIONS:** Multiple-pass counter boring, rough and finish pocketing of large holes with cutter overlap

**MULTIPLE Z DEPTH PASS:** Using **I** only, or, **I**, **K**, and **Q**. (G91 and **L**)

**APPLICATIONS:** Deep rough and finish pocketing; incremental **Z** depth stepping.

**NOTE:** The tool must be positioned at the center of the circle, either in a previous block or in the G12/G13 line by using **X** and **Y**.

There are two G codes, G12 and G13, that will provide for pocket milling of a circular shape. They are different only in the direction of rotation that is used.

### ***G12, G13      Circular Pocket Milling      Group 00***

D*	Tool Geometry Offset Selection
I	Radius Of First Circle (Or Finish If No K)
K	Radius Of Finished Circle (if using I, K, and Q)
L	Loop Count For Incremental Z Depth Stepping
Q	Radius Increment Or Stepmover
F	Feed Rate in Inches (mm) per minute
Z	Z Depth Of Cut Or Increment

\*In order to get the exact programmed circle diameter, the control uses the selected D code tool size. If this compensation is not desired, program D0.

This **G12** Code implies the use of **G42** and will mill clockwise.

The tool must be positioned at the center of the circular pocket, either in a previous block or in this command using X and Y position. The cut is performed entirely with circular motions of varying radius'. G12 belongs to Group zero and thus is non-modal. If G91 (incremental) is specified and an L count is included, the Z increment is repeated L times at the F feed rate command. If no K is specified, the center roughing passes of this command are removed completely and only one finish pass of the circular pocket is performed.

### **G13 Circular Pocket Milling Counterclockwise**

This G code implies the use of G41 and is otherwise the same as G12. G13 belongs to Group 00 and thus is non-modal.

#### **EXAMPLE: G13 Single Pass Using "I" Only**

```

02345                (.500 entered in the radius/diameter offset column)
T1 M06              (Tool #1 is a .500 diameter end mill)
G90 G54 G00 X1.5 Y1.5 (XY position of center on circular pocket)
S3600 M03
G43 H01 Z.1 M08
G13 Z-.75 I.5 D01 F15. (will complete a 1.0 diameter hole in one pass)
G00 Z1.0 M09
G28 G91 Y0 Z0
M30

```

#### **EXAMPLE: G13 Multiple Pass Using "I", "K", and "Q"**

```

03456                (.750 entered in the radius/diameter offset column)
T3 M06              (Tool #3 is a .750 diameter end mill)
G90 G54 G00 X.5 Y1.5 (XY position of center on circular pocket)
S3600 M03
G43 H01 Z.1 M08
G1 Z0 F15.
G13 G91 Z-.5 I.65 K2. Q.65 D03 L4 F15. (will complete a 4.0 diameter hole in four passes)
G00 Z1.0 M09
G28 G91 Y0 Z0
M30

```

The last program uses G91 and an "L" count of four. This cycle is multiplied by the "L" command and will execute a total of four times at the Z depth increment of .500 to a total depth of 2.0 inches.

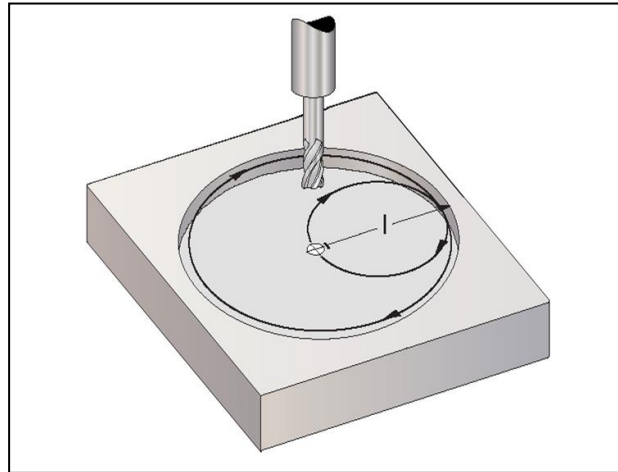
The G91 and "L" count can also be used in G12 and G13 "I" only line.

The figure below shows the tool path during the G12 and G13 cycles. One uses I only and the other uses I, K, and Q.

Example of one pass milling using only the I variable.

Milling a 0.8" diameter 0.5" deep pocket using a 0.5" end mill. The picture shows the tool path for the code given. Note with G12 a clockwise rotation is created where the mill is conventional cutting.

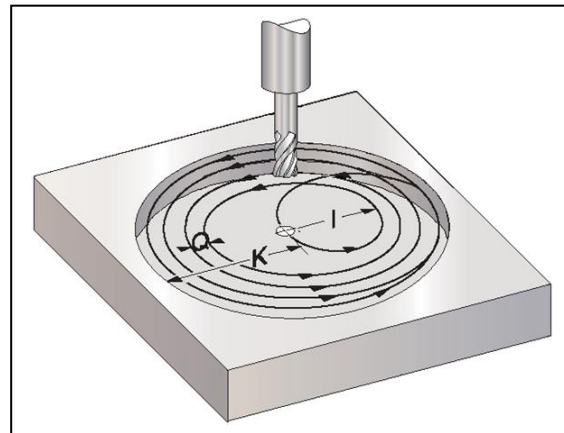
G12 Z-0.5 I0.4 D01 F15



Example of multiple pass milling using I, K, and Q variables. The code is also turned into incremental positioning with G91 and an L added. So the code is repeated 3 times stepping down Z-.5 each pass.

Milling a 3.0" diameter 1.5" deep pocket using a 0.5" end mill.

```
O0010 ;  
T1 M06 ;  
G90 G54 G00 X1.0 Y1.0 ;  
S1500 M03 ;  
G43 Z0.1 H1 M08 ;  
G1 Z0 F30.  
G13 G91 Z-0.5 I0.3 K1.5 Q0.3 D01 F15. L3 ;  
G90 ;  
G00 Z0.1 M09 ;  
G28 G91 Y0 Z0 ;  
M30 ;
```



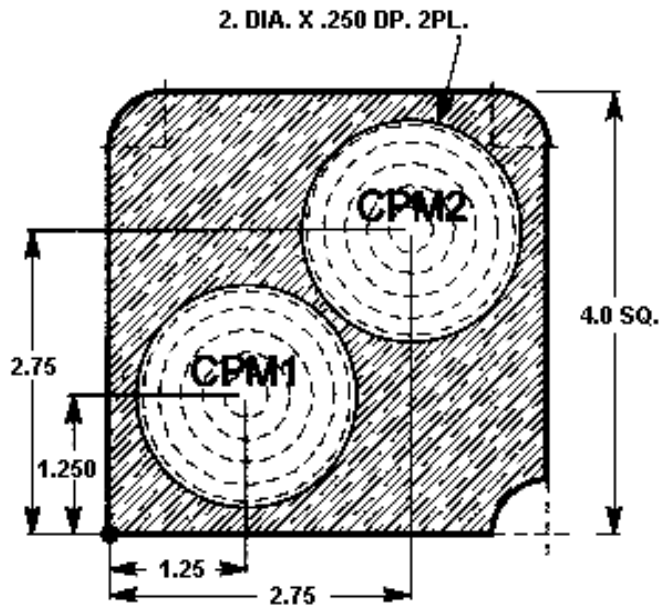
Note the motion of the cutter is counter clockwise using G13 instead of G12. The radius of the first pass is determined by I.3. First the 1/2 end mill plunges in creating 1/2 hole, then the first cut creates a .6 diameter pocket with I .3. The second cut adds Q.3 to the I.3 to equal a .6 radius cut out to a 1.2 diameter. Additional cuts are added with the Q.3 step over until the final K radius or finish pass is completed. Then the whole process is repeated two more times stepping down incrementally the Z-.5. The final pocket depth equals 3 x .5 to give 1.5. The above illustrations just show the initial step of three.

Note the end mill is positioned at the center of the pocket and fed down to **Z0** the top face of the part.

The above illustrations and some text taken from Haas Microsoft PowerPoint titled "Unique G-Codes."

## ***Circular Pocket Milling Exercise***

- G12 Circular Pocket Milling CW  
Or  
G13 Circular Pocket Milling CCW
- D - Tool Radius or Diameter Selection
  - I - Radius of First Circle  
(Or finish If K is not used)
  - K - Radius of Finished Circle  
(if specified)
  - L - Loop count for repeating deeper cuts
  - Q - Radius step over Increment  
(must be used with K)
  - F - Feedrate in inches (mm) per minute
  - Z - Z Depth of cuts or increments



### **CPM1**

Circular Pocket Mill CPM 1 to a depth of .250 and spiral out to size using I, K and Q. Pocket mill another command to run on CPM 1 using I only as a finish pass.

### **CPM2**

Circular Pocket Mill CPM 2 a 2.0 Diameter x .250 depth pocket and incremental stepping down .05 depth using an L count and a G91 incremental command.

**Program with a  $\frac{3}{4}$  Diameter 2 flute E.M. at 190 SFM with .0065 in/rev per flute.** (Use the HAAS calculator to get the spindle speed and feed.)

***(Put this program in after tool #1, but right before the T3 on program 00010)***

T\_\_\_\_ M\_\_\_\_ (Tool #2 – 3/4 2 Flute Diameter End Mill)

G\_\_\_\_ G\_\_\_\_ G\_\_\_\_ X\_\_\_\_ Y\_\_\_\_ S\_\_\_\_ M\_\_\_\_ (X Y Position to CPM1)

G\_\_\_\_ H\_\_\_\_ Z\_\_\_\_ M\_\_\_\_

G\_\_\_\_ I\_\_\_\_ K\_\_\_\_ Q\_\_\_\_ Z\_\_\_\_ D\_\_\_\_ F\_\_\_\_ (CPM1 using I, K, & Q)

G\_\_\_\_ I\_\_\_\_ Z\_\_\_\_ D\_\_\_\_ F\_\_\_\_ (Do a finish pass on CPM1 using I only)

G\_\_\_\_ Z\_\_\_\_ (Z clearance position above part)

X\_\_\_\_ Y\_\_\_\_ (X Y Position to CPM2)

G01 Z0 F20. (Feed down to start point of incremental depth)

G\_\_\_\_ G\_\_\_\_ I\_\_\_\_ K\_\_\_\_ Q\_\_\_\_ Z\_\_\_\_ D\_\_\_\_ L\_\_\_\_ F\_\_\_\_ (CPM2 using I, K, & Q with G91)

G90 G\_\_\_\_ Z\_\_\_\_ M\_\_\_\_

G\_\_\_\_ G\_\_\_\_ Z\_\_\_\_

M\_\_

## ***Circular Plane Selection***

The plane used for circular motions must be comprised of two of the X, Y, or Z axes. The plane selection is modal and will stay in effect for all subsequent circular interpolation commands until canceled by another plane selection code. There are three G codes used to select the circular plane: G17 for XY, G18 for XZ, and G19 for YZ.

### **G17 XY Circular Plane Selection**

The G17 code is used to select the XY plane for circular motion. In this plane, circular motion is defined as clockwise for the operator looking down onto the X-Y table from above.

### **G18 XZ Circular Plane Selection**

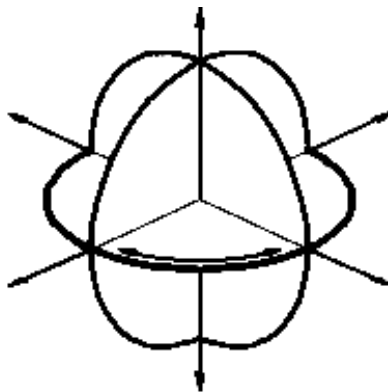
The G18 code is used to select the XZ plane for circular motion. In the X-Z plane (G18), circular motion is defined as clockwise for the operator looking towards the rear of the machine from the front control panel.

### **G19 YZ Circular Plane Selection**

The G19 code is used to select the YZ plane for circular motion. In the Y-Z plane (G19), circular motion is defined as clockwise for the operator looking across the table from the side of the machine the control panel is mounted.

The default plane selection, when the machine is powered on, is G17 for the X-Y plane. This means that a circular motion in the plane of the X-Y table may be programmed without first selecting G17. A helical motion is possible with G02 or G03 by programming the linear axis, which is not in the selected plane. This third axis will be interpolated along the specified axis in a linear manner while the other two axes will be moved in the circular motion. The speed of each axis will be controlled so that the helical rate matches the programmed feed rate.

If cutter radius compensation is selected (G41 or G42), you may only use the X-Y plane for circular motions (G17). Cutter radius compensation is only available in the X and Y axes.





## ***Inch / Metric Selection (G20, G21)***

### **G20 Inch Programming Selection**

### **G21 Metric Programming Selection**

Selection between inch and metric programming can **only be done from Setting 9**. Inch programming allows displacements up to 838 inches and a resolution of 0.0001 inches. Metric programming uses millimeter units with a maximum displacement of 8380 mm and a resolution of 0.001 mm.

When in metric, the feed rate is also defined as millimeters per minute with a range of 1000 to 0.001 mm/min.

When jogging in metric, the speeds and units on the keypad are interpreted as mm/min, but the value used is ten times larger than shown on the keypad.

The optional fourth or fifth axis programming is not affected by the selection of metric; it is always programmed in degrees. The auxiliary C axis is also always in degrees.

Changing the setting from inches to metric or back again will change the content of any programs already stored in memory. Programs must be reloaded with metric values after changing this setting.

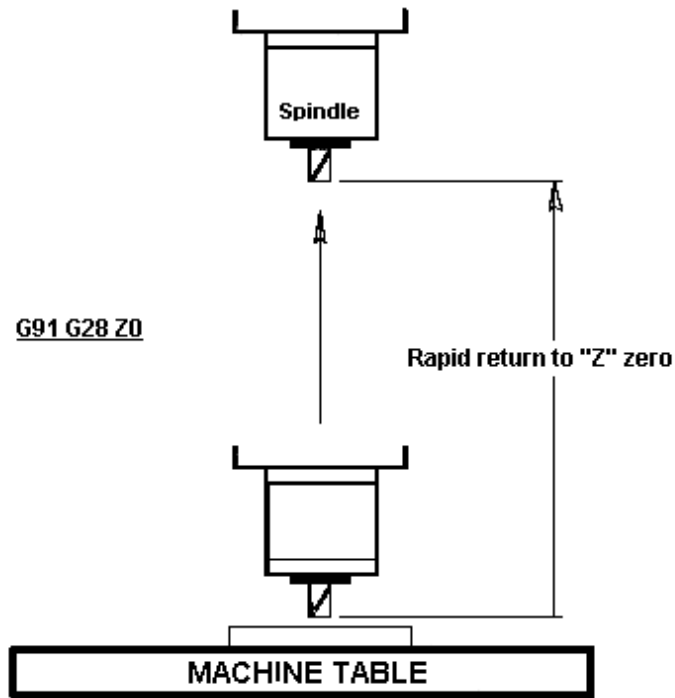
The standard G codes, G20 and G21, are sometimes used to select between inch and metric, **however**, in this control, the G20 (inch) and G21 (mm) codes can only be used to ensure that the inch/metric setting is set correctly for that program.

## ***Reference Point Definition and Return***

### **G28 Return To Reference Point, Set Optional Intermediate Point**

The G28 code is used to return to the machine zero position on all axes. If an X, Y, Z, or A axis is on the same block and specifies a location, only those axes will move and return to the machine's zero reference point and the movement to the machine's zero reference point will be through that specified location. This point is called the intermediate point and is saved, if needed, for use in G29. If you do not want to position through an intermediate point while specifying a specific axis to position to machine zero, then add an incremental (G91) command to this line along with a Z0, Y0, and/or X0 for the specific axis you want to send to machine zero. This will command those axes specified to position incrementally to a zero distance as an intermediate point, and then those axes specified will then position to machine zero. Be sure to program in an absolute (G90) command in the startup lines for the next tool, which is usually needed for the beginning of each tool.

If no X, Y, Z, or A is specified, all axes will be moved directly to machine zero. Any auxiliary axes (B, C,...) are returned to their machine home after the X, Y, Z, and A axes. G28 also cancels tool length offsets.



## ***Setting Work, Tool Offsets through the Program (G10)***

The G10 code can set or change tool and work offset within the program. This code may be used when dedicated work holding fixtures and tools are used to reduce set up time.

Definitions: L- Selects an offset category

L2 Work coordinate offset for G52 and G54-G59

L20 Auxiliary work coordinate offsets for G110-G129, G154

L10 Tool Length offset (for H code)

L1 or L11 Tool wear offset (for H code)

L12 Diameter Offset (for D code)

L13 Diameter Wear offset (for D code)

P- Selects a specific offset

P0 G52 references work coordinate (L2)

P1-P6 references work coordinates respectively G54-G59 (L2)

P1-P20 references auxiliary work coordinates G110-G129 (L20)

P1-P99 references auxiliary work coordinates G154 P1-P99 (L20)

Q- Offset value or increment for length of diameter.

X-axis zero location

Y-axis zero location

Z-axis zero location

A-axis zero location

Examples:

G10 L2 P1 G90 X-16.8420 Y-4.9700 Z-0.0100 (SETS G54 OFF)

G10 L20 P50 G90 X-16.8420 Y-4.9700 Z-0.0100 (SETS G154 P50 OFF)

G10 L10 G90 P1 R-13.2760 (SETS T1 LENGTH OFF)

G10 L12 G90 P1 R3.0000 (SETS T1 DIAM OFF)

G10 L2 P1 G91 X6.0 (INCREMENTAL MOVE OF OFFSET G54 X6.0)

## ***General Purpose Pocket Milling (G150)***

### **G150 General Purpose Pocket Milling**

D	Cutter size selection
F	Feed rate
I	X-axis cut increment
J	Y-axis cut increment
K	Finishing cut allowance
L	Optional repetition count
P	Subroutine number defining outside of shape
Q	Incremental Z axis cut depth per pass
R	R plane position
S	Optional spindle speed
X	X position of starting hole
Y	Y position of starting hole
Z	Final depth of the pocket

This G code provides for general purpose pocket milling. The shape of the pocket to be cut must be defined by a series of motions within a subroutine. A series of motions in either the X or Y-axis will be used to cut out the specified shape followed by a finishing pass to clean up the outer edge - either I or J must be specified. If I is used, the pocket is cut from a series of strokes in the X-axis. If J is used, the pocket is cut from a series of strokes in the Y-axis. I and J must be positive numbers.

Multiple passes over the area can be selected to control the depth of the cut. At least one pass is made over the pocket. Multiple passes are made after feeding down by Q amount until the Z depth is reached. Q must be positive. If an L count is specified, the entire block is repeated and an incremental X or Y (G91) will reposition the pocket.

The subroutine must define a closed area by a series of G01, G02, or G03 motions in X and Y and must end with an M99. G codes G90 and G91 can also be used in the subroutine to select absolute or incremental. Any codes other than G, I, J, R, X, or Y are ignored in the subroutine. This subroutine must consist of less than 20 strokes.

Pocket milling should begin from a clearance hole, which has been previously drilled to the Z depth in order to clear the tool on entry to the pocket. The G150 block must specify this hole location with X and Y.

The first motion in the subroutine should move from this clearance hole to the starting point of the block shape. The final motion in the subroutine should return to the same point as the starting motion of the subroutine, i.e., the example on the following page. The start point of the G150 line is X3.25 Y4.5 and the first move of the sub is Y7.0. Therefore, the end of the sub must return to X3.25 Y7.0.

If a K is specified, the finishing pass is taken along the outside edge, but is done at the full pocket depth, and the previous cuts will cut inside the programmed pocket size by K.

```

O0040 (G150 pocket example)
T1 M06 (tool #1 cuts entry for end mill)
G90 G56 G00 X3.25 Y4.5 (starting hole position)
S2000 M03
G43 H01 Z1.0 M08
G83 Z-2.0 R.1 Q.5 F10.
G80 G00 Z1.0 M09
G28 G91 Z0 M05

```

```

T2 M06 (tool #2 – ½ diameter end mill cuts pocket, four passes to Z depth)
G90 G56 G00 X3.25 Y4.5 (starting hole position)
S2000 M03
G43 Z1.0 H02 M08
G1 Z0 F15.0
G150 P200 J0.4 K0.01 G41 D02 Z-2. Q0.5 R.1 F15.
G01 G40 X3.25 Y4.5 (be sure to cancel cutter comp.)
G00 Z.1 M09
G28 G91 Y0 Z0 M05
M30

```

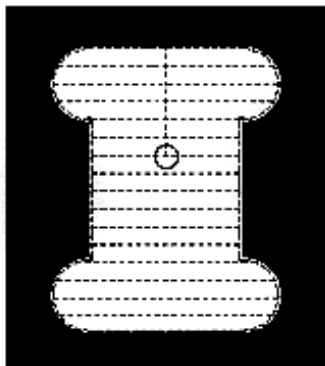
SUBROUTINE

```

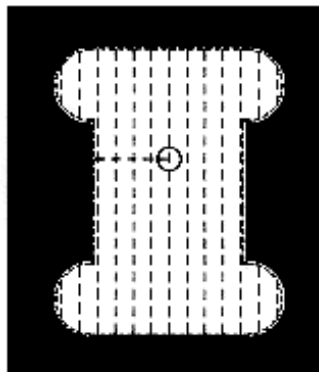
O0200 (G150 pocket geometry)
G01 Y7.
X1.5
G03 Y5.25 R0.875
G01 Y2.25
G03 Y0.5 R0.875
G01 X5.
G03 Y2.25 R.875
G01 Y5.25
G03 Y7. R0.875
G01 X3.25
M99 (return to main program)

```

**Using J.4**



**Using I.4**



## PROGRAM LINE REQUIREMENTS:

X = X-axis position of the starting hole  
Y = Y-axis position of the starting hole  
Z = Final depth of the hole  
F = Feed rate  
R = Reference plane  
Q = Incremental Z-axis cut depth per pass  
I = X-axis cut increment  
or  
J = Y-axis cut increment  
K = finish cut allowance  
P = Subprogram number  
D = Geometry offset number  
G41 or G42 = Cutter compensation turn **ON**

The shape of the pocket to be cut must be defined by a series of motions within a subprogram. One of either **I** or **J** must be specified. If **I** is used, the pocket is cut from a series of strokes in the X-axis. If **J** is used, the pocket is cut from a series of strokes in the Y-axis. The value entered with the **I** or **J** will be the shift amount or cutter overlap. The **K** amount is the finishing allowance for the walls of the pocket.

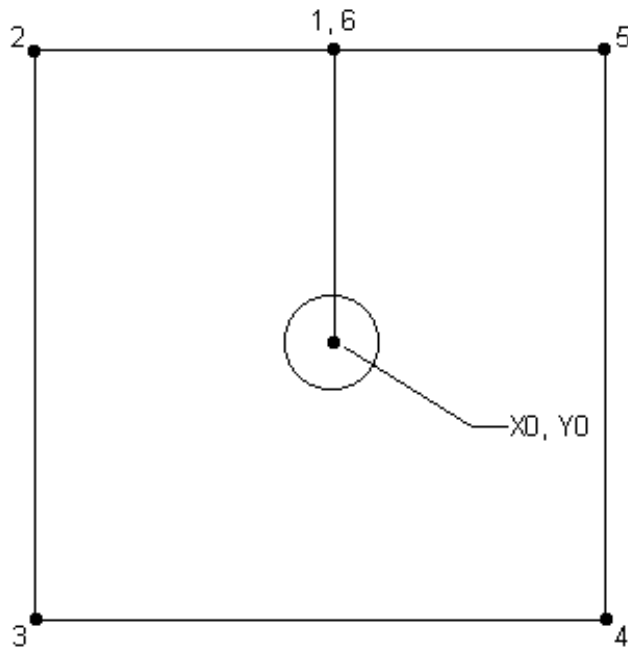
The subprogram must define a closed area by a series of G01, G02, or G03 motions on X- and Y-axes, and must end with an M99. The only other codes that can be used in the subprogram are: G90, G91, **I, J, R, X,** and **Y**, any other codes are ignored. This subprogram must not exceed 20 strokes.

**NOTE:** When defining the contour in the subprogram, the idea to keep in mind is to only connect the contour - not to return to the starting point.

### G150 EXAMPLES -

4.0 x 4.0 x .500 DP. SQUARE POCKET:

```
O1000
T1 M06 (Tool #1 is a .500 diameter end mill)
G00 G90 G54 X0 Y0 S2000 M03
G43 H01 Z.1 M08
G01 Z0 F10.
G150 X0 Y0 Z-.5 F10. R.1 Q.25 I.4 K.01 P500 D01 G41
G01 G90 X0 Y0
G00 Z1.0 M09
G40 G28 G91 Y0 Z0
M30
```



*Fig. 8-6 Pocket Milling exercise for G150 operation.*

**ABSOLUTE SUBPROGRAM:**

```
00500
G01 Y2.0
X-2.0
Y-2.0
X2.0
Y2.0
X0
M99
```

**INCREMENTAL SUBPROGRAM:**

```
00500
G01 G91 Y2.0
X-2.0
Y-4.0
X4.0
Y4.0
X-2.0
G90
M99
```

**SQUARE ISLAND:**

```
01000
T1 M06 (Tool is a .500 diameter end mill)
G00 G90 G54 S2500 M03
G43 H01 Z.1 M08
G01 Z0 F10.
G150 X1.0 Y1.0 Z-.5 F15. R.1 Q.25 I.4 K.01 P500 D01 G41
G00 Z1.0 M09
G40 G28 G91 Y0 Z0
M30
```

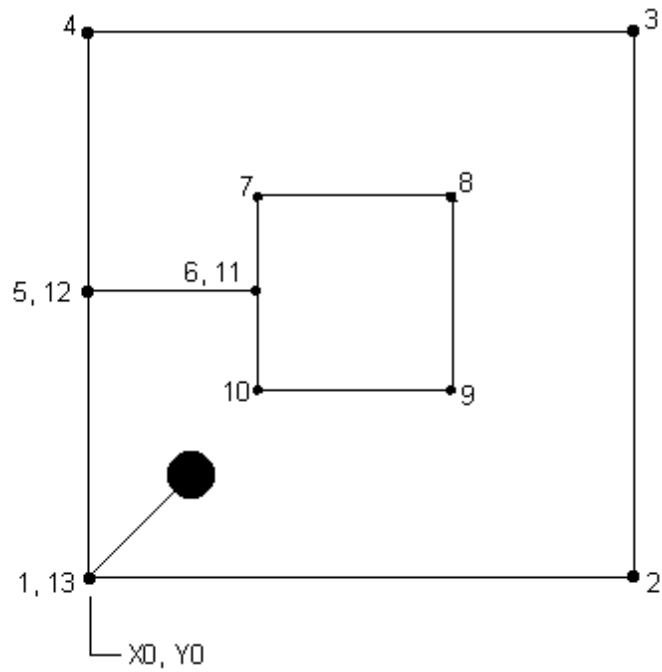


Fig. 8 - 7 Square island programming exercise using G150.

```

00500
G01 X0 Y0
X6.0
Y6.0
X0
Y3.0
X2.0
Y4.0
X4.0
Y2.0
X2.0
Y3.0
X0
Y0
M99

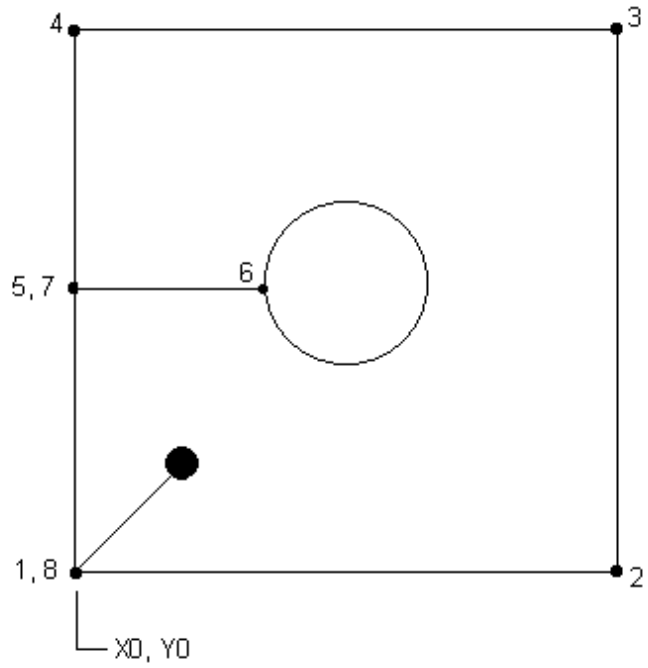
```

**ROUND ISLAND:**

```

01000
T1 M06 (Tool is a .500 diameter end mill)
G00 G90 G54 S2500 M03
G43 H01 Z.1 M08
G01 Z0 F10.
G150 X1.0 Y1.0 Z-.5 F15. R.1 Q.25 I.4 K-01 P500 D01 G41
G00 Z1.0 M09
G40 G28 G91 Y0 Z0
M30

```



*Fig. 8 - 8 Round island programming using G150*

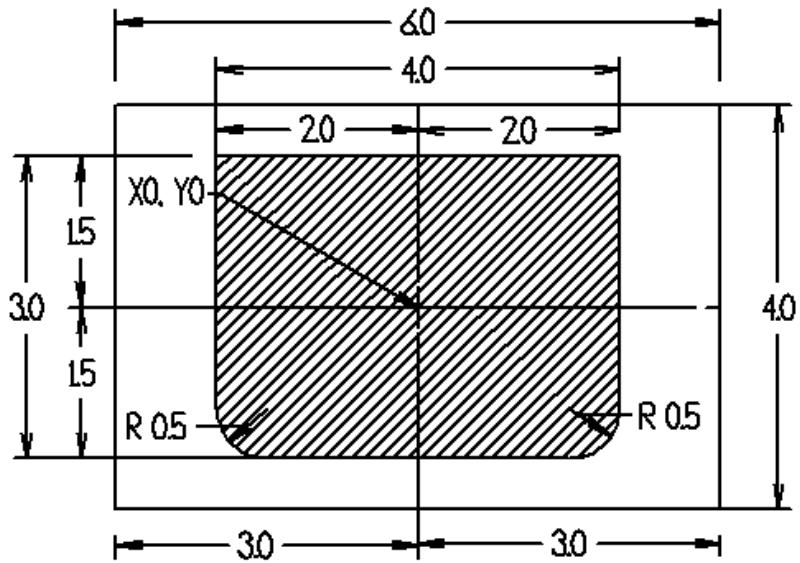
```

O0500
G01 X0 Y0
X6.0
Y6.0
X0
Y3.0
X2.0
G02 I1.0
G01 X 0
Y0
M99.

```



**G150 EXERCISE:**



Mill pocket using a 1/2" 4 flute End Mill, to a final pocket depth of .5 inch. Step down in .25 increments. Run end mill at surface feed of 120 ft/min with a chip load of .002in/rev. Set middle of part as the origin for programming.

04500 (G150 pocket exercise)

T1 M06 (1/2" Drill tool #1 cuts entry for end mill)

G90 G54 G00 X0 Y0 (starting hole position)

S2000 M03

G43 H01 Z1.0 M08

G83 Z-.50 R.1 Q.25 F10.

G80 G00 Z1.0 M09

G28 G91 Z0 M05

T2 M06

G00 G90 G54 X0 Y0 S \_\_\_\_ M03

G43 H2 Z.1 M08

G01 Z0 F10.

G150 X \_\_\_\_ Y \_\_\_\_ Z \_\_\_\_ F \_\_\_\_ R \_\_\_\_ Q \_\_\_\_ I \_\_\_\_ K \_\_\_\_ P4600 D \_\_\_\_ G41 OR G42

G00 Z1.0 M09

G28 G40 G91 Y0 Z0

M30

04600

G01 X \_\_\_\_ Y \_\_\_\_

X \_\_\_\_

Y \_\_\_\_

G \_\_\_\_ X \_\_\_\_ Y \_\_\_\_ R \_\_\_\_

G \_\_\_\_ X \_\_\_\_

G \_\_\_\_ X \_\_\_\_ Y \_\_\_\_ R \_\_\_\_

G1 Y \_\_\_\_

X \_\_\_\_

M99

## Engraving (G47)

### G47 ENGRAVING

### GROUP 00

E	=	Plunge rate (units/min)
F	=	Engraving feed rate (units/min)
I	=	Angle of rotation (-360. to +360.), default is 0.
J	=	Scaling factor in inches (minimum = 0.001 inches), default is 1.0 inch
P	=	I for Sequential Serial Number Engraving
	=	0 for Literal String Engraving
R	=	Return plane
X	=	X start of engraving
Y	=	Y start of engraving
Z	=	Depth of cut

The text to engrave should be in the form of a comment on the same line as G47, with either a P1 or P0 before it. P1 selects Sequential Serial Number Engraving and P0 selects Literal String Engraving.

### SEQUENTIAL SERIAL NUMBER ENGRAVING

This method is used to engrave numbers on a series of parts, with the number being incremented by one each time. The '#' symbol is used to select the number of digits in the serial number. For example: **G47**

**P1 (####)**

will limit the serial number to four digits.

The initial serial number can be either programmed or set manually. If it is programmed , for example:

**G47 P1 (1234)**

This code will set the initial serial number to "1234".

The initial serial number can also be set manually into a macro variable. The "MACROS" option does not have to be enabled to do this. Macro variables are temporary storage locations for numbers. **Macro variable #599** is used to hold the initial serial number to be engraved. To set this variable, go to the CURNT COMDS page and press the PAGE DOWN key until the "Macro Variables" page appears. Then type in "599" and press the (DOWN ARROW). Now enter the desired initial serial number at the cursor and press the WRITE key. For example, when macro variable #599 is set to "1234",

G47 P1 (####) will produce this: 1234

If the number in macro variable #599 has more characters than specified in the format string, only the quantity specified will be printed. For example, if #599 is set to "12345" and only four places are specified in the format string, only "2345" will be engraved.

## LITERAL STRING ENGRAVING

This method is used to engrave desired text on a part. The characters available for engraving are:

A..Z

a..z

0..9

! " # & ' ( ) \* + , / : ; < = > ? [ \ ] ^ { }

However, programs downloaded through the serial port or the floppy drive can take advantage of characters not available on the mill keypad. For Literal String Engraving, the text should be in the form of a comment on the same line as the P0 statement.

For example: G47 P0 (ENGRAVE THIS)

will produce: ENGRAVE THIS

Note that P0 is used, instead of P1, for Literal String Engraving.

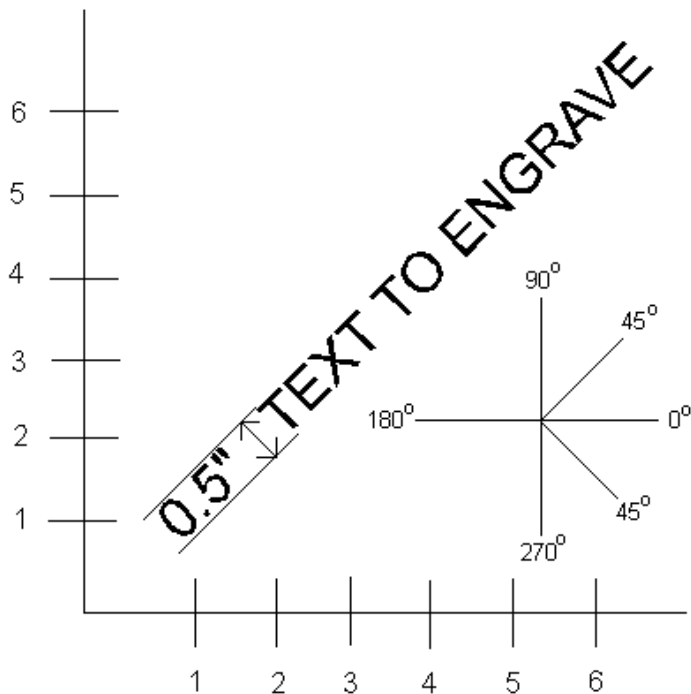
## EXAMPLE

This example will create the engraving opposite.

```
O05555 (TEXT ENGRAVING)
T7M06
G00 G90 G98 G54 X2.0 Y2.0
M03 S3000
G43 H07 Z0.03 M08
G47 P0 X2.0 Y2.0 I45.J.5 R.05 Z-.005 F15.0 E10.0 (TEXT TO ENGRAVE)
G00G80Z0.03M09
M05
G28G91Z0
G00G90G54X0Y0
M30
```

In this example:

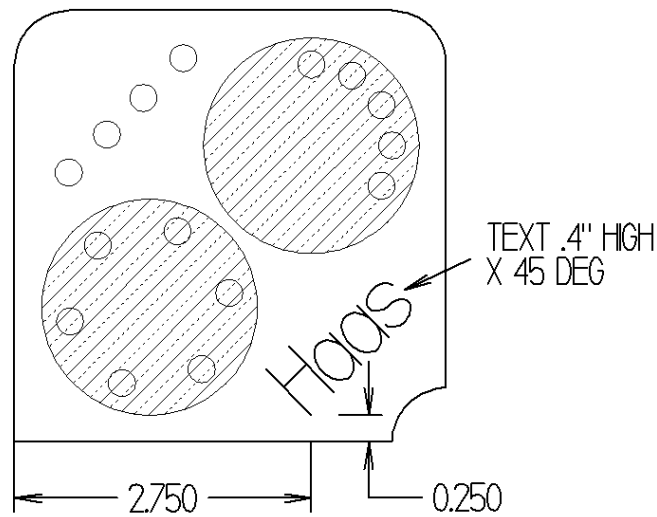
G47 P0	select Literal String Engraving
X2.0 Y2.0	select 2.0, 2.0 as the starting point for the text.
I45.	places the text at a positive 45 degree angle
J.5	sets the text height to 0.5 inch
R.05	commands the cutter to retract to .05 inches (mm) above the cutting plane after engraving each character
Z-.005	selects a .005 inch (mm) deep cut
F15.0	selects an engraving feed rate of 15 units/min
E10.0	commands the cutter to plunge at a rate of 10 in/min



## Text Engraving Exercise

Use T4 Engraving Tool. Run at RPM S5000, F10. In feed E3.0

Insert code after drilling operation of program on page 73.



(TEXT ENGRAVING)

```
T4 M06
G00 G90 G98 G54 X____ Y____
S____ M03
G43 H04 Z0.03 M08
G47 P__ X__ Y__ I__ J__ R__ Z__ F__ E__ (____)
G00 G80 Z0.03 M09
M05
G28 G91 Z0
G00 G90 G54 X0 Y0
M30
```



## ***Subroutines (Subprograms)***

Subprograms are a very important part of CNC programming, however, they can make programming become more complicated. This unit will give a brief introduction to subprograms and some of their basic applications.

### ***Objectives:***

Upon completion of this unit, the student will:

- 1) Have a good working definition of a “subprogram.”
- 2) Understand the basic application and deployment of subprograms.
- 3) Know the differences between M97 (local) and M98 (separate) subprogram options.
- 4) Be able to integrate a subprogram into a standard program to achieve the desired results.
- 5) Looping subprograms with an “L” variable.\*

\*Denotes exposure only – subject not covered in detail.

## ***Subroutines***

A subprogram is a separate program called up by another program. The use of subprograms can significantly reduce the amount of programming on some parts. Subroutines allow the CNC programmer to define a series of commands that might be repeated several times in a program and, instead of repeating them many times, can be “called up” when needed.

A subroutine call is performed using an **M97** or **M98** and a **Pnnnn**. The P code command identifies the O program number being used when executed with M98 or an N sequence number to identify the block where a local subroutine starts when executed with M97.

**Local** subroutines are called with an **M97**. This can be easier to use than the M98 because the subroutine is contained within the main program without the need to define a separate Onnnn program. With local subroutines, you define an M30 for the end of your main program portion followed by a sequence number to define the beginning of your subroutine.

This subroutine call causes the blocks in the subroutine to be executed as if they were included in the main program. To return back to the main program, you need to end the subroutines with an **M99**, which sends it back to the next line after the subroutine call in the main program.

Another important feature of a “subroutine call” is that the M98 block may also include an L (loop) or repeat count. If there is an Ln with the subroutine call, it is repeated that number of times before the main program continues with the next block.

The most common use of subroutines is in the definition of a series of holes, which may need to be center drilled, peck drilled, taped, and/or chamfered. If a subroutine is defined that consists only of the X-Y position of the holes, the main program can define the canned cycles and the hole locations can be called up in the subroutine to do each of the tool operations. Thus, the X-Y positions can be entered only once and used several times for each tool.

```
O0100 (main program using subroutines);  
T01 M06 (center drill);  
G90 G54 G00 X0.5 Y0.5;  
S2000 M03;  
G43 Z0.5 H01;  
G81 Z-0.3 R0.1 F15. (define canned cycle);  
M98 P0200 (call subroutine O200);
```

```
T02 M06 (peck drill);  
G90 G54 G00 X0.5 Y 0.5;  
S1500 M03;  
G43 Z0.5 H02;  
G83 Z-1. R0.1 Q.100 F10. (define canned cycle);  
M98 P0200 (call subroutine O200);
```

```
T03 M06 (1/4 – 20 tap);  
G90 G54 G00 X.5 Y.5;  
S400 (G84 will turn spindle on, no M03 is needed);  
G43 Z1. H03;  
G84 Z-1. R.1 F10. (define canned cycle);  
M98 P0200 (call subroutine O200);  
T04 M06 (chamfer);
```



```
G90 G54 G00 X.5 Y.5;
S2500 M03;
G43 Z0.1 H04;
G81 Z-0.1 R0.1 F20. (define canned cycle);
M98 P0200 (call subroutine O200);
G28 Y0.;
M30 (end of main program);
```

(SUBROUTINE) (LISTING ALL THE HOLE POSITIONS)

```
O0200
X1.5;
X2.5;
X2. Y1.;
X1.;
X.5 Y1.5;
X1.5;
X2.5;
G80 Z1.0 M09;
G28 G91 Z0. M05;
M99 (end of subroutine);
```

*To return back to the main program, you need to end the subroutines with an M99, which sends it back to the next line after the subroutine call in the main program.*

```
O0300 (example using a local subroutine);
T1 M06;
G90 G54 G00 X.5 Y.5;
S2000 M03;
G43 Z1.0 H01;
G81 Z-0.1 R0.1 F20 (define canned cycle);
M97 P0500 (call local subroutine using sequence N#);
```

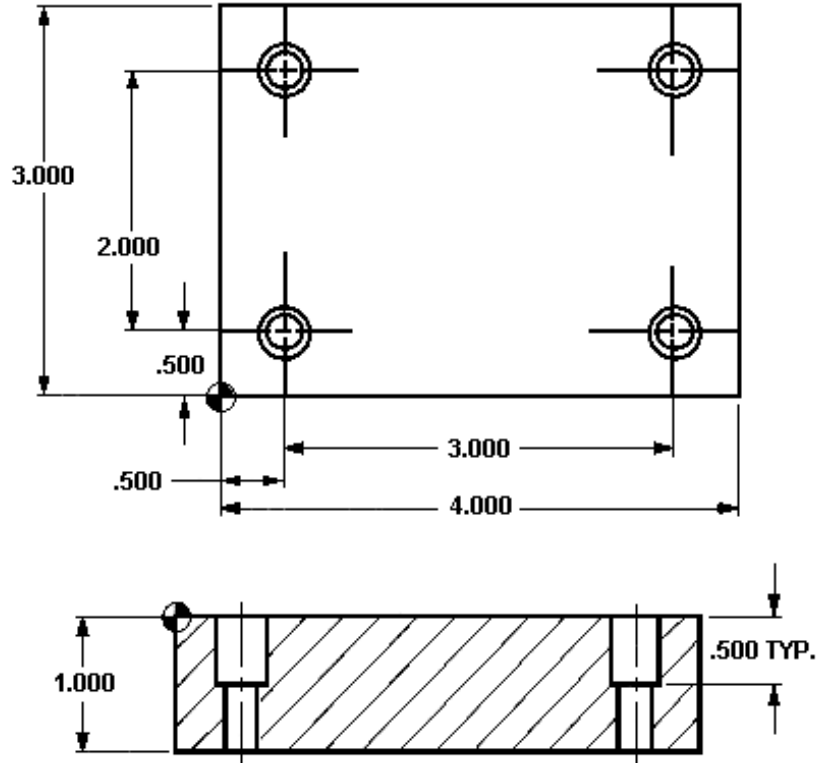
```
T02 M06 (peck drill);
G90 G54 G00 X.5 Y.5;
S1000 M03;
G43 Z1. H02 M08;
G83 R0.1 Z-1. F10. (define canned cycle);
M97 P0500 (call local subroutine using sequence number 500);
```

```
G28 Y0.;
M30 (end of main program);
```

```
N0500 (local subroutine example listing all hole positions);
X1.5;
X2.5;
Y1. X2.;
X1.;
X0.5 Y1.5;
X1.5;
X2.5;
G80 G00 Z1. M09;
G91 G28 Z0. M05;
M99 (end of local subroutine);
```

# Exercises

## Canned Cycle Exercise #1



### TOOL NUMBER ONE

- ½ Diameter Center/Spot Drill (use G81 canned cycle)
- surface speed (SFM) = 190
- feed inch/rev = .0035

### TOOL NUMBER TWO

- ¼ Diameter Drill (use G83 canned cycle)
- surface speed (SFM) = 190
- feed inch/rev = .006

### TOOL NUMBER THREE

- 3/8 Diameter 4 flute End Mill (use G82 canned cycle)
- surface speed (SFM) = 190
- inch/rev/tooth = .003

**00068 (CANNED CYCLE EXERCISE)**

N1 (1/2 DIA. SPOT DRILL)

G53 Z0 G49

T\_\_ M\_\_

G\_\_ G\_\_ G\_\_ X.\_\_ Y.\_\_ M\_\_ S\_\_\_\_

G\_\_ Z.\_ H\_\_ M\_\_

G\_\_ Z-.\_\_ R\_\_ F\_\_ L\_

M\_\_ P\_\_\_\_

M\_\_

N2 (1/4 DIA. DRILL)

T\_\_ M\_\_

G\_\_ G\_\_ G\_\_ X\_\_ Y\_\_ M\_\_ S\_\_\_\_

G\_\_ Z\_\_ H\_\_ M\_\_

G\_\_ Z-\_\_ R\_\_ F\_\_ Q.\_\_\_\_ L\_

M\_\_ P\_\_\_\_

M\_\_

N3 (3/8 DIA. ENDMILL)

T\_\_ M\_\_

G\_\_ G\_\_ G\_\_ X\_\_ Y\_\_ M\_\_ S\_\_\_\_

G\_\_ Z\_\_ H\_\_ M\_\_

G\_\_ Z-\_\_ R\_\_ F\_\_ P\_\_ L\_

M\_\_ P\_\_\_\_

G\_\_ G\_\_ Y\_\_

M\_\_

N1000 (HOLE SUBROUTINE)

X\_\_ Y\_\_

Y\_\_

X\_\_

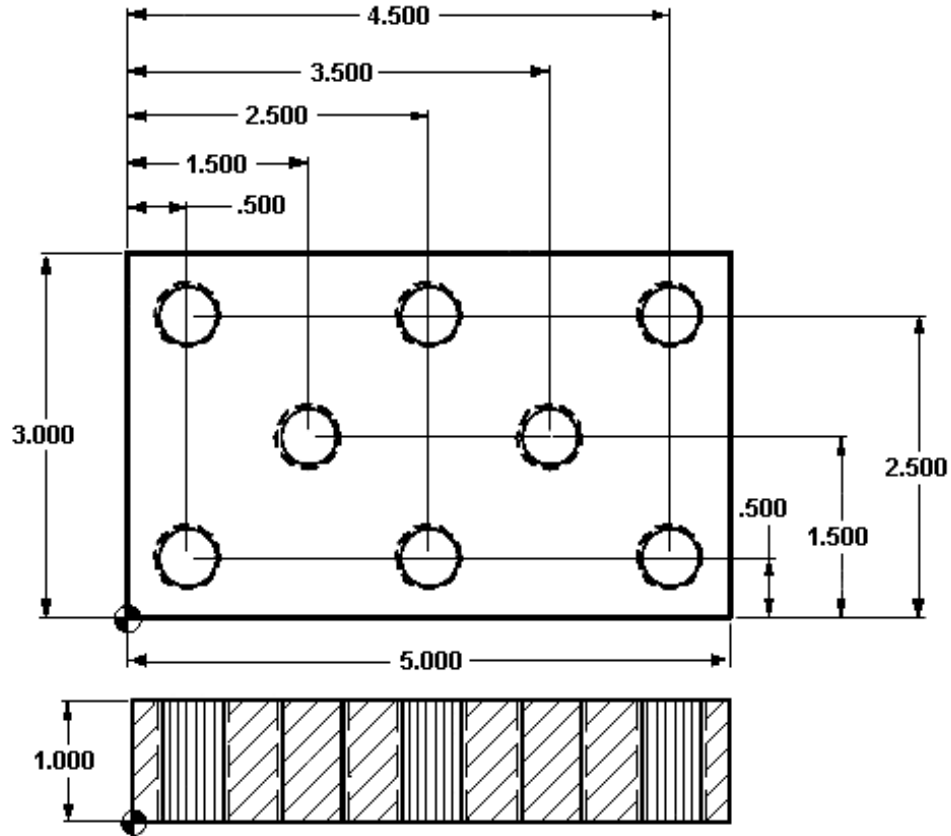
Y\_\_

G\_\_ G\_\_ Z\_\_ M\_\_

G\_\_ G\_\_ Z\_\_

M\_\_

## Canned Cycle Exercise #2



### TOOL NUMBER ONE

- 5/8 Diameter Spot Drill (use G81 canned cycle)
- surface speed (SFM) = 190
- feed in/rev. = .0035

### TOOL NUMBER TWO

- 27/64 Diameter Drill (use G73 canned cycle)
- surface speed (SFM) = 190
- feed in/rev = .006

### TOOL NUMBER THREE

- 1/2-13 Tap (use G84 canned cycle)
- 600 RPM

**00071(CANNED CYCLE EXERCISE)**

N1 (5/8 DIA. SPOT DRILL)

T\_\_ M\_\_  
G\_\_ G\_\_ G\_\_ X.\_\_ Y.\_\_ M\_\_ S\_\_  
G\_\_ Z.\_\_ H\_\_ M\_\_  
G\_\_ Z-.\_\_ R\_\_ F\_\_ L\_\_  
M\_\_ P\_\_  
M\_\_

N2 (27/64 DIA. DRILL)

T\_\_ M\_\_  
G\_\_ G\_\_ G\_\_ X\_\_ Y\_\_ M\_\_ S\_\_  
G\_\_ Z\_\_ H\_\_ M\_\_  
G\_\_ Z-\_\_ R\_\_ F\_\_ L\_\_ Q.\_\_  
M\_\_ P\_\_  
M\_\_

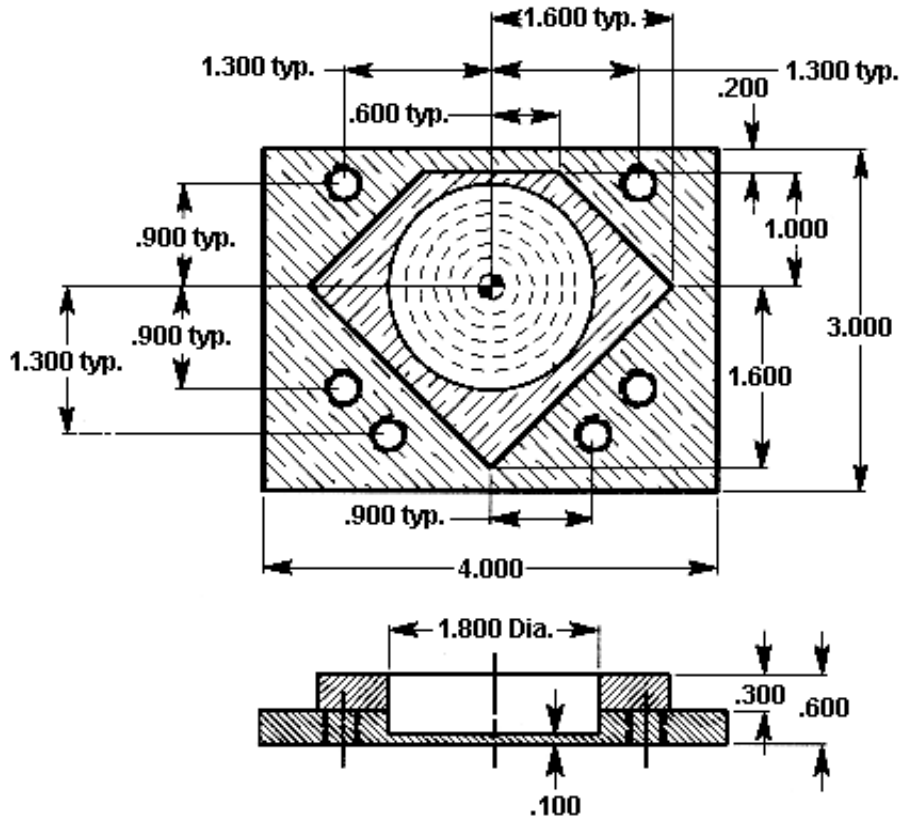
N3 (1/2 - 13 TAP)

T\_\_ M\_\_  
G\_\_ G\_\_ G\_\_ X\_\_ Y\_\_ S\_\_  
G\_\_ Z\_\_ H\_\_ M\_\_  
G\_\_ Z-\_\_ R\_\_ F\_\_ L\_\_  
M\_\_ P\_\_  
G\_\_ G\_\_ Y\_\_  
M\_\_

N2000 (HOLE SUBROUTINE)

X\_\_ Y\_\_  
X\_\_  
X\_\_  
X\_\_ Y\_\_  
X\_\_  
X\_\_ Y\_\_  
X\_\_  
X\_\_  
G\_\_ G\_\_ Z\_\_ M\_\_  
G\_\_ G\_\_ Z\_\_  
M\_\_

## FINAL EXERCISES



Tool #1 - 2.00 diameter 4 flute end mill – finish diamond shape contour 1 pass at .3 depth  
275 SFM - .005 inch/rev per tooth

Tool #2 – 1 3/4 diameter insert drill – rough 1.800 circular pocket at .480 depth  
250 SFM - .0075 inch/rev

Tool #3 – 5/8 diameter 4 flute end mill – finish mill 1.800 diameter circular pocket .5 depth  
220 SFM - .0045 inch/rev per tooth

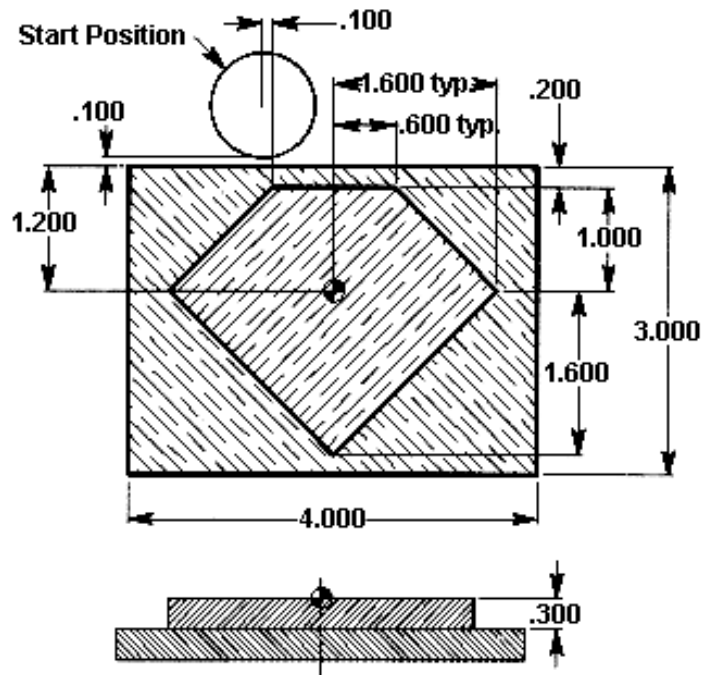
Tool #4 – 1/2 diameter spot drill – drill 6 holes .2 depth  
200 SFM - .005 inch/rev

Tool #5 – 5/16 diameter drill (6) holes through  
200 SFM - .005 inch/rev

Tool #6 – 3/8 – 16 tap (6) holes through  
Tap at 600 RPM and your machine has rigid tapping

Tool #7- Engrave your name anywhere on the part using a 1/8 inch ball end mill  
Engrave at 200ft/min - .001 in/rev

## CLASS EXERCISE



**Tool #1** – 2.00 diameter 4 flute end mill - climb cut a finish pass around diamond shape contour - 1 pass at .3 depth using cutter compensation.

275 SFM - .005 inch/rev per tooth

01234

T\_\_\_ M\_\_\_ (tool #1 – 2.00 diameter end mill)

G\_\_\_ G\_\_\_ G\_\_\_ X\_\_\_ Y\_\_\_

M\_\_\_ S\_\_\_

G\_\_\_ H\_\_\_ Z\_\_\_ M\_\_\_

G\_\_\_ Z\_\_\_ F\_\_\_

G\_\_\_ Y\_\_\_ D\_\_\_ F\_\_\_

X\_\_\_

X\_\_\_ Y\_\_\_

X\_\_\_ Y\_\_\_

X\_\_\_ Y\_\_\_

X\_\_\_ Y\_\_\_

G\_\_\_ Y\_\_\_

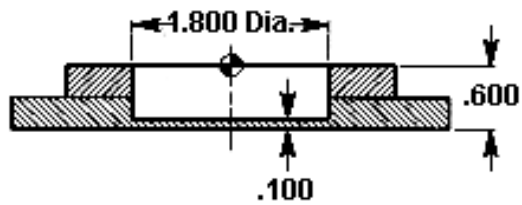
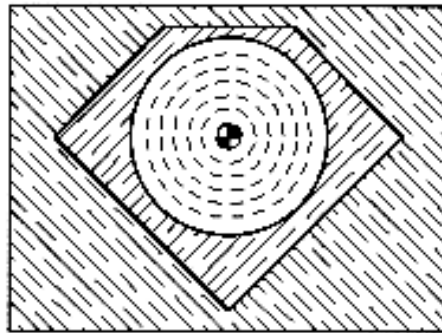
G\_\_\_ Z\_\_\_ M\_\_\_

G\_\_\_ G\_\_\_ Z\_\_\_

M\_\_\_

*Be sure to position cutter at least half the cutter diameter off of the part surface before canceling cutter comp.*

**CLASS EXERCISE**



**Tool #2** – 1 3/4 diameter insert drill – rough 1.800 circular pocket at .480 depth

Speed 250 SFM. Feed .0075 inch/rev

T\_\_\_ M\_\_\_ (tool #2 – 1 3/4 diameter insert drill)

G\_\_\_ G\_\_\_ G\_\_\_ X\_\_\_ Y\_\_\_

M\_\_\_ S\_\_\_

G\_\_\_ Z\_\_\_ H\_\_\_ M\_\_\_

G\_\_\_ Z\_\_\_ R\_\_\_ F\_\_\_

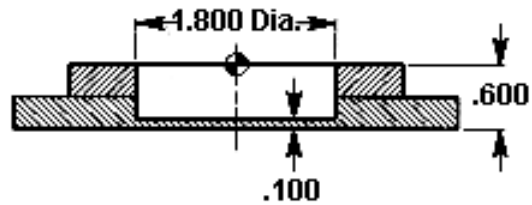
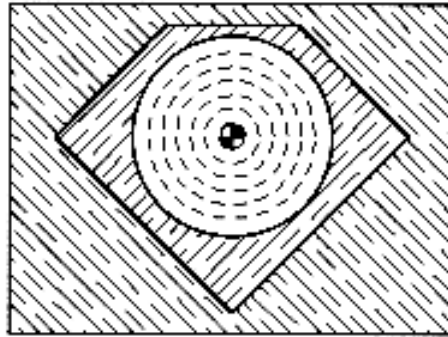
G\_\_\_ G\_\_\_ Z\_\_\_ M\_\_\_

G\_\_\_ G\_\_\_ Z\_\_\_

M\_\_\_



**CLASS EXERCISE**



**Tool #3** – 5/8 diameter 4 flute end mill – mill a 1.800 diameter circular pocket .5 depth

220 SFM - .0045 inch/rev per tooth

T\_\_\_\_ M\_\_\_\_ (tool #3 – 5/8 diameter 4 flute end mill)

G\_\_\_\_ G\_\_\_\_ G\_\_\_\_ X\_\_\_\_ Y\_\_\_\_

M\_\_\_\_ S\_\_\_\_

G\_\_\_\_ Z\_\_\_\_ H\_\_\_\_ M\_\_\_\_

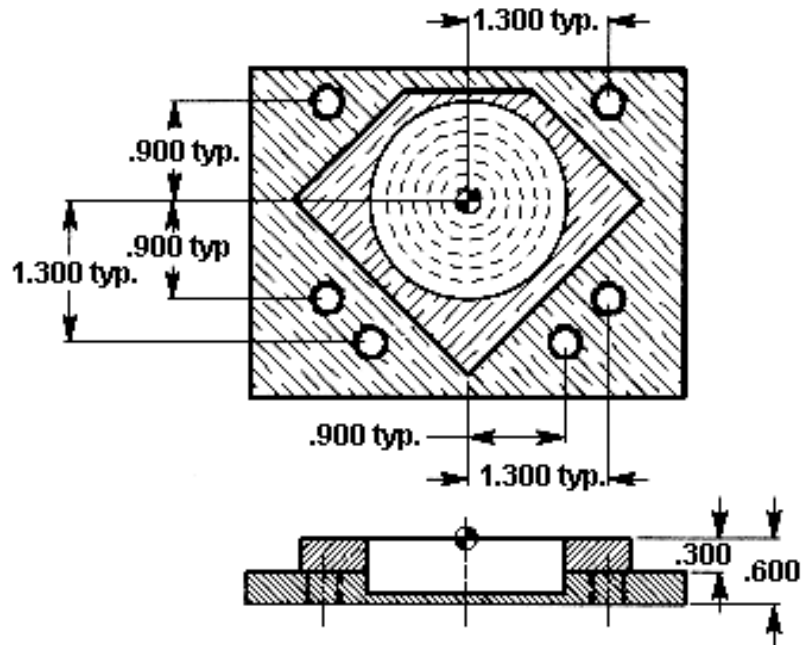
G\_\_\_\_ I\_\_\_\_ K\_\_\_\_ Q\_\_\_\_ Z\_\_\_\_ D\_\_\_\_ F\_\_\_\_

G\_\_\_\_ Z\_\_\_\_ M\_\_\_\_

G\_\_\_\_ G\_\_\_\_ Z\_\_\_\_

M\_\_\_\_

**CLASS EXERCISE**



**Tool #4** – 1/2 diameter spot drill – spot drill 6 holes .2 depth

200 SFM - .005 inch/rev

**Tool #5** – 5/16 diameter drill – drill 6 holes through

200 SFM - .005 inch/rev

**Tool #6** – 3/8 – 16 Tap (6) holes through – tap at 600 RPM

Your machine has rigid tapping

T\_\_\_ M\_\_\_ (tool #4 – 1/2 diameter spot drill)

G\_\_\_ G\_\_\_ G\_\_\_ X\_\_\_ Y\_\_\_

M\_\_\_ S\_\_\_

G\_\_\_ Z\_\_\_ H\_\_\_ M\_\_\_

G\_\_\_ G\_\_\_ Z\_\_\_ R\_\_\_ F\_\_\_ L\_\_\_

M\_\_\_ P\_\_\_

G\_\_\_ G\_\_\_ Z\_\_\_

M\_\_\_

T\_\_\_ M\_\_\_ (tool #5 – 5/16 diameter drill)

G\_\_\_ G\_\_\_ G\_\_\_ X\_\_\_ Y\_\_\_

S\_\_\_ M\_\_\_

G\_\_\_ Z\_\_\_ H\_\_\_ M\_\_\_

G\_\_\_ G\_\_\_ Z\_\_\_ R\_\_\_ F\_\_\_ L\_\_\_

M\_\_\_ P\_\_\_

G\_\_\_ G\_\_\_ Z\_\_\_

M\_\_\_

T\_\_\_ M\_\_\_ (tool #6 – 3/8-16 tap)  
G\_\_\_ G\_\_\_ G\_\_\_ X\_\_\_ Y\_\_\_  
S\_\_\_  
G\_\_\_ Z\_\_\_ H\_\_\_ M\_\_\_  
G\_\_\_ G\_\_\_ Z\_\_\_ R\_\_\_ F\_\_\_ L\_\_\_  
M\_\_\_ P\_\_\_  
G\_\_\_ G\_\_\_ Z\_\_\_  
M\_\_\_

T\_\_\_ M\_\_\_ (tool #7 1/8" ball end mill)  
G\_\_\_ G\_\_\_ G\_\_\_ X\_\_\_ Y\_\_\_  
M\_\_\_ S\_\_\_  
G\_\_\_ H\_\_\_ Z\_\_\_ M\_\_\_  
G\_\_\_ P\_\_\_ X\_\_\_ Y\_\_\_ I\_\_\_ J\_\_\_ R\_\_\_ Z\_\_\_ F\_\_\_ E\_\_\_ (YOUR NAME)  
G\_\_\_ G\_\_\_ Z\_\_\_ M\_\_\_  
M\_\_\_  
G\_\_\_ G\_\_\_ Z\_\_\_  
G\_\_\_ G\_\_\_ Y\_\_\_  
M\_\_\_

*(Subprogram)*

O1000 (subprogram)  
X-\_\_\_ Y-\_\_\_  
G\_\_\_ X-\_\_\_ Y-\_\_\_  
Y-\_\_\_  
X-\_\_\_  
G\_\_\_ Y-\_\_\_  
X-\_\_\_ Y-\_\_\_  
G\_\_\_ G\_\_\_ Z\_\_\_ M\_\_\_  
M\_\_\_

# NOTES

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---