CINCINNATI

VICKERS

Basic I.S.O. Programming

Dart A2100
Arrow A2100
Sabre A2100
Lancer A2100

A2100 ISO 1/1/2001

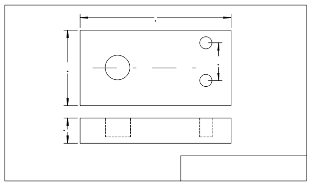
Index

CHAPTER 1

Process from Drawing to Product completion

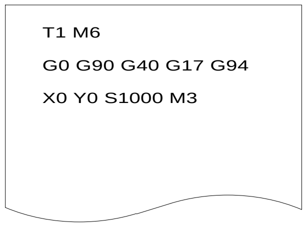
1. Drawing

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



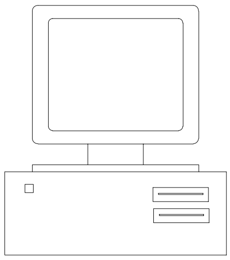
2. Program preparation

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



3. Program creation

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



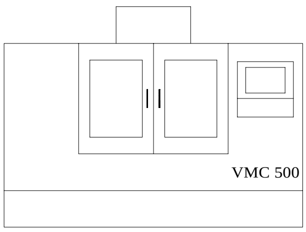
4. Test run

Test the mathematics of the program using the test run facilities i.e. Graphics & program run.

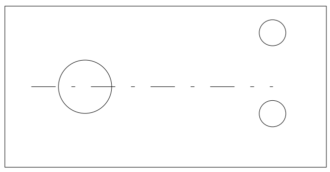
(MACH LOCK)

5. Machining

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



6. Product completion and Inspection



Introduction to Programming

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

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

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

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

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

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

i.e.	N100	S1000	M03
	(Line number)	(Speed)	(Machine function)

The “Block’s” of information sequentially listed form the ***<u>“Program”</u>***

Cutting Condition Commands

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

The following cutting conditions are required for all tooling used:

Spindle Speed – R.P.M. (Revolutions per Minute)

Designated with an S command.

400 rpm ⇒ S400

Formula

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

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

Feedrate – mm/min. , inch/min. , feed/tooth, feed/rev.

Designated with an F command.

400 mm/min. ⇒ F400

Formula

Feed = Number of teeth x feed/tooth (pitch) x R.P.M.

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

Table of preparatory Codes (G & M Functions)

a) All Codes are divided into group types.

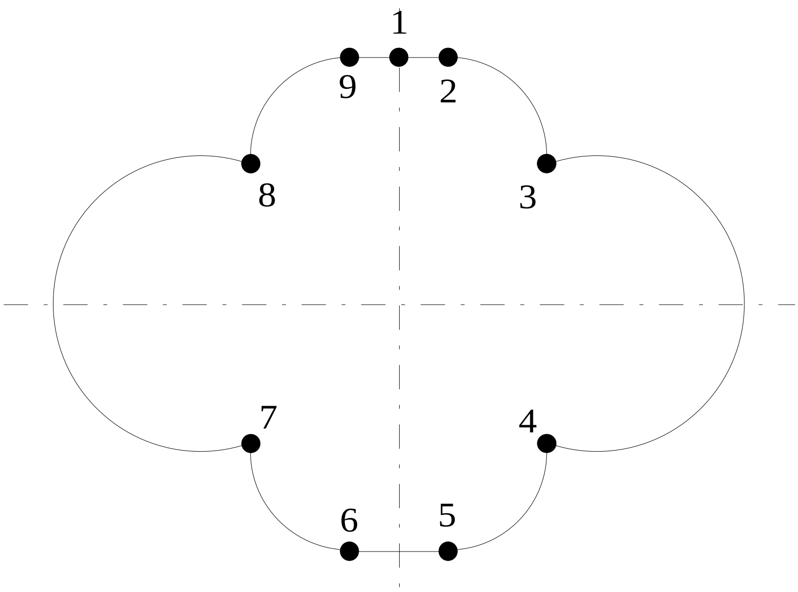
b) There are two types of “G” & “M” codes:

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

Modal - The code remains active when programmed for every line of program and until it is replaced by another action code of the same group type.

i.e. G01 and G00 are modal codes in group “Interpolation”

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



1	2	
2	3	
3	4	
4	5	
5	6	
6	7	
7	8	
8	9	
9	1	

G & M Functions

1) “G” codes marked ☐ on the next page are initial (defaulted) “G” codes when the power is turned on.

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

3) If a “G” code not listed or not purchased as an optional extra is commanded, an alarm will be displayed.

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

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

Note:

Operators must note that programming G70/G71 will convert all information in fixture-offset registers & tool registers when the program is executed, and therefore programs can be stored in the library of either inch or metric format.

G Codes

☐ = Configurable Default settings at "Power On".

** = Non-Configurable settings.

(M) = Non-Modal codes which can be used with machine motion blocks.

G CODE	GROUP	FUNCTION (* Option)
☐G00	Interpolation	Rapid Positioning
G01		Straight Line "Feed"
G02		Circular Clockwise "Feed"
G02.01		Circular Clockwise "Feed" (I,J,K values are Absolute)
G02.02		Circular Clockwise "Feed" (I,J,K values are Incremental)
G03		Circular Anti-Clockwise "Feed"
G03.01		Circular Anti-Clockwise "Feed" (I,J,K values are Absolute)
G03.02		Circular Anti-Clockwise "Feed" (I,J,K values are Incremental)
G04	Non-Modal	Dwell
G07.1	Interpolation	Cylindrical Interpolation *
G08	Non-Modal	Suppress Interpolation
G09	Non-Modal (M)	Exact Stop
G12	Non-Modal	Contouring Rotary Axis Unwind *
**G13.1	Polar Coordinate Interpolation	Cylindrical Interpolation Cancel *
☐G15.1	Polar Program	Polar Co-ordinates Programming (Bolt Circle)
G15.2	Polar Program	Polar Co-ordinates Command (Part Contour)
☐G17	Plane Select	XY Plane – Plan View (Z- Direction)
G18	Plane Select	XZ Plane – Front View (Y- Direction)
G19	Plane Select	YZ Plane – Side View (X- Direction)
G22, G22.1	Interpolation	Milling Cycle Rectangular Face
G23, G23.1		Milling Cycle Rectangular Pocket
G24, G24.1		Milling Cycle Inside Frame
G25, G25.1		Milling Cycle Outside Frame
G26		Milling Cycle Circular Face
G26.1		Milling Cycle Circular Pocket
G27		Milling Cycle Inside Frame
G27.1		Milling Cycle Outside Frame
G28 (M)	Non-Modal	Auto Return To Reference Position
G29 (M)	Non-Modal	Auto Return From Reference Position
G36	Non-Modal	Move To Next Operation Location
G36.1	Non-Modal	Move To Next Operation Location
**G37	Pattern Cycles	Cancel Pattern
G38		Rectangular Pattern
G39		Circular Pattern
**G40	CDC	Cutter Diameter Compensation Cancel
G41		Cutter Diameter Compensation Left
G42		Cutter Diameter Compensation Right
G43		PQR Cutter Diameter Compensation Right

☐G45	ACC / DEC	Acceleration / Deceleration ON
G45.1		Acceleration / Deceleration ON (Die Roughing) *
G45.2		Acceleration / Deceleration On (Die Finishing) *
G46		Acceleration / Deceleration OFF
G50	Non-Modal	Pallet Co-ordinates
G52	Local Coordinates	Datum Shift
G60	Cornering	Positioning Mode
☐G61		Contouring Mode
G61.1		Cutter Path Left of Work
G61.2		Cutter Path Right of Work
G61.3	Non-Modal	Automatic Corner Speed Overide
G70	Inch / Metric	Inch Programming
☐G71		Metric Programming
G80	Interpolation	Cancel Fixed Cycle
G81		Simple Drilling Cycle
G82		Counterbore / Spotdrill Cycle With Dwell
G83		Peck Drilling Cycle
G84		Tap Cycle (Conventional)
G84.1		Rigid Tap Cycle **(dependant on machine type)*
G85		Bore / Ream Cycle
G86		Finish Bore Cycle
G87		Back Bore Cycle
G88		Web Drill / Bore Cycle
G89		Bore / Ream With Dwell Cycle
☐G90	Abs / Inc	Absolute Dimensioning
G91		Incremental Dimensioning
G92	Non-Modal	Position Set
G92.1	Non-Modal	Multi-Setup Offsets Setting
G92.2	Non-Modal	Pallet Offsets Setting
G93	Feedrate	Inverse Time Feedrate
☐G94		Feed Per Minute Feedrate Mode
G95		Feed Per Revolution Mode
G96	Spindle	Constant Surface Speed
☐G97		Constant Spindle Speed
G98	Non-Modal (M)	Machine Coordinates using Tool Tip
G98.1		Machine Coordinates using Spindle Nose
G99	Non-Modal	Position Set Cancel
**G150	Scaling	Scaling OFF
G151		Scaling ON

All groups settings are reset to its default state at Program End, Data Reset or a Colon Block.

G4 Code Dwell

Programmable dwell provides the capability to delay program execution for a specific period.

Dwell is programmed using a G4 code with either a F or S word.

The S word is used to specify a dwell period in spindle revolutions.

The F word is used to specify a dwell period in seconds.

A block containing a dwell code cannot contain any other information other than the F or S word and a line number.

Examples:

G4 – Dwell for 0.5 seconds

G4 S5 – Dwell for 5 spindle revolutions at the current spindle speed

G4 F5 – Dwell for a time of 5 seconds

Type II NC Block Format

The control supports some extensions to EIA-274D that require additional programming information. This is accomplished using parentheses ( ) to enclose the type II block. A type II block contains a 3-character command followed by a variable number of program words specifying the additional information needed by the command.

* option (Advanced programming)

Type	NC Format
Alarm *	(ALM, < other info >)
Call sub-program	(CLS, < other info >)
Define sub-program	(DFS, < other info >)
End of sub-program	(ENS)
Journals	(JRN, < other info >)
Messages	(MSG, < other info >)
Mirror Image	(INV, < other info >)
Operator Input *	(INP, < other info >)
Operator query *	(OPR, < other info >)
Page format *	(PAG, < other info >)
Printing *	(PRT, < other info >)
Rotation	(ROT, < other info >)
Set high limit	(SHI, < other info >)
Set low limit	(SLO, < other info >)
Write to file *	(WTF, < other info >)

M codes

M Code	FUNCTION (* *Option)*	START OF SPAN	END OF SPAN
M00	Program Stop		•
M01	Program Stop by switch		•
M02	End Of Program		•
M03	Spindle Clockwise	•	
M04	Spindle Anti-Clockwise	•	
M05	Spindle Stop		•
M06	Toolchange	•	
M08	External Coolant On	•	
M8.1 – M8.8	Coolant Jet Position Number *	•	
M09	Coolant Off		•
M10	4th Axis Unclamp *	•	
M11	4th Axis Clamp *		•
M13	Spindle Clockwise With External Coolant	•	
M14	Spindle Anti-Clockwise With External Coolant	•	
M19	Spindle Orientates To Toolchange Position		•
M26	Z Axis Full Retract		•
M27	Thro' Spindle Coolant *	•	
M30	End Of Program – Return Spindle tool to Drum		•
M50	5th Axis Unclamp *	•	
M51	5th Axis Clamp *		•
M60	Swarf Conveyor ON *	•	
M61	Swarf Conveyor OFF *	•	
M70 – M79	User Definable M Codes *	•	
M83	Part Counter		•

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

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

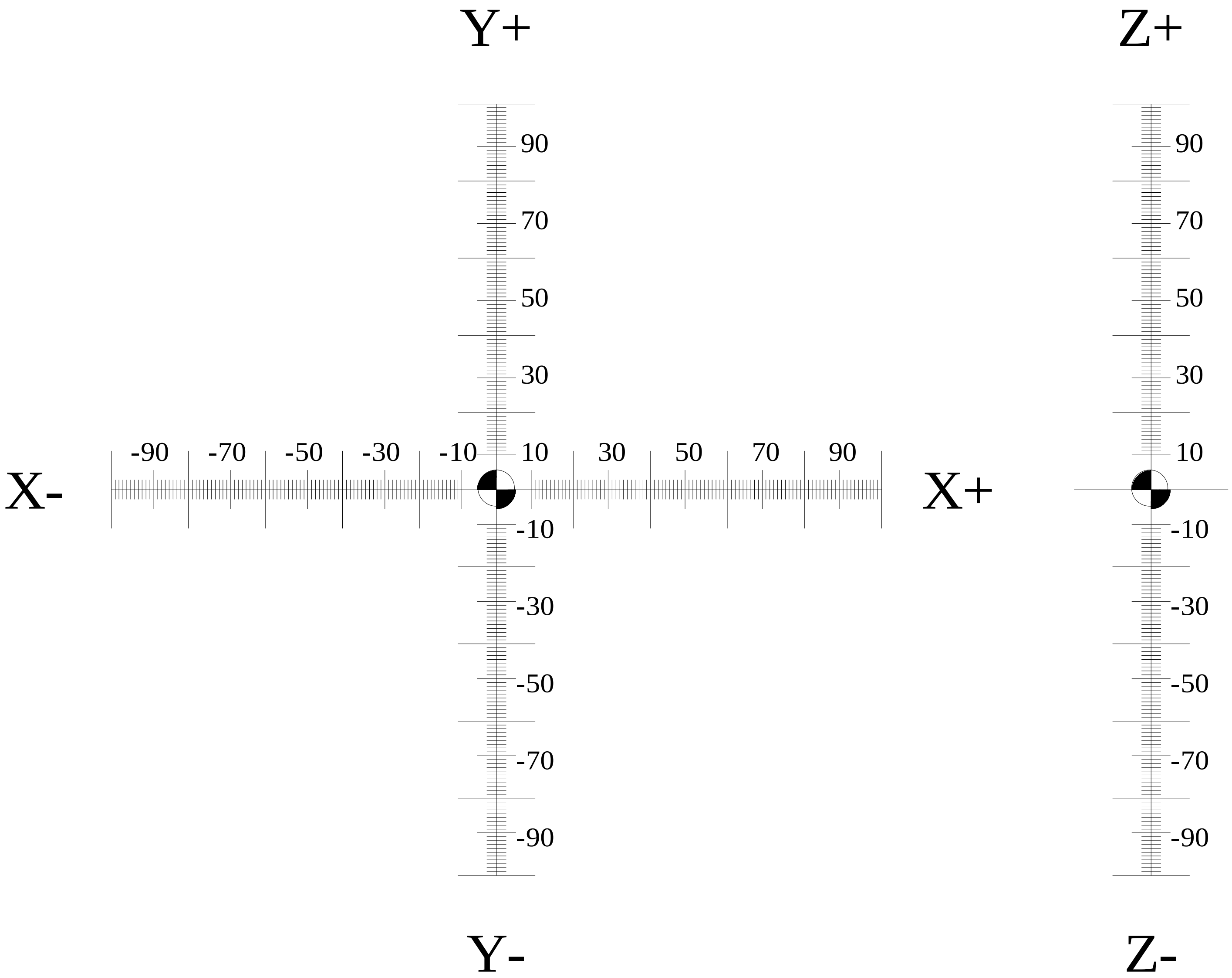
M02 – Program End:
This code informs the control that the program is at the end and will automatically rewind the program for the next process start. The control screen will prompt the operator to "Open & Close the door" before starting the next operation. The spindle tool remains in the spindle until the next toolchange command.

M30 – Program End:
This code informs the control that the program is at the end and will automatically replace the spindle tool back into the appropriate carousel pocket storage and rewind the program for the next process start. The control screen will prompt the operator to "Open & Close the door" before starting the next operation.

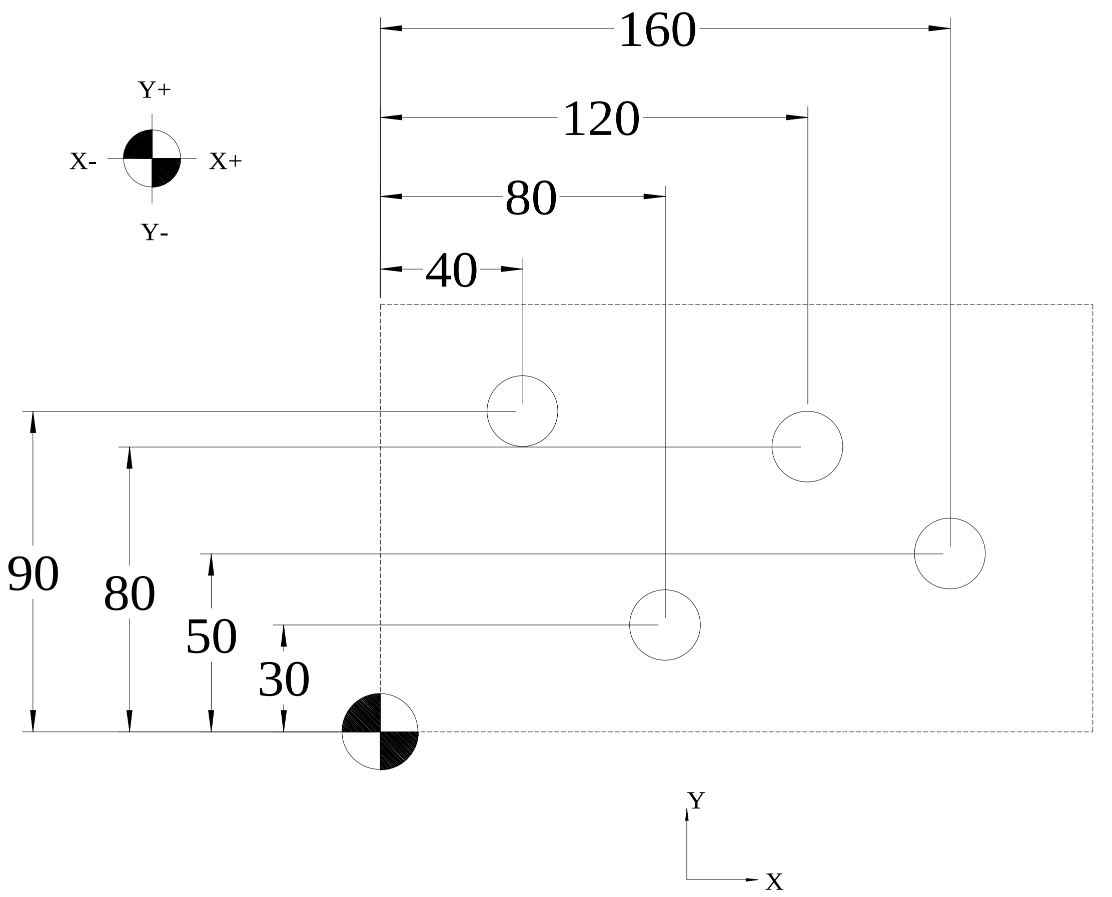
CHAPTER 2

<u>“Absolute (G90) Scale”</u>

<u>*Tool motion assumes now that the spindle moves and not the Table*</u>

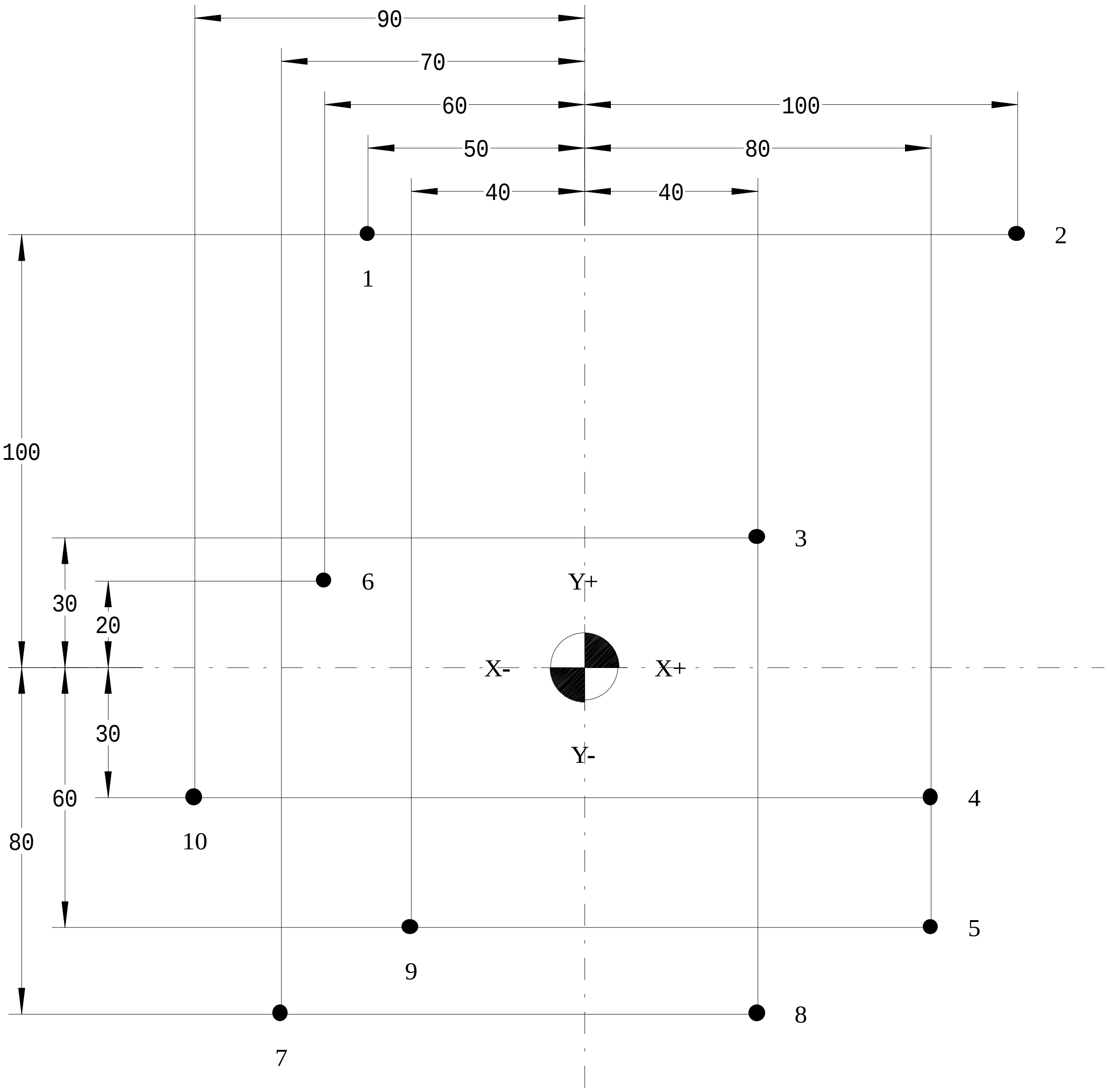


G90 Absolute Programming



G90 X40 Y90
X80 Y30
X120 Y80
X160 Y50

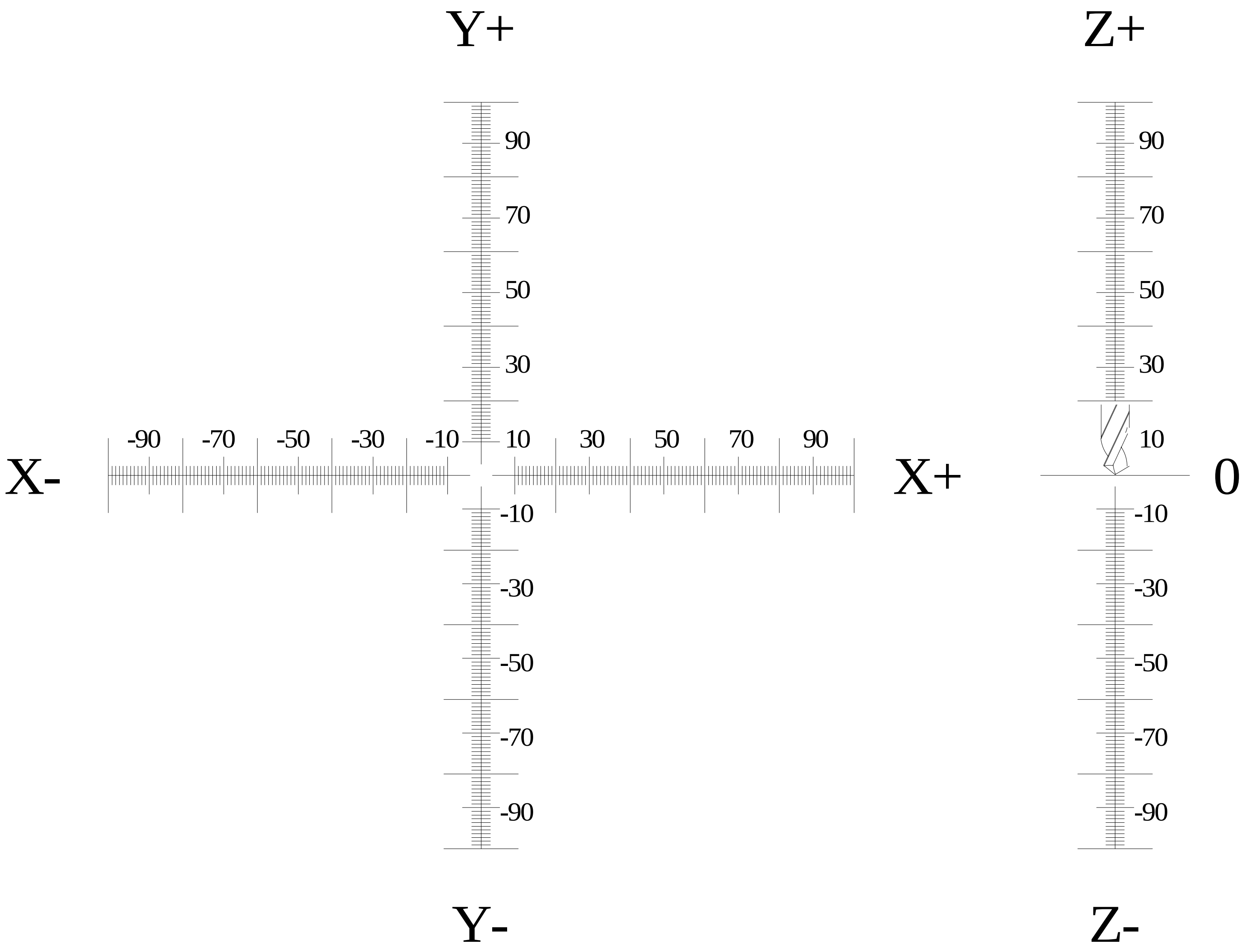
G90 Absolute Example Programming



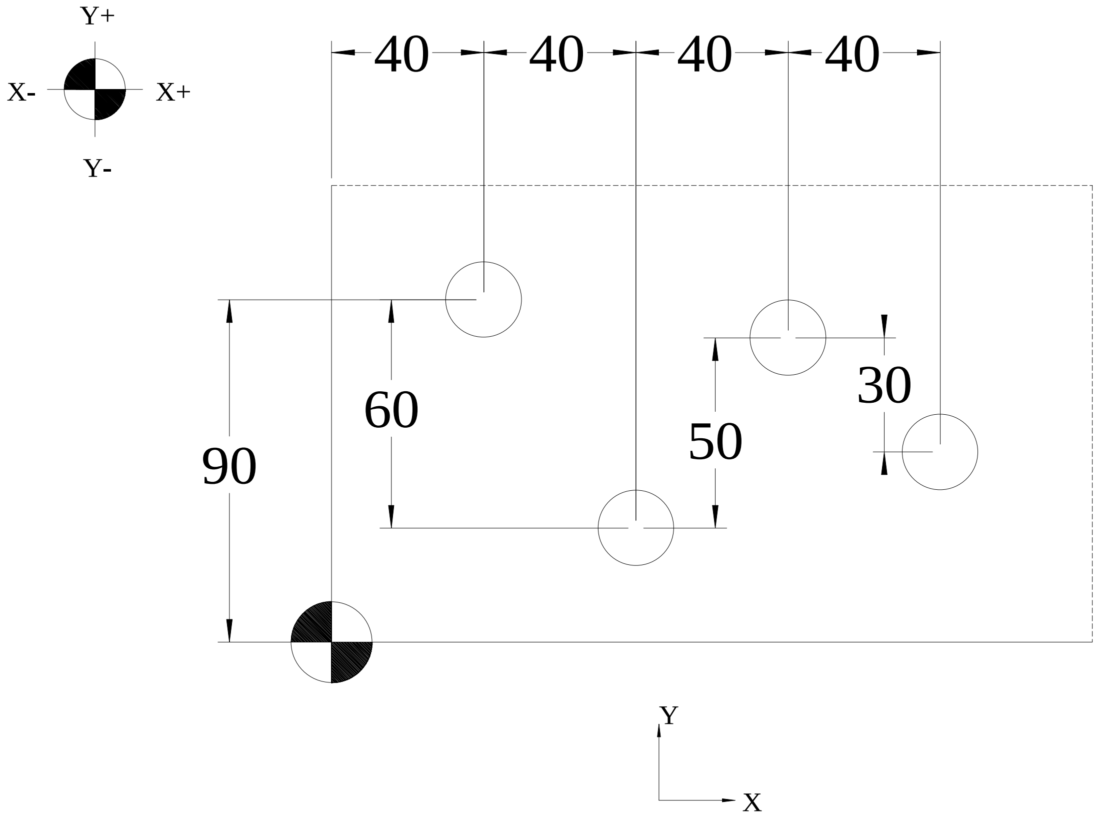
N1 G90 X-50 Y100; Absolute Move to position 1
N2
N3
N4
N5
N6
N7
N8
N9
N10

“Incremental (G91) Scale”

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



G91 Incremental Programming



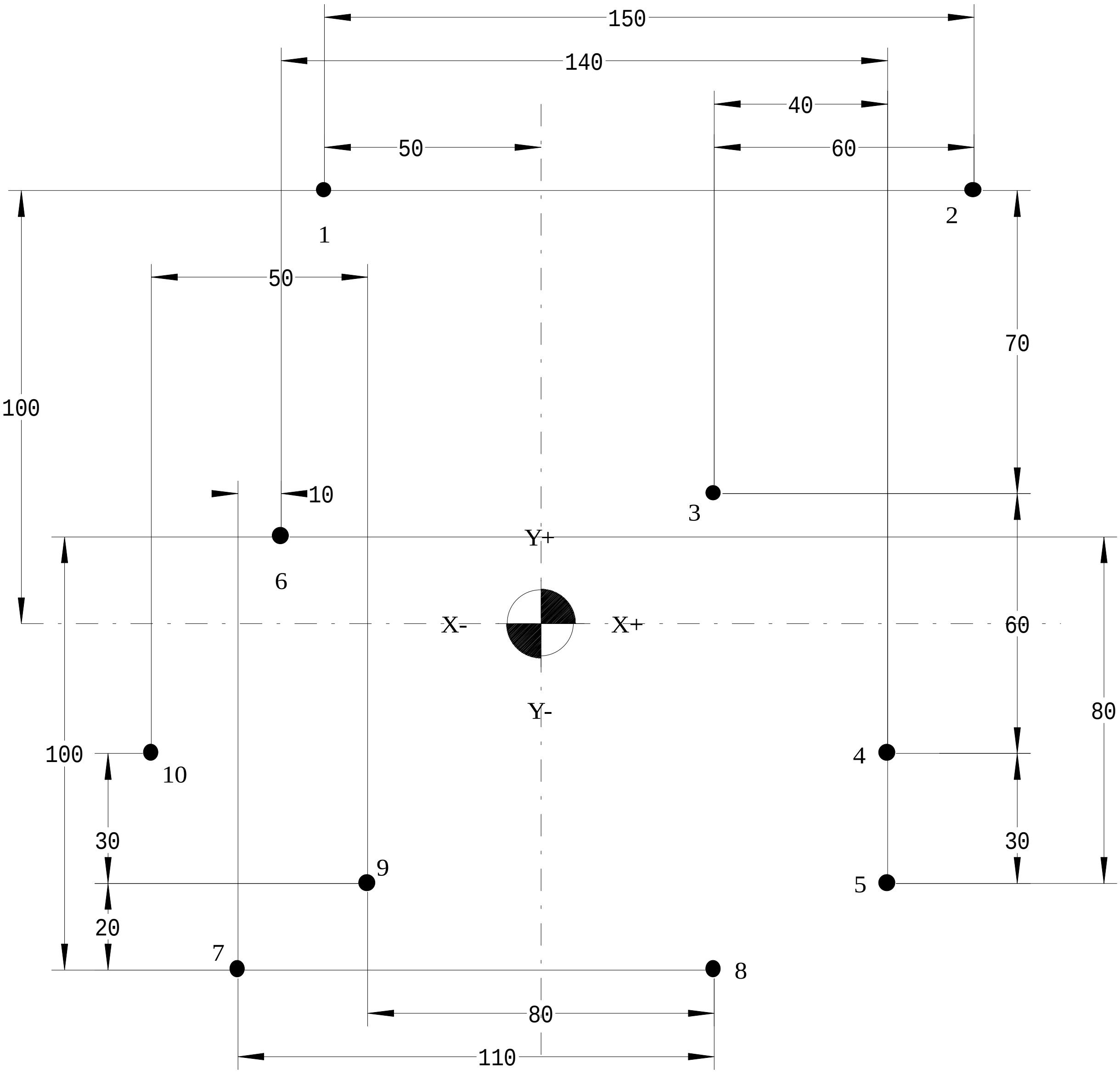
G90 X40 Y90

G91 X40 Y-60

X40 Y50

X40 Y-30

G91 Incremental Example Programming



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

CHAPTER 3

Table Fixture Offsets

1) Pallet Offset = Absolute distance from the Machine Reference

Pallet Offsets

	X	Y	Z
1	+0.0000	+0.0000	+0.0000

2) Multi-Setup Offset = Incremental distance from the Pallet Offset
(64 Offsets = 64 Pallets)

Multi-Setup Offsets: Pallet 1

	X	Y	Z
1	+0.0000	+0.0000	+0.0000
2	+0.0000	+0.0000	+0.0000
3	+0.0000	+0.0000	+0.0000
4	+0.0000	+0.0000	+0.0000

“H” Offsets

3) Fixture Offset = Incremental distance from the “Active” Multi-Setup Offset
(32 Offsets = 32 Parts)

Fixture Offsets: Pallet 1, Setup 1.

	X	Y	Z
1	+0.0000	+0.0000	+0.0000
2	+0.0000	+0.0000	+0.0000
3	+0.0000	+0.0000	+0.0000
4	+0.0000	+0.0000	+0.0000
5	+0.0000	+0.0000	+0.0000

“D” Offsets

4) NC Program Offset = Incremental distance from the “Active” Multi-Setup Offset
(32 Offsets = 32 Trims)

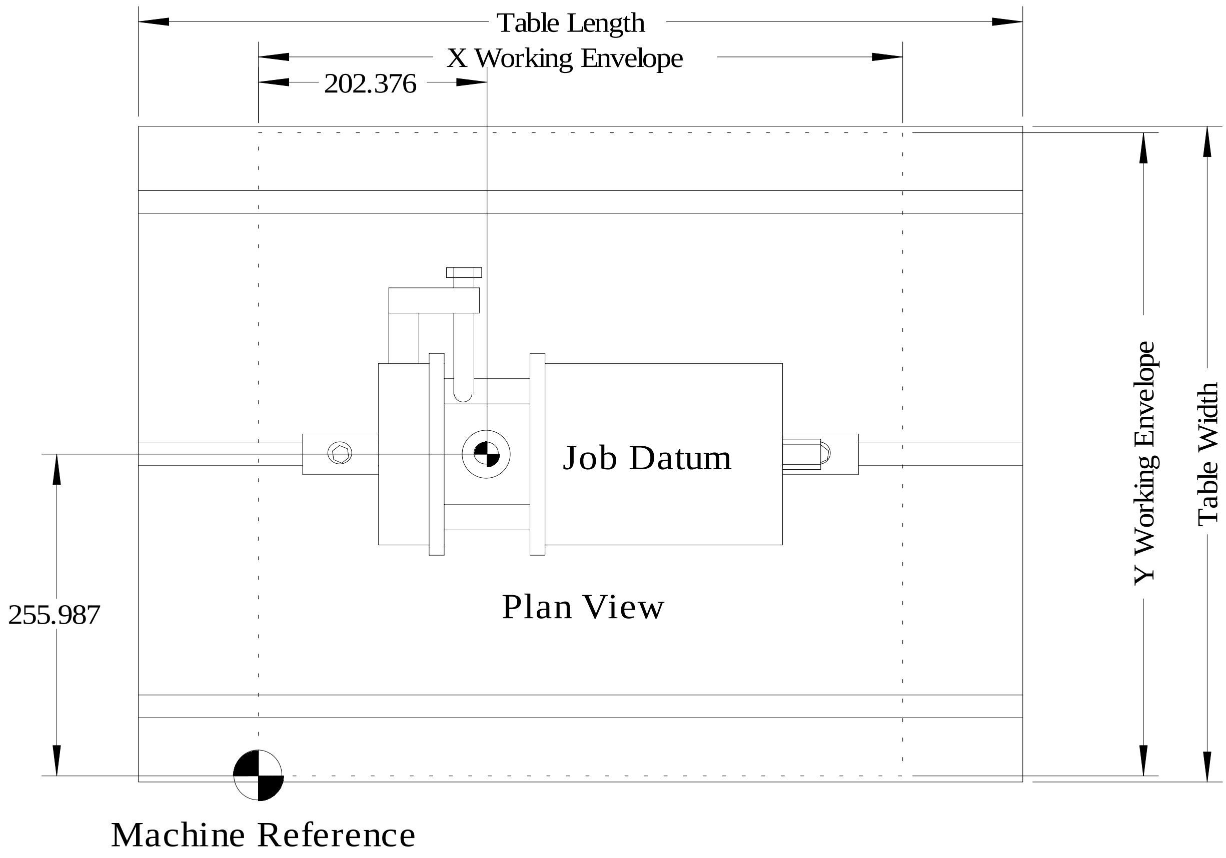
NC Program Offsets: Pallet 1, Setup 1.

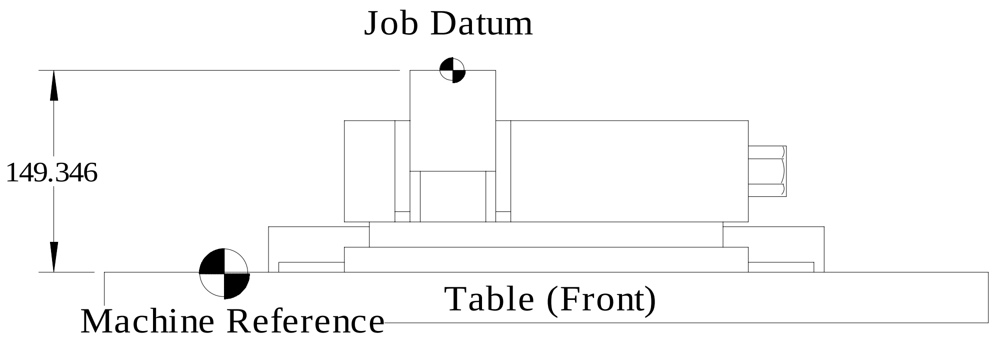
	X	Y	Z
1	+0.0000	+0.0000	+0.0000
2	+0.0000	+0.0000	+0.0000
3	+0.0000	+0.0000	+0.0000
4	+0.0000	+0.0000	+0.0000
5	+0.0000	+0.0000	+0.0000

Note:

Each Multi-Setup Offset row has 32 Fixture Offsets and 32 NC Program Offsets. Make sure when setting the “H” & “D” Offsets that the appropriate Multi-setup offset row has been pre-selected.

Table Fixture Offsets

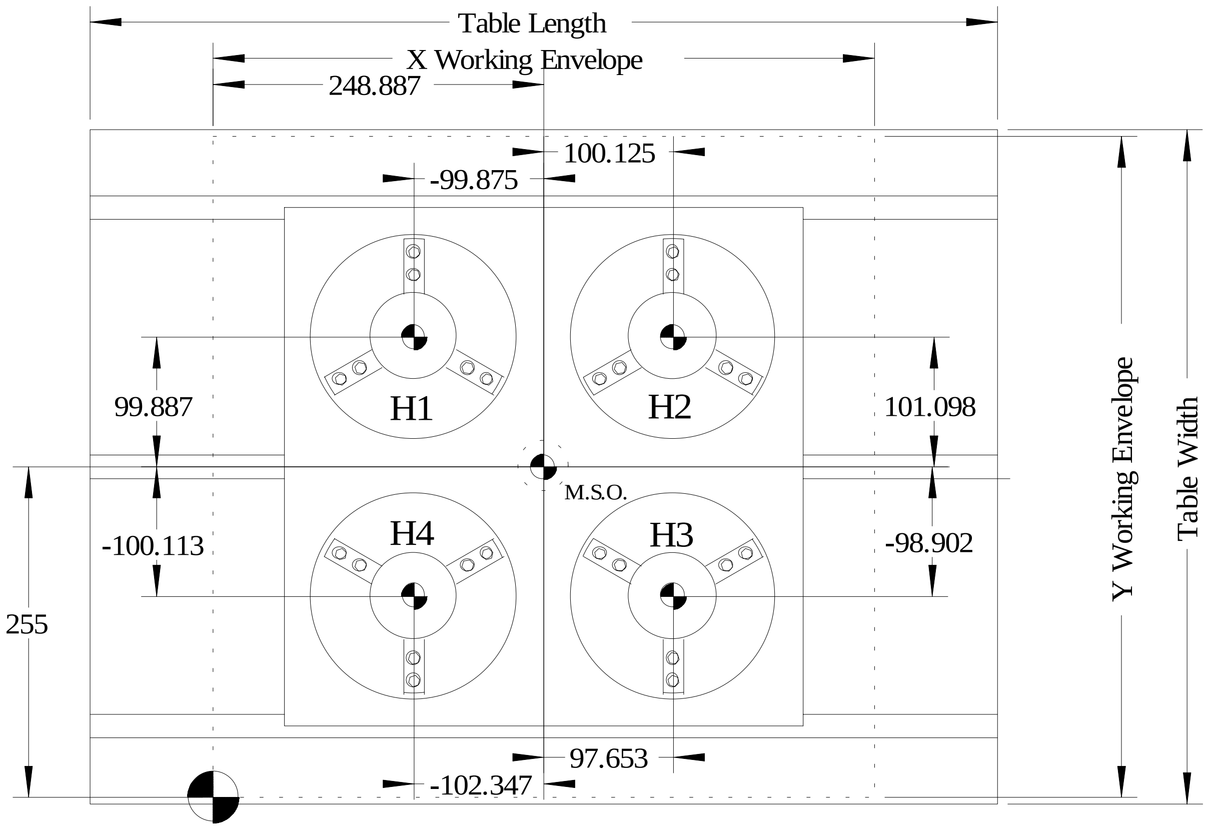


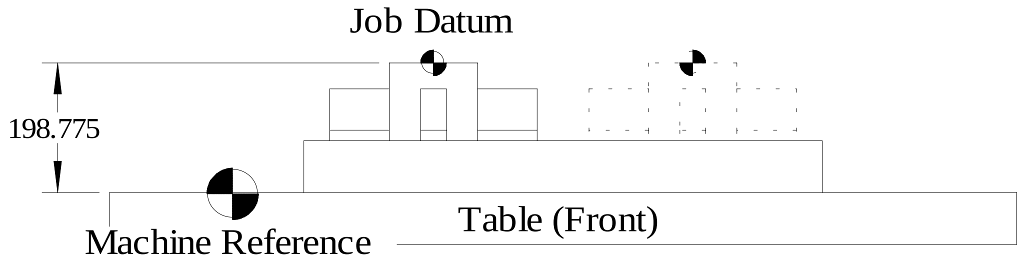


Fixture Offsets: Pallet 1, Setup 1.

	X	Y	Z
1	+202.3760	+255.9870	+149.3460
2	+0.0000	+0.0000	+0.0000
3	+0.0000	+0.0000	+0.0000
4	+0.0000	+0.0000	+0.0000
5	+0.0000	+0.0000	+0.0000

Table Fixture Offsets





Multi-Setup Offsets: Pallet 1

	X	Y	Z
1	+248.8870	+255.0000	+0.0000
2	+0.0000	+0.0000	+0.0000
3	+0.0000	+0.0000	+0.0000
4	+0.0000	+0.0000	+0.0000

Fixture Offsets: Pallet 1, Setup 1.

	X	Y	Z
1	-99.8750	+99.8870	+198.7750
2	+100.1250	+101.0980	+198.7750
3	+97.6530	-98.9020	+198.7750
4	-102.3470	-100.1130	+198.7750

The values contained in the “Fixture Offsets” can be stored in the control as a file or in the program using the system variables to save setting on the next batch.
The values are incremental from the “Multi-Setup Offset” so will never change. The operator only has to set the main setting bore on the next batch set-up in XY.

CHAPTER 4

Program Start

:T? M6 ; Tool-change line
(MSG, Rapid / Linear / Feed - Absolute - Initial start of program)
G0 G90 G40 G71 G17 G94 ; Safety default line.
X? Y? Z? H? S? M? ; Initial move

Toolchange information (Set by the programmer as required)
: - Colon Block required to start or restart a program and can be configured under *Machine Application* to act as a Data Reset.
T? - Tool number – 1, 2 or 3 Digit number (Reference number)
- 4-9 Digit number (Identifier) (minimum number = 1001).
M6 - Toolchange code.

Text messages (Set by the programmer as required)
; - Program Text (not visible on screen) - 132 characters limit.
(MSG, ?) - On screen message (an independent line) - 132 characters limit.

Program Defaults (Set by the programmer as required)
G0 - Maximum Rapid Traverse of the machine
G90 - Absolute Co-ordinates taken from Datum set position
G40 - Cutter Compensation Cancel (Cutter follows program centreline path).
G71 - Metric Dimensions. (G70 = Imperial Dimensions).
G17 - X & Y Plane (Tool is in the Z axis - Spindle)
G94 - Programmed feed is in mm/min.(G95 = mm/tooth) dependant on metric or imperial selection.

Any others may be set by the programmer as and when required

Initial move to start position
X? - X axis start position.
Y? - Y axis start position.
Z? - Z axis start position.
H? - Fixture offset .
S? - Spindle speed.
M? - Miscellaneous M function for spindle start.

Programming Terms

Colon Block (:)

: T1 M6

A colon is required at the start of a program block where the programmer designates the easiest starting position for every new machining process. Any starting procedure on any line of the program without a colon will create an alarm.

Note:

A colon also acts as a "Data Reset" and will return the control back to its default values including in some cases the cancellation of fixture offsets, tool offsets and co-ordinate transformation cycles (i.e. rotation).

Sequence Number:

N 0002

A numerical number follows the letter "N" at the program line beginning. "N" numbers are used as a search facility to enable simple program editing. "N" numbers have no effect on the program itself but does require memory. Any line number required on a colon block line must omit the "N" character (i.e. 1 :T1 M6). Sequence numbers can be added to the program after completion by the use of the sequence number editor in the "Edit" mode of the control.

The sequence number can be allocated as the following examples:

Example 1: (sequence numbering at each tool change line)

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

Example 2: (all line numbering)

N100 **T1 M6**
N101 (Program for Tool 1)
N200 **T2 M6**
N201 (Program for Tool 2)
N300 **T3 M6**
N301 (Program for Tool 3)

Programming Terms (cont.)

Block:

A block is the minimum amount of **“WORD”** commands necessary for the machine to perform their operations.

A block takes up one line when written on a program sheet. Each line is called a block.

1 : T1 M6;	*The second “Block”*
N2 G0 G90 G40 G71 G17 G94	*The third “Block”*
N3 X? Y? Z? H? S? M3	*The fourth “Block”*
N4 Z?	*The fifth “Block”*

Word:

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

N1	G0	X0	Y0	Z0
word	*word*	*word*	*word*	*word*

Address:

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

N1 G0 X0 Y0 Z0

Numerical Data:

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

N1 G0 X0 Y0 Z0

Linear Interpolation

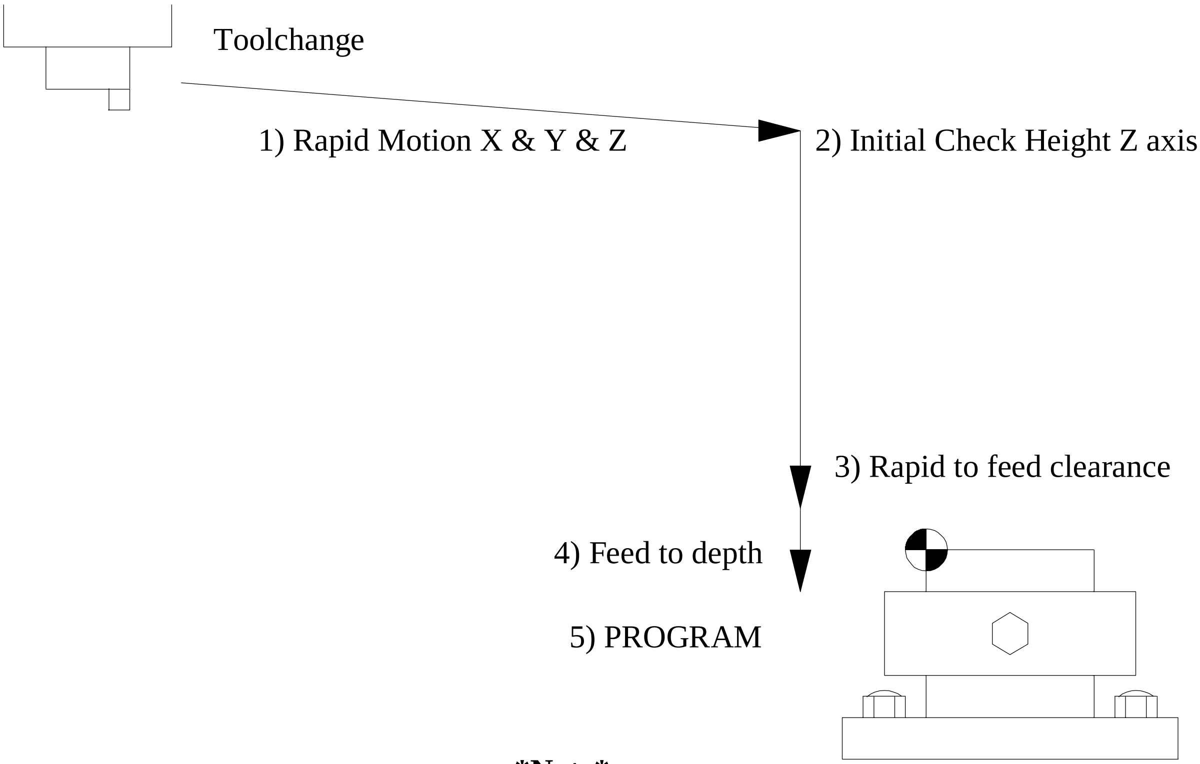
The axis of the machine will move at either “Rapid” or “Feed” traverse rates. The rapid rates vary on all machine types.

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

When programming a straight line “Rapid” (G00), with 2 axis (i.e X & Y), both axis will arrive at their programmed destination at the same time, irrespective of their length of motion, creating an angled motion. If one axis has to travel further than the other axis then this axis will move at an automatically calculated slower feed than that of machine rapid to allow both axis to arrive together.

As standard maximum of 3 axis can be programmed in one BLOCK

Initial Start of Program/Tool



Note

1) The *"Rapid Motion"* towards the *"Initial Check Height"* will contain X & Y & Z axis motion together with the required *"Workpiece Co-ordinate System"* code, spindle speed (S?) and the required M code to start the spindle (M3, M4, M13, or M14).

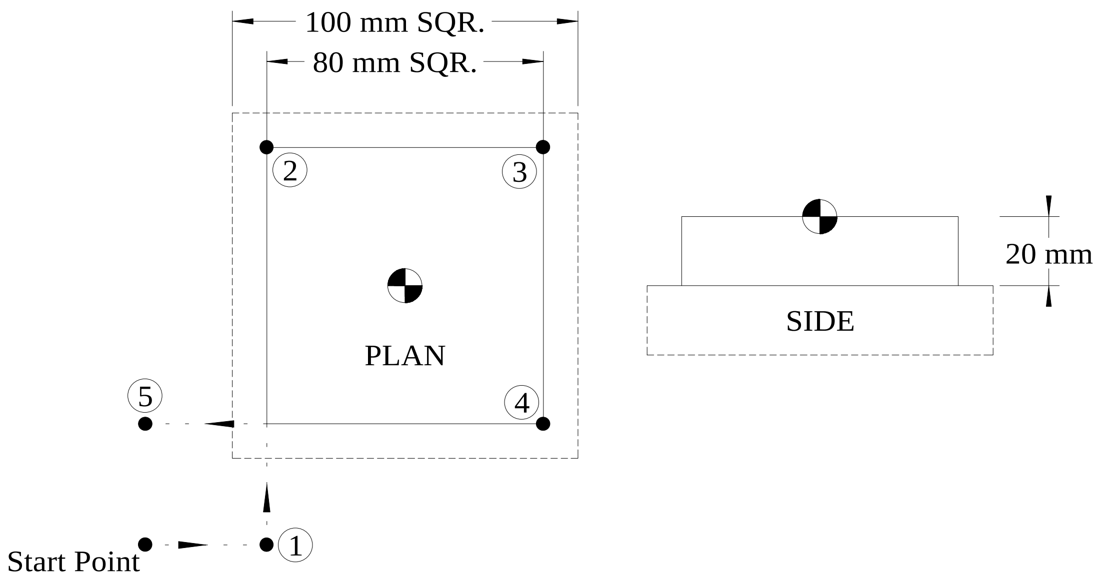
Release 2.32 and lower

If using *"Fixture Offsets - H word"* &or *"NC Program Offsets - D word"* then a three axis rapid motion must be specified on ANY program line containing a "H" &or "D" offset word.

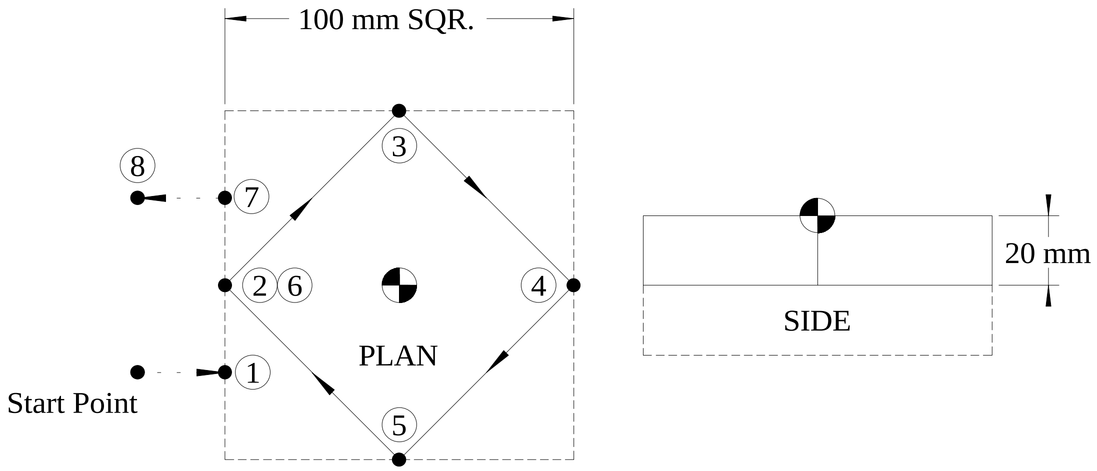
2) The *"Rapid Motion"* towards the *"Feed Clearance"* will contain a Z axis motion and does not require any additional information with regard tool length setting. The tool change line automatically captures all relevant tool data from the *"Tool Library"*.
3) At the end of the program for the tool in the spindle, an end of tool safety line is programmed to enable safe retraction from the part.

: T? M6 ; Tool change line
G0 G90 G40 G71 G17 G94 ; Safety default line
X? Y? Z100 H? S? M3 ;First move setting Work co-ordinate system & Speed R.P.M.
Z5 ; Feed clearance above component
("PROGRAM")
G0 G90 Z100 M1 ; End of tool safety line

Point to Point - example 1
(no compensation)



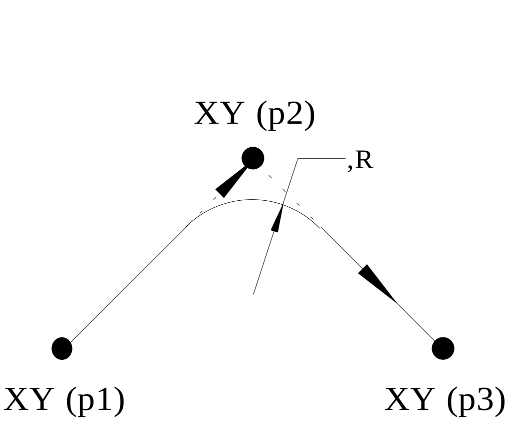
Point to Point - example 2
(no compensation)



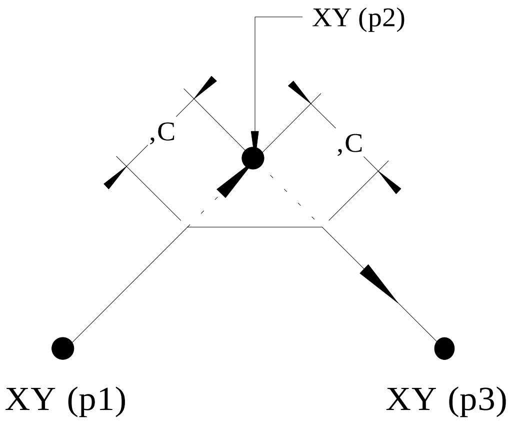
Corner Radius/Chamfer

Corner Radius & Uniform Chamfers

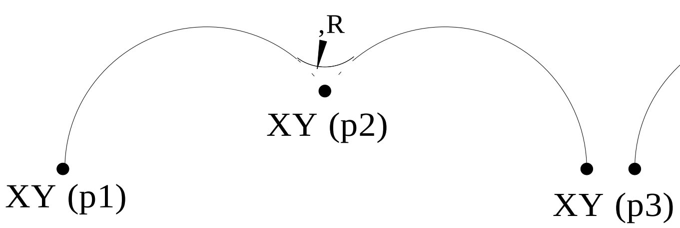
Blend chamfers and Blend Radii can be created within a program with the aid of short word addresses (,C for chamfers and ,R for radius).
The word address ,C & ,R are **Non-Modal** values and are added to the program line on the approach to the **known intersection programmed point.**



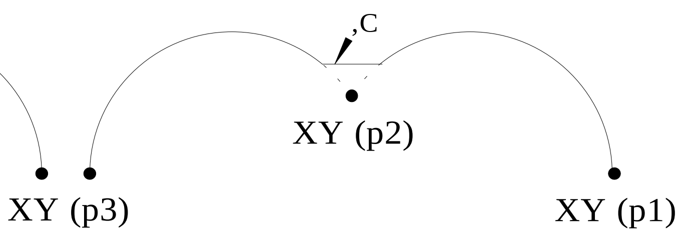
(p1) G1 X? Y?
(p2) X? Y? ,R?
(p3) X? Y?



(p1) G1 X? Y?
(p2) X? Y? ,C?
(p3) X? Y?



(p1) X? Y?
(p2) G2 X? Y? P? ,R?
(p3) X? Y? R?

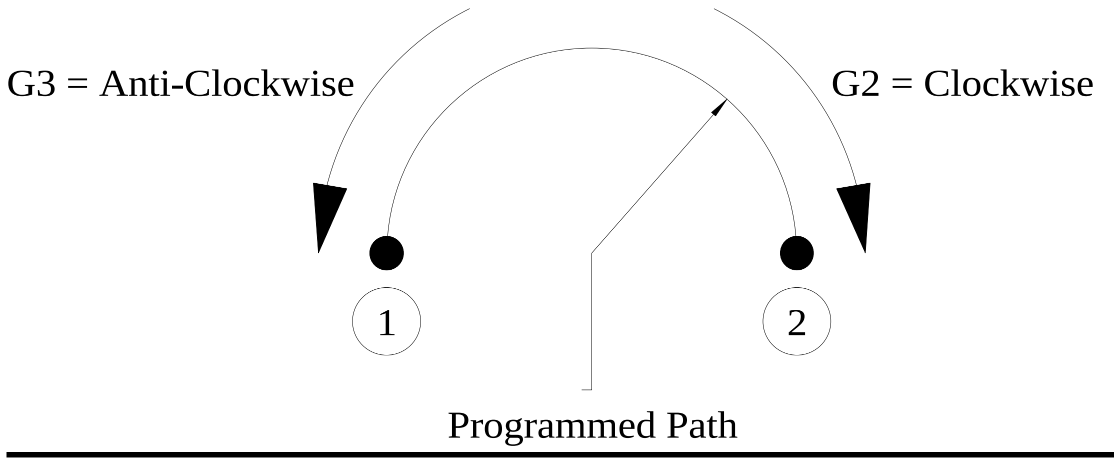


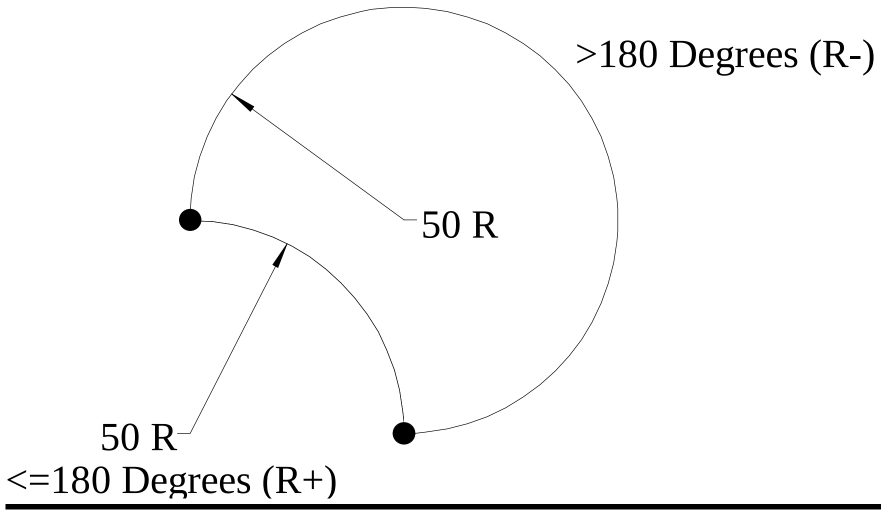
(p1) X? Y?
(p2) G3 X? Y? P? ,C?
(p3) X? Y? R?

CHAPTER 5

Circular Programmed Movements

1) ARCS WITH A KNOWN RADIUS





Note:

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

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

To specify the required arc, the following applies:

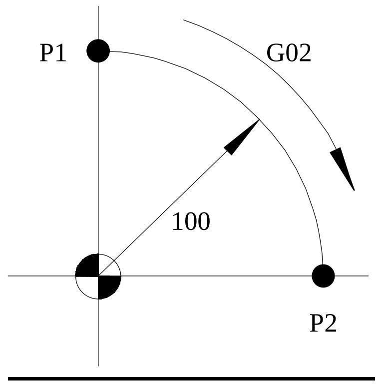
0.001° 180.000° = P+
180.001° 359.999° = P-

The "P" word can only be used with open Arcs up to 359.999°

Arc Interpolation (G02/G03 - P?)

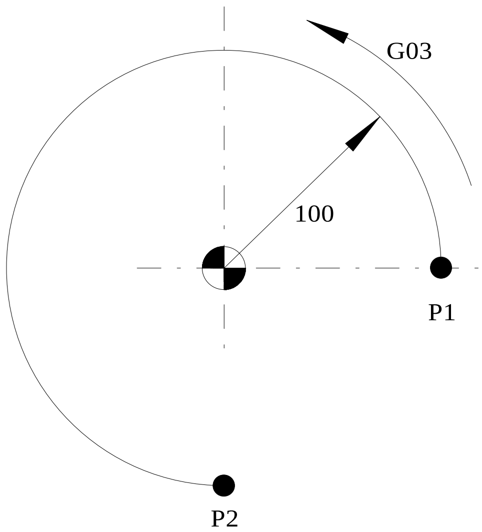
The information required to move in an arc involves the following “Word” addresses:

G17 Example:



G02 X100 Y0 P100

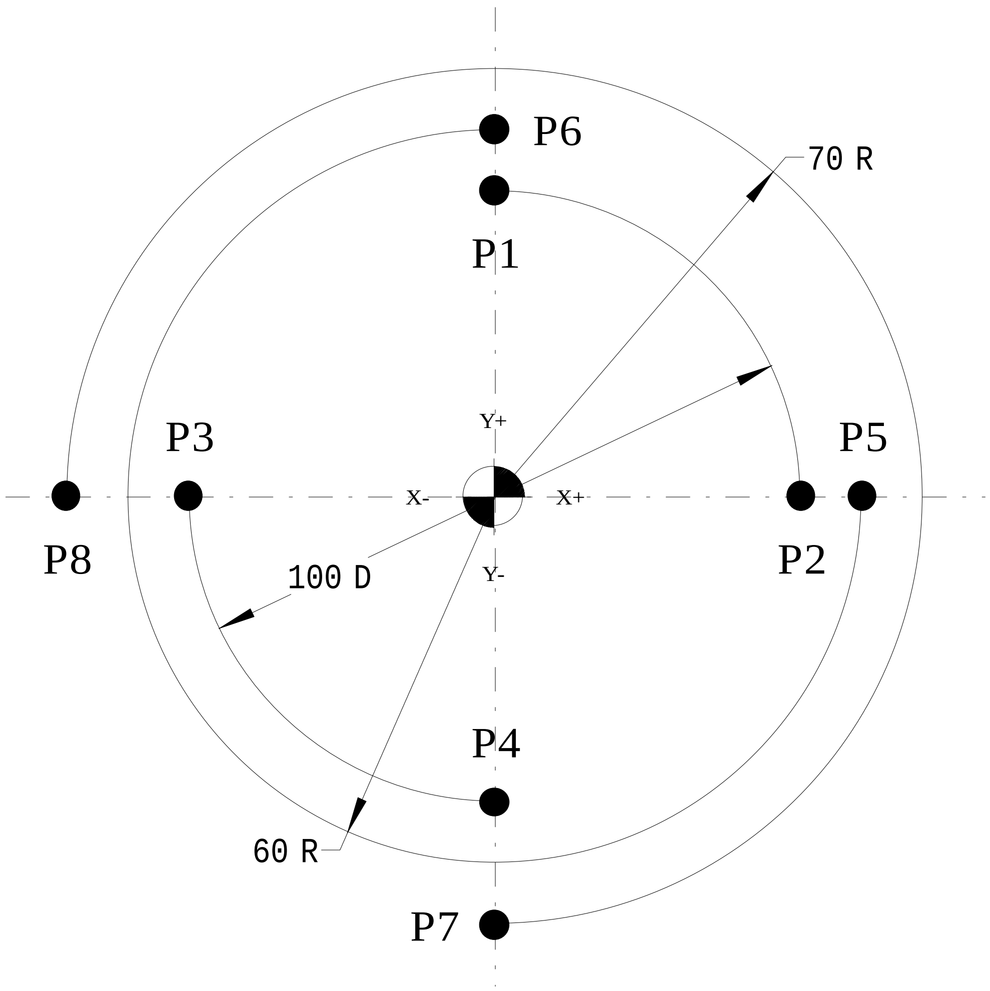
Where:
G02 = Modal Clockwise Feed Motion.
X100 = Radius end point in X axis.
Y0 = Radius end point in Y axis.
P100 = Radius of Arc (0.001° 180.000° arc motion = P+)



G03 X0 Y-100 P-100

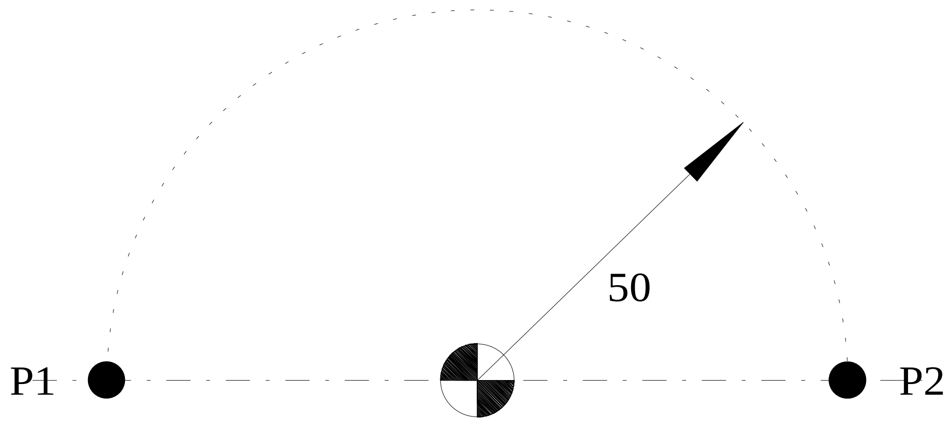
Where:
G03 = Modal Counter Clockwise Feed Motion.
X0 = Radius end point in X axis.
Y-100 = Radius end point in Y axis.
P-100 = Radius of Arc (180.001° 359.999° arc motion = P-)

Circular Programmed Movements – e.g. 1
(Absolute)



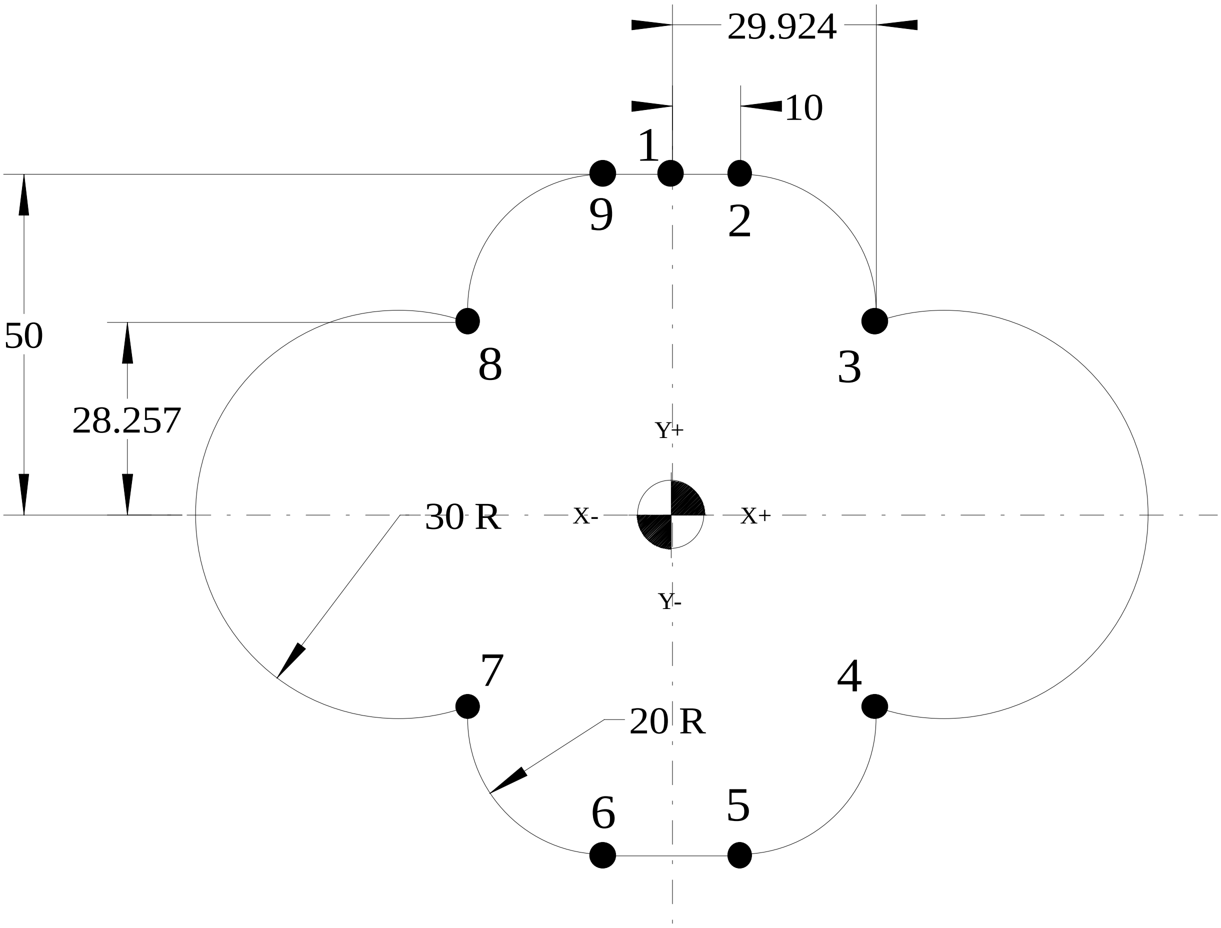
1	(Absolute Rapid XY to point 1)
	(Clockwise to point 2)
2	(Absolute Rapid XY to point 3)
	(Anti-Clockwise to point 4)
3	(Absolute Rapid XY to point 5)
	(Clockwise to point 6)
4	(Absolute Rapid XY to point 7)
	(Anti-Clockwise to point 8)

Circular Programmed Movements – e.g. 2
(Absolute)



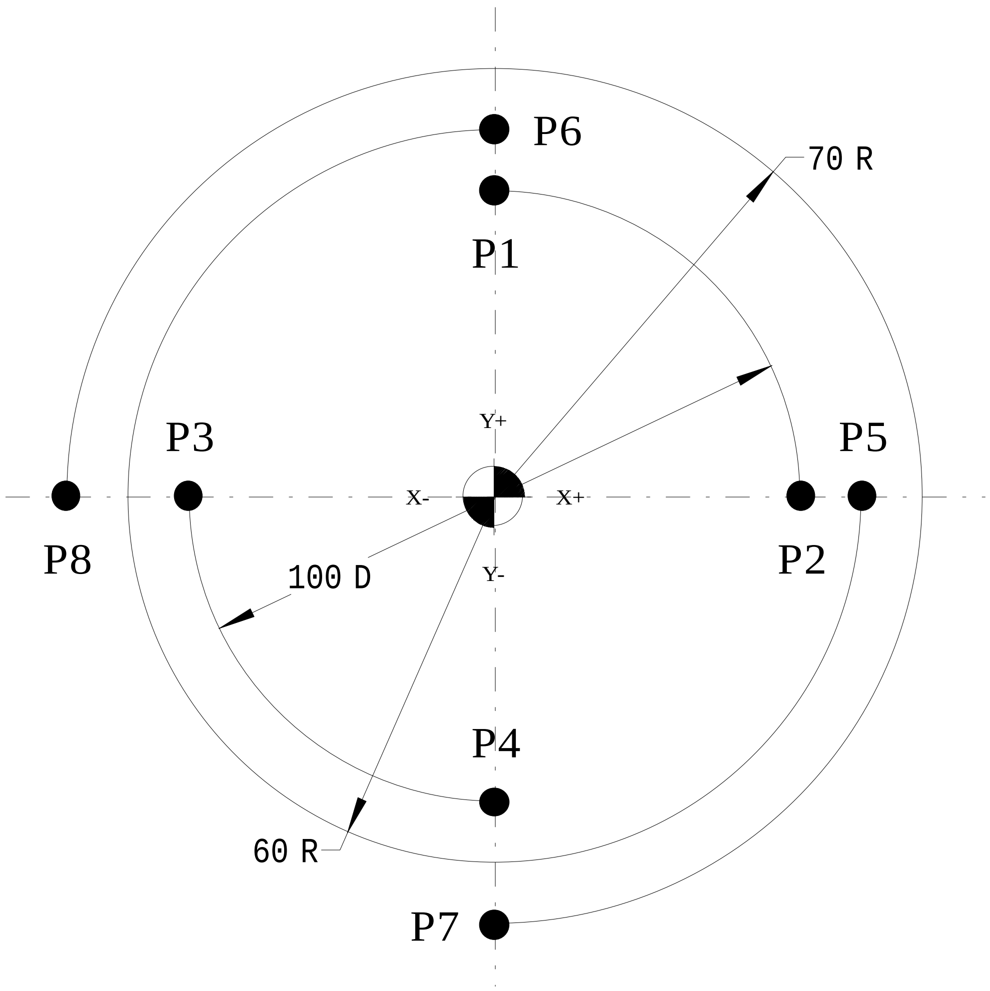
1	(Absolute Rapid XY to point 1)
	(Clockwise to point 2)
2	(Absolute Rapid XY to point 2)
	(Anti-Clockwise to point 1)

Circular Programmed Movements – e.g. 3



	1 G0 X0 Y50
1	2
2	3
3	4
4	5
5	6
6	7
7	8
8	9
9	1

Circular Programmed Movements – e.g. 4
(Incremental)



1	(Absolute Rapid XY to point 1)
	(Incremental Clockwise to point 2)
2	(Absolute Rapid XY to point 3)
	(Incremental Anti-Clockwise to point 4)
3	(Absolute Rapid XY to point 5)
	(Incremental Clockwise to point 6)
4	(Absolute Rapid XY to point 7)
	(Incremental Anti-Clockwise to point 8)

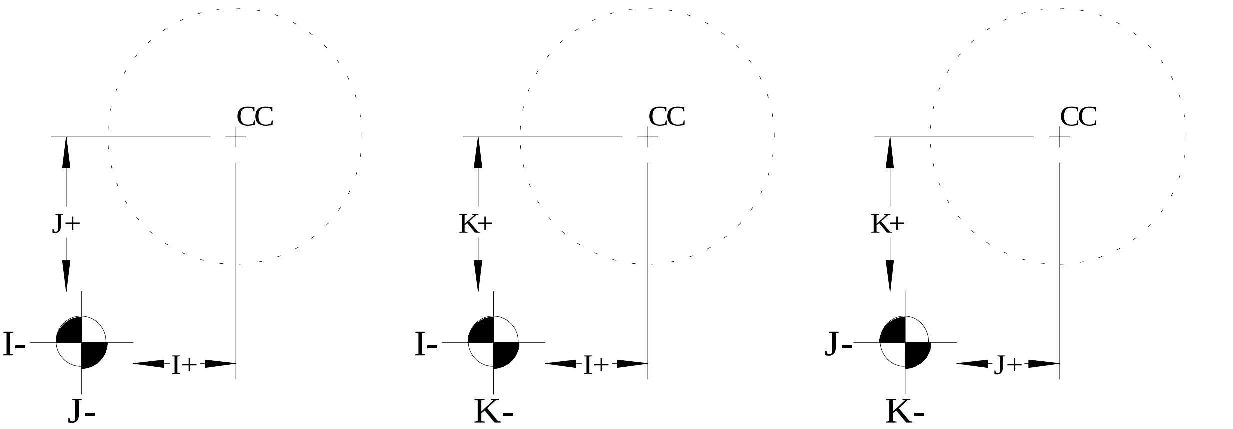
*__Note:__
All Incremental values are taken “FROM” the programmed start point “TO” the programmed endpoint.

Circular Programmed Movements

2) ARCS/FULL CIRCLES USING THE CIRCLE CENTRE

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

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



G17 Plan View (Z-)	G18 Front View (Y-)	G19 Side View (X-)
I = X axis circle centre	I = X axis circle centre	J = Y axis circle centre
J = Y axis circle centre	K = Z axis circle centre	K = Z axis circle centre

All the planes are viewed through the negative direction of the plane axis i.e. with the XY plane the view is through the Z- direction.

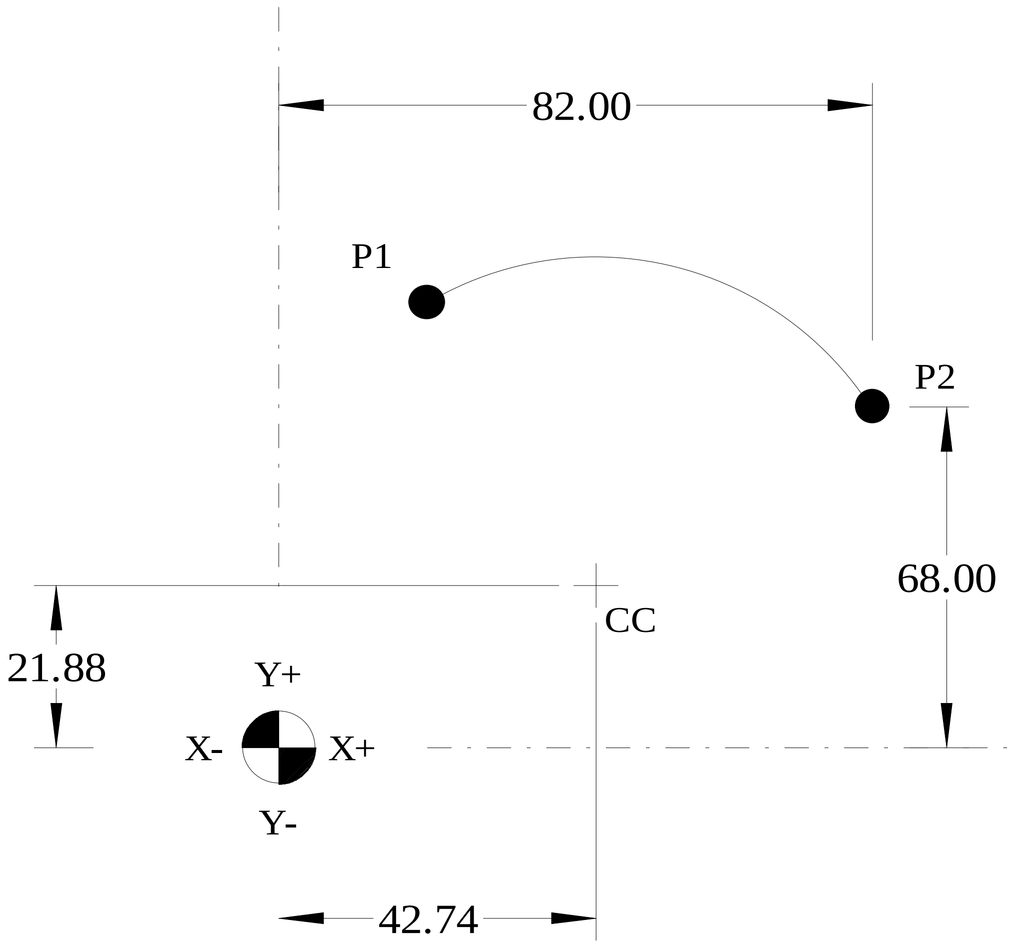
Using G18 plane the plane is viewed through the Y- direction, which is from the column towards the operator. Using ISO standards the codes for G2/G3 remain the same but from an operator view the direction will look the opposite
(G2 will look as G3 and G3 will look as G2)

Note
The axis information I, J & K are absolute values taken *"FROM"* the current datum *"TO"* the arc/circle centre position if programming in G90 co-ordinates.

Circular Programmed Movements

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

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

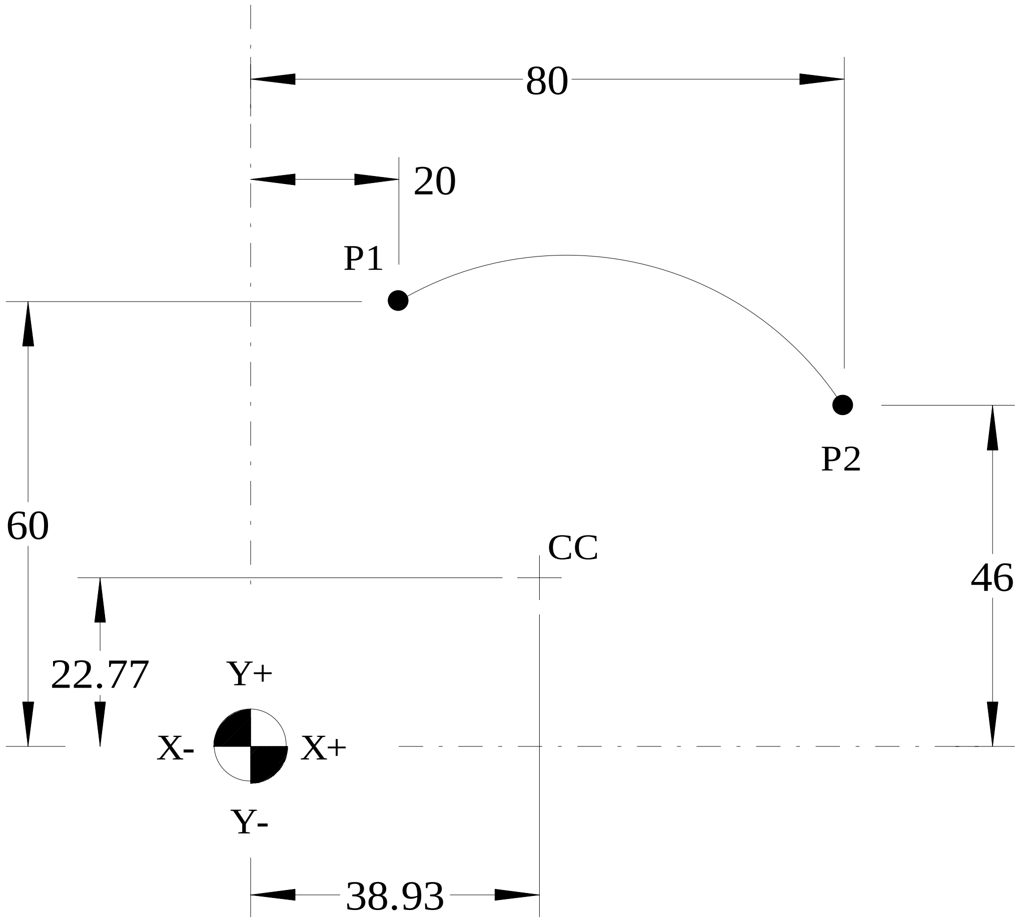


G02 X82 Y68 I42.74 J21.88 F?

Where:
G02 = Modal Clockwise Feed Motion.
X? = Arc end point in X axis.
Y? = Arc end point in Y axis.
I? = X axis position to Circle centre.
J? = Y axis position to Circle centre.
F? = Feedrate

Circular Programmed Movements – e.g. 5

X, Y, I & J (Absolute)



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

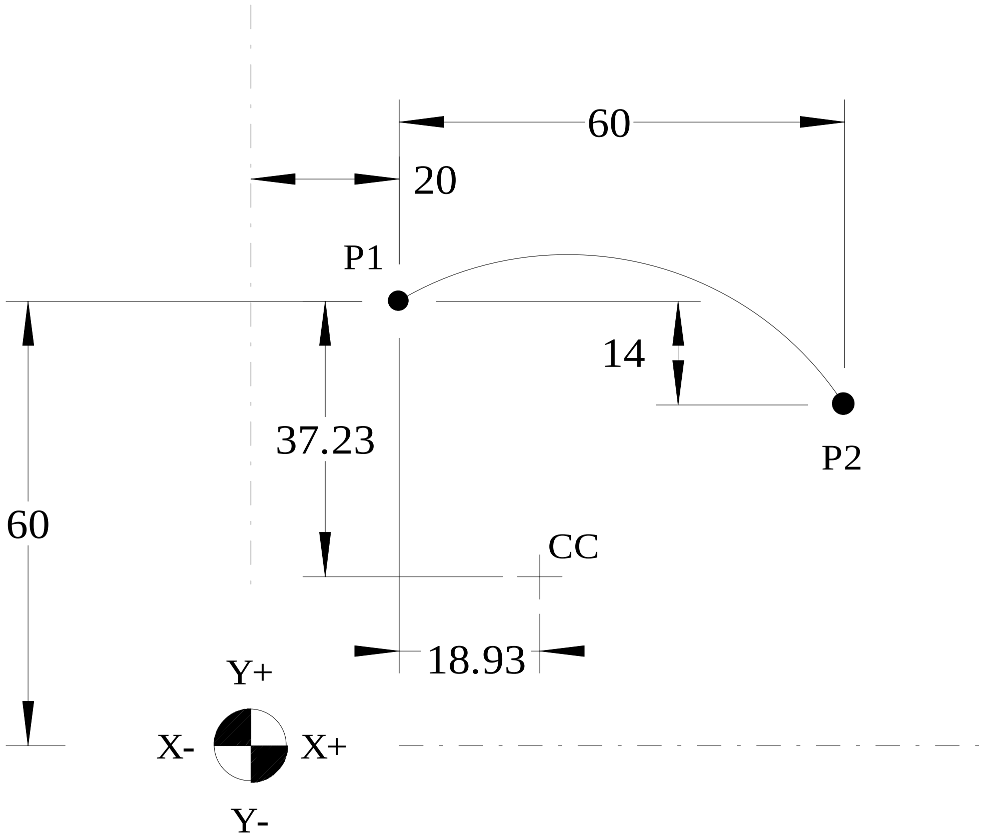
****

	(Absolute Rapid XY to point 2)
	(Anti-clockwise to point 1)

"I" = X axis circle centre position
"J" = Y axis circle centre position

Circular Programmed Movements – e.g. 6

X, Y, I & J (Incremental)

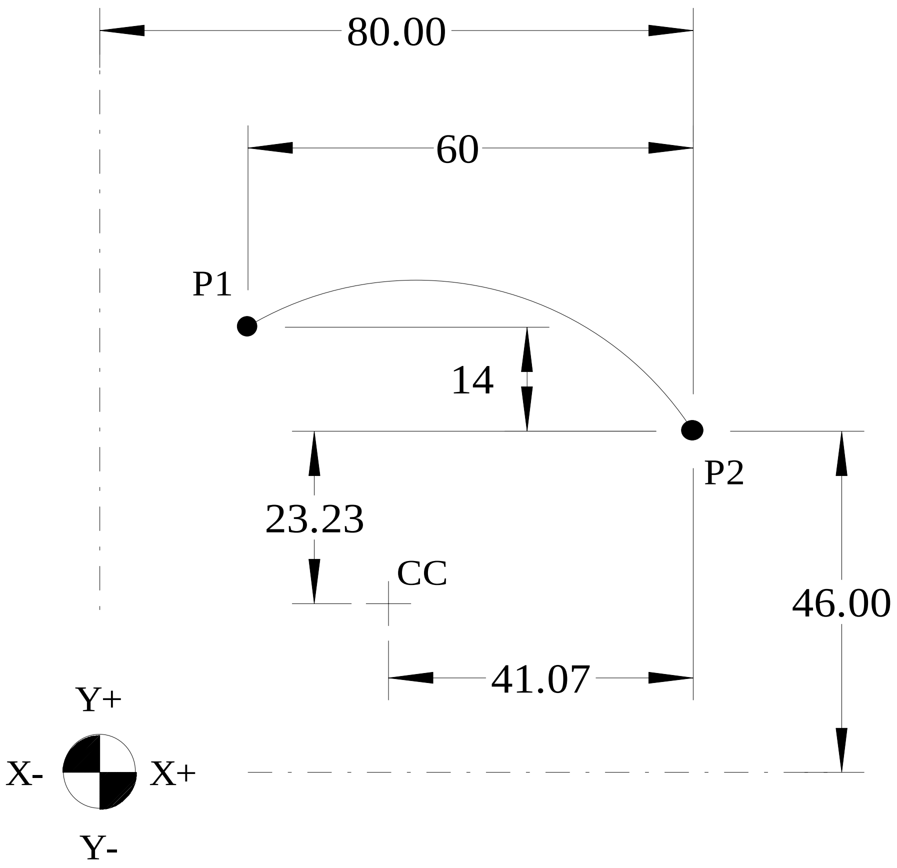


(Absolute Rapid XY to point 1)
(Incremental clockwise to point 2)

"I" = X axis circle centre position
"J" = Y axis circle centre position

Circular Programmed Movements – e.g. 7

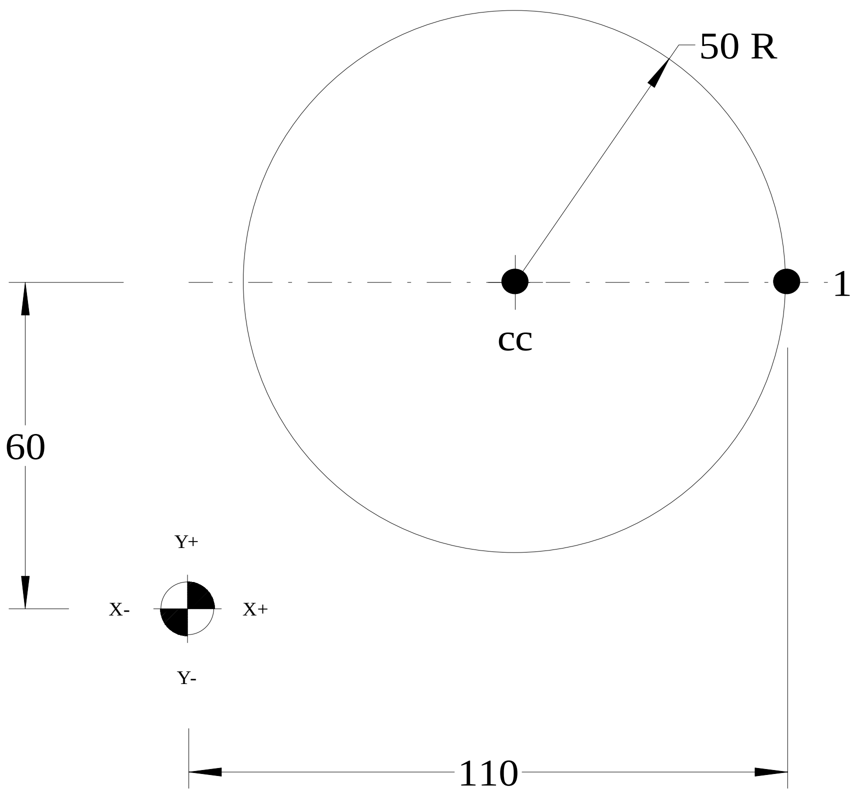
X & Y, I & J (Incremental)



(Absolute Rapid XY to point 2)
(Incremental anticlockwise to point 1)

"I" = X axis circle centre position
"J" = Y axis circle centre position

Full Circular Movements – e.g.8



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

Incremental

Note:

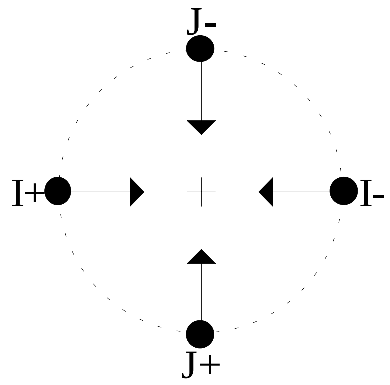
The end points and circle centre positions are taken from the "Start point"

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

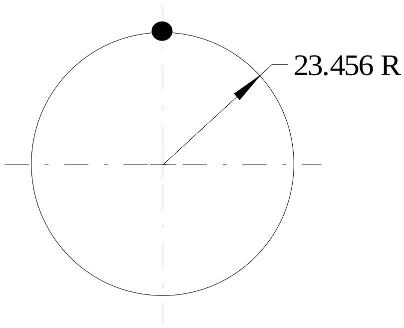
Full Circular Movements – e.g.9

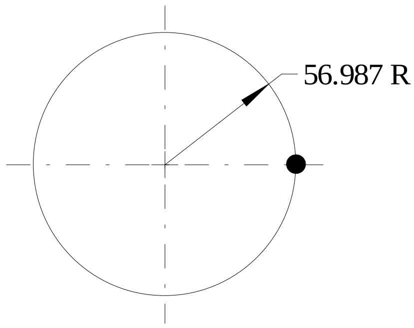
By selecting a pole point of a circle (12, 3, 6 or 9 o'clock position) and using an "Incremental line of program" to create a full circle, all values on this line of program will have a zero value except for the I or J axis on the appropriate pole axis which will represent the radius to be produced:

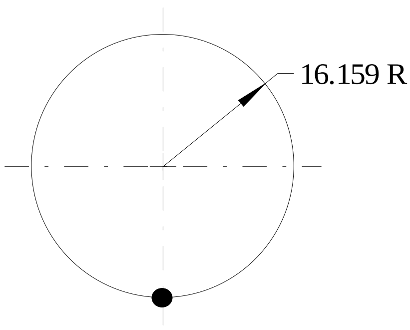
i.e. G91 G2 X0 Y0 I0 J-50

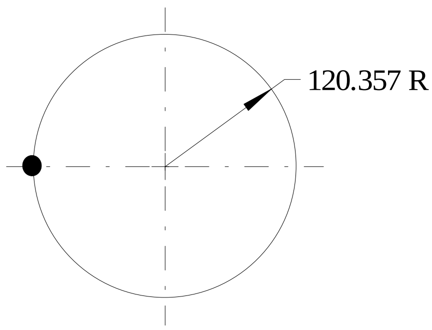


Create the line of circle program for each of the following quadrant points in the diagrams below using "Clockwise".









CHAPTER 6

Programmable Cutter Radius Compensation

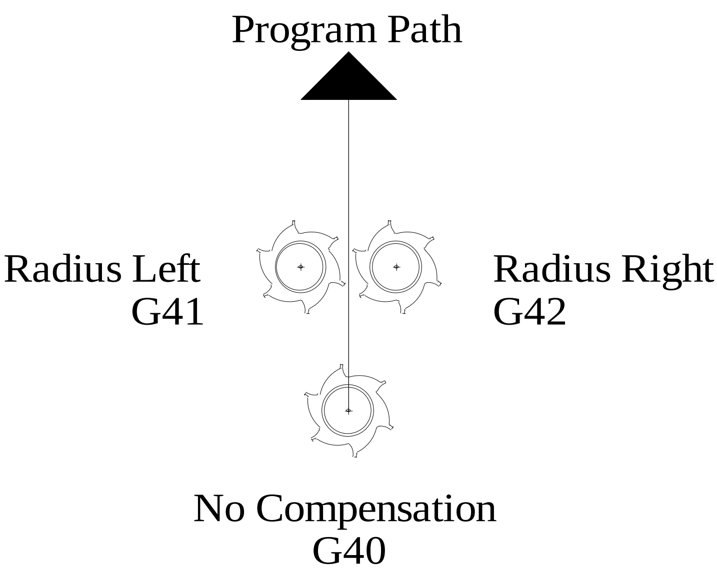
Programs can be created to create a part within tolerance dimensions. Once the program has been created and the part machined, any adjustments can be made by adjusting the tool radius or tool length, which are stored in the tool library.

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

“Job Path” compensation automatically adds half the diameter of the tool (radius) which is stored in the tool library **“Diameter Offset”** table, during program running and so makes programming very simple. “Cutter path” compensation still uses information taken from the tool offset table but this information is usually an adjustment value of the original programmed radius to make adjustments to the finished part size since the radius is already added to the program dimensions. Adding or subtracting the tool radius to every dimension can make creating the program very long and difficult.

The program line of information contains a modal “G” code (G41 / G42) which determines the offset side of the program path, and an axis motion:

i.e. G41 X100



Programmed Radius Compensation has 3 modes:

G40 – No compensation so the path is directly over the program path.

G41 – Compensation is to the left of the programmed path (Climb Milling) *Best method

G42 – Compensation is to the right of the programmed path (Conventional Milling)

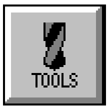
Making machining adjustments

To leave material on the contour or pocket “ADD” the offset value to the value in the tool library “Diameter Offset” column.

To remove material on the contour or pocket “SUBTRACT” the offset value from the tool library “Diameter Offset” column.

Tool Offset Page for Compensation

Tool diameter information relative to Cutter Compensation is stored in the “Tool Library” – “Diameter Offset” table.



Nom Diameter	Diam Offset
+0.0000	+0.0000
+0.0000	+0.0000
+0.0000	+0.0000
+0.0000	+0.0000
+0.0000	+0.0000
+0.0000	+0.0000
+0.0000	+0.0000
+0.0000	+0.0000
+0.0000	+0.0000
+0.0000	+0.0000

Hole tools Only	All Milling Cutters

The “Nom Diameter” column of the tool table is used to store the diameters of any hole making tool i.e. Drills, Taps, Boring tools etc.

The “Diam Offset” column of the tool table is used to store the diameters of all types of milling tools i.e. Endmills, Facemills etc.

The milling cycles of the A2100 control takes the “Nom Diameter” + “Dia Offset” as the overall diameter of the tool being used for machining.

Example

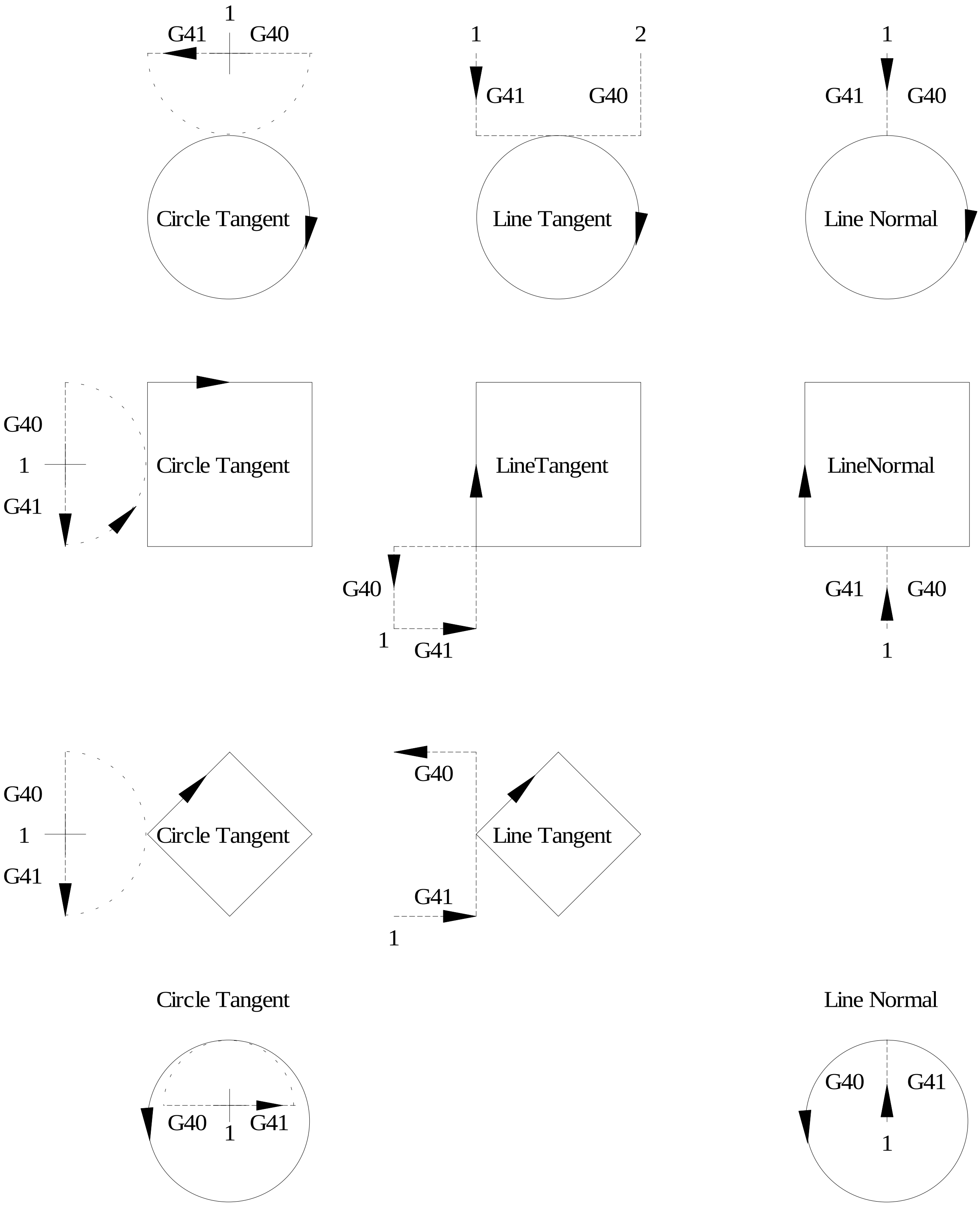
Type	Nom Diameter	Diam Offset
Tap	+10.0000	+0.0000
Rough End Mill	+0.0000	+10.0000
Unknown	+0.0000	+0.0000
Finish End Mill	+10.0000	+10.0000
Unknown	+0.0000	+0.0000

This is wrong

Compensation types

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

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



Simple Compensation Rules

Applying compensation

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

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

G0/G1 Z? G41 X? Y?	**G0/G1 Z? G41 Y? X?**

Cancelling compensation

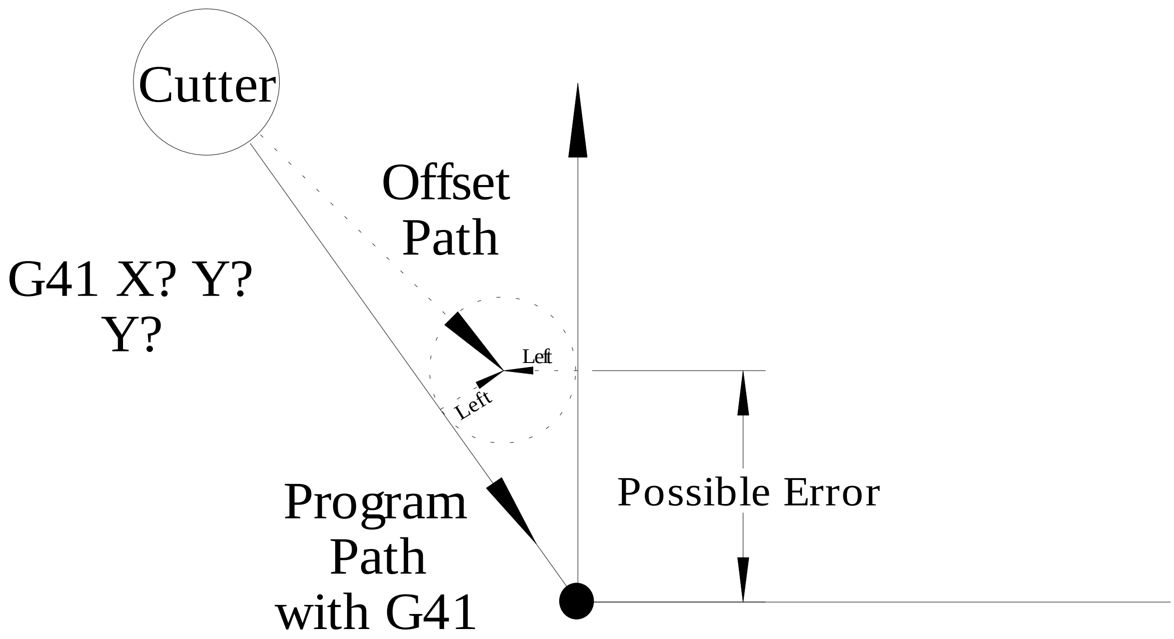
1) **"DO NOT MOVE"** "Z" until compensation has been cancelled.

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

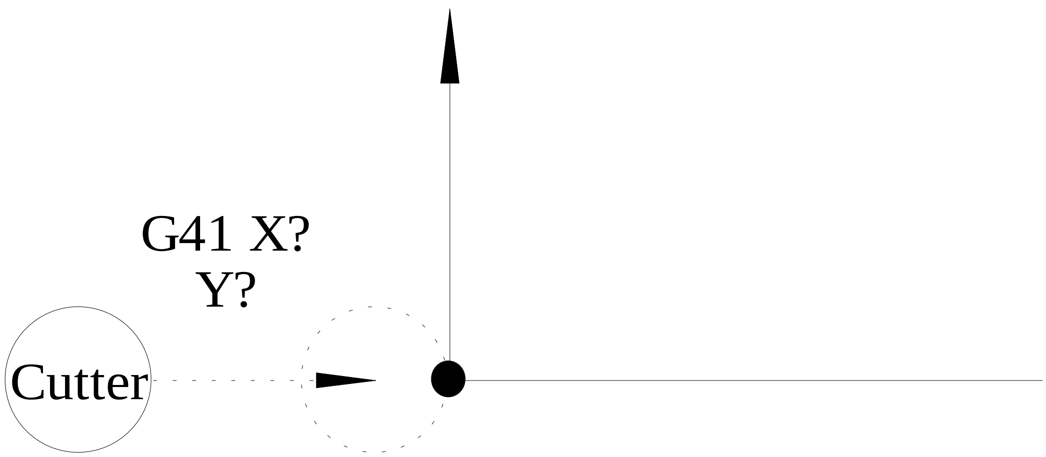
X? G40 Y? G0/G1 Z?	**Y? G40 X? G0/G1 Z?**

Possible Compensation Errors and Correction

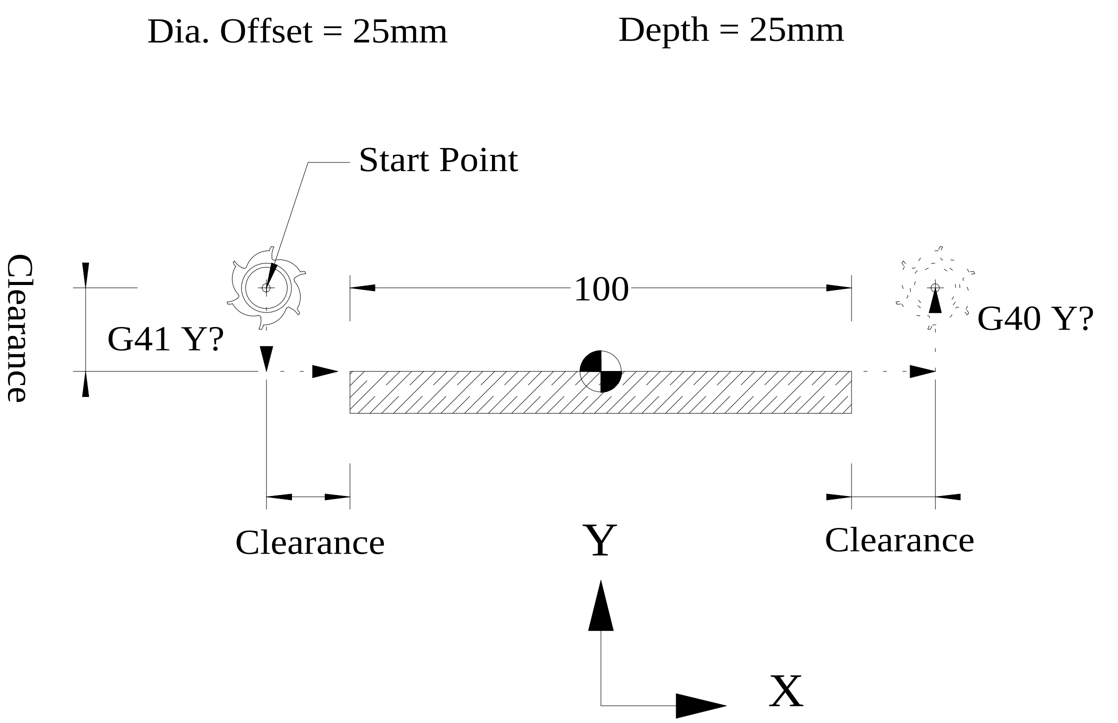
Incorrect



Correct

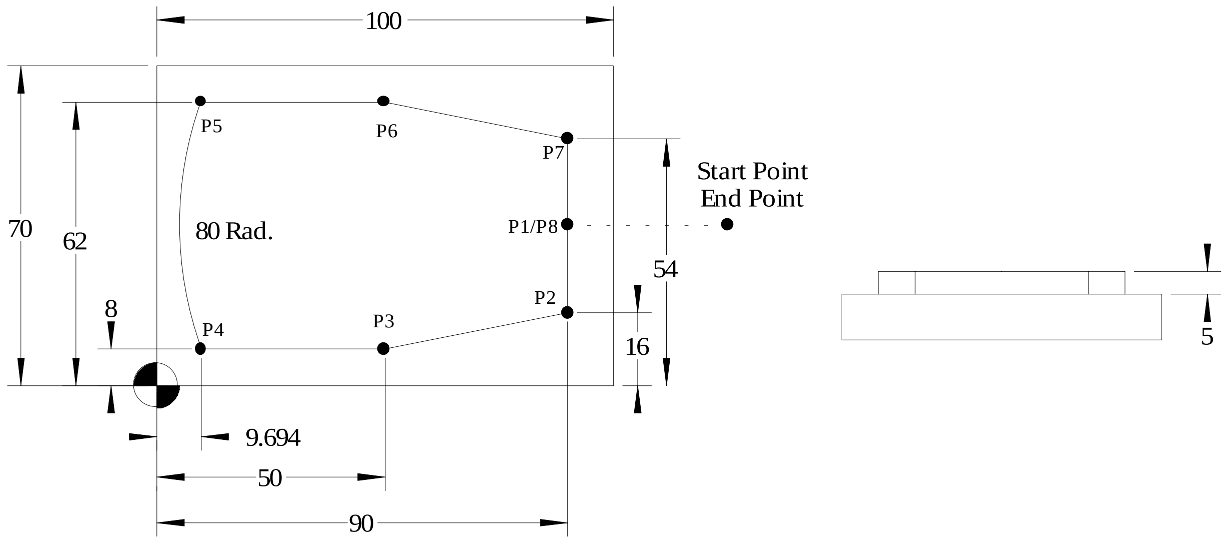


Compensation

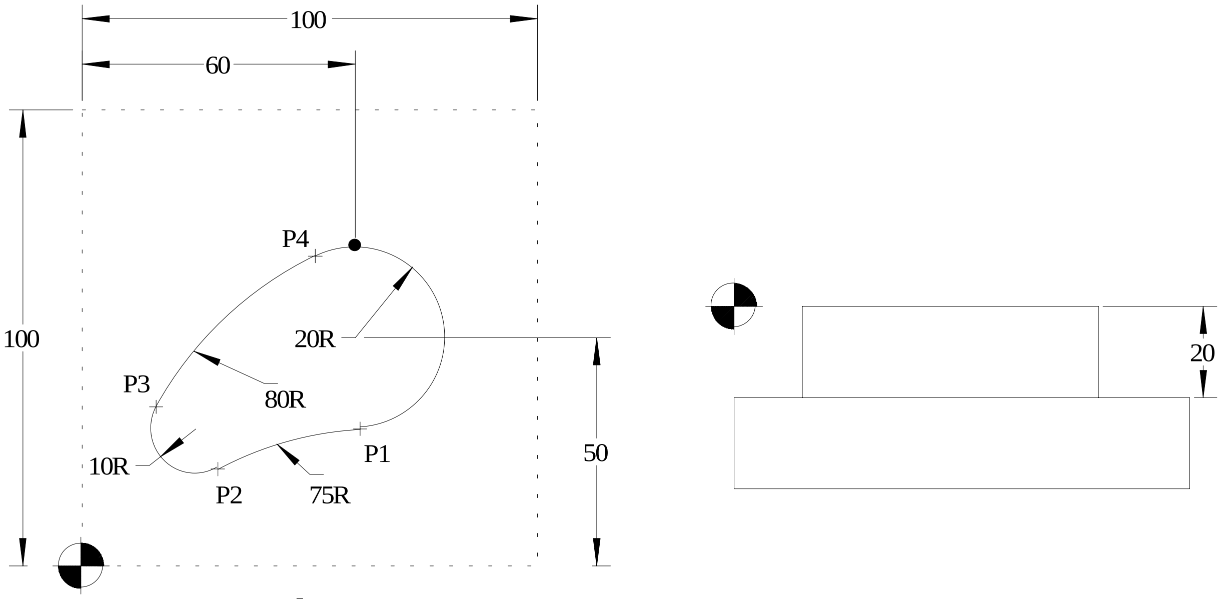


Program	***Command Position***	***Current Position***
G0 X-75 Y25 Z100	**X-75 Y25 Z100**	**X-75 Y25 Z100**
Z5	**X-75 Y25 Z5**	**X-75 Y25 Z5**
G1 Z-25 F?	**X-75 Y25 Z-25**	**X-75 Y25 Z-25**
G41 Y0	**X-75 Y0 Z-25**	**X-75 *_Y12.5_* Z-25**
X75	**X75 Y0 Z-25**	**X75 *_Y12.5_* Z-25**
G40 Y25	**X75 Y25 Z-25**	**X75 *_Y25_* Z-25**
G0 G90 Z100	**X75 Y25 Z100**	**X75 Y25 Z100**

Compensation e.g. 1



Compensation e.g. 2



Plan

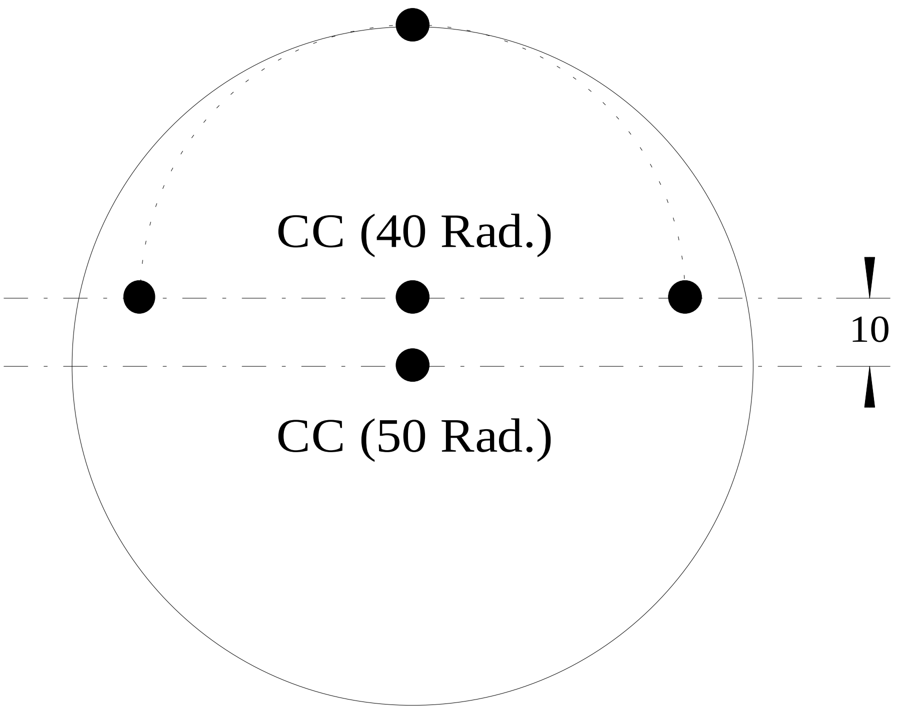
Front

	P1	P2	P3	P4
X	61.11	29.738	16.257	51.266
Y	30.031	21.194	34.854	67.992

Programmable Cutter Radius Compensation

Circle Tangent inside a full Circle

Main C/Bore = 50mm Radius



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

Note:

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

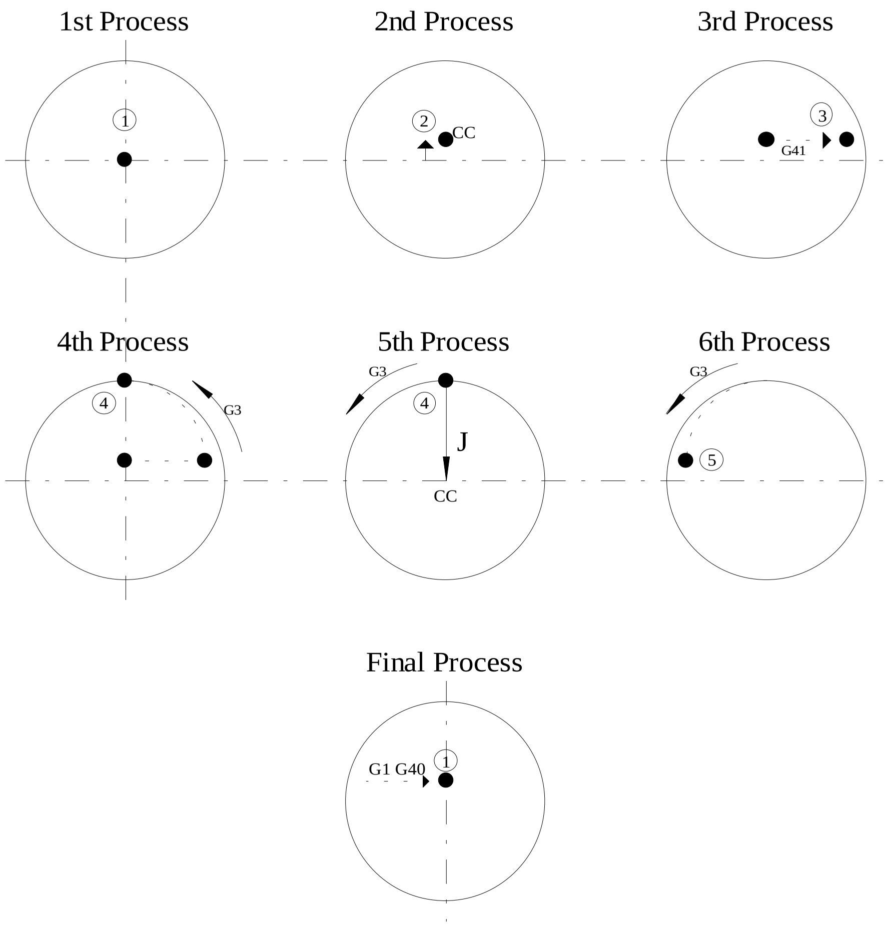
i.e.

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

As above graphical example
AR = 50
SR = 40
YD = 50 - 40 = 10

Programmable Cutter Radius Compensation

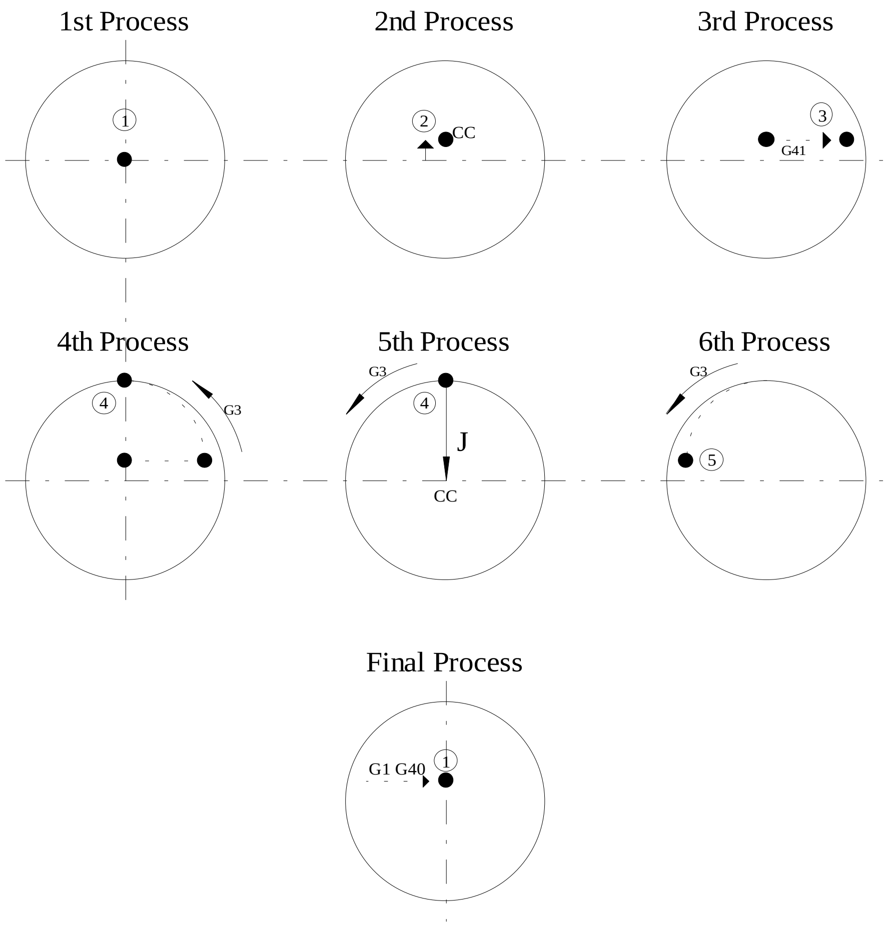
Circle Tangent



: T? M6	;Toolchange line - 25mm Endmill cutter
G0 G90 G40 G71 G17 G94	;Safety default line
X0 Y0 Z100 H? S? M3	;Absolute Start Point – Centre of Actual radius – position 1
Z5	;Rapid to a position above material
G1 Z-? F?	;Feed to required cut depth before compensation has been applied
G91 Y(YD)	;Incremental move to centre of Arc on/off as calculated – position 2
G41 X(SR) M8	;Move to point 3 - Apply compensation incrementally
G3 X-(SR) Y(SR) P(SR)	;Move to position 4 - Arc On.as SR Rad.
X0 Y0 I0 J-(AR)	;Move 360 Degrees back to position 4 by Radius of AR
X-(SR) Y-(SR) P(SR)	;Move to position 5 - Arc Off as SR Rad.
G1 G40 X(SR)	;Move to start position cancelling compensation
G0 G90 Z100	;Move to Absolute safe height above material after comp is cancelled
M30	;End program

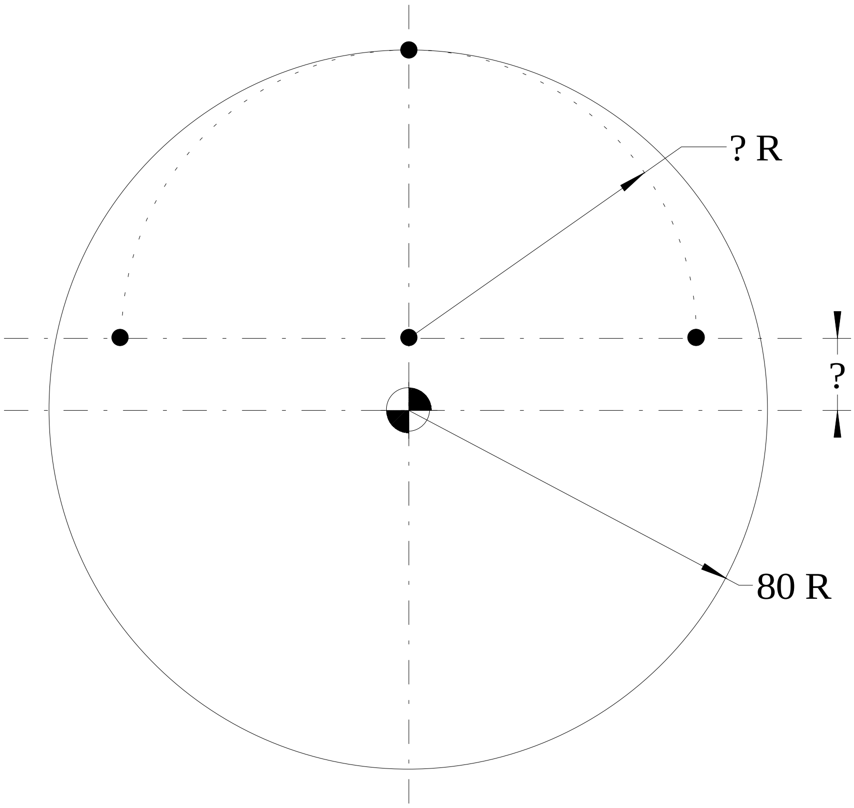
Programmable Cutter Radius Compensation

Circle Tangent



: T? M6 ;Toolchange line - 25mm Endmill cutter	
G0 G90 G40 G71 G17 G94 ;Safety default line	
X0 Y0 Z100 H? S? M3 ;Absolute Start Point – Centre of Actual radius – position 1	
Z5 ;Rapid to a position above material setting length offset	
G1 Z-? F? ;Feed to required cut depth before compensation has been applied	
G91 Y10 ;Incremental move to centre of Arc on/off as calculated – position 2	
G41 X40 M8 ;Move to point 3 - Apply compensation incrementally	
G3 X-40 Y40 P40 ;Move to position 4 - Arc On.as SR Rad.	
X0 Y0 I0 J-50 ;Move 360 Degrees back to position 4 by Radius of AR	
X-40 Y-40 P40 ;Move to position 5 - Arc Off as SR Rad	
G1 G40 X40 ;Move to start position cancelling compensation	
G0 G90 Z100 ;Move to Absolute safe height above material after comp is cancelled	
M30 ;End program	

Circle Tangent Compensation e.g. 1



CHAPTER 7

Helical Milling

The pitch **(K)** is programmed on the line of information requiring "X", "Y", and "Z" moves with circular programming.

NOTE
The actual "Z" movement must be a derivative of the "K" word to create a whole number as an angle i.e Z2 / K4 = ½ = 180 Deg..

i.e. If the Pitch = 4mm
One full circular movement = 360 Degrees
1 FULL PITCH
Z movement (4mm) & K = 4

G3 X? Y? Z4 K4 I? J?

Half of a full circular movement = 180 Degrees
1/2 FULL PITCH
Z movement (2mm) & K = 4

G3 X? Y? Z2 K4 P?

Quarter of a full circular movement = 90 Degrees
1/4 FULL PITCH
Z movement (1mm) & K = 4

G3 X? Y? Z1 K4 I? J?

5 full circular movements = 360 x 5 =1800 Degrees
5 FULL PITCHES
Z movement (4mm x 5 = 20mm) & K = 4

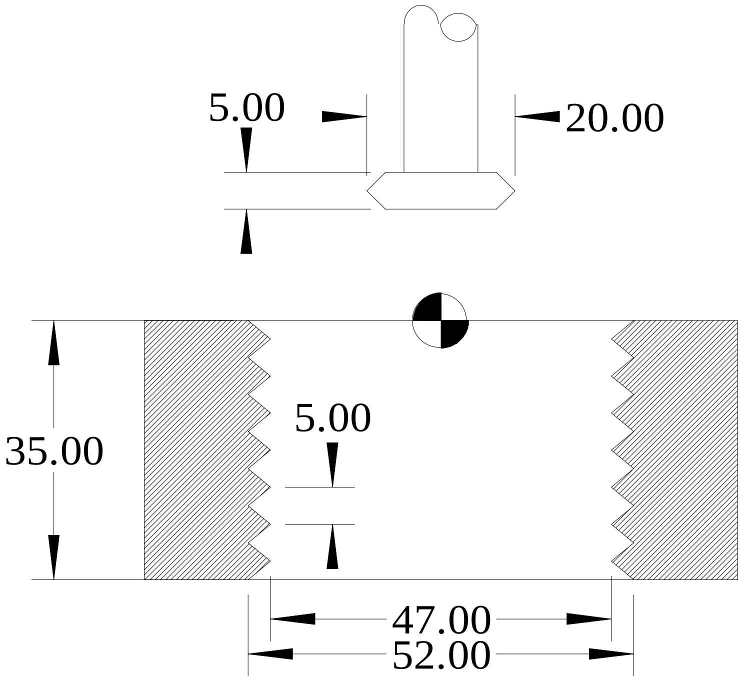
G3 X? Y? Z20 K4 I? J?

If the Pitch = 4mm and the arc movement is 90 Deg. the Z movement will be a 1/4 of 4mm = 1mm and is added or subtracted to the Z positioning movement at the lead on and the lead off the helix.
The Z value is programmed as the total thread depth and is then divided by the K word to give the total number of revolutions (or total pitches) eg 35 / 5 = 7 pitches or 7 x 360 Deg.

The line of program could look like this:
G2 X? Y? Z? I? J? K?
or
G2 X? Y? Z? P? K?

Where:
G? = Motion
XYZ = Endpoint
I,J,P = Circle centre or radius
K? = Lead of helix over 360 Degrees.

Helical Milling
"Single Point Tools"
Absolute Programming

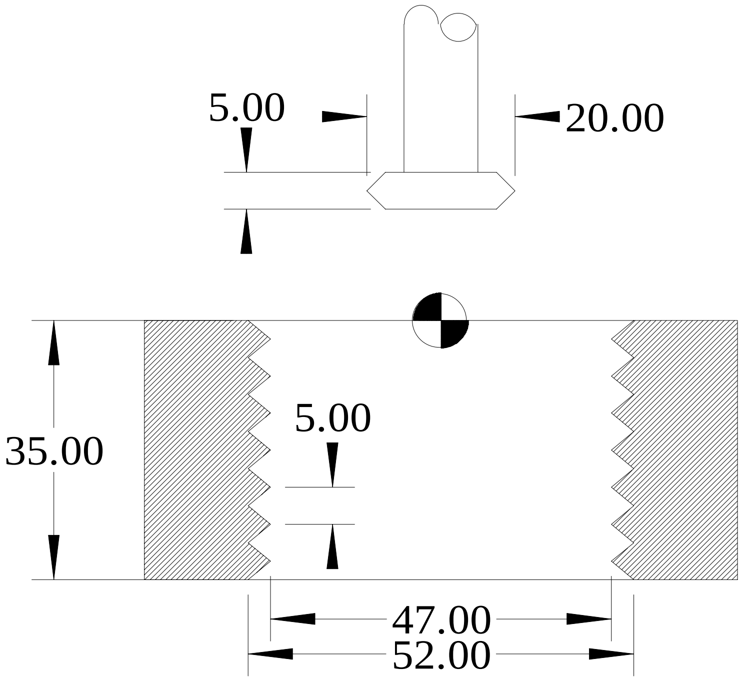


: T1 M6 ;Tool change line.
G0 G90 G40 G71 G17 G94 ;Safety Line
X0 Y0 Z100 H? S? M3 ;Move to centreline of bore
Z5 ;Move to bore top
G1 Z-38.75 F? ;Feed to depth + ¼ of pitch for arcing ON + nose width/2
Y5 ;Move to Arc On bore centre 26(AR) – 21(SR) = 5(YD)
G41 X21 ;Apply compensation as a straight line
G3 X0 Y26 Z-37.5 P21 K5 ;This line creates a ¼ arc + Z movement of a ¼ of pitch
X0 Y26 Z-2.5 I0 J0 K5 ;This line will create 7 pitches of 5mm
X-21 Y5 Z-1.25 P21 K5 ;This line creates a ¼ arc + Z movement of a ¼ of pitch
G1 G40 X0 ;Cancel Compensation as a straight line
G0 G90 Z100 ;Clear the workpiece
M30 ;End the Program

Helical Milling

“Single Point Tools”

Absolute & Incremental Programming



: T1 M6 ;Tool change line.

G0 G90 G40 G71 G17 G94 ;Safety Line

X0 Y0 Z100 H? S? M3 ;Move to centreline of bore

Z5 ;Move to bore top

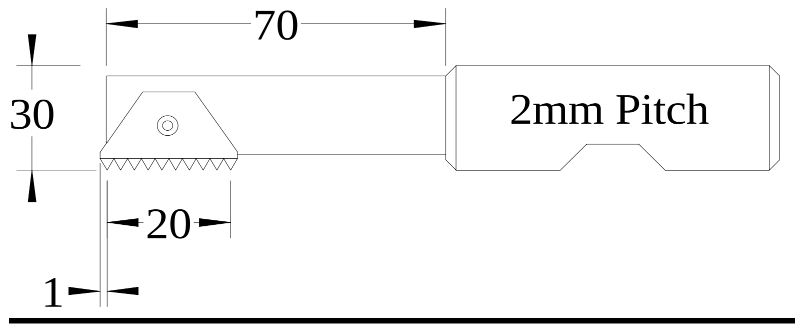
G1 Z-38.75 F? ;Feed to depth + ¼ of pitch for arcing ON + nose width/2

- *G91 Y5 ;Move to Arc On bore centre 26(AR) – 21(SR) = 5(YD)*
- *G41 X21 ;Apply compensation as a straight line*
- *G3 X-21 Y21 Z1.25 P21 K5 ;This line creates a ¼ arc + Z movement of a ¼ of pitch*
- *X0 Y0 Z35 I0 J-26 K5 ;This line will create 7 pitches of 5mm*
- *X-21 Y-21 Z1.25 P21 K5 ;This line creates a ¼ arc + Z movement of a ¼ of pitch*
- *G1 G40 X21 ;Cancel Compensation as a straight line*

G0 G90 Z100 ;Clear the workpiece

M30 ;End the Program

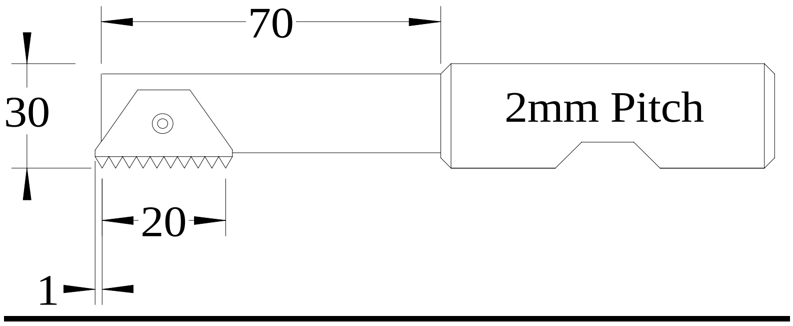
Helical Milling
"Multi toothed Tools"
Absolute Programming



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

: T1 M6 ;Tool change line
G0 G90 G40 G71 G17 G94 ;Safety Line.
X0 Y0 Z100 H? S? M3 ;Move to centreline of bore
Z5 ;Move to feed clearance
G1 Z-61.5 F? ;Feed to depth + ¼ of pitch for arcing ON + ½ tooth form width to tool end
Y10 ;Move to Arc On bore centre 30(AR) – 20(SR) = 10(YD)
G41 X20 ;Apply compensation as a straight line
G3 X0 Y30 Z-61 P20 K2 ;This line creates a ¼ arc + Z movement of a ¼ pitch
X0 Y30 Z-59 I0 J0 K2 ;This line will create 1 pitch.
X-20 Y10 Z-58.5 P20 K2 ;Creates a ¼ arc + Z movement of a ¼ of pitch
G1 G40 X0 ;Cancel Compensation as a straight line
G0 Z-41.5 ;Subtract edge length from 1st Z positioning move 61.5 – 20 = 41.5
G1 G41 X20 ;Apply compensation as a straight line
G3 X0 Y30 Z-41 P20 ;This line creates a ¼ arc + Z movement of a ¼ pitch
X0 Y30 Z-39 I0 J0 K2 ;This line will create 1 pitch.
X-20 Y10 Z-38.5 P20 K2 ;Creates a ¼ arc + Z movement of a ¼ of pitch
G1 G40 X0 ;Cancel Compensation as a straight line
G0 Z-21.5 ;Subtract edge length from 2nd Z positioning move 41.5 – 20 = 21.5
G41 X20 ;Apply compensation as a straight line
G3 X0 Y30 Z-21 P20 K2 ;This line creates a ¼ arc + Z movement of a ¼ pitch
X0 Y30 Z-19 I0 J0 K2 ;This line will create 1 pitch.
X-20 Y10 Z-18.5 P20 ;Creates a ¼ arc + Z movement of a ¼ of pitch
G1 G40 X0 ;Cancel Compensation as a straight line
G0 G90 Z100 ;Clear the workpiece
M30 ;End the Program.

Helical Milling
"Multi toothed Tools"
Absolute & Incremental Programming



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

: T1 M6 ;Tool change line
G0 G90 G40 G71 G17 G94 ;Safety Line.
X0 Y0 Z100 H? S? M3 ;Move to centreline of bore
Z5 ;Move to feed clearance
G1 Z-61.5 F? ;Feed to depth + ¼ of pitch for arcing ON + ½ tooth form width to tool end
Y10 ;Move to Arc On bore centre 30(AR) – 20(SR) = 10(YD)
• *G41 G91 X20 ;Apply compensation as a straight line*
• *G3 X-20 Y20 Z0.5 P20 K2 ;This line creates a ¼ arc + Z movement of a ¼ pitch*
• *X0 Y0 Z2 I0 J-30 K2 ;This line will create 1 pitch.*
• *X-20 Y-20 Z0.5 P20 K2 ;Creates a ¼ arc + Z movement of a ¼ of pitch*
• *G1 G40 X20 ;Cancel Compensation as a straight line*
G0 G90 Z-41.5 ;Subtract edge length from 1st Z positioning move 61.5 – 20 = 41.5
• *G41 G91 X20 ;Apply compensation as a straight line*
• *G3 X-20 Y20 Z0.5 P20 K2 ;This line creates a ¼ arc + Z movement of a ¼ pitch*
• *X0 Y0 Z2 I0 J-30 K2 ;This line will create 1 pitch.*
• *X-20 Y-20 Z0.5 P20 K2 ;Creates a ¼ arc + Z movement of a ¼ of pitch*
• *G1 G40 X20 ;Cancel Compensation as a straight line*
G0 G90 Z-21.5 ;Subtract edge length from 2nd Z positioning move 41.5 – 20 = 21.5
• *G41 G91 X20 ;Apply compensation as a straight line*
• *G3 X-20 Y20 Z0.5 P20 K2 ;This line creates a ¼ arc + Z movement of a ¼ pitch*
• *X0 Y0 Z2 I0 J-30 K2 ;This line will create 1 pitch.*
• *X-20 Y-20 Z0.5 P20 K2 ;Creates a ¼ arc + Z movement of a ¼ of pitch*
• *G1 G40 X20 ;Cancel Compensation as a straight line*
G0 G90 Z100 ;Clear the workpiece
M30 ;End the Program.

CHAPTER 8

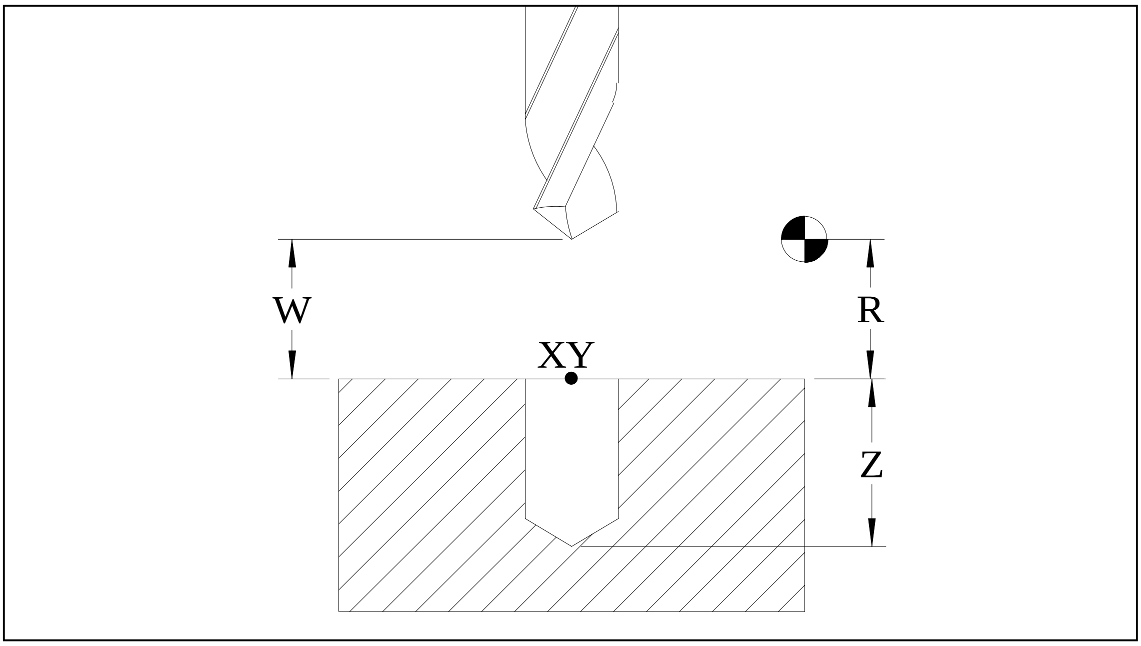
Hole Canned Cycles

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

G? X? Y? R? Z-? F? [W?] M?

Where:
G? = Cycle machining code.
X? & **Y?** = Absolute hole centre position.
R? = Hole surface position.
Z-? = Incremental depth of the hole from "R" position.
F? = Feedrate.
W? = Non-modal incremental retract amount above the "R" position at the end of hole machining.
M? = Coolant M8 or M27 (option).

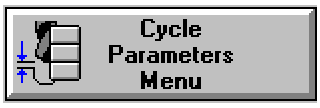
[ ] Optional input



Cancel Canned Cycle
G80

Group Interpolation Codes
The following G codes also are effective in cancelling any Hole Canned Cycle:
G0, G01, G02, G03

Cycle Parameters



Most of the hole canned cycles have the ability to perform other functions within the cycle itself to aid in manufacture. These can use facilities as dwell times, different feed factors, breakthrough clearances etc.

One of the most important parameter is the “Gauge Height Parameter” which tells the control how much clearance the tool point is to stop above the programmed “R” position (part surface) of the part.
The control uses the value contained in the “Programmable Column” as the stopping distance above the part surface.

Drilling Cycle Parameters	Program Reference	Base Value	Programmable Value
Gage Height Inch	GAGE_HT_INCH	+2.5400	+2.5400
Gage Height Metric	GAGE_HT_MM	+3.0000	+3.0000
Hole Depth Programming Mode	HOLE_DEPTH	+3	+3
G82 Finish Depth	G82_FIN_DPTH	+0.0000	+0.0000
G82 Finish Feed Factor	G82_FEED_FAC	+100	+100
G82 Dwell Time	G82_DWELL	+0.50	+0.50
G83 Retract Distance	G83_RET_DIST	+1.2500	+1.2500
G83 Short Retract Distance	G83_SHRT_RET	+10.0000	+10.0000
G83 Relief Amount	G83_RELIEF	+1.2500	+1.2500
G84 Dwell Time	G84_DWELL	+0.50	+0.50
G84 Chip Break Spindle Rev	G84_CHIP_BRK	+1	+1
G86 Bottom Retract Distance	G86_BOT_RET	+1.2500	+1.2500
G87 Dwell Time (seconds)	G87_DWELL	+0.50	+0.50
G87 Bottom Retract Distance	G87_BOT_RET	+0.1000	+1.2500
G87 Backbore Clearance	G87_BK_CLR	+50.0000	+50.0000
G88 Breakthrough Distance	G88_BRK_DIST	+10.0000	+10.0000
G89 Dwell Time (seconds)	G89_DWELL	+0.50	+0.50

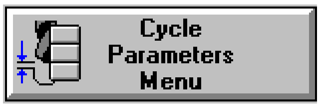
The parameter “Hole Depth Programming Mode” tells the control how to use the “Z” value on the line of program.

* Default

Value	Z Function	Includes Drill Point
0	Absolute depth	Yes
1	Incremental depth from R	Yes
2	Absolute depth	No
3 *	Incremental depth from R	No

The control only adds the drill tip point to the Z move if the **“Nom Diameter”, “Drill Tip Angle”** and **“Type”** of tool in the tool library are set with the correct data and the **“Hole Depth Programming Mode”** of the parameters is set to 0 or 1.

Cycle Parameters



Since the cycle parameters can control the final result of the machining operation, these need to be set by the operator before the cycle is machined. To avoid any operator errors in setting these parameters, it is possible to use the program itself to pre-set the required parameters to avoid any manual operator setting errors. This is achieved by the use of "System Registers" which allow the program to write to the appropriate table.

Parameter	System Register
Gage Height Inch	[$CYCLE_PARAMS(2)GAGE_HT_INCH] = ?
Gage Height Metric	[$CYCLE_PARAMS(2)GAGE_HT_MM] = ?
Hole Depth Program Mode	[$CYCLE_PARAMS(2)HOLE_DEPTH] = ?
G82 Finish Depth	[$CYCLE_PARAMS(2)G82_FIN_DPTH] = ?
G82 Finish Feed Factor	[$CYCLE_PARAMS(2)G82_FEED_FAC] = ?
G82 Dwell Time (seconds)	[$CYCLE_PARAMS(2)G82_DWELL] = ?
G83 Retract Distance	[$CYCLE_PARAMS(2)G83_RET_DIST] = ?
G83 Short Retract Distance	[$CYCLE_PARAMS(2)G83_SHRT_RET] = ?
G83 Relief Amount	[$CYCLE_PARAMS(2)G83_RELIEF] = ?
G84 Dwell Time (seconds)	[$CYCLE_PARAMS(2)G84_DWELL] = ?
G84 Chip Break Spindle Rev	[$CYCLE_PARAMS(2)G84_CHIP_BRK] = ?
G86 Bottom Retract Distance	[$CYCLE_PARAMS(2)G86_BOT_RET] = ?
G87 Dwell Time (seconds)	[$CYCLE_PARAMS(2)G87_DWELL] = ?
G87 Bottom Retract Distance	[$CYCLE_PARAMS(2)G87_BOT_RET] = ?
G87 Backbore Clearance	[$CYCLE_PARAMS(2)G87_BK_CLR] = ?
G88 Breakthrough Clearance	[$CYCLE_PARAMS(2)G88_BRK_DIST] = ?
G89 Dwell Time (seconds)	[$CYCLE_PARAMS(2)G89_DWELL] = ?

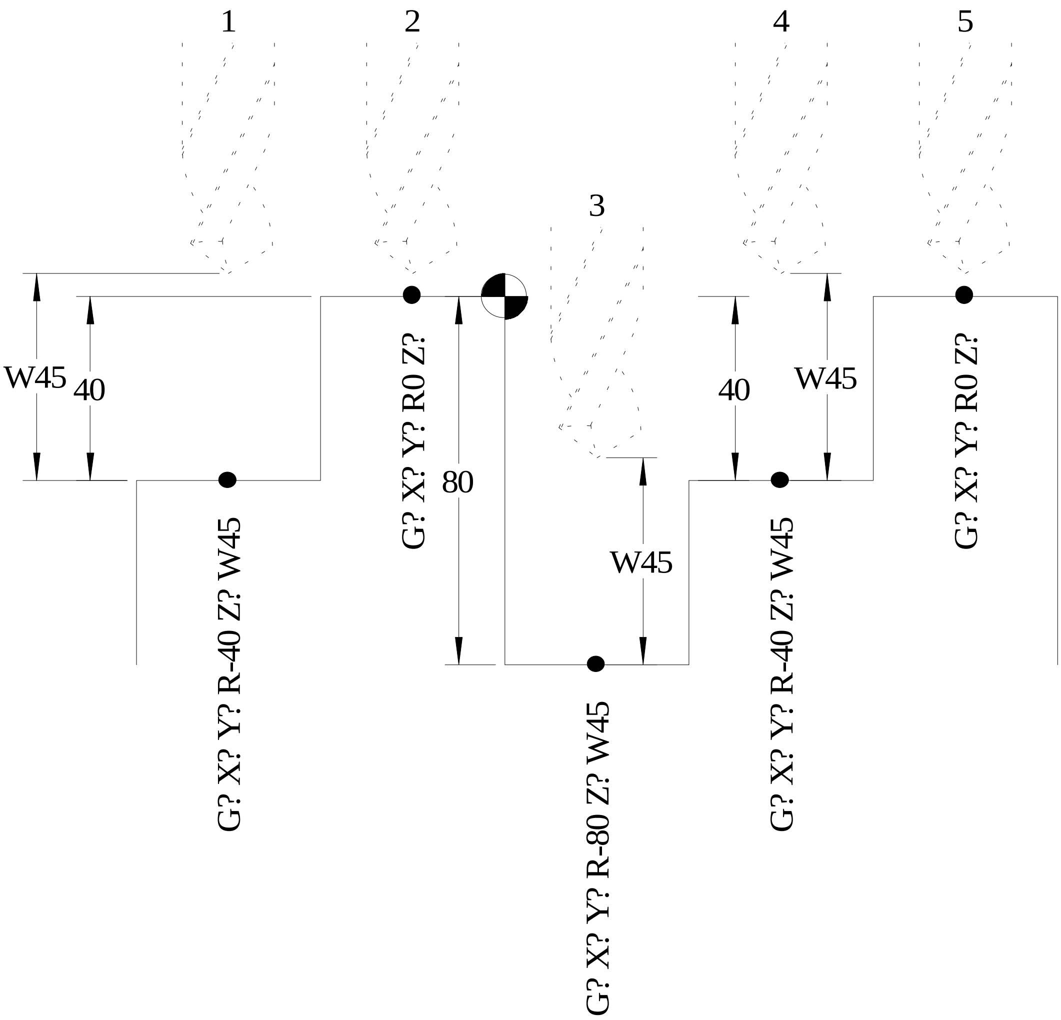
Example

:T1 M6
G0 G90 G40 G71 G17 G94
[$CYCLE_PARAMS(2)GAGE_HT_MM] = 5 (will set the parameter to 5)
X? Y? Z100 H? S? M3
Z5
G? R? Z? F? M8 (the gage height parameter is now 5mm)

<u>“W” Retraction at Hole End</u>

A basic canned cycle always follows a sequence of four operations:
1) Rapid traverse X & Y axis to hole centre position.
2) Rapid traverse down to “R” position stopping short by the “Gauge Height Parameter” value.
3) Feed to “Z” depth (including the “Gauge Height Parameter”.
4) Rapid to “R” position + “Gauge Height Parameter” or “W” retract distance above “R” position.

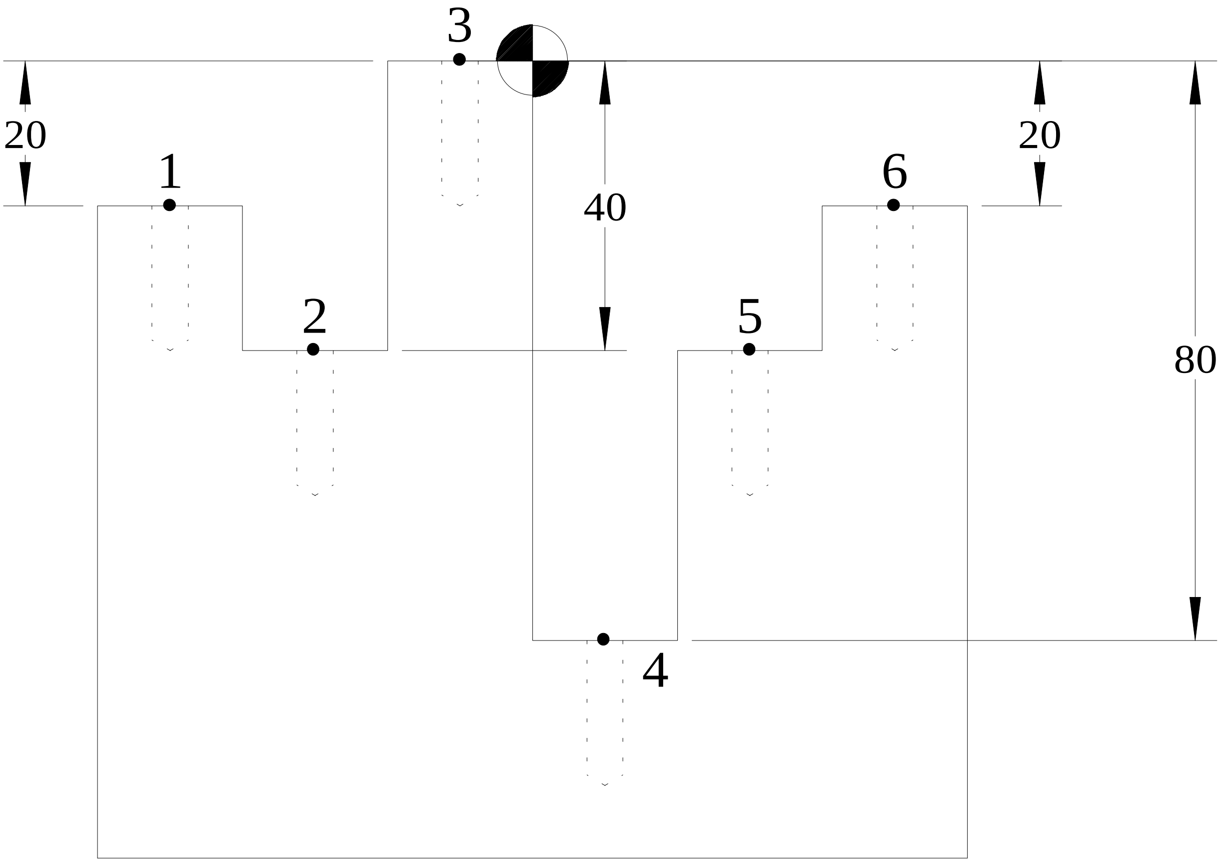
A **<u>non-modal incremental</u>** “W” word together with a numerical value on the hole cycle determines the final Z axis position after hole completion.



The actions are:
“W?” = Return the tool an incremental distance above the active programmed “R” position (this does not include the “Gauge Height Parameter” value).
No “W” = Return the tool point to the active programmed “R” position + “Gauge Height Parameter (3mm).

<u>DO NOT PROGRAM “W” WITH A ZERO</u>

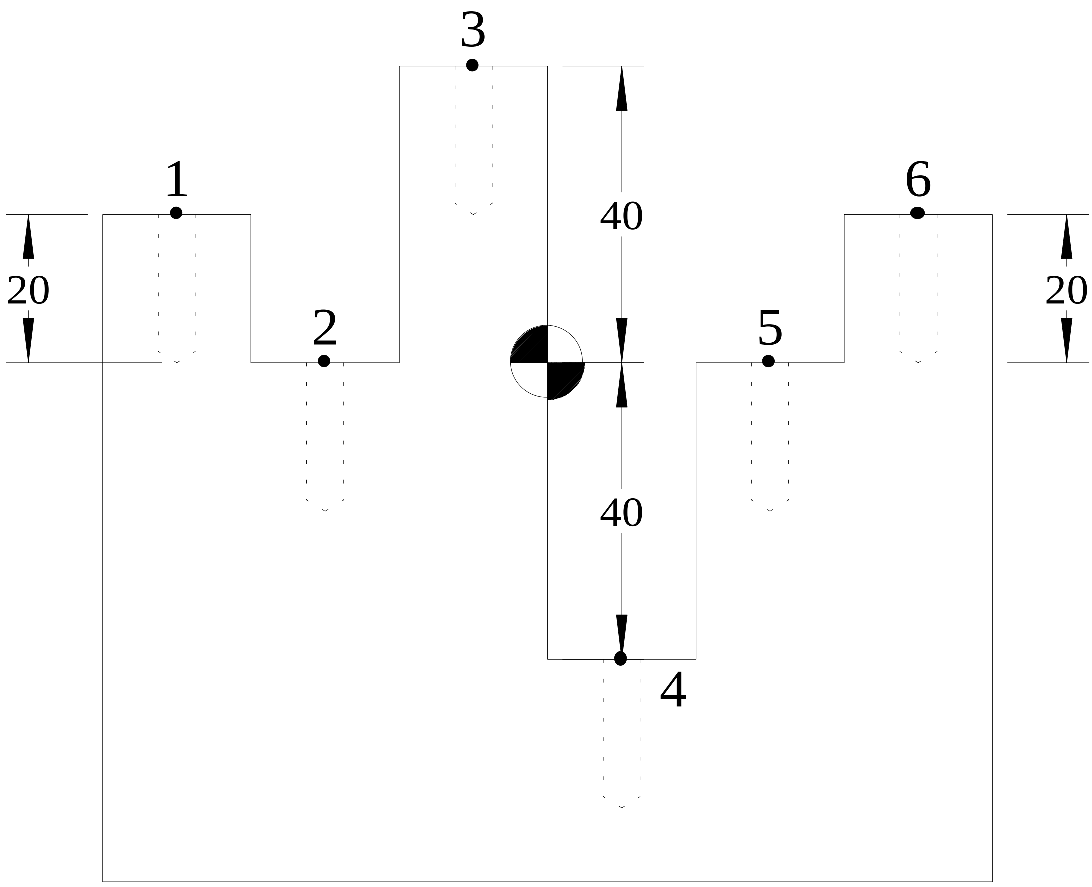
Hole Canned Cycles
“R” & “W” Positions



Section View YZ plane
All holes 20mm Deep

(Point 1)	G? X? Y? R-20 Z-20 F?
(Point 2)	X? Y? R-40 W45
(Point 3)	X? Y? R0
(Point 4)	X? Y? R-80 W45
(Point 5)	X? Y? R-40 W25
(Point 6)	X? Y? R-20

Hole Canned Cycles
“R” & “W” Positions



Section View YZ plane
All holes 20mm Deep

(Point 1)
(Point 2)
(Point 3)
(Point 4)
(Point 5)
(Point 6)

Hole cycle formats

Simple Drilling Cycle

G81 Z_____ R_____ F_____ ;minimum required for 1st operation.

X_____ Y_____ W_____ ;additional

Counterboring / Reaming (feed in / dwell / rapid out)

G82 Z_____ R_____ F_____ ;minimum required for 1st operation.

X_____ Y_____ W_____ ;additional

Peck Drilling

G83 Z_____ R_____ K_____ J_____ F_____ ;minimum required for 1st operation.

X_____ Y_____ W_____ ;additional

Standard Tapping

G84 Z_____ R_____ F_____ ;minimum required for 1st operation.

X_____ Y_____ W_____ ;additional

Rigid Tapping

G84.1 Z_____ R_____ F_____ ;minimum required for 1st operation.

X_____ Y_____ K_____ P_____ W_____ ;additional

Boring (feed in / feed out)

G85 Z_____ R_____ F_____ ;minimum required for 1st operation.

X_____ Y_____ W_____ ;additional

Boring (feed in / shift / rapid out)

G86 Z_____ R_____ F_____ ;minimum required for 1st operation.

X_____ Y_____ U_____ V_____ J_____ W_____ ;additional

Back Boring

G87 Z+_____ R_____ U_____ V_____ I_____ K_____ F_____ ;minimum for 1st operation.

X_____ Y_____ J_____ W_____ ;additional

Web Drilling /Boring

G88 Z_____ R_____ I_____ K_____ F_____ ;minimum for 1st operation (drilling).

X_____ Y_____ U_____ V_____ J_____ W_____ ;additional

Boring (feed in / dwell / feed out)

G89 Z_____ R_____ F_____ ;minimum for 1st operation.

X_____ Y_____ W_____ ;additional

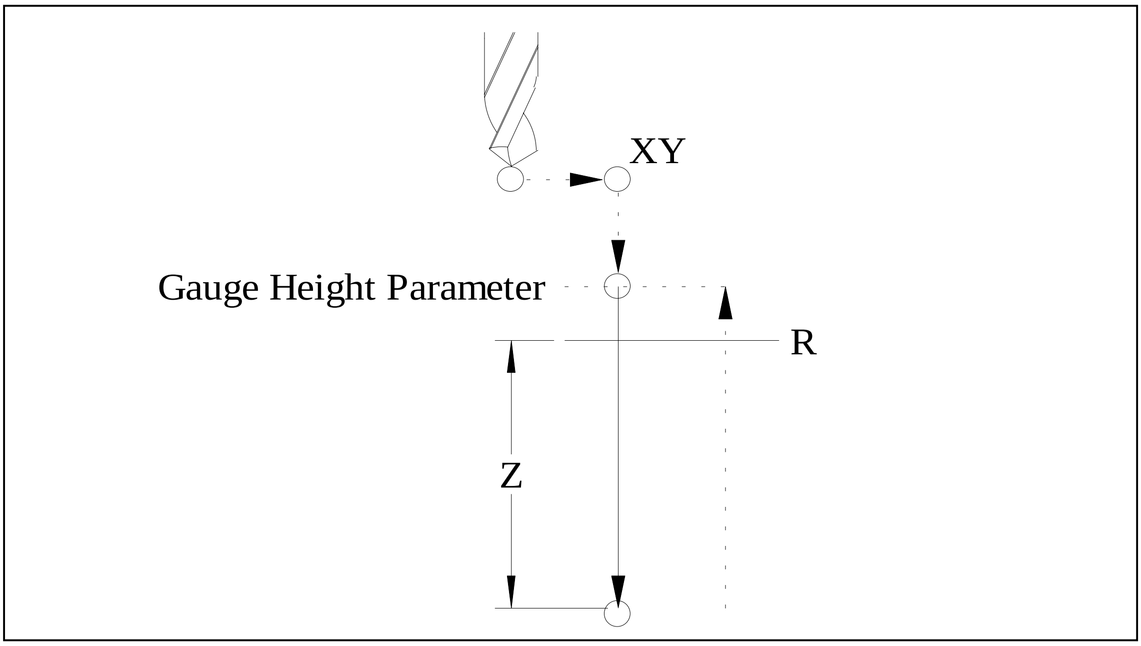
Simple Drilling
G81

G81 [X? Y?] Z? R? F?

X? = Modal hole centre position
Y? = Modal hole centre position
Z? = Modal hole depth position from "R"
R? = Modal hole surface position
F? = Modal cutting feedrate

[ ] denotes optional input for the first hole.

G81



S1000 M3	**- Spindle start.**
G81 X? Y? Z? R? F? M?	**- Position to 1st hole setting all data.**
X?	**- Position to 2nd hole.**
Y?	**- Position to 3rd hole**
G0 G90 Z100	**- Tool to a safe height (G0 cancel)**

<u>At Z finish position the tool retracts automatically.</u>

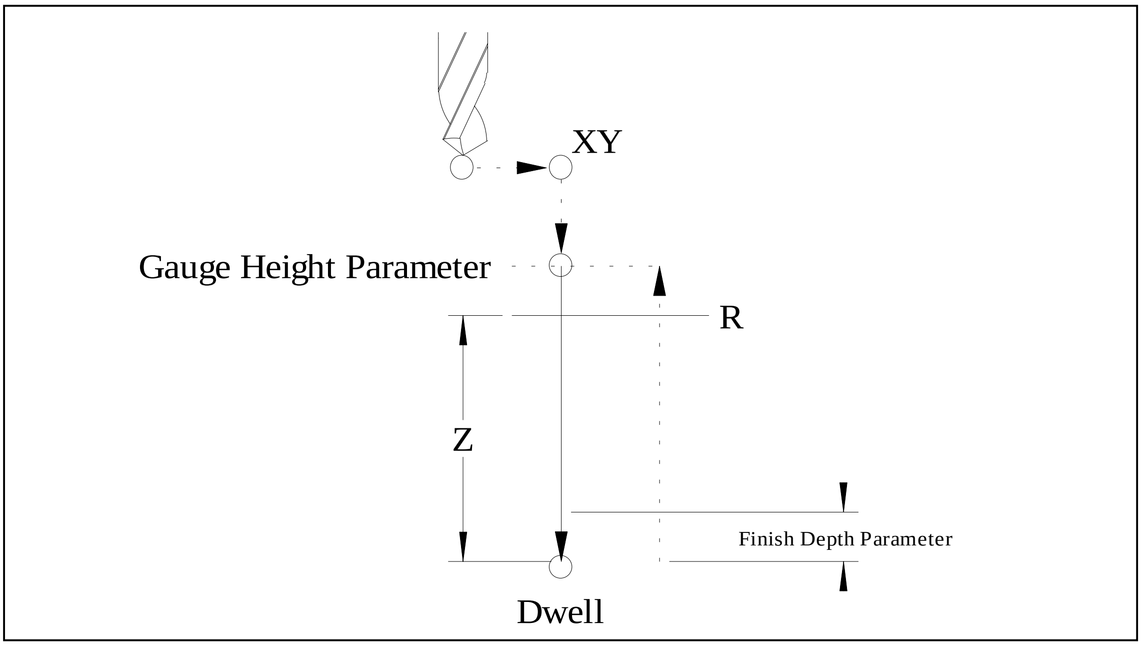
Drilling with a Dwell
G82

G82 [X? Y?] Z? R? F?

X? = Modal hole centre position
Y? = Modal hole centre position
Z? = Modal hole depth position from "R"
R? = Modal hole surface position
F? = Modal cutting feedrate

[ ] denotes optional input for the first hole.

G82



S1000 M3	**- Spindle start.**
G82 X? Y? Z? R? F? M?	**- Position to 1st hole setting all data.**
X?	**- Position to 2nd hole.**
Y?	**- Position to 3rd hole**
G0 G90 Z100	**- Tool to a safe height (G0 cancel)**

At the Z depth, the cycle will dwell at a pre-set time in seconds contained in the "Cycle Parameters" ("G82 Dwell").

The cycle parameters also allow the tool to change the feed by a percentage ("G82 Finish Feed Factor") at a set incremental depth from the hole bottom ("G82 Finish Depth")

At Z finish position the tool retracts automatically.

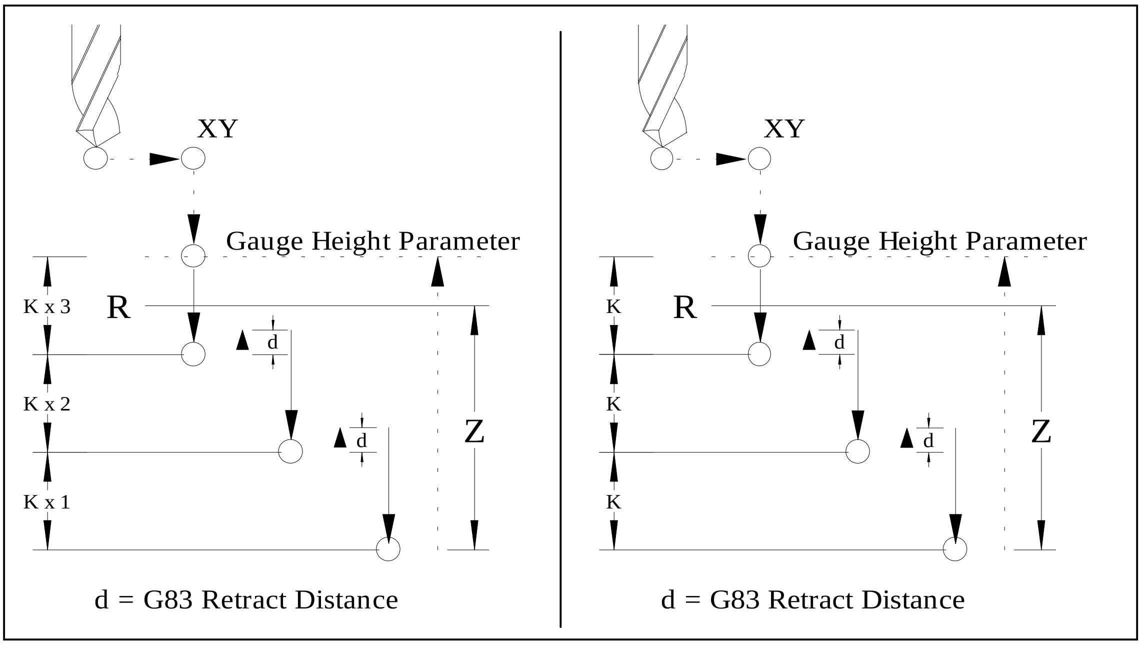
High Speed Peck Drilling
G83 J1 / J11

G83 [X? Y?] Z? R? K? [J?] F?

X? = Modal hole centre position
Y? = Modal hole centre position
Z? = Modal hole depth position from "R"
R? = Modal hole surface position
K? = Modal incremental depth of cut for each peck
J? = Type of peck (J1 = Variable peck – J11 = Standard peck)
F? = Modal cutting feedrate

[ ] denotes optional input for the first hole.

G83 J1	G83 J11



S1000 M3	**- Spindle start.**
G83 X? Y? Z? R? K? J11 F? M?	**- Position to 1st hole setting all data.**
X?	**- Position to 2nd hole.**
Y?	**- Position to 3rd hole**
G0 G90 Z100	**- Tool to a safe height (G0 cancel)**

This cycle creates a peck at the programmed pecking type (J?) at a value (K?) with a short chip break retraction at a value as set in the cycle parameters "G83 Retract Distance" before creating the next peck at value K? (High Speed Pecking).

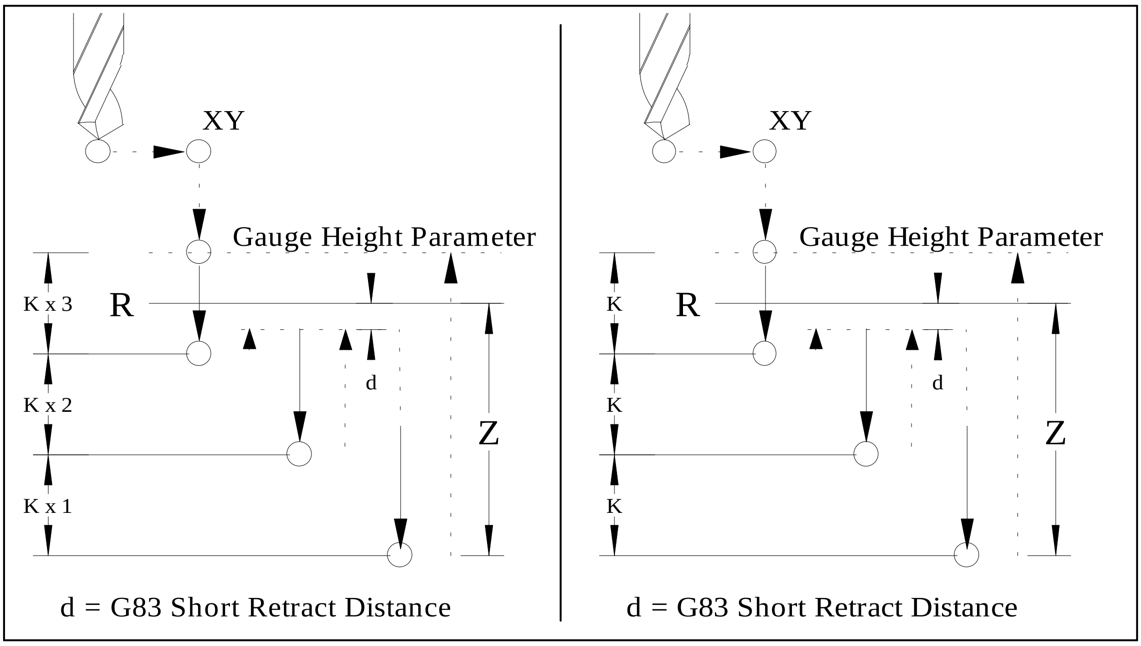
J1	**J11**
1st Peck = K x 3 2nd Peck = K x 2 All further Pecks = K x 1	All Pecks = K x 1

Deep Hole Drilling
G83 J2 / J12

G83 [X? Y?] Z? R? K? J? F?

X? = Modal hole centre position
Y? = Modal hole centre position
Z? = Modal hole depth position from “R”
R? = Modal hole surface position
K? = Modal incremental depth of cut for each peck
J? = Type of peck (J2 = Variable peck – J12 = Standard peck)
F? = Modal cutting feedrate
[ ] denotes optional input for the first hole.

G83 J2	G83 J12



S1000 M3	**- Spindle start.**
G83 X? Y? Z? R? K? J12 F? M?	**- Position to 1st hole setting all data.**
X?	**- Position to 2nd hole.**
Y?	**- Position to 3rd hole**
G0 G90 Z100	**- Tool to a safe height (G0 cancel)**

This cycle creates a peck at the programmed pecking type (J?) at a value (K?) with a long retraction leaving the drill in the hole by the amount set in the cycle parameters “Short Retract Distance“, advancing back into the hole stopping by a distance above the last peck by a value in “G83 Relief Amount” before creating the next peck at value K? until the cycle is completed. At the cycle end the drill will return to the “R” position + Gauge Height Parameter.

NOTE:
The pecking value (K?) must be greater than the cycle parameter “Short Retract Distance“ if using J2 or J12

Peck Drilling

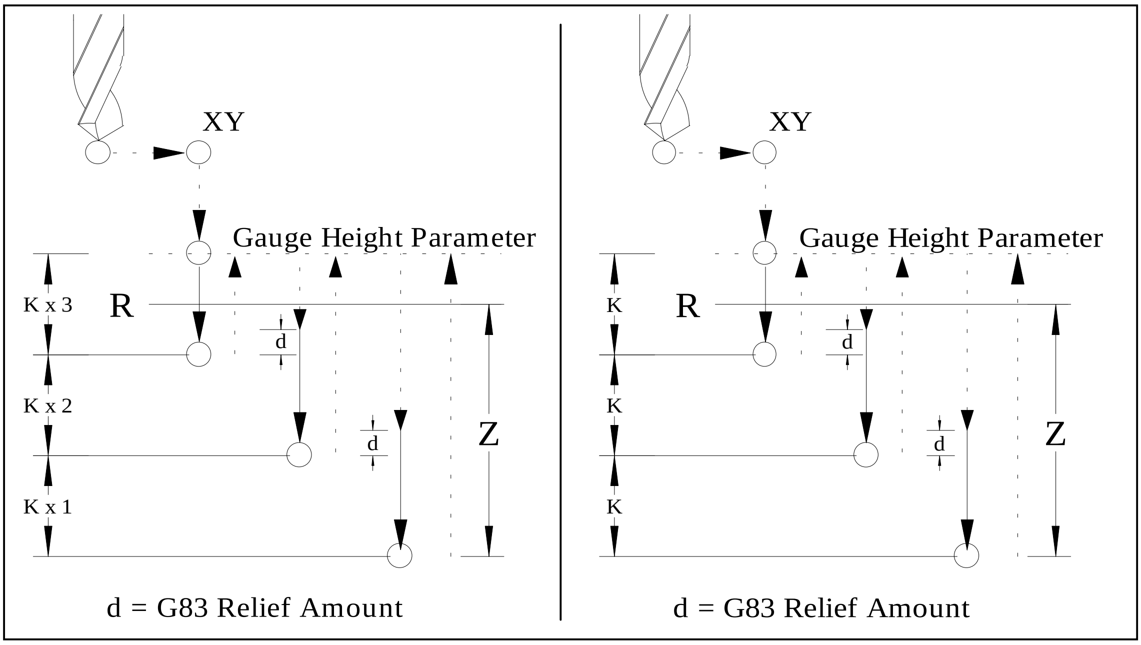
G83 J3 / J13

G83 [X? Y?] Z? R? K? J? F?

X? = Modal hole centre position
Y? = Modal hole centre position
Z? = Modal hole depth position from “R”
R? = Modal hole surface position
K? = Modal incremental depth of cut for each peck
J? = Type of peck (J3=Variable peck – J13=Standard peck)
F? = Modal cutting feedrate

[ ] denotes optional input for the first hole.

G83 J3	G83 J13



S1000 M3	**- Spindle start.**
G83 X? Y? Z? R? K? J13 F? M?	**- Position to 1st hole setting all data.**
X?	**- Position to 2nd hole.**
Y?	**- Position to 3rd hole**
G0 G90 Z100	**- Tool to a safe height (G0 cancel)**

This cycle creates a peck at the programmed pecking type (J?) at a value (K?) with a chip break retraction fully out of the hole between each peck back to the “R” position + “Gauge Height Parameter”, advancing back into the hole stopping short by the parameter “G83 Relief Amount” before creating the next peck at value K? until the cycle is completed.

At Z finish position the tool retracts out of the hole completely.

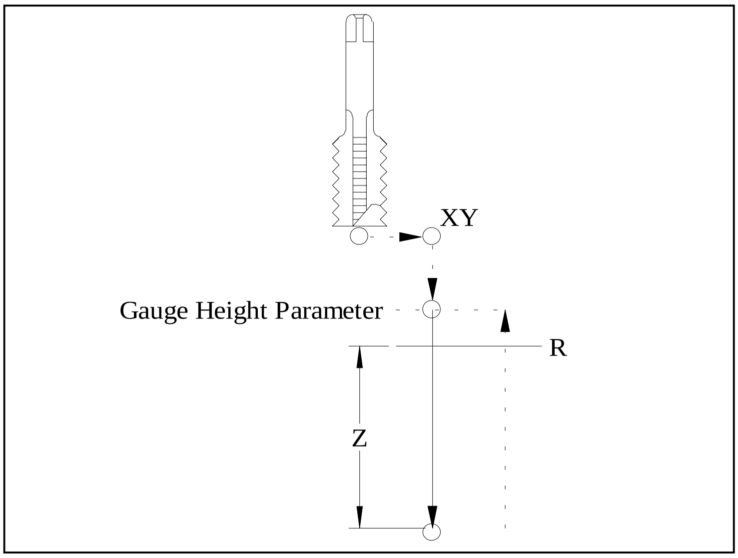
Standard Tapping
G84

G84 [G95 X? Y?] Z? R? [J?] F?

G95 = Feed per revolution (F? = pitch of tap)
X? = Modal hole centre position
Y? = Modal hole centre position
Z? = Modal absolute hole depth position
R? = Modal tool starting position above hole surface
J? = Spindle retract factor
F? = Modal cutting feedrate

[ ] denotes optional input for the first hole.

G84



S1000 M3 or M4 - **Spindle speed + spindle direction**
G84 X? Y? Z? R? J? F? M? - **Position to 1st hole setting all data.**
X? - **Position to 2nd hole.**
Y? - **Position to 3rd hole**
G0 G90 Z100 - **Tool to a safe height (G0 cancel)**

This cycle allows the tool to be retracted at a faster Speed/Feed to aid quicker cycle times. The J? word on the line of program will do this i.e. J3 will retract out of the hole 3 times faster than it went in.

<u>At Z finish position the tool retracts automatically.</u>

G95 (feed/revolution) on the line of program, the feedrate = the pitch of tap.
i.e. M8 x 1.25 pitch tap
<u>G95</u> G84 X? Y? Z? R? <u>F1.25</u> M?

<u>Ensure the next tool is set back to G94 (feed per minute) if programming feed per minute.</u>

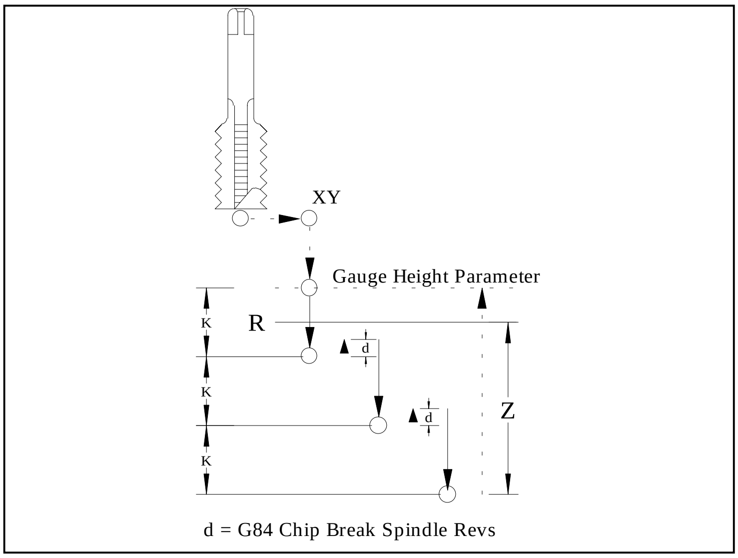
Synchronised Tapping
G84.1

G84.1 [G95 X? Y?] Z? R? [J? P? K?] F?

G95 = Feed per revolution (F? = pitch of tap)
X? = Modal hole centre position
Y? = Modal hole centre position
Z? = Modal absolute hole depth position
R? = Modal tool starting position above hole surface
J? = Spindle retract factor (see G84 notes)
P? = Modal spindle reverse amount in revolutions
K? = Modal pecking amount (K0 or not programmed = no peck)
F? = Modal cutting feedrate

[ ] denotes optional input for the first hole.

G84.1



S1000 M3 or M4	**- Spindle speed + spindle direction**
G84.1 X? Y? Z? R? J? K? P? F? M?	**- Position to 1st hole setting all data.**
X?	**- Position to 2nd hole.**
Y?	**- Position to 3rd hole**
G0 G90 Z100	**- Tool to a safe height (G0 cancel)**

This cycle allows the tap to peck tap if programmed with a “K” value. The “P” value is the amount of spindle revolutions required to retract from the last peck before commencing the next peck. If a “P” is not programmed then the parameter “G84 Chip Break Spindle Revs” will be used instead.

At Z finish position the tool retracts automatically.

G95 (feed/revolution) on the line of program, the feedrate = the pitch of tap.
i.e. M8 x 1.25 pitch tap
G95 G84.1 X? Y? Z? R? P? K? F1.25 M?

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

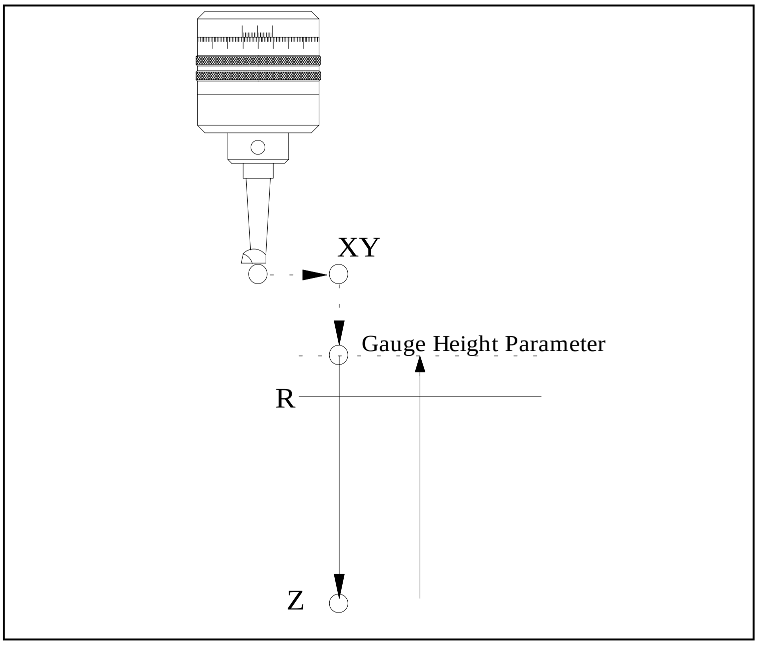
Boring / Reaming
G85

G85 [X? Y?] Z? R? F?

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

[ ] denotes optional input for the first hole.

G85



S1000 M3 ;	**- Spindle start.**
G85 X? Y? Z? R? F? M? ;	**- Position to 1st hole setting all data.**
X? ;	**- Position to 2nd hole.**
Y? ;	**- Position to 3rd hole**
G0 G90 G80 Z100 ;	**- Tool to a safe height (G80 cancel)**

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

At Z finish position the tool retracts automatically.

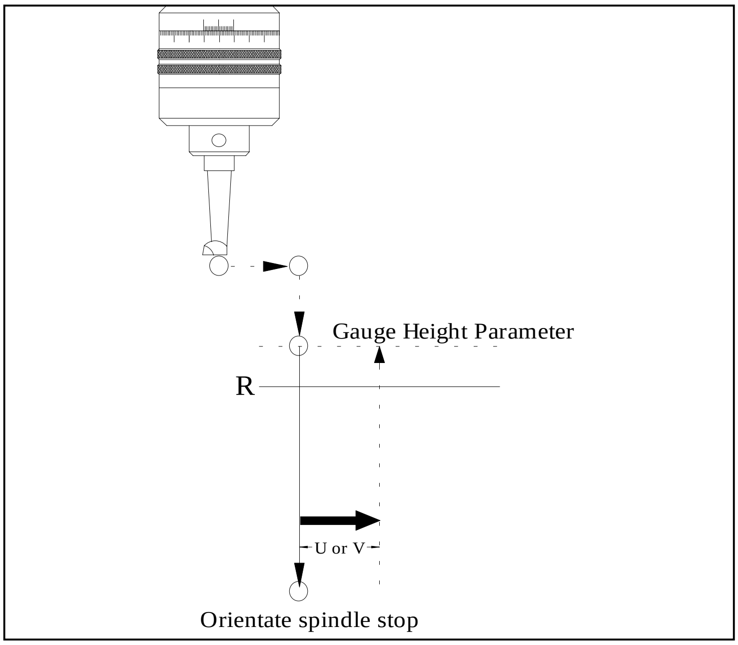
Fine Boring
G86

G86 [X? Y?] Z? R? U? or V? F?

X? = Modal hole centre position
Y? = Modal hole centre position
Z? = Modal hole depth position from "R"
R? = Modal hole surface position
U? or V? = Modal incremental axis shift off centreline (U=X shift, V=Y shift)
F? = Modal cutting feedrate

[ ] denotes optional input for the first hole.

G86



S1000 M3	**- Spindle start.**
G86 X? Y? Z? R? U? F? M?	**- Position to 1st hole setting all data.**
X?	**- Position to 2nd hole.**
Y?	**- Position to 3rd hole**
G0 G90 Z100	**- Tool to a safe height (G0 cancel)**

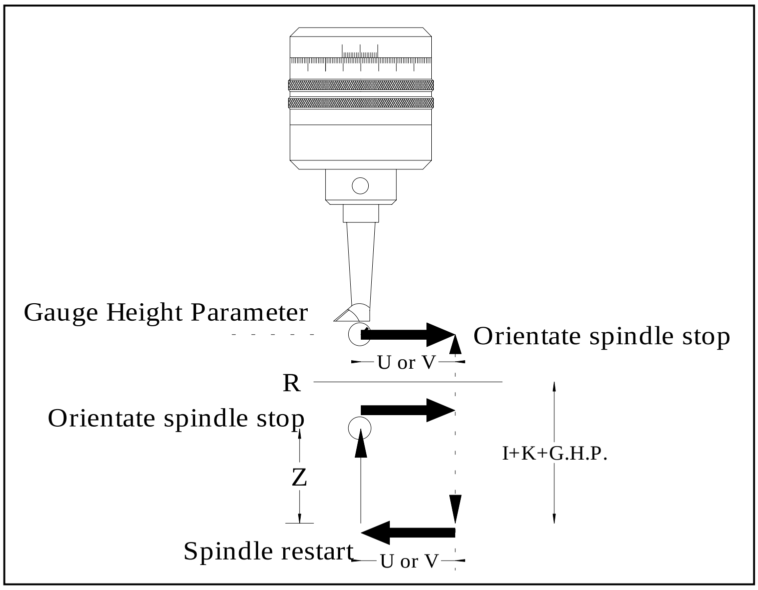
At Z finish position the tool will feed up automatically by a distance set in cycle parameters "G86 Bottom Retract Distance" while the spindle is in motion, spindle will stop, orientates to the toolchange angle position, shifts off bore centre line by the value in the program as U? or V? and retracts out of the hole in rapid.

Caution
U / V words must only be used with single point tools.
Tools must be mounted at the correct orientation and sufficient clearance must exist on the non cutting side of the boring tool, otherwise U / V offsets could cause a tool / part collision.

Back Boring
G87

G87 [X? Y?] Z? R? U? or V? I? K? F?
X? = Modal hole centre position
Y? = Modal hole centre position
Z? = Modal hole depth position from "R"
R? = Modal hole surface position
U? or V? = Modal incremental axis shift off centreline (U=X shift, V=Y shift)
I? = Modal pre-drilled hole depth
K? = Modal distance from tool end to cutting tip.
F? = Modal cutting feedrate
[ ] denotes optional input for the first hole.

G87



S1000 M3	**- Spindle start.**
G87 X? Y? Z? R? U? F? M?	**- Position to 1st hole setting all data.**
X?	**- Position to 2nd hole.**
Y?	**- Position to 3rd hole**
G0 G90 Z100	**- Tool to a safe height (G0 cancel)**

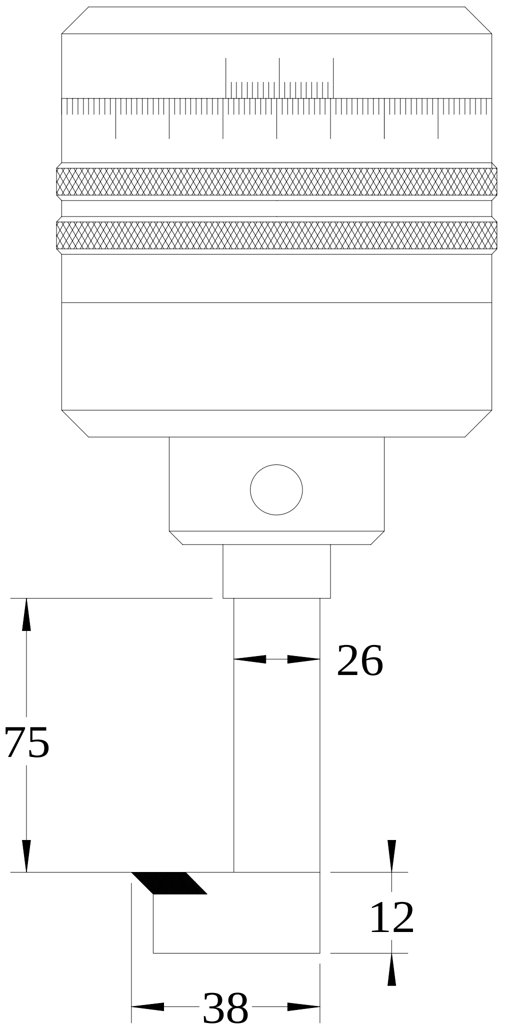
At G.H.P. position the tool will stop and orientates to the tool-change angle, shift off centreline by the value set in either U? or V?, plunge rapid through the pre-drilled hole by the distance of "I" + "K" + G.H.P., shift back onto centreline, start spindle and feed to the "Z" distance. At the Z position the tool will dwell by parameter "G87 Dwell Time" then feed back down by a value set in parameter "Bottom Retract Distance", spindle will stop and orientates, shift off centreline by U? or V? and retract out of the hole in rapid back to the top G.H.P..

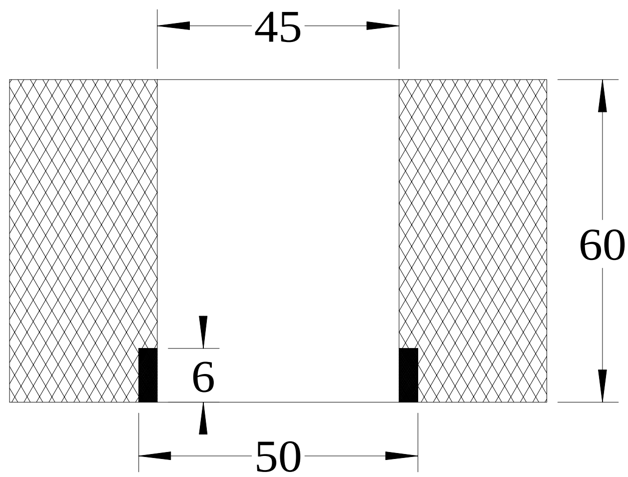
Caution
U / V words must only be used with single point tools.
Tools must be mounted at the correct orientation and sufficient clearance must exist on the non-cutting side of the boring tool, otherwise U / V offsets could cause a tool / part collision.

Back Boring
G87

MIN. U? = (38-(26/2))-(45/2) = 2.5

MAX. U? = ((45/2)-(26/2)) = 9.5





G87 X? Y? Z6 R0 I60 K12 U5 F? M8

<u>If no “K” word is programmed then the parameter “G87 Backbore Clearance” value will be used.</u>

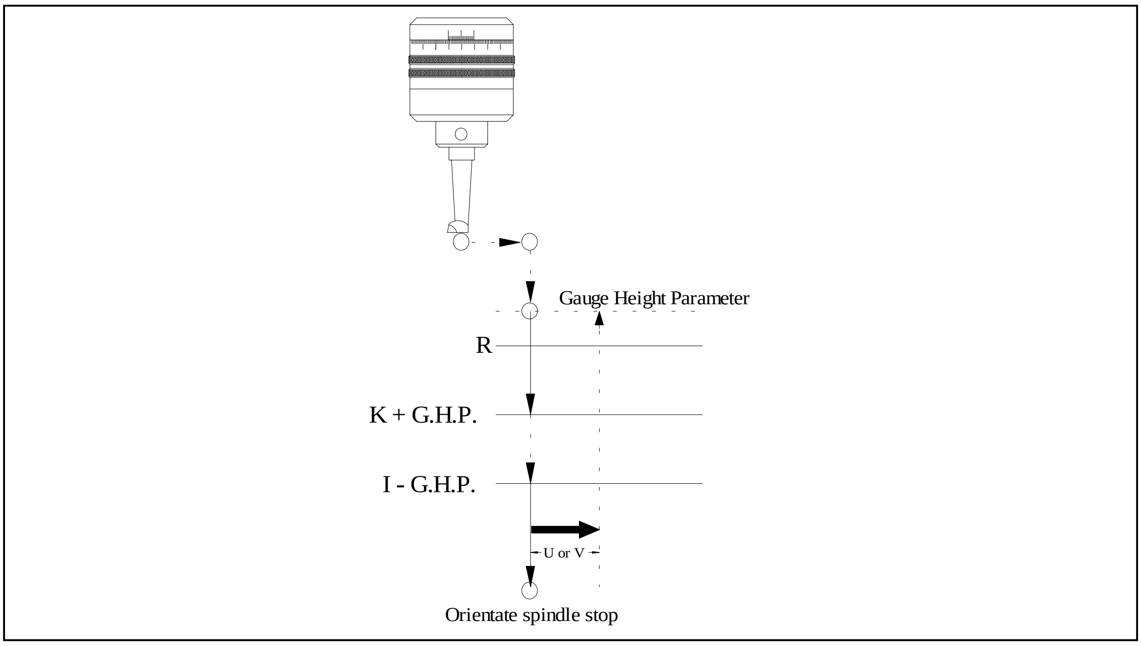
Web Boring/Drilling
G88

G88 [X? Y?] Z? R? I? K? U? or V? F?

X? = Modal hole centre position
Y? = Modal hole centre position
Z? = Modal hole depth position from “R”
R? = Modal hole surface position
K? = Modal 1st hole depth from “R”
I? = Modal distance between bottom of 1st hole and top of 2nd hole
U? or V? = Modal incremental axis shift off centreline (U=X shift, V=Y shift)
F? = Modal cutting feedrate

[ ] denotes optional input for the first hole.

G88

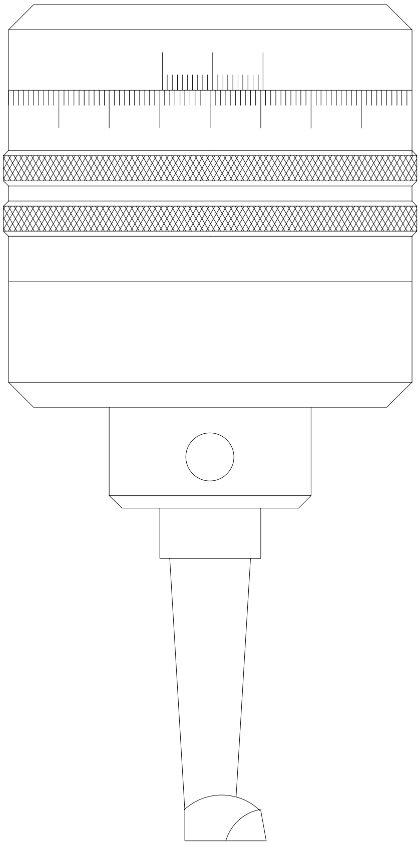


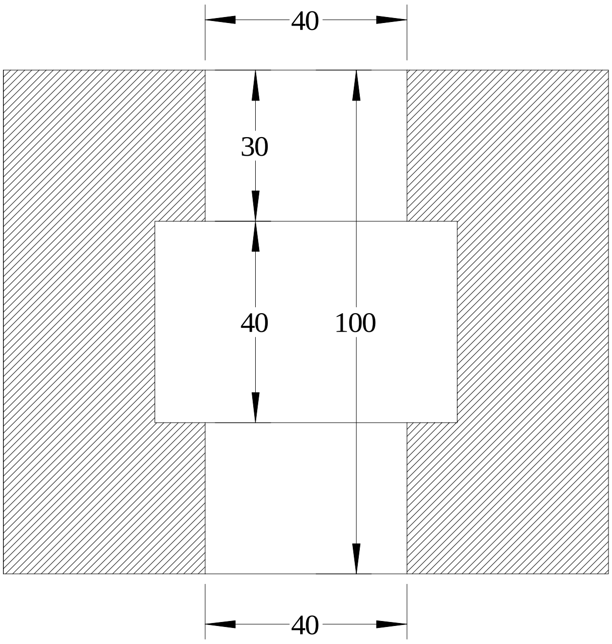
S1000 M3 ;	**- Spindle start.**
G88 X? Y? Z? R? I? K? U? F? M? ;	**- Position to 1st hole setting all data.**
X? ;	**- Position to 2nd hole.**
Y? ;	**- Position to 3rd hole**
G0 G90 Z100 ;	**- Tool to a safe height (G0 cancel)**

This cycle allows 2 bores in alignment to be machined in one cycle.
The tool will move in rapid to the XY location with the spindle rotating. The tool will then move in rapid to the “R” position + the Gauge Height Parameter. Feed motion will then occur to “K” + G.H.P. +”G88 Breakthrough Distance”. Rapid motion will then occur the distance of “K” +”I” – G.H.P. Feed motion to the “Z” programmed depth before retraction in rapid to the “R” point + G.H.P.

Omitting the U or V value will activate the cycle without a shift off centreline.

Web Boring/Drilling
G88





G88 X? Y? Z-100 R0 I40 K30 U-0.5 F? M8

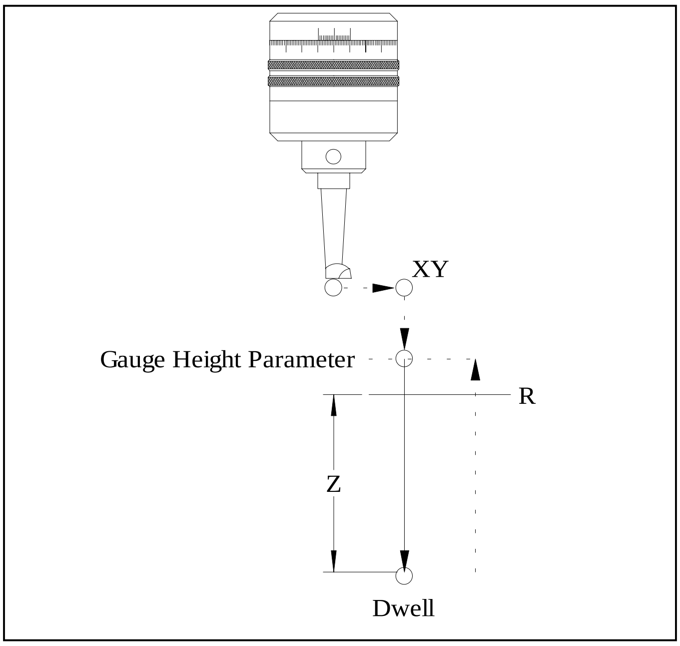
Boring/Ream with a Dwell
G89

G89 [X? Y?] Z? R? F?

X? = Modal hole centre position
Y? = Modal hole centre position
Z? = Modal hole depth position from “R”
R? = Modal hole surface position
F? = Modal cutting feedrate

[ ] denotes optional input for the first hole.

G89

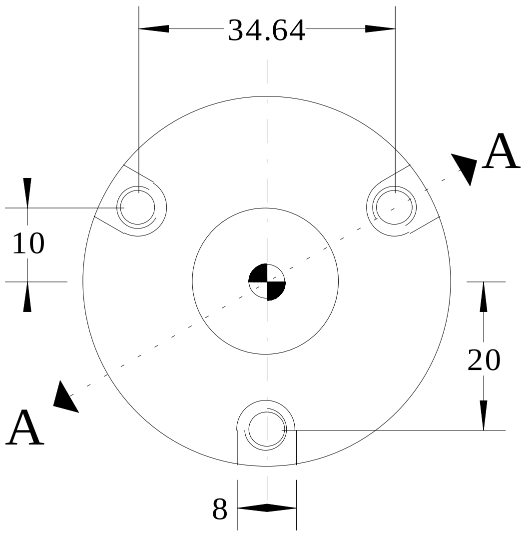


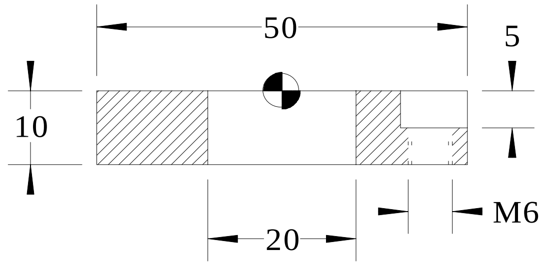
S1000 M3 ;	**- Spindle start.**
G89 X? Y? Z? R? F? M? ;	**- Position to 1st hole setting all data.**
X? ;	**- Position to 2nd hole.**
Y? ;	**- Position to 3rd hole**
G0 G90 Z100 ;	**- Tool to a safe height (G0 cancel)**

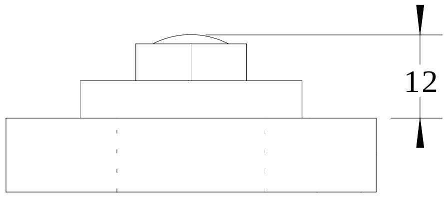
This cycle feeds to the programmed depth position, dwells, then retract back out of the hole at the same speed/feedrate.

At Z finish position the tool retracts automatically.

Hole Example







Fixture View

CHAPTER 9

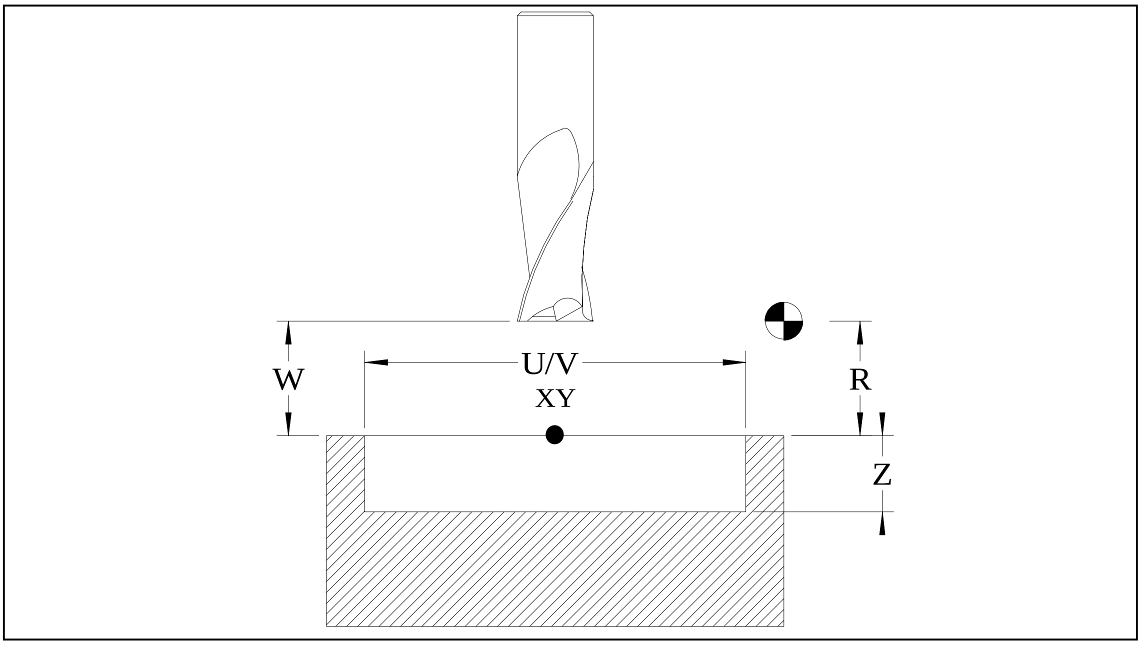
Milling Canned Cycles

The control has the ability to machine basic shapes using a series of "G" codes for different milling cycles.
These are simple circle/square pocket cycles, face milling cycles and frame cycles.
A basic line of program consists of modal words all containing numerical values as:

G? X? Y? R? Z-? Q? U? &or V? [W?] M?

Where:
G? = Cycle machining code.
X? & Y? = Absolute start position.
R? = feature surface position.
Z-? = Incremental depth of the feature to be machined from "R" position.
Q? = Type of milling
U? &or V? = Modal length/width or diameter of finished feature
W? = Non-modal incremental retract amount above the "R" position at the end of feature machining.
M? = Coolant M8 or M27 (option).

[ ] Optional input



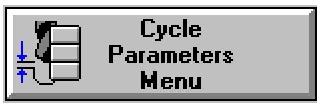
Cancel Canned Cycle

G80

Group Interpolation Codes

The following G codes also are effective in cancelling any Hole Canned Cycle:
G0, G01, G02, G03

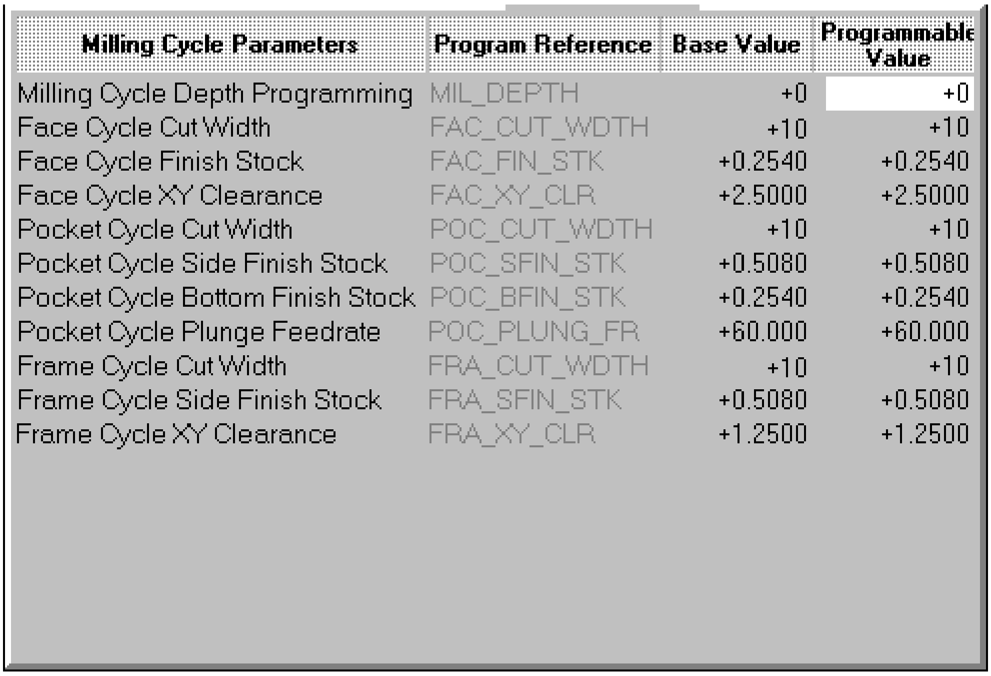
Cycle Parameters



Most of the milling canned cycles have the ability to store as default some of the data to be programmed if the data never changes to aid in easier programming.

If the data is not entered in the program then the values contained in the Mill Cycle Parameters is then used as the calculations.

The Gauge Height Parameter stored in the Hole Cycles parameter page is still used as the clearance value for the tool at the starting Z position.



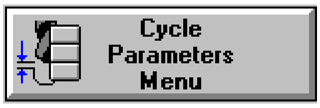
Milling Cycle Parameters	Program Reference	Base Value	Programmable Value
Milling Cycle Depth Programming	MIL_DEPTH	+0	+0
Face Cycle Cut Width	FAC_CUT_WDTH	+10	+10
Face Cycle Finish Stock	FAC_FIN_STK	+0.2540	+0.2540
Face Cycle XY Clearance	FAC_XY_CLR	+2.5000	+2.5000
Pocket Cycle Cut Width	POC_CUT_WDTH	+10	+10
Pocket Cycle Side Finish Stock	POC_SFIN_STK	+0.5080	+0.5080
Pocket Cycle Bottom Finish Stock	POC_BFIN_STK	+0.2540	+0.2540
Pocket Cycle Plunge Feedrate	POC_PLUNG_FR	+60.000	+60.000
Frame Cycle Cut Width	FRA_CUT_WDTH	+10	+10
Frame Cycle Side Finish Stock	FRA_SFIN_STK	+0.5080	+0.5080
Frame Cycle XY Clearance	FRA_XY_CLR	+1.2500	+1.2500

The parameter "Mill Cycle Depth Programming" tells the control how to use the "Z" value on the line of program.

* Default

Value	Z Function
0	Absolute Depth
1*	Incremental Depth

Cycle Parameters



The Milling cycle parameters can be pre-set to save on programming time and memory to certain values to be used by the program.
This can be done by the operator or by the use of "System Registers" which allow the program to write to the appropriate table.

Parameter	System Register
Milling Cycle Depth Programming	[$CYCLE_PARAMS(2)MIL_DEPTH] = ?
Face Cycle Cut Width	[$CYCLE_PARAMS(2)FAC_CUT_WDTH] = ?
Face Cycle Finish Stock	[$CYCLE_PARAMS(2)FAC_FIN_STK] = ?
Face Cycle XY Clearance	[$CYCLE_PARAMS(2)FAC_XY_CLR] = ?
Pocket Cycle Cut Width	[$CYCLE_PARAMS(2)POC_CUT_WDTH] = ?
Pocket Cycle Side Finish Stock	[$CYCLE_PARAMS(2)POC_SFIN_STK] = ?
Pocket Cycle Bottom Finish Stock	[$CYCLE_PARAMS(2)POC_BFIN_STK] = ?
Pocket Cycle Plunge Feedrate	[$CYCLE_PARAMS(2)POC_PLUNG_FR] = ?
Frame Cycle Cut Width	[$CYCLE_PARAMS(2)FRA_CUT_WDTH] = ?
Frame Cycle Side Finish Stock	[$CYCLE_PARAMS(2)FRA_SFIN_STK] = ?
Face Cycle XY Clearance	[$CYCLE_PARAMS(2)FRA_XY_CLR] = ?

Example

:T1 M6
G0 G90 G40 G71 G17 G94
[$CYCLE_PARAMS(2)MIL_DEPTH] = 1 (will set the parameter to 1 – Inc. Depth)
X? Y? Z100 H? S? M3
Z5
G? R? Z? Q? U? V? M8 (the Z dimension on the program line is incremental)

Mill cycle formats

Linear Facemilling

G22 X_____ Y_____ Z_____ R_____ U_____ V_____ Q_____ K_____ ;minimum required

O_____ P_____ J_____ F_____ S_____ W_____ ;additional

Linear Pocket

G23 X_____ Y_____ Z_____ R_____ U_____ V_____ Q_____ K_____ L_____ ;minimum required

,R_____ ,D_____ I_____ E_____ O_____ P_____ J_____ F_____ S_____ W____;additional

Linear Internal Frame

G24 X_____ Y_____ Z_____ R_____ U_____ V_____ Q_____ K_____ J_____ ;minimum required

,R_____ ,D_____ I_____ O_____ P_____ J_____ F_____ S_____ W_____ ;additional

Linear External Frame

G25 X_____ Y_____ Z_____ R_____ U_____ V_____ Q_____ K_____ J_____ ;minimum required

,R_____ ,D_____ I_____ O_____ P_____ J_____ F_____ S_____ W_____ ;additional

Circular Facemilling

G26 X_____ Y_____ Z_____ R_____ U_____ Q_____ K_____ ;minimum required

J_____ P_____ F_____ S_____ W_____ ;additional

Circular Pocket

G26.1 X_____ Y_____ Z_____ R_____ U_____ Q_____ K_____ ;minimum required

E_____ I_____ L_____ J_____ P_____ F_____ S_____ W_____ ;additional

Circular Internal Frame

G27 X_____ Y_____ Z_____ R_____ U_____ Q_____ J_____ K_____ ;minimum required

I_____ P_____ F_____ S_____ W_____ ;additional

Circular External Frame

G27.1 X_____ Y_____ Z_____ R_____ U_____ Q_____ J_____K_____ ;min.required

I_____ P_____ F_____ S_____ W_____ ;additional

Linear Pocket cutter calculations

The control needs to machine safely the shapes programmed and so the control will check that the cutter is small enough to machine the feature. The control follows the following calculations to check this.

The maximum size roughing cutter =
(Smaller of Length or Width - (2*STOCK ON SIDE))

The maximum size finish cutter =
(Smaller of ((Length or Width - 1mm) - (4*STOCK ON SIDE))

Circular Pocket cutter calculations

The control needs to machine safely the shapes programmed and so the control will check that the cutter is small enough to machine the feature. The control follows the following calculations to check this.

The maximum size roughing cutter =
(Diameter - (2*STOCK ON SIDE))

The maximum size finish cutter =
(Diameter - (4*STOCK ON SIDE))

The milling cycles of the A2100 control takes the “Nom Diameter” + “Dia Offset” as the overall diameter of the tool being used for machining.

Milling Canned Cycles

"Q" Cycle Types

Climb Milling Uni-Direction	Conventional Milling Bi-Direction	Operation	Applicable On:
Q0	Q10	Rough & Finish with a single finish pass.	All
Q1	Q11	Rough & Finish (multiple finish peck passes)	**Not G22 & G26 Facing Cycles**
Q2	Q12	Rough & leave Finish stock.	All
Q3	Q13	Rough to specified size.	All
Q4	Q14	Finish with a single finish pass.	All
Q5	Q15	Finish with multiple finish pecking passes.	**Not G22 & G26 Facing Cycles**

Plunging Types applicable on Pocket cycles.

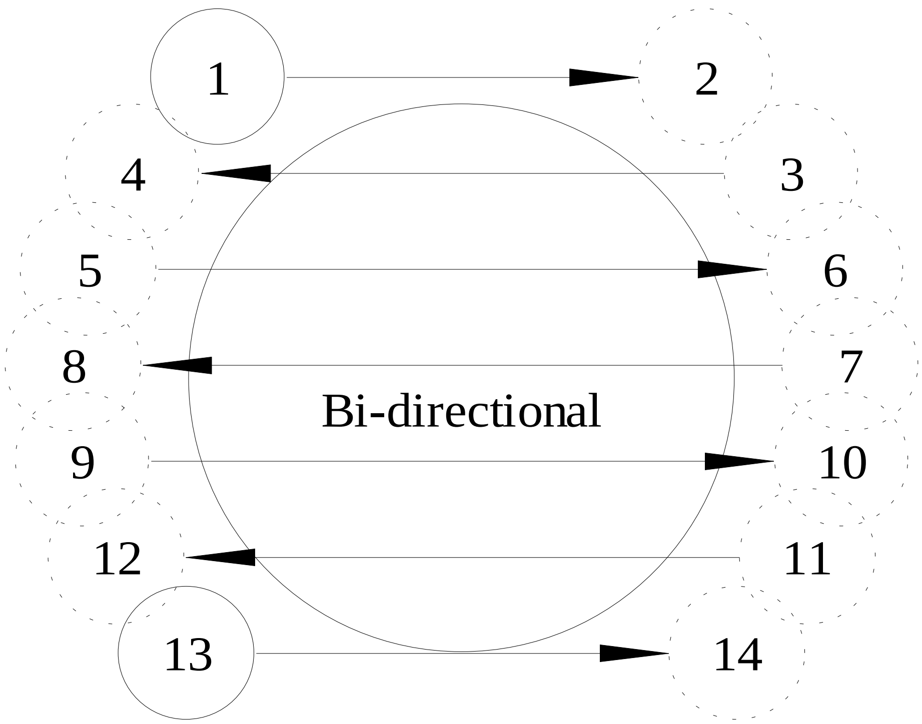
"L" Type	Operation	Remarks
L0	Ramp, plunge or Helical Ramp relative to "K" (peck).	See notes on G23 Pre-drill calculations.
L-1	The use of a pre-drilled Hole	See notes on G23 Pre-drill calculations.
L > 0	Ramp, plunge or Helical Ramp relative to user angle	**Not applicable on G26.1**

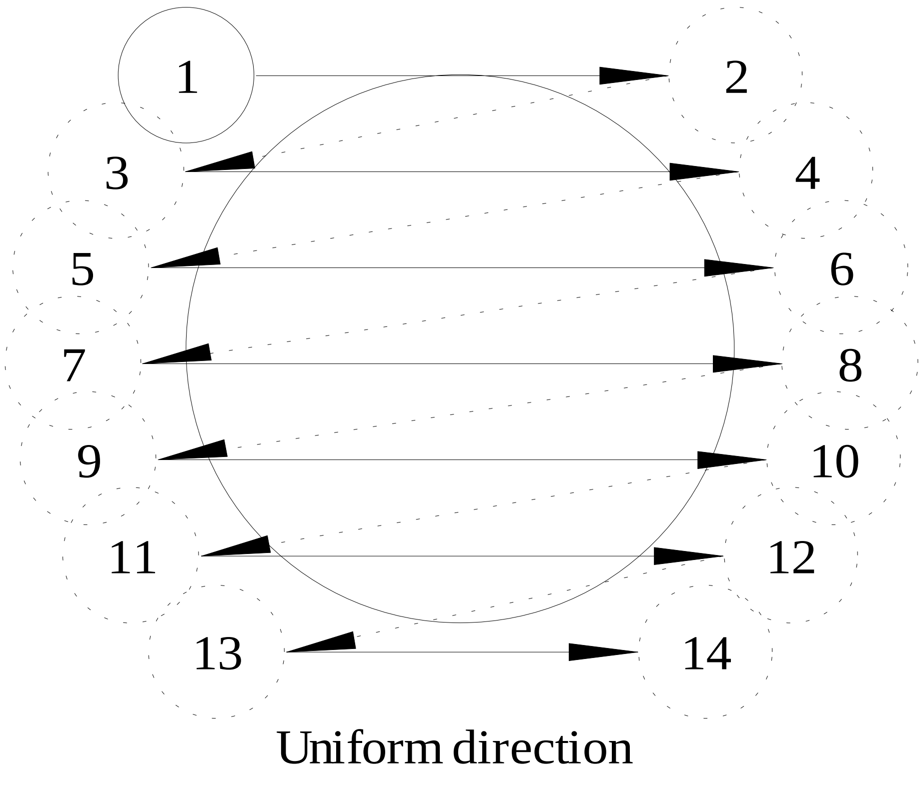
****NOTE****

Roughing "SPEED" & "FEED" must be programmed before the programmed cycle line.

Milling Canned Cycles

"Facemill Paths"





Linear Feature Facemill – XY Centre

G22

G22 [X? Y?] U? V? Z? R? Q? K? [O? W?] (J? P?) { F? S? }

Where:
X? = Modal feature centre position
Y? = Modal feature centre position
U? = Modal length of feature in the X axis
V? = Modal width of feature in the Y axis
Z? = Modal feature depth position from "R"
R? = Modal surface position
Q? = Modal type of milling (Q0, Q2, Q3, Q4, Q10, Q12, Q13, Q14)
K? = Modal incremental pecking amount in the Z axis (if "K"="Z" 1 peck)
O? = Modal angle of feature from horizontal plan view
W? = Non-modal incremental retraction in Z axis from "R"
J? = Finish stock to be removed (J0 = no stock)
P? = Modal width of cut as a percentage (10% - 80%)
F? = Modal cutting finishing feedrate
S? = Modal cutting finish spindle speed

[ ] denotes optional input
( ) if not programmed – values taken from cycle parameters.
{ } Only required with Q0, Q1, Q4, Q5, Q10, Q11, Q14, Q15

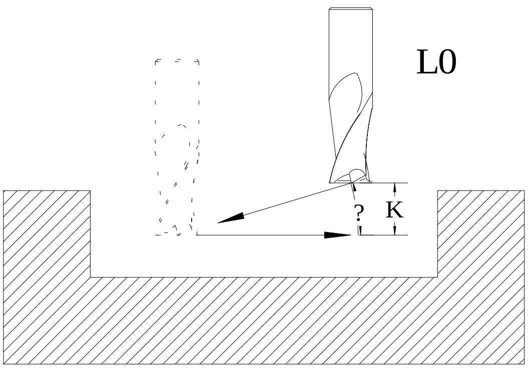
S1000 F? M3	**- Spindle start + Rough mill feed.**
G22 X? Y? U? V? Z? R? Q? P? J? K?	**- Position to 1st position setting all data.**
X?	**- Position to 2nd position**
Y?	**- Position to 3rd position**
G0 G90 Z100	**- Tool to a safe height (G0 cancel)**

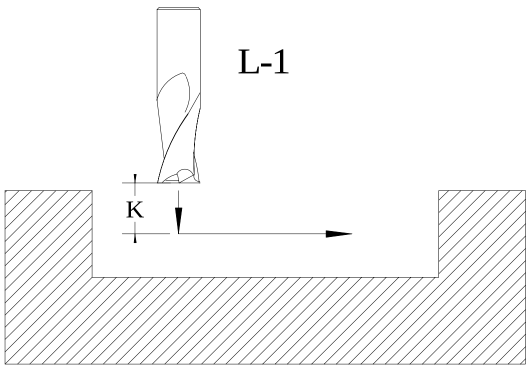
At Z finish position the tool retracts automatically.

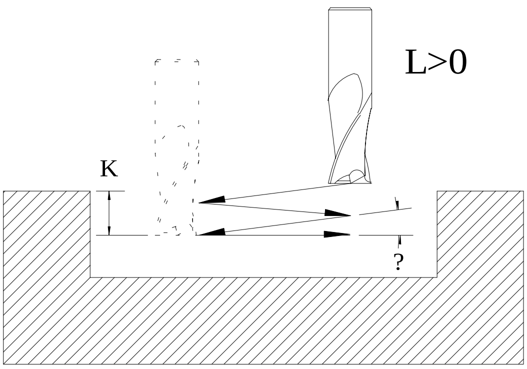
Pocket Mill
“L” values

The control allows up to three types of plunging with the pocket milling cycles. These are:

L0 = Ramp or Helical ramp at “K” pecking amount
L-1 = Feed to “K” depth in a straight Z axis motion
L? greater than 0 = Ramp at chosen angle (G23 only)







Linear Feature Pocket Mill – XY Centre

G23

G23 [X? Y?] U? V? Z? R? Q? K? L? E? [O? W? ,R? ,D?] (I? J? P?) { F? S? }

Where:
X? = Modal feature centre position
Y? = Modal feature centre position
U? = Modal length of feature in the X axis
V? = Modal width of feature in the Y axis
Z? = Modal feature depth position from "R"
R? = Modal surface position
Q? = Modal type of milling (All Q's)
K? = Modal incremental pecking amount in the Z axis (if "K"="Z" 1 peck)
L? = Modal type of plunging (L0, L-1, L>0)
E? = Modal Z axis plunge feedrate
O? = Modal angle of feature from horizontal plan view
W? = Non-modal incremental retraction in Z axis from "R"
,R? = Modal corner radius of feature
,D? = Corner feed ratio override (,D0 = Full slowdown, ,D100 = No slowdown)
I? = Finish stock on side to be removed (I0 = no stock)
J? = Finish stock on bottom to be removed (J0 = no stock)
P? = Modal width of cut as a percentage (10% - 80%)
F? = Modal cutting finishing feedrate
S? = Modal cutting finish spindle speed

[ ] denotes optional input
( ) if not programmed – values taken from cycle parameters.
{ } Only required with Q0, Q1, Q4, Q5, Q10, Q11, Q14, Q15

S1000 F? M3	**- Spindle start + Rough mill feed.**
G23 X? Y? U? V? Z? R? Q? L? E? P? I? J? K?	**- Position to 1st position setting all data.**
X?	**- Position to 2nd position**
Y?	**- Position to 3rd position**
G0 G90 Z100	**- Tool to a safe height (G0 cancel)**

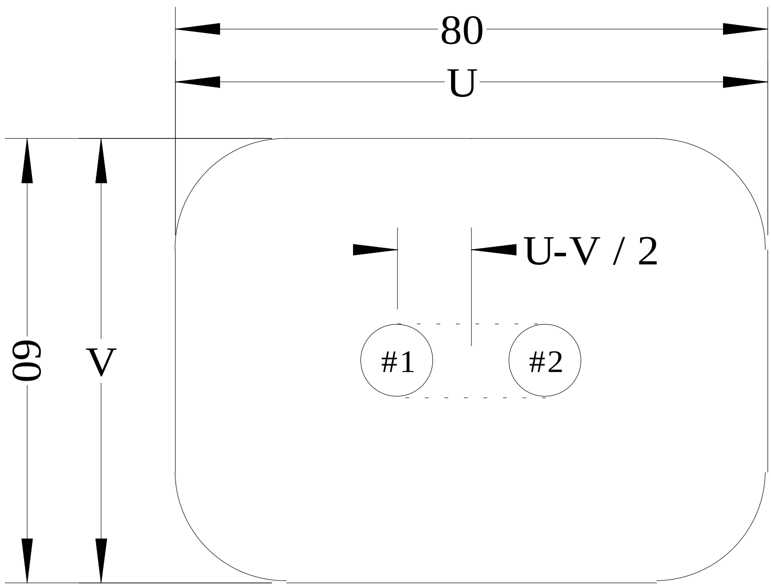
At Z finish position the tool retracts automatically.

Linear Feature Pocket Mill – XY Centre

G23 – Pre-drilled hole calculations

In order for the machine to maintain a constant cutting width throughout the milling cycle (“P” word) to preserve tool life, the milling cycle for G23 will start cutting in the pocket centre if the shape is a true square with equal sides.

If the shape to be machined is not a true equal sided square then constant cutting would not be performed if the cut path started in the centre. Therefore the control calculates a cut starting position at positions #1 or #2 as the graphic example shows dependant on the type of Z axis plunging selected with the “L” word (L0, L-1 or L>0).



The formula to calculate the cutting starting point is:

$$\frac{\textbf{U (length) – V (width)}}{\textbf{2}}$$

The cutter will start at either #1 or #2 of the longest side length, feed to pecking depth “K” using one of the selected "L" words with the plunging feed “E” and produce an equal length slot to the opposite side as the graphics above shows. In all cases of “L” plunge the tool end position will be at #2 as a flat slot. The cutter will now follow this slot profile until the pocket is completed.

“L”	**Starting position**
L0	#2
L-1	#1
L greater than 0 (ie L3)	Dependant on peck depth, slot length & chosen angle.

Linear Feature Inside Frame – XY Centre

G24

G24 [X? Y?] U? V? Z? R? Q? K? J? [O? W? ,R? ,D?] (I? P?) { F? S? }

Where:
X? = Modal feature centre position
Y? = Modal feature centre position
U? = Modal length of feature in the X axis
V? = Modal width of feature in the Y axis
Z? = Modal feature depth position from "R"
R? = Modal surface position
Q? = Modal type of milling (All Q's)
K? = Modal incremental pecking amount in the Z axis (if "K"="Z" 1 peck)
J? = Total stock on inside to be removed
O? = Modal angle of feature from horizontal plan view
W? = Non-modal incremental retraction in Z axis from "R"
,R? = Modal corner radius of feature
,D? = Corner feed ratio override (,D0 = Full slowdown, ,D100 = No slowdown)
I? = Finish stock on side to be removed (I0 = no stock)
P? = Modal width of cut as a percentage (10% - 80%)
F? = Modal cutting finishing feedrate
S? = Modal cutting finish spindle speed

[ ] denotes optional input
( ) if not programmed – values taken from cycle parameters.
{ } Only required with Q0, Q1, Q4, Q5, Q10, Q11, Q14, Q15

S1000 F? M3	**- Spindle start + Rough mill feed.**
G24 X? Y? U? V? Z? R? Q? P? I? J? K?	**- Position to 1st position setting all data.**
X?	**- Position to 2nd position**
Y?	**- Position to 3rd position**
G0 G90 Z100	**- Tool to a safe height (G0 cancel)**

At Z finish position the tool retracts automatically.

THIS CYCLE ASSUMES NO MATERIAL IS TO BE REMOVED IN THE POCKET CENTRE AND SO WILL RAPID Z-AXIS TO "K" DEPTH.

Linear Feature Outside Frame – XY Centre

G25

G25 [X? Y?] U? V? Z? R? Q? K? J? [O? W? ,R? ,D?] (I? P?) { F? S? }

Where:
X? = Modal feature centre position
Y? = Modal feature centre position
U? = Modal length of feature in the X axis
V? = Modal width of feature in the Y axis
Z? = Modal feature depth position from "R"
R? = Modal surface position
Q? = Modal type of milling (All Q's)
K? = Modal incremental pecking amount in the Z axis (if "K"="Z" 1 peck)
J? = Total stock on outside to be removed
O? = Modal angle of feature from horizontal plan view
W? = Non-modal incremental retraction in Z axis from "R"
,R? = Modal corner radius of feature
,D? = Corner feed ratio override (,D0 = Full slowdown, ,D100 = No slowdown)
I? = Finish stock on side to be removed (I0 = no stock)
P? = Modal width of cut as a percentage (10% - 80%)
F? = Modal cutting finishing feedrate
S? = Modal cutting finish spindle speed

[ ] denotes optional input
( ) if not programmed – values taken from cycle parameters.
{ } Only required with Q0, Q1, Q4, Q5, Q10, Q11, Q14, Q15

S1000 F? M3	**- Spindle start + Rough mill feed.**
G25 X? Y? U? V? Z? R? Q? P? I? J? K?	**- Position to 1st position setting all data.**
X?	**- Position to 2nd position**
Y?	**- Position to 3rd position**
G0 G90 Z100	**- Tool to a safe height (G0 cancel)**

At Z finish position the tool retracts automatically.

Circular Facemill

G26

G26 [X? Y?] U? Z? R? Q? K? [W?] (J? P?) { F? S? }

Where:
X? = Modal feature centre position
Y? = Modal feature centre position
U? = Modal finish diameter of feature.
Z? = Modal feature depth position from "R"
R? = Modal surface position
Q? = Modal type of milling (Q0, Q2, Q3, Q4, Q10, Q12, Q13, Q14)
K? = Modal incremental pecking amount in the Z axis (if "K"="Z" 1 peck)
W? = Non-modal incremental retraction in Z axis from "R"
J? = Finish stock to be removed (J0 = no stock)
P? = Modal width of cut as a percentage (10% - 80%)
F? = Modal cutting finishing feedrate
S? = Modal cutting finish spindle speed

[ ] denotes optional input
( ) if not programmed – values taken from cycle parameters.
{ } Only required with Q0, Q1, Q4, Q5, Q10, Q11, Q14, Q15

S1000 F? M3	**- Spindle start + Rough mill feed.**
G26 X? Y? U? Z? R? Q? P? J? K?	**- Position to 1st position setting all data.**
X?	**- Position to 2nd position**
Y?	**- Position to 3rd position**
G0 G90 Z100	**- Tool to a safe height (G0 cancel)**

At Z finish position the tool retracts automatically.

Circular Pocket Mill

G26.1

G26.1 [X? Y?] U? Z? R? Q? K? L? E? [W?] (I? J? P?) { F? S? }

Where:
X? = Modal feature centre position
Y? = Modal feature centre position
U? = Modal finish diameter of feature
Z? = Modal feature depth position from "R"
R? = Modal surface position
Q? = Modal type of milling (All Q's)
K? = Modal incremental pecking amount in the Z axis (if "K"="Z" 1 peck)
L? = Modal type of plunging (L0, L-1,)
E? = Modal Z axis plunge feedrate
W? = Non-modal incremental retraction in Z axis from "R"
I? = Finish stock on side to be removed (I0 = no stock)
J? = Finish stock on bottom to be removed (J0 = no stock)
P? = Modal width of cut as a percentage (10% - 80%)
F? = Modal cutting finishing feedrate
S? = Modal cutting finish spindle speed

[ ] denotes optional input
( ) if not programmed – values taken from cycle parameters.
{ } Only required with Q0, Q1, Q4, Q5, Q10, Q11, Q14, Q15

S1000 F? M3	**- Spindle start + Rough mill feed.**
G26.1 X? Y? U? Z? R? Q? L? E? P? I? J? K?	**- Position to 1st position setting all data.**
X?	**- Position to 2nd position**
Y?	**- Position to 3rd position**
G0 G90 Z100	**- Tool to a safe height (G0 cancel)**

At Z finish position the tool retracts automatically.

Circle Inside Frame

G27

G27 [X? Y?] U? Z? R? Q? K? J? [W?] (I? P?) { F? S? }

Where:
X? = Modal feature centre position
Y? = Modal feature centre position
U? = Modal finish diameter of feature
Z? = Modal feature depth position from "R"
R? = Modal surface position
Q? = Modal type of milling (All Q's)
K? = Modal incremental pecking amount in the Z axis (if "K"="Z" 1 peck)
J? = Total stock on inside to be removed
W? = Non-modal incremental retraction in Z axis from "R"
I? = Finish stock on side to be removed (I0 = no stock)
P? = Modal width of cut as a percentage (10% - 80%)
F? = Modal cutting finishing feedrate
S? = Modal cutting finish spindle speed

[ ] denotes optional input
( ) if not programmed – values taken from cycle parameters.
{ } Only required with Q0, Q1, Q4, Q5, Q10, Q11, Q14, Q15

S1000 F? M3	**- Spindle start + Rough mill feed.**
G27 X? Y? U? Z? R? Q? P? I? J? K?	**- Position to 1st position setting all data.**
X?	**- Position to 2nd position**
Y?	**- Position to 3rd position**
G0 G90 Z100	**- Tool to a safe height (G0 cancel)**

At Z finish position the tool retracts automatically.

THIS CYCLE ASSUMES NO MATERIAL IS TO BE REMOVED IN THE POCKET CENTRE AND SO WILL RAPID Z-AXIS TO "K" DEPTH.

Circle Outside Frame

G27.1

G27.1 [X? Y?] U? Z? R? Q? K? J? [W?] (I? P?) { F? S? }

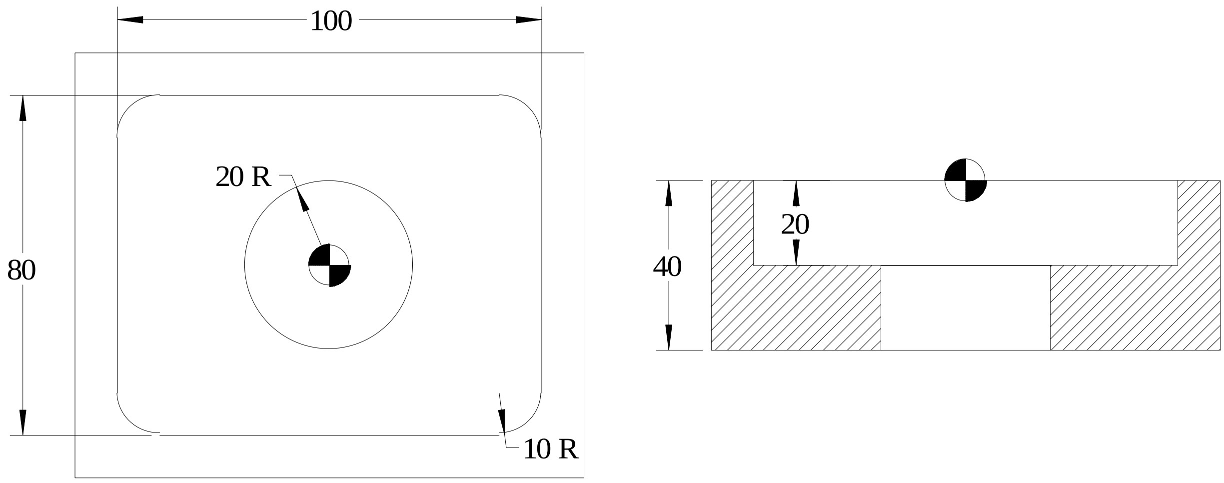
Where:
X? = Modal feature centre position
Y? = Modal feature centre position
U? = Modal finish diameter of feature
Z? = Modal feature depth position from "R"
R? = Modal surface position
Q? = Modal type of milling (All Q's)
K? = Modal incremental pecking amount in the Z axis (if "K"="Z" 1 peck)
J? = Total stock on outside to be removed
W? = Non-modal incremental retraction in Z axis from "R"
I? = Finish stock on side to be removed (I0 = no stock)
P? = Modal width of cut as a percentage (10% - 80%)
F? = Modal cutting finishing feedrate
S? = Modal cutting finish spindle speed

[ ] denotes optional input
( ) if not programmed – values taken from cycle parameters.
{ } Only required with Q0, Q1, Q4, Q5, Q10, Q11, Q14, Q15

S1000 F? M3	**- Spindle start + Rough mill feed.**
G27.1 X? Y? U? Z? R? Q? P? I? J? K?	**- Position to 1st position setting all data.**
X?	**- Position to 2nd position**
Y?	**- Position to 3rd position**
G0 G90 Z100	**- Tool to a safe height (G0 cancel)**

At Z finish position the tool retracts automatically.

Milling Cycle Example



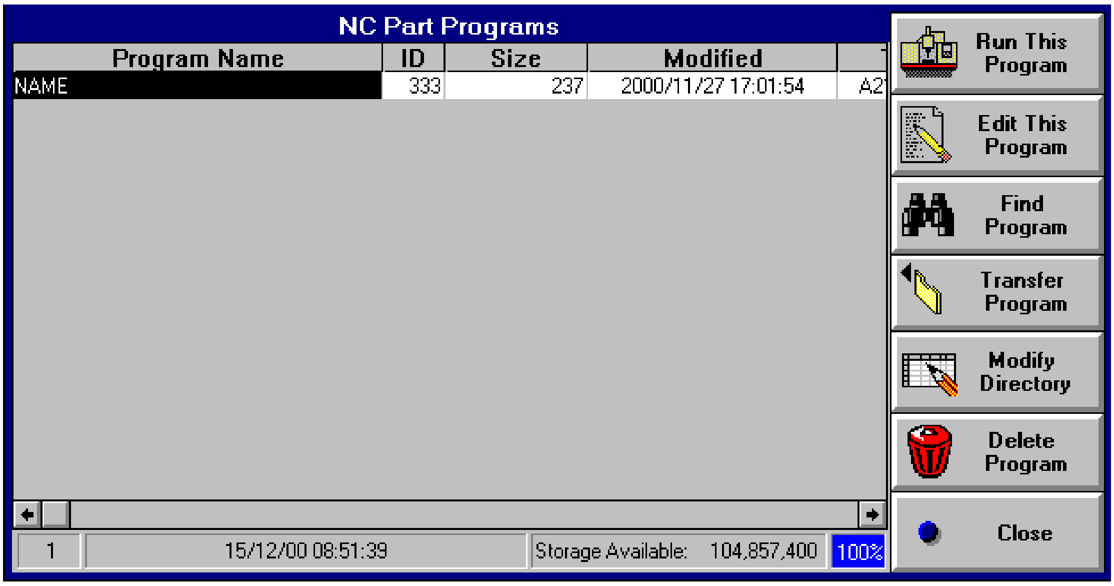
CHAPTER 10

Sub-Programming

The control can store part of a program, which can be used as many times as required. It can be stored within the program library as a separate program (sub-program) or within the main program itself to save programming time and memory.

By storing the program separately, any main program can access this stored sub-program. If the program is stored within the main program itself, then access by other programs is not possible.

Sub-programs can be stored as program names or identifying numbers (ID). Program names are limited to a total of 32 text characters and ID numbers are limited to a number between 0 – 99999 (use of the "Modify Directory" key allows editing of the program name or ID columns.



If the Sub-program being called is stored in the main directory (stored as a name) or stored in the main program itself, then access is by the use of a "Call Sub-Program" command CLS, followed by the sub-program name. If using the name then the name must be enclosed in " " all of which is enclosed in open and closed brackets ( ).

i.e. **N10 (CLS, "NAME")**

If the Sub-program being called is stored in the main directory (stored as an ID number) or stored in the main program itself, then access is by the use of a "Call Sub-Program" command CLS, followed by the sub-program ID number. If using the ID number then the number does not include the " " all of which is enclosed in open and closed brackets ( ).

i.e. **N10 (CLS, 333)**

Since the data to be used as a sub-program can be stored within the main program, there needs to be a method of separating the main program data from the information to be used as the sub-program. The use of a starting block and an ending block is used to separate the main program from the sub-program. The starting block uses a DFS, command with the identical name or ID number of the program in the CLS, block of information as stored. Programs stored in the main library do **not** require a DFS, block at the beginning.

i.e. **N100 (DFS, "NAME")** or **N100 (DFS,333)**

Sub-Programming - continued

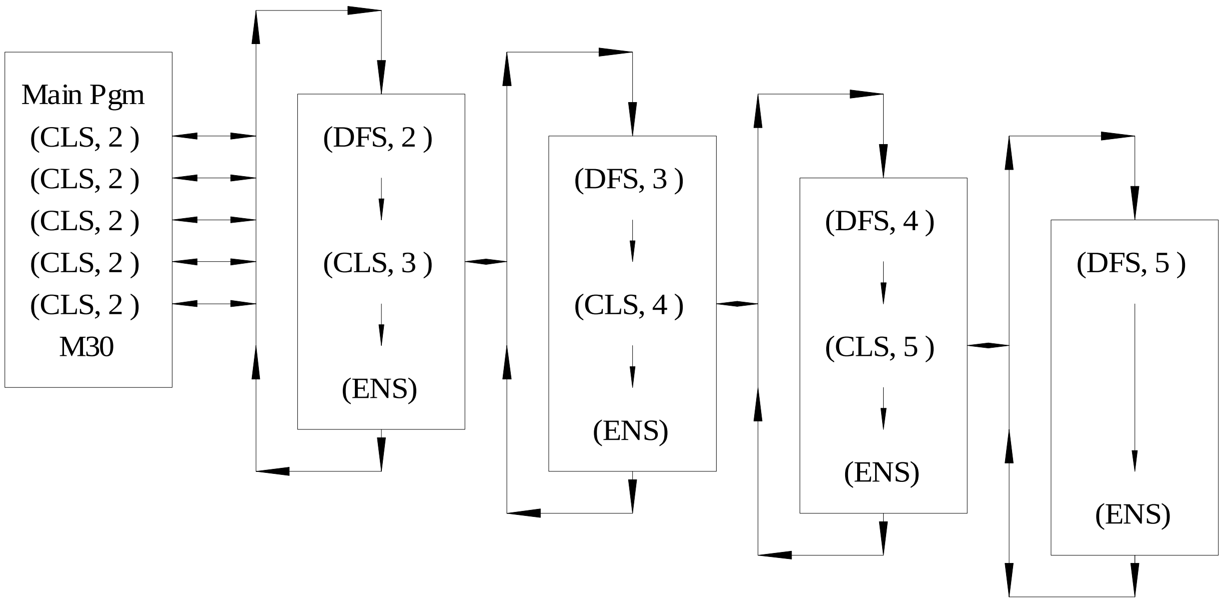
If the sub-program is stored in the main program and starts with a DFS, block, the program must recognize the end of the sub-program. To enable the control to recognize the end and return to the last CLS position for the program to continue, then an (ENS) block command on the last line of the sub-program will enable this.

i.e. **N200 (ENS)**

When actually running the main program, the control cannot read any data programmed in the subprogram until it actually reads a CLS block of the same name or ID number. If the control gets to a point of a program where the next line is a DFS, block before it reads the corresponding CLS, block, then the control skips all lines of information up to the ENS block of the sub-program then carries on with the rest of the program.

(CLS, “NAME”	Call sub-program
(DFS, “NAME”)	Define starting block of the sub-program
______________	{ Program
______________	{ Program
______________	{ Program
(ENS)	Ending block of the sub-program

Sub-programs can call other sub-programs to a maximum of 4 levels (nests)



i.e. program 1 can call program 2 which can call program 3 which can call program 4 whuch can call program 5. If program 5 tries to call program 6 then an alarm will occur. The program must go back at least one nest before another program call is required from this nest.

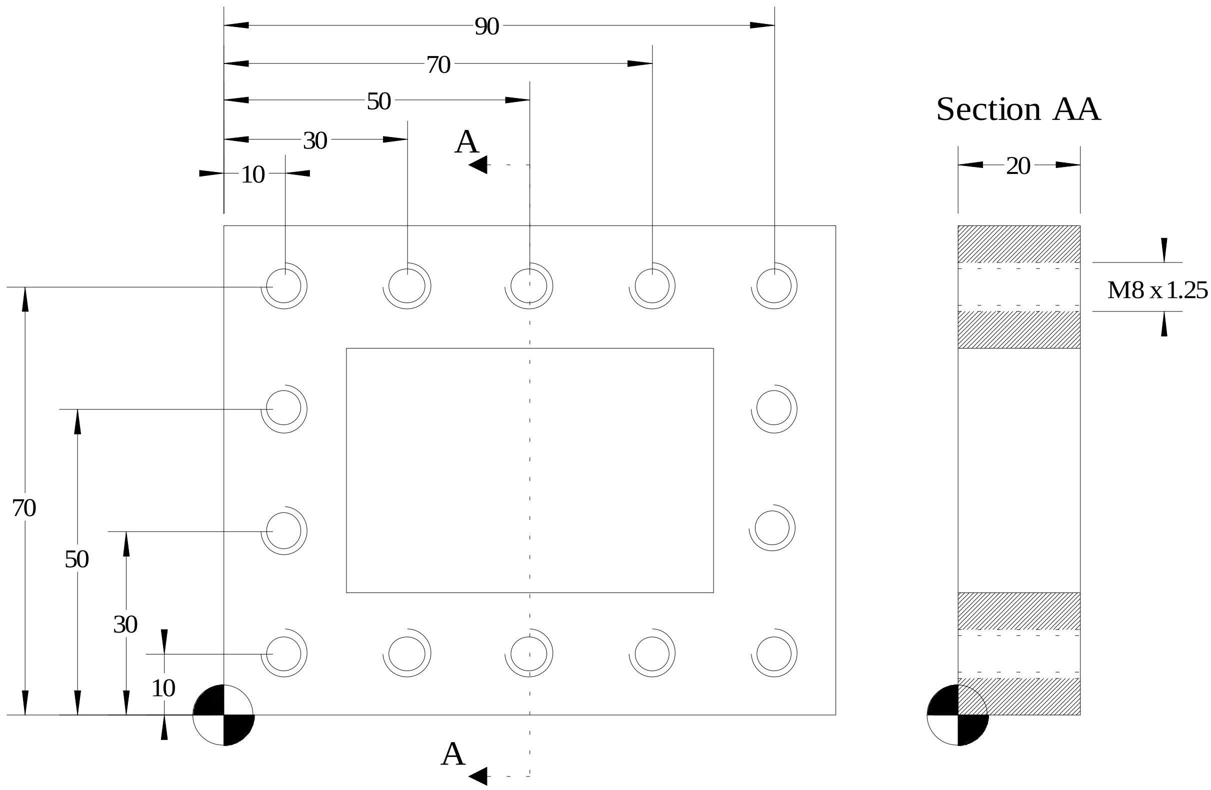
There is no limit to the amount of sub-programs called within each nested level.

Sub-Program repeats

The control also has the ability to contain a repeat command as part of the CLS, program line. Any numerical value programmed after the Name or ID number (as below) is acted as a repeat of this line of program.

i.e. **N10 (CLS, “NAME” , 26)**

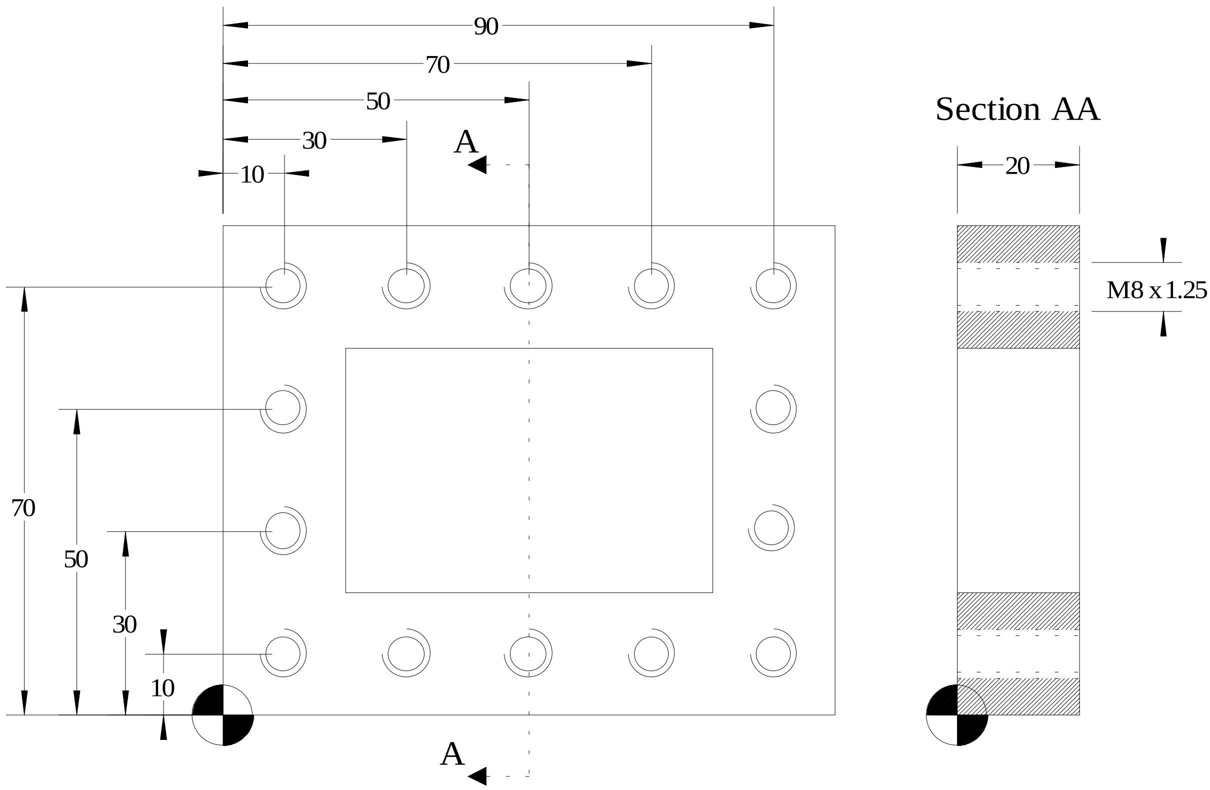
Sub-Programming
"Normal Program"



: T1 M6	Y10	X30
G0 G90 G40 G71 G17 G94	X70	X50
X10 Y10 Z100 H1 S? M3	X50	X70
Z5	X30	X90
G81 R0 Z-20 F? M8	G80	Y50
Y30	G0 G90 Z100	Y30
Y50	: T2 M6	Y10
Y70	G0 G90 G40 G71 G17 G94	X70
X30	X10 Y10 Z100 H1 S? M3 ;	X50
X50	Z5	X30
X70	G84.1 G95 R0 Z-20 F1.25 M8	G80
X90	Y30	G0 G90 Z100
Y50	Y50	M30
Y30	Y70	

Sub-Programming

"Sub-Program stored separately in library as NAME"



NC Part Programs			
Program Name	ID	Size	Modified
HOLES	27	237	2000/11/27 17:01:54
MAIN1	0	377	2000/11/30 12:41:40

(Program Name = MAIN1)

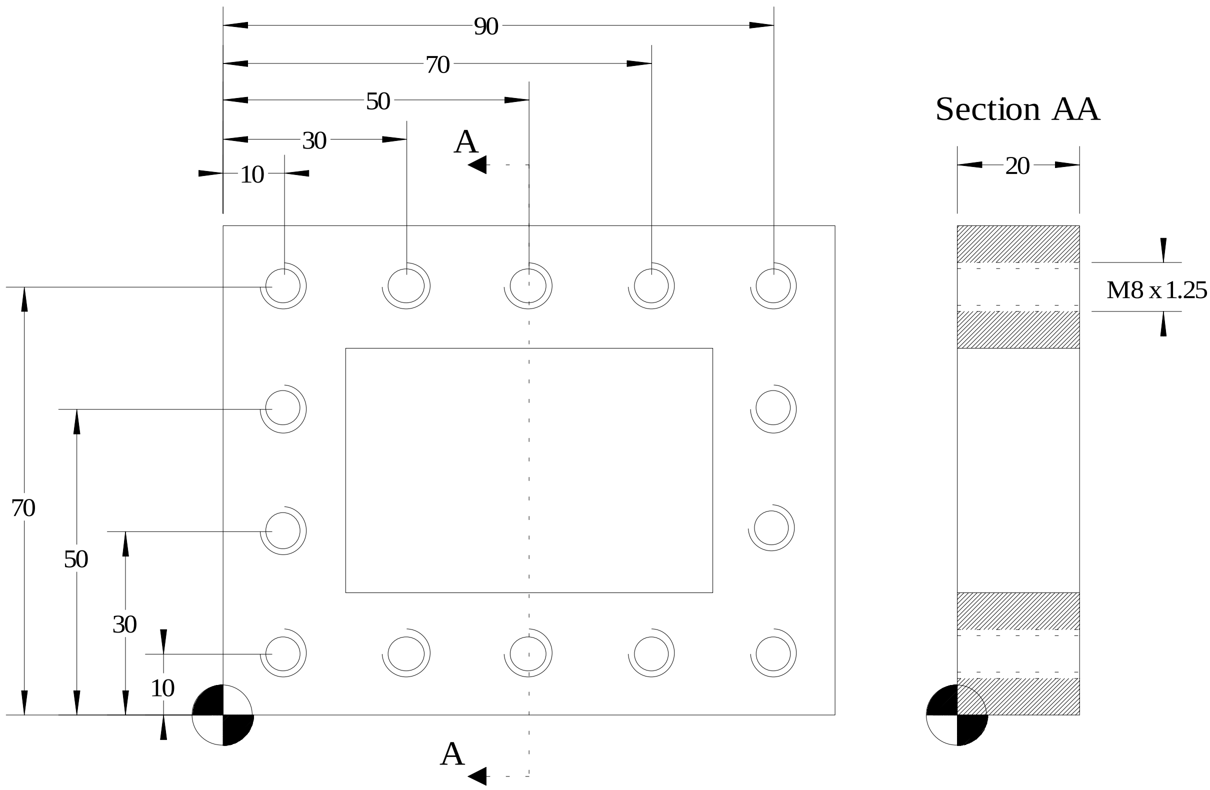
1 : T1 M6 ;
N2 G0 G90 G40 G71 G17 G94
N3 X10 Y10 Z100 H1 S? M3
N4 Z5
N5 G81 R0 Z-20 F? M8
N6 (CLS, "HOLES")
N7 G0 G90 Z100
8 : T2 M6
N9 G0 G90 G40 G71 G17 G94
N10 X10 Y10 Z100 H1 S? M3
N11 Z5
N12 G84.1 G95 R0 Z-20 F1.25 M8
N13 (CLS, "HOLES")
N14 G0 G90 Z100
N15 M30

Sub-program name = HOLES

N101 Y30
N102 Y50
N103 Y70
N104 X30
N105 X50
N106 X70
N107 X90
N108 Y50
N109 Y30
N110 Y10
N111 X70
N112 X50
N113 X30
N114 G80
N115 (ENS) ; back to N6 & N13

Sub-Programming

"Sub-Program stored separately in library as ID"



NC Part Programs			
Program Name	ID	Size	Modified
HOLES	27	237	2000/11/27 17:01:54
MAIN1	0	377	2000/11/30 12:41:40

(Program Name = MAIN1)

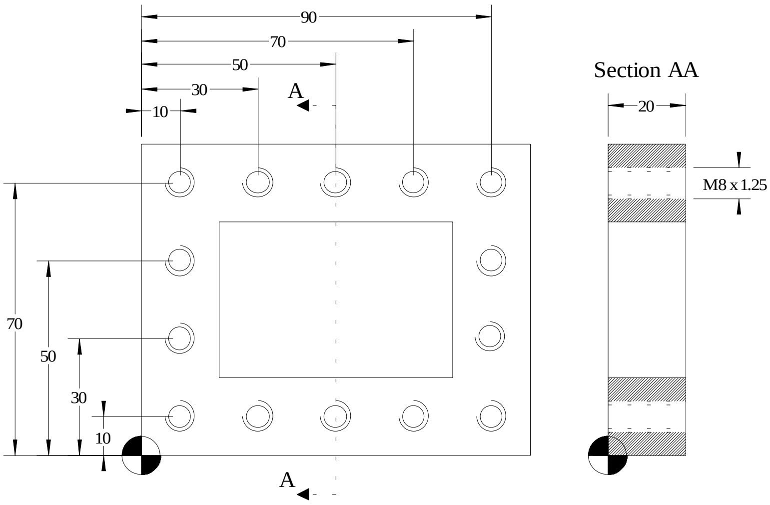
```
1 : T1 M6  ;
N2 G0 G90 G40 G71 G17 G94
N3 X10 Y10 Z100 H1 S? M3
N4 Z5
N5 G81 R0 Z-20 F? M8
N6 (CLS, 27)
N7 G0 G90 Z100
8 : T2 M6
N9 G0 G90 G40 G71 G17 G94
N10 X10 Y10 Z100 H1 S? M3
N11 Z5
N12 G84.1 G95 R0 Z-20 F1.25 M8
N13 (CLS, 27)
N14 G0 G90 Z100
N15 M30
```

Sub-program ID = 27

```
N101 Y30
N102 Y50
N103 Y70
N104 X30
N105 X50
N106 X70
N107 X90
N108 Y50
N109 Y30
N110 Y10
N111 X70
N112 X50
N113 X30
N114 G80
N115 (ENS) ; back to N6 & N13
```

Sub-Programming

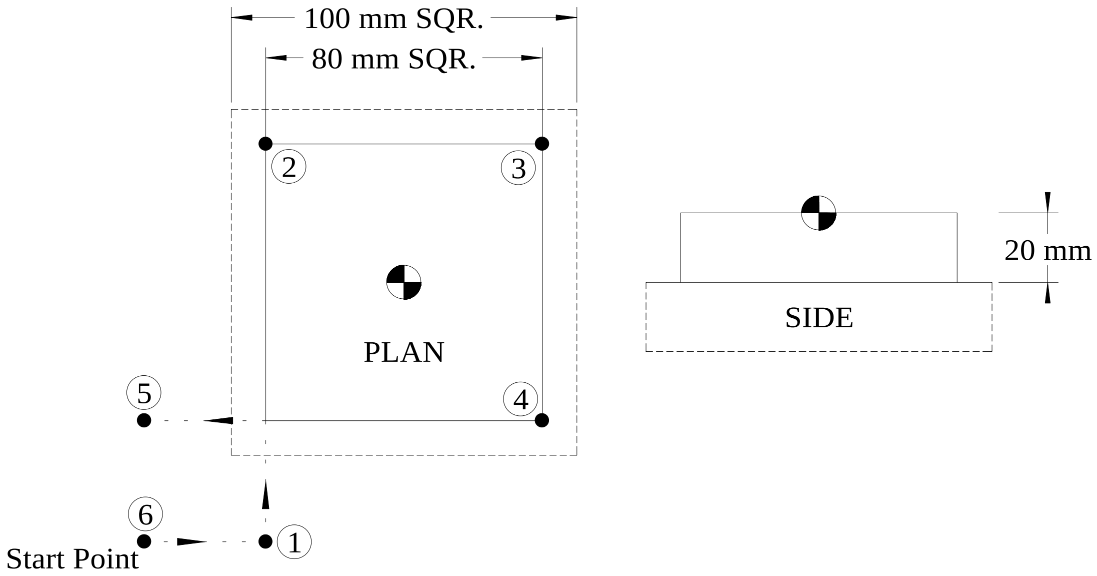
"Sub-Program stored inside the main program as name"



(Program Name = MAIN1)

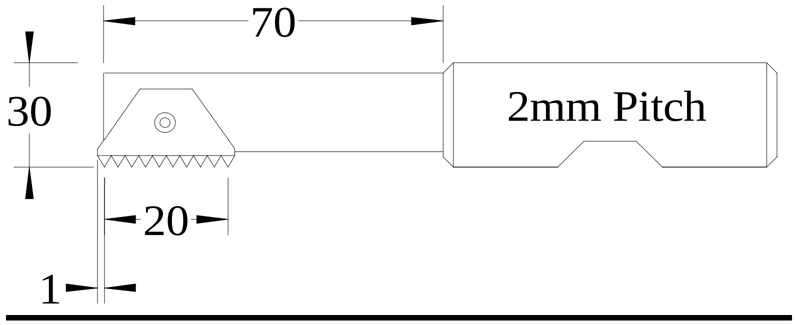
```
1: T1 M6  ;
N2 G0 G90 G40 G71 G17 G94
N3 X10 Y10 Z100 H1 S? M3
N4 Z5
N5 G81 R0 Z-20 F? M8
(DFS, "HOLES")
N6 Y30
N7 Y50
N8 Y70
N9 X30
N10 X50
N11 X70
N12 X90
N13 Y50
N14 Y30
N15 Y10
N16 X70
N17 X50
N18 X30
N19 G80
N20 (ENS) ; back to N21 & N28
N21 (CLS, "HOLES")
N22 G0 G90 Z100
23 : T2 M6
N24 G0 G90 G40 G71 G17 G94
N25 X10 Y10 Z100 H1 S? M3
N26 Z5
N27 G84.1 G95 R0 Z-20 F1.25 M8
N28 (CLS, "HOLES")
N29 G0 G90 Z100
N30 M30
```

Sub-Programming
"Contour pecking"




```
: T? M6
G0 G90 G40 G71 G17 G94
X-75 Y-75 Z100 H1 S? M3
Z5
G1 Z0 F? (Feed to surface top) ;
(DFS,"MILL SQUARE")
G1 G91 Z-2 ;Incremental peck depth
G90 G41 X-40 M8 ;Move to position 1 with comp. - Switch on coolant
Y40 ;Move to position 2
X40 ;Move to position 3
Y-40 ;Move to position 4
X-75 ;Move to position 5 - Clear of material - cutter diameter
G40 Y-75 ;Cancel compensation
(ENS)
(CLS, "MILL SQUARE",10) (Call sub-program & repeat 10 times) ;
G0 G90 Z100 (Move to Safe height above material.) ;
M30 (End program)
```

Helical Milling
"Multi toothed Tools"
Absolute & Incremental Programming



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

: T1 M6 ;Tool change line
G0 G90 G40 G71 G17 G94 ;Safety Line.
X0 Y0 Z100 H? S? M3 ;Move to centreline of bore
Z5 ;Move to feed clearance
G1 Z-61.5 F? ;Feed to dpth + ¼ of pitch for arcing ON + ½ tooth form width to tool end
G91 Y10 ;Move to Arc On bore centre 30(AR) – 20(SR) = 10(YD)
(DFS, "HELIX")
G41 G91 X20 ;Apply compensation as a straight line
G3 X-20 Y20 Z0.5 P20 K2 ;This line creates a ¼ arc + Z movement of a ¼ pitch
X0 Y0 Z2 I0 J-30 K2 ;This line will create 1 pitch.
X-20 Y-20 Z0.5 P20 K2 ;Creates a ¼ arc + Z movement of a ¼ of pitch
G1 G40 X20 ;Cancel Compensation as a straight line
(ENS)
(CLS, "HELIX")
G0 G90 Z-41.5 ;Subtract edge length from 1st Z positioning move 61.5 – 20 = 41.5
(CLS, "HELIX")
G0 G90 Z-21.5 ;Subtract edge length from 2nd Z positioning move 41.5 – 20 = 21.5
(CLS, "HELIX")
G0 G90 Z100 ;Clear the workpiece
M30 ;End the Program.

CHAPTER 11

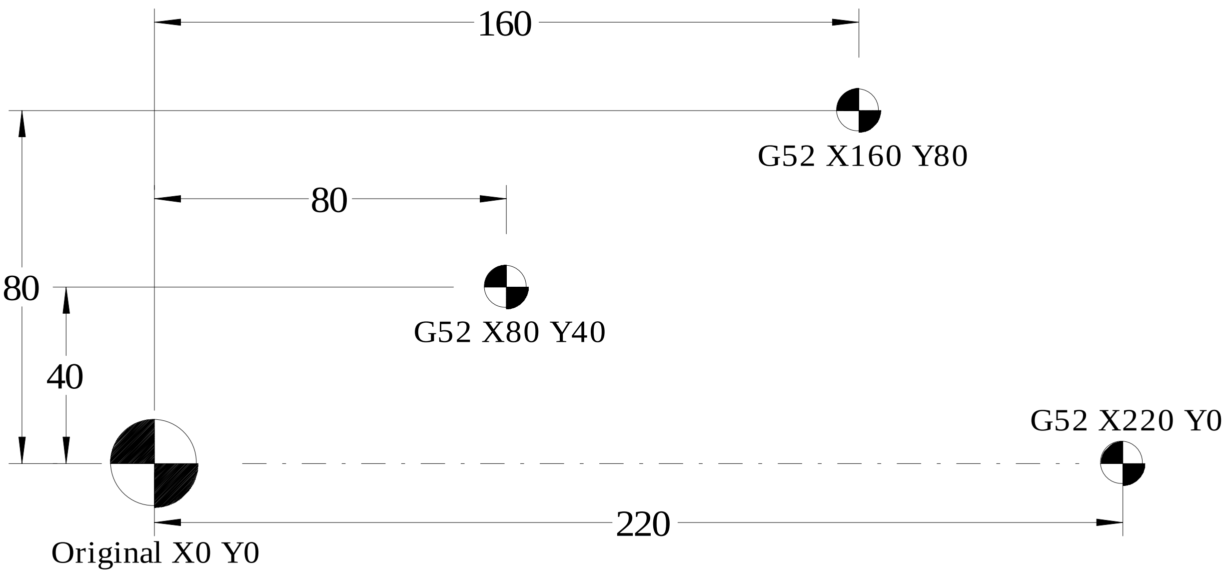
Datum Shift

Datum Shift

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

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

This code cannot be used Incrementally



The datum shift must be cancelled when finished with.

The following are ways to achieve this:

1) Programming G52 with zero axis motion
2) Data reset
3) End of program
4) Machine power off

Rotation

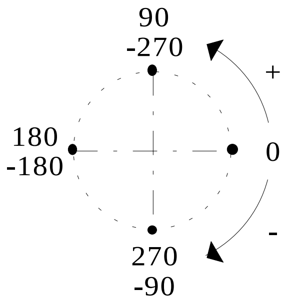
Rotation

A programmed shape can be rotated around a programmed pole position. Programming an **ROT,** block with an axis position for the centre of rotation together with the angle of rotation all contained within ( ) will do this.

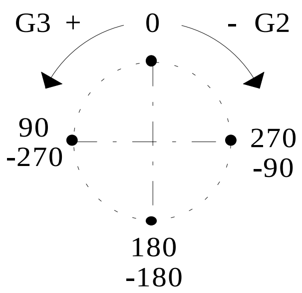
(ROT, X? Y? A? G?)

Where:
X? & Y? = Absolute centre of rotation
A? = Angle of rotation around the current plane (A+ = Anti-clockwise)
G? = On/off code to denote if angle is absolute or incremental (G1 = Abs. G0 = Inc.)

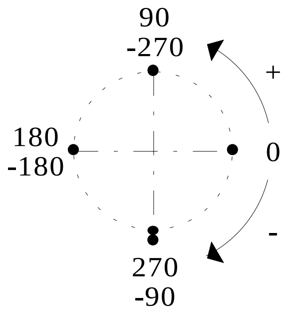
All information programmed after this line of command will instruct the control to calculate the new axis position at the displaced angle.
The smallest increment of angular displacement = 0.001 Deg.



G17
XY Plane (Z-)
Plan View



G18
XZ Plane (Y-)
Front View



G19
YZ Plane (X-)
Side View

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

Incremental can be used with rotation. The "A" is established as incremental values if (ROT, A? G?) is specified with a G0 code. An angle must have been established first so an incremental angle has a base to work from.

The Rotation must be cancelled when finished with.

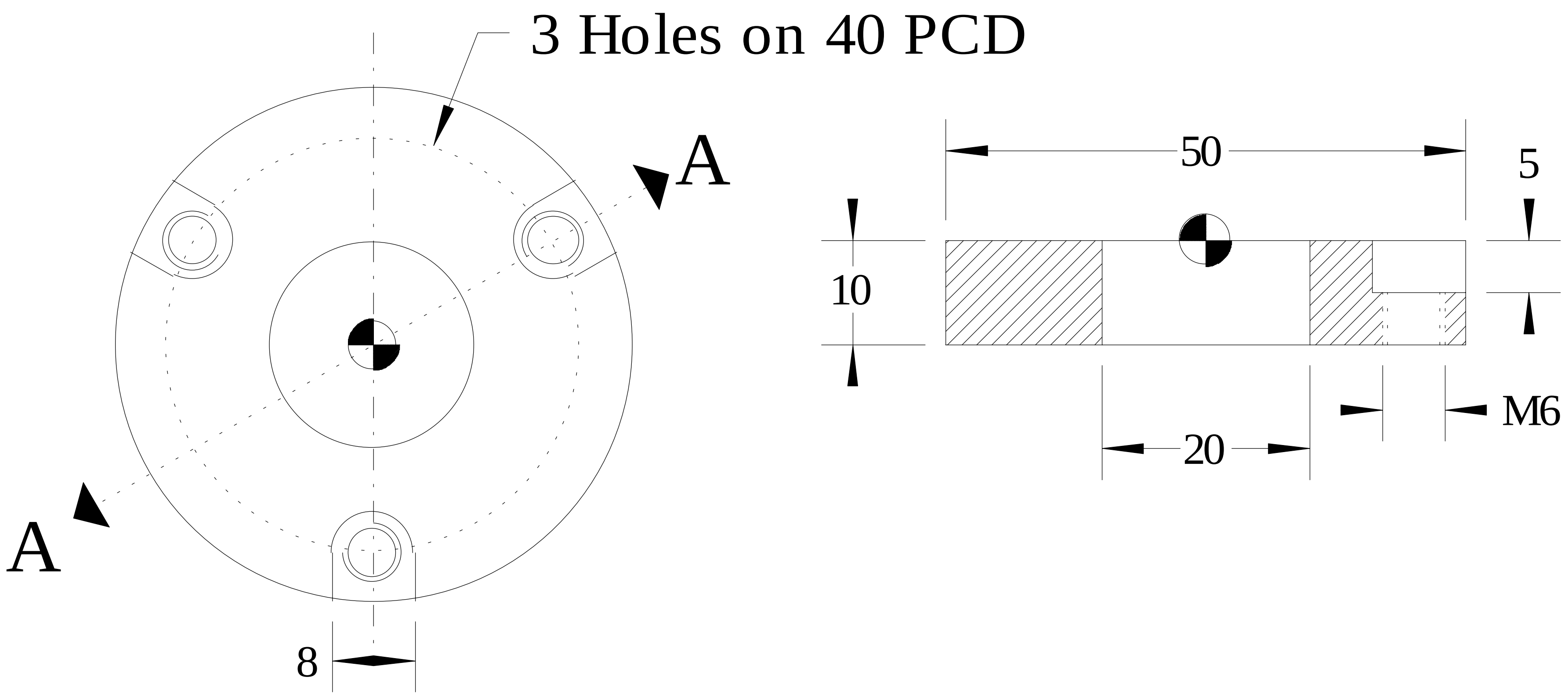
The following are ways to achieve this:

1) Programming (ROT, A0)
2) Data reset or Colon block (If "System Config.", "Machine Application" is configured to do so)
3) End of program
4) Machine power off

NOTE

Always cancel rotation at the program X0 Y0 zero position

Rotation

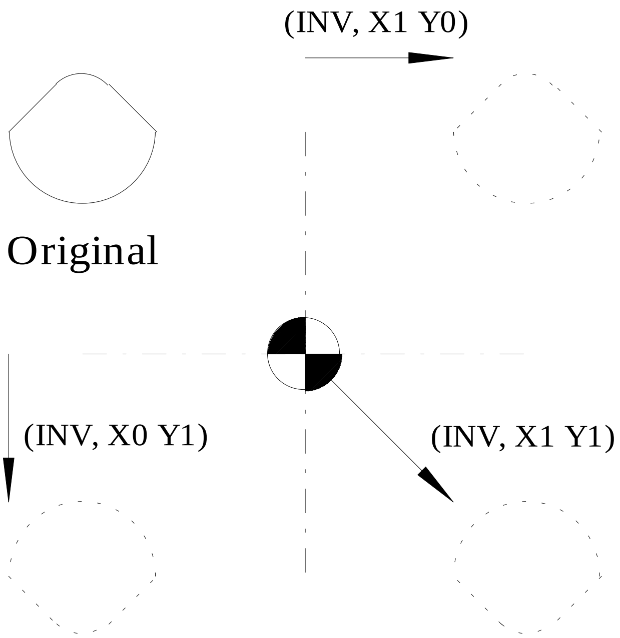


Rotation

: T1 M6 ; 6mm endmill
G0 G90 G40 G71 G17 G94
X0 Y-30 Z100 H1 S? M3
Z5
(DFS, “SLOT”)
X0 Y-30
G1 Z-5 F? M8
G41 X4
Y-20
G3 X-4 Y-20 P4
G1 Y-30
G40 X0
G0 Z5
(ROT, X0 Y0 A120 G0)
(ENS)
(CLS, “SLOT”,3)
G0 G90 X0 Y0
(ROT, A0)
G0 G90 Z100 M1
: T2 M6 ; 5mm drill
G0 G90 G40 G71 G17 G94
X0 Y-20 Z100 H1 S? M3
Z5
G81 R-5 Z-10 W10 F? M8
(DFS, “PCD”)
(ROT, X0 Y0 A120 G0)
X0 Y-20
(ENS)
(CLS, “PCD”,2)
G0 G90 X0 Y0
(ROT, A0)
G0 G90 Z100 M1
: T3 M6 ; 6mm tap
G0 G90 G40 G71 G17 G94
X0 Y-20 Z100 H1 S? M3
Z5
G84.1 G95 R-5 Z-10 W10 F1 M8
(CLS, “PCD”,2)
G0 G90 X0 Y0
(ROT, A0)
G0 G90 Z100 M30

Programmable Mirror Image

Mirror Image



A program shape can be mirrored over a current datum position
Programming an INV**,** block with information to inform the control which axis is to be used for mirror image all contained within ( ) will do this.

(INV, X? Y?)

Where:
X? & Y? = Inform the control which axis is to be switched on for mirror image.
1 = On
0 = Off

Mirror Code

(INV, X0 Y0)	Cancel Mirror Image	X+100 Y+100 = X+100 Y+100
(INV, X1 Y0)	Mirror image all X axis motion	X+100 Y+100 = X-100 Y+100
(INV, X0 Y1)	Mirror image all Y axis motion	X+100 Y+100 = X+100 Y-100
(INV, X1 Y1)	Mirror image all X & Y axis motion	X+100 Y+100 = X-100 Y-100

Mirror Image must be cancelled when finished with.
The following are ways to achieve this:

1) Programming (INV, X0 Y0)
2) End of program
3) Machine power off

P.C.D.

&

Line/Grid

PCD & GRID cycle formats

Rectangular Grid Cycle

G38 I_____ U_____ ;minimum required

J_____ V_____ O_____ R_____ S_____ W_____ ;additional

"CYCLE LINE"

G37

PCD Pattern Cycle

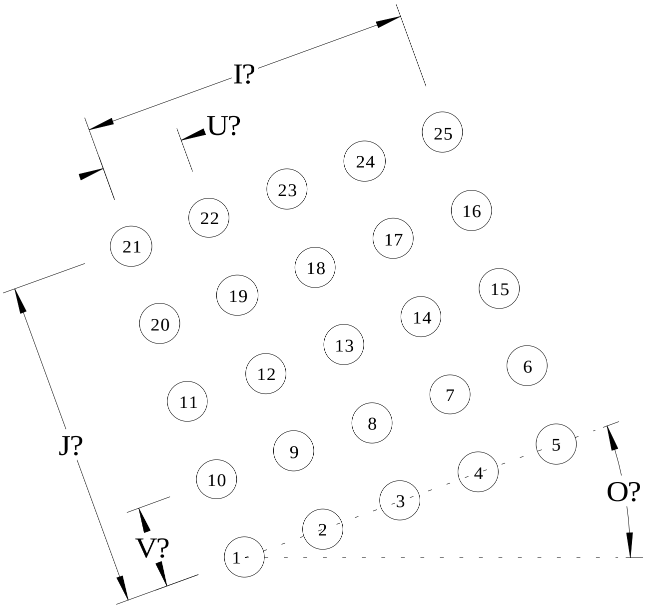
G39 D_____ P_____ K_____ ;minimum required

X_____ Y_____ O_____ I_____ J_____ R_____ S_____ W_____ ;additional

"CYCLE LINE"

G37

Grid/Line Pattern Cycle
G38



Program movement to first pattern position must be established before the G38 cycle

G38 I? U? J? V? [O? R? W?]

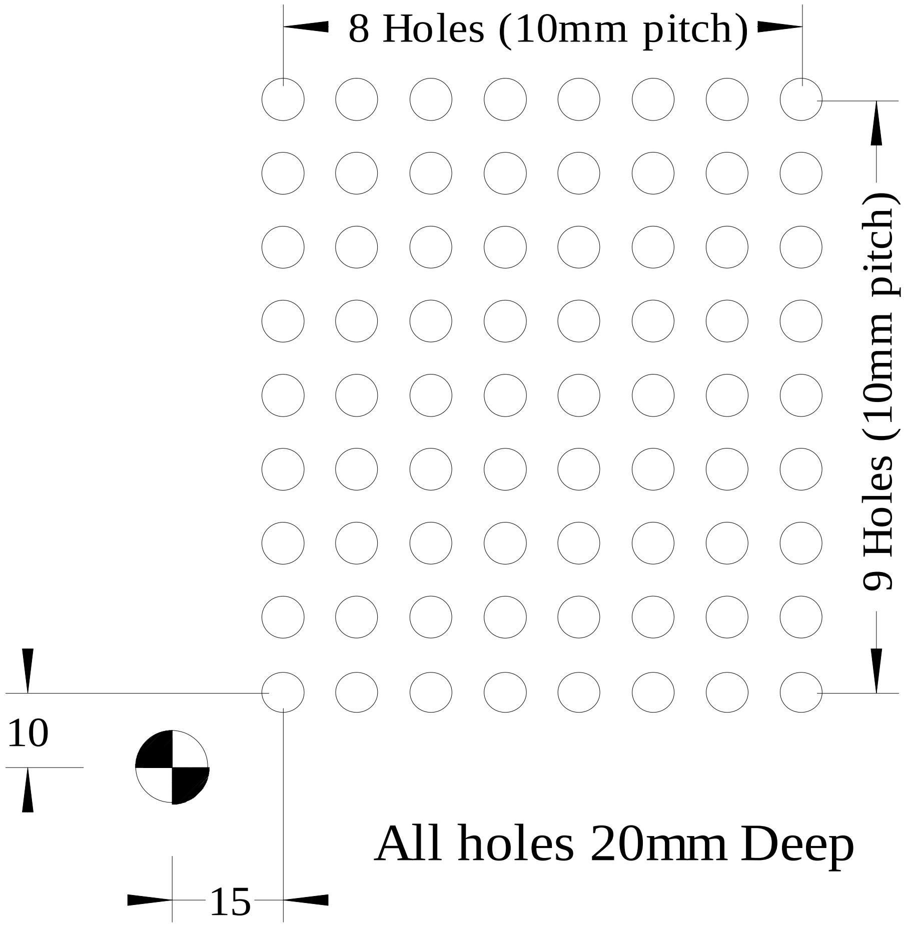
[ ] Denotes optional inputs
Where:
G38 = Modal cycle code
I? = Modal number of operations per line (No. of columns).
U? = Modal incremental spacing between the operations (Pitch between columns).
J? = Modal number of lines of operations (No. of rows) only required with a grid.
V? = Modal incremental space between lines (Pitch between rows) only required with a grid..
O? = Modal angle of rotation of pattern from reference point (Not required for 0).
R? = Pattern rotates with angle (0 = yes / 1 = No)
W? = Non-modal additional retract at pattern end.

G38 MUST BE CANCELLED ON A SEPARATE PROGRAM LINE WITH A G37 WHEN FINISHED (see format sheet).

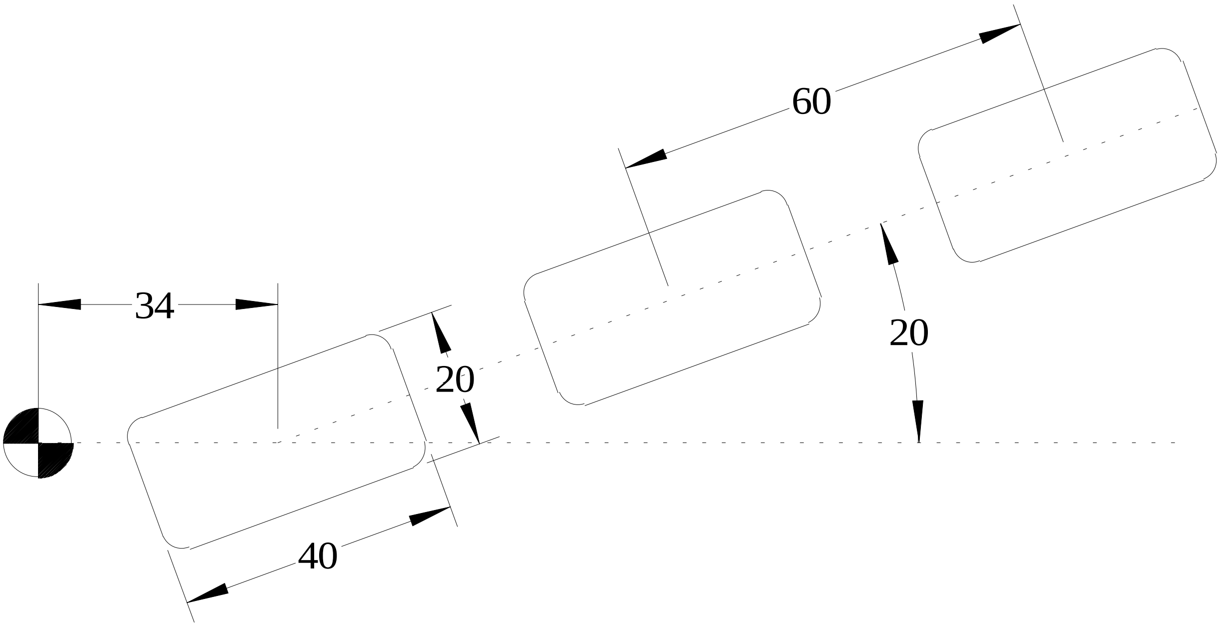
Specifying the U & V as signed values the program can start at any corner of the grid and proceed using the sequences below.

Reference Corner	Sign	Sequence
1	U+ V+	1-5, 6-10, 11-15, 16-20, 21-25
5	U- V+	5-1, 10-6, 15-11, 20-16, 25-21
21	U+ V-	21-25, 16-20, 11-15, 6-10, 1-5
25	U- V-	25-21, 20-16, 15-11, 10-6, 5-1

Grid Pattern Cycle
G38

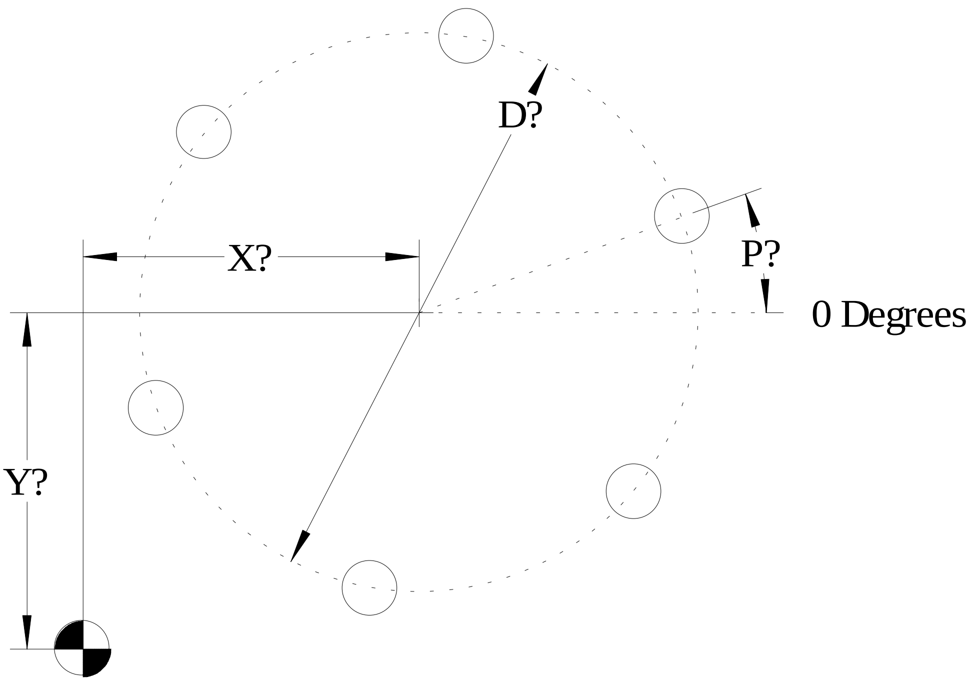


Line Pattern Cycle
G38




```
:T1 M6
G0 G90 G40 G71 G17 G94
X34 Y0 Z100 H? S? M3
Z5 F200
G38 I3 U? O20
G23 U40 V20 R? Z-? Q? L? E? K? ,R? ,D? P? I? J? F? S? M8
G37
G0 G90 Z100 M30
```

Circular (P.C.D.) Pattern Cycle
G39



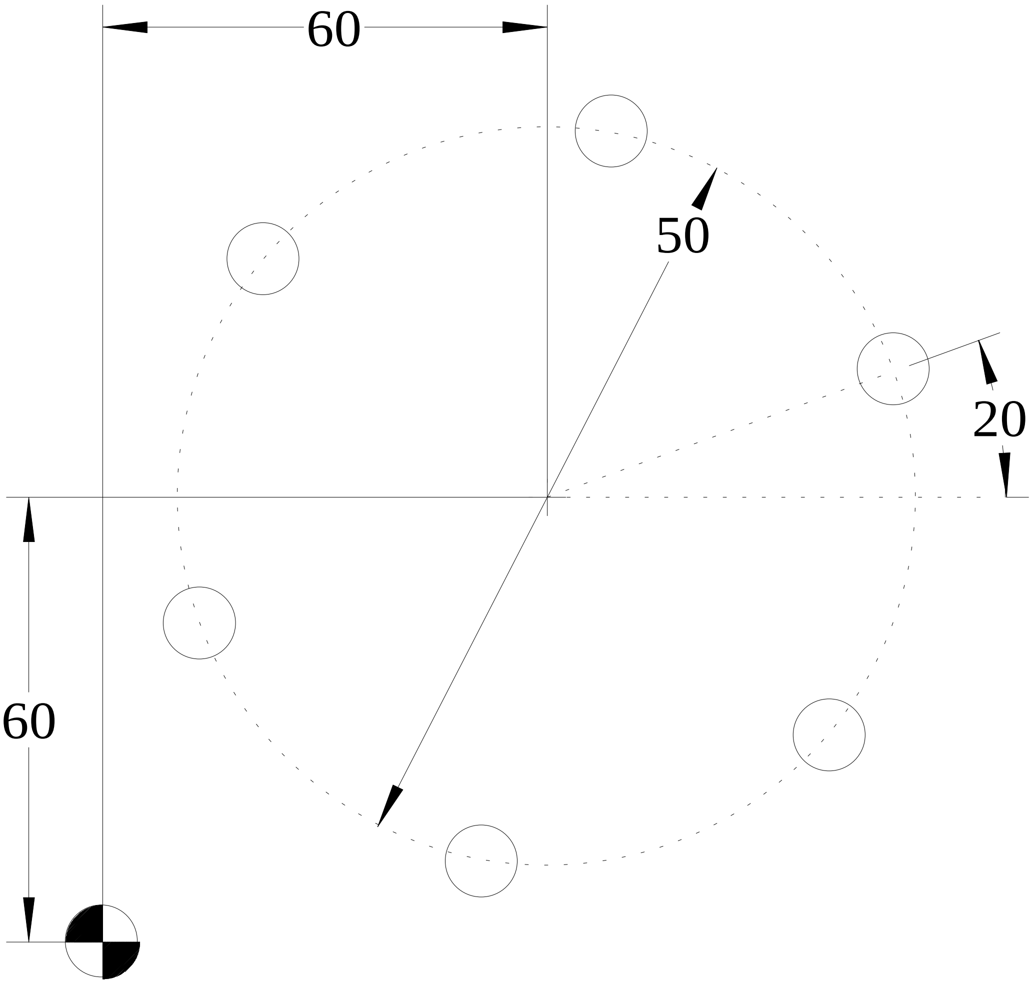
Program movement to first pattern position is not required. If X & Y are not programmed then the current tool position becomes the P.C.D. centre.

G39 [X? Y?] D? P? K? [O? R? W?]

[ ] Denotes optional inputs
Where:
G39 = Modal cycle code
X? = Modal X axis co-ordinate to centre of PCD.
Y? = Modal Y axis co-ordinate to centre of PCD.
D? = Modal PCD (Pitch Circle Diameter).
P? = Modal 1st pattern angle location from zero Degrees.
O? = Modal angle between 1st & last pattern (Required only for Arcs).
R? = Pattern rotates with angle (0 = yes / 1 = No)
K? = Number of patterns.
W? = Non-modal additional retract at pattern end.

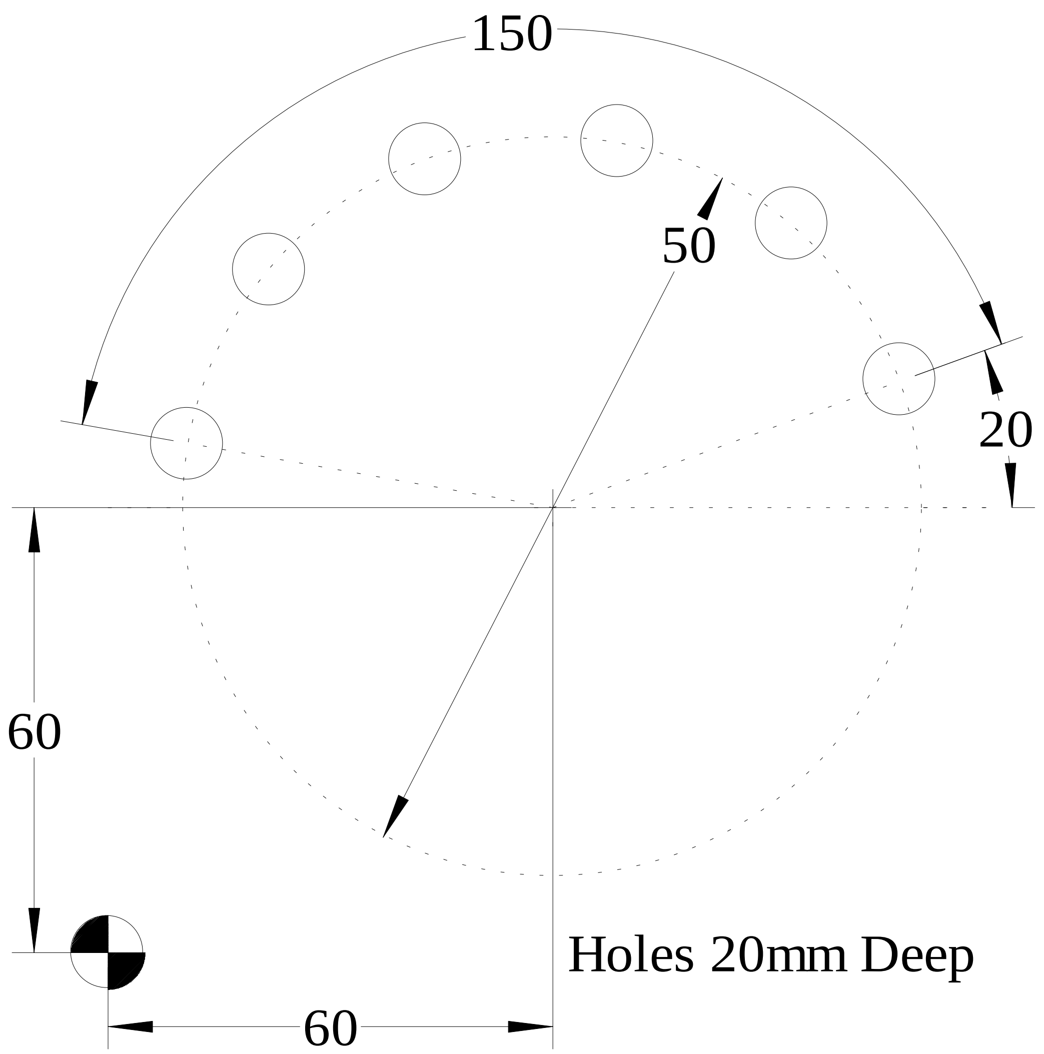
G39 MUST BE CANCELLED ON A SEPARATE PROGRAM LINE WITH A G37 WHEN FINISHED (see format sheet).

P.C.D. Pattern Cycle
G39



Holes 20mm Deep

P.C.D. Arc Pattern Cycle
G39

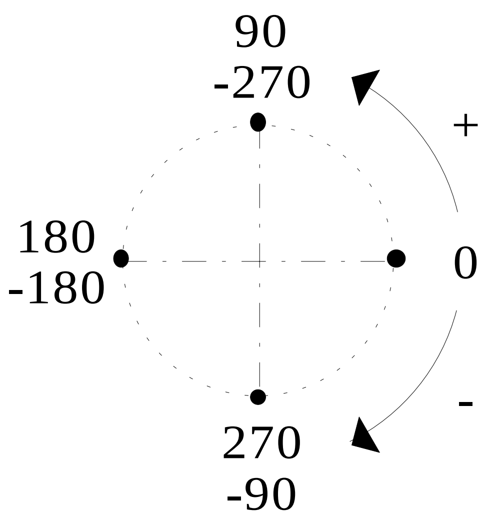


: T1 M6
G0 G90 G40 G71 G17 G94
X60 Y60 Z100 H? S? M3
Z5
G39 U50 P20 O150 K6
G81 R? Z-20 F? M8
G37
G0 G90 Z100 M30

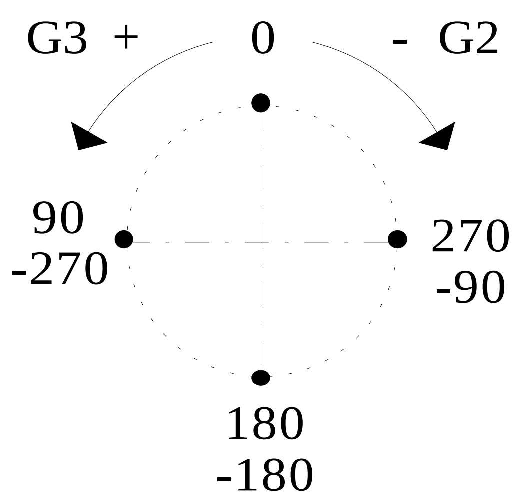
CHAPTER 12

Polar Co-ordinates

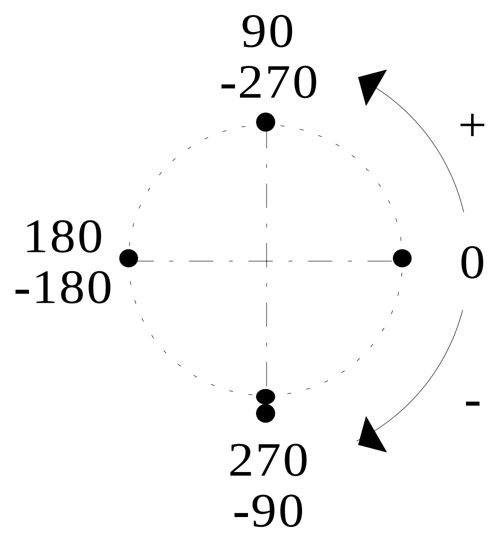
Working Planes



G17
XY Plane (Z-)
Plan View

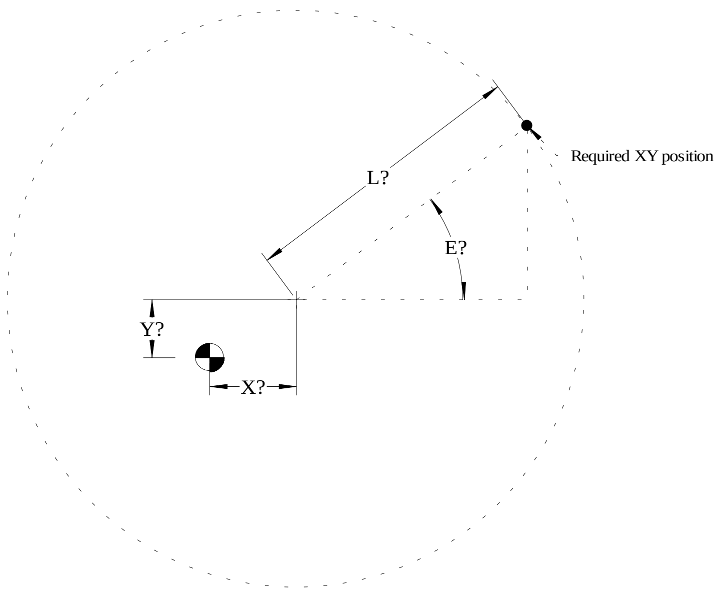


G18
XZ Plane (Y-)
Front View



G19
YZ Plane (X-)
Side View

Polar Co-ordinate Command



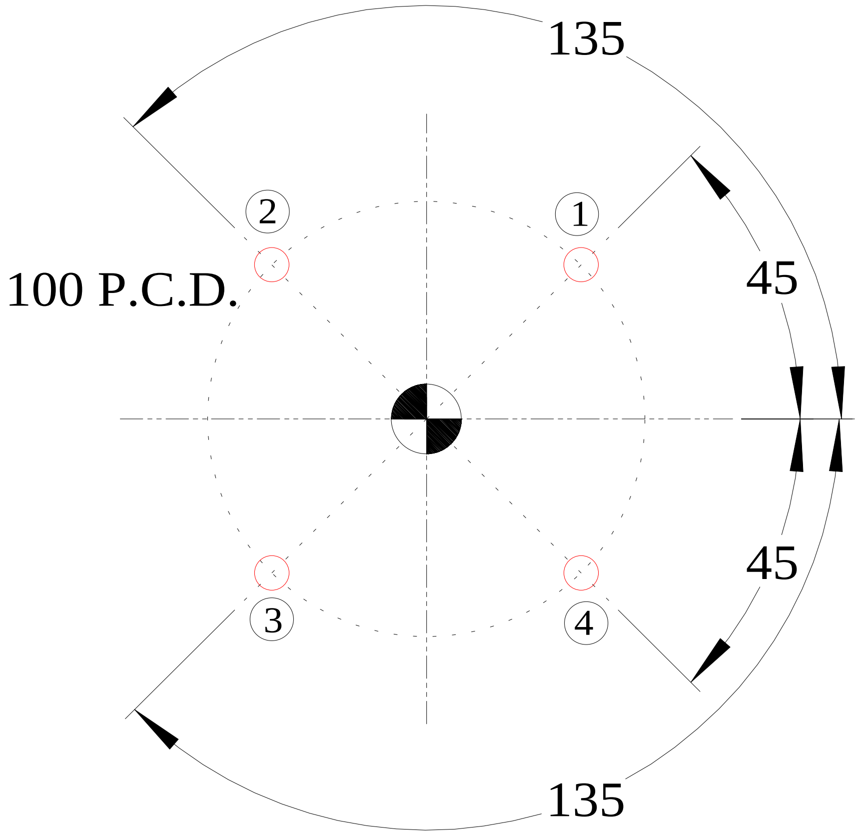
The control has the ability to position itself to an endpoint in any plane using a **G15.1** code with only information regarding the length of the move from the PCD centre point with the angle of the line as that of a right angled triangle.
All angles are relative to the current plane.

G15.1 X? Y? L? E?
Where:
G15.1 = PCD Polar command
X? = Absolute X axis position of PCD centre
Y? = Absolute Y axis position of PCD centre
L? = Hypotenuse of the right angle triangle
E? = Angle of the hypotenuse line from the 0 Deg. point of the plane.

It is not possible to use G91 incremental to control the angle.
If the X & Y positions to the PCD centre are omitted, then the current tool position will become the pole PCD centre.

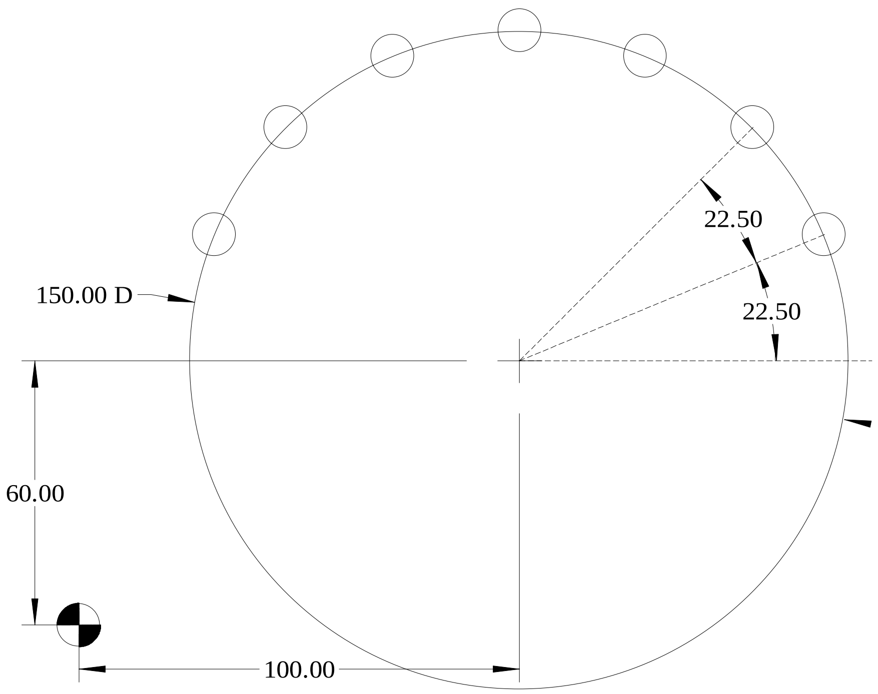
G15.1 PCD Polar command is a code from group "Polar Program" and is only active with the use of the L & E words on the line of program.
G15.1 is set as a machine default at power on.

Polar Co-ordinate Command – e.g. 1

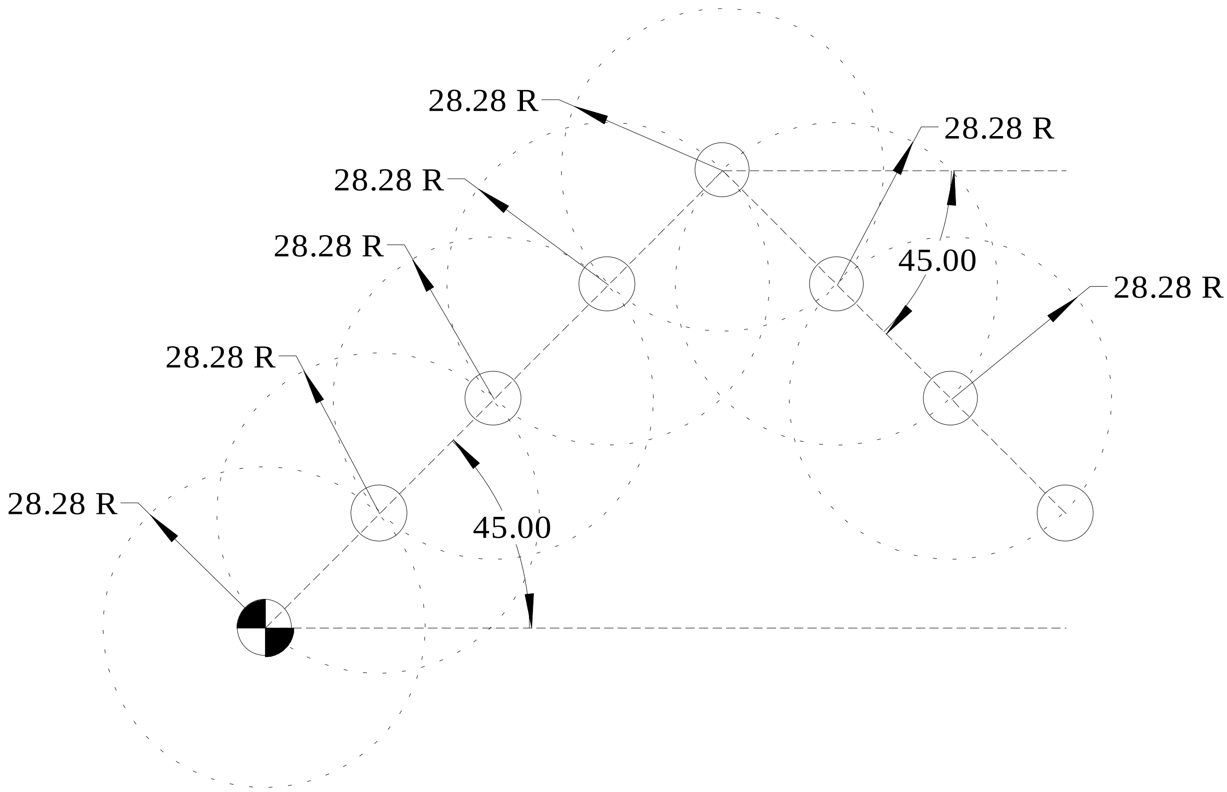


(Rapid to point 1) G0 G15.1	
(Rapid to point 2)	
(Rapid to point 3)	
(Rapid to point 4)	

Polar Co-ordinate Command – e.g. 2

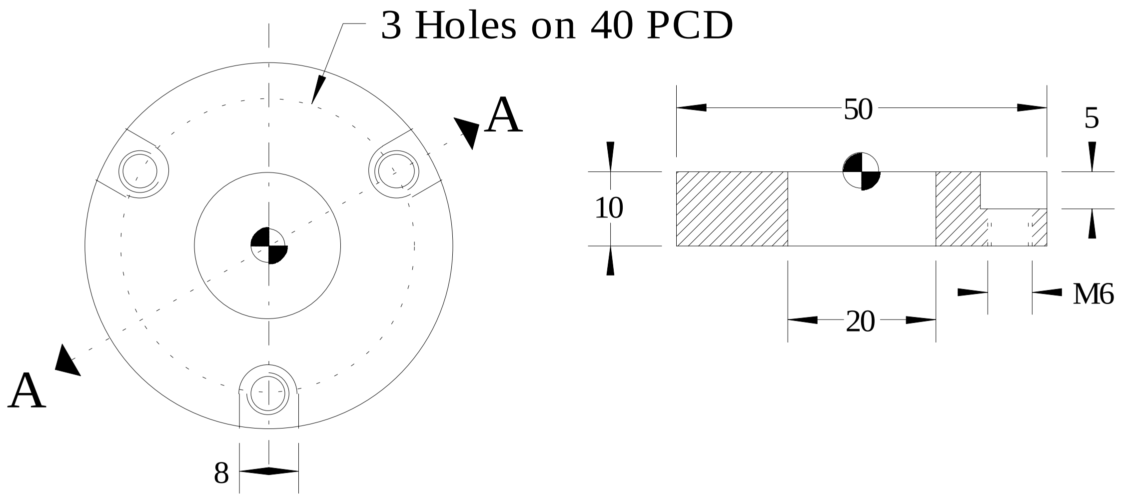


Polar Co-ordinate Command – e.g. 3



Polar Co-ordinate Command

Hole position Example



: T? M6
G0 G90 G40 G71 G17 G94
G15.1 X0 Y0 Z100 L20 E30 H? S? M3
Z5
G81 R-5 Z-10 F? W10 M8
X0 Y0 L20 E150 W10
X0 Y0 L20 E270 W10
G0 G90 Z100 M1
: T? M6
G0 G90 G40 G71 G17 G94
G15.1 X0 Y0 Z100 L20 E30 H? S? M3
Z5
G84.1 R-5 Z-10 F? W10 M8
X0 Y0 L20 E150 W10
X0 Y0 L20 E270 W10
G0 G90 Z100 M30

Programmable Tool Offsets

Programmable Tool Offset "O" word

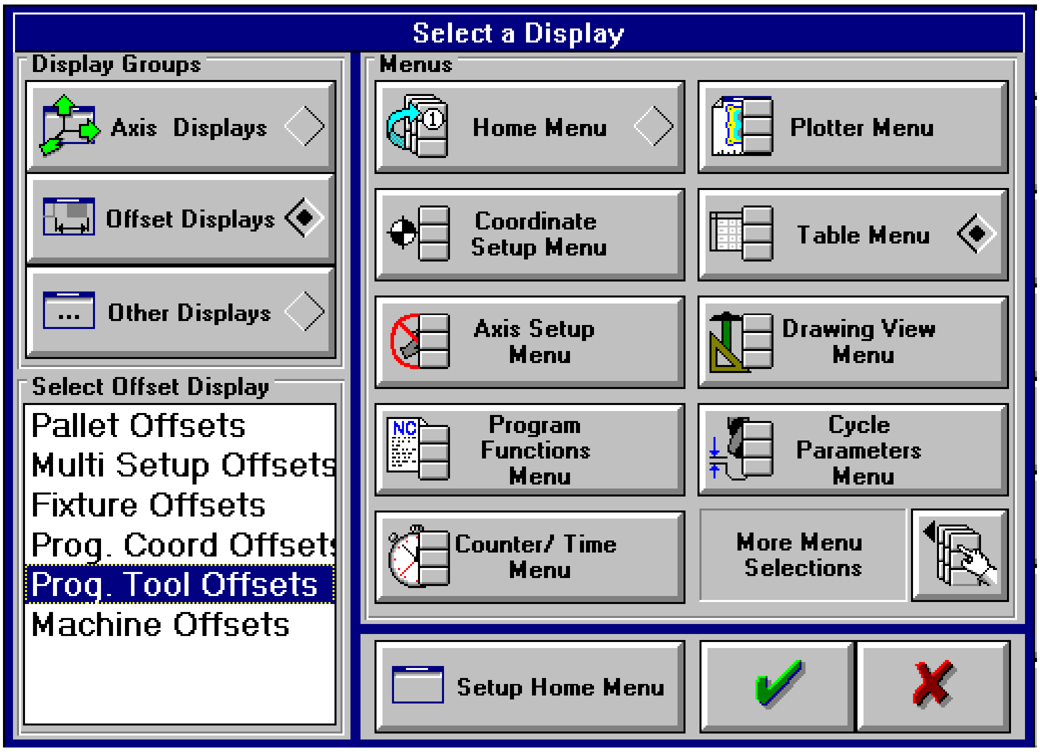
The "Programmable Tool Offset" is used if the same tool is being used for roughing and finishing operations.

The "Programmable Tool Offset" table contains a value which is "**Incremental"** to the stored tool values and does not require a tool change to activate.

The value can be stored by operator or by program.

The use of the non-zero "O" word programmed on a G41/G42 axis compensation move activates the "Diameter" column.

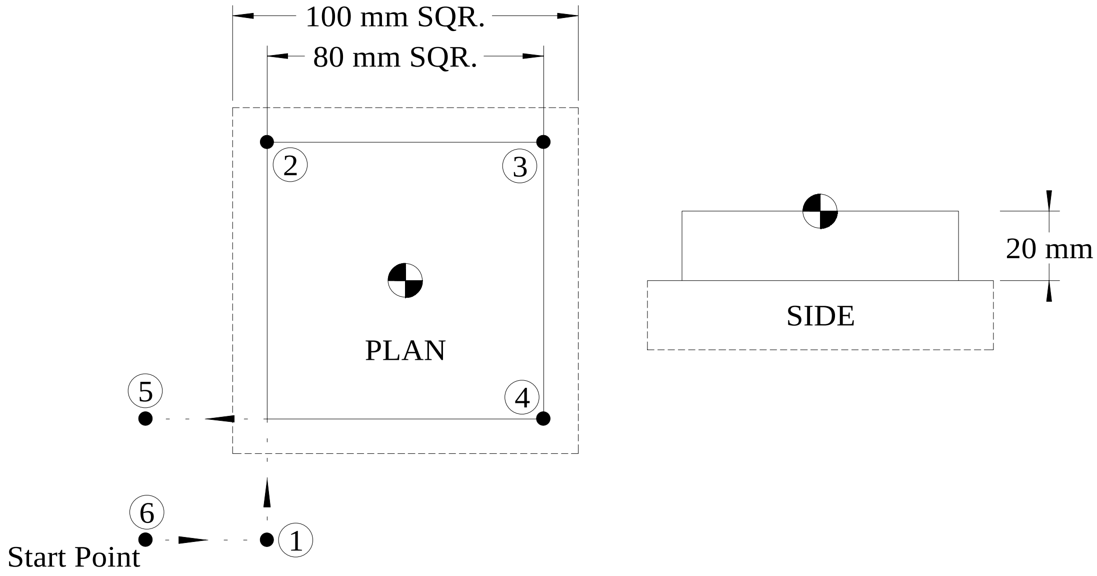
The use of a zero "O" word programmed on a linear axis move cancels the offset.



	Diameter	Length
1	+0.0000	+0.0000
2	+0.0000	+0.0000
3	+0.5000	+0.0000
4	+0.0000	+0.0000
5	+0.0000	+0.0000
6	+0.0000	+0.0000
7	+0.0000	+0.0000
8	+0.0000	+0.0000
9	+0.0000	+0.0000
10	+0.0000	+0.0000

The "Programmable Tool Offset" can store 99 total offsets.

Programmable Tool Offset "O" word – e.g.



:T? M6
G0 G90 G40 G71 G17 G94
[$PROG_OFFSET(3)DIAMETER]=0.5 ; Set Programmable offset - DIAMETER
X-75 Y-75 Z100 H1 S? M3
Z5
(DFS "SQUARE")
G1 Z-20 F?
G41 X-40 **O3** M8 ; Move to position 1 - Apply compensation using "O" word
Y40
X40
Y-40
X-75
G40 Y-75
(ENS)
(CLS, "SQUARE")
[$PROG_OFFSET(3)DIAMETER]=0 ; Reset Programmable offset - DIAMETER
(CLS, "SQUARE")
G0 G90 Z100
M30

Programmable Data Entry

Programmable Data Entry

System Variables – [$???( )?]

The $ symbol is used as part of the computers pre-set registers. It allows data to be either written to or read from all areas of the A2100 control.

i.e..
[$FIXTURE(1)X] = 100 ; Will store 100 in the H1 work offset if written in the program

The types of transferable data can include information to the Tool table, Offset tables, Process control table. Information can also be read from areas such as the System registers and information saved to the process control data for calculation use.

System Registers ($)

[$PALLET(*Row No.*)*Field*]

(1 Table = 1 row only)

Pallet Data	***Field***	***Value***
X axis offset	**X**	Numerical (99999.9999mm)
Y axis offset	**Y**	Numerical (99999.9999mm)
Z axis offset	**Z**	Numerical (99999.9999mm)
A axis offset	**A**	Numerical (359.9999 degrees)
B axis offset	**B**	Numerical (359.9999 degrees)
Offsets Rotate	**ROTATES**	1 = YES / 0 = NO
Rotary Position	**ROTARY_POS**	Numerical (359.9999 degrees)
Pallet State	**STATE**	0 = Absent / 1 = Present 2 = Last / 3 = New
Pallet Status	**STATUS**	0 = Pending / 1 = Started 2 = Aborted / 3 = Complete 4 = Setup Aborted
Pallet Order	**ORDER**	0 to 50
Pallet Identifier	**PALLET_ID**	0 to 9999
Pallet Location	**LOCATION**	0 to 9999

[$SETUP(*Row No.*)*Field*]

(64 Offsets -Row number is related "Active setup number" (Home Page))

Setup Data	***Field***	***Value***
X axis offset	**X**	Numerical (99999.9999mm)
Y axis offset	**Y**	Numerical (99999.9999mm)
Z axis offset	**Z**	Numerical (99999.9999mm)
A axis offset	**A**	Numerical (359.9999 degrees)
B axis offset	**B**	Numerical (359.9999 degrees)
Part State	**SETUP_STATE**	0 = Absent / 1 = Active 2 = Last / 3 = New
Part Status	**PART_STATUS**	0 = Pending / 1 = Started 2 = Aborted / 3 = Complete 4 = Setup Aborted
NC Program ID	**NC_PROG_ID**	0 to 9999

[$FIXTURE(*Row No.*)*Field*]

(32 "H" Offsets -Row number is related "Active setup number" (Home Page))

Fixture Data	***Field***	***Value***
X axis offset	**X**	Numerical (99999.9999mm)
Y axis offset	**Y**	Numerical (99999.9999mm)
Z axis offset	**Z**	Numerical (99999.9999mm)
Offsets Rotate	**ROTATES**	1 = YES / 0 = NO
Rotary Position	**ROTARY_POS**	Numerical (359.9999 degrees)

[$PROG_OFFSET(*Row No.*)*Field*]

(32 "D" Offsets -Row number is related "Active setup number" (Home Page))

Trim Data	***Field***	***Value***
X axis offset	**X**	Numerical (99999.9999mm)
Y axis offset	**Y**	Numerical (99999.9999mm)
Z axis offset	**Z**	Numerical (99999.9999mm)

[$TOOL_OFFSET(*Row No.*)*Field*]

(Used with "O" word)
(Row number is maximum 99)
Values are always incremental values to the tool table values.

Programmable Tool Data	***Field***	***Value***
Length	**LENGTH**	Numerical (99999.9999mm)
CDC Value	**DIAMETER**	Numerical (99999.9999mm)

[$MACH_OFFSET(*Row No.*)*Field*]

(Used with "D" word & G98,G98.1)

Machine offset Data	***Field***	***Value***
X axis offset	**X**	Numerical (99999.9999mm)
Y axis offset	**Y**	Numerical (99999.9999mm)
Z axis offset	**Z**	Numerical (99999.9999mm)

[$PROCESS_DATA(*Row No.*)*Field*]

(Data Storage Table)

Process Data	***Field***	***Value***
X field	**X**	Numerical (99999.9999mm)
Y field	**Y**	Numerical (99999.9999mm)
Z field	**Z**	Numerical (99999.9999mm)
I field	**I**	Numerical (99999.9999mm)
J field	**J**	Numerical (99999.9999mm)
K field	**K**	Numerical (99999.9999mm)
A field	**A**	Numerical (99999.9999mm)
B field	**B**	Numerical (99999.9999mm)
C field	**C**	Numerical (99999.9999mm)

[$CYCLE_PARAMS(*Column*)*Field*]

(Cycle Parameters Table)

Column = 2(Programmable Column)

Only 2 is allowed to be programmed. Any other value will cause an alarm

Drilling Cycle Parameters	*Field*	*Value*
Imperial value stop increment above "R"	**GAGE_HT_INCH**	0 - 99.9999 inch
Metric value stop increment above "R"	**GAGE_HT_MM**	0 - 99.9999 mm
Drilling Type (with or without drill tip)	**HOLE_DEPTH**	0 = ABS. with drill tip 1 = INC. with drill tip 2 = ABS. without drill tip 3 = INC. without drill tip
Finishing INC. depth with G82	**G82_FIN_DPTH**	0 - 999.9999mm
Finishing Feed % with G82	**G82_FEED_FAC**	0 - 999%
G82 Dwell time.	**G82_DWELL**	0 - 99.99 seconds
Amount drill remains in the hole with G83 J2 / J12	**G83_RET_DIST**	0 - 999.9999mm
Retraction with G83 J1 / J11	**G83_SHRT_RET**	0 - 999.9999mm
Rapid return to this incremental value from bottom of hole.	**G83_RELIEF**	0 - 999.9999mm
G84 Dwell time.	**G84_DWELL**	0 - 99.99 seconds
Pre-set chip breaking return revolutions.	**G84_CHIP_BRK**	0 - 999 revolutions
G86 Bottom retract distance.	**G86_BOT_RET**	0 - 999.9999mm
G87 Dwell time.	**G87_DWELL**	0 - 99.99 seconds
G87 Bottom retract distance.	**G87_BOT_RET**	0 - 999.9999mm
G87 Backbore clearance.	**G87_BK_CLR**	0 - 999.9999mm
G88 Breakthrough distance.	**G88_BRK_DIST**	0 - 999.9999mm
G89 Dwell time.	**G89_DWELL**	0 - 99.99 seconds

Milling Cycle Parameters	***Field***	***Value***
Mill cycle depth programming.	**MILL_DEPTH**	0 = Absolute depth 1 = Incremental depth
Face cycle cut width %.	**FAC_CUT_WDTH**	10% - 80%
Face cycle finish stock.	**FAC_FIN_STK**	0 - +/-9.9999mm
Face cycle XY clearance.	**FAC_XY_CLR**	0 - +/-999.9999mm
Pocket cycle cut width %.	**POC_CUT_WDTH**	10% - 80%
Pocket cycle side finish stock.	**POC_SFIN_STK**	0 - +/-9.9999mm
Pocket cycle bottom finish stock	**POC_BFIN_STK**	0 - +/-9.9999mm
Pocket cycle plunge feedrate.	**POC_PLUNG_FR**	0 - 9999.9999mm
Frame cycle cut width %.	**FRA_CUT_WDTH**	10% - 80%
Frame cycle side finish stock.	**FRA_SFIN_STK**	0 - +/-9.9999mm
Frame cycle XY clearance.	**FRA_XY_CLR**	0 - +/-999.9999mm

Probing Cycle Parameters	***Field***	***Value***
Probe Gauge Height.	**PROBE_GH**	0 - 20.0000mm
Tram Surface.	**TRAM_SURFACE**	0 - 999.9999mm
+X Stylus Tip Dimension.	**X_POS_TIP**	0 - 99.9999mm
-X Stylus Dimension.	**X_NEG_TIP**	0 - 99.9999mm
+Y Stylus Dimension.	**Y_POS_TIP**	0 - 99.9999mm
-Y Stylus Dimension.	**Y_NRG_TIP**	0 - 99.9999mm
Fixed Probe Tram.	**FIX_PRB_TRAM**	0 - 999.9999mm
Probe Approach Feedrate.	**PRB_APPR_FRT**	0 - 500.000mmpm
Probe Measurement Feedrate.	**PRB_MEAS_FRT**	0 - 500.000mmpm

[$TOOL_DATA(*Record No.*)*Field*]

(Tool Data Storage Table)

Tool Data	***Field***	***Value***
Tool Number	**RECORD_NUM**	Read Only
Tool Pocket	**POCKET**	0 - 999
Tool Identifier	**IDENTIFIER**	1001 - 9999999999
Tool Serial Number	**SERIAL_NO**	32 Alphanumeric field.
Tool type	**TYPE**	0 = Unknown 1 = ROUGH_END_MILL 2 = FINISH_END_MILL 3 = BALL_END_MILL 4 = FACE_MILL 5 = SHELL_MILL 6 = SPOT_FACE_MILL 7 = KEY_CUTTER 8 = FLY_CUTTER 9 = THREAD_MILL 10 = DRILL 11 = SPOT_DRILL 12 = COUNTER_SINK 13 = REAMER 14 = TAP 15 = RIGID_TAP 16 = BORE 17 = BACKBORE 18 = PROBE 19 = SPECIAL_1 20 = SPECIAL_2 21 = SPECIAL_3 22 = SPECIAL_4 23 = SPECIAL_5 24 = SPECIAL_6 25 = SPECIAL_7 26 = SPECIAL_8 27 = SPECIAL_9
Tool Length	**LENGTH**	Numerical (+/-999.9999mm)
Nominal Tool Diameter	**NOM_DIA**	Numerical (0 - 999.9999mm)
Tool Tip Angle	**TIP_ANGLE**	Angle from tool centreline in degrees. Value is 0 - 359.999
Diameter Offset	**DIA_OFFSET**	Used for CDC compensation (value is +/-999.9999mm)
Tool Flute Length	**FLUTE_LENGTH**	Used for tip angle calculation (value is 0 to +/-999.9999mm)

Number of Teeth	**TEETH**	Used in feed / tooth calculations. Range 1-99 teeth. 1 tooth specifies FPR mode.
Threads per Inch	**TPI**	Threads per Inch for tapping. Range 1-99 (TPI).
Tool Material	**MATERIAL**	0 = Unknown 1 = HSS 2 = HSS_TIN_COATED 3 = CARBIDE_INSERT 4 = CARBIDE_COATED 5 = CARBIDE_SOLID 6 = DIAMOND 7 = CERAMIC 8 = OTHER
Load Method	**LOAD_METHOD**	0 = Auto 1 = Manual 2 = Cradle
Tool Size	**SIZE**	0 = PREVIOUS_0_NEXT_0 1 = PREVIOUS_0_NEXT_1 2 = PREVIOUS_0_NEXT_2 3 = PREVIOUS_1_NEXT_0 4 = PREVIOUS_1_NEXT_1 5 = PREVIOUS_1_NEXT_2 6 = PREVIOUS_2_NEXT_0 7 = PREVIOUS_0_NEXT_1 8 = PREVIOUS_2_NEXT_2
Migrating Mode	**MIGRATING**	0 = Inactive 1 = Active
Spindle Direction	**SPDL_DIR**	0 = DIR_STOP 1 = DIR_CW 2 = DIR_CCW 3 = DIR_EITHER
Feedrate Override	**FDRT_OVR**	Expressed as a % (0-999%)
Spindle Override	**SPEED_OVR**	Expressed as RPM (0-99999)
Maximum RPM	**MAX_RPM**	Range 0 - 99999 RPM
Maximum feed / tooth	**MAX_FEED**	Range 0 - 99999
Tool Status	**TOOL_STATUS**	0 = GOOD 1 = WORN 2 = BROKEN 3 = NEW
Tool cycle time	**CYCLE_TIME**	Optional accumulated cycle time (0 - 9999.99 minutes).

Cycle time limit	**CYC_TIME_LIM**	Optional tool cycle time limit (0 - 9999.99 minutes).
Cycle time mode	**CYC_TM_MODE**	(Option) 0 = TIME_INACTIVE 1 = TIME_ACTIVE
Tool usage count	**USAGE_COUNT**	Optional number of times used (0 - 99999)
Tool usage count limit	**USAGE_LIMIT**	Optional maximum number of times used (0 - 99999)
Tool usage count mode	**USAGE_MODE**	(Option) 0 = COUNT_INACTIVE 1 = COUNT_ACTIVE
Alternate Tool	**ALT_TOOL**	Alternative tool used if programmed tool is worn. This field is not accessible from the NC program.
X and Y probe offset	**X_PRB_OFFSET** **Y_PRB_OFFSET**	Tool centreline offset for tool setting / Probe offset (Range +/- 999.9999)

References to the tool table generally refer to the data for tool in the control tool data table. However, references to [$TOOL_DATA(0)<*Field Name*> refer to the tool in the spindle.

Other Registers

Name	***Field***	***Value***
Count of NC blocks since last cycle start.	**[$BLOCK_COUNT]**	
Current Machine Position.	**[$CURPOS_MCH(*Field*)]**	
Current Program Position.	**[$CURPOS_PGM(*Field*)]**	
Array of floating point computed values.	**[$DATA_CAPTURE]**	
Auto tool recovery/contains a value identifying the condition.	**[$EXCEPTION]**	Broken = 1 Worn = 2 Oversize = 3
Maximum Machine axis travel.	**[$HIGH_LIMIT*(Field)*]**	
Minimum Machine axis travel.	**[$LOW_LIMIT*(Field)*]**	
Pocket number of next but not loaded tool.	**[$NEXT_POCKET]**	Minimum 1 up to maximum machine set.

Set variable when tool probe is out of tolerance.	**[$OUT_OF_TOL]**	TRUE = 1 (No hit or out of tolerance) FALSE = 0
User created Pattern Variable.	**[$PATTERN_END]**	FALSE (G36) = 0 TRUE (G36.1) = 1
Measured Bore or Boss Diameter with G78.	**[$PRB_AVG_DIA]**	
Probe tool error containing the tool diameter deviation as measured by the fixed probe G69 cycle.	**[$PRB_DIA_ERR]**	
Stored co-ordinates for the measured part feature with G75-79 cycles.	**[$PRB_PART_LOC*(Field)*]**	
Position of recent probe hit in machine co-ordinates.	**[$PRB_POS_MC*(Field)*]**	
Position of recent probe hit in program co-ordinates.	**[$PRB_POS_PC*(Field)*]**	
Tool length deviation created when using G69 cycle.	**[$PRB_TOOL_ERROR]**	
Measured width of pocket or web after a G79 cycle.	**[$PRB_WIDTH]**	
Measured X axis diameter of a bore or boss after a G78 cycle.	**[$PRB_X_DIA]**	
Measured Y axis diameter of a bore or boss after a G78 cycle.	**[$PRB_Y_DIA]**	
Probe Hit.	**[$PROBE_HIT]**	TRUE = 1 (HIT) FALSE = 0 (NO-HIT)
Tool data / Record number.	**[$RECORD_NO]**	0 = Current spindle tool -1 = No tool in spindle
Tool probe hit location.	**[$TOOL_PRB_LOC*(Field)*]**	

The word *(Field)* relates to one of either X,Y,Z axis words.

Date/Time Stamp	**[$CALENDER(0)*Field*]** **[$CALENDER(1)*Field*]** **[$CALENDER(2)*Field*]**	Calender 0 = read only Calender 1 = store data Calender 1 = store data

The word *(Field)* in Date/Time Stamp relates to one of either year, month, day, dayofweek, hour, minutes or second

Programmable Coolant

Automatic Coolant Jet Control (Option)

The Coolant Jet System, mounted beneath the spindle carrier, has eight positions to ensure the coolant is directed to the cutting tip of any tool within the maximum tool length and diameter specified for the machine.
The automatic positioning occurs using the length and diameter data stored in the tables of the tool table.

The group of miscellaneous codes (M8.1 - M8.8) can also manually control the positioning of the Automatic Coolant Jet.

The table below can be used to identify the M code to an active tool length and radius. Code M8.1 is associated with the smallest/shortest tool and code M8.8 is associated with the largest/longest tool.

Tool Length		< 100	< 150	< 200	< 250	> 250
Tool Diameter	> 100	M8.3	M8.5	M8.6	M8.7	M8.8
	< 100	M8.2	M8.4	M8.5	M8.6	M8.6
	< 60	M8.2	M8.3	M8.4	M8.5	M8.6
	< 30	M8.1	M8.2	M8.3	M8.4	M8.5

Example Program:

```
: T1 M6
X? Y? Z100 H? S? M3
Z5                      { Establish length offset
G1 Z-? F? M8            { Switch on coolant
G41 X? M8.5             { Establish radius offset & coolant direction
Y?
```

Milling Cycles Corner Specified

Mill cycle formats

Linear Facemilling (corner)

G22.1 X_____ Y_____ Z_____ R_____ U_____ V_____ Q_____ K_____ ;minimum required

O_____ P_____ J_____ F_____ S_____ W_____ ;additional

Linear Pocket (corner)

G23.1 X_____ Y_____ Z_____ R_____ U_____ V_____ Q_____ K_____ L_____ ;minimum required

,R_____ ,D_____ I_____ E_____ O_____ P_____ J_____ F_____ S_____ W____;additional

Linear Internal Frame (corner)

G24.1 X_____ Y_____ Z_____ R_____ U_____ V_____ Q_____ K_____ J_____ ;minimum required

,R_____ ,D_____ I_____ O_____ P_____ J_____ F_____ S_____ W_____ ;additional

Linear External Frame (corner)

G25.1 X_____ Y_____ Z_____ R_____ U_____ V_____ Q_____ K_____ J_____ ;minimum required

,R_____ ,D_____ I_____ O_____ P_____ J_____ F_____ S_____ W_____ ;additional

Linear Feature Facemill – XY Corner
G22.1

G22.1 X? Y? U? V? Z? R? Q? K? [O? W?] (J? P?) { F? S? }

X? = Modal feature corner position
Y? = Modal feature corner position
U? = Modal length of feature in the X axis (signed)
V? = Modal width of feature in the Y axis (signed)
Z? = Modal feature depth position from "R"
R? = Modal surface position
Q? = Modal type of milling (Q0, Q2, Q3, Q4, Q10, Q12, Q13, Q14)
K? = Modal incremental pecking amount in the Z axis (if "K"="Z" 1 peck)
O? = Modal angle of feature from horizontal plan view
W? = Non-modal incremental retraction in Z axis from "R"
J? = Finish stock to be removed (J0 = no stock)
P? = Modal width of cut as a percentage (10% - 80%)
F? = Modal cutting finishing feedrate
S? = Modal cutting finish spindle speed

[ ] denotes optional input
( ) if not programmed – values taken from cycle parameters.
{ } Only required with Q0, Q1, Q4, Q5, Q10, Q11, Q14, Q15

S1000 F? M3	**- Spindle start + Rough mill feed.**
G22.1 X? Y? U? V? Z? R? Q? P? J? K?	**- Position to 1st position setting all data.**
X?	**- Position to 2nd position**
Y?	**- Position to 3rd position**
G0 G90 Z100	**- Tool to a safe height (G0 cancel)**

At Z finish position the tool retracts automatically.

Linear Feature Pocket Mill – XY Corner
G23.1

G23.1 [X? Y?] U? V? Z? R? Q? K? L? E? [O? W? ,R? ,D?] (I? J? P?) {F? S?}

X? = Modal feature corner position
Y? = Modal feature corner position
U? = Modal length of feature in the X axis (signed)
V? = Modal width of feature in the Y axis (signed)
Z? = Modal feature depth position from "R"
R? = Modal surface position
Q? = Modal type of milling (All Q's)
K? = Modal incremental pecking amount in the Z axis (if "K"="Z" 1 peck)
L? = Modal type of plunging (L0, L-1, L>0)
E? = Modal Z axis plunge feedrate
O? = Modal angle of feature from horizontal plan view
W? = Non-modal incremental retraction in Z axis from "R"
,R? = Modal corner radius of feature
,D? = Corner feed ratio override (,D0 = Full slowdown, ,D100 = No slowdown)
I? = Finish stock on side to be removed (I0 = no stock)
J? = Finish stock on bottom to be removed (J0 = no stock)
P? = Modal width of cut as a percentage (10% - 80%)
F? = Modal cutting finishing feedrate
S? = Modal cutting finish spindle speed

[ ] denotes optional input
( ) if not programmed – values taken from cycle parameters.
{ } Only required with Q0, Q1, Q4, Q5, Q10, Q11, Q14, Q15

S1000 F? M3	**- Spindle start + Rough mill feed.**
G23.1 X? Y? U? V? Z? R? Q? L? E? P? I? J? K?	**- Position to 1st position setting all data.**
X?	**- Position to 2nd position**
Y?	**- Position to 3rd position**
G0 G90 Z100	**- Tool to a safe height (G0 cancel)**

At Z finish position the tool retracts automatically.

Linear Feature Inside Frame – XY Corner
G24.1

G24.1 [X? Y?] U? V? Z? R? Q? K? J? [O? W? ,R? ,D?] (I? P?) { F? S? }

X? = Modal feature corner position
Y? = Modal feature corner position
U? = Modal length of feature in the X axis (signed)
V? = Modal width of feature in the Y axis (signed)
Z? = Modal feature depth position from "R"
R? = Modal surface position
Q? = Type of milling (All Q's)
K? = Modal incremental pecking amount in the Z axis (if "K"="Z" 1 peck)
J? = Total stock on inside to be removed
O? = Modal angle of feature from horizontal plan view
W? = Non-modal incremental retraction in Z axis from "R"
,R? = Modal corner radius of feature
,D? = Corner feed ratio override (,D0 = Full slowdown, ,D100 = No slowdown)
I? = Finish stock on side to be removed (I0 = no stock)
P? = Modal width of cut as a percentage (10% - 80%)
F? = Modal cutting finishing feedrate
S? = Modal cutting finish spindle speed

[ ] denotes optional input
( ) if not programmed – values taken from cycle parameters.
{ } Only required with Q0, Q1, Q4, Q5, Q10, Q11, Q14, Q15

S1000 F? M3	**- Spindle start + Rough mill feed.**
G24.1 X? Y? U? V? Z? R? Q? P? I? J? K?	**- Position to 1st position setting all data.**
X?	**- Position to 2nd position**
Y?	**- Position to 3rd position**
G0 G90 Z100	**- Tool to a safe height (G0 cancel)**

At Z finish position the tool retracts automatically.

THIS CYCLE ASSUMES NO MATERIAL IS TO BE REMOVED IN THE POCKET CENTRE AND SO WILL RAPID Z-AXIS TO "K" DEPTH.

Linear Feature Outside Frame – XY Corner
G25.1

G25.1 [X? Y?] U? V? Z? R? Q? K? J? [O? W? ,R? ,D?] (I? P?) { F? S? }

X? = Modal feature corner position
Y? = Modal feature corner position
U? = Modal length of feature in the X axis (signed)
V? = Modal width of feature in the Y axis (signed)
Z? = Modal feature depth position from "R"
R? = Modal surface position
Q? = Type of milling (All Q's)
K? = Modal incremental pecking amount in the Z axis (if "K"="Z" 1 peck)
J? = Total stock on outside to be removed
O? = Modal angle of feature from horizontal plan view
W? = Non-modal incremental retraction in Z axis from "R"
,R? = Modal corner radius of feature
,D? = Corner feed ratio override (,D0 = Full slowdown, ,D100 = No slowdown)
I? = Finish stock on side to be removed (I0 = no stock)
P? = Modal width of cut as a percentage (10% - 80%)
F? = Modal cutting finishing feedrate
S? = Modal cutting finish spindle speed

[ ] denotes optional input
( ) if not programmed – values taken from cycle parameters.
{ } Only required with Q0, Q1, Q4, Q5, Q10, Q11, Q14, Q15

S1000 F? M3	**- Spindle start + Rough mill feed.**
G25.1 X? Y? U? V? Z? R? Q? P? I? J? K?	**- Position to 1st position setting all data.**
X?	**- Position to 2nd position**
Y?	**- Position to 3rd position**
G0 G90 Z100	**- Tool to a safe height (G0 cancel)**

At Z finish position the tool retracts automatically.

Macro's

Arithmetical Expressions

Mathematical expressions

X8.5 + 2.5	= 11
X3 * 2	= 6
X3 ** 2	= 9
X COS (15)	= 0.9659
X3/2	= 1.5
RND(12.453287)	= 12
SQR(25)	= 5
<>	Not equal to?
>	Greater Than ?
<	Less Than ?
=	Equal to ?

The order in which the control evaluates expressions is important, since a change in the order of evaluation can result in a change of the resulting value. The order of evaluation follows standard algebraic practise. That is:

1) The inner most parenthetic expression ( ) is evaluated first.
2) The next to inner set is solved next and each is solved in turn working towards the end of the sum.

The order of solving arithmetic operations within parentheses or when parentheses are not present is as follows:

1) All multiplication, division and modulo operations are performed in order from left to right.
2) All addition and subtraction are performed in order from left to right.
3) Relational operators are evaluated.

If an expression has a + or **-** in front of it then the value is treated as though a zero is in front of it.

Example

-2.*3-10/5+9 = 1

Calculated as: 2.*3 = 6 & 10/5 = 2 **Value** = 0-6-2+9 = 1

Typical Error **(2+2/10) = 2.2** OR **(2+2)/10 = 0.4**

Macro Variables

All macro variables must start with a letter or an underscore "_" and can be a mixture of numbers, letters and underscores all contained within [ ]. Maximum number of characters is 12 and does not include the [ ].

Local variables – [#???]

can only be created and used within a main program or created and used within a sub-program.
A # macro cannot be created within a main program to be used within a sub-program.

Maximum of 50 allowed

i.e..
[@A2100] = 35
G1 X[@A2100] Y[@A2100]

Common Variables – [@???]

@ Can be created and used within a main program. A sub-program of a main program can use the same macro.

Maximum of 100 allowed

i.e..
[@A2100] = 35
(CLS, "SUB1")

(DFS."SUB1")
G1 X[@A2100] Y[@A2100]
(ENS)

Parameter variables – [&???]

& can be used to transfer information from a CLS, sub-program call line into the sub-program itself. If the letter being used on the CLS, block is the same parameter variable letter in the sub-program that matches the program word its attached to, a ! can be used in the subprogram instead.

Letter "G" cannot be used

i.e.**(CLS, "SUB1", A81 X50 Y30 Z20 R0 F200)**

(DFS,"SUB1")	**(DFS,"SUB1")**
G[&A] X-[&X] Y-[&Y] Z[&Z] R[&R] F[&F] M8	**G[&A] X-! Y-! Z! R! F! M8**
X[&X]	**X!**
Y[&Y]	**Y!**
X-[&X]	**X-!**
(ENS)	**(ENS)**

Macro Variables

Branch (GOTO <label>) statements

Macro programs use conditional branch statements to check to see if the macro formula is complete. The use of a GOTO statement can contain mathematical check functions to see if the statement is complete. If the statement (logical expression i.e. X=Y) is not complete (false), then a branching statement on this GOTO line of program will send the control to a "Label Identifier" to carry on with the program. If the statement is complete (true) then the program will carry on.

(GOTO <label>)

or

(IF <logical expression> GOTO <label>)

Label Identifiers

The A2100 control allows the programmer to set "Label Identifiers" within a program to allow macro programs to search for a program position for use with a GOTO statement.

Repetition example

Repeat 5 times

```
[@COUNT]=0
[LOOP] (Label at start of loop)
[@COUNT] = [@COUNT] + 1
"PROGRAM"
(IF [@COUNT] < 5 GOTO [LOOP])   ( < less than)
```

Macro for a Counterbore

```
:T? M6 ; ENDMILL
G0 G90 G40 G71 G17 G94
X? Y? Z100 H1 S? M3 ;Move to bore centre
Z?
;
(CLS,"C_BORE",A?) ;A = C/BORE DIAMETER
G0 G90 Z100
M2
```

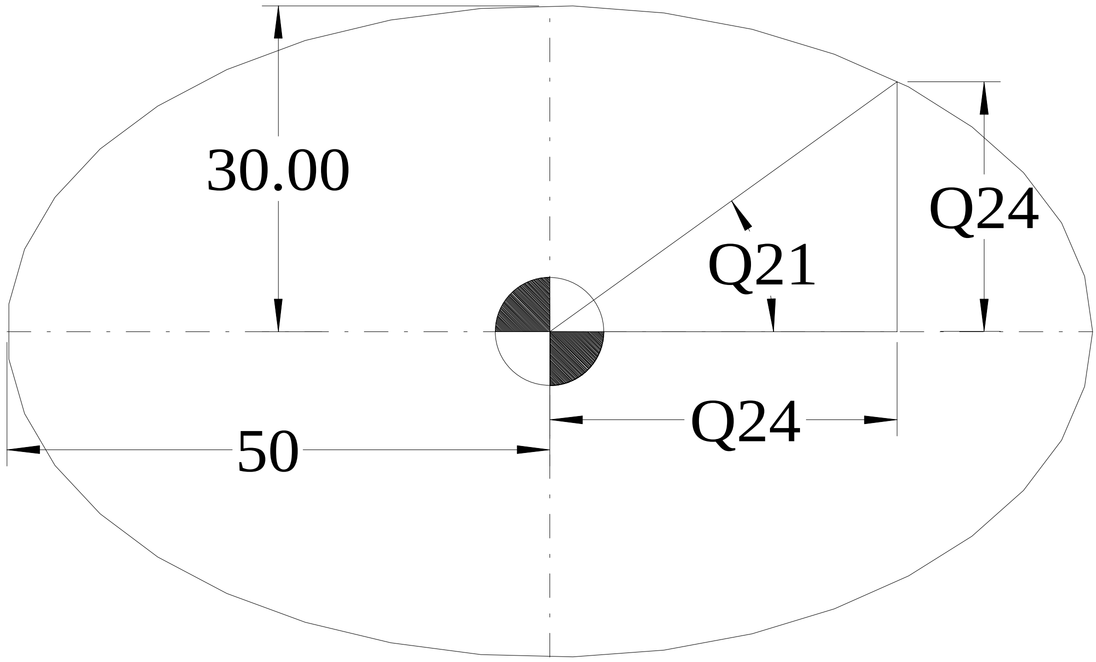
```
;NEW PROGRAM SAVED AS C_BORE
[@B]=(([&A]*0.8)/2)
[@C]=(([&A]/2)-[@B])
G91 Y[@C]
G41 X[@B]
G3 X-[@B] Y[@B] P[@B]
X0 Y0 I0 J-([&A]/2)
X-[@B] Y-[@B] P[@B]
G1 G40 X[@B]
Y-[@C]
(ENS)
```

Macro for Internal Helical

```
:T? M6 ; THREADMILL
G0 G90 G40 G71 G17 G94
X? Y? Z100 H1 S? M3 ;Move to bore centre
Z?
;
(CLS,”HELICAL”,A? B?)
;A = THREAD DIAMETER
;B = PITCH
G0 G90 Z100
M30
```

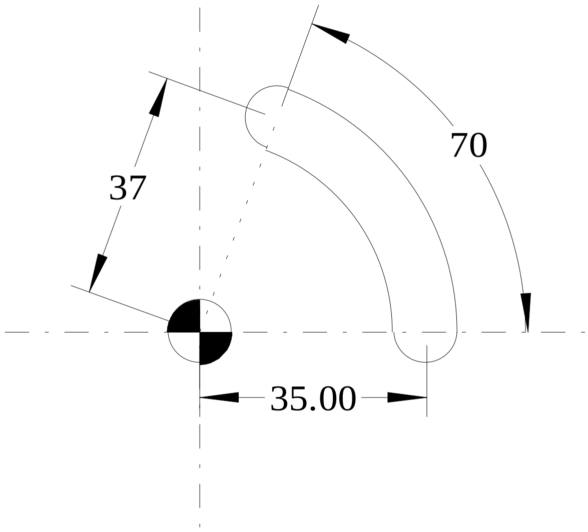
```
;NEW PROGRAM SAVED AS HELICAL
[@B]=(([&A]*0.8)/2)
[@C]=(([&A]/2)-[@B])
;******************
G91 Y[@C]
G41 X[@B]
G3 X-[@B] Y[@B] P[@B] Z[&B]/4 K[&B]
X0 Y0 I0 J-([&A]/2) Z[&B] K[&B]
X-[@B] Y-[@B] P[@B] Z[&B]/4 K[&B]
G1 G40 X[@B]
Y-[@C]
(ENS)
```

Internal Elipse




```
:T1 M6
G0 G90 G40 G71 G17 G94
X0 Y0 Z5 S? M3
G1 Z-? F?
[@Q20] = 2 ; Incremental degree calculation
[@Q21] = 0 ; Start Angle
[@Q22] = 30 ; Y Axis Radius
[@Q23] = 50 ; X Axis Radius
G41 X[@Q23]  ; Compensation motion to right side of internal pocket
[START]
[@Q21] = ([@Q21] + [@Q20])  ; Angular Count
[@Q24] = SIN([@Q21])  ; Incremental Y axis calculation
[@Q25] = COS([@Q21]) ; Incremental X axis calculation
[@Q24] = ([@Q24]*[@Q22])  ; Absolute Y calculation
[@Q25] = ([@Q25]*[@Q23])  ; Absolute X calculation
X[@Q25] Y[@Q24] ; Movement in X & Y axis
(IF [@Q21] < 360 GOTO [START]) ; Restart if less than 360 degree motion
(IF [@Q21] > 360 GOTO [NEXT]) ; If final angle becomes greater than 360 degrees recalculate
(IF [@Q21] = 360 GOTO [END]) ; Finish if total angle is equal to 360 degree
[NEXT]
[@Q21] = 360
(GOTO [START])
[END]
G40 X0
M30
```

Macro for an Increasing Radius.




```
:T? M6
G0 G90 G40 G71 G17 G94
X35 Y0 Z100 H? S? M3
Z5
G1Z? F?
(CLS,"CAM")
G0G90Z100M30
;*****************************
(DFS,"CAM")
; Information for radius
[@start_angle1]=0
[@start_rad1]=35
[@inc_angle]=.01
[@tot_angle1]=70
[@fin_rad1]=37
[@feed] = 500
;*****************************
;*****************************
[@total_moves]=[@tot_angle1]/[@inc_angle]
[@inc_rad]=(([@fin_rad1]-[@start_rad1])/[@total_moves])
[start]
[@start_rad1]=[@start_rad1]+[@inc_rad]
[@start_angle1]=[@start_angle1]+[@inc_angle]
[@x]=[@start_rad1]*cos([@start_angle1])
[@y]=[@start_rad1]*sin([@start_angle1])
G1 X[@x] Y[@y] f[@feed]
(if [@start_angle1] < [@total_angle1] goto [start])
G0Z10
(ENS)
```

Tapered Helical Milling

USING THE TAPERED HELICAL INTERPOLATION PROGRAM SUBROUTINE FOR TAPERED THREADMILLING APPLICATIONS ON THE ACRAMATIC A2100 CONTROL SYSTEM REQUIRES "ADVANCE PROGRAMMING PACKAGE"

The subroutine has been written with a name of **TAPER THREAD MILL.** That is, it is to be registered in the program directory with that name, and must be called from the main program by the same name, i.e. by **(CLS,"TAPER THREAD MILL")**

ONLY INTERNAL THREADS CAN BE PRODUCED WITH THIS SUB-ROUTINE

The sub-routine performs automatic helical lead-on and lead-off to and from the thread. The main-line program should initially position the slides at the thread centre in X and Y axis. The Z axis should be positioned at the full thread depth so as to cut the thread in a positive Z direction. The helical lead-on and lead-off will do all necessary extra movement. The programmer should make sure that **AT LEAST** one quarter of the thread pitch dimension is available in the Z axis above the top of the thread and below the full thread depth to allow the extra lead-on and lead-off moves.

Cutter diameter compensation must be **OFF** , to avoid problems with non-move spans during calculation blocks. The sub-routine takes into account the diameter of the active tool directly by reference to the system variables. Because of this, the tool diameter in the tool table must be entered into the **NOMINAL DIAMETER** & not a deviation. Any diameter deviation should be entered in the **DIAMETER OFFSET** field of the tool table.

The call subroutine block will read as follows:

(CLS,"TAPER THREAD MILL",G? X? Y? Z? I? J? K?)

Where:

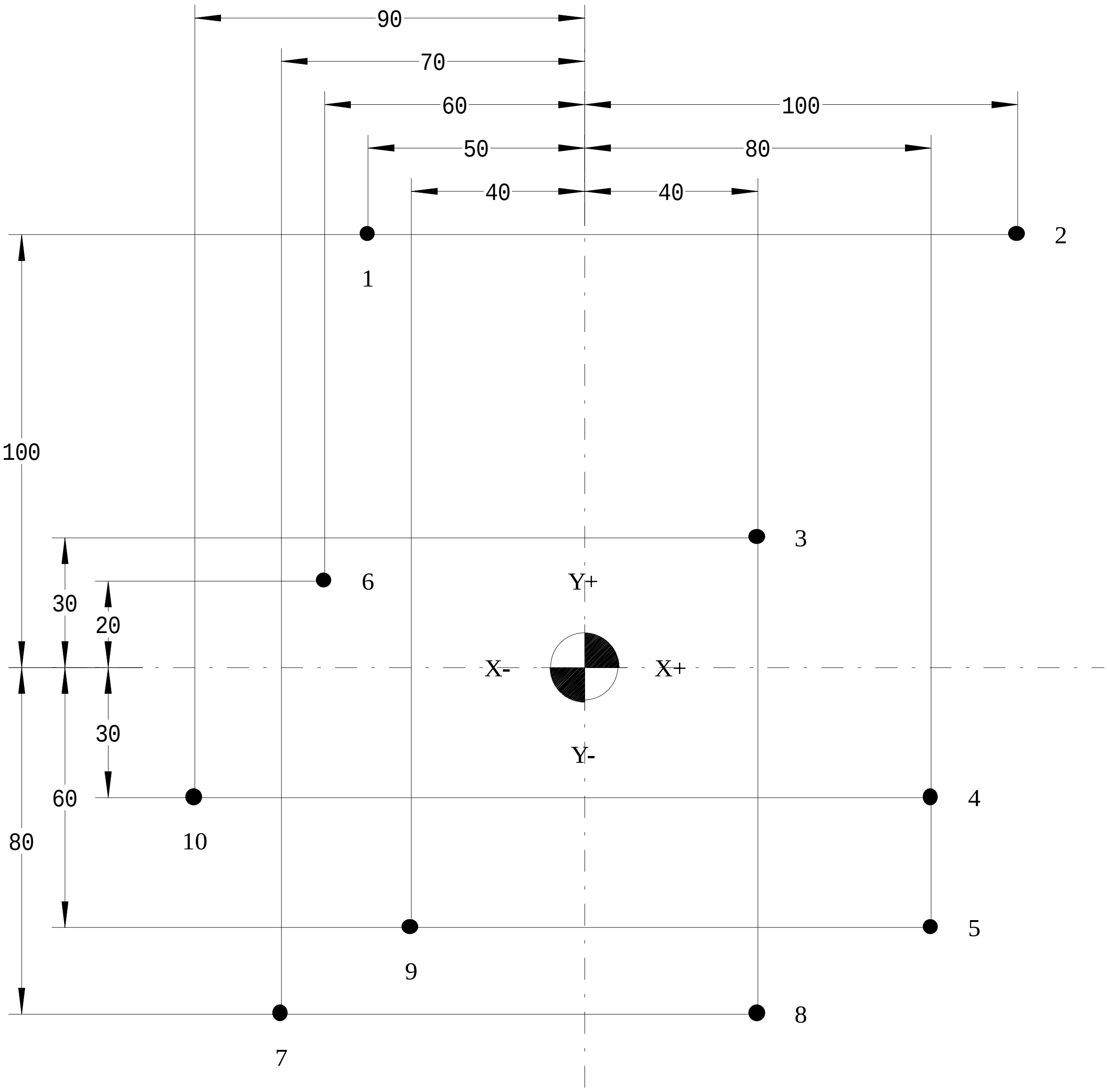
G? = Rotational direction
X? = Initial Radius
Y? = Final Radius
Z? = Total "Z" length (signed increment)
I? = Thread centre in the "X" axis (absolute co-ordinates)
J? = Thread centre in the "Y" axis (absolute co-ordinates)
K? = Helical Pitch ("Z" movement per revolution)October 23, 1997

SUB-ROUTINE CODE

```
[@INITIAL_Z] = [$CURPOS_PGM(Z)]
[@TOOL_RAD] = [$TOOL_DATA(0)NOM_DIA] + [$TOOL_DATA(0)DIA_OFFSET]
[@CORR_RAD] = [&X] - [@TOOL_RAD] / 2
[@STEP_COUNT] = 1
(IF ABS(([&Y] - [&X])/([&Z]/[&K]) < 0.2 THEN)
[@SP_PER_REV] = 4
(ELSE)
[@SP_PER_REV] = 360/ARCSIN(1-SQR(0.2/ABS(([&Y]-[&X])/([&Z]/[&K]))))
(ENDIF)
[@NO_OF_STEPS] = [@SP_PER_REV]*[&Z]/[&K]
[@NO_OF_STEPS] = ABS(INT([@NO_OF_STEPS]))
[@ANG_STEP] = 360*(ABS([&Z])/[&K])/[@NO_OF_STEPS]
(IF [&G] = 2 THEN)
[@ANG_STEP] = -[@ANG_STEP]
(ENDIF)
[@RAD_INCRMNT] = ([&Y] - [&X])/[@NO_OF_STEPS]
[@Z_STEP] = [&Z]/[@NO_OF_STEPS]
[@APP_RAD] = ([&X] + ([@TOOL_RAD]/2))/2
[@APP_PITCH] = [&K]*[@APP_RAD]/[&X]
[@APP_RAD] = [@APP_RAD]-[@TOOL_RAD]/2
(IF [&G] = 3 THEN)
G90G1 X[&I]+[@CORR_RAD]-[@APP_RAD] Y[&J]-[@APP_RAD] Z[@INITIAL_Z]-[@APP_PITCH]/4
G91G3 X[@APP_RAD] Y[@APP_RAD] Z[@APP_PITCH]/4 K[@APP_PITCH] P[@APP_RAD]
(ELSE)
G90G1 X[&I]+[@CORR_RAD]-[@APP_RAD] Y[&J]-[@APP_RAD] Z[@INITIAL_Z]-[@APP_PITCH]/4
G91G2 X[@APP_RAD] Y[@APP_RAD] Z[@APP_PITCH]/4 K[@APP_PITCH] P[@APP_RAD]
(ENDIF)
(DO)
[@SPIRAL_ANGLE] = ([@ANG_STEP]*[@STEP_COUNT])/360
[@SPIRAL_RAD] = [@CORR_RAD] + [@RAD_INCRMNT]*[@STEP_COUNT]
(IF [&G] = 3 THEN)
G90G3 X[&I]+[@SPIRAL_RAD]*COS([@SPIRAL_ANGLE])
Y[&J]+[@SPIRAL_RAD]*SIN([@SPIRAL_ANGLE])
Z[@INITIAL_Z]+[@Z_STEP]*[@STEP_COUNT] P[@SPIRAL_RAD]-[@RAD_INCRMNT]/2
K[&K]
(ELSE)
G90G2 X[&I]+[@SPIRAL_RAD]*COS([@SPIRAL_ANGLE])
Y[&J]+[@SPIRAL_RAD]*SIN([@SPIRAL_ANGLE])
Z[@INITIAL_Z]+[@Z_STEP]*[@STEP_COUNT] P[@SPIRAL_RAD]-[@RAD_INCRMNT]/2
K[&K]
(ENDIF)
[@STEP_COUNT] = [@STEP_COUNT]+1
(LOOP WHILE [@STEP_COUNT] < [@NO_OF_STEPS]+1)
[@APP_PITCH] = [@APP_PITCH]*[&X]/[&Y]
(IF [&G] = 3 THEN)
G91G3 X-[@APP_RAD]*(COS([@SPIRAL_ANGLE]) + SIN([@SPIRAL_ANGLE]))
Y[@APP_RAD]*(COS([@SPIRAL_ANGLE])-SIN([@SPIRAL_ANGLE])) Z[@APP_PITCH]/4
K[@APP_PITCH] P[@APP_RAD]
(ELSE)
G91G2 X-[@APP_RAD]*(COS([@SPIRAL_ANGLE]) + SIN([@SPIRAL_ANGLE]))
Y[@APP_RAD]*(COS([@SPIRAL_ANGLE])-SIN([@SPIRAL_ANGLE])) Z[@APP_PITCH]/4
K[@APP_PITCH] P[@APP_RAD]
(ENDIF)
G90G1
(ENS)
```

EXAMPLE ANSWERS

G90 Absolute Example Programming



N1 G90 X-50 Y100

N2 X100

N3 X40 Y30

N4 X80 Y-30

N5 Y-60

N6 X-60 Y20

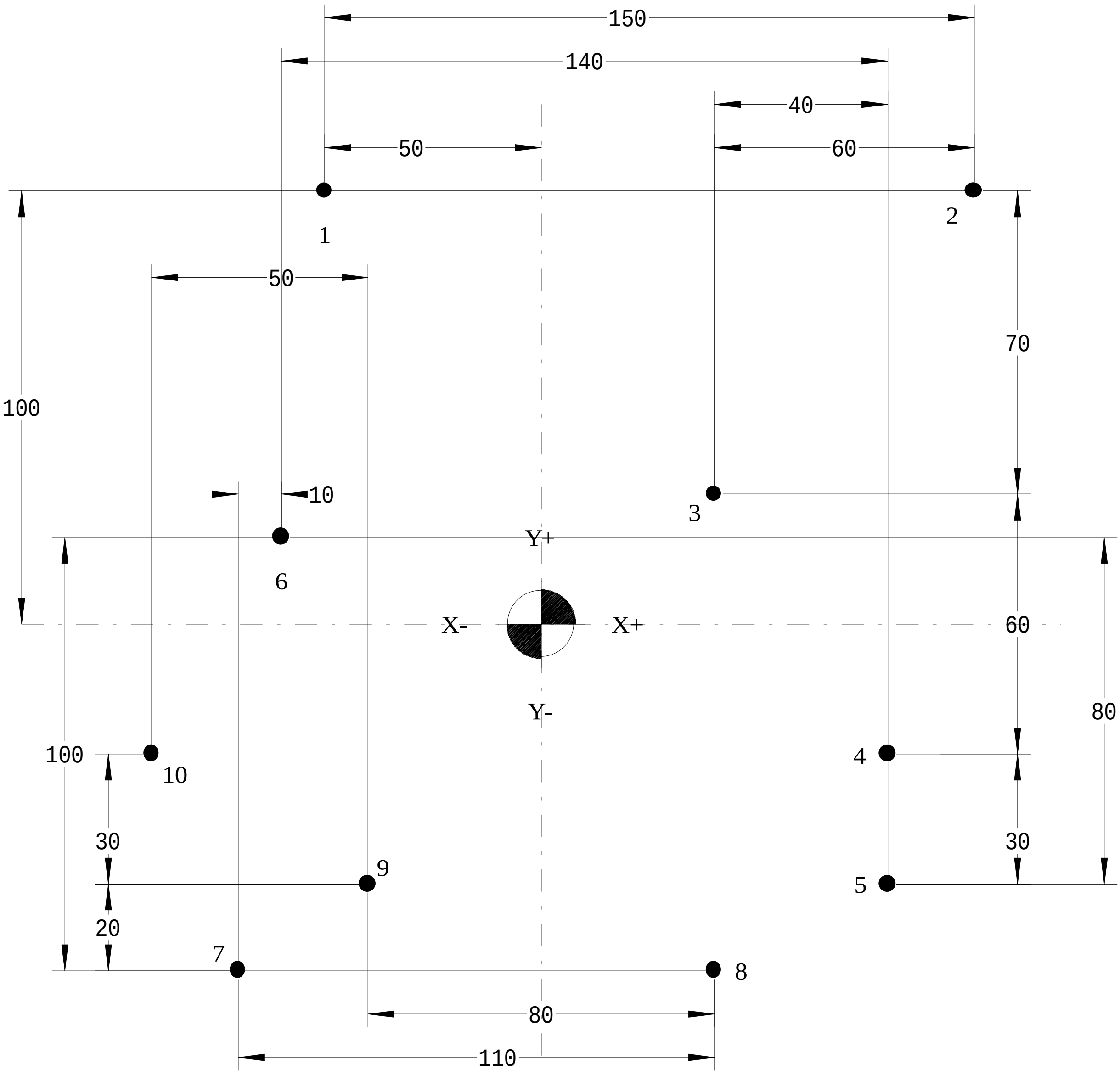
N7 X-70 Y-80

N8 X40

N9 X-40 Y-60

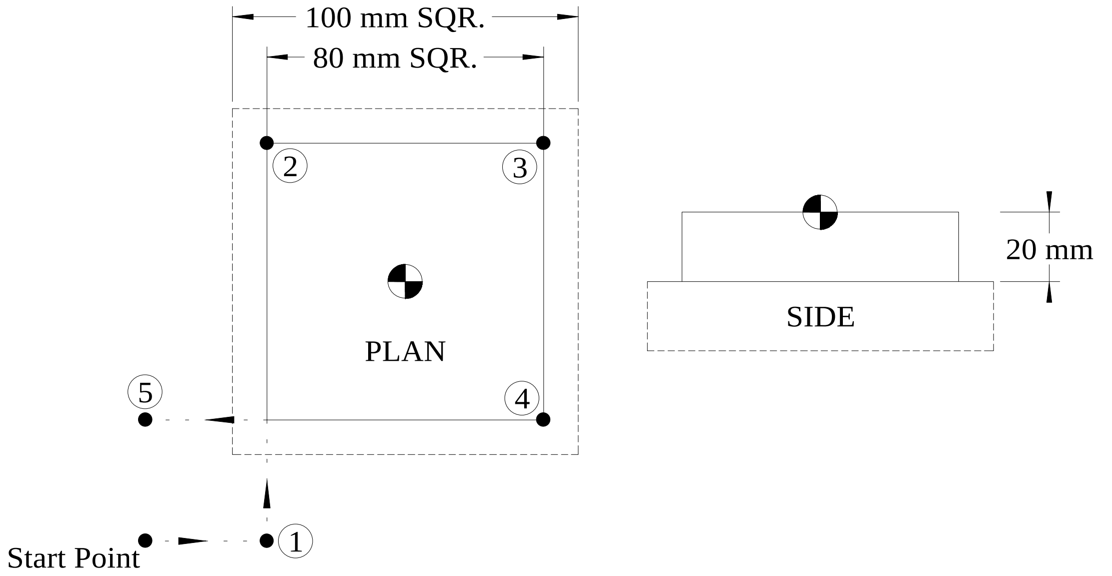
N10 X-90 Y-30

G91 Incremental Example Programming



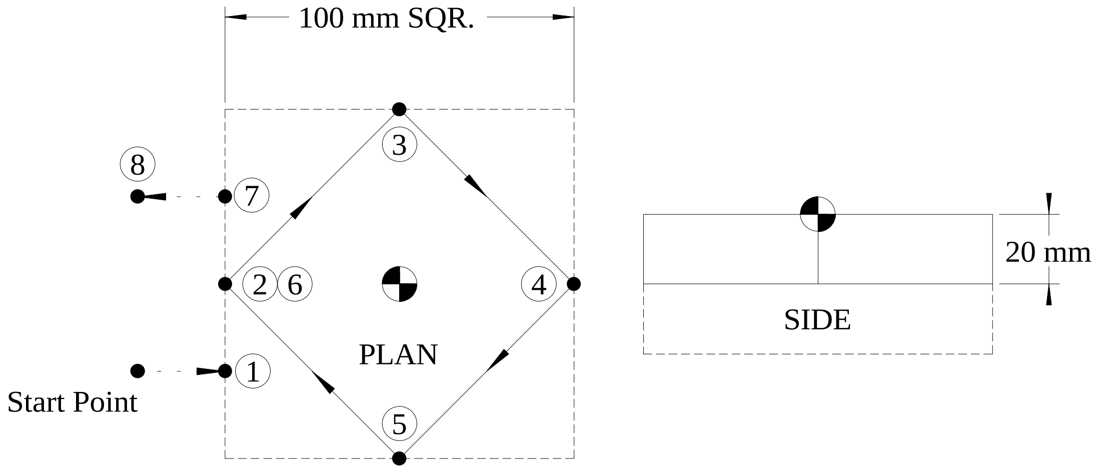
N1 G90 X-50 Y100	
N2 G91 X150	
N3 X-60 Y-70	
N4 X40 Y-60	
N5 Y-30	
N6 X-140 Y80	
N7 X-10 Y-100	
N8 X110	
N9 X-80 Y20	
N10 X-50 Y30	

Point to Point - example 1
(no compensation)



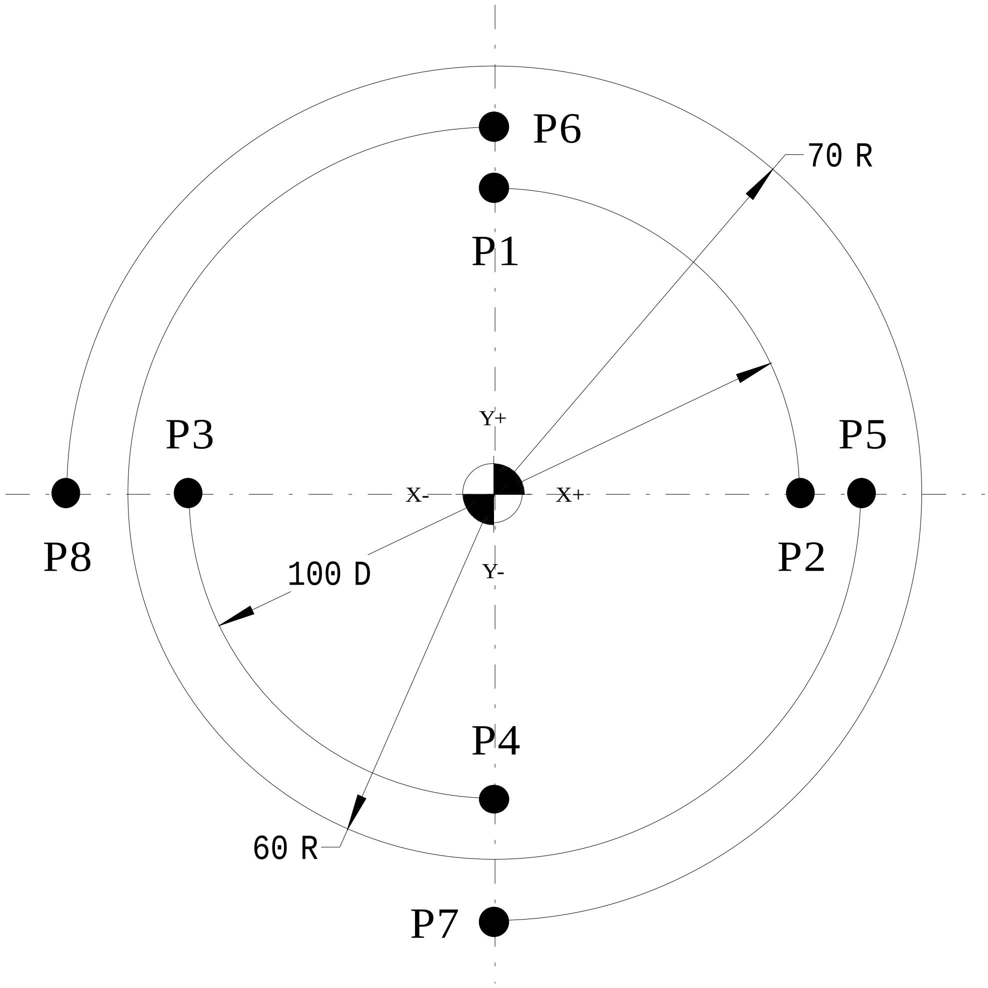
: T? M6
G0 G90 G40 G71 G17 G94
X-75 Y-75 Z100 H1 S? M3
Z5
G1 Z-20 F?
X-40
Y40 M8
X40
Y-40
X-75
G0 G90 Z100
M30

Point to Point - example 2
(no compensation)



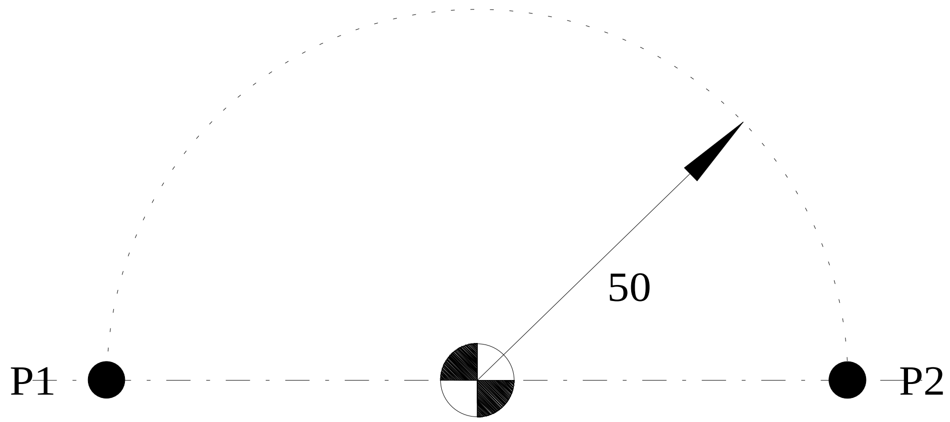
: T? M6
G0 G90 G40 G21 G17 G94 G80
X-75 Y-25 Z100 H1 S? M3
Z5
G1 Z-20 F?
X-50 M8
Y0
X0 Y50
X50 Y0
X0 Y-50
X-50 Y0
Y25
X-75
G0 G90 Z100
M30

Circular Programmed Movements – e.g. 1
(Absolute)



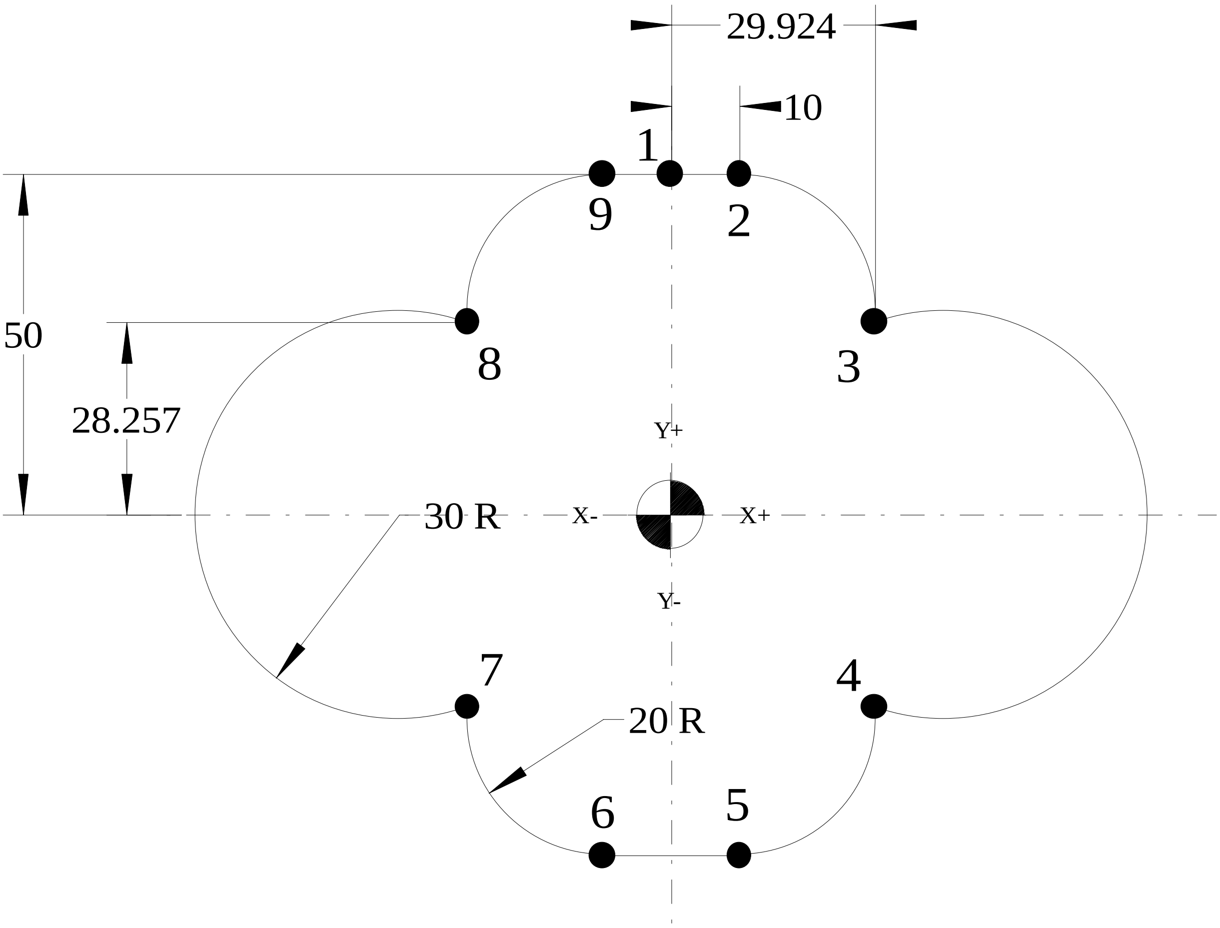
1	G90 G0 X0 Y50
	G2 X50 Y0 P50
2	G90 G0 X-50 Y0
	G3 X0 Y-50 P50
3	G90 G0 X60 Y0
	G2 X0 Y60 P-60
4	G90 G0 X0 Y-70
	G3 X-70 Y0 P-70

Circular Programmed Movements – e.g. 2
(Absolute)



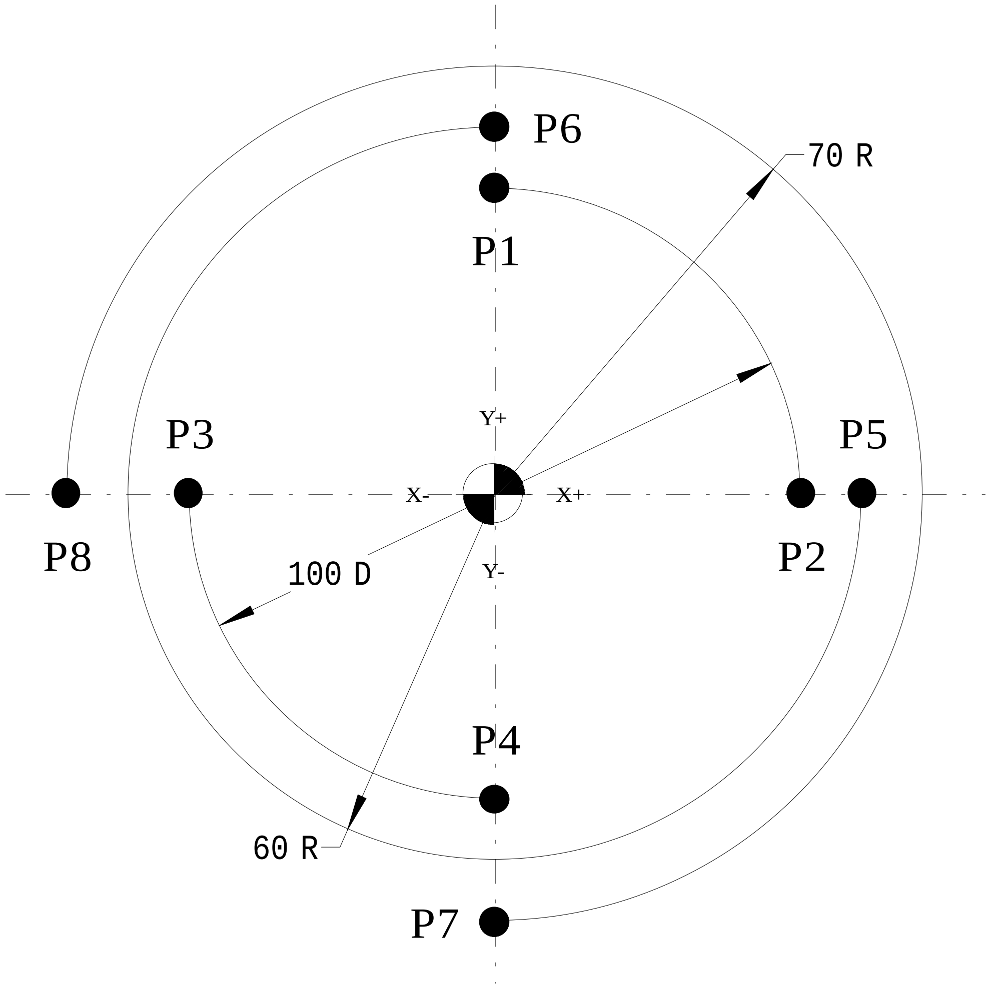
1	G90 G0 X-50 Y0
	G2 X50 P50
2	G90 G0 X50 Y0
	G3 X-50 P50

Circular Programmed Movements – e.g. 3



	1 G0 X0 Y50
1	2 G1 X10
2	3 G2 X29.924 Y28.257 P20
3	4 X29.924 Y-28.257 P-30
4	5 X10 Y-50 P20
5	6 G1 X-10
6	7 G2 X-29.924 Y-28.257 P20
7	8 X-29.924 Y28.257 P-30
8	9 X-10 Y50 P20
9	1 G1 X0

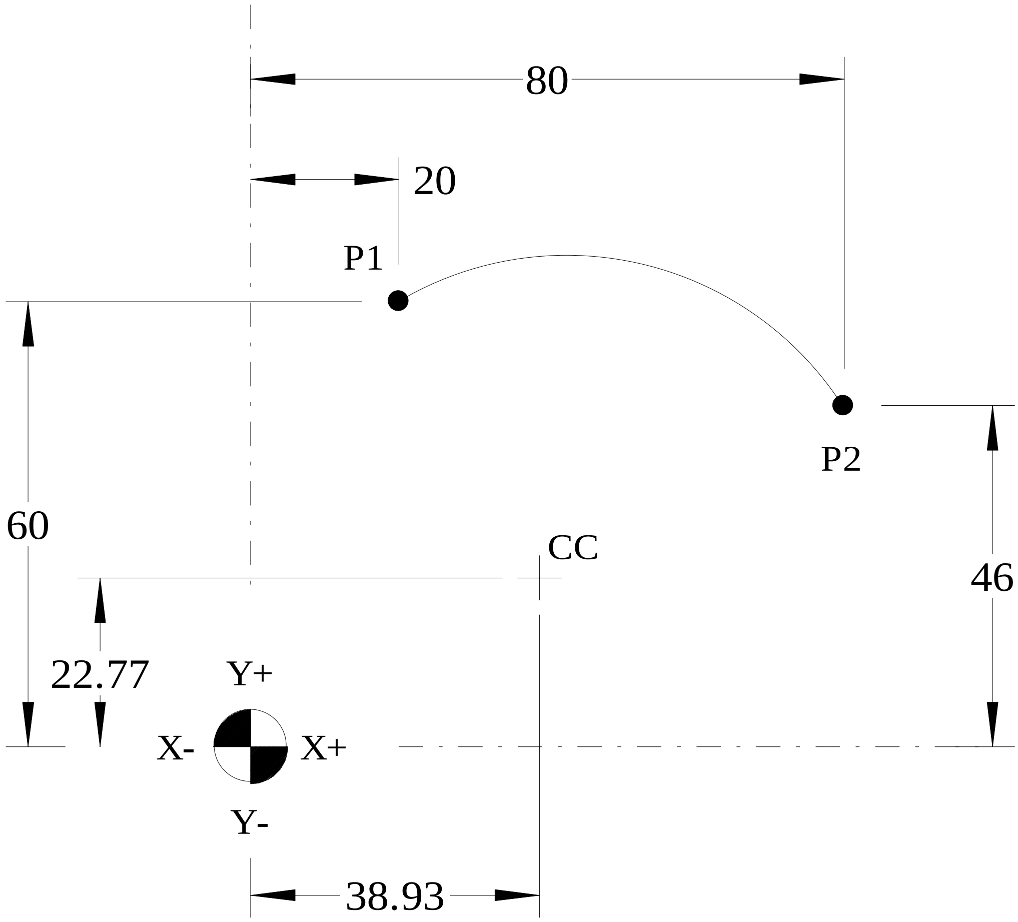
Circular Programmed Movements – e.g. 4
(Incremental)



1	G90 G0 X50 Y0
	G91 G2 X50 Y-50 P50
2	G90 G0 X-50 Y0
	G3 X-50 Y-50 P50
3	G90 G0 X60 Y0
	G2 X-60 Y60 P-60
4	G90 G0 X0 Y-70
	G3 X-70 Y70 P-70

Circular Programmed Movements – e.g. 5

X, Y, I & J (Absolute)



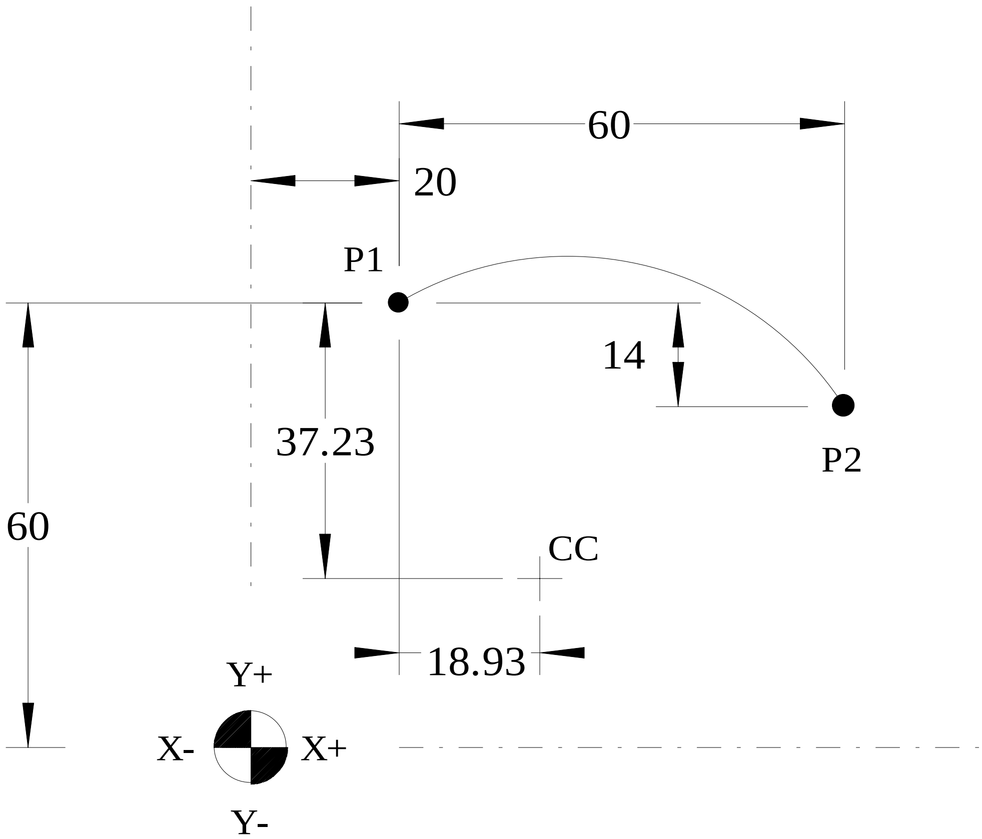
G0 G90 X20 Y60
G2 X80 Y46 I38.93 J22.77

****

G0 G90 X20 Y60
G3 X20 Y60 I38.93 J22.77

Circular Programmed Movements – e.g. 6

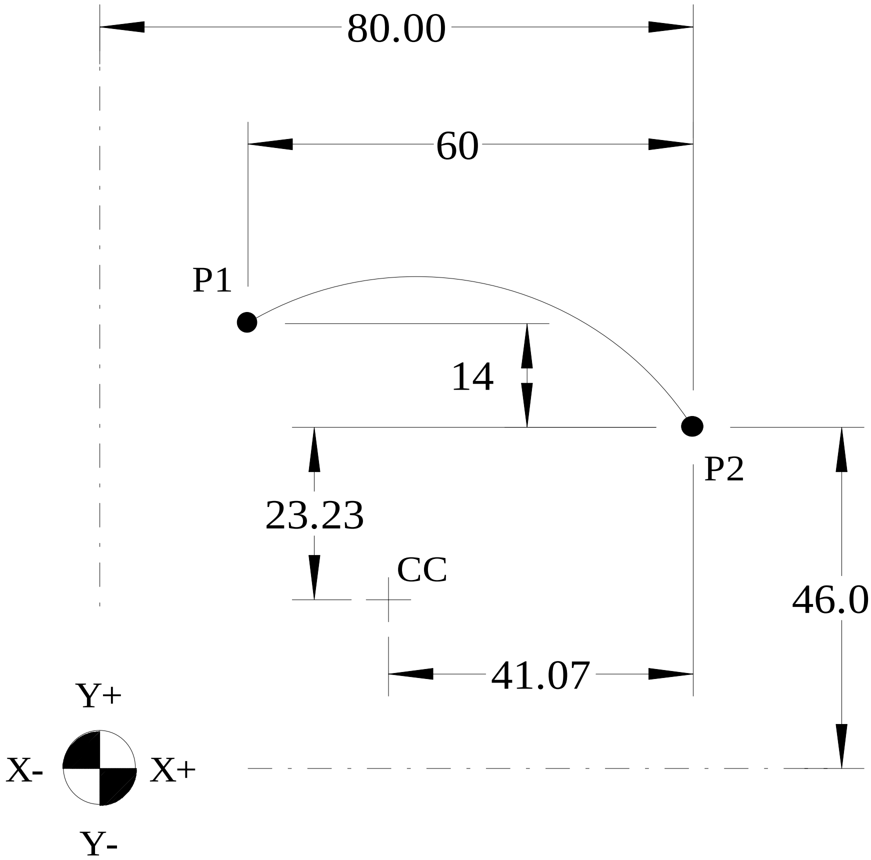
X, Y, I & J (Incremental)



G0 G90 X20 Y60
G2 G91 X60 Y-14 I18.93 J-37.23

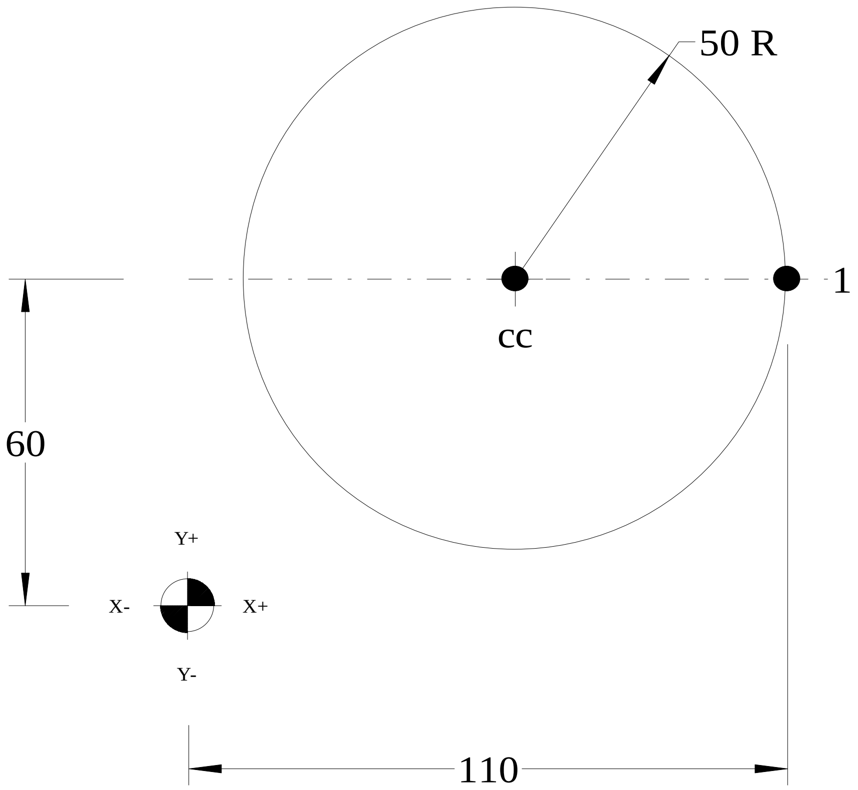
Circular Programmed Movements – e.g. 7

X & Y, I & J (Incremental)



G0 G90 X80 Y46
G3 G91 X-60 Y14 I-41.07 J-23.23

Full Circular Movements – e.g.8



G0 G90 X110 Y60
G2 X110 Y60 I60 J60

Incremental

Note:

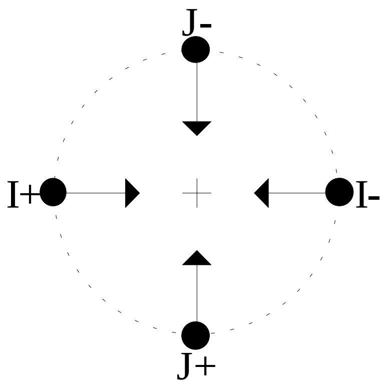
The end points and circle centre positions are taken from the “Start point”

G0 G90 X110 Y60
G2 G91 X0 Y0 I-50 J0

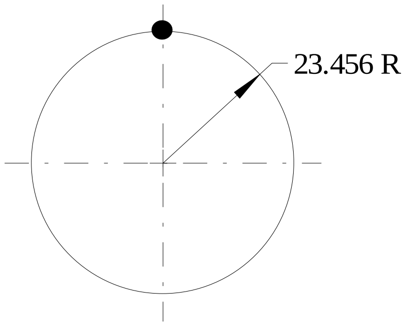
Full Circular Movements – e.g.9

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

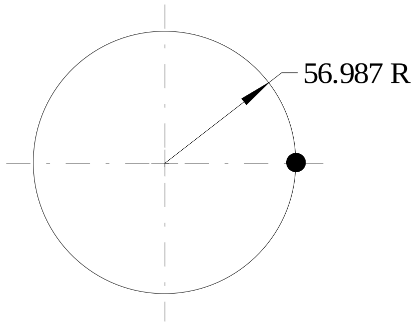
i.e. G91 G2 X0 Y0 I0 J-50



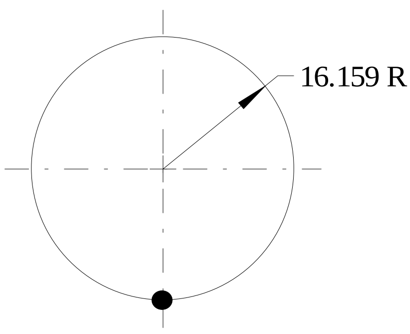
Create the line of circle program for each of the following quadrant points in the diagrams below using "Clockwise".



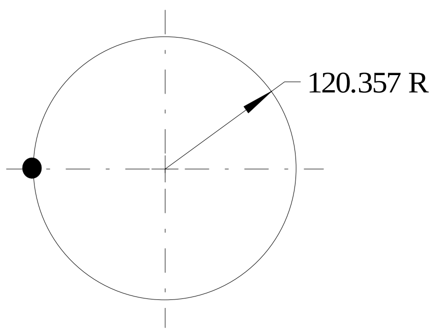
G91 X0 Y0 I0 J-23.456



G91 X0 Y0 I-56.987 J0

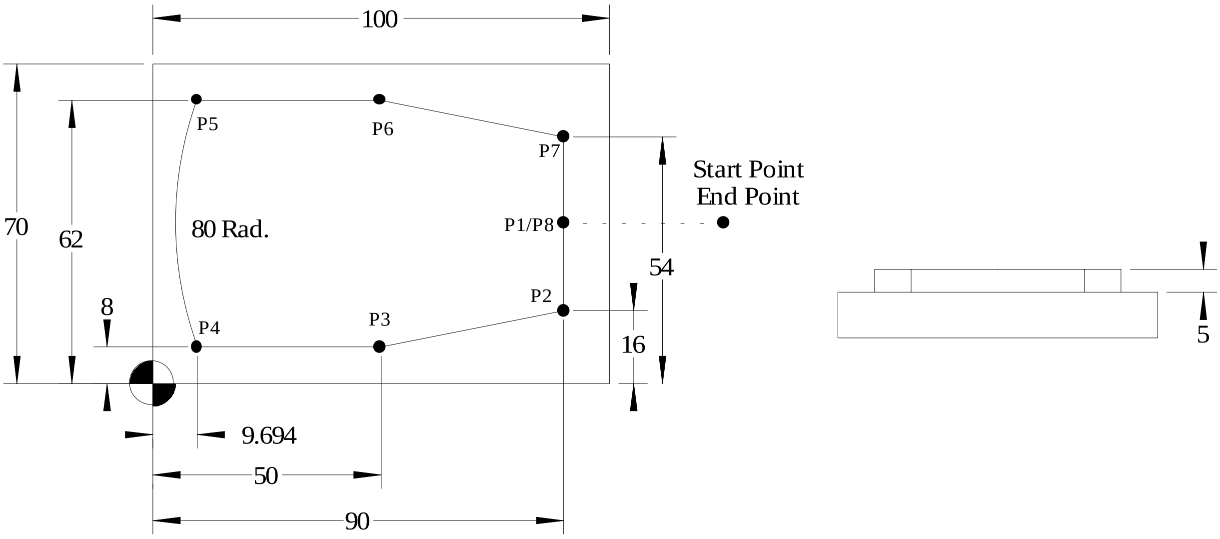


G91 X0 Y0 I0 J16.159



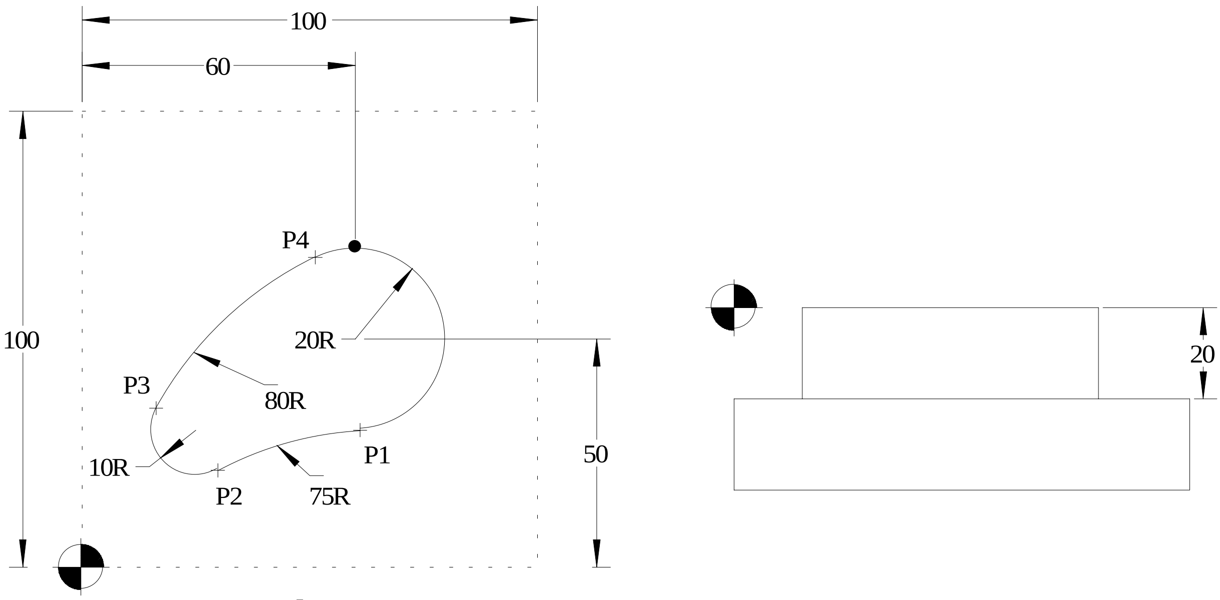
G91 X0 Y0 I120.357 J0

Compensation e.g. 1



: T1 M6
G0 G90 G40 G71 G17 G94
X125 Y35 Z100 H1 S? M3
Z5
G1 Z-5 F?
G41 X90 M8
Y16
X50 Y8
X9.694
G2 Y62 P80
G1 X50
X90 Y54
Y35
G40 X125
G0 G90 Z100 M30

Compensation e.g. 2

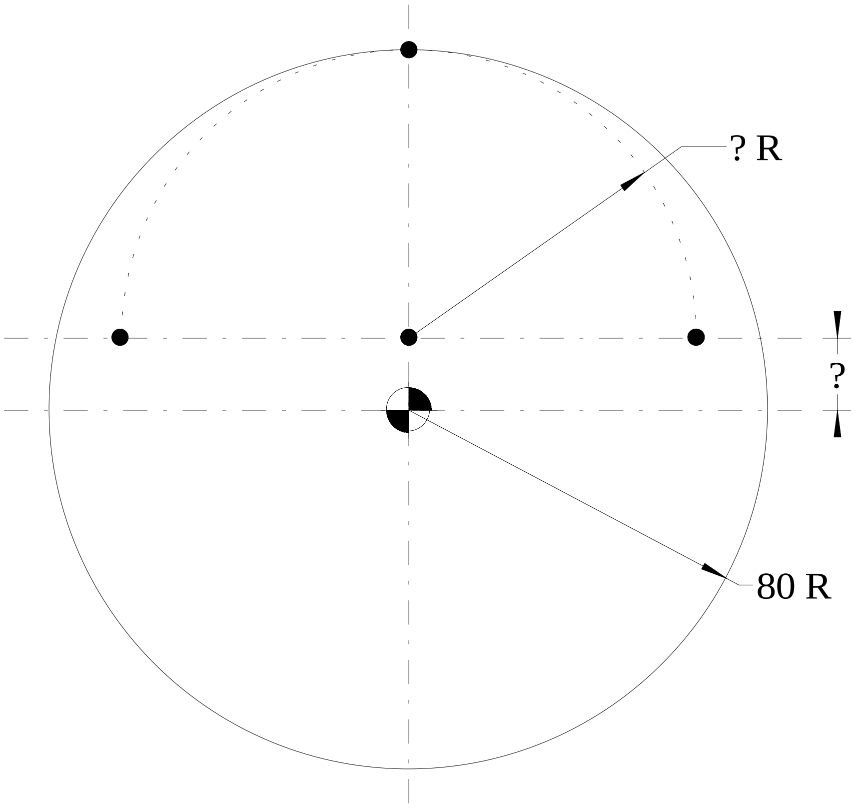


Plan Front

	P1	P2	P3	P4
X	61.11	29.738	16.257	51.266
Y	30.031	21.194	34.854	67.992

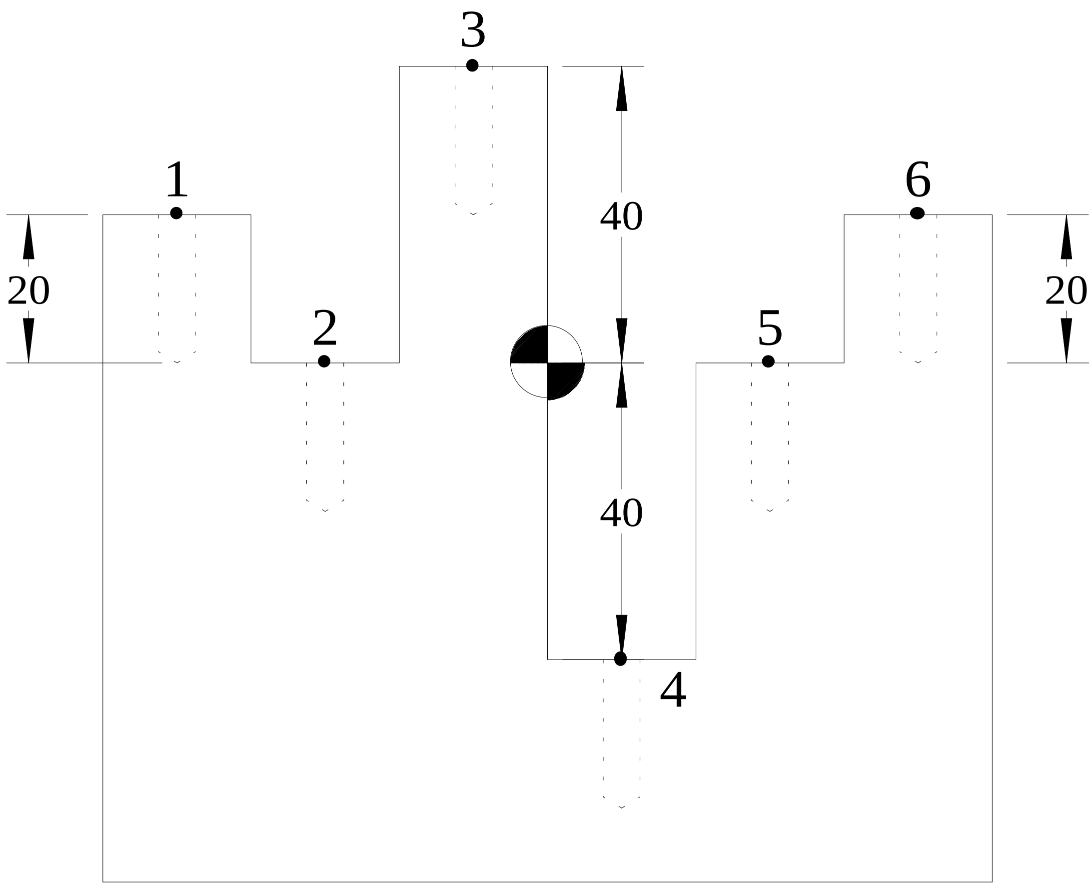
```
: T? M6
G0 G90 G40 G71 G17 G94
X-50 Y-120 Z100 H1 S? M3
Z5
G1 Z-20 F?
G41 Y70
X60 M8
G2 X61.11 Y30.031 P20
G3 X29.738 Y21.194 P75
G2 X16.257 Y34.854 P10
X51.266 Y67.992 P80
X60 Y70 P20
G1 X150
G40 Y120
G0 G90 Z100
M30
```

Circle Tangent Compensation e.g. 1



: T? M6
G0 G90 G40 G71 G17 G94
X0 Y0 Z100 H1 S? M3
Z5
G1 Z-? F?
G91 Y20
G41 X60 M8
G3 X-60 Y60 P60
X0 Y0 I0 J-80
X-60 Y-60 P60
G1 G40 X60
G0 G90 Z100
M30

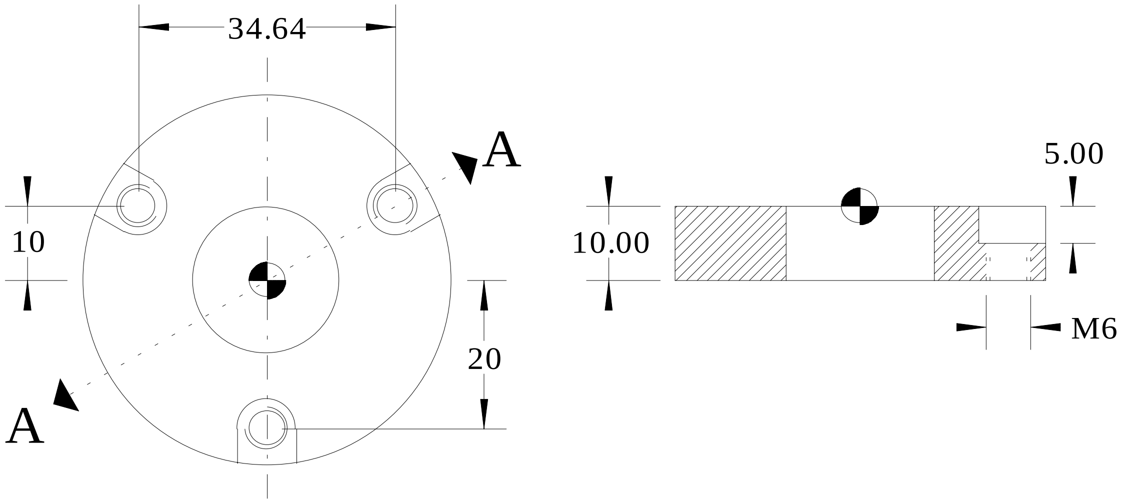
Hole Canned Cycles
“R” & “W” Positions

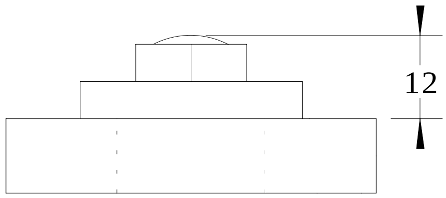


Section View YZ plane
All holes 20mm Deep

R20 Z-20
R0 (Z-20) W45
R40 (Z-20)
R-40 (Z-20) W45
R0 (Z-20) W25
R20 (Z-20)

Hole Example

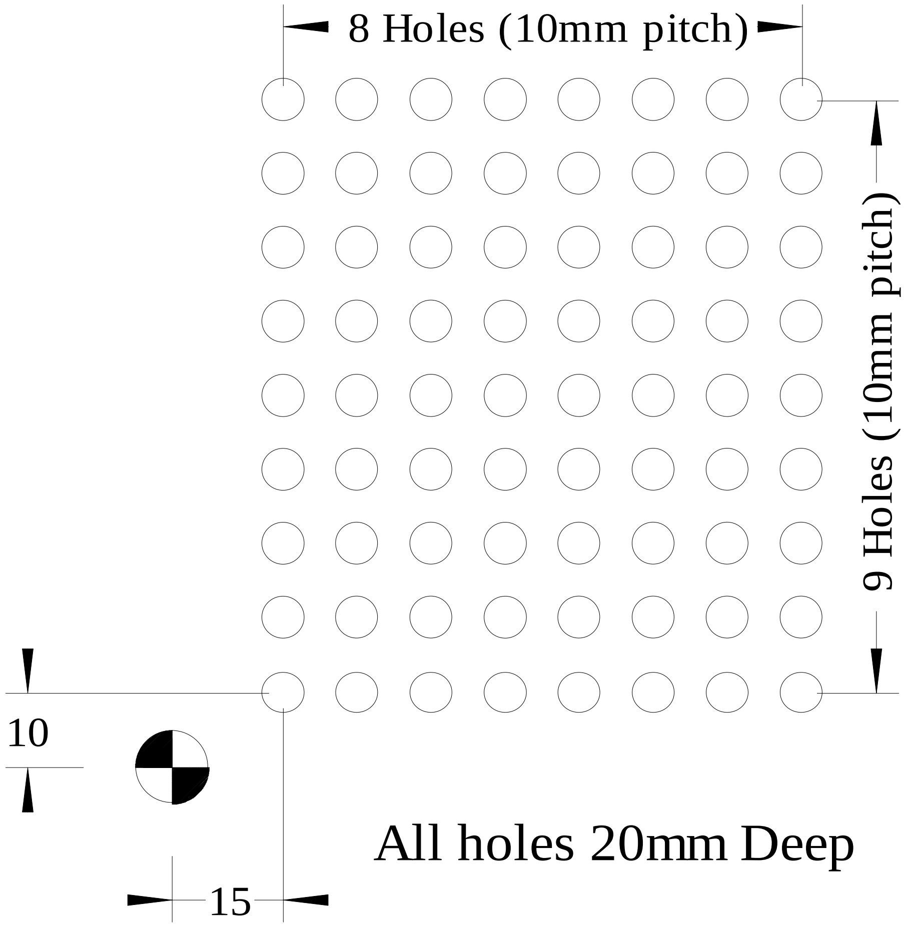




Fixture View

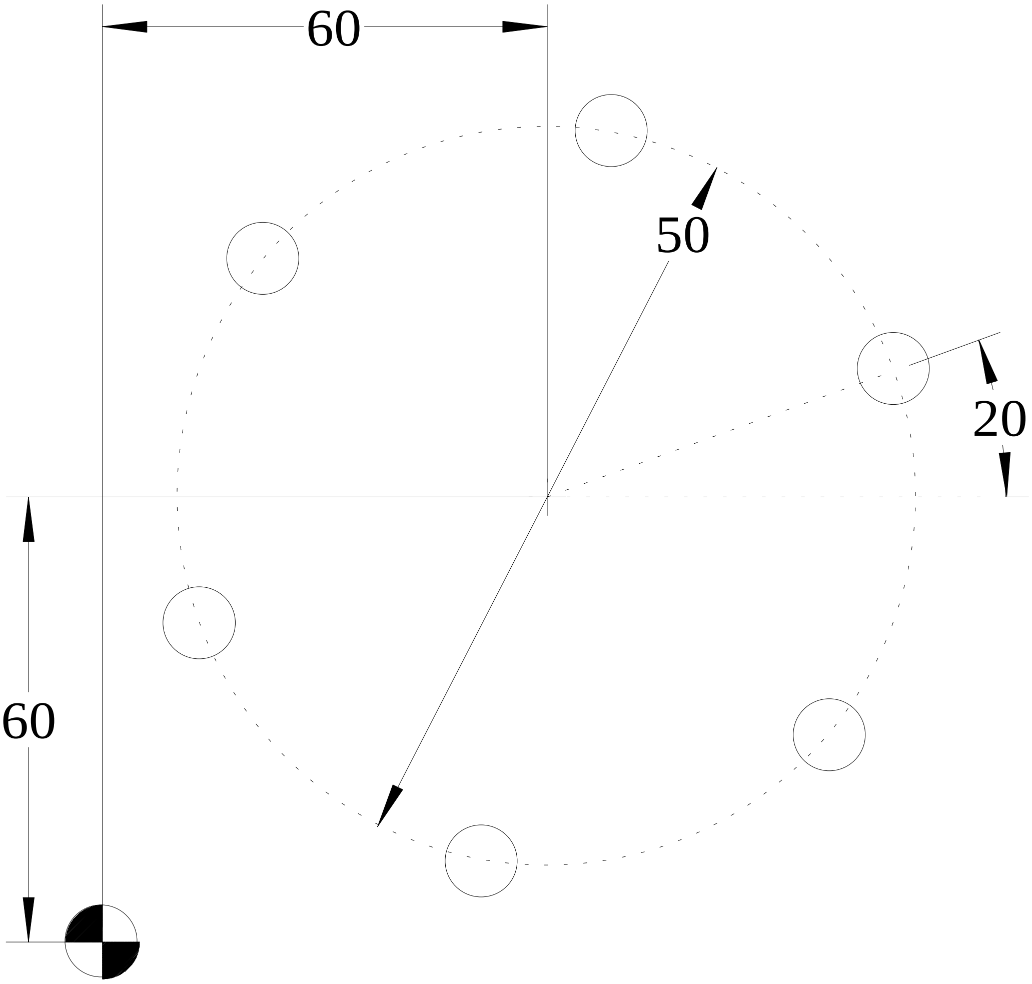
```
: T? M6
G0 G90 G40 G71 G17 G94
X0 Y-20 Z100 H1 S? M3
Z5
G81 R-5 Z-5 F? W10 M8
X17.32 Y10 W10
X-17.32 W10
G80
G0 G90 Z100 M1
: T? M6
G0 G90 G40 G71 G17 G94
X0 Y-20 Z100 H1 S? M3
Z5
G84 G95 R-5 Z-5 W10 F1 M8
X17.32 Y10 W10
X-17.32 W10
G80
G0 G90 Z100 M30
```

Grid Pattern Cycle
G38



: T? M6
G0 G90 G40 G71 G17 G94
X15 Y10 Z100 H? S? M3
Z5
G38 U10 I8 J9 V10
G81 R0 Z-20 F? M8
G37
G0 G90 Z100 M30

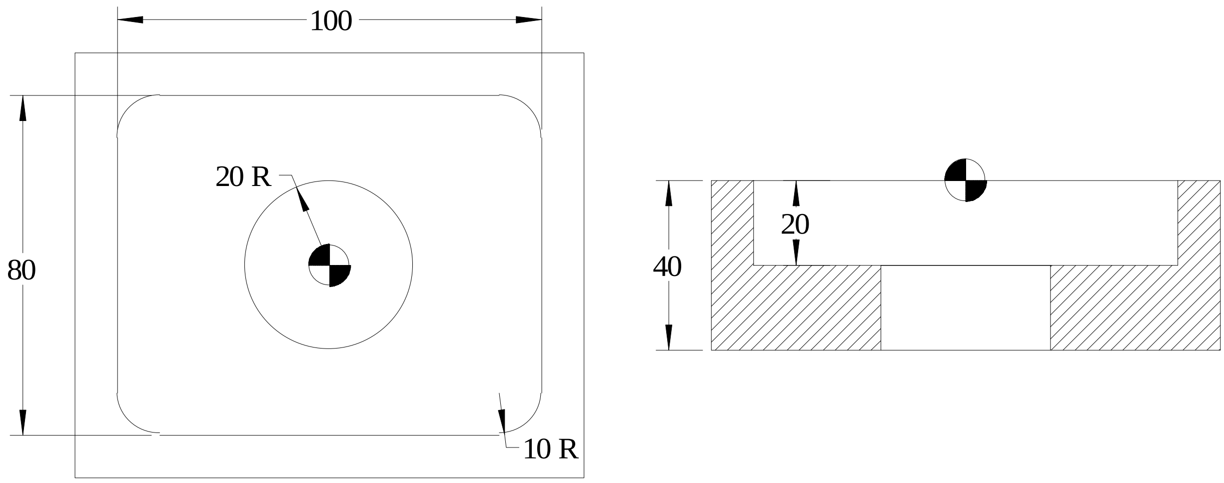
P.C.D. Pattern Cycle
G39



Holes 20mm Deep

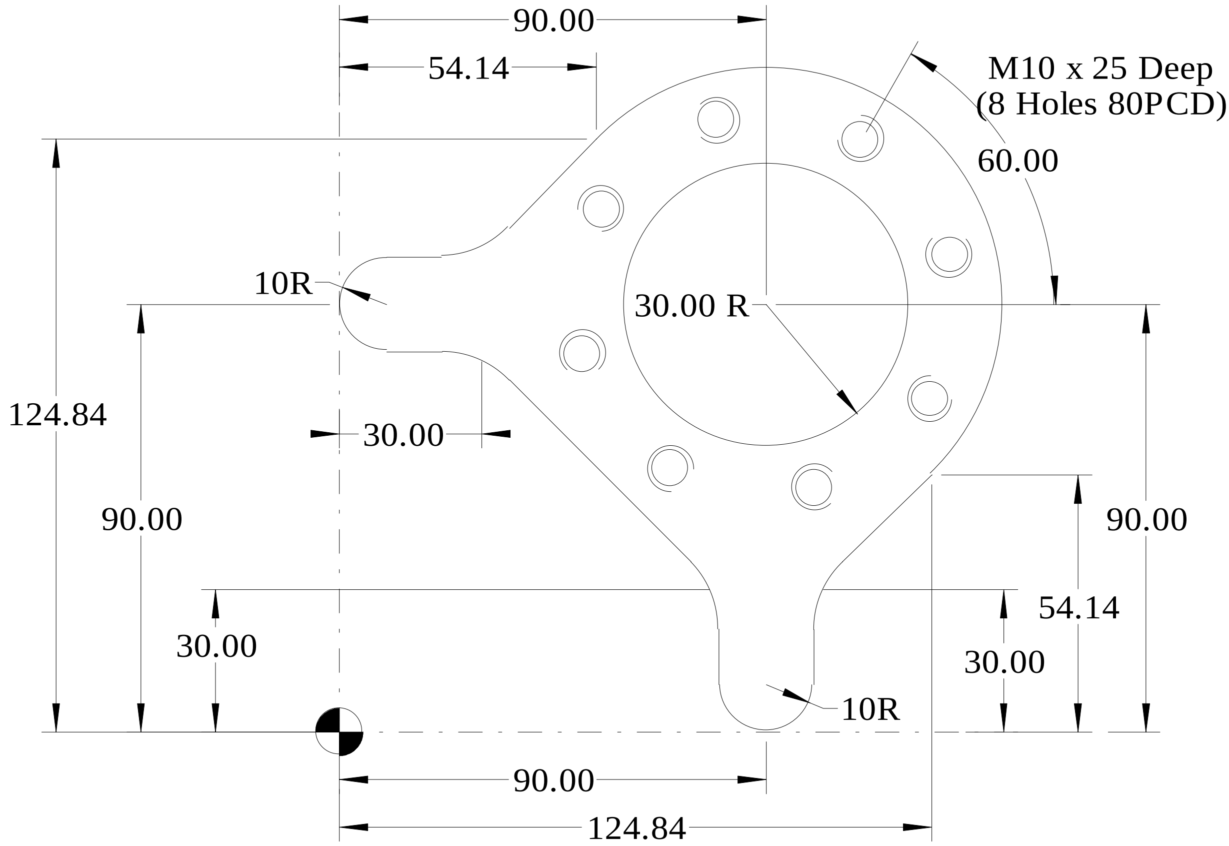
: T? M6
G0 G90 G40 G71 G17 G94
X60 Y60 Z100 H? S? M3
Z5
G39 D50 P20 K6
G81 R0 Z-20 F? M8
G37
G0 G90 Z100 M30

Milling Cycle Example




```
: T? M6
G0 G90 G40 G71 G17 G94
X0 Y0 Z100 H? S? M3
Z5 F?
G23 U100 V80 R0 Z-20 K5 Q3 I0 J0 E? P66 ,R10 ,D0 L3 M8
G26.1 U40 R-20 Z-20 K5 Q3 I0 J0 E? P66 L-1 W25
G0 G90 Z100 M30
```

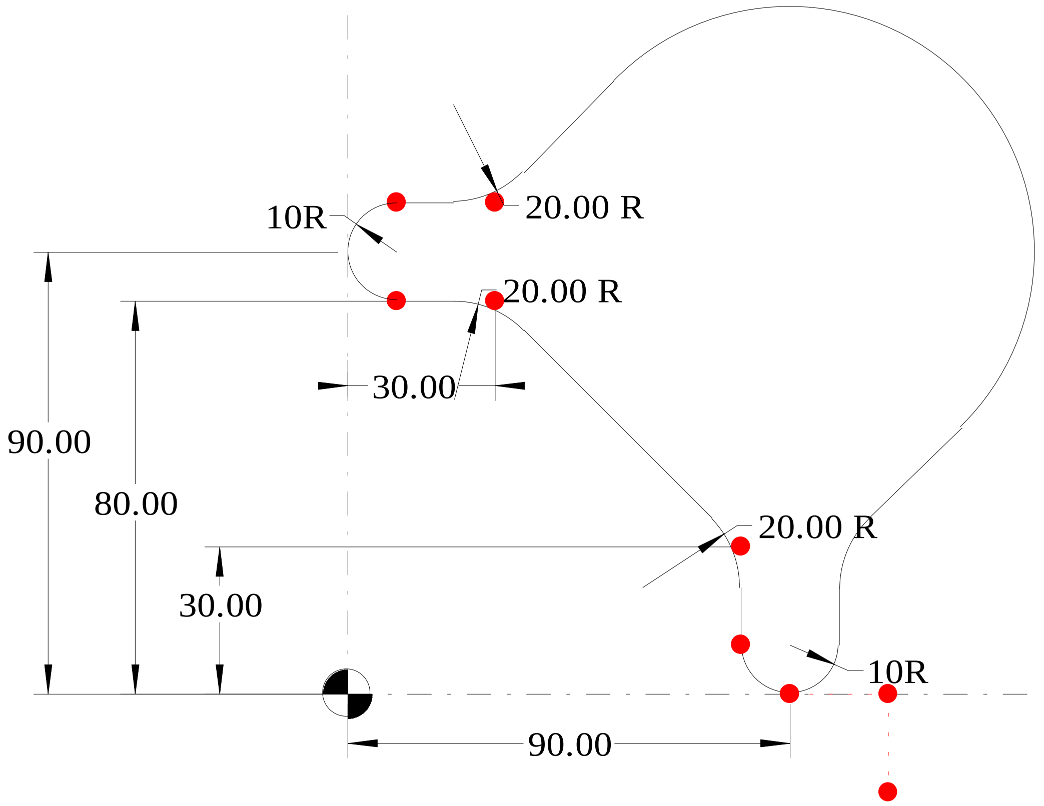
CHAPTER 14

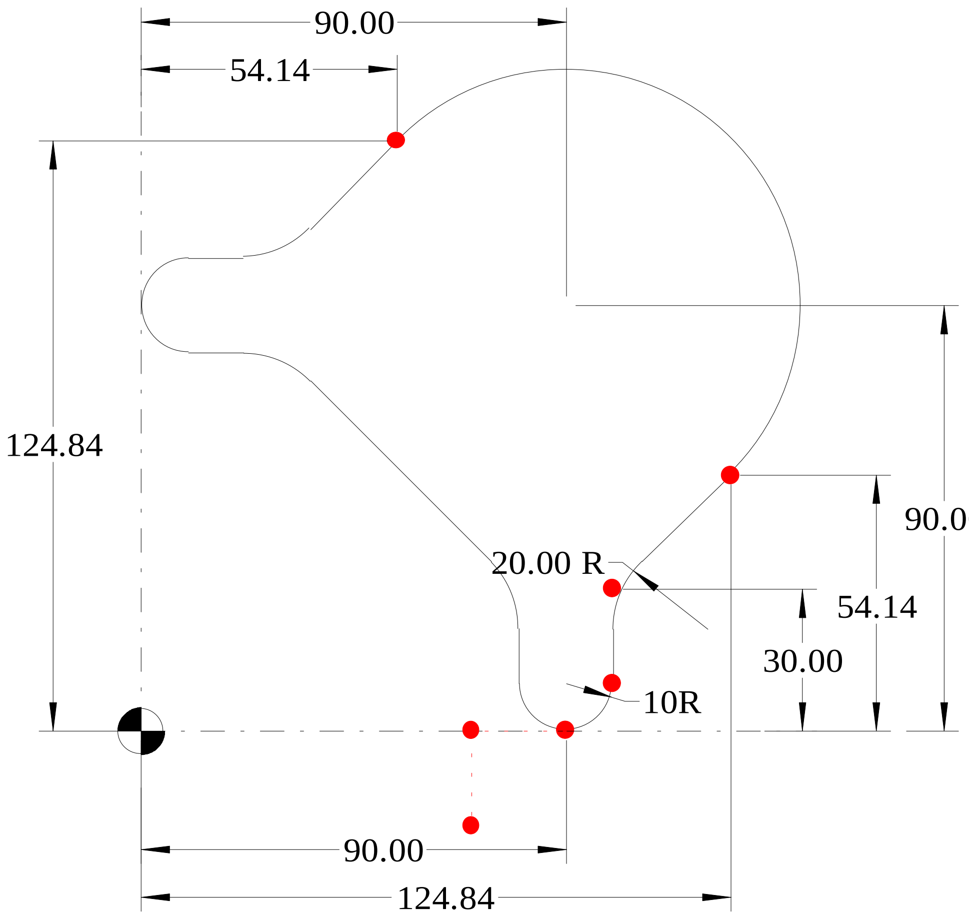


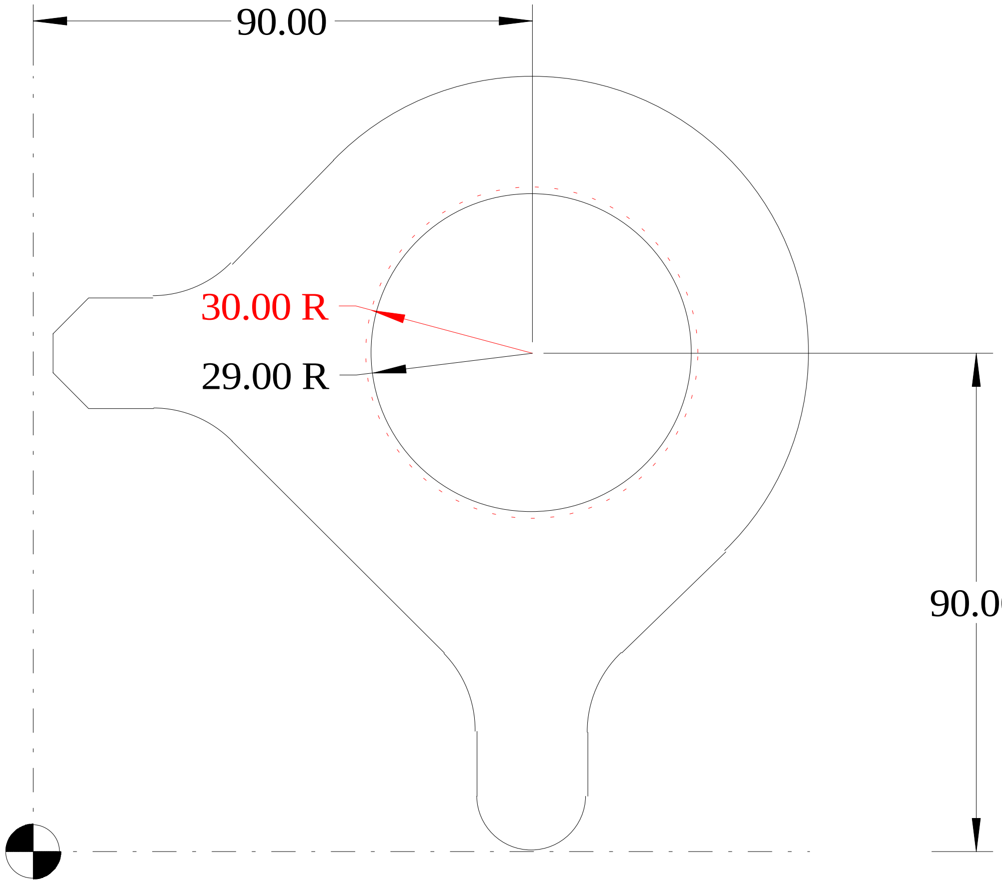
Mill Contour using Cutter Diameter Compensation
Contour and Pocket depth = 35mm

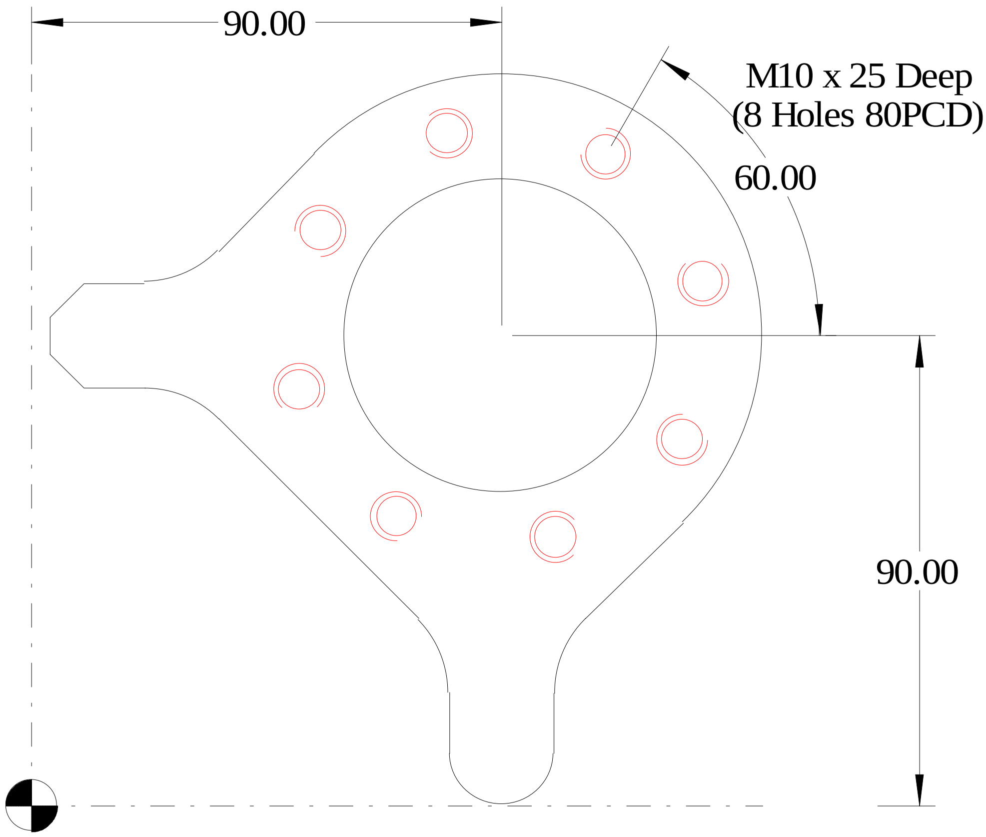
All Corner break radii = 20mm

T1 = 16mm Centre cutting Ripper.
T2 = 16mm Centre cutting Finishing Endmill
T3 = 16mm Centre Drill
T4 = 8mm Drill
T5 = M10 x 1.5mm Tap


20.00 R
10R
20.00 R
30.00
90.00
80.00
20.00 R
30.00
10R
90.00


90.00
54.14
124.84
90.00
20.00 R
54.14
30.00
10R
90.00
124.84


90.00
30.00 R
29.00 R
90.00


90.00
M10 x 25 Deep
(8 Holes 80PCD)
60.00
90.00

```
:T1 M6
G0 G90 G40 G71 G17 G94
X106 Y-16 Z100 H1 S? M3
Z5
G1 Z-35 F?
G41 Y0
X90 M8
G2 X80 Y10 P10
G1 Y30 ,R20
X30 Y80 ,R20
X10
G2 X10 Y100 P10
G1 X30 ,R20
X54.14 Y124.84
G2 X124.84 Y54.14 I90 J90
G1 X100 Y30 ,R20
Y10
G2 X90 Y0 P10
G1 X74
G40 Y-16
G0 Z10
G26.1 X90 Y90 Z-35 R0 U30 Q2 K5 E? P66 L0 I0.5 J0
G0 G90 Z100 M1
```

```
:T2 M6
G0 G90 G40 G71 G17 G94
X106 Y-16 Z100 H1 S? M3
Z5
G1 Z-35 F?
G41 Y0
X90 M8
G2 X80 Y10 P10
G1 Y30 ,R20
X30 Y80 ,R20
X10
G2 X10 Y100 P10
G1 X30 ,R20
X54.14 Y124.84
G2 X124.84 Y54.14 I90 J90
G1 X100 Y30 ,R20
Y10
G2 X90 Y0 P10
G1 X74
G40 Y-16
G0 Z10
G27 X90 Y90 Z-35 R0 U30 Q4 K35 P66 J0.5 I0
G0 G90 Z100 M1
```

```
:T3 M6
G0 G90 G40 G71 G17 G94
X90 Y90 Z100 H1 S? M3
Z5
G39 D80 P60 K8
G82 R0 Z-5 F? M8
G37
G0 G90 Z100 M1

:T4 M6
G0 G90 G40 G71 G17 G94
X90 Y90 Z100 H1 S? M3
Z5
G39 D80 P60 K8
G83 R0 Z-30 K5 J13 F? M8
G37
G0 G90 Z100 M1

:T5 M6
G0 G90 G40 G71 G17 G94
X90 Y90 Z100 H1 S? M3
Z5
G39 D80 P60 K8
G95 G84.1 R0 Z-25 F1.5 M8
G37
G0 G90 Z100 M1
M26
G98.1 X250 Y500
M30
```