



## HAAS SERVICE AND OPERATOR MANUAL ARCHIVE

### VS Series Operators Manual 96-0103J RevJ English June 2004

- This content is for illustrative purposes.
- Historic machine Service Manuals are posted here to provide information for Haas machine owners.
- Publications are intended for use only with machines built at the time of original publication.
- As machine designs change the content of these publications can become obsolete.
- You should not do mechanical or electrical machine repairs or service procedures unless you are qualified and knowledgeable about the processes.
- Only authorized personnel with the proper training and certification should do many repair procedures.

**WARNING: Some mechanical and electrical service procedures can be extremely dangerous or life-threatening.  
Know your skill level and abilities.**

**All information herein is provided as a courtesy for Haas machine owners for reference and illustrative purposes only. Haas Automation cannot be held responsible for repairs you perform. Only those services and repairs that are provided by authorized Haas Factory Outlet distributors are guaranteed.**

**Only an authorized Haas Factory Outlet distributor should service or repair a Haas machine that is protected by the original factory warranty. Servicing by any other party automatically voids the factory warranty.**



## 1. MAINTENANCE

### 1.1 GENERAL REQUIREMENTS

Operating Temperature Range                    41°F to 104°F (5 to 40°C)  
Storage Temperature Range                    -4°F to 158°F (-20 to 70°C)  
Ambient Humidity: 20% – 95% relative humidity, non-condensing  
Altitude: 0-7000 ft.

### ELECTRICITY REQUIREMENTS

#### **IMPORTANT! REFER TO LOCAL CODE REQUIREMENTS BEFORE WIRING MACHINES.**

##### **ALL MACHINES REQUIRE:**

Three phase 50 or 60Hz power supply.  
Line voltage that does not fluctuate more than +/-10%

<b>40-30 HP System</b>	Voltage Requirements (195-260V)
Power Supply	125 AMP
Haas Circuit Breaker	100 AMP
If service run from ele.panel is less than 100' use:	2 GA. WIRE
If service run from ele. panel is more than 100' use:	0 GA. WIRE

#### **WARNING!**

A separate earth ground wire of the same conductor size as the input power is required to be connected to the chassis of the machine. This ground wire is required for operator safety and for proper operation. This ground must be supplied from the main plant ground at the service entrance, and should be routed in the same conduit as the input power to the machine. A local cold water pipe, or ground rod adjacent to the machine cannot be used for this purpose.

Input power to the machine must be grounded. For wye power, the neutral must be grounded. For delta power, a central leg ground or one leg ground should be used. The machine will not function properly on ungrounded power.

The rated horsepower of the machine may not be achieved if the imbalance of the incoming voltage is beyond an acceptable limit. The machine may function properly, yet may not deliver the advertised power. This is noticed more often when using phase converters. A phase converter should only be used if all other methods cannot be used.

The maximum voltage leg-to-leg or leg-to-ground should not exceed 260 volts.

The high voltage requirements shown reflect the Internal 400V option which is available only in Europe. Domestic and all other users must use the External 480V option.

The current requirements shown in the table reflect the circuit breaker size internal to the machine. This breaker has an extremely slow trip time. It may be necessary to size the external service breaker up by 20-25%, as indicated by "power supply", for proper operation.



VS-Series

### AIR REQUIREMENTS

The VMC requires a minimum of 100 PSI at 4 scfm at the input to the pressure regulator on the back of the machine. This should be supplied by at least a two horsepower compressor, with a minimum 20-gallon tank, that turns on when the pressure drops to 100 PSI.

Machine Type	Main Air Regulator	Input Airline Hose Size
VS-3	85 psi	1/2" I.D.

The recommended method of attaching the air hose is to the barb fitting at the back of the machine with a hose clamp. If a quick coupler is desired, use at least a 3/8".

---

**NOTE:** Excessive oil and water in the air supply will cause the machine to malfunction. The air filter/regulator has an automatic bowl dump that should be empty before starting the machine. This must be checked for proper operation monthly. Also, excessive contaminants in the air line may clog the dump valve and cause oil and/or water to pass into the machine.

---

**NOTE:** The nipple between the air filter/regulator and the Bijur oil lubricator (See illustration in "Air Connection" section) reservoir tank below the control box on the back of the machine is for the optional rotary table. DO NOT use this as a connection for an auxiliary air line. Auxiliary connections should be made on the left side of the air filter/regulator.

### WARNING!

**When the machine is operating and the pressure gauge (on the machine regulator) drops by more than 10 psi during tool changes or pallet changes, insufficient air is being supplied to the machine.**

## 1.2 MAINTENANCE SCHEDULE

The following is a list of required regular maintenance for the HAAS VS Series Vertical Machining Centers. Listed are the frequency of service, capacities, and type of fluids required. These required specifications must be followed in order to keep your machine in good working order and protect your warranty.

INTERVAL	MAINTENANCE PERFORMED
DAILY	<ul style="list-style-type: none"> <li>✓ Check coolant level each eight hour shift (especially during heavy TSC usage).</li> <li>✓ Check way lube lubrication tank level.</li> <li>✓ Clean chips from way covers and bottom pan.</li> <li>✓ Clean chips from tool changer.</li> <li>✓ Wipe spindle taper with a clean cloth rag and apply light oil.</li> </ul>
WEEKLY	<ul style="list-style-type: none"> <li>✓ Check Through the Spindle Coolant (TSC) filters. Clean or replace element if needed.</li> <li>✓ Check for proper operation of auto drain on filter regulator.</li> <li>✓ On machines with the TSC option, clean the chip basket on the coolant tank. Remove the tank cover and remove any sediment inside the tank. Be careful to disconnect the coolant pump from the controller and POWER OFF the control before working on the coolant tank. <b>Do this MONTHLY for machines without the TSC option.</b></li> <li>✓ Check air gauge/regulator for 85 psi.</li> <li>✓ For machines with the TSC option, place a dab of grease on the V-flange of tools. <b>Do this MONTHLY for machines without the TSC option.</b></li> <li>✓ Clean exterior surfaces with mild cleaner. <b>DO NOT</b> use solvents.</li> <li>✓ Check the hydraulic counterbalance pressure according to the machine's specifications.</li> </ul>
MONTHLY	<ul style="list-style-type: none"> <li>✓ Check oil level in gear box. Check oil level in sightglass. Add from side of gearbox if necessary.</li> <li>✓ Inspect way covers for proper operation and lubricate with light oil, if necessary.</li> </ul>
SIX MONTHS	<ul style="list-style-type: none"> <li>✓ Replace coolant and thoroughly clean the coolant tank.</li> <li>✓ Check all hoses and lubrication lines for cracking.</li> </ul>
ANNUALLY	<ul style="list-style-type: none"> <li>✓ Replace the gearbox oil. Drain the oil from the bottom of the gearbox. Remove inspection cover beneath spindle head. Add oil from the side of the transmission.</li> <li>✓ Check oil filter and clean out residue at bottom of filter.</li> <li>✓ <b>Replace air filter on control box every (2) years.</b> 1.1 Through the Spindle Coolant (TSC) Maintenance</li> <li>✓ Check SMTC oil level in sight glass, (see Side Mount Tool Changer Oil Level in this section).</li> </ul>

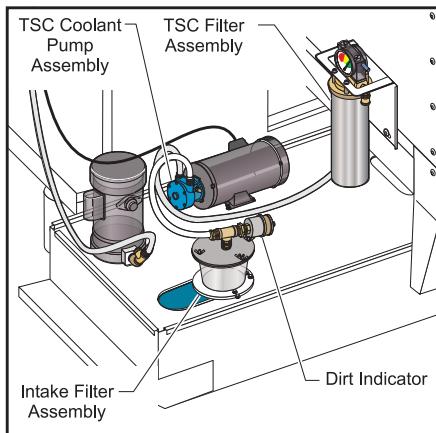


VS-Series

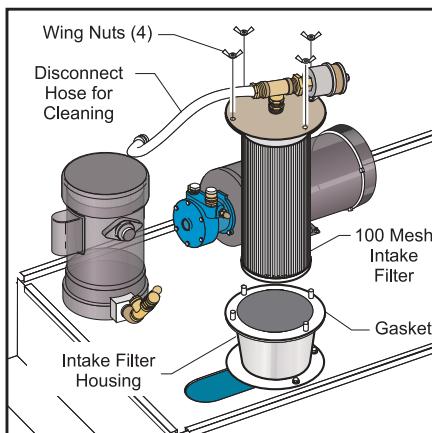
### 1.3 TSC MAINTENANCE

The TSC pump is a precision gear pump and will wear out faster and lose pressure if abrasive particles are present in the coolant.

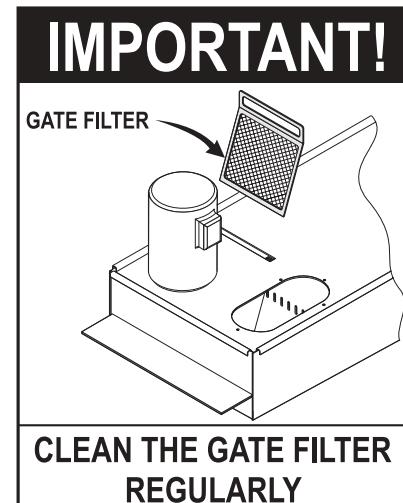
- Check the dirt indicator on the 100-micron mesh filter with the TSC system running and no tool in the spindle. Change the element when the indicator reaches the red zone.
- Clean the pump intake filter when indicator is in the red zone. Reset indicator with button. All intake filters can be cleaned with a wire brush.
- After changing or cleaning filter elements, run TSC system with no tool in spindle for at least one minute to prime system.
- Coolant will be used more quickly when the TSC system is in use. Make sure to keep the coolant level up and to check the level more frequently (check after every eight hour shift). **Premature wear of the pump can result from running with a low coolant level in the tank.** The spindle will shut off automatically if the coolant level gets too low.



TSC Coolant Pump Assembly.



Cleaning the Intake Filter.



**CLEAN THE GATE FILTER  
REGULARLY**

Gate Filter

#### Warnings

Use of coolants with extremely low lubricity can damage the TSC coolant tip and pump.

Shortened pump life, reduction of pressure and increased maintenance are normal and to be expected in abrasive environments and are not covered by warranty. A special filter, in addition to the standard filter should be used; contact HAAS for recommendations.

Machining of ceramics and the like voids all warranty claims for wear and is done entirely at customer's risk. Increased maintenance schedules are absolutely required with abrasive swarf. The coolant must be changed more often and the tank thoroughly cleaned of sediment on the bottom. An auxiliary coolant filter is recommended.

#### To Clean the Gate Filter:

1. Turn off the coolant pump.
2. Remove the filter.
3. Clean and reinstall filter.

#### 1.4 LUBRICATION CHART

SYSTEM	WAY LUBE AND PNEUMATICS	TRANSMISSION	COOLANT TANK
LOCATION	Under the control panel at the rear of the machine	Above the spindle head	Side of machine
DESCRIPTION	Piston pump with 30 minute cycle time. Pump is only on when spindle is turning or when axis is moving.		
LUBRICATES	Linear guides, ball nuts and spindle	Transmission only	
QUANTITY	2-2.5 Qts. depending on pump style	36 oz	80 Gallons
LUBRICANT	Mobil Vactra #2	Mobile DTE 25	Water-soluble synthetic oil based or synthetic based coolant/lubricant. * <b>No Flammable Liquids.</b>

\* Mineral cutting oils will damage rubber components throughout the machine.

**Do not use pure water as a coolant; machine components will rust.**

**WARNING!**

The TSC pump is a precision gear pump and will wear out faster and lose pressure if abrasive particles are present in the coolant.

Use of coolants with extremely low lubricity can damage the TSC coolant tip and pump.

When machining castings, sand from the casting process and the abrasive properties of cast aluminum and cast iron will shorten pump life unless a special filter is used in addition to the 100 mesh suction filter. Contact Haas Automation for recommendations.

Machining of ceramics and the like voids all warranty claims for wear and is done entirely at the customer's risk. Increased maintenance schedules are absolutely required with abrasive swarf. The coolant must be changed more often, and the tank thoroughly cleaned of sediment on the bottom. A larger coolant tank is recommended.

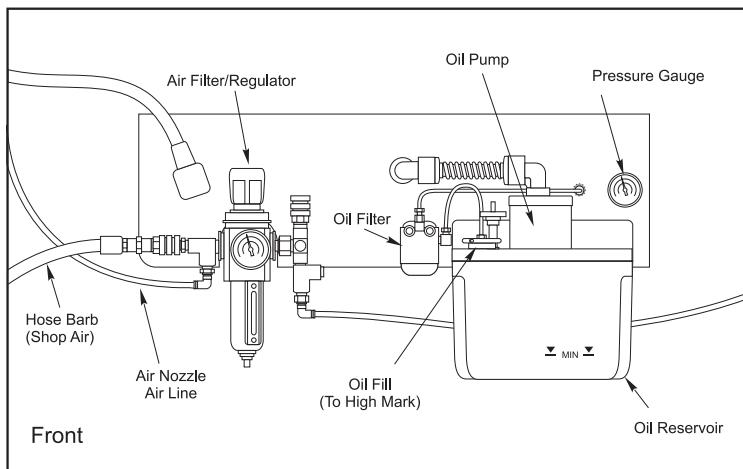
Shortened pump life, reduction of pressure and increased maintenance are normal and to be expected in abrasive environments and is not covered by warranty.



VS-Series

## 1.5 LUBRICATION SYSTEM

All machine lubrication is supplied by the external lubrication system. The reservoir is located on the lower rear of the machine (see Figure below). Current lube level is visible in the reservoir. If additional lube needs to be added, remove the cap from the fill port and add lube to proper level.



*External Lubrication System*

### WARNING!

**DO NOT ADD LUBE ABOVE THE "HIGH" LINE MARKED ON THE RESERVOIR.  
DO NOT ALLOW THE LUBE LEVEL TO GO BELOW THE "LOW" LINE MARKED  
ON THE RESERVOIR AS MACHINE DAMAGE COULD RESULT.**

To lubricate the system, pull up on the primer pull-tab located next to the fill port. The primer will automatically send 3cc of lube through the system.

## SPINDLE AIR PRESSURE

Verify Spindle air pressure using the gauge located behind the Air Regulator panel. VF machines should show 17 psi. Adjust if necessary.

### 15K Spindle

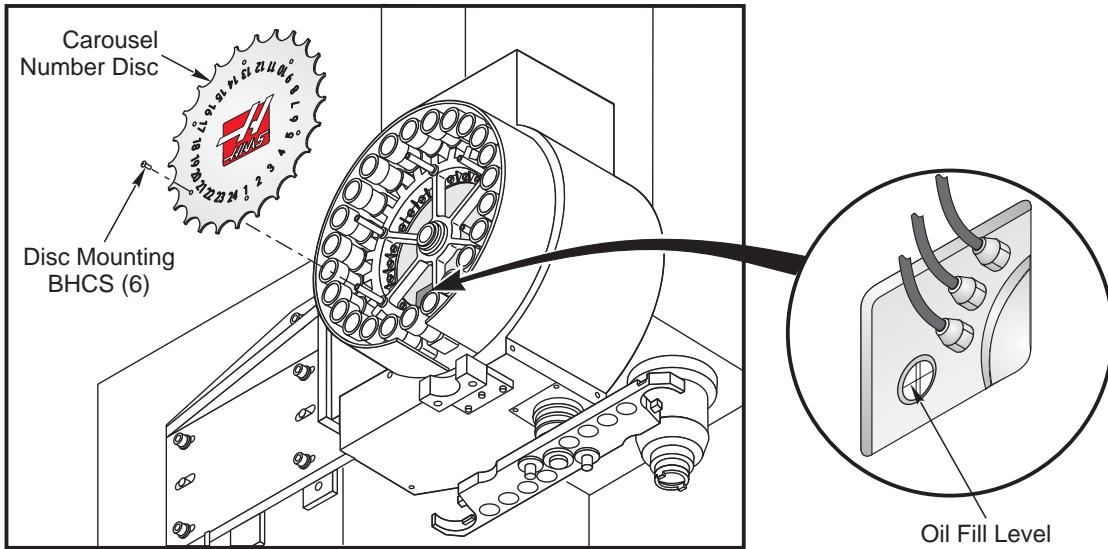
The air pressure for the 15K Spindle is 20 psi. The 15K Spindle requires higher pressure to slightly reduce the amount of oil and speed the delivery of the oil to the bearings.

## 1.6 CHIP AUGER

### MAINTENANCE

During normal operation, most chips are discharged from the machine at the discharge tube. However, very small chips may flow through the drain and collect in the coolant tank strainer. To prevent drain blockage, clean this trap regularly. Should the drain become clogged and cause coolant to collect in the machine's pan, stop the machine, loosen the chips blocking the drain, and allow the coolant to drain. Empty the coolant tank strainer, then resume operation.

### 1.7 SIDE MOUNT TOOL CHANGER OIL LEVEL CHECK



The SMTC is factory filled with the appropriate level of oil and does not need to be changed under normal conditions. As a precaution, check the oil level annually. Oil will not need to be added as long as the level remains viewable in the sight glass as shown above. Should the level drop below the sight glass, call the HAAS service department.



## 1.8 PERIODIC MAINTENANCE

A periodic maintenance page has been added to the control, it is found on the Current Commands screens titled SCHEDULED MAINTENANCE and accessed by pressing PAGE UP or PAGE DOWN which allows the operator to activate and deactivate a series of checks (see list below).

An item on the list can be selected by pressing the up and down arrow keys. The selected item is then activated or deactivated by pressing ORIGIN. If an item is active, the remaining hours will be displayed to the right. If an item is deactivated, “—” will be displayed instead. Items are tracked either by the time accumulated while power is on (ON-TIME) or by cycle-start time (CS-TIME). When power is applied, and every hour thereafter, the remaining time for each item is decremented. When it reaches zero (or has gone negative) the message MAINTENANCE DUE is displayed at the bottom of the screen. A negative number of hours indicates the hours past expiration. The maintenance item can have its time adjusted by using the left and right arrows. One hour is added or subtracted for each keypress, upto a maximum of 10,000 hours, and a minimum of 1 hour. Pressing the Origin key will reinstate the default time.

This message is not an alarm and does not interfere with machine operation in any way. The intent is to warn the operator that one of the items on the list requires attention. After the necessary maintenance has been performed, the operator can select that item on the SCHEDULED MAINTENANCE screen, press ORIGIN to deactivate it, then press ORIGIN again to reactivate it, and the countdown begins again with a default number of hours remaining (this value is determined by the software and cannot be altered by the operator.) Items available for checking are:

COOLANT - needs replacement	100 ON-TIME
AIR FILTER in control enclosure - replace	250 ON-TIME
OIL FILTER - replace	250 ON-TIME
GEARBOX OIL - replace	1800 ON-TIME
COOLANT TANK - check level, leakage, oil in coolant	10 ON-TIME
WAY LUBE SYSTEM - check level	50 CS-TIME
GEARBOX OIL - check level	250 ON-TIME
SEALS/WIPERS missing, torn, leaking - check	50 CS-TIME
AIR SUPPLY FILTER - check for water	10 ON-TIME
HYDRAULIC OIL - check level	250 ON-TIME

## 1.9 WINDOWS / GUARDING

Polycarbonate windows and guarding can be weakened by exposure to cutting liquids and chemicals that contain amines. It is possible to lose up to 10% of the remaining strength annually. If degradation is suspected, window replacement should occur at no more than a two year interval.

**Windows and guarding should be replaced if damaged or severely scratched - Replace damaged windows immediately**

## 2. OPERATION

The Information contained in this manual is constantly being updated. The latest updates and other helpful information can be downloaded in .pdf format from the Haas website. Go to [www.HaasCNC.com](http://www.HaasCNC.com) and click on "Manual Updates" under the "Customer Services" drop-down menu on the navigation bar.

Additional technical information is available in the Haas Reference Manual. The Reference Manual contains definitions of parameters and alarms, machine installation instructions, and explanations of the machine's mechanical and electrical subassemblies.

The Haas Reference Manual is available online as a free download in .pdf format (go to [www.HaasCNC.com](http://www.HaasCNC.com) and click on "Manual Updates" under the "customer services" drop-down menu in the navigation bar), or may be purchased through the Haas Service Department.

### 2.1 BASIC INTRODUCTION

This section provides the basic programming and operation principles necessary to begin operating the machine. The remainder of this manual is divided into more detailed **Programming** and **Operation** sections.

In an "NC" (Numerically Controlled) 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 the machine tool is monitored by a computer.

The same principles used in operating a manual machine are used in programming an NC or CNC machine. The main difference is that instead of cranking handles to position 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 VS-Series Vertical Machining Center requires that a part program be designed, written, and entered into the memory of the control. The most common way of writing part programs is off-line, that is, away from the CNC in a facility that can save the program and send it to the CNC control. The most common way of sending a part program to the CNC is via an RS-232 interface. The HAAS VS-Series Vertical Machining Center has an RS-232 interface that is compatible with most existing computers and CNC's.

In order to operate and program a CNC controlled machine, a basic understanding of machining practices and a working knowledge of math is 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 can be used as both an operator's manual and as a programmer's manual. It 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. Much more training and information is necessary before attempting to program on the machine.

The programming section of this manual is meant as a supplementary teaching aid to users of the HAAS Vertical Machining Center. The information in this section may apply in whole or in part to the operation of other CNC machines. Its use is intended only as an aid in the operation of the HAAS Vertical Machining Center.



## 2.2 THE COORDINATE SYSTEM

The first diagram we are concerned with is called a NUMBER LINE. This number line has a reference point zero that is called ABSOLUTE ZERO and may be placed at any point along the line.

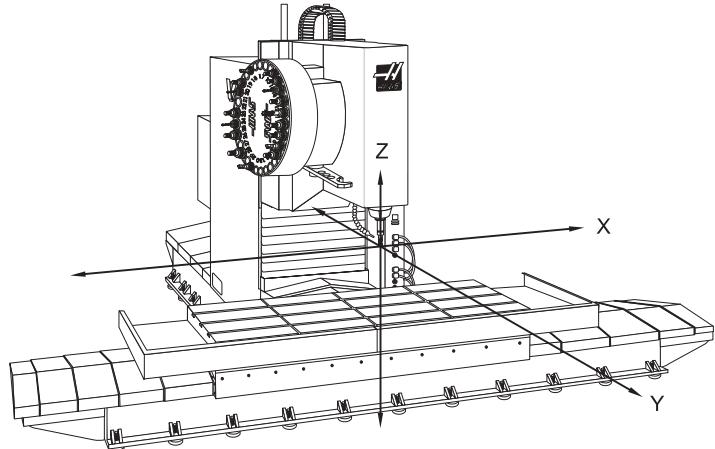


Horizontal number line.

The number line also has numbered increments on either side of absolute zero. Moving away from zero to the right are positive increments. Moving away from zero to the left are negative increments. The “+”, or positive increments, are understood, therefore no sign is needed.

- We use positive and negative along with the increment's value to indicate its relationship to zero on the line. In the case of the previous line, if we choose to move to the third increment on the minus (-) side of zero, we would call for -3. If we choose the second increment in the plus range, we would call for 2. Our concern is with distance and direction from zero.
- Remember that zero may be placed at any point along the line, and that once placed, one side of zero has negative increments and the other side has positive increments.
- The next illustration shows the three directions of travel on a vertical machining center. To carry the number line idea a little further, imagine such a line placed along each axis of the machine.

The first number line is easy to conceive as belonging to the left-to-right, or "X", axis of the machine. If we place a similar number line along the front-to-back, or "Y", axis, the increments toward the operator are the negative increments, and the increments on the other side of zero away from the operator are the positive increments.



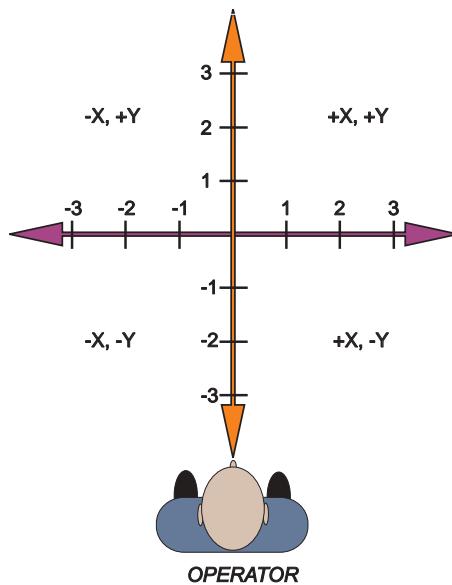
VS-3 showing X, Y, and Z axis lines.

The final axis of travel on our machine is the up-and-down, or "Z", axis. When we place a number line on the Z travel, the positive increments are up — above zero — and the negative values are down — below zero.

Remember, when we are moving the machine, we are concerned with positioning the spindle. Although the machine table is the moving part, we have to keep in mind our coordinates are based off our theoretical spindle movement.

Keep in mind that the zero position may be placed at any point along each of the three number lines, and in fact will probably be different for each setup of the machine. It is noteworthy to mention here that the Z-axis is usually set with the machine zero position in the full upward position, or the tool change position. This will place all the Z moves in a negative range of travel. However, the work zero in the Z-axis is usually set at the top of the part surface, and this will be entered in the tool length offset as a negative value.

The diagram shows a top view of the grid as it would appear on the machine tool. This view shows the **X** and **Y** axes as the operator faces the machine tool. Note that at the intersection of the two lines, a common zero point is established. The four areas to the sides and above and below the lines are called “**QUADRANTS**” and make up the basis for what is known as rectangular coordinate programming.



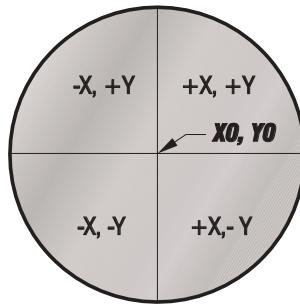
*View of X, Y grid from above.*

Whenever we set a zero somewhere on the X-axis and somewhere on the Y-axis, we have automatically caused an intersection of the two lines. This intersection where the two zeros come together will automatically have the four quadrants to its sides, above, and below it. How much of a quadrant we will be able to access is determined by where we placed the zero within the travel of the machine axis.



## 2.3 MACHINE HOME

The principal may be seen when doing a manual reference return of all machine axes. When a zero return (ZERO RET) is performed at machine start up, all three axes are brought to the extreme positive direction until the limit switch is reached. When this condition is satisfied, the only way to move any of the three axes is in the negative direction. This is because a new zero was set for each of the three axes automatically when the machine was brought Home. This is placed at the edge of each axes travel. In effect, now the positive quadrants cannot be reached, and all the **X** and **Y** moves will be found to be in the **X-**, **Y-** quadrant. It is only by setting a new part zero somewhere within the travel of each axis that other quadrants are able to be reached.



*All four quadrants will have to be accessed to machine this part.*

Sometimes it is useful in the machining of a part to utilize more than one of the **X**, **Y** quadrants. A good example of this is a round part that has its datum lines running through the center. The setup of such a part may look like the previous figure.

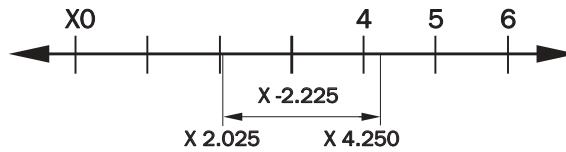
These are just some examples of how to make use of the four quadrants of the **X** and **Y** axes on the machine. As more experience is gained in the machine tool programming and setup techniques, each programmer and setup person develops their own methods and style. Some methods will be faster than others, but each individual will have to determine the needs of each job in question, and reflect back on notes and the previous jobs completed.

## 2.4 ABSOLUTE AND INCREMENTAL POSITIONING

Up to this point, we have dealt with a system of positioning the tool that is known as absolute programming. In absolute, all coordinate points are given with regard to their relationship to the origin, a fixed zero point, or considered as part zero. This is the most common type of positioning.

Another type of positioning is called incremental positioning. Incremental positioning concerns itself with distance and direction. A new coordinate is entered in terms of its relationship to the previous position, and not from a fixed zero or origin. In other words, after a block of information has been executed, the position that the tool is now at is the new zero point for the next move to be made.

An example of the use of the incremental system is below. Note that to move from **X 4.25** to **X 2.025** on the scale, an incremental move of **X -2.225** was made, even though the move still places the tool on the plus side of the scale. Therefore the move was determined from the last point, with no regard for the zero position. The **+** and **-** signs are used in terms of direction, and not in regard to the position of zero.



*An example of an incremental move.*

Keep in mind that when positioning in **absolute**, we are concerned with distance and direction from a fixed zero reference point, and when positioning in **incremental** we are concerned with distance and direction from the last position.

## 2.5 PROGRAMMING WITH CODES

A program is written as a set of instructions given in the order they are to be performed. The instructions, if given in English, might look like this:

Line #1 =	Select cutting tool.
Line #2 =	Turn the spindle on, select the rpm and the direction.
Line #3 =	Turn the coolant on.
Line #4 =	Rapid to the starting position of the part.
Line #5 =	Choose the proper feed rate and make the cut(s).
Line #6 =	Turn off the spindle and the coolant.
Line #7 =	Return tool to holding position and select next tool.

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

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 has an alphabetical address. The rule here is that except for G codes and macro calls, Codes with the same alphabetical address cannot be used more than once on the same line.
- 2) **G** codes come in groups. Each G code group has a specific group number. G codes from the same group cannot be used more than once on the same line.
- 3) There are modal **G** codes groups which, once established, remain effective until replaced with another code from the same group.
- 4) There are non-modal **G** codes groups which, once called, are effective only in the calling block, and are immediately forgotten by the control.



## 2.6 MACHINE DEFAULTS

A **default** is an automatic function of the machine tool control. When powering up the machine, the control looks for the home position of all axes, then will read the default values or the preset G codes. If you have ever wondered why the machine went to the part zero that was entered in the G54 when it was never specified in the actual program, it is because the machine automatically reads G54 upon start-up. That is a **default**.

The defaults for the Haas mill are indicated by an asterisk ( \* ) in the "Preparatory Functions (G Codes)" section of this manual.

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

G00	Rapid traverse
G17	X,Y plane selection
G20	Select inches
G40	Cutter Compensation cancel
G49	Tool length compensation cancel
G54	Work coordinate zero #1 (1 of 26 available)
G64	Exact stop cancel
G80	Canned cycle cancel
G90	Command absolute programming
G94	Feed per minute mode
G98	Canned cycle 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.

## 2.7 PROGRAM FORMAT

Program format, or style, is an important part of CNC machining. Each individual will format their programs differently and, in most cases, a programmer could not identify a program written by themselves. The point is that a programmer needs to be consistent and efficient, writing code in the way it is listed and in the order it appears in the program. For example:

**X, Y, Z** is 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.

The first line or block in a program using active G codes should be a tool number and tool change command. This would be a good safety measure.

The second 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).

The third line or block will contain a “Read tool length offset” command (G43), a tool length offset number (H01), a Z-axis positioning move (Z.1), and an optional coolant ON command (M08).

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

```
T1 M06;
G00 G90 G54 X0 Y0 S2500 M03;
G43 H01 Z.1 M08;
```

All the necessary codes for each operation are listed above. This format is a good practice and will separate your style from other programmers.

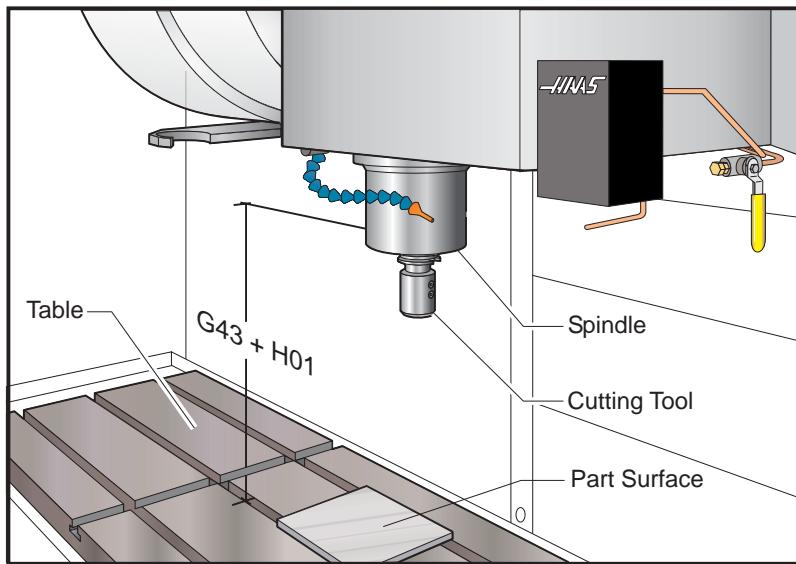


Fig. 3-7 Tool length offset and tool length compensation.

Tool number should always remain numerically matched with the tool length offset number. Setting 15 (the H & T agreement) will ensure the tool number and tool length offset will match. (Ex. T1 in line #1 should have H01 in line #3, and T2 should have H02 in line #3.)



VS-Series

## 2.8 CANNED CYCLES

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

- 1) Positioning of **X** and **Y** axes (optional A, rotary axis).
- 2) Rapid traverse to the reference plane.
- 3) Drilling, boring, or tapping action.
- 4) Operation at the bottom of the hole.
- 5) Retraction to the reference plane.
- 6) Rapid traverse to the initial starting point.

A canned cycle is presently limited to operations in the Z-axis; that is, only the G17 plane is allowed. This means the canned cycle will be executed in the Z-axis whenever a new position is selected in the **X** or **Y** axis. The operation of a canned cycle will vary according to whether incremental (G91) or absolute (G90) is active. Incremental motion in a canned cycle is often useful as a loop count (**L**) and can be used to repeat the operation with an incremental **X** and/or **Y** move. G98 and G99 are modal commands which change the way the canned cycles operate. When G98 (the system default) is active, the Z-axis will be returned to the starting position at the completion of the canned cycle. When G99 is active, the Z-axis will be returned to the reference plane when the canned cycle is completed.

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

For more detailed information on canned cycles, refer to the "G Codes" section of this manual.

## 2.9 SETUP PROCEDURES

The following sequence of operations is strongly recommended for setting up this machine:

- 1) Load a program into memory. This is either manually entered or downloaded from a CAM package via the RS-232 interface.
- 2) Determine the tools needed and get them ready.

**CAUTION!** If machine is equipped with the optional 50 taper spindle, follow these guidelines:

- 25 lb. maximum per tool, and 300 lb. maximum total tool weight.
- Extremely heavy tool weights should be distributed evenly.

**NOTE:** If machine is equipped with the optional **extended Z-clearance table**, follow these guidelines:

- Avoid extreme loads on center of table or far end of table. Weight of part should be distributed evenly over the table or one pad.
- Flatness of part should be within .002"

- 3) Use a vise or fixture to hold the part and mount it on the mill.

- 4) Locate the **X** and **Y** zero points of your program on the part. Usually these points will coincide with the print reference point where dimensioning begins and needs to be clearly indicated by the programmer. Use an edge finder or indicator to locate this point with the handle function. After locating the programmed zero point, push the display offset key and page down until the work zero offset page appears. Cursor down to G54 **X**, the work zero offset. Push the PART ZERO SET button and the **X**-axis machine value will be stored at this location. Cursor to the G54 **Y** location and repeat the above. You have now told the machine where part zero is located. Usually **Z** and **A** values will not have to be set and should be zero.
- 5) Remove any tools from changer and MDI a T1 M6 command to install tool #1 into spindle (it should be empty). Put your tool #1 into the spindle using the TOOL RELEASE button. Push the OFSET key and page down to get to the tool offset page and cursor to tool #1. Do not install any tools directly into the carousel. Use MDI or ATC FWD/REV to retrieve tools.
- 6) Push the Z- JOG key until you are close to the top of your part. (The top of your part should be Z0). Use the Handle to accurately position the tool edge to Z0. Push the TOOL OFSET MESUR key and the **Z** machine value will be stored in tool offset #1. Note that this automatic offset measurement works with G43 only and the work **Z** offset must be zero.
- 7) Push the NEXT TOOL key and the Z-axis will retract to tool change position and tool #2 (empty) will be installed in the spindle. Put your tool #2 into the spindle and jog to **Z** zero as you did for tool #1. The cursor will automatically be on offset #2. Push TOOL OFSET MESUR.
- 8) Repeat this procedure until all tools are measured and installed.
- 9) MDI a T1 M6 to return to tool #1.
- 10) You are now ready to run your program.
- 11) Please note that in order to load and measure all of your tools, you do not need to use any keys other than JOG, TOOL OFSET MESUR, and NEXT TOOL. Also note that this automatic offset measurement works with G43 only.



VS-Series

## 2.10 OPERATORS CONTROL PANEL



Control panel keypad with operating and display keys highlighted.

In operation, it is important to be aware of the operating mode selected for the CNC. There are six operating modes and one simulation mode in this control. The operating mode is selected with the six buttons labeled:

EDIT	To edit a program already in memory
MEM	To run a program stored in memory
MDI / DNC	To directly run manually entered program or to select DNC mode
HANDLE JOG	To use jog keys or jog handle
ZERO RET	To establish machine zero
LIST PROG	To list, send, or receive programs

The Graphics simulation mode is entered with the DISPLAY select buttons.

In MEM or MDI mode, a program can be started with the CYCLE START button. While a program is running, you cannot change to another mode; you must wait until it finishes or press RESET to stop the program.

When already in MDI, a second push of the MDI button will select DNC if the DNC mode is enabled by settings and parameters in your machine.

In any of the above modes, you can select any of the following displays using the eight DISPLAY buttons:

PRGRM / CNVRS	To show the program selected
POSIT	To show the axes positions
OFSET	To show or enter working Offsets
CURNT COMDS	To show Current Commands and times
ALARM / MESGS	To show Alarms and user messages
PARAM / DGNOS	To show Parameters or Diagnostic data
SETNG / GRAPH	To show or enter Settings or to select Graphics simulation mode
HELP / CALC	To show the Help data and calculator

In addition to the above displays, when a program is already running, you may press LIST PROG to select a list of the programs in memory. This is useful to determine what programs can be edited in BACKGROUND EDIT. BACKGROUND EDIT is selected from the PROGRAM DISPLAY.

All operation of the CNC is controlled from the operator's panel. The control panel is composed of the display, the keypad, On/Off switch, Load meter, Handle, EMERGENCY STOP, CYCLE START, and FEED HOLD buttons.

The **keypad** is a flat membrane type that requires approximately eight (8) ounces of pressure. The **SHIFT** button replaces the function of the numeric buttons with the white characters in the upper left corner. The **SHIFT** button must be pressed once before each shifted character. Pressing the **SHIFT** button twice will turn off shift.

The **jog handle** is used to jog one of the axes. Each step of the crank can be set to 0.0001, 0.001, 0.01 or 0.1 inch (0.001, 0.01, 0.1, or 1.0 degree per step for a rotary axis). When using metric units, the smallest handle step is 0.001 mm and the largest is 1.0 mm. The handle has 100 steps per rotation. It can also be used to move the screen cursor while in EDIT mode, or to change feed/spindle overrides by +/-1%.

The EMERGENCY STOP button will instantly stop all motion of the machine including the servo motors, the spindle, the tool changer, and the coolant pump. It will also stop any auxiliary axes.

CYCLE START will start a program running in MEM or MDI mode, continue motion after a FEED HOLD, or continue after a SINGLE BLOCK stop. The CYCLE START button on the optional remote jog handle performs exactly the same functions.

FEED HOLD will stop all axis motion until the CYCLE START is pressed. The FEED HOLD button on the optional remote jog handle will perform exactly the same functions.

### WARNING!

**FEED HOLD will not stop the spindle, the tool changer, or the coolant pump. It will not stop motion of any auxiliary axes.**



The optional **Memory Lock Key Switch** will prevent the operator from editing programs and from altering settings when turned to the locked position.

The following describes the hierarchy of locks:

- Key switch locks Settings and all programs
- Setting 7 locks parameters - parameters 57, 209 and 278 lock other features
- Setting 8 locks all programs
- Setting 23 locks 9xxx programs
- Setting 119 locks offsets
- Setting 120 locks macro variables

The SINGLE BLOCK button on the keypad will turn on and off the SINGLE BLOCK condition. When in SINGLE BLOCK, the control will operate one block and stop. Every press of the START button will then operate one more block.

The RESET button on the keypad will always stop motion of the servos, the spindle, the coolant pump, and tool changer. It will also stop the operation of a running program. This is not, however, a recommended method to stop the machine as it may be difficult to continue from that point. SINGLE BLOCK and FEED HOLD provide for continuation of the program. RESET will not stop motion of any auxiliary axes but they will stop at the end of any motion in progress.

## Function Buttons

The F1, F2, F3, and F4 buttons perform different functions depending on what display and mode are selected. The following is a quick summary of the **Fn** buttons:

- F1 In the Advanced Editor, F1 will open the drop-down menus.  
In LIST PROG mode, F1 will duplicate a program already stored and give it a new name from the command line. In OFFSET display, F1 will set the entered value into the offsets.
- F2 In Advanced Editor mode and Program Display, this will start and end a block definition.  
In OFFSET display, F2 will set the negative of the entered value into the offsets.
- F3 In EDIT and MDI modes, the F3 key will copy the highlighted circular help line (result from calculator) into the data entry line at the bottom of the screen. This is useful when you want to use the solution developed for a circular motion. Push INSERT to add that circular motion command line to your program. In the calculator HELP function, this button copies the value in the calculator window to the highlighted data entry for Trig or Circular Help.
- F4 In MEM mode and PROGRAM DISPLAY, this will select either BACKGROUND EDIT or PROGRAM REVIEW. BACKGROUND EDIT is selected by entering **Onnnnn** with the program number to edit. Program review is selected with just F4. Program review shows the running program on the left half of the screen and allows the operator to review the program on the right half of the screen.  
  
When in EDIT mode with no program running, entering **Onnnnn** in the input line and pressing F4 will load program Onnnnn into the editor (Program must be in memory).

In the Calculator Help function, this button uses the highlighted Trig, Circular, or Milling data value to load, add, subtract, multiply, or divide with the calculator.

## REAL TIME CLOCK

Included in the software is a real time clock. The current date and time are displayed on the diagnostics screen. The date and time are also applied to any alarms that are generated. The alarm history will display the alarm, the date and time that it occurred (see the alarms section on how to access the alarm history). Floppy disk directory contains file creation date and time. When files are saved to the floppy disk, the directory will now show the file creation date and time. Note that this feature also requires floppy disk driver software rev. 2.0 or later. When outputting a parameter file to a floppy disk or the serial port, it will contain two comments near the top containing the current date and time.

Macro variables are used to set the date and time. Macro variable #3011 contains the date in the format yymmdd (two digit year\* 10000+ month\* 100+ day). Macro variable #3012 contains the time in the format hhmmss (hours\* 10000 + minutes\* 100 + seconds).

## 2.11 KEYBOARD

The control panel keyboard is divided into nine separate regions. They are:

RESET keys	Three (3) keys
FUNCTION keys	Eight (8) keys
JOG keys	Fifteen (15) keys
OVERRIDES	Fifteen (15) keys
DISPLAYS	Eight (8) keys
CURSOR keys	Eight (8) keys
ALPHA keys	Thirty (30) keys
MODE keys	Thirty (30) keys
NUMERIC keys	Fifteen (15) keys

The following are short descriptions of the control panel keys' usage.

**RESET KEYS:** The RESET keys are in the upper left corner of the control panel.

RESET	Stops all machine motion and places the program pointer to the top of the current program.
POWER UP/ RESTART	Automatically initializes the machine at power up. After initial power up, when this key is pressed, the axes zero return and tool one is put in the spindle.
TOOL CHANGER RESTORE	Restores the tool changer to normal operation after the tool changer has encountered an interruption during a tool change. The button initiates a user prompt screen to assist the operator in recovering from a tool changer crash.

A flow chart showing the steps to restore the side mount tool changer is shown in the Programming section.

**FUNCTION KEYS:** Below the reset keys are the function keys. There are eight function keys. They are used to execute special functions implemented throughout the control software.

F1-F4	Used in editing, graphics, background edit, and the help/calculator to execute special functions.
TOOL OFSET MESUR	Used to record tool length offsets in the offset page during part setup.



VS-Series

NEXT TOOL

Used to select the next tool during part setup.

TOOL RELEASE

Releases the tool from the spindle when in MDI mode, zero return, or handle jog. (The remote TOOL RELEASE button is located on the front of the cover to the spindle head. It operates the same as the one on the keypad. It must be held for  $\frac{1}{2}$  second before the tool will be released, and the tool will remain released for  $\frac{1}{2}$  second after the button is released. While the tool is unclamped, air is forced down the spindle to clear chips, oil, or coolant away from the tool holder.

PART ZERO SET

Used to automatically set work coordinate offsets during part setup.

**JOG KEYS:**

The jog keys are on the left below the function keys. These keys select which axes the jog handle sends signals to and provides for continuous jogging. When a key is pressed briefly, that axis is selected for use by the jogging handle. When a key is pressed and held down, that axis is moved as long as the key is held down. If a "+" key is pressed and held, the axis is moved so that the tool position is changed in a positive direction relative to the work coordinates. If a "-" key is pressed and held, the axis is moved so that the tool position is changed in a negative direction relative to the work coordinates. The jog keys are locked out if the machine is running.

+A, -A

Selects the A axis. Selects the B axis when used with the shift key and control is configured with a fifth-axis option.

+Z, -Z

Selects the Z axis.

+Y, -Y

Selects the Y axis.

+X, -X

Selects the X axis.

JOG LOCK

When pressed prior to one of the above keys, the axis is moved in a continuous motion without the need to hold the axis key depressed. Another press of the JOG LOCK key stops jogging motion.

To the left side of the jog keys are three keys to control the chip auger. If the auger is enabled with Parameter 209, these keys perform the following functions:

CHIP FWD

Turns the auger in a direction that removes chips from the work cell.

CHIP STOP

Stops auger movement.

CHIP REV

Turns the chip auger in the reverse direction. The key must be held down for operation.

To the right side of the jog keys are three keys to control the optional automatic spigots. If the spigot is enabled with Parameter 57, these keys perform the following functions:

CLNT UP

Pressing this key positions the coolant stream direction one position higher, if possible.

CLNT DOWN

Pressing this key positions the coolant stream direction one position lower, if possible.

AUX CLNT

Pressing this key while in MDI mode will turn on the Through the Spindle Coolant system, and pressing it again will shut off the system.

**OVERRIDES:**

The overrides are at the lower left of the control panel. They give the user the ability to override the speed of rapid traverse motion, as well as programmed feeds and spindle speeds.

HANDLE CONTROL FEEDRATE Allows jog handle to be used to control feedrate in +/- 1% increments (from 0 to FEEDRATE 999%).

- 10 Decreases current feed rate by 10% (from 10 to 200%).
- 100% Sets control feed rate to programmed feed rate.
- +10 Increases current feed rate by 10% (from 10 to 200%).

HANDLE CONTROL SPINDLE

- 10 Decreases current spindle speed by 10% (from 10 to 150%).
- 100% Sets spindle speed to programmed speed.
- +10 Increases current spindle speed by 10% (from 10 to 150%).
- CW Starts the spindle in the clockwise direction. Except CE machines.
- STOP Stops the spindle.
- CCW Starts the spindle in the counterclockwise direction. Except CE machines.
- 5% RAPID Limits rapid traverse to 5 percent of maximum.
- 25% RAPID Limits rapid traverse to 25 percent of maximum.
- 50% RAPID Limits rapid traverse to 50 percent of maximum.
- 100% RAPID Allows rapid traverse to feed at its maximum.

**DISPLAYS:**

The display keys are in the center at the top. These eight keys provide access to the different displays and operational information and help routines available to the user. Some of these keys are multi-action keys in that they will display different screens when pressed multiple times. The current display is always displayed on the top left line of the video screen.

- PRGRM / CONVRS Displays the currently selected program. Also used in Quick Code applications.
- POSIT Displays the position of the machine axes. Pressing PAGE UP and PAGE DOWN will show operator, machine, work, and distance-to-go formats in large letter format.
- OFSET Displays the tool length and radius offsets. PAGE UP will display the values of the axes' work offsets. If the ORIGIN button is pressed while the offsets are displayed, the control will prompt the user: ZERO ALL (Y/N)? Entering Y will zero all the offsets, in the section displayed.
- CURNT COMDS Displays the current program, modal program values, and position during run time. Succeeding presses of the PAGE DOWN key will display modal values, system timers, macro variables, tool life and tool load information.



VS-Series

## ALARM / MESGS

Shows the full text of an alarm when the alarm message is flashing. Pressing the left or right arrow keys will display an alarm history. Pressing PAGE DOWN will display a page for user messages and notes.

## PARAM / DGNOS

Displays and allows changing of parameters that define machine character. Pressing PAGE UP will display lead screw compensation values. Successive PAGE DOWN presses will display general parameters as well as the X, Y, Z, A and B parameters. A second press of the PARAM key will display the first page of diagnostic data. The first page of diagnostic data is discrete inputs and outputs. Pressing PAGE DOWN will display the second page of diagnostic data that consists of additional inputs and analog data.

## SETNG / GRAPH

Displays and allows changing of user settings. Pressing the SETNG key twice enables graphics mode where the user can debug the current program and view the program's generated tool path.

## HELP / CALC

Displays a brief, on-line manual. Pressing HELP a second time will display the help calculator. There are three pages of calculator help. Pressing the PAGE DOWN key will display milling and tapping help, trigonometry help, or circle help.

**CURSOR KEYS:**

The cursor keys are in the center of the control panel. They give the user the ability to move to various screens and fields in the control. They are used extensively for editing of CNC programs.

## HOME

Context-sensitive key that generally moves the cursor to the topmost item on the screen. In editing, this is the top block of the program. In graphics zoom, it will select full view.



(UP ARROW)

The up arrow moves up one item, block, or field. In graphics, the zoom window is moved up.

## PAGE UP

Used to change displays, move up one page in the editor, or zoom out when in graphics.



(LEFT ARROW)

Used to select individually editable items within the editor, moves cursor to the left. It selects optional data in fields of the settings page and moves the zoom window left when in graphics.



(RIGHT ARROW)

Used to select individually editable items within the editor, moves cursor to the right. It selects optional data in fields of the settings page and moves the zoom window right when in graphics.

## END

Context-sensitive key that generally moves the cursor to the bottom most item on the screen. In editing, this is the last block of the program.

	(DOWN ARROW)	The down arrow moves down one item, block, or field. In graphics, the zoom window is moved down.
PAGE DOWN		Used to change displays, move down one page in the editor, or zoom closer when in graphics.
<b>ALPHA KEYS:</b>		The alpha keys allow the user to enter the 26 letters of the alphabet along with some special characters.
SHIFT		The shift key provides access to the white characters on the keyboard. Pressing SHIFT and then the white character will cause that character to be sent to the control. When entering text, UPPER CASE is the default. To access lower case characters, press and hold the SHIFT key while pressing the appropriate characters. The SHIFT key can also be continuously held down while a number of other keys are pressed.  When a control has a fifth-axis installed, the B axis is selected for jogging by pressing SHIFT and then the +,-A keys.
EOB		This is the END-OF-BLOCK character. It is displayed as a semicolon on the screen and it signifies the end of a programming block. It is the same as a carriage return and then a line feed.
( )		The parenthetical brackets are used to separate CNC program commands from user comments. They must always be entered as a pair and may or may not have additional characters separating them. Any time an invalid line of code is received through the RS-232 port while receiving a program, it is added to the program between these two brackets.
/		The right slash is used as a block delete flag. If this symbol is the first symbol in a block and a BLOCK DELETE is enabled, then that block is ignored at run time. The symbol is also used for division in macro expressions.  In some FANUC compatible controls, the block delete symbol can be used to choose between two options when the "/" symbol is not at the beginning of the line. For instance, in the following line, T2 is executed when the block delete option is off, and when the block delete option is on, T1 is executed.
		T1 / T2; N1 G54
		This cannot be done on a HAAS control.
		A coding method for achieving the same results on a HAAS control is given below:
		/ T2 M99      (T2 executed when block delete is off) T1                (T1 executed when block delete is on) N1 G54
[ and ]		Square brackets are used in macro expressions and functions.

**MODE KEYS:**

The mode keys are in the upper right part of the control panel. These keys change the operational state of the CNC machine tool. There are six major operation modes. The user can enter a specific mode by pressing the desired "arrow" shaped key on the left. The keys in the same row as the pressed mode key are then made available to the user. Otherwise, these keys are not available. The current mode is always displayed on the top line just to the right of the current display on the video screen.

**EDIT****Selects edit mode.**

INSERT	Inserts the text in the input buffer after the current cursor location. Also used to copy blocks of code in a program.
ALTER	Changes the item that the cursor is on to the text in the input buffer. Places an MDI program in the program list.
DELETE	Deletes the item that the cursor is on.
UNDO	Backs out or undoes up to the last 9 edit changes.

**MEM****Selects MEM mode.**

SINGLE BLOCK	Turns single block <b>on</b> so that when the cycle start button is pressed, only one block of the program running is executed.
DRY RUN	Used to check actual machine movement without cutting a part. Programmed feeds are replaced by the speed keys in the handle jog row.
OPT STOP	Turns on optional stops. If an M01 code is encountered in the program and OPT STOP is on, a stop is executed. Depending on the lookahead function, it may not stop immediately. If the program has been interpreted many blocks ahead, and OPT STOP is pressed, then the nearest M01 may not be commanded. See G103. <ol style="list-style-type: none"><li>1. OPT STOP will take effect on the line after the highlighted line when OPT STOP is pressed.</li><li>2. M01 is not allowed during cutter compensation. Alarm 349 will be generated in this case, as for M02, M30, and M00.</li></ol>
BLOCK DELETE	Blocks with a slash ("/") as the first item are ignored or not executed when this option is enabled. If a slash is within a block, address codes after the slash will be ignored until after the block, if this option is enabled. <ol style="list-style-type: none"><li>1. When not in cutter compensation, block delete will take effect two lines after BLOCK DELETE is pressed.</li><li>2. When in cutter compensation, blocks must be processed earlier. Therefore, block delete will not take effect until at least four lines after the highlighted line when BLOCK DELETE is pressed.</li><li>3. If BLOCK DELETE changes state during the processing of the first block of a chamfering/rounding pair, and at least one of the pair is block-deleted, the behavior is undefined.</li><li>4. Processing will slow down for paths containing block deletes during high-speed machining, because the lookahead queue will be emptied as processing approaches the block-deleted line(s). This limits the speed at which the previous blocks can run.</li></ol>

<b>MDI/DNC</b>	<b>Selects MDI or DNC mode.</b>
COOLNT	Turns the coolant on and off.
ORIENT SPINDLE	Rotates the spindle to a known position and then locks the spindle. Can be used during setup to indicate parts.
ATC FWD	Rotates the tool turret forward to the next sequential tool. If <b>Tnn</b> is in the input buffer, the turret will advance to tool <b>nn</b> .
ATC REV	Rotates the tool turret backwards to the previous tool. If <b>Tnn</b> is in the input buffer, the turret will advance to tool <b>nn</b> .
<b>HANDLE JOG</b>	<b>Selects Jogging mode.</b>
.0001, .1	.0001 inches or .001 mm for each division on the jog handle. For dry run .1 inches/min.
.001, 1.	.001 inches or .01 mm for each division on the jog handle. For dry run 1. inches/min.
.01, 10.	.01 inches or .1 mm for each division on the jog handle. For dry run 10. inches/min.
.1, 100.	0.1 inches or 1.0 mm for each division on the jog handle. For dry run 100. inches/min.
<b>ZERO RET</b>	<b>Selects Zero Return mode.</b>
AUTO ALL AXES	Searches for all axes' machine zero.
ORIGIN	Zeros out various displays and timers.
ZERO SINGLAXIS	Searches for machine zero on the axis that is specified in the input buffer.
HOME G28	Returns all axes to machine zero in rapid motion. Does not search.
SINGLE AXIS HOME G28	Either the X, Y, Z, A, or B axis can be returned to zero alone. The operator enters 'X', 'Y', 'Z', 'A' or 'B', then presses the HOME G28 key.  Pressing HOME G28 without first entering an axis letter will cause all enabled axes to be returned to zero.  If the chosen axis is disabled, the message DISABLED AXIS will be generated.
<b>CAUTION!</b>	There is no warning message to alert the operator of any possible collision. For example, if the Z-axis is down in amongst parts on the table when X or Y is zeroed a crash can result.

**SECOND HOME**

This is an optional feature. When this button is pressed, the control will rapid all axes (which have the 2ND HOME BTN bit =1) to the coordinates specified in Work Offset G129. The sequence is as follows: First, assuming the X or Y axis need to be moved, the Z axis is returned to zero, then the X and Y axes are moved to their final positions, then the Z axis is moved to its final position. G129 Work Offsets must be set to the desired values for this feature to work correctly. This feature will work in any mode except DNC, just like the HOME G28 button.

**LIST PROG**     **Selects Program List mode and displays a list of the programs in the control.**

SELECT PROG	Makes the highlighted program on the program list the current program. The current program will have an asterisk preceding it in the program list.
SEND RS232	Transmits programs out the RS-232 serial port. If <b>ALL</b> is highlighted, all the programs will be sent with one "%" at the beginning and one at the end of the stream.
RECV RS232	Receives programs from the RS-232 serial port. Unless <b>ALL</b> is highlighted, enter a valid program name in the form Onnnnn before pressing RECV RS232. If <b>ALL</b> is highlighted, do not enter a program name. The program names will be entered automatically from the input stream data.
ERASE PROG	Erases the highlighted program or the program specified in the input buffer.

**NUMERIC KEYS:** The numeric keys give the user the ability to enter numbers and a few special characters into the control.

CANCEL	The Cancel key is used to delete the last character entered during editing or field input.
SPACE	This is a space and can be used to format comments placed into programs.
WRITE / ENTER	This acts as the general purpose enter key. Any time that user needs to change any information in the control, this key is pressed.
-, .	Used to negate numbers, or provide decimal precision.
+, =, #, and *	These symbols are accessed by first pressing the SHIFT key and then the key with these symbols. They are used in macro expressions.
?, %, \$, !, &, @, and :	These are additional symbols, accessed by pressing the SHIFT key. They can be used in program comments.

## 2.12 POWER ON / OFF

### POWER ON

There is only one way to turn on this CNC. This is by pressing the green “On” button at the top left of the control panel. The main breaker at the rear of the mill must be on before this button will turn on the mill. Any interruption to power will turn the mill off and this button must be used to turn power back on again.

Upon power up, the machine must find its fixed reference point before any operations can occur. After power on, pressing the POWER UP/RESTART will establish this point. The ZERO RET mode and AUTO ALL AXES button may also be used to initialize the system after all alarms are cleared. A single axis can be selected by first pushing the **X**, **Y**, **Z**, **A**, or **B** key and then the ZERO SINGL AXIS key. The position thus found is used as machine zero. Note that the Z-axis will shift downwards about five inches as the search for zero is finished; so, keep clear.

**CAUTION !!!!** After power on, the machine does not know its home position or stored stroke limits until it has been zero returned by the POWER UP/RESTART key or the ZERO RET/AUTO ALL AXES key. It is possible to jog the machine with the handle or jog keys at the lower feeds. If it is jogged unchecked in either direction, you may damage the sheet metal covers or overload the ball screws. To avoid this, always properly ZERO RET the machine immediately after power on before doing anything else.

---

**NOTE:** Tool changer goes to tool #1 first, then to tool designated in Setting 81. After initializing, all machine position displays are reset to zero.

The HOME G28 key should be used any time after the initial power up. This will return the Z-axis first and then the **X**, **Y**, **A**, and **B** axes all at rapid rate. If the Z-axis is positioned above the machine zero, the **X**, **Y**, **A**, and **B** axes are moved first. This key will work in any of the operating modes. The manual G28 button does not use any intermediate return point the way the programmed G28 does. Any auxiliary axes (**C**, **U**, ...) are returned to home after **X**, **Y**, **Z**, **A**, and **B**.

---

**NOTE:** Repairs to the motor, ball screw, or home switch will affect the zero return point and must be done only by a factory trained technician. Serious damage to the ball screw, way covers, linear guides, or tool changer may occur if the zero return point is not properly set.

### POWER OFF

Pressing the red POWER OFF button will remove power to the machine instantly. The machine can also be programmed to turn off at an end of cycle (M30) or after a preset amount of time that the machine sees no activity. These are Settings 1 and 2.

A sustained overvoltage condition or sustained overheat condition will also shut this machine off automatically. If either of these conditions exists for 4.5 minutes, the machine will start the 30 second auto-shutdown. Alarm 176 is displayed when an overheat shutdown begins and Alarm 177 is displayed when an overvoltage shutdown begins.

Any power interruption, including the rear cabinet main circuit breaker, will also turn this machine off. Power must be restored and the POWER ON button pressed to restore operation.



## 2.13 MANUAL OPERATION

### MDI

Manual data input allows you to enter data that can be executed on a line by line basis instantly without having to use the EDIT and MEM modes. In this control, MDI is actually a scratch pad memory that can execute many lines of instruction without having to disturb your main program in memory. The data in MDI will be retained even when switching modes or in power off.

Editing with MDI is the same as memory editing.

The MDI mode also allows for manual operation of coolant, spindle, and tool changer.

A program in MDI can be saved as a normal named program in memory by placing the cursor at the beginning of the first line (HOME), typing **Onnnnn** (new program number), then pushing ALTER. This will add that name to the program list and clear MDI.

The entire MDI program may be cleared by pressing the ERASE PROG key while in MDI.

A fast way to select a tool is to type **Tnn** and, instead of INSERT, press either ATC FWD or ATC REV. This will directly select that tool.

When DNC is enabled with Setting 55, a second push of the MDI button will put the control into DNC mode.

When the Parameter 57 flag DOOR STOP SP is set to 1, the maximum spindle speed is 750 RPM with the doors open.

### HANDLE/JOG

Manually moving the axes is accomplished by pressing the mode button labeled HANDLE JOG and then by using the JOG keys or the Handle to move the axis. Both the JOG buttons and the Handle are enabled simultaneously without needing to select between them. The display is changed to the Position Display and the currently selected axis for jogging will blink.

Jog feed rate or handle resolution is selected by the four keys to the right of the HANDLE JOG key. Jog feeds from .1 inch per minute to 100 inch per minute or handle divisions from .0001 inch to .1 inch are selectable. Auxiliary axes can be manually jogged from the front panel.

During jogging, the FEED RATE override buttons will adjust the rates selected from the keypad. This allows for very fine control of the jog speed. It does not change the handle step size.

In the center of the jog buttons is a key labeled JOG LOCK. This key will cause the axis you are jogging to continue jogging even after you release the key. Press this key and then press the selected axis motion key to start. Motion will stop as soon as the JOG LOCK button is pressed again, or RESET is pressed.

---

**NOTE:** Selecting another axis will cause that axis to move.

This feature is handy, for example, when you are slow milling the soft jaws of a vise.

In order to select another axis for jogging while using the handle, use +/- X, Y, Z, or A buttons. When one of these buttons is pressed, that axis is selected for HANDLE JOG but does not move unless the button is held down for more than ½ second. After ½ second, that axis is moved in the selected direction and at the selected feed rate.

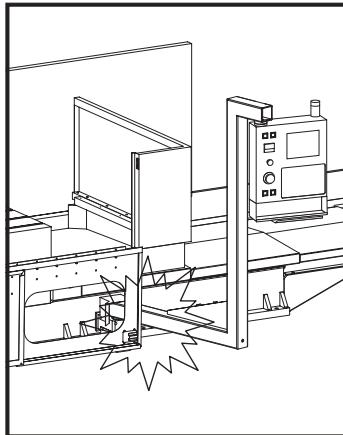
All aspects of handle jogging for the fifth axis work as they do for the other axes. The exception is the method of selecting jog between axis **A** and axis **B**. By default the '+A' and '-A' keys will select the **A** axis for jogging. The display will show "JOGGING A AXIS HANDLE .01" while you are jogging the **A** axis. The **B** axis can be selected by pressing the SHIFT key, and then either the '+A' or '-A' key. When this is done the control will switch to jogging the **B** axis and the display will change to "JOGGING B AXIS HANDLE .01".

The axis assigned to the '+A' and '-A' keys will remain selected for jogging even if the operating mode is changed or if the machine is turned off. The selected axis for '+A' and '-A' can be toggled by pressing the shift key prior to pressing the '+A' or '-A' keys.

---

**CAUTION:** Do not move the table with the doors folded down.

---





## 2.14 AUTOMATIC OPERATION

### OPERATION MODE

There are six modes of operation of the VF Series CNC Mill. They are:

EDIT	Used to make manual changes to a part program.
MEM	Used to run a users part program stored in memory.
MDI/DNC	Used to quickly manually enter and run a program.
HANDLE/JOG	Used to move the axes with the handle or JOG buttons.
ZERO RET	Used to search for machine zero and to return to machine zero automatically.
LIST PROG	Used to list, send, receive and delete programs.

Changes to the mode are made by pressing of the buttons on the top right quadrant of the keypad that have the above labels. If an operation is started, such as running a program, you cannot change modes until the operation is stopped (motion is stopped by pressing feed hold or reset). The six mode selection buttons are arranged vertically and, generally, the keys to their right apply only in that selected mode.

### PROGRAM SELECTION

Program selection is done from the LIST PROG mode. This mode will list all of the programs stored in memory and allow you to select one as the current program. This is the program that will be run when you press START in MEM mode. It is the program with the “\*” on the LIST PROG display. It is also the one that you will see on the EDIT display.

To select an existing program, press the CURSOR **up** or **down** buttons until the program you want is highlighted and then press the SELECT PROG button. The “\*” will move to that program.

To select a new program (create a new program) or to select an existing program, you may also enter **Onnnnn** from the keyboard and then press SELECT PROG button.

### STARTING AUTOMATIC OPERATION

Before you can run a program, it must be loaded in the current memory. To select a program, push the LIST PROG mode key. Use the cursor to find the desired program and then push SELECT PROG. The program list includes the program name and the first comment. If the control was turned off while running, that program will automatically be in current memory and selected.

If the machine has just powered up, you need to first push the POWER UP / RESTART key. This will initialize all axes and the tool changer, display the Current Commands, and go to MEM mode with the control ready to run. Pushing the CYCLE START button in the lower left of the control panel will begin execution.

To start a program other than at the beginning, scan to the block number using the **down** arrow or PAGE DOWN until you reach the desired starting place. Push the MEM key and CYCLE START to begin. The Program Restart function, selected from Setting 36, will change the way a program operates if you start from other than the first block. The setting called Program restart “ON” will ensure that the correct tool and axis positions are selected when you start from part way through a program.

Any errors in your program will cause an alarm and stop the running of the program. Typical alarms are travel limits and missing **I**, **J**, and **Q** codes. Attempts to move outside of the limits of travel will also cause an alarm.

At any time that a program is running, the bottom left corner of the display will show RUNNING. If it does not show this, the program has completed, has been stopped by the operator, or has been stopped by a fault condition.

### *PROGRAM RESTART*

Program Restart is designed to help the operator start a program from the middle while still properly interpreting all of the preceding lines of the program. To use program restart, turn on setting 36, move the cursor pointer to where you want to start (the restart point), and press Cycle Start. You do this by using the CURSOR up and down keys in MEM mode. The control will begin "invisible" interpretation of the program from the beginning and you will see the cursor move through the program. When it gets to the line you wanted to start on, the control will establish all of the conditions that would normally have been true at the end of the previous line and then execute the highlighted line and the rest of your program. Most program interpretation alarms which you might have gotten in the invisible phase will not occur until after the line at your restart point is complete.

As an example, if the following program were to restart at the T2 line, the control would change to T1 and then change to T2 before start axes motion. If the H and T agreement setting (15) was on, you would still not get an alarm. The double tool change is probably the most difficult to understand. The control does that because it insures that everything that must be true on the previous line is established and that means go tool 1. It then executes the restart line which says go to tool 2

```
O0123 ;
T1 M06 :
G00 G90 G54 X0 Y0 :
G01 F20. Z-2. ;
T2 M06 H03;           (RESART HERE)
G00 G90 G54 X0 Y0 ;
G01 F20. Z-2 ;
G28 ;
M30 ;
```

### *STOPPING AUTOMATIC OPERATION*

There are several ways a program can be stopped. They include both normal stops and abnormal, or alarm caused, stops. The normal stops are:

- 1) Normal completion at M00, M01, M02, or M30.
- 2) A FEED HOLD stop by the operator. The program is continued by pressing CYCLE START again.
- 3) A SINGLE BLOCK stop when operator selected. The program is continued by pressing CYCLE START again.
- 4) A Door Hold stop caused by the operator opening the enclosure doors. The condition changes to Feed Hold after the doors are closed and the operator must press cycles start to continue.



---

Note: Even the Door hold override setting will not change this characteristic with the machines with their "as shipped" default parameters. The parameter 57:Door Stop Sp parameter must be set to zero in order to change this performance BUT this is a less safe condition.

When parameter 57:Door Stop Sp is zero, opening the door will stop a program and closing the door will result in the program continuing. The door hold override setting will work also and allow the control to run even with the door open.

The abnormal stops are:

1) **Operator Reset**

This stops all axes' motion, stops the tool changer, turns off the spindle, and turns off the coolant pump. Program operation cannot be continued from the stopping point. If Setting 31 is On, the program pointer is reset to the beginning.

2) **Emergency Stop**

This stops all axes' motion, disables the servos, stops the tool changer, turns off the spindle, and turns off the coolant pump. Program operation cannot be continued from the stopping point. This will also stop any auxiliary axes' motion. RESET must be used at least twice to remove the alarms and start again.

3) **Alarm condition**

This can occur any time an alarm comes on during program operation. Since a program cannot be restarted until RESET is pressed, a program execution cannot be continued from the stopping point. Alarms can be caused by programming errors or machine faults. Use the Graphics simulation mode to test your program first for errors.

4) **Power-off**

This will stop all motors within one second but does not guarantee any conditions when the machine is powered-on again.

### EMERGENCY STOP SWITCH

The EMERGENCY STOP switch is normally closed. If the switch opens or is broken, power to the servos will be removed instantly. This will also shut off the tool changer, spindle drive, and coolant pump. The EMERGENCY STOP switch will shut down motion even if the switch opens for as little 0.005 seconds.

Be careful of the fact that Parameter 57 contains a status switch that, if set, will cause the control to be powered down when EMERGENCY STOP is pressed.

You should not normally stop a tool change with EMERGENCY STOP as this will leave the tool changer in an abnormal position that takes special action to correct.

Note the tool changer alarms can be easily corrected by first correcting any mechanical problem, pressing RESET until the alarms are clear, selecting ZERO RETURN mode, and selecting AUTO ALL AXES.

If the shuttle should become jammed, the control will automatically come to an alarm state. To correct this, push the EMERGENCY STOP button and remove the cause of the jam. Push the RESET key to clear any alarms. Push the ZERO RETURN and the AUTO ALL AXES keys to reset the Z-axis and tool changer. Never put your hands near the tool changer when powered unless the **EMERGENCY STOP** button is pressed.

### WORK BEACONS

The red and green work beacons located directly on top of the control arm allow the operator to monitor the machine status.

While a program is running normally, the GREEN beacon will be on.

The beacon will *flash* GREEN if: - the operator selects FEEDHOLD or SINGLE BLOCK stop.

- the control is in a M00, M01, M02, M30. It will stop flashing when RESET is pressed. If the control is in an M02 or M30, and door hold overide is not on, the beacon will stop flashing when the door is opened.

The beacon will *flash* RED if: - the control encounters an alarm, such as when EMERGENCY STOP is pressed. It will stop flashing when RESET is pressed to clear all alarms.



## 2.15 PART PROGRAM STORAGE AND EDIT

When using anything other than HELP or Messages function, alphanumeric key entries are displayed along the bottom line of the display. This is called the data entry line. When the line contains what you want to enter, press the WRITE, ALTER, or INSERT key as appropriate.

When the HELP display is selected, the alphanumeric keys are used to select one of the topics; so they are not displayed on the data entry line of the display.

When the Message function is selected, the cursor is positioned on the screen and you type directly into the display.

### CREATING PROGRAMS

To create a new program, you must be in the PRGRM/CONVRS display and LIST PROG mode. Enter **O** (letter, not number) and a five digit program number and press SELECT PROG key or ENTER. The selected program is the "Main" program and is the one you will see in the MEM and EDIT modes. Press EDIT to show the new program. A new program will begin and consist of only the **Onnnnn** and an EOB (;). All further entries are made by typing a letter followed by a numeric value and pressing INSERT, ALTER, or WRITE. All items entered into a program are either addressed data (a letter of the alphabet followed by a number), or a comment (text surrounded by parenthesis; maximum of 80 characters) or the End-Of-Block (EOB or ;).

The CURSOR **up** and **down** keys can be used to search for the entered value. Simply enter the value to search for on the bottom line and press the CURSOR **up** or **down** keys. The CURSOR **up** key will search for the entered item backwards to the start of the program. The CURSOR **down** key will search forward to the end of the program. Searching also works in MEM mode. If you enter a letter without a number, the search will stop on the first use of that letter with any value.

---

**NOTE:** When INSERT is pressed, the new data is put in after the highlighted data. The CURSOR **up**, **down**, **left**, and **right** keys are used to select the highlighted item. The PAGE UP and PAGE DOWN keys move farther distances and the HOME and END keys go to the start or end of the program. All of these keys work in EDIT, MEM, and MDI modes.

A comment can be edited without entering the entire comment again. Simply highlight the characters you wish to change, enter the new characters, and press ALTER. To add characters move the cursor to where text is to be added, enter the new characters, and press INSERT. To remove characters highlight the characters and press DELETE. Use the UNDO button to reverse any changes. The UNDO button will work for the last nine entries.

After creating a program, the name can be very easily changed by simply altering the **Onnnnn** on the first line. If the maximum number of programs are already present, the message "DIR FULL" will be displayed and the program cannot be created.

### EDITING PROGRAMS

The EDIT mode is used to make changes to a program already in memory or use the LIST PROG mode to create a new one. A newly created program contains only the program **Onnnnn** name and an EOB.

To enter the EDIT mode, press the EDIT mode key. The screen will display the current program. If no program file exists, program O0000 will be displayed. To change a program name, move the cursor to the existing **Onnnnn**, type in the letter "**O**" followed by a five digit number, such as O12345, and press the ALTER key. The upper right hand screen will display the new program number. Your data will first appear in the lower left screen and will be input to the upper screen upon pressing INSERT, ALTER, or WRITE.

To enter a program from the keypad, type in the data you wish and press the INSERT key. More than one code, such as **X**, **Y**, and **Z**, can be entered before you press INSERT. After a program is entered, you may wish to change the data. Highlight the characters you wish to change, enter the new characters, and press ALTER. To add characters move the cursor to where text is to be added, enter the new characters, and press INSERT. To remove characters highlight the characters and press DELETE. Use the UNDO button to reverse any changes. The UNDO button will work for the last nine entries.

The CURSOR **up** and **down** keys can be used to search for the entered value. Simply enter the value being searched for on the bottom line and press the CURSOR **up** or **down** keys. The CURSOR **up** key will search for the entered item backwards to the start of the program. The CURSOR **down** key will search forward to the end of the program. Searching also works in MEM mode. If you enter a letter without a number, the search will stop on the first use of that letter with any value.

You can change to a different program while in the EDIT or MEM mode by using the CURSOR **up** and **down** keys, enter **Onnnnn** on the input line and then press the CURSOR **up** and **down** keys or the **F4** key. **Onnnnn** is the program you wish to change to.

The jog handle can be used to move the cursor during editing. Parameter 57 is used to enable this function. If enabled, the handle will act like the CURSOR **left** and **right** buttons.

Background editing is also possible with this machine and is now a standard feature. If background editing is available on your control, all of the above editing functions can be used while a program is running in MEM.

#### Editing error messages:

Guarded Code	You tried to remove the <b>Onnnnn</b> from start of a program.
Bad Code	A line contained invalid data or comment over 80 characters.
Editing Error	Some previous edit was not completed; fix the problem or press UNDO.
Bad Name	Program name <b>Onnnnn</b> is invalid or missing.
Invalid Number	The number with an alphabet code was invalid.
Block Too Long	A block may only be 256 characters.
No Code	An insert was done without any data to insert.
Can't Undo	May only use undo for previous nine changes.
End Of Prg	End of prog EOB cannot be deleted.

#### BACKGROUND EDIT

With BACKGROUND EDIT, you may edit a program in memory while any other program is being run. BACKGROUND EDIT can be enabled and disabled by Parameter 57.

BACKGROUND EDIT is selected from MEM mode when in PROGRAM DISPLAY by typing **Onnnnn** for the program you want to edit and pressing F4. If you do not enter the **Onnnnn**, you will instead get the PROGRAM REVIEW display.

While in BACKGROUND EDIT, you may perform any of the operations available in the EDIT mode. The last five lines of the display will, however, display the status of the running program and the top line will show the name and line number of the running program.

Selecting any other display or pressing F4 will exit from BACKGROUND EDIT. In order to list the programs that are in memory, a display function is available to view the program memory list while a program is running. This display is called LIST. It is selected by pressing the LIST PROG button while a program is running. The display is just like the LIST PROG mode display but it does not allow any send, receive, copy, select, or erase functions.

The CYCLE START button may not be used while in BACKGROUND EDIT. If the program contains an M00 stop, you must exit BACKGROUND EDIT and then press CYCLE START to resume the program.



All of the changes made during BACKGROUND EDIT are saved in a different memory area until the running program stops. This means that you can even edit the program that is running, or any of its subprograms, and those changes will not affect the running program.

The first time you select a program for BACKGROUND EDIT, you will get the message PROG EXISTS if the program is already in memory or NEW PROG if it is not. The NEW PROG message means that the program is being created and will be initially empty. In either case, you will then be able to edit that program. The second time you select a program for BACKGROUND EDIT without stopping the running program, you will get the message SECOND EDIT.

BACKGROUND EDIT is not available from MDI or from DNC operating modes

#### *DELETING PROGRAMS*

To delete an existing program, you must be in LIST PROG mode. The programs will be listed here by program number. Use the CURSOR up or down keys to highlight the program number, or type in the program number at the blinking cursor, then press the ERASE PROG key.

All the programs in the list may be deleted by selecting ALL at the end of the list and pressing the ERASE PROG key. Use caution when deleting single programs, and read all prompts, to ensure that ALL programs are not selected. The UNDO key will not recover programs that are deleted.

#### *SPECIAL FUNCTION KEYS*

The F1, F2, F3, and F4 buttons perform different functions depending on what display and mode is selected. The following is a quick summary of the **Fn** buttons:

- F1      In EDIT mode and PROGRAM DISPLAY, this will start a block definition.  
In LIST PROG mode, F1 will duplicate a program already stored and give it a new name from the command line. In the Advanced Editor, F1 will open the drop-down menus. In OFFSET display, F1 will set the entered value into the offsets.
- F2      In EDIT mode, PROGRAM DISPLAY, this will end a block definition.  
In OFFSET display, F2 will set the negative of the entered value into the offsets.
- F3      In EDIT and MDI modes, the F3 key will copy the highlighted circular help line into the data entry line at the bottom of the screen. This is useful when you want to use the solution developed for a circular motion. Push INSERT to add that circular motion command line to your program. In the calculator HELP function, this button copies the value in the calculator window to the highlighted data entry for Trig or Circular Help.
- F4      When in EDIT mode with no program running, entering **Onnnnn** in the input line and pressing F4 will change the program being edited to **Onnnnn**.

When in MEM mode and PROGRAM DISPLAY, F4 can be pressed to select either BACKGROUND EDIT or PROGRAM REVIEW. BACKGROUND EDIT is selected by entering the program number at the input line and pressing F4. BACKGROUND EDIT can only be selected when a program is running. PROGRAM REVIEW can be selected whether or not a program is running, simply by pressing F4. If a program is running, PROGRAM REVIEW will show the running program on the left half of the screen, and allows the operator to review the program on the right half of the screen.

In the calculator HELP function, F4 uses the highlighted Trig, Circular, or Milling data value to load, add, subtract, multiply or divide with the calculator.

#### *THE UNDO KEY*

A very powerful keyboard button available in this control is the UNDO button. When editing, this button will allow you to basically undo any changes or edits you have made but wish you had not. Any time you use the INSERT, ALTER, or DELETE buttons, the condition of the original block is saved and can be restored with the UNDO button. In fact, the previous nine changes can be undone in the opposite order that they were entered by pressing the UNDO button for each change that is to be backed out.

The UNDO button can be used in EDIT, BACKGROUND EDIT, and MDI. But if you change operating modes between EDIT and MDI, you cannot use the UNDO button as the list of saved data is cleared.

### BLOCK OPERATIONS

Block operations can be performed on a group of one or more blocks of the program. These operations include BLOCK DUPLICATE, BLOCK MOVE, and BLOCK DELETE. Prior to a block being defined, the bottom right of the screen shows how to define a block; the F1 key is pressed when the cursor is on the first line of the block and the F2 key is pressed when the cursor is on the last line of the block.

Once a block is defined, the lower right of the screen shows how to manipulate the block; the INSERT key is used to duplicate the defined block wherever the cursor is positioned, the DELETE key is used to delete the block, the ALTER key is used to move the block, and the UNDO key cancels the block definition.

When a block is defined, the cursor is indicated by the “>” symbol and is always at the beginning of a line. When a block is copied or moved, the lines are added after the block with the cursor. Only whole command lines may be moved with the block functions.

Parts of programs can be copied from one program to another with the block copy feature. This is done by highlighting the section of code that is to be copied using the F1 and F2 keys. Once a section of code is highlighted, you then change to another program by selecting an existing program or create a new one. Cursor to the location that the previously defined block is to be inserted and press the INSERT or write key. A copy of the defined block will be inserted into the current program and the copied code segment becomes the currently-defined block. Press the UNDO key to exit the BLOCK COPY mode.

Blocks of code can be copied into an MDI program, but blocks of code cannot be copied from an MDI program into another program. You can always rename the MDI program and then copy its text to any other program in the manner described above.



VS-Series

## 2.16 PART PROGRAM INPUT/OUTPUT

### RS-232 DATA INPUT/OUTPUT

Programs are sent or received through the first RS-232 port located on the rear control box pendant side. All data sent or received is ASCII. In order to use this port, you will need to obtain a cable and connectors with the following wiring:

pin #1	Shield Ground	pin #2	TXD-Transmit Data
pin #3	RXD-Receive Data	pin #4	RTS (optional)
pin #5	CTS (optional)	pin #7	Signal Ground

**Cables for the RS- 232 must be shielded**

The following lists baud rate and the respective maximum cable length. This list assumes proper cable shielding and no signal boost.

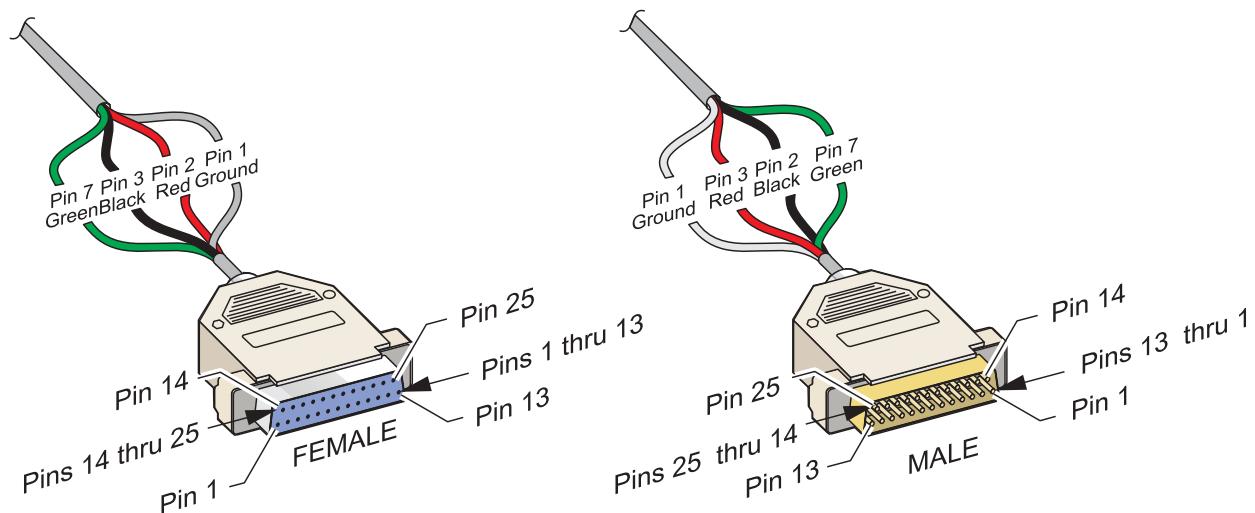
9,600 baud rate: 100feet (30m) RS-232

38,400 baud rate: 25 feet (8m) RS-232

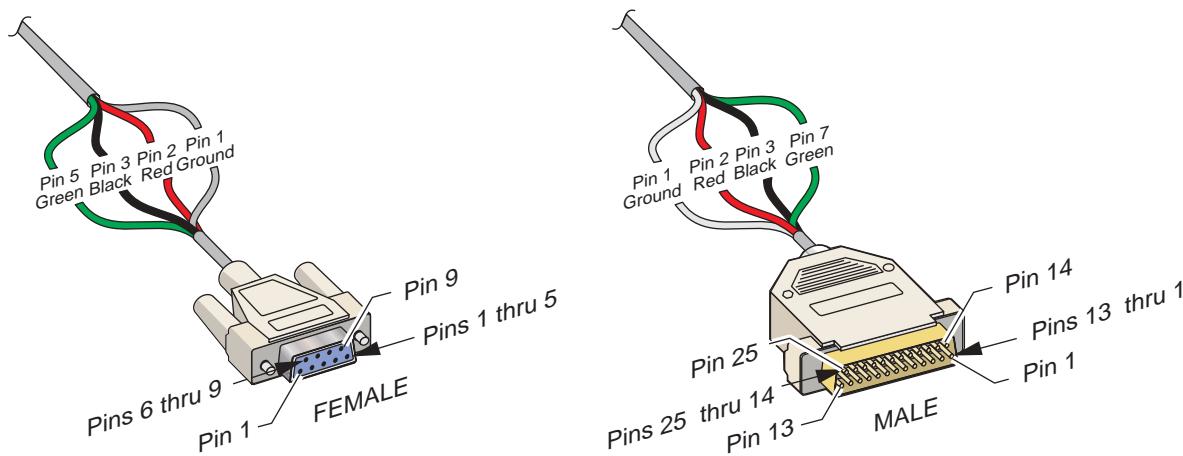
115200 baud rate: 6 feet (2m) RS-232

It is possible to use RS-232 to RS-422 converters on each end of the cable to accomplish longer cables at up to 115,200 baud. Proper twisted pair wire greatly improves reliability and increases maximum distance. A pentium processor should be used for baud speed of 115,200.

#### 25-25 Pin



## 9-25 Pin



All other pins are optional and are not usually used. The RS-232 connector is a DB-25 and is wired as a DTE. This means that we send data on the TXD wire and receive data on the RXD wire. If you do not understand this, your dealer will be glad to help.

All RS-232 data is ASCII but the number of bits, parity and speed can be changed from settings. The number of data bits is selected with Setting 37 for either 7 or 8. Parity is selected with Setting 12 and is none, even, odd, or zero. Zero parity will always set the parity bit to 0. The data speed is selected with Setting 11.

Once the connection to your computer has been made and verified, go to the Setting page and set the baud rate, parity, number of stop bits, end of block (EOB) format, and leader parameters to match your requirements.

All programs sent to the control must begin with a line containing a single % and must end with a line containing a single %. All programs sent by the control will have these % symbols.

To receive a program, push the LIST PROG key. Move the cursor to the word ALL and push the RECV RS-232 key and the control will receive all main and sub programs until it receives a % sign indicating end of input. Please note that when using "ALL", all your programs must have an address **Oxxxxx** to be filed. If you do not have a program number, type in a program number before you push RECV RS-232 and the program will be stored under that number. You can also select an existing program for input and it will be replaced. An ASCII EOF character (code 04) will also terminate input. The colon character (:) may be used in place of the O for a program name, but it is always displayed as O.

When receiving RS-232 data, there is a status message at the bottom of the screen. It will update as follows:

WAITING	When you first press RECV RS-232.
LOADING XXX	When first % is received; if in XMODEM, XXX is the current block being loaded.
LOADING Onnnnn	When program name is received.
RS232 DONE	When complete and last % is received.
RS232 ABORT	When anything causes abnormal stop.



To send a program, use the cursor as above to select the program and push the SEND RS-232 key. You can select "ALL" to send all of the programs in memory. A setting can be turned on to add spaces to the RS-232 output and improve the readability of your programs.

The synchronization protocol used to send data to slower computers is selected from the Setting 14. Setting 14 may be set to XON/XOFF, RTS/CTS, DC CODES, or XMODEM. Transmission can be stopped with either the XON/XOFF characters or the RTS/CTS wires.

Parameters, settings, offsets, and macro variables pages may also be sent individually via RS-232 by selecting the "LIST PROG" mode, selecting the desired display screen, and pushing the SEND key. They can be received by pushing the RECV key.

The settings that control RS-232 are:

- 11 BAUD RATE
- 12 PARITY
- 13 STOP BITS
- 14 SYNCHRONIZATION
- 24 LEADER TO PUNCH
- 25 EOB PATTERN
- 37 NUMBER DATA BITS

The EOB (semicolon) character is not normally sent by the RS-232 port. If it is received by the input port, it will cause a blank line in the program.

The format of data sent and received for parameters, settings, and offsets is the following:

```
%  
N0 Vnnnnnn  
N1 Vnnnnnn  
N2 Vnnnnnn  
. . .  
%
```

The following table shows the use of the N numbers in the output file when offsets are saved to RS-232 or disk.

Tool length:

Tool/function	length	wear	diameter	wear
1	001	201	401	601
2	002	202	402	602
3	003	203	403	603
:	:	:	:	:
200	200	400	600	800

The nnn numbers for work offsets are as follows:

Offset/axis	X	Y	Z	A	B
G52	801	802	803	804	805
G54	806	807	808	809	810
G55	811	812	813	814	815
:	:	:	:	:	:
G59	831	832	833	834	835
G110	836	837	838	839	840
G111	841	842	843	844	845
:	:	:	:	:	:
G128	926	927	928	929	930
G129	931	932	933	934	935

The nnnn numbers for coolant spigot positions are as follows:

1	936
2	937
3	938
:	:
200	1135

The format of data sent and received for macro variables is the same as above except there is a line N9999, and no line N0. The **N** number is the data number and **V** is the value. N0 is a CRC code that is computed by the control prior to sending the data. The N0 value is mandatory with parameters but is optional with settings and offsets. If you make a change to some saved data value and leave the old CRC, you will get an alarm when you try to load that data. With settings and offsets, you should delete the N0 line if you make changes to the saved data.

---

**NOTE:** Data will be loaded even though an alarm has been generated.

Data that is received garbled is usually converted into a comment and stored into your program while an alarm is generated. In addition, any parity errors or framing errors will generate an alarm and they will also stop the receive operation.

At the end of a send or receive function, the bottom left corner of the display will show either: "RS232 DONE" for normal completion or "RS232 ABORT" if any errors cause it to stop. The actual errors are listed in the ALARM display.

The Haas CNC serial ports optionally support the full DC1, DC2, DC3, DC4 code sequence that is compatible with paper tape readers and punches. Setting 14 is used to select this mode of operation. Setting 14 can be set to "DC CODES". When this setting is selected, the following occurs:

- 1) When sending out of the serial port, a DC2 (0x12) will precede all other data. This code is used to turn on a paper punch.
- 2) When sending out of the serial port, a DC4 (0x14) will follow all other data. This code is used to turn off a paper tape punch.
- 3) When receiving from the serial port, a DC1 (0x11 Xon) is sent first. This code is used to turn on a paper reader.
- 4) When receiving from the serial port, a DC3 (0x13 Xoff) is sent after the last % is received. This code is used to turn off a paper tape reader.



VS-Series

Note that the Setting 14 selection XON/XOFF is similar to the “DC CODES” selection. Both of these settings use the DC1/DC3 XON/XOFF codes to start/stop the sender when data is received too fast. When DC CODES is selected for Setting 14 (synchronization), serial port #1 will transmit an XON (DC1) if a character has not been received for five (5) seconds.

XMODEM may also be selected in setting 14. It is a receiver–driven communications protocol that sends data in blocks of 128 bytes. Setting **synchronization** to XMODEM gives your RS-232 communication an added level of reliability because each block is checked for integrity. If the receiver determines that the most recently sent block is in error, it will request that the sender try to send the block again.

In order to use XMODEM, parity must be none, and RS-232 data bits must be set to 8. Also, the computer that is sending the data must be equipped with a communications package that supports the XMODEM protocol. It must be set to XMODEM to operate.

This version of XMODEM supports **checksum** verification only. Also, 8K bytes of memory must be available before using XMODEM with DNC. XMODEM must use 8 data bits and no parity.

#### WARNING!

**One of the biggest causes of electronic damage is a lack of a good earth ground on both the CNC and the computer that is connected by RS-232. A ground fault condition (i.e., a lack of good ground on both) will damage the CNC or the computer, or both.**

Port #2 on the side cabinet is dedicated to auxiliary axes communication.

### *DIRECT NUMERICAL CONTROL (DNC)*

As a standard feature, this machine is shipped with a DNC capability. With DNC, there is no limit to the size of your CNC programs. The programs are directly executed by the control as they are sent over the RS-232 interface. Note, that this is the first serial port or the top connector. Do not confuse DNC with RS-232 uploading and downloading.

If you wish to use DNC, it is enabled by Parameter 57 and Setting 55.

---

**NOTE:** Floppy disk DNC is selected by entering the floppy disk file name and pressing MDI a second time when already in MDI mode. Do not press MDI **three** consecutive times or a “DISK ABORT” will result.

When enabled, DNC is selected by pressing MDI a second time when already in MDI. DNC mode will not be enabled unless there is a minimum of 8k bytes of user memory available. When DNC is selected, the PROGRAM DISPLAY will show:

WAITING FOR DNC...

This means that no DNC data has been received yet and you may begin sending data. You must start sending the program to the control before the CYCLE START button can be pushed. After the beginning of the program is seen by the control, the display will show part of the program and a message at the bottom, left of the display will show DNC PROG FOUND. After the program is found, you may push CYCLE START just like running any other program from Memory.

If you try to press CYCLE START before receiving a program, you will get the message: NO DNC PROG YET. The reason for not allowing the command of CYCLE START before receiving the DNC program is for safety. If the operation is allowed to start from a remote location, the operator may not be present to ensure that the machine is operating safely.

While a DNC program is executing, you are not allowed to change modes. You must first press RESET to stop the program.

When the end of the DNC program is received, the message DNC END FOUND is displayed. When the DNC program is finished running, the PROGRAM DISPLAY will show the last few lines of the program. You must press RESET or exit the DNC mode before you can run any other programs. If you try to press CYCLE START before RESET of the previous DNC, you will get the message: RESET FIRST.

DNC supports Dripmode. The control will execute one block at a time from the RS-232 port. Each block entered will be executed immediately with no block lookahead buffering. The exception is that Cutter Compensation requires three blocks of motion commands to be buffered prior to a compensated block being executed.

The program must begin with a % just like any other program sent over RS-232 and the program must end with a %. The data rate selected for the RS-232 port by settings must be fast enough to keep up with the rate of block execution of your program. If the data rate is too slow, the tool may be stopped in a cut when you might otherwise expect continuous cutter motion. The highest standard RS-232 data rate available is 115,200 bits per second.

It is recommended that DNC be run with Xmodem or parity selected because an error in transmission will then be detected and will stop operation of the DNC program without crashing. The settings page is used to select parity. The recommended RS-232 settings for DNC are:

SETTINGS:	11	BAUDE RATE SELECT:	19200
	12	PARITY SELECT:	NONE
	13	STOP BITS:	1
	14	SYNCHRONIZATION:	XMODEM
	37	RS-232 DATA BITS:	8

Full duplex communication during DNC is possible by using the G102 command or DPRNT to output axes coordinates back to the controlling computer. When DNC is running, BACKGROUND EDIT is not available.

To run DNC in graphics, you must select DNC first , then go to graphics display and send your program to the CNC.

### FLOPPY DISK OPERATION



All files must be on MS-DOS formatted 1.44M floppy disks and must reside in the root directory. Parameter 209 DISK ENABLE must be 1.

---

**NOTE:** In order to enable the floppy disk drive, an unlock code must be entered. If necessary contact the Service Department for more information.

---

**NOTE:** Use an empty (containing no other files) floppy disk for faster operation. All programs must begin with a line containing a single % and must end with a line containing a single %. All programs saved by the control will have these % symbols.

**Programs** may be loaded and saved from the floppy disk. To **LOAD** a program, press the LIST PROG key with PRGM selected. Enter the floppy disk file name and press F3 and the control will receive all main and sub programs until it receives a % sign indicating end of input. Please note that when using "ALL", all your programs must have an address **Oxxxx** to be filed. An ASCII EOF character (code 04) will also terminate input. The colon character (:) may be used in place of the O for a program name, but it is always displayed as O.

When loading floppy disk data, there is a status message at the bottom of the screen. It will update as follows:

LOADING Onnnnn	When program name is received.
DISK DONE	When complete and last % is received.
DISK ABORT	When anything causes abnormal stop.

To **SAVE** a program to floppy disk, press the LIST PROG key with PRGM selected. Enter the floppy disk file name, use the cursor as above to select the program, and press the F2 key. You can select "ALL" to send all of the programs in memory.

---

**NOTE:** Setting 23 must be off.

**Parameters, Settings, Macro Variables, and Offsets** may also be sent individually to the floppy disk by selecting the "LIST PROG" mode, entering the floppy disk file name, selecting the desired display screen (PARAM, SETNG, OFFSET, or the Macro Variables page of CRNT CMDS), and pressing the F2 key. They can be received by pressing the F3 key.

If an EOB (semicolon) is loaded, it will cause a blank line in the program.

The format of data sent and received for settings, offsets, and parameters is the following:

```
%  
N0 Vnnnnnn  
N1 Vnnnnnn  
N2 Vnnnnnn  
. . .  
%  
%
```

The following table shows the use of the N numbers in the output file when offsets are saved to RS-232 or disk.

Tool length:

Tool/function	length	wear	diameter	wear
1	001	201	401	601
2	002	202	402	602
3	003	203	403	603
:	:	:	:	:
200	200	400	600	800

The nnn numbers for work offsets are as follows:

Offset/axis	X	Y	Z	A	B
G52	801	802	803	804	805
G54	806	807	808	809	810
G55	811	812	813	814	815
:	:	:	:	:	:
G59	831	832	833	834	835
G110	836	837	838	839	840
G111	841	842	843	844	845
:	:	:	:	:	:
G128	926	927	928	929	930
G129	931	932	933	934	935

The nnnn numbers for coolant spigot positions are as follows:

1	936
2	937
3	938
:	:
200	1135

The **N** number is the data number and **V** is the value. **N0** is a CRC code that is computed by the control prior to sending the data. The **N0** value is mandatory with parameters but is optional with settings and offsets. If you make a change to some saved data value and leave the old CRC, you will get an alarm when you load that data. With settings and offsets, you should delete the **N0** line if you make changes to the saved data.

Data that is received garbled is usually converted into a comment and stored into your program while an alarm is generated. Errors generating an alarm may also stop the receive operation.

To get a **DIRECTORY LISTING**, select the PRGM/LIST PROG mode, and press F4. This will generate a Disk Directory Listing that will be saved in program 0xxxx where xxxx is defined in parameter 227. The default value is 8999.

On the List Prog page, type "DEL <filename>" where <filename> is the name of a file on the floppy disk. Press write. The message "DISK DELETE" will appear, and the file will be deleted from the floppy disk.

At the end of a save or load function, the bottom left corner of the display will show either: "DISK DONE" for normal completion or "DISK ABORT" if any errors cause it to stop. The actual errors are listed in the ALARM display.



VS-Series

## *PRINTING FROM HAAS MACHINES*

**Important! You must have a serial cable and serial printer.**

### **Setting up the Printer**

- Plug in the printer power cord.
- Plug the printer into the control in the RS232 port.
- Turn on the printer.
- Make sure the following settings are correct.

---

**Note:** Write down the old settings in case you need change them back later.

### **The Machine Settings for Printing**

- |               |                  |
|---------------|------------------|
| • Setting #11 | 9600 (Baud)      |
| • Setting #12 | None (Parity)    |
| • Setting #13 | 1 (Stop Bit)     |
| • Setting #14 | Xon/Xoff (synch) |
| • Setting #37 | 8 (Data Bits)    |

### **Printing Rules**

- The program must be in memory. A program cannot be printed in MDI.
- A program cannot be printed while it is running.

### **Printing a Program**

- Make sure the printer is turned on.
- Make sure there is paper in the printer.
- Go to "List Programs".
- Highlight the program to be printed.
- Push "Send RS232".
- To get the last page of the program out, push the paper feed button on the printer.

## 2.17 DRY RUN OPERATION

The DRY RUN function is used to check a program quickly without actually cutting parts. DRY RUN is selected by pressing the DRY RUN button while in MEM or MDI mode. When in DRY RUN, all rapids and feeds are run at the DRY RUN speed selected from the JOG speed buttons. The bottom of the screen will display the rate as 100, 10, 1.0 or 0.1 inches per minute.

DRY RUN cannot be turned on while a program is running. It can only be turned on or off when a program has completely finished or is reset. The first push of the DRY RUN button turns on this function and the second push will turn it off again. DRY RUN will still make all of the requested tool changes. The speed used in DRY RUN can be changed at any time and the operator can then check that the motions that are programmed are exactly what were intended. Note that Graphics mode is just as useful and may be even safer since it does not begin moving the machine before the program is checked.

## 2.18 DISPLAYS

You can select any of the following displays using the eight DISPLAY select buttons:

<b>PRGRM / CONVRS</b>	To show or edit the program selected OR to select conversational mode (QuickCode)
<b>POSIT</b>	To show the axes positions.
<b>OFSET</b>	To show or enter working offsets.
<b>CURNT COMDS</b>	To show current commands and times.
<b>ALARM / MESGS</b>	To show alarms and user messages.
<b>PARAM / DGNOS</b>	To show parameters and diagnostic data.
<b>SETNG / GRAPH</b>	To show or enter settings OR to select graphics simulation mode.
<b>HELP / CALC</b>	To show the help data and calculator.

In addition, when a program is running, you may press LIST PROG to select a list of the programs in memory. This is helpful in determining which programs can be edited in BACKGROUND EDIT, which is selected from the PROGRAM display.



## MESSAGES

The screen will ALWAYS show some of the current conditions selected in the control. These are fixed status displays that describe the condition of the machine. The following conditions are displayed on the screen:

The selected display in the top left corner;	
The selected mode in parentheses;	
The selected program in the top right corner;	
The most recent line number in the top right corner;	
Up to 18 lines of variable display data;	
Any of the following conditions that apply:	
ALARM	Blinking in lower right corner when alarm occurs.
BLKDEL	BLOCK DELETE is turned on.
A UNCLAMPED	The A-axis brake is unclamped
B UNCLAMPED	The B-axis brake is unclamped.
C CLAMPED	The C-axis brake is clamped.
C UNCLAMPED	The C-axis brake is unclamped.
CS TO CONTINUE	Indicates program will continue after pressing CYCLE START.
CYCLE DOOR	On CE machines the door must be cycled before continuing.
CYCLE SFTY	ON CE machines without a door, the safety switch must be pressed before continuing.
DOOR HOLD	An open door has stopped the program.
DOOR IS CLOSED	The message DOOR IS CLOSED is displayed if the door is closed when the Tool Release button on the control is pressed. This is to remind the operator to open the door and hold the tool so it does not fall out of the spindle.
DR CLOS TIMEOUT	Displayed if automatic pallet station door does not close in the time allowed by parameter 251.
DR OPEN TIMEOUT	Displayed if automatic pallet station door does not open in the time allowed by parameter 251.
DR MTR TIMEOUT	Displayed if automatic pallet station door does not begin moving to open in the time allowed by parameter 252.
DRYRUN OVERRIDE	Displayed in dry run when a Feed Rate or Rapid Override key is pressed.
DRY RUN SPEEDnnnn	When DRY RUN is selected
DRYRUN	DRY RUN is selected.
DWELL	When a G04 is being performed.
FEED	When a feed motion is in progress.
FEED HOLD	FEED HOLD is active.
FEED nnn%	Feed rate override is active.
FILE DELETE	Deleting a file from floppy drive.
FUNCTION ABORT	Requested function has been aborted.
INDEXING	In index mode with X, Y or Z axis in motion.
INSERT	Selected text now being inserted.
INSERT TOOL <nn>	Tool change is commanded by M06.
JOG INDEX ON	When in INDEX JOG mode
JOG LOCK ON	When JOG LOCK is pressed.
JOGGING x AXIS HANDLE	When HANDLE JOG is pressed.
LOADING...	Reading programs or data from RS-232.
LOW AIR PR	Highlighted, indicates low air pressure.
LOW COOLANT	Low level in coolant tank.
NEW PROGRAM	A new program may be entered.
NO FLOPPY FOUND	Floppy drive is not present.
OFF IN nnSEC	Blinking when AUTO OFF key is pressed.
OPTSTP	OPTIONAL STOP is turned on.

**OVERWRITE xxxxxxxx.xxx (Y/N)?**

Do you want to overwrite the file, yes or no ?

PAL DOOR HOLD	Displayed during servo axis rotation, if the operator door or the pallet changer access door is opened.
PLEASE WAIT	Wait while control finishes processing edit command.
PROGRAM END	The program being run has completed.
RAPID %	Rapid override is active.
RUNNING	When a program is running.
SEARCHING...	Searching program for requested text or G code.
SENDING OFFSET	Sending offsets via RS-232.
SENDING PARS	Sending parameters via RS-232.
SENDING SETTING	Sending settings via RS-232.
SENDING VARS	Sending variables via RS-232.
SENDING...	Floppy write or RS-232 output is in process.
SETTING 104 ON	SETTING 104 and HANDLE CONTROL FEED RATE or HANDLE CONTROL SPINDLE pressed.
SFTY HOLD	The safety switch was released, and it has stopped the program.
SHIFT	When the SHIFT key is pressed.
SINGBK	SINGLE BLOCK is turned on.
SINGBK STOP	When a program is stopped in SINGLE BLOCK.
SORTING	The control is allocating resources and will soon start copying programs.
SPIND nnn%	Spindle speed override is active.
T UNCLAMPED	The tool turret is unclamped.
TOOL CLAMP	The tool is in the spindle.
TOOL UNCLAMP	The tool has been unclamped.
TOOL UNCLP	Highlighted when the tool is unclamped.
TURRET IN	The tool changer is in position for a tool change.
TURRET OUT	The tool changer is out of position for a tool change.
XY-MANJOG	Displays appropriate axis.
XYZA-MIR	These axes are set to mirror image.

The following error messages are received when the wrong button is pressed.



VS-Series

ALARM ON	Cannot start an operation until alarms are reset.
ALTER	The selected text can now be altered.
AUX AXIS BUSY	One or more auxiliary axes are busy in an operation.
BAD CODE	Code entered is not understood.
BAD FILE NAME	Not a valid file name.
BAD NAME	Name entered is not <b>Onnnnn</b> .
BLOCK TOO LONG	Block being edited would be too long.
CAN NOT COPY	The selected program cannot be copied.
CAN'T RENAME	The selected program cannot be renamed.
CAN'T UNDO!	The last function cannot be undone.
CNVEYR DISABLED	Chip conveyor has been disabled by parameters.
DIR FULL	Maximum number of programs exceeded.
DIR NOT FOUND	Directory of floppy disk not found.
DISK NOT IN DRV	No disk in floppy disk drive.
DISK NOT READABLE	Floppy disk cannot be read by control.
DISK WRT PROTECT	Cannot save to floppy disk, it is write protected.
DISPLAYS OFF	Indicates that M76 was used to turn off displays.
EMPTY PROG	No program found between the % and %.
END OF PROG	The program being run has completed.
EXIT BG EDIT	Must exit Background Edit mode to perform this function.
FILE NOT FOUND	The file was not found.
FLPY NOT ENBLED	Floppy disk drive has not been enabled by parameters.
FUNCTION LOCKED	Function attempted is locked from settings.
GUARDED CODE	Cannot remove <b>Onnnnn</b> at start of program.
INSUF DSK SPACE	Insufficient disk space to save the selected file.
INVALID AXIS	Selected axis is invalid.
INVALID NUMBER	Number entered is invalid.
MACHINE LOCKED	Front panel has been locked by Setting 6.
MACRO LOCKED	Macros 9000 to 9099 are locked by Setting 120.
MEMORY FULL	Memory space is full.
MEMORY LOCKED	Memory locked by Setting 108.
NO DISK FOUND	Cannot find the floppy disk drive.
NO DNC PROG YET	Attempted to start program before it was completely received.
NO INPUT	Cannot alter until something has been entered.
NO NAME ENTRY	A file name has not been entered.
NO ZERO A	Cannot run machine until search for zero is complete on A axis.
NO ZERO X	Cannot run machine until search for zero is complete on X axis.
NO ZERO Y	Cannot run machine until search for zero is complete on Y axis.
NO ZERO Z	Cannot run machine until search for zero is complete on Z axis.
NOT AVAILABLE	Function requested is not available at this time.
NOT IN DRYRUN	The function requested applies to DRY RUN, but control is not presently in that mode.
NOT FOUND	Item not found during search in editor.
MACHINE LOCKED	Front panel has been locked by setting.
MACRO LOCKED	Macros 9000 to 9099 are locked by setting.
MEMORY FULL	Memory space is full.
MEMORY LOCKED	Memory lock is set in settings.
NEW PROGRAM	A new program may be entered.
NO DISK FOUND	Cannot find the floppy disk drive.
NO DNC PROG YET	Attempted to start program before it was completely received.



VS-Series

NO INPUT	Cannot alter until something has been entered.
NO NAME ENTRY	No file name has been entered.
NO PROG YET	Cannot press Cycle Start until a program is received.
NO ZERO A	Cannot run machine until search for zero is complete on A axis.
NO ZERO X	Cannot run machine until search for zero is complete on X axis.
NO ZERO Y	Cannot run machine until search for zero is complete on Y axis.
NO ZERO Z	Cannot run machine until search for zero is complete on Z axis.
NOT AVAILABLE	Function requested is not available at this time.
NOT FOUND	Item not found during search in editor.
NOT IN DRYRUN	The function requested applies to DRY RUN, but control is not presently in that mode.
ONE BIT ONLY	Only 0 or 1 is accepted to alter a parameter switch.
PLEASE WAIT	Wait until spindle is stopped.
PRESS +a OR -a (a = X, Y or Z)	Jog key for another axis has been pressed.
PROG EXISTS	Cannot rename to an existing program.
PROG NOT FOUND	Requested program not in memory.
PROG READY	Program has been received and is ready to run.
PROGRAM IN USE	Program is already in use.
RANGE ERROR	Data being entered is outside of the valid range.
RESET FIRST	Must press RESET before performing this function.
RS-232 ABORT	RS-232 was aborted by operator action.
RS-232 DONE	RS-232 operation is complete.
RS-232 ERROR	RS-232 error (shown in alarms).
SAFETY SWITCH	The safety switch is not depressed; some functions not allowed.
SEL HI GEAR	High gear selected in program.
SEL LOW GEAR	Low gear selected in program.
SERVO IS OFF	When servos are <b>off</b> , you cannot start a program.
SERVO IS ON!	Parameter change was attempted with servo on. <b>This is dangerous!</b>
SHUTTLE IN	Tool changer is in position for a tool change.
SHUTTLE NOT HOME	The control is preventing the operator from zeroing the machine. Move the tool shuttle to the "Standby Position"; this is the only safe place for the shuttle to be when zeroing the machine.
SPEED COMD	A spindle speed must be commanded.
SPINDLE IN USE	Spindle is being controlled by program - manual controls not available at this time.
SPINDLE LOCKED	Spindle is locked in place.
SPINDLE ORI	Spindle is being oriented.
SPINDLE STOP	Spindle is not turning.
STRING TOO LONG	The text being entered is too long.
SYSTEM ERROR	Call your dealer.
TOOL CH LOCKED	Tool changer has been disabled by parameter 57 bit 1.
TOOL OVERLOAD	Cutting tool is overloaded.
WAIT OR RESET	Cannot perform requested function until program finishes or RESET is pressed.
WAIT...	Requested function is being performed.
WAITING...	Waiting for RS-232 input.
WRONG MODE	Function requested is available only in another mode.

And the following responses are received only in graphics mode:



VS-Series

CIRCULAR	A circular motion is being performed.
COOLANT OFF	Coolant pump is off.
COOLANT ON	Coolant pump is on.
JOG COMD	An axis jog has been commanded.
LINEAR	A linear motion is being performed.
M30 FOUND	End of program found and execution stopped.
NO ZOOM IN 3D	Zoom is not allowed in 3D graphics mode.
RAPID	A rapid motion is being performed.
RIGID TAP	Rigid tapping is being performed.
SPINDLE CCW	Spindle is turning counterclockwise.
SPINDLE CW	Spindle is turning clockwise.
SPINDLE HIGH	Spindle is in high gear.
SPINDLE LOW	Spindle is in low gear.

In addition, the screen display can show one of the following eight types of data in the 18 lines of variable display:

#### **Program Displays:**

The PROGRAM DISPLAY is used to show your program while in either MEM, EDIT, or MDI modes.

#### **Position Display:**

The POSITION DISPLAY is used to select the **X**, **Y**, **Z**, or **A** axes positions in any of several coordinate systems. The PAGE UP and PAGE DOWN keys select between these.

#### **Offsets Display:**

The OFFSETS DISPLAY is used to enter and display tool length offsets, tool radius offsets, coolant positions, and work offsets. The PAGE UP and PAGE DOWN keys will scroll between these.

#### **Current Commands Display:**

The CURRENT COMMANDS DISPLAY is used to display the Program Command Check, the Current Commands, Running Timers, Tool Life Timers, Tool Load Monitor, and Axis Load Monitor. The PAGE UP and PAGE DOWN keys select between these.

#### **Alarms / Messages Displays:**

The ALARMS/MESSAGES DISPLAY is used to display alarms and to enter and display user messages. The second push of the ALARM button will select the messages display. Entering the Alarms display will show the full text of an alarm when the alarm message is flashing. There are three types of Alarms screens. The first shows the currently active alarms. Pressing the RIGHT ARROW key switches to the Alarm History screen. Pressing RIGHT ARROW again switches to the alarm viewer screen. This screen simply shows one alarm at a time with its description (if any). The default will be the last alarm in the alarm history. The user can then scroll through all the alarms by pressing the UP and DOWN ARROW keys. Also, the user can enter an alarm number and press WRITE and it will be displayed. Pressing RIGHT ARROW again returns the user to the home page of the current alarms screen. Pressing PAGE DOWN will display a page for user messages and notes.

#### **Parameters / Diagnostics Displays:**

The PARAMETERS DISPLAY shows all of the machine dependent control parameters and the Diagnostic data. The second push of the PARAM / DGNOS button will select the diagnostic display. The PAGE UP, PAGE DOWN, up and down cursor keys, and the jog handle can be used to scroll through the parameter display screens in the control. The left and right cursor keys are used to scroll through the bits in a single parameter.

#### **Settings / Graphics Displays:**

The SETTINGS DISPLAY is used to display and change user controlled parameters. The second push of the SETNG / GRAPH button will select the Graphics display.

### Help / Calculator Displays:

The HELP DISPLAY shows a mini-manual with a directory of available help information. Each alphabet button will select a different topic within the HELP display. The second push of the HELP button will select the Calculator display. The PAGE UP and PAGE DOWN buttons will select different calculator functions.

## PROGRAM DISPLAYS

The PROGRAM DISPLAY is used to show a program being edited in EDIT mode or a program being run in MEM. In MEM mode, there is also a PROGRAM REVIEW display available.

The PROGRAM DISPLAY uses 18 lines of the text display area of the display to show the command blocks of a CNC program. The display is 40 positions wide and blocks that are longer than 40 positions are continued on the next line of the display.

The PROGRAM REVIEW function is available whenever a program is being run. This allows you to review the program that is running. This is selected by pressing F4 while in MEM mode and PROGRAM DISPLAY. The screen is changed to an 80 column display with the normal MEM display on the left and PROGRAM REVIEW on the right. The CURSOR and PAGE UP and PAGE DOWN keys can be used to change the display on the right to a different part of the program. The display on the left will show the progress of the running program. To exit PROGRAM REVIEW, select any other display.

While you are running a program, the BACKGROUND EDIT function is available as a standard feature. BACKGROUND EDIT allows you to edit any named program in memory while any program is being run in memory. BACKGROUND EDIT is selected from MEM mode in PROG display by entering **Onnnnn** with the program number and pressing F4. The display will change to the selected program while still running the first program. BACKGROUND EDIT is enabled by parameters in your machine.

## POSITION DISPLAYS

The following are the five position displays in this control:

### Home Page

This display shows the four displays simultaneously in small characters. The other displays are shown in large characters. The PAGE UP and PAGE DOWN keys will change displays. The coordinates displayed (operator, work, machine, or distance to go) are selected using the cursor **up** and **down** keys. The last display selected will be shown in Current Comands, Graphics and Calculator displays when they are selected. In this display, any axis that is at the zero position will be highlighted. In the other four displays, the selected axis will be blinking.

### Operator Display

This display is for the operator/setup person to use as desired, and is not used by the control for any positioning functions. In JOG mode, and with the axis selected, pressing the ORIGIN button will set the position to zero. Another way to set the position to zero is to enter the desired axis letter (e.g. Y) and then press Origin. This display will then show position relative to the selected zero position. To enter a value into the position display, enter the letter followed by the position (e.g. X -2.) and press Origin.

### Work Display

This display shows how far away the tool is from the **X**, **Y** and **Z** zeros of the programmed part. On power-up, it will display the value in work offset G54 automatically. It can only be changed by G54 through G59, G110 through G129, or by a G92 command. The machine uses this coordinate system to run the part.



### Machine Display

This display is the machine coordinate system that is automatically set on power-up and the first ZERO RET. It cannot be changed by the operator or any work coordinate systems, and will always show the distance from machine zero. It can be used by a non-modal G53 command.

### Distance To Go

This display is an incremental display that shows the travel distance remaining before the axes stop. When in the ZERO RET mode, this display shows a diagnostic value. When in JOG mode, this display shows the total distance jogged. In rigid tapping, this number decreases to zero at the bottom of the hole and then increases again as the reverse stroke occurs. The Distance to go can also be zeroed by switching operation modes and then back (e.g. press MDI followed by Handle Jog).

## OFFSET DISPLAY

Tool length and part zero offsets are displayed and can be entered in the Tool Offsets display. The same offset number for both Z-axis offset (H1) and tool diameter compensation (D1) can be used because the offset contains separate values for each. Work offsets can also be specified for the fifth axis **B** address, in the work offset display. If the fifth axis is enabled (Setting 78), then additional data fields are made available for the fifth axis.

PAGE DOWN in the OFSET display will go through all possible tool offsets and then change to the work zero offsets. PAGE UP will go directly to work zero offsets. The jog handle can also be used to scroll through the offsets. Work zero offsets may be entered and displayed from this page or using the PART ZERO SET key. Refer to the "Setup Procedures" section for more information on using this key.

When entering offsets, pressing F1 will set the value to your entry. Pressing WRITE will cause the new value to be added to the old value. This allows small adjustments to be made to the offsets. Note that entering a negative entry and pressing WRITE will decrease the value of the offset. F2 will set the negative of the entered value into the offsets.

There is a geometry and wear value with each offset and these are added together by the control during operation. The initial value is entered into the geometry column by the setup person. During operation, the operator makes minor wear changes in the wear column. This method allows the operator to see actual tool wear by limiting it to the wear column. The geometry values can also be entered automatically by using the TOOL OFSET MESUR key during setup procedures. Note that this automatic offset measurement works with G43 only. Refer to the "Setup Procedures" section for more information on using the TOOL OFSET MESUR key.

The function of the offset display page has been modified slightly to accommodate multiple axes. Only the axes that are enabled are displayed on the work offset display. For example, if the fourth axis is enabled (Setting 30), and the fifth axis is not, then the display will show X,Y,Z,A. In this case, all axes can be displayed on the entire display. If only the fifth axis is enabled, then the display will show X,Y,Z,B. Since the fourth axis is disabled, there is no need to display it.

Setting 15, "H & T Code Agreement", may be used to force the spindle tool number and offset number to be equal; otherwise an alarm will occur. This is the preferred setting as it avoids crashes.

When Surface Feet per Minute (SFM) is being calculated, if there is an "Actual Diameter" column for the selected tool. If there is no entry, then SFM is calculated using the regular tool geometry diameter or radius.

Tool diameter may be entered as either radius or diameter. Setting 40 is used to select between these. The value used for cutter compensation is the sum of the geometry and wear values. The number of flutes on each tool can be entered. This is used in the calculation of Chip Load. The default is two, the number of flutes for each tool will also be set to two when the ORIGIN key is pressed while this screen is displayed.

Cutter compensation is controlled by G41 and G42 and the selected tool diameter (D value). Positive values for cutter compensation work normally. Negative values for cutter compensation cause the opposite side cutter compensation to be used. This means that G41 with a negative number will be the same as G42 with the same but positive number.

Offsets may be sent and received with the RS-232 port. Refer to the "Data Input / Output" section for a description of how to do this.

When the COOLANT SPGT of Parameter 57 is enabled, the CLNT POS column of the offset display is accessible. The spigot can be positioned when an M08 is encountered in a program. The current **H** code is used by the M08 to determine where to position the coolant spigot. If offset #5 (H05) has **10** entered under the CLNT POS column, the spigot will be moved to position 10. When the cursor is positioned on the CLNT POS column, the current spigot position will be shown at the bottom of the display. If the spigot has not been zeroed, the position will be blank.

The permissible values that can be entered into this column are controlled by Parameter 206, SPIGOT POSITIONS. Entering a value of zero (0) indicates that the spigot will not be moved when an M08 code sequence is encountered.

The following code sequence demonstrates how the spigot can be commanded:

---

**NOTE:** That the tool length and radius offset numbers usually are the same as the tool number

O0001 (Sample coolant positioning)

T1 M06

G90 G54 G00 X\_ Y\_ S1400 M03

G43 H01 Z\_ M08

(Moves spigot to H1 position)

T2 M06

G90 G54 X\_ Y\_ S900 M03

G43 H02 Z\_ M08

(Moves spigot to H2 position)

G43 H42 Z.1M08

(Position to new location with a second offset H42)

(More than one offset number can be used for a tool if needed)

M30



M codes 34 and 35 can also be used to control the spigot position.

### CURRENT COMMANDS DISPLAY

The following are the seven current command pages in this control:

- Program Command Check (Home Page),**
- Current Display Command,**
- Operation Timers,**
- Macro Variables,**
- Periodic Maintenance**
- Tool Life Timers,**
- and Tool and Axis Load Monitor**

The PAGE UP and PAGE DOWN keys are used to select among these displays.

#### **Program Command Check Display**

This display, which is the Home Page for the Current Commands Display, shows a current overview of the important commands. It shows the programmed spindle speed (**Snnnnn**), the spindle speed commanded to the spindle drive (**CMDnnnnn**), and if Parameter 278 bit ACT DISPLAY is set to 1, the actual encoder spindle speed (**ACTnnnnn**). In addition, this display shows the CW, CCW, or stopped command being sent to the spindle and the current gear position.

This display also shows the position of the axes. The coordinates displayed (operator, work, machine, or distance to go) are selected using the cursor **up** and **down** keys.

If the spigot is enabled, the current spigot position will be displayed.

#### **Current display command**

This display shows all of the alphabetical address codes (i.e. G, M, S, H, D) and their current value. These values may not be changed in this display. The default value is shown for the address codes that are not being used in the current program.

#### **Macro Variables Display**

This display shows a list of the macro variables and their present values. As the control interprets a program, the variable changes are displayed on this page and the results can be viewed. The variables may be modified in this display. For more information on this display, refer to the "Macros" section of this manual.

#### **Operation Timers Display**

This display shows the current power-on time, cycle start time, and the feed time. These times may be reset to zero by using the cursor **up** and **down** keys to highlight the desired title and pressing the ORIGIN button.

Listed below these times are two M30 counters that are used for counting completed parts. They may be set to zero independently to provide for the number of parts per shift and total parts. Both counters are increased when an M30 is operated.

#### **Tool Life Display**

This display shows the time the tool is in feed (Feed-Time), the time the tool is in the spindle (Total-Time), and the number of times the tool has been selected (Usage). This information can be used to assist in predicting tool life. The values in this display can be reset to zero using the Cursor and ORIGIN buttons. This is done by putting the cursor on the title line, and pressing ORIGIN to zero all of the data in that column. The maximum value is 32767, once this value is reached, and not reset, the control will start back at zero.



VS-Series

This display may also be used to generate an alarm when a tool has been used a specific number of times. The last column is labeled "Alarm", and if the number for a tool is not zero, an alarm will be generated when that count is reached. This number can be changed by the operator. Alarm 362 is generated when the count is reached, and may be cleared with RESET.

The tool that is currently in the spindle is highlighted.

### Tool Load Monitor and Display

With the tool load display, the operator can enter the maximum load that is expected for each tool, and when this load is exceeded in a feed, a certain action will be taken. This display provides for the entry of this alarm point and also displays the largest load that tool has seen in any previous feed.

---

**NOTE:** Surface Feet per Minute (SFM) and Chip Load are displayed on the Current Commands page. Note that Setting 40 selects radius or diameter for the tool geometry. SFM is displayed as fpm (feet per minute) or mpm (meters per minute), depending on setting 9. This is displayed in inches (typically a few thousandths) or millimeters (typically a few hundredths).

The tool load monitor function operates whenever the machine is in a feed operation (G01, G02, or G03). The values entered into the tool load display are checked against the actual spindle motor load. If the limit is exceeded, the tool overload action specified in Setting 84 (alarm, feedhold, beep, or Autofeed) will be taken. If "alarm" is selected and the limit is exceeded, Alarm 174, "Tool Load Exceeded", will be generated. This alarm will stop the axis motors and the spindle motor, turn off the coolant, and disable the servos.

If, during a feed the load exceeds the tool limit and the AUTOFEED feature, is selected, it will automatically override the feed rate (reduce it) down to the percentage specified by parameter 301 (i.e. 1%) at the rate specified by parameter 300 (i.e. 0% per second). If the tool load later falls below 95% of the tool load limit percentage, the AUTOFEED feature will automatically override the feed rate (increase it) back to the feed rate that was in effect at the start of the feed at the rate specified by parameter 299 (i.e. 0% per second). These adjustments will be made in 0.1 second increments.

### Axis Load Monitor

Axis load is 100% to represent the maximum continuous load. Up to 250% can be shown, and above 100% for an extended period of time, can lead to an axis overload alarm.

### Maintenance

Accessed by pressing PAGE UP or PAGE DOWN which allows the operator to activate and deactivate a series of checks (see Maintenance section).

## ALARMS / MESSAGES DISPLAY

The **ALARMS DISPLAY** can be selected at any time by pressing the ALARM / MESGS button. When there are no alarms, the display will show NO ALARM. If there are any alarms, they will be listed with the most recent alarm at the bottom of the list. There are three types of Alarms screens. The first shows the currently active alarms. Pressing the RIGHT ARROW key switches to the Alarm History screen. Pressing RIGHT ARROW again switches to the alarm viewer screen. This screen simply shows one alarm at a time with its description (if any). The default will be the last alarm in the alarm history. The user can then scroll through all the alarms by pressing the UP and DOWN ARROW keys. Also, the user can enter an alarm number and press WRITE and it will be displayed. Pressing RIGHT ARROW again returns the user to the home page of the current alarms screen. Pressing PAGE DOWN will display a page for user messages and notes.

The CURSOR and PAGE UP and PAGE DOWN buttons can be used to move through a large number of alarms. The CURSOR **right** and **left** buttons can be used to turn on and off the ALARM history display.



The **MESSAGE DISPLAY** can be selected at any time by pressing the ALARM / MESGS button a second time. This is an operator message display and has no other effect on operation of the control. Any message can be typed into the message display and called up later.

You may leave an electronic note to yourself or anyone else by using this feature. The note may be for the operator to change tools after running a number of parts or it may be a diary for machine maintenance intervals that are performed. Data is automatically stored and maintained even in a power off state. The message display page will come up during power up if there are no alarms present.

To enter messages, press the ALARM / MESGS button twice. You may now enter data by simply typing directly onto the screen. The cancel and space keys can be used to remove existing messages. The DELETE button can be used to remove an entire line.

#### *PARAMETER / DIAGNOSTIC DISPLAY*

The **PARAMETER DISPLAY** can be selected at any time by pressing the PARAM DGNOS button. Changes to parameters can be made when in any mode except when running a program. The CURSOR **up** and **down** buttons move to different parameters and the PAGE UP and PAGE DOWN buttons move through groups of parameters. Parameters 1, 15, 29, 43, and 57 are displayed as a single page of discrete flags. Selecting among the flags is done with the CURSOR **left** and **right** buttons. Parameters cannot be changed with the servos on. The EMERGENCY STOP button can be used to turn off the servos.

For machines with the fifth axis option, the parameter display organization has been modified to accommodate the extra axis parameters. Parameter numbers have remained the same as in a four-axis control. Additional parameters have been added for the fifth axis.

The parameters have been reorganized, so that logically-associated parameters are grouped together. These logical groupings are placed together into contiguous screens called pages. The most commonly changed parameters have been placed at the beginning of the page list.

The HOME key displays the first parameter page. Pressing the PAGE DOWN key will display the next page of parameters in the above list. The END key displays the last parameter page. Pressing the PAGE UP key will display the preceding page of parameters in the above list. All other features on the parameters display have remained the same. So, if you are unfamiliar with the new format of the parameters, you can still search by parameter number. Enter the number of the parameter you want to see or view and press the **up** or **down** arrow key. The page that the parameter is on will be displayed and the parameter being searched for will be highlighted.

The **DIAGNOSTIC DATA DISPLAY** can be selected at any time by pressing the PARAM DGNOS button a second time. There are two pages of diagnostic data and the PAGE UP and PAGE DOWN buttons are used to select between them. After this, the current run time and the number of tool changes is displayed.

A five-axis control has additional diagnostic data to be aware of: The first page of diagnostic data shows two discrete outputs that control the rotary axes brakes; they are labeled "Brake 4th Axis" and "Brake 5th Axis". The 4th axis is synonymous with the A-axis, and the 5th axis is synonymous with the B-axis. If the Air Brake Parameter for the axis is set to 1, the HAAS control unclamps the brake whenever motion is commanded to the rotary axis and sets the brake back to its previous state as soon as the motion stops. The brake is activated by a relay. The two discrete outputs show the state of the brake relays. If the output is high (1) the brake is unclamped. When the machine is first powered up the brake will be clamped, so the output will be low (0). When the brake is unclamped the message A UNCLMP or B UNCLMP is displayed in the message area near the bottom of the display.

The M11 and M13 codes command the brake to be unclamped. When these codes are in effect, the axis brake will remain unclamped even after motion has stopped.



VS-Series

The second page of diagnostic data shows the status of inputs from the motor interface board. Additional inputs for the expanded motor interface board are listed under "INPUTS4". These bits are monitored by the control to determine if the interface for the **B** axis is working correctly.

### **SETTING / GRAPHIC DISPLAY FUNCTION**

The **SETTINGS DISPLAY** can be selected at any time by pressing the SETNG / GRAPH button. When the settings are displayed, changes can be made to any of the settings. There are some special functions in the settings; refer to the "Settings" section for a more detailed description.

The **GRAPHICS FUNCTION** is a visual dry run of your part program without the need to move the axes and risk tool damage from programming errors. This function is far more powerful than using the DRY RUN mode because all of your work offsets, tool offsets, and travel limits can be checked before any attempt is made to move the machine. The risk of a crash during setup is greatly reduced.

To run a program in Graphics, you must be in either MEM or MDI mode.

After loading the program into memory, select MEM (or MDI) and press the SETNG/GRAFH key twice to select the Graphics Simulation mode. This function operates the same as if running a program on the machine except no physical machine action occurs.

The graphics screen is composed of the following areas:

#### **DISPLAY TITLE AREA**

The title area is on the top left line of the screen and indicates the display (GRAPHICS), the mode you are in (MEM or MDI), the program number, and the current program line being executed. It is the same as the top line of all displays.

#### **KEY HELP AREA**

The right side of the top line is the function key help area. Function keys that are currently available are displayed here with a brief description of their usage.

#### **LOCATOR WINDOW**

The lower right part of the screen has two functions: it can display the whole table area and indicate where the tool is currently located during simulation, or it can be used to display four lines of the program that is being executed. The F4 key can be used to toggle between these two modes.

#### **TOOL PATH WINDOW**

In the center of the display is a large window that represents a look down perspective of the X-Y axis. It displays tool paths during a graphics simulation of a CNC program. Rapid moves are displayed as coarse dotted lines, while feed motion is displayed as fine continuous lines. The rapid path can be disabled by Setting 4. The places where a drill can or canned cycle can be executed are marked with an X. The drill mark can be disabled by Setting 5.



VS-Series

The tool path window can be scaled. After running a program, you can scale any portion of the tool path by pressing F2 and then using the PAGE DOWN key and the ARROW keys to select the portion of the tool path that you want to see enlarged. During this process, a rectangle will appear within the TOOL PATH window and the Locator window indicating what the TOOL PATH window will represent when the zoom process is complete. The KEY HELP AREA will flash, indicating that the user is rescaling the view. The locator window always portrays the entire table with an outline of where the TOOL PATH window is zoomed to. The PAGE UP key unzooms the rectangle one step. After sizing or moving the rectangle, pressing the WRITE key will complete the zoom process and re-scale the TOOL PATH window. Pressing F2 and then the HOME key will expand the TOOL PATH window to cover the entire table. After the TOOL PATH window is re-scaled, the TOOL PATH window is cleared and you must rerun the program, or a portion of it, to see the tool path. The tool path is not retained in the control.

The scale and position of the TOOL PATH window is saved in Settings 65 through 68. Any scaling performed on the TOOL PATH window is retained. You can leave graphics to edit your program and when you return, your previous scaling is still in effect.

### Z AXIS WINDOW

A long window on the rightmost part of the screen shows the location of the Z-axis and indicates spindle movement. A horizontal line in the top part of this window represents the tool change position.

### Z AXIS PART ZERO LINE

This feature consists of a horizontal line displayed on the Z axis bar at the top-right corner of the graphics screen to indicate the position of the current Z axis work offset plus the length of the current tool. While a program is running, the shaded portion of the bar indicates the depth of Z axis motion. The user will now be able to watch the position of the tool tip relative to the Z axis part zero position as the program runs.

### CONTROL STATUS

The lower left portion of the screen displays control status. It is the same as the last four lines of all other displays.

### POSITION WINDOW

The location of all enabled axes can be viewed in this window. By default it is OFF. This window can be opened by pressing the F3 key. Additional presses of the F3 key or the up and down arrows will display the various position formats that the control keeps track of. This window also displays the current scale of the tool path window and the current simulated tool number. The value represented by the vertical dimension of the Tool Path window is labeled Y-SIZE. At power-on, this will be the full Y-axis table travel. When you zoom into a table area, this value will become smaller, indicating that you are viewing a smaller portion of the table.

To exit the Graphic mode, select any other display or mode. When you exit Graphics, the graphics image is lost and must be built again by running the program.

## HELP / CALCULATOR FUNCTION

The **HELP FUNCTION** is selected by pressing the HELP display button. This will bring a mini-manual up on the display. There are 26 topic areas selectable with the A-Z keys. This also includes a directory of the topics. The areas covered are:



VS-Series

A	START UP AND RUNNING	N	SET UP PROCEDURES
B	PROG. REVIEW / DNC / BGEDIT / POWER DOWN	O	OVERRIDES: FEED/SPIN/ COOLANT
C	G/M/S/T COMMAND CODES	P	PARAMETERS / DIAGNOSTICS
D	RETURN TO THIS DIRECTORY	Q	POSITION DISPLAYS
E	EDITING PROGRAMS	R	RECV / SEND PROGRAMS
F	SETTING PAGE	S	SAMPLE PROGRAM
G	SPECIAL G CODES	T	TOOL OFS/TOOL LIFE/LOAD
H	TROUBLE SHOOTING	U	GRAPHIC FUNCTION
I	MDI / MANUAL DATA INPUT	V	TOOL CHANGER
J	JOGGING / HANDLE FUNCTION	W	WORK COORDINATES
K	DISPLAY / KEYBOARD	X	CREATING PROGRAMS
L	ALARMS / MESSAGES	Y	SPECIAL FUNCTIONS
M	MAINTENANCE REQUIREMENTS	Z	ZERO RETURN

When the HELP display is selected, the alphanumeric keys are used to select one of the above topics; they are not displayed on the data entry line of the screen.

The **CALCULATOR FUNCTION** is selected by pressing the HELP key a second time. There are three calculator pages: Trig Help, Circular Interpolation Help, and Milling/Tapping Help. All of these have a simple calculator and an equation solver. Trig, Circular, and Milling Help are selected using the PAGE UP and PAGE DOWN keys. The **Fn** keys also allow data to be moved from other displays to/from the calculator.

All of the Calculator Help functions have a calculator for simple add, subtract, multiply, and divide operations. When one of these functions (Trig, Circular, or Milling) is selected, a calculator window appears in the upper left corner of the screen, and below it the possible operations (**LOAD + - \* /**). **LOAD** is initially highlighted, and the other options can be selected with the left and right cursor arrows. Numbers are entered by typing them in at the cursor in the lower left corner of the screen and pressing the **WRITE** key. When a number is entered and **LOAD** is selected, that number will be entered into the calculator window directly. When a number is entered when one of the other functions (**+ - \* /**) is selected, that calculation will be performed with the number just entered and any number that was already in the calculator window. The calculator will also accept a mathematical expression such as  $23*4-5.2+6/2$ . It will evaluate it (doing multiplication and division first) and place the result, 89.8 in this case, in the register (the window).

- F3** In EDIT and MDI modes the F3 key will copy the highlighted triangle/circular/milling/tapping value into the data entry line at the bottom of the screen. This is useful when you want to use a solution developed for a program input.

In the Calculator Help function, this button copies the value in the calculator window to the highlighted data entry for Trig, Circular or Milling/Tapping calculations.

- F4** In the Calculator Help function, this button uses the highlighted Trig, Circular or Milling/Tapping data value to load, add, subtract, multiply, or divide with the calculator.

### Trigonometry Help Function

The Trig Help page will help you solve a triangular problem. You enter the lengths and the angles of a triangle and when enough data has been entered, the control will solve for the triangle and display the rest of the values. Use the CURSOR **up** and **down** buttons to select the value to be entered with **WRITE**. For inputs that have more than one solution, entering the last data value a second time will cause the next possible solution to be displayed.



VS-Series

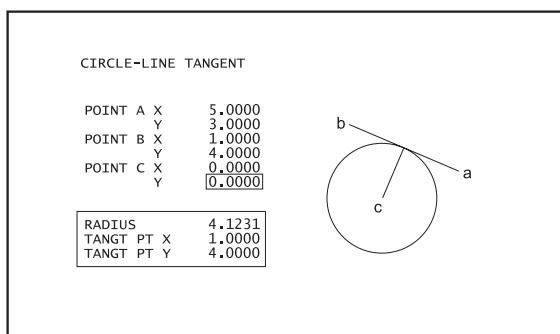
## Circular Interpolation Help

The Circular Help page will help you solve a circle problem. You enter the center, radius, angles, start and end points and when enough data has been entered, the control will solve for the circular motion and display the rest of the values. Use the CURSOR up and down buttons to select the value to be entered with WRITE. In addition, it will list the four ways that such a move could be programmed with a G02 or G03. Those four lines can be selected using the CURSOR up or down buttons, and the F3 button will import the highlighted line into a program you are editing.

For inputs that have more than one solution, entering the last data value a second time will cause the next possible solution to be displayed. The CW/CCW entry is changed to the other value by pressing WRITE.

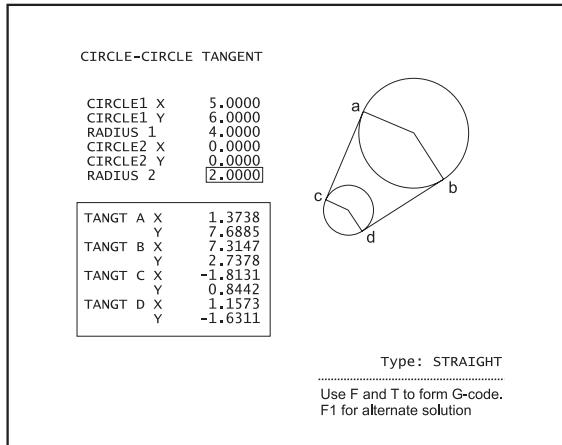
## Circle-Line Tangent Calculator

This is a new feature that provides the ability to determine points of intersection where a circle and a line meet as tangent. The user provides two points, A and B, on a line and a third point, C, away from that line. The control then calculates the point of intersection where a normal line from point C will intersect with the line AB, as well as, the perpendicular distance to that line. This is equivalent to calculating the radius and the tangent point of a circle, which has its center located at point C, and has a radius equal to the distance from point C to the line AB. Note that until all information is entered, the message NO SOLUTION YET is displayed. Note also that if the center of the circle lies on the line, no solution can be calculated.



## Circle-Circle Tangent Calculator

This is a new feature that provides the ability to determine points of intersection between two circles or points. The user provides the location of two circles and their radii. (A zero radius specifies a point instead of a circle.) The control then calculates all the intersection points that are formed by lines tangent to both circles/points. It is important to note that for every input condition where there are two disjointed circles specified, there are up to eight intersection points. Four points are obtained from drawing straight tangents and four points by forming cross tangents. The F1 key is used to toggle between the two diagrams. When F is pressed, the control will prompt for the 'from' and 'to' points (A, B, C, etc.) that specify a segment of the diagram. If the segment is an arc, the control will also prompt for C or W (CW or CCW). Sample G code is then displayed on the input line at the bottom of the screen. When T is entered, the previous 'to' point becomes the new 'from' point and the control prompts for a new 'to' point. When M is pressed, the G code will toggle between G90 and G91 modes. The mode remains active until changed by the user. Because the G code is already on the input line, the user can switch to MDI or EDIT mode and press ENTER to insert the G code into a program.



## Milling/Tapping Help

The Milling/Tapping Help page will help you solve three equations relating to milling and tapping. They are:

- 1) SFM = (CUTTER DIAMETER IN.) \* RPM \* 3.14159 / 12
- 2) (CHIP LOAD IN.) = (FEED IN./MIN.) / RPM / #FLUTES
- 3) (FEED IN./MIN.) = RPM / (THREAD PITCH)

With all three equations, you may enter all but one of the values and the control will compute the remaining value and display it. Note that the RPM value for equations 1 and 2 are the same entry.

When Metric units are selected, the units displayed change to millimeters, mm per minute, threads per mm, and meters, respectively.

## Materials

The Milling calculator has been enhanced to include a new field called MATERIAL, which when highlighted, allows the operator to select a type of material from the list below using the left and right arrow keys. Note that one of the materials is always selected (the first in the list is the default) and the list wraps around at the end.

- LOW CARBON UNALLOYED STEEL
- MEDIUM CARBON UNALLOYED STEEL
- HIGH CARBON UNALLOYED STEEL
- NORMAL CONDITION LOW ALLOY STEEL
- HEAT TREATED TO 32 Rc LOW ALLOY STEEL
- NORMAL CONDITION HIGH ALLOY STEEL
- HEAT TREATED TO 32Rc HIGH ALLOY STEEL
- FERRITIC/MARTENSITIC STAINLESS STEEL
- AUSTENITIC STAINLESS STEEL I
- AUSTENITIC STAINLESS STEEL II
- AUSTENITIC PRECIP. HARDENED STAINLESS

- IRON BASED HEAT RESISTANT ALLOY
- NICKEL BASED HEAT RESISTANT ALLOY
- COBALT BASED HEAT RESISTANT ALLOY
- TITANIUM HEAT RESISTANT ALLOY
- GRAY CLASS 20 CAST IRON
- GRAY CLASS 30, CLASS 40 CAST IRON
- NODULAR CAST IRON
- ALUMINUM ALLOY
- BRASS - BRONZE ALLOY
- HI-VELOCITY MACHINING ALUMINUM ALLOY



VS-Series

A recommended surface speed and chip load will be displayed based on the material chosen, as shown below on the right.

SURFACE SPEED \*.\*\*\* FT/MIN RECOMMENDED \*\*\*\* TO \*\*\*\*\*  
CHIP LOAD \*.\*\*\* IN RECOMMENDED \*.\*\*\* TO \*.\*\*\*

Also, the required horsepower will be calculated and displayed as shown below on the right.

CUT DEPTH \*.\*\*\* IN REQUIRED POWER \*.\* HP

When in metric mode, the required power is displayed as KW. The remaining calculator functions are unchanged.

### Unspecified Material

The material type displayed when the calculator is first accessed is "PRESS < or > FOR MATERIALS" and no recommendations are displayed. The user can select any material from the list and the recommended data will be displayed as before, or return to the unspecified material.

## 2.19 TRAVEL LIMITS

Travel limits in this machine are defined by a limit switch in the positive direction and by stroke limits set by parameter in the negative direction. Prior to establishing the home positions with the POWER UP/RESTART or AUTO ALL AXES buttons, there are no travel limits and the user must be careful not to run the table into the stops and damage the screws or way covers.

Prior to establishing the home positions (POWER UP/RESTART or AUTO ALL AXES), jogging is normally not allowed. Setting 53 can be turned on to allow jogging prior to zero return but this defeats the travel limits and you may damage the machine running the axes into the stops.

Note that all motion is in a negative direction from machine zero except for the Z-axis that can move about 4.5 inches up from machine zero. There are no travel limits for the fourth, A, axis. Travel limits for any auxiliary axes are set into those single axis controls.

When jogging, an attempt to move past the travel limits will not cause an alarm but the axis will stop at the limit. The JOG handle inputs may be ignored in this case.

When running a program, an attempt to move outside of the travel limits will cause an alarm prior to starting the motion and the program will stop. An exception is a circular motion which starts and ends inside of the travel limits but moves outside of the limits during the motion. This will cause an alarm to occur part way through the motion.

Travel limits apply even when running a program in Graphics mode. An alarm is generated and the program will stop.



VS-Series

## 2.20 FEED/RAPID/SPINDLE OVERRIDES

The feed rate can be varied from 10% to 200% of the programmed value while in operation. This is done with the feed rate +10%, -10% and 100% buttons. The FEED RATE override is ineffective during G74 and G84 tapping cycles. FEED RATE override does not change the speed of any auxilliary axes. During manual jogging, the feed rate override will adjust the rates selected from the keypad. This allows for fine control of the jog speed.

The spindle speed can also be varied, from 10% to 150%, using the SPINDLE overrides as above. It is also ineffective for G74 and G84. In the SINGLE BLOCK mode, the spindle may be stopped. It will automatically start up upon continuing the program.

By pressing the HANDLE CONTROL FEEDRATE key, the jog handle can be used to control feedrate from 0% to 999% in +/-1% increments. By pressing the HANDLE CONTROL SPINDLE key, the jog handle can be used to control spindle speed in +/-1% increments (from 0 to 999%).

Rapid moves (G00) may be limited to 5, 25, or 50 % of maximum using the keypad. If the 100% rapid is too fast, it may be set to 50% of maximum by Setting 10.

In the Setting page, it is possible to disable the override keys so that the operator cannot select them. This is Setting 19, 20 and 21.

The FEED HOLD button acts as an override button as it sets the rapid and feed rates to zero when it is pressed. The CYCLE START must be pressed to proceed after a FEED HOLD. When in a FEED HOLD, the bottom left of the screen will indicate this. The door switch on the enclosure also has a similar result but it will display "Door Hold" when the door is opened. When the door is closed, the control will be in Feed Hold and Cycle Start must be pressed to continue. Door Hold and FEED HOLD do not stop any auxiliary axes.

When Parameter 57 flag DOOR STOP SP is set to 1, the door switch will stop the servos and the spindle. In addition, the maximum spindle speed is 750 RPM with the doors open.

There is also an override function for the coolant supply. This is done from the Setting 32. The "NORMAL" setting checks the low coolant alarm and turns the pump on and off with **M** codes. The "OFF" setting ignores the coolant alarm but will alarm if an attempt is made to turn the coolant on. The "IGNORE" setting is used to ignore all coolant commands and the low coolant alarm.

At any time a program is running, the operator may override the coolant setting by pressing the MDI Coolant button. The pump will remain either on or off until the next **M** command or operator action.

Overrides can now be reset through to defaults upon processing M30 and/or RESET. This feature is selected by Setting #83.



## 2.21 Run-Stop-Jog-Continue

This feature allows the operator to interrupt program execution, jog away from the part, return to the interruption point, and then resume program execution. The following is an operation procedure:

1. While a program is running, press FEED HOLD. This will stop all motion (after any canned cycle in process has been completed.)
2. Press X, Y or Z followed by HANDLE JOG. The control will store the current XYZ positions. Note that axes other than X, Y, and Z cannot be jogged.
3. The control will display JOG AWAY. The operator can jog (using the jog handle, remote jog handle, jog and jog-lock buttons) to move the tool away from part, press AUX CLNT to cycle the Through Spindle Coolant (TSC), press COOLNT to cycle the coolant stream, press CW, CCW, STOP, TOOL RELEASE, the jog increment buttons and use the increment switch on the remote jog handle. Note that AUX CLNT requires that the spindle is rotating and the door is closed. Tools can be swapped out and the associated length and diameter offsets adjusted, however, when the program is continued, the old offsets will still be used for the return position and any motion commands already in the queue. It is therefore unsafe to swap out tools and adjust offsets when the program was interrupted during a cut.
4. Jog to a position as close as possible to the stored position, or to a position where there will be an unobstructed rapid path back to the stored position.
5. Return to the previous mode by pressing MEM, MDI, or DNC. The control will only continue normally if the mode that was in effect at the time of the interrupt is re-entered.
6. Press CYCLE START. The control will display the message JOG RETURN and rapid X and Y at 5% to the position when FEED HOLD was pressed, then the same for Z. The rapid rate override keys have no effect during JOG RETURN. Caution: The control will not follow the path the operator used to jog away. Instead, it will perform simple moves without regard for obstacles. Therefore, a crash is possible. If FEED HOLD is pressed during this motion, the control will go into a feed hold state and display the message JOG RETURN HOLD. Pressing CYCLE START will cause the control to resume the JOG RETURN motion. When the motion is completed, the control will again go into a feed hold state.
7. Press CYCLE START again and the program will resume normal operation.



VS-Series

## 2.22 HIGH SPEED MACHINING

High speed machining makes it possible for an increase in the removal rate of material, improve surface finish, and reduce cutting forces. This will reduce machining costs and extend the life of the tools. An additional advantage to using high speed machining is to utilize the more current tool cutting edge materials and tool coatings that are available. The only limitation to high speed machining is the need for rigid, powerful machines capable of the required speeds and feeds, and having a fixture strong enough to hold the workpiece.

### Recommended Tooling

Balanced tools and tool holders are a must to maintain excellent machining conditions. Vibrations during chip cuttings relate directly to early tool failure, poor part finish and even possible spindle damage. Therefore the design of the tool holder increases in importance as the speed increases.

The items necessary for proper high speed machining are toolholder taper cone accuracy, concentric relationship of the cutting tool pocket to the tool holder cone, and actual size tolerance of the tool pocket. The fit between the tool holder and the spindle is the starting point for the machining center. An improved fit for these two items is essential for hi-speed machining success. Size and tolerance of toolholding features such as bore or pilot size should not exceed two or three ten-thousandths of an inch. Concentricity of the toolholding feature (bore or pilot) relative to the tapered shank this should also be within two or three thousandths of an inch.

Where possible use the minimum collet envelope relative to bore size. In other words use the largest bore size for the smallest collet envelope to achieve high grip force with reduced tool holder mass. This also helps in keeping the amount of centrifugal force low and will allow the highest speed possible in relation to the balance specification limitations.

The tool assembly must be balanced to a degree of accuracy that matches the machine spindle requirement. This means a balancing operation will have to be performed, or check the balanced condition of the assembly, each time there is a new tool assembly created or a significant change is made to the existing tool assembly. A significant change could be adjusting cutting tools, changing any part of the tool holder, regrinding or altering a cutting tool, or changing to a new cutting tool or a new tool holder. In order to thoroughly balance the tool holder the retention knob and cutting tool must be in place. Commercial balancing machines give accurate measurement regarding the balance of the tool assembly. Such equipment would be necessary for the initial balancing and eventual rebalancing operations.

The tool holders should be an AT-3 or better with a nylon back-up screw. The tolerances maintained in the AT-3 design are the minimum that would be recommended for a high speed process. The nylon back-up screw increases collet grip on the tool and creates a better seal to aid in coolant transfer.

Use single angle collet chucks and collets for best grip and concentricity. These collet systems are made up of a long single angle located in the holder. The angle per side should be eight degrees or less for best results. Avoid double angle collet systems when maximum rigidity and close tolerance are dictated. It is recommended that minimum engagement of 2/3 of the full length of the bore in the double split single angle collet. However for better results 3/4 to full engagement is preferred if possible.

Collet envelope is the combination of the maximum bore size and the outside shape designation.

Consult the tool holder manufacturer for the current specifications and capabilities of their tool holders. This would include the AT rating and maximum RPM that the tool holder is rated for.



## 2.23 200 HOUR TRY-OUT

Options that normally require a unlock code to activate (Rigid Tap, Macros, etc.) can now be activated and deactivated as desired simply by entering the letter **1** instead of the unlock code to turn it on. Enter a **0** to turn off the option. An option activated in this manner will be automatically deactivated after a total of 200 power-on hours. Note that the deactivation only occurs when power to the machine is turned off, not while it is running. An option can be activated permanently by entering the unlock code. Note that the letter **T** will be displayed to the right of the option on the parameter screen during the 200 hour period. Note that the safety circuit option is an exception; it can be turned on and off only by unlock codes.

## 3. PROGRAMMING

### 3.1 WORK COORDINATE SYSTEM

These machines have three 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 the upper right corner of the mill table. All moves from this point are in a negative machine direction. If a rotary table is connected, an additional **A**-axis work offset is provided.

Work offsets are found in the Offset display by pushing the PAGE UP key. You can display and manually enter work offsets from here. The work coordinate systems on a control with a fifth axis have all been expanded to accommodate **B**, the fifth axis. Work coordinate offsets can be set for the **B** axis in the offset display. Note that the auxiliary axes **C**, **U**, **V**, and **W** do not have any offsets; they are always programmed in machine coordinates.

The Home or machine zero position is X0, Y0, Z0. Travel in the positive direction for the **X** and **Y** axes is limited simply to values less than zero. Positive travel for the **Z** axis is limited to the highest position used for tool changing (about Z4.5). In addition, positive travel for all axes is limited by the home switch, which acts as a limit switch.

Before a tool can machine your part, the control must know where your part is. The work coordinate system tells the control the distance from the work zero point of your part to the machine zero position. The work zero point of the part is decided by the programmer and usually is the common point from which all print dimensions are referenced. The machine zero position is fixed by the machine on power up and does not change. The operator must determine this distance and enter the work zero offset value.

This control automatically chooses the G54 offset values on power-up. If you do not wish to use these, zero out the values in the G54 **X**, **Y**, and **Z** columns or select another work offset.

The G54 through G59 or G110 through G129 offsets can be set by using the PART ZERO SET key. Position the axes to the work zero point of your part. Using the cursor, select the proper axis and work offset number. Press the PART ZERO SET key and the current machine position will be automatically stored in that address. This will work only with the work zero offset display selected. Note that entering a nonzero **Z** work offset will interfere with the operation of an automatically entered tool length offset.

Work coordinates are usually entered as positive numbers, except when Parameter 57, bit "Neg. Work Offset" is set to 1. In this case, the work coordinate numbers are entered as negative numbers.

Work coordinates are entered into the display using numbers only. To enter an **X** value of X2.00 into G54, you would cursor over to the **X** column and enter the number 2.0 only.

See the sections on G52, G92, and Setting 33 for more on work coordinate systems.

Offsets can be sent and received with the RS-232 port. See the "Data Input / Output" section for information on how to do this.



### 3.2 PROGRAM STRUCTURE

A CNC part program consists of one or more blocks of commands. When viewing the program, a block is the same as a line of text. Blocks shown on the display are always terminated by the “ ; ” symbol which is called an EOB. Blocks are made up of alphabetical address codes and the “ / ” symbol. Address codes are always an alphabetical character followed by a numeric value. For instance, the specification of the position to move the X-axis would be a number preceded by the X symbol.

The “ / ” symbol, sometimes called a slash, is used to define an optional block. A block that contains this symbol can be optionally deleted with the BLKDEL button when running a program.

There is no positional requirement for the address codes. They may be placed in any order within the block. The following is a sample program as it would appear on the display. The words following the “.” are not part of the program but are put here as further explanation.

This program will drill four holes and mill a two-inch hole in a four-inch square plate with X and Y zero at the center. The program with comment statements would appear like this.

%	:PROGRAM MUST BEGIN AND END WITH %
O1234 (OP1 SAMPLE MILL PART)	:PROGRAM # AND COMMENT STATEMENT
N1 (TOOL IS A 1/2 INCH STUB DRILL)	:***** NOTES TO OPERATOR
N5 G40 G49 T M06 :	
N100 G00 X0 Y0 Z.5 G43 H1 M3 S1400 T2	:RAPID TO POS, OFFSET 1, SPIN FWD
N101 G01 Z.2 F30.	:FEED 30 INCH/MINUTE TO Z DEPTH
N102 G83 G98 Z-.625 R.03 Q.2 F5.	:PECK TO Z-.625 START .03 ABOVE
N103 X1.5 Y1.5	:DRILL ANOTHER HOLE AT NEW X,Y
N104 Y-1.5	:DRILL 3RD HOLE, PECK DEPTH IS .20
N105 X-1.5	:DRILL FOURTH HOLE
N106 Y1.5	:DRILL FIFTH HOLE
N107 G00 G80 Z.5	:CANCEL CANNED CYCLE
N108 T2 M06	:TOOL CHANGE TO TOOL #2
N2 (T IS 5/8 90 DEG. COUNTERSINK)	:N### ARE LINE NUMBERS
N200 G00 X0 Y0 Z.5 G43 H2 M3 S500	:OFFSET 2, SPINDLE SPEED 500 RPM
N201 G01 Z.2 F30.	:FEED TO Z AT 30 INCH PER MINUTE
N202 G82 G98 Z-.27 R.0 F5.	:SPOT DRILL CYCLE, DRILL AT X0 Y0
N203 X1.5 Y1.5	:SEC HOLE R=START PLANE ABOVE ZERO
N204 Y-1.5	:3RD HOLE G98=RETURN TO INIT POINT
N205 X-1.5	:FOURTH HOLE
N206 Y1.5	:FIFTH HOLE
N207 G00 G80 Z.5	:RAPID TO Z.5
N208 G28 X0 Y0 Z2.0	:ZERO RETURN AFTER MOVE TO X0, Y0
N209 T M06	:TOOL CHANGE
N3 (TOOL IS A 1/2 END MILL)	:N #'S ARE FOR YOUR CONVENIENCE
(SET DIAMETER VALUE TOOL)	:COMMENTS ARE IGNORED BY CONTROL
N300 G00 X0 Y0 Z.5 G43 H3 M3 S1000	:G43 = OFFSET Z IN MINUS DIRECTION
N301 G01 Z.2 F30.	:G01 CAN BE SPECIFIED AS G1
N302 Z-.625 F5.	:FEED TO DEPTH
N303 G01 G41 X-1.00	:COMPENSATE CUTTER LEFT OF LINE
N304 G03 I1.0 D1	:CUT CIRCLE CCW WITH TOOL DIA D1
N305 G00 G40 X00	:RAPID TO CENTER, G40 CANCELS COMP
N306 G00 Z.5	:RAPID OUT OF PART
N307 G28	:ZERO RETURN, Z GOES FIRST THAN X,Y
M30	:RESET PROGRAM TO BEGINNING
%	:END OF TAPE

Please note that each tool is different, which shows the flexibility of the control. For example, to change tools, all that is needed is an M06 command even without a G28 in the previous line. Also, a G28 command can be specified as G28 X0 Y0 Z0 or simply as G28. A "T" command can be put in with the M06 or it can be specified earlier in the program. This gives maximum compatibility with other controls.

More than one program can be stored in the memory of the CNC. Every program stored has an Onnnnn program name address code to define the number of that program. Those numbers are used to identify the program for selection as the main program being run or as a subprogram called from a main program.

### 3.3 ALPHABETICAL ADDRESS CODES

The following is a list of the address codes used in programming the CNC.

#### A Fourth axis rotary motion

The **A** address character is used to specify motion for the optional fourth, **A**, axis. It specifies an angle in degrees for the rotary axis. It is always followed by a signed number and up to three decimal positions. If no decimal point is entered, the last digit is assumed to be 1/1000 degree. The smallest magnitude is 0.001 degree, the most negative value is -99999.000 degrees, and the largest number is 99999.000 degrees.

#### B Fifth axis rotary motion

The **B** address character is used to specify motion for the optional fifth, **B**, axis. It specifies an angle in degrees for the rotary axis. It is always followed by a signed number and up to three decimal positions. If no decimal point is entered, the last digit is assumed to be 1/1000 degree. The smallest magnitude is 0.001 degree, the most negative value is -8380.000 degrees, and the largest number is 8380.000 degrees.

#### C Auxiliary external rotary axis

The **C** address character is used to specify motion for the optional external sixth, **C**, axis. It specifies an angle in degrees for the rotary axis. It is always followed by a signed number and up to three decimal positions. If no decimal point is entered, the last digit is assumed to be 1/1000 degree. The smallest magnitude is 0.001 degree, the most negative value is -8380.000 degrees, and the largest number is 8380.000 degrees.

#### D Tool diameter selection

The **D** address character is used to select the tool diameter or radius used for cutter compensation. The number following must be between 0 and 200. D0 specifies that the tool size is zero and serves to cancel a previous **Dn**. Any other value of **D** selects the numbered entry from the tool diameter/radius list in the Offsets display.

#### E Contouring accuracy

The **E** address character is used, with G187, to select the accuracy required when cutting a corner during high-speed machining operations. The range of values possible for the E code is 0.0001 to 0.25. Refer to the "Contouring Accuracy" section for more information.

#### F Feed rate

The **F** address character is used to select the feed rate applied to any interpolation functions, including pocket milling and canned cycles. It is either in inches per minute with four fractional positions or mm per minute with three fractional positions. When G93 (Inverse Time) is programmed, F is in blocks per minute, up to a maximum of 15400.0000 inches per minute (39300.000 millimeters per minute).

**G Preparatory functions (G codes)**

The **G** address character is used to specify the type of operation to occur in a block. The **G** is followed by a two- or three-digit number between 00 and 187. Each **G** code is part of a numbered group. The Group 0 codes are non-modal; that is, they specify a function applicable to this block only and do not affect other blocks. The other groups are modal and the specification of one code in the group cancels the previous code applicable from that group. A modal **G** code applies to all subsequent blocks so those blocks do not need to re-specify the same **G** code. Multiple **G** codes can be placed in a block in order to specify all of the setup conditions for an operation, provided no two are from the same numbered group. See Section 3 for a detailed list of **G** codes.

**H Tool length offset selection**

The **H** address character is used to select the tool length offset entry from the offsets memory. The **H** is followed by a number between 0 and 200. **H0** will cause no offset to be used and **Hn** will use the tool length entry **n** from the Offsets display. Note that G49 is the default condition and will clear the tool length offsets; so you must select either G43 or G44 to activate tool length offsets. The TOOL OFSET MESUR button will enter a value into the offsets to correspond to the use of G43.

**I Canned cycle and circular optional data**

The **I** address character is used to specify data for some canned cycles and circular motions, either in inches with four fractional positions or mm with three fractional positions. **I** is followed by a signed number between -15400.0000 and 15400.0000 in inches or between -39300.0000 and 39300.0000 mm for metric.

**J Canned cycle and circular optional data**

The **J** address character is used to specify data for some canned cycles and circular motions. It is formatted just like the **I** data. **J** is followed by a signed number between -15400.0000 and 15400.0000 in inches or between -39300.000 and 39300.000 mm for metric.

**K Canned cycle and circular optional data**

The **K** address character is used to specify data for some canned cycles and circular motions. It is formatted just like the **I** data. **K** is followed by a signed number between -15400.0000 and 15400.0000 in inches or between -39300.000 and 39300.000 mm for metric.

**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 M Code miscellaneous functions**

The **M** address character is used to specify an **M** code for a block. These codes are used to control miscellaneous machine functions. Note that only one **M** code is allowed per block of the CNC program and all **M** codes are performed at the end of the block. See the "M Codes" section for a detailed list of **M** codes.

**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 function must 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 **Onnnnn** identification in the first block; it cannot be deleted. Altering the **O** in the first block causes the program to be renamed. An **Onnnnn** can be placed in other blocks of a program but will have no effect and can be confusing to the reader. A colon (:) may be used in the place of **O** in a program, but is always displayed as "**O**".

**P Delay time or program number**

The **P** address character is used to enter either a time in seconds or a program number for a subroutine call. If it is used as a time (for a G04 dwell), it may be a positive decimal between 0.001 and 1000.0. If it is used as a program name (for an M98) or a line number (for an M97), the value may be a positive number without a decimal point, up to 9999.

**Q Canned cycle optional data**

The **Q** address character is used in canned cycles and is followed by a signed number, between 0 and 8380.000 for inches or between 0 and 83800.00 for metric.

**R Canned cycle and circular optional data**

The **R** address character is used in canned cycles and circular interpolation. It is either in **inches** with four fractional positions or **mm** with three fractional positions. **R** is followed by a signed number between -15400.0000 and 15400.0000 for inches or between -39300.000 and 39300.000 for millimeters. It is usually used to define the reference plane for canned cycles.

**S Spindle speed command**

The **S** address character is used to specify the spindle speed in conjunction with M41 and M42. The **S** is followed by an unsigned number between 1 and 99999. The **S** command does not turn the spindle on or off; it only sets the desired speed. If a gear change is required in order to set the commanded speed, this command will cause a gear change to occur even if the spindle is stopped. If the spindle is running, a gear change operation will occur and the spindle will continue running at the new speed.

**T Tool selection code**

The **T** address character is used to select the tool for the next tool change. The number following **T** must be a positive number between 1 and the number in Parameter 65. It does not cause the tool change operation to occur. The **Tn** may be placed in the same block that starts the tool change (M6 or M16) or in any previous block.

**U Auxiliary external linear axis**

The **U** address character is used to specify motion for the optional external linear, **U**, axis. It specifies a position or distance along the **U** axis. It is either in inches with three or four fractional positions or millimeters with three fractional positions. **U** is followed by a signed number between -838.0000 and 838.0000 in inches or -8380.00 and 8380.00 or in millimeters. If no decimal point is entered, the last digit is assumed to be 1/1000 inches or 1/1000 mm.

**V Auxiliary external linear axis**

The **V** address character is used to specify motion for the optional external linear, **V**, axis. It specifies a position or distance along the **V** axis. It is either in inches with three or four fractional positions or millimeters with three fractional positions. **V** is followed by a signed number between -838.0000 and 838.0000 in inches or -8380.00 and 8380.00 or in millimeters. If no decimal point is entered, the last digit is assumed to be 1/1000 inches or 1/1000 mm.

**W Auxiliary external linear axis**

The **W** address character is used to specify motion for the optional external linear, **W**, axis. It specifies a position or distance along the **W** axis. It is either in inches with three or four fractional positions or millimeters with three fractional positions. **W** is followed by a signed number between -838.0000 and 838.0000 in inches or -8380.00 and 8380.00 or in millimeters. If no decimal point is entered, the last digit is assumed to be 1/1000 inches or 1/1000 mm.

**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 -15400.0000 and 15400.0000 for inches or between -39300.000 and 39300.000 for millimeters. 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 -15400.0000 and 15400.0000 for inches or between -39300.000 and 39300.000 for millimeters. 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 -15400.0000 and 15400.0000 for inches or between -39300.000 and 39300.000 for millimeters. If no decimal point is entered, the last digit is assumed to be 1/10000 inches or 1/1000 mm.

**3.4 TIPS AND TRICKS****General Tips**

**Δ Cursor Searching for a Program.** When in EDIT or MEM mode, you can select and display another program quickly by entering the program number (Onnnnn) you want and pressing either the up or down cursor arrow or **F4**.

**Δ Searching for a Program Command.** Searching for a specific command in a program can be done in either MEM or EDIT mode. Enter the address letter code (A, B, C, etc.) or address letter code with the value (A1.23), and press the up or down cursor arrow. If you enter just the address code and no value, the search will stop at the next use of that letter, regardless of the value

**Δ Spindle Command.** You can stop or start the spindle with **CW** or **CCW** any time you're at a Single Block stop or a Feed Hold. When you restart the program with **CYCLE START**, the spindle will be turned back on to the previously defined speed.

**Δ Coolant Pump.** The coolant pump can be turned on or off manually while a program is running, by pressing the **COOLNT** button. This will override what the program is doing until another M08 or M09 coolant command is executed. This also applies to the chip conveyor.

**Δ Coolant Spigot Position.** The coolant spigot position can be changed manually when a program is running, with the **CLNT UP** or **CLNT DOWN** key. This will override what the program commands until another spigot position is commanded with an Hnn or M08 command.

**Δ Optional Stop.** Takes effect on the line after the highlighted line when pressed.

**Δ Block Delete.** Takes effect four lines after that key is pressed when Cutter Compensation is in use, or two lines later when Cutter Compensation is not in use.

**Δ Block Look-Ahead.** This control actually does look ahead for block interpretation, up to 20 blocks. This is not needed for high-speed operation. It is instead used to ensure that DNC program input is never starved, and to allow Cutter Compensation to have non-XY moves inserted while Cutter Compensation is On.

**Δ Memory Lock Key Switch.** This is a machine option that prevents the operator from editing programs and from altering settings when in the locked position.



VS-Series

Δ **Chip Conveyor.** The chip conveyor can be turned on or off when a program is running, either manually using the control keys or in the program using M codes. The M code equivalent to **CHIP FWD** is M31, **CHIP REV** is M32, and **CHIP STOP** is M33. You can set the Conveyor Cycle time (in minutes) with Setting 114, and the Conveyor On-Time (in minutes) with Setting 115.

Δ **Transferring an MDI Program.** You can transfer and save a program in MDI to your list of programs. When in the MDI display, *make sure that the cursor is at the beginning* of the MDI program. Enter a program number (Onnnnn) that's not being used. Then press **ALTER** and this will transfer the MDI data into your list of programs under that program number.

Δ **To Rapid an Axis Home.** You can rapid *all* axes to machine zero by pressing the **HOME G28** key. You can also send just one axis (X, Y, Z, A or B) to machine zero in rapid motion. Enter the letter X, Y, Z, A or B, then press **HOME G28** and that axis alone will rapid home. **CAUTION!** There is no warning to alert you of any possible collision! For example, if the Z axis is down near the part or fixture on the table, and then the X or Y axis is sent home using HOME G28, a crash may result. Care must be exercised. (Any Mill Control ver. 9.49 and above; any Lathe Control ver. 2.24 and above.)

Δ **Rotate Tool Carousel without Changing Tools (M39).** You can rotate the tool carousel without changing tools by using M39 and Tnn. This can be used to recover from some unusual conditions. However, it will also tell the control you have a different tool in the spindle, which may not be the case. (Any Mill Control ver. 2.3 and above.)

## PRGRM /CONVRs

Δ **Program Review.** In the PRGRM display, Program Review allows you to cursor through and review the active program on the right side of the display screen, while also viewing the same program as it's running on the left side of the screen. To bring up Program Review, press **F4** while running a program in MEM mode and PRGRM display.

Δ **Background Edit.** This feature allows you to edit a program in MEM mode and PRGRM display while a program is running. Type in the program number you want to edit (**Onnnnn**) and press **F4**. You can then do simple edits (**INSERT**, **ALTER**, **DELETE** and **UNDO**) either to an existing program, a new program or even the program that is presently running. However, edits to the program that is running will not take effect until that program cycle ends with an M30 or **RESET**.

## POSIT

Δ **Quick Zero on DIST-TO-GO Display.** To clear out and get a quick zero position display, for a distance reference move, use the DIST-TO-GO position display. When you're in the POSIT display and in HANDLE JOG mode, press any other operation mode (EDIT, MEM, MDI, etc.) and then go back to HANDLE JOG. This will zero out all axes on the DIST-TO-GO display and begin showing the distance moved.

Δ **To Origin the POS-OPER Display.** This display is used for reference only. Each axis can be zeroed out independently, to then show its position relative to where you selected to zero that axis. To zero out a specific axis, **PAGE UP** or **PAGE DOWN** in the POSIT display to the POS-OPER large-digit display page. When you Handle Jog the X, Y or Z axis and then press **ORIGIN**, the axis that is selected will be zeroed. Or, you can press an **X**, **Y** or **Z** letter key and then **ORIGIN** to zero that axis display. You can also press the **X**, **Y** or **Z** key and enter a number (X2.125), then press **ORIGIN** to enter the number in that axis display.

Δ **Jog Keys.** The **JOG** keys (+X, -X, +Y, -Y, +Z, -Z, +A, -A, +B, -B) use the jog speeds of 100., 10., 1. and .1 inches per minute listed next to the **HANDLE JOG** key (jogging with the handwheel uses the .1, .01, .001 and .0001 inch increments). You can also adjust feedrate using the **FEED RATE OVERRIDE** buttons, which allow you to increase or decrease feedrate in 10% increments, up to 200%.

Δ **Jog Keys.** You can also select an axis for jogging by entering the axis letter on the input line and then pressing the **HANDLE JOG** button. This works for the X, Y, Z, and A axes as well as the B, C, U, and V auxiliary axes.



## OFFSET

Δ **Entering Offsets.** Pressing **WRITE/ENTER** will add the number in the input buffer to the cursor-selected offset value. Pressing **F1** will replace the selected offset with the number in the input buffer.

Δ **Entering Offsets.** Pressing **OFFSET** again will toggle back and forth between the Tool Length Offsets and Work Zero Offsets pages.

Δ **Entering Offsets.** In the OFFSET display, **F1** will set the entered value into the offsets. **F2** will set the negative of the entered value into the offsets.

Δ **Coolant Spigot Position.** The coolant spigot can be programmed to move to the location entered in the OFFSET display, on the Tool Geometry Offset page under CLNT POS. You can adjust the coolant nozzle up to 10 different positions for a tool. Position 1 is the farthest up and 10 is the farthest down. The nozzle will shift to that position whenever an M08 or Hnn code is executed in the program. When you cursor onto the CLNT POS column, the coolant nozzle location is displayed in the lower left corner.

Δ **Clearing All Offsets and Macro Variables.** When you're in the Tool Length Offset display, you can clear all the offsets by pressing the **ORIGIN** key. The control will prompt: "ZERO ALL (Y/N)?" to make sure this is what you really want to do. If **Y** is entered, all the offsets in the area being displayed will be zeroed. The Work Zero Offset page and the Macro Variables page in the CURNT COMDS display will do the same thing. (Any Mill Control ver. 10.02 and above; any Lathe Control ver. 3.00 and above.)

Δ **The Mill Offers up to 200 Tool Offsets.** Mills now offer up to 200 tool offsets. In all previous versions, the maximum number available was 100 tool offsets. (Any Mill Control ver. 10.22 and above.)

## CURNT COMDS

Δ **Current Commands.** You can see the programmed spindle speed and feedrate on the first page of the CURNT COMDS display, by looking at PGM Fnxxxx and PGM Snxxxx in the column on the far right. You can verify the actual feedrate and commanded spindle speed in the same column, with ACT Fnxxxx and CMD Snxxxx. The actual speed and feed are what the spindle speed and feedrate are really operating at with any adjustments to the **OVERRIDE** keys.

Δ **Tool Life Management.** In the CURNT COMDS display, you can **PAGE DOWN** to the Tool Life management page. On this page the tool Usage register is added to every time that tool is called up in the spindle. You enter the number of times you want that tool to be used in the Alarm column. When the Usage number for that tool reaches the number of uses in the Alarm column, it will stop the machine with an alarm. This will help you monitor tools to prevent them from breaking and parts being scrapped.

Δ **Tool Load Management.** The next **PAGE DOWN** in CURNT COMDS will bring you to the Tool Load page. Spindle load condition can be defined for a particular tool, and the machine will stop if it reaches the spindle load limit defined for that tool. A tool overload condition can result in one of four actions by the control. The action is determined by Setting 84. ALARM will generate an alarm when overload occurs; FEEDHOLD will stop with a Feed Hold when overload occurs; BEEP will sound an audible alarm when overload occurs; or AUTOFEED will automatically decrease the feedrate. This will also help you monitor tools.

Δ **Clearing Current Commands Values.** The values in the CURNT COMDS display pages for Tool Life, Tool Load and Timer registers can be cleared by cursor-selecting the one you wish to clear and pressing **ORIGIN**. To clear everything in a column, cursor to the top of that column (onto the title) and press **ORIGIN**.

## HELP

Δ **Helpful Information.** The HELP display has a list of all the G and M codes available. To see them, press the **HELP** display key and then the letter **C**. For a listing of all the different subjects available in the Help directory, press the letter **D**, then press the letter that's next to the subject area you wish to see.

## CALC

**Δ Transferring Simple Calculations.** While in the Trigonometry, Circular or Milling and Tapping calculator, the number in the simple calculator box (in the upper left corner) can be transferred down to any cursor-selected data line. Cursor to the register you wish to transfer the calculator number to and press **F3**.

**Δ Transferring Calculated Values.** You can transfer the highlighted value in a Trig, Circular, or Milling data register into the calculator box by pressing F4. Use the up and down arrow keys to select the data registers, including the calculator box, and the left and right arrows to select LOAD + - \* /. To enter a highlighted data value into the calculator box, LOAD must be selected when you press **F4**. If one of the operations is selected, pressing **F4** will perform that operation using the number in the highlighted data register and the number in the calculator box.

**Δ Transferring to EDIT or MDI.** In either EDIT or MDI mode, pressing **F3** will transfer the number in the calculator box (when the cursor is on the number in the box) to either the EDIT or MDI input buffer. You will need to enter the letter (X, Y or Z) you wish to use with the number from the calculator.

**Δ Circular Calculator.** The Circular Calculator will list four different ways that a circular move can be programmed using the values entered for a calculated solution. Four different program lines for executing the circular move will be listed at the bottom of the display. One of the four program lines can be transferred to either EDIT or MDI.

1. In the circular calculator, cursor onto the program line you wish to use.
2. Press either **EDIT** or **MDI**, where you wish to insert the circular move.
3. Press the **F3** key, which will transfer the circular move that you highlighted into the input buffer line at the bottom of the EDIT or MDI display.
4. Press **INSERT** to add that circular command line into your program.

**Δ One-Line Expressions.** The CALC display will now accept and evaluate a simple expression. This is a new feature; previously it was only possible to enter a number into the input line of the calculator. Now, the calculator allows you to enter a simple, one-line expression without parentheses, such as  $23*4-5.2+6/2$ . It will be evaluated when the **WRITE/ENTER** key is pressed and the result (89.8 in this case) displayed in the calculator box. Multiplication and division are performed before addition and subtraction. (Any Mill Control ver. 9.49 and above; any Lathe Control ver. 2.24 and above.)

## SETNG

**Δ Scrolling through Settings with Jog Handle.** The jog handle can now be used to scroll through the settings. (Any Mill Control ver. 10.15 and above; any Lathe Control ver. 3.05 and above.)

**Δ** There are so many settings which give the user powerful and helpful command over the control that you should read the entire Settings section of the operator's manual. Here are some of the useful settings, to give you an idea of what is possible.

**Setting 1** AUTO POWER OFF TIMER. This turns the machine off after it is idle for the number of minutes defined in this setting.

**Setting 2** POWER OFF AT M30. This will power off the machine when an M30 command is executed. In addition, for safety reasons, the control will turn itself off if an over-voltage or overheat condition is detected for longer than four minutes.

**Setting 8** PROG MEMORY LOCK. When this is Off, control program memory can be modified. When this setting is turned On, memory edits cannot be done and programs cannot be erased.



**Setting 9 DIMENSIONING.** This changes the machine control from inch to metric, which will change all offset values and position displays accordingly. This setting *will not* change your program to either inch or metric.

**Setting 31 RESET PROGRAM POINTER.** When this is On, the **RESET** key will send the cursor (program pointer) back to the beginning of the program.

**Setting 77 SCALE INTEGER F.** This can be used to change how the control interprets a feedrate. *A feedrate that is entered in your program can be misinterpreted if you do not enter a decimal point* in the Fnn command. The selections for this setting are Default, which assumes a 4-place decimal if no decimal point is entered (i.e., if you enter a 1, it assumes you mean 0.0001); Integer, which assumes a whole number (enter a 1, it assumes 1.0); or .1 (enter 1, it assumes 0.1), .01 or .001 (you get the idea), or .0001, which is the same as the Default setting.

**Setting 84 TOOL OVERLOAD ACTION.** This is used to determine tool overload conditions as defined by the Tool Load monitor page in the CURNT COMDS display (use **PAGE DOWN** to get there). A tool overload condition can result in one of four actions by the control, depending on Setting 84. ALARM will generate an alarm when overload occurs; FEEDHOLD will stop with a Feed Hold when overload occurs; BEEP will sound an audible alarm when overload occurs; or AUTOFEED will automatically decrease the feedrate.

**Setting 85 MAX CORNER ROUNDING.** On a Haas control, this setting is used to set the corner rounding accuracy required by the user. The accuracy defined in Setting 85 will be maintained even at maximum feedrate. The control will only slow at corners *when it is needed*. (This setting defeats all the years of discussion by competitors who say you need multiple blocks of look-ahead.)

**Setting 88 RESET RESETS OVERRIDE.** When this is On, the **RESET** key sets all overrides back to 100%.

**Setting 103 CYC START / FH SAME KEY.** This is really good to use when you're carefully running through a program. When this setting is On, the **CYCLE START** button functions as the Feed Hold key as well. When **CYCLE START** is pressed and held in, the machine will run through the program; when it's released, the machine will stop in a Feed Hold. This gives you much better control when setting up a new program. When you're done using this feature, turn it Off. This setting can be changed while running a program. It cannot be On when Setting 104 is On. (Any Mill Control ver. 9.06 and above.)

**Setting 104 JOG HANDL TO SNGL BLK.** When running a program in MEM mode in the Program or Graphics display, you can use the **SINGLE BLOCK** key to cycle through your program one line at a time, with each press of the **CYCLE START** button, whether the machine is in operation or you're in Graphics. If you turn Setting 104 On, then each counterclockwise click of the jog handle will step through a program line. Turning the handle clockwise will cause a Feed Hold. This setting can be changed while running a program. It cannot be On when Setting 103 is On. (Any Mill Control ver. 9.06 and above.)

**Setting 114 CONVEYOR CYCLE (MIN).** If this is set to zero, the conveyor will operate normally. If another number is entered, it defines how long (in minutes) each cycle will be when the chip conveyor is turned on. The chip conveyor cycle is started with either an M code (M31 or M32) or with the control **CHIP FWD/REV** keys. It will stay on for the time defined in Setting 115, then turn off and not restart until the cycle time in Setting 114 has elapsed.

**Setting 115 CONVEYOR ON TIME (MIN).** This setting works with Setting 114, which defines the conveyor cycle time. Setting 115 defines how long the chip conveyor will stay on during each cycle.

**Setting 118 M99 BUMPS M30 CNTRS.** When this setting is Off, the feature is disabled. When it is On, an M99 command that is used to run a program repeatedly will activate the M30 counters that are in the CURNT COMNDS display (**PAGE DOWN** twice). Note that an M99 will only activate the counters when it is used in a loop mode in a *main* program, not a subprogram. An M99 that's used as a subprogram return or with a P value to jump to another part of the program won't be counted. (Any Mill Control ver. 9.58 and above. Any Lathe Control ver. 3.00 and above.)



VS-Series

**Setting 130 RIG. TAP RETRACT MULT.** This feature augments one introduced in version 10.13, the quick reversal out of a G84 rigid-tapped hole. If it is set to 0 or 1, the machine behaves normally. Setting it to 2 is the equivalent of a G84 command with a J value of 2. That is, the spindle will retract twice as fast as it went in. If this setting is set to 3, the spindle will retract three times as fast. Note that specifying a J value in a G84 command for rigid tapping will override Setting 130. (Any Mill Control ver. 10.18 and above).

## GRAPH

**Δ Zooming in.** In Graphics display, use **F2** to zoom in on the graphic. After pressing F2, press **PAGE DOWN** to zoom in further and **PAGE UP** to expand the view. Use the cursor arrows to position the zoom window over the section of the part that you wish to view in close-up. Press **WRITE/ENTER** to save the new zoom window, and **CYCLE START** to see the close-up graphic run. Press **F2** and then **HOME** to get back to the original full table view.

## PARAM

**Δ Changing Parameters.** Parameters are seldom-modified values that change the operation of the machine. These include servo motor types, gear ratios, speeds, stored stroke limits, lead screw compensations, motor control delays and macro call selections. *Modifying some of these functions will void the warranty on the machine.* If you need to change parameters, contact Haas Automation or your dealer. Parameters are protected from being changed by Setting 7. Be sure to *download and save* a copy of your machine parameters so you'll have a back-up if needed (refer to the LIST PROG section to see how to save your offsets, settings and parameters to a floppy disk).

## DGNOS

This display is used to show the status of the machine for diagnostics and servicing.

## ALARM

**Δ Alarm History Display.** There is an alarm history that displays the previous 100 alarms. Pressing the right or left cursor arrow while in the Alarm display will list the last 100 alarms, with their date and time. You will need to use the cursor up arrow to see the alarms previous to the last one. Pressing either the left or right arrow again will bring you back to the normal Alarm display.

**Δ Alarm History Saved to Disk or RS232.** The last 100 alarms can be saved to a floppy disk by entering a file name and pressing F2 while on the Alarm history page. Alternately, the alarm history can be sent to a PC using RS232 by pressing **SEND RS232**.

## MESGS

**Δ Leaving Messages.** You can enter a message in the MESGS display for the next person, or for yourself. It will be the first display shown when you power up the machine, *if there are no alarms* other than the usual 102 SERVOS OFF alarm. If the machine was powered down using **EMERGENCY STOP**, the MESGS display will not show up when you turn the machine on again. Instead, the control will display the active alarm generated by the emergency stop. In this case, you would have to press the **ALARM/MESGS** key to view a message.

## EDIT

**Δ The Edit Display.** When you press **EDIT**, the first display you see is the Advanced Editor display. This display has menus in the upper left that allow the user to access features of the Advanced Editor; the menus are activated by pressing **F1**. Pressing the **PRGRM/CONVRS** key will bring up the (visually) larger Edit display. Pressing **PRGRM/CONVRS** a second time will get you into the Quick Code editor, which can also be accessed from the Advanced Editor within the F1: HELP pull-down menu.



Δ **Advanced Editor Menus.** Pressing **F1** in the Advanced Editor will activate the menus. The menu selections are made with the jog handle, turning it either clockwise or counterclockwise, or with the cursor arrow keys. Press the **WRITE/ENTER** key to activate the cursor-selected menu item.

Δ **Advanced Editor On-line Help.** In the Advanced Editor, after pressing **F1** to access the menus, on-line Help is displayed in the lower right corner of the screen. To scroll through the Help text, use the **PAGE UP**, **PAGE DOWN**, **HOME**, and **END** keys (remember, the cursor arrows move you through the menu items, not the Help text). In addition, if the **F1** key is pressed during the use of a menu option, the Help text is likewise displayed. Pressing **F1** again will exit the Help display.

Δ **Lower-Case Text.** In the Editor, you can enter lower-case text if it's between parentheses (that is, for comments only). Press the **SHIFT** key first (or hold it in) and then the letter you want in lower case (remember, this works for parenthetical comments only). When lower-case text is selected (highlighted), it will appear in caps; deselected, it returns to lower case. To type the white symbol in the upper left corner of the numeric keys, press **SHIFT** and then the key. These symbols are used for parenthetical comments or for macros.

Δ **Editing in the Advanced Editor.** In the 80-column Advanced Editor you can select a program block and copy it to another location, move it from one location to another, or delete it. To start the block definition, press **F1** to get into the menus, use the jog handle or the cursor arrows to select to select the **EDIT** menu and the **SELECT TEXT** menu item, then press **WRITE/ENTER**. Another way to begin text selection is to put the cursor on the program line where you want the selection to begin and press **F2**. In either case, once you've defined the beginning, you then use the cursor arrows to go to the line where the selection should end, and press **F2** or **WRITE/ENTER**. This will highlight the section you want to copy, move or delete. Then, use the **EDIT** menu (or the **EDIT** keys) to copy (**INSERT**), move (**ALTER**) or delete (**DELETE**) the selected block.

Δ **To Undo an Edit.** Pressing **UNDO** will change back as many as the last ten simple edits (**INSERT**, **ALTER**, **DELETE**) that were done. Sometimes you can even edit some code, run the program, and then if you need to change it back after running it, it may let you do that using the **UNDO** key. **UNDO** does not undo program edits in Block Edit. In Block Edit, it only deselects text.

Δ **Block Editing in the Advanced Editor.** You can copy a line or a block of lines from one program into another. Select the program block you wish to transfer to another program using the method outlined above (in the "Editing in the Advanced Editor" paragraph). Then press **SELECT PROG** (or use the **PROGRAM** menu), scroll to the program you want to copy to, and press **WRITE/ENTER** to select it. It will open up on the right side of the screen. Cursor to where you want the selected text to be placed and press **INSERT** (to deselect text after it's been moved, press **UNDO**). Use the **EDIT** key to go back and forth between two open programs on the screen.

Δ **Block Editing in the Larger Editor Display.** You can move a block of lines from one program to another in the (visually) larger Editor display (press **EDIT** and then **PRGRM/CONVRS** once to get to the larger editor). In this Editor, define the first block by pressing **F1**, and then cursor to the last line you want in the selection and press **F2**. This will highlight that section of the program. Then select another program using **LIST PROG** and copy the selection into the new program by pressing **INSERT**.

Δ **Exiting Block Edit.** You can turn off Block Edit highlighting by pressing the **UNDO** key; the cursor will remain where you're at in the program. **UNDO** will not change back an edit done in Block Edit. **RESET** will also turn off the block highlight, but the cursor will go back to the beginning of the program.

Δ **Advanced Editor Searching.** When the **SEARCH** menu **FIND TEXT** item is used and the text is found, the next press of **F1** to activate the menus will automatically select the **FIND AGAIN** option. Likewise, when the **SELECT TEXT** function on the **EDIT** menu is used, the next activation of the menus will cause the **COPY SELECTED TEXT** option to be highlighted.



VS-Series

Δ **Editing Two Versions of the Same Program.** In Edit mode, **F4** is the hot key that displays another version of the active program for editing. The same program will be displayed on both halves of the screen, and each program can be edited alternately by using the **EDIT** key to switch from one side to the other. Both programs will be updated with the edits done while you're switching back and forth. This is useful for editing a long program; you can view and edit one section of the program on one side of the screen and another section on the other side.

Δ **A Quick Cursor Arrow in the Advanced Editor.** You can call up a cursor arrow with which to scroll through your program quickly, line by line, when you're in the Advanced Editor. For the quick cursor arrow, press **F2** once; then you can use the jog handle to scroll line by line through the program. To get out of this quick-cursor mode and remain where you are in the program, just press the **UNDO** key. (Any Mill Control ver. 9.49 and above; any Lathe Control ver. 2.24 and above.)

## PROGRAMMING

Δ **Program Format at the Beginning and End.** Programs written on a PC and sent to the control from a floppy disk or through the RS-232 port must start and end with a % sign, on a line by itself. The second line in a program received via floppy or RS-232 (which will be the first line the operator sees) must be **Onnnnn**, a six-character program number that starts with the letter **O** followed by five digits. When you create a program on the Haas control the percent (%) signs will be entered automatically, though you won't see them displayed.

Δ **Program Format with M06.** It is not necessary to turn off the coolant (M09), stop the spindle (M05), or move the Z axis home (G28) prior to a tool change. The control will do these tasks for you during a tool change M06 command. However, you may decide to program these commands anyway for convenience and timely execution of a tool change sequence. If you're using Single Block to step through a program you will be able to see the commands when you stop on that line.

Δ **Tapping with G84 or G74.** When tapping, you don't need to start the spindle with an M03 or M04 command. The control starts the spindle for you automatically with each G84 or G74 cycle, and it will in fact be faster if you don't turn on the spindle with an M03 or M04. The control stops the spindle and turns it back on again in the G84 or G74 tapping cycle to get the feed and speed in sync. The operator just needs to define the spindle speed.

Δ **Quick Reverse Out of a G84 Rigid Tapping Cycle.** This feature for rigid tapping has the spindle back out faster than it went into a tapped hole. The way to specify this is with a J code on the G84 command line. J2 retracts twice as fast as the entry motion; J3 retracts three times as fast, and so on, up to J9. A J code of zero will be ignored. If a J code less than 0 or greater than 9 is specified, Alarm 306 — “INVALID I, J, K or Q” — is generated. The J code is not modal and must be specified in each block where this effect is wanted. The J value should not contain a decimal point. (Any Mill Control ver. 10.13 and above.)

Δ **M19 (Orient Spindle) with a P Value.** This feature works on any vector drive mill. Previously, the M19 command would simply orient the spindle to only one position — that suitable for a tool change. Now, a P value can be added that will cause the spindle to be oriented to a particular position (in degrees). If a whole number is used for the P value, no decimal point is needed. For example, M19 P270 will orient the spindle to 270 degrees. Note that P270.001 (or any other fraction) will be truncated to P270, and P365 will be treated as P5. (Any Mill Control ver. 9.49 and above. Any Lathe Control ver. 2.21 and above.)

Δ **M19 (Orient Spindle) with a Fractional R Value.** This feature works on any vector drive mill. An M19 R123.4567 command will position the spindle to the angle specified by the R fractional value; up to 4 decimal places will be recognized. This R command now *needs a decimal point*: if you program M19 R60, the spindle will orient to 0.060 of a degree. Previously, R commands were not used for this purpose and only integer P values could be used. (Any Mill Control ver. 9.49 and above; any Lathe Control ver. 2.29 and above.)



**Δ Duplicating a Program in LIST PROG.** In the LIST PROG mode, you can duplicate an existing program by cursor-selecting the program number you wish to duplicate, typing in a new program number (**O**nnnnn), and then pressing **F1**. You can also go to the Advanced Editor menu to duplicate a program, using the PROGRAM menu and the DUPLICATE ACTIVE PROGRAM item.

## COMMUNICATIONS

**Δ Program Format to Receive.** You can receive program files from a floppy disk or the RS-232 port into the Haas control. Each program must begin and end with a % sign on a line with nothing else on that line. There also must be an Onnnnn program number on the line after the % sign in each program. If there is a (Program Name), it should be entered between parentheses, either after the program number on the same line or on the next line. The program name will appear in the program list.

**Δ Receiving Program Files from a Floppy Disk.** You can load program files from a floppy disc using the I/O menu and the FLOPPY DIRECTORY item of the Advanced Editor. Pressing **WRITE/ENTER** when this menu item is selected will display a list of the programs on the floppy disk. Use the cursor arrow keys or the handwheel to select the file you need to load, and press **WRITE/ENTER**. After loading that file, the floppy directory will remain on display to allow more files to be selected and loaded into the control. **RESET** or **UNDO** will exit this display.

**Δ I/O Menu SEND RS232 or SEND FLOPPY Commands.** You can send programs to the RS232 port or a floppy disk from the Advanced Editor. After selecting the menu item you want (SEND RS232 or SEND FLOPPY), a program list will appear. Select the program you want to save, or "ALL" (at the end of the list) if you wish to send all programs under one file name. You can also select any number of programs using the up and down cursor arrow keys or the handwheel and the **INSERT** key to mark the specific programs to send. If no programs are selected from the list using the **INSERT** key, the currently highlighted program will be sent.

**Δ Sending Multiple Programs Under One File Name.** In the Advanced Editor, you can send multiple program files via the RS232 port or a floppy disk, using the SEND 4S232 or SEND FLOPPY commands under the I/O menu (see the previous paragraph for how to do this). The Advanced Editor allows you to choose several programs (select them using the cursor and the **INSERT** key) and save them under one file name that you type in; then press **WRITE/ENTER** to save and send it.

(Any Mill Control ver. 9.49 and above; any Lathe Control ver. 3.00 and above.)

**Δ Sending Multiple Programs Using Program Numbers.** The SEND FLOPPY item from the I/O menu of the Advanced Editor allows the operator to select one or more programs to be saved to floppy disk. It will prompt you to "ENTER FLOPPY FILENAME." In previous versions, the control would insist on a file name. Now, however, if you do not enter a file name, but simply press **WRITE/ENTER**, the control will save each program (the ones you selected using the cursor and the **INSERT** key) to a separate file on the floppy and use the five-digit program number as the file name. For example, if programs O00123 and O45678 are selected, the new file names created will be O00123 and O45678. (Any Mill Control ver. 9.49 and above; any Lathe Control ver. 3.00 and above.)

**Δ Sending a Program File from LIST PROG Display.** You can send a file or files to a floppy disk or through the RS-232 port from the LIST PROG display. Use the cursor arrows and the **INSERT** key to select the program(s) you want, or "ALL" if you want to send all of them under one file name. When you press **F2** to send the selected program(s), the control will ask for a floppy file name, which can be up to eight characters long with a three-letter extension (8CHRTRS.3XT). Then press **F2** again to send it. You can also use the I/O menu in the Advanced Editor to send and receive program files.

**Δ Sending Multiple Programs from LIST PROG Using SEND RS232.** Several programs can be sent to the serial port by typing all the program names together on the input line without spaces (e.g., O12345O98765O45678) and pressing SEND RS232.

Δ **RS-232 Communications Using X-Modem.** If you are seeing occasional errors when using RS-232 communications, X-Modem (Setting 14) is a standard communications mode which is very reliable when only a few errors occur. Our control supports this, as do almost all software communication packages for PCs.

Δ **Haas Rotary Table Using the Serial Port and Macros.** It is possible to regulate a Haas rotary table using the serial port and macros from the Haas control, or any Fanuc-compatible control. There is a set of sample macros available from the Haas applications department.

Δ **Send and Receive Offsets, Settings, Parameters and Macro Variables to/from Disk.** You can save offsets, settings, and parameters to a floppy disk. Press **LIST PROG** first, then select an OFSET, SETNG or PARAM display page. Type in a file name and then press **F2** to write that display information to disk (or F3 to read that file from a disk). You can also do this with the macro variables by pressing **LIST PROG** first, then selecting the macro variable display page (**PAGE DOWN** from CURNT COMDS).

Δ **Send and Receive Offsets, Settings, Parameters and Macro Variables to/from RS232.** You can also save offsets, settings, and parameters via the RS-232 port. Press **LIST PROG** first, and then select an OFSET, SETNG, or PARAM display page. Press **SEND RS232** to send that display page to the RS-232 port under the file name that you enter. Press **RECV RS232** to read the file via RS-232. You can also do this with the macro variables by pressing **LIST PROG** first, then selecting the macro variable display page (**PAGE DOWN** from CURNT COMDS).

Δ **Deleting a Program File from a Floppy Disk.** A file can be erased from the floppy drive. On the LIST PROG display, type "DEL file name" where "file name" is the name of the file on the floppy disk. Do not use the program number, unless it's also the file name. Press **WRITE/ENTER**. The message "FLOPPY DELETE" will appear, and the file will be deleted from the floppy disk. Note that this feature requires the latest floppy driver EPROM chip version 2.11. (Any Mill Control ver. 10.02 and above; any Lathe Control ver. 3.00 and above.)



### 3.5 FOURTH AXIS SETUP

#### INTRODUCTION

#### Full-Fourth Axis

When adding an indexer to the Haas mill, the settings must be set for the specific table. You must set the mill setting for the indexer that you are using. This will automatically give you the parameters that you need for that series indexer. Warning: failure to match the correct brush or brushless rotary setting to the actual product being installed on the mill may cause motor damage. "B" in the settings denotes a brushless rotary product. Brushless indexers have two cables from table and two connectors at the mill control.

#### SETTINGS

Turn the mill on, press the E-stop switch and leave it in. Go to the settings page by pressing the setting button once. Go to setting 30 by pressing the 3 and 0 keys and then the down arrow key. This is the setting for fourth axis. Use the right arrow key to scroll to the indexer that you have. Press the write key to save the new setting. Press the down arrow key, then input the approximate diameter of your part with decimal point and the write key. (This sets the feed rate at the diameter of your part.) The position page will now show the position of that axis in degrees. Do the same for the fifth axis if you have a TRT.

#### PARAMETERS

In rare cases you may be required to modify some parameters to get a specific performance out of your indexer. Do not do this without a list of parameters to change. (If you did not get a list of special parameters with your indexer, then you do not need to change the default parameters—DO NOT CHANGE THEM. This will void your warranty.)

To change parameters for a forth or fifth axis indexer, proceed as follows: Press the e-stop switch and leave it in. You must then turn off the parameter lock. Go to the settings page by pressing the setting button. Press the 7 button and push the down arrow. This will get you to setting 7. With the cursor on setting 7, press the right arrow key and press the write key. Setting 7 is now off. Go to the parameter page and press the digits for the parameter that you need to change and press the down arrow key. Press the digits for the new parameter and press the write button. Verify the parameter reads as it is printed on your list, with the same digits. If not, retype and write it. With all parameters changed and verified to be correct, rotate and release the e-stop switch. Press the reset switch 3 times. The "servos off" alarm will disappear from the screen. Go back to the Settings page and lock the parameters. Home the indexer and verify that it operates properly by pressing the A button on the alpha keypad and pressing the handle jog mode button. Jog the A axis. The indexer should move to position without any noise (other than the air brake if applicable). Check for proper ratio by marking the table, rotate 360degrees as viewed in the position page and check mark to be in the same location. If it is close (within 10 degrees) then the ratio is correct.

#### INITIAL START-UP

Turn on the mill (and servo control, if applicable) and home the indexer. All Haas indexers home in the clockwise direction as viewed from the platter. If the indexer homes counter-clockwise, press E-stop and call the Haas service department.

#### AUXILIARY AXIS CONTROL

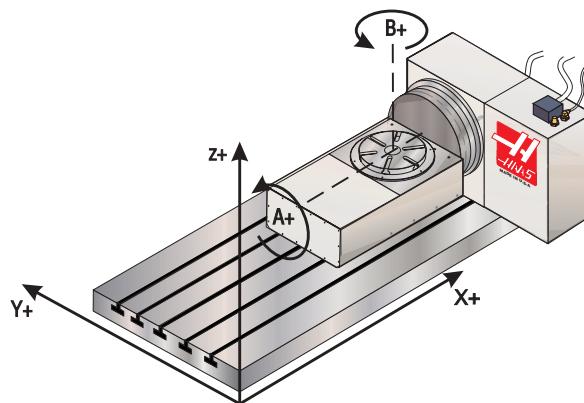
Auxiliary axes are C,U,V & W, and are limited to single axis non-interpolated moves. For more information contact your dealer.

### 3.6 FIFTH AXIS CONTROL

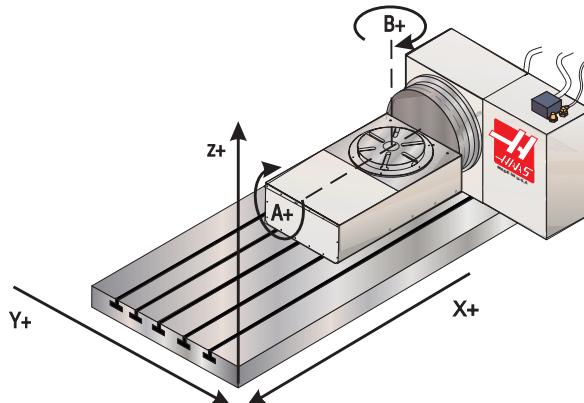
This section describes the features of a HAAS five-axis control.

#### The Coordinate System

The layout of the **A** and **B** axes on the HAAS five-axis control are depicted below. The **A** axis is rotary motion about the **X** axis, while the **B** axis determines rotary motion about the **Y** axis. The right hand rule can be used to determine axis rotation for the **A** and **B** axes. When placing the thumb of the right hand along the positive **X** axis, the fingers of the right hand will point in the direction of tool movement for a positive **A** axis command. Likewise, when placing the thumb of the right hand along the positive **Y** axis, the fingers of the right hand will point in the direction of tool movement for a positive **B** axis command. It is important to remember that the right hand rule determines direction of tool movement and not the table movement direction. For the right hand rule, the fingers will point opposite of the positive rotary table movement. Refer to the figures.



*Work coordinates (Positive direction).*



*Table movement (Positive command).*

**NOTE:** The figures represent one of many possible machine tool and table configurations. You may have different table movements for positive directions, depending on the equipment, parameter settings, or five-axis software being used.



## Alphabetical Address Codes

### B Fifth axis rotary motion

The **B** address character is used to specify motion for the optional fifth, **B**, axis. It specifies an angle in degrees for the rotary axis. It is always followed by a signed number and up to three fractional decimal positions. If no decimal point is entered, the last digit is assumed to be 1/1000 degrees. The smallest magnitude is 0.001 degrees, the most negative value is -8380.000 degrees, and the largest number is 8380.000 degrees.

### Preparatory Functions (G codes)

In a control configured with a fifth axis, all G codes that have an option for an **A** axis motion command can also simultaneously command fifth axis, **B**, motion. Of course, since address **B** is modal, it can be entered on any line. The following G codes are users of address B:

G00	G03	G29	G73	G77	G83	G86	G89	G101
G01	G10	G31	G74	G81	G84	G87	G92	G102
G02	G28	G36	G76	G82	G85	G88	G100	G136

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

Fifth-axis programming is not effected by the selection of metric. It is always programmed in degrees.

## Work Coordinate Systems

The work coordinate systems on a control with a fifth axis have all been expanded to accommodate **B**, the fifth axis. Work coordinate offsets can be set for the **B** axis in the offset display.

## M Codes

### M12 Engages the fifth axis brake.

This M code waits until the M-fin signal is received through the secondary RS-232 port.

### M13 Releases the fifth axis brake.

This code activates the fifth axis brake relay and waits for the M-fin signal, releases the relay upon receipt, and waits for the M-fin signal to cease. The RESET key will terminate any operation that is hung-up waiting for M-fin.

Both M12 and M13 are aliased by M10 and M11 on fourth-axis machines.

## Handle/Jog

All aspects of handle jogging for the fifth axis work as they do for the other axes. The exception is the method of selecting jog between axis **A** and axis **B**.

By default the '+A' and '-A' keys, when pressed, will select the **A** axis for jogging. The display will show "JOGGING A AXIS HANDLE .01" while you are jogging the **A** axis. The **B** axis can be selected for jogging by pressing the 'shift' key, and then pressing either the '+A' or '-A' key. When this is done the control will switch to jogging the **B** axis and the display will change to "JOGGING B AXIS HANDLE .01".

The axis assigned to the '+A' and '-A' keys will remain selected for jogging even if the operating mode is changed or if the machine is turned off. The selected axis for '+A' and '-A' can be toggled by pressing the shift key prior to pressing the '+A' or '-A' keys.

## Settings

The fifth axis, or rotary table, may be disabled from the setting page and removed from the machine. Do not connect or disconnect any cables with the control on. If you do not disable the fifth axis when it is disconnected, you will get an alarm.

The settings listed here are settings that have been added to a fifth-axis control.

<u>Number:</u>	<u>Name:</u>	<u>Range of value:</u>
78	FIFTH-AXIS ENABLE	OFF,S5C,7RT,9RT,SRT,11RT, HA5C, HRT 160, HRT 210, HRT 310
79	FIFTH-AXIS DIAMETER	0 to 29.9999 inches
80	MIRROR IMAGE B AXIS	ON or OFF

The following is a detailed description of each of the settings.

### 78 FIFTH-AXIS ENABLE

This is an on/off setting. When it is off, the fifth axis is disabled and no commands can be sent to that axis. When it is on, the selected rotary table type parameters are called up. A change to rotary parameters is saved under the selected table type for later recall.

### 79 FIFTH-AXIS DIAMETER

This is a numeric entry. It is used to set the angular feed rate of the **B**-axis. It must be in the range of 0.0 to 29.9999 inches. Since the feed rate specified in a program is always inches per minute (or mm per minute), the control must know the diameter of the part being worked in the **B**-axis in order to compute the angular feed rate. When this setting is set correctly, the surface feed rate on a rotary cut will be exactly the feed rate programmed into the control. The feed rate will be correct only as long as the axis remains orthogonal (at right angles to) to the Y axis.

### 80 MIRROR IMAGE B-AXIS

This is an On/Off settings. When it is off, axes motions occur normally. When it is on, the B axis motion is mirrored (or reversed) around the work zero point.

#### Tool Offsets

Work offsets can be specified for the fifth axis **B** address.

The work offset display is found on the offset display page. If the fifth axis is enabled, setting 78, then additional data fields are made available for the fifth axis.

The function of the offset display page has been modified slightly to accommodate multiple axes. Only the axes that are enabled are displayed on the work offset display. For example, if the fourth axis is enabled, setting 30, and the fifth axis is not, then display will show X,Y,Z,A. In this case, all axes can be displayed on the entire display. If only the fifth axis is enabled, then the display will show X,Y,Z,B. Since the fourth axis is disabled, there is no need to display it.

If both **A** and **B** axes are enabled, then the last column of the offsets display serves a dual purpose. Either **A** or **B** can be accessed in the last column. By using the left or right arrow keys, the screen cursor can be moved into the axis field that is to be set. When the right most field is highlighted by the cursor and the top of the column indicates **A**, then the values in this column represent **A** axis offsets. If you press the right arrow key once, the cursor stays in the same place, but the column and its values will change to **B** axis parameters. You can view and modify work offsets in this manner using the left and right arrow keys when both axes are enabled.

#### Parameter Display

The Parameter Display organization has been modified to accommodate the extra axis parameters. Parameter numbers have remained the same as in a four-axis control. Additional parameters have been added for the fifth.



The parameters have been reorganized, so that logically-associated parameters are grouped together. These logical groupings are placed together into contiguous screens called pages. The most commonly changed parameters have been placed at the beginning of the page list. A list of the parameter pages and the order of succession in the control are given in the following table.

PAGE TITLE	DATA DESCRIPTION
COMMON SWTCH	Non-axis bit switches.
COMMON PAGE1	First page of non-axis parameters.
COMMON PAGE2	Second page of non-axis parameters.
COMMON PAGE3	Third page of non-axis parameters.
MACRO M CALL	Parameters that alias M codes to subroutines.
MACRO G CALL	Parameters that alias G codes to macros.
X BIT SWITCH	Bit switches for the X axis.
X PARAMETERA	First page of X axis parameters.
X PARAMETERB	Second page of X axis parameters.
Y BIT SWITCH	Bit switches for the Y axis.
Y PARAMETERA	First page of Y axis parameters.
Y PARAMETERB	Second page of Y axis parameters.
Z BIT SWITCH	Bit switches for the Z axis.
Z PARAMETERA	First page of Z axis parameters.
Z PARAMETERB	Second page of Z axis parameters.
A BIT SWITCH	Bit switches for the A axis.
A PARAMETERA	First page of A axis parameters.
B BIT SWITCH	Bit switches for the B axis.
B PARAMETERA	First page of B axis parameters.
B PARAMETERB	Second page of B axis parameters.
X SCREW COMP	X axis screw compensation value.
Y SCREW COMP	Y axis screw compensation value.
Z SCREW COMP	Z axis screw compensation value.

The HOME key displays the first parameter page "COMMON SWTCH". Pressing the PAGE DOWN key will display the next page of parameters in the above list. The END key displays the last parameter page "B PARAMETERB". Pressing the PAGE UP key will display the preceding page of parameters in the above list. All other features on the parameters display have remained the same. So, if you are unfamiliar with the new format of the parameters, you can still search by parameter number. Enter the number of the parameter you want to see or view and press the up or down arrow key. The page that the parameter is on will be displayed and the parameter being searched for will be highlighted.

### Diagnostic Data Display

The first page of diagnostic data shows two discrete outputs that control the rotary axes brakes; they are labeled "Brake 4th Axis" and "Brake 5th Axis". The 4th axis is synonymous with the A-axis, and the 5th axis is synonymous with the B-axis. If the Air Brake Parameter for the axis is set to 1, the HAAS control unclamps the brake whenever motion is commanded to the rotary axis and sets the brake back to its previous state as soon as the motion stops. The brake is activated by a relay. The two discrete outputs show the state of the brake relays. If the output is high (1) the brake is unclamped. When the machine is first powered up the brake will be unclamped, so the output will be low (0). When the brake is unclamped the message A UNCLMP or B UNCLMP is displayed in the message area near the bottom of the display.

The M11 and M13 codes command the brake to be unclamped. When these codes are in effect, the axis brake will remain unclamped even after motion has stopped.

The second page of diagnostic data show the status of inputs from the motor interface board. Additional inputs for the expanded motor interface board are listed under "INPUTS4". These bits are monitored by the control to determine if the interface for the B axis is working correctly.

## Parameters

Additional parameters specifying the fifth axis have been added to the parameter list **B** axis parameter numbers range from 151 through 175. The parameter descriptions and their number assignments are listed below in the order that they are arranged on the **B** parameter pages.

### PARAMETERS LISTED HERE IN ORDER OF APPEARANCE ON DISPLAY PAGES.

#### **B BIT SWITCH**

Parameter	151	B	SWITCHES
-----------	-----	---	----------

#### **B PARAMETER A**

Parameter	152	B	P GAIN
Parameter	153	B	D GAIN
Parameter	154	B	I GAIN
Parameter	155	B	RATIO (STEPS/INCH)
Parameter	156	B	MAX TRAVEL (STEPS)
Parameter	157	B	ACCELERATION
Parameter	158	B	MAX SPEED
Parameter	159	B	MAX ERROR
Parameter	160	B	FUSE LEVEL
Parameter	161	B	BACK EMF
Parameter	162	B	STEPS/REVOLUTION
Parameter	163	B	BACKLASH
Parameter	164	B	DEAD ZONE
Parameter	165	B	IN POSITION LIMIT B
Parameter	166	B	MAX CURRENT
Parameter	167	B	D*D GAIN FOR B

#### **B PARAMETER B**

Parameter	168	B	ACC/DEC T CONST
Parameter	169	B	PHASE OFFSET
Parameter	170	B	GRID OFFSET
Parameter	171	B	EXACT STOP DIST
Parameter	172	B	FRICTION FACTOR
Parameter	173	B	ACCEL FEED FORWARD
Parameter	175	B	AIR BRAKE DELAY

## Setting Up a Fifth Axis

Installing a fifth axis is the same as installing a fourth axis. Refer to the "Setting Up A Fourth Axis" section of this manual for detailed instructions on fifth axis installation.



### 3.7 HIGH SPEED MACHINING

High Speed Machining is an option that can be added to any Haas mill. It is most often required for the machining of smoothly sculpted shapes as is typical of mold making.

The Haas option for "High Speed Machining" increases the amount of lookahead to 80 blocks and allows full speed (500 inches per minute) blending of feed strokes. Determining when you need the High Speed Machining option can be done with the following table of stroke lengths:

Machine Type	VF0..VF2	VF3..VF5	VF6..VF9	HS-3/R VS-3
Feed in/min:	Stroke Length in inches			
20	0.0039	0.0051	0.0077	0.0154
50	0.0241	0.0321	0.0482	0.0963
100	0.0963	0.1284	0.1927	0.3853
150	0.2167	0.2890	0.4335	0.8670
200	0.3853	0.5138	0.7707	1.5413
300	0.8670	1.1560	1.7340	3.4680
400	1.5413	2.0551	3.0826	6.1652
500	2.4083	3.2111	4.8166	9.6332

To use the above table starting with your machine and feed rate, find the listed stroke length. If you are using linear moves longer than the stroke length, you do not need high speed machining. If you are using linear moves shorter than the stroke length, you do need high speed machining.

It is important to understand that high speed machining works best with smoothly blended shapes where the feed rate can remain high through the blend of one stroke to the next. If there are sharp corners, the control will always need to slow down; even if you have the high speed machining option. Sharp corners can never occur at high speed; either rounding of the corners occurs or the control must slow the feed rate.

The affect that blending of strokes can have on feed rate is always to slow down motion. It can never speed up motion. The programmed feed rate (F) is thus a maximum and the control will sometimes go slower than that in order to achieve the required accuracy.

Remember that too short of a stroke length can result in too many data points and that can result in a "blocks per second" demand which is above what the control can do. Check how your CAD system generates data points to insure that you do not exceed 1000 blocks per second.

Also remember that too few data points can result in either "facetting" or blending angles which are so great that the control must slow down the feed rate. Facetting is where the desired smooth path is actually made up of short, flat, strokes that are not close enough to the desired smoothness of the path. When the angle of blending of strokes is too large, the control must reduce the feed rate in order to get that sharp angle.

### 3.8 PROGRAMMABLE COOLANT SPIGOT

The optional programmable coolant spigot allows the user to direct the coolant stream to the most optimum location in order to flush out chips from the cutting area. The direction of the coolant can be changed by the CNC program.

#### Warning

**Do not rotate the coolant spigot by hand. Serious motor damage will occur.**

### OPERATING THE COOLANT SPIGOT

When the spigot is enabled it will search for home if it ever loses position. If after 3 contiguous searches for home the spigot has not found home, then Alarm, 193 COOLANT SPIGOT FAILURE, is generated.

### OFFSET DISPLAY WITH SPIGOT ENABLE

When the spigot is enabled, an additional field can be accessed on the tool offset display. The left column indicates CLNT POS for coolant position. By default this column has all zeros for the coolant position. If the position is zero, then the spigot will not be moved when the H code for that tool offset is in effect and an M08 is encountered in the program. If a value is placed into the field, then the spigot will be moved to that value if the corresponding H code is in effect and M08 is executed.

### SPIGOT PROGRAM COMMANDS

There are two ways that the spigot can be moved under program control. The first, as just discussed, is by entering positional values into the CLNT POS fields on the tool offset display page. Having positional values entered into the CLNT POS field does not mean that the spigot will move. Only when the M08 command is executed and the current H code has a value in its CLNT POS field, will the spigot move to the designated position.

The second method of moving the spigot is by programming M34 or M35.

M34 moves the spigot in a positive direction. If the spigot is at the end of travel then no spigot movement occurs. For example, if the current spigot position is at 8 and M34 is executed, then the spigot will move to position 9. On a vertical mill this would lower the coolant stream.

M35 moves the spigot in a negative direction. If the spigot is at home position then no spigot movement occurs. For example, if the current spigot position is at 8 and M35 is executed, then the spigot will move to position 7. On a vertical mill this would raise the coolant stream.

It is important to note that each programming method requires the operator to specifically program the spigot in reference to the particular tool being used, while also taking into consideration the tool length, width and size of the part.



## SPIGOT PARAMETERS

The parameters that control the spigot are below.

SPIGOT POSITIONS	Parameter	206
SPIGOT TIMEOUT (MS)	Parameter	207

## SPIGOT M CODES

### M34 Increment Coolant Spigot Position

M34 increments the current spigot position one place. Incrementing the spigot position causes the spigot to advance away from the spigot home position. The home position is designated as zero. If the current spigot position is 5 and M34 is executed, then the current spigot position will advance to position 6. The Spigot home position places the spigot at the most positive Z axis location that the spigot can attain. Incrementing the spigot then lowers the coolant stream direction.

### M35 Decrement Coolant Spigot Position

M35 Decrements the current spigot position one place. Decrementing the spigot position causes the spigot to move toward the spigot home position. The home position is designated as zero. If the current spigot position is 5 and M35 is executed, then the current spigot position will move to 4. The Spigot home position places the spigot at the most positive Z axis location that the spigot can attain. Decrementing the spigot will raise the coolant stream direction.

#### Warning

**Do not rotate the coolant spigot by hand. Serious motor damage will occur.**

## 3.9 THROUGH THE SPINDLE COOLANT (TSC)

\*This option is not field installable

## OPERATION

The HAAS Through The Spindle Coolant (TSC) option includes an auxiliary coolant pump that is used to supply high pressure coolant to the cutting tool. The wear surfaces of the seal are engaged only when TSC is in use.

Maximum pressure is 300 psi when using small orifice tools. Pressure will be lower when using large orifice tools. Maximum flow is 5 gpm. The maximum spindle speed when using the TSC system is 10,000 RPM.

M88 and M89 control the TSC. M88 is used to turn the system on, and M89 is used to turn it off.

The AUX CLNT button on the control panel may also be used to control TSC. Pressing this button while in MDI mode will turn on the TSC system, and pressing it again will shut off the system.

---

**NOTE:** Running an M04 (Spindle Reverse) command with TSC on is not recommended.

Coolant will flow from the drain line during normal TSC operation. Up to 2 cups per minute flow is normal.

When the coolant system is turned off (M89), the spindle is stopped, the pump is shut off, and air flows through the spindle and the TSC drain line for 2-1/2 seconds to purge leftover coolant.

---

**NOTE:** When using small orifice tooling, increasing the value in parameter 237 will help purge left over coolant. For example, 1/4" twist drill - set parameter 237 to 5000 (units are in milliseconds). The minimum value is 2500.

If coolant pressure does not come on within 60 seconds, the system shuts down and gives Alarm 151 (Low Tool Coolant), check the coolant level in the coolant tank.

When TSC is on, and RESET or EMERGENCY STOP are pressed the control turns off the TSC coolant pump, turns on purge for the time in Parameter 237 and then turns off purge.

Refer to the "Maintenance" section for TSC system maintenance information.

#### SAMPLE PROGRAM

Note that the M88 command appears before the spindle speed command in this program. This is a good programming practice; otherwise, having the M88 after the spindle speed command will stop the spindle, start TSC, then restart the spindle, slowing the cycle time.

T1 M6;	(TSC Coolant Through Drill)
G90 G54 G00 X0 Y0;	
G43 H06 Z.5;	
M88;	(Turn TSC on)
S4400 M3;	
G81 Z-2.25 F44. R.03;	
M89 G80;	(Turn TSC off)
G91 G28 Z0;	
M30;	

#### GENERAL WARNINGS

- The TSC pump is a precision gear pump and will wear out faster and lose pressure if abrasive particles are present in the coolant.
- Shortened pump life, reduction of pressure and increased maintenance are normal and to be expected in abrasive environments and are not covered by warranty.
- Use of coolants with extremely low lubricity can damage or destroy the TSC coolant tip and pump.
- Running an M04 (Spindle Reverse) command with TSC on is not recommended.
- When machining castings, sand from the casting process and the abrasive properties of cast aluminum and cast iron will shorten pump life unless a special filter is used in addition to the 100 mesh suction filter. Contact HAAS for recommendations.
- Machining of ceramics and the like voids all warranty claims for wear and is done entirely at customer's risk. Increased maintenance schedules are absolutely required with abrasive swarf. The coolant must be changed more often and the tank thoroughly cleaned of sediment on the bottom. An auxiliary coolant filter is recommended.
  - Proper tooling, with a through-hole, must be in place before using the TSC system. **Failure to use proper tooling will flood the spindle head with coolant and void the warranty.**



- Use a pull stud with "45 Degree, P40T Type 1, inch threads" built to JMTBA standard "MAS 403-1982". If the machine is equipped with the optional BT tool changer, use BT tooling only. Contact the tool manufacturer for further information. **Pull studs are available through HAAS. Refer to the Technical Reference section Reference manual for the proper tool part numbers and identification.**
- Coolant will be used more quickly when the TSC system is in use. Make sure to keep the coolant level up and to check the level more frequently (check after every eight hour shift). **Premature wear of the pump can result from running with a low coolant level in the tank.** The spindle will shut off automatically if the coolant pressure gets too low.

### TSC PARAMETERS

The following parameters (and bits) apply only to the Through The Spindle Coolant system:

#### Parameter 237            **TSC CLNT LINE PURGE**

This can be increased by the machine user if desired to help purge coolant when using small orifice tooling. Use values from 2500 to 6000 milliseconds. Slower tool changes during TSC use will result from increasing the purge time.

#### WARNING!

**These are factory preset parameters. Changing them may void the warranty.**

**Parameter 235            TSC PISTON SEAT**

**Parameter 236            TSC LOW PR FLT**

**Parameter 238            MAX TSC SPINDLE RPM**

**Parameter 209            COMMON SWITCH 2**

The bit, "TSC ENABLE" in **Parameter 209**, is set to "1".

**Parameter 278            COMMON SWITCH 3**

The bit, "TSC PRG ENBL" in **Parameter 278**, is set to "1".

### TSC M CODES

The following M codes apply only to the Through The Spindle Coolant system:

#### **M88        Thru Spindle Clnt ON**

M88 performs the following operations:

- Stop the spindle
- Turn on the TSC pump
- Wait for coolant pressure
- Restart spindle

#### **M89        Thru Spindle Clnt OFF**

M89 performs the following operations:

- Stop the spindle
- Turn off the TSC pump
- Turn on purge
- Wait for [Parameter 237] for coolant to purge
- Turn off purge

The following pre-existing M codes perform a slightly different function when Through the Spindle Coolant is turned ON:

**M00 Stop Prog****M01 Optional Stop**

When TSC is ON, M00 and M01 will shut it OFF, as in M89.

**M06 Tool Change**

When TSC is ON, M06 causes the following operations to be performed:

- Orient the spindle and move Z-axis to tool change position
- Turn OFF the TSC pump
- Turn on purge
- Wait for [Parameter 237] for coolant to purge
- Turn off purge
- Perform tool change

TSC then remains OFF until M88 is used.

**M30 Prog End Rew**

When TSC is ON, M30 will shut it OFF as in M89, then perform an M30 operation.

**ALARM DESCRIPTION****151 Low Tool Coolant**

This alarm will shut off the spindle, feed and pump all at once. It will turn on purge, wait for the time specified in parameter 237 for the coolant to purge, and then turn off the purge. If this alarm is received, check the coolant tank level, the filter and intake strainer for any clogging. If no problems are found with any of these, and none of the coolant lines are clogged or kinked, call you dealer.

**198 Precharge Failure**

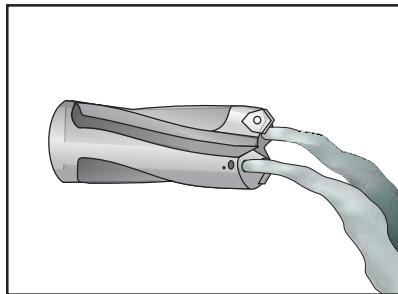
This alarm is received if the precharge fails for greater than 0.1 seconds. It will shut off the feed, spindle and pump all at once. If received, check all air lines and the air supply pressure. **This alarm does not apply to 50 taper.**



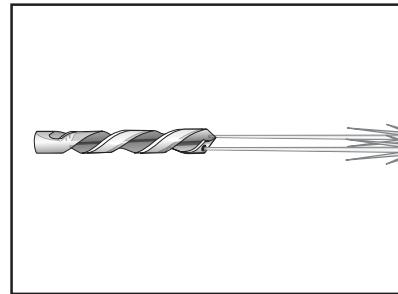
VS-Series

### TSC PRESSURE EFFECTS

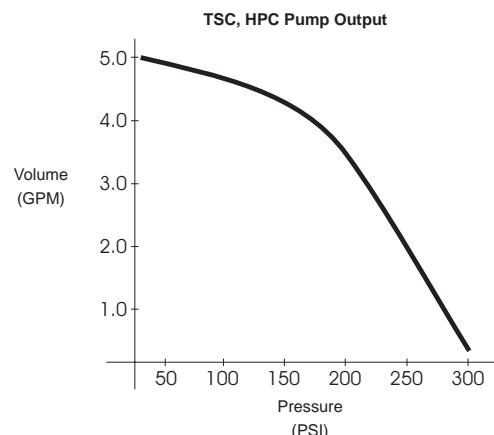
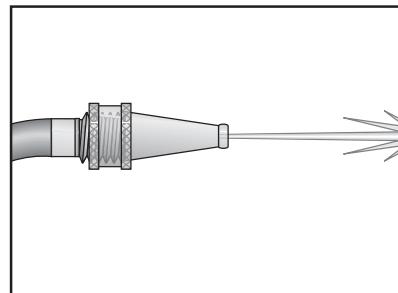
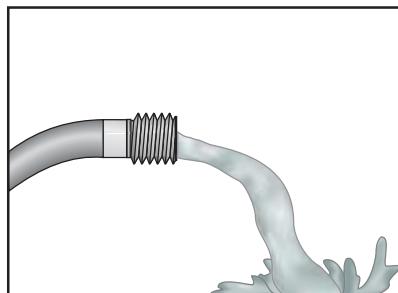
On machines using TSC and/or HPC during cutting operations, tooling size will have to be taken into consideration. As shown below, proper TSC and/or HPC system operation will deliver different pressures at the orifice of the tool; this depends on the diameter and number of coolant passages in the tool.



Larger tooling has larger diameter coolant passages. Coolant flow is higher at lower pressures.

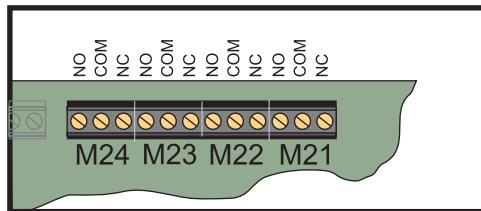


Smaller tooling has smaller diameter coolant passages. This produces higher pressures at lower flow.



### 3.10 8 "M" OPTION

This option adds 8 additional outputs for each 8M option. The machine can be fitted with two 8M options for a total of 16 additional outputs. These outputs can be used to activate probes, auxiliary pumps or clamping devices etc. The 8M relay board contains 8 relay outputs (M21- M28) and 2 terminal strips P4 and P5. Each terminal strip has 12 positions which are Normally Open, Normally Closed and Common.



*8M Relay Board*

A total of 4 banks of 8 relays are possible in the Haas system. Banks 0 and 1 are internal to the I/O PCB. Bank 1 includes the M21-25 relays at the top of the IOPCB. Bank 2 addresses the first 8M option PCB. Bank 3 addresses the second 8M option PCB.

---

**NOTE:** Bank 3 may be used for some Haas installed options and may not be available. Contact the Haas factory for more details.

Only one bank of outputs may be addressable with M-codes at a time. This is controlled by parameter 352 "Relay Bank Select". Relays in the non-activated banks are only accessible with macro variables. Parameter 352 is shipped set to "1" as standard. When either one or two 8M options are installed, the M-fin and probe cables are moved to the first 8M option PCB and parameter 352 is set to "2". With the 8M option, M-codes M21-28 correspond to relays labeled M21-28.

Bank addressing on the 8M PCB itself is done through selectable jumpers. Only one address should be selected at a time. The MCD jumper should be set to JP1 for bank 1 (first 8M option). The MCD jumper should be set to JP2 for bank 2 (second 8M option). The other positions are used by service only for installation in older controls. See the following figure.

M51-M58 will turn on the relays and M61-M68 will turn off the relays. M51 and M61 correspond to M21, etc. on the 8M relay board.

---

**NOTE:** Some or all of the M21-25 on the I/O PCB may be used for factory installed options. Inspect the relays for existing wires to determine which have been used. Contact the Haas factory for more details.

Terminals normally closed: 1, 4, 7, 10

Terminals normally open: 3 , 6, 9, 12

Common Terminals: 2, 5, 8, 11



VS-Series

### 8M Relay Board Connectors:

#### P4 Contains:

- M21 M Function
- M22 Probe Option
- M23 Spare
- M24 Spare

#### P5 Contains:

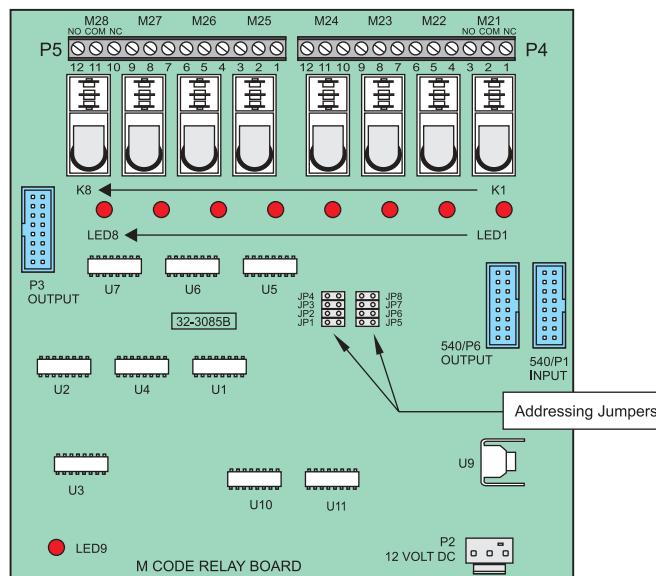
- M25 Spare
- M26 Spare
- M27 Spare
- M28 Spare

P1 16-Pin Relay Drivers From IOPCB (M21- M28) (540) (INPUT)

P3 Unused

P2 12 VDC From Power Supply Board (860A)

P6 16-Pin Relay Output To 2nd 8M Relay Board



*M-Code relay board.*

---

**NOTE:** If the 8M option is installed, relays M21-28 become available on the secondary board. These relays will be controlled by outputs M21-28.

### 3.11 SUBROUTINES

One of the more important programming features of a CNC machine is called subroutines. Subroutines allow the CNC programmer to define a series of commands which might be repeated several times in a program and, instead of repeating them many times, they can be “called.” A subroutine call is done with M97 or M98 and a **Pnnnn**. The **P** code is the same as the **O** number of the subroutine to be called.

It is important to note that there is little difference between the main program and the subroutines. In the LIST PROG display, they all appear as numbered programs. When starting execution of a program, the LIST PROG display is used to select the **main** program and any subroutines used are called from within the main program.

Local subroutines can be used with M97. This can be even easier to use than M98 because the subroutine is part of a single main program, without the need to define a different **Onnnnn** program. With local subroutines, you can code an M30 for the end of your main program, followed by a line number and a subroutine that ends with an M99.

The subroutine call causes the blocks in the subroutine to be executed just as if they were included in the main program. In order to return control to the main program, subroutines must end with an M99.

Another very important feature of subroutines is that the M98 “call” block may also include an **L** or repeat count. If there is an **L**, the subroutine call 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 must first be center-drilled, then peck-drilled, tapped, and chamfered. If a subroutine is defined that consists only of the X-Y position of the holes, the main program can call that subroutine after defining a canned cycle to do each of the operations. Thus, the X-Y positions can be used several times and need not be repeated for each tool. An example follows:

O0100	(MAIN PROGRAM FOR EXAMPLE OF SUBROUTINES) ;
G54 G00 G90 X0. Y0. ;	
T01 M06	(CENTER DRILL) ;
G81 R0.2 Z-0.1 F20. L0	(NO OPERATION HERE, JUST DEFINE CANNED CYCLE) ;
S2000 M03 ;	
M98 P0200	(CENTER DRILL EACH HOLE) ;
T02 M06	(PECK DRILL) ;
G83 R0.2 Z-1. F10. L0	(NO OPERATION HERE, JUST DEFINE CANNED CYCLE) ;
S1000 M03 ;	
M98 P0200	(PECK DRILL EACH HOLE) ;
T03 M06	(TAP IN FLOATING HOLDER OR HARD TAP) ;
G84 R0.2 Z-1. F10. L0	(NO OPERATION HERE, JUST DEFINE CANNED CYCLE) ;
S200	(1/4-20) ;
M98 P0200	(TAP EACH HOLE) ;
T04 M06	(CHAMFER) ;
G81 R0.2 Z-0.1 F20. L0	(NO OPERATION HERE, JUST DEFINE CANNED CYCLE) ;
S2000 M03 ;	
M98 P0200	(CHAMFER EACH HOLE) ;
G28 M30	(END OF MAIN PROGRAM) ;
O0200	(SUBROUTINE EXAMPLE LISTING ALL HOLE POSITIONS) ;



VS-Series

X0. Y0. ;	
X1. Y0. ;	
X2. Y0. ;	
X0. Y1. ;	
X1. Y1. ;	
X2. Y1. ;	
X0. Y2. ;	
X1. Y2. ;	
X2. Y2. ;	
M99	(END OF SUBROUTINE) ;
O0300	(EXAMPLE USING A LOCAL SUBROUTINE)
G54 G00 G90 X0. Y0. ;	
G81 R0.2 Z-0.1 F20 L0	(NO OPERATION HERE, JUST DEFINE CANNED CYCLE);
S2000 M03;	
M97 P0500	(CENTER DRILL EACH HOLE);
T02 M06	(PECK DRILL);
G83 R0.2 Z-1. F10. L0	(NO OPERATION HERE, JUST DEFINE CANNED CYCLE);
S1000 M03;	
M97 P0500	(PECK DRILL EACH HOLE);
G28 M30	(END OF MAIN PROGRAM);
N0500	(LOCAL SUBROUTINE EXAMPLE LISTING ALL HOLE POSITIONS);
X0. Y0. ;	
X1. Y0. ;	
X2. Y0. ;	
X0. Y1. ;	
X1. Y1. ;	
X2. Y1. ;	
X0. Y2. ;	
X1. Y2. ;	
X2. Y2. ;	
M99	(END OF SUBROUTINE);

### *SUBPROGRAMS AND CANNED CYCLES*

After reviewing the canned cycle, we can get a good idea of the amount of lines of code required to produce the five holes. The best way to conserve on program space and programming time is to use a subprogram. We can do this by grouping the **X** and **Y** locations of the holes into a separate program and then calling up this program when we need to tell a canned cycle the X,Y coordinates.

Instead of writing the X,Y locations once for each tool, we can write the X,Y locations once for any number of tools.

The canned cycle program that we reviewed on the previous page could use some constructive rearranging.

%	%
O1234 (Example program)	O1000 (X,Y LOC. SUB)
T1 M06	X 1.115 Y-2.750
G90 G54 G00 X.565 Y-1.875 S1275 M03	X 3.365 Y-2.875
G43 H01 Z.1 M08	X 4.188 Y-3.313
G82 Z-.175 P.03 R.1 F10.	X 5.0 Y-4.0
M98 P1000	M99
G80 G00 Z1.0 M09	%
T2 M06	
G00 G90 G54 X.565 Y-1.875 S2500 M03	
G43 H02 Z.1 M08	
G83 Z-.720 Q.175 R.1 F15.	
M98 P1000	
G00 G80 Z1.0 M09	
T3 M06	
G00 G90 G54 X.565 Y-1.875 S900 M03	
G43 H03 Z.2 M08	
G84 Z-.600 R.2 F56.25	
M98 P1000	
G80 G00 Z1.0 M09	
G28 G91 Y0 Z0	
M30	
%	

**SUBPROGRAMS WITH MULTIPLE FIXTURES**

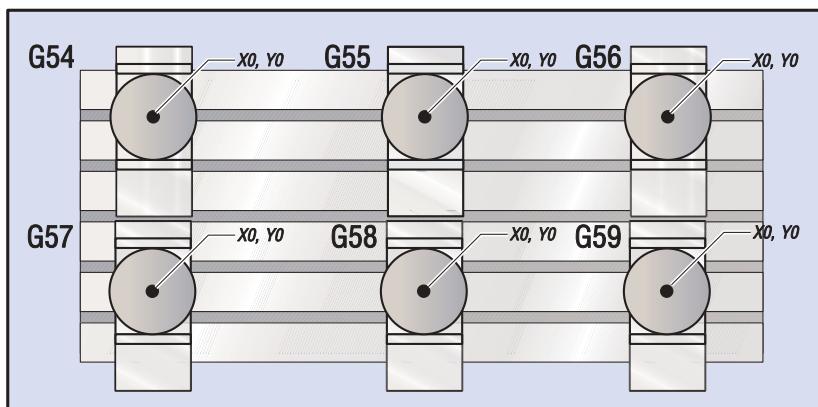
So far we have learned that using subprograms with canned cycles can save programming time and help reduce coordinate input error. Let's take this one step further. There are six vises mounted on the table. Each of these vises will use a new X, Y zero. They will be called up in the program as G54 through G59. The machine will have to be told where each of the vises is located on the table. By using an edge finder or an indicator, the zero point on each part can be established. Use the part zero set key in the work coordinate offset page to record each X, Y location. Once the X, Y zero position for each vise is in the offset page, the programming can begin.

By looking at the next page, we can get a good idea of what this setup would look like on the machine table.

For an example, each of these six parts will need to be drilled at the center, **X** and **Y** zero.

%	%
O2000	O3000
T1 M06	X0 Y0
G00 G90 G54 X0 Y0 S1500 M03	G83 Z-1.0 Q.2 R.1 F15.
G43 H01 Z.1 M08	G00 G80 Z.2
M98 P3000	M99
G55	%
M98 P3000	
G56	
M98 P3000	
G57	
M98 P3000	
G58	
M98 P3000	
G59	
M98 P3000	
G00 Z1.0 M09	
G28 G91 Y0 Z0	
M30	
%	

The following figure represents a multiple-fixture setup. Each vise will have an absolute zero once it is specified in the program. This is done by using G54 through G59 and G110 through G129, a total of 26 possible positions.

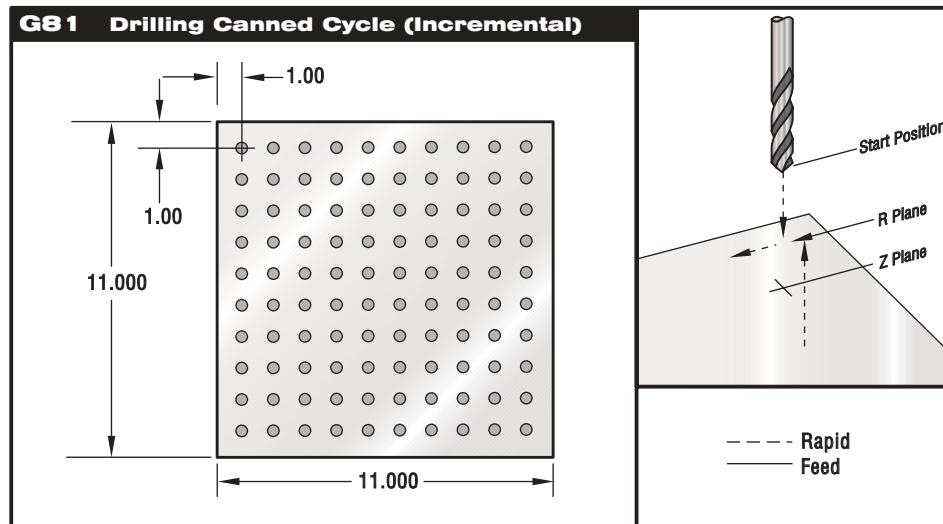


*Multiple fixture setup.*

## Looping Canned Cycles

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

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



Grid plate for multiple-fixture subprogram exercise.

### PROGRAM EXAMPLE

```
%  
O3400  
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  
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  
%
```

### DESCRIPTION

(Drilling grid plate)

(Or stay in G91 and repeat Y-1.0)



## MODIFYING CANNED CYCLES

In this section we will cover canned cycles that have to be customized in order to make the programming of difficult parts easier. In result, making the machining process more efficient.

Using G98 and G99 to clear clamps:

For example, we have a square part being held to the table with one inch tall table clamps. We need to write a program to clear the table clamps.

### PROGRAM EXAMPLE

```
%  
O4500  
T1 M06  
G00 G90 G54 X1.0 Y-1.0 S3500 M03  
G43 H01 Z1.125 M08  
G81 G99 Z-1.500 R.05 F20.  
X2.0 G98  
X6.0 G99  
X8.0  
X10.0  
X12.0 G98  
X16.0 G99  
X18.0 G98  
G00 G80 Z2.0 M09  
G28 G91 Y0 Z0  
M30  
%
```

### DESCRIPTION

( Will return to starting point after executing cycle )  
( Will return to reference plane after executing cycle )

### **X,Y Plane Obstacle Avoidance In A Canned Cycle:**

So far we have learned how G98 and G99 can be used to avoid an obstacle in the Z-axis. There is also a way to avoid an obstacle in the X,Y plane during a canned cycle by placing an L0 in a canned cycle line, we can tell the control to make an X,Y move without executing the Z-axis canned operation.

For example, we have a six inch square aluminum block, with a one inch by one inch deep flange on each side. The print calls for two holes centered on each side of the flange. We need to write a program to avoid each of the corners on the block.

## PROGRAM EXAMPLE

```
%  
O4600  
(X0,Y0 is at the top left corner)  
(Z0 is at the top of the part)  
T1 M06  
G00 G90 G54 X2.0 Y-.5 S3500 M03  
G43 H01 Z-.9 M08  
G81 Z-2.0 R-.9 F15.  
X4.0  
X5.5 L0  
Y-2.0  
Y-4.0  
Y-5.5 L0  
X4.0  
X2.0  
X.5 L0  
Y-4.0  
Y-2.0  
G00 G80 Z1.0 M09  
G28 G91 Y0 Z0  
M30  
%
```

## DESCRIPTION

(angular corner avoidance)

## SPECIAL CANNED CYCLES

In this section, we will cover the special canned cycles that the Haas control offers. These canned cycles are used in conjunction with other drilling, boring, and tapping cycles.

### **G70 = BOLT HOLE CIRCLE**

### **G71 = BOLT HOLE ARC**

### **G72 = BOLT HOLES ALONG AN ANGLE**

The sample program below will show the format for using a G70 to drill a three inch diameter bolt hole pattern combined with a G81 drilling canned cycle.

```
%  
O5000  
T1 M06  
G00 G90 G54 X0 Y0 S1500 M03  
G43 H01 Z.1 M08  
G70 I1.5 J0 L8 G81 Z-1.0 R.1 F15.  
G00 G80 Z1.0 M09  
G28 G91 Y0 Z0  
M30  
%
```

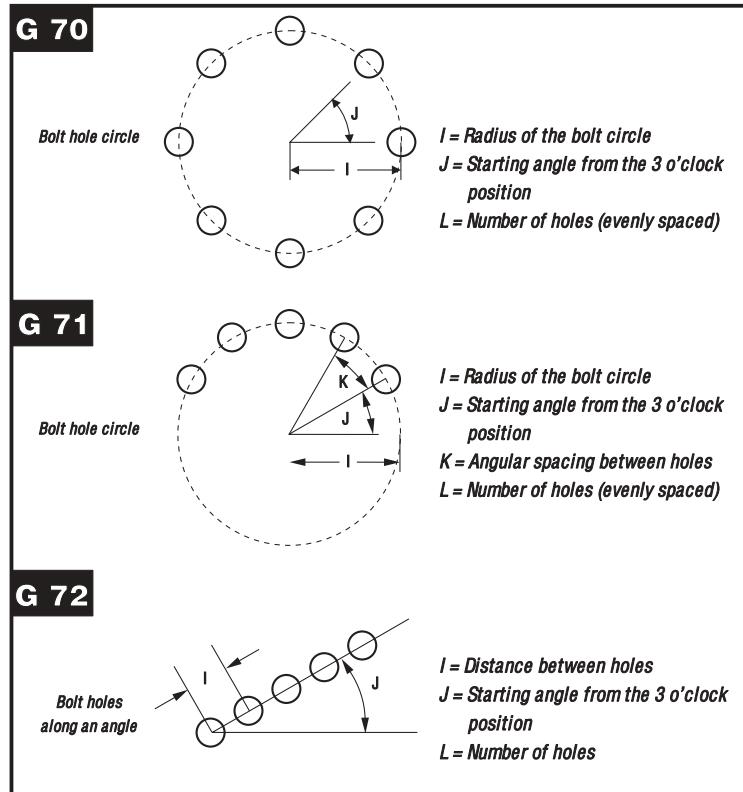
### **RULES FOR BOLT PATTERN CANNED CYCLES:**

1. The tool must be placed at the center of the bolt pattern before the canned cycle execution. The center is usually X0, Y0.
2. The **J** code is the angular starting position and is always 0 to 360 degrees counterclockwise from the three o'clock position.

In the case of conflicting address codes, You can specify a drilling cycle prior to the block that invokes the special canned cycle. For instance:



VS-Series



#### Bolt Hole Pattern Canned Cycles

%  
05000  
T1 M06  
G00 G90 G54 X0 Y0 S1500 M03  
G43 H01 Z.1 M08  
G83 R.1 Z-1.0 I.25 J.03 K.15 F15. L0 (L0 PREVENTS DRILLING AT CENTER)  
G70 I1.5 J0 L8  
G00 G80 Z1. M09  
G28 G91 Y0 Z0  
M30  
%

### 3.12 TOOL FUNCTIONS (*TNN*)

#### *TNN CODE*

The **Tnn** code is used to select the next tool to be placed in the spindle from the tool changer. The **T** address does not start the tool change operation; it only selects which tool will be used next. M06 and M16 are used to start a tool change operation. The **Tnn** does not need to be in a block prior to the M06 or M16; it can be in the same block.

---

**NOTE:** There is no **X** or **Y** motion required prior to performing a tool change and it would waste time in most cases to return **X** or **Y** to the home position. However, if your work piece or fixture is quite large, you may need to position **X** or **Y** prior to a tool change in order to prevent a crash between the tools and your fixture.

A tool change may be commanded with **X**, **Y**, and **Z** in any position and the control will bring the **Z** up to the machine zero position prior to starting the tool change. The control will move the **Z** axis to a position above machine zero during a tool change, but will never move below machine zero. At the end of a tool change, the **Z** axis will be at machine zero.

Tools are always loaded through the spindle and should never be installed directly in the carousel in order to avoid crashes.

The tool holders used are CT #50 taper, V flange, commonly called "CT 50."

Toolholders and pull studs must be in good condition and tightened together with wrenches or they may stick in the spindle taper. Clean the tool tapers with a lightly oiled rag to leave a film to prevent rusting. When the TOOL RELEASE button is pressed the tool should be pushed out of the spindle by a small amount (approximately 0.07 Inch). This is an indication that the pull stud is correctly touching the release mechanism.

#### WARNING

**Follow these guide lines to load the tool changer:**  
**Maximum tool diameter 4"**  
**Maximum tool diameter if declared over size tool 10"**  
**Maximum tool length 16"**  
**Maximum tool weight 30 lbs.**



VS-Series

## Tool Holders/Pull Studs

CT CAT V-Flange						40T	20-7594 (TSC)	5/8-11 Inch Threads JMTBA Standard MAS 403 P40T-1	0.990 Thru. 45°	0.172 Ø Thru.	Kit # TPS24CT
A	B	C	D	E							
40T	2.69	2.50	.44	5/8"-11	1.75		20-7164 (non-TSC)	5/8-11 Inch Threads JMTBA Standard MAS 403 P40T-1	0.990 Thru. 45°	0.172 Ø Thru.	Kit # PS24CT
							22-0075 (TSC)	1"-8 Inch Threads JMTBA Standard MAS 403 P50T-1	1.386 Thru. 45°	1.780 Ø 0.31	Kit # TPS24CT50
50T	4.00	3.87	.44	1"-8	2.75		22-0039 (non-TSC)	1"-8 Inch Threads JMTBA Standard MAS 403 P50T-1	1.386 Thru. 45°	1.780 Ø 0.31	Kit # PS24CT50

BT MAS 403						40T	20-7595 (TSC)	M16 X 2 Threads JMTBA Standard MAS 403 P40T-1	1.104 Thru. 45°	0.172 Ø Thru.	Kit # TPS24BT
A	B	C	D	E							
40T	2.57	2.48	.65	M16X2	1.75		20-7165 (non-TSC)	M16 X 2 Threads JMTBA Standard MAS 403 P40T-1	1.104 Thru. 45°	0.172 Ø Thru.	Kit # PS24BT
							22-7171 (TSC)	M24 X 3 Threads JMTBA Standard MAS 403 P50T-1	1.386 Thru. 45°	1.780 Ø 0.31	Kit # TPS24E50
50T	4.00	3.94	.91	M24X3	2.75		22-7170 (non-TSC)	M24 X 3 Threads JMTBA Standard MAS 403 P50T-1	1.386 Thru. 45°	1.780 Ø 0.31	Kit # PS24E50

DIN-69871 (MIKRON) ISO-7388						40T	20-7556 (TSC)	M16 X 2 Threads JMTBA Standard MAS 403 P40T-1	0.990 Thru. 45°	0.172 Ø Thru.	Kit # TPS24E
A	B	C	D	E							
40T	2.69	2.50	.44	M16X2	1.75		20-7164 (non-TSC)	M16 X 2 Threads JMTBA Standard MAS 403 P40T-1	0.990 Thru. 45°	0.172 Ø Thru.	Kit # PS24E
							22-7171 (TSC)	M24 X 3 Threads JMTBA Standard MAS 403 P50T-1	1.386 Thru. 45°	1.780 Ø 0.31	Kit # TPS24E50
50T	4.00	3.84	.44	M24X3	2.75		22-7170 (non-TSC)	M24 X 3 Threads JMTBA Standard MAS 403 P50T-1	1.386 Thru. 45°	1.780 Ø 0.31	Kit # PS24E50

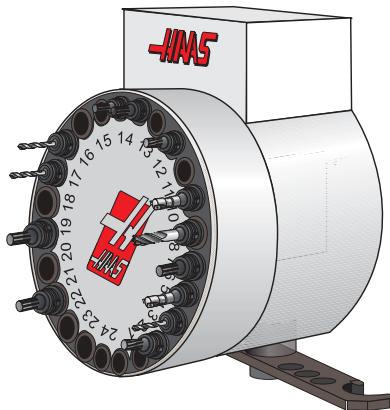
NOTE: CT 40T Pullstud = One Identification Groove  
 BT 40T Pullstud = Two Identification Grooves  
 MIKRON 40T Pullstud = Three Identification Grooves

The tool changer is protected by fuse FU5, located on the POWER PCB. It might be blown by an overload or jam of the tool changer. Operation of the tool changer can also be interrupted by problems with the tool clamp/unclamp and the spindle orientation mechanism.

There are some other **M** codes which will also cause tool operations to occur:

- M19 Will orient the tool for special user functions
- M39 Will rotate the tool turret without changing tools (be careful of crashes)
- M82 Will unclamp the tool (be careful, it will fall!)
- M86 Will clamp the tool

### 3.13 SIDE MOUNT TOOL CHANGER



**NOTE:** On mills equipped with a side-mount tool changer, air pressure is checked prior to moving the carousel. Alarm 120, LOW AIR PRESSURE, is generated if such a problem exists.

**CAUTION:** Keep clear of the tool changer during power up, power down, and any tool changer operations.

#### POWER UP PROCEDURE

1. Press the green **POWER ON** button and wait for the control to complete the boot-up process.
2. Press the **POWER UP/RESTART** button and wait for the machine to perform a Zero Return.

Setting 81, TOOL AT POWERDOWN, will cause one of the following actions to occur during power-up:

- A. If Setting 81 is set to zero, the carousel will be rotated to pocket #1. No tool change will be performed.
- B. If Setting 81 contains the tool number of the tool currently in the spindle, the carousel will remain at the same pocket and no tool change will be performed.
- C. If Setting 81 contains the tool number of a tool not currently in the spindle, the carousel will be rotated to pocket #1 and then to the pocket containing the tool specified by Setting 81. A tool change will be performed to change the specified tool into the spindle.



## OVERVIEW

---

**NOTE:** If alarms occur while using a Vertical Mill side-mount tool changer, the associated parameters should first be verified (and corrected if necessary), and then the tool change recovery procedure should be performed.

The side-mount tool changers allow the control to rotate the carousel to the next tool needed while the current tool is still in the spindle. When a tool change is commanded, the two tools are swapped by the tool arm. Thus, a tool will be taken from one pocket and returned to another. This means that pockets designated as normal size are not dedicated to any particular tool; during tool change operations a normal size tool can be taken from one pocket and put back into another. Tool pockets designated as large size are dedicated only to large tools; large tools will not migrate to an empty 'normal' pocket during a tool change.

## TOOL LOADING

---

**NOTE:** A normal size tool has a diameter of less than 3" for 40-taper machines, or less than 4" for 50-taper machines. Tools larger than these measurements are considered large size.

1. Ensure the tools you will be loading are of the correct pull stud type for your mill.
2. Press the **POWR UP/RESTART** button and wait for the machine to perform a Zero Return.
3. Enter Handle Jog Mode.
4. Press the **OFFSET** button. From a Power Up/Restart, press the **END** key, then the **PAGE DOWN** key once to reach the Tool Pocket Table display. From a normal operating state, press Page Up/Down until you reach the Tool Pocket Table.
5. Clear any current Large tool designations. Use the cursor keys to scroll to any tool pocket that has an "L" next to it. Press **SPACE**, then **WRITE/ENTER** to clear the Large tool designations.

---

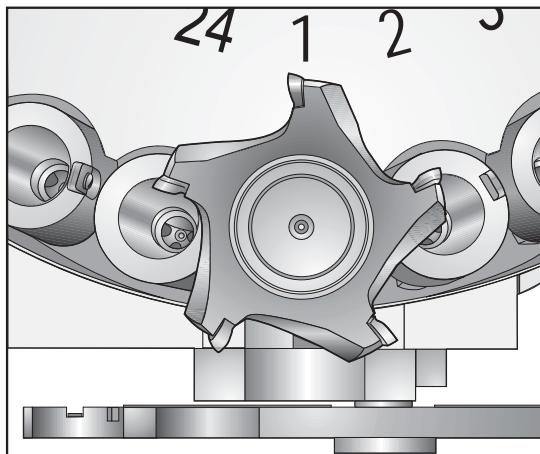
**Note:** The high speed side-mount tool changer has the added ability to designate a tool as "Heavy" in the Pocket-tool table. A "Heavy Tool" is defined as a tool weighing 4 pounds or heavier.

H - Heavy, but not necessarily large (large tools require empty pockets on either side).  
h - Heavy small diameter tool in a pocket designated for a large tool (must have empty pocket on both sides). The lower case "h" is placed by the control; never enter a lower case "h" into the tool table.

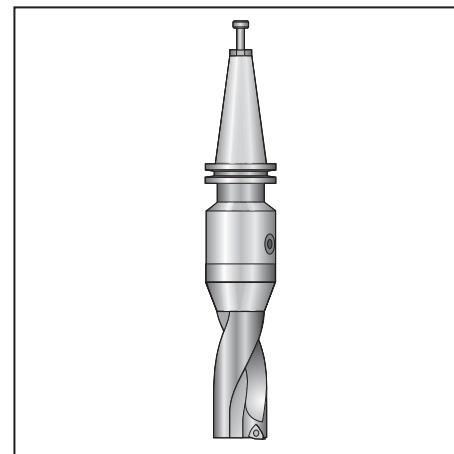
**Large tools are assumed to be heavy.**

**Heavy tools are not assumed to be large.**

On non-high speed tool changers, "H" and "h" have no effect.



*Large (and heavy) tool with surrounding pockets empty*



*Heavy Tool (not large)*

6. Press **ORIGIN** to reset the Tool Pocket Table to default values. This will place tool 1 in the spindle, tool 2 in pocket 1, tool 3 in pocket 2, etc. This is done to clear the previous Tool Pocket Table settings as well as renumber the Tool Pocket Table for the next program. Another way to reset the Tool Pocket Table is to enter 0 (zero) and press Origin, this will reset all the values to zero.

---

**NOTE:** You cannot have two different tool pockets holding the same tool number. Trying to enter a tool number already displayed in the Tool Pocket Table will result in an "Invalid Number" error.

7. Determine if your next program will need any large tools. A large tool has a diameter of greater than 3" for 40-taper machines and greater than 4" for 50-taper machines. If you will not be using any large tools, proceed to Step 10. If you will be using large tools, proceed to the next step.
8. Organize your tools to match your CNC program. Determine the numerical positions of any large tools and designate those pockets as Large in the Tool Pocket Table. To designate a tool pocket as Large, scroll to that pocket and press L, then **WRITE/ENTER**.

---

**NOTE:** When setting up the tooling for your CNC program, large tools need to have the surrounding pockets empty to prevent a tool changer crash. However, large tools can share adjoining empty pockets. Placing large tools next to each other will help conserve space in the tool changer.

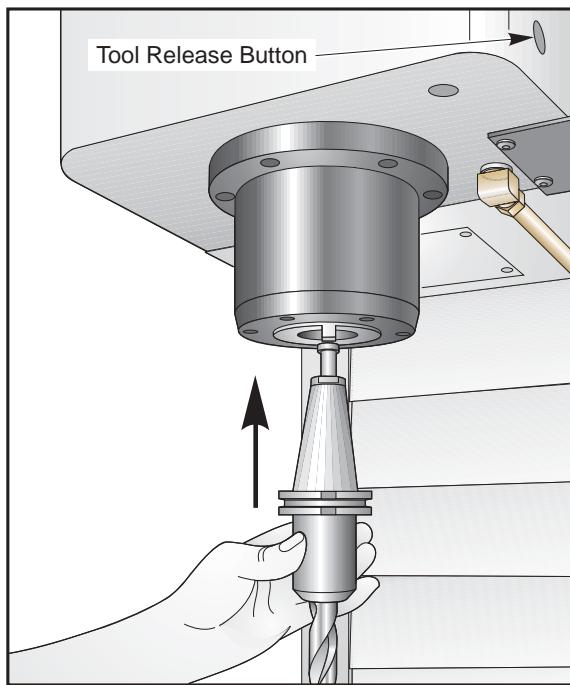
---

**CAUTION:** A large tool cannot be placed in the tool changer if one or both of the surrounding pockets already contain tools. Proceeding with this action will cause the tool changer to crash.

---

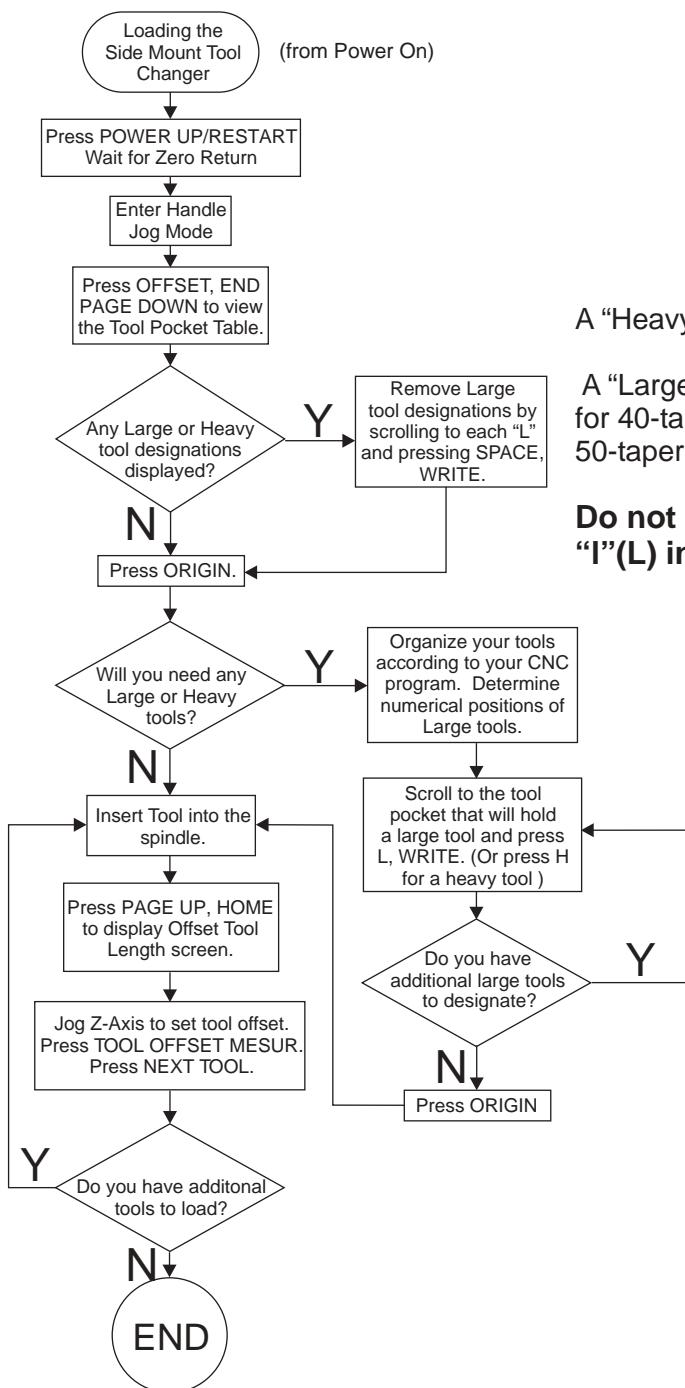


9. When you have designated all necessary Large tool pockets, press the **ORIGIN** key to renumber the Tool Pocket Table. At this point, the machine is ready to accept tool 1 into the spindle.
10. Take tool 1 in your hand and insert the tool (pull stud first) into the spindle. Turn the tool so that the two cutouts in the tool line up with the dogs of the spindle. Push the tool upward while pressing the **TOOL RELEASE** button. When the tool is fitted into the spindle, let go of the **TOOL RELEASE** button. One **TOOL RELEASE** button is located on the right side of the head cover just above the spindle; a second one is located on the keypad.



11. Press **PAGE UP**, then **HOME** to go to the Offset Tool Length display.
12. Jog the Z axis to set the tool offset. Press the **TOOL OFFSET MEASUR** (measure) button on the keypad. Press the **NEXT TOOL** button.
13. Repeat Steps 10 through 12 until you have loaded all your tools.

## TOOL LOADING FLOWCHART



A “Heavy Tool” is a tool weighing 4 or more pounds.

A “Large Tool” has a diameter greater than 3" for 40-taper mills, or greater than 4" for 50-taper mills.

**Do not enter a lower case “h” or lower case “l”(L) in the tool table.**



VS-Series

## MOVING TOOLS IN THE CAROUSEL

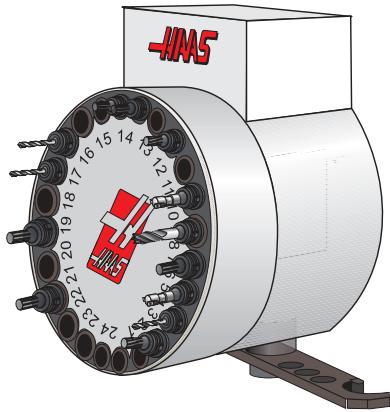
Should you need to move tools around in the carousel, follow the steps below.

### CAUTION:

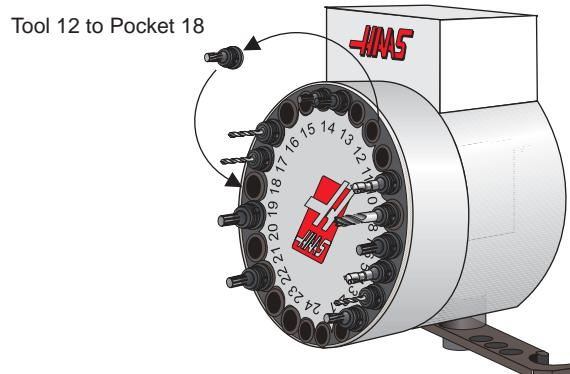
Plan out your reorganization of the tools in the carousel ahead of time. To reduce the potential for tool changer crashes, keep your tool movement to a minimum. If you have any large size tools currently in the tool changer, ensure that you only move them between tool pockets designated as large.

### Creating Room for a Large Size Tool

The tool changer pictured below has an assortment of normal size tools. For the purposes of this example, tool 12 will be moved to pocket 18 to create room for a large size tool to be placed in pocket 12.



1. Select MDI mode. Press the **OFFSET** button. Press **PAGE UP/PAGE DOWN** (if necessary) until you reach the Tool Pocket Table display. Verify which tool number is in pocket 12.
2. Enter **Tnn** into the control (where **Tnn** the tool number from step 1). Press **ATC FWD**. This will place the tool from pocket 12 into the spindle.
3. Enter **P18** into the control, then press **ATC FWD**. This will place the tool currently in the spindle into pocket 18.

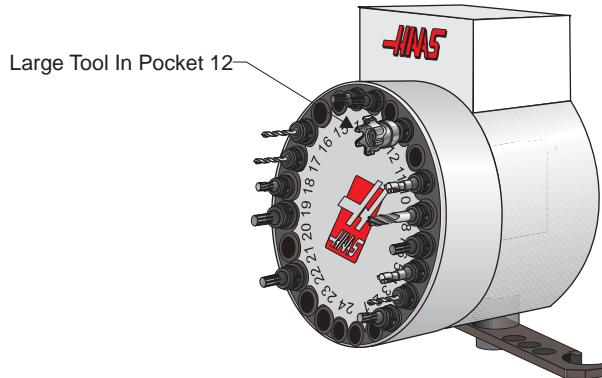


4. Scroll to pocket 12 in the Tool Pocket Table and press L, **WRITE/ENTER** to designate that pocket as Large.
5. Enter the tool number into SPNDL (spindle) on the Tool Pocket Table. Insert the tool into the spindle.

---

**NOTE:** You cannot have two different tool pockets holding the same tool number. Trying to enter a tool number already displayed in the Tool Pocket table will result in an "Invalid Number" error.

6. Enter P12 into the control and press **ATC FWD**. The tool will be placed into pocket 12.



#### POWER DOWN PROCEDURE

1. Assuming the machine has been zeroed since it was powered up, press the **POWER UP/RE-START** button and wait for the machine to perform a Zero Return.

Setting 81, TOOL AT POWERDOWN, will cause one of the following actions to occur during power down:

- A. If Setting 81 is set to zero the carousel will remain at the same pocket. No tool change will be performed.
- B. If Setting 81 contains the number of the tool that is already in the spindle, the carousel will remain at the same pocket and no tool change will be performed.
- C. If Setting 81 contains the number of a tool that is **not** currently in the spindle, the carousel will be rotated to pocket #1, then to the pocket containing the tool specified by Setting 81. A tool change will be performed that places the specified tool in the spindle.
- D. If Setting 81 contains the number of a tool not listed in the Tool Pocket Table, the carousel will be rotated to pocket #1 and no tool change will be performed.

2. Press the red **POWER OFF** button.



## TOOL CHANGER RECOVERY

If a problem occurred during a tool change, a tool changer recovery needs to be performed. Enter the tool changer recovery mode by pressing **TOOL CHANGER RESTORE**. Once in tool changer recovery mode, the operator is given instructions and questions to perform a proper tool changer recovery. The entire tool changer recovery process must be completed before exiting. If an exit was done prematurely the tool changer recovery must be started from the beginning.

The flow chart for the tool changer restore function is at the end of this section.

## FEATURES

### POCKET AND TOOL Screen

This screen has been added to support the Disk type tool changers. When a side-mount tool changer has been specified by parameter 278 bits, it will appear in the Offsets display screens (press **PAGE UP** or **PAGE DOWN** to scroll to it). It lists the tool carousel pockets and the tools that they each contain, and the tool that is currently in the spindle.

## PARAMETERS

### Parameter 278 bit 27 (S MNT BIT 1), 28 (S MNT BIT 2), and 29 (MNT BIT 3)

Bits 27,28, and 29 work together to enable the Vertical mill side-mount tool changers.

The following table shows the bit combinations that must be used:

Bit 27	28	29	Numeric value	Type
0	0	0	0	No side-mount tool changer installed
1	0	0	1	Serpentine 1
0	1	0	2	Serpentine 2
1	1	0	3	Serpentine 3
0	0	1	4	Disk 1
1	0	1	5	Disk 2
0	1	1	6	Disk 3
1	1	1	7	Disk 4

### Parameter 60 TURRET START DELAY

On mills with the Disk type 1 tool changer, this parameter is used to specify the time (in milliseconds) allowed for the carousel to move off the TC mark. If the carousel has not moved within the time allowed by this parameter, alarm 625, INVALID TC START CONDITION, is generated.

### Parameter 61 TURRET STOP DELAY

On mills with the Disk type 1 tool changer, this parameter is used to specify the time (in milliseconds) allowed for the carousel to move to the TC mark. If the carousel has not moved within the time allowed by this parameter, alarm 625, INVALID TC START CONDITION, is generated.

### Parameter 62 SHUTTLE START DELAY

On mills with a Disk type 1 tool changer, this parameter is used to specify the time (in milliseconds) needed to allow the tool pocket to settle (stop bouncing) after being lowered in preparation for a tool change.

### Parameter 63 SHUTTLE STOP DELAY

The parameter specifies the maximum delay time (in milliseconds) allowed during motion of the tool shuttle. After this time an alarm is generated.

#### Parameter 248 SMTC RLY ON/OFF DLY

On mills with a Disk type 1 tool changer, this specifies the time needed (in milliseconds) between turning off one relay and turning on the other one when reversing direction of the carousel. For vertical mills without a side-mount tool changer, this parameter should be set to 0.

#### Parameter 306 POCKET UP/DOWN DELAY

This parameter supports the Disk type 1 tool changers. It specifies the time allowed (in milliseconds) for the tool pocket to be raised or lowered. If the pocket does not move to its commanded position within the time allowed by this parameter and parameter 62, then alarm 626, TOOL POCKET SLIDE ERROR, is generated. For vertical mills without a side-mount tool changer, this parameter should be set to 0.

#### Parameter 307 POCKT UN/LOCK DELAY

This parameter supports the serpentine type 1 tool changer. It specifies the time needed (in milliseconds) to lock or unlock a tool pocket. For vertical mills without a side-mount tool changer, this parameter should be set to 0.

#### Parameter 308 ARM ROTATE TIME

This parameter supports the Disk type 1 tool changers. It specifies the time allowed (in milliseconds) for the tool arm to rotate to the next position. The positions are: Clamp, Unclamp and Origin. If the arm does not move to the commanded position within the allowed time, alarm 622, TOOLARM FAULT, is generated. For vertical mills without a side-mount tool changer, this parameter should be set to 0.

#### Parameter 309 MOTOR COAST TIME

This parameter supports the Disk type 1 tool changers. It specifies the time allowed (in milliseconds) for the tool arm motor to start moving after being commanded to do so, and to stop moving after being commanded to do so. In either case, if the arm has not responded after the allowed time, alarm 627, ATC ARM POSITION TIMEOUT, is generated. For vertical mills without a side-mount tool changer, this parameter should be set to 0.

#### Parameter 310 CAM LOCK DELAY

This parameter supports the side-mount tool changers. It specifies the time allowed (in milliseconds) to lock the cam by pushing the shot pin in, or to unlock the cam by pulling the shot pin out. If the shot pin has not moved to its commanded position within the allowed time, alarm 625, INVALID TC START CONDITION, is generated.

#### Parameter 311 ARM BUMP TIME

This parameter supports the Disk type 1 tool changers. During tool change recovery, the arm may be moved a small amount by pressing the **ATC FWD** or **ATC REV** key. Each press of the key will cause the arm motors to run for the amount of time (in milliseconds) specified by this parameter. For vertical mills without a side-mount tool changer, this parameter should be set to 0.

#### Parameter 312 CAROUSEL BUMP TIME

This parameter supports the Disk type 1 tool changers. During tool change recovery, the carousel may be moved a small amount by pressing the **ATC FWD** or **ATC REV** key. Each press of the key will cause the carousel motors to run for the amount of time (in milliseconds) specified by this parameter. For vertical mills without a side-mount tool changer, this parameter should be set to 0.



## ALARMS

### Alarm 622 TOOLARM FAULT

This alarm supports the Disk type 1 tool changers. It is generated if the arm is not at the Origin position, or the arm motor is already on when a tool change process is started.

### Alarm 623 SIDE MOUNT CAROUSEL ERROR

This alarm supports the Disk type 1 tool changers. It is generated if the carousel motor is still on when the tool pocket is unlocked and lowered prior to a tool change.

### Alarm 624 INVALID TOOL

This alarm supports the Disk type 1 tool changers. It is generated if the tool specified by the G-code program is not found in the Tool Pocket Table.

### Alarm 625 INVALID TC START CONDITION

This alarm supports the Disk type 1 tool changers. It is generated if conditions are not correct when a tool change is started. Incorrect conditions include:

- Tool not found in Tool Pocket Table
- Unable to move carousel to specified tool
- Tool carousel not at TC mark (Disk type 1)
- Arm not in Origin position
- Arm motor already on.

### Alarm 626 TOOL POCKET SLIDE ERROR

This alarm supports the Disk type 1 tool changers. It is generated if the tool pocket has not moved to its commanded position (and settled) within the time allowed by parameters 306 and 62. This alarm can also be generated if the tool changer is in an invalid position at power-up, or the start of a gear change or tool change.

### Alarm 627 ATC ARM POSITION TIMEOUT

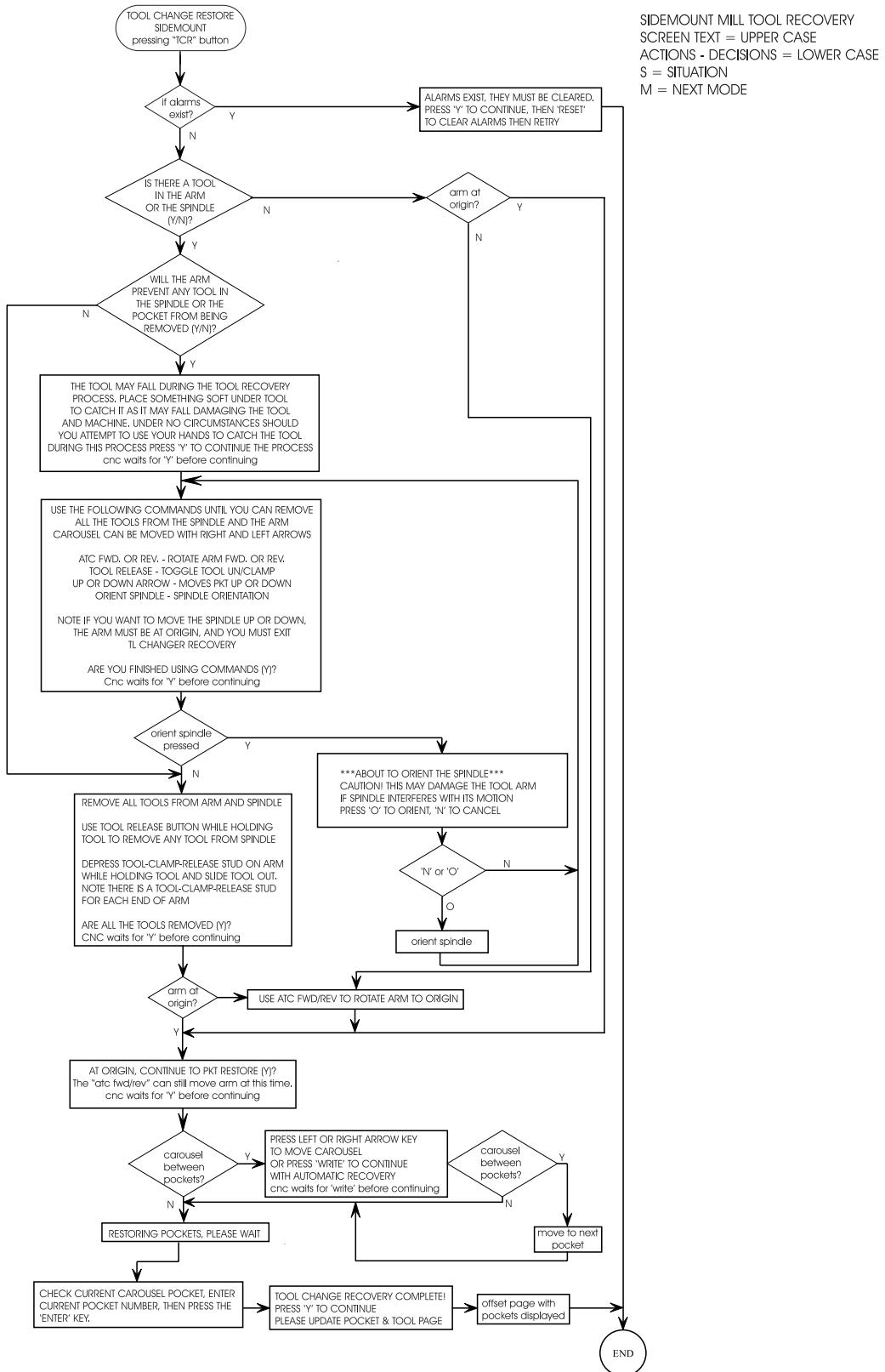
This alarm supports the Disk type 1 tool changers. It is generated if the tool arm has not moved after the allowed time or has not stopped after the allowed time. Refer to Parameter 309, MOTOR COAST TIME.

### Alarm 628 ATC ARM POSITIONING ERROR

This alarm supports the Disk type 1 tool changers. It is generated if:

- The arm was being moved from the ORIGIN position to the CLAMP position and it coasted past the MOTOR STOP point or could not get to the CLAMP point.
- The arm was being moved from the CLAMP position to the UNCLAMP position and it coasted past the MOTOR STOP point or could not get to the UNCLAMP point (same physical point as CLAMP).
- The arm was being moved back to the ORIGIN position and it coasted past the MOTOR STOP point or could not get to the ORIGIN point.

## SIDE MOUNT TOOL CHANGER RECOVERY





### 3.14 SPINDLE SPEED FUNCTIONS

#### SPINDLE SPEED COMMANDS

Spindle speed functions are controlled primarily by the **S** address code. The **S** address specifies RPM in integer values from 1 to maximum spindle speed (Parameter 131, NOT TO BE CHANGED BY USER!).

Speeds from S1 to the Parameter 142 value (High/ Low Gear Change; usually 1200) will automatically select low gear and speeds above Parameter 142 will select high gear. Two **M** codes can be used to override the gear selection. M41 for low gear override and M42 for high gear override. Low gear operation above S1250 is not recommended. High gear operation below S100 may lack torque or speed accuracy.

Three **M** codes are used to start and stop the spindle. M03 starts the spindle clockwise, M04 starts the spindle counterclockwise, and M05 stops the spindle.

Note that only one **M** code is allowed in a block. This means that if you wish to override the gear with M41 or M42, you must put the **Snnnn** and M41 (or M42) in one block and the M03 (or M04) in the next block. The **Snnnn** should always be in the same block as the M41 or M42, as an unneeded double gear change might otherwise be performed.

### 3.15 TAPPING WITH THE VS-SERIES CNC MILL

Making tapped holes with the VS Series CNC Mill can be done with several devices. Threads may be generated with a tap held in a rigid tool holder (called rigid tapping), a floating tap holder, a reversing tapping head, or helical thread milling. Each method has distinct advantages.

Tapping is done using canned cycles. You must select the tapping rpm and, using the pitch (threads per inch), calculate the feedrate that is entered in the **F** command. The HELP/CALC page will compute these numbers for you.

#### RIGID TAPPING

Rigid tapping eliminates the cost of special tap holders since taps can be held in drill collet holders. The spindle is accurately synchronized with the Z-axis feed, thereby producing threads as accurately as a lead screw tapper. No side forces are generated on the flanks of the threads and tighter thread tolerances are produced. Rigid tapping also eliminates the pullout and distortion of the first thread that occurs on all spring compression/tension devices and tapping heads. While this is not usually a problem on medium to coarse threads, small diameter, fine pitch or soft material tapped holes can have their last thread damaged when the tap pops out of the hole. You can also re-tap a hole without cross-threading, provided the tap and **Z** depths have not been changed. Rigid tapping is used with canned cycles G74 and G84. Example:

N100 G84 Z-1. R.3 F37.5 (for a 20-pitch thread at 750 rpm)

A word of caution on rigid tapping: As the term implies, the tap is rigidly held in place. This requires that runout be less than 0.001 TIR or the tap will generate an oversized thread. This problem can be minimized by using a small diameter drill extension to provide flex, a radial floating tool holder, or specially-designed chucks for holding taps, because tap shanks are not common collet sizes.

The mill can retract from a tap faster than it went in. The way to specify this is to use a **J** code in the line that commands the tap. J2 retracts twice as fast as the entry motion, J3 retracts three times as fast, and so on, up to J9. A **J** code of zero will be ignored. If a **J** code of less than zero or greater than nine is specified , Alarm 306,

"INVALID I, J, K, or Q" is generated. The **J** code is not modal, and must be specified in each block where the effect is wanted.

Rigid tapping is enabled with the Parameter 57 “Rigid Tap” flag. In addition, if the “REPT RIG TAP” flag in Parameter 57 is set, every repetition of a tapping operation will control the orientation of the spindle so that the tapping is repeatable.

Pitch control is within 0.0005 inch. Bottom depth control is  $\pm 0.020$  inch and repeatability is  $\pm 0.005$  inch. Rigid tapping will operate from 100 to 2000 RPM and up to 100 inches per minute feed. Bottom depth control is better at lower speeds. Thread pitch is limited, from 4 to 100 TPI.

The pitch of a tapped hole is defined by the ratio between the feedrate and spindle speed. When rigid tapping is selected, these two must be set exactly. An encoder mounted with the spindle tracks the position of the spindle and the Z axis is moved precisely to match the pitch of the thread. If the repeatable option is selected, a position pulse from the encoder is used to synchronize the starting of the **Z** motion with the position of the spindle.

Note that with G74 and G84, you do not ever need to use M03, M04, or M05. These canned cycles start and stop the spindle automatically. This applies to using normal or rigid tapping.

The second page of diagnostic data will show the actual spindle speed.

### *USING FLOATING TAP HOLDERS*

Floating tap holders are probably the most common method of tapping holes. The tap is held in a quick change holder that can float up and down slightly. This is done to allow the tap to follow the hole it is tapping and compensate for differences in the acceleration and deceleration of the spindle versus the feed of the Z axis. Upon reaching the bottom of the hole, the feed stops and the spindle reverses; if you watch closely, you will see the tap pull the floating holder out slightly. Upon reversal, the tap will be pushed back into the holder.

If the holder is pulled out or pushed in to its mechanical limits while tapping, you can break the tap, damage the threaded part, or pull the tap completely out of the holder. Carefully watch for this condition when setting up a job, because it usually becomes a problem after the job has been running a while. Also, tapping of diameters less than 5/16 of an inch while below 1201 rpm (low gear shift point parameter 142) should be done in high gear. Spindle reversal is quicker in high gear and will minimize tap pullout. This is done by putting an M42 code with the speed command, such as: M42 S900. Tapping is done with G74 and G84 cycles which automatically reverse the spindle at **Z** depth. The feedrate can be calculated by using the HELP display, paging down to the tapping calculator and inputting your speed and tap pitch into the control to obtain your feedrate, which is then input to the **F** command of the cycle. Example:

N100 G84 Z-1.0 R.3 F46.875 (for a 32-pitch tap at 1500 RPM)

### *AUTOREVERSING TAPPING HEADS*

Auto reversing tapping heads eliminates the need for the spindle to reverse at the bottom and provides for high production rates. The reversing function of the tapping head requires an arm to prevent the body from rotating. This must be considered when changing tools so as not to interfere with operation. The tool block on the VS Series CNC Mill will accommodate the Tapmatic series of heads. Several sizes are available and should be chosen dependent on tap size. Choose the head specifically for NC use as they have a 1:1 feed rate. Manual types have a faster withdrawal rate that leads to clatter on the upstroke. A disadvantage to these types of heads is that when the inevitable crash occurs, you can destroy an expensive device.

Use the G85 or G89 (dwells at bottom) cycle when using a tapping head. Example:

N100 G98 G85 Z-1.0 R0.25 F46.875



## THREAD MILLING

Thread milling uses a cutter formed with the pitch of the thread to mill the thread. The cutters are solid carbide, fragile and expensive. Some companies sell replaceable insert holders that are more economical. Internal holes smaller than 3/8 inch may not be possible or practical. It does allow for making thread diameter compensation and external threads. For large threads, port threads and blind hole threads, thread milling can be the most economical method.

Thread milling is accomplished with helical milling. Use a standard G02 or G03 move to create the circular move in X-Y and then insert a Z move on the same block corresponding to the thread pitch. The feed rate is selected as in standard milling practice. This will generate one turn of the thread. The multiple teeth of the cutter will generate the rest. A typical line would be as follows:

N100 G02 I-1.0 Z-.05 F5. (generates a 1-inch radius for 20-pitch thread)

We will use the following example and go through the thread milling procedures step-by-step to get the desired result:

### DATA:

- I.D. Thread milling a 1.5 x 8 TPI hole.
- Using .750 diameter x 1.0 thread hob.
- Take the hole diameter 1.500. Subtract cutter diameter .750 = .750 Then divide by 2 = .375.

**STEP 1:** Within this space we need to turn on cutter compensation and ramp on to the circle to be machined.

**STEP 2:** Perform complete circle while simultaneously moving in the Z-axis the amount of one full pitch of the thread. This is called **helical interpolation**.

**STEP 3:** Ramp off the circle and turn off the cutter compensation.

---

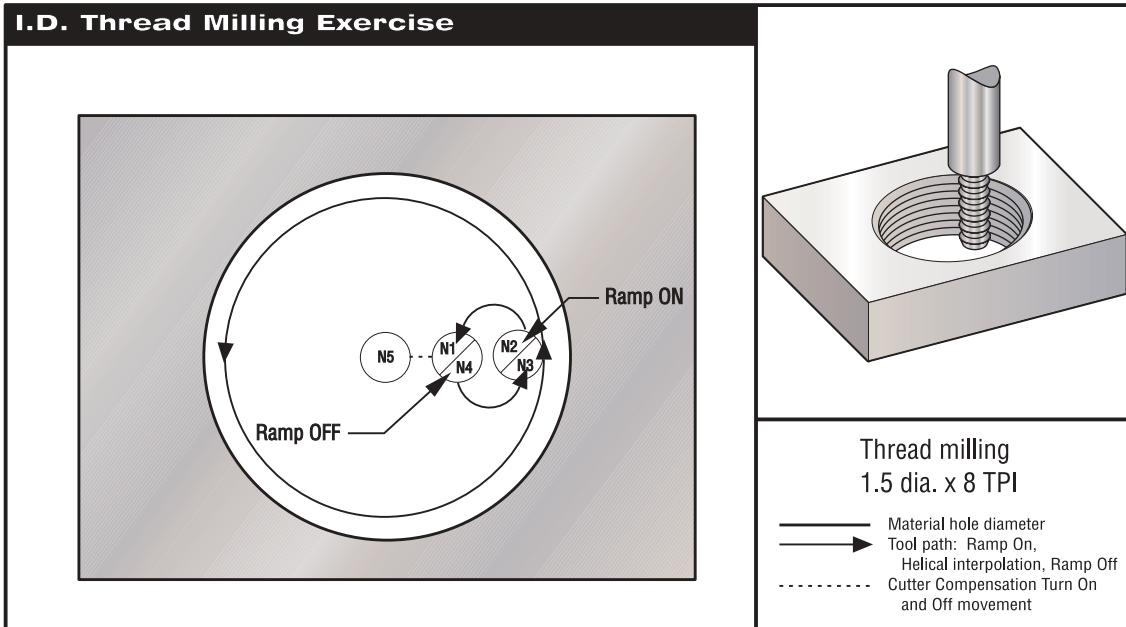
**NOTE:** Always climb cut the cutter.

**I.D.** will be G03; **O.D.** will be G02. 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 = 1.0/Threads per inch

**Example      1.0 divided by 8 TPI = .125**

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

**I.D. Thread Milling Exercise**

**PROGRAM EXAMPLE**

```
%  
O2300  
(X0, Y0 is at the center of the hole)  
(Z0 is at the top of the part)  
(Using .5 thick material)  
G00 G90 G54 X0 Y0 S400 M03  
G43 H01 Z.1 M08  
Z-.6  
N1 G01 G41 D01 X.175 F25.  
N2 G03 X.375 R.100 F7.  
N3 G03 I-.375 Z-.475  
N4 G03 X.175 R.100  
N5 G01 G40 X0 Y0  
G00 Z1.0 M09  
G28 G91 Y0 Z0  
M30  
%
```

**DESCRIPTION**

(Thread milling 1.5 diameter x 8 TPI)

(Turn on Cutter Comp)  
(Ramp on move)  
(One full revolution with Z moving up.125)  
(Ramp off move)  
(Cancel Cutter Comp)

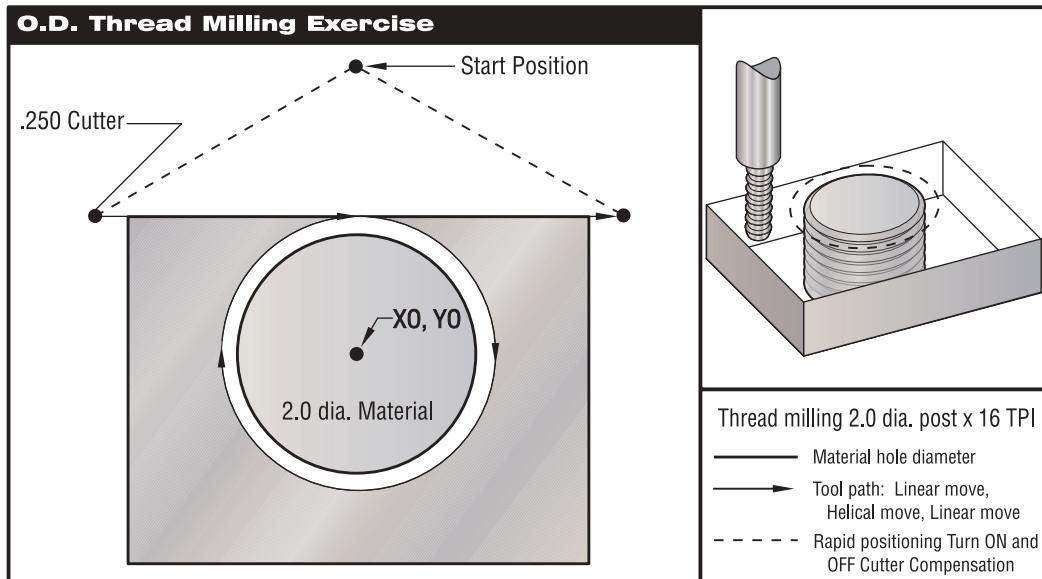
**NOTE:** Maximum cutter compensation adjustability is .175, which is more than enough for this application.

Start with zero in the diameter offset column and enter a negative number to increase the thread diameter.



VS-Series

## O.D. THREAD MILLING -



*O.D. thread milling exercise.*

### PROGRAM EXAMPLE

```
%  
O2400  
(X0,Y0 is at the center of the post)  
(Z0 is at the top of the part)  
(Post height is 1.125 inch)  
G00 G90 G54 X0 Y2.0 S2000 M03  
G43 H01 Z.1 M08  
Z-1.0  
G41 D01 X-1.5 Y1.125  
G01 X0. F15.  
G02 J-1.125 Z-1.0625  
G01 X1.5  
G00 G40 X0 Y2.0  
Z1.0 M09  
G28 G91 Y0 Z0  
M30  
%
```

### DESCRIPTION

*(Thread milling a 2.0 diameter post x 16 TPI)*

*(Turning on cutter compensation.)  
(Linear interpolation onto the post.)  
(360° helical circle; negative Z move.)  
(Linear interpolation off the post.)*

*(Turning off cutter compensation.)*

**NOTE:** A cutter compensation turn on move can consist of any X or Y move from any position just as long as the move is greater than the amount being compensated for. The same rule applies for turning off cutter compensation.

### SINGLE-POINT THREAD MILLING

Using the following data, we will write a program for single-point thread milling procedures:

#### DATA:

- 2.500 Ø hole
- Diameter of cutter (Subtract .750): 1.75
- Radial value (Divide by 2) : .875
- Thread pitch: .0833 (12 TPI)
- Part thickness: 1.00

#### PROGRAM EXAMPLE

%

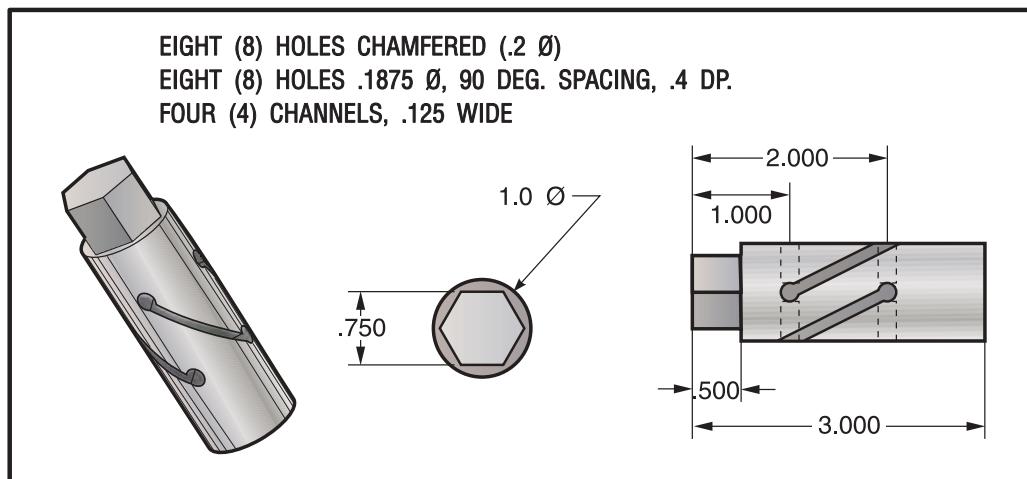
O1000

(X0,Y0 is at the center of the hole)

(Z0 is at the top of the part)

T1 M06	(Tool #1 is a .750 diameter single-point thread tool)
G00 G90 G54 X0 Y0 S2500 M03	
G43 H01 Z.1 M08	
G01 Z-1.083 F35.	
G41 X.275 I1	(Radial value)
G3 X.875 I.3 F15.	
G91 G3 I-.875 Z.0833 L14	(Multiply .0833 pitch x 14 passes = 1.1662 = total in Z-axis)
G90 G3 X.275 I-.300	
G00 G90 Z1.0 M09	
G1 G40 X0 Y0	
G28 G91 Y0 Z0	
M30	
%	

### FOURTH AXIS PROGRAMMING



Fourth axis programming exercise.



VS-Series

### PROGRAM EXAMPLE

%

O1234

(Material is 1.0 Ø round stock x 3.0 L)

(Set material in collet to protrude 2.25.)

(Set fixture parallel with the table T-slots on the right side.)

(X0 is the front of the material.)

(Y0 is the centerline of the spindle and material.)

(Z0 is the top of the part.)

T1 M06

G00 G90 G54 X.250 Y-.500 A0 S4500 M03

G43 H01 Z.1 M08

M98 P1235 L6

G00 G90 Z.1 M09

T2 M06

G00 G90 G54 X1.0 Y0 A0 S5000 M03

G43 H02 Z.1 M08

G82 Z-.1 F10. R.1 P300

X2.0

A90.

X1.0

A180.

X2.0

A270.

X1.0

G00 G80 Z.1 M09

(Cont'd.)

T3 M06

G00 G90 G54 X1.0 Y0 A0 S5000 M03

G43 H03 Z.1 M08

G83 Z-1.125 F12. R.1 Q.25

X2.0

A90.

X1.0

G00 G80 Z.1 M09

T4 M06

G00 G90 G54 X1.0 Y0 A0 S5000 M03

G43 H04 Z.1 M08

M98 P1236

G00 G90 Z.1 M09

G28 G91 Y0 Z0

M30

%

%

O1235

G01 Z-.125 F50.

Y.5 F35.

G00 Z.1

G91 Y-1.0 A60.

G90

M99

%

### DESCRIPTION

(Fourth axis program using a Haas Servo 5C)

(Tool #1 is a .500 end mill to mill hex.)

(Tool #2 is a .375 Ø NC spot drill.)

(Tool #3 is a .1875 Ø stub twist drill.)

(Tool #4 is a .125 Ø end mill.)

(Subprogram to mill hex.)

The following subprogram can be written in absolute or incremental programming. Examine each program and determine which style would be faster and easier to understand and to program in the future.

#### **ABSOLUTE:**

```
%  
O1236 (Subprogram to mill channels.)  
G01 Z-.25 F15.  
X2.0 A90.  
G00 Z.1  
A180.  
G01 Z-.25  
X1.0 A90.  
G00 Z.1  
A180.  
G01 Z-.25  
X2.0 A270.  
G00 Z.1  
A360.  
G01 Z-.25  
X1.0 A270.  
G00 Z.1  
M99  
%
```

#### **INCREMENTAL:**

```
%  
O1236 (Subprogram to mill channels.)  
G91  
G01 Z-.35 F15.  
X1.0 A90.  
G00 Z.35  
A90.  
G01 Z-.35  
X-1.0 A-90.  
G00 Z.35  
A90.  
G01 Z-.35  
X1.0 A90.  
G00 Z.35  
A90.  
G01 Z-.35  
X-1.0 A-90.  
G00 G90 Z.1  
M99  
%
```

## **FORMULAS**

### **TAPPING -**

STANDARD thread formula:

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

METRIC thread formula:

Pitch (P) multiplied by .03937 = \_\_\_\_\_ multiplied by RPM = Feed rate in inches per minute  
**(P x .03937) x RPM = F**

### **SPEED AND FEEDS -**

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

.262 multiplied by the cutter diameter multiplied by the RPM = SFM  
**.262 x Cutter Diameter x RPM = SFM**

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

3.82 multiplied by the recommended SFM divided by the cutter diameter = RPM  
**(3.82 x SFM) / Cutter Diameter = RPM**

I.P.M. (Inch Per Minute):

Feed per tooth multiplied by the number of cutter teeth multiplied by the RPM = Feed rate in inches per minute.

**(Feed/tooth x n) x RPM = IPM or F**

### CUBIC INCH PER MINUTE: MATERIAL REMOVAL RATE

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

**(E Diameter x d) x IPM = CIPM**



### 3.16 AUTOMATIC CHIP AUGER OPERATION

The automatic chip auger assists the user in removal of chips for jobs with heavy material removal. When running, the chip auger will sense auger motor overcurrent and reverse direction momentarily, thus attempting to free up chip jams. This procedure will be repeated until chips are cleared or auger retry limit (Parameter 219) is reached. If the chip auger is running and the door is opened, the chip auger will stop, thus adding a degree of safety to auger operation. If there is no axis motion or keyboard action within the time set in Parameter 255, the auger will automatically shut off.

**NOTE:** It is recommended that the chip auger be used intermittently. Continuous operation will cause the motor to overheat.

**NOTE:** On a machine with a safety circuit, the chip auger will only run with the door closed regardless of the Conveyor Door Override bit.

#### AUGER KEYBOARD COMMANDS

The auger can be started at any time from the keyboard. The auger can be enabled in either direction by pressing the CHIP FWD or CHIP REV and stopped by pressing the CHIP STOP key. The reverse direction of the auger will stop when the CHIP REV key is released. The auger will also stop by pressing the RESET key.

#### AUGER PROGRAM COMMANDS

Use M codes M31 and M33 to control the auger from within a program or in MDI. M31 commands the auger forward and M33 stops the auger. Refer to the "M Codes" section for a more detailed description.

#### AUGER PARAMETERS

The parameters that control the auger are below.

CNVYR RELAY DELAY	Parameter 216	CNVYR IGNORE OC TIM	Parameter 217
CNVYR RETRY REV TIM	Parameter 218	CNVYR RETRY LIMIT	Parameter 219
CNVYR RETRY TIMEOUT	Parameter 220	CNVYR TIMEOUT	Parameter 255

A complete description of auger parameters is given in the "Parameters" section of this manual.

### 3.17 WARMUP COMPENSATION

When the machine is powered on, if setting 109, and at least one of Settings 110, 111 or 112, are set to a nonzero value, the following warning will be displayed:

CAUTION! Warm up Compensation is specified!

Do you wish to activate

Warm up Compensation (Y/N)?

If the operator responds 'Y', the control immediately applies the total compensation (Setting 110, 111 and/or 112), and the compensation begins to decrease as the time elapses. For instance, after 50% of the time in Setting 109 has elapsed, the compensation distance for the X axis, in Setting 110, will be 50%.

As with other settings, the Warm up Compensation settings can be changed at any time. Updating the Warm-up Compensation Time may activate compensation, but changes to the X, Y or Z distance settings will not activate compensation. To “restart” the time period, it is necessary to power the machine off and on, and then answer “yes” to the compensation query at start-up.

### **WARNING!**

**Changing settings 110, 111 or 112 while compensation is in progress  
can cause a sudden movement of up to 0.0044 inch.**

The amount of remaining warmup time is displayed on the bottom right hand corner of the Diagnostics Inputs 2 screen using the standard hh:mm:ss format. The initial amount of warmup time to be used, starting when power is applied, is specified in Setting 109, WARMUP TIME IN MIN.

## **3.18 How the Control Moves the Machine**

Acceleration and deceleration is what the machine does when it is changing speed. Acceleration is what the machine does when the speed is increasing and deceleration is what the machine does when it is slowing down. The machine cannot change speed instantly so that a change of speed occurs over some amount of time and a distance.

Changes of speed affect how the control moves for both rapid motion and feed motion. Rapid motion occurs independently for each axis in motion and uses acceleration set for each axis. Feed motion coordinates one or more axes to accelerate in unison, move in unison, and decelerate in unison. This type of feed motion is called acceleration before interpolation and uses a fixed acceleration rate for all axes.

### **RAPID MOTION**

Rapid motion uses constant acceleration and deceleration, with maximum acceleration and maximum speed set as parameters per axis. End-point arrival in rapid motion occurs with S-curve velocity to prevent shock vibration to the machine. A rapid motion followed by another rapid motion is blended with a rounded corner, controlled by a parameter called “In Position Limit.” It is usually about 0.06 inch. A rapid motion followed by a feed motion or a rapid motion in “Exact Stop” mode will always decelerate to an exact stop before the next motion.

S-curve velocity control refers to the rate of change of acceleration or deceleration. Without S-curve, there may be abrupt changes in deceleration, resulting in machine vibration. With S-curve at the end of a rapid move, there are only gradual changes to deceleration and thus machine vibration is reduced.

### **FEED MOTION**

A feed motion coordinates or “interpolates” the motion of multiple axes. Feed motions always use constant acceleration before interpolation. With the Haas control, up to five axes can be in motion in a feed. These are X, Y, Z, A and B. Maximum feedrate is 500 inches per minute for the linear (XYZ) axes and 300 degrees per minute for the rotary (AB) axes.

The blending of axis motion is controlled by Setting 85 (Max Corner Rounding) and the G187 command. Setting 85 and G187 provide for a continuously adjustable range of corner rounding between an exact stop and inexact stop. The value of the setting is the maximum deviation allowed from the exact programmed path.



A linear move (G01) started at an exact stop and ending at an exact stop will have zero positioning error. That is, it will follow exactly that programmed path. It is only at the start or end at speed that blending or corner rounding can occur. This blending action should not be confused with overshoot. Overshoot is where a control would go past a corner and then reverse back onto the required path. The Haas control does not overshoot under any circumstances.

Circular moves (G02 or G03) are not treated any differently than a linear feed motion. In a circular motion, if a feed starts at an exact stop and ends at an exact stop, there is no positional error introduced no matter what the feedrate. Feed in a circular motion is limited to 300 inches per minute. Corner rounding can still occur when a circular motion is blended with a linear or a circular motion, but is also controlled by Setting 85 and G187.

#### *FEED PATH LOOK-AHEAD*

Block look-ahead is something that is needed when the distance which the control requires to get up to speed is more than half the length of the programmed linear strokes. Without look-ahead, the control will simply overlap the deceleration of one stroke with the acceleration of the next stroke. This would limit the speed of the motion based on the length of those strokes. The Haas control has block look-ahead standard, but without the High-Speed Machining option it is limited. With look-ahead, blending of one stroke to another can occur at full speed, but the angle of blending is small without the High-Speed Machining option.



VS-Series

### 3.19 CUTTER COMPENSATION

Cutter compensation is a method of shifting the tool path so that the actual centerline of the tool is moved to either the left or right of the programmed path. Normally cutter compensation is programmed to shift the tool in order to control feature size. The OFFSET display is used to enter the amount that the tool is to be shifted. The offset can be entered as either a diameter or radius value, depending on setting 40, for both the geometry and wear values. If diameter is specified, the shift amount is half of the value entered. The effective offset values are the sum of the geometry and wear values. Cutter compensation is only available in the X-axis and the Y-axis for 2D machining (G17). For 3D machining, cutter compensation is available in the X-axis, Y-axis and Z-axis (G141).

#### GENERAL DESCRIPTION OF CUTTER COMPENSATION

G41 will select cutter compensation left; that is, the tool is moved to the left of the programmed path to compensate for the amount entered in the offsets page (Refer to setting 40). G42 will select cutter compensation right, which will move the tool to the right of the programmed path. A **Dnnn** must also be programmed with G41 or G42 to select the correct offset number from the radius/diameter offset column. If the offset contains a negative value, cutter compensation will operate as though the opposite G code was specified. For example, a negative value entered for a G41 will behave as if a positive value was entered for G42. Also, if cutter compensation is selected (G41 or G42), you may only use the X-Y plane for circular motions (G17). Cutter Compensation is limited to compensation in only the X-Y plane.

The code G40 will cancel cutter compensation and is the default condition when a machine is powered on. When canceled, the programmed path is the same as the center of the cutter path. You may not end a program (M30, M00, M01, or M02) with cutter compensation active.

The control operates on one motion block at a time. However, it will look ahead to check the next two blocks containing **X** or **Y** motions. Interference checks are performed on these three blocks of information. Setting 58 controls how this part of cutter compensation works. It can be set to Fanuc or Yasnac.

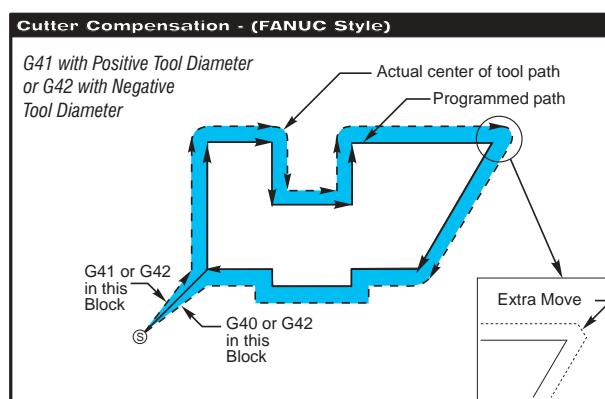
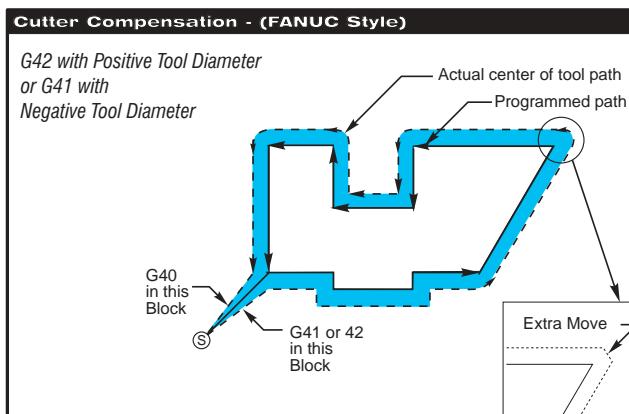
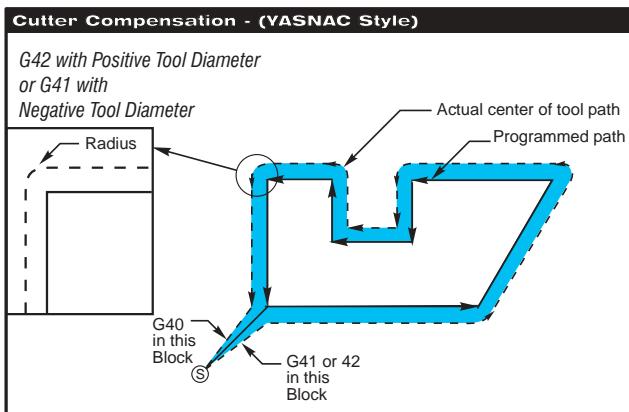
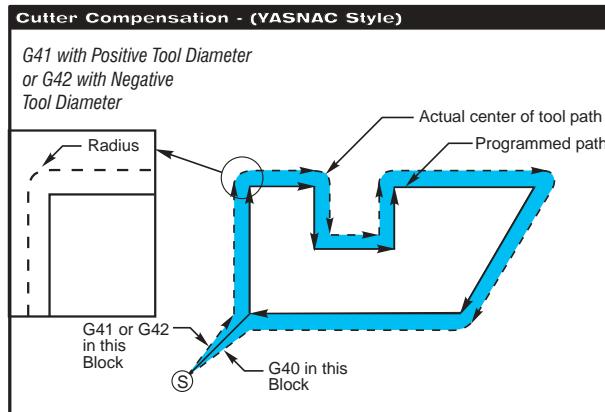
Selecting Yasnac for Setting 58, the control must be able to position the side of the tool along all of the edges of the programmed contour without over cutting the next two motions. A circular motion joins all of the outside angles.

Selecting Fanuc for Setting 58, the control does not require that the tool cutting edge be placed along all edges of the programmed contour, preventing over-cutting. However an alarm will be generated if the cutter's path is programmed such that over-cutting cannot be avoided. Outside angles less than or equal to 270 degrees are joined by a sharp corner and outside angles of more than 270 degrees are joined by an extra linear motion (See the following diagrams).

The following diagrams show how cutter compensation works for the two possible values of Setting 58. Note that a small cut of less than tool radius and at a right angle to the previous motion will only work with the Fanuc setting.



VS-Series



## ENTRY AND EXIT FROM CUTTER COMPENSATION

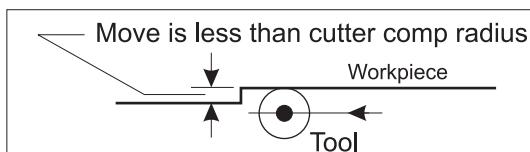
When entering and exiting cutter compensation or when changing from left side to right side compensation, there are special considerations to be aware of. Cutting should not be performed during any of these moves. To activate cutter compensation, a nonzero **D** code must be specified with either G41 or G42 and G40 must be specified in the line that cancels cutter compensation. In the block that turns on cutter compensation, the starting position of the move is the same as the programmed position, but the ending position will be offset, to either the left or right of the programmed path, by the amount entered in the, radius/diameter, offset column. In the block that turns off cutter compensation, the starting point is offset and the ending point is not offset. Similarly, when changing from left to right or right to left side compensation, the starting point of the move needed to change cutter compensation direction will be offset to one side of the programmed path and end at a point that is offset to the opposite side of the programmed path. The result of all this is that the tool moves through a path that may not be the same as the intended path or direction (See drawing A). If cutter compensation is turned on or off in a block without any X-Y move, there is no change made to cutter compensation until the next **X** or **Y** move is encountered. To exit from cutter compensation, you must specify G40.

You should always turn off cutter compensation in a move that clears the tool away from the part being cut. If a program is terminated with cutter compensation still active, an alarm is generated. In addition, you cannot turn cutter compensation on or off during a circular move (G02 or G03); otherwise an alarm will be generated.

An offset selection of D0 will use zero as the offset value and have the same effect as turning off cutter compensation. If a new D value is selected while cutter compensation is already active, the new value will take effect at the end of the proceeding move. You cannot change the D value or change sides during a circular motion block.

When turning on cutter compensation in a move that is followed by a second move at an angle of less than 90 degrees, there are two ways of computing the first motion: cutter compensation type A and type B (Setting 43). Type A is the default in Setting 43 and is what is normally needed; the tool moves directly to the offset start point for the second cut. Type B is used when clearance around a fixture or clamp is needed, or in rare cases when part geometry demands it. The diagrams on the following pages illustrate the differences between type A and type B for both Fanuc and Yasnac settings (Setting 58).

## Improper Cutter Compensation Application



Note that a small cut of less than tool radius and at a right angle to the previous motion will only work with the Fanuc setting. A cutter compensation alarm will be generated if the machine is set to the Yasnac setting.

## FEED ADJUSTMENTS IN CUTTER COMPENSATION

When using cutter compensation in circular moves, there is the possibility of speed adjustments to what has been programmed. If the intended finish cut is on the inside of a circular motion, the tool should be slowed down to ensure that the surface feed does not exceed what was intended by the programmer. There are problems, however, when the speed is slowed by too much. For this reason, Setting 44 is used to limit the amount by which the feed is adjusted in this case. It can be set between 1% and 100%. If set to 100%, there will be no speed changes. If set to 1% the speed can be slowed to 1% of the programmed feed.

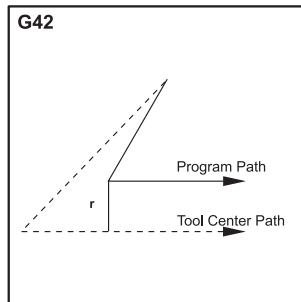
When the cut is on the outside of a circular motion, there is no speed-up adjustment made to the rate feed.



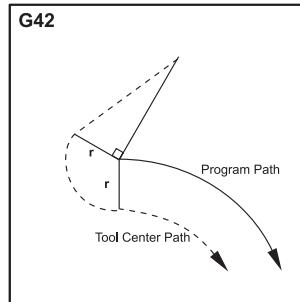
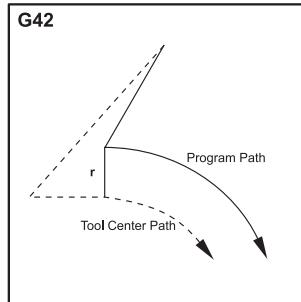
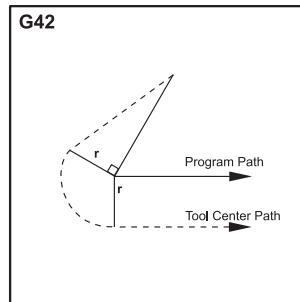
VS-Series

## CUTTER COMPENSATION ENTRY (Yasnac)

TYPE A

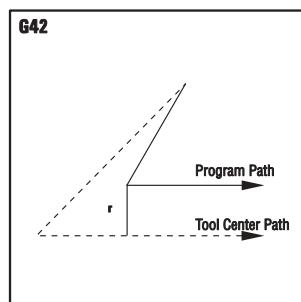


TYPE B

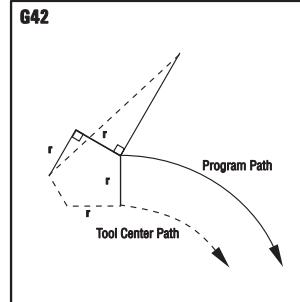
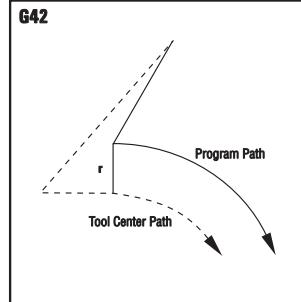
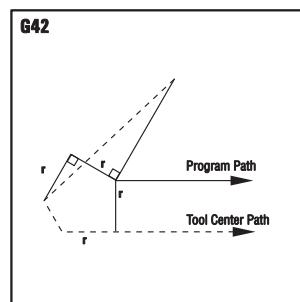


## CUTTER COMPENSATION ENTRY (Fanuc style)

TYPE A



TYPE B



## CIRCULAR INTERPOLATION AND CUTTER COMPENSATION

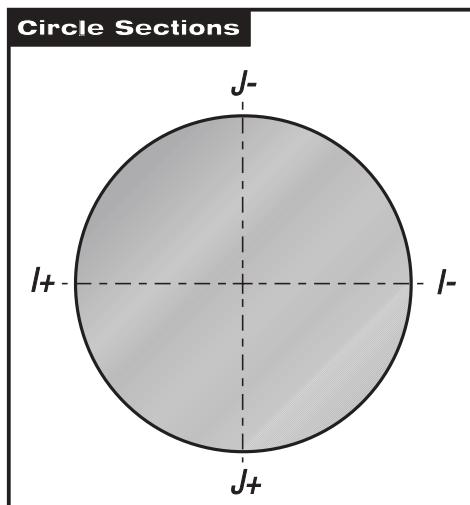
In this section, the usage of G02 (Circular Interpolation Clockwise), G03 (Circular Interpolation Counterclockwise) and Cutter Compensation (G41: Cutter Compensation Left, G42: Cutter Compensation Right) is described.

Using G02 and G03, we can program the machine to cut circular moves and radii. Generally, when programming a profile or a contour, the easiest way to describe a radius between two points is with an **R** and a value. For complete circular moves (360o), an **I** or a **J** with a value must be specified. The circle section illustration will describe the different sections of a circle.

By using cutter compensation in this section, the programmer will be able to shift the cutter by an exact amount and be able to machine a profile or a contour to the exact print dimensions. By using cutter compensation, programming time and the likelihood of a programming calculation error is reduced due to the fact that real dimensions can be programmed, and part size and geometry can be easily controlled.

The following are a few rules about cutter compensation that have to be closely followed in order to perform successful machining operations. Always refer to these rules when programming.

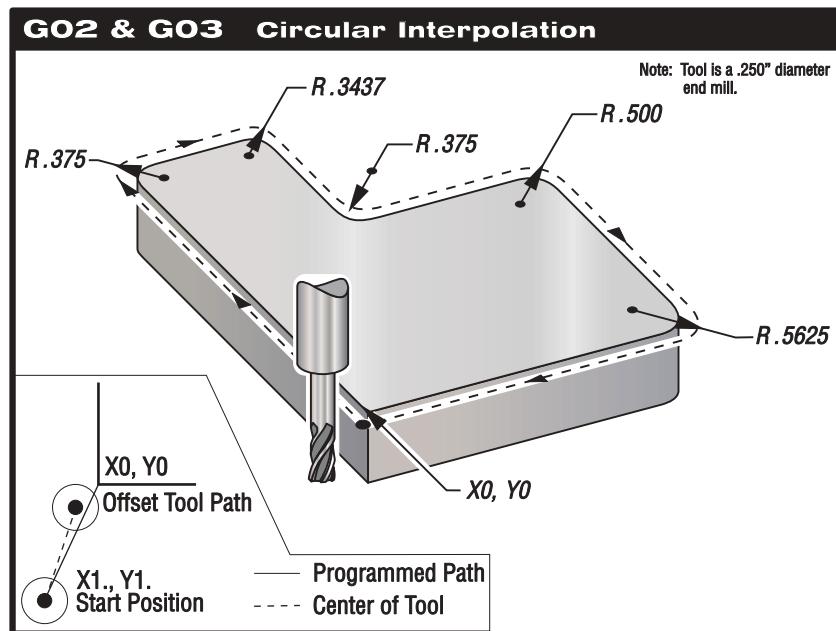
1. Cutter compensation must be turned ON during a G01 X,Y move that is equal to or greater than the cutter radius, or the amount being compensated for.
2. When an operation using cutter compensation is done, the cutter compensation will need to be turned OFF, using the same rules as the turn ON process, i.e., what is put in must be taken out.
3. In most machines, during cutter compensation, a linear X,Y move that is smaller than the cutter radius may not work. (Setting 58 - set to Fanuc - for positive results.)
4. Cutter compensation cannot be turned ON or OFF in a G02 or G03 arc movement.
5. With cutter compensation active, machining an inside arc with a radius less than what is defined by the active D value will cause the machine to alarm.





VS-Series

The following illustration shows how the tool path is calculated for the cutter compensation. The detail section shows the tool in the starting position and then in the offset position as the cutter reaches the workpiece.



*Programming exercise showing tool path.*

The following program uses no cutter compensation. Tool path is programmed to centerline of the cutter. This is also the way the control calculates for cutter compensation.

The following program uses cutter compensation. Tool path is programmed to centerline of the cutter. This is also the way the control calculates for cutter compensation.

```
%  
O6100  
T1 M06  
G00 G90 G54 X-1. Y-1. S5000 M03  
G43 H01 Z.1 M08  
G01 Z-1.0 F50.  
G41 G01 X0 Y0 D1. F50.  
Y4.125  
G02 X.250 Y4.375 R.375  
G01 X1.6562  
G02 X2.0 Y4.0313 R.3437  
G01 Y3.125  
G03 X2.375 Y2.750 R.375  
G01 X3.5  
G02 X4.0 Y2.25 R.5  
G01 Y.4375  
G02 X3.4375 Y-.125 R.5625  
G01 X-.125  
G40 X-1. Y-1.  
G00 Z1.0 M09  
G28 G91 Y0 Z0  
M30  
%
```

### 3.22 Quick Code

#### INTRODUCTION

This programming option can be activated by contacting your local HAAS dealer.

Quick Code is an innovative new way to program CNC machines. It combines the simplicity and flexibility of G code programming with English descriptive sentences to enable even beginning programmers to construct most 2 dimensional parts. Experienced programmers will also love the speed they can now enter programs manually. This is possible because with one menu selection you can replace a large number of individual keystrokes, with just a few. And what if you don't like the way Quick Code is programmed? Simple! You can change it to suit your needs or programming tastes. Make it as complex or as simple as you like.

#### Background

When NC machines were first introduced they had very limited or no memory at all. They were often run from tapes and instructions needed to be as concise as possible. In order to accomplish this a sort of encryptive language evolved which we called G code programming. A command to "TURN OFF COOLANT" which requires 16 letters and spaces is reduced to "M09" which takes only 3 characters. This made tape lengths and memory requirements manageable to say the least. As it evolved, hundreds of instructions and canned cycles were encrypted into G and M code programming. For an experienced programmer, the G codes are actually very easy to use but the learning process requires constant referring back to the manual to figure out which code to use to accomplish the task. And even the most experienced programmers would have to admit that every once in a while they forgot to put the right "I,K,Q or P's" into say, a G83 drilling cycle. Quick Code eliminates this tedium. Simply handle cursor over to the drill cycle you want and press the write button and all the code you need to drill the hole is inserted with default values for all necessary "I,K,Q,P's". And you can edit those values to suit your individual needs.

#### How It Works

Quick Code reverses the G code encryption confusion. On the right side of the screen you have English commands that describe the operation to perform. By selecting the operation and with one button push, the code is inserted in your program on the left side of the screen. A program is constructed by selecting English commands that are then changed over to machine language or G codes. In doing this you will learn quickly the G code format without studying any manual. Another feature is the ability to cursor through a program and Quick Code will tell you what all the G and M codes mean, shown at the bottom of the screen, a great help in learning the code.

#### An Open System

Quick Code is adaptable to the way you program. Everybody programs a little differently and have special preferences, such as, do you put the "T" command on the same line as the tool change command or before? With Quick Code you can edit the program so that any English command you desire can be matched with any G code to be inserted. Because of this open format we are letting you define innovative new ways to program complex parts using Quick Code.



## What it is Not

Quick Code is not a CAD/CAM package for generating complex moves on 3 dimensional parts. With most CAM packages you have to draw a drawing much like you would in AUTO CAD and then indicate the moves around the drawing and finally generate the code through the post processor. Not a simple task. Nor is it a conversational program with icons where you are asked to fill in the blanks. The difference with these packages is that they require training and much like learning a second language you have to have the time and determination to learn them. They have a tremendous amount of power but you don't always need it. Quick Code is a bridge between high end CAD/CAM and slow and cumbersome G code programming. It is our expectation that it can be used by anyone with very minimal training. For most simple parts we believe that Quick Code is an ideal choice.

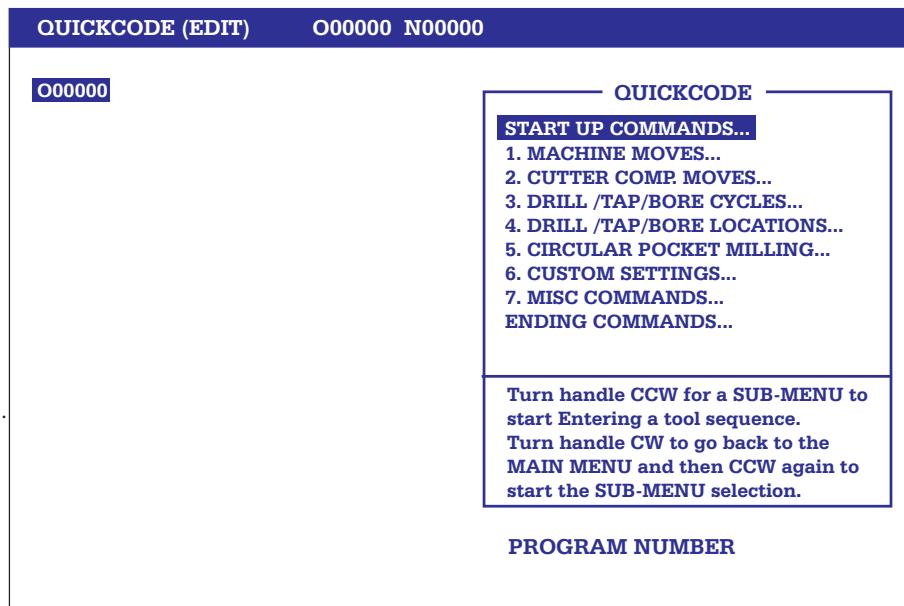
## Conversational Quick code

Conversational Quick Code makes programming with Quick Code even easier. This feature can be used to "prompt" the operator for the information necessary to create a program. Refer to the " Conversational Quick Code" section for a description of how to use this feature.

### QUICK CODE TERMINOLOGY

Before describing the Quick Code environment you need to know the terms listed below. Following this brief list is an illustration of the Quick Code display and how the terms are related to the display.

<b>EDIT WINDOW</b>	Portion of the display that shows the currently edited program.
<b>GROUP WINDOW</b>	Portion of the display which presents a list of groups and items.
<b>GROUP</b>	A list of items that usually have something in common so that they can be grouped together.
<b>ITEM</b>	A line of text representing code that can be added to the edit window when it is selected.
<b>HELP WINDOW</b>	Portion of the display which presents user created help, address code help, and warning messages.



## ACCESSING QUICK CODE

Before Quick Code can be used, the bit labeled QUIK CODE in parameter 57 bit 27 must be set to 1. When this bit is set to 0, you will not be able to access the Quick Code screen. Enter Quick Code by selecting edit mode and then pressing the PRGRM/CONVRS key twice. The first press of the PRGRM/CONVRS key enters the standard editor, whereas the second press of this key will enter the 80 column format of the Quick Code screen. Each additional press of this key will switch between Visual Quick code, Advanced editor, the standard editor, and Quick Code modes.

## THE EDIT WINDOW

The Quick Code edit window is exactly the same as the standard editor on the HAAS control. Each time that you select a group item, as described below, the edit window will be updated to show you what code has been added to the currently edited program. You have access to all of the edit functions with the exception of the jog handle and the block copy function keys. In the standard editor, you can use the jog handle to traverse program text quickly. While in Quick Code, the jog handle is reserved to maneuver through the group list. You can still cursor through the program text by using the cursor keys provided on the center of the keypad. You are also restricted from using the block copy keys while in Quick Code. For this, you can always switch back to standard edit mode by pressing the PROGRM/CONVRS key. At this point you have access to the jog handle, for long comments, and the block copy functions. Quick Code is not available while in BACKGROUND EDIT MODE.

## THE GROUP WINDOW

The group window displays a list of groups that are defined in the Quick Code source file. The groups can be moved through for selection by turning the jog handle in the plus (clockwise) direction. For each jog handle click in the plus direction, the group window cursor will advance to the next group. In this manner you can move through every group in the list. When the last group is highlighted, the next plus click will move the cursor to the first group in the list. To view and cursor through items within a group, turn the jog handle in the minus, counter clockwise, direction. As long as you turn the jog handle in the minus direction the cursor will advance through, display and highlight items in the current group. By turning the jog handle one click clockwise, the group item list will be closed and additional plus clicks will continue to traverse the group list.

## THE HELP WINDOW

The help window is just below the group window. It is used to display Quick Code source file help, address code help, and warning messages to the user.

The Quick Code source file can contain comments that will not be placed into the edit window. These comments will be displayed on the first five lines of the help window. These comments are typically used for explaining item code and usage.

As the user cursors through a program, each address code that is highlighted will be interpreted and a short description of its usage is displayed below the edit window. This address code help is as accurate as possible. Since the program is not being interpreted sequentially as it is when a program is run, full interpretation cannot take place. When the context of an address code cannot be fully determined, the most likely usage is displayed.

Sometimes during editing we can determine if a run time error will occur without actually running the program. For instance we can tell if multiple codes from one G code group are on a line. In this case Quick Code will display a highlighted warning message to the user indicating that there is a problem. This is found below the help window.



## SPECIAL KEYS

Quick Code makes use of the jog handle to select from the group list and group items. This is described in the group window section above. Quick Code action takes place when the WRITE key is pressed. If there is text on the input line, normal text insertion takes place when the WRITE key is pressed. When the input line is blank, pressing the WRITE key will cause Quick Code to take the following action:

-If the currently highlighted Quick Code item is designed as a text help item only, the edit window is not modified.

-If numeric program code is found associated with the highlighted Quick Code item, the edit window cursor is moved to the end of the current edit block and the associated code is inserted after that block. The edit cursor is left at the end of the last Quick Code block that was inserted.

### CONVERSATIONAL QUICK CODE

Quick Code is used to "prompt" the operator for the information necessary to create a program. The "prompting comments" are created by placing a '?' as the first character of a (?comment) in the Quick Code source file (O9999). A comment is any text, up to 34 characters, that is contained in parentheses. When a program is being written using Quick Code, the prompting comments will appear on the screen, requiring a response from the operator. The numeric value entered by the operator will be assigned to the G-code item that immediately precedes the prompting comment in the source file. The Quick Code source file program is O9999.

For example, defining an X axis feed move, the following line of code would be in the source file:

G01 X2. (?WHAT IS THE X LOCATION) F15. (?WHAT IS THE FEEDRATE)

Will produce the following prompt when creating a new program, under another program number (O1234) using Quick Code. And the default X location value displayed below with the prompt, as shown:

WHAT IS THE X LOCATION

X2.  and you ENTER a new X location value of 2.25

X2.25  and then the next command prompt comes up.

With the default feed rate value displayed below the prompt, as shown:

WHAT IS THE FEED RATE

F15.  and you decide to keep this default feedrate value.

The operator must then enter a numeric value and press the WRITE key to change the default feed rate, or simply press the WRITE key to accept the default feed rate. The control will wait for an operator response before entering the block to the edit window. Unacceptable responses, such as those containing too many digits or an unnecessary decimal, will cause the control to flash an error message and wait for another response.

Once the operator has input a value for all of the 'variable' G-code items in a block of code, the entire (revised) block is displayed on the input line, as shown:

CORRECT (Y/N) ?

G01 X2.5 F15. ;

If the block of code is too long to fit on the screen, the operator can scroll to view the entire line using the right or left arrow keys, the HOME key, or the END key. The operator then must enter 'Y' to accept the block, or 'N' to cancel it. If it's accepted, the block is written to the edit file, and the Quick Code processing resumes with the next block (if there is one). If it is not accepted, the prompting process is repeated for the same block.

Pressing the UNDO key while in Quick Code, will exit the current block at any time and remain where you are in the program.

Pressing RESET will exit Quick Code and send the cursor back to the beginning.

### A SAMPLE QUICK CODE SESSION

The following illustrates how Quick Code can be used to build a program. A program will be built to spot, drill, and tap 5 holes on a circular bolt hole pattern. We will assume that tool 1 is a spot drill, tool 2 is a drill for a 10-32 tap, and tool 3 is the tap. Before you proceed, make sure that Quick Code is enabled in parameter 57 bit 27. QUIK CODE should be set to 1. You will also need the Quick Code source program O9999 in the control.

The jog handle is an integral part of using Quick Code and is used quite often. For brevity we use JHCW to mean jog handle clockwise and JHCCW to mean jog handle counter clockwise. For instance, seeing JHCW means that you should turn the jog handle in a clockwise direction.

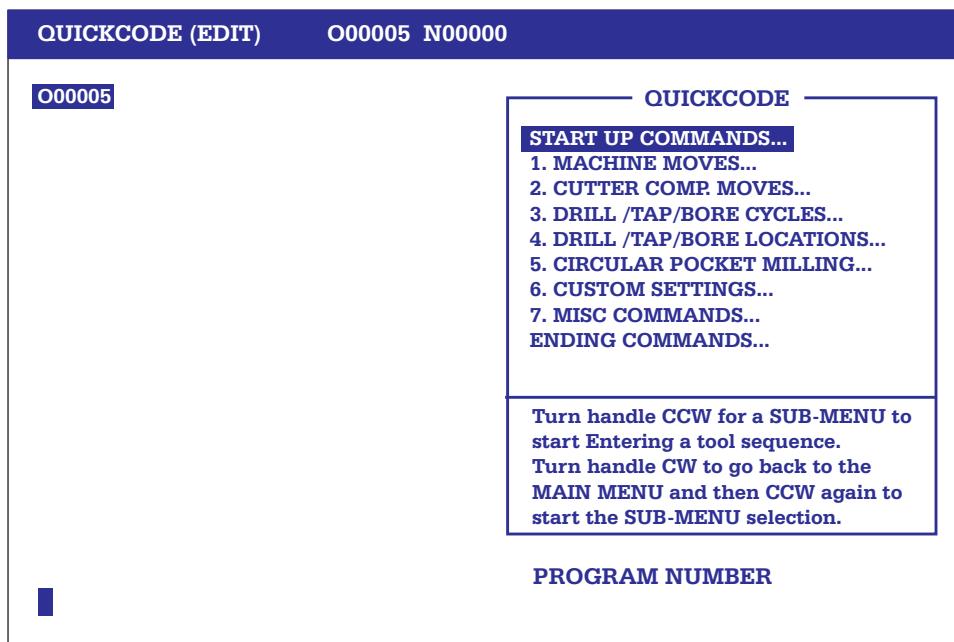
### CREATE A PROGRAM

Quick Code will not generate the new program number for you. So the first thing you must do is to create or select a program. To create a program do the following.

1. Press LIST PROG.
2. Type O0005 (or any other convenient program number)
3. Press WRITE.

This creates a program in the usual manner. Proceed to edit the program by pressing EDIT. The control will switch to the PROGRAM display and you will see the program number and semicolon in the top left of the screen. Now press the PRGRM/CONVRS key twice to enter Quick Code **or in the advanced edit menus under the HELP menu is a sub menu selection for Quick Code.**

The following screen is presented.

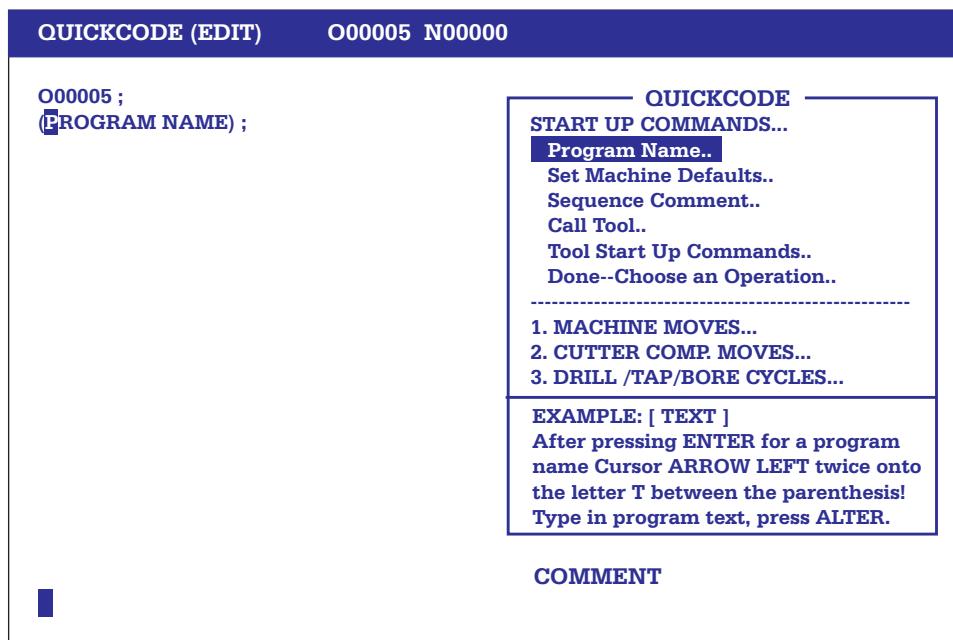




## SELECT THE START UP COMMANDS

1. JHCW until the group titled **START UP COMMANDS** in the group window is highlighted.
2. JHCCW one click. The items belonging to **START UP COMMANDS** will appear and the item, **Program Name** is the one highlighted.
3. Press the WRITE key. This will enter in a (T), for you to cursor arrow left twice onto the T in-between the parenthesis, then type in a program name and press ALTER.

The following figure shows what the screen should look like. Note that for the code that was added, you have text information displayed in the help window just below the group window. When the Quick Code source file is constructed properly, you will see examples of the program code, or instructions, for that command in this HELP window. This can be helpful in determining which item in a group you want.



*Start a program using Quick Code.*

### - CALL TOOL 1

1. While on the START UP COMMANDS menu JHCCW and highlight the group item titled **CALL TOOL**.
2. Press the WRITE key to have the control query you for a tool number in your program, and the control will be flashing with a 1 in the lower left corner as the default value. Press WRITE to accept the number 1.

Press WRITE or Y to except the block listed in the lower left corner. Or N for no, to start the questioning again.

3. JHCCW and highlight the group item titled **TOOL START UP COMMANDS**.
4. Press the WRITE key to have the control query you for the commands to define the start up of tool 1, and enter into your program.

Note that the cursor will move to the end of the block, when you start selecting Quick Code menu items for your program. This is where the next block of code will be entered in after.

**QUICKCODE (EDIT)**      **O00005 N00000**

```
O00005 ;
(PROGRAM NAME) ;
T1 M06 (T) ;
G90 G54 G00 X0 Y0 ;
S750 M03 ;
G43 H01 Z1. M08 ;
```

**QUICKCODE**

---

**START UP COMMANDS...**

- Program Name..
- Set Machine Defaults..
- Sequence Comment..
- Call Tool..

**Tool Start Up Commands..**

Done--Choose an Operation..

---

**1. MACHINE MOVES...**

**2. CUTTER COMP. MOVES...**

**3. DRILL /TAP/BORE CYCLES...**

---

**EXAMPLE:** G90 G54 G00 X0. Y0.  
 . . . S750 M03  
 . . . G43 H01 Z1. M08

ENTER fixture offset, x y location,  
 Spindle Speed and Z start position.

**END OF BLOCK**

*Programmed with the Start Up Command selections entered in with Quick Code for tool 1.*

Remember that after pressing WRITE you can always edit the program to make minor adjustments to code that Quick Code inserts into your program. You do not have to leave the Quick Code display to do this. But you must remember to reposition the cursor back to the block where you want to add the next item. Quick Code will automatically seek the end of the current block that the cursor is on, so there is no need to cursor to the end of the block.

We will assume that the material is aluminum and that the work coordinate zero for G54 is at the center of the bolt hole pattern. The Quick Code source file (O9999) was created with a common program format and menu selections. You could have a different Quick Code source file (O9998 or O9997) for different menu selections and formats. By changing parameter 228 (to 9998 or 9997) you can quickly change the file that Quick Code works with if you had created one.

#### - INVOKE THE SPOT DRILLING CANNED CYCLE G82

1. JHCW and highlight the group titled **4. DRILL/TAP/BORE CYCLES**.
2. JHCCW two clicks. **DRILL with Dwell G82** will be highlighted.
3. Press the WRITE key to have the control query you, and answer the questions, to set up drilling with G82.

Note that Quick Code defined a block of code to execute a spot drill cycle at that present location. You can add more X and Y drill cycle locations if needed by selecting **6. DRILL/TAP/BORE LOCATIONS**.

Note: We do not want to drill a hole at X0 Y0, which is the center of the bolt hole circle, so manually edit in an L0 on the end of the G82 command line. This will ignore the G82 canned cycle until the next location.



VS-Series

Your program will look like this.

QUICKCODE (EDIT)	O00005 N00000
<pre>O00005 ; (PROGRAM NAME) ; T1 M06 (T) ; G90 G54 G00 X0 Y0 ; S750 M03 ; G43 H01 Z1. M08 ; G82 G99 Z-0.109 P0.2 R0.1 F5.;</pre>	
<p>— QUICKCODE —</p> <p>3. DRILL/TAP/BORE CYCLES...</p> <p>Drill G81.. Drill with Dwell G82.. Deep Hole Peck Drill G83.. High Speed Peck Drill G73.. H.S.P.D. W/Return R plane G73.. Bore IN Bore OUT G85.. Bore IN Rapid OUT G86.. Bore IN Shift Rapid OUT G76.. Right Hand Tapping G84.. G80 CANCEL Canned Cycle..</p> <p>EXAMPLE: G82 G99 Z-.15 P2 R.1 F5. G98 Initial point return G99 Rapid plane return P5 = 1/2 Second dwell at Z depth. ENTER drill locations with menu #6.</p>	
<p>END OF BLOCK</p>	

*Program with spot drilling invoked.*

## - EXECUTE A CIRCULAR BOLT HOLE PATTERN

- 1.) JHCW and highlight the group titled **6. DRILL/TAP/BORE LOCATIONS**.
- 2.) JHCCW and highlight the group titled **BOLT HOLE CIRCLE LOCATIONS**.
- 3.) Press the WRITE key to have the control query you for the code on positioning around a bolt hole circle.
- 4.) Enter in the numbers to answer all the questions in the lower left corner of the control screen, to define all the commands necessary to position around a G70 Bolt Hole Circle to drill with a G82 canned cycle.

Here Quick Code will question you for, a G70 command to execute a bolt hole circle pattern. The control will first query you for an X Y center location of the bolt hole pattern.

The next question will be for the radius of the bolt hole circle. The value 1.5 will be flashing in the lower left corner of the screen as the default value for the bolt hole circle radius. If the radius of the bolt circle is different, then enter in the new radius value.

The next questions will be for the number of holes. The value 6 will be flashing in the lower left corner of the screen as the default meaning that 6 holes on a circle will be drilled. We want 5 holes to be drilled. So here you will enter the number 5 to change the pattern so that L5 will be on the G70 line.

Your program will look like this.

QUICKCODE (EDIT)	O00005 N00000
<pre>O00005 ; (PROGRAM NAME) ; T1 M06 (T) ; G90 G54 G00 X0 Y0 ; S750 M03 ; G43 H01 Z1. M08 ; G82 G99 Z-0.109 P0.2 R0.1 F5. ; G70 X0 Y0 I2.5 J30. L5 ;</pre>	<b>QUICKCODE</b> <b>4. DRILL/TAP/BORE LOCATIONS...</b> <ul style="list-style-type: none"> <li>X Location..</li> <li>Y Location..</li> <li>X &amp; Y Location..</li> <li>A Location..</li> <li>Initial Point or R Plane Return..</li> <li><b>Bolt Hole Circle Locations..</b></li> <li>Bolt Hole Arc Locations..</li> <li>Bolt Holes At Angle Locations..</li> <li>More Bolt Hole Pattern Help..</li> <li>G80 CANCEL Canned Cycle..</li> </ul> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> First define drill cycle and center Location of B.H.C.  If you don't want hole in center of B.H.C. Than put an L0 on the line before B.H.C. To not drill hole. </div> <p style="text-align: center;"><b>END OF BLOCK</b></p>

*Program with a bolt hole circle added.*

By now you should have a good idea of how your program changes, after selecting a group item with the handwheel and pressing WRITE. To save space we will not show you each display as a selection is made. Instead we will list the remaining actions needed to finish drilling and tapping the 5 holes. The remaining selections are very similar to what we have already done.

#### - CALL TOOL 2

1. JHCW to the group titled **START UP COMMANDS** then JHCCW and highlight the group item titled **CALL TOOL**.
2. Press the WRITE key to have the control query you for a tool and enter in the number 2.
3. JHCCW and highlight the group item titled **TOOL START UP Commands**.
4. Press the WRITE key to have the control query you for defining all the commands to start up tool 2 in your program.

#### - INVOKE THE SPOT DRILLING CANNED CYCLE G83

1. JHCW and highlight the group titled **4. DRILL/TAP CYCLES**.
2. JHCCW until **Deep Hole Peck DRILL G83** is highlighted.
3. Press the WRITE key to have the control query you for the code on setting up drilling with G83.



Note that Quick Code defined a block to execute a spot drill cycle at that present location. You can add more X and Y drill cycle locations if needed by selecting **6. DRILL/TAP/BORE LOCATIONS**.

Note: We do not want to drill a hole at X0 Y0, which is the center of the bolt hole circle, so manually edit in an L0 on the end of the G82 command line. This will ignore the G82 canned cycle until the next location.

#### - EXECUTE A CIRCULAR BOLT HOLE PATTERN

1. JHCW and highlight the group titled **6. DRILL/TAP/BORE LOCATIONS**.
2. JHCCW and highlight the group titled **BOLT HOLE CIRCLE LOCATIONS**.
3. Press the WRITE key to have the control query you for the code on positioning around a bolt hole circle.
4. Enter in the numbers to answer all the questions in the lower left corner of the control screen, to define all the commands necessary to position around to drill a Bolt Hole Circle with a G83 canned cycle.

And now for tool 3, the 10-32 tap.

#### - CALL TOOL 3

1. JHCW to the group titled **START UP COMMANDS** then JHCCW and highlight the group item titled **CALL TOOL**.
2. Press the WRITE key to have the control query you for a tool and enter in the number 3.
3. JHCCW and highlight the group item titled **TOOL START UP Commands**.
4. Press the WRITE key to have the control query you for defining all the commands to start up tool 3 in your program.

#### - INVOKE THE TAPPING CANNED CYCLE G84

1. JHCW and highlight the group titled **4. DRILL/TAP CYCLES**.
2. JHCCW until **Right Hand Tapping G84** is highlighted.
3. Press the WRITE key to have the control query you for the code on setting up tapping with G84.

Note that Quick Code defined a block to execute a tapping cycle at that present location. You will need to enter in the appropriate speed and feed for this tapping cycle. You can add more X and Y tap cycle locations if needed by selecting **6. DRILL/TAP/BORE LOCATIONS**.

Note: We do not want to tap a hole at X0 Y0, which is the center of the bolt hole circle, so manually edit in an L0 on the end of the G84 command line. This will ignore the G84 canned cycle until the next location.

#### - EXECUTE A CIRCULAR BOLT HOLE PATTERN

1. JHCW and highlight the group titled **6. DRILL/TAP/BORE LOCATIONS**.
2. JHCCW and highlight the group titled **BOLT HOLE CIRCLE LOCATIONS**.
3. Press the WRITE key to have the control query you for the code on positioning around a bolt hole circle.
4. Enter in the numbers to answer all the questions in the lower left corner of the control screen, to define all the commands necessary to position around to tap a Bolt Hole Circle with a G84 canned cycle.

At this point you may decide to position the table forward to remove the part. You can do this in your program with the following.

#### - Rapid Z axis coolant OFF

1. JHCW and highlight the group titled **ENDING COMMANDS**.
2. JHCCW until the **RAPID Z Axis COOLANT OFF** is highlighted.
3. Press WRITE.

This will query you for the location to Rapid the Z axis and turn OFF the coolant with an M09.

And finally, you can properly end your program with:

#### - Sending the machine home and ending the program

1. JHCW and highlight the group titled **ENDING COMMANDS**.
2. JHCCW until **HOME Y and Z Axes** is highlighted.
3. Press WRITE.
4. JHCCW and highlight the group titled **END PROGRAM and Reset M30**.
5. Press WRITE.

You now have a program ready to run. You should always verify everything in graphics to make certain that you have not forgotten any steps. Although this looks like a lot of steps, it is actually very easy once you become familiar with the Quick Code environment. The above program can be generated in less than a minute.

#### *THE QUICK CODE SOURCE FILE*

All of the text seen in the group window, all of the code associated with items of groups, and much of the help text observed in the help window is contained in a G code program. This program is called the Quick Code source file. With this design, the user can modify Quick Code and tailor it to his specific needs. You can add or change groups and items. The user can develop his own Quick Code file, or program, by editing this file. Dealers can develop new applications and distribute them to their customers. The ability to edit the source file makes Quick Code an extremely flexible tool.

#### SOURCE FILE PROGRAM DESIGNATION

Program number 9999 is the default Quick Code source file. Every HAAS control equipped with Quick Code comes with a sample O9999 program installed. The default program number can be changed by changing parameter 228. If the file number in parameter 228 is not found in the control, the message FILE NOT FOUND is displayed and you will not be able to enter the Quick Code screen. The source file must be formatted as defined below. If program O9999 is not formatted in the appropriate manner then you may not see all, or any, of the defined Quick Code groups. You can use the following skeleton as a start for defining a Quick Code source file.

```
%  
O9999 (QUICK CODE – HAAS AUTOMATION INC.)  
(  
(ADD ANY COMMENTS HERE THAT PERTAIN TO)  
(THE ENTIRE SOURCE FILE. FOR INSTANCE)
```



VS-Series

```
(YOU CAN RECORD WHO MADE THE FILE, THE)
(DATE AND TIME OF THE LAST CHANGE, A)
(VERSION NUMBER, OR ANYTHING ELSE YOU )
(WANT. ALL COMMENTS PRIOR TO THE FIRST)
(GROUP ARE NOT SEEN BY THE USER.)
()
(QUICK CODE GROUP DEFINITIONS FOLLOW)

.
.

()
(END OF QUICK CODE)
%
```

### DEFINING A GROUP IN THE GROUP LIST (\*)

To define a group that will show up in the group window, simply enter a comment where the first character is an asterisk. For instance if you want five groups to show up in the group window, then you would include the following five lines in the Quick Code source file.

```
(*GROUP1)
(*GROUP2)
(*GROUP3)
(*GROUP4)
(*GROUP5)
```

Of course, you can use any descriptive title for the group that is appropriate to what the group will contain. Group titles can be up to 35 characters long. Any additional characters beyond 35 will not be displayed.

### GROUP HELP

The first six comments after the group definition will be displayed in the help window. These comments can be used to explain what is contained in the group. For example:

```
(*GROUP)
(THIS GROUP CONTAINS HELP ON HOW TO)
(USE QUICK CODE. WHEN THIS GROUP IS)
(HIGHLIGHTED, TURN THE JOG HANDLE IN)
(THE MINUS DIRECTION FOR MORE HELP.)
```

This is a method of documenting the source file for the developer of the Quick Code file. Documenting comments can also be hidden in the source file by placing an empty comment after group help comments. In the following example only the first two comments are displayed in the help window.

```
(*HELP)
(ONLY THE FIRST TWO COMMENTS ARE)
(DISPLAYED IN THE HELP WINDOW.)
()
(THIS COMMENT IS NOT DISPLAYED)
```

If more than five lines are required to comment on a group, then you can use several groups to display 5, 10 or 15 lines of help. With this method you can add any amount of information you want about that group.

## GROUP CODE

What happens when a group definition is highlighted and the user presses the WRITE key? If there is a G code after the group definition and before any other group or item definitions, then that G code will be inserted into the program that is being developed. Groups do not have to contain items for generating G code. A group title can stand alone as a code generating entity. The following group definition would add a G28 M30 to the program being developed when WRITE is pressed.

```
(*END OF PROGRAM)
(THIS RETURNS ALL AXES TO MACHINE)
(ZERO AND ENDS PROGRAM EXECUTION)
(G28 M30)
()
G28 M30
()
```

Note that the user will not see what G code is generated until the WRITE key is pressed and the code is inserted into the program. For this reason you may want to place the code that is to be generated in a help comment as is done above.

Quick Code can also generate comments in the program being generated. Any comments following an empty comment will be added to the currently edited program. In fact all code following an empty comment is inserted into the program until another empty comment is encountered or until a group or item definition is encountered. The empty comment must be the first code in the block. Any code in the same block as the empty comment is not entered into the program. In the following example, only the code in blocks between the empty comment blocks are added to the program being generated.

```
(*GENERATES COMMENTS AND CODE)
(THIS IS NOT ADDED TO PROGRAM)
()(THIS IS NOT ADDED TO PROGRAM)
(THESE COMMENTS WILL BE ADDED TO THE)
(PROGRAM WHEN THIS GROUP IS)
(HIGHLIGHTED AND WRITE IS PRESSED)
G0 G90 G54 (THIS CODE IS ADDED)
()
(THESE COMMENTS ARE NOT ADDED TO THE)
(PROGRAM BEING GENERATED)
```

## DEFINING AN ITEM BELONGING TO A GROUP (\*\*)

To define an item belonging to a group simply enter a comment after a group definition where the first two characters of the comment are asterisks. For instance, the following code generates a group with four subordinate items.

```
(*GROUP)
(**ITEM1)
(**ITEM2)
(**ITEM3)
(**ITEM4)
```

With the above Quick Code source file, only one group is displayed in the group window when the jog handle is turned clockwise. When the jog handle is turned counter clockwise, the five items are displayed and traversed. The item titles are indented one space so that you can differentiate items from groups. Only 34 characters of the item definition comment are displayed in the group window. Additional characters are ignored. The group that items belong to will always be displayed on the screen. The only limit to the number of items in a group is the amount of control memory available.



## ITEM HELP

Item help works the same way as group help. The first six comments after the item definition are displayed in the help window. If more than six lines are required, it is recommended that the desired comments are continued on the next item. In this case instructions would have to be added to indicate which item generates the G code.

For example:

```
(*GROUP)
(**HELP FOR THE FOLLOWING ITEM)
(THESE LINES OF CODE ARE HELP)
(COMMENTS THAT REQUIRE MORE THAN)
(FIVE LINES OF COMMENTARY)
(THIS IS THE LAST LINE OF THIS ITEM)
(**ITEM THAT GENERATES CODE)
(AND HERE WE FINISH THE COMMENTARY)
(FOR CODE GENERATED BY THIS ITEM)
G0 G90 G01 F30
()(******)
(*NEXT GROUP)
```

Although the above example is somewhat awkward, it does provide a method that will satisfy unusual cases. The line with all of the asterisks is legal. It is not inserted into the current program when the WRITE key is pressed. It is used to visually separate groups.

## ITEM CODE

Code generated by group items is formatted in the same manner as group code is formatted. Refer to the section on group code for an explanation of how code is generated.

### A SAMPLE Quick Code SOURCE FILE

After developing or modifying a Quick Code file, it is recommended that you save an off-line copy in a computer. You can keep comments in the Quick Code source file prior to the first group indicating what version the file is and how it differs from other versions. Maintain this program as you would any other G code program in your control with a proper backup scheme. **Remember! This source program file operates the Quick Code feature in your HAAS machine. And you can have more than one Quick Code file, but the one the control is using is the program number listed under parameter 228 which should be a 9000 number and the one Haas uses is program number O9999.**

A sample Quick Code source file can be found on the floppy that comes with the control. It contains many examples of how Quick Code can be used.

## VISUAL QUICK CODE

Visual Quick Code (VQC) is a graphical editor made to help simplify programming for commonly made, simple parts. Given a standard part template and a set of dimensions, a program is created.

### Quick Start Guide

1. Before starting VQC, if you wish to insert G-code to an existing program, open your program in the Advanced Editor and place the cursor at the ";" (End of Block) where the new program will be added.
2. In Edit mode, press the PRGRM/CONVR key three times to enter VQC. You can also enter VQC by using the pull-down menus in the Advanced Editor under HELP. After entering you will see a mostly empty screen with a list of words or short phrases on the right. These are the part categories.
3. Using the up and down arrow keys, select the part category you want, then press WRITE. Part templates will be seen in the large square area.
4. Using the up, down, left and right arrow keys, select a part template and press WRITE or press CANCEL to return to the category selection screen (step 3). Pressing WRITE (on the part template) will display an enlarged image of the selected part in the large square area including variables identifying the part dimensions.
5. Enter the data for the part. NOTE: Z0 will typically be 0, and the other Z values will typically be negative. R and C values are used to specify the radius or chamfer of a corner.
6. When the last value is entered, the control will open a window displaying four options.

#### A) SELECT/CREATE A PROGRAM

If selected, another window will open prompting the user to select a program name. Simply highlight the desired name and press WRITE. This will cause VQC to output to the selected program. If the program already contains G-code, VQC will output to the beginning before the existing G-code and will not pass any M30s. The user also has the option to CREATE a new program by simply entering a program name using 'O'. Then by simply pressing WRITE again will cause VQC to output to the newly created program.

#### B) ADD TO CURRENT PROGRAM

If selected, the G-code generated by VQC will be inserted at the last position of the cursor within a program before entering VQC. If the program was empty, then any M30s. in the G-CODE section of the VQC template will be passed into the program. If the program contained anything, then NO M30s. will be passed.

#### C) MDI

If selected, the newly created VQC G-code will output to MDI. Note: Anything that may be in MDI will be overwritten.



#### D) CANCEL

If selected, the window will close and the focus returns to the parameter list. Note: the UNDO and CANCEL keys also perform this operation.

Check the program that was created, for example, run the program in graphics mode and check the tool paths. Verify the tool offsets, and run the preliminary part using reduced feeds.

---

**Note:** Cutter Compensation may be applied, if so all Cutter Compensation rules will be used, refer to the Cutter Compensation section for complete details.

### *VISUAL QUICK CODE INTRODUCTION*

#### Starting

You have the choice of either starting from scratch by creating a new empty program; or use VQC to insert code into an existing program. To insert into an existing program, select the program, enter Advanced Editor and position the cursor at the ";" (end of block) where you want the new code to be inserted **after**.

If you choose to start with a new program, VQC will end the program with an M30 (program end and rewind), if it exists in the template.

VQC will not end the code with an M30 if it is inserted into an existing program. Regardless if there is an M30 in the template (this is to prevent unwanted or multiple M30s).

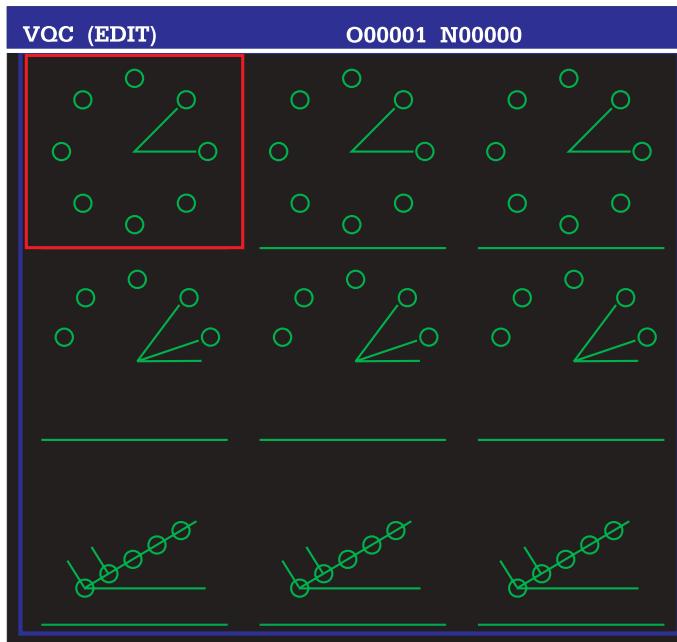
To start Visual Quick Code (VQC) enter Edit mode then press the PRGRM/CONVRS key three times. Another way is to use the pull down menus in the Advanced Editor under HELP.

The ability to load different templates is useful for users wanting to try templates without the need of making the template name 9997 or deleting an older template. By pressing the F2 key, a window will open prompting the user to SELECT a template. The default template name which VQC will attempt to load after power up is: O9997. If O9997 is not found, a window will open prompting the user to load or select a different template.

After a different template has been selected, VQC will remember the selected template while the machine is powered up. After the power has been cycled, VQC will switch back to the default template file name: O9997. If the selected program is not a valid VQC template, the message "No Category" will be shown on the lower left corner of the screen and the Category List will be empty.

#### Selecting a Category

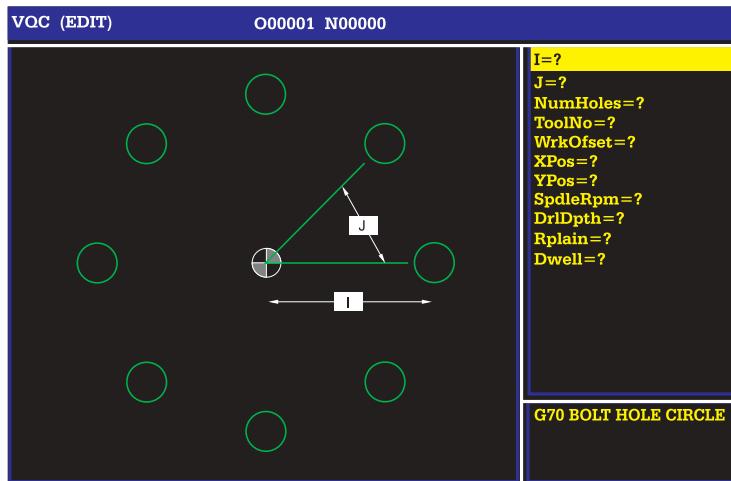
Use the arrow keys to select the parts category that most closely matches the desired part and press WRITE. A set of thumbnail illustrations of the parts in that category will appear. These are the part templates for that category.



*G70 Bolt Hole Circle Selected*

### Selecting a Part Template

Use the arrow keys to select a template on the page. Pressing WRITE will display an outline of the part and allow the programmer to enter dimensions and other information to make the selected part. Press CANCEL to return to the list of categories.



### Entering the Data

The control will prompt the programmer for information about the selected part. The data is not checked for consistency, so be careful to enter the numbers correctly. Once the information is entered, another prompt displays asking the user where the G-code is to be placed. There are 4 options within this prompt:

**1) SELECT/CREATE A PROGRAM**

If selected, another window will open prompting the user to select a program name. Simply highlight the desired name and press WRITE. This will cause VQC to output to the selected program. If the program already contains G-code, VQC will output to the beginning before the existing G-code and will not pass any M30s. The user also has the option to CREATE a new program by simply entering a program name using 'O'. Then by simply pressing WRITE again will cause VQC to output to the newly created program.

**2) ADD TO CURRENT PROGRAM**

If selected, the G-code generated by VQC will be inserted at the last position of the cursor within a program before entering VQC. If the program was empty, then any M30s. in the G-CODE section of the VQC template will be passed into the program. If the program contained anything, then NO M30s. will be passed.

**3) MDI**

If selected, the newly created VQC G-code will output to MDI. Note: Anything that may be in MDI will be overwritten.

**4) CANCEL**

If selected, the window will close and the focus returns to the parameter list. Note: the UNDO and CANCEL keys also perform this operation.

Additionally the program will be put into the Advanced Editor in order to double check the program. Verify the program by first running it in Graphics mode.

PROGRAM (EDIT)                    000001                    F1 KEY TURNS MENU ON/OFF

PROGRAM EDIT SEARCH MODIFY I/O HELP

000001

```
O00001 (G70 BOLT HOLE CIRCLE);
(I = 3.); 
(J = 40);
(NumHoles=8);
(ToolNo=1);
(WrkOfset=54);
(XPos=);
(YPos=);
(SpdleRpm=);
(DrlDpth=);
(Rplain=);
(Dwell=);
(Feedrate=54);
T1 M06;
C00 G90 G54 X-3. Y-3.;
S2000 M08;
G43 H01 Z1. M08;
G82 G98 Z-1. R0.1 P1. F10. L0;
```

G70 BOLT HOLE CIRCLE

## Extending Visual Quick Code

The Visual Quick Code system uses program O09997 to generate the icons and questions that the user sees, and the G code that is produced. Program O09997 consists entirely of comments. The comments contain keywords that the Visual Quick Code system understands. Some of the keywords are used to divide program O09997 into sections. The sections are hierarchical, that is the whole program is divided into CATEGORY sections, a CATEGORY section is divided into part TEMPLATE sections, and a TEMPLATE is divided into DIAGRAM, PARAMETER, and CODE sections.

Other keywords are used within sections to set the attributes of the object defined in that section. For example, within the PARAMETER section, we might see the following lines:

```
(#DC)
(LABEL DEPTH CUT)
(POSITION [20,6])
```

The first line defines the variable which is used to hold the information entered by the user. The second line defines the "label", to put on the visual portion of the screen. This "label" represents the variable on the first line for better clarity. "LABEL" tells the Visual Quick Code system to print anything following the keyword "LABEL" on the screen. The third line with the key word "POSITION" tells the Visual Quick Code where to put the label on the screen. The following is a complete list of the keywords used by Visual Quick Code.

**LIST OF KEYWORDS**

Keyword	Definition	Section
CATEGORY	The beginning of a CATEGORY section	
END CATEGORY	The end of a CATEGORY section	
TEMPLATE	The beginning of a TEMPLATE section	Category
END TEMPLATE	The end of a TEMPLATE section	Category
DIAGRAM	The beginning of a DIAGRAM section	Template
END DIAGRAM	The end of a DIAGRAM section	Template
LINE	Specifies a LINE in the DIAGRAM	Diagram
DATUM	Specifies a DATUM in the DIAGRAM	Diagram
ARROW	Specifies an ARROW in the DIAGRAM	Diagram
CW	Specifies a clockwise arc	Diagram
CCW	Specifies a counterclockwise arc	Diagram
CIRCLE	Specifies a circle	Diagram
THREAD	Specifies a thread in the DIAGRAM	Diagram
PRINT	Specifies a label in the DIAGRAM	Diagram
PARAMETER	The beginning of a PARAMETER section	Template
END PARAMETER	The end of a PARAMETER section	Template
LABEL	The LABEL attribute of a Parameter	Parameters
NO DECIMAL	Sets the NO DECIMAL attribute	Parameters
ONE PLACE	Sets the ONE PLACE attribute	Parameters
TWO PLACE	Sets the TWO PLACE attribute	Parameters
THREE PLACE	Sets the THREE PLACE attribute	Parameters
FOUR PLACE	Sets the FOUR PLACE attribute	Parameters
GCODE	The beginning of a G-code section	Template
END GCODE	The end of a Template section	Template

The Visual Quick Code system expects the keyword to appear in the Section column. If the keyword appears in a different area, Visual Quick Code will ignore it, or produce an error message because it mistook the keyword for one that it expected, and the text following the keyword did not fit into the Visual Quick Code pattern.

**PROGRAMMING VISUAL QUICK CODE**

The selections fall into two groups, categories and parts. The programmer first selects, from a list, the type of machining that will be used, for example, Drill/tap, bolt hole patterns, bore cycles, etc. This is the category group. Selecting one of these categories displays an illustrated group of parts. The programmer chooses from these illustrations the one that most resembles the desired part. Once chosen the control now prompts the programmer for the dimensions of the parts. Programming code is generated after the programmer enter the dimensions.

Program O09997 is the Visual Quick Code model. The program consists of several Category sections which define the categories available to the programmer.

The following is a basic outline of program O9997 using a top-down approach, becoming more and more specific. This is the way that Visual Quick Code is used. First the user sees a list of categories. After selecting a category, the user sees a list of parts. After selecting a part, the user sees what dimensions he or she can specify, and then the G-code is produced.

```

%
O09997
(CATEGORY)
...
(END CATEGORY)
(CATEGORY)
...
(END CATEGORY)
(CATEGORY)
...
(END CATEGORY)
(CATEGORY)
...
(END CATEGORY)
%

```

Each CATEGORY section in turn consists of several TEMPLATE sections. These sections define which parts are available to the user once a category has been selected.

```

%
O09997
(CATEGORY)
...
(TEMPLATE)
...
(END TEMPLATE)
(TEMPLATE)
...
(END TEMPLATE)
(TEMPLATE)
...
(END TEMPLATE)
(END CATEGORY)
%
```

Each TEMPLATE section consists of a DIAGRAM section, a PARAMETER section, and a GCODE section. The GCODE section is where the programming code is stored, but is missing some values that are entered by the programmer, via variables.

```

%
O09997
(CATEGORY)
...
(TEMPLATE)
...
(DIAGRAM)
...
(END DIAGRAM)
(PARAMETER)
...
(END PARAMETER)
(GCODE)
...
(END GCODE)
(END TEMPLATE)
(END CATEGORY)
%
```



## CATEGORY Section

The Category section is a collection of Part Templates. The necessary items are a beginning, a name, templates, and an end. CATEGORY marks the beginning of a Category section, and END CATEGORY marks the end. All of the templates that appear between the beginning and the end of a category belong to that category. NAME (your category name) should be the first line in the category section. The entered name will appear in the list of Visual Quick Code categories (this list appears when Visual Quick Code is first started).

Example:

```
%  
(CATEGORY)  
(NAME: Parts With holes)  
(TEMPLATE)  
...  
(DIAGRAM)  
...  
(END DIAGRAM)  
(PARAMETER)  
...  
(END PARAMETER)  
(GCODE)  
...  
(END GCODE)  
(END TEMPLATE)  
(END CATEGORY)  
%
```

## Part TEMPLATE Section

The Part TEMPLATE section specifies all the information about a typical part. This includes an illustration of the part and what variables can be entered to machine the part.

## DIAGRAM Section

The DIAGRAM section is the part of the program that creates the part illustration on the screen. The illustration can be drawn with lines, arcs, and jagged lines that represent threads. This drawing is both for the thumbnail sketch and the full-sized illustration. The labels that appear on the full-sized version of the part are specified in the PARAMETERS section (see Parameters description).

## DIAGRAM Coordinates

Each of the elements of the diagram must have a starting point and an ending point. The notation is [X,Y] where X is the horizontal coordinate and Y is the vertical coordinate. The best way to find out what the starting and ending points should be is to use graph paper. First sketch, on the graph paper, what is to appear on the screen. Then pick a point on the graph paper to be the origin, [0,0] (Any point will work, as the diagram will be scaled to fit wherever it is drawn). You can then determine the starting and ending points of all the lines, arcs (CW or CCW) and threads.

## DIAGRAM Elements

The elements that make up a diagrams are lines, arcs (CW or CCW) and Threads. For each element, the starting point is specified first, then the ending point.

The format for a line is:

```
(LINE [X1,Y1] [X2,Y2])
```

The format for a clockwise arc is:

```
(CW [X1,Y1] [X2,Y2] r)
```

where r is the radius of the arc.

The format for a counterclockwise arc is:

(CCW [X1,Y1] [X2,Y2] r)

where r is the radius of the arc.

The format for a jagged line to represent a thread is;

(THREAD [X1,Y1] [X2,Y2])

---

**NOTE:** Arcs (CW or CCW) may only cover 180 degrees, or half a circle. If an arc of more than 180 degrees is needed, another arc must be used.

## PARAMETERS Section

The PARAMETERS section lists all of the parameters that can be used to customize the standard part. Some of these would be the physical dimensions of the raw material and the part. Others would be tool and offset information, feed rates, and spindle speeds.

Each parameter begins with "#", which tells Visual Quick Code that value followed by the "#" will be the name for a specific variable. The format is:

```
(PARAMETERS)
(#your variable name)
(END PARAMETERS)
```

After a variable (parameter) has been specified, then any attributes of that particular parameter can be specified.

### The POSITION Attribute

If you wish the parameter to appear in the diagram, you must supply a position. The format is (POSITION [X,Y]), where X is the horizontal coordinate and Y is the vertical coordinate. These coordinates are relative to the coordinates specified in the DIAGRAM section. Typically, only physical dimensions of the part will have a POSITION attribute.

### Formatting Attributes

Several attributes are used to modify the value entered by the user. This is so that when the PARAMETER is used in the GCODE section, it will appear correctly. The Format column in the following table shows what would result from the G code template X#A, if the user enters 1 when asked for the value of parameter A. If none of the formatting attributes are used, the resulting G code would be X1.

Attribute	Format	Description
(NO DECIMAL)	X1	The value will appear in the final G-code output without a decimal point. Can be used for spindle speeds, tool numbers and offsets.
(ONE PLACE)	X.1	Numbers entered without a decimal point are automatically scaled to tenths.
(TWO PLACE)	X.01	Numbers entered without a decimal point are automatically scaled to hundredths.
(THREE PLACE)	X.001	Numbers entered without a decimal point are automatically scaled to thousandths.
(FOUR PLACE)	X.0001	Numbers entered without a decimal point are automatically scaled to ten-thousandths.



If more than one of these attributes are used with a single parameter, the results are not defined.

---

**NOTE:** Do not use more than one formatting attribute for a single parameter.

## G CODE Section

The GCODE section is responsible for producing the G code necessary to cut the specified part. Similar to the previous sections of program O09997, the GCODE section consists only of comments. The comments contain standard programming code, just as a user would type it into the editor, except that the end-of-block marker (;) is not used within the comments. The other difference is an extension similar to macro variables: in place of a numeric value, “#” followed by a letter may be entered. The letter represents the variable name of a parameter in the PARAMETERS section.

For example, one of the lines might be:

(X#A)

Which means, "X followed by the number entered for parameter A." For example, if the user entered 3.5 for parameter A, the resulting G-code would be:

X3.5;

Remember, the “#letter” combination can be used anywhere a number would be used; this means in expressions, as well as with simple codes. For example, (X [#A - #B]) is valid, as long as both A and B exist in the PARAMETER section.

---

**NOTE:** Be sure to use the parameter formatting attributes to make sure the G-code that is produced is valid. For example, “T101.;” is not a valid G-code, because of the decimal point. So if a line in the G-code section reads (T#E), then parameter E must have the NO\_DECIMAL attribute set.

### EXAMPLE PROGRAM

Below is an example of a simple program. This example is a complete O09997 template and is provided to help complete what you have just read.

```
%  
O9997  
(CATEGORY)  
  
(TEMPLATE)  
(NAME SQUARE MILLING )  
  
(DIAGRAM)  
(LINE [0,0] [40,0]) (CENTER LINES)  
(LINE [0,0] [0,37])  
(DATUM [34,31])  
  
(LINE [4,31] [34,31])  
(LINE [34,31] [34,3])  
(LINE [34,3] [4,3])  
(LINE [4,3] [4,31])  
  
(LINE [4,32] [4,34])  
(LINE [34,32] [34,34])  
(LINE [35,31] [37,31])  
(LINE [35,3] [37,3])
```

```

(ARROW [16,33] [4,33])
(ARROW [22,33] [34,33])
(ARROW [36,17] [36,31])
(ARROW [36,13] [36,3])

(END DIAGRAM)

(PARAMETERS)
(#ToolNo)
(NO DECIMAL)

(#WrkOffset)
(NO DECIMAL)

(#CuterRad)

(#SpdleRpm )
(NO DECIMAL)

(#DpthCut)

(#XDist)
(POSITION [17,34])

(#YDist)
(POSITION [35,16])

(#Feedrate)

(END PARAMETERS)

(GCODE)
(T#ToolNo M06)
(G00 G90 G#WrkOffset X[#CuterRad+.1] Y[#CuterRad+.1])
(S#SpdleRpm M03)
(G43 H#ToolNo Z1. M08)
(G01 Z-#DpthCut F50.)
(G01 G41 D#ToolNo X0)
(G01 Y-#YDist F#Feedrate)
(G01 X-#XDist)
(G01 Y0)
(G01 X|0+[#CuterRad+.1])
(G01 G40 Y[#CuterRad+.1])
(G00 Z1. M09)
(G28 G91 Z0 M05)
(M30)

(END GCODE)

(END TEMPLATE)

(END CATEGORY)
%

```



### 3.23 ADVANCED EDITOR

The HAAS Advanced Editor gives the user the ability to view and edit two CNC programs at a time. This makes it easier to modify existing programs and to create new ones. The editor includes pull-down menus that allow the user to access the features of the editor. Additionally, a context-sensitive help function is available to provide information on all of the editor's features.

The Advanced Editor is entered by pressing the EDIT key. The user can alternate between the Advanced Editor, the 40 column editor, and Quick Code with successive presses of the PRGRM/CONVRS key.

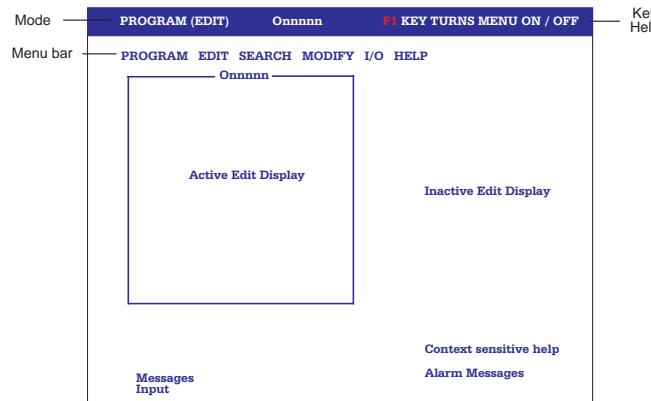
Whenever the pull-down menu system is active, the current menu is pulled down and one item is highlighted. The user can then use the jog handle to scroll through the menu items. The user can also use the up and down arrow keys to scroll through the items of that menu, or use the left and right arrow keys to open other menus.

The UNDO key is used to deactivate the pull-down menu system. If UNDO is pressed after invoking an executing function from a pull-down menu, it will abort that function.

The EDIT key can be used to "switch", left or right, between the two programs that have been selected to edit.

Pressing the F4 key will open another copy of the current program in the Advanced Editor. The user can quickly edit two different locations in the same program by pressing F4, moving to the second location, and then using the EDIT key to move back and forth between the two locations. If the user enters 'Onnnnn' and then presses F4, program Onnnnn is opened in the inactive window.

The following figure illustrates the layout of the advanced editor.



*The advanced editor screen layout.*

The advanced editor screen is divided into the following areas:

- **Mode and Control Status** - contain the Current Display page, Operating mode and Control status.
- **Messages** - where control status messages are displayed. This area is used to display prompts for user input in the advanced editor, and any alarm messages.
- **Input** - where the user's input is displayed.
- **Menu Bar** - contains the pull-down menu banner.
- **Key Help** - contains short immediate help messages. Shows the user the most important keys that can be used in the current context or operation.



VS-Series

- **Left/Right Side Display** - shows the active and inactive programs. When first entering the editor, the current program will be displayed on the left side and the right side will be blank. These areas can display another program or display program lists and help pages, when the appropriate menu item is selected from the pull-down menu.
- **Context-Sensitive Help** - where context-sensitive help is displayed when you cursor onto a menu item.

## ADVANCED EDITOR FEATURES

This section briefly describes each feature of the advanced editor, each item found in the pull-down menus, and any prompts that might appear.

### CONTEXT SENSITIVE HELP (The F1 key)

While in the advanced editor, press the F1 key to get into the menus, use the arrow keys or the jog handle to highlight the menu item. The help text for that item will appear in the lower right corner of the screen. PAGE UP and PAGE DOWN keys are used to view the help text. Press UNDO to go back to the menu, or RESET to exit the menus entirely.

## THE PROGRAM MENU

### Create New Program

This menu item will create a new program, providing there is room in the program directory and enough memory is available. Enter a program name (Onnnnn) in the range of 0 through 99999 that is not already in the program directory.

### Select Program From List

The HAAS control maintains a directory of programs that the user can select. Select this menu item to edit a program that exists in the directory.

When this menu item is selected, a list of programs is presented for viewing. Scroll through the list by using the cursor keys or the jog handle. Pressing the ENTER or SELECT PROG key will select the program that is highlighted and will replace the selection list with the selected program. The selected program is now active, and the previously active program will appear on the inactive edit screen.

### Duplicate Active Program

This menu item will create a new program, copy the contents of the current program into it, rename it as specified, and make it the active program.

### Delete Program From List

This menu item will delete a program from the program directory. A list of all programs is presented, with 'ALL' at the end.

To delete a single program, cursor to the program number and press the ENTER key. A prompt will ask for a confirmation of the deletion operation. Enter 'Y' to delete the highlighted program. If any other key is pressed, the program will not be deleted. After a program is deleted, the list of programs will again be presented.

To delete all programs, cursor to 'ALL' and press ENTER. Confirm deletion of all programs by pressing 'Y'. When all programs are deleted, program O00000 is created, and it is made the active program.

### Switch To Left Or Right Side

This will switch left and right between the two programs, to make the active program inactive and the inactive program active. If there is no inactive program, then nothing happens. There are only two possible displays: one on the left and one on the right. The inactive display is used when a second program is selected or created.



## THE EDIT MENU

### **Undo**

The last insert, delete, or alter (simple edit) operation will be undone. Pressing UNDO again will restore the previous editing operation, up to the last 9 editing operations. If a block has been selected, choosing this item will simply exit block select mode without undoing anything.

### **Select Text**

This item will set the start point of the block selection. To set the end point, scroll up or down to the desired place, and press the F2 or ENTER key. The selected block will then be highlighted. To deselect the block, press UNDO. This function works the same as in the 40 column editor, except this menu option is used to start selecting text, instead of the F1 key. Either the ENTER or F2 key can be used to end the selection. The following prompt will appear when this item is selected:

SCROLL UP/DOWN, PRESS ENTER OR F2 (to complete the text selection)

### **Move Selected Text**

All selected text will be moved to the line following the cursor arrow position.

### **Copy Selected Text**

All selected text will be copied to the line following the cursor arrow position.

### **Delete Selected Text**

This item deletes any selected block. If no block is selected, the currently highlighted item is deleted. The UNDO key will restore any deleted comment or individual commands, but will not restore any blocks of code that were deleted. The DELETE key deletes individual characters from comments and is the hot key for this menu item.

### **Cut Selection To Clipboard**

All selected text will be moved from the current program to a new program called the clipboard. Any previous contents of the clipboard are deleted. The program number (8998) for the clipboard is specified by Parameter 226 and can be altered, if necessary.

### **Copy Selection To Clipboard**

All selected text will be copied from the current program to a new program called the clipboard. Any previous contents of the clipboard are deleted. The program number (8998) for the clipboard is specified by Parameter 226 and can be altered, if necessary.

### **Paste From Clipboard**

The contents of the clipboard are copied into the current program at the line following the current cursor position.

## THE SEARCH MENU

### **Find Text**

This menu item will search for one or more G-Code items in the current program. The search can be performed in either the forward or backward direction from the current cursor location. If the item is found, the cursor will be positioned on it.

### **Find Again**

This menu item will search the current program for the last block of code that was searched for. It will begin to search at the current cursor location, in the direction that was specified in the previous search. This function will search both selected and unselected blocks.



VS-Series

### **Find And Replace Text**

This menu item will search the current program for one or more occurrences of a specified G-Code item and optionally replace each (or all) with another G-Code item. The search can be performed in either the forward or backward direction from the current cursor location. As each G-Code item is found, the cursor will be positioned on it, and a prompt will ask whether to replace the item, continue the search, both, or neither.

## **THE MODIFY MENU**

### **Remove All Line Numbers**

This menu item will automatically remove all unreferenced N-Codes from the edited program. If a block is selected, only the G-Code blocks contained within it will be affected.

### **Renumber All Lines**

This menu item will either renumber all selected blocks in the program or, if a block is selected, renumber only those G-Code blocks contained in that block.

### **Renumber By Tool**

Searches selected text, or the entire program, for T codes and renames program blocks grouped by T code.

### **Reverse + & - Signs**

This menu item will reverse the signs of all numeric values associated with one or more address codes in the program. If a certain block is selected, only the address codes in the selected block will be affected.

### **Reverse X & Y**

When this menu item is selected, all X address codes in the program will be changed to Y address codes, and Y's to X. If a block is selected, only the address codes contained within it will be affected.

## **THE I/O MENU**

### **Send RS-232**

This menu will send program(s) that are selected from the program directory to the RS-232 port. When this menu item is selected, a list of all the programs in memory is presented, with 'ALL' at the end.

To select a program, cursor to the program number and press the INSERT key. A highlighted space will appear before the program to indicate it has been selected. Pressing INSERT again will deselect the program, and the highlighted space will disappear. The DELETE key can be used to deselect all selected programs. When the cursor is on "ALL", all the programs are selected regardless of highlighting

To send the selected program(s), press the ENTER key. If more than one program or 'ALL' is selected, the data will be sent with one "%" at the beginning of the stream and one at the end.

### **Receive RS-232**

This menu item will receive program(s) from the RS-232 serial port. The program(s) will then be stored in the CNC memory with the corresponding Onnnnn program number(s).

On LISTPROG "ALL" must first be highlighted before using this menu item. The Onnnnn program numbers will be entered automatically from the input stream data. Note, "ALL" must be reselected on the LIST PROG screen after each file transfer.



VS-Series

### **Send Disk**

This menu item will send program(s) to the disk. When this menu item is selected, a list of all the programs in memory is presented, with 'ALL' at the end.

To select a program, cursor to the program number and press the INSERT key. A highlighted space will appear before the program to indicate it has been selected. Pressing INSERT again will deselect the program, and the highlighted space will disappear. The DELETE key can be used to deselect all selected programs. When the cursor is on "ALL", all programs are selected regardless of highlighting.

**ENTER DISK FILENAME:** Type in the desired disk filename (in standard PC DOS format) for the disk file being sent, then press the ENTER key. If more than one program or "ALL" is selected, the data will be sent with one "%" at the beginning of the stream and one at the end. If a filename is not entered, the controller will send each selected file separately using the Onnnnn program number as the filename.

### **Receive Disk**

This menu item will receive programs from the disk. The program(s) will then be stored in the CNC memory with the corresponding Onnnnn program number(s).

**ENTER DISK FILENAME:** Type in the filename (in standard PC DOS format) of the disk file being received, then press the ENTER key.

### **Disk Directory**

This menu item will display the directory of the disk, with the first entry highlighted. To select a file, press the up and down arrow buttons or use the jog handle. To load a file, select it and press the WRITE key. The G-code programs in the file will be loaded into memory.

If there is insufficient memory for the entire file, Alarm 429 will be generated, and only a partial directory will be displayed.

## **THE F1:HELP MENU**

### **How To Use The Editor**

The on-line help is displayed whenever a menu is accessed. The help manual gives a brief description of the editor and its features. The up and down arrows and the jog handle control the menus, and the Page Up, Page Down Home and End keys are used to scroll through the help display. In addition, if the F1 key is pressed during the use of one of the menu options, the help is likewise displayed. Pressing F1 again will exit the help display. Pressing the UNDO key returns to the active program.

### **Quick Code**

Selecting this menu item will place Quick Code on the inactive side of the editor. All Quick Code functions are now available to the user. Refer to the Quick Code section of the Operator's Manual for a full description of Quick Code. Pressing the EDIT key will exit Quick Code.



VS-Series

## ADVANCED EDITOR SHORTCUTS

Pressing these keys, when in the Advanced Editor display, will quickly get you to these menu items without having to press the F1 key and cursoring to that selection.

HOT KEYS	DESCRIPTION OF HOT KEY
SELECT PROG	Will quickly bring up the program list on the inactive side of edit display to SELECT PROGRAM FROM LIST.
F2	This key will begin to SELECT TEXT and define the starting line of a block to be edited. Scroll down to the last line in the block definition, and press the F2 or WRITE key. The selected block of text will then be highlighted.
EDIT	This key can be used to SWITCH TO LEFT OR RIGHT SIDE between two programs that have been selected to edit.
F4	Pressing F4 will open another copy of the same program on the other side of the Advanced Editor display. The user can quickly edit two different locations in the same program. The edit key will switch you back and forth and update between the two programs.
INSERT	If you enter the program number (Onnnn) and then press F4 or the arrow down key, that program will be brought up on the other side of the Advanced Editor.
	INSERT can be used to COPY SELECTED TEXT in a program to the line after where you place the cursor arrow point.
ALTER	ALTER can be used to MOVE SELECTED TEXT in a program to the line after where you place the cursor arrow point.
DELETE	DELETE can be used to DELETE SELECTED TEXT in a program.
UNDO	If a block has been selected, pressing UNDO will simply exit a block definition.
SEND RS232	Pressing the SEND RS-232 key will activate that I/O menu selection.
RECV RS232	Pressing RECV RS-232 key will activate that I/O menu selection.
ERASE PROG	Pressing the ERASE PROG key will activate that I/O menu selection. This will bring up program list on the inactive side of edit display for you to cursor to a program and delete it.



### 3.24 MACROS

#### INTRODUCTION

This control function is optional. If you would like further information on installing this feature please call Haas Automation or your dealer for more information.

This is an introduction to macros as implemented on the HAAS CNC controls. MACROS adds capabilities and flexibility to standard G-code programming that allow the programmer to better define a tool path in a quicker and more natural way. With few exceptions, MACROS, as implemented on the HAAS controls, is compatible with FANUC 10M and 15M controls. Macro features not included in the current release are listed at the end of this section. Programmers already familiar with macro programming will want to review this section in order to avoid unnecessary work.

In traditional CNC programming, a program consists of subroutines that CANNOT be changed or altered except by editing individual values with an editor. MACROS allows the capability to program subroutines where the tool path or location of the tool path is changed, depending on the values contained within variables set by the programmer. These variables can be passed to the subroutine as parameters, or the values can reside in what are called *global variables*.

What this all means is that a programmer can create a collection of subroutines that have been fully debugged. These programs can be used as high level tools that can enhance programmer and machinist productivity. MACROS is not intended to replace modern CAD/CAM software, but it can and has improved machine productivity for those who use it.

Here are a few examples of the applications for MACROS. Rather than give macro code here, we will outline the general applications that MACROS can be used for.

#### Tools For Immediate, On-Table Fixturing

Many setup procedures can be semi-automated to assist the machinist. Tools can be reserved for immediate situations that were not anticipated during tool design. For instance, suppose a company uses a standard clamp with a standard bolt hole pattern. If it is discovered, after setup, that a fixture will need an additional clamp and if macro subroutine 2000 has been programmed for drilling the bolt pattern of the clamp, then the following two-step procedure is all that is needed for adding the clamp to the fixture.

1. Determine X, Y, and Z coordinates and angle where the clamp is to be placed by jogging the machine to the proposed clamp position and reading the position coordinates from the machine display.
2. Execute the following command in MDI mode:  
`G65 P2000 X??? Y??? Z??? A??? ;`  
Where ??? are the values determined in Step 1.

Here, macro 2000 (not shown) takes care of all the work since it was designed to drill the clamp bolt hole pattern at the specified angle of A. Essentially, the machinist has created his own custom canned cycle.

## Simple Patterns That Are Repeated Over And Over Again In The Shop

Patterns that recur over and over again can be parameterized and kept around for easy, immediate use. For example:

- 1) Bolt hole patterns.
- 2) Slotting.
- 3) Angular patterns, 5 holes at 30 degrees 1 inch apart.
- 4) Specialty milling such as soft jaws.
- 5) Matrix Patterns, 12 across and 15 down.
- 6) Flycutting a surface, 12 inches by 5 inches using a 3 inch fly cutter.

## Automatic Offset Setting Based On The Program

With macros, coordinate offsets can be set in each program so that setup procedures become easier and less error-prone.

### Probing

Probing enhances the capabilities of the machine in many ways. Below is just a hint of the possibilities.

- 1) Profiling of a part to determine unknown dimensions for later machining.
- 2) Tool calibration for offset and wear values.
- 3) Inspection prior to machining to determine material allowance on castings.
- 4) Inspection after machining to determine parallelism and flatness values as well as location.

Macros allow less experienced personnel to operate the machine. Conditions can be detected and custom operator messages or alarms can be displayed on the console to notify the operator.

### *MACRO SUBROUTINE CALL (G65)*

G65 is the command that calls a subroutine with the ability to pass arguments to it. The format follows.

[N#####] G65 P##### [L#####] [arguments] ;

Anything enclosed in brackets is optional. This should not be confused with expression brackets that are explained below. The G65 command requires a **P** address parameter corresponding to any program number currently in memory. When the optional **L** address is used the macro call is repeated the specified number of times.

In Example 1, subroutine 1000 is called once with no parameters passed to the routine. G65 calls are similar to, but not the same as, M98 calls. Up to four G65 calls can be made at the same time, (Nesting four deep).

<b>Example 1:</b>	G65 P1000 ;	(Call subroutine 1000 as a macro)
	M30 ;	(Program stop)
	O1000 ;	(Macro Subroutine)
	...	
	M99 ;	(Return from Macro Subroutine)

In Example 2, subroutine 9010 is designed to drill a sequence of holes along a line whose slope is determined by the X and Y arguments that are passed to it in the G65 command line. The Z drill depth is passed as **Z**, the feed rate is passed as **F**, and the number of holes to be drilled is passed as **T**. The line of holes is drilled starting from the current tool position when the macro subroutine is called.



**Example 2:**

G00 G90 X1.0 Y1.0 Z.05 S1000 M03;	(Position tool)
G65 P9010 X.5 Y.25 Z.05 F10. T10 ;	(Call 9010)
G28 M30 ;	
O9010 ;	(Diagonal hole pattern)
F#9 ;	(F=Feed rate)
WHILE [#20 GT 0] DO1;	(Repeat T times)
G91 G81 Z#26 ;	(Drill To Z depth)
#20=#20-1 ;	(Decrement counter)
IF [#20 EQ 0] GOTO5 ;	(All holes drilled)
G00 X#24 Y#25 ;	(Move along slope)
N5 END1 ;	
M99 ;	(Return to caller)

### ALIASING

Aliasing is a means of assigning a G code to a G65 P##### sequence. For instance, in Example 2 it would be easier if one could write:

G06 X.5 Y.25 Z.05 F10. T10 ;

Here, we have substituted an unused G code, G06 for G65 P9010. In order for the above block to work we must set the parameter associated with subroutine 9010 to 06 ( Parameter 91).

Note that G00, G65, G66, and G67 can not be aliased. All other codes between 1 and 255 can be used for aliasing.

Program numbers 9010 through 9019 are reserved for G code aliasing. The following table lists which HAAS parameters are reserved for macro subroutine aliasing.

### Requires Macros Option

Haas Parameter	O Code
91	9010
92	9011
93	9012
94	9013
95	9014
96	9015
97	9016
98	9017
99	9018
100	9019

Setting an aliasing parameter to 0 disables aliasing for the associated subroutine. If an aliasing parameter is set to a G-code and the associated subroutine is not in memory, then an alarm will be given.

### M code Aliasing

Haas Parameter	M Macro Call
81	9000
82	9001
83	9002
84	9003
85	9004
86	9005
87	9006
88	9007
89	9008
90	9009

## MACRO ARGUMENTS

The arguments in a G65 statement are a means of sending values to and setting the local variables of a called macro subroutine.

In Example 2 above, the arguments X and Y are passed to the macro subroutine local variables. Local variable #24 is associated with X and is set to 0.5. Similarly, Local variable #25 is associated with Y and is set to 0.25.

The following two tables indicate the mapping of the alphabetic address variables to the numeric variables used in a macro subroutine.

### Alphabetic Addressing

Address:	A	B	C	D	E	F	G	H	I	J	K	L	M
Variable:	1	2	3	7	8	9	-	11	4	5	6	-	13
Address:	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Variable:	-	-	-	17	18	19	20	21	22	23	24	25	26

### Alternate Alphabetic Addressing

Address:	A	B	C	I	J	K	I	J	K	I	J
Variable:	1	2	3	4	5	6	7	8	9	10	11
Address:	K	I	J	K	I	J	K	I	J	K	I
Variable:	12	13	14	15	16	17	18	19	20	21	22
Address:	J	K	I	J	K	I	J	K	I	J	K
Variable:	23	24	25	26	27	28	29	30	31	32	33

Arguments accept any floating point value to four decimal places. If you are in metric, the control will assume thousandths (.000). In Example 3 below, local variable #7 will receive .0004.

If a decimal is not included in an argument value, such as:

G65 P9910 A1 B2 C3

The values are passed to macro subroutines according to the following table:

### Integer Argument Passing (no decimal point)

Address:	A	B	C	D	E	F	G	H	I	J	K	L	M
Variable:	.001	.001	.001	1.	1.	1.	-	1.	.0001	.0001	.0001	1.	1.
Address:	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Variable:	-	-	-	.0001	.0001	1.	1.	.0001	.0001	.0001	.0001	.0001	.0001

All 33 local macro variables can be assigned values with arguments by using the alternate addressing method. The following example shows how one could send two sets of coordinate locations to a macro subroutine. Local variables #4 through #9 would be set to .0001 through .0006 respectively.

**Example 3:** G65 P2000 I1 J2 K3 I4 J5 K6 ;

The following letters cannot be used to pass parameters to a macro subroutine: G, L, N, O or P.



## MACRO CONSTANTS

Constants are floating point values placed in a macro expression. They can be combined with addresses A...Z or they can stand alone when used within an expression. Examples of constants are .0001, 5.3 or -10.

## MACRO VARIABLES

There are three categories of macro variables: *system* variables, *global* variables, and *local* variables.

### Variable Usage

All variables are referenced with a number sign (#) followed by a positive number. Examples are: #1, #101, and #501.

Variables are decimal values that are represented as floating point numbers. If a variable has never been used, it can take on a special “undefined” value. This indicates that it has not been used. A variable can be set to undefined with the special variable #0. #0 has the value of undefined or 0.0 depending on the context it is used in. More about this later. Indirect references to variables can be accomplished by enclosing the variable number in brackets.

#[<expression>]

The expression is evaluated and the result becomes the variable accessed. For example:

```
#1=3;  
#[#1]=3.5 + #1;
```

This sets the variable #3 to the value 6.5.

Variables can be used in place of G-code address constants where “address” refers to the letters A..Z.

In the block

**N1 G0 G90 X1.0 Y0 ;**

the variables can be set to the following values:

```
#7=0;  
#11=90;  
#1=1.0;  
#2=0.0;
```

and the block replaced by:

**N1 G#7 G#11 X#1 Y#2 ;**

The values in the variables at runtime are used as the address values.

## Local Variables

Local variables range between #1 and #33. A set of local variables is available at all times. When a call to a subroutine with a G65 command is executed, the local variables are saved and a new set is available for use. This is called “nesting” of the local variables. During a G65 call, all of the new local variables are cleared to undefined values and any local variables that have corresponding address variables in the G65 line are set to the G65 line values. Below is a table of the local variables along with the address variable arguments that change them.

<b>Variable:</b>	1	2	3	4	5	6	7	8	9	10	11
<b>Address:</b>	A	B	C	I	J	K	D	E	F	H	J
<b>Alternate:</b>							I	J	K	I	J
<b>Variable:</b>	12	13	14	15	16	17	18	19	20	21	22
<b>Address:</b>	M					Q	R	S	T	U	V
<b>Alternate:</b>	K	I	J	K	I	J	K	I	J	K	I
<b>Variable:</b>	23	24	25	26	27	28	29	30	31	32	33
<b>Address:</b>	W	X	Y	Z							
<b>Alternate:</b>	J	K	I	J	K	I	J	K	I	J	K

Note that variables 10, 12, 14..16 and 27..33 do not have corresponding address arguments. They can be set if a sufficient number of I, J and K arguments are used as indicated above in the section about arguments.

Once in the macro subroutine, the local variables can be read and modified by referencing the variable numbers 1..33.

When the **L** argument is used to do multiple repetitions of a macro subroutine, the arguments are set only on the first repetition. This means that if local variables 1..33 are modified in the first repetition, then the next repetition will have access only to the modified values. Local values are retained from repetition to repetition when the **L** address is greater than 1.

Calling a subroutine via an M97 or M98 does not nest the local variables. Any local variables referenced in a subroutine called by an M98 are the same variables and values that existed prior to the M97 or M98 call.

## Global Variables

Global variables are variables that are accessible at all times. There is only one copy of each global variable. Global variables occur in two ranges: 100..199, 500..699 and 800...999. The global variables remain in memory when power is turned off. They are not cleared as in the FANUC controls.

## System Variables

System variables give the programmer the ability to interact with a variety of control parameters and settings. By setting a system variable, the function of the control can be modified or altered. By reading a system variable, a program can modify its behavior based on the value in the variable. Some system variables have a READ ONLY status. This means that they can not be modified by the programmer. A brief table of currently implemented system variables follows with an explanation of their use.

**VARIABLES****USAGE**

#0	Not a number (read only)
#1-#33	Macro call arguments
#100-#199	General purpose variables saved on power off
#500-#699	General purpose variables saved on power off
#700-#749	Hidden variables for internal use only.
#800-#999	General purpose variables saved on power off
#1000-#1063	64 discrete inputs (read only)
#1064-#1068	Maximum axis loads.
#1080-#1087	Raw analog to digital inputs (read only)
#1090-#1098	Filtered analog to digital inputs (read only)
#1094	Spindle load with OEM spindle drive (read only)
#1098	Spindle load with Haas vector drive (read only)
#1100-#1139	40 discrete outputs
#1140-#1155	16 extra relay outputs via multiplexed output
#1264-#1268	Maximum axis loads.
#1601-#1800	Number of Flutes of tools #1 through 200
#1801-#2000	Maximum recorded vibrations of tools 1 through 200
#2001-#2200	Tool length offsets
#2201-#2400	Tool length wear
#2401-#2600	Tool diameter/radius offsets
#2601-#2800	Tool diameter/radius wear
#3000	Programmable alarm
#3001	Millisecond timer
#3002	Hour timer
#3003	Single block suppression
#3004	Override control
#3006	Programmable stop with message
#3011	Year, month,day
#3012	Hour, minute, second
#3020	Power on timer (read only)
#3021	Cycle start timer (read only)
#3022	Feed timer (read only)
#3023	Present part timer (read only)
#3024	Last complete part timer (read only)
#3025	Previous part timer (read only)
#3026	Tool in spindle (read only)
#3027	Spindle RPM (read only)
#3030	Single Block (read only)
#3031	Dry Run (read only)
#3032	Block Delete (read only)
#3033	Opt Stop (read only)
#3201-#3400	Actual Diameter for tools 1 through 200
#3401-#3600	Programmable coolant positions for tools 1 through 200
#3901	M30 count 1
#3902	M30 count 2
#4000-#4021	Previous block group codes
#4101-#4126	Previous block address codes

**NOTE:** Mapping of 4101 to 4126 is the same as the alphabetic addressing of "Macro Arguments" section; e.g. the statement x1.3 sets variable #4124 to 1.3.

#5000-#5005	Previous block end position
#5020-#5025	Present machine coordinate position
#5041-#5045	Present work coordinate position
#5061-#5065	Present skip signal position - X,Y,Z,A,B
#5081-#5085	Present tool offset
#5201-#5205	Common offset
#5221-#5225	G54 work offsets
#5241-#5245	G55 work offsets
#5261-#5265	G56 work offsets
#5281-#5285	G57 work offsets
#5301-#5305	G58 work offsets
#5321-#5325	G59 work offsets
#5401-#5500	Tool feed timers (seconds)
#5501-#5600	Total tool timers (seconds)
#5601-#5699	Tool life monitor limit
#5701-#5800	Tool life monitor counter
#5801-#5900	Tool load monitor maximum load sensed so far
#5901-#6000	Tool load monitor limit
#6001-#6277	Settings (read only)
#6501-#6999	Parameters (read only)

---

**NOTE:** The low order bits of large values will not appear in the macro variables for settings and parameters

#7001-#7005	G110 additional work offsets
#7021-#7025	G111 additional work offsets
#7041-#7045	G112 additional work offsets
#7061-#7065	G113 additional work offsets
#7081-#7085	G114 additional work offsets
#7101-#7105	G115 additional work offsets
#7121-#7125	G116 additional work offsets
#7141-#7145	G117 additional work offsets
#7161-#7165	G118 additional work offsets
#7181-#7185	G119 additional work offsets
#7201-#7205	G120 additional work offsets
#7221-#7225	G121 additional work offsets
#7241-#7245	G122 additional work offsets
#7261-#7265	G123 additional work offsets
#7281-#7285	G124 additional work offsets
#7301-#7305	G125 additional work offsets
#7321-#7325	G126 additional work offsets
#7341-#7345	G127 additional work offsets
#7361-#7365	G128 additional work offsets
#7381-#7385	G129 additional work offsets



## SYSTEM VARIABLES IN-DEPTH

This section fully describes system variables.

### 1-Bit Discrete Inputs

Inputs designated as "spare" can be connected to external devices and used by the programmer.

### 1-Bit Discrete Outputs

The HAAS control is capable of controlling up to 56 discrete outputs. However, a number of these outputs are already reserved for use by the HAAS controller.

---

**CAUTION!**

Do not use outputs that are reserved by the system. Using these outputs may result in injury or damage to your equipment.

---

The user can change the state of these outputs by writing to variables designated as "spare". If the outputs are connected to relays, then an assignment of "1" sets the relay. An assignment of "0" clears the relay.

Referencing these outputs will return the current state of the output and this may be the last assigned value or it may be the last state of the output as set by some user M code. For example, after verifying that output #1108 is "spare":

```
#1108=1;                                (Turns #1108 relay on)
#101=#3001+1000;                          (101 is 1 second from now)
WHILE [[#101 GT #3001] AND [#1109 EQ 0]] D01
END1                                         (Wait here 1 second or until relay #1109 goes high)
#1108=0;                                    (Turns #1108 relay off)
```

The number of outputs available to the user and where user M codes are mapped is model dependent. If your control is not equipped with the M-code relay board, then M21 through M28 will be mapped from #1132-#1139. If you have equipment with the M-code relay board installed, see the 8M option section for information and instructions.

You should always test or dry run programs that have been developed for macros that is running with new hardware.

### Maximum axis loads available in variables

The following variables are used to contain the maximum load values for each axis. They can be cleared by cycling power to the machine or by setting the macro to zero in a program (for example, #1064=0;).

1064 = X axis 1264 = C axis  
1065 = Y axis 1265 = U axis  
1066 = Z axis 1266 = V axis  
1067 = A axis 1267 = W axis  
1068 = B axis 1268 = T axis

## Tool Offsets

HAAS macros have been implemented with FANUC control memory C option in mind. This means that each tool offset has a length (H) and radius (D) along with associated wear values.

- #2001-#2200 H geometry offsets (1-200) for length.
- #2200-#2400 H geometry wear (1-200) for length.
- #2401-#2600 D geometry offsets (1-200) for diameter.
- #2601-#2800 D geometry wear (1-200) for diameter.

## Programmable Messages

#3000 ALARMS can be programmed. A programmable alarm will act just like HAAS internal alarms. An alarm is generated by setting the macro variable #3000 to a number between 1 and 999.

#3000= 15 (MESSAGE PLACED INTO ALARM LIST);

When this is done, ALARM flashes in the lower right hand corner of the display and the text in the next comment is placed into the alarm list. The alarm number (in this example, 15) is added to 1000 and used as an alarm number. If an alarm is generated in this manner all motion stops and the program must be reset to continue. Programmable alarms can always be identified in alarm history because the alarm numbers range between 1000 and 1999.

The first 34 characters of the comment will be used for the alarm message. The first 15 characters of the comment will be displayed on the lower left corner of the screen.

## Timers

HAAS macros supports access to two timers. These timers can be set to a value by assigning a number to the respective variable. A program can then later read the variable and determine the time passed since the timer was set. Timers can be used to emulate dwell cycles, determine part to part time or wherever time dependent behavior is desired.

- #3001 MILLISECOND TIMER - The millisecond timer is updated every 20 milliseconds and thus activities can be timed with an accuracy of only 20 milliseconds. At POWER ON, the millisecond timer is reset. The timer has a limit of 497 days. The whole number returned after accessing #3001 represents the number of milliseconds.
- #3002 HOUR TIMER - The hour timer is similar to the millisecond timer except that the number returned after accessing #3002 is in hours. The hour and millisecond timers are independent of each other and can be set separately.



## System Overrides

#3003 Variable 3003 is the Single Block Suppression parameter. It overrides the Single Block function in G-code. In the example below, suppression of Single Block is initiated when #3003 is set equal to 1. After M3003 is set =1, each G-code instruction block (lines 2-4) are executed continuously even though the Single Block function is enabled. When #3003 is set equal to zero, the operator of Single Block will resume as normal. That is, the user must press Cycle Start to initiate each new code block (lines 6-8).

```
#3003=1;
G54 G00 G90 X0 Y0;
G81 R0.2 Z-0.1 F20 L0;
S2000 M03;
#3003=0;
T02 M06;
G83 R0.2 Z-1 F10. L0;
X0. Y0.;
```

#3004 Variable #3004 is a bitmapped variable that overrides specific control features during runtime.

The first bit disallows FEED HOLD from the keypad. If you do not want feed hold to be executed during any section of code, then bracket that code with assignments to variable #3004. Assigning "1" to #3004 disables the console's feed hold button. Assigning "0" to #3004 re-enables the FEED HOLD button. For example:

Approach code	(FEED HOLD allowed)
#3004=1;	(Disables FEED HOLD button)
Non-stoppable code	(FEED HOLD not allowed)
#3004=0;	(Enables FEED HOLD button)
Depart code	(FEED HOLD allowed)

The following is a map of variable #3004 bits and the associated overrides. E=Enabled D=Disabled

#3004	FEED HOLD	FEED RATE OVERRIDE	EXACT STOP CHECK
0	E	E	E
1	D	E	E
2	E	D	E
3	D	D	E
4	E	E	D
5	D	E	D
6	E	D	D
7	D	D	D

### Programmable Stop

#3006 Stops can be programmed. A programmable stop acts like an M00. In the following example, when the assignment statement is executed, the first 15 characters of the comment are displayed in the messaging area on the lower left part of the screen above the command input line. The control stops and waits for a cycle start from the operator. Upon cycle start, operation continues with the next block after the assignment statement.

```
IF [#1 EQ #0] THEN #3006=101(ARG.A REQUIRED);
```



VS-Series

### Last Block (MODAL) Group Codes

- #4001-#4021      The grouping of G codes permits more efficient processing. G codes with similar functions are usually under the same group. For instance, G90 and G91 are under group 3. Variables have been set aside to store the last or default G code issued for any of 21 groups. By reading the group code, a macro program can change its behavior based on the contents of the group code. If 4003 contains 91, then a macro program could determine that all moves should be incremental rather than absolute. There is no associated variable for group zero, group zero G codes are NON-modal.

### Last Block (MODAL) Address Data

- #4101-#4126      Address codes A..Z (excluding G) are also maintained as modal values. The modal information represented by the last block interpreted by the lookahead process is contained in variables 4101 through 4126. The numeric mapping of variable numbers to alphabetic addresses corresponds to the mapping under alphabetic addresses. For instance, the value of the previously interpreted **D** address is found in #4107 and the last interpreted **I** value is #4104.

### Last Target Position

- #5001-#5005      The final programmed point, target position, for the most recent motion block can be accessed through variables #5001-#5005, X, Y, Z, A, and B, respectively. Values are given in the current work coordinate system and can be used while the machine is in motion.

### Current Machine Coord Position

- #5021-#5025      The current position in machine coordinates can be obtained through #5021-#5025, X, Y, Z, A, and B, respectively. The values CANNOT be read while the machine is in motion. #5023 (Z) represents the value after tool length compensation has been applied.

### Current Work Coord Position

- #5041-#5045      The current position in the current work coordinates can be obtained through #5041-5045, X, Y, Z, A, and B, respectively. The values can NOT be read while the machine is in motion. #5043 (Z) represents the value after tool length compensation has been applied.

### Current Skip Signal Position

- #5061-#5065      The position where the last skip signal was triggered can be obtained through #5061-#5065, X, Y, Z, A, and B, respectively. Values are given in the current work coordinate system and can be used while the machine is in motion. #5063 (Z) represents the value after tool length compensation has been applied.

### Tool Length Compensation

- #5081-#5085      The current total tool length compensation that is being applied to the tool is returned. This includes tool length offset referenced by the current modal value set in H (#4008) plus the wear value.



## Offsets

All tool work offsets can be read and set within a macro expression. This allows the programmer to preset coordinates to approximate locations, or to set coordinates to values based upon the results of skip signal locations and calculations. When any of the offsets are read, the interpretation lookahead queue is stopped until that block is executed.

#5201-#5205	G52 X, Y, Z, A, B OFFSET VALUES
#5221-#5225	G54 " " " " "
#5241-#5245	G55 " " " " "
#5261-#5265	G56 " " " " "
#5281-#5285	G57 " " " " "
#5301-#5305	G58 " " " " "
#5321-#5325	G59 " " " " "
#7001-#7005	G110 X, Y, Z, A, B OFFSET VALUES
" "	" " " " "
#7381-#7385	G129 X, Y, Z, A, B OFFSET VALUES

### ADDRESS CONSTANT SUBSTITUTION

The usual method of setting control addresses A..Z is by appending a constant to the address. For instance,

**G01 X1.5 Y3.7 F20. ;**

sets addresses G, X, Y and F to 1, 1.5, 3.7 and 20.0 respectively and thus instructs the control to move linearly, G01, to position X=1.5 Y=3.7 at a feed rate of 20 inches per minute. Macro syntax allows the constants to be replaced with any variable or expression in any section of code (i.e., you do not have to be in a macro subroutine).

The previous statement can be replaced by the following code:

```
#1=1;
#2=.5;
#3=3.7;
#4=20;
G#1 X[#1+#2] Y#3 F#4 ;
```

The permissible syntax on addresses A..Z (exclude N or O) is as follows:

<address><-><variable>	A-#101
<address>[<expression>]	Y[#5041+3.5]
<address><->[<expression>]	Z-[SIN[#1]]

If the value of the variable does not agree with the range of the address, then the usual control alarm will result. For instance, the following code would result in a range error alarm because tool diameter numbers range from 0..50.

```
#1=75;
D#1;
```

When a variable or expression is used in place of an address constant, then the floating point value is rounded to the least significant digit. If #1=.123456, then G1X#1 would move the machine tool to .1235 on the X axis. If the control is in the metric mode, the machine would be moved to .123 on the X axis.

When an UNDEFINED variable is used to replace an address constant, then that address reference is ignored. For example, if #1 is undefined then the block

**G00 X1.0 Y#1 ;**

becomes

**G00 X1.0.**

No Y movement takes place.

## MACRO STATEMENTS

Macro statements are lines of code that allow the programmer to manipulate the control with features similar to any standard programming language. Included are functions, operators, conditional and arithmetic expressions, assignment statements, and control statements.

Functions and operators are used in expressions to modify variables or values. The operators are essential to expressions while functions make the programmer's job easier.

### Functions

Functions are built-in routines that the programmer has available to use. All functions have the form **<function\_name>[argument]**. Functions can be passed any expression as arguments. Functions return floating point decimal values. The function provided with the HAAS control are as follows:

FUNCTION	ARGUMENT	RETURNS	NOTES
SIN[ ]	Degrees	Decimal	Sine
COS[ ]	Degrees	Decimal	Cosine
TAN[ ]	Degrees	Decimal	Tangent
ATAN[ ]	Decimal	Degrees	Arctangent
			Same as FANUC ATAN[ ]/[1]
SQRT[ ]	Decimal	Decimal	Square root
ABS[ ]	Decimal	Decimal	Absolute value
ROUND[ ]	Decimal	Decimal	Round off a decimal
FIX[ ]	Decimal	Integer	Truncate fraction
ACOS[ ]	Degrees	Decimal	Arccosine
ASIN[ ]	Degrees	Decimal	Arcsine
#[ ]	Integer	Integer	Variable Indirection
DPRNT[ ]	ASCII text		External Output

### Notes on Functions

The function ROUND works differently depending on the context that it is used. When used in arithmetic expressions, the round function works as one would expect. That is, any number with a fractional part greater than or equal to .5 is rounded up to the next whole integer; otherwise, the fractional part is truncated from the number.

```
#1= 1.714 ;
#2= ROUND[#1] ; (#2 is set to 2.0)
#1= 3.1416 ;
#2= ROUND[#1] ; (#2 is set to 3.0)
```

When round is used in an address expression, then the argument of round is rounded to the addresses significant precision. For *metric* and *angle* dimensions, three-place precision is the default. For *inch*, four-place precision is the default. Integral addresses such as D, T and H are rounded normally.

```
#1= 1.00333 ;
G0 X[ #1 + #1 ] ;
        (Table moves to 2.0067) ;
G0 X[ ROUND[ #1 ] + ROUND[ #1 ] ] ;
        (Table moves to 2.0066) ;
G0 A[ #1 + #1 ] ;
        (Axis moves to 2.007) ;
G0 A[ ROUND[ #1 ] + ROUND[ #1 ] ] ;
        (Axis moves to 2.006) ;
D[1.67] (Diameter 2 is made current) ;
```



## Operators

Operators can be classified into three categories: Arithmetic operators, Logical operators and Boolean operators.

### Arithmetic Operators

Arithmetic operators consist of the usual unary and binary operators. They are:

+	- Unary plus	+1.23
-	- Unary minus	-[COS[30]]
+	- Binary addition	#1=#1+5
-	- Binary subtraction	#1=#1-1
*	- Multiplication	#1=#2*#3
/	- Division	#1=#2/4
MOD	- Remainder	#1=27 MOD 20 (#1 contains 7)

### Logical Operators

Logical operators are operators that work on binary bit values. Macro variables are floating point numbers. When logical operators are used on macro variables, only the integer portion of the floating point number is used. The logical operators are:

OR - logically OR two values together  
XOR - Exclusively OR two values together  
AND - Logically AND two values together

Examples:

```
#1=1.0; 0000 0001
#2=2.0; 0000 0010
#3=#1 OR #2  0000 0011      Here the variable #3 will contain 3.0 after the OR operation.
#1=5.0;
#2=3.0;
IF [#1 GT 3.0] AND [#2 LT 10]] GOTO1 Here control will transfer to block 1 because #1 GT 3.0 evaluates
                                         to 1.0 and #2 LT 10 evaluates to 1.0, thus 1.0 AND 1.0 is 1.0
                                         (TRUE) and the GOTO occurs.
```

As can be seen from the previous examples, CARE must be taken when using logical operators so that the desired result is achieved.

### Boolean Operators

Boolean operators always evaluate to 1.0 (TRUE) or 0.0 (FALSE). There are six Boolean operators. These operators are not restricted to conditional expressions, but they most often are used in conditional expressions. They are:

EQ - Equal to  
NE - Not Equal to  
GT - Greater Than  
LT - Less Than  
GE - Greater than or Equal to  
LE - Less Than or Equal to

The following are four examples of how Boolean and Logical operators can be used:

### Example

```
IF [#1 EQ 0.0] GOTO100;
WHILE [#101 LT 10] DO1;
#1=[1.0 LT 5.0];
IF [#1 AND #2 EQ #3] GOTO1
```

### Explanation

Jump to block 100 if value in variable #1 equals 0.0.  
 While variable #101 is less than 10 repeat loop DO1..END1.  
 Variable #1 is set to 1.0 (TRUE).  
 If variable #1 logically ANDed with variable #2 is equal to the value in #3 then control jumps to block 1.

## Expressions

Expressions are defined as any sequence of variables and operators surrounded by the square brackets "[" and "]". There are two uses for expressions: conditional expressions or arithmetic expressions. Conditional expressions return FALSE (0.0) or TRUE (any non zero) values. Arithmetic expressions use arithmetic operators along with functions to determine a value.

### Conditional Expressions

In the HAAS control, ALL expressions set a conditional value. The value is either 0.0 (FALSE) or the value is nonzero (TRUE). The context in which the expression is used determines if the expression is a conditional expression. Conditional expressions are used in the IF and WHILE statements and in the M99 command. Conditional expressions can make use of Boolean operators to help evaluate a TRUE or FALSE condition.

The M99 conditional construct is unique to the HAAS control. Without macros, M99 in the HAAS control has the ability to branch unconditionally to any line in the current subroutine by placing a P code on the same line. For example:

N50 M99 P10 ;

branches to line N10. It does not return control to the calling subroutine. With macros enabled, M99 can be used with a conditional expression to branch conditionally. To branch when variable #100 is less than 10 we could code the above line as follows.

N50 [#100 LT 10] M99 P10 ;

In this case, the branch occurs only when #100 is less than 10, otherwise processing continues with the next program line in sequence. In the above, the conditional M99 can be replaced with

N50 IF [#100 LT 10] GOTO10 ;

### Arithmetic Expressions

An arithmetic expression is any expression using constants, variables, operators, or functions. An arithmetic expression returns a value. Arithmetic expressions are usually used in assignment statements, but are not restricted to them.

Examples of Arithmetic expressions: #101=#145\*#30;

```
#1=#1+1;
X[#105+COS[#101]];
#[#2000+#13]=0;
```

### Assignment Statements

Assignment statements allow the programmer to modify variables. The format of the assignment statement is:

<expression>=<expression>

The expression on the left of the equal sign must always refer to a macro variable, whether directly or indirectly. The following macro initializes a sequence of variables to any value. Here both direct and indirect assignments are used.



```
O0300                                (Initialize an array of variables) ;
N1 IF [#2 NE #0] GOTO2                (B=base variable) ;
#3000=1                               (BASE VARIABLE NOT GIVEN) ;
N2 IF [#19 NE #0] GOTO3               (S=size of array);
#3000=2                               (SIZE OF ARRAY NOT GIVEN) ;
N3 WHILE [#19 GT 0] DO1 ;             (DECREMENT COUNT) ;
#19=#19-1                            (V=value to set array to) ;
#[#2+#19]=#22                         ;
END1 ;                                 ;
M99 ;
```

The above macro could be used to initialize three sets of variables as follows:

```
G65 P300 B101. S20 (INIT 101..120 TO #0) ;
G65 P300 B501. S5 V1 (INIT 501..505 TO 1.0) ;
G65 P300 B550. S5 V0 (INIT 550..554 TO 0.0) ;
```

The decimal point in B101., etc. would be required.

## Control Statements

Control statements allow the programmer to branch, both conditionally and unconditionally. They also provide the ability to iterate a section of code based on a condition.

### Unconditional Branch (GOTOnnn and M99 Pnnnn)

In the HAAS control, there are two methods of branching unconditionally. An unconditional branch will always branch to a specified block. M99 P15 will branch unconditionally to block number 15. The M99 can be used whether or not macros is installed and is the traditional method for branching unconditionally in the HAAS control. GOTO15 does the same as M99 P15. In the HAAS control, a GOTO command can be used on the same line as other G coding. The GOTO is executed after any other control commands as are traditional M codes.

### Computed Branch (GOTO#n and GOTO[expression])

Computed branching allows the program to transfer control to another block in the same subprogram. The block can be computed on the fly, as in the case of the GOTO[expression] form, or the block can be passed in through a local variable, as in the GOTO#n form.

The GOTO will round the variable or expression result that is associated with the Computed branch. For instance, if #1 contains 4.49 and GOTO#1 is executed, the control will attempt to transfer to a block containing N4. If #1 contains 4.5, then execution will transfer to a block containing N5.

The following code skeleton could be developed to make a program that serializes parts:

```

O9200                                (Engrave digit at current location.)
;
(D=DECIMAL DIGIT TO ENGRAVE);
IF [[#7 NE #0] AND [#7 GE 0] AND [#7 LE 9]] GOTO99;
#3000=1                               (INVALID DIGIT)
;
N99
#7=FIX[#7]                            (TRUNCATE ANY FRACTIONAL PART) ;
;
GOTO#7                                (NOW ENGRAVE THE DIGIT)  ;
;
N0                                     (DO DIGIT ZERO)
...
M99
;
N1                                     (DO DIGIT ONE)
;
M99
;
N2                                     (DO DIGIT TWO)
;
...
;
(etc,...)

```

With the above subroutine, you would engrave digit five with the following call:

G65 P9200 D5;

Computed GOTOs using expression could be used to branch processing based on the results of reading hardware inputs. An example might look like the following:

```

GOTO[[#1030*2]+#1031]    ;
NO (1030=0, 1031=0)      ;
...
M99;
N1 (1030=0, 1031=1)      ;
...
M99;
N2 (1030=1, 1031=0)      ;
...
M99;
N3 (1030=1, 1031=1)      ;
...
M99;

```

The discrete inputs always return either 0 or 1 when read. The GOTO[expression] will branch to the appropriate G-code based on the state of the two discrete inputs #1030 and #1031.

### **Conditional Branch (IF and GOTOn)**

Conditional branching allows the program to transfer control to another section of code within the same subroutine. Conditional branching can only be used when macros are enabled. The HAAS control allows two similar methods for accomplishing conditional branching.

IF [<conditional expression>] GOTOn



Here, as discussed above, <conditional expression> is any expression that uses the six Boolean operators EQ, NE, GT, LT, GE, or LE. The brackets surrounding the expression are mandatory. In the HAAS control, it is not necessary to include these operators. For example:

IF [#1 NE 0.0] GOTO5;

could also be:

IF [#1] GOTO5 ;

In this statement, if the variable #1 contains anything but 0.0, or the undefined value #0, then branching to block 5 will occur; otherwise, the next block will be executed. If portability to a control other than HAAS is desired, then it is recommended that the conditional operators be used.

In the HAAS control, a conditional expression can also be used with the M99 Pnnnn format, providing that macros have been enabled. An example is as follows:

G0 X0 Y0 [#1EQ#2] M99 P5;

Here, the conditional is for the M99 portion of the statement only. The machine tool is instructed to X0, Y0 whether or not the expression evaluates to TRUE or FALSE. Only the branch, M99, is executed based on the value of the expression. It is recommended that the IF GOTO version is used if portability is desired.

### Conditional Execution (IF THEN)

Execution of control statements can also be achieved by using the IF THEN construct. The format is:

IF [<conditional expression>] THEN <statement>;

---

**NOTE:** To preserve compatibility with FANUC syntax "THEN" may not be used with GOTOn.

This format is traditionally used for conditional assignment statements such as:

IF [#590 GT 100] THEN #590=0.0 ;

Here, variable #590 is set to zero when the value of #590 exceeds 100.0. In the HAAS control, if a conditional evaluates to FALSE (0.0), then the remainder of the IF block is ignored. This means that control statements can also be conditioned so that we could write something like:

IF [#1 NE #0] THEN G1 X#24 Y#26 F#9 ;

This executes a linear motion only if variable #1 has been assigned a value. You might try something like this:

IF [#1 GE 180] THEN #101=0 M99 ;

This says that if variable #1 (address A) is greater than or equal to 180, then set variable #101 to zero and return from the subroutine.

Here is an example of an IF statement that branches if a variable has been initialized to contain any value. Otherwise, processing will continue and an alarm will be generated. Remember, when an alarm is generated, program execution is halted.

N1 IF [#9NE#0] GOTO3 (TEST FOR VALUE IN F) ;  
N2 #3000=11(NO FEED RATE) ;  
N3 (CONTINUE) ;

## Iteration/Looping (WHILE DO END)

Essential to all programming languages is the ability to execute a sequence of statements a given number of times or to loop through a sequence of statements until a condition is met. Traditional G coding allows this with the use of the **L** address. A subroutine can be executed any number of times by using the **L** address.

M98 P2000 L5 ;

This is limited since you can not terminate execution of the subroutine on condition. Macros allows more flexibility with the WHILE-DO-END construct. The syntax is as follows:

```
WHILE [<conditional expression>] DOn ;
<statements> ;
ENDn ;
```

This executes the statements between DOn and ENDn as long as the conditional expression evaluates to TRUE. The brackets in the expression are necessary. If the expression evaluates to FALSE, then the block after ENDn is executed next. WHILE can be abbreviated to WH. The DOn-ENDn portion of the statement is a matched pair. The value of n is 1..3. This means that there can be no more than three nested loops per subroutine. A nest is basically a loop within a loop. A good example of how nesting of WHILE loops can be used is in defining a matrix.

```
#101= 3 ;
#102= 4 ;
G0 X#101 Y4. ;
F2.5 ;
WH [ #101 GT 0 ] DO1 ;
#102= 4 ;
WH [ #102 GT 0 ] DO2 ;
G81 X#101 Y#102 Z-0.5 ;
#102= #102 - 1 ;
END2 ;
#101= #101 - 1 ;
END1 ;
;
M30 ;
```

This program drills a 3 x 4 matrix hole pattern.

Although nesting of WHILE statements can only be nested to three levels, there really is no limit since each subroutine can have up to three levels of nesting. If there ever is a need to nest to a level greater than 3, then the segment containing the three lowest levels of nesting can be made into a subroutine thus overcoming the limitation.

If two separate WHILE loops are in a subroutine, they can use the same nesting index. For example:

```
#3001=0 (WAIT 500 MILLISECONDS) ;
WH [#3001 LT 500] DO1 ;
END1 ;
```

```
<other statements>
#3001=0 (WAIT 300 MILLISECONDS) ;
WH [#3001 LT 300] DO1 ;
END1 ;
```

This is valid code.

You can use GOTO to jump out of a region encompassed by a DO-END, but you can not use a GOTO to jump into it. Jumping around inside a DO-END region using a GOTO is allowed.



An infinite loop can be executed by eliminating the WHILE and expression. Thus,

```
DO1 ;  
<statements>  
END1 ;
```

Executes until the RESET key is pressed.

**CAUTION!** The following code can be confusing:

```
WH [ #1 ] D01 ;  
END1 ;
```

In the above, an alarm results indicating no “then” was found, here “then” refers to the D01. Change D01 (zero) to DO1 (letter O).

#### *COMMUNICATION WITH EXTERNAL DEVICES - DPRNT[ ]*

Macros allow additional capabilities to communicate with peripheral devices. One can do digitizing of parts, provide runtime inspection reports, or synchronize controls with user provided devices. The commands provided for this are POPEN, DPRNT[ ] and PCLOS.

#### **Communication preparatory commands**

POpen and PCLOS are not required on the HAAS mill. It has been included so that programs from different controls can be ported to the HAAS. On some controls POPEN is required prior to using a DPRNT statement. POPEN prepares the device on the serial port by sending it a DC2 code. PCLOS terminates communication with external devices by sending it a DC4 code.

#### **Formatted output**

The DPRNT statement allows the programmer to send formatted text to the serial port. Any text and any variable can be printed to the serial port. Variables can be formatted. The form of the DPRNT statement is as follows:

```
DPRNT[ <text> <#nnnn[wf]>... ] ;
```

DPRNT must be the only command in the block. In the above, <text> is any character from A to Z or the letters (+,-,/,\*, and the space). When an asterisk is output, it is converted to a space. The <#nnnn[wf]> is a variable followed by a format. The variable number can be any legal macro variable. The format [wf] is required and consists of two digits within square brackets. Remember that macro variables are real numbers with a whole part and a fractional part. The first digit in the format designates the total places reserved in the output for the whole part. The second digit designates the total places reserved for the fractional part. The total places reserved for output cannot be equal to zero or greater than eight. Thus the following formats are illegal:

```
[00] [54] [45] [36] /* not legal formats */
```

A decimal point is printed out between the whole part and the fractional part. The fractional part is rounded to the least significant place. When zero places are reserved for the fractional part, then no decimal point is printed out. Trailing zeros are printed as necessary if there is a fractional part. At least one place is reserved for the whole part, even when a zero is used there. If the value of the whole part has fewer digits than have been reserved, then leading spaces are output. If the value of the whole part has more digits than have been reserved, then the field is expanded so that these numbers are printed.

A carriage return is sent out after every DPRNT block.

## DPRNT[ ] Examples

<u>Code</u>	<u>Output</u>
N1 #1=1.5436 ;	X1.5436 Z 1.544 T 1
N2 DPRNT[X#1[44]*Z#1[03]*T#1[40]] ;	MEASURED INSIDE DIAMETER
N3 DPRNT[***MEASURED*INSIDE*DIAMETER***] ;	(no text, only a carriage return)
N4 DPRNT[] ;	
N5 #1=123.456789 ;	X-123.45679 ;
N6 DPRNT[X-#1[25]] ;	

### EXECUTION

DPRNT statements are executed at block interpretation time. This means that the programmer must be careful about where the DPRNT statements appear in the program, particularly if the intent is to print out positional information. Generally, a program is interpreted many blocks ahead in order to prevent the machine from pausing between movements.

G103 is useful for limiting lookahead. If you wanted to limit lookahead interpretation to one block, you would include the following command at the beginning of your program: (This actually results in a two block lookahead.)

G103 P1 ;

To cancel the lookahead limit, then issue a G103 P0 ;. G103 can not be used when cutter compensation is active.

### OPERATION NOTES

This section explains the additional screens and operator actions that come with macros.

Macro variables can be saved and restored to RS-232 or the optional floppy, much like parameters, settings, and offsets. Refer to the "Part Program Input / Output" section for RS-232 sending and receiving of macro variables, or the "Floppy Operation" section for sending and receiving them with this method.

### Variable Display Page

The macro variables are displayed and can be modified through the current commands display. The variable display is located after the operation timers display. To get to this page, press CURNT COMDS and use the page up/down key.

As the control interprets a program, the variable changes are displayed on the variable display page and results can be viewed.

Pages contain up to 32 variables and the display can be "paged" by pressing the left/right arrow keys.

Setting of a variable is accomplished by entering a value and then pressing the WRITE key. The variable that is highlighted on the screen is the variable that is affected.

Searching for a variable can be done by entering the variable number and pressing the up/down arrow. The page will change to the one that contains that variable and the entered variable will become the highlighted item.

The variables displayed represent the values of the variables at program interpretation time. At times, this may be up to 15 blocks ahead of the actual machine activity. Debugging of programs can be made easier by inserting a G103 at the beginning of a program to limit block buffering and then removing the G103 block after debugging is completed.



VS-Series

## Editing

For the most part, the editing of macro programs from the control is the same as before. There are a few peculiarities to be aware of.

Editing macro statements is more open than previously. For instance, it is possible to place a floating point constant within a standard G-code block, but it doesn't make much sense, and the control will raise an alarm at runtime. For all instances of improperly structured or improperly placed macro statements, the control will raise an appropriate alarm. Most of these alarms have been put off until runtime so that operator editing can be more flexible. Be careful when editing expressions. Brackets must be balanced and you will not receive an alarm until runtime.

The DPRNT[ ] function can be edited much like a comment. You can delete it or move it as a whole item, or you can edit individual items within the brackets. Variable references and format expressions must be altered as a whole entity. If you wanted to change [24] to [44], place the cursor so that [24] is highlighted, enter [44] and press the write key. Remember, you can use the crank handle to maneuver through long

DPRNT[ ] expressions.

Addresses with expressions can be somewhat confusing. In this case, the alphabetic address stands alone. For instance, the following block contains an address expression in X:

G1 G90 X [ COS[ 90 ] ] Y3.0 (CORRECT) ;

Here, the **X** and brackets stand alone and are individually editable items. It is possible, through editing, to delete the entire expression and replace it with a floating point constant.

G1 G90 X 0 Y3.0 (!!! WRONG !!!) ;

The above block will result in an alarm at runtime. The correct form looks as follows:

G1 G90 X0 Y3.0 (CORRECT) ;

Note that the zero is attached to **X**. REMEMBER when you see an alpha character standing alone it is an address expression.

*FANUC-STYLE MACRO FEATURES NOT INCLUDED IN HAAS CNC CONTROL*

This section lists the FANUC macro features that have not yet been implemented.

M ALIASING	REPLACE G65 Pnnnn WITH Mnn PROGS 9020-9029.
G66	MODAL CALL IN EVERY MOTION BLOCK
G66.1	MODAL CALL IN EVERY BLOCK
G67	MODAL CANCEL
M98	ALIASING, T CODE PROG 9000, VAR #149, ENABLE BIT
M98	ALIASING, S CODE PROG 9029, VAR #147, ENABLE BIT
M98	ALIASING, B CODE PROG 9028, VAR #146, ENABLE BIT
SKIP/N	N=1..9
#3007	MIRROR IMAGE ON FLAG EACH AXIS
#4201-#4320	CURRENT BLOCK MODAL DATA
#5101-#5106	CURRENT SERVO DEVIATION
NAMES FOR VARIABLES	FOR DISPLAY PURPOSES
ATAN [ ]/[ ]	ARCTANGENT, FANUC VERSION
BIN [ ]	CONVERSION FROM BCD TO BIN
BCD [ ]	CONVERSION FROM BIN TO BCD
FUP [ ]	TRUNCATE FRACTION CEILING
LN [ ]	NATURAL LOGARITHM
EXP [ ]	BASE E EXPONENTIATION
ADP [ ]	RE-SCALE VAR TO WHOLE NUMBER
BPRNT [ ]	

The following can be used as alternative methods for achieving the same results for a few unimplemented FANUC macro features.

GOTO-nnnn

Searching for a block to jump in the negative direction, i.e. backwards through a program, is not necessary if you use unique N address codes.

A block search is made starting from the current block being interpreted. When the end of the program is reached, searching continues from the top of the program until the current block is encountered.



VS-Series

## 4. G CODES (PREPARATORY FUNCTIONS)

The following is a summary of the **G** codes. A “\*” indicates the default within each group, if there is one.

<b>Code:</b>	<b>Group:</b>	<b>Function:</b>	<b>Description On Page:</b>
G00	01*	Rapid Motion	205
G01	01	Linear Interpolation Motion	205
G02	01	CW Interpolation Motion	206
G03	01	CCW Interpolation Motion	207
G04	00	Dwell	208
G09	00	Exact Stop	208
G10	00	Set Offsets	208
G12	00	CW Circular Pocket Milling	209
G13	00	CCW Circular Pocket Milling	209
G17	02*	XY Plane Selection	212
G18	02	ZX Plane Selection	212
G19	02	YZ Plane Selection	212
G20	06*	Select Inches	213
G21	06	Select Metric	213
G28	00	Return To Reference Point	213
G29	00	Return From Reference Point	213
G31	00	Feed Until Skip (optional)	214
G35	00	Automatic Tool Diameter Measurement (optional)	214
G36	00	Automatic Work Offset Measurement (optional)	215
G37	00	Automatic Tool Offset Measurement (optional)	214
G40	07*	Cutter Comp Cancel	215
G41	07	2D Cutter Compensation Left	215
G42	07	2D Cutter Compensation Right	215
G43	08	Tool Length Compensation +	216
G44	08	Tool Length Compensation -	216
G47	00	Text Engraving	216
G49	08*	G43/G44/G143 Cancel	216
G50	11	G51 Cancel	221
G51	11	Scaling (optional)	221
G52	12	Set Work Coordinate System G52 (Yasnac)	225
G52	00	Set Local Coordinate System G52 (Fanuc)	226
G52	00	Set Local Coordinate System (HAAS)	226
G53	00	Non-Modal Machine Coordinate Selection	226
G54-59	12*	Select Work Coordinate System 1-6	226
G60	00	Unidirectional Positioning	226
G61	13	Exact Stop Modal	226
G64	13*	G61 Cancel	226
G65	00	Macro Subroutine Call (optional)	179
G68	16	Rotation (optional)	223
G69	16	G68 Cancel (optional)	225
G70	00	Bolt Hole Circle	227
G71	00	Bolt Hole Arc	227
G72	00	Bolt Holes Along an Angle	227
G73	09	High Speed Peck Drill Canned Cycle	230
G74	09	Reverse Tap Canned Cycle	231
G76	09	Fine Boring Canned Cycle	232
G77	09	Back Bore Canned Cycle	232



G80	09*	Canned Cycle Cancel	233
G81	09	Drill Canned Cycle	233
G82	09	Spot Drill Canned Cycle	235
G83	09	Normal Peck Drill Canned Cycle	226
G84	09	Tapping Canned Cycle	237
G85	09	Boring Canned Cycle	239
G86	09	Bore/Stop Canned Cycle	239
G87	09	Bore/Stop/Manual Retract Canned Cycle	240
G88	09	Bore/Dwell/Manual Retract Canned Cycle	240
G89	09	Bore/ Dwell Canned Cycle	241
G90	03*	Absolute	241
G91	03	Incremental	241
G92	00	Set Work Coordinates - (FANUC or HAAS)	241
G92	00	Set Work Coordinates - (Yasnac)	242
G93	05	Inverse Time Feed Mode	242
G94	05*	Feed Per Minute Mode	242
G95	05	Feed Per Revolution	243
G98	10*	Initial Point Return	243
G99	10	R Plane Return	243
G100	00	Cancel Mirror Image	243
G101	00	Enable Mirror Image	243
G102	00	Programmable Output To RS-232	247
G103	00	Limit Block Buffering	247
G107	00	Cylindrical Mapping	248
G110-129	12	Select Work Coordinate System 7-26	251
G136	00	Automatic Work Offset Center Measurement	215
G141	07	3D+ Cutter Compensation	252
G143	08	5-Axis Tool Length Compensation	252
G150	00	General Purpose Pocket Milling	254
G153	09	5-Axis High Speed Peck Drilling Canned Cycle	259
G154	12	Select Work Coordinates P1-P99	x51
G155	09	5-Axis Reverse Tap Canned Cycle	260
G161	09	5-Axis Drill Canned Cycle	261
G162	09	5-Axis Spot Drill Canned Cycle	262
G163	09	5-Axis Normal Peck Drilling Canned Cycle	262
G164	09	5-Axis Tapping Canned Cycle	264
G165	09	5-Axis Boring Canned Cycle	265
G166	09	5-Axis Bore/Stop Canned Cycle	266
G169	09	5-Axis Bore/Dwell Canned Cycle	267
G174	00	CCW Non-Vertical Rigid Tap	267
G184	00	CW Non-Vertical Rigid Tap	267
G187	00	Accuracy Control for High Speed Machining	268

Each G code defined in this control is part of a group of G codes. The Group 00 codes are non-modal; that is, they specify a function applicable only to the block they are in and do not affect other blocks. The other groups are modal and apply to all subsequent blocks until the end of the program or until replaced by a different G code from the same group. A modal G code will remain in effect until another G code from the same group is specified.

There is also one case where the Group 01 G codes will cancel the Group 09 (canned cycles) codes. If a canned cycle (G73 through G89) is active, the use of G00 or G01 will cancel the canned cycle.

The control will now store up to 500 G-code programs in memory.

## RAPID POSITION COMMANDS (G00)

<b>G00</b>	<b>Rapid Motion Positioning</b>
X	Optional X-axis motion command
Y	Optional Y-axis motion command
Z	Optional Z-axis motion command
A	Optional A-axis motion command

## Group 01

This G code is used to cause rapid traverse of the three or four axes of the machine. The auxiliary axes **B**, **C**, **U**, **V**, and **W** can also be moved with G00. This G code is modal, so a block with G00 causes all following blocks to be rapid motion until another Group 01 code is specified. The rapid traverse rate is dependent on the maximum speed possible for each axis independently, as modified by the RAPID OVERRIDE keys.

Generally, rapid motion will not be in a straight line. Each axis specified is moved at the same speed, but all axes will not necessarily complete their motions at the same time. The control will wait until all motions are complete before implementing the next command. Only the axes specified are moved, and the incremental or absolute position commands (G90 or G91) will change how those axis motion values are interpreted. Setting 57 can change how closely the machine waits for a precise stop before and after a rapid move.

## INTERPOLATION COMMANDS (G01, G02, G03)

<b>G01</b>	<b>Linear Interpolation Motion</b>
F	Feedrate 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
,R	Radius of the circle
,C	Distance from center of intersection where the chamfer begins

## Group 01

This G code provides for straight line (linear) motion from point to point. Motion can occur in 1, 2 or 3 axes. All axes will start and finish motion at the same time. The rotary axis may also be commanded with G01, providing helical motion. The speed of all axes is controlled so that the feedrate specified is achieved along the actual path. Rotary axis feedrate (helical motion) is dependent on the rotary axis diameter setting (Setting 34). The F command is modal and may be specified in a previous block. Only the axes specified are moved and the incremental or absolute position commands (G90 or G91) will change how those values are interpreted. The auxiliary axes **B**, **C**, **U**, **V**, and **W** can also be moved with G01, but only one axis is moved at a time.

### Corner Rounding and Chamfering Example

```

F1.0;
G17;
G00 X-10. Y-6. Z-8.;          (Rapid to start point)
G01 X-10.0 Y-11,C1.;         (Chamfer)
G01 X-5. Y-11.,R1.;          (Rounding)
X-5. Y-6.;                   (Terminating stroke)
M30;

```



A chamfer block or a corner-rounding block can be automatically inserted between two linear interpolation blocks by specifying ,C (chamfering) or ,R (corner rounding). There must be a terminating linear interpolation block following the beginning block (a G04 pause may intervene). These two linear interpolation blocks specify a corner of intersection. If the beginning block specifies a C, the value following the C is the distance from the intersection to where the chamfer begins, and also the distance from the intersection to where the chamfer ends. If the beginning block specifies an R, the value following the R is the radius of a circle tangent to the corner at two points: the beginning of the corner-rounding arc and the endpoint of that arc. There can be consecutive blocks with chamfering or corner-rounding specified. There must be movement on the two axes specified by the selected plane, whether the active plane is XY (G17), XZ (G18) or YZ (G19).

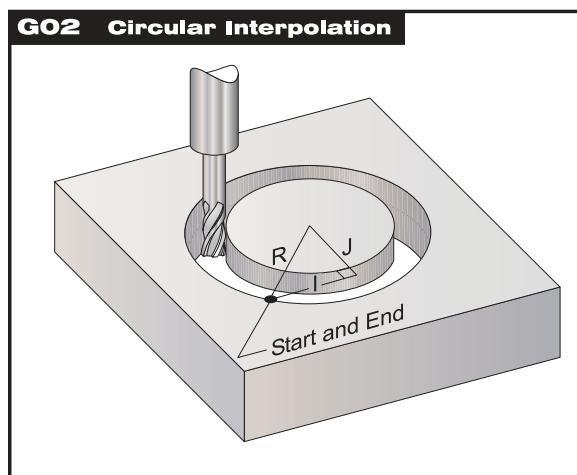
**G02 CW Circular Interpolation Motion****Group 01**

- F Feedrate 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

This G code is used to specify clockwise circular motion of two linear axes. Circular motion is possible in any two of the X, Y, and Z axes as selected by G17, G18, and G19. The X, Y, and Z commands specify the end point of the motion, which may be either absolute (G90) or incremental (G91) motion. If the X, Y, or Z location for the selected plane is not specified, the endpoint of the arc is the same as the starting point for that axis. There are two ways to specify the center of the circular motion; the first uses I, J, or K to specify the distance from the starting point to the center of the arc, and the second uses R to specify the radius of the arc (maximum 7740. inches). The following describes these further.

**I, J, K:** When I, J, or K is used to specify the center of the arc, R may not be used. Only the I, J, or K specific to the selected plane (I,J for G17, I,K for G18, J,K for G19) is allowed. If only one of the I, J, K values is specified, the others are assumed to be zero. The I, J, or K value is the signed distance from the starting point to the center of the circle. Small errors in these values are tolerated, up to 0.0010 inch. Use of 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 value is needed.

To cut a complete circle of 360 degrees ( $360^\circ$ ), 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. The following line will cut a complete circle:

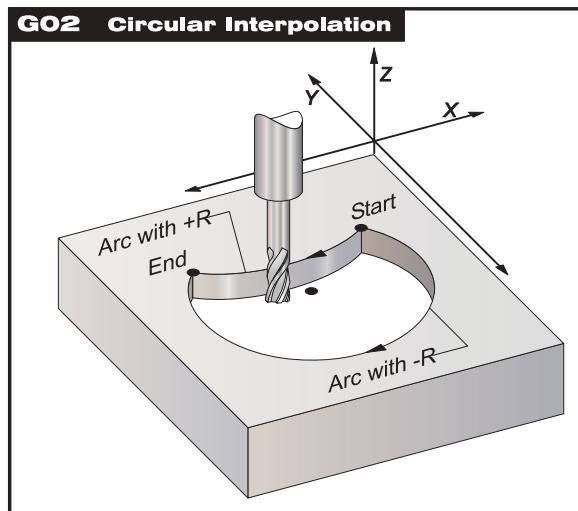


G02 I3.0 J4.0 (Assumes G17; XY plane)

In cases where you are cutting less than a complete circle, it is much easier to use **R** instead of **I**, **J**, **K**.

**R:** When **R** is used to specify the center of the circle, a complete 360-degree arc is not possible. An **X**, **Y**, or **Z** value is required to specify an endpoint different from the starting point. **R** is the distance from the starting point to the center of the circle. With a positive **R**, the control will generate a path of 180 degrees or less; to generate an angle of more than 180 degrees, specify a negative **R**. Small errors in this value are tolerated, up to 0.0010 inch.

The following lines will cut an arc less than 180 degrees ( $180^\circ$ ):



G01 X3.0 Y4.0

G02 X-3.0 R5.0

and the following lines will cut an arc of more than 180 degrees ( $180^\circ$ ):

G01 X3.0 Y4.0

G02 X-3.0 R-5.0

### G03 CCW Circular Interpolation Motion

### Group 01

G03 will generate counterclockwise circular motion but is otherwise the same as G02.

### HELICAL

Helical motion is possible with G02 or G03 by programming the linear axis that 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 feedrate.

**MISCELLANEOUS G CODES (G04, G09)****G04 Dwell**

P The dwell time in seconds or milliseconds

**Group 00**

G04 is used to cause a delay or dwell in the program. The block containing G04 will delay for the time specified by the P code. If P has no fractional part, the delay is in milliseconds (0.001 second); otherwise the delay is in seconds.

**G09 Exact Stop****Group 00**

The G09 code is used to specify an exact stop. It is non-modal and does not affect the following blocks. Rapid and interpolated moves will decelerate to an exact stop before another block is processed. In exact stop, moves will take longer and continuous cutter motion will not occur. This may cause deeper cutting where the tool stops.

**PROGRAMMABLE OFFSET SETTING (G10)****G10 Set Offsets****Group 00**

- L Selection of length, length wear, diameter, diameter wear, or work coordinates.
- P Selection of offset number.
- R Offset value or increment for length and diameter.
- X Optional X-axis zero location.
- Y Optional Y-axis zero location.
- Z Optional Z-axis zero location.
- A Optional A-axis zero location.

G10 can be used to change the tool length and work offsets from inside a program. The following codes are used for selection of offsets:

- L2 Work coordinate origin for G52 and G54–G59
- L10 Length offset amount (for H code)
- L1 or L11 Tool wear offset amount (for H code)
- L12 Diameter offset amount (for D code)
- L13 Diameter wear offset amount (for D code)
- L20 Auxiliary work coordinate origin for G110–G129

The P code is used to index the appropriate offsets.

P1–P100	Used to reference D or H code offsets	L10–L13
P0	G52 references work coordinate	L2
P1–P6	G54–G59 references work coordinates	L2
P1–P20	G110–G129 references auxiliary coordinates	L20

The R, X, Y, Z, and A codes are signed, fractional numbers in inches (or mm). The R, X, Y, Z, and A values are absolute or incremental, depending on the current G90/G91 mode.

**G10 Examples:**

G10	L2	P1	G91	X6.0	{Move coordinate G54 6.0 to the right};
G10	L20	P2	G90	X10. Y8.	{Set work coordinate G111 to X10.0 ,Y8.0};
G10	L10	G90	P5	R2.5	{Set offset for Tool #5 to 2.5};
G10	L12	G90	P5	R.375	{Set diameter for Tool #5 to 3/8};

### CIRCULAR POCKET MILLING (G12, G13)

There are two G codes that will provide for pocket milling of a circular shape. They are different only in which direction of rotation is used. Both are functional only in the default XY circular plane (G17). This G code implies the use of G42.

#### G12 Circular Pocket Milling Clockwise

**Group 00**

- \*D Tool radius or diameter selection
- \*\*I Radius of first circle (or finish if no K)
- K Radius of finished circle (if specified)
- L Loop count for repeating deeper cuts
- Q Radius increment (must be used with K)
- F Feedrate in inches (mm) per minute
- Z depth of cut or increment

\*In order to get the programmed circle diameter, the control uses the selected D code tool size. If you want to program tool centerline select D0.

\*\* The I value must be greater than the Tool Radius, but less than the K value.

The tool must be positioned at the center of the circle, either in a previous block or in this block using X and Y. The cut is performed entirely with circular motions of varying radius. To remove all the material within the circle, use I and Q values less than the tool diameter and a K value equal to the circle radius. To cut a circle radius only, use an I value set to the radius and no K or Q value. G12 belongs to Group 00 and thus is non-modal. If G91 (incremental) is specified and an L count is included, the Z increment is repeated L times at feedrate F.

#### G13 Circular Pocket Milling Counterclockwise

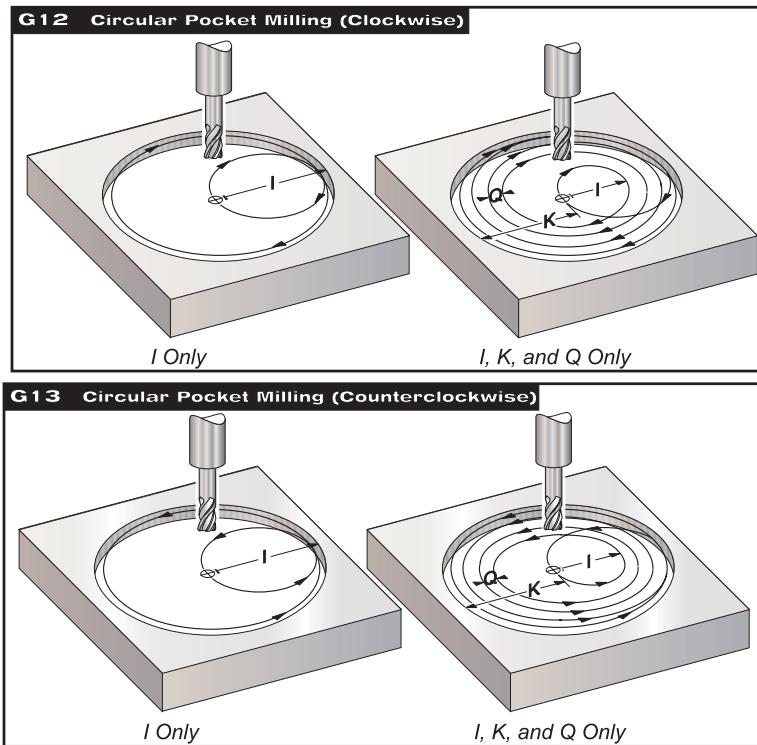
**Group 00**

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

%	
O00098(SAMPLE G12 AND G13)	
(OFFSET D01 SET TO APPROX. TOOL SIZE)	
(TOOL MUST BE MORE THAN Q IN DIAM.)	
T1M06	
G54G00G90X0Y0	(Move to center of G54)
G43Z0.1H01	
S2000M03	
G12I1.5F10.Z-1.2D01	(Finish pocket clockwise)
G00Z0.1	
G55X0Y0	(Move to center of G55)
G12I0.3K1.5Q0.3F10.Z-1.2D01	(Rough and finish clockwise)
G00Z0.1	
G56X0Y0	(Move to center of G56)
G13I1.5F10.Z-1.2D01	(Finish pocket counterclockwise)
G00Z0.1	
G57X0Y0	(Move to center of G57)
G13I0.3K1.5Q0.3F10.Z-1.2D01	(Rough and finish counterclockwise)
G00Z0.1	
G28	
M30	
%	



VS-Series



### CIRCULAR POCKET MILLING

The control software includes a Yasnac-style circular pocket milling program (G12, Clockwise Circular Pocket; G13, Counterclockwise Pocket). These G codes assume the use of cutter compensation, so a G41 or G42 is not required in the program line. However, a D offset number, for cutter radius or diameter, is required 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 various applications:

**SINGLE PASS:** Use **I** only.

**APPLICATIONS:** One-pass counterboring; rough and finish pocketing of smaller holes; ID keyway cutting of O-ring grooves.

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

**APPLICATIONS:** Multiple-pass counterboring; 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** may also be used).

**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**.

The previous figures show the tool path during the G12 and G13 cycles. One uses **I** only and the other uses **I, K, and Q**.

**EXAMPLE** G13 single-pass using I only:

**PROGRAM EXAMPLE**

```
%  
O2000  
T1 M06  
G00 G90 G54 X0 Y0 S4000 M03  
G43 H01 Z.1 M08  
G13 Z-1.0 F20. D01 I.500  
G00 Z1.0 M09  
G28 G91 Y0 Z0  
M30  
%
```

**DESCRIPTION**

(0.500 entered in the Radius/Diameter offset column)  
(Tool #1 is a 0.500" diameter endmill)

(Will complete a 1-inch diameter hole 1 inch deep)

**EXAMPLE** G13 multiple-pass using I, K, and Q:

This example will complete a 3-inch diameter hole, 1 inch deep, with a cutter overlap of 0.100 inch.

**PROGRAM EXAMPLE**

```
%  
O3000  
T1 M06  
G00 G90 G54 X0 Y0 S4000 M03  
G43 H01 Z.1 M08  
G13 Z-1.0 I.400 K1.5 Q.400 D01 F20.  
G00 Z1.0 M09  
G28 G91 Y0 Z0  
M30  
%
```

**DESCRIPTION**

(0.500 entered in the Radius/Diameter offset column)  
(Tool #1 is a 0.500" diameter endmill)

**EXAMPLE** G13 multiple-pass using I, K, Q, L, and G91:

This program uses G91 and an L count of 4, so this cycle will execute a total of four times. The Z depth increment is 0.500. This is multiplied by the L count, making the total depth of this hole 2.000.

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

---

**NOTE:** If the geometry column of the control Offsets display has a value inserted, the circular pocket milling cycle will automatically read the data, regardless of whether D0 is present or not. The only effective way to cancel cutter compensation for pocket milling is to insert a D00 in the program line. This will bypass the value in the Offsets geometry column.

**PROGRAM EXAMPLE**

```
%  
O4000  
T1 M06  
G00 G90 G54 X0 Y0 S4000 M03  
G43 H01 Z.1 M08  
G01 Z0 F10.  
G13 G91 Z-.5 I.400 K2.0 Q.400 L4 D01 F20.  
G00 G90 Z1.0 M09  
G28 G91 Y0 Z0  
M30  
%
```

**DESCRIPTION**

(0.500 entered in the Radius/Diameter offset column)  
(Tool #1 is a 0.500" diameter endmill)

**CIRCULAR PLANE SELECTION (G17, G18, G19)**

The plane used for circular motions must consist of two of the three axes **X**, **Y**, and **Z**. One of three G codes used to select the plane: G17 for XY, G18 for XZ, and G19 for YZ. Each are modal and apply to all subsequent circular motions until another Group 02 code is found.

**G17 XY Plane Selection****Group 02**

The G17 code is used to select the XY plane for circular motion. It is modal and applies to all circular motions until another Group 02 command is programmed. In this plane, circular motion is defined as clockwise for the operator looking down onto the XY table from above. This defines the motion of the tool relative to the table.

**G18 XZ Plane Selection****Group 02**

The G18 code is used to select the XZ plane for circular motion. It is modal and applies to all circular motions until another Group 02 command is programmed. In the XZ plane, circular motion is defined as clockwise for the operator looking from the rear of the machine toward the front control panel.

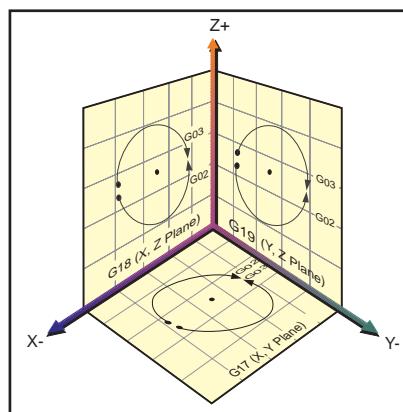
**G19 YZ Plane Selection****Group 02**

The G19 code is used to select the YZ plane for circular motion. It is modal and applies to all circular motions until another Group 02 command is programmed. In the YZ plane, circular motion is defined as clockwise for the operator looking across the table from the side of the machine where the control panel is mounted.

The default plane selection when the machine is powered on is G17, the XY plane. This means that a circular motion in the plane of the XY table may be programmed without first selecting G17. Plane selection also applies to G12 and G13, circular pocket milling, which must always be in the XY plane.

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 feedrate.

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



**INCH / METRIC SELECTION (G20, G21)**

**G20      Select Inches**  
**G21      Select Metric**

**Group 06**  
**Group 06**

The standard G codes G20 and G21 are sometimes used to select between inch and metric, BUT, in the Haas control, the G20 (inch) and G21 (mm) codes can only be used to ensure that the inch/metric setting is set correctly for the program in use. Selection between inch and metric programming can only be done from Setting 9.

**REFERENCE POINT DEFINITION AND RETURN**

**G28 Return to Machine Zero Thru Optional G29 Reference Point                          Group 00**

The G28 code is used to return all axes to machine zero. If an **X**, **Y**, **Z**, or **A** axis is specified in the same block, only those axes will move, through the location specified to the machine's zero point. Movement to machine zero will be through the specified point after applying the relevant work and tool offsets. This point is called the G29 reference point and is saved for use in G29. If no **X**, **Y**, **Z**, or **A** is specified, all axes will move directly to machine zero. Any auxiliary axes (**B**, **C**, etc.) are returned to home after the **X**, **Y**, **Z**, and **A** axes. After using the current offsets in moving to machine zero, G28 cancels tool length offsets for subsequent blocks.

**Example 1**

Work Offset G54: Z = -2.0  
 Tool 2 Length: -12.0

Program segment:

```
G90 G54;  

G43 H02;  

G28 Z0.;  

G00 Z1.
```

The G28 block will move to machine coordinate Z = -14.0 before moving to Z = 0. The following block (G00 Z1.) will move to machine coordinate Z = -1.

**Example 2**

(same work and tool offsets as Example 1)

Program segment:

```
G91 G54;  

G43 H02;  

G28 Z0
```

The G28 block will move directly to machine coordinate Z = 0 since incremental positioning is in effect.

**G29      Return From REFERENCE point**

**Group 00**

The G29 code is used to move the axes to a commanded **X**, **Y**, **Z** or **A** position. The axes selected in this block are moved to the G29 reference point saved or recorded in G28, and then moved to the **X**, **Y**, **Z** or **A** location specified in the G29 command. The positions are interpreted in the current coordinate system.

**Skip Function (G31)**

<b>G31</b>	<b>Feed Until Skip</b> (This G-code is optional and requires a probe)	<b>Group 00</b>
F	Feedrate 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	
B	Optional B-axis motion command	

The skip function is a non-modal operation that causes a linear move to the specified **X**, **Y**, **Z**, and/or **A** position. It applies only to the block in which G31 is specified. A feedrate must be defined previously or in this block. The specified move is started and continues until the end point or the skip signal. The skip signal is a discrete input that usually indicates that the end of travel has been reached; this is usually a probe. The control will beep when the end of travel is reached.

Cutter compensation may not be active during a skip function. M78 or M79 may be used to test if the skip signal was received. An M75 can be used to mark the probed point as the reference point for G35 or G136.

**Automatic Tool Measurement (G35, G37)**

<b>G35 Automatic Tool Diameter Measurement</b> (This G-code is optional and requires a probe)	<b>Group 00</b>
<b>G37 Automatic Tool Offset Measurement</b> (This G-code is optional and requires a probe)	<b>Group 00</b>
F Feedrate in inches (mm) per minute	
D Tool diameter offset number (G35)	
H Tool offset number (G37)	
Z Required Z-axis offset	

Automatic tool offset measurement (G37) is a non-modal operation that causes linear motion of the Z axis until the skip signal is received or the **Z** travel limit is reached. A non-zero **H** code must be active, G43 or G44 must be active, a **Z** value must be specified, and a feedrate must be defined. No **X**, **Y**, or **A** code is allowed. When the move is terminated, the specified **Z** and the final **Z** positions are used to set the specified tool offset (**Hnnn**). The active coordinate system is taken into account.

The coordinate system (G54–G59, G110–G129) and tool length offsets (H01–H200) may be selected in this block or in a previous block. The end point of the **Z** move is controlled only by the machine's travel limits.

The resulting tool offset value is such that a subsequent move to the **Z** value specified in the G37 command will move the tool to the position where the skip signal was sensed. The skip signal is a discrete input that usually indicates that the end of travel has been reached; this is sometimes a probe. Cutter compensation may not be active during a skip function. M78 or M79 may be used to test if the skip signal was received. The resulting tool offset is the offset between the work zero point and the point where the probe is touched.

The automatic tool diameter measurement function (G35) is used to set the tool diameter (or radius) using two different probe passes, one on each side of the tool. The first point is set with a G31 block using an M75, and the second point is set with the G35 block. The distance between these two points is set into the active **Dnnn** value. A nonzero **D** code must be selected. Setting 63 is used to reduce this measurement by the width of the tool probe.



VS-Series

## AUTOMATIC WORK OFFSET MEASUREMENT (G36, G136)

### G36 Automatic Work Offset Center Measurement (optional G-code, requires a probe) G136 Automatic Work Offset Center Measurement

**Group 00**  
**Group 00**

F	Feedrate in inches (mm) per minute
I	Optional offset distance along X axis
J	Optional offset distance along Y axis
K	Optional offset distance along Z axis
X	Optional X-axis motion command
Y	Optional Y-axis motion command
Z	Optional Z-axis motion command
A	Optional A-axis motion command

Automatic work offset measurement is a non-modal operation that causes linear motion of the **X**, **Y**, **Z**, and **A** axes until the skip signal is received or the end of the programmed motion is reached. The **X**, **Y**, **Z**, and **A** axes are moved to the programmed position in a linear move but will stop early if the skip signal is received. Tool offsets must not be active when this function is performed. M78 or M79 may be used to test if the skip signal was received. The currently active work coordinate system is set for each axis programmed. The point where the skip signal is received becomes the work zero position. The work coordinate system may be selected in this block or in a previous block.

The points probed are offset by the values in Settings 59 through 62.

A G36 will set the work coordinates to the point where the probe hits. G136 will set the work coordinates to a point at the center of a line between the probed point and the point set with M75. This allows the center of a part to be found using two separate probed points.

Note that the **X**, **Y**, **Z**, and/or **A** commands programmed into this block are interpreted in the coordinate system that is about to be set. Thus, the end point of the move will be interpreted within the old work coordinate value. For this reason, it is easier to program these moves as incremental (G91).

If an **I**, **J**, or **K** is specified, the appropriate axis work offset is shifted by the amount in the **I**, **J**, or **K** command. This allows the work offset to be shifted some distance away from where the probe actually hits.

## CUTTER COMPENSATION (G40, G41, G42)

### G40 Cutter Comp Cancel

**Group 07**

G40 will cancel G41 or G42 cutter compensation. Programming a D00 will also cancel cutter compensation.

### G41 2D Cutter Compensation Left

**Group 07**

G41 will select cutter compensation left; that is, the tool is moved to the left of the programmed path to compensate for the size of the tool. A **Dnnn** must also be programmed to select the correct tool size from compensation memory. If compensation memory contains a negative value for cutter size, cutter compensation will operate as though G42 was specified.

### G42 2D Cutter Compensation Right

**Group 07**

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 **Dnnn** must also be programmed to select the correct tool size from compensation memory. If compensation memory contains a negative value for cutter size, cutter compensation will operate as though G41 was specified.

Refer to the "Cutter Compensation" section for more information.

**TOOL LENGTH COMPENSATION (G43, G44, G49)****G43 Tool Length Compensation + (plus)****Group 08**

This code selects tool length compensation in a positive direction. That is, the tool length offsets are added to the commanded axis positions. A nonzero **Hnnn** must be programmed to select the correct entry from offset memory. The automatically entered offsets using the TOOL OFSET MESUR key assume that G43 is being used.

**G44 Tool Length Compensation – (minus)****Group 08**

This code selects tool length compensation in a negative direction. That is, the tool length offsets are subtracted from the commanded axis positions. A nonzero **Hnnn** must be programmed to select the correct entry from offsets memory.

**G49 G43/G44/G143 Cancel****Group 08**

This **G** code cancels tool length compensation. Putting in an H0 will also cancel tool length compensation. G28, M30, and RESET will also cancel tool length compensation.

**ENGRAVING (G47)****G47 Text Engraving****Group 00**

In order to use G47, the part program must be in G90 (absolute) mode, and setting 29 (G91 non-modal) must be OFF.

- E Plunge rate (units/min)
- F Engraving feedrate (units/min)
- I Angle of rotation (-360. to +360.); default is 0
- J Scaling factor in inches (minimum = 0.001 inch); default is 1.0 inch
- P 0 for literal string engraving  
1 for sequential serial number engraving  
32–126 for ASCII characters
- 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 the G47 command, 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 incrementally on a series of parts, with the number being increased 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 either be programmed or set manually. For example, if it is programmed:

G47 P1 (1234)

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/ENTER 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,  
! " # \$ % & ' ( ) \* + , - . / : ; < = > ? [ \ ] ^ \_ { }

Not all of these characters can be entered at the control. 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

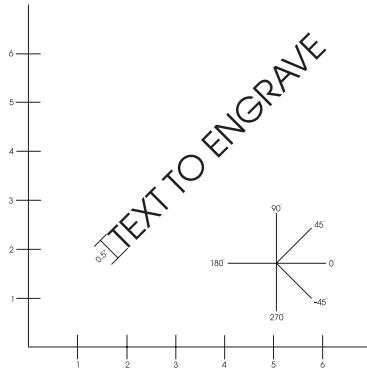
ENGRAVE THIS

Note that P0 is used, instead of P1, for literal string engraving.

### EXAMPLE

This example will create the figure shown.

G47 P0 X2.0 Y2.0 I45. J5 R.05 Z-.005 F15.0 E10.0 (TEXT TO ENGRAVE)





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 0.05 inch (mm) above the cutting plane after engraving
Z-.005	selects a 0.005 inch (mm) deep cut
F15.0	selects an engraving feedrate of 15 units/min
E10.0	commands the cutter to plunge at a rate of 10 units/min

The milling strokes defining each character, i.e., the font, are compiled G-code in the HAAS control. The font characters may be redefined by supplying a different G-code program, giving it the name O9876. This program must conform to the format expected by the HAAS control. For guidance, some of the code from the built-in font program is shown below. The example below may be used as a template. Code should be written exactly as shown.

P values to engrave specific characters:

32	blank	59	;
33	!	60	<
34	"	61	=
35	#	62	>
36	\$	63	?
37	%	64	@
38	&	65-90	A-Z
39	'	91	[
40	(	92	\
41	)	93	]
42	*	94	^
43	+	95	-
44	,	96	,
45	-	97-122	a-z
46	.	123	{
47	/	124	
48-57	0-9	125	}
58	:	126	~

## Example O9876 G-Code Program

	<b>Comments</b>
%	% marks the start of the program.
O9876 (Engraving)	The program number must be 9876.
#700= #4003	Save G90/G91
#701= #4001	Save G00/G01 etc.
G00 X#24 Y#25	
Z#18	If R, move there with user's G90/G91
#702= #5003 - #26	
IF [ #9 EQ #0 ] #9= #4109	Use present F if none specified
IF [ #8 EQ #0 ] #8= #9	If no E, use F
G91	All incremental from here on
IF [ #4 EQ #0 ] #4= 0.0	
IF [ #5 EQ #0 ] #5= 1.0	
G68 R#4	
G51 P [ #5 * 1000 ]	
N1000	
M97	M97 auto M99 at end of string
GOTO1000	
N125	
M99	
(SPACE)	This section mills a space.
N126	
G00 X0.864 F#8	
M99	
N127	
G#700	Restore G90/G91
G#701	Restore G00/G01 etc.
M99	
N1	This section mills an exclamation point
(!)	
G00 X0.2692	
G01 Z - #702 F#8	
G03 J0.0297 F#9	
G00 Z#702	
G00 Y0.2079	
G01 Z - #702 F#8	
G01 X0.0495 Y0.6732 F#9	
G03 X-0.099 R0.0495	
G01 X0.0495 Y-0.6732	
G00 Z#702	
G00 X0.2692 Y-0.2079	
M99	
N2	This section mills double quotes.
(«)	
G00 X0.2345 Y0.792	
G01 Z - #702 F#8	
G01 X0.0148 Y0.198 F#9	
G01 X-0.0297	



## VS-Series

```

G01 X0.0148 Y-0.198
G00 Z#702
G00 X0.1485
G01 Z - #702 F#8
G01 X0.0148 Y0.198 F#9
G01 X-0.0297
G01 X0.0148 Y-0.198
G00 Z#702
G00 X0.2346 Y-0.792
M99
N3
(#)
G00 X0.4082 Y0.1666
G01 Z - #702 F#8
G01 X0.0433 Y0.8086 F#9
G00 Z#702
G00 X0.2627 Y0.0148
G01 Z - #702 F#8
G01 X-0.0433 Y-0.8234 F#9
G00 Z#702
G00 X0.2194 Y0.2374
G01 Z - #702 F#8
G01 X-0.6676 F#9
G00 Z#702
G00 X0.0155 Y0.319
G01 Z - #702 F#8
G01 X0.6614 F#9
G00 Z#702
G00 X0.2167 Y-0.723
M99

```

...  
%

This section mills a # sign.

% marks the end of the program.

For the creation of each character, there is a different label to start the code. Each section terminates with an M99.

Label Character	N126 space	N1 !	N2 "	N3 #	N4 \$	N5 %	N6 &	N7 '	N8 (	N9 )
Label Character	N10 *	N11 +	N12 ,	N13 -	N14 .	N15 /	N16 0	N17 1	N18 2	N19 3
Label Character	N20 4	N21 5	N22 6	N23 7	N24 8	N25 9	N26 :	N27 ;	N28 ,	N29 =
Label Character	N30 >	N31 ?	N32 @	N33 A	N34 B	N35 C	N36 D	N37 E	N38 F	N39 G
Label Character	N40 H	N41 I	N42 J	N43 K	N44 L	N45 M	N46 N	N47 O	N48 P	N49 Q
Label Character	N50 R	N51 S	N52 T	N53 U	N54 V	N55 W	N56 X	N57 Y	N58 Z	N59 [
Label Character	N60 \	N61 ]	N62 ^	N63 _	N64 `	N65 a	N66 b	N67 c	N68 d	N69 e

Label	N70	N71	N72	N73	N74	N75	N76	N77	N78	N79
Character	f	g	h	i	j	k	l	m	n	o
Label	N80	N81	N82	N83	N84	N85	N86	N87	N88	N89
Character	p	q	r	s	t	u	v	w	X	Y

Label	N90	N91	N92	N93
Character	Z	{		}

### COORDINATE ROTATION AND SCALING (G50, G51, G68, G69)

This control function is optional. If you would like further information on installing this feature, please call Haas Automation or your dealer for more information.

#### G50 Cancel Scaling

#### Group 11

G50 cancels scaling on all axes. Any axis scaled by a previous G51 command is no longer in effect.

#### G51 Scaling (This G-code is optional and requires Rotation and Scaling.)

#### Group 11

- X optional center of scaling for the X axis
- Y optional center of scaling for the Y axis
- Z optional center of scaling for the Z axis
- P optional scaling factor for all axes; three-place decimal from 0.001 to 8383.000.

G51 [X...] [Y...] [Z...] [P...]

When scaling is invoked, all subsequent X, Y, Z, I, J, K, or R values pertaining to machine motion are multiplied by a scaling factor and are offset relative to a scaling center.

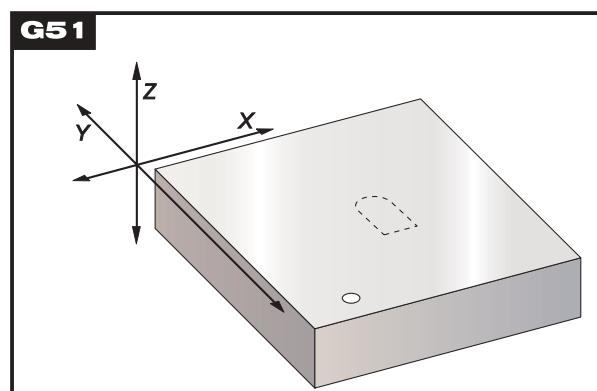
G51 is modal and modifies appropriate positional values in the blocks following the G51 command. It does not change or modify values in the block from which it is called. Axes X, Y, and Z are all scaled when the P code is used. If the P code is not used, the scaling factor currently in Setting 71 is used. The default scaling factor in Setting 71 is 1.0. A scaling factor of 1.0 means that no scaling is done. That is, all values are multiplied by 1.0 before being interpreted by the control.

A scaling center is always used by the control in determining the scaled position. If any scaling center is not specified in the G51 command block, then the current work coordinate position is used as the scaling center.

The following programs illustrate how scaling is performed when different scaling centers are used. All three examples call subroutine O0001, which follows.

```
0001 (GOTHIC WINDOW) ;
F20. S500 ;
G00 X1. Y1. ;
G01 X2. ;
Y2. ;
G03 X1. R0.5;
G01 Y1. ;
G00 X0 Y0 ;
M99 ;
```

○ = Work coordinate origin  
No Scaling

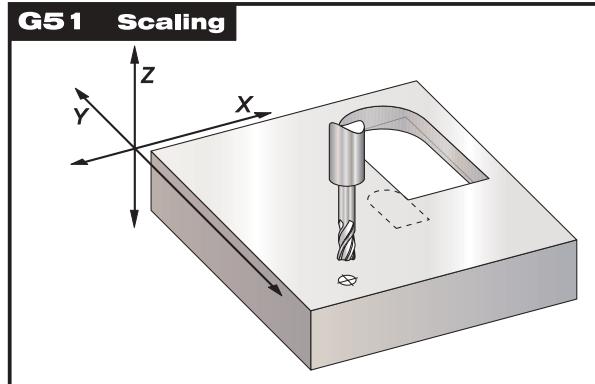




The first example illustrates how the control uses the current work coordinate location as a scaling center. Here, it is X0 Y0 Z0.

```
00010 ;  
G59 ;  
G00 G90 X0 Y0 Z0 ;  
M98 P1 ;  
G51 P2. (scaling center is X0 Y0 Z0) ;  
M30 ;
```

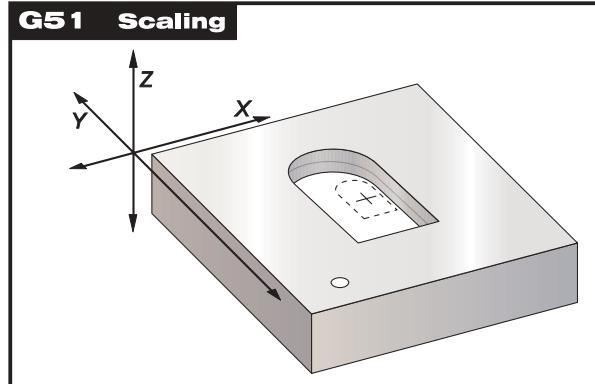
- = Work coordinate origin
- + = Center of scaling



The next example specifies the center of the window as the scaling center.

```
00011 ;  
G59 ;  
G00 G90 X0 Y0 Z0 ;  
M98 P1 ;  
G51 X1.5 Y1.5 P2. ;  
M98 P1 ;  
M30 ;
```

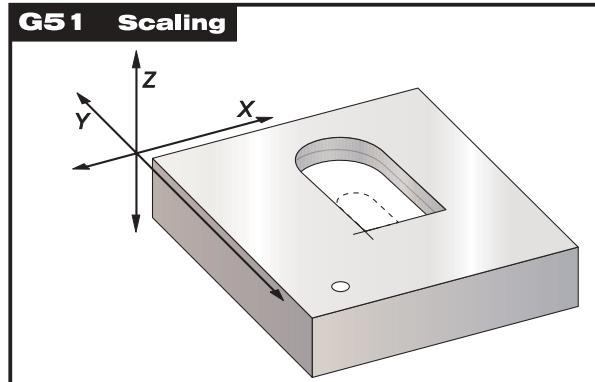
- = Work coordinate origin
- + = Center of scaling



The last example illustrates how scaling can be placed at the edge of tool paths as if the part was being set against locating pins.

```
00011 ;  
G59 ;  
G00 G90 X0 Y0 Z0 ;  
M98 P1 ;  
G51 X1.0 Y1.0 P2 ;  
M98 P1 ;  
M30 ;
```

- = Work coordinate origin
- + = Center of scaling



If macros are enabled, G65 arguments are not affected. Tool offsets and cutter compensation values are not affected by scaling.

The stored program is not changed by G51, so program lines displayed by the control will not reflect actual machine positions. Position displays WILL reflect the proper scaled values.

Scaling does not affect canned cycle Z axis movements such as clearance planes and incremental values. The final results of scaling are rounded to the lowest fractional value of the variable being scaled.

## G68      Rotation (This G-code is optional and requires Rotation and Scaling.)

## Group 16

G17, G18, G19      optional plane of rotation, default is current  
 A      optional center of rotation for the first axis of the selected plane  
 B      optional center of rotation for the second axis of the selected plane  
 R      optional angle of rotation specified in degrees  
 Three-place decimal –360.000 to 360.000.

[ G17 | G18 | G19 ] G68 [a...] [b...] [R...] ;

In the above example, **a** and **b** correspond to the axes of the current rotation plane. If G17 is the current rotation plane, then **a** is X and **b** is Y.

When rotation is invoked, all subsequent X, Y, Z, I, J, and K values are rotated through a specified angle **R** using a center of rotation.

G68 is modal and modifies appropriate positional values in the blocks following the G68 command. Values in the block containing G68 are not rotated. For subsequent blocks, only the values in the plane of rotation are rotated. Thus, if G17 is the current plane of rotation, only X and Y values are affected.

For a positive angle, rotation is counterclockwise. If the angle of rotation – the R code – is not specified in the G68 command block, then the angle of rotation is taken from Setting 72. The default rotation angle in Setting 72 is 0.0 degrees.

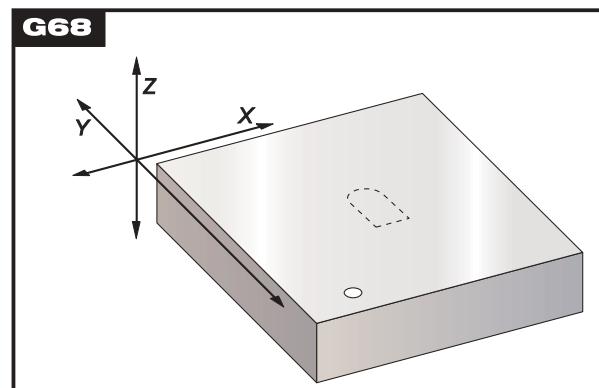
A center of rotation is always used by the control to determine the positional values passed to the control after rotation. If any axis' center of rotation is not specified, then the current location of the work coordinate is used as the center of rotation.

In G90 mode (absolute), the rotation angle takes on the value specified by **R**. When Setting 73 (G68 INCREMENTAL R) is set to ON, then the rotational value can be changed incrementally on each call of G68. In G91 mode (incremental), the rotation angle is changed incrementally by the value in **R**. Each G68 command block in G91 mode will change the rotation angle by the value specified in **R**. Angles are modulo 360, so that when an angle is specified as larger than 360 degrees, the angle will become an equivalent value between 0 and 360 degrees. The rotational angle is set to zero upon cycle start, or it can be set explicitly using a G68 block in the G90 mode.

The following examples illustrate rotation using G68.

```
0001 (GOTHIC WINDOW) ;
F20, S500 ;
G00 X1. Y1. ;
G01 X2. ;
Y2. ;
G03 X1. R0.5
G01 Y1. ;
M99 ;
```

 = Work coordinate origin  
No Rotation

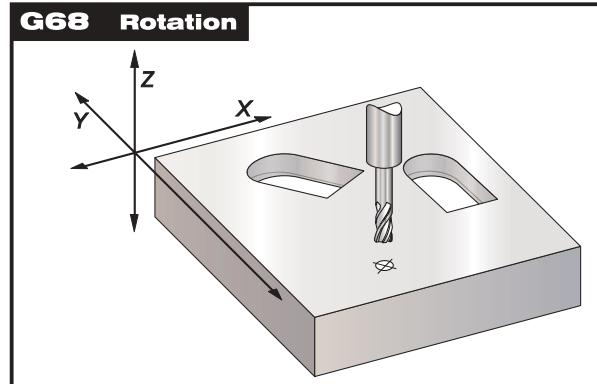




The first example illustrates how the control uses the current work coordinate location as a rotation center. Here, it is X0 Y0 Z0.

```
00002 ;  
G59 ;  
G00 G90 X0 Y0 Z0 ;  
M98 P1 ;  
G90 G00 X0 Y0 ;  
G68 R60. ;  
M98 P1 ;  
G69 G90 G00 X0 Y0 ;  
M30 ;
```

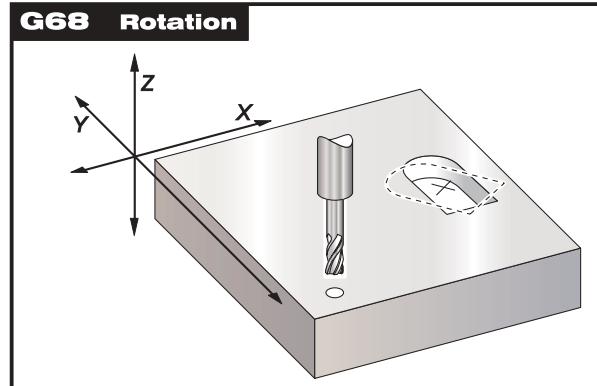
- = Work coordinate origin  
+ = Center of rotation



The next example specifies the center of the window as the rotation center.

```
00003 ;  
G59 ;  
G00 G90 X0 Y0 Z0 ;  
M98 P1 ;  
G00 G90 X0 Y0 Z0 ;  
G68 X1.5 Y1.5 R60. ;  
M98 P1 ;  
G69 G90 G00 X0 Y0 ;  
M30 ;
```

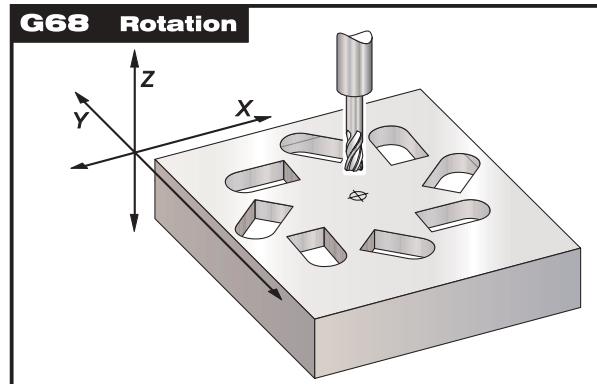
- = Work coordinate origin  
+ = Center of rotation



This example shows how the G91 mode can be used to rotate patterns about a center. This is often useful for making parts that are symmetric about a regular polygon.

```
00004 ;  
G59 ;  
G00 G90 X0 Y0 Z0 ;  
M98 P10 L8 (SUBROUTINE 00010) ;  
M30 ;  
  
00010 ;  
G91 G68 R45. ;  
G90 M98 P1 ;  
G90 G00 X0 Y0 ;  
M99 ;
```

- = Work coordinate origin  
+ = Center of rotation



Do not change the plane of rotation while G68 is in effect.

## ROTATION WITH SCALING

If scaling and rotation are used simultaneously, it is recommended that scaling be turned on prior to rotation, and that separate blocks be used. Use the following template when doing this.

```
G51 ..... (SCALING) ;
...
G68 ..... (ROTATION) ;
.
.
.
program.

.
.
.
G69 ..... (ROTATION OFF) ;
...
G50 ..... (SCALING OFF) ;
```

When rotating after scaling, any center specified as the center of rotation will be scaled. Any angle specified in the G68 block is NOT scaled. The control applies scaling and then rotation to any block with motion commands.

Below is an example of a program that has been scaled and rotated:

```
O0004 ;
G59 ;
G00 G90 X0 Y0 Z0 ;
M98 P1 ;
G90 G00 X0 Y0 ;
G51 P3.0 ;
G68 R60. ;
M98 P1 ;
G69 G51 G90 G00 X0 Y0 ;
M30 ;
```

## ROTATION WITH CUTTER COMPENSATION

Cutter compensation should be turned on after the rotation and scaling commands are issued. Compensation should also be turned off prior to turning rotation or scaling off.

### **G69 Cancel G68 Rotation (This G-code is optional and requires Rotation and Scaling.)**

**Group 16**

G69 cancels any rotation specified previously.

### **Work Coordinate System Selection (G52, G53, G54-59)**

---

**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 Set Work Coordinate System G52 YASNAC**

**Group 12**

This code selects the G52 work coordinate system. G52 works the same as G54, etc., except that the G52 system will be set by a G92 command as well as from the Offsets display.

YASNAC compatible.

**G52 Set Local Coordinate System FANUC****Group 00**

This code sets the origin of the local (child) coordinate system to the commanded location, relative to the current work system origin. G52 is a non-modal, no motion code. The G52 coordinate system will stay in effect for all work systems until it is canceled. The G52 command is canceled when RESET is pressed and at the end of a program. It is also canceled during a program by M30, G52 X0 Y0 Z0, or by a G92 command.

**G52 Set Local Coordinate System HAAS****Group 00**

This code works the same as in the Fanuc control except that G52 is not cleared at power-up, RESET, or when an M30 is performed.

**G53 Non-Modal Machine Coordinate Selection****Group 00**

This code temporarily cancels work coordinate offsets and uses the machine coordinate system. It is non-modal, so the next block will revert to whatever conditions were previously selected.

**G54-59 Select Coordinate System #1 - #6****Group 12**

These codes select one of the six user coordinate systems stored within the Offset memory. All subsequent references to axis positions will be interpreted in the new coordinate system. Work coordinate system offsets are entered from the Offsets display page.

**MORE MISCELLANEOUS G CODES (G60, G61, G64)****G60 Uni-Directional Positioning****Group 00**

This **G** code is used to provide positioning always from the positive direction. In older systems it was used to reduce backlash, and it is not recommended for use with this control. It is provided only for compatibility. It is non-modal, so does not affect the blocks that follow it. Setting 35 controls the distance an axis will travel past the target point prior to reversing for an approach in the positive direction.

**G61 Exact Stop Modal****Group 13**

The G61 code is used to specify an exact stop. It is modal and thus affects the blocks that follow it. Rapid and interpolated moves will decelerate to an exact stop before another block is processed. With exact stop in effect, moves will take longer and continuous cutter motion will not occur.

This may cause deeper cutting where the tool stops.

**G64 G61 Cancel (Select Normal Cutting Mode)****Group 13**

The G64 code is used to cancel exact stop. It is modal and thus affects subsequent blocks. Rapid and interpolated moves will not decelerate to an exact stop before another block is processed. Rapid blocks will decelerate to within the distance specified in Parameters 101-104 before another block is processed, and interpolated motion will not decelerate at all before the next block is processed.

**BOLT HOLE PATTERNS (G70, G71, G72)**

There are three **G** codes that provide patterns usually used for bolt holes. These are G70, G71, and G72. They are normally used with one of the Group 09 canned cycles. If the J value is not specified in a G70, G71 or G72 block, the control will assume it is zero. Also, if I, K, or L is needed and not specified, alarm 366 will be generated.

**G70 Bolt Hole Circle****Group 00**

- I Radius (minus reverses direction)
- J Starting angle (0 to 360.0 degrees CCW from horizontal)
- L Number of holes evenly spaced around the circle

This **G** code must be used with one of the canned cycles G73, G74, G76, G77, or G81-G89. 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 00 and thus is non-modal. For a G70 to work correctly, a canned cycle should be active so that at each position, some type of drill or tap function is performed.

**G71 Bolt Hole Arc****Group 00**

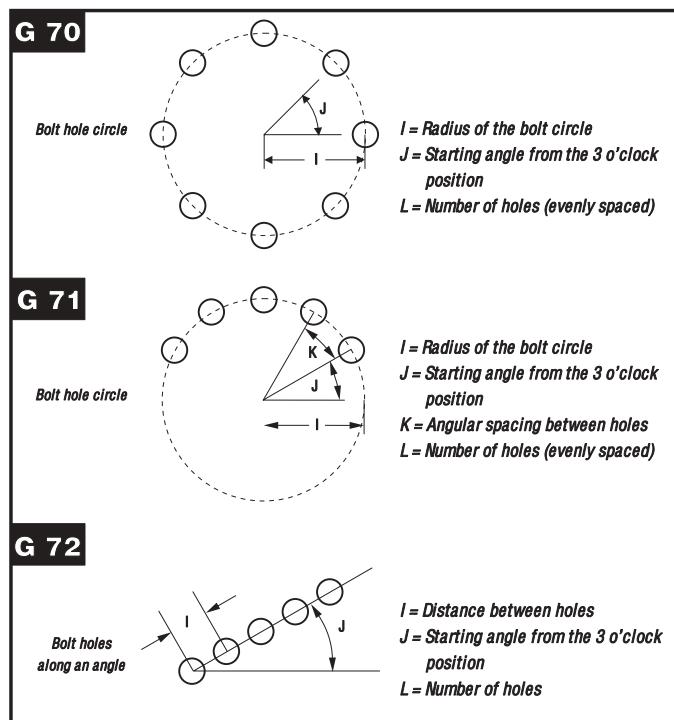
- I Radius
- J Starting angle (degrees CCW from horizontal)
- K Angular spacing of holes (+ or -)
- L Number of holes

This **G** code is similar to G70 except that it is not limited to one complete circle. G71 belongs to Group 00 and thus is non-modal. For a G71 to work correctly, a canned cycle should be active so that at each position, some type of drill or tap function is performed.

**G72 Bolt Holes Along an Angle****Group 00**

- I Distance between holes (minus will reverse direction)
- J Angle of line (degrees CCW from horizontal)
- L Number of holes

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 00 and thus is non-modal. For a G72 to work correctly, a canned cycle should be active so that at each position, some type of drill or tap function is performed.



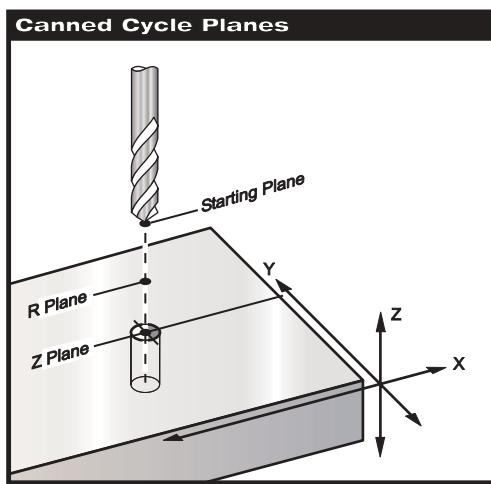


## CANNED CYCLES

### Introduction

A canned cycle is used to simplify programming of a part. Canned cycles are defined for most common Z-axis repetitive operations, such as drilling, tapping, and boring. Once selected, a canned cycle is active until canceled with G80. When active, the canned cycle is executed every time an X and/or Y-axis motion is programmed. Those XY motions are executed as rapid commands (G00) and the canned cycle operation is performed after the XY motion. There are five operations involved in every canned cycle:

1. Positioning of **X** and **Y** (and/or optional **A**) axes
2. Rapid traverse to **R** plane
3. Drilling, tapping or boring
4. Operation at bottom of hole
5. Retraction to **R** plane (G99) or initial starting point (G98)



### Canned Cycle Summary

G Code	Spindle at Start	Z Drilling direction	Operation at bottom of hole	Retraction Z Direction	Application
G73	—	intermittent feed	none	rapid	high speed peck drilling
G74	CCW	feed	spindle CW	feed	left hand tapping
G76	CW	feed then stop	orient spindle	rapid	fine boring
G77	CW	feed	spindle stop orient spindle	rapid	back boring
G81	—	feed	none	rapid	spot drilling
G82	—	feed	dwell	rapid	counter boring
G83	—	intermittent feed	none	rapid	peck drilling
G84	CW	feed	spindle CCW	feed	tapping cycle
G85	—	feed	none	feed	boring cycle
G86	CW	feed	spindle stop	rapid	boring cycle
G87	CW	feed	spindle stop	manual / rapid	back boring
G88	CW	feed	dwell, then spindle stop	manual / rapid	boring cycle
G89	—	feed	dwell	feed	boring cycle

## Using Canned Cycles

The positioning of a canned cycle in the X and/or Y axes can be done in either absolute (G90) or incremental (G91). Incremental (G91) motion in a canned cycle is often useful with a loop (Lnn) count which will repeat the canned cycle operation that many times with each incremental X or Y move for the canned cycle.

### Example:

G81 G99 Z-0.5 R0.1 F6.5 (This will drill one hole at the present location)

G91 X-0.5625 L9 (This will drill 9 more holes .5625 equally spaced in the minus direction)

If a canned cycle is defined without an **X** or **Y** and with a loop count of 0 (L0), the cycle will not be performed initially. The operation of the canned cycle will vary according to whether incremental (G91) or absolute (G90) positioning is active. Incremental motion in a canned cycle is often useful as a loop (**L**) count. It can be used to repeat the operation with an incremental **X** or **Y** move between each cycle.

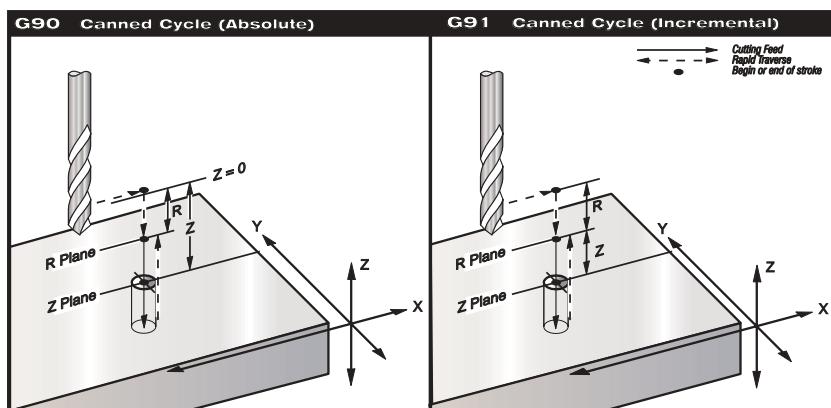
### Example:

X1.25 Y-0.75 (center location of bolt hole pattern)

G81 G99 Z-0.5 R0.1 F6.5 L0 (L0 on the G81 line will not drill a hole in the bolt hole circle)

G70 I0.75 J10. L6 (6-hole bolt hole circle)

Once a canned cycle is defined, that operation is defined at every X-Y position listed in a block. Some of the canned cycle numerical values can 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 listed in a block with XY commands, the XY move is done and all subsequent canned cycles are performed with the new **R** or **Z** value.



The positioning of the X and Y axes prior to a canned cycle is normally done with rapid moves, without an exact stop prior to plunging the Z axis to the **R** depth. This may cause a crash with a close-tolerance fixture. Setting 57 can be used to select an exact stop for these XY moves.

G98 and G99 are modal commands that change the way the canned cycles operate. When G98 is active, the Z axis will return to the initial start point at the completion of the canned cycle. This is used to gain clearance for the next XY hole location.

When G99 is active, the Z axis returns to the **R** (rapid) plane when the canned cycle is completed for clearance to the next XY location. Changes to the G98/G99 selection can also be made after the canned cycle is active. If changes are made to the G98/G99 value, all later canned cycles will be changed as well.

If a canned cycle is defined in a block without an **X** or **Y** motion, the control will either execute the command or not depending on the selection of Setting 28.

---

**Note:** Spindle control **M** codes should not be used while a canned cycle is active.



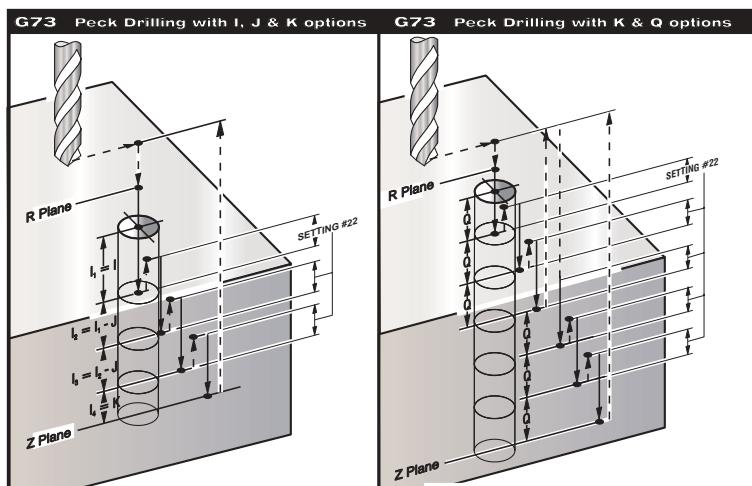
## Canceling a Canned Cycle

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

### G73 High-Speed Peck Drilling Canned Cycle

Group 09

- F Feedrate in inches (mm) per minute
- I Optional size of first cutting depth
- J Optional amount to reduce cutting depth on each pass
- K Optional minimum depth of cut / number of pecks between retract
- L Number of repeats
- P Optional pause at end of last peck, in seconds
- 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



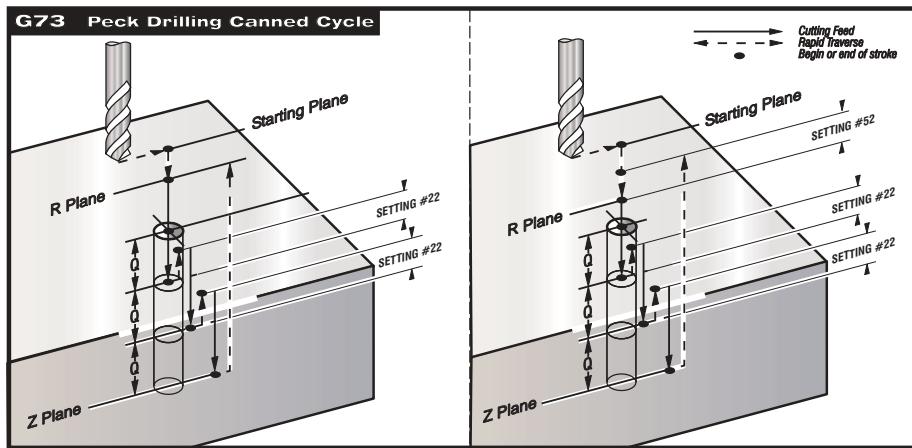
This G code is modal; once activated, every programmed motion of **X** and/or **Y** will cause the canned cycle to be executed, until it is canceled or another canned cycle is selected. This is a high-speed peck cycle in which the retraction distance is determined by Setting 22.

If **I**, **J**, and **K** are specified, a different operating mode is selected. The first pass will cut in by amount **I**, each succeeding cut will be reduced by amount **J**, and the minimum cutting depth is **K**. If **P** is specified, the tool will pause at the bottom of the hole after the last peck for that amount of time.

The same dwell time will apply to all subsequent blocks that do not specify a dwell time. When the canned cycle is cancelled (with G00, G01, G80, or RESET) the dwell time will be reset to zero. This dwell cannot be used in the same block as an M97, M98, M99, or G65, because these codes use **P** for different purposes.

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 the number of passes totals up to the **K** amount. This allows much faster drilling than G83 but still returns the spindle to the **R** plane occasionally to clear chips.

**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-clearing motion actually allows the chips to get out of the hole, but this causes 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 chip-clearing move to **R** occurs, the **Z** axis will be moved above **R** by this setting.

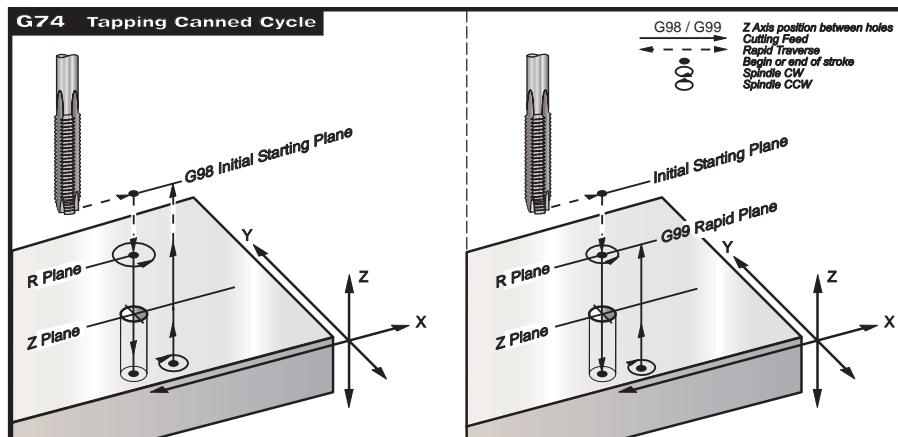
#### G74 Reverse Tap Canned Cycle

Group 09

- F Feedrate in inches (mm) per minute
- J Retract Multiple (see Setting 130)
- L Number of repeats
- R Position of the R plane
- X Optional X-axis motion command
- Y Optional Y-axis motion command
- Z Position of bottom of tap

This G code is modal. Once activated, every commanded motion of **X** and/or **Y** will cause this canned cycle to be executed, until it is canceled or another canned cycle is selected. Note that operation of this cycle is different if rigid tapping is selected. When rigid tapping is used, the ratio between the feedrate 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.

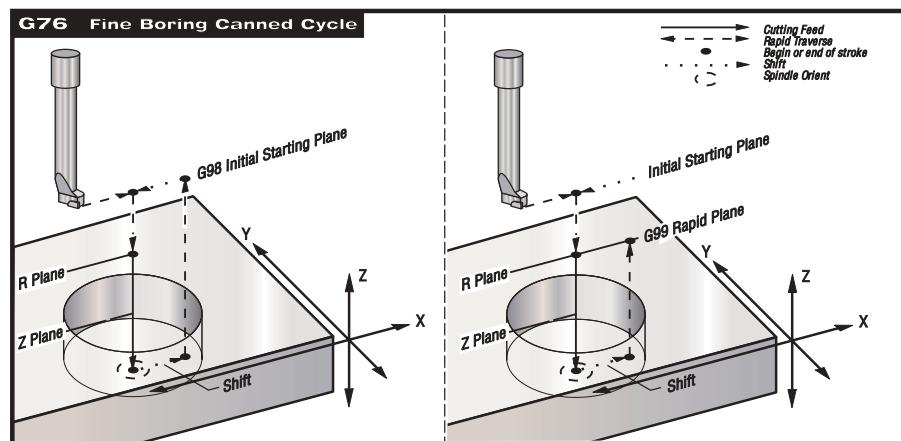




VS-Series

**G76 Fine Boring Canned Cycle****Group 09**

- F Feedrate in inches (mm) per minute
- I Optional shift value, if Q is not specified
- J Optional shift value, if Q is not specified
- L Number of repeats
- 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

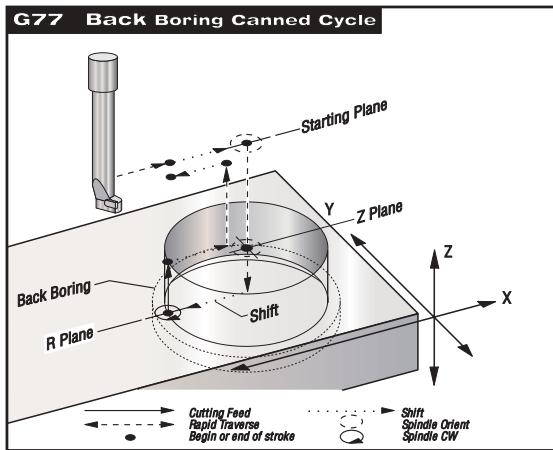


This G code is modal. Once activated, every commanded motion of **X** and/or **Y** will cause this canned cycle to be executed, until it is canceled or another canned cycle is selected. This cycle will shift the **X** and/or **Y** axis prior to retracting in order to clear the tool while exiting the part. The shift direction is determined by Setting 27. If Q is not specified, the optional I and J values are used to determine the shift direction and distance.

**G77 Back Bore Canned Cycle****Group 09**

- F Feedrate in inches (mm) per minute
- I Optional shift value, if Q is not specified
- J Optional shift value, if Q is not specified
- L Number of repeats
- 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

This G code is modal. Once activated, every commanded motion of **X** and/or **Y** will cause this canned cycle to be executed, until it is canceled or another canned cycle is selected. This cycle will shift the X and/or Y axis prior to and after cutting in order to clear the tool while entering and exiting the part. If Setting 57 is on, the tool will perform an exact stop between rapids. This will prevent tool breakage and any nicking at the bottom of the hole. The shift direction is determined by Setting 27. If Q is not specified, the optional I and J values are used to determine the shift direction and distance.



## G80 Canned Cycle Cancel

## Group 09

This G code is modal. It deactivates all canned cycles until a new one is selected. Note that use of G00 or G01 will also cancel a canned cycle.

## G81 Drill Canned Cycle

## Group 09

- F Feedrate in inches (mm) per minute
- L Number of repeats
- R Position of the R plane
- X Optional X-axis motion command
- Y Optional Y-axis motion command
- Z Position of bottom of hole

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

### G81 FORMAT

G81 Z\_\_ R\_\_ F\_\_

Z = Position of the bottom of the hole being drilled

R = Reference plane above Z0

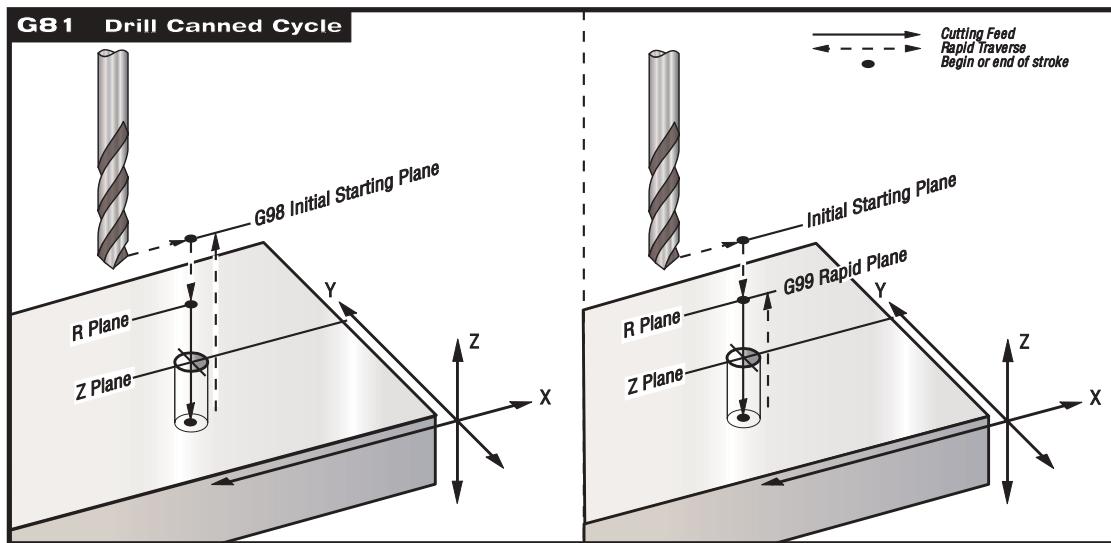
F = Feedrate in inches per minute

**NOTE:** The **Z**, **R**, and **F** codes are required data for all canned cycles.

**NOTE:** The optional **X** and **Y** can be included in the canned cycle line. In most cases, this would be the location of the first hole to be drilled.



VS-Series



Canned cycle programming example using aluminum block.

## PROGRAM EXAMPLE

The following is a program to drill through an aluminum plate:

```
T1 M06  
G00 G90 G54 X1.125 Y-1.875 S4500 M03  
G43 H01 Z0.1  
G81 G99 Z-0.35 R0.1 F27.  
X2.0  
X3.0 Y-3.0  
X4.0 Y-5.625  
X5.250 Y-1.375  
G80 G00 Z1.0  
G28  
M30
```

**G82 Spot Drill Canned Cycle**
**Group 09**

- F Feedrate in inches (mm) per minute
- L Number of repeats
- 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

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

**G82 FORMAT**

G82 Z\_\_\_\_ P\_\_\_\_ R\_\_\_\_ F\_\_\_\_ ;

These are the required codes for **spot drilling**.

P = Dwell time, in milliseconds, at the bottom of the Z-axis move.

For example, P300 is approximately 1/3 second.

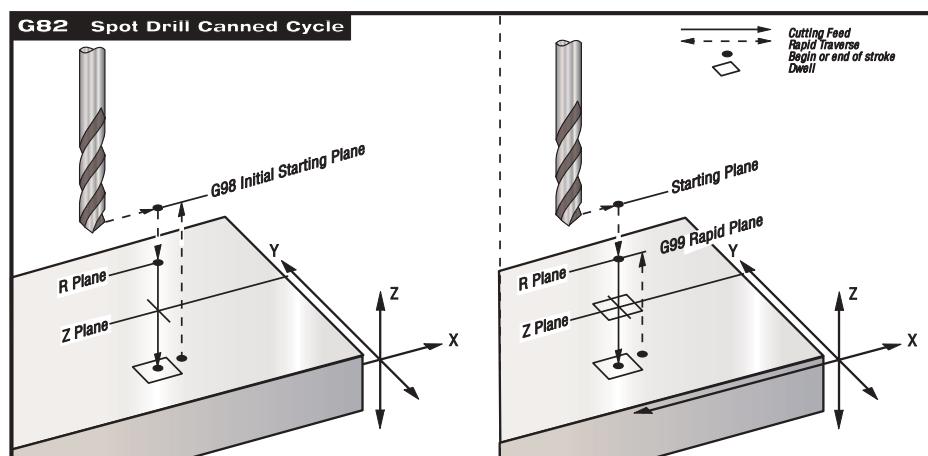
**PROGRAM EXAMPLE**

```
%  
O1234  
T1 M06  
G90 G54 G00 X.565 Y-1.875 S1275 M03  
G43 H01 Z0.1 M08  
G82 Z-0.175 P.3 R0.1 F10.  
X1.115 Y-2.750  
X3.365 Y-2.875  
X4.188 Y-3.313  
X5.0 Y-4.0  
G80 G00 Z1.0 M09
```

**DESCRIPTION**

(Sample program)  
(Tool #1 is a 0.5" x 90-degree spot drill)

(90-degree spot drill; the depth is)  
(half of the chamfer diameter)



*G82 Spot drilling example*

**G83 Normal Peck Drilling Canned Cycle****Group 09**

F	Feedrate 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
L	Number of repeats
P	Optional pause at end of last peck, in seconds
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

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

**G83 FORMAT**

G83 Z\_\_\_\_ Q\_\_\_\_ R\_\_\_\_ F\_\_\_\_ ;

These are the required codes for **peck drilling**.

Q = Incremental pecking amount in minus **Z** direction.

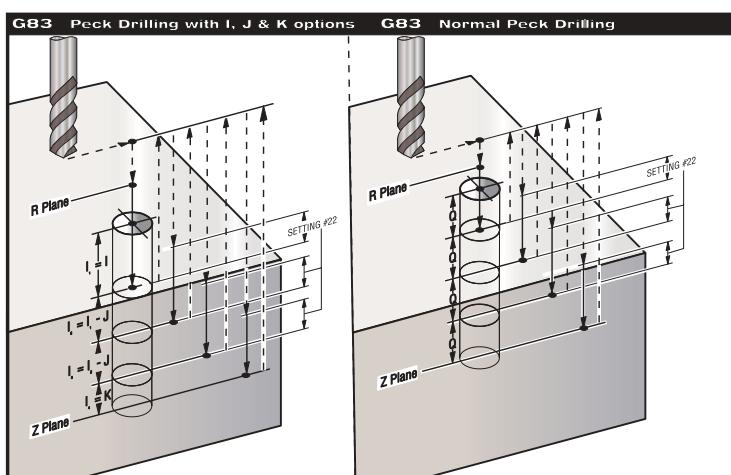
For example, Q.200 in a G83 line will peck toward the specified **Z** depth, taking 0.200" per peck until final depth is reached.

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

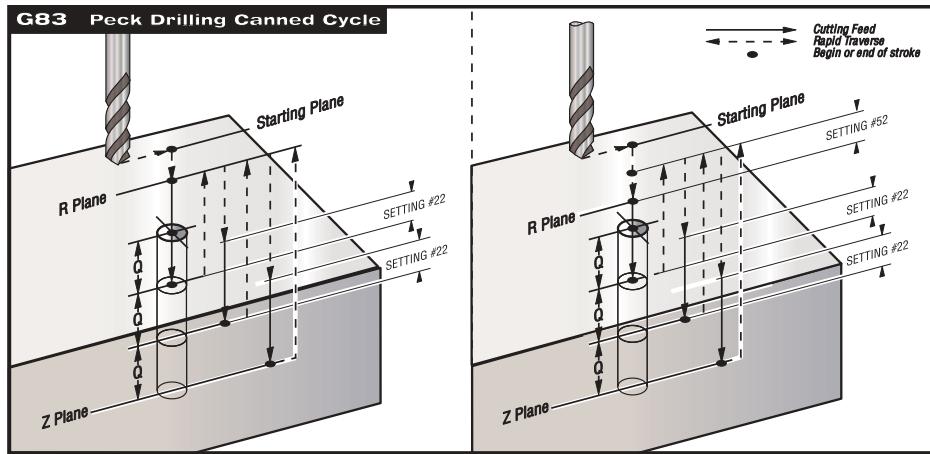
If **P** is specified, the tool will pause at the bottom of the hole after the last peck for that amount of time. The following example will peck several times and dwell for 1.5 seconds at the end:

G83 Z-0.62 F15. R0.1 Q0.175 P1.5.

The same dwell time will apply to all subsequent blocks that do not specify a dwell time. When the canned cycle is canceled (with G00, G01, G80, or RESET), the dwell time will be reset to zero. This dwell cannot be used in the same block as an M97, M98, M99, or G65, because these codes use **P** for different purposes.



Setting 52 changes the way G83 works when it returns to the **R** plane. Most programmers set the **R** plane well above the cut to ensure that the chip-clearing motion actually allows the chips to get out of the hole, but this causes 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 chip-clearing move to **R** occurs, the **Z** axis distance above **R** is determined by this setting.



### PROGRAM EXAMPLE

```

T2 M06
G90 G54 G00 X0.565 Y-1.875 S2500 M03
G43 H02 Z0.1 M08
G83 Z-0.720 Q0.175 R0.1 F15.
X1.115 Y-2.750
X3.365 Y-2.875
X4.188 Y-3.313
X5.0 Y-4.0
G80 G00 Z1.0 M09

```

### DESCRIPTION

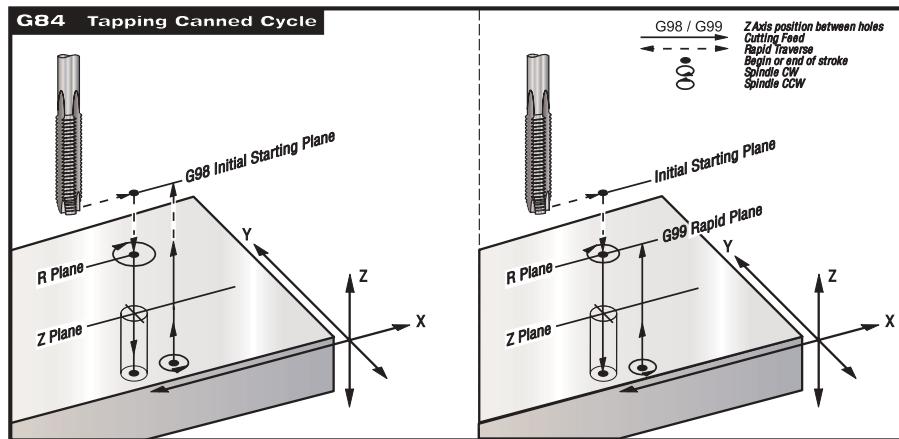
(Tool #2 is a 0.3125" stub drill)

(Drill point is 1/3 of the drill diameter)

### G84 Tapping Canned Cycle

### Group 09

- F Feedrate in inches (mm) per minute
- J Retract Multiple (see Setting 130)
- L Number of repeats
- R Position of the R plane
- X Optional X-axis motion command
- Y Optional Y-axis motion command
- Z Position of bottom of tap



G 84 Tapping Canned Cycle Example



This G code is modal. Once activated, every commanded motion of **X** and/or **Y** will cause this canned cycle to be executed, until it is canceled or another canned cycle is selected. Note that operation of this cycle is different if rigid tapping is selected (see Rigid Tapping in the Programming section of this manual). When rigid tapping is used, the ratio between the feedrate 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.

## G84 FORMAT

G84 Z\_\_\_\_ R\_\_\_\_ F\_\_\_\_

These are the codes required for **tapping**.

No new codes to review.

---

**NOTE:** The most important item to be aware of during tapping is the speed and feed calculation.

**FEED FORMULA:** Spindle speed divided by Threads per inch of the tap = Feedrate in inches per minute

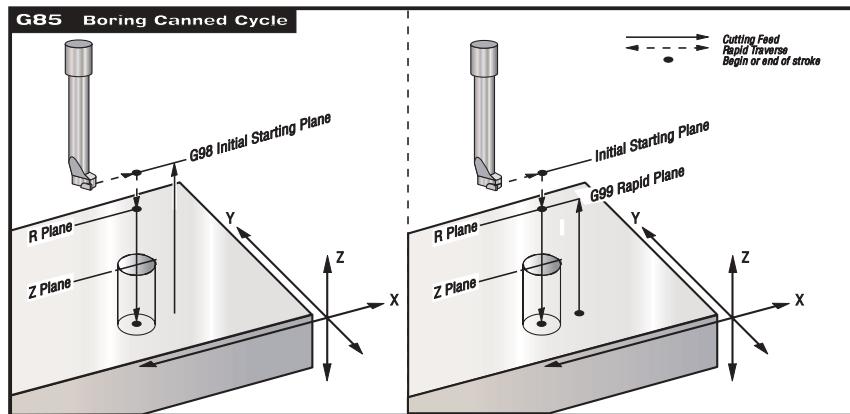
### Canned cycle PROGRAM EXAMPLE using G84

Helpful notes are listed in parentheses ( ).

T3 M06	(Tool #3 is a 3/8-16 tap)
G90 G54 G00 X0.565 Y-1.875 S900 M03	
G43 H03 Z0.2 M08	
G84 Z-0.600 R0.2 F56.25	(900 rpm divided by 16 tpi = 56.25 ipm)
X1.115 Y-2.750	
X3.365 Y-2.875	
X4.188 Y-3.313	
X5.0 Y-4.0	
G80 G00 Z1.0 M09	
G28 G91 Y0 Z0	
M30	
%	

**G85 Boring Canned Cycle**
**Group 09**

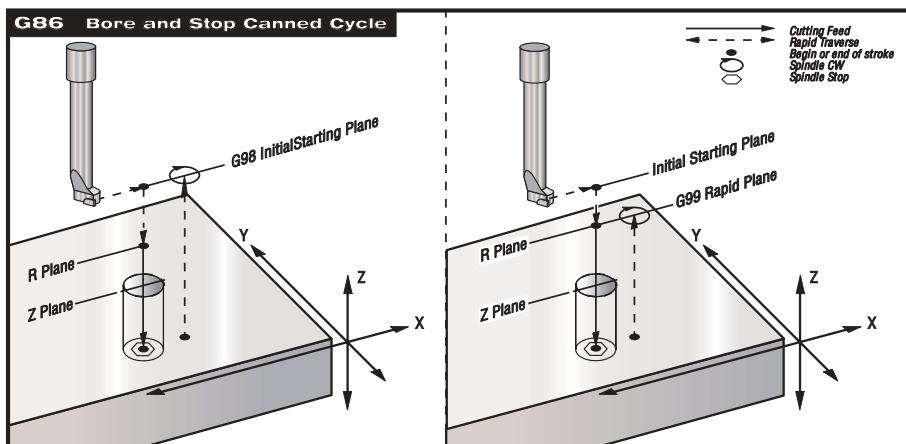
- F Feedrate in inches (mm) per minute
- L Number of repeats
- R Position of the R plane
- X Optional X-axis motion command
- Y Optional Y-axis motion command
- Z Position of bottom of hole



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

**G86 Bore and Stop Canned Cycle**
**Group 09**

- F Feedrate in inches (mm) per minute
- L Number of repeats
- R Position of the R plane
- X Optional X-axis motion command
- Y Optional Y-axis motion command
- Z Position of bottom of hole



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

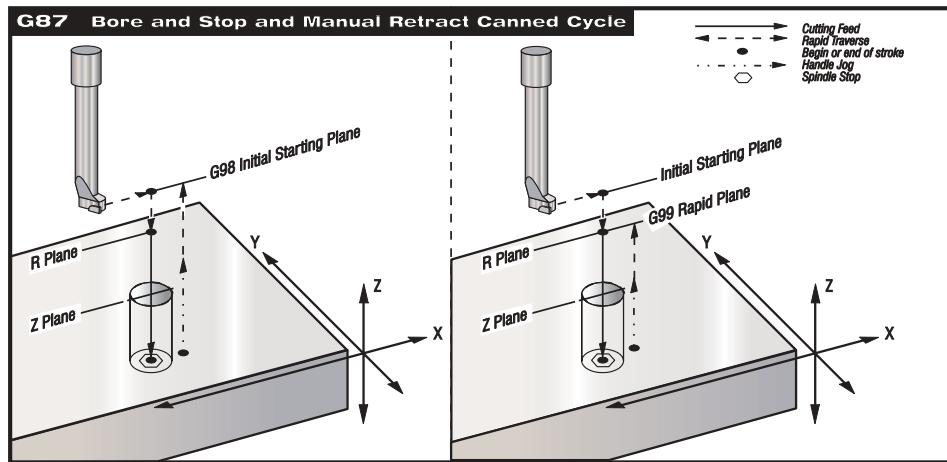


VS-Series

**G87****Bore In and Manual Retract Canned Cycle****Group 09**

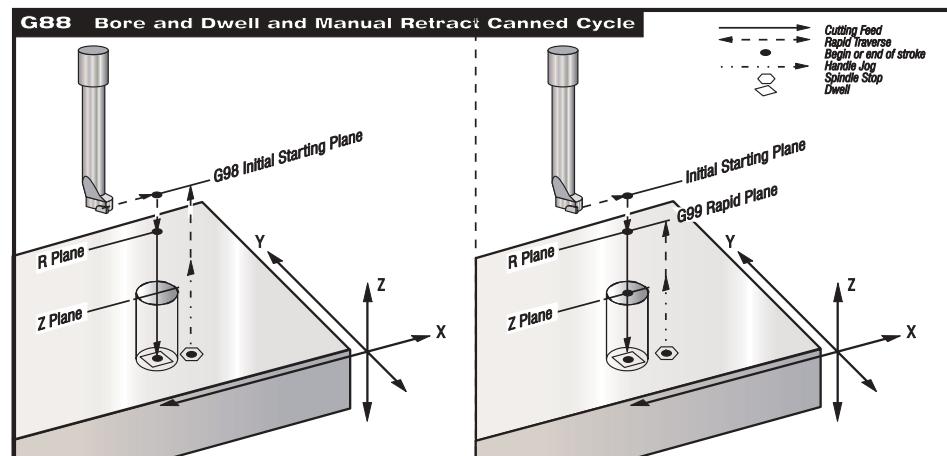
- F Feedrate in inches (mm) per minute
- L Number of repeats
- R Position of the R plane
- X Optional X-axis motion command
- Y Optional Y-axis motion command
- Z Position of bottom of hole

This G code is modal. Once activated, every commanded motion of **X** and/or **Y** will cause this canned cycle to be executed, until it is canceled or another canned cycle is selected. After manually jogging the axes, press CYCLE START to resume the program.

**G88****Bore In, Dwell, Manual Retract Canned Cycle****Group 09**

- F Feedrate in inches (mm) per minute
- L Number of repeats
- 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

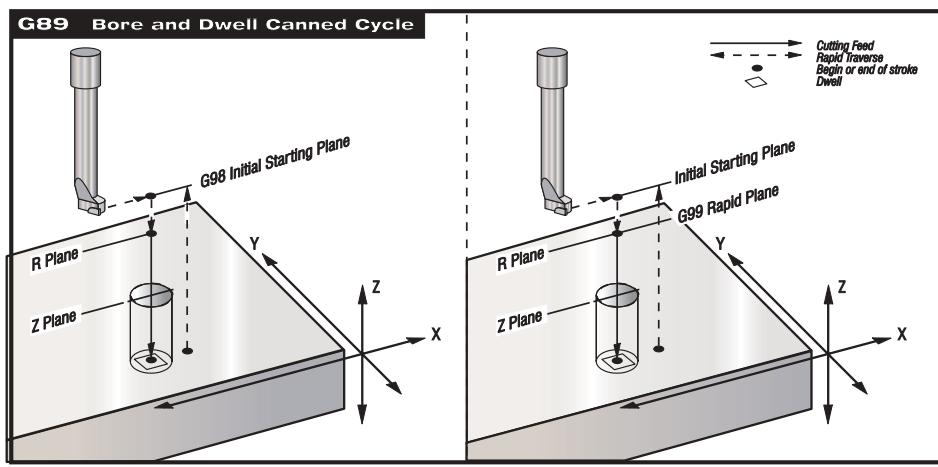
This G code is modal. Once activated, every commanded motion of **X** and/or **Y** will cause this canned cycle to be executed, until it is canceled or another canned cycle is selected. After manually jogging the axes, press CYCLE START to resume the program.



**G89      Bore In, Dwell, Bore Out Canned Cycle**
**Group 09**

- F Feedrate in inches (mm) per minute
- L Number of repeats
- 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

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


**Absolute / Incremental Selection (G90, G91)**
**G90      Absolute Position Commands**
**Group 03**

This code is modal and changes the way axis motion commands are interpreted. G90 makes all subsequent commands within the selected user coordinate system absolute positions. Each axis moved will be placed at the position coded in the command block.

**G91      Incremental Position Commands**
**Group 03**

This code is modal and changes the way axis motion commands are interpreted. G91 makes all subsequent commands incremental. Each axis moved will be moved by the amount coded in the command block. This code is not compatible with G143 (5-Axis Tool Length Compensation).

**MORE WORK COORDINATE SELECTION (G92)**

The way this command works depends on Setting 33, which selects a FANUC, HAAS, or YASNAC coordinate system. This command does not move any of the axes; it only changes the values stored as user work offsets.

**G92      Set Work Coordinate Systems Shift Value – FANUC OR HAAS**
**Group 00**

A G92 command effectively shifts all work coordinate systems (G54-59, G110-129) so that the commanded position becomes the current position in the active work system. G92 is a non-modal, non-motion code.

A G92 command cancels any G52 in effect for the commanded axes. Example: G92 X1.4 cancels the G52 for the X axis. The other axes are not affected.



The G92 shift value is displayed at the bottom of the Work Offsets page and may be cleared there if necessary. It is also cleared automatically after power-up, and any time ZERO RET and AUTO ALL AXES or ZERO SINGLE AXIS are used.

## G92 Set Work Coordinate Systems Shift Value - YASNAC

**Group 00**

A G92 command sets the G52 work coordinate system so that the commanded position becomes the current position in the active work system. The G52 work system then automatically becomes active until another work system is selected. G92 is a non-modal, non-motion code.

### *INVERSE TIME (G93, G94, G95)*

## G93 Inverse Time Feed Mode

**Group 05**

This G code specifies that all F (feedrate) values are to be interpreted as **strokes per minute**. This is equivalent to saying that the F code value, when DIVIDED INTO 60, is the number of seconds that the motion should take to complete.

G93 is generally used in 5-axis work, and sometimes in 4-axis work as well. It's a way of translating the linear (inches/min) feedrate assigned to the program – F30, say – into a value that takes rotary motion into account. In G93 mode, the F value will tell you how many times per minute the stroke (tool move) can be repeated, based on the linear F value.

G93 activates Inverse Time Feed Mode and G94 deactivates it.

Any interpolated motion that involves only the auxiliary axes is NOT affected by G93 – the F code specified will still be interpreted as feed per minute.

When G93 is active, the feedrate specification is MANDATORY for all interpolated motion blocks; i.e., each non-rapid motion block MUST have its own feedrate specification. If it doesn't, a **NO FEEDRATE** alarm is generated. Mixing auxiliary axes with regular axes in a G01/02/03 motion in G93 mode will generate the alarm:

### **"AUX AXIS IN G93 BLOCK"**

\* All Group 9 motion commands, as well as any G12, G13, G70, G71, G72, or G150 command, will generate a syntax alarm when in G93 mode.

\* Pressing RESET will reset the machine to G94 (Feed per Minute) mode.

\* Settings 34 and 79 (4th & 5th axis diameter) are meaningless in inverse time.

\* Alarm 309, "**EXCEEDED MAX FEEDRATE**," will not be generated by G93, because the machine will automatically be limited by the slowest axis.

## G94 Feed Per Minute Mode

**Group 05**

This code deactivates G93 (Inverse Time Feed Mode) and returns the control to Feed Per Minute mode. Pressing RESET returns the machine to G94 mode.

## G95 FEED PER REVOLUTION

When G95 is active, a spindle revolution will result in a travel distance specified by the Feed value. If the Setting 9 DIMENSIONING is set to INCH, then the Feed value F will be taken as inches/Rev. If Setting 9 is set to MM, then the Feed will be taken as mm/Rev. The G95 mode will stay effective until another feed mode such as G94 Feed Per Minute has been specified. Note that the default at power-up is G94. Feed Override and Spindle override will affect the behavior of the machine while G95 is active. When a Spindle override is selected, any change in the Spindle will result in a corresponding change in Feed in order to keep the chip load uniform. However, if a Feed override is selected, then any change in the Feed override will only affect the Feed rate and not the spindle. A command block that contains a Tap cycle will ignore any Feed and spindle overrides and perform that block at 100%. Setting 56 RESTORE DEFAULT G will determine whether or not an M30 should reset the Feed Per Rev mode. If this setting is enabled, then an M30 at the end of a program will reset this mode to its default G94 Feed Per Minute mode. Alarm 309 EXCEEDED MAX FEEDRATE will be generated if the combination of Spindle speed and Feed value (F) exceeds the limit specified by Parameter 59 MAX FEED IN/MIN.

### CANNED CYCLE AUXILIARY FUNCTIONS (G98, G99)

#### G98 Canned Cycle Initial Point Return

**Group 10**

This G code is modal and changes the way canned cycles operate. With G98, the Z axis returns to its initial starting point (the Z position it was at when the canned cycle was first defined) between each new X and/or Y location. This allows for positioning up and around areas of the part and/or clamps and fixtures that are in the tool path.

#### G99 Canned Cycle R Plane Return

**Group 10**

This G code is modal and changes the way canned cycles operate. With G99, the Z axis will stay down at the R plane between each new X and/or Y location, when no obstruction is in the path of the tool to the next location.

### PROGRAMMABLE MIRROR IMAGE (G100, G101)

#### G100 Cancel Mirror Image

**Group 00**

#### G101 Enable Mirror Image

**Group 00**

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

At least one of the axis commands is required. Settings 45 through 48 are used to select mirror imaging.

Programmable mirror imaging can be turned on or off individually for any of the four axes. When any one of these is OFF, axis motions occur normally. When one is ON, axis motion may be mirrored (or reversed) around the work zero point. The two G codes (G100 and G101) are non-modal, but the mirror image status of each axis is modal. The bottom of the screen will indicate when an axis is mirrored. These G codes should be used in a command block without any other G codes. They do not cause any axis motion. G101 will turn on mirror imaging for any axis listed in that block. G100 will turn off mirror imaging for any axis listed in the block. The actual value given for the X, Y, Z, or A code has no effect. G100 or G101 by itself will have no effect and should be entered as zero value.

#### Example:

- G101 X0 = Will turn on mirror imaging for the X axis.
- G100 X0 = Will turn off mirror imaging for the X axis.



Most mirror imaging applications would consist of irregular pockets and contours, and would most likely be set up in subprograms for convenience.

When using cutter compensation with mirror imaging, follow this guideline: After turning mirror imaging on or off with G100 or G101, the next motion block should be to a different work coordinate position than the first one.

The following code is an example:

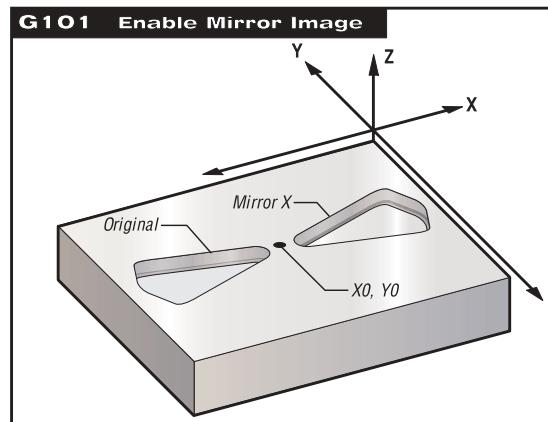
Incorrect:	Correct:
G41 X1.0 Y1.0	G41 X1.0 Y1.0
G01 X2.0 Y2.0	G01 X2.0 Y2.0
G101 X0	G101 X0
G00 Z1.0	G00 Z1.0
G00 X2.0 Y2.0	G00 X1.0
	G00 X2.0 Y2.0
G40	G40.

The mirror function can change the direction of motion along any of the axes. If any one of these are selected, the display will show the status. Mirror imaging will reflect programmed motion around your work coordinate zero point. **Be careful:** mirroring only one of the **X** or **Y** axes will cause the cutter to move along the opposite side of a cut. In addition, if mirror imaging is selected for only one axis of a circular motion plane, then circular motion commands G02 and G03 are reversed, and left and right side cutter compensation commands G41 and G42 are reversed.

---

**NOTE:** After completion of the first item, a Z-axis clearance move should be made. Then, the mirror imaging should be turned on with an axis specification. The next line needs the coordinates of the starting location of the original pocket. The next line will feed to the required Z-axis depth, the next would contain a subprogram call or a contour definition, and last would be a positive Z-axis clearance move.

The pockets should be arranged around a given origin, usually described as X0,Y0.



Mirror image and pocket milling exercise for O3600.

---

**NOTE:** When milling a shape with XY motions, turning **on** MIRROR IMAGE for just one of the X or Y axes will change climb milling to conventional milling and/or conventional milling to climb milling. As a result, you may not get the type of cut or finish that was desired. Mirror imaging of both X and Y will eliminate this problem.

## PROGRAM CODE FOR MIRROR IMAGING IN THE X AXIS:

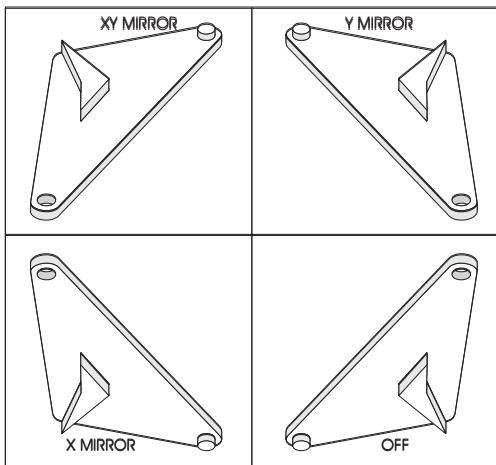
### PROGRAM EXAMPLE

```
%  
O3600  
T1 M06  
G00 G90 G54 X-.4653 Y.052 S5000 M03  
G43 H01 Z.1 M08  
G01 Z-.25 F5.  
F20.  
M98 P3601  
G00 Z.1  
G101 X0.  
X-.4653 Y.052  
G01 Z-.25 F5.  
F20.  
M98 P3601  
G00 Z.1  
G100 X0.  
G28 G91 Y0 Z0  
M30  
%  
%  
O3601  
G01 X-1.2153 Y.552  
G03 X-1.3059 Y.528 R.0625  
G01 X-1.5559 Y.028  
G03 X-1.5559 Y-.028 R.0625  
G01 X-1.3059 Y-.528  
G03 X-1.2153 Y-.552 R.0625  
G01 X-.4653 Y-.052  
G03 X-.4653 Y.052 R.0625  
M99  
%
```

### DESCRIPTION

(Mirror image X axis)  
(Tool #1 is a 0.250" diameter endmill)

## PROGRAM CODE FOR MIRROR IMAGE IN THE X, Y, AND XY AXES:





VS-Series

### **PROGRAM EXAMPLE**

```
%  
O3700  
T1 M06  
G00 G90 G54 X-.2923 Y.3658 S5000 M03  
G43 H01 Z.1 M08  
G01 Z-.25 F5.  
F20.  
M98 P3701  
G00 Z.1  
G101 X0.  
X-.2923 Y.3658  
G01 Z-.25 F5.  
F20.  
M98 P3701  
G00 Z.1  
G100 X0.  
G101 Y0.  
X-.2923 Y.3658  
G01 Z-.25 F5.  
F20.  
M98 P3701  
G00 Z.1  
G100 Y0.  
G101 X0. Y0.  
X-.2923 Y.3658  
G01 Z-.25 F5.  
F20.  
M98 P3701  
G00 Z.1  
G100 X0. Y0.  
G28 G91 Y0 Z0  
M30  
  
O3701  
G01 X-.469 Y1.2497  
G03 X-.5501 Y1.2967 R.0625  
G01 X-1.0804 Y1.12  
G03 X-1.12 Y1.0804 R.0625  
G01 X-1.2967 Y.5501  
G03 X-1.2497 Y.469 R.0625  
G01 X-.3658 Y.2923  
G03 X-.2923 Y.3658 R.0625  
M99  
%
```

### **DESCRIPTION**

(Mirror image X, Y, and XY axes)  
(Tool #1 is a 0.250" diameter endmill)

(Turn on mirror image X axis)  
(Position to original coordinates)  
(Feed to Z depth)  
(Pocket feedrate)  
(Pocket contour subprogram call)  
(Part clearance)  
(Cancel mirror image X axis)  
(Turn on mirror image Y axis)

(Cancel mirror image Y axis)  
(Turn on mirror image X and Y axes)

(Cancel mirror image X and Y axes)

(Contour subprogram)

## *PROGRAMMABLE OUTPUT TO RS-232 (G102)*

### **G102 Programmable Output to RS-232**

**Group 00**

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

Programmable output to the RS-232 port allows the current work coordinates of the four axes to be output. This **G** code (G102) is non-modal, so it only affects the block in which it is programmed. It should be used in a command block without any other **G** codes and it will not cause any axis motion. The actual value given for the **X**, **Y**, **Z**, or **A** code has no effect. One complete line of text is sent to the first RS-232 port (same one used for upload, download, and DNC). Each axis listed in the G102 command block is output to the RS-232 port in the same format as values displayed in a program.

Optional spaces (Setting 41) and EOB control (Setting 25) are applied. The values sent out are always the current axis positions referenced to the current work coordinate system.

Digitizing of a part is possible using this **G** code and a program which steps over a part in **XY** and probes downward in **Z** with a G31. When the probe hits, the next block could be a G102 to send the **X**, **Y**, **Z** position out to a computer which could store the coordinates as a digitized part.

## *LIMIT BLOCK BUFFERING FUNCTION (G103)*

### **G103 Limit Block Buffering**

**Group 00**

P = 0-15 Maximum number of blocks the control will look ahead

G103 [P..]

"Block Look-ahead" is a term used to describe what the control is doing in the background during machine motion. A motion block may take several seconds to execute. The control can take advantage of this by preparing additional blocks of the program ahead of time. While the current block is executing, the next block has already been interpreted and prepared for continuous, uninterrupted motion between consecutive blocks. Block look-ahead is also important for obtaining the information necessary for predicting compensated positions for cutter compensation.

When G103 P0 is programmed, block limiting is disabled. Block limiting is also disabled if G103 appears in a block without a P address code.

When G103 Pn is programmed, look-ahead is limited to n blocks.

At this time G103 cannot be used if cutter compensation, G41 or G42, is in effect. Alarm 387 (Cutter Comp Not Allowed with G103) is generated if you attempt to do so.

G103 is also useful for debugging programs using macros. Macro expressions are executed at look-ahead time. By inserting a G103 P1 into the program, macro expressions will be performed one block ahead of the currently executing block.

G103 is not a Fanuc-compatible command.

**CYLINDRICAL MAPPING (G107)****G107 Cylindrical Mapping****Group 00**

X	Optional X-axis command	A	Optional A-axis command
Y	Optional Y-axis command	Q	Optional diameter of cylindrical surface
Z	Optional Z-axis command	R	Optional radius of rotary axis

This G code translates all programmed motion occurring in a specified linear axis into the equivalent motion along the surface of a cylinder (attached to a rotary axis). It is a Group 0 G code, but its default operation is subject to Setting 56 (M30 RESTORE DEFAULT G). The G107 command is used to either activate or deactivate cylindrical mapping.

- Any linear-axis program can be cylindrically mapped to any rotary axis (one at a time).
- An existing linear-axis G-code program can be cylindrically mapped without modification by inserting a G107 command at the beginning of the program. If Setting 56 is OFF, it can even be specified in a previous program as long as RESET has not been used.
- The radius (or diameter) of the cylindrical surface can be redefined, allowing cylindrical mapping to occur along surfaces of different diameters without having to change the program.
- The radius (or diameter) of the cylindrical surface can either be synchronized with or be independent of the rotary axis diameter(s) specified in the Settings page.
- G107 can also be used to set the default diameter of a cylindrical surface, independently of any cylindrical mapping that may be in effect.

**G107 Description**

1) Three optional parameters (address codes) can accompany G107: **X**, **Y** or **Z**; **A** or **B**; and **Q** or **R**.

**X, Y, or Z:** An X, Y, or Z address specifies the linear axis that will be mapped to the specified rotary axis (A or B). When one of these linear axes is specified, a rotary axis must also be specified, or an alarm will be generated.

**A or B:** An A or B address identifies which rotary axis holds the cylindrical surface.

**Q or R:** Q specifies the diameter of the cylindrical surface, while R specifies the radius. When Q or R is specified, a rotary axis must also be specified, or an alarm will be generated at run time. If neither Q nor R is specified, then the last G107 diameter specified since power-up will be assumed. If no G107 command has been issued since power-up, or if the last value specified was zero, then the diameter will be whatever is currently defined in the Settings page for this rotary axis. When Q or R is specified, that value will become the new G107 default for the specified rotary axis until the machine is turned off.

If this parameter has not been specified in a G107 command since the machine was powered up, or if its last specified value was zero, or if its last specified value was numerically equal to that defined for the rotary axis on the Settings page, then any change made to the rotary axis diameter on the Settings page will also change the default value for this parameter. In such a situation, the default G107 diameter is said to be "in sync" with the Settings.

However, if the last non-zero G107 diameter specified since power-on was *not* numerically equal to the Settings page value, then the G107 diameter is "out of sync" with the Settings page, and the default value thus established will be retained regardless of any changes made to the Settings page, until or unless they become equal in value.

This means that the rotary axis diameter as specified on the Settings page is always used as the default cylindrical mapping surface diameter, unless a previously issued G107 has established a different value. This allows the default cylindrical surface diameter to either be synchronized with, or operate independently from, the rotary axis diameter defined in the Settings.

To force the default cylindrical surface diameter to be "in sync" with the Settings page rotary axis diameter (which may be desirable if the current default diameter status is unknown), the following G107 command can be used:

### **G107 A0 Q0**

This command will reset the A-axis default cylindrical surface diameter to zero, which forces the Settings page value to be used as the default for that axis.

R specifies the radius of the rotary axis. If Q has already been specified, R will be ignored. Except as noted above, the conditions and/or restrictions associated with the use of this parameter are exactly the same as for Q.

- 2)** The above parameters can be entered in any order.
- 3)** Pressing the RESET key will turn off any cylindrical mapping that is currently in effect, regardless of the status of Setting 56. Cylindrical mapping will also be turned off automatically whenever the G-code program ends, but only if Setting 56 is ON. If Setting 56 is OFF, any cylindrical mapping in effect when the program ends will remain in effect.
- 4)** A G107 code issued without any parameters turns off cylindrical mapping. This should normally be done at the end of the program. However, if cylindrical mapping is not turned off and Setting 56 is OFF, then the mapping parameters established can be used by other programs.
- 5)** Re-issuing a G107 command with either the same or a different linear axis specification will turn off the cylindrical mapping currently in effect before new mapping is initiated.
- 6)** Issuing a G107 command with only a rotary axis specification and either a Q or an R will only change the default diameter or radius for the specified rotary axis; it will not turn cylindrical mapping on or off.

---

**NOTE:** Note: Due to the fact that the maximum acceleration of a rotary axis (as parameterized in steps/second/second on the Parameters page) is different from the maximum acceleration of a linear axis, there is a maximum achievable feedrate that can be programmed for a circular interpolated move when one of the rotary axes has been cylindrically mapped. Refer to the next section for a description of feedrate alteration.

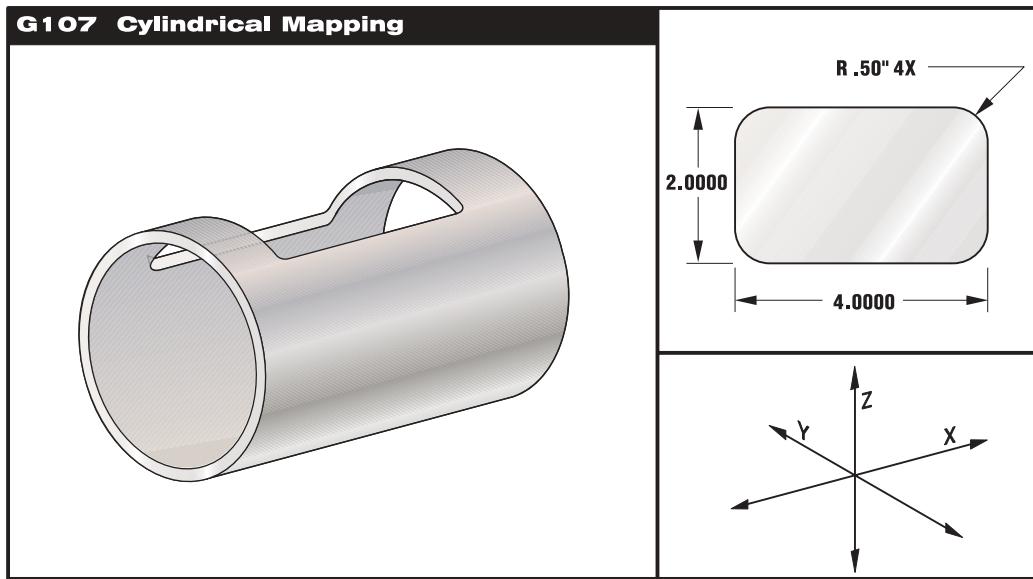
### **Feedrate Alteration To Ensure Concurrent Axis Motion**

The actual feedrate achieved in any interpolated motion on the CNC Mill is limited by the maximum speed achievable in each axis of motion. The feedrate may be lowered from its programmed value if that value would cause the Mill to command an axis to move faster than its parameterized maximum speed.

This means that the actual feedrate achieved may be less than or equal to the programmed feedrate, ensuring that all axes of interpolated motion move concurrently. When this condition occurs, the warning "LIM" will be displayed on the screen immediately to the left of the "FEED" display. Note also that the MAX FEEDRATE EXCEEDED alarm will still be issued if the programmed feedrate exceeds parameter 59, MAX FEEDRATE (currently 500 ipm), but only if the mill is not in Inverse Time mode.



VS-Series



### SAMPLE PART PROGRAM

```
%  
O0079 (G107 TEST)  
G00 G40 G49 G80 G90  
G28 G91 A0  
G90  
G00 G54 X1.75 Y0 S5000 M03  
G107 A0 Y0 R2. (IF NO R OR Q VALUE, MACHINE WILL USE VALUE IN SETTING 34)  
G43 H01 Z0.25  
G01 Z-0.25 F25.  
G41 D01 X2. Y0.5  
G03 X1.5 Y1. R0.5  
G01 X-1.5  
G03 X-2. Y0.5 R0.5  
G01 Y-0.5  
G03 X-1.5 Y-1. R0.5  
G01 X1.5  
G03 X2. Y-0.5 R0.5  
G01 Y0.  
G40 X1.75  
G00 Z0.25  
M09  
M05  
G91 G28 Z0.  
G28 Y0.  
G90  
G107  
M30  
%
```

**MORE WORK COORDINATE SELECTION (G110-G129)**
**G110-G129 Coordinate System #7-26**
**Group 12**

These codes select one of the additional 20 user coordinate systems stored within the Offsets memory. All subsequent references to axis positions will be interpreted in the new coordinate system. Operation of G110 to G129 is the same as G54 to G59.

**G154 Select Work Coordinates P1-P99**
**Group 12**

This feature provides 99 additional work offsets. G154 with a P value from 1 to 99 will activate the additional work offsets. For example G154 P10 will select work offset 10 from the list of additional work offset. Note that G110 to G129 refer to the same work offsets as the G154 P1 through P20; they can be selected by using either method. The Work Offset display screens have been adjusted accordingly. When a G154 work offset is active, the heading in the upper right work offset will show the G154 P value.

**G154 work offsets format**

```
#14001-#14006 G154 P1 (also #7001-#7006 and G110)
#14021-#14026 G154 P2 (also #7021-#7026 and G111)
#14041-#14046 G154 P3 (also #7041-#7046 and G112)
#14061-#14066 G154 P4 (also #7061-#7066 and G113)
#14081-#14086 G154 P5 (also #7081-#7086 and G114)
#14101-#14106 G154 P6 (also #7101-#7106 and G115)
#14121-#14126 G154 P7 (also #7121-#7126 and G116)
#14141-#14146 G154 P8 (also #7141-#7146 and G117)
#14161-#14166 G154 P9 (also #7161-#7166 and G118)
#14181-#14186 G154 P10 (also #7181-#7186 and G119)
#14201-#14206 G154 P11 (also #7201-#7206 and G120)
#14221-#14221 G154 P12 (also #7221-#7226 and G121)
#14241-#14246 G154 P13 (also #7241-#7246 and G122)
#14261-#14266 G154 P14 (also #7261-#7266 and G123)
#14281-#14286 G154 P15 (also #7281-#7286 and G124)
#14301-#14306 G154 P16 (also #7301-#7306 and G125)
#14321-#14326 G154 P17 (also #7321-#7326 and G126)
#14341-#14346 G154 P18 (also #7341-#7346 and G127)
#14361-#14366 G154 P19 (also #7361-#7366 and G128)
#14381-#14386 G154 P20 (also #7381-#7386 and G129)
#14401-#14406 G154 P21
#14421-#14426 G154 P22
#14441-#14446 G154 P23
#14461-#14466 G154 P24
#14481-#14486 G154 P25
#14501-#14506 G154 P26
#14521-#14526 G154 P27
#14541-#14546 G154 P28
#14561-#14566 G154 P29
#14581-#14586 G154 P30
#14781-#14786 G154 P40
#14981-#14986 G154 P50
#15181-#15186 G154 P60
#15381-#15386 G154 P70
#15581-#15586 G154 P80
#15781-#15786 G154 P90
#15881-#15886 G154 P95
#15901-#15906 G154 P96
#15921-#15926 G154 P97
#15941-#15946 G154 P98
#15961-#15966 G154 P99
```



VS-Series

## COMPENSATION (G141, G143)

### G141 3D+ Cutter Compensation

Group 07

This feature performs 3D+ cutter diameter compensation. The form is:

G141 Xnnn Ynnn Znnn Dnnn Innn Jnnn Knnn

Subsequent lines can be of the form:

G01 Fnnn Xnnn Ynnn Znnn Innn Jnnn Knnn

or:

G00 Xnnn Ynnn Znnn Innn Jnnn Knnn

The 3D+ G141 cutter compensation command is not just for 5-axis work. Any CAD system can output the I, J, K values to shift the tool by the amount in the Offsets memory of the control, even if the motions are only in 2 or 3 axes. In the Haas control, only G00 and G01 can use G141 cutter compensation; no other functions or canned cycles will get the offset. G91 incremental motion also cannot be used. G141 is used to indicate, without any doubt, what type of compensation is being requested. G141 is modal with G40, G41, and G42. Inverse time is usually used with this type of motion but is not required. The Dnn code selects which radius or diameter offset to use. The I, J, and K values give the direction in which cutter compensation is to be applied. When G141 is active, the tool diameter will have a vector component that will be added to commanded motion of the X, Y, or Z axis, based on the direction defined by I, J, and K. For example:

T1 M06

G00 G90 G54 X0 Y0 Z0 A0 B0

G141 D01 X0.Y0.Z0. (RAPID POSIT WITH 3 AX C COMP)

G01 G93 X.01 Y.01 Z.01 I.1 J.2 K.9747 F300. (FEED INV TIME)

X.02 Y.03 Z.04 I.15 J.25 K.9566 F300.

X.02 Y.055 Z.064 I.2 J.3 K.9327 F300

.

.

.

X2.345 Y.1234 Z-1.234 I.25 J.35 K.9028 F200. (LAST MOTION)

G94 F50. (CANCEL G93)

G0 G90 G40 Z0 (RAPID TO ZERO, CANCEL 3 AXIS C COMP)

X0 Y0

M30

G40 will cancel G141 3D+ cutter compensation.

---

**NOTE:** G141 is a group 7 G code, G40 cancels G141, G91 is not compatible with G141,  
G141 uses a D code

### G143 5-Axis Tool Length Compensation +

Group 08

(This G-code is optional; it is only used on the bridge mill and VR-11)

This G code allows the user to correct for variations in the length of cutting tools without the need to revert to CAD/CAM or post-processing steps. 5-axis length compensation applies only to machines on which all rotary motion is movement of the cutting tool. It does not apply to machines where any rotary axis motion involves the part or fixture. An H code is required to select the tool length from the existing length compensation tables. A G49 or H00 command will cancel 5-axis compensation. If only normal (Z axis) tool length compensation is desired, simply program G43 or G44. For G143 to work correctly, there must be two rotary axes, A and B. G90, absolute positioning mode, must be active and G91 must not be used. Work position 0,0 for the A and B axes must be such that the tool is parallel with Z axis motion.

G143 tool length compensation works only with rapid (G00) and linear feed (G01) motions. No other feed functions (G02 or G03) or canned cycles (drilling, tapping, etc.) can be used. When G143 is active, tool length compensation has a vector component based on the work coordinates of the A and B axes. This vector component may affect commanded motion of the X, Y or Z axis. For a positive tool length, the Z axis would move upward (in the + direction). If one of X, Y or Z is not programmed, there will be no motion of that axis, even if the motion of A or B produces a new tool length vector. Thus a typical program would use all 5 axes on one block of data.

Feedrate in G143 is complicated by the vector offset added to XYZ motion; thus inverse feed mode (G93) is strongly recommended. If the inverse feedrate was correct before applying G143, it will still be correct with any tool length compensation unless the maximum speed of an axis is exceeded. An example follows:

```

T1 M06
G00 G90 G54 X0 Y0 Z0 A0 B0
G143 H01 X0. Y0. Z0. A-20. B-20. (RAPID POSIT W. 5AX COMP)
G01 G93 X.01 Y.01 Z.01 A-19.9 B-19.9 F300. (FEED INV TIME)
X0.02 Y0.03 Z0.04 A-19.7 B-19.7 F300.
X0.02 Y0.055 Z0.064 A-19.5 B-19.6 F300
X2.345 Y.1234 Z-1.234 A-4.127 B-12.32 F200. (LAST MOTION)
G94 F50. (CANCEL G93)
G0 G90 G49 Z0 (RAPID TO ZERO, CANCEL 5 AXS COMP)
X0 Y0
M30

```

5-Axis Tool Length Compensation is canceled by G49.

**NOTES:** The intention behind G143 is to compensate for the difference in tool length between the originally posted tool and a substitute tool. Using G143 allows you to run the program without having to repost a new tool length.

Helical motion is not supported.

G49 or H00 should be used to cancel the G143 and H code in effect.

Insert G143 into the program after the tool call and the corresponding H code. Enter the tool length compensation in the corresponding tool geometry column associated with the H code called out in the program. G143 and H codes are modal and remain in effect until the code is cancelled or changed by another code from the same group.

The absolute Z-axis work position on the display does not reflect the actual programmed position for the Z axis. This is due to compensation being applied to all of the axes and the control not updating the screen accordingly.

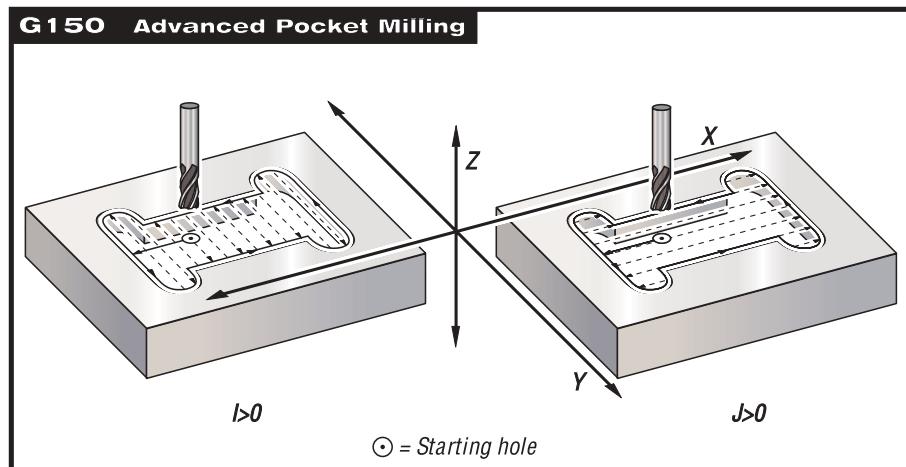


VS-Series

**GENERAL PURPOSE POCKET MILLING FUNCTION (G150)****G150 General Purpose Pocket Milling****Group 00**

- D Cutter size selection
- F Feedrate
- 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, positive values only ( $> 0$ )
- 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 and must be a positive number; this is the shift amount or cutter overlap. 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 finishing pass is of width **K**, and **K** must be a positive number. There is no finishing pass in the **Z** depth. The **R** value should be specified even if it is zero (R0). Otherwise the last value specified for **R** will be used.



Multiple passes over the area can be selected to control the depth of the cut. At least one pass is made over the pocket and multiple passes are made after feeding down by amount **Q** until the **Z** depth is reached. **Q** must be a positive number. 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 positioning. 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.

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

The first motion in the subroutine should move from this clear 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. In the example on the previous page, the start point of the G150 line is X3.25 Y4.5 and the first move of the subroutine is Y7.0. Therefore, the end of the subroutine must return to X3.25 Y7.0.

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

### **G150 EXAMPLES**

O0100	(G150 Pocket example)
G58 G00 G90 X3.25 Y4.5 S1200 M03	(Starting hole position)
T1 M06	(T1 cuts entry for endmill)
G83 R.1 Q0.5 Z-2. F20.	
T2 M06	(Endmill t2 cuts pocket 0.4 dia cutter, two passes to Z depth leave 0.01 for finish pass)
G150 G41 F15. D02 J0.35 K.01 Q0.5 R.1 X3.25 Y4.5 Z-2. P200	
G40 X3.25 Y4.5	
G28	
M30	
O0200	(G150 Pocket subroutine)
G01 Y7.	
X1.5	
G03 Y5.25 R0.875	
G01 Y2.25	
G03 Y0.5 R0.875	
G01 X5.	
G03 Y2.25 R0.875	
G01 Y5.25	
G03 Y7. R0.875	
G01 X3.25	
M99	(Return from subroutine)

### **G150 FORMAT EXAMPLE**

```
%  
O4500  
T1 M06  
G00 G90 G54 X0 Y0 S3500 M03  
G43 H01 Z.1 M08  
G150 X__ Y__ Z__ F__ R__ Q__ I__ OR J__ K__ P4600 D__ G41 OR G42  
G00 Z1.0 M09  
G28 G40 G91 Y0 Z0  
M30  
%  
%  
O4600  
G01 X__ Y__  
X__  
Y__  
X__ Y__  
M99  
%
```

---

**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.



VS-Series

## MORE G150 EXAMPLES

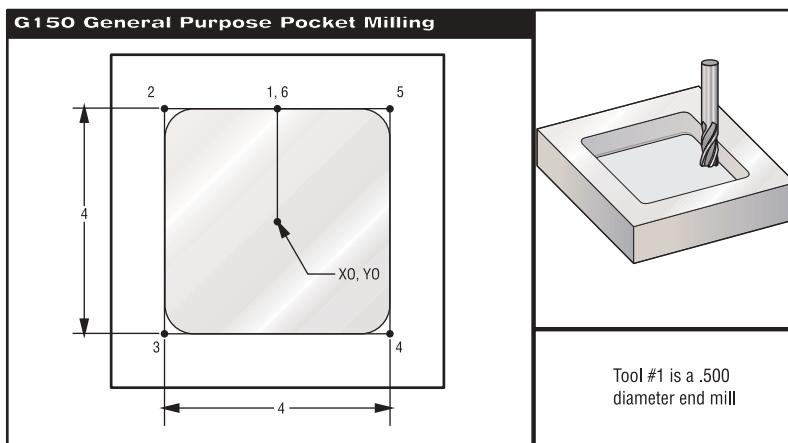
### ABSOLUTE SUBPROGRAM:

```
%  
O0500  
G01 Y2.0  
X-2.0  
Y-2.0  
X2.0  
Y2.0  
X0  
M99  
%
```

### INCREMENTAL SUBPROGRAM:

```
%  
O0500  
G01 G91 Y2.0  
X-2.0  
Y-4.0  
X4.0  
Y4.0  
G90  
M99  
%
```

### SQUARE POCKET:



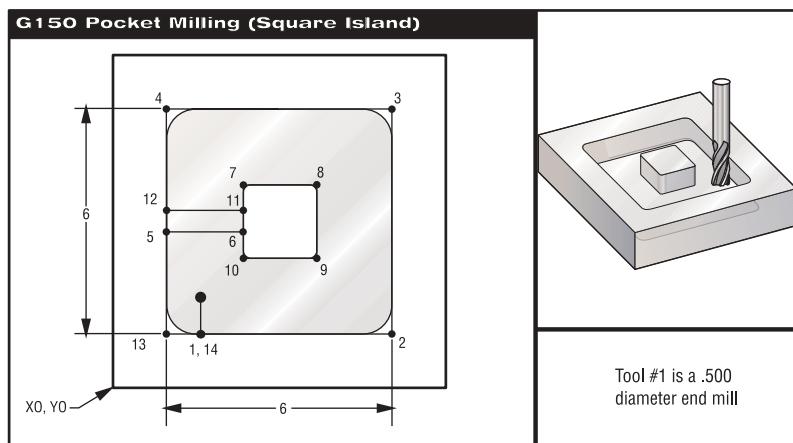
Pocket milling exercise for G150 operation.

### 4.0 x 4.0 x 0.500 DP. SQUARE POCKET:

```
%  
O1000  
T1 M06  
G90 G54 G00 X0 Y0  
S2000 M03  
G43 H01 Z 0.1 M08  
G01 Z0.01 F30.  
G150 P511 Z-0.5 Q0.25 R0.01 J0.3 K.01 G41 D01 F10.  
G40 G01 X0 Y0  
G00 Z1. M09  
G28 G91 Y0 Z0  
M30  
%  
%  
O00511  
G01 Y2.  
X-2.  
Y-2.  
X2.  
Y2.  
X0  
M99  
%
```

(Tool #1 is a 0.500" diameter endmill)

## SQUARE ISLAND:



*Square island programming exercise using G150.*

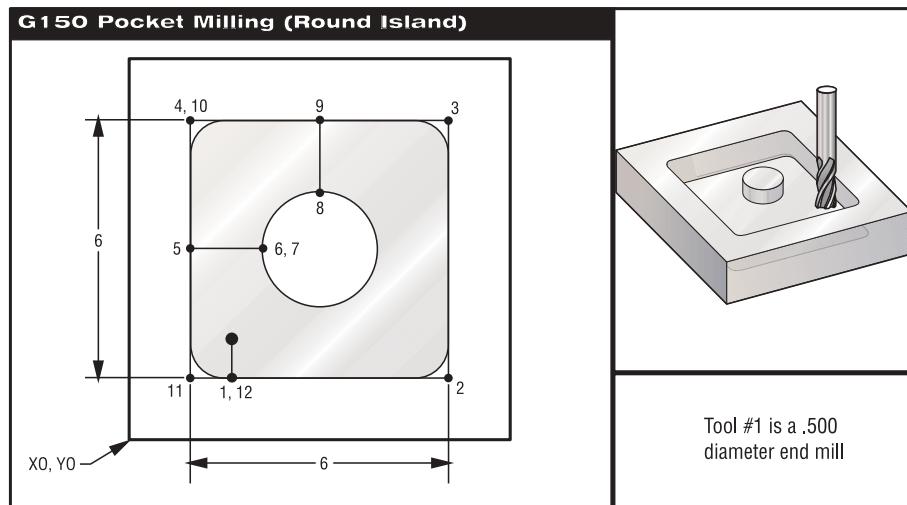
```
%  
O1000  
T1 M06  
G90 G54 G00 X2. Y2.  
S2500 M03  
G43 H01 Z0.1 M08  
G01 Z0.01 F30.  
G150 P500 X2. Y2. Z-0.5 Q0.5 R0.01 I0.3 K0.01 G41 D01 F10.  
G40 G01 X2. Y2.  
G00 Z1.0 M09  
G28 G91 Y0 Z0  
M30  
%  
%  
O0500  
G01 Y1.  
X7.  
Y7.  
X1.  
Y3.75  
X3.5  
Y4.5  
X4.5  
Y3.5  
X3.5  
Y4.25  
X1.  
Y1.  
X2.  
M99  
%
```

*(Tool is a 0.500" diameter endmill)*



VS-Series

## ROUND ISLAND:



*Round island programming exercise using G150.*

```
%  
O00188  
T1 M06  
G90 G54 G00 X2. Y2.  
S2500 M03  
G43 H01 Z0.1 M08  
G01 Z0 F30.  
G150 P531 X2. Y2. Z-0.5 Q0.5 R0.01 I0.3 K0.01 G41 D01 F10.
```

```
G40 G01 X2. Y2.  
G00 Z1. M09  
G28 G91 Y0 Z0  
M30  
%
```

```
%  
O00531  
G01 Y1.  
X7.  
Y7.  
X1.  
Y4.  
X3.  
G02 I1.  
X4. Y5. R1.  
G01 Y7.  
X1.  
Y1.  
X2.  
M99  
%
```

### 5-Axis Canned Cycles (Group 09)

G153 5-Axis High Speed Peck  
 G155 5-Axis Reverse Tap Cycle  
 G161 5-Axis Drill Cycle  
 G162 5-Axis Spot Drill  
 G163 5-Axis Normal Peck  
 G164 5-Axis Tapping Cycle  
 G165 5-Axis Bore In/Bore Out  
 G166 5-Axis Bore/Stop/Rapid  
 G169 5-Axis Bore/Dwell/Manual

These correspond to the existing G73, G74, G81, G82, G83, G84, G85, G86 and G89 3-axis canned cycles.

There are three differences between a 3-axis canned cycle and a 5-axis canned cycle:

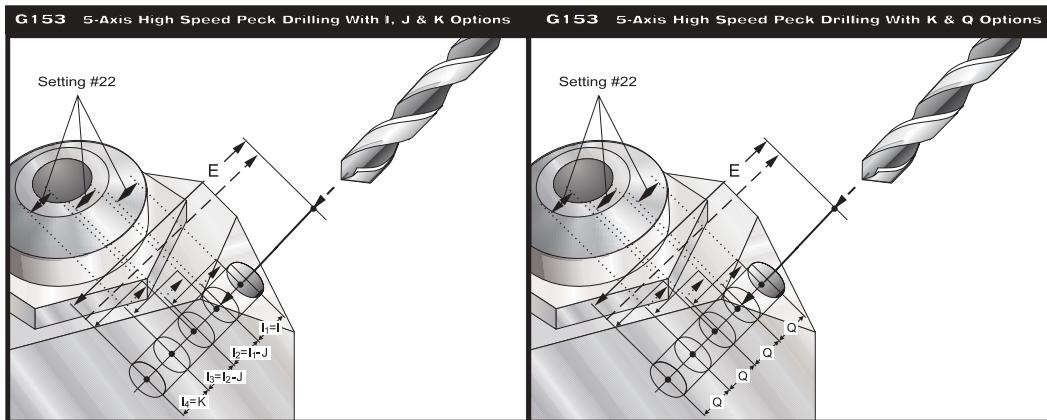
- 1) Five-axis canned cycles will rapid to the specified X, Y, Z, A, B position, which is used as the starting point.
- 2) Five-axis canned cycles require an **E** code, which specifies the distance from the starting position to the bottom of the hole. If this is not given, an alarm will be generated.
- 3) Five-axis canned cycles ignore any **R** codes. The starting point is always the **R** position.

Notes: Retraction and pecking, which would be rapid motions in a normal canned cycle, are fast feeds, not true rapids. Currently, G154 and G164 only perform floating taps. G184 is available for 5-axis rigid tapping.

#### G153 5-Axis High Speed Peck Drilling Canned Cycle

**Group 09**

E	Specifies the distance from the start position to the bottom of the hole
F	Feedrate 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 / number of pecks between retract
L	Number of repeats
P	Optional pause at end of last peck, in seconds
Q	The cut-in value, always incremental
A	A-axis tool starting position
B	B-axis tool starting position
X	X-axis tool starting position
Y	Y-axis tool starting position
Z	Z-axis tool starting position



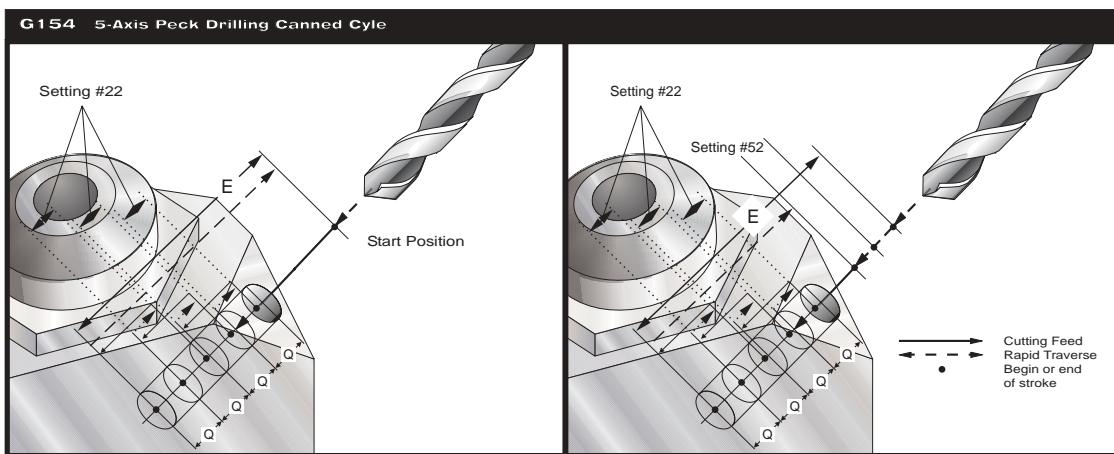


This G code is modal. Once activated, every commanded motion of **X** and/or **Y** will cause this canned cycle to be executed, until it is canceled or another canned cycle is selected. This is a high-speed peck cycle where the retract distance is set by Setting 22.

A specific X, Y, Z, A, B position must be programmed before the canned cycle is programmed. This position is used as the "Start position" by the canned cycle.

If **I**, **J**, and **K** are specified, a different operating mode is selected. The first pass will cut in by amount **I**, each succeeding cut will be reduced by amount **J**, and the minimum cutting depth is **K**. If **P** is specified, the tool will pause at the bottom of the hole after the last peck for that amount of time.

The same dwell time applies to all subsequent blocks that do not specify a dwell time. When the canned cycle is cancelled (by G00, G01, G80, or RESET) the dwell time will be reset to zero. This dwell cannot be used in the same block as an M97, M98, M99, or G65, because these codes use **P** for different purposes.



Setting 52 also changes the way G153 works when it returns to the **start position**. Most programmers set the **R** plane well above the cut to ensure that the chip-clearing motion actually allows the chips to get out of the hole, but this causes wasted motion when first drilling through this "empty" space. If Setting 52 is set to the distance required to clear chips, the **start position** can be put much closer to the part being drilled. When the chip-clearing move to the **start position** occurs, the **Z** axis will be moved above the **start position** by the amount given in this setting.

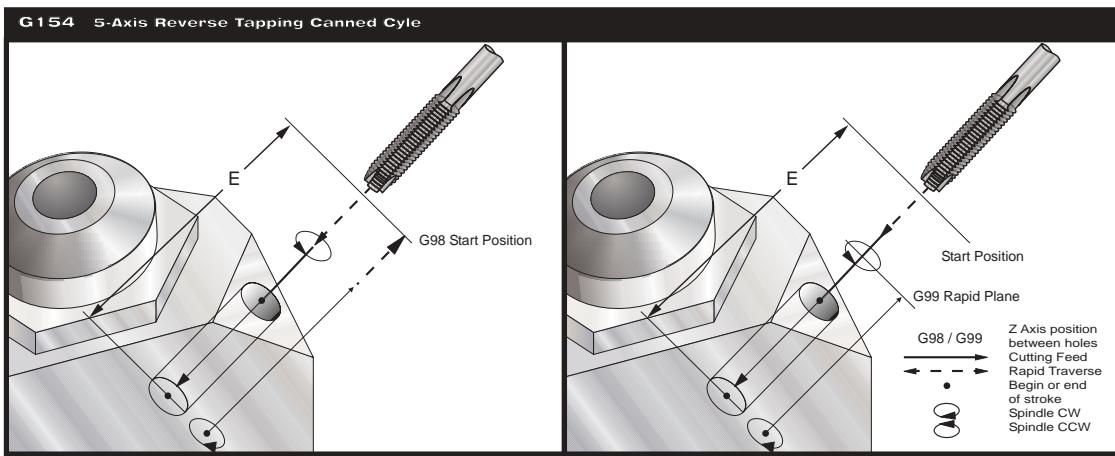
### G155 5-Axis Reverse Tap Canned Cycle

### Group 09

- E** Specifies the distance from the start position to the bottom of the hole
- F** Feedrate in inches (mm) per minute
- L** Number of repeats
- A** A-axis tool starting position
- B** B-axis tool starting position
- X** X-axis tool starting position
- Y** Y-axis tool starting position
- Z** Z-axis tool starting position

This G code is modal. Once activated, every commanded motion of **X** and/or **Y** will cause this canned cycle to be executed, until it is canceled or another canned cycle is selected. A specific X, Y, Z, A, B position must be programmed before the canned cycle is programmed. This position is used as the "Start position" by the canned cycle.

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



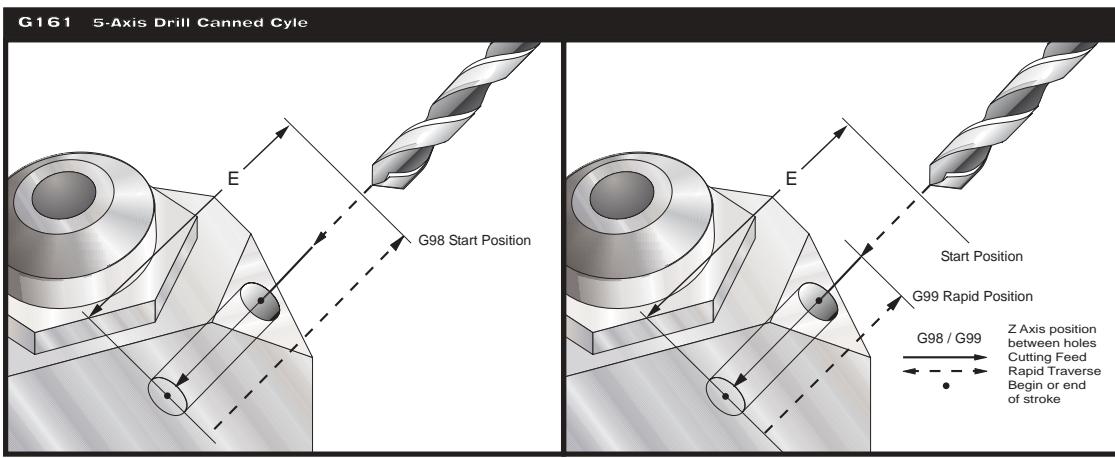
If **K** and **Q** are both specified, a different operating mode is selected for this canned cycle. In this mode, the tool is returned to the **start position** after the number of passes totals up to amount **K**. This allows much faster drilling than G83 but still returns to the **start position**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

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

### G161 5-Axis Drill Canned Cycle

### Group 09

- E Specifies the distance from the start position to the bottom of the hole
- F Feedrate in inches (mm) per minute
- L Number of repeats
- A A-axis tool starting position
- B B-axis tool starting position
- X X-axis tool starting position
- Y Y-axis tool starting position
- Z Z-axis tool starting position



This G code is modal. Once activated, every commanded motion of **X** and/or **Y** will cause this canned cycle to be executed, until it is canceled or another canned cycle is selected. A specific **X, Y, Z, A, B** position must be programmed before the canned cycle is programmed. This position is used as the "Start position" by the canned cycle.

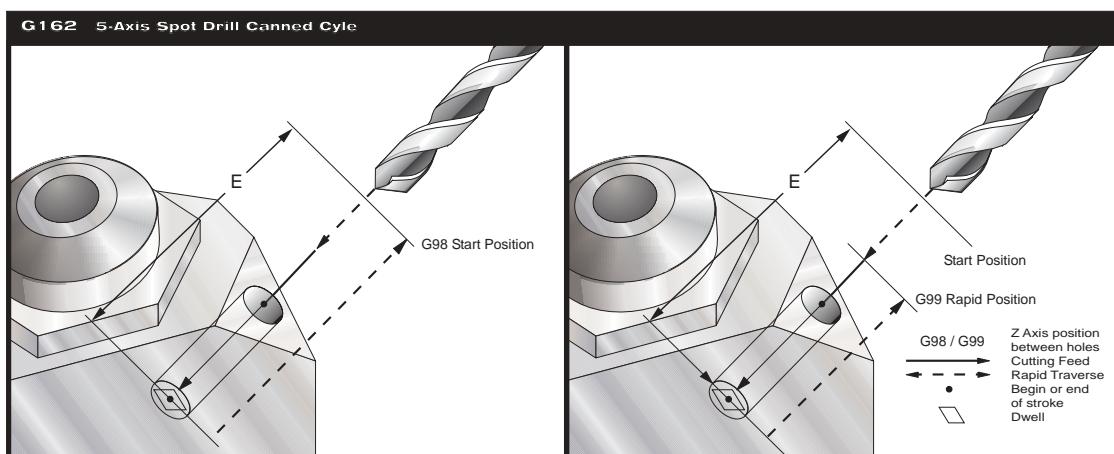


VS-Series

**G162 5-Axis Spot Drill Canned Cycle****Group 09**

- E Specifies the distance from the start position to the bottom of the hole
- F Feedrate in inches (mm) per minute
- L Number of repeats
- P The dwell time at the bottom of the hole
- A A-axis tool starting position
- B B-axis tool starting position
- X X-axis tool starting position
- Y Y-axis tool starting position
- Z Z-axis tool starting position

This G code is modal. Once activated, every commanded motion of **X** and/or **Y** will cause this canned cycle to be executed, until it is canceled or another canned cycle is selected. A specific **X**, **Y**, **Z**, **A**, **B** position must be programmed before the canned cycle is programmed. This position is used as the "Start position" by the canned cycle.

**G163 5-Axis Normal Peck Drilling Canned Cycle****Group 09**

- E Specifies the distance from the start position to the bottom of the hole
- F Feedrate 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
- L Number of repeats
- P Optional pause at end of last peck, in seconds
- Q The cut-in value, always incremental
- A A-axis tool starting position
- B B-axis tool starting position
- X X-axis tool starting position
- Y Y-axis tool starting position
- Z Z-axis tool starting position

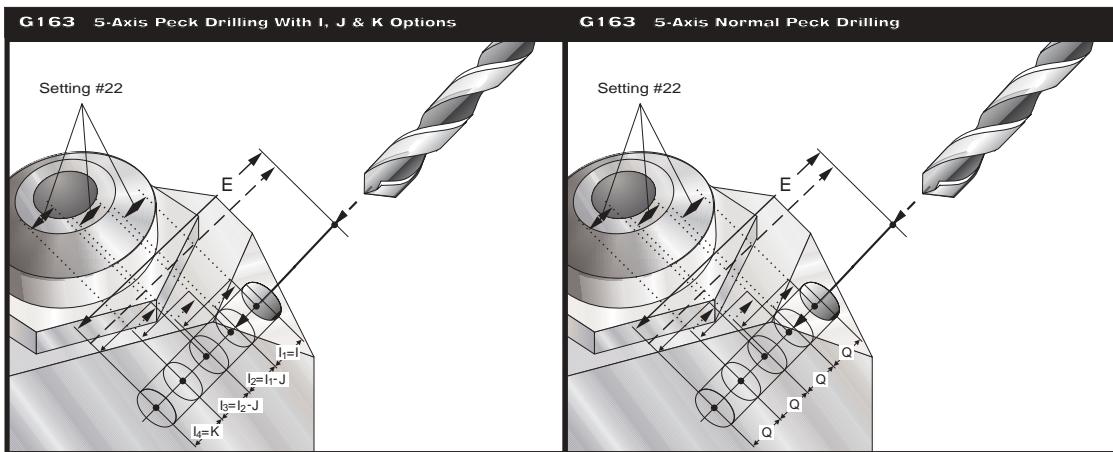
This G code is modal. Once activated, every commanded motion of **X** and/or **Y** will cause this canned cycle to be executed, until it is canceled or another canned cycle is selected. A specific **X**, **Y**, **Z**, **A**, **B** position must be programmed before the canned cycle is programmed. This position is used as the "Start position" by the canned cycle.

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

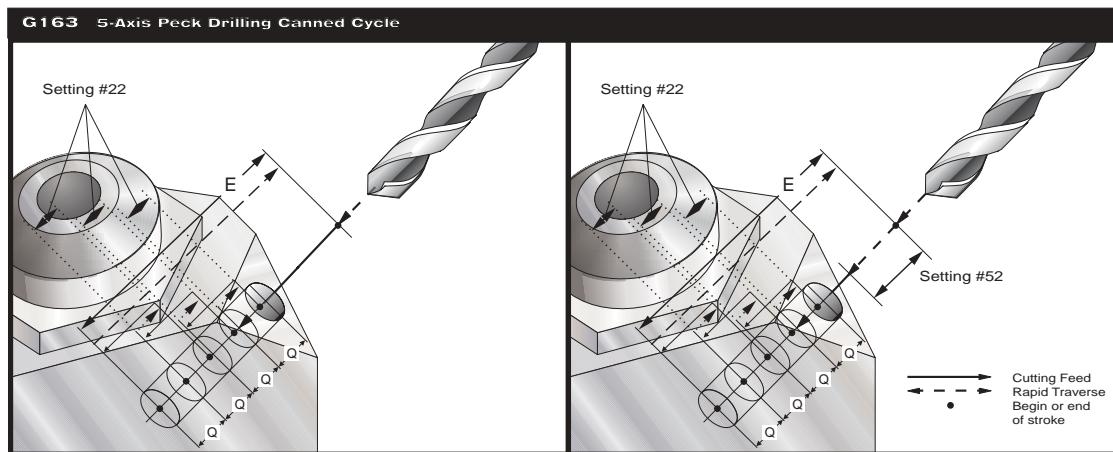
If **P** is specified, the tool will pause at the bottom of the hole after the last peck for that amount of time. The following example will peck several times and dwell for one and a half seconds at the end:

G163 Z-0.62 F15. R0.1 Q0.175 P1.5.

The same dwell time applies to all subsequent blocks that do not specify a dwell time. When the canned cycle is cancelled (by G00, G01, G80, or RESET) the dwell time will be reset to zero. This dwell cannot be used in the same block as an M97, M98, M99, or G65, because these codes use **P** for different purposes.



Setting 52 also changes the way G163 works when it returns to the **start position**. Most programmers set the **R** plane well above the cut to ensure that the chip-clearing motion actually allows the chips to get out of the hole, but this causes wasted motion when first drilling through this "empty" space. If Setting 52 is set to the distance required to clear chips, the **start position** can be put much closer to the part being drilled. When the chip-clearing move to the **start position** occurs, the **Z** axis will be moved above the **start position** by the amount given in this setting.

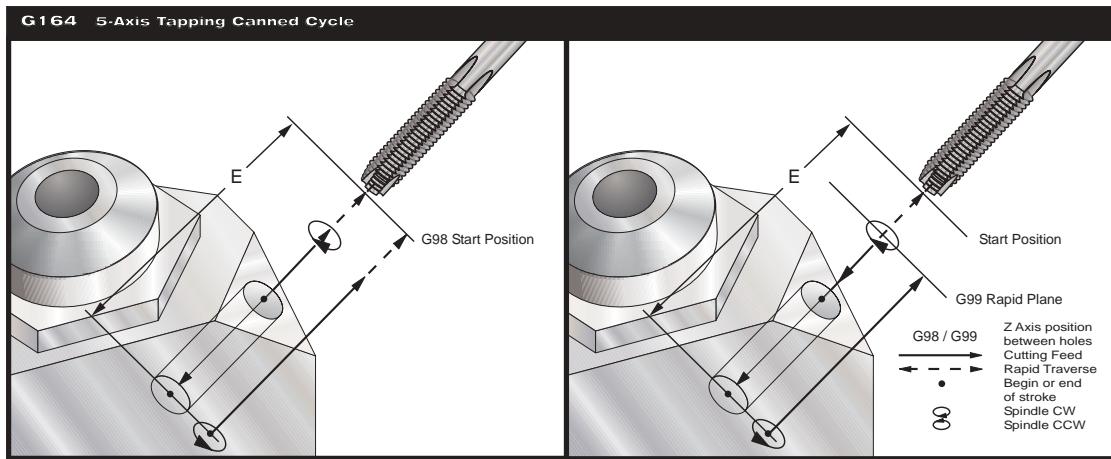




VS-Series

**G164 5-Axis Tapping Canned Cycle****Group 09**

- E Specifies the distance from the start position to the bottom of the hole
- F Feedrate in inches (mm) per minute
- L Number of repeats
- A A-axis tool starting position
- B B-axis tool starting position
- X X-axis tool starting position
- Y Y-axis tool starting position
- Z Z-axis tool starting position

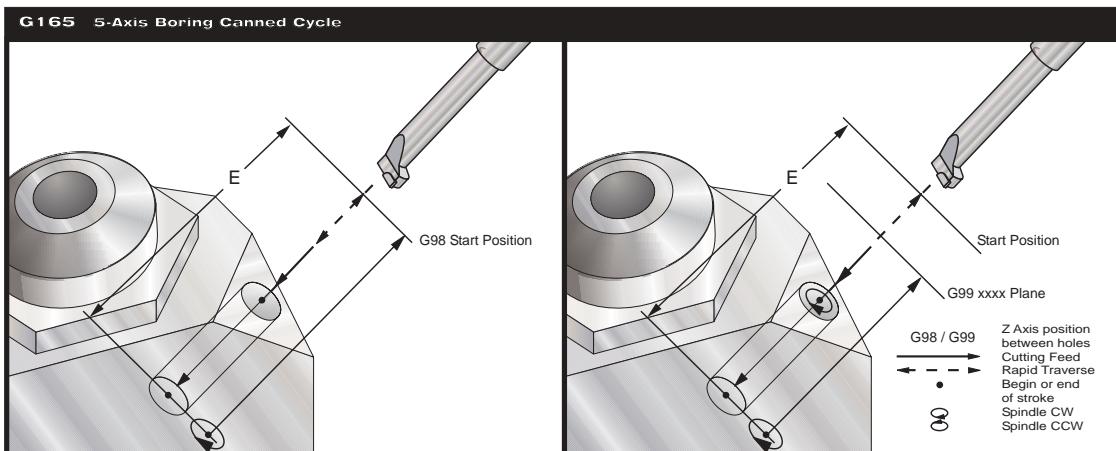


This G code is modal. Once activated, every commanded motion of **X** and/or **Y** will cause this canned cycle to be executed, until it is canceled or another canned cycle is selected. A specific **X**, **Y**, **Z**, **A**, **B** position must be programmed before the canned cycle is programmed. This position is used as the "Start position" by the canned cycle.

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

**G165 5-Axis Boring Canned Cycle**
**Group 09**

- E Specifies the distance from the start position to the bottom of the hole
- F Feedrate in inches (mm) per minute
- L Number of repeats
- A A-axis tool starting position
- B B-axis tool starting position
- X X-axis tool starting position
- Y Y-axis tool starting position
- Z Z-axis tool starting position



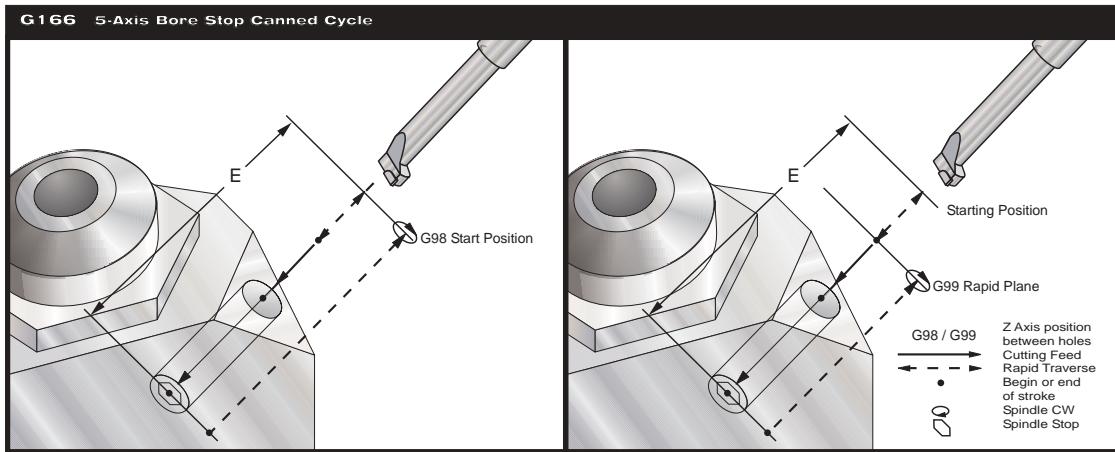
This G code is modal. Once activated, every commanded motion of **X** and/or **Y** will cause this canned cycle to be executed, until it is canceled or another canned cycle is selected. A specific **X**, **Y**, **Z**, **A**, **B** position must be programmed before the canned cycle is programmed. This position is used as the "Start position" by the canned cycle.



VS-Series

**G166 5-Axis Bore and Stop Canned Cycle****Group 09**

- E Specifies the distance from the start position to the bottom of the hole
- F Feedrate in inches (mm) per minute
- L Number of repeats
- A A-axis tool starting position
- B B-axis tool starting position
- X X-axis tool starting position
- Y Y-axis tool starting position
- Z Z-axis tool starting position

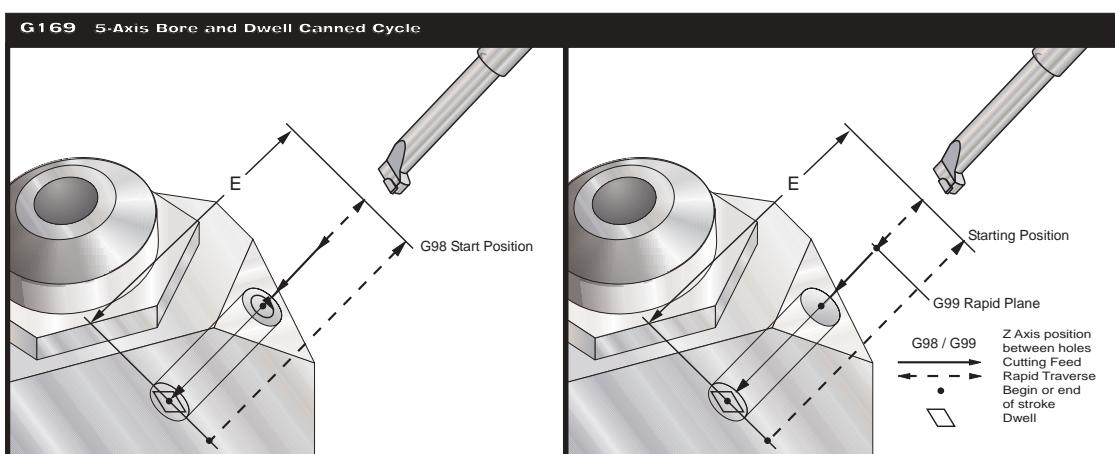


This G code is modal. Once activated, every commanded motion of **X** and/or **Y** will cause this canned cycle to be executed, until it is canceled or another canned cycle is selected. A specific **X**, **Y**, **Z**, **A**, **B** position must be programmed before the canned cycle is programmed. This position is used as the "Start position" by the canned cycle.

**G169 5-Axis Bore and Dwell Canned Cycle**
**Group 09**

- E Specifies the distance from the start position to the bottom of the hole
- F Feedrate in inches (mm) per minute
- L Number of repeats
- P The dwell time at the bottom of the hole
- A A-axis tool starting position
- B B-axis tool starting position
- X X-axis tool starting position
- Y Y-axis tool starting position
- Z Z-axis tool starting position

This G code is modal. Once activated, every commanded motion of **X** and/or **Y** will cause this canned cycle to be executed, until it is canceled or another canned cycle is selected. A specific **X**, **Y**, **Z**, **A**, **B** position must be programmed before the canned cycle is programmed. This position is used as the "Start position" by the canned cycle.


**RIGID TAPPING**
**G174 CCW Non-Vertical Rigid Tap**  
**G184 CW Non-Vertical Rigid Tap**
**Group 00**  
**Group 00**

- F Feedrate in inches per minute
- X Optional X position at bottom of hole
- Y Optional Y position at bottom of hole
- Z Optional Z position at bottom of hole

This G code is used to perform rigid tapping for non-vertical holes. It may be used with a right-angle head to perform rigid tapping in the **X** or **Y** axis on a three-axis mill, or to perform rigid tapping along an arbitrary vector with a five-axis mill. When performing a five-axis rigid tap, the machinist must ensure that the head is positioned correctly before the G184 command is given. If the head is not aligned with the direction of motion, the tool will break. The ratio between the feedrate and spindle speed must be precisely the thread pitch being cut. Because only the end point of the tap is specified with this canned cycle, it is not modal. A motion to the next starting position will normally occur between G174/G184 canned cycles. G184 commands clockwise motion of the spindle during entry, and G174 commands counterclockwise motion. You do not have to start the spindle before this canned cycle.



VS-Series

## *IN POSITION ACCURACY (G 187)*

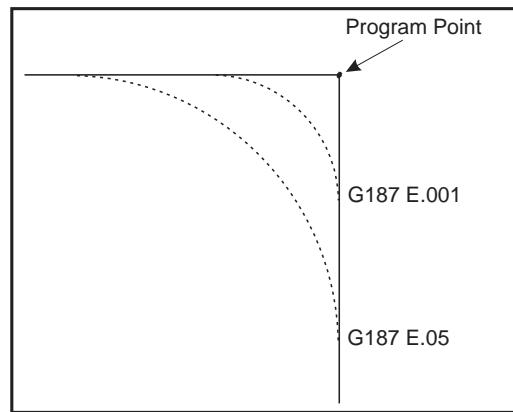
### **G187 Accuracy Control**

### **Group 00**

The G187 code is used to select the accuracy with which corners are machined. The format for using G187 is **G187 Ennnn**, where **nnnn** is the desired accuracy. Refer to Setting 85 "Max Corner Rounding" for more information. The initial value of G187 is Setting 85.

Program G187 as follows:

G187 E0.01                                          (to set value)  
G187                                                      (to revert to Setting 85 value)



## 5. M CODES (MISCELLANEOUS FUNCTIONS)

### M CODE SUMMARY

Only one **M** code may be programmed per block. All **M** codes take effect or cause an action to occur at the end of the block. However, when Parameter 278, bit "CNCR SPINDLE," is set to 1, an M03 (Spindle On) will occur at the beginning of a block.

M00	Stop Program
M01	Optional Program Stop
M02	Program End
M03	Spindle Forward (CW)
M04	Spindle Reverse (CCW)
M05	Spindle Stop
M06	Tool Change
M08	Coolant On
M09	Coolant Off
M10	Engage 4th Axis Brake
M11	Release 4th Axis Brake
M12	Engage 5th Axis Brake
M13	Release 5th Axis Brake
M16	Tool Change (same as M06)
M17	Unclamp APC Pallet And Open APC Door (used for maintenance only)
M18	Clamp APC Pallet And Close APC Door (used for maintenance only)
M19	Orient Spindle (P and R values optional)
M21-M28	Optional Pulsed User M Function with Fin
M30	Prog End and Rewind
M31	Chip Conveyor Forward
M33	Chip Conveyor Stop
M34	Coolant Spigot Position Down, Increment
M35	Coolant Spigot Position Up, Decrement
M36	Pallet Part Ready
M39	Rotate Tool Turret
M41	Low Gear Override
M42	High Gear Override
M50	Execute Pallet Change
M51-M58	Set Optional User M Code
M59	Set Output Relay
M61-M68	Clear Optional User M Code
M69	Clear Output Relay
M75	Set G35 or G136 Reference Point
M76	Control Display Inactive
M77	Control Display Active
M78	Alarm If Skip Signal Found
M79	Alarm If Skip Signal Not Found
M80	Auto Door Open
M81	Auto Door Close
M82	Tool Unclamp
M83	Auto Air Gun On
M84	Auto Air Gun Off
M86	Tool Clamp
M88	Through-the-Spindle Coolant ON
M89	Through-the-Spindle Coolant OFF



M93	Start Axis Position Capture
M94	Stop Axis Position Capture
M95	Sleep Mode
M96	Jump if No Input
M97	Local Sub-Program Call
M98	Sub-Program Call
M99	Sub-Program Return or Loop
M109	Interactive User Input

### **M00 Stop Program**

The M00 code is used to stop a program. It stops the spindle, turns off the coolant and stops interpretation look-ahead processing. The program pointer will advance to the next block and stop. Pressing **CYCLE START** will continue program operation from the next block. If the through-spindle coolant option is On, M00 will shut it off.

### **M01 Optional Program Stop**

The M01 code is identical to M00 except that it only stops if OPTIONAL STOP is turned on from the front panel. **CYCLE START** will continue program operation from the next block. If the through-spindle coolant option is On, M01 will shut it off.

### **M02 Program End**

The M02 code will stop program operation the same as M00, but does not advance the program pointer to the next block.

### **M03 Spindle Forward (CW)**

The M03 code will start the spindle moving in a clockwise direction at whatever speed was previously set. Block activation will be delayed until the spindle reaches about 90% of commanded speed.

When Parameter 278 bit 10, "CNCR SPINDLE," is set to 1, block activation does not depend on the spindle coming up to speed. Interpolated motion will not be initiated until spindle speed is reached, but other commands, such as rapid motion, will be.

### **M04 Spindle Reverse (CCW)**

The M04 code will start the spindle moving in a counterclockwise direction at whatever speed was previously set. Block activation will be delayed until the spindle reaches about 90% of commanded speed.

When Parameter 278 bit 10, "CNCR SPINDLE," is set to 1, block activation does not depend on the spindle coming up to speed. Interpolated motion will not be initiated until spindle speed is reached, but other commands, such as rapid motion, will be.

Running an M04 command with through-spindle coolant on is not recommended.

### **M05 Spindle Stop**

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

### **M06 Tool Change**

The M06 code is used to initiate a tool change. If the spindle is running, it will be stopped. No axis commands are required prior to the tool change unless there is a problem with tool/part/fixtures clearance. The Z axis will automatically move up to the machine zero position and the selected tool (**Tnn**) will be put into the spindle. The **Tnn** must be in the same block as the M06 or in a previous block. The Z axis will remain at machine zero. The spindle will not start again after the tool change until it is commanded, but the **Snnnn** speed and gear will be unchanged. The coolant pump will be turned off during a tool change.

When the through-spindle coolant (TSC) is On, M06 will orient the spindle and move the Z axis to tool change position, turn off the TSC pump, purge the coolant from the drawbar, then perform a tool change. TSC will remain Off until an M88 is called.

**M08 Coolant On**

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

**M09 Coolant Off**

The M09 code will turn off the coolant supply.

**M10 Engage 4th Axis Brake**

The M10 code is used to apply the brake to the 4th axis. The brake is normally engaged, so M10 is only required when M11 has been used to release the brake.

M11 activates a relay that releases the brake. M10 deactivates this relay, engaging the brake.

**M11 Release 4th Axis Brake**

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

**M12 Engage 5th Axis Brake**

The M12 code is used to apply the brake to the 5th axis. The brake is normally engaged, so M12 is only required when M13 has been used to release the brake.

M13 activates a relay that releases the brake. M12 deactivates this relay, engaging the brake.

**M13 Release 5th Axis Brake**

The M13 code will “pre-release” the 5th axis brake. This is useful in preventing the delay that otherwise occurs when a 5th axis is used with a brake and motion is commanded in that axis. It is not required, but without a prior M13, motion will be delayed 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.

**M17 Unclamp APC Pallet and Open APC Door**

Used for maintenance only.

**M18 Clamp APC Pallet and Close APC Door**

Used for maintenance only.

**M19 Orient Spindle (P and R values optional)**

The M19 code is used to electronically orient the spindle to a fixed position. An optional P value can be added that will specify the angle of orientation (in degrees). For example, M19 P270 will orient the spindle to 270 degrees. An optional R value allows the operator to specify up to four decimal places. An M19 R123.4567 will position the spindle to the angle specified by the R value.

**M21-M28 Optional User M Function with M-Fin**

The M21 through M28 codes are optional for user interfaces. They will activate one of the relays (see Parameter 352), wait for the M-fin signal (discrete input 1009), 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 (including TSC). The program pointer will be reset to the first block of the program. The parts counters displayed on the Current Commands display incremented. M30 cancels tool length offsets.

**M31 Chip Conveyor Forward**

M31 starts the chip conveyor motor in the forward direction, defined as the direction that the conveyor must move to transport chips out of the work cell. The conveyor will not turn if the door is open. This may be overridden by setting bit 17 of Parameter 209 (CNVY DR OVRD) to 1.

**M33 Chip Conveyor Stop**

M33 Stops Conveyor motion.

**M34 Coolant Spigot Position Down, Increment**

M34 moves the spigot by one increment, advancing it one place from the home position. The home position is designated as zero. If the current home position is designated as 5 and M34 is executed, then the spigot will advance to position 6. The spigot home position places it at the most positive Z axis location the spigot can attain. An increment lowers the spigot and thus the coolant stream direction.

**M35 Coolant Spigot Position Up, Decrement**

M35 moves the coolant spigot one decrement, that is, one place toward the spigot home position. The home position is designated as zero. If the current spigot position is 5 and M35 is executed, then the spigot will move to position 4. The spigot home position places the spigot at the most positive Z axis location. A decrement will raise the spigot and thus the coolant stream direction.

**M36 Pallet Part Ready**

**Used on machines with pallet changers.** This M code causes the PART READY button on the switch box to flash on and off. It delays the pallet change until the PART READY button is depressed.

**M39 Rotate Tool Turret**

The M39 code is used to rotate the tool turret without performing a tool change. The desired tool pocket number (**Tn**) must be programmed previous to the M39.

This M code may be useful for moving an empty pocket to face the spindle. This is not normally required but is useful for diagnostic purposes or to recover from a tool changer crash. Remember that the pocket facing the spindle must always be empty for a tool change.

**M41 Low Gear Override**

The M41 code is used to override the spindle gear implied by the **Snnn** command. With M41, the spindle gear will always be low. If the speed commanded is above the low gear limit, the spindle speed will be the low gear limit. This **M** code does not turn the spindle on or off. If the spindle was turning before this command, it will be started again. If it was stopped before this command it will be left off. M41 is ignored if there is no gear box.

**M42 High Gear Override**

The M42 code is used to override the spindle gear implied by the **Snnn** command. With M42, the spindle gear will always be high. Note that this may reduce the torque at the tool. This **M** code does not turn the spindle on or off. If the spindle was turning before this command, it will be started again. If it was stopped before this command, it will be left off. M42 is ignored if there is no gear box.

**M50 Execute Pallet Change**

Signals the calling program and executes a pallet change sequence.

**M51-M58 Set Optional User M Codes**

The M51 through M58 codes are optional for user interfaces. They will activate one of the relays (see Parameter 352) and leave it active. These are the same relays used for M21-M28. Use M61-M68 to turn these off. The RESET key will turn off all of these relays.

See the section on optional M functions for more information on additional user outputs.

## M59 Set Output Relay

This M code directly enables a discrete output relay. The syntax for its usage is **M59 Pnn**, where "nn" specifies the number of the relay being turned on. An M59 command can be used to turn on any of the discrete output relays in the range from 1100 to 1155. For example, **M59 P1103** does the same thing as **#1103=1**, except that it is processed as an M code (processed when coming out of the queue instead of when going into the queue). To turn off a relay, use M69.

## M61-M68 Clear Optional User M Codes

The M61 through M68 codes are optional for user interfaces. They will deactivate one relay (see Parameter 352). These are the same relays used for M21-M28.

## M69 Clear Output Relay

This M code directly disables a discrete output relay. The syntax for its usage is **M69 Pnn**, where "nn" specifies the number of the relay being turned off. An M69 command can be used to turn off any of the discrete output relays in the range from 1100 to 1155. For example, **M69 P1103** does the same thing as **#1103=0**, except that it is processed as an M code (processed when coming out of the queue instead of when going into the queue). To turn on a relay, use M59.

## M75 Set G35 or G136 Reference Point

This code is used to set the reference point for G35 and G136 commands. It must be used after a motion which is terminated with the skip function.

## M76 Control Display Inactive

This code is used to disable the updating of the screen displays. It is not necessary for machine performance.

## M77 Control Display Active

This code is used to enable the updating of the screen displays. It is only used when M76 has been used to disable the displays.

## M78 Alarm if Skip Signal Found

This code is used to generate an alarm if a programmed skip function actually got the skip signal. This is usually used when a skip signal is not expected, and may indicate a probe crash. This code can be placed in a block with the skip function or in any subsequent block. The skip functions are G31, G36, and G37.

## M79 Alarm if Skip Signal Not Found

This code is used to generate an alarm if a programmed skip function did not get the skip signal. This is usually done when the absence of the skip signal means a probe positioning error. This code can be placed in a block with the skip function or in any subsequent block. The skip functions are G31, G36, and G37.

## M80 / M81 Auto Door Open / Close

The M codes that control the Auto Door have been changed; they are now M80 to open and M81 to close the Auto Door. Note: Setting 51, Door Hold Override, must be set to ON; Parameter 57 bit 31, Door Stop SP, must be set to zero; and Setting 131, Auto Door, set to ON. An M80 will cause the door to open and an M81 will cause it to close. Also, the control will beep while the door is in motion.

## M82 Tool Unclamp

This code is used to release the tool from the spindle. It is not normally needed, as 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, as the TOOL WILL BE DROPPED from the spindle and MAY DAMAGE the tool, the machine, or your setup.

## M83 / M84 Auto Air Gun On / Off

An M83 will turn the Air Gun on, and an M84 will turn it off. Additionally, an **M83 Pnnn** (where nnn is in milliseconds) will turn it on for the specified time, then off automatically. Note: The Air Gun is assigned to discrete output #1138.

**M86 Tool Clamp**

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

**M88 Through-Spindle Coolant On**

This code is used to turn on the through-spindle coolant (TSC) option. When M88 is called, the spindle will stop, then the TSC pump will turn on, and then the spindle will restart.

**M89 Through-Spindle Coolant Off**

This code is used to turn off the through-spindle coolant (TSC) option. When M89 is called, the spindle will stop and the TSC pump will shut off. M89 also turns on a coolant line purge for the amount of time specified in Parameter 237, then turns off the purge.

**M93 / M94 Start Axis Position Capture / Stop Axis Position Capture**

These M codes permit the control to capture the position of an auxiliary axis when a discrete input goes high. The format is:

M93 Px Qx (where P is the axis number and Q is a discrete input number from 0 to 63) M94

M93 causes the control to watch the discrete input specified by the Q value, and when it goes high, captures the position of the axis specified by the P value. The position is then copied to hidden macro variable 749. M94 stops the capture. M93 and M94 were introduced to support the Horizontal Mill Bar Feeder.

**M95 Sleep Mode**

Sleep mode is essentially a long dwell. Sleep mode can be used when the user wishes the machine to begin warming itself up early in the morning and be ready for use upon the operator's arrival. The format of the M95 command is:

M95 (hh:mm)

The comment immediately following the M95 must contain the hours and minutes that the machine is to sleep for. For example, if the current time were 6 p.m. and the user wanted the machine to sleep until 6:30 a.m. the next day, the following command would be used:

M95 (12:30)

Up to 99 hours can be specified, thus allowing the machine to sleep for over 4 days. If the time is specified using an incorrect format, Alarm 324, Delay Time Range Error, will be generated. When the machine enters sleep mode, the following message is displayed:

Haas  
SLEEP MODE  
REMAINING TIME nnn MIN.

The message will be re-displayed in a different position on the screen each second, so that the user can see at a glance that the machine is sleeping. This has the advantage of preventing the message from being "burned" into one spot on the screen.

When less than 1 minute of sleep time remains, the message will change to:

REMAINING TIME nn SEC.

If the user presses any key or opens the door, sleep mode will be cancelled and the active program will wait at the block following the M95 until the user presses the **CYCLE START** key.

For the last 30 seconds of sleep time, the machine will beep and display an additional message:

WAKE UP IN nn SECONDS

When the sleep time has elapsed, the active program will continue at the block following M95.

**M96      Jump If No Input**

- P      Block to branch to when conditional test succeeds  
Q      Discrete input to test (0 to 63)

This code is used to test a discrete input for 0 (off) status. The Q value must be in the range 0 to 63, corresponding to the discrete inputs found on the Diagnostic display, with the upper left being input 0 and the lower right being input 63. Q is not required within the M96 block; the last specified Q will be used. When this block is executed and the input signal specified by Q has a value of 0, a branch to the block specified by P is performed. A **Pnnnn** code is required and must match a line number within the same program.

This command stops the look-ahead queue until the test is made at run time. Since the look-ahead queue is exhausted, M96 cannot be executed when cutter compensation is invoked. M96 cannot be executed from a main DNC program. If you wish to use M96 in DNC, it must be in a resident subroutine called from the DNC program.

The following is an M96 example:

```
N05 M96 P5 Q8    (Test input #5, Door Switch, until closed);  
N10               (Start of some program loop);  
.                 (Program that machines part);
```

```
N85 M21          (Execute an external user function)  
N90 M96 P10 Q27 (Loop to N10 if spare input [#27] is 0);  
N95 M30          (If spare input is 1 then end program);
```

**M97      Local Sub-Program Call**

This code is used to call a subroutine referenced by a line number **N** within the same program. A **Pnnnnnnnn** code is required and must match a line number within the same program. This is useful for simple subroutines within a program and does not require the complication of a separate program. The subroutine must still be ended with an M99. An **L** count in the M97 block will repeat the subroutine call that number of times.

**M98      Sub Program Call**

This code is used to call a subroutine. The **Pnnnn** code is the number of the program being called. The **Pnnnn** code must be in the same block. The program with the same number must already be loaded into memory, and it must contain an M99 to return to the main program. An **L** count can be put on the line containing the M98 and will cause the subroutine to be called **L** times before continuing to the next block.

**M99      Sub-Program Return or Loop**

This code is used to return to the main program from a subroutine or macro. It will also cause the main program to loop back to the beginning without stopping if it is used other than in a sub-program without a **P** code. If **M99 Pnnnn** is used, it will cause a jump to line **Nnnnn** containing the same number.

**M99 Pnnnn** in the Haas control varies from that seen in Fanuc-compatible controls, in which **M99 Pnnnn** will return to the calling program and resume execution at block **N** specified in **Pnnnn**. For the Haas control, **M99 Pnnnn** will NOT return to the calling program, but instead will jump to block **N** specified in **Pnnnn** in the current program.



You can simulate Fanuc behavior by using the following code.

calling program:	Haas	Fanuc
	O0001	O0001
	...	...
	N50 M98 P2	N50 M98 P2
	N51 M99 P100	...
	...	N100 (continue here)
	N100 (continue here)	...
	...	M30
subroutine:	M30	
	O0002	O0002
	M99	M99 P100

If you have macros, you can use a global variable and specify a block to jump to by adding **#nnn=ddd** in the subroutine and then using **M99 P#nnn** after the subroutine call. There are many ways to jump conditionally after a M99 return when using macros.

## M109 INTERACTIVE USER INPUT

This M code allows a G-code program to place a short prompt on the screen, get a single-character input from the user and store it in a macro variable. The first 15 characters from the comment following the M109 will be displayed as a prompt in the lower left corner of the screen. A macro variable in the range 500 through 599 must be specified by a P code. Note also that due to the look-ahead feature, it is necessary to include a loop in the program following the M109 to check for a non-zero response before continuing. The program can check for any character that can be entered from the keyboard by comparing with the decimal equivalent of the ASCII character. Here are a few common characters:

A - 65	a - 97	N - 78	n - 110	1 - 49	- 45
B - 66	b - 98	Y - 89	y - 121	2 - 50	* 42
C - 67	c - 99	0 - 48	+ - 43	3 - 51	/ 47

The following sample program will ask the user a Yes or No question, then wait for either a Y or an N to be entered. All other characters will be ignored.

```
N1 #501=0.          (Clear the variable)
M109 P501           (Sleep 1 min?)
N5 IF [ #501 EQ 0. ] GOTO5   (Wait for a key)
IF [ #501 EQ 89. ] GOTO10    (Y)
IF [ #501 EQ 78. ] GOTO20    (N)
GOTO1               (Keep checking)
N10                 (A Y was entered)
M95 (00:01)
GOTO30
N20                 (An N was entered)
G04 P1.              (Do nothing for 1 second)
N30                 (Stop)
M30
```

The following sample program will ask the user to select a number, then wait for a 1, 2 or a 3 to be entered. All other characters will be ignored.



VS-Series

O00234	(Sample program)
N1 #501=0.	(Clear the variable)
M109 P501	(Pick 1, 2 or 3:)
N5 IF [ #501 EQ 0. ] GOTO5	(Wait for a key)
IF [ #501 EQ 49. ] GOTO10	(1)
IF [ #501 EQ 50. ] GOTO20	(2)
IF [ #501 EQ 51. ] GOTO30	(3)
GOTO1	(Keep checking)
N10	(A 1 was entered)
M95 (00:01)	
GOTO30	
N20	(A 2 was entered)
G04 P5.	(Do nothing for 5 seconds)
N30	(A 3 was entered)
M30	



VS-Series

## 6. SETTINGS

The Settings pages contain values that control machine operation. Most settings can be changed by the operator. The Settings display contains a short description on the left, and the numeric value or other choice on the right. In general, settings allow the operator or setup person to lock out, turn on or specify functions.

The settings are organized into pages of functionally similar groupings. This makes it easier for the user to remember where the settings are located and reduces the amount of time spent maneuvering through the Settings display.

Use the vertical cursor keys to move to the desired setting. Depending on the setting, you may change it by entering a new number or, if the setting has specific values, press the horizontal cursor keys to display the choices. Press the **WRITE/ENTER** key to enter or change the value. The message near the top of the screen tells you how to change the selected setting.

Setting 26 is the machine serial number; it is protected from user change.

One of the more commonly adjusted settings will be number 34, 4th Axis Diameter, which controls the surface feedrate of a fourth axis cutting feed. Setting 34 is used to translate the linear feedrate (ipm) into a rotary feedrate (degrees). When the diameter setting is correct, the programmed surface feedrate will be correct for helical cuts. A-axis feeds depend on this setting to determine the correct angular rate.

If you have a fourth-axis rotary table or 5C indexer, or a fifth axis, it should be disabled from the Settings display when it is removed from the machine. Do not connect or disconnect any cables with the control on. If you do not disable the rotary axis setting when it is disconnected, you will get an alarm.

Settings may be sent and received with the RS-232 port. See the Data Input / Output section for a description of how to do this.

Following is a detailed description of each setting.

### 1 AUTO POWER OFF TIMER

This is a numeric setting. When it is set to a number other than zero, the machine will automatically be turned off after that many minutes of idle operation. This will not occur while a program is running and will not occur while the operator is pressing any keys. The auto-off sequence gives the operator a 15-second warning, and pressing any key will interrupt the sequence.

### 2 POWER OFF AT M30

This is an On/Off setting. If it is set to ON, the machine will begin an automatic power-down when an M30 ends a program. The auto-off sequence gives the operator a 30-second warning, and pressing any key will interrupt the sequence.

### 3 RESERVED

This setting is reserved for future use.

### 4 GRAPHICS RAPID PATH

This is an On/Off setting. It changes what is displayed in graphics. When it is OFF, the rapid motions do not leave a trail. When it is ON, rapid motions leave a dashed line on the screen.

### 5 GRAPHICS DRILL POINT

This is an On/Off setting. It changes what is displayed in graphics. When it is OFF, nothing is added to the graphics display. When it is ON, any motion in the Z axis will leave an X mark on the screen.



VS-Series

## 6 FRONT PANEL LOCK

This is an On/Off setting. When it is OFF, the machine operates normally. When it is ON, the **SPINDLE CW** and **CCW** buttons are disabled.

## 7 PARAMETER LOCK

This is an On/Off setting. When it is OFF, parameters can be changed. When it is ON, parameter changes are locked out except for parameters 81 through 100. When the control is powered up, this setting is ON.

## 8 PROG MEMORY LOCK

This is an On/Off setting. When it is OFF, memory can be edited. When it is ON, memory edit functions are locked out.

## 9 DIMENSIONING

This is an inch/metric setting. When it is set to INCH, the programmed units for **X**, **Y**, and **Z** are inches to 0.0001. When it is set to METRIC, programmed units are millimeters to 0.001. When this setting is changed from inches to metric, or vice versa, all offset values will be converted accordingly. However, changing this setting will **not** automatically translate a program already stored in memory; you must change the programmed axis values for the new units. When set to INCH, the Group 6 default G code is G20. When set to METRIC, the default G code is G21.

	INCH	METRIC
<b>Feed</b>	inches/min.	mm/min.
<b>Max Travel</b>	+/- 15400.0000	+/-39300.000
<b>Min. Programmable Dimension</b>	.0001	.001
<b>Feed Range</b>	.0001 to 300.000 in/min.	.001 to 1000.000

Axis Jog Keys		
<b>.0001 Key</b>	.0001 in/jog click	.001 mm/jog click
<b>.001</b>	.001 in/jog click	.01 mm/jog click
<b>.01</b>	.01 in/jog click	.1 mm/jog click
<b>.1 Key</b>	.1 in/jog click	1 mm/jog click

## 10 LIMIT RAPID AT 50%

This is an On/Off setting. When it is OFF, the highest rapid speed of 100% is available. When it is ON, the highest rapid rate is limited to 50% of maximum; when you press the 100% button, the display will indicate a 50% rapid override. If this setting is turned on when the machine is powered up, the maximum override will automatically be limited to 50%.

## 11 BAUD RATE SELECT

This setting allows the operator to change the data rate for the first serial port. This applies to upload and download of programs, settings, offsets and parameters, and to DNC functions.

## 12 PARITY SELECT

This setting defines parity for the first serial port. The possible values are: NONE, ODD, EVEN, ZERO. When set to NONE, no parity bit is added to the serial data. When set to ZERO, a 0 bit is added. EVEN and ODD work like normal parity functions. Make sure you know what your system needs. XMODEM must use 8 data bits and no parity.

## 13 STOP BIT

This setting designates the number of stop bits for the first serial port. It can be 1 or 2.



VS-Series

## 14 SYNCHRONIZATION

This changes the synchronization protocol between sender and receiver for the first serial port. When set to RTS/CTS, the signal wires in the serial data cable are used to tell the sender to temporarily stop sending data while the receiver catches up. When it is set to XON/XOFF, those ASCII character codes are used by the receiver to tell the sender to temporarily stop. XON/XOFF is the most common setting.

DC CODES is like XON/XOFF, except that paper tape punch or reader start/stop codes are sent. XMODEM is a receiver-driven communications protocol that sends data in blocks of 128 bytes. XMODEM gives the RS-232 communication added reliability because each block is checked for integrity. Refer to the Data Input/Output section for more information. XMODEM must use 8 data bits and no parity.

## 15 H & T CODE AGREEMENT

This is an On/Off setting. When it is OFF, no special functions occur. When it is ON, the machine checks to ensure that the **H** offset code matches the tool presently in the spindle. This check can help to prevent crashes. In program restart, this check is not done until motion begins.

## 16 DRY RUN LOCK OUT

This is an On/Off setting. When it is OFF, the machine operates normally. When it is ON, the **DRY RUN** function cannot be turned on.

## 17 OPT STOP LOCK OUT

This is an On/Off setting. When it is OFF, the machine operates normally. When it is ON, the **OPTIONAL STOP** function cannot be turned on.

## 18 BLOCK DELETE LOCK OUT

This is an On/Off setting. When it is OFF, the machine operates normally. When it is ON, the **BLOCK DELETE** function cannot be turned on.

## 19 FEEDRATE OVERRIDE LOCK

This is an On/Off setting. When it is OFF, the machine operates normally. When it is ON, the **FEEDRATE** override buttons are locked out.

## 20 SPINDLE OVERRIDE LOCK

This is an On/Off setting. When it is OFF, the machine operates normally. When it is ON, the **SPINDLE** speed override buttons are locked out.

## 21 RAPID OVERRIDE LOCK

This is an On/Off setting. When it is OFF, the machine operates normally. When it is ON, the **RAPID** override buttons are locked out.

## 22 CAN CYCLE DELTA Z

This is a numeric entry. It must be in the range of 0.0 to 29.9999 inches. This setting specifies the delta **Z** used in the G73 canned cycle when the Z axis is retracted to clear chips.

## 23 9xxx PROGS EDIT LOCK

This is an On/Off setting. When it is OFF, the machine operates normally. When it is ON, the 9000 series programs (usually macro programs) are invisible to the operator and cannot be uploaded or downloaded. They also cannot be listed, edited, or deleted.

## 24 LEADER TO PUNCH

This setting is used to control the leader sent to a paper tape punch device connected to the first RS-232 port. The values that can be selected are NONE, BLANK, or NUL CHAR. NONE causes no extra data to be sent as a leader. BLANK causes two feet of blanks to be punched at the start of a program and one foot of blanks at the end. NULL causes the same blanks but uses the ASCII code null, which is all zero.



## 25 EOB PATTERN

This setting controls what is sent out and what is expected as input to represent the EOB (end of block) on serial port 1. The possible selections are: CR LF, LF only, CR only, or LF CR CR.

## 26 SERIAL NUMBER

This is a numeric entry. It is the serial number of your machine. It cannot be changed.

## 27 G76/G77 SHIFT DIR.

This setting controls the shift direction used to clear a boring tool during a G76 or G77 canned cycle. The possible selections are: X+, X-, Y+, or Y-.

## 28 CAN CYCLE ACT W/O X/Y

This is an On/Off setting. When it is OFF, an initial canned cycle definition without an **X** or **Y** motion will not cause the canned cycle to be executed. When it is ON, the initial canned cycle definition will cause one cycle to be executed even if there is no **X** or **Y** motion in that command block. Note that if there is an **L0** in that block, it will never execute the canned cycle on the definition line.

## 29 G91 NON-MODAL

This is an On/Off setting. When it is OFF, the machine operates normally. When it is ON, G91 is not modal and applies only to the program block in which it occurs.

## 30 4TH AXIS ENABLE

When this setting is OFF, the fourth axis is disabled and no commands can be sent to that axis. Otherwise, it calls up the parameters for the selected rotary table. A change to rotary parameters is saved under the selected table type for later recall. In order to change this setting, the servos must be turned off.

## 31 RESET PROGRAM POINTER

This is an On/Off setting. When it is OFF, the **RESET** button will not change the executing program pointer. When it is ON, **RESET** will move the program pointer to the beginning of the program.

## 32 COOLANT OVERRIDE

This setting controls how the coolant pump operates. The possible selections are: NORMAL, OFF, or IGNORE. The NORMAL setting allows the operator to turn the pump on and off manually or with **M** codes. The OFF setting will generate an alarm if an attempt is made to turn the coolant on manually or from a program. The IGNORE setting will ignore all coolant commands, but the pump can be turned on manually.

## 33 COORDINATE SYSTEM

This setting changes the way the G92/G52 offset system works. It can be set to FANUC, HAAS, or YASNAC. The Fanuc and Haas class of controls use the G52 offset differently than the Yasnac class of controls. In a Fanuc or Haas control, G52 sets a local coordinate system. In a Yasnac control, the G52 stands on its own as another work offset.

Setting 33 set to YASNAC with G52:

If set to YASNAC, G52 is just another work offset, just like using a G55.

Setting 33 set to FANUC with G52:

If set to FANUC, any values in the G52 register will be added to **all** other work offsets, as a global coordinate shift. This G52 value can be entered either manually or through a part program, to redefine locations for numerous part features or multiple parts on a fixture. When FANUC is the Setting 33 selection, pressing **RESET**, **M30**, or Power Down will clear out any value in G52.

Setting 33 set to HAAS with G52:

If set to HAAS, G52 will work the same way as with FANUC - as a global coordinate shift. The difference is that the G52 value cannot be cleared as easily as with a FANUC setting (by hitting **RESET**, **M30**, or Power Down). If Setting 33 is set to HAAS, you can zero the coordinate shift value in G52 by manually entering zero, or by programming it with G52 X0, Y0 and/or Z0.



VS-Series

Setting 33 set to YASNAC with G92:

If set to YASNAC when programming in a G92 X0 Y0, the control will enter the machine's present location as a new Work Zero Offset, and that location will be entered into and seen in the G52 register. However, if you program a G92 X-2.5 Y1.0 (for example), the X and Y values will define the given distance and direction away from the machine's present location to recognize for the new Work Zero Offset. Again, that zero location will be entered into the G52 register, and **the G52 will be used separately** from any other work offsets.

Setting 33 set to FANUC or HAAS with G92:

If Setting 33 is set at FANUC or HAAS with a G92, it will work like the YASNAC setting, except that the new Work Zero location value will be loaded into a new G92 register at the bottom of the Work Zero Offset list. This new value in the G92 register **will be used in addition to the presently recognized work offset** to define the new Work Zero location.

### 34 4TH AXIS DIAMETER

This is a numeric entry that must be in the range of 0.0 to 29.9999 inches. It is used to set the diameter of the **A** axis, which determines the angular feedrate. Since the feedrate specified in a program is always inches per minute (or mm per minute), the control must know the diameter of the part being worked in the **A** axis in order to compute the angular feedrate. When this setting is set correctly, the surface feedrate on a rotary cut will be exactly the feedrate programmed into the control. The feedrate will be correct only as long as the axis remains orthogonal (at right angles) to all other axes. If the user wishes to program in degrees per minute, this can be done in METRIC mode and by changing this setting to 114.592. A feedrate of F1.0 (1 mm/min) will result in a rotation of 1 degree per minute. The Fanuc convention is to translate 1 inch per minute to 10 degrees per minute. In this case, this setting should be set to 11.459 in. (This also applies to Setting 79 "Fifth Axis Diameter").

### 35 G60 OFFSET

This is a numeric entry in the range 0.0 to 0.9999 inches. It is used to specify the amount of overshoot when unidirectional positioning (G60) is programmed.

### 36 PROGRAM RESTART

This is an On/Off setting. When it is OFF, starting a program from anywhere other than the beginning may produce inconsistent results. When it is ON, restarting a program from the middle causes the entire program to be scanned to ensure that the correct tools, offsets, **G** and **M** codes, and axis positions are set correctly before the program starts at the block where the cursor is positioned. Some alarm conditions are not detected prior to motion starting.

---

**Note:** The following M codes will be processed when Setting 36, PROGRAM RESTART, is enabled:

M08 Coolant On	M41	Low Gear Override
M09 Coolant Off	M42	High Gear Override
M10 Engage 4th Axis Brake	M51-M58	Set Optional M Code
M11 Release 4th Axis Brake	M61-M68	Clear Optional M Code
M12 Engage 5th Axis Brake	M83	Air Gun On
M13 Release 5th Axis Brake	M84	Air Gun Off
M34 Increment Coolant Spigot Position	M88	Thru-Spindle Coolant ON
M35 Decrement Coolant Spigot Position	M89	Thru-Spindle Coolant OFF

### 37 RS-232 DATA BITS

This setting can be either 7 or 8. It is used to change the number of data bits for serial port 1. Normally, 7 data bits should be used, but some computers require 8. Note that parity is added to this count. XMODEM must use 8 data bits and no parity.



VS-Series

### **38 AUX AXIS NUMBER**

This is a numeric entry between 0 and 4. It is used to select the number of external auxiliary axes added to the system. If it is set to 0, there are no auxiliary axes. If it is set to 1, there is a C axis. If it is set to 2, there are **C** and **U** axes.

### **39 BEEP AT M30**

This is an On/Off setting. When it is OFF, nothing is changed. When it is ON, a program ending in an M30 will cause the keyboard beeper to sound until any key is pressed.

### **40 TOOL OFFSET MEASURE**

This setting selects how tool size is specified for cutter compensation. It can be set to either Radius or Diameter. The values in the tool offsets are used differently depending on this setting. In addition, the label on the Offsets page changes to indicate how offsets should be entered.

### **41 ADD SPACES RS232 OUT**

This is an On/Off setting. When it is OFF, programs sent out the serial port have no spaces and are difficult to read. When it is ON, spaces are added between address codes when a program is sent out via RS-232 serial port 1. This can make a program much easier to read.

### **42 M00 AFTER TOOL CHANGE**

This is an On/Off setting. When it is OFF, tool changes occur normally. When it is ON, a program stop will occur after a tool change and M00 AFTER TOOL CHANGE will be displayed at the bottom left. This affects only programmed tool changes.

### **43 CUTTER COMP TYPE**

This setting controls how the entry to cutter compensation occurs. It can be A or B. It affects only the first stroke that begins cutter compensation, and establishes the way the tool is cleared from the part being cut.

### **44 MIN F IN RADIUS CC %**

This setting is a numeric entry between 1 and 100. It affects the feedrate when cutter compensation moves the tool toward the inside of a circular cut. Such a cut will be slowed down in order to maintain a constant surface feedrate. This setting specifies the minimum feedrate as a percentage of the programmed feedrate.

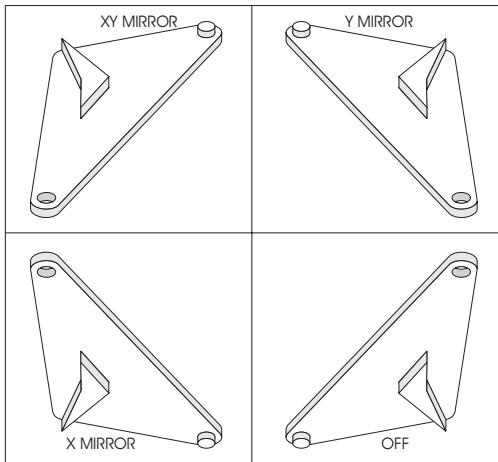
### **45 MIRROR IMAGE X-AXIS**

### **46 MIRROR IMAGE Y-AXIS**

### **47 MIRROR IMAGE Z-AXIS**

### **48 MIRROR IMAGE A-AXIS**

These are On/Off settings. When any one of these is OFF, axis motions occur normally. When one is ON, axis motion may be mirrored (or reversed) around the work zero point. See also G101Enable Mirror Image in G-codes section.



## 49 SKIP SAME TOOL CHANGE

This is an On/Off setting. When it is OFF, an M06 will always cause a tool change sequence to occur, even if the same tool is put back into the spindle. When it is ON, a tool change will only occur if a different tool will be placed in the spindle.

## 50 AUX AXIS SYNC

This changes the synchronization protocol between sender and receiver for the second serial port. When set to RTS/CTS, the signal wires in the serial data cable are used to tell the sender to temporarily stop sending data while the receiver catches up. When it is set to XON/XOFF, those ASCII character codes are used by the receiver to tell the sender to temporarily stop. XON/XOFF is the most common setting. Make sure that the Haas servo control is set to the same condition.

DC CODES is like XON/XOFF, but paper tape punch or reader start/stop codes are sent. XMODEM is a receiver-driven communications protocol that sends data in blocks of 128 bytes. XMODEM gives the RS-232 communication added reliability because each block is checked for integrity. Refer to the Data Input/Output section for more information.

## 51 DOOR HOLD SWITCH OVERRIDE (SAFETY SWITCH OVERRIDE)

This is an On/Off setting. When it is OFF, a program cannot be started when the doors are open, and opening the doors will cause a running program to stop, just like using FEED HOLD. When it is ON, and Parameter 57 bits DOOR STOP SP and SAFETY CIRC are set to zero, the door condition is ignored. When the control is powered up, this setting automatically returns to OFF. Machines equipped with a safety switch will go into feed hold if the safety switch is released.

## 52 G83 RETRACT ABOVE R

This is a numeric entry in the range 0.0 to 9.9999 inches. This setting changes the way G83 and G73 work when the tool returns to the **R** plane. Most programmers set the **R** plane well above the cut to ensure that the chip clearing motion actually allows the chips to get out of the hole, but this causes 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 clearance move to **R** occurs, the **Z** axis will be moved above the **R** plane by this setting.

## 53 JOG W/O ZERO RETURN

This is an On/Off setting. When it is OFF, jogging an axis is inhibited until the zero return operation is completed. When it is ON, jogging an axis is allowed prior to the zero return. The ON condition can be dangerous in that an axis can be run into the stops; however, the maximum speed allowed is 1.0 inch per minute or 0.0010 inch per handle increment. When the control is powered up, this setting automatically returns to OFF.



## 54 AUX AXIS BAUD RATE

This setting allows the operator to change the data rate for the second serial port. This applies to the interface with the optional **C**, **U**, **V**, and **W** axes. The possible values include: 50, 110, 200, 300, 600, 1200, 2400, 4800, 7200, 9600, 19200, 38400. Note that 4800 is standard in Haas servo controls, and this should be set to the same value.

## 55 ENABLE DNC FROM MDI

This is an On/Off setting. When it is OFF, DNC cannot be selected. When it is ON, DNC is selected by pressing the MDI key while already in MDI mode. The DNC option must be enabled in the control.

## 56 M30 RESTORE DEFAULT G

This is an On/Off setting. When it is OFF, no change to the modal **G** codes occurs at the end of a program (normally M30). When it is ON, an M30 will reset all of the modal **G** codes to their defaults and will also reset the FEED PER REV back to its default. If this setting is ON, RESET will also reset defaults.

## 57 EXACT STOP CANNED X-Y

This is an On/Off setting. When it is OFF, the rapid XY motion associated with a canned cycle may not achieve an exact stop, depending on other conditions. When it is ON, the XY motion always gets an exact stop. This will make canned cycles slower, but less likely to run into a close-tolerance fixture.

## 58 CUTTER COMPENSATION

This setting establishes the type of cutter compensation used. The selections are similar to the types of cutter compensation available in other classes of controls.

## 59 PROBE OFFSET X+

## 60 PROBE OFFSET X-

## 61 PROBE OFFSET Y+

## 62 PROBE OFFSET Y-

Settings 59 through 62 are used to define the displacement and size of the spindle probe. These numbers only apply to the probing option. These four numbers specify the travel distance in four directions, from where the probe is triggered to where the actual sensed surface is located. These settings are used by G31, G36, G136, and M75 codes. They can be either positive or negative numbers. If the probe width were 0.23 inch in diameter and the probe was set exactly at the center of the spindle, these four settings would all be 0.115 inch.

## 63 TOOL PROBE WIDTH

This setting is used to specify the width of the probe that is used to test tool diameter. This setting only applies to the probing option. It is used by G35.

## 64 T. OFS MEAS USES WORK

This is an On/Off setting. It changes the way the TOOL OFSET MESUR button works. When this is ON, the entered tool offset will be relative to the currently selected work coordinate Z offset. When it is OFF, the tool offset equals the Z machine position.

## 65 GRAPH SCALE (HEIGHT)

This setting specifies the view height of the work area that is displayed on the Graphics screen. The maximum size is automatically limited to default height; the default shows the machine's entire work area. A specific scale can be set by using the following formula. The default setting is the total Y travel.

$$\text{Total Y travel} = \text{Parameter 20 / Parameter 19 (16.25 VF-0 through VF-2)}$$

$$\text{Scale} = \text{Total Y travel / Setting 65}$$

## 66 GRAPHICS X OFFSET

This setting locates the right side of the scaling window relative to the machine X zero position (see Displays section). Its default is zero.



VS-Series

## 67 GRAPHICS Y OFFSET

This setting locates the top of the scaling window relative to the machine Y zero position (see Displays section). Its default is zero.

## 69 DPRNT LEADING SPACES

This is an On/Off setting. It suppresses leading spaces that are generated by a macro DPRNT format statement. In a DPRNT statement the format specifies the number of characters allotted, and sent to the serial port, for a given variable. If the number of characters is less than allowed for, then leading spaces are inserted and sent to the serial port. When this setting is OFF (which is the default value), then no leading spaces are generated. The following example illustrates control behavior when this setting is OFF or ON.

#1= 3.0 ;	Setting 69:	OFF	ON
G0 G90 X#1 ;	OUTPUT:	X3.0000	X 3.0000
DPRNT[X#1[44]] ;			

## 70 DPRNT OPEN/CLOS DCODE

This is an On/Off setting. It controls whether the POPEN and PCLOS statements in macros send DC control codes to the serial port. When the setting is ON, these statements will send DC control codes. When it is OFF, the control codes are suppressed. Its default value is ON.

## 71 DEFAULT G51 SCALING

This specifies the scaling for a G51 command when the P address is not contained in the same block. It must be in the range 0.001 to 8380.000. This setting's default is 1.000.

## 72 DEFAULT G68 ROTATION

This specifies the rotation, in degrees, for a G68 command when the R address is not contained in the same block. It must be in the range 0.0000 to 360.0000. This setting's default is 0.0000.

## 73 G68 INCREMENTAL ANGLE

This is an On/Off setting. It is a switch that allows the internal variable that controls rotation to be incremented for each call to a G68 command. When this switch is ON and a G68 command is executed in the incremental mode (G91), then the value specified in the R address is added to the internal variable. Otherwise, the internal variable is set to the rotation value specified by R. The default setting is OFF.

## 74 9xxxx PROGS TRACE

This setting, along with Setting 75, is useful for debugging CNC programs. When Setting 74 is ON, the control will display all blocks that are executed in programs that have an O number of 9000 or above. When the setting is OFF, the control will not display 9000 series blocks. The default setting is ON.

## 75 9xxx PROGS SINGLE BLK

When Setting 75 is ON and the control is operating in SINGLE BLOCK mode, then the control will stop at each block in a 9000 series program and wait for the operator to press CYCLE START. When Setting 75 is OFF, then all blocks in a 9000 series program are executed in a continuous manner, even if Single Block is on. The default setting is ON.

When Setting 74 and Setting 75 are both ON, the control acts normally. That is, all blocks executed are highlighted and displayed, and when in Single Block mode there is a pause before each block is executed.

When Setting 74 and Setting 75 are both OFF, the control will execute 9000 series subroutines without displaying the blocks contained in that subroutine. If the control is in Single Block mode, no single-block pause will occur within a 9000 series subroutine.

When Setting 75 is ON and Setting 74 is OFF, then 9000 series subroutines will be displayed as they are executed.

## 76 TOOL RELEASE LOCK OUT

When this setting is ON, the tool release key is disabled. The default setting is OFF.



## 77 SCALE INTEGER F

This setting aids those wishing to run programs developed on a control other than HAAS. It allows the operator to select how the control interprets an F value (feedrate) that does not contain a decimal point. (It is recommended that programmers always use a decimal point.) The setting can be set to the following values (the default setting is DEFAULT):

Setting Value:	F12 is interpreted as:
DEFAULT	0.0012 units/minute
INTEGER	12.0        "
.1	1.2        "
.01	0.12        "
.001	0.012        "
.0001	0.0012        "

## 78 FIFTH-AXIS ENABLE

When this setting is OFF, the fifth axis is disabled and no commands can be sent to that axis. Otherwise, the selected rotary table parameters are called up. A change to rotary parameters is saved under the selected table type for later recall. In order to change this setting the servos must be turned off.

## 79 FIFTH-AXIS DIAMETER

This is a numeric entry that must be in the range 0.0 to 50 inches. It is used to set the diameter of the **B** axis, which determines the angular feedrate. Since the feedrate specified in a program is always inches per minute (or mm per minute), the control must know the diameter of the part being worked in the **B** axis in order to compute the angular feedrate. When this setting is set correctly, the surface feedrate on a rotary cut will be exactly the feedrate programmed into the control. The feedrate will be correct only as long as the axis remains orthogonal (at right angles) to all other axes.

## 80 MIRROR IMAGE B-AXIS

This is an On/Off setting. When it is OFF, axis motions occur normally. When it is on, **B** axis motion may be mirrored (or reversed) around the work zero point.

## 81 TOOL AT POWER DOWN

When the POWER UP/RESTART key is pressed, the control will change to the tool specified in this setting. If zero (0) is specified, no tool change occurs at power up. The default setting is 1.

## 82 LANGUAGE

This setting includes an UPDATE option which enables languages other than English to be loaded from disk files. Instructions for loading the files are displayed on the screen.

## 83 M30 RESETS OVERRIDE

When this setting is ON, an M30 causes feedrate override, rapid override, and spindle override to be reset to default values.

## 84 TOOL OVERLOAD ACTION

This setting causes the specified action (ALARM, FEEDHOLD, BEEP, AUTOFEED) to occur anytime a tool becomes overloaded. When set to FEEDHOLD, the message "Tool Overload" will be displayed when this condition occurs. Pressing any key will clear the message. When set to AUTOFEED, the mill automatically limits the feedrate based on the tool load (see Tool Load Monitor display).

---

**NOTES:** When tapping (rigid or floating), the feed and spindle overrides will be locked out, so the AUTOFEED feature will be ineffective (although the display will appear to respond to the override buttons).

---

The AUTOFEED feature should not be used when thread milling or autoreversing tapping heads, as it may cause unpredictable results or even a crash.

---

The last commanded feedrate will be restored at the end of the program execution, or when the operator presses RESET or turns off the AUTOFEED feature.

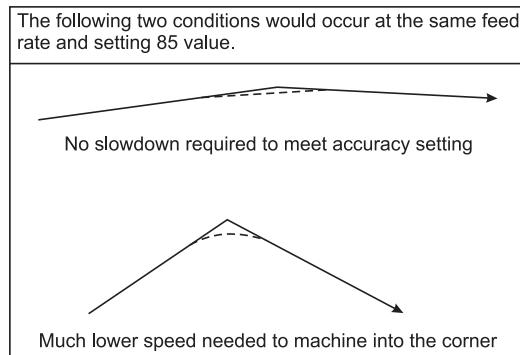
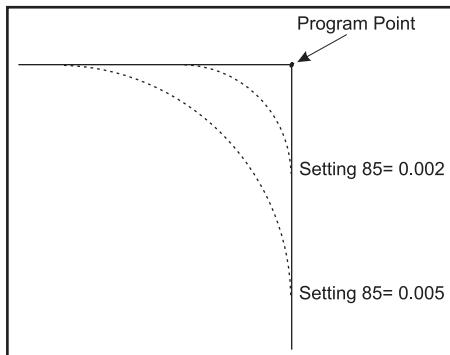
---

The operator may use the feedrate override buttons while the AUTOFEED feature is active. As long as the tool load limit is not exceeded, these buttons will have the expected effect and the overridden feedrate will be recognized by the AUTOFEED feature as the new commanded feedrate. However, if the tool load limit has already been exceeded, the control will ignore the feedrate override buttons and the commanded feedrate will remain unchanged.

## 85 MAX CORNER ROUNDING

This setting defines the accuracy of rounded corners within a selected tolerance. The initial default value is 0.05 inch. If this setting is zero (0), the control acts as if an exact stop is commanded in each motion block. Since ballscrew pitch varies from machine to machine, the control will take the lesser of the allowed error in encoder counts from all axes in order to maximize machining accuracy.

In conjunction with this setting, Parameter 134 (Exact Stop Dist.) is used as a floor so the machine will not slow down to extremely slow speeds. Alternatively, G187 can be used in the program to alter the effective value of Setting 85, but without permanently changing the setting. This method likewise takes advantage of the floor but does not require that the machine be rebooted. The amount of slow-down that occurs depends on how one stroke blends with the next.



*Setting 85 Examples*

## 86 M39 LOCKOUT

This is an On/Off setting. When it is ON, it locks out rotation of the tool changer.

## 87 M06 RESETS OVERRIDE

This is an On/Off setting. When this setting is ON and an M06 command is executed, any overrides are canceled and set to their programmed values.

## 88 RESET RESETS OVERRIDES

This is an On/Off setting. When it is ON and the RESET key is pressed, any overrides are canceled and set to their programmed values.



VS-Series

## **90 MAX TOOLS TO DISPLAY**

This setting limits the number of tools displayed on the Tool Geometry screen. This setting will accept a number from 1 to 200.

## **91 ADVANCED JOG**

This is an On/Off setting. Turning it ON enables the Index Jog and Jog Travel Limits features.

## **100 SCREEN SAVER DELAY**

This feature is intended to prevent the monitor screen from becoming etched (burned-in) when the machine is idle or unattended for many hours. When the setting is zero, the screen saver is disabled. If it is set to some number of minutes, then after that amount of time with no key presses, no jog handle motion and no alarms, the screen saver will be activated. (It will be deactivated by any key press, jog handle motion or alarm.) Note that the screen saver will not activate if the control is in Sleep mode, Jog mode, Edit mode, MEM or MDI mode with the PRGRM screen displayed, Graphics mode, or on any of the Editor screens. When it is active, the words SCREEN SAVER will be displayed in random places on an otherwise blank screen and will be changed every 2 seconds.

## **101 FEED OVERRIDE -> RAPID**

When this setting is OFF, the machine will behave normally. When it is ON and HANDLE CONTROL FEEDRATE is pressed, the jog handle will affect both the feedrate override and the rapid rate override simultaneously. That is, changing the feedrate override will cause a proportional change in the rapid rate. The maximum rapid rate will be maintained at 100% or 50% according to setting 10.

## **103 CYC START/FH SAME KEY**

This is an On/Off setting. When it is OFF, the machine operates normally. When it is ON, the CYCLE START key must be pressed and held to run a program. When CYCLE START is released, a feed hold is generated.

This setting cannot be on while Setting 104 is on. When one of them is set to ON, the other will automatically turn off. This setting can be changed while running a program.

## **104 JOG HANDL TO SNGL BLK**

This is an On/Off setting. When it is OFF, the machine operates normally. When it is ON and SINGLE BLOCK is selected, the jog handle can be used to single-step through a program. Reversing the jog handle direction will generate a feed hold. This can be useful when an unexpected long motion block is encountered.

CYCLE START must be used to begin running a program.

This setting cannot be ON while Setting 103 is on. When one of them is set to ON, the other will automatically turn off. This setting can be changed while running a program.

## **108 QUICK ROTARY G28 [requires CIRC. WRAP. = 1]**

In order to use Setting 108, the appropriate A, B or TT axis parameter CIRC. WRAP. (bit 10 for all axes) must be set to 1. Otherwise, the control will ignore setting 108. A axis parameter is 43; B axis, Parameter 151; TT, Parameter 462.

## **109 WARMUP TIME IN MIN.**

This is the number of minutes (maximum = 300 minutes from power-up) during which the compensations specified in Settings 110-112 are applied. If this is set to zero, no compensation will be applied.

## **110 WARMUP X DISTANCE**

## **111 WARMUP Y DISTANCE**

## **112 WARMUP Z DISTANCE**

These settings are used in conjunction with Setting 109. Settings 110, 111 and 112 specify the amount of compensation ( $\text{max} = \pm 0.0020"$  or  $\pm 0.051 \text{ mm}$ ) applied to the X, Y and Z axes respectively. No compensation is applied if the setting is zero.



VS-Series

## 114 CONVEYOR CYCLE TIME (MINUTES)

### 115 CONVEYOR ON-TIME (MINUTES)

These two settings control the intermittent chip conveyor function. Setting 114 controls how often the cycle is to be repeated, and Setting 115 specifies the number of minutes the conveyor will run. For example, if Setting 114 is set to 30 and Setting 115 is set to 2, the chip conveyor will turn itself on every half hour, run for two minutes, then turn itself off. On-time (Setting 115) should be no greater than 80% of cycle time.

---

**NOTE:** The CHIP FWD button (or M31) will start the conveyor in the forward direction and activate the cycle.

---

**NOTE:** The CHIP REV button (or M32) will start the conveyor in the reverse direction and activate the cycle.

---

**NOTES:** The CHIP STOP button (or M33) will stop the conveyor and cancel the cycle. If Parameter 209 bit 16, CNVY DR OVRD, is set to 0 and the chip conveyor is cycling, opening the machine door will cause the conveyor to stop and suspend cycling. When the door is closed, the cycle will resume.

If Parameter 209 CNVY DR OVRD is set to 1 and the chip conveyor is cycling, the conveyor will continue to run when the door is open but will stop at the end of the cycle and cancel the cycle. When the door is closed, the cycle will resume.

Under no circumstances will the chip conveyor automatically start running when the door is open.

## 116 PIVOT LENGTH [only changeable in DEBUG mode]

Setting 116 is intended to be set once, when the machine is first built, and then never changed unless the machine is physically altered. This is to prevent accidental loss of the setting value. A settings file can be loaded from a disk or via RS-232, and the values of all settings will be replaced. However, if the control is not in DEBUG mode at the time, and the value of Setting 116 is to be changed, all settings **except** 116 will be changed and Alarm 202, SETTING CRC ERROR, will be generated.

## 117 G143 GLOBAL OFFSET

This setting is provided for customers who have several 5-axis mills and want to transfer the programs and tools from one to another. The pivot length difference can be entered into this setting, and it will be applied to the G143 tool length compensation.

## 118 M99 BUMPS M30 CNTRS

When this setting is ON, an M99 will add an increment to the M30 counters that are visible in the CURNT COMNDS display (press PAGE DOWN twice). Note that an M99 will only increase the counters when it occurs in loop mode in a main program, not a sub-program. An M99 used as a sub-program return or used with a P value to jump to another part of the program will not affect the M30 counters.

## 119 OFFSET LOCK

This is an On/Off setting. When it is OFF, no special functions occur. When it is ON, the values in the Offset display may not be altered. However, programs which alter offsets will still be able to do so.

## 120 MACRO VAR LOCK

This is an On/Off setting. When it is OFF, no special functions occur. When it is ON, macro variables may not be altered at the control. However, programs which alter macro variables will still be able to do so.



VS-Series

### **121 APC PAL. ONE LOAD X**

This is the APC pallet #1 X axis load position. It should be set at the time the APC is installed. Units are inch or mm (Setting 9).

### **122 APC PAL. ONE LOAD Y**

This is the APC pallet #1 Y axis load position. It should be set at the time the APC is installed. Units are inch or mm (Setting 9).

### **123 APC PAL. ONE UNLOAD X**

This is the APC pallet #1 X axis unload position. It should be set at the time the APC is installed. Units are inch or mm (Setting 9).

### **124 APC PAL. ONE UNLOAD Y**

This is the APC pallet #1 Y axis unload position. It should be set at the time the APC is installed. Units are inch or mm (Setting 9).

### **125 APC PAL. TWO LOAD X**

This is the APC pallet #2 X axis load position. It should be set at the time the APC is installed. Units are inch or mm (Setting 9).

### **126 APC PAL. TWO LOAD Y**

This is the APC pallet #2 Y axis load position. It should be set at the time the APC is installed. Units are inch or mm (Setting 9).

### **127 APC PAL. TWO UNLOAD X**

This is the APC pallet #2 X axis unload position. It should be set at the time the APC is installed. Units are inch or mm (Setting 9).

### **128 APC PAL. TWO UNLOAD Y**

This is the APC pallet #2 Y axis unload position. It should be set at the time the APC is installed. Units are inch or mm (Setting 9).

### **129 APC PAL. SAFE X POS**

This is the receiver X axis position where it is safe to open the door. It should be set at the time the APC is installed. Units are inch or mm (Setting 9).

### **130 RIGID-TAP RETRACT MULT**

If it is set to zero or 1, the machine behaves normally. If it is set to 2, it will be the equivalent of running a G84 with a J code of 2. That is, the spindle will retract twice as fast as it went in. If this setting is set to 3, it will retract three times as fast. Note that specifying a J code for a rigid tap will override setting 130.

Note also that if RIGID TAP is turned off, this setting has no effect.

### **131 AUTO DOOR**

This setting supports the Auto Door option. It should be set to ON when the automatic door hardware is installed and the operator wants it to function. Otherwise it should be set to OFF. When it is ON, the following items need to be verified:

Setting 51, DOOR HOLD OVERRIDE, is set to ON.

Parameter 57 bit 8, SAFETY CIRC, is set to zero.

Parameters 292, 293 and 251 are set appropriately (see definitions under Parameters).

The door will close when CYCLE START is pressed and will open when the program reaches an M00, M01 (with Optional Stop turned on) or M30 and the spindle has stopped turning. Note that if any of the aforementioned parameters and settings are set incorrectly, the Auto Door option will not function.



VS-Series

This setting will enable and disable the autodoor operation via CYCLESTART, M00, M01, M30.

**WARNING!** With this change, M80 and M81 cannot be disabled with this setting. Unexpected movement may occur. To prevent M80 and M81 codes from causing the autodoor to move, parameter 251 A DOOR OPEN ERRTIME must be changed to zero. Note: If the autodoor is disabled in this manner, the user must make a note of the value of parameter 251 so that it can be properly re-enabled later. If an incorrect value is entered into parameter 251, the autodoor may malfunction.

### **133 REPEAT RIGID TAP**

This setting ensures that the spindle is oriented during tapping so that the threads will line up properly.

### **134 CONNECTION TYPE**

This setting can be FLOPPY, NET, or ZIP. When it is set to FLOPPY, loading and saving programs is performed in the usual way via the floppy disk drive installed in the control. When it is set to NET, loading and saving occur via the user-supplied network connection (provided that connection was successfully established at power-up). When it is set to ZIP, program loading and saving occurs via the user-supplied Zip™ drive (assuming such a device is connected.) When Setting 133 is ON, the Setting 134 connection type will appear on the LISTPROG screen as follows: F4 DIR-FLOPPY, F4 DIR-NET, or F4 DIR-ZIP.

### **135 NETWORK TYPE**

This setting can be NONE, NOVELL, NT/IPX, NT/TCP or ADV/TCP. It specifies the user-supplied network connection type. When it is set to NONE, only a floppy disk or a user-supplied Zip™ drive is accessible. If ADV/TCP is selected, a setting page will appear in place of setting 135. To revert back, chngeNET TYPE to none.

### **136 SERVER**

This setting contains the user-supplied server name (up to 8 characters long.) If this setting will not be used, a semicolon (EOB symbol) should be entered.

### **137 USERNAME**

This setting contains the user-specified account name (up to 8 characters long.) If this setting will not be used, a semicolon (EOB symbol) should be entered.

### **138 PASSWORD**

This setting contains the user-specified password (up to 8 characters long.) If this setting will not be used, a semicolon (EOB symbol) should be entered.

### **139 PATH**

This setting contains the user-specified Novell-Path or NT Root Directory name (up to 18 characters long), depending on the network server used. For a Novell network, this is the user's path name; for example, U:\USERS\JOHNDOE. For a Microsoft network, this is the root directory\desired directory name; for example, USERS\JOHNDOE. If this setting will not be used, a semicolon (EOB symbol) should be entered.

### **140 TCP ADDR**

This setting is used only for TCP networks and contains the static user-specified TCP/IP address in the server domain (up to 15 characters long). For example: 192.168.1.2. If this setting will not be used, a semicolon (EOB symbol) should be entered.

### **141 SUBNET**

This setting is used only for TCP networks and contains the user-specified subnet mask (up to 15 characters long; for example, 255.255.255.0). If this setting will not be used, a semicolon (EOB symbol) should be entered.



VS-Series

## 142 OFFSET CHNG TOLERANCE

This setting generates a warning message if an offset is changed by more than the specified amount. It is intended to help prevent operator errors. It can be set to any number from 0 to 99.9999. When set to zero, the feature is inactive. When it contains a non-zero number and an attempt is made to change an offset by more than the amount of the non-zero number (either positive or negative), the following prompt will be displayed: "XX changes the offset by more than Setting 142! Accept (Y/N)?" If "Y" is entered, the control updates the offset as usual; otherwise, the change is rejected.

## 143 MACHINE DATA COLLECT

This setting enables the user to extract data from the control using a Q command sent via the RS-232 port. This feature is software-based and requires a computer to request, interpret and store data from the control. Note that the control will only respond to a Q command when this setting is ON. The following output format is used:

<STX> <CSV response> <ETB> <CR/LF> <0x3E>

Note: STX = 0x02 (ctrl-B); ETB = 0x17 (ctrl-W)

The following commands can be used:

Q100 - Machine Serial Number

Q101 - Control Software Version

Q102 - Machine Model Number

Q104 - Mode (LIST PROG, MDI, MEM, JOG, etc.)

Q200 - Tool Changes (total)

Q201 - Tool Number in use

Q300 - Power-on Time (total)

Q301 - Motion Time (total)

Q303 - Last Cycle Time

Q304 - Previous Cycle Time

Q400 - not currently used

Q401 - not currently used

Q402 - M30 Parts Counter #1 (resettable at control)

Q403 - M30 Parts Counter #2 (resettable at control)

Q500 - Three-in-one (PROGRAM, Oxxxxx, STATUS, PARTS, xxxx)

If the control is busy, the control will output "STATUS, BUSY." If a request is not recognized, the control will output "UNKNOWN."

In addition, when Setting 143 MACHINE DATA COLLECT is ON, the user has the ability to request the contents of any macro or system variable by using the Q600 command. Additionally, variables 1 thru 999 and 2001 thru 2800 can be written to using the new E command. Syntax is .Q600 xxxx. where xxxx is the macro variable. To write to a macro variable, use .Exxxxx yyyy.yyyyy. where xxxx is the macro variable and yyyy.yyyyy is the new, floating point value. Note that this command should only be used when there are no alarms present.

This feature can be enhanced with the installation of the following optional parts, M-code relay board, a power ON relay and a new E-STOP switch (set parameter 209 bit 23 MCD RLY BRD and parameter 315 bit 26 STATUS RELYS to 1). The Haas Service personnel will install these optional parts. Once installed output relays 40 thru 47, a power ON relay and the E-STOP switch are used to communicate the status of the control.

The following communications will be received when used with the optional parts.

- \* E-STOP contacts. This will be closed when the E-STOP button is pushed.
- \* Power ON - 115 VAC. Indicates that the control is turned ON. It should be wired to a 115 VAC coil relay for interface.
- \* Spare Output Relay 40. Indicates that the control is In-Cycle (running.)
- \* Spare Output Relay 41 and 42:
  - 11 = MEM mode & no alarms (AUTO mode.)
  - 10 = MDI mode & no alarms (Manual mode.)
  - 01 = Single Block mode (Single mode)
  - 00 = other modes (zero, DNC, jog, list prog, etc.)
- \* Spare Output Relay 43 and 44:
  - 11 = Feed Hold stop (Feed Hold.)
  - 10 = M00 or M01 stop
  - 01 = M02 or M30 stop (Program Stop)
  - 00 = none of the above (could be single block stop or RESET.)
- \* Spare Output Relay 45 (Feed Rate Override is active and Feed Rate is NOT 100%)
- \* Spare Output Relay 46 (Spindle Speed Override active and Spindle Speed is NOT 100%)
- \* Spare Output Relay 47 (Control is in EDIT mode)

#### **144 FEED OVERRIDE->SPINDLE**

This setting is intended to keep the chip load constant when an override is applied. When this setting is ON, any feedrate override will also be applied to the spindle speed, and the spindle overrides will be disabled.

#### **146 APC PALLET 3 LOAD X**

See Setting 121

#### **147 APC PALLET 3 LOAD Y**

See setting 122

#### **148 APC PALLET 3 UNLOAD X**

See Setting 123

#### **149 APC PALLET 3 UNLOAD Y**

See Setting 124

#### **150 APC PALLET 4 LOAD X**

See Setting 121

#### **151 APC PALLET 4 LOAD Y**

See Setting 122

#### **152 APC PALLET 4 UNLOAD X**

See Setting 123

#### **153 APC PALLET 4 UNLOAD Y**

See Setting 124

#### **154 APC PALLET 3 & 4 SAFE X**

See Setting 129



## 155 LOAD POCKET TABLES

This setting should only be used when a software upgrade is performed and/or memory has been cleared and/or the control is re-initialized. Setting 155 automatically defaults to OFF when the machine is turned on. This setting controls loading of a Pocket Tool table from an Offset file. On a mill with a side-mount tool changer that requires the use of a Pocket Tool table, the contents of the Pocket Tool and Pocket Flag tables are included in the Offset file when it is saved to a floppy disk or via RS-232. If this setting is OFF when loading an Offset file from a floppy disk or RS-232, the contents of the Pocket Tool table will be unaltered. In order to replace the contents of the Pocket Tool table with the data from the file, the setting must be ON.

## 156 SAVE OFFSET WITH PROG

This On/Off setting controls whether offsets are automatically saved with programs to floppy disk and RS-232. The offsets are saved (to floppy disk or RS-232) in the same file as the programs, but under the heading O999999. The offsets will appear in the file before the final % sign.

## 157 OFFSET FORMAT TYPE

This setting controls the format in which offsets are saved with programs to floppy disk or RS-232. When it is set to B, the old-style format will be used; that is, each offset is saved on a separate line with an N value and a V value. When it is set to A, a newer format is used that is easier to read and edit. The newer format resembles the display screen format, and contains decimal points and column headings. Offsets saved in this format can be more easily edited on a PC and later reloaded into the control.

## 158,159,160 XYZ SCREW THERMAL COMP%

These settings that can be set from -30 to +30 and will adjust the existing screw thermal compensation by -30% to +30% accordingly.

## 162 DEFAULT TO FLOAT

This feature is compatible with Fanuc. When this setting is ON, the control will append a decimal point to values entered without a decimal point (for certain address codes). When this setting is Off, the machine behaves as before, that is, values following address codes that do not include decimal points are treated as machinist's notation (i.e., thousandths or ten-thousandths.) For example, in INCH mode:

Value entered	With Setting OFF	With Setting ON
X-2	X-.0002	X-2.

In MM mode:

X-2	X-.002	X-2.
-----	--------	------

This feature applies to the following address codes:

X,Y,Z,A,B,C,E,F,I,J,K,U,W  
D (except with G73)  
R (except with G71 in YASNAC mode)

Note that this setting affects the interpretation of all programs entered either manually or from disk or via RS-232. It does not alter the effect of setting 77 Scale Integer F.

## 163 DISABLE .1 JOG RATE

This setting disables the highest jog rate. If the highest jog rate is selected, the next lower rate is automatically selected instead. During times when it would be dangerous to jog at the highest rate, the operator can turn this setting on and it will help to prevent accidental damage to tools and parts.

## 164 ROTARY INCREMENT

This setting applies to the ROTARY INDEX button on the EC300. It specifies the rotation distance for the outside rotary table when the ROTARY INDEX button is pressed. It should be set to a value from -360 to 360 degrees. If it is set to zero, the rotary table will not rotate.