L7 CODING EXERCISE

Just a little more for awareness!

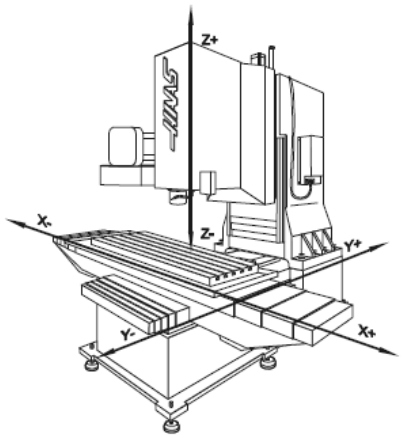


FIGURE A -TYPICAL MILL

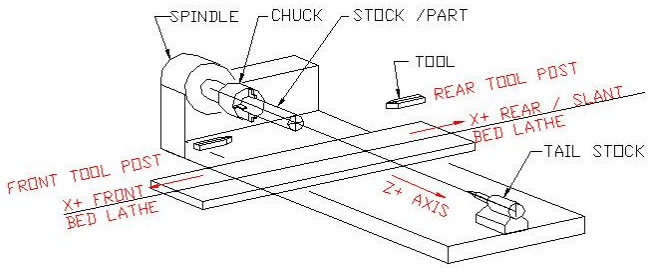


FIGURE B- TYPICAL LATH

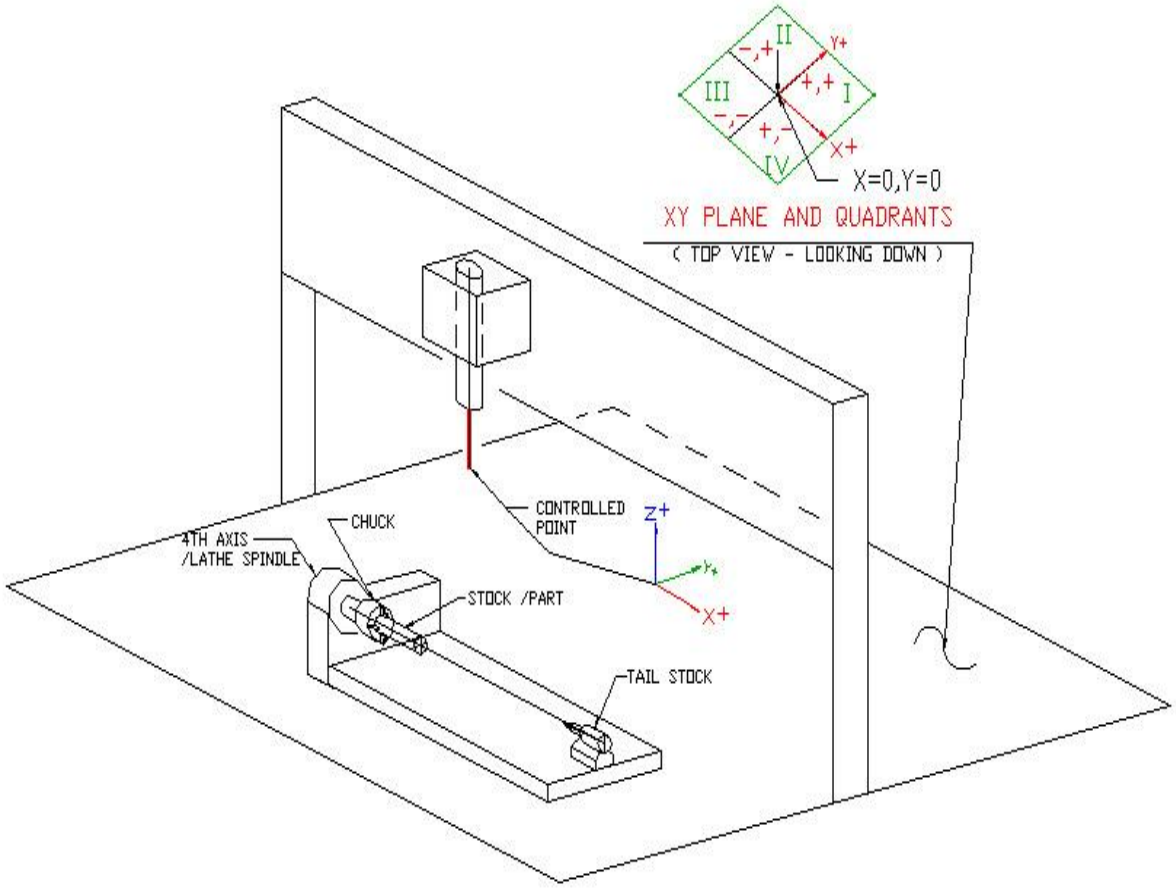


FIGURE C - TYPICAL ROUTER W/ 4TH AXIS

1.0 Right Hand Cartesian Dimensional Coordinate System (RHCS)

The figure below pictorially portrays the right hand rule which can serve as a visual aide in defining the positive axes direction and associated planes no matter how the coordinated system / hand is rotated / viewed.

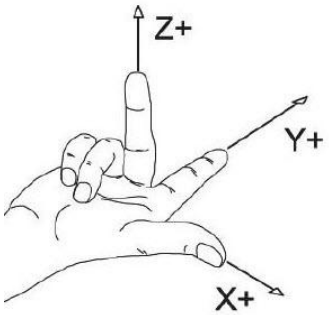


FIGURE 1.0

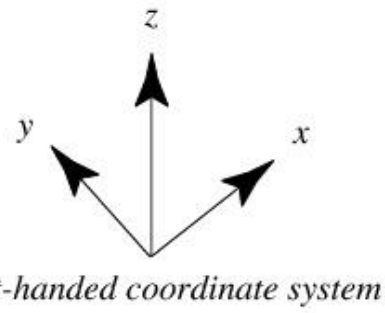


FIGURE 1.1

Note the following:

- For three-dimensional systems, a "convention" is to portray the xy-plane horizontally with the z-axis added to represent height. Note positive directions.
- In different CAD or CAM the "*convention*" of orientation of the three axes may be arbitrary but the orientation of the axes relative to each other should always comply with the right-hand rule.
- All laws of math use the right hand Cartesian Dimensional Coordinate System as it provides for consistency. Programs such as Fusion 360, Fanuc 16C Controller, Mach, OpenGL use the right hand rule.

2.0 QUADRANT and PLANES

Figure 2.0 below portrays quadrants for the XY math plane.
Every plane has four quadrants and they use roman numerals to name the quadrants of the plane, namely I,II,III,IV. Note that the how the sign for a point in the plane changes in the quadrant, and , that a common reference point is 0,0.

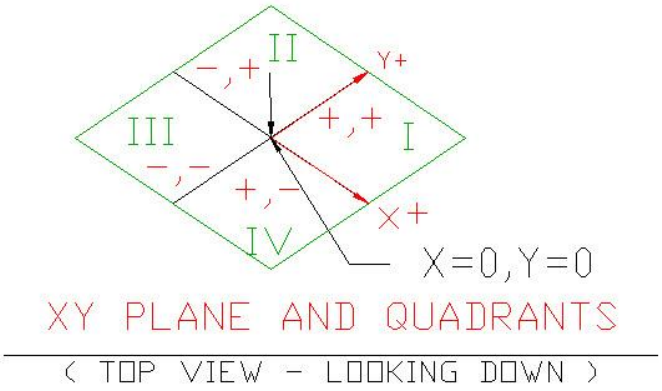


FIGURE 2.0

There are three math planes XY, YZ, ZX. The standard for math planes is the first letter defines the Horizontal axis and the second letter defines the Vertical axis. The black arrows show viewing direction perpendicular to the plane. Only the I quadrant of the plane is shown the figures below. *The plane is only partially shown for viewing ease.*

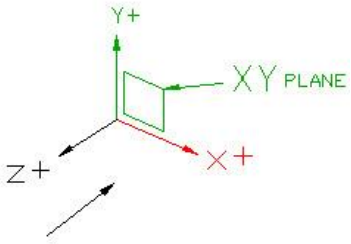


FIGURE 2.1

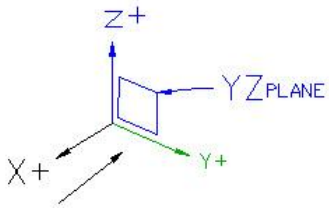


FIGURE 2.2

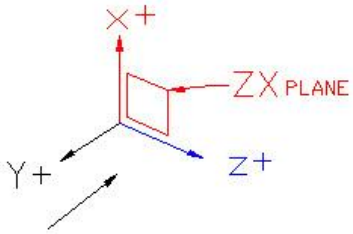


FIGURE 2.3

The physical plane is the same if an axis is rotated. Note the black arrows which show viewing direction.

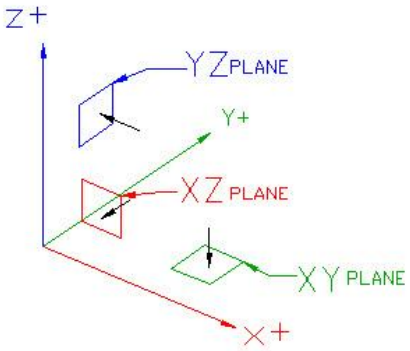


FIGURE 2.4

SOFTWARE COMMENTS

- You can't do true 3D work as that requires a 5 axis machine and CAD/CAM involving 3D modeling and code generation.

An indexed 4th axis does not rotate continuously as the toolpath is running. Instead, a number of angles are defined when the part is loaded. The toolpath will index the 4th axis to each angle and then run a complete 3-axis toolpath. An indexed 4th axis is ideal for machining a part from several sides without having to manually reposition the part.

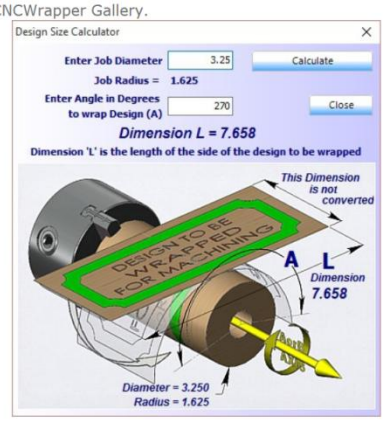
- Most lower end software just slices the XY plane into 2D depths and provides the pathing with consideration of the tooling used. The programs provide for adjusting the controlled point to achieve roundness when machining. You can have 1000's of lines of code and you still need to design / draft the part in 3D.

- You always need to consider how you want to machine the work.

- Some programs allow for indexing (rotate /cut linear, linear, repeat.....).
Some combine rotational moves with the linear axes. They are similar but different in what is provided.

One example is Wrapper, which takes 2D gcode and modifies the code so the pathing is wrapped around a defined diameter.

CNCWrapper Gallery.



You can a do a lot "basic" work without any software by hand coding. One can manually manipulate code from Wizards or Lathe programs for use in indexing operations. You can do unconventional coding, some would call it "tricking Mach3"…I hate that term, rather say, I provide the proper code to machine what I want and the controller just provides for the movement.

A simplistic look at typical "gcode program" could be as follows:
1. Program comments – remarks about the program
2. Preparatory commands – all commands have a typical sequence, definition of machine state
3. Intial commands for axis movement
4. Lines of code – all commands, individual or combined, cycles, calls to macros / subroutines
5. End of program commands

As so noted before, all info given here is "very focused" relative to the 4th axis, even the commands. Go read the first 150 pages of the Smid book, the Mach3 manuals for details. And also start learning CAD.
Just think about this:
"If one could draw a straight line, a circle, an arc, and if one could properly command the equivalent in CNC at a desired feedrate all the other "stuff" available just assists in getting it done." I have only made you "aware" of many things.

So lets get going with gcoding to define PATHING of the controlled point. Pathing is just devoid of a lot of "stuff" but must be defined at some level for it to make sense.

- It is assumed that axis motor tuning is correct, the axes are calibrated, is configured properly as depicted in Figure C.
- You will be manually coding for 2D work in the G17 (XY plane).
- Machine Coordinates and Work Coordinates have the same reference point of 0,0,0, so there are no work offsets (G54). M1000 macro ring a bell?
- Basic commands will be used for XYZ +A for G91, G94 modes.
- Only feedrates descriptions 1 and 2 in G94 mode will be used.

Oh BTW,
Create the coding using a text editor such as notepad editor and save the file 7E1 .TXT.
(LESSON 7, EXERCISE, 1 is indicative of the exercise number, SO, 7E1 seems logical)
Add line numbers to each coded line.

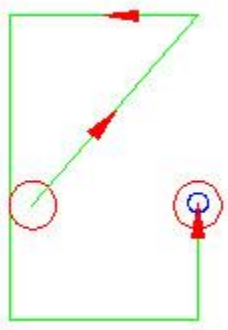
Preparatory commands in the program alter the current machine state as necessary before running the code. Look at the Mach screen upper left corner to see what is current.

ie; 

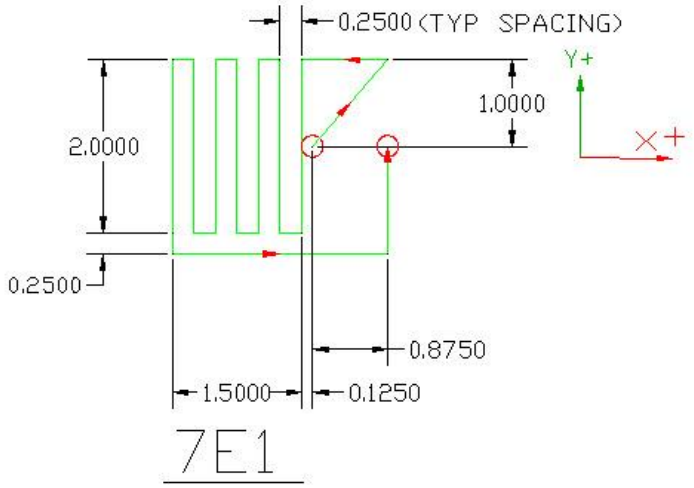

```
N0 ( THE  FUN FINALY STARTS)
N10 G91
N20 M1000   (XYZA  axis reference is set to zero)
N30  G0        (start of gcode from reference point)
N 40 G0        (continue coding, G0 is modal but have it in the code)

N100  M30  (end of program and rewind )
```

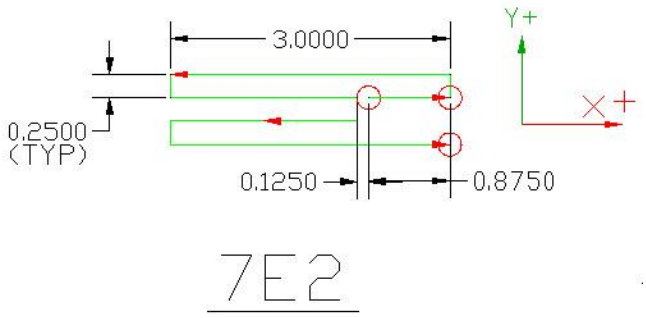
Typical in the following pictures:
- Red arrow shows direction of the path from start to finish.
- Red circle is the tool and the controlled point is the center and face of the tool.
- Blue circle is a change in the Z



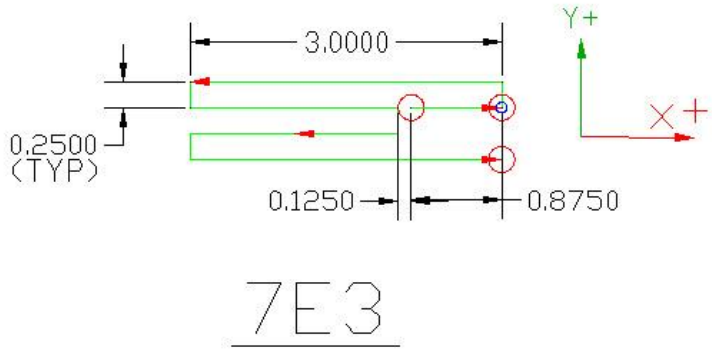
Create a program of gcode for 7E1.



Create a program of gcode for 7E2.



Create a program of gcode for 7E3 by modifying 7E2.



Now create programs for 7E1,7E2,7E3 for absolute moves (G90).
Can put an I or A after the name so you know which file is what mode.

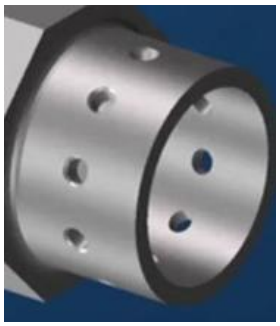
7E4

Lets do some index coding. But before we start another code exercise lets add another command, namely G4 which is defined below. It will be used just to "mimic" some operation being done when the A axis has moved to position.

Dwell - G4 For a dwell, program G4 P~ . This will keep the axes unmoving for the period of time in seconds or milliseconds specified by the P number. The time unit to be used is set up on the Config>Logic dialog. For example, with units set to Seconds, G4 P0.5 will dwell for half a second. It is an error if: the P number is negative.

So in this exercise, do a program for putting 8 holes equally spaced around the cylinder pictured below with the holes located 1 inch from the face. Assume the reference position of XYZA is at 0,0,0,0, which happens to be the center and face of the cylinder, and the tool is a center cutting end mill 0.25" in diameter with its diameter touching the cylinder face. Make some program / remarks in the beginning of the code which you think are relevant and use a value of 10 seconds for P in G4. The coding should be done for G91.

Remember what you did in lesson # 4 ?. Now you will be adding linear moves to position the tool to drill do the holes.

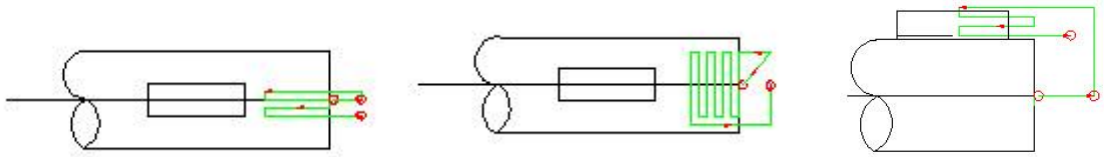


7E5

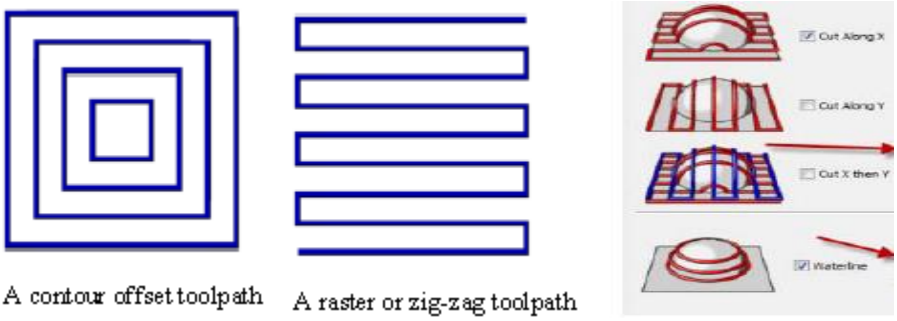
Same as TE4 but modify to add a second set of holes 2" from the face and the coding should be done for G90.

7E6

The figures below show the exercise coding overlaid above a cylinder (with a block on top in the first two pics and a block on the side of the cylinder for the last one.



- Stepover of adjacent pathing is very important in machining as the value directly relates to finish quality and is based on the diameter and type of tooling. The programs done were basic raster paths. Most programs apply the rastering in different directions relative to the material. See pictures below.



How to Choose a Stepover is explained very well on the Grzsoftware Site.
Here is a link:
http://www.grzsoftware.com/blog/how-to/choose-stepover/

So 7E6 is simply to think about the different ways of machining the block on top of the cylinder. Also just the cylinder using indexing.
Consider tool diameters, tool type, stepover, scalloping, surface finish desired, and simple single axis moves and what machining will be done to the work.

To be continued,
RICH