DM 2400
DYNA MILL
PROGRAMMING & OPERATING MANUAL

2400 C461 V
4800 7IC

TABLE OF SECTIONS

SECTION 1
INSTALLATION

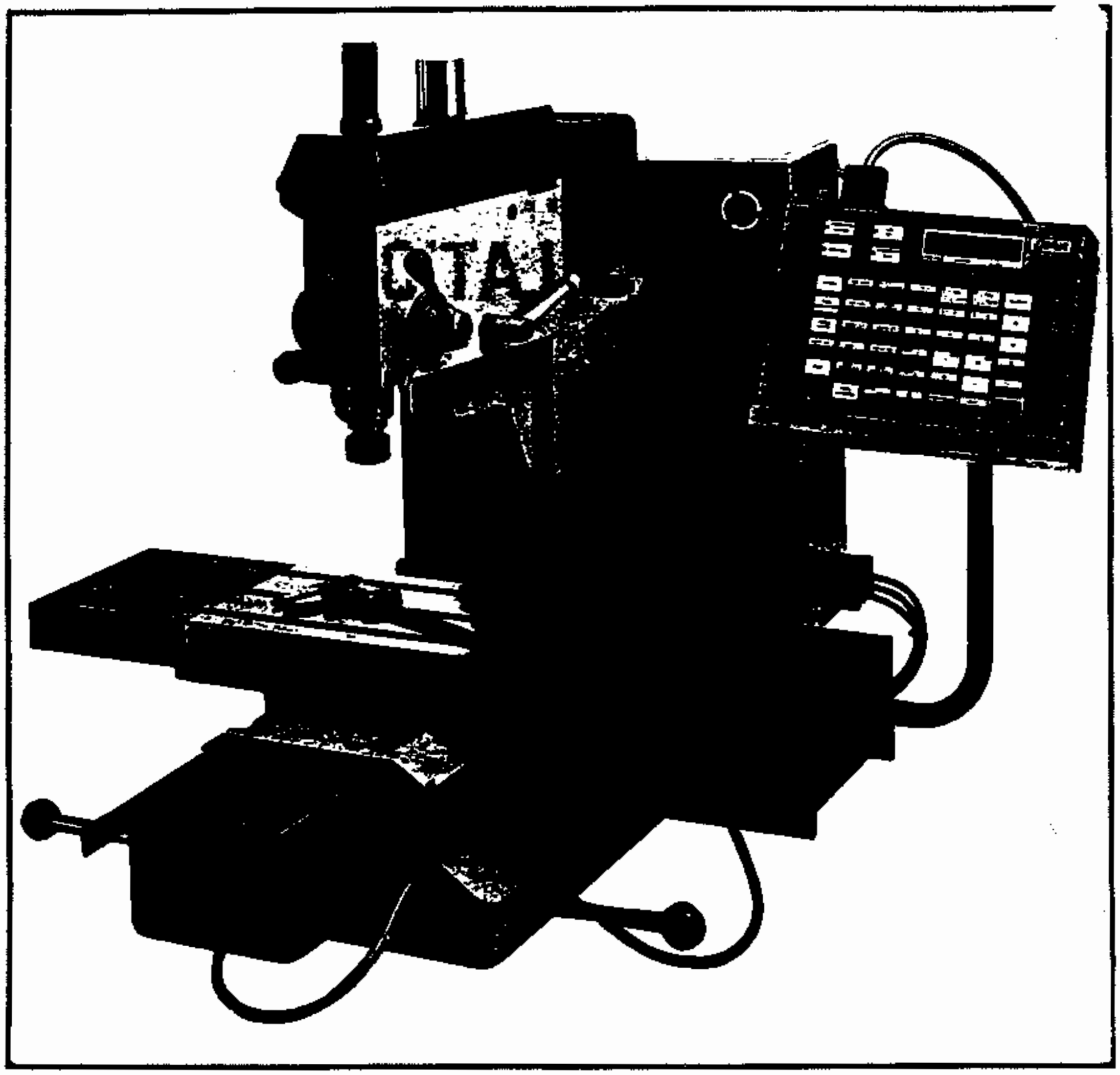


FIGURE 1 DYNA MYTE 2400 OPERATING CONTROLS

SAFETY RULES AND PRECAUTIONS

1. KNOW YOUR MACHINE - Read the Operating Manual CAREFULLY. Learn the machine features, applications, and limitations. Follow all recommended operating procedures.

2. GROUND THE MACHINE - Follow instructions for grounding as described in the manual.

3. KEEP WORK AREA CLEAN - Cluttered areas and benches invite accidents.

4. AVOID DANGEROUS ENVIRONMENTS - Do not use this machine in damp, wet, gaseous or explosive locations. Keep work area well lighted.

5. KEEP SMALL CHILDREN AWAY - Small visitors should be kept away from the work area.

6. DO NOT FORCE THE TOOL - It will do a better and faster job in removing material.

7. USE THE RIGHT TOOL - It will do a better and faster job in removing material.

8. WEAR PROPER CLOTHES - Loose clothing and ties can get caught in moving parts.

9. USE SAFETY GLASSES - Most cutting tools can throw dangerous and hot chips. Wear a face and dust mask if the cutting operation creates dust.

10. SECURE WORK - Use a clamp or vise to hold work. It is safer than using ~~your~~ hands and it frees both hands to operate the machine. *NEVER Hold work w/ hands*

11. MAINTAIN TOOLS WITH CARE - Keep tools sharp and clean at all times for best and safer prformance.

12. DISCONNECT MACHINE - When not in use. *(avoids lightning damage also use surge suppressor)*

13. REMOVE ADJUSTING KEYS AND WRENCHES - Form a habit of removing adjusting wrenches and keys before operating the machine. Do not leave parts or tools on the table.

14. KEEP HANDS AWAY FROM CUTTING EDGES AND MOVING PARTS.

15. DO NOT USE AN AIR GUN TO CLEAR CHIPS - This can blast the chips in between the slides and through the louvers into the electronics...VACUUM OR BRUSH ONLY.

16. <u>NEVER RUN SPINDLE WITH COVER REMOVED</u> - Exposed belts and pulleys are dangerous.

UNPACKING AND INSPECTING

Unpack and inspect the machine as soon as possible after receipt. Save all packing materials until inspection is complete. These materials may be required for re-shipment should you find any damage.

The machine is bolted to a wooden pallet by four bolts. The cover is bolted to the pallet by ~~twelve~~ 8 screw bolts around the base of the cover. These should be removed first and the cover raised up and over the machine. The four bolts that hold the machine to the pallet can now be removed by partially sliding the pallet over the edge of the table to expose the bolt heads from underneath. The machine can then be placed where desired (two people required). Save the bolts as they might be useful in permanently attaching the machine to a bench.

Inspect the machine for signs of damage. If there is any indication of damage, file a claim with the carrier. For other damages refer to the WARRANTY Section of this manual.

STANDARD ACCESORIES

Open the tool box and shipping box containing the machine accessories. They should contain the following items:

ITEM DESCRIPTION	QUANTITY
1. Wrench for collet nut	1
2. Wrench for spindle	1
3. Hex wrench keys	8
4. Double ended wrenches	4
5. Screw drivers	2
6. Oil can	1
7. RS232 Cable	1
8. Drain hose (2400)	1
9. Protective cover (2400)	1
10. T-handle hex wrench (2400)	1
11. Leveling Bolts (2400)	4

These accessories are used for adjustment, operation, and routine maintenance of the machine.

OPTIONAL ACCESSORIES

Optional accessories and the machine controller are packaged separately. Operation and assembly instructions for the optional accessories are included in their own shipping cartons. Operating instructions for the controller are contained in this manual.

PREPARATION FOR INSTALLATION

Refer to Figure 1 for identification and location of the various operating controls and features of the machine.

For shipment the sliding tables and the power head are positioned in their most compact position and are locked in place with the slide locking levers. Unlock these levers by turning them counter clockwise half a turn.

Remove the shipping spacer between the head and the table. Clear off all shipping material debris from the machine using a cloth or a soft brush. Exercise care not to sweep any shipping debris into the table slides. Wipe the anti-rust protective coating from the table and sliding rails using a soft cloth moistened with a light oil or WD-40.

Open the carton containing the machine controller. Remove th packing material. Inspect for physical damage and dust off ar packing debris from the surface of the controller using a clean soft cloth. Do not rub the display window excessively, as it can be easily scratched.

ASSEMBLY

Position the controller mounting bracket by loosening the four bolts at the rear and bottom of the machine base.Slide the bracket out until it is approximately 2-4 inches away from the right side of the power supply. Re-tighten the bolts after the bracket is so positioned. Place the controller into the mounting bracket.

The controller is plugged into the polarized connector that is attached to the cable going into the top right hand side of the power supply module at the back of the machine.

Adjust the angle of the bracket by loosening the single bolt behind the bracket which permits the bracket to be swiveled.

Adjust the angle of the bracket for best visibility of the display and convenient operation of the keyboard and re-tighten the bolt.

INSTALLATION

The DYNA 2200/2400 is a light (200 lb./290 lb.) bench top type machine. It can be installed on any bench top capable of carrying its weight. It can be either permanently bolted down or left to stand freely on its base or leveling bolts. For a bolted installation, a hole pattern as shown in Figure 2 should be drilled in the bench top, and the machine attached to it by using the four shipping bolts.

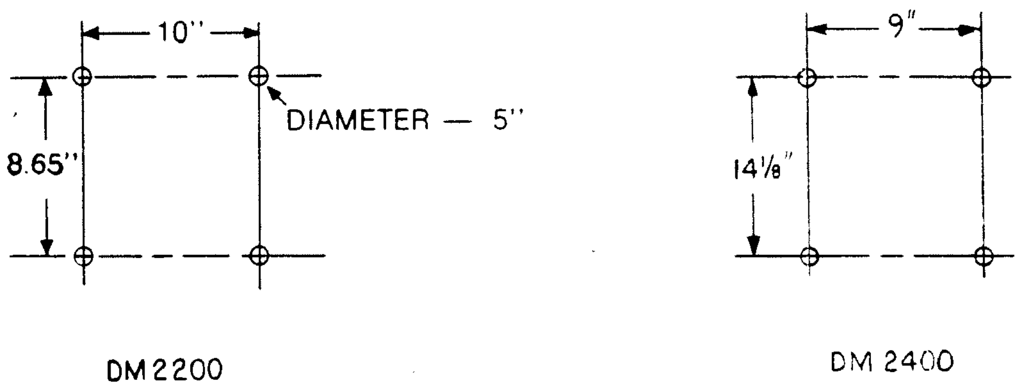


Figure 2-Mounting Hole Pattern

The bolts are installed from the underside of the bench top into the threaded holes provided in the bottom surface of the machine base. Use a rubber washer (about 2 inches O.D.,0.625 inch I.D. and 1/8 inch thick) at each bolt location and between the machine base and the bench top. DO NOT tighten the bolts severely. Excessive tightening will distort the base and cause binding in the sliding tables.

POWER REQUIREMENTS

The machine requires a single phase power source of 120 VAC (220 VAC with optional power package), 50/60 Hz. Power requirements are 350/500 watts for the 2200 and 2400, respectively. If the voltage drops below 108 volts due to start-up of other machines, the machine will perform erratically during this "brown-out" period. To protect operating personnel the machine power cord is equipped with three conductors and a three prong plug. When the machine is plugged into an appropriate three prong grounded receptacle, it is safely grounded.

CAUTION

The power receptacle supplying the power to the machine MUST BE a three prong grounded type. DO NOT CUT THE ROUND GROUNDING PRONG OFF THE MACHINE POWER PLUG. Doing so is UNSAFE and voids all warranty. The wiring polarity to the receptacal must also be correct, (i.e. white wire to the SILVER terminal). If it is not the machine keyboard operation will be erratic. To preserve the protection feature when operating the machine from a two prong outlet, use a three prong to two prong adaptor permanently connected to a known ground. DO NOT attach the pigtail of the adaptor to the cover mounting screw since these are not always connected to a ground.

If an extension cord is used between the machine power cord and the power receptacle, the extension cord MUST BE a 3-wire cord to permit proper grounding of the machine. The wire size of the extension cord should be NOT LESS than that indicated in the following table:

LENGTH OF EXTENSION CORD	WIRE SIZE
25 to 75 foot	18 AWG
100 foot	16 AWG
200 foot	14 AWG

If a transformer is to be used, ensure it is of adequate wattage, and grounds connected on either side.

START-UP 6 7

CHECK LIST BEFORE TURN-ON

Before plugging the machine into the power receptacle, check the following:

1. The main power switch on the left side of the rear housing is in the OFF (un-pressed) position.

2. Rotate the spindle by hand to assure that it is free and there is no excessive friction.

3. The controller is plugged into the power supply.

4. The cord from the spindle head is plugged into the power supply.

5. The X, Y, and Z cables are plugged into the power supply.

6. The locking levers on the three slides are unlocked (positioned fully counter clockwise).

7. The Quill is free to move up and down by the use of the Quill lever. Position it in the uppermost position.

8. The spindle head can slide up and down on the head elevator post. To do this, suport the head firmly at its bottom surface, unlock the head locking wheel/lever by rotating it counter clockwise and move the head up and down. Position it at a convenient location in the vertical position and align it so that it is perpendicular to the X axis table. Tighten the head locking wheel/lever so that the head position is firm.

9. Rotate the spindle speed dial through 360 degrees to assure that it is free and position it at the lowest speed

10. Lift the cover off the head and make sure the driving belt is mounted on the pulleys.

After the above checks have been made, plug the power cord into the power receptacle. You are now ready to machine as described in the next section.

(OVER)

DYNA ELECTRONICS INC.
1234 Elko Drive
Sunnyvale, CA 94089
(408) 734-0270
Twx: 910 3399526 SUVL

(START UP CON T'D)
see PG 1-7 first

POWER ON PROCEDURE

first connect keyboard to cable

1. Turn power ON
2. Display → "READY"→ answer "NO"
3. Emergency move →"AXIS?" select Z axis
4. With the up jogging button, move Z axis up allowing wood block to be removed. Remove wood block.
5. Touch the same up jogging button, Z axis will stop
6. Press the "NEXT" key - display "READY?" ~~answer "YES", machine will initialize~~

** Must touch again to stop

now to pg 2-6

SECTION 2
MECHANICAL and ELECTRICAL OPERATION

MECHANICAL OPERATION

The machine is generally operated from the controller keybcard. However, there are several mechanical adjustments and operations which must be done manually. These adjustments are described in this section.

SPINDLE SPEED ADJUSTMENT

The range of the spindle speed is 0-10,000 RPM. This is achieved in two steps by a change in the position of the spindle drive belt as shown in Figure 3.

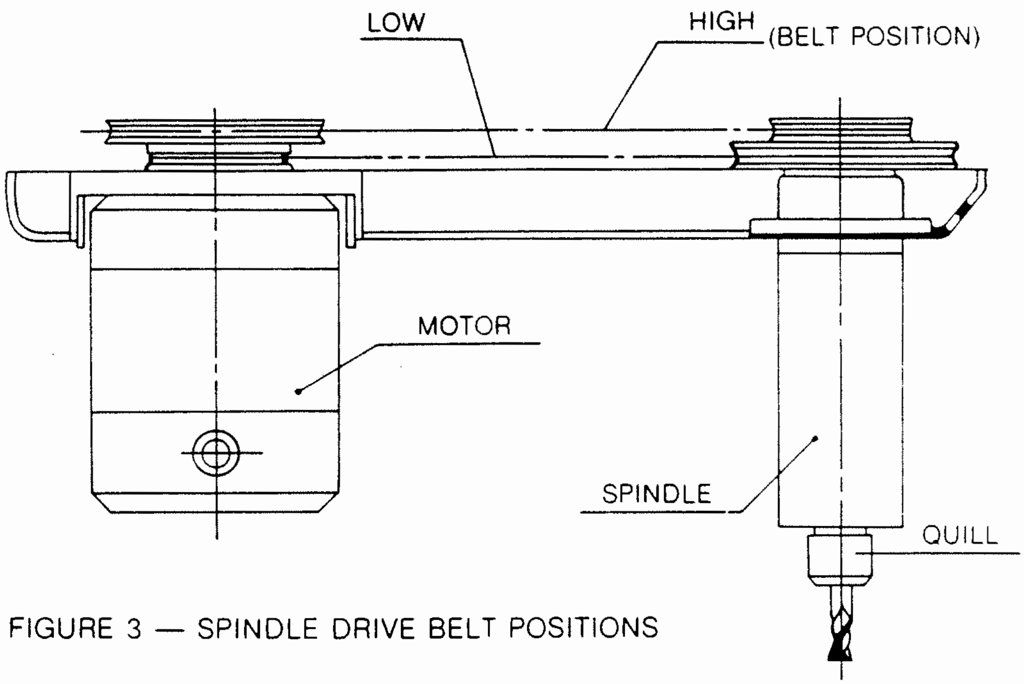
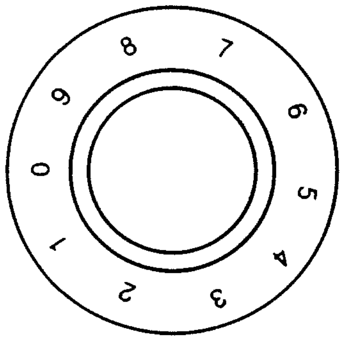


FIGURE 3 — SPINDLE DRIVE BELT POSITIONS

Within each step the speed is continuously adjustable with the Variable Spindle Speed Control. This control is calibrated from 0 to 9 and the relationship between the calibration points and spindle speed are shown in Figure 4.



VARIABLE
SPINDLE SPEED
CONTROL

DIAL NUMBER	MOTOR RPM	SPINDLE SPEED L	RPM H
1	457	217	446
2	1089	516	1063
3	1795	851	1751
4	2638	1250	2574
5	3622	1716	3534
6	5158	2444	5033
7	7118	3373	6945
8	9932	4707	9691
8.7 (MAX.)	10776	5107	10514

Figure 4 Spindle Speed Control Settings

The Spindle can be turned ON and OFF either manually or by an instruction in the program. For MANUAL control the Spindle ON/OFF control switch, which is located on the left side of the power supply must be in the LOCAL position. The spindle can be turned ON/OFF by the switch on the side of the spindle head.

With the spindle switch in PROGRAM position the Spindle will be turned on when the program encounters the SPINDLE ON instruction and off when the program encounters the SPINDLE OFF instruction.

The switch at the Spindle head should be turned ON in this mode before running the program. The program ON command may be by passed during a program run by turning this switch off.

SPINDLE HEAD POSITIONING

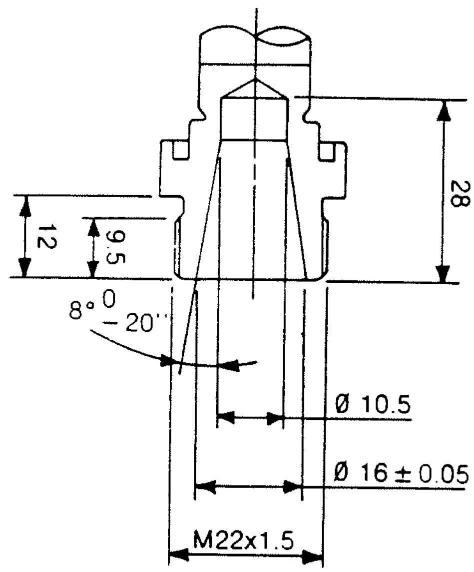
The entire spindle head can be lowered, elevated, and rotated on the head post. This is used to accomodate different tool heights and to provide greater access to the workpiece on the table. The head is adjusted by loosening the head adjusting wheel/lever on the right side of the head. CAUTION--hold the head firmly at the bottom when it is loosened otherwise the head will drop under its own weight. Adjust the height and position of the head as desired and tighten it. MAKE SURE THAT THE WHEEL/LEVER IS TIGHTENED FIRMLY, OTHERWISE THE HEAD WILL ROTATE UNDER THE LOAD ON THE SPINDLE DURING THE CUTTING OPERATION. As will be described later, the controller will be informed as to the actual position of the spindle relative to the table so the initial rotary position of the head is NOT IMPORTANT. However, if this position shifts during machining due to the machining forces on the tool and spindle then position accuracy will be lost. The head tightening wheel/lever must be tight enough to assure that the head does not rotate during machining.

QUILL ADJUSTMENT

The Quill can be lowered and elevated using the quill lever on the right side of the head. This permits easy adjustment for tool height during the set-up operation and during manual drilling. To adjust the quill position, the quill locking lever must be unlocked by rotating the lever counterclockwise one half turn. The quill is positioned by the quill lever to the desired height and then the locking lever is tightened by rotating it clockwise until it is tight. The quill travel can be limited at both the top and bottom positions. This control can also be used to obtain fine movement of the quill. This feature is used for control pecking increments in manual drilling.

COLLET INSTALLATION AND ADJUSTMENT

The spindle nose of the machine is equipped with a tapered nose. This provides high centering accuracy and permits rapid change of tools, collets and accessories. The geometries and dimensions of the nose are shown below. Special tool holders for a variety of tools and a range of precision collets from 1-10MM in diameter are available as optional accessories. Mounting of these are shown on the next page.

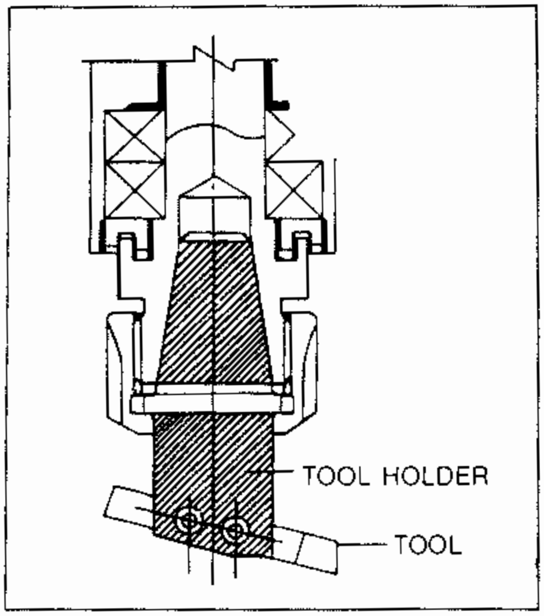


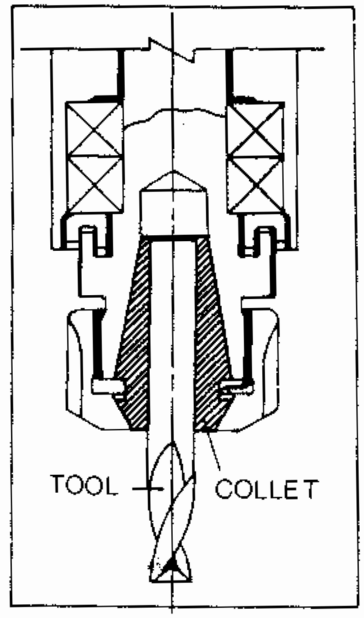
SPINDLE NOSE DIMENSIONS

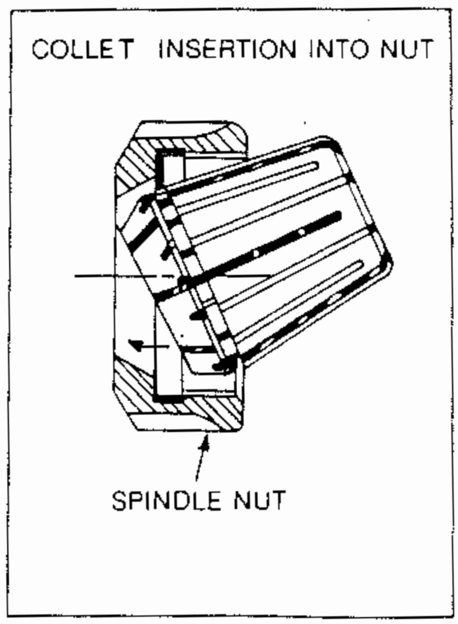
1) Insert the collet into the nut at an angle and engage the extraction tongue in the groove of the collet.

2) Screw the nut onto the collet holder (with the collet held in the nut).

3) Insert the tool to be gripped and then lock the nut using the two wrenches provided as standard accessories. One wrench is used to hold the spindle and the other to tighten the collet nut.







TOOL MOUNTING DIAGRAMS

START-UP

CONT'D

ELECTRICAL OPERATIONS

Turn emergency stop button to the right to ensure it is disengaged. Turn on the machine by pushing the main power switch. The indicator light within the switch will light and the machine will automatically perform the following operations:

1. The display will ask READY? Press YES (pressing NO will allow the user to move each axis. See machine halting in the next section.

2. The X and Y tables and the Z axis head will go to the home position, Z first then X & Y.

3. The machine will measure the backlash on all three axes and will show the value of each (in MM) sequentially in the display.

4. The display will indicate:

MODE?

This is a prompt (i.e. a request) to the user asking him to select the desired operating MODE. All operations of the machine are done from the keyboard of the controller.

BACKLASH MEASUREMENT

This measurement is very important for the correct operation of this machine. The user should switch the machine off and on several times to ensure that the values are consistent. They should range within .005 mm each time. This measurement is made on each axis electronically at switch-on by means of a special contact/pogo switch. These should be kept scrupulously clean. Both the X & Y axis have covers over these switches.

If the backlash measurements are erratic, this is usually a very good indication of particle or oil contamination. These contacts should then be cleaned with a light cloth. of switches

If the backlash measurement is too large (the display will show this) it is an indication of either oil dry-out on the slides or the slide adjustment is too tight, (check the slide lock first). The user should only adjust the tapered slide screws as a last resort. (For the 2400, the wiper holders mut be removed first).

If the axis does not move, please check the service manual.

SECTION 3
PERIPHERALS and ACCESSORIES

MACHINE ACCESSORIES

1. MECHANICAL

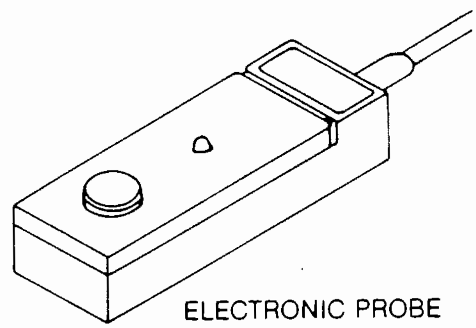
On the next page is a drawing of the tooling for the machine series. Further accessories are covered in the accessory brochure.

The ESX 16 collets are ideal for operations requiring only one tool. However the tool height is not repeatable for production use. For multi-tool operations in production, tool holders are required. Built into the language is the capability for TOOL CHANGE under a variety of circumstances.

2. ELECTRICAL

1. ELECTRONIC PROBE

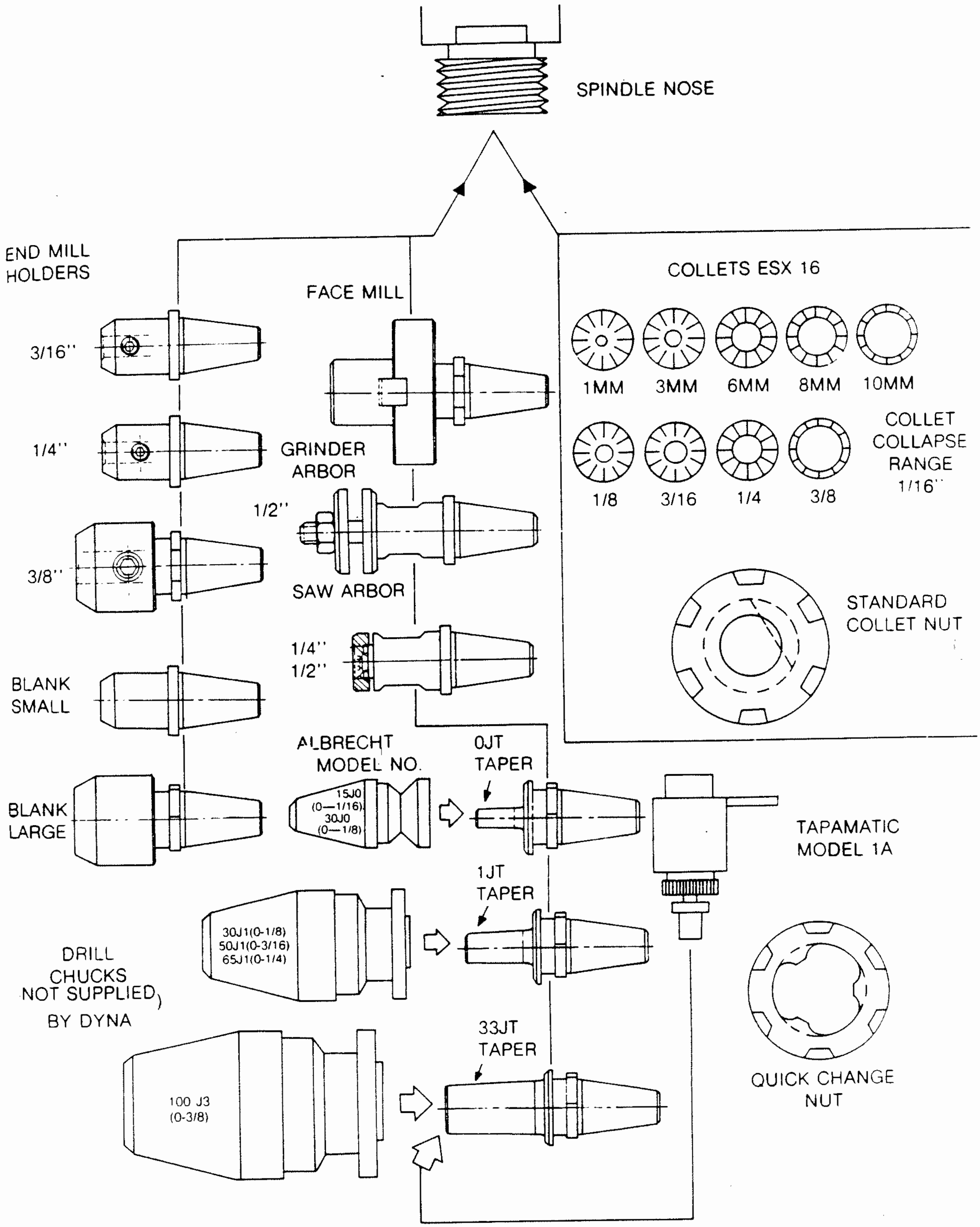
Opposite is an optional accessory called Electronic Probe that plugs into the power pack at the side. It is very useful for setup operations and is necessary for tool calibration if the user intends to use pre-set tool holders. Each one has slight variations in height (measured from the base to the top of the button) and the user enters this value into the controller under TOOL CALIBPROBE in millimeters.


ELECTRONIC PROBE

2. FOURTH AXIS

This is an optional-stepper controlled rotary table accessory which plugs into the power section at the back. It is also built into the language as the U-axis.

TOOLING FOR DYNAMYTE 2400



3. INTERFACE CONSOLE

The user may have an optional desktop interface console. It is shown on the next page. The basic unit without the printer or cassette option allows the user to enter programs at his desk directly into the CMOS memory of the controller. He can then unplug the controller and replug it into the machine to run his program. He may also run his program at the desk for time estimation.

After entering the program, press the program run key, answer YES to NONSTOP?. It will halt at SET UP. The user will have to go through this set up procedure to position the REF COODS correctly. Pressing the NEXT will restart the program run.

With the interface console, the user may add a 16 column alphanumeric printer and/or digital micro-cassette for off-line storage.

1. 16 CHARACTER ALPHANUMERIC PRINTER

Simply plug the unit into the back of the interface console, push the paper feed switch, and the paper will feed through the printer. To print out a program, go to LINE MODE and position the display at the PROGRAM START line then press the shift and READ/WRITE keys. The display will ask

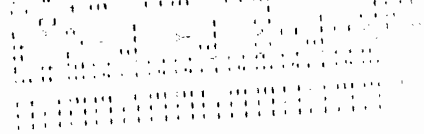
PRINTER ?

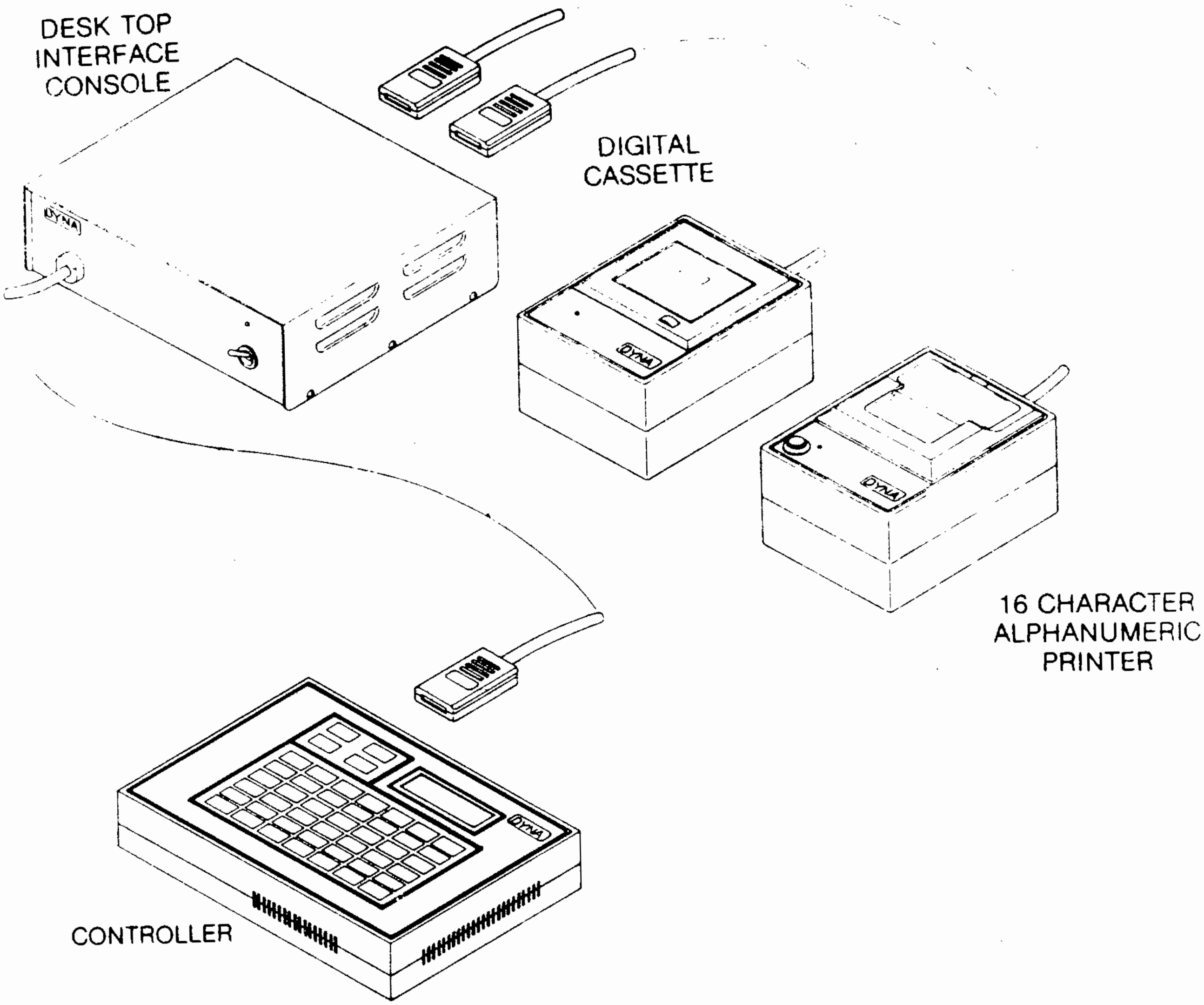
Answering YES will start the printer to print line by line, automatically. It will halt by itself when it comes to the END statement. It may be restarted by pressing the shift and READ/WRITE keys in the LINE MODE again. Pressing HALT will stop the printer, pressing NEXT will restart it. Pressing a mode key will exit the controller from the printer.

We will print directly from pc file

The user will need to push the paper feed key manually if he wishes blank space beneath his program. Loading a new roll of paper is done by feeding the paper through the roller and pushing the paper feed key.

What kind of paper?





INTERFACE ACCESSORIES

4. THE MICRO-DIGITAL CASSETTE

Plug this in at the back of the interface console. The red light will come on to indicate power when the power switch is turned on. Now a word of explanation about how the data is organized. The controller holds 901 instruction lines, this is the line number amount available to the user. We call this a FILE. The FILE may contain a program of only 2 lines in length or it may be filled completely with a 901 line program.

One micro-cassette will store 40 FILES. The user may choose to have 1 program per FILE per micro-cassette or he may choose to put several programs in each FILE with 40 FILES per cassette. If there exists a strong probability of cassette loss or damage, then the former situation may be preferable to the latter.

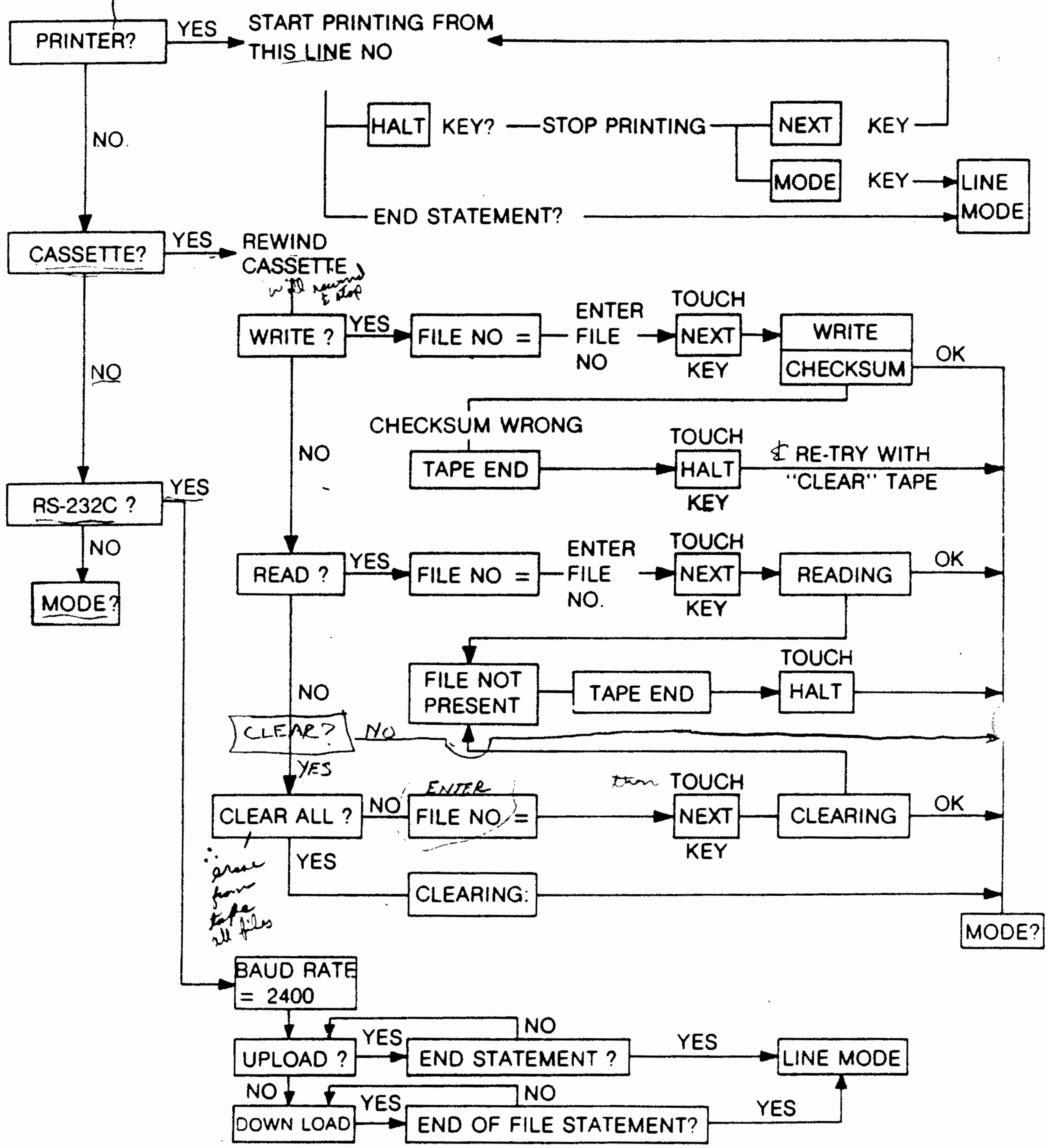
Each FILE has a number. It is up to the user to remember which program resides in which FILE. When the user wishes to store his current program or set of programs in the controller, he switches to LINE MODE, answers "NO" to PRINTER?, and "YES" to CASSETTE? The controller will ask him WRITE?, he answers "YES." The controller will then ask for the FILE NO. assignment. The user enters the FILE NO., hits NEXT key, and then the whole controller memory will be moved into the cassette. If the user has a file in tape which is no longer used or required, it is necessary to clear the FILE NO. first with the clear instruction (answer "NO" to WRITE?, "NO" to READ?, "NO" to CLEAR ALL?. Enter the FILE NO. to be cleared and press the NEXT key.

A visual picture of the cassette would be a row of shelves with numbers on them. You may put in or take out one controller memory (file) at a time.

To read back, the user ("NO" to WRITE?, "YES" to READ?) enters the required FILE NO. and it will be automatically transferred across. It will destroy any program currently residing in program memory. If the requested FILE cannot be found, the controller will so tell the user with a TAPE END message.

On the next page is the flow diagram for the printer and cassette.

FLOWCHART FOR READ/WRITE



INSTRUCTIONS

(1) GO TO LINE NO. MODE SELECT THE LINE WHERE THE PROGRAM STARTS

(2) TOUCH THE READ/WRITE KEY

(3) FOLLOW THE FLOW CHART

SECTION 4
CONVENTIONS

OPERATING RULES AND CONVENTIONS

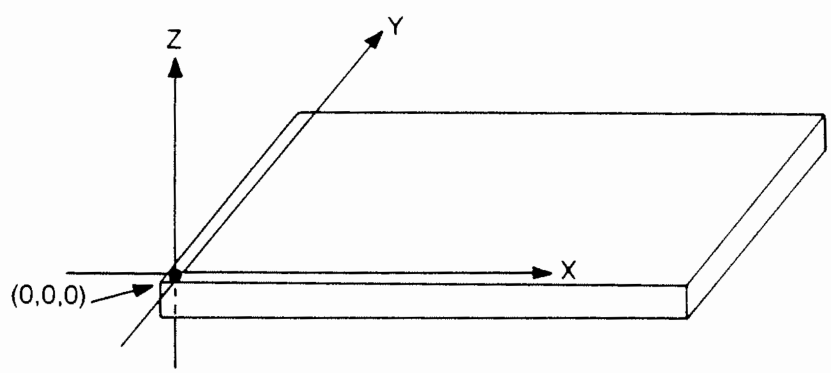
In operating and programming the machine there are certain rules and conventions which MUST BE FOLLOWED EXACTLY as they are defined. All machine moves are controlled by micro-computers in the controller. These micro-computers will recognize instructions only if they are entered in a certain way and in a specific sequence. All moves are executed by either the program in the controller or by pressing certain keys on the membrane keyboard of the controller.

Some of the general purpose rules and conventions are described in this section. Others, particularly those that affect programming, are described in other sections of the manual.

1. AXIS CONVENTION

Although the X and Y tables and the spindle head (Z axis) are the ones that actually move, it is convenient to always view any movement as a MOVEMENT OF THE **TOOL TIP**, since this is the point that will be doing the actual cutting on the surface of the workpiece.

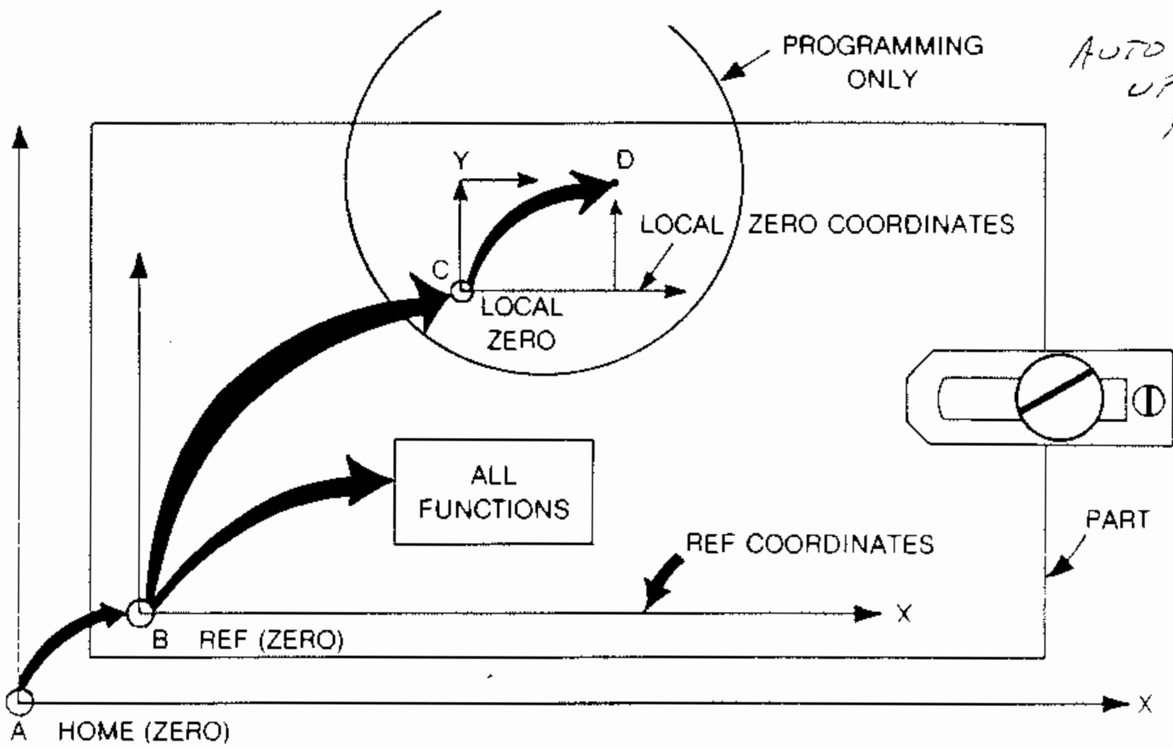
The diagram below shows the axis convention which has been adopted to describe the position and motion of the tool tip.



The three axes X,Y, and Z are always orthogonal or mutually perpendicular to each other. The intersection point of the three axes is defined as the Zero coordinate point and is expressed as (0,0,0). The tool tip position is always expressed in terms of its coordinate point in the following order (X,Y,Z). The numeric value and sign of each coordinate point is always relative to the Zero coordinate point. The signs of the coordinates are positive if the tool tip is positioned in the positive space and negative if it is positioned in the negative space as defined in the above diagram.

For example, if the tool tip is positioned two units to the right of zero the coordinate point is (2,0,0). If it is positioned 1 unit down along the Z axis and two units to the left along the X axis, its coordinate point is (-2,0,-1).

The Zero coordinate point can be placed anywhere we want it to be. Later in the manual we will show why it is advantageous to be able to place the Zero coordinate at different locations. For now we will simply define and describe three positions of the Zero coordinate which are often used in machining and programming. These positions are defined as the HOME Zero, the REFERENCE Zero and the LOCAL Zero. Their relationship to each other is shown in the diagram below:



The HOME zero (point A) is defined as the position when the X,Y, and Z axes are physically at their limit switches. When the machine is first turned on, it always goes to the HOME position.

The REF zero (point B) is programmed into the machine either in the SET UP instruction or by the PROG REF instruction. It is the point to which all programmed FUNCTIONS are referenced.

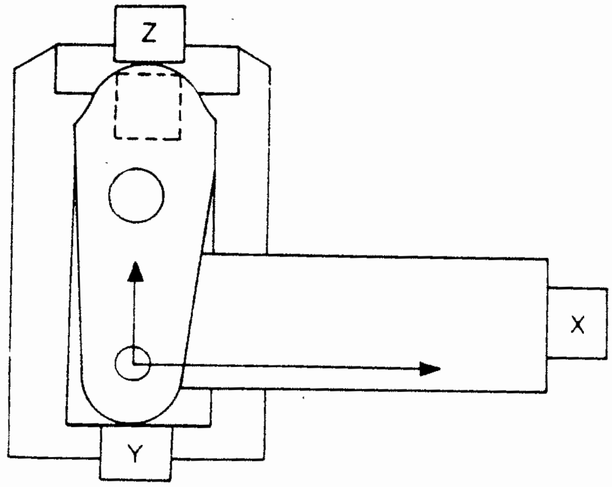
The LOCAL zero (point C) is specified by the user within a program. In programming, the user can set-up a LOCAL zero, do the operation relative to that point and then switch back to the REF zero or to another LOCAL zero.

Each of these special Zero positions is further discussed below.

2. The HOME ZERO Position

The machine, as shown in the diagram below, consists of a table which can move in X and Y axes and a spindle head which moves in the Z axis.

1. HOME POSITION



When facing the front of the machine, the movements are as follows:

X slide moves	LEFT ⟷ RIGHT
Y slide moves	OUT ⟷ IN
Z slide moves	DOWN ⟷ UP

The maximum travel in the three axes are:

X=6.2 inches
Y=5.0 inches
Z=4.0 inches

The HOME zero is defined as the position of the slides when they are at their limit switches. This occurs when:

The X slide is fully to the right.
The Y slide is fully in.
The Z slide is fully up.

The HOME zero is built into the machine. The user can not change it.

3. The REFERENCE ZERO Position

The REF Zero must be entered into ~~the program by~~ the ~~user~~. It can be placed anywhere within the working space of the machine. However it is advisable to place it so that it coincides with a point on the workpiece to be machined; preferably one to which all the dimensions of the part's machining geometry are referenced. This simplifies workpiece placement on the table during set-up and permits using most of the part's drawing dimensions directly in the program.

For example, most drawings, which show the geometry of a part to be machined, are dimensioned from the left and bottom edge of the part. If we made the intersection of these edges the machine's X & Y REF Zero then we can use the dimensions given on the drawing directly, without doing any recalculation, as an input to the program. Likewise the depth of cuts on the drawings of the part are usually referenced to the top surface of the part. By using the machine's Z REF Zero we can also use all the given dimensions directly.

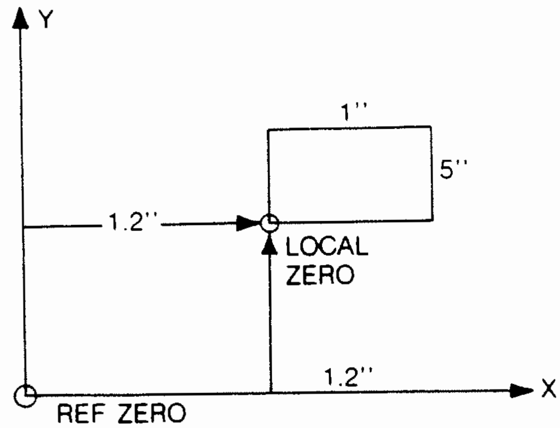
When a workpiece is first placed on the machine, we have to tell the machine where it is. A simple way to do this is to define some point on the workpiece as the REF Zero point, and then to tell the machine where that point is. The machine makes this operation easy by having a pre-programmed SET UP instruction. When the controller comes to this instruction in the program it will stop and allow the user to manually position the center of the tool tip at the desired reference point on the workpiece. This position of the X,Y tables and the spindle head is defined as the REFERENCE COORDINATE point. By entering this position of the table and head into the program, the machine then knows where that point is on the workpiece. Further details on the procedure used in SET UP are given in a later section of the manual.

If the part to be machined already posses a reference edge, then this edge must be precisely aligned parallel to the X or Y axis of the table when the part is clamped in place. The center of the tool tip must also be precisely positioned at the required point on the workpiece. If the part is to be machined out of a larger block then the block need not be precisely located since the edges of the part will be defined by the machine.

The machine controller has been preprogrammed to cut several commonly-used machining shapes. We call these FUNCTIONS. These consist of: Mill, Rect Pocket, Rect Frame, Circle Pocket, Arc Frame, Bolt Circle and Drill. All of these functions are referenced to the REF Zero.

4. The LOCAL ZERO Position

In machining several geometries on the same part, it is often not convenient to use the REF Zero point for all of them. For example, in the geometry shown below if we were to machine the rectangle



by using the REF ZERO we would have to do some calculations to find the coordinate points of the corners of the rectangle referenced to the REF Zero. On the other hand, if we define a LOCAL ZERO as shown, then we can cut the rectangle by referencing our moves to this point and using the dimensions as given in the drawing.

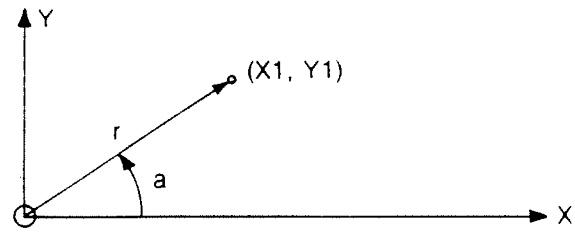
A LOCAL Zero can be set up by the user anywhere in the X,Y,Z space and anywhere in the program. It is used mainly to simplify programming and to avoid unnecessary calculations of coordinate points.

A LOCAL Zero is set up by using either the ZERO AT or the ZERO COODS instructions within the program. Both of these instructions are fully described in the Programming Section. For now we will simply say that the ZERO AT instruction tells the machine to set-up a LOCAL Zero at a specific coordinate point. The ZERO COODS instruction tells the machine to set-up a LOCAL Zero at the point where the tool is at that time.

When the controller encounters these instructions in the program it will make all moves in the program below it as relative to this new LOCAL ZERO coordinate point. It will continue this until it encounters a REF COODS instruction which tells the machine to go back to the REF ZERO or until it encounters another LOCAL ZERO coordinate instruction. It is possible to go from one LOCAL ZERO coordinate point to another or to return to the REFERENCE COORDINATES after each one.

Polar Coordinate Convention

The user may also represent a point in XY plane by means of polar coordinates.



Where $r = \sqrt{(X1^2 + Y1^2)}$ $a = \tan^{-1}(Y1/X1)$

We follow the normal convention.

A clockwise movement to decrease "a" is defined negative in a GO RELATIVE angle move.

A counter clockwise movement to increase "a" is defined positive in a GO RELATIVE angle move.

Likewise, a go relative r positive will increase r.
go relative r negative will decrease r.

Default Convention

The controller must have a "zero" around which it can make the moves.

If a local zero is not set, then the controller defaults to the REF ZERO. If the REF ZERO is not set, then it will default to the HOME ZERO.

SECTION 5
BASIC MACHINE OPERATION

THE FOUR OPERATING MODES

The machine has four operating modes. The desired MODE is selected by pressing any one of these four keys:

MANUAL	LINE NO.	PROGRAM ENTER	PROGRAM RUN

When one of these keys is pressed the indicator next to it will light and will stay lit throughout the time the controller is in that mode. To exit a particular mode, press that mode key again.

The MANUAL Mode-is used for tool calibration and for manual operation of the machine. (and also certain diagnostics).

The LINE NO.MODE-is used for review of programs stored in the controller and for editing these programs.

The PROGRAM ENTER Mode-is used for entering a program into the controller.

The PROGRAM RUN Mode-is used for operating the machine under program control.

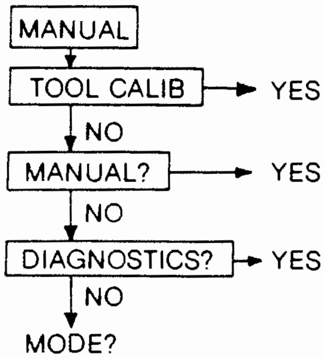
The capabilities of all four modes will be briefly introduced in the following section.

MANUAL MODE

This mode is used to:

a. Calibrate the height of tools which are mounted in tool holders.
b. Calibrate the electronic probe.
c. Move the machines X,Y, and Z axes, for manual operation (i.e. nonprogrammed machining).
d. Call up machine diagnostics.
e. Demonstrate the machine using a stored program.

When the "MANUAL" key is pressed the controller will go into a sequence of pre-programmed prompts which will guide the user in selecting what he wants to do in this mode. A prompt is a question which appears on the controller display and which asks the user whether he wants to execute that particular operation. The user MUST respond by pressing either the YES or NO key. Depending on the user's response, the controller will either ask another question or it will exit the MANUAL mode by displaying MODE?, which is a prompt asking the user to select another mode. A flow chart of the MANUAL mode prompts is shown on the next page.



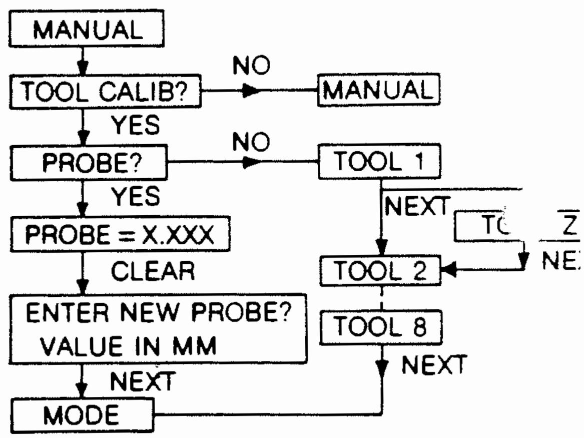
This is the prompt sequence to do Tool Calibration.

This is the prompt sequence to perform manual operation of the machine.

This is the prompt sequence for the Diagnostic routine.

Exits the Manual Mode and asks for selection of another mode.

The above flow chart illustrates that if the user wants to do Diagnostics on the machine he must answer NO to the first two prompts and YES to the Diagnostic prompt. The flow charts for each of the above prompts and a description of how to respond to them is described below as follows.



TOOL CALIB?

The flow chart for this is shown opposite. This asks if the user wants to calibrate:

1) The electronic probe.
2) A set of quick change tools.

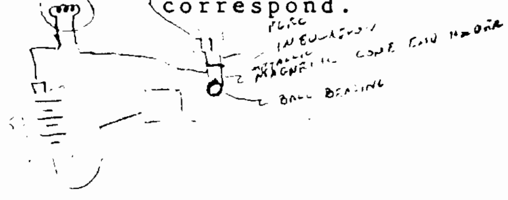
NOTE: If tool calibration is required, then it should only be done immediately following initialization.

THE ELECTRONIC PROBE

This is an optional accessory that plugs into the side of the machine that allows the user a simpler way to do the set up of the (machine) reference coordinate point. The probe is necessary for measuring the tool offsets in 2) above for the quick change tools. It detects the end of the tool when it touches the button, or the side of the tool if the probe is held at right angles.

The probe has its own offset (it's thickness) which must be entered in initially and is permanently stored in memory. Since there are slight variations in thickness, the user must measure this offset in millimeters (from button top to base), then enter it after clearing the display. Typically, it is around 14.89mm. If the measurement is in inches then multiply by 25.4).

Once entered it may be forgotten. However, if different controllers use the same machine, make sure the values correspond.



OFFSET OF QUICK CHANGE TOOLS

These are used to facilitate production runs where each part may involve several tools. The height of the tool is fixed with respect to the spindle head and not sliding up and down as with a collet. The offsets measured for tools 2 to 8 are all with respect to tool 1. In the program, tool 1 must be used during set-up. Any subsequent tool change will thus be referenced to tool 1 and the offsets will be corrected in tools 2 to 8 to make the height of the tool the same as tool 1. To calibrate the tools, the user must place tool 1 into the Spindle and position the electronic probe directly below the tool tip on a flat surface of the workpiece table. The user must then press the NEXT key. The Z axis will automatically descend until the tool tip touches the electronic probe whereupon the Z axis motion will stop and display tool Z. The user must now press the NEXT key again; this tells the machine that Tool 1 has been calibrated. The machine will respond by moving the Z axis to the home position and display tool 2.

The user must now put tool no.2 into the spindle, place the electronic probe below its tip and press NEXT. The Z axis will again descend to touch the probe. Pressing the NEXT key again will result in the tool Z displayed and Z axis going up and the display will show tool 3. This sequence can be repeated for a maximum of 8 tools.

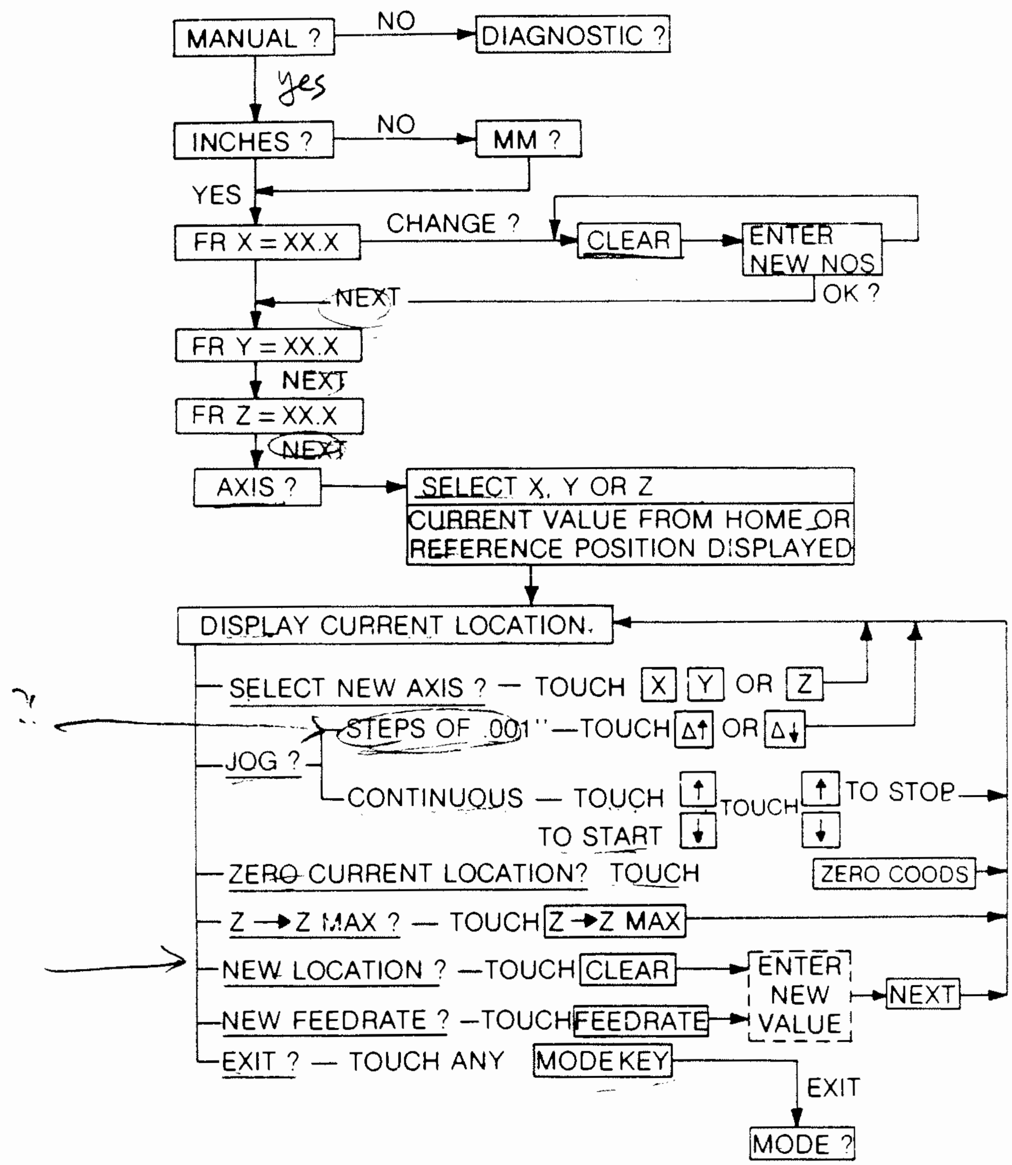
If fewer than 8 tools are calibrated, the user can simply ~~hit~~ PRESS the HALT key, and the controller will exit the sequence by displaying MODE?

MANUAL?

The purpose of this "sub mode" is to give the user a feel for the machine. It is exactly like a hand operated milling machine with a digital display on each axis, but the machine moves the handles for you. In some cases, it is much easier just to go to the manual mode, do some operations, and exit, rather than going through the program mode, especially if the user just wishes to drill a few holes. etc.

The flow chart for this prompt is shown below. In this mode the user can operate the machine manually by moving the X,Y, and Z axes, one at a time, from the keyboard. Before he can do this however, he must respond to some prompts as shown in the Flow Chart so that the machine knows at what rate to make the move and which dimensional system (ie. English or Metric) to use.

FLOWCHART FOR MANUAL?



On a YES response to the MANUAL? prompt the display will show EITHER..

INCHES?	or	MM?

If the machine had been previously used in the English units, then INCHES? will be displayed. If it had been used in the metric units then MM? will be displayed. In either case, a YES response by the user will put the machine into the next sequence of prompts. If the user response is NO the display will then temporarily show the alternate selection of units and then will continue with the next sequence of prompts.

This sequence of prompts guides the user in specifying what FEED RATES will be used on each axis and which axis will be moved. When the first FR X = XXX prompt appears it will display the value of the previously used Feed Rate on the X axis. If the user wants to use this feed rate he simply responds by pressing NEXT. If he wants to change it, he responds by..

> Pressing [CLEAR]
> Entering the new Feed Rate using the numeric keys, and
> Pressing [NEXT]

After the X feed rate entry is completed, the controller will prompt the user through the Y and Z axes and will then display.

> [AXIS ?]

The user has to enter the axis which he wants to move by pressing either X,Y,or Z keys. The display will then show the current position of that axis, i.e. If the user selects the X axis, the display will show:

> [X = 1.1605] for example

This indicates the present position of the X axis. To move the X axis to another position, the user CAN

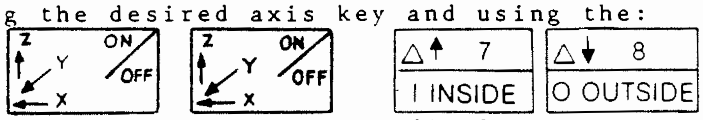
> Press [CLEAR]
> Enter the desired position, then press the [NEXT] key to move the axis to the new location.

The [↑ ON/OFF] and [↓ ON/OFF] keys will move the axis continuously at the fixed feed rate of 12 inches per minute. One may clear the previous value and enter new dimension of axis and press NEXT key, the machine will move at the feed rate previously entered. The [1 INSIDE] and [0 OUTSIDE] will move the axis in discrete steps of 0.001 inches. In all cases, the axis movement will stop automatically when the above newly entered position is reached.

In this mode, the user can also perform the following manual operations from the keyboard.

1. Move any axis from its present position to a new position simply by touching the desired axis keys, and entering the new position value into the display and pressing NEXT.

2. Any axis can be moved or jogged in either direction by touching the desired axis key and using the:



keys.

3. Zero the current value on the display by pressing:

ZERO COODS

4. Find the current position of any axis by pressing the desired axis (i.e. X,Y, or Z) keys

5. Change the Feed Rate on any axis by pressing "FEED RATE", the corresponding axis key and entering the new feed rate using the numeric keys.

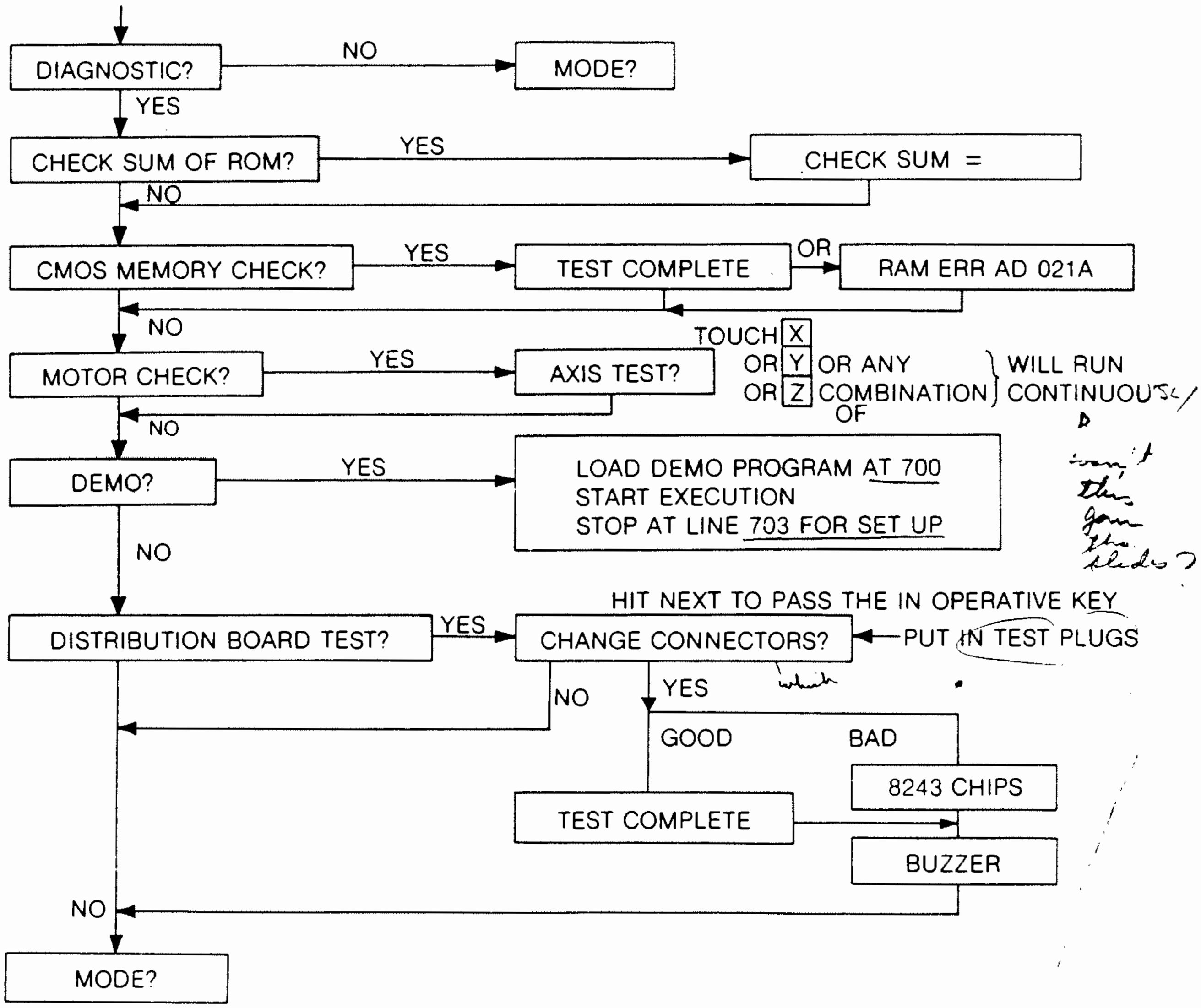
6. Move the Z axis to its HOME position by pressing the shift and then the "Z→Z MAX" key

7. The user can exit this mode at any time by pressing any other "MODE" key

Observe that until the user has zeroed a particular coordinate, the value displayed will be the distance from the HOME position or the reference position.

DIAGNOSTIC

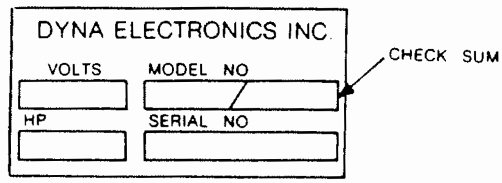
The flow chart for this prompt is shown below. As can be seen from this flow chart, the DIAGNOSTIC prompt has several branches which permit a variety of checks to be performed on the machine and controller to verify if it is operating correctly.



Are these furnished or optional? no! will travel & return home

SUMMARY OF DIAGNOSTICS

CHECKSUM OF ROM   This value should agree with the value on the back of the controller.



It is the software release number.

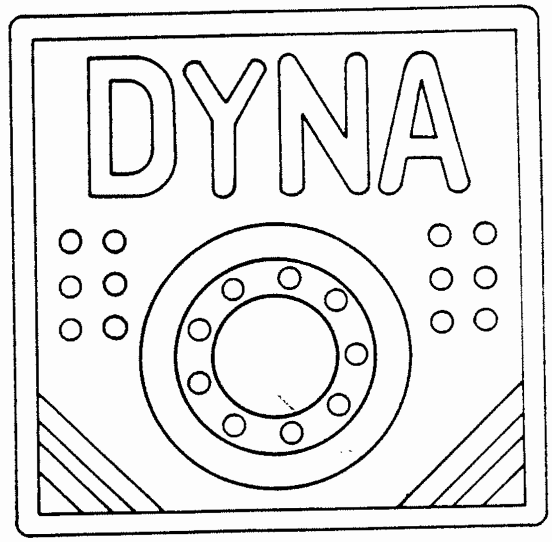
CMOS MEMORY CHECK   This checks the CMOS memory. If bad, location and chip number will be indicated.

MOTOR CHECK   At axis test? press any combination of X,Y,Z to be tested, then **press the NEXT key. The axis will travel to** its maximum distance and return to its home position. This is very useful in checking axis operation and in lubrication of slides.

DEMO   See the next page.

DISTRIBUTION BOARD CHECK   This is the board that the axis connectors plug into at the back side of the machine. When doing this test, the display will ask change connectors? The user must disconnect the axis connectors and insert dummy plugs. Pressing the YES key will initiate the test. If the test fails, the bad chip will be indicated and the buzzer will sound. This test is used to isolate a fault, either in the motor driver board or the distribution board. For further information please see the maintenance manual.

This sequence also contains a DEMO program which cuts the following geometry.



This Demo program is written for use with a .125 dia end mill and a 1/4 inch thick 6x6 inch acrylic, plexiglass or aluminum sheet. The sheet must be placed and clamped in the center of the table. This program is automatically dumped into program location 700 to 830 and will start automatically as well. It will stop at set-up to allow the user to position the tool. If called, it will destroy any programs residing in this space.

NOTE: If cutting harder materials, the feed rate will have to be decreased.

This pgm wasn't in memory, only a pgm to check full axis movement & spindle control was in memory.

LINE NO. MODE

This mode is used to select a line number for entering the program or for reviewing a current program line by line. Line numbers range from 000 to 900. Any line number can be selected by pressing three number keys. When this is done the program instructions residing at that line number will appear in the display and can be reviewed. No modification of a program statement is possible in this mode. However, this mode has some editing features, which are described below:

INCREMENTING AND DECREMENTING LINE NOS.

Line numbers can be incremented a line at a time by pressing the "NEXT" key or they can be decremented a line at a time by pressing the "PREVIOUS" key.

CLEARING THE MEMORY or SECTIONS OF THE MEMORY

Press the CLEAR key. The display will read **CLEAR MEMORY?.** Press the YES key and the display will read.

START LINE → nnn

Enter the line number and press the NEXT key. The display will then show.

END LINE → nnn

Enter the line number and press the NEXT key. The display will then ask.

nnn → nnn SURE?

Careful you can't get it back

Press YES and line numbers nnn, to nnn will be cleared. Entering 000 to 900 will clear the entire memory.

STORING/PRINTING PROGRAMS

The READ/WRITE key is activated in the LINE NO. MODE, only when the controller is connected to the interface console to permit printing and storage of programs. This is described in greater detail in the PERIPHERALS section.

see page 3-3

INSERT AND DELETE

The line mode allows the user to insert or delete a program line. Pressing INSERT (shift, previous) will automatically push the program down one line. Thus, for example if the program is as shown:

```
         000  START INS 01                000  START INS 01
         001  TD = 0.125                  001  TD = 0.125
Display  002  FR  XYZ = 8.0  will go to   002
Window   003  GO  X  2.4                  003  FR  XYZ = 8.0
                                          004  GO  X  2.4
```

On pressing INSERT at line 002, the user can then switch to program enter mode to insert an additional instruction.

Insertion can only be done if there are blank lines between the program end and the program start of the next program.
→ NOTE ALWAYS LEAVE SOME BLANK LINES FOR THIS PURPOSE

Deletion operates similarly so.

```
         000  START INS 01                000  START INS 01
         001  TD = 0.125                  001  TD = 0.125
Display  002  FR  XYZ = 8.0  will go to   003  GO  X  2.4
Window   003  GO  X  2.4                  004
```

Care must be exercised when using these instructions if skipping is used in the program.

if there is a skip to instruction in the program, del or insert will change the line numbers causing entry to occur at the wrong code lines.

PROGRAM ENTER MODE

In this mode the user can enter the program at the given line number. Pressing the "NEXT" key or the "PREVIOUS" key will confirm the entry. Program lines can be erased with the "CLEAR" key. Maximum program entry is 901 lines. For insertion or deletion of lines the user must switch to LINE NO. MODE. Programs must begin with a "START" instruction and must end with an "END" instruction.
A blank is treated as no operation instruction and will be ignored during execution. Details on how to enter a program are given in Section 3.

PROGRAM RUN MODE

Pressing this key initiates the program run mode. The user should position the display at the PROGRAM START either via the LINE MODE (enter the line number) or by backstepping through the program to the beginning of the program.

The controller goes into a precheck sequence to determine the following:

1. Does the program have a start and an end (or END NEW PART or END NEW REF)?

2. Do all REPEATS have REPEAT ENDS?

3. Do all SUBROUTINE calls have SUBROUTINES existing somewhere in the memory space?

4. Do all SUBROUTINES have SUB RETURNS?

The controller will stop in precheck at the line number in question if it can to flag the error, if not, it will indicate on the display the error. It does not check for going too far on the table, that is a run time error. If the precheck is correct the display will ask:

NONSTOP?

Do you wish to go straight from the prechecks directly to part program execution & without stop in between lines?

Pressing YES will start the program running. If NO then the display will ask:

SINGLE STEP?

Pressing YES will cause the program to be run single stepped. Each move must be activated by the NEXT key. If NO then the display will return to:

MODE?

During the program run, the controller will stop at the SET UP instruction. This allows the user to set the reference coordinates and has been explained in the program entry section. If not set, the instant the user presses the "NEXT" key, the UNSET coordinates will become SET at the machine home point and the program continues running.

The line display will indicate the current instruction being executed.

The controller recognises two kinds of halts.

1. PROGRAMMED HALT

This the user enters during program entry. The controller will halt at this instruction and will continue on the NEXT key.

During halt the user **can exit, and in fact run** another **program** elsewhere, (providing there is no alteration of fixed parameters like SET UP, TOOL DIAMETER, etc.) then RE-ENTER this program at the same halt location and continue running by pressing program run exactly as if it were at the start.

2. PRESSING HALT DURING PROGRAM RUNNING

This is not an instantaneous halt. The controller will halt at the end of the current move and can be re-started by pressing the NEXT key. As in Programmed Halt, the user can do exactly the same moves, however, if the halt occurs in the middle of a FUNCTION, re-entry is impossible, and the user has to back track to the beginning of the function to re-start.

In general, this exit - re-entry has to be done with care, especially in the middle of loops.

Program execution will also be halted during the TOOL CHANGE Instruction. Depending on the type of "END" instructions the controller may automatically restart the program, recycle or simply go home. At any stage the user can push the ON/OFF switch (or emergency stop) to abort the operation. In this event, the user will have to re-start the program from the beginning. Also at any stage the user can start or halt (go to another mode e.g. do a program change) and then restart.

EMERGENCY STOPPING

TOOL BREAKAGE

The tables can generate a thrust of 80 lbs. This is sufficient to easily break 0.25 inch end mills. The best way to handle this is to touch the HALT key then unlock the spindle lock and raise the spindle. Recovery may or may not be possible. Since Z ref is lost, this will have to be re-set with a new end mill.

The user can re-start the program at the beginning and go through SET UP to adjust Z and clear Z, then jump through the program to the last position and continue. If a function is involved, it has to be at the beginning for re-start.

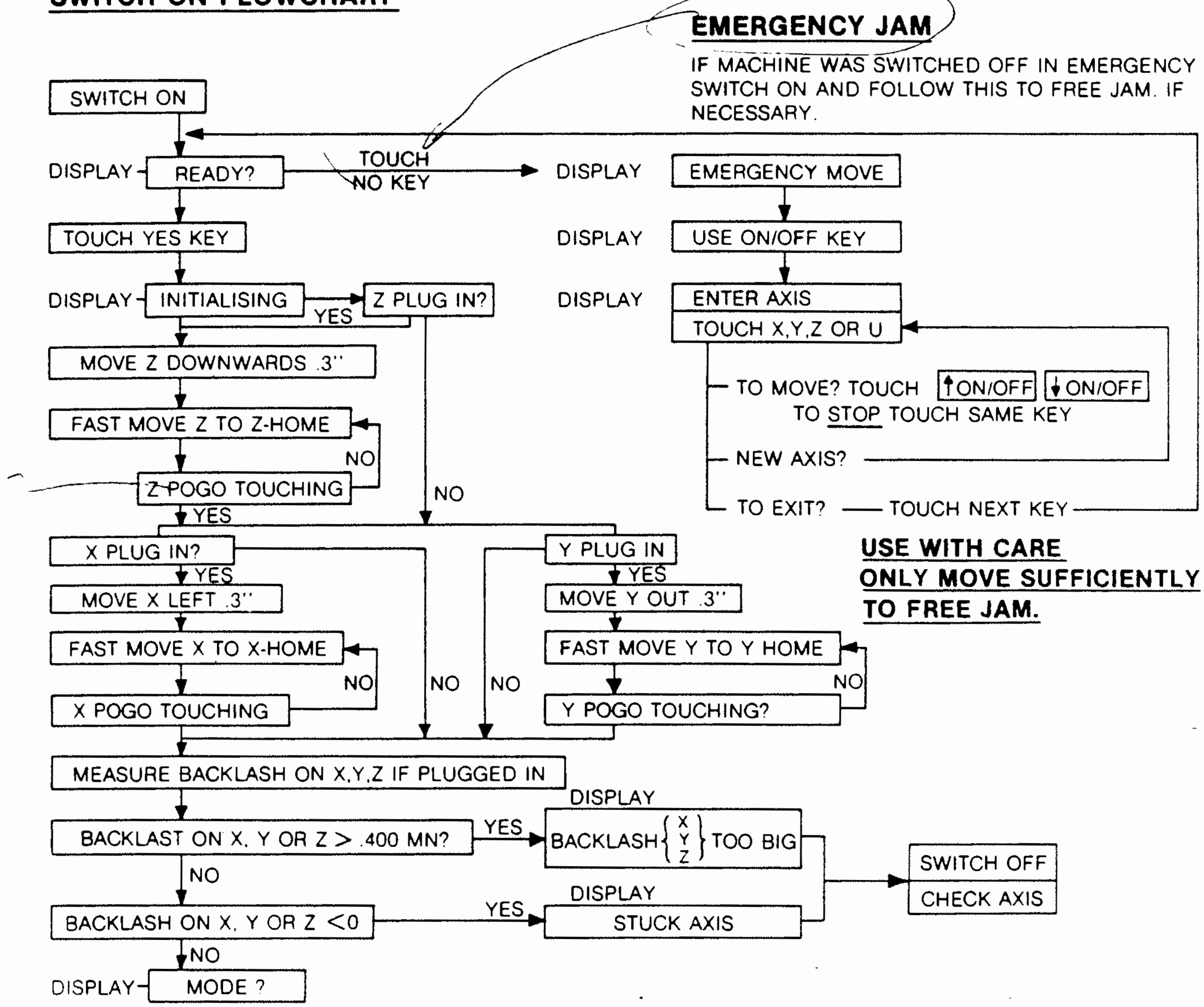
JAMMING

Sometimes a program move is made that hits the clamp, some jig, or the feed rate is too fast. THE USER SHOULD HIT THE EMERGENCY STOP SWITCH, free the spindle lock, clear the obstruction then re-start. It is not possible to recover from such a situation.

If the spindle is jammed for an excessive length of time, the circuit breaker will pop. This has to be reset. It is located under the belt cover next to the motor.

If the user has so jammed the machine up that re-initialisation is impossible (e.g., a saw blade locked in horizontally so that the Z initialisation is inhibited) the user should answer NO to the READY? at switch on. The user can then select the X,Y,or Z axis and use the [↑] or [↓] keys to move the table to free the obstruction. Touching the NEXT key and answering YES to READY? will automatically re-initialise the machine. These moves should be used with extreme caution.

SWITCH ON FLOWCHART



SECTION 6
PROGRAMMING WITH PROMPTS

PROGRAMMING WITH PROMPTS

To simplify the entry of programs and to assure that all the data required to execute a program is entered, the machine controller has been preprogrammed to guide the user in making data and programming entries wherever possible This is done by displaying a "prompt" on the controller display. The user MUST respond to the prompt. If he does not respond, or if he responds incorrectly, the controller will halt and will not go on to the next step or accept any other entry. There are two kinds of prompts. One is a question prompt. The other is a data entry prompt.

An example of a question prompt is:

000 START INCH?

This prompt asks the user whether the dimensions for this program are in inches. The user must respond by touching either a YES or a NO key. If the user response is a YES, the controller will then display:

000 START INS NN

By removing the question mark after "inch" in the display, the controller indicates that it has accepted the instruction and is ready to go on. If the response had been a NO, the ~~prompt~~ will change to:

000 START MM?

Since there are only two choices, inches or millimeters, the response this time must obviously be YES. All question prompts must end up with a YES before the controller will continue to the next step. If the response to the above had been NO the controller will simply cycle back to:

000 START INCH?

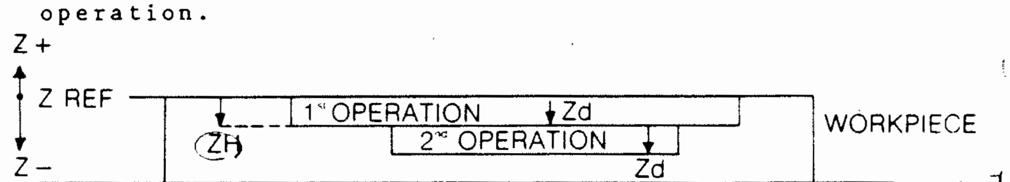
An example of a data entry prompt is:

002 TD=?

This prompt requests the entry of numeric data, in this case, the desired tool diameter for the program. The user must enter the desired value using the numeric keys. Since the machine does not know when the entry of numeric data has been completed, the user must tell it. This is done by pressing the NEXT key after the last number has been entered. If the NEXT key is not pressed to terminate the data entry, the controller will not go on to the next step.

DESCRIPTION & DEFINITION OF PROMPTS

PROMPT	DEFINITION AND REQUIRED USER RESPONSE
nn	Requests an entry of a number between 0 and 99. A single digit entry must be entered as On. Example 5 inches is entered as 05.
F?	Asks if a finish cut is required. Response is YES or NO. If a finish cut is specified, the tool will be offset by .0064 inches from the final dimension in making the first cut in the X,Y, & Z axes. After completion of the operation the tool will return to remove the extra .0064 inches of material from the sides (X & Y) and bottom (Z).
Zd =?	Total depth of cut in the Z axis relative to the reference Z plane. Requires a numeric entry.
ZH =?	This is **a vector value that defines a new surface** height from the original Z reference. It is used when the surfaces to be cut are nested as illustrated below: Observe ZH is negative in this case. ZH is zero when the operation occurs at Z REF. NOTE that the depth of the first operation becomes the ZH for the second operation.



Z% nnn	The distance the Z axis is lowered on each pass as a multipass milling operation. Always expressed as a percentage of the tool diameter. FOR EXAMPLE: If the tool diameter is 0.25" then... Z% 050 will lower the tool 0.125" on each pass. Z% 100 will lower the tool .250" on each pass. Z% 200 will lower the tool 0.5" on each pass. As Z% decreases, the number of passes will increase.
XY CUT %nn	Specifies the amount of cut that is made in the XY plane on each pass in a pocket mill operation. Always expressed as a percentage of the tool diameter. FOR EXAMPLE: XY CUT % 050 means that half the tool diameter will be in the work in the XY plane.
FR AXIS ?	FEEDRATE on the X,Y, and Z axes. The range of permissable values is .10 to 32 inches/min or .25 to 80 CM/min. The user can specify the same feed rate on all axes or a different one for each axis.

Any of the following three entries are acceptable:

FR XYZ = 10.0
FR XY = 10.0, FR Z = 5.0
FR X = 10.0, FR Y = 2.0, FR Z = 1.0

i?	Asks if the cut is to be made on the inside of the pattern outline. The response must be YES or NO. If NO the prompt will change to 0? This asks if it is to be on the outside of the pattern outline. If the response is still NO, the prompt will change to ?. This asks if the cut is to be made on the pattern outline. In this case the response must be YES. If the response is still NO, the prompt will cycle back to i?.
TD =?	Asks the user to enter the value of the diameter of the tool.
TOOL CALIB?	Asks if the tool calibration is to be done. Response must be a YES or NO.
PROBE?	Asks if the thickness value of the electronic probe accessory is to be entered into memory. The response must be YES or NO.
MODE?	Asks in which MODE the user wishes to operate the controller. There are four MODE keys on the controller. The response is to press the desired mode key.
AXIS?	Asks for an entry of either X,Y, or Z axis from the keyboard. Response is to press X,Y, or Z keys. In some cases as in the Feed Rate prompt, all three axes can be entered in response to this prompt.
TOOL 1-8	This occurs during the Tool Calib sequence. This is a simple indicator to the user as to which tool is being calibrated.
X1=? XA=? Y1=?YB= X,Y, Z1=?	Asks the user to enter the numeric value of the or Z coordinate point. The modifier 1,2 simply identifies the specific point. A and B specify the dimensions on X and Y.
XC=? YC=?	Asks the user to enter the X & Y coordinates of the geometry used in circles, arcs & bolt circles.
a1=?	Asks the user to enter the value of the angle through which the tool will start.
a2=?	Asks the user to enter the value of the angle through which the tool will move.

r1=? r2=? r=?	Asks the user to enter the value of the radius.
N=nn	Asks the user to enter the value of the number of holes or spaces required. Used in the bolt circle routine. Limited to two digits.
PECK=nn	Asks the user to enter the number of pecks that will be used in a drilling operation. For accurate drilling the drill bit must be removed from the hole periodically to clear out the chips. The number nn specifies the number of times this removal will be done in drilling a specific hole.
ELSEWHERE?	This prompt appears at the end of a single key-stroke function program. It permits the replication of the geometry defined by the program above it in another location on the workpiece. A YES response to it results in additional prompts for the coordinates of the new location. Use of this prompt is further described and illustrated in the programming section.
REPEAT ? REPEAT X nn REPEAT Y nn	This prompt always follows the ELSEWHERE? prompt above. It permits the replication of the geometry defined by the program above it on a regular X Y grid. A YES response to it results in additional prompts which ask for how many times the geometry is to be replicated in the X Y axes and the spacing Xi or Yi between the geometries in each axis. Use of this prompt is further described and illustrated in the programming section.
SET UP zcxyu	This prompt is used in the SET UP instruction. It permits the set up of the reference coordinates on the workpiece. Details of the responses to this prompt are given in the SET UP section.

SECTION 7
WRITING
A
PROGRAM

"PART PROGRAM"

PROGRAM STRUCTURE

A program to machine a part must consist of three sections which we will simply call, START, MIDDLE and END, because they MUST BE entered in that order.

A typical program will look like this:

	LINE NO	PROGRAM STATEMENT	
SET-UP → START SECTION	000	START INS 01	This section contains all of the required machine parameters such as, program no. dimension systems, tool diameter, feed rates, and set up instructions.
	001	TD =.125	
	002	FR XYZ =10.0	
	003	SET UP→zcxyu	
PGM (MOVES) → MIDDLE SECTION	004	RECT F Z% 050	This section contains instructions for machine moves, coordinate point data and dimension data of the geotry to be machined.
	005	XY % 050	
	006	ZH = 0.0000	
	007	Zd = 0.1000	
	008	X1 = 1.0000	
	009	Y1 = 1.0000	
	010	XA = 0.5000	
	011	YB = 0.5000	
END → END SECTION	012	END	This section contains either the END instruction or variations of the END instructions to allow the user to replicate or repeat the program.

Rev 7-12

ENTERING A PROGRAM

We have to locate the program somewhere in the 000-900 line space. The user may wish to clear the entire memory or locate an area in memory which is free (LINE MODE, line number). For simplicity, we shall assume a cleared memory and we are starting at line number 000. To enter a program simply press the "PROGRAM ENTER" key. We go down the left most column of keys.

Pressing this key	Results in this display
1. START	`000 START INCH?`

The display is asking the user to specify whether the dimensions of the geometry that will be machined by the program are in inches. The user must respond by pressing either the YES or NO key. When either one is pressed the display will change to either one of the two examples shown below, depending on whether the answer was YES or NO.

Pressing this key	Results in this display
YES →	`000 START INS nn?`
NO	`000 START mm?`

which must be confirmed by pressing the YES key and the display will show

`000 START mm nn`

The controller has now been programmed to recognize all subsequent dimension data entries in this program as being either inches (case 1) or millimeters (case 2). Other programs which are stored in the controller memory can be in either inches or millimeters. To insure that the user remembers which system he is in, all subsequent prompts for data entry in a program that was set-up for inches will have four digits after the decimal point. A program which was set-up for millimeters will have data entry prompts with three digits after the decimal point.

The display has also shown a new prompt nn. This is a request to enter the Program Identification Number. This number can be any number from 00 to 99. Single digit numbers must be entered as "0n". Example, 5 is entered as 05. It is simply a user convenience.

To enter the number, the user must touch the corresponding numeric keys and the NEXT key.

(4) 2. The ~~next~~ following key (which may be required) is...

Resulting in this display..

the TOOL DIAMETER key | 001 TD = ? | → .250 & next

The tool user must enter the numeric value of the diameter of the tool using the numeric keys and touch NEXT . The tool diameter information is used by the controller to compensate for various kinds of cuts. If a tool is changed further in the program, by the use of the TOOL CHANGE statement a new tool diameter must be entered into the program following the TOOL CHANGE line number. If no number is entered for the tool diameter, the controller will assume it is the same as before.

Clearly, if we are only going to drill, then we can skip this key.

(6) 3. The ~~next~~ following key in this start section is the feedrate. Pressing

Results in this display..

FEED RATE | 002 FR AXIS = | → calculate from formula or ...

1) The axis can be any combination of X,Y or Z. Thus, X,Y,Z, XY, YZ, ZY, XYZ are valid selections.

2) This must be followed by a numeric entry

	in inches	0.05	through	~~16~~/30 inches per minute.
or	in cm	0.13	through	41/76 cm per minute.

Note that the leading zero must be entered in fractional speeds.

If no feed rate is specified, then the default speed is mid range. (8 ins/min)

The feed rate can be changed anytime, one axis or all at any stage in the program. Thus our feed rate may be specified as for example.

```
002  FR  XYZ=10.0     or     002  FR  XY=10.0
                             003  FR   Z=2.0
```

In general. tougher materials require slower ~~speeds.~~ Plastics (10 - 16), Aluminium (5 - 10), Steel (0.1 - 5) inches per minute.

TYP. FEED RATES

The next and last line in the START section of the program is:

SET UP | 003   SET UP→zcxyu |

This prompt is simply entered into the program by pressing NEXT. When a workpiece to be machined is placed on the table, and the controller is placed into the PROGRAM RUN MODE the program will stop at this prompt and will wait for the entry of the required information.

This prompt asks for the entry of the reference point for the Z,X,Y, and U coordinates. (The U coordinate is used only when an optional rotary table is in use). This is a point in space, or more usually on the workpiece, to which all workpiece cutting geometries and their dimensions are referenced in this program. The information for this point is entered into the program ONLY by physically positioning the center and tip of the tool at this point. This information CANNOT be entered from the keyboard as the machine does not know where the center and tip of the tool is until it is positioned there. During the time this prompt is displayed, the the following keys are activated:

Key	Function
↑ ON/OFF, ↓ ON/OFF	Moving axis at a constant rate. Must be retouched to stop.
△↑ 7 / inside, △↓ 8 / outside	Jog increments .001 inches on each touch.
NO / 2	Rotate to select axis to be set.
SET UP REF / 0 / PROG. REF	Set ref. cood here.
CLEAR	Clear this ref.

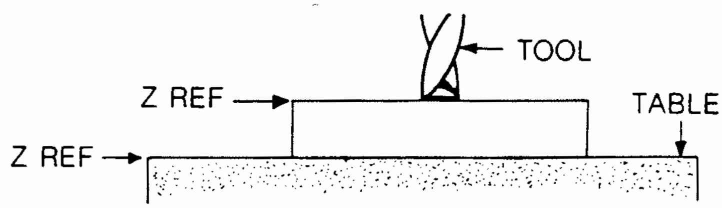
There are two ways to set the SET UP point in space

1) MANUALLY or

2) WITH THE PROBE

1) **MANUAL SET UP**

The entry of data required by the SET-UP→zcxyu Prompt is done in the following way. The arrow in this prompt is always pointing to the axis to be set. In this case, the Z axis which is usually the first axis to be set. The Z axis (spindle head) is now positioned by the use of the [↑ON/OFF] [△ ↑] or [↓ON/OFF] [△ ↓] jog keys or by moving the quill with the quill adjusting lever, until the tip of the tool JUST touches the Z reference plane. This position of the Z slide is now entered into the program as the Z REFERENCE by pressing SET UP REF KEY. The lower case z will change to capital Z, indicating that it has been set. One can clear the entry by pressing CLEAR in which case Z will revert to z.



If the quill is moved manually, it must be locked in this position for the duration of the program.

At this stage having set the Z reference, the display will show:

003 SET UP → Zcxyu

We wish now to set clear Z (abbreviated to C). Touch the NO key.

The display will then rotate to:

003 SET UP→cxyuZ

and we are ready to set the clear Z reference. This is the position that the tool tip will be elevated to when the tool moves from one operation to another. It must be set to clear the highest surface on the workpiece but not too high as to minimize useless move time. The spindle head is now moved by pressing

[↑ON/OFF]

key until the tool tip is visually observed to be in the clear position. Press [↑ON/OFF] key again to stop. This position is entered by pressing the SET UP REF key. The display will show:

003 SET UP→CxyuZ

If this operation is not performed, the default CLEAR Z position is the HOME or (Z MAX) position.

Pressing the NO key will again rotate the display to:

SET UP→xyuZC

The x, y, and u axes are now set up following the same procedure.

NOTE:

1) The user may find that he needs X to move, then Y, then X to position the tool point at the reference point. Simply select, by the NO key, which axis is required. It will only be set when the user presses the SET UP REF key, and the axis goes from lower case to upper case.

Moving an axis will not affect the SET-UP point on that axis if set.

2) After the reference point is set in all the axis and clear Z is set, the user presses the NEXT key to continue running the program.

3) If the set up instruction is not included in the program, the REF point defaults to the home position. This may cause error coods because of tool offset when running a program.

4) If the SET-up instruction is included in the program, but the user simply presses the NEXT key (essentially by-passing the SET-UP procedure) the REF point will default to the home position again and will have to the same error code as (3).

The user may execute some go statements from the HOME position before SET-up and jog the part to the tool tip. Then the only SET-UP the user requires is Z and clear Z.

5) Once the set up instruction is concluded and the operator has continued on, the letters will revert back to the lower case WITHOUT loss of their set up ref points.

2. PROBE SET UP

For the Z reference, place the probe on the surface to be referenced and touch the [↓ON/OFF] key. The tool tip will descend and stop when the button on the probe is touched. The display will then show:

SET UP ⟶ Zcxyu

At this time you may remove the probe and press NO to rotate the display to:

SET UP ⟶ cxyuZ

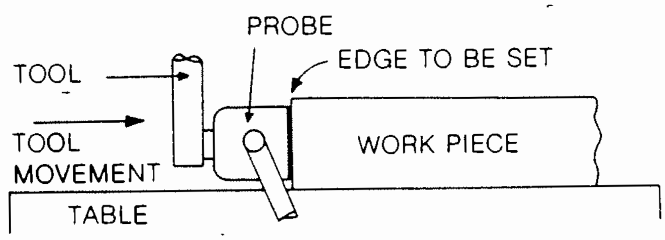
Press [↑ON/OFF] to raise or adjust the tool tip to set clear Z. Remember to press again to stop.
Press SET UP REF to set clear Z so the display will show:

SET UP ⟶ CxyuZ

Press NO to rotate display to:

SET UP ⟶ xyuZC

To set X the user may have to lower Z further to arrange the following configuration.



The X or Y ref will be taken as TOOL RADIUS + PROBE OFFSET and set at that point.

NOTE:

1) The SET UP reference points are not stored in the controller when the power is turned off. They are automatically reset to zcxyu (i.e. all references are cleared on power off).

2) There can be more than one SET UP per program.

3) At the SET UP stage one can display the current location of the axes by this sequence of keys.

SHIFT	DISPLAY	X
		Y
		Z

4) If the axis is not set and the user touches the next key, XYZ will default to the home position in space and clear Z set at Z max.

5) Tool diameter must be entered before one can do the set up operation.

All the machine parameters have now been entered and this terminates the START SECTION.

THE MIDDLE SECTION

This section contains the program for machine moves, coordinate changes and dimensional data of the geometry to be machined. The program can be entered by using the single keystroke preprogrammed functions (SKIP SYSTEM), or high level DYNALAN language statements or a combination of both.

In this first example, we will use the SKIP SYSTEM. Programming in DYNALAN will be described in a later section.

PROGRAMMING WITH FUNCTION KEYS (SKIP SYSTEM)

In this method, the most often occuring machine geometries have been preprogrammed into the machine controller ROM (Read Only Memory) as subroutines. Seven of these are identified below.

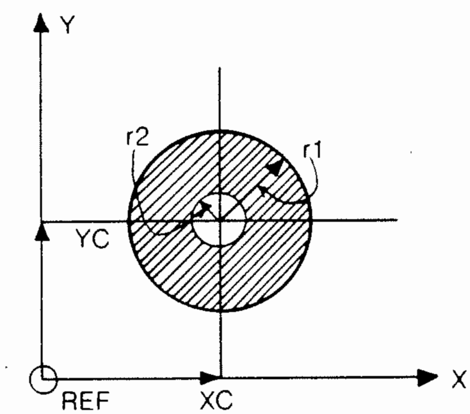
MILL	RECT POCKET	RECT FRAME	CIRCLE POCKET	ARC FRAME	BOLT CIRCLE	DRILL

These geometries are printed in blue on the bottom half of some of the keys. They are accessed by pressing the SHIFT key on the lower left hand side of the keyboard. When the SHIFT key is pressed, the red light next to it will light. After an instruction requiring a shift is executed, the shift mode will automatically reset and the light will go out. The SHIFT key must be pressed EACH TIME a blue instruction is to be entered.

When one of these function keys is pressed, the controller display will respond with a series of prompts which ask the user to enter the information that is required to execute that particular geometry.

We will illustrate this method by using the "CIRCLE POCKET" function as an example.

The geometry of this function is shown below:



This function generates a circular pocket of radius r1 with a center post of radius r2. The material in the shaded portion will be removed. The center of the geometry is at XC and YC relative to the REFERENCE COORDINATE POINT. If r2 is zero the post disappears. The total depth of the pocket is Zd. Maximum R is 2.75 inches. r2 should not exceed r1-Tool diameter. When r2 = r1-Tool Diameter, we get a Circular Frame Cut.

When the "CIRCLE POCKET" key is pressed the display will respond with the following prompts which will appear in the sequence shown on next page.

ENTRY FORMAT	DESCRIPTION OF STATEMENT	USER RESPONSE
CIRC F?i? Z%NNN	F is Finish Cut,	YES or NO
	i is Inside/Outside Cut	YES or NO
	Z% nnn is Z increment per pass. – expressed as % of tool dia	numeric keys+NEXT
XY CUT % nnn	XY increment per pass	numeric keys+NEXT
ZH=	Z ref. offset	numeric keys+NEXT
Zd=	Z depth of pocket	numeric keys+NEXT
XC= YC=	COODS of CENTER	numeric keys+NEXT
r1=	Outer radius	numeric keys+NEXT
r2=	Inner radius	numeric keys+NEXT
ELSEWHERE? (Do you want this geometry repeated "elsewhere"?)		YES or NO
~~if YES then:~~		
XC= YC=	COODS of CENTER in another location	numeric keys+NEXT
r1=	Outer Radius	numeric keys+NEXT
r2=	Inner Radius	numeric keys+NEXT
if NO then:		
REPEAT?		YES or NO
if YES then:		
REPEAT X nn (How many rows)	How many times in the X axis	numeric keys+NEXT
Xi =	Interval in the X axis	numeric keys+NEXT
REPEAT Y nn (How many columns)	How many times in the Y axis	numeric keys+NEXT
Yi =	Interval in the Y axis	numeric keys+NEXT
if NO then:		
Line No NNN	End of this Circle Pocket routine.	

The above program constitutes the MIDDLE SECTION of the program. Programming with FUNCTIONS requires only simple responses to question or data prompts. The middle section of the program can contain several of these functions and/or DYNALAN statements.

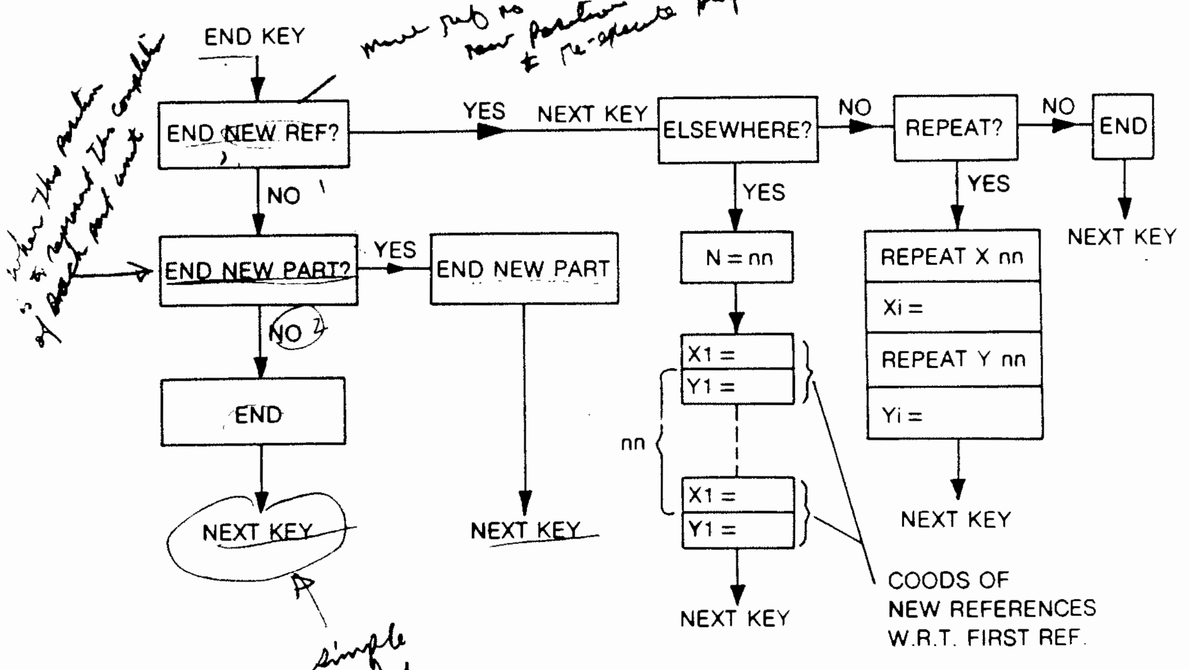
Additional functions are described in the AVAILABLE FUNCTIONS section. PG 8-1

For this illustration we will stop with just this one geometry and will now enter the END SECTION of the program.

3. The END Section

All programs must be terminated with an END statement, otherwise the machine does not know where one program stops and another one begins. This is done by pressing the END key. Since there are several possible endings to a machining program, the END key has been programmed to give the user several choices by using appropriate prompts. A Flow Chart of the END key prompts is shown below.

FLOWCHART OF END STATEMENT



1) END

The simplest ending is just END (press NO key TWICE). This is used when we wish to machine one part or to debug a more complex ending.

The program can be run again on a fresh part but the user has to go through the SET-UP procedure again. This can be very inconvenient. Hence the user should select the END NEWPART ending (NO,YES) for repetative machining.

2) END NEWPART

The parts to be machined should be located in a jig. On the first run through, the machine will halt at SET-UP, allowing the user to set the reference point. This reference point is stored. After running the program, the controller will encounter the END NEWPART statement. The following happens:

1) The spindle head is retracted (Z to Zmax) to provide clearance for removal and insertion of fresh part.

2) An XY move takes place to the home position to check location. This corrects for accumulated error that may occur in very long sequential incremental arc moves.

3) An XY move back to the REF point as previously set.

4) At this stage, the user inserts the fresh part into the jig.

5) The program is automatically switched back to the SET-Up instruction.

6) At this stage the user can press the NEXT key and the program will cycle through again, or the user can choose to re SET-UP again on any or all axis to absolutely confirm the REF point. The new SET-UP replaces the old set up temporarily.

NOTE:

1) Any REF point drift is due to contamination of the limit switches. Debris should be swept off carefully, not blown off. Also refer to statement (2).

2) Keep the slides well oiled.

3) END NEW REF

The program can be replicated on the same workpiece by two ways assuming that Z and clear Z remain constant.

1) ELSEWHERE Here the user enters nn the number of times followed by the nn coordinates of each reference point on the XY plane.

2) REPEAT The X number and X spacing followed by the Y number and Y spacing will replicate the program periodically on the workpiece on the XY plane.

Upon termination of these programs, the controller effectively sees an END NEWPART instruction and will automatically go into this mode.

The set up point will still be the original set up point. So the user may cycle through additional fresh parts to be machined.

SECTION 8
AVAILABLE FUNCTIONS

command cycle

AVAILABLE FUNCTIONS

This section describes each of the Functions which have been preprogrammed into the basic machine controller. The description for each includes the entry format, the flow chart of the prompts, the geometry of the functions and the conventions employed.

1) One critical point that the user should be aware of is the concept of REPEAT and ELSEWHERE as applied to the functions.

Once the data has been entered for the function, the question is asked REPEAT?. This asks if the function is to be repeated AS IS in a periodic fashion in the XY plane or a portion there of. a grid of repetitions

a single repeated @ a space loc

If not, then the function can be located ELSEWHERE at random. This may be followed by a REPEAT X & Y at this new location.

So, the ELSEWHERE is to MOVE the function to a new location.

The REPEAT function must be repeated at each ELSEWHERE location. The ELSEWHERE is NOT GLOBAL in operation, (i. e. ELSEWHERE only re-locates the function).

IMP

*2) These functions operate in the first quadrant, so the REF point should be made in the lower left hand side of the part. pattern — 1st pt where geometry is given

*3) Negative values for repeat spacings are not allowed.

*4) On exit from a function, all local zeros are cleared if set previously.

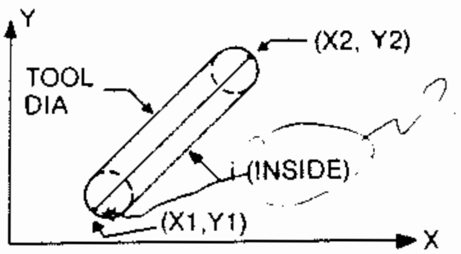
On entrance, all local zeros are cleared so the function must be defined with respect to the REF point.

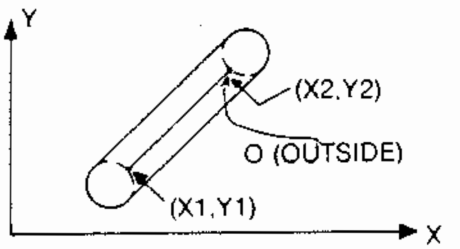
ALL LOCAL ZERO'S ARE WIPED OUT WHEN A FUNCTION IS EXITED or entered & MUST BE RECREATED (IF NEEDED) AGAIN

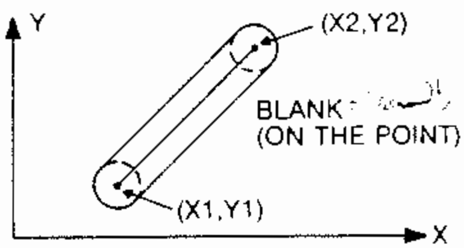
NOTE It might have to be ever completely of part in order for the tool center when traveling along the perimeter of part never to cross outside the 1st quadrant of C0.00

MILL

The entry format is:



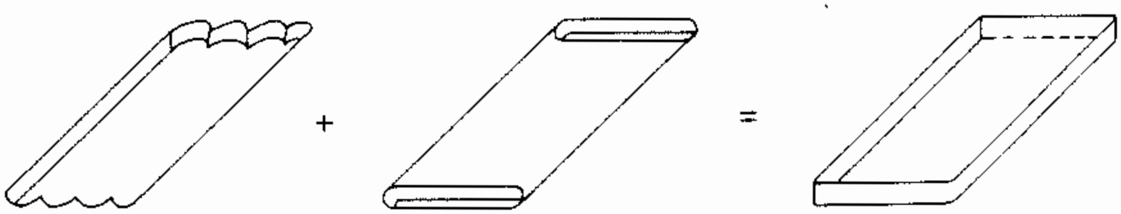




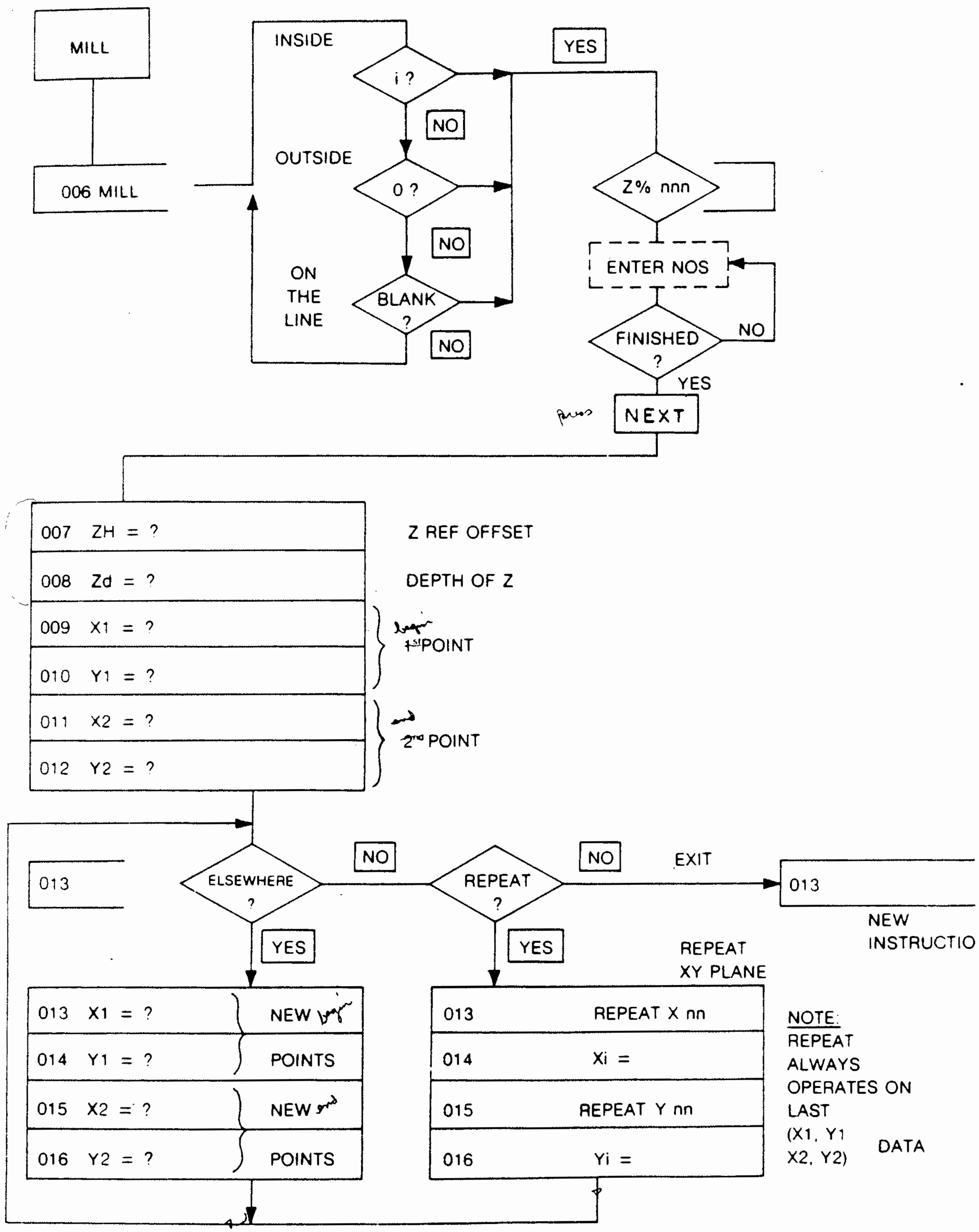

```
MILL        i          Z% nnn
            o
            blank

            ZH =       Surface displacement
                       from Z REF.

            Zd =       Total depth

            X1 =       START COODS
            Y1 =

            X2 =       END COODS
            Y2 =

ELSEWHERE?             (NEW X1Y1, X2Y2 ?)

REPEAT?
```

This function mills a slot from point (X1,Y1) to point (X2,Y2) on the XY plane. It will cycle back and forth with each Z increment a percentage (Z% nnn) of the tool diameter until Zd is reached. The width is the tool diameter.

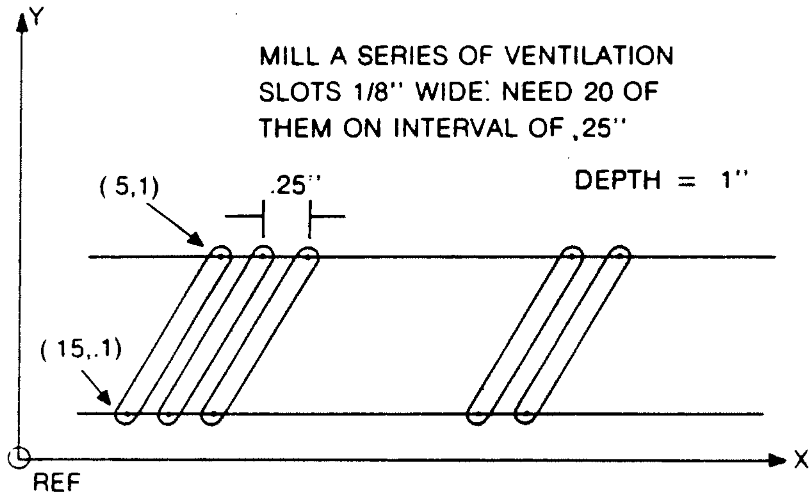
NOTE: With the REPEATED capability,angled pocketing is very easy easy. For example, the six milling operations required to achieve the angled pocket as shown below are done by REPEATING the MILL INSTRUCTION and then removing the top and bottom radii via an ELSEWHERE instruction.



FLOWCHART FOR MILL



EXAMPLE OF MILL

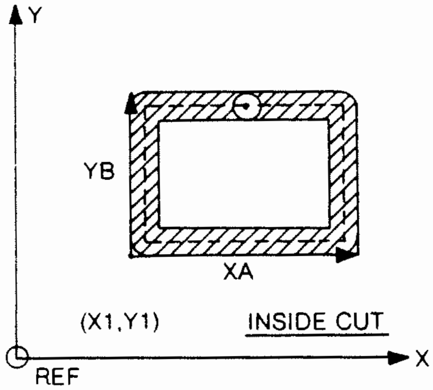


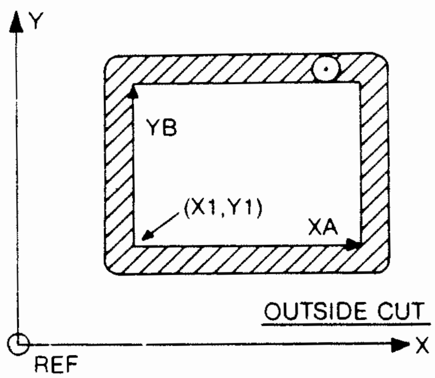
The program using MILL would appear as:

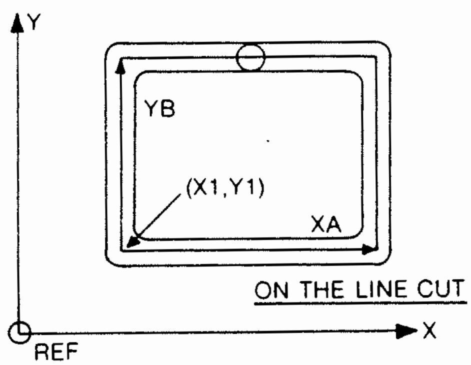
```
000        START INS 01
001        TD = .125
002        FR XYZ = 10.0
003        SET UP——► zcxyu
004        MILL   Z% 050
005        ZH = 0            (HT above surf)
006        Zd = .1           (total depth to mill to)
007        X1 = .15
008        Y1 = .1
009        X2 = .5
010        Y2 = 1.0
011        REPEAT  X   20    — # columns of repeated pattern
012        Xi = .25
013        REPEAT  Y   01    — # rows of repeated pattern
014        Yi = 0
015        END
```

FRAME

The entry format is: FRAM (F) i Z% nnn
o
blank







ZH = Surface displacement from Z ref

Zd = Total depth

X1 = Coordinates of lower
Y1 = LHS of rectangle

XA = Length of rectangle in X direction

YB = Length of rectangle in Y direction

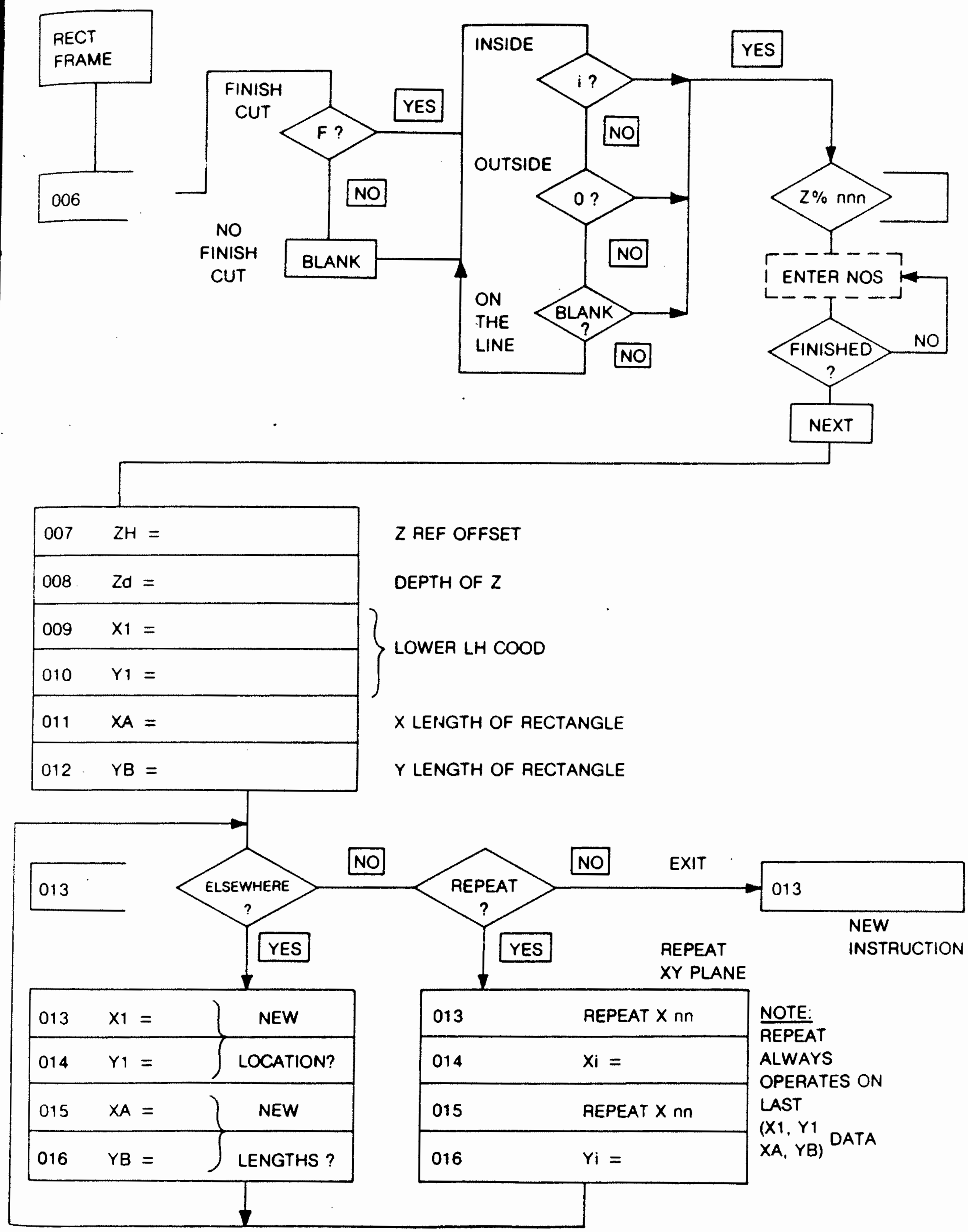
ELSEWHERE? (New X,Y, XA,YB)

REPEAT?

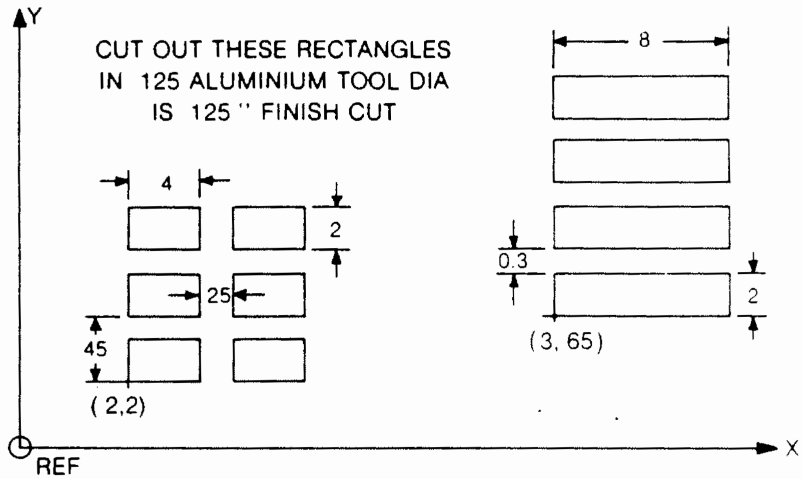
This will cut a rectangular frame in the XY plane. If the cutting is to be on the rectangle there is no finish cut. The tool offset is increased by .0064" if the finish option is exercised and a second pass is made to remove the .0064" For either inside or outside a climb cut is made (inside: counterclockwise, outside: clockwise).

NOTE: For smaller diameter mills, care must be exercised in specifying the Z feedrate. Too high a rate will produce skitter (i.e. the tool will deflect during entry). Similarly if the mill is extra long, plunge cutting at too high an X,Y feedrate will result in deflection of the tool, or tool breakage.

FLOWCHART FOR RECTANGULAR FRAME



EXAMPLE OF RECTANGULAR FRAME



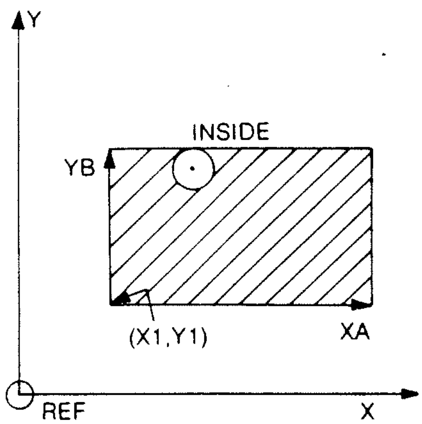
The program using the RECT FRAME would be:

```
000   START INS 02
001   TD = .125
002   FR XYZ = 8.0
003   SET UP→zcxyu
004   FRAM F i Z% 050
005   ZH = 0
006   Zd = .128
007   X1 = .2
008   Y1 = .2
009   XA = .4
010   YB = .2
011   REPEAT X 02
012   Xi = .65
013   REPEAT Y 03
014   Yi = .45
      (ELSEWHERE) YES
015   X1 = 3
016   Y1 = .65
017   XA = .8
018   YB= .2
019   REPEAT X 01
020   Xi = 0.0
021   REPEAT Y 04
022   Yi = 0.5
023   END
```

RECT POCKET

The entry format is: RECT (F) i

Z% nnn	(note: inside only)
XY cut%nnn	XY increment
ZH =	Surface displacement from Z-REF.
Zd =	Total depth of pocket
X1 =	Coods of lower left
Y1 =	hand corner of pocket
XA =	Length of pocket X axis
YB =	Length of pocket Y axis
ELSEWHERE?	(new X,Y,and XA, XB ?)
REPEAT?	

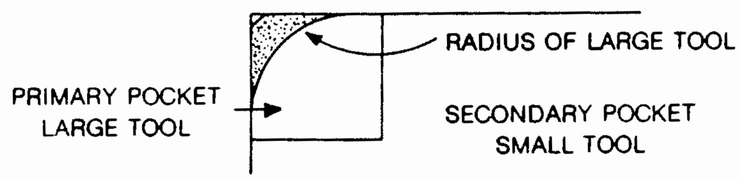


This function generates a rectangular pocket in the (XY) plane. If a finish cut is done, the tool offset is increased by .0064" and the Zd is decreased by .0064" as well. The finish cut will clear out this additional .0064" from the sides and the bottom. The sides will be a climb cut.

Do not make Zd greater than the cutter length or the tool may be damaged.

Each pocket pass will be at Z% of the tool diameter, so if the tool diameter is 0.25" and Z% is 50% then each pocket pass will be 0.125" deep. The last pass will remove the remainder.

NOTE 1: Removal of large corner radii in Pocket cuts is frequently a problem because one wishes to excavate at maximum speed with a large diameter tool while still maintaining small corner radii. The solution is very simple. Use a large diameter tool for the major excavation and the desired radii tool for 4 small pocket excavations at the corners. A TOOL CHANGE will be requried after the major excavation.



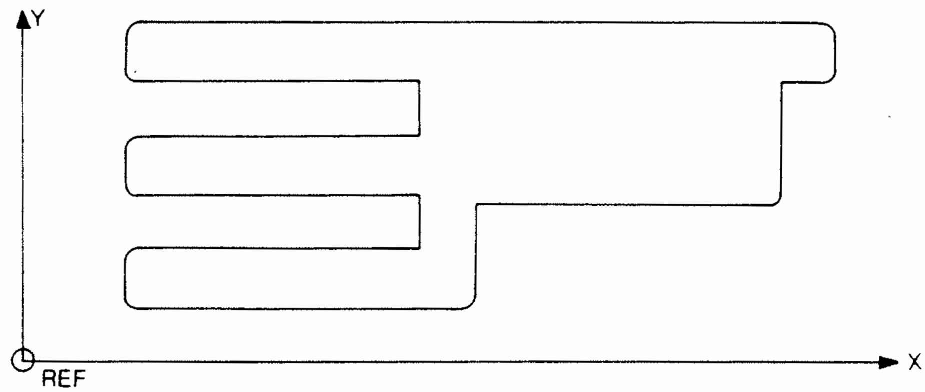
The secondary pocket is a simple repeat (X=2, Y=2) for each corner. The user must evaluate the time trade off between using a small tool all the way or the sequence of a large tool, a tool change, and a small tool.

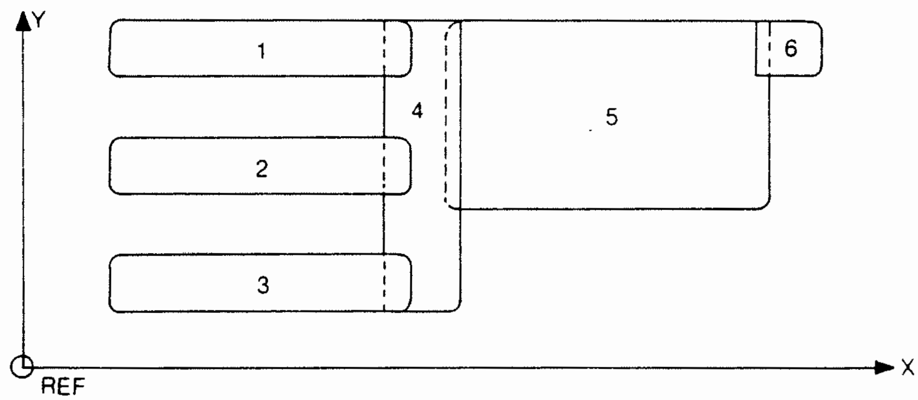
NOTE 2: Surface Milling. For surface preparation it is frequently necessary to face mill the entire surface. Using 2.5" dia. Face mill cutter, simply treat the entire surface as a pocket cut with a very shallow depth. The depth should not exceed .010" and observe that the minimum Z increment (.01% of 2.5") is .025" so the pocket pass would be a remainder cut.

NOTE 3: With small diameter mills it is essential that the feed rate be reduced. Excesive feed rates produce tool bending resulting in inaccurate pocket dimensions.

NOTE 4: The ELSEWHERE? and REPEAT? statements are very powerful. Pocketing is additive, so for irregular shaped geometries formed from lines parallel to the X and Y axis partitioning into subrectangles is always possible.

Consider the following pocket.

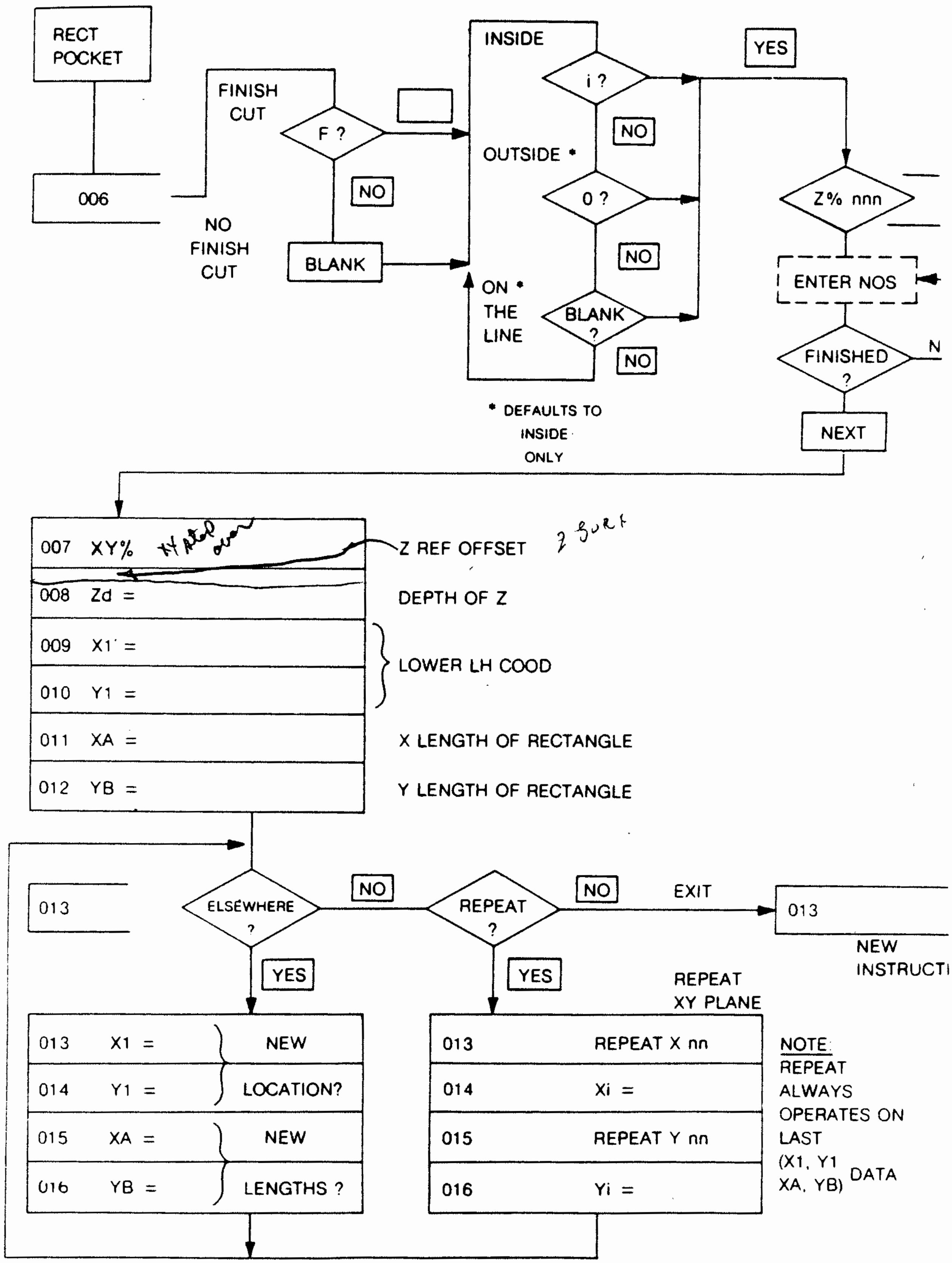




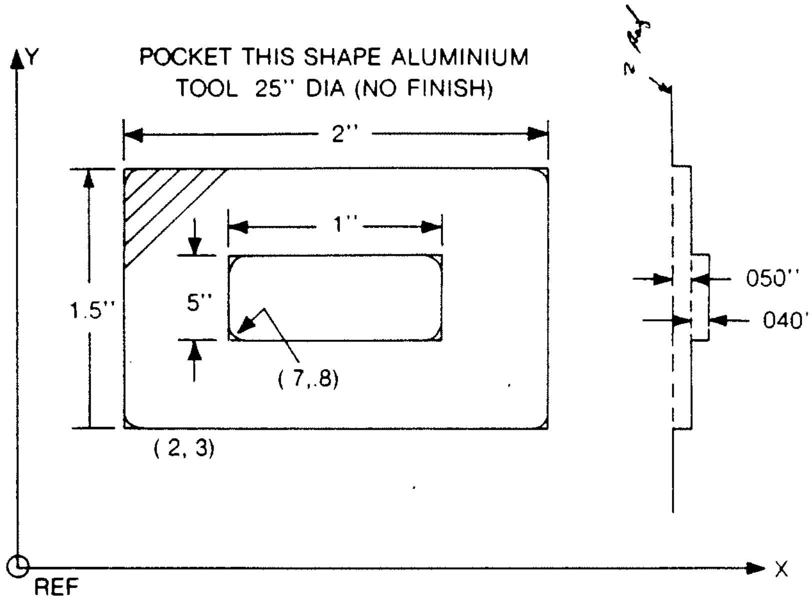
The trick is to partition the areas into slightly overlapping rectangles to eliminate the radii at the corners. The circles denote the coods which have to be entered sequentially in the ELSEWHERE request.

Alternatively 3 can be entered first, 2 and 1 are REPEATS along the Y axis, (X = 1, Y = 3) and 4,5 & 6 can be treated as ELSEWHERES. This will clearly hold for island posts within the rectangles as well.

FLOWCHART FOR RECTANGULAR POCKET



RECTANGULAR POCKET EXAMPLE

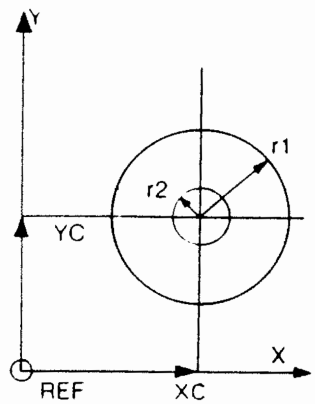


This is a pocket in a pocket. The program would be:

```
000  START INS 01
001  TD = .250
002  FR  XY = 8
003  FR  Z = 2
004  SET UP→zcxyu
005  RECT  i  Z% 050
006  XY CUT% 050
007  ZH = 0
008  Zd = .050
009  X1 = .2
010  Y1 = .3
011  XA = 2.0
012  YB = 1.5
013  RECT  i Z% 050
014  XY CUT% 050
015  ZH = -.050
016  Zd = .040
017  X1 = .7
018  Y1 = .8
019  XA = 1.0
020  YB = .5
021  END
```

CIRCLE POCKET

The entry format is:

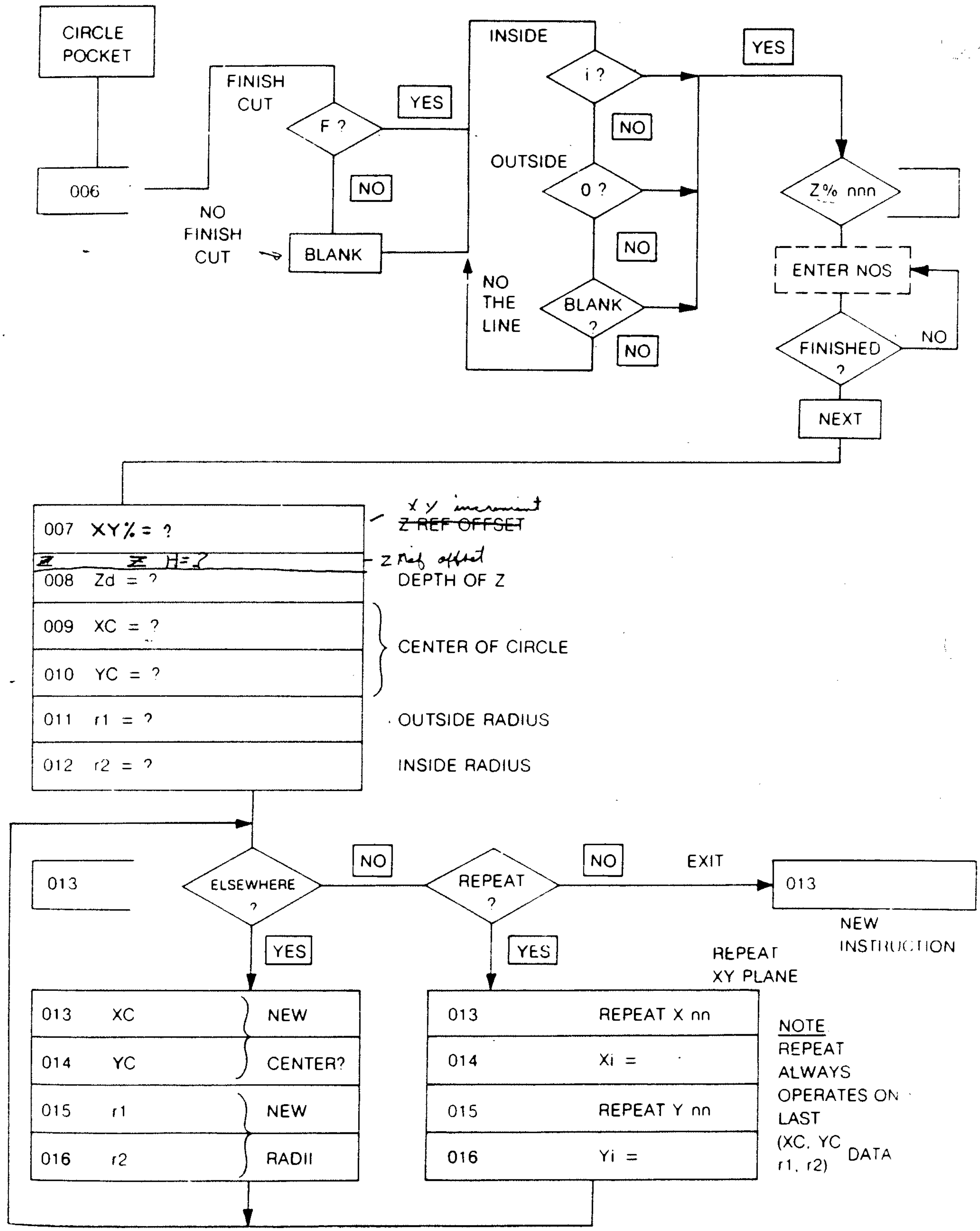


CIRC F? Z%nnn	F is Finish Cut Option Z% is Z increment
X Y CUT% nnn	XY increment
ZH = ?	Offset from Z ref.
Zd = ?	Depth of cut from surface
XC = ?	COODS OF CENTER
YC =?	
r1 = ?	Outer radius
r2 = ?	Inner radius
ELSEWHERE?	(New R1 and R2)
REPEAT?	

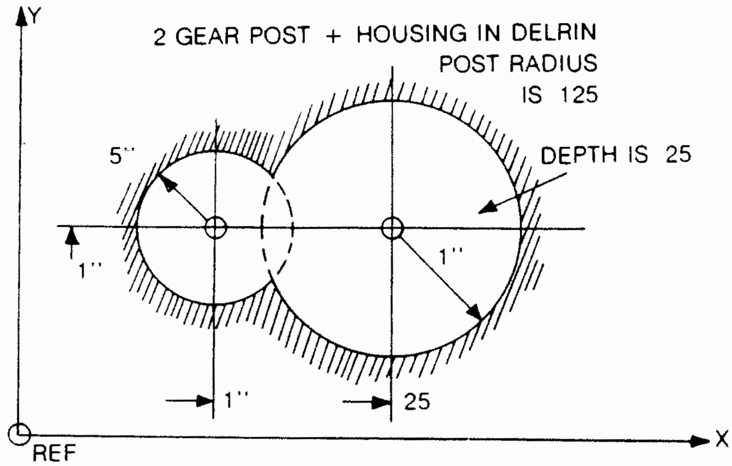
This function generates a circular pocket of radius r1 with a center post of radius r2 on the inside. The tool is automatically compensated for, on the inside of r1 and the outside of r2. If r2 is zero then the post disappears. Zd is the depth of the pocket. The maximum size of r1 is (Y travel) 5/2=2.50". The maximum circle diameter is 5". r2 should not exceed r1-Tool Diameter.

NOTE: For a frame cut, make r2 = r1-Tool Diameter with no finish cut.

FLOWCHART FOR CIRCULAR POCKET



CIRCLE POCKET EXAMPLE

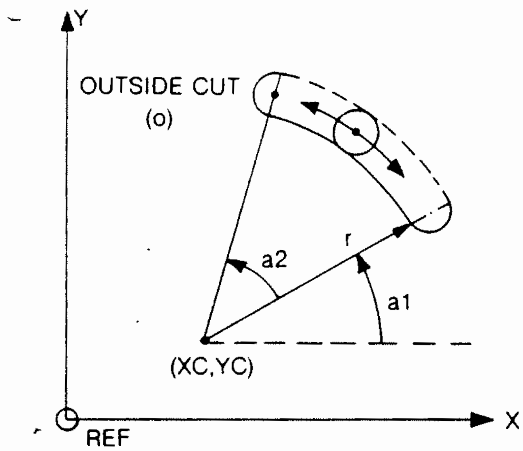


The program would be:

```
000   START INS 01
001   TD = .125
002   FR XYZ = 10
003   SET UP→zcxyu
004   CIRC F Z% 050
005   XY CUT % 050
006   ZH = 0
007   Zd = .25
008   XC = 1
009   YC = 1
010   r1 = .5
011   r2 = .125
      (ELSEWHERE)
012   XC = 2.25
013   YC = 1
014   r1 = 1
015   r2 = .125
016   END
```

ARC FRAME

The entry format is: ARC F i Z% nnn
o
blank



ZH = Surface offset from Z ref

Zd = Depth of cut from surface

XC =
YC = Center of arc

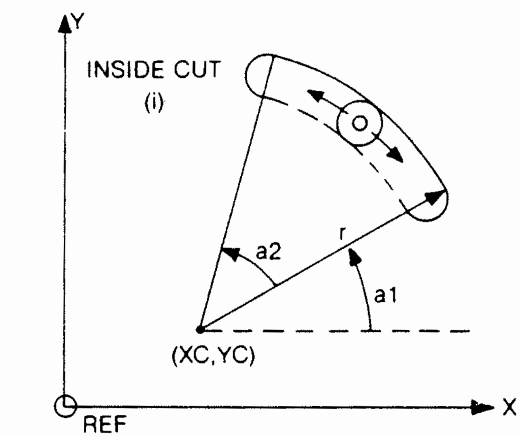
a1 = Starting angle

a2 Traveling angle

r = Arc radius

There is no elsewhere or repeat.

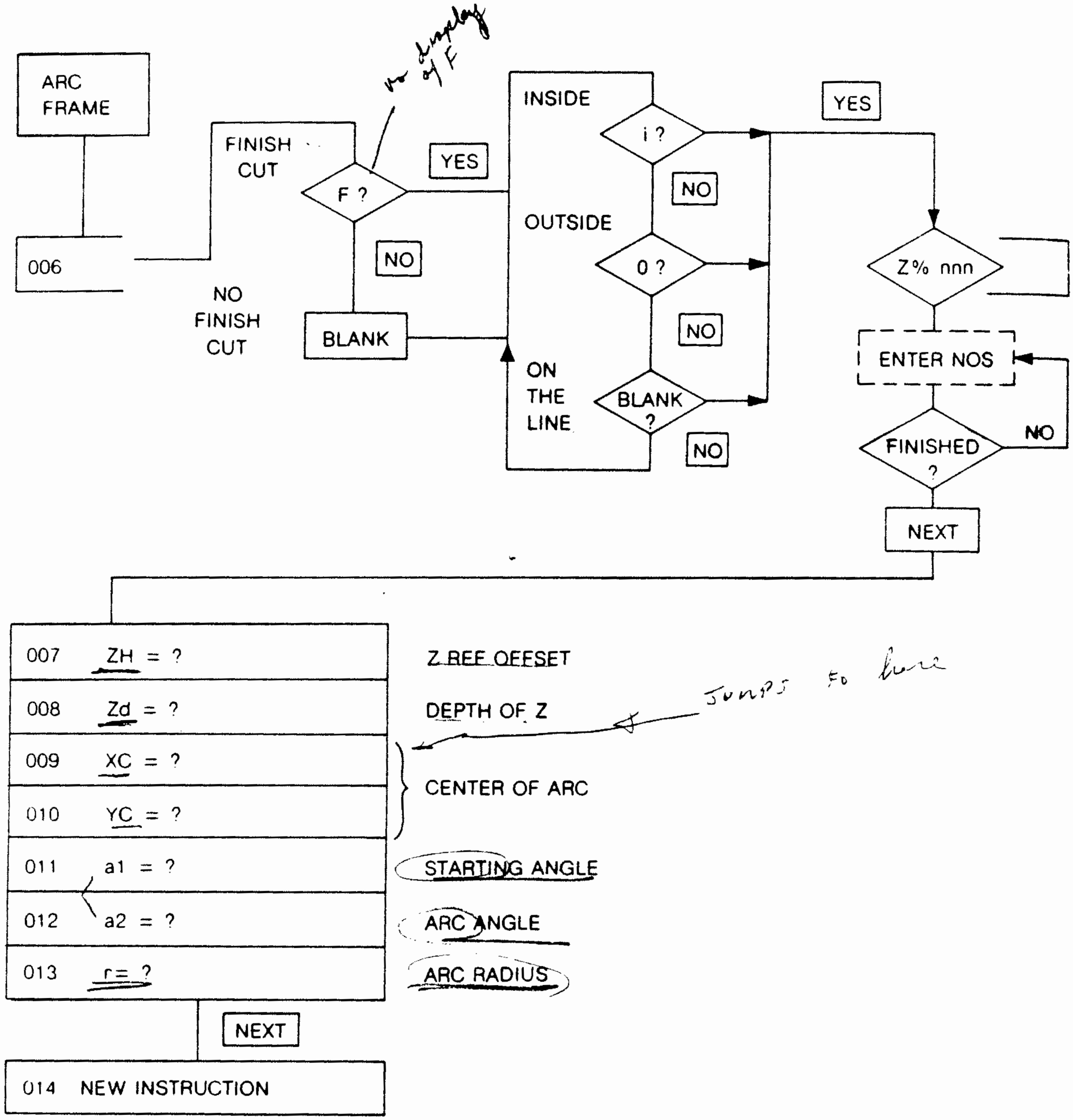
This function mills a direct arc from angle a1, through a displacement angle a2 in the XY plane. The center of the arc is at XC,YC and the radius is r. If it is multipass, the tool will go back to a1 along CLEAR Z to repeat a2 is minus (-) in a clockwise direction, and plus (+) in a counterclockwise direction (conventional axis notation) a1 is always referenced from the X axis.



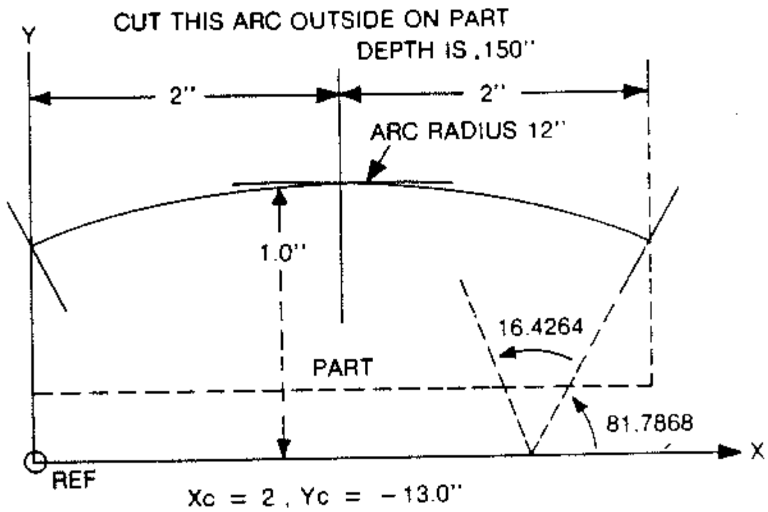
what happens if the tool runs out of 1st quadrant

NOTE: The center may be located off the table. The maximum value of r is 36 inches (91.44 cms).

FLOWCHART FOR ARC FRAME



ARC EXAMPLE

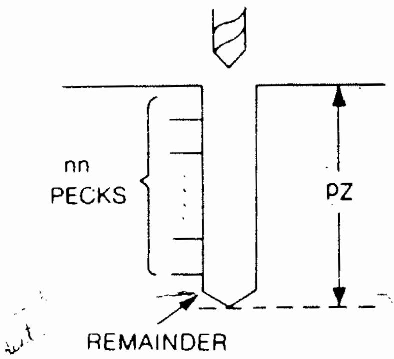


Program becomes:

```
000   START INS 01
001   TD = .125
002   FR XYZ = 10
003   SET UP→zcxyu
004   ARC o Z% 050
005   ZH = 0
006   Zd = .150
007   XC = 2.0
008   YC = -13.0
009   A1 = 81.787
010   A2 = 16.426
011    r = 14
012   END
```

DRILL

The entry format is: DRIL PECK = nn



ZH =	Surface height from Z ref
Zd	Depth to be drilled from surface
X^c =	
	COODS of the point
Y^c =	
ELSEWHERE?	
REPEAT?	

Peck = nn are the number of pecks required, nn=00 - 99. The total number of pecks = nn + 1, where the extra 1 is the remainder. These pecks are bigpecks. (i.e. the tool will return at maximum speed to the surface to completely clear the drill. On the way back in, it will go at maximum speed within .005" of the last depth, then continue at the specified feed rate for the peck amount.

NOTE:

1. If tapping, there must be no pecking, so nn = 0.

2. The COODS for subsequent holes (as in ELSEWHERE & REPEAT) Must be consistent with the first. (i.e. all referenced from the same COODS).

3. For REPEAT the following rules must be observed: For an array of holes, enter the number required along the X axis with the interval Xi, then the number required along the Y axis with the interval Yi.

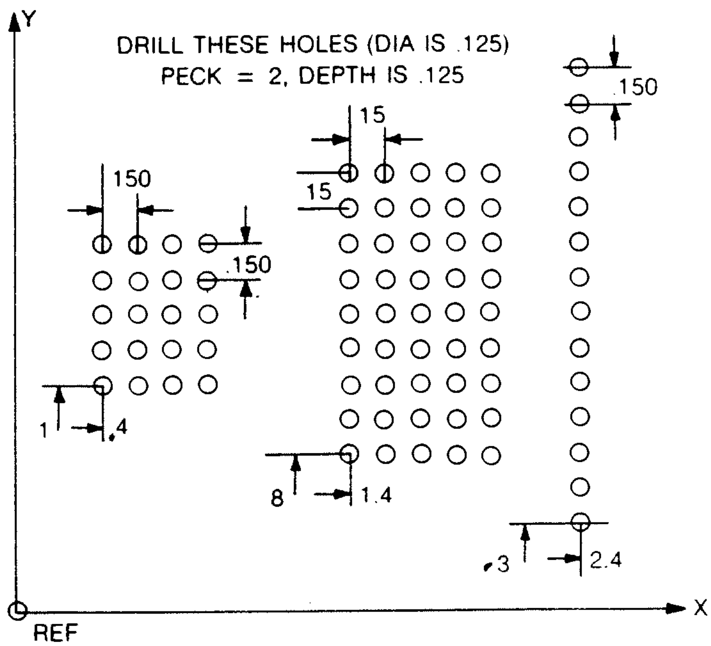
REPEAT X nn from 00 - 99

Xi=

REPEAT Y nn from 00 - 99

Yi=

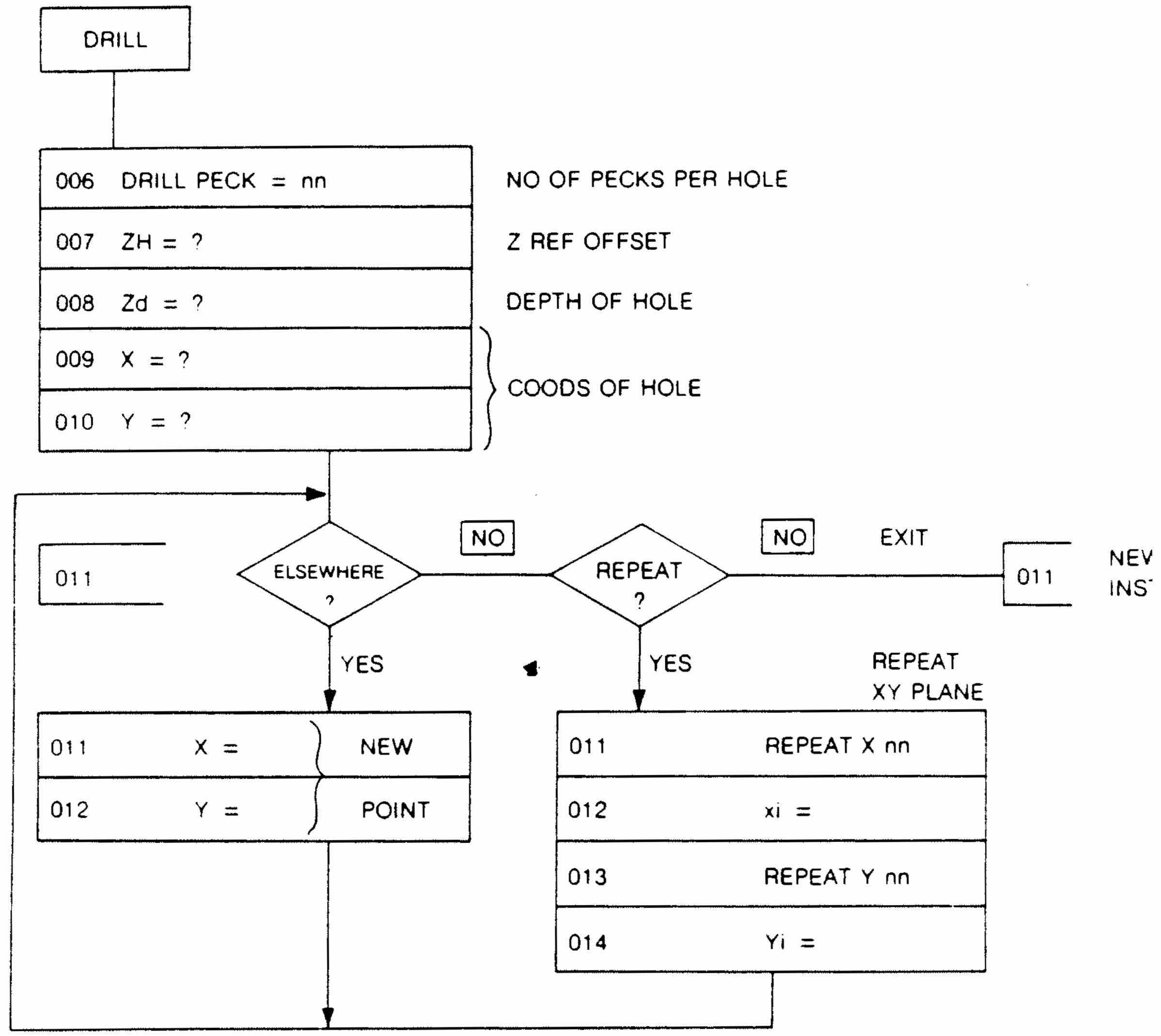
DRILL EXAMPLE



The program would be:

```
000   START INS 01
001   FR XY 16
002   FR Z 10
003   SET UP→zcxyu
004   DRIL PECK=02
005   ZH = 0
006   Zd = .25
007   X = .4
008   Y = 1
009   REPEAT X 04
010   Xi = .150
011   REPEAT Y 05
012   Yi = .150

 (ELSEWHERE)
013   X = 1.4
014   Y = .8
015   REPEAT X 05
016   Xi = .150
017   REPEAT Y 09
018   Yi = .150

 (ELSEWHERE)
019   X = 2.4
020   Y = .3
021   REPEAT X 01
022   Xi = 0
023   REPEAT Y 14
024   Yi = .150
025   END
```

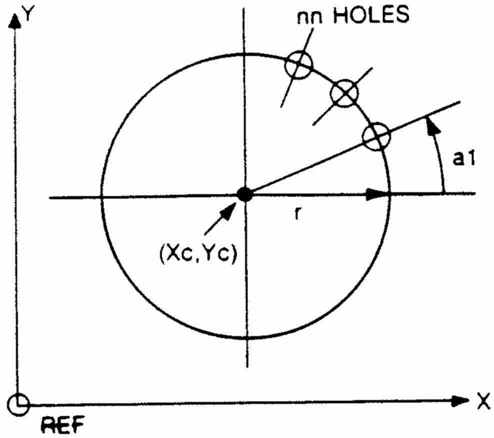
FLOWCHART FOR DRILL



NEW INSTRUCTION

BOLT CIRCLE

The entry format is: BOLT PECK=nn nn=no. of pecks



ZH = Surface displacement from Z ref.

Zd = Depth of hole

XC =
YC = Center of Bolt Hole Circle

a1 = Angle offset from X axis

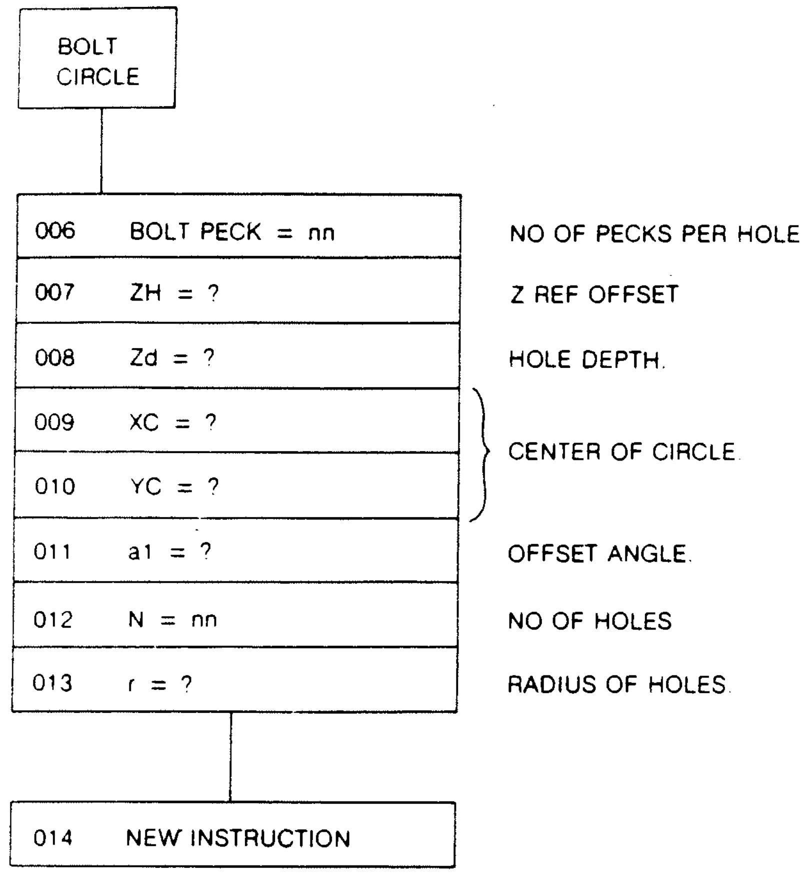
N = nn No. of holes required nn can go up to 99

r = Radius of arc

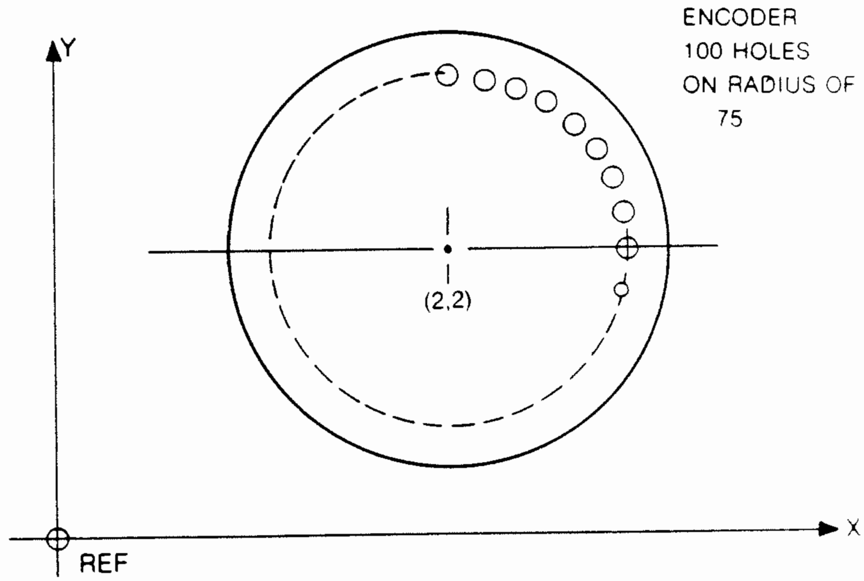
This function generates up to 99 drill holes each pecked nn times to a depth Zd on a circle of radius r at center (XC,YC).

If a greater number of holes are desired, as for example in encoders, simply use another BOLT CIRCLE in the same place, adjusting a1 the starting offset appropriately. For example, if we require 100 holes each at 3.6 degrees, do 50 at a1 = 1.8, then 50 at a1 = 3.6 degrees.

FLOWCHART FOR BOLT CIRCLE



BOLT HOLE EXAMPLE



The program would be:

```
000   START INS 01
001   FR XY = 16
002   FR Z = 8
003   SET UP→zcxyu
004   BOLT PECK = 0
005   ZH = 0
006   Zd = .1
007   XC = 2.0
008   YC = 2.0
009   a1 = 1.8 deg
010    N = 50
011    r = .75
012   BOLT PECK = 0
013   ZH = 0
014   Zd = .1
015   XC = 2.0
016   YC = 2.0
017   a1 = 3.6
018    N = 50
019    r = .75
020   END
```

SECTION 9
A USER EXERCISE

A USER EXERCISE

The best way to learn anything is to do it. We are now at t stage in the manual where some "hands on" experience is the be way too learn the features and capabilities of the machine. Th exercise will take you step-by-step through simple programming a machine operation under program control. However, if you wish y can skip this section and go on to the programming principles.

Users who have never operated a computer-controlled machine, but are familiar with hand operated machines are forewarned. There are no handles on this machine. You cannot start machining at random and sneak up on the final geometry. You must know precisely what you want to machine and all the dimensions of the geometry. Once this is understood and you have done a few machining exercises you will realize that this is the only way to go.

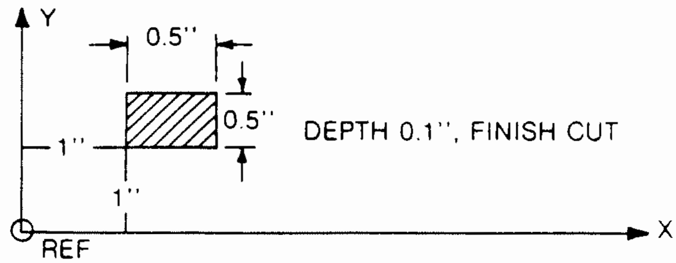
Users who have never machined anything before will, after a few exercises, realize that they can turn out parts with a precision and repeatability that would have required years of machining.

It is recommended that users do a trial run in plastic to get the feel of the machine. Acrylic or plexiglass, and delrin are excellent materials. Cast resins are terrible as the material melts and sticks to the tool. Since there is no mechanism built into the machine to prevent the user from milling or drilling the cast iron table inadvertently, we recommend that initially the thickness of the workpiece be at least 1/2 inch. This will provide enough time to hit the emergency stop button when the Z axis does not stop where you thought you programmed it to stop. The size of the piece should be about 5 x 6". This workpiece should be adequately clamped to the table.

Put a 1/8 inch diameter mill with a 3/8 shank into the 3/8 diameter collet and lock tightly. Make sure the mill is a plunge cutter (i.e. the cutters go to the center of the tool). Some cutters are used only for edging and are not designed for plunging. Lower the head as far as it will go on a vertical shaft and lock tightly. Extend the quill 3/4" approximately to give room for adjustment in height. Put the spindle motor mode switch in the MANUAL MODE. This switch is directly below the main ON/OFF switch. Switch the machine on and after it has figured the backlash, the display will ask MODE?

At this stage let us clear all the memory. Press the LINE MODE key and then press the CLEAR key. Enter 000, then 900 and answer YES to "are you sure?"

At this stage we are free to select the stating line number. Let us keep it at 000 for now. Press the PROGRAM ENTER key, the ref light next to this key will come on. We are now ready to enter the program. Let us make a pocket rectangle as shown below:



Remember, we are entering the start section of the program, so we will have to enter all the machine parameters by pressing START, TOOL DIA, FEED RATE and SET-UP keys in that order. Press the START key. The display will show:

```
000 START INCH?
```

Press the YES key. The display will show:

```
000 START INS nn
```

This is the program Identification Number. Enter 0,1. The display will show:

```
000 START INS 01
```

Press the NEXT key and the display will show the next line

```
001
```

Press the TOOL DIA key and the display will show:

```
001  TD = ?
```

Enter .125. The display will show:

```
001  TD = .125
```

Press the NEXT key and then the FEED RATE key. The display will show:

```
002 FR Axis = ?
```

We will use the same feed rate of 10 inches/millimeter on all axes so touch X,Y,Z and then enter 10. The display will show:

```
002 FR XYZ = 10.0
```

Press the NEXT key and then the SET-UP key. The display will show:

```
003 SETUP zcxyu
```

Press the NEXT key.

We have just finished programming the START portion of the program. Since the geometry we want to program is a rectangular pocket, we will use the pre-programmed RECT POCKET routine. Press the RECT POCKET key. The dislay will show:

```
004 RECT F ?
```

The machine is asking if we want a finish cut. We do, so press YES. The display will show:

```
004 RECT F i ?
```

The machine is asking if we want to make the cut on the inside of the rectangle periphery. We do so press YES. The display will show:

```
004 RECT F i Z%nnn
```

The machine is asking what depth of cut do we want to make on each pass. This must be entered as a percentage of the tool diameter. We will use 50% therefore, enter 050, the display will be:

```
004 RECT F i Z% 050
```

and press the NEXT key. The display will show:

```
005 XY Cut % nnn
```

The machine is asking what width of cut we want to use in the XY plane. We will use the same 50% for the XY cut, so enter 050. The display will show:

```
005 XY Cut% 050
```

Press the NEXT key, the display will show:

```
006 ZH = ?
```

This is the offset, which in this case is 0. Enter 0, display will be:

```
006 ZH=0
```

and press the NEXT key. The display will show:

```
007 Zd = ?
```

This is the total depth of the pocket. The drawing calls for 0.1 inch, so enter .1, display will be:

```
007 Zd=.1
```

and press the NEXT key. The display will show:

```
008    X1 = ?
```

This is the X coordinate of the lower left hand corner of the rectangle. From the drawing, the value is 1. Enter 1 and press the NEXT key. The display will show:

```
009 Y1 = ?
```

This value is also 1. Enter 1 and press the NEXT key. The display will show:

```
010 XA = ?
```

This is the length of the rectangle along the X axis. From the drawing, the value is .5. Enter 0.5 and press the NEXT key. The display will show:

```
011 YB = ?
```

This is the length of the rectangle along the Y axis. Enter 0.5 as above and press the NEXT key. The display will ask?

```
012 ELSEWHERE?
```

Since we want to cut only one rectangle, then press NO. The display will show:

```
012 REPEAT?
```

The answer again is no. Press the NO key. We have just finished programming the middle section of the machine. The next step is to program the END section of the program. To do this press the END key. The display will show:

```
012 END NEW REF?
```

Since we only want to machine one part the answer is NO. Press the NO key and the display will show:

```
012 END NEW PART?
```

Press the NO key again. The display will show:

```
012 END
```

Press the NEXT key to end the program.

The above example has gone into great detail of how to enter a program. This is necessary for the first time. Since you are now familiar with the individual keystrokes then in the future we will skip all the keystrokes in the text and will show only the final program.

This program can be reviewed by pressing the PREVIOUS key. We can go back to the start of the program by pressing this key until we get to the line number 000 or we can switch to LINE mode and enter 000. If an entry error is made, simply press the clear key in the PROGRAM ENTER MODE. One can change the parameters at any time. However, during the ENTRY OF A FUNCTION, one cannot change one's mind half way through the entries. If this function is not desired, the user must enter zeros for all prompts until the end of the function is reached and then go back to the start of the function.

One can erase the function at the beginning line by pressing the CLEAR twice. A prompt will then ask ARE YOU SURE? Pressing YES will erase the function from memory. During the entry of a function, one cannot use the PREVIOUS key half way through. There is no deviating from the entry format in programming a function.

Let us review the program we have just entered. To do this press any mode key. The display will show:

```
MODE?
```

Press LINE NO key and then enter 000. This will put us at the first line of the program.

```
000 START IN 01
```

This shows that we are in the start of the program, and that the program is written in inches and that its label is 01. Stepping through the program can now be done by touching the NEXT key successively till we reach the end. If this is done, the program should read as follows:

```
000  START INS      01
001  TD         =0.1250
002  FR   XYZ    =10.0
003  SET UP ───► zcxyu
004  RECT FiZ%     050
005  XY CUT %      050
006  ZH = 0.0000
007  Zd = 0.1000
008  X1 = 1.0000
009  Y1 = 1.0000
010  XA = 0.5000
011  YB = 0.5000
012  END
```

We are now ready to run the program. Go to line number 000 by entering 000 while in the LINE NO mode and then switch to PROGRAM RUN mode the program will start running and show PRE-CHECKING on display, then the display will ask:

NONSTOP?

The program can be run a step at a time or continuously. If we choose a step at a time, then press the NO key. Otherwise answer YES to NONSTOP?. The display will ask:

SINGLE STEP?

If you answer YES, the program must be singled stepped until it encounters the SET UP instruction. Stop at this instruction and the display will show:

003 SET UP→zcxyu

This is the point at which the user must tell the machine where the reference zero point is on the workpiece. The value for this point can not be entered from the keyboard only set into memory by the "set up ref" key. IT MUST BE DONE BY ACTUALLY POSITIONING THE TIP OF THE TOOL at that point. It also permits the user to manually move the X,Y, & Z axes of the machine with the continuous motion keys and the jog keys.

In addition, the user can also use the optional electronic probe to greatly simplify the mechanical positioning of the tool tip.

In the SET UP prompt the user is asked to enter into the machine the actual X,Y,Z and U location of the reference zero point. (The U is used only with an optional rotary table accessory). The prompt does this as follows. At the start the SET UP display looks like this:

003 SET UP→zcxyu

The arrow in the display is pointing at the coordinate point to be set. All the letters are lower case indicating that the coordinate points have not yet been set. The first coordinate point is the Z coordinate. After this point has been set, as will be demonstrated shortly, the lower case will change to upper case Z indicating that it has been set. Pressing NO will rotate the coordinates so the user can select the required axis. At any time the user can tell which coordinate points have been set by looking at the prompt. The coordnate points that have been set are in upper case. Those that have NOT been set are in lower case.

Now we will demonstrate how the SET UP is executed. There are two ways to do it. One is with the aid of the electronic probe. The other way is manually. We will first illustrate the manual method. First install the end mill into the collet on the spindle. Since the first coordinate point is z, we lower z axis (spindle head) by pressing the [↓] and [Δ ↓] keys until the tip of the tool is approximately 1" above the work surface, move the quill lever until the tool tip just touches the top surface of the workpiece and lock the quill in place. When the tool tip is so positioned press the [SET UP REF] key. The lower case z in the display will change to the upper case Z and the display will now look like this:

003 SET UP→Zcxyu

Since Z has been set and we want to go on to setting c, we must tell the machine to shift to it. This is done by pressing the NO key to the above display. Press NO and the display will show:

003 SET UP→cxyuZ

We are now ready to set c. The symbol c stands for clearance of Z. This prompt asks the user to set the Z axis to a position where the tip of the tool will clear the highest plane on the workpiece when the tool moves from one location to another. This is done by positioning the Z axis (spindle head) using the [↓] and [Δ ↑] keys to the position where the tool tip will physically clear the highest obstruction on the workpiece. When this has been done, then pressing the SET UP key will set the Z clearance position and the display will change to:

003 SET UP→CxyuZ

indicating that Z-clearance has been set.

The rest of the coordinate axes i.e. X & Y are set up in a similar manner.

An alternate and far easier procedure for performing set up is to use the optional electronic probe accessory. To do this, position the electronic probe on the surface of the workpiece and directly below the tool center. Lower the Z axis by pressing the key. When the tip of the tool touches the button on the electronic probe, the Z coordinate will be automatically set. There is no need to press the SET UP key or to accurately press the reference surface with the tool tip.

In the case of the X & Y reference set up, the probe can be positioned so that it's bottom surface touches the sides of the workpiece. The Z axis is lowered so that the tool tip goes past the Z referencce plane. The Z or Y axis table is manually positioned using the [↑] or [↓] keys until the side of the tool touches the button on the electronic probe. When this happens, the X or Y reference is set automatically by factoring in the thickness of the probe and the diameter of the tool.

After the above procedure has been performed, the machine is ready to cut the geometry that has been programmed into the controller. This is done by turning the spindle ON, setting the spindle speed at about position 5 and pressing the NEXT key.

When the pocket is machined the controller will come to the END statement. At this statement, the tool tip will automatically elevate to the Z home position. ‡ go to X0 Y0 —

The above exercise has provided a detailed demonstration of how the machine operates. The user may wish to program and machine some additional functions such as DRILL, MILL, CIRCLE, BOLT CIRCLE, etc. to better familiarize himself with the machine.

‡ IF YOU USED "END NEW PART" IT WILL STOP ON SET UP INSTRUCTION & WAIT FOR NEW PART BLANK TO BE LOADED, THEN IT WILL CUT ANOTHER PART IF YOU PRESS NEXT

actual	Drawing
Tool : .375	.375
XA 1.75	1.5
YB 1.25	1.0
X1 1.875	2.000
Y1 .735	.85
Zd = .285	.250

See 9-3 "for a pocket cycle"

SECTION 10
THE
DYNA
LANGUAGE

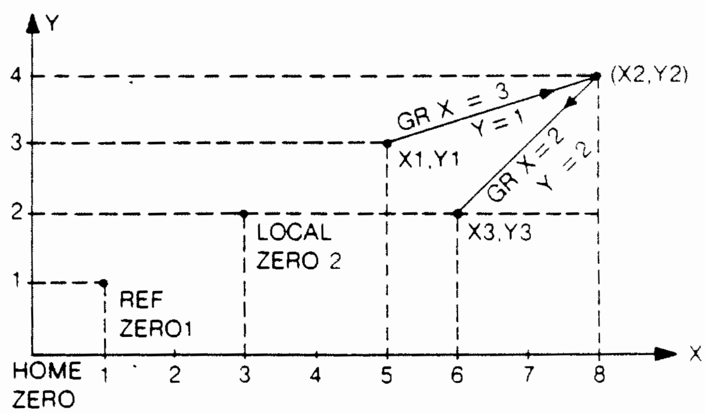
PROGRAMMING WITH DYNALAN (DYNA LANGUAGE)

The DYNALAN consists of a few very powerful instructions whic are designed to simplify machine move programming. These instru ctions are printed in white on the top half of some of the key

The following section will describe each of the DYNALAN in structions, what they do, and how they are used, by using a appropriate example.

GO REL INSTRUCTION - incremental move

This key enters the GO RELATIVE instruction into the program where it will appear as an abbreviation GR. This instructio applies to any X,Y,Z radius (r) or angle (a) moves. This instruc tion will move the tool from its present position to a ne position which is stated relative to the tool's present position For example, in the diagram below ~~the tool is~~ presently i position X1,Y1. If we wish to move the tool to the new positio X2,Y2, we can do so with the GR instruction by entering




```
004    GR    X   3.0
00?          Y   1.0
```

This instruction tells the machine to move the tool + 3 units in the X direction and + 1 unit in the Y direction <u>relative</u> to its present position which is X1,Y1. Likewise, if we then wanted to do another move from the X2,Y2 position to X3,Y3 position we would program:

```
006   GR   X  -2.0
007        Y  -2.0
```

This statement instructs the machine to move the tool minus 2 units in the X direction and minus 2 units in the Y direction <u>relative</u> to its current position which is X2,Y2.

The GR instruction can also be applied in the same way to move the tool in the Z axis. The convention for the Z axis is that all moves down or into the workpiece are negative and all moves up or out of the workpiece are positive. If the tool's tip is presently resting on the surface of the workpiece, then a move INTO the workpiece of 0.1 units is programmed as:

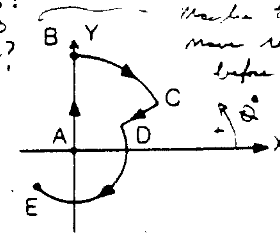
```
008   GR   Z  -.1
```

and a subsequent move out of the workpiece and back to the surface is programmed as

```
009   GR   Z   .1
```

The GR instruction can also be applied to any moves in the polar coordinate system using radius (r) and angle (a) moves. For example, in the diagram below the successive moves from points A to point B,C,D and E, are programmed as follows:

```
010   GO   r    2.0
011        a     90     A ——► B
012   GR   a    -60     B ——► C
013   GR   r   -1.0     C ——► D
014   GR   a   -165     D ——► E
```



Note that in moving from B to C for example, we need not specify a radius because the tool is already at a radius of 2. Likewise from C to D we need not specify an angle. Also note that angle moves in counterclockwise direction are positive and clockwise direction are negative.

<u>GO ABS INSTRUCTION</u>

This key enters the GO ABSOLUTE instruction into the program, applies to any X,Y,Z radius (r) and angle (a) moves. This in-struction will move the tool from its present position to a new position which is stated relative to the LOCAL ZERO coordinate point. If the LOCAL ZERO coordinate had not been specified (or set) then the move will be relative to the REF Zero coordinate. If the REF Zero coordinates had not been specified (or set) then the move will be relative to the HOME coordinate.

For example, referring to our previous diagram if we wanted to move from point X1,Y1, to point X2,Y2, using the GO statement and a previously set local zero 2 as shown we would program as fol-lows:

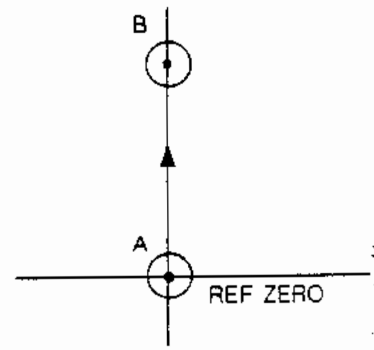
```
014    GO   X   5
015         Y   2
```

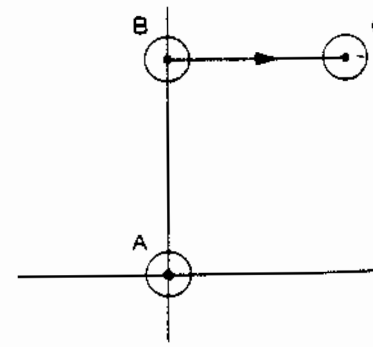
This instruction tells the machine to move the tool from where it is at present, which is X1,Y1, to a point X2,Y2 which is 5 units in the X direction and 2 units in the Y direction relative to the LOCAL ZERO 2 reference. If the LOCAL ZERO coordinate point had not been set but the REF ZERO had been set, then we can make the move to the same X2,Y2 point but the instruction will now have to be:

```
016    GO X   8
017       Y   4
```

Another useful illustration of the GO instruction as it is applied to rectangular or polar coordinates is given below. Here Point A is set as tha REF Zero point.

This Command	Moves Tool
005 GO Y 2.5	A ⟶ B
006 GO X 2.0	B ⟶ C





Mugh, this is the alternative to the software Bug. I will let you know when new version comes. Karen

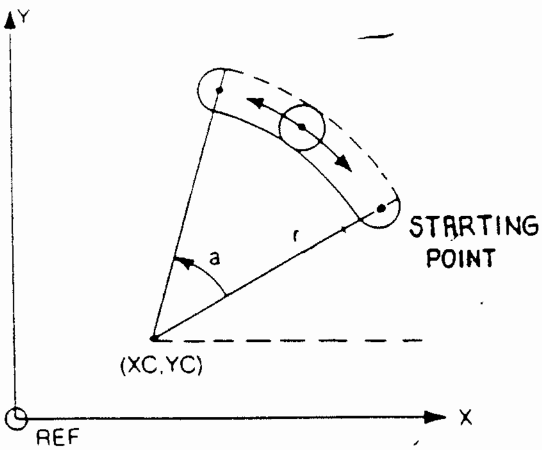
ARC FRAME

The ARC frame canned cycle consists of a combination of *Dynalan and canned cycle programming.

The entry format is:

1) Position tool at starting location including starting depth, using Dynalan programming.

2) Call the ARC frame cycle.



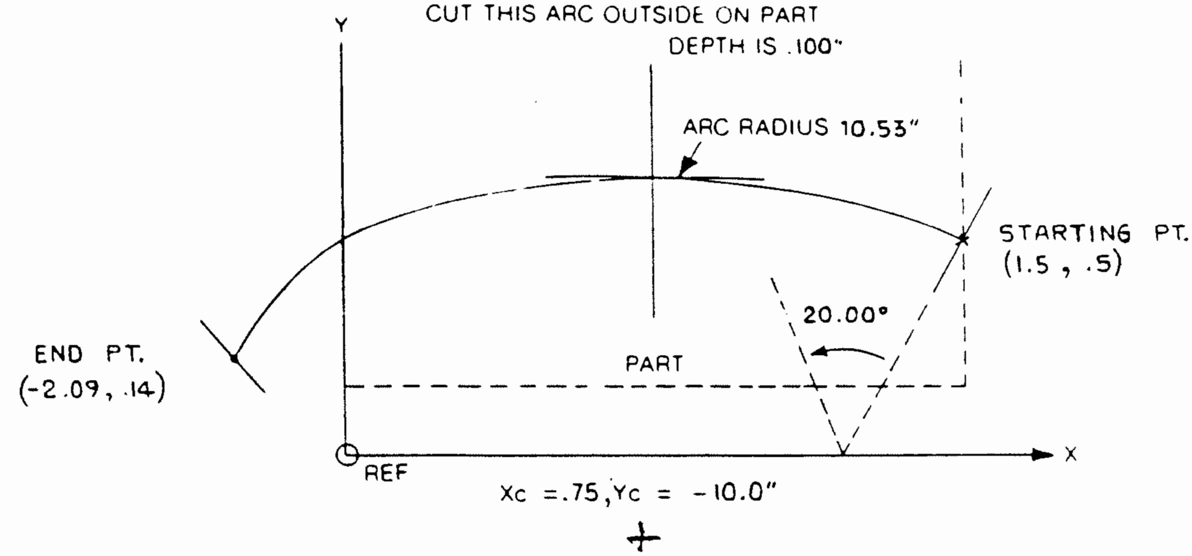
ARC:
XC
YC = Center of ARC, measured from reference zero.
a = traveling angle

This function mills a direct ARC from your established starting point, through a displacement angle, a, in the XY plane. The center of the ARC is at XC, YC and the radius is determined by the distance from the starting point to the center. The tool will remain at this end point until it recieves the next Dynalan command. (a) is minus (-) in a clockwise direction and (+) in a counter clockwise direction (conventional axis notation).

NOTE: The center may be located off the table. The maximum off table distance is 99 inches or 999mm.

* See Chapter 10

ARC EXAMPLE



If we want to make multipass cuts along this same ARC, the format would be as follows:

```
000 START INS 01
001 TD = .125
002 FR XYZ = 10.0
003 SET UP - zcxyu
004*GO X 1.5
005     Y .5
006*GO Z - .100
007 ARC
008 XC = .75
009 YC - 10.0
010 a = 20.0
011*Z - C
012 END
```

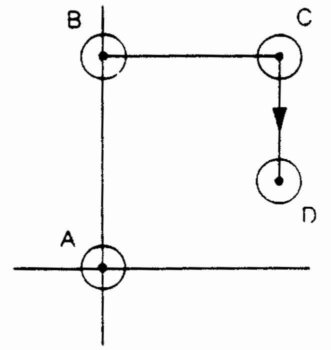
```
000 START INS 01
001 TD = .125
002 FR XYZ = 10.0
003 SET UP - zcxyu
004*GO X 1.5
005     Y .5
006*GO Z 0.0
007*REPEAT 05
008*GR Z - .01
009 ARC
010 XC = .75
011 YC - 10.0
012 a = 20
013*GR Z - .01
014 ARC
015 XC = .75
016 YC - 10.0
017 a - 20.0
018*REPEAT END
019*Z - C
020*END
```

NOTE: This cycle allows uninterrupted machining of complex contours.

* Denotes Dynalan Command

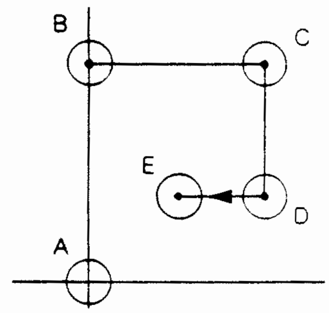
```
007    GO   Y   1.0
```

C → D




```
008    GO   X   1.0
```

D → E



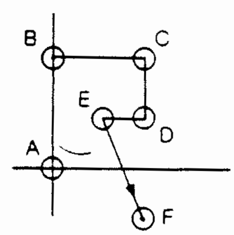
Or using a combination of X and Y:

```
009    GO   X   2.0
010         Y  -1.0
```

E → F

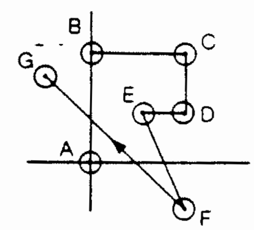
How do I specify RAPID TRAVERSE OR FEED (see 'feed' 10-3)

NOTE THAT COMBINED AXIS MOVES AREN'T IN ONE LINE ∴ YOU MUST THINK OF EACH NEW GO OR GR ETC AS BEGINNING A NEW BLOCK!




```
011    GO   X  -1.0
012         Y   1.5
```

F → G

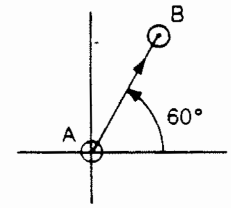


If the tool starts at Z=0 (surface of work)

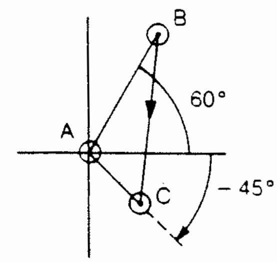
```
013   GO   Z   -.1    will lower tool .1" below surface.
014   GO   Z    0     will raise tool to surface.
```

And in polar coordinates:

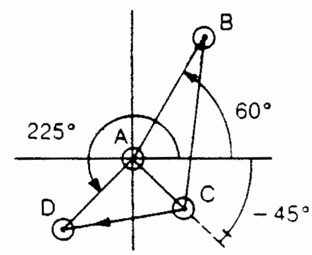
```
006   GO   r   3.0
007        a   60          A ──────► B
```




```
008   GO   r   1.0
009        a   -45         B ──────► C
```




```
010   GO   r   1.5
011        a   225         C ──────► D
```



NOTE: 1. Entering the 2nd axis (or 3rd axis) in this instruction on the next line requires the user to press the axis key TWICE to confirm entry.

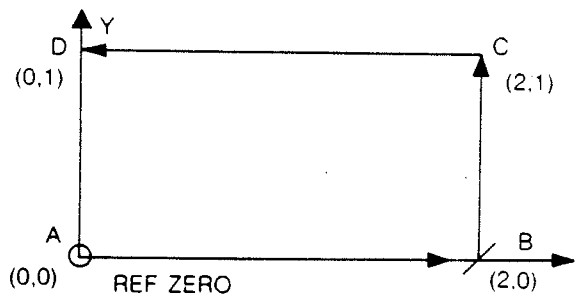
2. Do not mix polar with rectangular in multi axis moves. The correct configurations are:

```
Single axis:  GO X     GO Y    GO Z    GO U    turn table
              GR X     GR Y    GR Z    GR U


2 Axis move:  GO X     GO Y    GO Z    GO r    GO r    GO a
                 Y        Z       Y       a

              GR X     GR Y    GR Z    GR r    GR r    GR a
                 Y        Z       Y       a

3 Axis move:  GO X     GR X
                 Y        Y
                 Z        Z
```

Use of GO and GR Instruction

These two instructions GR and GO are extremely powerful and provide a tremendous flexibility in programming complicated tool paths as we will demonstrate further on. However, care must be exercised in their use. The user must always remember that the GO instruction will always be executed with respect to the last LOCAL ZERO that had been set in the program. Therefore, one should always check the program to see if a LOCAL ZERO had been set or conversely one can set a LOCAL ZERO relative to which the next GO instructions are programmed.

The GO and GR instructions can be used to move to the same new point from the same previous point. Which one the user chooses to use depends on the geometry, the given dimensions and desired simplicity of the program. In other words, GO and GR can be used to achieve the same end but by different means. To illustrate this, let us move the tool tip around from point A, around the rectangle as shown in the diagram below.



Assume that the tool is at point A which is also the REF ZERO in the program. Then instructions..

GO X 2.0 or GR X 2.0

will move the tool from A to B

GO Y 1.0 or GR Y 1.0

will move the tool from B to C

GO X 0.0 or GR X -2.0

will move the tool from C to D

GO Y 0,0 or GR Y -1.00

will move the tool back to A.

The difference between GO and GR is quite simple. The GO assumes one knows the coordinate points. The GR assumes one knows the length one has to move and the direction. Use whatever is appropriate. In this case the GO instruction is appropriate. The program which will make the above moves is then written as:

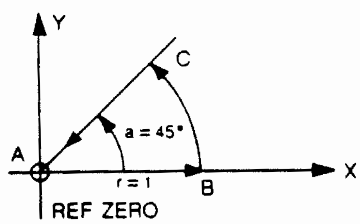
```
000   START INS   02
001   TD = .125
002   FR    YZ = 8
003   SET UP→zcxyu
004   GO   X   2
005   GO   Y   1
006   GO   X   0
007   GO   Y   0
008   END
```

Let us factor in the Z axis so we can lower the tool, cut the rectangle and then raise the tool. Remember that Z reference should be set on the surface of the workpiece. A move downward in the Z axis is always negative. **Let us go .1 of an inch into the workpiece.**
The program becomes:

```
004   GO   Z   -.1
005   GO   X   2
006   GO   Y   1
007   GO   X   0
008   GO   Y   0
009   GO   Z   0
010   END
```

and this will cut the rectangle.

The GO and GR instructions apply equally to radius (r) and angle (a) moves in the XY plane. Consider the following pie slice:



The moves from A to B, C and back to A can be achieved by either GO or GR moves,i.e.

```
A→B:   GO   r   1    or   GR   r   1
B→C:   GO   a   45   or   GR   a   45
C→A:   GO   r   0    or   GR   r   -1
```

Either of the above set will move the tool around the pie. The GO and GR statements can be modified by the use of qualifiers. The generalized format for GO and GR is:

```
GO
or   q* +/-   nn.nnnn
GR
```

Where GO is GO ABSOLUTE, GR is GO RELATIVE, q is an optional qualifier, * is axis choice (X,Y,Z,U,r,a) and nn.nnnn is the number (displayed in inches for this example).

The qualifier (q) is optional and if entered it may be

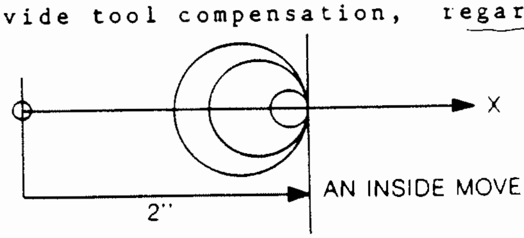
```
i = inside
o = outside
c = comeback
f = fast
```

The i or o qualifiers subtract or add the current tool radius to the entered value. The c qualifier simply causes the tool tip to comeback to its original position. The f qualifier makes the move at top speed.

Each qualifier has its own merit. For example:

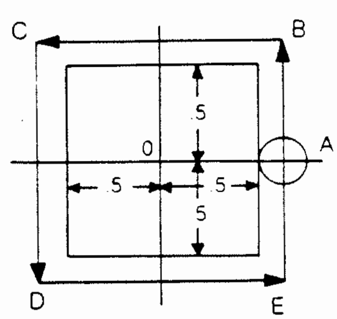
```
GO i x 2.0    or    GR i x 2.0
```

Will always provide tool compensation, regardless of the tool diameter.



This move will always position the tool circumference at the line X=2. Likewise for o on the outside. This makes the move TOOL DIAMETER independent. Consider going round the outside of the unit rectangle with any tool diameter:

```
START AT 0,0 A  GO o X .5
             B  GO o Y .5
             C  GO o X -.5
             D  GO o Y -.5
             E  GO o X .5
             A  GO   Y 0.0
```



The qualifier c for comeback is useful for drilling, sawing etc. For example:

```
GO c Z -.125
```

will lower the tool .125 into the workpiece and return. To speed this up we can use the f qualifier.

```
GO   Z  -.125
GO f Z +.005
```

drill in .125 at current feed rate, return at max speed to .005 above ref.

The GO and GR instructions can be used in multi axis moves.

```
GO  *1 +/- nn.nnnn
GR

    *2 +/- nn.nnnn

    *3 +/- nn.nnnn
```

where * i may be any one of x,y,z,u.

There are NO qualifiers and the maximum number of axes to be moved at one time is three. This maximum is set simply by power supply considerations.

```
For example:   004  GO  X  2.0
               005      Y  1.0
               006      Z   .3
```

will move the tool tip to (2,1,.3) in space and if followed by

```
007  GR  X  -.5
008      Y  -.2
```

will move the tool tip to (1.5,.8,.3)

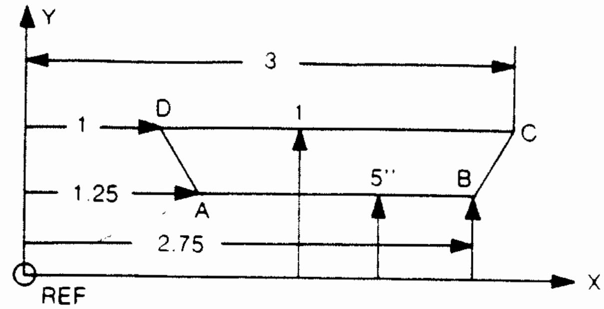
In the XY plane, this example will cut a D hole on the line.

NOTE: Qualifiers will only function with single axis movement.

Then how can you cut an arc (using compensation) (we're Ralph & ...)

what about linear interpolation to pt X-5 Y-4

```
          A:  GO X 1.25
                 Y .5
Lower Tool:   GO Z -.050
          B:  GO X 2.75
          C:  GO X 3.0
                 Y 1.0
          D:  GO X 1.0
          A:  GO X 1.25
                 Y  .5
Raise Tool:   Z ——> Z MAX
```



POLAR MOVES

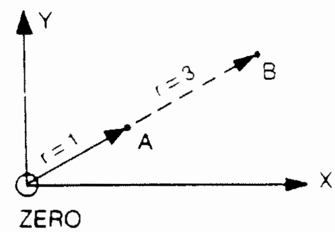
These operate only on the XY plane around a local zero. Here is a summary of the r (radius) and a (angle) moves.

GO r n.nnnn

MOVES TOOL ALONG RADIUS TO r=nn...nnn. This defines a new radius. A negative value is not allowed.

Qualifiers allowed: 1) Inside (subtracts off Tool radius).
2) Outside (adds on Tool radius).
3) Comeback

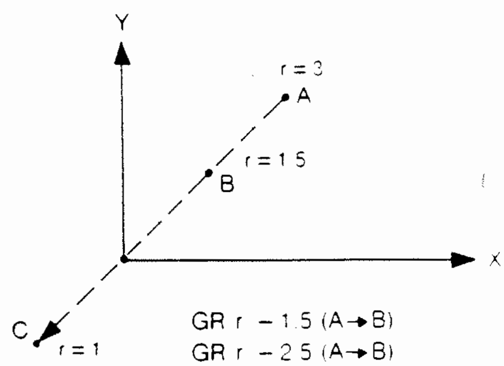
NOTE FAST f. is not allowed.



GO r 3 (MOVE A→B)

GR r + n.nnnn

MOVES TOOL INCREMENTALLY (POSITIVE OR NEGATIVE) ALONG RADIUS BY AMOUNT + nn.nnnn FROM CURRENT RADIUS POSITION. A negative final value will create a new radius 180 degrees from current. Qualifiers similar to above.



GR r −1.5 (A→B)
GR r −2.5 (A→B)

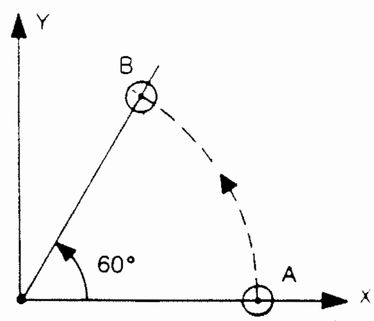
GO a + nnn.nnn

(DEGREES)

MOVE TOOL IN AN ARC TO THIS ABSOLUTE ANGLE AT FIXED RADIUS. Qualifiers are similar to above except inside and outside. Tool compensation is.

$$\angle C = \sin^{-1}\left(\frac{\text{Tool RAD}}{r}\right)$$

which is subtracted or added to the angle.



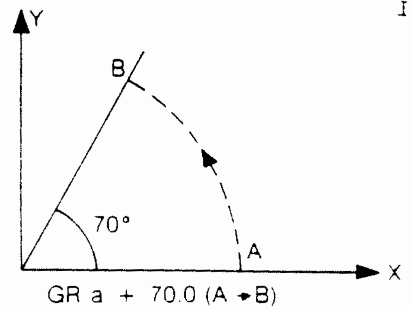
GO a 60.0 (A→B)

(USEFUL FOR PIE CUT)

GR a + nnn.nn..	(DEGREES)

MOVES TOOL INCREMENTALLY IN AN ARC BY THIS AMOUNT OF DEGREES AT FIXED RADIUS. Qualifiers operate like GO a. Observe that: + move tool counter clockwise.
- moves tool clockwise.

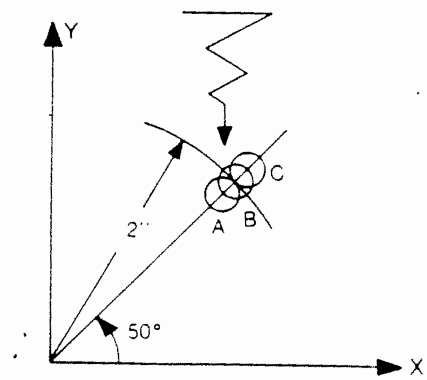
USEFUL IN GOING ROUND CONTOURS.



GO r n.nnn a + nnn.nnn	DEGREES

MOVES TOOL IN A STRAIGHT LINE TO THIS POINT. ON ENTRY r AND a MAY BE INTERCHANGED. Only 2 qualifiers allowed, i and o.
i will subtract off TOOL RADIUS from r.
o will add on TOOL RADIUS to r.
USE OF i AND o MAKES POSITION, INDEPENDENT OF TOOL DIAMETER.
USEFUL IN GOING ROUND CONTOURS.

Position is not actually independent in this mode, but the code for the desired position doesn't require knowledge of the tool radius to be used



TO MOVE TO:

A) USE GO i r = 2
a = 50

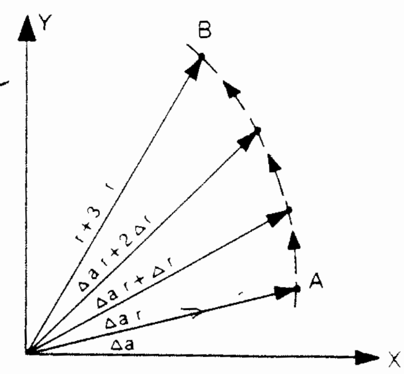
B) USE GO r = 2
a = 50

C) USE GO o r = 2
a = 50

GR r n.nnn a + nnn.nnn	DEGREES

MOVES INCREMENTALLY IN A STRAIGHT LINE TO THIS POINT.

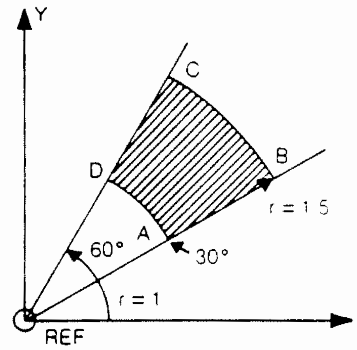
Qualifiers same as GO r,a.
USEFUL IN CAM CUTTING.



O MOVE A ——→ B
REPEAT 4
GR r Δr
a Δa
REPEAT END

Once the generalized polar cood has been entered, we can use the simplified version (treating r and a as one axis each) to move the tool tip. In essence a GO/GR r instruction assumes the value of the angle previously used, likewise a GO/GR a assumes the value of an angle previously used.

For example let us do a pie-pie cut out.



004	GO	r	1	GO to A
005		a	30.0	
006	GO	z	-.1	Lower tool
00	GR	r	0.5	(A → B)
008	R	a	+30.0deg	(B → C)
009	GR	r	-0.5	(C → D)
010	GR	a	-30.0deg	(D → A)
011	Z → Z max			

The critical ingredient in doing polar coods is the setting up of a LOCAL ZERO around which these instructions operate. To do this we will explain the ZERO COO S and the ZERO AT instructions.

"ZERO COODS" INSTRUCTION

This key enters a LOCAL ZERO instruction which instructs the machine to creates a LOCAL ZERO for a set of specified axes at a point where the center and tip of the tool is presently positioned.

To enter this instruction, one must press the ZERO COODS key. The display will prompt with:

007 ZERO AXIS ?

which is a request to specify which axes are to be zeroed at this point. The user can specify any one or all axes that he desires to zero at this point by pressing any combination of the X,Y,Z or U keys. For example if the user wants to zero X & Y he has to press the X and Y key and the display will respond with:

017 ZERO X Y

This procedure will have set a LOCAL ZERO at the present center position of the tool.

Another method of setting a LOCAL ZERO is to use the ZERO AT instruction.

ZERO AT" INSTRUCTION

This key enters a LOCAL ZERO instruction which instructs the machine to create a LOCAL ZERO at a point that is specified by the user, within this instruction. For example, pressing the ZERO AT key will result in the following display:

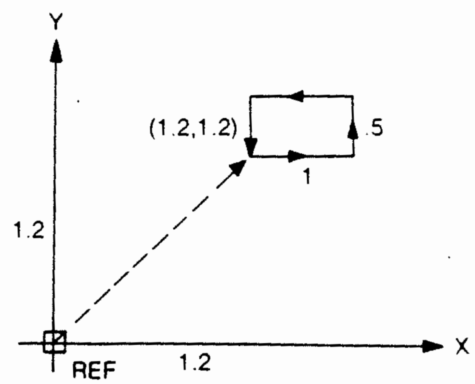
18 ZERO AT

The user then has to press the NEXT key to enter this instruction into the program and then specify the axis and their value at which he desires to create a LOCAL zero. The new LOCAl zero will be set up relative to a previous LOCAL ZERO if one had been set-up or relative to the REF. The entry format is:

```
ZERO AT

X    nn.nnnn

Y    nn.nnnn

Z    nn.nnnn

U    nnn.nn
```

The entry of selected axis and coordinate value permits the setting up of a LOCAL ZERO without moving the tool tip. The following example will illustrate the use of these instructions, given the geometry and the dimensions, as shown in the diagram below, it is very easy to write the program using the ZERO COODS instructions.

```
004  GO    X   1.2
005        Y   1.2
006  ZERO  XY
007  GO    Z   -.1
008  GR    X   1
009  GR    Y   .5
010  GR    X   -1
011  GR    Y   -.5
012  Z——►Z MAX
```

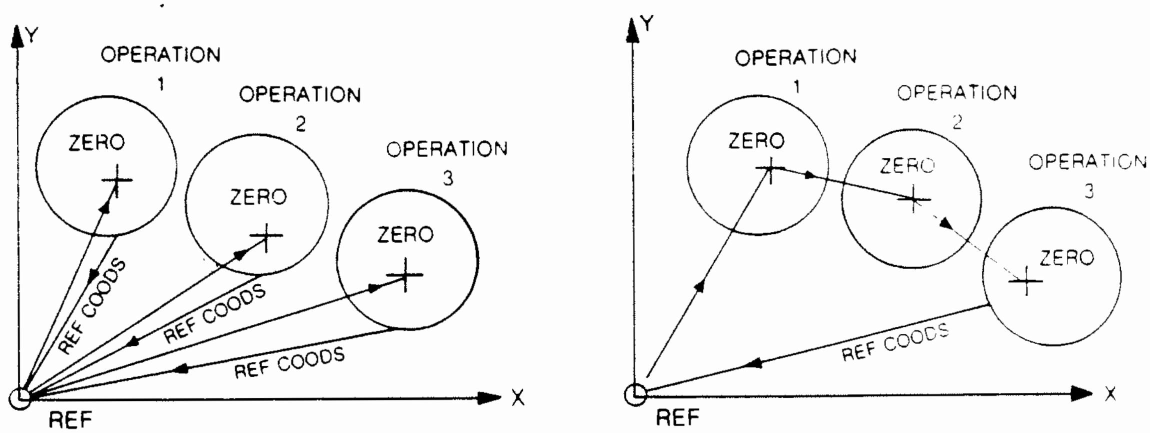


This greatl, simplifies the writing of programs from very simple drawings or even sketches. The drawings need not specify the coordinates for each corner-just the dimensions of the geometry and a location of one of its corners relative to the REF zero.

Any LOCAL Zero set-up by the ZERO AT or ZERO COODS instruction can be cleared by using the REF COODS instruction.

REF COODS INSTRUCTION

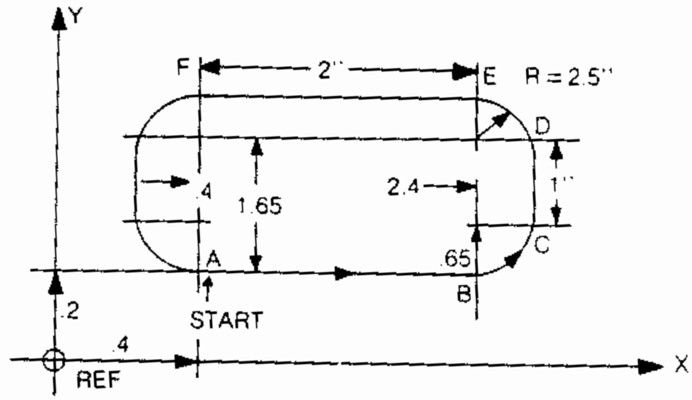
This key is an instruction which restores the REF ZERO coordinates in the program. It is usually used in returning from local zero to the REF ZERO coordinates which was set-up during the SET UP instruction. The diagram below illustrates its use.



At any place in the program, the next operation may be referenced from the REF coordinates or it may be referenced from the previous operation's local zero. In the first case any change in the zero location is easy to change and the operation is independent of the other operations. In the second case any change in the zero location of operation 1, propagates through to the other operations so the whole block will be repositioned. Depending on the application this may or may not be advantageous.

The user can use the ZERO AT, ZERO COODS and REF COODS instruction in any sequence he desires and in any location. He can go from one LOCAL zero to another or back to REF ZERO in between each LOCAL zero.

As an illustration of how REF COODS, ZERO AT, and ZERO COODS Instructions work, let us program a frame rectangle cut with a radius at each corner Remember that for an arc (or circle), we must set a local zero at the center and then move through an angle.

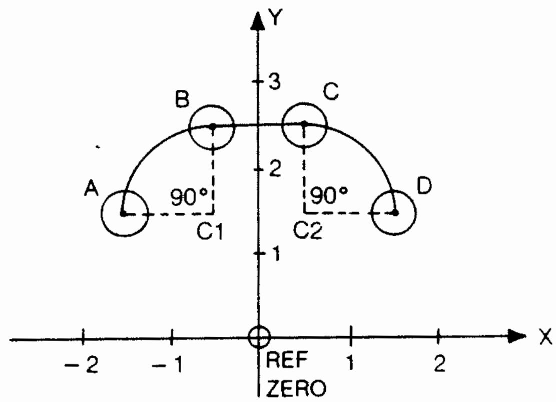



```
004   GO   X   .4          Move tool to A
           Y   .2
006   GO   Z  -.050        Lower it .05 into the work.
007   GR   X               Move to B
008   ZERO   X             Zero X then ZERO Y AT .65
009   ZERO A                 (the ARC center)
010        Y .65
011   GR   a   90          Cut ARC B to C
012   GR   Y    1          With same ZERO, go from C to D
013   ZERO Y               ZERO Y at ARC center
014   GR   a   90          Cut ARC D to E
015   GR   X -2
016   ZERO X
017   GR   a 90
018   GR   Y -1
019   ZERO   Y
020   GR   a 90
021   END
```

THE USE OF GO, GR, ZERO COODS, ZERO AT INSTRUCTIONS

The ability to be able to set-up LOCAL ZERO and to make GO and GR moves relative to these points permits very easy programming of very complicated shapes and contours which are composed of linked geometric arcs and straight lines. The ability to do this is best illustrated by examples.

Suppose we want to cut a path as shown in the drawing below. With the tool center running on the path. We will assume that the tool starts from any arbitrary position. We program as follows:



LINE NO.	PROGRAM STATEMENT	WHAT THIS STATEMENT DOES
004	GO X 1.5	Moves the tool from any position to point A referenced
005	Y 1.5	to the REF ZERO.
006	GO Z-.05	Lowers the tool .05 units from the REF Z position (Z is workpiece surface)
007	ZERO Y	To cut the arc A to B with C1 as center we create a LOCAL ZERO at C1. Since the tool is already at A, which Y location is the same as C1, we zero Y using the ZERO COODS instruction.
008	ZERO AT	We then zero X using the ZERO AT instruction.
009	.5	
010	GR a-90	To end the arc from A to B we use GR instruction in polar coordinate, remember that a clockwise angle move is negative.

011	GR X 1	To go from point B to C in a straight line we use the GR instruction again for an X move only since the tool is already at the required Y coordinate.
012	ZERO X	To cut the arc from C to D we set up a new LOCAL ZERO at C1 We need only to ZERO X since the Y coordinate of C2 is the same as C1.
010	GR a-9	We now cut the arc using the GR statement.

Note 1-Notice that the REF coordinate had been set up in such a way that some areas of the workpiece are in the -X,-Y, plane. This illustrates that REF coordinate point can be set up anywhere. If it had been set up as the lower left hand corner of the workpiece, then all of the above GO statements would have positive direction.

XY- REF 0 INSTRUCTION

This key is an instruction which simply moves the center of the tool to the REF Zero point. It can be used to automatically position the tool at the end of the program so that it will be in the right position when a new part is to be clamped on the table and positioned to the program REF. coordinate.

The move actually goes via the home position for verification of the position of REF ZERO point, and is analogous to the END NEW PART instruction in this manner of movement. Thus it can be used as well in very very complex go relative angle sequential moves as an additional verification on accuracy on exit from them.

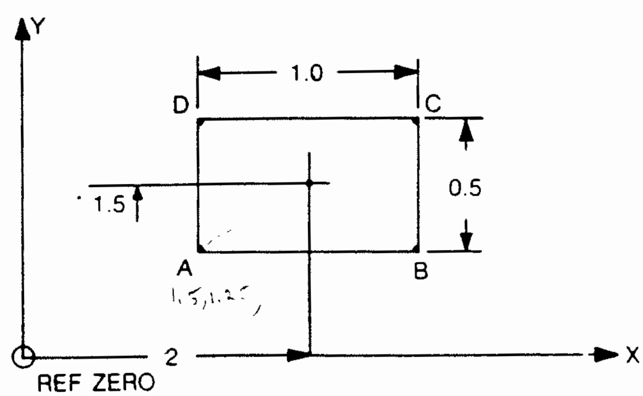
CS INSTRUCTION (Change Sign)

This is an instruction which simply reverses the sign of the current tool tip position. It can be used with any axis i.e. X,Y,or Z.

For example if the tool tip is at X=2, CS X will move the tool to X=-2.

Many machining geometries exhibit symmetry and by exploiting it one can greatly simplify the programming and minimize entry of coordinate points.

To illustrate it's use, let us do a rectangle frame cut (1" x 0.5") centered at (2,1.5).
To program this we need only to compute one corner A (1.5,1.25).



The program i then written as follows:

005	ZERO AT	Sets up a local zero at (2,1.5)
006	X 2	
007	Y 1.5	
008	GO X - .5	Moves the tool to point A
009	Y - .25	
010	GO Z - .05	Lower tool into workpiece to a depth of .05 units
011	CS X	Will move the tool from point A to B to C to D to A
012	CS Y	
013	CS X	
014	CS Y	

Alternatively the instructions could be:

CSY   Move the tool from point A to D to C to B to A
CSX
CSY
CSX

CYCLE XY INSTRUCTION

This Key is an instruction which combines The previous four CS instructions into one. Thus after zeroing at the center of the rectangle and moving to one corner, CYCLE XY will cut the frame. Depending on which corner the tool starts at, it will go clockwise or counterclockwise round. If one slides down the diagonal after each cycle we will generate a pocket cut.

Thus the previous example reduces to:

```
ZERO AT
     X  2
     Y  1.5
GO   X  1.5
     Y - .25
CYCLE XY
```

This will cut A→B→C→D→A

DWELL INSTRUCTION

This key is an instruction which will cause a elay of nn seconds in the program. Thus DWELL 06 will cause a program delay of 6 seconds.

DISPLAY INSTRUCTION

This key is an instruction which will display the current value of X,Y,or Z position of the tool. One can enter this instruction into the program anywhere and as many times as desired. During execution, the controller will stop and display the current value at the axis selected. All of the examples below are valid entries for this instruction.

DISPLAY XYZ
DISPLAY X
DISPLAY r
DISPLAY a Z

This instruction is very useful for debugging programs.

CONTROL INSTRUCTION

This key is an instruction to activate peripherial devices.

CONTROL 1 activates a buzzer.The buzzer will sound for 5 seconds. Combined with a dwell instruction various sounds may be generated CONTROL 2-9 are un-assigned at the present.

SPINDLE ON/OFF INSTRUCTION

This key is an instruction to activate or deactivate the spindle. When pressed it will display:

SPINDLE OFF

Pressing the +/- key will rotate OFF to ON for the user to select what he wants.

USE THIS WITH CARE and insert only after the program has been debugged and is working. The switch underneath the main ON/OFF switch should be switched from LOCAL to PROGRAM. The spindle ON/OFF should be switched to ON (the spindle motor will remain stationary) the speed dial should be set the desired speed. During program execution this instruction will turn the spindle ON. The user may still override this with the spindle OFF/ON switch (i.e. it may be turned OFF then ON for tool change).

Alternatively, we can program spindle off before a tool change and spindle on after. Just before the end, SPINDLE OFF should be inserted. This instruction is very useful for running a program overnight.

TOOL CHANGE INSTRUCTION

This key is an instruction which permits the user to change tools which are required by the program and the geometry of the part. There are three ways to change a tool in the middle of a program.

1. THE SIMPLEST WAY

During program entry simply put in this sequence in instructions:

```
... Z ──► Z MAX
... HALT
... GO Z 0
... HALT
... Z ──► CLEAR Z
```

During execution the controller will raise the spindle to it maximum height then halt. Insert the new tool and raise the quill. Press the NEXT key and the tool tip will descend to the Z REF position. Adjust the quill so the tip of the new tool touches the surface. Lock the quill and press the NEXT key to continue. The tool is rezeroed manually To do this, the first tool must be the longest one.

Sometimes the old Z REF position may have been machined away. In this case some GO statements are necessary before this section to position the tool over a Z ref area.

Observe that this little program block may be written as a subroutine and used many times in the program.

2. USING THE TOOL CHANGE INSTRUCTION WITH NO OFFSET STORAGE

Pressing the TOOL CHANGE key, the user will be asked.

TOOL n?

Press the NO key and the display will ask:

TC NEW COOD ?

Here the controller wants to know if the user wishes to reference the new tool tip at a different location. If the answer is NO then the tool change operation will take place at the REF COODS. During execution, the controller will raise the tool (Z ──► Z MAX). The user changes the tool, places the probe at the REF COODS, then by pressing the NEXT key, the controller lowers the tool tip until the probe is touched. Thus the TOOL is ZEROED at Z = 0. Clear Z remains as previous and the tool goes to this point.

SETTING TOOL OFFSETS

However, had the user requested new coordinates, the display entry would show:

TC NEW COOD

and the user would be prompted for:

TX =
TY =

the tool change coordinate points. The sequence is exactly as before except the re-zeroing is done over these new coods. The tool tip will be positioned to these coods, the tool changed, and then re-positioned back, via clear.

3. USING THE TOOL CHANGE INSTRUCTIONS WITH OFFSET STORAGE

This time after pressing the TOOL CHANGE key, the user answers YES to

TOOL n ?

The display becomes TOOL N then the user enters the desired tool number, ($1 \leq n \leq 8$)
1 thru 8

The user must employ tool holders where the length of the tool tip from the spindle is fixed either by set screws in the tool holder or pre-clamped in a jacobs chuck. There is no guarantee that the collet will hold the tool at the same height since they have no end stop. The tool offsets (from Z REF) are entered sequentiallly before starting the program via the TOOL CALIBRATION in the MANUAL MODE. (see Manual Mode section for further details)

Thus when programming the user simply enters the tool number AND immediately afterwards the TOOL DIAMETER.

For example:

```
....GO X 2.4
... TOOL 6
... TD = 1.358
... GO Z 1.6
```

for more info see 5-3

would put in TOOL 6 (dia. 1.358) at this point in the program.

During execution, the program stops, the spindle is raised and the user inserts the TOOL with the correct number. Pressing the NEXT key activates the program again.

HALT INSTRUCTION

This key is an instruction which halts the running of a prograı It may be put anywhere in the program. It is extremely usef: initially to incorporate these (for example between functions to check progress, do a simple tool change and to have in ɛ infinite loop. When the controller comes to this instruction, : halts. Pressing the NEXT key will continue program executioı The user may also exit via a mode key (to abort the program a this point or change some instruction) then re-enter and contint

SKIP INSTRUCTION

This key is an instruction which tells the program to go to specific line number and continue the program from that liı number. It is an unconditional jump either forwards or backwarc within a program. This instruction is useful in repeating ceı tain parts of a program automatically. For example, the followiı program is a perpetual hole driller

```
000   PROG START   INS  01
001   TOOL DIA
002   FR XYZ = 10.0
003   SET UP——➔zcxyu  (put part in jig and set up)
004

      HOLE DRILL       ANY
      ROUTINE          OPERATION

016
017   HALT             (take out part-press
018   SKIP 0            NEXT to continue)
019   END
```

After the routine has been executed, it will halt at 017. Tl user can remove the part, put a new one in, press the NEXT kı and the program will jump to 004 and continue ad infinitem. Tl ONLY operations the user has to do is load in the new part a: then press the NEXT key at each halt.

SUBROUTINE
SUB RETURN INSTRUCTIONS

These two keys are instructions which tell the program to go to specific section of the program, execute that section and th return to continue the program.

Very often in long programs a particular operation is repeated many times. For example, the user might wish to write his own drill and peck routine, a set up routine, a tool change routine or some particular geometry that is to be repeated elsewhere. It is a severe imposition to have to repeat these instructions throughout the program, hence subroutines:
This section is written as:

```
SUB nn                                    ( 00 ≤ n ≤ 99 )
- - - -
- - - -  LITTLE SECTION
- - - -
SUB RETURN
```

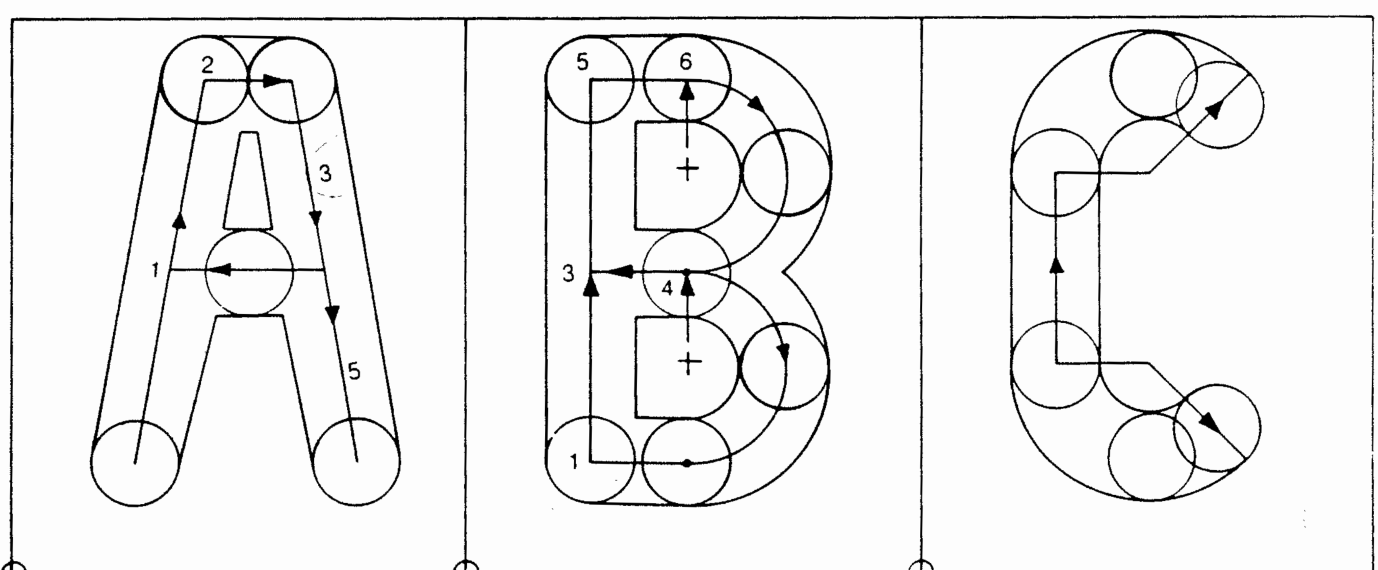
and can be put anywhere in the program space 000-900 line number. Subroutines are usually placed at the end of the program space. When the user wishes to call this subroutine in his main program he puts in:

```
CALL SUB nn
```

The controller will branch to this subroutine during execution, execute it, then return to begin executing the next statement of the program.

There is no restriction on the number of nesting levels i.e. subroutines can call other subroutines.

Here is an example of how they are used. Suppose the user wanted to mill characters (A,B,C, etc.). Let us use a 1/16" DIA MILL (a lot depends on how small a diameter mill is used) and let us make the characters .34" high by .23" wide. If the user is not happy with the aesthetics he may compose his own. We have scaled up the characters (X10) to illustrate the moves per character. By inspection one can enter the move coordinates.



LOCAL ZERO for A   LOCAL ZERO for B   LOCAL ZERO for C

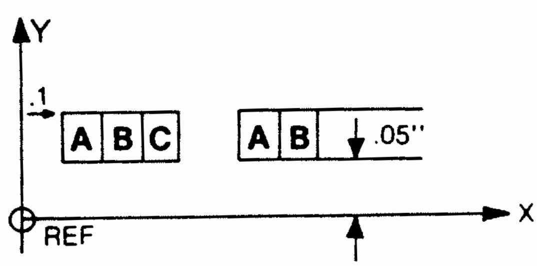
```
SUB 01

ZERO   XY
GO   X   .04
     Y   .05
GO   Z  -.05
GR   X   .05
     Y   .24
GR   X   .03
GR   X   .025
     Y  - .12
GR c X  -.08
GO   X   .13
     Y   .05
Z ——→ C
GO   X    0
     Y    0
→ REF COODS
SUB RETURN
```

```
SUB 02

ZERO   XY
GO   X   .05
     Y   .05
GO   Z  -.05
GR c X  .07
GR   Y   .12
GR c X  .07
GR   Y   .12
GR   X   .07
ZERO   AT
     X  .07
     Y  .24
GR   a  -180
ZERO   AT
     Y  -.12
GR   a  -180
Z ——→ C
GO   X  -.12
     Y  -.11
→ REF COODS
SUB RETURN
```

```
SUB 03

ZERO   XY
GO   X   .05
     Y   .12
GO   Z  -.05
GR   Y   .1
ZERO   AT
     X   0.12
     Y   0.22
GR   a  -135 deg.
     Z ——→ C
ZERO   AT
     Y  -.1
GO   r   .07
     a  -45 deg.
GO   z  -.05
GR   a  -135 deg.
Z ——→ C

GO   X  -.12
     Y  -.12
→ REF COODS
SUB RETURN
```

Each of the above subroutines will mill a specific character. If we now wish to lay out the characters as shown in the example below, then the main program would look like this.

```
000   START INS   04
001   TD = .0625
002   FR   XYZ = 10
003   SET UP→ zcxyu
004   GO   X   .1      Go to botton LH side of
005        Y   .05      1st character
006   CALL SUB 01     Do the "A"
007   GR   X   .23
008   CALL SUB 02     Do the "B"
009   GR   X   .23
010   CALL SUB 03     Do the "C"
011   GR   X   .46      SPACE
012   CALL SUB 01     Do the "A"
013   GR   X   .23
014   CALL SUB 02     Do the "B"
015   END
```



The subroutines can be located arbitrarily from 300 onwards (in any order). Observe that in each subroutine we immediately set a local zero and do all moves with respect to it. When we exit we go to the local zero then switch back to the REF COODS so the main program can locate the characters where it pleases.

looping

REPEAT
REPEAT END INSTRUCTIONS

These keys are instructions which permit the program to repeat any section of a program a specified number of times. From 1 to 99. This is done by enclosing the section of the program by REPEAT nn and terminating it by REPEAT END. For example:

begin with

No of passes thru loop!

end with

```
REPEAT 20

OPERATIONS

REPEAT END
```

will repeat the operation 20 times.

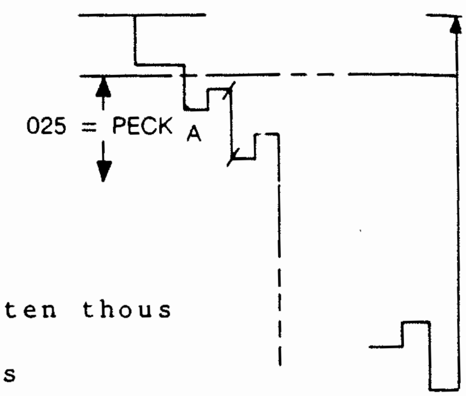
These REPEATS can be nested to any level, providing there is always a REPEAT END for each REPEAT nn. For example all of these are valid.

```
┌REPEAT 10
│
│ ┌REPEAT 20
│ │
│ │   ┌REPEAT 60
│ │   │
│ │   └REPEAT END
│ │
│ │   ┌REPEAT 40
│ │   │
│ │   └REPEAT END
│ │
│ └REPEAT END
│
└REPEAT END
```

In this example, the operations in the center are repeated 60 x 20 x 10 (=12000) times. This instruction is extremely useful in many applications. For example in pocketing and framing, we repeat the operation each time incrementing Z which in turn is repeated in the X & Y axis.

Here is one example. Suppose the user wishes to write his own drill routine with small pecks. We shall write it as a subroutine with a REPEAT in it. The main program would be...

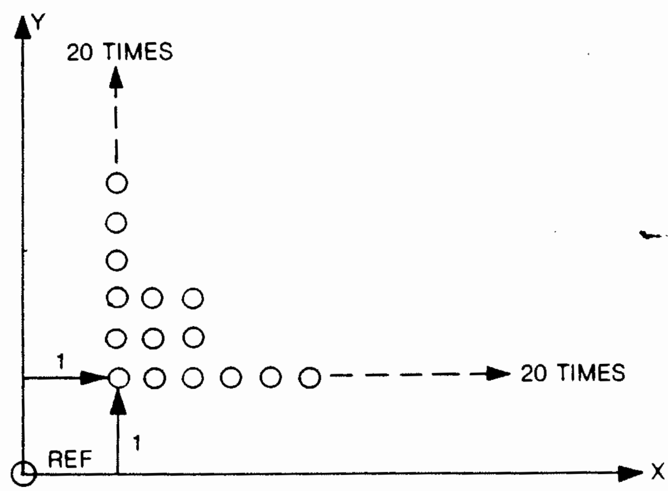
```
MAIN PROGRAM        SUBROUTINE

----                SUB   99
CALL SUB  99        GO  f Z   .005
------              GO    Z   -.025
                    REPEAT  9
_______             GR   f Z +.010    Clear drill
                    GR   f Z -.010
                                      GO up down ten thous
                    GR    Z   -.025   drill pecks
                    REPEAT END        Loop 9 times
                    Z——►Clear Z
                    SUB RETURN
```



In this example the A total depth is .025+9 X .025=.25"

Now suppose we wish to use this in a 20 X 20 hole pattern in the XY plane. The program would be:

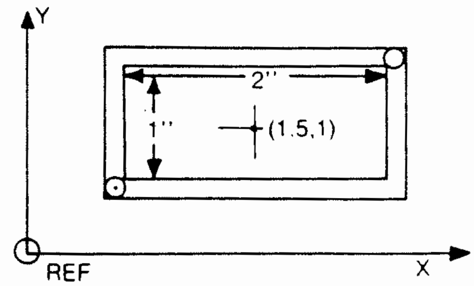
```
000   START INS  05
001   TD = .0625
002   FR   XYZ = 15
003   SET UP→zcxyu
004   GO   X   .1
005        Y   .1
006   REPEAT  20
007   REPEAT  20
008   CALL SUB  99
009   GR   X   .1
010   REPEAT END
011   GO   X   .1
012   GR   Y   .1
013   REPEAT END
014   END
```



Another example is the use of REPEAT/REPEAT END in Frame cuts.

It is usually difficult to frame cut in one pass, particularly when the material is thick. It is necessary to make several passes.

Here is an example of a simple rectangular frame cut. The material is .25" thick and the tool is .125" in diameter. Each pass we lower Z the desired increment.




```
000   START INS   06
001   TD = .125
002   FR   XYZ = 10
003   SET UP→zcxyu
004   ZERO   AT              Zero at center
005          X   1.5        of rectangle
006          Y   1.0
007   GO o X -1.0
008   GO o Y -0.5             Go outside of rectangle-Lower LHS
009   GO    Z   0             Lower tool to surface
010   REPEAT   5              Do 5 passes
011   GR    Z   -.05          each of 50 thousands
012   CYCLE XY                Cut rectangle
013   REPEAT END
014   END
```

This is exactly how the FRAME function works. For more complex geometries the cycle XY instruction (line 012) is replaced by the move statements necessary to cut the required geometry.

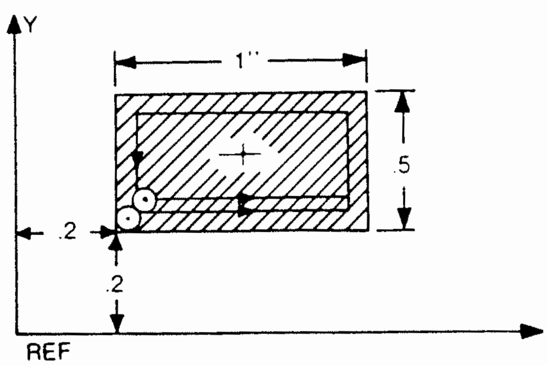
Observe that the END statement will automatically home the Z axis first, then X and Y.

Pockets can be made by doing successive frame cuts. Each frame is made smaller until the center is reached. This is an ideal situation for a REPEAT instruction.

Here is an example:

Suppose we wish to mill out a pocket 1"x.5" to a depth of .1" with a .125 dia tool as shown below

```
000   START INS   06
001   TD = .125
002   FR XYZ = 10
003   SET UP→zcxyu
004   ZERO AT
005        X   .7             Zero at center
006        Y   .45            (of rectanle)
007   GO i X -.5
008   GO i Y -.25             Go to lower LHS side
009   GO   Z   -.1            Drop tool
010   REPEAT   3              Repeat 3 times
011   CYCLE XY                Cut frame
012   GR   X    .1            Reduce frame size
013        Y    .1
014   REPEAT END
015   END
```



How did we set the repeat at 3? This hinges on the size of the frame reduction (lines 012 and 013) and hence the size of the XY cut. It is usual to allow some of the tool diameter to be outside the cut to clean up the previous cut so we moved the tool in .1" (not .125). Now the critical dimension is .25, the distance from the X edge of the rectangle to the center. This is the shortest, so 3 frames of .1 will cover 2.5.

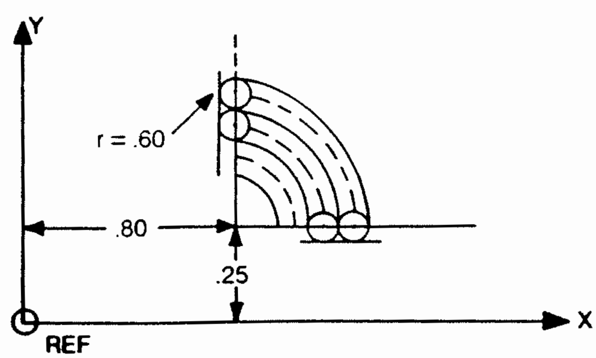
In the previous example the depth is .1", suppose we had wanted a depth of .3". We simply repeat on the Z increment. The program becomes:

```
000   START INS  06
001   TD = .125
002   FR  XYZ = 10
003   SET——►zcxyu
004   ZERO AT
005       X  .7
006       Y  .45
007   GO i X -.5
008   GO i Y -.25
009   GO   Z  0.0      Drop tool to surface
010   REPEAT  3
011   GR  Z -.1        Increment Z into work
012   REPEAT  3        on each pass
013   CYCLE  XY
014   GR  X -.1        3 passes
015       Y -.1
016   REPEAT END
017   GO i X  -.5   Go back to start
018   GO i Y -.25
019   REPEAT END
020   END
```

The function pocket mill does essentially the above program. It calculates the repeat n's based on Z% and XY%.

Arc pocketing is done similarly. Suppose we wished to pocket an arc 90 Deg on a radius of .60 ", with a 1/8 inch dia mill, to a depth of .1". We snall move the tool along the radii to clean up the tool cuts.

```
000   START INS  07
001   TD = .125
002   FR  XYZ = 10
003   SET UP—►zcxyu
004   GO  X  .8
005       Y  .25
006   ZERO  XY
007   GO  Z  -.1
008   GO  r  .5375
009   GO  a  90
```



```
010  GO  r   0
011  GO  r   .5
012  REPEAT  2
013  GR  a  90
014  GR  r  -.1
015  GO  a  0
016  GR  r  -.2
017  REPEAT END
018  END
```

PROG REF. INSTRUCTION

This key is an instruction which takes the current position of the tool center and makes that position the REFERENCE coordinate point. Up until now the only mechanism to set the reference has been through the SET UP key. This involves manual intervention by the operator which in some cases may be good while in others onerous. For example, it allows the user to do this:

```
000  START INS  02
001  TD = .125
002  FR XYZ = 10
003  GO  X  .1
004      Y  .1
005  GO  Z  2.0
006  HALT
007  PROG REF        SET REF point at (.1,.1) in space
008  .......
```

The user MOVES the part to this location point for his reference, then clamps it in the jig. The halt can be removed later. The next time the program is run, it will go non-stop. This pre-supposes tool length etc. remain invariant.

It also allows the user global program repeats. Just before the end tool is moved to the new reference and is set by PROG REF, we then REPEAT the whole program.

NOTE: Prog Ref is only good for X and Y set up.

SECTION 11
WORKED EXAMPLES

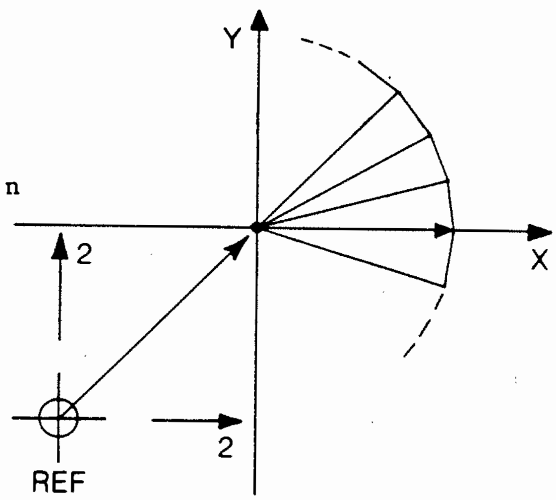
EXAMPLE 1

The object is to cut a polygon with 17 sides and radius 1 inch. It illustrates the use of the instruction.

GR	r
	a

The program becomes:

```
000   START INS 01
001   TD = .125
002   FR   XY = 10
003   FR   Z = 6
004   SET UP→zcxyu
005   GO   X    2     Zero at center
006        Y    2     2,2 from ref. point
007   ZERO   XY
008   GO   X    1     Go to start of polygon
009   GO   Z   -.050  drop tool
010   REPEAT 17
011   GR   r   0.000
012        a   21.176
013   REPEAT END
014   Z→clear Z
015   END
```



NOTE:

1. Observe that a GO relative (r,a) instruction will move the tool in a straight line from the current position to the next.

2. In the above example, r was kept zero so there was no increment in r, but "a" was incremented by 21.176 degrees each time through the loop. The 21.176 degrees is simply 360deg/17.

3. By changing r each time, one can easily generate cams or spirals.

EXAMPLE 2

The object is to cut a cam. The radius of the cam is 1 inch at 0 degrees going to 0.9 inches at 180 degrees.

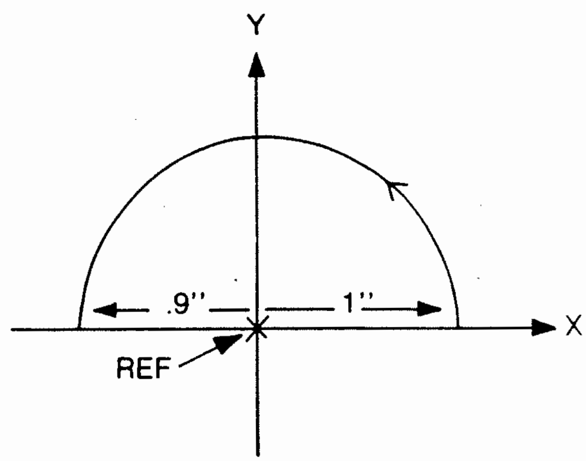
Therefore the radius is reduced by 0.1 inch over 180 degrees. We have to decide on the decrement for r. Fix it at 0.0004 and find the angle.

No. of steps is (.1)/(0.0004) = 250
So the angle increment is 180/250 = 0.72 degrees

Thus we take 250 steps increasing "a" by 0.72 degrees and decreasing r by .0004 each time.

The program becomes:

```
000  START INS 01
001  TD = .125
002  FR  XYZ = 10
003  SET UP——► zcxyu
004  GO  X   1.0625
005  GO Z -.050
006  REPEAT  10
007  REPEAT  25
008  GR  r -.0004
009        a  .72
010  REPEAT END
011  REPEAT END
012  Z——► Clear Z
013  END
```



NOTE

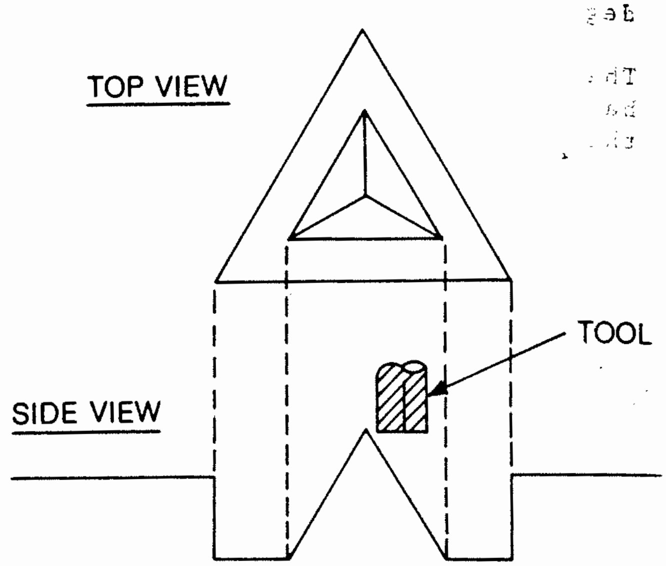
1. We can cycle round to form a spiral.

2. Within the loop one can change Z, for example
   GR Z .001
   So the tool will rise as it cuts the cam or spiral.

EXAMPLE 3

Here are two examples that illustrate the generation of 3D shapes by little vector moves.

1. To generate a Pyramid

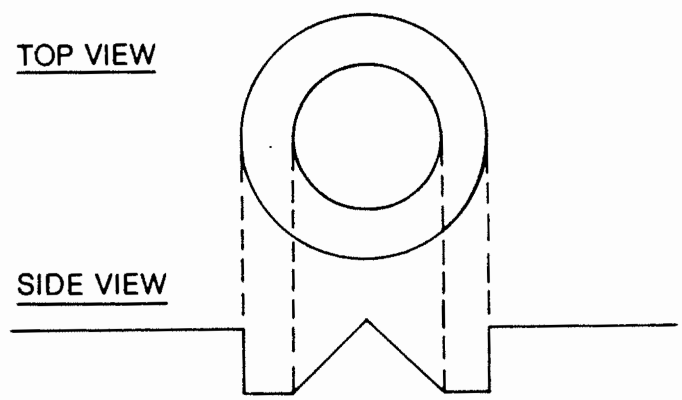


The program becomes:

```
000   START INS 01
001   TD = .125
002   FR   XYZ = 10.0
003   SET UP→zcxyu
004   GO   X    0.135
005        a   -30
006   REPEAT 42
007   GR   Z   -.0005
008   GR   r   .005
009   REPEAT 03
010   GR   r 0.0
011        a 120.0
012   REPEAT END
013   REPEAT END
014   END
```

Note Lines 9-12 generate a triangle. This could be replaced by any n-gon. (See example 1)

2. To generate a Cone



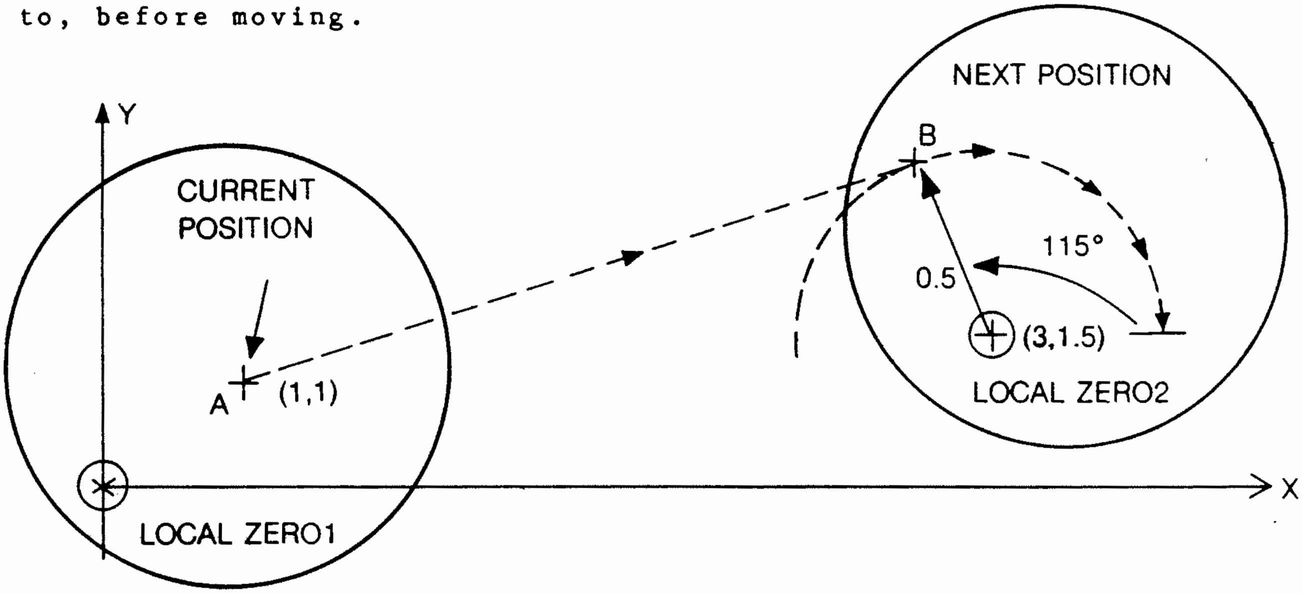
The program is:

```
000   START INS 01
001   TD = .125
002   FR   XYZ = 6.0
003   SET UP→zcxyu
004   GO   X 0.0625
005   GO   Z 0.0
006   REPEAT 42
007   GR   Z   -0.0005
008        X   0.005
009   GO   a   360.00
010   REPEAT END
011   END
```

NOTE: The user should use a ball cutter.

The next example illustrates the flexibility in switching between rectangular coordinates and polar coordinates.

Wherever you are in XY space and wherever you are referenced to, the tool can be moved to another point (expressed in polar or rectangular) provided you create another local zero to reference it to, before moving.



Thus to go from A to B where A is your current position and B is expressed in polar coordinates at local zero 2, we

```
simply ──────────► REF COODS        Insert if Local Zero 2 is ref-
                                    erenced to main reference and
               ZERO AT              not to Local Zero 1.
                   X   3
                   Y   1.5

              GO   r   0.5
                   a   115
```

will move the tool to B

If you wish to swing round on a radius of 0.5

```
               GR  a - 115
```

will swing the tool around clockwise until the radius is parallel to the X axis.

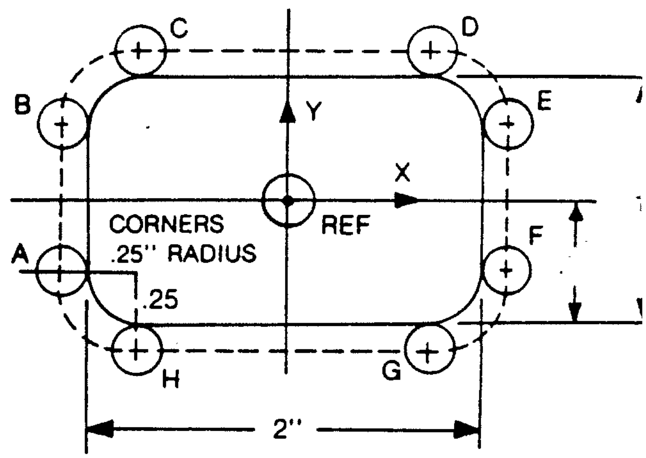
This concept is very powerful in contouring applications as will be seen in the next example.

EXAMPLE 4

The object is to make an outside frame cut with large rounded corners. The tool diameter is specified as 0.125 inch. It illustrates the use of the ZERO AT instruction and GR a (90deg).

The program becomes:

```
000  START INS 01
001  TD = .125
002  FR  XYZ = 10
003  SET UP—►zcxyu
004  GO  X  -1.0625    A
005      Y  -.25
006  GO  Z  -.05       Lower tool .05 into part
007  CHS  Y
008  ZERO AT           Set zero at
009      X   -.75      center of radius
010      Y   .25
011  GR  a -90deg.     Go round arc to C
012  GR  X   1.5       C  to  D
013  —►REF COODS       Switch to ref coods
014  ZERO AT           Set zero at center
015      X   .75       of next arc
016      Y   .25
017  GR  a -90deg.     D  to  E
018  GR  Y  -.5        E  to  F
019  —►REF COODS       Switch to ref coods
020  ZERO AT           Set zero at center
021      X   .75       of next arc
022      Y   -.25
023  GR  a -90deg      F  to  G
024  GR  X -1.5        G  to  H
025  —►REF COODS       switch to ref coods
026  ZERO AT           Set zero at center
027      X   -.75      of next arc
028      Y   -.25
029  GR  a -90deg      H  to  A
030  Z —► C            Raise tool
031  END
```

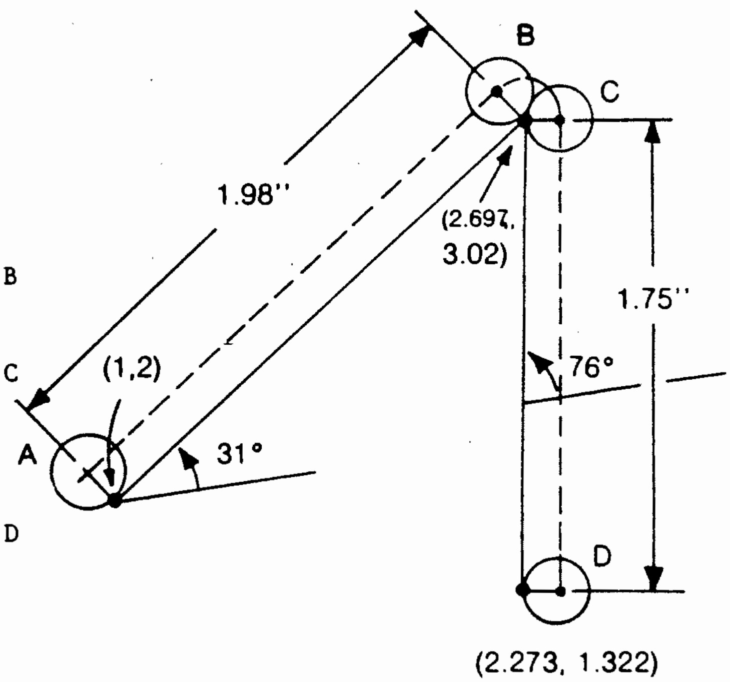


EXAMPLE 5

The object is to go round a sharp angle on an outside cut. If we used XY coordinates for all moves we would need to calculate the tool offsets in XY coordinates. This is very tedious. We can eliminate these calculations by using polar coordinate moves. The tool diameter is .25 inches the sharp angle is opposite.

The program would be:

```
000  START INS 01
001  TD = .25
002  FR   XYZ = 10
003  SET UP→zcxyu
004  ZERO AT
005       X   1
006       Y   2
007  GO   a   31+90=121  Tool is at A
008       r   Tool Radius = .125
009  GO   Z -.050             Drop tool
010  ——→ REF COODS
011  ZERO AT
012       X   2.697
013       Y   3.02
014  GO   r   .125            Tool is at B
015       a    121
016  GR   a -135
017  ——→ REF COODS            Tool is at C
018  ZERO AT
019       X    2.273
020       Y    1.322
021  GO   r    -125
022       a    -14            Tool is at D
023  END
```



NOTE:

1. On line 16, total angle move is 121deg+90deg-76deg=135deg.

2. THE TOOL IS ROTATED ROUND THE POINT OF INTERSECTION.

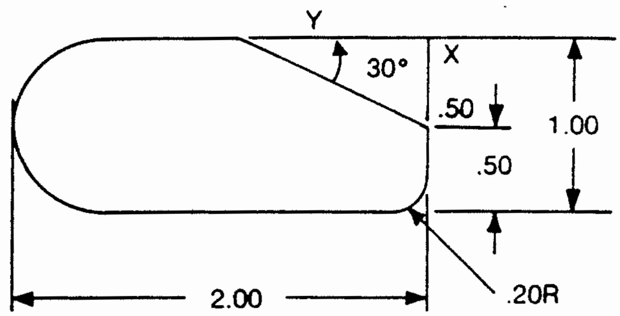
EXAMPLE 6

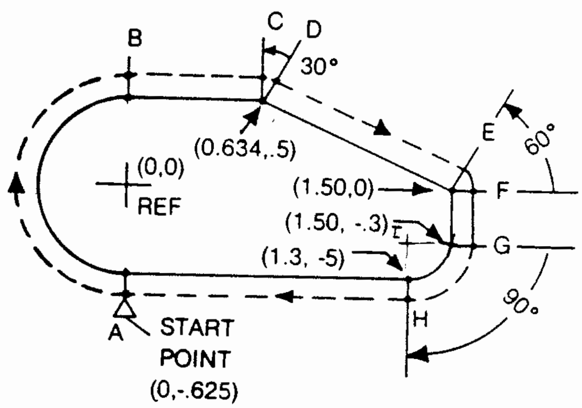
Outside contour this part. The tool diameter is .25 inches, part is .1 inches thick.

Observe that there is NO calculation of tool path in space off the part. Only the starting point is required.

The program becomes:

```
000   START INS 01
001   TD = 0.25
002   FR   XY = 10
003   FR   Z = 4
004   SET UP→zcxyu
005   GO   Y    -.625      (Ref.→A)
006   GO   Z    -.125
007   GR   a    -180       (A→B)
008   ZERO AT
009        X    .634
010        Y    .5
011   GO   r    .125       (B→C)
012        a    90
013   GR   a    -30        (C→D)
014   → REF COODS
015   ZERO AT
016        X    1.50
017        Y    0
018   GO   r    .125       (D→E)
019        a    60
020   GR   a    -60        (E→F)
021   → REF COODS
022   ZERO AT
023        X    1.5
024        Y    -0.3
025   GO   r    .125       (F→G)
026        a    0
027   GR   a    -90        (G→H)
028   GR   X    -1.3       (H→A)
029   END
```





I am not positive this will work until local ref is moved to "I".

EXAMPLE 7

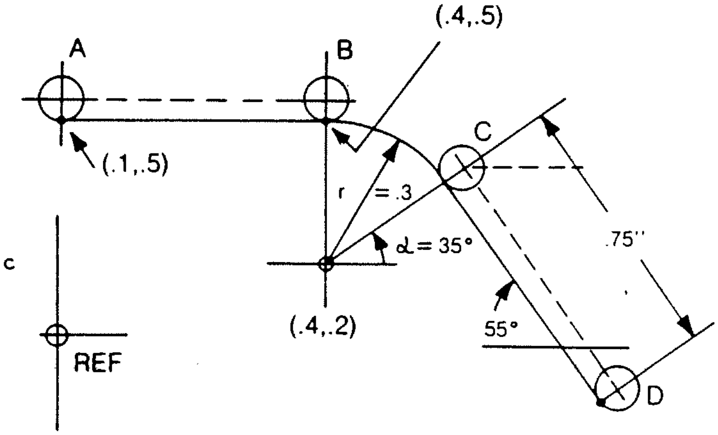
By this time you will realize how easy it is to program very complex shapes with very little calculation. Contouring is rather trivial and the only case you may have problems with is two lines with an arc in between. If the center of the arc is not given.

If the arc center is given then there is NO calculation required for the tool location in space.

Opposite, the object is to move the tool from A to B.

The program is:

```
000  START INS 01
001  TD = .25
002  FR XYZ = 10
003  SET UP→zcxyu
004  GO  X   .1      (Ref.→A)
005      Y   .625
006  GO  X   .4      (A →B)
007  ZERO AT         Zero for arc
008      X   .4
009      Y   .2
010  GR  a   -55     (B →C)
011  ZERO  XY
012  GO  r   .75     (C →D)
013      a   -55
014  END
```



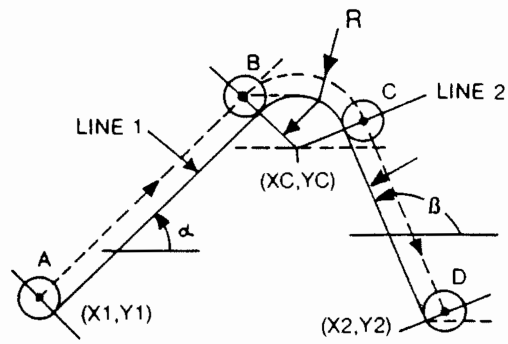
NOTE:

This example is relatively easy because one line (AB) was horizontal. The next example gives the general case with the center of the arc given, but the two lines are at angle & respectively to the X axis.

EXAMPLE 8

We are given two lines with an arc of radius R between them. The tool has to go round the outside.

Line 1 starts at X1, Y1 and has angle
Line 2 ends at X2, Y2 and has angle
We are given XC, YC the center of the arc. This we can do very easily. Suppose the tool radius is r1. Start at A.

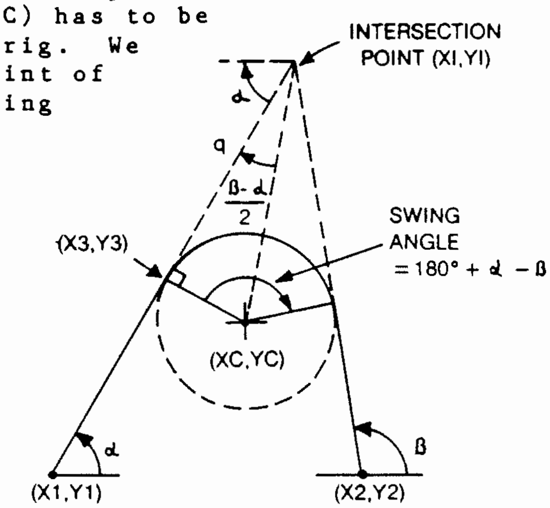



```
006   ZERO AT       (Set local zero at first point)
007        X  X1
008        Y  Y1
009   GO   r    r1  (Tool is moved to A) r = Tool Radius
010        a  α +90
011   GO   Z - .05  (Drop tool)
012   ──► REF COODS
013   ZERO AT       (Reset local zero at center of arc)
014        X  XC
015        Y  YC
016   GO   r   R+r1 (Express B as a polar cood with respect
017        a  α +90  to the center of the circle and move to B)
018   GR   a  -(180 + α - β   )  (Go round the arc, arc angle
019   ──► REF COODS            is -(180 + α - β      )
020   ZERO AT       (Reset local zero at end point)
021        X  X2
022        Y  Y2
023   GO   r    r1  (Express D as a polar cood with respect
024        a (β -90deg).  to the end point and move to D)
```

The critical information is the center of the arc.

EXAMPLE 9 (math note)

This is the next higher level of complexity in going round two lines with an included arc. The intersection point or the two lines is given (XI,YI) but (XC,YC) has to be calculated. It becomes an exercise in trig. We have to calculate (X3,Y3) the tangent point of the line to the arc in route to calculating (XC,YC).



From inspection

q (the length of (XI,YI) to (X3,Y3))

is given by $q = \dfrac{r}{\tan\left(\dfrac{\beta - \alpha}{2}\right)}$

So $X3 = XI + K1 q \cos\alpha$
$Y3 = YI + K2 q \sin\alpha$ where K1 = 1,K2 = 1

In this case K1 and K2 are both negative.
The sign depends on the orientation of the figure.

The rule is $K1 = +1$ if $XI \leqslant X1$
$= -1$ if $XI \geqslant X1$

and $K2 = +1$ if $YI < Y1$
$= -1$ if $YI \geqslant Y1$

So (in this case) $X3 = XI - q \cos\alpha$
$X3 = YI - q \sin\alpha$

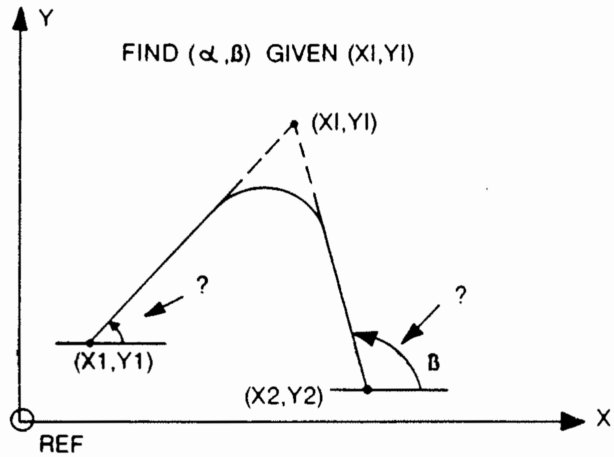
Then $XC = X3 + r \sin\alpha$
$YC = Y3 - r \cos\alpha$

and the swing angle is $(180 + \alpha - \beta)$

EXAMPLE 10

Here is a worked example of the previous calculations. The data is given opposite, r is .25 inches.

First calculate q, the length along the first line to the arc intersection.



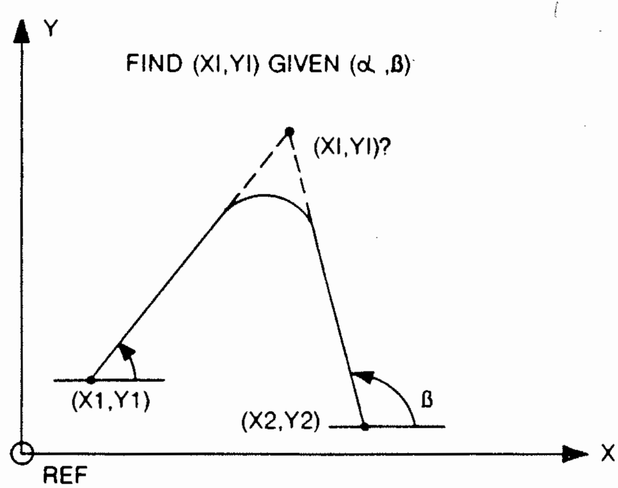
$$q = \frac{25}{\tan\left(\frac{105-45}{2}\right)} = 433$$

So X3 = 1.204 - .433 COS 45 = .898
 Y3 = 1.304 - .433 SIN 45 = .998

Thus XC = .898 + .25 COS45 = 1.075
 YC = .998 - .25 SIN 45 = .821

and swing angle is

180 + 45 - 105 = 120



Note: A perceptive observer will note a redundancy of information. Not only do we have the intersection of the two lines, but we also have their angles. Rarely is this supplied together. The user will have to calculate one or the other if not given. The next math note example shows how.

EXAMPLE 11 (math note)

The angles are simply given as:

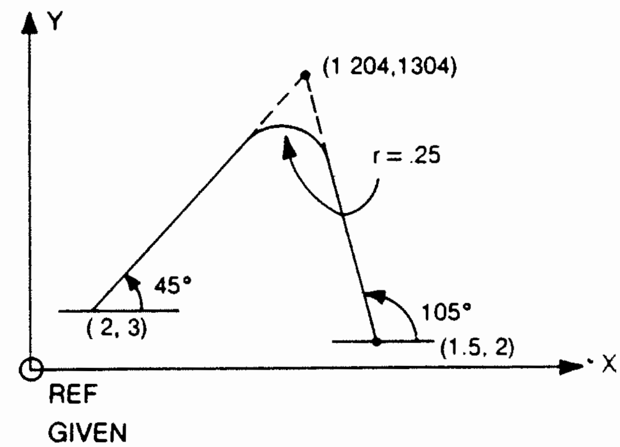
$$\alpha = \tan^{-1} \frac{YI - Y1}{XI - X1}$$

$$\beta = \tan^{-1} \frac{YI - Y2}{XI - Y2}$$

In the previous example:

$$\alpha = \tan^{-1} \frac{1.304 - .3}{1.204 - .2} = \tan^{-1} 1 = 45°$$

$$\beta = \tan^{-1} \frac{1.304 - .3}{1.204 - 1.5} \quad \tan^{-1}(.373) = 105°$$



The intersection point is given as:

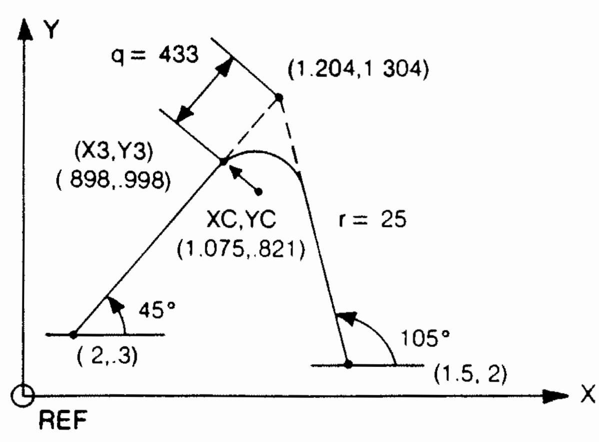
$$XI = \frac{Y2 - Y1 + X1 \tan\alpha - X2 \tan\beta}{\tan\alpha - \tan\beta}$$

$$YI = (XI - X1) \tan + Y1$$

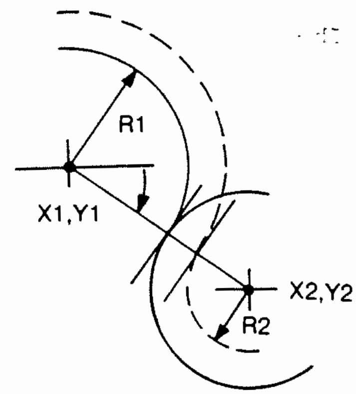
In the previous example:

$$XI = \frac{2 - 3 + 2 \tan 45° - 1.5 \tan 105°}{\tan 45° - \tan 105°}$$

$$YI = (1.204 - 2) \tan 45° + 3$$
$$= 1.304$$



EXAMPLE 12



Intersection of 2 arcs, radii and centers given. This is very easy with polar coordinates. The only piece of information the user has to calculate is the angle i.e. how far to swing round the arc. The radius is easy, either add add or subtract the tool radius to R1 or R2. Simply swing round the first arc, to the intersection point then re-zero at the center of the second arc and swing round the required amount.

$$\left(\alpha = \tan^{-1} \frac{Y2 - Y1}{X2 - X1}\right)$$

and watch the signs.

SCALE

OBTAINED BY: Touching the FUNCTION key and entering 00 to nn. Scale is function 00.

WHAT IT DOES: It is a linear scaling on X,Y,Z or any combination there of, either up or down. It is used as a switch for example.

```
SCALE ON
PROG STATEMENT n
  —
  —
  —
  —
PROG STATEMENT m
SCALE OFF
PROG STATEMENT m+1
  —
  —
  —
```

analogous to SPINDLE ON/OFF. All move statements between scale on and scale off will be scaled.

FORMAT: The exact format is:

TO SET	SCALE ON	
	(X x.xxxx)	( ) Optional
	(Y y.yyyy)	
	(Z z.zzzz)	
TO CLEAR	SCALE OFF (X)(Y)(Z)	Select scale off Axis

The value entered after the axis is the scale factor. If fractional then the moves are scaled down, if greater that 1 then moves are scaled up. Thus

X 0.5 will half all x moves

X 2.0 will double all x moves

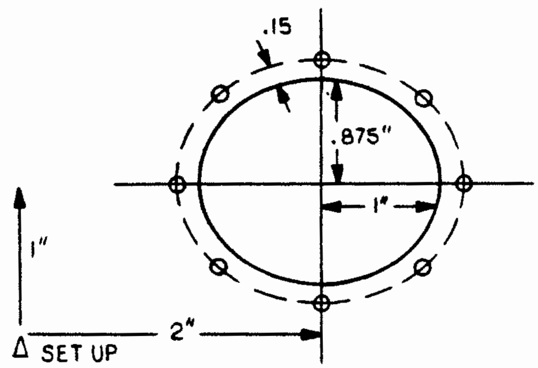
Automatically any (r,a) moves will be scaled on r ($r=\sqrt{X^2 + Y^2}$) but not on a (they remain the same).

RESTRICTIONS

One can only scale down from maximum axis move or scale up to the maximum axis move. For example, any scaled up move that results in X exceeding 6.2 inches will be flagged as an error and any entry that exceeds 6.2 inches to be scaled down will be flagged.

EXAMPLE

Make an ellipsoidal pocket with ellipsoidal bolt hole pattern round the outside. Major axis is 2 inch, minor axis is 1.75 inches, depth is .1 inch and tool diameter is .125.



What we do is to write a program for a circle pocket + bolt hole circle, in between a scale down of Y of .875.

```
000  START INS  22
001  TD = .125
002  FR  XYZ = 10.0
003  SET UP→zcxyu
004  ZERO→ AT
005       X = 2.0
006       Y = 1.0
007  SCALE ON
008       Y  .875
009  CIRC  Z% 050
010  XY  CUT % 050
011  ZH  0.0
012  Zd  .1
013  XC  0.0
014  YC  0.0
015  r1  1.0
016  r2  0.0
017  BOLT PECK  01
018  ZH  0.0
019  XC  0.0
020  YC  0.0
021  a1  0.0
022  N   8
023  r  1.15
024  SCALE OFF Y
025  END
```

NOTE: The user should be aware of scale round off. Thus 2.4576 scaled down by 10 results in 0.2457, which when scaled up by 10, results in 2.4570.

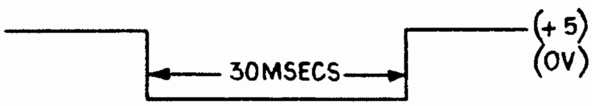
Control 2/Control 3

How Obtained From `CONTROL n = 2 or 3`

What it Does These are for external event operation or synchronization.

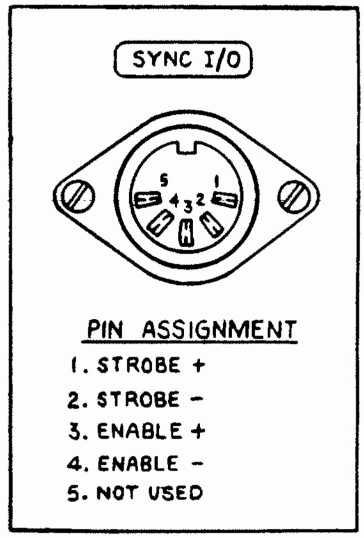
Control 2 sends an active low pulse out of length 100 msecs. and continues on in the program.

Control 3 expects to receive a low signal. (The high value can range from 5 to 24v as it is internally current limited) The program will wait at control 3 until it sees a low signal, then it will continue. The signal must remain low for at least 30 msecs. It is sampled 3 times at 10 msecs intervals.



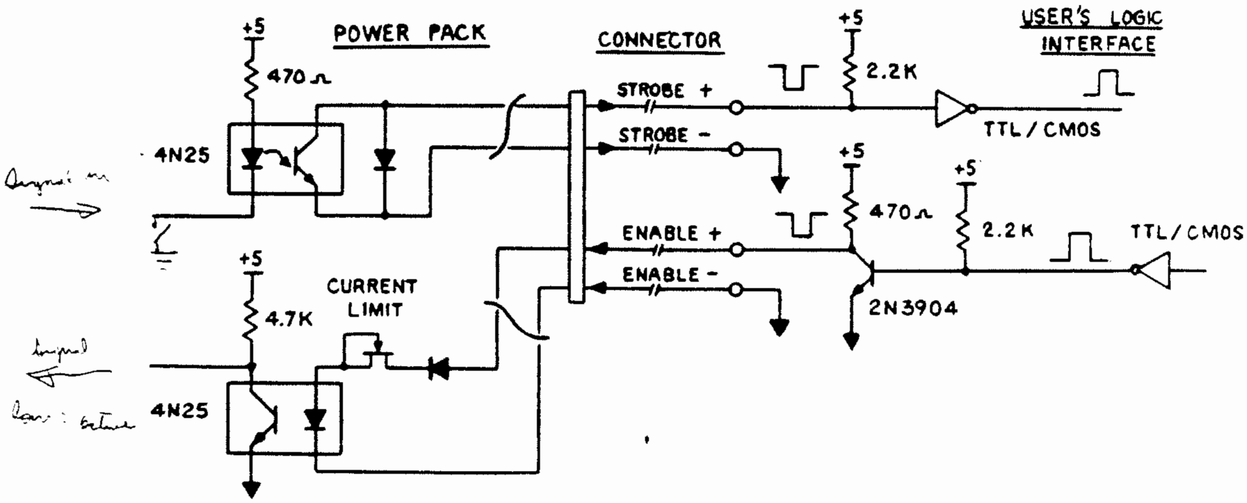
How Interfaced

At the right hand side of the power pack the user will find this connector. It is a standard female miniature audio one with 5 pins from Switchcraft.



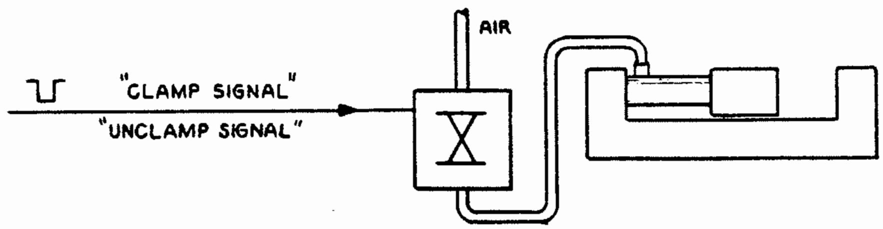
The cable length should not exceed 15 feet.

Internally we have the following optocoupled circuits.



HOW USED EXAMPLES

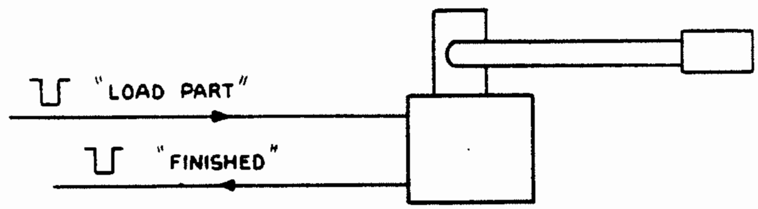
1. AUTO CLAMPING



NOTE: The signal can toggle a flip flop

so the first time the signal is sent it is "clamp" and the second time it is "unclamp". & THIRD = ROBOT ON LOAD ETC

2. AUTO LOADING

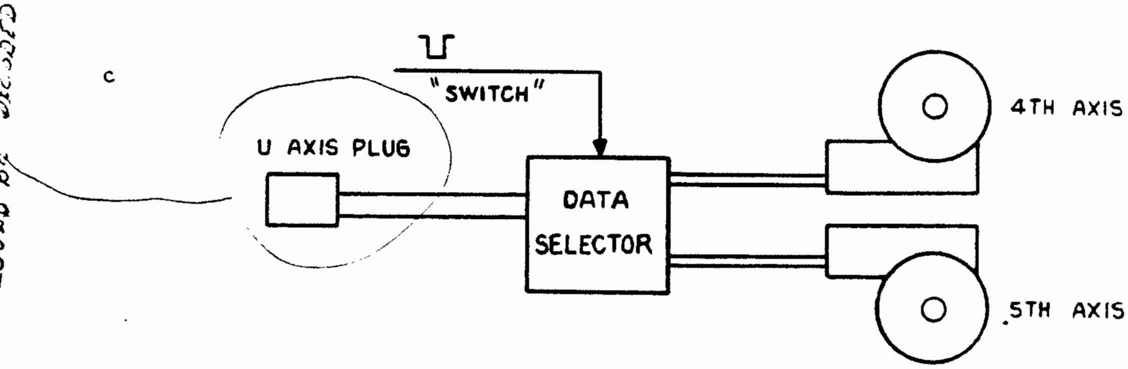


"U" AXIS SIGNALS COULD BE DECODED FOR ROBOT SYNCHRONIZATION ETC.

3. FIFTH AXIS

This is slightly more complex but the signal can be used to toggle a data selector between 2 rotary tables.

c

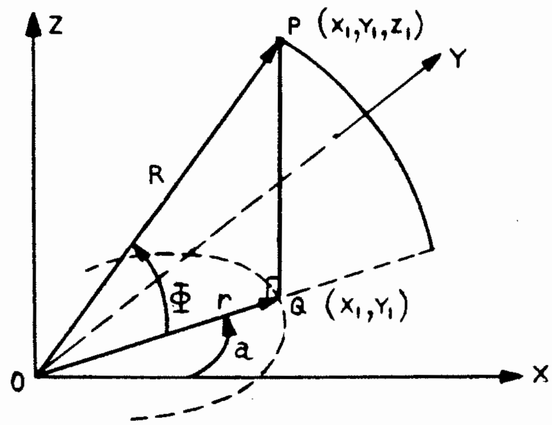


COORDINATE Φ

OBTAINED BY Φ key

WHAT IT DOES:

It allows the user to generate circular arcs at right angles to the XY plane. A point P in space (see opposite) is given by coordinates (X1,Y1,Z1). It is at a height Z1, vertically above the point Q (X1, Y1) on the XY plane. The angle QOP is Φ. If we keep OP (=R) fixed in length, and vary Φ, then P will describe a circular arc in the plane QOP at right angles to the XY plane. The point Q is of course (r,a) in polar coordinates. If we vary a (0 to 360) P will trace out a circle (a height Z1) above the XY plane of radius r. (R,a,Φ) are called spherical coordinates of the point P. By keeping R fixed and varying a and Φ, P will generate a sphere.



NOTE: If P has coordinates (X,Y,Z) then the relationship to a and o is:

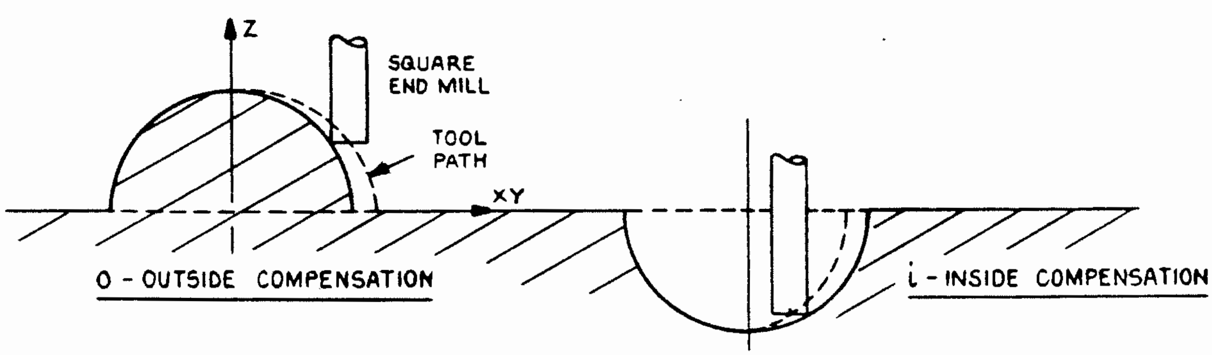
$$X = R \cos \Phi \cos a \qquad \text{where} \qquad R = \sqrt{X^2 + Y^2 + Z^2}$$
$$Y = R \cos \Phi \sin a \qquad\qquad a = \tan^{-1} Y/X$$
$$Z = R \sin a$$

QUALIFIERS ON Φ

The usual qualifiers apply to the Φ move:

f........fast
c........comeback

However i (inside) and o (outside) are different. Built in is automatic compensation for a square end mill.



For outside compensation the tool path is:

$$X = R \cos \Phi \cos a + \frac{Td}{2} \cos a$$

$$Y = R \cos \Phi \sin a + \frac{Td}{2} \sin a$$

$$Z = R \sin \Phi$$

For Inside compensation the tool path is:

$$X = R \cos \Phi \cos a - \frac{Td}{2} \cos a$$

$$Y = R \cos \Phi \sin a - \frac{Td}{2} \sin a$$

$$Z = R \sin \Phi$$

Where Td is tool diameter.

For a <u>ball end</u> mill the user must compensate for the starting position of the radius of the ball cutter.

<u>FORMAT</u>

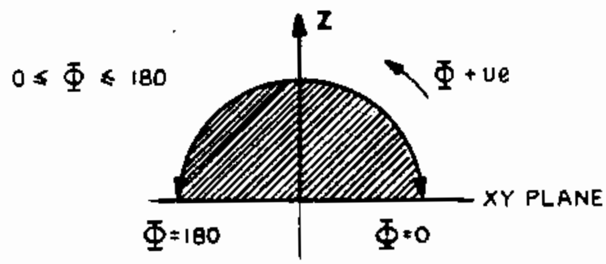
The instructions are

GO Φ + ddd.ddd	Go to this absolute angle Φ degrees
GR Φ + ddd.ddd	Go relative this angle amount on Φ

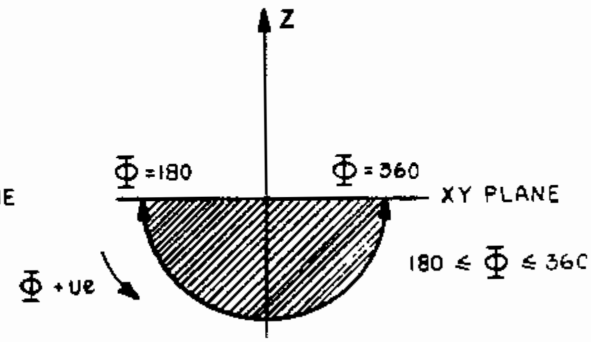
There is only one major restriction on Φ. The Φ move can only operate in one of the following two range

$$0^\circ \leq \Phi \leq 180^\circ \quad \text{AND} \quad 180^\circ \leq \Phi \leq 360^\circ$$

corresponding to Φ operation ABOVE the XY plane and Φ operations BELOW the XY plane. This is to allow different tool compensation for above and below the plane.



Φ <u>range above the XY plane</u>



Φ <u>range below the XY plane</u>

To go from $\Phi = 90^\circ$ to $\Phi = 270^\circ$, the user must make two Φ moves, either $\Phi \rightarrow 0^\circ$ or $\Phi \rightarrow 180^\circ$ then $\Phi \rightarrow 270^\circ$ depending whether the move is clockwise or counter clockwise. One cannot do this in one move.

HOW TO USE Φ

1. The user must first decide on a ball end mill cutter or a flat end mill cutter. If a flat end mill cutter, then all moves on Φ must be qualified by inside or outside.

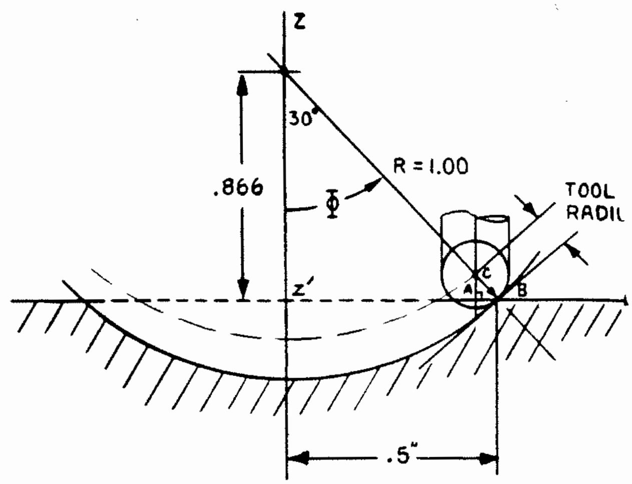
2. The tool must be positioned in space correctly. Examples are given, on the following pages.

3. The zero around which Φ operates then must be set correctly, to correspond with the desired radius and angle.

4. Then the moves are generated with a combination of Φ, x, y or (r,a) to cut the required surface.

EXAMPLES IN INITIAL POSITIONING

1. BALL CUTTER



Suppose we wished to cut a 30 down hemisphere with a ball cutter. The radius R of the hemisphere is 1 inch. The ball radius is .125 inches. The zero point for Z is

$R \cos \Phi = 1 * \cos 30$
$= .866$

This is the point around which Φ moves.

Z'B = R sin Φ = 1* sin 30 = 0.5" is the surface radius of the hemisphere. As can be seen we have to raise the tool by the amount AC and shorten the radius by AB in order to position the ball end mill tangentially to the surface of the hemisphere.

Hence AC = BC cos 30 = .0541
AB = BC sin 30 = .03125 (BC = .125)

So the correct initial move is

```
GO Z  .0541
GO r  .4687      (.5 - .03125)
   a  0
```

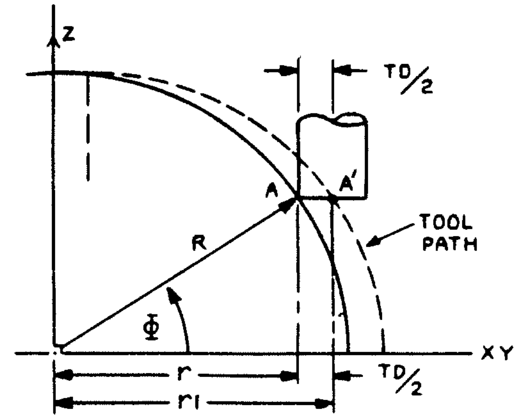
NOTE: This is correct if the ball cutter is zereod correctly. Remember that the ball center is not zeroed at the surface, it is TD/2 up Z. The way to do it is as follows:

SET UP →	zcxyu	ZERO BALL TIP ON SURFACE
GO Z	TD/2	BALL DIA/2
HALT		LOWER SPINDLE SO BALL TOUGHES SURFACE

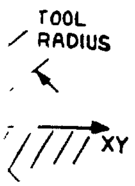
This corrects for the zero point. Make sure that the clearance height is greater than TD/2.

The following examples have assumed that the ball cutter is zereod correctly.

2. SQUARE END MILL



OUTSIDE COMPENSATION



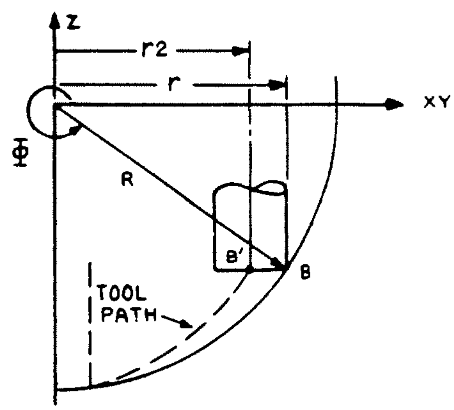
The user must position the square end mill such that the edge of the tool touches the hemisphere at the required radius. The controller will then move the tool as indicated i.e. compensating for the tool offset as a function of the Φ angle.

Suppose $\Phi = 40$, R = 1, TD = .125
Then $Z = R \sin 40^\circ = 0.6428$
$r = R \cos 40^\circ = 0.7660$

So the tool center is positioned at (Z, r1)

$Z = 0.6428$
$\underline{R1} = 0.766 + \frac{TD}{2} = \underline{0.8285}$ (Add on the tool radius)

INSIDE COMPENSATION



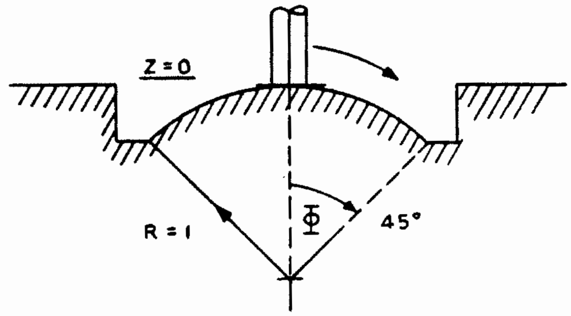
The center of the tool must be positioned at B', on the inside of the hemisphere
Suppose $\Phi = 320$, R = 1, TD = .125
Then $Z = 1 * \sin(-40) = -.6428$
$r = 1 * \cos(-40) = .7660$
So $r2 = .7660 - \frac{TD}{2} = .7035$

So the tool is positioned at (Z, r2) = (-.6428, .7035)

Thus we subtract off the tool radius.

EXAMPLE 1 SQUARE END MILL

Mill an up 45 hemisphere ball.
R = 1 inch, TD = .125 inches
With a square end mill and with outside compensation.



A) Starting at the Top

```
000   START INS 01
001   TD = 0.125
002   FR XY = 16
003   FR Z  = 10
004   SETUP ——► zcxyu
005   SPINDLE ON
006   GO  Z  0
007   ZERO AT
008         Z  -1              Set center 1 inch below surface
```

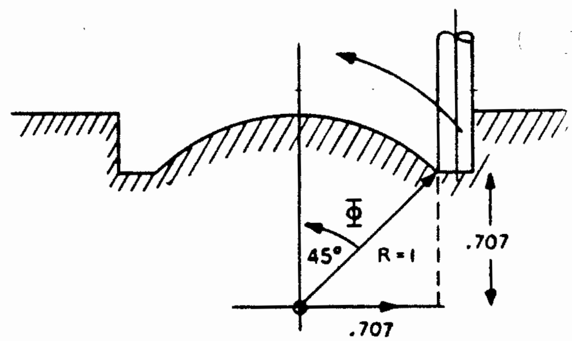
```
+------------------------------+
| 009   REPEAT  90             |
| 010      GR  0  Φ   -0.5     |   Move Φ outside 1/2 degrees
| 011      GR     a    360     |   Cut circle
| 012   REPEAT END             |
+------------------------------+
```

```
013   SPINDLE OFF
014   Z ——► C
015   END
```

B. Starting at the Bottom

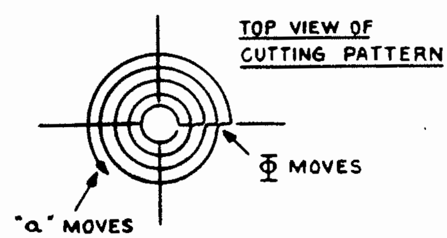
Replace above box by

```
009   GO   r   0.7695    (.707 + .0625)
010   GO   Z   0.707
011   REPEAT 90
012   GR  0  Φ  0.5
013   GR   a   360
014   REPEAT END
```

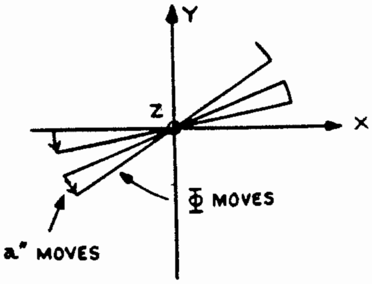


In these examples the cutting pattern is small Φ moves followed by a large circular "a" move as opposite. We can easily switch this to a large Φ move followed by a small "a" move.

The cutting pattern is shown opposite. We arc the tool over, move round slightly then arc back.



TOP VIEW OF CUTTING PATTERN



C) ARCING BACK AND FORTH (GO/GR Φ)

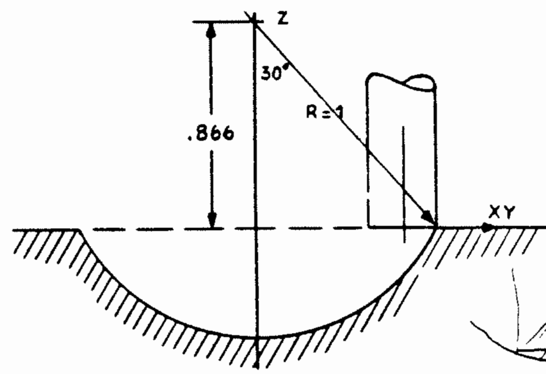
The box is replaced by:—

```
009  REPEAT 90
010            GO r 0.7695 } Add this juct to make sure
011            GO Z 0.707  }                 OR
012  GO   0 Φ 135     Swing Φ across        GR o Φ 90
013  GR     a  2      Advance a 2 degrees
014  GO   0 Φ  45     Swing Φ back          GR o Φ -90
015  GR     a  2      Advance a 2 degrees
016 REPEAT END
```

NOTE: This way is much slower than A or B. There is a good deal of calculation involved in continuous Φ moves with compensation, so it is not recommended.

EXAMPLE 2 (Square End Mill)

Mill a down 30° hamisphere
R = 1 inch, TD = .125 inches.




```
000  START INS 01
001  TD = 0.125
002  FR XY = 16
003  FR  Z = 10
004  SETUP→zcxyu
005  SPINDLE ON
006  GO   Z   0
007  ZERO AT
008       Z   0.866
009  GO r 0.4375  (0.5 - 0.0625)
010     a 0
```

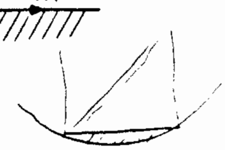
CIRCLE PATTERN

```
011  REPEAT 60
012  GR i Φ -0.5
013  GR a    360
014  REPEAT END
```

ARCING PATTERN

```
011  REPEAT 90
012  GO  r 0.4375
013  GO  Z 0.866
014  GO  i Φ 240
015  GR    a 2
016  GO  i Φ 300
017  GR    a 2
018  REPEAT END
```

```
015  SPINDLE OFF
016  Z ——→ C
017  END
```

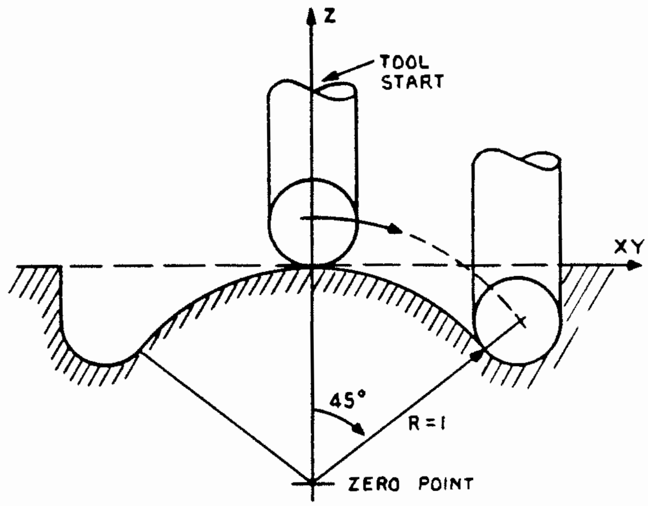


you will always have a flat spot like this w/ a sq end mill however if the radius of cut is large this will be almost unnoticeable

EXAMPLE 3

The up 45 hemisphere

The center of the hemisphere is 1 inch below the surface. The program becomes:



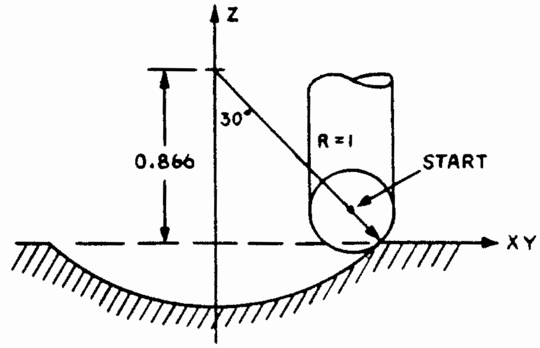

```
000  START INS 01
001  TD = 0.125 (ball cutter)
002  FR  XY = 16
003  FR   Z = 10
004  SET UP——▶ zcxyu
005  SPINDLE ON
006  GO  Z  0.0
007  ZERO AT
008       Z -1
009  Repeat 90
010  GR  f Φ -0.5
011  GR  a  360
012  REPEAT END
013  Z ——▶ C
014  SPINDLE OFF
015  END
```

The critical program section is in the above rectangle. We move O in 1/2 degree decrements that trace out a circle in the ZX plane. After each O move, we rotate "a" to generate a circle in the XY plane.

EXAMPLE 4

The down 30 hemisphere with ball end mill.

The radiuss R is 1 inch, the tool diameter is .125.



The program becomes:

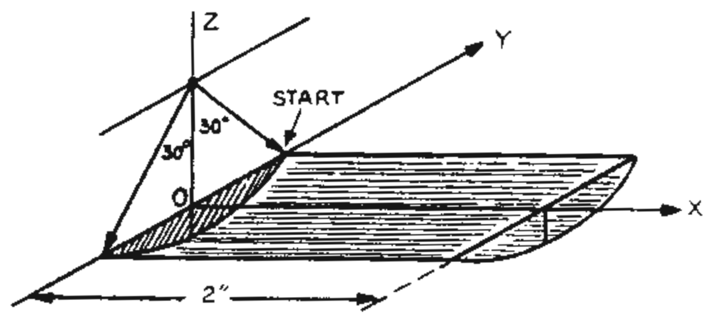
```
000   START INS 01
001   TD = 0.125
002   FR XY = 16
003   FR  Z = 10
004   SET UP ──► zcxyu
005   SPINDLE ON
006   GO Z .0541        }  INITIAL POSITION OF TOOL
007   GO r .4687        }
008      a  0
009   ZERO AT           }  SET CENTER OF SPHERE
010      Z = 0.866      }
011   REPEAT 60            MOVE Φ 30
012   GR f Φ -0.5
013   GR a  360            CUT CIRCLE ON XY PLANE
014   REPEAT END
015   Z ──► C
016   SPINDLE OFF
017   END
```

EXAMPLE 5

Cut a circular slot along the X axis.

R = 1 inch, ball end mill diameter .125 inches.

The simplest way to do this is to move the tool back and forth along the X axis while moving o in small increments.



The key section becomes:-

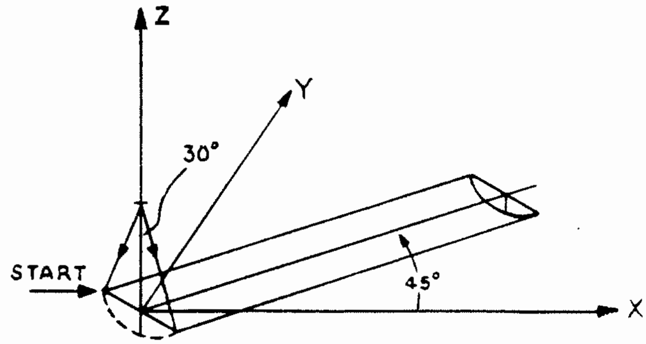
```
GO Z 0.0541   }  POSITION TOOL OUT ALONG Y AXIS
GO r 0.4687   }
   a 90
ZERO AT       }  FIX CENTER HEIGHT
   Z .866     }

REPEAT 3
REPEAT 40
GR f Φ -0.5      MOVE Φ 120 TIMES X 0.5 DEGREES = 60 DEGREES
GR c X  2.0      MOVE X ALONG 2 INCHES AND COME BACK
REPEAT END
REPEAT END
```

Note: We can also do this by incrementing along X, re-zeroing X then swinging Φ from 240 degrees to 300 degrees.

EXAMPLE 6

With a .125 ball end mill, cut a 30 degree circular slot at 45 degrees to the X axis.

Again the simplest way to do this is to move the tool back and forth along the slot, then to move Φ in small increments.

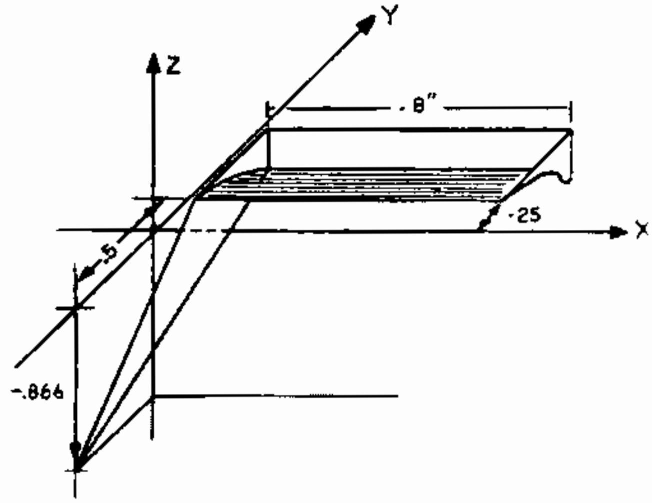


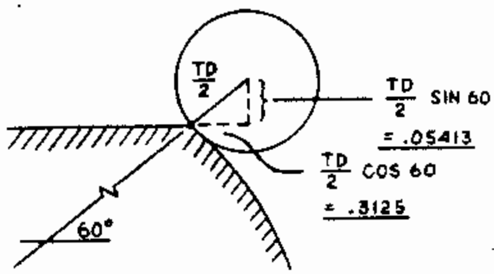

```
GO   Z   0.0541   }   POSITION TOOL
GO   r   0.4687
     a   135
ZERO AT           }   FIX CENTER HEIGHT
     Z   .866

REPEAT 3
REPEAT 40
GR f Φ -0.5
GR    X   1.4142      GO OUT AT 45° TO X
      Y   1.4142
GR    X  -1.4142      COME BACK
      Y  -1.4142
REPEAT END
REPEAT END
```

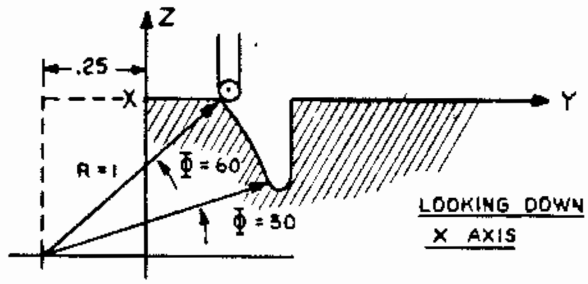
EXAMPLE 7

This is an arc frame on the YZ plane with a ball end mill diameter of .125 inches. R=1

We have to position the tool correctly at the start of the arc.







First we position the tool

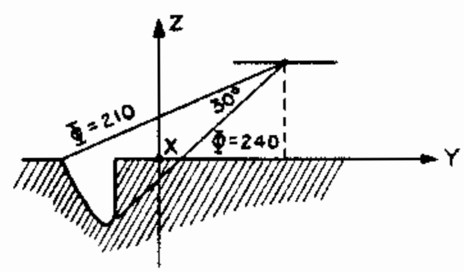
```
Go Y .2813   (.25 + .03125)
GO Z .0541
```

Next we zero the axis (about which Φ moves)

```
ZERO AT
Z -.866
Y -.25
```

Next we generate the moves

```
     REPEAT 60
       GR f Φ -0.5
       GR c X  0.8
REPEAT END
```



Similar arc frames can be generated with Φ above the XY plane.

THE 4TH AXIS (U AXIS)

This is the optional rotary table which plugs into the power section at the back. It is the U axis.

MECHANICAL

The rotary table has a 6 inch diameter table, can be mounted horizontally or vertically, and has a maximum variation of +1 minute 20 seconds (+ .02 degrees).

Each step of the stepper corresponds to .004 degrees, or in other words the circular resolution is 1 part in 90,000. The maximum rotation speed is 10 and 20 degrees per second for the 2200 and 2400, respectively.

BACKLASH

This is almost negligible. The worm, worm gear spacing is set approximately by the cam actions of the worm bearing assembly, then fine-tuned by a set screw. This should not be tampered with, as it has been set very carefully.

If the user is concerned by this error, all moves should be one direction, any back moves should swing past the desired value then forward again. There is no electronic measurement of the backlash.

ELECTRICAL

The drive is pre-set for a specific machine, therefore a rotary table designated for the DM 2000/2200 should not be plugged into a DM 2400 or vice versa. They are not machine interchageable.

DIRECTION

The table moves clockwise for a positive U move, so the tool appears to move counter-clockwise.

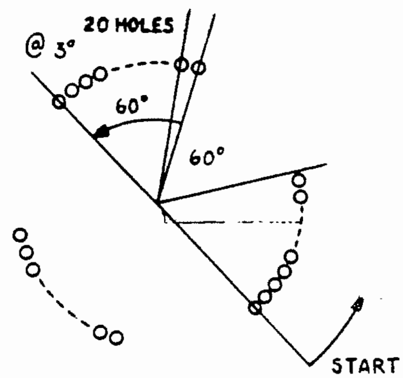
SOFTWARE

The commands for U are:

```
1. GO  (f) U        0 to +/- 999.999 degrees
       (c)
2. GR  (f) U        0 to +/- 999.999 degrees
       (c)
3. ZERO U           Zero U at this point.
4. SETUP U          SETUP reference point for U
5. DISPLAY U        Display U value.
6. CS U             Change XY sign on U and go there
7. FEEDRATE U       Set U feedrate
8. END              Rotate  U back to REF 0 point on  any  END
                    statement.
```

EXAMPLE

Drill nozzle hole patterns




```
000     START INS 11
002     FR XYU = 16
003     FR Z = 5
004     SET UP──►zcxyu
005     REPEAT 3
006     REPEAT 20
007     GO Z -.1   } DRILL HOLE
008     Z──►C
009     GR f U  3.0   EVERY 3 DEGREES
010     REPEAT END
011     GR f U 60     JUMP 60 FOR NEXT BLOCK
012     REPEAT END
013     END
```

RS232-C INTERFACE

WHAT IT DOES

This allows the user to interface to the controller with an external computer or peripheral communication link that can be up to 50 feet away.

The user may: DOWNLOAD a program from the computer to the controller, then execute it

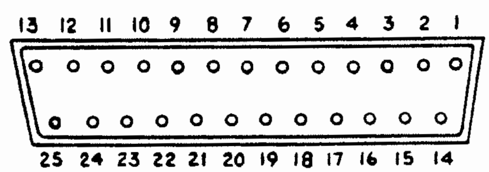
UPLOAD a program from the controller to the computer.

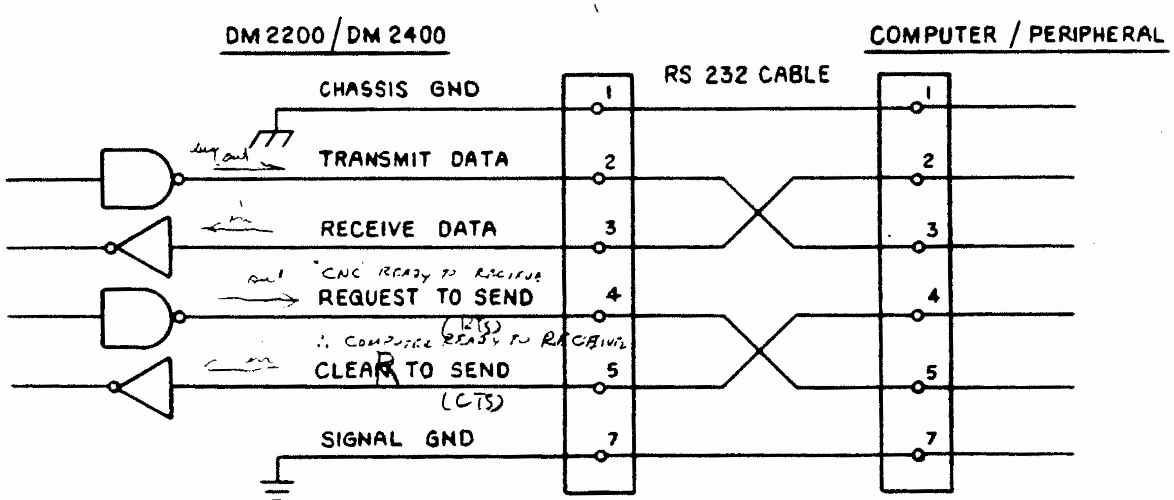
INCREMENTALLY operate the machine (line by line download and execute) where the computer generates the instructions very much like an X,Y plotter. There is no limit to the number of instructions that can be sent.

The desktop unit also has the RS232-C interface so the user can download or upload to the controller off line. This link allows the user much greater flexibility in part generation, from font style libraries for inscribing to complex three dimensional shapes via CAD software.

PHYSICAL LINK

At the top right hand side of the machine there is the standard RS232-C connector. Each pin is numbered from 1 to 25. There will be a similar connector at the back of the computer. The general connections are as follows:





Depending on the serial interface, some computers do not have REQUEST TO SEND and CLEAR TO SEND. In this case, these pins should be jumped together at the DM2400 in the connector. The cabling for RS232-C is readily available in a variety of lengths. Do not exceed 50 feet.

LINK PARAMETERS

Data is transmitted and received asynchronously with the following parameters

Data Rate (speed)	=	2400 bits per second (bps)
Parity	=	none
Data Size	=	8 bits
Number of Stop Bits in standard ascii.	=	1

Appendix 2 contains the ASCII table.

RESTRICTIONS

In upload and download.

0 is changed to Q	ascii (51) hex
→ is changed to >	ascii (3E) hex

Any programs written at the computer should use this nomenclature. Some systems support these symbols others don't, so we have defaulted to Q and >.

SETTING UP THE 2400

With the RS232-C cable linking the 2400 with the computer, one can switch on. The user then selects from

[LINE MODE] [READ/WRITE]

what he wishes to do. There are four possible operations (see flow chart). They are

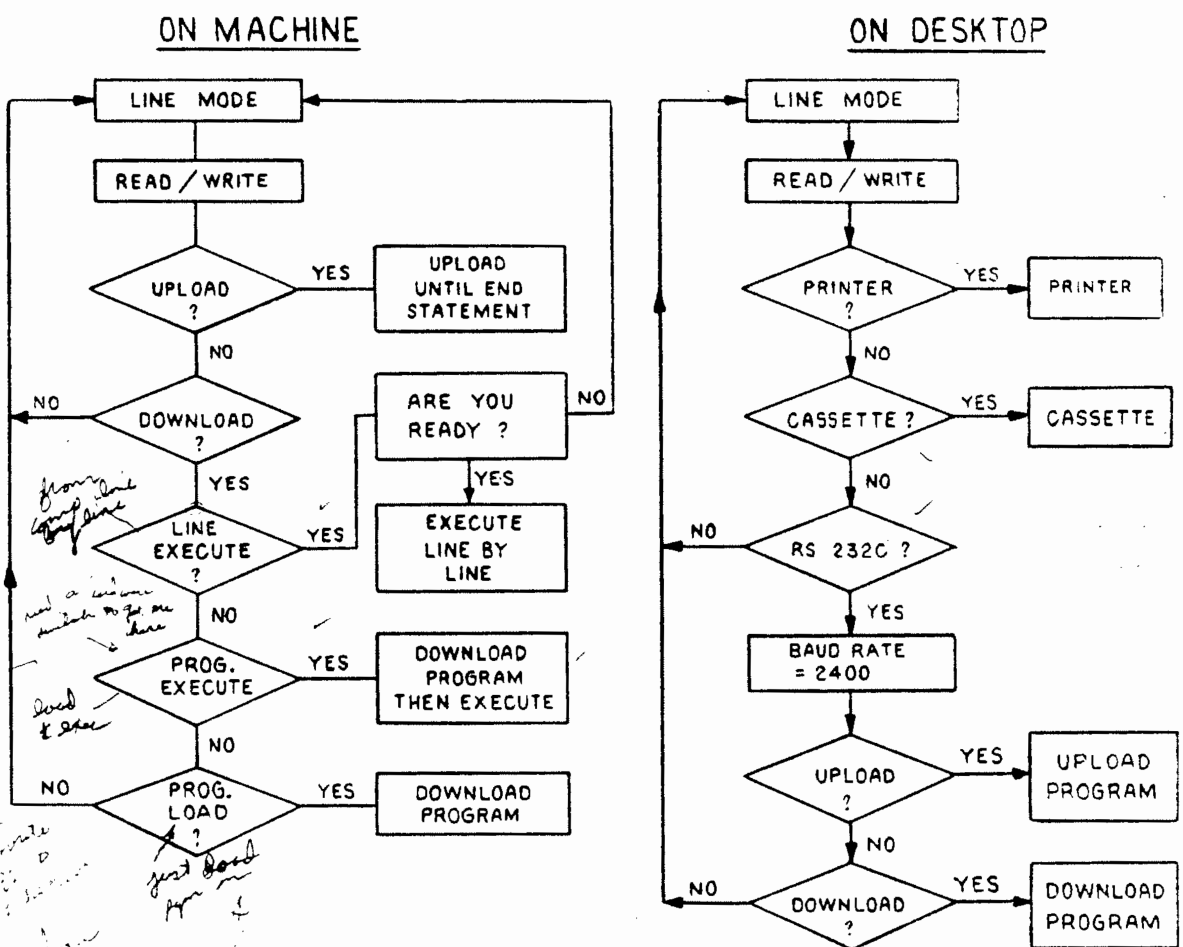
(1) UPLOAD This will transmit the program at the current line number to the computer until an end statement is encountered. This function must be restarted for subroutines as they are treated as programs. Upload therefore provides a means to save a program in the computer memory or provide a printout on a standard RS232 compatible printer.

(2) LINE EXECUTE This will receive line by line instructions from the computer and execute them directly. This sub mode is for computer generated instructions or programs exceeding the 900 lines of the controller. Subroutines, looping and skip are not supported. It is the function of the computer to handle these program control statements independently.

(3) PROG EXECUTE This will receive the program from the computer (download) until end of file is detected then automatically go to the start and begin execution. It should not exceed 900 lines.

(4) DOWNLOAD This is as above without the automatic begin. The controller can therefore be downloaded with a variety of programs which the user can select to run.

At the desktop unit just UPLOAD and DOWNLOAD modes are supported.



start basic like this: BASICA/C:4096

NOTE: as written this software will result in a communication buffer overflow unless you increase the size of the communication buffer when you start basic or incorporate handshaking statements

```
010  REM Dyna Myte Controller / IBM PC Communications Package
011  REM
020  Width 80
030  OPEN "COM1:2400,N,8,1,CS65000,DS0,CD0" AS #1
100  CLS
110  PRINT;PRINT;PRINT
120  PRINT "   1 : Upload (from Controller to IBM PC)"
130  PRINT "   2 : Download (from IBM PC to Controller)"
140  PRINT "   3 : Incremental "
150  PRINT "   4 : External "
163  PRINT "   5 : Exit
165  LOCATE 12,24,1
166  PRINT "
167  Locate 12,1,1
170  INPUT "Enter your selection = ",N$
180  IF VAL (N$)=0 OR VAL (N$)>5 GOTO 165
182  IF VAL (N$)=5 THEN END
200  INPUT "Enter file name = ", FILE$
320  ON VAL (N$) GOSUB 200,400,3000,4000
330  GOTO 100
400  REM
401  REM Download Command
402  REM
403  OPEN FILE$ FOR INPUT AS #2
405  PRINT: PRINT: INPUT "     Hit return key, when ready",B$
410  IF EOF (2) GOTO 710
500  LINE INPUT #2, A$
600  PRINT #1, A$; CHR$(10); CHR$(13)
650  PRINT A$
700  GOTO 410
710  PRINT #1, CHR$(26); CHR$(13);' End-of-file
720  CLOSE #2
1000 RETURN
2000 REM
2001 REM Upload Command
2002 REM
2003 OPEN FILE$ FOR OUTPUT AS #2
2050 PRINT: PRINT: INPUT "     Hit return key, when ready", B$
2100 IF LOC(1) < 1 THEN GOTO 2100'Wait for character
2200 A$=INPUT$(LOC(1),#1)
2300 FOR I=1 TO LEN(A$)
2400 IF MID$(A$,I,1)=CHR$(26)  THEN GOTO 2800' IF EOF THEN done
2500 NEXT I
2620 FOR I=1 TO LEN(A$)
2630 IF  MID$(A$,I,1)=CHR$(10) THEN GOTO  2640'FIlter  out  line
     feeds
2632 PRINT MID$(A$,I,1);
2634 PRINT #2, MID$(A$,I,1);
2637 IF MID$(A$,I,1)=CHR$(13) THEN PRINT #2, CHR$(10);'Add line
     feed now
2640 NEXT I
2700 GOTO 2100
```

```
2800 PRINT #2, A$;
2850 PRINT A$
2860 CLOSE #2
2900 RETURN
3000 REM
3001 REM Incremental Command
3002 REM
3003 OPEN FILE$ FOR INPUT AS #2
3010 PRINT: PRINT: INPUT "Hit return key, when ready", B$
3050 IF LOC(1)=0 THEN GOTO 3100
3060 A$=INPUT$(LOC(1),#1)
3100 IF EOF (2) THEN GOTO 3800
3200 LINE INPUT#2,A$
3250 PRINT A$
3300 FOR I=1 TO 200:NEXT I: PRINT #1,A$
3400 IF LOC(1) <9 THEN GOTO 3400
3410 A$=LEFT$(INPUT$(LOC(1),#1),8)
3500 IF MID$(A$,1,2)="NO" THEN GOTO 3050
3600 IF A$= "Error 01"THEN GOTO 3900
3700 IF A$= "Error 02" THEN GOTO 3910
3800 PRINT "Program finish":  CLOSE #2:  RETURN
3900 PRINT: PRINT: PRINT "Over limit": CLOSE #2: RETURN
3910 PRINT: PRINT: PRINT "Parameter error": CLOSE #2: RETURN
4000 REM
4001 REM External Command
4002 REM
4003 GOTO 400
```

?

3920 PRINT: PRINT: PRINT "IMPROPER SUBROUTINE, REPEAT, SKIP ETC
: CLOSE #2 : RETURN

3750 if A$ = "ERROR_03" THEN GO TO 3920

SECTION 12
USER NOTES
and
INFORMATION

USERS NOTES AND INFORMATION SECTION

1. INS/MM?

When the machine is switched on, the controller automatically defaults to mm unless the controller reads a PROGRAM START INS statement at location 000.

If there is no program start instruction at this address, the controller is in mm until the user passes through with the NEXT key, or line number entry, a program start instruction in inches. That is, the controller tries to make as intelligent a decision as possible as to the ins or mm setting for each program, either when running or entering information. The user can instantly determine the setting by looking at the number of digits behind the decimal place, 3 for mm 4 for inches.

However, it is possible to fool the controller. One example is on switch on to have location 000 blank, then to jump through line mode to the middle of a program or subroutine and start entering data in inches which will be taken by mistake as mm. Because the controller has not received any contrary information, the result will be a scaling down by 25.4 when the program is run.

2. HOW DO I KEEP THE SET UP WHEN SWITCHED OFF?

The set up coordinates are not stored in the program and are cleared on each controller switch on. If the user wishes to store them for example on a production run, with a fixed jig for parts then it is very easy to do this. So when the machine is switched on, it will go automatically to the set up point when the program is run.

Change the program from:

```
000  START INS 01                          000  START INS 01
001  TD = .125              TO             001  TD = .125
002  FR  XYZ = 10                          002  FR  XYZ = 10
003  SET UP→zcxyu                          003  GO  X  0.0
                                           004  GO  Y  0.0
                                           005  GO  Z  0.0
                                           006  SET UP→zcxyu
```

During the first SET UP ask the controller (DISPLAY COOD) for the coordinates of X,Y,& Z before setting them. Write these down and then enter them as moves in statements 003 - 005.

When the program is run the second time, the tool will move to the set up point. The user has only to confirm the set up to proceed.

3. WHAT DO I DO WHEN THE TOOL BREAKS?

Press the halt key. This will stop the axis at the end of that move, note the line number. Switch to line number mode, then to a tool-break program, which can be at the end of the current program. Run this program. EXAMPLE:

800	START INS 02	
801	Z ➔ Z MAX	
802	HALT	Put in new tool, slacken off spindle lock, raise spindle.
803	GO X 0	Go to some point where we can reference the
804	Y 0	new height.
805	GO Z 0	Lower head.
806	HALT	Adjust spindle to Z ➔ Z REF. Lock spindle.
807	Z ➔ CLEAR Z	
808	HALT	
809	END	

Exit at 808 to old line number in first program and continue running.

The user can re-enter at any point in the program, but care should be exercised. You cannot re-enter in functions, they have to be repeated. Repeat loops can be re-entered at the same location if the coordinates are known. These may have to be entered after 807 above. Usually re-starting in the middle of repeat loops can only be done if the tool is positioned accurately or the indexing will be off.

Careful in relative instructions unless position tool back to exact position (x y z) @ time of breakage & careful as that initself could break the new tool!

4. <u>WHAT DO I HAVE TO WATCH FOR ON PRODUCTION RUNS?</u>

When the user is satisfied on a one run operation he can then switch to a production run mode.

1. Use END NEW PART ending. This has built in verification of set up position.

2. Insert SPINDLE ON/OFF instructions.

3. Insert pre-SET-UP moves if necessary.

4. Insert PAUSES if necessary to clear work and buzzer to signal end new part

During running, the user must ensure the slideways are adequately lubricated. Also try to vacuum the debris out, not blow it out. This is particularly true of ceramic dust, graphite and metal parts.

If the set up point abruptly jumps then this is also due to limit switch contamination.

The set up on Y will drift negative (.0001" per degree C) initially due to the warm up time on the spindle head. The user should pre-run the head before starting to preserve the Y set up point if this is required.

SECTION 13
ERROR CODES

RUNTIME ERRORS

Certain errors can be checked at run time. Here is a list. The error message format is:

ERROR nn

The user can switch to LINE NO MODE when this is displayed to examine his program and check his entry.

00	X,Y,Z	AXIS DESTINATION BEYOND MAXIMUM TRAVEL or GO r too big.
01	X,Y,Z	AXIS DESTINATION BEYOND MAXIMUM TRAVEL.Thie occurs most often when doing outside cuts with or without finish option with the part clamped to close to the hom_ position, the part must be clamped to allow for the tool radius (+ option of 6.4 thous of finish cut) clear of the home position.
02	XA,XB	Too small for inside or on the line cut or tool dia-emeter too large.
03	XA,YB	Values must be positive.
04	X,Y	Axis beyond minimum for finish cut.
06		Tool diameter is zero.
07		Z% is zero.
08		Zd must be positive.
09	GO,r	r must be positive.
10		In circle function r1-r2 too small for tool diameter with or without finish option.

IT SEEMS IN CIRC. POCKET ETC IF r TAKES TOOL TO NEG VALUES OF X & Y THIS ERROR OCCURS

COMMON PROBLEMS AND THEIR SOLUTIONS

SOFTWARE

1. Program missing or parts of program data incorrect.- You are disconnecting controller from machine or desktop unit without first switching off power. This will scramble the memory.

2. Program runs occasionally off at random.- This is due to electrical noise coming down the line from other machines. Try another outlet. Always avoid outlets wired to large machines.

3. Drift in set-up reference zero position.- Occurs when you omit END NEWPART in program or use SKIP TOP before END NEWPART.

4. Inch values changed to metric.- May occur when line number 000 is ignored and program start is on a different line number of the memory stack.

HARDWARE

1. Sticking axis.- Inadequate lubrication is usually at fault. Run axis test (diagnostics in manual mode) to check axis while lubricating. Only ater exhausting other service procedures, should you adjust gibs.

2. Large variations in backlash measurments.- Contamination of limit switches. They should be free and clear of oil, dirt, grime and other debris. Large inaccurate backlash will produce circles skewed at 0 and 180 or 90 and 270 degrees.

3. Inoperative axis.- Loose axis plug at socket. A loose axis plug can cause intermittent operation or total axis failure.

4. No spindle operation.- Check circuit breaker under spindle belt cover. Check spindle on-off control for proper position. Check 3 amp and 10 amp fuse.

5. Noisy spindle.- Caused by a loose spindle belt. Tighten belt by adjusting position of motor to minimize belt noise.

SECTION 15
INDEX

APPENDIX 1

RS232-C INSTRUCTION FORMAT

INSTRUCTION		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
		LINE NUMBER			FORMAT												
START	(INCHES)	*	*	*		S	T	A	R	T		I	N	S		n	n
	(mm)	*	*	*		S	T	A	R	T		M	M			n	n
TOOL DIAMETER	(INCHES) FORMAT	*	*	*			T	D	=		n	n	.	n	n	n	n
	(mm) FORMAT	*	*	*			T	D	=		m	m	m	.	m	m	m
FEED RATE	(INCHES)	*	*	*		F	R		(X)	(Y)	(Z)	(U)	=	n	n	.	n
	(mm)	*	*	*		F	R		(X)	(Y)	(Z)	(U)	=	n	n	.	n
SET UP		*	*	*		S	E	T	U	P		>	z	c	x	y	u
GO ABS	1 AXIS MOVE	*	*	*		G	O R	(q)	(X) (Y) (Z) (r) (u)	(S)	n m	n m	. m	n .	n m	n m	n m
GO REL									(a) (φ)		d	d	d	.	d	d	d
(q) ONLY ALLOWED ON 1 AXIS MOVE	2 AXIS MOVE ABOVE +	*	*	*+1					(X) (Y) (Z) (r)	(S)	n m	n m	. m	n .	n m	n m	n m
									(a)		d	d	d	.	d	d	d
q = { i, O, BLANK	3 AXIS MOVE ABOVE +	*	*	*+2						(S)	n m	n m	. m	n .	n m	n m	n m
Z ⟶ Z CLEAR		*	*	*		Z	>	C									
Z ⟶ Z MAX		*	*	*		Z	>	C	M	A	X						
ZERO COODS		*	*	*		Z	E	R	O					(X)	(Y)	(Z)	(U)
ZERO AT		* * * *	* * * *	* *+1 *+2 *+3		Z	E	R	O (X) (Y) (Z)	 (S) (S) (S)	A n n n	T n n n	 . . .	 n n n	 n n n	 n n n	 n n n
CS		*	*	*		C	S	I	G	N		r	a	X	Y	Z	U
CYCLE		*	*	*		C	Y	C	L	E		X	Y				
⟶ REF COODS		*	*	*		>	R	E	F		C	O	O	D	S		
XY ⟶ REF O		*	*	*		X	Y	>	R	E	F		O				
PROG REF		*	*	*		P	R	O	G	.	R	E	F				

RS232-C INSTRUCTION FORMAT COTD

	LINE NUMBER				FORMAT											
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
CONTROL (BUZZER)	*	*	*		C	O	N	T	R	O	L					1
(PULSE OUT)	*	*	*		C	O	N	T	R	O	L					2
(PULSE IN)	*	*	*		C	O	N	T	R	O	L					3
SPINDLE OFF	*	*	*		S	P	I	N	D	L	E			O	F	F
ON	*	*	*		S	P	I	N	D	L	E				O	N
DISPLAY	*	*	*		D	S	P	L	Y		(r)	(a)	(X)	(Y)	(Z)	(U)
DWELL	*	*	*		D	W	E	L	L						n	n
NOP (NO INSTRUCTION)	*	*	*													
END END	*	*	*		E	N	D									
END NEWPART	*	*	*		E	N	D		N	E	W	P	A	R	T	
END NEW REF	*	*	*		E	N	D		N	E	W		R	E	F	
.ELSEWHERE	*	*	*+1				N	=							n	n
POINT 1	*	*	*+2			X	1	=		n	n	.	n	n	n	n
	*	*	*+3			Y	1	=		n	n	.	n	n	n	n
POINT 2	*	*	*+4			X	1	=		n	n	.	n	n	n	n
	*	*	*+5			Y	1	=		n	n	.	n	n	n	n
POINT nn						X	1	=		n	n	.	n	n	n	n
	*	*	*			Y	1	=		n	n	.	n	n	n	n
.REPEAT	*	*	*			R	E	P	E	A	T		X		n	n
	*	*	*			X	i	=		n	n	.	n	n	n	n
	*	*	*			R	E	P	E	A	T		Y		n	n
	*	*	*			Y	i	=		n	n	.	n	n	n	n
SKIP TO	*	*	*		S	K	I	P		T	O			n	n	n
CALL	*	*	*		C	A	L	L		S	U	B			n	n
SUBROUTINE	*	*	*		S	U	B								n	n
SUB RETURN	*	*	*		S	U	B		R	E	T	U	R	N		
REPEAT	*	*	*		R	E	P	E	A	T					n	n
REPEAT END	*	*	*		R	E	P	E	A	T				E	N	D
HALT	*	*	*		H	A	L	T								

RS232-C INSTRUCTION FORMAT COTD

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	LINE NUMBER										FORMAT					
TOOL CHANGE — TOOL N	*	*	*		T	O	O	L			N					
TOOL CHANGE	*	*	*		T	C										
TOOL CHANGE (NEW COOD)	*	*	*		T	C		N	E	W		C	O	O	D	
X COOD	*	*	*			T	X	=	(S)	n	n	.	n	n	n	n
Y COOD	*	*	*			T	Y	=	(S)	n	n	.	n	n	n	n
FUNCTION — 00 (SCALE)	*	*	*		S	C	A	L	E		O	N				
	*	*	*				(X)			n	n	.	n	n	n	n
	*	*	*				(Y)			n	n	.	n	n	n	n
	*	*	*				(Z)			n	n	.	n	n	n	n
	*	*	*		S	C	A	L	O	F	F		(X)	(Y)	(Z)	
RECT FRAME	*	*	*		F	R	A	M		(F)	(q)	Z	%	n	n	n
	*	*	*			Z	H	=		n	n	.	n	n	n	n
	*	*	*			Z	d	=		n	n	.	n	n	n	n
	*	*	*			X	1	=		n	n	.	n	n	n	n
	*	*	*			Y	1	=		n	n	.	n	n	n	n
	*	*	*			X	A	=		n	n	.	n	n	n	n
	*	*	*			Y	B	=		n	n	.	n	n	n	n
ELSEWHERE (OPTION)	*	*	*			X	1	=								
	*	*	*			Y	1	=								
	*	*	*			X	A	=								
	*	*	*			Y	B	=								
REPEAT (OPTION)	*	*	*			R	E	P	E	A	T		X		n	n
	*	*	*			X	i	=		n	n	.	n	n	n	n
	*	*	*			R	E	P	E	A	T		Y		n	n
	*	*	*			Y	i	=		n	n	.	n	n	n	n
CIRCLE POCKET	*	*	*		C	I	R	C		(F)	(q)	Z	%	n	n	n
	*	*	*					X	Y	C	U	T	%	n	n	n
	*	*	*			Z	H	=		n	n	.	n	n	n	n
	*	*	*			Z	d	=								
	*	*	*			X	C	=								
	*	*	*			Y	C	=								
	*	*	*			r	1	=								
	*	*	*			r	2	=								
ELSEWHERE (OPTION)	*	*	*			X	C	=								
	*	*	*			Y	C	=								
	*	*	*			r	1	=								
	*	*	*			r	2	=								
REPEAT (OPTION)	*	*	*			R	E	P	E	A	T		X		n	n
	*	*	*			X	i	=			n	.	n	n	n	n
	*	*	*			R	E	P	E	A	T		Y		n	n
	*	*	*			Y	i	=			n	.	n	n	n	n

RS232-C INSTRUCTION FORMAT COTD

	LINE NUMBER			FORMAT												
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
BOLT CIRCLE	*	*	*		B	O	L	T		P	E	C	K	=	n	n
	*	*	*			Z	H	=		n	n	.	n	n	n	n
	*	*	*			Z	d	=		n	n	.	n	n	n	n
	*	*	*			X	C	=		n	n	.	n	n	n	n
	*	*	*			Y	C	=		n	n	.	n	n	n	n
	*	*	*			a	1	=		d	d	d	.	d	d	d
	*	*	*			n	=								n	n
	*	*	*			r	=			n	n	.	n	n	n	n
MILL (q) = i / o / blank	*	*	*		M	I	L	L			(q)	Z	%	n	n	n
	*	*	*			Z	H	=	(S)		n	.	n	n	n	n
	*	*	*			Z	d	=	(S)		n	.	n	n	n	n
	*	*	*			X	1	=	(S)		n	.	n	n	n	n
	*	*	*			Y	1	=	(S)		n	.	n	n	n	n
	*	*	*			X	2	=	(S)		n	.	n	n	n	n
	*	*	*			Y	2	=	(S)		n	.	n	n	n	n
ELSEWHERE (OPTION)	*	*	*			X	1	=								
	*	*	*			Y	1	=								
	*	*	*			X	2	=								
	*	*	*			Y	2	=								
REPEAT (OPTION)	*	*	*			R	E	P	E	A	T		X		n	n
	*	*	*			X	i	=		n	n	.	n	n	n	n
	*	*	*			R	E	P	E	A	T		Y		n	n
	*	*	*			Y	i	=		n	n	.	n	n	n	n
RECT POCKET	*	*	*		R	E	C	T		(F)	(q)	Z	%	n	n	n
	*	*	*					X	Y	C	U	T	%	n	n	n
	*	*	*			Z	H	=		n	n	.	n	n	n	n
	*	*	*			Z	d	=								
	*	*	*			X	1	=								
	*	*	*			Y	1	=								
	*	*	*			X	A	=								
	*	*	*			Y	B	=								
ELSEWHERE (OPTION)	*	*	*			X	1	=								
	*	*	*			Y	1	=								
	*	*	*			X	A	=								
	*	*	*			Y	B	=								
REPEAT (OPTION)	*	*	*			R	E	P	E	A	T		X		n	n
	*	*	*			X	i	=		n	n	.	n	n	n	n
	*	*	*			R	E	P	E	A	T		Y		n	n
	*	*	*			Y	i	=		n	n	.	n	n	n	n

RS232-C INSTRUCTION FORMAT COTD

	LINE NUMBER 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
ARC FRAME	*	*	*		A	R	C			(F)	(q)	Z	%	n	n	n
						Z	H	=	(S)		n	.	n	n	n	n
						Z	d	=								
NO ELSEWHERE						X	C	=								
NO REPEAT						Y	C	=								
						a	1	=								
						a	2	=								
							r	=								
DRILL	*	*	*		D	R	I	L		P	E	C	K	=	n	n
	*	*	*			Z	H	=	(S)		n	.	n	n	n	n
	*	*	*			Z	d	=								
	*	*	*			X	=									
	*	*	*			Y	=									
ELSEWHERE (OPTION)	*	*	*			X	=									
	*	*	*			Y	=									
REPEAT (OPTION)	*	*	*			R	E	P	E	A	T		X		n	n
	*	*	*			X	i	=			n	.	n	n	n	n
	*	*	*			R	E	P	E	A	T		Y		n	n
	*	*	*			Y	i	=			n	.	n	n	n	n

APPENDIX 2

L	M = MSD		0	1	2	3	4	5	6	7
	LSD		000	001	010	011	100	101	110	111
0	0	0000	NUL	DLE	SP	0	@	P	\	P
1	1	0001	SOH	DC1	!	1	A	Q	a	q
2	2	0010	STX	DC2	"	2	B	R	b	r
3	3	0011	ETX	DC3	#	3	C	S	c	s
4	4	0100	EOT	DC4	$	4	D	T	d	t
5	5	0101	ENQ	NAK	%	5	E	U	e	u
6	6	0110	ACK	SYN	&	6	F	V	f	v
7	7	0111	BEL	ETB	/	7	G	W	g	w
8	8	1000	BS	CAN	(	8	H	X	h	x
9	9	1001	HT	EM	)	9	I	Y	i	y
10	A	1010	LT	SUB	*	:	J	Z	j	z
11	B	1011	VT	ESC	+	;	K	[	k	{
12	C	1100	FF	FS	,	<	L	\	l	|
13	D	1101	CR	GS	–	=	M	]	m	}
14	E	1110	SO	RS	.	>	N	^	n	
15	F	1111	SI	US	/	?	O	_	o	DEL

Ascii code in decimal form $= L \times 16^0 + M \times 16^1$

$= L + 16M$

Exp:

M L

① 1A = SUB = 10 + 16 = 26

② 3E = > = 14 + 3×16 = 62

③ 51 = Q = 1 + 5×16 = 81

APPENDIX 2

L	M = MSD		0	1	2	3	4	5	6	7
	LSD		000	001	010	011	100	101	110	111
0	0	0000	NUL	DLE	SP	0	@	P	\	P
1	1	0001	SOH	DC1	!	1	A	Q	a	q
2	2	0010	STX	DC2	"	2	B	R	b	r
3	3	0011	ETX	DC3	#	3	C	S	c	s
4	4	0100	EOT	DC4	$	4	D	T	d	t
5	5	0101	ENQ	NAK	%	5	E	U	e	u
6	6	0110	ACK	SYN	&	6	F	V	f	v
7	7	0111	BEL	ETB	/	7	G	W	g	w
8	8	1000	BS	CAN	(	8	H	X	h	x
9	9	1001	HT	EM	)	9	I	Y	i	y
10	A	1010	LT	SUB	*	:	J	Z	j	z
11	B	1011	VT	ESC	+	;	K	[	k	{
12	C	1100	FF	FS	,	<	L	\	l	|
13	D	1101	CR	GS	–	=	M	]	m	}
14	E	1110	SO	RS	.	>	N	^	n	
15	F	1111	SI	US	/	?	O	_	o	DEL

Ascii code in decimal form $= L \times 16^0 + M \times 16^1$

$= L + 16M$

EXP:

M L

① 1A = SUB = 10 + 16 = 26

② 3E = > = 14 + 3×16 = 62

③ 51 = Q = 1 + 5×16 = 81