Denford

DENFORD

COMPUTERISED MACHINES AND SYSTEMS

G and M Programming for CNC Milling Machines

Denford Limited
Birds Royd
Brighouse
West Yorkshire
England
HD6 1NB
Tel: +44 (0) 1484 712264
Fax: +44 (0) 1484 722160
Email: education@denford.co.uk

NOTES.

Contents.

Contents.

Contents.

Title	Page No.

M Codes - Miscellaneous Functions.

Introduction.

The Denford CNC (Computer Numerical Control) unit fitted to Denfords range of machine tools is a FANUC compatible system which uses ISO code format.

This manual covers the stages involved in producing the coded instructions, used by the CNC unit to make the component. These coded instructions are called the part program.

Each part program contains a number of different codes, the most important being the collection of G and M codes. Essentially, these form the basic language used to describe how a component will be manufactured, the order in which to carry out machining tasks, when to change tools, how far to cut into the material etc.....

The front sections of this manual cover the basics of part programming, including guidelines for general layout and commands. Each section builds progressively, using plain, easy to follow text, to cover the most common aspects of programming. At the end of this stage, the operator should be confident enough to tackle basic part programming.

Naturally, this manual cannot "teach" the operator everything there is to know about programming. The subject is simply too vast to include it all. The content of this manual does, however, form a good basis from which to start learning and hopefully inspires confidence in using more technically structured documents.

The *G Codes* and *M Codes* sections contain information which are more specific to certain commands and functions - these sections are intended more as a reference guide, once the operator is confident with the basics of programming.

What is CNC?

CNC (Computer Numerical Control) is the general term used for a system which controls the functions of a machine tool using coded instructions processed by a computer.

Example CNC Manufacturing Process.

The diagram on page 8 shows the main stages involved in producing a component on a CNC system.

1) A part program is written, using G and M codes. This describes the sequence of operations that the machine must perform in order to manufacture the component.

This program can be produced off-line, ie, away from the machine, either manually or with the aid of a CAD/CAM system.

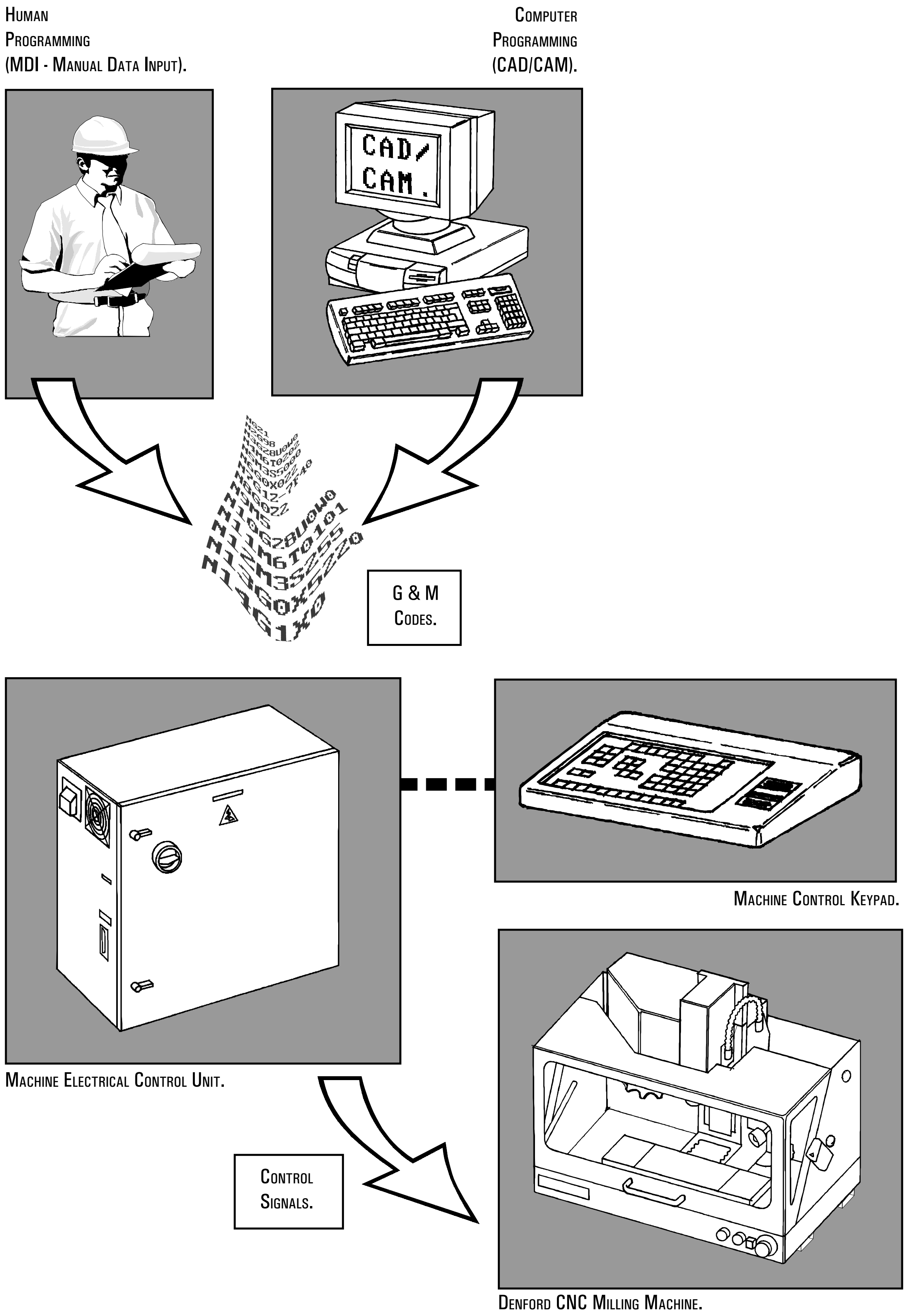
2) The part program is loaded into the machines computer, called the controller. At this stage, the program can still be edited or simulated using the machine controller keypad/input device.

3) The machine controller processes the part program and sends signals to the machine components directing the machine through the required sequence of operations necessary to manufacture the component.

The application of CNC to a manual machine allows its operation to become fully automated.

Combining this with the use of a part program enhances the ability of the machine to perform repeat tasks with high degrees of accuracy.

Diagram - Example CNC Manufacturing Process.



Composition of a Part Program.

A *Part Program* is a list of coded instructions which describes how the designed component, or part, will be manufactured.

These coded instructions are called *data* - a series of letters and numbers. The part program includes all the geometrical and technological data to perform the required machine functions and movements to manufacture the part.

The part program can be further broken down into separate lines of data, each line describing a particular set of machining operations. These lines, which run in sequence, are called *blocks*.

A block of data contains *words*, sometimes called *codes*. Each word refers to a specific cutting/movement command or machine function. The programming language recognised by the CNC, the machine controller, is an I.S.O. code, which includes the *G and M code groups*.

Each program word is composed from a letter, called the *address*, along with a number.

Part Program example -

```
(Mill CAM Designer
- star.MCD)
(2/10/1997)
(Triac PC (metric))
(Post fanucm:1.20
24 June 1994)
O0050 ;
N010  G21 ;
[BILLET X240 Y170 Z10
[EDGEMOVE X0 Y0
[TOOLDEF T1 D2
N020  G91 G28 X0 Y0 Z0 ;
N030  M6 T1 ;
N040  G43 H1 ;
N050  M3 S3000 ;
N060  G90 G00 X90 Y120 ;
N070  Z2 ;
N080  G01 Z-0.5 F40 ;
N090  X105 Y160 F60 ;
N100  X120 Y120 ;
N110  X165 ;
N120  X130 Y95 ;
N130  X145 Y50 ;
N140  X105 Y80 ;
N150  X65 Y50 ;
N160  X80 Y95 ;
N170  X45 Y120 ;
N180  X90 ;
N190  G00 Z2 ;
N200  M5 ;
N210  G91 G28 X0 Y0 Z0 ;
N220  M30 ;
```

Block example - N080 G01 Z-0.5 F40 ;

↓

Word example - G01

↓

Address example - G

Main Program Structure.

The part program can contain a number of separate programs, which together describe all the operations required to manufacture the part.

The *Main Program* is the controlling program, ie, the program first read, or accessed, when the entire part program sequence is run. This controlling program can then call a number of smaller programs into operation. These smaller programs, called *Sub Programs*, are generally used to perform repeat tasks, before returning control back to the main program.

Normally, the controller operates according to one program. In this case the main program is also the part program.

Main Programs are written using I.S.O. address codes listed below.

Addresses -

- N refers to the block number.
- G refers to the G code (Preparatory function).
- X refers to the absolute/incremental distance travelled by the slide tool in the X axis direction.
- Y refers to the absolute/incremental distance travelled by the slide tool in the Y axis direction.
- Z refers to the absolute/incremental distance travelled by the slide tool in the Z axis direction.
- F refers to the feed rate.
- M refers to the M code (Miscellaneous function).
- S refers to the spindle speed.
- T refers to the tooling management.

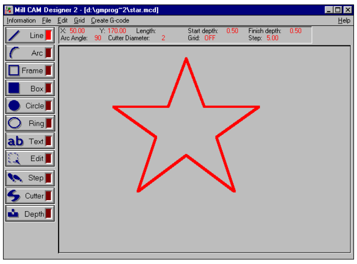
Each block, or program line, contains addresses which appear in this order :

N , G , X , Y , Z , F , M , S , T ;

This order should be maintained throughout every block in the program, although individual blocks may not necessarily contain all these addresses.

Main Program Structure.

The organisation of blocks of data within the program follows a layout. Again, it is recommended that the programmer keeps to this program layout.



The main program can be generated using Denfords MillCAM Designer CAD/CAM software package, saved onto disk and transferred to the CNC.

Alternatively, the main program can be manually entered into the CNC memory when the controller is set in Edit Mode.

CAD/CAM software package information.

```
(Mill CAM Designer
- star.MCD)
(2/10/1997)
(Triac PC (metric))
(Post fanucm:1.20
24 June 1994)
```

Program number.

```
O0050 ;
```

Units definition (Metric or Imperial) and billet size for simulation.

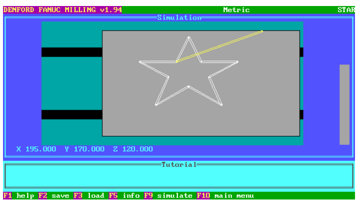
```
N010 G21 ;
[BILLET X240 Y170 Z10
[EDGEMOVE X0 Y0
[TOOLDEF T1 D2
```

Main program information.

```
N020 G91 G28 X0 Y0 Z0 ;
N030 M6 T1 ;
N040 G43 H1 ;
N050 M3 S3000 ;
N060 G90 G00 X90 Y120 ;
N070 Z2 ;
N080 G01 Z-0.5 F40 ;
N090 X105 Y160 F60 ;
N100 X120 Y120 ;
N110 X165 ;
N120 X130 Y95 ;
N130 X145 Y50 ;
N140 X105 Y80 ;
N150 X65 Y50 ;
N160 X80 Y95 ;
N170 X45 Y120 ;
N180 X90 ;
N190 G00 Z2 ;
N200 M5 ;
N210 G91 G28 X0 Y0 Z0 ;
```

Program end.

```
N220 M30 ;
```



Machine Simulation of G and M code program.

The above listing shows an example program using the Denford programming system.

For the program to operate correctly on a genuine FANUC control, the CAD/CAM software information and billet size definitions must be removed from the listing.

Sub Program Structure.

A program which contains fixed sequences or frequently repeated patterns may be entered into memory as a *Sub Program*, in order to simplify the main program.

A sub program is entered into the machine controller memory in Edit Mode, in the same manner as the main program.

Differences between a sub and main program :

1) A sub program does not have a billet size definition at the top of the program listing.

2) A sub program is ended by the M99 code.

The sub program can be called into operation when the machine is set to run in Auto Mode. Sub programs can also call other sub programs into operation.

When the main program calls one sub program into operation, the process is called a one-loop sub program call. It is possible to program a maximum four-loop sub program call within the main program. Shown below is an illustration of a two-loop sub program call.

Main Program.

```
O0001
N0010 G21;
[BILLET X.... Y.... Z....
N0020 ......      ;
N0030 ......      ;
N0040 ......      ;
N0050 ......      ;
N0060 ......      ;
N0070 ......      ;
N0080 M98 P1000;
N0090 ......      ;
N0100 ......      ;
N0110 ......      ;
N0120 ......      ;
N0130 ......      ;
N0140 ......      ;
N0150 M30;
```

Sub Program - 1.

```
O1000
N0010 ......      ;
N0020 ......      ;
N0030 ......      ;
N0040 ......      ;
N0050 ......      ;
N0060 ......      ;
N0070 ......      ;
N0080 ......      ;
N0090 ......      ;
N0100 ......      ;
N0110 M98 P2000;
N0120 ......      ;
N0130 ......      ;
N0140 ......      ;
N0150 ......      ;
N0160 M99;
```

Sub Program - 2.

```
O2000
N0010 ......      ;
N0020 ......      ;
N0030 ......      ;
N0040 ......      ;
N0050 ......      ;
N0060 ......      ;
N0070 ......      ;
N0080 ......      ;
N0090 ......      ;
N0100 ......      ;
N0110 ......      ;
N0120 ......      ;
N0130 ......      ;
N0140 ......      ;
N0150 ......      ;
N0160 M99;
```

One-Loop Nesting. Two-Loop Nesting.

Sub Program Commands - General Notes.

Note 1.

A sub program must be saved to memory using a four digit number.

Note 2.

If cutter compensation is required on a tool and the co-ordinates for the tool are within the sub program, the cutter compensation must be applied and cancelled within the sub program.

Note 3.

To call a sub program the M98 code is used followed by PØØØØ (the number of the sub program required).

For example,

M98 P2ØØØ

This command is read call program number 2ØØØ.

Note 4.

A sub program call command (M98 PØØØØ) can be specified along with a move command in the same block.

For example,

GØ1 X42.5 M98 P1ØØØ;

Note 5.

At the end of a sub program, the M99 code is entered. This returns control to the main program.

The M99 code will return control to the next block after the M98 sub program call block in the main program.

If the code M99 PØØØØ is entered, control will pass to the main program at a block with the N number equal to that of the P number stated after the M99 code.

For example,

M99 PØ16Ø

This command is read return to the main program at block number NØ16Ø.

Sub Program Repeat Command.

A call command can be set to call a sub program repeatedly. This call can specify upto 999 repetitions of a sub program.

A sub program repeat command has the following format :

M98 PØØØ ØØØØ

Call command.

Number of times the sub program is to be repeated.

Sub program number.

When the repetition is omitted, the sub program will be called once only.

For example,

M98 P1ØØØØ1

This command is read call the sub program number ØØØ1 ten times.

Billet Definition.

The *Billet Definition* is a feature which is only used in the Denford programming system.

It defines the size of the workpiece billet for use in the simulation sections of the Denford machine software. The billet definition command has no outcome on the actual machining of the part.

The billet definition command is written at the start of the main program. The previous block usually states the units of measurement to be applied to the billet dimensions, ie, G21 (Metric data input) or G2Ø (Imperial data input).

For example,

NØØ1Ø G21;

[BILLET X1ØØ Y15Ø Z2Ø;

These two commands are read.

- program line number 1Ø states that all units are to be measured in Metric,

- the billet is a rectangular piece of material, measuring 1ØØmm x 2ØØmm, with a thickness of 2Ømm.

Note 1.

A program that has been written on a Denford control (or using Denford CAD/CAM post processor software) will not operate directly on a genuine FANUC machine. The simulation sections of the program are incompatible with the FANUC control.

For the program to run successfully, lines referring to the CAD/CAM software (at the beginning of the program) and the billet definition block must be deleted.

Program Numbering and Saving.

The Denford system of program numbering relies on the programmer saving the program to disk or computer hard drive at the time of writing.

When saving a program using the Denford Desktop Tutor keypad, the program number can range from 1 to 99999999.

Writing the program on an offline system with a qwerty keyboard allows the programmer to save the program using letters and/or numbers.

Note 1.

Sub program numbers must be saved between the ranges ØØØ1 to 9999, ie, using a four digit number. It is recommended that all programs saved have filenames also between these numbers.

Note 2.

Before saving a program to disk or hard drive, check that the program name you wish to use has not been used on another file. If the program is saved using a name identical to an old program file, the old program file will be overwritten.

Note 3.

Programs that need to be used at a later date on genuine FANUC controls must have their program number stated on the first program block. The format for inserting a FANUC compatible program number line is as follows :

O ØØØØ

Address code O.

Four digit program number.

Program Block Numbering.

A program is composed of several commands, each command instructing the machine to carry out a particular operation. Each command is a separate line of data within the program, called a *Block*.

One block is separated from another block using an end of block code, ie, effectively signifying the end of a program line. The Denford programming system uses a semicolon (;) as the end of block code.

A four digit sequence number can be specified (ØØØ1 - 9999) following the address code N, at the start of each block. The order of these block numbers is arbitrary and need not be consecutive. Block numbers can be specified for every program line, or just on program lines requiring them.

Note 1.

The block number must be written at the start of a program line when used.

Note 2.

It is recommended that all blocks are numbered using a four digit number which rises between each block in steps of 1Ø. This allows the program to be edited at a later date, ie, new blocks can be inserted or deleted as required.

For example :

N ØØ1Ø ....

N ØØ2Ø ....

N ØØ3Ø ....

N ØØ4Ø ....

etc....

Note 3.

Even when block numbering is not a priority, it is useful to insert block numbers at important points in the program, such as tool change commands. This will help if a program search is used in the future.

Block Configuration.

The sequence in which address codes appear in each block should remain consistent throughout the program. It is recommended that the order of these address codes follows the example shown below :

NØØØØ	GØØ	XØØ.Ø YØØ.Ø ZØØ.Ø	FØØØØ	MØØ	SØØØØ	TØØ	;
Block sequence number.	Preparatory function.	Interpolation function.	Feedrate function.	Miscellaneous function.	Spindle function.	Tool function.	End of block signal.

Note 1.

Each block may not necessarily contain all these items.

G Codes (Preparatory Functions).

Preparatory functions, called *G codes*, are used to determine the geometry of tool movements and operating state of the machine controller; functions such as linear cutting movements, drilling operations and specifying the units of measurement.

They are normally programmed at the start of a block.

A G code is defined using the G address letter and a two digit number as follows,

G ØØ

Address. Two digit number.

Tool Movement.

The tool moves along straight lines and arcs forming the workpiece shape.

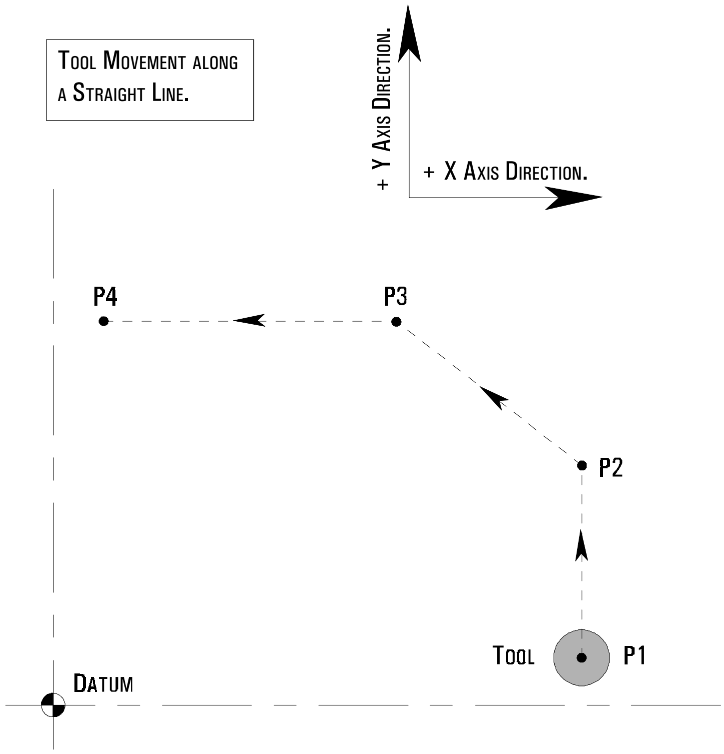
A) Tool Movement along a Straight Line.

Program command format:

GØ1 Y _ _ _ _ ; (P1 - P2)

X _ _ _ _ Y _ _ _ _ ; (P2 - P3)

X _ _ _ _ ; (P3 - P4)

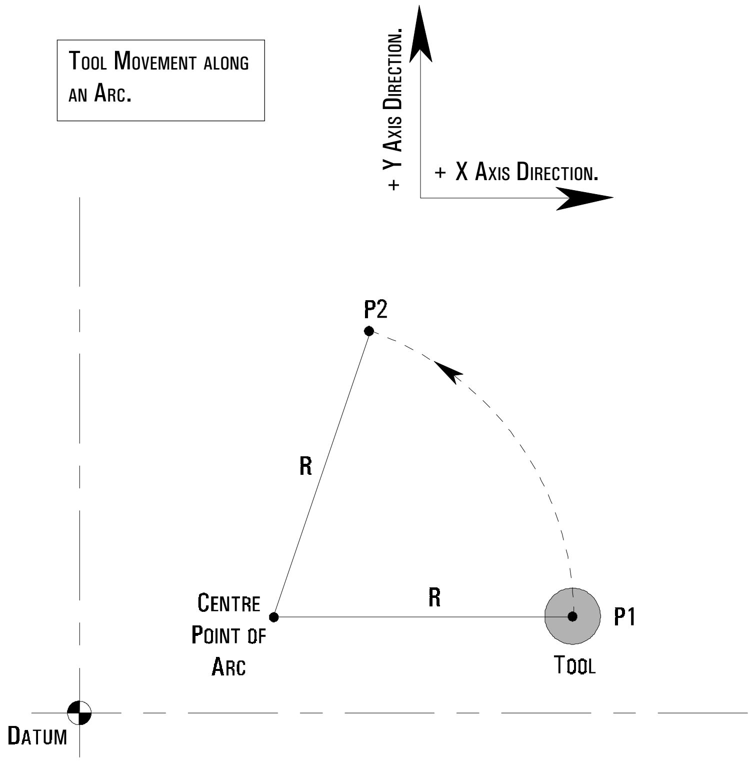


Tool Movement.

A) Tool Movement along an Arc.

Program command format:

GØ3 X _ _ _ _ Y _ _ _ _ R _ _ _ _ ; (P1 - P2)



The function of moving the tool along straight lines and arcs is called the *Interpolation*. Symbols of the programmed commands GØ1, GØ2 and GØ3 are called the *Preparatory* functions and specify the type of interpolation conducted in the control unit.

Note 1.

On an actual machine, the table moves in relation to the cutter. To make the command diagrams easier to understand, this manual assumes the tool moves with respect to the workpiece.

Feed Function.

The movement of the tool at a specified speed for cutting is called the Feedrate.

The feedrate is defined using the F address lettter followed by a numerical value.

Using the G2Ø code, the feedrate is defined in Inches per minute.

Using the G21 code, the feedrate is defined in Millimetres per minute.

M Codes (Miscellaneous Functions).

Miscellaneous functions, called *M codes*, are used by the CNC to command on/off signals to the machine functions. ie, MØ3 - spindle forward (CW), MØ5 - spindle stop, etc.....

The functions allocated to lower M code numbers are constant in most CNC controls, although the higher M code number functions can vary from one make of controller to the next.

An M code is defined using the M address letter and a two digit number as follows,

M ØØ

Address. Two digit number.

Spindle Speed Function (Cutting Speed).

The rotational speed of the tool, with respect to the workpiece being cut, is called the spindle (or cutting) speed.

The spindle speed is defined using the S address letter, followed by a numerical value, signifying the spindle RPM (revolutions per minute).

The spindle speed value specified must fall between the machine tool RPM range for the command to be effective.

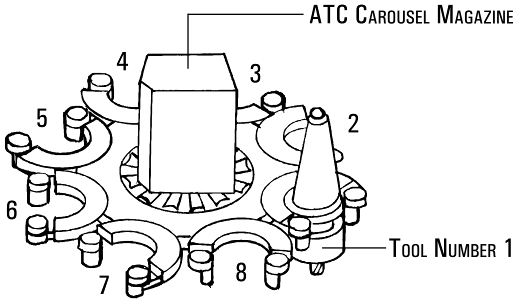
Note 1.

When a move command and an S code are specified in the same block, a simultaneous execution of the commands is performed.

Note 2.

Only one S code is allowed in each program block.

Tool Function.



Tool profiles can be changed during a program using the tool function command. Each tool profile is assigned a number, which in the case of an ATC (Automatic Tool Changer) will also coincide with one of the free bays on its carousel magazine.

The tool number is defined using the address letter T, followed by a number assigned to the tool profile. To command a tool change, the MØ6 code would precede the number of the "new" tool required.

For example,

MØ6 TØ1

This command is read perform a tool change to tool number Ø1.

Note 1.

The MØ6 code (automatic tool change) must immediately precede the T code within the program block.

Note 2.

Only one T code is allowed in each program block.

Note 3.

If the machine control reads an MØ6 T _ _ command when running in Automatic Mode, the three axes will drive to the tool change position and the spindle will stop. At this point, the tool change will be performed, if an ATC is fitted. This will always happen, irrespective of the tool position when the tool change command is read from the program.

Tool Compensation (Tool Offset).

Generally, several different tool profiles are required to machine a workpiece, all of different diameters and lengths.

It would be very difficult to write a program that allowed for this difference in size between all the various tools. To account for this, the difference in diameter and length is measured, in advance, for all the tools that will be used. Essentially, this means that the cutting paths for all the tools now coincide. The values are entered into the offset file.

This tool offset is also called tool compensation.

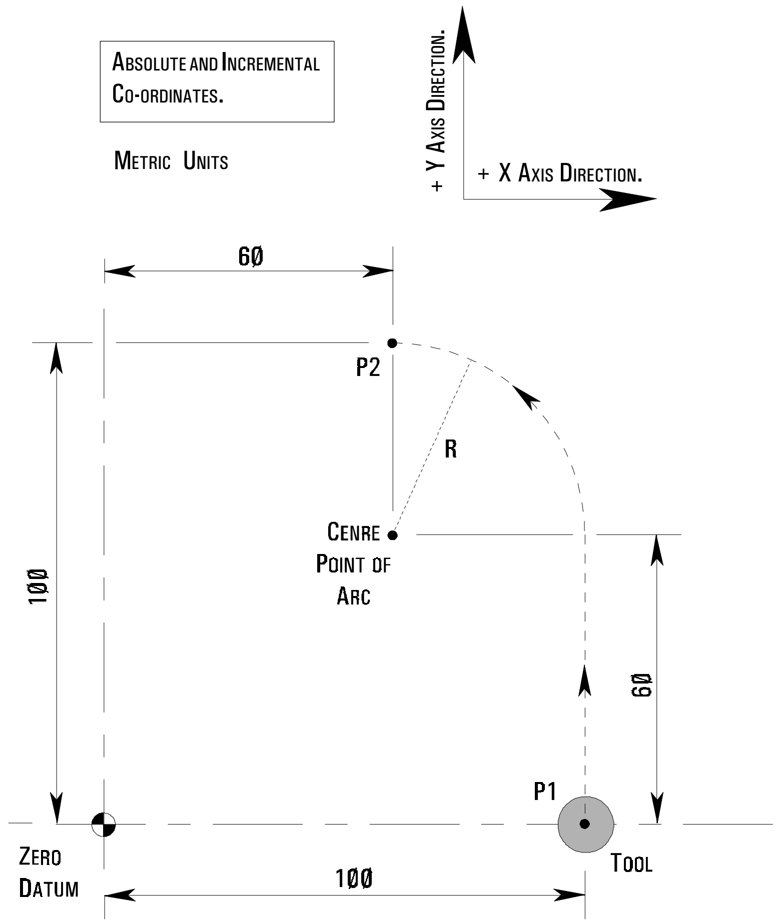
Note 1.

When a T code is read by the CNC, it will load the offset length for that particular tool. The code G41 or G42 (Tool Compensation Left or Right) must be programmed for the radius offset to used.

ABSOLUTE AND INCREMENTAL CO-ORDINATES.

The addresses X, Y and Z within a program, when G90 (Absolute co-ordinates) is active, relate to a co-ordinate position from the workpiece datum (the zero position).

The addresses X, Y and Z within a program, when G91 (Incremental co-ordinates) is active, relate to the individual axis movements required to reach the new position, from the last position reached by the tool.



The example move illustrated above can be written in two ways:

G90 Absolute co-ordinates selected

GØ1 Y60 F150 ;

G03 X60 Y100 R40 ;

G91 Incremental co-ordinates selected

G01 Y60 F150 ;

G03 X-40 Y40 R40 ;

Optional Block Skip.

When a forward slash mark (/) is followed by a block number (at the beginning of a block) and the block skip switch on the machine operator panel is set to "on", the block will be ignored in memory operation. When the block switch is set to "off", then the blocks indicated by the "/" marks will be considered as valid.

For example,

N3Ø X4Ø ;

/ N4Ø Y5Ø ;

/ N5Ø X7Ø ;

/ N6Ø Y9Ø ;

N7Ø .....

If the block skip switch is set to "on" in the above program example, then blocks indicated by the "/" mark are skipped.

Note 1.

A "/" mark must be specified at the start of the block. If it is placed elsewhere in the block, the information from the "/" mark to the ";" mark (the end of block mark) will be ignored, whilst the information before the "/" mark will be effective.

Tutorials and Comments.

If the program is written off-line with a qwerty keyboard, information relating to the program can be inserted within the program.

Tutorial information appear in the Tutorial dialog box of the machine controlling software (ie, the tutorial message "Now performing pocket cutting cycle" could be written to appear when the pocket cutting operation starts in the program).

Comments information only appear in the text of the program itself (ie, the comment "Tool 5 is 8mm slot cutting tool" could be inserted in the program for use as reference only).

G Codes (Preparatory Functions) - Introduction.

Preparatory functions, called *G codes*, are used to determine the geometry of tool movements and operating state of the machine controller; functions such as linear cutting movements, drilling operations and specifying the units of measurement.

They are normally programmed at the start of a block.

A G code is made from the G address letter and a two digit number as follows,

G ØØ

Address. Two digit number.

Modal and Non-modal G codes.

Additionally, G codes are split into two categories -

1) *Modal* (retained) G codes.

A modal G code, once programmed into a block, will affect any subsequent blocks of the program without having to be restated.

Additionally, modal G codes are further split into *groups* according to their task and function. A modal G code will remain active until another G code from the same group is programmed into a block, or it is cancelled.

For example,

GØ1 and GØØ are modal G codes from group 1:

```
GØ1 X _ _ _ _ ;  ┐
    Y _ _ _ _ ;  ├─ GØ1 is effective
    X _ _ _ _ ;  ┘  in this range.
GØØZ _ _ _ _ ;  ]── GØØ replaces the
                    GØ1 command.
```

2) *Non-modal* (one-shot) G codes.

A non-modal G code must be programmed into every block when it is required, ie, it is only effective in the block in which it is specified.

G Codes (Preparatory Functions) - Introduction Notes.

Note 1.

Remember there are two types of G code.
A modal G code is retained in memory - it is effective until another G code from the same modal group is commanded.
A non-modal G code is one-shot - it is effective only within the block in which it is specified.

Note 2.

It is not necessary to enter a modal G code in repetitive blocks within a program.
For example :
If all movements are GØ1 (linear cutting command) then GØ1 is entered on the first block and omitted from all subsequent blocks. This G code will remain active until an interpolation change is commanded (using GØØ, GØ2 or GØ3).

Note 3.

The machine controller has the ability to interpret a maximum of four G codes in one single block of data. However, these G codes must be from separate modal groups. When two or more G codes from the same group are specified in the same block, the CNC will only make the last stated G code from that modal group effective.

Notes for G Codes Listing shown right.

Note 1.

G codes marked with an * are set/reactivated as default values at machine power up and when the machine is reset or the emergency stop button is activated.

Note 2.

G codes from group Ø are non-modal (they must be programmed into every program block when required). All other G codes are modal (they remain active through subsequent program blocks, until replaced or cancelled by a G code from their particular group).

List of G Codes supported by Denford CNC Controls.

Note - Not all G codes apply to each machine.

G Code.	Group.	Function.
GØØ	1	Positioning (Rapid Traverse)
GØ1	1	Linear Interpolation (Cutting Feed)
GØ2	1	Circular Interpolation CW
GØ3	1	Circular Interpolation CCW
GØ4	Ø	Dwell, Exact Stop
G2Ø	6	Imperial Data Input (Inches)
G21	6	Metric Data Input (Millimetres)
G28	Ø	Reference Point Return
G4Ø	7	Cutter Compensation Cancel
G41	7	Cutter Compensation Left
G42	7	Cutter Compensation Right
G73	9	High Speed Peck Drilling Cycle
G74	9	Counter Tapping Cycle
G76	*9*	*Fine Boring Cycle(not recommended on Denford Machines)*
G8Ø*	9	Canned Cycle Cancel
G81	9	Drilling Cycle, Spot Boring
G82	9	Drilling Cycle, Counter Boring
G83	9	Deep Hole Peck Drilling Cycle
G84	9	Tapping Cycle
G85	9	Boring Cycle
G86	9	Boring Cycle
G87	*9*	*Back Boring Cycle(not recommended on Denford Machines)*
G89	9	Boring Cycle
G9Ø*	3	Absolute Zero Command
G91	3	Incremental Command
G94*	5	Feed per Minute
G95	5	Feed per Revolution
G98*	1Ø	Return to Initial Level in Canned Cycle
G99	1Ø	Return to R Point Level in Canned Cycle
G17Ø	Ø	Circular Pocket Canned Cycle
G171	Ø	Circular Pocket Canned Cycle
G172	Ø	Rectangular Pocket Canned Cycle
G173	Ø	Rectangular Pocket Canned Cycle

Code listing full and correct at the time of printing.

G Codes - GØØ

(Rapid Positioning/ Traverse).

The GØØ code executes a non cutting movement, at a rapid feedrate, to a specific co-ordinate position in the working area (operating under absolute co-ordinate movement) or when a certain distance from a previously stated position (under incremental co-ordinate movement) is programmed.

A GØØ command is written in the following format:

GØØ X _ _ _ _ Y _ _ _ _ Z _ _ _ _ ;

Rapid Positioning/ Traverse code.

X, Y and Z co-ordinate values.

The axis co-ordinate moves following a GØØ command can be programmed as either:

(i) absolute values (relative to a set datum point) following setting of the G9Ø code, or,

(ii) incremental values (relative to the last stated co-ordinate in the program) following setting of the G91 code.

Note 1.

The rate of movement is set by the manufacturer of the machine tool. The rate of movement can be reduced from 1ØØ% to Ø%, but only in increments of 1Ø%, by using the feed override controls (see specific machine operating manual).

Note 2.

The GØØ code freezes the tool radius compensation, codes G 41 and G42. If G41 or G42 are active when a GØØ command is programmed, the tool radius compensation will not function again until a GØ1, GØ2 or GØ3 command is programmed.

Note 3.

The GØØ code is modal and is therefore incompatible with GØ1, GØ2 and GØ3 codes in the same block.

Note 4.

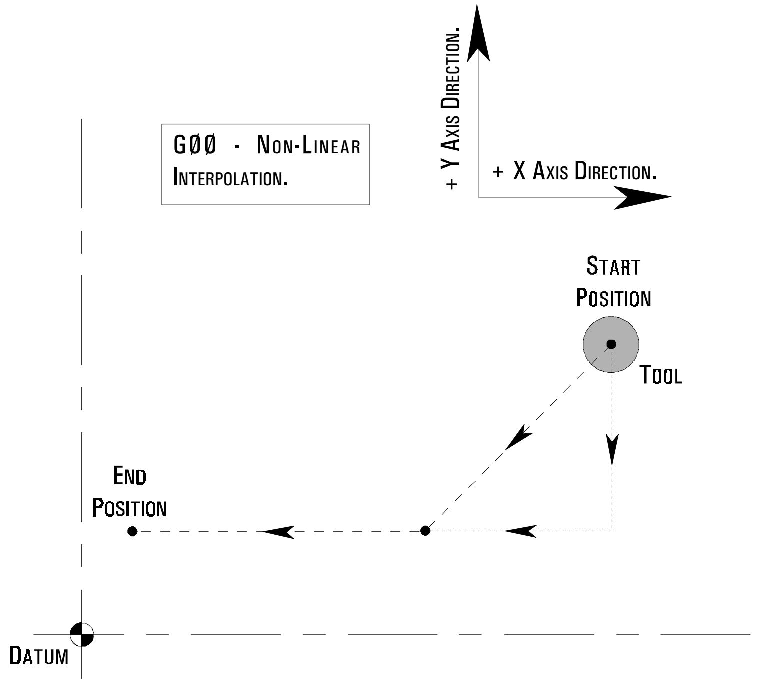
The GØØ code can be written into a program in two ways:
GØØ or GØ.

G Codes - GØØ

(Rapid Positioning/ Traverse).

Note 5.

On instruction to perform the GØØ command, the three slides (the X, Y and Z axes) move completely independant of each other at a maximum feedrate, along a non-vector (sometimes called a non-linear) type path.



In the above example, the GØØ command has instructed the X and Y slides to begin moving, both at a maximum feedrate. When both slides begin moving the tool will appear to traverse diagonally, a composite movement of the two axes moving together. When one axis reaches its finishing co-ordinate, the other axis will continue to move until it reaches its own finishing co-ordinate. This gives the impression that the tool "changes" direction.

G CODES - GØ1 (LINEAR INTERPOLATION).

The GØ1 code executes a cutting movement following a straight line, at a set feedrate.

A GØ1 command is written in the following format:

GØ1 X _ _ _ _ Y_ _ _ _ Z _ _ _ _ ;

Linear Interpolation code.

X, Y and Z co-ordinate values.

The feedrate value programmed into the GØ1 command is the actual feedrate along the proposed tool path, not the feedrate of each axis/slide.

On single axis moves (ie, the tool moves exactly parallel to the X, Y or Z axis direction), the slide will feed at the rate stated in the GØ1 command.

On two or three axis moves (ie, the tool is moving in a straight diagonal line), all the slides have to operate exactly the same length of time, in order to produce a single diagonal (vector) move. The machine controller will calculate the separate feedrates for the X, Y and Z slides, enabling the actual vector feedrate to equal that stated in the GØ1 command.

The axis co-ordinate moves following a GØ1 command can be programmed as either:

(i) absolute values (relative to a set datum point) following setting of the G9Ø code, or,

(ii) incremental values (relative to the last stated co-ordinate in the program) following setting of the G91 code.

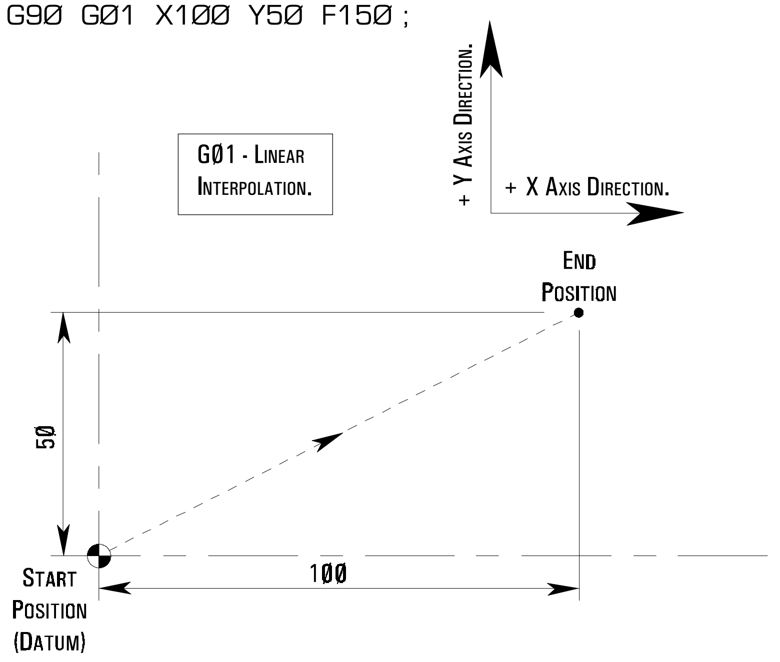
G Codes -

GØ1

(Linear Interpolation).

Example of programming a GØ1 linear interpolation:

G9Ø GØ1 X1ØØ Y5Ø F15Ø ;



Note 1.

In the example shown above, G9Ø, GØ1 and the F _ _ _ _ feedrate are modal and can be continued onto the next block (without having to be restated) if required.

Note 2.

The programmed feedrate F _ _ _ _ can be varied in Automatic Mode from Ø% to 15Ø% by using the feed override controls (see specific machine operating manual). 1ØØ% is the programmed feedrate.

Note 3.

When there is no feedrate programmed within the part program, the CNC will set a feedrate of 1Ø millimetres per minute in G21 Metric Data Mode, or Ø.4 inches per minute in G2Ø Imperial Data Mode.

Note 4.

The GØ1 code is modal and is therefore incompatible with GØØ, GØ2 and GØ3 codes in the same block.

Note 5.

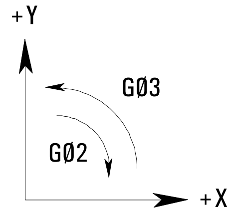
The GØ1 code can be written into a program in two ways:
GØ1 or G1.

G Codes - GØ2 GØ3 (Circular Interpolation).

The GØ2 code executes a cutting movement following a clockwise circular path, at a set feedrate.

The GØ3 code executes a cutting movement following a counterclockwise circular path, at a set feedrate.

The definitions of clockwise (GØ2) and counterclockwise (GØ3) are fixed according to the system of co-ordinates in the diagram below.



Right hand Cartesian Co-ordinate System.

When programming arcs using absolute values (G9Ø), the X and Y values describe the end point of the arc, in relation to the datum position of the workpiece. The arc end point is sometimes referred to as the target position.

When programming arcs using incremental values (G91), the X and Y values relate to the distance moved along the X and Y axes, from the start point of the arc to the end point of the arc. The sign of the X and Y axis moves (+/-) will depend on the movement of the machine slides in relation to their start position.

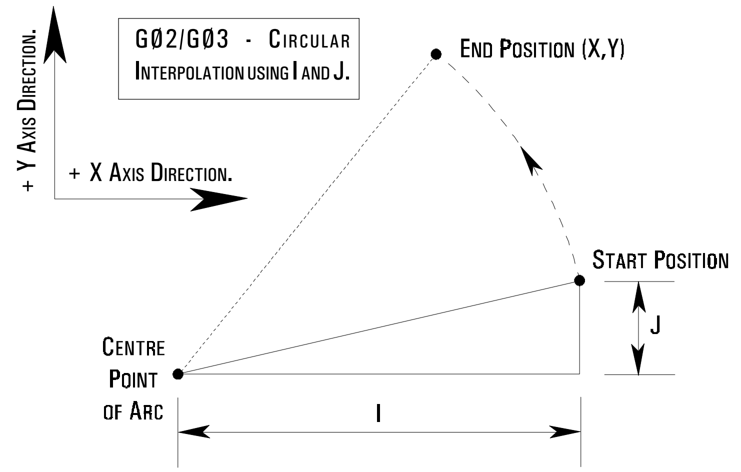
G Codes - GØ2 GØ3 (Circular Interpolation).

I and J

To program an arc when only the arc centre is given (the radius is unknown) use the address letters I and J.

I relates to the address X and is the incremental value and direction (+/-) from the start point of the arc in the X axis to the arc centre (see diagram below).

J relates to the address Y and is the incremental value and direction (+/-) from the start point of the arc in the Y axis to the arc centre (see diagram below).



G CODES - GØ2 GØ3 (CIRCULAR INTERPOLATION).

The format to program a circular interpolation in Cartesian co-ordinates is written as follows :

There are four ways to program a clockwise circular path using the GØ2 code:

G9Ø GØ2 X _ _ _ _ Y _ _ _ _ R _ _ _ F _ _ _ ;
G9Ø GØ2 X _ _ _ _ Y _ _ _ _ I _ _ _ J _ _ _ F _ _ _ ;
G91 GØ2 X _ _ _ _ Y _ _ _ _ R _ _ _ _ F _ _ _ _ ;
G91 GØ2 X _ _ _ _ Y _ _ _ _ I _ _ _ J _ _ _ F _ _ _ ;

There are four ways to program an anticlockwise circular path using the GØ3 code:

G9Ø GØ3 X _ _ _ _ Y _ _ _ _ R _ _ _ F _ _ _ ;
G9Ø GØ3 X _ _ _ _ Y _ _ _ _ I _ _ _ J _ _ _ F _ _ _ ;
G91 GØ3 X _ _ _ _ Y _ _ _ _ R _ _ _ _ F _ _ _ _ ;
G91 GØ3 X _ _ _ _ Y _ _ _ _ I _ _ _ J _ _ _ F _ _ _ ;

where,

GØ2 defines the clockwise direction circular interpolation.

GØ3 defines the counterclockwise direction circular interpolation.

G9Ø X _ _ _ _ Y _ _ _ _ defines the arc end point in the work co-ordinate system.

G91 X _ _ _ _ Y _ _ _ _ defines the signed distance of the arc end point from the arc start point.

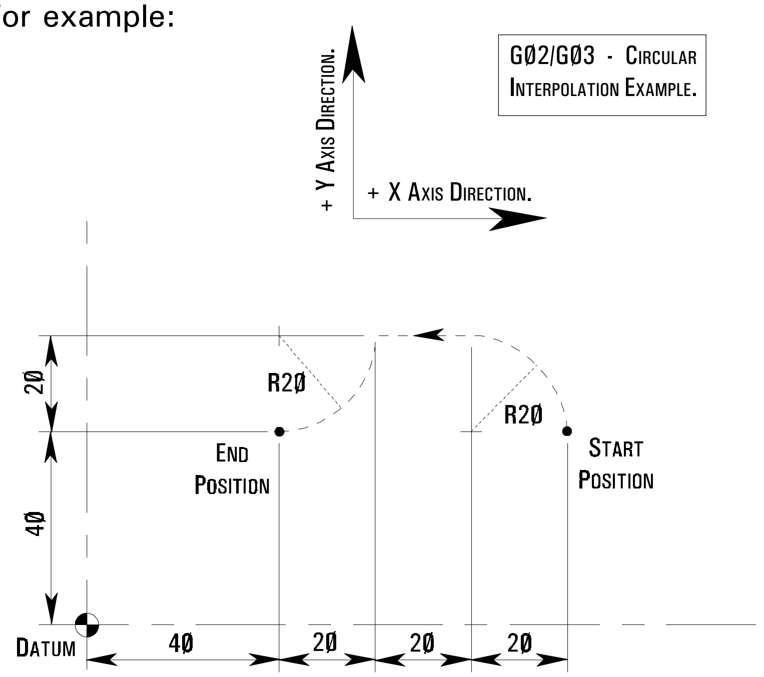
I _ _ _ J _ _ _ defines the signed distance of the arc start point from the centre point of the arc.

R _ _ _ defines the length of the arc radius.

F _ _ _ defines the feedrate along the arc.

G Codes - GØ2 GØ3 (Circular Interpolation).

For example:



The above tool path can be programmed as follows (In absolute mode, G9Ø):

GØ1 X1ØØ Y4Ø F125 ;

GØ3 X8Ø Y6Ø I-2Ø ;

GØ1 X6Ø ;

GØ2 X4Ø Y4Ø I-2Ø ;

or,

GØ1 X1ØØ Y4Ø F125 ;

GØ3 X8Ø Y6Ø R2Ø ;

GØ1 X6Ø ;

GØ2 X4Ø Y4Ø R2Ø ;

The above tool path can be programmed as follows (In incremental mode, G91):

GØ3 X-2Ø Y2Ø I-2Ø ;

GØ1 X-2Ø ;

GØ2 X-2Ø Y-2Ø I-2Ø ;

or,

GØ3 X-2Ø Y2Ø R2Ø ;

GØ1 X-2Ø ;

GØ2 X-2Ø Y-2Ø R2Ø ;

G Codes -

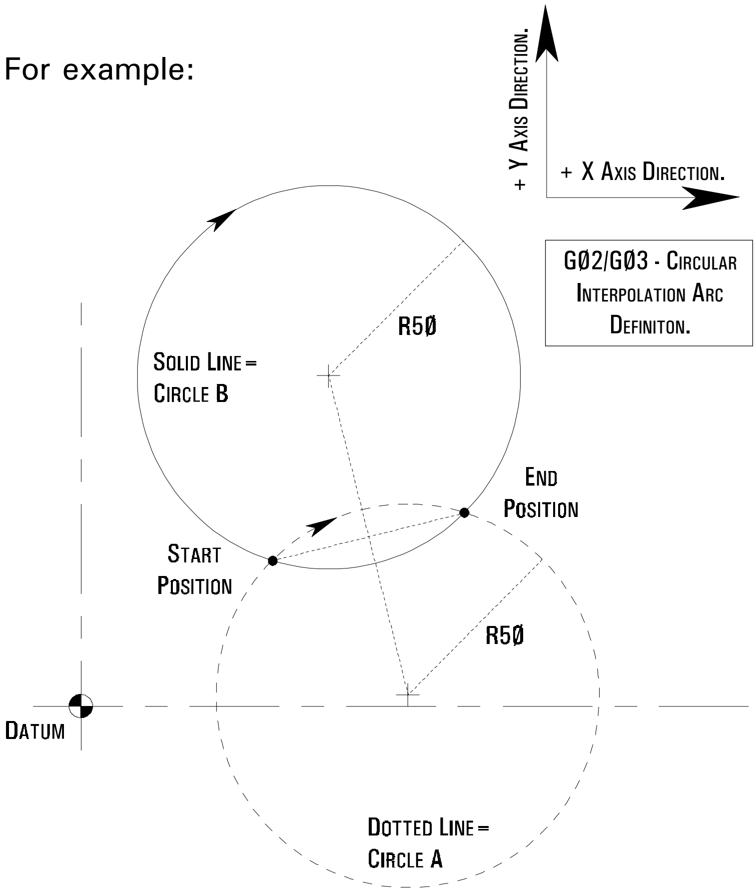
GØ2

GØ3

(Circular Interpolation).

When using the address R _ _ _ _ , two types of arcs can be considered. One arc is less than 18Ø degrees, whilst the other arc is greater than 18Ø degrees. When an arc exceeding 18Ø degrees is commanded, the radius value (R _ _ _ _) must be specified as a negative signed (-) value.

For example:



The above tool path for an arc less than 18Ø degrees (circle A) can be programmed as follows (In absolute mode, G9Ø):

GØ2 X8Ø Y4Ø R5Ø F125 ;

The above tool path for an arc greater than 18Ø degrees (circle B) can be programmed as follows (In absolute mode, G9Ø):

GØ2 X8Ø Y4Ø R-5Ø F125 ;

G CODES - GØ2 GØ3 (CIRCULAR INTERPOLATION).

NOTE 1.

When programming arcs using the address R (arc radius), the value of R must be equal to, or greater than half the longest distance travelled by either axis.

NOTE 2.

IØ and JØ can be omitted from program lines.

NOTE 3.

When X or Y are omitted from program lines, the arc end point is located at the same position as the arc start point and the arc centre is commanded by I or J, an arc of 36Ø degrees (ie, a complete circle) is assumed. When R is used, an arc of Ø degrees is assumed and the cutter does not move.

NOTE 4.

When I, J and R addresses are specified simultaneously in the same program line, the address R takes precedence and the other addresses are ignored.

NOTE 5.

A GØ2 code can be written into a program in two ways.
GØ2 or G2.
A GØ3 code can be written into a program in two ways.
GØ3 or G3.

G Codes - GØ4 (Dwell).

The GØ4 code is used to enter a set time delay into the program (called a "dwell").

A GØ4 command is written in the following format:

GØ4 X _ _ _ _ ;

or GØ4 P _ _ _ _ ;

where the dwell value is programmed using the address letters X (time in seconds) or P (time in 1/1ØØØ seconds), followed by a number indicating this dwell value.

For example :

```
GØ4 X1.5 ;
```

This command is read perform a dwell of 1.5 seconds duration.

```
GØ4 P25ØØ ;
```

This command is read perform a dwell of 2.5 seconds duration.

Note 1.

A decimal point cannot be used with the address P.

Note 2.

The dwell is performed at the start of the block in which it is programmed.

Note 3.

The dwell begins when the commanded feedrate of the previous block reaches zero.

Note 4.

The maximum value of a dwell time is 999 seconds.

Note 5.

GØ4 is a non-modal G code. It is only active in the block in which it is programmed.

Note 6.

A GØ4 code can be written into a program in two ways.
GØ4 or G4.

G Codes - G2Ø G21 (Imperial /Metric Data Input).

The machine controller can be programmed in either Imperial (inch) unit input (G2Ø) or Metric (millimetre) unit input (G21). The standard format for a CNC part program is to write the G2Ø or G21 code in the first block of the program.

G code.	Type.	Units.	Lowest input value.
G2Ø	Imperial	Inch	Ø.ØØØ1 inch
G21	Metric	Millimetre	Ø.ØØ1 mm

The unit systems of the following items are changed depending on whether G2Ø or G21 is set.

1) Positioning commands (X, Y and Z).

2) Incremental movement distances.

3) Feedrates commanded by the F code.

4) Offset values.

Note 1.

The status of G2Ø or G21 in the machine controller is dependant on the option that is saved to the disc.

Note 2.

A G2Ø code must not be changed for a G21 code (or vice versa) during the program.

Note 3.

When switching between G2Ø and G21, the offsets must be set according to the units of measurement being used.

Note 4.

G 2Ø and G21 are both modal G codes within the same modal group.

G CODES - G28 (REFERENCE POINT RETURN).

The reference point is a fixed position on the machine, to which the tool can be moved.

On machines fitted with Denford milling software, this point is also used as the Home position, the point used by the machine to set the limits of movement for the X, Y and Z slides.

A G28 code instructs the tool to automatically move to this reference point.

A G28 command is written in the following format :

G9Ø G28 X _ _ _ _ Y _ _ _ _ Z _ _ _ _ ;

or G91 G28 X _ _ _ _ Y _ _ _ _ Z _ _ _ _ ;

where X, Y and Z can be used to indicate an intermediate point, through which the tool will pass, before continuing to the reference point.

This intermediate point allows the tool to be programmed to follow a more "predictable" path, keeping it sufficiently clear from any part of the machine or billet it could hit when moving to the reference point.

The move to any intermediate point and the reference point are performed at a rapid traverse rate, using a non-vector (non-linear) type path, ie, the tool may appear to "change" direction due to the non-vector type positioning being used.

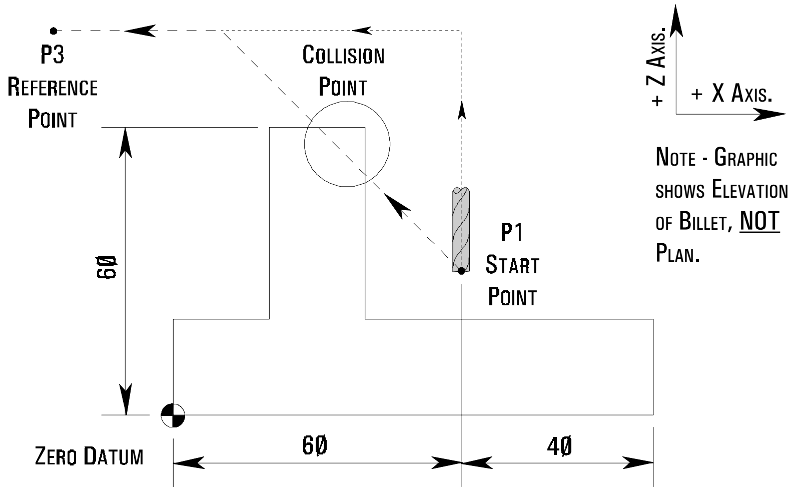
G Codes - G28

(Reference Point Return).

Note 1.

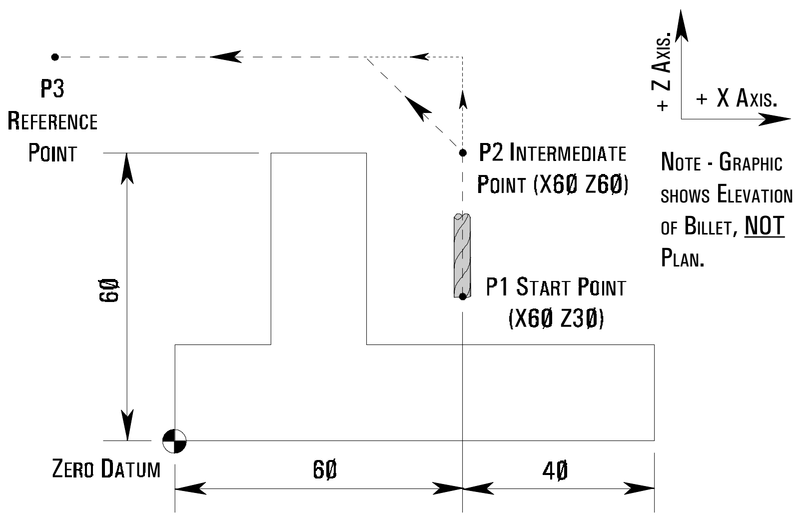
The diagram below shows how the tool could collide with the billet when manoeuvring towards the reference point. This is a result of the non-vectored movements forcing the tool to follow a path which "cuts" through the edge of the billet.

G28 - Potential Collision through not using an Intermediate position.



To avoid this collision, the tool is sent on a path which includes the additional, or intermediate, point P2. The intermediate point is used to allow the tool to move completely clear from the billet, before continuing onto the reference point, P3, shown below.

G28 - Avoiding Collision by using an Intermediate position.



The above toolpath can be programmed as follows (In absolute mode, G9Ø):

G9Ø G28 X6Ø Z6Ø ;

The above toolpath can be programmed as follows (In incremental mode, G91):

G91 G28 XØ Z4Ø ;

G Codes - G28

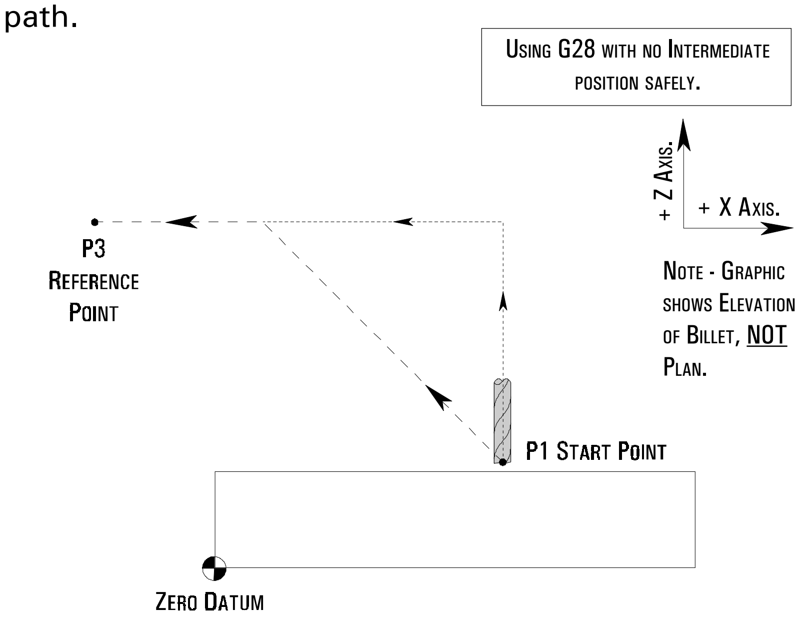
(Reference Point Return).

Note 2.

In the diagram below, the tool is in a position (P1) where no collision is possible. The intermediate point, in this case, is not required, so the block can be written as follows (In incremental mode, G91):

```
G91 G28 XØ YØ ZØ ;
```

The intermediate point co-ordinates are still stated, but all their values are set to zero, indicating no axis movement. Therefore, the tool will move from point P1 to the reference point, P3, along a non-vector type path.



Note 3.

G28 is a non-modal G code. It is only active in the block in which it is programmed.

G Codes - G4Ø G41 G42 (Cutter Compensation).

The collection of G4Ø, G41 and G42 codes allow the machine controller to produce very accurate arcs and tapers on the billet, by compensating for the tool radius.

Complex workpiece shapes are therefore programmed with cutter compensation mode active. The radius of the tool (the offset amount) is measured, then entered into the offset file in the machine controller. Once set, the tool path can be offset by this value, regardless of the program.

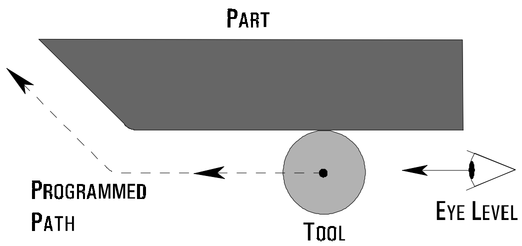
Work Position and Movement Command.

When tool nose radius compensation is required in a CNC program, the position of the billet in respect to the tool must be specified using the table below.

G code	Direction	Tool Path
G4Ø	Cancel	Movement along programmed path
G41	Left hand	Movement on the left hand side of the programmed path
G42	Right hand	Movement on the right hand side of the programmed path

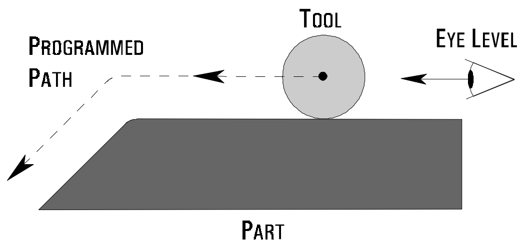
The two diagrams below illustrate the direction of compensation codes G41 and G42, in relation to your eye level.

G41 - Left Hand



The tool is positioned on the left hand side of the part, as seen following the direction of movement, from behind the tool.

G42 - Right Hand



The tool is positioned on the right hand side of the part, as seen following the direction of movement, from behind the tool.

G Codes - G4Ø G41 G42 (Cutter Compensation).

Cutter Compensation Start-up (G41 - G42).

The operation instructing a machine to switch to cutter compensation mode is called the *start-up* block, or *ramping on* block. The start-up block is used to allow the tool time to change from moving along the programmed path line to following either side of the programmed path line.

The start-up block should satisfy the following points:

1) A G41 or G42 code must be contained in the block, or specified in the previous block.
2) A GØ1 X, Y, or X and Y move is specified in the block and the distance of the linear move must be greater than the tool radius.
3) The tool radius value, "R", entered into the tool offsets table must not be ØØ.

Note 1.

A GØ2 or GØ3 circular interpolation command cannot be specified in the start-up block.

Note 2.

In cutter compensation start-up, two blocks are read into the machine controller. The first block is performed and the second block is entered and held in memory.
In subsequent compensation moves, two blocks are read in advance, so the machine controller has the block currently being performed and the next two blocks in memory.
This is because cutter compensation always needs to know what happens in the move following the one being currently performed. The machine controller can plan ahead to calculate the correct end position for the current move, that will also be the correct start position allowing for cutter compensation, for the next move.

Note 3.

The codes G4Ø, G41 and G42 are modal, belonging to the same modal family. They are incompatible with each other on the same block.

G Codes - G4Ø G41 G42 (Cutter Compensation).

Cancellation of Cutter Compensation (G4Ø).

The G4Ø code is used to cancel cutter compensation.

A G4Ø command can only be performed in a block in which a linear move (ie, GØØ, GØ1, G28) is programmed.

Note 1.

Following the machining of an internal pocket, it is recommended that the Z axis is withdrawn by using the GØ1 command, to a position clear of the workpiece, before the cutter compensation mode is cancelled.

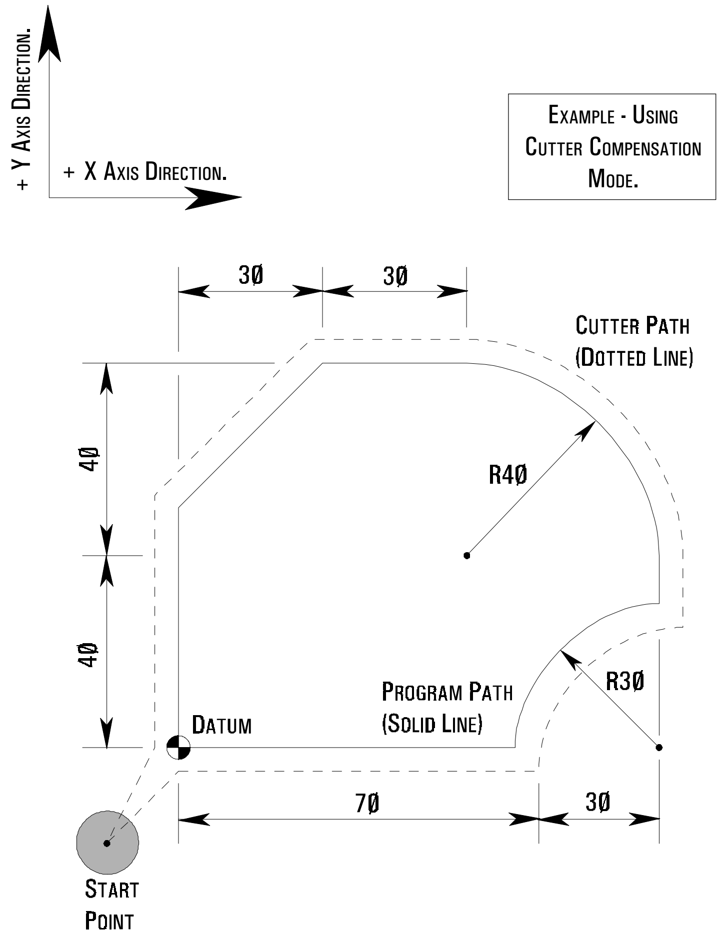
Note 2.

The machine controller enters compensation cancel mode automatically when :

1) the machine power is first switched on.
2) the reset button on the CRT/MDI controller panel is pressed.
3) a program is forced to end by performing an MØ2 or M3Ø command.

G Codes - G4Ø G41 G42 (Cutter Compensation).

The following part program for a finishing pass shows the recommended method for start-up and cancellation of cutter compensation :




```
O ØØ1Ø
NØØ1Ø   GØØ X-15 Y-15 ;
NØØ2Ø   G41 XØ YØ F1ØØ ;  (Start-Up Move)
NØØ3Ø   Y4Ø ;
NØØ4Ø   X3Ø Y8Ø ;
NØØ5Ø   X6Ø ;
NØØ6Ø   GØ2 X1ØØ Y4Ø R4Ø ;
NØØ7Ø   GØ1 Y3Ø ;
NØØ8Ø   GØ3 X7Ø YØ R3Ø ;
NØØ9Ø   XØ ;
NØ1ØØ   X-15 Y-15 ;  (Cancellation Move)
```

G Codes - G73-G89 (Canned Cycles).

A canned cycle simplifies the program by replacing complex machining sequences, programmed by several blocks of information, with just one or two blocks.

Generally, a canned cycle consists of a sequence of six operations, as shown below:

Operation 1 - Positioning of the X and Y axes.

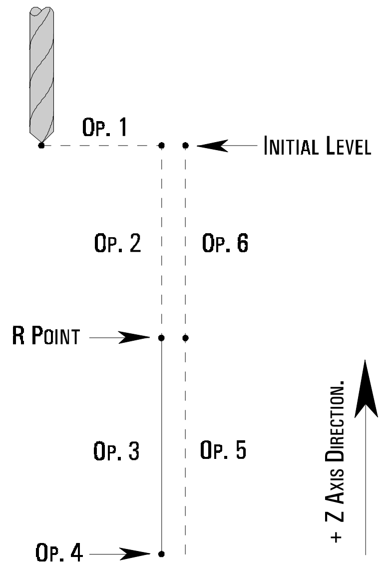
Operation 2 - Rapid traverse in the Z axis to the "R" point.

Operation 3 - Hole machining procedure.

Operation 4 - Operation at bottom of hole.

Operation 5 - Retraction to R point.

Operation 6 - Rapid traverse in the Z axis to the Initial level.



Hole positioning is performed in the X and Y axis hole machining is performed in the Z axis.

G Codes - G73-G89 (Canned Cycles).

There are three command modes for canned cycles, as follows:

1) Data Format (G9Ø and G91).

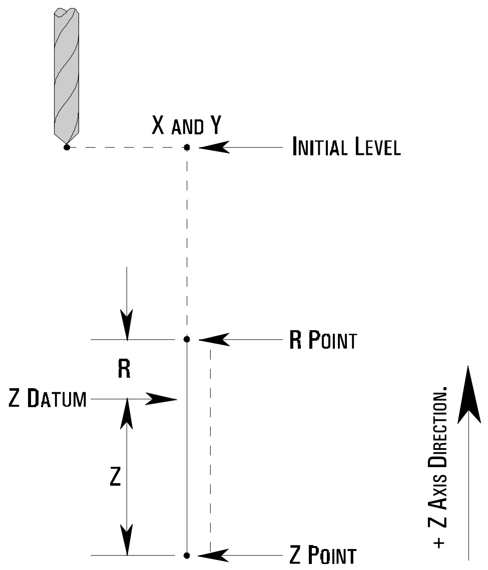
2) Return Point Level (G98 and G99).

3) Cycle Mode (G73 to G89).

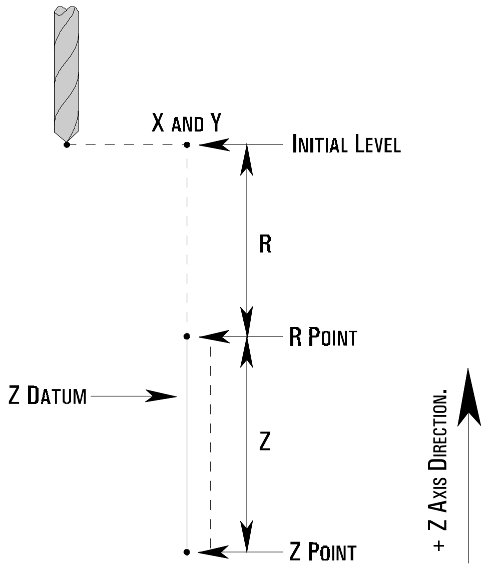
Data Format Command Modes.

The data format used in canned cycles is specified by the codes G9Ø and G91, as shown below:

G9Ø - Absolute Data Format.



G91 - Incremental Data Format.

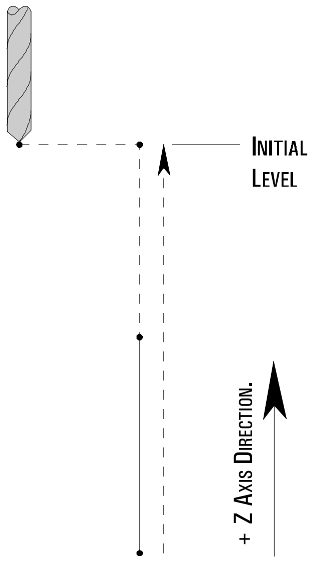


G Codes - G73-G89 (Canned Cycles).

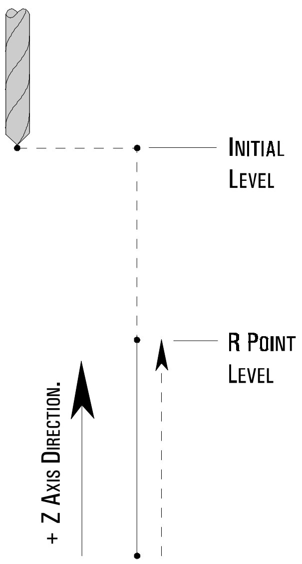
Return Point Level Command Modes.

The return point position of the tool (ie, to the Initial Level, or the R Level) is specified by the codes G98 and G99, as shown below:

G98 - Initial Point Level Return.



G99 - R Point Level Return.



The Initial level refers to the absolute value of the Z axis, at the time of change from the positioning mode top the canned cycle mode.

G Codes - G73-G89 (Canned Cycles).

The format for machining data in a canned cycle is written as follows:

(G9Ø) (G98)
or or G.... X.... Y.... Z.... R.... P.... Q.... K.... F.... ;
(G91) (G99)

where,

G.... is defined as the canned cycle.

X.... Y.... is defined as the hole position, in absolute or incremental value.

Z.... is defined as the distance from the R point to the bottom of the hole in incremental mode, or the position of the hole bottom in absolute mode.

R.... is defined as the distance from the initial level to the R point level in incremental mode, or the position of the Z datum in relation to the R point level in absolute mode.

P.... is defined as the dwell time to be performed at the bottom of the hole (see the GØ4 code for more details).

Q.... is defined as the cut-in distance value or shift value (Note - this is always specified as an incremental value).

K.... is defined as the number of repeats, for a series of holes. When not specified, K=1.

F.... is defined as the feedrate for machining.

Note 1.

The addresses P and Q are omitted within some canned cycles.

Note 2.

Once the drilling data has been specified and read into the machine controller, it is retained until it is either changed, or the canned cycle cancelled. All the required data must be specified when the canned cycle is started and only the data to be changed has to be specified during the cycle.

G Codes - G73-G89 (Canned Cycles).

The following example shows a canned cycle for drilling 4 holes, where the third hole is to be machined 10mm deeper:

G90 G99 G81 X10 Y10 Z-15 R2 F100 ;

X20 ; (X axis move)

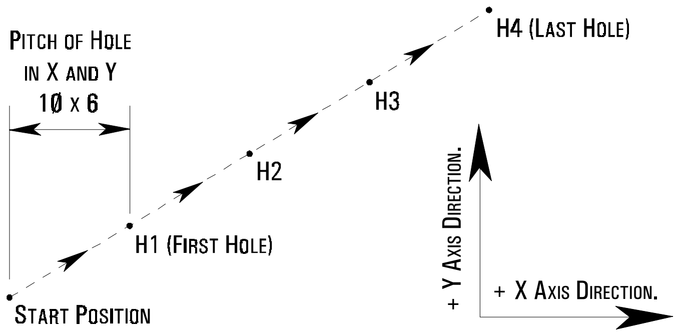
X30 Z-25 ; (X and Z change)

X40 Z-15 ; (X and Z change)

G80 ; (Cancel)

The following example shows a repeat canned cycle:

G91 G99 G81 X10 Y6 Z-10 R-8 K4 F100 ;

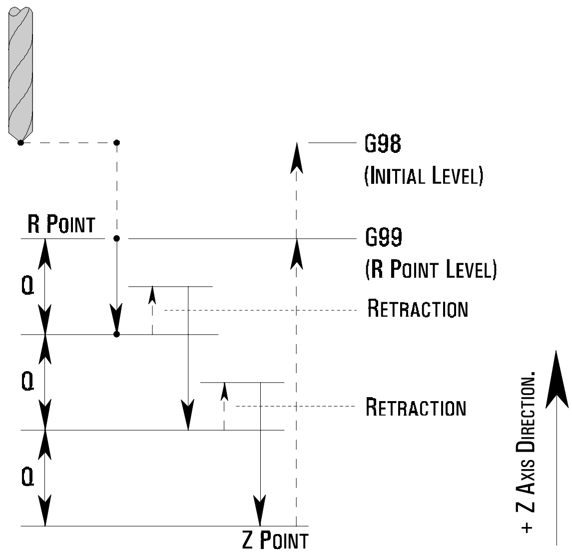


G Codes - G73 (High Speed Peck Drilling).

A G73 (High Speed Peck Drilling) command is written in the following format:

(G9Ø) (G98)
or or G73 X.... Y.... Z.... Q.... F.... ;
(G91) (G99)

please refer to page 52 for the variable definitions.



When machining, the drill is positioned at the co-ordinate point of the first hole, for the X and Y axes and at the initial level, for the Z axis. The G73 command is then read into the machine controller and the cycle begins. The drill will rapid traverse to the R point level and begin to feed in, until a cut-in distance of Q is attained. At this point, the drill will retract a small distance (set within the machine controller). A cut-in distance of Q at the same feedrate will begin again, followed by a similar retraction. These movements will continue until the total Z depth has been reached. The drill will rapid traverse out to the Initial level, if a G98 code is programmed within the cycle, or to the R point level, if a G99 code is programmed within the cycle. At this point the next block is read into the machine controller. If this block contains an X, Y or X and Y co-ordinate the drill will position itself at that point and the high speed peck drilling cycle will begin again.

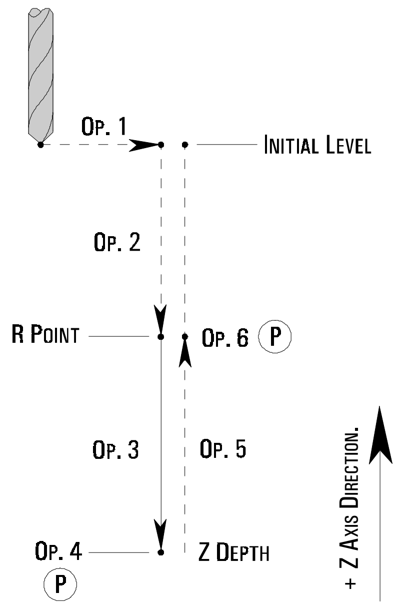
G Codes - G74

(Counter Tapping).

A G74 (Counter/Left Hand Tapping) command is written in the following format:

(G9Ø) (G98)
or or G74 X.... Y.... Z.... P.... R.... F.... ;
(G91) (G99)

please refer to page 52 for the variable definitions.



Sequence of moves:

Op 1) Rapid position to X, Y and Z (the Initial level).

Op 2) Rapid traverse to R point level.

Op 3) Feed to Z depth.

Op 4) Dwell P (time for spindle stop and start CW direction).

Op 5) Feed to R point level.

Op 6) Dwell P (time for spindle stop and start CCW direction).

If the G98 code is programmed within the cycle, the next move will be a rapid traverse to the Initial level. If the G99 code is programmed within the cycle, there will be no movement.

Note 1.

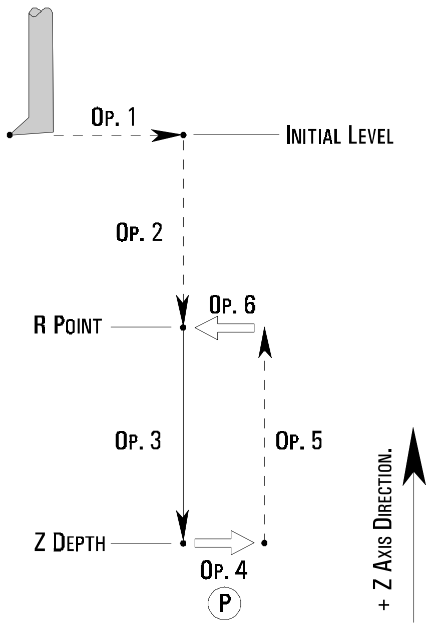
F (Feed) = RPM x Pitch.

G Codes - G76 (Fine Boring).

A G76 (Fine Boring) command is written in the following format:

(G9Ø) (G98)
or or G76 X.... Y.... Z.... R.... P.... Q.... F.... ;
(G91) (G99)

please refer to page 52 for the variable definitions.



Sequence of moves:

Op 1) Rapid position to X, Y and Z (the Initial level).

Op 2) Rapid traverse to R point level.

Op 3) Feed to Z depth.

Op 4) Dwell P (time for spindle stop and move Q value).

Op 5) Feed to R point level.

Op 6) Move back Q value.

The above moves vary depending on the setting of the codes G98 and G99.

Note 1.

THIS CYCLE CAN ONLY BE USED ON A MACHINE FITTED WITH A SPINDLE CAPABLE OF ORIENTATION. BECAUSE THE TOOL MOVES WITHIN THE HOLE AFTER SPINDLE STOP TO FACE THE OPPOSITE DIRECTION.

G Codes - G8Ø (Canned Cycle, Cancel).

Some of the addresses used within a canned cycle are modal (Z, P, Q and R), so their respective values are retained in the machine controller memory after the cycle has finished. The canned cycle must be cancelled, automatically removing these modal values, before the next canned cycle can be programmed into the machine controller.

This is achieved by programming a G8Ø code, following the last block of the canned cycle within the part program.

Note 1.

The G8Ø code is active when:

1) the machine power is first switched on.
2) the reset button on the CRT/MDI controller panel is pressed.
3) the Emergency Stop button is pressed.

G Codes -

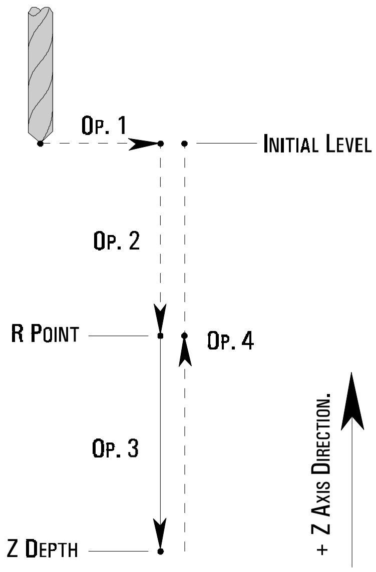
G81

(Drilling - Spot Boring).

A G81 (Drilling - Spot Boring) command is written in the following format:

(G9Ø) (G98)
 or   or   G81 X.... Y.... Z.... R.... F.... ;
(G91) (G99)

please refer to page 52 for the variable definitions.



Sequence of moves:

Op 1) Rapid position to X, Y and Z (the Initial level).

Op 2) Rapid traverse to R point level.

Op 3) Feed to Z depth.

Op 4) Rapid traverse to Initial level (G98) or R point level (G99).

G Codes -

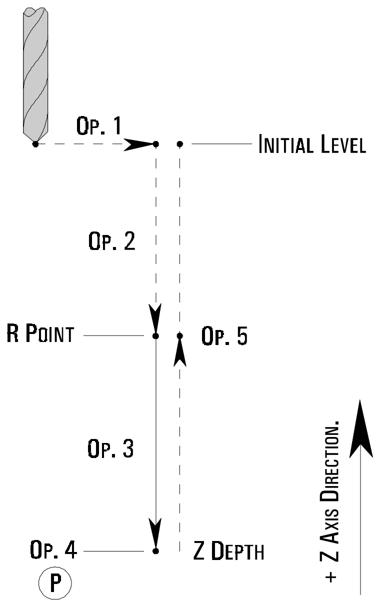
G82

(Drilling - Counter Boring).

A G82 (Drilling - Counter Boring) command is written in the following format:

(G9Ø) or (G91)	(G98) or (G99)	G82 X.... Y.... Z.... P.... R.... F.... ;

please refer to page 52 for the variable definitions.



Sequence of moves:

Op 1) Rapid position to X, Y and Z (the Initial level).

Op 2) Rapid traverse to R point level.

Op 3) Feed to Z depth.

Op 4) Dwell for value P.

Op 5) Rapid traverse to Initial level (G98) or R point level (G99).

G Codes -

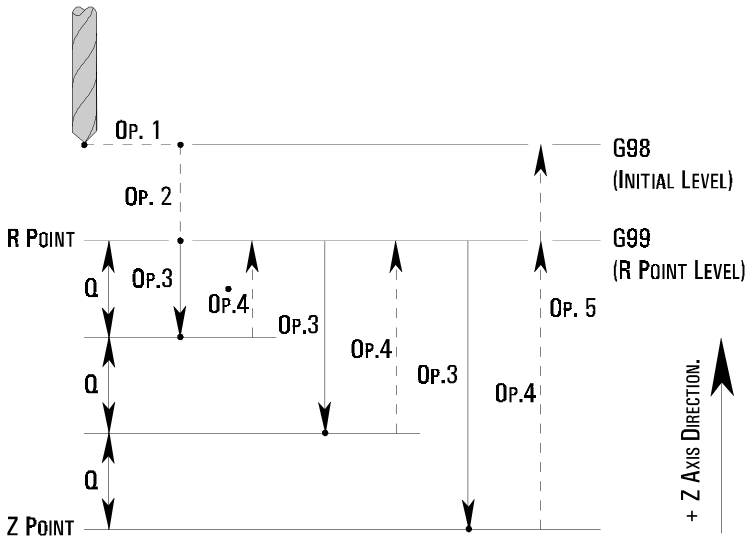
G83

(Deep Hole Peck Drilling).

A G83 (Deep Hole Peck Drilling) command is written in the following format:

(G9Ø) (G98)
or or G83 X.... Y.... Z.... Q.... R.... F.... ;
(G91) (G99)

please refer to page 52 for the variable definitions.



Sequence of moves:

Op 1) Rapid position to X, Y and Z (the initial level).

Op 2) Rapid traverse to R point level.

Op 3) Feed in to the value of Q.

Op 4) Rapid traverse out to R point. Rapid traverse back to within 1mm of depth of Q cut. Operation moves 2 and 4 are repeated until Z depth is reached.

Op 5) Rapid traverse to Initial level (G98) or R point level (G99).

G Codes -

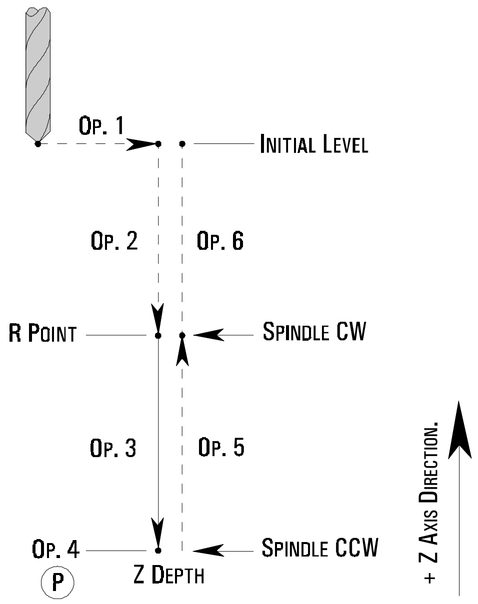
G84

(Tapping).

A G84 (Tapping) command is written in the following format:

(G9Ø) (G98)
or or G84 X.... Y.... Z.... R.... P.... F.... ;
(G91) (G99)

please refer to page 52 for the variable definitions.



Sequence of moves:

Op 1) Rapid position to X, Y and Z (the initial level).

Op 2) Rapid traverse to R point level.

Op 3) Feed to Z depth.

Op 4) Dwell P (time for spindle stop and start CCW direction).

Op 5) Feed to R point level.

Op 6) Dwell P (time for spindle stop and start CW direction).

If the G98 code is programmed within the cycle, the next move will be a rapid traverse to the Initial level. If the G99 code is programmed within the cycle, there will be no movement.

Note 1.

F (Feed) = RPM x Pitch.

G Codes -

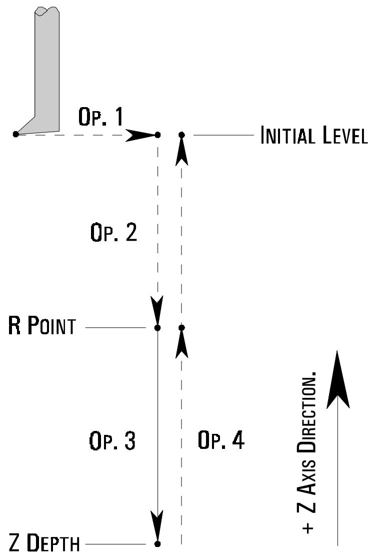
G85

(Boring).

A G85 (Boring) command is written in the following format:

(G9Ø) (G98)
or or G85 X.... Y.... Z.... R.... F.... ;
(G91) (G99)

please refer to page 52 for the variable definitions.



Sequence of moves:

Op 1) Rapid position to X, Y and Z (the initial level).

Op 2) Rapid traverse to R point level.

Op 3) Feed in to the Z depth.

Op 4) Feed back to R point level.

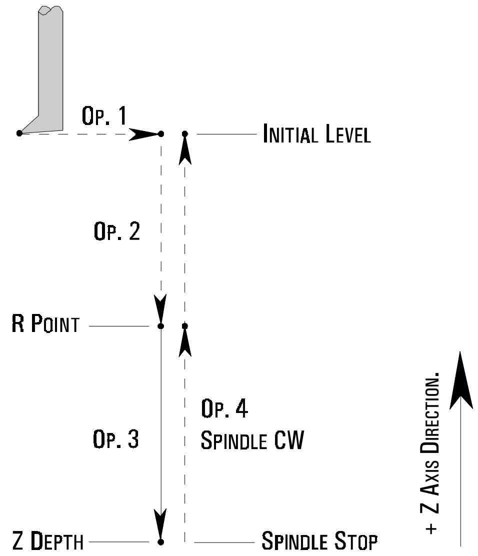
If the G98 code is programmed within the cycle, the next move will be a rapid traverse to the Initial level. If the G99 code is programmed within the cycle, there will be no movement.

G Codes - G86 (Boring).

A G86 (Boring) command is written in the following format:

(G9Ø) (G98)
or or G86 X.... Y.... Z.... R.... F.... ;
(G91) (G99)

please refer to page 52 for the variable definitions.



Sequence of moves:

Op 1) Rapid position to X, Y and Z (the initial level).

Op 2) Rapid traverse to R point level.

Op 3) Feed to Z depth and spindle stop.

Op 4) Rapid traverse to the initial level and spindle CW for G98, or rapid traverse to R point level and spindle CW for G99.

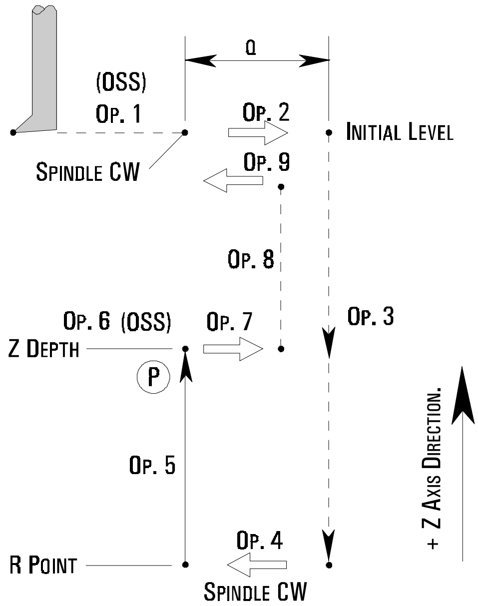
G Codes - G87

(Back Boring).

A G87 (Back Boring) command is written in the following format:

(G9Ø) (G98)
or or G87 X.... Y.... Z.... P..... Q.... R.... F.... ;
(G91) (G99)

please refer to page 52 for the variable definitions.



Sequence of moves:

Op 1) Rapid position to X, Y and Z (the initial level).

Op 2) Spindle stop and orientation. Move the value of Q.

Op 3) Rapid traverse to R point level.

Op 4) Spindle CW and move back the value of Q.

Op 5) Feed in to Z depth (positive direction) and dwell P.

Op 6) Spindle stop and orientate.

Op 7) Move the value of Q.

Op 8) Rapid traverse to R point level.

Op 9) Move back the value of Q and spindle CW.

Note 1.

THIS CYCLE CAN ONLY BE USED ON A MACHINE FITTED WITH A SPINDLE CAPABLE OF ORIENTATION. BECAUSE THE TOOL MOVES WITHIN THE HOLE AFTER SPINDLE STOP TO FACE THE OPPOSITE DIRECTION.

Note 2.

A G99 return to R point level is not possible within this cycle.

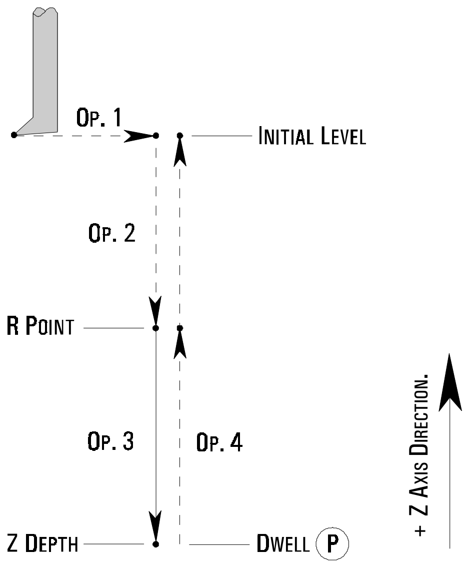
G Codes - G89

(Boring).

A G89 (Boring) command is written in the following format:

(G9Ø) (G98)
 or   or   G86 X.... Y.... Z.... P.... R.... F.... ;
(G91) (G99)

please refer to page 52 for the variable definitions.



Sequence of moves:

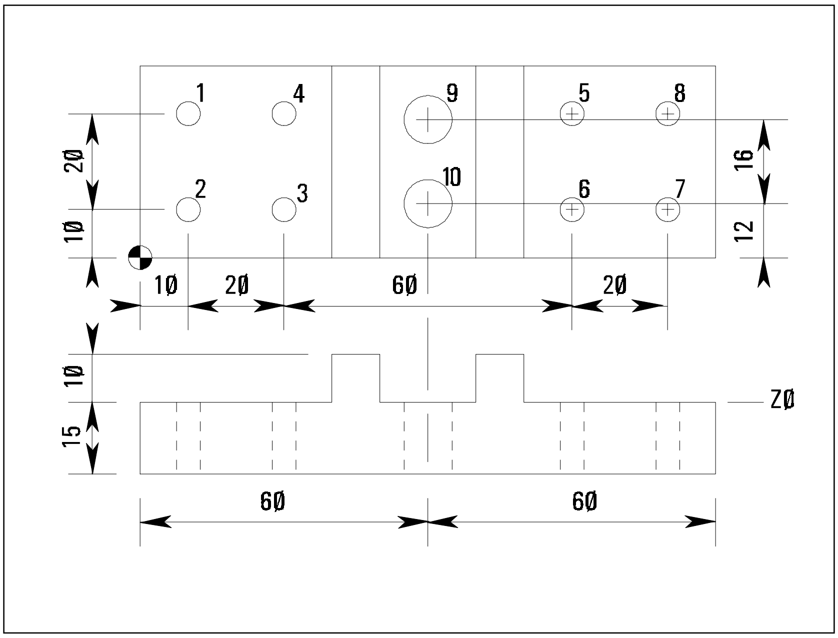
Op 1) Rapid position to X, Y and Z (the initial level).

Op 2) Rapid traverse to R point level.

Op 3) Feed to Z depth and dwell for value P.

Op 4) Feed out to R point and rapid traverse to initial level for G98, or feed out to R point for G99.

G Codes - Program Example Using Canned Cycles.



NØØ4Ø MØ6 TØ1 ;	Tool change.
NØØ5Ø G9Ø GØØ X1Ø Y3Ø Z12 S1ØØØ MØ3 ;	Tool position to initial level.
NØØ6Ø G99 G81 X1Ø Y3Ø Z-17 R2 F75 ;	Hole 1 retract R point.
NØØ7Ø Y1Ø ;	Hole 2 retract R point.
NØØ8Ø X3Ø ;	Hole 3 retract R point.
NØØ9Ø G98 Y3Ø ;	Hole 4 retract initial level.
NØ1ØØ G99 X9Ø ;	Hole 5 retract R point.
NØ11Ø Y1Ø ;	Hole 6 retract R point.
NØ12Ø X11Ø ;	Hole 7 retract R point.
NØ13Ø G98 Y3Ø ;	Hole 8 retract initial level.
NØ14Ø G91 G8Ø G28 XØ YØ ZØ MØ5 ;	Home position spindle stop.
NØ15Ø MØ6 TØ2 ;	Tool change.
NØ16Ø G9Ø GØØ X6Ø Y28 Z12 S75Ø MØ3 ;	Tool position initial level.
NØ17Ø G99 G83 X6Ø Y28 Z-17 Q6 R2 F6Ø ;	Hole 9 retract R point.
NØ18Ø G98 Y12 ;	Hole 1Ø retract initial level.
NØ19Ø G91 G8Ø G28 XØ YØ ZØ MØ5 ;	Home position spindle stop.
NØ2ØØ M3Ø ;	Program stop.

G Codes - G9Ø (Absolute Zero Command).

When G9Ø is active, all co-ordinates are relative to the workpiece datum (the zero position).

Note 1.

The G9Ø code is active when:

1) the machine power is first switched on.
2) the reset button on the CRT/MDI controller panel is pressed.
3) the Emergency Stop button is pressed.

G Codes - G91 (Incremental Command).

When G91 is active, all movement command values are distance moved (including the +/- sign) from last known programmed position.

G Codes - G94 (Feed per Minute).

When G94 is active, all feedrates stated within the program are defined in either millimetres per minute when operating in G21 Metric Mode, or inches per minute when operating in G2Ø Imperial Mode.

For example:

(G2Ø) F6 = 6 in/min.

(G21) F15Ø = 15Ø mm/min.

G Codes - G95

(Feed per Revolution).

When G95 is active, all feedrates stated within the program are defined in either millimetres per revolution when operating in G21 Metric Mode, or inches per revolution when operating in G2Ø Imperial Mode.

Note 1.

THE G95 CODE IS ONLY AVAILABLE WHEN THE MACHINE IS FITTED WITH A SPINDLE ENCODER. SEE YOUR MACHINE SPECIFICATION.

G Codes - G98

(Return to Initial Level).

A G98 code, when used within a canned cycle, will return the drill or boring bar back to the initial level after machining a hole.

G Codes - G99

(Return to R Point Level).

A G99 code, when used within a canned cycle, will return the drill or boring bar back to the R point level after machining a hole.

G CODES - G17Ø-G173

(CIRCULAR/ RECTANGULAR POCKET CANNED CYCLES).

The following canned cycles, when programmed correctly, will machine either a circular pocket to any diameter and depth, or a rectangular pocket to any side length and depth.

Both canned cycles require two blocks of information each, with each block having its own G code:

G17Ø, G171 — Circular pocket.

G172, G173 — Rectangular pocket.

NOTE 1.

Great care must be taken when using these canned cycles, since each canned cycle can be written three different ways.
This is achieved according to the values assigned following the addresses P, I and J in canned cycle G17Ø-171 and the values assigned following the addresses P, I and K in canned cycle G172-173. By adding these values, the cutter will move in a different path when machining.

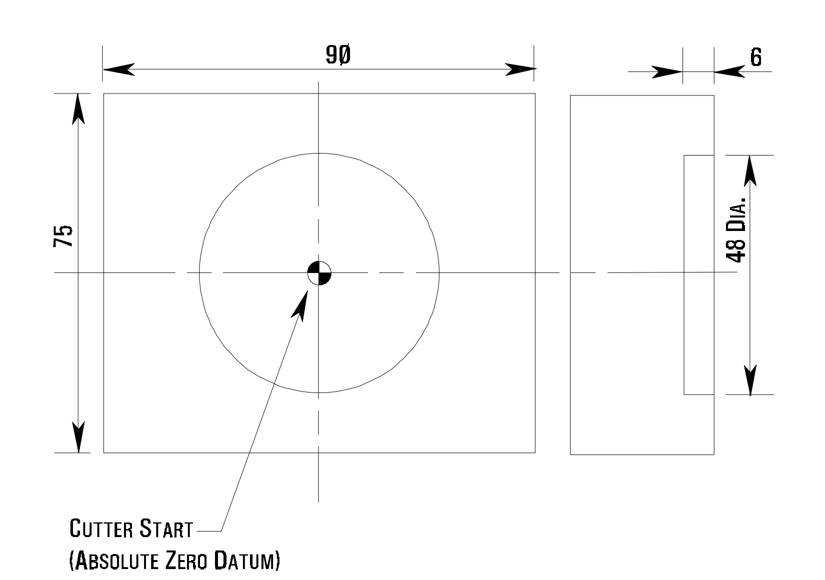
The following pages show six programs for these canned cycles....

Programs ØØØ2, ØØØ3 and ØØØ4 are for circular pockets.

Programs ØØØ5, ØØØ6 and ØØØ7 are for rectangular pockets.

G Codes - G17Ø G171

(Circular Pocket Canned Cycle Example A).



Note 1.
The absolute zero datum position is set in the offset file.

Program Number ØØØ2 - Roughing Program - Circular Pocket - G17Ø and G171.

[BILLET X75 Y9Ø Z3Ø ;	Billet size.
[EDGEMOVE X-37.5 Y-45 ;	Position of datum from the bottom LH corner of billet.
[TOOLDEF T1 D6 ZØ ;	Tool no. dia. and position.
O ØØØ2 ;	Program no.
NØØ4Ø G91 G28 XØ YØ ZØ ;	Metric, reference point.
NØØ5Ø MØ6 TØ1 ;	Tool no.
NØØ6Ø G9Ø GØØ XØ YØ Z1Ø S3ØØØ MØ3 ;	Absolute, rapid, tool 1Ømm above surface, spindle speed and start.
NØØ7Ø GØ1 ZØ F3ØØ ;	Tool to surface of job, feed set.
NØØ8Ø G17Ø RØ PØ Q3 XØ YØ Z-6 IØ JØ K-24 ;	Circular pocket canned cycle.
NØØ9Ø G171 P75 S3ØØØ R75 F25Ø B35ØØ J2ØØ ;	Circular pocket canned cycle.
NØ1ØØ GØØ Z25 MØ5 ;	Rapid, tool to 25mm above surface, spindle stop.
NØ11Ø G91 G28 XØ YØ ZØ ;	Incremental, return to reference point.
NØ12Ø M3Ø ;	Program reset.

G CODES -

G17Ø

G171

(CIRCULAR POCKET CANNED CYCLE EXAMPLE A).

Definitions for the terms used in the G17Ø and G171 circular pocket canned cycle from program number ØØØ2 follows:

NØØ8Ø G17Ø RØ PØ Q3 XØ YØ Z-6 IØ JØ K-24 ;

NØØ9Ø G171 P75 S3ØØØ R75 F25Ø B35ØØ J2ØØ ;

For G17Ø block,

R defines the position of the tool to start cycle ie. Ø (surface of job).

P defines when P is zero(Ø) the cycle is a roughing cycle.

Q defines the peck increment, in program number ØØØ2, 2 pecks each of 3mm.

X defines the pocket centre in X axis (Ø).

Y defines the pocket centre in Y axis (Ø).

Z defines the pocket base (-6mm) from job surface.

I defines the side finish allowance (Ø as this is a roughing cycle only).

J defines the base finish allowance (Ø as this is a roughing cycle only).

K defines the radius of pocket (-24) negative value - cut in CCW direction).

For G171 block,

P defines the cut width percentage.

S defines the roughing spindle speed (S3ØØØ).

R defines the roughing Feed in Z (75).

F defines the roughing feed XY (25Ø).

B defines the finishing spindle speed (35ØØ, not applicable as roughing only).

J defines the finishing feed (2ØØ, not applicable as roughing only).

When setting offsets the value R must be included, R being the radius of the cutter.

The direction of the cutter path is controlled by K, a negative (K-24) value for K means the cutter path is in a CCW direction and if the K value is positive (K+24) the cutterpath is in a CW direction. The Q value is always positive (Q+3).

When the tool has finished cutting the tool retracts 1mm in the Z axis, moves to the centre of the circular pocket at rapid traverse, retracts again in the Z axis .

Program number ØØØ2 is for a two cut roughing cycle.

G Codes -

G17Ø

G171

(Circular Pocket Canned Cycle Example B).

Program Number ØØØ3 - Roughing and Finishing Program - Circular Pocket - G17Ø and G171.

The difference between program numbers ØØØ2 and ØØØ3 is that program ØØØ3 leaves a finishing allowance on the diameter of the pocket and on the base.

```
[BILLET X75 Y9Ø Z3Ø ;
[EDGEMOVE X-37.5 Y-45 ;
[TOOLDEF T1 D6 ZØ ;
OØØØ3 ;
NØØ4Ø G91 G21 G28 XØ YØ ZØ ;
NØØ5Ø MØ6 TØ1 ;
NØØ6Ø G9Ø GØØ XØ YØ Z1Ø S3ØØØ MØ3 ;
NØØ7Ø GØ1 ZØ F3ØØ ;
NØØ8Ø G17Ø RØ PØ Q3 XØ YØ Z-6 IØ.5 JØ.1 K-24 ;
NØØ9Ø G171 P75 S3ØØØ R75 F25Ø B35ØØ J2ØØ ;
NØ1ØØ GØØ Z25 MØ5 ;
NØ11Ø G91 G28 XØ YØ ZØ ;
NØ12Ø M3Ø ;
```

G Codes -

G17Ø

G171

(Circular Pocket Canned Cycle Example B).

Definitions for the terms used in the G17Ø and G171 circular pocket canned cycle from program number ØØØ3 follows:

NØØ8Ø G17Ø RØ PØ Q3 XØ YØ Z-6 IØ.5 JØ.1 K-24 ;

NØØ9Ø G171 P75 S3ØØØ R75 F25Ø B35ØØ J2ØØ ;

For G17Ø block,

R defines the position of the tool to start cycle ie. Ø (surface of job).

P defines when P is zero(Ø) the cycle is a roughing cycle.

Q defines the peck increment, in program number ØØØ3, 2 pecks each of 3mm.

X defines the pocket centre in X axis (Ø).

Y defines the pocket centre in Y axis (Ø).

Z defines the pocket base (-6mm) from job surface.

I defines the side finish allowance (leaves a finishing allowance of Ø.5).

J defines the base finish allowance (leaves a finishing allowance of Ø.1).

K defines the radius of pocket (-24) negative value - cut in CCW direction).

For G171 block,

P defines the cut width percentage.

S defines the roughing spindle speed (S3ØØØ).

R defines the roughing Feed in Z (75).

F defines the roughing feed XY (25Ø).

B defines the finishing spindle speed (35ØØ).

J defines the finishing feed (2ØØ).

G Codes -

G17Ø

G171

(Circular Pocket Canned Cycle Example C).

Program Number ØØØ4 - One Step Finishing Cycle Program - Circular Pocket - G17Ø and G171.

The difference between program numbers ØØØ3 and ØØØ4 is that in program ØØØ4 the tool moves directly to the finish depth and executes a final finishing pass only.

```
[BILLET X75 Y9Ø Z3Ø ;
[EDGEMOVE X-37.5 Y-45 ;
[TOOLDEF T1 D6 ZØ ;
OØØØ4 ;
NØØ4Ø G91 G21 G28 XØ YØ ZØ ;
NØØ4Ø MØ6 TØ1 ;
NØØ4Ø G9Ø GØØ XØ YØ Z1Ø S3ØØØ MØ3 ;
NØØ4Ø GØ1 ZØ F3ØØ ;
NØØ4Ø G17Ø RØ P1 Q3 XØ YØ Z-6 IØ.5 JØ.1 K-24 ;
NØØ4Ø G171 P75 S3ØØØ R75 F25Ø B35ØØ J2ØØ ;
NØØ4Ø GØØ Z25 MØ5 ;
NØØ4Ø G91 G28 XØ YØ ZØ ;
NØØ4Ø M3Ø ;
```

G Codes - G17Ø G171 (Circular Pocket Canned Cycle Example C).

Definitions for the terms used in the G17Ø and G171 circular pocket canned cycle from program number ØØØ4 follows:

NØØ4Ø G17Ø RØ P1 Q3 XØ YØ Z-6 IØ.5 JØ.1 K-24 ;

NØØ4Ø G171 P75 S3ØØØ R75 F25Ø B35ØØ J2ØØ ;

For G17Ø block,

R defines the position of the tool to start cycle ie. Ø (surface of job).

P defines when P is 1 (P1) the cycle is a finishing cycle only.

Q is ignored when P=1.

X defines the pocket centre in X axis (Ø).

Y defines the pocket centre in Y axis (Ø).

Z defines the pocket base (-6mm) from job surface.

I is ignored when P=1.

J is ignored when P=1.

K defines the radius of pocket (-24) negative value - cut in CCW direction).

For G171 block,

P defines the cut width percentage.

S defines the roughing spindle speed (S3ØØØ). An S value must be entered but is ignored on the finishing cycle.

R defines the roughing Feed in Z (75). An R value must be entered but is ignored on the finishing cycle.

F defines the roughing feed XY (25Ø). An F value must be entered but is ignored on the finishing cycle.

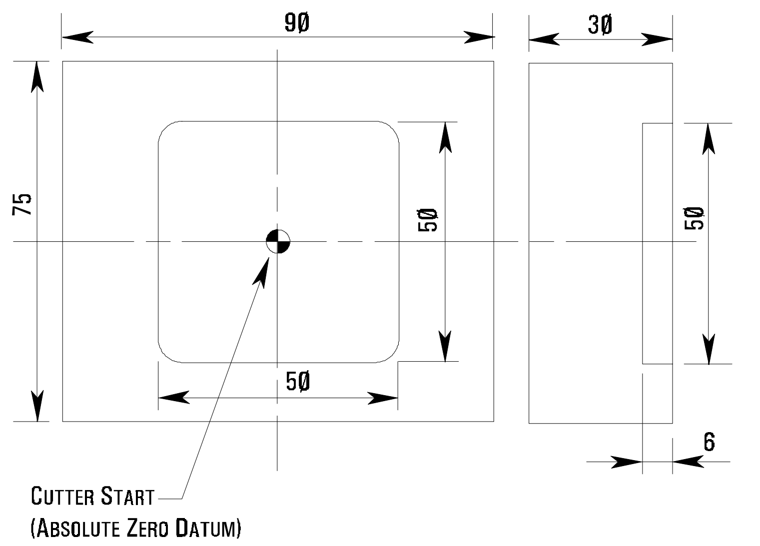
B defines the finishing spindle speed.

J defines the finishing feed.

Note 1.

Although characters are ignored by the program when run, in certain modes the characters must be programmed and given a value so that the sequential control of the program is maintained and errors avoided.

G Codes - G172 G173 (Rectangular Pocket Canned Cycle Example A).



Note 1.
The absolute zero datum position is set in the offset file.

Program Number ØØØ5 - Roughing Program - Rectangular Pocket - G172 and G173.

[BILLET X75 Y9Ø Z3Ø ;	Billet size.
[EDGEMOVE X-37.5 Y-45 ;	Position of datum from the bottom LH corner of billet.
[TOOLDEF T1 D6 ZØ ;	Tool no. dia. and position.
O ØØØ5 ;	Program no.
NØØ4Ø G91 G21 G28 XØ YØ ZØ ;	Metric, reference point.
NØØ5Ø MØ6 TØ1 ;	Tool no.
NØØ6Ø GØØ XØ YØ Z1Ø S3ØØØ MØ3 ;	Absolute, rapid, tool 1Ømm above surface, spindle speed and start.
NØØ7Ø GØ1 ZØ F3ØØ ;	Tool to surface of job, feed set.
NØØ8Ø G172 I-5Ø KØ PØ Q3 RØ X-25 Y-25 Z-6 ;	Rectangular pocket cycle.
NØØ9Ø G173 IØ KØ P75 T1 S3ØØ R75 F25Ø B35ØØ J2ØØ Z5 ;	Rectangular pocket cycle.
NØ1ØØ GØØ Z25 MØ5 ;	Rapid, tool to 25mm above surface, spindle stop.
NØ11Ø G91 G28 XØ YØ ZØ ;	Incremental, return to reference point.
NØ12Ø M3Ø ;	Program reset.

G Codes - G172 G173

(Rectangular Pocket Canned Cycle Example A).

Definitions for the terms used in the G172 and G173 rectangular pocket canned cycle from program number ØØØ5 follows:

NØØ8Ø G172 I-5Ø KØ PØ Q3 RØ X-25 Y-25 Z-6 ;

NØØ9Ø G173 IØ KØ P75 T1 S3ØØ R75 F25Ø B35ØØ J2ØØ Z5 ;

For G172 block,

I defines the pocket X length (-5Ø).
J defines the pocket Y length (-5Ø)
K defines the radius of corner roundness (not applicable to Denford software).
P defines that Ø = roughing cycle.
Q defines the pocket Z increment (peck increments in above cycle 2-3mm pecks).
R defines the Absolute Z 'R' point.
X defines the pocket corner X (Absolute position relative to the X datum position).
Y defines the pocket corner Y (Absolute position relative to the Y datum position).
Z defines the absolute Z base of pocket (-6, ie, a depth of 6mm).

For G173 block,

I defines the pocket side finish (Ø as this is a roughing cycle).
K defines the pocket base finish (Ø as this is a rough-ing cycle).
P defines the cut width percentage (75% of tool dia.).
T defines the pocket tool (tool 1).
S defines the spindle speed for roughing (3ØØØrpm).
R defines the roughing feed for Z (75).
F defines the roughing feed X and Y (25Ø).
B defines the finishing spindle speed (35ØØ rpm).
J defines the finishing feed (2ØØ).
Z defines the safety Z (5mm above 'R' point).

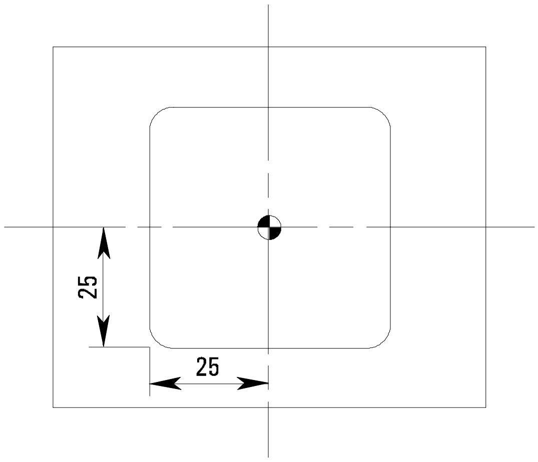
Program number ØØØ5 is for a two cut roughing cycle.

G Codes - G172 G173

(Rectangular Pocket Canned Cycle Example A).

For G172 block:

- The I and J are signed according to the direction of travel +/positive being CW and -/negative being CCW.
- K must be programmed as Ø (zero). For Denford software, the corner radius is the cutter radius.
- X and Y relate to an absolute zero on the workpiece, ie, the centre of the workpiece. In program number ØØØ5, the zero is set in the centre of the workpiece so the distance to the bottom LH corner is Y -25 and X -25, as shown below.



For G173 block:

- If I and K are set to zero (Ø), the program will perform the two roughing cuts only (as in the circular pocket canned cycle).
- The tool number (T1) must be programmed.

G Codes - G172 G173 (Rectangular Pocket Canned Cycle Example B).

Program Number ØØØ6 - Roughing and Finishing Program - Rectangular Pocket - G172 and G173.

```
[BILLET X75 Y9Ø Z3Ø ;
[EDGEMOVE X-37.5 Y-45 ;
[TOOLDEF T1 D6 ZØ ;
OØØØ6 ;
NØØ4Ø G91 G21 G28 XØ YØ ZØ ;
NØØ5Ø MØ6 TØ1 ;
NØØ6Ø G9Ø GØØ XØ YØ Z1Ø S3ØØØ MØ3 ;
NØØ7Ø GØ1 ZØ F3ØØ ;
NØØ8Ø G172 I-5Ø J-5Ø KØ PØ Q3 RØ X-25 Y-25 Z-6 ;
NØØ9Ø G173 IØ.5 KØ.1 P75 T1 S3ØØØ R75 F25Ø B35ØØ J2ØØ Z5 ;
NØ1ØØ GØØ Z25 MØ5 ;
NØ11Ø G91 G28 XØ YØ ZØ ;
NØ12Ø M3Ø ;
```

G Codes - G172 G173 (Rectangular Pocket Canned Cycle Example B).

Definitions for the terms used in the G172 and G173 rectangular pocket canned cycle from program number ØØØ6 follows:

```
NØØ8Ø G172 I-5Ø J-5Ø KØ PØ Q3 RØ X-25 Y-25 Z-6 ;
NØØ9Ø G173 IØ.5 KØ.1 P75 T1 S3ØØØ R75 F25Ø B35ØØ J2ØØ Z5 ;
```

For G172 block,

I defines the pocket X length (-5Ø).
J defines the pocket Y length (-5Ø)
K defines the radius of corner roundness (not applicable to Denford software).
P defines that Ø = roughing cycle.
Q defines the pocket Z increment (peck increments in above cycle 2-3mm pecks).
R defines the Absolute Z 'R' point.
X defines the pocket corner X (Absolute position relative to the X datum position).
Y defines the pocket corner Y (Absolute position relative to the Y datum position).
Z defines the absolute Z base of pocket (-6, ie, a depth of 6mm).

For G173 block,

I defines the pocket side finish (Ø.5 finishing allowance) on the finishing pass.
K defines the pocket base finish (Ø.1 finishing allowance) on the finishing pass.
P defines the cut width percentage (75% of tool dia.).
T defines the pocket tool (tool 1).
S defines the spindle speed for roughing (3ØØØrpm).
R defines the roughing feed for Z (75).
F defines the roughing feed X and Y (25Ø).
B defines the finishing spindle speed (35ØØ rpm).
J defines the finishing feed (2ØØ).
Z defines the safety Z (5mm above 'R' point).

When values are stated for the I and K elements, the program will perform a finishing pass after completion of the final roughing cut.

G Codes - G172 G173 (Rectangular Pocket Canned Cycle Example C).

Program Number ØØØ7 - One Step Finishing Cycle Program - Rectangular Pocket - G172 and G173.

```
[BILLET X75 Y90 Z30 ;
[EDGEMOVE X-37.5 Y-45 ;
[TOOLDEF T1 D6 Z0 ;
O0007 ;
NØØ4Ø G91 G21 G28 X0 Y0 Z0 ;
NØØ4Ø M06 T01 ;
NØØ4Ø G90 G00 X0 Y0 Z10 S3000 M03 ;
NØØ4Ø G01 Z0 F300 ;
NØØ4Ø G172 I-50 J-50 K0 P1 Q3 R0 X-25 Y-25 Z-6 ;
NØØ4Ø G173 I0.5 K0.1 P75 T1 S3000 R75 F250 B3500 J200 Z5 ;
NØØ4Ø G00 Z25 M05 ;
NØØ4Ø G91 G28 X0 Y0 Z0 ;
NØØ4Ø M30 ;
```

G Codes - G172 G173 (Rectangular Pocket Canned Cycle Example C).

Definitions for the terms used in the G172 and G173 rectangular pocket canned cycle from program number ØØØ7 follows:

NØØ4Ø G172 I-50 J-50 K0 P1 Q3 R0 X-25 Y-25 Z-6 ;

NØØ4Ø G173 I0.5 K0.1 P75 T1 S3000 R75 F250 B3500 J200 Z5 ;

For G172 block,

- I defines the pocket X length (-5Ø).
- J defines the pocket Y length (-5Ø)
- K defines the radius of corner roundness (not applicable to Denford software).
- P defines that 1 = finishing cycle only.
- Q defines the pocket Z increment (peck increments in above cycle 2-3mm pecks). This is ignored for a one step finishing cycle.
- R defines the Absolute Z 'R' point.
- X defines the pocket corner X (pocket length).
- Y defines the pocket corner Y (pocket width).
- Z defines the absolute Z base of pocket (-6, ie, a depth of 6mm).

For G173 block,

- I defines the pocket side finish allowance. This is ignored for a one step finishing cycle.
- K defines the pocket base finish allowance. This is ignored for a one step finishing cycle.
- P defines the cut width percentage (75% of tool dia.).
- T defines the pocket tool (tool 1).
- S defines the spindle speed for roughing. An S value must be entered but is ignored on the finishing cycle.
- R defines the roughing feed for Z. An R value must be entered but is ignored on the finishing cycle.
- F defines the roughing feed X and Y. An F value must be entered but is ignored on the finishing cycle.
- B defines the finishing spindle speed (35ØØ rpm).
- J defines the finishing feed (2ØØ).
- Z defines the safety Z (5mm above 'R' point).

Note 1.

Although characters are ignored by the program when run, in certain modes the characters must be programmed and given a value so that the sequential control of the program is maintained and errors avoided.

M Codes (Miscellaneous Functions) - Introduction.

Miscellaneous functions, called *M codes*, are used by the CNC to command on/off signals to the machine functions. ie, MØ3 - spindle forward (CW), MØ5 - spindle stop, etc.....

The functions allocated to lower M code numbers are constant in most CNC controls, although the higher M code number functions can vary from one make of controller to the next.

An M code is defined using the M address letter and a two digit number as follows,

M ØØ

Address. Two digit number.

List of M Codes Supported by Denford CNC Controls.

Note - Not all M codes apply to each machine.

M code.	Function.
MØØ*	Program Stop
MØ1*	Optional Stop
MØ2*	Program Reset
MØ3	Spindle Forward (clockwise)
MØ4	Spindle Reverse (counter clockwise)
MØ5*	Spindle Stop
MØ6	Automatic Tool Change
MØ8	Coolant On
MØ9*	Coolant Off
M1Ø	Vice/Work Clamp Open
M11	Vice/Work Clamp Close
M13	Spindle Forward and Coolant On
M14	Spindle Reverse and Coolant On
M19	Spindle Orientation
M2Ø	ATC Arm In
M21	ATC Arm Out
M22	ATC Arm Down
M23	ATC Arm Up
M24	ATC Drawbar Unclamp
M25	ATC Drawbar Clamp
M27	Reset Carousel to Pocket One
M3Ø*	Program Reset and Rewind
M32	Carousel CW
M33	Carousel CCW
M38	Door Open
M39	Door Close
M62	Auxiliary Output 1 On
M63	Auxiliary Output 2 On

List of M Codes Supported by Denford CNC Controls.

continued....

M code.	Function.
M64	Auxiliary Output 1 Off
M65	Auxiliary Output 2 Off
M66*	Wait for Auxiliary Output 1 On
M67*	Wait for Auxiliary Output 2 On
M7Ø	Mirror in X On
M71	Mirror in Y On
M76	Wait for Auxiliary Output 1 Off
M77	Wait for Auxiliary Output 2 Off
M8Ø	Mirror in X Off
M81	Mirror in Y Off
M98	Sub Program Call
M99	Sub Program End and Return

Code listing full and correct at the time of printing.

Notes for M Codes Listing shown left.

Note 1.

M codes marked with an * are executed at the end of a block, ie, after axis movement.

Note 2.

Only one M code can be programmed within each block. If more than one M code is programmed, the machine controller will only perform the last stated M code.

Note 3.

The M codes listed between M19 to M27 inclusive and M32 to M33 inclusive are used for maintenance only. They are entered through the MDI panel on the machine tool and do not appear in a CNC part program.

M Codes - MØØ (Program Stop).

MØØ* - Program Stop.

When the machine controller reads the code MØØ within a block, it halts the program. The [CYCLE START] key must be pressed to allow the program to continue.

M Codes - MØ1 (Optional Stop).

MØ1* - Optional Stop.

The MØ1 code performs the same function as the MØØ code, except the machine controller only recognises the signal to halt the program if the optional [STOP] input key is activated.

M Codes - MØ2 (Program Reset).

MØ2* - Program Reset.

This code indicates the end of a program and performs a general reset function on the machine controller, ie, the CNC reverts to its initial state. The code also acts as an MØ5.

M CODES - MØ3 (SPINDLE FORWARD).

MØ3 - Spindle Forward (Clockwise).

The clockwise direction of the spindle is determined by viewing from the back of the machine headstock, along the Z axis towards the tailstock.

The spindle start command is activated at the beginning of the block in which it is programmed, ie, before any axis movement occurs.

M CODES - MØ4 (SPINDLE REVERSE).

MØ4 - Spindle Reverse (Counter Clockwise).

An MØ4 code acts in the same way as an MØ3 code, only the spindle rotates in the opposite direction.

M CODES - MØ5 (SPINDLE STOP).

MØ5* - Spindle Stop.

The MØ5 code, to stop the spindle rotating, is activated at the end of the block in which it is programmed, ie , after any axis movement.

M Codes - MØ6 (Automatic Tool Change).

MØ6 - Automatic Tool Change.

This code activates the machine turret and is followed by the code T_ _ _ _, instructing it to move to the stated tool number.

For example :

MØ6 TØ3Ø3 ;

This command is read change automatically from the current tool number to tool number 3.

M Codes - MØ8 (Coolant On).

MØ8 - Coolant On.

This code switches the coolant pump on.

M Codes - MØ9 (Coolant Off).

MØ9* - Coolant Off.

This code switches the coolant pump off.

M Codes - M1Ø (Vice Open).

M1Ø - Vice/Work Clamp Open.

This code will open the jaws of a power vice.

M Codes - M11 (Vice Close).

M11 - Vice/Work Clamp Close.

This code will close the jaws of a power vice.

M Codes - M13 (Spindle Forward and Coolant On).

M13 - Spindle Forward and Coolant On.

This code combines the functions of MØ3 and MØ8 together. The MØ5 code will stop both the spindle and coolant.

M Codes - M14 (Spindle Reverse and Coolant On).

M14 - Spindle Reverse and Coolant On.

This code performs the same function as M13 but the spindle rotates in the opposite direction.

M Codes - M19 (Spindle Orientation).

M19 - Spindle Orientation.

This code will orientate the machine spindle - see your machine specification.

M Codes - M2Ø (ATC Arm In).

M2Ø - ATC Arm In.

This code moves the Automatic Tool Changer arm in (beneath the spindle).

M Codes - M21 (ATC Arm Out).

M21 - ATC Arm Out.

This code moves the Automatic Tool Changer arm out (away from the spindle).

M Codes - M22 (ATC Arm Down).

M22 - ATC Arm Down.

This code moves the Automatic Tool Changer in a downwards direction.

M Codes - M23 (ATC Arm Up).

M23 - ATC Arm Up.

This code moves the Automatic Tool Changer in an upwards direction.

M Codes - M24 (ATC Drawbar Unclamp).

M24 - ATC Drawbar Unclamp.

This code unclamps the tool collet currently held in the spindle.

M Codes - M25 (ATC Drawbar Clamp).

M24 - ATC Drawbar Clamp.

This code clamps a tool collet in the spindle.

M Codes - M27 (Reset Carousel to Pocket One).

M27 - Reset to Carousel Pocket One.

This code is used to reset the carousel back to position one.

M Codes - M3Ø (Program Reset and Rewind).

M3Ø* - Program Stop and Rewind.

This code stops the program running, ie, it signals the end of the program. Control is then reset back to the beginning of this program.

If the M3Ø code is followed by a block number, the program will be reset back to the stated block number.

For example :

M3Ø PØ14Ø ;

This command is read stop the program running and reset it back to block number 18Ø.

The M3Ø code also acts as an MØ5 and MØ9.

M Codes - M32 (Carousel CW).

M32 - Carousel CW.

This code is used to rotate the carousel in a clockwise direction (viewed in plan view).

M Codes - M33 (Carousel CCW).

M33 - Carousel CCW.

This code is used to rotate the carousel in a counter clockwise direction (viewed in plan view).

M Codes - M38 (Door Open).

M38 - Door Open.

This code opens the machine door.

M Codes - M39 (Door Close).

M39 - Door Close.

This code closes the machine door.

M Codes - M62 M63 M64 M65 M66 M67 M76 M77 (Auxiliary Output Functions).

M62 - Auxiliary Output 1 On.

M63 - Auxiliary Output 2 On.

M64 - Auxiliary Output 1 Off.

M65 - Auxiliary Output 2 Off.

M66* - Wait for Auxiliary Output 1 On.

M67* - Wait for Auxiliary Output 2 On.

M76 - Wait for Auxiliary Output 1 Off.

M77 - Wait for Auxiliary Output 2 Off.

These codes allow a signal to be sent from the machine controller to a different device, such as a robot, then wait for a return signal instructing that the device has completed its function.

M Codes - M7Ø (Mirror in X On).

M7Ø - Mirror in X On.

This code changes the sign of X within a program around the datum.

For example:

GØ1 X25.5 ;
M7Ø
GØ1 X25.5 ; (Tool would move to X-25.5).

M Codes - M71 (Mirror in Y On).

M71 - Mirror in Y On.

This code performs the same function as G7Ø, but affects the Y axis.

M Codes - M8Ø (Mirror in X Off).

M8Ø - Mirror in X Off.

This code cancels the mirror image in the X axis.

M Codes - M81 (Mirror in Y Off).

M81 - Mirror in Y Off.

This code cancels the mirror image in the Y axis.

M Codes - M98 (Sub Program Call).

M98 - Sub Program Call.

This code will cause the machine controller to jump across from the main program to read a different program in its memeory (called a sub program).

M Codes - M99 (Sub Program End and Return).

M99 - Sub Program End and Return.

On the last line of a sub program, the code M99 is entered. This reverts control back to the main program.

If an M99 code is programmed at the end of a main program, a continuous loop will be established.

If an M99 code is followed by a block number, P_ _ _ _ , control will return to the program line with the same number as stated in P_ _ _ _.