

T

R

O

F

S

E

D



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

CONTENTS.

TITLE	PAGE No.
Introduction	6
What is CNC?	7
BASIC PROGRAMMING.	
Composition of a Part Program	9
Main Program Structure	10
Sub Program Structure	12
Sub Program Commands - General Notes	13
Sub Program Repeat Command	14
Billet Definition	15
Program Numbering and Saving	16
Program Block Numbering	17
Block Configuration	18
G Codes (Preparatory Functions)	18
Tool Movement	19
Feed Function	21
M Codes (Miscellaneous Functions)	21
Spindle Speed Function (Cutting Speed)	22
Tool Function	23
Tool Compensation (Tool Offset)	24
Absolute and Incremental Co-ordinates	25
Optional Block Skip	26
Tutorials and Comments	26

CONTENTS.

TITLE	PAGE No.
G CODES - PREPARATORY FUNCTIONS.	
G Codes - Introduction	27
List of G Codes supported by Denford CNC Controls	29
G00 (Rapid Positioning/Traverse)	30
G01 (Linear Interpolation)	32
G02 & G03 (Circular Interpolation)	34
G04 (Dwell)	40
G20 & G21 (Imperial /Metric Data Input)	41
G28 (Reference Point Return)	42
G40, G41 & G42 (Cutter Compensation)	45
G73-G89 (Canned Cycles)	49
G73 (High Speed Peck Drilling)	54
G74 (Counter Tapping)	55
G76 (Fine Boring)	56
G80 (Canned Cycle, Cancel)	57
G81 (Drilling - Spot Boring)	58
G82 (Drilling - Counter Boring)	59
G83 (Deep Hole Peck Drilling)	60
G84 (Tapping)	61
G85 (Boring)	62
G86 (Boring)	63
G87 (Back Boring)	64
G89 (Boring)	65
G Codes - Program Example Using Canned Cycles ...	66
G90 (Absolute Zero Command)	67
G91 (Incremental Command)	67
G94 (Feed per Minute)	67
G95 (Feed per Revolution)	68
G98 (Return to Initial Level)	68
G99 (Return to R Point Level)	68
G170-G173 (Circular/Rectangular Pocket)	69
G170 & G171 (Circular Pocket Example A)	70
G170 & G171 (Circular Pocket Example B)	72
G170 & G171 (Circular Pocket Example C)	74
G172 & G173 (Rectangular Pocket Example A) ...	76
G172 & G173 (Rectangular Pocket Example B) ...	79
G172 & G173 (Rectangular Pocket Example C) ...	81

CONTENTS.

TITLE	PAGE No.
M CODES - MISCELLANEOUS FUNCTIONS.	
M Codes - Introduction	83
List of M Codes Supported by Denford CNC Controls	84
M00 (Program Stop)	86
M01 (Optional Stop)	86
M02 (Program Reset)	86
M03 (Spindle Forward)	87
M04 (Spindle Reverse)	87
M05 (Spindle Stop)	87
M06 (Automatic Tool Change)	88
M08 (Coolant On)	88
M09 (Coolant Off)	88
M10 (Vice Open)	89
M11 (Vice Close)	89
M13 (Spindle Forward and Coolant On)	89
M14 (Spindle Reverse and Coolant On)	90
M20 (ATC Arm In)	90
M19 (Spindle Orientation)	90
M21 (ATC Arm Out)	91
M22 (ATC Arm Down)	91
M23 (ATC Arm Up)	91
M24 (ATC Drawbar Unclamp)	92
M27 (Reset Carousel to Pocket One)	92
M25 (ATC Drawbar Clamp)	92
M30 (Program Reset and Rewind)	93
M32 (Carousel CW)	93
M33 (Carousel CCW)	93
M38 (Door Open)	94
M39 (Door Close)	94
M62, M63, M64, M65, M66, M67, M76 & M77 (Auxiliary Output Functions)	95
M70 (Mirror in X On)	96
M71 (Mirror in Y On)	96
M80 (Mirror in X Off)	96
M81 (Mirror in Y Off)	97
M98 (Sub Program Call)	97
M99 (Sub Program End and Return)	97

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.

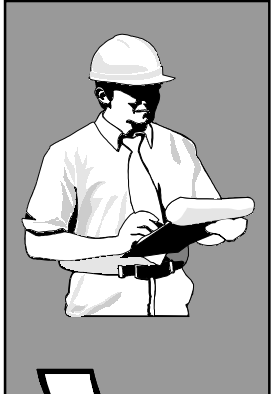
3) The machine controller processes the part program and sends signals to the machine components directing the machine through the required sequence of operations necessary to manufacture the component.

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

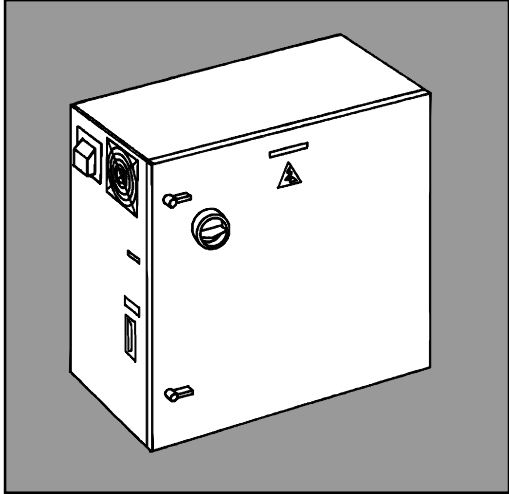
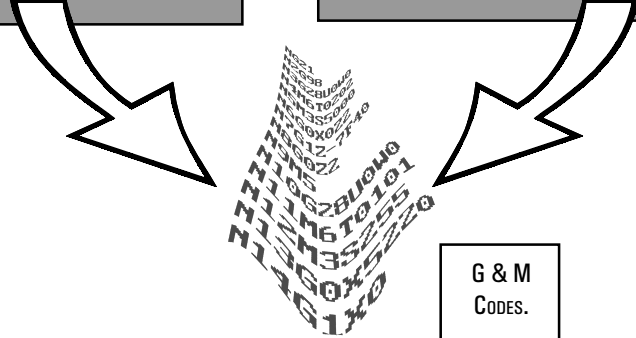
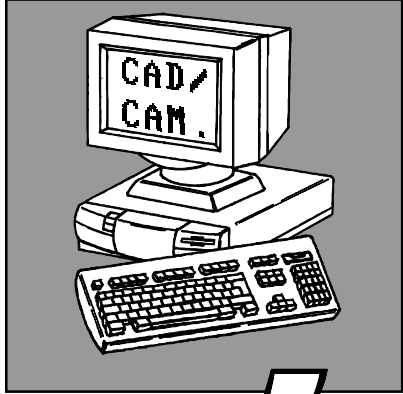
Combining this with the use of a part program enhances the ability of the machine to perform repeat tasks with high degrees of accuracy.

DIAGRAM - EXAMPLE CNC MANUFACTURING PROCESS.

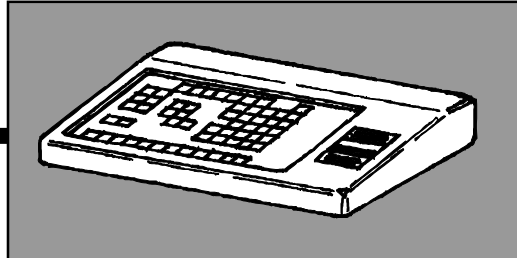
HUMAN PROGRAMMING (MDI - MANUAL DATA INPUT).



COMPUTER PROGRAMMING (CAD/CAM).

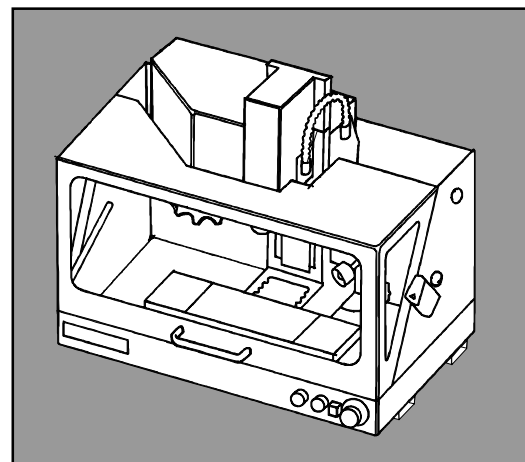
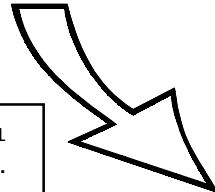


MACHINE ELECTRICAL CONTROL UNIT.



MACHINE CONTROL KEYPAD.

CONTROL SIGNALS.



DENFORD CNC MILLING MACHINE.

COMPOSITION OF A PART PROGRAM.

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 ;
```

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.

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.

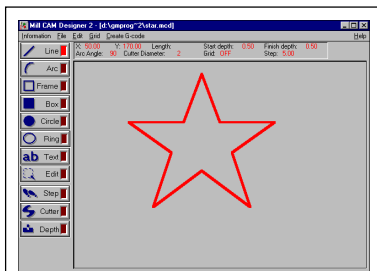
Each block, or program line, contains addresses which appear in this order :

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.

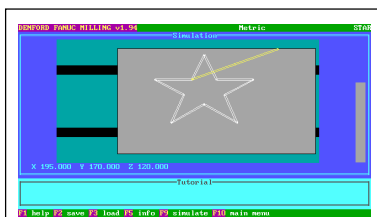
MAIN PROGRAM STRUCTURE.

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.



Machine Simulation of G and M code program.

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 ;
N220 M30 ;

Program end.

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.

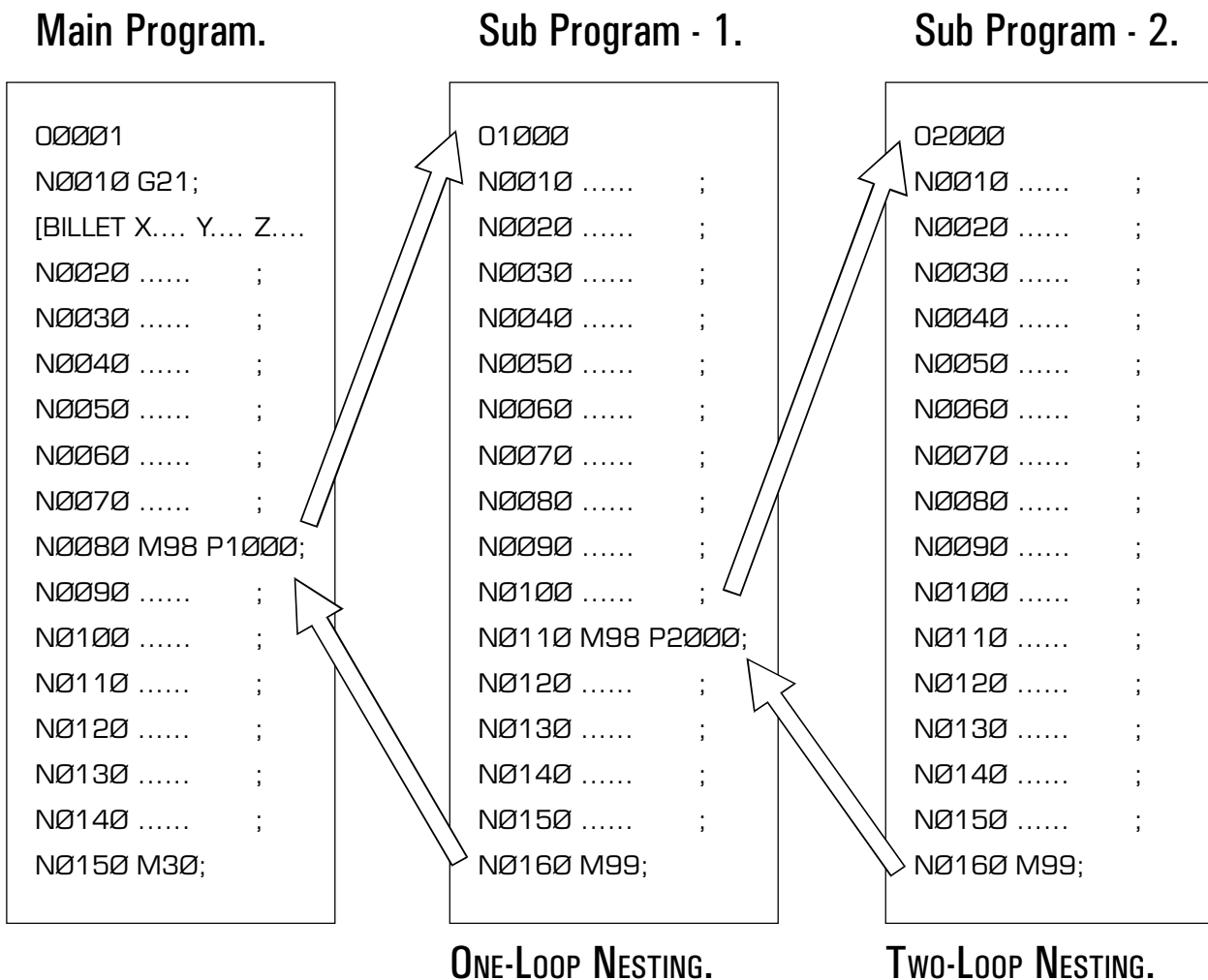
A sub program is entered into the machine controller memory in Edit Mode, in the same manner as the main program.

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.



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 P0000 (the number of the sub program required).

For example,

```
M98 P2000
```

This command is read call program number 2000.

NOTE 4.

A sub program call command (M98 P0000) can be specified along with a move command in the same block.

For example,

```
G01 X42.5 M98 P1000;
```

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 P0000 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,

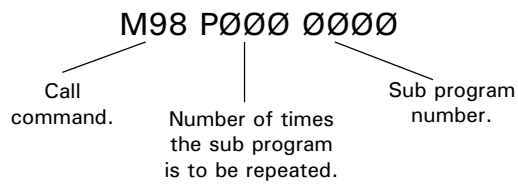
```
M99 P0160
```

This command is read return to the main program at block number N0160.

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 :



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

For example,

M98 P100001

This command is read call the sub program number 0001 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 G20 (Imperial data input).

For example,

```
N0010 G21;
```

```
[BILLET X100 Y150 Z20;
```

These two commands are read.

- program line number 10 states that all units are to be measured in Metric,
- the billet is a rectangular piece of material, measuring 100mm x 200mm, with a thickness of 20mm.

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.

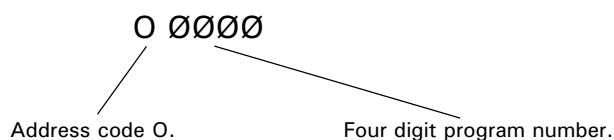
Sub program numbers must be saved between the ranges 0001 to 9999, ie, using a four digit number. It is recommended that all programs saved have filenames also between these numbers.

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 :



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 (0001 - 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 10. This allows the program to be edited at a later date, ie, new blocks can be inserted or deleted as required.

For example :

N 0010

N 0020

N 0030

N 0040

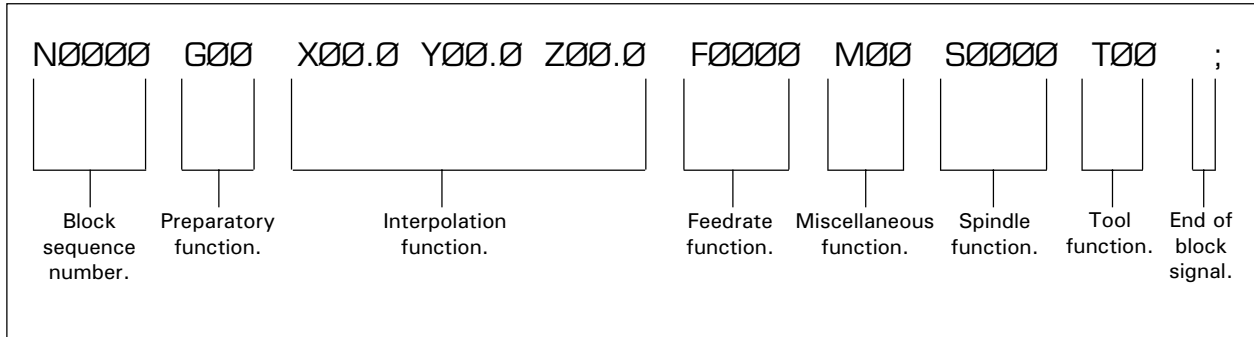
etc....

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 :



NOTE 1.

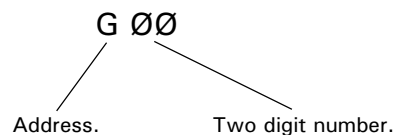
Each block may not necessarily contain all these items.

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,



TOOL MOVEMENT.

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

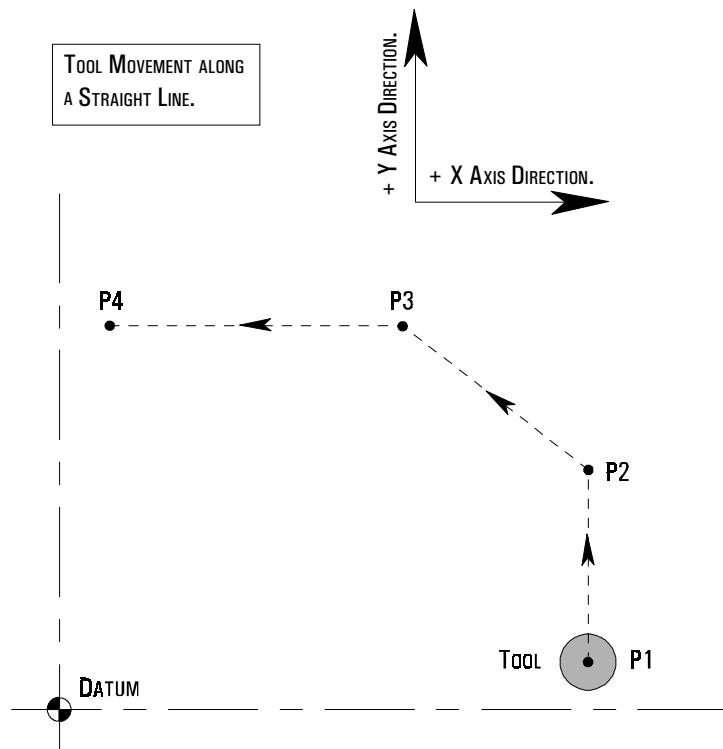
A) TOOL MOVEMENT ALONG A STRAIGHT LINE.

Program command format:

```
G01 Y _____ ; (P1 - P2)
```

```
X _____ Y _____ ; (P2 - P3)
```

```
X _____ ; (P3 - P4)
```

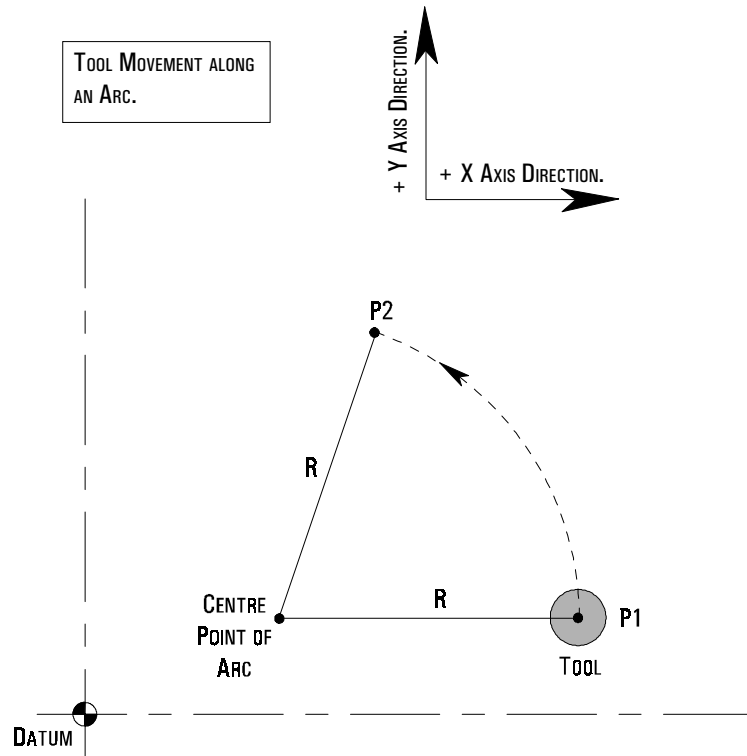


TOOL MOVEMENT.

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 letter followed by a numerical value.

Using the G20 code, the feedrate is defined in Inches per minute.

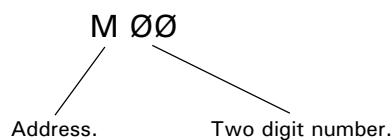
Using the G21 code, the feedrate is defined in Millimetres per minute.

M CODES (MISCELLANEOUS FUNCTIONS).

Miscellaneous functions, called *M codes*, are used by the CNC to command on/off signals to the machine functions. ie, M03 - spindle forward (CW), M05 - 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,



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.

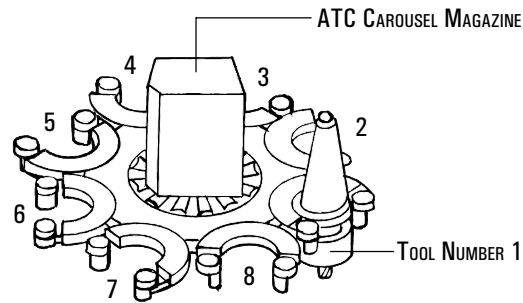
NOTE 1.

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 M06 code would precede the number of the "new" tool required.

For example,

```
M06 T01
```

This command is read perform a tool change to tool number 01.

NOTE 1.

The M06 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 M06 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.

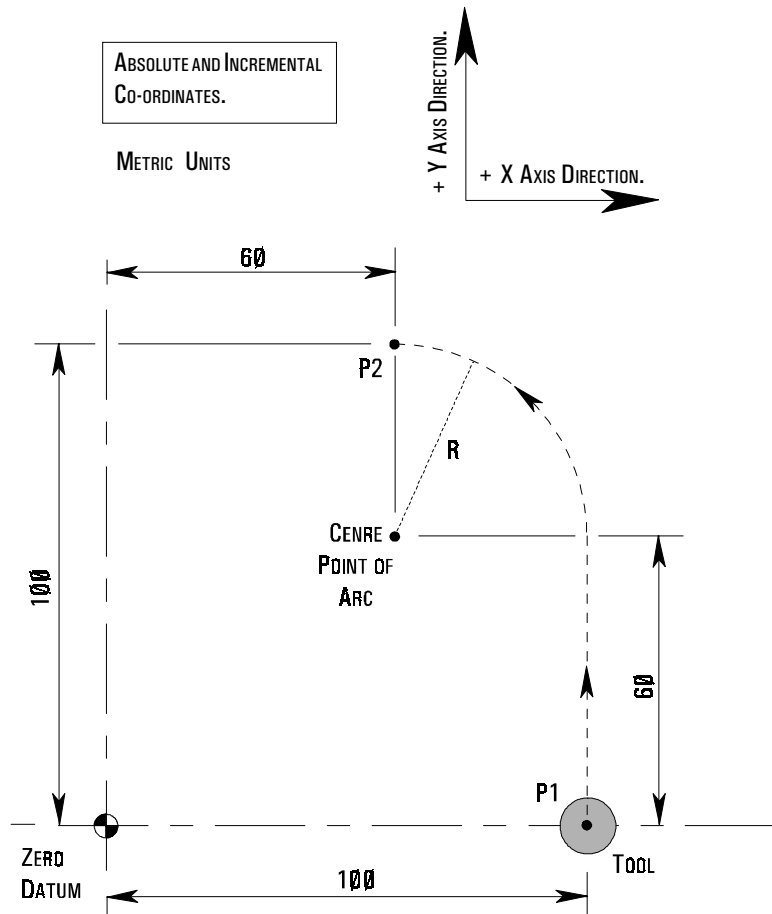
NOTE 1.

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 be 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

G01 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).

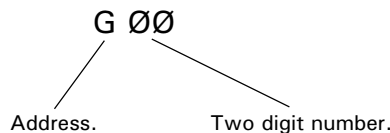
Comments information only appear in the text of the program itself (ie, the comment "Tool 5 is 8mm slot cutting tool" could be inserted in the program for use as reference only).

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,



MODAL AND NON-MODAL G CODES.

Additionally, G codes are split into two categories -

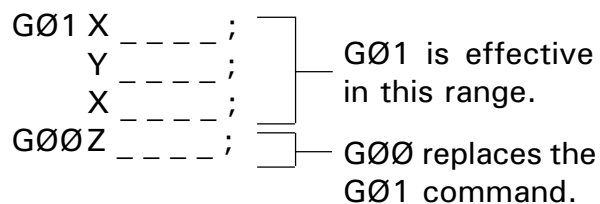
1) *Modal* (retained) G codes.

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,

G01 and G00 are modal G codes from group 1:



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 G01 (linear cutting command) then G01 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 G00, G02 or G03).

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 0 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.
G00	1	Positioning (Rapid Traverse)
G01	1	Linear Interpolation (Cutting Feed)
G02	1	Circular Interpolation CW
G03	1	Circular Interpolation CCW
G04	0	Dwell, Exact Stop
G20	6	Imperial Data Input (Inches)
G21	6	Metric Data Input (Millimetres)
G28	0	Reference Point Return
G40	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	<i>Fine Boring Cycle(not recommended on Denford Machines)</i>
G80*	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	<i>Back Boring Cycle(not recommended on Denford Machines)</i>
G89	9	Boring Cycle
G90*	3	Absolute Zero Command
G91	3	Incremental Command
G94*	5	Feed per Minute
G95	5	Feed per Revolution
G98*	10	Return to Initial Level in Canned Cycle
G99	10	Return to R Point Level in Canned Cycle
G170	0	Circular Pocket Canned Cycle
G171	0	Circular Pocket Canned Cycle
G172	0	Rectangular Pocket Canned Cycle
G173	0	Rectangular Pocket Canned Cycle

Code listing full and correct at the time of printing.

G CODES -

G00

(RAPID POSITIONING/ TRAVERSE).

The G00 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 G00 command is written in the following format:

G00 X _ _ _ _ Y _ _ _ _ Z _ _ _ _ ;

/

Rapid Positioning/
Traverse code.

X, Y and Z
co-ordinate values.

The axis co-ordinate moves following a G00 command can be programmed as either:

- (i) absolute values (relative to a set datum point) following setting of the G90 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 100% to 0%, but only in increments of 10%, by using the feed override controls (see specific machine operating manual).

NOTE 2.

The G00 code freezes the tool radius compensation, codes G 41 and G42. If G41 or G42 are active when a G00 command is programmed, the tool radius compensation will not function again until a G01, G02 or G03 command is programmed.

NOTE 3.

The G00 code is modal and is therefore incompatible with G01, G02 and G03 codes in the same block.

NOTE 4.

The G00 code can be written into a program in two ways:
G00 or G0.

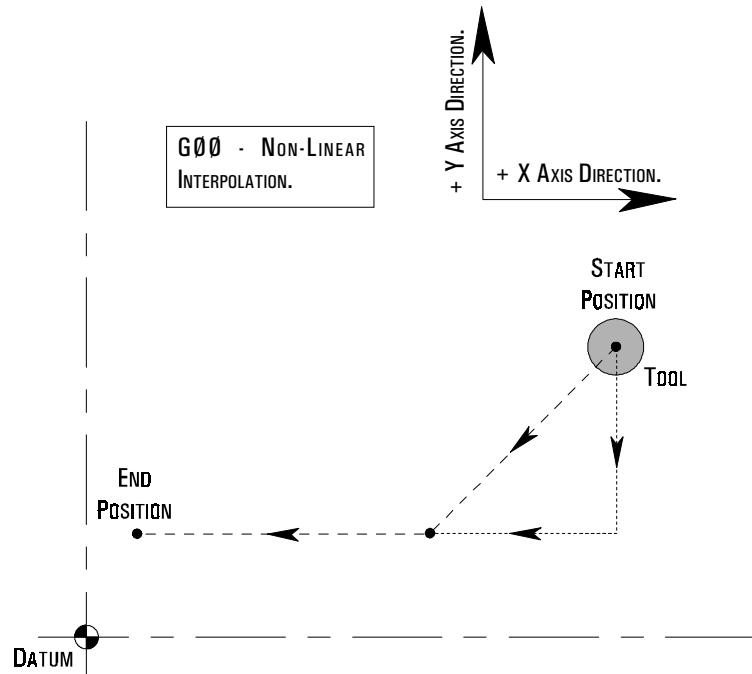
G CODES -

G00

(RAPID
POSITIONING/
TRAVERSE).

NOTE 5.

On instruction to perform the G00 command, the three slides (the X, Y and Z axes) move completely independent of each other at a maximum feedrate, along a non-vector (sometimes called a non-linear) type path.



In the above example, the G00 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 -

G01

(LINEAR INTERPOLATION).

The G01 code executes a cutting movement following a straight line, at a set feedrate.

A G01 command is written in the following format:

G01 X _ _ _ _ Y _ _ _ _ Z _ _ _ _ ;

Linear Interpolation code.

X, Y and Z co-ordinate values.

The feedrate value programmed into the G01 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 G01 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 G01 command.

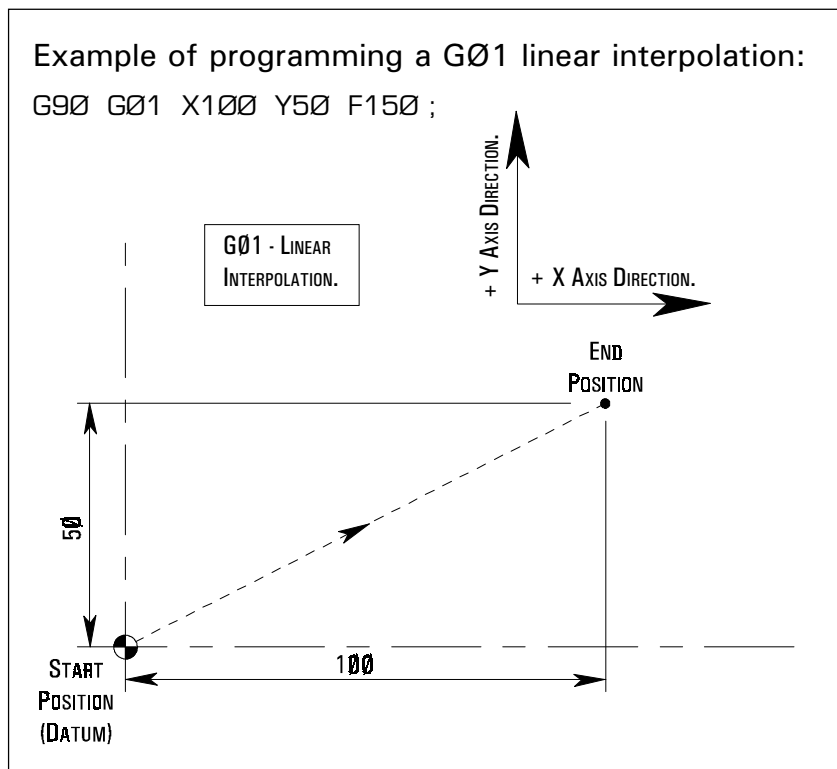
The axis co-ordinate moves following a G01 command can be programmed as either:

- (i) absolute values (relative to a set datum point) following setting of the G90 code, or,
- (ii) incremental values (relative to the last stated co-ordinate in the program) following setting of the G91 code.

G CODES -

G01

(LINEAR INTERPOLATION).



NOTE 1.

In the example shown above, G90, G01 and the _____ 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 0% to 150% by using the feed override controls (see specific machine operating manual). 100% is the programmed feedrate.

NOTE 3.

When there is no feedrate programmed within the part program, the CNC will set a feedrate of 10 millimetres per minute in G21 Metric Data Mode, or 0.4 inches per minute in G20 Imperial Data Mode.

NOTE 4.

The G01 code is modal and is therefore incompatible with G00, G02 and G03 codes in the same block.

NOTE 5.

The G01 code can be written into a program in two ways:
G01 or G1.

G CODES -

GØ2

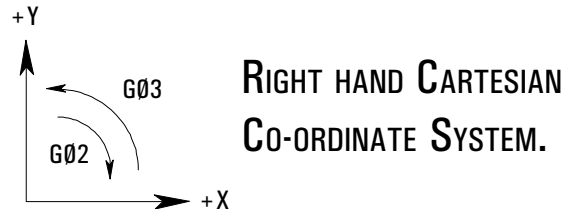
GØ3

(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.



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 -

G02

G03

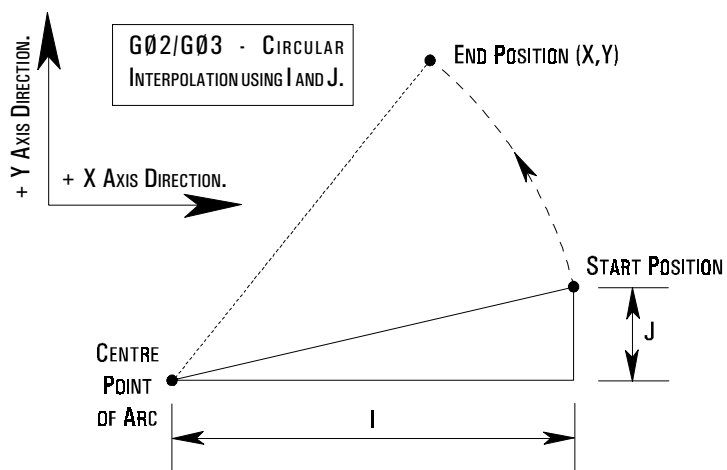
(CIRCULAR INTERPOLATION).

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:

```
G90 GØ2 X _____ Y _____ R ____ F ____ ;  
G90 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:

```
G90 GØ3 X _____ Y _____ R ____ F ____ ;  
G90 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.

G90 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.

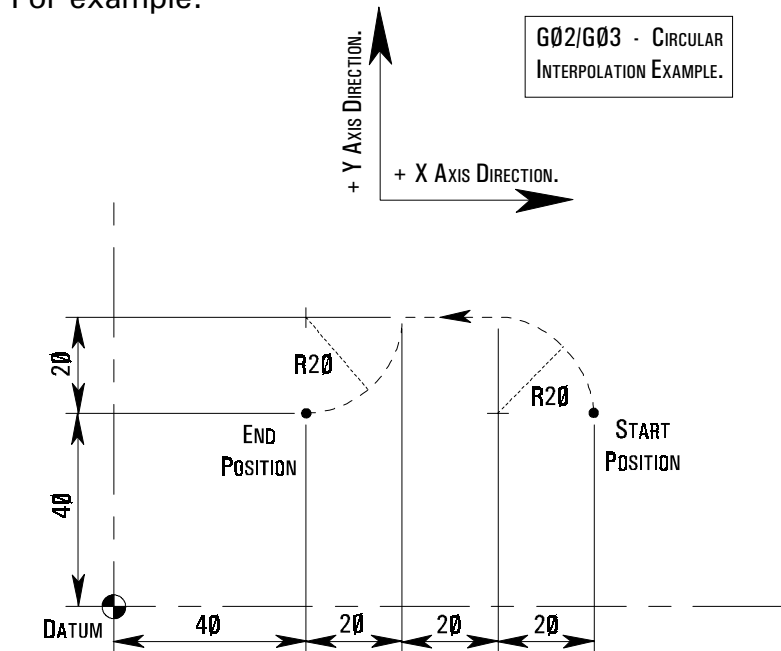
G CODES -

G02

G03

(CIRCULAR INTERPOLATION).

For example:



The above tool path can be programmed as follows (In absolute mode, G90):

```
G01 X100 Y40 F125 ;  
G03 X80 Y60 I-20 ;  
G01 X60 ;  
G02 X40 Y40 I-20 ;
```

or,

```
G01 X100 Y40 F125 ;  
G03 X80 Y60 R20 ;  
G01 X60 ;  
G02 X40 Y40 R20 ;
```

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

```
G03 X-20 Y20 I-20 ;  
G01 X-20 ;  
G02 X-20 Y-20 I-20 ;
```

or,

```
G03 X-20 Y20 R20 ;  
G01 X-20 ;  
G02 X-20 Y-20 R20 ;
```

G CODES -

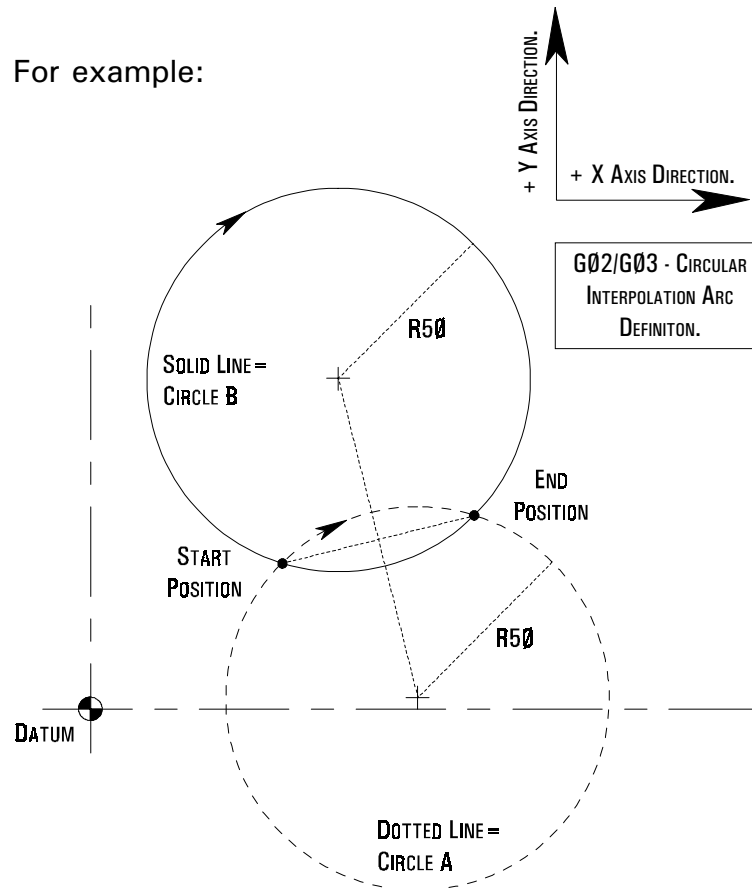
G02

G03

(CIRCULAR INTERPOLATION).

When using the address R _____, two types of arcs can be considered. One arc is less than 180 degrees, whilst the other arc is greater than 180 degrees. When an arc exceeding 180 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 180 degrees (circle A) can be programmed as follows (In absolute mode, G90):

```
G02 X80 Y40 R50 F125 ;
```

The above tool path for an arc greater than 180 degrees (circle B) can be programmed as follows (In absolute mode, G90):

```
G02 X80 Y40 R-50 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 360 degrees (ie, a complete circle) is assumed. When R is used, an arc of 0 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 1/1000 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 P2500 ;

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 -

G20

G21

(IMPERIAL

/METRIC

DATA INPUT).

The machine controller can be programmed in either Imperial (inch) unit input (G20) or Metric (millimetre) unit input (G21). The standard format for a CNC part program is to write the G20 or G21 code in the first block of the program.

G code.	Type.	Units.	Lowest input value.
G20	Imperial	Inch	0.0001 inch
G21	Metric	Millimetre	0.001 mm

The unit systems of the following items are changed depending on whether G20 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 G20 or G21 in the machine controller is dependant on the option that is saved to the disc.

NOTE 2.

A G20 code must not be changed for a G21 code (or vice versa) during the program.

NOTE 3.

When switching between G20 and G21, the offsets must be set according to the units of measurement being used.

NOTE 4.

G 20 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 :

```
G90 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.

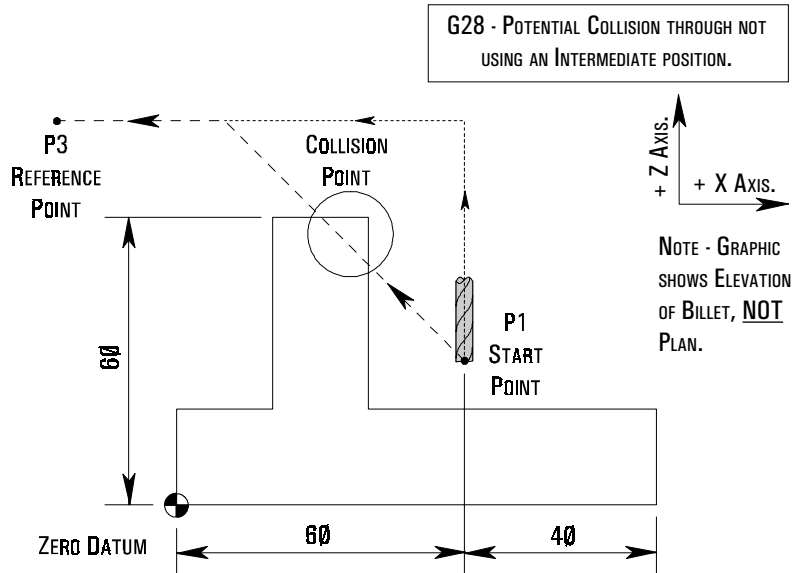
The move to any intermediate point and the reference point are performed at a rapid traverse rate, using a non-vector (non-linear) type path, ie, the tool may appear to "change" direction due to the non-vector type positioning being used.

G CODES - G28

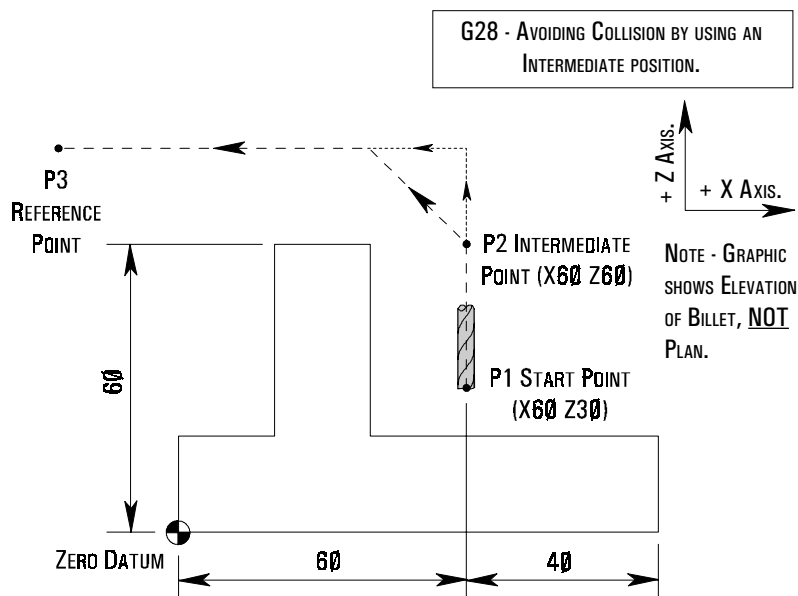
(REFERENCE
POINT RETURN).

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-vectorised movements forcing the tool to follow a path which "cuts" through the edge of the billet.



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.



The above toolpath can be programmed as follows (In absolute mode, G90):

```
G90 G28 X60 Z60 ;
```

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

```
G91 G28 X0 Z40 ;
```

G CODES -

G28

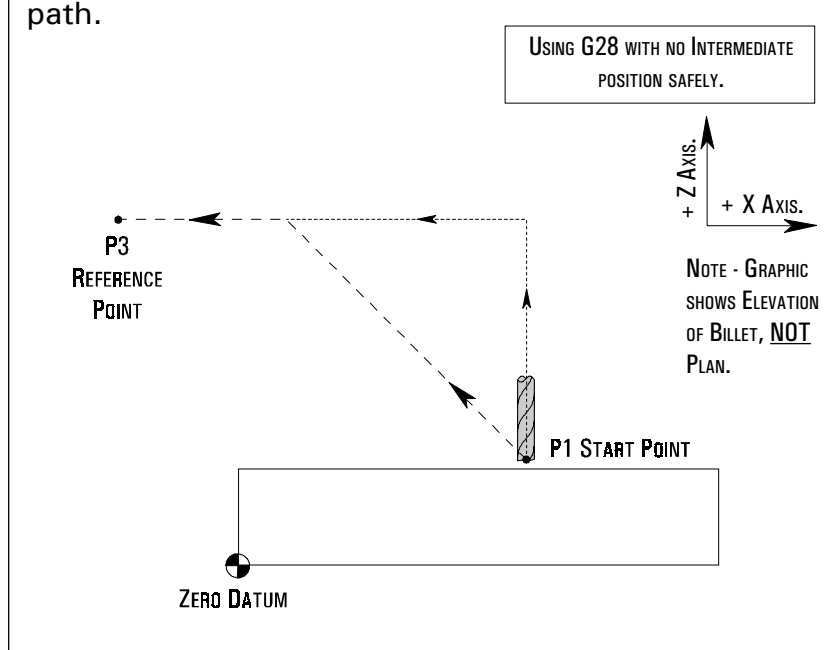
(REFERENCE POINT RETURN).

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 X0 Y0 Z0 ;
```

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 -

G40

G41

G42

(CUTTER COMPENSATION).

The collection of G40, 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.

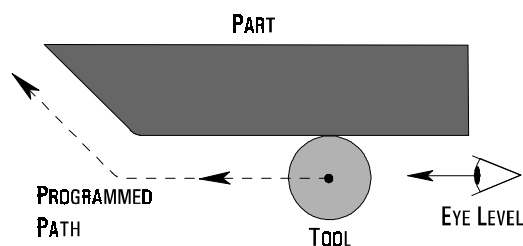
WORK POSITION AND MOVEMENT COMMAND.

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
G40	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

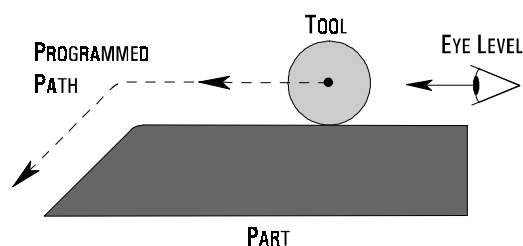
The two diagrams below illustrate the direction of compensation codes G41 and G42, in relation to your eye level.

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 -

G40

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 G01 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 00.

NOTE 1.

A G02 or G03 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 G40, G41 and G42 are modal, belonging to the same modal family. They are incompatible with each other on the same block.

G CODES -

G40

G41

G42

(CUTTER COMPENSATION).

CANCELLATION OF CUTTER COMPENSATION

(G40).

The G40 code is used to cancel cutter compensation.

A G40 command can only be performed in a block in which a linear move (ie, G00, G01, G28) is programmed.

NOTE 1.

Following the machining of an internal pocket, it is recommended that the Z axis is withdrawn by using the G01 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 M02 or M30 command.

G CODES -

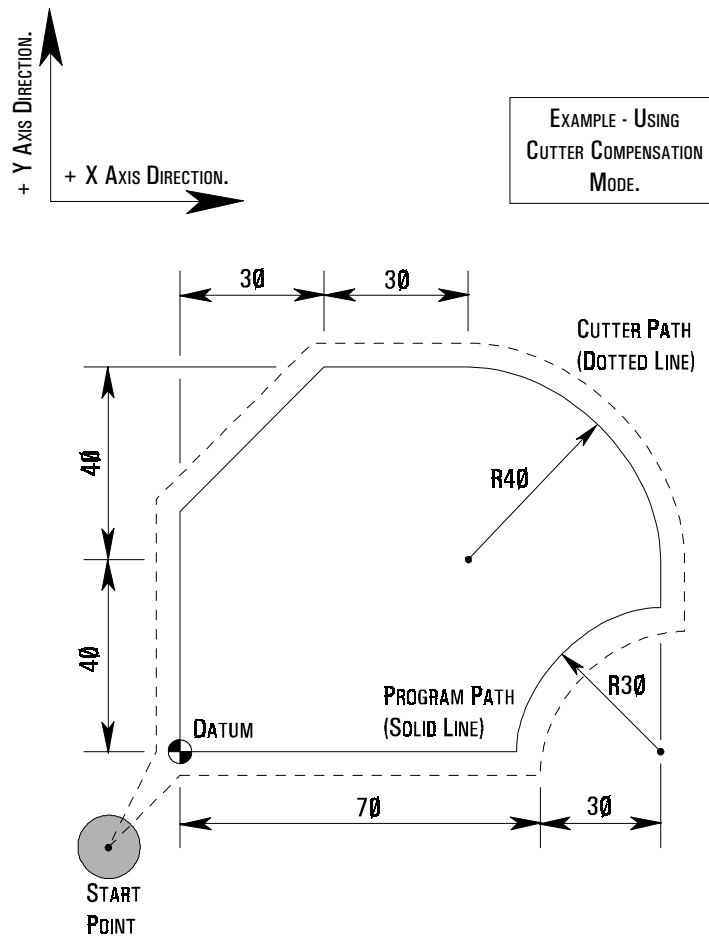
G40

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 0010
N0010 G00 X-15 Y-15 ;
N0020 G41 X0 Y0 F100 ; (Start-Up Move)
N0030 Y40 ;
N0040 X30 Y80 ;
N0050 X60 ;
N0060 G02 X100 Y40 R40 ;
N0070 G01 Y30 ;
N0080 G03 X70 Y0 R30 ;
N0090 X0 ;
N0100 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.

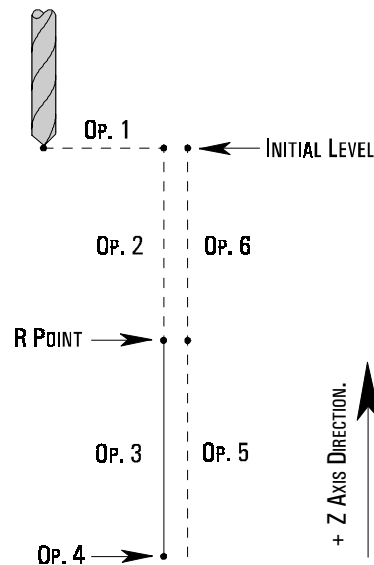
Operation 2 - Rapid traverse in the Z axis to the "R" point.

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).

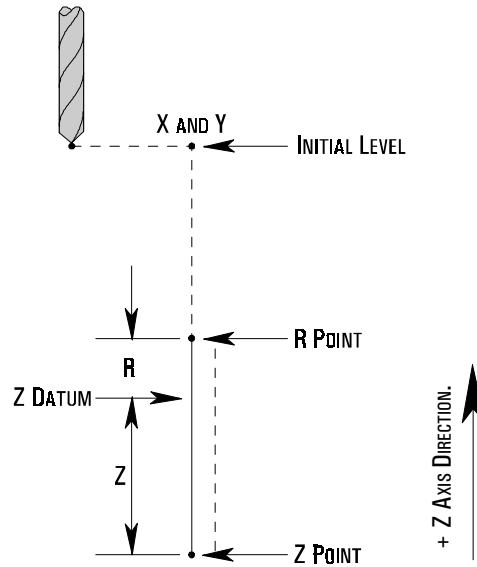
There are three command modes for canned cycles, as follows:

- 1) Data Format (G90 and G91).
- 2) Return Point Level (G98 and G99).
- 3) Cycle Mode (G73 to G89).

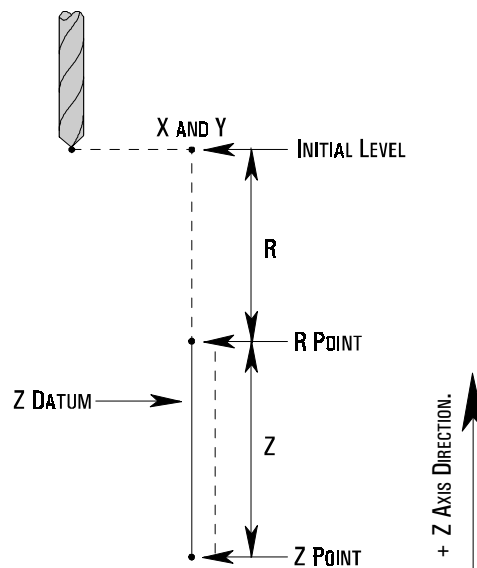
DATA FORMAT COMMAND MODES.

The data format used in canned cycles is specified by the codes G90 and G91, as shown below:

G90 - ABSOLUTE DATA FORMAT.



G91 - INCREMENTAL DATA FORMAT.



G CODES -

G73-

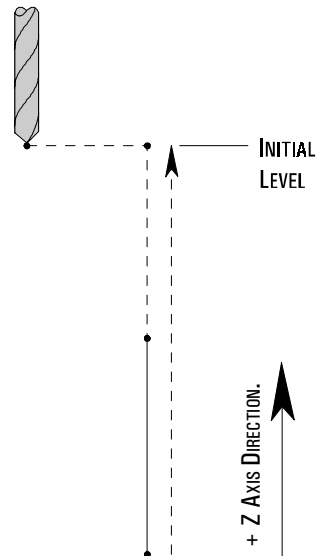
G89

(CANNED CYCLES).

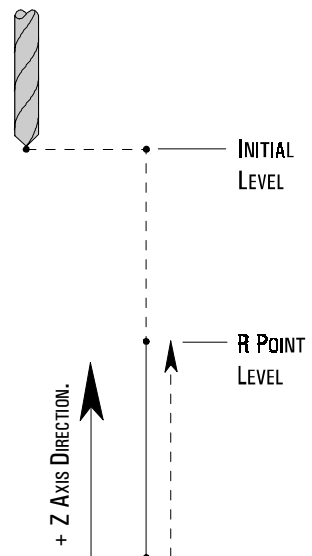
RETURN POINT LEVEL COMMAND MODES.

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 to the canned cycle mode.

G CODES -

G73-

G89

(CANNED CYCLES).

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

(G90) (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 G04 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

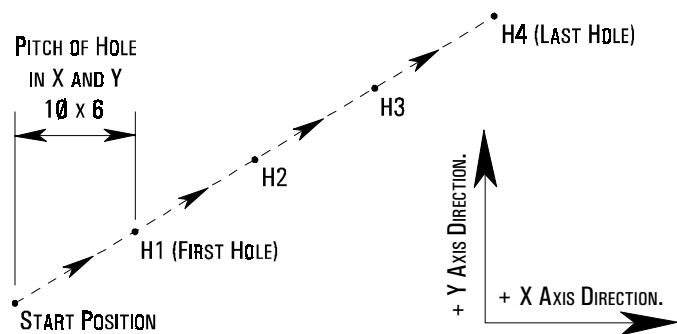
(CANNED CYCLES).

The following example shows a canned cycle for drilling 4 holes, where the third hole is to be machined 1Ømm 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 -

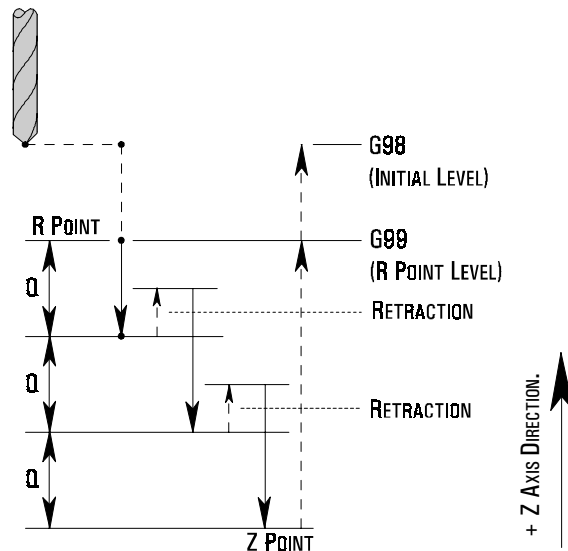
G73

(HIGH SPEED PECK DRILLING).

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

(G90) (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 -

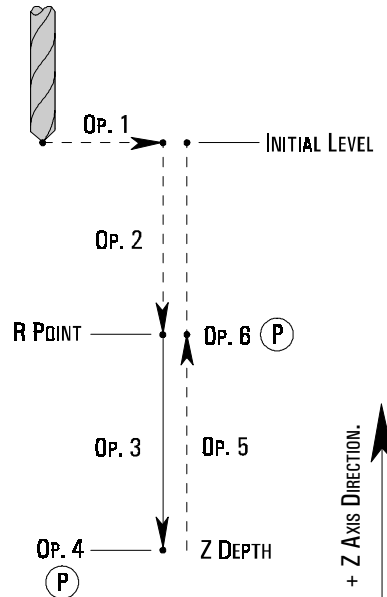
G74

(COUNTER TAPPING).

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

(G90) (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 -

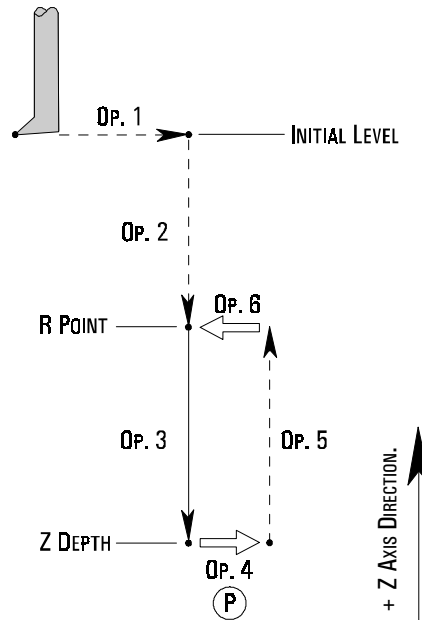
G76

(FINE BORING).

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

(G90) (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 -

G80

(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 G80 code, following the last block of the canned cycle within the part program.

NOTE 1.

The G80 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 -

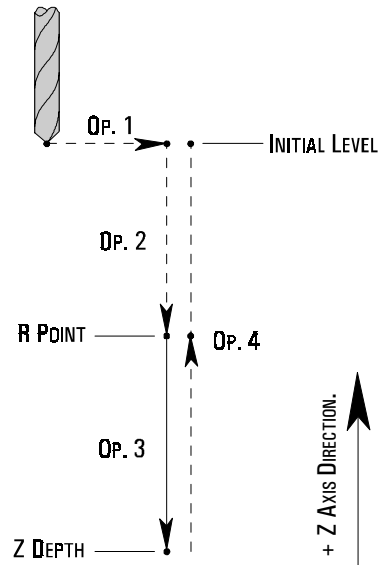
G81

(DRILLING - SPOT BORING).

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

(G90) (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 -

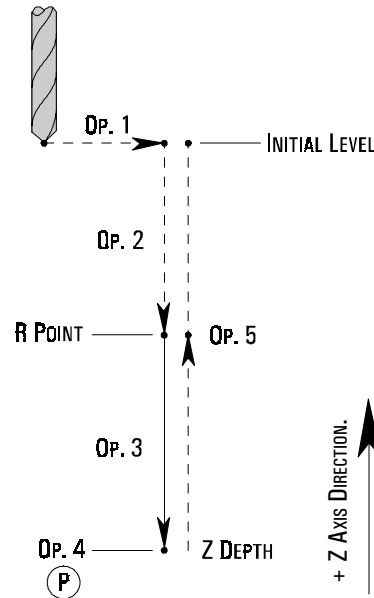
G82

(DRILLING - COUNTER BORING).

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

(G90) (G98)
or or G82 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 for value P.

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

G CODES -

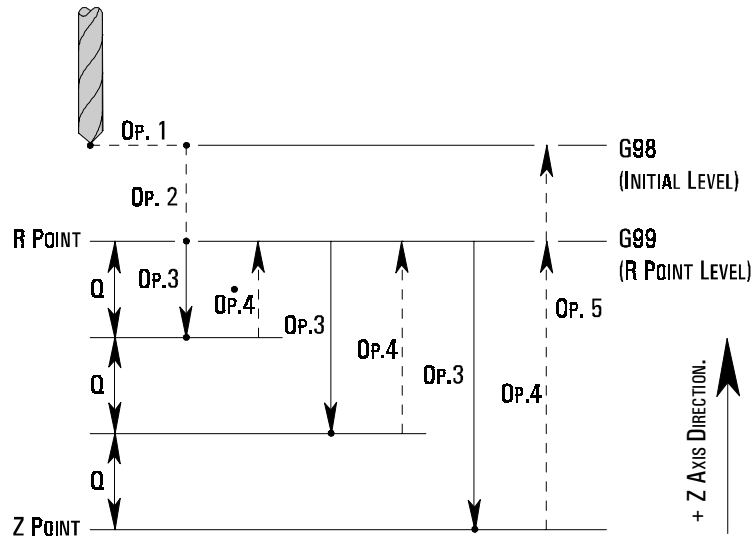
G83

(DEEP HOLE PECK DRILLING).

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

(G90) (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. 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 -

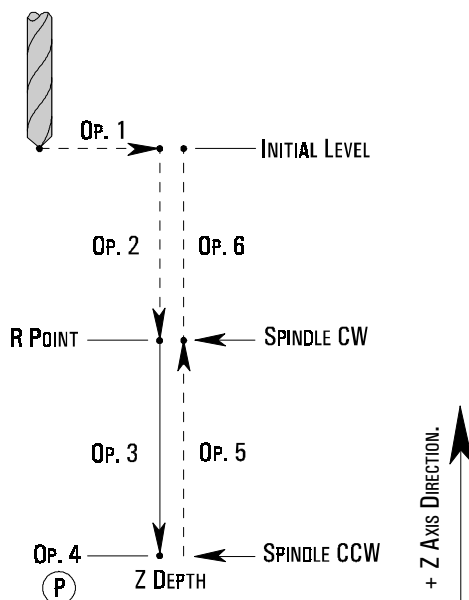
G84

(TAPPING).

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

(G90) (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 -

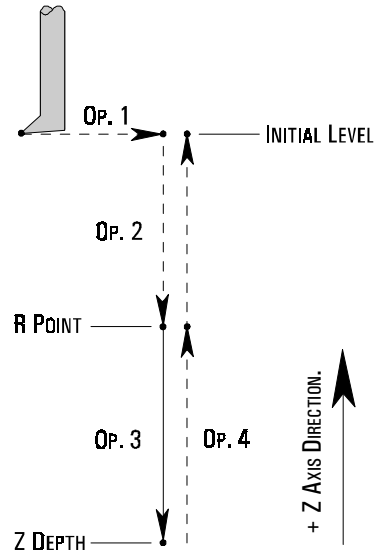
G85

(BORING).

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

(G90) (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 -

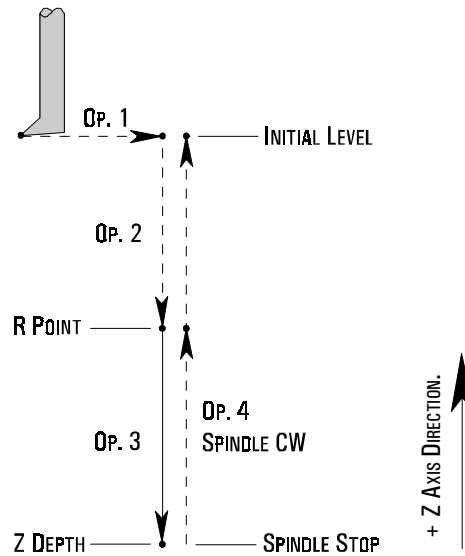
G86

(BORING).

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

(G90) (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 -

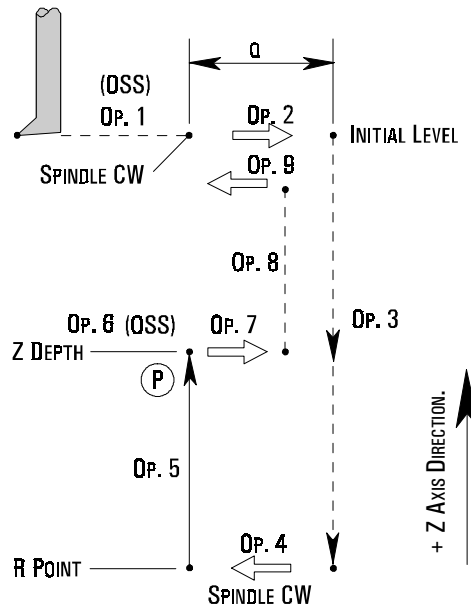
G87

(BACK BORING).

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

(G90) (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 -

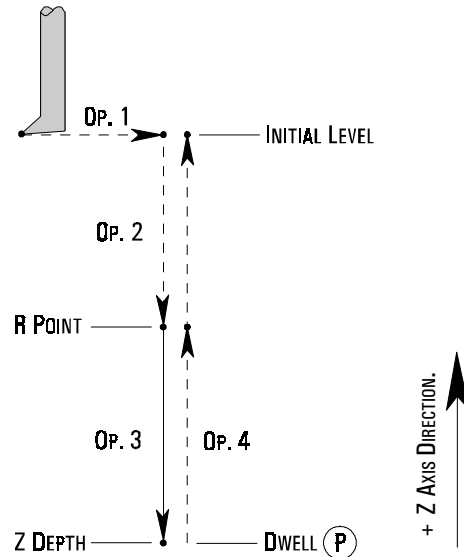
G89

(BORING).

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

(G90) (G98)
or or G86 X.... Y.... Z.... P.... R.... F.... ;
(G91) (G99)

please refer to page 52 for the variable definitions.



Sequence of moves:

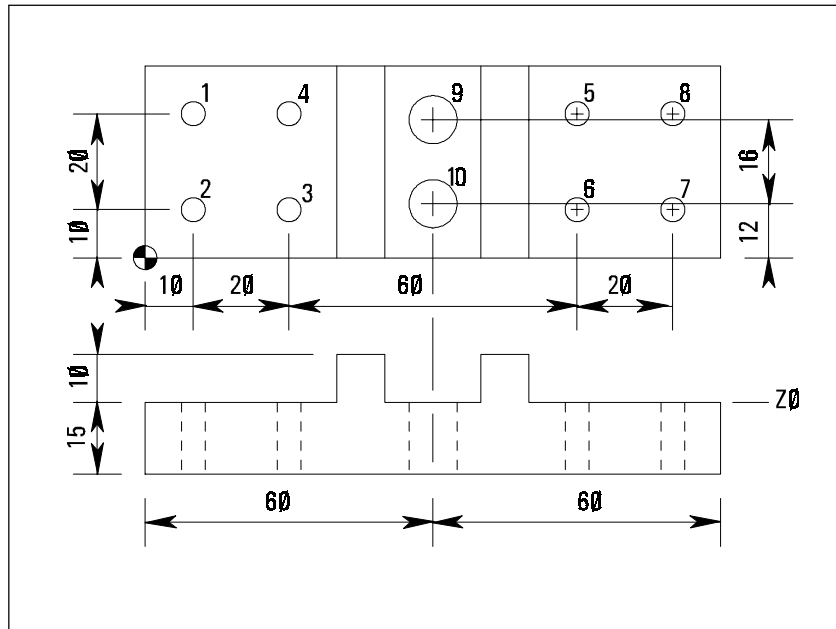
Op 1) Rapid position to X, Y and Z (the initial level).

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.



```

N0040 M06 T01 ;
N0050 G90 G00 X10 Y30 Z12 S1000 M03 ;
N0060 G99 G81 X10 Y30 Z-17 R2 F75 ;
N0070 Y10 ;
N0080 X30 ;
N0090 G98 Y30 ;
N0100 G99 X90 ;
N0110 Y10 ;
N0120 X110 ;
N0130 G98 Y30 ;
N0140 G91 G80 G28 X0 Y0 Z0 M05 ;
N0150 M06 T02 ;
N0160 G90 G00 X60 Y28 Z12 S750 M03 ;
N0170 G99 G83 X60 Y28 Z-17 Q6 R2 F60 ;
N0180 G98 Y12 ;
N0190 G91 G80 G28 X0 Y0 Z0 M05 ;
N0200 M30 ;

```

Tool change.

Tool position to initial level.

Hole 1 retract R point.

Hole 2 retract R point.

Hole 3 retract R point.

Hole 4 retract initial level.

Hole 5 retract R point.

Hole 6 retract R point.

Hole 7 retract R point.

Hole 8 retract initial level.

Home position spindle stop.

Tool change.

Tool position initial level.

Hole 9 retract R point.

Hole 10 retract initial level.

Home position spindle stop.

Program stop.

G CODES -

G90

(ABSOLUTE
ZERO COMMAND).

When G90 is active, all co-ordinates are relative to the workpiece datum (the zero position).

NOTE 1.

The G90 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 G20 Imperial Mode.

For example:

(G20) F6 = 6 in/min.

(G21) F150 = 150 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 G20 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 -

G170-

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:

G170]
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 G170-171 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 0002, 0003 and 0004 are for circular pockets.

Programs 0005, 0006 and 0007 are for rectangular pockets.

G CODES -

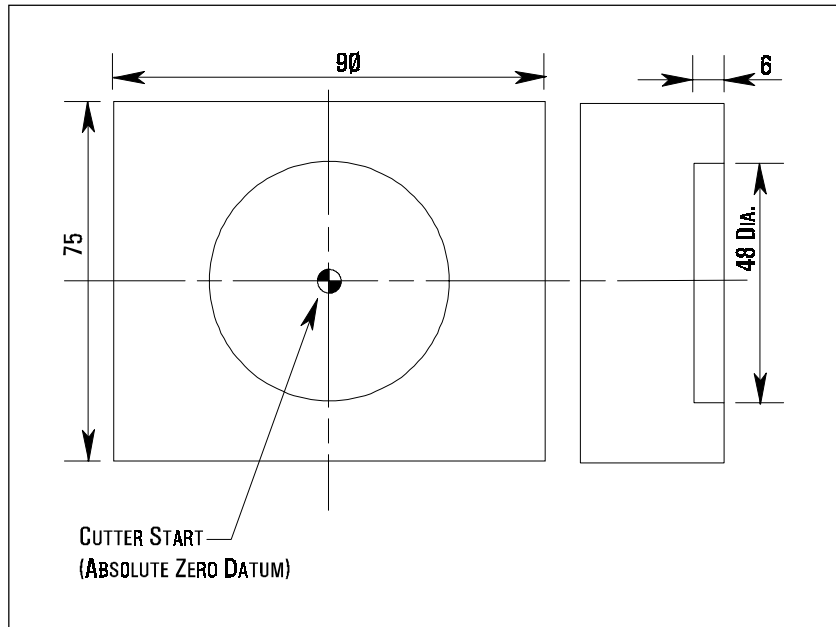
G170

G171

(CIRCULAR POCKET

CANNED

CYCLE EXAMPLE A).



NOTE 1.

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

PROGRAM NUMBER 0002 - ROUGHING PROGRAM - CIRCULAR POCKET - G170 AND G171.

[BILLET X75 Y90 Z30 ;	Billet size.
[EDGEMOVE X-37.5 Y-45 ;	Position of datum from the bottom LH corner of billet.
[TOOLDEF T1 D6 Z0 ;	Tool no. dia. and position.
O 0002 ;	Program no.
N0040 G91 G28 X0 Y0 Z0 ;	Metric, reference point.
N0050 M06 T01 ;	Tool no.
N0060 G90 G00 X0 Y0 Z10 S3000 M03 ;	Absolute, rapid, tool 10mm above surface, spindle speed and start.
N0070 G01 Z0 F300 ;	Tool to surface of job, feed set.
N0080 G170 R0 P0 Q3 X0 Y0 Z-6 I0 J0 K-24 ;	Circular pocket canned cycle.
N0090 G171 P75 S3000 R75 F250 B3500 J200 ; ...	Circular pocket canned cycle.
N0100 G00 Z25 M05 ;	Rapid, tool to 25mm above surface, spindle stop.
N0110 G91 G28 X0 Y0 Z0 ;	Incremental, return to reference point.
N0120 M30 ;	Program reset.

G CODES -

G170

G171

(CIRCULAR POCKET CANNED CYCLE EXAMPLE A).

Definitions for the terms used in the G170 and G171 circular pocket canned cycle from program number 0002 follows:

```
N0080 G170 R0 P0 Q3 X0 Y0 Z-6 I0 J0 K-24 ;
```

```
N0090 G171 P75 S3000 R75 F250 B3500 J200 ;
```

For G170 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 0002, 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 (S3000).

R defines the roughing Feed in Z (75).

F defines the roughing feed XY (250).

B defines the finishing spindle speed (3500, not applicable as roughing only).

J defines the finishing feed (200, 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 0002 is for a two cut roughing cycle.

G CODES -

G170

G171

(CIRCULAR POCKET
CANNED
CYCLE EXAMPLE B).

PROGRAM NUMBER 0003 - ROUGHING AND FINISHING PROGRAM - CIRCULAR POCKET - G170 AND G171.

The difference between program numbers 0002 and 0003 is that program 0003 leaves a finishing allowance on the diameter of the pocket and on the base.

```
[BILLET X75 Y90 Z30 ;  
[EDGEMOVE X-37.5 Y-45 ;  
[TOOLDEF T1 D6 Z0 ;  
O0003 ;  
N0040 G91 G21 G28 X0 Y0 Z0 ;  
N0050 M06 T01 ;  
N0060 G90 G00 X0 Y0 Z10 S3000 M03 ;  
N0070 G01 Z0 F300 ;  
N0080 G170 R0 P0 Q3 X0 Y0 Z-6 I0.5 J0.1 K-24 ;  
N0090 G171 P75 S3000 R75 F250 B3500 J200 ;  
N0100 G00 Z25 M05 ;  
N0110 G91 G28 X0 Y0 Z0 ;  
N0120 M30 ;
```


G CODES -

G170

G171

(CIRCULAR POCKET
CANNED
CYCLE EXAMPLE B).

Definitions for the terms used in the G170 and G171 circular pocket canned cycle from program number 0003 follows:

```
N0080 G170 R0 P0 Q3 X0 Y0 Z-6 I0.5 J0.1 K-24 ;
```

```
N0090 G171 P75 S3000 R75 F250 B3500 J200 ;
```

For G170 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 0003, 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 (S3000).

R defines the roughing Feed in Z (75).

F defines the roughing feed XY (250).

B defines the finishing spindle speed (3500).

J defines the finishing feed (200).

G CODES -

G170

G171

(CIRCULAR POCKET
CANNED
CYCLE EXAMPLE C).

PROGRAM NUMBER 0004 - ONE STEP FINISHING CYCLE

PROGRAM - CIRCULAR POCKET - G170 AND G171.

The difference between program numbers 0003 and 0004 is that in program 0004 the tool moves directly to the finish depth and executes a final finishing pass only.

```
[BILLET X75 Y90 Z30 ;
```

```
[EDGEMOVE X-37.5 Y-45 ;
```

```
[TOOLDEF T1 D6 Z0 ;
```

```
0004 ;
```

```
N0040 G91 G21 G28 X0 Y0 Z0 ;
```

```
N0040 M06 T01 ;
```

```
N0040 G90 G00 X0 Y0 Z10 S3000 M03 ;
```

```
N0040 G01 Z0 F300 ;
```

```
N0040 G170 R0 P1 Q3 X0 Y0 Z-6 I0.5 J0.1 K-24 ;
```

```
N0040 G171 P75 S3000 R75 F250 B3500 J200 ;
```

```
N0040 G00 Z25 M05 ;
```

```
N0040 G91 G28 X0 Y0 Z0 ;
```

```
N0040 M30 ;
```

G CODES -

G170

G171

(CIRCULAR POCKET CANNED CYCLE EXAMPLE C).

Definitions for the terms used in the G170 and G171 circular pocket canned cycle from program number 0004 follows:

```
N0040 G170 R0 P1 Q3 X0 Y0 Z-6 I0.5 J0.1 K-24 ;
```

```
N0040 G171 P75 S3000 R75 F250 B3500 J200 ;
```

For G170 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 (S3000). 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 (250). 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.

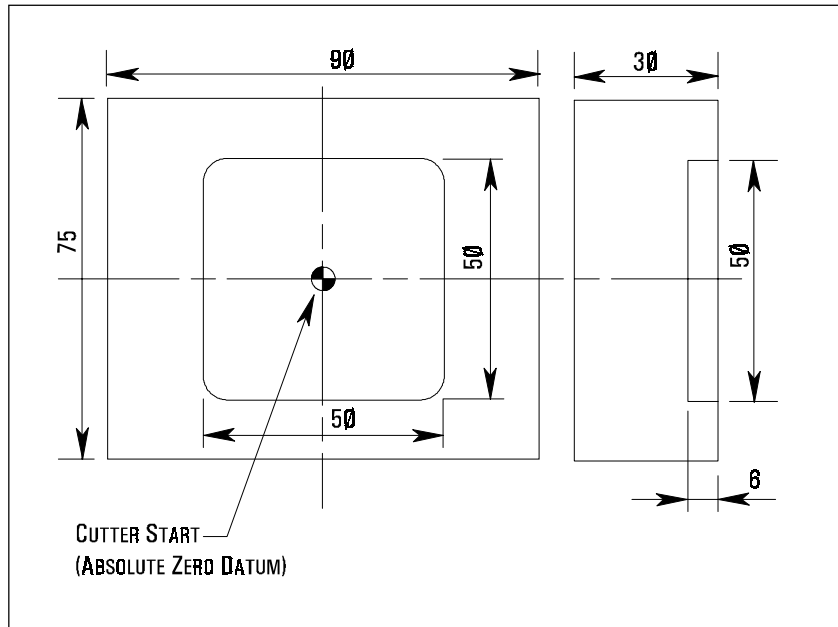
Although characters are ignored by the program when run, in certain modes the characters must be programmed and given a value so that the sequential control of the program is maintained and errors avoided.

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 0005 - ROUGHING PROGRAM - RECTANGULAR POCKET - G172 AND G173.

[BILLET X75 Y90 Z30 ;	Billet size.
[EDGEMOVE X-37.5 Y-45 ;	Position of datum from the bottom LH corner of billet.
[TOOLDEF T1 D6 Z0 ;	Tool no. dia. and position.
O 0005 ;	Program no.
N0040 G91 G21 G28 X0 Y0 Z0 ;	Metric, reference point.
N0050 M06 T01 ;	Tool no.
N0060 G00 X0 Y0 Z10 S3000 M03 ;	Absolute, rapid, tool 10mm above surface, spindle speed and start.
N0070 G01 Z0 F300 ;	Tool to surface of job, feed set.
N0080 G172 I-50 K0 P0 Q3 R0 X-25 Y-25 Z-6 ;	Rectangular pocket cycle.
N0090 G173 I0 K0 P75 T1 S300 R75 F250 B3500 J200 Z5 ;	Rectangular pocket cycle.
N0100 G00 Z25 M05 ;	Rapid, tool to 25mm above surface, spindle stop.
N0110 G91 G28 X0 Y0 Z0 ;	Incremental, return to reference point.
N0120 M30 ;	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 0005 follows:

```
N0080 G172 I-50 K0 P0 Q3 R0 X-25 Y-25 Z-6 ;
```

```
N0090 G173 I0 K0 P75 T1 S300 R75 F250 B3500 J200 Z5 ;
```

For G172 block,

I defines the pocket X length (-50).

J defines the pocket Y length (-50)

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 roughing 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 (3000rpm).

R defines the roughing feed for Z (75).

F defines the roughing feed X and Y (250).

B defines the finishing spindle speed (3500 rpm).

J defines the finishing feed (200).

Z defines the safety Z (5mm above 'R' point).

Program number 0005 is for a two cut roughing cycle.

G CODES -

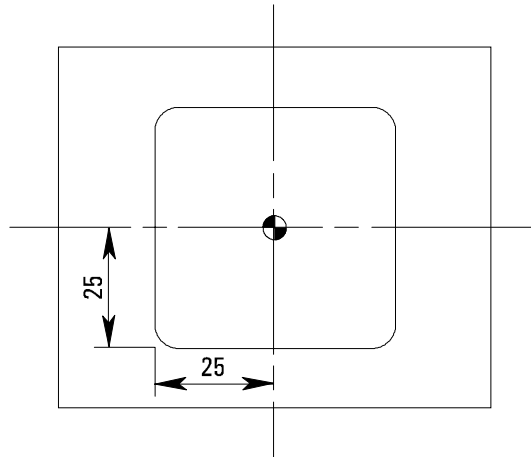
G172

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 \emptyset (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 $\emptyset\emptyset\emptyset5$, 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 (\emptyset), 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 0006 - ROUGHING AND FINISHING
PROGRAM - RECTANGULAR POCKET - G172 AND G173.

```
[BILLET X75 Y90 Z30 ;  
[EDGEMOVE X-37.5 Y-45 ;  
[TOOLDEF T1 D6 Z0 ;  
O0006 ;  
N0040 G91 G21 G28 X0 Y0 Z0 ;  
N0050 M06 T01 ;  
N0060 G90 G00 X0 Y0 Z10 S3000 M03 ;  
N0070 G01 Z0 F300 ;  
N0080 G172 I-50 J-50 K0 P0 Q3 R0 X-25 Y-25 Z-6 ;  
N0090 G173 I0.5 K0.1 P75 T1 S3000 R75 F250 B3500 J200 Z5 ;  
N0100 G00 Z25 M05 ;  
N0110 G91 G28 X0 Y0 Z0 ;  
N0120 M30 ;
```

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 0006 follows:

```
N0080 G172 I-50 J-50 K0 P0 Q3 R0 X-25 Y-25 Z-6 ;
```

```
N0090 G173 I0.5 K0.1 P75 T1 S3000 R75 F250 B3500 J200 Z5 ;
```

For G172 block,

I defines the pocket X length (-50).

J defines the pocket Y length (-50)

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 (0.5 finishing allowance) on the finishing pass.

K defines the pocket base finish (0.1 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 (3000rpm).

R defines the roughing feed for Z (75).

F defines the roughing feed X and Y (250).

B defines the finishing spindle speed (3500 rpm).

J defines the finishing feed (200).

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 0007 - 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 ;  
N0040 G91 G21 G28 X0 Y0 Z0 ;  
N0040 M06 T01 ;  
N0040 G90 G00 X0 Y0 Z10 S3000 M03 ;  
N0040 G01 Z0 F300 ;  
N0040 G172 I-50 J-50 K0 P1 Q3 R0 X-25 Y-25 Z-6 ;  
N0040 G173 I0.5 K0.1 P75 T1 S3000 R75 F250 B3500 J200 Z5 ;  
N0040 G00 Z25 M05 ;  
N0040 G91 G28 X0 Y0 Z0 ;  
N0040 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 0007 follows:

```
N0040 G172 I-50 J-50 KO P1 Q3 R0 X-25 Y-25 Z-6 ;
```

```
N0040 G173 IO.5 KO.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 (3500 rpm).

J defines the finishing feed (200).

Z defines the safety Z (5mm above 'R' point).

NOTE 1.

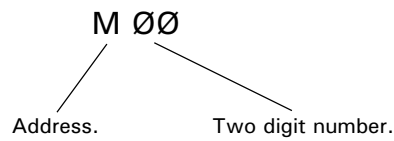
Although characters are ignored by the program when run, in certain modes the characters must be programmed and given a value so that the sequential control of the program is maintained and errors avoided.

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, M03 - spindle forward (CW), M05 - 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,



LIST OF M CODES SUPPORTED BY DENFORD CNC CONTROLS.

Note - Not all M codes apply to each machine.

M code.	Function.
M00*	Program Stop
M01*	Optional Stop
M02*	Program Reset
M03	Spindle Forward (clockwise)
M04	Spindle Reverse (counter clockwise)
M05*	Spindle Stop
M06	Automatic Tool Change
M08	Coolant On
M09*	Coolant Off
M10	Vice/Work Clamp Open
M11	Vice/Work Clamp Close
M13	Spindle Forward and Coolant On
M14	Spindle Reverse and Coolant On
M19	Spindle Orientation
M20	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
M30*	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
M70	Mirror in X On
M71	Mirror in Y On
M76	Wait for Auxiliary Output 1 Off
M77	Wait for Auxiliary Output 2 Off
M80	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 -

M00

M00* - Program Stop.

When the machine controller reads the code M00 within a block, it halts the program. The [CYCLE START] key must be pressed to allow the program to continue.

(PROGRAM STOP).

M CODES -

M01

M01* - Optional Stop.

The M01 code performs the same function as the M00 code, except the machine controller only recognises the signal to halt the program if the optional [STOP] input key is activated.

(OPTIONAL STOP).

M CODES -

M02

M02* - 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 M05.

(PROGRAM RESET).

M CODES -

M03

(SPINDLE FORWARD).

M03 - 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 -

M04

(SPINDLE REVERSE).

M04 - Spindle Reverse (Counter Clockwise).

An M04 code acts in the same way as an M03 code, only the spindle rotates in the opposite direction.

M CODES -

M05

(SPINDLE STOP).

M05* - Spindle Stop.

The M05 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 -

M06

**(AUTOMATIC
TOOL CHANGE).**

M06 - 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 :

M06 T0303 ;

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

M CODES -

M08

(COOLANT ON).

M08 - Coolant On.

This code switches the coolant pump on.

M CODES -

M09

(COOLANT OFF).

M09* - Coolant Off.

This code switches the coolant pump off.

M CODES -

M10

(VICE OPEN).

M10 - 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 M03 and M08 together. The M05 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 -

M20

(ATC ARM IN).

M20 - 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).**

M25 - 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 -

M30

**(PROGRAM RESET
AND REWIND).**

M30* - 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 M30 code is followed by a block number, the program will be reset back to the stated block number.

For example :

M30 P0140 ;

This command is read stop the program running and reset it back to block number 180.

The M30 code also acts as an M05 and M09.

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 -

M70

(MIRROR IN
X ON).

M70 - Mirror in X On.

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

For example:

```
G01 X25.5 ;
```

```
M70
```

```
G01 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 G70, but affects the Y axis.

M CODES -

M80

(MIRROR IN
X OFF).

M80 - 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 memory (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_ _ _ _ .

