Welcome to Trident Machine Tools

Mill Program Troubleshooting

Mill Program Troubleshooting

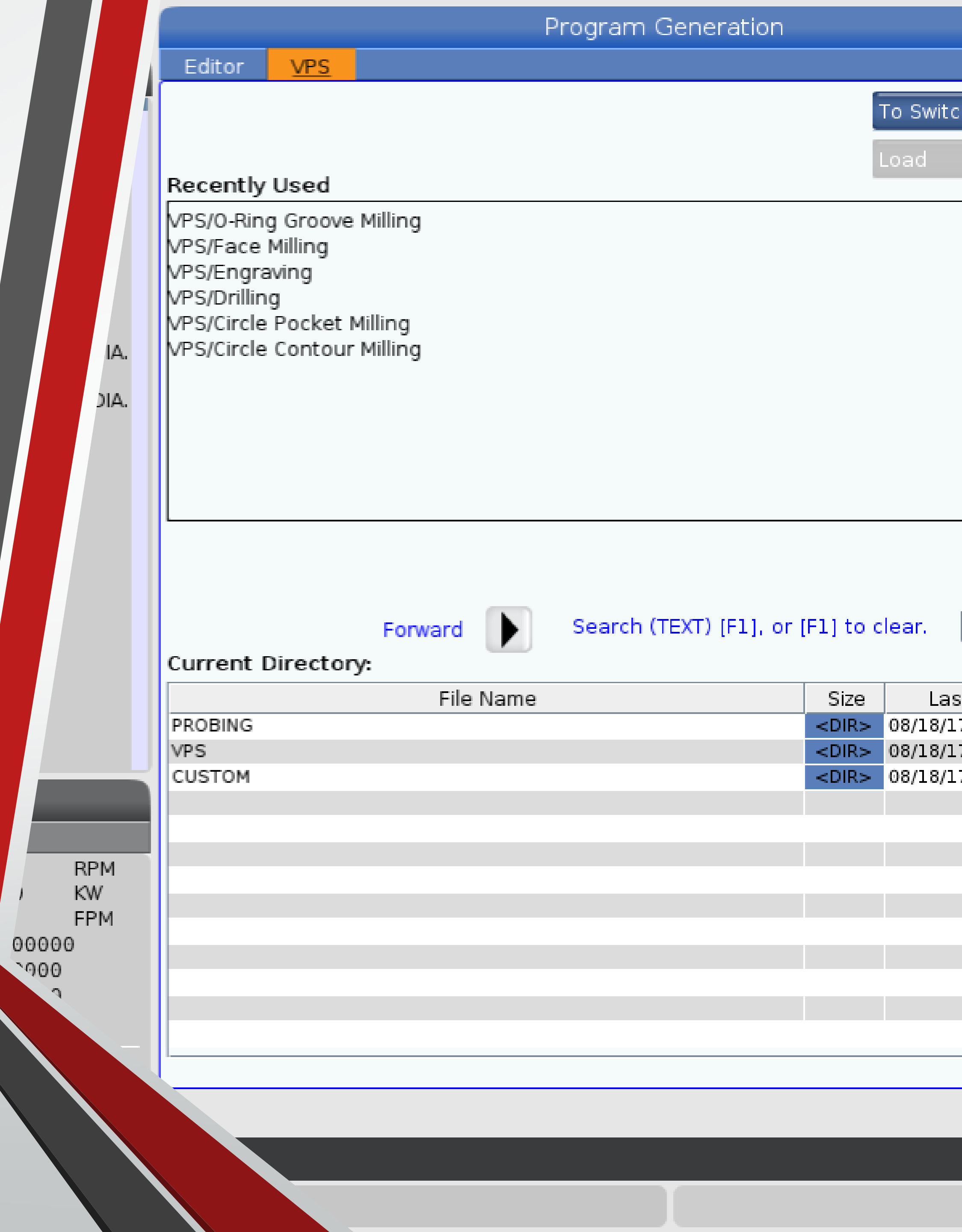
- This one day course is designed to provide the user with a basic understanding of basic mill programming and trouble shooting common program issues.

Schedule

- Introductions
- VPS overview
  - VPS demo
- Break
- Program structure
- Cutter Compensation
- Canned cycles
- Lunch
- Canned cycles
- Break
- Program trouble shooting
- Questions

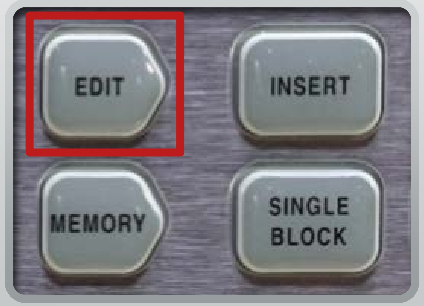
VPS Programming Overview

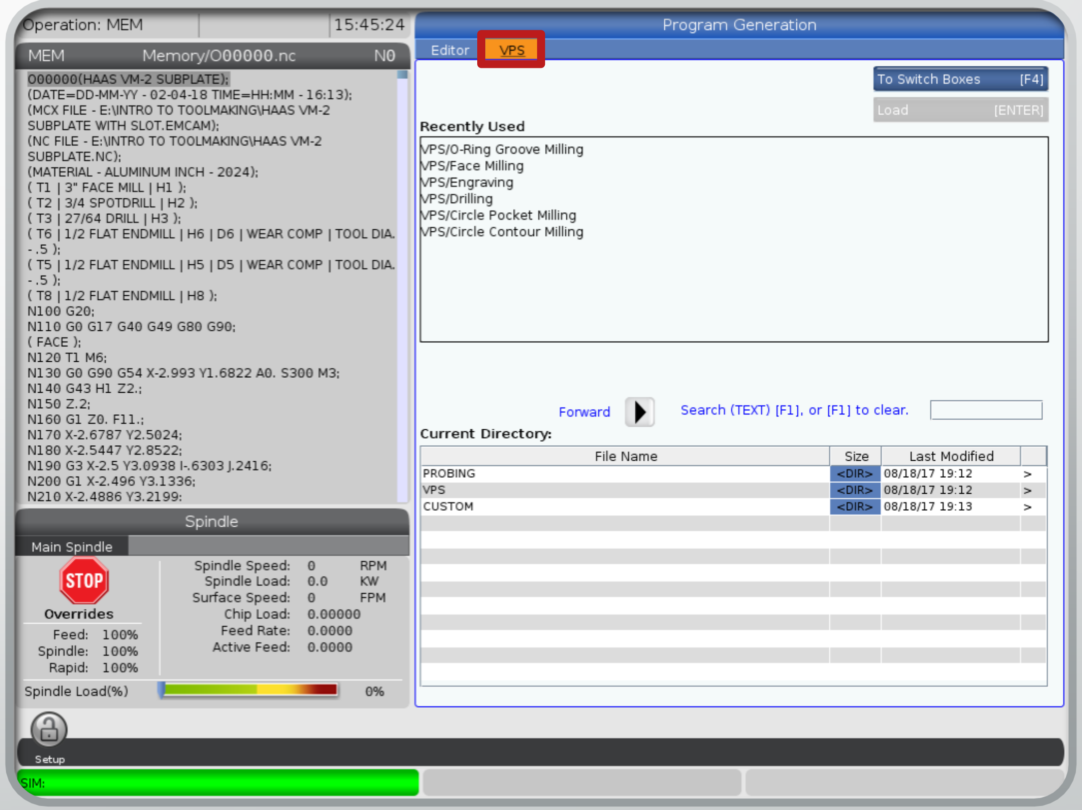
- VPS allows for conversational programming. This entails inputting values in a menu to output the needed code for the particular situation.
- For mill this might entail facing, drilling, contouring, among other options.



VPS Example – Circle Contour Milling

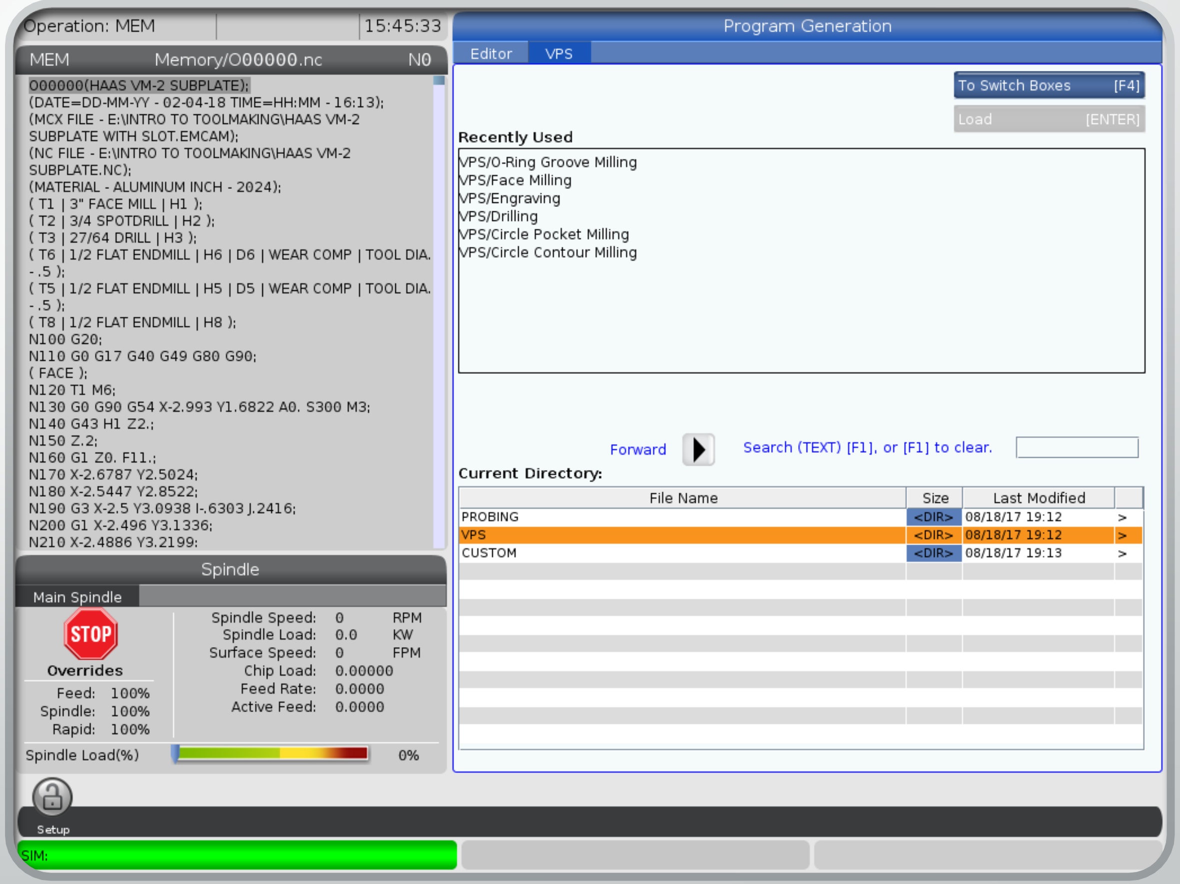
- The following is an example in how to make a VPS program, specifically, contour milling.
- Start by selecting the “Edit” button, then shift over to VPS from editor.





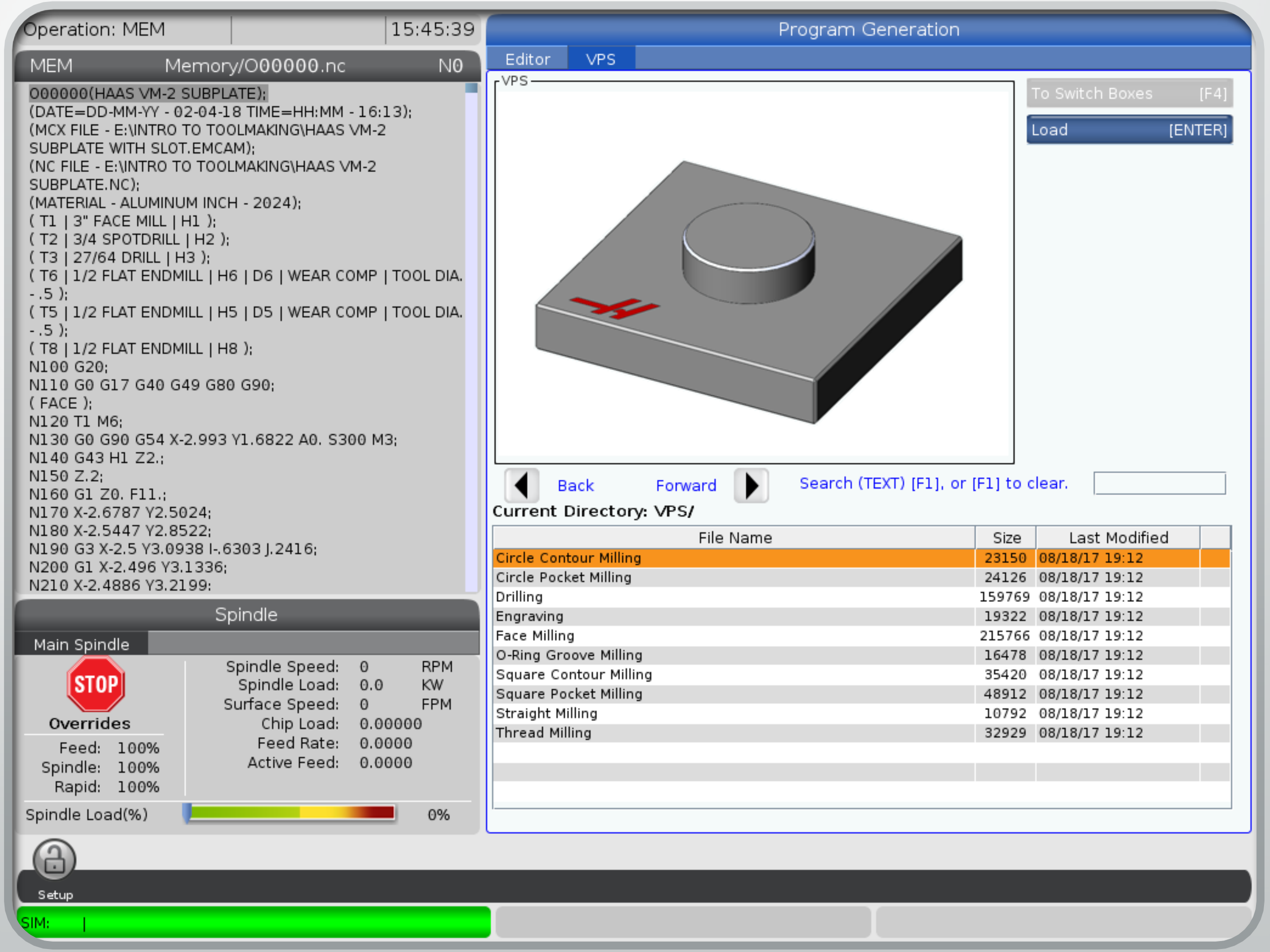
VPS Example – Circle Contour Milling

- Cursor down to VPS in the menu and select it.
  - Don't forget that the arrow keys are used to navigate the Haas menus.



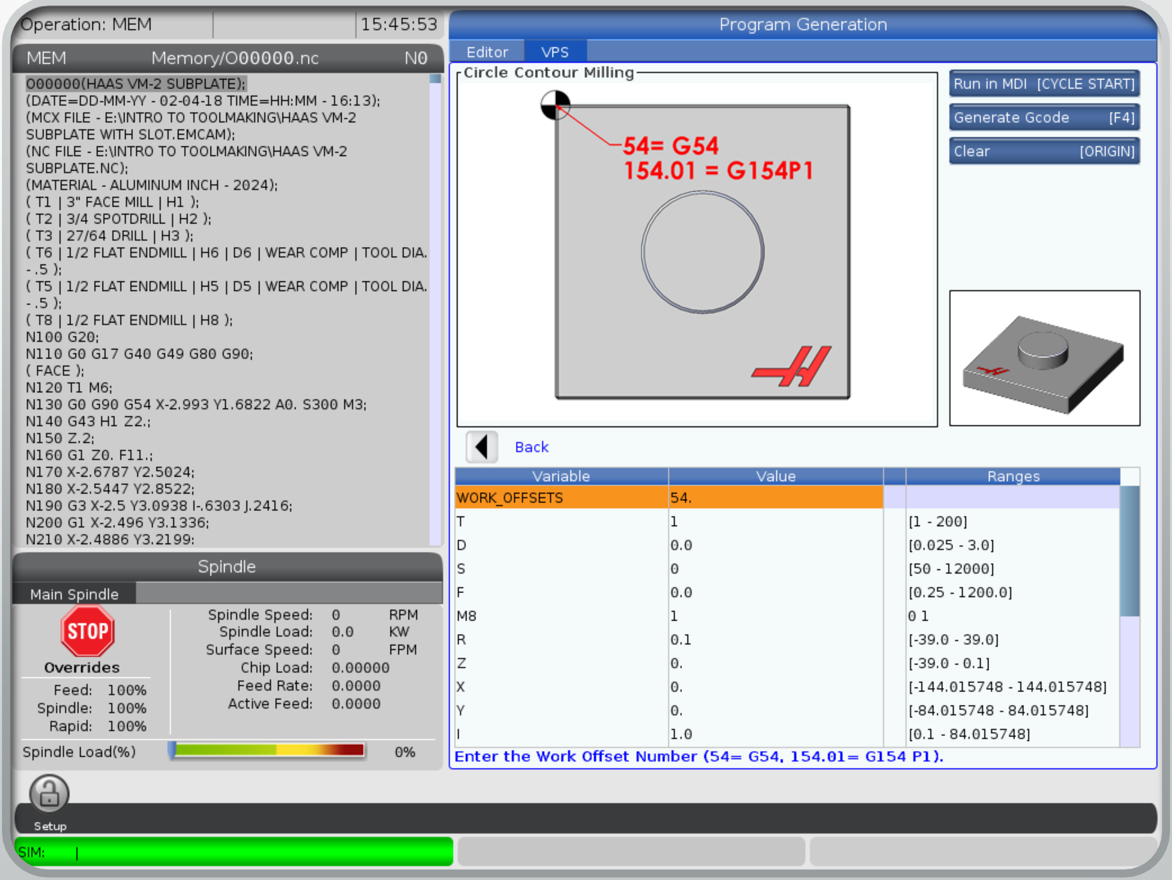
VPS Example – Circle Contour Milling

- Select “Circle Contour Milling”.



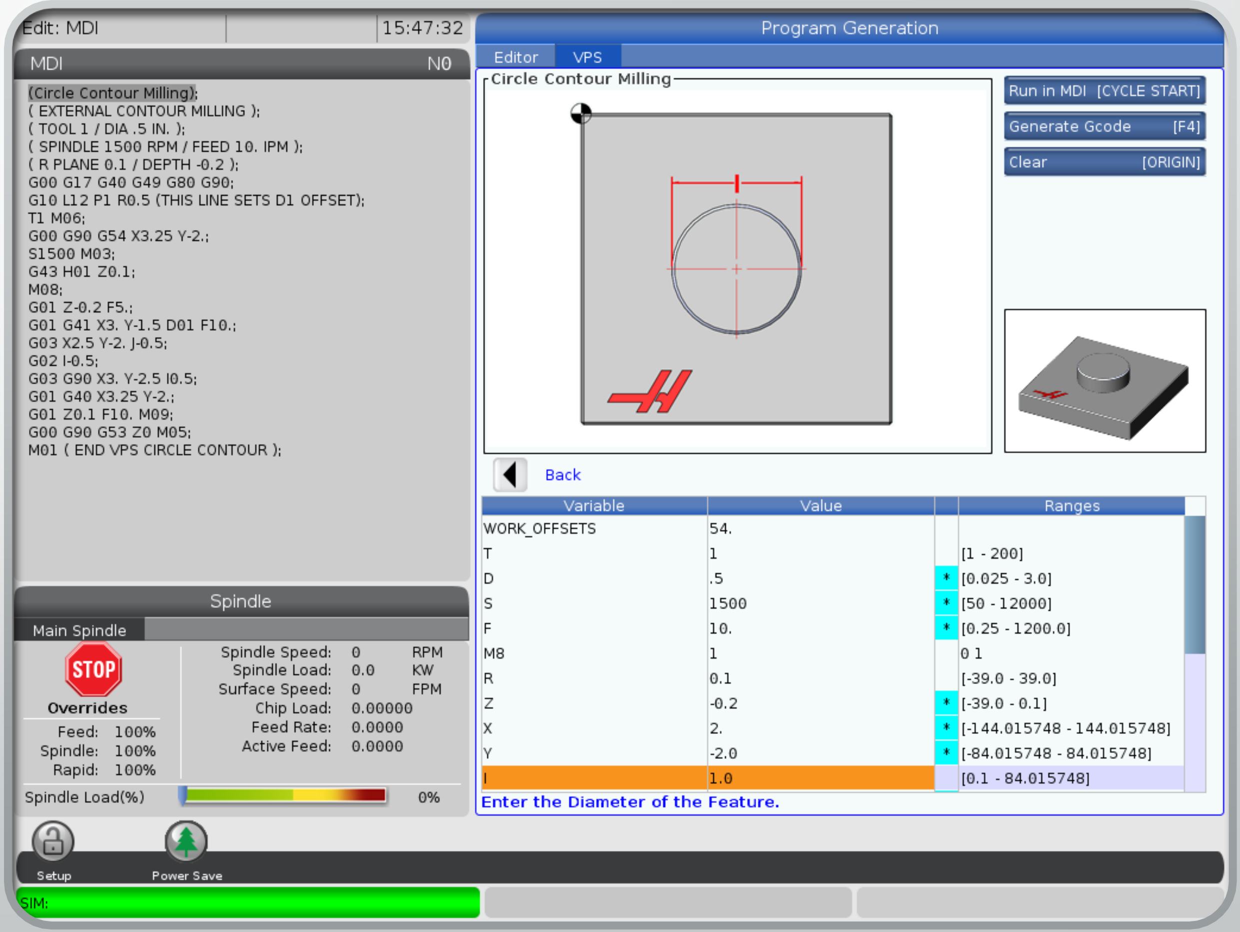
VPS Example – Circle Contour Milling

- Start filling out the variables in the form.
- The prompts will walk the operator through each inquiry. In this case, it says to insert the desired work offset number.



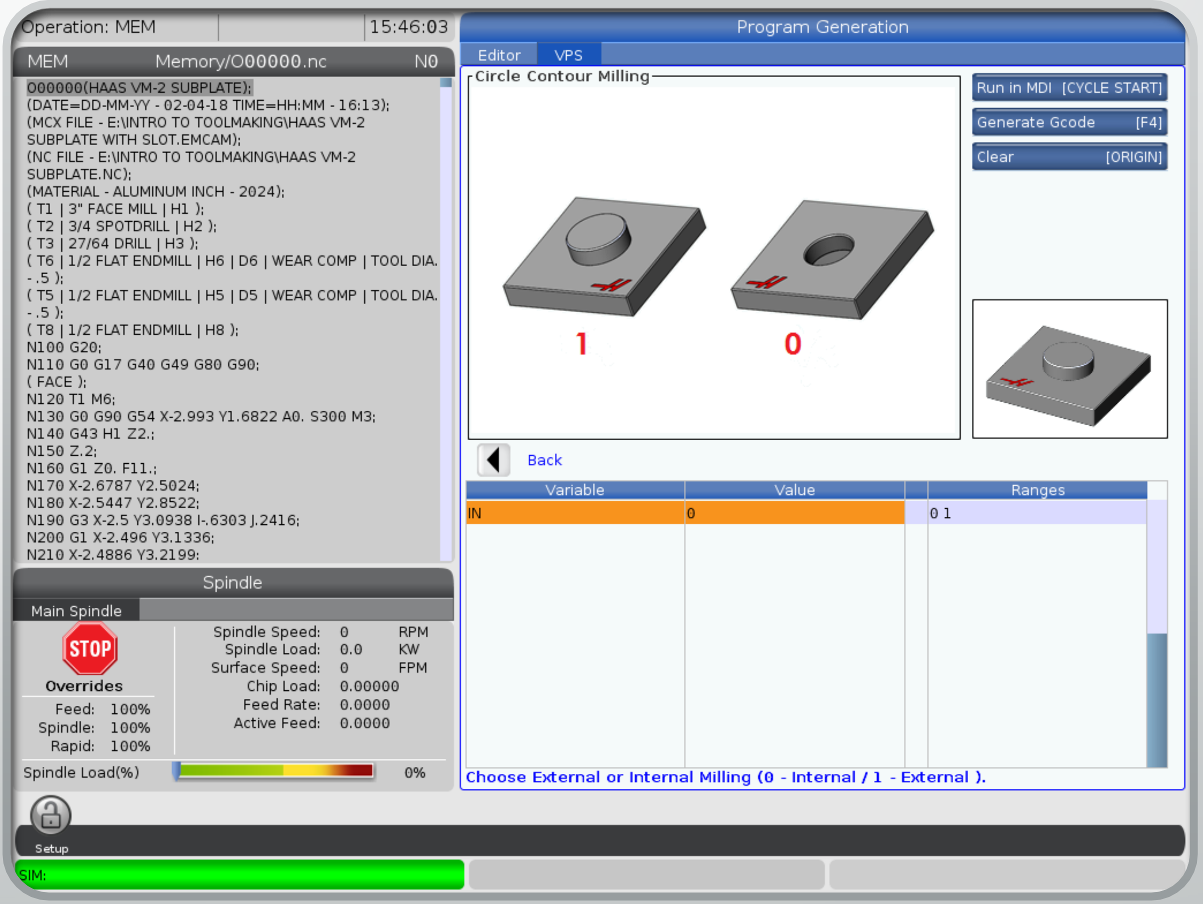
VPS Example – Circle Contour Milling

- After the variables are filled in, it should look something like this.
- This tells the machine how to machine the part in relation to the PRZ.



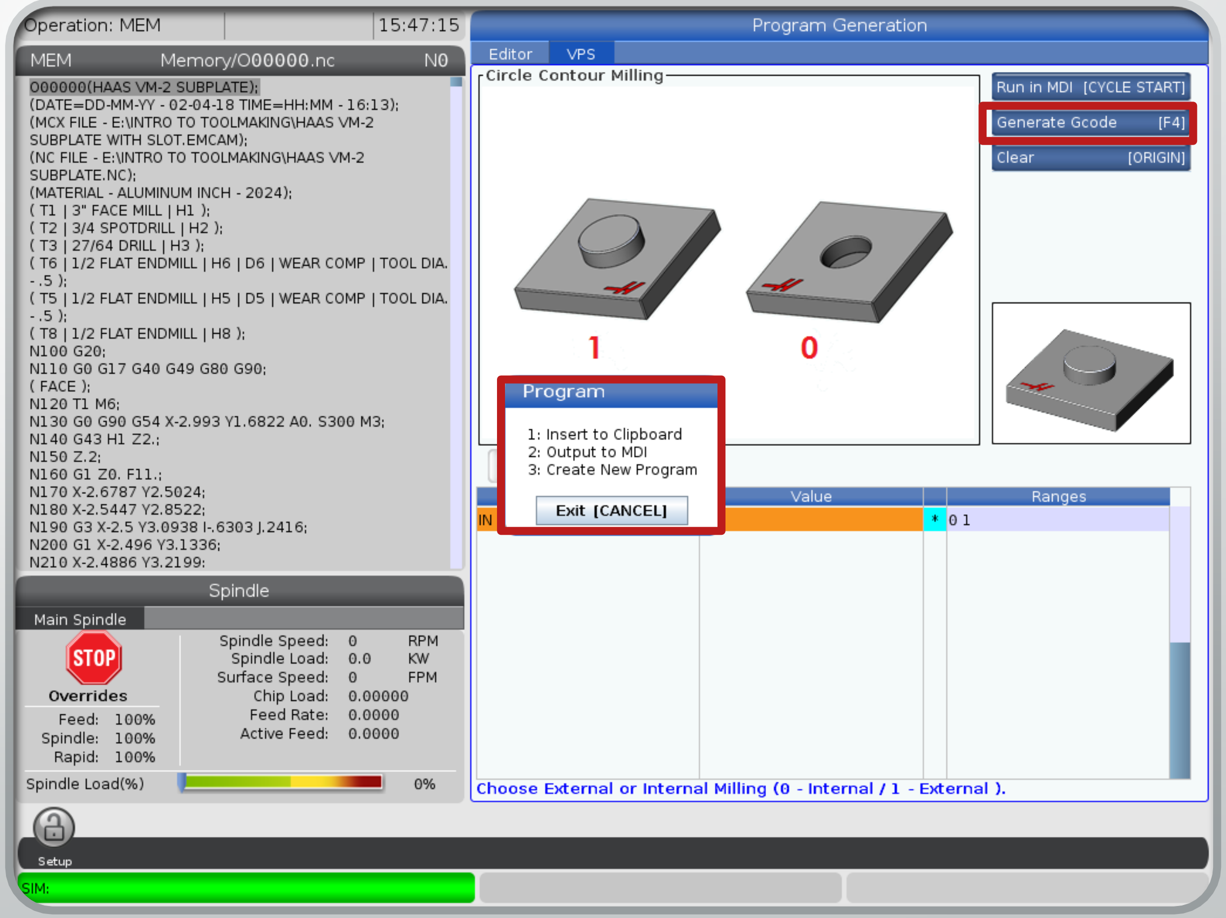
VPS Example – Circle Contour Milling

- The next screen asks whether the feature is a boss or pocket. In this case it is a pocket.



VPS Example – Circle Contour Milling

- Select generate code to bring up the output menu.
- Select "2", which will output the code to MDI.

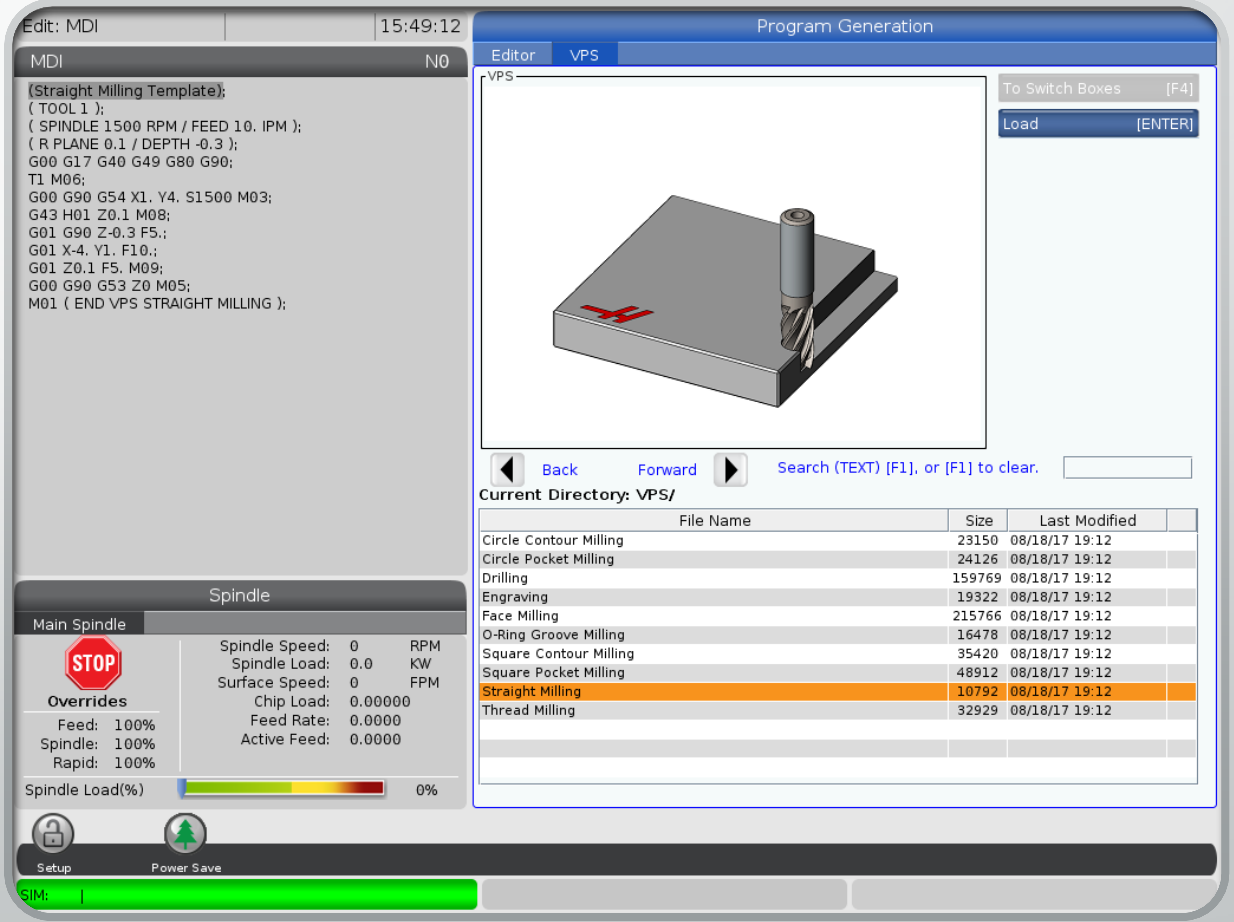


VPS Example – Circle Contour Milling

- The program is then output to MDI.

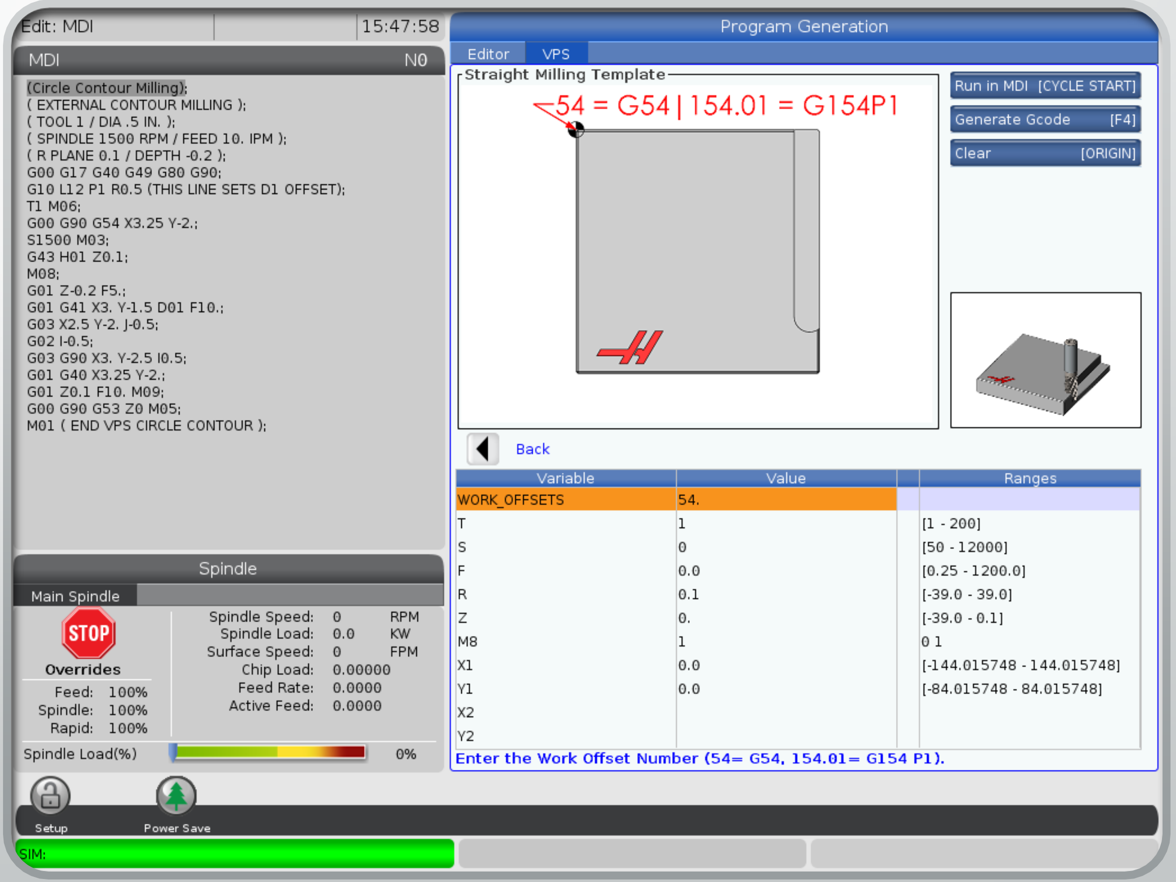
VPS Example – Straight Milling

- This next VPS programming example will go over how to program a "Straight Milling toolpath".
- Start by going back to Edit/VPS, and select "Straight Milling" from the VPS menu.



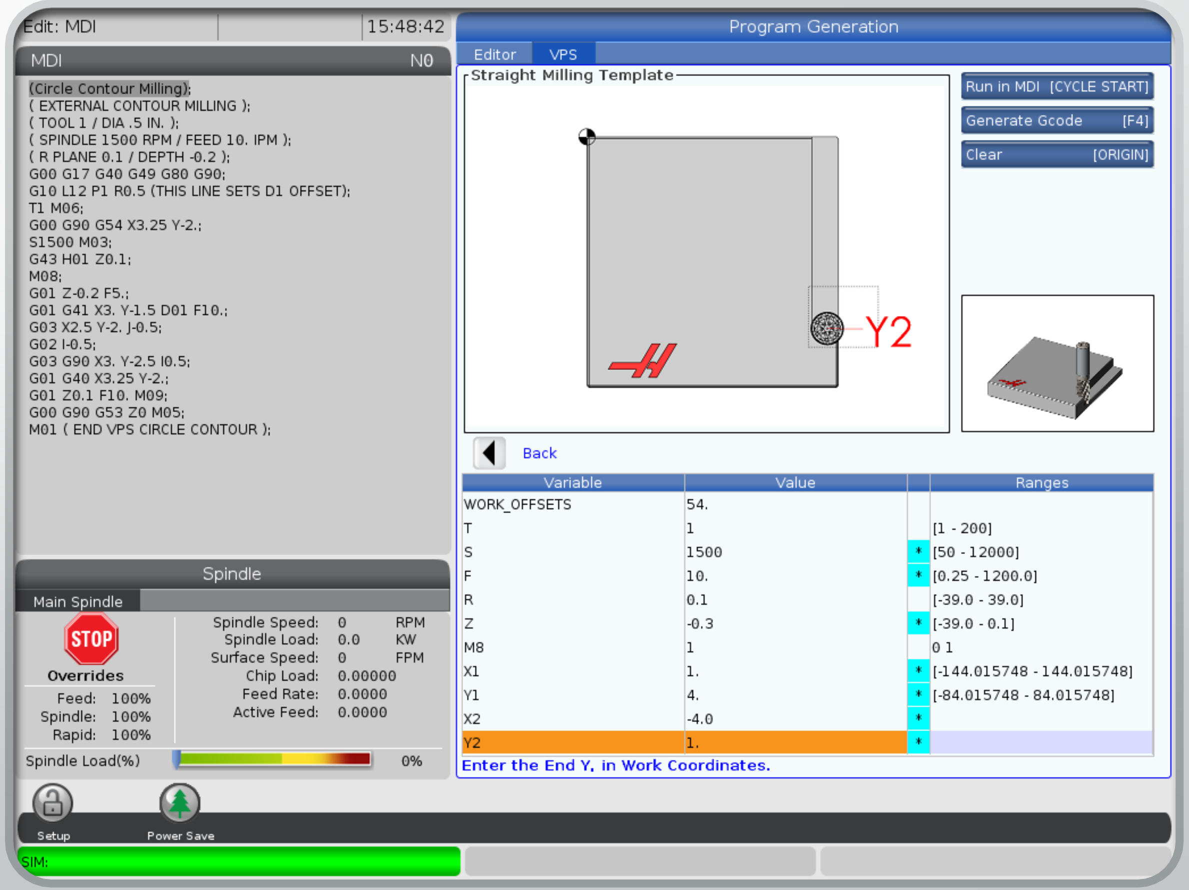
VPS Example – Straight Milling

- Again, fill in the variables. The first inquiry asks which work offset to use.



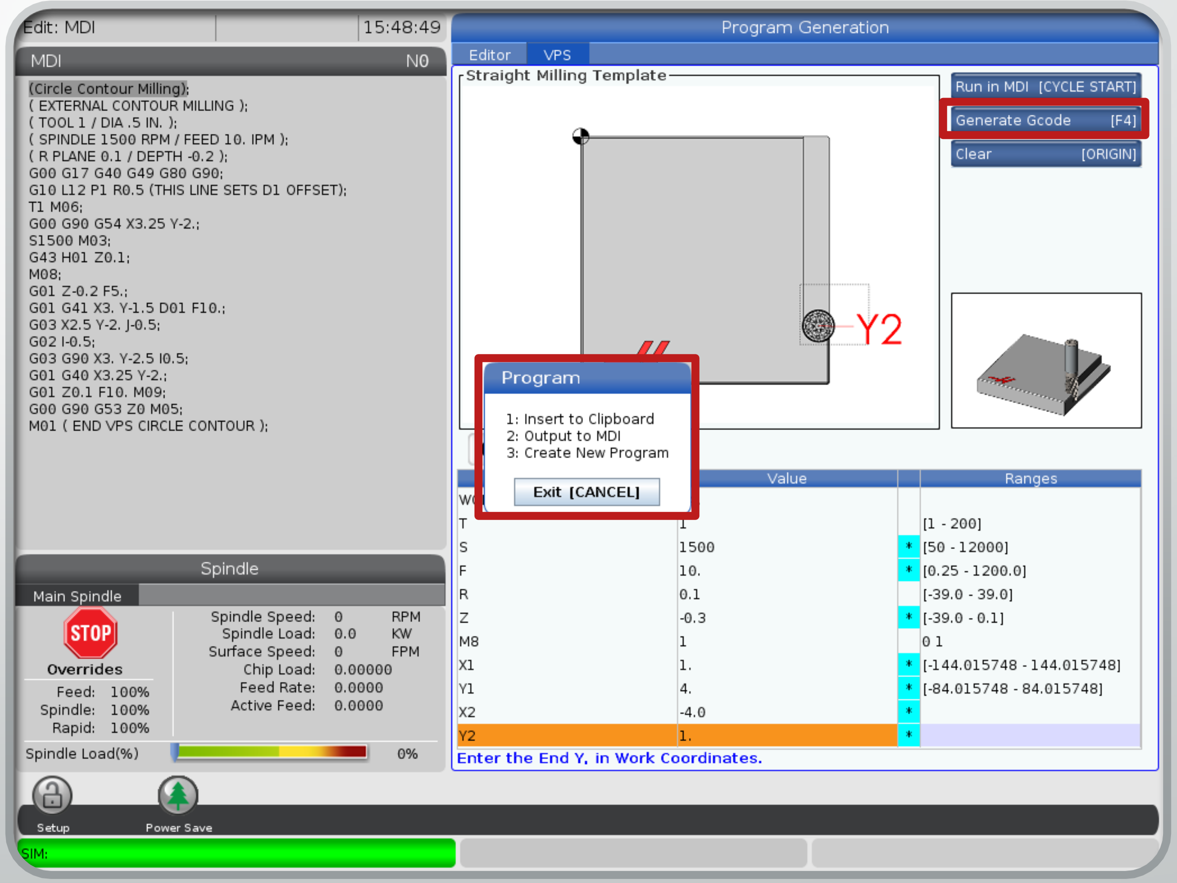
VPS Example – Straight Milling

- Fill out the variables like so. This tells the machine how to cut the part in relation to the PRZ.



VPS Example – Straight Milling

- Select "Generate Gcode" by pressing F4 as prompted. Select "2" to output to MDI.



VPS Example – Straight Milling

- The program is then output to MDI.
- Notice that both of these examples were very similar. VPS is designed to be intuitive, and should be easy to navigate using prompts.
- This code can also be posted to memory, or even saved to the clipboard for later use.

Program Structure

- When starting a new program, the material, stock size, PRZ, and tool list should be included in the beginning of the program as notes.
- This verifies that the correct stock, tools, and PRZ have been set-up.
- Each operation should have a note of what is happening. This helps the person operating the machine.

```
%
O00001 (MILL FORMAT);
N1 (NOTES)
G00 G17 G20 G40 G49 G54 G64 G80 G90 G94 G98 G100;
G53 Z0.;
T1 M06;
X0. Y0. S1000 M03;
G43 H01 Z1. M08;
(BEGIN TOOL BODY);
;
;
;
;
(END TOOL BODY);
G00 G80 Z1. M09;
G49 G53 Z0.;
M01
```

Program Structure

- Programs are broken up into 3 basic sections:
  - Start-up
    - In the example program to the right, line 4-8 will be the same sequence for each tool path we program. The variables such as RPM, feed rate and positions are the only thing that change.
  - Cutting
    - In the example program this would be lines 9-14. This will vary in length and complexity depending upon the operation.
  - Shut Down
    - In the example program this would be lines 15-17. These lines will have the same format for each tool that is programmed.

```
%
O00001 (MILL FORMAT);
N1 (NOTES)
```

Start-up

```
G00 G17 G20 G40 G49 G54 G64 G80 G90 G94 G98 G100;
G53 Z0.;
T1 M06;
X0. Y0. S1000 M03;
G43 H01 Z1. M08;
```

Cutting

```
(BEGIN TOOL BODY);
;
;
;
;
(END TOOL BODY);
```

Shut down

```
G00 G80 Z1. M09;
G49 G53 Z0.;
M01
```

Tool Path Startup

- The first 4 lines of each tool path contain the start up of the tool. This should be done for each toolpath.
- The format is as follows:
  - Safe start-up line
  - Send Z home
  - Send X & Y home, Tool change/offset
  - Turn spindle on

```
%
O00001 (MILL FORMAT);
N1 (NOTES)
G00 G17 G20 G40 G49 G54 G64 G80 G90 G94 G98 G100;
G53 Z0.;
T1 M06;
X0. Y0. S1000 M03;
G43 H01 Z1. M08;
(BEGIN TOOL BODY);
;
;
;
;
(END TOOL BODY);
G00 G80 Z1. M09;
G49 G53 Z0.;
M01
```

Tool Path Startup

- Safe Start-up
  - This line is used to prepare the machine for cutting:
    - G00 – Rapid
    - G17 – Set work area plains to X & Y
    - G20 – Inch
    - G40 – Cancel tool nose radius comp.
    - G49 – Cancel tool length comp.
    - G54 – Active work offset
    - G64 – Cancels exact stop (G61)
    - G80 – cancel canned cycle
    - G90 – Set dimensioning to absolute
    - G94 – Deactivates Inverse Time Feed Mode – Returns the control to Feed Per Minute mode.
    - G98 – Set Canned Cycle Initial Point Return – Canned cycles retract to the start point instead of the retract point.
    - G100- Turns off mirror imaging

```
%
O00001 (MILL FORMAT);
N1 (NOTES)
G00 G17 G20 G40 G49 G54 G64 G80 G90 G94 G98 G100;
G53 Z0.;
T1 M06;
X0. Y0. S1000 M03;
G43 H01 Z1. M08;
(BEGIN TOOL BODY);
;
;
;
;
(END TOOL BODY);
G00 G80 Z1. M09;
G49 G53 Z0.;
M01
```

Tool Path Startup

- T1 M6
  - This calls up tool 1.
  - An M6 is required for the tool to change.
- X0. Y0.
  - This moves both X and Y to the work PRZ.
- S1000 M3
  - This turns the spindle on clockwise at 1000 RPM.

```
%
O00001 (MILL FORMAT);
N1 (NOTES)
G00 G17 G20 G40 G49 G54 G64 G80 G90 G94 G98 G100;
G53 Z0.;
T1 M06;
X0. Y0. S1000 M03;
G43 H01 Z1. M08;
(BEGIN TOOL BODY);
;
;
;
;
(END TOOL BODY);
G00 G80 Z1. M09;
G49 G53 Z0.;
M01
```

Shut Down

- After the cutting is complete, the tool path can be shut down. This is done in the final 3 lines of the program. The shut down procedure places the machine in rapid, turns off the spindle and coolant (if used), sends Z then X and Y home (separately to avoid collisions), and then uses an optional stop.
  - G0 G80 Z1. M09
    - This puts the machine back in rapid, cancels canned cycles, lifts Z 1” above the PRZ, and turns off the coolant.
  - G49 G53 Z0.
    - This cancels tool length comp. while returning Z to home.
  - M1
    - This stops the program when optional stop is active. This is most commonly used at the end of a cycle.

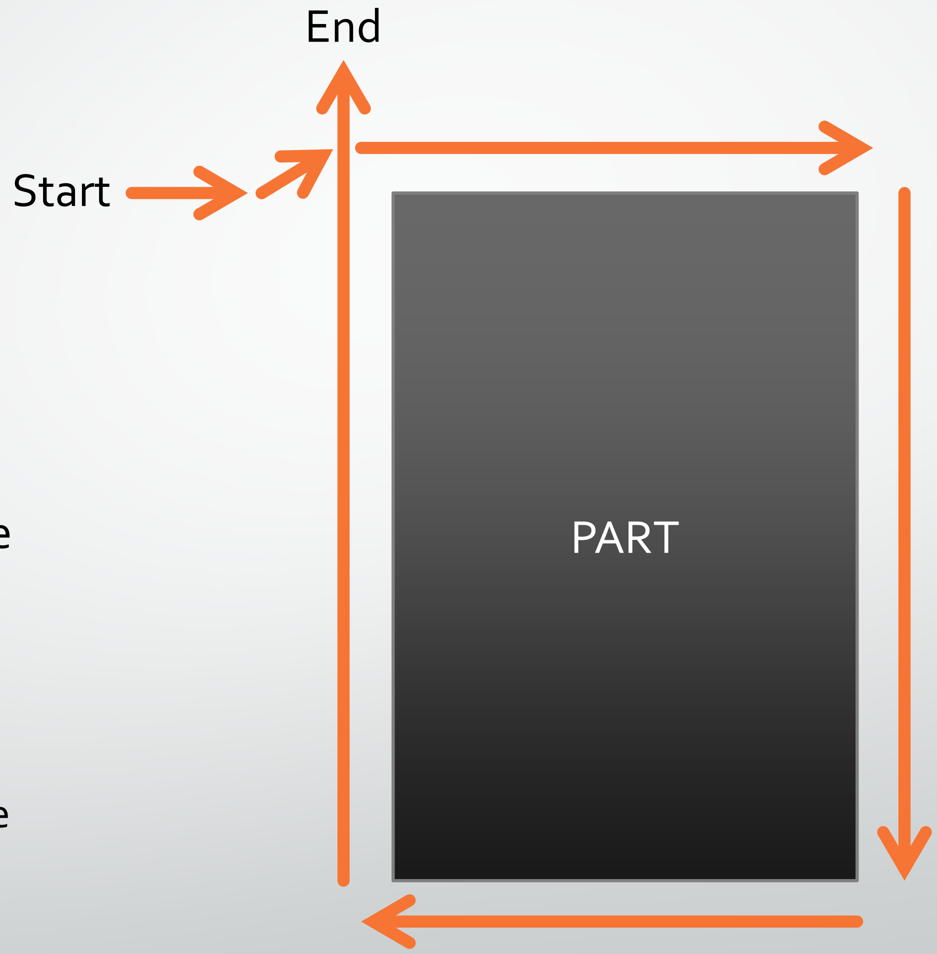
```
%
O00001 (MILL FORMAT);
N1 (NOTES);
G00 G17 G20 G40 G49 G54 G64 G80 G90 G94 G98 G100;
G53 Z0.;
T1 M06;
X0. Y0. S1000 M03;
G43 H01 Z1. M08;
(BEGIN TOOL BODY);
;
;
;
;
(END TOOL BODY);
G00 G80 Z1. M09;
G49 G53 Z0.;
M01
```

Writing Safe Programs

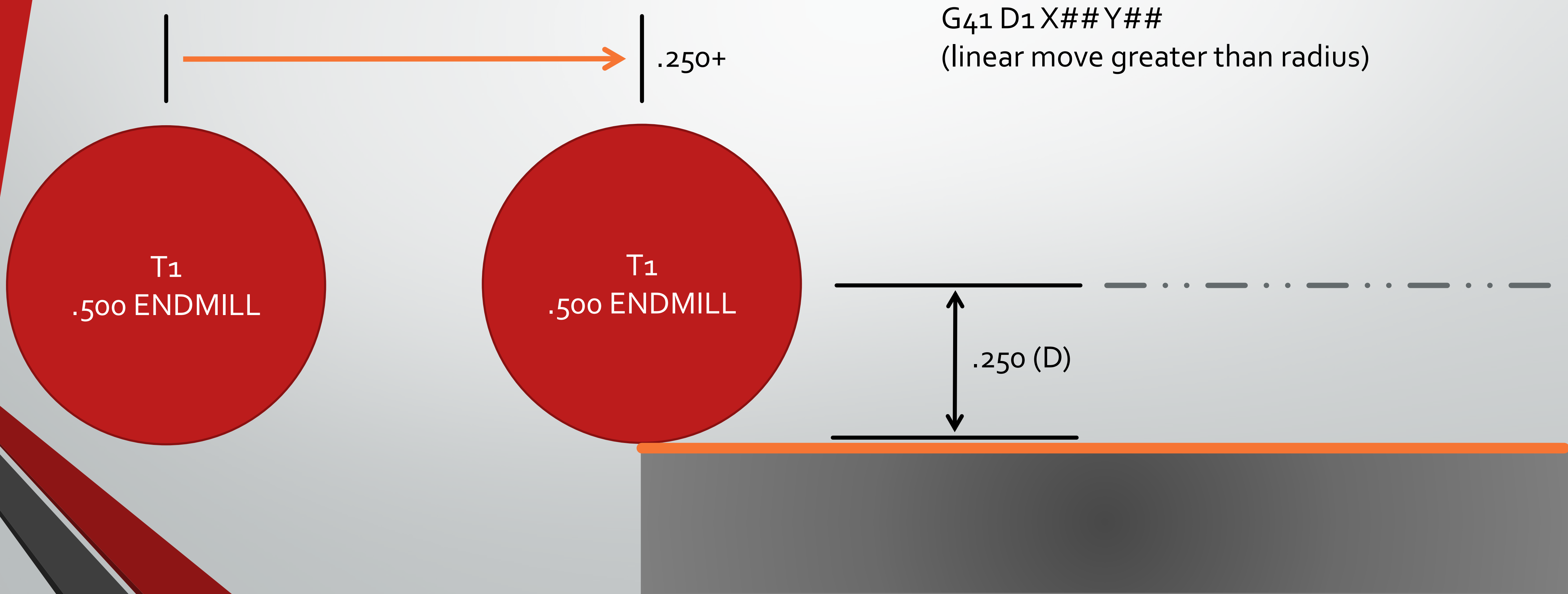
- Regardless of how many tools are used, each operation should be written as a separate program.
- This means that each tool has the proper safety start-up line, calls the tool change each time, and the cycle ends with the tool going to Z then X & Y home.
- Writing programs this way ensures that any operation in the program can be re-run without issue.
    - I.E. – If a tool is used twice in a row in a program and a person does not write in a tool change for the second operation, a machinist can not rerun the second operation without manually changing the tool.

Diameter Cutting Compensation

- Diameter comp. factors in the tools diameter and offset the tool while cutting. It does this by shifting over away from the part the distance of the tool radius.
- Cutter compensation can only be activated on linear moves.
  - Cutter compensation will not activate on G0, G2 or G3 moves.



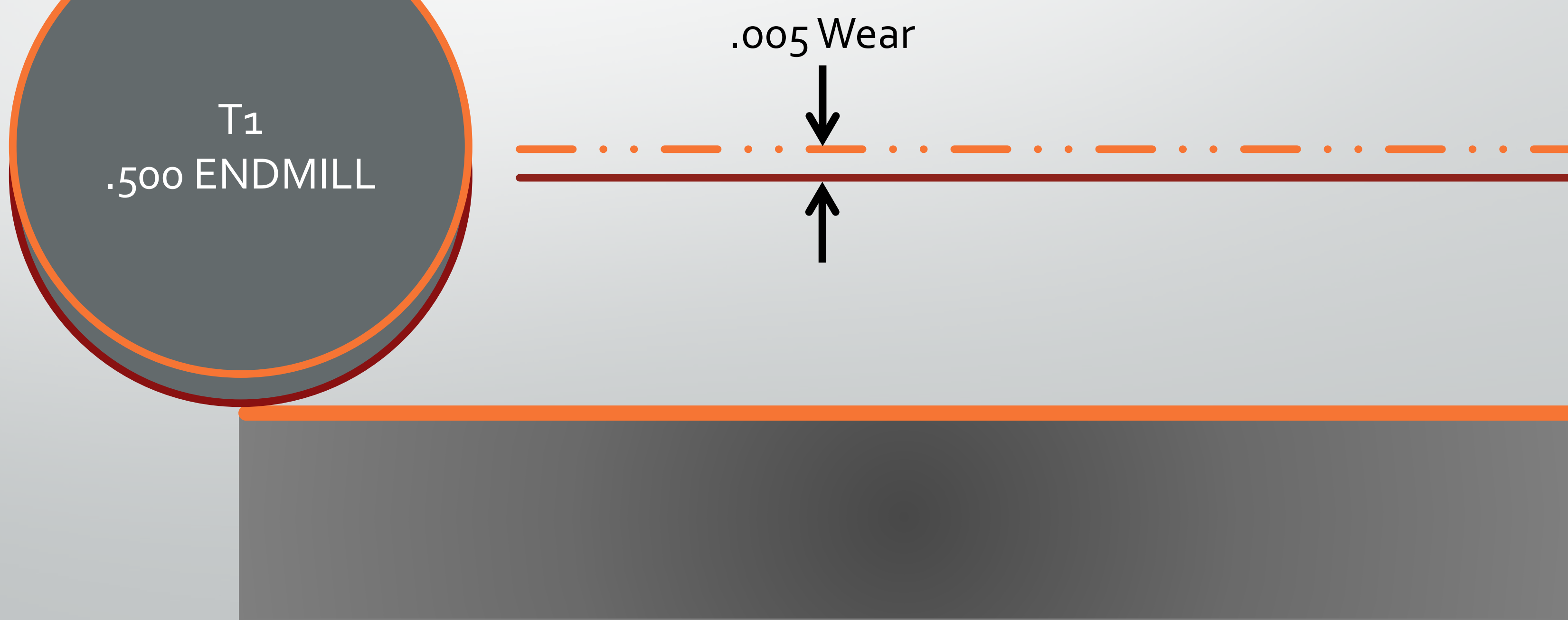
Diameter Cutting Compensation



Wear Compensation

- Wear compensation is used to adjust the placement of the tool in small increments to make up for variables such as tool wear and deflection.
- Wear uses these small adjustments to fine tune the tool and cut as precisely as possible.

Wear Compensation



Incremental Looping

- M97 combined with P and L values allow for the program to run through repeated code for a given number of times.
  - P- Program number to find in program
  - L- number of times to loop through the program
  - M99- ends the repeated code and returns to the M97 line
- This strategy is especially useful when a single toolpath needs to happen repeatedly. As the machine can be programed to run a toolpath, then move to another location before looping back to the beginning of that toolpath.

```
M97 P1000 L2 (L2 command
will run the N1000 line twice)
M30
N1000 (sub routine)
;
;
;
M99 (end sub routine)
```

Multiple Tool and Work Offsets

- There are times when using multiple tool and work offsets at the same time are beneficial. There are many reasons to use more than one work offset or tool offsets:
  - Multiple setups can be included in one program, often a first and second operation.
  - One tool can be used with multiple offsets. For example, this can be helpful when two different tolerances are needed while using the same tool.
  - For this to work, Setting 15 must be turned on in the control settings.

Canned Cycles

- Canned cycles are like toolpath templates. They use set values to complete toolpaths with minimal programming.
- There are multiple milling canned cycles for operations such as drilling and tapping.
- Canned cycles remain active until they are cancelled.

G73 High Speed Peck Drilling

- G73 is used to drill holes quickly while pecking. This helps to evacuate chips and improve coolant flow over the tool and in the hole.
- A G73 Cycle looks like this:
  - G73 X5. Y2. Z-.75 Q.25 R.1 F3.

```
%
O00001 (MILL FORMAT);
N1 (NOTES)
G00 G17 G20 G40 G49 G54 G64 G80 G90 G94 G98 G100;
G53 Z0.;
T1 M06;
X0. Y0. S1000 M03;
G43 H01 Z1. M08;
(BEGIN TOOL BODY);
G73 X5. Y2. Z-.75 Q.25 R.1 F3.
;
;
;
(END TOOL BODY);
G00 G80 Z1. M09;
G49 G53 Z0.;
M01
```

G73 High Speed Peck Drilling

- G73 X5. Y2. Z-.75 Q.25 R.1 F3.
  - X5. & Y2. Is the hole location
  - Z-.75 Is the hole depth
  - Q.25 Is the peck depth
  - R.1 Is the the retract plain
  - F3. Is the feedrate

G73 High Speed Peck Drilling

- G73 X5. Y2. Z-.75 R.1 Q.25 F3.
- Optional G73 codes:
  - I – First peck depth
  - J – Amount to reduce each peck
  - K – Minimum peck depth
  - L – Number of holes to drill if in incremental mode (G91)
  - P – Pause at hole bottom (seconds)

G81 Drilling Canned Cycle

- G81 is a drilling canned cycle. Once the Z travel starts, it does not stop until the depth is reached. After the depth is achieved, the tool then rapids out of the hole.
- A G81 cycle looks like this:
  - G81 X3. Y2. Z-1. R.1 F3.

```
%
O00001 (MILL FORMAT);
N1 (NOTES)
G00 G17 G20 G40 G49 G54 G64 G80 G90 G94 G98 G100;
G53 Z0.;
T1 M06;
X0. Y0. S1000 M03;
G43 H01 Z1. M08;
(BEGIN TOOL BODY);
 G81 X3. Y2. Z-1. R.1 F3.
;
;
;
(END TOOL BODY);
G00 G80 Z1. M09;
G49 G53 Z0.;
M01
```

G81 Drilling Canned Cycle

- G81 X3. Y2. Z-1. R.1 F3.
  - X3. & Y2. Is the hole location
  - Z-1. Is the hole depth
  - R.1 Is the retract plain
  - F3. Is the feedrate

G82 Drilling Canned Cycle w/ Dwell

- G82 is the same cycle as G81 but adds a dwell when the depth is achieved. This allows time for the tool to rotate and cut, relieving any possible cutting pressure.
  - This is often done with spot drills
- A G82 cycle looks like this:
  - G82 X3. Y2. Z-1. P1 R.1 F3.

```
%
O00001 (MILL FORMAT);
N1 (NOTES)
G00 G17 G20 G40 G49 G54 G64 G80 G90 G94 G98 G100;
G53 Z0.;
T1 M06;
X0. Y0. S1000 M03;
G43 H01 Z1. M08;
(BEGIN TOOL BODY);
 G82 X3. Y2. Z-1. P.1 R.1 F3.
;
;
;
(END TOOL BODY);
G00 G80 Z1. M09;
G49 G53 Z0.;
M01
```

G82 Drilling Canned Cycle W/ Dwell

- G82 X3. Y2. Z-1. P.1 R.1 F3.
  - X3. & Y2. Is the hole location
  - Z-1. Is the hole depth
  - P.1- dwell 1 second
  - R.1 Is the retract plain
  - F3. Is the feedrate

G83 Normal Peck Drilling

- G83 is used to drill while pecking. The drill retracts completely out of the hole at a set increment.
- A G83 cycle looks like this:
  - G83 X3. Y2. Z-1. R.1 Q.25 F3.

```
%
O00001 (MILL FORMAT);
N1 (NOTES)
G00 G17 G20 G40 G49 G54 G64 G80 G90 G94 G98 G100;
G53 Z0.;
T1 M06;
X0. Y0. S1000 M03;
G43 H01 Z1. M08;
(BEGIN TOOL BODY);
G83 X3. Y2. Z-1. R.1 Q.25 F3.
;
;
;
(END TOOL BODY);
G00 G80 Z1. M09;
G49 G53 Z0.;
M01
```

G83 Normal Peck Drilling

- G83 X3. Y2. Z-1. R.1 Q.25 F3.
  - X3. & Y2. Is the hole location
  - Z-1. Is the hole depth
  - R.1 Is the retract plain
  - Q.25 Is the pecking increments
  - F3. Is the feedrate

G83 Normal Peck Drilling

- G83 X3. Y2. Z-1. R.1 Q.25 F3.
- Optional G83 codes:
  - I – First peck depth
  - J – Amount to reduce each peck
  - K – Minimum peck depth
  - L – Number of holes to drill if in incremental mode (G91)
  - P – Pause at hole bottom (seconds)

G84 Tapping

- G84 is a tapping cycle. Tapping cycles orient the tool before starting a repeatable plunge into the part at the desired feed. The speed and feed need to correspond to the pitch of the thread for this toolpath to work. Otherwise, the tool or part will break and the thread will be ruined.
- A G84 cycle looks like this:
  - G84 X3. Y2. Z-.625 R.1 F10.

```
%
O00001 (MILL FORMAT);
N1 (NOTES)
G00 G17 G20 G40 G49 G54 G64 G80 G90 G94 G98 G100;
G53 Z0.;
T1 M06;
X0. Y0.  S250   M03;
G43 H01 Z1. M08;
(BEGIN TOOL BODY);
 G84 X3. Y2. Z-.625 R.1 F19.23
;
;
;
(END TOOL BODY);
G00 G80 Z1. M09;
G49 G53 Z0.;
M01
```

G84 Tapping

- G84 X3. Y2. Z-.625 R.1 F19.23
  - X3. & Y2. Is the hole location
  - Z-.625 Is the hole depth
  - R.1 Is the retract plain
  - F19.23 Is the feed rate
    - Feed rate is found with the following;
      - RPM/TPI= IPM
      - 250/13= 19.23

- Setting 130 on the control can be used to control the tap retract speed. The range is 0-4, 2 being 2 times the feed rate, 3 for 3 times the feed rate and 4 for 4 times the feed rate.
- Setting 133 on the control can be used during tapping. This ensures the spindle is oriented during tapping. This is done so threads can be tapped a second time or peck tapping can be used.

G84 Tapping

- G84 X3. Y2. Z-.625 R.1 F10.
- Optional G84 codes:
  - J – Retract multiple (How much faster exit is than entry)
  - L – Number of holes to drill if in incremental mode (G91)
  - S – Spindle speed

G187 Smoothing

- *G187* is an accuracy command that can set and control both the smoothness and max corner rounding value when cutting a part. The format for using *G187* is *G187 Pn Ennnn*.
- **P** - Controls the smoothness level, *P1*(rough), *P2*(medium), or *P3*(finish). Temporarily overrides Setting 191.
  E - Sets the max corner rounding value. Temporarily overrides Setting 85.
  Setting 191 sets the default smoothness to the user specified ROUGH, MEDIUM, or FINISH when *G187* is not active. The Medium setting is the factory default setting.

  - Think of this as a toolpath tolerance. How far the tool can deviate from the programmed line while cutting. In the example, the straight line is the programmed path, while the curved line is the actual machined path. Rough would be faster but allow for greater deviation, while finish would be slower but stay closer to the programmed line.

G12, G13 Circular Pocket Milling

- Circular Pocket Milling CW (G12) & Circular Pocket Milling CCW (G13) mill circular pockets.
- They can both be used for roughing or finishing operations. Radial cutter comp. is built in, so no G41 or G42 is needed.
- The only difference between G12 & G13 is the cutting direction. G12 is CW and G13 is CCW.

G12, G13

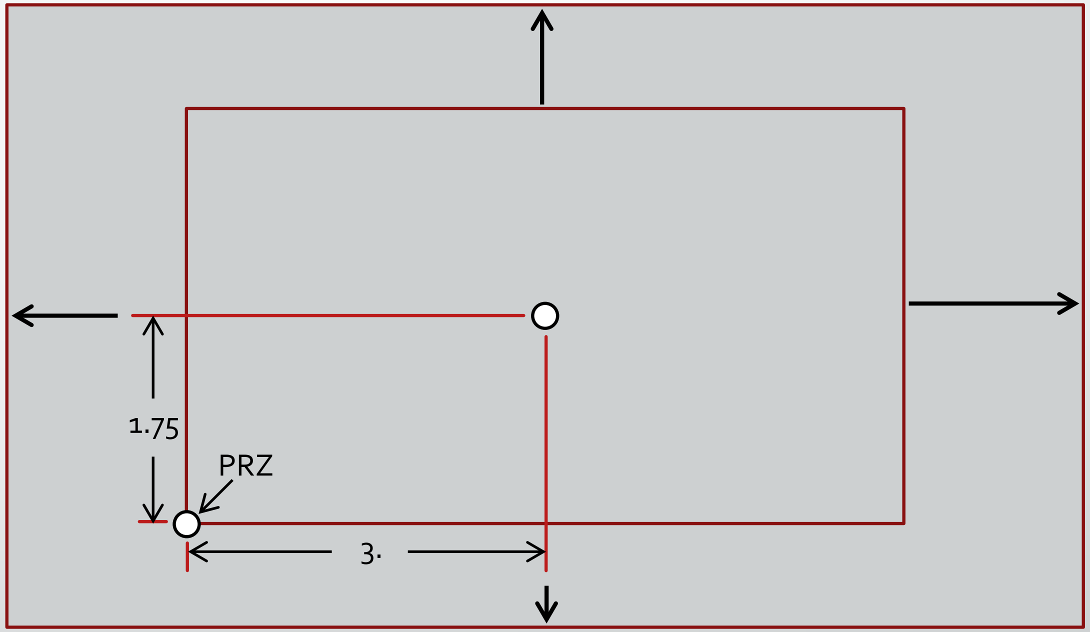
- G12 I0.3 K1.5 Q1. F10. Z-1.2 D01
  - D – Tool radius or diameter selection
  - F – Feedrate
  - I – Radius of first circle (or finish if no K). I value must be greater than Tool Radius, but less than K value.
  - K – Radius of finished circle (if specified)
  - L – Loop count for repeating deeper cuts
  - Q – Radius increment, or stepover (must be used with K)
  - Z – Depth of cut or increment

G51 Scaling

- G51 is used to scale toolpaths. It uses a set reference point to scale from with the desired scale factor.
- G51 example:
  - G51 X3. Y1.75 P1.5
    - X, Y, and Z – The scale reference point
    - P – Scale factor
- Features can be scaled by a factor of 0.001 to 999.999.
- G51 doesn't need a reference location provided, but if one isn't included, the last location will be used instead.

G51 Scaling

G51 X3. Y1.75 P1.5



G52 Set Work Coordinate Shift

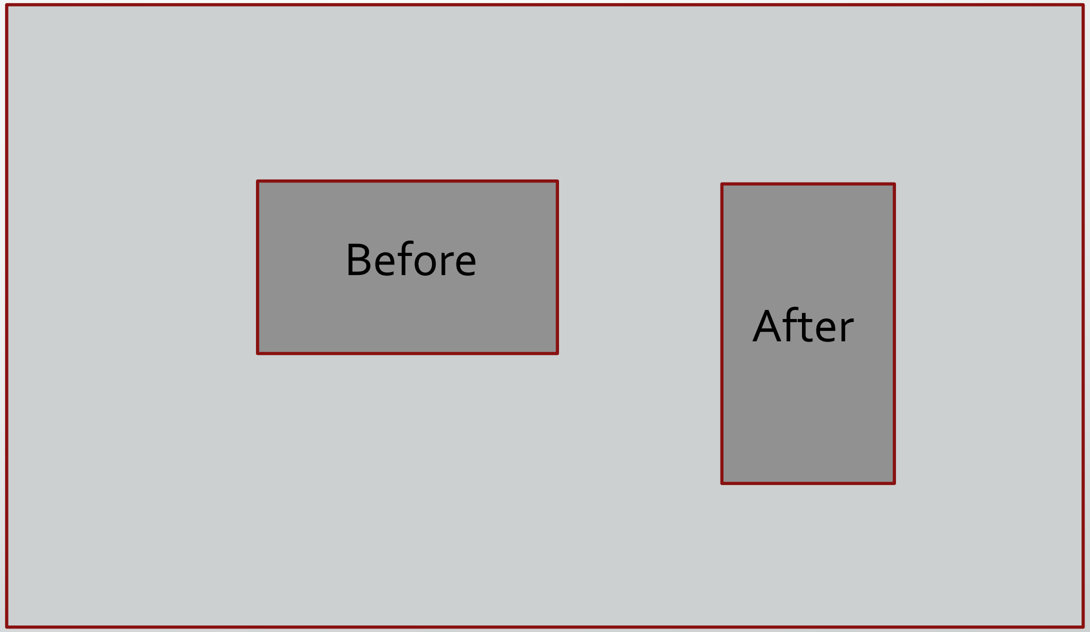
- G52 can be used as a global work shift.
  - The X,Y,Z values that are input in the control shift all work offsets by the specified amount. G52 does not change the work offset values, rather when the offset is activated it adds the values to the specified work offset.
  - G52 can be used on nested parts in a fixture or on identical features on a part.

G68 Rotational Shift

- G68 rotates the toolpath in relation to a point at a set angle.
- G68 example:
  - G68 X0. Y0. R-90.
    - X & Y – Point of rotation
    - R – Degree of rotation
- G68 rotates about the active plain, which is X & Y (G17) in this case. G18 or G19 can be activated as well to rotate the toolpath about X & Z or Y & Z.

G68 Rotational Shift

G68 Xo.Yo. R-90.



G150 General Purpose Pocket Milling

- General Purpose Pocket Milling cuts out a defined profile defined by a P value and a local sub-program.
- This toolpath leads into the pocket in Z, Roughs the profile, then runs a finish pass.

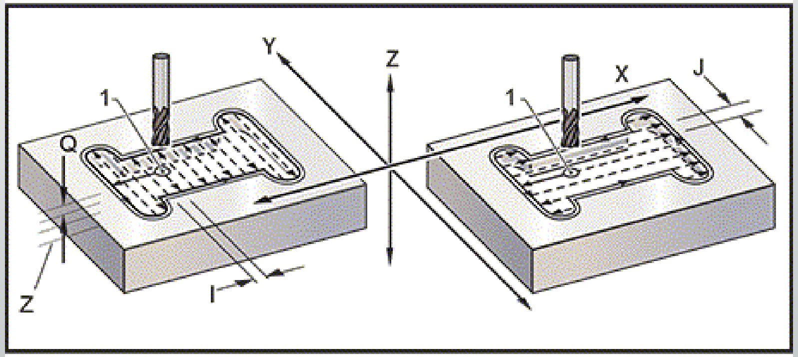


Photo curtesy of Haas Automations Inc.
https://www.haascnc.com/service/codes-settings.type=gcode.machine=mill.value=G150.html

G150 General Purpose Pocket Milling

- G150 example:
  - G150 X2.25 Y3.5 Z-.5 G41 J0.35 K.01 Q0.25 R.1 P1002 D01 F15.
    - X & Y – X and Y start position
    - Z – Final depth
    - G41 – Radial tool comp.
    - I – X axis cut increment (positive value)
    - J – Y axis cut increment (positive value)
    - K – Finishing pass amount (positive value)
    - Q – Incremental Z axis cut depth/pass (positive value)
    - R – Retract plain position
    - P – Sub-program number
    - D – Tool radius/diameter offset
    - F – Feed

M97, M98, M99 Sub-programs

- M97 and M98 both call up sub-programs, which are mini programs that can be called upon throughout a program.
  - M97 calls up a sub routine that is stored within the main program, after the M30.
  - M98 calls up a sub program that is stored as a separate program in the machine memory.
- Both options use two values:
  - P – Line No. (M97) and Program number (M98)
  - L – Repeat sub-program (1-99)

M97, M98, M99 Sub-programs

- M99 has three purposes:

1. It can return a program to its beginning and continue running it.
2. It is used at the end of a sub-program to return to the main program.
3. It can act as a GOTO command.

- Therefore, M99 is essential when using M97 & M98, otherwise the program would be stuck in the sub program.

M97, M98, M99 Sub-programs

- There are many possibilities once M97, M98, and M99 are put together. Here are some examples of how sub programs are called up to and from the main program.

```
%
O00001;
M97 P100 L2 (Calls N100 sub 2 times);
M30;
N100 (Subprogram);
G00 G91 X1.;
G83 Z-1.5 R.1 Q.375 F4.;
M99 (Returns to main program);
%
```

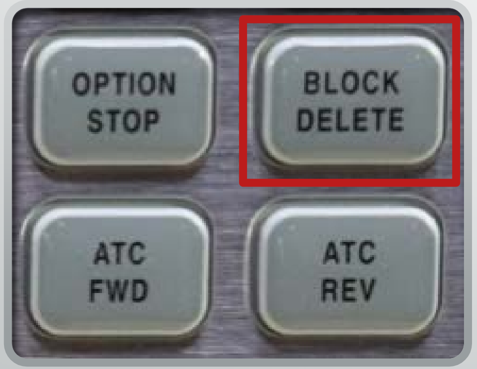
```
%
O00002;
M98 P200 L4 (Calls O200 sub 4 times) ;
M30 ;
%
```

↓Separate program in memory↓

```
%
O00200 (Subprogram);
M00 ;
M99 (Return to main program) ;
%
```

Block Delete

- If Block Delete is activated, all program lines with a "/" in the front will be skipped.



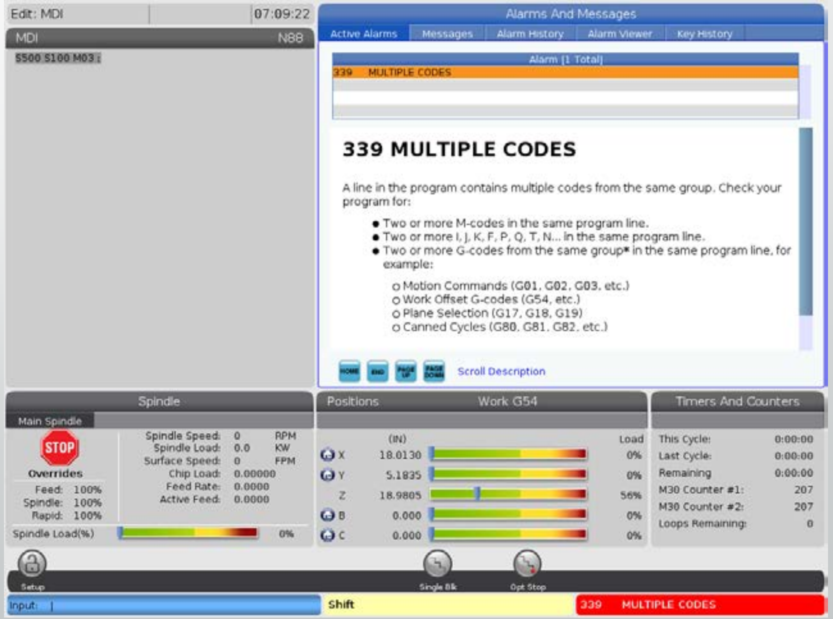

```
%
O00001 (MILL FORMAT);
N1 (NOTES)
G00 G17 G20 G40 G49 G54 G64 G80 G90 G94 G98 G100;
G53 Z0.;
T1 M06;
X0. Y0. S1000 M03;
G43 H01 Z1. M08;
(BEGIN TOOL BODY);
/G83 X3. Y2. Z-1. R.1 Q.25 F3.
;
;
;
(END TOOL BODY);
G00 G80 Z1. M09;
G49 G53 Z0.;
M01
```

Program Trouble Shooting

- New programs often have issues, especially if they are hand programmed.
- New programs should be verified before cutting the part.
- It is a good habit to use single block and slow the rapid to 5% when running a new program.

Program Trouble Shooting

- If an issue with the program does come, the machine will give an alarm.
- It is important not to hit reset right away.
  - Select the alarm button from the display keys and read the active alarm. This will give information about the issue.
  - Then go back to memory and read the line the alarm happened on. If it looks ok, read the following lines.
    - The machine has the ability to look ahead in the program. This means that the line the machine alarmed may not be the line of code with the issue.



Program Trouble Shooting

- The graphics simulation can also be used to get a 2D visual of the programmed toolpath without running the machine in memory.
- If a decimal point was missing in a line (G1 X2. Y2) the graphics would show the Y axis not moving to the proper position.