I.S.O. Programming

Dart 21i M
Arrow 18i M
Sabre 18i M
Lancer 18i M
Index

1.0  G & M Code Introduction .....................................................

2.0  Machine Motion (G90 & G91).............................................

3.0  Fixture Offsets (G54, G55, G56, G57, G58, G59)..............

4.0  Program start ........................................................................
    Toolchange........................................................................
    Programmable defaults....................................................
    Work Co-ordinate Setting.................................................
    Tool Length Setting (G43)..............................................

5.0  Arcs & Circles (G2, G3 R, I, J & K).................................

6.0  Cutter Radius Compensation (G40, G41, G42)............

7.0  Helical..................................................................................

8.0  Hole canned cycles (G73, G74, G76, G80, G81, G82, G83,
    G84, G85, G86, G87, G88, G89)........................................

9.0  Sub-Programming (M98, M99)...........................

10.0 Example work answers..................................................

11.0 Supplement Information..................................................
    Macro’s.............................................................................
    Datum Shift.....................................................................
    Rotation.........................................................................
    Programmable Coolant...................................................
    Corner Radius / Chamfer............................................... 
    Programmable Data Transfer............................
    Polar Co-ordinates....................................................... 
    Automatic Corner Override..............................
    User Supplement Cycles...........................
Process from Drawing to Product completion

1. Drawing
Examine drawing to determine fixturing, machining origin, process and tooling.

2. Program preparation
Prepare a program while considering cutting conditions as R.P.M., depth of cuts and feedrates.

3. Program creation
Write the program in the control or another editing source (P.C.) as per the program preparation.

4. Test run
Test the mathematics of the program using the test run facilities i.e. Graphics (if available) & program run.

5. Machining
Set tools, set offset values and then process a trial test workpiece.

6. Product completion and Inspection
Introduction to Programming

Programming of the C.N.C. control involves the sequential study of the operations required to produce a component part using established production engineering methods.

The priority of operations (determined by either the programmer or planning engineer) is then written into a format, which can be interpreted by the control. This is known as “Word Address” programming format.

Each “Word” is a complete command, and will instruct the control to perform one specific operation, i.e. S1000 M03 will set the spindle speed to 1000 R.P.M. (S1000) and start the spindle in clockwise rotation (M03).

A number of “Words” can be programmed on the same line, (as the above example) thus reducing the amount of program steps needed in any one program.

Each “Word” has its own “Letter Address” followed by its “Numerical Data” i.e. S1000.

The value must fall within its programming range. These “Words” written on one line will complete a block of information when the “End of Block” key (EOB) is used.

i.e. 

<table>
<thead>
<tr>
<th>i.e.</th>
<th>N100</th>
<th>S1000</th>
<th>M03</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Line number)</td>
<td></td>
<td>(Speed)</td>
<td>(Machine function)</td>
</tr>
</tbody>
</table>

The “Block’s” of information sequentially listed form the “Program”
Cutting Condition Commands

Cutting conditions should be carefully examined when preparing a program, since these conditions greatly influence cutting efficiency and accuracy. The cutting conditions that determine the rate of metal removal are the “Cutting Speed”, the “Feedrate”, the “Depth of Cut” & the “Width of Cut”. These cutting conditions and the nature of the material to be cut determine the power required to take the cut. The cutting conditions must be adjusted to stay within the power available on the machine tool to be used. These conditions also effect the tool life, which would need consideration.

The following cutting conditions are required for all tooling used:

**Spindle Speed – R.P.M. (Revolutions per Minute)**
Designated with an S command.

400 rpm ⇒ S400

*Formula*

\[ \text{R.P.M.} = \frac{\text{Constant Surface Speed (C.S.S.)} \times 1000}{\pi \times \text{Diameter}} \]

C.S.S. can be found in all manufacturers tooling guides.

**Feedrate – mm/min. , inch/min. , feed/tooth, feed/rev.**
Designated with an F command.

400 mm/min. ⇒ F400

*Formula*

Feed = Number of teeth \times \text{feed/tooth (pitch)} \times \text{R.P.M.}

Feed/tooth can be found in all manufacturers tooling guides.
Programming Terms

Program Number:

O 1234

A four-digit number follows the letter O in program numbering.
The range of numbering can be as follows:

O0000 → O9999

The program numbers can be configured in a manner that allows “General Programs”, “Custom Macro Programs” & “Machine Tool Macro Programs”. The “Custom Macro Programs” & the “Machine Tool Macro Programs” can be created and “locked” by parameter settings to prevent accidental deletion or editing. Since this is a facility of the control the created programs can be split into 3 group numbers as follows:

<table>
<thead>
<tr>
<th>Program Number</th>
<th>Program Type</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>O0000 – O7999</td>
<td>“General”</td>
<td>No protection by parameter</td>
</tr>
<tr>
<td>O8000 – O8999</td>
<td>“Custom Macro”</td>
<td>Parameter 3202 #0</td>
</tr>
<tr>
<td>O9000 – O9999</td>
<td>“Machine Macro”</td>
<td>Parameter 3202 #4</td>
</tr>
</tbody>
</table>

Sequence Number:

N 0002

A numerical number follows the letter “N” at the program line beginning. “N” numbers are used as a search facility to enable simple program editing and starting. “N” numbers have no effect on the program itself but does require memory. They can be switched on or off by parameter number 0000 #5. The sequence of numbers can be set by parameter 3216.
The sequence number can be allocated as the following examples:

Example 1: (sequence numbering at each toolchange line)

N001  T1 M6
(Program for Tool 1)
N002  T2 M6
(Program for Tool 2)
N003  T3 M6
(Program for Tool 3)

Example 2: (all line numbering)

N100  T1 M6
N101  (Program for Tool 1)
N200  T2 M6
N201  (Program for Tool 2)
N300  T3 M6
N301  (Program for Tool 3)
Programming Terms (cont.)

Block:
A block is the minimum amount of “WORD” commands necessary for the machine to perform their operations.
A block takes up one line when written on a program sheet. Each line is called a block.

O1111;                               The first “Block”
N1 T1 M6;                              The second “Block”
N2 G0 G90 G40 G21 G17 G94 G80;         The third “Block”
N3 G54 X? Y? S? M3;                    The fourth “Block”
N4 G43 Z100 H?;                        The fifth “Block”

Word:
A “WORD” is the minimum command to activate a function. It is composed of an “ADDRESS” and “NUMERICAL DATA” including a sign.

\[
\begin{align*}
\text{N1} & \quad \text{word} & \quad \text{G0} & \quad \text{word} & \quad \text{X0} & \quad \text{Y0} & \quad \text{Z0} ; \\
\end{align*}
\]

Address:
An “ADDRESS” is the alphabetical letter in a word.

\[
\begin{align*}
\text{N1} & \quad \text{G0} & \quad \text{X0} & \quad \text{Y0} & \quad \text{Z0} ; \\
\end{align*}
\]

Numerical Data:
“NUMERICAL DATA” refers to the number part of a word.

\[
\begin{align*}
\text{N1} & \quad \text{G0} & \quad \text{X0} & \quad \text{Y0} & \quad \text{Z0} ; \\
\end{align*}
\]

End Of Block (EOB):
Refers to the action created at the end of a “BLOCK” to allow a new “Block” to be created. The control recognizes this as the end of this sequence of events.

\[
\begin{align*}
\text{N1} & \quad \text{G0} & \quad \text{X0} & \quad \text{Y0} & \quad \text{Z0} ; \\
\end{align*}
\]
Table of preparatory Codes (G & M Functions)

a) All Codes are divided into group types.

b) There are two types of “G” & “M” codes:

   **Non-Modal** - The code is active only in the block in which it is specified & is self-canceling.

   **Modal** - The code remains active when programmed on every line of program and does not require reprogramming on any following blocks until it is replaced by another action code of the same group number.

i.e. G01 and G00 are modal codes in group 01

```
G01 X?_____;  }
    Z?_____;  } - G01 is effective in this range.
    X?_____;  }
G00 Z?_____;  
```

---

```
1 2
2 3
3 4
4 5
5 6
6 7
7 8
8 9
9 1
```
G & M Functions

1) “G” codes marked □ on the next page are initial (defaulted) “G” codes when the power is turned on. For G20/G21 (Inch/MM), the “G” code last programmed before the machine power is turned off remains the defaulted.

2) “G” codes of group 00 are “Non Modal”. They are only effective in the block in which they are specified.

3) If a “G” code not listed or not purchased as an optional extra is commanded, an alarm (No. 010) will be displayed.

4) A number of “G” codes can be specified in the same block. When more than one “G” of the same group is specified, an alarm will be activated to inform the operator of this.

5) If any “G” code of group 01 is specified in a canned cycle mode, the canned cycle is automatically cancelled and the G80 condition entered.

Note:
Operators must note that programming G20/G21 will not convert information in offset registers, and therefore if several programs are stored in the library of either inch or metric format then the offsets must be manually changed to inch or metric units.
## G Codes

<table>
<thead>
<tr>
<th>G CODE</th>
<th>GROUP</th>
<th>FUNCTION ( * Option)</th>
</tr>
</thead>
<tbody>
<tr>
<td>G00</td>
<td>01</td>
<td>Rapid Positioning</td>
</tr>
<tr>
<td>G01</td>
<td></td>
<td>Straight Line “Feed”</td>
</tr>
<tr>
<td>G02</td>
<td></td>
<td>Circular Clockwise “Feed”</td>
</tr>
<tr>
<td>G03</td>
<td></td>
<td>Circular Anti-Clockwise “Feed”</td>
</tr>
<tr>
<td>G04</td>
<td>00</td>
<td>Dwell</td>
</tr>
<tr>
<td>G05</td>
<td></td>
<td>High Speed Cycle Machining *</td>
</tr>
<tr>
<td>G07</td>
<td></td>
<td>Hypothetical Axis Interpolation *</td>
</tr>
<tr>
<td>G07.1</td>
<td></td>
<td>Cylindrical Interpolation *</td>
</tr>
<tr>
<td>G08</td>
<td></td>
<td>Look-Ahead Control *</td>
</tr>
<tr>
<td>G09</td>
<td></td>
<td>Exact Stop</td>
</tr>
<tr>
<td>G10</td>
<td></td>
<td>Programmable Data Input *</td>
</tr>
<tr>
<td>G11</td>
<td></td>
<td>Programmable Data Input Cancel *</td>
</tr>
<tr>
<td>G15</td>
<td>17</td>
<td>Polar Co-ordinates Command Cancel *</td>
</tr>
<tr>
<td>G16</td>
<td></td>
<td>Polar Co-ordinates Command *</td>
</tr>
<tr>
<td>G17</td>
<td>02</td>
<td>XY Plane – Plan View (Z- Direction)</td>
</tr>
<tr>
<td>G18</td>
<td></td>
<td>XZ Plane – Front View (Y- Direction)</td>
</tr>
<tr>
<td>G19</td>
<td></td>
<td>YZ Plane – Side View (X- Direction)</td>
</tr>
<tr>
<td>G20</td>
<td>06</td>
<td>Imperial Dimensions</td>
</tr>
<tr>
<td>G21</td>
<td></td>
<td>Metric Dimensions</td>
</tr>
<tr>
<td>G27</td>
<td>00</td>
<td>Reference Position Return Check</td>
</tr>
<tr>
<td>G28</td>
<td></td>
<td>Return To Reference Position</td>
</tr>
<tr>
<td>G29</td>
<td></td>
<td>Return From Reference Position</td>
</tr>
<tr>
<td>G30</td>
<td></td>
<td>2nd, 3rd, &amp; 4th Reference Position Return</td>
</tr>
<tr>
<td>G40</td>
<td>07</td>
<td>Cutter Radius Compensation Cancel</td>
</tr>
<tr>
<td>G41</td>
<td></td>
<td>Cutter Radius Compensation Left</td>
</tr>
<tr>
<td>G42</td>
<td></td>
<td>Cutter Radius Compensation Right</td>
</tr>
<tr>
<td>G43</td>
<td>08</td>
<td>Tool Length Compensation +</td>
</tr>
<tr>
<td>G44</td>
<td></td>
<td>Tool Length Compensation -</td>
</tr>
<tr>
<td>G45</td>
<td>00</td>
<td>Tool Offset Increase</td>
</tr>
<tr>
<td>G46</td>
<td></td>
<td>Tool Offset Decrease</td>
</tr>
<tr>
<td>G47</td>
<td></td>
<td>Tool Offset Double Increase</td>
</tr>
<tr>
<td>G48</td>
<td></td>
<td>Tool Offset Double Decrease</td>
</tr>
<tr>
<td>G49</td>
<td>08</td>
<td>Tool Length Compensation Cancel</td>
</tr>
<tr>
<td>G50</td>
<td>11</td>
<td>Scaling Cancel *</td>
</tr>
<tr>
<td>G51</td>
<td></td>
<td>Scaling *</td>
</tr>
<tr>
<td>G50.1</td>
<td>22</td>
<td>Programmable Mirror Image Cancel *</td>
</tr>
<tr>
<td>G51.1</td>
<td></td>
<td>Programmable Mirror Image *</td>
</tr>
<tr>
<td>G52</td>
<td>00</td>
<td>Datum Shift</td>
</tr>
<tr>
<td>G53</td>
<td></td>
<td>Machine Co-ordinate Dimensioning</td>
</tr>
</tbody>
</table>
### FANUC I.S.O. PROGRAMMING NOTES

#### Chapter 1

<table>
<thead>
<tr>
<th>Command</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>G54</td>
<td>14</td>
<td>Workpiece Co-ordinate Selection 1</td>
</tr>
<tr>
<td>G55</td>
<td></td>
<td>Workpiece Co-ordinate Selection 2</td>
</tr>
<tr>
<td>G56</td>
<td></td>
<td>Workpiece Co-ordinate Selection 3</td>
</tr>
<tr>
<td>G57</td>
<td></td>
<td>Workpiece Co-ordinate Selection 4</td>
</tr>
<tr>
<td>G58</td>
<td></td>
<td>Workpiece Co-ordinate Selection 5</td>
</tr>
<tr>
<td>G59</td>
<td></td>
<td>Workpiece Co-ordinate Selection 6</td>
</tr>
<tr>
<td>G60</td>
<td>00</td>
<td>Single Direction Positioning</td>
</tr>
<tr>
<td>G61</td>
<td>15</td>
<td>Exact Stop Mode</td>
</tr>
<tr>
<td>G62</td>
<td></td>
<td>Automatic Corner Feed Override *</td>
</tr>
<tr>
<td>G63</td>
<td></td>
<td>Tapping Mode</td>
</tr>
<tr>
<td>G64</td>
<td></td>
<td>Cutting Mode</td>
</tr>
<tr>
<td>G65</td>
<td>00</td>
<td>Macro Call</td>
</tr>
<tr>
<td>G66</td>
<td>12</td>
<td>Macro Modal Call</td>
</tr>
<tr>
<td>G67</td>
<td></td>
<td>Macro Modal Call Cancel</td>
</tr>
<tr>
<td>G68</td>
<td>16</td>
<td>Rotation *</td>
</tr>
<tr>
<td>G69</td>
<td></td>
<td>Rotation Cancel *</td>
</tr>
<tr>
<td>G73</td>
<td>09</td>
<td>High Speed Peck Drilling Cycle</td>
</tr>
<tr>
<td>G74</td>
<td></td>
<td>Left Hand Tapping Cycle</td>
</tr>
<tr>
<td>G76</td>
<td></td>
<td>Fine Boring Cycle</td>
</tr>
<tr>
<td>G80</td>
<td></td>
<td>Canned Cycle Cancel</td>
</tr>
<tr>
<td>G81</td>
<td></td>
<td>Simple Drilling Cycle</td>
</tr>
<tr>
<td>G82</td>
<td></td>
<td>Drilling or Counterboring Cycle</td>
</tr>
<tr>
<td>G83</td>
<td></td>
<td>Peck Drilling Cycle</td>
</tr>
<tr>
<td>G84</td>
<td></td>
<td>Right Hand Tapping Cycle</td>
</tr>
<tr>
<td>G85</td>
<td></td>
<td>Boring Cycle</td>
</tr>
<tr>
<td>G86</td>
<td></td>
<td>Boring Cycle</td>
</tr>
<tr>
<td>G87</td>
<td></td>
<td>Back Boring Cycle</td>
</tr>
<tr>
<td>G88</td>
<td></td>
<td>Boring Cycle</td>
</tr>
<tr>
<td>G89</td>
<td></td>
<td>Boring Cycle</td>
</tr>
<tr>
<td>G90</td>
<td>03</td>
<td>Absolute Dimensions</td>
</tr>
<tr>
<td>G91</td>
<td></td>
<td>Incremental Dimensions</td>
</tr>
<tr>
<td>G92</td>
<td>00</td>
<td>Work Co-ordinate System Setting</td>
</tr>
<tr>
<td>G94</td>
<td>05</td>
<td>Feed Rate Per Minute</td>
</tr>
<tr>
<td>G95</td>
<td></td>
<td>Feed Rate Per Revolution</td>
</tr>
<tr>
<td>G98</td>
<td>10</td>
<td>Return To Initial Point During Canned Cycle</td>
</tr>
<tr>
<td>G99</td>
<td></td>
<td>Return To “R” Point During Canned Cycle</td>
</tr>
</tbody>
</table>

**G04 – Program Dwell:**

A program dwell time can be created at any point within in a program. This is a non-modal code which can only be programmed on it’s own line of program. The dwell time is programmed in milli-seconds using a \( P \) word to a maximum of 999999 milliseconds (99.9999 seconds). Some cycles have their own dwell facilities within the cycle itself.

**G04 P1000** (equals 1 second)
M codes

<table>
<thead>
<tr>
<th>M Code</th>
<th>FUNCTION ( * Option)</th>
<th>START OF SPAN</th>
<th>END OF SPAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>M00</td>
<td>Program Stop</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M01</td>
<td>Program Stop by switch</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M02</td>
<td>End Of Program</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M03</td>
<td>Spindle Clockwise</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M04</td>
<td>Spindle Anti-Clockwise</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M05</td>
<td>Spindle Stop</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M06</td>
<td>Toolchange</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M08</td>
<td>External Coolant On</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M09</td>
<td>Coolant Off</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M10</td>
<td>4th Axis Unclamp *</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M11</td>
<td>4th Axis Clamp *</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M13</td>
<td>Spindle Clockwise With External Coolant</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M14</td>
<td>Spindle Anti-Clockwise With External Coolant</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M15</td>
<td>Programmable Coolant Nozzle *</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M19</td>
<td>Spindle Orientates To Toolchange Position</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M30</td>
<td>End Of Program</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M33</td>
<td>Spindle Clockwise With Thro’ Spindle Coolant *</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M34</td>
<td>Spindle Anti-clockwise with Thro’ Spindle Coolant *</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M38</td>
<td>Thro’ Spindle Coolant *</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M50</td>
<td>5th Axis Unclamp *</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M51</td>
<td>5th Axis Clamp *</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M98</td>
<td>Sub-Program Call</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M99</td>
<td>Sub-Program End</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**M00 – Program Stop:**
After executing the block where the M00 is commanded, automatic operation stops the machine including the feed, spindle and coolant. Pressing the Cycle Start button resumes all operations.

**M01 – Optional Program Stop:**
M01 is identical to M00 but is actioned by a switch on the operators control panel. When this switch is “ON” then the code acts as M00, but when the switch is “OFF” the code is ignored and operation continues as programmed. Applications include: Checks on dimensions, Checks on tools and to remove chips during machining.

**M02 – Program End:**
This code informs the control that the program is at the end. Re-pressing the cycle start button will allow the program to be rewound before another cycle start press to run the program again. This is used in special applications, and all other applications should use M30.

**M30 – Program End:**
This code informs the control that the program is at the end and will automatically rewind the program for the next process start. The control screen will prompt the operator to “Open & Close the door” before starting the next operation.
“Absolute (G90) Program Machine Movement”

Tool motion assumes now that the spindle moves and not the Table
G90 Absolute Programming

G90 X40 Y90
X80 Y30
X120 Y80
X160 Y50
G90 Absolute Example Programming

N1 G90 X-50 Y100; Absolute Move to position 1
N2
N3
N4
N5
N6
N7
N8
N9
N10
“Incremental (G91) Program Tool Movement”

Tool motion assumes now that the spindle moves and not the Table

X+ X- Y+ Y- Z+ Z-
G91 Incremental Programming

G90 X40 Y90
G91 X40 Y-60
X40 Y50
X40 Y-30
G91 Incremental Example Programming

N1 G90 X-50 Y100; Absolute Move to position 1
N2
N3
N4
N5
N6
N7
N8
N9
N10
**Component Fixture Offsets**

**Work Coordinate System Programming (G54 - G59)**

The work co-ordinate system allows for the setting of datum's relative to the machine reference co-ordinate system.
Component Fixture Offsets
Work Coordinate System Programming (G54 - G59)

Co-ordinates values (G54 – G59) are set using the axis data information values contained within the (MACHINE) axis position table to the required spindle centreline.

When the position of the component datum has been determined in all axis, it can be entered into the appropriate work offset register.

*Note:
Co-ordinates are specified using one of the following co-ordinates systems for each datum to be set:

Workpiece co-ordinate system - G54, G55, G56, G57, G58, G59
Fixture Offsets

1) Local Offset = Global offset which is relative to all offsets G54 – G59

![Local Offset Table]

2) Work Co-ordinate Offsets = Individual offsets from either the Local Offset or Machine Offset if the Local Offset is all zero’s.

![Work Co-ordinate Offsets Table]
Part Offset

Table Length
X Working Envelope

202.376

Y Working Envelope
Table Width

Machine Reference

255.987

Plan View

Job Datum

Machine Reference (Z)

449.346

Job Datum

Table (Front)

<table>
<thead>
<tr>
<th>Table</th>
<th>00</th>
<th>X</th>
<th>0.000</th>
<th>01</th>
<th>X</th>
<th>202.376</th>
</tr>
</thead>
<tbody>
<tr>
<td>(EXT)</td>
<td>Y</td>
<td>0.000</td>
<td>(EXT)</td>
<td>Y</td>
<td>255.987</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Z</td>
<td>0.000</td>
<td></td>
<td>Z</td>
<td>-449.346</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>0.000</td>
<td></td>
<td>A</td>
<td>0.000</td>
<td></td>
</tr>
</tbody>
</table>
Fixture Offsets – Multiple Parts

Table Length

X Working Envelope

248.877

100.125

-99.875

G54

G55

99.887

-98.902

G56

G57

-100.113

101.098

97.653

Machine Reference

Table Width

Y Working Envelope

Local

255

-102.347

Machine Reference (Z)

449.346

Job Datum

Table (Front)
Fixture Offsets – Multiple Parts (cont.)

The values contained in the “G54 – G59 Offsets” can be stored in the program using the G10 program data transfer system to save setting on the next batch resetting time (see Chapter 11 for more information on G10 – Programmable Data Entry).

The values are incremental from the “Local Offset” so will never change. The operator only has to set the main setting bore on the next batch set-up in External XY offsets.

Program example

O1000
G10 L2 P1 X-99.875 Y99.887
G10 L2 P2 X100.125 Y101.098
G10 L2 P3 X-102.347 Y-100.113
G10 L2 P4 X97.653 Y-98.902
T1 M6
Etc.

Any alterations to the part locations must be set in the program and not the actual offset page as the current values would be overwritten with the program values.
**Program Start:**

O1111 ;
T? M6 (Tool change line) ;
G0 G90 G40 G21 G17 G94 G80 (Safety default line) ;
G54 X? Y? S? M3 (First move setting Work co-ordinate system & Spindle R.P.M.) ;

**Program Number Setting**
O???? - 4 digit program number and starts with an O word.

**Tool change information**
T? – Tool/pocket number (any number above the available machine pocket numbers is recognised as a manual toolchange).
M6 - Tool change code.
; - EOB (End of block)

**Text messages**
( ) - Program message

**Program Defaults (Set by the programmer as required)**
G00 - Maximum Rapid Traverse of the machine
G90 - Absolute Co-ordinates taken from Datum set position
G40 - Cutter Compensation Cancel (Cutter follows program centreline path).
G21 - Metric Dimensions. (G20 = Imperial Dimensions).
G17 - X & Y Plane (Tool is in the Z axis - Spindle)
G94 - Programmed feed is Feed/min.(G95 = Feed/rotation)
G80 - Canned Cycle Cancel.

**Initial Start**
G54 - Work co-ordinate system.
X? - X axis start position.
Y? - Y axis start position.
S? - Spindle speed.
M3 - Spindle Clockwise rotation.
Linear Interpolation

The axis of the machine will move at either “Rapid” or “Feed” traverse rates. The rapid rates vary on all machine types.

When programming a straight line “Feed” (G01), with 2 axis (i.e X & Y), both axis will arrive at their programmed destination at the same time, irrespective of their length of motion, creating an angled motion. If one axis has to travel further than the other axis then this axis will move at an automatically calculated slower feed than that programmed to allow both axis to arrive together.

When programming a straight line “Rapid” (G00), with 2 axis (i.e X & Y), both axis will arrive at their programmed destination at different times as both will complete their motion at machine rapid. If one axis has to travel a shorter distance than the other axis then this axis will arrive at its programmed destination before the other axis creating a “Dogleg Effect” as per the example below.

A maximum of 3 axis can be programmed in one BLOCK

Parameter 1401.1 = 1 To remove Dog-leg effect (21i & 18i controls)
Tool Length Offset (G43)

The tool length offset facility is used to set the new tool length & make adjustments in the programmed axis.

G43 - Applies tool length offset which is stored in the “Offset Setting” table, in the + direction and must be applied on a single axis motion.

An added “H” word (tool offset row number) adds the stored value from the length column + wear column to the single axis move to set tool length to the program.

e.g. G43 Z100 H?

Note:
There are 4 columns contained within the tool offset register type C.

<table>
<thead>
<tr>
<th>LENGTH (H word)</th>
<th>RADIUS (D word)</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image-url" alt="Table" /></td>
<td><img src="image-url" alt="Table" /></td>
</tr>
</tbody>
</table>

1 – 32 (99 – 200) Length storage rows
Geometry = Length of the tool
Wear = Trimming value

1 – 32 (99 – 200) Radius storage rows
Geometry = Radius of the tool
Wear = Trimming value
**Initial Start of Program**

1) The “Rapid Motion” towards the starting position of the workpiece will contain Absolute X & Y axis motion together with the required “Workpiece Co-ordinate System” G code, spindle speed (S?) and the required M code to start the spindle (M3, M4, M13, or M14).

2) The “Rapid Motion” towards the “Initial Check Height” will contain a Z axis motion only together with the required “Tool Length Set” G code (G43) and the appropriate length offset storage number (H???).

---

O1111

T? M6 (Tool change line) ;

G0 G90 G40 G21 G17 G94 G80 (Safety default line) ;

G54 X? Y? S? M3 (First move setting Work co-ordinate system & Spindle R.P.M.) ;

G43 Z? H? (Set Tool Length) ;
Point to Point - example 1
(no compensation)

100 mm SQR.
80 mm SQR.

PLAN

SIDE

Start Point
Point to Point - example 2
(no compensation)

100 mm SQR.

Start Point

20 mm
Circular Programmed Movements

1) ARCS WITH A KNOWN RADIUS

G3 = Anti-Clockwise
G2 = Clockwise

Programmed Path

Note:
All arc movements where a radius of arc is specified in the line of program potentially have two arcs between the programmed endpoints. One arc will be greater than 180° and the other will be equal to or less than 180°.

When programming arcs, the created line of program uses the designated motion code (G02 or G03) X & Y as the programmed endpoints and the letter "R" to assign the radius value to the movement.

To specify the required arc, the following applies:

- 0.001° to 180.000° = R+
- 180.001° to 359.999° = R-

The “R” word can only be used with open Arcs up to 359.999°.
Arc Interpolation (G02/G03 - R?)

The information required to move in an arc involves the following “Word” addresses:

**G17 Example:**

```
G02 X100 Y0 R100
```

Where:
- G02 = Modal Clockwise Feed Motion.
- X100 = Radius end point in X axis.
- Y0 = Radius end point in Y axis.
- R100 = Radius of Arc (0.001° - 180.000° arc motion = R+)

```
G03 X0 Y-100 R-100
```

Where:
- G03 = Modal Counter Clockwise Feed Motion.
- X0 = Radius end point in X axis.
- Y-100 = Radius end point in Y axis.
- R-100 = Radius of Arc (180.001° - 359.999° arc motion = R-)
Circular Programmed Movements – e.g. 1
(Absolute)

1. (Absolute Rapid XY to point 1)
   (Clockwise to point 2)

2. (Absolute Rapid XY to point 3)
   (Anti-Clockwise to point 4)

3. (Absolute Rapid XY to point 5)
   (Clockwise to point 6)

4. (Absolute Rapid XY to point 7)
   (Anti-Clockwise to point 8)
Circular Programmed Movements – e.g. 2
(Absolute)

1. (Absolute Rapid XY to point 1)
2. (Clockwise to point 2)
3. (Absolute Rapid XY to point 2)
4. (Anti-Clockwise to point 1)
Circular Programmed Movements – e.g. 3

1 G0 X0 Y50
1 2
2 3
3 4
4 5
5 6
6 7
7 8
8 9
9 1
**Circular Programmed Movements – e.g. 4**

(Incremental)

---

1. (Absolute Rapid XY to point 1)  
   (Incremental Clockwise to point 2)

2. (Absolute Rapid XY to point 3)  
   (Incremental Anti-Clockwise to point 4)

3. (Absolute Rapid XY to point 5)  
   (Incremental Clockwise to point 6)

4. (Absolute Rapid XY to point 7)  
   (Incremental Anti-Clockwise to point 8)

*Note:*  
All Incremental values are taken “FROM” the programmed start point “TO” the programmed endpoint.
Circular Interpolation (G02/G03 - I?/J?/K?)

The control can also produce arcs or full circles in any of the 3 planes (G17 plan view, G18 front view and G19 side view). The program line contains the end points and the circle centre positions in all the relevant axis. Since the program line cannot contain duplicate information i.e. X? Y? for the endpoint and X? Y? for the circle centre, the control recognises other “Words” for the circle centre axis information. These are, in the relevant planes as follows:

**G17 Plan View**
- I = X axis circle centre
- J = Y axis circle centre

**G18 Front View**
- I = X axis circle centre
- K = Z axis circle centre

**G19 Side View**
- J = Y axis circle centre
- K = Z axis circle centre

**Note**
The axis information I, J & K are incremental values taken "FROM" the arc/circle starting point "TO" the arc/circle centre position.
Circular Programmed Movements

Circular Interpolation (G02/G03 - I?/J?/K?)

The information required to move in an arc using the arc centre involves the following “Word” addresses:

\[
\text{G02 X82 Y68 I27 J-42 F?}
\]

Where:
- G02 = Modal Clockwise Feed Motion.
- X82 = Arc end point in X axis.
- Y68 = Arc end point in Y axis.
- I27 = X axis Incremental distance from arc start point to Circle centre.
- J-42 = Y axis Incremental distance from arc start point to Circle centre.
- F? = Feedrate

Note:
The motions in any other machining plane (G18/G19) will require the Circle centre position for the Z axis. In this case the “Word” used to denote the Circle centre position for the Z axis is “K” and is again the incremental distance from the tool start point to the Circle centre.
Circular Programmed Movements – e.g. 6
X & Y (Absolute), I & J (Incremental)

(Absolute Rapid XY to point 1)
(Clockwise to point 2)

Note:
All Incremental Circle Centre values are taken “FROM” the programmed start point “TO” the Circle Centre.

“I” = X axis circle centre position
“J” = Y axis circle centre position
Circular Programmed Movements – e.g. 7
X & Y (Absolute), I & J (Incremental)

Note:
All Incremental Circle Centre values are taken “FROM” the programmed start point “TO” the Circle Centre.

“I” = X axis circle centre position
“J” = Y axis circle centre position
Circular Programmed Movements – e.g. 8
X & Y, I & J (Incremental)

Note:
All Incremental Circle Centre values are taken “FROM” the programmed start point “TO” the Circle Centre.

“T” = X axis circle centre position
“J” = Y axis circle centre position
Circular Programmed Movements – e.g. 9
X & Y, I & J (Incremental)

Note:
All Incremental Circle Centre values are taken “FROM” the programmed start point “TO” the Circle Centre.

“\(I\)” = X axis circle centre position
“\(J\)” = Y axis circle centre position

(Absolute Rapid XY to point 2)
(Anti-Clockwise to point 1)
Full Circular Movements – e.g. 10

I & J (Incremental)

*Note:
The end points are taken from the job datum and the circle centre positions are “Incremental” from the start point 1.

*Note:
The end points and circle centre positions are taken from the “Start point”
Full Circular Movements – e.g.11

By selecting a pole point of a circle (12, 3, 6 or 9 o’clock position) and using an “Incremental line of program” to create a full circle, all values on this line of program will have a zero value except for the I or J axis on the appropriate pole axis which will represent the radius to be produced:

i.e. G91 G2 X0 Y0 I0 J-50

Since the I & J are already incremental the G91 is active on the X & Y values only. If starting from a pole axis, the only axis that needs programming is the pole axis that represents the radius.

i.e. G2 J-50

Create the line of circle program for each of the following quadrant points in the diagrams below using “Clockwise”. 

23.456 R
56.987 R
16.159 R
120.357 R
Programmable Cutter Radius Compensation

Programs can be created in a way that allows adjustments to be made to create a part within tolerance dimensions. Once the program has been created and the part machined, any adjustments can be made by adjusting the tool radius or tool length, which are stored in the offset tables.

Programs are created in two ways. **“Job Path”** where the programmer creates the program using exact dimensions used on the part drawing or **“Cutter Path”** where the programmer creates a program adding to each dimension the radius of the milling cutter to be used.

“Job Path” automatically adds the radius of the tool which is stored in the tool offset table, during program running and so makes programming very simple. “Cutter path” stills uses information taken from the tool offset table but this information is usually a trimming value of the original programmed radius to make adjustments to the finished part size since the radius is already added to the program dimensions. Adding or subtracting the tool radius to every dimension can make creating the program very long and difficult.

The program line of information contains a modal “G” code (G41 / G42) which determines the offset side of the program path, an axis motion and a modal “D” word with a numerical value to indicate the row number of the offset table where the RADIUS value is stored:

i.e. G41 X100 D5

---

Program Path

Radius Left
G41

Radius Right
G42

No Compensation
G40

---

Programmed Radius Compensation has 3 modes:

- **G40** – No compensation so the path is directly over the program path.
- **G41** – Compensation is to the left of the programmed path (Climb Milling) *Best method
- **G42** – Compensation is to the right of the programmed path (Conventional Milling)

**Making machining adjustments**

To leave material on the contour or pocket “ADD” the offset value to the value in the radius offset “Wear” column.

To remove material on the contour or pocket “SUBTRACT” the offset value from the radius offset “Wear” column.
### Tool Offset Page for Compensation

Tool information relative to the Length and Radius is stored in the “Offset Setting” menu as:

<table>
<thead>
<tr>
<th>NO.</th>
<th>GEOMETRY</th>
<th>WEAR</th>
<th>GEOMETRY</th>
<th>WEAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>002</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>003</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>004</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>005</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>006</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>007</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>008</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>009</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>010</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>011</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>012</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>013</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>014</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>015</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>016</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
</tbody>
</table>
Compensation types

There are three main ways of applying compensation to a feature. Graphically described below, these are:

1) Circle Tangent – used in roughing/semi-roughing & finishing passes.
2) Line Tangent – used in roughing/semi-roughing & finishing passes.
3) Line Normal – used in roughing/semi-roughing passes.
Simple Compensation Rules

Applying compensation

1) Move “Z” to its programmed depth position before compensation is applied.

2) Compensation is best activated on a single axis motion towards the machining feature i.e. G41 X? and perpendicular (90°) to the next axis motion which should be the opposite single axis motion to the one used to apply compensation.

Cancelling compensation

1) “DO NOT MOVE” “Z” until compensation has been cancelled.

2) Compensation is best cancelled on a single axis motion away from the machining feature i.e. G40 X? and perpendicular (90°) to the last axis motion which should be the opposite single axis motion to the one used to cancel compensation.
Programmable Cutter Radius Compensation

Dia. Offset = 25mm  Depth = 25mm

Program

| G54 X-75 Y25 | X-75 Y25 |
| G43 Z5 H?   | X-75 Y25 Z5 |
| G1 Z-25 F?  | X-75 Y25 Z-25 |
| G41 Y0 D?   | X-75 Y12.5* Z-25 |
| X75         | X75 Y12.5* Z-25 |
| G40 Y25     | X75 Y25 Z-25 |
| G0 G90 Z100 | X75 Y25 Z100 |
Compensation e.g. 1

Diagram showing a series of points P1 to P8 with distances and angles labeled. The points are connected to form a path with a specified radius and start and end points marked.

- P1 to P8
- Distances: 57, 9.694, 50, 90, 8, 62, 16, 80 Rad.
- 80 Rad.
- Start Point: P8
- End Point: P5

Diagram dimensions: 70, 62, 8, 9, 694, 50, 90, 100, 54, 16, 5, 5.
Compensation e.g. 2

<table>
<thead>
<tr>
<th>P1</th>
<th>P2</th>
<th>P3</th>
<th>P4</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>61.11</td>
<td>29.738</td>
<td>16.257</td>
</tr>
<tr>
<td>Y</td>
<td>30.031</td>
<td>21.194</td>
<td>34.854</td>
</tr>
</tbody>
</table>
Programmable Cutter Radius Compensation

Circle Tangent inside a full Circle

Main C/Bore = 50mm Radius

CC (40 Rad.)

10

CC (50 Rad.)

Main Radius = AR
Arc ON/OFF Radius = SR
CC Difference = YD

Note:
Make the Approach & Departure Arc a value less than the original radius to be produced, greater than the cutter radius being used, a radius value which can be subtracted from the original arc to leave a whole number for the “CC Difference” and attached to one of the pole points as the example above.

i.e.
SR = Radius less than AR
YD = AR – SR (i.e. 121.946 – 101.946 = 20)

As above graphical example
AR = 50
SR = 40
YD = 50 - 40 = 10
Programmable Cutter Radius Compensation

Circle Tangent

1st Process

2nd Process

3rd Process

4th Process

5th Process

6th Process

Final Process

O1000;
T? M6 (Toolchange line - 25mm Endmill cutter);
G0 G90 G40 G21 G17 G94 G80 (Safety default line);
G54 X0 Y0 S? M3 (Absolute Start Point – Centre of Actual radius – position 1);
G43 Z5 H?? (Rapid to a position above material setting length offset);
G1 Z.? F? (Feed to required cut depth before compensation has been applied.);
G91 Y(YD) (Incremental move to centre of Arc on/off as calculated – position 2);
G41 X(SR) D?? M8 (Move to point 3 - Apply compensation incrementally);
G3 X-(SR) Y(SR) R(SR) (Move to position 4 - Arc On.as SR Rad.);
X0 Y0 I0 J-(AR) (Move 360 Degrees back to position 4 by Radius of AR);
X-(SR) Y-(SR) R(SR) (Move to position 5 - Arc Off as SR Rad.)
G1 G40 X(SR) (Move to start position cancelling compensation);
G0 G90 Z100 (Move to Absolute safe height above material after comp is cancelled);
M30 (End program);
Programmable Cutter Radius Compensation

Circle Tangent

O1000;
T? M6  (Toolchange line - 25mm Endmill cutter);
(Internal circular contour - Arc on / Arc off);
G0 G90 G40 G21 G17 G94 G80 (Safety default line);
G54 X0 Y0 S? M3 (Absolute Start Point – Centre of Actual radius – position 1);
G43 Z5 H?? (Rapid to a position above material setting length offset);
G1 Z.? F? (Feed to required cut depth before compensation has been applied.);
G91 Y10 (Incremental move to centre of Arc on/off as calculated – position 2);
G41 G91 X40 D?? M8 (Move to point 3 - Apply compensation incrementally);
G3 X-40 Y40 R40 (Move to position 4 - Arc On.as SR Rad.);
X0 Y0 I0 J-50 (Move 360 Degrees back to position 4 by Radius of AR);
X-40 Y-40 R40 (Move to position 5 - Arc Off as SR Rad.)
G1 G40 X40 D0 (Move to start position cancelling compensation);
G0 G90 Z100 (Move to Absolute safe height above material after comp is cancelled);
M30 (End program);
Circle Tangent Compensation e.g. 1

80 R

? R

?
**Helical Milling**

The pitch is programmed on the line of information requiring “X”, “Y”, and “Z” moves with circular programming.

i.e. **If the Pitch = 2mm**

One full circular movement = 360

\[
\text{1 FULL PITCH} \quad Z \text{ incremental movement } = 2
\]

Half of a full circular movement = 180

\[
\text{½ FULL PITCH} \quad Z \text{ incremental movement } = 1\text{mm}
\]

Quarter of a full circular movement = 90

\[
\text{¼ FULL PITCH} \quad Z \text{ incremental movement } = 0.5\text{mm}
\]

If the Lead is 2mm and the arc is 90 then the Z move will be a ¼ of 2mm = 0.5mm.

This value is then added or subtracted to the Z absolute positioning move at the lead on and the lead off the helix.

If incremental programming is used then the Z word would be the pitch relative to the angle of motion.

The line of program could look like this:

\[
\]

or

\[
\text{G2 X? Y? Z? R?}
\]
FANUC I.S.O.PROGRAMMING NOTES
Chapter 7

Helical Milling
“Single Point Tools”
Absolute Programming

O1000 ;
T1 M6 (Tool change line.) ;
G0 G90 G40 G21 G17 G94 G80  (Safety Line.) ;
G54 X0 Y0 S? M3 (Move to centreline of bore) ;
G43 Z100 H? (Set tool length) ;
Z5 (Move to bore top) ;
G1 Z-38.75 F? (Feed to depth + ¼ of pitch for arcing ON + nose width/2.) ;
Y5 (Move to Arc On bore centre 26(AR) – 21(SR) = 5(YD)) ;
G41 X21 D?  (Apply compensation as a straight line.) ;
G3 X0 Y26 Z-37.5 R21 (This line creates a ¼ arc + Z movement of a ¼ of pitch) ;
X0 Y26 Z32.5 I0 J-26 (This line will create 1\textsuperscript{st} pitch of 5mm.) ;
X0 Y26 Z27.5 I0 J-26 (This line will create 2\textsuperscript{nd} pitch of 5mm.) ;
X0 Y26 Z22.5 I0 J-26 (This line will create 3\textsuperscript{rd} pitch of 5mm.) ;
X0 Y26 Z17.5 I0 J-26 (This line will create 4\textsuperscript{th} pitch of 5mm.) ;
X0 Y26 Z12.5 I0 J-26 (This line will create 5\textsuperscript{th} pitch of 5mm.) ;
X0 Y26 Z7.5 I0 J-26 (This line will create 6\textsuperscript{th} pitch of 5mm.) ;
X0 Y26 Z2.5 I0 J-26 (This line will create 7\textsuperscript{th} pitch of 5mm.) ;
X-21 Y5 Z-2 R21 (This line creates a ¼ arc + Z movement of a ¼ of pitch) ;
G1 G40 X0 (Cancel Compensation as a straight line) ;
G0 G90 Z100 (Clear the workpiece) ;
M30 (End the Program) ;
Helical Milling
“Single Point Tools”
Absolute & Incremental Programming

O1000 ;
T1 M6 (Tool change line.) ;
G0 G90 G40 G21 G17 G94 G80 (Safety Line.) ;
G54 X0 Y0 S? M3 (Move to centreline of bore) ;
G43 Z100 H? (Set tool length) ;
Z5 (Move to bore top) ;
G1 Z-38.75 F? (Feed to depth + ¼ of pitch for arcing ON + nose width/2.) ;
G91 Y5 (Move to Arc On bore centre 26(AR) – 21(SR) = 5(YD) incrementally) ;
G41 X21 D? (Apply compensation as a straight line.) ;
G3 X-21 Y21 Z1.25 R21 (This line creates a ¼ arc + Z movement of a ¼ of pitch) ;
* Z5 J-26 (This incremental line will create 1ⁿ pitch of 5mm.) ;
* Z5 J-26 (This incremental line will create 2ⁿ pitch of 5mm.) ;
* Z5 J-26 (This incremental line will create 3ⁿ pitch of 5mm.) ;
* Z5 J-26 (This incremental line will create 4ⁿ pitch of 5mm.) ;
* Z5 J-26 (This incremental line will create 5ⁿ pitch of 5mm.) ;
* Z5 J-26 (This incremental line will create 6ⁿ pitch of 5mm.) ;
* Z5 J-26 (This incremental line will create 7ⁿ pitch of 5mm.) ;
X-21 Y-21 Z1.25 R21 (This line creates a ¼ arc + Z movement of a ¼ of pitch) ;
G1 G40 X21 (Cancel Compensation as a straight line) ;
G0 G90 Z100 (Clear the workpiece) ;
M30 (End the Program) ;
Helical Milling
“Multi toothed Tools”
Absolute Programming

*Produce a Thread M60 x 2mm pitch (60mm Deep)*

O1000;
T1 M6 (Tool change line.);
G0 G90 G40 G21 G17 G94 G80 (Safety Line.);
G54 X0 Y0 S? M3 (Move to centreline of bore);
G43 Z100 H? (Set tool length);
Z5 (Move to feed clearance);
G1 Z-61.5 F? (Feed to depth + ¼ of pitch for arcing ON + ½ tooth form width to tool end);
Y10 (Move to Arc On bore centre 30(AR) – 20(SR) = 10(YD));
G41 X20 D? (Apply compensation as a straight line);
G3 X0 Y30 Z-61 R20 (This line creates a ¼ arc + Z movement of a ¼ pitch);
Z-59 J-30 (This line will create 1 pitch.);
X-20 Y10 Z-58.5 R20 (Creates a ¼ arc + Z movement of a ¼ of pitch);
G1 G40 X0 (Cancel Compensation as a straight line);
G0 Z-41.5 (Subtract edge length from 1st Z positioning move 61.5 – 20 = 41.5);
G1 G41 X20 D? (Apply compensation as a straight line);
G3 X0 Y30 Z-41 R20 (This line creates a ¼ arc + Z movement of a ¼ pitch);
Z-39 J-30 (This line will create 1 pitch.);
X-20 Y10 Z-38.5 R20 (Creates a ¼ arc + Z movement of a ¼ of pitch);
G1 G40 X0 (Cancel Compensation as a straight line);
G0 Z-21.5 (Subtract edge length from 2nd Z positioning move 41.5 – 20 = 21.5);
G41 X20 D? (Apply compensation as a straight line);
G3 X0 Y30 Z-21 R20 (This line creates a ¼ arc + Z movement of a ¼ pitch);
Z-19 J-30 (This line will create 1 pitch.);
X-20 Y10 Z-18.5 R20 (Creates a ¼ arc + Z movement of a ¼ of pitch);
G1 G40 X0 (Cancel Compensation as a straight line);
G0 G90 Z100 (Clear the workpiece);
M30 (End the Program.) ;
Helical Milling
“Multi toothed Tools”
Absolute & Incremental Programming

*Produce a Thread M60 x 2mm pitch (60mm Deep)

O1000;
T1 M6 (Tool change line.) ;
G0 G90 G40 G21 G17 G94 G80 (Safety Line.) ;
G54 X0 Y0 S? M3 (Move to centreline of bore) ;
G43 Z100 H? (Set tool length) ;
Z5 (Move to feed clearance) ;
G1 Z-61.5 F? (Feed to depth + ¼ of pitch for arcing ON + ½ tooth form width to tool end) ;
G91 Y10 (Move to Arc On bore centre 30(AR) – 20(SR) = 10(YD)) ;
  • G1 G41 X20 D? (Apply compensation as a straight line) ;
  • G3 X-20 Y20 Z0.5 R20 (This line creates a ¼ arc + Z movement of a ¼ pitch) ;
  • Z2 J-30 (This line will create 1 pitch.) ;
  • X-20 Y-20 Z0.5 R20 (Creates a ¼ arc + Z movement of a ¼ of pitch) ;
  • G1 G40 X20 (Cancel Compensation as a straight line) ;
G0 G90 Z-41.5 (Subtract edge length from 1st Z positioning move 61.5 – 20 = 41.5) ;
  • G1 G41 X20 D? (Apply compensation as a straight line) ;
  • G3 X-20 Y20 Z0.5 R20 (This line creates a ¼ arc + Z movement of a ¼ pitch) ;
  • Z2 J-30 (This line will create 1 pitch.) ;
  • X-20 Y-20 Z0.5 R20 (Creates a ¼ arc + Z movement of a ¼ of pitch) ;
  • G1 G40 X20 (Cancel Compensation as a straight line) ;
G0 G90 Z-21.5 (Subtract edge length from 2nd Z positioning move 41.5 – 20 = 21.5) ;
  • G41 X20 D? (Apply compensation as a straight line) ;
  • G3 X-20 Y20 Z0.5 R20 (This line creates a ¼ arc + Z movement of a ¼ pitch) ;
  • Z2 J-30 (This line will create 1 pitch.) ;
  • X-20 Y-20 Z0.5 R20 (Creates a ¼ arc + Z movement of a ¼ of pitch) ;
  • G1 G40 X20 (Cancel Compensation as a straight line) ;
G0 G90 Z100 (Clear the workpiece) ;
M30 (End the Program.) ;
**Canned Cycles**

The control has the ability to machine holes using a series of “G” codes for different cycles. These are simple drilling, peck drilling, tapping and boring cycles. A basic line of program consists of modal words all containing numerical values as:

\[
\]

Where:

- G? = Cycle machining code.
- G? = Code to determine action at end of cycle (see G98 / G99 next page)
- X & Y = Absolute hole centre position.
- Z? = Absolute Z position at bottom of hole.
- R? = Z axis starting position above surface.
- F? = Feedrate.
- M? = Coolant M8 or M38.

### Absolute vs Incremental Mode

<table>
<thead>
<tr>
<th>Absolute</th>
<th>Incremental</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Absolute Mode Diagram" /></td>
<td><img src="image" alt="Incremental Mode Diagram" /></td>
</tr>
</tbody>
</table>

**Incremental Mode**

When using G91 on any Hole Canned Cycle the “R” value is the incremental distance from the “Initial Height” and the “Z” is the incremental distance from the “R” word.

**Cancel Canned Cycle**

**G80**

**Group 01 Codes**

The following G codes also are effective in cancelling any Hole Canned Cycle:

G0, G01, G02, G03
"G98" & "G99"

The G98 / G99 action code determines the final Z axis position after hole completion.

The actions are:
G98 = Return the tool to the last Z axis program position before the cycle line.
G99 = Return the tool point to the programmed “R” position on the cycle line.

* The machine default for G98/G99 is set to G98 *

A basic canned cycle always follows a sequence of four operations:
1) Rapid traverse X & Y axis to hole centre position.
2) Rapid traverse down to “R” position.
3) Feed to Z absolute depth.
4) Rapid to “R” position or Initial level.
Hole Canned Cycles
“G98/G99” & “R” Positions

Initial Plane (G98) Z45

Section View YZ plane
All holes 20mm Deep

(Set Initial point) Z45
(Point 1) G99 X? Y? R23 Z0 F?
(Point 2) G98 X? Y? R3 Z-20
(Point 3) G99 X? Y? R43 Z20
(Point 4) G98 X? Y? R-37 Z-60
(Point 5) (G98) X? Y? R3 Z-20
(Point 6) (G98 or G99) X? Y? R20 Z0
Hole Canned Cycles
“G98/G99” & “R” Positions

Initial Plane (G98) Z?

Section View YZ plane
All holes 20mm Deep

(Set Initial point)
(Point 1)
(Point 2)
(Point 3)
(Point 4)
(Point 5)
(Point 6)
High Speed Peck Drilling

G73


X? = Modal hole centre position
Y? = Modal hole centre position
Z? = Modal absolute hole depth position
R? = Modal tool starting position above hole surface
Q? = Modal incremental depth of cut for each peck
F? = Modal cutting feedrate

[ ] denotes optional input for the first hole.

<table>
<thead>
<tr>
<th>G73 (G98)</th>
<th>G73 (G99)</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Diagram" /></td>
<td><img src="image2.png" alt="Diagram" /></td>
</tr>
</tbody>
</table>

S1000 M3 ;
X? ;
Y? ;
G0 G90 G80 Z100 ;

- Spindle start.
- Position to 1st hole setting all data.
- Position to 2nd hole.
- Position to 3rd hole.
- Tool to a safe height (G80 cancel)

This cycle creates a peck at the programmed pecking value (Q?) with a short chip break retraction at a value as set in parameter 5114 before creating the next peck at value Q?.

At Z finish position the tool retracts automatically.
Left Hand Tapping

G74

G74 [G98 or G99 X? Y?] Z? R? [Q?] F?

X? = Modal hole centre position
Y? = Modal hole centre position
Z? = Modal absolute hole depth position
R? = Modal tool starting position above hole surface
Q? = Modal incremental depth of cut for each peck
F? = Modal cutting feedrate
[ ] denotes optional input for the first hole.

<table>
<thead>
<tr>
<th>G74 (G98)</th>
<th>G74 (G99)</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Diagram" /></td>
<td><img src="image2.png" alt="Diagram" /></td>
</tr>
</tbody>
</table>

S1000 ; - Spindle speed.
X? ; - Position to 2nd hole.
Y? ; - Position to 3rd hole
G0 G90 G80 Z100 ; - Tool to a safe height (G80 cancel)

This cycle can create a peck at a value of Q? if programmed with either a short chip break retraction or a full retraction as set in parameter 5200 #5 (1 = full retract).

At Z finish position the tool retracts at twice speed/feed automatically.

Adding G95 (feed/revolution) on the line of program, the feedrate = the pitch of tap.
i.e. M8 x 1.25 pitch tap

Ensure the next tool is set back to G94 (feed per minute) if programming feed per minute.
Fine Boring

G76


X? = Modal hole centre position
Y? = Modal hole centre position
Z? = Modal absolute hole depth position
R? = Modal tool starting position above hole surface
Q? = Modal incremental axis shift off centreline.
F? = Modal cutting feedrate

[ ] denotes optional input for the first hole.

<table>
<thead>
<tr>
<th>G76 (G98)</th>
<th>G76 (G99)</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Diagram" /></td>
<td><img src="image2" alt="Diagram" /></td>
</tr>
</tbody>
</table>

S1000 M3 ;  - Spindle start.
X? ;  - Position to 2nd hole.
Y? ;  - Position to 3rd hole
G0 G90 G80 Z100 ;  - Tool to a safe height (G80 cancel)

This cycle is a finish boring cycle and will shift off bore centre line at depth after the spindle stops and orientates to the toolchange angle. The shift is dependent on parameter 5101 as:

<table>
<thead>
<tr>
<th>Bit Number</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>X+</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>X-</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Y+</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Y-</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

At Z finish position the tool retracts automatically.
Drilling

**G81**

G81 [G98 or G99 X? Y?] Z? R? F?

- **X?** = Modal hole centre position
- **Y?** = Modal hole centre position
- **Z?** = Modal absolute hole depth position
- **R?** = Modal tool starting position above hole surface
- **F?** = Modal cutting feedrate

[ ] denotes optional input for the first hole.

<table>
<thead>
<tr>
<th>G81 (G98)</th>
<th>G81 (G99)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Level</td>
<td>R Level</td>
</tr>
</tbody>
</table>

S1000 M3 ; - Spindle start.
X? ; - Position to 2nd hole.
Y? ; - Position to 3rd hole
G0 G90 G80 Z100 ; - Tool to a safe height (G80 cancel)

At Z finish position the tool retracts automatically.
**Drilling**

**G82**


- **X?** = Modal hole centre position
- **Y?** = Modal hole centre position
- **Z?** = Modal absolute hole depth position
- **R?** = Modal tool starting position above hole surface
- **P?** = Dwell time in milliseconds (1sec. = P1000)
- **F?** = Modal cutting feedrate

[ ] denotes optional input for the first hole.

<table>
<thead>
<tr>
<th>G82 (G98)</th>
<th>G82 (G99)</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Diagram of G82 (G98)" /></td>
<td><img src="image2" alt="Diagram of G82 (G99)" /></td>
</tr>
</tbody>
</table>

**S1000 M3 ;**
- Spindle start.

- Position to 1st hole setting all data.

**X? ;**
- Position to 2nd hole.

**Y? ;**
- Position to 3rd hole

**G0 G90 G80 Z100 ;**
- Tool to a safe height (G80 cancel)

*At Z finish position the tool retracts automatically.*
Peck Drilling

G83


X? = Modal hole centre position  
Y? = Modal hole centre position  
Z? = Modal absolute hole depth position  
R? = Modal tool starting position above hole surface  
Q? = Modal incremental depth of cut for each peck  
F? = Modal cutting feedrate

[ ] denotes optional input for the first hole.

<table>
<thead>
<tr>
<th>G83 (G98)</th>
<th>G83 (G99)</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="Diagram1.png" alt="Diagram" /></td>
<td><img src="Diagram2.png" alt="Diagram" /></td>
</tr>
</tbody>
</table>

S1000 M3 ;  
X? ;  
Y? ;  
G0 G90 G80 Z100 ;  
- Spindle start.  
- Position to 1st hole setting all data.  
- Position to 2nd hole.  
- Position to 3rd hole  
- Tool to a safe height (G80 cancel)

This cycle creates a peck at the programmed pecking value (Q?) with a full chip break retraction back to R position. The tool repositions itself into the hole and stops above the last peck position at a value as set in parameter 5115 before creating the next peck at value Q?.

At Z finish position the tool retracts automatically.
Right Hand Tapping

G84

G84 [G98 or G99 X? Y?] Z? R? [Q?] F?

X? = Modal hole centre position  
Y? = Modal hole centre position  
Z? = Modal absolute hole depth position  
R? = Modal tool starting position above hole surface  
Q? = Modal incremental depth of cut for each peck  
F? = Modal cutting feedrate  
[ ] denotes optional input for the first hole.

G84 (G98)  

G84 (G99)  

S1000 ; - Spindle speed.  
X? ; - Position to 2nd hole.  
Y? ; - Position to 3rd hole  
G0 G90 G80 Z100 ; - Tool to a safe height (G80 cancel)  

This cycle can create a peck at a value of Q? if programmed with either a short chip break retraction or a full retraction as set in parameter 5200 #5 (1 = full retract).  

At Z finish position the tool retracts at twice speed/feed automatically.

Adding G95 (feed/revolution) on the line of program, the feedrate = the pitch of tap.  
i.e. M8 x 1.25 pitch tap  

Ensure the next tool is set back to G94 (feed per minute) if programming feed per minute.
Boring

G85


X? = Modal hole centre position
Y? = Modal hole centre position
Z? = Modal absolute hole depth position
R? = Modal tool starting position above hole surface
F? = Modal cutting feedrate

[ ] denotes optional input for the first hole.

<table>
<thead>
<tr>
<th>G85 (G98)</th>
<th>G85 (G99)</th>
</tr>
</thead>
</table>

S1000 M3 ;  
X? ;  
Y? ;  
G0 G90 G80 Z100 ;

- Spindle start.
- Position to 1st hole setting all data.
- Position to 2nd hole.
- Position to 3rd hole
- Tool to a safe height (G80 cancel)

This cycle feeds to the programmed depth position and will then retract back out of the hole at the same speed/feedrate.

At Z finish position the tool retracts automatically.
**Boring**

**G86**

G86 [G98 or G99 X? Y?] Z? R? F?

- X? = Modal hole centre position
- Y? = Modal hole centre position
- Z? = Modal absolute hole depth position
- R? = Modal tool starting position above hole surface
- F? = Modal cutting feedrate

[ ] denotes optional input for the first hole.

<table>
<thead>
<tr>
<th>G86 (G98)</th>
<th>G86 (G99)</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Diagram" /></td>
<td><img src="image2.png" alt="Diagram" /></td>
</tr>
</tbody>
</table>

S1000 M3 ;
X? ;
Y? ;
G0 G90 G80 Z100 ;

- Spindle start.
- Position to 1st hole setting all data.
- Position to 2nd hole.
- Position to 3rd hole
- Tool to a safe height (G80 cancel)

This cycle feeds to the programmed depth position and will then retract back out of the hole at rapid feedrate.

**At Z finish position the tool retracts automatically.**
Back Boring

G87


X? = Modal hole centre position
Y? = Modal hole centre position
Z? = Modal absolute hole depth position
R? = Modal tool starting position above hole surface
Q? = Modal incremental axis shift off centreline.
P? = Dwell time in milliseconds (1sec. = P1000)
F? = Modal cutting feedrate

[ ] denotes optional input for the first hole.

G87 (G98)

- Spindle start.
- Position to 1st hole setting all data.
- Position to 2nd hole.
- Position to 3rd hole
- Tool to a safe height (G80 cancel)

This cycle creates a machining cycle as counterbores or chamfers at the bottom of a hole. The tool positions itself above the hole with the spindle at orientation position, rapids thro’ hole off centreline, repositions on centreline with the spindle revolving to allow feed motion up. Opposite sequence to above for retraction.

See G76 parameter shift notes
At Z finish position the tool retracts automatically.
Boring
G88


X? = Modal hole centre position
Y? = Modal hole centre position
Z? = Modal absolute hole depth position
R? = Modal tool starting position above hole surface
P? = Dwell time in milliseconds (1sec. = P1000)
F? = Modal cutting feedrate

[ ] denotes optional input for the first hole.

<table>
<thead>
<tr>
<th>G88 (G98)</th>
<th>G88 (G99)</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Diagram" /></td>
<td><img src="image2.png" alt="Diagram" /></td>
</tr>
</tbody>
</table>

S1000 M3 ;  
X? ;  
Y? ;  
G0 G90 G80 Z100 ;

- Spindle start.
- Position to 1st hole setting all data.
- Position to 2nd hole.
- Position to 3rd hole.
- Tool to a safe height (G80 cancel)

This cycle will automatically stop the spindle at Z position after dwell. The tool is then manually retracted out of the hole using the handwheel or power feeds. At the requested return position the spindle will restart and rapid motion is performed to the next position for cycle.
Boring
G89


- X? = Modal hole centre position
- Y? = Modal hole centre position
- Z? = Modal absolute hole depth position
- R? = Modal tool starting position above hole surface
- P? = Dwell time in milliseconds (1sec. = P1000)
- F? = Modal cutting feedrate

[ ] denotes optional input for the first hole.

<table>
<thead>
<tr>
<th>G89 (G98)</th>
<th>G89 (G99)</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Diagram" /></td>
<td><img src="image2.png" alt="Diagram" /></td>
</tr>
</tbody>
</table>

S1000 M3 ; - Spindle start.
X? ; - Position to 2nd hole.
Y? ; - Position to 3rd hole
G0 G90 G80 Z100 ; - Tool to a safe height (G80 cancel)

This cycle feeds to the programmed depth position, dwells, then retract back out of the hole at the same speed/feedrate.

At Z finish position the tool retracts automatically.
Hole Example

Fixture View

---

...
**Repeats**

By using a numerical “K” word on an incremental X &or Y axis during a canned cycle allows the cycle to repeat (as the above example). (K0 = no Z axis action for the specified line).

*The maximum allowable repeats are 9999.*
Sub-Programming

The control provides the ability to access other part programs stored inside the main directory. If the Sub-program being called is stored in the main directory (stored as a normal program with the letter O as the header), then access is by the use of an M98 command followed by the sub-program number preceded with a letter P.

i.e. N10 M98 P1004

To enable the control to return to the last program position for the program to continue, then an M99 command on the last line of program in the sub-program will enable this.

i.e. N100 M99

Notes:
1) When a sub-program is being written, the letter O is still assigned to the program number.
2) When a sub-program is called, the letter P is assigned to the program number.
3) M99 can also be written at the end of a main program, and would result in a continuous program loop.

Sub-programs can be nested to a maximum of 4 levels as below:

Sub-Program repeats

The control also has the ability to contain a repeat command as part of the M98 program line. When the program line is written with the M98 P1004 command the control actually reads the line of information as M98 P00001004 , the first 4 digits after the P word being the repeat amount.

To repeat a sub-program (O1004) 33 times, the program line would read as follows:

i.e. M98 P331004

The control also has the ability to jump to a specific program line number on its return to the main program using the M99 command as:

i.e. M99 P100

This command above will move the control to line number N100 in the main program.
**Sub-Programming**

**“Normal Program”**

```
O1000;
T1 M6;
G0 G90 G40 G21 G17 G94 G80;
G54 X10 Y10 S? M3;
G43 Z100 H1;
Z5;
G81 R3 Z-20 F? M8;
Y30;
G0 G90 Z100;
X30;
G80;
Y50;
G0 G90 G40 G21 G17 G94 G80;
G54 X10 Y10 S? M3;
G43 Z100 H1;
X30;
Z5;
X50;
G84 G99 G95 R3 Z-20 F1.25 M8;
X70;
Y30;
Y50;
X90;
Y50;
T0 M6;
M30;
```
Sub-Programming
“Sub-Program version”

O1000 ;
N1 T1 M6 ;
N2 G0 G90 G40 G21 G17 G94 G80 ;
N3 G54 X10 Y10 S? M3 ;
N4 G43 Z100 H1 ;
N5 Z5 ;
N6 G81 R3 Z-20 F? M8 ;

**N7 M98 P1001 ;**
N8 G0 G90 Z100
N9 T2 M6 ;
N10 G0 G90 G40 G21 G17 G94 G80 ;
N11 G54 X10 Y10 S? M3 ;
N12 G43 Z100 H1 ;
N13 Z5 ;
N14 G84 G99 G95 R3 Z-20 F1.25 M8 ;

**N15 M98 P1001 ;**
N16 G0 G90 Z100 ;
N17 T0 M6 ;
N18 M30 ;

O1001 ;
N101 Y30 ;
N102 Y50 ;
N103 Y70 ;
N104 X30 ;
N105 X50 ;
N106 X70 ;
N107 X90 ;
N108 Y50 ;
N109 Y30 ;
N110 Y10 ;
N111 X70 ;
N112 X50 ;
N113 X30 ;
N114 G80 ;
N115 M99 (back to N7 & N15) ;

Section AA
Sub-Programming
“Contour pecking”

O4000 ;
T? M6 ;
G0 G90 G40 G21 G17 G94 G80 (Safety default line) ;
G54 X-75 Y-75 S? M3 (Absolute Start Point.) ;
G43 Z100 H? (Initial check height)
Z5 (Rapid to a position above material.) ;
G1 Z0 F? (Feed to surface top) ;
M98 P104001 (Call sub-program & repeat 10 times) ;
G0 G90 Z100 (Move to Safe height above material.) ;
M30 (End program leaving tool in the spindle ready for next load.)

O4001 ;
G1 G91 Z-2 (Incremental peck depth) ;
G90 G41 X-40 D? M8 (Move to position 1 with comp. - Switch on coolant) ;
Y40 (Move to position 2.) ;
X40 (Move to position 3.) ;
Y-40 (Move to position 4.) ;
X-75 (Move to position 5 - Clear of material - cutter diameter) ;
G40 Y-75 (Cancel compensation) ;
M99 ;
Sub-Programming
“Single Point Tools”
Absolute & Incremental Programming

O2000 ;
T1 M6 (Tool change line.) ;
G0 G90 G40 G21 G17 G94 G80 (Safety Line.) ;
G54 X0 Y0 S? M3 (Move to centreline of bore) ;
G43 Z100 H? (Set tool length) ;
Z5 (Move to bore top) ;
G1 Z-38.75 F? (Feed to depth + ¼ of pitch for arcing ON + nose width/2) ;
G91 Y5 (Move to Arc On bore centre 26(AR) – 21(SR) = 5(YD) incrementally) ;
G41 X21 D? (Apply compensation as a straight line.) ;
G3 X-21 Y21 Z1.25 R21 (This line creates a ¼ arc + Z movement of a ¼ of pitch) ;
M98 P72001 ;
X-21 Y-21 Z1.25 R21 (This line creates a ¼ arc + Z movement of a ¼ of pitch) ;
G1 G40 X21 (Cancel Compensation as a straight line) ;
G0 G90 Z100 (Clear the workpiece) ;
M30 (End the Program) ;

O2001 ;
Z5 J-26 (This incremental line will create 1 pitch of 5mm.) ;
M99 ;
Sub-Programming
“Multi toothed Tools”
Absolute & Incremental Programming

*Produce a Thread M60 x 2mm pitch (60mm Deep)

O3000 ;
T1 M6 (Tool change line.) ;
G0 G90 G40 G21 G17 G94 G80 (Safety Line.) ;
G54 X0 Y0 S? M3 (Move to centreline of bore) ;
G43 Z100 H? (Set tool length) ;
Z5 (Move to feed clearance) ;
G1 Z-61.5 F? (To depth + ¼ of pitch for arcing ON + ½ tooth form width to tool end) ;
M98 P3001 ;
G0 G90 Z-41.5 (Subtract edge length from 1" Z positioning move 61.5 – 20 = 41.5) ;
M98 P3001 ;
G0 G90 Z-21.5 (Subtract edge length from 2nd Z positioning move 41.5 – 20 = 21.5) ;
M98 P3001 ;
G0 G90 Z100 (Clear the workpiece) ;
M30 (End the Program.) ;

O3001 ;
G91 Y10 (Move to Arc On bore centre 30(AR) – 20(SR) = 10(YD)) ;
G1 G91 G41 X20 D? (Apply compensation as a straight line) ;
G3 X-20 Y20 Z0.5 R20 (This line creates a ¼ arc + Z movement of a ¼ pitch) ;
Z2 J-30 (This line will create 1 pitch.) ;
X-20 Y-20 Z0.5 R20 (Creates a ¼ arc + Z movement of a ¼ of pitch) ;
G1 G40 X20 (Cancel Compensation as a straight line) ;
Y-10 (Move to Main arc centre) ;
M99
G90 Absolute Example Programming

N1 G90 X-50 Y100
N2 X100
N3 X40 Y30
N4 X80 Y-30
N5 Y-60
N6 X-60 Y20
N7 X-70 Y-80
N8 X40
N9 X-40 Y-60
N10 X-90 Y-30
G91 Incremental Example Programming

N1 G90 X-50 Y100
N2 G91 X150
N3 X-60 Y-70
N4 X40 Y-60
N5 Y-30
N6 X-140 Y80
N7 X-10 Y-100
N8 X110
N9 X-80 Y20
N10 X-50 Y30
Point to Point - example 1
(no compensation)

O1000
T? M6
(Linear / Feed - Absolute) ;
G0 G90 G40 G21 G17 G94 G80
G54 X-75 Y-75 S? M3
G43 Z100 H?
Z5
G1 Z-20 F?
X-40
Y40 M8
X40
Y-40
X-75
G0 G90 Z100
M30
Point to Point - example 2
(no compensation)

O1000
T? M6
(Linear / Feed - Absolute) ;
G0 G90 G40 G21 G17 G94 G80
G54 X-75 Y-25 S? M3
G43 Z100 H?
Z5
G1 Z-20 F?
X-50 M8
Y0
X0 Y50
X50 Y0
X0 Y-50
X-50 Y0
Y25
X-75
G0 G90 Z100
M30
Circular Programmed Movements – e.g. 1

(Absolute)

1. G90 G0 X Y50
   G2 X50 Y0 R50

2. G90 G0 X-50 Y0
   G3 X0 Y-50 R50

3. G90 G0 X60 Y0
   G2 X0 Y60 R-60

4. G90 G0 X0 Y-70
   G3 X-70 Y0 R-70
Circular Programmed Movements – e.g. 2
(Absolute)

1. G90 G0 X-50 Y0
   G2 X50 R50

2. G90 G0 X50 Y0
   G3 X-50 R50
Circular Programmed Movements – e.g. 3

1 G0 X0 Y50
2 G1 X10
3 G2 X29.924 Y28.257 R20
4 Y-28.257 R-30
5 X10 Y-50 R20
6 G1 X-10
7 G2 X-29.924 Y-28.257 R20
8 Y28.257 R-30
9 X-10 Y50 R20
10 G1 X0
Circular Programmed Movements – e.g. 4

(Incremental)

1. G90 G0 X50 Y0
   G91 G2 X50 Y-50 R50

2. G90 G0 X-50 Y0
   G3 X-50 Y-50 R50

3. G90 G0 X60 Y0
   G2 X-60 Y60 R-60

4. G90 G0 X0 Y-70
   G3 X-70 Y70 R-70
Circular Programmed Movements – e.g. 5
X & Y (Absolute), I & J (Incremental)

G90 G0 X20 Y60
G2 X80 Y46 I18.93 J-37.23
Circular Programmed Movements – e.g. 6
X & Y (Absolute), I & J (Incremental)

G90 G0 X80 Y46
G3 X20 Y60 I-41.07 J-23.23
Circular Programmed Movements – e.g. 7
X & Y, I & J (Incremental)

G90 G0 X20 Y60
G91 G2 X60 Y-14 I18.93 J-37.23
Circular Programmed Movements – e.g. 8

X & Y, I & J (Incremental)

G90 G0 X80 Y46

G91 G3 X-60 Y14 I-41.07 J-23.23
Full Circular Movements – e.g. 9

I & J (Incremental)

G90 G0 X110 Y60
G2 X110 Y60 I-50 J0

*Note:
The end points are taken from the job datum and the circle centre positions are “Incremental” from the start point 1.

G90 G0 X110 Y60
G2 I-50

*Note:
The end points and circle centre positions are taken from the “Start point”
**Full Circular Movements – e.g. 10**

By selecting a pole point of a circle (12, 3, 6 or 9 o’clock position) and using an “Incremental line of program” to create a full circle, all values on this line of program will have a zero value except for the I or J axis on the appropriate pole axis which will represent the radius to be produced:

i.e. G91 G2 X0 Y0 I0 J-50

Since the I & J are already incremental the G91 is active on the X & Y values only. If starting from a pole axis, the only axis that needs programming is the pole axis that represents the radius.

i.e. G2 J-50

Create the line of circle program for each of the following quadrant points in the diagrams below using “Clockwise”.

<table>
<thead>
<tr>
<th>Axis</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>J</td>
<td>-23.456</td>
</tr>
<tr>
<td>I</td>
<td>-56.987</td>
</tr>
<tr>
<td>J</td>
<td>16.159</td>
</tr>
<tr>
<td>I</td>
<td>120.357</td>
</tr>
</tbody>
</table>

![Diagram showing full circular movements](image-url)
O1000
T1 M6
G0 G90 G40 G21 G17 G94 G80
G54 X125 Y35 S? M3
G43 Z100 H?
Z5
G1 Z-5 F?
G41 X90 D? M8
Y16
X50 Y8
X9.694
G2 Y62 R80
G1 X50
X90 Y54
Y35
G40 X125
G0 G90 Z100 M30
Compensation e.g. 2

```
O1000
T? M6
(Circular Arc Example) ;
G0 G90 G40 G21 G17 G94 G80
G54 X-50 Y-120 S? M3
G43 Z100 H?
Z5
G1 Z-20 F?
G41 Y70 D?
X60 M8
G2 X61.11 Y30.031 R20
G3 X29.738 Y21.194 R75
G2 X16.257 Y34.854 R10
X51.266 Y67.992 R80
X60 Y70 R20
G1 X150
G40 Y120
G0 G90 Z100
M30
```

<table>
<thead>
<tr>
<th></th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
<th>P4</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>61.11</td>
<td>29.738</td>
<td>16.257</td>
<td>51.266</td>
</tr>
<tr>
<td>Y</td>
<td>30.031</td>
<td>21.194</td>
<td>34.854</td>
<td>67.992</td>
</tr>
</tbody>
</table>
Circle Tangent Compensation e.g. 1

O1000
T? M6

(Internal circular contour - Arc on / Arc off) ;
G0 G90 G40 G21 G17 G94 G80
G54 X0 Y20 S? M3
G43 Z100 H?
Z5
G1 Z-? F?
G41 G91 X60 D? M8
G3 X-60 Y60 R60
J-80
X-60 Y-60 R60
G1 G40 X60
G0 G90 Z100
M30
Hole Canned Cycles
“G98/G99” & “R” Positions

Initial Plane (G98) Z?

Section View YZ plane
All holes 20mm Deep

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Z5</td>
<td>G99 R-17 Z-40</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>G98 R-37 Z-60</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>G99 R-3 Z-20</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>G98 R-77 Z-100</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>R-37 Z-60</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>R-17 Z-40</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Hole Example

```
O1000
T? M6
G0 G90 G40 G21 G17 G94 G80
G54 X0 Y-20 S? M3
G43 Z100 H?
Z15
G81 G98 R-2 Z-12 F? M8
X17.32 Y10
X-17.32
G80
G0 G90 Z100 M1
T? M6
G0 G90 G40 G21 G17 G94 G80
G54 X0 Y-20 S? M3
G43 Z100 H?
Z15
G84 G95 G98 R-2 Z-15 F1 M8
X17.32 Y10
X-17.32
G80
G0 G90 Z100 M30
```
Macro's
Macro Variables

Local variables - #1 to #33

Local variables can only be used within a custom macro (program or subprogram) to hold data such as the results of calculations.

When the power is turned off, all Local variables are initialised to null (nothing).

i.e.

#21 = 35
G1 X#21  Y#21

or

#21 = 35
#21 = #21 + 10

Note:
#0 = Null value

Common Variables - #100 to #149 (#199) / #500 to #532 (#999)

Common variables can be shared among different custom macros (program or subprogram) and again are used to hold data such as the results of calculations.
When the power is turned off, all #100 to #149 (#199) Common variables are initialised to null.
Variables #500 to #533 (#999) will hold data even when the power is turned off.

As an option, common variables #150 to #199 and #533 to #999 are also available.

System Variables - #1000 -

This is used as part of the computer's pre-set registers and are used to read and write a variety of NC data such as the current position and tool compensation values all of which can be found in the programming manual.

i.e..

#4109 = Current Feedrate
System Variables (#1000 - )

The following System Variables are the ones applicable to the current set-up of the machine configuration as by Cincinnati Machine.

### Tool compensation C

<table>
<thead>
<tr>
<th>Compensation Number</th>
<th>Tool Length (H)</th>
<th>Cutter Compensation (D)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Geometry</td>
<td>Wear</td>
</tr>
<tr>
<td>:</td>
<td>#11001 (#2201)</td>
<td>#10001 (#2001)</td>
</tr>
<tr>
<td>200</td>
<td></td>
<td></td>
</tr>
<tr>
<td>:</td>
<td>#11201 (#2400)</td>
<td>#10201 (#2200)</td>
</tr>
<tr>
<td>999</td>
<td>#11999</td>
<td>#10999</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Geometry</th>
<th>Wear</th>
</tr>
</thead>
<tbody>
<tr>
<td>#13001</td>
<td>#12001</td>
</tr>
<tr>
<td>#13999</td>
<td>#12999</td>
</tr>
</tbody>
</table>

### Macro Alarms

<table>
<thead>
<tr>
<th>Variable Number</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>#3000</td>
<td>When a value from 0 - 200 is assigned to variable #3000, the CNC stops with an alarm. After the expression, alarms message not longer than 26 characters can be described. The screen displays the alarm number by adding 3000 to the value in the variable #3000 along with the alarm message.</td>
</tr>
</tbody>
</table>

**Example:**

#3000 = 1(TOOL NOT FOUND) ;

The alarm screen displays “3001 TOOL NOT FOUND”

### Time Variables

<table>
<thead>
<tr>
<th>Variable Number</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>#3001</td>
<td>This variable functions as a timer that counts in 1 millisecond increments at all times. When the power is turned on, the value is of this variable is set to zero. When 2147483648 milliseconds is reached, the value of this timer returns to zero.</td>
</tr>
<tr>
<td>#3002</td>
<td>This variable functions as a timer that counts in 1-hour increments when the cycle start lamp is on. This timer preserves its value even when the power is turned off. When 9544.371767 hours is reached, the value of this timer returns to zero.</td>
</tr>
<tr>
<td>#3011</td>
<td>This variable can be used to read the current date (year/month/day). This is converted to a decimal number E.g. September 28, 1994 is represented as 19940928.</td>
</tr>
<tr>
<td>#3012</td>
<td>This variable can be used to read the current time (hours/minutes/seconds). This is converted to a decimal number E.g. 3:34:56 p.m. is represented as 153456.</td>
</tr>
</tbody>
</table>
Number of machined parts

<table>
<thead>
<tr>
<th>Variable Number</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>#3901</td>
<td>Number of completed machined parts</td>
</tr>
<tr>
<td>#3902</td>
<td>Number of required parts.</td>
</tr>
</tbody>
</table>

System Information

<table>
<thead>
<tr>
<th>Variable number</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>#4001</td>
<td>G0, G1, G2, G3, G33</td>
</tr>
<tr>
<td>#4002</td>
<td>G17, G18, G19</td>
</tr>
<tr>
<td>#4003</td>
<td>G90, G91</td>
</tr>
<tr>
<td>#4004</td>
<td>(Group 04 codes)</td>
</tr>
<tr>
<td>#4005</td>
<td>G94, G95</td>
</tr>
<tr>
<td>#4006</td>
<td>G20, G21</td>
</tr>
<tr>
<td>#4007</td>
<td>G40, G41, G42</td>
</tr>
<tr>
<td>#4008</td>
<td>G43, G44, G49</td>
</tr>
<tr>
<td>#4009</td>
<td>G73, G74, G76, G80-G89</td>
</tr>
<tr>
<td>#4010</td>
<td>G98, G99</td>
</tr>
<tr>
<td>#4011</td>
<td>G50, G51</td>
</tr>
<tr>
<td>#4012</td>
<td>G65, G66, G67</td>
</tr>
<tr>
<td>#4013</td>
<td>G96, G97</td>
</tr>
<tr>
<td>#4014</td>
<td>G54-G59</td>
</tr>
<tr>
<td>#4015</td>
<td>G61-G64</td>
</tr>
<tr>
<td>#4016</td>
<td>G68, G69</td>
</tr>
<tr>
<td>#4022</td>
<td>(Group 22 codes)</td>
</tr>
<tr>
<td>#4102</td>
<td>B code</td>
</tr>
<tr>
<td>#4107</td>
<td>D code</td>
</tr>
<tr>
<td>#4109</td>
<td>F code</td>
</tr>
<tr>
<td>#4111</td>
<td>H code</td>
</tr>
<tr>
<td>#4113</td>
<td>M code</td>
</tr>
<tr>
<td>#4114</td>
<td>Sequence number</td>
</tr>
<tr>
<td>#4115</td>
<td>Program number</td>
</tr>
<tr>
<td>#4119</td>
<td>S code</td>
</tr>
<tr>
<td>#4120</td>
<td>T code</td>
</tr>
<tr>
<td>#4130</td>
<td>P code (number of the currently selected additional workpiece coordinate system)</td>
</tr>
</tbody>
</table>

Example:
When #1 = #4001 is executed, the result in #1 would be 0, 1, 2, 3, or 33
## Workpiece Coordinate System Variables

<table>
<thead>
<tr>
<th>Axis</th>
<th>Function</th>
<th>Variable number</th>
</tr>
</thead>
<tbody>
<tr>
<td>First axis</td>
<td>External workpiece zero point offset</td>
<td>#2500 5201</td>
</tr>
<tr>
<td></td>
<td>G54 workpiece zero point offset</td>
<td>#2501 5221</td>
</tr>
<tr>
<td>X</td>
<td>G55 workpiece zero point offset</td>
<td>#2502 5241</td>
</tr>
<tr>
<td></td>
<td>G56 workpiece zero point offset</td>
<td>#2503 5261</td>
</tr>
<tr>
<td></td>
<td>G57 workpiece zero point offset</td>
<td>#2504 5281</td>
</tr>
<tr>
<td></td>
<td>G58 workpiece zero point offset</td>
<td>#2505 5301</td>
</tr>
<tr>
<td></td>
<td>G59 workpiece zero point offset</td>
<td>#2506 5321</td>
</tr>
<tr>
<td>Second axis</td>
<td>External workpiece zero point offset</td>
<td>#2600 5202</td>
</tr>
<tr>
<td>Y</td>
<td>G54 workpiece zero point offset</td>
<td>#2601 5222</td>
</tr>
<tr>
<td></td>
<td>G55 workpiece zero point offset</td>
<td>#2602 5242</td>
</tr>
<tr>
<td></td>
<td>G56 workpiece zero point offset</td>
<td>#2603 5262</td>
</tr>
<tr>
<td></td>
<td>G57 workpiece zero point offset</td>
<td>#2604 5282</td>
</tr>
<tr>
<td></td>
<td>G58 workpiece zero point offset</td>
<td>#2605 5302</td>
</tr>
<tr>
<td></td>
<td>G59 workpiece zero point offset</td>
<td>#2606 5322</td>
</tr>
<tr>
<td>Third axis</td>
<td>External workpiece zero point offset</td>
<td>#2700 5203</td>
</tr>
<tr>
<td>Z</td>
<td>G54 workpiece zero point offset</td>
<td>#2701 5223</td>
</tr>
<tr>
<td></td>
<td>G55 workpiece zero point offset</td>
<td>#2702 5243</td>
</tr>
<tr>
<td></td>
<td>G56 workpiece zero point offset</td>
<td>#2703 5263</td>
</tr>
<tr>
<td></td>
<td>G57 workpiece zero point offset</td>
<td>#2704 5283</td>
</tr>
<tr>
<td></td>
<td>G58 workpiece zero point offset</td>
<td>#2705 5303</td>
</tr>
<tr>
<td></td>
<td>G59 workpiece zero point offset</td>
<td>#2706 5323</td>
</tr>
<tr>
<td>Fourth axis</td>
<td>External workpiece zero point offset</td>
<td>#2800 5204</td>
</tr>
<tr>
<td>A</td>
<td>G54 workpiece zero point offset</td>
<td>#2801 5224</td>
</tr>
<tr>
<td></td>
<td>G55 workpiece zero point offset</td>
<td>#2802 5244</td>
</tr>
<tr>
<td></td>
<td>G56 workpiece zero point offset</td>
<td>#2803 5264</td>
</tr>
<tr>
<td></td>
<td>G57 workpiece zero point offset</td>
<td>#2804 5284</td>
</tr>
<tr>
<td></td>
<td>G58 workpiece zero point offset</td>
<td>#2805 5304</td>
</tr>
<tr>
<td></td>
<td>G59 workpiece zero point offset</td>
<td>#2806 5324</td>
</tr>
</tbody>
</table>
System Position Information

<table>
<thead>
<tr>
<th>Variable number</th>
<th>Position Information</th>
<th>Coordinate system</th>
<th>Tool compensation</th>
<th>Read during motion</th>
</tr>
</thead>
<tbody>
<tr>
<td>#5001-#5005</td>
<td>Block end point</td>
<td>Workpiece</td>
<td>Not Included</td>
<td>Enabled</td>
</tr>
<tr>
<td>#5021-#5025</td>
<td>Current position</td>
<td>Machine</td>
<td>Included</td>
<td>Disabled</td>
</tr>
<tr>
<td>#5041-#5045</td>
<td>Current position</td>
<td>Workpiece</td>
<td></td>
<td></td>
</tr>
<tr>
<td>#5061-#5065</td>
<td>Skip signal position</td>
<td></td>
<td></td>
<td>Enabled</td>
</tr>
<tr>
<td>#5081-#5085</td>
<td>Tool length offset</td>
<td></td>
<td></td>
<td>Disabled</td>
</tr>
<tr>
<td>#5101-#5105</td>
<td>Deviated servo position</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- The first digit (i.e. #5001) represents an axis number (1=X, 2=Y, 3=Z, 4=A, 5=B).

Mathematical expressions

<table>
<thead>
<tr>
<th>Function</th>
<th>Format</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Definition</td>
<td>#i = #j</td>
<td></td>
</tr>
<tr>
<td>Addition</td>
<td>#i = #j + #k</td>
<td></td>
</tr>
<tr>
<td>Subtraction</td>
<td>#i = #j - #k</td>
<td></td>
</tr>
<tr>
<td>Multiplication</td>
<td>#i = #j * #k</td>
<td></td>
</tr>
<tr>
<td>Division</td>
<td>#i = #j / #k</td>
<td></td>
</tr>
<tr>
<td>Sine</td>
<td>#i = SIN[#j]</td>
<td>The angle is specified in degrees. 90 degrees and 30 minutes is typed as 90.5 degrees.</td>
</tr>
<tr>
<td>Cosine</td>
<td>#i = COS[#j]</td>
<td></td>
</tr>
<tr>
<td>Tangent</td>
<td>#i = TAN[#j]</td>
<td></td>
</tr>
<tr>
<td>Arctangent</td>
<td>#i = ATAN[#j]</td>
<td></td>
</tr>
<tr>
<td>Square root</td>
<td>#i = SQRT[#j]</td>
<td></td>
</tr>
<tr>
<td>Absolute</td>
<td>#i = ABS[#j]</td>
<td></td>
</tr>
<tr>
<td>Rounding off</td>
<td>#i = ROUND[#j]</td>
<td></td>
</tr>
<tr>
<td>Rounding down</td>
<td>#i = FIX[#j]</td>
<td></td>
</tr>
<tr>
<td>Rounding up</td>
<td>#i = FUP[#j]</td>
<td></td>
</tr>
<tr>
<td>Natural Logarithms</td>
<td>#i = LN[#j]</td>
<td></td>
</tr>
<tr>
<td>Exponential function</td>
<td>#i = EXP[#j]</td>
<td></td>
</tr>
</tbody>
</table>
Mathematical operators

<table>
<thead>
<tr>
<th>Operator</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>EQ</td>
<td>Equal to (=)</td>
</tr>
<tr>
<td>NE</td>
<td>Not equal to!</td>
</tr>
<tr>
<td>GT</td>
<td>Greater than (&gt;)</td>
</tr>
<tr>
<td>GE</td>
<td>Greater than or equal to!</td>
</tr>
<tr>
<td>LT</td>
<td>Less than (&lt;)</td>
</tr>
<tr>
<td>LE</td>
<td>Less than or equal to!</td>
</tr>
</tbody>
</table>

The order in which the control evaluates expressions is important, since a change in the order of evaluation can result in a change of the resulting value.

The order of evaluation follows standard algebraic practice. That is:
1) The inner most parenthetic expression is evaluated first.
2) The next to inner set is solved next and each is solved in turn working towards the end of the sum.

The order of solving arithmetic operations within parentheses or when parentheses are not present is as follows:
1) All multiplication, division and modulo operations are performed in order from left to right.
2) All addition and subtraction are performed in order from left to right.
3) Relational operators are evaluated.

Typical Error $2+\frac{2}{10} = 2.2 \text{ OR } \frac{2+2}{10} = 0.4$

Conditional Branch statement (IF)

The control allows a conditional expression to be programmed. If the statement is read true a branch to a sequence number specified in the statement line occurs. If the statement is read false then the next program line is read. A conditional statement must include an operator between two variables or between a variable & constant and must be enclosed in brackets [ ]

Conditional repetition example

Repeat 5 times
#1 = 0
N10 (Label at start of loop)
#1 = #1 + 1
“PROGRAM”
IF [#1 LT 5] GOTO 10
Repetition statement (WHILE - D0)

The control allows a conditional expression to be programmed. If the statement is read true, a repeat is specified until the statement is read false. If the statement is read false, then the next program line is read after the END program line.. A conditional statement must include an operator between two variables or between a variable & a constant and must be enclosed in brackets [ ]. The identification number of the DO-END loop is limited to 3 and can be used as many times as desired. Overlapping Repetition statements cannot overlap.

Conditional repetition example

<table>
<thead>
<tr>
<th>Correct</th>
<th>Incorrect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Repeat 5 times</td>
<td>Repeat 5 times</td>
</tr>
<tr>
<td>#1 = 0</td>
<td>#1 = 0</td>
</tr>
<tr>
<td>WHILE [#1 LT 5] DO 2</td>
<td>WHILE [#1 LT 5] DO 2</td>
</tr>
<tr>
<td>#1 = # 1 + 1</td>
<td>#1 = # 1 + 1</td>
</tr>
<tr>
<td>“PROGRAM”</td>
<td>“PROGRAM”</td>
</tr>
<tr>
<td>END 2</td>
<td>WHILE [#1 LT 5] DO 3</td>
</tr>
<tr>
<td></td>
<td>END 2</td>
</tr>
<tr>
<td>This is Correct</td>
<td></td>
</tr>
<tr>
<td>Repeat 5 times</td>
<td>Repeat 5 times</td>
</tr>
<tr>
<td>#1 = 0</td>
<td>#1 = 0</td>
</tr>
<tr>
<td>WHILE [#1 LT 5] DO 2</td>
<td>WHILE [#1 LT 5] DO 2</td>
</tr>
<tr>
<td>#1 = # 1 + 1</td>
<td>#1 = # 1 + 1</td>
</tr>
<tr>
<td>“PROGRAM”</td>
<td>“PROGRAM”</td>
</tr>
<tr>
<td>WHILE [#1 LT 5] DO 3</td>
<td>WHILE [#1 LT 5] DO 3</td>
</tr>
<tr>
<td>“PROGRAM”</td>
<td>END 3</td>
</tr>
<tr>
<td>END 3</td>
<td>“PROGRAM”</td>
</tr>
<tr>
<td>END 2</td>
<td>END 2</td>
</tr>
</tbody>
</table>
Macro Call

A macro program (Macro sub-program) can be called using the following methods instead of the use of an M98 code:

<table>
<thead>
<tr>
<th>Macro Call</th>
<th>Simple call (G65)</th>
<th>Modal Call (G66, G67)</th>
</tr>
</thead>
</table>

With a **non-modal** G65 command, data contained with this line of program can be passed to the macro sub-program. This is not possible with the M98 command.

The **modal** G66 command will call the macro sub-program on every program line within the main program until the G66 command is cancelled with a G67 command.

**Information passed to the macro sub-program using G66 is only active on macro’s contained in the sub-program that are used for axis motion and not calculations.**

**G65/G66 Macro Simple Call**

G65 P? L? <argument - values>

Where:
- G65 = Non-modal macro call command.
- P? = Number of sub-program to call.
- L? = Number of times to repeat the sub-program.

<argument - values> = Data to be passed to the variables in the sub-program.

**Example:**

O1000  
G65 P1001 L2 A35 B45 ;

O1001 ;
G91 G0 X#1 Y#2 ;
M99 ;

**Argument - Values**

<table>
<thead>
<tr>
<th>Address</th>
<th>Variable number</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>#1</td>
</tr>
<tr>
<td>B</td>
<td>#2</td>
</tr>
<tr>
<td>C</td>
<td>#3</td>
</tr>
<tr>
<td>D</td>
<td>#7</td>
</tr>
<tr>
<td>E</td>
<td>#8</td>
</tr>
<tr>
<td>F</td>
<td>#9</td>
</tr>
<tr>
<td>H</td>
<td>#11</td>
</tr>
<tr>
<td>I</td>
<td>#4</td>
</tr>
<tr>
<td>J</td>
<td>#5</td>
</tr>
<tr>
<td>K</td>
<td>#6</td>
</tr>
<tr>
<td>M</td>
<td>#13</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>Variable number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q</td>
<td>#17</td>
</tr>
<tr>
<td>R</td>
<td>#18</td>
</tr>
<tr>
<td>S</td>
<td>#19</td>
</tr>
<tr>
<td>T</td>
<td>#20</td>
</tr>
<tr>
<td>U</td>
<td>#21</td>
</tr>
<tr>
<td>V</td>
<td>#22</td>
</tr>
<tr>
<td>W</td>
<td>#23</td>
</tr>
<tr>
<td>X</td>
<td>#24</td>
</tr>
<tr>
<td>Y</td>
<td>#25</td>
</tr>
<tr>
<td>Z</td>
<td>#26</td>
</tr>
</tbody>
</table>
G65 Macro for a Counterbore

```
T? M6 (ENDMILL)
G0 G90 G40 G21 G17 G94 G80
G54 X? Y? S? M3 (Move to bore centre)
G43 Z? H?
;
G65 P1001 A? D?
(A = C/BORE DIAMETER)
(D = RADIUS OFFSET NUMBER)
M30

O1001
#11=[[#1*0.8]/2]
#12=[[#1/2]-#11]
G91 Y#12
G41 X#11 D#7
G3 X-#11 Y#11 R#11
J-[#1/2]
X-#11 Y-#11 R#11
G1 G40 X#11
G0 G90 Z100
M99
```

Macro for Internal Helical

```
T? M6 (THREADMILL)
G0 G90 G40 G21 G17 G94 G80
G54 X? Y? S? M3 (Move to bore centre)
G43 Z? H?
;
G65 P1002 A? B? D?
(A = THREAD DIAMETER)
(B = PITCH)
(D = RADIUS OFFSET NUMBER)
M30

O1002
#11=[[#1*0.8]/2]
#12=[[#1/2]-#11]
;
G91 Y#12
G41 X#11 D#7
G3 X-#11 Y#11 R#11 Z#2/4
J-[#1/2] Z#2
X-#11 Y-#11 R#11 Z#2/4
G1 G40 X#11
G0 G90 Z100
M99
```
G65 Macro for an Increasing Radius.

A = #1 (Start Angle 0 degrees)
B = #2 (Start Radius)
C = #3 (Increment angle for accuracy calculations.)
I = #4 (Finish Angle)
J = #5 (Finish radius)
K = #6 (Milling feed)

O2222
T5 M6
G0 G90 G40 G21 G17 G94 G80
G54 X35 Y0 S500 M3
G43 Z100 H?
Z5
G1 Z-0.5 F200
G65 P8999 A0 B35 C0.01 I70 J37 K500
G0 G90 Z100 M30

O8999
#7 = #4 / #3
#8 = [#5 - #2] / #7
N1 #2 = #2 + #8
#1 = #1 + #3
#9 = #2 * COS [ #1 ]
#10 = #2 * SIN [ #1 ]
G1 X#9 Y#10 F#6
IF [#1 LT #4] GOTO 1
G0 Z10
M99

1) Total no. of moves 70 / 0.01
2) Increase in radius 37-35/7000
3) Next Radius i.e. 35+inc. radius.
4) Increase in angle
5) New X axis position
6) New Y axis position
7) Feed move to new positions
8) If new angle is less than finish angle
go to line N1.
G65 Macro for Internal Ellipse

T1 M6
G0 G90 G40 G21 G17 G94 G80
G54 X0 Y0 S? M3
G43 Z5 H?
G1 Z-? F?
#20 = 2 ; Incremental degree calculation
#21 = 0 ; Start Angle
#22 = 30 ; Y Axis Radius
#23 = 50 ; X Axis Radius
G41 X#23  D? ; Compensation motion to right side of internal pocket
N10 #21 = [#21 + #20] ; Angular Count
#24 = SIN[#21] ; Incremental Y axis calculation
#25 = COS[#21] ; Incremental X axis calculation
#24 = [#24*#22] ; Absolute Y calculation
#25 = [#25*#23] ; Absolute X calculation
X#25 Y#24 ; Movement in X & Y axis
IF [#21 LT 360] GOTO 10 ; Restart if less than 360 degree motion
IF [#21 GT 360] GOTO 20 ; If final angle becomes greater than 360 degrees recalculate
IF [#21 EQ 360] GOTO 30 ; Finish if total angle is equal to 360 degree
N20 #21 = 360
GOTO 10
N30 G40 X0
G0 G90 Z100 M30
Datum Shift
Datum Shift

A temporary shift of the set datum can be achieved within the program itself. This shift is known as a “Local Co-ordinate System”.

The use of a G52 command with an absolute axis position allows this temporary shift of the workpiece co-ordinate which has been set.

This code cannot be used Incrementally

The datum shift must be cancelled when finished with.

The following are ways to achieve this:
1) Programming G52 with zero axis motion
2) Data reset
3) End of program
4) Machine power off
Rotation
**Rotation**

*(Option)*

A programmed shape can be rotated around a programmed pole position. Programming a **G68** with an axis position for the centre of rotation together with the angle of rotation will do this.

**G68 X? Y? R?**

Where:
- X? & Y? = Centre of rotation
- R? = Angle of rotation around the current plane (R+ = Anti-clockwise)

All information programmed after this line of command will instruct the control to calculate the new axis position at the displaced angle.

The smallest increment of angular displacement = 0.001 Deg.

If the X & Y values are not programmed then the current tool position becomes the rotation pole centre.

Incremental can be used with G68 rotation. The X, Y & R are established as incremental values if G68 is specified with a G91 code.

The Rotation must be cancelled when finished with.

The following are ways to achieve this:

1) Programming G69
2) Data reset
3) End of program
4) Machine power off
Rotation

All holes 20mm Deep

T1 M6
G0 G90 G40 G21 G17 G94 G80
G54 X20 Y0 S? M3
G43 Z100 H?
Z5
G81 R3 Z-20 F? M8
X30
X45
G68 X0 Y0 R120
X20 Y0
X30
X45
G68 X0 Y0 R240
X20 Y0
X30
X45
G69 G80
G0 G90 Z100 M30
Programmable Coolant
Automatic Coolant Jet Control (Option)

The Coolant Jet System, mounted beneath the spindle carrier, has eight positions to ensure the coolant is directed to the cutting tip of any tool within the maximum tool length and diameter specified for the machine.

The group of miscellaneous codes (M15 E1 - M15 E8) control the positioning of the Automatic Coolant Jet.

The coolant must be activated before programming the automatic coolant nozzle

The table below is used to identify the M code to an active tool length and radius. Code M15 E1 is associated with the smallest/shortest tool and code M15 E8 is associated with the largest/longest tool.

<table>
<thead>
<tr>
<th>Tool Length</th>
<th>Upto 100</th>
<th>Upto 150</th>
<th>Upto 200</th>
<th>Upto 250</th>
<th>Over 250</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;50</td>
<td>M15E3</td>
<td>M15E5</td>
<td>M15E6</td>
<td>M15E7</td>
<td>M15E8</td>
</tr>
<tr>
<td>&lt;50</td>
<td>M15E2</td>
<td>M15E4</td>
<td>M15E5</td>
<td>M15E6</td>
<td>M15E6</td>
</tr>
<tr>
<td>&lt;30</td>
<td>M15E2</td>
<td>M15E3</td>
<td>M15E4</td>
<td>M15E5</td>
<td>M15E6</td>
</tr>
<tr>
<td>&lt;15</td>
<td>M15E1</td>
<td>M15E2</td>
<td>M15E3</td>
<td>M15E4</td>
<td>M15E5</td>
</tr>
</tbody>
</table>

Example:

M15 E4 is selected for an end mill tool, 170 mm long with a radius of 30 mm. When an M6 command is processed, the control automatically retracts the coolant jet to the M15 E1 position to ensure clearance with the tool magazine guard. The coolant jet will remain in position on completion of the tool change until another M15 code from the group is programmed.

Also the coolant jet can be programmed to move to a calculated position relative to the active tool length and radius by utilizing code M15 with no “E” word.

Example Program:

<table>
<thead>
<tr>
<th>T1 M6</th>
</tr>
</thead>
<tbody>
<tr>
<td>G54 X? Y? S? M3</td>
</tr>
<tr>
<td>G43 Z100 H1</td>
</tr>
<tr>
<td>Z5</td>
</tr>
<tr>
<td>G1 Z-? F? M8</td>
</tr>
<tr>
<td>G41 X? D?</td>
</tr>
<tr>
<td>M15</td>
</tr>
<tr>
<td>Y?</td>
</tr>
</tbody>
</table>
Corner

Radius/Chamfer
Corner Radius & Uniform Chamfers

(Option)

Blend chamfers and Blend Radii can be created within a program with the aid of short word addresses (,C for chamfers and ,R for radius). The word address ,C & ,R are Non-Modal values and are added to the program line on the approach to the known intersection programmed point.
Programmable Data Entry
**Programmable Data Entry**

*(Option)*

Certain areas of the Fanuc control can be written to from a part program using a G10 command with various pieces of information.

The types of transferable data can include information to the Tool offsets & Fixture offsets.

### Tool data entry

<table>
<thead>
<tr>
<th>Tool Length</th>
<th>G10 L10 P? R?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tool Length Wear</td>
<td>G10 L11 P? R?</td>
</tr>
<tr>
<td>Tool Radius</td>
<td>G10 L12 P? R?</td>
</tr>
<tr>
<td>Tool Radius Wear</td>
<td>G10 L13 P? R?</td>
</tr>
</tbody>
</table>

Where:

- G10 = Code to transfer data
- L? = Appropriate tool column
- P? = Tool offset row number
- R? = Value within the allowed range.

### Fixture Offset data entry

<table>
<thead>
<tr>
<th>Fixture Offset</th>
<th>G10 L2 P? X? Y? Z?</th>
</tr>
</thead>
</table>

Where:

- G10 = Code to transfer data
- L2 = Fixture Offset data
- P? = Fixture Offset data number
- (P1=G54, P2=G55, P3=G56, P4=G57, P5=G58, P5=G59)
- X, Y, Z = Axis value to be transferred

**Using G91 on the G10 program line of information will add the value to the existing data.**
Polar Co-ordinates
Polar Co-ordinate Command

*(Option)*

The control has the ability to position itself to an endpoint in any plane using a **G16** code with only information regarding the length of the move from the X, Y & Z datum point with the angle of the line as that of a right angled triangle.

All angles are relative to the current plane.

\[
\text{G16 G17 } X? \hspace{1em} Y?
\]

Where:
- **G16** = Polar command
- **G17** = XY Plane
- **X**? = Hypotenuse of the right angle triangle
- **Y**? = Angle of the hypotenuse line from the 0 Deg. point as below.

It is also possible to use **G91** Incremental to control the angle.

**G16 Polar command is cancelled by the programming of a G15 command.**
Polar Co-ordinate Command

Hole position Example

3 Holes on 40 PCD

T? M6
G0 G90 G40 G21 G17 G94 G80
G54 X0 Y0 S? M3
Z5
G81 G16 G98 X20 Y30 R-2 Z-10 F? M8
Y150
Y270
G15 G80
G0 G90 Z100
T? M6
G0 G90 G40 G21 G17 G94 G80
G54 X0 Y0 S? M3
Z5
G84 G16 G95 G98 X20 Y30 R-2 Z-10 F1 M8
Y150
Y270
G15 G80
G0 G90 Z100
M30
User Supplement Cycles
Circular Hole Pattern
A variable number of holes can be requested in the circular hole pattern as well as specifying the required drilling cycle.


[ ] denotes optional inputs.
If no X & Y axis are on the line of program then the current tool position will become the PCD centre.

<table>
<thead>
<tr>
<th>G</th>
<th>Cycle Call (G65)</th>
<th>U</th>
<th>Diameter of PCD</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>8951 – Bolt hole circle</td>
<td>H</td>
<td>No. of holes</td>
</tr>
<tr>
<td>X</td>
<td>X axis to PCD centre</td>
<td>I</td>
<td>Drilling cycle</td>
</tr>
<tr>
<td>Y</td>
<td>Y axis to PCD centre</td>
<td>J</td>
<td>Return point code (98 / 99)</td>
</tr>
<tr>
<td>Z</td>
<td>Z axis hole end position</td>
<td>Q</td>
<td>Peck depth (G73/G83) *</td>
</tr>
<tr>
<td>A</td>
<td>Angle of hole from 0</td>
<td>D</td>
<td>Dwell</td>
</tr>
<tr>
<td>R</td>
<td>R Plane</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note
1) I must be programmed before J
2) *Q is mandatory with I83 or I73
3) A absent = zero angle
4) D absent = no dwell
Rectangular Hole Pattern
A different number of holes can be requested in both X & Y axis in a rectangular pattern, optionally rotated at an angle specifying the required drilling cycle.


[ ] denotes optional inputs.
If no X & Y axis are on the line of program then the current tool position will become the 1st hole position.

<table>
<thead>
<tr>
<th>G</th>
<th>Cycle Call (G65)</th>
<th>U</th>
<th>Length in the X axis</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>8952 – Rectangular pattern</td>
<td>V</td>
<td>Length in the Y axis</td>
</tr>
<tr>
<td>X</td>
<td>X axis to 1st hole</td>
<td>I</td>
<td>Drilling cycle</td>
</tr>
<tr>
<td>Y</td>
<td>Y axis to 1st hole</td>
<td>J</td>
<td>Return point code (98 / 99)</td>
</tr>
<tr>
<td>Z</td>
<td>Z axis hole end position</td>
<td>Q</td>
<td>Peck depth (G73/G83) *</td>
</tr>
<tr>
<td>A</td>
<td>Angle of rotation from 0</td>
<td>D</td>
<td>Dwell</td>
</tr>
<tr>
<td>R</td>
<td>R Plane</td>
<td>H</td>
<td>No. of holes in the X axis</td>
</tr>
<tr>
<td></td>
<td></td>
<td>K</td>
<td>No. of holes in the Y axis</td>
</tr>
</tbody>
</table>

Note
1) I, J & K are programmed in alphabetical order
2) *Q is mandatory with I83 or I73
3) A absent = zero angle
4) D absent = no dwell
Grid or Line Hole Pattern

A Grid or Line of holes can be specified, optionally rotated at an angle specifying the required drilling cycle.


[ ] denotes optional inputs.

If no X & Y axis are on the line of program then the current tool position will become the 1st hole position.

<table>
<thead>
<tr>
<th>G</th>
<th>Cycle Call (G65)</th>
<th>U</th>
<th>Length in the X axis</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>8953 – Grid or Line</td>
<td>V</td>
<td>Length in the Y axis</td>
</tr>
<tr>
<td>X</td>
<td>X axis to 1st hole</td>
<td>I</td>
<td>Drilling cycle</td>
</tr>
<tr>
<td>Y</td>
<td>Y axis to 1st hole</td>
<td>J</td>
<td>Return point code (98 / 99)</td>
</tr>
<tr>
<td>Z</td>
<td>Z axis hole end position</td>
<td>Q</td>
<td>Peck depth (G73/G83) *</td>
</tr>
<tr>
<td>A</td>
<td>Angle of rotation from 0</td>
<td>D</td>
<td>Dwell</td>
</tr>
<tr>
<td>R</td>
<td>R Plane</td>
<td>H</td>
<td>No. of holes in the X axis</td>
</tr>
<tr>
<td></td>
<td></td>
<td>K</td>
<td>No. of holes in the Y axis</td>
</tr>
</tbody>
</table>

Note

1) I, J & K are programmed in alphabetical order
2) *Q is mandatory with I83 or I73
3) A absent = zero angle
4) D absent = no dwell

For a line of holes in the X axis: H > 1, K = 1
For a line of holes in the Y axis: K>1, H = 1
**Rectangular Facing**

Bi-directional facing in the X & Y axis can be performed with the ability to also make allowances for finishing results.

\[
\]

[ ] denotes optional inputs.

<table>
<thead>
<tr>
<th>G</th>
<th>Cycle Call (G65)</th>
<th>U</th>
<th>Length in the X axis</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>8954 – Rectangular face</td>
<td>V</td>
<td>Length in the Y axis</td>
</tr>
<tr>
<td>X</td>
<td>X axis to L/H corner</td>
<td>K</td>
<td>Cut width % (10% – 80%)</td>
</tr>
<tr>
<td>Y</td>
<td>Y axis to L/H corner</td>
<td>D</td>
<td>Tool radius offset number</td>
</tr>
<tr>
<td>Z</td>
<td>Z axis end position</td>
<td>Q</td>
<td>Finish allowance</td>
</tr>
<tr>
<td>R</td>
<td>R Plane</td>
<td>F</td>
<td>Finish feedrate *</td>
</tr>
<tr>
<td>W</td>
<td>Number of cuts in Z axis</td>
<td>S</td>
<td>Finish speed *</td>
</tr>
<tr>
<td>C</td>
<td>Clearance</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note**

1) *Mandatory with Q?
2) If U < V then machining will be performed in the Y axis.
3) If U > V then machining will be performed in the X axis.

**Roughing Feed/Speed**

The roughing feed and speed are programmed before the cycle call.
Circular Facing

Bi-directional facing in the X & Y axis of a circular billet can be performed with the ability to also make allowances for finishing results.

\[
\]

[ ] denotes optional inputs.

<table>
<thead>
<tr>
<th>G</th>
<th>Cycle Call (G65)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>8955 – Circular Face</td>
</tr>
<tr>
<td>X</td>
<td>X axis to centre</td>
</tr>
<tr>
<td>Y</td>
<td>Y axis to centre</td>
</tr>
<tr>
<td>Z</td>
<td>Z axis end position</td>
</tr>
<tr>
<td>R</td>
<td>R Plane</td>
</tr>
<tr>
<td>W</td>
<td>Number of cuts in Z axis</td>
</tr>
<tr>
<td>C</td>
<td>Clearance</td>
</tr>
<tr>
<td>U</td>
<td>Diameter of billet</td>
</tr>
<tr>
<td>K</td>
<td>Cut width % (10% – 80%)</td>
</tr>
<tr>
<td>D</td>
<td>Tool radius offset number</td>
</tr>
<tr>
<td>Q</td>
<td>Finish allowance</td>
</tr>
<tr>
<td>F</td>
<td>Finish feedrate *</td>
</tr>
<tr>
<td>S</td>
<td>Finish speed *</td>
</tr>
</tbody>
</table>

**Note**

1) *Mandatory with Q?*

**Roughing Feed/Speed**

The roughing feed and speed are programmed before the cycle call.
Circular Pocket

Internal circular pocketing can be performed with the ability to also make allowances for finishing results.


[ ] denotes optional inputs.

<table>
<thead>
<tr>
<th>G</th>
<th>Cycle Call (G65)</th>
<th>U</th>
<th>Diameter of pocket</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>8956 – Circular Pocket</td>
<td>K</td>
<td>Cut width % (10% – 80%)</td>
</tr>
<tr>
<td>X</td>
<td>X axis to centre</td>
<td>D</td>
<td>Tool radius offset number</td>
</tr>
<tr>
<td>Y</td>
<td>Y axis to centre</td>
<td>Q</td>
<td>Finish allowance</td>
</tr>
<tr>
<td>Z</td>
<td>Z axis end position</td>
<td>F</td>
<td>Finish feedrate *</td>
</tr>
<tr>
<td>R</td>
<td>R Plane</td>
<td>S</td>
<td>Finish speed *</td>
</tr>
<tr>
<td>W</td>
<td>Number of cuts in Z axis</td>
<td>E</td>
<td>Z axis feed</td>
</tr>
</tbody>
</table>

**Note**

1) *Mandatory with Q?*

2) Climb milling (down cut) is performed.

**Roughing Feed/Speed**

The roughing feed and speed are programmed before the cycle call.
Rectangular Pocket

Internal rectangular pocketing can be performed with the ability to also make allowances for finishing results.


[ ] denotes optional inputs.

<table>
<thead>
<tr>
<th>G</th>
<th>Cycle Call (G65)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>8957 – Rectangular Pocket</td>
</tr>
<tr>
<td>X</td>
<td>X axis to centre</td>
</tr>
<tr>
<td>Y</td>
<td>Y axis to centre</td>
</tr>
<tr>
<td>Z</td>
<td>Z axis end position</td>
</tr>
<tr>
<td>R</td>
<td>R Plane</td>
</tr>
<tr>
<td>W</td>
<td>Number of cuts in Z axis</td>
</tr>
<tr>
<td>U</td>
<td>X axis length</td>
</tr>
<tr>
<td>V</td>
<td>Y axis width</td>
</tr>
<tr>
<td>K</td>
<td>Cut width % (10% – 80%)</td>
</tr>
<tr>
<td>D</td>
<td>Tool radius offset number</td>
</tr>
<tr>
<td>Q</td>
<td>Finish allowance</td>
</tr>
<tr>
<td>F</td>
<td>Finish feedrate *</td>
</tr>
<tr>
<td>S</td>
<td>Finish speed *</td>
</tr>
<tr>
<td>E</td>
<td>Z axis feed</td>
</tr>
</tbody>
</table>

**Note**
1) *Mandatory with Q?
2) Climb milling (down cut) is performed.

**Roughing Feed/Speed**
The roughing feed and speed are programmed before the cycle call.